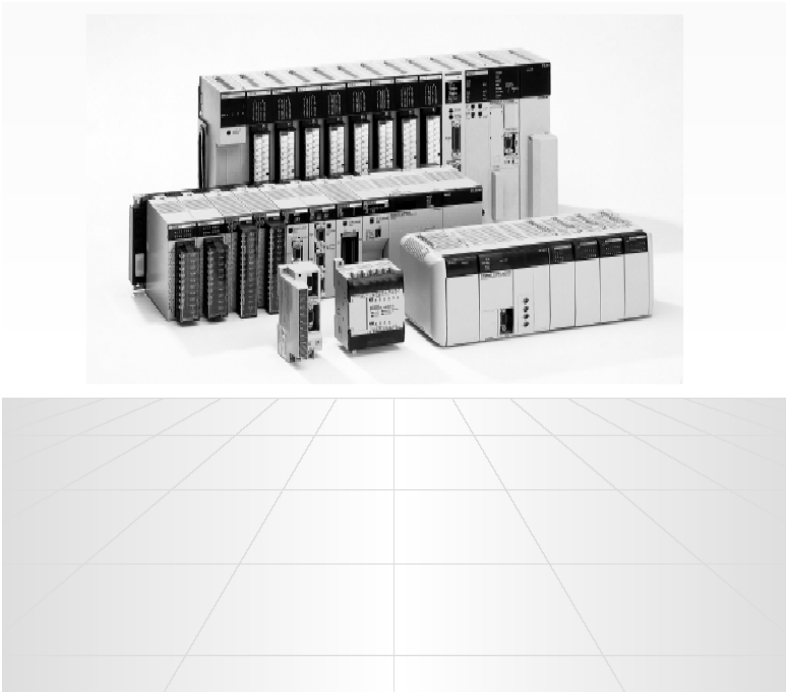
مطالب درسی دوره ی PLC I



مقدمه

به طور کلی سیستم های کنترل مجموعه ای از تجهیزات الکترونیکی هستند کـه در جهـت تضـمین پایـداری, دقت و صحت یک فرآیند, برای رسیدن به یک هدف کنترلی در کنار هم قرار گرفته اند. این تجهیـزات باتوجـه به نوع کاربرد و نوع کنترل، به کار گرفته می شوند. در نتیجهٔ پیشرفت سریع تکنولـوژی, انجـام و پیـاده سـازی کنترل های پیچیده, توسط سیستمهای کنترل خودکار همانند کنترل کننده های منطقی برنامـه پـذیر (PLC) انجام می گیرد.

PLC ها برای اولین بار در اوایل دهـهٔ 1970 در صـنعت اتوماسـیون مـورد اسـتفاده قـرار گرفتـه و جـایگزین تابلوهای عظیم رله ای شدند، این سیستم ها با حجم کوچک ولی قابلیتهای فراوان به عنوان قلـب سیسـتمهای کنترل به کار گرفته شدند.

در دنیای پررقابت امروز کارآمد بودن، انعطاف پذیری و همچنین مقرون به صـرفه بـودن از لحـاظ اقتصـادی از جمله مواردی است که باعث می شود تولید کنندگان بتوانند در عرصـهٔ رقابـت پـا برجـا بـاقی بماننـد. شـرکت OMRON نیز با در نظر گرفتن این فاکتورها سعی بر آن داشته که با ارائه محصـولات بـا عملکـرد سـاده ولـی سریع، مطمئن و انعطاف پذیر یکی از تولید کنندگان معتبر و اصلی در زمینهٔ تولید کنترل کننده های منطقـی برنامه پذیر باشد.

در این جزوه سعی بر آن بوده است که شرح واضـح و کـاربردی، دربـارهٔ سـخت افـزار و نـرم افـزار و همچنـین کاربردهای کنترل کننده های منطقی برنامه پذیر (PLC) ساخت شرکت OMRON، ارائه شود. با این امید کـه توانسته باشیم گامی هر چند کوچک در جهت آشنایی کارشناسان و علاقه مندان به PLC و صنعت اتوماسـیون برداریم.

سعیده لاله ریحانی
تابستان 1385

1- مقایسهٔ سیستمهای کنترل در گذشته و حال 1

1-1- رله ها 1
1-2- PLC ها 2

2- مقدمه ای دربارهٔ PLC 3

2-1- توابع PLC 3
2-2- بلوک دیاگرام ساختمان PLC 4

3- آشنایی با PLC های ساخت شرکت OMRON 10

3-1- PLC های سری فشرده (COMPACT PLC SERIES) - قابلیت پذیرش تا 320 ورودی/خروجی 10
3-2- PLC های سری ماژولار (MODULAR PLC SERIES)- قابلیت پذیرش تا 2500 ورودی/خروجی 11
3-3- PLC های سری RACK (RACK PLC SERIES)- قابلیت پذیرش تا 5000 ورودی/خروجی 11

4- سخت افزار PLC 12

4-1- سری CJ 12
4-1-1- نمای بسط داده شده 12
4-1-2- واحد CPU 13
4-1-3- جزییات واحد CPU 14
4-1-4- نمایشگرهای LED 14
4-1-5- تعریف دیپ سوئیچ ها 16
4-2- سری CS 17
4-2-1- نمای بسط داده شده 17
4-2-2- واحد CPU 17
4-2-3- جزییات واحد CPU 18
4-3- کارت های ورودی/خروجی 19
4-3-1- اطلاعات کلی کارتهای ورودی/خروجی 19

5- فضای حافظه 21

5-1- تعریف فضای داده ها 21
5-1-1- ناحیهٔ CIO (Common Input/Output Area) 21
5-1-2- ناحیهٔ W (Work Area) 22
5-1-3- ناحیهٔ H (Holding Area) 22
5-1-4- ناحیهٔ A (Auxiliary Area) 22
5-1-5- ناحیهٔ D (Data Memory Area) 23
5-1-6- ناحیهٔ E (Extended Data Memory Area) 24
تعیین آدرسهای ناحیهٔ E 24
مشخص کردن بانک و آدرس مورد نظر در آن بانک 25
مشخص کردن آدرس در بانک فعلی 25
5-1-7- ناحیهٔ تایمر (Timer Area) 25
5-1-8- ناحیهٔ کانتر (Counter Area) 26

5-1-9- فلگهای وضعیت 27
5-1-10- پالسهای زمانی 28
5-1-11- ناحیهٔ TK (Task Flag Area) 28
5-1-12- ناحیهٔ IR (Index Register) 28
5-1-13- ناحیهٔ DR (Data Register) 28
5-2- نقشهٔ حافظه 29

6- مقدمه ای بر CX-PROGRAMMER 30

6-1- آماده شدن برای کار با CX-PROGRAMMER 30
6-2- کار با امکانات بخش MEMORY 32
6-2-1- داده های باینری 32
6-2-2- داده های هگزادسیمال 33
6-2-3- داده های BCD 33
6-2-4- داده های دسیمال 34

7- تخصیص آدرس ورودی/ خروجی 35

7-1- ورودی/ خروجی های اصلی 35
7-2- کارت های ورودی/خروجی اصلی در RACK های بسط یافته 39
7-3- اختصاص آدرس به کارتهای ویژهٔ ورودی/ خروجی 42
7-3-1- چگونگی تخصیص Word ها 42
7-4- ایجاد جدول ورودی/خروجی 44
7-4-1- تعریف جدول ورودی/خروجی به صورت خودکار در زمان راه اندازی PLC 44
7-4-2- تعریف جدول ورودی/خروجی توسط کاربر 44
7-5- تعیین اولین WORD از هر RACK 45
7-5-1- تنظیم اولین Word برای آدرس دهی هر Rack توسط CX-Programmer 46
7-6- اندوختن WORD ها برای تغییرات پیش بینی شده 46

8- آماده کردن PLC برای برنامه نویسی 48

8-1- برقراری ارتباط 48
8-1-1- برقراری ارتباط با PLC 48
8-1-2- برنامهٔ کنترلی 49
8-1-3- دستورالعملهای اولیه 49
کنتاکتها 49
کویل ها 49
وارد کردن کنتاکتها و کویل ها 50
آدرس دهی کنتاکتها و کویل ها 51
آدرس دهی با استفاده از نام ها 51
آدرس دهی با استفاده از آدرسهای ورودی/خروجی 51
8-2- نوشتن برنامهٔ نردبانی 51
8-2-1- ایده های اولیهٔ برای نوشتن برنامه نردبانی 51
8-2-2- ممنوعیت های موجود در برنامه نویسی 52

8-2-3- مدهای عملکرد PLC 54
مد برنامه (PROGRAM Mode) 54
مد نمایش (MONITOR Mode) 54
مد اجرا (RUN Mode) 54

9- دستورالعمل ها 55

9-1- دستور LD :LOAD 55
9-1-1- نماد در برنامهٔ نردبانی 55
9-1-2- شرح عملکرد دستور LD 55
9-2- دستور AND 55
9-3- دستور OR 56
9-4- دستور NOT 56
9-5- دستور تایمر:TIM 57
9-5-1- نماد تایمر در برنامهٔ نردبانی 57
9-5-2- عملگرهای دستور تایمر 57
9-5-3- مشخصات عملگرهای تایمر 57
9-5-4- شرح عملکرد تایمر 58
9-5-5- فلگهای دستور تایمر 58
9-5-6- نکات قابل توجه در مورد دستورالعمل تایمر 59
9-5-7- مثال از دستور تایمر 59
9-6- دستور کانتر/شمارنده:CNT 59
9-6-1- نماد کانتر در برنامه نویسی نردبانی 60
9-6-2- عملگرهای دستور شمارنده 60
9-6-3- مشخصات عملگرهای شمارنده 60
9-6-4- شرح عملکرد شمارنده 61
9-6-5- فلگهای دستور کانتر 61
9-6-6- نکات قابل توجه در مورد دستورالعمل کانتر 61
9-7- دستور KEEP : KEEP(011) 63
9-7-1- نماد دستور KEEP(011) در برنامه نویسی نردبانی 63
9-7-2- مشخصات عملگر دستور KEEP(011) 63
9-7-3- شرح عملکرد دستور KEEP(011) 63
9-7-4- نکات قابل توجه در مورد دستورالعمل KEEP(011) 65
9-7-5- مثال از دستور KEEP(011) 66
9-8- دستورهای SET و RESET : SET و RESET 66
9-8-1- نماد دستور SET و RESET در برنامه نویسی نردبانی 66
9-8-2- مشخصات عملگر دستورSET و RESET 67
9-8-3- شرح عملکرد دستور SET و RESET 67
9-8-4- نکات قابل توجه در مورد دستورالعملهای SET و RESET 68
9-9- دستورهای آشکارساز لبهٔ بالاروند/پایین رونده : DIFU(013) و DIFD(014) 68
9-9-1- نماد دستور DIFU(013) و DIFD(014) در برنامه نویسی نردبانی 68

9-9-2- مشخصات عملگر دستورهای DIFU(013) و DIFD(014)..... 68
9-9-3- شرح عملکرد دستورهای DIFU(013) وDIFD(014) 69
9-9-4- مثال از عملکرد دستورهای DIFU(013) و DIFD(014) 69
9-10- دستور مقایسه کننده : CMP(020)..... 70
9-10-1- نماد دستور CMP(020) در برنامه نویسی نردبانی..... 70
9-10-2- مشخصات عملگرهای دستور CMP(020) 70
9-10-3- شرح عملکرد دستور CMP(020)..... 71
9-10-4- وضعیت فلگهای محاسباتی 71
9-10-5- نکات قابل توجه در مورد دستورالعمل CMP(020)..... 71
9-11- دستور افزایش دهندهٔ BCD : B(594)++..... 71
9-11-1- نماد دستور B(594)++ در برنامه نویسی نردبانی 71
9-11-2- مشخصات عملگرهای دستور B(594)++..... 71
9-11-3- شرح عملکرد دستور B(594)++ 72
9-11-4- نکات قابل توجه در مورد دستورالعمل B(594)++ 72
9-11-5- مثال از عملکرد دستور B(594)++ 72
9-11-6- مثال از عملکرد دستور B(594)++@ 73
9-12- دستور کاهش دهندهٔ BCD : B(596)--..... 73
9-12-1- نماد دستور B(596)-- در برنامه نویسی نردبانی 73
9-12-2- مشخصات عملگرهای دستور B(596)-- 73
9-12-3- شرح عملکرد دستور B(596)--..... 74
9-12-4- نکات قابل توجه در مورد دستورالعمل B(596)-- 74
9-12-5- مثال از عملکرد دستور B(596)--..... 75
9-12-6- مثال از عملکرد دستور B(596)--@..... 75
9-13- دستور MOVE: MOV(021) 76
9-13-1- نماد دستور MOV(021) در برنامه نویسی نردبانی 76
9-13-2- مشخصات عملگرهای دستور MOV(021) 76
9-13-3- شرح عملکرد دستور MOV(021)..... 77
9-13-4- فلگهای دستور MOV(021) 77
9-13-5- مثال از عملکرد دستور MOV(021)..... 77

10- مثالهای کاربردی 78

10-1- طراحی تعیین الویت 78
10-2- کنترل خط بسته بندی..... 79
10-3- کنترل پر و خالی شدن مخزن 80
10-4- کنترل خودکار در انبار..... 81
10-5- اخطار زمانی 82
10-6- کنترل پارکینگ 83

11- طبقه بندی دستورالعملها با توجه به عملکرد 84

1- مقایسهٔ سیستمهای کنترل در گذشته و حال

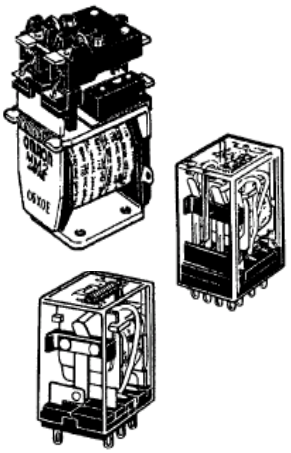
1-1- رله ها

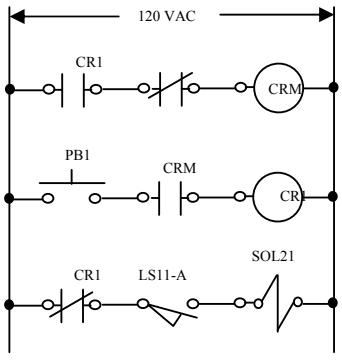
تا اواسط دههٔ 1970 بسیاری از سیستمها توسط رله هایی که در تابلوهای کنترل بزرگ قرار داشتند، کنترل می شدند. این رله ها معمولا میزان قابل توجه ای گرما تولید می کردند، همچنین مصرف انرژی زیادی داشته و با ولتاژهای سطح بالا کار می کردند.

سیستمهای کنترل رله ای برای مهندسین و کارشناسان فنی مشکلات زیادی ایجاد می کردند. اتصالات سیم بندی شده معمولا خیلی زیاد و به هزاران اتصال می رسید، این موضوع منجر به وجود آمدن مشکلات زیادی به هنگام از دست دادن یکی از اتصالات می شد. تایمرها به صورت پنوماتیکی بوده و به همین دلیل نیاز به تنظیمات دستی دوره ای داشتند که این امر باعث ایجاد مشکلاتی برای مهندسین می شد.

به دلیل اینکه رله ها یک عنصر مکانیکی هستند، روشن و خاموش شدن کویل رله ها به آهستگی صورت می گرفت و همچنین زمان لازم برای عملکرد رله بسته به نوع آن متفاوت بود. این مشکل دیگری بود که در سیستمهای رله ای وجود داشت. از لحاظ مکانیکی نیز رله ها نیاز به نگهداری دوره ای برای تمیز کردن کنتاکتها و یا تعویض کامل رله ها داشته اند. همچنین با توجه به محدودیت تعداد کنتاکتهای موجود در رله ها باید برای دستیابی به کنتاکتهای بیشتر از رله ها به صورت موازی استفاده می شده است.

ایجاد تغییرات به منظور تغییر در منطق عملکرد سیستم کنترل نیاز به جا به جایی و یا برداشتن بعضی از سیمهای سیم بندی شده داشت که این امر نیز منجر به اشتباهات زیادی می شده است. تغییرات ایجاد شده در سیم بندی نیز معمولا جایی به ثبت نمی رسید که باعث افزایش مشکلات در هنگام رفع اشکال مدار می شده است. رفع اشکال نیز با مشکلات زیادی همراه بود که شامل اندازه گیری ولتاژها، خواندن اسناد مربوط به تابلو کنترل، بیرون کشیدن سیمها از تابلو کنترل و دنبال کردن سیمها برای پیدا کردن قطعی و یا مشکلات در مسیر سیم کشی می شد. منطق کنترل نیز به شکل " منطق نردبانی رله ای (RLL) "ترسیم می شده است. که در این روش "ستون های" عمودی نشان دهندهٔ مسیر قدرت مدار منطقی و "پله های" افقی نیز نشان دهندهٔ منطق رله ای کنترل دستگاه بوده است.





منطق برنامه RLL

1-2- PLC ها

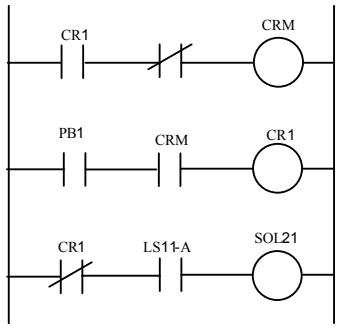
کنترل کننده های منطقی برنامه پذیر (PLC) برای رفع و یا کاهش استفاده از رله ها طراحی شده اند. PLCها باعث کاهش سایز تابلوهای کنترل و همچنین انرژی مصرفی سیستمهای کنترل شده اند. در PLC های قدیمی برای جلوگیری از سیم بندی دوبارهٔ سیستمهایی رله ای که با ولتاژ های سطح پایین کار می کردند از ولتاژ 120VAC استفاده شده است.

عناصر ورودی به یک نقطه اتصال در PLC متصل می شوند. از دست دادن اتصالات و قطعی آنها همچنان به عنوان مشکل باقی مانده است ولی با استفاده از PLC تعداد اتصالات به میزان قابل توجه ای کاهش پیدا کرده است. تایمرها و شمارنده های موجود در PLC الکترونیکی بوده و بسیار باثبات تر از نوع پنوماتیکی قدیمی می باشند. امروزه تایمرهای PLC به طرز باور نکردنی دقیق هستند به طوری که قابلیت محاسبهٔ زمان با دقت بسیار بالا را دارا می باشند.

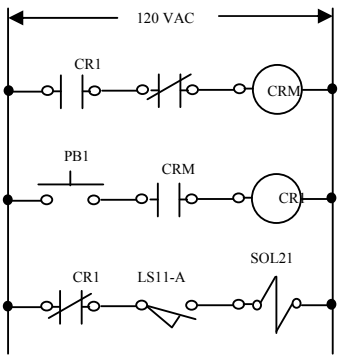
با استفاده از فناوری ترانزیستور در PLC ها بسیاری از محدودیت های مکانیکی که در سیستمهای رله ای وجود داشته است برطرف شده است. برای اتصال بسیاری از خروجی های PLC به بارهای خارجی هنوز از رله ها استفاده می شود. این رله ها در ساختار داخلی PLC دارای تعداد نامحدودی کنتاکت برای استفاده در برنامه نویسی هستند. بنابراین یک PLC می تواند جایگزین هزاران هزار رله ولی در فضایی کوچک باشد.

برنامه نویسی مجدد به جای تغییر در سیم بندی سیستم، برای تغییر منطق عملکرد سیستم کنترل استفاده می شود. رفع اشکال با استفاده ازعناصر برنامه نویسی که در عملکرد منطقی برنامه دیده می شود انجام می گیرد. این روش بسیار ساده تر از دنبال کردن سیمها و یا تست کردن کنتاکت رله ها می باشد.

بسیاری از متخصصان برق برای خواندن منطق RLL برای نصب و رفع اشکال سیستمهای کنترل رله ای دوره دیده اند. این امر باعث شده تا در زبان برنامه نویسی PLC با الهام از منطق رله ای، از همان دید برنامه نویسی با منطق رله ای استفاده شود، که نمونهٔ بارز آن زبان برنامه نویسی نردبانی می باشد.



برنامهٔ نردبانی



منطق برنامه RLL

2- مقدمه ای دربارهٔ PLC

2-1- توابع PLC

در این بخش به معرفی توابع مختلف PLC که در موارد زیر مورد استفاده قرار می گیرد، می پردازیم:

منطقی
Logic
ارتباط بین حالت ورودیها با توجه به یک ترتیب تعیین شده برای کنترل خروجی ها.
مثال: با زدن کلید و یا کشیدن زنجیر لامپ روشن خواهد شد.

زمانی
Timing
وقفهٔ زمانی که با شروع یک اتفاق معین، مانند بسته شدن یک کنتاکت، برای ایجاد محدودیت زمانی معین به کار می رود.
مثال: موتور پمپ بعد از فشردن کلید Start ،به مدت 2 دقیقه کار می کند.

شمارشی
Counting
محاسبه کل و مجموع یک رویداد تکراری برای محدود کردن آن به تعداد از پیش تعیین شده.
مثال: شمارش تعداد قرصها در هر بطری، شمارش محصولات پذیرفته نشده.

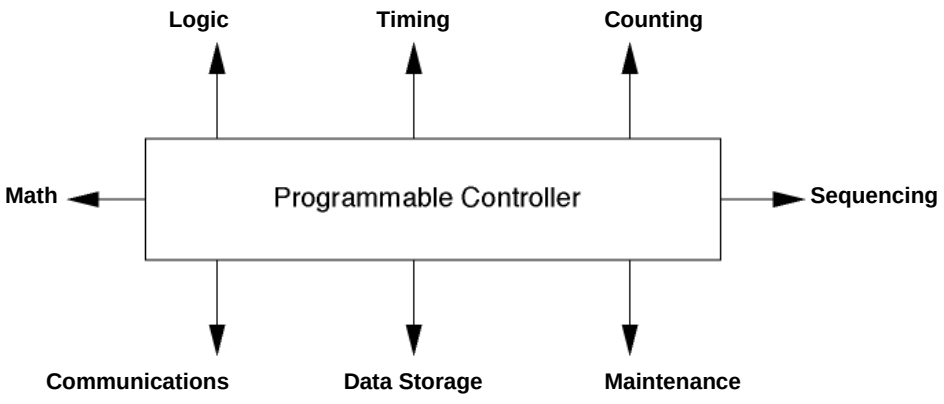
ترتیبی
Sequencing
تعیین ترتیب اجراء رویدادها در یک فرآیند و یا یک دستگاه.
مثال: یک میز گردان باید قطعه ای را بارکند، سوراخ کند، سوراخ ایجاد شده را مته زده، لبه های قطعه را پلیسه گیری کند و قطعه را تحویل دهد.

ریاضی
Math
برای محاسبهٔ اندازه، مسافت، سرعت، درصد و غیره به کار می رود.
مثال: برای به دست آوردن بازدهٔ دستگاه در مبنای درصد، باید تعداد محصولات بازگشتی را به کل محصولات تقسیم کرده و در 100 ضرب کرد.

ذخیرهٔ داده ها
Data Storage
ثبت داده ها در یک دورهٔ زمانی مشخص برای مرور و بازبینی داده های گذشته.
مثال: آزمایش داده ها برای فشار و دمای یک فرآیند.

تعمیر و نگهداری
Maintenance
امکانات عیب یابی سیستم، همچنین امکانات عیب یابی یک فرآیند یا دستگاه و زمانبندی برای تعریف خطا.
مثال: تعیین تعداد آلارمهای دستگاه و یا خطاهای سخت افزاری سیستم، اگر رفت و برگشت سیلندر بیشتر از 5 ثانیه به طول بینجامد، این نشان دهندهٔ وجود نشتی در سیلندر خواهد بود.

ارتباطات
Communication
انتقال داده ها و یا دریافت داده بین PLC و عناصر موازی متصل به آن.
مثال: جمع آوری اطلاعات از یک ترازوی وزنی و یا دستگاه خوانندهٔ کد میله ای.



2-2- بلوک دیاگرام ساختمان PLC

به دست آوردن داده ها برای PLC

دادهٔ دیجیتال سیگنالهای روشن/خاموش دیجیتال و یا گسسته.

مثال: شستی ها (Push Buttons)، لیمیت سوئیچ ها (Limit Switches)، سنسورهای مجاورتی (Proximity Switches)، سنسورهای نوری

دادهٔ آنالوگ سیگنالهای متغیر نشان دهندهٔ همهٔ حالتهای بین دو مقدار می نیمم و ماکزیمم.

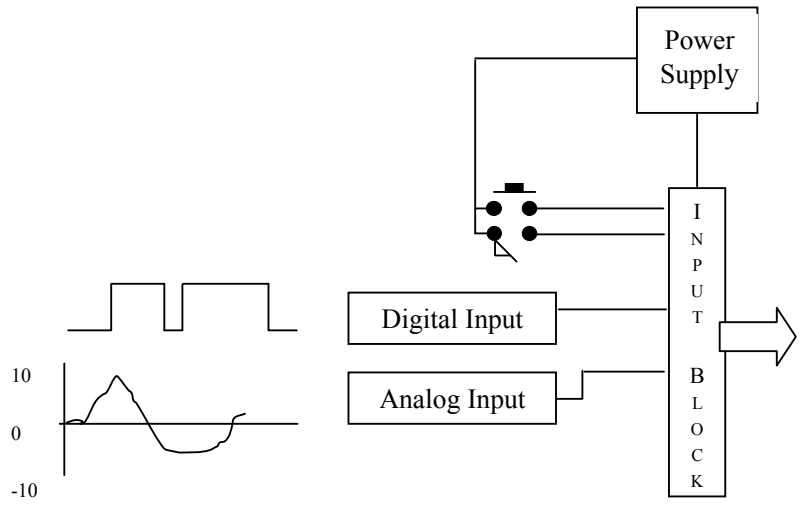
مثال: سیگنال 4 تا 20 میلی آمپر از یک سنسور سنجش وزن، سیگنال 0 تا 10 ولت نشان دهندهٔ فشار و یا سیگنال نشان دهندهٔ دما از یک ترموکوپل.

منبع تغذیهٔ خارجی تأمین کنندهٔ تغذیهٔ مدارات ورودی.

مثال120VAC، 220VAC، 24VDC، 12VDC

بخش ورودی اتصالات فیزیکی بین عناصر ورودی و PLC.

مثال: شستی ها به عنوان عناصر ورودی به ترمینالهای ورودی CS1W بسته می شود.



Inputs

پردازش اطلاعات

زمانیکه PLC ورودیها را بررسی می کند، قبل از تجدید کردن (Refresh) حالت خروجی ها به پردازش اطلاعات می پردازد. برنامهٔ ذخیره شده در PLC شامل دستورالعملهایی است که چگونگی عملکرد دستگاه را تعیین می کند.

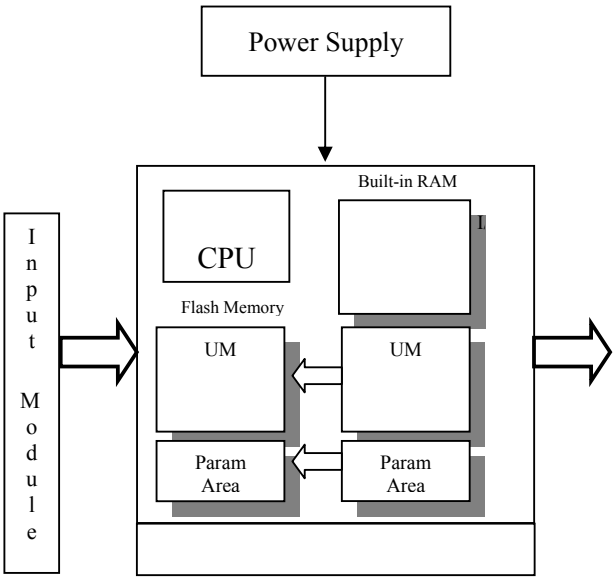
CPU واحد پردازش مرکزی که از یک یا چند ریزپردازنده تشکیل شده است که وظیفهٔ نظارت بر همه فعالیتهای PLC را برعهده دارد. از جمله وظایف اولیه CPU بررسی ورودیها، اجرای برنامه و تجدید حالت خروجی ها می باشد. وظایف دیگری از جمله بررسی سلامت دستگاه از لحاظ سخت افزاری، برقراری ارتباطات سریال با عناصر خارجی که به صورت سریال به PLC متصلند نیز بر عهدهٔ واحد CPU می باشد.

حافظه این قسمت بخش ذخیره سازی PLC است که برای ذخیره کردن برنامهٔ کاربر و داده ها به کار می رود.

منبع تغذیه منبع تغذیه تأمین کنندهٔ تغذیهٔ CPU و حافظه می باشد. که معمولا ولتاژ 5VDC مدارات منطقی و 24VDC تغذیه مدارات الکترونیکی کارت های ورودی و خروجی را تأمین می کند.

حافظهٔ ورودی تغییرات حالتهای عناصر ورودی متصل به PLC را ثبت می نماید.

حافظهٔ خروجی مقادیر پردازش شده در حافظه مربوط به خروجیها ثبت می شود.



فرستادن داده ها به خروجی

بخشهای خروجی وظیفهٔ ارسال داده های دیجیتال و آنالوگ را به دنیای خارج PLC برای روشن کردن موتورها، روشن کردن لامپها، فرمان دادن به سولنوئیدها و ... برعهده دارند.

دادهٔ دیجیتال سیگنالهای روشن/خاموش دیجیتال و یا گسسته

مثال: سولنوئیدها، رله های واسط، موتورها، لامپها

دادهٔ آنالوگ سیگنالهای متغیر نشان دهندهٔ همهٔ حالتهای بین دو مقدار می نیمم و ماکزیمم.

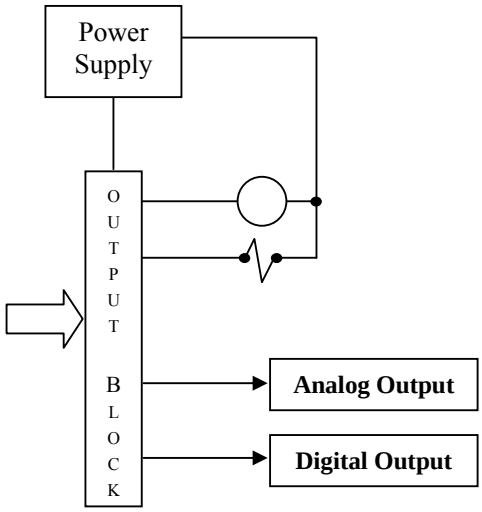
مثال: سیگنال 4 تا 20 میلی آمپر برای کنترل دما، سیگنال 0 تا 10 ولت برای کنترل سرعت.

منبع تغذیهٔ خارجی تأمین کنندهٔ تغذیهٔ مدارات خروجی.

مثال 120VAC، 220VAC، 24VDC، 12VDC

بخش خروجی اتصالات فیزیکی بین PLC و عناصر خروجی.

