

به نام خدا

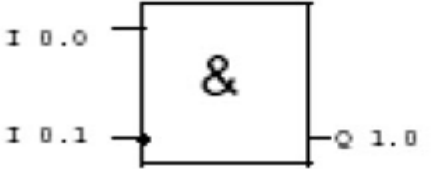
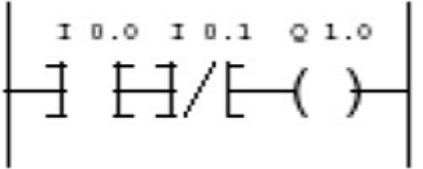
دستور کار کنترل صنعتی

(فصل ششم)

دکتر فلاح

مقدمه ای بر برنامه نویسی S5

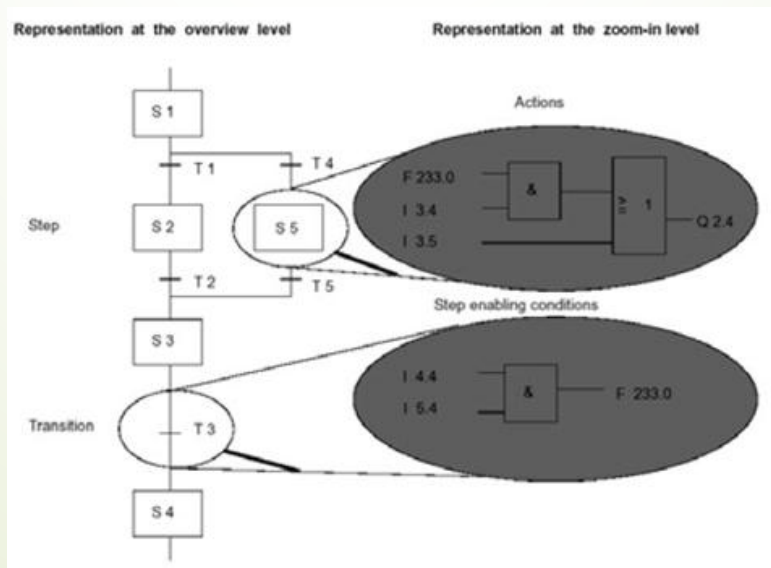
زبان S5 جهت برنامه نویسی PLC های ساخت زیمنس و توسط همین شرکت ابداع گردیده است. در این زبان برنامه کنترل می تواند علاوه بر دو روش گزاره ای و نرده بانی، به روش فلوچارتی هم تدوین گردد. در روش فلوچارتی منطق کنترل به صورت فلوچارت نمایش داده شده و هر جعبه نوع عمل منطقی را نشان می دهد. اتصال قسمت های مختلف بهم، روند حرکت فرمان ها را نمایان می سازد، بدین ترتیب برنامه ای که به روش فلوچارتی نوشته می شود. عبارت است از اتصال یکسری جعبه به یکدیگر به گونه ای که خروجی هر یک می تواند به عنوان ورودی وارد جعبه یا جعبه های دیگر شود. شکل (۱-۶) مقایسه این سه روش در قسمت کوچکی از یک برنامه را نشان می دهد.

STL	CSF	LAD
<pre> A I 0.0 AN I 0.1 = Q 1.0 </pre>		

شکل (۱-۶)

مقدمه ای بر برنامه نویسی S5

GRAPH5 یک زبان سطح بالا و گرافیکی برای برنامه نویسی **PLC** های مدل **S5** می باشد، این زبان برنامه نویسی به کاربر اجازه می دهد که برنامه کنترلی خود را در قالب فلوچارت بنویسد. این زبان شامل دو واحد عملیاتی به نام های **STEP** و **TRANSATION** می باشد. استپ ها دستورات اجرایی بوده و ترانزیشن ها شروطی می باشند که باید برقرار شوند تا اجازه اجرای دستورات استپ داده شود. هر برنامه کاربر به فرم **GRAPH5** با **S1** شروع شده و با **S0** پایان می یابد. در شکل (۲-۶) شمای کلی یک برنامه به شکل **GRAPH5** نمایش داده شده است.



