بسم الله الرحمن الرحیم

جزوه آز کنترل صنعتی

فصل ۳

مدرس دکتر فلاح

فصل ۳

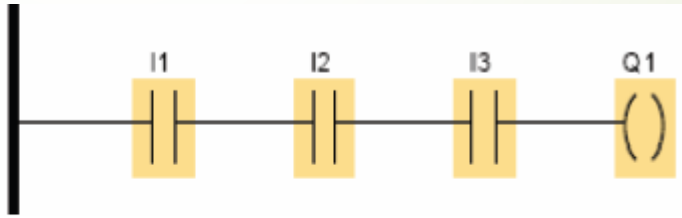
در این فصل دستورات لوگو را به زبان ساده همراه با مثالهای کاربردی برای هر دستور بیان می کنیم.

زبان های برنامه نویسی در لوگو

دو نوع زبان برای برنامه نویسی لوگو وجود دارد:

LAD(Ladder Diagram) :

زبان نردبانی که با توجه به شبیه بودن به مدل مدار فرمان بیشتر برای طراحی مدارات فرمان مورد استفاده قرار می گیرد . نمونه ای از این برنامه نویسی را در شکل زیر مشاهده می کنید.

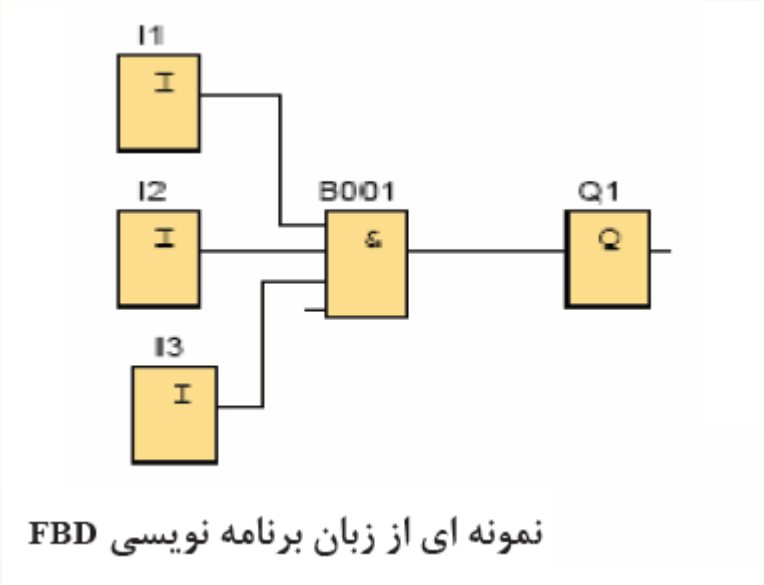


نمونه ای از زبان برنامه نویسی LAD

فصل ۳

FBD(Function Block Diagram) :

زبانی بلوکی که برای درک بهتر مسئله مناسب است . نمونه ای از این نوع برنامه نویسی در شکل زیر آمده است.

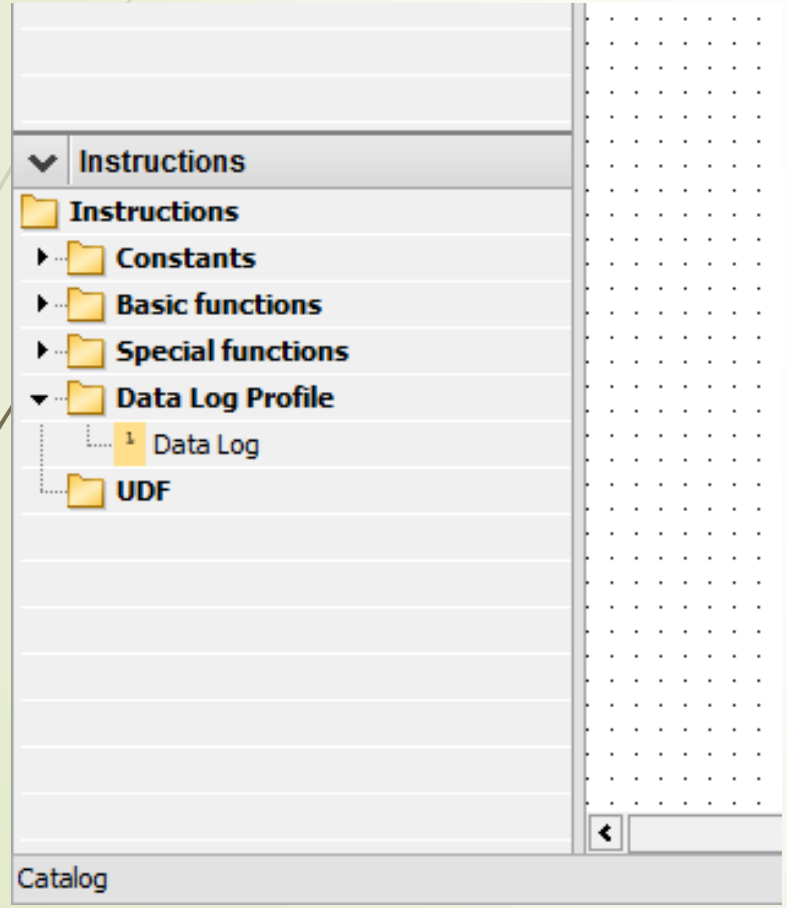


نمونه ای از زبان برنامه نویسی FBD

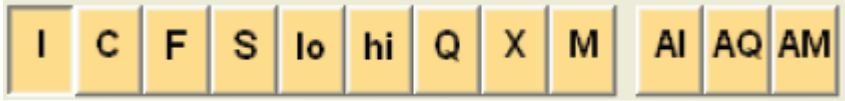
فصل ۳

عناصر برنامه نویسی در لوگو به سه قسمت کلی تقسیم می شوند:

1) ثبات ها –(Constants)
2) توابع بیسیک (پایه) –(Basic Function)
3) توابع ویژه –(Special Function)



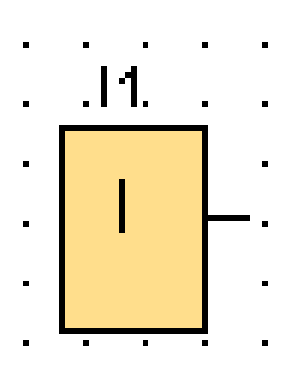
ثبات ها :



