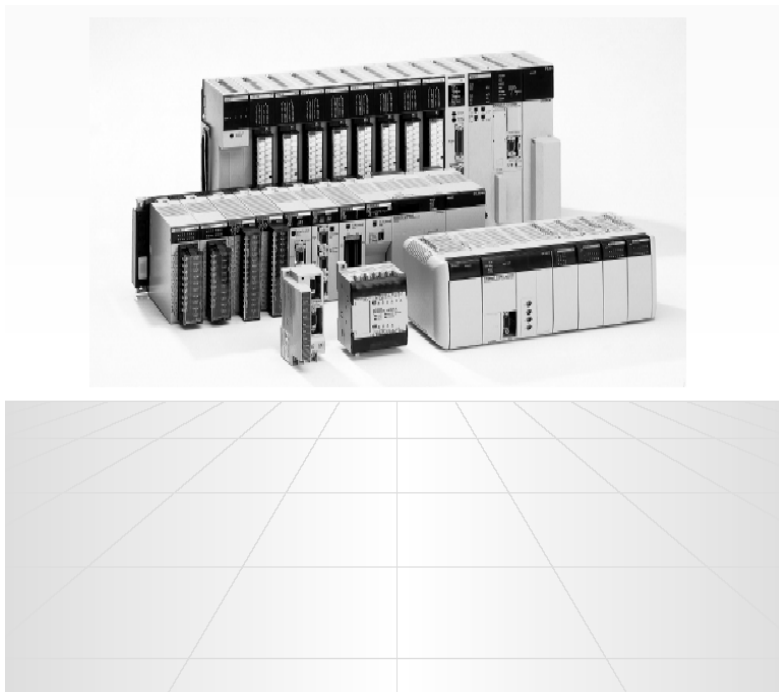


مطالب درسی دوره ی PLC II



مقدمه

ویرایش اول :

سیستم های کنترل مجموعه ای از تجهیزات الکترونیکی هستند که در جهت تضمین پایداری، دقت و صحت یک فرآیند، برای رسیدن به یک هدف کنترلی در کنار هم قرار گرفته اند. این تجهیزات با توجه به نوع کاربرد و نوع کنترل، به کار گرفته می شوند. در نتیجه پیشرفت سریع تکنولوژی، انجام و پیاده سازی کنترل های پیچیده، توسط سیستمهای کنترل خودکار همانند کنترل کننده های منطقی برنامه پذیر (PLC) انجام می گیرد.

PLC ها برای اولین بار در اوایل دهه 1970 در صنعت اتوماسیون مورد استفاده قرار گرفته و جایگزین تابلوهای عظیم رله ای شدند، این سیستم ها با حجم کوچک ولی قابلیت های فراوان به عنوان قلب سیستمهای کنترل به کار گرفته شدند.

در دنیای پررقابت امروز کارآمد بودن، انعطاف پذیری و همچنین مقرون به صرفه بودن از لحاظ اقتصادی از جمله مواردی است که باعث می شود تولید کنندگان بتوانند در عرصه رقابت پا برجا باقی بمانند. شرکت **OMRON** نیز با در نظر گرفتن این فاکتورها سعی بر آن داشته که با ارائه محصولات با عملکرد ساده ولی سریع، مطمئن و انعطاف پذیر یکی از تولید کنندگان معتبر و اصلی در زمینه تولید کنترل کننده های منطقی برنامه پذیر باشد.

در این نوشته سعی بر آن بوده است که کاربردهای دیگر و پیچیده تری از کنترل کننده های منطقی برنامه پذیر (PLC) ساخت شرکت **OMRON** ارائه شود. با این امید که توانسته باشیم گامی هر چند کوچک در جهت آشنایی بیشتر کارشناسان و علاقه مندان به PLC و صنعت اتوماسیون برداریم.

سعیده لاله ریحانی

پاییز 1385

ویرایش دوم :

در چاپ دوم این نوشتار با توجه به بازخورد های دریافتی از خوانندگان برخی از تغییرات به شرح زیر انجام پذیرفت :

- 1- اشتباه های چاپی وجود داشته بر طرف گردید.
- 2- جدول کد ASCII جهت استفاده در بخش ایجاد خطاها به انتهای جزوه اضافه گردید.
- 3- بخش وقفه های PLC مدل CJ1M تکمیل گردید و وقفه های مربوط به PLC مدل CP1H حذف گردید.

مریم یوسفی

فاطمه ژبانی

پاییز 1387

سرفصل مطالب:

- 1- سیستمهای اعداد.....1
- 1-1-1 نگاه کلی به سیستم های اعداد.....1
- 1-1-1-1 دسیمال (Decimal).....1
- 2-1-1-1 باینری (Binary).....2
- 3-1-1-1 باینری کد شده به دسیمال (BCD).....4
- 2- کارت های ورودی و خروجی آنالوگ.....6
- 1-2-1 مشخصات کلی کارت ورودی/خروجی آنالوگ سری CJ1 (CJ1W-MAD42).....6
- 1-1-2-1 مشخصات.....7
- 2-2-1 نمودار عملکرد ورودی/خروجی.....9
- 3-2-1 مشخصات ورودی.....9
- 4-2-1 مشخصات خروجی.....11
- 5-2-1 رویه عملکرد.....12
- 1-5-2-1 مثالهای چگونگی عملکرد.....13
- 6-2-1 جزییات و تنظیمات مربوط به سویچ ها.....18
- 1-6-2-1 نمایشگرها.....19
- 2-6-2-1 سویچ های Unit Number.....19
- 3-6-2-1 سویچ جریان/ولتاژ.....20
- 7-2-1 سیم بندی.....20
- 1-7-2-1 ترتیب ترمینالها.....20
- 2-7-2-1 ترتیب مدارهای داخلی.....21
- 3-7-2-1 مثالی از سیم بندی ورودی/خروجی ها.....22
- 4-7-2-1 ملاحظات در سیم بندی ورودی/خروجی ها.....22
- 8-2-1 تبادل اطلاعات با واحد CPU.....23
- 1-8-2-1 نگاه کلی بر تبادل داده ها.....23
- 2-8-2-1 تنظیمات مربوط به Unit Number.....24
- 3-8-2-1 تنظیمات مربوط به مد عملکرد.....24
- 4-8-2-1 بیت های راه اندازی مجدد کارت آنالوگ.....25
- 5-8-2-1 تخصیص داده های ثابت.....26
- 6-8-2-1 داده های مربوط به تجدید حالت ورودی/خروجی ها.....29
- 9-2-1 توابع ورودی های آنالوگ و شیوه عملکرد آنها.....32
- 1-9-2-1 تنظیمات ورودی و تبدیل مقادیر.....32
- 2-9-2-1 تنظیمات زمان تبدیل و دقت.....34
- 3-9-2-1 پردازش مقدار میانگین.....34
- 4-9-2-1 تابع نگه داشتن مقدار حداکثر.....36
- 5-9-2-1 توابع مقیاس گذاری ورودی.....37
- 6-9-2-1 تابع مشخصه قطعی ورودی.....39
- 10-2-1 توابع خروجی های آنالوگ و شیوه عملکرد آنها.....40

40 1-10-2 تنظیمات خروجی و تبدیل مقادیر
42 2-10-2 تنظیمات زمان تبدیل و دقت
43 3-10-2 تابع تعیین حالت خروجی در صورت توقف تبدیل
44 4-10-2 توابع مقیاس گذاری خروجی
47 5-10-2 خطاهای تنظیمات خروجی
47 11-2 تنظیم آفست و بهره
49 1-11-2 چگونگی عملکرد در Adjustment Mode
50 12-2 بررسی خطاها و هشدارها
50 1-12-2 نمایشگرها و نمودار خطا
51 2-12-2 خطاهای کارت ورودی/خروجی آنالوگ
52 3-12-2 خطاهای CPU
53 4-12-2 راه اندازی مجدد کارتهای ویژه
55 دستورالعملها
55 3- دستورالعملهای شیفت داده ها
55 1-3- دستور ثبات تغییر مکان (Shift Register-SFT)
55 1-1-3 مشخصات عملگرهای دستور SFT(010)
56 2-1-3 شرح عملکرد دستور SFT(010)
56 3-1-3 فلگ های مربوط به دستورالعمل SFT(010)
56 4-1-3 نکات قابل توجه در مورد دستورالعمل SFT(010)
57 5-1-3 مثال از عملکرد دستور SFT(010)
57 2-3 ثبات تغییر مکان برگشت پذیر (Reversible Shift Register-SFTR)
57 1-2-3 مشخصات عملگرهای دستور SFTR(084)
58 2-2-3 شرح عملکرد دستور SFTR(084)
59 3-2-3 فلگ های مربوط به دستورالعمل SFTR(084)
59 4-2-3 نکات قابل توجه در مورد دستورالعمل SFTR(084)
59 5-2-3 مثال از عملکرد دستور SFTR(084)
60 3-3 تغییر مکان Word (Word Shift-WSFT)
61 1-3-3 مشخصات عملگرهای دستور WSFT (016)
61 2-3-3 شرح عملکرد دستور WSFT (016)
61 3-3-3 فلگ های مربوط به دستورالعمل WSFT(016)
62 4-3-3 مثال از عملکرد دستور WSFT (016)
62 4-3 تغییر مکان حسابی به چپ (Arithmetic Shift Left-ASL)
62 1-4-3 مشخصات عملگرهای دستور ASL (025)
63 2-4-3 شرح عملکرد دستور ASL (025)
63 3-4-3 فلگ های مربوط به دستورالعمل ASL (025)
63 4-4-3 مثال از عملکرد دستور ASL (025)
63 5-3 تغییر مکان حسابی به راست (Arithmetic Shift Right-ASR)
64 1-5-3 مشخصات عملگرهای دستور ASR (026)
64 2-5-3 شرح عملکرد دستور ASR(026)
64 3-5-3 فلگ های مربوط به دستورالعمل ASR (026)

65	4-5-3	مثال از عملکرد دستور (026) ASR
65	6-3	تغییر مکان یک دیجیت به چپ (One Digit Shift Left-SLD)
65	1-6-3	مشخصات عملگرهای دستور (074) SLD
66	2-6-3	شرح عملکرد دستور (074) SLD
66	3-6-3	فلگ های مربوط به دستورالعمل (074) SLD
66	4-6-3	مثال از عملکرد دستور (074) SLD
66	7-3	تغییر مکان یک دیجیت به راست (One Digit Shift Right-SRD)
67	1-7-3	مشخصات عملگرهای دستور (075) SRD
67	2-7-3	شرح عملکرد دستور (075) SRD
67	3-7-3	فلگ های مربوط به دستورالعمل (075) SRD
68	4-7-3	مثال از عملکرد دستور (075) SRD
68	4	دستورالعملهای تغییر مکان داده ها
68	1-4	تغییر مکان بیتی (Move Bit-MOVB)
69	1-1-4	مشخصات عملگرهای دستور (082) MOVB
70	2-1-4	شرح عملکرد دستور (082) MOVB
70	3-1-4	فلگ های مربوط به دستورالعمل (082) MOVB
70	4-1-4	مثال از عملکرد دستور (082) MOVB
71	2-4	تغییر مکان دیجیتی (Move Digit-MOVD)
71	1-2-4	مشخصات عملگرهای دستور (083) MOVD
72	2-2-4	شرح عملکرد دستور (083) MOVD
73	3-2-4	فلگ های مربوط به دستورالعمل (083) MOVD
73	4-2-4	مثال از عملکرد دستور (083) MOVD
74	3-4	انتقال یک بلوک از حافظه (Block Transfer-XFER)
74	1-3-4	مشخصات عملگرهای دستور (070) XFER
75	2-3-4	شرح عملکرد دستور (070) XFER
76	3-3-4	نکات قابل توجه در مورد دستورالعمل (070) XFER
76	5-3-4	مثال از عملکرد دستور (070) XFER
76	4-4	بار کردن یک بلوک از حافظه (Block Set-BSET)
76	1-4-4	مشخصات عملگرهای دستور (071) BSET
78	2-4-4	شرح عملکرد دستور (071) BSET
78	3-4-4	فلگ های مربوط به دستورالعمل (071) BSET
78	4-4-4	مثال از عملکرد دستور (071) BSET
79	5	دستورالعملهای محاسباتی
79	1-5	جمع BCD بدون Carry (BCD Add, +B)
79	1-1-5	مشخصات عملگرهای دستور (404) +B
80	2-1-5	شرح عملکرد دستور (404) +B
80	3-1-5	فلگ های مربوط به دستورالعمل (404) +B
80	4-1-5	مثال از عملکرد دستور (404) +B
80	2-5	جمع باینری بدون Carry (Binary Add, +)
81	1-2-5	مشخصات عملگرهای دستور (400) +
81	2-2-5	شرح عملکرد دستور (400) +

81 + (400) 3-2-5- فلگ های مربوط به دستورالعمل
82 + (400) 4-2-5- مثال از عملکرد دستور
82 (BCD Subtract,-B) Carry بدون BCD 3-5- تفریق
82-B(414) 1-3-5- مشخصات عملگرهای دستور
83-B(414) 2-3-5- شرح عملکرد دستور
83-B(414) 3-3-5- فلگ های مربوط به دستورالعمل
83-B(414) 4-3-5- مثال از عملکرد دستور
83 (Binary Subtrct,-) Carry بدون باینری 4-5- تفریق
84- (410) 1-4-5- مشخصات عملگرهای دستور
84- (410) 2-4-5- شرح عملکرد دستور
85- (410) 3-4-5- فلگ های مربوط به دستورالعمل
85- (410) 4-4-5- مثال از عملکرد دستور
85 (BCD Multiply,*B) Carry بدون BCD 5-5- ضرب
86*B(424) 1-5-5- مشخصات عملگرهای دستور
86*B(424) 2-5-5- شرح عملکرد دستور
86*B(424) 3-5-5- فلگ های مربوط به دستورالعمل
87*B(424) 4-5-5- مثال از عملکرد دستور
87 (Binary Multiply,*) Carry بدون ضرب باینری 6-5- ضرب
87*(420) 1-6-5- مشخصات عملگرهای دستور
88*(420) 2-6-5- شرح عملکرد دستور
88*(420) 3-6-5- فلگ های مربوط به دستورالعمل
88*(420) 4-6-5- مثال از عملکرد دستور
88 (BCD Devide,/B) Carry بدون BCD 7-5- تقسیم
89/B(434) 1-7-5- مشخصات عملگرهای دستور
89/B(434) 2-7-5- شرح عملکرد دستور
89/B(434) 3-7-5- فلگ های مربوط به دستورالعمل
90/B(434) 4-7-5- مثال از عملکرد دستور
90 (Binary Devide,/) Carry بدون تقسیم باینری 8-5- تقسیم
90/ (430) 1-8-5- مشخصات عملگرهای دستور
91/ (430) 2-8-5- شرح عملکرد دستور
91/ (430) 3-8-5- فلگ های مربوط به دستورالعمل
91/ (430) 4-8-5- مثال از عملکرد دستور
91 6- دستورالعملهای تشخیص عیب
92 (Failure Alarm-FAL) هشدار دهنده خطا 1-6- هشدار
92FAL(006) 1-1-6- مشخصات عملگرهای دستور
93FAL(006) 2-1-6- شرح عملکرد دستور
97FAL(006) 3-1-6- فلگ های مربوط به دستورالعمل
98 (Severe Failure Alarm-FALS) هشدار دهنده خطاهای شدید 2-6- هشدار
98FALS(007) 1-2-6- مشخصات عملگرهای دستور
99FALS(007) 2-2-6- شرح عملکرد دستور
104FALS(007) 3-2-6- فلگ های مربوط به دستورالعمل

104	7- دستورات کنترل ترتیب
105	1-7-1- دستورالعملهای Interlock Clear و Interlock (IL و ILC)
105	1-1-7- شرح عملکرد دستور IL(002) و ILC(003)
106	2-1-7- نکات قابل توجه در مورد دستورالعمل IL(002) و ILC(003)
107	3-1-7- مثال از عملکرد دستور IL(002) و ILC(003)
108	2-2- دستورالعملهای پرش و انتهای پرش (JMP, JME و Jump End)
108	1-2-7- مشخصات عملگرهای دستور JMP(004) و JME(005)
109	2-2-7- شرح عملکرد دستور JMP(004) و JME(005)
109	3-2-7- فلگ های مربوط به دستورالعمل JMP(004) و JME(005)
110	4-2-7- نکات قابل توجه در مورد دستورالعمل JMP(004) و JME(005)
111	5-2-7- مثال از عملکرد دستورات JMP(004) و JME(005)
111	8- دستورات مقیاس کردن
112	1-8-1- دستورالعمل مقیاس گذاری (Scaling-SCL)
112	1-1-8- مشخصات عملگرهای دستور SCL(194)
113	2-1-8- شرح عملکرد دستور SCL(194)
114	3-1-8- فلگ های مربوط به دستورالعمل SCL(194)
114	4-1-8- مثال از عملکرد دستور SCL(194)
116	2-8-2- دستورالعمل مقیاس گذاری 2 (Scaling 2-SCL2)
117	1-2-8- مشخصات عملگرهای دستور SCL2(486)
117	2-2-8- شرح عملکرد دستور SCL2(486)
119	3-2-8- فلگ های مربوط به دستورالعمل SCL2(486)
119	4-2-8- مثال از عملکرد دستور SCL2(486)
120	3-8-3- دستورالعمل مقیاس گذاری 3 (Scaling 3-SCL3)
121	1-3-8- مشخصات عملگرهای دستور SCL3(487)
122	2-3-8- شرح عملکرد دستور SCL3(487)
123	3-3-8- فلگ های مربوط به دستورالعمل SCL3(487)
123	4-3-8- مثال از عملکرد دستور SCL3(487)
124	9- SUBROUTINE ها
124	1-9-1- فراخوانی Subroutine (SBS)
124	1-1-9- مشخصات عملگرهای دستور SBS(091)
124	2-1-9- شرح عملکرد دستور SBS(091)
126	3-1-9- Subroutine و دستورات لبه
127	4-1-9- فلگ های مربوط به دستورالعمل SBS(091)
127	5-1-9- نکات قابل توجه در مورد دستورالعمل SBS(091)
127	6-1-9- مثال هایی از عملکرد دستور SBS(091)
128	2-9-2- تعریف کردن Subroutine (SBN)
128	1-2-9- مشخصات عملگرهای دستور SBN(092)
129	2-2-9- شرح عملکرد دستور SBN(092)
130	3-9-3- بازگشت Subroutine (RET)
130	1-3-9- شرح عملکرد دستور RET(093)
131	10- آدرس دهی غیر مستقیم

131	1-10-1-10	مقدمه ای بر آدرس دهی غیر مستقیم
132	1-1-10	آدرس دهی غیر مستقیم در ناحیه DM
133	2-1-10	آدرس دهی غیر مستقیم در ناحیه IR
135		11- توابع وقفه
135	1-11	نگاهی کلی به توابع وقفه (CJM)
135	1-1-11	انواع توابع وقفه
136	2-1-11	ایجاد Task وقفه
137	3-1-11	الویت Task های وقفه
137	4-1-11	پردازش دوتایی توسط Task های دوره ای و Task های وقفه
140	2-11	ورودی وقفه (حالت مستقیم)
140	1-2-11	بیت‌های ورودی وقفه و ترمینال‌های مختص آن
140	2-2-11	شیوه عملکرد
141	3-2-11	تنظیمات PLC
141	4-2-11	نوشتن برنامه نردبانی
142	5-2-11	تنظیمات ورودی های وقفه و عملکرد آنها
143	3-11	ورودی وقفه (حالت شمارشی)
143	1-3-11	نگاه کلی
145	2-3-11	تنظیمات PLC
145	3-3-11	نوشتن برنامه نردبانی
146	4-3-11	تنظیمات ورودی های وقفه و عملکرد آنها
147	4-11	وقفه های برنامه ریزی شده
147	1-4-11	عملکرد
147	2-4-11	تنظیمات PLC
149	3-4-11	نوشتن برنامه نردبانی
150	4-4-11	تنظیمات ورودی های وقفه و عملکرد آنها
150	6-11	وقفه های شمارنده های پرسرعت
151	1-6-11	عملکرد
151	7-11	وقفه های خارجی
151	8-11	وقفه قطع برق
152		12- مروری دوباره بر فضای حافظه (یادآوری)
152	1-12	تعریف فضای داده ها
152	1-1-12	ناحیه CIO (Common Input/Output Area)
153	2-1-12	ناحیه W (Work Area)
153	3-1-12	ناحیه H (Holding Area)
154	4-1-12	ناحیه A (Auxiliary Area)
154	5-1-12	ناحیه D (Data Memory Area)
155	6-1-12	ناحیه E (Extended Data Memory Area)
156		مشخص کردن آدرسهای ناحیه E

156.....	مشخص کردن بانک و آدرس مورد نظر در آن بانک.....
156.....	مشخص کردن آدرس در بانک جاری
157.....	7-1-12- ناحیه تایمر (Timer Area).....
158.....	8-1-12- ناحیه کانتر (Counter Area).....
158.....	9-1-12- فلگهای وضعیت
159.....	10-1-12- پالسهای زمانی
159.....	11-1-12- ناحیه TK (Task Flag Area).....
160.....	12-1-12- ناحیه IR (Index Register).....
160.....	13-1-12- ناحیه DR (Data Register).....
160.....	2-12- نقشه حافظه
161.....	13- مثالهای کاربردی
161.....	1-13- مکانیزم دسته بندی توپ ها
162.....	2-13- کنترل جرثقیل سقفی برای عملکرد روغن زدایی
163.....	3-13- محاسبه عمر تیغه برش
164.....	14- طبقه بندی دستورالعملها با توجه به عملکرد.....
	15-جدول کد ASCII.....
172.....

1- سیستمهای اعداد

1-1- نگاهی کلی به سیستم های اعداد

به دلیل اینکه PLC ها از جمله تجهیزات دیجیتال می باشند، فقط به دو حالت و شرایط "روشن" و "خاموش" پاسخ خواهند داد. در این صورت عدد "1" نشان دهنده حالت "روشن" و عدد "0" نشان دهنده حالت خاموش می باشد. با توجه به محدود بودن اطلاعات به صورت "0" و "1" باید روشی وجود داشته باشد که این صفر و یکها را بتوان به صورت گروهی نمایش داد.

چگونگی نمایش به صورت گروهی را PLC می تواند به شکلهای مختلف نشان دهد. دسیمال، باینری، اکتال، هگزادسیمال، و باینری کد شده به دسیمال (BCD) از جمله حالت‌های نمایش گروهی می باشد. در ادامه به توضیح این سیستمهای اعداد پرداخته خواهد شد.

1-1-1- دسیمال (Decimal)

سیستم عددی دسیمال برای همه کس آشنا می باشد. در این سیستم از اعداد فقط از 10 عدد برای نمایش تعداد استفاده می گردد. در این سیستم از رقم های 0,1,2,3,4,5,6,7,8,9 برای نمایش اعداد استفاده می گردد. یکی از دلایل انتخاب مبنای اعداد به صورت دسیمال شاید وجود ده انگشت که نشان دهنده ده حالت مختلف از اعداد است، باشد. زمانی که از مبنای دسیمال استفاده می گردد هر رقم (از سمت راست به سمت چپ) به صورت افزایشی تعبیر کننده یکی از توانهای ده می باشد. به عنوان مثال عدد 106,324 نشان دهنده یکصد و شش هزار و سیصد و بیست و چهار واحد می باشد. تقسیم این عدد به صورت توان 10 به صورت زیر خواهد بود.

پایه	10^5	10^4	10^3	10^2	10^1	10^0
مقدار عددی	1	0	6	3	2	4

$$10^0 = 1,$$

$$10^1 = 10,$$

$$10^2 = 100,$$

$$10^3 = 1,000,$$

$$10^4 = 10,000,$$

$$10^5 = 100,000,$$

پس هر عدد باید در پایه اش ضرب گردد تا عدد نهایی به دست آید.

$$1 * 10^5 = 100,000$$

$$0 * 10^4 = 0$$

$$6 * 10^3 = 6,000$$

$$3 * 10^2 = 300$$

$$2 * 10^1 = 20$$

$$4 * 10^0 = 4$$

حاصل جمع این اعداد 106,324 خواهد بود.

1-1-2- باینری (Binary)

سیستمهای باینری فقط از دو رقم "0" و "1" استفاده می نمایند و بر مبنای 2 می باشند. PLC ها از این سیستم و روش شماره گذاری استفاده می نمایند زیرا این روش برای نمایش حالت‌های ورودی/خروجی ها مناسب می باشد. رقم "1" نشان دهنده حالت روشن و عدد "0" نشان دهنده حالت خاموش می باشد. بنابراین با اختصاص عدد "0" و یا "1" به یک خروجی باید آن خروجی خاموش و روشن گردد. این رقم ها را به صورت بیت (BIT: Binary Unit) تعبیر می کنند که کوچکترین بخش از اعداد باینری می باشد. اعداد باینری معمولا از 8 و یا 16 بیت تشکیل شده اند. هر هشت بیت از داده های باینری را یک بایت (Byte) و هر 16 بیت را یک Word و یا کانال (Channel) می نامند. 10011001 مثالی از یک عدد باینری می باشد. در این مثال 8 بخش وجود دارد که هر کدام نشان دهنده یکی از توانهای 2 می باشد.

پایه	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
مقدار عددی	1	0	0	1	1	0	0	1

$$2^0 = 1$$

$$2^1 = 2$$

$$2^2 = 4$$

$$2^3 = 8$$

$$2^4 = 16$$

$$2^5 = 32$$

$$2^6 = 64$$

$$2^7 = 128$$

برای نشان دادن این عدد باینری به صورت دسیمال باید به صورت زیر عمل نمود:

$$1 * 2^7 = 128$$

$$0 * 2^6 = 0$$

$$0 * 2^5 = 0$$

$$1 * 2^4 = 16$$

$$1 * 2^3 = 8$$

$$0 * 2^2 = 0$$

$$0 * 2^1 = 0$$

$$1 * 2^0 = 1$$

که مجموع آن عدد 153 می باشد.

مثال:

تبدیل نزولی عدد 010110 به صورت زیر است:

$$1 * 2^4 = 16$$

$$0 * 2^3 = 0$$

$$1 * 2^2 = 4$$

$$1 * 2^1 = 2$$

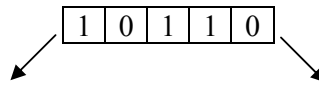
$$0 * 2^0 = 0$$

جمع این مقادیر 22 می باشد.

دو اصطلاح مهم که برای تعریف بیتها مورد استفاده قرار می گیرند:

پر ارزش ترین بیت: Most Significant Bit (MSB)

کم ارزش ترین بیت: Least Significant Bit (LSB)



پر ارزش ترین بیت: Most Significant Bit (MSB)

کم ارزش ترین بیت: Least Significant Bit (LSB)

هگزادسیمال (Hexadecimal)

برای ساده کردن استفاده از سیستم اعداد باینری سیستم اعداد هگزادسیمال طراحی شده است. این اعداد بر مبنای 16 می باشند که از رقم های 0 تا 9 و همچنین حروف A تا F (A=10 و F=15) تشکیل شده اند. همانطور که در زیر دیده می شود هر گروه که از 4 بیت تشکیل شده است 16 حالت مختلف روشن و خاموش بودن را شامل می شوند. هر کدام از این 16 حالت نشان دهنده یک عدد در مبنای هگزادسیمال خواهد بود.

F, E, D, C, B, A, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0

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

10F0 مثالی از اعداد هگزادسیمال می باشد. برای تبدیل این اعداد به معادل دسیمال آن باید هر کدام از این رقمها را بر پایه 16 ضرب نمود.

1 0 F 0

$$0 * 16^0 = 0$$

$$15 * 16^1 = 240$$

$$0 * 16^2 = 0$$

$$1 * 16^3 = 4096$$

مجموع این اعداد 4336 بر مبنای دسیمال خواهد شد، بدین معنی که این عدد معادل 4336 در مبنای دسیمال می باشد.

با استفاده از بیتهای چهار تایی و تبدیل هر چهار بیت به عدد هگز معادل آن می توان اعداد باینری را در مبنای هگزا دسیمال کد نمود.

هر چهار بیت داده های باینری را می توان به یک رقم هگزادسیمال تبدیل نمود:

1110 →→→ E

هر هشت بیت از داده های باینری را می توان به صورت دو رقم هگزادسیمال تبدیل نمود.

0010 0101 →→→ 2 5

نهایتاً هر شانزده بیت از داده های باینری را می توان به صورت چهار رقم هگزادسیمال تبدیل نمود.

0001 0010 0011 0100 →→→ 1 2 3 4

با توجه به اینکه کنترل کننده های شرکت **OMRON** از داده های 16 بیتی (Words) استفاده می نمایند، می توان حالت هر Word را به صورت چهار رقم هگزادسیمال نمایش داد.

16^3	16^2	16^1	16^0
1011 8421	0111 8421	1010 8421	1101 8421

به این سیستم اعداد باینری کد شده به هگزادسیمال (BCH) می گویند.

به عنوان مثال بیتهای تحت ارزش 16^0 تا 16^3 معادل اعداد زیر به هگز می باشند.

$$16^0 = 1 + 4 + 8 = D$$

$$16^1 = 2 + 8 = A$$

$$16^2 = 1 + 2 + 4 = 7$$

$$16^3 = 1 + 2 + 8 = B$$

بنابراین، معادل این 16 بیت در مبنای هگزادسیمال عدد B7AD می باشد

B 7 A D

$$13 * 16^0 = 13$$

$$10 * 16^1 = 160$$

$$7 * 16^2 = 1792$$

$$11 * 16^3 = 45056$$

معادل این عدد در مبنای دسیمال عدد 47021 می باشد.

سیستم اعداد BCH (باینری کد شده به هگزادسیمال) قابلیت نمایش رشته اعداد باینری بزرگ را به صورت 4 رقمی دارا می باشند.

اگر تعداد بیتهای باینری کد شده به هگزادسیمال از 16 بیت کمتر باشد می توان با افزودن صفر بعد از با ارزشترین بیت آن تعداد بیتها را به تعداد مجاز برای کد کردن به هگز رساند.

در مثال زیر یک داده 9 بیتی به داده ای 16 بیتی تبدیل شده است.

1 0110 1101 →→→ 0000 0001 0110 1101

نتیجه این داده باینری 16 بیتی عدد D 016 خواهد بود.

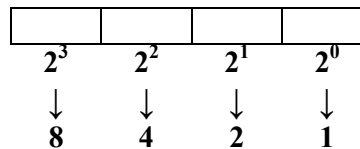
1-1-3- باینری کد شده به دسیمال (BCD)

اعداد باینری کد شده به دسیمال مانند داده های باینری کد شده به هگزادسیمال می باشند، تنها تفاوت بین

این اعداد این است که در سیستم BCD فقط از ده عدد اول اعداد هگزادسیمال (0 تا 9) استفاده می شود. در

این سیستم، اعداد بین A تا F مجاز نمی باشد.

- قوانین موجود در اعداد BCD به صورت زیر می باشد:
- هر چهار بیت معادل یک رقم BCD می باشد.
 - هر عدد BCD نمایشگر یک عدد در مبنای 10 می باشد. به عنوان مثال $10^0 = 1$ ، $10^1 = 10$ ، $10^2 = 100$ و ...
- برای تبدیل هر چهار بیت به معادل BCD آن کافی است ارزش هر یک از چهار بیت را آنچنانکه در زیر اشاره شده است، بدانیم.



10^3	10^2	10^1	10^0
0000	0100	0000	0100
8421	8421	8421	8421

$$4 * 10^0 = 4$$

$$0 * 10^1 = 0$$

$$4 * 10^2 = 400$$

$$0 * 10^3 = 0$$



عدد باینری 0100 0000 0100 را به صورت زیر می توان به BCD تبدیل نمود.

که معادل این عدد در مبنای دسیمال عدد 404 می باشد.
در جدول زیر سیستمهای عددی دسیمال، باینری و BCD با هم مقایسه شده اند.

اعداد دسیمال	اعداد باینری	اعداد باینری کد شده به دسیمال
0	0000	0000 0000
1	0001	0000 0001
2	0010	0000 0010
3	0011	0000 0011
4	0100	0000 0100
5	0101	0000 0101
6	0110	0000 0110
7	0111	0000 0111
8	1000	0000 1000
9	1001	0000 1001
10	1010	0001 0000

2- کارت های ورودی و خروجی آنالوگ

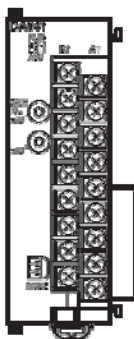
Analog Input Unit

CJ1W-AD041-V1
CJ1W-AD081-V1
CJ1W-AD081

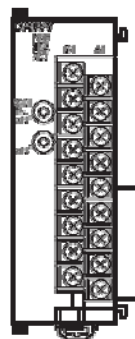


Analog Output Unit

CJ1W-DA021
CJ1W-DA041

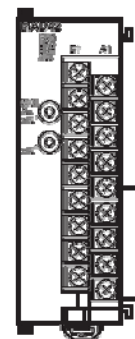


CJ1W-DA08V
CJ1W-DA08C



Analog I/O Unit

CJ1W-MAD42



کارت‌های ورودی و خروجی آنالوگ از سری کارت‌های ویژه ورودی/خروجی PLC ها می باشند، که استفاده از ورودی و خروجی های آنالوگ را با دقت های بالا فراهم می سازند. کارت‌های ورودی آنالوگ مانند CJ1W-AD041/081 سیگنال ورودی آنالوگ را به داده ای باینری تبدیل نموده و این داده تبدیل شده را به CPU منتقل می نماید، در عوض کارت‌های خروجی آنالوگ مانند CJ1W-DA041 داده های دیجیتال را به فرمت سیگنال آنالوگ به خروجی های آنالوگ منتقل می نماید.

کارت‌های آنالوگ مانند CJ1W-MAD42 به صورت همزمان استفاده از سیگنال ورودی و خروجی آنالوگ را برای کاربر فراهم می سازد. این کارت‌ها همچنین دارای دقت بالا و توابع مختلفی جهت سهولت استفاده از سیگنال‌های آنالوگ می باشند، توابعی همچون "مشخصه قطعی ورودی ها"، "نگه داشتن مقدار ماکزیمم"، "مقیاس گذاری سیگنال آنالوگ" و ... از جمله مواردی است که می توان در تنظیمات کارت‌های آنالوگ لحاظ نمود.

برای آشنایی با چگونگی انجام تنظیمات کارت‌های آنالوگ و همچنین چگونگی تبدیل داده های ورودی و خروجی آنالوگ و توابع آنها، در ادامه تنظیمات مربوط به کارت آنالوگ CJ1W-MAD42 که دارای 4 ورودی آنالوگ و 2 خروجی آنالوگ می باشد و همچنین ورودی/خروجی های آنالوگ موجود روی PLC مدل CP1H شرح داده می شود.

تنظیمات دیگر کارت‌های آنالوگ با اندکی تفاوت مشابه با تنظیمات کارت CJ1W-MAD42 می باشد.