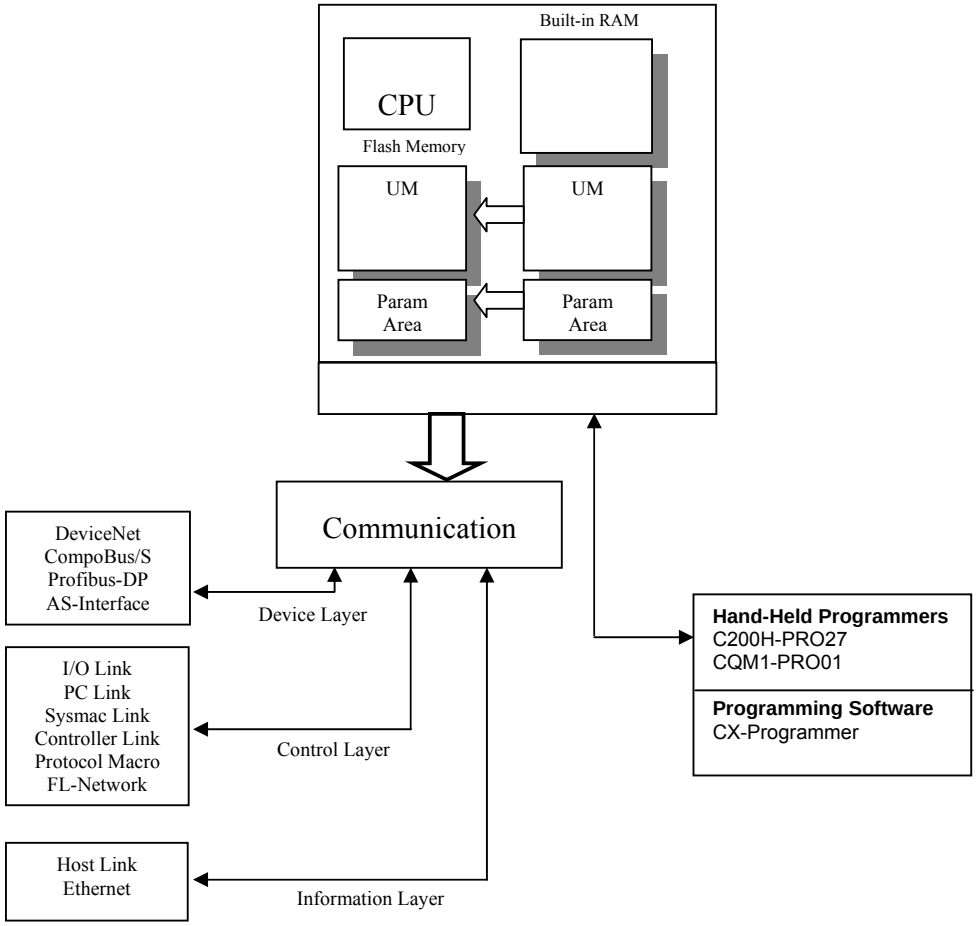
مثال: ترمینالها خروجی CS1W که به یک سولنوئید بسته می شوند.



ارتباط با دنیای خارج:

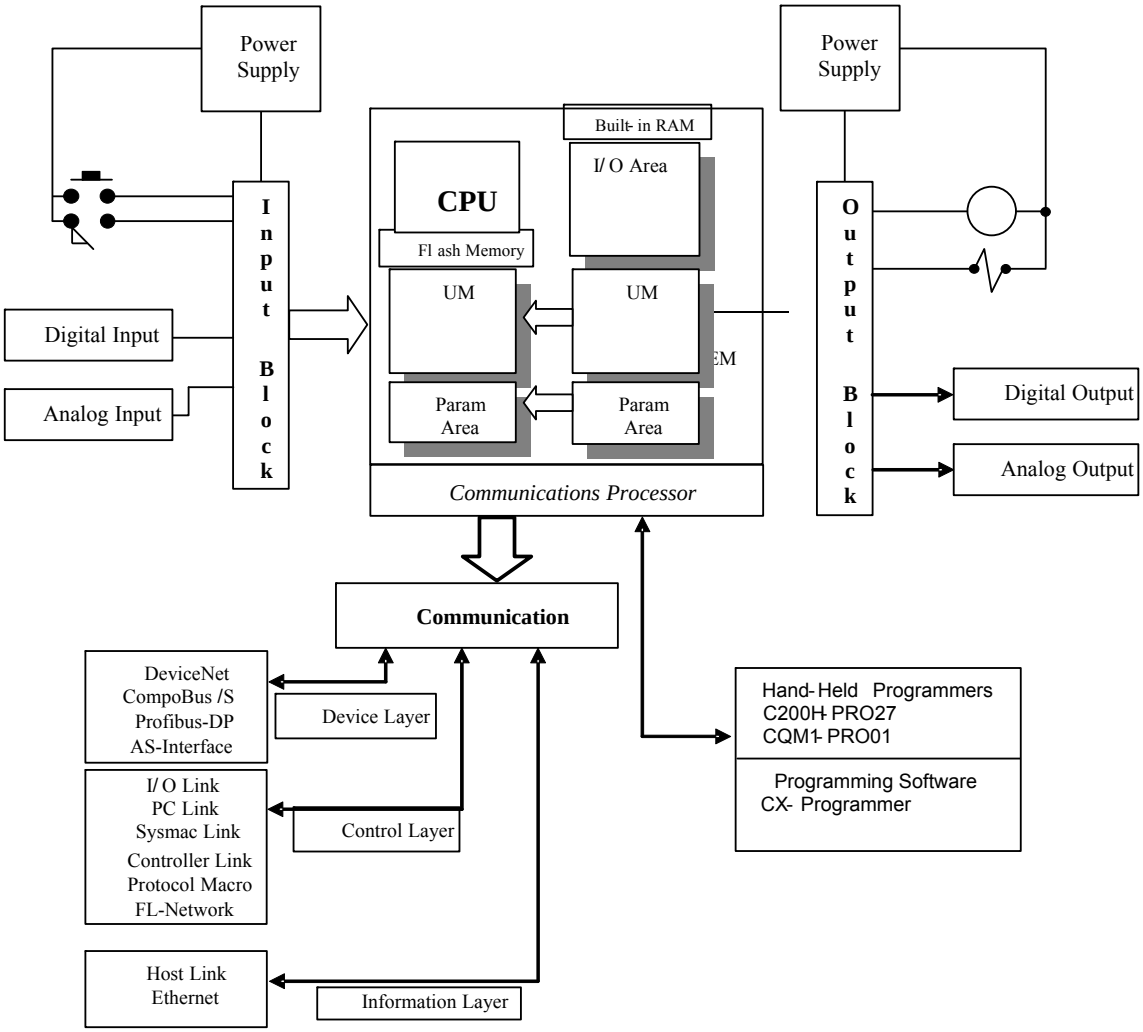
برنامه ریزی یک PLC به ارتباط با یک رایانه و یا یک عنصر برنامه نویسی نیاز دارد. برقراری ارتباط با المانهای نصب شده در فواصل دور، نمایشگرها، و شبکه ها برای تبادل اطلاعات بین PLC ها، توسط پردازشگرهای ارتباطات صورت می گیرد.

تجهیزات دوردست متصل در فیلد **Remote Field Devices**	تجهیزات DeviceNet ، CompoBus/S، Profibus-DP، AS-Interface
واسط بین دستگاه و اپراتور **Operator Interface**	نمایشگرهای NT و NS، CX-Supervisor
شبکه های بین PLC ها **Peer-To-Peer Network**	PC Link، Sysmac Link، Controller Link، Ethernet، FL-Network
تجهیزات جانبی **Peripherals**	تجهیزات برنامه نویسی دستی، نرم افزار CX-Programmer


Built-in RAM
CPU
Flash Memory
UM
UM
Param Area
Param Area
Communication
DeviceNet
CompoBus/S
Profibus-DP
AS-Interface
Device Layer
I/O Link
PC Link
Sysmac Link
Controller Link
Protocol Macro
FL-Network
Control Layer
Host Link
Ethernet
Information Layer
Hand-Held Programmers
C200H-PRO27
CQM1-PRO01
Programming Software
CX-Programmer

یکی کردن همه بخش ها :

با در نظر گرفتن تمام بخشهای نام برده شده و یکی کردن آنها می توان اجزاء ضروری تشکیل دهندهٔ یک PLC را چنین تعریف کرد:



نکتهٔ قابل توجه دیگر- زمان دوره (Cycle Time) :

در PLC ها اصطلاحی به نام زمان دوره (Cycle Time) وجود دارد. زمان دوره، مدت زمانی است که طول می کشد تا PLC تمام وظایف خود را یکبار به انجام برساند، PLC این سیکل را بارها و بارها تا ایست کار خود و یا قطعی برق تکرار می کند. PLC ها برای عملکرد خود منتظر ایجاد تغییر در حالت یک ورودی نمی شوند بلکه پردازش خود را بارها و بارها انجام داده و تغییرات ایجاد شده در حالت ورودیها را در یک دوره در این پردازش لحاظ می کنند.

به طور ساده می توان گفت که زمان دوره زمانی است که در آن PLC موارد زیر را انجام می دهد:

1. شناسایی اولیه
2. سرپرستی پردازش
3. اجرا برنامه
4. دوباره سازی و تجدید ورودی/خروجی ها
5. سرویس دهی به تجهیزات جانبی

شناسایی اولیه که درهنگام وصل تغذیهٔ PLC انجام می شود. شناسایی سخت افزاری سیستم، پاک کردن جدول ورودی/خروجی، بررسی حافظهٔ کاربر، انتقال خودکار برنامه از کارت حافظه به حافظهٔ PLC از جمله مواردی است که هنگام شناسایی اولیه انجام می گیرد.

سرپرستی پردازش شامل خواندن دیپ سوئیچ ها، بررسی باتری، بررسی مسیرهای ورودی و خروجی و حافظهٔ مربوط به برنامه می شود.

اجرا برنامه شامل اجرا برنامهٔ کاربر می باشد. برنامه که نوشته می شود شامل ترتیب، منطق، محاسبهٔ زمان، شمارش و ... می باشد.

دوباره سازی و تجدید ورودی/خروجی ها به معنی تجدید حالت ورودی و خروجی ها است. حالت جدید خروجی ها در بخش خروجی جدول ورودی/خروجی قرار می گیرد. حالت ورودی های متصل به بخش ورودی در قسمت ورودی جدول ورودی/خروجی برای استفاده در سیکل بعدی ذخیره می شود.

سرویس دهی به تجهیزات جانبی خواسته های تجهیزاتی که به صورت سریال به PLC متصل شده اند را برآورده می سازد.

3- آشنایی با PLC های ساخت شرکت OMRON

کنترل کننده های منطقی برنامه پذیر (PLC) ساخت شرکت OMRON بر حسب قابلیتها و کارآیی به سه دستهٔ اصلی تقسیم می شوند:

1. Compact PLC Series
2. Modular PLC Series
3. Rack PLC Series

3-1- PLC های سری فشرده (Compact PLC Series) - قابلیت پذیرش تا 320 ورودی/خروجی

PLC های سری Compact با منبع تغذیه، ورودی و خروجی یکپارچه و مجتمع، از کنترل کننده هایی همچون کنترل کنندهٔ غیر قابل بسط CPM1A با 10 ورودی/خروجی تا کنترل کنندهٔ قدرتمندی با قابلیتهای بسیار بالای CP1H تشکیل شده است. این کنترل کننده ها با قابلیت پذیرش تعداد زیاد دستورالعملهای برنامه نویسی یکی از انتخابهای ایده آل برای کنترل دستگاههای کوچک می باشند. وجود شمارنده های با سرعت بالا (High Speed Counters)، خروجی های پالس (Pulse Output)، کنترل کننده های موقعیت و سرعت (Position and Speed Control)، این کنترل کننده ها را به کنترل کننده های با توانایی بالای کاربردی برای انجام کنترل سیستمها تبدیل کرده است.

PLC های سری فشرده (Compact PLC Series) حداکثر I/O 320	CPM1A	CPM2A	CPM2C	CP1H
حداکثر تعداد ورودی/خروجی دیجیتال	10-100	80-120	106-192	320
زمان اجرا (دستورهای بیتی)	0.72-1.72µs	0.26-064µs	0.26-0.64µs	0.1µs
حافظهٔ برنامه	2 kWords	4 kWords	4 kWords	20 kSteps
حافظهٔ داده ها	1 kWords	2 kWords	2 kWords	32 kWords
توابع ویژه	ورودی کانتر 5kHZ خروجی پالس 2kHz	ورودی کانتر 20kHZ دو خروجی پالس 10kHz	ورودی کانتر 20kHZ دو خروجی پالس 10kHz	CPU های با I/O پالس و آنالوگ. قابل بسط با I/O های سری CPM1A و یا کارتهای ویژهٔ سری CJ1
ارتباطات	Serial, DeviceNet, PROFIBUS-DP, CompoBus/S	Serial, DeviceNet, CompoBus/S, DeviceNet-to-CompoBus/S gateway function	Serial, DeviceNet, PROFIBUS-DP, CompoBus/S	Ethernet, Controller Link, Serial, DeviceNet, PROFIBUS-DP, CAN, CompoBus/S

3-2- PLC های سری ماژولار (Modular PLC Series)- قابلیت پذیرش تا 2500 ورودی/خروجی

شرکت OMRON با ساخت PLC های سری CJ1 در اواخر دهۀ 1990 پیشگام در ساخت PLC ها با Back-Plane مجازی بوده است. با وجود دامنۀ وسیعی از CPU های قابل تبدیل به یکدیگر، کارتهای شبکه و کارتهای Motion، این PLC یکی از کامل ترین PLC های ماژولار به حساب می آید. علاوه بر قابلیت اجرای کنترل به بهترین شیوه، این PLC قابلیت قرارگرفتن و برقراری ارتباط با شبکه های هوشمندی همچون Ethernet، DeviceNet، Profibus-DP و همچنین CAN را دارا می باشد.

PLC های سری ماژولار (Modullar PLC Series) حداکثر 2500 I/O	CQM1H	CJ1M	CJ1G/H
حداکثر تعداد ورودی/خروجی دیجیتال	256-512	160-640	960-2560
زمان اجرا (دستورهای بیتی)	0.375µs	0.1µs	0.04/0.02µs (G-type/H-type)
حافظۀ برنامه	3-15 kWords	10-20 kSteps	10-250 kSteps
حافظۀ داده ها	3-12 kWords	32 kWords	64-448 kWords
توابع ویژه	کانترهای سرعت بالا و ورودی/خروجی های آنالوگ به عنوان بردهای موجود قابل انتخاب در واحد CPU	CPU با I/O های پالسی روی آن و یا پورت Ethernet، کارتهای اختصاصی برای کنترل موقعیت، کنترل دما، ارتباطات تعریف شده توسط کاربر	CPU های کنترل حلقه ای، کارتهای اختصاصی برای کنترل موقعیت، کنترل دما، ارتباطات تعریف شده توسط کاربر
ارتباطات	Controller Link, Serial, DeviceNet, PROFIBUS-DP, CompoBus/S, AS-Interface	Ethernet, Controller Link, Serial, DeviceNet, PROFIBUS-DP, CAN, CompoBus/S,	Ethernet, Controller Link, Serial, DeviceNet, PROFIBUS-DP, CAN, CompoBus/S,

3-3- PLC های سری Rack (Rack PLC Series)- قابلیت پذیرش تا 5000 ورودی/خروجی

سازگاری این PLC با PLC های سری C200H ساخت شرکت OMRON دامنۀ وسیعی از کارتهای ورودی/خروجی و کارتهای خاص را برای انتخاب ایجاد کرده است. دقت و سرعت بالا در جمع آوری اطلاعات آنالوگ از خصوصیات بارز این PLC به حساب می آید.

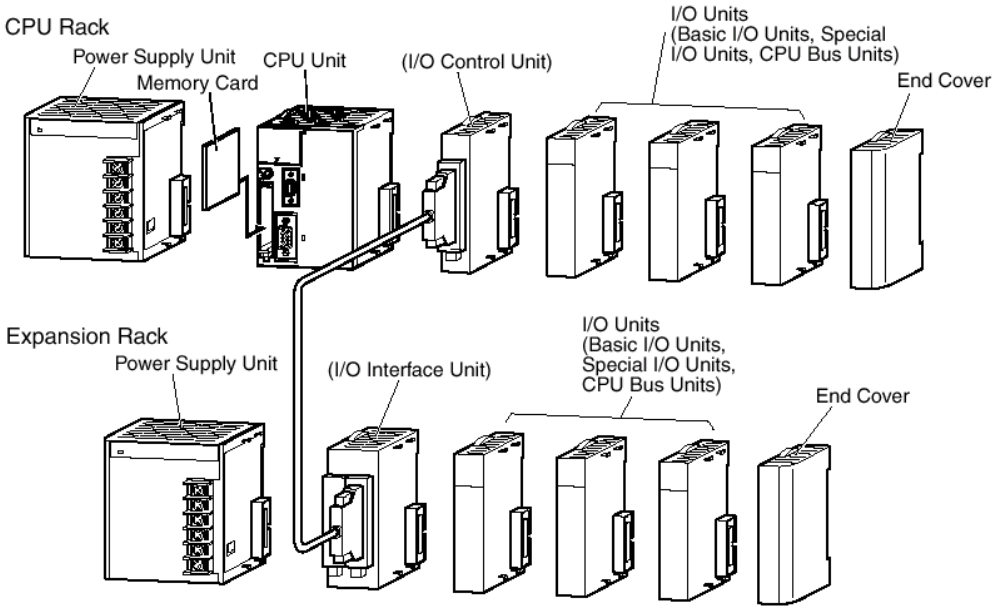
وجود کنترل کنندۀ حرکت (Motion Control) و کارتهای قابل برنامه ریزی، PLC های CS1 را به کنترل کننده های چند منظوره با کاربرد بسیار وسیع تبدیل کرده است. همچنین CS1 با داشتن CPU های دوتایی (Duplex) و همچنین کنترل حلقه ای (Loop Control) قابلیت کنترل فرآیندها (Process Automation) را نیز دارا می باشد.

CS1D	CS1G/H	PLC های سری Rack (Modullar PLC Series) حداکثر 5000 I/O
960-5120	960-5120	حداکثر تعداد ورودی/خروجی دیجیتال
0.04/0.02µs	0.04/0.02µs (G-type/H-type)	زمان اجرا (دستورهای بیتی)
10-250 kSteps	10-250 kSteps	حافظهٔ برنامه
64-448 kWords	64-448 kWords	حافظهٔ داده ها
CPU های کنترل Dual-Redondancy، حلقه ای، Hot-Swapping	CPU های کنترل حلقه ای، کارتهای اختصاصی برای کنترل موقعیت، کنترل دما، ارتباطات تعریف شده توسط کاربر	توابع ویژه
Ethernet, Controller Link, Serial, DeviceNet, PROFIBUS-DP, CAN, CompoBus/S,	Controller Link, Serial, DeviceNet, PROFIBUS-DP, CompoBus/S, AS-Interface	ارتباطات

4- سخت افزار PLC

4-1- سری CJ

4-1-1- نمای بسط داده شده



4-1-2- واحد CPU

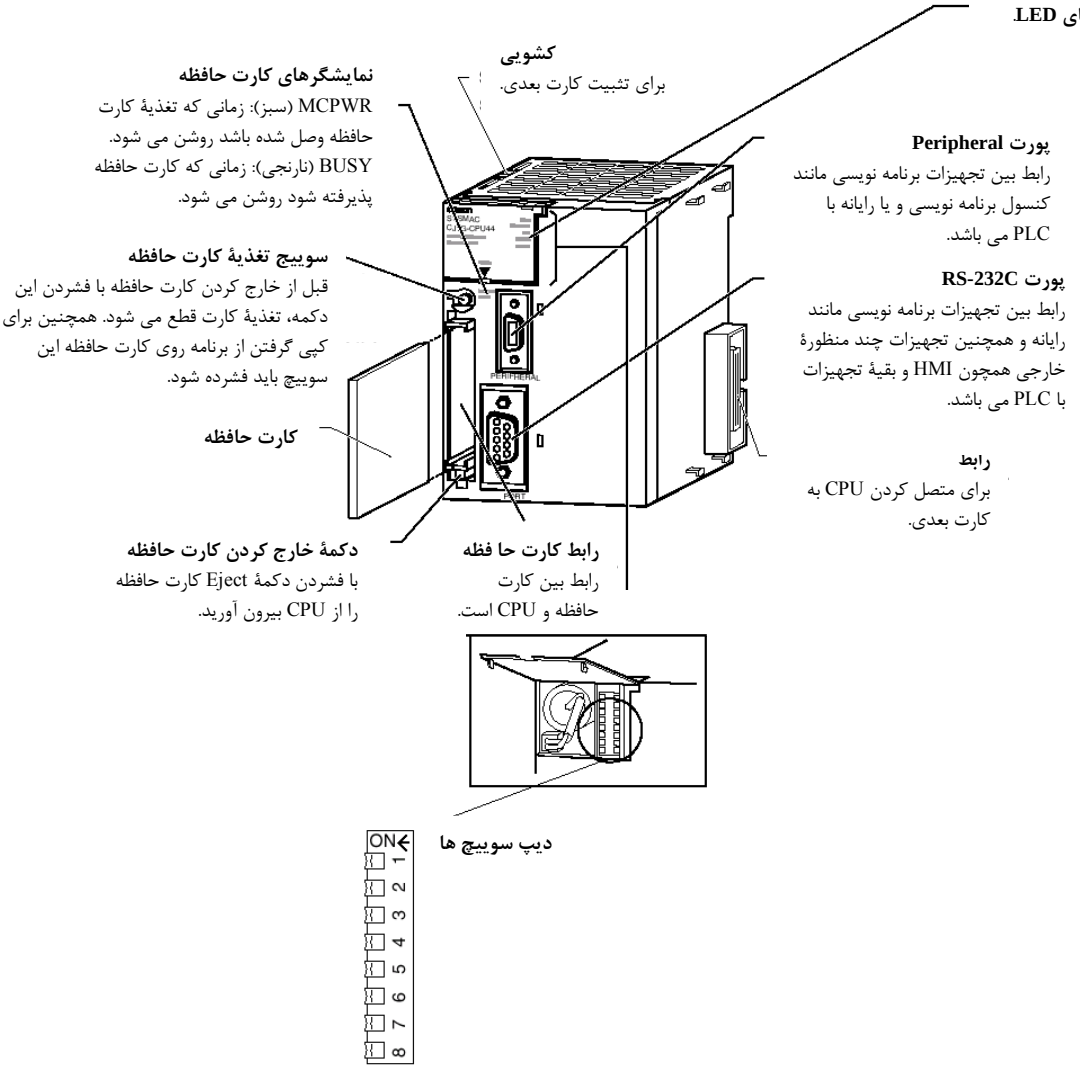
مدل CJ1، واحد CPU

وزن	مدل	زمان پردازش دستور LD	حافظه داده ها (DM + EM)	ظرفیت برنامه نویسی	Rack های بسط یافته	تعداد I/O
200 g max.	CJ1G-CPU45	0.08 µs	128 Kwords	60 Ksteps	3 max.	1,280
	CJ1G-CPU44		64 Kwords	30 Ksteps		

مدل CJ1-H، واحد CPU

وزن	مدل	زمان پردازش دستور LD	حافظه داده ها (DM + EM)	ظرفیت برنامه نویسی	Rack های بسط یافته	تعداد I/O
200 g max.	CJ1H-CPU66H	0.02 µs	256 Kwords	120 Ksteps	3 max.	2,560
	CJ1H-CPU65H		128 Kwords	60 Ksteps		
190 g max.	CJ1G-CPU45H	0.04 µs	128 Kwords	60 Ksteps		
	CJ1G-CPU44H		64 Kwords	30 Ksteps	3 max.	1,280
	CJ1G-CPU43H		64 Kwords	20 Ksteps	2 max.	960
	CJ1G-CPU42H		64 Kwords	10 Ksteps		

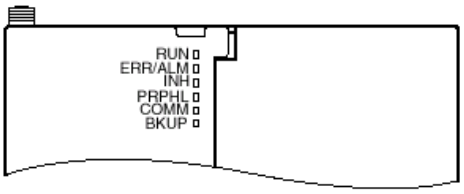
4-1-3- جزییات واحد CPU



نکته: همواره زمانی که از پورتهای روی CPU استفاده نمی شود، برای جلوگیری از نشستن گرد و غبار روی پورت باید از پوششی که برای رابط های Peripheral و RS-232C در نظر گرفته شده است استفاده شود.

4-1-4- نمایشگرهای LED

جدول زیر نمایشگرهای LED موجود روی CPU را شرح می دهد.



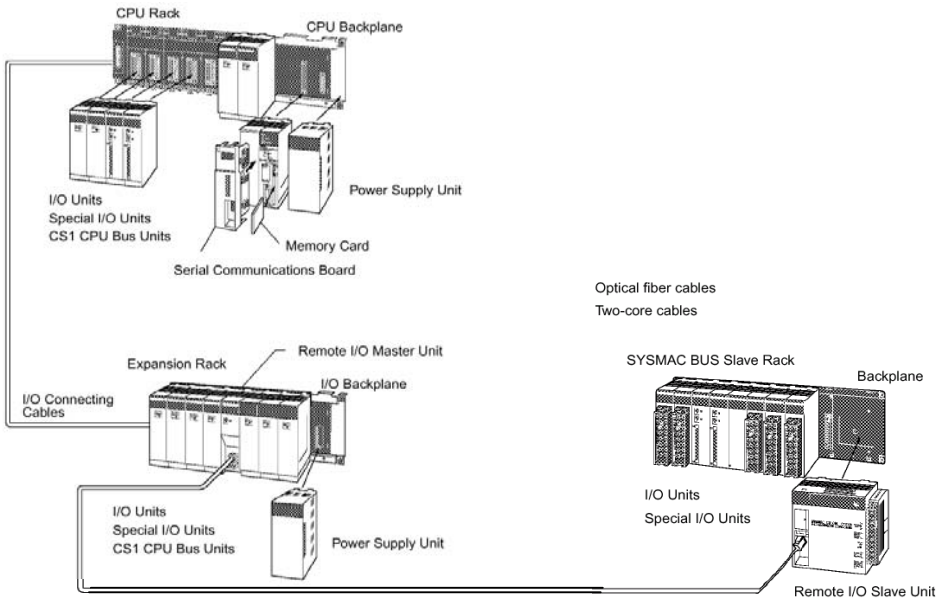
نمایشگر	رنگ	حالت	مفهوم
RUN	سبز	روشن	PLC در حال عملکرد معمول خود در مد Run و یا Monitor می باشد.
		چشمک زن	خطایی در بارگیری سیستم ویا تنظیمات دیپ سوییچ ها به وجود آمده است.
		خاموش	PLC در مد Program قرار دارد ویا وجود خطایی جدی باعث ایست کار PLC شده است و یا در حال دریافت اطلاعات از رایانه می باشد.
ERR/ALM	قرمز	روشن	در PLC خطایی جدی و یا خطای سخت افزاری روی داده است. در این حالت CPU از کار می ایستد و تمام خروجی ها قطع خواهد شد.
		چشمک زن	خطای غیرجدی رخ داده است و CPU به عملکرد خود ادامه می دهد.
		خاموش	CPU در حال عملکرد معمول است.
INH	نارنجی	روشن	بیت مربوط به قطع کردن خروجیها (A500.15) روشن شده است. خروجیهای مربوط به همهٔ بخشها قطع می باشد.
		خاموش	بیت مربوط به قطع کردن خروجی ها خاموش شده است.
PRPHL	نارنجی	چشمک زن	CPU در حال تبادل اطلاعات (ارسال و دریافت) توسط پورت Peripheral است.
		خاموش	CPU در حال تبادل اطلاعات توسط پورت Peripheral نمی باشد.
COMM	نارنجی	چشمک زن	CPU در حال تبادل اطلاعات (ارسال و دریافت) توسط پورت RS-232C است.
		خاموش	CPU در حال تبادل اطلاعات توسط پورت RS-232C نمی باشد.
BKUP	نارنجی	روشن	برنامهٔ کاربر و همچنین پارامترهای CPU به Flash Memory منتقل می شوند و یا داده ها از Flash Memory به CPU منتقل می شود. **نکته** : در این حالت تغذیهٔ PLC را قطع نکنید.
		خاموش	داده ها روی Flash Memory نوشته نمی شوند.

4-1-5- تعریف دیپ سوئیچ ها

شماره	کارآیی	تنظیمات		توضیحات
1	حفاظت در برابر نوشتن روی حافظهٔ کاربر (UM)	روشن	محافظت در برابر نوشتن	زمانی که این سوییچ روشن باشد حافظهٔ کاربر در برابر نوشتن حفاظت می شود. این سوییچ را برای جلوگیری از تغییرات تصادفی برنامه روشن کنید.
		خاموش	قابل نوشتن و خواندن	
2	انتقال خودکار برنامه در زمان راه اندازی PLC	روشن	بله	اگر این سوییچ روشن باشد، برنامه و همچنین تنظیمات PLC به صورت خودکار در زمان راه اندازی PLC از روی کارت حافظه به CPU منتقل می شود. تنها با قراردادن یک کارت حافظه که شامل برنامه و تنظیمات باشد می توان PLC را به طور کامل برنامه ریزی کرد. با استفاده از این روش می توان تغییرات در برنامه و تنظیمات را با سرعت و سهولت به انجام رساند. **نکته:** زمانی که دیپ سوییچ شمارهٔ 7 روشن و شمارهٔ 8 خاموش باشد الویت با Backup گرفتن برنامه است حتی اگر سوییچ شمارهٔ 2 روشن باشد.
		خاموش	خیر	
3	بدون استفاده	---	---	---
4	پارامترهای ارتباط با پورت Peripheral	روشن	استفاده از پارامترهای موجود در تنظیمات PLC	**نکته:** این سوییچ را در صورتی که پورت Peripheral به کنسول برنامه نویسی و یا نرم افزار CX-Programmer متصل است در حالت خاموش قرار دهید. **نکته:** اگر از این پورت برای برقراری ارتباط با تجهیزاتی غیر از کنسول برنامه نویسی و یا نرم افزار CX-Programmer استفاده شود باید این سوییچ در حالت روشن قرار گیرد.
		خاموش (پیش فرض)	یافتن خودکار تجهیزات برنامه نویسی.	
5	پارامترهای ارتباط با پورت RS-232C	روشن	یافتن خودکار تجهیزات برنامه نویسی.	**نکته:** اگر از پورت RS-232C برای برقراری ارتباط با تجهیزاتی غیر از نرم افزار CX-Programmer همچون سیستمهای مونیتورینگ استفاده شود باید این سوییچ در حالت خاموش قرار گیرد. **نکته:** این سوییچ را در صورتی که این پورت به نرم افزار CX-Programmer متصل است در حالت روشن قرار دهید.
		خاموش (پیش فرض)	استفاده از پارامترهای موجود در تنظیمات PLC	
6	تعریف و انتخاب توسط کاربر	روشن	بیت A395.12 روشن	خاموش و یا روشن بودن این سوییچ بر روی بیت A395.12 از حافظه تأثیر می گذارد، این بیت را می توان در برنامه نویسی بدون استفاده از یک ورودی خارجی به کار برد.
		خاموش (پیش فرض)	بیت A395.12 خاموش	
7	تنظیمات مربوط به Backup آسان	روشن	نوشتن اطلاعات از CPU برروی کارت حافظه	برای این امر باید دکمهٔ مربوط به تغذیهٔ کارت حافظه را برای سه ثانیه فشرده و نگه داشت.
			برگرداندن اطلاعات از کارت حافظه بر روی CPU	برای خواندن از کارت حافظه به CPU باید این سوییچ را روشن و تغذیهٔ PLC را وصل کرد. این عمل به انتقال از کارت حافظه حتی در صورت روشن بودن سوییچ شمارهٔ 2 الویت دارد.
		خاموش (پیش فرض)	بازبینی محتوای کارت حافظه	برای این امر باید دکمهٔ مربوط به تغذیهٔ کارت حافظه را برای سه ثانیه فشرده نگاه داشت.
8	بدون استفاده	خاموش (پیش فرض)	همیشه خاموش.	---

4-2- سری CS

4-2-1- نمای بسط داده شده



4-2-2- واحد CPU

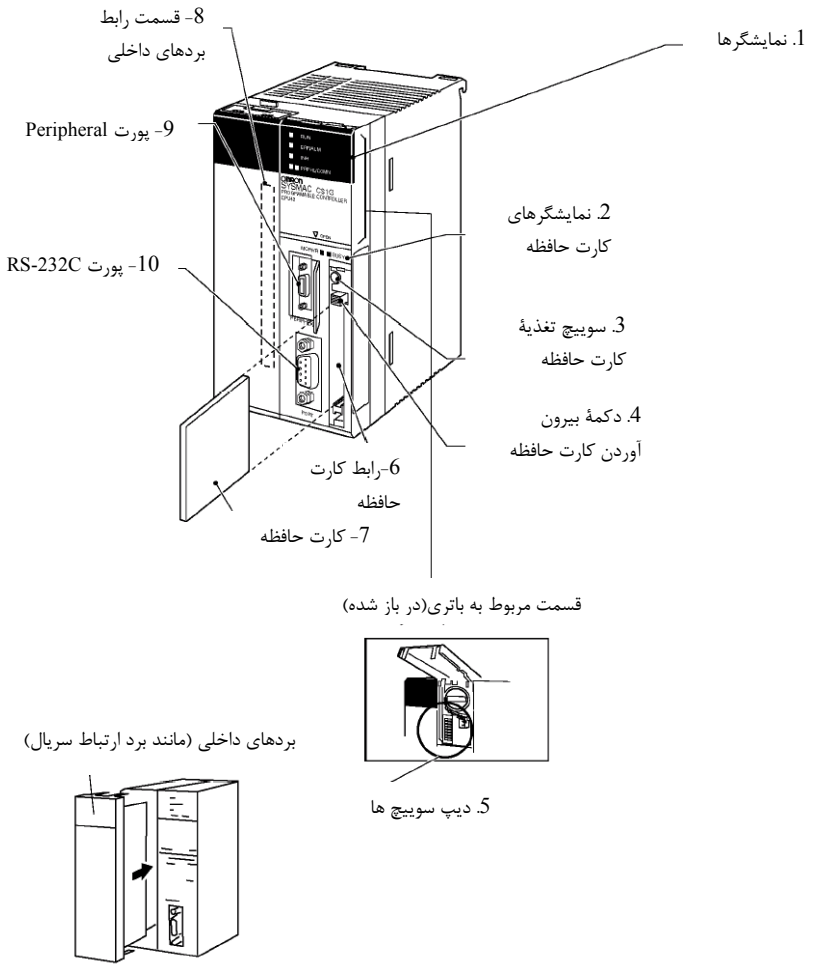
مدل CS1، واحد CPU

وزن	مدل	زمان پردازش دستور LD	حافظه داده ها (DM + EM)		ظرفیت برنامه نویسی	Rack های بسط یافته	تعداد I/O
400 g max.	CS1H-CPU67-EV1	0.04 µs	448K words	13	250K steps	7 max.	5,120
	CS1H-CPU66-EV1		256K words	8	120K steps		
	CS1H-CPU65-EV1		128K words	4	60K steps		
	CS1H-CPU64-EV1		64K words	2	30K steps		
	CS1H-CPU63-EV1		32K words	1	20K steps		
	CS1G-CPU45-EV1	0.08 µs	128K words	4	60K steps		
	CS1G-CPU44-EV1		64K words	2	30K steps	3 max.	1,280
	CS1G-CPU43-EV1		32K words	1	20K steps	2 max.	960
	CS1G-CPU42-EV1		32K words	1	10K steps		

مدل CS1-H، واحد CPU

وزن	مدل	زمان پردازش دستور LD	حافظه داده ها (DM + EM)		ظرفیت برنامه نویسی	Rack های بسط یافته	تعداد I/O
350 g max.	CS1H-CPU67H	0.02 µs	448K words	13	250K steps	7 max.	5,120
	CS1H-CPU66H		256K words	8	120K steps		
	CS1H-CPU65H		128K words	4	60K steps		
	CS1H-CPU64H		64K words	2	30K steps		
	CS1H-CPU63H		64K words	2	20K steps		
	CS1G-CPU45H	0.04 µs	128K words	4	60K steps		
	CS1G-CPU44H		64K words	2	30K steps	3 max.	1,280
	CS1G-CPU43H		64K words	2	20K steps	2 max.	960
	CS1G-CPU42H		64K words	2	10K steps		

4-2-3- جزییات واحد CPU



نکته: لازم به ذکر است که نمایشگرهای LED در PLC های سری CS1 همانند سری CJ1 می باشد.
نکته: همچنین دیپ سوییچ ها در PLC های سری CS1 مانند سری CJ1 است. به جز دیپ سوییچ شماره 3 که برای تنظیمات زبان کنسول برنامه نویسی مورد استفاده قرار می گیرد.