شکل (۲-۶)

مقدمه ای بر برنامه نویسی S5

در نرم افزار S5 به دو صورت خطی یا ساختار یافته می توان برنامه نویسی کرد. هنگامی که پروسه کنترلی ساده باشد یک بلوک برای نوشتن برنامه کافی است، در این مورد برنامه در OB1 نوشته می شود و PLC این برنامه را به صورت متوالی اجرا می کند. بدین معنی که بعد از اجرای آخرین بخش از برنامه، PLC اولین بخش را دوباره اجرا می کند. برای حل پروسه های کنترلی پیچیده بهتر است که برنامه را به بخش های مختلف تقسیم کنیم. این نوع برنامه نویسی که به عنوان برنامه ساختار یافته مشهور است. دارای امتیازات به شرح ذیل می باشد.

1. برنامه نویسی ساده و شفاف تر می شود.
2. می توان بخش های مختلف برنامه را قانونمند کرد.
3. به راحتی می توان تغییرات مورد نظر را اجرا کرد.
4. تست برنامه راحت تر می شود.
5. با استفاده از تکنیک سابروتین از تکرار برنامه نویسی مشابه جلوگیری می شود.

مقدمه ای بر برنامه نویسی S5

در نمایش نردبانی برنامه بصورت نماد اتصال و سیم پیچ های مدار فرمان رله ای نشان داده می شود. لذا ساختار برنامه شبیه به مدار فرمان رله ای می باشد. در نمایش فلوچارتی از نماد های مستطیل شکل استفاده می شود. این طرز از نمایش بیشتر در هنگام طراحی برنامه استفاده می شود. در هر بلوک عمل منطقی نشان داده می شود. در روش **STATEMENT LIST** یا **STL** از دستورات و یا جملات نوشتاری برای نوشتن برنامه کمک گرفته می شود. هر عبارت دارای دو بخش است.

1. عملکرد یا OPERATION: به عمل منطقی که در یک عبارت صورت می گیرد ، عملکرد گفته می شود، یکسری از عملکردهای مهم در جدول (۶-۱) آورده شده است.

در زبان برنامه نویسی	در زبان محاوره ای	در ریاضی
A	AND	ترکیب عطفی
O	OR	ترکیب فصلی
AN	AND NOT	نقیض ترکیب عطفی
ON	OR NOT	نقیض ترکیب فصلی
=	ASSIGN TO	ترکیب هم ارزی

جدول (۶-۱)

مقدمه ای بر برنامه نویسی S5

2. عملوند یا OPERAND : به قسمتی از عبارت که یک عمل منطقی روی آن انجام می شود گفته می شود. مانند ورودی ها یا I ، خروجی یا Q ، فلگ یا F عملوندهای مورد استفاده در زبان S5 در جدول (۲-۶) آمده است.

جدول (۲-۶)

ورودیها ، ازپروسه تحت کنترل به plc	INPUT	I
خروجیها ، از plc به پروسه تحت کنترل	OUTPUT	Q
فلگها، حافظه ای جهت نگهداری مقادیر میانی حاصل از عملیات باینری	Flags	F
دیتا، حافظه ای جهت نگهداری مقادیر میانی حاصل از عملیات دیجیتال	DATA	D
زمان سنجها ، حافظه ای جهت تفصیص زمان سنجها	TIMERS	T
شمارنده ها ، حافظه ای جهت تفصیص شمارش گرها	Counters	C
ثابت ها	Constant	K

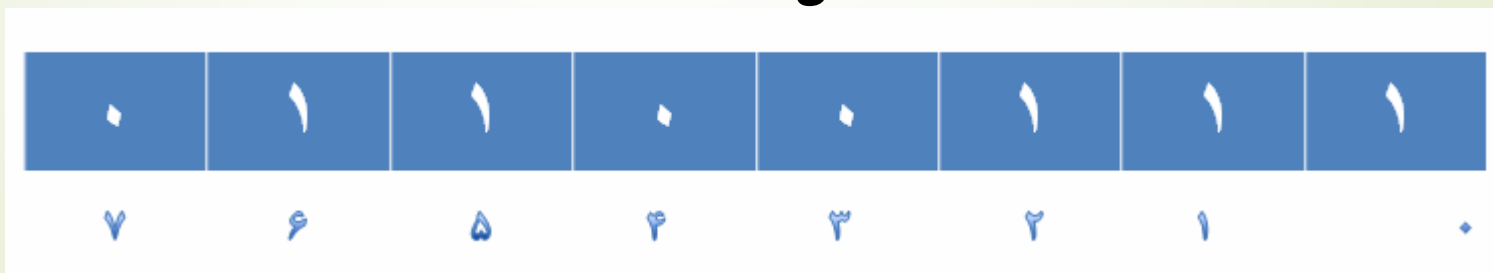
مقدمه ای بر برنامه نویسی S5

نحوه آدرس دهی عملوندها :

