جلسه دوم

. . . . . .

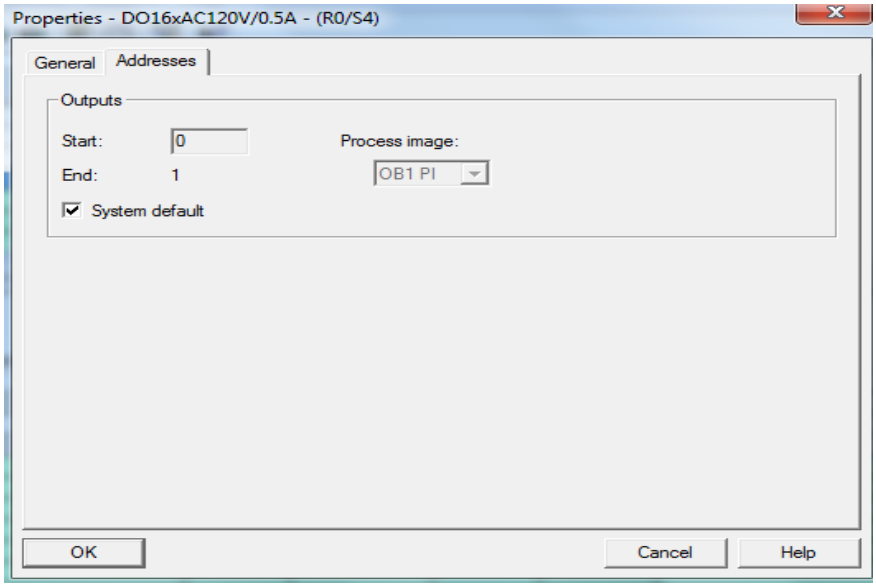
انواع کارت های ورودی و خروجی (انالوگ ودیجیتال)

تنظیم پارامترهای کارت های DO (Digital out put) :

تقسیم بندی کارت های do :

از نظر تعداد خروجی	از نظر جریان خروجی	از نظر ولتاژ	از نظر قابلیت های خاص
۴ خروجی	بدون رله با جریان های 2A ,1.5,1,0.5 با رله و جریان های 8A,5	24 VDC	بدون ویژگی خاص
۸ خروجی		48VDC	تشخیص قطعی
۱۶ خروجی		120VAC	تشخیص اتصال
۳۲ خروجی		230VAC	واکنش در موقع توقف CPU

برای کارت هایی که قابلیت خاص نداشته باشند زمانی روی انها کلیلک کنیم پنجره زیر ظاهر میشود:



General : در این بخش توضیحاتی راجع به کارت و ویژگی ها و کد سفارش ان همراه با نام امده است.