1-2- مشخصات کلی کارت ورودی/خروجی آنالوگ سری CJ1 (CJ1W-MAD42)

2-1-1- مشخصات

Item	CJ1W-MAD42
Unit type	CJ-series Special I/O Unit
Isolation	Between I/O and PLC signals: Photocoupler (No isolation between individual I/O signals.)
External terminals	18-point detachable terminal block (M3 screws)
Current consumption	580 mA max. at 5 V DC
Dimensions (mm) (See note 1.)	31 x 90 x 85 (W x H x D)
Weight	150 g max.
General specifications	Conforms to general specifications for SYSMAC CJ-series Series.
Mounting position	CJ-series CPU Rack or CJ-series Expansion Rack (Cannot be mounted to a C200H Expansion I/O Rack or a SYSMAC BUS Slave Rack.)
Maximum number of Units (See note 2.)	CPU Rack: 7 Units max. Expansion Rack: 8 Units max. Overall system: (7 Units max. on CPU Rack) + (8 Units per Expansion Rack x 3 Racks) = 31 Units max.
Data exchange with CPU Units	Special I/O Unit Area CIO 200000 to CIO295915 (Words CIO 2000 to CIO 2959): Exchanges 10 words of data per Unit. Internal Special I/O Unit DM Area (D20000 to D29599)

نکته : حداکثر تعداد کارتهای ورودی/خروجی آنالوگ که می توان در یک Rack از PLC نصب نمود، بسته به منبع تغذیه PLC و همچنین مصرف جریان مابقی کارتهای نصب شده روی Rack متفاوت می باشد.

Power Supply Units	Maximum number of Units
CJ1W-PA205R/PD025	CPU Rack: 7 Units max. Expansion Racks: 8 Units/Rack max.
CJ1W-PA202	CPU Rack: 3 Units max. Expansion Racks: 4 Units/Rack max.

مشخصات ورودی آنالوگ و توابع آن

Item	Voltage Input	Current Input
Number of analog inputs	4	
Input signal range (See note 3.)	1 to 5 V 0 to 5 V 0 to 10 V -10 to 10 V	4 to 20 mA (See note 4.)
Maximum rated input (for 1 point) (See note 5.)	±15 V	±30 mA
External input impedance	1 MΩ min.	250 Ω (rated value)
Resolution	4,000/8,000 (full scale) (See note 8.)	
Converted output data	16-bit binary data	
Accuracy (See note 6.)	25°C	±0.2% of full scale
	0°C to 55°C	±0.4% of full scale
A/D conversion time (See note 7.)	1.0 ms/500 μs max. per point	
Mean value processing	Stores the last "n" data conversions in the buffer, and stores the mean value of the conversion values. Buffer number: n = 2, 4, 8, 16, 32, 64	
Peak value holding	Stores the maximum conversion value while the Peak Value Hold Bit is ON.	

Item	Voltage Input	Current Input
Scaling	Enabled only for conversion time of 1 ms and resolution of 4,000. Setting any values within a range of ±32,000 as the upper and lower limits allows the A/D conversion result to be output with these values as full scale.	
Input disconnection detection	Detects the disconnection and turns ON the Disconnection Detection Flag.	

نکته:

- محدوده سیگنال ورودی و خروجی را می توان برای هر ورودی و یا خروجی به صورت جداگانه تعیین نمود.
- ورودی جریان و یا ولتاژ را می توان با استفاده از سویچ ولتاژ/جریان موجود در زیر ترمینالها تنظیم نمود.
- ورودی/خروجی های آنالوگ با توجه به مشخصات ذکر شده در جدول بالا به کار گرفته می شود، استفاده از آنها در شرایطی غیر از موارد زیر باعث آسیب ورودی/خروجی های آنالوگ خواهد شد.
- دقت با توجه به مقیاس کامل می باشد.
- زمان تبدیل A/D زمانی است که طول می کشد تا سیگنال آنالوگ ورودی به صورت یک داده باینری در حافظه PLC ذخیره گردد. قبل از اینکه مقدار تبدیل شده توسط واحد CPU خوانده شود حداقل زمانی به اندازه یک زمان دوره به طول می انجامد.
- با استفاده از تنظیمات D(m+18) می توان دقت را 8000 انتخاب نمود در نتیجه زمان تبدیل به 500µs تقلیل می یابد.

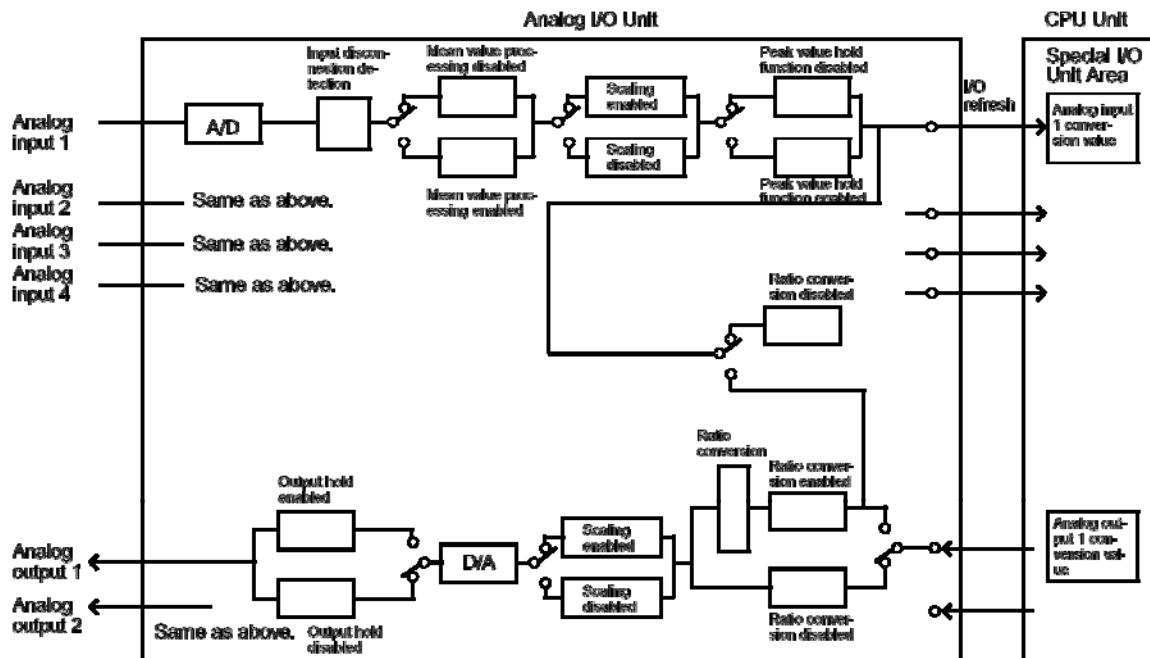
مشخصات خروجی

Item	Voltage output	Current output
Number of analog outputs	2	
Output signal range (See note 1.)	1 to 5 V 0 to 5 V 0 to 10 V -10 to 10 V	4 to 20 mA
External output impedance	0.5 Ω max.	---
Maximum external output current (for 1 point)	2.4 mA	---
Maximum allowed load resistance	---	600 Ω
Resolution	4,000/8,000 (full scale) (See note 5.)	
Set data	16-bit binary data	
Accuracy (See note 2.)	25°C	±0.3% of full scale
	0°C to 55°C	±0.5% of full scale
D/A conversion time (See note 3.)	1.0 ms/500 µs max. per point	
Output hold function	Outputs the specified output status (CLR, HOLD, or MAX) under any of the following circumstances. When the Conversion Enable Bit is OFF. (See note 4.) In adjustment mode, when a value other than the output number is output during adjustment. When there is an output setting error or a fatal error occurs at the PLC. When the CPU Unit is on standby. When the Load is OFF.	
Item	Voltage output	Current output
Scaling	Enabled only for conversion time or 1 ms and resolution of 4,000. Setting any values within a range of ±32,000 as the upper and lower limits allows D/A conversion to be executed and analog signals to be output with these values as full scale.	
Ratio conversion function	Stores the results of positive and negative gradient analog inputs calculated for ratio and bias as analog output values. Positive gradient: Analog output = A × Analog input + B (A = 0 to 99.99, B = 8,000 to 7FFF hex) Negative gradient: Analog output = F - A × Analog input + B (A: 0 to 99.99, B = 8,000 to 7FFF hex, F: Output range maximum value)	

نکته:

- محدوده سیگنال ورودی و خروجی را می توان برای هر ورودی و یا خروجی به صورت جداگانه تعیین نمود.
- دقت با توجه به مقیاس کامل می باشد.
- زمان تبدیل D/A زمانی که برای تبدیل داده های باینری به سیگنال آنالوگ خروجی به طول می انجامد. حداقل به اندازه یک زمان دوره طول می کشد تا داده باینری موجود در حافظه PLC به صورت سیگنال آنالوگ خروجی در خروجی های PLC نمایان گردد.
- زمانی که مد عملکرد PLC از Run و یا Monitor به مد تغییر وضعیت دهد و یا بیت مربوط به فعال نمودن خروجی های PLC خاموش باشد، سیگنال خروجی با توجه به تنظیمات مربوط به حالت خروجی معین می گردد.
- با استفاده از تنظیمات $D(m+18)$ می توان دقت را 8000 انتخاب نمود در نتیجه زمان تبدیل به $500\mu s$ تقلیل می یابد.

2-2- نمودار عملکرد ورودی/خروجی

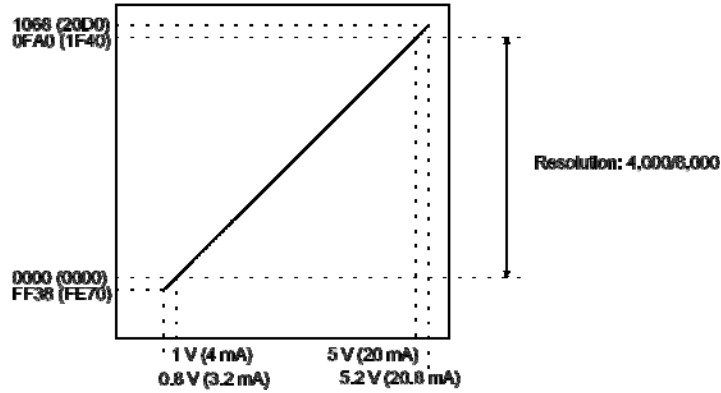


2-3- مشخصات ورودی

اگر سیگنال ورودی خارج از محدوده تعیین شده به عنوان ورودی باشد، مقدار تبدیل شده حداقل و یا حداکثر سیگنال ورودی خواهد بود.

محدوده: 1 تا 5V (4 تا 20mA)

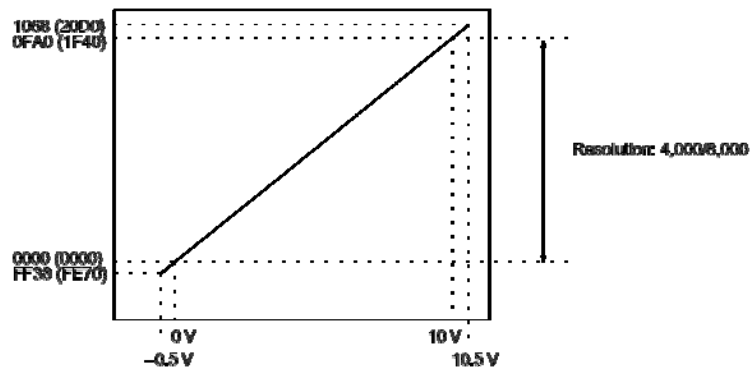
مقدار تبدیل شده (داده 16 بیتی باینری)



(: داده های داخل پرانتز مربوط به دقت 8000 می باشد.

محدوده: 0 تا 10V

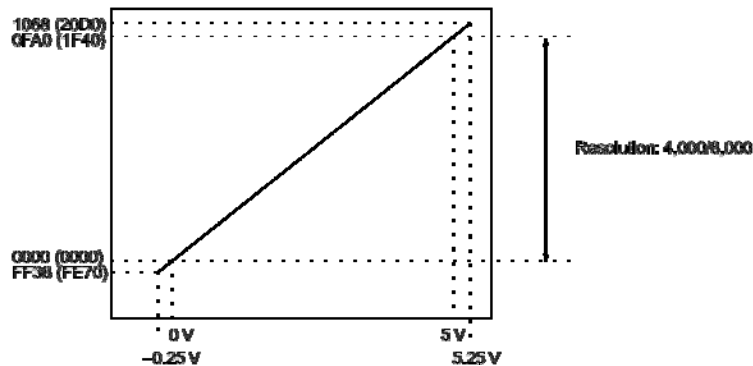
مقدار تبدیل شده (داده 16 بیتی باینری)



(: داده های داخل پرانتز مربوط به دقت 8000 می باشد.

محدوده: 0 تا 5V

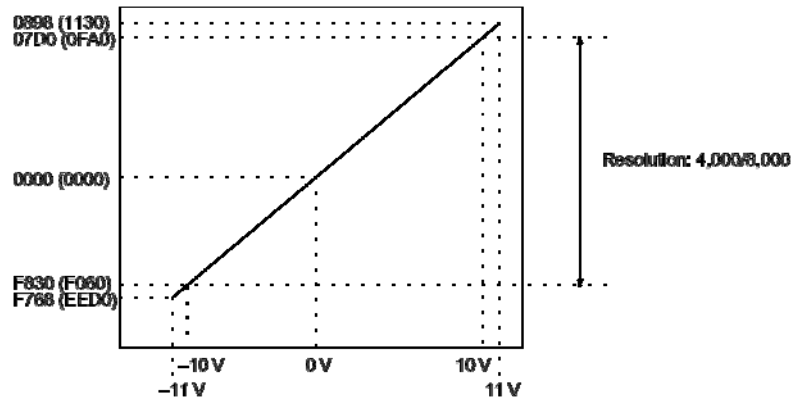
مقدار تبدیل شده (داده 16 بیتی باینری)



(: داده های داخل پرانتز مربوط به دقت 8000 می باشد.

محدوده: -10 تا 10V

مقدار تبدیل شده (داده 16 بیتی باینری)

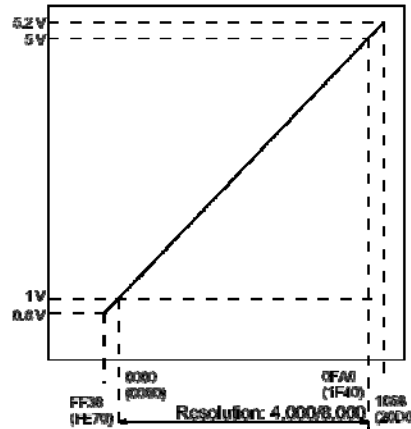


() : داده های داخل پرانتز مربوط به دقت 8000 می باشد.

4-2- مشخصات خروجی

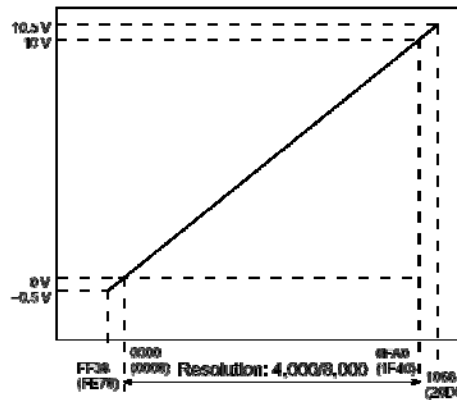
اگر مقدار خروجی از محدوده تنظیم شده خارج شود، خروجی در حداقل و یا حداکثر محدوده ثابت باقی خواهد ماند.

محدوده: 1 تا 5V



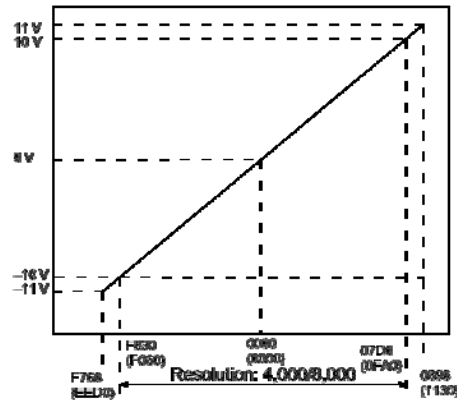
() : داده های داخل پرانتز مربوط به دقت 8000 می باشد.

محدوده: 0 تا 10V



() : داده های داخل پرانتز مربوط به دقت 8000 می باشد.

محدوده: 10- تا 10V



() : داده های داخل پرانتز مربوط به دقت 8000 می باشد.

تبدیل داده ها و مقادیر برای محدوده 10- تا 10 ولت به صورت زیر می باشد:

16-bit binary data	BCD (Resolution: 4,000)
F768	-2200
:	:
FFFF	-1
0000	0
0001	1
:	:
0898	2200

2-5- رویه عملکرد

برای استفاده از کارتهای آنالوگ باید رویه عملکرد زیر انجام گیرد.

نصب و تنظیمات

1. تنظیم سویچ ولتاژ/جریان موجود در زیر ترمینالها.
2. سیم بندی ورودی و خروجی های آنالوگ.
3. تنظیم Unit Number با استفاده از سویچ های موجود در روی کارت.
4. وصل کردن تغذیه PLC.
5. تعریف کردن جدول ورودی/خروجی.
6. تنظیمات مربوط به کارتهای ویژه در ناحیه DM.
 - تنظیم تعداد ورودی و خروجی های استفاده شده.
 - تنظیم محدوده سیگنال ورودی/خروجی ها
 - تنظیم تعداد نمونه های مورد نظر برای فرآیند میانگین گیری.
 - تنظیم توابع مربوط به حالت خروجی.
 - تنظیم توابع مقیاس گذاری.
 - تنظیم نسبت تبدیل و مقدار بایاس.
 - تنظیم زمان تبدیل و دقت.

7. خاموش و روشن کردن PLC و یا روشن و خاموش کردن بیت متناظر برای راه اندازی دوباره کارتهای ویژه.

اگر المان ورودی و یا خروجی متصل به کارت آنالوگ نیاز به کالیبراسیون و تنظیم داشته باشد باید رویه زیر را دنبال نمود.

تنظیمات افست و بهره

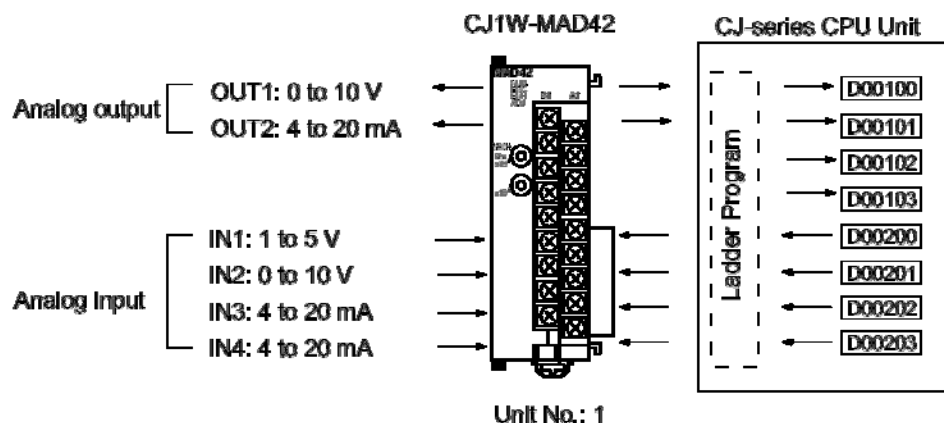
1. تنظیم سویچ ولتاژ/جریان موجود در زیر ترمینالها.
2. وصل کردن تغذیه PLC.
3. انتخاب Adjustment Mode با استفاده از تنظیمات کارتهای ویژه در ناحیه DM.
4. خاموش و روشن کردن PLC و یا روشن و خاموش کردن بیت متناظر برای راه اندازی دوباره کارتهای ویژه.
5. تنظیم افست و بهره.
6. انتخاب Normal Mode با استفاده از تنظیمات کارتهای ویژه در DM.
7. راه اندازی مجدد کارت آنالوگ با استفاده از بیت مربوط به راه اندازی مجدد کارتهای ویژه و یا خاموش و روشن کردن PLC.

عملکرد

برنامه نردبانی:

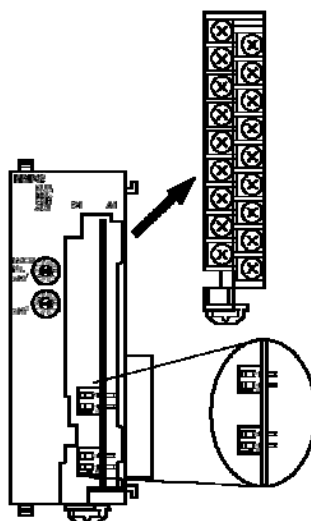
- خواندن مقادیر تبدیل شده و یا نوشتن تنظیمات با استفاده از دستور MOV(021) و یا XFER(070).
- شروع یا توقف تبدیل خروجی.
- تعیین تابع مربوط به نگاه داشتن مقدار حداکثر.
- به دست آوردن اخطارهای مربوط به قطعی و تنظیمات و همچنین خطاها.

2-5-1- مثالهای چگونگی عملکرد

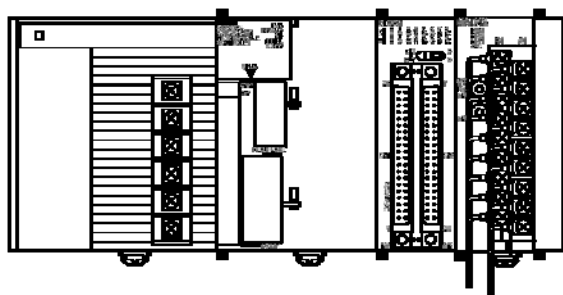


تنظیمات کارت ورودی/خروجی آنالوگ

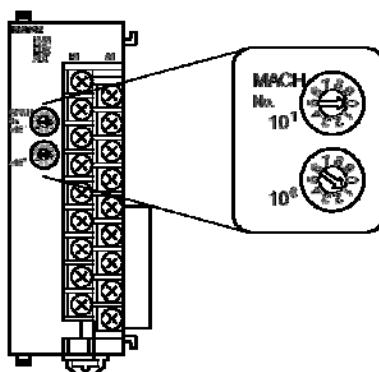
1. تنظیم سویچ های مربوط به ولتاژ و جریان.



2. نصب و سیم بندی کارت ورودی/خروجی آنالوگ.



3. تنظیم سویچ مربوط به Unit Number.



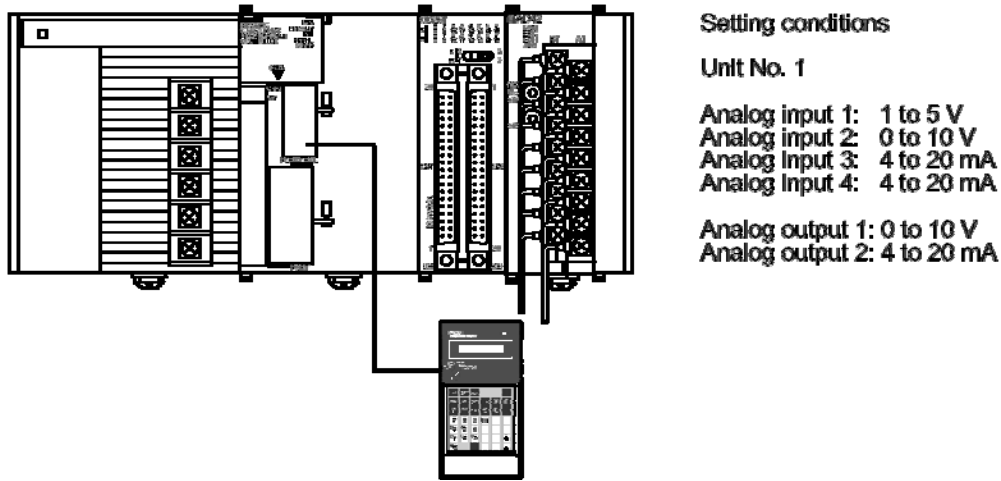
4. روشن کردن تغذیه PLC.

تعریف جدول ورودی/خروجی ها

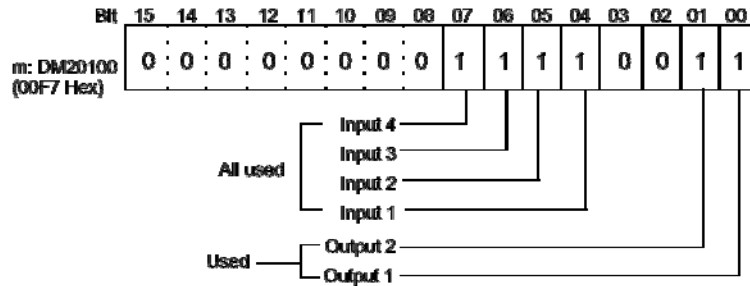
بعد از وصل نمودن تغذیه PLC، باید از تعریف جدول ورودی/خروجی اطمینان حاصل نمود.

انجام تنظیمات

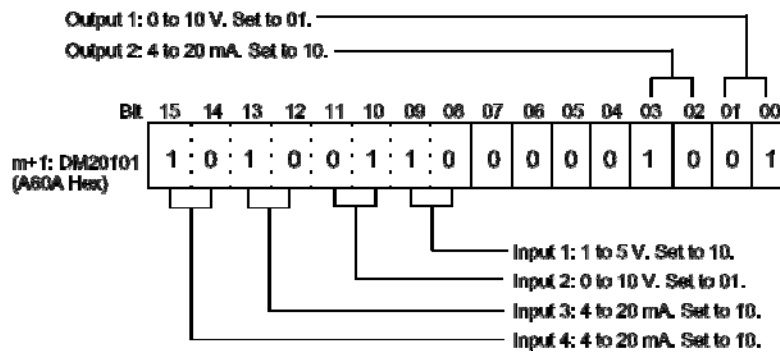
1. تعیین تنظیمات مربوط به کارتهای ورودی/خروجی ویژه در ناحیه DM از حافظه PLC.



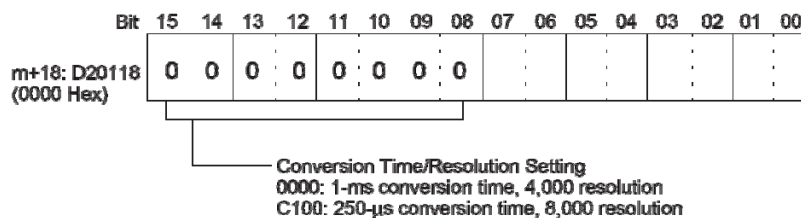
- این بخش از تنظیمات مربوط به ورودی و خروجی هایی است که مورد استفاده قرار گرفته اند.



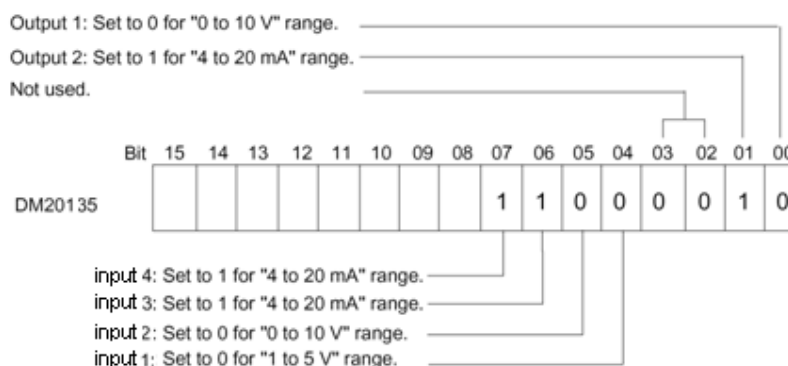
- این بخش از تنظیمات مربوط به محدوده سیگنال های ورودی و خروجی می باشد.



- تنظیم زمان تبدیل و دقت.



- تنظیم ولتاژ و یا جریان بودن ورودی و خروجی ها.



2. خاموش و روشن کردن CPU جهت راه اندازی مجدد.

نوشتن برنامه نردبانی

1. با استفاده از مثال زیر چگونگی استفاده از ورودی های آنالوگ و همچنین نوشتن برنامه نردبانی شرح داده می شود.

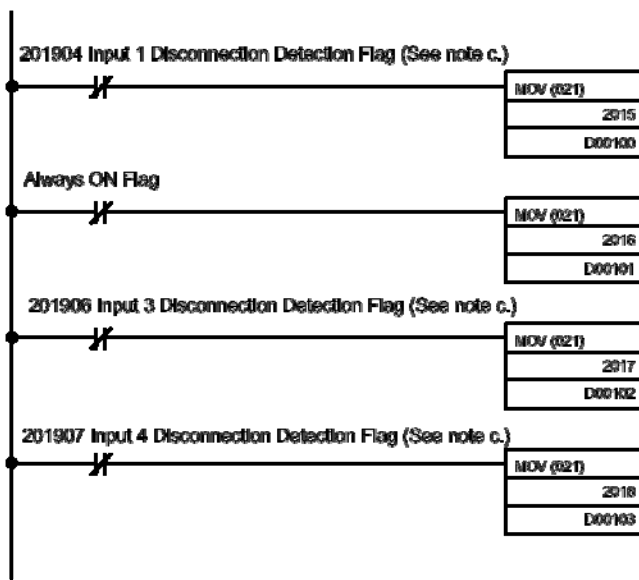
در این مثال داده های آنالوگ تبدیل شده به دیجیتال که در آدرسهای CIO (n+5) تا CIO (n+8) از حافظه مربوط به ورودی/خروجی های ویژه قرار گرفته اند (CIO 2015 تا CIO 2018)، به صورت یک داده باینری بین Hex 0000 تا Hex 0FA0 در آدرسهای D100 تا D103 ذخیره می شوند. جدول زیر آدرس های استفاده شده برای ورودی های آنالوگ را نشان می دهد.

Input number	Input signal range	Input conversion value address (n = CIO 2010) (See note 1.)	Conversion data holding address (See note 2.)
1	1 to 5 V	(n+5) = CIO 2015	D00100
2	0 to 10 V	(n+6) = CIO 2016	D00101
3	4 to 20 mA	(n+7) = CIO 2017	D00102
4	4 to 20 mA	(n+8) = CIO 2018	D00103

نکته:

- آدرسها با توجه به Unit Number تنظیم شده روی کارت آنالوگ معین می گردد.

- تنظیمات بسته به نیاز انجام می گیرد.



برای ورودی 1 تا 5V مقدار باینری بین Hex 0000 تا Hex 0FA0 در آدرس CIO 2015 ذخیره می گردد. پس اگر قطعی وجود نداشته باشد این داده های در آدرس D100 قرار خواهد گرفت.

همچنین برای ورودی 0 تا 10V مقدار باینری آدرس CIO 2016 در D101 قرار می گیرد. در این محدوده فلگ مشخصه قطعی معتبر نخواهد بود. در نتیجه استفاده از این فلگ به عنوان شرط عملکرد دستورالعمل MOV مجاز نمی باشد.

همین روال برای ورودی 4 تا 20mA نیز وجود دارد.

- فلگ مشخصه قطعی ورودی ها بیت 04 تا 07 آدرس CIO (n+9) خواهد بود.

2. با استفاده از مثال زیر چگونگی استفاده از خروجی های آنالوگ و همچنین نوشتن برنامه نردبانی شرح داده می شود.

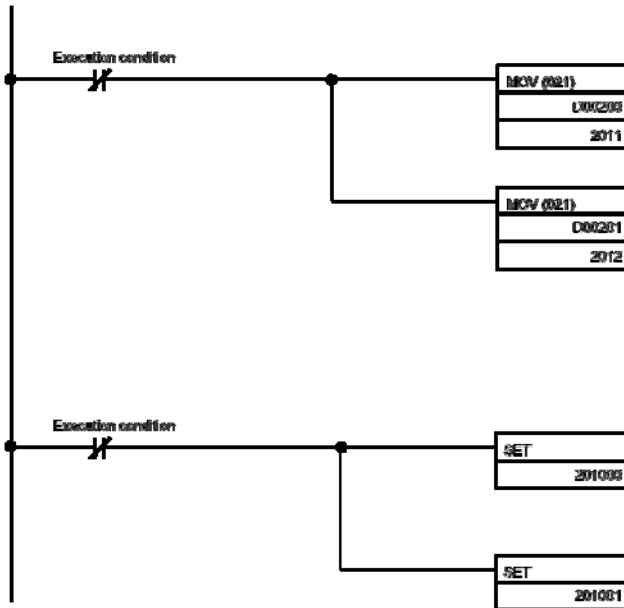
داده های باینری در محدوده 0000Hex تا 0FA0Hex موجود در آدرس D200 و D201 به آدرسهای CIO (n+1) و CIO (n+2) منتقل می گردد تا به عنوان سیگنال خروجی در خروجی های آنالوگ نمایان گردند.

جدول زیر آدرس های استفاده شده برای خروجی های آنالوگ را نشان می دهد.

Output number	Input signal range	Output setting address (n = CIO 2010) See note 1.	Original conversion address
1	0 to 10 V	(n+1) = CIO 2011	D00200
2	4 to 20 mA	(n+2) = CIO 2012	D00201

نکته:

- آدرسها با توجه به Unit Number تنظیم شده روی کارت آنالوگ معین می گردد.
- تنظیمات بسته به نیاز انجام می گیرد.



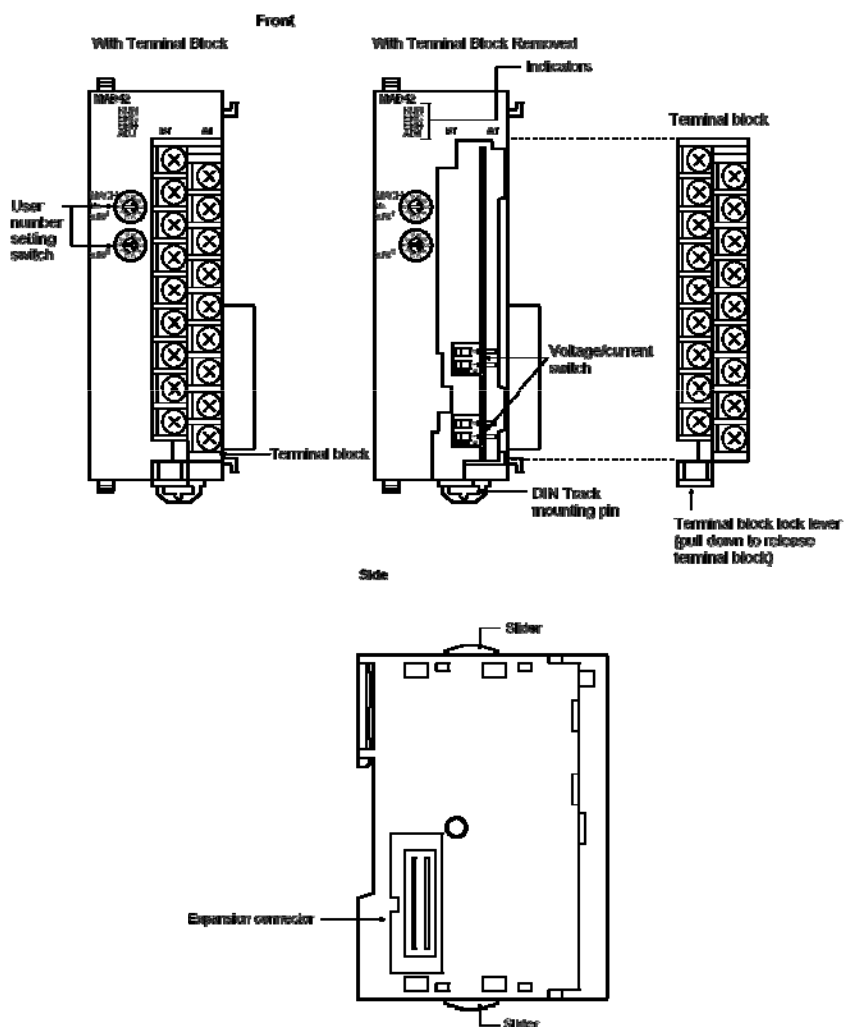
محتوای D200 در حافظه CIO 2011 قرار خواهد گرفت.

