








- بررسی المان‌های بیت لاجیک، با ارائه مثال برای هر یک از المان‌ها شامل:
  - کنتاکت‌های NO و NC، المان Not، کوئیل‌های خروجی میانی و پایانی،
  - کوئیل‌های Set، Reset و کوئیل Save
  - R-S فیلپ فلاپ و S-R فیلپ فلاپ و تفاوت عملکرد آنها
  - آشکار کننده لبه بالا رونده و لبه پائین روندهء عملیات یک شبکه
  - آشکار کننده لبه بالا رونده و یا لبه پائین رونده یک آدرس با شرط اینکه ورودی دیگر المان مربوطه یک باشد
- بررسی انتقال داده با المان Move
- معرفی سیمولاتور PLC:

## ۳-۱) بررسی Bit Logic ها

مجموعه المان‌ها برای اجرای عملیات بیتی در جدول زیر نشان داده شده است. همه این المان‌ها دارای آدرس بیتی هستند. در ادامه پس از معرفی این المان‌ها مثالی برای کار برد هر یک از آن‌ها ارائه شده است.

مجموعه المانهای Bit logic در این فولدر قرار دارند	
ابزار ایجاد شبکه، با این ابزار می‌توان به تعداد مورد نیاز شبکه در برنامه ایجاد کرد	
کنتاکت NO، در صورتی که محتوای آدرس مربوطه یک منطقی باشد ارتباط بین دو پایه آن برقرار می‌شود	
کنتاکت NC، در صورتی که محتوای آدرس مربوطه صفر منطقی باشد ارتباط بین دو پایه آن برقرار می‌شود	
معکوس کننده (Not)، نتیجه عملیات ما قبل خود را معکوس می‌کند	
کوئیل خروجی، همواره محتوای آن برابر نتیجه عملیات شبکه متصل به آن است	
خروجی میانی، نتیجه عملیات ما قبل خودش را در آدرس مربوطه ذخیره می‌کند.	
کوئیل Reset، در صورتی که نتیجه عملیات شبکه متصل به آن یک شود و محتوای این کوئیل یک باشد، محتوای آن صفر شده و با تغییر نتیجه شبکه از یک به صفر این کوئیل در وضعیت صفر باقی می‌ماند	
کوئیل Set، در صورتی که نتیجه عملیات شبکه متصل به آن یک شود و محتوای این کوئیل صفر باشد، محتوای این کوئیل یک شده و با تغییر نتیجه شبکه از یک به صفر این کوئیل در	

