

فصل اول

# مروری بر کنترل دیجیتال

## مقدمه ای بر مبانی دیجیتال

در سالهای اخیر استفاده از کنترل کننده های دیجیتال در سیستم های کنترل افزایش یافته است . در واقع بسیاری از سیستم های کنترل صنعتی ، کنترل کننده های مبتنی بر پردازشگر را به عنوان یک جزء اساسی عملیات خود محسوب می نمایند . اخیراً کاربرد کنترل دیجیتال ، انجام اموری از قبیل بهینه سازی مصرف سوخت در اتومبیلها ، عملیات پیچیده در لوازم خانگی و ماشین آلات مانند دستگاههای CNC ، دستگاههای ریسندگی و بافندگی و غیره را امکان پذیر ساخته است . از جمله مزایای سیستم های کنترل دیجیتال ، قابلیت تصمیم گیری و انعطاف پذیری در برنامه کنترل این چنین سیستم هایی می باشد . از جمله دلایل گرایش بسمت کنترل دیجیتال ، به جای کنترل آنالوگ سیستم های دینامیکی می توان به دسترس پذیر بودن کنترل کننده های دیجیتال ارزان قیمت و مزایای کار با سیگنالهای دیجیتال به جای سیگنالهای آنالوگ اشاره نمود . اجزاء گسسته اطلاعات در یک سیستم دیجیتال را کمیت های فیزیکی به نام سیگنال می سازند . سیگنال ها در تمام سیستم های دیجیتال الکترونیکی ، تنها دو مقدار مجزاء داشته و دودویی نامیده می شوند . لذا قدم اول در شناخت یک سیستم دیجیتال ، آشنایی با مفاهیم سیستم های دودویی می باشد .

## اعداد دودویی

اعداد را می توان در مبناهای عددی مختلف نمایش داد . آشنا ترین مبنا عددی ، مبنا ده می باشد . در مبنا ده ، کلیه اعداد با ترکیبی از عددهای ۰ تا ۹ حاصل می گردند . هر رقم یک عدد ، دارای ارزشی خاص می باشد که ما آنرا با عبارات یکان ، دهگان ، صدگان و ... می شناسیم . به عنوان مثال در عدد ۵۴۳ ، عدد ۳ در رتبه یکان  $(10^0)$  ، عدد ۴ در رتبه دهگان  $(10^1)$  و عدد ۵ در رتبه صدگان  $(10^2)$  قرار دارد . از دیگر مبناهای عددی رایج می توان به مبنا دو اشاره نمود . همانند اعداد مبنا ده ، هر رقم یک عدد در مبنا ۲ نیز دارای ارزش خاص خود می باشد . در این مبنا تنها اعداد صفر و یک موجود می یاشند . عدد ۱۰۱۰۱۰۱ را می توان یک عدد ۸ رقمی در مبنا ۲ نامید . هر رقم در مبنا ۲ ، یک بیت و هر ۸ بیت یک بایت نامیده می شوند . جهت بدست آوردن معادل مبنا دو یک عدد دهدهی ، این عدد را بطور متناوب بر ۲ تقسیم می نماییم تا جایی که خارج قسمت نهایی بر ۲ قابل تقسیم نباشد . ( یعنی یکی از اعداد صفر یا یک باشد ) . این عدد با ارزش ترین بیت ، معادل دودویی عدد را تشکیل می دهد . باقیمانده های بدست آمده از انتها تا ابتدا نیز بترتیب سایر بیت های پر ارزش معادل دودویی عدد را بدست می دهند .

مثال : معادل دودویی عدد ۱۴ عبارتست از:

$$\begin{array}{r} 14 \\ - 14 \\ \hline 0 \end{array} \quad \begin{array}{r} 2 \\ 7 \\ - 6 \\ \hline 1 \end{array} \quad \begin{array}{r} 2 \\ 3 \\ - 2 \\ \hline 1 \end{array} \quad \begin{array}{r} 2 \\ 1 \end{array} \quad \longrightarrow \quad (14)_{10} = (1110)_2$$



جهت تبدیل یک عدد از مبنای ۲ به مبنای ۱۰ می توان هر رقم را در ارزش مکانی خود ضرب نموده و سپس حاصلضرب های بدست آمده را با هم جمع نمود .

مثال : معادل مبنای ده عدد ۱۰۰۱۰۰۱۱ عبارتست از:

$$(10010011)_2 = 1 \times 2^0 + 1 \times 2^1 + 0 \times 2^2 + 0 \times 2^3 + 1 \times 2^4 + 0 \times 2^5 + 0 \times 2^6 + 1 \times 2^7$$

$$= 1 + 2 + 0 + 0 + 16 + 0 + 0 + 128 = (147)_{10}$$

از دیگر مبناهای عددی پر کاربرد مبنای ۱۶ می باشد . یک عدد در مبنای ۱۶ معادل یک عدد دودویی چهار رقمی است . جدول ارائه شده در شکل ۱-۱ اعداد مبنای شانزده و معادل دهمی و دودویی آنها را نمایش می دهد .

مبنای ۱۰	مبنای ۲	مبنای ۱۶
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
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

شکل ۱-۱: نمایش اعداد در مبناهای مختلف

جهت تبدیل یک عدد از مبنای ۱۶ به مبنای ۲ به جای هر عدد آن معادل دودویی چهاررقمی معادل را جایگزین می نماییم .

مثال : معادل عدد ۱۶ (5E2) در مبنای دو عبارتست از :

$$(5E2)_{16} = (\underline{0101} \underline{1110} \underline{0010})_2$$

جهت تبدیل یک عدد مبنای ۲ به مبنای ۱۶ نیز کافی است از سمت راست اعداد را چهاررقم ، چهاررقم جدا نموده و سپس معادل مبنای ۱۶ آنها را جایگزین نماییم .

مثال : عدد مبنای دو ۱۰۱۱۰۱ در مبنای ۱۶ عبارتست از:

$$(\underline{0010} \underline{1101})_2 = (2D)_{16}$$



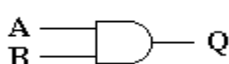
در این منطق، متغیرها دارای دو ارزش مجزا هستند که ما آنها را با ارزش یک و صفر و یا سطوح منطقی بالا و پایین معرفی می‌نمائیم. مدارهای طرح شده بر این اساس را که در آن تمام متغیرهای بکار برده شده تنها دارای دو مقدار می‌باشند را مدارات منطقی یا دیجیتال می‌نامند.

در منطق دیجیتال سه نوع عملیات اصلی وجود دارد. AND، OR و NOT.

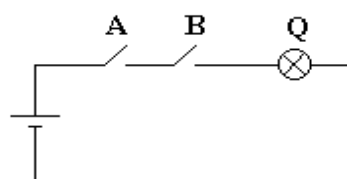
۱- عملگر AND: این عملگر همانند کلیدهای سری در مدارات الکتریکی تعریف می‌گردد. به عبارت دیگر در یک عملگر AND تنها زمانی خروجی خواهیم داشت که تمام متغیرهای موجود دارای ارزش یک باشند. در شکل ۱-۲ مدار سوئیچینگ نمایش دهنده عملگر AND، سمبل مداری این گیت و جدول عملکرد آن ارائه شده است.

A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

شکل ج



شکل ب



شکل الف

شکل ۱-۲: عملگر AND شکل الف: مدار سوئیچینگ شکل ب: سمبل مداری شکل ج: جدول عملکرد

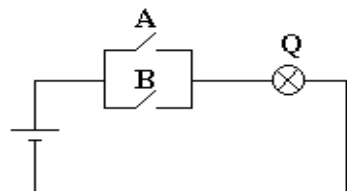
۲- عملگر OR: این عملگر همانند کلیدهای موازی در مدارات الکتریکی تعریف می‌گردد. در این عملگر هرگاه حداقل یکی از ورودی‌ها یک باشد خروجی دارای ارزش یک خواهد بود. در شکل ۱-۳ مدار سوئیچینگ نمایش دهنده عملگر OR، سمبل مداری این گیت و جدول عملکرد آن ارائه شده است.

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

شکل ج



شکل ب



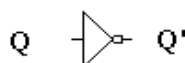
شکل الف

شکل ۱-۳: عملگر OR شکل الف: مدار سوئیچینگ شکل ب: سمبل مداری شکل ج: جدول عملکرد

۳- عملگر NOT: NOT یک عملگر تک ورودی است که خروجی آن مکمل ورودی می‌باشد. به عبارت دیگر هرگاه ورودی یک باشد خروجی صفر خواهد بود و بالعکس. شکل ۱-۴ سمبل مداری و جدول عملکرد این گیت را نمایش می‌دهد.

Q	Q'
0	1
1	0

شکل ب



شکل الف

شکل ۱-۴: عملگر NOT شکل الف: سمبل مداری شکل ب: جدول عملکرد



سایر عملگرهای منطقی پایه که با استفاده از سه عملگر اصلی ساخته می شوند عبارتند از:  
 ۱- عملگر NAND: این عملگر از ترکیب عملگرهای AND و NOT ساخته شده و عملکرد آن برخلاف AND می باشد ، یعنی در این عملگر خروجی تنها زمانی مقدار صفر دارد که هر دو ورودی یک باشند . شکل ۵-۱ مشخصات این گیت را نمایش می دهد .

A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

شکل ب



شکل الف

شکل ۵-۱: عملگر NAND شکل الف: سمبل مداری شکل ب: جدول عملکرد

۲- عملگر NOR: این عملگر از ترکیب عملگرهای OR و NOT ساخته شده و عملکرد آن برخلاف OR می باشد ، یعنی در این عملگر خروجی تنها زمانی مقدار یک دارد که هر دو ورودی صفر باشند . شکل ۶-۱ مشخصات این گیت را نمایش می دهد .

A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

شکل ب



شکل الف

شکل ۶-۱: عملگر NOR شکل الف: سمبل مداری شکل ب: جدول عملکرد

۳- عملگر XOR: در این عملگر خروجی زمانی یک خواهد بود که تعداد ورودیهای یک ، فرد باشند. شکل ۷-۱ مشخصات این گیت را نمایش می دهد .

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

شکل ب



شکل الف

شکل ۷-۱: عملگر XOR شکل الف: سمبل مداری شکل ب: جدول عملکرد

۴- عملگر XNOR: در این عملگر که از ترکیب XOR و NOT ساخته می شود خروجی زمانی یک خواهد بود که تعداد ورودیهای یک ، زوج باشند. شکل ۷-۱ مشخصات این گیت را نمایش می دهد .

A	B	Q
0	0	1
0	1	0
1	0	0
1	1	1

شکل ب



شکل الف

شکل ۷-۱: عملگر XNOR شکل الف: سمبل مداری شکل ب: جدول عملکرد



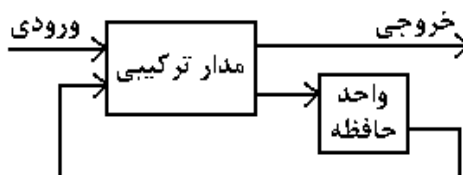
عملگرهای فوق الذکر را با استفاده از توابع منطقی نیز می توان بیان نمود . جدول ارائه شده در شکل ۸-۱ این توابع را نمایش می دهد .

عملگر منطقی	تابع منطقی متناظر
AND	$Q = A \cdot B$
OR	$Q = A + B$
NAND	$Q = (A \cdot B)'$
NOR	$Q = (A + B)'$
XOR	$Q = A'B + AB'$
XNOR	$Q = AB + A'B'$

شکل ۸-۱ : عملگرهای منطقی و توابع منطقی متناظر با آنها

### مدارهای منطقی

مداراتی که در آن تمام متغیرها دارای دو مقدار بوده و بوسیله عملگرهای منطقی به هم مرتبط می گردند را مدار منطقی می نامیم . مدارهای منطقی به دو دسته کلی تقسیم می گردند . مدارهای ترکیبی و مدارهای ترتیبی . در مدارات ترکیبی خروجی لحظه فعلی ، به ورودی در همان لحظه بستگی دارد . به عبارت دیگر هر ورودی اعمال شده به سیستم ، خروجی متناظر خود را تولید می نماید . ولیکن در مدارات ترتیبی حالت فعلی خروجی علاوه بر وضعیت فعلی ورودی ها به وضعیت قبلی خروجی نیز بستگی دارد . یعنی خروجی مدار که در لحظه های قبل بدست آمده و در یک واحد حافظه ذخیره گردیده است ، بر وضعیت فعلی خروجی اثر می گذارد . شکل ۹-۱ : بلوک دیاگرام یک مدار ترتیبی را بیان می نماید که از یک مدار ترکیبی ساخته شده است .



شکل ۹-۱ : بلوک دیاگرام یک مدار ترتیبی

کوچکترین عنصر حافظه در یک مدار ترتیبی را فلیپ فلاپ می نامند . یک فلیپ فلاپ قادر است مادامی که ورودی هایش تغییر نکرده و جریان تغذیه آن نیز قطع نشده باشد یک مقدار را بمدت نامحدود حفظ نماید . انواع مختلفی از فلیپ فلاپ وجود دارد که عبارتند از فلیپ فلاپ نوع D ، فلیپ فلاپ نوع JK ، فلیپ فلاپ نوع RS و ....

فلیپ فلاپ RS که مدار پایه سایر فلیپ فلاپها نیز می باشد ، دارای دو ورودی S (Set) و R (Reset) می باشد . در وضعیت  $SR = 10$  خروجی فلیپ فلاپها یک می گردد و در وضعیت  $SR = 01$  خروجی صفر می گردد .



در انتهای این مبحث چند واژه را که در این کتاب از آنها استفاده می گردد ، تعریف می نمایم.

۱- پالس : ایجاد یک تغییر در سطوح ولتاژ را پالس می نامیم. پالسها به دو گروه تقسیم بندی می گردند . پالس دائم یا موقت .

پالس دائم عبارتست از یک شکل موج مربعی که دارای دو سطح صفر و یک منطقی بوده و با فرکانس خاصی نوسان می کند .

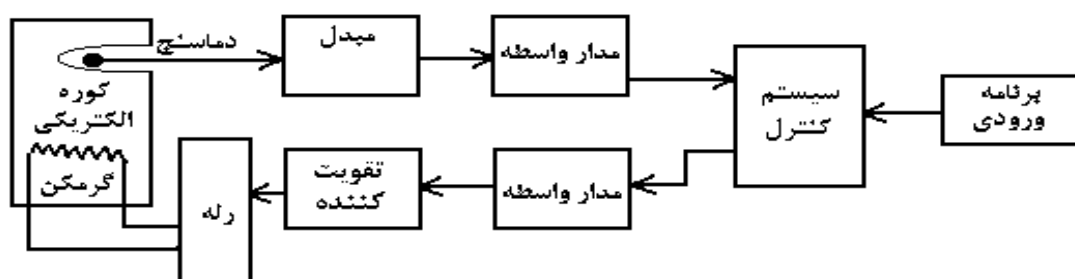
پالس موقت عبارتست از یک شکل موج که بمدت معینی در یکی از سطوح قرار گرفته و سپس به حالت دائمی باز می گردد .

۲- تریگر نمودن ( Trigger ) : واژه تریگر به معنای تحریک بوده و به این معنی است که یک واحد را با اعمال یک پالس وادار به واکنش مشخصی نمایم .

۳- لبه بالا رونده و لبه پایین رونده : هر تغییر وضعیت پالس از سطح بالا به سطح پایین را یک لبه پایین رونده نامیده و هر تغییر وضعیت پالس از سطح پایین به سطح بالا را لبه بالا رونده می نامیم .

### مقدمه ای بر سیستم های کنترل :

بنا بر تعریف یک سیستم کنترل ، سیستمی است که با دریافت اطلاعات ورودی ، نسبت به ایجاد خروجی هایی مطابق با خواسته های کاربر اقدام نماید . به عنوان مثال می توان به سیستم کنترل درجه حرارت یک کوره الکتریکی اشاره نمود . درجه حرارت داخل کوره توسط دماسنج که وسیله ای آنالوگ است ، سنجیده می شود . مبدل A/D سیگنال آنالوگ دما را به یک سیگنال دیجیتالی دما تبدیل می کند . این سیگنال از طریق مدار واسط به کنترل کننده داده می شود . دمای دیجیتال با دمای برنامه ریزی شده در ورودی مقایسه می شود و در صورت وجود اختلاف ( خطا ) سیگنالی از طریق کنترل کننده به کوره ، باز هم از طریق یک مدار واسط داده شده و رله دما را به حد مطلوب می رساند . شکل ۱-۱۰ این سیستم را بصورت بلوک دیاگرامی نمایش می دهد .



شکل ۱-۱۰ : سیستم کنترل حرارت

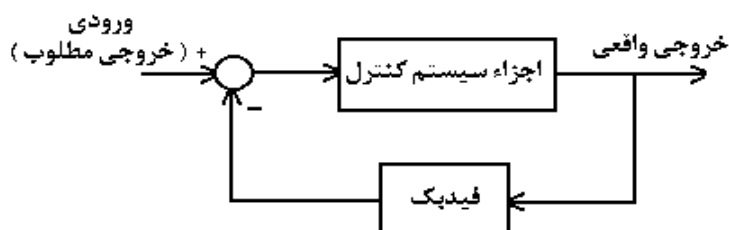
با توجه به مثال فوق الذکر می توان گفت هر سیستم کنترل از سه بخش اصلی تشکیل می گردد .

- ۱- ورودی ها
- ۲- اجزاء سیستم کنترل
- ۳- خروجی ها



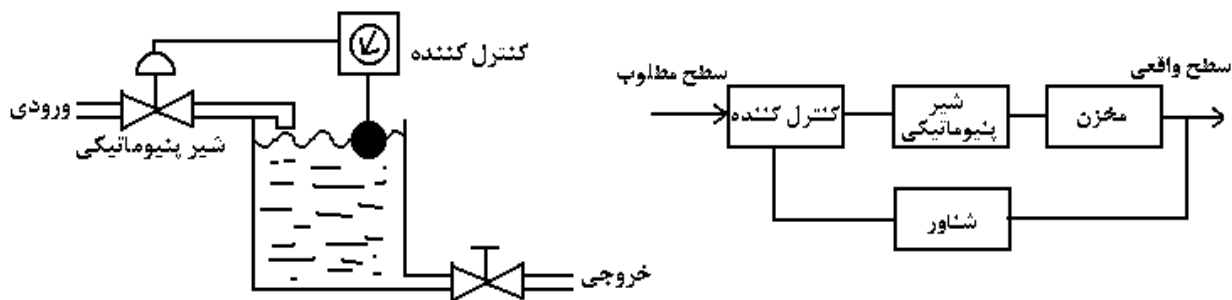
### سیستم های کنترل فیدبک دار

سیستمی که برای ایجاد ارتباط مطلوب بین خروجی و ورودی مرجع از مقایسه آنها و تفاضلشان استفاده می کند ، سیستم کنترل فیدبک دار نامیده می شود . سیستم کنترل درجه حرارت که در مثال قبل بیان شد نمونه ای از چنین سیستمی است . اینگونه سیستم های کنترل فیدبک دار را سیستم های کنترل حلقه بسته نیز می نامند . منظور از کنترل حلقه بسته استفاده از عمل فیدبک برای کاهش خطای سیستم است . شکل ۱-۱۱ چنین سیستمی را نمایش می دهد .



شکل ۱-۱۱ : سیستم کنترل فیدبک دار

مثال : شکل ۱-۱۲ نمودار یک سیستم کنترل سطح مایع را نشان می دهد . در این سیستم کنترل کننده با مقایسه سطح مایع و سطح مطلوب سیگنالی برای تنظیم شیر پنیوماتیکی ارسال می کند تا خطا اصلاح گردد .



شکل ب

شکل الف

شکل ۱-۱۲ : سیستم کنترل سطح مایع      شکل الف : نمودار بلوکی      شکل ب : نمای شماتیک

### سیستم های کنترل حلقه باز

سیستم هایی که در آنها خروجی بر عمل کنترل تاثیر ندارد سیستم های کنترل حلقه باز نامیده می شوند . به عبارت دیگر خروجی سیستم کنترل حلقه باز نه اندازه گیری می شود ، نه برای مقایسه با ورودی فیدبک می شود . یک مثال ، نوعی ماشین لباسشویی است . خیس کردن ، شستن و آبکشی بر اساس یک زمانبندی از قبل معلوم ، انجام می شود و ماشین سیگنال خروجی را که تمیزی لباسهاست اندازه گیری نمی کند . در سیستم های حلقه باز خروجی با ورودی مرجع مقایسه نمی شود ، پس به ازاء هر ورودی مرجع یک شرایط کاری ثابت وجود دارد . بنابراین دقت سیستم به تنظیم آن بستگی دارد . هر سیستم کنترلی که بر اساس زمانبندی کار می کند حلقه باز است . چراغهای راهنمایی که بر اساس زمانبندی کار می کنند نمونه دیگری از کنترل حلقه باز هستند .



### سیستم های کنترل دیجیتال

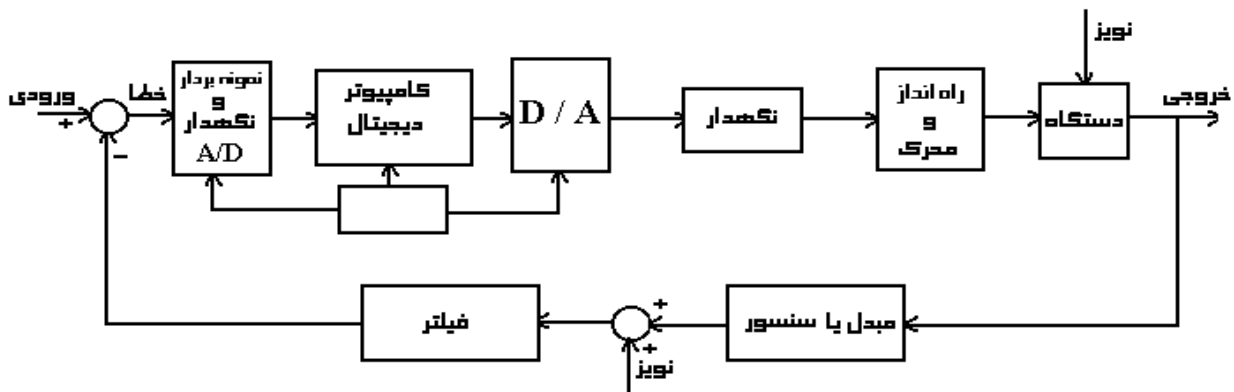
روش کنترل را که در آن برای انجام پردازش سیگنال با شیوه مطلوب ، کامپیوتر دیجیتالی در حلقه کنترل گنجانیده می شود ، کنترل دیجیتال مستقیم نامند . کنترل دیجیتال مستقیم یک فرآیند یا دستگاه در مقایسه با کنترل آنالوگ متناظر دارای مزایای زیر است .

۱- پردازش داده ها در کنترل کننده های دیجیتال سر راست بوده و محاسبات کنترلی پیچیده را می توان بسهولة انجام داد .

۲- در این روش برنامه های کنترل ( مشخصه های کنترل کننده ) را می توان بسادگی تغییر داد .

۳- کنترل کننده های دیجیتال نویز داخلی کمتری نسبت به کنترل کننده های آنالوگ دارا می باشند .

دیگرام بلوکی یک سیستم کنترل دیجیتال در شکل ۱-۱۳ ارائه شده است . در این سیستم مدار نمونه بردار و نگهدار از سیگنال آنالوگ خطا نمونه برداری کرده و سپس توسط مبدل آنالوگ به دیجیتال ، سیگنال آنالوگ به دیجیتال تبدیل می گردد . این سیگنال وارد کامپیوتر دیجیتال شده و پس از پردازش ، اطلاعات دیجیتال خروجی به مدار مبدل دیجیتال به آنالوگ وارد می گردد . اطلاعات آنالوگ خروجی پس از تقویت به دستگاههای کنترل کننده وارد شده و تغییرات لازم را به خروجی ها اعمال می نماید .



شکل ۱-۱۳: بلوک دیگرام سیستم کنترل دیجیتال

در انتهای این مبحث اصطلاحات بکار رفته در یک سیستم کنترل دیجیتال را تعریف می نمایم .

#### مدار نمونه بردار و نگهدار

نمونه بردار و نگهدار یک اصطلاح کلی است که در مورد یک تقویت کننده ، نمونه بردار و نگهدار بکار می رود . این عبارت مداری را توصیف می کند که سیگنال ورودی آنالوگی را دریافت کرده و آنرا بصورت یک مقدار ثابت برای مدت زمان مشخص نگه می دارد . معمولاً سیگنال بصورت الکتریکی است ، اما به شکلهای دیگر نیز مانند نوری و مکانیکی امکان پذیر است .

#### مبدل آنالوگ به دیجیتال

این مبدل که یک رمز گذار نیز خوانده می شود ، دستگاهی است که یک سیگنال آنالوگ را به سیگنال دیجیتال که معمولاً یک سیگنال رمز شده عددی است تبدیل می کند . چنین تبدیلی به عنوان واسط میان قطعه آنالوگ و قطعه دیجیتال لازم است .





### مبدل دیجیتال به آنالوگ

این مبدل که رمز گشا نیز خوانده می شود عملی عکس مبدل آنالوگ به دیجیتال را انجام می دهد .

### دستگاه یا فرآیند

دستگاه ، شی فیزیکی است که باید کنترل شود . مثالهایی برای یک دستگاه عبارتند از : کوره ، برخی دستگاههای خانگی ، دسته ای از اجزاء ماشین که برای انجام عمل خاصی با هم بکار می روند و ...

### مبدل یا سنسور

دستگاهی است که سیگنال ورودی را به سیگنال خروجی به شکل دیگری تبدیل می کند . مثلاً دستگاهی که سیگنال فشار را به خروجی ولتاژ تبدیل می کند نوعی سنسور است . سنسورها انواع مختلفی دارند که جهت تبدیل انواع مختلف سیگنالهای غیر الکتریکی به سیگنال الکتریکی متناظر بکار می روند .

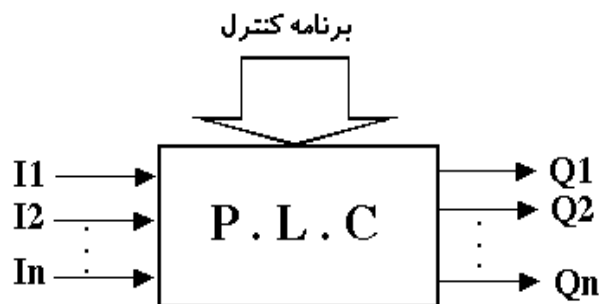


فصل دوم

# کنترل کننده های منطقی برنامه پذیر

## کنترل کننده منطقی برنامه پذیر :

پیشرفت های چشمگیر فن آوری نیمه هادی در زمینه ساخت ریزپردازنده و حافظه های با حجم بالا امکان ساخت کنترل کننده های منطقی الکترونیکی برنامه پذیر را فراهم آورد . در این کنترل کننده ها برخلاف کنترل کننده های مبتنی بر قسمت های الکترومکانیکی ، برای تغییر منطق کنترل کافی است بدون تغییری در سیم کشی یا قطعات ، فقط برنامه کنترل را تغییر دهیم ، در این صورت می توانیم از یک کنترل کننده منطقی برنامه پذیر هر جا که خواسته باشیم استفاده نماییم . شکل ۱-۱ یک کنترل کننده منطقی برنامه پذیر را بگونه نمایشی تعریف می نماید .



شکل ۱-۱ : شمای کلی یک کنترل کننده منطقی برنامه پذیر

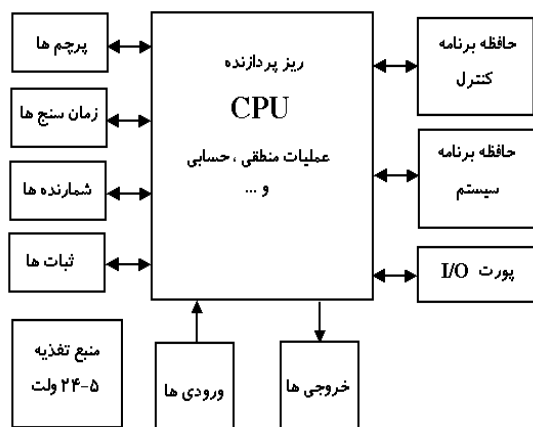
مزایای استفاده از کنترل کننده های منطقی :

- ۱- استفاده از PLC حجم تابلوهای فرمان را کاهش می دهد .
- ۲- استفاده از PLC مخصوصاً در فرآیندهای پیچیده موجب صرفه جویی فراوان در هزینه می گردد .
- ۳- PLC استهلاک مکانیکی ندارد ، بنابراین علاوه بر طول عمر بیشتر نیازی به سرویس و تعمیرات دوره ای ندارد .
- ۴- مصرف انرژی PLC بسیار کمتر از مدارهای رله ای است .
- ۵- PLC نویزهای صوتی و الکتریکی ایجاد نمی کند .
- ۶- طراحی و اجرای مدارهای کنترل منطقی با PLC آسان و سریع است .
- ۷- ایجاد تغییرات ( Modifications ) و تنظیمات در PLC آسان و سریع است .
- ۸- عیب یابی مدارات کنترل و فرمان با PLC سریع و آسان است و معمولاً PLC خود دارای برنامه عیب یابی می باشد .



**ساختمان داخلی PLC**

ساختمان داخلی یک PLC کم و بیش مانند ساختمان داخلی هر سیستم ریزپردازنده دیگر است. شکل ۲-۱ حالت کلی مربوط به ساختمان داخلی یک PLC را بیان می نماید.

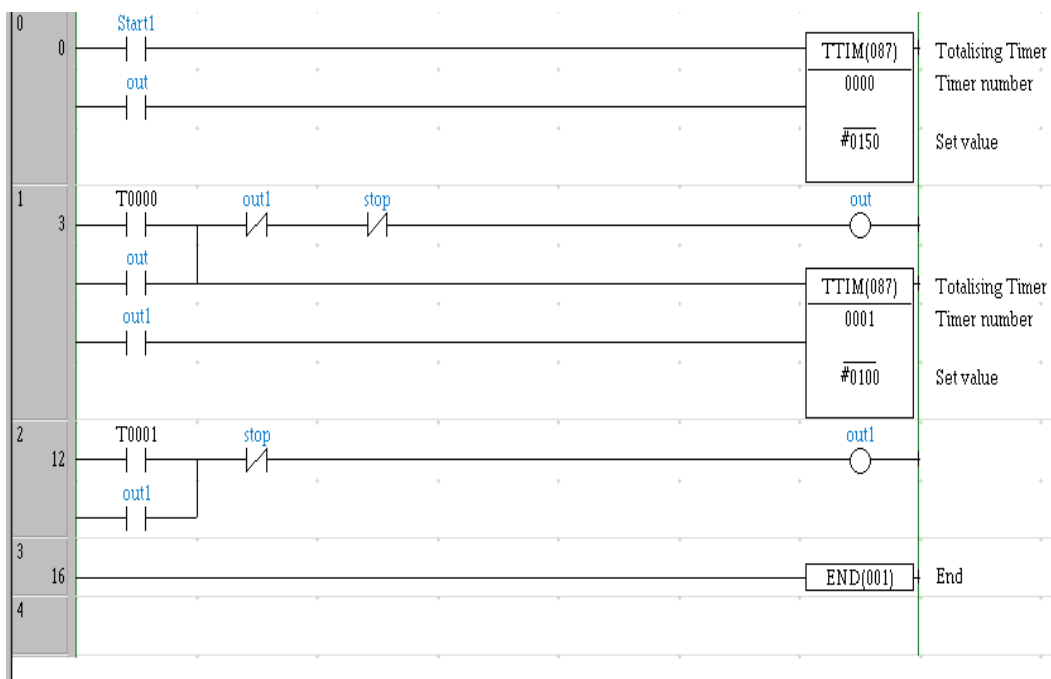


شکل ۲-۱: ساختمان داخلی PLC

**روش و زبان برنامه نویسی PLC**

هر PLC دارای زبان برنامه نویسی خاص خود بوده که رابط مابین کاربر و سخت افزار PLC می باشد. بوسیله برنامه کنترل است که یک PLC پروسه مورد نظر را کنترل می نماید. از آنجا که مهمترین گروه علمی \_ شغلی مرتبط با PLC گروههای مرتبط با مهندسی برق می باشند لذا سازندگان PLC اقدام به طراحی زبانهای برنامه نویسی خاصی نمودند که به دانسته های قبلی این گروه کاری نزدیکتر باشد. مهمترین روشهای برنامه نویسی عبارتند از: برنامه نویسی به روش نردبانی ( Ladder )

از آنجا که تمام نقشه های کنترل و فرمان منطقی قبل از ظهور PLC ها به صورت نردبانی و یا چیزی شبیه به آن تهیه و طراحی می شد، لذا سازندگان PLC این روش برنامه نویسی را بعنوان یکی از روشهای ممکن برنامه نویسی انتخاب نمودند. شکل ۳-۱ یک نمونه برنامه نویسی به زبان LAD را نمایش می دهد. 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 به ترتیب از چپ به راست و از بالا به پایین انجام می گردند.

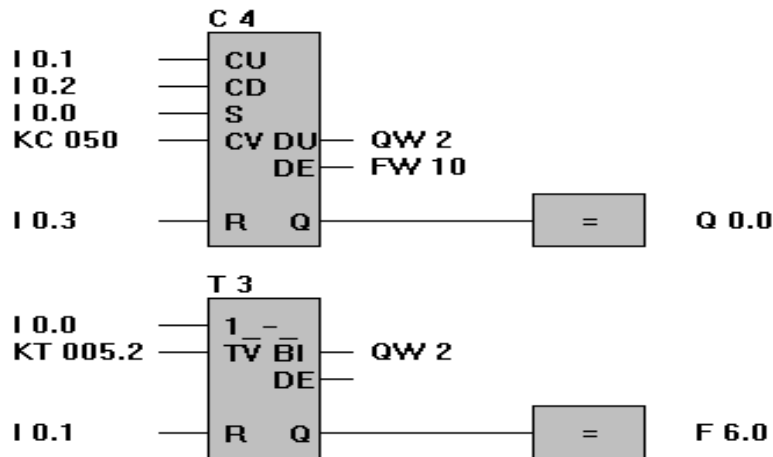


شکل ۳-۱: یک برنامه نوشته شده به زبان LAD در PLC های خانواده CS1G از شرکت OMRON



برنامه نویسی به روش فلوچارتی (Control System Flowchart) CSF  
یا نمایش جعبه ای تابع (Function Block Diagram) FBD

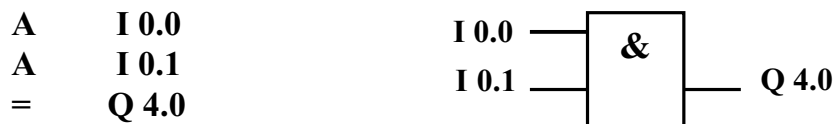
در این روش برنامه بصورت بلوکی نوشته شده که در آن هر بلوک بیانگر یک عملگر (Operation) می باشد. بدین ترتیب برنامه های نوشته شده به روش FBD عبارتند از یک سری جعبه که به یکدیگر متصل گردیده اند. روشهای فوق الذکر معمولاً بطور مستقل کاربرد چندانی ندارد و اغلب برای عیب یابی و یا شناخت منطق کنترل سیستم ناشناخته بسیار مفید است. شکل ۴-۱ یک نمونه برنامه نوشته شده به روش CSF را نمایش می دهد.



شکل ۴-۱: برنامه نمونه نوشته شده به روش CSF در PLC های خانواده S5 از شرکت Siemens

برنامه نویسی به روش لیست جملات (Statement List) STL

در این روش هر عمل منطقی توسط یک جمله یا عبارت مناسب نوشته می شود. مثال ارائه شده در شکل ۵-۱ نمونه ای از برنامه نوشته شده به روش STL را نمایش می دهد. در این مثال حرف A بیانگر دستور AND می باشد. 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. روش STL نیازهای گرافیکی بسیار کمتری نسبت به دو روش قبل دارد، لذا نوع و تعداد دستورات قابل درک و اجرا در این روش بسیار از روش های LAD و FBD می باشد. به همین دلیل برنامه هایی که به روش LAD یا FBD نوشته می شود معمولاً قابل تبدیل به STL می باشد در حالیکه عکس این قضیه همواره امکان پذیر نیست.



شکل ۵-۱: نمونه برنامه نوشته شده به روش STL و برنامه معادل آن به روش CSF

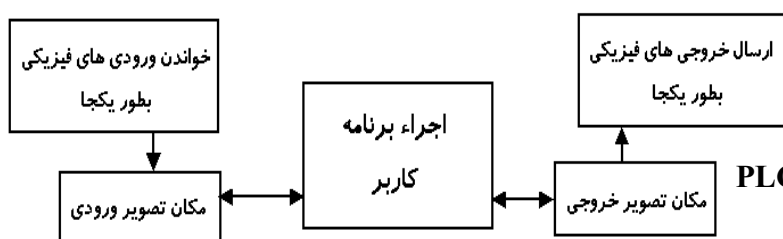


### کنترل کننده های منطقی برنامه پذیر امروزی

از اولین سالهای تولد PLC تاکنون بیش از سه دهه می گذرد. در این مدت شاهد تغییرات بسیار در ساختار PLC ها بوده ایم. از جمله این تغییرات می توان به افزایش سرعت عملکرد، توانایی کار با سیگنالهای آنالوگ و دیجیتال و همچنین برخوردار شدن از امکانات ارتباطی (Communication) سریع و ... اشاره نمود. برای برنامه نویسی PLC های قدیمی نیاز به یک Programmer مخصوص بود که این امر قیمت تمام شده یک سیستم کنترل منطقی با PLC را افزایش می داد. در حال حاضر امکان برنامه ریزی PLC ها با استفاده از کامپیوترهای شخصی فراهم گردیده است و این امر سهولت و صرفه جویی قابل ملاحظه ای را ایجاد نموده است.

### نمونه کار PLC

در ابتدای راه اندازی، مانند هر سیستم مبتنی بر پردازنده، در PLC نیز برنامه سیستمی اجرا می گردد. پس از اجرای برنامه سیستمی و چک شدن سخت افزار، در صورتی که شرایط لازم برای ورود به حالت اجرا (RUN) فراهم باشد، برنامه کاربر فرا خوانده می شود. برای اجرای برنامه کاربر ابتدا تمام ورودی های PLC بطور یکجا فرا خوانده می شود و وضعیت آنها (صفر یا یک) در مکانی بنام تصویر ورودی (Input - Image - Area) نوشته می شود. PLC در خلال اولین Scan برنامه، از داده های تصویر ورودی استفاده می نماید. توجه نمایید در صورتی که در طول اولین Scan، تغییراتی در ورودی ها حاصل شود، این تغییرات تا Scan بعدی به مکان تصویر ورودی ها منتقل نمی گردد. PLC ضمن Scan برنامه کاربر نتایج حاصل را در مکانی بنام تصویر خروجی (Output - Image - Area) می نویسد و بعد از اجرای کامل برنامه و در پایان، نتایج را بطور یکجا به خروجی ها ارسال می دارد. خواندن یکجای ورودی ها و ارسال یکجای خروجی ها، صرفه جویی قابل توجهی در زمان بدنال دارد، زیرا خواندن یا نوشتن با آدرس دهی یک به یک زمان زیادی را به خود اختصاص می دهد. از جمله مزایای دسترسی به مکانهای تصویر خروجی یا ورودی آن است که امکان Set یا Reset نمودن هر یک از بیت های ورودی یا خروجی را مستقل از وضعیت فیزیکی آنها فراهم می نماید و این کار مزیت بزرگی به هنگام عیب یابی یا آزمایش یک برنامه نوشته شده محسوب می شود. روش فوق در عین مزایایی که ذکر گردید، مسئله ای بنام زمان پاسخ دهی برنامه (Program Response Time) را بوجود می آورد. زمان پاسخ دهی مدت زمانی است که طول می کشد تا PLC تمام برنامه کاربر را Scan نماید و در این مدت تغییرات بوجود آمده در ورودی ها وارد مکان تصویر ورودی نمی گردد و خروجی ها نیز به حالتی که در Scan قبلی بودند باقی می ماند این امر در فرآیندهایی با سرعت تغییرات زیاد، مشکل ساز است مخصوصاً زمانی که برنامه کاربر طولانی بوده و مدت زمان زیادی صرف Scan برنامه می گردد. همچنین گاهی ملاحظات ایمنی لازم می دارد که تغییرات آنی بعضی از ورودی ها همواره مورد توجه قرار گیرد که در این صورت زمان پاسخ دهی ممکن است مانع از ثبت به موقع این تغییرات شود. برای حل این مشکل در زبانهای برنامه نویسی دستورات خاصی گنجانده شده است. با توجه به سرعت بالای PLC های امروزی و کندی فرآیندهایی که توسط آن کنترل می گردند (سیستم های الکترو مکانیکی) زمان پاسخ دهی در شرایط عادی، معمولاً مشکلی ایجاد نمی نماید. شکل ۱-۶ طرز کار PLC را بیان می دارد.



شکل ۱-۶: بیان اجراء برنامه کاربر توسط PLC

## فصل سوم

## نگاهی به دافل PLC

یک PLC در واقع کامپیوتری است که با آنچه احتمالاً درباره آن شنیده اید یا با آن کار کرده اید فرق دارد. بیشتر مردم با کامپیوترهای خانگی آشنایی دارند. نوع دیگری رایانه نیز وجود دارد که به عنوان رایانه کنترل فرآیند شناخته می شود. هر چند که این رایانه نیز داده ها را پردازش می کند ولی وظیفه اصلی آن کنترل فرآیندهای صنعتی و تولیدی است (ماشین آلات تولید، روباتها، خطوط تولید و ...).

هر چند که این رایانه ها ممکن است صفحه کلید نیز داشته باشند، ورودی های کنترل آنها سوئیچها و حسگرها هستند و خروجی های آنها علاوه بر نمایشگرها و چاپگر، سیگنالهای کنترلی برای انواع موتورها، سولنوئیدها و ... می باشند.

PLC ها، نوعی رایانه کنترل فرآیند هستند که کوچک، بطور نسبی ارزان قیمت و در برابر تغییرات شرایط فیزیکی محیط کار مقاوم بوده و به سهولت قابل برنامه ریزی، راه اندازی و تعمیر و نگهداری می باشند. آنها معمولاً در مجاورت ماشین آلات تولید یا فرآیندی که آنها را کنترل می کنند نصب شده و لذا معمولاً به عنوان توسعه تجهیزات کارخانه قلمداد می شوند.

وامدهای تشکیل دهنده PLC

در PLC های کوچک، پردازنده، حافظه نیمه هادی، ماژول های I/O و منبع تغذیه در یک واحد جای داده شده اند. در PLC های بزرگتر، پردازنده و حافظه در یک واحد، منبع تغذیه در واحد دوم و واسطه های I/O در واحدهای بعدی قرار دارند.

ابزار برنامه نویسی، که معمولاً یک واحد پردازنده با صفحه نمایش و صفحه کلید می باشد (بعنوان مثال یک کامپیوتر شخصی، یک PG در خانواده زیمنس و یا کنسول در خانواده Omron) به عنوان یک واحد مجزا از طریق یک سیم به واحد اصلی متصل می گردد.

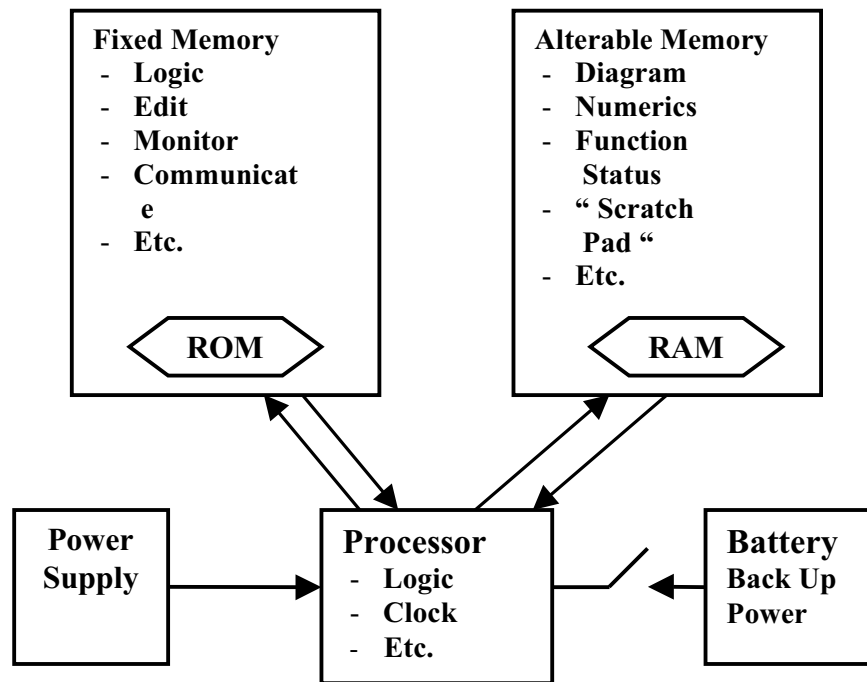
شکل ۳-۱ قسمتهای اصلی یک سیستم پردازش در PLC را نمایش می دهد.

حافظه ثابت سیستم، حاوی برنامه ای است که توسط کارخانه سازنده تعبیه شده است. این برنامه وظیفه ای مشابه سیستم عامل Dos در دستگاههای PC دارد که بر روی تراشه های خاصی بنام حافظه فقط خواندنی (ROM) قرار گرفته است. برنامه های ثابت در ROM، در حین عملیات CPU نمی توانند تغییر یابند یا پاک شوند. برنامه موجود در این حافظه غیر فرار به هنگام قطع تغذیه CPU نیز حفظ می شود.

اطلاعات حافظه تغییر پذیر بر روی تراشه های نیمه هادی ذخیره می شود که امکان برنامه ریزی، تغییر و پاک کردن آنها توسط برنامه ریز میسر است. این حافظه عمدتاً از نوع حافظه های با قابلیت دسترسی تصادفی (RAM) انتخاب می گردند. اطلاعات موجود در حافظه های RAM با قطع تغذیه، پاک می گردند.

**توجه: جهت کسب اطلاعات بیشتر در خصوص حافظه های نیمه هادی به ضمیمه ۳ مراجعه نمایید.**





شکل ۳-۱: واحدهای اصلی تشکیل دهنده سیستم پردازش در PLC

اغلب CPU ها مجهز به یک باتری پشتیبان هستند. بنابراین اگر تغذیه ورودی قطع شود و متعاقباً منبع تغذیه نتواند ولتاژ سیستم را تامین کند، باتری پشتیبان برنامه ذخیره شده در RAM را حفظ می کند. همانگونه که در شکل ۳-۱ ملاحظه می گردد، قسمت پردازنده دارای ارتباطاتی با قسمت های مختلف داخل و خارج خود می باشد.

### پردازنده

تمام پردازنده های رایانه ای، به گونه ای طراحی شده اند که بتوانند محاسبات منطقی و حسابی را انجام دهند. این عملیات بوسیله ریزپردازنده (Microprocessor) و از طریق بکارگیری دستورالعمل های متفاوت انجام می گیرد. ریزپردازنده ها بر حسب میزان قدرت طبقه بندی می گردند. دو عامل در تعیین میزان قدرت ریزپردازنده ها عبارتند از تعداد بیت ها و سرعت پالس ساعت (Clock). ریزپردازنده های فعلی امکان پردازش داده ها، بصورت ۴، ۸، ۱۶ یا ۳۲ بیتی را دارا می باشند. هرچه تعداد این بیتها بیشتر باشد، قدرت پردازنده بیشتر است. میزان پالس ساعت، سرعت اجرای هر دستورالعمل را نشان می دهد. محدوده سرعت پالس ساعت در حال حاضر از محدوده 1MHz تا 66MHz متغیر می باشد. جدول ارائه شده در شکل ۳-۲ تعدادی از میکروپروسسور های معروف را مقایسه نموده است. تعدادی از PLC ها از ریزپردازنده های عنوان شده در جدول فوق الذکر بهره می گیرند و تعدادی دیگر از آنها از CPU های انحصاری خود کارخانه سازنده استفاده می نمایند. با اختراع پردازنده های پنتیوم، نسل جدید PLC ها نیز با بکارگیری این پردازنده های سرعت بالا یا به عرصه صنعت گذاشته اند. از جمله این PLC ها می توان به خانواده های C7 و M7 از شرکت زیمنس اشاره نمود.

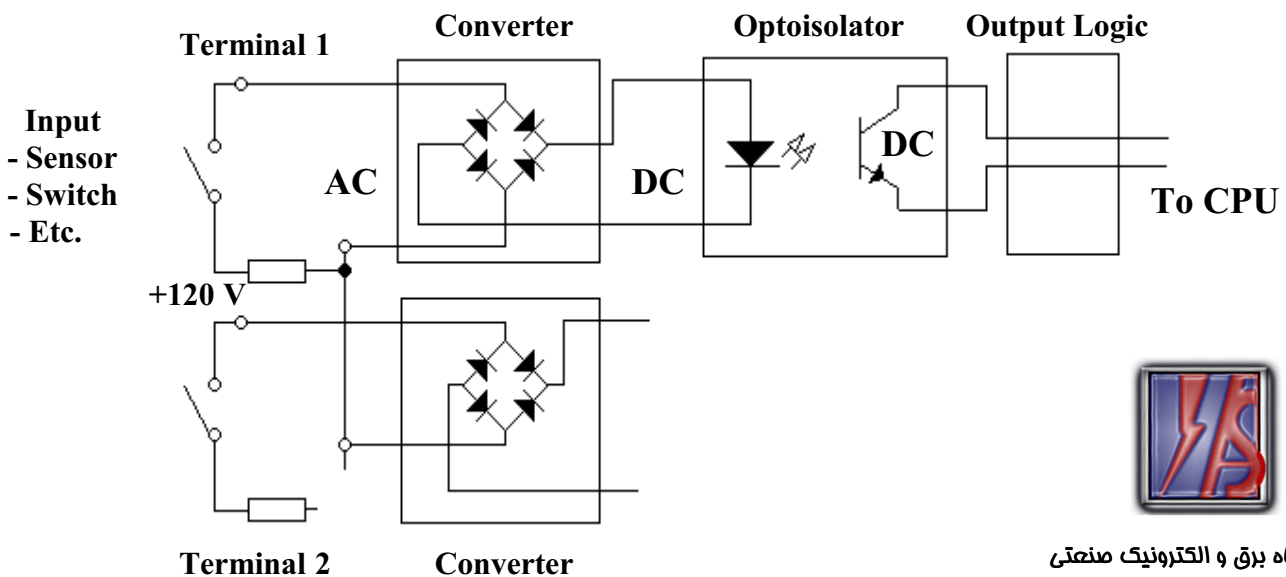


Microprocessor	Bit Size	Clock Speed
8085	8-bit	1MHz
8086	16-bit	4.77MHz
80186	16-bit	8MHz
80286	16-bit	12.5MHz
80386	32-bit	33MHz
80486	32-bit	50MHz

شکل ۳-۲: مقایسه تعدادی از ریز پردازنده های معروف

ماژول های ورودی و خروجی ( Input / Output )

ماژول ورودی به صورت الکترونیکی چهار کار اصلی را انجام می دهد. اولاً این ماژول حضور یا عدم حضور سیگنال الکتریکی در تمام ورودیها را بررسی می کند. این سیگنالهای ورودی، وضعیت قطع یا وصل سوئیچها، حسگرها و سایر عناصر در فرآیند تحت کنترل را نمایش می دهند. ثانیاً این ماژول سیگنال مربوط به وصل بودن را از نظر الکتریکی به سطحی DC که توسط مدارات الکترونیکی ماژول I/O قابل استفاده باشد، تغییر می دهد. برای سیگنال ورودی قطع، هیچ تبدیل سیگنالی صورت نمی گیرد و نشان دهنده حالت قطع است. ثالثاً این ماژول، جداسازی الکترونیکی را با جدا کردن خروجی ماژول ورودی از ورودی اش به صورت الکترونیکی انجام می دهد. در نهایت این ماژول سیگنالی را که توسط CPU سیستم PLC قابل تشخیص است، ایجاد می کند. تمام این وظایف در طرح شکل ۳-۳ نشان داده شده است.



آموزشگاه برق و الکترونیک صنعتی  
اندیشه سازان صنعت

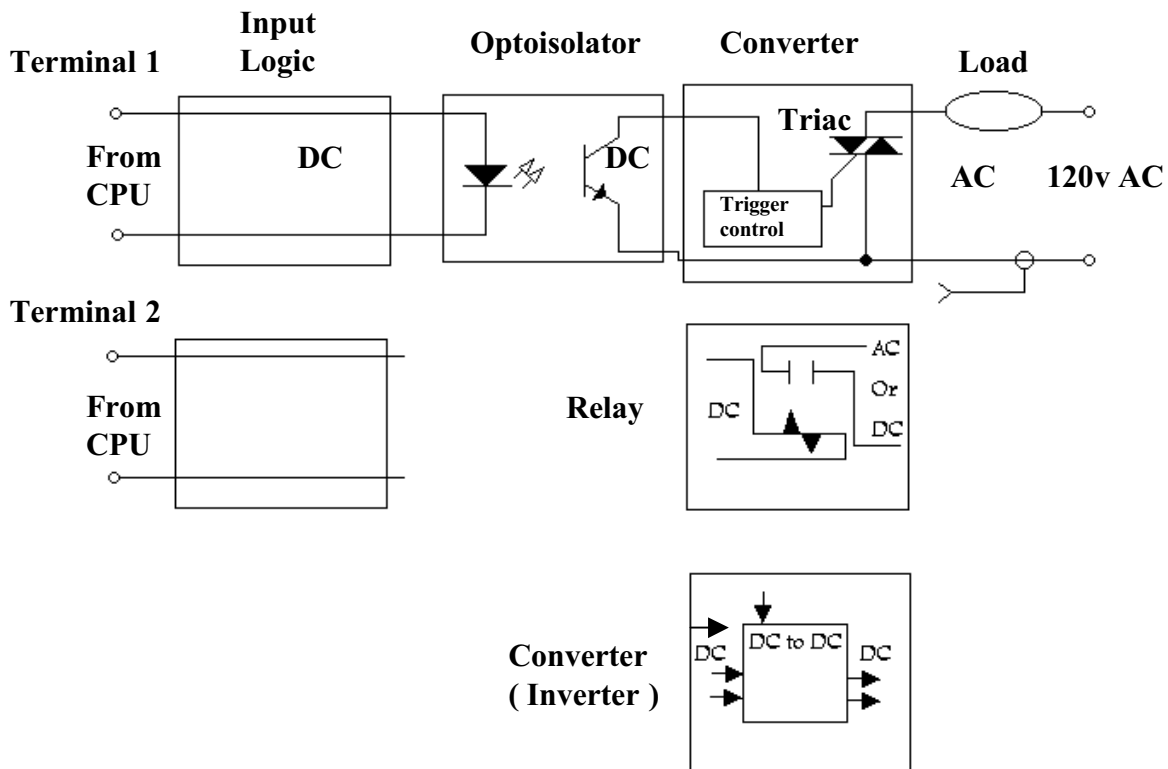
شکل ۳-۳: طرح ماژول ورودی PLC

یک نمونه ماژول ورودی می تواند ۴، ۶، ۸، ۱۲، ۱۶ یا ۳۲ ترمینال علاوه بر ترمینالهای مشترک و ارت، داشته باشد.



شکل ۳-۳ فقط مدار یک ترمینال رانمایش می دهد. تمام ترمینالهای یک ماژول، دارای مدار یکسان هستند. بلوک اولیه سیگنالهای ورودی را از سوئیچها، سنسورها و غیره دریافت می نمایند. سیگنالهای AC، در واحد مبدل DC که به یکسوسازها، المانهای مربوط به کاهش ولتاژ و تنظیم کننده ها مجهز است، به سیگنالهای DC تبدیل می شوند. برای سیگنالهای DC به نوعی از مدارات مبدل DC به DC نیاز است. خروجی این مبدل مستقیماً به ورودی CPU متصل نمی شود، چون این امر در صورت یک جهش ولتاژ یا نقص فنی در مدار ممکن است، سبب آسیب دیدن CPU شود. به عنوان مثال اگر واحد مبدل دچار اتصال کوتاه شود، ولتاژ ۱۲۰ ولت متناوب در ورودی مستقیماً به CPU می رسد و چون CPU با ولتاژ ۵ ولت مستقیم کار می کند، احتمال سوختن آن زیاد خواهد بود. بلوک جداسازی (Isolator)، CPU را از چنین صدمه ای حفظ می نماید. این جداسازی معمولاً توسط جداسازهای نوری، همانطور که در شکل نمایش داده شده است، انجام می گردد. قطع و وصل سیگنال، از طریق فعال شدن یک اشعه نورانی (تولید شده توسط یک LED) و نهایتاً فعال سازی توسط یک ترانزیستور نوری به خروجی این طبقه منتقل می گردد.

ماژول خروجی به گونه ای عکس ماژول ورودی عمل می نماید. یک سیگنال DC که از CPU ارسال می گردد، در هر ماژول خروجی به سیگنال الکتریکی با سطح ولتاژ مناسب به صورت AC یا DC که توسط دستگاهها قابل استفاده باشد، تبدیل می گردد. شکل ۳-۴ نمودار بلوکی یک ماژول خروجی را نمایش می دهد.

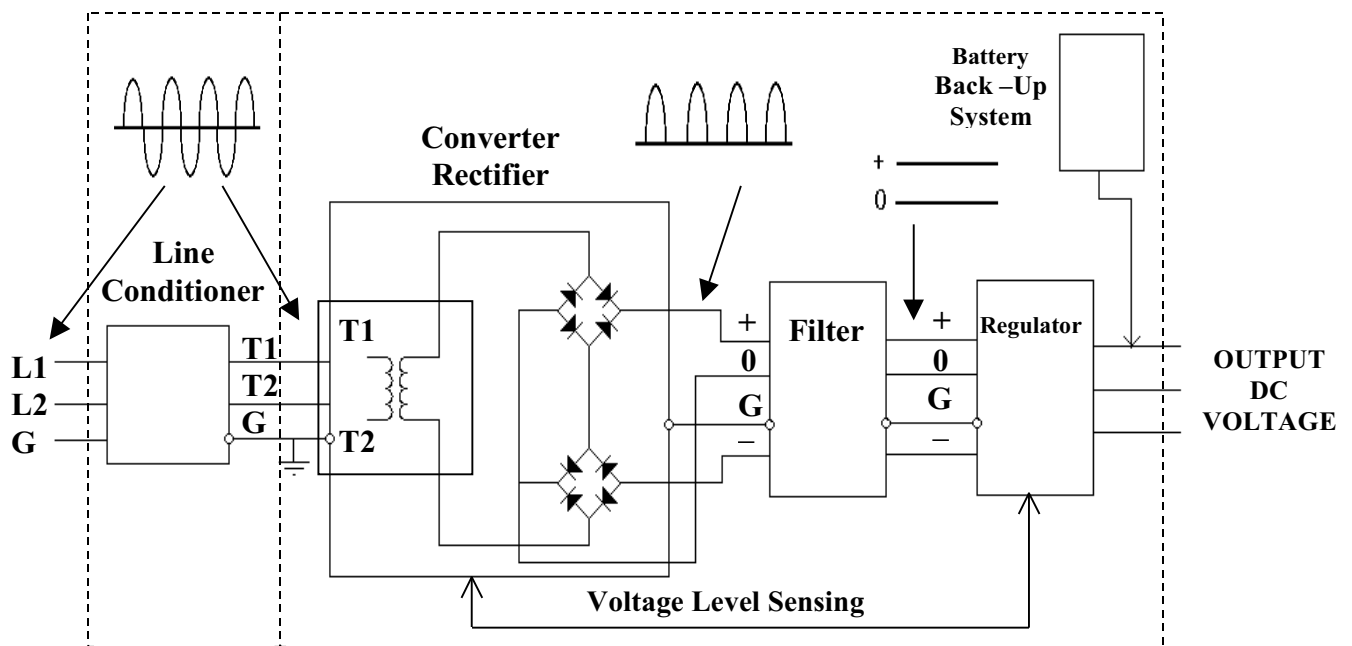


شکل ۳-۴: طرح ماژول خروجی PLC



**منابع تغذیه**

منبع انرژی الکتریکی که معمولاً استفاده می شود منبع جریان متناوب ۲۲۰ ولت با فرکانس ۵۰ الی ۶۰ هرتز می باشد. از آنجا که اغلب PLC ها ، با ولتاژهای +۵ ، -۵ و ۲۴ ولت کار می نمایند ، لذا هر PLC باید مجهز به مدارهایی باشد که بتواند این تبدیل ولتاژها را انجام دهد . این تبدیل با استفاده از یک منبع تغذیه داخلی انجام می شود . شکل ۳-۵ ، ساختار کلی یک منبع تغذیه نمونه را بصورت نمودار بلوکی ، نمایش می دهد . در این شکل نمودارهای ولتاژ - زمان نقاط مختلف مدار دیده می شوند .



شکل ۳-۵ : ساختار بلوکی یک منبع تغذیه PLC

چهار قسمت در نمودار به همراه یک مدار کلیدزنی برای باتری پشتیبان دیده می شود . اولین بلوک سمت چپ ، بلوک بهسازی AC ( Conditioner ) است . این بلوک دو کار اساسی بر روی شکل موج ورودی که معمولاً بصورت سینوسی می باشد انجام می دهد . اولاً در برخی مواقع ممکن است این شکل موج در اثر عمل کلیدزنی یا مسائل فنی ژنراتور اعوجاج پیدا نماید و ثانیاً تجهیزات کارخانه ، ممکن است شوکهای الکتریکی برگشتی خاصی را ایجاد کنند که شکل سینوسی را از حالت اصلی خارج نماید. دومین بلوک نشان داده شده در شکل ، بلوک Converter / Rectifier است . این بلوک شکل موج AC متناوب را به شکل موج DC یکسو شده و ضربان دار تبدیل می نماید . کاهش ولتاژ به میزان مورد نیاز نیز با استفاده از یک ترانسفورماتور درونی انجام می شود . سپس یکسوسازهای پل ، ولتاژهای DC ضربان داری را تولید می کنند . یک رایانه جهت عملکرد صحیح ، احتیاج به یک ولتاژ DC ثابت دارد . بنابراین وسیله ای برای صاف کردن ولتاژ DC ضربان دار مورد نیاز است . سومین قسمت معرفی شده در نمودار ، یک فیلتر است که عمل صاف کردن تغییرات را انجام می دهد . قسمت فیلتر دارای مدارهای شامل خازن ، مقاومت یا سلف است . البته عمل صاف کردن توسط مدارات الکترونیکی نیز میسر است .



چهارمین بلوک ، تنظیم کننده ( Regulator ) است که همواره در مدار تغذیه وجود دارد . یک تنظیم کننده ، بدون توجه به میزان مصرف CPU ، ولتاژ را در محدوده ولتاژ مورد نظر ثابت می نماید . سوئیچ باتری پشتیبان در قسمت بالای سمت راست نمودار مشاهده می شود . سوئیچ تعبیه شده می تواند منبع انرژی الکتریکی را، از منبع تغذیه به باتری تغییر دهد . این سوئیچ به صورتی طراحی شده است که به محض قطع شدن منبع تغذیه ، باتری ذخیره را در مدار قرار می دهد . پیوستگی ولتاژ تغذیه سبب جلوگیری از پاک شدن برنامه کاربر می گردد . منابع تغذیه جدید مورد استفاده در سیستم های PLC از نوع منابع تغذیه سوئیچینگ بوده که نسبت به نمونه منابع دارای ترانسفورماتور دارای حجم و وزن کمتری می باشند .

**توجه :** جهت کسب اطلاعات بیشتر در خصوص منابع تغذیه سوئیچینگ به ضمیمه ۴ مراجعه نمایید .



## فصل چهارم

## آشنایی با انواع مختلف PLC

## کنترل صنعتی و پیشرفت PLC

مبداء پیدایش اتوماسیون صنعتی با ماشینها و فرآیندهای کنترل آن به دهه ۱۹۲۰، با ظهور کنترل کننده های بادی ( Pneumatic Controllers ) اولیه برمی گردد. این وسایل که از هوای فشرده استفاده می کردند، ابزاری قابل انعطاف، اقتصادی و بی خطر بودند. آنها هیچ خطر جرقه زنی در فشارهای زیاد ایجاد نمی کردند و می توانستند حتی در محیط های مرطوب بدون ایجاد خطر شوک الکتریکی کار کنند. اتصال یک وسیله به وسیله دیگر به آسانی و توسط لوله کشی و یا شیلنگ قابل انعطاف صورت می گرفت. عمل کنترل با دستکاری ساده شیرها که توسط رله ها و سوئیچها کنترل می شدند، انجام می شد. این سیستم بادی، انعطاف پذیری زیادی برای کنترل حرکت و سرعت بوجود آورد. به سبب تعداد کم قسمت های متحرک، قابلیت اعتماد و هزینه پایین نگهداری در این سیستم ها بدیهی است. حتی امروزه کنترل کننده های بادی به طور گسترده در ارتباط با PLC ها و رایانه های صنعتی، در انواع کاربردهای کنترل ماشین ابزار و فرآیند ها، مورد استفاده قرار می گیرند.

گرچه کنترل کننده های بادی مزایای خاص خود را داشتند، و هنوز هم دارند ( مخصوصاً با ظهور کنترل کننده های بادی منطق دیجیتال به عنوان یک ابزار واقعی کنترل ) جایگزینی آنها با کنترل کننده های نیمه هادی گسسته در دهه ۶۰ شروع شد. این کنترل کننده های جدید که در ابتدا از مدارات ترانزیستوری تشکیل شده بودند و بعدها از مدارات مجتمع مجزا با مقیاس پایین و متوسط ساخته شدند، کنترل کننده هایی کوچک، با قابلیت اعتماد بالا و با توان مصرفی پایین هستند.

در اوایل دهه ۱۹۷۰ با ورود ریزپردازنده ها، جهش بزرگی در زمینه اتوماسیون و کنترل صنعتی به وقوع پیوست. اکنون مدلسازی و شبیه سازی با برنامه نویسی یا نرم افزارهای مخصوص انجام می شود، در حالی که قبلاً این کار توسط سیم کشی های پیچیده گیت های گسسته، دیکدرها و انکدرها، شمارنده ها، تایمرها و زمان سنج ها، فلیپ فلاپ ها و مدارات دیجیتال مشابه انجام می گرفت.

با وجود اینکه کنترل کننده های ریزپردازنده ای که اغلب میکرو کنترلر نامیده می شوند، جای خود را در میان کنترل ماشین آلات و فرآیندهای صنعتی پیدا کرده است، ولیکن با وجود PLC که اتوماسیون صنعتی را در قبضه خود در آورده است، میکروکنترلرها در حد یک ابزار کنترل کننده باقی مانده اند. امروزه PLC با اختصاص ورودی/ خروجی ( آنالوگ و دیجیتال )، سخت افزار مقاوم، پردازش پوینده ( Scanner ) و روش های مختلف برنامه نویسی، یک روش بهینه در کنترل صنعتی بوجود آورده است.



### انواع سیستم های PLC

در صنعت PLC بیش از یکصد کارخانه با تنوع بیش از هزار مدل از انواع مختلف PLC فعالیت می نمایند. این نمونه های مختلف دارای سطوح مختلفی از کارایی می باشند. PLC ها را می توان از نظر اندازه حافظه یا تعداد ورودی / خروجی دسته بندی نمود. جدول ارائه شده در شکل ۴-۱ نمونه ای از این تقسیم بندی را نمایش می دهد.

اندازه حافظه	تعداد خطوط I/O	اندازه PLC
1K	۴۰/۴۰	کوچک
4K	۱۲۸/۱۲۸	متوسط
بیش از 4K	بیشتر از ۱۲۸ / بیشتر از ۱۲۸	بزرگ

شکل ۴-۱: دسته بندی PLC ها

البته برای ارزیابی قابلیت یک PLC باید ویژگی های دیگری نظیر پردازنده، زمان اجرای یک سیکل، سادگی زبان برنامه نویسی، قابلیت توسعه و غیره را در نظر گرفت. در یک تقسیم بندی PLC ها در دو غالب PLC های با کاربرد محلی و PLC های با کاربرد وسیع تقسیم می گردند.

### PLC ها با کاربرد محلی:

این نوع PLC ها برای کنترل سیستم هایی با حجم کوچک با تعداد ورودی و خروجی های محدود استفاده می گردند. به علت قابلیت محدود تر، این نوع PLC ها برای کنترل همزمان تعداد کمتری از پروسه ها و یا کنترل دستگاههای مجزای صنعتی مورد استفاده قرار می گیرند. اغلب شرکت های سازنده، این نوع PLC ها را به همراه سایر PLC ها به بازار ارائه نموده اند ولی برخی از شرکت های سازنده آنها با نام میکرو PLC به بازار ارائه می نمایند. از جمله این نوع PLC ها می توان به نمونه های زیر اشاره کرد:

۱- مینی PLC ساخت کارخانه زیمنس آلمان با نام LOGO

۲- مینی PLC ساخت کارخانه تله مکانیک فرانسه با نام Zelio

۳- PLC مولر آلمان

۴- PLC ، LG کره

۵- ...

شکل های ۴-۲ الی ۴-۵ تعدادی از نمونه های مختلف این نوع PLC ها را نمایش می دهند.

### PLC های بزرگ با کاربرد گسترده:

این نوع PLC ها برای کنترل سایت کارخانجات بزرگ، از جمله کارخانجات سیمان، پتروشیمی و... استفاده می گردند. معمولاً در این نوع صنایع، PLC ها یا پورت های ورودی، خروجی در قسمت های مختلف سایت کارخانه وجود داشته و کنترلی محلی بر قسمت های تحت پوشش خود انجام می دهند.



سپس اطلاعات مورد نیاز با استفاده از روشهای مختلف انتقال DATA به اتاق کنترل مرکزی منتقل شده که در آن محل با استفاده از روش های مختلف مونیتورینگ صنعتی ، اطلاعات را به شکل گرافیکی تبدیل کرده و بر روی صفحه مونیتور نمایش می دهند . در این حال اپراتور تنها با دانستن روش کار با کامپیوتر و بدون نیاز به اطلاعات تخصصی می تواند سیستم را کنترل نماید .

از جمله این PLC ها می توان به نمونه های زیر اشاره کرد :

۱- خانواده PLC های S5 و S7 زیمنس آلمان

۲- خانواده PLC های OMRON ژاپن

۳- خانواده PLC تله مکانیک فرانسه

۴- خانواده PLC میتسویشی ژاپن

۵- خانواده PLC ، LG کره

۶- خانواده PLC آلن برادلی آمریکا

۷- .....

شکل های ۴-۶ الی ۴-۸ برخی از نمونه های مهم این PLC ها را نمایش می دهند .



شکل ۴-۲: میکرو PLC ساخت شرکت زیمنس ( LOGO ) - ماژول اصلی به همراه ماژول های اضافی





شکل ۳-۴: PLC نوع GM7 از شرکت LG به همراه ماژول های اضافی



شکل ۴-۴: نمونه های مختلف PLC های خانواده CPM1A از شرکت Omron





شکل ۴-۵: نمونه های مختلف میکرو PLC های شرکت Moller



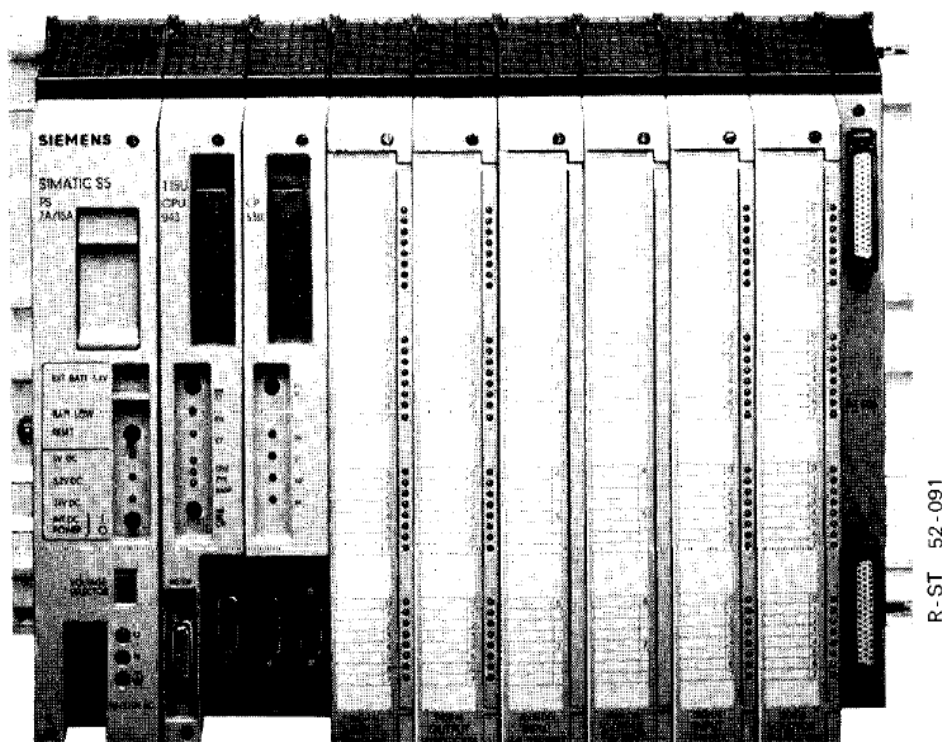
شکل ۴-۶: نمونه ای از PLC خانواده CS1 از شرکت Omron







شکل ۴-۷: PLC نوع GM4 از شرکت LG



شکل ۴-۸: PLC خانواده S5 (CPU 115U) از شرکت زیمنس



## فصل پنجم

# آشنایی با LOGO (سخت افزار)

مینی PLC ، LOGO یک ماژول همه منظوره دیجیتال از تولیدات شرکت زیمنس آلمان می باشد . همانند هر سیستم دیجیتال مبتنی بر پردازنده این کنترل کننده نیز از واحدهای کوچکتری که در ارتباط تنگاتنگ با یکدیگر می باشند تشکیل شده است . مهمترین واحدهای تشکیل دهنده یک مینی PLC ، LOGO عبارتند از :

- ۱- واحد پردازش مرکزی ( CPU )
- ۲- حافظه برنامه
- ۳- منبع تغذیه
- ۴- واحد های ورودی و خروجی
- ۵- عملگر های منطقی پایه جهت انجام عملیات منطقی بر روی داده های ورودی
- ۶- زمان سنج ها ، شمارنده ها و ...
- ۷- بیت های حافظه
- ۸- واحد نمایش و صفحه کلید
- ۹- رابط سخت افزاری جهت برقراری ارتباط با کامپیوتر
- ۱۰- کارت های حافظه

### واحد پردازش مرکزی

در سیستم های مبتنی بر پردازشگر ، CPU به مغز انسان تشبیه می گردد . چون در این واحد است که دستورات رمزگشایی شده ، تصمیم گیری های لازم جهت انجام دستورات گرفته شده و زمانبندی سیستم تولید می شود . این واحد مجموعه ای از مدارات منطقی است که بطور پیوسته عمل مکش دستورات عمل ها و اجرای آنها را تکرار می نماید . این واحد توانایی فهم و اجرای دستورات عمل هایی را دارد که بر اساس یک سری کدهای خاص نوشته شده اند و هر یک بیانگر یک عملکرد ساده می باشند .