شامل ورودی خروجی و فلگ های (پرچم) دیجیتال و آنالوگ

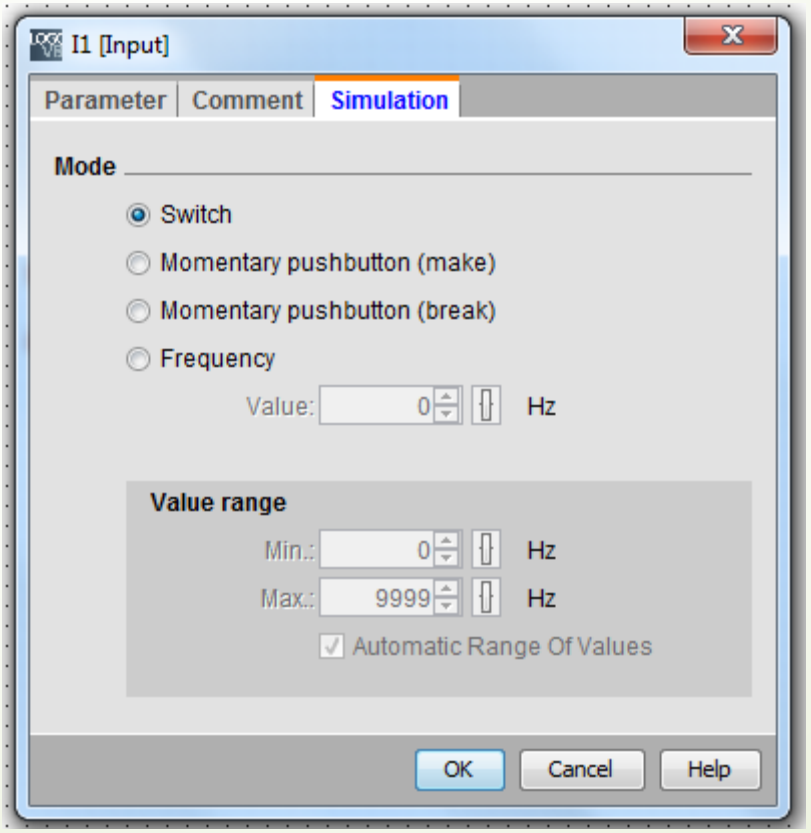
الف) دیجیتال

ورودی – Input



فصل ۳

ورودی های دیجیتال می توانند دارای مقدار صفر و یک باشند . لوگو دارای ۲۴ ورودی دیجیتال می باشد . با دو بار کلیک کردن بر روی هر ورودی دیجیتال در محیط برنامه نویسی پنجره مشخصات آن به شکل زیر باز خواهد شد:

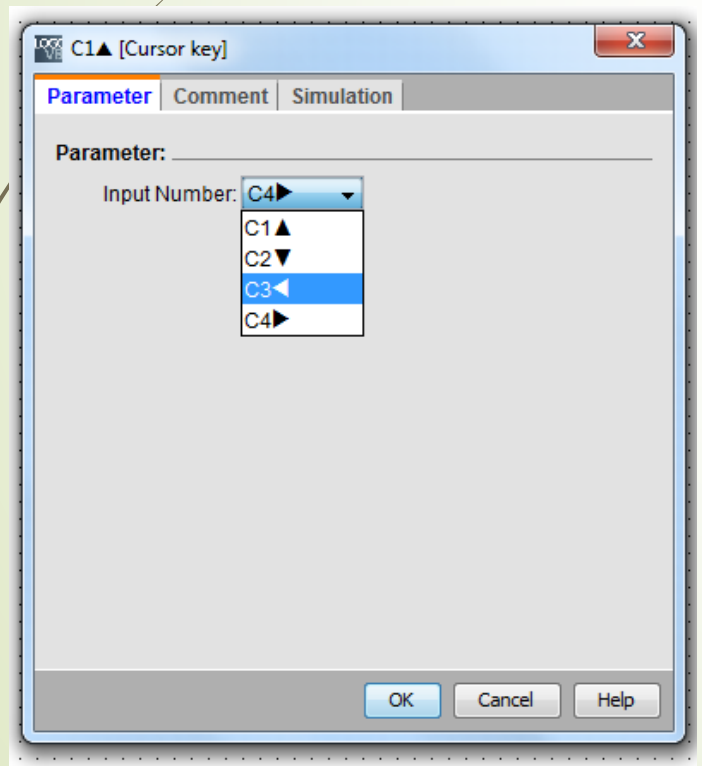


فصل ۳

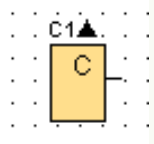
در سربرگ Paramter : از این پنجره می توان شماره مربوط به ورودی را که حداکثر ۲۴ است ، انتخاب نمود.

در سربرگ Comment از این پنجره می توان توضیحاتی در مورد این ورودی ارائه دارد.

در سربرگ Simulation از این پنجره می توان نوع این ورودی برای شبیه سازی را تعیین کرد. یک ورودی دیجیتال می تواند از نوع کلید ، شستی استارت ، شستی استوپ و یا فرکانسی باشد.

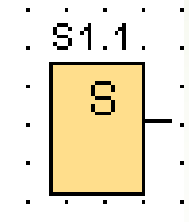


کلید مکان نما – Curser Key



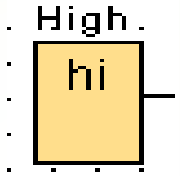
از کلیدهای مکان نما که بروی بعضی از لوگوها وجود دارند می توان به عنوان ورودی دیجیتال استفاده نمود . از این کلیدها بروی سخت افزار لوگو موقع برنامه نویسی استفاده می شود. در نرم افزار نیز می توان آن ها را با توجه به جهت نشان داده شده ، از پنجره مشخصات آن انتخاب کرد.

فصل ۳



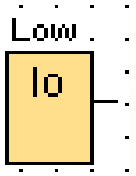
بیت های شیفت رجیستر – Shift register bits

بیت های شیفت رجیستر که ۸ عدد می باشند ، به همراه تابع شیفت رجیستر استفاده می شوند. بهتر است برای توضیح بیشتر به توضیح تابع شیفت رجیستر مراجعه کنید.



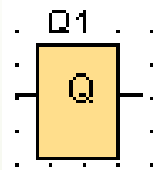
یک – High

به عنوان یک ورودی ثابت با مقدار یک در داخل برنامه می باشد و نیاز به اتصال کلیدی از خارج به PLC برای این ورودی نمی باشد و تعداد آن محدود است.