وضعیت یک باقی می ماند																
<b>RS فلیپ فلاپ</b> ، این المان دو ورودی بیتی به نام های Reset و Set و یک خروجی بیتی دارد و بر طبق جدول زیر عمل می کند.	 RS															
<table><tr><td>Reset</td><td>Set</td><td>Output</td></tr><tr><td>0</td><td>0</td><td>بدون تغییر</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	Reset	Set	Output	0	0	بدون تغییر	1	0	0	0	1	1	1	1	1	
Reset	Set	Output														
0	0	بدون تغییر														
1	0	0														
0	1	1														
1	1	1														
<b>SR فلیپ فلاپ</b> ، این المان دو ورودی بیتی به نام های Set و Reset و یک خروجی بیتی دارد و بر طبق جدول زیر عمل می کند.	 SR															
<table><tr><td>Reset</td><td>Set</td><td>Output</td></tr><tr><td>0</td><td>0</td><td>بدون تغییر</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	Reset	Set	Output	0	0	بدون تغییر	1	0	0	0	1	1	1	1	0	
Reset	Set	Output														
0	0	بدون تغییر														
1	0	0														
0	1	1														
1	1	0														
<b>آشکار کننده لبه پائین رونده</b> ، در صورتیکه نتیجه عملیات شبکه متصل به ورودی آن از یک به صفر تغییر کند، خروجی آن به مدت یک Scan time، یک می شود. این المان به ازاء هر محل که در شبکه به کار گرفته شود، به یک بیت حافظه نیاز دارد.	 --(N)--															
<b>آشکار کننده لبه بالا رونده</b> ، در صورتیکه نتیجه عملیات شبکه متصل به وردی آن از صفر به یک تغییر کند، خروجی آن به مدت یک Scan time، یک می شود. این المان به ازاء هر محل که در شبکه به کار گرفته شود، به یک بیت حافظه نیاز دارد.	 --(P)--															
<b>کوئیل Save</b> ، نتیجه عملیات لاجیکی شبکه متصل به آن در حافظه بیتی BR ذخیر شده و می توان این نتیجه با شبکه بعدی AND خواهد شد.	 --(SAVE)															
<b>آشکار کننده لبه پائین رونده متغیر از یک آدرس</b> ، در صورتیکه ورودی این المان یک باشد و متغیر آدرس داده شد به این المان از وضعیت یک به صفر تغییر کند خروجی آن به مدت یک Scan Time یک می شود. این المان به ازاء هر جا که در برنامه به کار گرفته شود، به یک بیت حافظه نیاز دارد.	 NEG															
<b>آشکار کننده لبه بالا رونده یک متغیر از یک آدرس</b> ، در صورتیکه ورودی این المان یک باشد و متغیر آدرس داده شد به این المان از وضعیت صفر به یک تغییر کند خروجی آن به مدت یک Scan Time یک می شود. این المان به ازاء هر جا که در برنامه به کار گرفته شود، به یک بیت حافظه نیاز دارد.	 POS															

### مثال ۱:

با استفاده از المانهای NO، NC، Not، کوئیل خروجی و خروجی میانی برنامه‌ای برای توابع زیر طرح کنید.

$$F1 = (A \oplus B)C$$

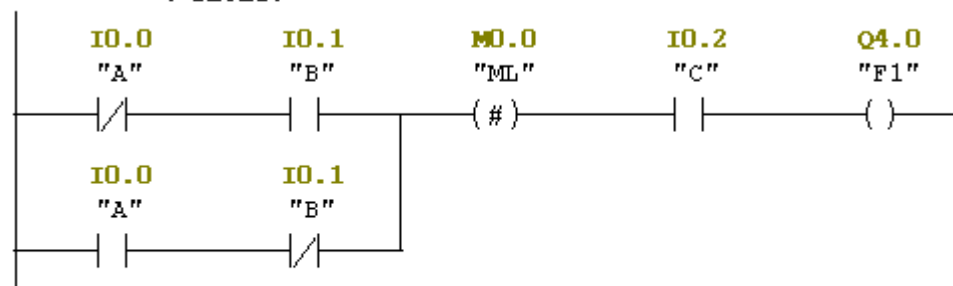
$$F2 = \overline{(A \oplus B)}D \bullet \overline{E}$$

F1: Q4.0      F2: Q4.1      A: IO.0      B: IO.1      C: IO.2      D: IO.3      E: IO.4

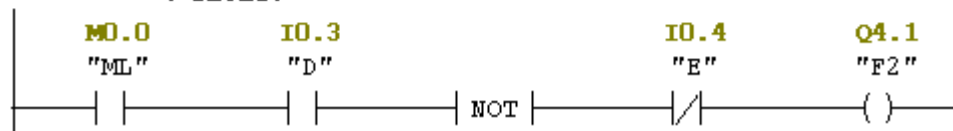
### برنامه:

OB1 : "Main Program Sweep (Cycle)"

Network 1 : Title:



Network 2 : Title:



### مثال ۲:

برای آشنا شدن با نحوه عملکرد کوئیل‌های (S) و (R) برنامه زیر را آماده و عملکرد آنرا بررسی کنید. توجه شود، حتما لازم است کوئیل‌های (S) و (R) در دو شبکه جداگانه به کار گرفته شوند. برای بررسی این دو کوئیل، بر طبق جدول زیر از بالا به پایین ورودی‌ها را تغییر داده نتایج خروجی را در جدول ۱ وارد کنید. طرح دو شبکه زیر را جابجا کرده و بررسی را مجدداً انجام داده و نتایج را در جدول ۲ وارد کنید. نتایجی را که با این بررسی در ارتباط با عملکرد این کوئیل‌ها بدست می‌آورد توضیح دهید.