### واحد حافظه

در یک سیستم مبتنی بر پردازشگر ، حافظه محل ذخیره سازی اطلاعات است . از این واحد است که CPU دستوراتی را که او را در انجام کار هدایت می کند بر می دارد . انواع مختلف حافظه وجود دارد که نوع حافظه بکار رفته در یک سیستم خاص به کاربرد آن سیستم بستگی دارد . در PLC ها معمولاً دو نوع حافظه RAM و EEPROM (و یا EPROM) وجود دارد .

### واحدهای ورودی و خروجی

واحدهای ورودی و خروجی از ملموس ترین واحدهای یک سیستم مبتنی بر پردازشگر جهت کاربر است ، زیرا کاربر از طریق این واحدهاست که می تواند با سیستم کنترل ارتباط برقرار کرده و دستورات لازم ورودی را به آن اعمال کرده و خروجی های مورد نیاز را جهت قسمت های مختلف سیستم تحت کنترل از آن دریافت می نماید .



با توجه به وجود انواع مختلف LOGO، تعداد و نوع کارکرد ورودی و خروجی های آنها نیز متفاوت می باشد. در برخی مدل ها خروجی از طریق رله در اختیار کاربر قرار گرفته و در انواع دیگر خروجی از طریق مدارات سوئیچینگ ترانزیستوری در اختیار است. همچنین در مدل های مختلف ورودی ها، قابلیت کار با سطوح ولتاژ متفاوتی را دارا می باشند. لازم بذکر است که انواع مختلف LOGO در ادامه این فصل مورد بررسی قرار خواهند گرفت.

### صفحه کلید و واحد نمایش

در برخی از نمونه های LOGO صفحه کلید و واحد نمایش نیز بر روی سخت افزار مورد نظر تعبیه گردیده است که با استفاده از این واحدها می توان بدون در اختیار داشتن کامپیوتر و نرم افزارهای مرتبط برنامه های موردنظر را بر روی LOGO نوشته و سیستم را کنترل نمود.

### کابل Interface

جهت برقراری ارتباط مابین LOGO و کامپیوتر و بمنظور انتقال اطلاعات، شرکت سازنده کابل های واسطی را به همراه سخت افزار LOGO ارائه کرده است، که به کابل Interface معروف می باشد.

### کارت های حافظه

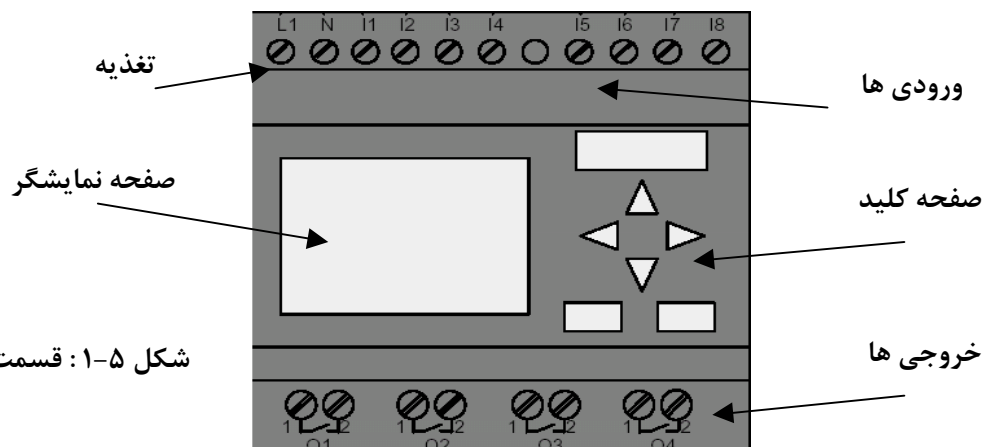
جهت ذخیره اطلاعات و یا انتقال اطلاعات از قسمتی به قسمت دیگر شرکت سازنده انواع مختلفی از کارت های حافظه را ارائه نموده است که با نصب آن بر روی LOGO و بکار بردن دستورات خاص این امر می توان انتقال اطلاعات بین LOGO و کارت های حافظه را انجام داد. این دستورات در فصول بعد معرفی خواهند گشت.

### مدل های مختلف LOGO:

با توجه به توضیحات قبل و زمینه های کاربردی متفاوت LOGO، شرکت سازنده انواع مختلفی از مینی PLC را روانه بازار نموده است. این مدل های مختلف با توجه به نوع سیگنال دریافتی ورودی، تعداد بیت های ورودی و خروجی و ... دسته بندی می گردند.

از نقطه نظر ولتاژ ورودی، انواع مختلف LOGO قابلیت کار با ولتاژهای ۱۲ و ۲۴ ولت مستقیم و همچنین ۲۴ و ۲۳۰ ولت متناوب را دارا می باشند. تیپ های مختلف LOGO و ابعاد فیزیکی آنها عبارتند از:

- ۱- نمونه استاندارد با ۶ ورودی و ۴ خروجی با ابعاد ۷۲×۹۰×۵۰ میلی متر
- ۲- نمونه بدون صفحه نمایشگر با ۶ ورودی و ۴ خروجی با ابعاد ۷۲×۹۰×۵۰ میلی متر
- ۳- نمونه با ۸ ورودی و ۴ خروجی با ابعاد ۷۲×۹۰×۵۰ میلی متر
- ۴- نمونه بزرگتر با ۱۲ ورودی و ۸ خروجی علاوه باس ارتباطی ASI با ابعاد ۱۲۶×۹۰×۵۰ میلی متر
- ۵- قسمت های مختلف یک LOGO را نمایش می دهد.



شکل ۵-۱: قسمت های مختلف LOGO



مدل های مختلف LOGO که در تیپ های چهارگانه فوق الذکر موجود می باشند ، با علامات اختصاری خاصی نامگذاری می گردند که هر یک از این علائم بیانگر مفهوم خاصی است .

۱۲: دارای قابلیت کار با ولتاژ ۱۲ ولت مستقیم

۲۴: دارای قابلیت کار با ولتاژ ۲۴ ولت مستقیم

۲۳۰: دارای قابلیت کار با ولتاژهای ۱۱۵/۲۳۰ ولت متناوب

R: خروجی ها از طریق رله در اختیار کاربر قرار می گیرد . ( در صورت عدم استفاده از این علامت ، خروجی LOGO از نوع ترانزیستوری است . )



C: دارای زمان سنج هفتگی ( نحوه عملکرد این زمان سنج در فصول بعد بررسی خواهد شد . )

O: این حرف علامت مدل های بدون صفحه نمایشگر می باشد .

L: در مدل هایی که در نامگذاری آنها حرف L وجود دارد تعداد ورودی و خروجی ها دو برابر حالت استاندارد می باشد .

B11: دارای قابلیت اضافه نمودن LOGO بصورت پیرو (Slave) به LOGO اصلی

جدول ارائه شده در شکل ۵-۲ نمونه هایی از LOGO های موجود را نمایش می دهد .

نشانه	نام	منبع تغذیه	ورودی ها	خروجی ها	مشخصات
	Logo!12/24RC	12/24V DC	8* Digital	4 Relays 230V×10A	
	Logo! 24	24V DC	8* Digital	4Transistor 24V×0.3A	بدون ساعت
	Logo!24RC	24V AC	8 Digital	4 Relays 230V×10A	
	Logo!230RC#	115/240 AC/DC	8 Digital	4 Relays 230V×10A	
	Logo!12/24RCO	12/24 V DC	8* Digital	4 Relays 230V×10A	بدون نمایشگر و صفحه کلید
	Logo! 24RCO	24 V AC	8 Digital	4 Relays 230V×10A	بدون نمایشگر و صفحه کلید
	Logo!230RCO	115/240 AC/DC	8 Digital	4 Relays 230V×10A	بدون نمایشگر و صفحه کلید

\* : در این نمونه ها می توان از دو ورودی آنالوگ (0-10V) و دو ورودی سریع استفاده کرد .

# : ورودی ها به دو گروه ۴ تایی تقسیم می گردند که ورودی های هر گروه را باید به فاز

یکسان متصل نمود ، ولی می توان در گروه های مختلف از فازهای متفاوتی استفاده کرد .

شکل ۵-۲: نمونه های مختلف LOGO





آموزشگاه برق و الکترونیک صنعتی اندیشه سازان صنعت مجری برگزاری دوره های آموزشی از مقدماتی تا پیشرفته در زمینه های برق ، الکترونیک ، اتوماسیون صنعتی ، ابزار دقیق و کامپیوتر- با مجوز رسمی از سازمان آموزش فنی و حرفه ای کشور و

توجه: نمونه هایی که خروجی آنها بصورت رله ای می باشند ، قابلیت ارائه خروجی با سطوح ولتاژ موردنظر کاربر را دارا می باشند . در این حالت می توان با اتصال پایه مشترک رله ها به ولتاژ موردنظر از کنتاکت های باز و بسته آن با سطح ولتاژ مورد نظر در مدار استفاده کرد .

### ماژول های ورودی / خروجی

از آنجا که ممکن است در طراحی یک مدار به تعداد ورودی - خروجی بیشتری احتیاج باشد ، بدین منظور شرکت سازنده ماژول های ورودی - خروجی جداگانه ای را تولید نموده است ، که با اتصال آنها به ماژول اصلی می توان از ورودی - خروجی بیشتری نیز در مدار استفاده نمود .  
شکل ۳-۵ ، جدولی از نمونه های مختلف این ماژول ها را نمایش می دهد .

نشانه	نام	منبع تغذیه	ورودی ها	خروجی ها
	Logo!DM 8 12/24 R	12/24V DC	4 Digital	4 Relays
	Logo! DM 8 24	24V DC	4 Digital	4 Transistor
	Logo!DM 8 230R	115/240 AC/DC	4 Digital	4 Relays
	Logo!AM 2	12/24 V DC	2 Analog 0-10V or 0-20mA	ندارد

شکل ۳-۵ : نمونه های مختلف ماژول های ورودی/خروجی

### سیم کشی LOGO

پس از نصب LOGO بر روی ریل ، می بایست اتصالات سیمی لازم را برقرار نماییم . با استفاده از یک پیچ گوشتی با اندازه دهنه 3mm می توان LOGO را سیم کشی نمود .

### اتصال منبع تغذیه

مدل های Logo!...230... ، جهت کار در ولتاژهای خط با مقادیر ۱۱۵/۲۳۰ ولت متناسب می باشند . همچنین نمونه های 24 و 12 می توانند در ولتاژهای ۲۴ ولت مستقیم و متناوب و یا ۱۲ ولت مستقیم از طریق منبع تغذیه کار نمایند .

برای اتصال LOGO به سیستم می بایست اتصالات لازم تغذیه را همانند شکل ۴-۵ انجام داد . بدین منظور می توان فاز یا ولتاژ مثبت را از طریق فیوز به ورودی مورد نظر متصل نمود . فیوز مورد نیاز که جهت محافظت LOGO در نمونه های مختلف 12/24 مورد نیاز می باشد ، عبارتند از:

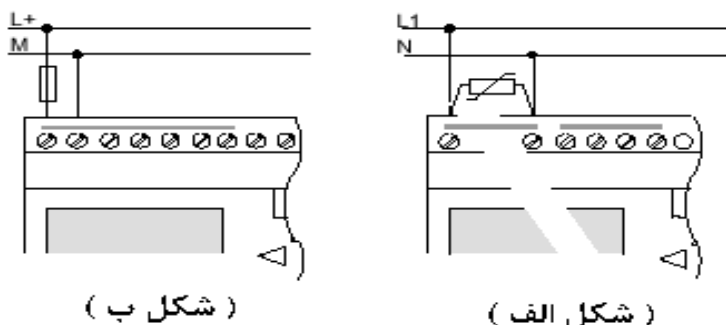
مدل های 12/24 Rc... : ۰/۸ آمپر

مدل های 24 : ۲ آمپر

مدل های 24 L : ۳ آمپر

در اتصال نمونه های با تغذیه AC ، توصیه می شود از یک ورستور با ولتاژ عمل کننده حدود ۲۰ درصد بزرگتر از ولتاژ نامی ، مابین فاز و نول استفاده گردد .





شکل ۴-۵: (الف) اتصال نمونه های ۲۳۰ به تغذیه (ب) اتصال نمونه های ۲۴/۱۲ به تغذیه

### اتصال ورودی ها

ورودی های LOGO که شامل اطلاعات دریافتی از کلیدها، شستی ها و انواع مختلف سنسورها می باشند، باید دارای مقادیر ولتاژ و جریان خاصی بوده تا بتوان آنها را در دو سطح دیجیتال بالا و پایین منظور نمود. لذا جهت شناخت این مقادیر تعریف شده جدول موجود در شکل ۵-۵ ارائه گردیده است. در این جدول مقادیر حداکثر و حداقل ولتاژ و جریانی که باعث می گردند یک ورودی توسط PLC در سطح بالا یا پایین در نظر گرفته شود، جهت نمونه های مختلف LOGO ارائه شده است.

	LOGO!12/24 RC/Rco LOGO! DM8 12/24 R I1...I6		LOGO! 24 LOGO! DM8 24 I1...I6	
		I7,I8		I7,I8
وضعیت مداری صفر جریان ورودی	< 5V DC <1.0 mA	< 5V DC < 0.05mA	< 5V DC <1.0 mA	< 5V DC < 0.05mA
وضعیت مداری یک جریان ورودی	> 8V DC >1.5 mA	> 8V DC >0.1 mA	> 8V DC >1.5 mA	> 8V DC >0.1 mA
	LOGO! 24 RC/Rco (AC)	LOGO! 230 RC/Rco (AC) LOGO!DM8 230R(AC)	LOGO! 230 RC/Rco (DC) LOGO!DM8 230R(DC)	
وضعیت مداری صفر جریان ورودی	< 5V AC <1.0 mA	< 40V AC <0.03 mA	< 30V DC <0.03 mA	
وضعیت مداری یک جریان ورودی	> 12V AC >2.5 mA	> 79V AC >0.08 mA	> 79V DC >0.08 mA	

شکل ۵-۵: مقادیر حد بالا و پایین در ورودی ها



توجه: به هنگام ایجاد یک لبه بالا رونده در مدار، حالت منطقی یک و بهنگام ایجاد لبه پایین رونده، حالت صفر منطقی سرعت و پس از گذشت یک سیکل زمانی در مدار ایجاد می گردد تا بدینوسیله حالت جدید مدار بسرعت توسط LOGO تشخیص داده شود. همچنین قابل ذکر است که یک سیکل زمانی پردازش برنامه به حجم برنامه تحت پردازش توسط LOGO بستگی دارد.

### ورودی های سریع (Fast)

برخی از نمونه های LOGO (230...، 24RC و 24Rco) ورودی هایی برای عملگرهای فرکانسی دارند. سایر ورودی های محدود نمی توانند به عنوان این ورودی های سریع استفاده گردند. ورودی های سریع، آخرین ورودی ها در LOGO ها می باشند. در نمونه های استاندارد ورودیهای I5/I6 و در مدل های ...L... ورودی های I11/I12 به عنوان ورودی سریع استفاده می گردند.

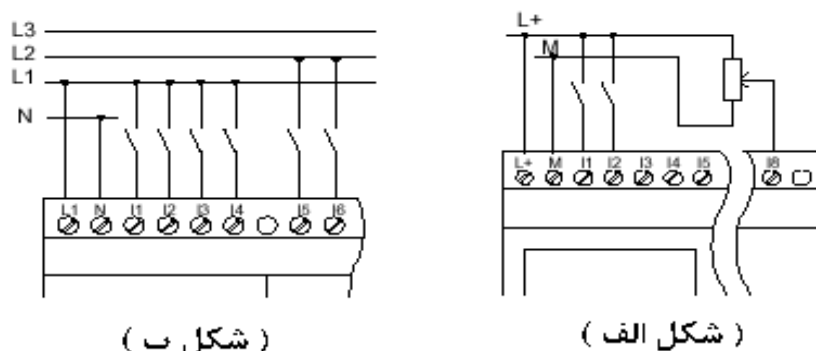
### ورودی های آنالوگ

در نمونه های 24، 12/24 RC و همچنین 12/24 Rco ورودی های I7 و I8 می توانند هم بعنوان ورودی دیجیتال و هم ورودی آنالوگ استفاده گردند. این ورودی ها را می توان با تعریف بصورت I7/I8 بصورت دیجیتال و با تعریف بصورت AI1/AI2 بصورت آنالوگ مورد استفاده قرار داد. این موضوع در آینده بطور کامل مورد بررسی قرار خواهد گرفت.

توجه: جهت ورودی های آنالوگ می بایست از سیم های به هم تابیده (Twisted Wire) و در طول کوتاه استفاده نمود.

شکل ۵-۶ روش اتصال ورودی ها را در نمونه های مختلف LOGO نمایش می دهد.

توجه: از آنجا که پایه های ورودی در مدل های ... 12/24 ایزوله نمی باشند بنابراین این نمونه ها به یک پتانسیل مرجع (زمین) نیاز دارند، لذا ورودی های آنالوگ را می توان ما بین تغذیه و زمین قرار داد. (شکل ۵-۶ - الف)



شکل ۵-۶: (شکل الف) اتصال ورودی در نمونه های 12/24 (شکل ب) اتصال ورودی در نمونه های 230

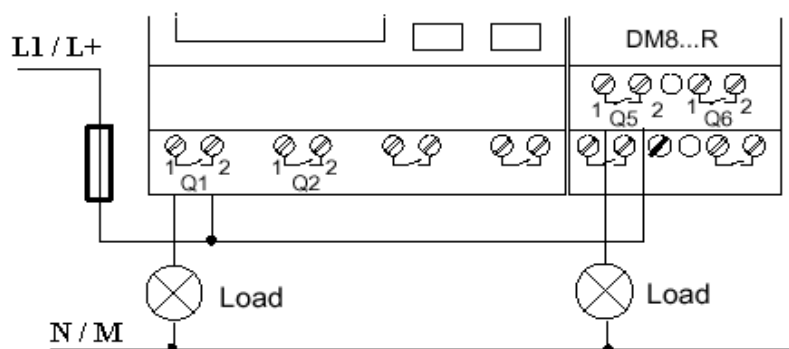
توجه: در نوع LOGO!...L...، ورودی ها به گروههای ۴ تایی دسته بندی شده اند. این گروهها به طور مشابه نوع استاندارد برای ورودی های منحصر بفرده استفاده می گردند. همچنین در این نوع می توان فاز های متفاوتی را به ورودی ها اعمال نمود، البته به این شرط که در هر گروه، ورودی ها به فازهای مشابه متصل گردند.



### اتصال خروجی ها

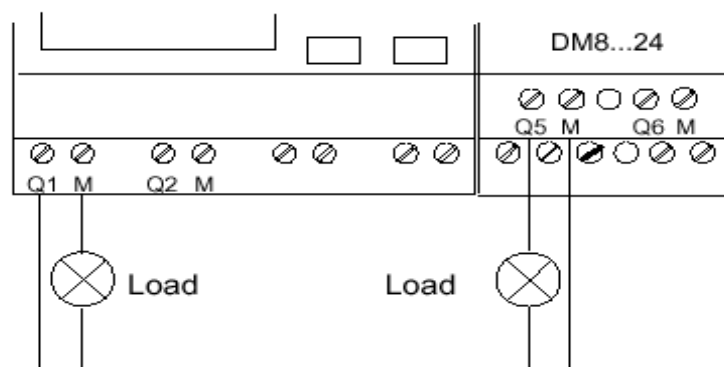
در برخی از نمونه های LOGO که در شماره مدل آنها حرف R وجود دارد ، بیت های خروجی به بوبین رله متصل گشته اند . در این نمونه ها کنتاکت های رله از منبع تغذیه و ورودی های مدار مجزا می باشند لذا می توان خروجی ها را با سطوح ولتاژ مورد نظر تغذیه نمود . این خروجی ها می توانند به بارهای مختلفی از قبیل لامپ ها ، موتورها ، کنتاکتورها و ... متصل گردند . باری که به خروجی مدل های LOGO!...R... متصل می گردند می بایست دارای خصوصیات زیر باشند .

- ۱- ماکزیمم جریان کلید زنی به نوع بار مصرفی و تعداد سیکل کلید زنی بستگی دارد .
  - ۲- در نمونه های استاندارد سطح ولتاژ بالا در خروجی ، در بارهای اهمی تا حداکثر ۱۰ آمپر و در بارهای القایی ماکزیمم تا ۳ آمپر جریان دهی دارد . ( ۲ آمپر در نمونه های ... 12/24 ولت مستقیم و متناوب )
- شکل ۵-۷ نحوه اتصال بار به یک LOGO با خروجی رله ای و همچنین ماژول ورودی / خروجی DM8 ...R را نمایش می دهد .



شکل ۵-۷ : نحوه اتصال بار به نمونه های دارای رله در خروجی

نمونه های LOGO با خروجی ترانزیستوری را می توان از عدم وجود حرف R در شماره مدل آنها تشخیص داد . در این اقسام ، خروجی ها در برابر اتصال کوتاه و اضافه بار مقاوم می باشند . حداکثر جریان سوئیچ در این نمونه ها ۰/۳ آمپر در هر خروجی است . شکل ۵-۸ نحوه اتصال بار به خروجی این نمونه از LOGO را نمایش می دهد .



شکل ۵-۸ : نحوه اتصال بار به نمونه های دارای ترانزیستور در خروجی ( بارها ۲۴ ولت و حداکثر ۰/۳ آمپر )

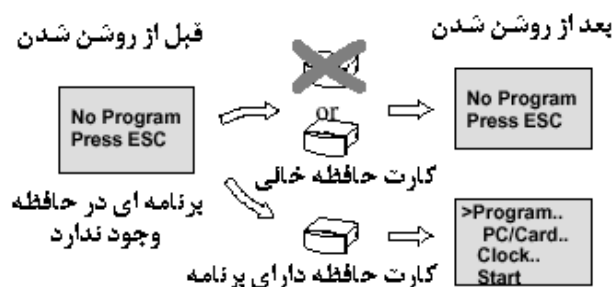




### عملکرد LOGO پس از اتصال تغذیه

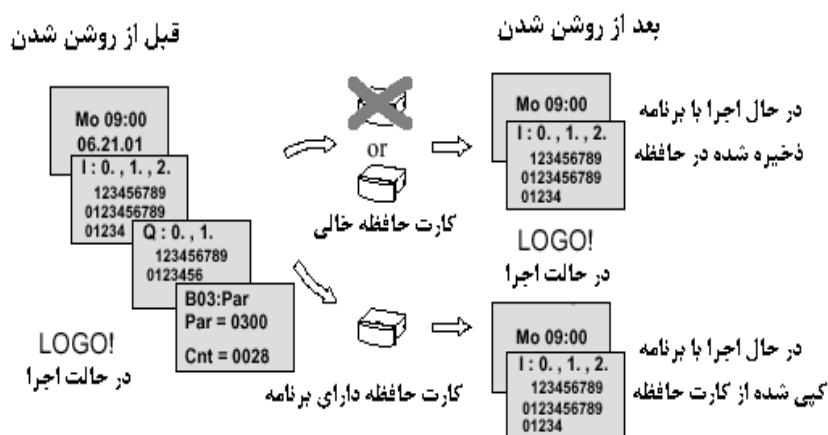
از آنجا که LOGO دارای کلید تغذیه نمی باشد، لذا عملکرد آن پس از اتصال تغذیه به موارد زیر بستگی دارد:

- ۱- آیا برنامه ای در حافظه LOGO وجود دارد یا خیر؟
  - ۲- کارت حافظه حاوی برنامه، بر روی PLC نصب است یا خیر؟
  - ۳- PLC مورد استفاده از نوع دارای نمایشگر می باشد یا خیر؟
  - ۴- PLC به هنگام خاموش شدن در مرحله قبل در چه حالت کاری بوده است؟
- شکل ۵-۹ وضعیت LOGO را قبل و بعد از روشن شدن در زمانی نمایش می دهد که برنامه ای در حافظه LOGO وجود ندارد. در این حال در صورت عدم وجود کارت حافظه دارای برنامه بر روی PLC پیغام "No Program Press ESC" بر روی صفحه نمایشگر ظاهر می شود.



شکل ۵-۹: وضعیت LOGO قبل و بعد از روشن شدن زمانی که در حافظه LOGO برنامه ای وجود ندارد.

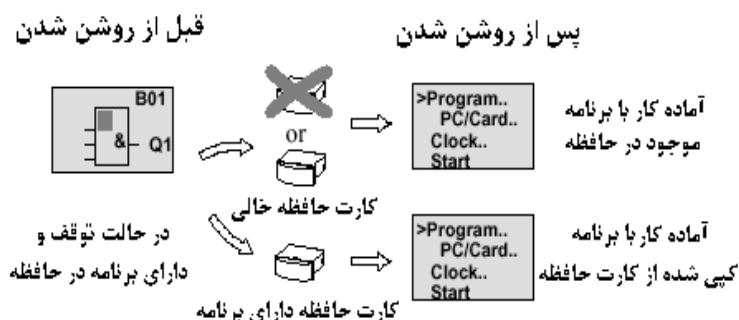
اگر برنامه ای در حافظه LOGO وجود داشته باشد و یا کارت حافظه حاوی برنامه بر روی LOGO نصب شده باشد LOGO پس از روشن شدن آخرین حالت کاری قبل از خاموش شدن مرحله قبل را به یاد می گیرد. اگر کارت حافظه نصب شده بر روی LOGO حاوی برنامه باشد این برنامه پس از روشن شدن بصورت اتوماتیک به درون LOGO کپی می گردد. در صورت وجود برنامه قبلی در حافظه LOGO این برنامه با برنامه موجود بر روی کارت مجدداً بار می گردد. شکل ۵-۱۰ این وضعیت کاری را نمایش می دهد.



شکل ۵-۱۰: وضعیت LOGO در حالی که در مرحله خاموش شدن قبلی، در حالت اجرا بوده است.

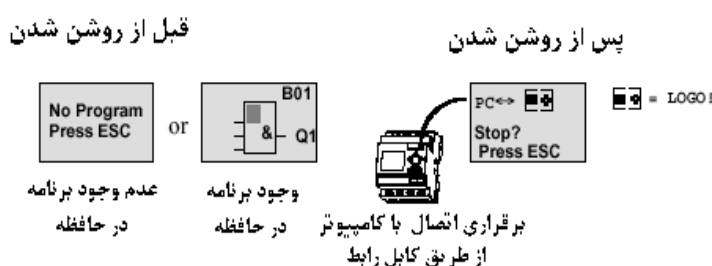


در صورتی که LOGO دارای برنامه بوده و در مرحله خاموش شده باشد. پس از روشن شدن صفحه انتخاب عملکرد LOGO بر روی نمایشگر ظاهر می گردد. با روشن شدن LOGO در صورتیکه کارت حافظه حاوی برنامه بر روی PLC نصب شده باشد، برنامه کارت به داخل LOGO بار می گردد. شکل ۵-۱۱ این وضعیت کاری را نمایش می دهد.



شکل ۵-۱۱: وضعیت LOGO در حالی که در مرحله خاموش شدن قبلی، در حالت توقف بوده است.

در صورت برقراری ارتباط LOGO با کامپیوتر از طریق کابل رابط، پس از روشن شدن LOGO صفحه برقراری ارتباط با PC بر روی صفحه نمایشگر ظاهر خواهد گشت. شکل ۵-۱۲ این وضعیت را نمایش می دهد.



شکل ۵-۱۲: وضعیت LOGO پس از روشن شدن در صورت اتصال کابل رابط

نکته ۱: در نمونه های بدون صفحه نمایشگر (LOGO!...Rco)، PLC پس از روشن شدن بصورت خودکار از حالت Stop به حالت Run سوئیچ می شود.

نکته ۲: در صورت استفاده از عملگرهایی که بصورت دائم خروجی خود را ذخیره می نمایند و یا عملگرهایی که این وضعیت را کاربر برای آنها تعریف می کند، در هنگام خاموش شدن PLC، مقادیر نهایی آنها ذخیره می گردد.

نکته ۳: در صورتیکه هنگام بارگذاری برنامه و ایجاد تغییرات در آن نقصانی در تغذیه بوجود آید، پس از اتصال مجدد تغذیه کل برنامه پاک خواهد شد، لذا بهتر است قبل از اعمال تغییرات، برنامه اصلی را بر روی کارت حافظه و یا کامپیوتر ذخیره نمایید.



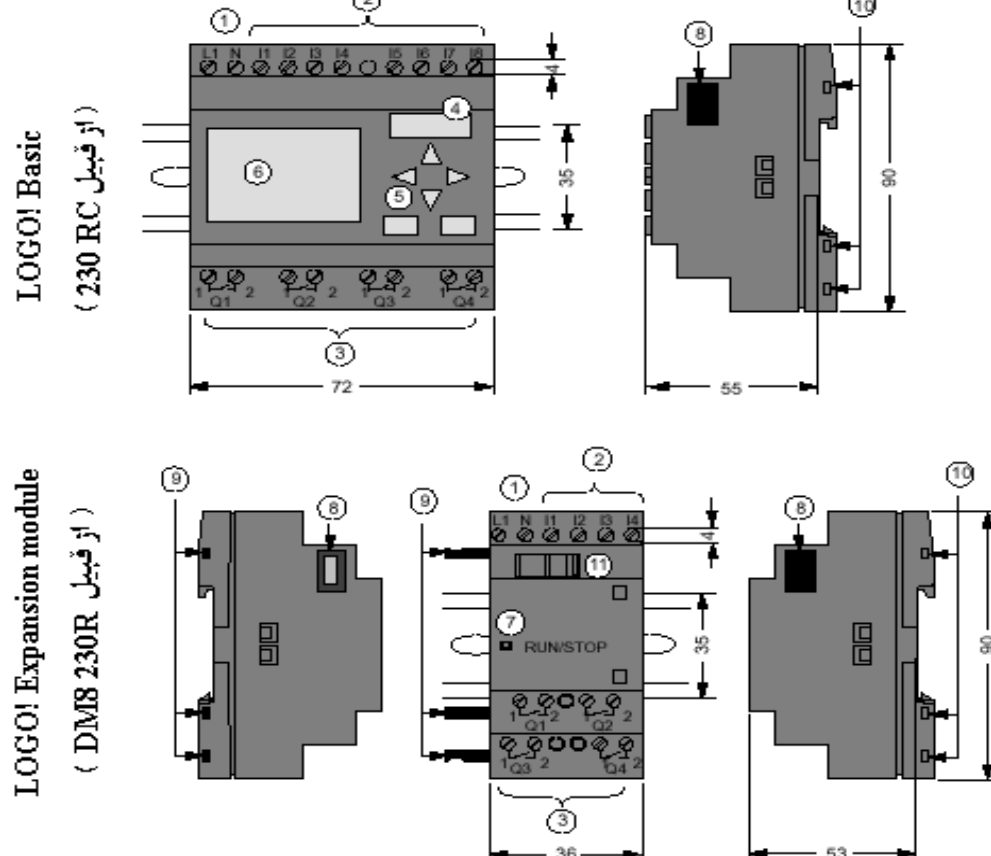
### اتصال LOGO به باس ارتباطی ASi

نمونه های LOGO!...B11 می توانند از طریق باس ارتباطی ASi در یک شبکه قرار گیرند. با استفاده از دو سیم می توان LOGO را در شبکه قرار داده و اقدامات زیر را انجام داد:

- ۱- خواندن و پردازش چهار ورودی اضافی از طریق باس ASi
  - ۲- استفاده از چهار خروجی اضافی در یک پوشش اصلی از باس ارتباطی ASi
- جهت اتصال LOGO از طریق این باس، می توان ابتدا کابل مربوط به باس را به کانکتور هایی که جهت این امر در اختیار است اتصال داده (رعایت پلاریته صحیح ضروری می باشد) و سپس با قراردادن کانکتور ارتباطی در محل مشخص آن بر LOGO، ارتباط لازم را برقرار نمود.
- جهت استفاده از LOGO در باس ارتباطی ASi، می باسیت LOGO در این باس شناخته شود. این امر بطور اتوماتیک پس از اتصال سیم های باس به LOGO صورت می پذیرد. در این حال ماژول اصلی، قسمت پیرو را با در نظر گرفتن آدرسی برای آن شناسایی می نماید.

### سافتار LOGO

LOGO دارای قسمت های مختلفی در ساختار خود می باشند. شکل ۵-۱۳ ابعاد و قسمت های مختلف نمونه استاندارد LOGO و همچنین ماژول های اضافی LOGO، ۲۳۰ را نمایش می دهد.

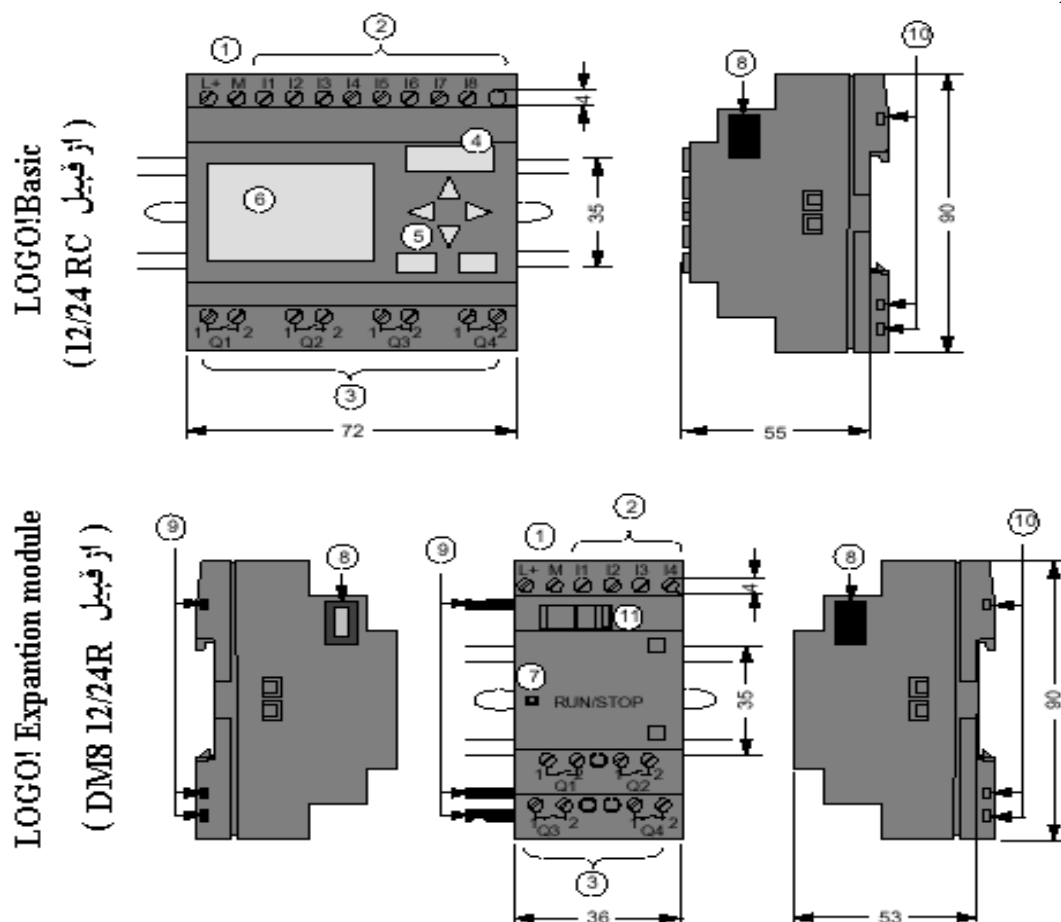


- ۱- منبع تغذیه ۲- ورودی ها ۳- خروجی ها ۴- محل قرار گرفتن کارت های حافظه ۵- صفحه کلید ۶- صفحه نمایشگر
- ۷- نمایشگر حالات کاری روشن یا خاموش ۸- رابط ماژول های اضافی ۹- پین های مکانیکی رمز شده جهت برقراری ارتباط LOGO با ماژول های اضافی ۱۰- سوکت های مکانیکی رمز شده جهت برقراری ارتباط LOGO با ماژول های اضافی ۱۱- پین لغزان جهت برقراری اتصال ماژول های اضافی به LOGO

شکل ۵-۱۳: ساختار LOGO های ۲۳۰



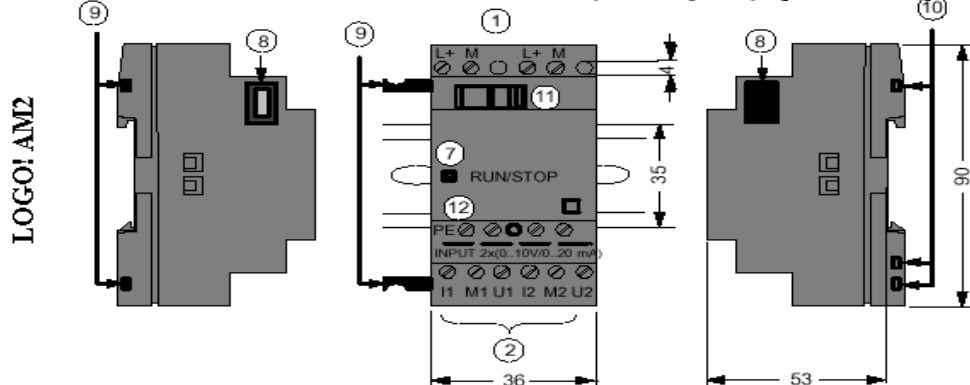
در شکل ۵-۱۴ ابعاد و قسمت های مختلف نمونه استاندارد LOGO و همچنین ماژول های اضافی LOGO ۱۲/۲۴ ارائه شده است.



۱- منبع تغذیه ۲- ورودی ها ۳- خروجی ها ۴- محل قرار گرفتن کارت های حافظه ۵- صفحه کلید ۶- صفحه نمایشگر ۷- نمایشگر حالات کاری روشن یا خاموش ۸- رابط ماژول های اضافی ۹- بین های مکانیکی رمز شده جهت برقراری ارتباط LOGO با ماژول های اضافی ۱۰- سوکت های مکانیکی رمز شده جهت برقراری ارتباط LOGO با ماژول های اضافی ۱۱- بین لغزان جهت برقراری اتصال ماژول های اضافی به LOGO

شکل ۵-۱۴: ساختار LOGO های ۱۲/۲۴

ساختار ماژول آنالوگ AM2 نیز در شکل ۵-۱۵ ارائه شده است.



۱۲- ترمینال زمین جهت اتصال زمین ومحافظ خطوط اندازه گیری آنالوگ

قسمت های مشخص شده با شماره های ۱۱ الی ۱۱ همانند قسمت های ارائه شده در شکل ۳-۱۴ می باشند.

شکل ۵-۱۵: ساختار ماژول ورودی آنالوگ



**توجه :** جهت کسب اطلاعات بیشتر در خصوص ماژول های اضافی LOGO به ضمیمه ۵ مراجعه نمایید .

### نصب LOGO و ماژول های آن بر روی ریل

مراحل مختلف نصب LOGO بر روی ریل در شکل ۳-۱۶ نمایش داده شده است . این مراحل به ترتیب عبارتند از :

۱ و ۲ - LOGO را بر روی ریل قرار داده و با کشیدن قسمت متحرک و حرکت دادن LOGO آنرا بر روی ریل محکم نمایید .

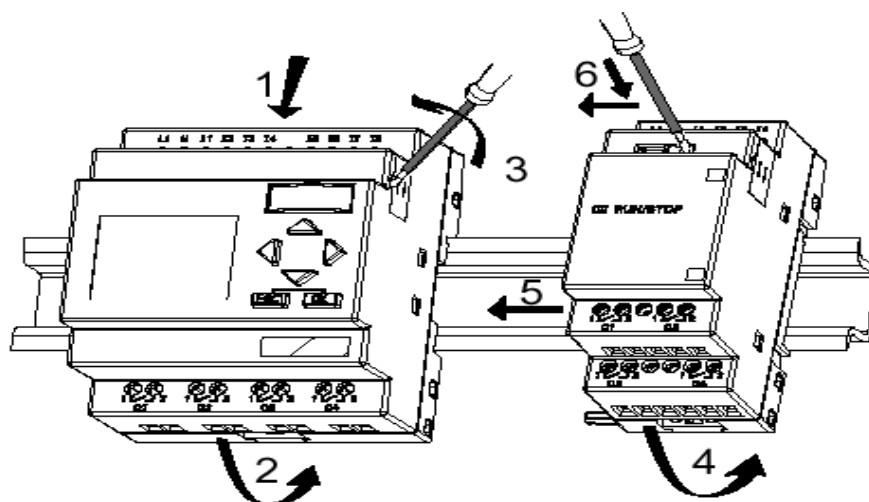
۳ - جهت نصب ماژول دیجیتال پوشش رابط مخصوص ماژول های اضافی را بردارید .

۴ و ۵ - پس از قراردادن ماژول دیجیتال بر روی ریل آنرا بسمت LOGO حرکت دهید .

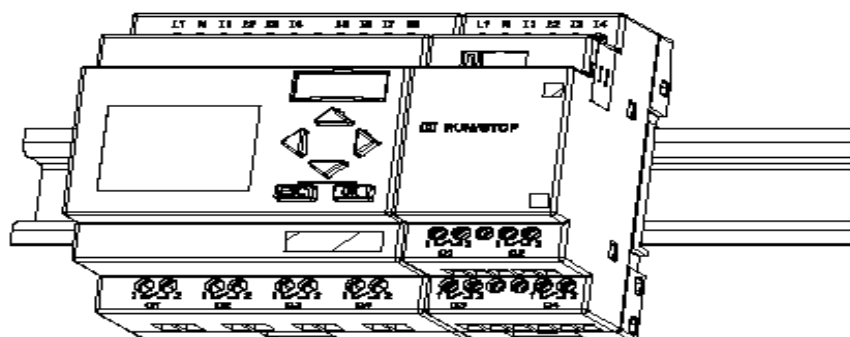
۶ - با استفاده از یک پیچ گوشتی قسمت متحرک لغزان را بسمت چپ و پایین فشار داده و ماژول دیجیتال را به LOGO اتصال دهید .

توجه : جهت اتصال ماژول های دیگر قسمت های ۳ تا ۶ را تکرار نمایید .

شکل ۵-۱۷ وضعیت LOGO و ماژول دیجیتال را پس از اتصال نمایش می دهد .



شکل ۵-۱۶ : نصب LOGO و ماژول دیجیتال بر روی ریل و برقراری ارتباط آنها



شکل ۵-۱۷ : وضعیت LOGO و ماژول دیجیتال پس از اتصال

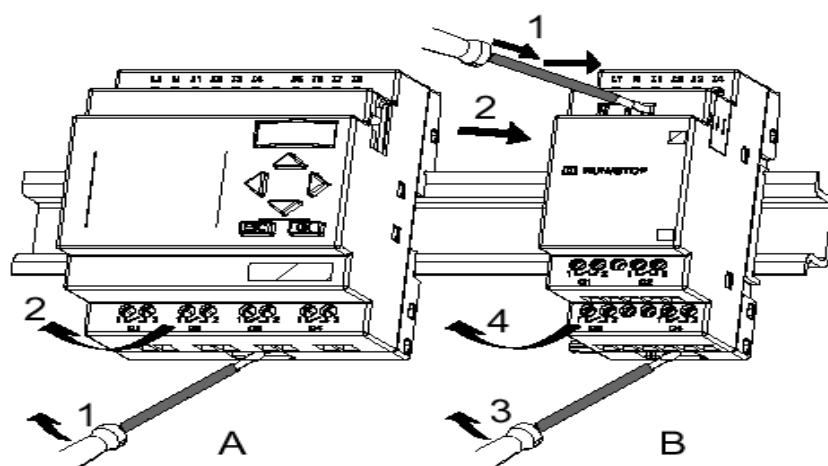
توجه : رابط اتصال ماژول های اضافی در آخرین ماژول می بایست با استفاده از کاور مخصوص پوشانده شود .



### جداسازی LOGO از روی ریل

جهت جداسازی LOGO از روی ریل مطابق شکل ۵-۱۸ بترتیب زیر عمل می نمایم .  
 قسمت A : جداسازی LOGO هنگامی که به تنهایی مورد استفاده قرار گرفته است  
 ۱ و ۲ - با قراردادن یک پیچ گوشتی در محل نمایش داده شده و با فشار اهرم بسمت پایین ، می توان با کمی حرکت LOGO را از روی ریل خارج نمود .  
 قسمت B : جداسازی ماژول اضافی  
 ۱- با استفاده از یک پیچ گوشتی قسمت لغزان بر روی ماژول را به پایین فشار داده و سپس آنرا به سمت راست بکشید .

- ۲- دراین حال ماژول اضافی را بسمت خارج بکشید تا از LOGO جدا گردد .
- ۳- موارد ۱و ۲ در قسمت A را جهت جداسازی ماژول از روی ریل نیز تکرار نمایید .



شکل ۵-۱۸ : جداسازی LOGO و ماژول اضافی از روی ریل



## فصل ششم

# آشنایی با LOGO ( نرم افزار )

نرم افزار LOGO! Soft Comfort نرم افزاری جهت فراهم سازی محیط برنامه نویسی جهت کار با مینی PLC های LOGO می باشد. از آنجا که برنامه نویسی از طریق صفحه کلید تعبیه شده بر روی سخت افزار کاری وقتگیر بوده و همچنین نوشتن برنامه های حجیم با صفحه کلید مزبور بسیار دشوار می نماید لذا شرکت سازنده نرم افزار مورد نظر را که محیطی مناسب جهت نوشتن و ویرایش برنامه در اختیار کاربر قرار داده و همچنین محیطی جهت تست برنامه های مربوط به پروژه های مختلف را فراهم می آورد، ارائه کرده است. لذا ضروری است کاربر پس از شناخت سخت افزار سیستم، شناخت کاملی نیز از نرم افزار مورد نظر بدست آورده تا بتواند با بکارگیری آن پروژه های مورد نظر را پیاده سازی نماید. در این فصل پس از معرفی محیط کاری نرم افزار، چگونگی برنامه نویسی و تست برنامه های مورد نظر با استفاده از آن شرح داده خواهد شد.

### معرفی محیط برنامه :

زمانی که برنامه LOGO! Soft Comfort را اجرا می کنید، صفحه ای شبیه به شکل ۶-۱ را ملاحظه می کنید. این صفحه به چندین ناحیه به شرح زیر تقسیم می گردد.

### محیط کاری :

این قسمت از محیط برنامه، محیط برنامه نویسی بوده که با قرار دادن اجزاء مختلف در این محیط و برقراری ارتباطات لازم می توان نرم افزار مورد نظر را پیاده سازی نمود.

### نوار منوها :

این نوار که در قسمت فوقانی پنجره اصلی قرار دارد شامل گزینه های مختلفی از جمله **Format, Edit, File, View, Tools** و ... می باشد. شکل ۶-۲ این نوار را نمایش می دهد.

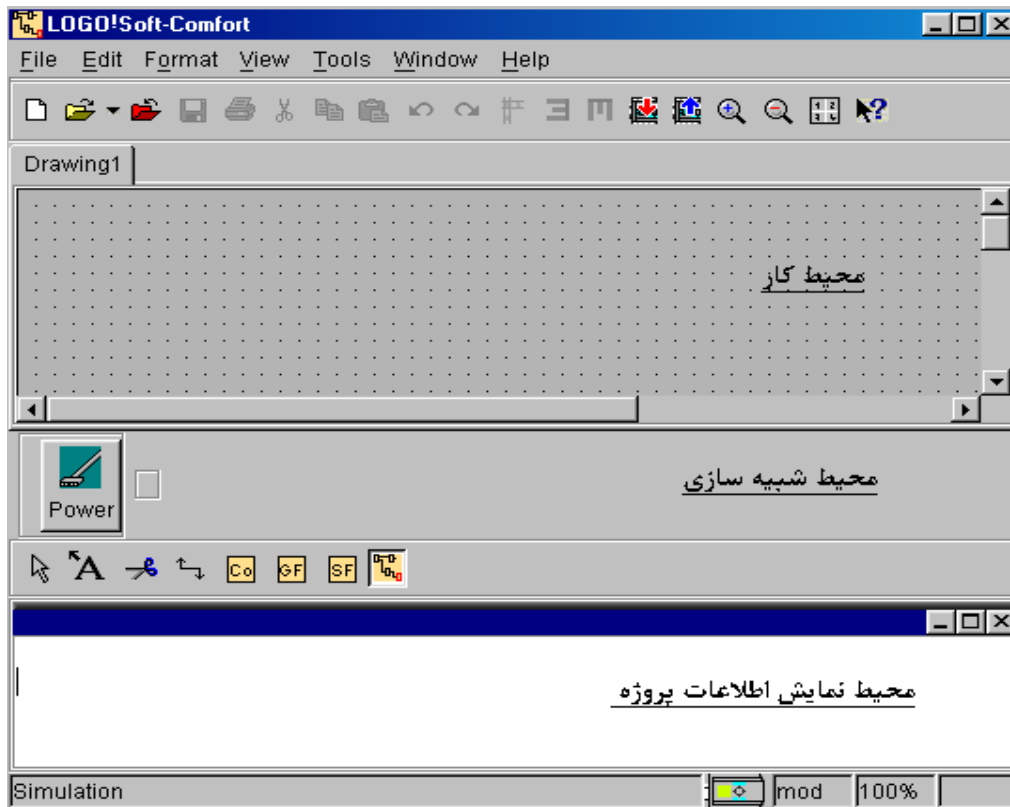
### نوار ابزار استاندارد :

در صفحه اصلی و در پایین نوار منو، نوار ابزار استاندارد مشاهده می شود که شامل دکمه های دستیابی سریع به امکانات مختلف برنامه از قبیل **New, Open, Close** و ... می باشد. همچنین در این نوار ابزار دکمه هایی مربوط به انتقال اطلاعات ما بین PLC و کامپیوتر نیز تعبیه شده است. ( با رفتن بر روی هر یک از این دکمه ها توضیحات مربوط به آن دکمه مشاهده می شود ). شکل ۶-۳ این نوار ابزار را نمایش می دهد.

### نوار ابزارها :

این نوار شامل کلیدهایی جهت انتخاب قطعات مختلف، ابزار برقراری اتصال و ... می باشد. این نوار در شکل ۶-۴ ارائه شده است.





شکل ۶-۱: محیط برنامه



شکل ۶-۲: نوار منوها



شکل ۶-۳: نوار ابزار استاندارد



شکل ۶-۴: نوار ابزارها

### صفحه شبیه سازی:

پس از طراحی و رسم مدار مورد نظر و به هنگام Simulation مدار جهت تست درستی آن در این صفحه اطلاعات مورد نظر نمایش داده می شود. این صفحه در شکل ۶-۱ و تحت نام محیط شبیه سازی نشان داده شده است.

### لیست پروژه ها:

این نوار که در شکل ۶-۵ ارائه شده است لیست صفحات مربوط به پروژه های باز شده را نمایش می دهد.



شکل ۶-۵: لیست پروژه ها

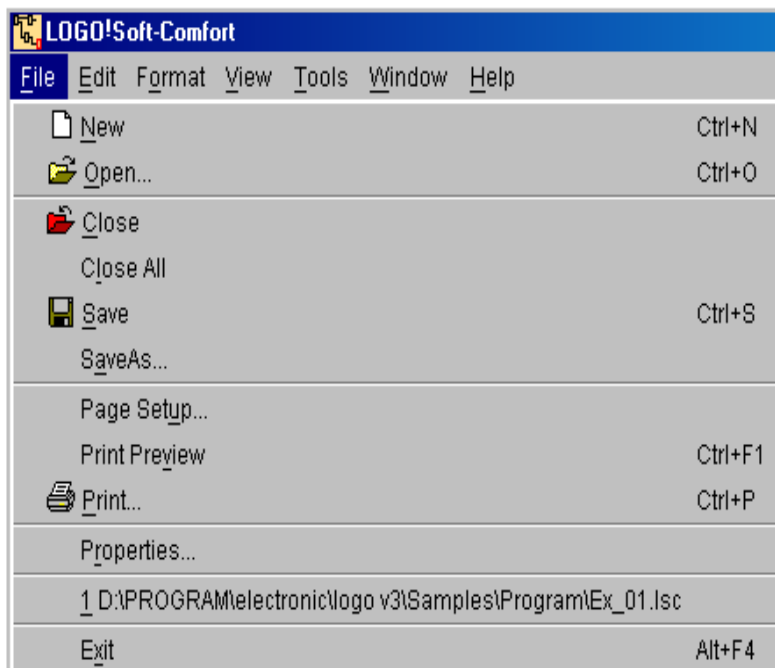




### نوار منوها

#### منوی File :

اولین گزینه در نوار منو ، گزینه File می باشد ، که با کلیک بر روی آن منوی مورد نظر نمایان می گردد . گزینه های مختلف موجود در این منو در شکل ۶-۶ نشان داده شده است .

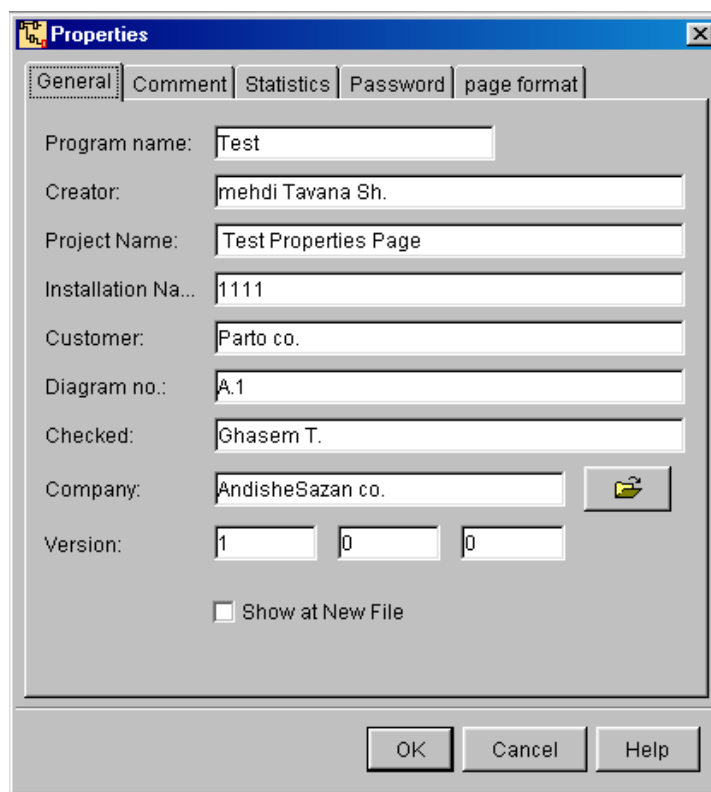


شکل ۶-۶ : منوی فایل

- **New** : جهت ایجاد یک پروژه جدید از این گزینه استفاده می گردد . ( کلید میانبر Ctrl+N )
- **Open** : پروژه طراحی شده قبلی را بارگذاری می کند . ( کلید میانبر Ctrl+O )
- **Close** : پروژه جاری را می بندد .
- **Close All** : تمام پروژه های باز شده را می بندد .
- **Save** : آخرین تغییرات در پروژه جاری را ذخیره می کند . ( کلید میانبر Ctrl+S )
- **Save as** : پروژه جاری را با یک نام دیگر ذخیره می نماید .
- **Page Setup...** : در این صفحه می توان مشخصات مربوط به صفحه کاری پروژه را تعریف نمود . از جمله این مشخصات می توان به قطع کاغذ ، حاشیه های کاغذ و ... اشاره کرد .
- **Print Preview** : با انتخاب این گزینه آنچه قرار است به پرینتر فرستاده شود بر روی صفحه نمایش داده می شود . در واقع با استفاده از گزینه می توان نتیجه حاصل از پرینت را قبل از عملیات چاپ بر روی صفحه مشاهده کرد .
- **Print** : صفحه جاری پروژه را برای چاپ به چاپگر می فرستد .
- **Properties** : با انتخاب این گزینه صفحه ای باز می گردد که می توان در آن مشخصات کلی پروژه مورد نظر را تعریف نمود . از جمله این مشخصات می توان به اطلاعات مربوط به کاربر از قبیل نام و ... ، تاریخ تنظیم پروژه و ... و همچنین قابلیت قراردادن Password جهت فایل ها اشاره کرد .

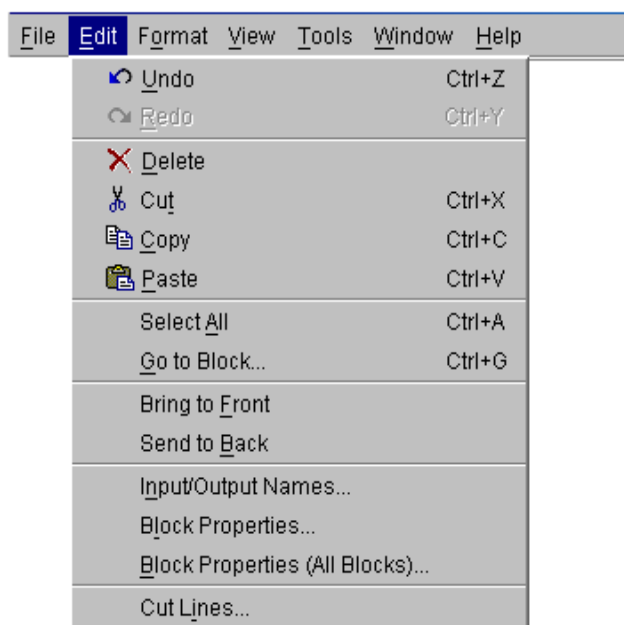


شکل ۶-۷ صفحه مشخصات را نمایش می دهد .



شکل ۶-۷ : صفحه مشخصات

- **Exit**: برای خروج از برنامه از این گزینه استفاده می شود .  
**Edit**:  
 دومین گزینه در نوار منوها بوده که با انتخاب آن صفحه ای مطابق شکل ۶-۸ باز می گردد .



شکل ۶-۸ : منوی EDIT

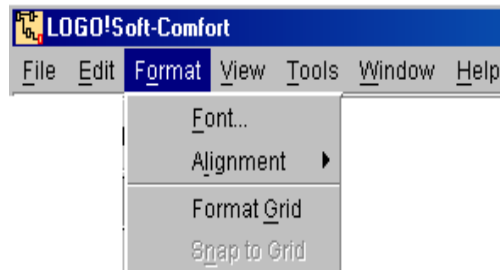
آیتم های مختلف منوی Edit عبارتند از :

- **Undo** : برای پاک کردن آخرین شی که بر روی صفحه کار ایجاد شده است از این گزینه استفاده می شود. ( کلید میانبر : **Ctrl+Z** )
- **Redo** : برای بازگردانیدن شی **Undo** شده استفاده می شود. ( کلید میانبر : **Ctrl+Y** )
- **Delete** : برای پاک کردن اشیاء انتخاب شده استفاده می شود .
- **Cut** : با انتخاب این گزینه اشیاء انتخاب شده را به حافظه موقت ( **Clipboard** ) انتقال می دهیم . ( کلید میانبر : **Ctrl+X** )
- **Copy** : با انتخاب این گزینه یک کپی از اشیاء انتخاب شده به حافظه موقت انتقال می یابد . ( کلید میانبر : **Ctrl+C** )
- **Past** : با انتخاب این گزینه یک کپی از اشیاء موجود در حافظه موقت در محل مورد نظر قرار داده می شود . ( کلید میانبر : **Ctrl+V** )
- **Select All** : تمام اجزای موجود در صفحه جاری را انتخاب کرده و در یک گروه قرار می دهد . ( کلید میانبر : **Ctrl+A** )
- **Go To Block ...** : در صورت نیاز به پیدا کردن یک بلوک در پروژه جاری این گزینه را انتخاب کرده و پس از تایپ نام بلوک و یا انتخاب آن در لیست ارائه شده با فشردن کلید **OK** ، بلوک مورد نظر انتخاب می گردد . ( کلید میانبر : **Ctrl+G** )
- **Bring to front** و **Send to back** : با استفاده از این دو گزینه می توان به مراحل کاری قبل از مرحله جاری و یا در صورت بازگشت به مراحل قبل به مراحل بعد رجوع کرد .
- **Input/ Output Names** : در صورتی که نیاز باشد برای بلوک های ورودی ، خروجی استفاده شده نام خاصی برگزید ، می توان با انتخاب این گزینه و باز شدن صفحه مربوطه نام مورد نظر را تایپ نمود .
- **Block Properties** : هر بلوک استفاده شده در برنامه اعم از بلوک های ورودی / خروجی ، عملگرهای عمومی و یا ویژه دارای مشخصات مربوط به خود می باشند . ( این مشخصات بطور کامل در ادامه بیان خواهند گشت . ) لذا برای ویرایش این اطلاعات می توان با انتخاب بلوک مورد نظر و سپس برگزیدن این گزینه نسبت به این امر اقدام نمود . ( این کار را همچنین می توان با انتخاب شی مورد نظروسپس کلیک راست بر روی آن وسپس انتخاب گزینه **Block Properties** نیزانجام داد . )
- **Block Properties ( All Blocks )** : با انتخاب این گزینه و باز شدن صفحه مورد نظر می توان پس از انتخاب بلوک مورد نظر مشخصات آنرا تغییر داد .
- **Cut Lines** : با انتخاب این گزینه و سپس علامت زدن قسمت های مورد نظر در صفحه باز شده می توان بر روی مسیرهای ارتباطی مابین بلوک ها تغییراتی ایجاد نمود . به عنوان مثال می توان اجازه عبور خط از روی بلوک ها را داده و یا این اجازه را نداد. همچنین می توان حداکثر طول خطوط را نیز تعریف نمود که در این صورت خطوط بزرگتر به صورت خودکار برش می خورند .



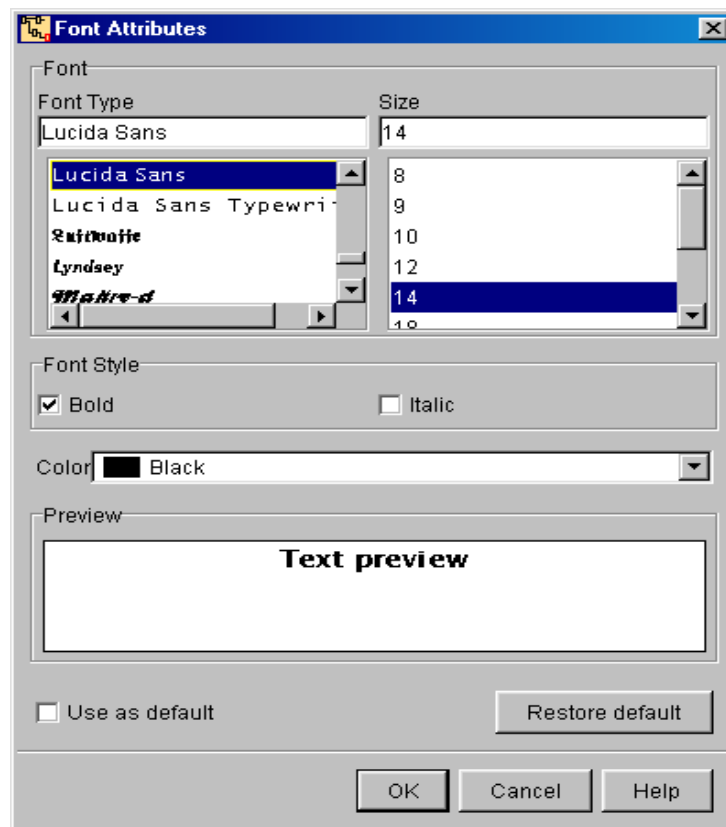
**منوی Format :**

این منو سومین گزینه در نوارمنوها بوده و دارای قسمت های زیر می باشد . شکل ۶-۹ این منو را نمایش می دهد .



شکل ۶-۹ : منوی Format

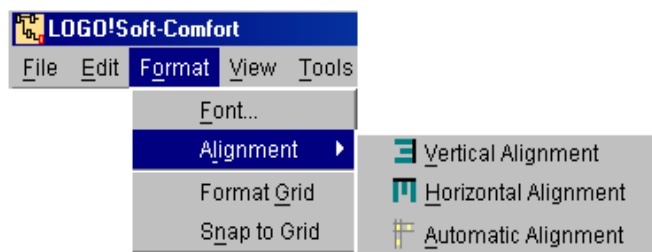
- **Font** : با انتخاب این گزینه صفحه ای همانند صفحه نشان داده شده در شکل ۶-۱۰ باز شده که می توان در آن نوع Font مربوط به متن های نوشته شده در صفحه اصلی و همچنین تنظیمات مختلف آنرا انجام داد . نکته قابل توجه در این صفحه این است که جهت تایید تنظیمات می بایست ابتدا گزینه **Use as default** را انتخاب کرده و سپس کلید **Ok** را بفشاریم .



شکل ۶-۱۰ : صفحه تنظیمات فونت

- **Alignment** : با انتخاب این گزینه می توان اشیاء انتخاب شده را بصورت عمودی و یا افقی صف بندی نمود . گزینه **Automatic Alignment** بصورت خودکار این صف بندی را انجام می دهد . شکل ۶-۱۱ این گزینه را نمایش می دهد .



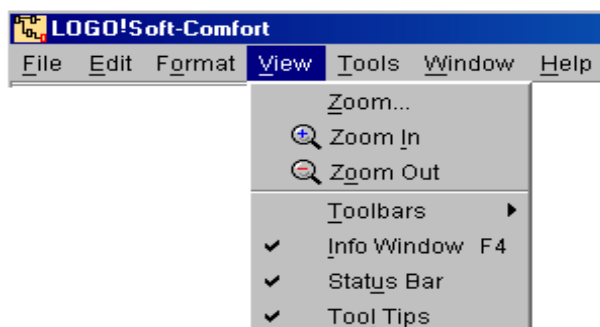


شکل ۶-۱۱: گزینه Alignment

- **Format Grid**: از طریق انتخاب این گزینه می توان در صورت نیاز ابعاد خانه های شطرنجی صفحه اصلی را تغییر داده و یا از نمایش آنها جلوگیری نمود.
- **Snap to Grid**: با کلیک بر روی این گزینه شی انتخاب شده به داخل کادر صفحه های شطرنجی مورد نظر پرش می کند.

منوی View :

چهارمین گزینه در نوار منوها بوده که با انتخاب آن صفحه نمایش داده شده در شکل ۶-۱۲ باز می گردد.



شکل ۶-۱۲: منوی View

گزینه های موجود در این منو عبارتند از :

- **Zoom**: می توان بزرگنمایی تصویر را بصورت درصد در آن انتخاب نمود.
- **Zoom In** و **Zoom Out**: با انتخاب این گزینه ها می توان بزرگنمایی و کوچکنمایی تصویر را به دلخواه تنظیم نمود.
- **Toolbars**: قسمت های مختلف این گزینه در شکل ۶-۱۳ نمایش داده شده است. با انتخاب هر یک از این گزینه ها نوار ابزار مربوطه در محیط کاری نرم افزار نمایش داده می شود.



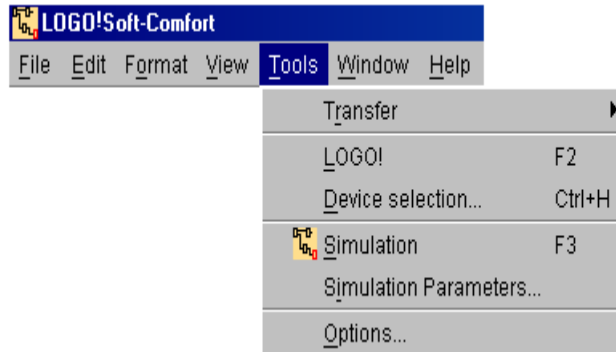
شکل ۶-۱۳: گزینه Toolbars

- **Status Bar, Info Window** و **Tool Tips**: با انتخاب این گزینه ها می توان پنجره اطلاعات پروژه و نوار وضعیت برنامه را در محیط نرم افزار نمایش داد.



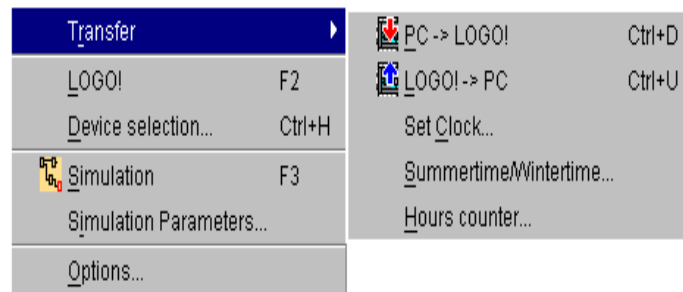
## منوی Tools:

گزینه های مختلف موجود در این منو در شکل ۶-۱۴ نمایش داده شده است .



شکل ۶-۱۴: منوی Tools

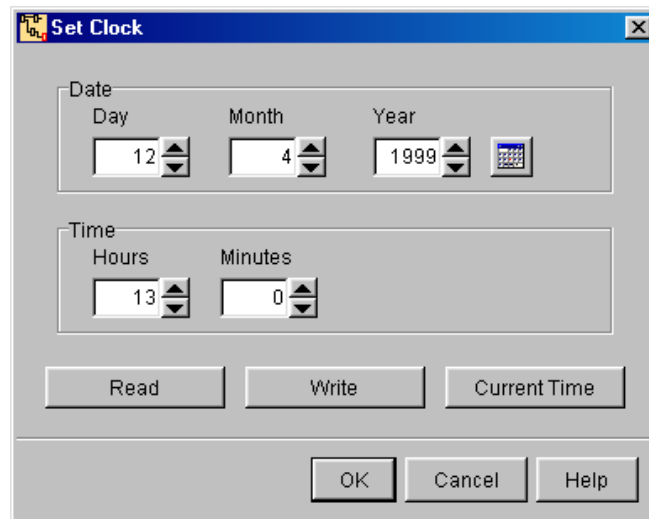
- **Transfer**: جهت مبادله اطلاعات مابین PC و LOGO و همچنین کارتهای حافظه از این گزینه استفاده می گردد . قسمت های مختلف این گزینه در شکل ۶-۱۵ نمایش داده شده است .
- ۱- **LOGO -> PC**: جهت انتقال اطلاعات از کامپیوتر به LOGO از این گزینه استفاده می شود . ( کلید میانبر **Ctrl + D** )
- ۲- **LOGO -> PC**: جهت انتقال اطلاعات از LOGO به PC از این گزینه استفاده می شود . ( کلید میانبر **Ctrl + U** )



شکل ۶-۱۵: گزینه Transfer

- ۳- **Set Clock**: از این گزینه جهت تنظیم زمان و ساعت داخلی موجود در LOGO استفاده می شود . صفحه تنظیمات ساعت در شکل ۶-۱۶ ارائه شده است .
- در این صفحه می توان با انتخاب گزینه **Current time** تاریخ و زمان فعلی کامپیوتر را انتخاب نموده و سپس با انتخاب گزینه **Write** این اطلاعات را به LOGO انتقال داد . همچنین می توان هر تاریخ و زمان مورد نظر دیگر را نیز از طریق کلید های مربوطه انتخاب نمود و به LOGO انتقال داد . از طریق گزینه **Read** نیز می توان اطلاعات ساعت موجود بر روی LOGO را به کامپیوتر انتقال داده و آنرا ویرایش نمود .
- در صورت عدم برقراری ارتباط مابین PC و LOGO به هنگام استفاده از دستورات **Read** و **Write** پیغام خطای **Cannot open the interface** ظاهر می گردد .





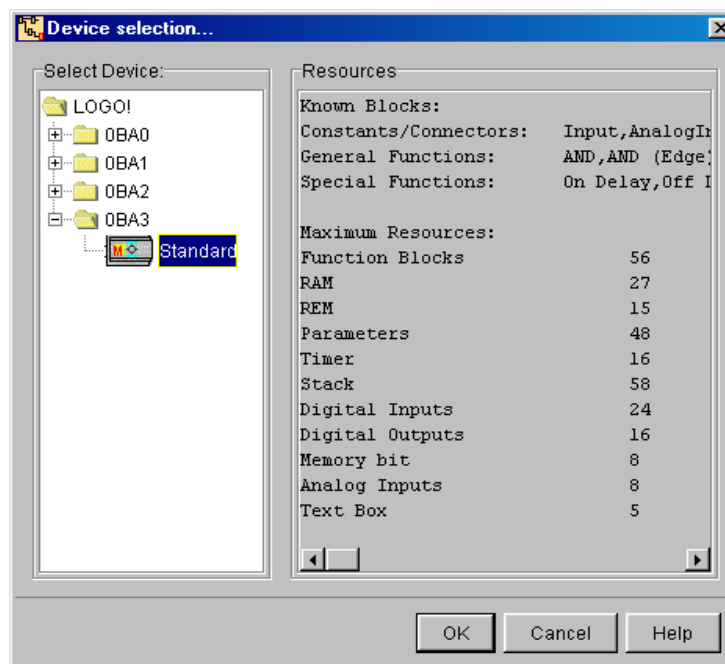
شکل ۳-۱۶: صفحه تنظیم ساعت

۴- **Summertime/Wintertime**: این قابلیت تنها در مدل 0BA3 وجود داشته و جهت تنظیم ساعت تابستانی و زمستانی مورد استفاده قرار می گیرد.

۵- **Hourse Counter**: این گزینه نیز تنها در مدل 0BA3 وجود داشته و جهت تنظیم شمارنده ساعت داخلی استفاده می گردد.

- **LOGO**: با انتخاب این گزینه و یا فشردن کلید F2 می توان حداقل نوع LOGO مورد نیاز را در نوار ابزار وضعیت مشاهده نمود.

- **Device selection**: با انتخاب این گزینه می توان انواع مختلف PLC را که توسط نرم افزار پشتیبانی می گردند را مشاهده کرد. همچنین می توان این صفحه را با کلیک بر روی نماد LOGO در نوار ابزار وضعیت رویت نمود. در این صفحه همچنین می توان مشخصات مربوط به هر یک از این نمونه ها را نیز بدست آورد. شکل ۶-۱۷ این صفحه را برای مدل 0BA3 نمایش می دهد.