محتوای D201 در حافظه CIO 2012 قرار خواهد گرفت.

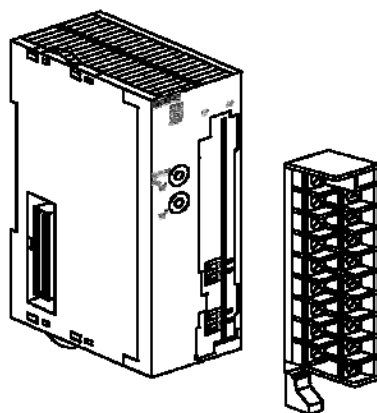
برای فعال کردن خروجی های آنالوگ، محتوای بیت فعال کننده خروجی ها که بیت 2010.00 برای ورودی اول و بیت 2010.01 برای ورودی دوم می باشد را 1 نمایید.

داده های موجود در آدرس های CIO 2011 و همچنین CIO 2012 در خروجی ها به صورت سیگنال آنالوگ 0 تا 10V و سیگنال 4 تا 20mA نمایان خواهد شد.

6-2- جزئیات و تنظیمات مربوط به سویچ ها



ترمینالها با استفاده از مکانیزم اتصالی خاص به کارت متصل می گردد. قابلیت جدا شدن از کارت را با پایین کشیدن اهرم موجود در بلوک ترمینال را دارند. در حالت عادی این اهرم باید بالا باشد و قبل از شروع کار باید از بالا بودن این اهرم اطمینان حاصل نمود.



2-6-1- نمایشگرها

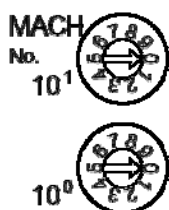
نمایشگرها حالت و چگونگی عملکرد کارت را نمایش می دهند. در جدول زیر مفهوم هر یک از نمایشگرها بیان شده است.

LED	مفهوم	نمایشگر	حالت عملکرد
RUN (سبز رنگ)	عملکرد نرمال	روشن	عملکرد در مد نرمال.
		خاموش	تبادل اطلاعات بین کارت آنالوگ و CPU متوقف شده است.
ERC (قرمز رنگ)	خطای نمایان شده توسط کارت آنالوگ	روشن	خطایی (همچون مشخصه قطعی) روی داده و یا تنظیمات آغازین نادرست بوده است.
		خاموش	عملکرد به صورت نرمال.
ADJ (زرد رنگ)	کالیبره کردن	روشن	عملکرد در حالت تنظیم آفست و بهره در مد Adjustment.
		خاموش	عملکرد در حالتی غیر از مد Adjustment.
ERH (قرمز رنگ)	خطای موجود در CPU	روشن	در تبادل اطلاعات بین کارت آنالوگ و CPU خطایی روی داده است.
		خاموش	عملکرد به صورت نرمال.

2-6-2- سویچ های Unit Number

کارت ورودی/خروجی آنالوگ و CPU داده های موجود در ناحیه های حافظه که مربوط به کارتهای ورودی/خروجی ویژه می باشند را مبادله می کنند. آدرسهایی که توسط کارت آنالوگ در ناحیه CIO و همچنین DM اشغال می شود توسط سویچ های مربوط به Unit Number موجود روی کارت آنالوگ معین می گردد.

برای تنظیم Unit Number باید تغذیه PLC قطع شود. با استفاده از یک آچار پیچ کوشتی کوچک می توان بدون آسیب رساندن به سویچ مربوط به Unit Number این سویچ را روی عدد مورد نظر تنظیم نمود. نباید سویچ را در وضعیتی بین تنظیمات رها کرد.

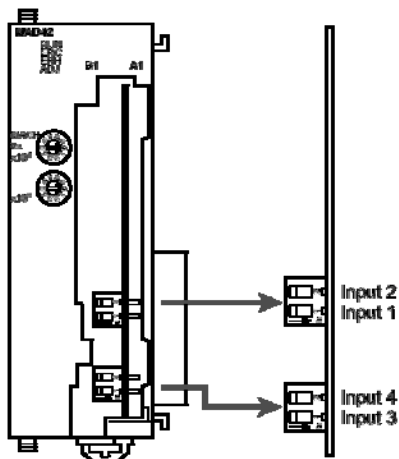


Switch setting	Unit number	Special I/O Unit Area addresses	Special I/O Unit DM Area addresses
0	Unit #0	CIO 2000 to CIO 2009	D20000 to D20099
1	Unit #1	CIO 2010 to CIO 2019	D20100 to D20199
2	Unit #2	CIO 2020 to CIO 2029	D20200 to D20299
3	Unit #3	CIO 2030 to CIO 2039	D20300 to D20399
4	Unit #4	CIO 2040 to CIO 2049	D20400 to D20499
5	Unit #5	CIO 2050 to CIO 2059	D20500 to D20599
6	Unit #6	CIO 2060 to CIO 2069	D20600 to D20699
7	Unit #7	CIO 2070 to CIO 2079	D20700 to D20799
8	Unit #8	CIO 2080 to CIO 2089	D20800 to D20899
9	Unit #9	CIO 2090 to CIO 2099	D20900 to D20999
10	Unit #10	CIO 2100 to CIO 2109	D21000 to D21099
--	--	--	--
n	Unit #n	CIO 2000 + (n x 10) to CIO 2000 + (n x 10) + 9	D20000 + (n x 100) to D20000 + (n x 100) + 99
--	--	--	--
95	Unit #95	CIO 2950 to CIO 2959	D29500 to D29599

نکته: اگر به دو یا چند کارت ورودی/خروجی ویژه یک Unit Number اختصاص داده شود. خطایی توسط CPU ایجاد می شود (بیت A401.13 روشن می گردد) و عملکرد PLC متوقف خواهد شد.

2-6-3- سویچ جریان/ولتاژ

با استفاده از دیپ سویچ های موجود در زیر بلوک ترمینال می توان تبدیل ورودی آنالوگ را بین دو حالت ولتاژ و جریان انتخاب نمود.



توجه : باید قبل از گذاشتن و یا برداشتن بلوک ترمینال از قطع بودن تغذیه PLC اطمینان حاصل نمود.

2-7-7- سیم بندی

2-7-1- ترتیب ترمینالها

ترتیب ترمینالها و همچنین ترمینالهای متناظر با هر ورودی و خروجی در شکل زیر نشان داده شده اند.

Voltage output 2 (+)	B1	A1	Voltage output 1 (+)
Output 2 (-)	B2	A2	Output 1 (-)
Current output 2 (+)	B3	A3	Current output 1 (+)
N.C.	B4	A4	N.C.
Input 2 (+)	B5	A5	Input 1 (+)
Input 2 (-)	B6	A6	Input 1 (-)
AG	B7	A7	AG
Input 4 (+)	B8	A8	Input 3 (+)
Input 4 (-)	B9	A9	Input 3 (-)

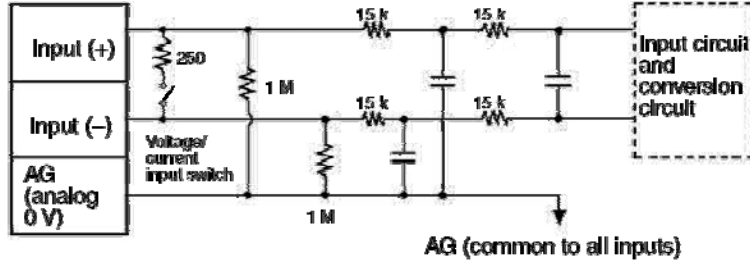
نکته:

1. تعداد ورودی/خروجی هایی که مورد استفاده قرار می گیرد را می توان در تنظیمات DM تعیین نمود.
2. محدوده سیگنالهای ورودی و خروجی را می توان به صورت جداگانه برای هر ورودی و خروجی با استفاده از تنظیمات DM معین نمود.
3. ترمینالهای AG (A7 و B7) به 0V مدار آنالوگ داخل کارت متصل شده اند. استفاده از کابل های پوشش دار برای ورودی های آنالوگ باعث افزایش مقاومت در برابر نویز خواهد شد.
4. ترمینالهای N.C. (A4 و B4) به مدارهای داخلی کارت آنالوگ متصل نمی باشند.

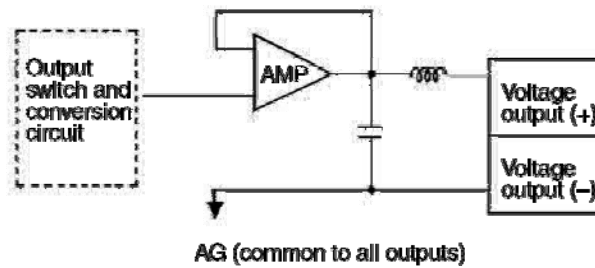
2-7-2- ترتیب مدارهای داخلی

در شکل‌های زیر مدارهای داخلی کارتهای آنالوگ ورودی/خروجی نمایش داده شده است.

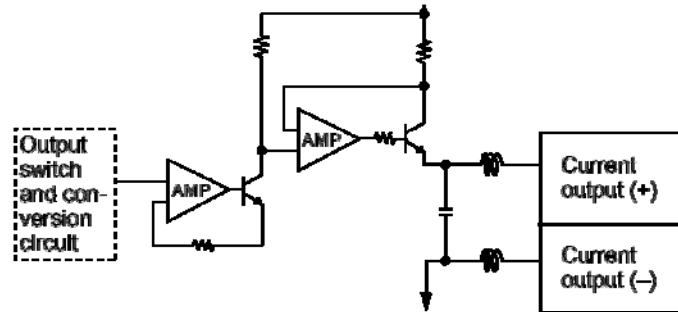
مدار ورودی



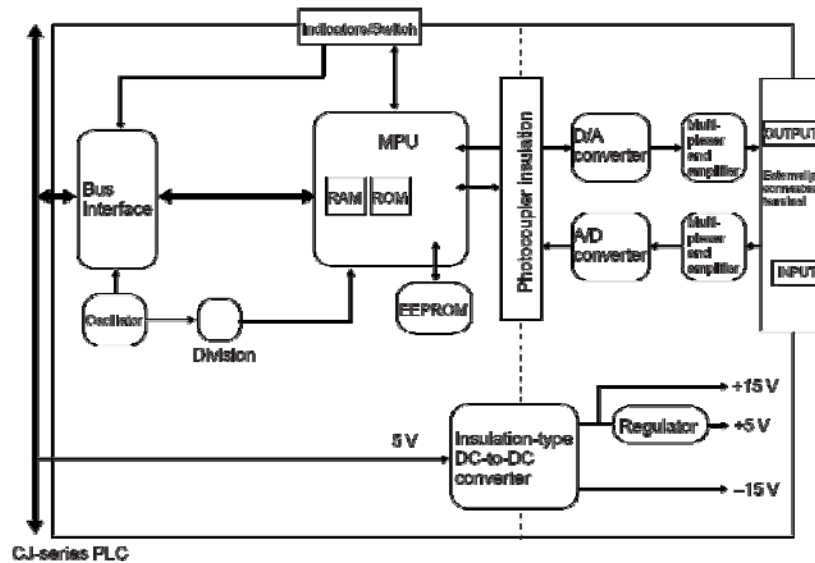
مدار خروجی



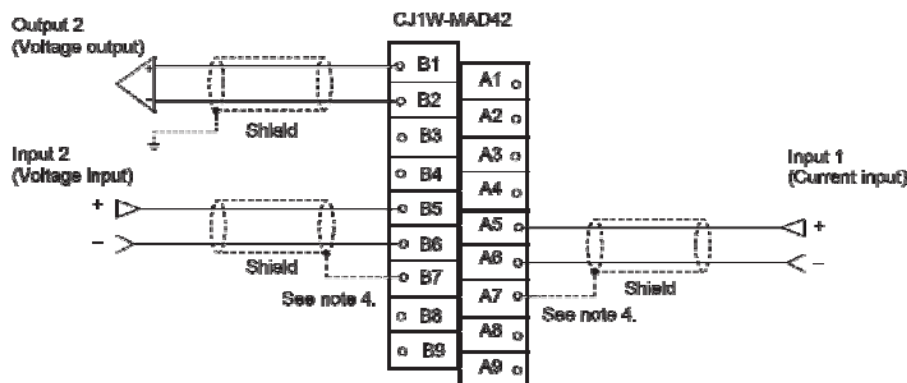
مدار خروجی جریان



مدارات داخلی

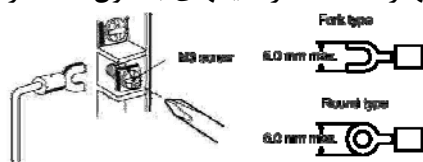


2-7-3- مثالی از سیم بندی ورودی/خروجی ها



نکته :

1. وقتی از ورودی جریان استفاده می شود باید سویچ مربوط به ولتاژ/جریان ورودی آنالوگ اول را روشن نمود. همچنین در تنظیمات کارتهای ویژه در ناحیه Data Memory از حافظه، باید بیت مربوط به ورودی اول را در $D(m+35)$ را 1 نمود.
2. برای ورودی هایی که مورد استفاده قرار نگرفته اند، علاوه بر این که می توان بیت مربوط به استفاده شدن و یا نشدن ورودی ها را در تنظیمات "0:Not used" انتخاب نمود می توان ترمینالهای مربوط به این ورودی ها ($V+$) و ($V-$) را نیز اتصال کوتاه کرد.
3. در کارتهای آنالوگ ترمینالهای پیچی مورد استفاده قرار می گیرد، در نتیجه باید از محکم بودن پیچ ترمینالها اطمینان حاصل شود.
4. زمانی که پوشش کابل های ورودی آنالوگ به ترمینالهای AG ($A7$ و $B7$) متصل می شود، (همانطور که در شکل دیده می شود) بهتر است که از سیمهای با طول حداکثر 30 سانتی متر استفاده شود.



اتصال پوشش کابل های ورودی آنالوگ به ترمینالهای AG باعث افزایش مقاومت سیگنالهای ورودی در برابر نویز می گردد. برای کاهش اثر نویز در خروجی ها باید سیمهای خروجی را زمین نمود.

2-7-4- ملاحظات در سیم بندی ورودی/خروجی ها

- در هنگام سیم بندی ورودی و خروجی های آنالوگ می توان با رعایت موارد زیر تأثیر نویز را حتی الامکان کاهش و عملکرد ورودی/خروجی ها را بهینه نمود.
- برای اتصالات خارجی باید از کابل دو رشته بهم پیچیده استفاده نمود.
 - مسیر کابل های ورودی/خروجی آنالوگ را باید از کابلهای AC و همچنین کابل های مدارات اصلی و یا کابل های ولتاژ بالا جدا نمود.
 - اگر تداخل نویز در خطوط تغذیه ایجاد شده است باید از یک نویز فیلتر در مسیر ورودی منبع تغذیه استفاده شود.

2-8- تبادل اطلاعات با واحد CPU

2-8-1- نگاه کلی بر تبادل داده ها

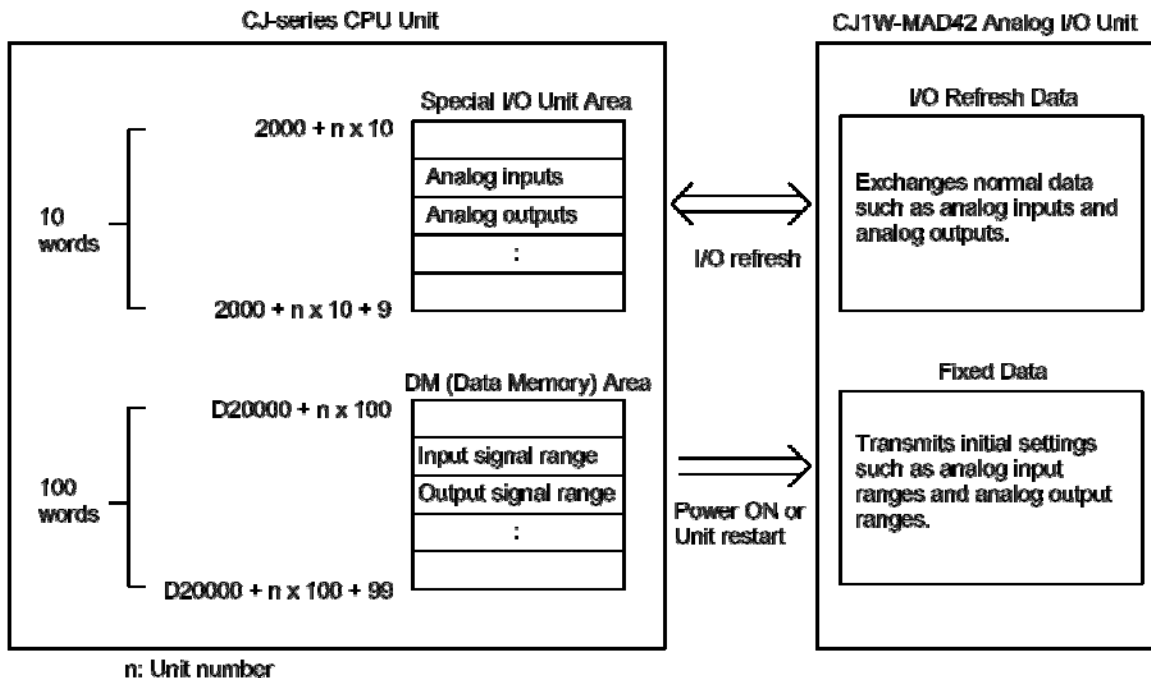
داده های مربوط به کارتهای ورودی/خروجی ویژه در ناحیه CIO (داده های مورد استفاده برای عملکرد کارت آنالوگ) و در ناحیه DM (داده های مورد استفاده برای تنظیمات ابتدایی کارت آنالوگ) بین واحد CPU و همچنین کارت آنالوگ مبادله می شود.

تجدید داده های ورودی/خروجی

داده های تبدیل شده به باینری سیگنالهای ورودی آنالوگ، مقدارهای خروجی آنالوگ و دیگر داده هایی که برای عملکرد کارت آنالوگ مورد استفاده قرار می گیرند، بسته به شماره کارت (Unit Number) در ناحیه ای از حافظه که به کارتهای ویژه اختصاص داده شده است قرار می گیرند و در طول عملکرد کارت آنالوگ و تجدید داده های ورودی/خروجی بین CPU و کارت آنالوگ مبادله می شوند.

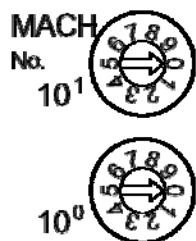
داده های ثابت

داده های ثابت کارت آنالوگ، همچون محدوده سیگنالهای ورودی و خروجی در بخشی از حافظه Data Memory که به کارتهای ورودی/خروجی ویژه اختصاص داده شده است قرار می گیرد. این داده های در هنگام روشن کردن PLC و یا زمان آغاز کار مجدد کارت آنالوگ بین CPU و کارت آنالوگ مبادله می شود.



2-8-2- تنظیمات مربوط به Unit Number

آدرس‌های Wordهایی که هر کارت آنالوگ در ناحیهٔ مربوط به ورودی/خروجی‌های ویژه از حافظه‌های CIO و DM اشغال می‌کند توسط سویچ مربوط به Unit Number که روی کارت آنالوگ قرار دارد تعیین می‌گردد.



Switch setting	Unit number	Special/IO Unit Area addresses	Special IO Unit DM Area addresses
0	Unit #0	CIO 2000 to CIO 2009	D20000 to D20099
1	Unit #1	CIO 2010 to CIO 2019	D20100 to D20199
2	Unit #2	CIO 2020 to CIO 2029	D20200 to D20299
3	Unit #3	CIO 2030 to CIO 2039	D20300 to D20399
4	Unit #4	CIO 2040 to CIO 2049	D20400 to D20499
5	Unit #5	CIO 2050 to CIO 2059	D20500 to D20599
6	Unit #6	CIO 2060 to CIO 2069	D20600 to D20699
7	Unit #7	CIO 2070 to CIO 2079	D20700 to D20799
8	Unit #8	CIO 2080 to CIO 2089	D20800 to D20899
9	Unit #9	CIO 2090 to CIO 2099	D20900 to D20999
10	Unit #10	CIO 2100 to CIO 2109	D21000 to D21099
~	~	~	~
n	Unit #n	CIO 2000 + (n x 10) to CIO 2000 + (n x 10) + 9	D20000 + (n x 100) to D20000 + (n x 100) + 99
~	~	~	~
95	Unit #95	CIO 2950 to CIO 2959	D29500 to D29599

نکته: اگر به دو یا چند کارت ورودی/خروجی ویژه یک Unit Number اختصاص داده شود. خطایی توسط CPU ایجاد می‌شود (بیت A401.13 روشن می‌گردد) و عملکرد PLC متوقف خواهد شد.

2-8-3- تنظیمات مربوط به مد عملکرد

مد عملکرد را می‌توان بین دو حالت Normal و Adjustment (برای انجام تنظیمات مربوط به آفست و بهره) انتخاب نمود. این انتخاب با تغییر تنظیمات بیت 00 تا 07 آدرس D(m+18) امکان پذیر خواهد بود.

تنظیمات D(m+18)

DM word	Bits															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
D(m+18)	Conversion time/resolution setting								Operation mode setting 00: Normal mode C1: Adjustment mode							

$$m: 20000 + (\text{unit number} \times 100)$$

2-8-4- بیت های راه اندازی مجدد کارت آنالوگ

برای راه اندازی مجدد کارت آنالوگ بعد از اعمال تغییرات در تنظیمات کارت که در ناحیه DM انجام می گیرد و یا بعد از تصحیح خطاهای روی داده، باید PLC را خاموش و روشن نمود یا اینکه بیت مربوط به راه اندازی مجدد کارت آنالوگ را با توجه به Unit Number آن روشن و خاموش کرد.

Special I/O Unit Area word address	Function	
A50200	Unit No. 0 Restart Bit	Restarts the Unit when turned ON and then OFF again.
A50201	Unit No. 1 Restart Bit	
~	~	
A50215	Unit No. 15 Restart Bit	
A50300	Unit No. 16 Restart Bit	
~	~	
A50715	Unit No. 95 Restart Bit	

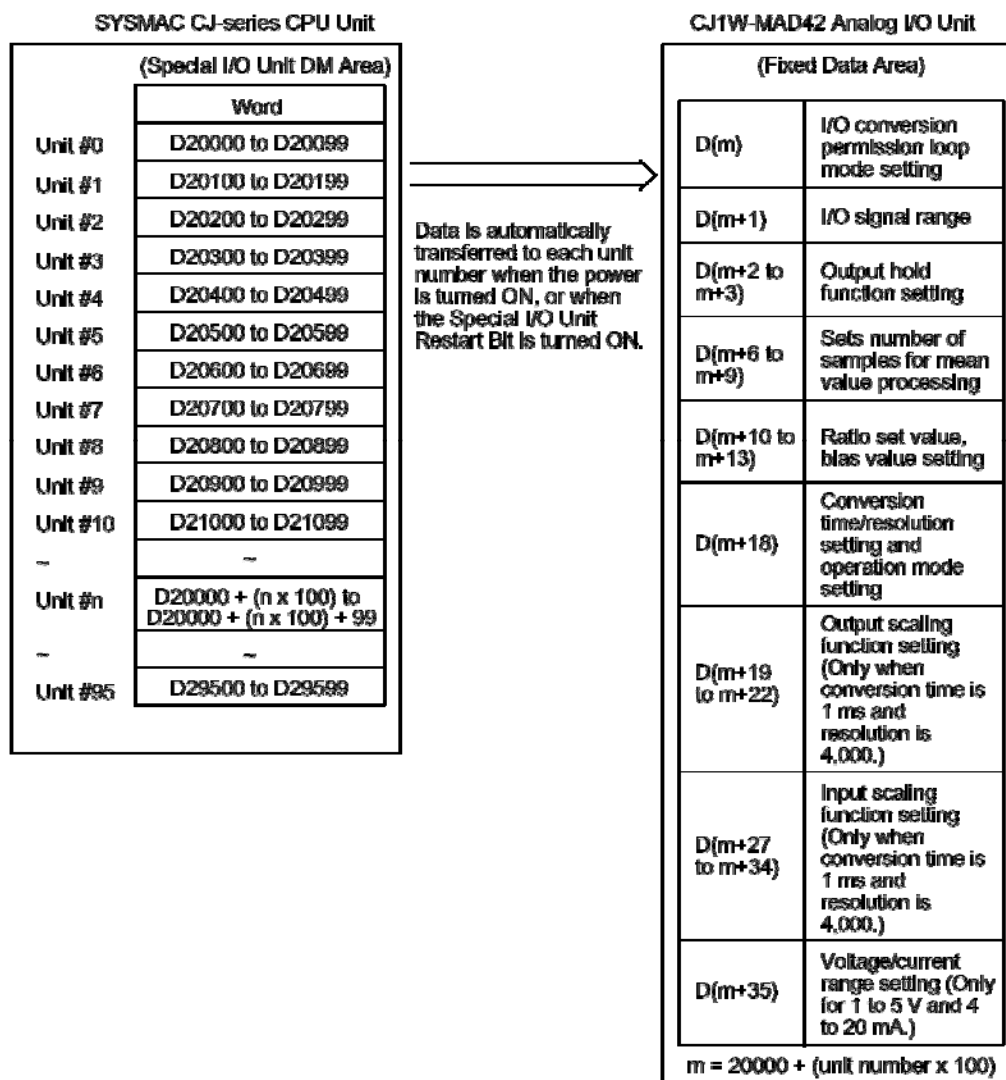
نکته: اگر خطای روی داده با روشن و خاموش کردن PLC و یا روشن و خاموش کردن بیت مربوط به راه اندازی مجدد کارت آنالوگ از بین نرود باید کارت تعویض گردد.



2-8-5- تخصیص داده های ثابت

DM و محتوای آن

تنظیمات اولیه کارت های ورودی/خروجی آنالوگ بسته به آدرس های اختصاص داده شده به کارت در ناحیه DM از حافظه PLC انجام می شود. تنظیماتی همچون تعیین ورودی/خروجی هایی که مورد استفاده قرار گرفته اند، محدوده سیگنالهای ورودی و خروجی و ... در این ناحیه انجام می گیرد.



نکته: - آدرسهایی از ناحیه DM که به کارتهای آنالوگ اختصاص داده می شوند با توجه به

Unit Number که روی کارت آنالوگ وجود دارد تعیین می گردد.

- اگر به دو یا چند کارت ورودی/خروجی ویژه یک Unit Number اختصاص داده شود.

خطایی توسط CPU ایجاد می شود (بیت A401.13 روشن می گردد) و عملکرد PLC

متوقف خواهد شد.

جزئیات تنظیمات در ناحیه DM

در جدول زیر جزئیات تنظیمات موجود در ناحیه DM برای هر دو حالت عملکرد Normal و Adjustment آورده شده است.

DM word	Bits															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
D(m)	Ratio conversion use setting								Input use setting				Output use setting			
	Not used.		Not used.		Loop 2		Loop 1		Input 4	Input 3	Input 2	Input 1	Not used.	Not used.	Output 2	Output 1
D(m+1)	Input signal range setting								Output signal range setting							
	Input 4		Input 3		Input 2		Input 1		Not used.		Not used.		Output 2		Output 1	
D(m+2)	Not used.								Output 1: Output status when conversion stopped							
D(m+3)	Not used.								Output 2: Output status when conversion stopped							
D(m+4)	Not used.															
D(m+5)	Not used.															
D(m+6)	Input 1: Mean value processing setting															
D(m+7)	Input 2: Mean value processing setting															
D(m+8)	Input 3: Mean value processing setting															
D(m+9)	Input 4: Mean value processing setting															
D(m+10)	Loop 1 (input 1 to output 1), A constant															
D(m+11)	Loop 1 (input 1 to output 1), B constant															
D(m+12)	Loop 2 (input 2 to output 2), A constant															
D(m+13)	Loop 2 (input 2 to output 2), B constant															
D(m+14)	Not used.															
D(m+15)	Not used.															
D(m+16)	Not used.															
D(m+17)	Not used.															
D(m+18)	Conversion time and resolution setting								Operation mode setting							
D(m+19)	Output 1 scaling lower limit (Enabled only for conversion time of 1 ms and resolution of 4,000.)															
D(m+20)	Output 1 scaling upper limit (Enabled only for conversion time of 1 ms and resolution of 4,000.)															
D(m+21)	Output 2 scaling lower limit (Enabled only for conversion time of 1 ms and resolution of 4,000.)															
D(m+22)	Output 2 scaling upper limit (Enabled only for conversion time of 1 ms and resolution of 4,000.)															
D(m+23)	Not used.															
D(m+24)	Not used.															
D(m+25)	Not used.															
D(m+26)	Not used.															
D(m+27)	Output 1 scaling lower limit															
D(m+28)	Output 1 scaling upper limit															
D(m+29)	Output 2 scaling lower limit															
D(m+30)	Output 2 scaling upper limit															
D(m+31)	Output 3 scaling lower limit															
D(m+32)	Output 3 scaling upper limit															
D(m+33)	Output 4 scaling lower limit															
D(m+34)	Output 4 scaling upper limit															
D(m+35)	Voltage/current range setting (Enabled only when set for 1 to 5 V, 4 to 20 mA)															
	Not used.								Input 4	Input 3	Input 2	Input 1	Not used.		Output 2	Output 1

مقادیر تنظیمات

Item	Contents	Page	
Input	Use setting	0: Not used. 1: Used.	304
	Input signal range	00: -10 to 10 V 01: 0 to 10 V 10: 1 to 5 V, 4 to 20 mA (See note 1.) 11: 0 to 5 V	304
	Voltage/current range setting	0: Voltage range (1 to 5 V) 1: Current range (4 to 20 mA)	
	Mean value processing setting	0000: Mean value processing for 2 buffers (See note 3.) 0001: No mean value processing 0002: Mean value processing for 4 buffers 0003: Mean value processing for 8 buffers 0004: Mean value processing for 16 buffers 0005: Mean value processing for 32 buffers 0006: Mean value processing for 64 buffers	306
	Scaling setting	Set any value in binary data from -32,000 (8,300) to +32,000 (7D00), except when upper limit = lower limit (not 0000).	
	Output	Use setting	0: Not used. 1: Used.
Output signal range		00: -10 to 10 V 01: 0 to 10 V 10: 1 to 5 V 11: 0 to 5 V	313
Voltage/current range setting		0: Voltage range (1 to 5 V) 1: Current range (4 to 20 mA)	
Output status when stopped		00: CLR Outputs 0 or minimum value of each range. (See note 2.) 01: HOLD Holds output just before stopping. 02: MAX Outputs maximum value of range.	316
Scaling setting		Set any value in binary data from -32,000 (8,300) to +32,000 (7D00), except when upper limit = lower limit (not 0000).	
Loop		Ratio conversion use setting	00: Not used. 01: Uses positive gradient conversion. 10: Uses negative gradient conversion. 11: Same as for setting "00" above.
	A constant	4 digits BCD (0 to 9999)	
	B constant	16-bit binary data	
Conversion time/resolution setting (for inputs and outputs)		00: Conversion time of 1 ms and resolution of 4,000 C1: Conversion time of 500 μs and resolution of 8,000	306

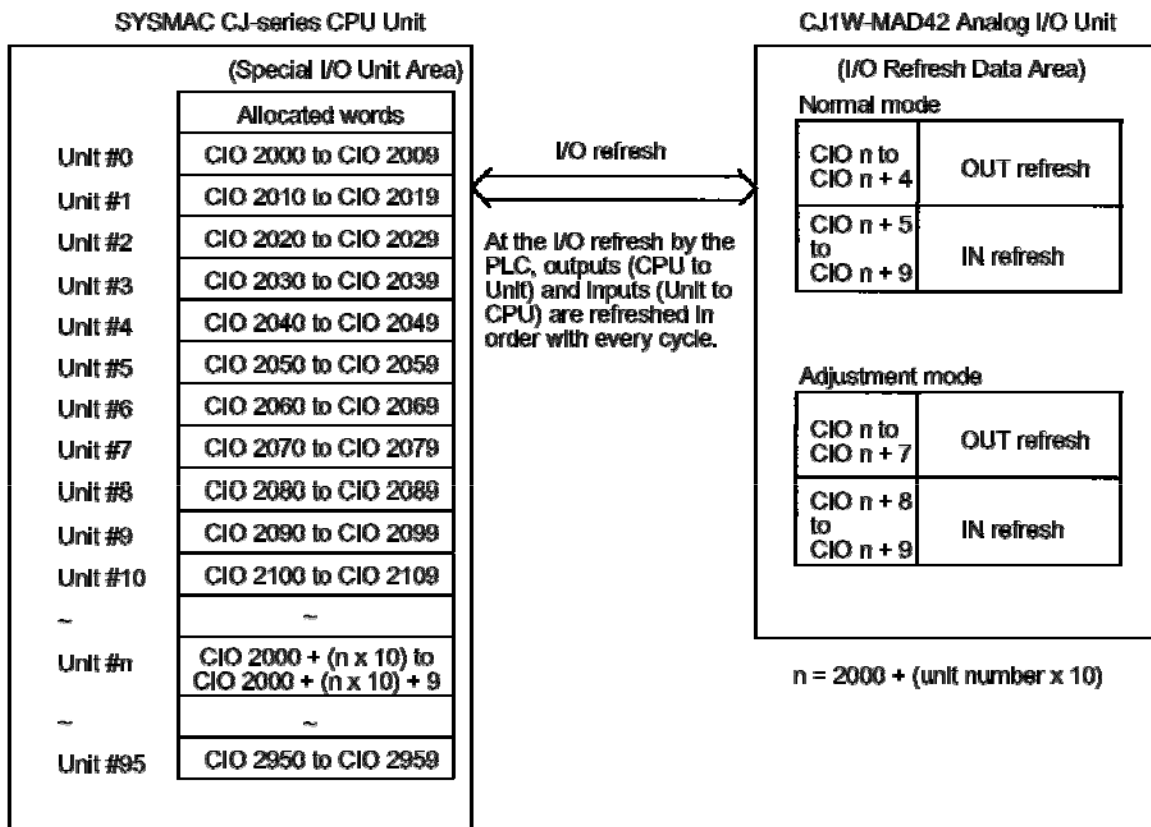
نکته:

1. برای سیگنال ورودی " 1 تا 5V " و " 4 تا 20mA " با استفاده از دیپ سویچ های ولتاژ/جریان موجود در زیر ترمینالها بین حالت ولتاژ و جریان می توان سویچ نمود.
2. برای محدوده 10- تا 10 ولت، سیگنال خروجی 0V خواهد بود. برای دیگر محدوده های سیگنال خروجی حداقل سیگنال به عنوان خروجی در حالت CLR در خروجی ها قرار می گیرد.
3. برای حالت پردازش مقدار میانگین (Mean Value Processsing) تعداد میانگیرهای از پیش تعریف شده 2 میانگیر می باشد.