صفر – Low

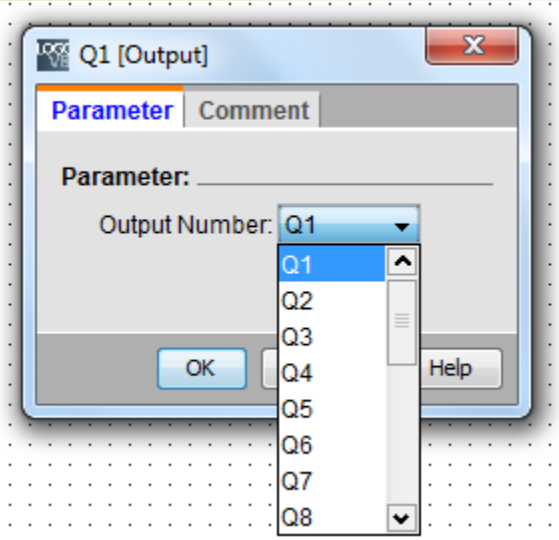
به عنوان یک ورودی ثابت با مقدار صفر در داخل برنامه می باشد و نیازی به اتصال کلیدی از خارج به PlC برای این ورودی نمی باشد و تعداد آن محدود است.



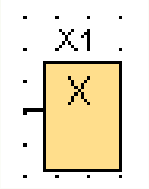
خروجی یا Output

در لوگو، ۱۶ خروجی دیجیتال وجود دارد که روشن و خاموش بودن آنها مبین عملکرد وصل یا قطع بودن رله ها در خروجی لوگو می باشد.

فصل ۳



در تصویر روبرو پنجره تنضیمات خروجی دیجیتال را مشاهده می کنید.



اتصال دهنده های باز (Open Connectors)

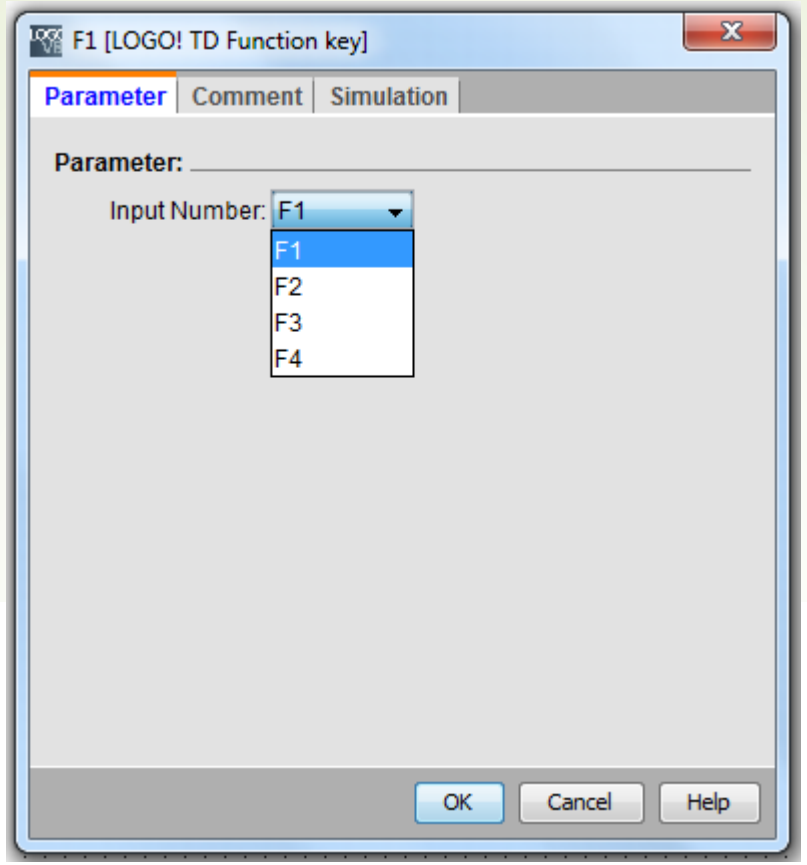
جهت نشان دادن متن پیام در بلوک Massage text به خروجی بلوک Massage text وصل می شود . تعداد آن با تعداد خروجی دیجیتال برابر است.

کلید تابع -LOGO! TD function key

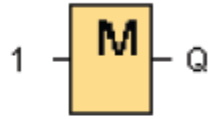
LOGO! TD چهار کلید دارد که شما می توانید از آنها همانند ورودیهای دیگر در برنامه استفاده کنید. این کلیدها با نـام هـای F1-F4 مشخص می باشند. بدین طریق می توانید با فشار دادن هریک از کلیدهای LOGO! TD فرمانهائی را همانند سایر ورودی‌ها صادر کنید.

فصل ۳

پنجره تنظیمات کلید Logo! TD



فلگ ها–Flags



فلگ یا پرچم جهت نشان دادن وضعیت نقاط مختلف مدار در داخل لوگو به کار می رود. عملکرد آن مثل خروجی است ولی در سخت افزار به رله های خروجی وصل نیست و می توان آنرا به عنوان رله داخلی مدار فرمان دانست. همچنین وقتی مشاهده کردید که در اتصال از خروجی تابعی به ورودی تابعی دیگر با مشکل مواجهید در مابین این دو تابع از فلگ استفاده کنید. همچنین در طراحی مدارات فرمان می توان از آن به عنوان کنتاکتور کمکی استفاده کرد.

فصل ۳

تعداد فلگ ها به تفکیک ورژن های لوگو به صورت زیر می باشد:

0BA6: 27 digital flags M1... M27
0BA4, 0BA5: 24 digital flags M1... M24
0BA3, 0BA2: 8 digital flags M1... M8
0BA1: 4 digital flags M1... M4
0BA0: 0 flags