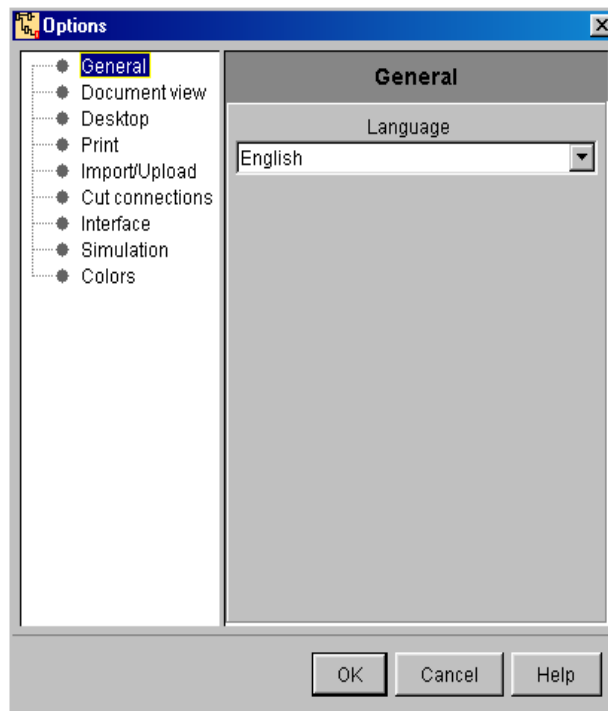
شکل ۶-۱۷: گزینه Device selection



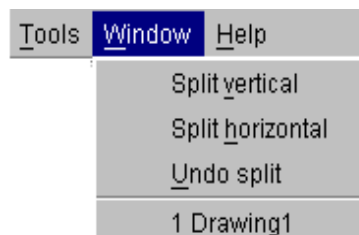
- **Simulation** : با انتخاب این گزینه و یا زدن کلید F3 صفحه شبیه سازی باز شده و امکان تست مدار فراهم می آید .
  - **Simulaton Parameters** : با انتخاب این گزینه اطلاعات مربوط به ورودی های استفاده شده در مدار از جمله اسم ، نوع و ... در صفحه ای نمایش داده خواهد شد .
  - **Option** : با کلیک بر روی این عبارت صفحه **Option** مطابق شکل ۶-۱۸ باز شده که شامل اطلاعات مربوط به عملکرد نرم افزار بوده که می توان آنها را به دلخواه تغییر داد . این اطلاعات عبارتند از :
    - ۱- **General**: اطلاعات عمومی مربوط به نرم افزار از قبیل زبان استفاده شده در نرم افزار می باشد
    - ۲- **Document View** : نحوه نمایش پروژه های باز شده در صفحه اصلی را نمایش می دهد . همچنین در آن می توان کلیدی را که جهت گردش در بین صفحات استفاده می شود را نیز تعیین نمود .
    - ۳- **Desktop** : اطلاعاتی که می خواهیم بر روی صفحه نمایش داده شوند در این گزینه انتخاب می نمایم .
    - ۴- **Print** : اطلاعاتی که در زمان چاپ کردن صفحه مورد نیاز است در این صفحه انتخاب می شوند .
    - ۵- **Import / Upload** : در این صفحه می توان قابلیت حذف آن ارتباطاتی را که در زمان انتقال و با وارد کردن اطلاعات می بایست بر اساس ضوابط خاص جدا شوند را انتخاب نمود .
    - ۶- **Cut Connection** : به قسمت **Cut Lines** در منوی **Edit** مراجعه نمایید .
    - ۷- **Interface** : از این گزینه جهت تعریف پورت ارتباطی کابل لینک استفاده می گردد .
    - ۸- **Simulation** : در این گزینه می توان دو عبارت مربوط به نمایش وضعیت سیگنالها بر روی نمودار اصلی را در حالت شبیه سازی معین کرد . به عبارت دیگر می توان مشخص نمود که آیا نرم افزار وضعیت فعال یا غیرفعال بودن هر بلوک را بر روی نمودار اصلی نیز نمایش دهد یا خیر .
    - ۹- **Colors** : وضعیت رنگ محیط شماتیک را می توان در این صفحه تغییر داد . از جمله می توان به قابلیت تغییر رنگ های مربوط به پس زمینه و وضعیت سیگنال یک و صفر در حالت شبیه سازی اشاره نمود .
- منوی Window :**
- گزینه های موجود در این منو جهت چیدمان صفحات باز شده به صورت افقی و یا عمودی استفاده می گردد . شکل ۶-۱۹ این منو را نمایش می دهد .
- منوی Help :**
- آخرین منو از نوار منو می باشد که با کلیک بر روی آن گزینه هایش پدیدار می گردد . شکل ۶-۲۰ این منو را نمایش می دهد . گزینه های مختلف این منو عبارتند از :
- **Contents** : فهرست موضوعی راهنمای نرم افزار در این قسمت قابل رویت است .
  - **Direct Help** : با انتخاب این گزینه ، نشانگر موس به صورت فلش درآمده که با کلیک بر روی هر آیتم راهنمایی درباره آن موضوع خاص نمایش داده می شود .
  - **Update center** : از طریق این گزینه می توان نرم افزار را از طریق اتصال به اینترنت و یا از طریق فایل های در دسترس به روز نمود .
  - **About** : با انتخاب این گزینه اطلاعاتی در خصوص نرم افزار ارائه می گردد .



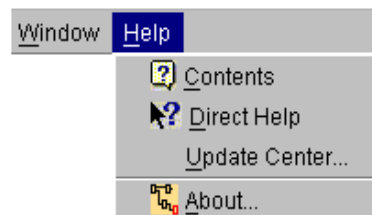




شکل ۶-۱۸: صفحه Option



شکل ۶-۱۹: منوی Window



شکل ۶-۲۰: منوی Help



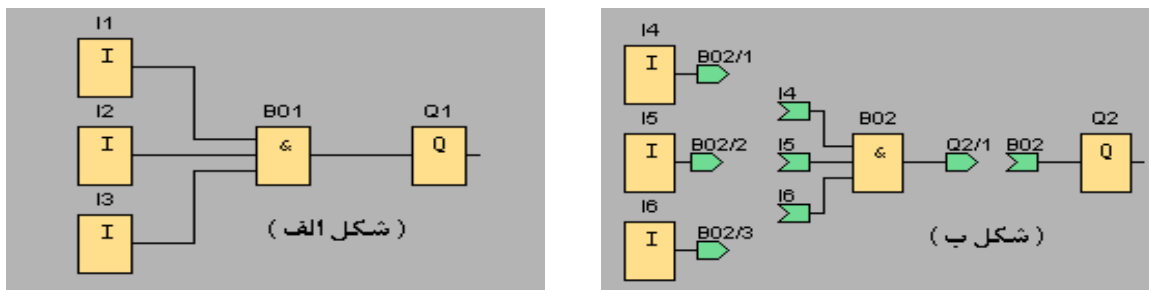
### نوار ابزار ها :

این نوار که در شکل ۶-۴ نیز معرفی شده است ، قسمت اصلی ابزارهای قابل استفاده در نرم افزار را شامل می گردد . با توجه به اهمیت موضوع ، این ابزارها در زیر به طور کامل ارائه می شوند ولیکن جهت تسلط یافتن به کار با این ابزارها پس از مطالعه مثالهای کاربردی ارائه شده در این جزوه نیاز به نوشتن برنامه های متنوع توسط شما خواننده محترم وجود دارد .

ابزار انتخاب : از این ابزار می توان جهت انتخاب یک یا چند شی موجود در صفحه اصلی استفاده نمود . جهت این کار پس از انتخاب ابزار ، بر نقطه ای از صفحه اصلی کلیک کرده و در همان حال بدون رها کردن موس بگونه ای در صفحه حرکت نمایید که اشیاء مورد نظر در کادر مربوطه قرار گیرند . در این حال با رها کردن موس اشیاء مورد نظر بصورت انتخاب شده نمایش داده می شوند . حال می توان عملیات مورد نظر از قبیل انتقال و پاک نمودن و ... را بطور همزمان بر روی کل اشیاء اعمال نمود .

ابزار نوشتن متن : با استفاده از این ابزار می توان اطلاعات مورد نظر را بر روی صفحه اصلی پروژه تایپ نمود. جهت تغییر فونت و رنگ نوشته از گزینه Font در منوی Format استفاده نمایید . ( رجوع شود به توضیحات منوی Format )

ابزار قطع و اتصال ارتباطات : در صفحه اصلی جهت پرهیز از به هم ریختگی اطلاعات و همچنین جهت آسان شدن مطالعه آن ، در برخی موارد لازم می باشد که کل ارتباطات بصورت سیم بندی نمایش داده نشوند ، بدین لحاظ از ابزار مورد نظر جهت قطع ارتباطات سیمی و نمایش آن بصورت Lable استفاده می گردد . جهت این کار ابزار فوق الذکر را انتخاب کرده و سپس بر روی اتصال مورد نظر کلیک نمایید . با ملاحظه مثال ارائه شده در شکل ۶-۲۱ می توانید به تفاوت دو طرز نمایش پی ببرید .



شکل ۶-۲۱: (الف) : نمایش ارتباط از طریق اتصال مستقیم (ب) : نمایش ارتباط از طریق Lable

ابزار برقراری ارتباط : جهت برقراری ارتباط و یا بعبارت دیگر سیم کشی مدار از این ابزار استفاده می گردد .



جهت اینکار ، ابزار را انتخاب کرده و در پایه های ورودی و خروجی اشیاء کلیک می نماییم ، بدین صورت سیم به پایه مورد نظر اتصال می یابد . قبل از برقراری ارتباط در پایه مقصد شماره پایه بصورت خودکار در صفحه نمایش داده می شود .

### : Constants/ Connectors

Co

با استفاده از این ابزار می توان المانهای نمایش دهنده ورودی های دیجیتال و آنالوگ ، خروجی ها ، ورودی و خروجی های Asi ، بیت های حافظه ، سطح منطقی بالا (High) و سطح منطقی پایین (Low) را انتخاب کرده و در صفحه کار اصلی قرار داده و در طراحی مدار از آنها استفاده نمود . لازم بذکر است که قبل از استفاده از این قطعات می بایست نرم افزار را با نوع LOGO در دسترس هماهنگ نمود زیرا با توجه به نوع LOGO مورد استفاده ممکن است برخی از المانهای فوق الذکر قابل استفاده نبوده و یا تعداد در دسترس آنها تفاوت نماید . شکل ۶-۲۲ المانهای موجود در این ابزار را برای LOGO نوع 0BA3 استاندارد نمایش می دهد .



شکل ۶-۲۲: ابزار Constants/ Connectors

در این شکل بلوک I به معنی ورودی دیجیتال ، AI ورودی آنالوگ ، Q خروجی ، M بیت حافظه ، hi سطح بالای منطقی و lo سطح پایین منطقی می باشد . هر یک از این المانها دارای تعداد مشخصی می باشند که نرم افزار آنرا با توجه به نوع PLC انتخابی توسط کاربر مشخص کرده و در صورت استفاده تمام آنها در یک پروژه گزینه مورد نظر غیر قابل دسترسی می گردد . حال اگر کاربر نرم افزار را با نوع PLC در دسترس هماهنگ ننماید ، ممکن است تعداد المانهای مورد استفاده در پروژه بیش از تعداد موجود در PLC باشد که این امر در هنگام بار کردن اطلاعات بر روی PLC مشکل ساز خواهد شد .  
حال به طور مفصل تر این قطعات را مورد مطالعه قرار داده و برگه های مشخصات آنها را بررسی می نماییم .

### : Digital Input

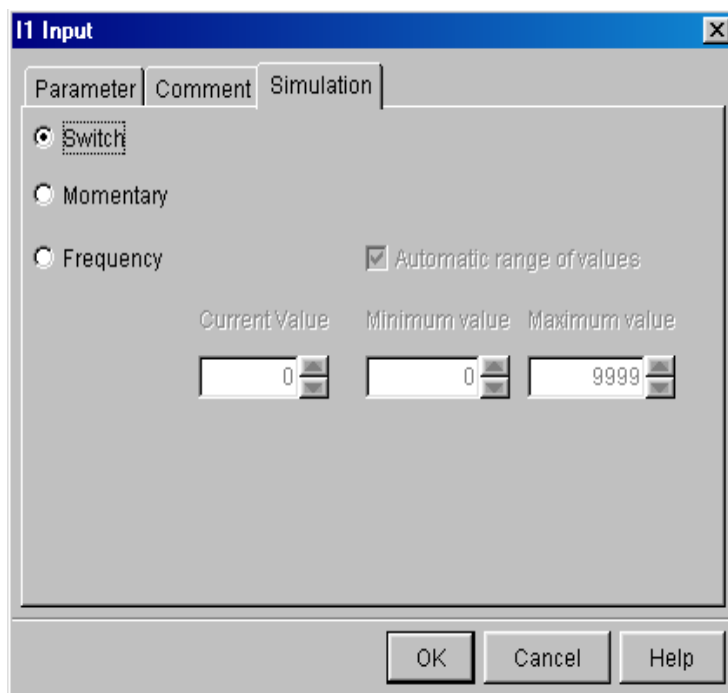
I

با انتخاب یک ورودی و قراردادن آن در صفحه کار این ورودی نمایانگر ورودی هم نام خود بر روی سخت افزار خواهد بود . با دو بار کلیک چپ بر روی این بلوک و یا با انتخاب بلوک مورد نظر و کلیک راست بر روی آن و نهایتاً انتخاب گزینه Block Properties صفحه ای مطابق شکل ۶-۲۳ باز می گردد که صفحه مشخصات این قطعه می باشد . در این صفحه سه برگه وجود دارد که عبارتند از : Comment ، Parameter و simulation .

#### • برگه Parameter :

در برگه پارامتر شماره بلوک ورودی انتخاب می گردد . قابل ذکر است که به هنگام قرار دادن بلوک ورودی در صفحه کار ، نرم افزار بطور خودکار شماره ای به آن نسبت می دهد که از طریق تعویض اطلاعات برگه پارامتر می توانیم آنرا تغییر دهیم .





شکل ۶-۲۳: صفحه مشخصات بلوک ورودی دیجیتال

- برگه **Comment**: در این برگه می توان توضیحات مورد نیاز جهت ورودی مورد نظر را یادداشت کرده تا بر روی صفحه اصلی نمایش داده شود.
- برگه **Simulation**: در این صفحه می توان نوع ورودی را با توجه به نیاز در مدار تعریف نمود. سه گزینه موجود عبارتند از:
  - ۱- **Switch**: کلید قطع و وصل
  - ۲- **Momentary**: کلید لحظه ای ( شستی فشاری )
  - ۳- **Frequency**: کلید فرکانسی با قابلیت تنظیم حد بالا و پایین ( ورودی سرعت بالا )

### AI : Analog Input

این ورودی که در انواع خاصی از مدل های LOGO وجود دارد، جهت دریافت مستقیم اطلاعات آنالوگ از سنسورها و یا هر وسیله دیگر استفاده می گردد. این ورودی ها به عملگرهای ویژه آنالوگ که در قسمت عملگرهای ویژه معرفی خواهند شد، متصل می گردند. اطلاعات ورودی می تواند ولتاژ 0 تا 10 ولت و یا جریان 0 تا 20 میلی آمپر باشند. در برگه مشخصات این ورودی ها که مشابه شکل ۳-۲۳ می باشد در قسمت **Simulation** بجای اطلاعات نمایش داده شده در شکل ۳-۲۳ حدود بالا و پایین قابل تنظیم جهت این ورودی ها وجود دارد.

### Q : Output

این بلوک نمایانگر خروجی های مدار می باشد. در برگه مشخصات خروجی ها تنها دو برگه **Parameter** و **Comment** وجود دارد. این برگه ها همانند برگه های همنام در بلوک های ورودی می باشد.



**M**

**Marker :**

**Marker** یا بیت های حافظه که از پرکاربردترین بلوک های موجود می باشد ، جهت برقراری ارتباط مابین بلوک هایی که ورودی های آنها بطریقی از خروجی خود آنها تغذیه می گردد ، استفاده می شوند تا ارتباط لازم برقرار گردد . در صورت عدم استفاده از این بیت ها در مسیر های ارتباطی فوق الذکر تلاش جهت برقراری ارتباط بی حاصل خواهد بود . از آنجا که تغییرات در خروجی این بیت ها در انتهای سیکل زمانی انجام می گیرد لذا با ایجاد تاخیر در روند تحلیل مدار باعث می شود که ابتدا سیگنالی که از خروجی یک بلوک گرفته شده تغییرات خود را انجام داده و سپس به ورودی آن اعمال گردد . حداکثر هشت عدد بیت حافظه در اختیار است که می توان آنها را در برنامه بکار برد .

بیت حافظه **M8** دارای خصوصیت منحصر بفردی می باشد . این بیت در اولین سیکل برنامه یک گشته و پس از اولین سیکل بطور خودکار صفر می شود ، لذا می توان از این بیت بعنوان یک بیت شناسایی استفاده نمود . در سایر موارد می توان از این بیت همانند سایر بیت های حافظه استفاده نمود . صفحه مشخصات این بلوک نیز شامل دوبرگه **Parameter** و **Comment** می باشد .

**hi**

**High :**

این بلوک نمایانگر سطح ولتاژ بالا در منطق دیجیتال می باشد . بعبارت دیگر با استفاده از این بلوک می توان بطور سخت افزاری یک ارتباط مابین تغذیه **LOGO** و نقاط مورد نظر مدار ایجاد نمود . صفحه مشخصات این بلوک تنها دارای برگه توضیحات می باشد .

**lo**

**Low :**

برخلاف بلوک **High** این بلوک بیانگر سطح ولتاژ پایین در مدار می باشد .

### عملگر های قابل استفاده در LOGO :

**LOGO** دارای دو نوع عملگر می باشد که به دودسته کلی عملگر های عمومی یا اساسی و عملگر های ویژه تقسیم بندی گردیده اند . گروه اول شامل گیت های پایه دیجیتال از قبیل گیت های **AND** ، **OR** ، **NOT** و ... می باشد . در این گروه عملگرهایی نیز وجود دارد که به لبه پالس ورودی حساس بوده و تحت عنوان عملگرهای پایه حساس به لبه در ادامه معرفی خواهند گشت . گروه دوم عملگر ها که تحت عنوان عملگر های ویژه معرفی خواهند گشت شامل شمارنده ها ، زمان سنج ها و منابع تولید پالس و ... می باشند . یادآوری مجدد این نکته ضروری است که نوع و تعداد عملگرهای قابل استفاده نیز در انواع مختلف **LOGO** متفاوت می باشد که در استفاده از آنها می بایست این نکته را مد نظر قرار داد .





**: General Function**

این عملگرها که تحت نام عملگرهای عمومی معرفی گردیده اند، پر استفاده ترین عملگرها در طراحی مدارات فرمان صنعتی می باشد. تعریف عمومی این عملگرها در فصل اول کتاب بطور کامل ارائه شده است. لذا در این قسمت به معرفی بلوک های مرتبط به این عملگرها پرداخته خواهد شد.



**: AND**

گیت AND موجود یک بلوک با سه ورودی می باشد. خروجی این گیت در صورتی فعال می گردد که هر سه ورودی یک گردند. جدول عملکرد مربوط به این بلوک در شکل ۶-۲۴ ارائه شده است. در این بلوک هر گاه یکی از ورودیها باز بماند بصورت خودکار به سطح ولتاژ بالا که عنصر خنثی در گیت AND می باشد متصل می گردد.

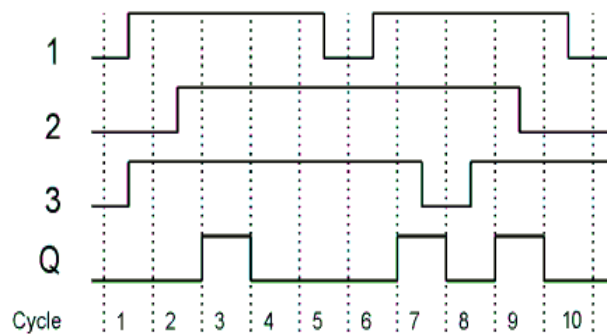
Input 1	Input 2	Input 3	Output
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

شکل ۶-۲۴: جدول عملکرد گیت AND با سه ورودی



**: AND with edge evaluation**

این بلوک یک گیت AND سه ورودی با عملکردی خاص می باشد که جهت استفاده در این مینی PLC طراحی شده است. با توجه به نمودار زمانی عملکرد این گیت که در شکل ۶-۲۵ ارائه شده است، می توان گفت در این گیت هرگاه دو ورودی در سطح بالا باشند با وارد شدن یک لبه بالا رونده در ورودی سوم، خروجی بمدت یک سیکل زمانی به سطح بالا رفته و مجدداً صفر می گردد. در صورتی که یکی از ورودیها باز بماند به طور خودکار به سطح ولتاژ بالا متصل می گردد.



شکل ۶-۲۵: نمودار زمانی گیت AND با تحریک لبه ای



از جمله کاربردهای این گیت می توان به کاربرد آن بعنوان یک آشکار ساز لبه سیگنال اشاره نمود .

**&\***  
: NAND

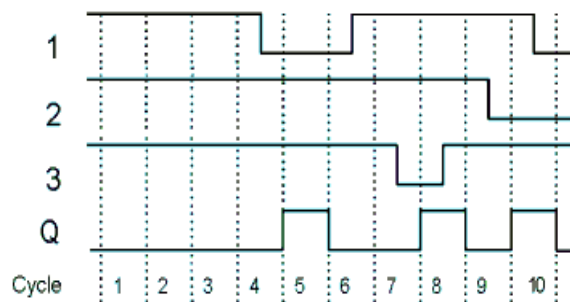
این گیت مکمل شده گیت AND می باشد که جدول عملکردی همانند جدول ارائه شده در شکل ۶-۲۶ دارد . خروجی این گیت همواره یک می باشد بجز زمانی که هر سه ورودی همزمان یک شوند.

Input 1	Input 2	Input 3	Output
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

شکل ۶-۲۶: جدول عملکرد گیت NAND با سه ورودی

**&↓\***  
: NAND with edge evaluation

در این گیت سه ورودی که نمودار زمانی عملکرد آن در شکل ۶-۲۷ ارائه شده است ، هرگاه دو ورودی یک باشند پس از ایجاد یک لبه منفی در ورودی سوم ، خروجی بمدت یک سیکل زمانی یک شده و سپس به سطح صفر باز می گردد .



شکل ۶-۲۷: نمودار زمانی گیت NAND با تحریک لبه ای

**≥1**  
: OR

در این گیت سه ورودی هر گاه حداقل یکی از ورودی ها یک باشد خروجی یک خواهد شد . شکل ۶-۲۸ جدول عملکرد این گیت را نمایش می دهد .

در این گیت هرگاه یکی از ورودی ها مورد استفاده قرار نگیرد نرم افزار بطور خودکار آنرا به سطح منطقی صفر که عنصر خنثی در گیت OR می باشد، متصل می نماید .



Input 1	Input 2	Input 3	Output
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

شکل ۶-۲۸: جدول عملکرد گیت OR

: NOR

≥1

این گیت مکمل گیت OR می باشد که در آن بجز زمانی که هر سه ورودی صفر می باشند ، خروجی همواره صفر می باشد . شکل ۶-۲۹ جدول عملکرد این گیت را نمایش می دهد .

Input 1	Input 2	Input 3	Output
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

شکل ۶-۲۹: جدول عملکرد گیت NOR

: XOR

=1

در گیت XOR که دارای دو ورودی می باشد خروجی در صورتی دارای منطق یک خواهد بود که ورودی ها متفاوت باشند . شکل ۶-۳۰ جدول عملکرد این گیت را نمایش می دهد .

Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	0

شکل ۶-۳۰: جدول عملکرد گیت XOR با دو ورودی





1. NOT :

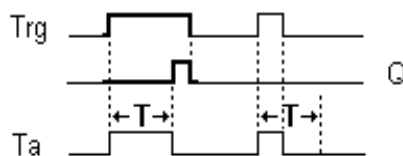
این بلوک یک گیت NOT را شبیه سازی می نماید که ورودی خود را مکمل می نماید . عبارت دیگر در صورت اعمال ورودی یک خروجی صفر می گردد و بالعکس .  
با وجود گیت NOT ، LOGO نیازی به کنتاکت بسته (Normally Closed) ندارد بلکه می توان با استفاده از یک کنتاکت باز (Normally Open) که با یک بلوک ورودی تعریف می گردد و پس از آن استفاده از یک گیت NOT این کنتاکت بسته را مدلسازی نمود .

SF : Special Function

این عملگرهای ویژه ، عملگرهای با کاربرد خاص بوده که از ویژگی های قابل توجهه LOGO می باشند . نحوه عملکرد این عملگرها بطور مفصل در ادامه شرح داده خواهد شد .

On delay Trg Q

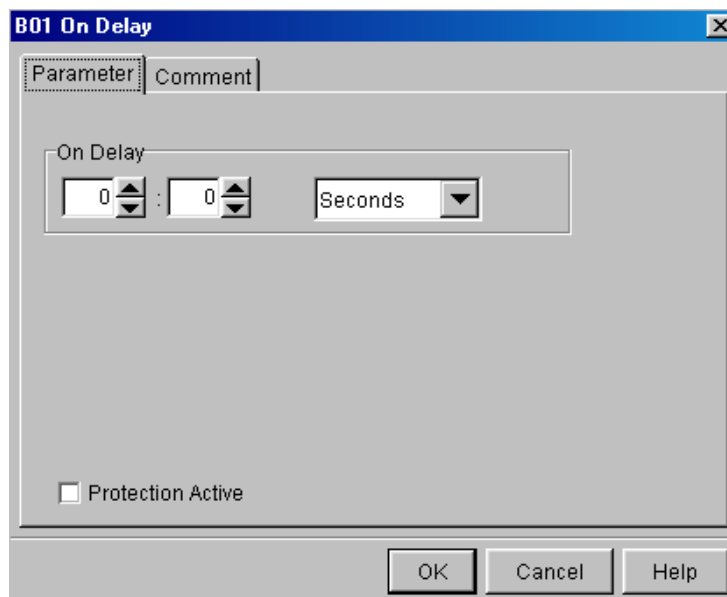
در On delay یا تایمر با تاخیر در وصل شدن ، خروجی تا قبل از گذشت یک زمان تعریف شده نمی تواند یک گردد . با توجه به نمودار زمانی عملکرد این تایمر که در شکل ۶-۳۱ ارائه شده است ، می توان گفت در این تایمر پس از ایجاد یک لبه بالا رونده در پایه ورودی تریگر و پس از گذشت مدت زمان T که توسط کاربر در برگه Parameter از صفحه مشخصات این تایمر تعریف می گردد ، خروجی به سطح بالا می رود . نکته قابل توجهه در این نوع تایمر این است که در طی مدت زمان T ، سیگنال اعمال شده به ورودی تریگر می بایست در سطح یک باقی بماند، در غیر این صورت خروجی فعال نخواهد شد. عبارت دیگر شرط فعال شدن خروجی پس از اعمال سیگنال تریگر ، یک ماندن سیگنال تریگر می باشد ، لذا پس از فعال شدن خروجی ، این مقدار مادامی که پایه تریگر صفر نگردد ، فعال باقی می ماند .



شکل ۶-۳۱ : دیاگرام زمانی عملکرد تایمر On delay

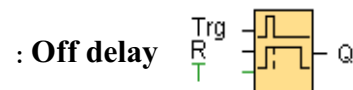
در صفحه مشخصات این بلوک که در شکل ۳-۳۲ ارائه شده است ، دو برگه Parameter و Comment وجود دارد در برگه Comment توضیحات مورد نیاز جهت نمایش بر روی صفحه کار تایپ می گردد . برگه پارامتر نیز جهت تعیین مدت زمان T استفاده می گردد. در این برگه ، اطلاعات زمانی رامی توان در سه محدوده زمانی تعیین نمود . با انتخاب گزینه Seconds اولین عبارت بیانگر ثانیه و عبارت دوم بیانگر صدم ثانیه می باشد . (شکل ۶-۳۲ را ملاحظه نمایید .) با انتخاب گزینه Minutes ، اولین عبارت بیانگر دقیقه و عبارت دوم بیانگر ثانیه می باشد . با انتخاب گزینه Hours نیز اولین عبارت بیانگر ساعت و عبارت دوم بیانگر دقیقه خواهد بود.



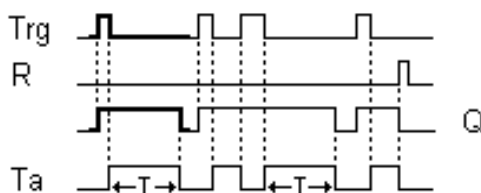


شکل ۶-۳۲: صفحه مشخصات تایمر On delay

با فعال سازی گزینه **Protection Active** این امکان بوجود می آید که پس از انتقال برنامه به LOGO، امکان نمایش و ایجاد تغییرات بر روی پارامتر **T** در حالت تنظیمات پارامتر وجود نداشته باشد. در این حالت تغییرات تنها در حالت برنامه نویسی ممکن خواهد بود. (این مسئله در ضمیمه ۶ که نحوه برنامه نویسی با صفحه کلید LOGO توضیح داده خواهد شد، بطور کامل بررسی می شود).



این تایمر نوعی تایمر با تاخیر در قطع شدن می باشد. این تایمر دارای دو ورودی قابل اتصال و یک ورودی پارامتر می باشد. ورودی تریگر (**Trg**) بعنوان فعال ساز این تایمر مورد استفاده قرار می گیرد و پایه ورودی **Reset (R)** که جهت صفر نمودن خروجی از آن استفاده می شود. با توجه به نمودار زمانی عملکرد این تایمر که در شکل ۶-۳۳ ارائه شده است می توان گفت در صورتی که ورودی **R** در سطح پایین باشد، به محض وارد شدن یک لبه بالا رونده در ورودی تریگر، خروجی **Set** می گردد، ولیکن مدت زمان خاموش شدن **T** (که از طریق برگه پارامتر وارد می گردد) پس از ایجاد لبه پایین رونده در ورودی تریگر محاسبه می گردد. پس از اتمام مدت زمان **T** خروجی به سطح صفر باز می گردد. در این تایمر هرگاه قبل از اتمام مدت زمان مشخص **T**، پایه تریگر دوباره تحریک گردد، زمان **T** مجدداً از آن لحظه اندازه گیری می گردد. بعبارت دیگر زمان **T** بمدت زمانی که تا آن لحظه محاسبه شده است اضافه می گردد. در این زمان سنج هرگاه ورودی **R** فعال گردد خروجی در صورت یک بودن، به سطح صفر باز می گردد.



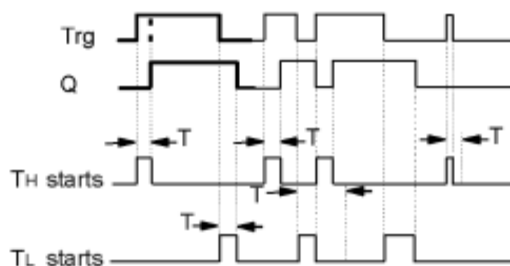
شکل ۶-۳۳: دیاگرام زمانی تایمر Off delay



صفحه مشخصات تایمر Off delay مشابه تایمر On delay می باشد که در شکل ۶-۳۲ ارائه شده است .

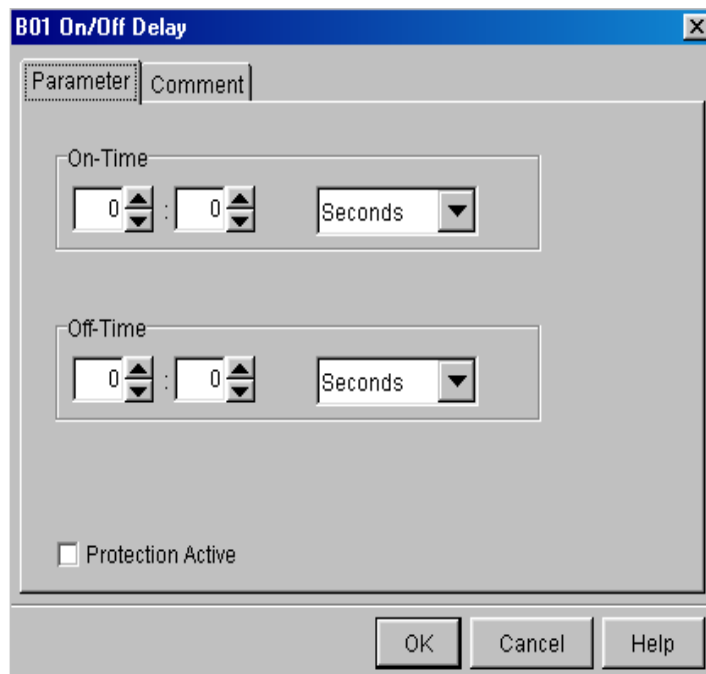


این نوع تایمر ترکیبی از دو نوع قبل بوده که نمودار زمانی آن در شکل ۶-۳۴ ترسیم شده است . در این تایمر با اعمال پالس به ورودی تریگر پس از مدت زمان TH خروجی فعال می گردد و با ایجاد لبه پایین رونده در پایه تریگر خروجی پس از مدت زمان TL به سطح صفر باز می گردد . زمانهای TH و TL را می توان از طریق برگه پارامتر در صفحه مشخصات مربوط به این تایمر تنظیم نمود . این برگه در شکل ۶-۳۵ نمایش داده شده است .



شکل ۶-۳۴: نمودار زمانی عملکرد تایمر On - Off delay

در صورتی که ورودی تریگر قبل از اتمام مدت زمان TH به سطح پایین بازگردد ، خروجی هیچ تغییری نمی نماید . همچنین اگر قبل از اتمام زمان TL ورودی تریگر به سطح بالا بازگردد ، زمان Reset می گردد .

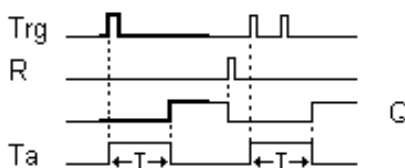


شکل ۶-۳۵: صفحه مشخصات تایمر On - Off delay





این بلوک یک تایمر با تاخیر در وصل حافظه دار است. به عبارت دیگر تفاوت این تایمر با تایمر On delay آن است که در این تایمر نیازی به فعال ماندن پایه تریگر تا انتهای عملیات نمی باشد بلکه تنها یک پالس کوتاه باعث فعال شدن این تایمر می گردد. شکل ۶-۳۶ نمودار زمانی این تایمر را بیان می کند.



شکل ۶-۳۶: دیاگرام زمانی تایمر Retentive on delay

نکته قابل توجه در این تایمر آن است که اگر قبل از اتمام مدت زمان T پالس تریگر دیگری به این بلوک وارد آید بی تاثیر خواهد بود. پایه ورودی Reset نیز جهت صفر نمودن خروجی مورد استفاده قرار می گیرد. در صفحه مشخصات این تایمر علاوه بر برگه توضیحات، برگه پارامتر نیز وجود دارد که یک زمان تاخیر مربوط به روشن شدن تایمر را می توان در آن تنظیم نمود.

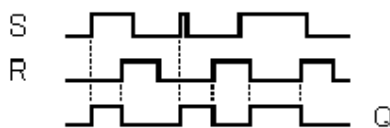


رله قفل شونده یا فلیپ فلاپ RS مشابه یک سلول حافظه باینری عمل می نماید. وضعیت خروجی به وضعیت ورودی های S و R و همچنین به حالت قبلی خروجی بستگی دارد. جدول ارائه شده در شکل ۶-۳۷ عملکرد این فلیپ فلاپ را بیان می کند.

S	R	Q
0	0	حالت قبلی خود را نگه می دارد
0	1	Reset
1	0	Set
1	1	Reset

شکل ۶-۳۷: جدول عملکرد فلیپ فلاپ RS

در این بلوک پایه Reset نسبت به پایه Set تقدم دارد، عبارت دیگر در زمان اعمال همزمان پالس به این دو ورودی خروجی صفر خواهد شد. شکل ۶-۳۸ نمودار زمانی عملکرد این فلیپ فلاپ را بیان می کند. در این نمودار مشاهده می شود که هرگاه هر دو ورودی بطور همزمان صفر باشند خروجی هر مقداری را که قبل از این زمان داشته است حفظ می نماید که این حالت Latching فلیپ فلاپ را بیان می کند.



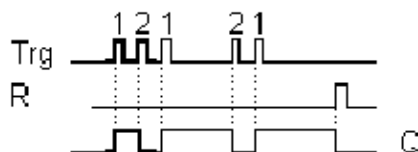
شکل ۶-۳۸: نمودار زمانی عملکرد فلیپ فلاپ RS



در برگه پارامتر این بلوک گزینه ای تحت نام **Retention** وجود دارد که با انتخاب آن، خروجی این فلیپ فلاپ در صورت قطع برق نیز حفظ خواهد شد. در این صورت با وصل مجدد تغذیه خروجی این فلیپ فلاپ برابر مقدار آن قبل از قطع برق خواهد بود.



این بلوک که به نام رله ضربه شهرت دارد، دارای نمودار عملکردی همانند دیاگرام نمایش داده شده در شکل ۳۹-۶ می باشد. در این رله ضربه یا رله پالسی با هر بار تریگر مدار وضعیت خروجی مکمل می گردد. یعنی اگر خروجی در وضعیت بالا باشد با اعمال یک پالس تریگر خروجی صفر خواهد شد و برعکس.

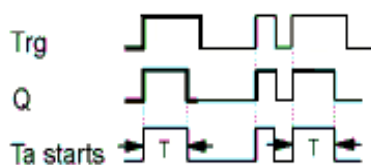


شکل ۳۹-۶: نمودار زمانی عملکرد P ulse Relay

در این بلوک خروجی در هر وضعیتی باشد با تحریک پایه **Reset**، صفر خواهد شد. این بلوک نیز همانند فلیپ فلاپ **RS** دارای گزینه **Retention** در برگه پارامتر خود می باشد که کارآیی آن همانند گزینه همانام در فلیپ فلاپ **RS** می باشد.



با توجه به دیاگرام زمانی این بلوک که در شکل ۴۰-۶ نمایش داده شده است، با اعمال پالس به ورودی تریگر و فعال ماندن آن خروجی به سطح بالا رفته و پس از مدت **T** به صفر باز می گردد. در صورت اعمال تریگر مجدد قبل از اتمام زمان قبلی، زمان **T** مجدداً اندازه گیری می شود بدین لحاظ به این بلوک رله پاک کننده گفته می شود.



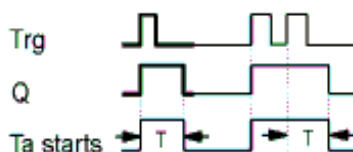
شکل ۴۰-۶: نمودار زمانی عملکرد Wiping Relay

در برگه پارامتر مربوط به این بلوک قابلیت تنظیم زمان خاموش شدن و همچنین انتخاب قابلیت محافظت از پارامتر وجود دارد.



این بلوک یک **Wiping Relay** با تحریک لبه ای می باشد. با توجه به دیاگرام زمانی این بلوک در شکل ۴۱-۶ در می یابیم که با اعمال یک پالس کوتاه به ورودی تریگر خروجی بمدن زمان **T** در سطح بالا قرار گرفته و سپس صفر می گردد. در این بلوک نیز با تحریک مجدد پایه تریگر زمان **T** از ابتدا اندازه گیری می شود.





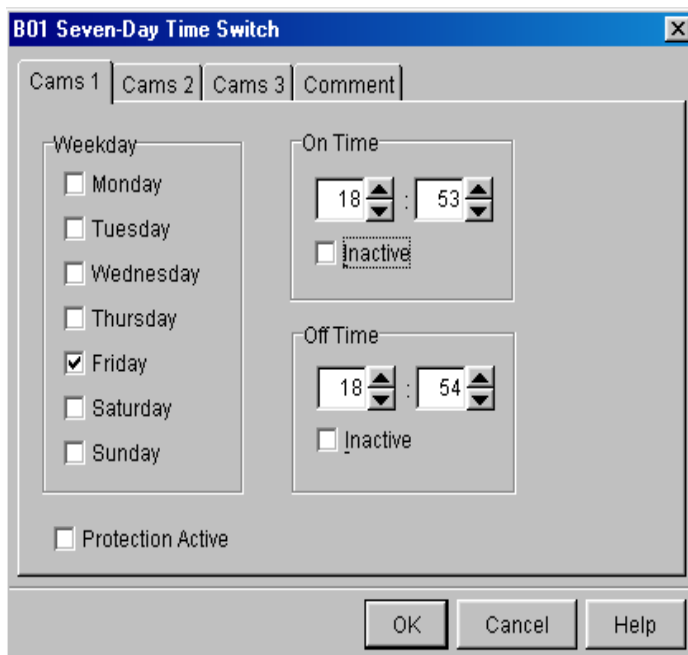
شکل ۶-۴۱: نمودار زمانی Interval Time-Delay Relay

صفحه مشخصات این بلوک نیز همانند Wiping Relay می باشد .

Seven-Day Time Switch 

این بلوک بعنوان یک کنترل کننده On/Off در تاریخهای قابل تعریف در یک هفته استفاده می گردد . هر بلوک دارای سه وضعیت قابل تنظیم می باشد . در هر یک از این وضعیت ها می توان زمان فعال یا غیر فعال شدن خروجی را با توجه به نیاز تعریف نمود . شکل ۶-۴۲ صفحه مشخصات این بلوک را نمایش می دهد . در این صفحه سه برگه با نام Cams وجود دارد که با انتخاب هر یک می توان زمان فعال یا غیر فعال شدن یک خروجی را تعیین نمود . جهت این کار ابتدا روز مورد نظر را انتخاب کرده و سپس با برداشتن علامت گزینه Inactive می توان زمان روشن و خاموش شدن را تعیین نمود .

نکته قابل توجه در این بلوک این است که برگه مشخصات Cam3 نسبت به Cam2 و برگه Cam2 نسبت به Cam1 دارای اولویت می باشند ، لذا اگر زمان روشن شدن در یک برگه با زمان خاموش شدن در برگه دیگر برابر باشد ، LOGO با توجه به اولویت برگه ها نسبت به آن تصمیم گیری می نماید . در صورتی که هیچیک از روزهای هفته انتخاب نگردد ، پروسه بصورت روزانه انجام می گیرد . همچنین می توان در یک برگه چندین روز را همزمان انتخاب کرد ، در این حالت در یک ساعت معین و در روزهای مشخص شده عمل مورد نظر انجام می گیرد .

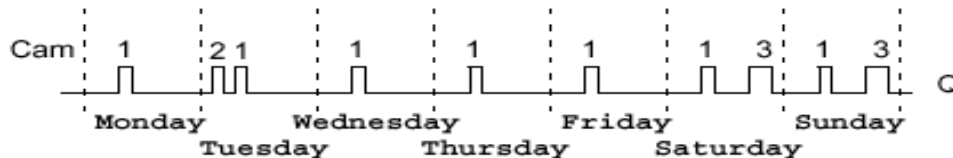


شکل ۶-۴۲: برگه مشخصات Seven-Day Time Switch



بعنوان مثال با انتخاب تاریخ های زیر در برکه های سه گانه خروجی مدار مطابق شکل ۶-۴۳ خواهد شد .

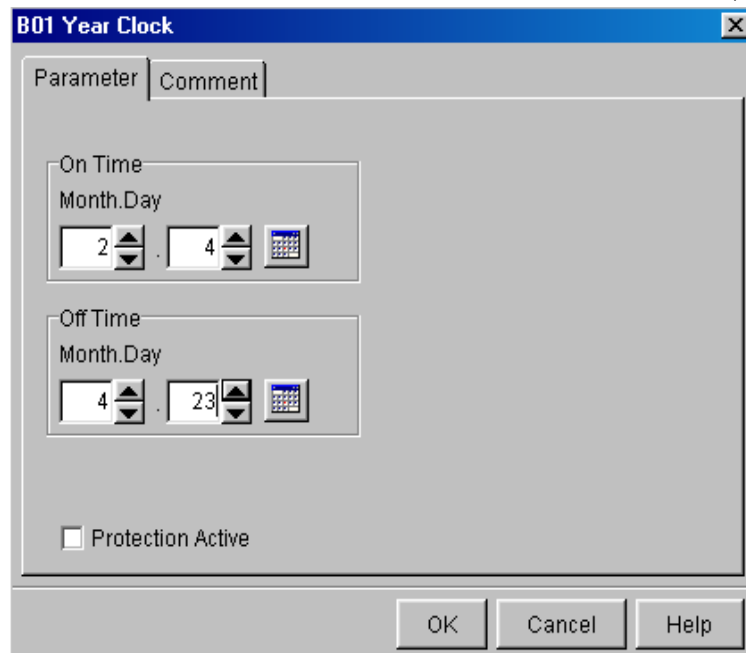
حالت ۱ : روزانه	۶/۳۰ تا ۸
حالت ۲ : سه شنبه	۳/۱۰ تا ۴/۱۵
حالت ۳ : شنبه و یکشنبه	۱۶/۳۰ تا ۲۳/۱۰



شکل ۶-۴۳ : نمونه ای از عملکرد زمان سنج هفتگی

:Yearly Timer Switch No MM  
DD Q

با استفاده از این تایمر می توان در یک روز سال، تایمر را روشن کرده ( خروجی آن را Set نمود ) و در روز دیگری از سال آن را خاموش نمود . با انتخاب یک روز ، در ساعت صفر آن روز تایمر عمل خواهد نمود. شکل ۶-۴۴ صفحه مشخصات این تایمر را نمایش می دهد . در برکه پارامتر این بلوک دو زمان روشن شدن و خاموش شدن قابل دسترس است . در هر یک از این قسمت ها گزینه سمت چپ بیانگر شماره ماه و زمان سمت راست بیانگر شماره روز در ماه مذکور می باشد .



شکل ۶-۴۴ : صفحه مشخصات Yearly Timer switch

تعداد قابل دسترسی از این نمونه بلوک نیز به نوع LOGO مورد استفاده بستگی دارد . به عنوان یک مثال دو بلوک از این تایمر را در نظر می گیریم که خروجی آنها از طریق گیت OR به یک خروجی مدار متصل گردیده است .

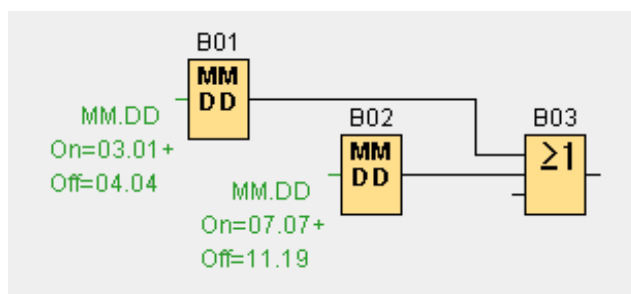


در این دو بلوک پارامترها بشکل زیر تعریف می گردد .

بلوک ۱: On Time = 03 : 01 و Off Time = 04 : 04

بلوک ۲: On Time = 07 : 07 و Off Time = 11 : 19

در این مثال در روز اول ماه سوم ( March ) خروجی فعال شده و در روز چهارم ماه چهارم ( April ) خروجی غیر فعال می گردد . مجدداً در روز هفتم ماه هفتم ( July ) خروجی فعال شده و در روز نوزدهم ماه یازدهم ( November ) خروجی صفر خواهد شد . مدار مورد نظر جهت این امر در شکل ۶-۴۵ ارائه شده است .

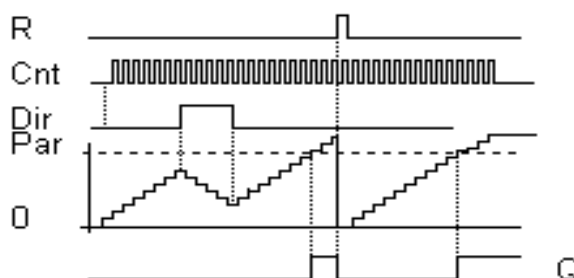


شکل ۶-۴۵: مثالی از عملکرد بلوک Yearly Timer Switch



Up - Down Counter :

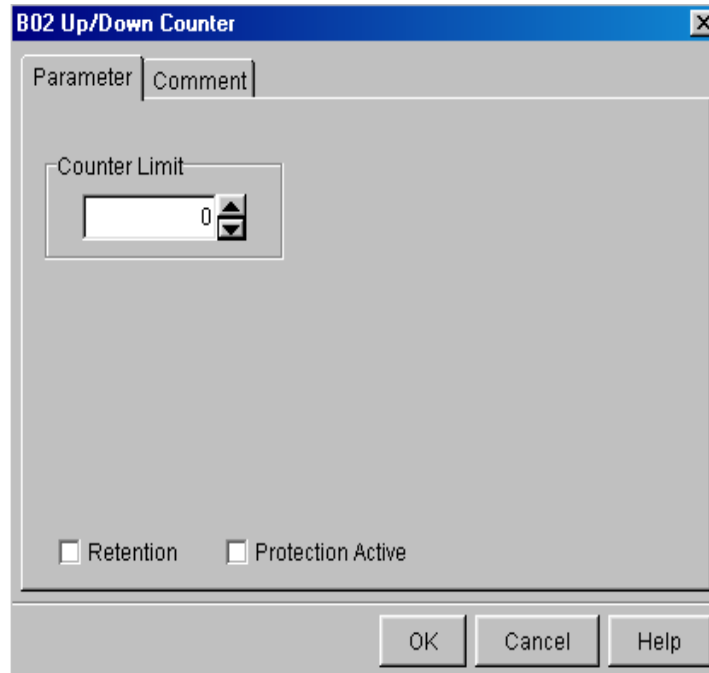
در صورت نیاز به شمارش تعداد یک رخداد می توان از این بلوک استفاده نمود . در این شمارنده که یک شمارنده صعودی / نزولی می باشد ، با اعمال پالس به لبه Cnt و باتعیین جهت شمارش از طریق پایه Dir مقدار شمارش شده داخلی افزایش یا کاهش می یابد . شکل ۶-۴۶ عملکرد این بلوک را نمایش می دهد. در این بلوک با اعمال سطح بالای منطقی به ورودی Dir شمارنده نزولی شده و با اعمال سطح منطقی پایین به این پایه شمارنده به صورت صعودی شمارش می نماید . پایه R جهت صفر نمودن خروجی و همچنین مقدار شمارش شده داخلی استفاده می گردد . در این بلوک در حالت شمارش صعودی ، پس از رسیدن مقدار شمارش شده داخلی به پارامتر مشخص شده مربوط به حد نهایی ، که از طریق تنظیم مقدار آن در برگه پارامتر صورت می پذیرد ، خروجی به سطح بالا می رود . در این حال با اعمال پالسهای دیگر شمارنده داخلی به شمارش ادامه می دهد ولیکن خروجی همچنان در سطح بالا باقی می ماند . حال اگر جهت شمارش را عوض نماییم ، مقدار شمارش داخلی یکی یکی کم شده و به محض رسیدن این مقدار به حد پارامتر خروجی صفر می گردد . شکل ۶-۴۷ صفحه مشخصات این بلوک را نمایش می دهد .



شکل ۶-۴۶: نمودار عملکرد شمارنده صعودی - نزولی

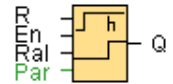




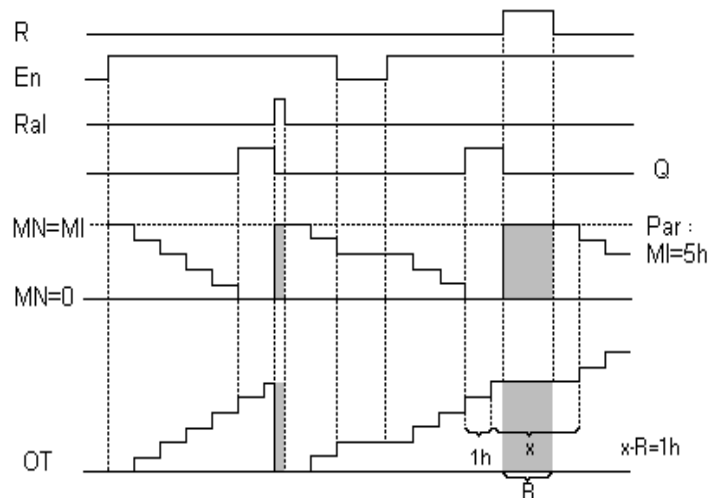


شکل ۶-۴۷: صفحه مشخصات شمارنده صعودی-نزولی

## H ource Counter



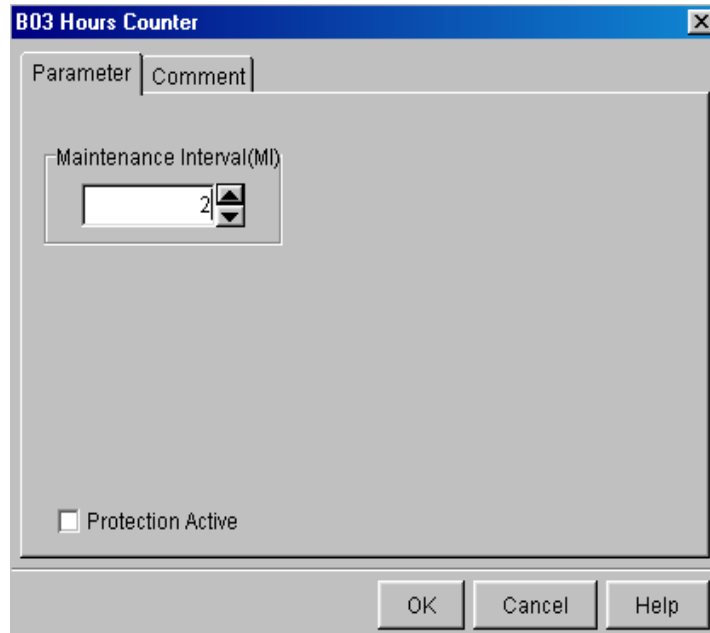
این بلوک یک شمارنده ساعت بوده که نمودار زمانی عملکرد آن در شکل ۶-۴۸ ارائه شده است. در این بلوک پس از طی ساعاتی که در برگه پارامتر توسط کاربر تعیین می گردد، خروجی یک خواهد شد. با توجه به نمودار زمانی عملکرد این بلوک می توان گفت مادامی که ورودی های **R** و **Ral** تحریک نگردیده اند، با فعال شدن پایه **En** زمان شمارش می شود، تا وقتی که زمان سپری شده برابر مدت زمان تعریف شده در برگه پارامتر گردد. با تحریک ورودی **R** تنها خروجی مدار صفر شده ولی زمان شمارش شده تا آن لحظه حفظ خواهد شد. در این حال به محض غیر فعال شدن پایه **R** زمان باقی مانده تا رسیدن به مدت زمان تعریف شده شمارش می گردد. ولیکن با تحریک پایه **Ral** نه تنها خروجی صفر می گردد بلکه مقدار شمارش شده داخلی نیز صفر می گردد. در این حال با غیر فعال شدن پایه **Ral** شمارش از ابتدا آغاز می گردد. نکته قابل توجه آن است که با غیر فعال شده پایه **En** شمارش در آخرین مقدار تا آن لحظه متوقف می گردد.



شکل ۶-۴۸: نمودار زمانی عملکرد H ource Counter

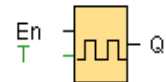


شکل ۶-۴۹ صفحه مشخصات این بلوک را نمایش می دهد. در برگه پارامتر این صفحه می توان ساعات مورد نظر جهت شمارش را تعیین نمود.

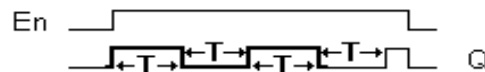


شکل ۶-۴۹: صفحه مشخصات H ource Counter

### : Symmetrical Clock Pulse Generator

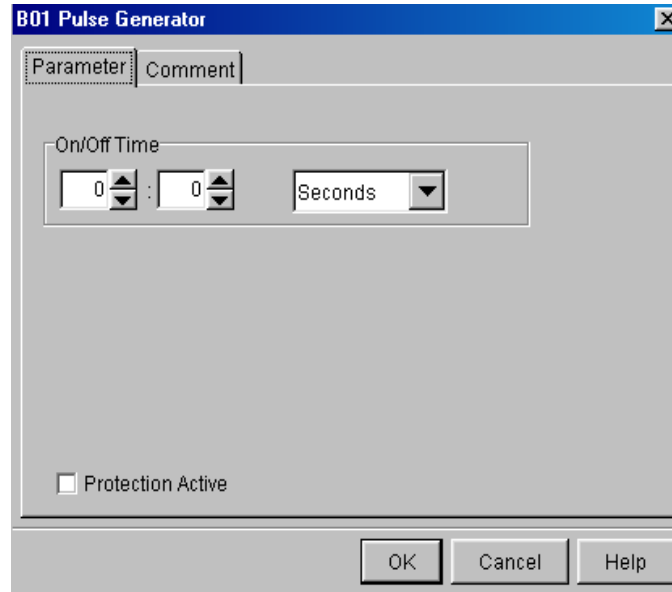


این بلوک بیانگر یک مولد پالس مربعی با دوره تناوب  $2T$  می باشد که زمان  $T$  در آن از طریق تنظیم مشخصات برگه پارامتر قابل تغییر می باشد. شکل ۶-۵۰ نمودار زمانی عملکرد این بلوک را نمایش می دهد. در این بلوک با فعال ماندن پایه  $En$  خروجی با فرکانس مشخص نوسان می نماید. شکل ۶-۵۱ صفحه مشخصات این بلوک را نمایش می دهد.



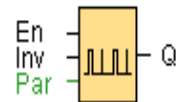
شکل ۶-۵۰: دیاگرام زمانی عملکرد پالس ژنراتور



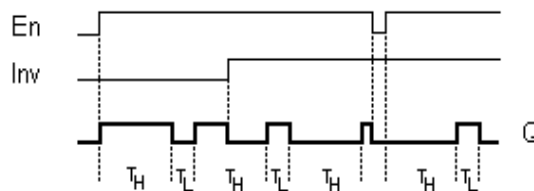


شکل ۶-۵۱: صفحه مشخصات پالس ژنراتور

## : Asynchronous Pulse Generator

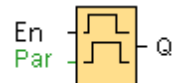


این مولد پالس عملکردی همانند پالس ژنراتور عنوان شده قبلی دارد با این تفاوت که در این بلوک زمانهای بالا و پایین ماندن خروجی می تواند بصورت متفاوت تعیین گردند. شکل ۶-۵۲ نمودار زمانی عملکرد این بلوک را نمایش می دهد. در این بلوک با فعال ماندن ورودی **En** خروجی بطور متناوب با فرکانس مشخص و با **duty cycle** قابل تنظیم پالس می دهد. با فعال ساختن ورودی **Inv**، خروجی عملکردی عکس حالت معمول خواهد داشت. به عبارت دیگر اگر در حالت معمول زمان **TH**، زمان بالا ماندن خروجی را بیان می نماید، با فعال ساختن ورودی **Inv** این زمان، مدت پایین ماندن خروجی را تعیین می نماید. شکل ۶-۵۳ صفحه مشخصات این بلوک را نمایش می دهد. در برگه پارامتر این صفحه می توان زمانهای **TH** و **TL** را به دلخواه تنظیم نمود.



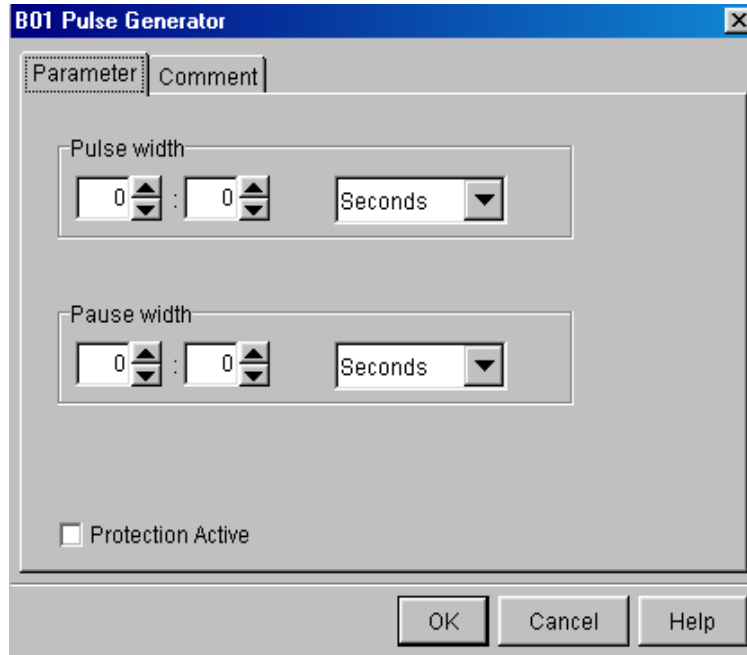
شکل ۶-۵۲: نمودار زمانی عملکرد پالس ژنراتور آسنکرون

## : Random Generator

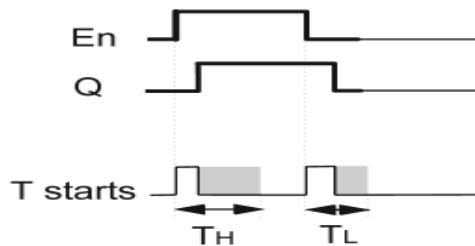


این بلوک یک مولد پالس با زمانهای بالا و پایین تصادفی می باشد. به عبارت دیگر در این مولد پالس زمانهای بالا یا پایین ماندن خروجی یک زمان تصادفی بین صفر و یک زمان حداکثر می باشد، که این زمان حداکثر توسط کاربر و با تغییر زمانهای **TH** و **TL** در برگه پارامتر ایجاد می گردند. شکل ۶-۵۴ نمودار زمانی و شکل ۶-۵۵ صفحه مشخصات این بلوک را نمایش می دهند. از این پالس ژنراتور می توان در طراحی مداراتی که در آنها بحث نمونه گیری تصادفی وجود دارد استفاده نمود.

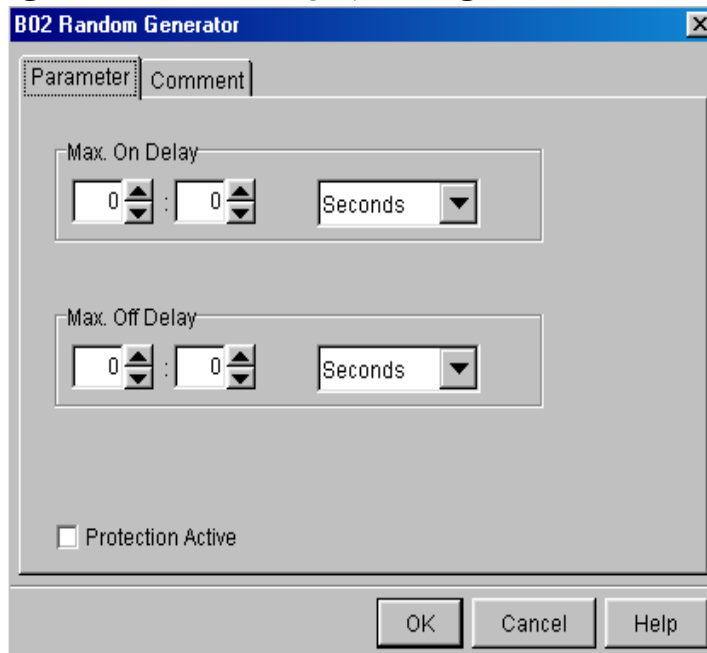




شکل ۶-۵۳: صفحه مشخصات پالس ژنراتور آسنکرون

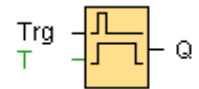


شکل ۶-۵۴: نمودار زمانی عملکرد پالس ژنراتور با زمانهای تصادفی



شکل ۶-۵۵: صفحه مشخصات پالس ژنراتور با زمانهای تصادفی

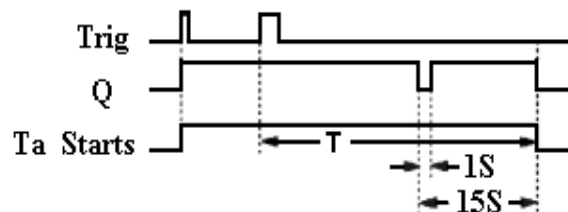




## : S Airway Light Switch

در این بلوک که با عنوان کلید روشنایی راه پله مورد استفاده قرار می گیرد ، با اعمال یک لبه بالارونده در ورودی تریگر خروجی به مدت زمان T که در برگه پارامتر آن تعریف می گردد به سطح بالا رفته و سپس به سطح پایین بر می گردد . تکرار تریگر مدار باعث اندازه گیری مجدد زمان T می گردد .

از مشخصات منحصر بفرد این بلوک وجود یک زمان اعلام خطر می باشد که در فاصله زمانی بمدت این زمان خطر از اتمام زمان T ، خروجی بمدت مشخصی به سطح پایین رفته و مجدداً تا انتهای زمان T در سطح بالا قرار می گیرد . بعبارت دیگر در زمان مشخصی از اتمام زمان T یک اعلام خطر با صفر شدن خروجی اعلام می گردد . شکل ۶-۵۶ نمودار زمانی عملکرد این بلوک را نمایش می دهد . جدول ارائه شده در شکل ۶-۵۷ زمان اعلام خطر را برای بازه های زمانی موردنظر بیان می کند . به عنوان مثال با انتخاب بازه زمانی دقیقه زمان اعلام خطر ۱۵ ثانیه قبل از اتمام زمان T می باشد که در این حال خروجی بمدت یک ثانیه به سطح پایین می رود .



شکل ۶-۵۶ : نمودار زمانی عملکرد سوئیچ روشنایی راه پله

Time Base T	Warning Time	Warning Duration
Seconds	750 ms	50 ms
Minutes	15 s	1 s
Hourse	15 min	1 min

شکل ۶-۵۷ : جدول بیان زمانهای اعلام خطر در بازه های مختلف زمانی

## : Multifunctional Switch



این عملکرد یک سوئیچ با دو کاربرد متفاوت می باشد .

- کلید پالسی با تایمر تاخیر در قطع
- کلید ( با وصل دائمی )

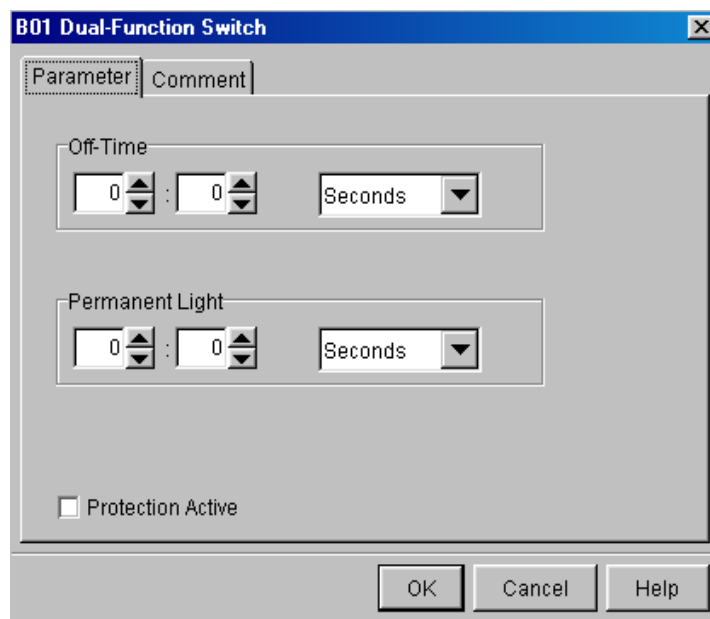
شکل ۶-۵۸ نمودار زمانی عملکرد این عملکرد را نمایش می دهد . با دقت در این نمودار می توان گفت با اعمال یک لبه مثبت به ورودی تریگر ، خروجی فعال شده و در صورتیکه قبل از اتمام زمان TH که در برگه مشخصات آن تعریف می گردد ، تحریک دیگری در ورودی تریگر بوقوع نپیوندد ، خروجی بعد از اتمام مدت زمان TH به سطح صفر بر می گردد . در صورت اعمال تریگر مجدد قبل از اتمام مدت زمان TH ، خروجی بلافاصله به سطح صفر باز می گردد . از نقاط قابل توجه در این عملکرد قابلیت تعیین مدت زمان مورد نظر جهت پالس تریگر می باشد .



به عبارت دیگر با تعریف مدت زمان TL برای این عملگر در صورتی که مدت زمان یک بودن پالس تریگر از این مقدار بیشتر گردد، حالت دوم عملکرد این عملگر بوقوع می پیوندد، یعنی خروجی به سطح بالا رفته و تا زمان اعمال پالس مجدد به ورودی تریگر در سطح بالا خواهد ماند. شکل ۶-۵۹ صفحه مشخصات این عملگر را نمایش می دهد.



شکل ۶-۵۸: نمودار زمانی عملکرد M multifunctional Switch



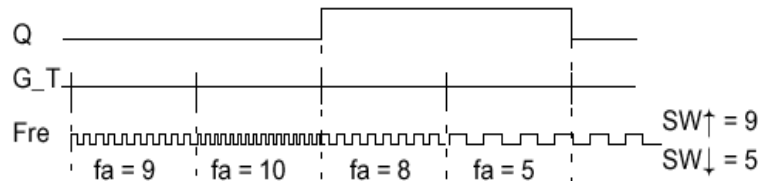
شکل ۶-۵۹: صفحه مشخصات M multifunctional Switch

↑ SW : Frequency Trigger

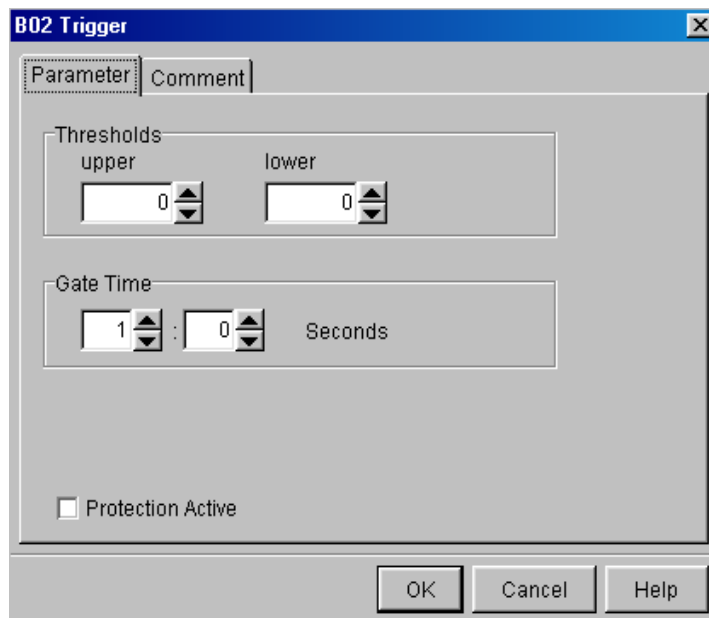
در این بلوک با اعمال یک سیگنال به ورودی Fre و با تعیین دو سطح فرکانسی در صفحه مشخصات می توان با توجه به فرکانس سیگنال ورودی و دو فرکانس مرجع، خروجی را یک یا صفر نمود. در این بلوک پارامترهای SW ↑ و SW ↓ به ترتیب حد آستانه روشن شدن و حد آستانه خاموش شدن را بیان می نمایند. زمان Gate-time نیز مدت زمانی را که تعداد پالسهای ورودی جهت محاسبه فرکانس آن شمارش می گردد را بیان می نماید. شکل ۶-۶۰ نمودار زمانی عملکرد این بلوک را برای دو فرکانس نمونه مرجع ۵ و ۹ نمایش می دهد. با توجه به این شکل مشخص است که هرگاه فرکانس ورودی مساوی حد آستانه روشن شدن و یا بیشتر از آن گردد، خروجی پس از طی مدت زمان G-T یک می گردد. همچنین هرگاه فرکانس ورودی برابر و یا کوچکتر از حد آستانه خاموش شدن گردد، خروجی صفر می گردد.



جهت ورودی این بلوک ، ورودی های I5/I6 ( تنها در نمونه های 12/24 RC/RCO و LOGO!24 ) و ورودی های I11/I12 در نمونه های ( LOGO!...L ) برای شمارش های سریع حداکثر تا یک کیلوهرتز با سطح ولتاژ ۲۴ ولت استفاده می گردند . سایر ورودی ها و اجزاء مدار برای شمارش فرکانس های پایین بکار می روند . شکل ۶-۶۱ صفحه مشخصات این بلوک را نمایش می دهد .



شکل ۶-۶۰: نمودار زمانی عملکرد بلوک Frequency Trigger



شکل ۶-۶۱: صفحه مشخصات بلوک Frequency Trigger

: A nalog Trigger 

در این بلوک مقدار خروجی با تغییر دامنه ورودی آنالوگ و تجاوز آن از حدود مرجع قابل تعریف تغییر می نماید . شکل ۶-۶۲ نمودار زمانی عملکرد این بلوک را نمایش می دهد . با توجه به این نمودار می توان گفت با رسیدن دامنه سیگنال ورودی به حد آستانه بالا که با  $SW \uparrow$  نمایش داده می شود خروجی یک شده و با کاهش دامنه ورودی و رسیدن به حد آستانه پایین  $SW \downarrow$  خروجی به سطح صفر باز می گردد . در حد فاصل این دو مقدار خروجی مقدار قبلی خود را حفظ می نماید . جهت ورودی این بلوک می توان یکی از ورودی های آنالوگ I7/I8 را در نظر گرفت .

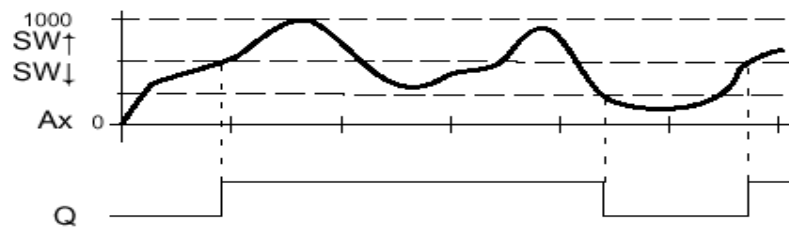


در این بلوک ورودی های 0-10 v و یا 0 - 20 mA مطابق مقدار داخلی 0 - 1000 در نظر گرفته می شوند .  
صفحه مشخصات این بلوک در شکل ۶-۶ ارائه شده است . سایر پارامترهای این صفحه عبارتند از :

- Gain بر حسب درصد ، که مقادیر مابین صفر تا ۱۰۰۰ درصد را شامل می گردد .
- Offset که مقادیر مابین +999 تا -999 را شامل می گردد .
- حداقل و حداکثر رنج اندازه گیری دامنه ورودی

با توجه به این پارامترها ، مقدار اندازه گیری شده از رابطه  $Value = ( AI + Offset ) * Gain$  محاسبه می گردد .

به عنوان یک مثال اگر حد بالا ۴۸۵ ، حد پایین ۸۶ ،  $Gain = 100\%$  ، و  $Offset = 0$  در نظر گرفته شوند ، با اعمال ولتاژ ۴/۸۵ ولت به ورودی آنالوگ که معادل مقدار  $4.85 * 100 = 485$  می گردد خروجی یک شده و با اعمال ولتاژ ۰/۸۶ ولت به ورودی آنالوگ خروجی صفر می گردد .