هر کدام از عملوندهای ورودی ، خروجی و فلگ در دسته ۸ بیتی سازماندهی می شوند و در آدرس دهی ، ابتدا باید آدرس بایت مربوط سپس بیت مربوط را تعیین نمود . اگر بایت ورودی ۸ ام یعنی **IB8** به صورت شکل (۳-۶) باشد پس **IB8.5** یعنی بیت ۵ از بایت ۸ ام ورودی دارای مقدار ۱ می باشد.



شکل (۳-۶)



شکل (۴-۶)

مقدمه ای بر برنامه نویسی S5

2. بلوک های تابع ساز انتصابی : این بلوک ها توسط کاربر و باتوجه به نیازهای کنترلی پروسه نوشته می شوند، لازم بذکر است. که **FB** ها فقط توسط روش **STL** برنامه نویس می شوند . هر **FB** از دو بخش تشکیل شده:

1. سرخط بلوک که شامل نام و سایر مشخصات **FB** است.

2. بدنه بلوک که شامل توابع و یا دستوراتی است که باید در **FB** اجرا شود.

اگر بایت ۶ ام فلگ به صورت شکل (۴-۶) باشد ، آنگاه **F6.1** یعنی بیت اول از بایت ۶ ام فلگ مقداری برابر ۱ دارد. به همین صورت بایت های خروجی به صورت **QB** می باشد. در روش نمایش **STL** به هر دستور یک رشته یا خط برنامه یا عبارت گفته می شود. هر دستور معمولاً یکی از ترکیب های منطق ریاضی را در بر می گیرد و همچنین کنترل فلیپ فلاپ ها را نیز برعهده دارد. در برنامه نویسی به روش **STL** به هر چند خط برنامه که عمل خاصی را انجام می دهد یک سگمنت گویند.

مقدمه ای بر برنامه نویسی S5

یک برنامه می تواند از یک سگمنت یا بیشتر تشکیل شود. هر برنامه با یک سمیکالون شروع شده و با **BE (BLOCK END)** خاتمه می یابد، برنامه نوشته شده با **LAD** یا **CSF** را می توان به **STL** تبدیل نمود ولی عکس آن لزوما برقرار نیست. سیکل زمانی اجرای برنامه ریزپردازنده از سطر اول برنامه شروع به خواندن و اجرای دستورات می کند تا به دستور **BE** برسد. مدت زمان لازم برای این کار را یک سیکل زمانی اجرای برنامه گویند. برای تسریع در اجرای برنامه و کاهش این سیکل زمانی می توان پردازنده ای به سرعت بالا به کار برد و یا برنامه را سازمان دهی نمود.

بلوک های **OB** :

بلوک های **OB** وظیفه هدایت و مدیریت برنامه را برعهده دارند ، **OB** ها ساختار برنامه استفاده کننده را مشخص می کنند و **OB** با یک شماره خاص مشخص می شود، در تعریف **OB** می توان گفت بلوک هایی هستند. که هر یک وظیفه خاصی را برعهده داشته و در واقع ارتباط بین سیستم عامل و برنامه کاربر را برقرار می کنند.

مقدمه ای بر برنامه نویسی S5

هر یک از بلوک های OB کار بخصوصی را انجام می دهد. این بلوک ها را می توان به ۶ بلوک عمده تقسیم کرد.

1. OB مخصوص اجرای برنامه ی کاربر به صورت سیکلی:

در OB1 بلوک های شامل برنامه کاربر به ترتیبی که کار بر تعیین می کند اجرا می شوند و پس از اجرای آخرین بلوک دوباره بلوک اول اجرا می شود و این کار ادامه پیدا می کند.

2. OB های مخصوص وقفه های پروسه ای :

OB2 تا OB5 مربوط به این وقفه ها هستند. بدین معنی که اگر سیگنال ورودی مشخص تغییر کند ، اجرای برنامه بگونه ای که در OB1 تعریف می شود، متوقف شده و یکی از OB های ۲ تا ۵ که مربوط به تغییر وضعیت آن سیگنال ورودی مشخص می باشد. اجرا می گردد. این OB ها توسط کاربر برنامه نویسی می شوند. پس از اتمام

مقدمه ای بر برنامه نویسی S5

11

اجرای دستورات OB ادامه برنامه طبق روند OB1 و از نقطه ی توقف برنامه به بعد اجرا می شوند .