<b>Network 1 : Title:</b> 	IO.1	IO.0	Q4.0		IO.1	IO.0	Q4.0
	1	0	0		1	0	0
	0	0			0	0	
	0	1			0	1	
<b>Network 2 : Title:</b> 	1	0			1	0	
	0	0			0	0	
	0	1			0	1	
	1	1			1	1	
جدول ۱					جدول ۲		

**مثال ۳:** برای بررسی المانهای RS و SR فلیپ فلاپ دو شبکه زیر را آماده کرده سپس بر طبق جدول‌های زیر از بالا به پایین ورودی‌ها را تغییر داده نتایج خروجی را در جدول‌های مربوطه وارد کنید.

<b>Network 3 : Title:</b> <pre>graph LR     I0_1[I0.1] --- OR1(( ))     I0_0[I0.0] --- OR1     OR1 --- R[R]     S[S] --- MO0[MO.0]     MO0 --- Q[Q]     Q --- Q4_1[Q4.1]</pre>	<b>IO.</b> <b>1</b>	<b>IO.</b> <b>0</b>	<b>Q4.</b> <b>1</b>	<b>IO.</b> <b>1</b>	<b>IO.</b> <b>0</b>	<b>Q4.</b> <b>2</b>
	0	0			0	0
	0	1			0	1
	0	0			0	0
	1	0			1	0
	0	0			0	0
	1	1			1	1
<b>جدول ۱</b>			<b>جدول ۲</b>			

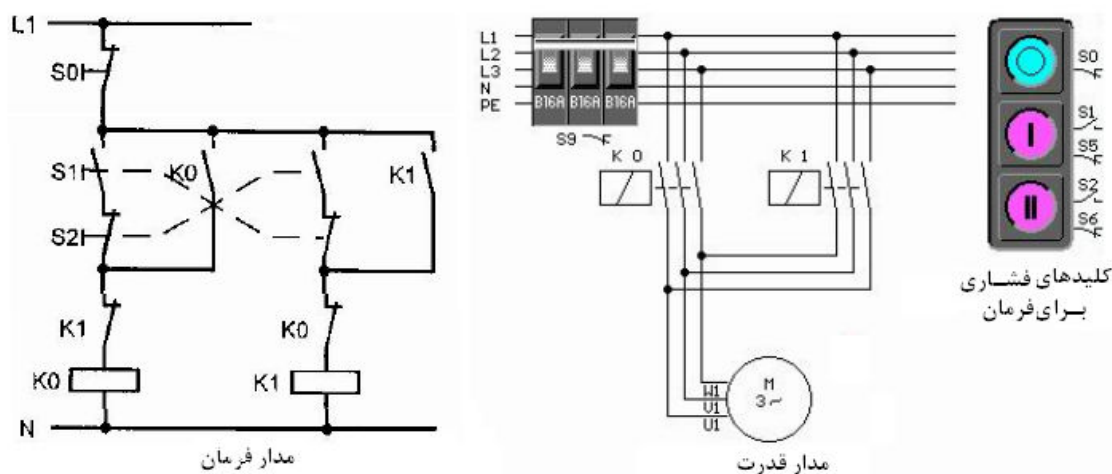
**سؤال:** چرا نتایج عملکرد این دو فلیپ فلاپ در ردیف آخر با هم متفاوت هستند.