شکل ۶-۶: نمودار زمانی عملکرد Analog Trigger

شکل ۶-۶: صفحه مشخصات بلوک Analog Trigger



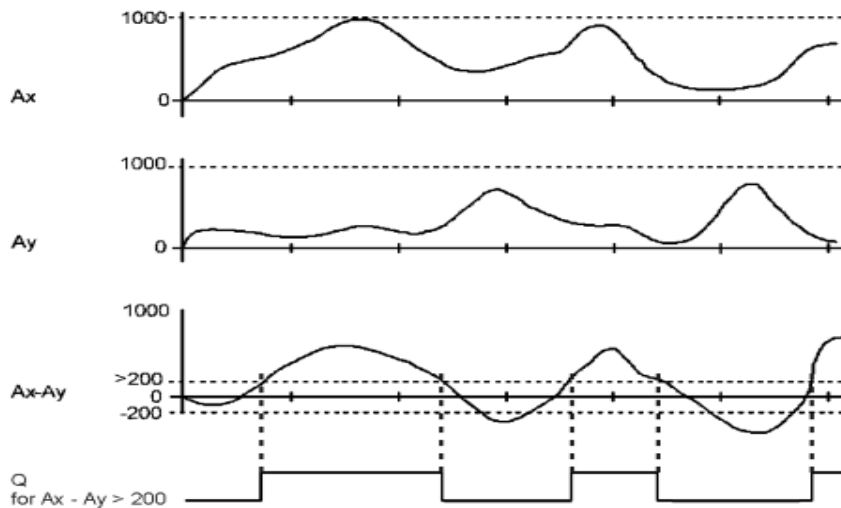


Analog Comparator 

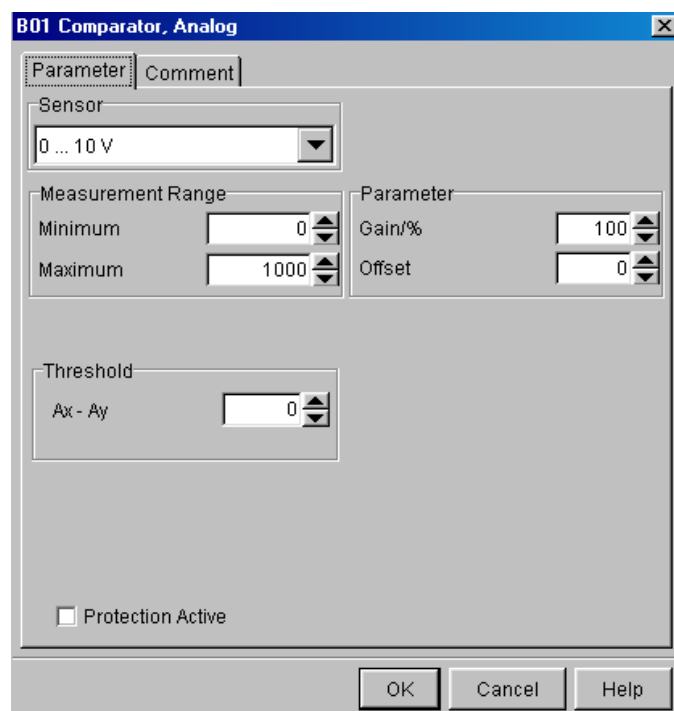
این بلوک یک مقایسه کننده آنالوگ می باشد . با توجه به شکل ۶-۶۴ که نمودار عملکرد این بلوک را نمایش می دهد ، می توان گفت در این بلوک هرگاه اختلاف دو ورودی آنالوگ به حد آستانه قابل تعریف برای آن برسد ، خروجی یک خواهد شد. ورودی های این بلوک می بایست به ورودی های آنالوگ AI1/AI2 متصل گردند . در برگه مشخصات این بلوک که در شکل ۳-۶۵ نیز نمایش داده شده است پارامترهای Gain ، Offset و Threshold Value وجود دارد که کاربر می تواند با توجه به نیاز ، آنها را به دلخواه تغییر دهد . در این بلوک خروجی زمانی یک خواهد شد که رابطه ریاضی زیر برقرار گردد .

$$[( Ax + Offset ) \times Gain ] - [ ( Ax + Offset ) \times Gain ] > Threshold Value$$

پارامتر گین که با علامت  $\nabla$  نمایش داده می شود بر حسب درصد بوده و مقادیری بین صفر تا ۱۰۰۰ را شامل می گردد . پارامتر Offset نیز که با علامت  $\uparrow$  مشخص می گردد می تواند مقادیر مابین ۹۹۹- تا ۹۹۹+ را بپذیرد .

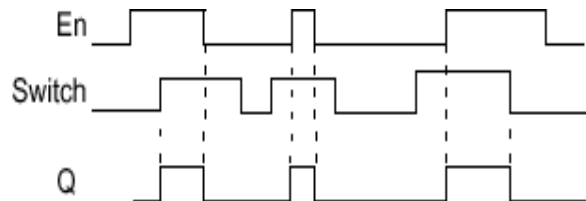


شکل ۶-۶۴: دیاگرام عملکرد مقایسه کننده آنالوگ

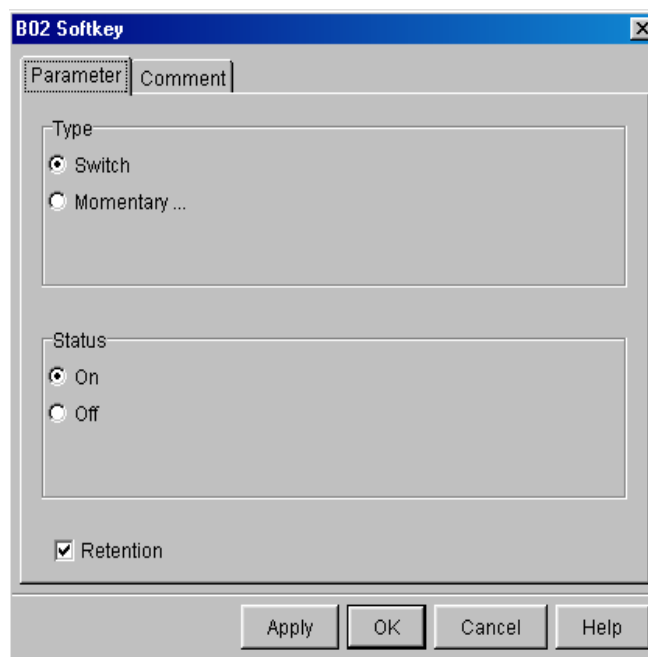


## S oft Key

این بلوک به مانند یک کلید عمل می نماید . در صورت وقوع یک انتقال سطح سیگنال از صفر به یک در ورودی En خروجی در صورت فعال نمودن حالت On در صفحه مشخصات ، یک خواهد شد . شکل ۶۶-۶ نمودار زمانی عملکرد این بلوک را نمایش می دهد . مشاهده می شود در صورت عدم انتخاب گزینه On در صفحه مشخصات و یا غیرفعال شدن آن خروجی به سطح صفر باز خواهد گشت . صفحه مشخصات این بلوک در شکل ۶۷-۶ ارائه شده است .

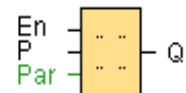


شکل ۶۶-۶ : نمودار زمانی عملکرد بلوک S oftKey



شکل ۶۷-۶ : صفحه مشخصات بلوک S oft Key

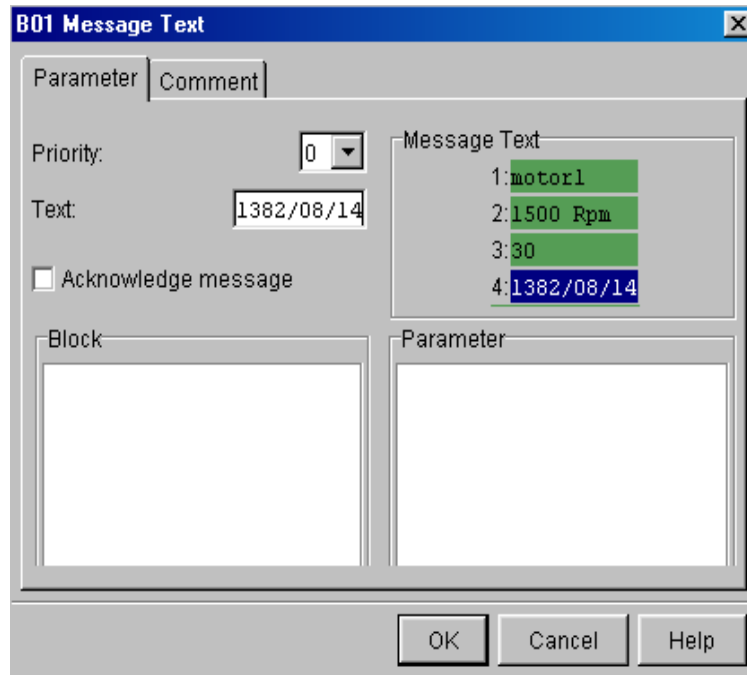
: message Texts



این بلوک جهت تعیین پیام های مورد نیاز جهت نمایش بر روی صفحه نمایشگر LOGO مورد استفاده قرار می گیرد . حداکثر ۵ عملگر نمایش متن پیام در دسترس می باشد . ایجاد یک تغییر وضعیت از صفر به یک در ورودی En باعث نمایش متن پیام بر روی صفحه نمایشگر خواهد شد . صفحه مشخصات این بلوک در شکل ۶۸-۶ ارائه شده است .



در این صفحه گزینه ای به نام Acknowledgment Message وجود دارد که در صورت فعال بودن آن با صفر شدن ورودی صفحه نمایش پیام تا زمانی که از طرف کاربر تصدیق نگردد پاک نخواهد شد. کاربر با انتخاب کلید Ok می تواند متن را رسید نماید. نکته قابل توجه آنکه در صورت یک بودن ورودی نمی توان متن را رسید نمود. می توان ۹ صفحه نمایش با اولویت های یک تا نه را برای نمایش اطلاعات تعریف نمود. با فعال شدن ورودی صفحه نمایش متن با اولویت بالاتر نمایش داده می شود که کاربر می تواند با استفاده از کلید های جهتی سایر برگه ها را نیز مشاهده نماید.



شکل ۶-۶۸: صفحه مشخصات بلوک نمایش پیام

### طراحی مدارات با استفاده از نرم افزار و تست آنها

پس از طرح مقدماتی مدارات فرمان موردنظر با استفاده از روش های طراحی دیجیتال و یا روش های موجود جهت طراحی مدارات رله کنتاکتوری، می بایست مدار مورد نظر را با استفاده از نرم افزار LOGO طراحی و تست نموده و پس از رفع معایب احتمالی آنرا به PLC منتقل نماییم. در این قسمت کل مراحل لازم جهت طراحی، تست و انتقال یک برنامه نمونه را شرح داده و در فصل هفتم مثالهای متنوعی در زمینه طراحی مدارات با استفاده از نرم افزار ارائه خواهد گشت. نهایتاً در ضمیمه ۶ این کتاب روش طراحی و برنامه نویسی با استفاده از صفحه کلید موجود بر روی LOGO ارائه خواهد شد.

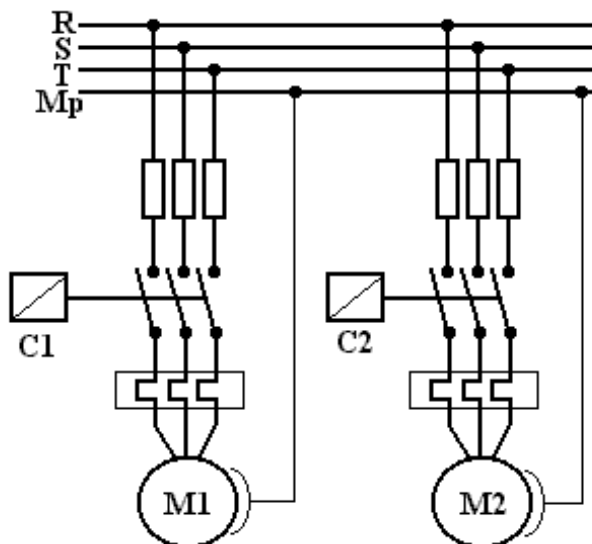
مثال: راه اندازی دو الکتروموتور سه فاز بصورت یکی پس از دیگری

هدف طراحی مداری است که با استفاده از دوکلید بتوان دو موتور را به شرح ذیل کنترل نمود:

- ۱- با زدن شستی S1، موتور M1 روشن گردد.
- ۲- با زدن شستی S2، موتور M2 به شرط روشن بودن موتور M1 روشن گردد.
- ۳- در غیر اینصورت دو موتور خاموش باشند.

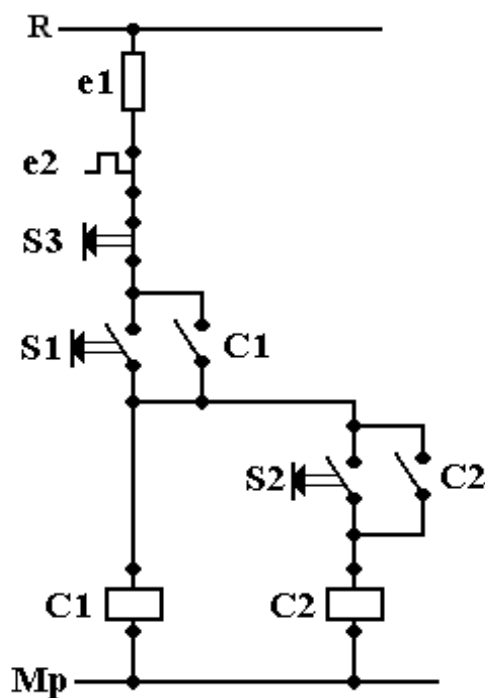


مدار قدرت مربوط به این مدار در شکل ۶-۶۹ ارائه شده است. در این مدار سه فاز R، S، T با عبور از عناصر حفاظتی مورد نیاز و از طریق کنتاکت های قدرت مورد نظر به موتور وارد می گردند.



شکل ۶-۶۹: مدار قدرت مربوط به راه اندازی دو موتور بصورت یکی پس از دیگری

با توجه به مدار فرمان ارائه شده برای این مدار در شکل ۶-۷۰ می توان گفت، با زدن شستی S1 کنتاکتور C1 جذب کرده و در نتیجه موتور M1 روشن می گردد، در این حال کنتاکت نگهدارنده C1 نیز بسته می شود. حال با زدن شستی S2، کنتاکتور C2 نیز جذب شده و در نتیجه موتور M2 را روشن می نماید. در این مدار موتور M2 تنها زمانی اجازه روشن شدن دارد که موتور M1 روشن بوده و کنتاکت C1 بسته باشد.



شکل ۶-۶۹: مدار فرمان مربوط به راه اندازی دو موتور بصورت یکی پس از دیگری



چون این مدار یک مدار دیجیتال ترکیبی است ، لذا جهت طراحی این مدار فرمان با استفاده از قطعات دیجیتال می توان ابتدا جدول عملکرد آنرا بدست آورده و سپس با بکارگیری آن مدار را طراحی نمود .  
جدول عملکرد مربوط به مدار یکی پس از دیگری در شکل ۶-۷۱ ارائه شده است . چون دو کلید S1 و S2 بعنوان ورودی های مدار بوده و کنتاکتور های C1 و C2 نیز خروجی های مدار محسوب می شوند ، لذا جدول عملکرد مورد نظر دارای ۲ ورودی و ۲ خروجی می باشد . ۲ ورودی مذکور دارای ۴ حالت مختلف نسبت به هم می باشند که این حالات را در جدول عملکرد در نظر می گیریم . حال می بایست وضعیت خروجی ها را با توجه به وضعیت فعال و یا غیر فعال بودن ورودی ها بررسی نماییم . در مدار مورد نظر در صورتی که شستی S1 فشرده نشده باشد هیچیک از کنتاکتور ها نمی توانند جذب گردند لذا در جدول به ازاء  $S1=0$  داریم  $C1=C2=0$  . در حالتی که S1 فشرده شده باشد و S2 زده نشده باشد تنها کنتاکتور C1 جذب می گردد ، لذا در جدول به ازاء  $S1=1$  و S2 = 0 داریم  $C1=1$  و  $C2=0$  . در حالت آخر هم که دو شستی فشرده می شوند می بایست دو کنتاکتور تحریک گردند .

S1	S2	C1	C2
0	0	0	0
0	1	0	0
1	0	1	0
1	1	1	1

شکل ۶-۷۱: جدول عملکرد مربوط به مدار یکی پس از دیگری

با توجه به جدول عملکرد بدست آمده می توان عبارت منطقی معادل هر یک از خروجی ها را به ازاء ورودی های مدار بدست آورد .

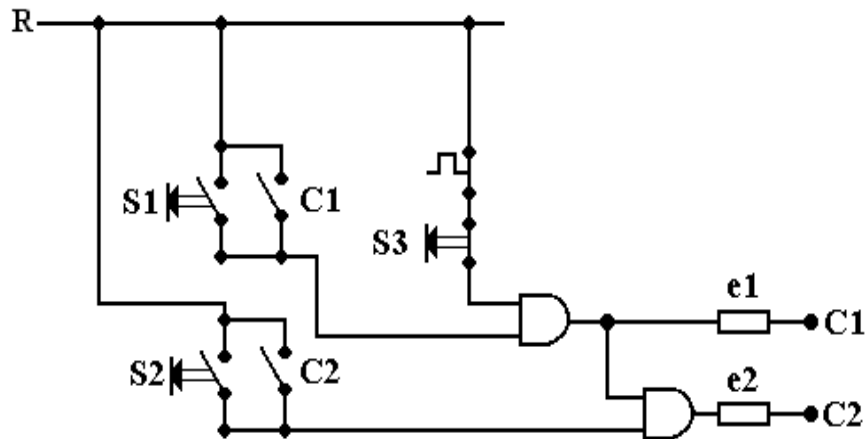
$$C1 = S1 S2' + S1 S2 \longrightarrow C1 = S1 (S2 + S2') = S1 (1) = S1$$

توجه : حاصل عمل OR منطقی یک متغیر منطقی و مکمل آن برابر سطح یک منطقی می باشد زیرا در هر حال یکی از ورودی های گیت OR برابر یک است .

$$C2 = S1 S2$$

از آنجا که بی متال و شستی استپ ( S3 ) بصورت سری با هم بوده و بصورت بسته می باشند ، لذا در حالت عادی می بایست جریان را عبور داده و در حال فعال شدن جریان را قطع نمایند لذا آنها را با هم سری کرده و به یک پایه گیت AND متصل می نماییم ، پایه دیگر گیت AND ورودی S1 را شامل می گردد . حال اگر به هر دلیل یکی از ورودی های سری شده فعال گردند در ورودی گیت AND سطح صفر منطقی بوجود می آید لذا ورودی S1 نمی تواند عمل نماید ( زیرا در این صورت خروجی گیت AND همواره صفر می باشد ) و در نتیجه چون اثر S1 برابر سطح منطقی صفر می گردد در این حالت مدار عملکردی مشابه زمان  $S1=0$  دارد و هر دو کنتاکتور قطع می باشند . بدین ترتیب توانستیم اثر قطعات حفاظتی و همچنین شستی Stop را در مدار اعمال نماییم .  
شکل ۶-۷۲ مدار دیجیتال معادل را نمایش می دهد .

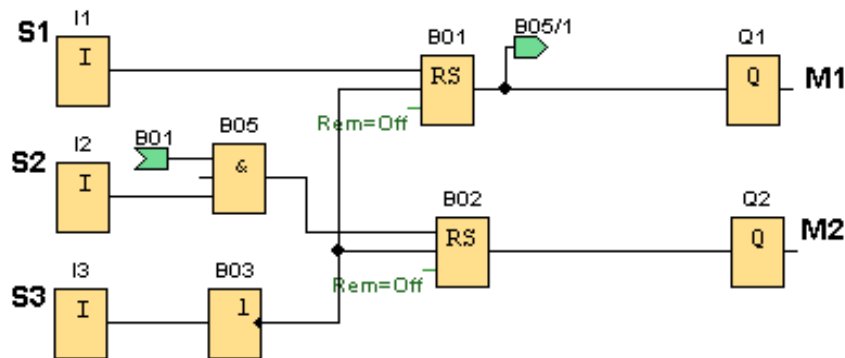




شکل ۶-۷۲: مدار فرمان مربوط به راه اندازی دو الکتروموتور سه فاز به صورت یکی پس از دیگری با استفاده از گیت های دیجیتال

حال زمان آن رسیده است که مدار مورد نظر را با استفاده از زبان برنامه نویسی LOGO نوشته و پس از تست آن ، برنامه مذکور را به LOGO انتقال دهیم .

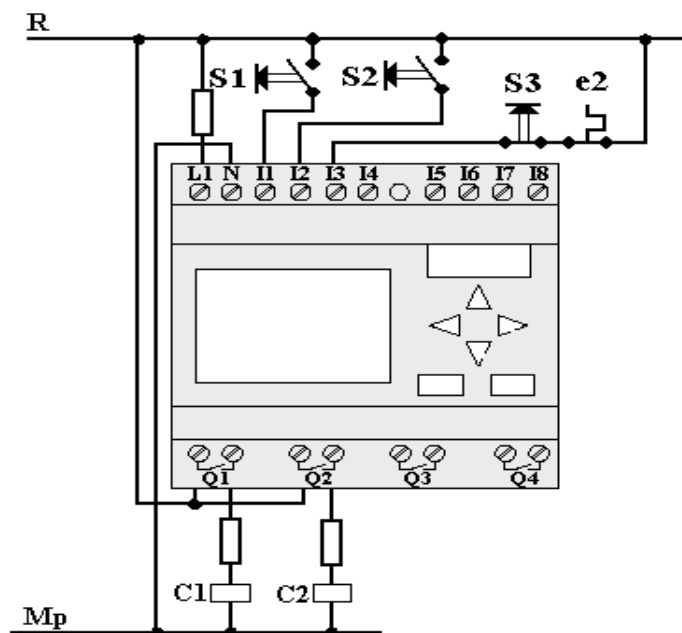
مدار مورد نظر در شکل ۶-۷۳ ارائه شده است . در این مدار سه ورودی I1 تا I3 به منزله شستی های S1 و S2 و همچنین مدار سری بی متال و شستی استپ در نظر گرفته شده اند . چون S1 و S2 بصورت شستی می باشند ، لذا ورودی های I1 و I2 را بصورت Momentary تعریف می کنیم . چون در مدار مورد نظر می بایست با تحریک لحظه ای شستی ها مدار فعال گردد ، لذا از فلیپ فلاپ RS جهت فعال نگه داشتن خروجی ها در زمان قطع شستی استفاده می گردد . به عبارت دیگر فلیپ فلاپ RS بجای کنتاکت نگهدارنده در مدار استفاده می گردد . از آنجا که در فلیپ فلاپ RS ورودی R نسبت به S تقدم دارد ، و همچنین بعلت اینکه ورودی I3 در حالت عادی فعال است ( زمانی که شستی استپ فشرده نشده و بی متال عمل نکرده است ) . لذا این ورودی را از طریق یک گیت NOT به پایه های Reset فلیپ فلاپها اعمال می نماییم تا در حالت عادی ورودی R غیر فعال بوده و در زمان عمل نمودن هر یک از عناصر سری ورودی R فعال گردد . در این مدار با فشردن شستی S1 ، پایه Set فلیپ فلاپ B01 تحریک شده که در صورت غیر فعال بودن پایه Rset (غیر فعال بودن عناصر سری بی متال و شستی استپ ) ، خروجی Q1 تحریک می گردد . با فعال بودن Q1 پایه اول گیت AND ( بلوک B05 ) یک بوده ، که در اینصورت اگر S2 فشرده شود Q2 نیز روشن می گردد .



شکل ۶-۷۳: مدار فرمان مربوط به راه اندازی دو الکتروموتور سه فاز به صورت یکی پس از دیگری با استفاده از نرم افزار LOGO



شکل ۶-۷۴ چگونگی اتصالات سخت افزاری شستی های مورد نظر به LOGO را نمایش می دهد.



شکل ۶-۷۴: نحوه اتصال شستی ها به LOGO در مدار یکی پس از دیگری



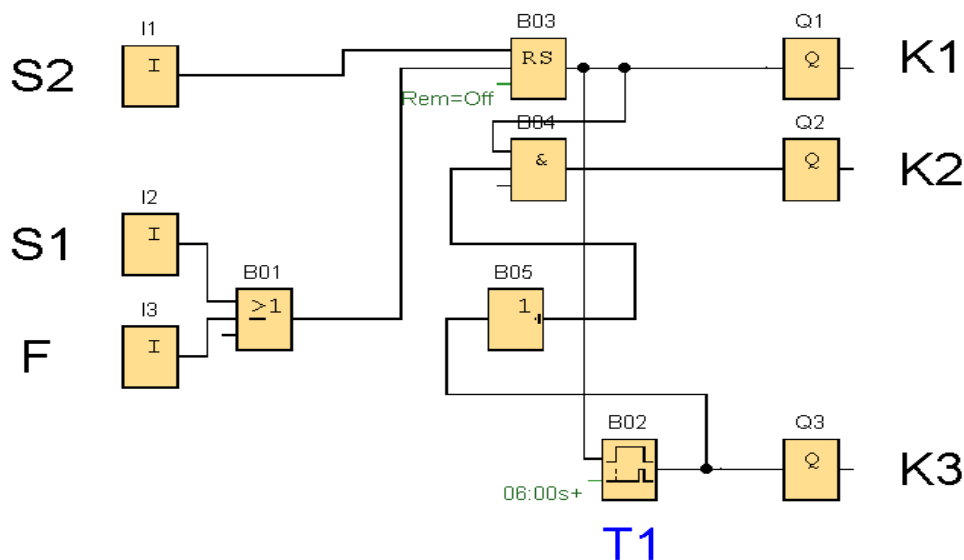
## فصل هفتم

## مدارات کاربردی نمونه با LOGO

همانطور که در مباحث قبل اشاره گردید ، LOGO جهت اتومات سازی سیستم های کوچک صنعتی ، تابلوهای محلی برق ، کنترل دستگاههای منفرد صنعتی و همچنین کاربردهای کنترلی غیر صنعتی کاربرد داشته و امروزه استفاده از آن به همراه سایر میکرو PLC های موجود گسترش فراوانی یافته است . بدین منظور و جهت آشنایی بیشتر خوانندگان محترم در این فصل تعدادی از مدارات کاربردی نمونه که نرم افزار کنترلی آن با این میکرو PLC نوشته شده است ارائه می گردد .

## ۱- مدار راه اندازی یک الکتروموتور سه فاز بصورت ستاره - مثلث اتومات ( شکل ۷-۱ )

- S2 : شستی استارت
- S1 : شستی استپ
- F : فیوز و المانهای سری حفاظتی
- K1 : کنتاکتور ثابت
- K2 : کنتاکتور ستاره
- K3 : کنتاکتور مثلث



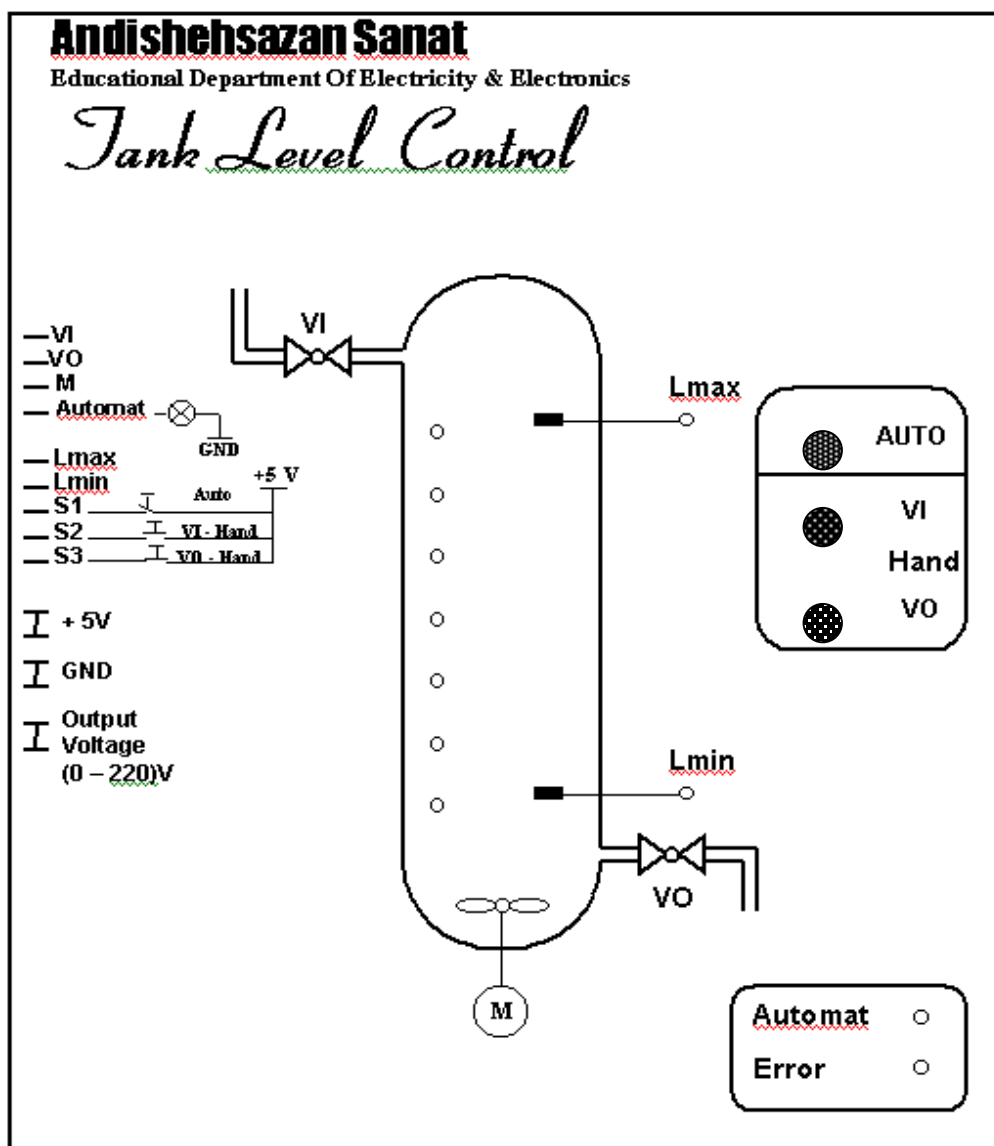
شکل ۷-۱: نرم افزار مربوط به راه اندازی الکتروموتور سه فاز به ستاره - مثلث اتومات



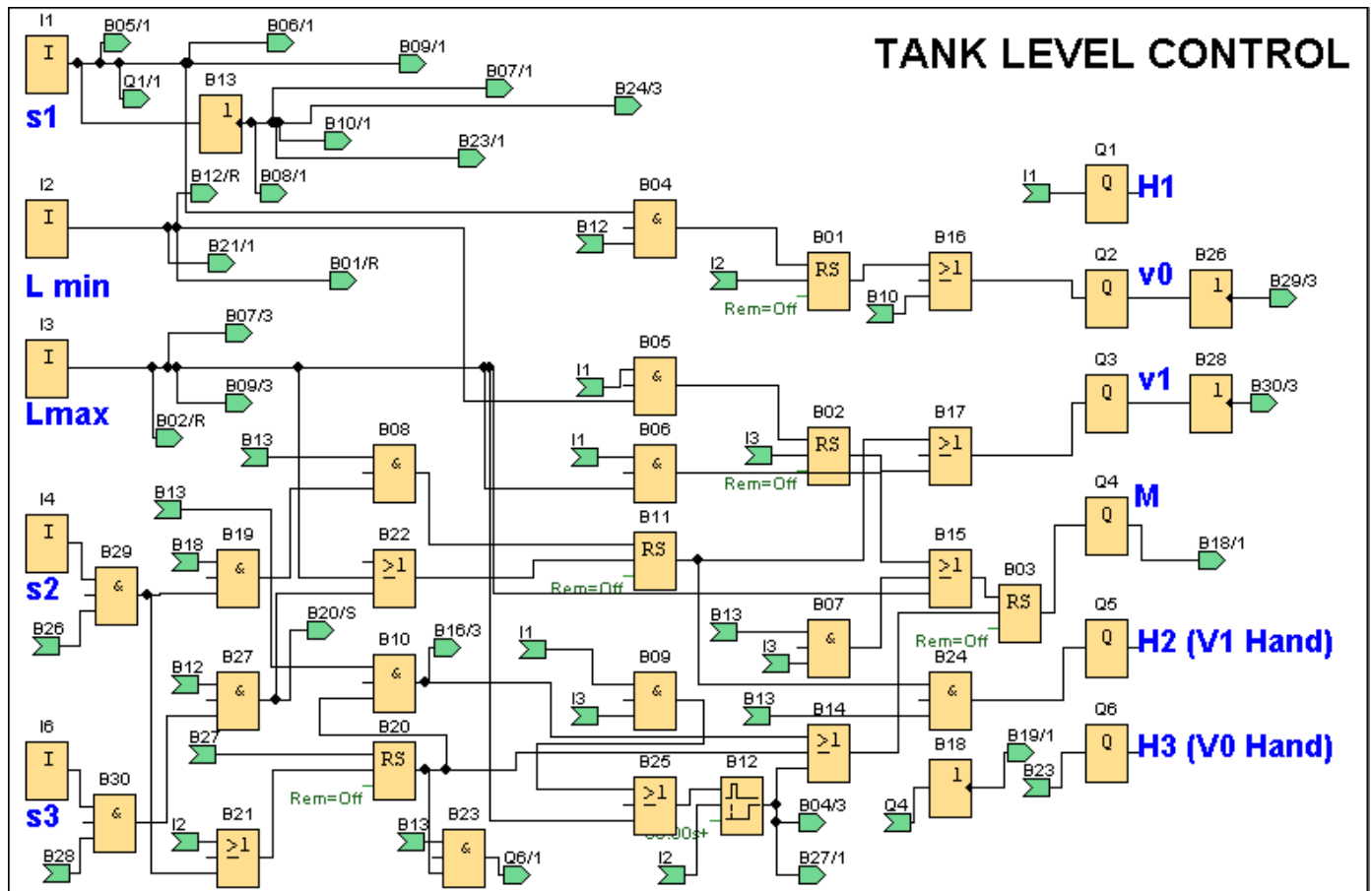


## ۲- سیستم کنترل سطح مایعات

در بسیاری از پروسه های صنعتی عملکرد اتوماتیک و دستی سیستم تحت کنترل همراه با نیازمندیهای اساسی پروسه باید در برنامه کنترلی در نظر گرفته شوند. سیمولاتور کنترل سطح مایع (نمایش داده شده در شکل ۷-۲) ضمن سادگی جنبه های مختلفی از این ملاحظات را در نظر می گیرد. در این سیمولاتور سنسورهای  $L_{max}$  و  $L_{min}$  نمایانگر سطح مایع در داخل مخزن می باشند. هدف از طراحی یک سیستم کنترل برای این سیمولاتور کنترل آن در دو حالت دستی و اتومات می باشد. در هر دو حالت کنترل دستی ضروری است خطای حاصل از عملکرد غلط اپراتور توسط کنترل کننده (PLC) مهار گردد. در این سیستم با فعال نمودن شیر ورودی VI مایع به داخل مخزن وارد می گردد. این عمل تا رسیدن سطح مایع به حد ماکزیمم ادامه داشته و پس از آن با قطع شدن شیر ورودی، موتور بمدتی مشخص همزن را فعال نموده و مایع درون مخزن مخلوط می گردد. پس از اتمام عملیات فوق شیر خروجی جهت تخلیه مایع باز می گردد. این پروسه تا زمان دلخواه ادامه می یابد. شکل ۷-۳ برنامه کنترل نوشته برای این سیمولاتور را نمایش می دهد.



شکل ۷-۲: سیمولاتور کنترل سطح مایع

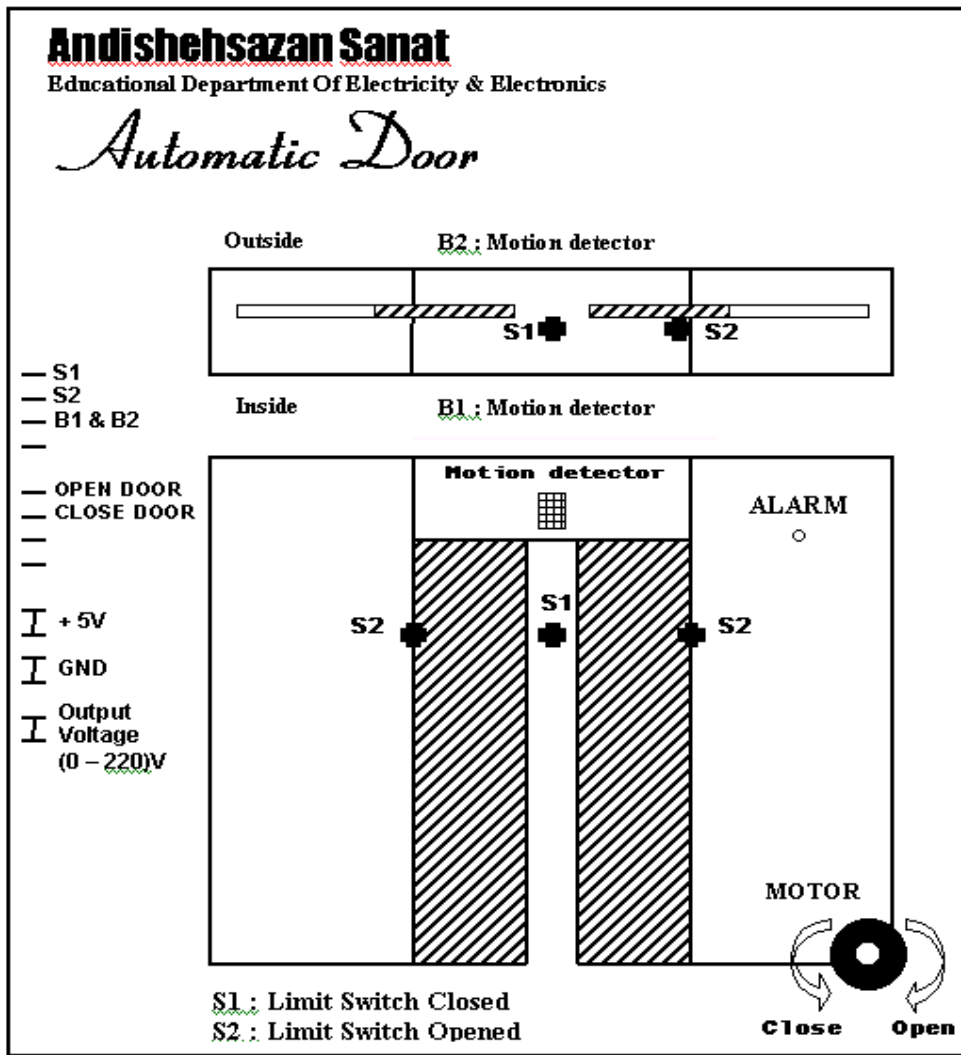


شکل ۷-۳: نرم افزار مربوط به سیستم کنترل سطح مایع با استفاده از LOGO

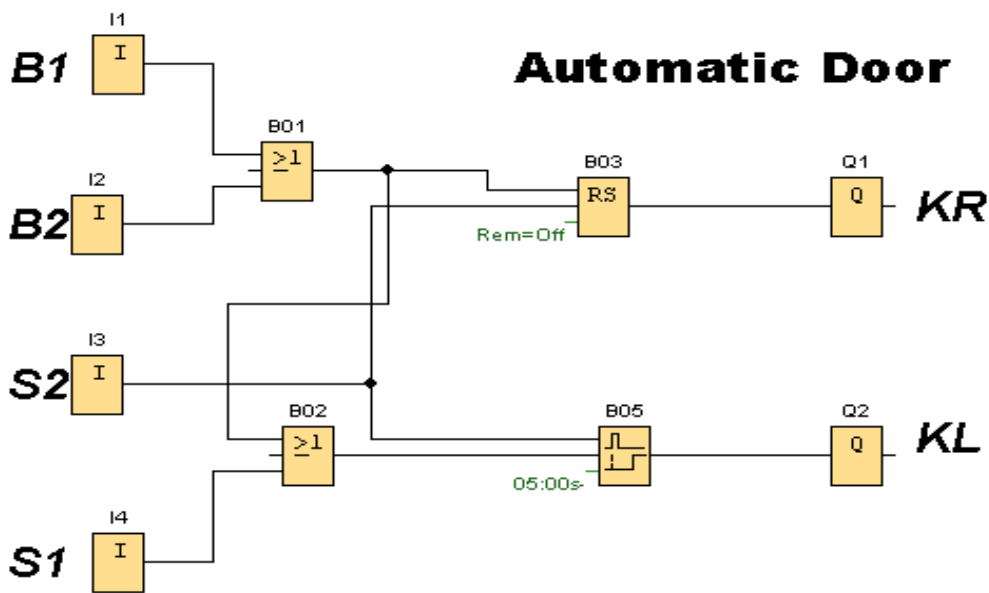
### ۳- سیستم کنترل درب اتومات

سیمولاتور ارائه شده در شکل ۷-۴ را به عنوان شبیه ساز یک درب اتومات در نظر بگیرید. این نمونه از درب های اتومات بعنوان درب ورودی هتل ها، سالن های انتظار و ... مورد استفاده قرار می گیرند. در این سیستم سنسورهای B1 و B2 که نوعی سنسور حساس به حرکت می باشند، با مشاهده نزدیک شدن افراد به سیستم کنترل کننده فرمان داده و در نهایت کنترل کننده با روشن نمودن موتور و حرکت آن در جهت مشخص باعث باز گردیدن درب می گردد. روشن بودن موتور می بایست تا زمانی ادامه پذیرد که میکروسوییچ S2 رسیدن درب به انتهای مسیر را مشخص نماید. در صورت خاموش نشدن موتور امکان آسیب دیدن قسمت های مکانیکی درب وجود دارد. پس از باز شدن کامل درب در صورتی که سنسورهای موجود پیغامی به کنترل کننده ارسال ندارند، کنترل کننده درب را پس از مدت زمان استراحت مشخصی می بندد. در طی زمان بسته شدن درب، اگر سنسورهای B1 و B2، مجدداً حضور فردی را تشخیص دهند می بایست جهت حرکت موتور معکوس شده و درب را باز نماید. نرم افزار مربوط به سیستم کنترل این درب با استفاده از LOGO در شکل ۷-۵ ارائه گردیده است.





شکل ۷-۴: سیمولاتور درب اتومات

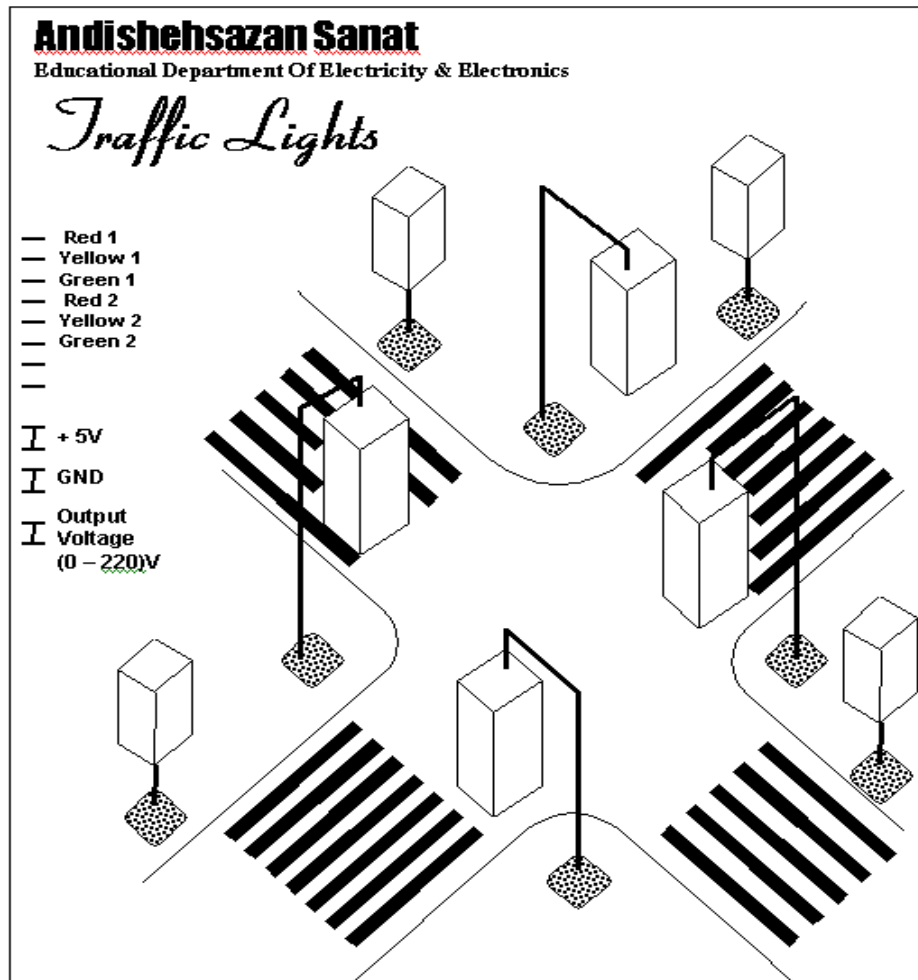


شکل ۷-۵: نرم افزار مربوط به سیستم کنترل سطح مایع با استفاده از LOGO



#### ۴- سیستم کنترل ترافیک

سیمولاتور ارائه شده در شکل ۶-۷ شبیه سازی سیستم کنترل ترافیک دو زمانه می باشد . شکل ۷-۷ نرم افزار مربوطه را نمایش می دهد .

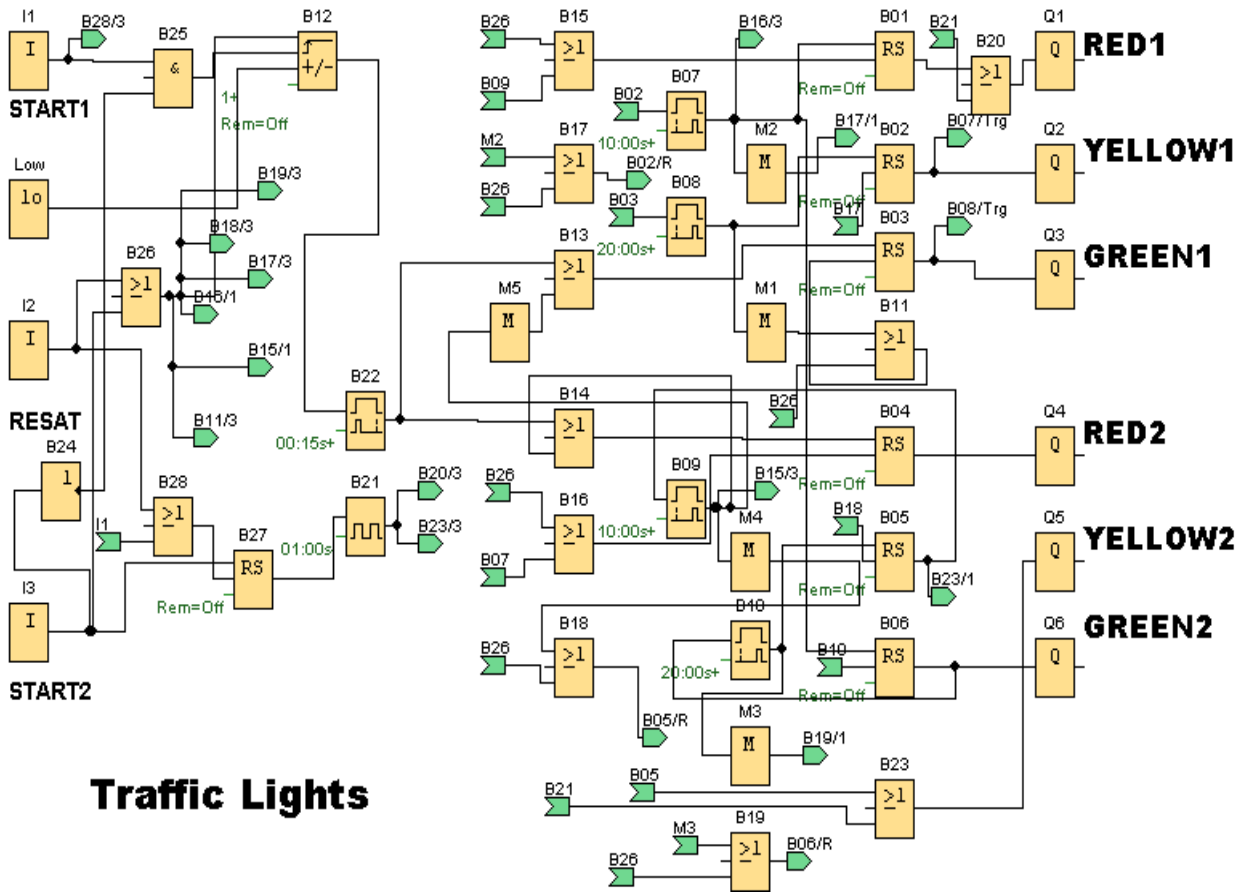


شکل ۶-۷: سیمولاتور کنترل ترافیک

#### ۵- سیستم کنترل میزان تفت فشار

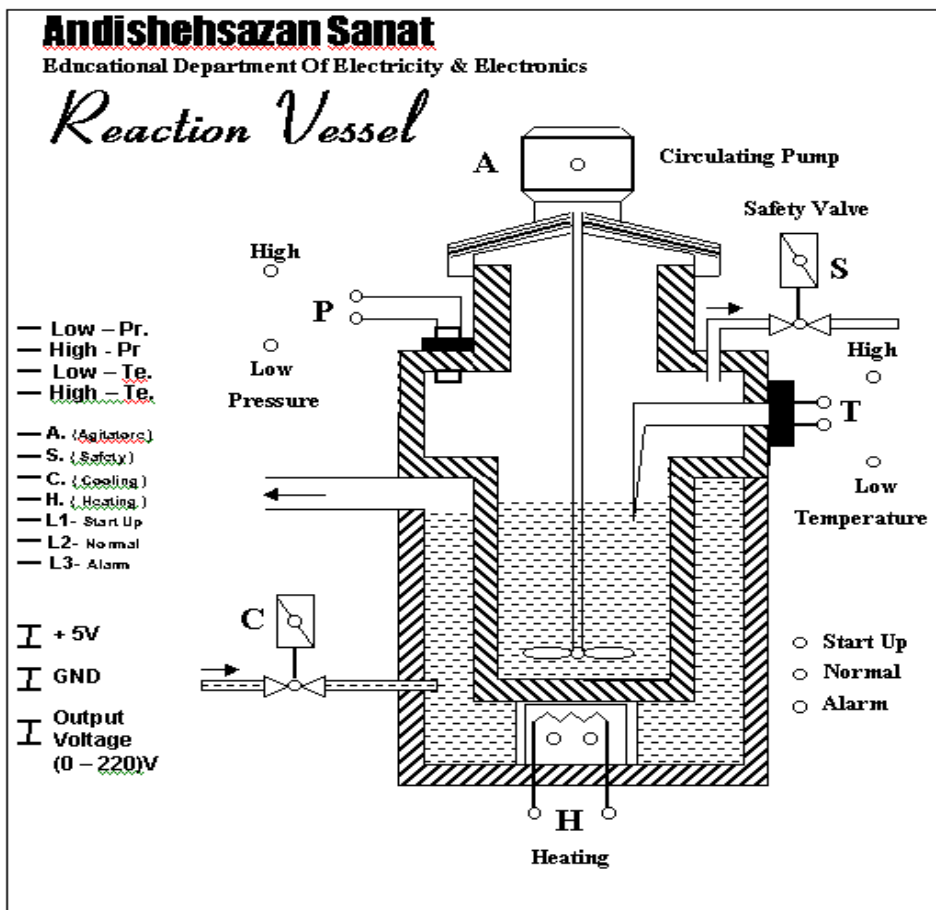
کنترل فرآیندهایی نظیر راکتورها، مخازن تحت فشار، کوره ها و ... رعایت مسائل ایمنی، صدور علائم خطر و نحوه برنامه ریزی سیستم های هشداردهنده، تصمیم گیری بر اساس اطلاعات ورودی مانند فشار، دما و ... از جمله مواردی است که می توان در برنامه ریزی سیمولاتوری مشابه سیمولاتور ارائه شده در شکل ۷-۸ با آن آشنا گشت. در این سیمولاتور کنترل کننده می بایست علاوه بر نظارت بر نحوه کار مدار بطور دائم وضعیت سنسورهای دیجیتال دما و فشار را بررسی نموده و با توجه به اطلاعات بدست آمده از آنها، نسبت به اعمال فرمانهای خاص تصمیم گیری نماید. شکل ۷-۹ نرم افزار مربوطه را نمایش می دهد.





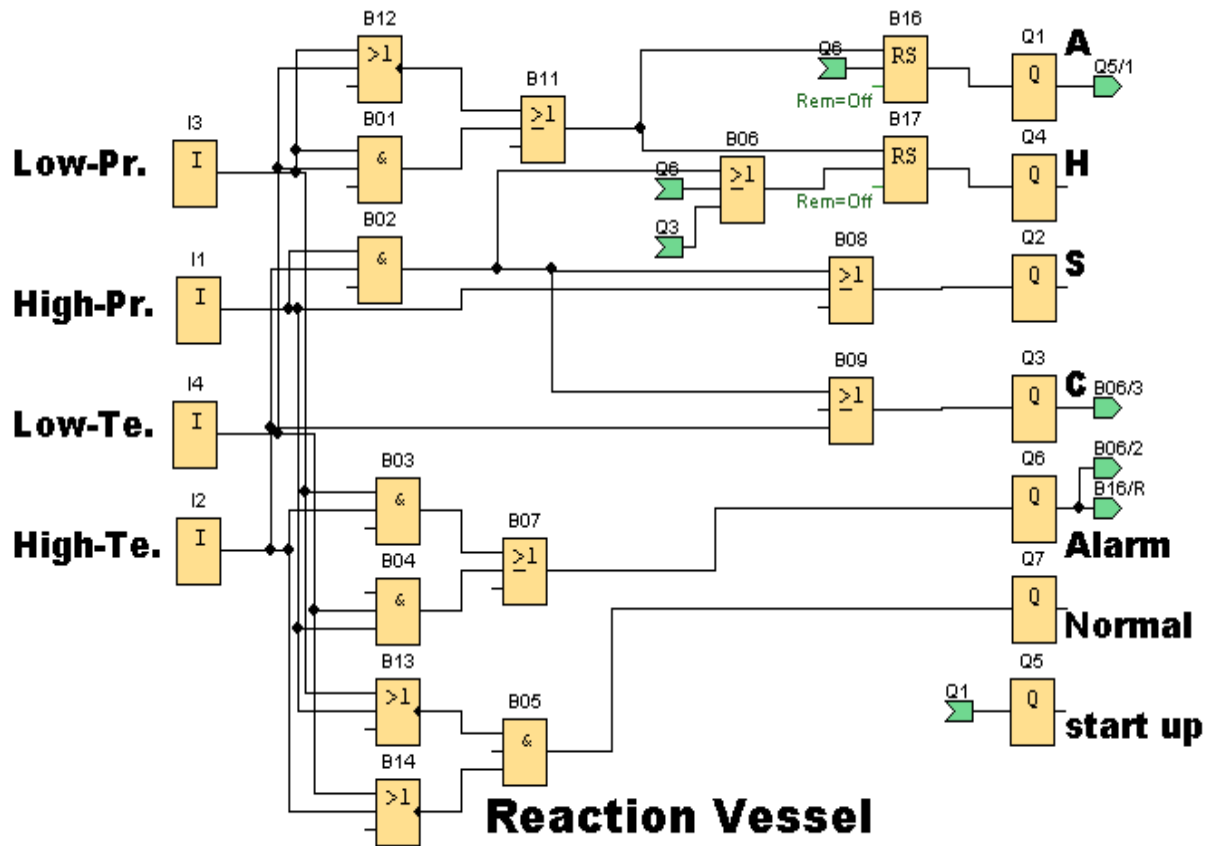
### Traffic Lights

شکل ۷-۷: نرم افزار مربوط به سیستم کنترل ترافیک با استفاده از LOGO



شکل ۸-۷: سیمولاتور کنترل مخزن تحت فشار





شکل ۷-۹: نرم افزار مربوط به کنترل مخزن تحت فشار با استفاده از LOGO



## فصل هشتم

## آشنایی با PLC خانواده S5 زیمنس (سفت افزار)

PLC های خانواده S5 از شرکت زیمنس آلمان یکی از قدیمی ترین کنترل کننده های برنامه پذیر در حد وسیع بوده که در کشور ما نیز در صنایع گوناگون از جمله صنعت سیمان ، پتروشیمی ، صنایع شیمیایی و صنایع وابسته به انرژی از جمله نفت ، آب ، گاز و ... به صورت گسترده استفاده گردیده است . البته در سالهای اخیر با توجه به تنوع تولید در PLC ها و ساخت نمونه های با ظرفیت نرم افزاری بیشتر از جمله PLC های خانواده S7 زیمنس و ... کمتر شاهد بکارگیری این نوع PLC در ساخت و راه اندازی سیستم های جدید صنعتی می باشیم . ولیکن از آنجا که به روز نمودن سیستم های کنترل قدیمی هزینه بسیار بالایی را به صنعت وارد می نماید ، لذا امروزه نیز PLC های خانواده S5 در صنایع ما مشغول بکار بوده و لذا هنوز نیاز به فراگیری نحوه کار با این نوع کنترل کننده ها در بین متخصصین احساس می گردد . بدین منظور در این فصل و فصل آینده به بررسی سخت افزار و نرم افزار مربوطه خواهیم پرداخت . قابل ذکر است با توجه به تعدد نوع PLC در این خانواده ( از جمله 110, 95, 115, ...) و شباهت بسیار زیاد این نمونه ها در این جزوه تاکید بر S5-115U خواهد بود .

## اجزاء سیستم

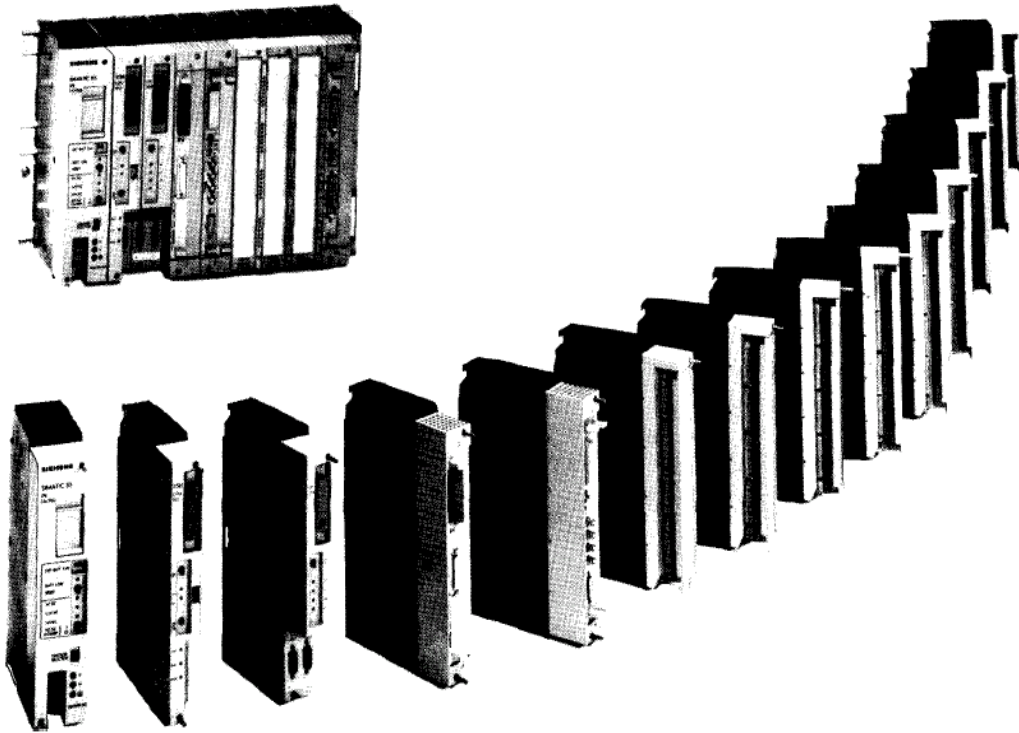
سیستم S5-115U از اجزاء مختلفی مشابه آنچه در شکل ۸-۱ نمایش داده شده ، ساخته می شود . این اجزاء عبارتند از :

- ماژول منبع تغذیه ( PS ) - Power Supply Module
- واحد پردازش مرکزی ( CPU ) - Central Processing Unit
- ماژول های ورودی و خروجی ( I/O ) - Input and Output Modules
- ماژول های ورودی و خروجی هوشمند ( IPs , WFs ) - Intelligent Input / Output Modules
- پردازنده های ارتباطی ( CPs ) - Communications Processors

## منبع تغذیه :

ماژول منبع تغذیه ولتاژ تغذیه ورودی را به یک ولتاژ مورد نیاز برای قسمت های مختلف داخلی تبدیل می نماید . ولتاژهای مورد نیاز برای این سیستم عبارتند از : 24V DC - 115V AC و 230V AC با توجه به نوع ماژول تغذیه مورد استفاده می توان به جریانهای ماکزیمم خروجی برابر با ۳ ، ۷ و ۱۵ آمپر دست یافت . برای منابع تغذیه کمتر از ۷ آمپر نیازی به استفاده از فن نمی باشد . یک باتری از نوع لیتیوم ، حافظه برنامه ، پرچم های پایدار داخلی و مقادیر تایمرها و شمارنده ها را در زمان قطع تغذیه حفظ می نماید . بر روی ماژول تغذیه یک LED برای نمایش وضعیت باتری تعبیه شده که در صورت عدم وجود و یا خرابی باطری روشن می گردد .





شکل ۸-۱: اجزاء مختلف یک PLC از خانواده S5

### واحد پردازش مرکزی

واحد پردازش مرکزی که مغز سیستم می باشد، وظیفه درک و اجرای دستورات مختلف را دارا می باشد. با انتخاب نوع CPU می توان به قابلیت های نرم افزاری بیشتر، حجم حافظه بیشتر و در نتیجه سرعت اجرای بالاتری دست یافت.

انواع مختلف CPU در خانواده S5-115U عبارتند از: CPU 941, CPU 942, CPU 943, CPU 944

### ماژول های ورودی و خروجی هوشمند

شمارش قطار پالس های سریع، آشکارسازی و پردازش لبه های پالس، اندازه گیری زمان و سرعت، کنترل حلقه بسته و تعیین موقعیت، پرو سه هایی وقت گیر بوده که بمدت زمان زیاد و حجم حافظه بالایی نیازمندند. از آنجا که CPU اصلی در اجرای این فرمانها کند می باشد، لذا ماژول های هوشمند خاصی که خود دارای پردازشگر می باشند جهت انجام این افعال طراحی و ساخته شده اند. این ماژول ها پس از تحلیل پروسه نتایج خود را برای استفاده ماژول اصلی به CPU اصلی ارسال می دارد. انواع مختلفی از این ماژول ها در ضمیمه ۶ ارائه شده است.

### ماژول های ارتباطی

سیستم S5-115U دارای تعدادی ماژول ارتباطی است که ارتباط بین دو سیستم و یا ارتباط بین کاربر و سیستم را برقرار می نماید،

دو گروه اصلی از CP ها وجود دارند:

۱- CP ها برای شبکه های محلی

۲- CP ها برای ایجاد شبکه، انتقال سیگنال و ثبت وقایع



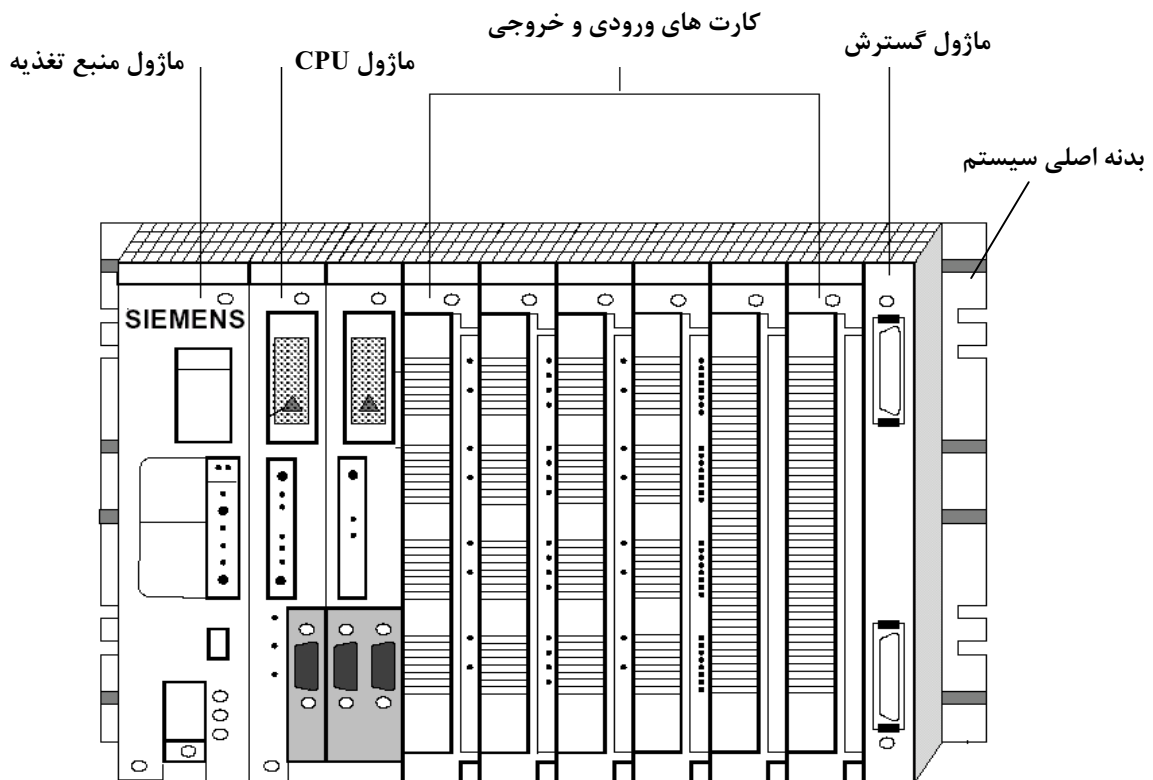


**قابلیت افزایش توانایی ( Expansion Capability )**

اگر کنترل کننده اصلی از توانایی کافی برای کنترل سیستم مورد نظر برخوردار نباشد ، با استفاده از واحدهای توسعه ( EUs ) می توان این عملکرد را بسط داد . یک ماژول واسط مناسب یک کنترل کننده اصلی را به این واحد های توسعه متصل می نماید .

**توضیحات فنی ( Technical Description )**

شکل ۸-۲ واحد مرکزی یک PLC خانواده S5-115U را نمایش می دهد . قسمت های مختلف بر روی شکل نمایش داده شده است .



شکل ۸-۲: واحد مرکزی یک PLC از خانواده S5-115U

**ماژول منبع تغذیه :**

ماژول تغذیه علاوه بر ایجاد ولتاژهای مورد نیاز PLC ، یک ولتاژ پشتیبان ( Backup ) برای Ram سیستم با استفاده از یک باتری لیتیم ایجاد می نماید . این واحد همچنین مونیتورینگ و عملگرهای مربوط به سیگنالینگ را هم اداره می نماید .

شکل ۸-۳ قسمت های مختلف یک منبع تغذیه را نمایش می دهد . این قسمت ها در ادامه بطور کامل توضیح داده می شوند .

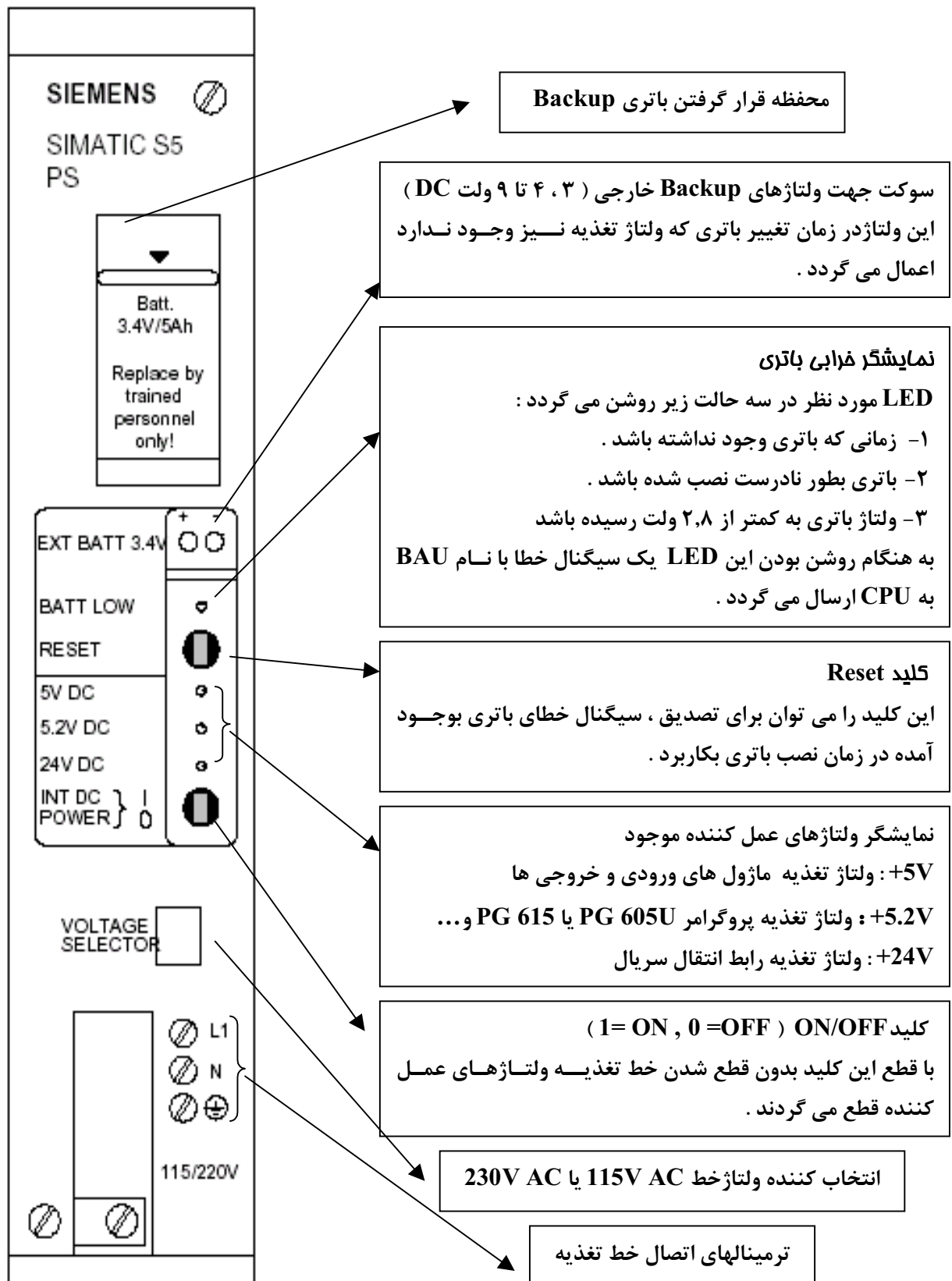
- انتخاب کننده ولتاژ ( Voltage Selector ) ، ولتاژهای خط ورودی ( 115V AC یا 230V AC ) را برای

ماژول های آنالوگ فعال می نماید . قابل ذکر است که ماژول تغذیه PS 951 ، همچنین می تواند با ولتاژ DC

۲۴ ولت نیز عمل نماید .



آموزشگاه برق و الکترونیک صنعتی اندیشه سازان صنعت مجری برگزاری دوره های آموزشی از مقدماتی تا پیشرفته در زمینه های برق ، الکترونیک ، اتوماسیون صنعتی ، ابزار دقیق و کامپیوتر - با مجوز رسمی از سازمان آموزش فنی و حرفه ای کشور و



شکل ۸-۳: قسمت های مختلف ماژول منبع تغذیه



**واحد پردازش مرکزی**

چهار نوع CPU در خانواده S5-115U قابل دسترسی می باشد . در جدول ارائه شده در شکل ۸-۴ ویژگی های مختلف این مدل ها مقایسه شده اند .

	CPU 941	CPU 942	CPU 943	CPU 944
زمان اجرای دستورات برای ۱۰۰۰ جمله	در حدود 10 ms	در حدود 10 ms	در حدود 5 ms	در حدود 1.5 ms
حافظه برنامه داخلی ( RAM )	2 Kbytes	10 Kbytes	48 Kbytes	96 Kbytes
ماکزیمم فضای حافظه در دسترس	18 Kbytes *	42 Kbytes *	48 Kbytes	96 Kbytes
ماکزیمم آدرس ورودی دیجیتال در دسترس	512 I 0.0 to I 63.7	1024 I 0.0 to I 127.7		
ماکزیمم آدرس خروجی دیجیتال در دسترس	512 Q 0.0 to Q 63.7	1024 Q 0.0 to Q 127.7		
ماکزیمم آدرس ورودی آنالوگ در دسترس	64 PW 128 to PW 254			
ماکزیمم آدرس خروجی آنالوگ در دسترس	64 PW 128 to PW 254			
پرچم	1024			
تایمرها	128			
شمارنده ها	128			
محدوده زمانی در دسترس تایمرها	0.01 to 9990 S			
رنج شمارش	0 to 999			
تعداد دستورالعمل ها	در حدود ۱۷۰ دستورالعمل			

\* : مجموع حافظه داخلی و ماژول های خارجی

شکل ۸-۴ : مقایسه انواع CPU های خانواده S5-115U

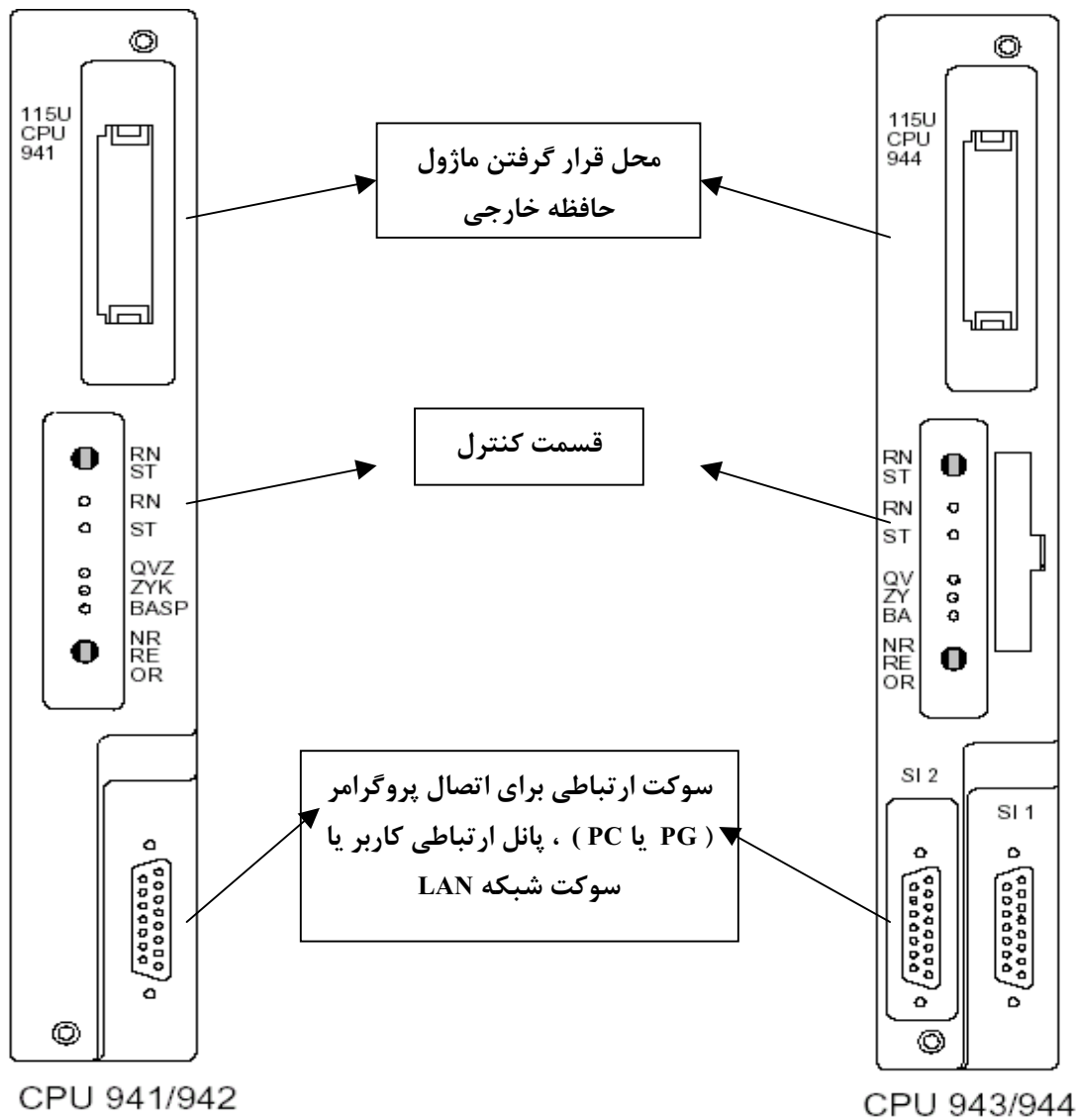
**قسمت های مختلف یک ماژول CPU**

قسمت های مختلف موجود بر روی ماژول CPU برای انجام اعمال زیر در نظر گرفته شده اند .