4-3- کارت های ورودی/خروجی

4-3-1- اطلاعات کلی کارتهای ورودی/خروجی

برای شناسایی کارتهای ورودی/خروجی نصب شده روی PLC از کدهایی که به صورت رنگی روی آنها نوشته شده است می توان استفاده کرد. این مسئله در مورد انواع PLC های OMRON صادق می باشد.

1. کارتهای ورودی

کارتهای ورودی برای پذیرفتن انواع مختلف عناصر ورودی که در صنعت استفاده می شود مانند سنسورهای نوری، لیمیت سوییچ ها و غیره طراحی شده اند، مشخصات ظاهری این کارتها در زیر آمده است:

- دارای برچسبی به رنگ قرمز و یا نارنجی می باشند.
- کد قطعه شامل حروف ID، IA، IM می باشد.
  - ID: ورودی DC (NPN و PNP)
  - IA: ورودی AC
  - IM: ورودی مختلط DC و AC. دارای ورودی های 12-48 VAC/VDC.

2. کارتهای خروجی

کارتهای خروجی از سه نوع مختلف تشکیل شده است:

رله ای:

رله یک تجهیز مکانیکی است که قابلیت سوییچ ولتاژ DC و AC را دارا می باشد. از آنجایی که رله ها یک عنصر مکانیکی هستند تعداد دفعات سوییچینگ آنها محدود می باشد، به عنوان مثال 500,000 بار.

ترانزیستوری:

ترانزیستورها عناصر حالت جامد هستند که قابلیت سوییچینگ 5-24VDC را دارا می باشند. ترانزیستورها قابلیت سوییچ زدن با سرعت بالا و به تعداد دفعات نامحدود را دارند. نرخ جریان این خروجیها به نوع ترانزیستور آنها بستگی دارد ولی یک ترانزیستور می تواند NPN و یا PNP باشد.

ترایاکی:

ترایاکها عناصر حالت جامد هستند که قابلیت سوییچینگ ولتاژ AC بین 110-240VAC را دارا می باشد. ترایاک ها قابلیت سوییچ زدن با سرعت بالا و به تعداد دفعات نامحدود را دارند. ترایاکها دارای جریان نشتی بالایی هستند که این جریان نشتی برای عملکرد یک عنصر خروجی در فیلد همچون سولنوئید ها کافی می باشد.

مشخصات ظاهری کارتهای خروجی در زیر آمده است:

- دارای برچسبی به رنگ زرد می باشند.
- کد قطعه شامل حروف OD، OC، OA می باشد.
  - OD: خروجی ترانزیستوری DC
  - OC: خروجی رله ای
  - OA: خروجی ترایاکی AC

3. کارتهای ورودی/خروجی ترکیبی

- معمولا دارای برچسبی به رنگ آبی می باشند.
- شمارهٔ قطعه شامل حروف MD می باشد.

4. کارتهای ویژهٔ ورودی/خروجی

غیر از کارتهای استاندارد ورودی/خروجی دیجیتال کارتهای ویژهٔ ورودی/خروجی نیز موجود می باشد. کارتهایی همچون کارتهای ورودی/خروجی آنالوگ، کارتهای کنترل دما، کانترهای پرسرعت (High Speed Counters) و . . . جزء این سری از کارتها می باشند.

مشخصات ظاهری این کارتها در زیر آمده است:

- معمولا دارای برچسبی به رنگ آبی می باشند.

5. کارتهای ارتباطی

این کارتها شامل کارتهای واسط شبکه می باشند، شرکت OMRON تجهیزات لازم برای برقراری ارتباط در شبکه های استاندارد و پرسرعت را با کمترین هزینه فراهم آورده است. ارتباط و تبادل اطلاعات بین PLC ها و یا بین PLC با سامانه های اطلاعاتی سطح بالاتر توسط ارتباطات سریال (Serial Communication) ویا شبکه هایی همچون Ethernet وController Link برقرار می گردد.

مشخصات ظاهری این کارتها در زیر آمده است:

- معمولا دارای برچسبی به رنگ سبز می باشند.

5- فضای حافظه

5-1- تعریف فضای داده ها

در زیر فضای حافظهٔ مربوط به PLC های سری CJ/CS و همچنین PLC مدل CP1H بـه تفصـیل شـرح داده خواهد شد.

5-1-1- ناحیهٔ CIO (Common Input/Output Area)

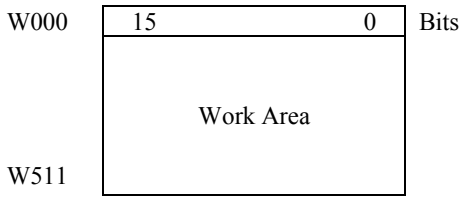
هنگام وارد کردن یک آدرس از منطقـهٔ CIO حافظـهٔ PLC نیـازی بـه وارد کـردن حـروف اختصـاری CIO در ابتدای آدرس نمی باشد. منطقهٔ حافظهٔ CIO عموماً برای تبادل و تجدید (Refreshing) اطلاعات بـین PLC و بخشهای ورودی و خروجی استفاده می شود. منطقهٔ CIO از 6144Word از 0000 تا 6143 (از بیـت 0000.00 تا 6143.15) تشکیل شده است. در ایـن منطقـه از حافظـه Word هـایی کـه بـه بخشـهای ورودی و خروجـی اختصاص داده نشده باشند را می توان در برنامه به عنوان Word ها وبیت های کاربردی و مجازی استفاده کرد.

CIO Address	CJ	CS	CIO Address
0000	I/O Area تبصرهٔ 1 Data Link Area Internal I/O Area CPU Bus Unit Area Special I/O Unit Area DeviceNet Area	I/O Area تبصرهٔ 1 Data Link Area Internal I/O Area CPU Bus Unit Area Special I/O Unit Area DeviceNet Area **Sysmac Bus Area** **I/O Terminal Area**	0000
6143			6143

1. می توان با انجام تنظیمات خاص و تعیین آدرس ابتدایی هر Rack از PLC از آدرسهای CIO0000 تـا CIO0999 برای بخشهای ورودی و خروجی دیجیتال استفاده کرد. این تنظیمات برای مشخص کـردن آدرس ابتدایی هر Rack توسـط نـرم افـزار CX-Programmer و در قسـمت جـدول ورودی/خروجـی (I/O Table) انجام می گیـرد. بـرای تعیـین آدرس ابتـدایی اولـین Rack مـی تـوان از CIO0000 تـا CIO0900 استفاده کرد.
2. بخشی از منطقهٔ حافظهٔ CIO که "بدون کـاربرد" مـی باشـد را مـی تـوان بـه عنـوان حافظـهٔ کـار در برنامه نویسی استفاده کرد. البته در آینده با اضافه شدن توابع جدید به PLC ها بیت هـایی از منطقـهٔ CIO که هم اکنون بدون استفاده است نیز بکار گرفته خواهند شد، به همین دلیل در ابتدا اسـتفاده از بیتهای منطقهٔ W(Work Area) به عنوان بیتهای کاربردی در برنامه توصیه میگردد.

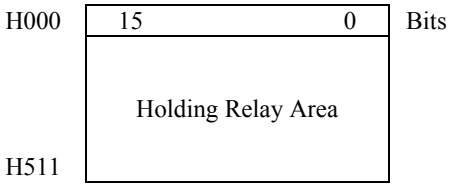
5-1-2- ناحیهٔ W (Work Area)

زمانی که از آدرسهای منطقهٔ کار (Work Area) استفاده می شود باید قبل از آدرس حرف "W" را برای مشخص کردن این منطقهٔ از حافظه قرار داد. منطقهٔ Work Area از 512Word که از W000 تا W511 (بیتهای W000.00 تا W511.15) می باشد، تشکیل شده است. Word های این منطقه از حافظه را فقط در برنامه نویسی می توان استفاده کرد، بدین منظور که از این Word ها نمی توان به عنوان آدرس های ورودی و خروجی که توسط ترمینالهای ورودی و خروجی به عناصر خارجی متصل است استفاده کرد. از آنجایی که در آینده تابع جدیدی به این منطقه از حافظهٔ PLC ها اختصاص داده نخواهد شد، لذا استفاده از این منطقهٔ حافظه به عنوان حافظهٔ مجازی قبل از استفاده از بیتهای بدون کاربرد منطقهٔ CIO توصیه می شود.
اگر این حافظه برای برنامه نویسی کافی نبود می توان از حافظهٔ CIO (CIO1200 تا CIO1499 و همچنین CIO3800 تا CIO6143) استفاده کرد.



5-1-3- ناحیهٔ H (Holding Area)

زمانی که از آدرسهای منطقهٔ پایدار (Holding) استفاده می شود باید قبل از آدرس حرف "H" را برای مشخص کردن این منطقهٔ حافظه قرار داد. از این منطقه از حافظه فقط در برنامه می توان استفاده کرد (نمی توان از این حافظه به عنوان حافظه مربوط به ورودی و خروجی های فیزیکی PLC استفاده کرد).
محتوای این حافظه از PLC در صورت خاموش شدن PLC و یا تغییر مد عملکرد آن ازمد Program به مد Run و یا مد Monitor حفظ خواهد شد. محتوای این منطقه در صورت وجود باتری حفظ می شود.
منطقهٔ Holding Area از 512Word که از H000 تا H511 (بیتهای H000.00 تا H511.15) می باشد، تشکیل شده است. می توان از این منطقه به صورت مورد نیاز، به صورت کنتاکتهای بسته و باز در برنامه استفاده کرد.



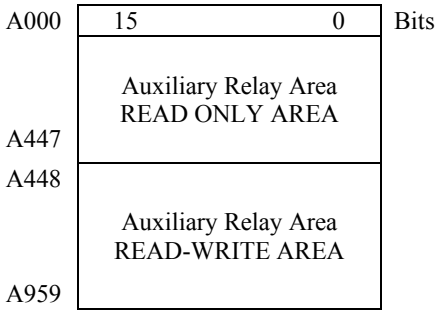
5-1-4- ناحیهٔ A (Auxiliary Area)

زمانی که از آدرسهای منطقهٔ کمکی (Auxiliary) استفاده می شود باید قبل از آدرس حرف "A" را برای مشخص کردن این منطقهٔ حافظه قرار داد. منطقهٔ Auxiliary شامل فلگهای از پیش تعیین شده و بیتهای

کنترلی است که برای نشان دادن سلامت CPU، و بخش های ورودی و خروجی خاص و همچنین مسیر ارتباطی CPU و باتری به کار می روند.

این منطقه شامل 960Word است که از A000 تا A959 می باشد. این منطقه از حافظه به دو بخش تقسیم شده است: A000 تا A447 منطقهٔ با قابلیت فقط خواندنی و A448 تا A959 منطقهٔ با قابلیت خواندن و نوشتن توسط برنامه می باشد.

CS	CJ
آدرسهای فقط خواندنی در منطقهٔ Auxiliary قابل کنترل به صورت مستقیم توسط برنامه و یا اختصاص دادن به بخش های ویژهٔ C200H نیستند.	آدرسهای فقط خواندنی در منطقهٔ Auxiliary قابل کنترل به صورت مستقیم توسط برنامه نیستند.



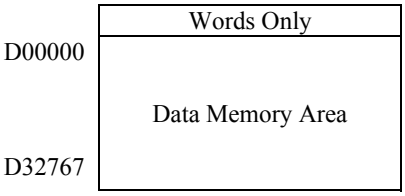
5-1-5- ناحیهٔ D (Data Memory Area)

زمانی که از آدرسهای منطقهٔ Data Memory استفاده می شود باید قبل از آدرس حرف "D" را برای مشخص کردن این منطقهٔ حافظه قرار داد (در گذشته این منطقه با حروف اختصاری "DM" نشان داده می شد که این مشخصه هنوز نیز در بعضی از PLC ها به کار می رود). این منطقه از حافظه، منطقه ای چند منظوره می باشد که فقط به صورت Word قابل استفاده است. محتوای این منطقه از حافظه در صورت خاموش شدن PLC و یا تغییر مد عملکرد آن از مد Program به مد Monitor و یا Run و بالعکس حفظ می شود. این منطقه از حافظه توسط باتری PLC پشتیبانی می گردد.

منطقه D از 32,768Word تشکیل شده است که از D00000 تا D32767 می باشد. این منطقه برای ذخیره کردن اطلاعات و همچنین انجام عملیات محاسباتی که به صورت Word انجام می گیرد، استفاده می شود.

منطقهٔ D معمولا برای ذخیره کردن اطلاعات همچون مقادیر ثابت، مقادیر ثبت شدهٔ گذشته، نتایج عملیاتهای محاسباتی و غیره . . . به کار می رود.

بیتهای این منطقه از حافظه قابل دسترسی به صورت مستقیم نمی باشند ولی می توان این بیتها را با دستورالعملهای BIT TEST همچون دستور TST(350) و TSTN(351) کنترل کرد. لازم به ذکر است که این بیتها را نمی توان با جبر (Force) روشن و یا خاموش کرد.



6-1-5- ناحیهٔ E (Extended Data Memory Area)

زمانی که از آدرسهای منطقهٔ Extended Data Memory استفاده می شود باید قبل از آدرس حرف "E" را برای مشخص کردن این منطقهٔ حافظه قرار داد (در گذشته این منطقه با حروف اختصاری "EM" نشان داده می شد که این مشخصه هنوز نیز در بعضی از PLC ها به کار می رود). این منطقه از حافظه، منطقه ای چند منظوره می باشد که فقط به صورت Word قابل دسترسی می باشد. محتوای این منطقه از حافظه در صورت خاموش شدن PLC و یا تغییر مد عملکرد آن از مد Program به مد Monitor و یا Run و بالعکس حفظ می شود. این منطقه از حافظه توسط باتری PLC پشتیبانی می گردد.

این منطقهٔ حافظه از بانکهای حافظه تشکیل شده است، که هر کدام از این بانکها شامل 32,768Word می باشند. حافظهٔ E از آدرس E0_0000 تا En_32767 تشکیل شده است. این منطقه برای ذخیره کردن اطلاعات و همچنین انجام عملیات محاسباتی که به صورت Word انجام می گیرد، استفاده می شود.

تعداد بانکهای موجود در این منطقه به مدل CPU بستگی دارد، که حداکثر تعداد آنها 13بانک (از 0 تا C) می باشد.

منطقهٔ E معمولا برای ذخیره کردن اطلاعاتی همچون مقادیر ثابت، مقادیر ثبت شدهٔ گذشته، نتایج عملیاتهای محاسباتی و غیره . . . به کار می رود.

بیتهای این منطقه از حافظه قابل دسترسی به صورت مستقیم نمی باشند ولی می توان این بیتها را با دستورالعملهای BIT TEST همچون دستور TST(350) و TSTN(351) کنترل کرد. لازم به ذکر است این بیتها را نمی توان با جبر (Force) روشن و یا خاموش کرد.

تعیین آدرسهای ناحیهٔ E

برای تعیین آدرس در منطقهٔ E دو روش وجود دارد: تعیین همزمان آدرس و همچنین بانک و یا مشخص کردن آدرس در بانک فعلی. به طور کلی آدرس دهی همزمان بانک مورد نظر و همچنین آدرس مورد نظر در آن بانک توصیه می گردد.

1. مشخص کردن بانک و آدرس

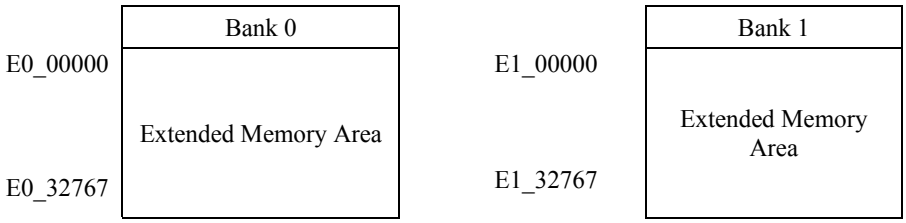
   در این روش شمارهٔ بانک مورد نظر باید بعد از "E" ذکر شود، مانند E2_00010 که مشخص کنندهٔ آدرس 00010 از بانک 2 در منطقهٔ E می باشد.

2. مشخص کردن آدرس در همان بانک فعلی

   در این روش فقط آدرس مورد نظر در E باید مشخص گردد. به عنوان مثال E00010 که مشخص کنندهٔ آدرس 00010 در بانک فعلی می باشد.

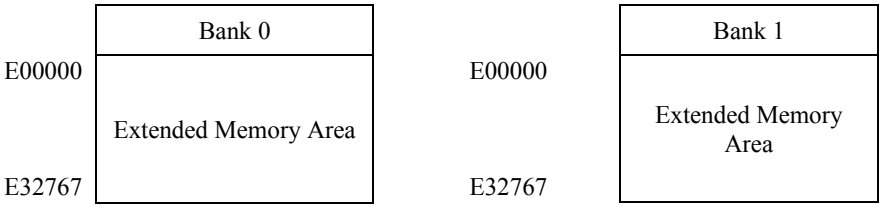
بانک فعلی زمانی که مد PLC از مد Program به مد Run/Monitor تغییر کند به 0 تبدیل می شود مگر اینکه بیت A500.12 که بیت نگهدارندهٔ حافظهٔ IO می باشد روشن باشد.

مشخص کردن بانک و آدرس مورد نظر در آن بانک



مشخص کردن آدرس در بانک فعلی

برای تغییر دادن شماره بانک باید از دستورالعمل EMBC(281) استفاده کرد.



5-1-7- ناحیهٔ تایمر (Timer Area)

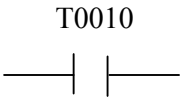
دو ناحیه برای اطلاعات تایمر وجود دارد: فلگ اتمام کار تایمرها (Completion Flag) و همچنین ارزش فعلی تایمرها (Present Value). هنگام مشخص کردن فلگ اتمام کار تایمرها باید حرف "T" قبل از آدرس ذکر شود. 4096 تایمر با شمارهٔ تایمر T0000 تا T4095 در PLC وجود دارد. که همهٔ دستورات تایمر، که شامل دستورات TIM، TIMH(015)، TMHH(540)، TTIM(087)، TIMW(813) و TMHW(815) می باشد بخشی از این فضای حافظه را اشغال می کنند. برای این دستورالعملها فلگ اتمام کار و همچنین ارزش فعلی با شماره تایمر مشخص می گردد. (برای دستورهای TIML(542) و MTIM(543) از این فضای حافظه استفاده نمی شود).

زمانیکه از آدرس یک تایمر در دستوری با عملگر بیتی، به عنوان عملگر استفاده می شود، این داده به عنوان فلگ اتمام کار تایمر در نظر گرفته می شود. ولی زمانیکه از آدرس تایمر در دستوری با عملگر Word استفاده شود، این داده به عنوان ارزش فعلی تایمر که خود داده ای به صورت Word است در نظر گرفته می شود. فلگ اتمام کار تایمر را می توان به دفعات مورد نیاز در برنامه به صورت کنتاکت باز و یا بسته استفاده کرد و همچنین ارزش فعلی تایمر را هم می توان به صورت یک دادهٔ Word به دفعات در دستورالعملهای مختلف استفاده کرد.

استفاده از دو تایمر مختلف با یک شمارهٔ تایمر در برنامه به هیچ عنوان توصیه نمی گردد، این امر باعث عملکرد نادرست تایمرها هنگام عملکرد همزمان آنان خواهد شد. (اگر از دو یا چند تایمر مختلف با یک شمارهٔ تایمر در برنامه استفاده شود در هنگام بررسی برنامه توسط نرم افزار خطایی ایجاد می گردد).

فلگ اتمام کار تایمر

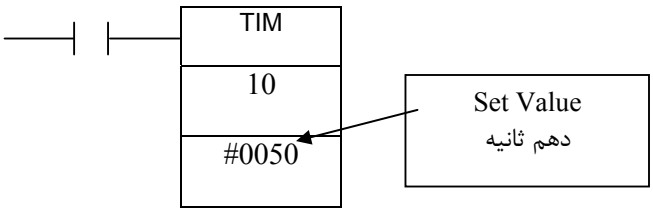
این فلگ به صورت بیتی فراخوانده می شود. این فلگ زمانی که زمان مشخص شده برای تایمر به پایان رسید و یا به عبارتی زمانی که کار تایمر به اتمام رسید روشن خواهد شد.



تایمر SV/PV

Set Value زمان تعیین شده برای تایمر است که باید سپری شود تا فلگ اتمام کار تایمر روشن گردد. Present Value به صورت Word در برنامه نوشته و یا فراخوانده می شود. در زمان فعال بودن تایمر PV به صورت معکوس شروع به کاهش می کند که نشان دهندهٔ زمان باقی مانده تایمر می باشد.

تایمرها برای ایجاد فاصلهٔ زمانی قبل از اجرای مرحلهٔ بعدی برنامه به کار می روند. تا این امکان را به وجود آورند که قبل شروع مرحلهٔ بعدی مرحلهٔ قبلی به پایان رسیده باشد.



5-1-8- ناحیهٔ کانتر (Counter Area)

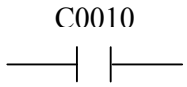
دو ناحیه برای اطلاعات کانتر وجود دارد: فلگ اتمام کار کانترها (Completion Flag) و همچنین ارزش فعلی کانترها (Present Value). هنگام مشخص کردن فلگ اتمام کار کانترها باید حرف "C" را قبل از آدرس قرار داد. 4096 کانتر با شماره کانتر C0000 تا C4095 وجود دارد. که همهٔ دستورات کانتر که شامل دستورات CNT، CNTR(012) وCNTW(814) می باشد، بخشی از این فضای حافظه را به خود اختصاص می دهند. برای این دستورالعملها فلگ اتمام کار و همچنین ارزش فعلی با شماره کانتر مشخص می گردد.

زمانیکه آدرس کانتر در دستوری با عملگر بیتی، به عنوان عملگر استفاده می شود، این داده به عنوان فلگ اتمام کار کانتر در نظر گرفته می شود. ولی زمانیکه آدرس کانتر در دستوری با عملگر Word استفاده شود، این داده به عنوان ارزش فعلی کانتر که خود داده ای Word است در نظر گرفته می شود.

استفاده از دو کانتر مختلف با یک شمارهٔ کانتر در برنامه به هیچ عنوان توصیه نمی گردد، این امر باعث عملکرد نادرست کانترها هنگام عملکرد همزمان آنان خواهد شد. (اگر از دو یا چند کانتر مختلف با یک شمارهٔ کانتر در برنامه استفاده شود در هنگام بررسی برنامه توسط نرم افزار به عنوان خطا در نظر گرفته می شود).

فلگ اتمام کار کانتر

این فلگ به صورت بیتی فراخوانده می شود. این فلگ زمانی که شمارش تعداد پالسهای مشخص شده برای کانتر به پایان رسید و یا به عبارتی زمانی که کار کانتر به اتمام رسید روشن خواهد شد.

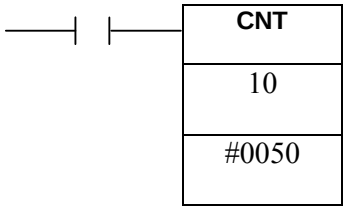


کانتر SV/PV

Set Value تعداد شمارش تعیین شده برای کانتر است که باید توسط کانتر شمرده شود تا فلگ اتمام کار کانتر روشن گردد.

Present Value به صورت Word در برنامه نوشته و یا فراخوانده می شود. در زمان فعال بودن کانتر PV به صورت افزایشی و یا کاهشی چگونگی شمارش را نشان می دهد.

کانترها برای شمارش کالاهای بازگشتی و یا کالاهای تولید شده به کار می روند.



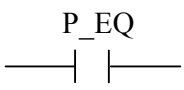
5-1-9- فلگهای وضعیت

این فلگها شامل فلگهای محاسباتی و منطقی مانند فلگ خطا (Error Flag) و فلگ مساوی (Equal Flag) می باشند، که نتیجهٔ اجرا یک دستورالعمل را نشان می دهند. علاوه بر آن فلگهای همیشه روشن (Always On Flag) و همیشه خاموش (Always Off Flag) نیز جز این فلگها می باشند. در PLC های قدیمی تر این فلگها در منطقهٔ SR حافظه قرار داشتند.

فلگهای وضعیت با اصطلاحات خاص مانند CY و یا ER مشخص می شوند و یا این فلگها با نمادهای خاصی همچون P_Carry و یا P_Error به جای آدرس آنها مورد استفاده قرار می گیرند. حالت این فلگها از نتیجهٔ اجرا دستورات در برنامه تأثیر می پذیرد. این فلگها دارای دسترسی فقط خواندنی بوده و نمی توان آنها را مستقیما با استفاده از دستورالعمل خاصی فرمان داد.

حالت تمام فلگهای وضعیت، زمانی که برنامه بین Task های مختلف سوییچ می کند باز نشانده (Reset) می شود. در نتیجه حالت فلگهای مانند ER و AER فقط در همان Task که استفاده می شوند صادق است.

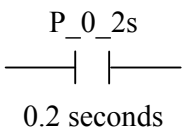
همچنین این فلگها را نمی توان با استفاده از جبر (Force) روشن و خاموش کرد.



در CX-Programmer فلگهای وضعیت نیز مانند بقیهٔ نمادها با "P_" آغاز می شوند.

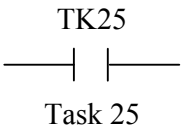
5-1-10- پالسهای زمانی

پالسهای زمانی توسط تایمر داخلی CPU روشن و خاموش می شوند. این بیتها را با استفاده از نمادها بـه جـای آدرس در برنامه مورد استفاده قرار می دهند.
پالسهای زمانی فقط خواندنی هستند، این پالسها را نمی توان با استفاده از دستورات فرمان داد.
این پالسها همگی در شروع عملکرد PLC بازنشانده (Reset) می شوند.



5-1-11- ناحیهٔ TK (Task Flag Area)

این ناحیه از حافظه از TK00 تا TK31 تشکیل شده است که مربوط به Task های سیکلی PLC می باشد. این فلگ در صورتی که Task مورد نظر با همان شماره قابل اجرا و یا در حالت Run باشد روشن و در صـورتی کـه Task مورد نظر قابل اجرا نباشد و یا در حالت انتظار باشد خاموش می شود.
این فلگها فقط نشان دهندهٔ حالت Task های سیکلی بوده و از Task های وقفه ای تأثیر نمی پذیرد.



5-1-12- ناحیهٔ IR (Index Register)

این 16 ثبات شاخص (IR00 تا IR15) برای ذخیـره کـردن آدرسـهای حافظـهٔ PLC بـرای آدرس دهـی غیـر مستقیم حافظه به کار برده می شود (آدرسهای مطلق حافظه در RAM). هر کدام از ثباتهـای شـاخص قابلیـت حفظ یکی از آدرس های PLC را دارا می باشد، که این آدرس ها به حافظه های مطلق PLC اشاره مـی کننـد.
این منطقه از حافظه را می تـوان بـه صـورت جداگانـه در یـک Task و یـا، بـرای PLC هـای مـدل CJ1H در همهٔ Task ها استفاده کرد.

5-1-13- ناحیهٔ DR (Data Register)

منطقه ثبات داده از حافظه (DR00 تا DR15) توأم با منطقهٔ IR استفاده می شود. زمانیکه داده ای درست قبل از وارد شدن یک آدرس در حافظهٔ IR ، در حافظهٔ DR وارد شود محتوای این حافظه بـه محتـوای حافظـهٔ IR برای مشخص کردن آدرس به صورت آدرس دهی غیرمستقیم اضافه می شود. از آنجـایی کـه محتـوای حافظـهٔ DR به صورت باینری علامت دار است، لذا محتوای IR می تواند به آدرسهای بالاتر و یا پایین تر تبـدیل شـود.
این منطقه از حافظه می تواند به صورت جداگانه در هر Task به کار برده شود و یـا بـرای PLC مـدل CJ1-H برای همهٔ Task ها به کار برده شود.
برای وارد کردن داده های مورد نیاز در منطقهٔ DR می توان از دستورالعملهای معمول برنامـه نویسـی اسـتفاده کرد.

5-2- نقشهٔ حافظه

TK00 – TK31	Task Flag Area
A000 – A959	Auxiliary Area
T0000 – T4095	Timer Completion Flags
C0000 – C4095	Counter Completion Flags
CIO 0000 – CIO 6143	Core I/O Area
H000 – H511	Holding Area
W000 – W511	Work Area
T0000 – T4095	Timer PVs
C0000 – C4095	Counter PVs
D00000 – D32767	Data Memory Area
E0_00000 – E*n*32767	Banks of Extended Memory

6- مقدمه ای بر CX-Programmer

در این بخش به استفاده از مناطق مختلف حافظه در نرم افزار CX-Programmer می پردازیم.
در این قسمت اجزاء سازندهٔ حافظهٔ PLC و قابلیت آنها برای نمایش و ویرایش داده ها به صورت بیتی و یا Word توسط کاربر توضیح داده خواهد شد.
در ابتدا به تعریف بیت (Bit)، کانال (Channel) و دیجیت (Digit) می پردازیم.