**تمرین ۱:** در شکل زیر مدارهای الکتریکی قدرت و فرمان یک تابلوی الکتریکی مربوط به یک موتور سه فاز که بصورت چپ گرد و راست گرد کار می کند نشان داده شده است. برنامه هائی برای PLC به منظور جایگزین کردن مدار فرمان این تابلو در سه طراح مختلف آماده کرده آنرا تست کنید. آدرس ها بصورت سیمبول استفاده شود

**الف:** فقط با استفاده از کنتاکت های NO، NC و کوئیل خروجی

**ب:** فقط با استفاده از کنتاکت های NO، NC و کوئیل Set و کوئیل Reset

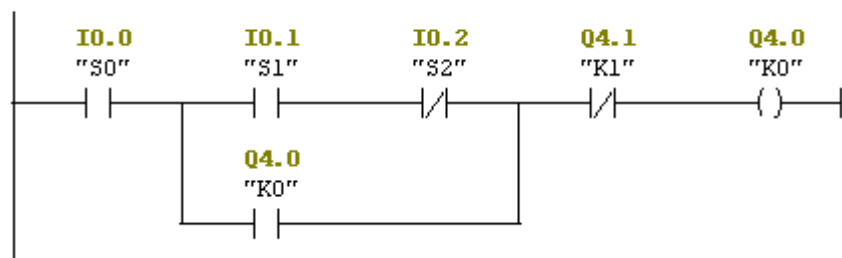
**ج:** فقط با استفاده از کنتاکت های NO، NC و فلیپ فلاپ SR



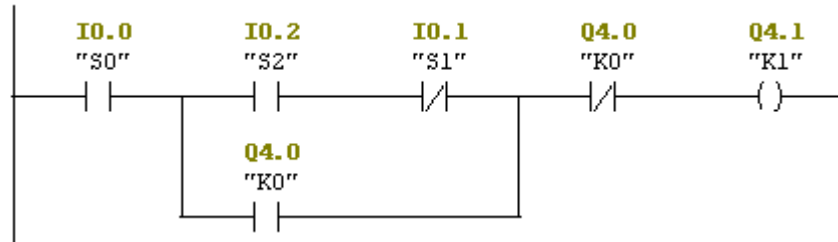
**برنامه الف:**

OB1 : "Main Program Sweep (Cycle)"

**Network 1: Title:**



**Network 2 : Title:**

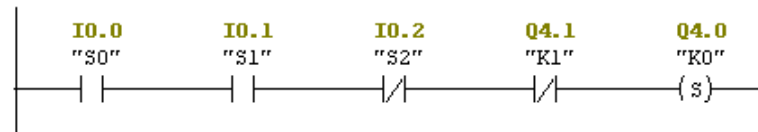


**سؤال:** چرا بجای کنتاکت بسته S0 در برنامه از کنتاکت NO و بجای کنتاکت‌های بسته K0 و K1 در برنامه از کنتاکت‌های NC استفاده شده است.

**برنامه ب:**

OB1 : "Main Program Sweep (Cycle)"