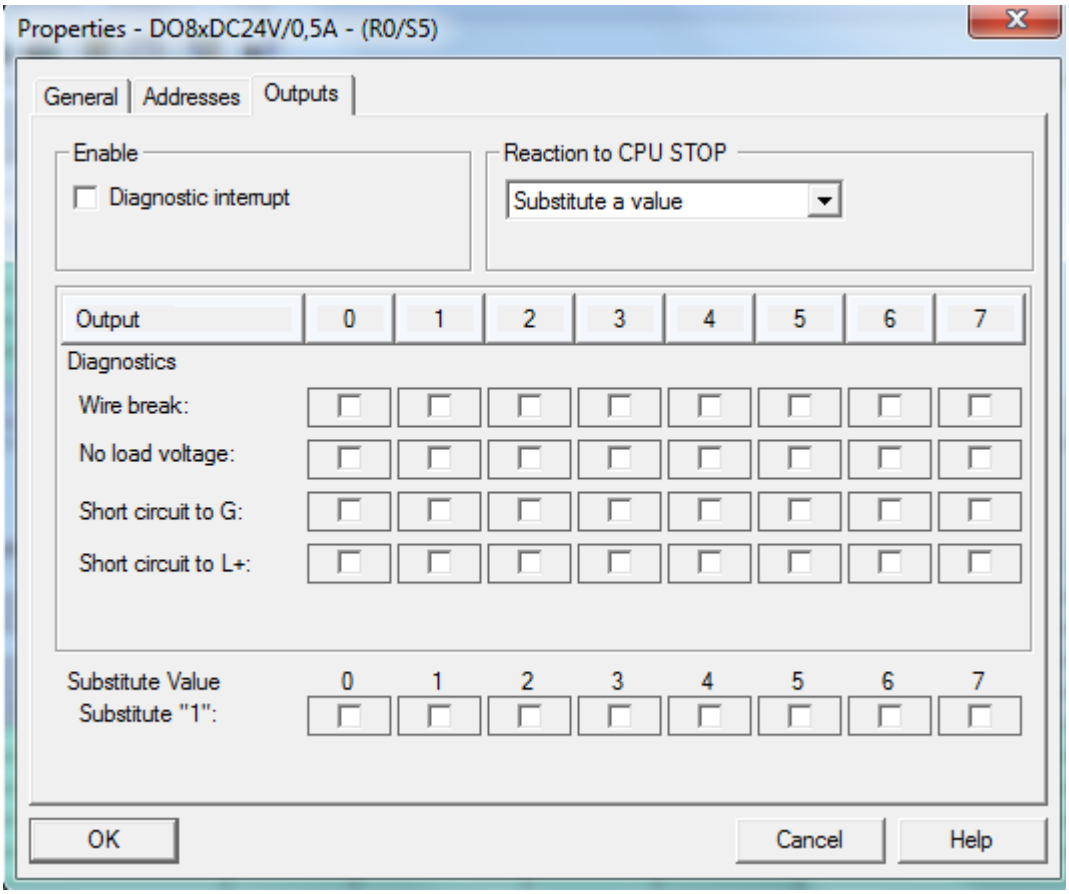
Address : در این بخش ادرس هایی که توسط سیستم به کارت اختصاص داده شده امده است.
Start ادرس شروع و end ادرس نهایی رانشان می دهد.

بعنوان مثال در شکل بالا برای کارت do با ۱۶ خروجی مشاهده میکنیم که ادرس شروع ۰ و ادرس نهایی ۱ است.

و دارای ۲ بایت ادرس است.

ادرس	کانال
۰.۰	۰
۰.۱	۱
.	
.	
۰.۷	۷
۱.۰	۸
.	
.	
۱.۷	۱۵

برخی از کارت ها که دارای ویژگی های خاص باشند در بخش properties یخش اضافی به اسم output دارند.



Wire break : برای تشخیص قطعی سیم قبل از ارسال خروجی

No load voltage : برای تشخیص عدم وجود ولتاژ

Short circuit to m : برای تشخیص اتصال کوتاه به زمین

Short circuit l+ : برای تشخیص اتصال کوتاه در ۲۴ ولت

برای هر یک از موارد زیر میتوان کانال مورد نظر رو علامت زد و فعال نمود.

Reaction last value : توسط این قسمت می توان تعیین کرد که در صورت توقف cpu خروجی کانال مربوطه از do چگونه باشد.ممکن است به خاطر مسایل امنیتی کاربر بخواهد این خروجی ها را یک نگه دارد.

با انتخاب substitute value و علامت زدن کانال ها در زیر ان این امر میسر می شود.(خروجی کانالی که علامت زده نشود صفر خواهد بود.)