بیت (Bit)

- بیت یک واحد باینری است. (Binary Unit)
- بیت دارای دو حالت روشن و یا خاموش می باشد.

دیجیت (Digit)

- هر دیجیت از چهار بیت تشکیل شده است.
- و هر کانال از چهار دیجیت تشکیل شده است.
- دیجیتهای هر کانال از شمارهٔ 0 تا 3 شماره گذاری شده اند.

کانال (Channel)

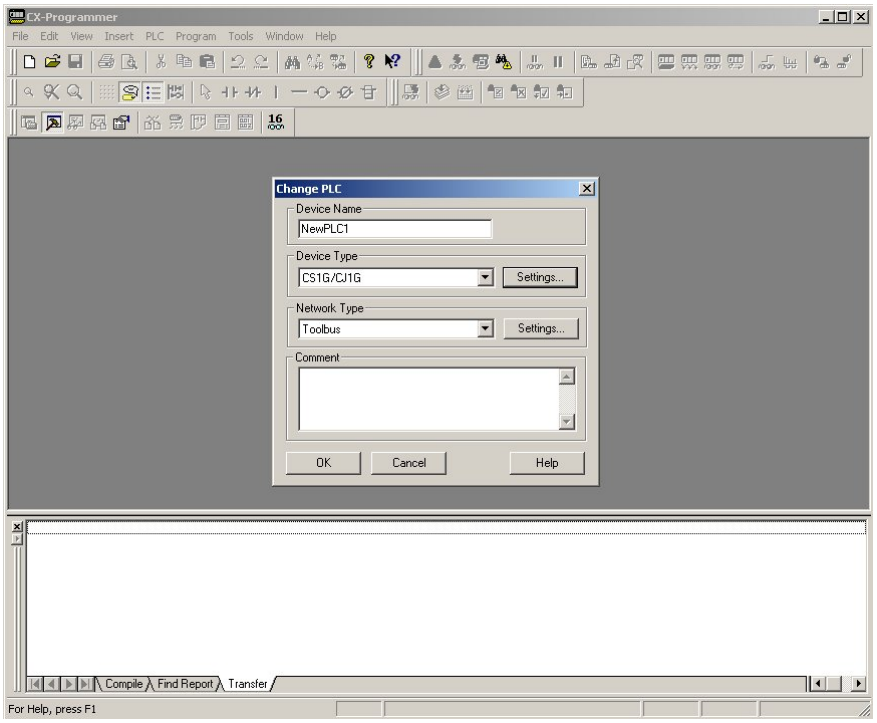
- کانال = Word
- هر کانال از شانزده بیت تشکیل شده است.
- بیتهای تشکیل دهندهٔ هر کانال از 0 تا 15 شماره گذاری شده اند.

بیت ها

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Digit 3				**Digit 2**				**Digit 1**				**Digit 0**			

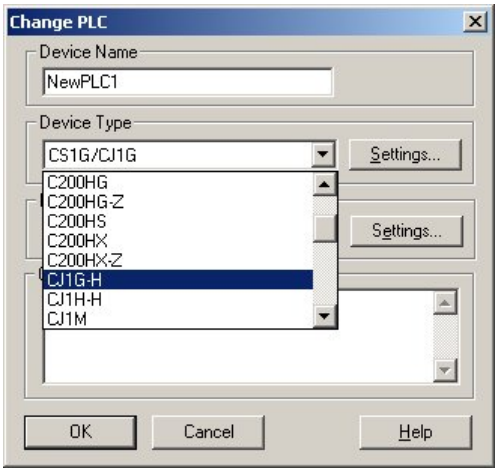
6-1- آماده شدن برای کار با CX-Programmer

1. قبل از شروع با کار نرم افزار (CX-Programmer) باید روی رایانه نصب گردد.
2. نرم افزار را از منوی Start رایانه انتخاب کرده و باز کنید.
3. یک پروژهٔ جدید ایجاد کنید.
   - از منوهای بالای نرم افزار منوی File را انتخاب کنید.
   - از فهرست این منو New را انتخاب کنید.
   - پنجره ای به نام Change PLC باز خواهد شد.



4. PLC مورد نظر را انتخاب کنید.

- می توان در قسمت Device Name برای PLC نام معنی داری که نشان دهندهٔ چگونگی عملکرد PLC باشد، به جای New PLC1 انتخاب کرد.
- با کلیک کردن روی فلش سمت راست در قسمت Device Type نوع PLC را انتخاب کنید.



- با کلیک کردن روی Setting نوع CPU مورد نظر را نیز مشخص کنید.
- برای بازگشت به پنجرهٔ Change PLC روی OK کلیک کنید.

5. نوع ارتباط

- در قسمت Network Type نوع ارتباط را Toolbus انتخاب کنید.

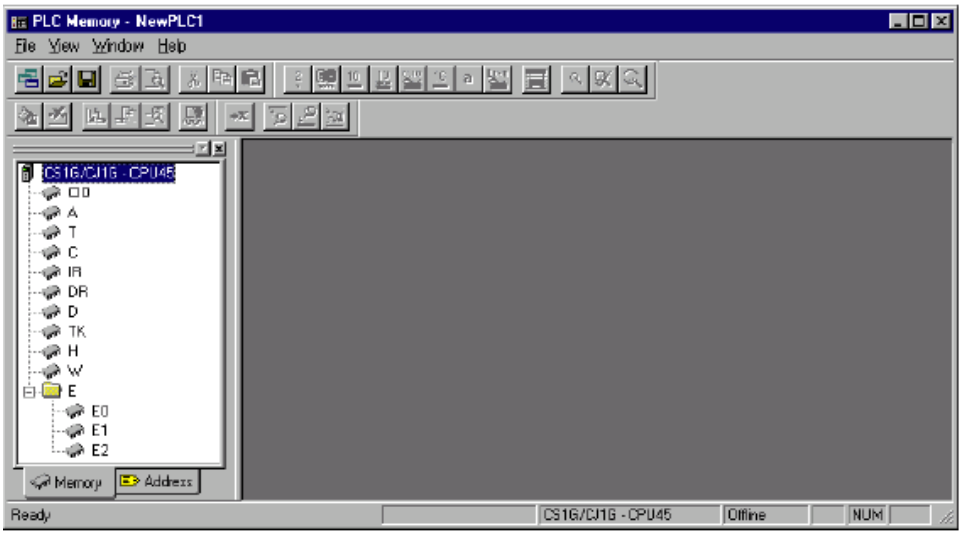
- OK را کلیک کنید تا وارد فضای کار نرم افزار شوید.
- روی قسمت Memory واقع در فضای کار نرم افزار دوبار کلیک کنید.

با نام بیت تعریف می شود.

برای آشنایی بیشتر با نرم افزار CX-Programmer به جزوهٔ "راهنمای استفاده از CX-Programmer" مراجعه شود.

6-2- کار با امکانات بخش Memory

جدول حافظه که نشان دهندهٔ اجزاء تشکیل دهندهٔ حافظهٔ PLC می باشد، امکان نمایش و ویرایش کامل فضای حافظه را به صورت Word برای کاربر فراهم می سازد.



در سمت چپ این پنجره که به فضای کار ناحیه های حافظه اختصاص دارد، بسته به نوع PLC می توان ساختار فضای حافظهٔ PLC را مشاهده کرد.

با دوبار کلیک کردن روی هر کدام از این مناطق می توان Word های تشکیل دهندهٔ آن منطقه و همچنین محتوای آنها را مشاهده کرد. همچنین قالب نمایش محتوای حافظه (باینری، Hex، BCD، Decimal) را نیز می توان انتخاب کرد.

6-2-1- داده های باینری

PLC نیز همچون رایانه برای نمایش داده ها، برنامه، تنظیمات، محاسبات و غیره از قالب باینری اعداد استفاده می کند. یک واحد باینری دارای دو حالت خاموش و روشن است که با نام بیت تعریف می شود.

یک بیت در حالت خاموش	**0**
یک بیت در حالت روشن	**1**

در نتیجه یک بیت می تواند نشان دهندهٔ حالت یک ورودی و یا خروجی سادهٔ دیجیتال همچون خاموش یا روشن بودن یک سنسور باشد. به عنوان مثال وقتی منطقهٔ CIO از حافظه را مشاهده می کنید این Word ها از 16 بیت تشکیل شده است.

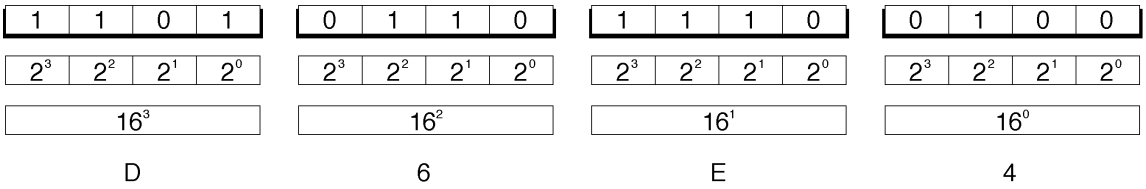
بیت ها

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}

6-2-2- داده های هگزادسیمال

سروکار داشتن با رشته ای از صفر و یکهای باینری کار مشکلی خواهد بود. بنابراین قالب Hex به عنوان روشی خلاصه برای نمایش و یا کار با اعداد باینری استفاده می شود. ترکیب هر چهار بیت را می توان به صورت یک عدد در مبنای Hex نمایش داد. در این قالب برای نمایش اعداد دسیمال 0 تا 15 از اعداد 0 تا 9 و حروف A تا F استفاده می شود.

هر Word که از 16 بیت تشکیل شده است را به چهار بخش که هرکدام متشکل از 4 بیت است تقسیم کرده و معادل Hex هر کدام محاسبه می شود.



همهٔ داده هایی که به PLC وارد می شود بدون توجه به قالب آن به داده ای با قالب باینری/ هگزادسیمال تبدیل می شود.

Binary	Hexadecimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

6-2-3- داده های BCD

به دلیل استفاده روزمره از داده های دسیمال این قالب از داده ها همواره برای کاربر قابل درک بوده است. به عنوان مثال اگر زمان تایمرها به صورت Hex و یا باینری مشخص شود عملکرد آنها قابل درک نخواهد بود.

با استفاده از بخشی از مبنای Hex که به صورت دسیمال می باشد برای تبدیل داده های باینری می توان مبنای جدیدی به نام BCD ایجاد کرد. در این مبنا فقط داده های باینری به اعداد 0 تا 9 دسیمال تبدیل می شود و حروف A تا F از این مبنا خارج می گردد. اعداد A تا F باعث ایجاد خطا در دستورالعملهایی که با داده های BCD کار می کنند، می گردد. به عنوان مثال تایمرها، کانترها و همچنین بعضی از دستورات محاسباتی با داده های باینری کار می کنند.

Binary	Hexadecimal	BCD
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	
1011	B	
1100	C	
1101	D	
1110	E	
1111	F	

6-2-4- داده های دسیمال

این داده ها در زندگی روزمره برای شمارش، اندازه گیری و . . . بقیهٔ امور استفاده می گردد. این قالب یکی از مبناهای اعداد است که از اعداد 0 تا 9 تشکیل شده است.

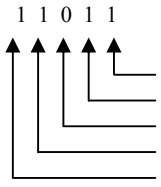
به عنوان مثال عدد 1538 در قالب دسیمال را می توان به صورت زیر در نظر گرفت:

10^0 = 1 x 8 = 8
10^1 = 10 x 3 = 30
10^2 = 100 x 5 = 500
10^3 = 1000 x 1 = 1,000
1,538 decimal

و همچنین عدد D6E6 در مبنای Hex، به صورت زیر به دسیمال تبدیل می شود:

16^0 = 1 x 4 = 4
16^1 = 16 x 14 = 224
16^2 = 256 x 6 = 1,536
16^3 = 4096 x 13 =
decimal

و عدد باینری 11011 نیز به صورت زیر به دسیمال تبدیل می شود.



2^0 = 1 x 1 = 1
2^1 = 2 x 1 = 2
2^2 = 4 x 0 = 0
2^3 = 8 x 1 = 8
2^4 = 16 x 1 = 16
27 decimal

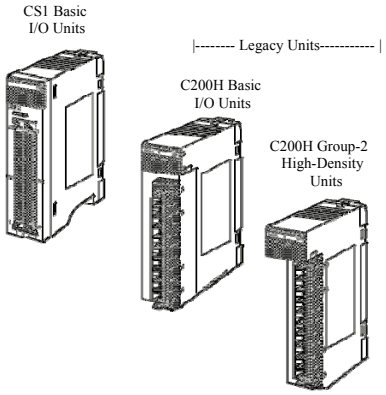
7- تخصیص آدرس ورودی/ خروجی

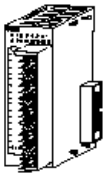
در PLC های سری CS/CJ بخشی از حافظهٔ ورودی/خروجی به هر کارت اختصاص داده می شود. این تخصیص حافظه برای کارتهای ورودی/ خروجی اصلی دیجیتال، کارتهای ورودی/ خروجی ویژه و همچنین واحدهای خطی CPU (CPU Bus Unit) متفاوت است.

7-1- ورودی/ خروجی های اصلی

اختصاص حافظهٔ I/O به ورودی/ خروجی های اصلی

CS	CJ
کارت های ورودی/ خروجی اصلی، شامل کارتهای ورودی/خروجی اصلی سری CS، گروه دوم از کارتهای با تراکم بالای ورودی/ خروجی سری C200H می باشد، این کارتها Word های منطقهٔ I/O (CIO0000 تا CIO0319) را به خود اختصاص می دهد و قابلیت قرارگیری در Rack اصلی و یا Rack های بسط یافته سری CS1 و C200H را دارا می باشند.	کارت های ورودی/ خروجی اصلی سری CJ، Word های حافظهٔ I/O (CIO0000 تا CIO0079) را به خود اختصاص می دهند و قابلیت قرارگرفتن در Rack اصلی و یا Rack های گسترش یافته را دارا می باشند.



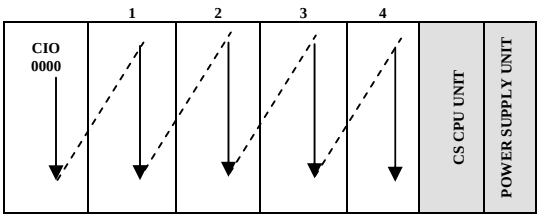


کارت های ورودی/ خروجی اصلی واقع در Rack اصلی PLC

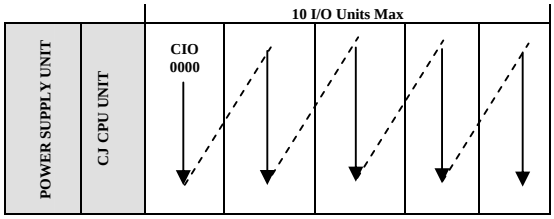
CS	CJ
در این سری از PLC ها کارتهای ورودی/ خروجی اصلی، از چپ به راست یعنی از دورترین کارت به واحد CPU، Word ها را به خود اختصاص می دهند. در این صورت هر کارت هر تعداد Word که لازم دارد را به خود اختصاص می دهد.	در این سری از PLC ها کارتهای ورودی/ خروجی اصلی، از چپ به راست یعنی از نزدیک ترین کارت به واحد CPU، Word ها را به خود اختصاص می دهند. در این صورت هر کارت هر تعداد Word که لازم دارد را به خود اختصاص می دهد..
در سری PLC های C200H شمارهٔ کارت (UnitNumber) کارتهای تراکم بالا از گروهٔ دو، نادیده گرفته می شود. در نتیجه با توجه به محل قرارگیری یک کارت در Rack آدرس آن مشخص می گردد.	
هیچ آدرسی به کارتهایی که جای آنها خالی است اختصاص نمی یابد. در صورت نیاز باید آدرس دهی را برای کارتهایی که جای آنها خالی است توسط نرم افزار و تغییرات در جدول I/O انجام داد.	

کارتهایی که از 1 تا 16 ورودی/خروجی تشکیل شده اند، 16 بیت را به خود اختصاص می دهند و کارتهایی که از 17 تا 32 ورودی/خروجی تشکیل شده اند، 32 بیت را به خود اختصاص می دهند. به عنوان مثال یک کارت 8 تایی ورودی/خروجی 16 بیت و یا یک Word کامل را به خود اختصاص می دهد. ولی بیتهای 00 تا 07 توسط این کارت استفاده می شود.

CS



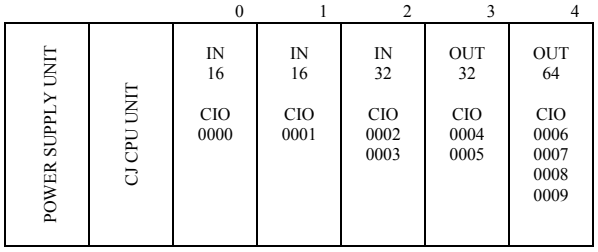
CJ



توجه: در PLC های سری CJ1 واحد CPU در سمت چپ و در PLC های سری CS1 واحد CPU در سمت راست قرار می گیرد.

مثال 1 از سری CJ1 :

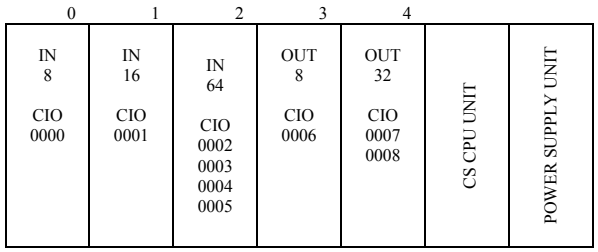
در این مثال چگونگی تخصیص آدرس به 5 کارت ورودی/خروجی اصلی واقع در Rack اصلی PLC نشـان داده می شود.



مکان قرارگیری از سمت چپ CPU	کارت	تعداد Word های مورد نیاز	Word های اختصاص داده شده
1	CJ1W-ID211 16-point DC Input Unit	1	CIO 0000
2	CJ1W-ID211 16-point DC Input Unit	1	CIO 0001
3	CJ1W-ID231 32-point DC Input Unit	2	CIO 0002 and CIO 0003
4	CJ1W-OD231 32-point Transistor Output Unit	2	CIO 0004 and CIO 0005
5	CJ1W-OD261 64-point Transistor Output Unit	4	CIO 0006 to CIO 0009

مثال 1 از سری CS1 :

در این مثال چگونگی تخصیص آدرس به 5 کارت ورودی/خروجی اصلی واقع در Rack اصلی PLC نشـان داده می شود.

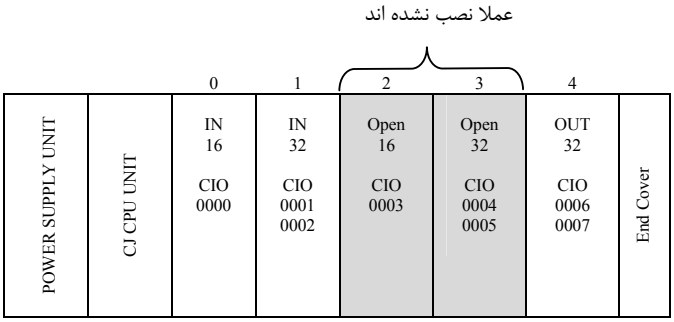


Slot	کارت	تعداد Word های مورد نیاز	Word های اختصاص داده شده
0	C200H-ID211 8-point DC Input Unit	1	CIO 0000
1	C200H-ID212 16-point DC Input Unit	1	CIO 0001
2	C200H-ID217 64-point DC Input Unit	4	CIO 0002 to CIO 0005
3	C200H-OD411 8-point Transistor Output Unit	1	CIO 0006
4	C200H-OD218 32-point Transistor Output Unit	2	CIO 0007 to CIO 0008

مثال 2 از سری CJ1 :

در PLC های سری CJ1، برای حفظ کردن Word های استفاده نشده به واحدهای ساختگی نیازی نمی باشد، Word های استفاده نشده به صورت نرم افزاری و در جدول ورودی/خروجی ها با تعریف واحدهای مجازی لحاظ می شوند، که با انتقال جدول ورودی/ خروجی به PLC این آدرس ها ذخیره می شوند.

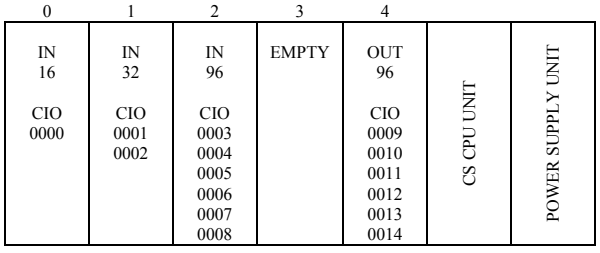
در مثال زیر چگونگی تخصیص آدرس به 3 کارت ورودی/خروجی اصلی همراه با دو واحد خالی استفاده نشده واقع در Rack اصلی PLC نشان داده می شود.



مکان قرارگیری از سمت چپ CPU	کارت	تعداد Word های مورد نیاز	Word های اختصاص داده شده
1	CJ1W-ID111 16-point DC Input Unit	1	CIO 0000
2	CJ1W-ID231 32-point DC Input Unit	2	CIO 0001 and CIO 0002
3	استفاده نشده (created in I/O table on CX-Programmer)	1	CIO 0003
4	استفاده نشده (created in I/O table on CX-Programmer)	2	CIO 0004 and CIO 0005
5	CJ1W-OD231 32-point Transistor Output Unit	2	CIO 0006 and CIO 0007

مثال 2 از سری CS1 :

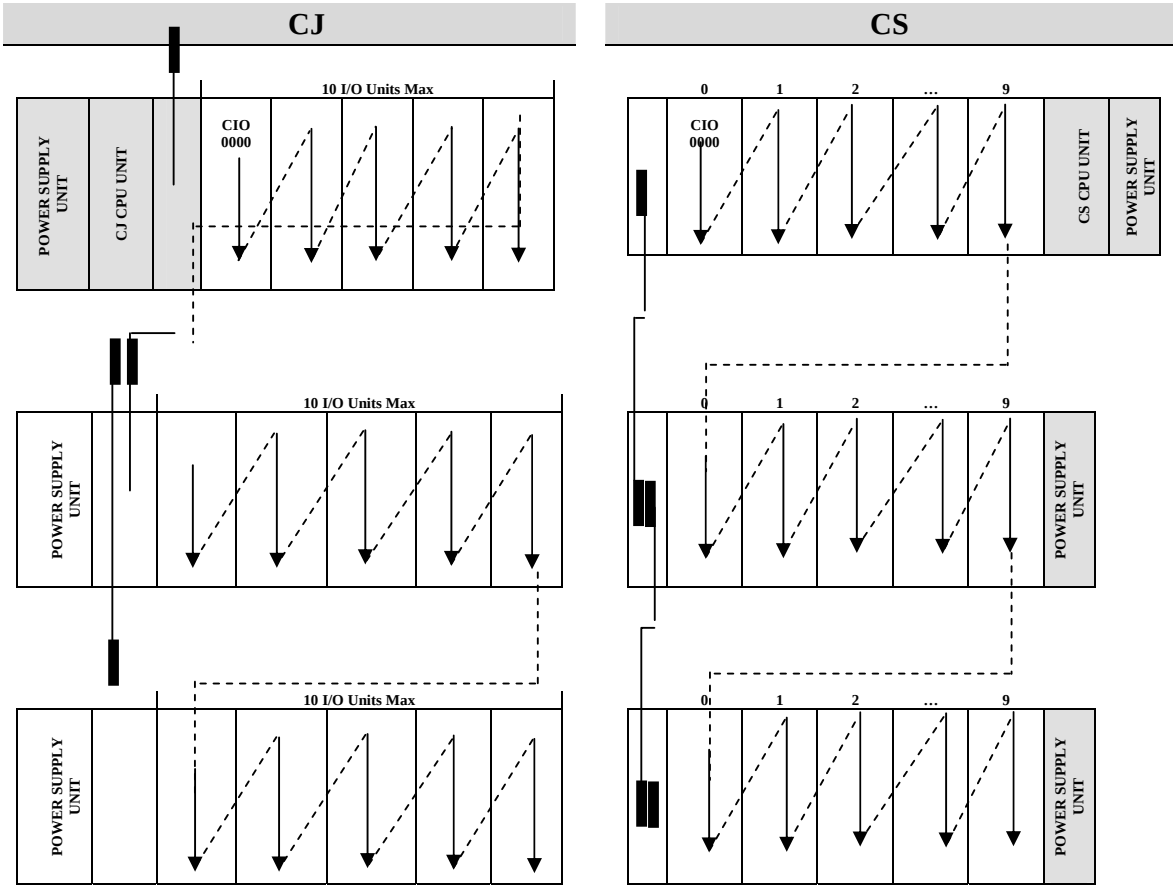
در مثال زیر چگونگی تخصیص آدرس به 4 کارت ورودی/خروجی اصلی همراه با 1 واحد خالی استفاده نشده واقع در Rack اصلی PLC نشان داده می شود.



Slot	کارت	تعداد Word های مورد نیاز	Word های اختصاص داده شده
0	C200H-ID212 16-point DC Input Unit	1	CIO 0000
1	C200H-ID216 32-point DC Input Unit	2	CIO 0001 to CIO 0002
2	CS1W-ID291 96-point DC Input Unit	6	CIO 0003 to CIO 0008
3	Empty	0	None
4	CS1W-OD291 96-point Transistor Output Unit	6	CIO 0009 to CIO 0014

7-2- کارت های ورودی/خروجی اصلی در Rack های بسط یافته

اختصاص آدرس به کارتهای ورودی/خروجی اصلی در Rack های بسط یافته متصل به CPU همچون Rack اصلی ادامه پیدا می کند. اختصاص Word ها از سمت چپ به راست صورت می گیرد و همچنین هر کارت به تعداد مورد نیاز Word اشغال می کند، همانند کارتها در Rack اصلی.

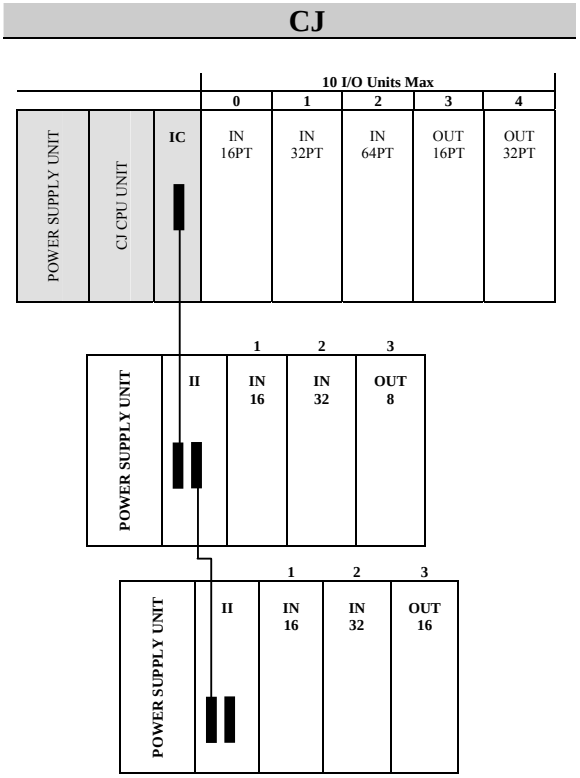


3 Expansion racks maximum

7 Expansion racks maximum

مثال 3 از سری CJ1 :

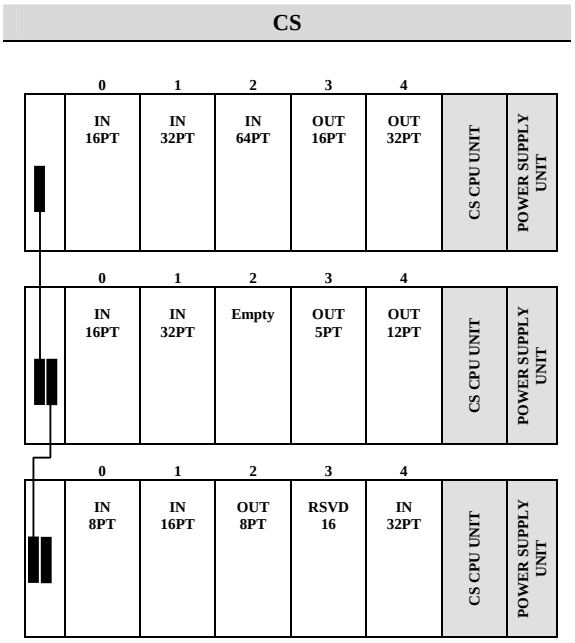
در مثال زیر چگونگی تخصیص آدرس به کارتهای اصلی ورودی/خروجی موجود در Rack اصلی و همچنین دو Rack بسط یافته از PLC های سری CJ1، نشان داده می شود.



Rack	مکان قرارگیری از سمت چپ CPU	کارت	تعداد Word های مورد نیاز	Word های اختصاص داده شده
CPU Rack	1	CJ1W-ID211 16-point DC Input Unit	1	CIO 0000
	2	CJ1W-ID231 32-point DC Input Unit	2	CIO 0001 and CIO 0002
	3	CJ1W-ID261 64-point DC Input Unit	4	CIO 0003 to CIO 0006
	4	CJ1W-OD211 16-point Transistor Output Unit	1	CIO 0007
	5	CJ1W-OD231 32-point Transistor Output Unit	2	CIO 0008 and CIO 0009
Expansion Rack	1	CJ1W-ID211 16-point DC Input Unit	1	CIO 0010
	2	CJ1W-ID231 32-point DC Input Unit	2	CIO 0011 and CIO 0012
	3	CJ1W-OC201 8-point Relay Output Unit	1	CIO 0013
Expansion Rack	1	CJ1W-ID211 16-point DC Input Unit	1	CIO 0014
	2	CJ1W-ID231 32-point DC Input Unit	1	CIO 0015 and CIO 0016
	3	CJ1W-OC211 16-point Relay Output Unit	1	CIO 0017

مثال 3 از سری CS1 :

در مثال زیر چگونگی تخصیص آدرس به کارتهای اصلی ورودی/خروجی موجود در Rack اصلی و همچنین دو Rack بسط یافته از PLC های سری CS1، نشان داده می شود.

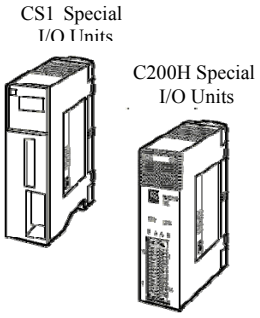


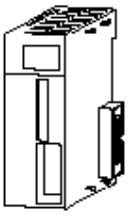
Rack	Slot	کارت	تعداد Word های مورد نیاز	Word های اختصاص داده شده
CPU Rack	0	C200H-ID212 16-point DC Input Unit	1	CIO 0000
	1	C200H-ID216 32-point DC Input Unit	2	CIO 0001 and CIO 0002
	2	C200H-ID217 64-point DC Input Unit	4	CIO 0003 to CIO 0006
	3	C200H-OD212 16-point Transistor Output Unit	1	CIO 0007
	4	C200H-OD218 32-point Transistor Output Unit	2	CIO 0008 and CIO 0009
CS-Series Expansion Rack	0	C200H-ID212 16-point DC Input Unit	1	CIO 0010
	1	C200H-ID216 32-point DC Input Unit	2	CIO 0011 and CIO 0012
	2	Empty	0	None
	3	C200H-OC223 5-point Relay Output Unit	1	CIO 0013
	4	C200H-OA224 12-point Triac Output Unit	1	CIO 0014
CS-Series Expansion Rack	0	C200H-IA121 8-point AC Input Unit	1	CIO 0015
	1	C200H-ID212 16-point DC Input Unit	1	CIO 0016
	2	C200H-OC222 12-point Relay Output Unit	1	CIO 0017
	3	Reserve one word. (See note.)	1	CIO 0018
	4	C200H-ID216 32-point DC Input Unit	2	CIO 0019 and CIO 0020

برای ذخیره کردن Word های مورد نیاز برای کارتهای خالی در Rack می توان از جدول ورودی/خروجی موجود در نرم افزار CX-Programmer استفاده کرد.