2-8-6- داده های مربوط به تجدید حالت ورودی/خروجی ها

تخصیص و محتوای حافظه های ورودی/خروجی های ویژه در CIO

داده های مربوط به تجدید حالت ورودی/خروجی ها بسته به تعریفی که برای آنها در حافظه مربوط به ورودی/خروجی های آنالوگ در محیط CIO شده است، تبادلی می شوند. داده های تبدیل شده ورودی و همچنین داده های باینری خروجی بین CPU و کارتهای آنالوگ برای تجدید ورودی و خروجی ها مبادله می شوند



نکته :

- حافظه هایی از محدوده ورودی/خروجی های ویژه که توسط یک کارت آنالوگ اشغال میگردند، با استفاده از سویچ های مربوط به Unit Number که در روی کارت وجود دارد تنظیم می گردد.
- اگر به دو یا چند کارت ورودی/خروجی ویژه یک Unit Number اختصاص داده شود. خطایی توسط CPU ایجاد می شود (بیت A401.13 روشن می گردد) و عملکرد PLC متوقف خواهد شد.

چگونگی تخصیص آدرسها در حالت Normal

جهت عملکرد در مد Normal، بیت‌های 00 تا 07 حافظه D(m+18) را باید مقدار Hex 00 قرار داد. در جدول زیر چگونگی تخصیص بیتها و Word های منطقه CIO نشان داده شده است.

I/O	Word	Bits																										
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0											
Output (CPU to Unit)	n	Not used.								Peak value hold				Not used.		Conversion enable												
						Input 4				Input 3				Input 2				Input 1		Out-put 2		Out-put 1						
	n + 1	Output 1 set value																										
		16 ³				16 ²				16 ¹				16 ⁰														
	n + 2	Output 2 set value																										
n + 3	Not used.																											
n + 4	Not used.																											
Input (Unit to CPU)	n + 5	Input 1 conversion value / Loop 1 calculation result																										
		16 ³				16 ²				16 ¹				16 ⁰														
	n + 6	Input 2 conversion value / Loop 2 calculation result																										
	n + 7	Input 3 conversion value																										
	n + 8	Input 4 conversion value																										
n + 9	Alarm Flags								Disconnection detection								Output setting error											
										Input 4				Input 3				Input 2				Input 1				Out-put 2		Out-put 1

مقادیر تنظیمات

I/O	Item	Contents
Input	Peak value hold function	0: Not used. 1: Peak value hold used.
	Conversion value Calculation result	16-bit binary data
	Disconnection detection	0: No disconnection 1: Disconnection
	Output	0: Conversion output stopped. 1: Conversion output begun.
Output	Set value	16-bit binary data
	Output setting error	0: No error 1: Output setting error
	Common	Alarm Flags

نکته: برای حافظه های CIO مقدار n با استفاده از رابطه "n=2000+unit number×10" به دست می آید. استفاده از تابع مشخصه قطعی ورودی، فقط برای دو محدوده 1 تا 5V و 4 تا 20mA امکان پذیر می باشد.

Input signal range	Voltage/current
1 to 5 V	0.3 V max.
4 to 20 mA	1.2 mA max.

چگونگی تخصیص آدرسها در حالت Adjustment

جهت عملکرد در Adjustment Mode، بیت‌های 00 تا 07 حافظه D(m+18) را باید مقدار C1 Hex قرار داد. در جدول زیر چگونگی تخصیص بیتها و Word های منطقه CIO نشان داده شده است.

I/O	Word	Bits																		
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Output (CPU to Unit)	n	Not used.								Inputs and outputs to be adjusted										
						16^1				16^0										
	n + 1	Not used.								Not used.	Clr	Set	Up	Down	Gain	Off-set				
	n + 2	Not used.																		
	n + 3	Not used.																		
	n + 4	Not used.																		
	n + 5	Not used.																		
	n + 6	Not used.																		
Input (Unit to CPU)	n + 7	Not used.																		
	n + 8	Conversion value or set value at time of adjustment																		
		16^3				16^2				16^1				16^0						
	n + 9	Alarm Flags								Disconnection detection				Not used.						
									Input 4	Input 3	Input 2	Input 1								

مقادیر تنظیمات

Item	Contents
Input or output to be adjusted	Sets input or output to be adjusted. Leftmost digit: 1 (output) or 2 (input) Rightmost digit: 1 to 2 (output)/ 1 to 4 (input)
Offset (Offset Bit)	When ON, adjusts offset error.
Gain (Gain Bit)	When ON, adjusts gain error.
Down (Down Bit)	Decrements the adjustment value while ON.
Up (Up Bit)	Increments the adjustment value while ON.
Set (Set Bit)	Sets adjusted value and writes to EEPROM.
Clr (Clear Bit)	Clears adjusted value. (Returns to default status)
Conversion value for adjustment	The conversion value for adjustment is stored as 16 bits of binary data.
Disconnection detection	0: No disconnection 1: Disconnection
Alarm Flags	Bit 12: Input value is outside adjustment limits (In adjustment mode) Bit 13: I/O number setting error (in adjustment mode) Bit 14: EEPROM write error (in adjustment mode) Bit 15: Operating in adjustment mode. (Always ON in adjustment mode.)

نکته: برای حافظه های CIO مقدار n با استفاده از رابطه "n=2000+unit number×10" به دست می آید. استفاده از تابع مشخصه قطعی ورودی، فقط برای دو محدوده 1 تا 5V و 4 تا 20mA امکان پذیر می باشد.

Input signal range	Voltage/current
1 to 5 V	0.3 V max.
4 to 20 mA	1.2 mA max.

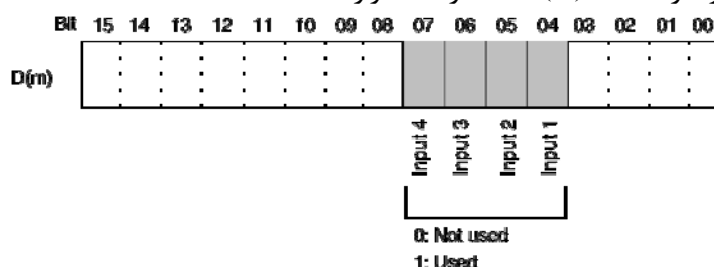
2-9- توابع ورودی های آنالوگ و شیوه عملکرد آنها

2-9-1- تنظیمات ورودی و تبدیل مقادیر

تنظیمات ورودی ها و محدوده آنها

تعداد ورودی ها:

کارتهای ورودی/خروجی آنالوگ، فقط ورودی های آنالوگ مشخص شده با شماره 0 تا 4 را تبدیل می نمایند. برای تعیین ورودی هایی که مورد استفاده قرار می گیرند باید بیت مربوط به هر ورودی را که در شکل زیر نشان داده شده است را در حافظه D(m)، مقدار "1" قرار داد.



فاصله زمانی نمونه برداری از ورودی های آنالوگ را می توان با قرار دادن "0" در بیت های مربوط به ورودی های استفاده نشده کاهش داد.

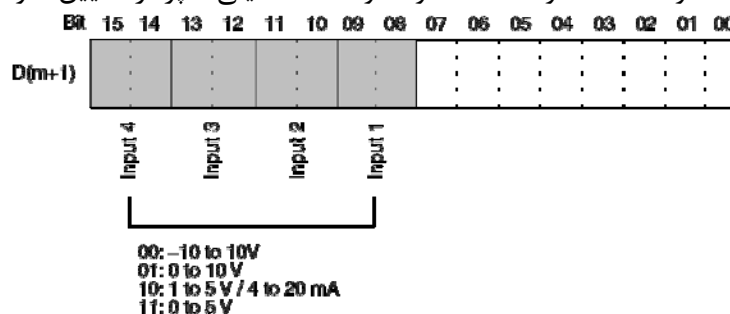
فاصله زمانی نمونه برداری = $1ms * \text{ورودی های آنالوگ استفاده شده}$.

نکته: در حالتی که دقت 8000 تعیین گردد زمان تبدیل $500\mu s$ خواهد بود، در نتیجه در رابطه بالا $1ms$ به $500\mu s$ تبدیل خواهد شد.

برای تعیین حافظه های منطقه DM مربوط به هر کارت آنالوگ، $m = 20000 + \text{unit number} \times 100$.

محدوده سیگنالهای ورودی:

برای هر یک از ورودی های آنالوگ می توان یکی از چهار محدوده سیگنال مربوط به ورودی های آنالوگ (-10 تا 10 ولت، 0 تا 10 ولت، 0 تا 5 ولت، 1 تا 5 ولت و 4 تا 20 میلی آمپر) را تعیین نمود.



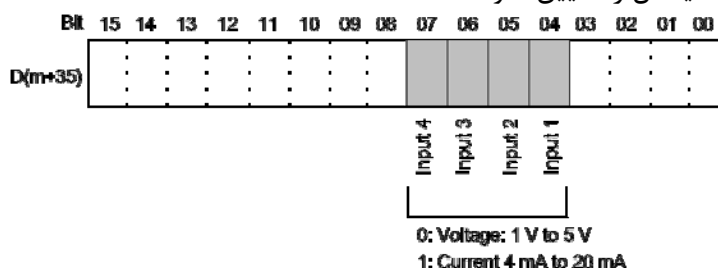
نکته:

- برای آدرسهای حافظه DM مقدار m از رابطه $m = 20000 + \text{unit number} \times 100$ به دست می آید.
- با استفاده از سویچ مربوط به ولتاژ/جریان، می توان برای محدوده 1 تا 5V و 4 تا 20mA، بین حالت ولتاژ و یا جریان انتخاب نمود.

- بعد از انجام تنظیمات ناحیه DM جهت انتقال این تنظیمات به کارت آنالوگ باید PLC را خاموش و روشن نمود و یا اینکه بیت مربوط به راه اندازی مجدد کارتهای ویژه را با در نظر گرفتن Unit Number روشن و خاموش نمود.

تنظیمات محدوده جریان و ولتاژ:

زمانی که محدوده سیگنال ورودی "1 تا 4/5V تا 20mA" انتخاب می گردد، باید با استفاده از D(m+35) جریان و یا ولتاژ بودن سیگنال را تعیین نمود.



خواندن مقادیر تبدیل شده:

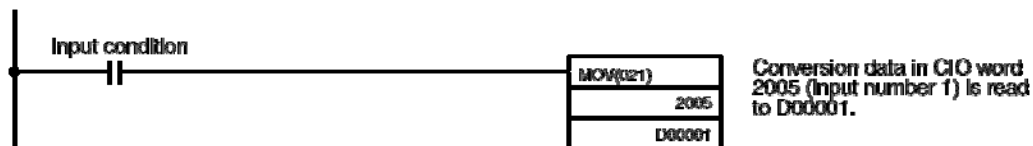
مقدارهای تبدیل شده سیگنالهای ورودی آنالوگ برای هر ورودی در آدرسهای CIO(n+5) تا CIO(n+8) قرار می گیرد.

Word	Function	Stored value
n+5	Input 1 conversion value	16-bit binary data
n+6	Input 2 conversion value	
n+7	Input 3 conversion value	
n+8	Input 4 conversion value	

نکته: برای حافظه های CIO مقدار n با استفاده از رابطه "n=2000+unit number×10" به دست می آید. برای خواندن مقادیر تبدیل شده سیگنالهای ورودی می توان از دستورات عملهای MOV(021) و XFER(070) در برنامه استفاده نمود.

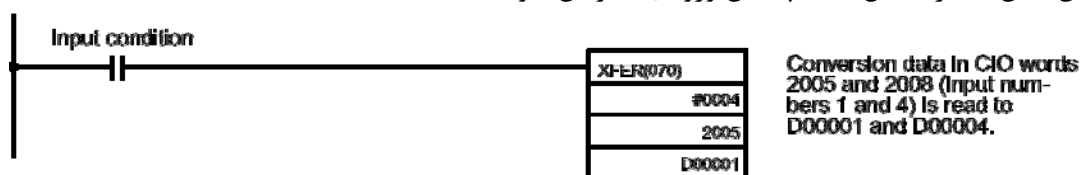
مثال 1:

در مثال زیر مقدار تبدیل شده یکی از ورودی ها خوانده شده است.



مثال 2:

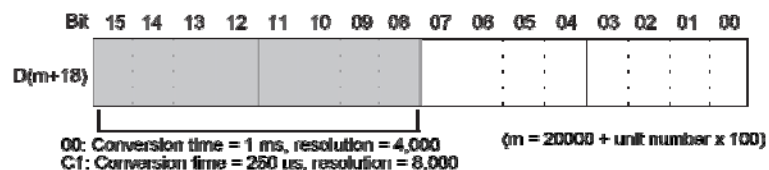
در این مثال مقدار تبدیل شده چندین ورودی همزمان خوانده شده است.



2-9-2- تنظیمات زمان تبدیل و دقت

بیت 08 تا 15 حافظه D(m+18) برای تنظیمات مربوط به زمان تبدیل، دقت و افزایش دقت و سرعت بخشیدن به تبدیل سیگنالهای ورودی آنالوگ مورد استفاده قرار می گیرد.

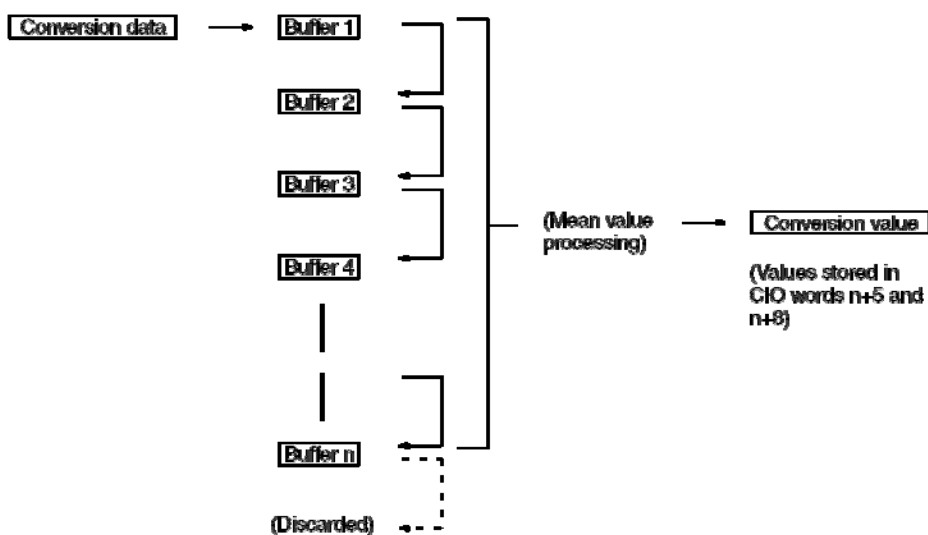
این تنظیمات به صورت همزمان برای هر چهار ورودی آنالوگ انجام می پذیرد به همین دلیل نمی توان تنظیمات زمان تبدیل و دقت را برای هر ورودی آنالوگ به صورت جداگانه انجام داد.



نکته: بعد از انجام تنظیمات ناحیه DM جهت انتقال این تنظیمات به کارت آنالوگ باید PLC را خاموش و روشن نمود و یا اینکه بیت مربوط به راه اندازی مجدد کارتهای ویژه را با در نظر گرفتن Unit Number روشن و خاموش نمود.

2-9-3- پردازش مقدار میانگین

کارت های ورودی آنالوگ توانایی محاسبه مقدار میانگین داده های تبدیل شده نمونه برداری شده را دارا می باشند. این عملیات با استفاده از میانگین گیری داده هایی که در میانگیرها (Buffer) ذخیره شده اند انجام می گیرد در نتیجه تأثیری بر زمان دوره PLC نخواهد داشت. (تعداد میانگیرهایی که می توان برای پردازش مقدار میانگین مورد استفاده قرار داد 2، 4، 8، 16، 32 و یا 64 میانگین خواهد بود).



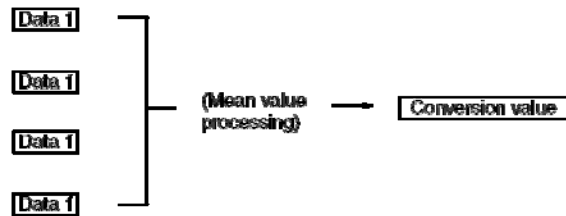
زمانی که تعداد "n" میانگیر مورد استفاده قرار می گیرد، اولین داده تبدیل شده در همه میانگیرها قرار خواهد گرفت.

زمانی که این تابع با تابع "نگه داشتن مقدار حداکثر" به صورت همزمان مورد استفاده قرار می گیرد، این تابع عمل نخواهد کرد و میانگین گیری انجام نخواهد شد.

برای انجام تنظیمات مربوط به پردازش مقدار میانگین، استفاده کردن و یا نکردن از این تابع و همچنین تعیین تعداد میانگیرها برای میانگین گیری، باید تنظیمات مربوطه را بر طبق جدول زیر در حافظه های $D(m+6)$ تا $D(m+9)$ انجام داد.

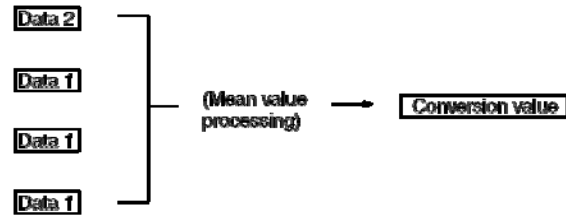
DM word	Function	Set value
$D(m+6)$	Input 1 mean value processing	0000: Mean value processing with 2 buffers 0001: No mean value processing
$D(m+7)$	Input 2 mean value processing	0002: Mean value processing with 4 buffers 0003: Mean value processing with 8 buffers
$D(m+8)$	Input 3 mean value processing	0004: Mean value processing with 16 buffers 0005: Mean value processing with 32 buffers
$D(m+9)$	Input 4 mean value processing	0006: Mean value processing with 64 buffers

عملیات محاسبه مقدار میانگین به صورت زیر انجام می پذیرد. (در این مثال با استفاده از 4 میانگیر) همراه با اولین نمونه برداری داده 1 در همه میانگیرها قرار خواهد گرفت.



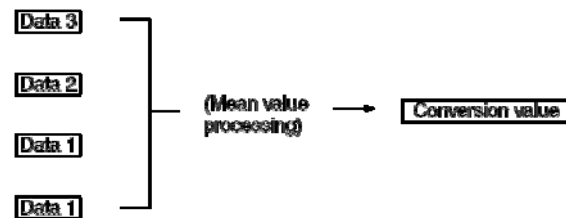
$$\text{مقدار میانگین} = \frac{4}{1} = \text{داده 1} + \text{داده 1} + \text{داده 1} + \text{داده 1}$$

2. با دومین نمونه برداری این مقدار داخل اولین میانگیر قرار می گیرد.



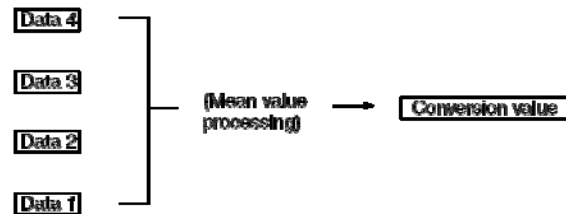
$$\text{مقدار میانگین} = \frac{4}{1} = \text{داده 2} + \text{داده 1} + \text{داده 1} + \text{داده 1}$$

3. با سومین نمونه برداری این مقدار داخل اولین میانگیر قرار می گیرد.



$$\text{مقدار میانگین} = \frac{4}{1} = \text{داده 3} + \text{داده 2} + \text{داده 1} + \text{داده 1}$$

4. با چهارمین نمونه برداری این مقدار داخل اولین میانگیر قرار می گیرد.

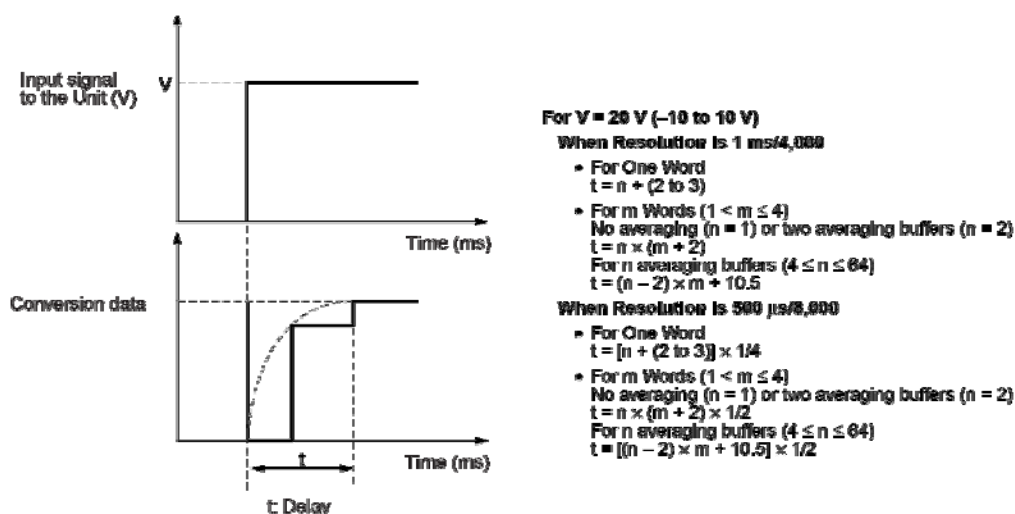


$$\text{مقدار میانگین} = \frac{4}{1} = \text{داده 4} + \text{داده 3} + \text{داده 2} + \text{داده 1}$$

با شروع مجدد کارت آنالوگ عملیات میانگین گیری از ابتدا آغاز می گردد.

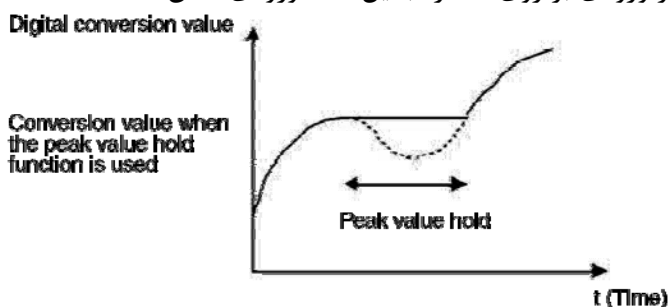
نکته:

1. زمانی که تابع پردازش مقدار میانگین مورد استفاده قرار می گیرد، تأخیری که در تجدید حالت مقدار تبدیل شده ورودی در هنگام تغییر سیگنال ورودی آنالوگ ایجاد می گردد، در نمودار زیر آمده است.
2. در صورت عدم استفاده از تابع میانگین گیری تغییرات سیگنال ورودی خیلی سریعتر در مقدار تبدیل شده ورودی نمایان می گردد.

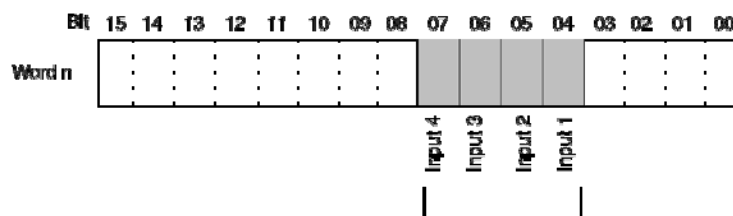


2-9-4- تابع نگه داشتن مقدار حداکثر

تابع نگه داشتن مقدار حداکثر (Peak Value Hold) حداکثر مقدار تبدیل شده باینری هر ورودی را نگاه خواهد داشت. این تابع با ورودی های آنالوگ مورد استفاده قرار می گیرد. در نمودار زیر تأثیر استفاده از تابع نگه داشتن مقدار حداکثر ورودی بر روی مقدار تبدیل شده ورودی نشان داده شده است.

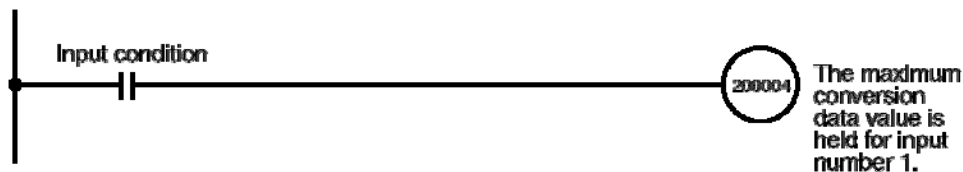


تنظیمات مربوط به نگه داشتن مقدار حداکثر به صورت جداگانه برای هر ورودی در بیت های 04 تا 07 حافظه CIO(n) انجام می گیرد.



تابع مربوط به نگه داشتن مقدار ماکزیمم برای هر یک از ورودی های آنالوگ با روشن کردن بیت متناظر با آن فعال خواهد شد.

در مثال زیر فعال کردن تابع نگه داشتن مقدار حداکثر برای ورودی اول آنالوگ زمانی که Unit Number مساوی 0 می باشد، نشان داده شده است.



زمانی که دو تابع "پردازش مقدار میانگین" و "نگه داشتن مقدار حداکثر" همزمان مورد استفاده قرار می گیرد، تابع پردازش مقدار میانگین عمل نخواهد کرد.

2-9-5- توابع مقیاس گذاری ورودی

زمانی که برای سیگنال آنالوگ حد بالا و پایین (در محدودهٔ دسیمال بین 32000- تا 32000) به صورت دادهٔ باینری 16 بیتی (بین 8300 تا 7D00) در حافظهٔ DM در واحد CPU تعیین می گردد، داده های ورودی آنالوگ به صورت خودکار در محدودهٔ تعیین شده توسط کاربر، با در نظر گرفتن حد بالا و پایین به عنوان مقیاس کامل و با توجه به دقت تبدیل می گردند. این قابلیت نیاز برنامه نویسی برای تبدیل سیگنال در مقیاس های دیگر را از بین برده است. اگرچه این قابلیت فقط در مورد دقت 4000 و زمان تبدیل 1ms صادق می باشد.

نکته:

1. برای تنظیم حد بالا و یا حد پایین به صورت یک عدد منفی باید از مکمل دوی عدد استفاده گردد. (8300 تا FFF به جای 32000 تا -1).
2. منظور از m در ناحیهٔ DM از حافظه $m=20000+\text{unit number} \times 100$ می باشد.
3. همواره حد بالا < حد پایین و یا حد پایین > حد بالا
4. زمانی که حد بالا و پایین در ناحیهٔ DM تعیین می گردد باید از انجام تنظیمات به صورت دادهٔ 16 بیتی باینری اطمینان حاصل نمود (در موارد منفی بودن عدد استفاده از مکمل دو). برای مقادیر مابین 32000- تا 32000 شانزده بیت باینری با محتوای 8300 تا 7D00 .
5. این قابلیت فقط با دقت 4000 و زمان تبدیل 1ms قابل اجرا می باشد. (در مورد دقت 8000 و زمان تبدیل $500\mu\text{s}$ قابل اجرا نمی باشد)
6. زمانی که مقیاس حد بالا و پایین برابر گردد و یا خارج از محدودهٔ 32000- تا 32000 گردد، خطایی ایجاد شده و دیگر مقیاس گذاری انجام نخواهد شد. در صورتی که حد بالا و پایین 0000 تعیین شود (مقدار از پیش تعریف شده) عملکرد به صورت معمول انجام خواهد شد.

تعیین حد بالا و پایین جهت مقیاس گذاری ورودی ها

حد بالا و پایین را برای ورودی های شماره 1 الی 4 در آدرس های $m+27$ تا $m+34$ از ناحیه DM باید تعیین نمود.

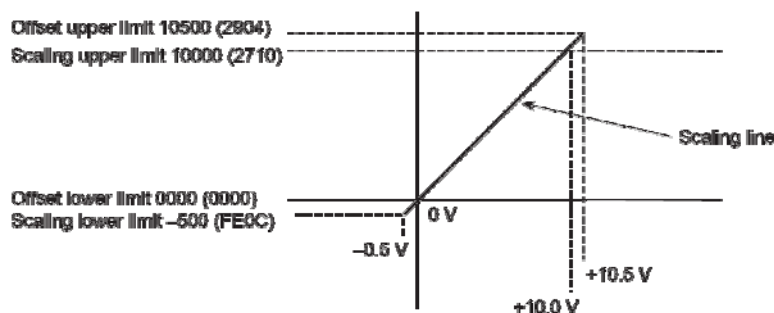
DM word	Bits															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
D(m+27)	Input 1 scaling lower limit															
D(m+28)	Input 1 scaling upper limit															
D(m+29)	Input 2 scaling lower limit															
D(m+30)	Input 2 scaling upper limit															
D(m+31)	Input 3 scaling lower limit															
D(m+32)	Input 3 scaling upper limit															
D(m+33)	Input 4 scaling lower limit															
D(m+34)	Input 4 scaling upper limit															

مثال 1:

ابتدا باید شرایط زیر را در $D(m+27)$ تا $D(m+34)$ تعیین نمود. (داده های داخل پرانتز داده های باینری می باشند)

Setting condition	Set value
Input signal range	0 to 10 V
Scaling lower limit	0000 (0000)
Scaling upper limit	10,000 (2710)

زمانی که محدوده سیگنال ورودی 0 تا 10V باشد:



جدول زیر داده های متناظر بین سیگنال آنالوگ ورودی و مقدار مقیاس شده را نشان می دهد.

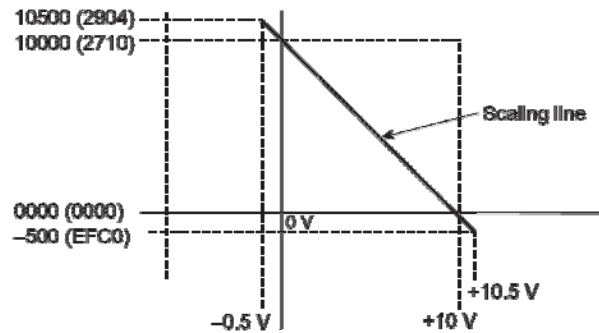
Input signal	Conversion result
0 V	0000 (0000)
10 V	10,000 (2710)
-0.5 V	-500 (FE0C)
10.5 V	10,500 (2904)

مثال 2:

ابتدا باید شرایط زیر را در $D(m+27)$ تا $D(m+34)$ تعیین نمود. (داده های داخل پرانتز داده های باینری می باشند)

Setting condition	Set value
Input signal range	0 to 10 V
Scaling lower limit	10000 (2710)
Scaling upper limit	0000 (0000)

زمانی که محدوده سیگنال ورودی 0 تا 10V باشد: (مقیاس گذاری معکوس)



جدول زیر داده های متناظر بین سیگنال آنالوگ ورودی و مقدار مقیاس شده را نشان می دهد. (داده های داخل پرانتز به صورت باینری می باشند)

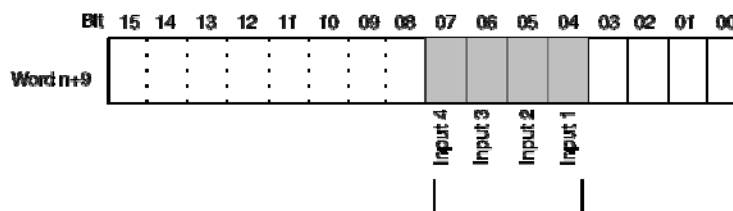
Input signal	Conversion result
0 V	10,000 (2710)
10 V	0000 (0000)
-0.5 V	10,500 (2904)
10.5 V	-500 (FE0C)

2-9-6- تابع مشخصه قطعی ورودی

زمانی که از سیگنال محدوده 1 تا 5V و یا 4 تا 20mA استفاده می شود قطعی احتمالی مدارات ورودی با استفاده از فلگ مشخصه قطعی ورودی نشان داده می شود. شرایط لازم برای فعال شدن این مشخصه برای هر کدام از محدوده های سیگنال ورودی در جدول زیر آمده است.

Range	Current/voltage
1 to 5 V	0.3 V max.
4 to 20 mA	1.2 mA max.

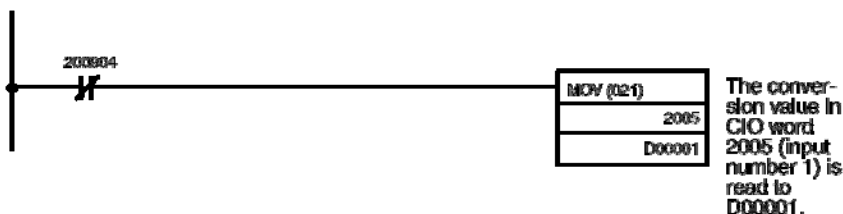
سطح ولتاژ و جریان بسته به تنظیمات بهره و آفست تغییر خواهد نمود. سیگنال نشان دهنده قطعی ورودی برای هر یک از ورودی ها در بیت 04 تا 07 حافظه CIO(n+9) قرار گرفته است. با قرار دادن این بیت ها به عنوان شرط عملکرد دستورالعملهای مربوط به ورودی های آنالوگ در برنامه نویسی می توان این سیگنال را در برنامه لحاظ نمود.



The respective bit turns ON when a disconnection is detected for a given input. When the disconnection is restored, the bit turns OFF.

زمانی که قطعی روی می دهد مقدار تبدیل شده برابر با 0000 خواهد بود.

در مثال زیر مقدار تبدیل شده ورودی آنالوگ شماره 1 فقط در صورتی که قطعی روی نداده باشد خوانده خواهد شد.



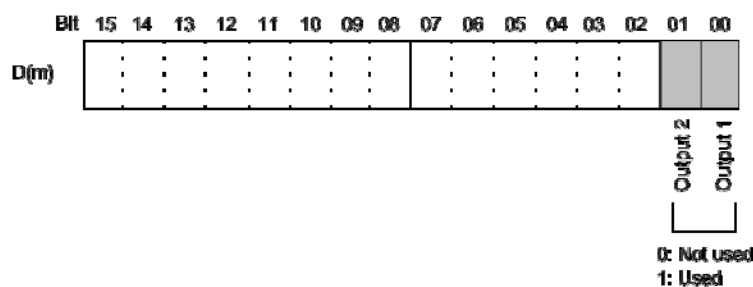
2-10-10- توابع خروجی های آنالوگ و شیوه عملکرد آنها

2-10-10-1- تنظیمات خروجی و تبدیل مقادیر

تنظیمات خروجی ها و محدوده آنها

تعداد خروجی ها:

کارتهای ورودی/خروجی آنالوگ، خروجی های آنالوگ مشخص شده با شماره 1 و 2 را تبدیل می نماید. برای مشخص نمودن خروجی هایی که مورد استفاده قرار می گیرد، باید همچنانکه در شکل مشاهده می گردد بیتهای مربوط به آن خروجی را در تنظیمات آدرس D(m) مساوی 1 قرار داد.