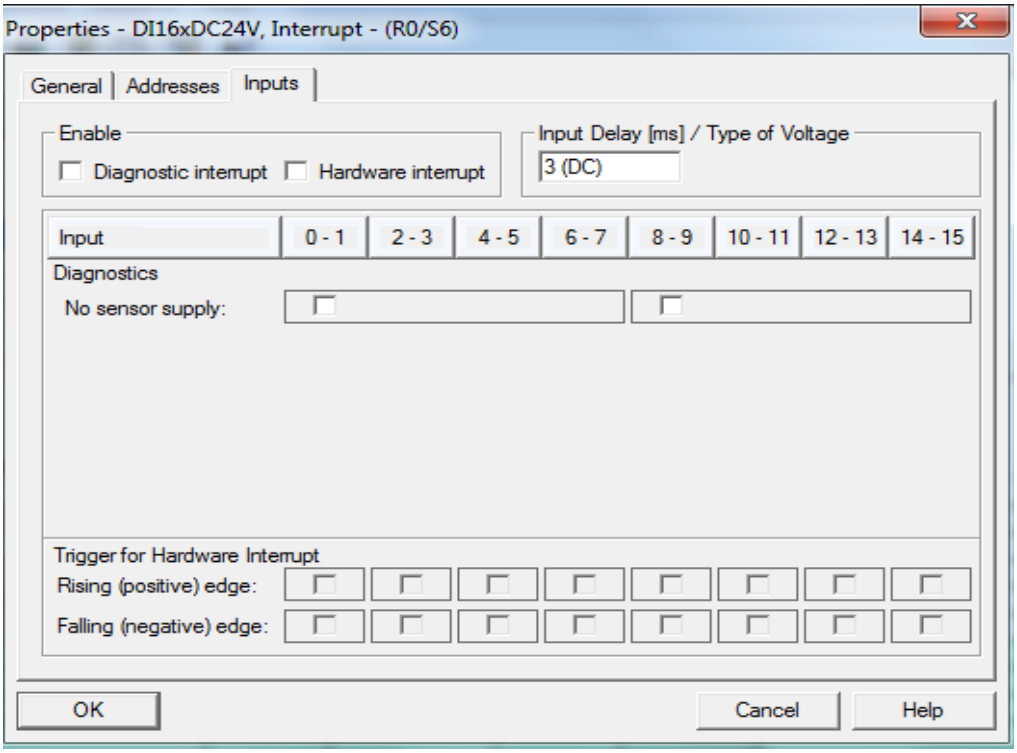
اگر keep last value انتخاب شود در هنگام توقف cpu اخرین مقدار قبلی سیگنال حفظ می گردد.

تنظیم پارامتر های کارت های DI (DIGITAL INPUT)

تقسیم بندی کارت های DI :

از نظر تعداد ورودی	از نظر ولتاژ	از نظر قابلیت های خاص
۴ ورودی	24VDC	بدون ویژگی خاص
۸ ورودی	48VDC	تشخیص قطع شدن تغذیه
۱۶ ورودی	120VDC	ایجاد وقفه براساس لبه ورودی
۳۲ ورودی	230VDC	تاخیر در گرفتن ورودی

برای کارت هایی که دارای قابلیت های خاص باشند در بخش properties قسمتی به نام inputs اضافه میشود.