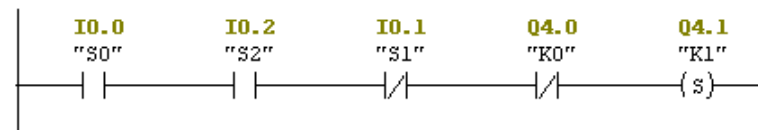
**Network 1 : Title:**



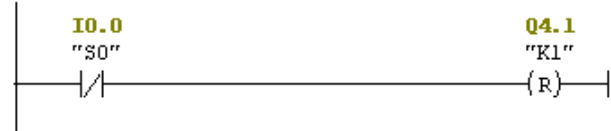
**Network 2 : Title:**



**Network 3 : Title:**



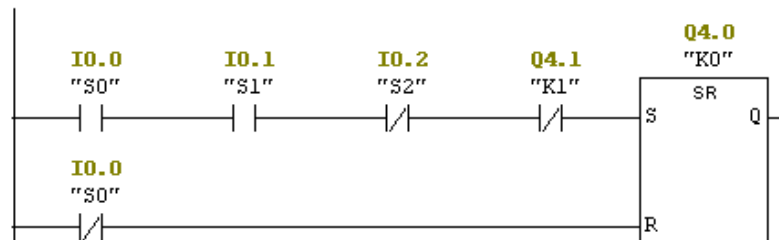
**Network 4 : Title:**



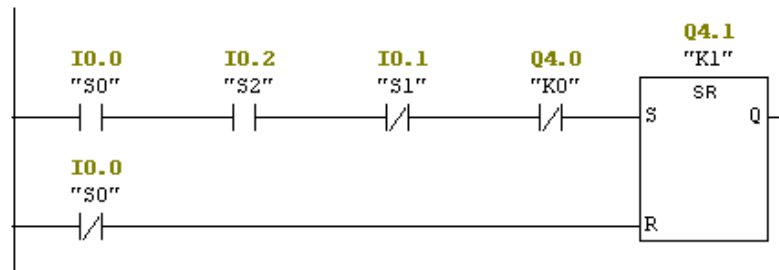
**برنامه ج:**

OB1 : "Main Program Sweep (Cycle)"

**Network 1 : Title:**



**Network 2 : Title:**





**مثال ۴:** برای بررسی آشکار کننده لبه بالا رونده و آشکار کننده لبه پائین رونده دو شبکه زیر را آماده کرده سپس بر طبق جدول-های زیر از بالا به پائین ورودی‌ها را تغییر داده نتایج خروجی را در جدول‌های مربوطه وارد کنید.

Network 5): Title:			<div><pre>graph LR     I0_1[I0.1] -- NO --- J1(( ))     MO_2[MO.2] -- P --- J1     J1 -- NO --- S((S))     I0_0[I0.0] -- NO --- R((R))     S --- MO_3[MO.3 SR]     MO_3 -- Q --- Q4_3[Q4.3]     R --- MO_3</pre></div>			<table><tr><th>I0.1</th><th>I0.0</th><th>Q4.3</th></tr><tr><td>0</td><td>0</td><td></td></tr><tr><td>0</td><td>1</td><td></td></tr><tr><td>0</td><td>0</td><td></td></tr><tr><td>1</td><td>0</td><td></td></tr><tr><td>0</td><td>0</td><td></td></tr><tr><td>1</td><td>1</td><td></td></tr></table>			I0.1	I0.0	Q4.3	0	0		0	1		0	0		1	0		0	0		1	1	
I0.1	I0.0	Q4.3																											
0	0																												
0	1																												
0	0																												
1	0																												
0	0																												
1	1																												
Network 6 : Title:			<div><pre>graph LR     I0_1[I0.1] -- NO --- J1(( ))     MO_4[MO.4] -- N --- J1     J1 -- NO --- S((S))     I0_0[I0.0] -- NO --- R((R))     S --- MO_5[MO.5 SR]     MO_5 -- Q --- Q4_4[Q4.4]     R --- MO_5</pre></div>			<table><tr><th>I0.1</th><th>I0.0</th><th>Q4.4</th></tr><tr><td>0</td><td>0</td><td></td></tr><tr><td>0</td><td>0</td><td></td></tr><tr><td>1</td><td>0</td><td></td></tr><tr><td>0</td><td>0</td><td></td></tr><tr><td>1</td><td>1</td><td></td></tr></table>			I0.1	I0.0	Q4.4	0	0		0	0		1	0		0	0		1	1				
I0.1	I0.0	Q4.4																											
0	0																												
0	0																												
1	0																												
0	0																												
1	1																												
			جدول ۱			جدول ۲																							

**سؤال:** آشکار ساز لبه بالا روند و آشکار ساز لبه پائین رونده در دو شبکه بالا چه تاثیری دارند.

**مثال ۵:** با تغییر مقادیر ورودی‌های آمده در جدول زیر، عملکرد دو شبکه مجاور جدول را بررسی کرده با این بررسی کار کوئیل Save را نتیجه‌گیری کنید