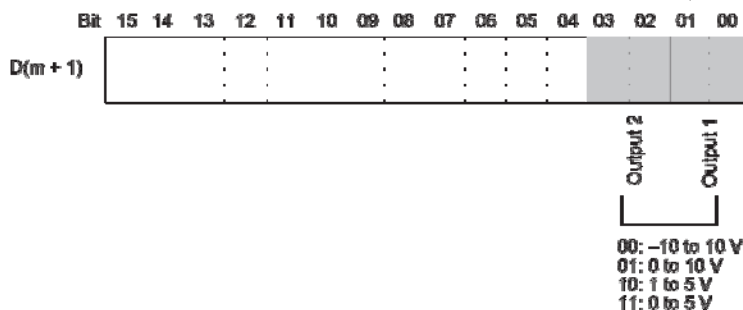
زمان تبدیل خروجی های آنالوگ با قرار دادن 0 در بیتهای متناظر با خروجی های استفاده نشده کاهش می یابد.

زمان تبدیل = 1ms * تعداد خروجی های آنالوگ استفاده شده.

نکته: در حالتی که دقت 8000 تعیین گردد زمان تبدیل 500µs خواهد بود، در نتیجه در رابطه بالا 1ms به 500µs تبدیل خواهد شد.

محدوده سیگنالهای خروجی:

برای هر یک از خروجی های آنالوگ می توان یکی از چهار محدوده سیگنال مربوط به خروجی های آنالوگ (-10 تا 10 ولت، 0 تا 10 ولت، 0 تا 5 ولت، 1 تا 5 ولت و 4 تا 20 میلی آمپر) را تعیین نمود. برای تعیین محدوده سیگنال برای هر یک از خروجی ها باید تنظیمات مربوط به محدوده سیگنال را با توجه به شکل زیر در حافظه $D(m+1)$ انجام داد.



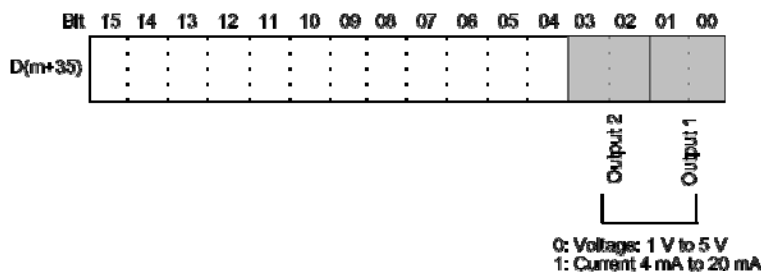
نکته:

- برای آدرسهای حافظه DM مقدار m از رابطه $m=20000+\text{unit number} \times 100$ به دست می آید.

- بعد از انجام تنظیمات ناحیه DM جهت انتقال این تنظیمات به کارت آنالوگ باید PLC را خاموش و روشن نمود و یا اینکه بیت مربوط به راه اندازی مجدد کارتهای ویژه را با در نظر گرفتن Unit Number روشن و خاموش نمود.

تنظیمات محدوده جریان و ولتاژ:

زمانی که گزینه "1 تا 4/5V تا 20mA" به عنوان محدوده سیگنال خروجی استفاده می گردد، ولتاژ (1 تا 5V) و یا جریان (4 تا 20mA) بودن این سیگنال خروجی را می توان با استفاده از تنظیمات $D(m+35)$ تعیین نمود.



نوشتن مقدار خروجی:

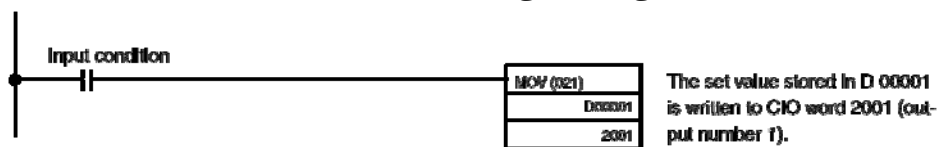
مقدارهای باینری سیگنالهای آنالوگ خروجی باید در حافظه های $CIO(n+1)$ و $CIO(n+2)$ نوشته شود.

Word	Function	Stored value
$n+1$	Output 1 set value	16-bit binary data
$n+2$	Output 2 set value	

برای حافظه های CIO مقدار n با استفاده از رابطه " $n=2000+\text{unit number} \times 10$ " به دست می آید. برای نوشتن مقادیر تنظیم شده برای سیگنالهای خروجی می توان از دستور عملهای MOV(021) و XFER(070) در برنامه استفاده نمود.

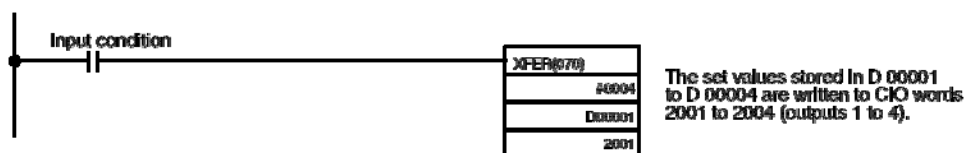
مثال 1:

در مثال زیر مقدار تعیین شده برای یکی از خروجی ها نوشته شده است.



مثال 2:

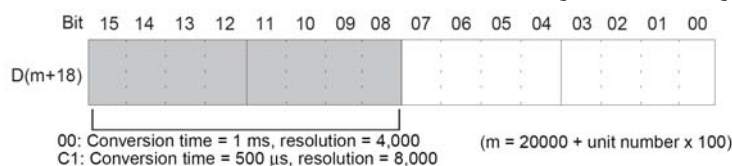
در این مثال مقدار تعیین شده برای چندین خروجی به صورت همزمان نوشته شده است.



اگر مقدار نوشته شده برای یک خروجی خارج از محدوده تعیین شده برای خروجی باشد، خطایی در تنظیمات خروجی روی خواهد داد.

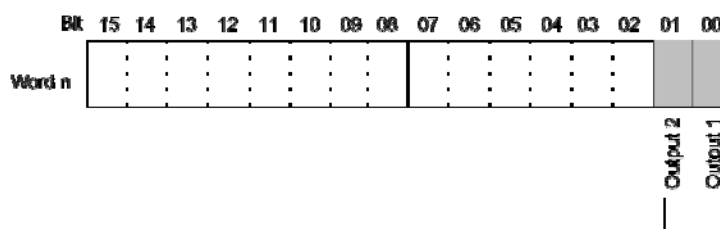
2-10-2- تنظیمات زمان تبدیل و دقت

تنظیمات مربوط به زمان تبدیل و دقت تبدیل داده ها برای ورودی های آنالوگ و همچنین خروجی ها به صورت همزمان انجام می گیرد. چگونگی این تنظیمات را می توان در بخش مربوط به تنظیمات زمان تبدیل و دقت ورودی های آنالوگ مشاهده نمود.



فعال نمودن و متوقف نمودن تبدیل:

برای فعال نمودن تبدیل داده های باینری به سیگنال خروجی آنالوگ باید بیت های مربوط به فعال نمودن خروجی های آنالوگ را متناظر با شماره خروجی آنالوگ روشن نمود. بیت های شماره 00 و 01 حافظه CIO(n) به این امر اختصاص داده شده است.



Analog conversion is executed while these bits are ON. When the bits are turned OFF, the conversion is stopped and the output data is held.

برای حافظه های CIO مقدار n با استفاده از رابطه "n=2000+unit number×10" به دست می آید.

در هنگام متوقف کردن تبدیل خروجی های آنالوگ، خروجی ها با توجه به محدوده آنها و همچنین تنظیمات مربوط به "حالت خروجی ها در زمان توقف تبدیل" که در ادامه به آن پرداخته می شود، عمل خواهند نمود. در شرایط زیر تبدیل داده های باینری به سیگنال آنالوگ خروجی حتی در زمانی که بیت مربوط به فعال نمودن خروجی ها نیز روشن می باشد، انجام نخواهد شد:

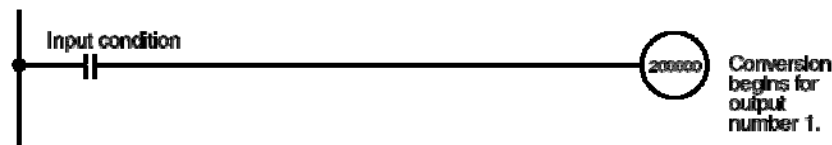
1. در حالت Adjustment Mode زمانی که چیزی غیر از خروجی با شماره معین در حال تنظیم شدن باشد.

2. زمانی که خطایی در تنظیمات خروجی روی می دهد.

3. زمانی که خطایی در عملکرد CPU روی می دهد.

4.

زمانی که مد عملکرد CPU از مد Run و یا Monitor به Program تغییر وضعیت دهد یا زمانی که تغذیه PLC وصل می گردد، تمام بیت های مربوط به فعال نمودن خروجی ها خاموش خواهند شد در نتیجه عملکرد خروجی آنالوگ با توجه به تنظیمات حالت خروجی ها در زمان توقف تبدیل انجام خواهد شد. در این مثال فعال نمودن تبدیل خروجی آنالوگ شماره 1 نشان داده شده است. (Unit Number=0).



2-10-3- تابع تعیین حالت خروجی در صورت توقف تبدیل

کارت ورودی/خروجی آنالوگ در صورت رویداد یکی از موارد زیر تبدیل را متوقف می نماید، مقدار خروجی در حالت توقف تبدیل با توجه به تنظیمات انجام شده در این بخش تعیین می گردد.

1. زمانی که بیت مربوط به فعال نمودن تبدیل خروجی خاموش باشد.

2. در حالت Adjustment Mode زمانی که چیزی غیر از خروجی با شماره معین در حال تنظیم شدن باشد.

3. زمانی که خطایی در تنظیمات خروجی روی می دهد.

4. زمانی که خطایی در عملکرد CPU روی می دهد.

5. زمانی که خطایی در باس واسط کارتهای ورودی/خروجی روی دهد.

6. زمانی که CPU در حالت LOAD OFF باشد.

7. زمانی که خطای WDT (WatchDog Timer) در CPU روی دهد.

سه حالت HOLD، CLR و یا MAX را می توان در صورت توقف تبدیل خروجی ها برای حالت خروجی انتخاب نمود.

MAX	HOLD	CLR	محدوده سیگنال خروجی
10.5 ولت (حداکثر 0.5% از مقیاس کامل)	ولتاژ خروجی درست قبل از توقف تبدیل	0.5- ولت (حداقل 0.5% از مقیاس کامل)	0 تا 10 ولت
11.0 ولت (حداکثر 0.5% از مقیاس کامل)	ولتاژ خروجی درست قبل از توقف تبدیل	0 ولت (حداقل 0.5% از مقیاس کامل)	10- تا 10 ولت
5.2 ولت (حداکثر 0.5% از مقیاس کامل)	ولتاژ خروجی درست قبل از توقف تبدیل	0.8 ولت (حداقل 0.5% از مقیاس کامل)	1 تا 5 ولت
5.2 ولت (حداکثر 0.5% از مقیاس کامل)	ولتاژ خروجی درست قبل از توقف تبدیل	0.25- ولت (حداقل 0.5% از مقیاس کامل)	0 تا 5 ولت
20.8 میلی امپر (حداکثر 0.5% از مقیاس کامل)	جریان خروجی درست قبل از توقف تبدیل	3.2 میلی امپر (حداقل 0.5% از مقیاس کامل)	4 تا 20 میلی امپر

مقادیر ذکر شده در جدول بالا در صورت تنظیم بهره و آفست ممکن است تغییر نماید. برای تعیین حالت خروجی در صورت توقف تبدیل باید تنظیمات در حافظه های $D(m+2)$ تا $D(m+5)$ بر طبق جدول زیر انجام شود.

DM word	Function	Set value
$D(m+2)$	Output 1: Output status when stopped	xx00: CLR Output 0 or minimum value of range (-5%). xx01: HOLD Hold output value prior to stop. xx02: MAX Output maximum value of range (105%). Set any value in the leftmost bytes (xx).
$D(m+3)$	Output 2: Output status when stopped	

- برای آدرسهای حافظه DM مقدار m از رابطه $m = 20000 + \text{unit number} \times 100$ به دست می آید.
- بعد از انجام تنظیمات ناحیه DM جهت انتقال این تنظیمات به کارت آنالوگ باید PLC را خاموش و روشن نمود و یا اینکه بیت مربوط به راه اندازی مجدد کارتهای ویژه را با در نظر گرفتن Unit Number روشن و خاموش نمود.

2-10-4- توابع مقیاس گذاری خروجی

زمانی که برای سیگنال آنالوگ حد بالا و پایین (در محدوده دسیمال بین 32000- تا 32000) به صورت داده باینری 16 بیتی (بین 8300 تا 7D00) در حافظه DM در واحد CPU تعیین می گردد، مقادیر تعیین شده برای خروجی آنالوگ به صورت خودکار در محدوده تعیین شده توسط کاربر، با در نظر گرفتن حد بالا و پایین به عنوان مقیاس کامل و با توجه به دقت تعیین شده، تبدیل می گردند. این قابلیت نیاز برنامه نویسی برای

تبدیل سیگنال در مقیاس های دیگر را از بین برده است. اگرچه این قابلیت فقط در مورد دقت 4000 و زمان تبدیل 1ms صادق می باشد.

نکته:

1. برای تنظیم حد بالا و یا حد پایین به صورت یک عدد منفی باید از مکمل دو عدد استفاده گردد.
(8300 تا FFF به جای 32000- تا -1).
2. منظور از m در ناحیه DM از حافظه $m=20000+\text{unit number} \times 100$ می باشد.
3. همواره حد بالا < حد پایین و یا حد پایین > حد بالا
4. تبدیل دیجیتال به آنالوگ در عمل بین 5% و 105% مقیاس کامل می باشد.
5. زمانی که حد بالا و پایین در ناحیه DM تعیین می گردد باید از انجام تنظیمات به صورت داده 16 بیتی باینری اطمینان حاصل نمود (در موارد منفی بودن عدد استفاده از مکمل دو). برای مقادیر مابین 32000- تا 32000 شانزده بیت باینری با محتوای 7D00 تا 8300.
6. این قابلیت فقط با دقت 4000 و زمان تبدیل 1ms قابل اجرا می باشد. (در مورد دقت 8000 و زمان تبدیل $500\mu\text{s}$ قابل اجرا نمی باشد)
7. زمانی که مقیاس حد بالا و پایین برابر گردد و یا خارج از محدوده 32000- تا 32000 گردد، خطایی ایجاد شده و دیگر مقیاس گذاری انجام نخواهد شد. در صورتی که حد بالا و پایین 0000 تعیین شود (مقدار از پیش تعریف شده) عملکرد به صورت معمول انجام خواهد شد.

تعیین حد بالا و پایین جهت مقیاس گذاری خروجی ها

حد بالا و پایین را برای خروجی های شماره 1 الی 2 در آدرس های $D(m+19)$ تا $D(m+22)$ از ناحیه DM باید تعیین نمود.

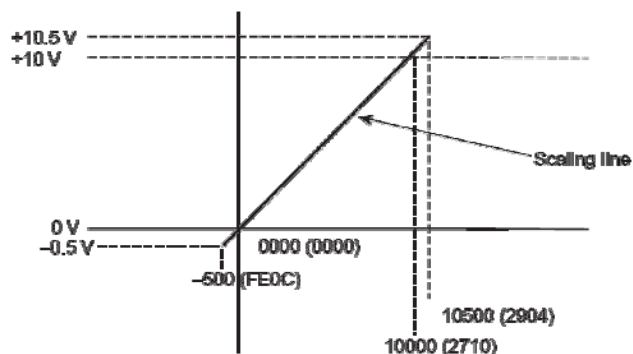
DM word	Bits															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
$D(m+19)$	Output 1 scaling lower limit															
$D(m+20)$	Output 1 scaling upper limit															
$D(m+21)$	Output 2 scaling lower limit															
$D(m+22)$	Output 2 scaling upper limit															

مثال 1:

در ابتدا باید شرایط زیر را در $D(m+19)$ تا $D(m+22)$ تعیین نمود. (داده های داخل پرانتز داده های باینری می باشند)

Setting condition	Set value
Output signal range	0 to 10 V
Scaling lower limit	0000 (0000)
Scaling upper limit	10,000 (2710)

زمانی که محدوده سیگنال خروجی 0 تا 10V باشد:



جدول زیر داده های متناظر بین سیگنال آنالوگ خروجی و مقدار مقیاس شده را نشان می دهد.

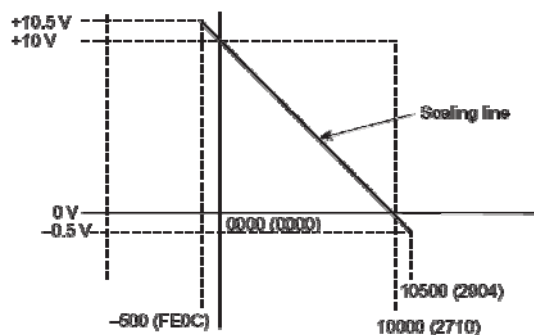
Output set value	Output signal
0000 (0000)	0 V
10,000 (2710)	10 V
-500 (FE0C)	-0.5 V
10,500 (2904)	10.5 V

مثال 2:

ابتدا باید شرایط زیر را در $D(m+19)$ تا $D(m+22)$ تعیین نمود. (داده های داخل پرانتز داده های باینری می باشند)

Setting condition	Set value
Output signal range	0 to 10 V
Scaling lower limit	10000 (2710)
Scaling upper limit	0000 (0000)

زمانی که محدوده سیگنال خروجی 0 تا 10V باشد:

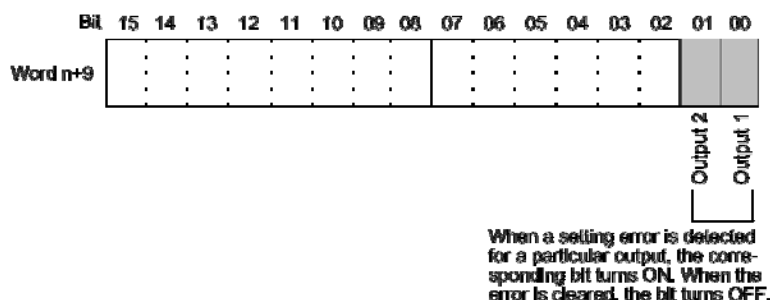


جدول زیر داده های متناظر بین سیگنال آنالوگ خروجی و مقدار مقیاس شده را نشان می دهد. (داده های داخل پرانتز به صورت باینری می باشند)

Conversion result	Output signal
10,000 (2710)	0 V
0000 (0000)	10 V
10,500 (2904)	-0.5 V
-500 (FE0C)	10.5 V

2-10-5- خطاهای تنظیمات خروجی

اگر مقدار تعیین شده برای خروجی از محدوده تعریف شده برای سیگنال خروجی بیشتر باشد، سیگنال مشخصه این خطا برای هر یک از خروجی ها در بیت های 00 و 01 حافظه CIO(n+9) قرار می گیرد.



نکته:

در صورت که خطایی در تنظیمات خروجی روی دهد حالت خروجی با توجه به تنظیمات حالت خروجی در صورت توقف تبدیل پیروی خواهد کرد.

2-11- تنظیم آفست و بهره

با استفاده از این تابع می توان ورودی و خروجی را با توجه به تجهیز متصل به آنها کالیبره نمود.

کالیبره کردن ورودی

زمانی که دقت 4000 انتخاب می شود، به وسیله این تابع می توان آفست ولتاژ (جریان) و بهره ولتاژ (جریان) موجود در تجهیز نصب شده به عنوان ورودی آنالوگ را تنظیم نموده و این سیگنال را در محدوده 0000 تا 0FA0 (07D0) که محدوده سیگنال 10- تا 10V انتخاب شود) تبدیل می نماید. به عنوان مثال زمانی که از یک تجهیز آنالوگ با محدوده سیگنال 1 تا 5V استفاده می شود، ممکن است در عمل سیگنال خروجی تجهیز آنالوگ 0.8 الی 4.8V باشد اگرچه محدوده سیگنال برای آن 1 تا 5V تعیین شده باشد. در چنین مواردی وقتی سیگنال آنالوگ ورودی دارای آفستی برابر با 0.8V باشد مقدار تبدیل شده سیگنال آنالوگ

ورودی در دقت 4000 برابر با FF38 خواهد شد و در زمانی که مقدار ولتاژ ورودی 4.8V باشد داده تبدیل شده سیگنال ورودی 0EDA خواهد شد. با استفاده از تابع تنظیم افسست و بهره زمانی که ولتاژ ورودی بین 0.8 تا 4.8V به عنوان ورودی آنالوگ اعمال گردد این داده ها به داده های باینری بین 0000 تا 0FA0 تبدیل خواهد شد. (به جای تبدیل به داده ای بین FF38 تا 0EDA)

Output device offset and gain voltage	Converted data before adjustment	Converted data after adjustment
0.8 V	FF38 (FE70)	0000 (0000)
4.8 V	0EDA (0DB4)	0FA0 (1F40)

(Resolution: 8,000)

کالیبره کردن خروجی

با استفاده از این قابلیت می توان ولتاژهای خروجی آنالوگ را بر طبق افسست و بهره تجهیز نصب شده به خروجی آنالوگ تنظیم نمود. به عنوان مثال فرض کنید که مقدار تعیین شده برای یک تجهیز که به خروجی آنالوگ نصب شده است (مانند یک نمایشگر که سیگنال آنالوگ خروجی را با مقیاس مشخص نمایش می دهد) بین 100.0 و 500.0 باشد.

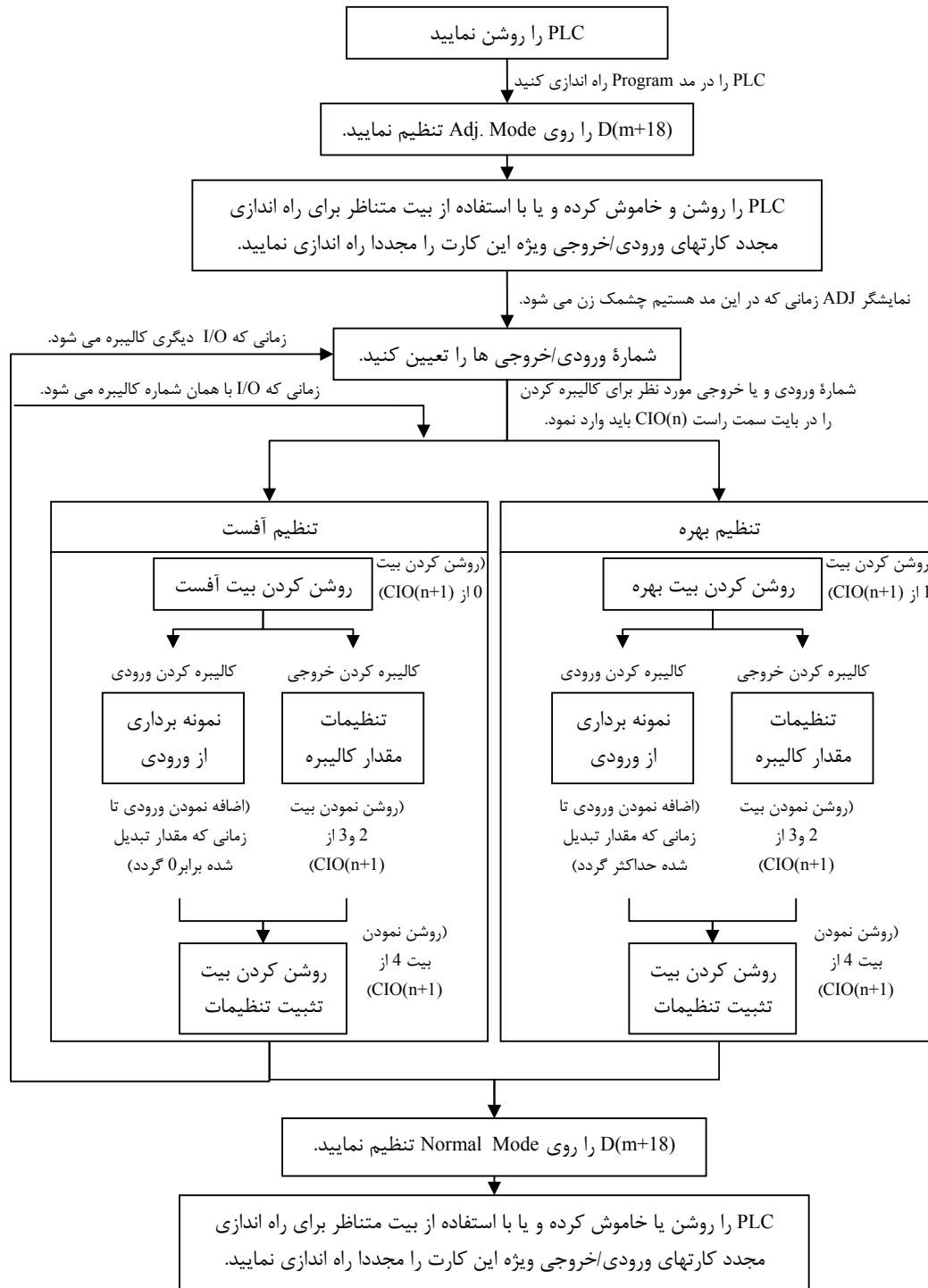
اگر مقدار تعیین شده برای ولتاژ خروجی مقدار 0000Hex باشد و نمایشگر متصل شده به خروجی آنالوگ به جای مقدار 100.0 به ازاء 0000Hex مقدار 100.5 را نمایش دهد ولتاژ خروجی را باید طوری تنظیم نمود (در این مورد کمتر نمود) که نمایشگر عدد 100.0 را نمایش دهد. در نتیجه برای نشان دادن مقدار 100.0 روی نمایشگر باید مقدار دیگری غیر از 0000 را در حافظه خروجی قرار داد (به عنوان مثال در این مورد 0FFBHex).

همچنین در مورد مقدار بهره خروجی، اگر مقدار تعیین شده برای خروجی مقدار 0FA0 Hex باشد و نمایشگر متصل شده به خروجی آنالوگ به جای مقدار 500.0 به ازاء 0FA0Hex مقدار 500.5 را نمایش دهد ولتاژ خروجی را باید طوری تنظیم نمود (در این مورد باید کمتر نمود) که نمایشگر عدد 500.0 را نمایش دهد. در نتیجه برای نشان دادن مقدار 500.0 روی نمایشگر باید مقدار دیگری غیر از 0FA0 را در حافظه خروجی قرار داد (به عنوان مثال در این مورد 0F9BHex).

Display at external input device	Set value before adjustment (word n+8)	Set value after adjustment
100.0	0FFB (FFFD)	0000 (0000)
500.0	0F9B (1F36)	0FA0 (1F40)

(Resolution: 8,000)

2-11-1- چگونگی عملکرد در Adjustment Mode



توجه: زمانی که کارت آنالوگ در حال Adj. Mode می باشد باید PLC را در حالت Program Mode قرار داد. اگر PLC در یکی از مدهای Run و یا Monitor باشد عملکرد کارت آنالوگ متوقف خواهد شد و ورودی/خروجی های آنالوگ به حالت قبل از این توقف برگردانده خواهند شد.

همچنین تنظیمات بهره و آفست را به صورت پیوسته انجام داد.

نکته: کالیبره نمودن ورودی زمانی که از تابع پردازش مقدار میانگین استفاده می شود بسیار دقیق تر انجام خواهد شد.

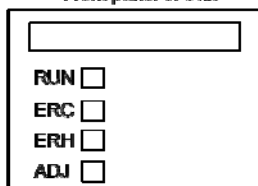
2-12- بررسی خطاها و هشدارها

2-12-1- نمایشگرها و نمودار خطا

نمایشگرها:

اگر خطا و یا هشدار در کارت آنالوگ روی دهد، نمایشگر ERC و یا ERH موجود روی کارت آنالوگ روشن خواهد شد.

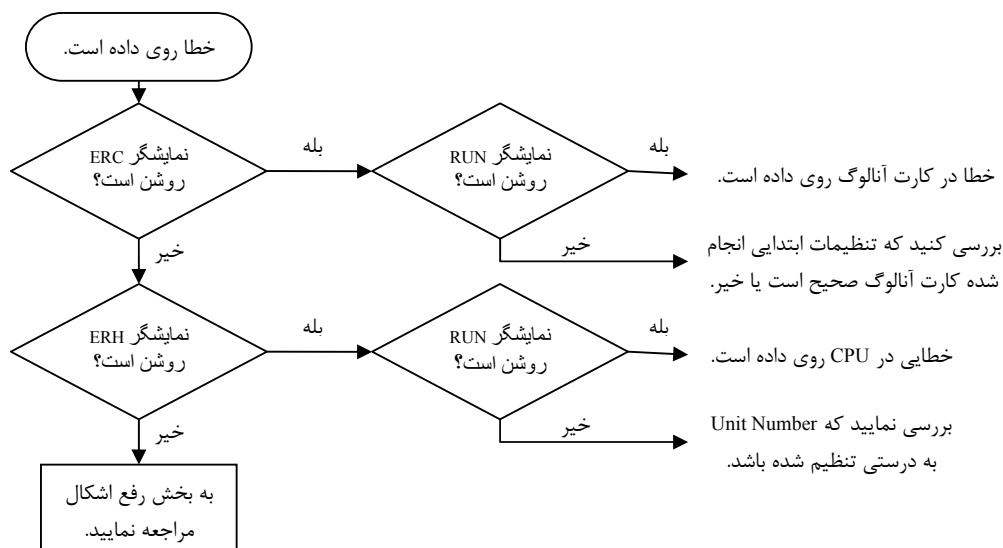
Front panel of Unit



LED	مفهوم	نمایشگر	حالت عملکرد
RUN (سبز رنگ)	عملکرد نرمال	روشن	عملکرد در مد نرمال.
		خاموش	تبادل اطلاعات بین کارت آنالوگ و CPU متوقف شده است.
ERC (قرمز رنگ)	خطای نمایان شده توسط کارت آنالوگ	روشن	خطایی (همچون مشخصه قطعی) روی داده و یا تنظیمات آغازین نادرست بوده است.
		خاموش	عملکرد به صورت نرمال.
ADJ (زرد رنگ)	کالیبره کردن	روشن	عملکرد در حالت تنظیم آفست و بهره در مد Adjustment.
		خاموش	عملکرد در حالتی غیر از مد Asjurement.
ERH (قرمز رنگ)	خطای موجود در CPU	روشن	در تبادل اطلاعات بین کارت آنالوگ و CPU خطایی روی داده است.
		خاموش	عملکرد به صورت نرمال.

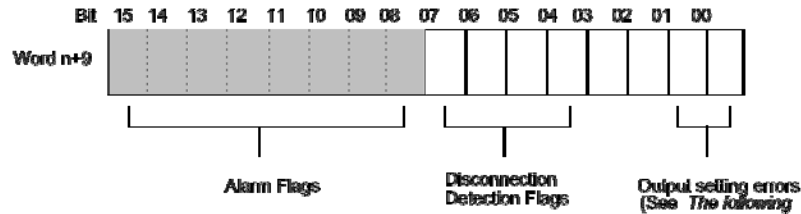
رویه رفع اشکال

با استفاده از رویه نشان داده شده در فلوجارت زیر می توان اشکال های موجود در کارت آنالوگ را رفع نمود.

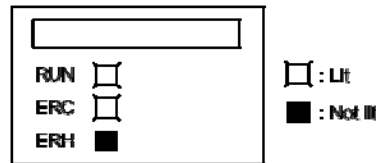


2-12-2- خطاهای کارت ورودی/خروجی آنالوگ

زمانی که خطا و یا هشدار در کارتهای ورودی/خروجی آنالوگ روی می دهد، نمایشگر ERC موجود روی کارت روشن شده و فلگ آلام در بیتهای 08 تا 15 حافظه CIO(n+9) ذخیره می گردد.



نمایشگرهای ERC و RUN هر دو روشن



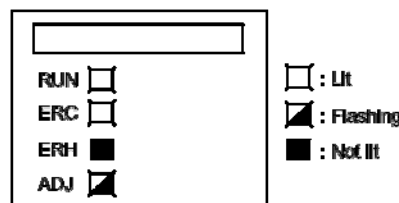
زمانی که خطایی روی داده است ولی کارت آنالوگ به عملکرد نرمال خود ادامه می دهد، هر دو نمایشگر ERC و RUN همزمان روشن خواهند شد. و فلگ خطای مربوطه در حافظه CIO(n+9) روشن خواهد شد. زمانی که خطا از بین می رود این فلگ خطا به صورت خودکار خاموش خواهد شد.

CIO(n+9)	فلگ خطا	تعریف خطا	حالت ورودی/خروجی	چگونگی رفع آن
بیتهای 00 و 01	خطا در مقدار تعیین شده برای خروجی	رنج مقدار تعیین شده برای خروجی از حد معین تجاوز نموده است.	حالت خروجی با توجه به تنظیمات تعیین حالت خروجی در هنگام توقف تبدیل مشخص می گردد.	مقدار تعیین شده را اصلاح نمایید.
بیتهای 04 تا 07	مشخص کننده قطعی	قطعی در ورودی ها روی داده است.	مقدار تبدیل شده ورودی 0000 خواهد بود.	ورودی های قطع شده را با توجه به شماره آن وصل نمایید.
بیت 14	(در مد Adjustment) خطا در نوشتن داده ها بر روی EEPROM	در زمان کار در مد Adj. خطایی در نوشتن داده ها روی EEPROM روی داده است.	مقدار خروجی ها به آخرین مقدار نگه داشته شده قبل از روی دادن خطا برگردانده می شود.	بیت مربوط به تثبیت حالت تنظیمات را خاموش، روشن و مجددا خاموش نمایید. اگر خطا از بین نرود باید کارت دیگری را جایگزین کارت فعلی نمود.

نکته : فلگ مشخص کننده قطعی ورودی فقط برای دو محدوده "1 تا 5V" و "4 تا 20mA" صادق می باشد.

برای حافظه های CIO مقدار n با استفاده از رابطه "n=2000+unit number×10" به دست می آید.

نمایشگرهای ERC و RUN روشن و نمایشگر ADJ چشمک زن

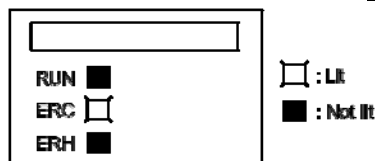


این خطا نتیجه عملکرد نادرست در Adjustment Mode می باشد. زمانی که کارت آنالوگ در این مد مشغول به کار باشد بیت 15 از حافظه CIO(n+9) روشن خواهد شد.