7-3- اختصاص آدرس به کارتهای ویژهٔ ورودی/ خروجی

CS	CJ
کارت های ورودی/ خروجی ویژه شامل کارتهای ویژهٔ سری CS1 و همچنین کارتهای ویژهٔ سری C200H می باشد. هر کدام از این کارتها ده Word را در منطقه ای که به آنها در فضای CIO اختصاص داده شده است (CIO2000 تا CIO2959) مطابق با شمارهٔ کارت، اشغال می کنند. این کارتها می توانند در Rack اصلی و Rack های بسط یافتهٔ سری CS1 و همچنین C200H نصب گردند*.	هر کدام از کارتهای ورودی/خروجی ویژه در سری CJ1 از PLC ها، ده Word را در منطقه ای که به آنها در فضای CIO اختصاص داده شده است (CIO2000 تا CIO2959) مطابق با شمارهٔ کارت، اشغال می کنند. این کارتها می توانند در Rack اصلی، Rack های بسط یافتهٔ سری CJ1 نصب گردند.
*کارت های ویژهٔ سری CS1 قابلیت نصب روی Rack های بسط یافتهٔ سری C200H را ندارند.	





7-3-1- چگونگی تخصیص Word ها

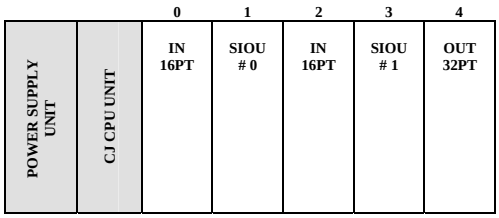
جدول زیر چگونگی تخصیص آدرس به کارتهای ویژهٔ ورودی/خروجی را نشان می دهد.

شمارهٔ کارت	Word های اختصاص داده شده
0	CIO 2000 to CIO 2009
1	CIO 2010 to CIO 2019
2	CIO 2020 to CIO 2029
:	:
15	CIO 2150 to CIO 2159
: :	: :
95	CIO 2950 to CIO 2959

در هنگام آدرس دهی کارتهای اصلی PLC، کارتهای ویژه نادیده گرفته می شود چون همانطور که گفته شد این کارتها منطقه ای خاص از حافظهٔ CIO را با توجه به شماره کارت که به صورت سخت افزاری از روی کارت تعیین می گردد، به خود اختصاص می دهند.

مثال 4 از سری CJ1 :

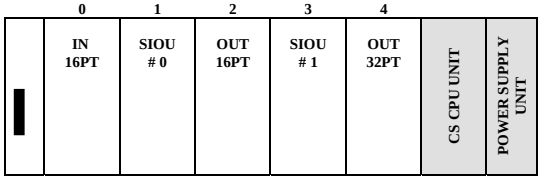
مثال زیر چگونگی آدرس دهی کارتهای اصلی و همچنین ویژه واقع در Rack اصلی را نشان می دهد.



Slot	کارت	تعداد Word های مورد نیاز	Word های اختصاص داده شده	شمارهٔ کارت	گروه
0	CJ1W-ID211 16-point DC Input Unit	1	CIO 0000	---	Basic I/O Unit
1	CJ1W-AD081 Analog Input Unit	10	CIO 2000 to CIO 2009	0	Special I/O Unit
2	CJ1W-OD211 16-point Transistor Output Unit	1	CIO 0001	---	Basic I/O Unit
3	CJ1W-TC001 Temperature Control Unit	20	CIO 2010 to CIO 2029	1	Special I/O Unit
4	CJ1W-OD231 32-point Transistor Output Unit	2	CIO 0002 and CIO 0003	---	Basic I/O Unit

مثال 4 از سری CS1 :

مثال زیر چگونگی آدرس دهی کارتهای اصلی و همچنین ویژه واقع در Rack اصلی را نشان می دهد.



Slot	کارت	تعداد Word های مورد نیاز	Word های اختصاص داده شده	شمارهٔ کارت	گروه
0	C200H-ID212 16-point DC Input Unit	1	CIO 0000	---	Basic I/O Unit
1	C200H-AD002 Analog Input Unit	10	CIO 2000 to CIO 2009	0	Special I/O Unit
2	C200H-OD21A 16-point Transistor Output Unit	1	CIO 0001	---	Basic I/O Unit
3	C200H-NC211 Position Control Unit	20	CIO 2010 to CIO 2029	1	Special I/O Unit
4	C200H-OD218 32-point Transistor Output Unit	2	CIO 0002 and CIO 0003	---	Basic I/O Unit

7-4- ایجاد جدول ورودی/خروجی

برای شناختن کارتهای متصل به PLC و همچنین آدرس دهی آنها باید جدول ورودی/خروجی را تعریف کرد. دو روش برای تعریف جدول ورودی/خروجی وجود دارد. انتخاب این روشها به چگونگی تنظیمات آدرس دهی توسط کاربر بستگی دارد.

7-4-1- تعریف جدول ورودی/خروجی به صورت خودکار در زمان راه اندازی PLC

مقصود از این روش:

در صورتی که نیاز به بررسی درست بودن کارتهای موجود در PLC و ذخیره کردن بعضی از Word های حافظه نباشد، تنظیمات از پیش تعیین شده برای تعریف جدول ورودی/خروجی به صورت خودکار استفاده می شود. که به صورت خودکار در زمان راه اندازی PLC به آدرس دهی کارتها می پردازد.

روش:

در این روش با هر بار راه اندازی PLC و وصل تغذیهٔ آن جدول ورودی/خروجی به صورت خودکار بسته به کارتهای نصب شده، تعریف می شود.

طریقهٔ عملکرد:

این روش به عنوان پیش فرض در تنظیمات قرار گرفته است، در نتیجه برای اجرا این روش عملکرد خاصی لازم نیست. در این روش کافی است کارتها نصب گردد و تغذیهٔ PLC وصل شود. می توان توسط نرم افزار حافظهٔ PLC و جدول ورودی/خروجی موجود در حافظه را پاک کرد. در صورتی که جدول ورودی/خروجی تعریف و به PLC انتقال داده نشده باشد، سیستم به صورت پیش فرض در هر بار راه اندازی CPU همین روش را برای تعریف جدول ورودی/خروجی در نظر می گیرد.

بررسی کارتها:

در صورت استفاده از این روش هیچ بررسی برای مقایسهٔ جدول ورودی/خروجی و سخت افزار نصب شدهٔ واقعی صورت نمی گیرد.

7-4-2- تعریف جدول ورودی/خروجی توسط کاربر

مقصود از این روش:

در صورتی که نیاز به بررسی سخت افزار سیستم از لحاظ قرارگیری کارتهای مختلف و همچنین ذخیرهٔ بعضی از Word های استفاده نشده باشد، تعریف جدول ورودی/خروجی توسط کاربر انجام می گیرد.

روش:

در این روش جدول ورودی/خروجی توسط کاربر تعریف و به PLC منتقل می شود.

طریقهٔ عملکرد:

باید جدول ورودی/خروجی را توسط نرم افزار CX-Programmer تعریف و آن را به CPU منتقل کرد. بعد از آن با هر بار روشن کردن PLC این جدول مورد استفاده قرار می گیرد.

با یکبار انتقال این جدول به PLC، این تنظیمات توسط PLC در نظر گرفته می شود مگر اینکه این جدول از حافظه حذف گردد.

بررسی کارتها:

در این روش با هربار راه اندازی PLC جدول موجود در PLC با ساختار اصلی سخت افزاری مقایسه شده و در صورت وجود مغایرت بیت A401.10 که نشان دهندهٔ این تفاوت است به عنوان خطا روشن می شود و عملکرد PLC امکان پذیر نخواهد بود.

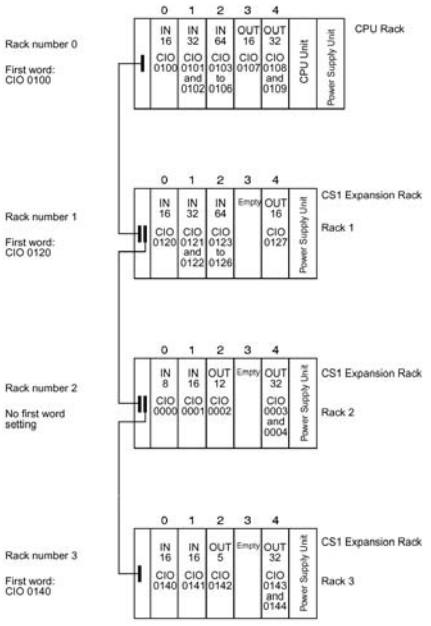
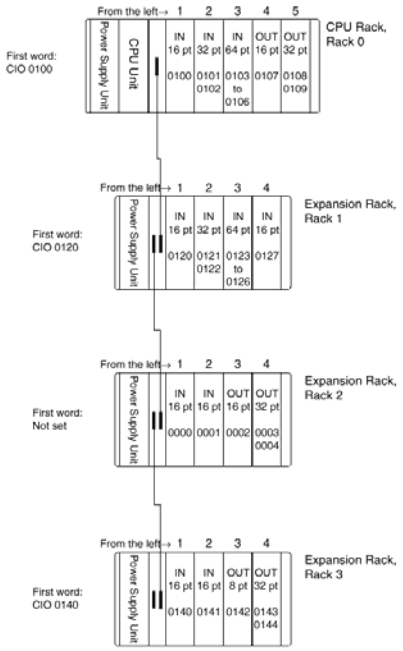
7-5- تعیین اولین Word از هر Rack

اولین Word برای آدرس دهی هر Rack را می توان توسط تعریف جدول ورودی/خروجی در نرم افزار CX-Programmer مشخص کرد.

CS	CJ
شمارهٔ Rack از 0 تا 7 به ترتیب قرارگیری Rackهای متصل به PLC توسط کابلهای مخصوص تعیین می شود. (Rack اصلی که شامل CPU می شود شمارهٔ 0 و Rack های بسط یافته به ترتیب از شمارهٔ 1 تا 7 می باشند.) شمارهٔ Rack را نمی توان تغییر داد زیرا این شماره به مکان قرارگرفتن آن نسبت به Rack اصلی بستگی دارد.	شمارهٔ Rack ها از 0 تا 3 به ترتیب قرارگیری Rackهای متصل به PLC توسط کابلهای مخصوص تعیین می شود. (Rack اصلی که شامل CPU می شود شمارهٔ 0 و Rack های بسط یافته به ترتیب از شمارهٔ 1 تا 3 می باشند.) شمارهٔ Rack را نمی توان تغییر داد زیرا این شماره به مکان قرارگرفتن آن نسبت به Rack اصلی بستگی دارد.

برای Rack هایی که اولین Word آدرس دهی آنها مشخص شده است، آدرس دهی با توجه به Word مشخص شده و به ترتیب برای کارتها از سمت چپ به راست صورت می گیرد. ولی برای Rack هایی که اولین Word آنها مشخص نشده باشد، آدرس دهی به ترتیب از سمت چپ به راست و به ترتیب شمارهٔ Rack از CIO0000 انجام می گیرد.

مثالی از آدرس دهی Rack هایی که اولین آدرس آنها توسط کاربر تعیین شده است در زیر آمده است.

CS	CJ
	

این مثال یک سیستم را که شامل یک Rack اصلی و سه Rack بسط یافته است را نشان می دهد.

7-5-1- تنظیم اولین Word برای آدرس دهی هر Rack توسط CX-Programmer

اولین Word برای آدرس دهی هر Rack را می توان توسط نرم افزار CX-Programmer تعیین کرد.

برای تعیین اولین Word از هر Rack باید مراحل زیر را انجام داد.

1. از منوی Option، Rack Start Addresses را انتخاب کنید.
2. در پنجره ای که باز شده است، تیکی که باعث غیرفعال شدن تعیین اولین آدرس Rack شده است را برداشته و آدرس موردنظر را به عنوان آدرس آغازین Rack وارد کنید.
3. با کلیک کردن روی OK تنظیمات انجام شده را تأیید کنید.

7-6- اندوختن Word ها برای تغییرات پیش بینی شده

در ساختار PLC ممکن است تغییراتی برای آینده در نظر گرفته شود، با ذخیره کردن بعضی از Word های حافظه برای تغییرات پیش بینی شده می توان تغییرات در برنامه را در صورت اضافه شدن و یا ایجاد تغییرات سخت افزاری به حداقل رساند. این ذخیره سازی Word ها توسط نرم افزار CX-Programmer و جدول ورودی/خروجی انجام می گیرد.

برای ذخیره کردن و همچنین انتقال جدول ورودی/خروجی جدید مراحل زیر باید انجام شود.

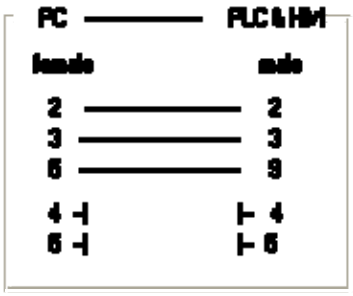
1. جدول ورودی/خروجی را باز کنید.

2. کارتهای موجود را تعریف کنید.
3. برای هر کارتی که قرار است به صورت فیزیکی استفاده نشده ولی آدرس موردنظر ذخیره شود یک کارت مجازی (Dummy) انتخاب کنید.
4. جدول را بار دیگر بررسی کنید.
5. به PLC متصل شده و جدول را به PLC انتقال دهید، با انتقال تنظیمات جدید، PLC بر طبق تنظیمات جدید کاربر شروع به کار می کند.

8- آماده کردن PLC برای برنامه نویسی

8-1- برقراری ارتباط

1. **CS1W-CN226** کابل "Toolbus". این کابل دارای دو متر طول می باشد و همچنین نوع دیگری از این کابل با طول شش متری نیز وجود دارد (CS1W-CN626). این کابل را می توان برای PLC های سری CS/CJ، CPM2C، CQM1H و همچنین PLC های جدید ساخت شرکت OMRON استفاده کرد. این کابل برای اتصال پورت منحصر بفرد Peripheral روی CPU استفاده می شود و فقط مخصوص شرکت OMRON می باشد ولی می توان با استفاده از تبدیل هایی آن را به پورت RS-232 تبدیل کرد.
2. **C200HS-CN220-EU** کابل "Sysmacway".این کابل از یک اتصال 9-pin نری و یک اتصال 9-pin ماده تشکیل شده است که پورت RS-232 موجود روی PLC را به پورت سریال رایانه متصل می نماید. برای ساخت کابل چگونگی اتصالات این کابل را می توان در اکثر کتابهای راهنمای برنامه نویسی PLC و یا سایت اینترنتی شرکت OMRON یافت.



8-1-1- برقراری ارتباط با PLC

برای برقراری ارتباط با PLC ابتدا باید کابل مناسب برای ارتباط بین PLC و رایانه را با توجه به مدلPLC انتخاب کرد و آن را به پورتهای ارتباطی PLC و رایانه متصل کرد، سپس یکی از روش های زیر را برای برقراری ارتباط انتخاب کرد.

در منوی PLC از نرم افزار CX-Programmer گزینهٔ "Work Online" را انتخاب کنید.

و یا

روی نماد زرد مثلثی شکل موجود در نوار ابزار کلیک کنید.

و یا

از روی صفحه کلید CTRL+W را فشار دهید.

هر کدام از این شیوه ها باعث برقراری ارتباط با PLC و یا قرار گرفتن در مسیر ارتباط با PLC می گردد.

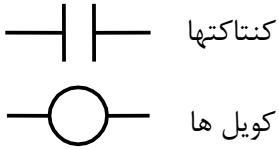
8-1-2- برنامهٔ کنترلی

برنامهٔ کنترل، برنامه ای نرم افزاری است که در حافظهٔ PLC قرار می گیرد. این برنامه همان کنترلی است که توسط کنترل کنندهٔ منطقی برنامه پذیر (PLC) اعمال می شود. کاربر و یا طراح سیستم کسی است که این برنامهٔ کنترلی را ایجاد می کند. برنامه کنترلی از بخشهایی به نام دستورالعمل تشکیل شده است. دستورالعمل ها در واقع کدهای کامپیوتری ساده ای هستند که ارتباط بین ورودی ها و خروجی های را برای رسیدن به هدف کنترلی برقرار می کنند.

8-1-3- دستورالعملهای اولیه

دستورالعملهای مختلفی برای برنامه نویسی وجود دارد که این دستورالعملها امکان انجام عملیات گوناگونی (جمع و تفریق داده ها، محاسبهٔ زمان و شمارش، مقایسهٔ داده ها و . . .) را به PLC می دهد. کاری که در برنامه نویسی صورت می گیرد استفاده از دستورالعمل ها با ترتیب مناسب است که دستور لازم را به تجهیزات خروجی آنچنان که مورد نظر برنامه نویس می باشد ارسال نماید.

PLC های مختلف دستورالعملهای مختلفی برای برنامه نویسی دارند. این دستورالعملها یکی از مواردی است که بین PLC ها تفاوت ایجاد می کند. به هر حال همهٔ PLC ها از دو دستورالعمل اولیه که در زیر نشان داده شده است، استفاده می کنند.



کنتاکتها

کنتاکتها دستورالعملهایی برای نمایش حالت ورودی در برنامهٔ کنترلی می باشند. کنتاکتها در واقع اطلاعات حالت می باشند که از طرف تجهیزات ورودی نصب شده در فیلد ارسال می گردند. هر کنتاکت در برنامهٔ کنترلی نشان دهندهٔ حالت یک عنصر واقعی ورودی می باشد. یک کنتاکت برای تغییر حالت (به عنوان مثال روشن ویا خاموش شدن و غیره . . . که بستگی به نوع کنتاکت دارد) منتظر ایجاد تغییر در عنصر ورودی می شود که موجب ایجاد تغییر در حالت خروجی ها بر طبق برنامهٔ کنترلی می گردد.

کویل ها

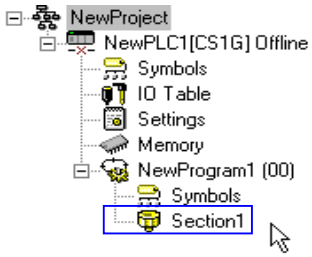
کویل ها دستورالعملهایی هستند که مشخص کنندهٔ حالت خروجی ها در برنامهٔ کنترلی می باشند. در واقع کویل ها تعیین کنندهٔ حالت تجهیزات خروجی هستند که از PLC فرمان می گیرند. همانند کنتاکتها، کویل ها نیز تعیین کنندهٔ حالت یک عنصر واقعی خروجی می باشند. اما برخلاف کنتاکتها که نمایش دهندهٔ حالت ورودی های بوده و حالت ورودی ها را برای PLC مشخص می کنند، کویل ها با توجه به حالت ورودی ها و منطق برنامهٔ کنترلی چگونگی عملکرد خروجی ها را تعیین می کنند. به عنوان مثال زمانی که یکی از

کنتاکتهایی که نمایندهٔ یک سوییچ در برنامهٔ کنترلی می باشد روشن شدن آن سوییچ را در فیلد به PLC اعلام می کند دستور روشن شدن یکی از موتورها توسط کویل ها به خروجی فرستاده می شود.

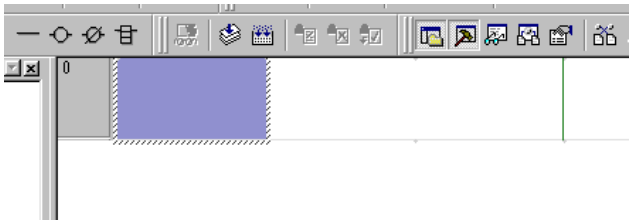
وارد کردن کنتاکتها و کویل ها

دو روش برای نوشتن برنامهٔ نردبانی وجود دارد، یکی با انتخاب عناصر برنامه نویسی توسط موس و قرار دادن آنها در برنامه و یا استفاده از صفحه کلید برای نوشتن برنامه.

با دوبار کلیک کردن روی Section1 در فضای کار، پنجرهٔ برنامه نویسی نردبانی باز می شود.

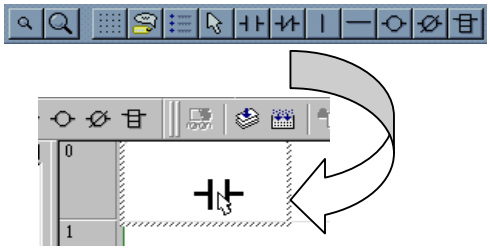


ظاهر صفحهٔ برنامه نویسی باز شده با توجه به موارد موجود در منوی Tools/Option ممکن است متفاوت باشد.



قسمت سایه زده شدهٔ آبی رنگ نشان دهندهٔ محل مکان نما در برنامه می باشد و قسمت طوسی رنگ واقع در سمت چپ جداکنندهٔ پله های برنامه می باشد.

می توان با استفاده از نوار ابزار برنامه نویسی یک کنتاکت باز را در صفحهٔ برنامه قرار داد. این کار با کلیک کردن روی عنصر مورد نظر و قرار دادن آن در صفحهٔ برنامه نویسی انجام می گیرد.



بعد از قرار دادن کنتاکت روی صفحه پنجره ای برای وارد کردن اطلاعات و آدرس کنتاکت باز می شود.

به این شیوه می توان همهٔ عناصر برنامه نویسی مانند کویل ها و یا دستورالعمل ها و یا رابط های افقی و عمودی را در صفحهٔ برنامه نویسی قرار داد.

آدرس دهی کنتاکتها و کویل ها

وقتی یک کنتاکت و یا کویل در صفحهٔ برنامه نویسی قرار داده مـی شـود بایـد آدرس مربـوط بـه آن مشـخص گردد. برای مشخص کردن کنتاکت یا کویل می توان از Symbol (نام) و یا آدرس دهی عـددی اسـتفاده کـرد. همچنین می توان برای المان مورد نظر توضیحی هم اضافه کرد.

آدرس دهی با استفاده از نام ها

CX-Programmer نرم افزاری است که بر پایهٔ نام عناصر برنامه نویسی پایه گذاری شده است. این امکان وجود دارد که برای هر کدام از ورودی ها و خروجی ها نامی تعریف کرد و بعد به جای استفاده از آدرس از نام ورودی و خروجی ها در برنامه نویسی استفاده کرد. برای وارد کردن عناصر برنامه نویسـی اسـتفاده از نـام آسـان تـر از استفاده از آدرس ها می باشد. نرم افزار CX-P به گونه ای طراحی شده است که با وارد کردن نام یک ورودی و یا خروجی آدرس آن نیز نمایش داده می شود.
نامهایی که برای ورودی/خروجی ها در نظر گرفته می شود باید تکراری نباشد یعنی نمی تـوان بـرای دو آدرس مختلف از یک نام استفاده کرد.
قوانین تعیین نام (مانند push_button_1) برای آدرس ها به شرح زیر می باشد:

- نام نمی تواند با یک عدد آغاز شود.
- نمی تواند از علایم نقطه گذاری در جملات تشکیل شده باشد. (به غیر از Underline)
- نمی تواند شامل Space باشد.
- و نباید به آدرس های PLC شباهت داشته باشد.

آدرس دهی با استفاده از آدرسهای ورودی/خروجی

در همهٔ برنامه ها برای مشخص کردن هویت ورودی و خروجی ها از نام و یا Symbol استفاده نمـی شـود. اگـر برای عناصر برنامه نویسی به کار برده شده در برنامه نامی تعریف نشده باشد می توان به راحتـی بـا اسـتفاده از آدرس هویت بیت و یا Word استفاده شده را مشخص نمود. وقتی این شیوه از آدرس دهی انتخـاب مـی شـود می توان برای هر کدام از عناصر تشکیل دهندهٔ برنامه توضیحی در قسمت Comment وارد کرد.

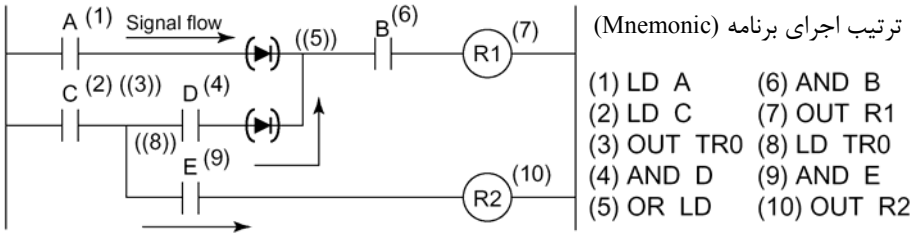
8-2- نوشتن برنامهٔ نردبانی

8-2-1- ایده های اولیهٔ برای نوشتن برنامه نردبانی

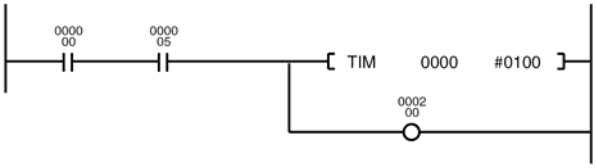
1. وقتی که برنامهٔ نردبانی توسط PLC اجرا می شود، تغذیهٔ مدار همواره از چـپ بـه راسـت جریـان مـی یابد، بدین معنی که برنامه از سمت چپ به راست توسط PLC اجرا می شود. برنامه هایی که نیـاز بـه اجرای از راست به چپ دارند را نمی توان به صـورت نردبـانی نوشـت. چگـونگی اجـرا برنامـه بـا اجـرا

مدارهای کنترل رله ای متفاوت است. به عنوان مثال زمانیکه مدار (a) در برنامهٔ PLC اجرا مـی گـردد، با در نظر گرفتن دیودهای فرضی در مسیر می توان طریقهٔ اجرا برنامه را مشاهده کـرد. در ایـن برنامـه ابتدا کنتاکت A و بعد کنتاکت C و بعد کنتاکت D توسط PLC خوانده می شود. ترتیب اجرای برنامـه را می توان در سمت راست به صورت زبان Mnemonic مشاهده کرد.

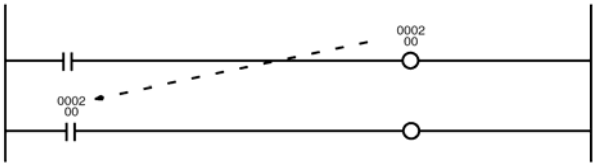
مدار (A)



2. هیچ محدودیتی در تعداد بیتهای ورودی/خروجی، بیتهای مجازی، تایمرها و دیگر بیتهای ورودی برای استفاده در یک پله از برنامه وجود ندارد. به هر حال پله های برنامه باید حتی الامکان ساده و قابل فهم باشند حتی اگر این امر موجب استفادهٔ بیشتر از بیتهای ورودی باشد.
3. هیچ محدودیتی در تعداد بیتهای استفاده شده به صورت سری و یا موازی در یک پله از برنامـه وجـود ندارد. یعنی به هر تعداد مورد نیاز می توان بیتها را در یک پله سری و یا موازی قرار داد.
4. دو ویا چندین خروجی را می توان به صورت موازی استفاده کرد.



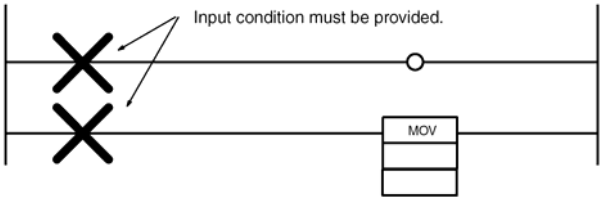
5. همچنان می توان از بیتهای خروجی به صورت ورودی در برنامه استفاده کرد.



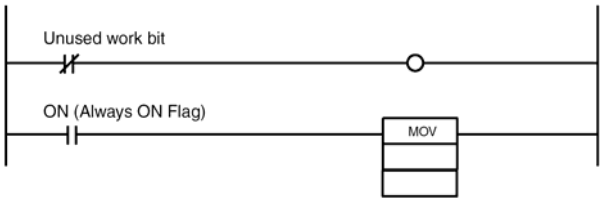
8-2-2- ممنوعیت های موجود در برنامه نویسی

1. برنامهٔ نردبانی باید بسته باشد از آنجایی که تغذیهٔ هر پله از سمت چپ به راست انتقال می یابد برنامـه باید از شین سمت چپ آغاز و به شین سمت چپ ختم شود. اگر پله ای از برنامه بسته نشود ایـن امـر به عنوان خطا در نظر گرفته شده و نمایش داده می شود.

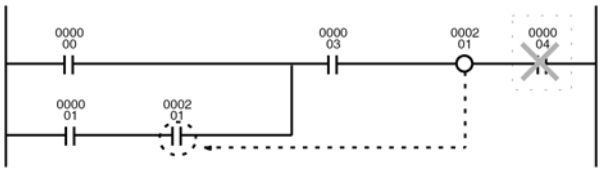
2. بیت های خروجی، تایمرها، کانترها و بقیهٔ دستورالعمل هایی که به عنوان خروجی استفاده می شود را نمی توان مستقیما به شین سمت چپ متصل کرد. اگر یکی از عناصر خروجی مستقیما به شین سمت چپ متصل شود، به عنوان خطا در نظر گرفته می شود.



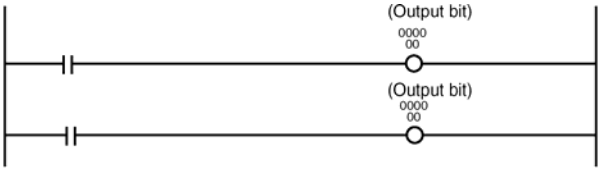
اگر باید ورودی برای این خروجی ها همیشه روشن باشد می توان از کنتاکت های بستهٔ بیتهای مجازی بدون استفاده و یا فلگ همیشه روشن(Always On Flag) استفاده کرد.



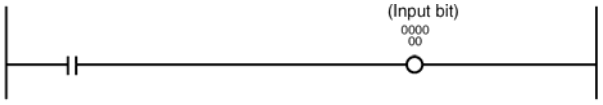
3. بیتهای ورودی باید همواره قبل از خروجی ها قرار گیرد و هرگز بعد از خروجی ها نمی توان از ورودی استفاده کرد. اگر ورودی بعد از خروجی قرار گیرد باعث ایجاد خطا در برنامه می گردد.



4. نمی توان از یک خروجی مانند یک کویل با یک آدرس بیشتر از یک بار استفاده کرد. از آنجایی که برنامهٔ نردبانی از بالا به پایین اجرا می شود، در نتیجه نهایتا حالت دستور خروجی واقع در پایین ترین پله ازبرنامه روی حالت خروجی تأثیر گذاشته و همهٔ دستورهای خروجی قبلی را هم تحت تأثیر قرار می دهد.



5. بیتهای ورودی را نمی توان در دستورهای خروجی و به عنوان خروجی استفاده کرد. حالت این بیتها وابسته به حالت ورودی هایی است که به صورت فیزیکی به ورودی PLC متصل می باشند و نمی توانند تحت تأثیر دستورهای خروجی قرار گیرند.



6. دستور العمل END(001) باید در انتهای هر برنامه در هر Task قرار داده شود. در غیر اینصورت خطای وجود نداشتن دستور END(001) ایجاد می گردد.

8-2-3- مدهای عملکرد PLC

مدهای عملکرد موجود در واحد CPU به شرح زیر می باشد.

1. مد برنامه (PROGRAM Mode)
2. مد نمایش (MONITOR Mode)
3. مد اجرا (RUN Mode)

مد برنامه (PROGRAM Mode)

اجرا برنامه در این مد متوقف می گردد، به همین دلیل نمایشگر RUN موجود روی CPU روشن نخواهد بود. این مد عملکرد برای ویرایش برنامه و همچنین عملیات آماده سازی PLC مانند موارد زیر به کار می رود:

- ایجاد جدول ورودی/خروجی (I/O Table)
- تغییر تنظیمات PLC و همچنین بقیهٔ تنظیمات
- انتقال و همچنین بررسی برنامه
- برای روشن و خاموش کردن به اجبار (Force) ورودی و خروجی های برنامه که معمولا برای بررسی سیم بندی استفاده می شود.