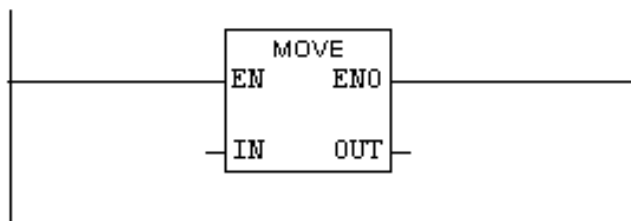
<b>Network 7 : Title:</b>		<b>I0.0</b>	<b>I0.1</b>	<b>I0.2</b>	<b>Q4.5</b>
		0	0	0	
		1	0	0	
		0	1	0	
		1	1	0	
		0	0	1	
<b>Network 8 : Title:</b>		0			جدول ۳
					
<b>I0.2</b>	<b>Q4.5</b>	1	1	1	

**مثال ۶:** عملکرد شبکه زیر را با تغییر مقادیر ورودی‌ها بر طبق جدول آمده در کنار آن بررسی کرده با این بررسی کار بلوک Pos را نتیجه‌گیری کنید.

<div>Network 1: Title:</div> <pre>graph LR     I0_0[I0.0] --- AND1[ ]     M0_0[M0.0] --- AND1     AND1 --- S[S]     I0_2[I0.2] --- R[R]     S --- Q1[Q]     Q1 --- Q4_0[Q4.0]</pre>	I0.0	I0.1	I0.2	Q4.0
	0	0	0	
	1	0	0	
	1	1	0	
	0	0	0	
	0	0	1	
	0	0	0	
	0	1	0	
	1	1	0	
	0	0	0	
	0	0	1	

### ۲-۳) بررسی انتقال داده (Move):

با بلوک Move می‌توان داده‌ها با طول Byte، Word، و Double Word را از محلی به محل دیگر (بدون تغییر فرمت آن) انتقال داد. سمبول این بلوک در زیر ارائه شده است. هرگاه Enable ورودی فعال باشد این بلوک اجرا و Enable خروجی فعال می‌شود



اگر طول آدرس ورودی Byte و طول آدرس خروجی Double Word باشد، محتوای آدرس ورودی به کم ارزش‌ترین بایت مربوط به Double Word خروجی منتقل می‌شود.

اگر طول آدرس ورودی Byte و طول آدرس خروجی Word باشد، محتوای آدرس ورودی به کم ارزش‌ترین بایت مربوط به Word خروجی منتقل می‌شود.

اگر طول آدرس ورودی Word و طول آدرس خروجی Double Word باشد، محتوای آدرس ورودی به Word کم ارزش - تر مربوط به Double Word خروجی منتقل می‌شود.

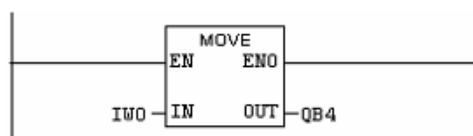
اگر طول آدرس ورودی Double Word و طول آدرس خروجی Byte باشد، محتوای کم ارزش‌ترین بایت Double Word ورودی به Byte خروجی منتقل می‌شود.

اگر طول آدرس ورودی Double Word و طول آدرس خروجی Word باشد، محتوای Word کم ارزش‌تر ورودی به Word خروجی منتقل می‌شود.

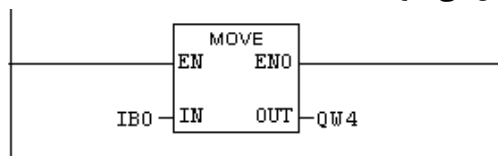
اگر طول آدرس ورودی Word و طول آدرس خروجی Byte باشد، محتوای بایت کم ارزش‌تر Word ورودی به Byte خروجی منتقل می‌شود.