3. OB های مخصوص کنترل اجرای برنامه از نظر زمانی:

OB10 الی OB13 مربوط به این وقفه ها هستند . برای اجرای یکباره برنامه کاربر به طور کامل بسته به نوع CPU و حجم برنامه نوشته شده و دستورات بکاررفته در برنامه گذشت یک مدت زمان خاص لازم است. پس هر دستور باید منتظر زمان اجرا شدنش باشد. در صورتی که کاربر بخواهد بخش خاصی از برنامه اش در لحظه ای بخصوص اجرا شود و مستقل از زمان اجرا شدنش در روند برنامه باشد. می تواند آن بخش مشخص از برنامه را در OB های ۱۰ تا ۱۳ بنویسد. تا آن را از زمان اجرا شدن کنترل کند.

4. OB مخصوص وقفه زمانی:

مقدمه ای بر برنامه نویسی S5

OB6 مخصوص به وقفه زمانی است. یعنی اگر چند پریود از زمان اجرای برنامه کاربر بگذرد، ولی برنامه بطور کامل اجرا نشود، یک وقفه زمانی رخ می دهد و اجرای برنامه در **OB1** متوقف شده و برنامه **OB6** اجرا می شود. کاربر می تواند تدبیر خود را برای مواجهه با چنین شرایطی در این بلوک بنویسد.

5. OB های مخصوص راه اندازی :

OB21 و OB22 بترتیب برای حالت های **RUN TO STOP** و **POWER OFF** **TO POWER ON** هستند. از این رو کاربر می تواند راه حل های مورد نظر خود را در صورت وقوع این حالات در این دو **OB** بنویسد.

6. OB های مخصوص وقوع خطا در **PLC** :

OB19 ، OB23 ، OB24 ، OB27 ، OB32 و OB34 مخصوص بروز خطا هستند. این **OB** ها در هنگام بروز خطا اجرا شده و کاربر برنامه ای که در نظر دارد که در هنگام وقوع آن خطای خاص اجرا شود را در آن **OB** می نویسد.

مقدمه ای بر برنامه نویسی S5

بلوک ها PB (PROGRAM BLOCK)

بلوک برنامه یا PB با توجه به پروسه تحت کنترل شامل دستوراتی است که کاربر برای کنترل پروسه می نویسد. به عبارت دیگر PB ها بلوک های تشکیل دهنده برنامه کنترل یک فرایند می باشد. که بدلیل مختلف از جمله جلوگیری از پیچیدگی و طولانی شدن برنامه ، کنترل فرایند به قسمت های کوچکتری که همان PB ها هستند تفکیک می شود، مثلا کاربر باتوجه به سلیقه و برداشت خود از فرایند تحت کنترل می تواند دستوراتی که مربوط به یک دیگر بوده و از نظر فنی عملیات خاصی را انجام می دهد ، در یک PB قرار دهد.

بلوک های FB (FUNCTION BLOCK)

بلوک های مخصوصی هستند که قسمت هایی از برنامه را که بارها تکرار می شوند و یا پیچیدگی های مخصوص دارند را در خود جا می دهند. این بلوک ها همانند توابع عمل کرده به این ترتیب که تعدادی ورودی را گرفته و تعدادی خروجی را در اختیار

مقدمه ای بر برنامه نویسی S5

قرار می دهند. این بلوک ها را می توان در هر قسمت از برنامه صدا کرد. در تقسیم بندی کلی FB ها را به می توان به دو دسته تقسیم نمود.

1. بلوک های تابع ساز استاندارد:

این بلوک ها شامل توابعی هستند که در آن ها اعمال منطقی نظیر ضرب ، تقسیم و ... تعریف می شوند و به صورت بسته های نرم افزاری توسط شرکت سازنده نوشته شده و همچنین به همراه یک دفترچه راهنما که حاوی اطلاعاتی در باره چگونگی وارد نمودن پارامترها و دیگر اطلاعات می باشد در اختیار کاربر قرار می گیرد.

2. بلوک های تابع ساز انتصابی:

این بلوک ها توسط کاربر و با توجه به نیازهای کنترلی پروسه نوشته می شوند، لازم به ذکر است که FB ها فقط توسط روش STL برنامه نویسی می شوند.