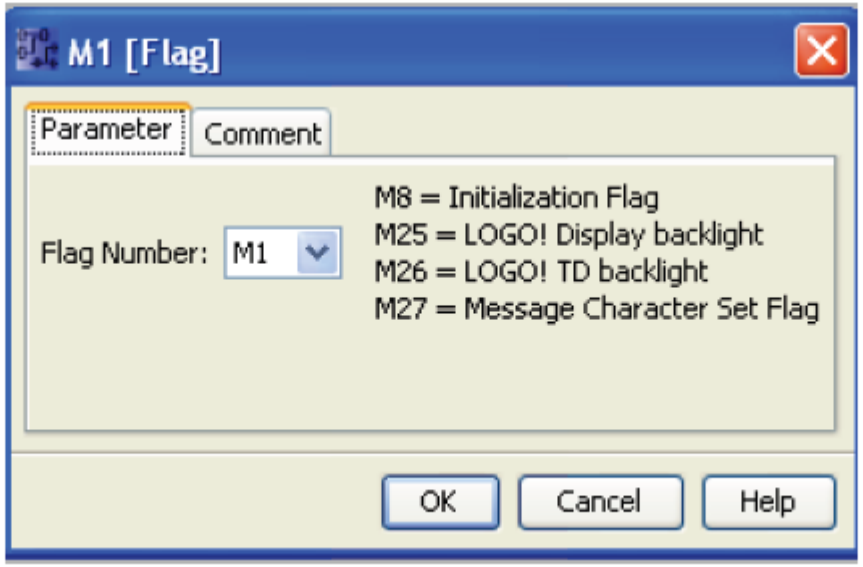
در ورژن ۶ لوگو ۲۷ فلگ وجود دارد که در این میان، می توان ازفلگ M8 به عنوان راه انداز اولیه استفاده کرد. چرا که این فلگ بدون نیاز به ورودی در ابتدای شروع به کار لوگو یک پالس می فرستد که با استفاده مناسب از آن، می توان موقعی که برق قطع شده و بایستی بعد از وصل برق سیستم دوباره به راه بیافتد، برنامه را راه اندازی کرد.

فلگ M25 برای کنترل نور پس زمینه نمایشگر لوگو به کار می رود. فلگ M26 نیز برای کنترل نور پس زمینه نمایشگر LOGO!TD مورد استفاده قرار می گیرد. در واقع وقتی که این فلگها در برنامه باشند و برنامه به داخل لوگو ریخته شود نمایشگر

فصل ۳

لوگو یا LOGO! TD تا زمانیکه این فلگ ها فعال باشند روشن خواهد بود. این فلگ ها می توانند دربرنامه نویسی در خروجی تابع Message text قرار گیرند.

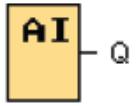
فلگ M27 از بین دو مجموعه کاراکتر یکی را برای نمایش در صفحه نمایشگر انتخاب می کند. اگر ورودی M27 یک باشد، مجموعه کاراکتر۱ و اگر صفر باشد، مجموعه کاراکتر۲ برای نمایشگر نمایش داده خواهد شد.



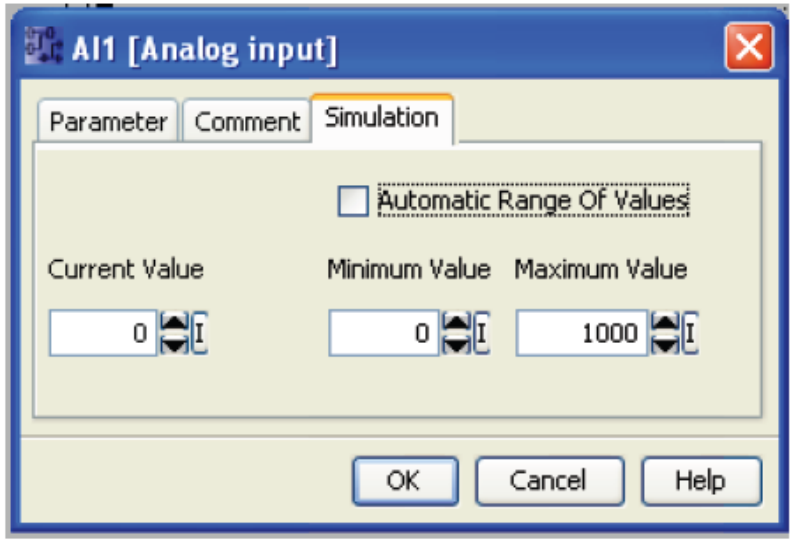
البته باید خاطر نشان کرد که در لوگو-۸ تعداد این فلگ ها به ۳۱ رسیده است و برخی کاربردهای جانبی به آن اضافه شده است

فصل ۳

آنالوگ – Analog
ورودی های آنالوگ – Analog Inputs



به عنوان ورودی آنالوگ می توان از این ورودیها استفاده کرد. ۸ ورودی آنالوگ در نسخه های جدید لوگو موجود می باشند. همچنین می توان محدوده کاری آنرا تنظیم کرد به عنوان مثال(۰تا ۱۰ ولت). در لوگو ورژن ۶ ورودیهای I1،I2،I7 و I8 می توانند به عنوان ورودیهای آنالوگ به ترتیب با آدرسهای AI1،AI4،AI3 و AI2 مورد استفاده قرار گیرند. با این حال در لوگو جمعا ۸ ورودی آنالوگ داریم.



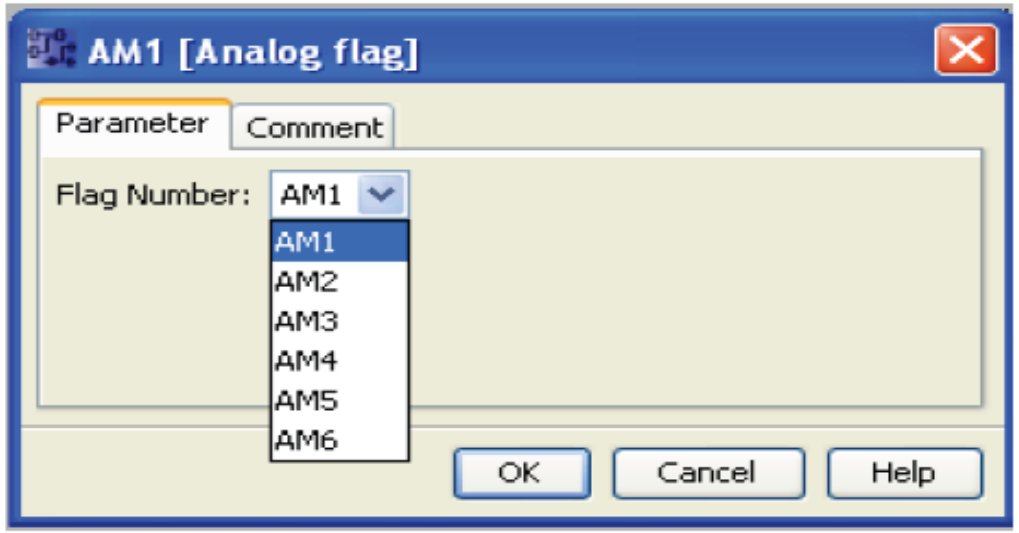
فصل ۳

خروجیهای آنالوگ– Analog outputs

لوگو دو خروجی آنالوگ دارد که در خروجی سیگنال آنالوگ تولید می کند. سیگنال های آنالوگ رنج ۰ تـا ۱۰ ولـت و ۰تـا ۲۰ میلی آمپر و ۴ تا ۲۰ میلی آمپر را دارا هستند.