- قراردادن یک ماژول اضافی حافظه
- اتصال یک برنامه ریز ( Programmer ) PG یا یک کامپیوتر شخصی و یا یک پانل کاری اپراتور ( OP )
- اتصال یک شبکه محلی LAN
- اتصال PLC های ساخته شده توسط سایر کارخانه های سازنده
- تنظیم مود عملکرد PLC

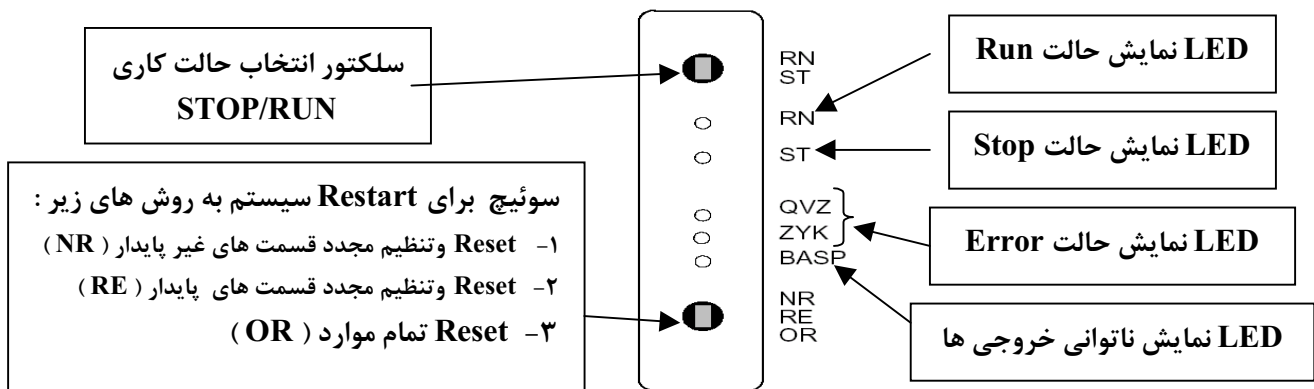


شکل ۵-۸ شمای ظاهری قسمت های مختلف های ماژول CPU را نمایش می دهد .



شکل ۵-۸ : قسمت های مختلف های ماژول CPU های خانواده S5-115U

شکل ۶-۸ قسمت های مختلف موجود بر روی پانل کنترل یک ماژول CPU را نمایش می دهد .



شکل ۶-۸ : قسمت های مختلف موجود بر روی پانل کنترل



**معنی عملکرد LED ها :**

دو LED موجود بر روی پانل کنترل وضعیت CPU را نمایش می دهند . ( LED های RN و ST در شکل ۸ - ۶ ) جدول نمایش داده شده در شکل ۸-۷ وضعیت های مختلف این LED ها را بیان می کند .

LED قرمز	LED سبز	پیغام
ON	ON	CPU در حال اجرای روتین راه اندازی سرد یا در حالت Restart ( * )
ON	OFF	حالت Stop
OFF	ON	حالت Start
OFF	OFF	چک کردن برنامه بصورت پی در پی

\* : روش راه اندازی سرد در ادامه توضیح داده می شود .

شکل ۸-۷ : جدول نمایش حالات مختلف کاری LED های پانل کنترل CPU

**حالات مختلف کاری PLC :**

با قرار دادن سلکتور انتخاب حالت ، در وضعیت STOP ( ST ) یا RUN ( RN ) می توان سیستم را در حالات کاری اجرا یا توقف بکار گرفت . CPU بطور اتومات در تغییر وضعیت اجرا به توقف ، حالت Restart را اجرا می نماید .

**حالت STOP :**

برنامه نوشته شده در حالت کاری STOP ، Scan نمی گردد . آخرین مقدار تایمرها ، شمارنده ها و پرچم ها در زمان رفتن به حالت STOP نگهداری می گردند . در حالت STOP همچنین ماژول های خروجی غیر فعال گشته ( در وضعیت صفر قرار می گیرند ) و LED مربوط به نمایش ناتوانی خروجی ( BASP ) روشن می گردد . این سیگنال ( BASP ) تنها بعد از اجرای بلوک های OB21 یا OB22 ( Restart ) غیر فعال می گردد .

توجه : بلوک های فوق الذکر در فصول بعد بطور کامل معرفی خواهند گشت .

**حالت Restart :**

در دو حالت کاری مشخصات مربوط به Restart سیستم انجام می گیرند :

- ۱- یک انتقال وضعیت از حالت Stop به Start ، که به این عمل Restart دستی سرد اطلاق می گردد .
- ۲- یک انتقال وضعیت از روشن شدن ( Power Up ) سیستم به حالت اجرا ، که به این عمل Restart اتوماتیک سرد اطلاق می گردد .

دو عمل متفاوت در طی عمل Restart می توانند بوقوع بپیوندند :

- ۱- روتین ( برنامه اجرایی ) مربوط به Restart سرد ( PLC نمی تواند بطور مستقیم تحت تاثیر واقع گردد ) .
- ۲- انجام Restart واقعی ( در این حالت مشخصات مختلف مربوط به PLC می تواند در بلوک های مربوط به Restart ( OB21 و OB22 ) تحت کنترل قرار گیرند ) .



### روال Restart سرد

در هنگام انجام روتین Restart سرد توسط CPU ، اعمال زیر صورت می پذیرد :

۱- LED مربوط به BASP روشن می گردد :

- وضعیت LED های مربوط به خطا در طی زمان Restart سرد دستی ، بدون تغییر می ماند .
- LED های مربوط به خطا در زمان Restart سرد اتوماتیک پس از روشن شدن سیستم ، بصورت لحظه ای روشن می گردند .

۲- خروجی ها مقدار سیگنال صفر را نمایش می دهند ( در صورت غیر فعال بودن تمام ماژول های خروجی )

۳- عمل Scan خطوط برنامه در هنگام انجام این عملیات ، صورت نمی پذیرد .

در طی انجام روتین Restart سرد ، پروسوسور وضعیت ماژول های I/O را ثبت نموده و اطلاعات را ذخیره می نماید . این پروسه با جزئیات بیشتر در زیر توضیح داده شده است .

برای ثبت وضعیت ماژول های I/O ، پروسوسور تمام منطقه آدرس ماژول های I/O را بصورت کلمه به کلمه چک می نماید . اگر آدرس یک ماژول بیشتر از یک کلمه باشد ، پروسوسور این کلمه را بوسیله یک نمودن یک بیت در یک منطقه حافظه ویژه که آنرا منطقه داده های سیستمی می نامیم ، ثبت می نماید . این بیت در صورتی که هر دو بایت ورودی یا خروجی در یک کلمه I/O آدرس پذیر باشند ، تنها بوسیله پروسوسور یک می گردد . پروسوسور یک کلمه اطلاعاتی سیستمی ( SD ) را برای چک کردن ۱۶ کلمه I/O بکار می برد .

با استفاده از این متد پروسوسور بایتهای Image ورودی / خروجی را برای سازگاری در طی فرآیند انتقال این اطلاعات به PLC تعیین می نماید . بعنوان مثال با این روش تعیین می گردد که آدرس مورد نظر مربوط به چه نوع ورودی یا خروجی مربوط می باشد ( ورودی دیجیتال یا آنالوگ و یا خروجی دیجیتال یا آنالوگ ) .

- قابل ذکر است که مدت زمان انجام روتین Restart سرد ، از طریق بکارگیری کلمه اطلاعات سیستمی ۱۲۶ ( EAFC h ) و مقداردهی متناسب به آن قابل تنظیم می باشد . این تغییر مدت زمان در مواقع نیاز به بررسی وضعیت سیستم در زمان Restart مورد استفاده بوده و حداکثر می تواند مدت ۶۵۵۳۵ میلی ثانیه ( برابر با عدد FFFF h ) را دارا باشد .

- توجه : در زمان عدم وجود باتری پشتیبان و یا خرابی آن و همچنین اجرای برنامه کنترل از طریق ماژول حافظه جانبی مدت زمان تاخیر Restart ، تقریباً یک ثانیه خواهد بود .

بطور کلی می توان گفت در زمان انجام عمل Restart ، اعمال ذیل صورت می پذیرد :

۱- LED های مربوط به خطا خاموش بوده و LED های مربوط به حالات RUN ، STOP و BASP روشن خواهند بود .

۲- همه خروجی ها غیر فعال می باشند .

۳- PII یا Process Input Image ( تصویر ورودی های فعال ) ، در این هنگام Update نمی گردد . در این

حال ارزیابی ورودی ها فقط از طریق دسترسی مستقیم به آنها ممکن است . ( این کار با بکارگیری دستور L که در فصل بعد معرفی خواهد شد امکان پذیر است ) .

۴- Scan برنامه کنترل در مدت زمان انجام Restart امکان پذیر نمی باشد .

۵- بلوک های سازماندهی مربوط به Restart فعالیت خود را انجام می دهند . ( در حال Restart دستی بلوک

OB21 و در حالت اتومات بلوک OB22 در زمانی که سلکتور در حالت RN باشد اجرا می گردند ) .



۶- تایمرها به فعالیت خود ادامه می دهند .

۷- بلوک های سازماندهی مربوط به وقفه ها ( OB2 تا OB6 ) و بلوک های سازماندهی مربوط به وقفه های زمانی ( OB10 تا OB13 ) تنها در زمانی که وقفه ها بطور کامل فعال گردند ، اجرا می شوند .

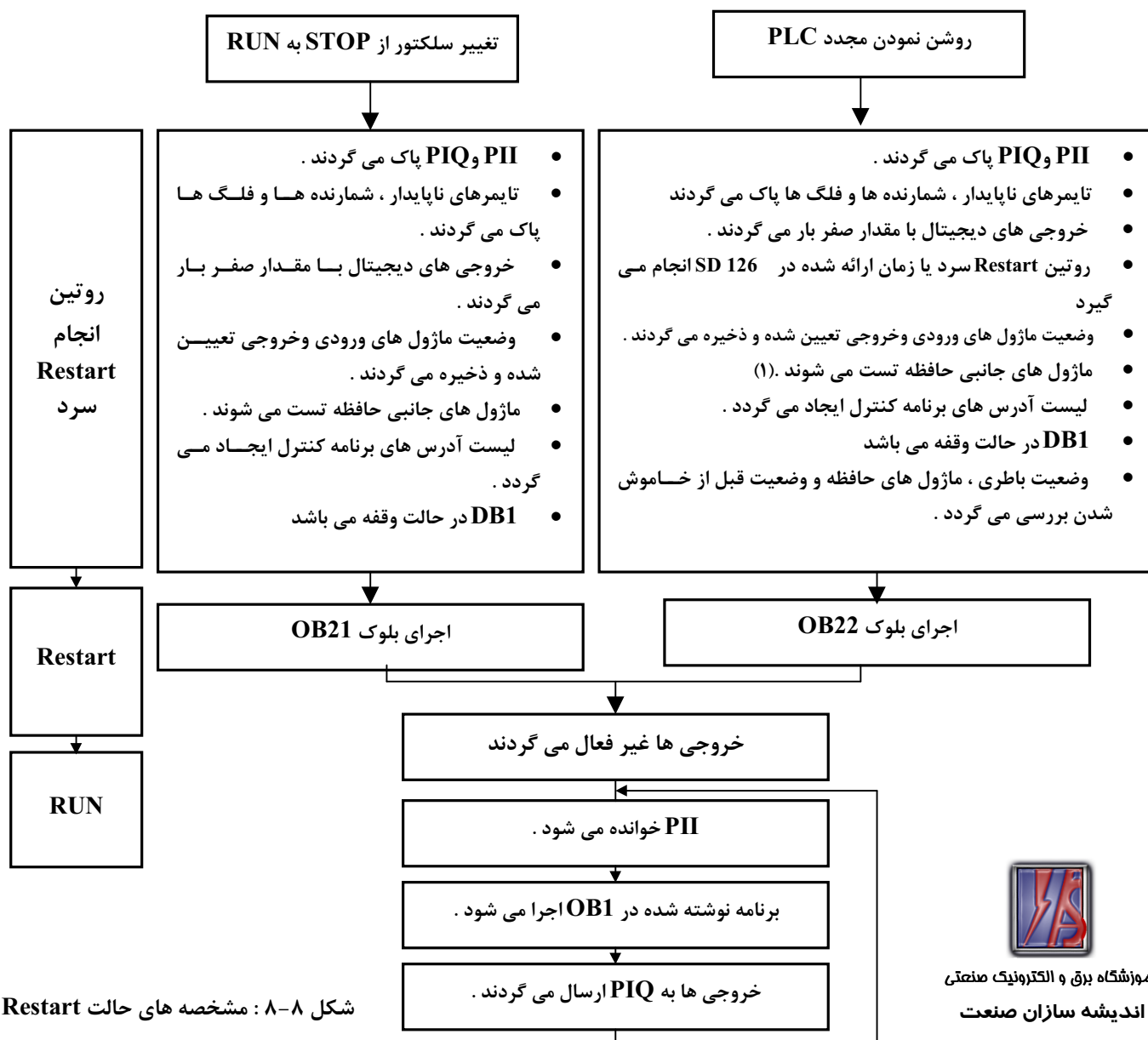
### حالت RUN :

پس از اینکه سیستم عامل مربوط به CPU ، برنامه های کاری مربوط به راه اندازی را به انجام رساند ، سیکل Scan و اجرای برنامه آغاز می گردد . (ابتدا برنامه موجود در OB1)

ورودی های موجود بر روی ماژول های ورودی به طور متناوب و در یک سیکل مشخص خوانده شده و در PII نقش می بندند ، همچنین در این حال مقادیر فلگ های داخلی Update می گردند .

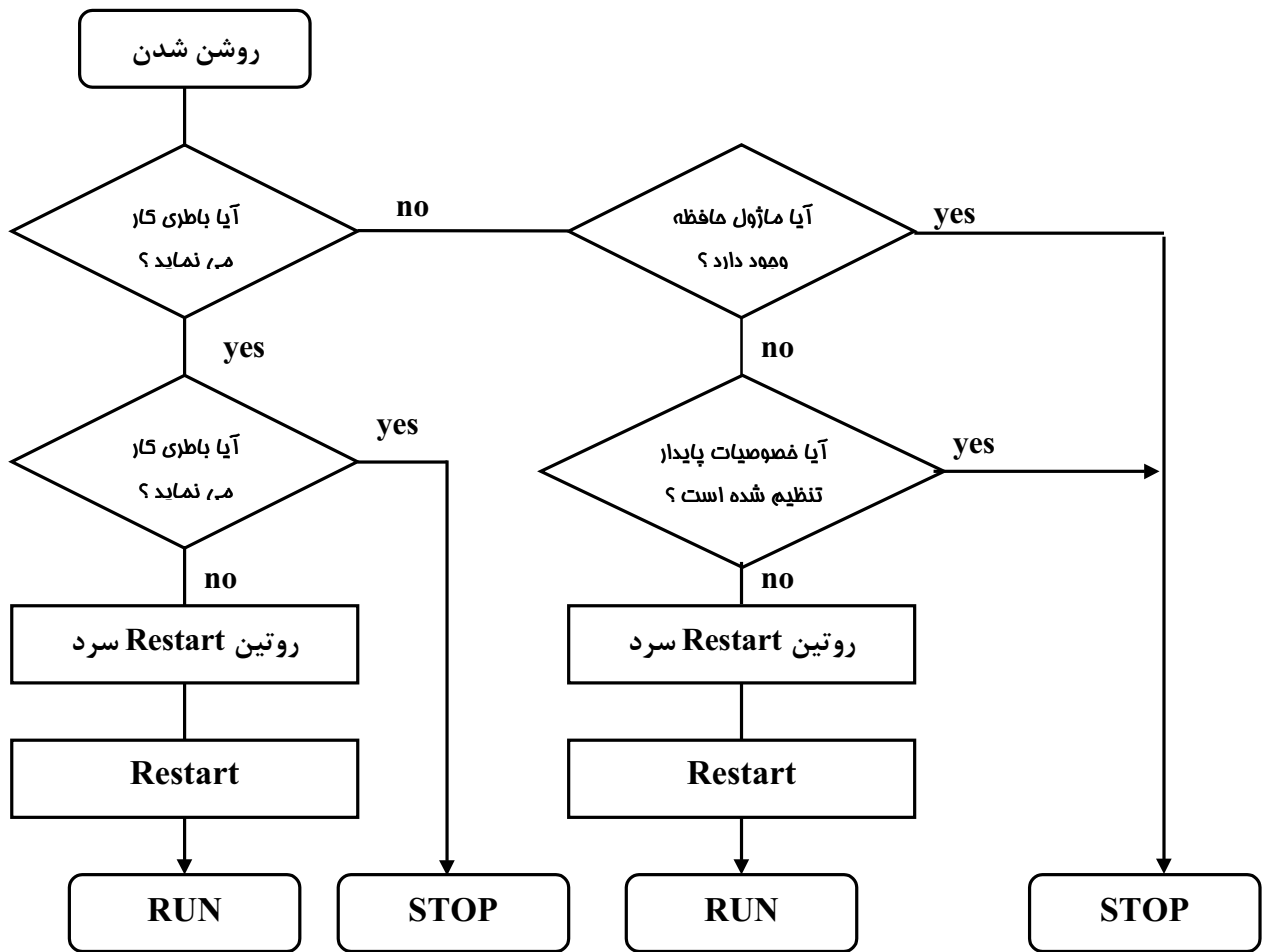
برنامه کنترلی نوشته شده توسط کاربر این اطلاعات به همراه مقادیر جاری فلگ ها ، تایمرها و شمارنده ها را پردازش می نماید . برنامه کنترل نوشته شده توسط کاربر شامل یک سری جملات نوشته شده پی در پی می باشد . پردازنده این جملات را یک به یک از حافظه برنامه فراخوانده و پس از تجزیه و تحلیل ، آنها را اجرا می نماید . نتیجه این پردازش به مکان تصویر خروجی ها ( PIQ ) انتقال می یابد .

نمودار ارائه شده در شکل ۸-۸ مشخصه های حالت Restart و حالت اجرای CPU را نمایش می دهد .



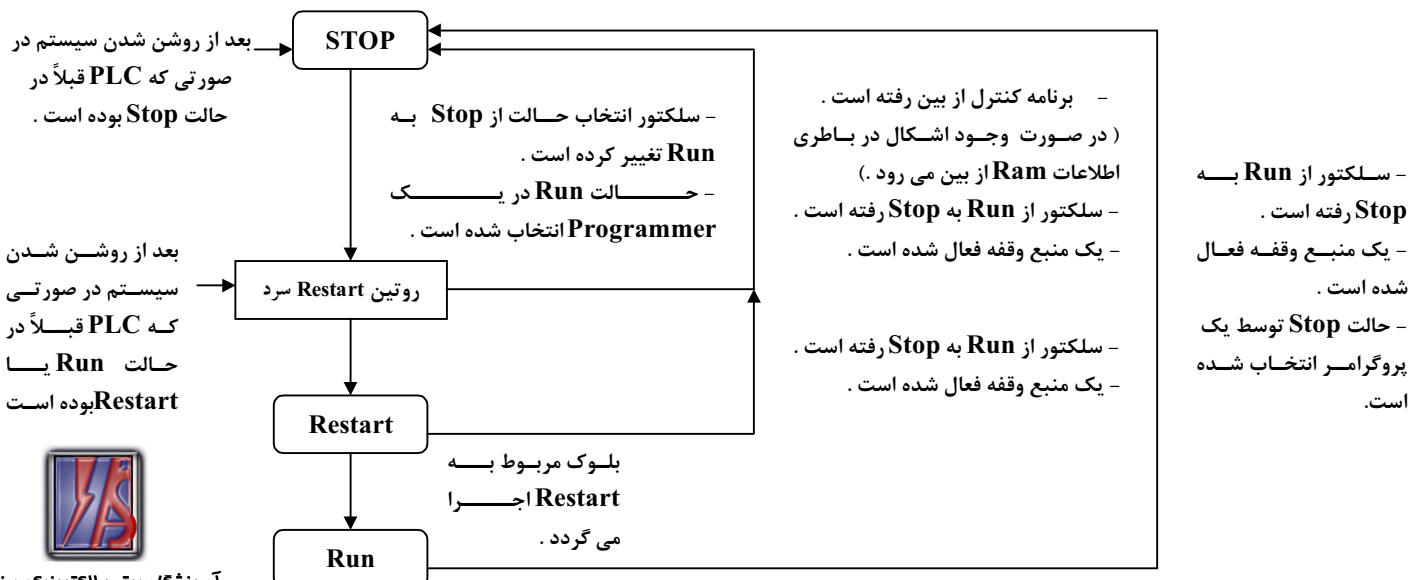
شکل ۸-۸ : مشخصه های حالت Restart

شکل ۸-۹ فلوجارت نحوه کار PLC را در بعد از روشن نمودن شرح می دهد .



شکل ۸-۹ : فلوجارت نحوه کار PLC پس از روشن شدن

شکل ۸-۱۰ شرایط سیستم را در هنگام تغییر حالت کاری نمایش می دهد .



شکل ۸-۱۰ : شرایط سیستم در هنگام تغییر حالت کار



### اندازه گیری زمان Scan برنامه

زمان Scan توسط CPU اندازه گیری شده و در منطقه اطلاعات سیستمی ذخیره می گردد. کاربر می تواند به زمان جاری، حداقل و حداکثر زمان Scan برنامه کنترل در هر زمان دلخواه دسترسی داشته باشد. دقت این زمان یک میلی ثانیه و رنج آن در محدوده صفر تا ۳۲۷۶۷ میلی ثانیه (معادل 7FFF h) می باشد. در انتهای زمان Scan و پس از نگاشت و Update مقادیر PIQ (تصاویر خروجی ها) و مقدار گیری فلگ های داخلی، سیستم عامل مدت زمان Scan را ذخیره می نماید. این مقادیر در مناطق زیر ذخیره می گردند:

- زمان جاری Scan در SD 121
- حداکثر زمان Scan در SD 122
- حداقل زمان Scan در SD 123

اگر مدت زمان Scan برنامه از ۳۲۷۶۷ میلی ثانیه بیشتر باشد، بیت ۱۵ در SD 121 که نشانگر بیت Overflow می باشد یک شده و در SD 122 وارد می گردد. از این پس در سیکل بعد زمان Scan مجدداً اندازه گیری می شود. در فصل بعد نحوه برنامه نویسی برای خواندن زمان Scan ذکر خواهد شد.

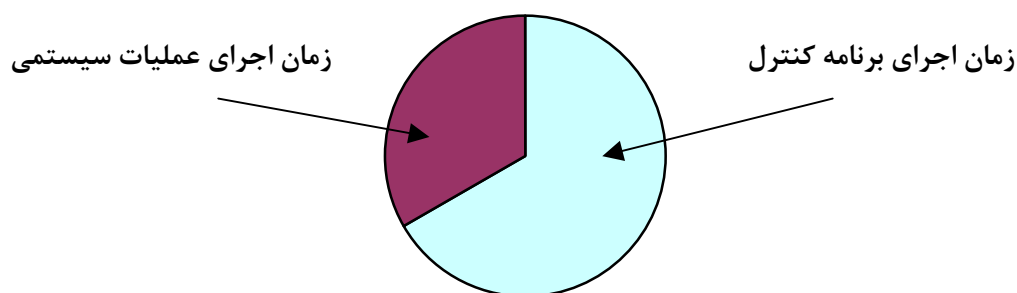
### توجه:

SD ها یا System Data Word ها در حافظه مربوط به سیستم عامل (Operating System Memory) که در ماژول CPU قرار دارد، واقع می شوند. این حافظه در CPU های مدل ۹۴۱ و ۹۴۲، ۳۲ کیلوبایت و در CPU های مدل ۹۴۳ و ۹۴۴، ۶۴ کیلو بایت می باشد.

قابل ذکر است که در ماژول CPU علاوه بر حافظه فوق الذکر که ترکیبی از دو نوع حافظه (PROM و EEPROM یا FLASH RAM) بوده یک حافظه RAM داخلی برای ذخیره برنامه کنترل که توسط کاربر نوشته شده و به PLC انتقال می یابد نیز وجود دارد که در مدل های ۹۴۱ و ۹۴۲ این حافظه ۱۰ کیلو بایت و در نمونه های ۹۴۳ و ۹۴۴، ۴۸ کیلو بایت در نظر گرفته شده است.

### تفمین زمان اجرای برنامه

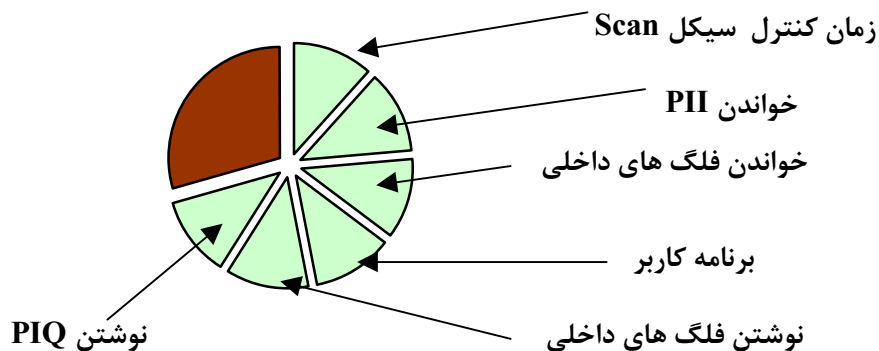
زمان Scan برنامه از دو قسمت زمان عملیات سیستمی و زمان اجرای برنامه کاربر تشکیل می گردد. شکل ۸-۱۱ این تقسیم بندی را نمایش می دهد.



شکل ۸-۱۱: تقسیم بندی زمان Scan برنامه



شکل ۸-۱۲ تقسیم بندی مربوط به زمان اجرای برنامه کنترل را نمایش می دهد .



شکل ۸-۱۲: تقسیم بندی زمان اجرای برنامه کنترل

جدول ارائه شده در شکل ۸-۱۳، مقدار زمان مورد نظر برای هر یک از قسمت های ارائه شده در شکل ۸-۱۲ را نمایش می دهد .

زمان اجرای برنامه کاربر		
زمان مورد نیاز	CPU	زمان بر حسب میکرو ثانیه
زمان سیکل کنترل Scan	CPU 941	160
	CPU 942	160
	CPU 943	160
	CPU 944	210
خواندن PII n = شماره بایت ورودی	CPU 941	140 + n( 30 + زمان تاخیر آماده شدن ماژول)
	CPU 942	
	CPU 943	
	CPU 944	60 + n( 1.7 + زمان تاخیر آماده شدن ماژول)
خواندن فلگ های داخلی n = شماره فلگ داخلی	CPU 941	530 + n( 44 + زمان تاخیر آماده شدن ماژول)
	CPU 942	
	CPU 943	
	CPU 944	60 + n( 1.7 + زمان تاخیر آماده شدن ماژول)
برنامه کاربر (شامل OB1 تا OB5 و OB10 تا OB13)		مجموع زمانهای دستورات استفاده شده در برنامه
نوشتن PIQ n = شماره بایت خروجی	همانند خواندن PII	
نوشتن فلگ های داخلی	همانند خواندن فلگ های داخلی	

شکل ۸-۱۳: مقدار زمان مورد نظر برای اجرای قسمت های مختلف برنامه کنترل



آموزشگاه برق و الکترونیک صنعتی اندیشه سازان صنعت مجری برگزاری دوره های آموزشی از مقدماتی تا پیشرفته در زمینه های برق ، الکترونیک ، اتوماسیون صنعتی ، ابزار دقیق و کامپیوتر- با مجوز رسمی از سازمان آموزش فنی و حرفه ای کشور و

زمان تاخیر آماده شدن ماژول عبارتست از مدت زمان مابین وارد شدن سیگنال درخواست تا زمان آماده شدن سیگنال پاسخ توسط ماژول. این تاخیر به عوامل زیر بستگی دارد:

- زمان تاخیر خود ماژول
  - تاخیر ماژول واسط و طول کابل استفاده شده.
- در یک پیکر بندی توزیع شده (وجود یک شبکه ارتباطی) تاخیر موجود در اتصالات این ارتباط نیز می بایست به زمانهای فوق الذکر افزوده گردد. در یک نرخ توزیع اطلاعات برابر با ۶ میکرو ثانیه بر کیلومتر تاخیر ۱۲ میکروثانیه ای می بایست در طی مسافت ۱۰۰۰ کیلومتر در نظر گرفته شود. اگر CPU سیگنال آمادگی را در عرض ۱۶۰ میکرو ثانیه دریافت نکند، به حالت STOP رفته و LED خطای QVZ را روشن می نماید. شکل ۸-۱۴ زمان تاخیر جهت آمادگی ماژول های مختلف I/O را نمایش می دهد.

I/O Modules	Ready Delay Time in $\mu$ sec
Digital modules	2
Analog modules	16
313 watchdog modules	1
IP 240	1
IP 241	1
IP 242 (Release A00)	140
IP 242 (Release A01)	50
IP 243 (Analog module)	35
IP 244	150
IP 245	0.5
IP 246	1.5
IP 247	1.5
IP 252	10
WF 625	3
CP 513-3M	5
CP 524	1
CP 525	3
CP 526	3
CP 527	3
CP 530	3 to 130
CP 535	3
CP 551	3
CP 552	3
CP 5430	1
CP 143	3

شکل ۸-۱۴: زمان تاخیر جهت آمادگی ماژول های مختلف I/O



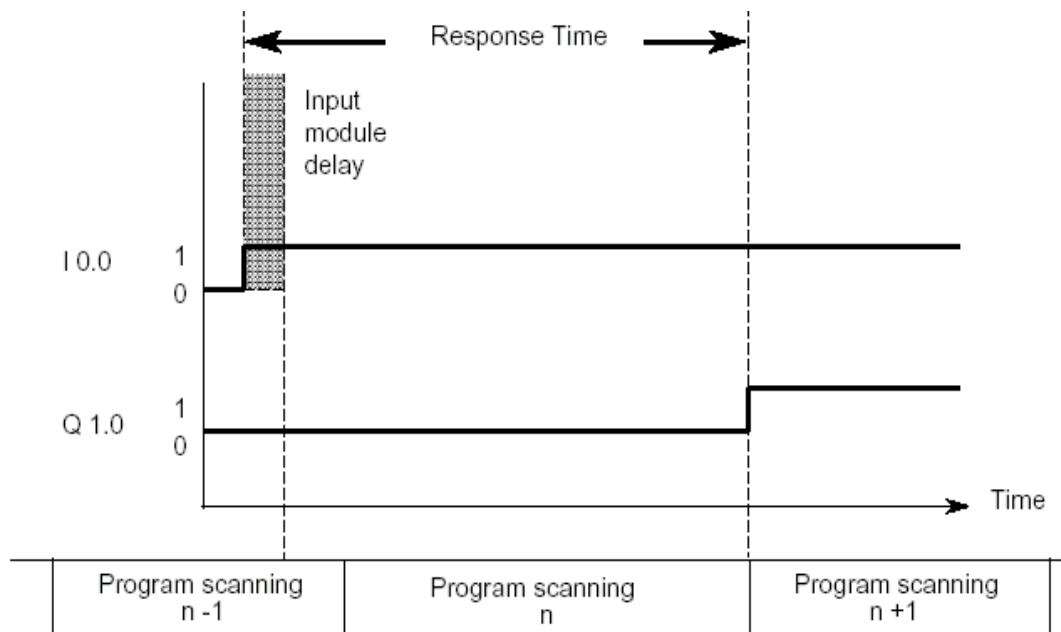
آموزشگاه برق و الکترونیک صنعتی اندیشه سازان صنعت مجری برگزاری دوره های آموزشی از مقدماتی تا پیشرفته در زمینه های برق، الکترونیک، اتوماسیون صنعتی، ابزار دقیق و کامپیوتر - با مجوز رسمی از سازمان آموزش فنی و حرفه ای کشور و

### زمان پاسخ دهی

زمان پاسخ دهی عبارتست از مدت زمان بین تحریک یک ورودی تا پاسخ دهی خروجی متناظر در طی یک برنامه که ورودی بطور مستقیم و از طریق نرم افزار به خروجی متصل گشته است. این مدت زمان به طور معمول مجموعی از عوامل زیر می باشد:

- تاخیر ذاتی ماژول های ورودی
- مدت زمان Scan برنامه

تأخیر ماژول خروجی در این محاسبه قابل چشم پوشی است. شکل ۸-۱۵ این زمان را نمایش می دهد.



شکل ۸-۱۵: دیاگرام زمانی مربوط به زمان پاسخ دهی خروجی

### باتری پشتیبان ( Backup )

این باتری در زمان خاموش بودن PLC برنامه های موجود در RAM و اطلاعات پایدار را حفظ می نماید. عمر مفید این باتری در حدود دو سال می باشد. قبل از قراردادن باتری لیتیوم در ماژول منبع تغذیه باید باتری بمدت دو ساعت به یک بار ۱۰۰ اهمی متصل شده تا تخلیه گردد.

### ماژول حافظه جانبی

سه نمونه ماژول حافظه با ظرفیت های متفاوت برای خانواده S5-115U در دسترس می باشد. جدول ارائه شده در شکل ۸-۱۶ مدل های مختلف این نمونه ها را نمایش می دهد.



Memory Submodule		توانایی استفاده در CPU مدل				ساختار
نوع	ظرفیت (۱)	941	942	943	944	
EPROM	8 Kbytes					byte
EPROM	16 Kbytes					byte
EPROM	32 Kbytes					byte
EPROM	64 Kbytes (۲)					byte
EPROM	128 Kbytes (۳)					Word
EEPROM	8 Kbytes					byte
EEPROM	16 Kbytes					byte
RAM	8 Kbytes					byte
RAM	16 Kbytes					byte
RAM	32 Kbytes					byte

(۱) : یک کیلو بایت حافظه تقریباً گنجایش ۱۰۰۰ جمله برنامه به زبان S5 را دارا می باشد .

(۲) : تنها ۴۸ کیلوبایت قابل استفاده می باشد .

(۳) : تنها ۹۶ کیلو بایت قابل استفاده می باشد .

(۴) : استفاده از ماژول حافظه RAM در مدل‌های ۹۴۳ و ۹۴۴ ضروری نمی باشد زیرا در این مدل‌ها حافظه مورد

نیاز در RAM داخلی قابل دسترس می باشد .

شکل ۸-۱۶: انواع مختلف ماژول های حافظه موجود برای خانواده S5-115U

### پروگرامر ( PG )

این دستگاه جانبی که برای PLC های زیمنس مورد استفاده قرار می گیرد دارای کاربردهای زیر می باشد :

- نوشتن برنامه و انتقال آن به PLC
- تست برنامه نوشته شده
- مانیتورینگ برنامه

پروگرامرها در حالت On-Line ( در ارتباط مستقیم با PLC ) و Off-Line ( بدون ارتباط با PLC ) قابل

استفاده بوده و در مدل‌های زیر در دسترس است :

PG 605U , PG 615, PG 635 , PG 670, PG 675, PG 685 , PG 695, PG 710, PG 730  
PG 750, PG 770



آموزشگاه برق و الکترونیک صنعتی اندیشه سازان صنعت مجری برگزاری دوره های آموزشی از مقدماتی تا پیشرفته در زمینه های برق ، الکترونیک ، اتوماسیون صنعتی ، ابزار دقیق و کامپیوتر- با مجوز رسمی از سازمان آموزش فنی و حرفه ای کشور و

مشهد مقدس - ۷۶۱۸۲۸۶-۷۶۴۴۰۷۸-۰۵۱۱

وزارت کار و امور اجتماعی - اعطای مدرک با کد بین المللی I.L.O

**صفحه کاری اپراتور ( Operator Panel )**

این صفحه که از ماژولهای جانبی PLC محسوب می گردد ، در موارد زیر کاربرد دارد :

- نمایش مقادیر جاری تایمرها و شمارنده های داخلی
  - تنظیم مقادیر جدید پارامترهای سیستم
  - نمایش متن های مورد نیاز تعریف شده در برنامه کنترل
  - نمایش مقادیر ورودی ها ، خروجی ها ، پرچم ها و ... ( تنها در مدل OP396 )
- مدل های مختلف OP : OP 393 , OP395 , OP 396
- مدل های مختلف مانیتور : DG 335 و DS 075 ( تنها با CPU 943, CPU 944 )

**توجه : جهت کسب اطلاعات بیشتر در خصوص ماژول های اضافی به ضمیمه ۷ مراجعه نمایید .**

**توجه : جهت کسب اطلاعات بیشتر در خصوص نحوه نصب اجزا. سخت افزار بر روی بدنه ( Rock ) اصلی سیستم و همچنین سیم بندی های مربوطه به فایل PDF مربوط در راهنمای PLC ( 115Umanual.pdf ) که در نسخه الکترونیک جزوه موجود است مراجعه نمایید .**



## فصل نهم

# برنامه نویسی به زبان STEP 5

### مقدمه :

یک برنامه کنترل مجموعه دستورات عملیاتی است که به سیستم PLC فرمانهایی جهت کنترل پروسه صادر می نماید ، لذا باید این برنامه به زبانی خاص و مطابق با دستورات قابل درک برای PLC نوشته شود . زبان برنامه نویسی که خانواده S5 زیمنس از آن استفاده می نمایند ، STEP 5 نامیده می شود . همانگونه که در قسمت های قبل به آن اشاره شد ، سه روش مختلف برنامه نویسی در خانواده S5 ، قابل استفاده می باشد ، این زبانها عبارتند از :

۱- نردبانی LAD ( Ladder )

۲- فلوجارتی CSF ( Control System flowchart )

۳- عبارتی STL ( Statement List )

در این فصل ابتدا به معرفی دستورات موجود در زبان S5 پرداخته و نحوه استفاده از آنها را در روش های مختلف برنامه نویسی بررسی می نمایم سپس در قالب چند مثال ، کاربرد این دستورات بصورت کامل تر بررسی می گردد .

### زبان برنامه نویسی S5

زبان برنامه نویسی S5 از سه گروه دستورات عملیاتی مختلف تشکیل شده است :

- دستورات اصلی ( Basic Operations ) که این دستورات در بلوکهای سازماندهی ( Organization Blocks ) ، بلوکهای برنامه ( Program Blocks ) ، بلوکهای ترتیبی ( Sequence Blocks ) و بلوک های عملگر ( Function Blocks ) استفاده می گردند . به جز عملگرهای + ( Addition ) - ( Subtraction ) و عملگرهای سازماندهی ، سایر دستورات عملیاتی اصلی می توانند به عنوان ورودی و خروجی در برنامه های نوشته شده به روش های STL ، CSF و LAD بکار گرفته شوند .
- دستورات تکمیلی ( Supplementary Operations ) که این دستورات شامل عملگرهای پیچیده ، عملگرهای جایگزینی ، تست ، انتقال ( Shift ) و تبدیل می باشند .
- عملگرهای سیستمی ، که جهت دسترسی مستقیم به سیستم عامل می توان از آنها استفاده نمود . تنها یک برنامه نویس با تجربه توانایی استفاده از این دستورات را دارد .



**روش های آدرس دهی :**

چون در این نوع PLC ، خطوط انتقال اطلاعات ۸ بیتی می باشند ، لذا ورودی ها ، خروجی ها و فلگ ها در دسته های ۸ بیتی ( یک بیتی ) سازماندهی می شوند .

فلگ ها یا پرچم ها ( Flag ) : محل هایی از حافظه هستند که جهت نگهداری وضعیت برخی نتایج و یا خروجی ها استفاده می شوند . فلگ ها به دو صورت پایدار ( Retentive ) و ناپایدار ( non-Retentive ) موجود می باشند . در نمونه های پایدار اطلاعات موجود با قطع تغذیه از بین نمی روند حال آنکه در نمونه های ناپایدار با قطع تغذیه اطلاعات موجود از بین خواهند رفت . تعداد فلگ ها در نمونه های مختلف PLC متفاوت می باشند ولیکن بطور معمول تعداد فلگ های پایدار و ناپایدار در یک PLC برابر می باشند . به عنوان مثال در صورتی که تعداد فلگ های یک PLC برابر M باشد ، فلگ های با شماره کمتر از M/2 پایدار و فلگ های با شماره بیش از M/2 ناپایدار خواهند بود .

در آدرس دهی ورودی ها ، خروجی ها و فلگ ها ابتدا نوع عملوند ( ورودی / خروجی / فلگ ) را بکار بردن حروف اختصاری ( F / O / I ) مشخص کرده و سپس شماره بایت و پس از یک علامت dot شماره بیت ذکر می گردد . به عنوان مثال عبارت I 5.3 ، بیانگر بیت سوم از بایت پنجم ورودی می باشد . جدول ارائه شده در شکل ۹-۱ تعداد ورودی ها ، خروجی ها ، فلگ ها ، تایمرها و شمارنده ها را در نمونه های مختلف CPU نمایش می دهد .

نوع عملوند	تعداد در دسترس	
	CPU 941	CPU 942 , 943,944
I	0.0 to 63.7	0.0 to 127.7
Q	0.0 to 63.7	0.0 to 127.7
F	0.0 to 255.7	0.0 to 255.7
T	0 to 127	0 to 127
C	0 to 127	0 to 127

شکل ۹-۱ : تعداد عملوندهای قابل دسترسی در نمونه های مختلف CPU

**عملگرهای عمومی**

**عملگر AND : ( شکل ۹-۲ )**

Example		Circuit Diagram	
<p>در این مثال خروجی Q3.5 زمانی یک خواهد شد که سه ورودی گیت AND یک باشند ، در غیر اینصورت خروجی صفر خواهد بود .</p>			
STL	CSF	LAD	
<p>A I 1.1 A I 1.3 A I 1.7 = Q 3.5</p>			





عملگر OR : ( شکل ۹-۳ ) □

Example		Circuit Diagram	
<p>در این مثال خروجی Q3.2 زمانی یک خواهد شد که حداقل یکی از سه ورودی OR گیت یک باشند ، در غیر اینصورت خروجی صفر خواهد بود .</p>			
STL	CSF	LAD	
<pre> O   I   1.2 O   I   1.7 O   I   1.5 =    Q   3.2                     </pre>			

شکل ۹-۳: طرز نمایش گیت OR در روش های مختلف برنامه نویسی

عملگر NOT - AND : ( شکل ۹-۴ ) □

با استفاده از این ترکیب می توان مقادیر صفر در ورودی PLC را خواند . به عبارت دیگر جهت خواندن کنتاکت های بسته ( Normally Closed ) می توان از این ترکیب استفاده نمود .

Example		Circuit Diagram	
<p>در این مثال خروجی Q3.0 زمانی یک خواهد شد که <math>I1.6 = 0</math> بوده و <math>I1.5 = 1</math> باشند .</p>			
STL	CSF	LAD	
<pre> A   I   1.5 AN  I   1.6 =    Q   3.0                     </pre>			

شکل ۹-۴: نحوه خواندن مقدار صفر در ورودی PLC



توجه : مقدار صفر را در دستور OR می توان با عبارت ON (NOT-OR) از ورودی خواند .

**کاربرد پرانتز در زبان S5**

همانند اکثر زبانهای برنامه نویسی در برنامه نویسی به زبان S5 نیز ، علامت پرانتز برای تعیین اولویت عملگرها استفاده می شود . در زبان S5 ، AND نسبت به OR دارای تقدم می باشد ، لذا برای دادن تقدم به گیت OR نسبت به AND می بایست عمل OR در داخل پرانتز صورت گیرد ( در روش STL) . مثالهای ارائه شده در شکل های ۵-۹ و ۹-۶ این مسئله را نمایش می دهند .

Example		Circuit Diagram	
<p>به علت تقدم عملگر AND نسبت به عملگر OR نیازی به استفاده از پرانتز در برنامه نویسی به روش STL نمی باشد . در این مثال در روش STL، ابتدا I 1. 5 و I 1. 6 ، همچنین I 1. 4 و I 1. 3 AND شده و سپس حاصل این دو مقدار OR می شوند .</p>			
STL	CSF	LAD	
<pre> A   I   1.5 A   I   1.6 O A   I   1.4 A   I   1.3 =    Q   3.1                     </pre>			

شکل ۹-۵ : تقدم عملگر AND نسبت به OR

Example		Circuit Diagram	
<p>جهت انجام عمل OR بر روی دو عملوند I 6. 2 و I 6. 3 ، قبل از انجام عمل AND حاصل آنها با عملوند I 6. 1 لازم است از پرانتز در روش STL جهت ایجاد تقدم استفاده گردد .</p>			
STL	CSF	LAD	
<pre> O O A( O   I   6.2 O   I   6.3 ) =    Q   2.1                     </pre>			

شکل ۹-۶ : ایجاد تقدم برای عملگر OR نسبت به AND



شکل ۷-۹ مثالی دیگر از ایجاد تقدم برای عملگر OR نسبت به AND را با بکارگیری پرائنتز نمایش می دهد .

Example		Circuit Diagram	
<p>جهت انجام عمل OR ، قبل از انجام عمل AND ، لازم است از پرائنتز در روش STL جهت ایجاد تقدم استفاده گردد . در این مثال به یک پایه ورودی گیت AND حاصل ( I 1 . 4 و I 1 . 5 ) که داخل پرائنتز قرار گرفته اند ( متصل شده و پایه دیگر AND حاصل OR ( I 1 . 4 و I 1 . 5 ) که داخل پرائنتز قرار گرفته اند )</p>			
STL	CSF	LAD	
<pre> A( O I 1.4 O I 1.5 ) A( O I 2.0 O I 2.1 ) = Q 3.0                     </pre>			

شکل ۷-۹ : ایجاد تقدم برای عملگر OR نسبت به AND

□ عملگر فلیپ فلاپ : ( شکل ۸-۹ )

این عملگر برای Set و Reset کردن فلگ ها و خروجی ها استفاده می گردد .

Example		Circuit Diagram	
<p>در این مثال در صورت فعال شدن I2.7 خروجی Q3.5 یک خواهد شد با تغییر وضعیت ورودی فوق الذکر از یک به صفر تغییری در خروجی حاصل نخواهد شد . به عبارت دیگر خروجی Latch خواهد شد . با یک شدن I 1 . 4 خروجی صفر خواهد شد . در بکارگیری فلیپ فلاپ ، عملی که به خط انتهایی برنامه نزدیکتر است ، نسبت به دیگری تقدم دارد . ( در این مثال Reset نسبت به Set تقدم دارد . )</p>			
STL	CSF	LAD	
<pre> Q I 2.7 S Q 3.5 Q I 1.4 R Q 3.5 NOP 0 *                     </pre>			

شکل ۸-۹ : عملگر فلیپ فلاپ



شکل ۹-۹، فلیپ فلاپ RS را به صورتی که خروجی آن در فلگ ذخیره می گردد، نمایش می دهد.

Example		Circuit Diagram	
<p>در این مثال ابتدا مقدار خروجی فلیپ فلاپ در فلگ F 1.7 قرار گرفته و سپس با خواندن مقدار فلگ آنرا در خروجی Q 3.4 قرار می دهد.</p> <p>علامت مساوی یک فرمان مقدار دهی در زبان STL می باشد، به عبارت دیگر با استفاده از این دستور مقدار خوانده شده از فلگ که در یک حافظه میانی قرار گرفته در خروجی Q 3.4 قرار داده می شود.</p>			
STL	CSF	LAD	
<pre>A I 2.6 S F 1.7 A I 1.3 R F 1.7 A F 1.7 = Q 3.4</pre>			

شکل ۹-۹: استفاده از فلگ برای ذخیره خروجی یک فلیپ فلاپ

□ دستور NOP 0:

این دستور، دستوری مختص PLC های زیمنس می باشد. از این دستور زمانی استفاده می نمایم که بخواهیم از یک خروجی استفاده ای ننماییم. به عبارت دیگر این دستور یک دستور جایگزین بوده که هیچ عمل خاصی را انجام نمی دهد. از دیگر کاربردهای این دستور استفاده در خطوط برنامه STL به جای دستوراتی است که حذف می گردند. این عمل این اجازه را به نرم افزار S5 می دهد که بتوان خطوط برنامه STL را به سایر روش ها از جمله LAD یا CSF تبدیل نماید (این مسئله در زمان معرفی تایمرها و شمارنده های S5 بیشتر بررسی می گردد). مثال ارائه شده در شکل ۹-۱۰ نمونه ای از کاربرد دستور NOP 0 را نمایش می دهد. در این مثال دو خط از برنامه اصلی برداشته شده و دستور NOP 0 جایگزین آن شده است. این روش یکی از روشهای عیب یابی برنامه های نوشته شده به زبان S5 می باشد.

A I 1.0	→	A I 1.0
S Q 2.0		S Q 2.0
A I 1.1		A I 1.1
R Q 2.0		R Q 2.0
A Q 2.0		NOP 0
= Q 2.1		: BE
: BE		

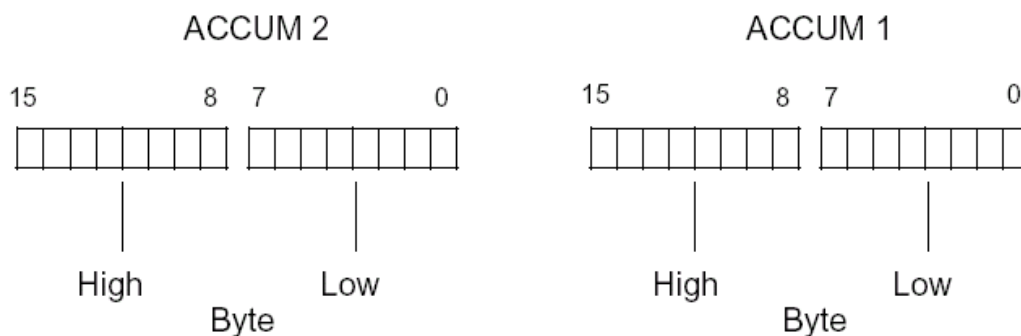
شکل ۹-۱۰: مثالی از کاربرد دستور NOP 0



**عملگرهای بارگذاری و انتقال ( Load & Transfer ) :**

**انبارک یا آکومولاتور ( Accumulator ) :**

جهت تبادل اطلاعات بین قسمت های مختلف نیاز به یک حافظه میانی است ، که عمل تبادل اطلاعات از طریق آن صورت پذیرد . به این حافظه میانی آکومولاتور گفته می شود . آکومولاتور ها از نوع ثبات ( Register ) می باشند . در PLC های خانواده S5 دو آکومولاتور وجود دارد که با نامهای ACCUM 1 و ACCUM 2 نامگذاری شده اند . شکل ۹-۱۱ ساختار آکومولاتورها را نمایش می دهد .



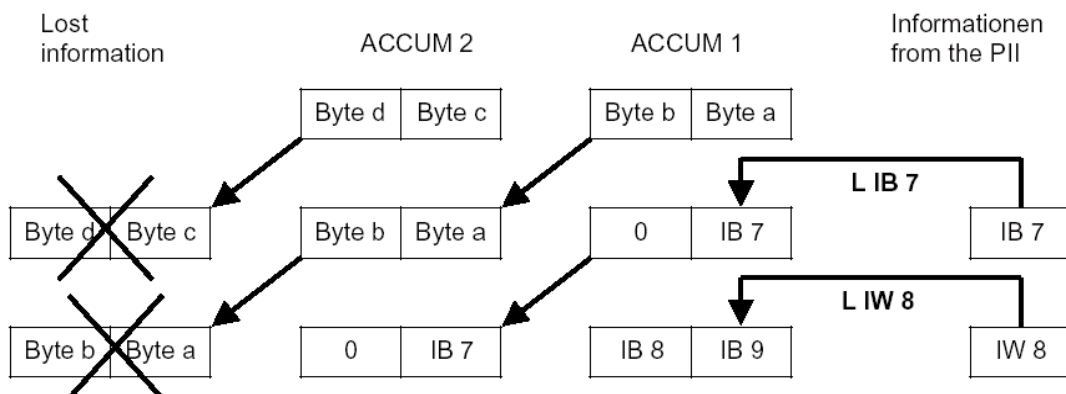
شکل ۹-۱۱ : ساختار آکومولاتورها

**دستور Load :**

در دستور Load ، اطلاعات از یک نقطه حافظه یا از منطقه تصویر ورودی ها به آکومولاتور یک منتقل می گردد . مقدار قبلی موجود در آکومولاتور یک به آکومولاتور دو منتقل می گردد . در این حال آخرین مقدار آکومولاتور ۲ از بین خواهد رفت .

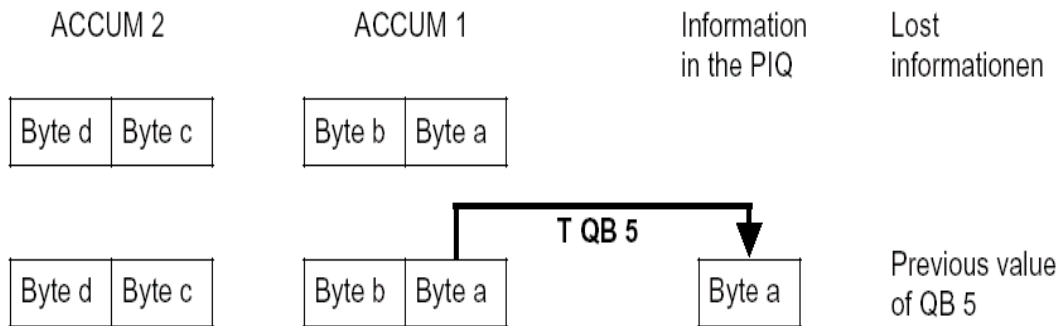
در مثال ارائه شده در شکل ۹-۱۲ دو بایت ورودی پشت سر هم IB 7 و IB 8 از منطقه تصویر ورودی ( PII ) به آکومولاتور بار می شوند . در اثر این عمل هیچ تغییری در PII حاصل نمی شود .

**توجه :** بایت ورودی در زبان S5 با IB n مشخص می گردد ، که n شماره بایت ورودی می باشد . یک بایت ورودی شامل ۸ بیت می باشد . به عنوان مثال بایت ورودی که بر روی کارت های ورودی قرار دارد شامل ۸ بیت از I 1.0 تا I 1.7 می باشد . قابل ذکر است که یک کارت ورودی دیجیتال دارای ۴ بایت ورودی می باشد . هر دو بایت پشت سر هم را یک کلمه ورودی می نامند . به عنوان مثال بایتهای ورودی یک و دو را با کلمه ورودی یک مشخص می نماییم . این کلمه بصورت IW 1 نمایش داده می شود .



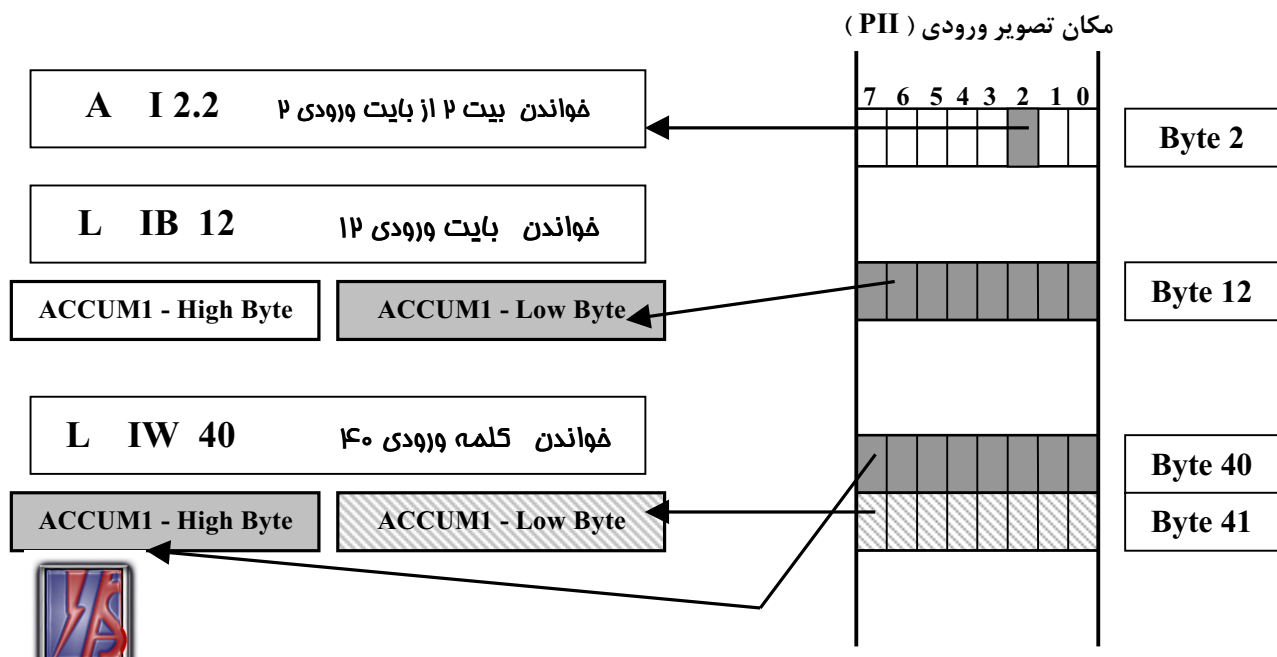
**دستور Transfer :**

در دستور Transfer ، اطلاعات از آکومولاتور یک به یک نقطه حافظه و یا منطقه تصویر خروجی (PIQ) منتقل می گردد . مقدار موجود در آکومولاتور یک در اثر عمل انتقال تغییری نخواهد کرد . در اثر انتقال اطلاعات به منطقه خروجی دیجیتال ، مقدار قبلی PIQ بصورت خودکار Update می گردد . در مثال ارائه شده در شکل ۹-۱۳ ، اطلاعات موجود در بایت پایین آکومولاتور ۱ به بایت خروجی QB 5 منتقل می گردد



شکل ۹-۱۳: انتقال اطلاعات از آکومولاتور به محل تصویر خروجی با استفاده از دستور Transfer

**توجه :** بایت خروجی در زبان S5 با QB n مشخص می گردد ، که n شماره بایت ورودی می باشد . یک بایت خروجی شامل ۸ بیت می باشد . به عنوان مثال بایت خروجی ۴ که بر روی کارت های خروجی قرار دارد شامل ۸ بیت از Q 4.0 تا Q 4.7 می باشد . قابل ذکر است که یک کارت خروجی دیجیتال دارای ۴ بایت خروجی می باشد . هر دو بایت پشت سر هم را یک کلمه خروجی می نامند . به عنوان مثال بایتهای خروجی ۴ و ۵ را با کلمه خروجی ۴ مشخص می نماییم . این کلمه بصورت QW 4 نمایش داده می شود . شکل های ۹-۱۴ و ۹-۱۵ روش های خواندن (Load) و انتقال (Transfer) اطلاعات از ورودی را بصورت های بیتی ، بایتی و کلمه ای نمایش می دهند .

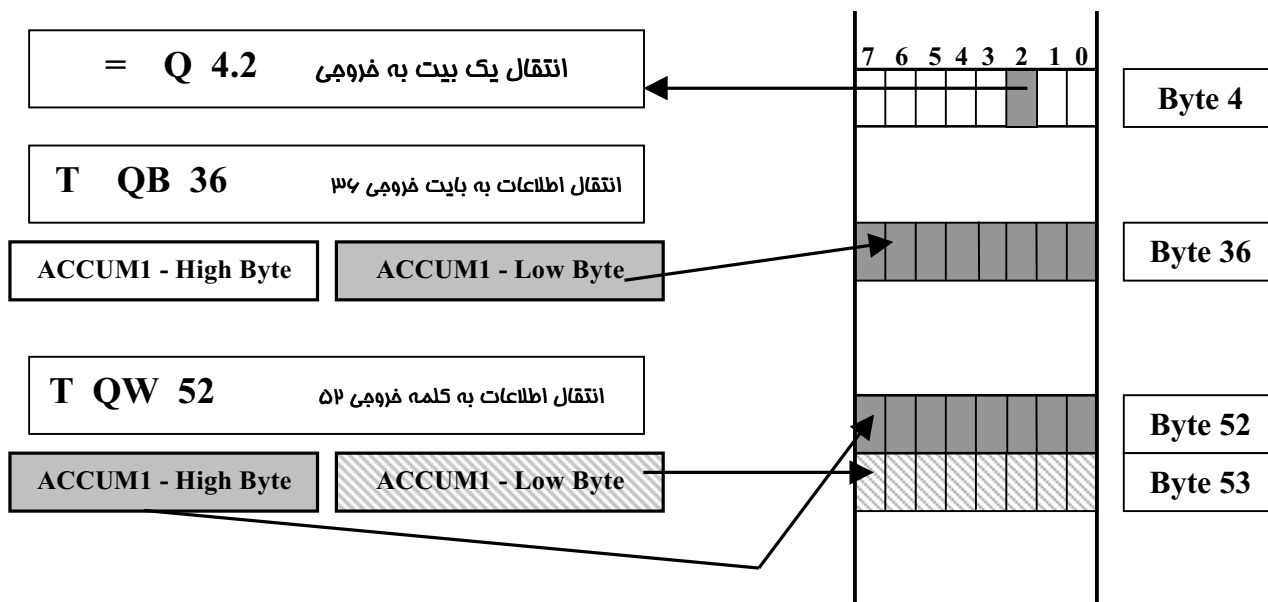


شکل ۹-۱۴: روش های مختلف خواندن اطلاعات از ورودی



با توجه به شکل ۹-۱۴ می توان گفت در خواندن بصورت بیتی از ورودی ها ، از دستور AND استفاده می گردد . در هنگام خواندن یک کلمه از ورودی ابتدا بایت پایین این کلمه ( در مثال فوق بایت ۴۰ ) به بایت پایین آکومولاتور ۱ منتقل می گردد .

مکان تصویر خروجی ( PIQ )



شکل ۹-۱۵: روش های مختلف انتقال اطلاعات به خروجی

توجه: جهت کسب اطلاعات بیشتر در خصوص حالت های مختلف دستورات فوق به ضمیمه ۷ مراجعه نمایید .

#### □ تایمرها یا زمان سنج ها ( Timers ) :

در خانواده S5-115U ، پنج نمونه مختلف از تایمرها در دسترس می باشد . انواع مختلف این تایمرها عبارتند از :

- ۱- Pulse Timer : تایمر پله ای ( SP )
- ۲- Extended Pulse Timer : تایمر پله ای گسترده ( SE )
- ۳- ON-delay Timer : تایمر با تاخیر در وصل ( SD )
- ۴- OFF-delay Timer : تایمر با تاخیر در قطع ( SF )
- ۵- Stored ON-delay Timer : تایمر با تاخیر در وصل پایدار ( SS )

#### بارگذاری زمان تایمر :

به هنگام Start تایمر ، مقدار موجود در آکومولاتور ۱ به عنوان پارامتر یا زمان کاری تایمر در نظر گرفته می شود ، لذا قبل از Start تایمر می بایست زمان مورد نظر را در آکومولاتور ۱ بارگذاری نماییم . زمان مورد نظر می تواند به یکی از فرم های زیر در نظر گرفته شود :

KT : مقادیر ثابت برای زمان تایمر

DW : مقادیر قرارداده شده در کلمات اطلاعاتی ( این کلمات در ادامه بررسی می شوند ) .

IW : کلمه ورودی



QW : کلمه خروجی

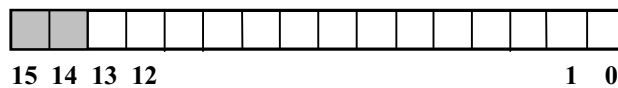
FW : کلمه فلگ

چهار نوع اطلاعات آخر ، می بایست به فرم BCD وارد کردند .

مثال زیر روش بارگذاری یک مقدار ثابت را نمایش می دهد :



پس از بارگذاری مقدار فوق ، در آکومولاتور ۱ مقادیر زیر قرار خواهد گرفت . بیت های ۱۴ و ۱۵ بیانگر ضریب زمانی مربوطه می باشند .



در مثال فوق ضریب بکار رفته می تواند مقادیری مطابق ، جدول ارائه شده در شکل ۹-۱۶ داشته باشد .

00 = 0	01 = 1	10 = 2	11 = 3
0.01 s	0.1 s	1 s	10 s

شکل ۹-۱۶ : ضرایب زمانی مربوط به زمان تایمرها ( اعداد ردیف اول معادل باینری و دهمی ضرایب می باشد ) .

با توجه به مطالب فوق الذکر می توان زمان تخصیص یافته در مثال فوق را بشرح ذیل محاسبه نمود :

$$\text{Time} = 40 * 1s = 40s$$

در صورتی که ضریب در نظر گرفته سایر مقادیر عنوان شده در جدول شکل ۹-۱۶ را شامل گردد ، زمانهای مطلوب عبارتند از :

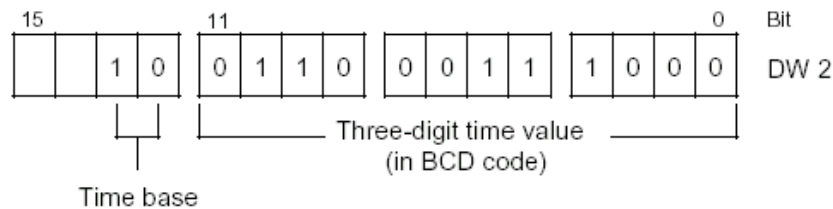
KT 040.0	Time = 40 * 0.01s = 0.4 s
KT 040.1	Time = 40 * 0.1 s = 4 s
KT 040.3	Time = 40 * 10 s = 400 s

قابل ذکر است که رنج خطا ( تولرانس ) در هر یک از ضرایب زمانی فوق الذکر به اندازه ضریب مورد نظر خواهد بود . به عنوان مثال در حالت 040.2 که ضریب زمانی 1s می باشد ، زمان تعریف شده می تواند مقادیر بین ۳۹ تا ۴۰ ثانیه را شامل گردد . ( دقت در این حالت در حد ثانیه ) می باشد . با توجه به این مورد می توان گفت در صورتی که کاربرد نیاز به دقت بیشتری داشته باشد می بایست از رنج زمانی صدم یا دهم ثانیه استفاده نموده و در زمانی که نیاز به رنج زمانی وسیعتر داشته باشد ، می بایست از رنج های ثانیه یا ده ثانیه استفاده نماید . کمترین زمان ممکن قابل اندازه گیری توسط یک تایمر ۰.۰۱ ثانیه و حداکثر زمان قابل اندازه گیری ۹۹۹۰ ثانیه می باشد . در صورت نیاز به زمانهای بیشتر می توان از ترکیب چند تایر استفاده نمود .





جهت بارگذاری یک تایمر از طریق کلمه های اطلاعاتی می بایست یک عدد با حداکثر سه رقم و بفرم BCD در DW بارگذاری نمود. شکل ۹-۱۷ طریقه قرار گرفتن این مقادیر را برای یک زمان برابر ۶۳۸ ثانیه در DW2 را نمایش می دهد. بیت های ۱۴ و ۱۵، ضرب زمان را نمایش می دهند.

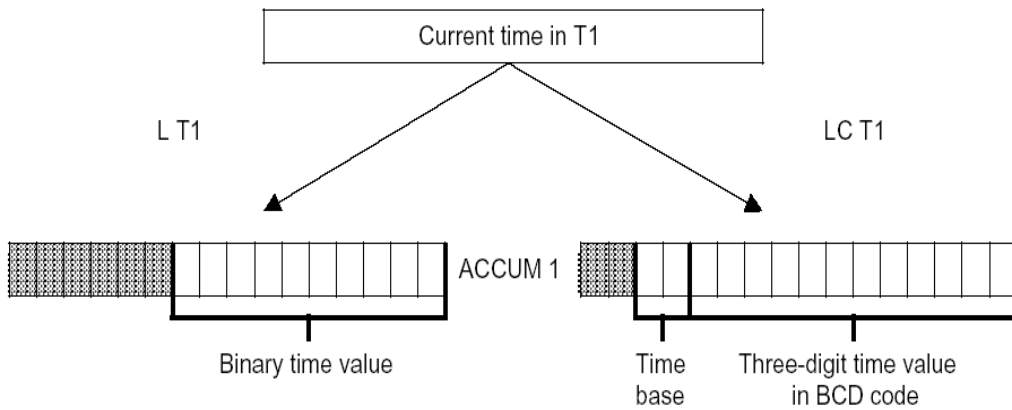


شکل ۹-۱۷: طریقه قرار گرفتن اعداد برای دست یابی به زمان ۶۳۸ ثانیه

توجه: نحوه فراخوانی بلوک های اطلاعاتی و مقداردهی کلمه های اطلاعاتی در ادامه توضیح داده خواهد شد.

**نمونه خواندن زمان جاری تایمرها**

دراکثر PLC ها این قابلیت وجود دارد که بتوان زمان جاری تایمرها را خوانده و به خروجی های PLC انتقال داد. این زمان به دو فرم باینری و BCD قابل خواندن و انتقال به خروجی می باشد. دستور L Tn برای خواندن زمان جاری تایمر n بفرم باینری و دستور LC Tn برای خواندن زمان جاری تایمر n بفرم BCD استفاده می شوند. این مقادیر را می توان برای استفاده نمایشگر های دیجیتال خروجی بکار برد. شکل ۹-۱۸ نحوه قرار گرفتن اطلاعات خوانده شده در آکومولاتور ۱ را نمایش می دهد. قسمت های هاشور خورده با مقدار صفر پر می گردند.



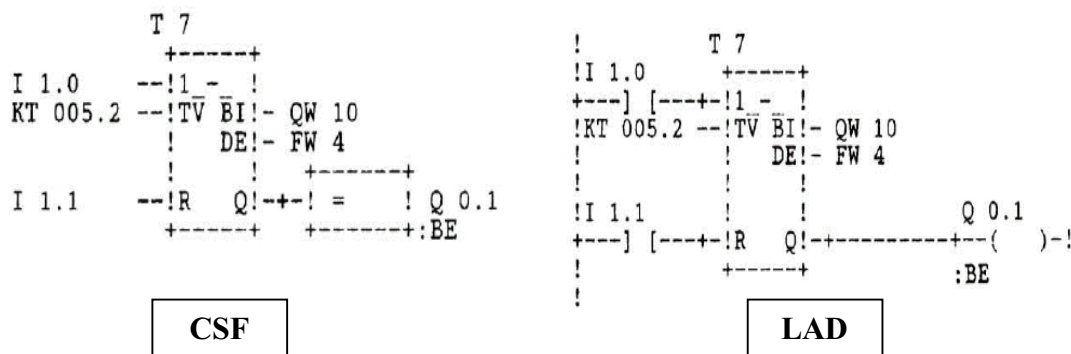
شکل ۹-۱۸: نحوه قرار گرفتن زمان جاری خوانده شده از تایمرها در آکومولاتور یک

در زبان برنامه نویسی S5 برای ایجاد یک تایمر به روش STL از مجموعه جملات زیر استفاده می شود. جهت

A I 1.0	}	ورودی I 1.0 به عنوان Start تایمر استفاده می گردد. این سه خط می بایست به ترتیب ذکر شده استفاده گردند. مقداردهی زمان سنج در خط دوم صورت می پذیرد.
L KT 005.2		
SP T 7		
A I 1.1	}	ورودی I 1.1 جهت Reset تایمر استفاده می گردد.
R T 7		
L T 7	}	خواندن زمان جاری تایمر بصورت باینری و انتقال آن (به خروجی یا فلگ)
T QW 10		
LC T 7		
T FW 4	}	خواندن زمان جاری تایمر بصورت BCD و انتقال آن (به خروجی یا فلگ)
A T 7		
= Q 0.1		خواندن خروجی تایمر و انتقال آن به بیت خروجی 0.1



شکل ۹-۱۹ تایمر را در روشهای LAD و CSF نمایش می دهد .



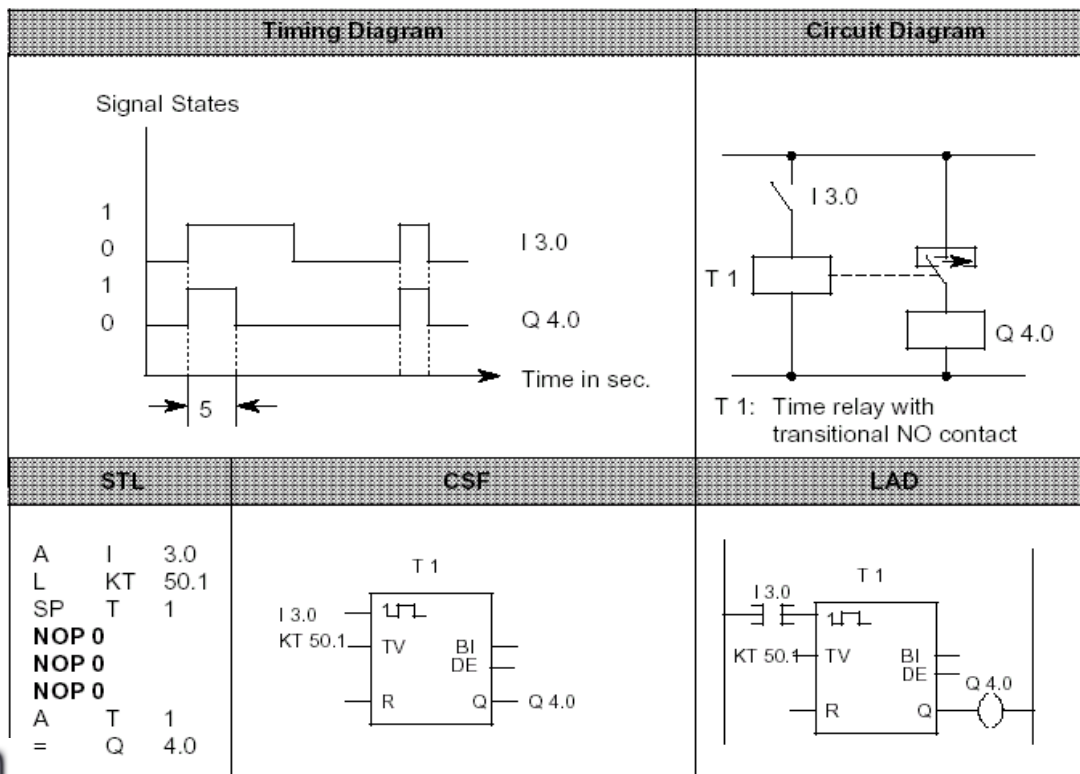
شکل ۹-۱۹: نمایش تایمر به دو روش CSF و LAD

در بکارگیری روش STL، می توان خطوط مورد نظر برای خواندن زمان جاری تایمرها و همچنین ورودی Reset را حذف نمود، در این صورت برای اینکه نرم افزار توانایی تبدیل برنامه نوشته شده STL را به سایر روشها داشته باشد، می بایست به جای خطوط استفاده نشده در برنامه تایمر از دستور NOP 0 استفاده نمود.