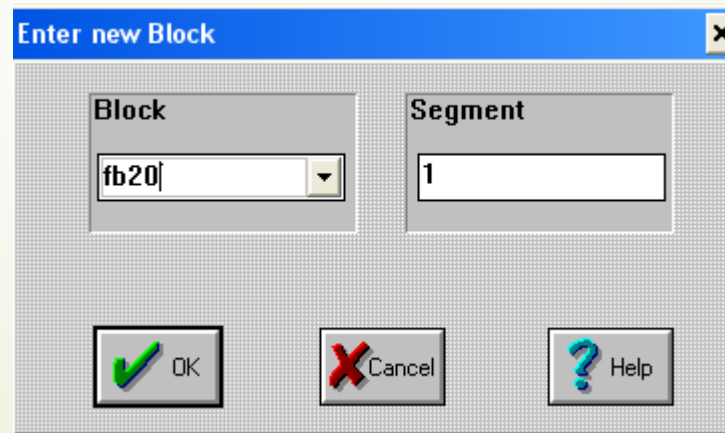
مقدمه ای بر برنامه نویسی S5

هر FB از دوبرخش تشکیل می شود:

1. سر خط بلوک که شامل نام و مشخصات FB است.

2. بدنه بلوک که شامل توابع و یا دستوراتی است که باید در FB اجرا شود.

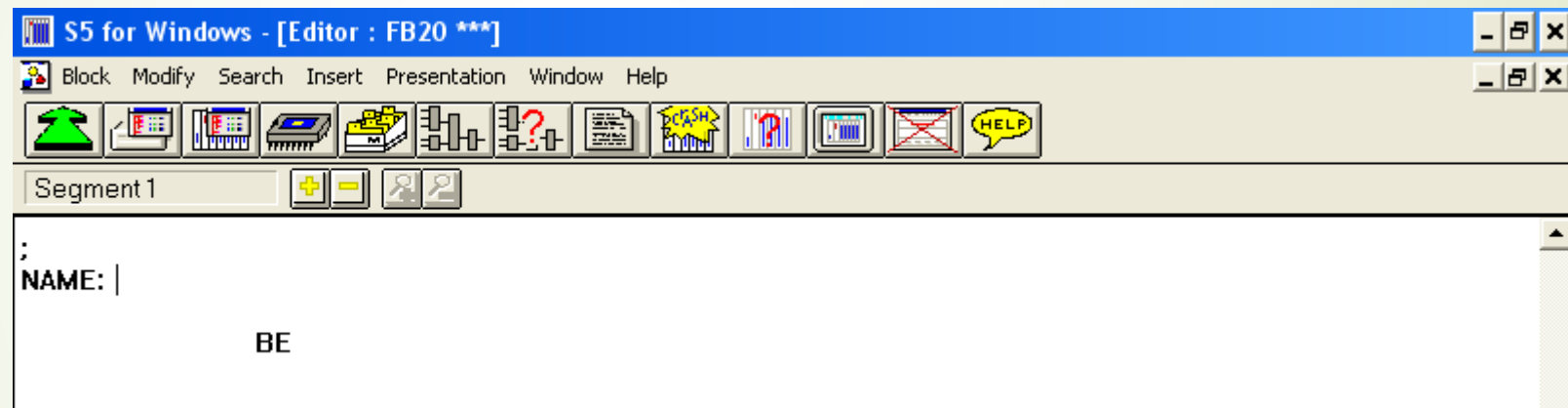
به دلیل آنکه نحوه تعریف و وارد نمودن پارامترهای انتصابی یک بلوک FB کمی سوال برانگیز می باشد و با روش ایجاد بلوک های OB ، PB و DB متفاوت است. در اینجا قصد داریم بلوک FB20 و به دنبال آن PB3 که بلوک مذکور را فراخوانی می کند بازنویسی کنیم . برای ایجاد بلوک FB20 در یک فایل پروژه جدید ، مسیر BLOCK بعد NEW BLOCK انتخاب می کنیم ، و در پنجره ظاهر شده نام بلوک را FB20 قرار می دهیم.