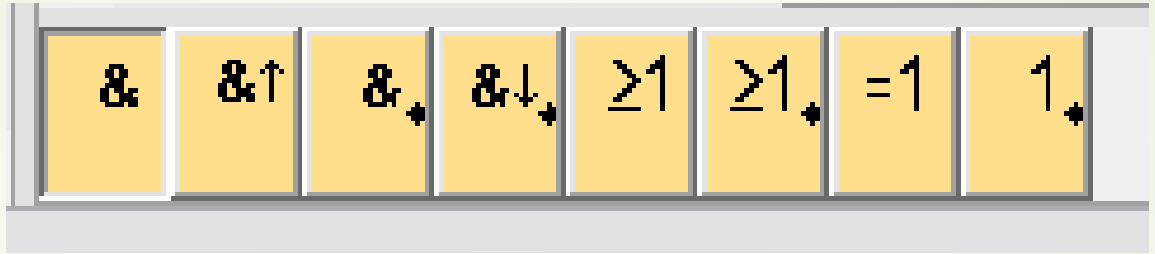
فلگ های آنالوگ–Analog flags

لوگو دارای ۶ فلگ آنالوگ می باشد، که می توانند یک مقدار آنالوگ را دریافت کرده و همان را به خروجی خود ارسال کنند.



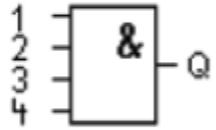
فصل ۳

توابع بیسیک (پایه)



منظور از توابع مبنا، توابع منطقی مانند AND، OR، XOR و... می باشند. که شرح آن به صورت زیر است.

AND



مشابه با اتصال سری در یک حلقه است. خروجی آن وقتی 1 است که تمام ورودیها 1 باشد.

ورودی X(ورودی بی اهمیت و یا متصل نشده) در این تابع 1 در نظر گرفته می شود.

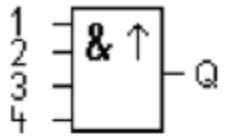
فصل ۳

جدول تریس تابع &

1	2	3	4	Q
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

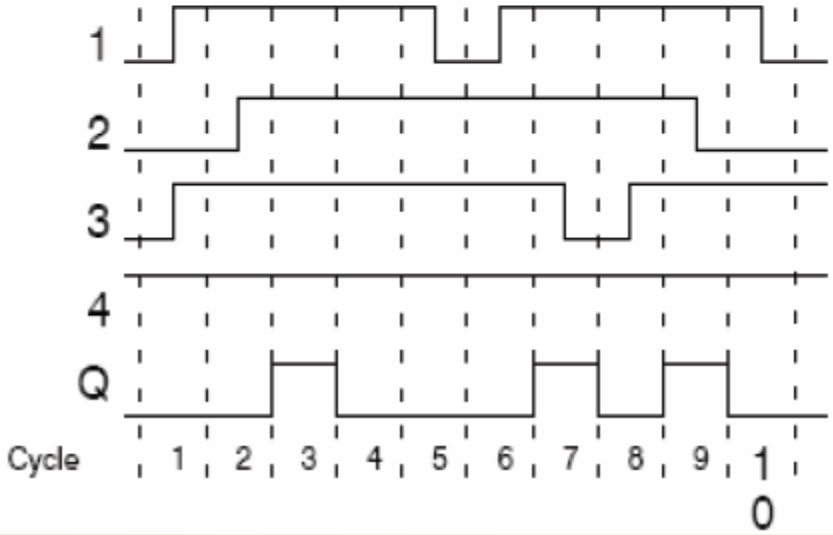
فصل ۳

AND تحریک شده با لبه بالا رونده



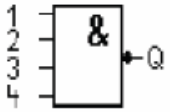
خروجی فقط هنگامی 1 است که همه ورودیها 1 بوده و حداقل یکی از ورودیها در سیکل قبل 0 بوده باشد.

ورودی X (ورودی بی اهمیت و یا متصل نشده) در این تابع 1 در نظر گرفته می شود.



فصل ۳

NAND

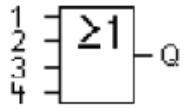


خروجی فقط وقتی 0 است که همه ورودیها 1 باشند. ورودی X (ورودی بی اهمیت و یا متصل نشده) در این تابع 1 در نظر گرفته می شود.

1	2	3	4	Q
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

فصل ۳

OR

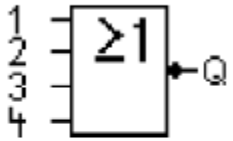


مشابه با اتصال موازی در یک حلقه است که خروجی آن وقتی 0 است که تمام ورودیها 0 باشند و خروجی آن وقتی 1 است که حداقل یکی از ورودیها 1 باشد.

1	2	3	4	Q
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

فصل ۳

NOR

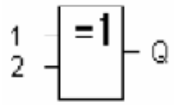


خروجی فقط هنگامی 1 است که همه ورودیها 0 باشند. ورودی X (ورودی بی اهمیت و یا متصل نشده) در این تابع 0 در نظر گرفته می شود. در تغییر 0 به 1 در یکی از ورودیها، خروجی 0 می شود.

1	2	3	4	Q
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

فصل ۳

XOR

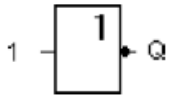


خروجی فقط وقتی 1 است که ورودیها غیریکسان باشند. ورودی X (ورودی بی اهمیت و یا متصل نشده) در این تابع 0 در نظر گرفته می شود.

1	2	Q
0	0	0
0	1	1
1	0	1
1	1	0

جدول (۶-۴): جدول منطقی گیت XOR

منفی کننده– NOT (Negation, Inverter)



این تابع معکوس کننده ورودی است، یعنی 0 را به 1 و 1 رابه 0 تغییر می دهد.