CIO(n+9)	فلگ خطا	تعریف خطا	حالت ورودی/خروجی	چگونگی رفع آن
بیت 12	(در مد Adjustment) خطا در مقدار سیگنال ورودی تنظیمی	در مد Adjustment نمی توان آفست و یا بهره را تنظیم نمود به دلیل اینکه مقدار ورودی از محدوده مجاز تجاوز کرده است.	مقدار تبدیل شده متناظر با سیگنال ورودی در CIO(n+8) نمایش داده خواهد شد.	اگر کالیبره کردن ورودی ها با اتصال تجهیز ورودی انجام می شود بهتر است در ابتدا این تجهیز را کالیبره نمود.
بیت 13	(در مد Adjustment) خطا در تنظیمات شماره ورودی و یا خروجی ها	در مد Adjustment به دلیل اینکه شماره ورودی و یا خروجی مورد نظر تعیین نشده است و یا شماره انتخاب شده صحیح نمی باشد در نتیجه کالیبره شدن انجام نخواهد شد.	مقدار خروجی ها به آخرین مقدار نگه داشته شده قبل از روی دادن خطا برگردانده می شود. هیچ داده ای تغییر نخواهد کرد.	بررسی محتوای CIO(n). محتوای این حافظه برای کالیبره کردن ورودی و خروجی باید عددی بین 11 تا 14 و یا 21 تا 24 باشد. همچنین بررسی کنید که ورودی و خروجی انتخاب شده برای تنظیم در تنظیمات DM به عنوان ورودی و یا خروجی استفاده شده تعیین شده باشد.
بیت 15	(در مد Adjustment) خطای PLC	کارت آنالوگ در مد Adjustment کار می کند در صورتی که PLC در یکی از مدهای Monitor و یا RUN می باشد.	مقدار خروجی ها به آخرین مقدار نگه داشته شده قبل از روی دادن خطا برگردانده می شود.	تنظیمات بیت 00 تا 07 حافظه D(m+18) را تغییر دهید و سپس PLC را خاموش و روشن نمایید و یا بیت مربوط به راه اندازی مجدد کارت آنالوگ را روشن و خاموش نمایید.

نکته: زمانی که خطایی در عملکرد PLC زمانی که کارت آنالوگ در مد Adjustment کار می کند روی دهد، عملکرد کارت متوقف خواهد شد.

نمایشگر ERC روشن و RUN خاموش

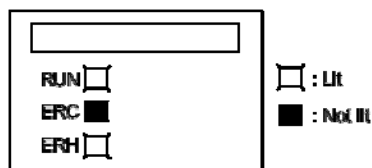


نمایشگر ERC زمانی که تنظیمات ابتدایی کارت آنالوگ به درستی انجام نشده باشد روشن خواهد شد. مشخصه هشدار متناظر با این خطا در حافظه CIO(n+9) روشن خواهد شد. زمانی که خطا برطرف شود این فلگ به صورت خودکار خاموش و یا کارت آنالوگ مجدداً راه اندازی خواهد شد.

2-12-3- خطاهای CPU

زمانی که خطایی در CPU و یا باس کارتهای ورودی/خروجی روی می دهد و تبادل داده ها بین CPU و کارت ویژه ورودی/خروجی به درستی انجام نمی گیرد، عملکرد کارت آنالوگ مختل شده و نمایشگر ERH روشن می شود.

نمایشگر ERH و RUN هر دو روشن



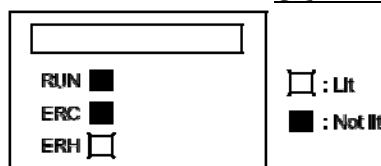
نمایشگرهای RUN و ERH زمانی که خطایی در باس ورودی/خروجی روی می دهد که باعث ایجاد خطای WDT(WatchDog Timer) در CPU می شود، به صورت همزمان روشن می گردند. نتیجه این خطا مختل شدن تبادل داده ها بین CPU و کارت آنالوگ خواهد بود.

در این مورد باید سیستم را خاموش و دوباره راه اندازی نمود.

حالت خروجی	حالت ورودی	تعریف خطا	خطا
بسته به تنظیمات حالت خروجی در زمان توقف تبدیل عمل می نماید.	مقدار تبدیل شده برای سیگنال ورودی در این حالت 0000 می باشد.	خطایی در تبادل داده ها با واحد CPU روی داده است.	خطای باس ورودی/خروجی
آخرین شرایط قبل از روی دادن خطا ثابت نگه داشته خواهد شد.	آخرین شرایط قبل از روی دادن خطا ثابت نگه داشته خواهد شد.	هیچ پاسخی از طرف CPU در فاصله زمانی ثابت مشاهده نمی شود.	خطا در نمایش واحد CPU
بسته به تنظیمات حالت خروجی در زمان توقف تبدیل عمل می نماید.	به حالتی تعریف نشده تغییر می کند.	خطایی در واحد CPU روی داده است.	خطای WDT (WatchDog Error)

نکته: در این حالت هیچ خطایی توسط CPU نمایش داده نمی شود، به این دلیل که CPU به کار خود ادامه خواهد داد.

نمایشگر ERH روشن و نمایشگر RUN خاموش



شماره کارت آنالوگ (Unit Number) به درستی تعیین نشده است.

حالت خروجی	حالت ورودی	تعریف خطا	خطا
مقدار خروجی 0V می باشد.	تبدیل انجام نخواهد شد و داده در این حالت 0000 می باشد.	از یک Unit Number برای دو یا چند کارت استفاده شده است و یا شماره ای غیر از 0 تا 95 به عنوان Unit Number انتخاب شده است.	استفاده از یک Unit Number برای دو کارت
		تناقض بین کارت انتخاب شده در جدول ورودی/خروجی ها و کارتی که واقعا بر روی PLC نصب شده است وجود دارد.	خطا در تنظیمات کارت ورودی/خروجی ویژه

2-12-4- راه اندازی مجدد کارتهای ویژه

برای راه اندازی مجدد کارتهای ویژه بعد از انجام تنظیمات کارت در ناحیه DM و یا برای برطرف کردن خطای روی داده در کارت آنالوگ دو راه وجود دارد. راه اول خاموش و روشن کردن تغذیه PLC می باشد و اما راه دوم روشن و خاموش کردن بیت مربوط به راه اندازی مجدد هر کارت ویژه است که با توجه به شماره کارت مشخص می گردد.

Bits	Functions	
A50200	Unit #0 Restart Bit	Turning the Restart Bit for any Unit ON and then OFF again restarts that Unit.
A50201	Unit #1 Restart Bit	
~	~	
A50215	Unit #15 Restart Bit	
A50300	Unit #16 Restart Bit	
~	~	
A50715	Unit #95 Restart Bit	

اگر خطا حتی با روشن و خاموش کردن کارت آنالوگ نیز از بین نرفت در این صورت باید کارت دیگری را جایگزین نمود.

در طول راه اندازی مجدد کارت آنالوگ داده ورودی 0000 و مقدار خروجی 0V و یا 0mA می باشد.

نکته:

"تنظیمات مربوط به کارتهای CPU Bus و کارتهای ویژه ورودی/خروجی با استفاده از نرم افزار CX-One" همچنان که در بالا دیده می شود در نسخه های قدیمی تر CPU ها و همچنین نرم افزار CX-Programmer تنظیمات اولیه کارتهای CPU Bus و همچنین کارتهای ورودی/خروجی ویژه با رجوع به مدارک و کتابچه های مربوط به هر کارت و محاسبه این تنظیمات به صورت یک عدد هگزادسیمال و قرار دادن در ناحیه DM که آدرس آن با توجه به Unit Number تعیین می گردد، صورت می پذیرفت. ولی هم اکنون این تنظیمات به سادگی و سهولت با استفاده از منوهای مشخصی که در جدول ورودی/خروجی نرم افزار CX-Programmer قرار دارد انجام می گیرد.

سهولت در انجام تنظیمات، در انتقال آنها و همچنین تغییر تنظیمات بدون محاسبه آدرس تعیین شده برای هر کارت از جمله نکات بارزی می باشد که با استفاده از نرم افزار CX-One فراهم می گردد.



دستورالعملها

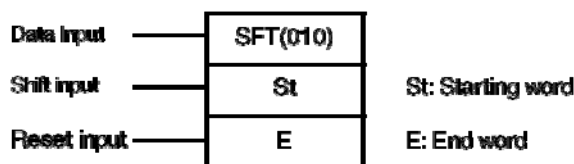
3- دستورالعملهای شیفت داده ها

در این بخش دستورالعملهای شیفت داده ها در داخل Word و یا شیفت بیت های یک Word در حالت های مختلف شرح داده می شود.

کد دستورالعمل	دستور العمل	نام دستور العمل
010	SFT	ثبات تغییر مکان (Shift Register)
084	SFTR	ثبات تغییر مکان برگشت پذیر (Reversible Shift Register)
016	WSFT	تغییر مکان Word (Word Shift)
025	ASL	تغییر مکان حسابی به چپ (Arithmetic Shift Left)
026	ASR	تغییر مکان حسابی به راست (Arithmetic Shift Right)
074	SLD	تغییر مکان یک دیجیت به چپ (One Digit Shift Left)
075	SRD	تغییر مکان یک دیجیت به راست (One Digit Shift Right)

3-1-1- دستور ثبات تغییر مکان (Shift Register-SFT)

این دستور برای شیفت داده ها و تغییر مکان آنها مورد استفاده قرار می گیرد.



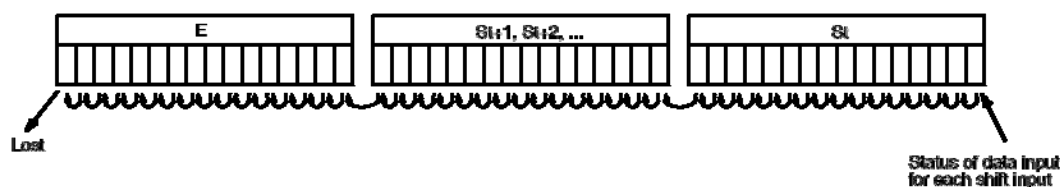
3-1-1- مشخصات عملگرهای دستور (SFT(010)

فضای حافظه	St	E
CIO Area	CIO 0000 to CIO 6143	
Work Area	W000 to W511	
Holding Bit Area	H000 to H511	
Auxiliary Bit Area	A448 to A959	
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	

نکته: دو Word انتخاب شده به عنوان آغاز و پایان محدوده شیفت داده ها باید از یک منطقه از حافظه باشند.

3-1-2- شرح عملکرد دستور SFT(010)

زمانی که شرط عملکرد استفاده شده به عنوان ورودی دستورالعمل SFT(010) (Shift Input) از حالت خاموش به روشن تغییر وضعیت می دهد، تمام داده ها از Word آغازین تا پایانی یک بیت به سمت چپ (از طرف بیت سمت راست به طرف چپ) شیفت پیدا می نماید. و حالت روشن و یا خاموش بودن بیت مربوط به داده ورودی (Data Input) در بیت سمت راست قرار خواهد گرفت.



3-1-3- فلگ های مربوط به دستورالعمل SFT(010)

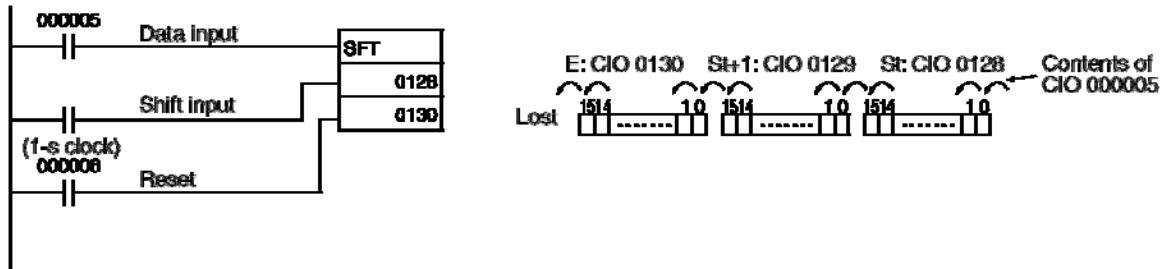
نام	نشانه	عملکرد
Error Flag	ER	زمانی که از آدرس دهی غیر مستقیم ناحیه IR استفاده می شود اگر اشاره گر به حافظه ای غیر از حافظه های مجاز اشاره کند این فلگ روشن می گردد. در بقیه موارد خاموش می باشد.

3-1-4- نکات قابل توجه در مورد دستورالعمل SFT(010)

آخرین بیت سمت چپ در محدوده شیفت با هر بار شیفت داده ها از بین خواهد رفت. زمانی که Reset Input فعال می شود تمام بیت‌های موجود از Word آغازین تا پایانی صفر می شوند. این ورودی نسبت به ورودی های دیگر دستورالعمل از الویت برخوردار می باشد بدین منظور که با فعال شدن Reset Input دیگر شیفتی انجام نخواهد شد. همواره باید شماره آدرس Word آغازین از شماره آدرس پایانی کمتر و یا مساوی با آن باشد ولی اگر حتی شماره آدرس Word آغازین از پایانی بیشتر باشد نیز خطایی در نظر گرفته نخواهد شد و داده های Word آغازین شیفت پیدا خواهد نمود. زمانی که برای آدرس دهی از روش آدرس دهی غیرمستقیم استفاده می شود و به آدرس هایی غیر از آدرس های مجاز برای عملگرها اشاره گردد خطایی روی داده است و این امر به عنوان خطا در نظر گرفته خواهد شد. و فلگ مربوط به خطا روشن خواهد شد.

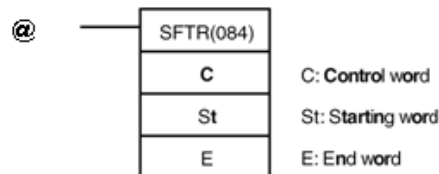
3-1-5- مثال از عملکرد دستور (SFT(010)

مثال زیر چگونگی عملکرد یک شیفت رجیستر را که از سه Word یعنی 48 بیت که آدرس CIO128 تا CIO130 را در برمی گیرد را نشان می دهد. از یک پالس 1s به عنوان ورودی استفاده شده است که حالت ورودی CIO0.05 را به عنوان داده ورودی وارد شیفت رجیستر می نماید. هر یک ثانیه داده های موجود از بیت CIO128.00 تا CIO130.15 یک بیت به سمت چپ شیفت پیدا می کند.



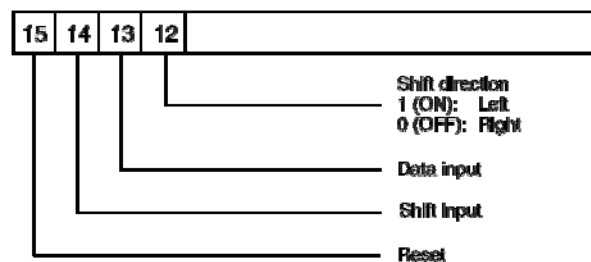
3-2- ثبات تغییر مکان برگشت پذیر (Reversible Shift Register-SFTR)

این دستورالعمل شیفت رجیستری با قابلیت شیفت داده ها به صورت بیتی به سمت چپ و راست را ایجاد می نماید.



3-2-1- مشخصات عملگرهای دستور SFTR(084)

Control Word : C

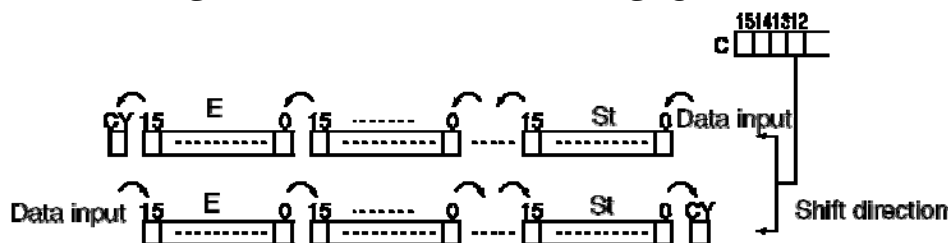


فضای حافظه	C	St	E
CIO Area	CIO 0000 to CIO 6143		
Work Area	W000 to W511		
Holding Bit Area	H000 to H511		
Auxiliary Bit Area	A000 to A959	A484 to A959	
Timer Area	T0000 to T4095		
Counter Area	C0000 to C4095		
DM Area	D00000 to D32767		
EM Area without bank	E00000 to E32767		
EM Area with bank	En_00000 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	---		
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		

نکته: دو Word انتخاب شده به عنوان آغاز و پایان محدوده شیفت داده ها باید از یک منطقه از حافظه باشند.

3-2-2- شرح عملکرد دستور SFTR(084)

زمانی که شرط عملکرد بیت ورودی شیفت رجیستر (بیت 14 از کنترل Word) از حالت خاموش به روشن تغییر وضعیت دهد، تمام داده ها از St تا E با توجه به جهت تعیین شده برای شیفت (بیت 12 از کنترل Word)، به صورت بیتی تغییر مکان می دهند. و همچنین حالت خاموش یا روشن بودن بیت مربوط به داده ورودی از سمت چپ و یا راست با توجه به جهت شیفت داده ها، وارد شیفت رجیستر می گردد. در این حالت داده هایی که از شیفت رجیستر خارج می گردند، در Carry Flag (CY) قرار می گیرند.



3-2-3- فلگ های مربوط به دستورالعمل SFTR(084)

نام	نشانه	عملکرد
Error Flag	ER	زمانی که St از E بزرگتر باشد این فلگ روشن می شود. در بقیه موارد این فلگ خاموش می باشد.
Carry Flag	CY	روشن زمانی که "1" از شیفت رجیستر خارج شود. خاموش زمانی که "0" از شیفت رجیستر خارج گردد. خاموش زمانی که ورودی Reset فعال گردد.

3-2-4- نکات قابل توجه در مورد دستورالعمل SFTR(084)

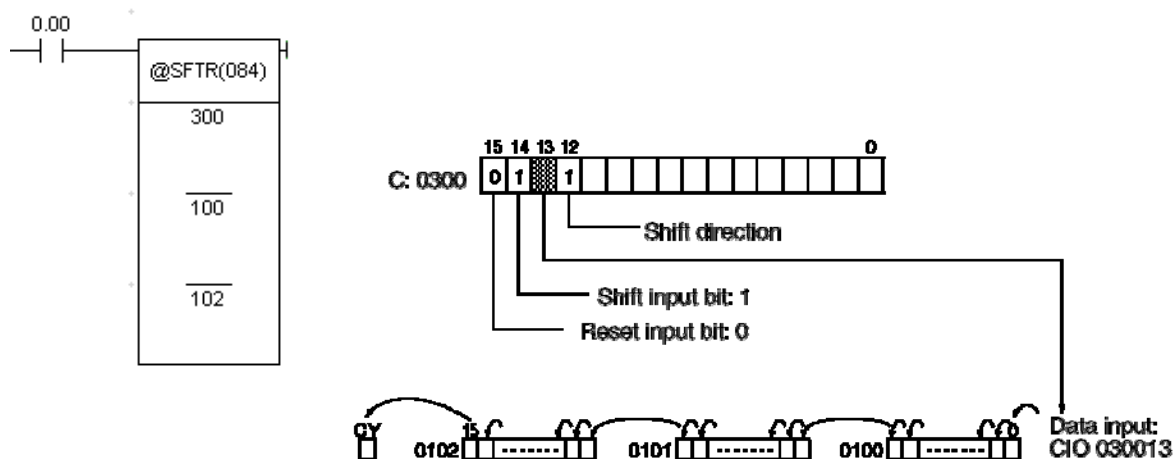
تمام عملیات مربوط به شیفت داده ها فقط زمانی که بیت مربوط به Reset Input خاموش باشد انجام می گیرد. زمانی که این بیت (Reset Input) یک باشد تمام بیت‌های موجود در شیفت رجیستر از ابتدا تا انتها صفر خواهد شد.

زمانی که شماره Word انتخاب شده به عنوان St از E بیشتر باشد Error Flag به منظور نشان دادن این خطا روشن خواهد شد.

3-2-5- مثال از عملکرد دستور SFTR(084)

شیفت داده ها

زمانی که ورودی شیفت رجیستر CIO300.14 روشن شود زمانی که CIO0.00 روشن است و همچنین بیت مربوط به ورودی Reset، CIO300.15 خاموش است، داده های موجود در Word های CIO100 تا CIO102 یک بیت بسته به جهت شیفت داده ها که توسط بیت CIO300.12 (به عنوان مثال 1: سمت راست) تعیین می گردد، شیفت پیدا می نماید. و حالت ورودی شیفت CIO300.13 وارد بیت سمت راست شیفت رجیستر CIO100.00 می گردد و محتوای CIO102.15 به فلگ Carry (P_CY) فرستاده خواهد شد.



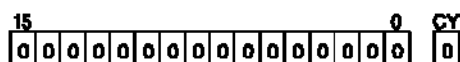
Reset کردن داده ها

زمانی که CIO0.00 روشن می باشد اگر CIO300.15 روشن گردد، تمام بیتهای CIO100.00 تا CIO102.15 و همچنین P_CY صفر خواهند شد.

کنترل داده ها

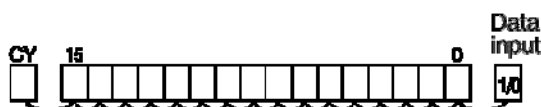
Reset کردن:

تمام بیتهای موجود در St تا E صفر شده و تا زمانی که بیت مربوط به ورودی Reset روشن می باشد (CIO300.15) هیچ داده ای وارد شیفت رجیستر نخواهد شد



شیفت داده ها به سمت چپ:

زمانی که جهت شیفت داده ها به سمت چپ انتخاب می گردد، محتوای بیت مربوط به داده ورودی وارد اولین بیت St می گردد و داده ها را به صورت بیتی به سمت چپ شیفت می دهد و محتوای آخرین بیت E به P_CY منتقل می شود.



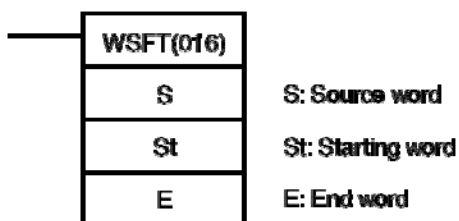
شیفت داده ها به سمت راست:

زمانی که جهت شیفت داده ها به سمت راست انتخاب می گردد، محتوای بیت مربوط به داده ورودی وارد آخرین بیت E می گردد و داده ها را به صورت بیتی به سمت راست شیفت می دهد و محتوای اولین بیت St به P_CY منتقل می شود.



3-3- تغییر مکان (Word Shift-WSFT) Word

این دستورالعمل داده ها را به صورت Word بین St و E شیفت می دهد.

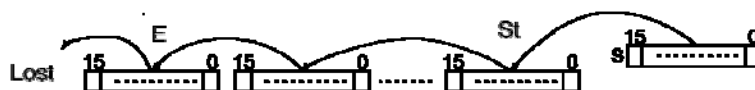


3-3-1- مشخصات عملگرهای دستور (016) WSFT

فضای حافظه	S	St	E
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	---	
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		

3-3-2- شرح عملکرد دستور (016) WSFT

این دستورالعمل داده ها را در واحد Word از St تا E شیفت داده و محتوای موجود در S وارد اولین Word (St) شده و محتوای E از بین خواهد رفت.



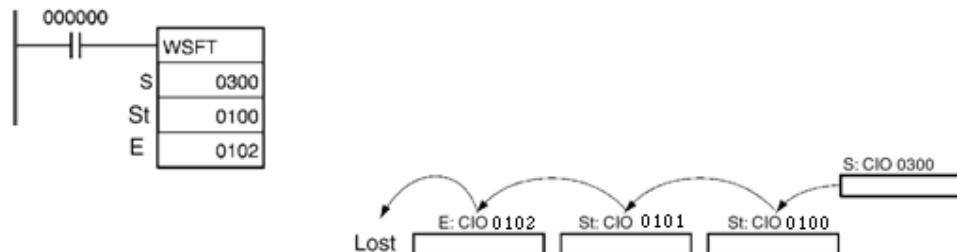
3-3-3- فلگ های مربوط به دستورالعمل (016) WSFT

نام	نشانه	عملکرد
Error Flag	ER	زمانی که St از E بزرگتر باشد این فلگ روشن می شود. در بقیه موارد این فلگ خاموش می باشد.

نکته: زمانی که حجم بالایی از داده ها شیفت داده می شوند باید به این نکته توجه شود که زمان اجرای دستورالعمل طولانی تر خواهد شد. باید از قطع نشدن تغذیه PLC هنگام اجرای این دستورالعمل اطمینان حاصل نمود چون این عمل باعث متوقف شدن اجرای این دستور در اواسط اجرا می گردد.

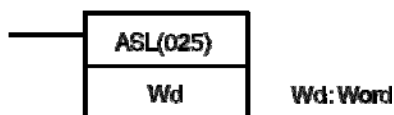
3-3-4- مثال از عملکرد دستور (016) WSFT

زمانی که بیت CIO0.00 روشن می گردد داده ها به صورت Word از CIO100 تا CIO102 شیفت داده می شوند. محتوای CIO300 به CIO100 انتقال داده شده و محتوای CIO102 از بین خواهد رفت.



3-4-4- تغییر مکان حسابی به چپ (Arithmetic Shift Left-ASL)

این دستورالعمل محتوای Word مشخص شده را یک بیت به سمت چپ شیفت میدهد.

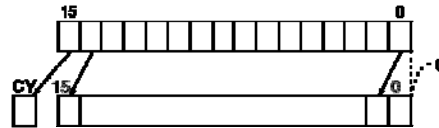


3-4-4-1- مشخصات عملگرهای دستور (025) ASL

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)
Indirect DM/EM addresses in BCD	*D00000 to *D32767 *E00000 to *E32767 *En_00000 to *En_32767 (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

3-4-2- شرح عملکرد دستور (025) ASL

این دستورالعمل محتوای Wd را یک بیت به سمت چپ شیفت می دهد (از سمت راست به سمت چپ) و عدد 0 را جایگزین اولین بیت سمت راست می نماید همچنین محتوای آخرین بیت سمت چپ در P_CY قرار می گیرد.

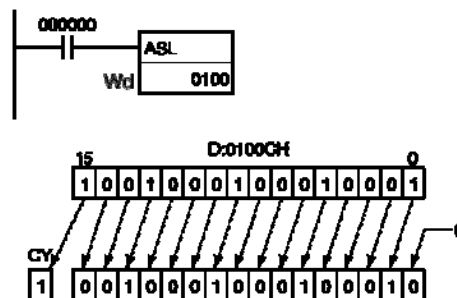


3-4-3- فلگ های مربوط به دستورالعمل (025) ASL

نام	نشانه	عملکرد
Error Flag	ER	خاموش
Equal Flag	=	روشن زمانی که نتیجه شیفت 0 باشد. دربقیه موارد خاموش.
Carry Flag	CY	روشن زمانی که "1" از شیفت رجیستر خارج شود. در بقیه موارد خاموش می باشد.
Negative Flag	N	زمانی که آخرین بیت سمت چپ 1 می شود روشن خواهد شد.

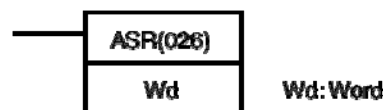
3-4-4- مثال از عملکرد دستور (025) ASL

زمانی که CIO0.00 روشن می گردد محتوای حافظه CIO100 یک بیت به سمت چپ شیفت داده می شود، صفر در اولین بیت سمت راست CIO100.00 وارد شده و آخرین بیت سمت چپ CIO100.15 به P_CY منتقل می شود.



3-5- تغییر مکان حسابی به راست (Arithmetic Shift Right-ASR)

این دستورالعمل محتوای Word مشخص شده را یک بیت به سمت راست شیفت میدهد.

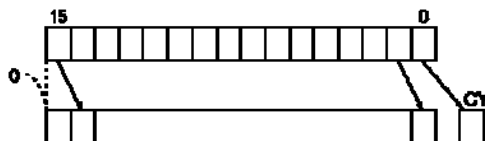


3-5-1- مشخصات عملگرهای دستور (026) ASR

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)
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(++), ,IR15(++) ,-(--) IR0 to ,-(--) IR15

3-5-2- شرح عملکرد دستور (026) ASR

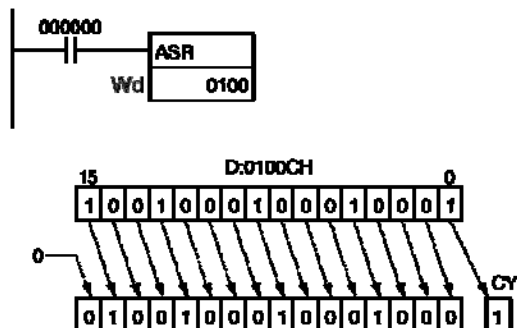
این دستورالعمل محتوای Wd را یک بیت به سمت راست شیفت می دهد (از سمت چپ به سمت راست) و عدد 0 را جایگزین اولین بیت سمت چپ می نماید همچنین محتوای آخرین بیت سمت راست در P_CY قرار می گیرد.


3-5-3- فلگ های مربوط به دستورالعمل (026) ASR

نام	نشانه	عملکرد
Error Flag	ER	خاموش
Equal Flag	=	روشن زمانی که نتیجه شیفت 0 باشد در بقیه موارد خاموش.
Carry Flag	CY	روشن زمانی که "1" از شیفت رجیستر خارج شود. در بقیه موارد خاموش می باشد.
Negative Flag	N	خاموش

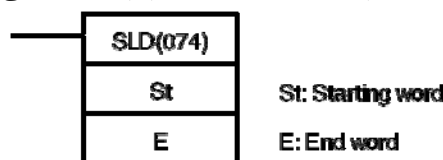
3-5-4- مثال از عملکرد دستور (026) ASR

زمانی که CIO0.00 روشن می گردد محتوای حافظه CIO100 یک بیت به سمت راست شیفت داده می شود، صفر در اولین بیت سمت چپ CIO100.15 وارد شده و آخرین بیت سمت راست CIO100.00 به P_CY منتقل می شود.



3-6-6- تغییر مکان یک دیجیت به چپ (One Digit Shift Left-SLD)

این دستورالعمل داده ها را یک دیجیت (چهار بیت) به سمت چپ شیفت می دهد.



3-6-6-1- مشخصات عملگرهای دستور (074) SLD

فضای حافظه	St	E
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)	
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	

3-6-2- شرح عملکرد دستور (074) SLD

این دستورالعمل داده های موجود در St تا E را به صورت دیجیتی (4 بیتی) به سمت چپ شیفت می دهد. در دیجیت سمت راست (بیت 0 تا 3) وارد شده و محتوای دیجیت سمت چپ (بیت 12 تا 15) از بین می رود.

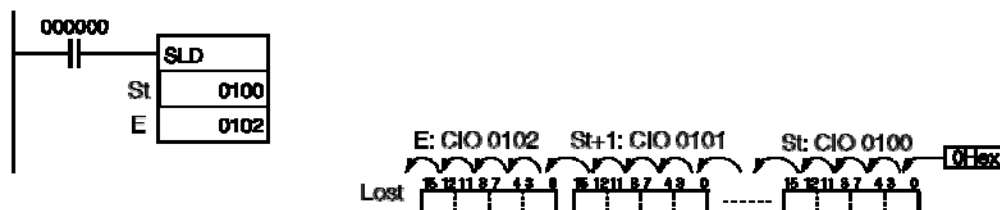


3-6-3- فلگ های مربوط به دستورالعمل (074) SLD

نام	نشانه	عملکرد
Error Flag	ER	زمانی که St از E بزرگتر باشد این فلگ روشن می شود. در بقیه موارد این فلگ خاموش می باشد.

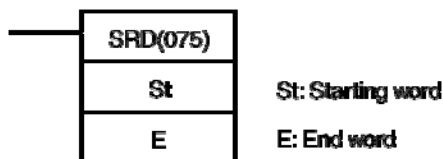
3-6-4- مثال از عملکرد دستور (074) SLD

زمانی که CIO0.00 روشن می گردد داده های موجود در CIO100 تا CIO102 به صورت دیجیتی (4 بیتی) به سمت چپ شیفت پیدا می کند. در بیت های 0 تا 3 حافظه CIO100 صفر قرار می گیرد و بیت های 12 تا 15 حافظه CIO102 از بین می رود.



3-7- تغییر مکان یک دیجیت به راست (One Digit Shift Right-SRD)

این دستورالعمل داده ها را یک دیجیت (چهار بیت) به سمت راست شیفت می دهد.



3-7-1- مشخصات عملگرهای دستور (075) SRD

فضای حافظه	St	E
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)	
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(++), IR15(++), ,-- IR0 to ,-- IR15	

3-7-2- شرح عملکرد دستور (075) SRD

این دستورالعمل داده های موجود در St تا E را به صورت دیجیتی (4 بیتی) به سمت راست شیفت می دهد. در دیجیت سمت چپ (بیت 12 تا 15) "0" وارد شده و محتوای دیجیت سمت راست (بیت 0 تا 3) از بین می رود.

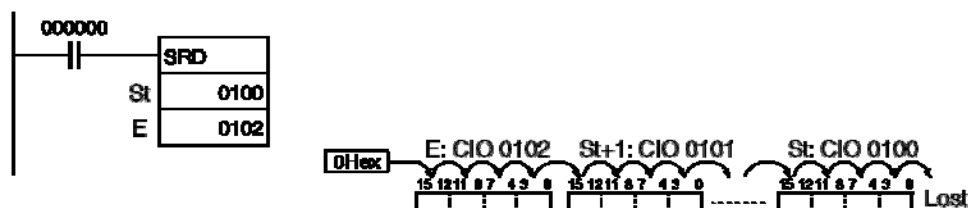


3-7-3- فلگ های مربوط به دستورالعمل (075) SRD

نام	نشانه	عملکرد
Error Flag	ER	زمانی که St از E بزرگتر باشد این فلگ روشن می شود. در بقیه موارد این فلگ خاموش می باشد.

3-7-4- مثال از عملکرد دستور (SRD(075

زمانی که CIO0.00 روشن می گردد داده های موجود در CIO100 تا CIO102 به صورت دیجیتی (4 بیتی) به سمت راست شیفت پیدا می کند. در بیت‌های 12 تا 15 حافظه CIO102 صفر قرار می گیرد و بیت‌های 0 تا 3 حافظه CIO100 از بین می رود.



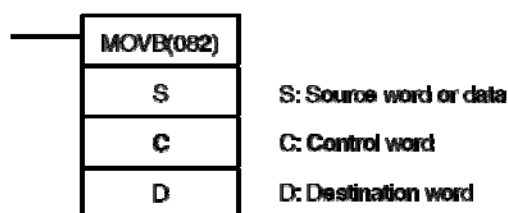
4- دستورات عملیاتی تغییر مکان داده ها

دستورات عملیاتی تغییر مکان داده ها شامل دستوراتی است که موجب حرکت داده ها در فرمت ساده بیتی و یا به صورت Word و در بعضی موارد به صورت اجتماعی از Word ها می باشد. این داده ها می توانند از یک ناحیه از حافظه به ناحیه ای دیگر با حفظ حالت خود تغییر مکان دهند. این دستورات عملیاتی با دستورات شیفت داده ها که باعث شیفت داده ها در یک ناحیه مشخص شده از حافظه می شوند، متفاوت می باشد. در این بخش به شرح عملکرد دستورات زیر پرداخته می شود.

کد دستورالعمل	دستور العمل	نام دستور العمل
082	MOVB	تغییر مکان بیتی (Move Bit)
083	MOVD	تغییر مکان دیجیتی (Move Digit)
070	XFER	انتقال یک بلوک از حافظه (Block Transfer)
071	BSET	بار کردن یک بلوک از حافظه (Block Set)

4-1- تغییر مکان بیتی (Move Bit-MOVB)

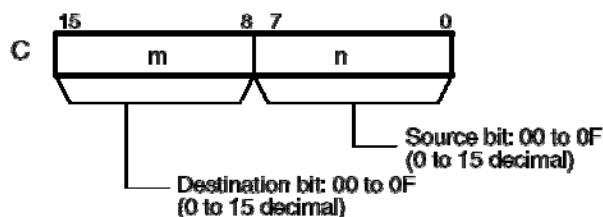
این دستورالعمل بیت مشخصی را انتقال می دهد.