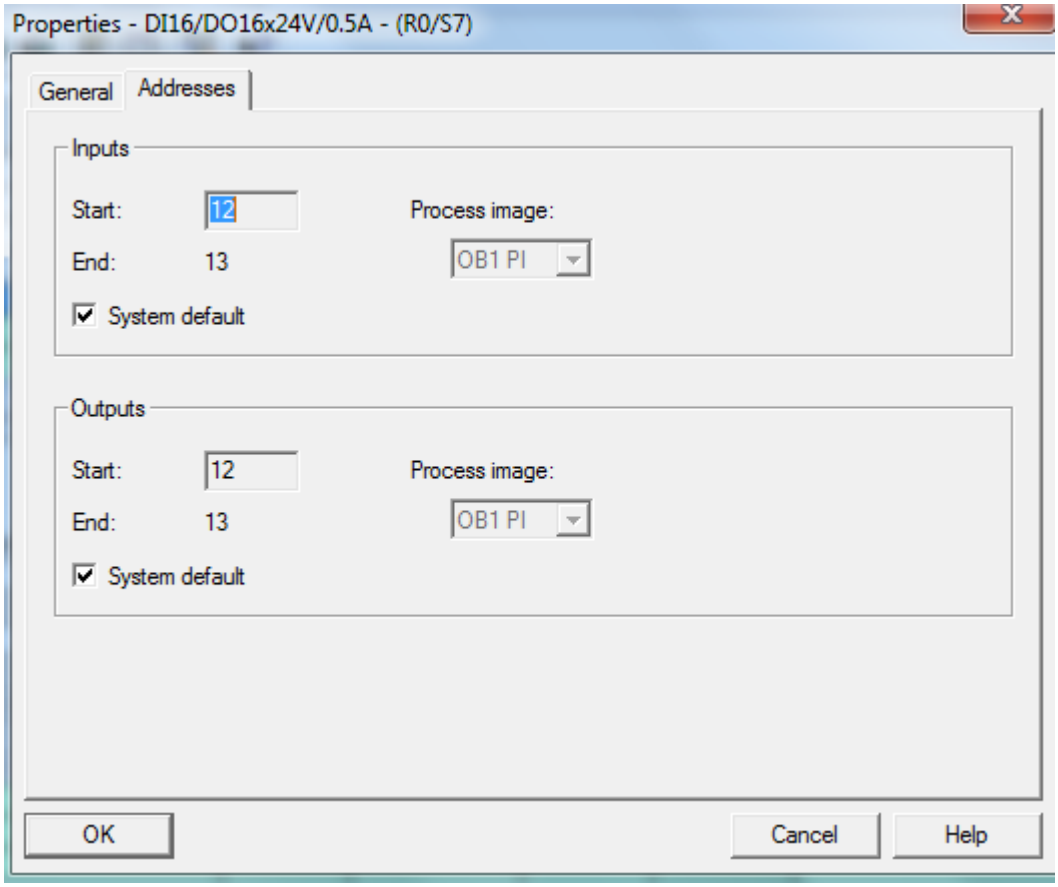


Diagnostic interrupt : در حالت عادی غیر فعال است و اگر فعال شود در صورت قطع شدن تغذیه سنسور شماره کانال مربوطه در بافر تشخیص عیب cpu ثبت میشود.همانطور که ملاحظه می شود در جلوی NO SENSOR SUPPLY یک گزینه برای ورودی های ۰ تا۷ و یک گزینه برای ورودی های ۸ تا ۱۵ وجود دارد.میتوان هردو یا یکی را بدلخواه فعال کرد .

HARDWARE INTERRUPT : در حالت عادی غیر فعال است و اگر فعال شود جدول پایین که مربوط به تریگر کردن این وقفه است فعال میگردد.در این جددول برای هر دو کانال ورودی یک گزینه وجود دارد..با انتخاب این گزینه میتوان تعیین کرد که وقتی ورودی کانال (لبه مثبت یا منفی) تغییر میکند وقفه نیز اعمال نماید.

تنظیم کارت های DI/DO :

این کارت ها ورودی و خروجی دیجیتال را باهم دارند .



صرفا دارای دو بخش address و general میباشد.

در مورد کارت های انالوگ در جلسات بعد توضیح داده می شود.

مدهای کاری plc :

Stop : در این مد پردازش برنامه متوقف میشود.دسترسی به I/O وجود ندارد. CPU به صورت read و write قابل دسترسی است.یعنی می توان برنامه ان را خواند یا برنامه جدیدی را به ان انتقال داد.

Run : در این مد برنامه اجرا میشود cpu به I/O ها دسترسی دارد و برنامه CPU به صورت read only است.یعنی نمی توان برنامه جدیدی را به ان download کرد.

Run-p : در این مد برنامه اجرا می شود . cpu به I/O ها دسترسی دارد در عین حال CPU بصورت READ و WRITE قابل دسترسی است.

MRES : این وضعیت برای ریست کردن حافظه CPU بکار میرود.یعنی هم مقادیر متغیر های حافظه و هم برنامه ای که توسط کاربر به حافظه ارسال شده پاک میگردد..

فرمت ادرس دهی ورودی ها:

ورودی های PLC میتواند از نوع **bit** و **byte** و**word** و **dword** باشد.

برای ادرس دهی یک بیت ابتدا باید شماره بایت رو بنویسیم و سپس با گذاشتن نقطه ادرس بیت را در ان بایت مشخص کنیم.

مثال)

I2.0

یعنی بیت صفرام از بایت دوم

کلیه ادرس های ورودی در S7 با I شروع میشوند.جدول زیر انواع ادرس دهی های ورودی را مشخص میکند.

نوع ورودی	نحوه نمایش	مثال
BIT	I	I0.1
BYTE	IB	IB1
WORD	IW	IM2
DWORD	ID	ID8

توجه:

وقتی یک IW را در برنامه بکار میبریم ادرس IW بعدی باید حداقل دو بایت با ادرس قبلی فاصله داشته باشد مثلا

IW2 , IW4 ,IW6,.....

IW 2 = BYTE 2 + BYTE 3

IW 3 = BYTE 3 + BYTE4

در بایت سه مشترک است.بنابراین اشتباه است و نمیتوان نوشت IW3(تداخل ادرسی)

برای DWORD نیز ادرس بعدی حداقل ۴ بایت فاصله داشته باشد

ID0,ID4,ID8,.....