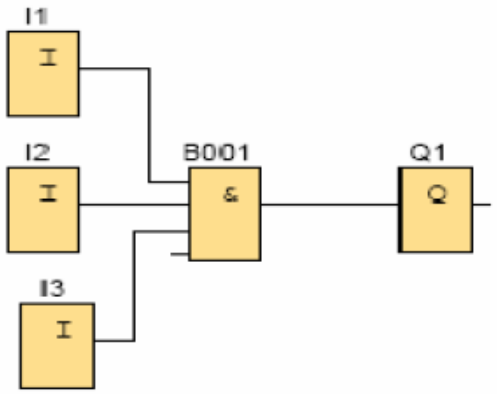
فصل ۳

I	Q
0	1
1	0

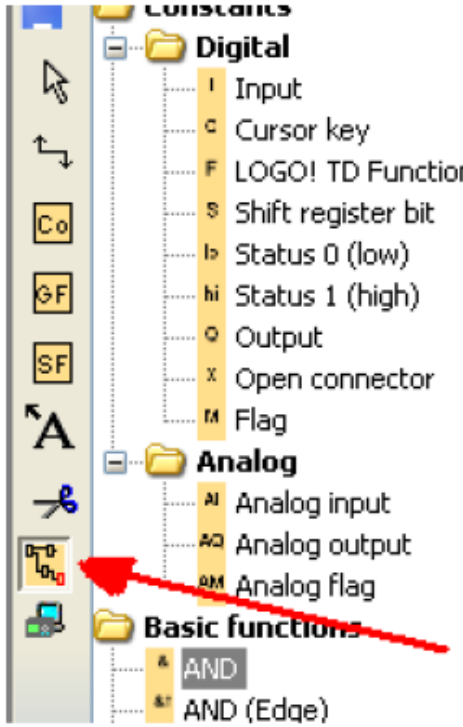
جدول منطقی گیت NOT

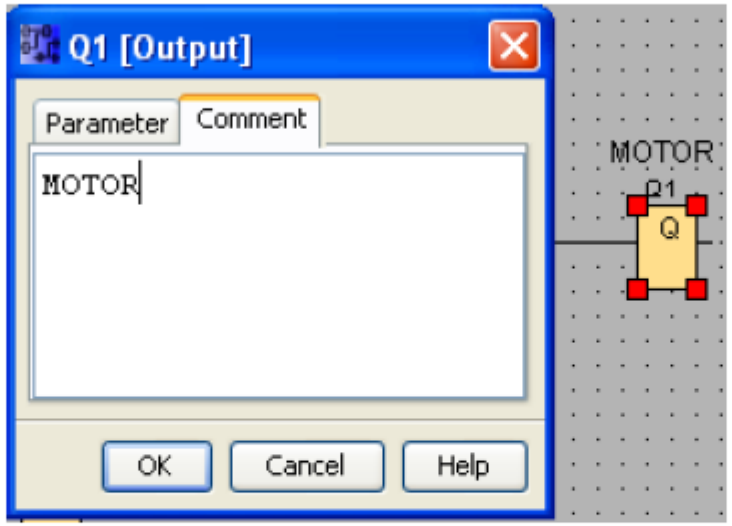
مثال(۶-۱): دستگاهی هنگامی کار می کند که هر سه کلید آن فعال باشند. برنامه مربوطه را به زبانهای LAD وFBD بنویسید.

جواب: اینکه هر سه کلید باید بطور همزمان فعال باشند، تا دستگاه کار کند، به معنی گیت AND می باشد. بنابراین در برنامه نویسی به روش FBD کافی است یک گیت AND از توابع پایه به صفحه برنامه نویسی آورده و input ها را به ورودیهای آن وصل کنیم. یک خروجی نیز به نشانه موتور دستگاه به گیت AND وصل می کنیم.

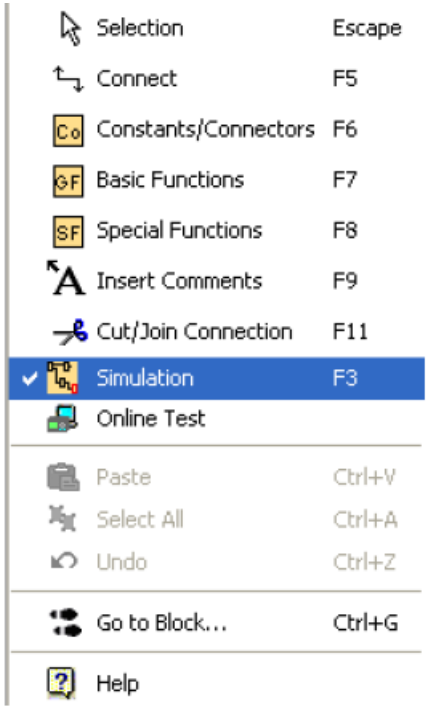


می توان با راست کلیک کردن بر روی خروجی Q1 وانتخاب Block properties و سپس انتخاب comment بـر روی خروجـی اسم نوشت.





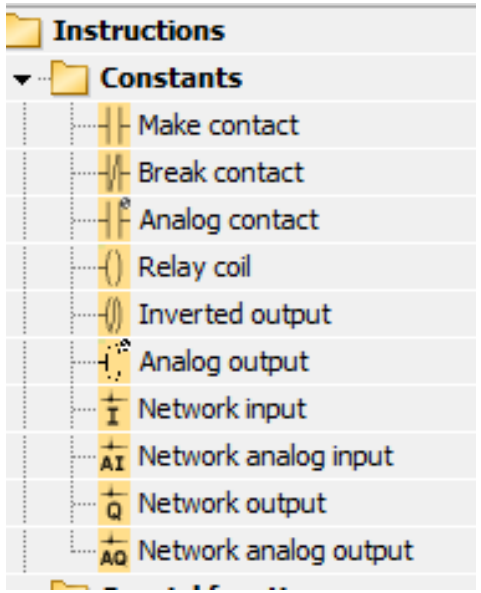
شکل (۶-۱۴): برچسب گذاری بر روی خروجی

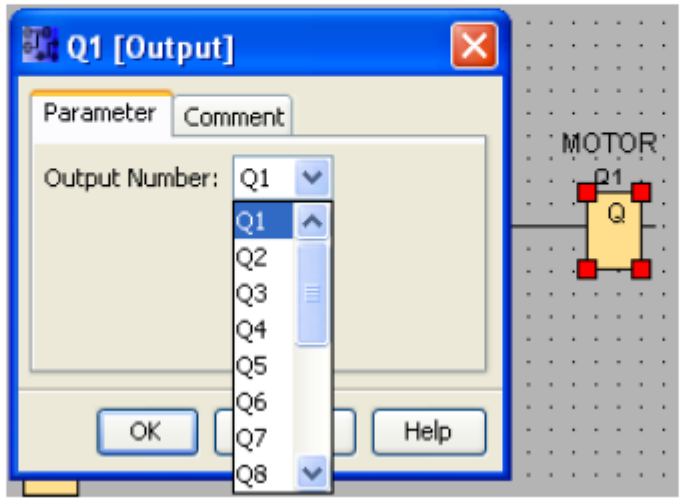


برای شبیه سازی می توانید به دو روش این کار را انجام دهید. در روش اول در صفحه برنامه نویسی کلیک راست کرده و گزینـه Simulation را انتخاب کنید. این کار را می توانید مستقیماً با زدن کلید F3 نیز انجام دهید.