4-1-1- مشخصات عملگرهای دستور (MOV B(082)

C: Control Word

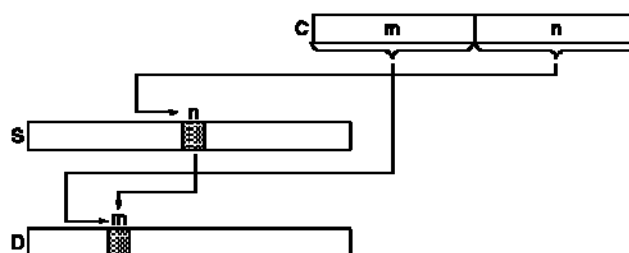
در دو دیجیت (رقم) سمت راست کنترل Word باید شماره بیت مورد نظر از میدا (S) برای انتقال تعیین گردد و در دو دیجیت سمت چپ این Word شماره بیت مورد نظر از مقصد (D) باید تعیین نمود.



فضای حافظه	S	C	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		A448 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	فقط مقدار معین	---
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		

4-1-2- شرح عملکرد دستور (MOV B(082)

این دستورالعمل یک بیت مشخص (n) از مبدأ (S) را به یک بیت مشخص (m) از مقصد (D) انتقال می دهد. بقیه بیتها در D بدون تغییر باقی می مانند.



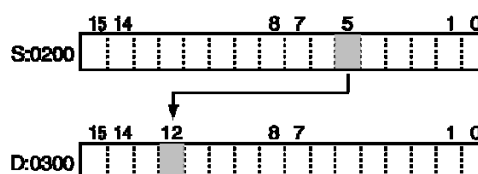
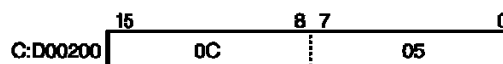
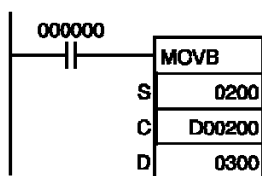
می توان از یک آدرس به عنوان مبدأ و مقصد استفاده نمود.

4-1-3- فلگ های مربوط به دستورالعمل (MOV B(082)

نام	نشانه	عملکرد
Error Flag	ER	در صورتی که محتوای C به عنوان تعیین کننده بیت مبدأ و مقصد خارج از محدوده 00 تا 0F باشد این فلگ روشن می گردد. در بقیه موارد خاموش است.

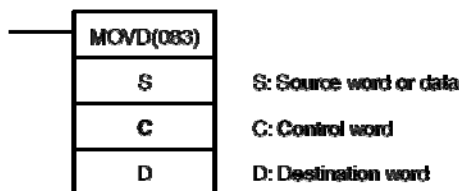
4-1-4- مثال از عملکرد دستور (MOV B(082)

در این مثال زمانی که بیت CIO0.00 روشن می گردد، با توجه به محتوای Control Word که 0C05 می باشد، محتوای بیت پنجم از مبدأ CIO200 به بیت دوازدهم از مقصد CIO300 منتقل می گردد.



4-2-2- تغییر مکان دیجیتی (Move Digit-MOVD)

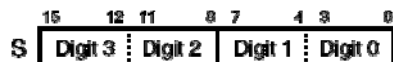
این دستورالعمل دیجیت و یا دیجیتهای مشخصی را در فضای حافظه حرکت میدهد.



4-2-1- مشخصات عملگرهای دستور MOVD(083)

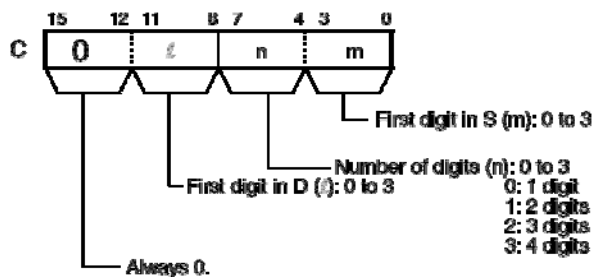
S: مبدأ

رقم ها و دیجیت های مبدأ از سمت راست به چپ خوانده می شوند. در صورت نیاز به دیجیت صفر برگردانده می شود.



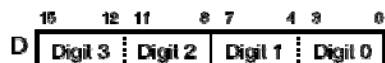
C: کنترل

سه رقم اول این Word آنچنانکه در زیر نیز مشاهده می شود، مشخص کننده اولین دیجیت مبدأ (m) تعداد دیجیت ها برای انتقال (n) و اولین دیجیت مقصد (l) می باشد.



D: مقصد

این Word از سمت راست به چپ نوشته می شود و در صورت نیاز به دیجیت صفر برگردانده می شود.

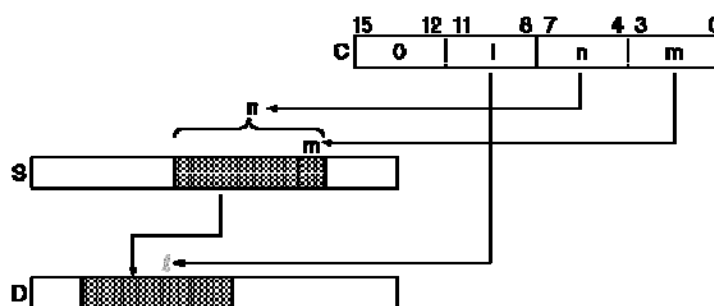


فضای حافظه	S	C	D
CIO Area	CIO 0000 to CIO 6143		
Work Area	W000 to W511		
Holding Bit Area	H000 to H511		
Auxiliary Bit Area	A000 to A959		A448 to A959
Timer Area	T0000 to T4095		
Counter Area	C0000 to C4095		
DM Area	D00000 to D32767		
EM Area without bank	E00000 to E32767		
EM Area with bank	En_00000 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	فقط مقدار معین	---
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		

4-2-2- شرح عملکرد دستور MOVD(083)

این دستور محتوای دیجیت های معین از S (شروع شده از دیجیت m) را به دیجیت های D (که از l شروع شده اند) منتقل می نماید. فقط دیجیتهای معین شده در D تحت تأثیر قرار می گیرند، بقیه دیجیتهای بدون تغییر باقی می مانند.

اگر تعداد دیجیتهای تعیین شده به منظور خواندن و یا نوشتن داده در S و D از تعداد دیجیتهای موجود تجاوز نماید، عملکرد دستور MOVD(083) به دیجیتهای سمت راست بازگردانده خواهد شد.



نکته: می توان از یک آدرس از حافظه به عنوان S و D استفاده نمود.

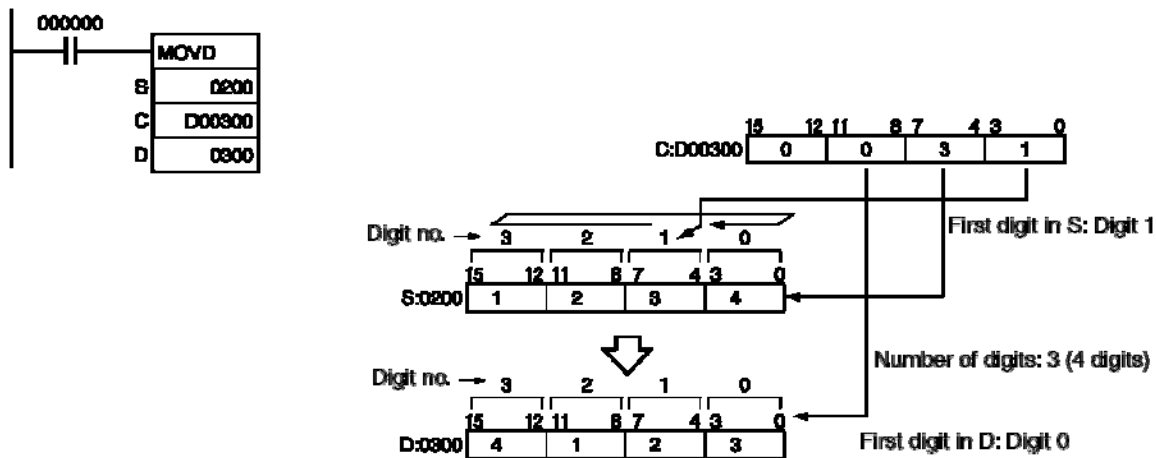
4-2-3- فلگ های مربوط به دستورالعمل (083) MOVD

نام	نشانه	عملکرد
Error Flag	ER	در صورتی که رقم تعیین شده به عنوان اولین دیجیت مبدأ، مقصد و یا تعداد دیجیتها از محدوده 0 تا 3 خارج باشد این فلگ روشن می گردد. در بقیه موارد خاموش است.

4-2-4- مثال از عملکرد دستور (083) MOVD

انتقال چهار دیجیت:

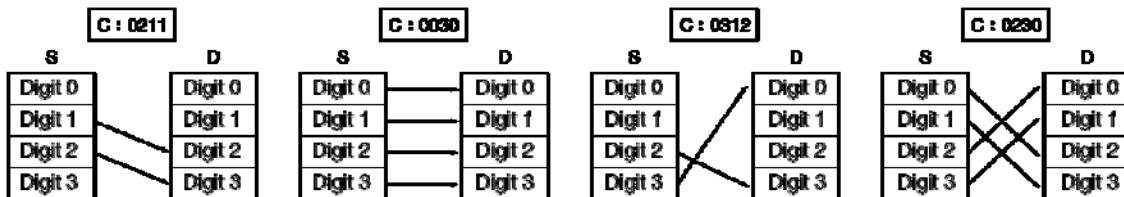
در مثال زیر زمانی که بیت ورودی CIO0.00 روشن می گردد، چهار دیجیت از داده های از CIO200 کپی شده و به CIO300 منتقل می گردد. این انتقال از دیجیت شماره 1 از CIO200 آغاز می گردد و به دیجیت شماره 0 به بعد CIO300 انتقال می یابد. این انتقال با توجه به محتوای Control Word که در این مثال #31 است، انجام می گیرد.



نکته: در این مثال بعد از خواندن آخرین دیجیت سمت چپ S (دیجیت 3) دستور (083) MOVD به خواندن اولین دیجیت سمت راست S (دیجیت 0) می پردازد.

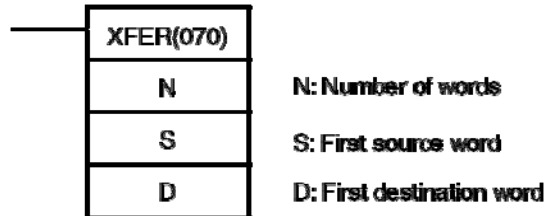
مثال از تعیین C:

در دیاگرام زیر مثالهایی از انتقال داده ها در صورت تعیین مقادیر مختلف C بیان شده است.



3-4- انتقال یک بلوک از حافظه (Block Transfer-XFER)

این دستورالعمل تعداد مشخصی از Word ها را در فضای حافظه انتقال می دهد.



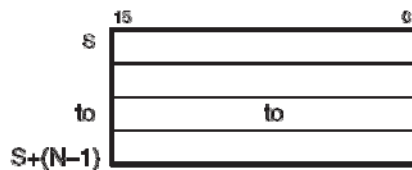
1-3-4- مشخصات عملگرهای دستور XFER(070)

N: تعداد Word ها

باید تعداد Word های مورد نظر برای انتقال را با این عملگر تعیین نمود. محدوده مجاز برای این عملگر 0000 تا FFFF می باشد.

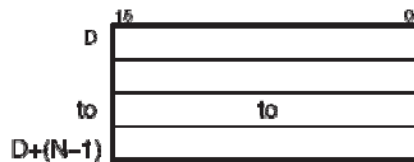
S: اولین Word از مبدأ

تعیین کننده اولین آدرس از حافظه برای انتقال می باشد.



D: اولین Word از مقصد

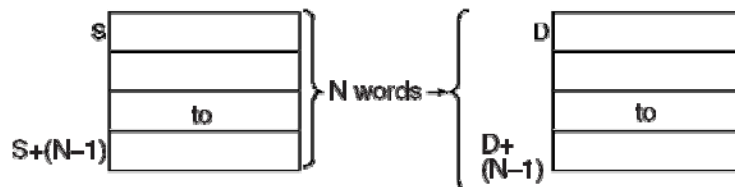
تعیین کننده اولین آدرس از مقصد برای نوشتن داده ها می باشد.



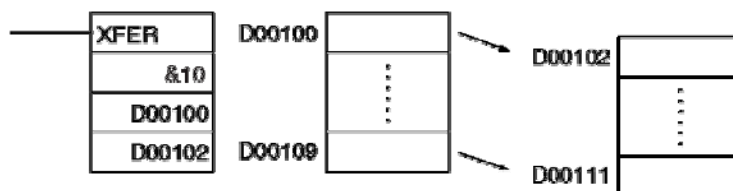
فضای حافظه	N	S	D
CIO Area	CIO 0000 to CIO 6143		
Work Area	W000 to W511		
Holding Bit Area	H000 to H511		
Auxiliary Bit Area	A000 to A959	A448 to A959	
Timer Area	T0000 to T4095		
Counter Area	C0000 to C4095		
DM Area	D00000 to D32767		
EM Area without bank	E00000 to E32767		
EM Area with bank	En_00000 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	---	---
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		

4-3-2- شرح عملکرد دستور XFER(070)

این دستورالعمل محتوای تعداد معین (N) از Word های حافظه را که از آدرس S آغاز می گردند (S تا S+(N-1)) به منطقه ای از حافظه که با D آغاز می گردد (D تا D+(N-1)) انتقال می دهد.



تداخل حافظه بین مبدأ و مقصد امکان پذیر می باشد، در این صورت (070) XFER مانند یک شیفت به صورت Word عمل می کند.



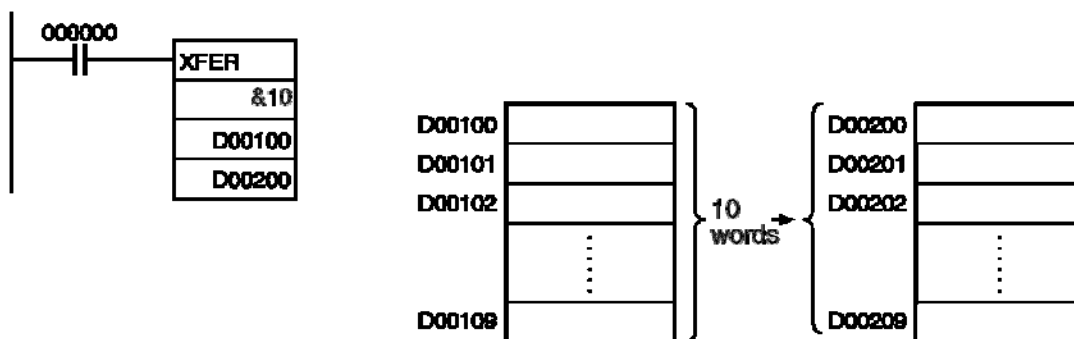
4-3-3- نکات قابل توجه در مورد دستورالعمل XFER(070)

باید به این نکته توجه شود که آدرس های مبدأ S تا S+(N-1) و همچنین آدرس های مقصد D تا D+(N-1) از محدوده انتهای حافظه تجاوز ننمایند.

زمانی که تعداد زیادی از Word ها برای انتقال تعیین می گردند زمان زیادی برای این انتقال لازم می باشد. در صورت ایجاد وقفه در تغذیه PLC هنگام اجرای این دستورالعمل انتقال به درستی انجام نخواهد شد.

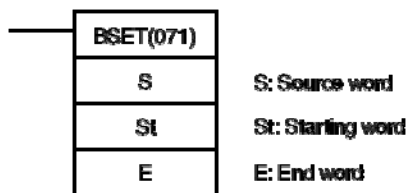
4-3-5- مثال از عملکرد دستور XFER(070)

در مثال زیر زمانی که CIO0.00 روشن می گردد، 10Word از D100 تا D109 به D200 تا D209 انتقال می یابد.



4-4- بار کردن یک بلوک از حافظه (Block Set-BSET)

این دستورالعمل محتوای یک Word را در یک بلوک از حافظه کپی می نماید.



4-4-1- مشخصات عملگرهای دستور BSET(071)

S: مرجع

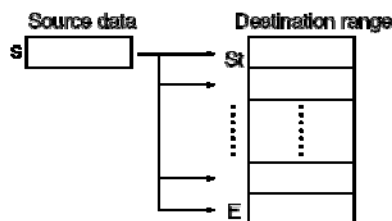
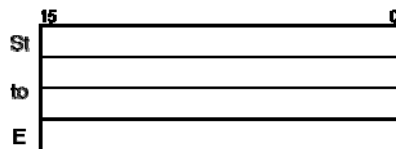
شامل مقدار مرجع و یا آدرس Word محتوای مقدار مرجع می باشد.

St: آدرس آغازین

شامل اولین Word از مقصد می باشد.

E: آدرس پایانی

شامل آخرین Word از مقصد می باشد.

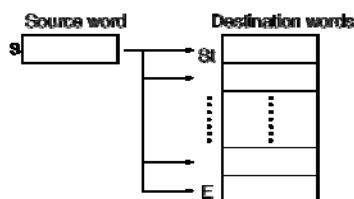


نکته : آدرسهای St و E باید در یک ناحیه از حافظه باشند.

فضای حافظه	S	St	E
CIO Area	CIO 0000 to CIO 6143		
Work Area	W000 to W511		
Holding Bit Area	H000 to H511		
Auxiliary Bit Area	A000 to A959	A448 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	---	
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		

4-4-2- شرح عملکرد دستور (BSET(071)

این دستورالعمل یک داده مرجع (S) را در تمام محدوده مشخص شده از St تا E کپی می نماید.



4-4-3- فلگ های مربوط به دستورالعمل (BSET(071)

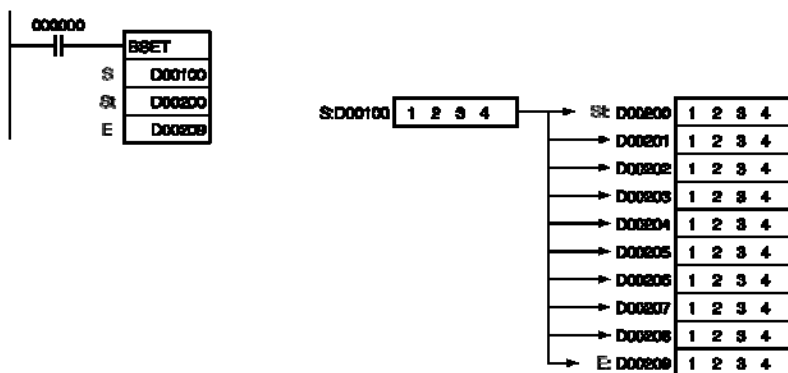
نام	نشانه	عملکرد
Error Flag	ER	در صورتی که St از E بزرگتر باشد این فلگ روشن خواهد شد. در بقیه موارد خاموش است.

نکته : همواره باید به این نکته توجه داشت که St و E باید در یک ناحیه از حافظه باشند و همچنین $St < E$ باشد.

زمانی که محتوای S به تعداد زیادی از Word ها انتقال می یابد زمان زیادی برای این انتقال لازم می باشد. در صورت ایجاد وقفه در تغذیه PLC هنگام اجرای این دستورالعمل انتقال به درستی انجام نخواهد شد.

4-4-4- مثال از عملکرد دستور (BSET(071)

در این مثال زمانی که بیت CIO0.00 روشن می گردد محتوای D100 از D200 تا D209 انتقال می یابد.



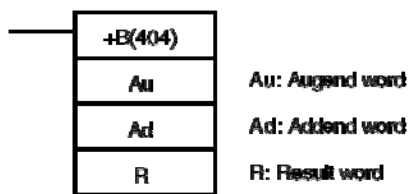
5- دستورالعملهای محاسباتی

در این بخش به توضیح دستورالعملهای محاسباتی که برای اجرای عملیات ریاضی در فرمت BCD و یا هگزادسیمال مورد استفاده قرار می گیرند می پردازیم.

کد دستورالعمل	دستور العمل	نام دستور العمل
404	+B	جمع BCD بدون Carry (BCD Add)
400	+	جمع باینری بدون Carry (Binary Add)
414	-B	تفریق BCD بدون Carry (BCD Subtract)
410	-	تفریق باینری بدون Carry (Binary Subtract)
424	*B	ضرب BCD بدون Carry (BCD Multiply)
420	*	ضرب باینری بدون Carry (Binary Multiply)
434	/B	تقسیم BCD بدون Carry (BCD Devide)
430	/	تقسیم باینری بدون Carry (Binary Devide)

5-1- جمع BCD بدون Carry (+B) (BCD Add)

چهار رقم داده های ثابت و یا محتوای یک Word از حافظه را با فرمت BCD جمع می نماید.

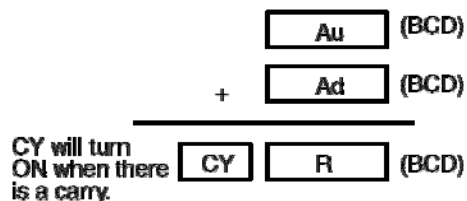


5-1-1- مشخصات عملگرهای دستور +B(404)

فضای حافظه	Au	Ad	R
CIO Area	CIO 0000 to CIO 6143		
Work Area	W000 to W511		
Holding Bit Area	H000 to H511		
Auxiliary Bit Area	A000 to A959		A448 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_32767 (n = 0 to C)		
Constants	#0000 to #9999		---
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		

5-1-2- شرح عملکرد دستور +B(404)

این دستورالعمل داده های BCD را که در Au و Ad وجود دارند جمع نموده و نتیجه را در R می ریزد.

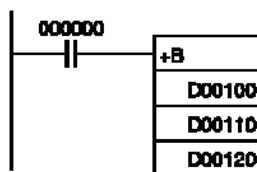


5-1-3- فلگ های مربوط به دستورالعمل +B(404)

نام	نشانه	عملکرد
Error Flag	ER	زمانی که محتوای Au یا Ad مقدار BCD نباشد این فلگ روشن خواهد شد. در بقیه موارد خاموش می باشد.
Equal Flag	=	روشن زمانی که نتیجه جمع 0 باشد. در بقیه موارد خاموش.
Carry Flag	CY	زمانی که نتیجه عملیات حاوی Carry باشد روشن می گردد. در بقیه موارد خاموش می باشد.

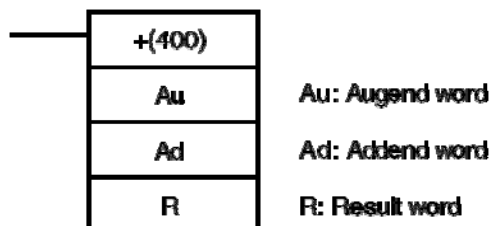
5-1-4- مثال از عملکرد دستور +B(404)

در مثال زیر هنگامی که بیت CIO0.00 روشن می گردد داده های D100 و D110 به صورت BCD با هم جمع شده و نتیجه در D120 ریخته می شود.



5-2- جمع باینری بدون Carry (+ Binary Add)

چهار رقم داده های ثابت و یا محتوای یک Word از حافظه را با فرمت هگزادسیمال جمع می نماید.

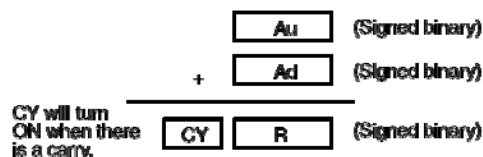


5-2-1- مشخصات عملگرهای دستور (400) +

فضای حافظه	Au	Ad	R
CIO Area	CIO 0000 to CIO 6143		
Work Area	W000 to W511		
Holding Bit Area	H000 to H511		
Auxiliary Bit Area	A000 to A959		A448 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		---
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		

5-2-2- شرح عملکرد دستور (400) +

این دستورالعمل داده های باینری Au و Ad را جمع نموده و نتیجه را در R می ریزد.

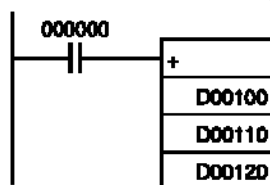


5-2-3- فلگ های مربوط به دستورالعمل (400) +

نام	نشانه	عملکرد
Error Flag	ER	خاموش.
Equal Flag	=	روشن زمانی که نتیجه جمع 0 باشد در بقیه موارد خاموش.
Carry Flag	CY	زمانی که نتیجه عملیات حاوی Carry باشد روشن می گردد. در بقیه موارد خاموش می باشد.
Overflow Flag	OF	زمانی که نتیجه جمع دو مقدار مثبت باینری بین 8000 تا FFFF باشد این فلگ روشن می گردد. در بقیه موارد خاموش می باشد.
Underflow Flag	UF	زمانی که نتیجه جمع دو مقدار منفی باینری بین 0000 تا 7FFF باشد این فلگ روشن می گردد. در بقیه موارد خاموش می باشد.
Negative Flag	N	زمانی که بیت سمت چپ نتیجه یک باشد این فلگ روشن می شود. در بقیه موارد خاموش می باشد.

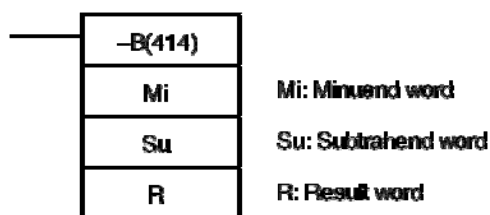
5-2-4- مثال از عملکرد دستور (400) +

در مثال زیر هنگامی که بیت CIO0.00 روشن می گردد داده های D100 و D110 به صورت هگزادسیمال با هم جمع شده و نتیجه در D120 ریخته می شود.



3-5-3- تفریق BCD بدون Carry (BCD Subtract,-B)

چهار رقم داده های ثابت و یا محتوای یک Word از حافظه را با فرمت BCD تفریق می نماید.

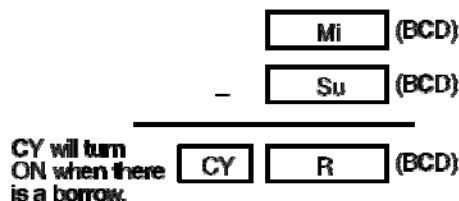


3-5-1- مشخصات عملگرهای دستور -B(414)

فضای حافظه	Mi	Su	R
CIO Area	CIO 0000 to CIO 6143		
Work Area	W000 to W511		
Holding Bit Area	H000 to H511		
Auxiliary Bit Area	A000 to A959		A448 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 #9999	---	
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		

5-3-2- شرح عملکرد دستور (B(414)-

این دستورالعمل داده های BCD در Su را از داده BCD در Mi کم نموده و نتیجه را در R می ریزد. در صورتی که نتیجه این تفریق منفی باشد نتیجه به صورت مکمل 10 نمایش داده خواهد شد.

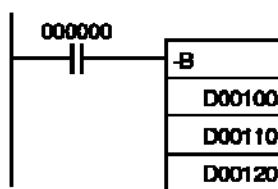


5-3-3- فلگ های مربوط به دستورالعمل (B(414)-

نام	نشانه	عملکرد
Error Flag	ER	زمانی که محتوای Mi یا Su مقدار BCD نباشد این فلگ روشن خواهد شد. در بقیه موارد خاموش می باشد.
Equal Flag	=	روشن زمانی که نتیجه عملیات 0 باشد در بقیه موارد خاموش.
Carry Flag	CY	زمانی که نتیجه عملیات قرض گرفتن باشد روشن می گردد. در بقیه موارد خاموش می باشد.

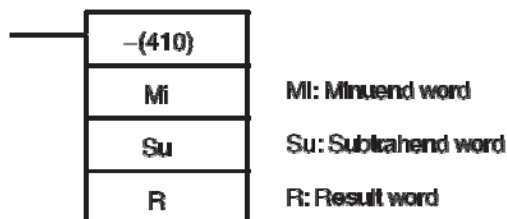
5-3-4- مثال از عملکرد دستور (B(414)-

در مثال زیر هنگامی که بیت CIO0.00 روشن می گردد داده های D110 از داده موجود در D100 به صورت BCD کم شده و نتیجه در D120 ریخته می شود



5-4-4- تفریق باینری بدون Carry (-) Binary Subtrct

چهار رقم داده های ثابت و یا محتوای یک Word از حافظه را با فرمت هگزادسیمال از هم کم می نماید.

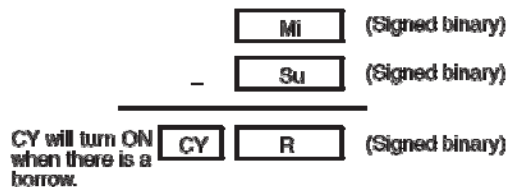


5-4-1- مشخصات عملگرهای دستور (410) -

فضای حافظه	Mi	Su	R
CIO Area	CIO 0000 to CIO 6143		
Work Area	W000 to W511		
Holding Bit Area	H000 to H511		
Auxiliary Bit Area	A000 to A959		A448 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		---
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		

5-4-2- شرح عملکرد دستور (410) -

این دستورالعمل داده های باینری در Mi را از داده باینری در Su کم نموده و نتیجه را در R می ریزد. در صورتی که نتیجه این تفریق منفی باشد نتیجه به صورت مکمل 2 نمایش داده خواهد شد.

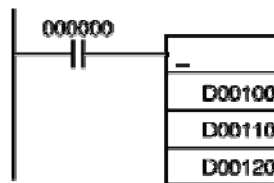


5-4-3- فلگ های مربوط به دستورالعمل (410) -

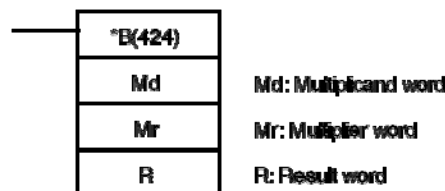
نام	نشانه	عملکرد
Error Flag	ER	خاموش.
Equal Flag	=	روشن زمانی که نتیجه تفریق 0 باشد در بقیه موارد خاموش.
Carry Flag	CY	زمانی که نتیجه عملیات قرض گرفتن باشد روشن می گردد. در بقیه موارد خاموش می باشد.
Overflow Flag	OF	زمانی که نتیجه تفریق یک مقدار منفی از یک مقدار مثبت باینری 8000 تا FFFF باشد این فلگ روشن می گردد. در بقیه موارد خاموش می باشد
Underflow Flag	UF	زمانی که نتیجه تفریق یک مقدار منفی از یک مقدار مثبت باینری بین 0000 تا 7FFF باشد این فلگ روشن می گردد. در بقیه موارد خاموش می باشد
Negative Flag	N	زمانی که بیت سمت چپ نتیجه یک باشد این فلگ روشن می شود. در بقیه موارد خاموش می باشد.

5-4-4- مثال از عملکرد دستور (410) -

در مثال زیر هنگامی که بیت CIO0.00 روشن می گردد داده های D110 از داده موجود در D100 به صورت باینری کم شده و نتیجه در D120 ریخته می شود.


5-5- ضرب BCD بدون Carry (*B) (BCD Multiply)

چهار رقم داده های ثابت و یا محتوای یک Word از حافظه را با فرمت BCD ضرب می نماید.

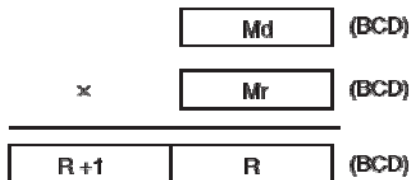


5-5-1- مشخصات عملگرهای دستور *B(424)

فضای حافظه	Md	Mr	R
CIO Area	CIO 0000 to CIO 6143		CIO 0000 to CIO 6142
Work Area	W000 to W511		W000 to W510
Holding Bit Area	H000 to H511		H000 to H510
Auxiliary Bit Area	A000 to A959		A448 to A958
Timer Area	T0000 to T4095		T0000 to T4094
Counter Area	C0000 to C4095		C0000 to C4094
DM Area	D00000 to D32767		D00000 to D32766
EM Area without bank	E00000 to E32767		E00000 to E32766
EM Area with bank	En_00000 to En_32767 (n = 0 to C)		En_00000 to En_32766 (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 #9999		---
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		

5-5-2- شرح عملکرد دستور *B(424)

این دستورالعمل داده های BCD را که در Md و Mr وجود دارند ضرب نموده و نتیجه را در R و R+1 می ریزد.

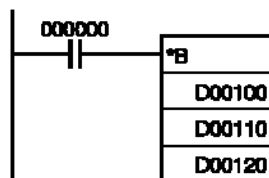


5-5-3- فلگ های مربوط به دستورالعمل *B(424)

نام	نشانه	عملکرد
Error Flag	ER	زمانی که محتوای Md یا Mr مقدار BCD نباشد این فلگ روشن خواهد شد. در بقیه موارد خاموش می باشد.
Equal Flag	=	روشن زمانی که نتیجه ضرب 0 باشد در بقیه موارد خاموش.

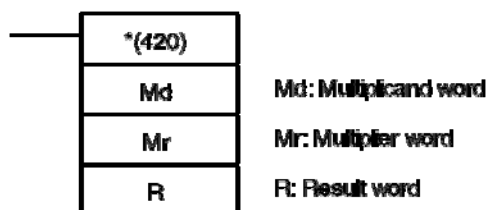
5-5-4- مثال از عملکرد دستور *B(424)

در مثال زیر هنگامی که بیت CIO0.00 روشن می گردد داده های D100 و D110 به صورت BCD در هم ضرب شده و نتیجه در D120 و D121 ریخته می شود.



5-6- ضرب باینری بدون Carry (* Binary Multiply)

چهار رقم داده های ثابت و یا محتوای یک Word از حافظه را با فرمت هگزادسیمال ضرب می نماید.

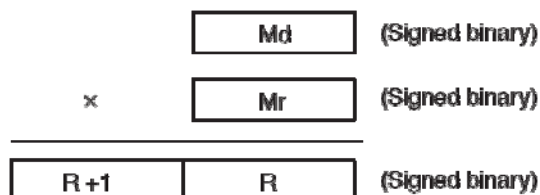


5-6-1- مشخصات عملگرهای دستور *(420)

فضای حافظه	Md	Mr	R
CIO Area	CIO 0000 to CIO 6143		CIO 0000 to CIO 6142
Work Area	W000 to W511		W000 to W510
Holding Bit Area	H000 to H511		H000 to H510
Auxiliary Bit Area	A000 to A959		A448 to A958
Timer Area	T0000 to T4095		T0000 to T4094
Counter Area	C0000 to C4095		C0000 to C4094
DM Area	D00000 to D32767		D00000 to D32766
EM Area without bank	E00000 to E32767		E00000 to E32766
EM Area with bank	En_00000 to En_32767 (n = 0 to C)		En_00000 to En_32766 (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		---
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		

5-6-2- شرح عملکرد دستور (420)*

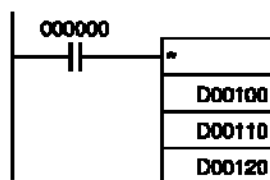
این دستورالعمل داده های باینری را که در M_d و M_r وجود دارند ضرب نموده و نتیجه را در R و R+1 می ریزد.