شکل (۵-۶)

مقدمه ای بر برنامه نویسی S5

در اولین سطر از این بلوک مطابق شکل (۶-۶) از کاربر خواسته می شود نامی برای این بلوک تابع ساز انتخاب کند. حداکثر تعداد حروف قابل استفاده برای نامگذاری

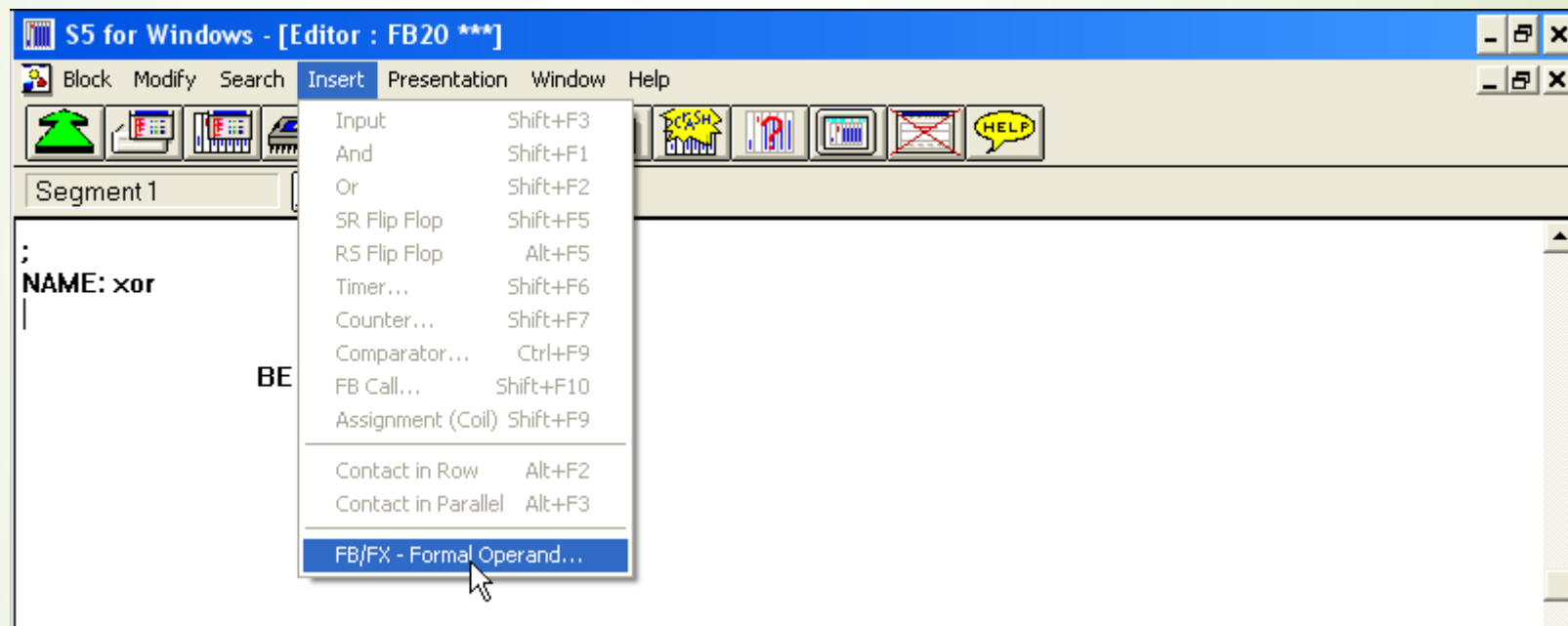


شکل (۶-۶)

یک متغیر ۴ حرف می باشد، کاراکترهای نام انتخاب شده به صورت خودکار به شکل حروف بزرگ نوشته می شوند. در این قسمت نام XOR را وارد کنید.