فصل ۳

برای تغییر آدرس ورودیها و یا خروجیها می توانید بعد از انتخاب Block properties در کادر Parameter آدرس ها را تغییردهید. مشخص است که ورودیهای دیجیتال تا I24 و خروجیها نیز تا Q16 موجود می باشند.

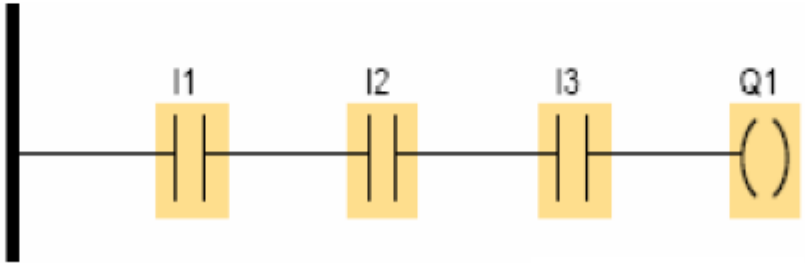




در زبان LAD دیگر گیت AND یا OR وجود ندارد. به جای AND ورودیها را سری می کنیم و به جای OR ورودیها را موازی می کنیم. در زبان LAD ورودیهای NO (در حالت عادی باز- normal open) را از Make contact و ورودیهای NC (normalcolse) را از Break contact می آوریم. خروجی نیز با نام Relay coil در سمت چپ صفحه موجود می باشد. لازم به

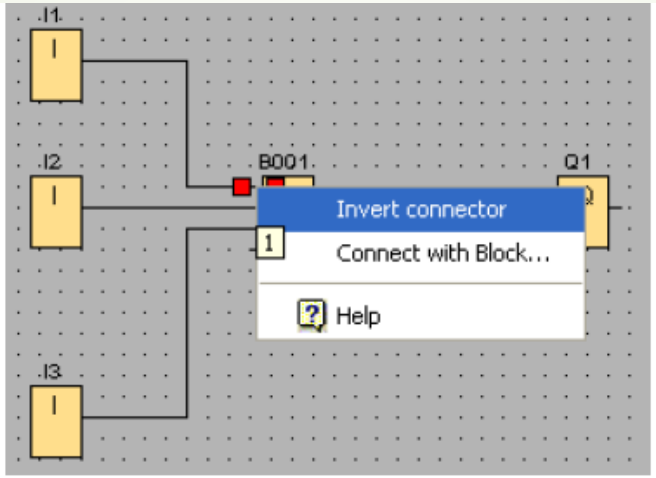
فصل ۳

یادآوری است که برنامه نوشته شده به هریک از زبانها قابل تبدیل به یکدیگر می باشند. برای اینکار در بالای صفحه آیکن convert to LAD/FBD را فشار دهید.



مدار مربوط به مثال (۶-۱) –LAD

جهت not کردن ورودی در زبان بلوکی(FBD)، بایستی در ورودی گیت ها کلیک راست کرده و گزینه Invert convertor را انتخاب کنید. در این حالت ورودی not می شود. برای برگشت به حال قبلی بایستی همان مراحل را تکرار کنید. در زبان LAD ورودیهای not شده یا به عبارتی NC را از بسته Break contact به صفحه بیاورید.

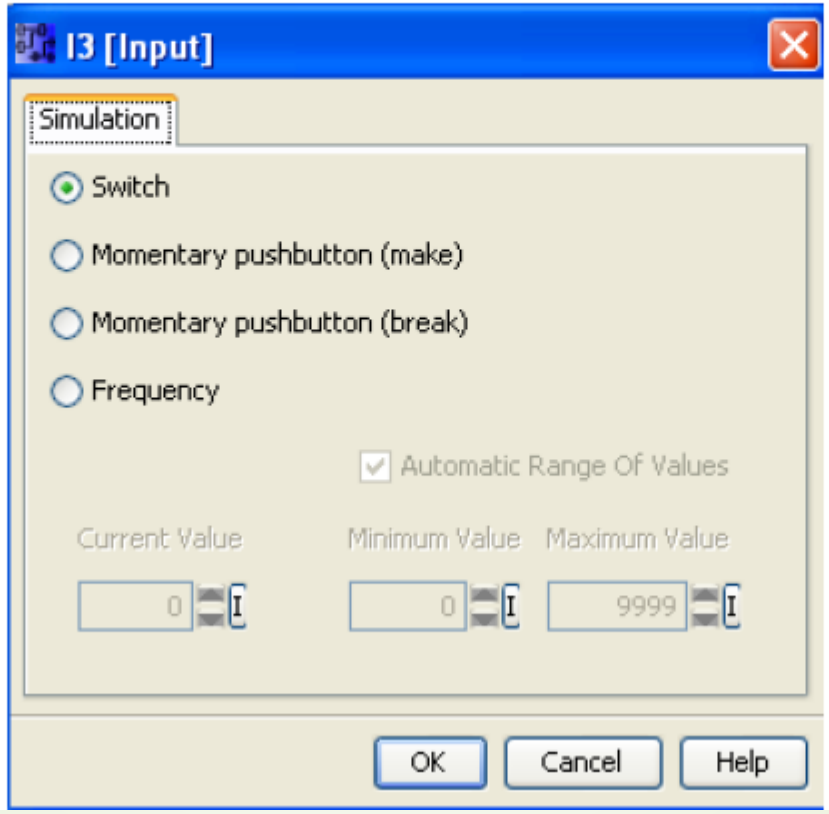


شکل (۶-۱۹): نحوه not کردن ورودی در زبان FBD

در صورتیکه بخواهید وضعیت ورودیها را از حالت سویچ به شستی تغییر دهید، می توانید به یکی از دو روش زیر اینکار را انجام دهید. در روش اول در مد شبیه سازی بر روی ورودیها در نوار سیمیلاتور کلیک راست کرده و گزینه Simulation parameters را انتخاب کنید.