ادرس دهی خروجی ها در PLC :

انچه برای ورودی ها شرح داده شد برای خروجی صادق است و بجای I از Q استفاده میکنیم.

Q,QB,QW,QD

ادرس دهی متغیرهای حافظه:

نوع متغیر حافظه	نحوه نمایش	مثال
BIT	M	M0.0
BYTE	MB	MB1
WORD	MW	MW2
DWORD	MD	MD8

ادرس دهی کانتر ها و تایمر ها :

تایمر با علامت T و کانتر با علامت C نمایش داده میشود

ادرس انها با یک علامت صحیح که بعد از انها بکار میرود.

مثال: T2 یا C4

آکومولاتور ها و رجیستر های حافظه CPU S7 :

آکومولاتور ها :

بیشتر CPU های S7 شبیه S5 دارای ۲ آکومولاتور هستند که با ACCU1 و ACCU2 شناخته می شوند. برخی CPU های S7 دارای ۴ آکومولاتور می باشند .یعنی علاوه بر ۲ مورد فوق ACCU3 و ACCU4 را نیز دارند.

مقادیری که به حافظه بار می شوند در ACUU1 قرار میگیرند در این شرایط وقتی مقدار جدیدی نیز قرار باشد که بلافاصله به حافظه بار شود در این صورت ابتدا محتویات ACUU1 به ACCU 2 منتقل شده و مقدار جدید وارد ACCU1 میگردد.

هرکدام از آکومولاتور ها ۳۲ بیتی هستند.

رجیسترها و STACK های حافظه CPU :

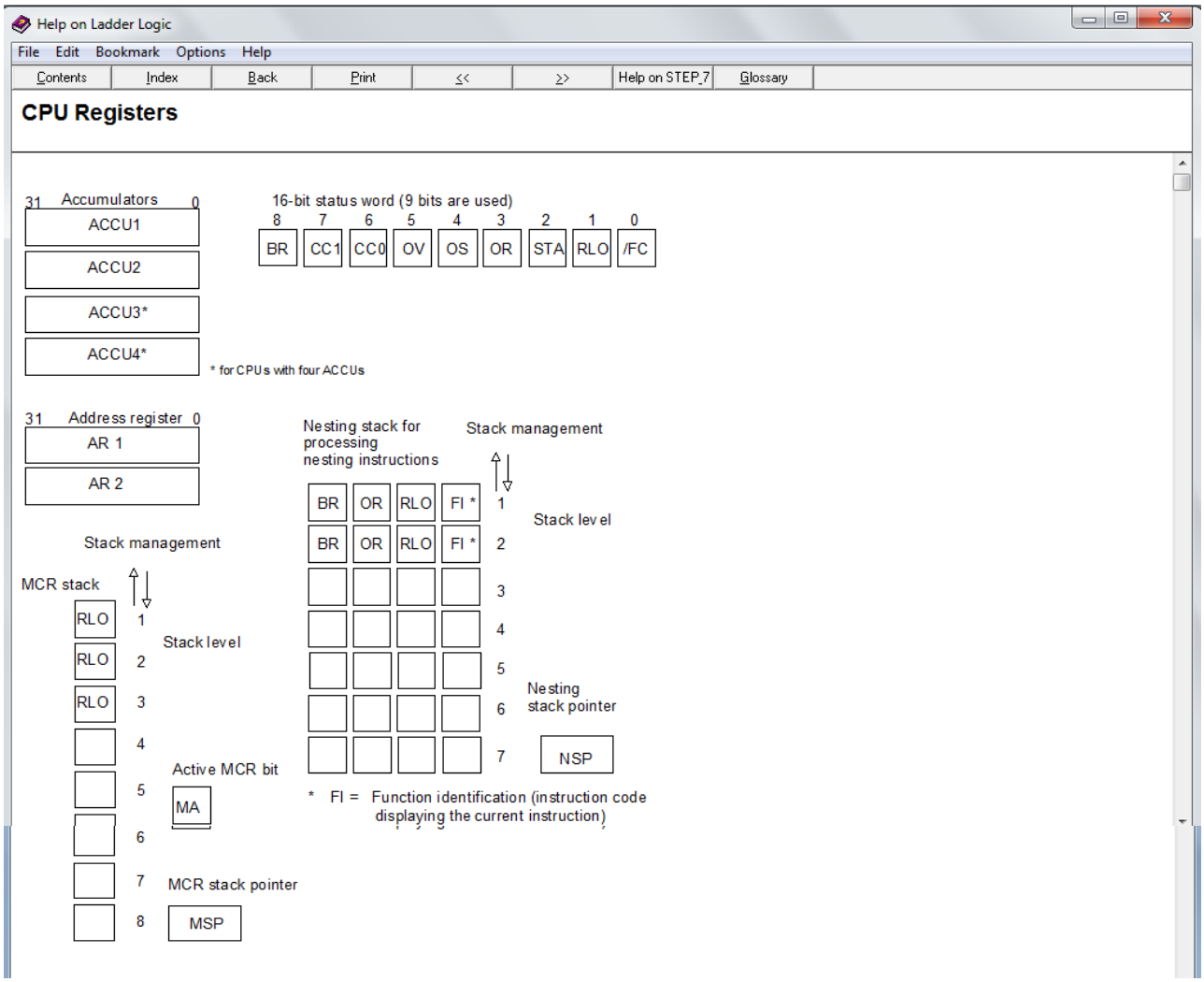
CPU های S7 دارای رجیسترها و STACK های مختلفی می باشد:

Status word : یک رجیستر ۱۶ بیتی است که ۹ بیت ان در هنگام پردازش برنامه بسته به نتایج پردازش تغییر میکند.

Address registers : دو رجیستر ۳۲ بیتی برای ذخیره ادرس است.

Nesting stack :پشته ای است که دارای ۷ سطح بوده و برای وقتیکه حالت nesting در برنامه اتفاق می افتد مانند پرانتز های تودرتو استفاده می شود.با باز شدن هر پرانتز مقدار RLO در این پشته ذخیره شده و با بسته شدن پرانتز با RLOجدید ترکیب می شود

MCR stack : پشته ای است که مربوط به دستورات master control relay می باشد.



رجيستر status word :

BR	CC1	CCO	OS	OV	OR	RLO	STA	FC

بیت های CC0 و CC1 :

نتیجه عملیات	CC0	CC1
مثبت	۰	۱
منفی	۱	۰
صفر	۰	۰
حالت تعریف نشده	۱	۱

بیت OV :

این بیت اگر در نتیجه عملیات محاسباتی سرریزی (OVERFLOW) اتفاق بیفتد ۱ میشود وقتی برنامه در همان سیکل اسکن به دستور محاسباتی جدیدی برسد این بیت ریست شده وبراساس نتایج جدید UPDATE میگردد.

بیت OS :

این بیت معرف Over flow stored است یعنی سرریز قبلی که در همان سیکل اسکن اتفاق افتاده را ذخیره میکند و برخلاف ov که با رسیدن دستور محاسباتی جدید ریست می شود این بیت مقدار سرریزی را تا پایان ان اسکن حفظ می کند.

بیت OR :

این بیت در صورتی یک میشود که در خلال دستورات منطقی(BIT LOGIC) عمل AND قبل از OR وجود داشته باشد.

بیت STA :

این بیت دقیقا وضعیت سیگنال را نشان میدهد که یک است یا صفر

بیت RLO :

این بیت معرف نتیجه عملیات منطقی (Result of logic operation) است.مثلا وقتی دو سیگنال یکی صفر و دیگری یک با هم and شوند این بیت نتیجه یعنی صفر را نشان می دهد.

بیت FC :

این بیت معرف First check است در واقع RLO را راهنمایی میکند که وارد NET WORK جدید از برنامه شده یا بلاکی که صدا زده شده برگشت به بلاک ماقبل اتفاق افتاده است در این شرایط باید RLO نتایج قبلی را دور بریزد و متناسب با عملیات جدید UPDATE شود.

بیت BR :

اگر بخواهیم تحت شرایطی که در First check ذکر شد نتیجه RLO قبلی را هم داشته باشیم این نتیجه با دستور برنامه نویسی SAVE در BR ذخیره می شود.