در این مد عملکرد، هیچ کدام از Task های برنامه اجرا نمی گردد، ولی تجدید حالت ورودی و خروجی ها توسط PLC انجام می گیرد.

مد نمایش (MONITOR Mode)

می توان با استفاده از تجهیزات برنامه نویسی مد عملکرد CPU را به مد نمایش تغییر داد. در این حالت نمایشگر RUN موجود روی CPU روشن خواهد بود. این مد برای RUN کردن آزمایشی برنامه و یا ایجاد تغییراتی که در زیر نامبرده شده به کار می رود.

- ویرایش برنامه در حالت Online
- روشن و خاموش کردن بیتها با اجبار (Force)
- ایجاد تغییرات در مقادیر حافظهٔ PLC

در این مد قسمتهای برنامه که باید در هر سیکل اجرا شوند توسط PLC اجرا می گردند.

مد اجرا (RUN Mode)

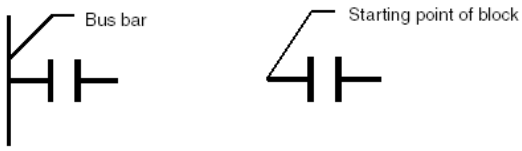
این مد عملکرد برای اجرا عادی برنامه توسط PLC به کار می رود. در این حالت نیز نمایشگر RUN موجود روی CPU روشن خواهد بود. مواردی همچون ویرایش برنامه، روشن و خاموش کردن بیتها با اجبار و همچنین ایجاد تغییرات در فضای حافظه در این مد عملکرد امکان پذیر نمی باشد. اما مواردی همچون نمایش اجرا برنامه و یا نمایش محتوای حافظه در این مد عملکرد ممکن می باشد.

9- دستورالعمل ها

9-1- دستور LOAD: LD

آغاز منطقی هر خط از برنامه با دستور LD می باشد که شرط عملکردی به صورت روشن یا خاموش است که از یکی از بیتهای حافظه تأثیر می پذیرد.

9-1-1- نماد در برنامهٔ نردبانی

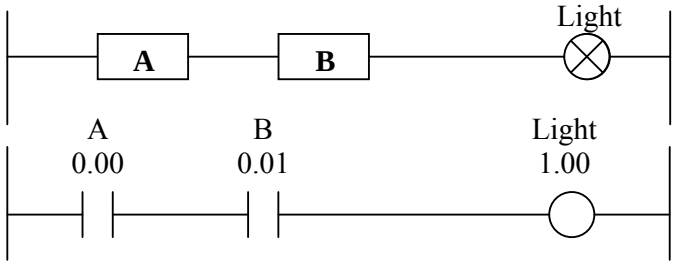


9-1-2- شرح عملکرد دستور LD

دستور LD به عنوان بیت معمولا باز آغازین یک خط از برنامه استفاده می شود. این بیت معمولا باز با توجه به آدرسی که به آن اختصاص داده شده است، حالت خود را با تأثیر از حالت آدرس تعیین شده در حافظه می پذیرد.

9-2- دستور AND

دو سوییچ که به صورت سری به یکدیگر متصل شده باشند مثال واضحی از دستور AND هستند. این عبارت منطقی به این صورت خوانده می شود: " اگر A و B هر دو روشن باشند، لامپ روشن می گردد."

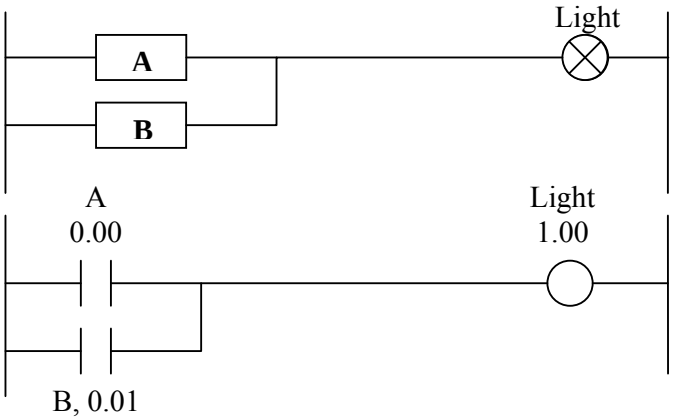


برنامه نویس باید به هر کدام از این ورودی و خروجی ها قبل از آغاز برنامه نویسی آدرسی اختصاص دهد.
باید به این نکته توجه کرد که سوییچ های A و B هر دو باید روشن باشند تا تغذیه لامپ تأمین شده و روشن گردد. این بدین معنی است که با خاموش کردن هر کدام از این سوییچ ها می توان لامپ را خاموش کرد.
برنامه نویس می تواند هر تعداد که لازم است دستور AND را برای فرمان دادن به یک خروجی استفاده کند.

A	B	AND
1	1	1
0	1	0
1	0	0
0	0	0

9-3- دستور OR

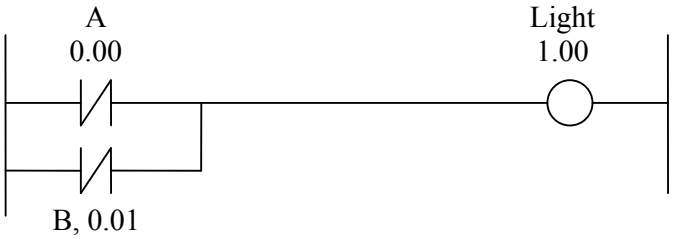
دو سوییچ که به صورت موازی به یکدیگر متصل شده باشند مثال واضحی از دستور OR هستند. این عبارت منطقی به این صورت خوانده می شود: "اگر A یا B (یا هردو) روشن باشند لامپ روشن می شود."
باید به این نکته توجه کرد که سوییچ های A وB هر دو می توانند به تنهایی لامپ را روشن کنند، اما برای خاموش کردن لامپ باید هر دو آنها خاموش باشند.



A	B	AND
1	1	1
0	1	1
1	0	1
0	0	0

9-4- دستور NOT

دستور NOT در واقع همان نماد سوییچ های معمولا بسته می باشد. اگر ورودی و یا خروجی به صورت بسته استفاده شود، در حالت خاموش یا "0" عکس العمل نشان می دهد. در این صورت باید ورودی و یا بیت مورد نظر "0" باشد تا جریان از اتصال معمولا بسته عبور کند. اگر یک بیت که به صورت اتصال بسته در برنامه استفاده شده است، "1" گردد این اتصال مانع عبور جریان می گردد.



در این مثال خروجی روشن باقی می ماند، تا زمانی که هر دو ورودی A و B روشن شود.

A	NOT
1	0
0	1

9-5- دستور تایمر:TIM

دستور TIM به صورت یک تایمر تأخیر در وصل (On-Delay) وکاهشی با واحد 0.1S عمل می کند، محـدودهٔ زمانی مشخص شده به عنوان SV برای تایمر از 0 تا 999.9 ثانیه است و دقت تایمر 0.1S می باشد.

9-5-1- نماد تایمر در برنامهٔ نردبانی

روش عملکرد	نماد	عملگرها
BCD	TIM N S	N : 0 تا 4095 (دسیمال) S : #0 تا #9999 (BCD)

9-5-2- عملگرهای دستور تایمر

N: شماره تایمر که باید بین 0 تا 4095 انتخاب شود.

S: Set Value (زمان تایمر)

مقدار Set Value باید بین #0000 تا #9999 به صـورت BCD انتخـاب شـود. (اگـر SV تـایمر #0000 درنظـر گرفته شود Flag اتمام کار تایمر به محض اجرا دستور تایمر روشن خواهد شد).

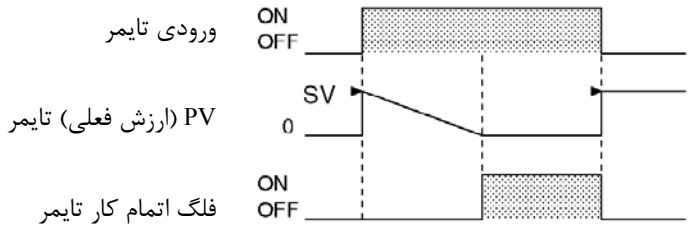
9-5-3- مشخصات عملگرهای تایمر

فضای حافظه	N	S
CIO Area	---	CIO 0000 to CIO 6143
Work Area	---	W000 to W511
Holding Bit Area	---	H000 to H511
Auxiliary Bit Area	---	A000 to A959
Timer Area	0000 to 4095	T0000 to T4095
Counter Area	---	C0000 to C4095
DM Area	---	D00000 to D32767
EM Area without bank	---	E00000 to E32767
EM Area with bank	---	En_00000 to En_3276 (n = 0 to C)
Indirect DM/EM addresses in binary	---	@ D00000 to @ D32767 @ E00000 to @ E32767 @ En_00000 to@ En_32767 (n = 0 to C)
Indirect DM/EM addresses in BCD	---	*D00000 to *D32767 *E00000 to *E32767 *En_00000 to *En_032767(n = 0 to C)
Constants	---	BCD: #0000 to 9999 (BCD) “&” cannot be used. Binary: &0 to &65535 (decimal) #0000 to #FFFF (hex)
Data Registers	---	DR0 to DR15
Index Registers	---	---
Indirect addressing using Index Registers	,IR0 to ,IR15 –2048 to +2047 ,IR0 to –2048 to +2047 ,IR15 DR0 to DR15, IR0 to IR15	

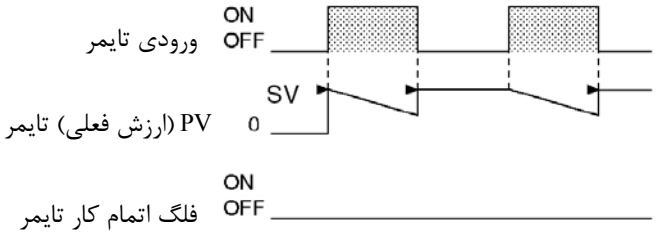
9-5-4- شرح عملکرد تایمر

زمانیکه ورودی تایمر خاموش است، تایمر مشخص شده با شمارهٔ N به حالت اولیه بازگردانده می شود، در این حالت مقدار PV(Present Value) تایمر با مقدار SV(Set Value) یکی شده و فلگ اتمام کار تایمر خاموش می شود. هنگامیکه ورودی تایمر از خاموش به روشن تغییر حالت می دهد، TIM شروع به کاهش مقدار PV می کند. این کاهش مقدار PV تا زمانیکه ورودی تایمر روشن باقی بماند ادامه دارد و به محض اینکه مقدار PV برابر با 0000 شود فلگ اتمام کار تایمر روشن خواهد شد.

بعد از این حالت PV و فلگ اتمام کار تایمر بعد از اتمام زمان مورد نظر به همان حالت باقی خواهد ماند. برای بازگرداندن تایمر به حالت اولیه برای محاسبهٔ دوبارهٔ زمان باید ورودی تایمر را خاموش و دوباره روشن کرد و یا اینکه مقدار PV تایمر را به یک مقدار غیرصفر تبدیل کرد (به عنوان مثال با استفاده از دستور MOV(021)).



نمودار زمانی زیر وضعیت PV (ارزش فعلی) و فلگ اتمام کار تایمر، هنگامی که ورودی تایمر قبل از به پایان رسیدن زمان تایمر خاموش شود را نشان می دهد.



9-5-5- فلگهای دستور تایمر

نام	نشانه	عملکرد
Error Flag	ER	این فلگ زمانی که N به صورت غیرمستقیم توسط حافظهٔ IR آدرس دهی شود ولی آدرس مورد نظر محتوای مقدار مجازی برای N نباشد روشن می شود. زمانی که در حالت BCD، S شامل دادهٔ BCD نباشد، این فلگ روشن می شود. در بقیهٔ موارد این فلگ خاموش است.
Equals Flag	=	خاموش یا بدون تغییر
Negative Flag	N	خاموش یا بدون تغییر

9-5-6- نکات قابل توجه در مورد دستورالعمل تایمر

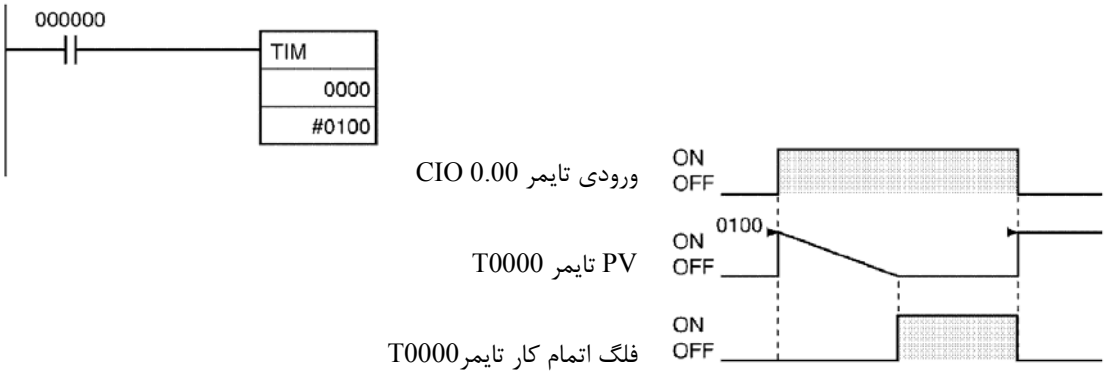
محدودهٔ مشخص شده برای شمارهٔ تایمر، برای همهٔ دستورات تایمر که شامل دستورات TIM، TIMX(550)، TIMH(015)، TIMHX(551)، TMHH(540)، TMHHX(552)، TTIM(087)، TTIMX(555)، TIMW(813)، TIMWX(816)، TMHW(815)، TMHWX(817) می شود یکسان است. این بدان معنی است که اگر دو تایمر مختلف با یک شمارهٔ تایمر در برنامه نویسی استفاده شود، ولی همزمان به کاربرده نشوند، در هنگام بررسی برنامه توسط نرم افزار، خطایی موسوم به خطای تکرار (Duplicate Error) ایجاد می شود، در این حالت تایمرها به طور معمول عمل خواهندکرد. ولی در صورتیکه دو تایمر با یک شمارهٔ تایمر به صورت همزمان به کار برده شود عملکرد تایمرها صحیح نخواهد بود.

در صورتی که زمان دورهٔ (Cycle Time) عملکرد PLC از 80ms تجاوز کند، عملکرد تایمرهای از شمارهٔ 2048 تا 4095 صحیح نمی باشد. در این حالت باید از تایمرهای شمارهٔ 0000 تا 2047 استفاده کرد.

9-5-7- مثال از دستور تایمر

در این مثال زمانیکه ورودی تایمر با آدرس CIO 0.00 از حالت خاموش به روشن تغییر حالت می دهد، مقدار PV تایمر شروع به کاهش از مقدار SV می کند، زمانیکه مقدار PV تایمر به 0000 برسد فلگ اتمام کار تایمر شمارهٔ صفر، آدرس T0000، روشن می شود.

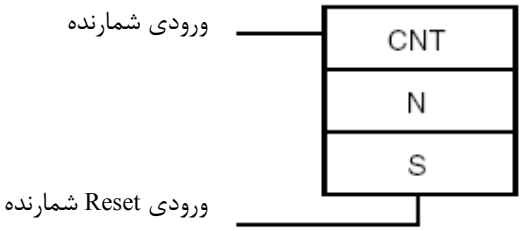
ولی زمانیکه CIO 0.00 خاموش می شود مقدار PV به مقدار SV بازگردانده شده و فلگ اتمام کار تایمر نیز خاموش می شود.



9-6- دستور کانتر/شمارنده:CNT

دستور CNT به صورت یک شمارندهٔ کاهشی عمل می کند. محدودهٔ شمارش برای کانتر بین 0000 تا 9999 می باشد.

9-6-1- نماد کانتر در برنامه نویسی نردبانی



N : شمارهٔ شمارنده

S : مقدار شمارنده (Set Value)

9-6-2- عملگرهای دستور شمارنده

N: شمارهٔ کانتر

شمارهٔ کانتر باید بین 0000 تا 4095 به صورت دسیمال انتخاب شود.

S: (Set Value)

مقدار شمارش توسط کانتر که بایستی بین #0000 تا #9999 و به صورت BCD تعیین گردد.

9-6-3- مشخصات عملگرهای شمارنده

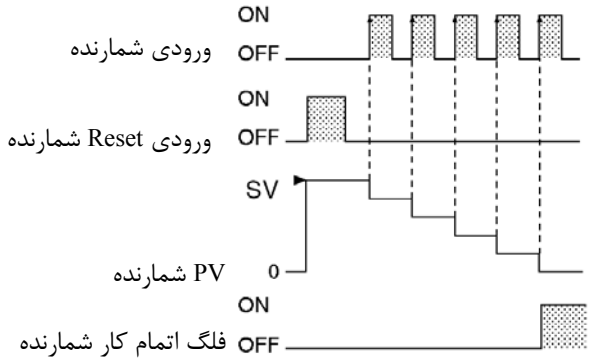
فضای حافظه	N	S
CIO Area	---	CIO 0000 to CIO 6143
Work Area	---	W000 to W511
Holding Bit Area	---	H000 to H511
Auxiliary Bit Area	---	A000 to A959
Timer Area	0000 to 4095	T0000 to T4095
Counter Area	---	C0000 to C4095
DM Area	---	D00000 to D32767
EM Area without bank	---	E00000 to E32767
EM Area with bank	---	En_00000 to En_32767 (n = 0 to C)
Indirect DM/EM addresses in binary	---	@ D00000 to @ D32767 @ E00000 to @ E32767 @ En_00000 to@ En_32767 (n = 0 to C)
Indirect DM/EM addresses in BCD	---	*D00000 to *D32767 *E00000 to *E32767 *En_00000 to *En_032767(n = 0 to C)
Constants	---	BCD: #0000 to 9999 (BCD) “&” cannot be used. Binary: &0 to &65535 (decimal) #0000 to #FFFF (hex)
Data Registers	---	DR0 to DR15
Index Registers	---	---
Indirect addressing using Index Registers	,IR0 to ,IR15 –2048 to +2047 ,IR0 to –2048 to +2047 ,IR15 DR0 to DR15, IR0 to IR15	

9-6-4- شرح عملکرد شمارنده

با هر بار روشن و خاموش شدن ورودی شمارنده مقدار PV شمارنده یک واحد BCD کاهش می یابد. زمانی که PV شمارنده برابر با عدد 0000 شود، فلگ اتمام کار شمارنده به منظور نشان دادن اتمام شمارش روشن می گردد.

پس از روشن شدن فلگ اتمام کار شمارنده، با فعال کردن پایهٔ Reset ، شمارنده قابلیت شمارش مجدد را به دست می آورد. در غیر اینصورت قادر به شمارش مجدد نخواهد بود.

زمانی که ورودی Reset شمارنده روشن باشد، ورودی اصلی نادیده گرفته میشود بدین منظور که حتی در صورت تولید پالس توسط ورودی، شمارش صورت نمی گیرد. (زمانی که کانتر Reset می شود مقدار PV برابر با مقدار SV شده و فلگ اتمام کار شمارنده نیز خاموش می گردد.)



9-6-5- فلگهای دستور کانتر

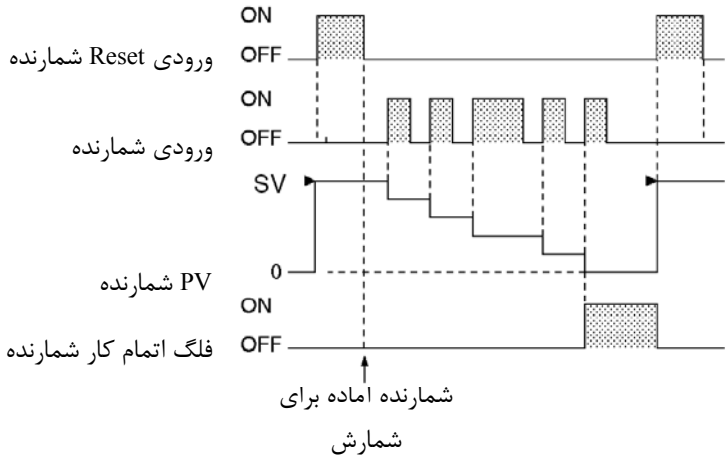
نام	نشانه	عملکرد
Error Flag	ER	این فلگ زمانی که N به صورت غیرمستقیم توسط حافظهٔ IR آدرس دهی شود ولی آدرس مورد نظر محتوای مقدار مجازی برای N نباشد روشن می شود. زمانی که در حالت BCD، S محتوای دادهٔ BCD نباشد، این فلگ روشن می شود. در بقیهٔ موارد این فلگ خاموش است.
Equals Flag	=	خاموش یا بدون تغییر
Negative Flag	N	خاموش یا بدون تغییر

9-6-6- نکات قابل توجه در مورد دستورالعمل کانتر

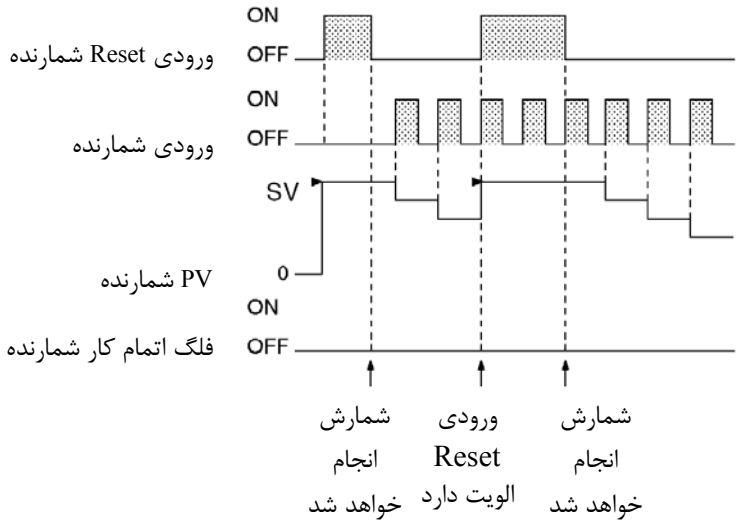
محدودهٔ مشخص شده برای شمارهٔ کانتر، برای همهٔ دستورات کانتر که شامل دستورات CNT، CNTX(546)، CNTR(012)، CNTRX(548)، CNTW(814)، CNTWX(818) می شود یکسان است. این بدان معنی است که اگر دو کانتر مختلف با یک شمارهٔ کانتر در برنامه نویسی استفاده شود، ولی همزمان به کاربرده نشوند، در هنگام بررسی برنامه توسط نرم افزار، خطایی موسوم به خطای تکرار (Duplicate Error) ایجاد می شود، در این حالت کانترها به طور معمول عمل خواهندکرد. ولی در صورتیکه دو کانتر با یک شمارهٔ کانتر به صورت همزمان به کار برده شود عملکرد کانترها صحیح نخواهد بود.

مقدار PV کانتر توسط PLC در هر زمانی که ورودی کانتر از حالت خاموش به روشن تغییر وضعیت می دهد، بررسی می شود ولی حالت فلگ اتمام کار کانتر در هربار اجرا برنامه توسط PLC (Cycle Time) بررسی می گردد. فلگ اتمام کار کانتر زمانی که PV شمارنده برابر 0000 باشد روشن و زمانیکه PV شمارنده مخالف 0000 است خاموش می شود.

باید از Reset شدن کانتر با تغییر وضعیت ورودی Reset آن ازخاموش ← روشن ← خاموش قبل از شروع شمارش اطمینان حاصل نمود. در صورت روشن بودن ورودی Reset شمارنده، حالت ورودی اصلی آن در نظر گرفته نخواهد شد و شمارشی توسط کانتر انجام نمی پذیرد.



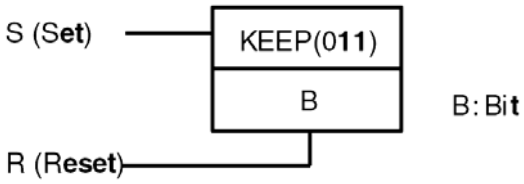
ورودی Reset شمارنده به ورودی اصلی آن الویت دارد، یعنی در صورتیکه هر دو پایهٔ ورودی کانتر روشن باشد کانتر Reset می شود. (مقدار PV کانتر با مقدار SV برابر شده و فلگ اتمام کار کانتر نیز خاموش می گردد.)



9-7- دستور Keep : KEEP(011)

دستور KEEP(011) در برنامه مانند یک رلهٔ خود نگهدار عمل می کند.

9-7-1- نماد دستور KEEP(011) در برنامه نویسی نردبانی

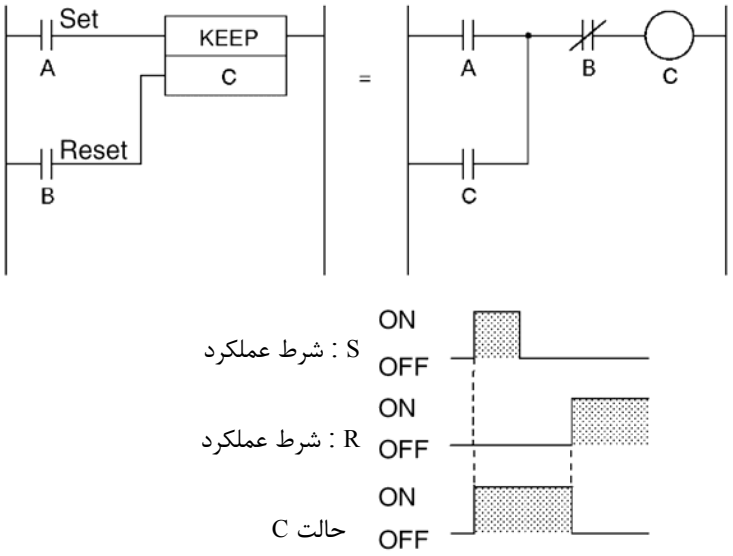


9-7-2- مشخصات عملگر دستور KEEP(011)

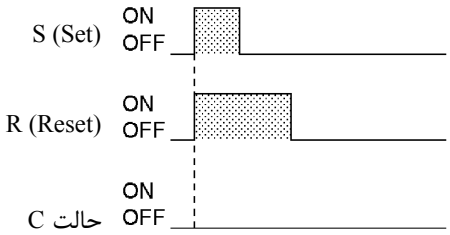
فضای حافظه	B
CIO Area	CIO 0000.00 to CIO 6143.15
Work Area	W000.00 to W511.15
Holding Bit Area	H000.00 to H511.15
Auxiliary Bit Area	A448.00 to A959.15
Timer Area	---
Counter Area	---
DM Area	---
EM Area without bank	---
EM Area with bank	---
Indirect DM/EM addresses in binary	---
Indirect DM/EM addresses in BCD	---
Constants	---
Data Registers	---
Index Registers	---
Indirect addressing using Index Registers	,IR0 to ,IR15 –2048 to +2047 ,IR0 to –2048 to +2047 ,IR15 DR0 to DR15, IR0 to IR15 ,IR0+(++) to ,IR15+(++) ,–(– –) IR0 to, –(– –) IR15

9-7-3- شرح عملکرد دستور KEEP(011)

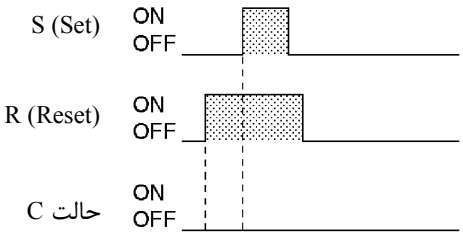
زمانیکه ورودی S(Set) روشن می شود بیت مورد نظر از حافظـه نیـز روشـن مـی شـود، ولـی زمانیکـه ورودی R(Reset) روشن شود، بدون در نظر گرفتن روشن یا خاموش بودن ورودی S(Set) باعث خـاموش شـدن بیـت تعیین شده می گردد. این بدان معنی است که با روشن شدن ورودی R(Reset) بیت مشخص شـده در دسـتور خاموش می گردد. در نمودار زیر رابطهٔ بین شرط عملکرد و دستور KEEP(011) نشان داده شده است.



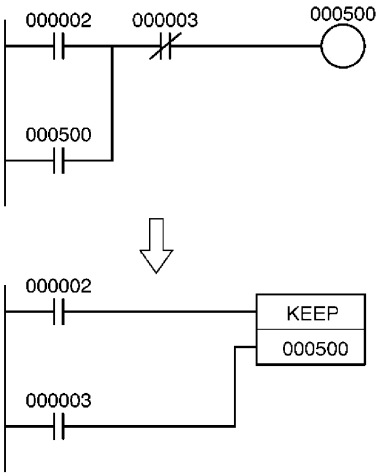
در صورتی که هر دو ورودی Set و Reset به طور همزمان روشن شوند الویت با ورودی Reset خواهد بود، یعنی بیت تعیین شده خاموش می شود.



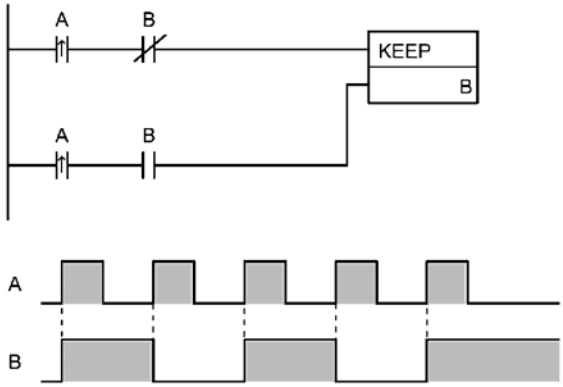
در صورتی که ورودی Reset روشن باشد حالت ورودی Set در نظر گرفته نمی شود.



دستور KEEP(011) مانند یک بیت خودنگهدار عمل می کند، با این تفاوت که نوشتن برنامه با استفاده از خود نگهدار به جای استفاده از دستور KEEP(011) نیاز به استفاده از یک دستور بیشتر دارد.



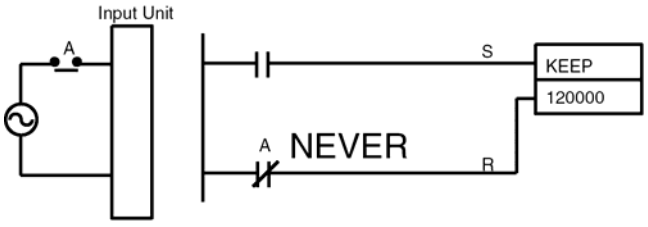
همچنین همانطور که در زیر مشاهده می شود، می توان از دستور KEEP(011) در ایجاد Flip-Flop نیز استفاده کرد:



اگر بیت تعیین شده در برنامه (B) از یکی از حافظه های پایدار PLC انتخاب شود، این بیت حتی در صورت ایجاد وقفه در تغذیهٔ PLC نیز حالت خود را حفظ خواهد کرد. در نتیجه از دستور KEEP(011) برای تعریف بیت های پایداری که باید حالت آنها در صورت ایجاد وقفه در عملکرد PLC حفظ شود استفاده می گردد.

9-7-4- نکات قابل توجه در مورد دستورالعمل KEEP(011)

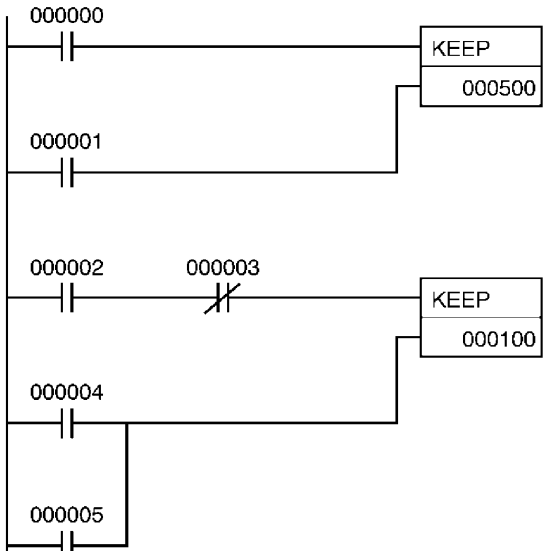
در صورتی که یکی از ورودیهای PLC از تغذیهٔ AC استفاده می کند هرگز نبایدکنتاکت بستهٔ این ورودی به عنوان ورودی R(Reset) دستور KEEP(011) استفاده شود. زیرا تأخیر موجود در خاموش شدن منبع تغذیهٔ DC به کار رفته در PLC باعث خاموش شدن بیت تعیین شده در دستور KEEP(011) می شود. این وضعیت در شکل زیر نشان داده شده است.