معرفی انواع مختلف تایمر

1- Pulse Timer ( SP ) شکل ۹-۲۰

در این مثال با تحریک ورودی Start ( I 3.0 ) تایمر یک که بصورت SP تعریف شده است، پس از بارگذاری مقدار 5s شروع بکار می نماید. نحوه کار این تایمر به این صورت است که با تحریک ورودی، خروجی بلافاصله یک شده و پس از طی مدت زمان تعریف شده برای آن ( در این مثال 5s ) صفر خواهد شد. نکته قابل توجه در این تایمر این است که این تایمر یک زمان سنج حساس به سطح می باشد. به عبارت دیگر با قطع شدن ورودی Start یا تریگر آن خروجی نیز قطع خواهد شد.

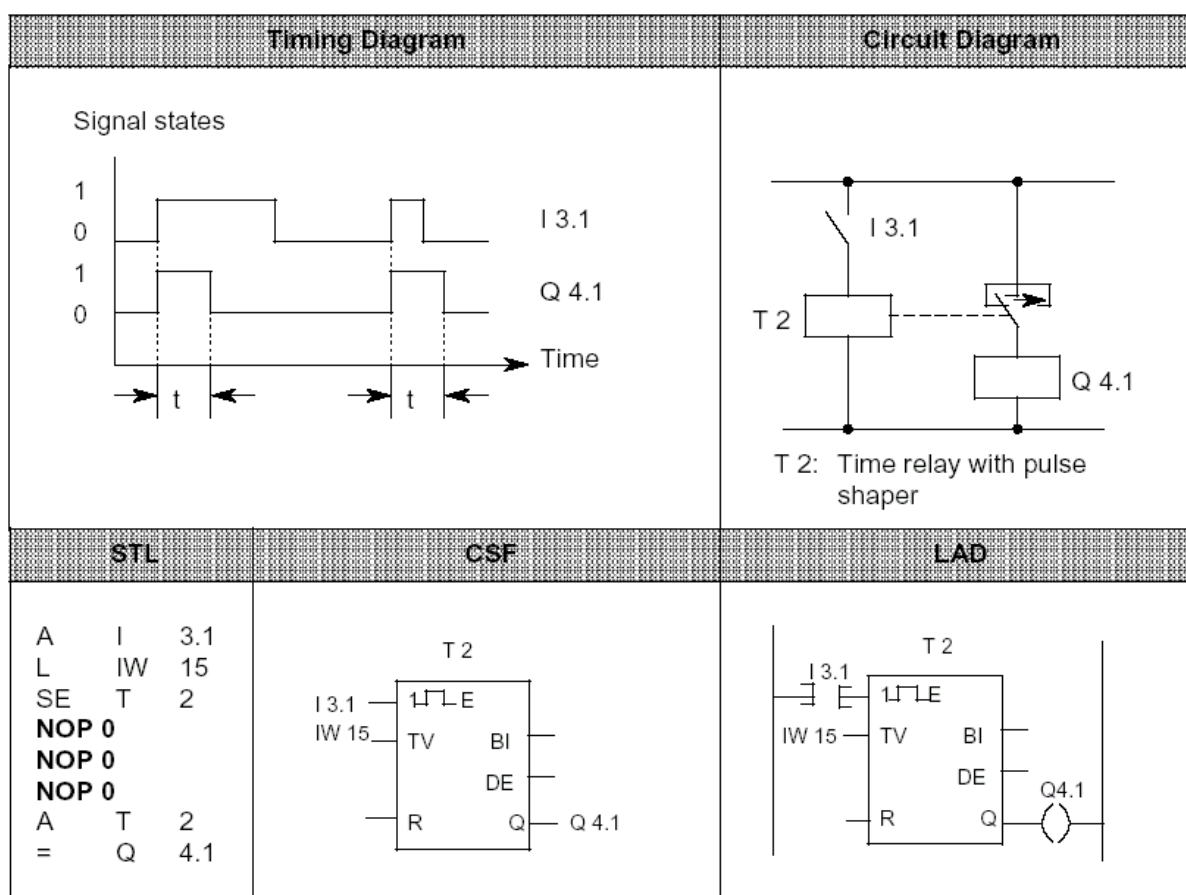


شکل ۹-۲۰: تایمر SP



**۲- Extended Pulse Timer (SE) شکل ۹-۲۱**

این تایمر همانند تایمر SP بوده با این تفاوت که بر خلاف تایمر SP این تایمر حساس به لبه می باشد. به عبارت دیگر با قطع شدن ورودی ترینگر تغییری در عملکرد تایمر ایجاد نخواهد شد و فقط لبه بالا رونده در ورودی ترینگر ( در این مثال I 3.1 ) برای فعال شدن تایمر کفایت می نماید. همانگونه که در قبل اشاره گردید و در برنامه نوشته شده به روش STL ( شکل ۹-۲۱ ) نیز مشاهده می شود، به علت عدم استفاده از خروجی های مقادیر جاری تایمر ( این خروجی های در روش های LAD و CSF با عبارات BI و DE نمایش داده شده اند ) و همچنین عدم استفاده از پایه Reset تایمر، برای ایجاد قابلیت تبدیل برنامه نوشته شده به روش STL به سایر روش ها، توسط نرم افزار مربوطه، از دستور NOP 0 به جای خطوط مورد نظر استفاده شده است. در صورت عدم استفاده از این دستور، برنامه نوشته شده به STL دارای اشکال نبوده ولیکن توسط نرم افزار به سایر روشها تبدیل نخواهد شد.

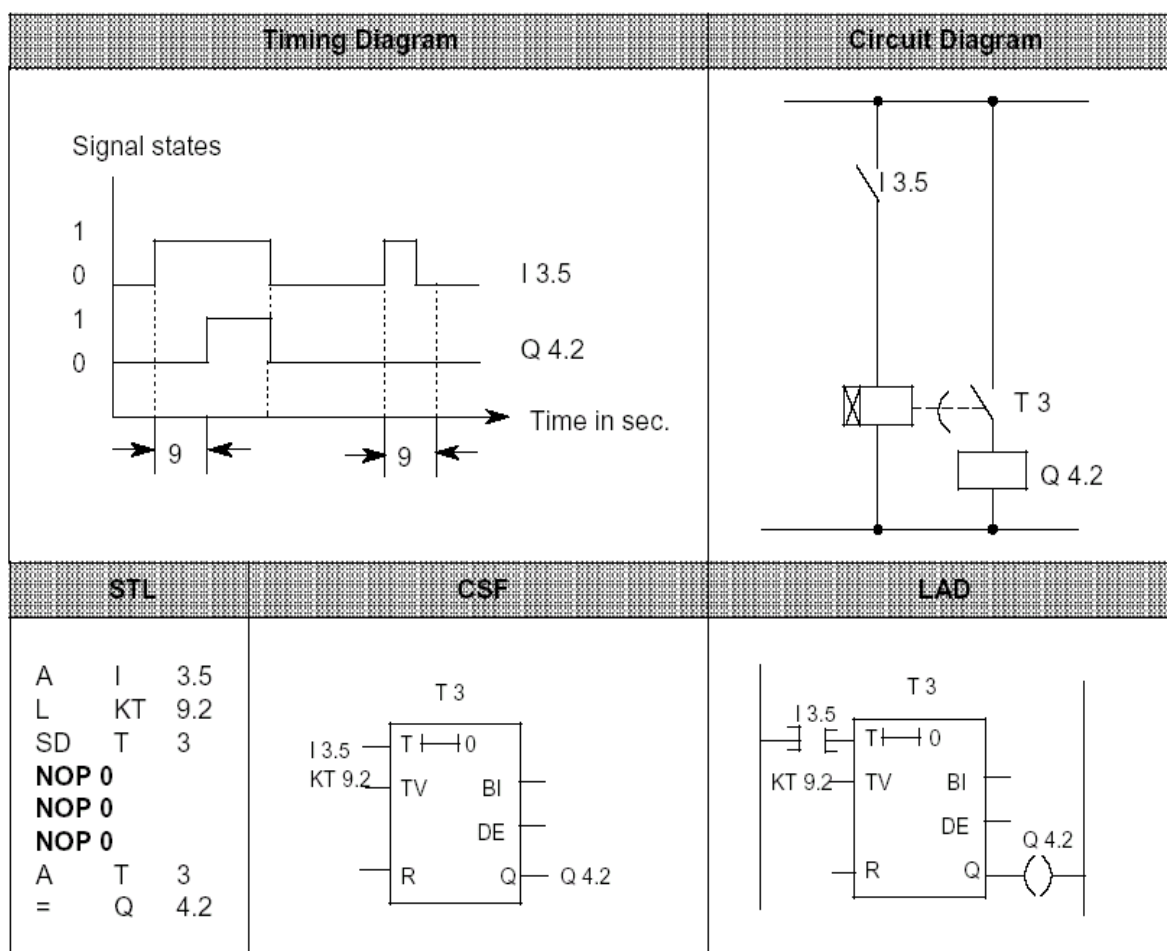


شکل ۹-۲۱: تایمر SE



**۳- On - Delay Timer ( SD ) شکل ۹-۲۲**

در این تایمر با تحریک ورودی تریگر یا Start ( ورودی I 3.5 ) و پس از طی مدت زمان تعریف شده برای تایمر ، خروجی یک خواهد شد . این خروجی مادامی که ورودی تریگر فعال باشد ، در سطح یک باقی می ماند و با قطع تریگر خروجی نیز صفر خواهد شد . در صورتی که مدت زمان یک بودن تریگر کمتر از زمان تعریف شده برای تایمر باشد ، هیچ تغییری در خروجی بوجود نخواهد آمد .

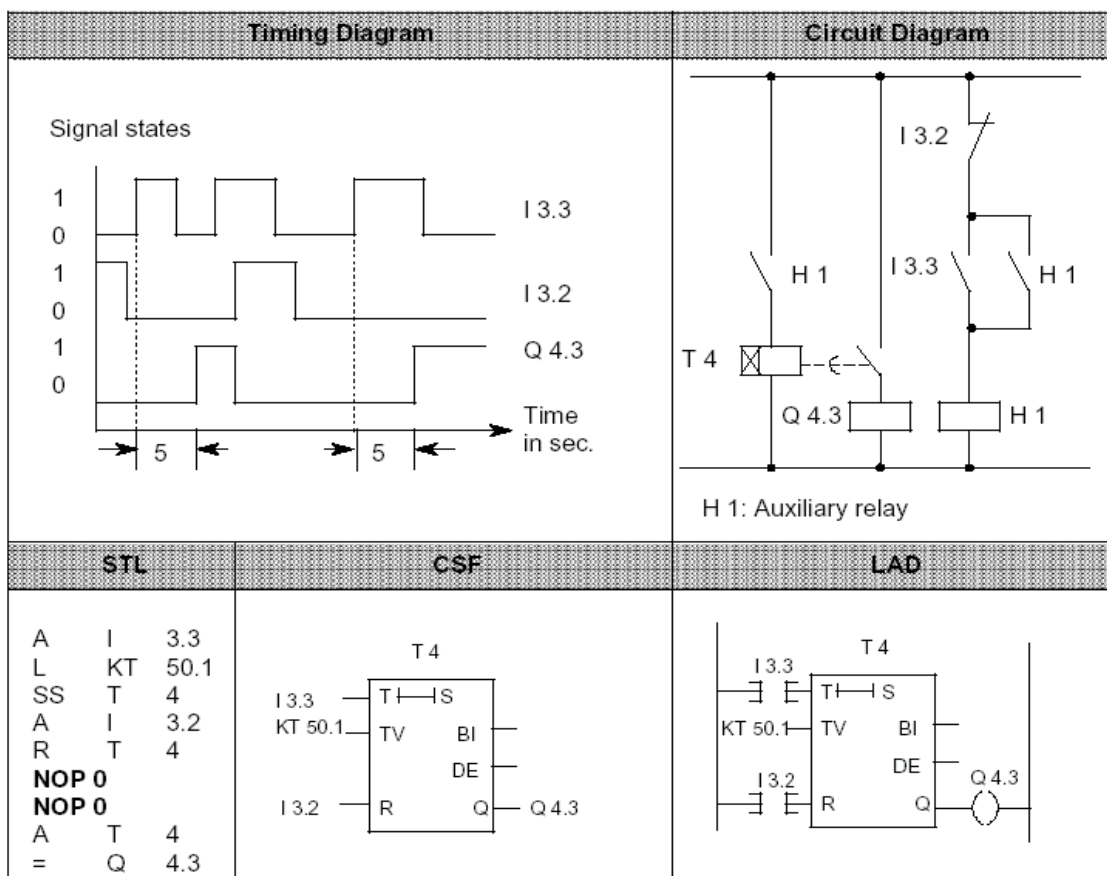


شکل ۹-۲۲: تایمر SD

**۴- Stored On - Delay Timer ( SS ) شکل ۹-۲۳**

این تایمر یک تایمر با تاخیر در وصل پایدار می باشد . در این تایمر یک ماندن سطح ورودی تریگر برای کارکرد مدار ضروری نیست ( حساس به سطح است ) . با استفاده از ورودی Reset در هر زمان می توان خروجی را صفر نمود . پس از یک شدن خروجی بدون توجه به وضعیت ورودی تریگر ، مادامی که ورودی Reset تحریک نشده است ، خروجی یک خواهد ماند .

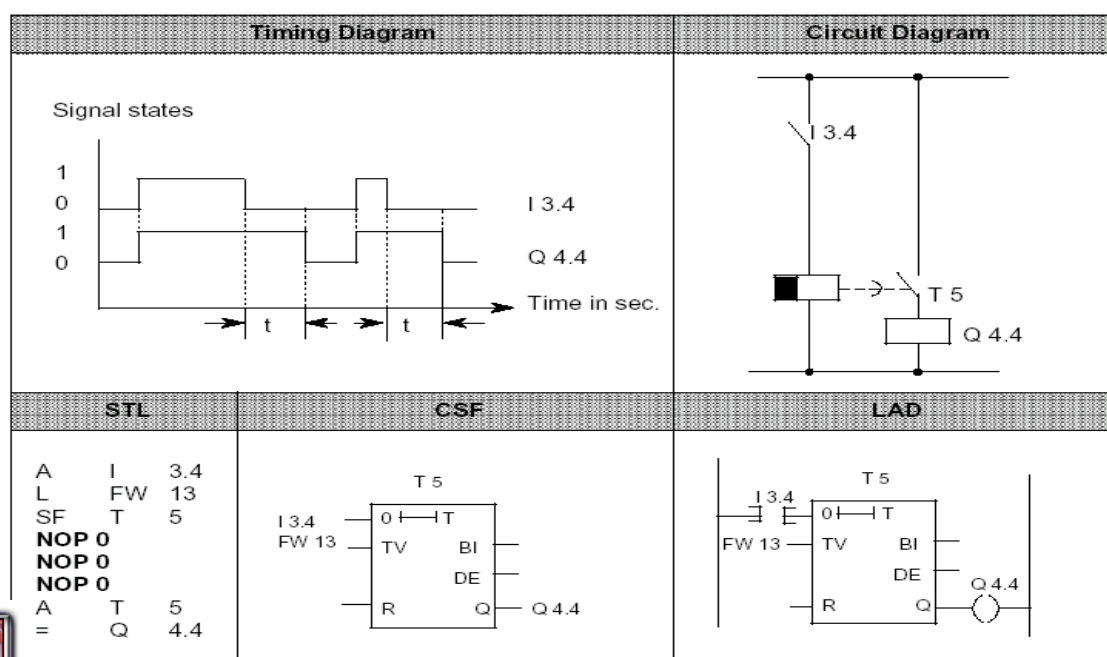




شکل ۹-۲۳: تایمر SS

**Off - Delay Timer ( SF ) ( شکل ۹-۲۴ )**

این تایمر یک تایمر با تاخیر در قطع می باشد. در این تایمر با تحریک ورودی تریگر ، خروجی بلافاصله یک شده و تا زمان قطع شدن تریگر به همین حالت باقی می ماند. با قطع شدن تریگر و ایجاد لبه منفی در این ورودی ، مدت زمان تعریف شده برای تایمر اندازه گیری شده و پس از آن خروجی تایمر صفر خواهد شد .



شکل ۹-۲۴: تایمر SF



□ شمارنده ها ( Counters )

در مواقعی که نیاز است تعداد دفعات وقوع یک رویداد را شمارش نماییم ، می توان از شمارنده ها استفاده نمود . شمارنده موجود در این نوع PLC که شمارنده صعودی و نزولی بوده که مقدار جاری شمارنده نیز در اختیار کاربر قرار می گیرد .

از آنجا که پس از Start شمارنده مقدار موجود در آکومولاتور یک به عنوان حد نهایی شمارش وارد شمارنده می گردد ، لذا می بایست قبل از Start شمارنده مقدار موردنظر را در یکی از فرمهای زیر در آکومولاتور یک بار نمود .

KC : مقادیر ثابت برای شمارنده

DW : مقادیر قرارداده شده در کلمات اطلاعاتی ( این کلمات در ادامه بررسی می شوند ) .

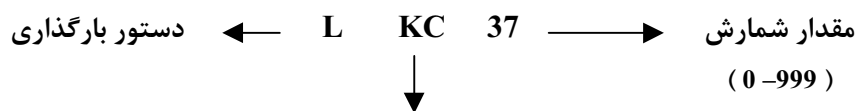
IW : کلمه ورودی

QW : کلمه خروجی

FW : کلمه فلگ

چهار نوع اطلاعات آخر ، می بایست به فرم BCD وارد گردند .

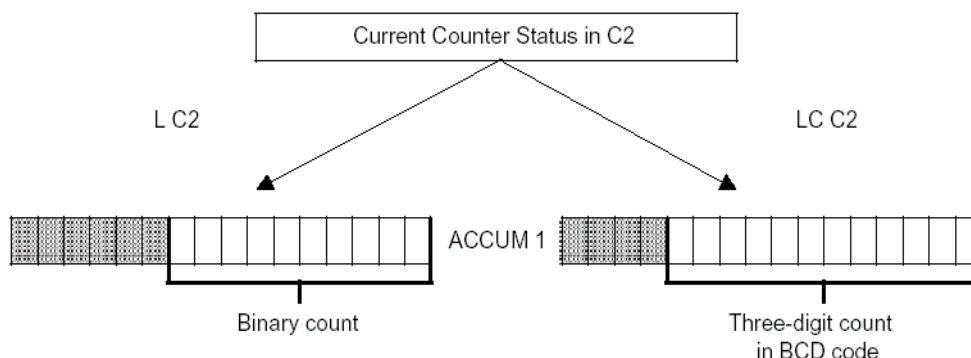
مثال زیر روش بارگذاری یک مقدار ثابت را در شمارنده نمایش می دهد :



↓  
تعریف مقدار ثابت

نمونه خواندن مقدار جاری شمارنده ها

در اکثر PLC ها این قابلیت وجود دارد که بتوان مقدار جاری شمارنده ها را خوانده و به خروجی های PLC انتقال داد . این مقدار به دو فرم باینری و BCD قابل خواندن و انتقال به خروجی می باشد . دستور L Cn برای خواندن زمان جاری شمارنده n بفرم باینری و دستور LC Cn برای خواندن مقدار جاری شمارنده n بفرم BCD استفاده می شوند . این مقادیر را می توان برای استفاده نمایشگر های دیجیتال در خروجی بکار برد . شکل ۹-۲۵ نحوه قرار گرفتن اطلاعات خوانده شده در آکومولاتور ۱ را نمایش می دهد . قسمت های هاشور خورده با مقدار صفر پر می گردند .



شکل ۹-۲۵ : نحوه قرار گرفتن مقدار جاری خوانده شده از شمارنده ها در آکومولاتور یک

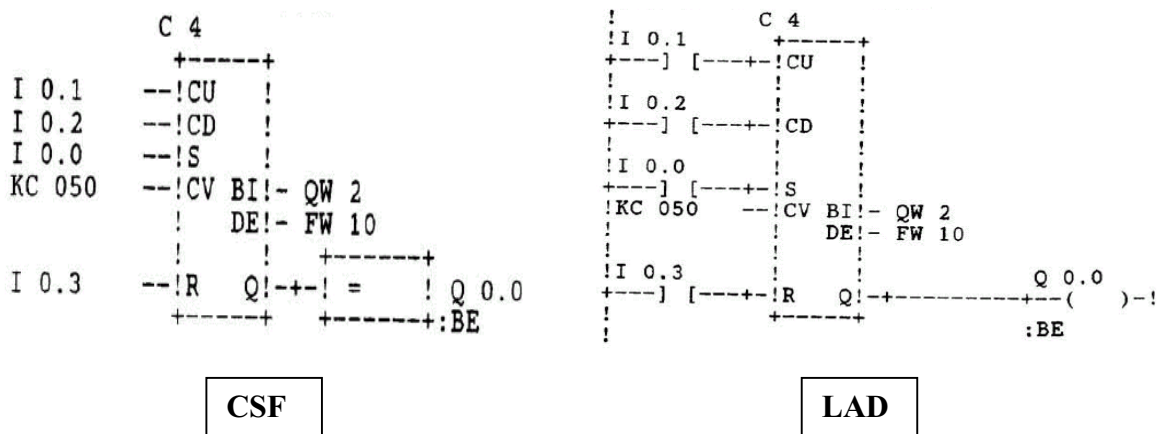


در زبان برنامه نویسی S5 برای ایجاد یک شمارنده به روش STL از مجموعه جملات زیر استفاده می شود .

- |          |   |   |  |
|----------|---|---|--|
| A I 1.0  | } | → | ورودی 1.0 به عنوان پالس ورودی شمارش صعودی استفاده می گردد . هر لبه مثبت      |
| CU C 4   |   |   |  |
| A I 1.1  | } | → | ورودی 1.1 به این ورودی باعث افزایش یک شماره در مقدار داخلی شمارنده می گردد . |
| CD C 4   |   |   |  |
| A I 1.3  | } | → | ورودی I 1.3 جهت Start شمارنده و مقداردهی اولیه استفاده می گردد .             |
| L KC 050 |   |   |  |
| S C 4    | } | → | ورودی I 1.2 جهت Reset شمارنده استفاده می گردد .                              |
| A I 1.2  |   |   |  |
| R C 4    | } | → | فوائدن زمان جاری شمارنده بصورت باینری و انتقال آن ( به فرجوی یا فلگ )        |
| L C 4    |   |   |  |
| T QW 10  | } | → | فوائدن زمان جاری شمارنده بصورت BCD و انتقال آن ( به فرجوی یا فلگ )           |
| LC C 4   |   |   |  |
| T FW 4   | } | → | فوائدن فرجوی شمارنده و انتقال آن به بیت فرجوی 0.1                            |
| A C 4    |   |   |  |
| = Q 0.1  |   |   |  |

در صورت عدم نیاز به هر یک از خطوط برنامه فوق کافیسست ، خطوط مرتبط به هم ( خطوطی که با علامت } مشخص شده اند ) را حذف نماییم ( به غیر از خطوط ۵ ، ۶ و ۷ که در صورت عدم وجود آنها شمارنده تعریف نمی گردد )، به عنوان مثال می توان در صورت عدم نیاز به انجام شمارش نزولی خطوط ۳ و ۴ را حذف نمود . همانگونه که قبلاً گفته شد می توان با استفاده از دستور NOP 0 امکان تبدیل برنامه نوشته شده به روش STL را به سایر روشها ایجاد نمود .

شکل ۹-۲۶ شمارنده را به روش های LAD و CSF نمایش می دهد .

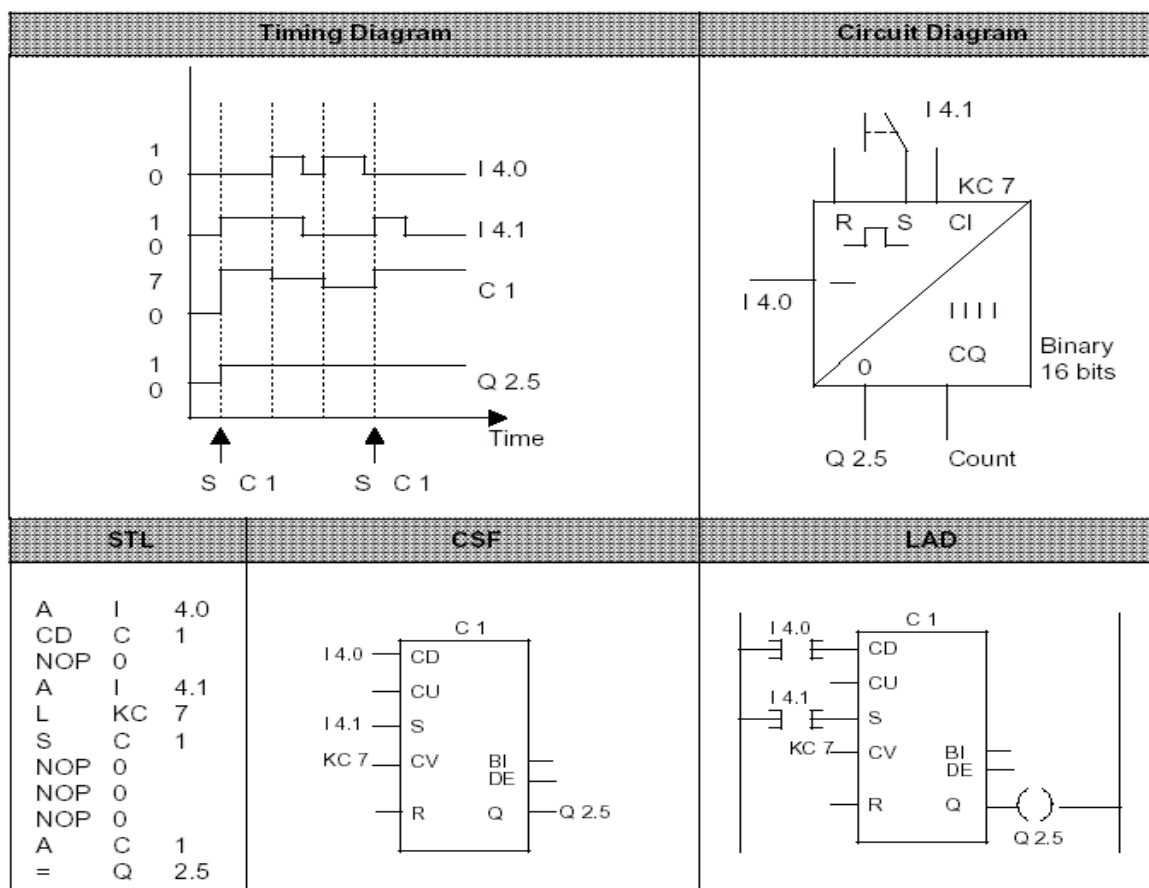


شکل ۹-۲۶: نمایش شمارنده ها در روش های برنامه نویسی CSF و LAD



**۱- فعال نمودن شمارنده و کار در حالت شمارش نزولی ( شکل ۹-۲۷ )**

در این مثال با یک شدن ورودی I 4.1 شمارنده با مقدار نهایی ۷ بار شده و فعال شده و در این حال خروجی شمارنده ( Q 2.5 ) نیز یک می گردد . با هر لبه بالا رونده در ورودی I 4.0 ، یک واحد از مقدار داخلی شمارنده کم می گردد . با رسیدن مقدار داخلی به صفر ، خروجی نیز صفر خواهد شد . بدلیل عدم نیاز به استفاده از سایر خطوط برنامه در روش STL ، به جای مقادیر مورد نظر از دستور NOP 0 استفاده شده است .



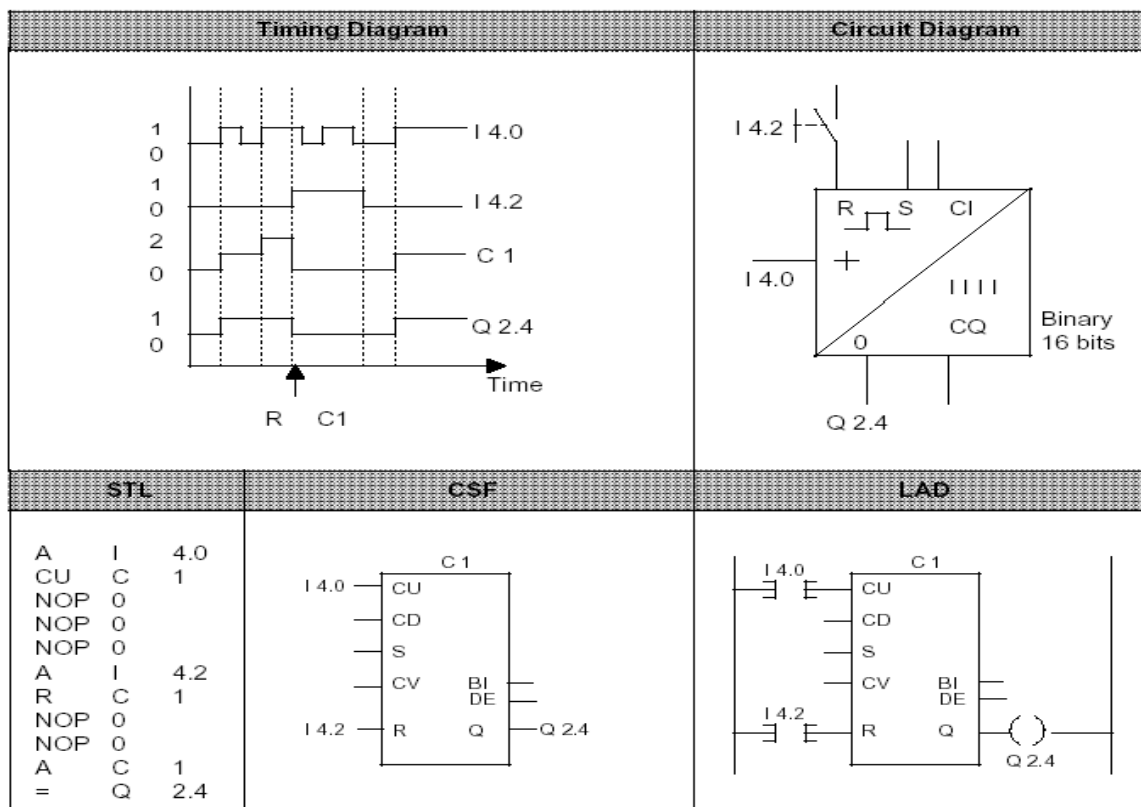
شکل ۹-۲۷: فعال نمودن شمارنده و کار در حالت شمارش نزولی

**۲- فعال نمودن شمارنده و کار در حالت شمارش صعودی ( شکل ۹-۲۸ )**

در مثال ارائه شده در شکل ۹-۲۸ که یک شمارنده را در حالت شمارش صعودی نمایش می دهد ، با ایجاد هر لبه مثبت در ورودی I4.0 مقدار داخلی شمارنده یک واحد افزایش می یابد . در حالت شمارش صعودی جهت شمارش یک تعداد واقعه می توان مقدار داخلی شمارنده را بصورت باینری یا BCD خوانده و آنرا با استفاده از دستورات مقایسه که در ادامه معرفی خواهند گشت مورد ارزیابی قرار داد . در حالت شمارش صعودی اگر شمارنده مقدار دهی اولیه گردد با فعال کردن کلید مربوط به بار شدن اطلاعات در شمارنده مقدار تعریف شده در شمارنده Load شده و شمارش از آن عدد دنبال می گردد . در طی مدت زمان شمارش در صورت فعال نشدن ورودی Reset خروجی بیتی شمارنده یک خواهد ماند .







شکل ۹-۲۸: بیان نحوه کار یک شمارنده در حالت شمارش صعودی

□ **مقایسه کننده ها ( Comparator's )**

این عملگرها مقادیر موجود در دو آکومولاتور را مقایسه کرده و در اثر نتیجه مقایسه بیت RLO را مقداردهی می نماید. این دستورات مقدار داخلی آکومولاتورها را تحت تاثیر قرار نمی دهند.

Operation	Operand	Meaning
! = F		Compare for "equal to" در این عملگر در صورتی که $AC1 = AC2$ باشد، بیت $RLO=1$ می گردد.
> < F		Compare for "not equal to" در این عملگر در صورتی که مقدار دو آکومولاتور نامساوی باشد، بیت $RLO=1$ می گردد.
> F		Compare for "greater than" در این عملگر در صورتی که $AC2 > AC1$ باشد، بیت $RLO=1$ می گردد.
> = F		Compare for "greater than or equal to" در این عملگر در صورتی که $AC2 \geq AC1$ باشد، بیت $RLO=1$ می گردد.
< F		Compare for "less than" در این عملگر در صورتی که $AC2 < AC1$ باشد، بیت $RLO=1$ می گردد.
< = F		Compare for "less than or equal to" در این عملگر در صورتی که $AC2 \leq AC1$ باشد، بیت $RLO=1$ می گردد.



مثال ارائه شده در شکل ۹-۳۰ دو عدد موجود در بایت های ورودی ۱۹ و ۲۰ را مقایسه کرده و در صورت برابر بودن آنها مقدار خروجی Q3.0 را یک خواهد کرد. باید دقت نمود در بارگذاری اطلاعات، ابتدا مقدار اول در AC1 قرار گرفته و پس از بارگذاری مقدار دوم، مقدار جدید در AC1 و مقدار قبلی AC1 در AC2 قرار می گیرد (به دستورات LOAD که قبلاً توضیح داده شده اسات مراجعه گردد). لذا توجه به این نکته ضروری است که در دستورات مقایسه برای بزرگتر یا کوچکتر بودن اولین عددی که بار گذاری می گردد در AC2 قرار می گیرد. مثال ارائه شده در شکل ۹-۳۱ این مطلب را بیان می دارد. در این مثال در صورتی که عدد موجود در بایت ورودی ۳۰ بزرگتر از عدد موجود در بایت ورودی ۳۸ باشد، خروجی Q 4.2 یک خواهد شد.

Circuit Diagram	STL	CSF/LAD
	<pre>L IB 19 L IB 20 ! = F = Q 3.0</pre>	

شکل ۹-۳۰: مثالی از دستورات مقایسه کننده

L	IB30
L	IB38
>	F
=	Q4.2

شکل ۹-۳۱: مثالی از دستورات مقایسه کننده

### □ دستورات ریاضی (Arithmetic Operation's)

این دستورات شامل دستورات جمع و تفریق بوده که بر روی دو آکومولاتور عمل می نمایند.

Operation	Operand	Meaning
+ F		<b>Addition</b> در دستور جمع دو آکومولاتور با هم جمع شده و نتیجه در آکومولاتور ۱ قرار می گیرد
- F		<b>Subtraction</b> در دستور تفریق مقدار موجود در آکومولاتور یک از آکومولاتور دو کم شده و نتیجه در آکومولاتور یک قرار می گیرد.

شکل ۹-۳۲: معرفی دستورات عمل های ریاضی



L C3 در مثال روبرو مقدار جاری موجود در شمارنده یک از مقدار جاری شمارنده سه کم شده و  
 L C1 نتیجه به کلمه خروجی هشت منتقل می گردد. ( QW8 = C3 - C1 )  
 -F  
 T QW8

**دستورات فراخوانی بلوک ها ( Block Call Operations )**

این دستورات برای ترتیب بندی اجرای بلوک ها و سازماندهی پرش به آنها مورد استفاده قرار می گیرند. با استفاده از دستورات پرش می توان از یک بلوک به بلوک دیگر پرش نمود.  
**توجه : انواع مختلف بلوک های برنامه نویسی S5 در ادامه مورد بررسی قرار خواهند گرفت .**

Operation	Operand	Meaning
JU		<b>Jump unconditionally</b> دستور پرش غیر شرطی : با بکارگیری این دستور می توان بدون شرط ( بررسی بیت RLO ) به بلوک مورد نظر پرش نمود .
JC		<b>Jump conditionally</b> دستور پرش شرطی : با بکارگیری این دستور می توان در صورت یک بودن بیت RLO به بلوک مورد نظر پرش نمود . در صورت صفر بودن R LO پرش انجام نمی گردد .
ID	OB PB FB SB	<b>Parameter</b> 0 to 255 * 0 to 255 0 to 255 0 to 255 انواع بلوک هایی که می توان با دستورات JU و JC به آنها پرش نمود .
C		<b>Call a data block</b> دستور فراخوانی بلوک های اطلاعاتی : با بکارگیری این دستور می توان یک بلوک اطلاعاتی را فعال کرده و از اطلاعات ذخیره شده در DW های آن استفاده نمود .
G		<b>Generate and delete a data block **</b> ایجاد و پاک کردن یک بلوک اطلاعاتی
ID	DB	<b>Parameter</b> 2 to 255 ***
BE		<b>Block end</b> دستور مشخص کردن انتهای بلوک : این دستور انتهای یک بلوک را نمایش می دهد . با رسیدن خط اجرای برنامه به این دستور ، بلوک جاری پایان می پذیرد .
BEU		<b>Block end, unconditional</b> دستور پایان دادن اجرای بلوک بصورت غیر شرطی : با استفاده از این دستور می توان بدون هیچ شرطی در وسط برنامه نوشته شده ، یک بلوک برنامه را پایان داد .
BEC		<b>Block end, conditional</b> دستور پایان دادن اجرای بلوک بصورت شرطی : با استفاده از این دستور می توان به شرط یک بودن بیت RLO در وسط برنامه نوشته شده ، یک بلوک برنامه را پایان داد .

شکل ۹-۳۳ : دستورات فراخوانی بلوک های برنامه



مثالهای زیر نحوه استفاده از دستورات فراخوانی بلوک را نمایش می دهند .

### Unconditional Block Call "JU"

Program Sequence	STL	Explanation
	<pre>         .         .         .         .         .         JU FB26         .     </pre>	<p>با استفاده از دستور FB26 JU، بلوک عملگر FB26 از داخل بلوک برنامه PB63 فراخوانی شده است . پس از اجرای دستور خط اجرای برنامه به FB26 منتقل شده و ادامه برنامه از آنجا پیگیری می گردد . این عمل تا اجرای دستور BE در FB26 ادامه می یابد .</p>

### Conditional Block Call "JC"

Program Sequence	STL	Explanation
	<pre>         .         .         S F 1.0         A I 31.7         JC FB63         .     </pre>	<p>با استفاده از دستور JC FB63، بلوک عملگر FB63 به شرط یک بودن بیت ورودی 31.7 در اثر آن بیت RLO نیز یک خواهد شد ، از داخل بلوک برنامه PB10 فراخوانی می گردد. پس از اجرای دستور خط اجرای برنامه به FB63 منتقل شده و ادامه برنامه از آنجا پیگیری می گردد .</p>

### Call a Data Block "C DB"

Program Sequence	STL	Explanation
	<pre>         C DB10         L DW 1         .         .         C DB20         T DW 3     </pre>	<p>در این مثال ابتدا با بکارگیری دستور C DB10 بلوک اطلاعاتی ۱۰ فعال شده و سپس با استفاده از دستور بارگذاری مقدار کلمه اطلاعاتی یک از DB 10 در آکومولاتور قرار می گیرد . با دستور C خط اجرای برنامه منتقل نشده و تنها بلوک اطلاعاتی مورد نظر فعال می گردد . این بلوک تا بکار گیری یک دستور جدید Call معتبر خواهد بود .</p>



آموزشگاه برق و الکترونیک صنعتی اندیشه سازان صنعت مجری برگزاری دوره های آموزشی از مقدماتی تا پیشرفته در زمینه های برق ، الکترونیک ، اتوماسیون صنعتی ، ابزار دقیق و کامپیوتر - با مجوز رسمی از سازمان آموزش فنی و حرفه ای کشور و وزارت کار و امور اجتماعی - اعطای مدرک با کد بین المللی I.L.O

Program Sequence	STL	Explanation
	<p>·</p> <p>·</p> <p>·</p> <p>·</p> <p>·</p> <p>BE</p>	<p>در این مثال پس از اینکه خط اجرای برنامه با استفاده از دستور پرش از OB1 به PB3 منتقل گردید ، در انتهای بلوک PB3 و با رسیدن به دستور BE ، خط اجرای برنامه به خط بعد از دستور پرش در OB1 باز خواهد گشت و ادامه برنامه از آن نقطه پیگیری می گردد .</p>

Program Sequence	STL	Explanation
	<p>·</p> <p>·</p> <p>·</p> <p>JC=</p> <p>BEU</p> <p>·</p> <p>·</p> <p>·</p> <p>BE</p>	<p>دستور JC= که در ادامه مورد بررسی قرار خواهد گرفت ، با توجه به نتیجه موجود در بیت RLO ( که شرایط موجود در برنامه را بیان می دارد . ) به یک آدرس مشخص شده برای آن پرش می نماید . در این مثال در صورت برقراری شرایط قبل از دستور JC= ادامه برنامه که شامل خطوط پس از دستور BEU تا دستور BE می باشد ادامه یافته و در غیر اینصورت با استفاده از دستور BEU ادامه برنامه در FB21 انجام نشده و خط اجرای برنامه به PB8 باز خواهد گشت . با این کار یک شرط برای ادامه اجرای برنامه ایجاد گردید .</p>



### انواع بلوک های برنامه نویسی در PLC خانواده S5

در نوشتن برنامه کنترل مربوط به فرآیندهای گسترده، جهت جلوگیری از پراکندگی و در نتیجه از هم گسیختگی برنامه از روش برنامه نویسی ساخت یافته (Structured Programming) استفاده می شود. در این روش یک برنامه کنترل وسیع پس از تقسیم بندی به قسمت های کوچکتر در بلوک های مخصوصی نوشته می شوند و سپس ارتباطات مابین قسمت های مختلف برنامه از طریق یک برنامه سازماندهی انجام می پذیرد. برنامه های سازماندهی شده، امکان عیب یابی برنامه را آسانتر کرده و به دلیل دسته بندی شدن برنامه به قسمت های مجزا، دست یابی به اجزاء مختلف برنامه در مدت زمان بسیار کوتاهی فراهم می گردد.

در دسته بندی ساختار برنامه در پروسورهای جدیدتر، کل ساختار برنامه در سه قالب بدنه اصلی برنامه (Main)، زیربرنامه ها (Subroutine) و روال های وقفه (Interrupt routine) دسته بندی می گردد. در این روش دسته بندی زیر برنامه ها جهت انجام عملیات تکراری و جهت کاهش حجم برنامه نوشته شده مورد استفاده قرار می گیرند. وقفه ها نیز برای انجام عملیات با اولویت بالا تعریف شده و روتین مربوط به هر وقفه در زمان ایجاد وقفه مربوطه فعال و اجرا می شود.

در PLC های خانواده S5، روش دسته بندی کمی قدیمی تر بوده و کل ساختار برنامه در قالب پنج نوع بلوک برنامه نویسی تعریف می گردد.

#### ♦ بلوک های برنامه PB (Program Block)

PB ها شامل دستورات استفاده کننده جهت کنترل پروسه مربوطه می باشند. برنامه نویس پروسه کنترل خود را به قسمت های کوچکتر تقسیم کرده و هر قسمت را در یکی از بلوک های برنامه قرار می دهد. ۲۵۶ بلوک برنامه از PB0 تا PB255 قابل دسترسی می باشد. هر بلوک برنامه خود دارای ۲۵۶ قسمت کوچکتر به نام Segment می باشد که از Segment ها برای دسته بندی قسمت های کوچکتر در برنامه استفاده می گردد. بدین ترتیب کوچکترین واحد در یک پروسه به راحتی از طریق دانستن بلوک برنامه و در نهایت Segment مورد نظر قابل ردیابی است. در پایان هر بلوک برنامه یک دستور BE نمایانگر انتهای بلوک مربوطه می باشد. در مقایسه با روش نوین برنامه نویسی بلوک های برنامه قسمتی از Main برنامه را تشکیل می دهند.

#### ♦ بلوک های ترتیبی SB (Sequence Block)

یکی از موارد بسیار مهم در کنترل یک پروسه صنعتی، استارت سیستم در حال توقف می باشد. در این موارد فعال شدن هر قسمت از پروسه با توجه به شرایط کاری خاصی انجام می پذیرد. به عنوان مثال در یک خط تولید خودکار شرط راه اندازی مرحله دوم، صحت عملکرد مرحله اول و در نتیجه آن ایجاد سیگنالهای مورد نظر جهت ارسال به مرحله دوم جهت استارت می باشد، لذا در این سیستم حتماً قبل از استارت مرحله دوم می بایست، مرحله اول انجام شده و فعالیت آن به پایان برسد.

در PLC خانواده S5 یک نوع از بلوک های برنامه نویسی به نام SB ها برای این منظور تعریف شده اند. در این بلوک ها تنها به شرط انجام و صحت تعاریف موجود در بلوک های با شماره پایین تر بلوک با شماره بالاتر اجرا خواهد شد.



### ♦ بلوک های عملگر FB (Function Block)

در کنترل برخی فرآیندها، گاه لازم است توابعی به صورت مداوم و تکراری مورد استفاده قرار گیرند. به عنوان مثال ممکن است عمل ضرب دو عدد باینری در طی یک پروسه مرتباً مورد استفاده گیرد. در چنین مواردی اگر بنا به تکرار برنامه ضرب در مکانهای مختلف سطح برنامه کنترل باشد، حجم برنامه نوشته بسیار زیاد شده و در نهایت بررسی، عیب یابی و یا تغییر برنامه با مشکل مواجه خواهد شد. برای رفع این مشکل در برنامه نویسی مدرن از زیر برنامه ها استفاده می شود. در PLC مورد بحث نیز این مشکل با بکارگیری بلوک های عملگر یا FB ها حل می گردد. بلوک های عملگر می توانند بصورت عباراتی همراه با متغیر قابل تعریف، استفاده شده و یا تنها یک عمل ثابت را بدون پارامترهای قابل تنظیم بیان نمایند. در این نوع PLC، ۲۵۶ بلوک عملگر از FB0 تا FB255، قابل تعریف می باشد. نکته قابل توجه در این مورد رعایت این نکته می باشد که برخی از این بلوک ها توسط شرکت سازنده و جهت انجام برخی از پروسه های مشخص از قبل تعریف شده و در اختیار کاربر قرار می گیرد. به این بلوک ها در اصطلاح بلوک های انتسابی (Assignable FB) گفته می شود.

### ♦ بلوک های اطلاعاتی DB (Data Block)

بلوک های اطلاعاتی برای ذخیره سازی داده های مورد استفاده در برنامه کاربر مورد استفاده قرار می گیرند. این اطلاعات که می توانند به فرمت های مختلف مورد استفاده قرار گیرند، شامل پیغام ها، آلارم ها و ... می باشند، که می توانند در طی پروسه مورد استفاده قرار گرفته یا بر روی صفحات نمایشگر یا پانل های اپراتور به نمایش در آیند. هر بلوک اطلاعاتی شامل تعدادی کلمه های اطلاعاتی می باشد که به آنها Data Word گفته شده و با علامت اختصاری DW نمایش داده می شوند.

### ♦ بلوک های سازماندهی OB (Organization Block)

از آنجا که در یک برنامه ساخت یافته نیازمند تعدادی بلوک مادر برای سازمان دادن به برنامه کاربر و تعریف شرایط مختلف برای اجرای برنامه ها می باشیم، لذا در این نوع PLC تعدادی بلوک سازماندهی، جهت تعریف شرایط مختلف کاری در نظر گرفته شده اند. بلوک OB1 بدنه اصلی برنامه کاربر را شامل می گردد. به عبارت دیگر اولین خط از برنامه کاربر که می بایست مورد پردازش قرار گیرد، در این بلوک نوشته می شود. از OB1 می توان برای سازماندهی برنامه اصلی کاربر استفاده کرده و تمام پرش ها و شرط های لازم برای انجام سایر مراحل کاری را در این بلوک تعریف نمود.

سایر بلوک های سازماندهی را می توان برای انجام برخی عملیات در شرایط خاص مورد استفاده قرار داد، به عنوان مثال برنامه نوشته شده در OB34 در زمان خرابی باطری پشتیبان (Backup) اجرا خواهد شد، لذا می توان دستورات و برنامه های مورد نیاز که می بایست در زمان خرابی باطری، هشدارهای لازم را جهت آگاه سازی اپراتور دستگاه صادر نماید را در این بلوک نوشت. فراخوانی این بلوک ها به برنامه کاربر مرتبط نبوده و تنها با توجه به شرایط کاری سیستم اجرا می گردند. جدول ارائه شده در شکل ۹-۳۴ انواع بلوک های سازماندهی مهم را نمایش می دهد.



OB No	Function	OB Integrated In CPU			
		941	942	943	944
<b>OB must be user-programmed and be called by the operating system</b>					
OB1	Cyclic program scanning				
<b>OBs for interrupt-driven and time-controlled program scanning</b>					
OB2	Interrupt A: Digital input module -434 and IP generate interrupt				
OB3	Interrupt B: IP generates interrupt				
OB4	Interrupt C: IP generates interrupt				
OB5	Interrupt D: IP generates interrupt				
OB6	Interrupt generated by internal timers				
OB10	Time-controlled				
OB11	program scanning				
OB12					
OB13	(variable in each case: 10 msec. to 10 min.)				
<b>OBs for controlling restart characteristics</b>					
OB21	Manual switch on (STOP RUN)				
OB22	Automatic switch on when power is restored				
<b>OBs for handling programming errors and PLC faults</b>					
OB19	When a block is called which has not been loaded				
OB23	Time-out during individual access to the S5 bus (e.g. LPB, LIR, etc.)				
OB24	Time-out during update of the process image and the interprocessor communication flags				
OB27	Substitution error				
OB32	Transfer errors in DB or with GDB operation				
OB34	Battery failure				
<b>OBs which offer operating functions</b>					
OB31	Scan time triggering				
OB160	Programmable time loop				
OB251	PID algorithm				
OB254	Read in process I/O image				
OB255	Output process I/O image				

شکل ۹-۳۴: بلوک های سازماندهی



آموزشگاه برق و الکترونیک صنعتی اندیشه سازان صنعت مجری برگزاری دوره های آموزشی از مقدماتی تا پیشرفته در زمینه های برق ، الکترونیک ، اتوماسیون صنعتی ، ابزار دقیق و کامپیوتر- با مجوز رسمی از سازمان آموزش فنی و حرفه ای کشور و وزارت کار و امور اجتماعی - اعطای مدرک با کد بین المللی I.L.O

مشهد مقدسی - ۷۶۱۸۲۸۶ - ۷۶۴۴۰۷۸ - ۰۵۱۱



**OB1:** این بلوک توسط کاربر نوشته شده و توسط سیستم عامل فراخوانی می گردد .  
**OB2:** این بلوک در صورت ایجاد وقفه توسط ماژول ورودی دیجیتال 437 یا ماژول های هوشمند دارای پردازنده اجرا می گردد .

توجه : وقفه ها یا عملیات با اولویت بالا در ادامه مورد بررسی قرار خواهند گرفت .  
**OB3 تا OB5:** در صورت ایجاد وقفه توسط ماژول های هوشمند اجرا خواهند گشت .  
**OB6:** در صورت ایجاد وقفه توسط تایمرهای درونی اجرا می گردد .  
**OB10 تا OB13:** این بلوک ها در صورت فعال سازی وقفه های زمانی در مدت زمانهای مشخص و به صورت مکرر اجرا خواهند گشت .

**OB21:** در زمانی که کلید روشن کردن دستی موجود بر روی ماژول منبع تغذیه زده می شود ( کلید انتخاب حالت RUN و STOP ) این ماژول اجرا می شود ، لذا می توان برخی از عملیات مورد نظر از جمله اعلام هشدار به اپراتور قبل از راه انداز دستگاه یا انجام عملیات ابتدایی قبل از راه اندازی کلی سیستم را در این بلوک نوشت که با RUN کردن سیستم ابتدا این مراحل انجام شده و سپس برنامه اصلی که در **OB1** سازماندهی گردیده است اجرا شود .

**OB22:** همانند **OB21** این بلوک نیز زمانی اجرا می گردد که کلید Power موجود بر روی کارت منبع تغذیه زده شود . با زدن این کلید سیستم تغذیه شده و اجازه استارت پیدا می کند .

#### بلوک های سازماندهی مربوط به قطاها

**OB19:** این بلوک زمانی اجرا می گرد که یک بلوک برنامه نویسی در طی برنامه فراخوانی شده ولی بارگذاری نمی گردد . به عنوان مثال ممکن است یک بلوک عملگر در طی یک بلوک برنامه صدا زده شود در حالی که بلوک عملگر مربوط بدلیلی موجود نباشد .

**OB23:** در صورت عدم امکان دسترسی به BUS اطلاعات در طی یک برنامه ، این بلوک اجرا می گردد . به عنوان مثال ارتباط اطلاعاتی یک کارت با Data Bus وجود نداشته باشد .

**OB24:** همانطور که در گذشته اشاره شد ، ورودی ها و خروجی ها با استفاده از دو منطقه ویژه به نامهای PI و PIQ با برنامه کاربر مرتبط می باشند که اطلاعات موجود در این مناطق در ابتدا و انتهای سیکل برنامه به روز ( Update ) می گردند ، حال اگر بدلیلی امکان به روز سازی این اطلاعات موجود نباشد ، بلوک اطلاعاتی **OB24** اجرا می گردد . همچنین این بلوک در زمان وجود اشکال در به روز سازی پرچم های ارتباطی که در ادامه مورد بررسی قرار خواهند گرفت نیز اجرا می گردد .

**OB 27:** در زمان وجود اشکال در جابجایی اطلاعات اجرا می شود .

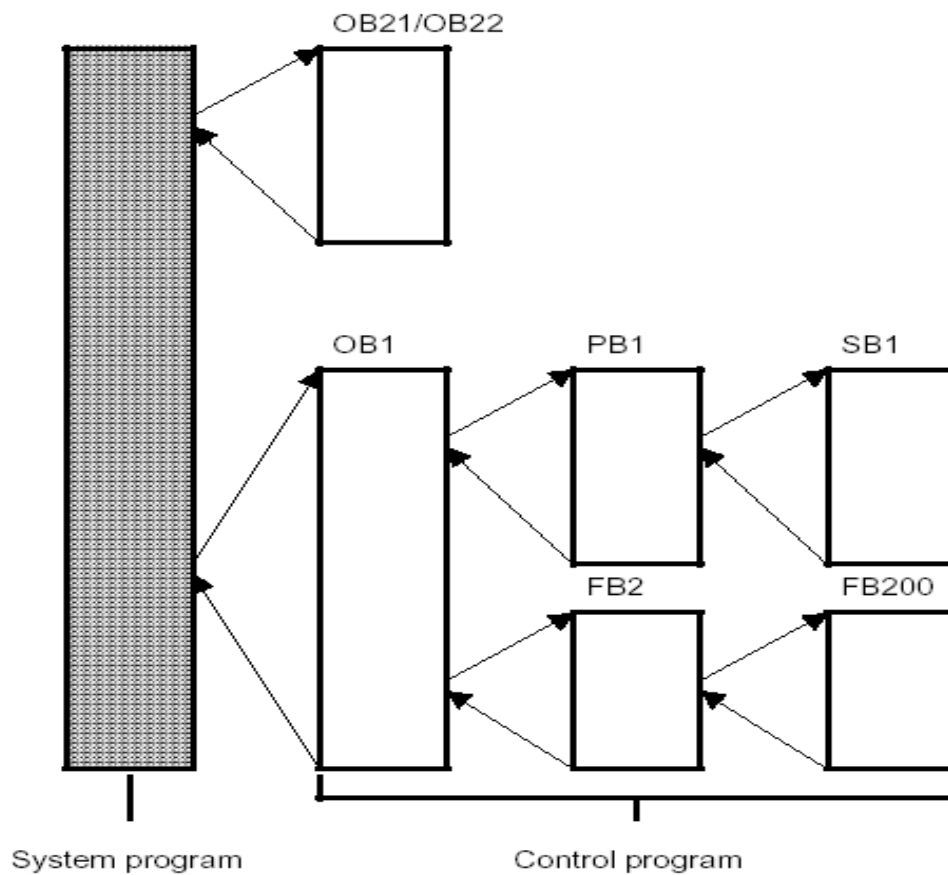
**OB32:** این بلوک به هنگام وجود اشکال در انتقال اطلاعات به یک بلوک اطلاعاتی و کلمه اطلاعاتی مربوط به آن فعال می گردد .

**OB34:** در صورتی که بدلیلی ولتاژ باتری پشتیبان به کمتر از ۳,۴ ولت افت نماید ، این بلوک اجرا می گردد .

□ سایر بلوک های سازماندهی معرفی شده در شکل ۹-۳۴ جهت انجام برقی پروسه های فاص از قبیل فواندن ورودی ، انتقال اطلاعات به فربجی ، ملقه های کنترلی ID P و ... مورد استفاده قرار می گیرند که در ادامه بررسی فوهند شد .



شکل ۹-۳۵ نحوه سازماندهی و اجرای بلوک های مختلف را توسط CPU نمایش می دهد.



شکل ۹-۳۵: نحوه اجرای بلوک های مختلف توسط CPU



### نحوه برنامه نویسی بلوک های اطلاعاتی :

انواع مختلف اطلاعات قابل ذخیره سازی در بلوک های اطلاعاتی عبارتند از :

۱- الگوهای بیتی ( Bit Pattern ) : عبارتند از اطلاعاتی که برای انجام برخی پروسه های خاص در برنامه کنترل استفاده می شود . به عنوان مثال می توان یک الگوی بیتی را یک سری اطلاعات بصورت بیتی یا کلمه ای در مبنای مختلف دانست که برای فعال کردن خروجی ها به یک نحو خاص ، مورد استفاده قرار می گیرند .

۲- اعداد در مبنای ده ، شانزده و دو : این اطلاعات می تواند ، زمان مربوط به یک عمل خاص یا اطلاعات حاصل از عملیات ریاضی مختلف باشد که در بلوک های اطلاعاتی ذخیره شده و در مواقع لازم مورد استفاده قرار می گیرند .

۳- کاراکترهای اسکی : جهت نمایش برخی هشدارها و پیغام ها ، اطلاعات مربوط به آنها ، بفرم اسکی در بلوک های اطلاعاتی ذخیره شده و در مواقع لزوم از آنها استفاده می شود .

بلوک های اطلاعاتی را می توان با تعیین شماره بلوک از DB2 تا DB255 تعریف نمود . DB0 برای استفاده سیستم عامل و DB1 برای استفاده عملگرهای داخلی و همچنین پرچم های ارتباط بین پروسسوری ( که در ادامه مورد بررسی قرار خواهند گرفت ) ذخیره شده اند . اطلاعات در این بلوک ها بصورت کلمه ای ذخیره می شوند ، که در صورت کمتر بودن تعداد بیت های مربوطه بیت های بالاتر با صفر پر می گردند . در هر بلوک اطلاعاتی تا ۲۰۴۲ کلمه قابل جاسازی است . دسترسی به DW0 تا DW255 با دستورات DW L و T DW انجام شده و دسترسی به DW256 تا DW2042 با دستورات LIR ، TIR و TNB صورت می پذیرد . ( این دستورات در هنگام معرفی دستورات تکمیلی بررسی می گردند ) .  
 شکل ۹-۳۶ نحوه برنامه نویسی یک بلوک اطلاعاتی و نحوه ذخیره سازی اطلاعات در کلمه های مربوطه را نمایش می دهد .

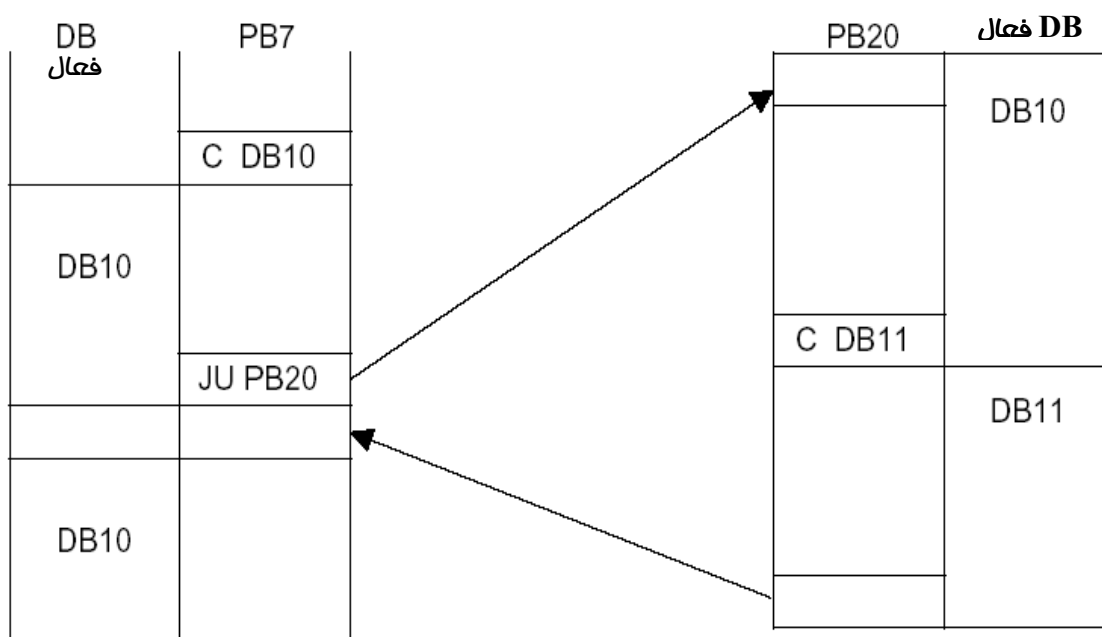
Input	Values stored
0000 : KH = A13C	DW0 A13C
0001 : KT = 100.2	DW1 2100
0003 : KF = +21874	DW2 5572

شکل ۹-۳۶ : نحوه برنامه نویسی بلوک های اطلاعاتی



**نحوه پردازش برنامه های حاوی بلوک های اطلاعاتی**

- در طی برنامه اصلی بلوک های اطلاعاتی با استفاده از فرمان X DB C که در آن X بیانگر شماره بلوک مربوطه می باشد ، فعال می گردد .
- مادامی که یک بلوک جدید فراخوانی نشده باشد ، بلوک اطلاعاتی قبلی فعال بوده و تمام کلمه های اطلاعاتی مربوطه از این بلوک برداشته می شود .
- در صورت پرش به بلوک های برنامه دیگر ، بلوک اطلاعاتی قبلی فعال می ماند .
- در صورتی که پس از پرش به یک بلوک برنامه جدید ، بلوک اطلاعاتی تغییر نماید ، به محض بازگشت به بلوک برنامه قبلی ، بلوک اطلاعاتی که قبلاً در آن فعال بوده است ، معتبر می گردد . این موضوع در شکل ۹-۳۷ توضیح داده شده است . در این شکل در PB7 و با استفاده از دستور C DB10 بلوک اطلاعاتی ۱۰ معتبر گردیده است . پس از پرش به PB20 ، بلوک اطلاعاتی قبل فعال مانده است . در این حال در PB20 بلوک اطلاعاتی DB11 فعال شده است ، ولیکن با بازگشت از PB20 به PB7 ، همان بلوک اطلاعاتی قبل ( DB10 ) که از قبل فعال بوده است ، مجدداً معتبر می گردد .



شکل ۹-۳۷: بررسی بلوک های اطلاعاتی فعال در طی دستورات پرش



**نحوه بیان فرمت اطلاعات در بلوک های اطلاعاتی :**

- 1) اعداد مبنای ۱۶ - ( 0000 - FFFF ) 16 بیت : KH
- 2) اعداد مبنای ۱۰ - ( -32768 +32767 ) 16 بیت : KF
- 3) اعداد ثابت زمان سنج ها - ( 001.0 - 999.3 ) 14 بیت : KT
- 4) اعداد ثابت شمارنده ها - ( 000 - 999 ) 12 بیت : KC
- 5) اعداد مبنای ۲ - ( 1111 1111 1111 1111 - 0000 0000 0000 0000 ) 16 بیت : KM
- 6) جهت نمایش اعداد ممیز شناور و نیز اعداد بسیار بزرگ - ( دو کلمه ای ) 32 بیت : KG
- 7) جهت ذخیره سازی اطلاعات بفرم اسکی : KS

این اعداد به شکل BCD تعریف می گردند .

در ذخیره سازی اطلاعات بفرم اسکی ، هر کارکتر در یک بایت از حافظه ذخیره می گردد .

مثال : برنامه ای بنویسید که به عدد 0000H که در یک سطر DB وجود دارد در هر سیکل یک واحد اضافه نماید .

PB30	DB27	
C	DB27	با فعال کردن DB27 ، در هر بار اجرای
L	DW0	PB30 ، اعداد موجود در DW1 و DW0
L	DW1	پس از فراخوانی با هم جمع شده و
+F		نتیجه در DW0 و همچنین QW2 قرار
T	DW0	می گیرد . و چون DW1=1 است پس
T	QW2	اطلاعات موجود در DW0 هر بار یک
BE		واحد افزایش می یابد .

**نحوه برنامه نویسی بلوک های عملگر :**

هر بلوک عملیاتی از دو قسمت بدنه بلوک ( Block Body ) و سرخط بلوک ( Block Header ) تشکیل می گردد . بدنه بلوک شامل برنامه اصلی نوشته شده توسط کاربر بوده و سر خط بلوک شامل نام و سایر مشخصات مربوط به بلوک عملگر از جمله تعاریف مربوط به پارامترها می باشد . جهت ایجاد یک بلوک عملیاتی ابتدا ، نام بلوک را وارد نموده و سپس در صورت نیاز به تعریف متغیرها پرداخته می شود . در تعریف متغیرها در مقابل اعلان DECL نام متغیر را که حداکثر می تواند یک کلمه با ۴ کاراکتر را شامل شود ، وارد می نمایم . سپس نوع پارامتر تعریف شده را مشخص می نمایم .

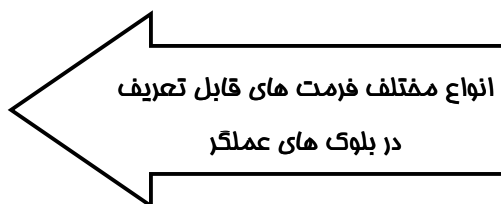
- I : پارامترهای ورودی
- Q : پارامترهای خروجی
- D : اطلاعات
- B : بلوک های مختلف
- T : زمان سنج ها
- C : شمارنده ها

انواع مختلف پارامترهای قابل تعریف در بلوک های عملگر

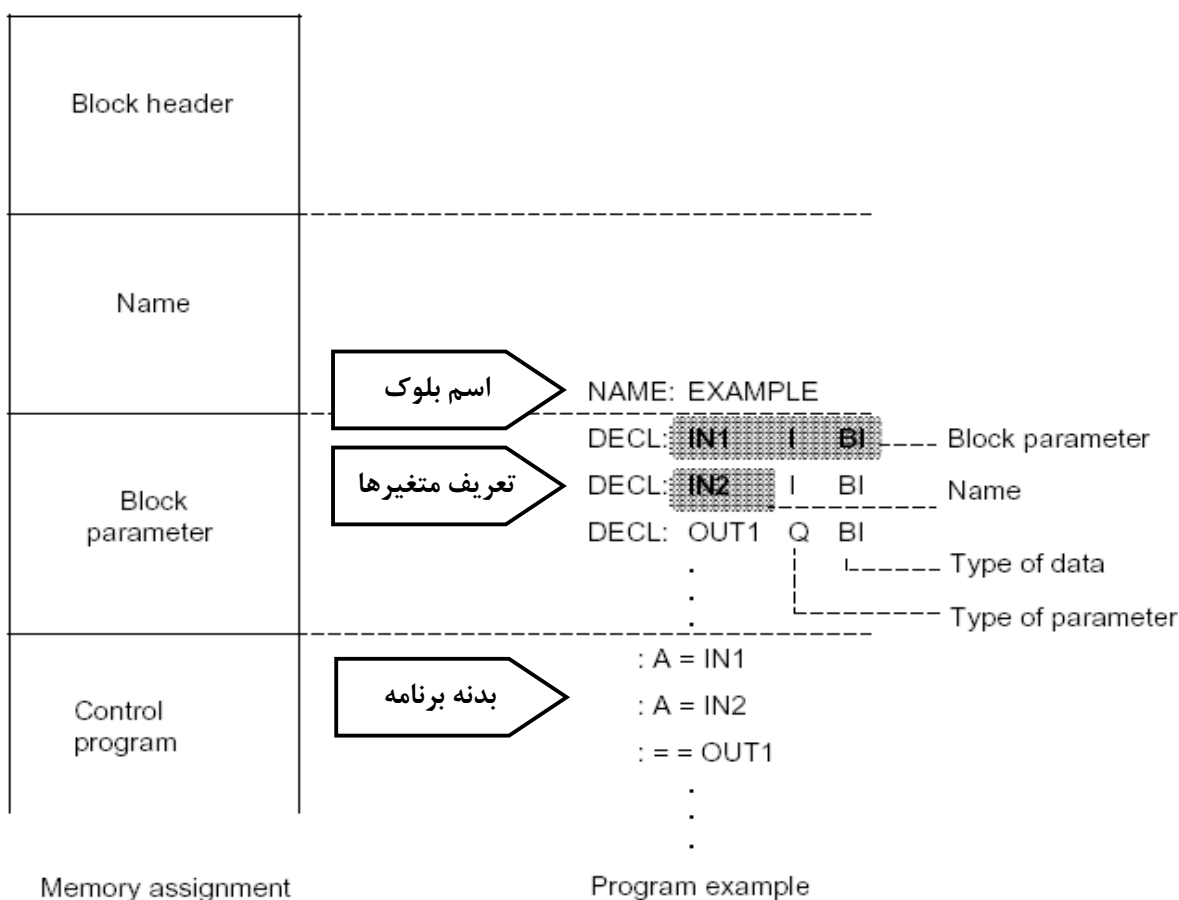


اطلاعات قابل تعریف در پارمتر به فرمت های زیر قابل تعریف می باشند :

- BI : اطلاعات بصورت بیتی
- BY: اطلاعات بصورت بایتی
- W : اطلاعات بصورت کلمه ای
- K : مقادیر ثابت



مثال ارائه شده در شکل ۹-۳۸ ، نحوه تعریف یک بلوک عملگر را بیان می نماید .

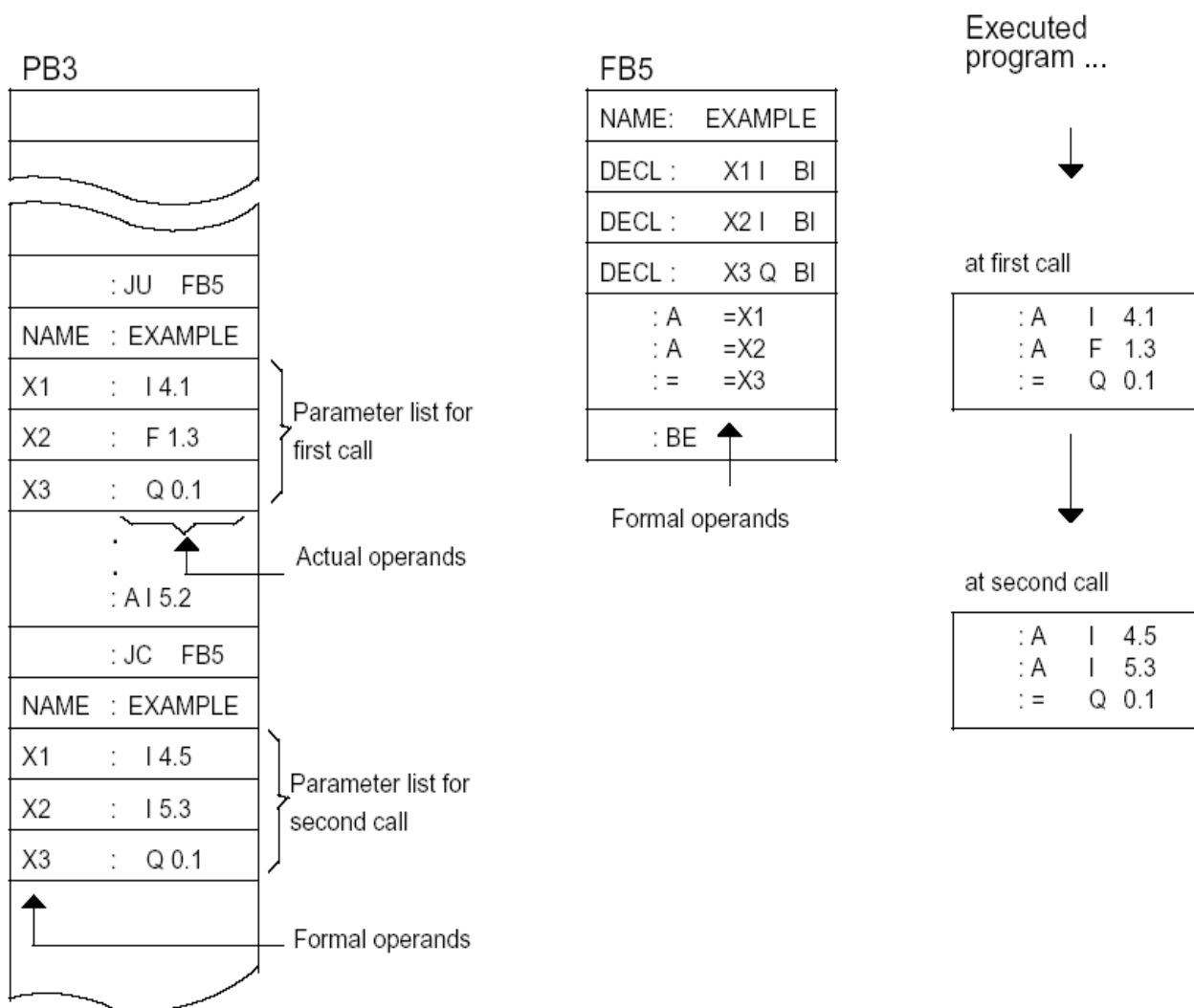


شکل ۹-۳۸ : تعریف یک بلوک عملگر



مثال ارائه شده در شکل ۹-۳۹ نحوه تعریف و فراخوانی یک بلوک عملگر را بطور کامل ارائه می نماید. در این مثال در PB3 و با استفاده از دستور FB5 JU بلوک عملگر EXAMPLE با نام EXAMPLE و با سه متغیر X1 الی X3 فراخوانی شده است. از آنجا که متغیرهای مورد نظر بنا بر تعریف در بلوک عملگر مربوطه، بصورت بیتی و با عنوان ورودی و خروجی (X1 و X2 ورودی بیتی و X3 خروجی بیتی) تعریف شده اند، لذا در PB3 به آنها مقادیری متناسب با تعریف انجام شده نسبت می دهیم. به عنوان مثال به X1 مقدار I4.1 منتسب شده است. مجدداً یادآوری می گردد که با استفاده از بلوک های عملگر و تعریف متغیر در آنها می توان یک کار مشخص را با فراخوانی بلوک عملگر مربوطه و انتساب متغیرهای موجود به داده های مرونظر به تعداد دفعات زیاد تکرار نموده بدون آنکه برنامه مورد نظر را تکرار نماییم. در سمت راست مثال زیر نیز برنامه اجرا شده در اثر اجرای PB3 و FB5 نمایش داده شده است.