مقدمه ای بر برنامه نویسی S5

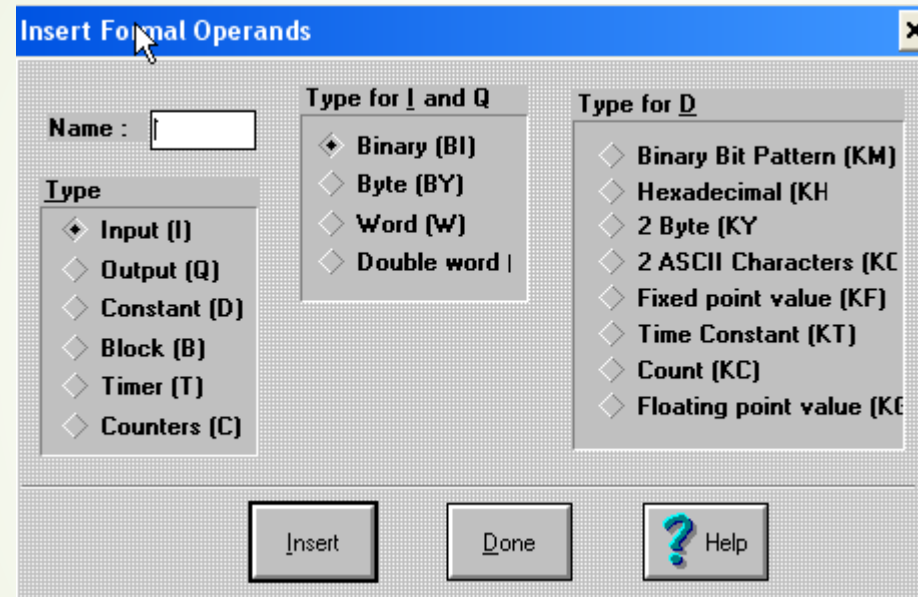
حال برای تعریف متغیرها و پارامترها این بلوک مطابق شکل (۶-۷) عمل می کنیم.



شکل (۶-۷)

مقدمه ای بر برنامه نویسی S5

در این حالت یک پنجره به شکل (۶-۸) باز می شود. که در آن نوع و نام پارامتر مورد نظر تعیین می شود.



شکل (۶-۸)

در بخش NAME عبارت MOT1 را وارد نموده و برای تعیین نوع پارامتر در دو بخش TYPE و TYPE FOR I AND Q به ترتیب دو گزینه input(I) و (BI) Binary را انتخاب کنید. سپس بروی گزینه Insert کلیک کنید و سپس برای برگشت به پنجره اصلی بروی گزینه Done کلیک کنید.

مقدمه ای بر برنامه نویسی S5

همین مراحل را برای **MOT2** انجام داده حال کد برنامه به شکل (۶-۹) تغییر می کند.

```
;
NAME: xor
DECL:      MOT1      IBI
DECL:      MOT2      IBI
|
```

شکل (۶-۹)

BE

همانطور که در شکل بالا دیده می شود دو خط به برنامه اضافه شد و اگر توجه شود ، روبروی کدهای **MOT1** و **MOT2** ، **IBI** نوشته شده است. **I** اول به معنای ورودی و **BI** به معنای باینری (یعنی به صورت یک بیت) در این بلوک مورد استفاده قرار می گیرد. حال متغیر سوم به نام **LED** را معرفی می کنیم ، که از نوع **OUTPUT** می باشد. در این حالت کد برنامه به شکل (۶-۱۰) تبدیل می شود

مقدمه ای بر برنامه نویسی S5

```

;
NAME: xor
DECL:      MOT1      IBI
DECL:      MOT2      IBI
DECL:      LED       QBI
          BE

```

شکل (۶-۱۰)

حال سطرهای دیگر برنامه را می نویسیم. در هنگام تایپ کدها نیاز به رعایت فاصله ها نیست . پس از تایپ کل برنامه کلید **f9** یا (از منو **presentation** گزینه **ladder**) را می زنیم، کل برنامه مرتب می شود و به شکل (۶-۱۱) تبدیل می شود.

```

;
NAME:      XOR
DECL:      MOT1      IBI
DECL:      MOT2      IBI
DECL:      LED       QBI

          A
          AN          =MOT1
          AN          =MOT2
          A           =MOT2
          =           =LED
          BE

```

شکل (۶-۱۱)

مقدمه ای بر برنامه نویسی S5

حال برنامه را از طریق مسیر **block** بعد زدن گزینه **Save** ، ذخیره می کنیم. حال بلوک **PB3** که بلوک **FB20** را فراخوانی می کند ف باز نویسی می کنیم. جهت ایجاد این بلوک ابتدا آیکن  را زده سپس گزینه **BLOCK** را می زنیم و از آن **NEW BLOCK** را انتخاب می کنیم و نام آن را **PB3** می گذاریم و باانتخاب مسیر **PRESENTATION** گزینه **STATEMENT LIST(STL)** را می زنیم . تا زبان برنامه نویسی **PB3** در روش **STL** قرار می گیرد. حال کلید اینتر را فشار داده و در سطر بعدی **Jufb20** را وارد کنید سپس کلید **f9** را فشار دهید . همانطور که ملاحظه می کنید. شکل برنامه از حالت (۶-۱۲) به (۶-۱۳) تبدیل می شود.