اگر طول آدرس‌های ورودی و خروجی مثل هم باشند داده‌های ورودی به آدرس‌ها با ارزش نظیر به نظیر به خروجی منتقل می‌شوند.

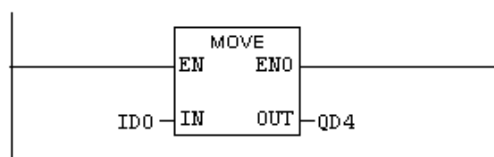
### چند مثال برای المان MOVE:



در این مثال محتوای IB1 به QB4 منتقل می‌شود.



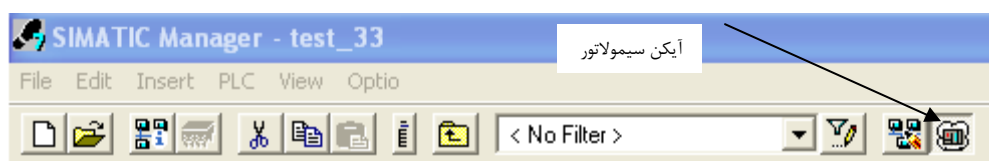
در این مثال محتوای IB0 به QB5 منتقل می‌شود.



در این مثال محتوای IB3 به QB7، IB2 به QB6، IB1 به QB5 و IB0 به QB4 منتقل می‌شوند.

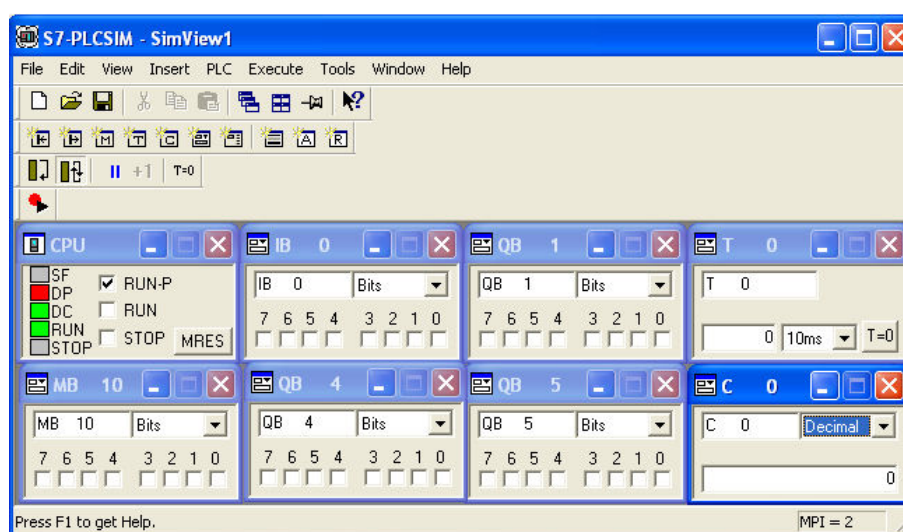
### ۳-۳) معرفی سیمولاتور PLC:

با آیکن نشان داده شده در شکل (۱-۳) که در میله ابزار صفحه Simatic Manager موجود است، نرم افزار سیمولاتور PLC نشان داده شده در شکل (۲-۳) باز می شود.



شکل (۱-۳)

**توجه:** توجه شود هرگاه سیمولاتور باز شده باشد و PG هم به PLC وصل باشد در این شرایط اگر برنامه ای Download شود آن برنامه به سیمولاتور منتقل می شود نه به PLC، برای Download کردن برنامه به PLC باید حتما سیمولاتور بسته شود.



شکل (۲-۳)

از ابزارهای این سیمولاتور می توان برای اجرای برنامه، مشاهده نتایج آن در آدرسهای ورودی؛ خروجی، حافظه، زمان سنج، شمارنده و ..... مشاهده کرد. در PLC سری 300 شرکت زیمنس فقط دو آکمولاتور وجود دارد.