**توجه:** به نحوه تعریف و بکارگیری متغیرها در FB ها که در مثال زیر ارائه شده است دقت نمایید.



شکل ۹-۳۹: مثالی از تعریف و فراخوانی بلوک های عملگر



### دستورالعمل های تکمیلی :

برخلاف دستورات اصلی که در تمام بلوک ها مورد استفاده قرار می گیرند ، دستورات تکمیلی تنها بصورت STL و در بلوک های عملگر قابل استفاده می باشند .

### دستور فعال کننده FR :

با استفاده از این دستورالعمل می توان یک تایمر را مجدداً استارت کرده و یک شمارنده را دوباره فعال نمود .

Operation	Operand	Meaning
FR		Enable a Timer/Counter
ID	T C	Parameter 0 to 127 0 to 127

Example	STL	Explanation
ورودی I2.5 تایمر T2 را فعال نموده و این تایمر خروجی Q4.2 را برای مدت زمان تعریف شده یک می نماید .	A I 2.5 L KT 5.3 * SE T 2 A T 2 = Q 4.2 . . . .	اگر خروجی Q3.4 یک شده و در نتیجه یک لبه مثبت در بیت RLO ایجاد نماید ، مادامی که بیت I2.5 یک می باشد ، تایمر بصورت مکرر استارت می گردد . بنابراین Q4.2 نیز بطور مکرر و با توجه به عملیات تعریف شده در برنامه ، یک و صفر می گردد .
اگر خروجی Q3.4 مکرراً صفر و یک شود ، تایمر نیز می تواند بصورت مکرر راه اندازی گردد .	A Q 3.4 FR T 2 BE	

شکل ۹-۴۰: دستور فعال کننده تایمر و شمارنده





**دستورات تست بیت :**

این دستورات بیت های دیجیتال را یک به یک SCAN کرده و آنها را تحت تاثیر قرار می دهد . این دستورات معمولاً در شروع عملیات منطقی مورد استفاده قرار می گیرند .

Operation	Operand	Meaning
TB		Test a bit for signal state "1" تست کردن یک بیت برای وضعیت یک منطقی
TBN		Test a bit for signal state "0" تست کردن یک بیت برای وضعیت صفر منطقی
SU		Set a bit unconditionally یک کردن بدون شرط یک بیت
RU		Reset a bit unconditionally صفر کردن بدون شرط یک بیت
ID	<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;">             ↑ T C D RS<sup>1</sup> </div> <div>             ↑ Parameter 0 to 127.15 0 to 127.15 0 to 255.15 0 to 255.15           </div> </div>	

شکل ۹-۴۱: دستورات تست بیت

برنامه ارائه شده در شکل ۹-۴۲ مثالی از نحوه استفاده از دستورات تست بیت را بیان می نماید .



Example	STL	Explanation
<p>در این مثال یک سنسور فتو الکترونیک که تعداد یک کالا را شمارش می نماید به ورودی I2.0 متصل شده است .</p> <p>در صورتی که تعداد کالای شمارش شده کوچکتر از ۱۰۰ باشد بلوک FB5 اجرا شده و در صورت بزرگتر یا مساوی بودن عدد شمارش شده با ۱۰۰ بلوک FB6 اجرا می گردد .</p> <p>پس از شمارش هر ۸۰۰ قطعه نیز شمارنده Reset می گردد .</p>	<pre> C   DB 10 A   I   2.0 CU  C   10 A   I   3.0 L   KC  0 S   C   10 O   I   4.0 O   F   5.2 R   C   10 LC  C   10 T   DW 12  TBN D 12.8  JC  FB  5  TB  D 12.8  JC  FB  6  TB  D 12.11  =   F   5.2                 </pre>	<p>در این مثال ابتدا بلوک اطلاعاتی I۰ فعال می گردد . I2.0 به عنوان پایه شمارش صعودی شمارنده تعریف گردیده است . با فعال نمودن I3.0 علاوه بر بار نمودن مقدار اولیه صفر شمارنده نیز استارت می گردد .</p> <p>شمارنده در دو صورت صفر می گردد : ۱- فشرد شدن I4.0 -۲ فعال شدن F 5.2</p> <p>به طور دائم مقدار شمارش شده داخلی بصورت BCD خوانده شده و در DW12 ریخته می شود . با بررسی بیت ۸ از کلمه اطلاعاتی I۲ می توان ۱۰۰ بودن عدد موجود در شمارنده را مورد بررسی قرار داد .</p>
<p>در این مثال یک سنسور فتو الکترونیک که تعداد یک کالا را شمارش می نماید به ورودی I10.0 متصل شده است .</p> <p>پس از هر ۲۵۶ قطعه شمارنده صفر شده و شمارش مجدداً ادامه می یابد .</p>	<pre> :A   I   10.0 :CU  C   20 :A   I   11.0 :L   KC  0 :S   C   20  :TB  C   20.8  :JC  = FULL :BEU  FULL:RU C 20.8 :BE                 </pre>	<p>I 10.0 : پایه شمارش صعودی</p> <p>I 11.0 : فعال کننده شمارنده و بارگذاری مقدار اولیه صفر در شمارنده</p> <p>در صورت یک بودن بیت ۸ از شمارنده ۲۰ و با توجه به اینکه مقدار شمارش شده داخلی شمارنده بصورت باینری می باشد ، پرش به آدرس مشخص شده با Lable با عنوان Full انجام گرفته و در آنجا بیت 8 از شمارنده 20 صفر می گردد ، در غیر اینصورت ادامه برنامه با استفاده از دستور BEU پایان یافته و اجرای برنامه از ابتدا ادامه می یابد .</p>

شکل ۹-۴۲ : دو مثال از بکارگیری دستورات تست بیت



**دستورات انجام عملیات دیجیتال :**

این دستورات محتویات دو آکومولاتور را بیت به بیت ترکیب کرده و نتیجه را در آکومولاتور یک قرار می دهند .

Operation	Operand	Meaning
AW		انجام عمل AND منطقی بصورت بیت به بیت
OW		انجام عمل OR منطقی بصورت بیت به بیت
XOW		انجام عمل XOR منطقی بصورت بیت به بیت

شکل ۹-۴۳ : دستورات منطقی

STL	Explanation
L IW 92	خواندن کلمه ورودی ۹۲ و قرار دادن آن در آکومولاتور ۱
L KH 00FF	قرار دادن مقدار ثابت 00FF بصورت هگزا دسیمال در آکومولاتور ۱ و در نتیجه انتقال مقدار قبلی موجود در آکومولاتور یک به آکومولاتور ۲
AW	انجام عمل AND منطقی بصورت بیت به بیت بین آکومولاتور یک و دو و قرار دادن نتیجه در آکومولاتور یک
T QW 82	انتقال نتیجه عملیات AND از آکومولاتور یک به کلمه خروجی ۸۲
Numeric Example	
<p>ACCUM 2</p> <p>IW 92</p> <p>15 0</p> <p>0 1 1 1 0 0 0 1 1 0 1 1 0 1 0 1</p>	نمایش بیتی اطلاعات در طی انجام برنامه فوق با انجام این برنامه ۸ بیت مرتبه بالای موجود در کلمه ورودی یک ، بی اثر گردیده اند . از این عمل می توان برای از بین بردن اثر بیت های غیر ضروری در طی یک پروسه استفاده نمود . به این عمل پوشاندن یا Mask کردن بیت ها گفته می شود .
<p>ACCUM 1</p> <p>KH 00FF</p> <p>AND</p> <p>0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1</p>	
<p>ACCUM 1</p> <p>Result</p> <p>0 0 0 0 0 0 0 0 1 0 1 1 0 1 0 1</p>	

شکل ۹-۴۴ : یک مثال از انجام دستورات منطقی



**دستورات انتقال اطلاعات به چپ یا راست ( Shift ) :**

با استفاده از این دستورات می توان الگوی بیتی موجود در آکومولاتور یک را به تعداد بیت مشخص شده در مقابل دستور که می تواند عددی بین ۰ تا ۱۵ باشد ، به چپ یا راست انتقال داد . در طی عمل انتقال به جای بیت های منتقل شده عدد صفر در آکومولاتور قرار می گیرد .

Operation	Operand	Meaning
SLW		Shift to the left. دستور انتقال به چپ
SRW		Shift to the right. دستور انتقال به راست
	Parameter	0 to 15

شکل ۹-۴۵ : دستورات شیفت اطلاعات

STL	Explanation
L DW 2	خواندن کلمه اطلاعاتی ۲ و قرار دادن آن در آکومولاتور یک
SLW 3	انتقال اطلاعات موجود در آکومولاتور یک بسمت چپ به اندازه ۳ واحد .
T DW 3	انتقال عدد حاصل از آکومولاتور یک به کلمه اطلاعاتی ۳

Numeric Example	
<p>ACCUM 1</p> <p>15 464<sub>10</sub> (DW 2) 0</p> <p>00000001 11010000</p>	<p>در طی این عمل انتقال می توان عدد را در ضرایب ۲ ، ضرب با تقسیم نمود . به ازاء هر بیت انتقال عدد به چپ ، عدد اولیه در ۲ ضرب شده و به ازاء هر بیت انتقال به راست عدد اولیه بر ۲ تقسیم می گردد .</p>
<p>ACCUM 1</p> <p>15 3712<sub>10</sub> 0</p> <p>00001110 10000000</p>	<p>در این مثال عدد اولیه برابر ۴۶۴ و عدد حاصل ۸ برابر آن یعنی ۳۷۱۲ می باشد .</p>

شکل ۹-۴۶ : مثالی از دستورات شیفت اطلاعات



**دستورات تبدیل اطلاعات :**

این دستورات اطلاعات موجود در آکومولاتور یک را تبدیل می نمایند .

Operation	Operand	Meaning
CFW		One's complement این دستور مکمل یک ، عدد موجود در آکومولاتور را ایجاد می نماید .
CSW		Two's complement این دستور مکمل دو ، عدد موجود در آکومولاتور را ایجاد می نماید .

شکل ۹-۴۷ : دستورات تبدیل اطلاعات

**یادآوری :**

- در دستور مکمل یک ، تمام بیت های عدد مکمل می گردد . به عبارت دیگر تمام یک های موجود در عدد به صفر و تمام صفرهای موجود در عدد به یک تبدیل می گردد .
- در انجام عمل مکمل ۲ ، از سمت راست به عدد نگاه شده و تا اولین یک تمام اعداد بازنویسی می شود ، اولین یک هم بازنویسی می شود . از آن پس تمام یک ها به صفر و تمام صفر ها به یک تبدیل می گردند .

STL	Explanation
L IW 12	Load the contents of input word IW 12 into ACCUM 1.
CSW	Invert all bits. Add a "1" at the least significant position.
T DW 100	Transfer the altered word to data word DW 100.
Numeric Example	
<p>ACCUM 1</p> <p style="text-align: center;">IW 12</p> <p style="text-align: center;">15 0</p> <p style="text-align: center;">0 1 0 1 1 0 0 1 1 1 0 0 0 1 0 1</p> <p style="text-align: center;">↓ ↓ CSW ↓ ↓ +1</p> <p style="text-align: center;">15 0</p> <p>ACCUM 1</p> <p style="text-align: center;">1 0 1 1 0 0 1 1 0 0 0 1 1 1 0 1 1</p>	Form the negative value of the value in input word IW 12.

شکل ۹-۴۸ : مثالی از دستور مکمل ۲



**دستورات افزایش و کاهش :**

با استفاده از این دستورات می توان از عدد موجود در آکومولاتور حداکثر به اندازه ۲۵۵ واحد کم نموده و یا به آن افزود . میزان افزایش یا کاهش با توجه به عدد نوشته شده در مقابل دستور محاسبه می گردد .

Operation	Operand	Meaning
D		Decrement
I		Increment
		Parameter 0 to 255

شکل ۹-۴۹: دستورات عمل های افزایش و کاهش

Example	STL	Explanation
<p>□ اضافه کردن ۱۶ واحد به عدد مبنای ۱۶ ، ۱۰۱۰ و قرار دادن آن در کلمه اطلاعاتی ۸ از بلوک اطلاعاتی ۶</p>	<pre>C DB 6 L KH 1010 I 16 T DW 8</pre>	<p>با استفاده از دستور C DB 6 ، بلوک اطلاعاتی ۶ را فعال کرده و از این پس تمام کلمه های اطلاعاتی مورد استفاده را از این بلوک بکار می بریم . با استفاده از دستور بارگذاری ، مقدار ثابت مبنای ۱۶ برابر با ۱۰۱۰ را در آکومولاتور یک بار گذاری کرده و سپس با استفاده از دستور افزایش و با تعیین میزان پارامتر آن به اندازه ۱۶ واحد مقدار داخل آکومولاتور را به اندازه ۱۶ واحد افزایش می دهیم . در این حال عدد ۱۰۲۰ در آکومولاتور قرار خواهد گرفت که این مقدار را با استفاده از دستور انتقال به کلمه اطلاعاتی ۸ ، انتقال می دهیم .</p>
<p>□ کاهش ۳۳ واحد از نتیجه حاصل در مرحله قبل و انتقال آن به کلمه اطلاعاتی ۹</p>	<pre>D 33 T DW 9</pre>	<p>کاهش ۳۳ واحد از نتیجه حاصل در مرحله قبل و انتقال آن به کلمه اطلاعاتی ۹</p>

شکل ۹-۵۰: مثالی از دستورات عمل های افزایش و کاهش



**دستورات فعال و غیر فعال سازی وقفه ها :**

این دستورات روی برنامه های مربوط به وقفه ها و وقفه های زمانی اثر گذاشته و از تداخل برنامه های مربوط به این وقفه ها جلوگیری می نماید . نحوه استفاده از وقفه ها در این نوع PLC در ادامه مورد بررسی قرار خواهند گرفت .

Operation	Operand	Meaning
IA		غیر فعال سازی وقفه ها Disable interrupt
RA		فعال سازی وقفه ها Enable interrupt

شکل ۹-۵۱ : دستورات فعال سازی و غیر فعال سازی وقفه ها

توجه : در صورتی که وقفه ها با بکار گیری دستور IA غیر فعال گردند ، CPU به هیچ وقفه ای پاسخ نخواهد داد

**دستور پردازش ( DO ) :**

این دستور امکان بکارگیری دستورات S5 را بصورت اندیسی ایجاد می نماید . در این روش می توان پارامترهای مربوط به عملوند را در برنامه ها تغییر داد . این دستور مستقل از RLO عمل می نماید شامل دو دستور وابسته زیر است :

- جمله اول شامل دستور پردازش بوده و یک FW یا DW را مشخص می کند .
- جمله دوم ، دستور و عملوندی که توسط برنامه کنترل ، پردازش خواهد شد را مشخص می نماید .

پس از انجام تعاریف فوق برنامه کنترل با پارامتری کار می نماید که در FW یا DW فراخوانی شده توسط جمله اول ذخیره شده است .

Operation	Operand	Meaning
DO		Process a flag or data word.
ID	↑ FW DW	↑ Parameter 0 to 254 0 to 255

شکل ۹-۵۲ : دستور پردازش



مثال ارائه شده در شکل ۹-۵۳ تاثیر گذاری دستور DO را نمایش می دهد. در این مثال در بلوک عملگر FBx، بلوک اطلاعاتی ۶، فعال گشته و با استفاده از دستور DO مقدار موجود در کلمه اطلاعاتی ۱۲ به عنوان یک پارامتر تعریف می گردد. در این حال در هنگام استفاده از دستور A مقدار پارامتر مربوطه که عدد 8.1 را بیان می نماید، جایگزین عملوند تعریف شده یعنی I 0.0 می گردد. همچنین در خطوط بعد نیز مقدار موجود در DW13 جایگزین عملوند تایمر شده است.

	DB6		FB x	Executed program
			:C DB 6	:C DB 6
			:	:
			:	:
DW 12	KH=0108		:DO DW 12	:
			:A I 0.0	:A I 8.1
DW 13	KH=0001		:DO DW 13	:
			:FR T 0	:FR T 1

شکل ۹-۵۳: مثالی از بکارگیری دستور پردازش

توجه: در بکارگیری دستور پردازش و در زمان کار کردن با عبارت بیتی مقدار موجود در بایت پایین کلمه، آدرس بایت و مقدار موجود در بایت بالای کلمه مقدار بیت مورد نظر را نمایش می دهد. در غیر این صورت بایت بالا با مقدار صفر پر می گردد. (به مثال ارائه شده در شکل ۹-۵۳ دقت نمایید.) دستورات مشخص شده در شکل ۹-۵۴ را می توان همراه دستور پردازش استفاده نمود.

Operations	Explanation
A <sup>1</sup> , AN, O, ON	Binary logic operations
S, R, =	Set/reset operations
FR T, RT, SF T, SD T, SI T, SS T, SE T	Timer operations
FR C, RC, SC, CD C, CU C	Counter operations
L, LC, T	Load and transfer operations
JU=, JC=, JZ=, JN=, JP=, JM=, JO=	Jump operations
SLW, SRW	Shift operations
D, I	Decrement and increment
C DB, JU, JU, TNB	Block calls

شکل ۹-۵۴: دستورات قابل استفاده با دستور پردازش





Example	STL	Explanation
در این مثال، با استفاده از DW1 به عنوان یک پارامتر، DW20 تا DW100 را با مقدار صفر پر می نماییم.	<pre> :C   DB 202 :L   KB 20  :T   DW 1  L 1 :L   KH 0  :DO  DW 1  :T   DW 0  :L   DW 1  :L   KB 1  :+F  :T   DW 1  :L   KB 100  :&lt;=F  :JC = L 1                     </pre>	<p>ابتدا با استفاده از دستور C DB 202، بلوک اطلاعاتی ۲۰۲ را به عنوان بلوک اطلاعاتی فعال انتخاب می نماییم. سپس مقدار باینری ۲۰ را در DW1 قرار می دهیم. این مقدار در ادامه به عنوان پارامتر مورد نظر تعریف می گردد.</p> <p>پس از صفر نمودن آکومولاتور، با استفاده از دستور DO، پارامتر مورد نظر را تعریف می نماییم.</p> <p>با استفاده از دستور T DW0، پارامتر مورد نظر جایگزین عملوند مربوطه (شماره کلمه اطلاعاتی) شده و لذا مقدار صفر موجود در آکومولاتور به DW20 منتقل می گردد.</p> <p>سپس برای مقدار دهی سایر کلمه های اطلاعاتی عدد موجود در DW1 را در هر مرحله با عدد یک جمع نموده و پس از مقایسه مقدار با عدد و در صورت کوچکتر یا مساوی بودن آن با عدد ۱۰۰، با استفاده از دستور پرش شرطی، ادامه برنامه از آدرس L1، دنبال می گردد. در غیر اینصورت برنامه پایان می پذیرد.</p>

شکل ۹-۵۵: مثالی از بکارگیری دستور پردازش



**دستورات پرش:**

با استفاده از دستورات پرش می توان در صورت برقراری شرایط ، حداکثر ۱۲۷ کلمه در حافظه برنامه به جلو و ۱۲۸ کلمه به عقب پرش نمود . برای ایجاد پرش های بلندتر می بایست از آدرس های میانی استفاده نمود . دستورات پرش تنها در یک بلوک قابل اجرا بوده و نمی توان از یک بلوک به بلوک دیگر پرش نمود . همچنین امکان پرش از مرز Segment ها ( هر بلوک برنامه ، سازماندهی و ترتیبی از ۲۵۶ بخش کوچکتر بنام Segment تشکیل شده اند . ) نیز وجود ندارد . جدول ارائه شده در شکل ۹-۵۶ انواع مختلف دستورات پرش را بیان می نماید .

Operation	Operand	Meaning
JU =		Jump unconditionally پرش به آدرس تعیین شده بدون شرط
JC =		Jump conditionally پرش به آدرس تعیین شده به شرط یک بودن بیت RLO
JZ =		Jump if the result is "zero" پرش به آدرس تعیین شده در صورت صفر بودن نتیجه موجود در آکومولاتور
JN =		Jump if the result is "not zero" پرش به آدرس تعیین شده در صورت صفر نبودن نتیجه موجود در آکومولاتور
JP =		Jump if the result is positive پرش به آدرس تعیین شده در صورت مثبت بودن نتیجه موجود در آکومولاتور
JM =		Jump if the result is negative پرش به آدرس تعیین شده در صورت منفی بودن نتیجه موجود در آکومولاتور
JO =		Jump on overflow پرش به آدرس تعیین شده در صورت وجود سرریز
ID Jump label (up to 4 characters)		

شکل ۹-۵۶ : انواع مختلف دستورات پرش





Call in OB1	Program in FB30	Executed Program
:JU FB 30	:A =ON 1	:A I 2.0
NAME :LOGIC	:AN =ON 2	:AN I 2.1
ON 1 : I2.0	:O =ON 3	:O I 2.2
ON 2 : I2.1	:S =MOT 5	:S Q 7.3
ON 3 : I2.2	:= =OFF 1	:= Q 7.1
VAL1 : I2.3	:A =VAL 1	:A I 2.3
OFF1 : Q 7.1	:A =ON 2	:A I 2.1
OFF2 : Q 7.2	:ON =ON 3	:ON I 2.2
MOT5 : Q 7.3	:RB =MOT 5	:R Q 7.3
: BE	:= =OFF 2	:= Q 7.2
	:BE	:BE



شکل ۹-۵۹: مثالی از دستورات SET و RESET

**دستورات بارگذاری و انتقال :**

همانند دستورات بارگذاری و انتقال در دستورات اصلی در اینجا هم امکان بارگذاری و انتقال بصورت باینری ، BCD و علاوه بر آن الگوی بیتی نیز وجود دارد . دستورات ارائه شده در شکل ۹-۶۰ ، انواع مختلف این دستورات و انواع پارامترهای قابل تعریف در آنها را بیان می دارد .

Operation	Operand	Meaning		
L =		Load a formal operand.		
LC =		Load a formal operand in BCD code.		
LW =		Load the bit pattern of a formal operand.		
T =		Transfer to a formal operand.		
	↑ Formal operand	Actual Operands Permitted	Parameter Type	Data Type
For L =		Inputs, outputs and flags <sup>1</sup> , timers and counters addressed in byte and word form	I, Q T, C	BY, W
For LC =		Timers and counters	T, C	
For LW =		Bit pattern	D	KF, KH, KM, KY, KS, KT, KC
For T =		Inputs, outputs and flags <sup>1</sup> addressed in byte and word form	I, Q	BY, W

شکل ۹-۶۰: انواع مختلف دستورات بارگذاری و انتقال



مثال ارائه شده در شکل ۹-۶۱ یک نمونه از کاربرد دستورات بارگذاری و انتقال را نمایش می دهد. در این مثال دو بلوک PB1 و FB34 استفاده شده است، که مجموع کارکرد برنامه نوشته شده در آنها در قسمت سوم ارائه شده است. نحوه کاربرد دستورات بارگذاری و انتقال در فرمت های گوناگون در این مثال قابل بررسی است.

Call in PB1	Program in FB34	Executed Program
	:A =I0	:A I 2.0
	:L =L1	:L FW 10
:JU FB 34	:S C 6	:S C 6
NAME :LOAD/TRAN	:A =I1	:A I 2.1
I0 : I2.0	:LW =LW1	:L KC 140
I1 : I2.1	:S C 7	:S C 7
L1 : FW 10	:A I 2.2	:A I 2.2
LW1 : KC 140	:CU C 6	:CU C 6
LC1 : C 7	:CU C 7	:CU C 7
T1 : QW 4	:LC =LC1	:LC C 7
LW2 : KC 160	:T =T1	:T QW 4
:BE	:A I 2.7	:A I 2.7
	:R C 6	:R C 6
	:R C 7	:R C 7
	:LW =LW2	:L KC 160
	:LC =LC1	:LC C 7
	:! =F	:! =F
	:R C 7	:R C 7
	:BE	:BE

شکل ۹-۶۱: مثالی در خصوص دستورات بارگذاری و انتقال

### دستورات زمان سنج و شمارنده ها :

با استفاده از این دستورات می توان یک تایمر یا شمارنده را با پرامتر تعریف شده در آکومولاتور فعال نمود. نوع و کارکرد این زمان سنج ها و شمارنده ها دقیقا شبیه به نمونه های مشابه تعریف شده در دستورات اصلی است. روش های مختلف نمایش این توابع در جدول ارائه شده در شکل ۹-۶۲ معرفی شده اند. همچنین در ادامه دو مثال جهت آشنایی با نحوه برنامه نویسی این توابع ذکر شده است.



Operation	Operand	Meaning		
FR =		<b>Enable</b> a formal operand for cold restart. (For a description, see "FT" or "FC", according to the formal operand).		
RD =		<b>Reset</b> a formal operand (digital).		
SP =		<b>Start</b> a pulse timer		
SR =		<b>Start</b> an on-delay timer		
SEC =		<b>Start</b> an extended pulse timer or <b>set</b> a counter		
SSU =		<b>Start</b> a stored on-delay timer or start the <b>count up</b>		
SFD =		<b>Start</b> an off-delay timer or start the <b>count down</b>		
	↑ Formal operand			
Formal operand		Actual Operands Permitted	Parameter Type	Data Type
		Timers and counters <sup>1</sup>	T, C <sup>1</sup>	

شکل ۹-۶۲: انواع دستورات تایمر و شمارنده

Function Block Call	Program in Function Block (FB32)	Executed Program
:JU FB 32	:AN =I 5	:AN I 2.5
NAME:TIME	:A =I 6	:A I 2.6
I5 : I 2.5	:L KT 5.2	:L KT 5.2
I6 : I 2.6	:SFD =TIM5	:SF T 5
TIM5 : T 5	:A =I 5	:A I 2.5
TIM6 : T 6	:AN =I 6	:AN I 2.6
OUT6 : Q 7.6	:L KT 5.2	:L KT 5.2
:BE	:SSU =TIM6	:SS T 6
	:A =TIM5	:A T 5
	:O =TIM6	:O T 6
	:= =OUT6	:= Q 7.6
	:A I 2.7	:A I 2.7
	:RD =TIM5	:R T 5
	:RD =TIM6	:R T 6
	:BE	:BE

شکل ۹-۶۳: مثالی از نحوه برنامه نویسی زمان سنج ها در بلوک های برنامه



Function Block Call	Program in Function Block (FB33)	Executed Program
STL	:A =I2	:A I 2.2
:JU FB 33	:L KC 17	:L KC 17
NAME:COUNT	:SEC =COU5	:S C 5
I2 : I2.2	:A =I 3	:A I 2.3
I3 : I2.3	:SSU =COU5	:CU C 5
I4 : I2.4	:A =I 4	:A I 2.4
COU5: C 5	:SFD =COU5	:CD C 5
OUT3: Q 7.3	:A =COU5	:A C 5
:BE	:= =OUT3	:= Q 7.3
	:A I 2.7	:A I 2.7
	:RD =COU5	:R C 5
	:BE	:BE

شکل ۹-۶۴: مثالی از نحوه برنامه نویسی شمارنده ها در بلوک های برنامه

### نحوه تولید کدهای شرطی :

CPU مربوط به PLC-S5-115U دارای سه کد شرطی به شرح زیر می باشد :

CC 0

CC 1

OV

دستورالعمل های زیر بر روی کدهای شرطی اثر می گذارند :

- دستورالعمل های ریاضی
- دستورات جابجایی اطلاعات
- دستورالعمل های مقایسه
- برخی از دستورات تبدیل داده

بررسی وضعیت کدهای شرطی برای بکاربردن دستورات پرش مورد استفاده قرار می گیرد. جدول های ارائه شده در شکل های زیر نحوه اثر گذاری دستورالعمل های مختلف بر روی کدهای شرطی و همچنین دستورات پرشی را که ، به ازاء برقراری این شرایط قابل اجرا خواهند بود را به نمایش می گذارد.



Contents of ACCUM 2 as Compared to Contents of ACCUM 1	Condition Codes			Possible Jump Operations
	CC 1	CC 0	OV	
Equal to	0	0		JZ
Less than	0	1		JN, JM
Greater than	1	0		JN, JP

شکل ۹-۶۵: وضعیت کدهای شرطی در اثر اعمال دستورات مقایسه بین آکومولاتور یک و دو

Result after Arithmetic Operation is Executed	Condition Codes			Possible Jump Operations
	CC 1	CC 0	OV	
< - 32768	1	0	1	JN, JP, JO
- 32768 to - 1	0	1	0	JN, JM
0	0	0	0	JZ
+1 to +32767	1	0	0	JN, JP
> +32767	0	1	1	JN, JM, JO
(-) 65536 *	0	0	1	JZ, JO

شکل ۹-۶۶: وضعیت کدهای شرطی پس از اجرای دستورات ریاضی

Contents of the ACCUM	Condition Codes			Possible Jump Operations
	CC 1	CC 0	OV	
Zero (KH=0000)	0	0		JZ
Not zero	1	0		JN, JP

شکل ۹-۶۷: وضعیت کدهای شرطی در اثر مقدار موجود در آکومولاتور پس از اجرای دستورات منطقی





Value of the Last Bit Shifted Out	Condition Codes			Possible Jump Operations
	CC 1	CC 0	OV	
"0"	0	0		JZ
"1"	1	0		JN, JP

شکل ۹-۶۸: وضعیت کدهای شرطی در اثر آخرین بیت خارج شده پس از اجرای دستورات شیفت

Result after Arithmetic Operation is Executed	Condition Codes			Possible Jump Operations
	CC 1	CC 0	OV	
- 32768 *	0	1	1	JN, JM, JO
- 32767 to - 1	0	1	0	JN, JM
0	0	0	0	JZ
+1 to +32767	1	0	0	JN, JP

شکل ۹-۶۹: وضعیت کدهای شرطی در اثر مقدار نهایی اطلاعات ، پس از اجرای دستورات تبدیل داده

### دستورات سیستمی:

از آنجا که استفاده از این دستورات در رابطه مستقیم با سیستم عامل می باشد ، لذا استفاده از آنها تنها برای برنامه نویسان حرفه ای که تسلط کافی بر روی سیستم عامل PLC داشته و از بکاربردن دستورات و در نهایت عدم آسیب دیدن سیستم مطمئن باشند مجاز می باشد .  
با عنایت به مقدمه فوق الذکر در این جزوه از بررسی این دستورات خودداری شده و تنها جهت مطالعه و در نهایت استفاده برنامه نویسان حرفه ای ، لیست این دستورات در ضمیمه آورده شده است .



## وقفه ها (Interrupt's)

در سیستم های مبتنی بر پردازنده از وقفه ها جهت اعمال یک وضعیت فوق العاده با اولویت بالاتر، نسبت به سیکل معمول اجرای برنامه ، استفاده می شود . منابع ایجاد وقفه در PLC-S5-115U عبارتند از :

۱- ماژول ورودی دیجیتال 434-7

۲- ماژول های هوشمند دارای پروسسور

۳- وقفه های زمانی

از آنجا که در اثر ایجاد هر وقفه یک بلوک سازماندهی خاص اجرا می گردد ، لذا می توان اعمال مورد نیاز ، جهت اجرا در زمان وقوع هر وقفه را در بلوک مربوطه برنامه ریزی نمود .

### انواع وقفه ها :

- وقفه A : این وقفه توسط کارت ورودی دیجیتال 434-7 یا بوسیله ماژول های هوشمند ورودی /خروجی ( Ips ) و یا توسط پروسسورهای ارتباطی ( CPs ) ایجاد گشته و در اثر وقوع آن بلوک OB2 اجرا می گردد .
- وقفه B : این وقفه بوسیله برخی از ماژول های هوشمند ورودی /خروجی ( Ips ) و یا توسط برخی از پروسسورهای ارتباطی ( CPs ) ایجاد گشته و در اثر وقوع آن بلوک OB3 اجرا می گردد .
- وقفه C : این وقفه بوسیله برخی از ماژول های هوشمند ورودی /خروجی ( Ips ) و یا توسط برخی از پروسسورهای ارتباطی ( CPs ) ایجاد گشته و در اثر وقوع آن بلوک OB4 اجرا می گردد .
- وقفه D : این وقفه بوسیله برخی از ماژول های هوشمند ورودی /خروجی ( Ips ) و یا توسط برخی از پروسسورهای ارتباطی ( CPs ) ایجاد گشته و در اثر وقوع آن بلوک OB5 اجرا می گردد .
- وقفه های زمانی که پس از تعریف و در مدت زمانهای مشخص بوقوع می پیوندند . در اثر اجرای این وقفه ها یکی از بلوک های OB10 تا OB13 اجرا می گردد .

### نکات مهم در بکارگیری وقفه ها :

- ۱- در حین پاسخ دهی به یک وقفه و اجرای OB مربوطه سایر وقفه ها غیر فعال می گردند .
- ۲- در زمان وقوع چند وقفه همزمان اولویت  $A \leftarrow B \leftarrow C \leftarrow D$  رعایت می گردد .
- ۳- وقفه های زمانی به ترتیب از OB10 تا OB13 بیشترین اولویت را در بین تمام وقفه ها دارا می باشند .
- ۴- در صورت عدم برنامه ریزی بلوک های وقفه ، برنامه اصلی به طور خودکار پس از وقوع وقفه اجرا می گردد .
- ۵- اگر در برخی از خطوط برنامه نیاز به وقوع وقفه نیست باید با استفاده از عملگر IA وقفه ها را غیر فعال نموده و در صورت نیاز به فعال سازی مجدد وقفه ها با استفاده از دستور RA آنرا فعال نماییم .
- ۶- مادامی که وقفه ها غیر فعال می باشند ، یکی از وقفه های وارد شده در خط وقفه ذخیره می گردد .
- ۷- در فراخوانی وقفه های تو در تو ، عمق حلقه های تودرتو نباید از ۳۲ سطح بیشتر شود .



۸- اگر پرچم های یکسانی در سیکل برنامه عادی و همچنین روتین های وقفه مورد استفاده قرار می گیرد ، در ابتدای برنامه وقفه می بایست این مقادیر ذخیره شده و در انتهای سیکل وقفه مقادیر ذخیره شده بازیابی گردند .

۹- اگر مایل به پاسخ دهی وقفه ها در روتین های Restart ( OB21 و OB22 ) هستیم ، باید وقفه ها در شروع بلوک با بکارگیری دستور RA فعال شوند . در غیر اینصورت وقفه ها تنها پس از اتمام روتین های مورد نظر عمل می نمایند .

### زمان پاسخ دهی به وقفه ها

مدت زمان پاسخ دهی به یک وقفه به سه عامل بستگی دارد :

- تاخیر سیگنال در ماژول های مولد وقفه
- زمان تاخیر پاسخ دهی CPU به وقفه
- زمان کل مربوط به اجرای روتین وقفه

### پردازش وقفه تولید شده توسط ماژول ورودی دیجیتال 7-434 :

این کارت ورودی دیجیتال که برکه مشخصات آن در شکل ۹-۷ ارائه شده است ، قابلیت ایجاد وقفه خارجی را برای PLC فراهم می کند . دو راه تشخیص وقفه وارد شده به این کارت وجود دارد :

- ۱- ورودی های ایجاد کننده وقفه بوسیله برنامه کنترل تشخیص داده شود .
- ۲- در اثر ایجاد وقفه یک LED زرد رنگ بر روی ماژول روشن شده و یک کنتاکت رله بسته می شود . این کنتاکت بصورت خارجی از طریق خروجی های MELD قابل دسترسی هستند . این سیگنالها با اعمال یک ولتاژ ۲۴ ولت به ورودی Reset از بین می روند .

جهت استفاده از این کارت می بایست موارد زیر در بلوک های Restart ( OB21 , OB22 ) برنامه ریزی گردند :

- ۱- کدام ورودی های کارت برای تحریک وقفه مورد استفاده قرار می گیرند .
  - ۲- تعیین نوع اینکه وقفه ها با لبه بالارونده یا پایین رونده تحریک گردند .
- با استفاده از یک کلمه باینری که با استفاده از دستور  $L \quad KM \quad ab$  مقداردهی شده و با استفاده از دستور  $x \quad PW \quad T$  به آدرس مورد نظر ارسال می گردد ، می توان اطلاعات فوق را برای کارت تعریف نمود . در دو دستور فوق حرف  $a$  و  $b$  بترتیب بیانگر بایت های بالا و پایین کلمه مورد نظر و حرف  $x$  بیانگر آدرس کارت می باشد . بایت بالای کلمه فوق الذکر جهت فعال نمودن بیت های صادر کننده وقفه استفاده شده و بایت پایین اطلاعات ، جهت تعیین لبه ایجاد کننده وقفه استفاده می گردد . این بایت در شکل ۹-۷ نمایش داده شده و نحوه مقداردهی آن نیز در همان شکل مشخص شده است .



## Digital Input Module 8 x 24 V DC (with P Interrupt), Floating

(6ES5 434-7LA12)

**Technical Specifications**

Number of inputs	8
Floating	yes (optocoupler)
- in groups of	1
Input voltage L+	
- rated value	24 V DC
- for "0" signal	30 to +5 V
- for "1" signal	13 to 30 V
Input current at "1" signal	
- for 24 V DC	typically 8.5 mA
Response time	
- from "0" to "1"	0.5 to 1.5 msec.
- from "1" to "0"	0.5 to 1.5 msec.
Interrupt signal (external)	latching relay contact (permissible load: max. 0.2 A 50 V DC - switching capacity max. 20 W or 35 VA)
Interrupt signal (internal)	via bus line PRAL-N
Acknowledgement	external via input reset 24 V DC
Cable length	
- shielded	maximum 1000 m (3281 ft.)
- unshielded	maximum 600 m (1969 ft.)
Isolation rating	according to VDE 0160
Rated isolation voltage (between groups)	30V
- isolation group	C
- tested with	500 V
Rated isolation voltage (L+ to $\perp$ )	30 V
- isolation group	C
- tested with	500 V
Connection of 2-wire BERO proximity switches	possible
- quiescent current	maximum 1.5 mA
Current Consumption	
- from 5 V (internal)	<70 mA
Power losses of the module	typically 2 W
Weight	approx. 0.7 kg (1.54 lb.)

**Terminal Assignment**

**Simplified Schematic**

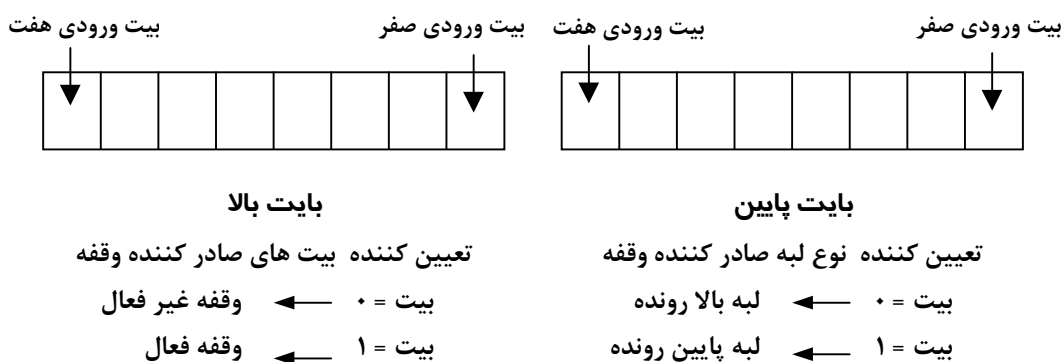
شکل ۹-۷۰: برگه مشخصات مربوط به کارت ورودی دیجیتال 434-7



آموزشگاه برق و الکترونیک صنعتی اندیشه سازان صنعت مجری برگزاری دوره های آموزشی از مقدماتی تا پیشرفته در زمینه های برق، الکترونیک، اتوماسیون صنعتی، ابزار دقیق و کامپیوتر - با مجوز رسمی از سازمان آموزش فنی و حرفه ای کشور و وزارت

مشهد مقدس - ۷۶۱۸۲۸۶ - ۷۸ - ۷۶۴۴۰ - ۵۱۱

کار و امور اجتماعی - اعطای مدرک با کد بین المللی I.L.O



شکل ۹-۷۱: کلمه باینری مربوط به تعریف وقفه در کارت 7-434

**مثال:** یک قطعه بر روی نوار نقاله و بین دو نقطه که توسط لیمیت سوئیچ های یک و دو مشخص شده است، قرار داشته و می تواند در این فاصله حرکت نماید. با استفاده از دو موتور می توان نوار نقاله را دو جهت حرکت داد. برای تعیین جهت حرکت از لیمیت سوئیچ های قرار داده شده استفاده می شود.

کارت ورودی دیجیتال با شماره 7-434 و با قابلیت ایجاد وقفه، توسط ماژول IM306 (که بر روی Rack اصلی قرار گرفته و قابلیت آدرس دهی اختیاری را برای SLOT های مختلف فراهم می نماید)، در آدرس ۸ قرار گرفته است. (نحوه آدرس دهی متغیر در انتهای مبحث وقفه ها توضیح داده می شود.)

لیمیت سوئیچ یک به کانال صفر و لیمیت سوئیچ دو به کانال یک از کارت ورودی متصل می باشند. طبق تعریف می خواهیم، لیمیت سوئیچ یک بالبه مثبت و لیمیت سوئیچ دو با لبه منفی تولید وقفه نمایند. این کار با مقدار دهی کلمه مربوطه و در نهایت انتقال آن به آدرس کارت انجام می گردد.

در این برنامه Q0.0 بیانگر درایو یک و Q0.1 بیانگر درایو دو برای راه اندازی دو موتور می باشند. همچنین برای نمایش وضعیت کارکرد مدار دو LED در نظر گرفته شده است. در صورت فعال بودن درایو یک LED1 (Q1.0) روشن بوده و در صورت فعال بودن درایو دو LED2 (Q1.1) روشن خواهد شد.

قابل ذکر است که روشن شدن نمایشگرها باعث روشن شدن درایوها شده و این عمل با توجه به وضعیت لیمیت سوئیچ ها در بدنه اصلی برنامه انجام می شود. در این مثال وقفه ها وضعیت خاموش شدن درایو ها را کنترل می نمایند.

قابل ذکر است کارت تعریف شده در آدرس ۸، دو بایت ۸ و ۹ از منطقه تصویر ورودی (PII) را شامل می شود، که بایت ۸، وضعیت ورودی های کارت را صرف نظر از اینکه آیا وقفه برای آنها تعریف شده است یا خیر، نگهداری کرده و بایت ۹، در صورت وقوع وقفه در یکی از بیت های ورودی کارت، وضعیت بیت های صادر کننده وقفه را نمایش می دهد. بدین ترتیب در صورت وقوع یک وقفه در هر یک از بیت های ورودی این کارت، بلوک مربوطه که برای این کارت OB2 می باشد، فراخوانی می گردد. در این حال تعیین بیت صادر کننده وقفه با بررسی بایت دوم از آدرس (در این مثال بایت شماره ۹) صورت می پذیرد.

برنامه های ارائه شده در شکل ۹-۷۲، مراحل مختلف برنامه نویسی جهت اجرای برنامه فوق را نمایش می دهند.



STL OB20/OB21	Meaning
L KM 0000 0011 0000 0010 T PW 8 BE	Initialization of the interrupt inputs: Enable channel 0: positive edge Enable channel 1: negative edge

STL OB1	Meaning
: L PY 8 T IB 8  A I 8.0 = Q 1.0  A I 8.1 = Q 1.1 : BE	<p>✓ خواندن بایت ورودی ۸ و قرار دادن آن در آدرس ۸ در منطقه تصویر ورودی . در بایت ۸ وضعیت ورودی ها بدون در نظر گرفتن اینکه آیا وقفه برای آنها تعریف شده یا خیر قرار داده می شود .</p> <p>✓ انتقال وضعیت لیمیت سوئیچ یک به LED1</p> <p>✓ انتقال وضعیت لیمیت سوئیچ دو به LED2</p>

STL OB2	Meaning
L PY 9 T IB 9  A I 9.0 R Q 0.0  A I 9.1 R Q 0.1  L QB 0 T PY 0 BE	<p>□ این بلوک در صورت ایجاد وقفه ( تغییر وضعیت لیمیت سوئیچ ها ) اجرا خواهد شد .</p> <p>✓ وضعیت بایت ورودی ۹ را به منطقه تصویر ورودی انتقال می دهیم ، زیرا دستورات مربوط به خواندن اطلاعات مقادیر مورد نظر خود را از این منطقه برداشت می نمایند . ( مانند دستور A )</p> <p>✓ در صورت بروز وقفه در هر یک از کانال ها خروجی مربوطه خاموش خواهد شد .</p>

شکل ۹-۷۲: برنامه های مربوط به مثال وقفه



### وقفه های زمانی (Time Interrupt's)

OB های ۱۰ تا ۱۳ برای اجرای برنامه بصورت سیکلی مورد استفاده قرار می گیرند . به عبارت دیگر با تعریف برخی پارامترهای خاص ، فراخوانی بلوک های ذکر شده را تابع زمان کرده و در فواصل زمانی مشخص و با ایجاد یک وقفه ، CPU برنامه اصلی را رها کرده و پس از انجام بلو کهای عملیاتی فوق الذکر ، روال اصلی برنامه را ادامه می دهد . در بلوک های فوق الذکر تقدم اجرا از بلوک OB13 به OB10 تعریف گشته است . لذا در صورت وقوع همزمان دو وقفه ، وقفه با اولویت بالاتر اجرا خواهد شد .

دستورات IA و RA همانند سایر وقفه ها ، در وقفه های زمانی نیز موجب فعال و غیر فعال شدن وقفه ها می گردد .

جهت استفاده از این وقفه های زمانی و برای تعیین فواصل زمانی مورد نظر ، می بایست کلمه های اطلاعاتی سیستمی خاصی ( این کلمه ها مورد استفاده سیستم عامل PLC می باشد ) را مقداردهی کرد . جدول ارائه شده در شکل ۹-۷۳ ، این کلمه ها را معرفی کرده و مقدار پیش فرض قرار گرفته در آنها را نمایش می دهد . در شکل ۹-۷۴ ، یک مثال از نحوه تعریف و بکارگیری وقفه های زمانی ارائه شده است . در این مثال با قرار دادن مقدار ۱۰۰ در کلمه سیستمی ۹۷ زمانی برابر  $100 \times 10ms = 1s$  برای وقفه زمانی OB13 ایجاد شده است .

System Data Word	Absolute Address	High Byte	Low Byte	Default
SD 97	EAC2	Time interval for OB13	(0 to FFFF <sub>H</sub> ·10 ms)	10 (=100 ms)
SD 98	EAC4	Time interval for OB12	(0 to FFFF <sub>H</sub> ·10 ms)	0 (=no call)
SD 99	EAC6	Time interval for OB11	(0 to FFFF <sub>H</sub> ·10 ms)	0 (=no call)
SD 100	EAC8	Time interval for OB10	(0 to FFFF <sub>H</sub> ·10 ms)	0 (=no call)

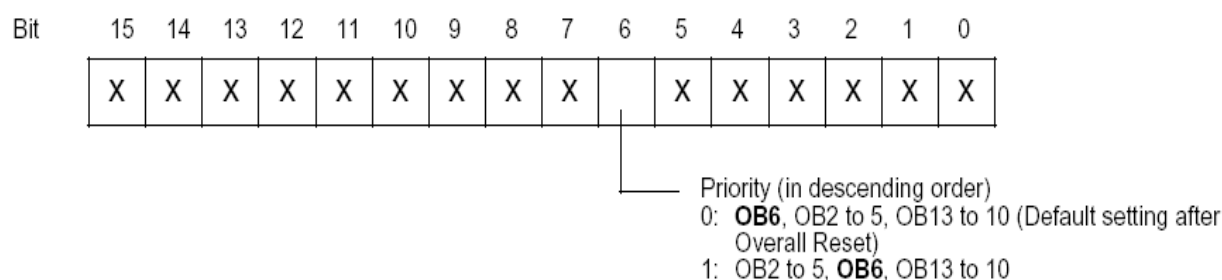
شکل ۹-۷۳: کلمه های سیستمی مربوط به برنامه ریزی وقفه های زمانی

Setting an Interval Time of 1 sec. for OB13:		
OB 21	OB 22	FB 21
: JU FB 21	:JU FB 21	
NAME : TIME ON	NAME :TIME ON	NAME :TIME ON
.	.	:L KF +100
.	.	:T BS 97
		:BE

شکل ۹-۷۴: مثالی از نحوه بکارگیری وقفه های زمانی



□ در این PLC ، علاوه بر وقفه های زمانی عنوان شده یک وقفه زمانی خاص وجود دارد ، که در صورت وقوع آن OB6 اجرا می گردد ، این وقفه با مقدار دهی کلمه سیستمی ۱۰۱ و با دستور BS 101 فعال شده و حتی با استفاده از دستور RA نیز غیر فعال نمی گردد . ( عدد دهدهی قرار داده شده در این کلمه سیستمی بر حسب میلی ثانیه محاسبه خواهد شد . )  
تنها راه متوقف کردن این وقفه بصورت نرم افزاری مقدار دهی کلمه ۱۰۱ با مقدار صفر می باشد . قابل توجه اینکه در حین زمان سنجی ، می توان با تغییر مقدار درونی این کلمه ، فواصل زمانی مربوطه را تغییر داد .  
زمانهای قابل استفاده در این کلمه بین ۳ تا ۶۵۵۳۵ میلی ثانیه ( 3H - FFFFH ) می باشد . کلمه سیستمی ۱۲۰ طبق اطلاعات ارائه شده در شکل ۹-۷۵ ، جهت تعیین اولویت اجرای OB6 قابل استفاده است .



x = Bits which determine system characteristics  
(Must not be changed when programming the priority of OB6!)

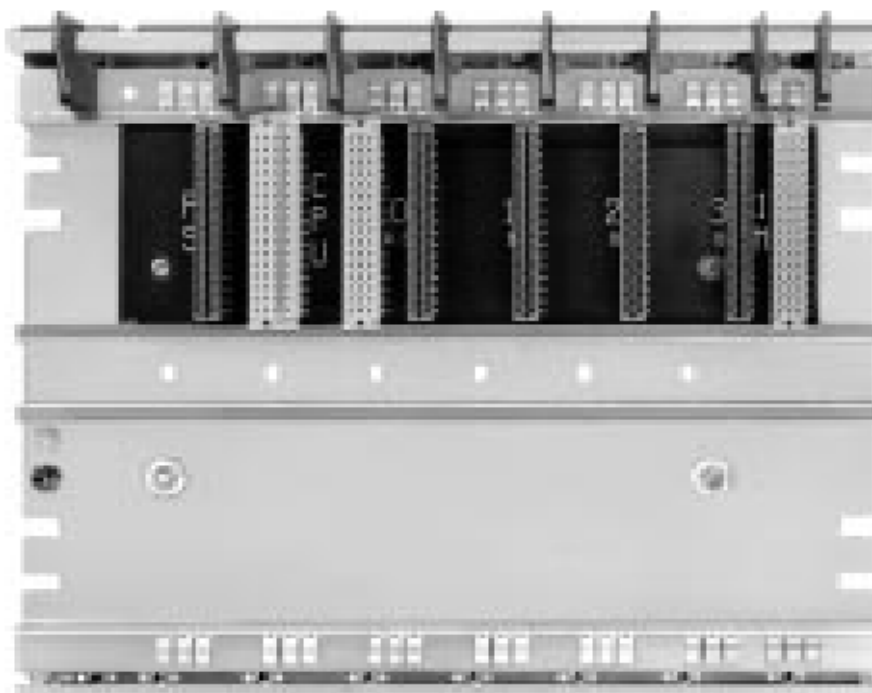
شکل ۹-۷۵: برنامه ریزی کلمه سیستمی ۱۲۰ برای تعیین اولویت OB6





### روشهای مختلف آدرس دهی

در PLC خانواده S5 ، ماژول های مختلف در محلی مشخص بر روی یک صفحه اصلی به نام Rack نصب می گردد . Rack از لحاظ کارکرد ، شبیه Main Board در سیستم های مبتنی بر PC عمل می نماید . شکل ۹-۷۶ نمونه ای از یک Rack را نمایش می دهد . بر روی این صفحه ، Slot هایی جهت اتصال ماژول های مختلف تعبیه شده است . ماژول تغذیه ، CPU ، کارتهای ورودی - خروجی آنالوگ و دیجیتال ، ماژول واسط IM306 و سایر ماژول های جانبی در محل های مشخصی از Rack نصب می گردند . شکل ۹-۷۷ محل قرار گرفتن اجزاء فوق الذکر را نمایش می دهد .



شکل ۹-۷۶: شکل ظاهری Rack با شماره سریال CR 700 - 0LA

از آنجا که نوع قرار گرفتن کارتهای ورودی و خروجی بر روی بدنه اصلی موجب ایجاد ، آدرس های مختلف برای بایت ها و بیت های مربوطه می گردد ، لذا شرکت سازنده دو نوع استاندارد مختلف برای آدرس دهی کارتهای ورودی خروجی تعریف نموده است . از آنجا که کارتهای فوق الذکر در نمونه های مختلف از لحاظ تعداد بایت و نوع کارکرد وجود دارد ، لذا در این خصوص علاوه بر شناخت نحوه آدرس دهی ، شناخت نوع کارتها نیز از اهمیت زیادی برخوردار است . انواع مختلف کارتهای ورودی - خروجی در دسترس در ضمیمه ۷ ارائه گردیده اند .



Slot No.	PS	CPU	0	1	2	3	IM
PS 951 power supply module							
CPU 941 to CPU 945							
Digital input and output, digital input/output, analog input and output modules (block type: 6ES5 4..-7....)							
Digital input and output, analog input and output modules (compact modules: 6ES5 4..-4...., 6ES5 466-3LA11)							
Communications processors	For slot assignment, see Section 11						
Intelligent I/O modules	For slot assignment, see Section 11						
IM 305, IM 306							

شکل ۹-۷۷: نحوه قرار گرفتن اجزاء مختلف بر روی Rack

### آدرس دهی کارتهای دیجیتال :

هر کارت دیجیتال چه ورودی و چه خروجی ، از تعدادی بایت ( هشت بیت ) تشکیل شده است . هر بیت بر روی کارت دیجیتال با شماره بایت و بیت مربوطه تعریف می شود . شماره بیت های هر بایت از صفر تا هفت مقدار دهی می گردد . شماره بایت مربوطه نیز با توجه به محل قرار گرفتن کارت بر روی Rack مشخص می گردد ، که این موضوع در ادامه مورد بررسی قرار خواهد گرفت . به عنوان مثال عبارت I 2.4 بیانگر یک بیت ورودی دیجیتال بوده که محل آن بیت پنجم ( از صفر تا چهار ) از بایت دوم کارت ورودی دیجیتال می باشد . همچنین Q 5.7 بیانگر آخرین بیت از بایت پنجم خروجی است .

### آدرس دهی کارتهای آنالوگ :

آدرس هریک از ورودی های یک کارت آنالوگ از یک کلمه ( دو بایت ) تشکیل می گردد که در برنامه کنترل ، هر ورودی با توجه به محل قرار گرفتن کارت بر روی بدنه اصلی و با آدرس بایت بالا از کلمه مورد نظر قابل دسترسی است .



### تعیین آدرس Slot ها :

دو روش برای تعیین آدرس اسلات ها قابل استفاده است :

- ۱- آدرس دهی ثابت ( Fixed Slot Addressing )
- ۲- آدرس دهی متغیر ( Variable Slot Addressing )

روش های آدرس دهی ثابت و متغیر برای ماژول های بلوکی قابل استفاده بوده و ماژول های هوشمند دارای پروسسور به وسیله خود آدرس دهی می شوند. ( بر روی این بلوک ها DIP Switch هایی برای تعریف آدرس تعبیه شده است. ) با این وصف ماژول IM306 که جهت تعیین و ثابت نمودن آدرس کارتها استفاده می شود در ارتباط با ماژول های هوشمند کاربردی ندارند.

### آدرس دهی ثابت :

در دو حالت CPU از روش آدرس دهی ثابت استفاده کرده که در آن آدرس هر اسلات از قبل و توسط سازنده تعریف گشته است. این دو حالت عبارتند از :

- ۱- عدم استفاده از ماژول های گسترش ( Expansion Module ) توسط PLC و استفاده از مقاومت تعیین پایان سخت افزار ( Terminating Resistor )
- ۲- استفاده از ماژول واسط IM305 جهت پیکر بندی متمرکز PLC ( این موضوع در ادامه بررسی می گردد. )

#### □ کارتهای دیجیتال :

هر اسلات می تواند تا حداکثر ۳۲ بیت را پشتیبانی نماید لذا در صورت استفاده از کارتهای دیجیتال ۸ یا ۱۶ بیت، بایت های بالای BUS مربوطه بدون استفاده خواهند بود.

#### □ کارتهای آنالوگ :

در آدرس دهی ثابت کارتهای آنالوگ تنها می توانند در اسلات های صفر تا سه قرار گیرند. از آنجا که هرورودی آنالوگ دو بایت آدرس را اشغال می نماید، لذا می توان یک کارت حداکثر با ۱۶ کانال ورودی را مرود استفاده قرار داد. در صورت استفاده از کارت با ۸ کانال ورودی، ۱۶ بیت بالای آدرس BUS بلا استفاده خواهد بود.

- در آدرس دهی کارتها و استفاده از آنها در برنامه نویسی، نکته زیر را می بایست مد نظر قرار داد :

اگر یک آدرس برای یک کارت ورودی مورد استفاده قرار می گیرد، نمی توان از همان آدرس برای یک کارت خروجی استفاده نمود. این موضوع برای استفاده از کارتهای دیجیتال و آنالوگ هم سازگار است.

شکل ۹-۷۸، نحوه آدرس دهی دقیق یک اسلات را به روش آدرس دهی ثابت بیان می نماید. در این شکل نحوه آدرس دهی کارتهای دیجیتال و آنالوگ در صورت قرار گرفتن در اسلات های مختلف بیان شده است. در شکل ۹-۷۹ نیز نحوه آدرس دهی اسلاتها بر روی یک Rack جانبی ارائه شده است.



Slot numbers in the central controller	PS	CPU	0	1	2	3	4	5	6	IM
Digital modules			0.0 3.7	4.0 7.7	8.0 11.7	12.0 15.7	16.0 19.7	20.0 23.7	24.0 27.7	
Analog modules			128 159	160 191	192 223	224 255	Analog modules cannot be plugged into these slots			
<b>Modules</b>	<b>Addresses</b>									

شکل ۹-۷۸: نحوه آدرس دهی اسلاتها بر روی Rack اصلی

Slot numbers in the expansion unit	0	1	2	3	4	5	6	7	8	IM
Digital modules	28.0 31.7	32.0 35.7	36.0 39.7	40.0 43.7	44.0 47.7	48.0 51.7	52.0 55.7	56.0 59.7	60.0 63.7	IM 305
Analog modules	Analog modules cannot be plugged in here.									
<b>Modules</b>	<b>Addresses</b>									

شکل ۹-۷۹: نحوه آدرس دهی اسلاتها بر روی Rack جانبی



### آدرس دهی متغیر اسلاتها :

در صورت استفاده از ماژول واسط IM306 بر روی Rack های اصلی و جانبی ، می توان از قابلیت آدرس دهی متغیر جهت اسلاتها ، استفاده نمود . بر روی این ماژول یک DIP Switch تعبیه شده است که با استفاده از آن می توان برای اسلاتهای مختلف آدرس های مورد نظر را تعیین نمود . IM306 علاوه بر ایجاد توانایی آدرس دهی متغیر برای اسلاتها، امکان افزایش واحد های جانبی ، حداکثر تا ۳ واحد و یا امکان اضافه نمودن یک واحد جانبی به همراه منبع تغذیه را برای سیستم فراهم می نماید . شکل ۹-۸۰ ، شکل ظاهری این ماژول را نمایش می دهد .



شکل ۹-۸۰ : ماژول واسط IM306

با بکارگیری IM306 و با در نظر گرفتن جدول ارائه شده در شکل ۹-۸۱ می توان ، روش آدرس دهی متغیر را برای اسلاتهای مختلف بکار برد . برای تنظیم آدرس مورد نظر برای یک اسلات ، می بایست مراحل زیر را طی نمود:

کلید سمت چپ موجود بر روی دیپ سوئیچ مربوطه را جهت تعیین نوع ماژول مورد استفاده تنظیم نمایید :

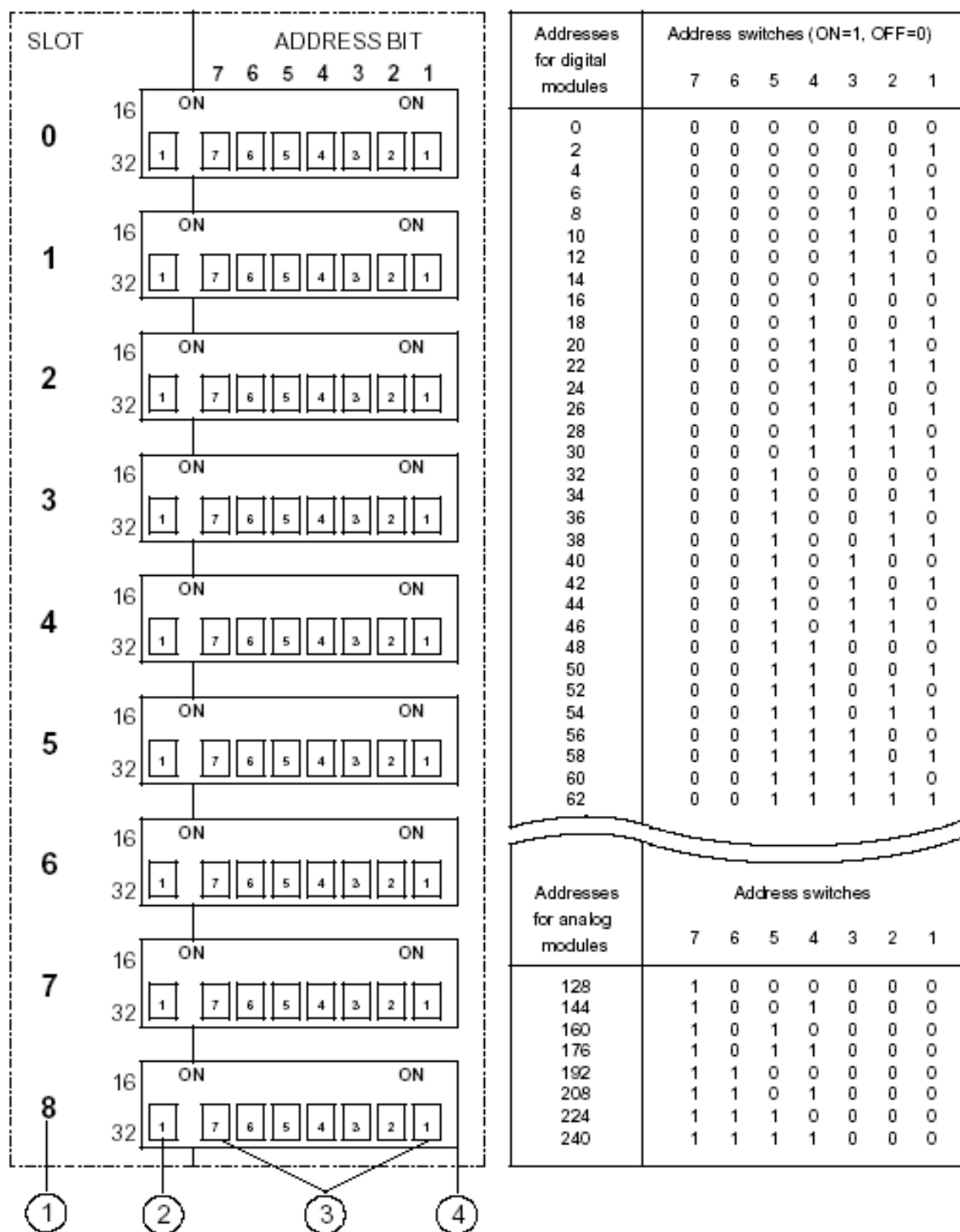
۱- کلید = ۰ : جهت بکاربردن یک ماژول دیجیتال ۳۲ بیتی یا یک ماژول آنالوگ ۱۶ کانال

۲- کلید = ۱ : جهت بکاربردن یک ماژول دیجیتال ۱۶ بیتی یا یک ماژول آنالوگ ۸ کانال

نکته : در خصوص ماژول های ۷-۴۳۴ ( ماژول ورودی دیجیتال با قابلیت ایجاد وقفه ) و ۷-۴۸۲ ( ورودی - خروجی دیجیتال ) نیز ، می بایست همانند ماژول های دیجیتال ۱۶ بیتی رفتار نمود .

پس از این کار با استفاده از هفت کلید دیگر و با استفاده از اطلاعات شکل ۹-۸۱ می توان آدرس هر اسلات را تعیین کرد .





شکل ۹-۸۱: آدرس دهی متغیر با استفاده از IM306

۱- شماره اسلات مورد نظر ۲- تعیین نوع اسلات ۳- کلید های تعیین آدرس مورد نظر ۴- دپ سوئیچ

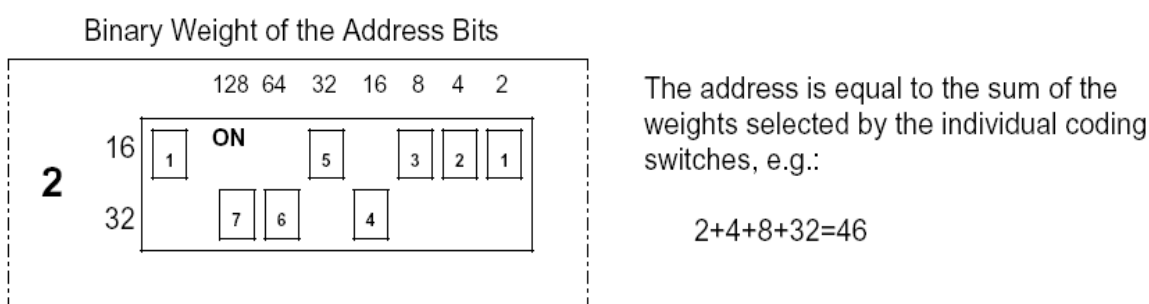


آموزشگاه برق و الکترونیک صنعتی اندیشه سازان صنعت مجری برگزاری دوره های آموزشی از مقدماتی تا پیشرفته در زمینه های برق، الکترونیک، اتوماسیون صنعتی، ابزار دقیق و کامپیوتر- با مجوز رسمی از سازمان آموزش فنی و حرفه ای کشور و وزارت

رعایت نکات زیر به هنگام تعیین آدرس کارتها ضروری است .

- ۱- آدرس شروع کارتهای ۳۲ بیت دیجیتالی می بایست با اعداد ضریب ۴ مانند ( ۰ ، ۴ ، ۸ ، ... ) آغاز گردد .
- ۲- آدرس شروع کارتهای ۱۶ بیت دیجیتالی می بایست با اعداد ضریب ۲ مانند ( ۰ ، ۲ ، ۴ ، ... ) آغاز گردد .
- ۳- آدرس شروع کارتهای ۱۶ کانال آنالوگ می بایست از بین اعدادی مانند ۱۶۰، ۱۹۲ و ۲۲۴ انتخاب گردد .
- ۴- آدرس شروع کارتهای ۸ کانال آنالوگ می بایست از بین اعدادی مانند ۱۴۴، ۱۶۰ تا ۲۴۰ انتخاب گردد .

**مثال :** یک کارت ورودی دیجیتالی ۱۶ بیتی را در اسلات ۲ قرار داده و می خواهیم آدرس بیت اول آنرا در آدرس 46.0 تنظیم نماییم . دیپ سوئیچ مورد نظر به صورت نمایش داده شده در شکل ۹-۸۲ تنظیم می گردد .

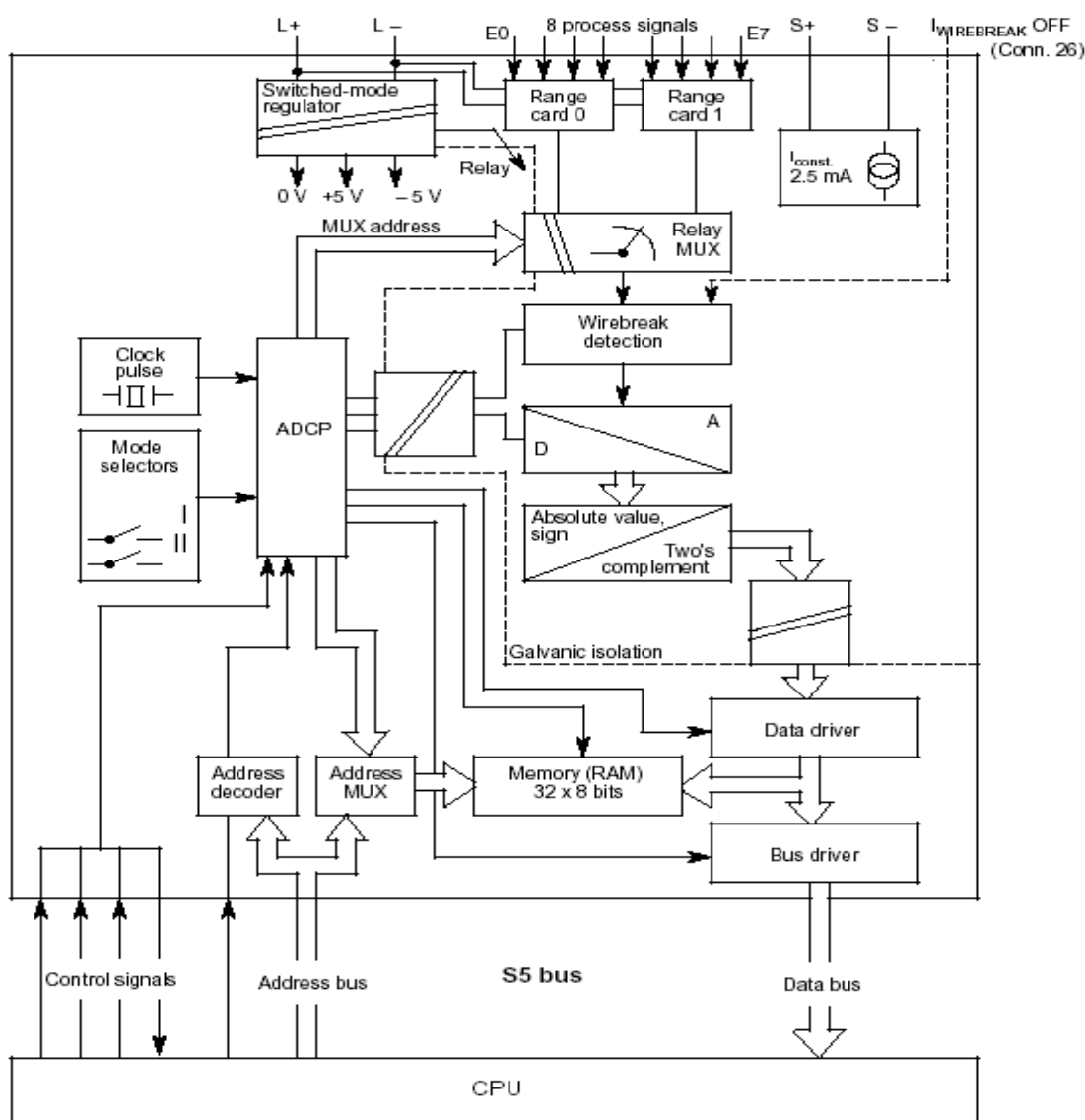


شکل ۹-۸۲ : مثالی از آدرس دهی متغیر با استفاده از IM306



### پردازش مقادیر آنالوگ :

مقادیر آنالوگ تولید شده توسط سنسورها ، توسط کارتهای ورودی آنالوگ ( که توانایی تشخیص سطوح مختلف ولتاژ بین دو حد مشخص را دارا می باشند ) خوانده شده و پس از تبدیل به یک کد دیجیتال ( معمولاً بین ۱۰ تا ۱۲ بیت ) ، در ثباتی در خود ماژول ذخیره می گردند . این مقادیر می توانند توسط CPU خوانده شده و مورد استفاده قرار گیرند . CPU مقادیر دیجیتال را از طریق یک بلوک عملیاتی FB 250 و با استفاده از دستورات بارگذاری از ماژول های آنالوگ خوانده و این مقادیر را که شامل دو بایت می گردد ، در RAM ذخیره می نماید . سه نمونه کارت ورودی آنالوگ با مشخصات متفاوت در دسترس می باشند که برگه های اطلاعاتی آنها در ضمیمه هفت ارائه شده است . به عنوان نمونه مدار داخلی و شماره پایه های کارت ورودی آنالوگ 460-7LA12 در شکل های ۸۳-۹ و ۸۴-۹ ارائه شده است .