;	;	JU	FB 20
jufb20	NAME:	XOR	
BE	MOT1:	F 0.0	
	MOT2:	F 0.0	
	LED :	F 0.0	
		BE	

شکل (۶-۱۲)

شکل (۶-۱۳)

مقدمه ای بر برنامه نویسی S5

در شکل (۶-۱۳) نام در نظر گرفته شده برای بلوک **FB20** یعنی **XOR** به همراه سه پارامتر تعریف شده در آن بروی صفحه ظاهر می شود. پس از فراخوانی هر بلوک **FB** کامپایلر به صورت پیش فرض عملوندهایی برای متغیرهای استفاده شده در بلوک در نظر می گیرد. در این بلوک برای انتصاب هر سه متغیر از عملوند **F0.0** استفاده شده است. حال عملوندهای ذکر شده را به عملوندهای نسبت داده شده به هریک از متغیرها تغییر می دهیم و به کدی به شکل (۶-۱۴) می رسیم. سپس آن را ذخیره می کنیم.

```

:
                                JU          FB 20
NAME:                          XOR
MOT1:                          i 0.0
MOT2:                          i 0.1
LED :                          Q 2.0
                                BE

```

شکل (۶-۱۴)

مقدمه ای بر برنامه نویسی S5

: DATA BLOCK

بلوک DB اطلاعات و داده های مورد نیاز برای اجرای برنامه کنترلی مانند مقادیر تایمرها ، شمارنده ها و دیگر اعداد ثابت مورد نیاز برنامه را در خود نگهداری می کند ، همچنین در برخی از پروسه ها لازم است ، مواردی همچون پیغام ها ، آلارم ها و ... بروی صفحه نمایش ظاهر می شوند ، محل نگهداری این پیام ها بلوک های اطلاعاتی می باشند سه نوع از اطلاعات در بلوک های DB وجود دارند.

1. اطلاعات یا DATA

2. متن یا TEXT

3. الگوی بیت یا BIT PATTERN

(این الگو تعدادی بیت ۰ - ۱ بصورت بایتی یا کلمه ای است، که عمل سیگنالینگ ، روشن و خاموش کردن خروجی را بر عهده دارند.)

مقدمه ای بر برنامه نویسی S5

می توان هر بلوک اطلاعات **DB** را فراخوانی نمود. برای فراخوانی سطر صدم از **DB50** به صورت شکل (۶-۱۵) عمل می کنیم.

C DB 50

نام بلوک

شکل (۶-۱۵)

L DW 100

نام سطر

اطلاعات ذخیره شده در **DB** ها به ۸ صورت که در شکل (۶-۱۶) آمده است.

(16Bits: 0000_(H) → FFFF_(H))

(16Bits: -32768 → 3278)

(14Bits: 001.0 → 999.3)(زمان)

(12Bits: 000 → 999)

KH ➤ 16Bits: (00...0)_(16Bits) → (11...1)_(16Bits)

KM ➤

KF ➤ (32Bits:Dmord)

KG ➤

KT ➤

KS ➤

KC ➤

KY ➤

شکل (۶-۱۶)

مقدمه ای بر برنامه نویسی S5

1. KH : برای اعداد در مبنای ۱۶
2. KF : برای اعداد در مبنای ۱۰
3. KT : برای اعداد ثابت
4. KC : برای شمارنده ها
5. KY : ۱۶ بیت که به دو بایت چپ و راست تقسیم شده و با نماد DL و DR وجود دارند که کاملاً مجزا از هم می باشند.
6. KM
7. KG
8. KS

مقدمه ای بر برنامه نویسی S5

ایجاد بلوک DB :

از آنجایی که این بلوک ها جزو بلوک های اطلاعاتی هستند . فقط به زبان STL نوشته شده و در بلوک های دیگر ما به ازاء ندارند. پس طبق موارد قبل از منوی **BLOCK** گزینه **NEW BLOCK** را زده و آن را **DB28** می نامیم. پس متنی به شکل (۶-۱۷) را تایپ کنید و در پایان کلید **f9** را بزنید و آن را ذخیر کنید. و حاصل به شکل (۶-۱۸) در می آید.

```
;
kf0
kf1
kf2000
```

شکل (۶-۱۷)

```
;
0:      KF +0
1:      KF +1
2:      KF +2000
```

شکل (۶-۱۸)