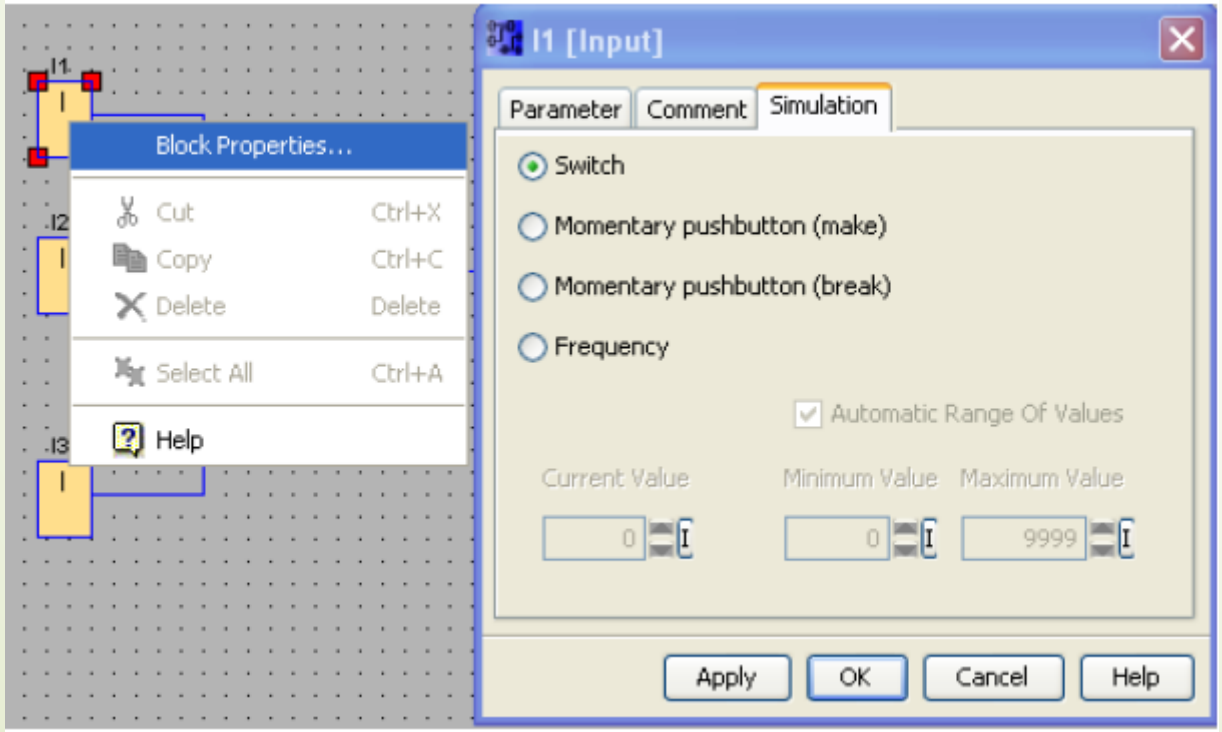
فصل ۳

در این حالت پنجره ای به شکل (۶-۲۱) باز می شود. گزینه اول حالت سویچ مـی باشـد. گزینـه دوم شسـتی NO (در حالـت عادی باز) و گزینه سوم شستی NC (در حالت عادی بسته) می باشد. گزینه چهارم ورودی فرکانسی می باشد.



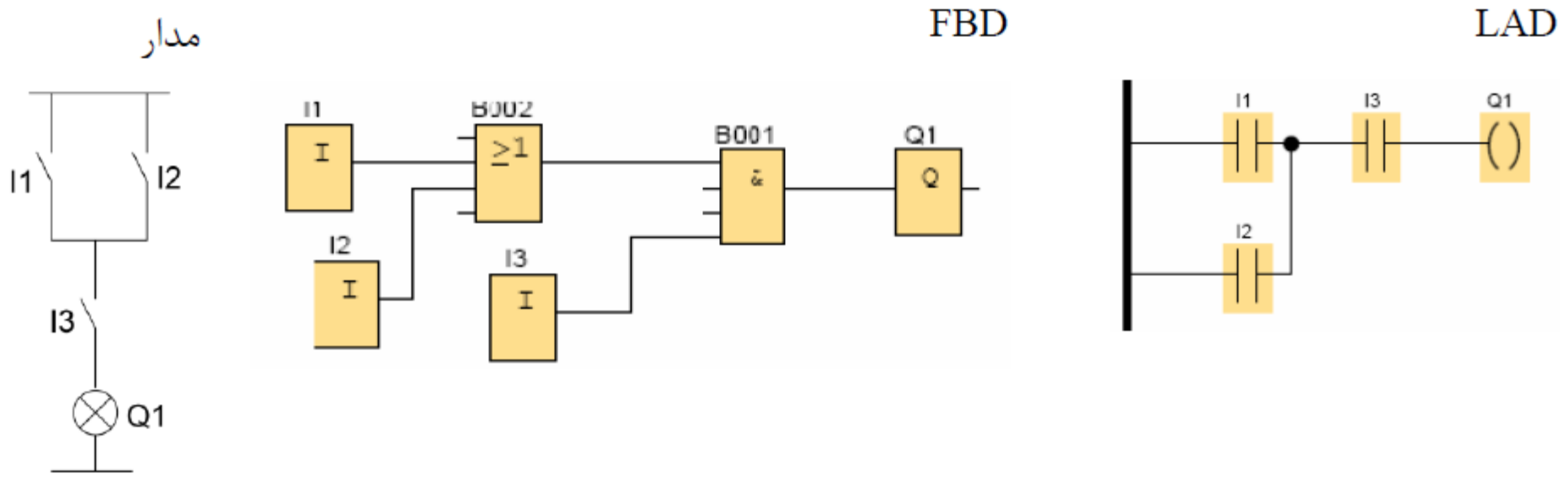
فصل ۳

همچنین می توانید بدون فعال کردن شبیه سازی در صفحه برنامه نویسی بر روی هر یک از ورودیها کلیک راست کرده و با انتخاب Block properties و سپس انتخاب کادر simulations از پنجره باز شده همان تنظیمات فوق را انجام دهید. (البته با دو بار کلیک کردن بر روی ورودی نیز می توانید وارد پنجره مشخصات ورودی شوید.)



فصل ۳

مثال(۶-۲): با توجه به مدار زیر مشخص است که ورودیهای I1 وI2 موازی هستند، لذا در زبان بلوکی از گیت OR استفاده می کنیم. نتیجه OR نیز با ورودی I3 سری می باشد، لذا از AND استفاده می کنیم.



شکل (۶-۲۳): مدار مربوط به مثال (۶-۲)