شکل ۸۳-۹ : مدار داخلی کارت ورودی آنالوگ 460-7LA12





1	L+=24V
2	
3	M0+
4	
5	M0-
6	
7	M1+
8	
9	M1-
10	
11	S+
12	
13	
14	
15	M2+
16	
17	M2-
18	
19	M3+
20	
21	M3-
22	
23	KOMP+
25	KOMP-
26	L+=24V*
27	M4+
28	
29	M4-
30	
31	M5+
32	
33	M5-
34	
35	S-
36	
37	
38	
39	M6+
40	
41	M6-
42	
43	M7+
44	
45	M7-
46	
47	L-

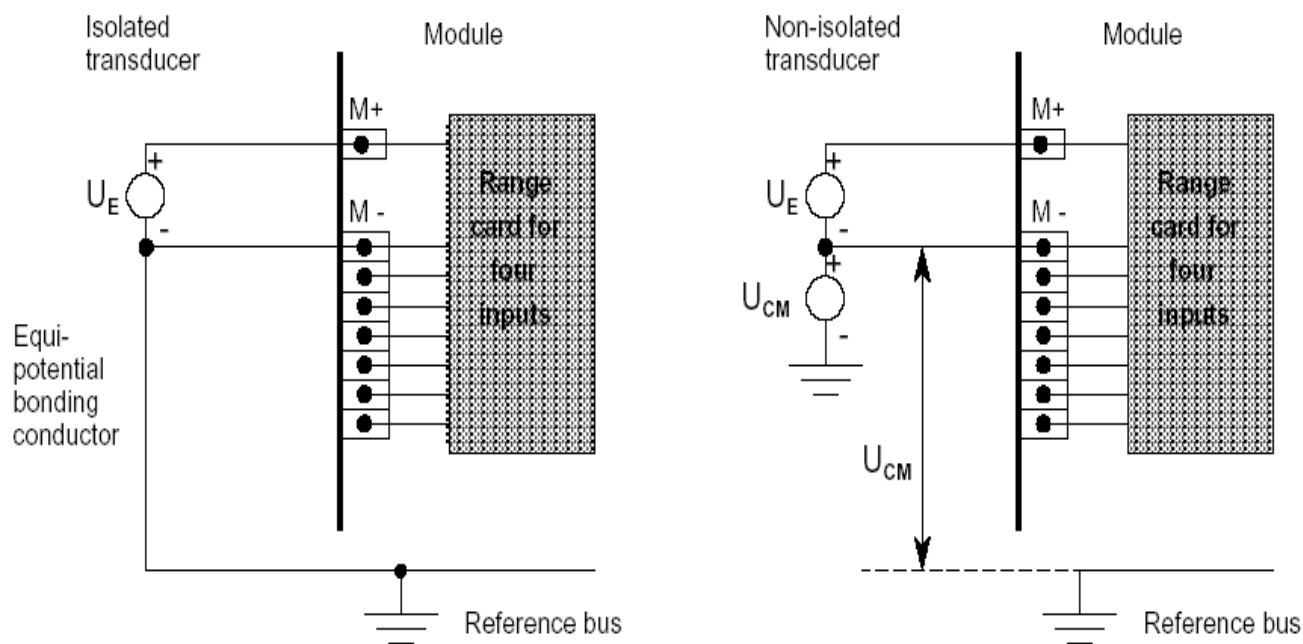
شکل ۹-۸۴ : شماره پایه های کارت ورودی آنالوگ 460-7LA12

### نحوه اتصال ترانسدیوسرها به ورودی آنالوگ :

ترانسدیوسر الکتریکی ، وسیله ای است که ورودی آن کمیتی غیر الکتریکی و خروجی آن الکتریکی است به عبارت دیگر وسیله ای است که کمیتی مانند دما، فشار ، رطوبت و ... را به کمیت‌های الکتریکی مانند ولتاژ و جریان تبدیل می کند .

ترانسدیوسرها در رنج های استاندارد ، می توانند خروجی های ولتاژی در محدوده صفر تا ده ولت ، و خروجی های جریان در محدوده ۴ تا ۲۰ میلی آمپر یا صفر تا ۲۰ میلی آمپر ایجاد نمایند . در مواقع وخیم ، ترانسدیوسرها ممکن است ، ولتاژی بیش از حد مجاز که می تواند به ماژول ورودی آنالوگ ، آسیب برساند را ایجاد می نمایند . در این موارد باید کارتهای ورودی آنالوگ را با استفاده از کانکتورهای هم پتانسیل که نمونه استفاده آنها برای ترانسدیوسرهای ایزوله شده و غیر ایزوله در شکل ۹-۸۵ ارائه شده است ، محافظت نمود .





شکل ۹-۸۵: نحوه اتصال ترانسدیوسرها به ورودی آنالوگ

□  $U_{CM}$  یک ولتاژ مجاز و تعریف شده است که در هنگام استفاده از ترانسدیوسرهای غیر ایزوله به عنوان ولتاژ تفاضلی بین ورودی ها و خط تغذیه مرجع ( زمین ) مورد استفاده قرار می گیرد .

### نحوه اتصال ترموکوپل به ورودی آنالوگ :

ترموکوپلها متداولترین اندازه گیرهای دما در صنعت می باشند از مزایای آنها می توان به سادگی ، ارزانی ، استحکام ، دوام و دقت مناسب اشاره نمود . از ترموکوپلها می توان در حوزه اندازه گیری نسبتا وسیع (۲۰۰- تا ۱۵۰۰+) درجه سانتیگراد و شرایط محیطی گوناگون استفاده کرد . ترموکوپلها دارای سرعت پاسخ دهی نسبتا خوب می باشند . این ترانسدیوسرها بر اساس پدیده ترمو الکتریک کار می کنند . طبق تعریف ، هر گاه دو فلز غیر همجنس تشکیل یک مدار بسته دهند ، جریان الکتریکی در مدار برقرار می شود این پدیده را اثر سیبک یا اثر ترمو الکتریک گویند .

انواع مختلفی از ترموکوپل ها در صنعت مورد استفاده قرار می گیرد ، که مشخصات مهمترین آنها در جدول شکل ۹-۸۶ ارائه شده است . برای جلوگیری از خوردگی و آلودگی که منجر به تغییر مشخصات ترموکوپل و اختلال در کار آن می شود و همچنین حفاظت مکانیکی آن ، معمولا ترموکوپل را در یک غلاف محافظ قرار می دهند . این غلاف همچنین امکان نصب ترموکوپل را در محل مورد نظر فراهم می آورد معمولا جنس غلاف متناسب با محیطی است که ترموکوپل در آنجا نصب می شود ولی متاسفانه استفاده از غلاف محافظ موجب افزایش ثابت زمانی ترموکوپل می گردد . محل و چگونگی نصب ترموکوپل دارای اهمیت فراوانی است چرا که نقاط مختلف و حتی نزدیک به هم در یک پروسه ممکن است دماهای مختلف داشته باشند.



Thermocouple		Maximum temperature	Defined up to	Positive limb	Negative limb
Fe-Con	J	750°C	1200°C	black	white
Cu-Con	T	350°C	400°C	brown	white
NiCr-Ni	K	1200°C	1370°C	green	white
NiCr-Con	E	900°C	1000°C	violet	white
NiCrSi-NiSi	N	1200°C	1300°C	mauve	white
Pt10Rh-Pt	S	1600°C	1540°C	orange	white
Pt13Rh-Pt	R	1600°C	1760°C	orange	white
Pt30Rh-Pt6Rh	B	1700°C	1820°C	no data	white

Table 1: Thermocouples to EN 60584

Thermocouple		Maximum temperature	Defined up to	Positive limb	Negative limb
Fe-Con	L	600°C	900°C	red	blue
Cu-Con	U	900°C	600°C	red	brown

Table 2: Thermocouples to DIN 43710

شکل ۹-۸۶: مشخصات انواع مختلف ترموکوپل

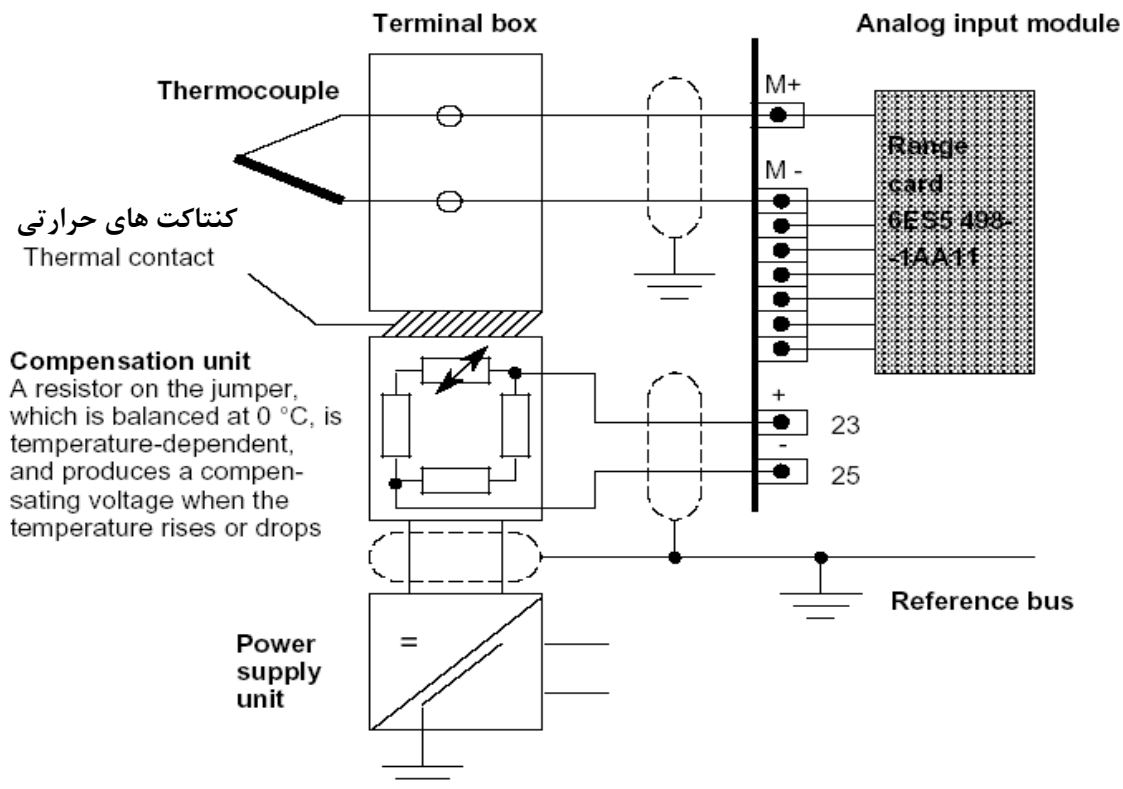
از آنجا که دمای محیط بر دقت اندازه گیری تاثیر گذار است، لذا برای ایجاد دقت بیشتر و کاهش اثر دمای محیط، می بایست از مدارات جبران کننده (Compensation Unit) استفاده نمود. این نمونه از این مدارات جبران کننده، که مدار داخلی و نحوه اتصال آن در شکل ۹-۸۷ ارائه شده است، از یک RTD (مقاومت حساس به حرارت از نوع PTC) که در یک مدار پل قرار گرفته است، تشکیل می گردد. از آنجا که خروجی مدار پل متناسب با تغییر مقاومت ها بوده و همچنین تغییر مقاومت ها نیز متناسب با دمای محیط می باشد، لذا در اثر تغییر دمای محیط یک ولتاژ متغیر از طریق پل ایجاد شده و به پایه های KOMP+ و KOMP- از کارت ورودی آنالوگ متصل می گردد. این ولتاژ به عنوان یک Offset برای کارت آنالوگ مورد استفاده قرار گرفته و اثر دمای محیط بر اندازه گیری را به حداقل می رساند.

در هنگام استفاده از واحدهای جبران کننده رعایت نکات زیر ضروری است:

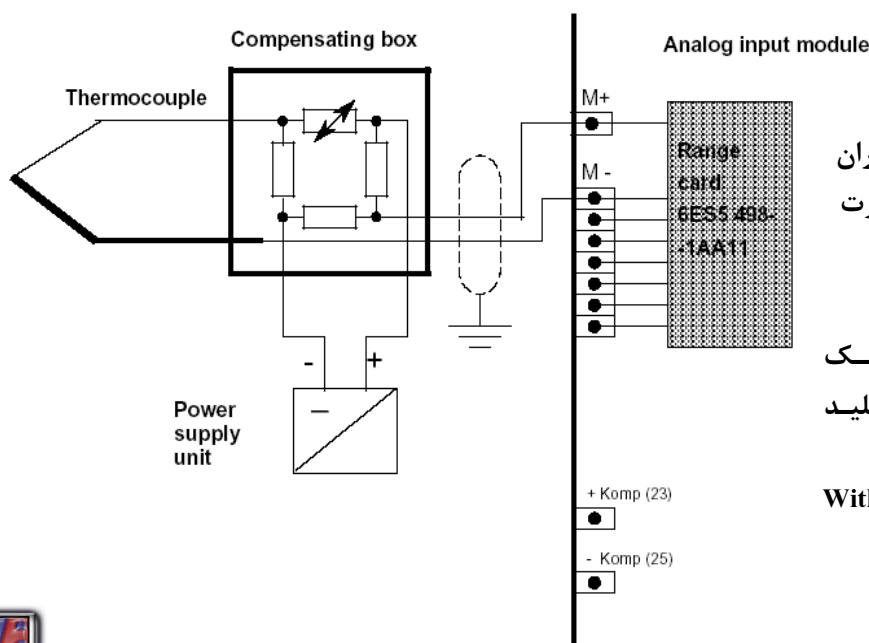
۱- در زمان استفاده از تعداد زیادی ورودی آنالوگ می بایست، برای هر ورودی از یک واحد جبران کننده استفاده شود.



- ۲- جبران کننده ها را می بایست با استفاده از ترمینال به کنتاكت های حرارتی اتصال داد .
- ۳- می بایست ، کلید تعیین عملگر بر روی کارت ورودی را برای فعال سازی جبران کننده ، تنظیم نمود .



شکل ۹-۸۷ : استفاده از واحد جبران کننده برای اتصال یک ترموکوپل به کارت ورودی آنالوگ



شکل ۹-۸۸ : نحوه استفاده از واحد جبران کننده برای اتصال چند ترموکوپل به کارت ورودی

- در این حال برای هر ورودی یک واحد جبران کننده استفاده شده و کلید تعیین عملگر را در حالت Without Reference junction Compensation تنظیم می نمایم .



### اتصال PT-100 به کارت ورودی آنالوگ AI-460

#### □ معرفی مقاومت‌های وابسته به دما

مقاومت‌های متغیر با دما به دو دسته کلی تقسیم می شوند :

۱- مقاومت‌هایی که با افزایش درجه حرارت ، مقاومتشان بیشتر می شود ( PTC )

۲- مقاومت‌هایی که با افزایش درجه حرارت ، مقاومتشان کاهش می یابد ( NTC )

تغییر مقاومت الکتریکی بر حسب دما می تواند برای اندازه گیری دما استفاده شود . هنگامیکه مقاومت الکتریکی بر حسب دما افزایش یابد ، به مقاومت عنصر دارای ضریب حرارتی مثبت یا به اصطلاح PTC می گویند برای مثال سنسور دمای پلاتین .

پلاتین متداولترین فلزی است که برای ساخت PTCها استفاده می شود تغییرات مقاومت بر حسب افزایش درجه حرارت در این فلز تقریباً خطی است ، ولی مشکل موجود این است که فلزات اساساً مقاومت الکتریکی کمی دارند بنابراین برای آشکار سازی تغییرات مقاومت الکتریکی بایستی اندازه مقاومت فلز در دمای صفر نسبتاً بزرگ باشد به همین دلیل با ساخت مقاومت بصورت سیم پیچ طول آنرا افزایش داده و مقاومت بزرگی می سازند . برای استفاده از این اثر برای اندازه گیری دما ، تغییرات مقاومت فلز بایستی بر اثر تغییرات دما زیاد بوده و مشخصه های فلز می بایست در طی عملکرد تغییر نماید .

یکی از سنسورهای پر کاربرد که از پلاتین ساخته می شود سنسور PT100 است یعنی اینکه در دمای صفر درجه مقدار مقاومت اهمی سنسور ۱۰۰ اهم است. برای رنج دمایی ۰ تا ۲۰۰ با استفاده از چند جمله ای زیر می توان مقدار مقاومت سنسور PT100 را در دماهای مختلف بدست آورد .

$$R(t) = R_0 ( 1 + A \times t + B \times t^2 + C \times (t - 100^\circ C) \times t^3 )$$

که در آن  $A = 3.9083 \times 10^{-3} \text{ } ^\circ C^{-1}$  و  $B = -5.775 \times 10^{-7} \text{ } ^\circ C^{-2}$  و  $C = -4.183 \times 10^{-12} \text{ } ^\circ C^{-4}$  می باشند .  
برای رنج دمایی بین ۰ تا ۸۵۰ درجه می توانیم از رابطه زیر استفاده کرده و مقاومت سنسور را در هر دمایی محاسبه کنیم .

$$R(t) = R_0 ( 1 + A \times t + B \times t^2 )$$

$R_0$  مقدار مقاومت سنسور در دمای صفر درجه است که این مقدار برای سنسور PT100 برابر با صد اهم و برای PT500 و PT1000 بترتیب برابر با ۵۰۰ و ۱۰۰۰ اهم است . برتری سنسورهای PT500 و PT1000 نسبت به PT100 این است که حساسیت آنها بیشتر می باشد . به عبارت دیگر تغییرات مقاومت آنها بر حسب تغییرات درجه حرارت شدید تر است . بطور مثال برای دماهای بین صفر تا صد درجه می توانیم از رابطه تقریبی زیر برای محاسبه مقدار مقاومت سنسورهای مختلف استفاده کنیم .

For PT100 0.4 ohm/°c

For PT500 2 ohm/°c

For PT1000 4 ohm/°c



یکی از مشکلات سنسورهای مقاومتی این است که در صورت برقراری ارتباط بین سنسور و کنترلر با دو سیم مقدار مقاومت مسیر ارتباطی نیز بعنوان مقاومت با مقاومت سنسور جمع شده و باعث ایجاد خطا در اندازه گیری می گردد. البته چون این خطا، خطای ثابت جمع شونده است می توانیم این خطا را برطرف نماییم. یکی از راههای کالیبره کردن کنترلر برای مسیر ارتباطی خاصی این است که یک مقاومت ۱۰۰ اهم دقیق به جای سنسور وصل نماییم. در این حالت کنترلر می بایست مقدار صفر را نشان دهد در غیر اینصورت بایستی کنترلر را کالیبره نماییم. راه دیگر از بین بردن این اثر استفاده از سنسورهای سه سیمه است که از سیم سوم جریانی نمی گذرد و فقط برای اندازه گیری ولتاژ مورد استفاده قرار می گیرد.

از آنجا که تنها راه اندازه گیری دما با استفاده از این سنسورها عبور یک جریان ثابت از سنسور و در نهایت ایجاد یک ولتاژ متناسب با مقاومت آن و در نتیجه متناسب با دمای محیط می باشد و همچنین با توجه به این امر که مقاومت یک جسم با حرارت تغییر می نماید، لذا این امر موجب ایجاد مساله **Self heating** یا خود گرمایی می شود و این مساله باعث ایجاد خطا در اندازه گیری می شود.

سازندگان ترمومتر معمولاً عددی را بعنوان ضریب **Self heating** تعریف می کنند که نشان می دهد این سنسور خاص به ازای چه مقدار تلفات توان چقدر افزایش درجه حرارت دارد. که این مسئله را می بایست در زمان کالیبره نمودن سیستم رعایت نمود.

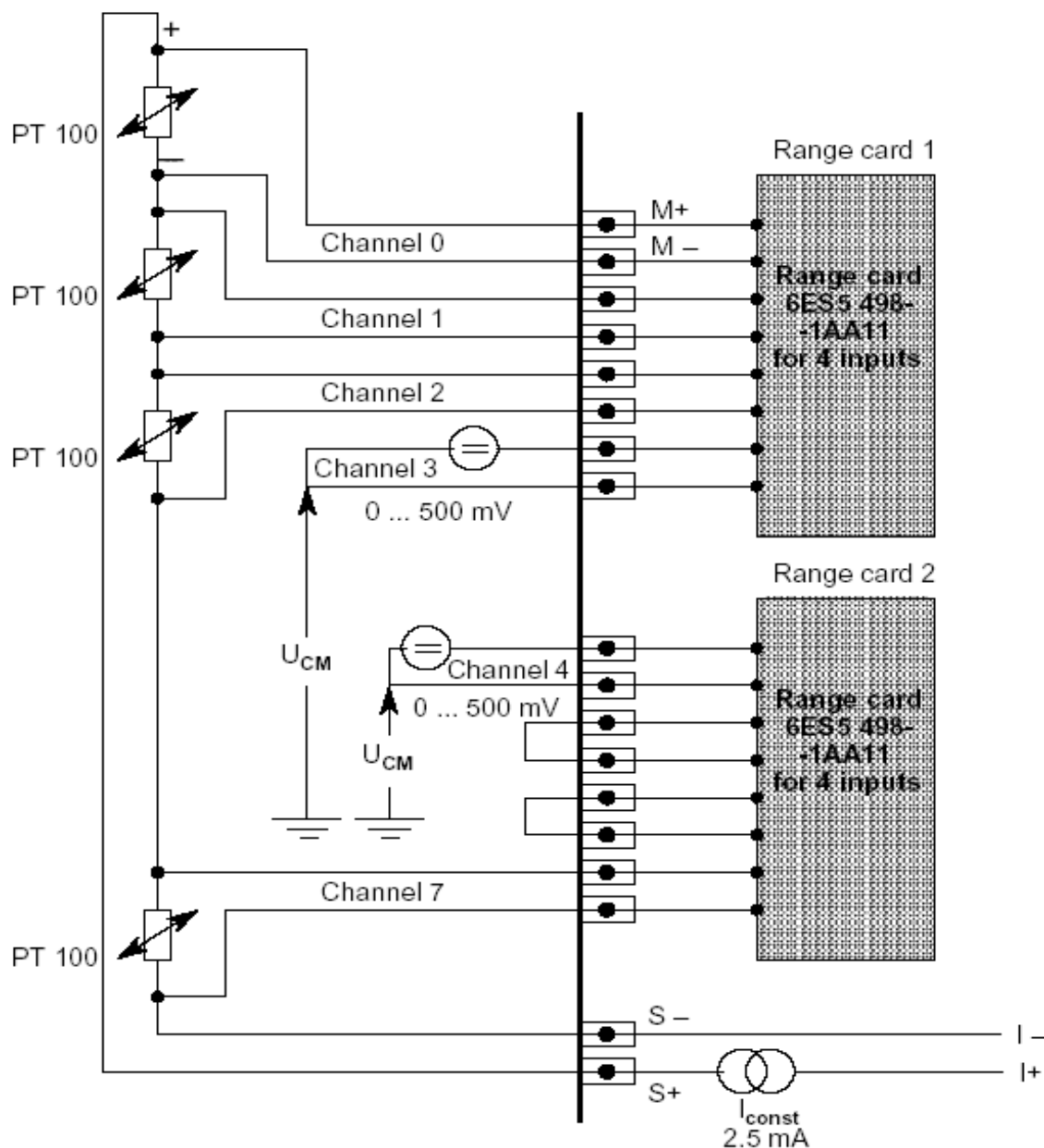
### ترمیستورها :

ترمیستورها معروفترین سنسورهای دما با شیب منفی می باشند. ترمیستورهای امروزی قادر به اندازه گیری دما های پایین حدود ۱۰۰- تا دماهای ۵۰۰ درجه می باشند شیب مشخصه ترمیستورها بسیار بیشتر از عناصر PTC می باشد و این به معنی حساسیت بیشتر و اندازه گیری دقیقتر می باشد درصد تغییرات مقاومت ترمیستور در اثر یک درجه تغییر دما چیزی در حدود ۳ الی ۵ درصد می باشد که در مقایسه با عدد ۰٫۴ درصد برای فلزات PTC عددی قابل توجه است. ترمیستور را می توان با مقاومتی حدود چند ده کیلو اهم ساخت در حالیکه مقاومتهای فلزی تا حدود چند صد اهم ساخته می شوند بنابراین اثر سیمهای ارتباطی در ترمیستورها اساساً منتفی می گردد در حالیکه در مورد سنسورهای PTC این اثر مشکل قابل توجهی را بوجود می آورد.

در برابر این مزایا مشخصه ترمیستورها بسیار غیر خطی تر از فلزات بوده و از طرف دیگر نیز حوزه اندازه گیری آنها کوچکتر از فلزات می باشد. از مشکلات دیگر ترمیستورها وجود تولهانس در آنهاست بطوریکه ترمیستورهای ساخت یک کمپانی و تحت یک نام و شماره از نظر رفتار و مشخصه با یکدیگر تفاوت دارند و این امر هنگام تعویض و جایجایی عناصر اندازه گیر مشکل آفرین خواهد شد. ترمیستورها بدلیل حجم کم و ثابت زمانی کوچک برای اندازه گیریهای سریع و در محلهایی که جای کمی برای نصب وجود دارد بسیار مناسب می باشند از این رو در وسایل و تجهیزات پزشکی بسیار مورد استفاده قرار می گیرند. پدیده خود گرمایی در ترمیستورها حادث است، زیرا ترمیستورها معمولاً دارای مقاومت بزرگتری بوده و بنا بر این عبور یک جریان مشخص، حرارت بیشتری در آنها تولید می کند و بعلاوه به دلیل کوچک بودن ثابت زمانی دمایی آنها در اثر این پدیده سریعتر تغییر می کند. تغییر ویژگیهای ذاتی و فیزیکی اندازه گیر با مرور زمان نیز بایستی مورد توجه قرار گیرد چرا که عنصر اندازه گیر در اثر تحمل حرارت، یا در اثر فعل و انفعال شیمیایی تدریجی با محیطی که با آن در تماس است ویژگیهای اولیه خود را بتدریج از دست می دهد.



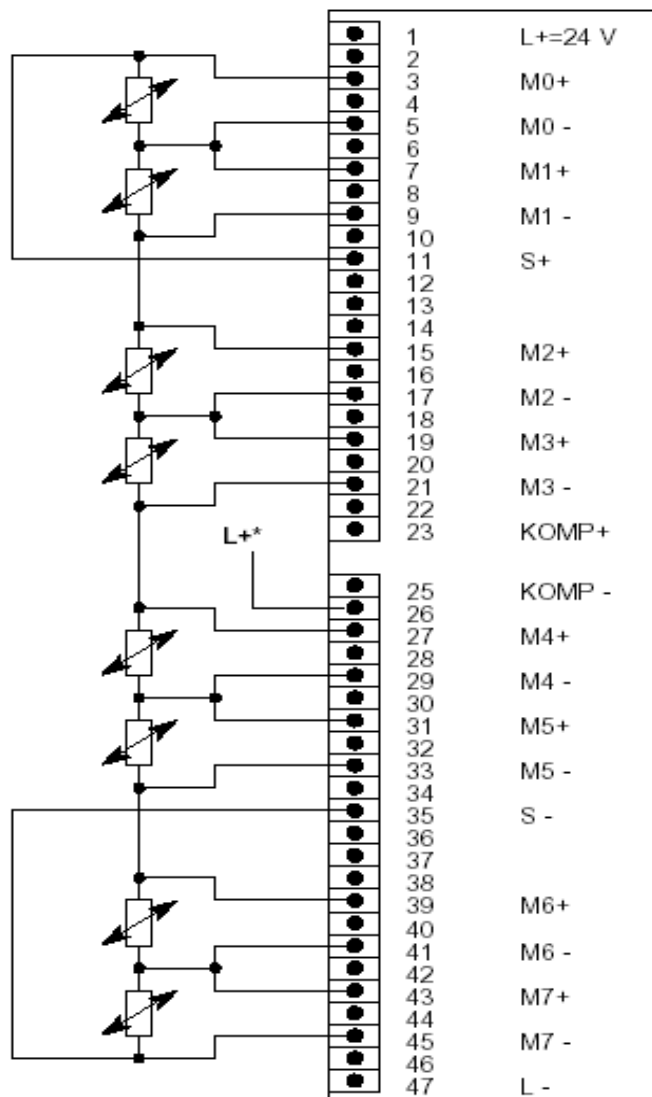
برای اتصال PT-100 به یک کارت ورودی آنالوگ با استفاده از منبع جریان ثابت  $2.5\text{ mA}$  که از طریق پایه های S+ و S- در دسترس می باشد، یک جریان ثابت برای تغذیه سنسورها ایجاد می گردد. همانگونه که در شکل ۸۹-۹ نمایش داده شده است، در کارت ورودی آنالوگ 460، ورودی های بدون استفاده می بایست اتصال کوتاه شود که این امر در کارت 498 نیازی نمی باشد.



شکل ۸۹-۹: نحوه اتصال سنسور دمای PT-100 به کارت ورودی آنالوگ



نحوه اتصال PT-100 به پین های مختلف کارت ورودی آنالوگ 460 نیز در شکل ۹-۹۰ ارائه شده است .



6ES5 460-7LA12

شکل ۹-۹۰ : نحوه اتصال و سیم بندی سنسور دمای PT-100 به کارت ورودی آنالوگ

توجه : جهت اطلاع از نحوه اتصال سایر ترانسدیوسرها به کارتهای ورودی آنالوگ به فایل های PDF مربوط به Manual این PLC مراجعه گردد .

















به کارتهای ورودی آنالوگ Range Card هایی جهت اتصال سنسورهای گوناگون و در نتیجه تغییر مقادیر جریان، ولتاژ و ... به مقادیر قابل استفاده برای PLC، متصل می شوند، که می توان با توجه به نیاز نسبت به انتخاب کارت مخصوص اقدام نمود. انواع مختلف این کارتها در جدول شکل ۹-۹۱ ارائه شده است.

Range card 6ES5 498	Circuitry (4 times each)	Function 500 mV/mA/PT100	Function 50 mV
- 1AA11		$\pm 500 \text{ mV};$ PT 100	$\pm 50 \text{ mV}$
- 1AA21		$\pm 1 \text{ V}$	$\pm 100 \text{ mV}^*$
- 1AA31		$\pm 10 \text{ V}$	$\pm 1 \text{ V}^*$
- 1AA41		$\pm 20 \text{ mA}$	$\pm 2 \text{ mA}^*$
- 1AA51**		+4 ... +20 mA two-wire transducer	
- 1AA61		$\pm 5 \text{ V}$	$\pm 500 \text{ mV}^*$
- 1AA71		+4 ... +20 mA four-wire transducer	

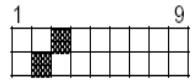
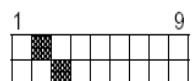
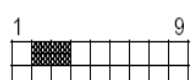
شکل ۹-۹۱: انواع Range Card های قابل اتصال به کارت ورودی آنالوگ و مدار داخلی آنها



بر روی کارتهای ورودی آنالوگ DIP Switch هایی جهت تنظیم عملکرد کارت موجود است که می توان از آنها جهت انتخاب حالت نمونه برداری ترتیبی ، تنظیم فرکانس نمونه برداری ، تعیین علامت دار بودن اعداد قرائت شده و... استفاده نمود . تعدادی از این سوئیچ ها در شکل های ۹-۹۲ الی ۹-۹۴ معرفی گشته اند .

Measuring Range	Switch Positions
0 - 20 mA	 ON OFF
0 - 1.25 V	
0 - 2.5 V	
0 - 5 V	
0 - 10 V	
±20 mA	
±1.25 V	
±2.5 V	
±5 V	
±10 V	
4 - 20 mA	
1 - 5 V	

شکل ۹-۹۲: تنظیم سوئیچ موقعیت (سوئیچ های S5 تا S8) برای تنظیم رنج اندازه گیری برای یک گروه

Data Format	Switch Position S 9
نمایش مکمل ۲	 ON OFF
نمایش با علامت	 ON OFF
نمایش باینری	 ON OFF

شکل ۹-۹۳: تنظیم سوئیچ S9 جهت تعیین فرمت اطلاعات ( پس از تبدیل مقدار ورودی به دیجیتال )



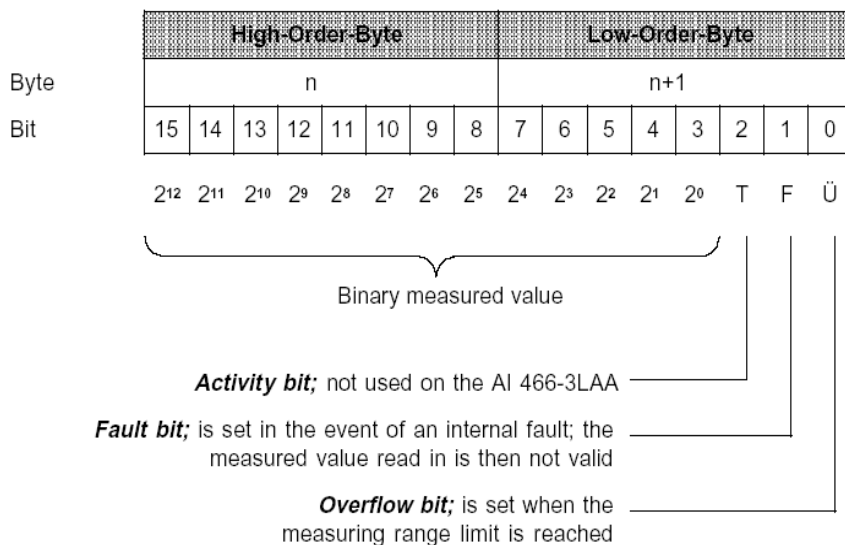
Module Starting Address (P area)	Switch Position S 9																											
128 (F080H)	<table border="1"> <tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>	1	2	3	4	5	6	7	8	9																		
1	2	3	4	5	6	7	8	9																				
144* (F090H)	<table border="1"> <tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>	1	2	3	4	5	6	7	8	9																		
1	2	3	4	5	6	7	8	9																				
160 (F0A0H)	<table border="1"> <tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>	1	2	3	4	5	6	7	8	9																		
1	2	3	4	5	6	7	8	9																				
176* (F0B0H)	<table border="1"> <tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>	1	2	3	4	5	6	7	8	9																		
1	2	3	4	5	6	7	8	9																				
192 (F0C0H)	<table border="1"> <tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>	1	2	3	4	5	6	7	8	9																		
1	2	3	4	5	6	7	8	9																				
208* (F0D0H)	<table border="1"> <tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>	1	2	3	4	5	6	7	8	9																		
1	2	3	4	5	6	7	8	9																				
224 (F0E0H)	<table border="1"> <tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>	1	2	3	4	5	6	7	8	9																		
1	2	3	4	5	6	7	8	9																				
240* (F0F0H)	<table border="1"> <tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>	1	2	3	4	5	6	7	8	9																		
1	2	3	4	5	6	7	8	9																				

شکل ۹-۹۴: تنظیم سوئیچ S9 جهت تعیین آدرس شروع کارت ورودی آنالوگ

**طرز نمایش مقادیر تبدیل شده به دیجیتال**

مقادیر آنالوگ خوانده شده پس از تبدیل به یک مقدار دیجیتال در دو بایت که محل آنها در RAM توسط آدرس کانال ورودی مورد نظر مشخص می گردد، ذخیره می شوند. شکل ۹-۹۵ طرز قرار گرفتن اطلاعات را در این دو

بایت نمایش می دهد.

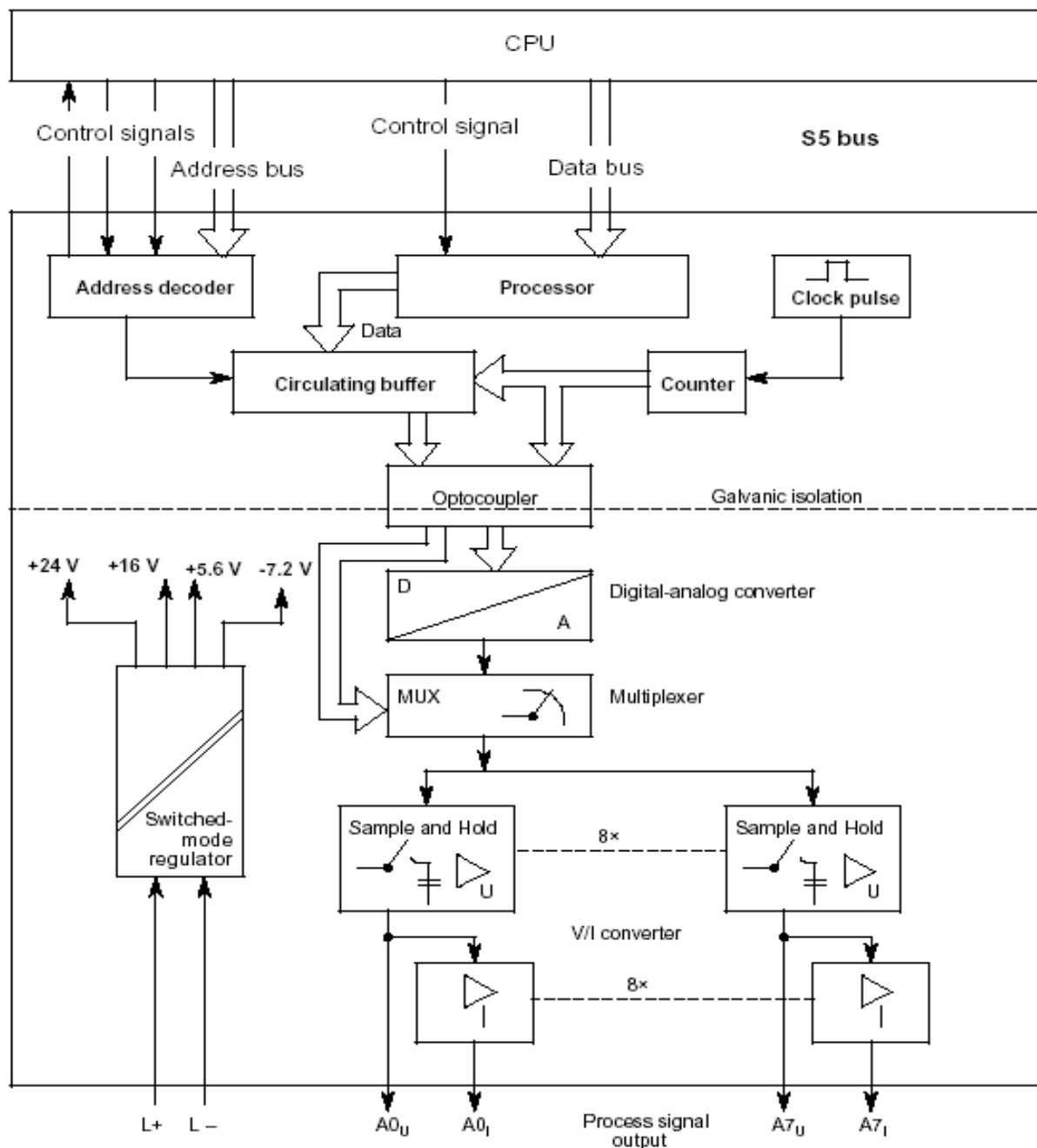


شکل ۹-۹۵: طرز نمایش مقادیر ورودی پس از تبدیل به مقدار دیجیتال متناظر



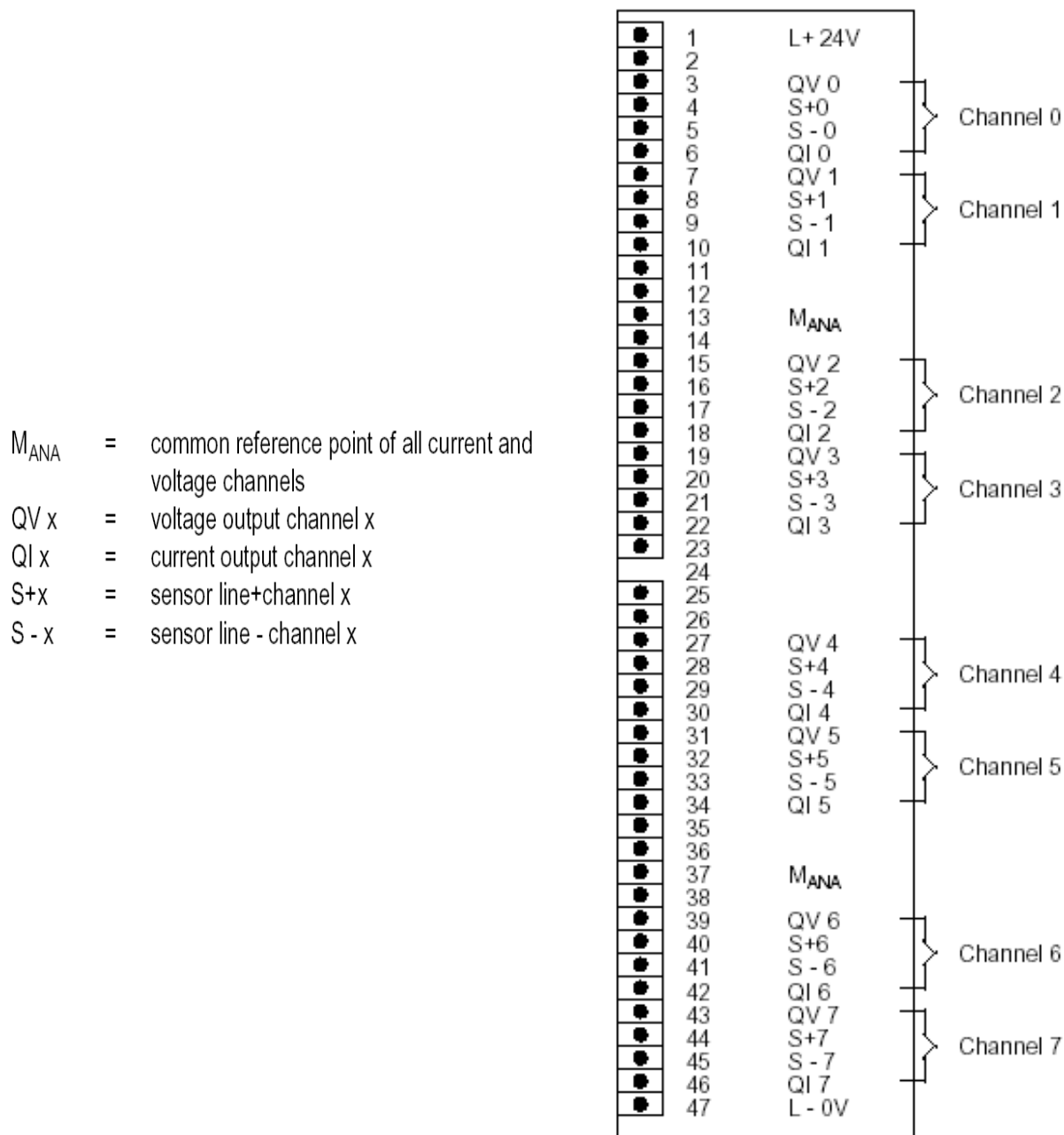
### کارت خروجی آنالوگ 470:

این کارت سیگنالهای دیجیتال تولید شده توسط CPU را به سیگنال آنالوگ مورد نیاز خروجی ها تبدیل می نماید. شکل ۹-۹۶ بلوک دیاگرام کارت خروجی آنالوگ را نمایش می دهد.



شکل ۹-۹۶: بلوک دیاگرام یک کارت خروجی آنالوگ





شکل ۹-۹۷: شماره پایه های کارت خروجی

این کارت دارای ۸ خروجی آنالوگ -10 تا +10 ولت یا صفر تا 20mA می باشد. حداقل مقاومت بار قابل استفاده در حالت ولتاژی و 3.3 KΩ و ماکزیمم مقاومت قابل استفاده در حالت جریانی ۳۰۰ اهم می باشد. حداکثر مسافت قابل انتقال خروجی ها با استفاده از کابل محافظ دار ۲۰۰ متر تعریف گردیده است. شکل ۹-۹۸ انواع مختلف کارت خروجی آنالوگ 470 را نمایش می دهد.



### 470-7LA analog output module

for S5-115U/H/F

8 outputs;  $\pm 10$  V/0 ... 20 mA

### 470-7LB analog output module

for S5-115U/H/F

8 outputs;  $\pm 10$  V

### 470-LC analog output module

for S5-115U/H/F

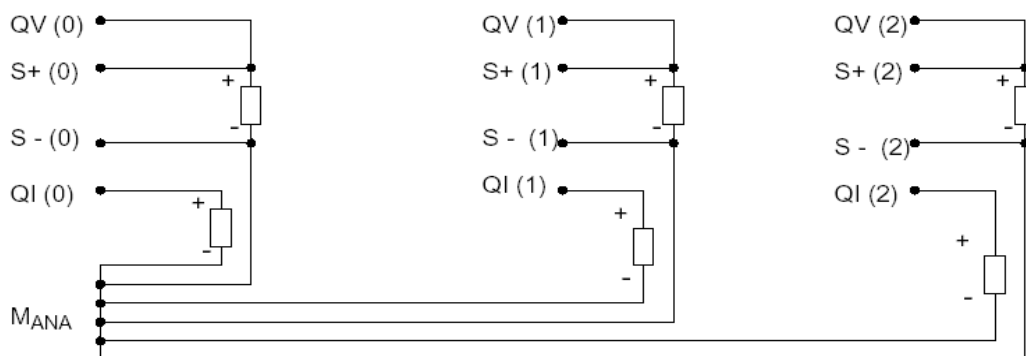
8 outputs;

+ 1 ... 5 V; + 4 ... 20 mA

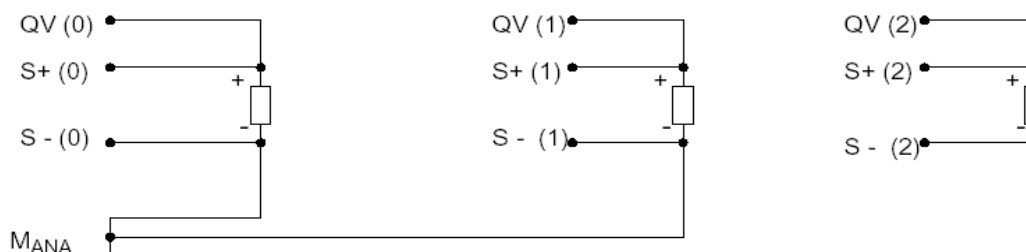
شکل ۹-۹۸: انواع کارت خروجی آنالوگ 470 با سطوح ولتاژ و جریان خروجی هر

#### نحوه اتصال بار به خروجی های آنالوگ :

مدارات ارائه شده در شکل ۹-۹۹ نحوه اتصال بار به کارت خروجی آنالوگ با خروجی های جریانی و ولتاژی را نمایش می دهد. رعایت این نکته ضروری است که به هنگام استفاده خروجی جریانی می بایست با استفاده از جامپر QV(x) را به S+(x) و S-(x) را به  $M_{ANA}$  متصل نمود.



6ES5 470-7LAxx  
6ES5 470-7LCxx



6ES5 470-7LBxx

شکل ۹-۹۹: نحوه اتصال بار به خروجی های آنالوگ



آموزشگاه برق و الکترونیک صنعتی اندیشه سازان صنعت مجری برگزاری دوره های آموزشی از مقدماتی تا پیشرفته در زمینه های برق، الکترونیک، اتوماسیون صنعتی، ابزار دقیق و کامپیوتر - با مجوز رسمی از سازمان آموزش فنی و حرفه ای کشور و وزارت کار و امور اجتماعی - اعطای مدرک با کد بین المللی I.L.O. مشهد مقدس - ۰۵۱۱-۷۶۴۴۰۷۸-۷۶۱۸۲۸۶

### نحوه خواندن و مقیاس نمودن ورودی های آنالوگ

بلوک عملگر FB250، مقادیر آنالوگ را از ماژول ورودی آنالوگ خوانده و متناسب با آن یک خروجی با نام XA در رنج معین ایجاد می نماید. تعیین رنج مورد نظر در این بلوک با مقداردهی دو متغیر به نامهای OGR برای حد بالا و UGR جهت حد پایین انجام می پذیرد. در این بلوک می بایست، نوع و مشخصات کانال آنالوگ مورد استفاده را با مقداردهی پارامتری به نام KNKT، معرفی نمود. در این بلوک پارامتری به نام BU و بصورت خروجی بییتی تعریف شده است که در زمان گذشتن مقدار ورودی آنالوگ از یک سطح مجاز تعریف شده، این بییت یک خواهد شد. جدول ارائه شده در شکل ۹-۱۰۰، پارامترهای مختلف قابل استفاده در بلوک عملگر FB250 را نمایش می دهد.

Parameter	Description	Type	Data Type	Assignment	STL
BG	آدرس ماژول	D	KF	128 ... 224	: JU FB 250
KNKT	KN : شماره کانال KT : نوع کانال	D	KY	KY =x,y x =0 to 15 y =3 to 6 نمایش مطلق مقادیر ( صفر تا ۲۰ میلی آمپر ) : 3: نمایش مقادیر بصورت مثبت : 4: نمایش مقادیر بصورت مثبت و منفی : 5: نمایش مقادیر بصورت مکمل ۲ : 6:	NAME : RLG:AE BG : KNKT : OGR : UGR : EINZ : XA : FB : BU : TBIT :
OGR	حد بالای مقدار خروجی کارت	D	KF	- 32768 to +32767	
UGR	حد پایین مقدار خروجی کارت	D	KF	- 32768 to +32767	
EINZ	نمونه گیری انتخابی	I	BI	در صورت یک بودن این بییت نمونه گیری بصورت انتخابی انجام می گیرد .	
XA	خروجی کارت	Q	W	مقدار خروجی استاندارد شده که بصورت یک کلمه دیجیتال در دسترس می باشد	
FB	بییت خطا	Q	BI	این بییت زمانی یک می گردد که شماره کانال یا اسلات و یا تعریف کانال بصورت نادرست انجام شده باشد .	
BU	بییت تجاوز از حد مجاز	Q	BI	زمانی که مقدار ورودی آنالوگ از حد مجاز بیشتر باشد این بییت یک می گردد	
TBIT	بییت نمایش فعال بودن بلوک عملگر	Q	BI	هر گاه بلوک عملگر در حال نمونه گیری انتخابی فعال باشد، این بییت یک است .	

شکل ۹-۱۰۰: پارامترهای مختلف بلوک عملگر FB250 جهت خواندن مقادیر ورودی آنالوگ



بلوک عملگر FB250 ، مقدار خوانده شده ورودی را جهت تطبیق با حدود بالا و پایین تعیین شده با استفاده از فرمول زیر به مقدار XA تبدیل می نماید .

برای کانال نوع ۳ : ( مقدار مطلق ۴ تا ۲۰ میلی آمپر )

$$X_A = \frac{UGR.(2560 - x_e) + OGR.(x_e - 512)}{2048}$$

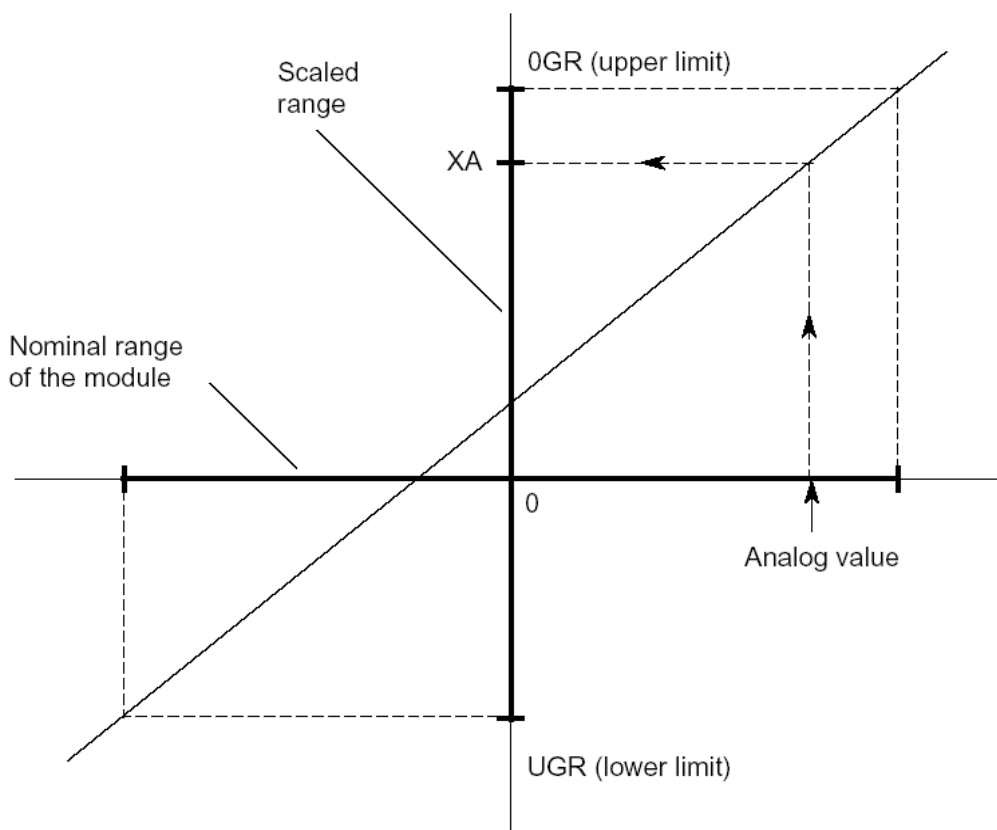
برای کانال نوع ۴ : ( مقدار مثبت )

$$X_A = \frac{UGR.(2048 - x_e) + OGR.x_e}{2048}$$

برای کانال نوع ۵ و ۶ ( مقادیر مثبت و منفی )

$$X_A = \frac{UGR.(2048 - x_e) + OGR.(x_e + 2048)}{4096}$$

در فرمول های فوق XA خروجی تبدیل شده در بلوک FB250 بوده و  $x_e$  مقدار آنالوگ خوانده شده از ماژول ورودی آنالوگ می باشد . شکل ۹-۱۰۱ این تبدیل را بصورت شماتیک نمایش می دهد .



شکل ۹-۱۰۱: نمایش تبدیل مقدار ورودی آنالوگ به یک مقدار نرمال شده





بلوک عملگر FB250 اجازه می دهد که نمونه گیری مقادیر آنالوگ بصورت انتخابی صورت پذیرد. این عمل با فعال نمودن پارامتر EINZ صورت می پذیرد. در طی عمل تبدیل اطلاعات که در حدود ۶۰ میلی ثانیه بطول می انجامد، اجازه نمونه گیری دیگری داده نمی شود. در نتیجه این عمل FB250 پارامتر TBIT را بمنزله فعال بودن بلوک یک می نماید. با اتمام نمونه گیری انتخابی TBIT یک می گردد.

### تعیین خروجی های آنالوگ با استفاده از بلوک عملگر FB251

بلوک عملگر FB251 برای ایجاد یک خروجی آنالوگ توسط ماژول خروجی آنالوگ مورد استفاده قرار می گیرد. در این بلوک، مقدار آنالوگ خروجی با استفاده از روابط زیر تعیین می گردد. برای کانال نوع صفر:

$$X_a = \frac{1024.(XE - UGR)}{OGR - UGR}$$

برای کانال نوع یک:

$$X_a = \frac{1024.(2.XE - OGR - UGR)}{OGR - UGR}$$

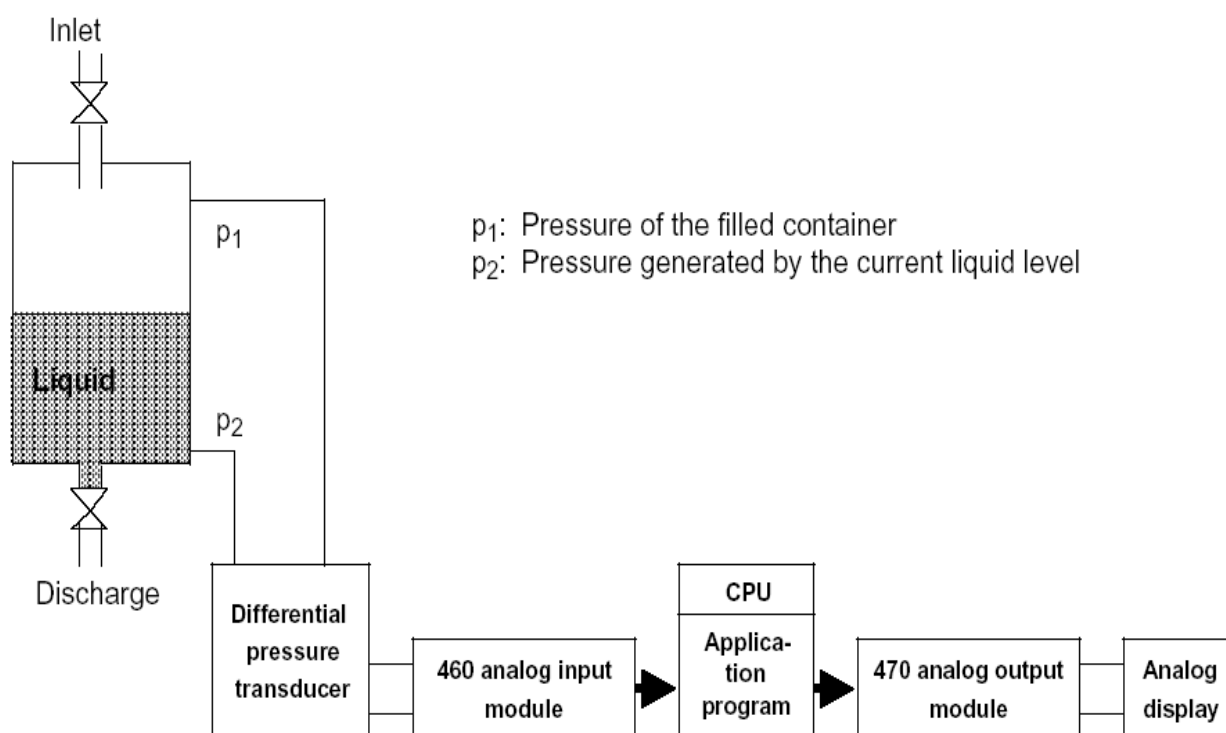
در روابط فوق XE مقدار دیجیتال تعیین شده در بلوک عملگر و Xa مقدار خروجی آنالوگ ماژول می باشد. جدول ارائه شده در شکل ۹-۱۰۲ پارامترهای مربوط به بلوک عملگر FB251 را نمایش می دهد.

Parameter	Meaning	Type	Data Type	Assignment	STL
XE	Analog value to be output	I	W	Input value (fixed-point) in the UGR to OGR range	: JU FB 251 NAME : RLG:AA XE :
BG	Module address	D	KF	128 to 240	BG : KNKT :
KNKT	KN= channel number KT= channel type	D	KY	KY =x,y x =0 to 7 y =0;1 0: unipolar representation 1: bipolar fixed-point number	OGR : UGR : FEH : BU :
OGR	Upper limit of the output value	D	KF	- 32768 to +32767	
UGR	Lower limit of the output value	D	KF	- 32768 to +32767	
FEH	Error when setting the limit value	Q	BI	"1" if UGR=OGR, for illegal channel or slot number, or illegal channel type	
BU	Analog value to be output exceeds UGR or OGR	Q	BI	At "1, "XE is outside the range (UGR;OGR). XE assumes the limit value	

شکل ۹-۱۰۲: پارامترهای مختلف بلوک عملگر FB251 جهت تولید خروجی آنالوگ



**مثال:** یک مخزن در بسته محتوی یک مایع را مطابق شکل ۹-۱۰۳ در نظر بگیرید. برنامه PLC می بایست قابلیت خواندن سطح مایع در زمان مورد نیاز و نمایش آن بر روی یک ابزار نمایشگر را داشته باشد. بارسیدن سطح مایع به یک حد مشخص، یک پرچم داخلی یک می گردد. یک ترانسدیوسر فشار تفاضلی صفر تا ۲۰ میلی آمپر، سیگنال بیان کننده سطح مایع را (بین صفر تا ۱ متر) به ماژول ورودی آنالوگ 460 ارسال می نماید. ماژول ورودی آنالوگ این مقدار خوانده شده را به یک واحد دیجیتال (واحدهای صفر تا ۲۰۴۸) که می تواند بوسیله برنامه های کاربردی S5-115U پردازش گردد، تبدیل می نماید. برنامه کاربردی مقادیر خوانده شده را با مقادیر حدی مجاز مقایسه کرده و در صورت نیاز پرچم عنوان شده فوق را یک نموده و این مقدار را به یک کارت خروجی آنالوگ 470 ارسال می کند. کارت خروجی آنالوگ این مقدار را به یک حد صفر تا ده ولت تبدیل کرده، که در نتیجه عقربه نمایشگر آنالوگ به میزان این ولتاژ تغییر می نماید.

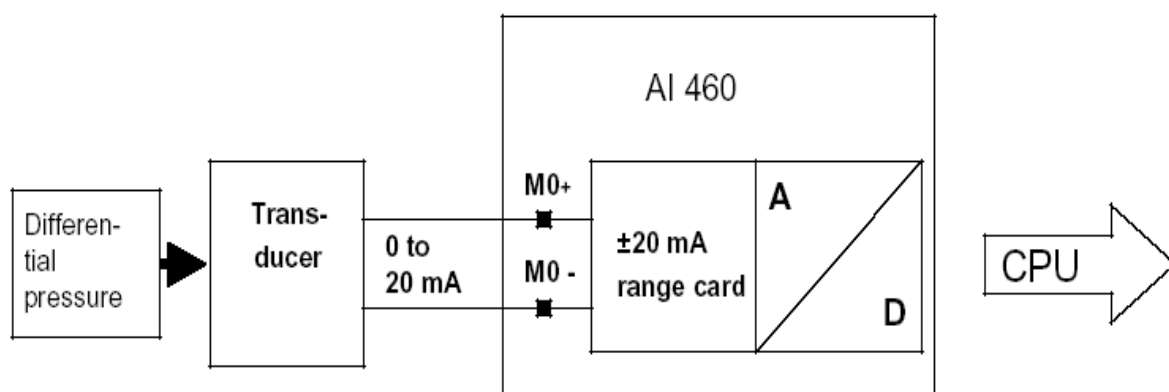


شکل ۹-۱۰۳: شکل مربوط به مثال ورودی - خروجی آنالوگ

### شرح پروسه:

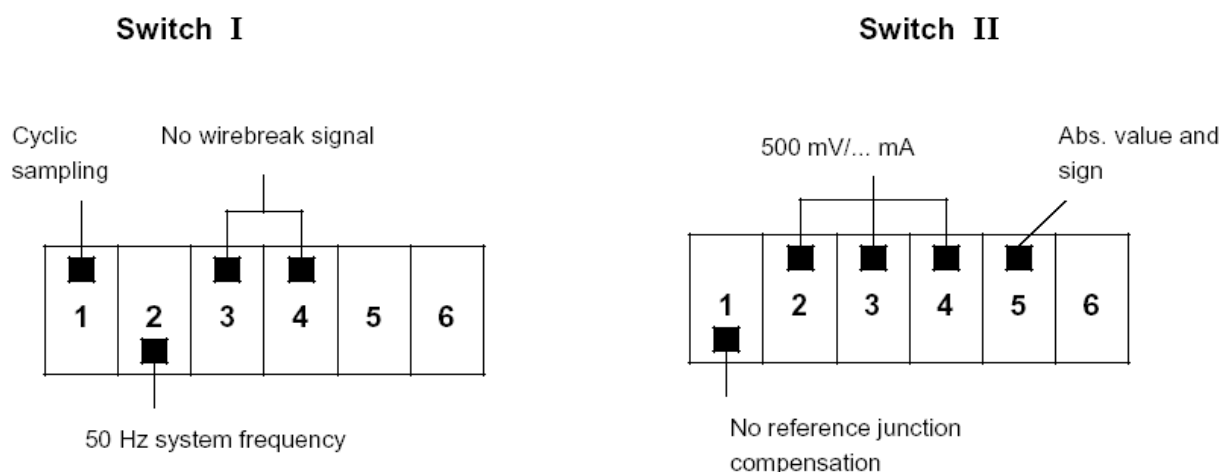
ترانسدیوسر را همانند شکل ۹-۱۰۴ و بطور مستقیم به فرانت کانکتور AI460 (ترمینال های M0+ و M0-) متصل نمایید. در این حال تنظیمات ترانسدیوسر را به نحوی انجام می دهیم که در ارتفاع صفر متر خروجی ترانسدیوسر صفر میلی آمپر و در ارتفاع ۱۰ متر خروجی ۲۰ میلی آمپر گردد. یک Range card (صفر تا ۲۰ میلی آمپر) نیز بر روی کارت ورودی قرار دهید.





شکل ۹-۱۰۴: نحوه اتصال ترانسدیوسر فشار به ورودی آنالوگ

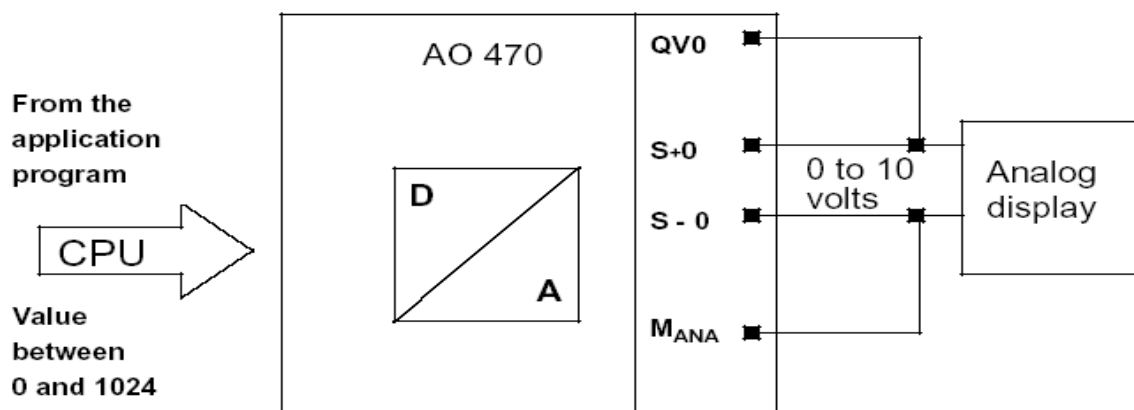
سوئیچ های تنظیم حالت را که در پشت کارت موجود است ، جهت تعیین شرایط کاری دلخواه ، بصورت ارائه شده در شکل ۹-۱۰۵ تنظیم نمایید .



شکل ۹-۱۰۵: نحوه تنظیم سوئیچ های مازول ورودی آنالوگ

کارت خروجی آنالوگ را نیز جهت نمایش سطح مایع بر روی یک نمایشگر آنالوگ ، همانند شکل ۹-۱۰۶ به سیستم متصل نمایید . در این حالت با ارسال اطلاعات دیجیتال بین صفر تا ۱۰۲۴ به کارت خروجی ، مقادیر ولتاژ متناسب و بین صفر تا ده ولت ، جهت تغییر عقربه نمایشگر آنالوگ تولید می گردد .





شکل ۹-۱۰۶: نحوه اتصال نمایشگر به ماژول خروجی آنالوگ

برنام کاربر شامل مراحل زیر می گردد:

- ۱- فراخوانی بلوک عملگر FB250 جهت تعیین رنج بین صفر تا ۱۰۰۰ سانتی متر (XA)
- ۲- تعیین مقادیر محدوده مجاز و مقایسه مقادیر جاری با آنها (PB9)
- ۳- یک نمودن F12.6 در فاصله ۹۰۰ سانتی متر
- ۴- فراخوانی FB251 جهت تعیین پارامتر XE به منظور نسبت دهی مقادیر دیجیتال بین صفر تا ۱۰۲۴ به فواصل صفر تا ۱۰۰۰ سانتی متر.

PB9 STL	Description
<pre> :L   KF +900 :L   FW 10 :&lt;=F :=F 12.6  :BE                     </pre>	<p>با توجه به تعریف پارامترهای مربوط به بلوک عملگر FB250 که در PB1 انجام گرفته و در ادامه آمده است ، مقدار اندازه گیری شده که خروجی FB250 نیز محسوب می گردد در FW10 قرار می گیرد .</p> <p>در PB9 این مقدار بطور دائم با عدد ثابت ۹۰۰ مقایسه شده و در صورت برقراری شرط <math>FW10 \geq 900</math> ، پرچم نمایشگر وضعیت با نام F12.6 یک می گردد .</p>

شکل ۹-۱۰۷: بلوک برنامه PB9 جهت مقایسه حدود



PB1 STL	Description
<pre> :JU FB 250 NAME :RLG:AE BG :KF +128  KNKT :KY 0,4 OGR :KF +1000 UGR :KF +0 EINZ :F 12.0  XA :FW 10 FB :F 12.1 BU :F 12.2 TBIT :F 12.3  :JU PB 9 :JU FB 251 NAME :RLG:AA XE :FW 10 BG :KF +160  KNKT :KY 0,0 OGR :KF +1000 UGR :KF +0 FEH :F 12.4 BU :F 12.5  :BE                     </pre>	<p>فراخوانی بلوک عملگر FB250 با تعیین نام و تعریف پارامترهای استفاده شده در آن .</p> <p>آدرس کارت ورودی آنالوگ با توجه به قرار گرفتن آن در اسلات مربوطه ۱۲۸ تعیین شده است .</p> <p>در تعریف پارامتر KNKT ، کانال صفر مربوط به کارت ورودی و با نوع مقادیر مثبت به عنوان کانال ورودی تعیین گشته است .</p> <p>رنج فیزیکی مورد اندازه گیری : <math>0 &lt; XA &lt; 1000</math> سانتی متر</p> <p>FW 10 مقدار خوانده شده از ورودی بوده که در پارامتر XA تعریف شده است .</p> <p>بیتهای خطای موجود در FB250 در پرچم های خاصی قرار گرفته است .</p> <p>پس از خواندن مقدار ورودی آنالوگ و نرمال کردن با استفاده از فرمول های تعریف شده در FB250 ، با پرش به PB9 مقایسه لازم در آنجا انجام گرفته و در صورت برقراری شرایط فرمان های لازم داده می شود .</p> <p>در ادامه با فراخوانی بلوک عملگر FB251 مقادیر خوانده شده از کارت ورودی که در FW10 قرار گرفته است ، بطور مستقیم به عنوان ورودی FB251 و نهایتاً جهت محاسبه مقادیر خروجی مورد استفاده قرار می گیرند .</p> <p>آدرس کارت خروجی آنالوگ نیز با توجه به سخت افزار سیستم ۱۶۰ تعیین شده است .</p>

شکل ۹-۱۰۸ : بلوک برنامه PB1 به عنوان بدنه اصلی برنامه



### بلوک های خاص :

این بلوک ها که برای انجام برخی اعمال خاص توسط شرکت سازنده تعیین و برنامه ریزی شده اند ، هیچ فضایی از حافظه برنامه را اشغال ننموده و همانند سایر بلوک ها مورد استفاده قرار می گیرند . جدول ارائه شده در شکل ۹-۱۰۹ این بلوک ها را معرفی می نماید .

Block		Call Length (in Words)	Execution time (in Millise-conds)	Function	
Type	No.				Name
FB	238	COMPR 1	4	< 0.6	Compress PLC memory
FB	239	DELETE	5	< 0.8	Delete block
FB	240	COD : B4	5	< 0.8	4-tetrad BCD code converter
FB	241	COD : 16	6	< 1.0	16-bit fixed-point converter
FB	242	MUL : 16	7	< 1.0	16-bit binary multiplier
FB	243	DIV : 16	10	< 2.3	16-bit binary divider
FB	244	SEND 2			Send data
FB	245	RECEIVE 2			Receive data
FB	246	FETCH		< 4	Fetch data
FB	247	CONTROL		approx. 0.6	Monitor job processing
FB	248	RESET		< 4	Delete job
FB	249	SYNCHRON		6.5 ms to 10 s	Initialize interface
FB	250	RLG : AE	11	2.0	Read analog value (see Chapter 10)
FB	251	RLG : AA	9	5.2	Output analog value (see Chapter 10)
OB	31			< 0.14	Restart scan time
OB	160				Variable time loop
OB	251			2.0	PID control algorithm
OB	254 3				Read digital inputs
OB	255 3				Transfer PIQ to outputs
DB	1				Parameterize internal functions

شکل ۹-۱۰۹: بلوک های خاص مورد استفاده در S5-115U

### الف: بلوک های تبدیل کد

بلوک های عملگر FB240 و FB241 جهت تبدیل کدهای BCD به باینری و بالعکس مورد استفاده قرار می گیرند. این بلوک ها و پارامترهای قابل تعریف در آنها در شکل های ۹-۱۱۰ و ۹-۱۱۱ ارائه شده است .



Parameter	Type	Data Type	Assignment	Meaning	STL
BCD	I	W	- 9999 to+9999	BCD number	: JU FB 240
SBCD	I	Bi	"1" for "-" "0" for "+"	Sign of the BCD number	Name : COD : B4 BCD :
BINARY	Q	W	16 bits "0" or "1"	Binary number	SBCD : BINARY:

شکل ۹-۱۱۰: بلوک عملگر FB240 جهت تبدیل کد BCD به باینری

Parameter	Type	Data Type	Assignment	Meaning	STL
BINARY	I	W	- 32768 to+32767	Binary number	: JU FB 241
SBCD	Q	Bi	"1" for "-" "0" for "+"	Sign of the BCD number	Name : COD : 16 BINARY:
BCD2	Q	By	2 tetrads	BCD number tetrads 4 and 5	SBCD :
BCD1	Q	W	4 tetrads	BCD number tetrads 0 to 3	BCD2 : BCD1 :

شکل ۹-۱۱۱: بلوک عملگر FB241 جهت تبدیل کد باینری به BCD

### ب: بلوک های عملیات ریاضی

بلوک عملگر FB242 برای انجام عملیات ضرب و بلوک عملگر FB243 برای انجام عملیات تقسیم تعریف گردیده اند. پارامترهای قابل تعریف برای این بلوک ها در اشکال ۹-۱۱۲ و ۹-۱۱۳ ارائه شده است.



Parameter	Type	Data Type	Assignment	Meaning	STL
Z1	I	W	- 32768 to+32767	Multiplier	: JU FB 242
Z2	I	W	- 32768 to+32767	Multiplicand	Name : MUL : 16
Z3=0	Q	Bi	"1", if the product is zero	Scan for zero	Z1 : Z2 :
Z32	Q	W	16 bits	Product, high-order word	Z3=0 : Z32 :
Z31	Q	W	16 bits	Product, low-order word	Z31 :

شکل ۹-۱۱۲: بلوک عملگر FB242 جهت انجام عملیات ضرب

Parameter	Type	Data Type	Contents	Meaning	STL
Z1	I	W	- 32768 to+32767	Dividend	: JU FB 243
Z2	I	W	- 32768 to+32767	Divisor	Name : DIV : 16
OV	Q	Bi	"1", if overflow	Overflow indicator	Z1 : Z2 :
FEH	Q	Bi	"1" for division by zero		OV : FEH :
Z3=0	Q	Bi	"0": quotient is zero	Scan for zero	Z3=0 : Z4=0 :
Z4=0	Q	Bi	"0": remainder is zero	Scan for zero	Z3 : Z4 :
Z3	Q	W	16 bits	Quotient	
Z4	Q	W	16 bits	Remainder	

**Dividend** : مقسوم

**Quotient** : خارج قسمت

**Divisor** : مقسوم علیه

**Remainder** : باقیمانده

شکل ۹-۱۱۲: بلوک عملگر FB243 جهت انجام عملیات تقسیم







آموزشگاه برق و الکترونیک صنعتی اندیشه سازان صنعت مجری برگزاری دوره های آموزشی از مقدماتی تا پیشرفته در زمینه های برق ، الکترونیک ، اتوماسیون صنعتی ، ابزار دقیق و کامپیوتر - با مجوز رسمی از سازمان آموزش فنی و حرفه ای کشور و وزارت کار و امور اجتماعی - اعطای مدرک با کد بین المللی I.L.O

مشهد مقدس - ۷۶۱۸۲۸۶ - ۷۶۴۴۰۷۸ - ۰۵۱۱



آموزشگاه برق و الکترونیک صنعتی اندیشه سازان صنعت مجری برگزاری دوره های آموزشی از مقدماتی تا پیشرفته در زمینه های برق ، الکترونیک ، اتوماسیون صنعتی ، ابزار دقیق و کامپیوتر- با مجوز رسمی از سازمان آموزش فنی و حرفه ای کشور و وزارت کار و امور اجتماعی- اعطای مدرک با کد بین المللی I.L.O

مشهد مقدس-۷۶۱۸۲۸۶-۷۶۴۴۰۷۸-۰۵۱۱