9-7-5- مثال از دستور KEEP(011)

در این مثال با روشن شدن ورودی دستور CIO 0.00 بیت CIO 5.00 روشن خواهد شد. بیت CIO 5.00 تا روشن شدن ورودی CIO 0.01 روشن باقی خواهد ماند.

در این مثال در صورتیکه CIO 0.02 روشن و CIO 0.03 خاموش باشد بیت CIO 1.00 روشن خواهد شد. CIO 1.00 تا زمانیکه یکی از بیتهای CIO 0.04 و یا CIO 0.05 روشن شود، روشن باقی خواهد ماند.

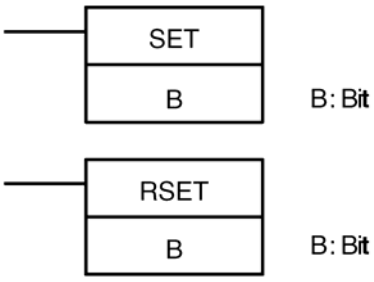


9-8- دستورهای Set و Reset : SET و RESET

زمانیکه شرط عملکرد دستور SET روشن شود، بیت عملگر تعیین شده روشن می گردد.

زمانیکه شرط عملکرد دستور RESET روشن شود، بیت عملگر تعیین شده خاموش می گردد.

9-8-1- نماد دستور SET و RESET در برنامه نویسی نردبانی

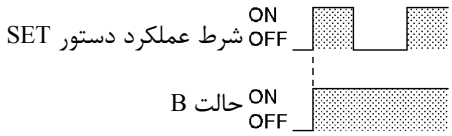


9-8-2- مشخصات عملگر دستورSET و RESET

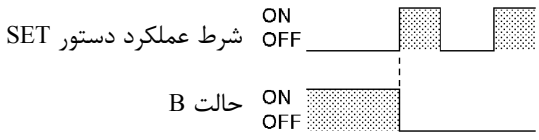
فضای حافظه	B
CIO Area	CIO 0000.00 to CIO 6143.15
Work Area	W000.00 to W511.15
Holding Bit Area	H000.00 to H511.15
Auxiliary Bit Area	A448.00 to A959.15
Timer Area	---
Counter Area	---
DM Area	---
EM Area without bank	---
EM Area with bank	---
Indirect DM/EM addresses in binary	---
Indirect DM/EM addresses in BCD	---
Constants	---
Data Registers	---
Index Registers	---
Indirect addressing using Index Registers	,IR0 to ,IR15 –2048 to +2047 ,IR0 to –2048 to +2047 ,IR15 DR0 to DR15, IR0 to IR15 ,IR0+(++) to ,IR15+(++) ,–(– –) IR0 to, –(– –) IR15

9-8-3- شرح عملکرد دستور SET و RESET

زمانیکه شرط عملکرد دستور SET روشن شود، این دستور، بیت تعیین شده به عنوان عملگر را روشن می کند، ولی اگر شرط عملکرد دستور خاموش باشد تأثیری در حالت عملگر نخواهد داشت. برای خاموش کردن بیتی که توسط دستور SET روشن شده است، باید از دستور RESET استفاده کرد.



زمانیکه شرط عملکرد دستور RESET روشن شود، این دستور، بیت تعیین شده به عنوان عملگر را خاموش می کند، ولی اگر شرط عملکرد دستور خاموش باشد تأثیری در حالت عملگر نخواهد داشت. برای روشن کردن بیتی که توسط دستور RESET خاموش شده است، باید از دستور SET استفاده کرد.



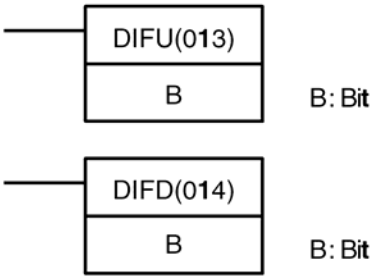
9-8-4- نکات قابل توجه در مورد دستورالعملهای SET و RESET

دستورهای SET و RESET برای Set کردن و یا Reset کردن تایمرها و کانترها استفاده نمی شوند.

9-9- دستورهای آشکارساز لبهٔ بالاروند/پایین رونده : DIFU(013) و DIFD(014)

دستور DIFU(013)، بیت تعریف شده را زمانیکه شرط عملکرد دستور از حالت خاموش به روشن تغییر کند (لبهٔ بالارونده)، به اندازهٔ یک زمان دورهٔ برنامه (یک سیکل) روشن می کند.
دستور DIFD(014) بیت تعریف شده را زمانیکه شرط عملکرد دستور از حالت روشن به خاموش تغییر کند (لبهٔ پایین رونده)، به اندازهٔ یک زمان دورهٔ برنامه (یک سیکل) روشن می کند.

9-9-1- نماد دستور DIFU(013) و DIFD(014) در برنامه نویسی نردبانی

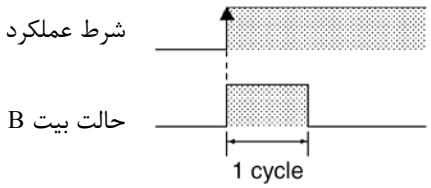


9-9-2- مشخصات عملگر دستورهای DIFU(013) و DIFD(014)

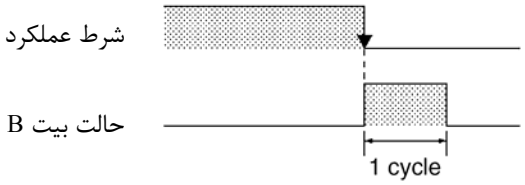
فضای حافظه	B
CIO Area	CIO 0000.00 to CIO 6143.15
Work Area	W000.00 to W511.15
Holding Bit Area	H000.00 to H511.15
Auxiliary Bit Area	A448.00 to A959.15
Timer Area	---
Counter Area	---
DM Area	---
EM Area without bank	---
EM Area with bank	---
Indirect DM/EM addresses in binary	---
Indirect DM/EM addresses in BCD	---
Constants	---
Data Registers	---
Index Registers	---
Indirect addressing using Index Registers	,IR0 to ,IR15 –2048 to +2047 ,IR0 to –2048 to +2047 ,IR15 DR0 to DR15, IR0 to IR15 ,IR0+(++) to ,IR15+(++) ,–(– –) IR0 to, –(– –) IR15

9-9-3- شرح عملکرد دستورهای DIFU(013) و DIFD(014)

زمانیکه شرط عملکرد دستور DIFU(013) از خاموش به روشن تبدیل شود، این دستور بیت B را روشن می کند. در سیکل بعدی زمانیکه برنامه به دستور DIFU(013) رسید، بیت B خاموش خواهد شد.



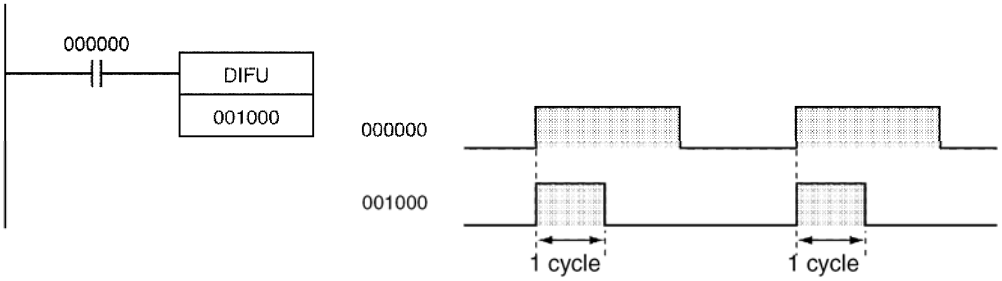
زمانیکه شرط عملکرد دستور DIFD(014) از روشن به خاموش تبدیل شود، این دستور بیت B را روشن می کند. در سیکل بعدی زمانیکه برنامه به دستور DIFD(014) رسید، بیت B خاموش خواهد شد.



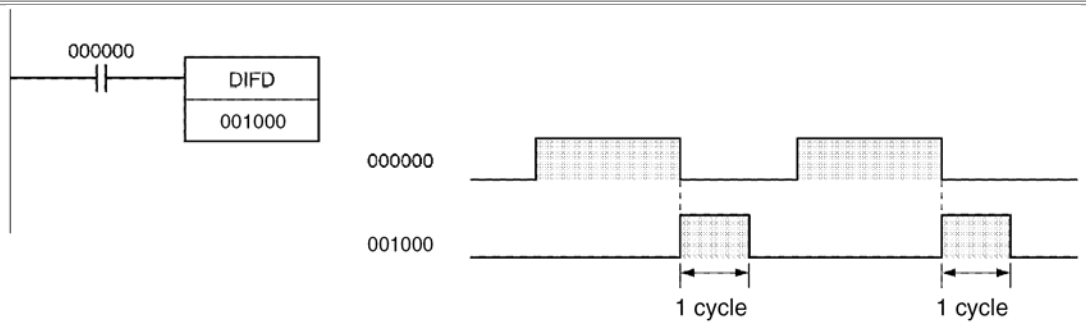
دستورهای UP(521) و DOWN(522) نیز برای اجرا یک دستور به اندازهٔ یک سیکل، زمانی که شرط عمکرد آن از روشن ⟵ خاموش ویا از خاموش ⟵ روشن تغییر می کند، استفاده می شود.

9-9-4- مثال از عملکرد دستورهای DIFU(013) و DIFD(014)

در مثال زیر زمانی که CIO 0.00 از حالت خاموش به روشن تبدیل می شود، CIO 10.00 به اندازهٔ یک سیکل روشن می شود.



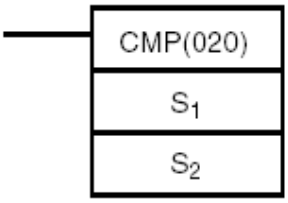
همچنین در مثال زیر زمانی که CIO 0.00 از حالت روشن به خاموش تبدیل می شود، CIO 10.00 به اندازهٔ یک سیکل روشن می شود.



9-10- دستور مقایسه کننده : CMP(020)

دستور CMP(020) دو دادهٔ باینری بدون علامت را، که این داده ها می توانند مقدار ثابت یا محتوای Word های حافظه باشد، با هم مقایسه و نتیجهٔ این مقایسه را به صورت فلگهای محاسباتی که در منطقهٔ Auxiliary حافظه وجود دارد نمایش می دهد.

9-10-1- نماد دستور CMP(020) در برنامه نویسی نردبانی



S1: داده مقایسه ای اول
S2: داده مقایسه ای دوم

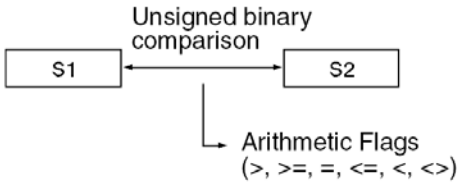
9-10-2- مشخصات عملگرهای دستور CMP(020)

فضای حافظه	S1	S2
CIO Area	CIO 0000 to CIO 6143	
Work Area	W000 to W511	
Holding Bit Area	H000 to H511	
Auxiliary Bit Area	A000 to A959	
Timer Area	T0000 to T4095	
Counter Area	C0000 to C4095	
DM Area	D00000 to D32767	
EM Area without bank	E00000 to E32767	
EM Area with bank	En_00000 to En_32767 (n = 0 to C)	
Indirect DM/EM addresses in binary	@ D00000 to @ D32767 @ E00000 to @ E32767 @ En_00000 to@ En_32767 (n = 0 to C)	
Indirect DM/EM addresses in BCD	*D00000 to *D32767 *E00000 to *E32767 *En_00000 to *En_032767(n = 0 to C)	
Constants	#0000 to #FFFF (binary)	
Data Registers	DR0 to DR15	
Index Registers	---	
Indirect addressing	,IR0 to ,IR15	

using Index Registers	–2048 to +2047 ,IR0 to –2048 to +2047 ,IR15 DR0 to DR15, IR0 to IR15 ,IR0+(++) to ,IR15+(++) ,–(– –) IR0 to, –(– –) IR15

9-10-3- شرح عملکرد دستور CMP(020)

دستور CMP(020) دو دادهٔ باینری بدون علامت S1 و S2 را با هم مقایسه کرده و نتیجه را به صورت فلگهای محاسباتی (فلگهای بزرگتر از، بزرگتر یا مساوی، مساوی، کوچکتر یا مساوی، کوچکتر از و نامساوی) در منطقهٔ Auxiliary حافظه، نشان می دهد.



9-10-4- وضعیت فلگهای محاسباتی

جدول زیر حالت فلگهای محاسباتی را بعد از اجرا دستور CMP(020) نشان می دهد.

نتیجهٔ CMP(020)	حالت فلگهای محاسباتی					
	>	>=	=	<=	<	<>
S1>S2	ON	ON	OFF	OFF	OFF	ON
S1=S2	OFF	ON	ON	ON	OFF	OFF
S1<S2	OFF	OFF	OFF	ON	ON	ON

9-10-5- نکات قابل توجه در مورد دستورالعمل CMP(020)

بین دستورالعمل CMP(020) و فلگهای محاسباتی نتیجهٔ آن که به عنوان ورودی در برنامه استفاده می شود، دستورالعمل دیگری را قرار ندهید، زیرا ممکن است که حالت این فلگها تحت تاثیر دستورات دیگر قرار گیرد.

9-11- دستور افزایش دهندهٔ BCD : ++B(594)

دستور ++B(594) محتوای یک Word که از چهار رقم BCD تشکیل می شود را یک واحد افزایش می دهد.

9-11-1- نماد دستور ++B(594) در برنامه نویسی نردبانی :

Wd: آدرس یک Word از حافظه

9-11-2- مشخصات عملگرهای دستور ++B(594)

فضای حافظه	Wd
CIO Area	CIO 0000 to CIO 6143
Work Area	W000 to W511

Holding Bit Area	H000 to H511
Auxiliary Bit Area	A484 to A959
Timer Area	T0000 to T4095
Counter Area	C0000 to C4095
DM Area	D00000 to D32767
EM Area without bank	E00000 to E32767
EM Area with bank	En_00000 to En_32767 (n = 0 to C)
Indirect DM/EM addresses in binary	@ D00000 to @ D32767 @ E00000 to @ E32767 @ En_00000 to@ En_32767 (n = 0 to C)

فضای حافظه	**Wd**
Indirect DM/EM addresses in BCD	*D00000 to *D32767 *E00000 to *E32767 *En_00000 to *En_032767 (n = 0 to C)
Constants	---
Data Registers	DR0 to DR15
Index Registers	---
Indirect addressing using Index Registers	,IR0 to ,IR15 –2048 to +2047 ,IR0 to –2048 to +2047 ,IR15 DR0 to DR15, IR0 to IR15 ,IR0+(++) to ,IR15+(++) ,–(– –) IR0 to, –(– –) IR15

9-11-3- شرح عملکرد دستور B(594)++

دستورالعمل B(594)++ یک واحد BCD به محتوای Wd اضافه می کند. تا زمانیکه شرط عملکرد دستور افزایش دهنده روشن است به ازاء هر سیکل برنامه محتوای Word تعیین شده یک واحد BCD افزایش می یابد. زمانیکه حالت لبهٔ بالارونده این دستورالعمل (@B(594)++) استفاده می شود، محتوای Word تعیین شده فقط زمانیکه ورودی دستور از خاموش به روشن (لبهٔ بالا رونده) تغییر وضعیت میدهد، یک واحد BCD افزایش می یابد.

با تغییر دستور B(594)++ به دستور @B(594)++ (با اضافه کردن @ به ابتدای دستور)، دستور به ازاء هر بار تغییر حالت شرط عملکرد از خاموش به روشن (لبهٔ بالارونده) یک واحد BCD محتوای Word تعیین شده را افزایش می دهد.

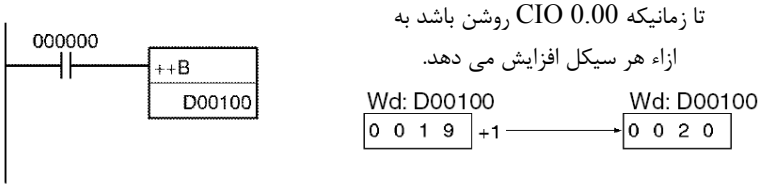
در هنگام استفاده از این دستور زمانیکه محتوای Word تعیین شده مساوی 0000 شود فلگ مساوی (P_EQ) و زمانیکه یکی از رقمهای BCD از 9 به 0 تبدیل شود فلگ Carry (P_Carry) روشن خواهد شد. و هر دو این فلگها زمانیکه محتوای Word از 9999 به 0000 تبدیل شود، روشن خواهند شد.

9-11-4- نکات قابل توجه در مورد دستورالعمل B(594)++

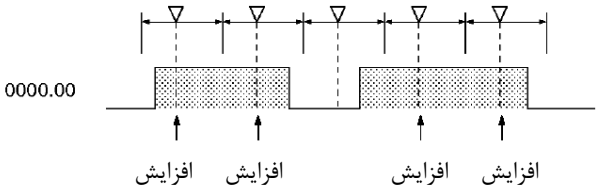
محتوای Wd باید به صورت BCD باشد در غیر اینصورت یعنی اگر محتوای Wd به صورت BCD نباشد، خطا ایجاد شده و فلگ خطا برای نشان دادن این خطا روشن خواهد شد.

9-11-5- مثال از عملکرد دستور B(594)++

در مثال زیر محتوای D100، تا زمانیکه CIO 0.00 روشن باشد به ازاء هر سیکل یک واحد BCD افزایش می یابد.



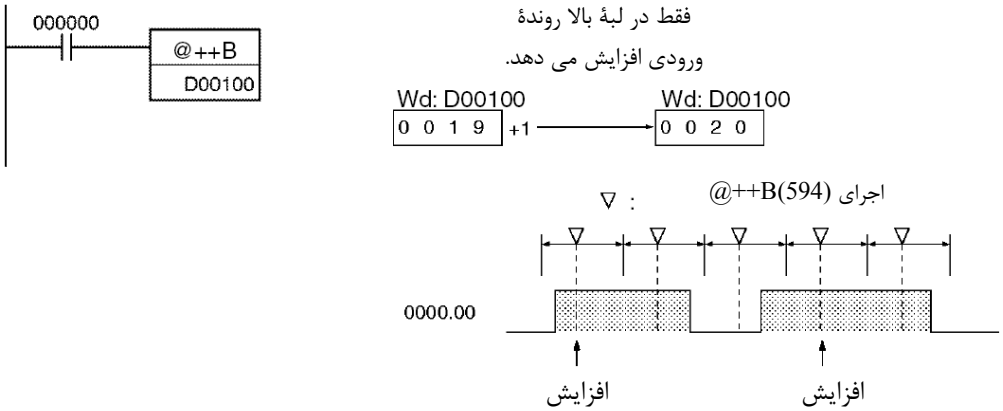
++B(594) □ □□□



9-11-6-

مثال از

عملکرد دستور @++B(594)

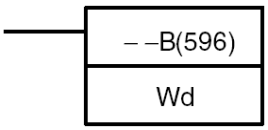


در این مثال که از حالت لبهٔ بالارونده دستور افزایش دهنده استفاده شده است، محتوای D100 فقط زمانی که CIO 0.00 از حالت خاموش به روشن تبدیل می شود یعنی یک لبهٔ بالارونده ایجاد می کند یک واحد BCD افزایش پیدا خواهد کرد.

9-12- دستور کاهش دهندهٔ BCD : B(596)--

دستور B(596)-- محتوای یک Word که از چهار رقم BCD تشکیل می شود را یک واحد کاهش می دهد.

9-12-1- نماد دستور B(596)-- در برنامه نویسی نردبانی



Wd: آدرس یک Word از حافظه

9-12-2- مشخصات عملگرهای دستور B(596)--

Wd	فضای حافظه

CIO Area	CIO 0000 to CIO 6143
Work Area	W000 to W511
Holding Bit Area	H000 to H511
Auxiliary Bit Area	A484 to A959
Timer Area	T0000 to T4095
Counter Area	C0000 to C4095
DM Area	D00000 to D32767
EM Area without bank	E00000 to E32767
EM Area with bank	En_00000 to En_32767 (n = 0 to C)
Indirect DM/EM addresses in binary	@ D00000 to @ D32767 @ E00000 to @ E32767 @ En_00000 to@ En_32767 (n = 0 to C)
Indirect DM/EM addresses in BCD	*D00000 to *D32767 *E00000 to *E32767 *En_00000 to *En_032767 (n = 0 to C)

فضای حافظه	**Wd**
Constants	---
Data Registers	DR0 to DR15
Index Registers	---
Indirect addressing using Index Registers	,IR0 to ,IR15 –2048 to +2047 ,IR0 to –2048 to +2047 ,IR15 DR0 to DR15, IR0 to IR15 ,IR0+(++) to ,IR15+(++) ,–(– –) IR0 to, –(– –) IR15

9-12-3- شرح عملکرد دستور B(596)--

دستورالعمل B(596)-- یک واحد BCD محتوای Wd را کاهش می دهد. تا زمانیکه شرط عملکرد دستور کاهش دهنده روشن است به ازاء هر سیکل برنامه محتوای Word تعیین شده یک واحد BCD کاهش می یابد. زمانیکه حالت لبهٔ بالارونده این دستورالعمل (@B(596)--) استفاده می شود، محتوای Word تعیین شده فقط زمانیکه ورودی دستور از خاموش به روشن (لبهٔ بالا رونده) تغییر وضعیت میدهد، یک واحد BCD کاهش می یابد.

در هنگام استفاده از این دستور زمانیکه محتوای Word تعیین شده مساوی 0000 شود فلگ مساوی و زمانیکه یکی از رقمهای BCD از 9 به 0 تبدیل شود فلگ Carry روشن خواهد شد. و هر دو این فلگها زمانیکه محتوای Word از 9999 به 0000 تبدیل شود، روشن خواهند شد.

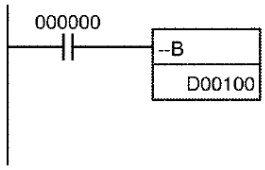
با تغییر دستور B(596)-- به دستور @B(596)-- (با اضافه کردن @ به ابتدای دستور)، دستور به ازاء هر بار تغییر حالت شرط عملکرد از خاموش به روشن (لبهٔ بالارونده) یک واحد BCD محتوای Word تعیین شده را کاهش می دهد.

9-12-4- نکات قابل توجه در مورد دستورالعمل B(596)--

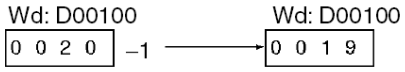
محتوای Wd باید به صورت BCD باشد در غیر اینصورت، یعنی اگر محتوای Wd به صورت BCD نباشد، خطا ایجاد شده و فلگ خطا برای نشان دادن این خطای رخ داده شده روشن خواهد شد.

9-12-5- مثال از عملکرد دستور --B(596)

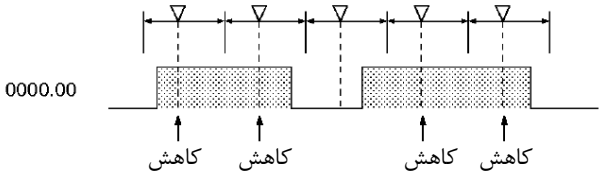
در مثال زیر محتوای D100، تا زمانیکه CIO 0.00 روشن باشد به ازاء هر سیکل یک واحد BCD کاهش می یابد.



تا زمانیکه CIO 0.00 روشن باشد به
ازاء هر سیکل کاهش می دهد.

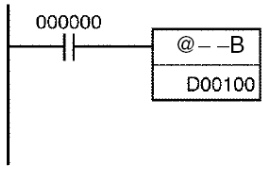


اجرای --B(596)

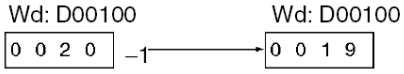


9-12-6- مثال از عملکرد دستور @--B(596)

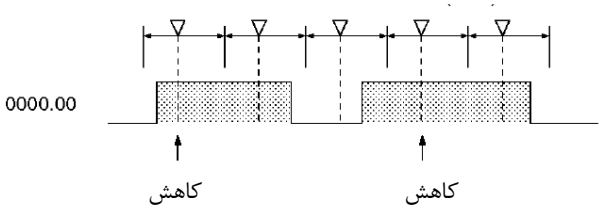
در این مثال که از حالت لبهٔ بالارونده دستور افزایش دهنده استفاده شده است محتوای D100 فقط زمانی که CIO 0.00 از حالت خاموش به روشن تغییر وضعیت می دهد، یعنی یک لبهٔ بالارونده ایجاد می کند یک واحد BCD کاهش پیدا خواهد کرد.



فقط در لبهٔ بالا روندهٔ
ورودی کاهش می دهد.



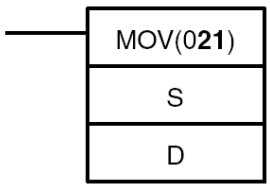
اجرای @--B(596)



9-13- دستور MOVE: MOV(021)

9-13-1- نماد دستور MOV(021) در برنامه نویسی نردبانی

دستور MOV(021) یک داده را به صورت Word به Word معینی از حافظه منتقل می کند.



S: مبدأ

D: مقصد

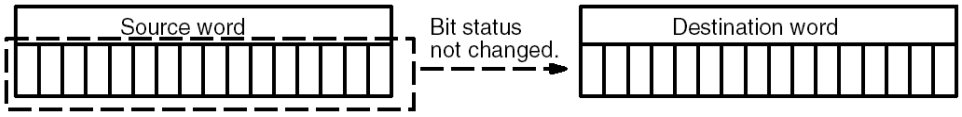
9-13-2- مشخصات عملگرهای دستور MOV(021)

فضای حافظه	S	D
CIO Area	CIO 0000 to CIO 6143	
Work Area	W000 to W511	
Holding Bit Area	H000 to H511	
Auxiliary Bit Area	A000 to A959	A484 to A959
Timer Area	T0000 to T4095	
Counter Area	C0000 to C4095	
DM Area	D00000 to D32767	
EM Area without bank	E00000 to E32767	
EM Area with bank	En_00000 to En_32767 (n = 0 to C)	
Indirect DM/EM addresses in binary	@ D00000 to @ D32767 @ E00000 to @ E32767 @ En_00000 to@ En_32767 (n = 0 to C)	
Indirect DM/EM addresses in BCD	*D00000 to *D32767 *E00000 to *E32767 *En_00000 to *En_032767 (n = 0 to C)	
Constants	#0000 to #FFFF (binary)	---
Data Registers	DR0 to DR15	
Index Registers	---	
Indirect addressing using Index Registers	,IR0 to ,IR15 –2048 to +2047 ,IR0 to –2048 to +2047 ,IR15 DR0 to DR15, IR0 to IR15 ,IR0+(++) to ,IR15+(++) ,–(– –) IR0 to, –(– –) IR15	

9-13-3- شرح عملکرد دستور MOV(021)

دستور MOV(021)، محتوای S (مبدأ) را به D (مقصد) منتقل می کند. در صورتی که S یک عدد ثابت باشد این دستور برای تنظیمات فضای حافظه استفاده می شود.

با تغییر دستور MOV(021) به دستور @MOV(021) (اضافه کردن @ به ابتدای دستور) این دستور زمانی که شرط عملکردش از حالت خاموش به روشن تغییر وضعیت دهد (لبهٔ بالارونده)، محتوای S را به D منتقل می کند.

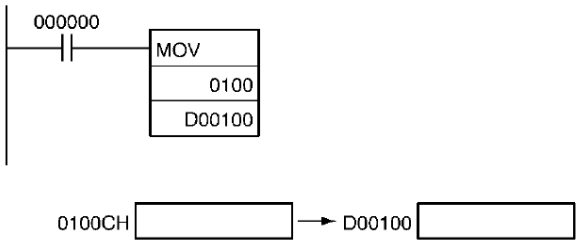


9-13-4- فلگهای دستور MOV(021)

نام	نشانه	عملکرد
Error Flag	ER	خاموش
Equals Flag	=	اگر 0000 به Word مورد نظر منتقل شود روشن می شود. در بقیهٔ موارد خاموش است.
Negative Flag	N	در صورتی که آخرین بیت سمت چپ دادهٔ انتقالی یک باشد روشن می شود. در بقیهٔ موارد خاموش است.

9-13-5- مثال از عملکرد دستور MOV(021)

در مثال زیر زمانی که CIO 0.00 روشن می شود، محتوای آدرس CIO 100 در آدرس D100 کپی می گردد.

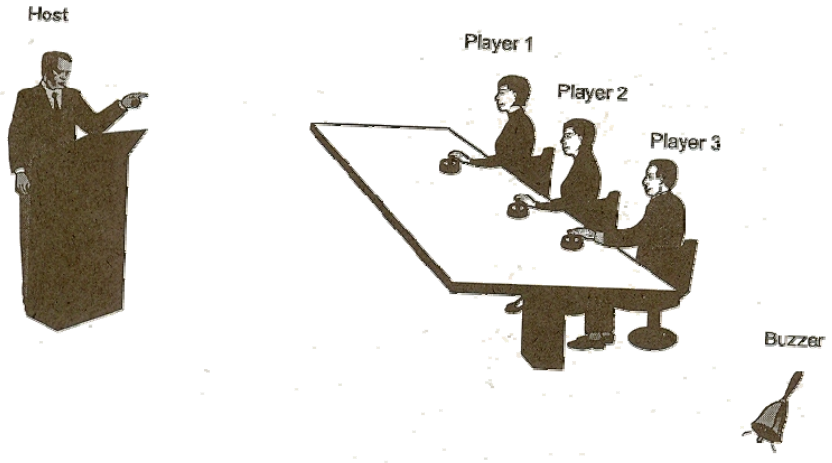


10- مثالهای کاربردی

10-1- طراحی تعیین الویت

شرایط لازم در کنترل زنگ مسابقه:

- زمانی که مجری مسابقه سؤال را به پایان رساند.
- برای مشخص شدن فردی که جواب سؤال را می دهد، سه شرکت کننده شستی های جلوی خود را خواهند فشرد.
- هر کدام از شرکت کنندگان که موفق به فشردن شستی شوند زنگ مسابقه به مدت 2 ثانیه به صدا در خواهد آمد.
- همچنین لامپ جلوی هر شرکت کننده ای که زودتر شستی خود را فشرده باشد روشن خواهد شد، این لامپ توسط مجری و با فشردن شستی Reset خاموش می گردد.



ورودی و خروجی ها:

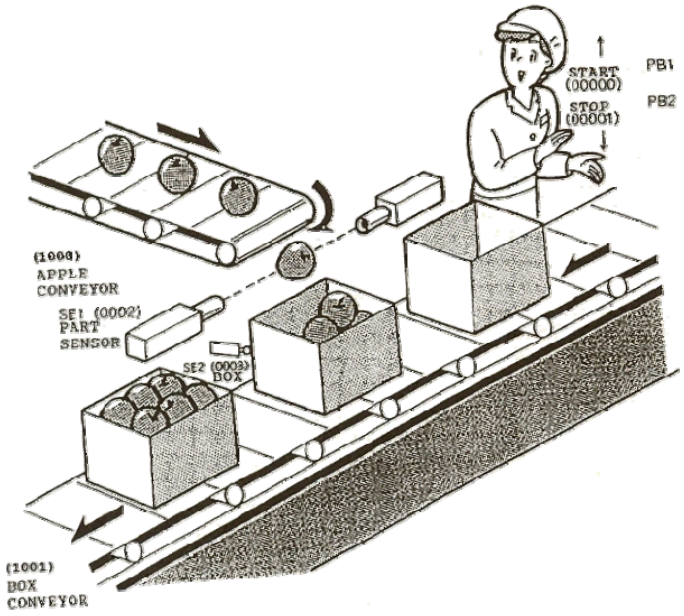
عنصر	خروجی
Buzzer	1.00
Light1	1.01
Light2	1.02
Light3	1.03

عنصر	ورودی
PB1	0.00
PB2	0.01
PB3	0.02
RST(Reset)	0.03

10-2- کنترل خط بسته بندی

عملکرد

وقتی PB1 (شستی Start) فشرده شود، کانوایر مربوط به جعبه حرکت می کند. تا زمانی که وجود جعبه توسط سنسور SE2 آشکار گردد، در این زمان کانوایر جعبه ایستاده و کانوایر سیب شروع به کار می کند. بعد از شمارش 10 سیب توسط سنسور تعبیه شده برای شمارش سیب ها (SE1)، کانوایر سیب ایستاده و کانوایر جعبه دوباره شروع به کار خواهد کرد. کانتر Reset شده و این عملیات تا زمان فشردن PB2 (شستی Stop) ادامه می یابد.



ورودی و خروجی ها:

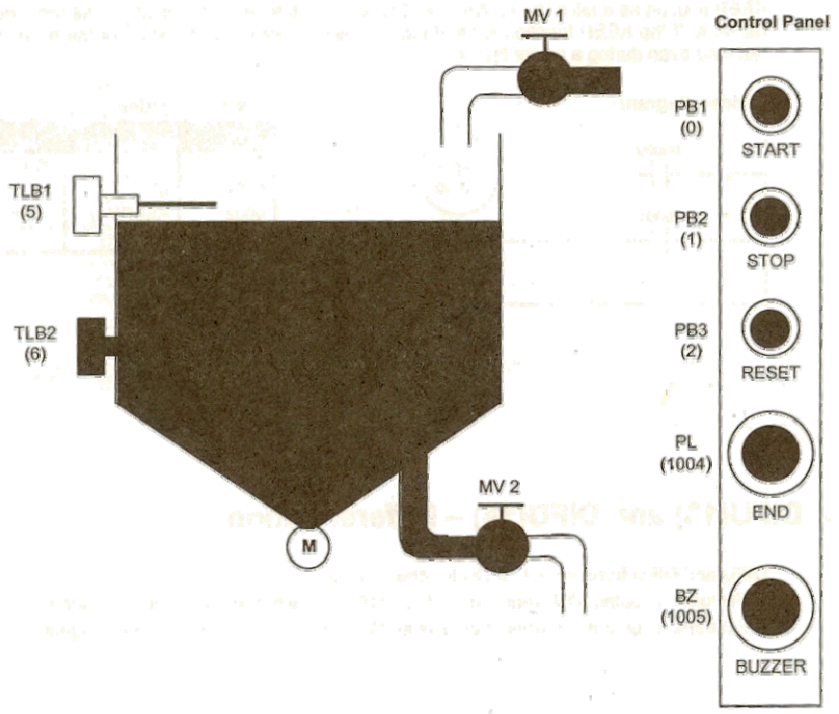
عنصر	خروجی
کانوایر سیب	1.00
کانوایر جعبه	1.01

عنصر	ورودی
START (PB1)	0.00
STOP (PB2)	0.01
SE1	0.02
SE2	0.03

10-3- کنترل پر و خالی شدن مخزن

عملکرد

- با فشردن PB1 شیر MV1 باز شده و مخزن توسط آب پر می شود. بعد از رسیدن سطح آب به سنسور TLB2 موتور همزن M نیز شروع به کار می کند.
- زمانی که سطح آب از TLB2 گذشته و به TLB1 رسید، شیر MV1 بسته شده و 10 ثانیه بعد از بسته شدن شیر MV1، موتور همزن نیز از کار می ایستد.
- بعد از آن شیر تخلیهٔ MV2 باز شده و شروع به تخلیهٔ مخزن می کند. زمانی که سطح آب از سنسور TLB2 پایین تر برود، شیر MV2 بسته شده و دوباره شیر MV1 برای پر کردن مخزن باز خواهد شد.
- زمانی که این سیکل برای چهار بار تکرار شد، نمایشگر END موجود روی تابلوی کنترل روشن شده و این عملیات حتی با فشردن START نیز ادامه نخواهد یافت، مگر اینکه ابتدا Reset و بعد Start فشرده شود.



ورودی و خروجی ها:

عنصر	خروجی
شیر پرکننده MV1	1.00
شیر تخلیه MV2	1.01

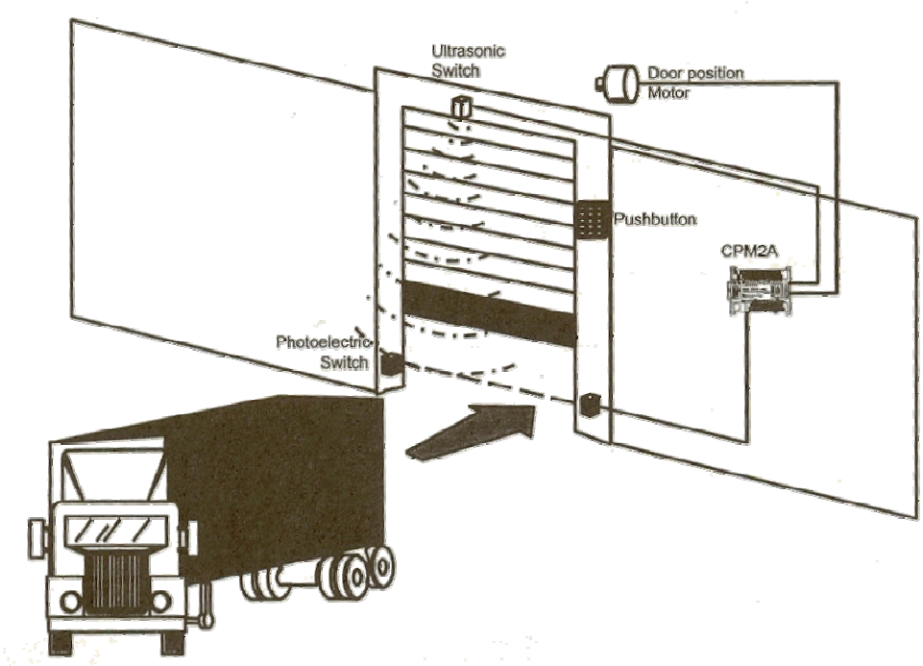
عنصر	ورودی
START (PB1)	0.00
STOP (PB2)	0.01

0.02	RESET (PB3)
0.03	سنسور سطح بالا (TLB1)
0.04	سنسور سطح پایین (TLB2)

1.02	موتور همزن (M)
1.03	نمایشگر END
1.04	Buzzer

10-4- کنترل خودکار در انبار

در این مثال سنسور فراصوتی (Ultrasonic) برای آشکارکردن حضور وسیلهٔ نقلیه و نزدیک شدن آن به کار گرفته شده است. از یک سنسور نوری جداگانه نیز برای نمایان ساختن عبور وسیلهٔ نقلیه و داخل شدن آن استفاده شده است. با توجه به این سیگنالها، هدف کنترلی باز و بسته کردن در انبار و فرمان دادن به موتور متصل به در می باشد.



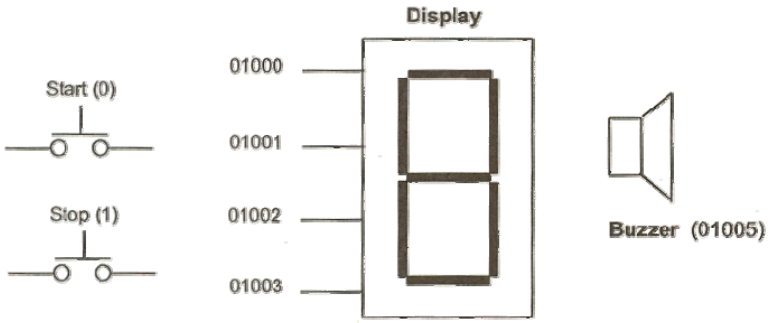
ورودی و خروجی ها:

عنصر	ورودی
سنسور فراصوتی	0.00
سنسور نوری	0.01
لیمیت سوییچ بالای در	0.02
لیمیت سوییچ پایین در	0.03

عنصر	خروجی
موتور (بالا بردن در)	1.00
موتور (پایین بردن در)	1.01

10-5- اخطار زمانی

در این مثال، یک 7-Segment به عنوان نمایشگر و یک زنگ اخطار به PLC متصل شده است. زمانی که دکمهٔ Start فشرده شود، نمایشگر عدد 9 را نشان می دهد و هر ثانیه یک واحد کاهش داده تا به عدد 0 می رسد در این زمان زنگ به صدا درخواهد آمد. این مدار به عنوان مدار اخطار می تواند به کار برده شود.



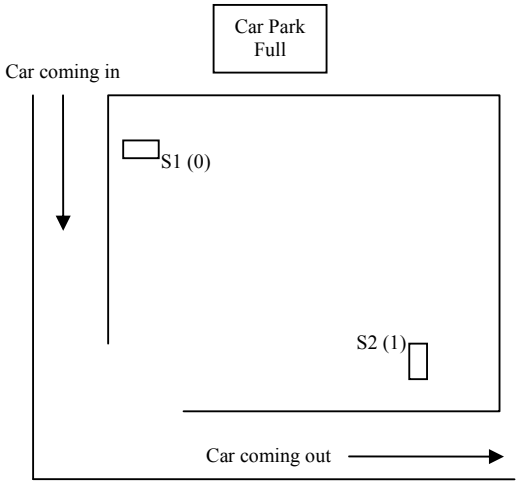
ورودی و خروجی ها:

عنصر	خروجی
نمایشگر 7-Seg	1.00 تا 1.03
زنگ	1.05

عنصر	ورودی
START	0.00
STOP	0.01

10-6- کنترل پارکینگ

این مثال نمونهٔ ساده ای از کنترل یک پارکینگ با ظرفیت 10 خودرو را نشان می دهد. با وارد شدن هر خودرو با توجه به سنسور S1، PLC یکی به محتوای پارکینگ اضافه می کند. به همین ترتیب با خارج شدن هر خودرو یکی از محتوای پارکینگ کم خواهد شد. زمانی که در پارکینگ 10 خودرو موجود باشد باید نمایشگر نشان دهندهٔ تکمیل بودن ظرفیت که جلوی در پارکینگ نصب شده است روشن گردد.



ورودی و خروجی ها:

عنصر	خروجی
تابلو "ظرفیت تکمیل"	1.00

عنصر	ورودی
S1	0.00
S2	0.01

11- طبقه بندی دستورالعملها با توجه به عملکرد

جدولهای زیر فهرستی از دستورالعملهای PLC های سری CS/CJ را که برحسب عملکـرد دسـتورالعمل طبقـه بندی شده اند را در اختیار برنامه نویس قرار می دهد.

Classifica-tion	Sub-class	Mnemonic	Instruction	Mnemonic	Instruction	Mnemonic	Instruction
Basic instructions	Input	LD	LOAD	LD NOT	LOAD NOT	AND	AND
		AND NOT	AND NOT	OR	OR	OR NOT	OR NOT
		AND LD	AND LOAD	OR LD	OR LOAD	---	---
	Output	OUT	OUTPUT	OUT NOT	OUTPUT NOT	---	---
Sequence input instructions	---	NOT	NOT	UP	CONDITION ON	DOWN	CONDITION OFF
	Bit test	LD TST	LD BIT TEST	LD TSTN	LD BIT TEST NOT	AND TST	AND BIT TEST NOT
		AND TSTN	AND BIT TEST NOT	OR TST	OR BIT TEST	OR TSTN	OR BIT TEST NOT
Sequence output instructions	---	KEEP	KEEP	DIFU	DIFFERENTI-ATE UP	DIFD	DIFFERENTI-ATE DOWN
		OUTB*	SINGLE BIT OUTPUT	---	---	---	---
	Set/Reset	SET	SET	RSET	RESET	SETA	MULTIPLE BIT SET
		RSTA	MULTIPLE BIT RESET	SETB*	SINGLE BIT SET	RSTB*	SINGLE BIT RESET
Sequence control instructions	---	END	END	NOP	NO OPERA-TION	---	---
	Interlock	IL	INTERLOCK	ILC	INTERLOCK CLEAR	MILH	MULTI-INTERLOCK DIFFERENTI-ATION HOLD
		MILR (See note 1.)	MULTI-INTERLOCK DIFFERENTI-ATION RELEASE	MILC (See note 1.)	MULTI-INTERLOCK CLEAR	---	---
	Jump	JMP	JUMP	JME	JUMP END	CJP	CONDI-TIONAL JUMP
		CJPN	CONDI-TIONAL JUMP	JMP0	MULTIPLE JUMP	JME0	MULTIPLE JUMP END
	Repeat	FOR	FOR-NEXT LOOPS	BREAK	BREAK LOOP	NEXT	FOR-NEXT LOOPS

Classification	Sub-class		Mnemonic	Instruction	Mnemonic	Instruction	Mnemonic	Instruction
Timer and counter instructions	BCD	Timer (with timer numbers)	TIM	TIMER	TIMH	HIGH-SPEED TIMER	TMHH	ONE-MS TIMER
			TTIM	ACCUMULATIVE TIMER	---	---	---	---
		Timer (without timer numbers)	TIML	LONG TIMER	MTIM	MULTI-OUTPUT TIMER	---	---
		Counter (with counter numbers)	CNT	COUNTER	CNTR	REVERSIBLE TIMER	CNR	RESET TIMER/ COUNTER
	Binary*	Timer (with timer numbers)	TIMX	TIMER	TIMHX	HIGH-SPEED TIMER	TMHHX	ONE-MS TIMER
			TTIMX	ACCUMULATIVE TIMER	---	---	---	---
		Timer (without timer numbers)	TIMLX	LONG TIMER	MTIMX	MULTI-OUTPUT TIMER	---	---
		Counter (with counter numbers)	CNTX	COUNTER	CNTRX	REVERSIBLE TIMER	CNRX	RESET TIMER/ COUNTER
Comparison instructions	Symbol comparison		LD, AND, OR + =, <>, <, <=, >, >=	Symbol comparison (unsigned)	LD, AND, OR + =, <>, <, <=, >, >= + L	Symbol comparison (double-word, unsigned)	LD, AND, OR + =, <>, <, <=, >, >= +S	Symbol comparison (signed)
			LD, AND, OR + =, <>, <, <=, >, >= + SL	Symbol comparison (double-word, signed)	LD, AND, OR + = DT, <> DT, < DT, <= DT, > DT, >= DT (See note 1.)	Time comparison	---	---
	Data comparison (Condition Flags)		CMP	UNSIGNED COMPARE	CMPL	DOUBLE UNSIGNED COMPARE	CPS	SIGNED BINARY COMPARE
			CPSL	DOUBLE SIGNED BINARY COMPARE	ZCP*	AREA RANGE COMPARE	ZCPL*	DOUBLE AREA RANGE COMPARE
	Table compare		MCMP	MULTIPLE COMPARE	TCMP	TABLE COMPARE	BCMP	UNSIGNED BLOCK COMPARE
			BCMP2 (See note 3.)	EXPANDED BLOCK COMPARE	---	---	---	---
Data movement instructions	Single/ double-word		MOV	MOVE	MOVL	DOUBLE MOVE	MVN	MOVE NOT
			MVNL	DOUBLE MOVE NOT	---	---	---	---
	Bit/digit		MOVB	MOVE BIT	MOVD	MOVE DIGIT	---	---
	Exchange		XCHG	DATA EXCHANGE	XCGL	DOUBLE DATA EXCHANGE	---	---
	Block/bit transfer		XFRB	MULTIPLE BIT TRANSFER	XFER	BLOCK TRANSFER	BSET	BLOCK SET
	Distribute/ collect		DIST	SINGLE WORD DISTRIBUTE	COLL	DATA COLLECT	---	---
	Index register		MOVR	MOVE TO REGISTER	MOVRW	MOVE TIMER/ COUNTER PV TO REGISTER	---	---

Classification	Sub-class	Mnemonic	Instruction	Mnemonic	Instruction	Mnemonic	Instruction
Data shift instructions	**1-bit shift**	SFT	SHIFT REGISTER	SFTR	REVERSIBLE SHIFT REGISTER	ASLL	DOUBLE SHIFT LEFT
		ASL	ARITHMETIC SHIFT LEFT	ASR	ARITHMETIC SHIFT RIGHT	ASRL	DOUBLE SHIFT RIGHT
	0000 hex asynchronous	ASFT	ASYNCHRONOUS SHIFT REGISTER	---	---	---	---
	Word shift	WSFT	WORD SHIFT	---	---	---	---
	1-bit rotate	ROL	ROTATE LEFT	ROLL	DOUBLE ROTATE LEFT	RLNC	ROTATE LEFT WITHOUT CARRY
		RLNL	DOUBLE ROTATE LEFT WITHOUT CARRY	ROR	ROTATE RIGHT	RORL	DOUBLE ROTATE RIGHT
		RRNC	ROTATE RIGHT WITHOUT CARRY	RRNL	DOUBLE ROTATE RIGHT WITHOUT CARRY	---	---
	1 digit shift	SLD	ONE DIGIT SHIFT LEFT	SRD	ONE DIGIT SHIFT RIGHT	---	---
	Shift n-bit data	NSFL	SHIFT N-BIT DATA LEFT	NSFR	SHIFT N-BIT DATA RIGHT	---	---
	Shift n-bit	NASL	SHIFT N-BITS LEFT	NSLL	DOUBLE SHIFT N-BITS LEFT	NASR	SHIFT N-BITS RIGHT
		NSRL	DOUBLE SHIFT N-BITS RIGHT	---	---	---	---
Increment/ decrement instructions	**BCD**	++B	INCREMENT BCD	++BL	DOUBLE INCREMENT BCD	– –B	DECREMENT BCD
		– –BL	DOUBLE DECREMENT BCD	---	---	---	---
	Binary	++	INCREMENT BINARY	++L	DOUBLE INCREMENT BINARY	– –	DECREMENT BINARY
		– –L	DOUBLE DECREMENT BINARY	---	---	---	---

Classification	Sub-class	Mnemonic	Instruction	Mnemonic	Instruction	Mnemonic	Instruction
Symbol math instructions	Binary add	+	SIGNED BINARY ADD WITHOUT CARRY	+L	DOUBLE SIGNED BINARY ADD WITHOUT CARRY	+C	SIGNED BINARY ADD WITH CARRY
		+CL	DOUBLE SIGNED BINARY ADD WITH CARRY	---	---	---	---
	BCD add	+B	BCD ADD WITHOUT CARRY	+BL	DOUBLE BCD ADD WITHOUT CARRY	+BC	BCD ADD WITH CARRY
		+BCL	DOUBLE BCD ADD WITH CARRY	---	---	---	---
	Binary subtract	–	SIGNED BINARY SUBTRACT WITHOUT CARRY	–L	DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY	–C	SIGNED BINARY SUBTRACT WITH CARRY
		–CL	DOUBLE SIGNED BINARY WITH CARRY	---	---	---	---
	BCD subtract	–B	BCD SUBTRACT WITHOUT CARRY	–BL	DOUBLE BCD SUBTRACT WITHOUT CARRY	–BC	BCD SUBTRACT WITH CARRY
		–BCL	DOUBLE BCD SUBTRACT WITH CARRY	---	---	---	---
	Binary multiply	*	SIGNED BINARY MULTIPLY	*L	DOUBLE SIGNED BINARY MULTIPLY	*U	UNSIGNED BINARY MULTIPLY
		*UL	DOUBLE UNSIGNED BINARY MULTIPLY	---	---	---	---
	BCD multiply	*B	BCD MULTIPLY	*BL	DOUBLE BCD MULTIPLY	---	---
	Binary divide	/	SIGNED BINARY DIVIDE	/L	DOUBLE SIGNED BINARY DIVIDE	/U	UNSIGNED BINARY DIVIDE
		/UL	DOUBLE UNSIGNED BINARY DIVIDE	---	---	---	---
	BCD divide	/B	BCD DIVIDE	/BL	DOUBLE BCD DIVIDE	---	---

Classification	Sub-class	Mnemonic	Instruction	Mnemonic	Instruction	Mnemonic	Instruction
Conversion instructions	**BCD/Binary convert**	BIN	BCD-TO-BINARY	BINL	DOUBLE BCD-TO-DOUBLE BINARY	BCD	BINARY-TO-BCD
		BCDL	DOUBLE BINARY-TO-DOUBLE BCD	NEG	2'S COMPLEMENT	NEGL	DOUBLE 2'S COMPLEMENT
		SIGN	16-BIT TO 32-BIT SIGNED BINARY	---	---	---	---
	Decoder/ encoder	MLPX	DATA DECODER	DMPX	DATA ENCODER	---	---
	ASCII/HEX convert	ASC	ASCII CONVERT	HEX	ASCII TO HEX	---	---
	Line/column convert	LINE	COLUMN TO LINE	COLM	LINE TO COLUMN	---	---
	Signed binary/BCD convert	BINS	SIGNED BCD-TO- BINARY	BISL	DOUBLE SIGNED BCD-TO- BINARY	BCDS	SIGNED BINARY-TO-BCD
		BDSL	DOUBLE SIGNED BINARY-TO-BCD	GRY (See note 1.)	GRAY CODE CONVERSION	---	---
Logic instructions	**Logical AND/OR**	ANDW	LOGICAL AND	ANDL	DOUBLE LOGICAL AND	ORW	LOGICAL OR
		ORWL	DOUBLE LOGICAL OR	XORW	EXCLUSIVE OR	XORL	DOUBLE EXCLUSIVE OR
		XNRW	EXCLUSIVE NOR	XNRL	DOUBLE EXCLUSIVE NOR	---	---
	Complement	COM	COMPLEMENT	COML	DOUBLE COMPLEMENT	---	---
Special math instructions	**---**	ROTB	BINARY ROOT	ROOT	BCD SQUARE ROOT	APR	ARITHMETIC PROCESS
		FDIV	FLOATING POINT DIVIDE	BCNT	BIT COUNTER	---	---
Floating-point math instructions	**Floating point/ binary convert**	FIX	FLOATING TO 16-BIT	FIXL	FLOATING TO 32-BIT	FLT	16-BIT TO FLOATING
		FLTL	32-BIT TO FLOATING	---	---	---	---
	Floating- point basic math	+F	FLOATING-POINT ADD	–F	FLOATING-POINT SUBTRACT	/F	FLOATING-POINT DIVIDE
		*F	FLOATING-POINT MULTIPLY	---	---	---	---
	Floating- point trigonometric	RAD	DEGREES TO RADIANS	DEG	RADIANS TO DEGREES	SIN	SINE
		COS	COSINE	TAN	TANGENT	ASIN	ARC SINE
		ACOS	ARC COSINE	ATAN	ARC TANGENT	---	---
	Floating- point math	SQRT	SQUARE ROOT	EXP	EXPONENT	LOG	LOGARITHM
		PWR	EXPONENTIAL POWER	---	---	---	---
	Symbol comparison and conversion*	LD, AND, OR + =, <>, <, <=, >, >= + F	Symbol comparison (single-precision floating point)	FSTR*	FLOATING-POINT TO ASCII	FVAL*	ASCII TO FLOATING-POINT

Classification	Sub-class	Mnemonic	Instruction	Mnemonic	Instruction	Mnemonic	Instruction
Double-precision floating- point instructions*	**Floating point/ binary convert**	FIXD	DOUBLE FLOATING TO 16-BIT	FIXLD	DOUBLE FLOATING TO 32-BIT	DBL	16-BIT TO DOUBLE FLOATING
		DBLL	32-BIT TO DOUBLE FLOATING	---	---	---	---
	Floating- point basic math	+D	DOUBLE FLOATING-POINT ADD	–D	DOUBLE FLOATING-POINT SUBTRACT	/D	DOUBLE FLOATING-POINT DIVIDE
		*D	DOUBLE FLOATING-POINT MULTIPLY	---	---	---	---
	Floating- point trigonometric	RADD	DOUBLE DEGREES TO RADIANS	DEGD	DOUBLE RADIANS TO DEGREES	SIND	DOUBLE SINE
		COSD	DOUBLE COSINE	TAND	DOUBLE TANGENT	ASIND	DOUBLE ARC SINE
		ACOSD	DOUBLE ARC COSINE	ATAND	DOUBLE ARC TANGENT	---	---
	Floating- point math	SQRTD	DOUBLE SQUARE ROOT	EXPD	DOUBLE EXPONENT	LOGD	DOUBLE LOGARITHM
		PWRD	DOUBLE EXPONENTIAL POWER	---	---	---	---
	Symbol comparison	LD, AND, OR + =, <>, <, <=, >, >= + D	Symbol comparison (double-precision floating point)	---	---	---	---
Table data processing instructions	**Stack processing**	SSET	SET STACK	PUSH	PUSH ONTO STACK	LIFO	LAST IN FIRST OUT
		FIFO	FIRST IN FIRST OUT	SNUM*	STACK SIZE READ	SREAD*	STACK DATA READ
		SWRIT*	STACK DATA OVERWRITE	SINS*	STACK DATA INSERT	SDEL*	STACK DATA DELETE
	1-record/ multiple-word processing	DIM	DIMENSION RECORD TABLE	SETR	SET RECORD LOCATION	GETR	GET RECORD NUMBER
	Record-to- word processing	SRCH	DATA SEARCH	MAX	FIND MAXIMUM	MIN	FIND MINIMUM
		SUM	SUM	FCS	FRAME CHECKSUM	---	---
	Byte processing	SWAP	SWAP BYTES	---	---	---	---
Data control instructions	---	PID	PID CONTROL	PIDAT*	PID CONTROL WITH AUTOTUNING	LMT	LIMIT CONTROL
		BAND	DEAD BAND CONTROL	ZONE	DEAD ZONE CONTROL	TPO (See note 1.)	TIME-PROPORTIONAL OUTPUT
		SCL	SCALING	SCL2	SCALING 2	SCL3	SCALING 3
		AVG	AVERAGE	---	---	---	---
Subroutines instructions	---	SBS	SUBROUTINE CALL	MCRO	MACRO	SBN	SUBROUTINE ENTRY
		RET	SUBROUTINE RETURN	GSBS*	GLOBAL SUBROUTINE CALL	GSBN*	GLOBAL SUBROUTINE ENTRY
		GRET*	GLOBAL SUBROUTINE RETURN	---	---	---	---

Classification	Sub-class	Mnemonic	Instruction	Mnemonic	Instruction	Mnemonic	Instruction
Interrupt control instructions	---	MSKS***	SET INTERRUPT MASK	MSKR***	READ INTERRUPT MASK	CLI***	CLEAR INTERRUPT
		DI	DISABLE INTERRUPTS	EI	ENABLE INTERRUPTS	---	---
High-speed counter/ pulse output instructions**	---	INI	MODE CONTROL	PRV	HIGH-SPEED COUNTER PV READ	PRV2 (See note 2.)	COUNTER FREQUENCY CONVERT
		CTBL	COMPARISON TABLE LOAD	SPED	SPEED OUTPUT	PULS	SET PULSES
		PLS2	PULSE OUTPUT	ACC	ACCELERATION Control	ORG	ORIGIN SEARCH
Step instructions	---	PWM	PULSE WITH VARIABLE DUTY FACTOR	STEP	STEP DEFINE	SNXT	STEP START
Basic I/O Unit instructions	---	IORF	I/O REFRESH	SDEC	7-SEGMENT DECODER	DSW (See note 1.)	DIGITAL SWITCH INPUT
		TKY (See note 1.)	TEN KEY INPUT	HKY (See note 1.)	HEXADECIMAL KEY INPUT	MTR (See note 1.)	MATRIX INPUT
		7SEG (See note 1.)	7-SEGMENT DISPLAY OUTPUT	IORD	INTELLIGENT I/O READ	IOWR	INTELLIGENT I/O WRITE
		DLNK*	CPU BUS UNIT I/O REFRESH	---	---	---	---
Serial communications instructions	---	PMCR	PROTOCOL MACRO	TXD	TRANSMIT	RXD	RECEIVE
		STUP	CHANGE SERIAL PORT SETUP	---	---	---	---
Network instructions	---	SEND	NETWORK SEND	RECV	NETWORK RECEIVE	CMND	DELIVER COMMAND
		EXPLT (See note 1.)	SEND GENERAL EXPICIT	EGATR (See note 1.)	EXPLICIT GET ATTRIBUTE	ESATR (See note 1.)	EXPLICIT SET ATTRIBUTE
		ECHRD (See note 1.)	EXPLICIT WORD READ	ECHWR (See note 1.)	EXPLICIT WORD WRITE	---	---
Display instructions	---	MSG	DISPLAY MESSAGE	---	---	---	---
File memory instructions	---	FREAD	READ DATA FILE	FWRIT	WRITE DATA FILE	---	---
Clock instructions	---	CADD	CALENDAR ADD	CSUB	CALENDAR SUBTRACT	SEC	HOURS TO SECONDS
		HMS	SECONDS TO HOURS	DATE	CLOCK ADJUSTMENT	---	---
Debugging instructions	---	TRSM	TRACE MEMORY SAMPLING	---	---	---	---
Failure diagnosis instructions	---	FAL	FAILURE ALARM	FALS	SEVERE FAILURE ALARM	FPD	FAILURE POINT DETECTION

Classification	Sub-class		Mnemonic	Instruction	Mnemonic	Instruction	Mnemonic	Instruction
Other instructions	---		STC	SET CARRY	CLC	CLEAR CARRY	EMBC	SELECT EM BANK
			WDT	EXTEND MAXIMUM CYCLE TIME	CCS*	SAVE CONDITION FLAGS	CCL*	LOAD CONDITION FLAGS
			FRMCV*	CONVERT ADDRESS FROM CV	TOCV*	CONVERT ADDRESS TO CV	IOSP***	DISABLE PERIPHERAL SERVICING
			IORS***	ENABLE PERIPHERAL SERVICING	---	---	---	---
Block programming instructions	**Define block program area**		BPRG	BLOCK PROGRAM BEGIN	BEND	BLOCK PROGRAM END	---	---
	Block program start/stop		BPPS	BLOCK PROGRAM PAUSE	BPRS	BLOCK PROGRAM RESTART	---	---
	EXIT		EXIT *bit_address*	Conditional END	EXIT NOT *bit_address*	Conditional END NOT	*input_condition* EXIT	Conditional END
	IF branch processing		IF *bit_address*	CONDITIONAL BLOCK BRANCHING	IF NOT *bit_address*	CONDITIONAL BLOCK BRANCHING (NOT)	ELSE	CONDITIONAL BLOCK BRANCHING (ELSE)
			IEND	CONDITIONAL BLOCK BRANCHING END	---	---	---	---
	WAIT		WAIT *bit_address*	ONE CYCLE AND WAIT	WAIT NOT *bit_address*	ONE CYCLE AND WAIT NOT	*input_condition* WAIT	ONE CYCLE AND WAIT
	Timer/ counter	**BCD**	TIMW	TIMER WAIT	CNTW	COUNTER WAIT	TMHW	HIGH-SPEED TIMER WAIT
		Binary*	TIMWX	TIMER WAIT	CNTWX	COUNTER WAIT	TMHWX	HIGH-SPEED TIMER WAIT
	Repeat		LOOP	LOOP BLOCK	LEND *bit_address*	LOOP BLOCK END	LEND NOT *bit_address*	LOOP BLOCK END NOT
			input_condition LEND	LOOP BLOCK END	---	---	---	---
Text string processing instructions	---		MOV$	MOV STRING	+$	CONCATENATE STRING	LEFT$	GET STRING LEFT
			RIGHT$	GET STRING RIGHT	MID$	GET STRING MIDDLE	FIND$	FIND IN STRING
			LEN$	STRING LENGTH	RPLC$	REPLACE IN STRING	DEL$	DELETE STRING
			XCHG$	EXCHANGE STRING	CLR$	CLEAR STRING	INS$	INSERT INTO STRING
			LD, AND, OR + =$, <>$, <$, <=$, >$, >=$	STRING COMPARISON	---	---	---	---
Task control instructions	---		TKON	TASK ON	TKOF	TASK OFF	---	---

پایکار بنیان صنعت

تهران15167، خیابان وزرا، روبروی خیابان سی و پنجم، شمارهٔ 114
صندوق پستی 15875/4558

114 Vozara Ave., Tehran 15167, Iran
P.O.Box: 15875/4558
Phone: +98(21)8888 22 86
Fax: +98(21)8888 22 63

www.pbs.ir info@pbs.ir