5-6-3- فلگ های مربوط به دستورالعمل (420)*

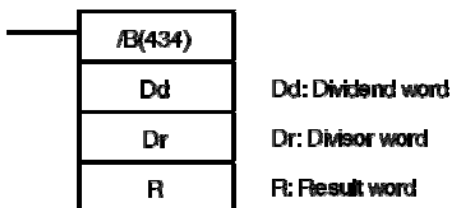
نام	نشانه	عملکرد
Error Flag	ER	خاموش
Equal Flag	=	روشن زمانی که نتیجه ضرب 0 باشد در بقیه موارد خاموش.
Negative Flag	N	زمانی که بیت سمت چپ نتیجه یک باشد این فلگ روشن می شود. در بقیه موارد خاموش می باشد.

5-6-4- مثال از عملکرد دستور (420)*

در مثال زیر هنگامی که بیت CIO0.00 روشن می گردد داده های D100 و D110 به صورت باینری در هم ضرب شده و نتیجه در D120 و D121 ریخته می شود.


5-7- تقسیم BCD بدون Carry (BCD Divide,/B)

چهار رقم داده های ثابت و یا محتوای یک Word از حافظه را با فرمت BCD تقسیم می نماید.

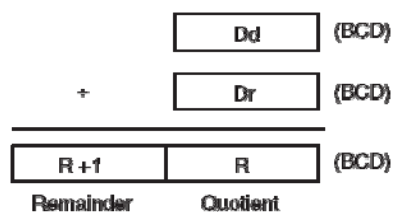


5-7-1- مشخصات عملگرهای دستور (B(434)

فضای حافظه	Dd	Dr	R
CIO Area	CIO 0000 to CIO 6143		CIO 0000 to CIO 6142
Work Area	W000 to W511		W000 to W510
Holding Bit Area	H000 to H511		H000 to H510
Auxiliary Bit Area	A000 to A959		A448 to A958
Timer Area	T0000 to T4095		T0000 to T4094
Counter Area	C0000 to C4095		C0000 to C4094
DM Area	D00000 to D32767		D00000 to D32766
EM Area without bank	E00000 to E32767		E00000 to E32766
EM Area with bank	En_00000 to En_32767 (n = 0 to C)		En_00000 to En_32766 (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 #9999	#0001 to #9999	---
	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		

5-7-2- شرح عملکرد دستور (B(434)

این دستورالعمل داده های BCD در Dd را بر Dr تقسیم کرده و خارج قسمت را در R و نتیجه را در R+1 می ریزد.


5-7-3- فلگ های مربوط به دستورالعمل (B(434)

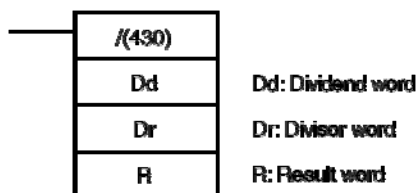
نام	نشانه	عملکرد
Error Flag	ER	زمانی که محتوای Dd یا Dr مقدار BCD نباشد این فلگ روشن خواهد شد. زمانی که باقیمانده برابر 0 شود. در بقیه موارد خاموش می باشد.
Equal Flag	=	روشن زمانی که نتیجه ضرب 0 باشد در بقیه موارد خاموش.

5-7-4- مثال از عملکرد دستور (B(434)

در مثال زیر هنگامی که بیت CIO0.00 روشن می گردد داده های D100 بر داده موجود در D110 به صورت BCD تقسیم شده و خارج قسمت در D120 و باقی مانده در D121 ریخته می شود.

5-8- تقسیم باینری بدون Carry (/ Binary Divide)

چهار رقم داده های ثابت و یا محتوای یک Word از حافظه را با فرمت هگزادسیمال تقسیم می نماید.

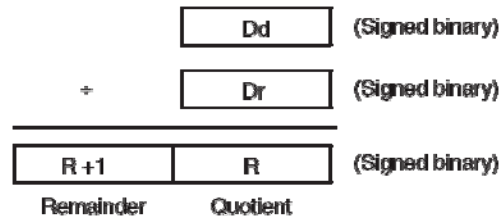


5-8-1- مشخصات عملگرهای دستور (430) /

فضای حافظه	Dd	Dr	R
CIO Area	CIO 0000 to CIO 6143		CIO 0000 to CIO 6142
Work Area	W000 to W511		W000 to W510
Holding Bit Area	H000 to H511		H000 to H510
Auxiliary Bit Area	A000 to A959		A448 to A958
Timer Area	T0000 to T4095		T0000 to T4094
Counter Area	C0000 to C4095		C0000 to C4094
DM Area	D00000 to D32767		D00000 to D32766
EM Area without bank	E00000 to E32767		E00000 to E32766
EM Area with bank	En_00000 to En_32767 (n = 0 to C)		En_00000 to En_32766 (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	#0001 to #FFFF	---
	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		

5-8-2- شرح عملکرد دستور (430) /

این دستورالعمل داده های باینری در Dd را بر Dr تقسیم کرده و خارج قسمت را در R و نتیجه را در R+1 می ریزد.

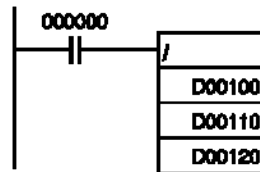


5-8-3- فلگ های مربوط به دستورالعمل (430) /

نام	نشانه	عملکرد
Error Flag	ER	خاموش
Equal Flag	=	روشن زمانی که نتیجه تقسیم 0 باشد در بقیه موارد خاموش.
Negative Flag	N	زمانی که بیت سمت چپ نتیجه یک باشد این فلگ روشن می شود. در بقیه موارد خاموش می باشد.

5-8-4- مثال از عملکرد دستور (430) /

در مثال زیر هنگامی که بیت CIO0.00 روشن می گردد داده های D100 بر داده موجود در D110 به صورت باینری تقسیم شده و خارج قسمت در D120 و باقی مانده در D121 ریخته می شود.



6- دستورات عملی تشخیص عیب

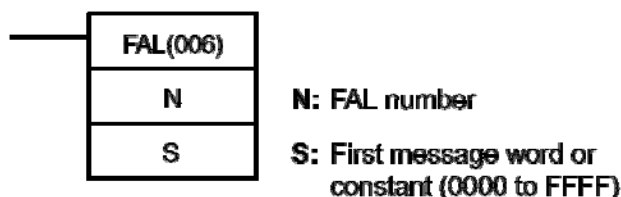
در این بخش به شرح دستورات عملی مورد استفاده برای تعریف و تعیین خطاهای سیستم می پردازیم.

کد دستورالعمل	دستور العمل	نام دستور العمل
006	FAL	هشدار دهنده خطا (Failure Alarm)
008	FALS	هشدار دهنده خطاهای شدید (Severe Failure Alarm)

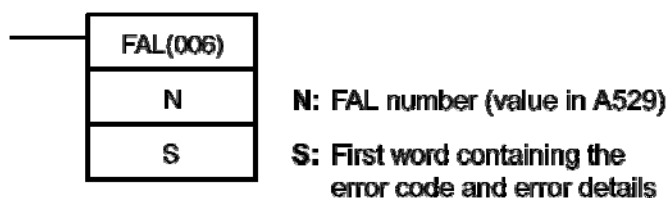
1-6-1- دستورالعمل هشدار دهنده خطا (Failure Alarm-FAL)

این دستورالعمل برای ایجاد و همچنین پاک کردن خطاهای غیر جدی توسط کاربر در برنامه مورد استفاده قرار می گیرد. این خطاها باعث توقف در کار CPU نمی گردند.

- برای تعریف کردن و یا پاک کردن خطاهای غیر جدی توسط کاربر



- جهت فعال کردن خطاهای غیر جدی سیستم. (فقط PLC های مدل CS1-H, CJ1-H, CJ1M و CS1D)



1-6-1- مشخصات عملگرهای دستور FAL(006)

عملگرهای دستورالعمل زمانی که از آن برای تعریف یا پاک نمودن خطاهای تعریف شده توسط کاربر استفاده می شود در مقایسه با حالتی که جهت فعال نمودن خطاهای سیستم به کار می رود اندکی متفاوت می باشد.

تعریف و یا پاک کردن خطاهای غیر جدی توسط کاربر

جدول زیر خصوصیات عملگرهای دستورالعمل را در این حالت نمایش می دهد.

تابع	S	N
خطای غیر جدی با شماره FAL متناظر را پاک می نماید.	#0001 تا #01FF	0
تمام خطاهای غیر جدی را پاک می نماید.	#FFFF	
خطاهای غیر جدی که از اهمیت بیشتری برخوردار می باشند را پاک می نماید.	بقیه مقادیر	
خطای غیر جدی با شماره منظر را تولید می نماید.	#0000 تا #FFFF	1 تا 511
خطایی غیر جدی با شماره متناظر را تولید می نماید. همچنین پیغامی متشکل از 16 کاراکتر با توجه به کدهای اسکی موجود در S تا S+7 تولید می نماید.	آدرس یک Word از حافظه	(شماره دستورالعمل بین FAL و FALS یکسان می باشد).

نکته: مقدار تعیین شده به عنوان عملگر N باید با محتوای A529 (مربوط به شماره FAL/FALS های تولید شده توسط سیستم) متفاوت باشد.

فعال کردن خطاهای غیرجدی سیستم

در جدول زیر عملگرهای دستورالعمل FAL(006) در حالتی که برای فعال کردن خطاهای غیرجدی سیستم در برنامه مورد استفاده قرار می گیرند شرح داده می شود.

عملگر	تابع
N	1 تا 511 (شماره دستورالعمل بین FAL و FALS یکسان می باشد).
S	کد خطایی که باید ایجاد گردد.
S+1	جزئیات کد خطایی که باید ایجاد گردد.

مشخصات کلی عملگرها

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	---	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	0 to 511	#0000 to #FFFF
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

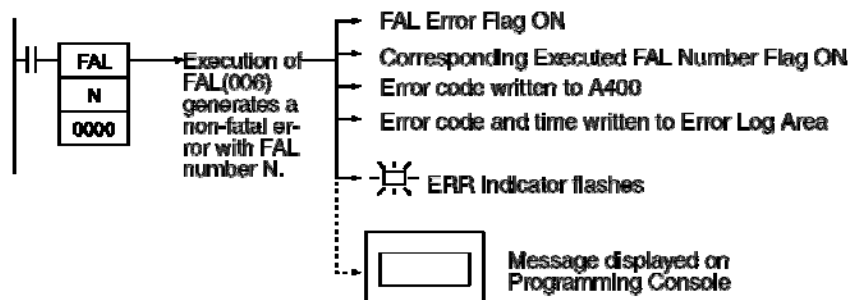
6-1-2- شرح عملکرد دستور FAL(006)

چگونگی عملکرد دستورالعمل FAL(006) به مقدار تعیین شده برای N بستگی دارد. برای پاک کردن خطاهای غیرجدی باید N را 0000 و برای تعریف و ایجاد خطا باید N را عددی بین 0001 تا 01FF تعیین نمود. در صورتی که N و محتوای حافظه A529 یکسان باشند یکی از خطاهای سیستم فعال خواهد شد. (فقط PLC های مدل CS1D و CJ1M، CJ1-H، CS1-H)

تعریف خطاهای غیرجدی توسط کاربر

هنگامی که دستور FAL(006) در حالتی که N عددی بین 1 تا 511 تعیین شده باشد و این عدد برابر با محتوای حافظه A529 نباشد (شماره FAL و یا FALS مربوط به خطاهای سیستم) اجرا می گردد، خطای غیرجدی با شماره FAL متناظر ایجاد خواهد شد و مراحل زیر انجام خواهد شد:

1. فلگ خطای FAL (A402.15) روشن خواهد شد. (عملکرد PLC ادامه پیدا خواهد کرد)
2. فلگ مربوط به شماره FAL برای نشان دادن FAL اجرا شده روشن خواهد شد. فلگهای A360.01 تا A391.15 متناظر با شماره FAL از 1 تا 1FF (1 تا 511) می باشد.
3. کد خطا در حافظه A400 نوشته خواهد شد. کد خطا متناظر با شماره FAL از 0 تا 511، 4101 تا 42FF می باشد.
- اگر یک خطای غیرجدی با اجرای دستور FAL و یک خطای جدی همزمان با هم روی دهند، کد مربوط به خطای جدی در A400 نوشته خواهد شد.
4. کد خطا و همچنین زمانی که خطا روی داده است در ناحیه مربوط به Error Log در حافظه نوشته خواهد شد. (A100 تا A199)
5. نمایشگر خطا در روی CPU شروع به چشمک زدن خواهد کرد.
6. اگر در S آدرس یک Word تعیین شده باشد می توان پیغامی را با استفاده از کد اسکی در تجهیزات برنامه نویسی نمایش داد.



نمایش پیغام خطا در صورت رویداد خطاهای غیرجدی تعریف شده توسط کاربر

زمانی که آدرس یک Word به عنوان S تعیین گردد می توان پیغام خطایی متناسب با خطای روی داده به صورت کد ASCII در S قرار داد، این پیغام خطا زمان اجرای دستورالعمل FAL توسط تجهیزات سریال جانبی متصل به PLC قابل رویت می باشد. (اگر نیازی به نمایش پیغام خطا نباشد می توان S را عددی ثابت تعیین نمود.)

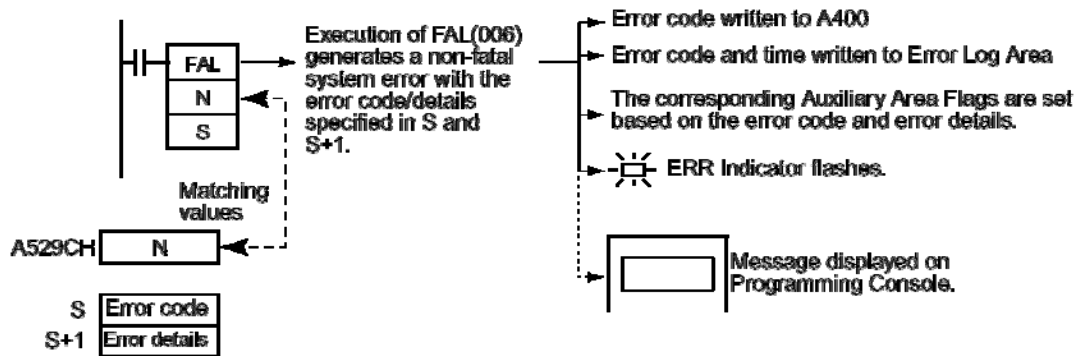
پیغامی که از S آغاز می گردد زمانی که دستورالعمل FAL(006) اجرا شود ثبت می گردد و می توان با وصل نمودن یک تجهیز برنامه نویسی همچون کنسول های برنامه نویسی این پیغام را نمایش داد.

پیغامی با حداکثر 16 کاراکتر را می توان به صورت کدهای ASCII در حافظه های S تا S+7 تعیین نمود. در هر یک از Wordها ابتدا بایت سمت چپ (بازرشته ترین بایت) نمایش داده خواهد شد.

اگر محتوای این حافظه ها بعد از اجرای دستور FAL تغییر نماید متناظر با آن تغییرات پیغام خطا نیز تغییر خواهد کرد.

فعال کردن خطاهای غیر جدی سیستم (فقط PLC های مدل CS1-H، CJ1-H، CJ1M و CS1D)

زمانی که دستورالعمل FAL با شماره FAL مابین 0 تا 511 اجرا گردد و این شماره با محتوای حافظه A529 برابر باشد (شماره FAL های تولید شده توسط سیستم) خطایی غیر جدی توسط سیستم با کد خطا و جزئیات خطای موجود در S و S+1 تولید خواهد شد و همزمان مراحل زیر انجام خواهد شد.



1. کد خطای معین در A400 قرار خواهد گرفت.
2. کد خطا و همچنین زمان رویداد خطا در حافظه مربوط به Error Log نوشته خواهد شد. (A100 تا A199)
3. بسته به کد خطا و همچنین جزئیات خطای روی داده فلگ متناظر در ناحیه AR روشن خواهد شد.
4. نمایشگر مربوط به خطا در روی CPU چشمک زن خواهد شد ولی PLC به عملکرد خود ادامه خواهد داد.
5. پیغام خطای مربوط به خطای رویداده را می توان با اتصال تجهیزات برنامه نویسی به PLC مشاهده نمود.

نکته:

1. خطاهای FAL(006) برای ایجاد خطاهای غیر جدی توسط سیستم در زمان رفع اشکال از برنامه تولید می گردد. به عنوان مثال زمانی که یکی از خطاهای تعریف شده سیستم روی می دهد برای نمایش این خطا توسط تجهیزات جانبی همچون PT ها از این دستور استفاده می گردد.
2. محتوای A529 یک شماره FAL مجازی می باشد (که بین دستورات FAL و FALS مشترک است) که زمانی که خطایی توسط سیستم روی می دهد مورد استفاده قرار می گیرد. محتوای این حافظه شماره FAL مجازی است که حالت فلگ های مربوط به FAL های اجرا شده (A360.01 تا 391.15) و یا کد خطاها را تغییر نمی دهد. زمانی که ایجاد دو یا چند خطای سیستم (خطاهای جدی و یا غیر جدی) به صورت همزمان ضروری می باشد، این خطاها را می توان با استفاده بیش از یکبار از دستورالعملهای FAL/FALS و محتوای ثابت A529 و N ولی مقادیر مختلف S و S+1 ایجاد نمود.

3. در صورتی که یک خطای جدی توسط سیستم و یا خطاهای ایجاد شده توسط دستور FALS همزمان با خطای غیرجدی توسط دستور FAL تولید گردد، کد خطای جدی ایجاد شده در حافظه A400 نوشته خواهد شد.

4. برای پاک کردن خطاهایی که توسط دستور FAL ایجاد می گردد کافی است PLC را خاموش و روشن نمود.

در جدول زیر چگونگی تعیین کد خطاها و همچنین جزئیات مربوط به خطاها و چگونگی تعیین آنها در S و S+1 آورده شده است.

Error name	S	S+1
Interrupt Task Error	008B hex	<ul style="list-style-type: none"> • Bit 15 OFF: Interrupt task error Bits 00 to 14: Task number of interrupt task where error occurred. • Bit 15 ON: Interrupt task execution conflicted with Special I/O Unit refreshing Bits 00 to 14: Unit number of Special I/O Unit with refreshing conflict
Basic I/O Error	009A hex	Rack location of Unit where error occurred <ul style="list-style-type: none"> • Bits 08 to 15: Rack number (binary) of Rack where the affected Unit is mounted • Bits 00 to 07: Slot number (binary) of slot where the affected Unit is mounted
PLC Setup Error	009B hex	PLC Setup Error Location
I/O Table Verification Error	00E7 hex	— (not fixed)
Non-fatal Inner Board Error	02F0 hex	Inner Board Error Information <ul style="list-style-type: none"> • Bits 00 to 03: Invalid • Bits 04 to 15: Error defined by the Inner Board
CS1 CPU Bus Unit Error	0200 hex	CS1 CPU Bus Unit's unit number: 0000 to 000F hex
Special I/O Unit Error	0300 hex	Special I/O Unit's unit number: 0000 to 005F hex or 00FF hex (unit number undetermined)
SYSMAC BUS Error	00A0 hex	SYSMAC BUS Master Unit's unit number: 0000 or 0001 hex
Battery Error	00F7 hex	— (not fixed)
CS1 CPU Bus Unit Setup Error	0400 hex	CS1 CPU Bus Unit's unit number: 0000 to 000F hex
Special I/O Unit Setup Error	0500 hex	Special I/O Unit's unit number: 0000 to 005F hex

پاک کردن خطاهای غیرجدی بدون استفاده از تجهیزات برنامه نویسی

- پاک کردن خطاهای غیرجدی تعریف شده توسط کاربر
زمانی که دستورالعمل (006) FAL با شماره 0 اجرا می گردد، خطاهای غیرجدی پاک می شود.
آنچنانکه در جدول زیر مشخص شده است مقدار S تعیین کننده چگونگی عملکرد این دستورالعمل می باشد.

S	Process
&1 to &511 (0001 to 01FF hex)	The FAL error of the specified number will be cleared.
FFFF hex	All non-fatal errors (including system errors) will be cleared.
0200 to FFFE hex or word specification	The most serious non-fatal error (even if it is a non-fatal system error) that has occurred. When more than one FAL error has occurred, the FAL error with the smallest FAL number will be cleared.

- پاک کردن خطاهای غیرجدی تولید شده توسط سیستم
دو روش برای پاک کردن خطاهای غیرجدی که توسط سیستم ایجاد می گردد وجود دارد :
 - روشن و خاموش نمودن PLC
 - با روشن نگه داشتن PLC این خطای سیستم زمانی که خطای دیگری روی دهد پاک می شود.

3-1-6- فلگ های مربوط به دستورالعمل (006) FAL

نام	نشانه	عملکرد
Error Flag	ER	زمانی که مقدار N در محدوده مجاز تعیین شده (0 تا 0511 دسیمال) نباشد روشن می شود. زمانی که خطایی غیرجدی توسط سیستم ایجاد گردد روشن می شود. در بقیه موارد خاموش می باشد.

در جدول زیر Word ها و فلگ های مرتبط در منطقه AR نشان داده شده است.

- Word ها و فلگ های منطقه AR برای خطاهای تعریف شده توسط کاربر

نام	آدرس	عملکرد
FAL Error Flag	A402.15	زمانی که خطایی توسط دستور (006) FAL ایجاد می گردد روشن می شود.
فلگهای شماره FAL اجرا شده	A360.01 تا A391.15	زمانی که خطایی توسط دستور (006) FAL ایجاد می گردد، فلگ متناظر با آن خطا روشن می گردد. این فلگ ها متناظر با دستورات FAL با شماره 1 تا 1FF می باشد.

- Word ها و فلگ های منطقه AR برای خطاهای ایجاد شده توسط سیستم

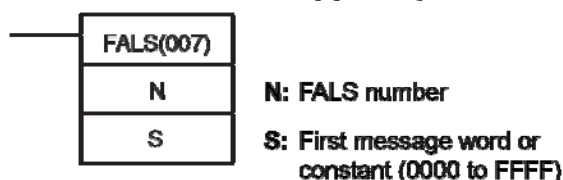
نام	آدرس	عملکرد
شماره FAL/FALS های تولید شده توسط سیستم	A529	زمانی که خطای سیستم توسط دستور (006) FAL روی می دهد یک شماره FAL/FALS مجازی مورد استفاده قرار می گیرد. شماره FAL/FALS مجازی را باید در این حافظه قرار داد.

- Word ها و فلگ های منطقه AR برای خطاهای ایجاد شده توسط سیستم و خطاهای تعریف شده توسط کاربر

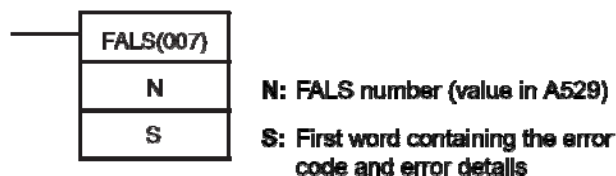
نام	آدرس	عملکرد
Error Log ناحیه	A100 تا A199	ناحیه Error Log شامل کد خطای روی داده تاریخ و زمان رویداد 20 خطای آخر می باشد این ناحیه شامل خطاهای رویداده توسط دستور FAL نیز می باشد.
کد خطا	A400	زمانی که خطایی روی می دهد کد این خطا در حافظه A400 ذخیره می گردد. کد خطا برای شماره FAL بین 0001 تا 01FF عددی مابین 4101 تا 42FF می باشد. اگر دو یا چند خطا به طور همزمان روی دهد کد خطایی که از اهمیت بیشتری برخوردار است در A400 نوشته می شود.

6-2- هشدار دهنده خطاهای شدید (Severe Failure Alarm-FALS)

- این دستورالعمل به کاربر اجازه می دهد که خطاهای جدی و شدید را تعریف نماید. این خطاها باعث توقف عملکرد PLC می شوند. در مورد PLC های مدل CS1-H, CJ1-H, CJ1-M و CS1-D می توان از دستور FALS برای ایجاد خطاهای سیستم نیز استفاده نمود.
- برای تعریف کردن خطاهای جدی توسط کاربر



- جهت فعال کردن خطاهای جدی سیستم. (فقط PLC های مدل CS1-H, CJ1-H, CJ1M و CS1D)



6-2-1- مشخصات عملگرهای دستور FALS(007)

ایجاد خطاهای جدی توسط کاربر

جدول زیر خصوصیات عملگرهای دستورالعمل را در این حالت نمایش می دهد.

عملگر	تابع
N	1 تا 511 (شماره دستورالعمل بین FAL و FALS یکسان می باشد).
S	تعیین کننده اولین آدرس هشت Word مربوط به نمایش پیغام خطا توسط کدهای ASCII که قابل نمایش روی تجهیزات جانبی می باشد. در صورتی که نیازی به پیغام خطا نباشد می تواند عددی بین 0000 تا FFFF تعیین گردد.

نکته: مقدار تعیین شده به عنوان عملگر N باید با محتوای A529 (مربوط به شماره FAL/FALS های تولید شده توسط سیستم) متفاوت باشد.

ایجاد خطاهای جدی سیستم

در جدول زیر عملگرهای دستورالعمل (007) FALS در حالتی که برای فعال کردن خطاهای جدی سیستم در برنامه مورد استفاده قرار می گیرند شرح داده می شود.

تابع	عملگر
1 تا 511 (شماره دستورالعمل بین FAL و FALS یکسان می باشد).	N
کد خطایی که باید ایجاد گردد.	S
جزئیات کد خطایی که باید ایجاد گردد.	S+1

مشخصات کلی عملگرها

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	---	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	تنها مقادیر مشخص	#0000 to #FFFF
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

6-2-2- شرح عملکرد دستور (007) FALS

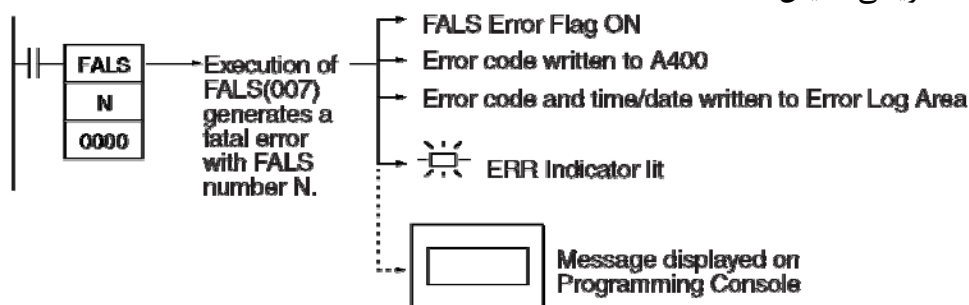
دستور (007) FALS خطایی جدی را ایجاد می نماید. در PLC های مدل CS1-H, CJ1-H, CJ1M و CS1D این دستور برای ایجاد خطاهای جدی سیستم نیز به کار برده می شود. (خطای سیستم زمانی که مقدار عملگر N با مقدار A529 برابر باشد ایجاد می گردد).

تعریف خطاهای جدی توسط کاربر

هنگامی که دستور (006) FAL در حالتی که N عددی بین 1 تا 511 تعیین شده باشد و این عدد برابر با محتوای حافظه A529 نباشد (شماره FAL و یا FALS مربوط به خطاهای سیستم) اجرا می گردد، خطای جدی با شماره FALS متناظر ایجاد خواهد شد و مراحل زیر انجام خواهد شد:

1. فلگ خطای FALS (A401.06) روشن خواهد شد. (عملکرد PLC متوقف خواهد شد)

2. کد خطا در حافظه A400 نوشته خواهد شد. کد خطا متناظر با شماره FALS از 1 تا 511، C101 تا C2FF می باشد.
- اگر یک خطای جدی با اهمیت بیشتر و کد خطای بزرگتر با خطاهای دیگر همزمان با هم روی دهند، کد مربوط به خطای با کد خطای بزرگتر در A400 نوشته خواهد شد.
3. کد خطا و همچنین زمانی که خطا روی داده است در ناحیه مربوط به Error Log در حافظه نوشته خواهد شد. (A100 تا A199)
4. نمایشگر خطا در روی CPU روشن خواهد شد.
5. اگر در S آدرس یک Word تعیین شده باشد می توان پیغامی را با استفاده از کد اسکی در تجهیزات برنامه نویسی نمایش داد.

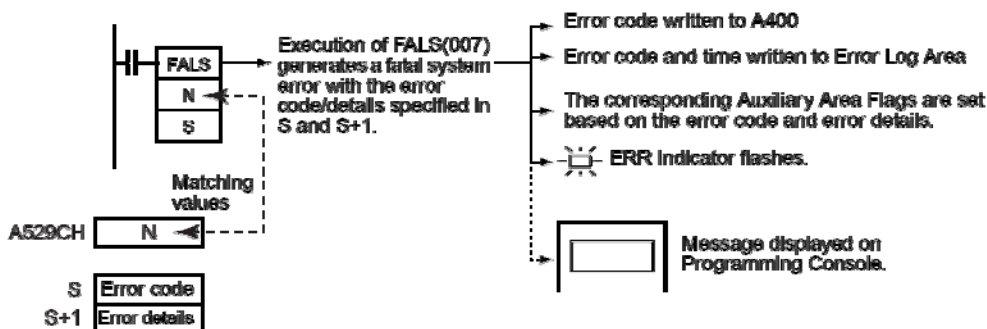


نمایش پیغام خطا در صورت رویداد خطاهای جدی تعریف شده توسط کاربر

زمانی که آدرس یک Word به عنوان S تعیین گردد می توان پیغام خطایی متناسب با خطای روی داده به صورت کد ASCII در S قرار داد، این پیغام خطا زمان اجرای دستورالعمل FALS توسط تجهیزات سریال جانبی متصل به PLC قابل رویت می باشد. (اگر نیازی به نمایش پیغام خطا نباشد می توان S را عددی ثابت تعیین نمود.)

پیغامی که از S آغاز می گردد زمانی که دستورالعمل FALS(007) اجرا شود ثبت می گردد و می توان با وصل نمودن یک تجهیز برنامه نویسی همچون کنسول های برنامه نویسی این پیغام را نمایش داد. پیغامی با حداکثر 16 کاراکتر را می توان به صورت کدهای ASCII در حافظه های S تا S+7 تعیین نمود. در هر یک از Wordها ابتدا بایت سمت چپ (بازرشته ترین بایت) نمایش داده خواهد شد. اگر محتوای این حافظه ها بعد از اجرای دستور FALS تغییر نماید متناظر با آن تغییرات پیغام خطا نیز تغییر خواهد کرد.

فعال کردن خطاهای جدی سیستم (فقط PLC های مدل CS1-H، CJ1-H، CJ1M و CS1D)



زمانی که دستورالعمل FALS با شماره FALS مابین 1 تا 511 اجرا گردد و این شماره با محتوای حافظه A529 برابر باشد (شماره FAL/FALS های تولید شده توسط سیستم) خطایی جدی توسط سیستم با کد خطا و جزئیات خطای موجود در S و S+1 تولید خواهد شد و همزمان مراحل زیر انجام خواهد شد.

1. کد خطای ایجاد شده در A400 قرار خواهد گرفت.
2. کد خطا و همچنین زمان رویداد خطا در حافظه مربوط به Error Log نوشته خواهد شد. (A100 تا A199)
3. بسته به کد خطا و همچنین جزئیات خطای روی داده فلگ متناظر در ناحیه AR روشن خواهد شد.
4. نمایشگر مربوط به خطا در روی CPU روشن خواهد شد و عملکرد PLC متوقف خواهد شد.
5. پیغام خطای مربوط به خطای رویداده را می توان با اتصال تجهیزات برنامه نویسی به PLC مشاهده نمود.

نکته:

1. محتوای A529 یک شماره FAL مجازی می باشد (که بین دستورات FAL و FALS مشترک است) که زمانی که خطایی توسط سیستم روی می دهد مورد استفاده قرار می گیرد. محتوای این حافظه شماره FAL مجازی است که تأثیری بر کد خطاهای جدی نخواهد داشت. زمانی که ایجاد دو یا چند خطای سیستم (خطاهای جدی و یا غیرجدی) به صورت همزمان ضروری می باشد، این خطاها را می توان با استفاده بیش از یکبار از دستورالعملهای FAL/FALS و محتوای ثابت A529 و N ولی مقادیر مختلف S و S+1 ایجاد نمود.
2. در صورتی که دو یا چند خطای جدی توسط سیستم و یا خطاهای ایجاد شده توسط دستور FALS همزمان با هم روی دهند، کد خطای جدی با اهمیت بیشتر در حافظه A400 نوشته خواهد شد.
3. برای پاک کردن خطاهایی که توسط دستور FALS ایجاد می گردد کافی است PLC را خاموش و روشن نمود. همچنین می توان PLC را روشن نگاه داشته ولی عملکرد خاصی را باید برای پاک نمودن خطا انجام داد.
4. جدول زیر تأثیر IOM Hold Bit را بر حالت بیتهای حافظه و همچنین خروجی های PLC زمانی که خطایی جدی سیستم با استفاده از دستور FALS(007) روی می دهد، نشان می دهد.

IOM Hold Bit (A50012)	Status of I/O memory	Status of outputs on Output Units
ON	Retained	OFF
OFF	Cleared	OFF

در جدول زیر چگونگی تعیین کد خطاها و همچنین جزئیات مربوط به خطاها و چگونگی تعیین آنها در S و S+1 آورده شده است.

Error name	S	S+1
	Error code	Error details
Memory Error	80F1 hex	<ul style="list-style-type: none"> Bits 00 to 09: Memory Error Location <ul style="list-style-type: none"> Bit 00: User program Bit 04: PLC Setup Bit 05: Registered I/O table Bit 07: Routing table Bit 08: CPU Bus Unit Setup Bit 09: Memory Card transfer error Bits 10 to 15: Invalid
I/O Bus Error	80C0 hex	<ul style="list-style-type: none"> Bits 00 to 07: Slot number where the I/O Bus error occurred <ul style="list-style-type: none"> Slot 0 to 9: 00 to 09 hex Slot unknown: 0F hex Bits 08 to 15: Rack number where the I/O Bus error occurred <ul style="list-style-type: none"> Slot 0 to 7: 00 to 07 hex Rack unknown: 0F hex
Unit Number Duplication Error	80E9 hex	CPU Bus Unit's duplicated unit number 0000 to 000F hex
		Special I/O Unit's duplicated unit number 8000 to 805F hex
Rack Number Duplication Error	80EA hex	Duplicated Rack number (overlapping word allocations) 0000 to 0006 hex
Fatal Inner Board Error	82F0 hex	Error Cause Bits 00 to 03: Error defined by Inner Board Bits 04 to 15: Invalid



Error name	S	S+1
	Error code	Error details
Too Many I/O Points Error	80E1 hex	Bits 13 to 15: Error Cause Bits 00 to 12: Details <ul style="list-style-type: none"> • Total number of I/O points is too high. Bits 13 to 15: 000 Bits 00 to 12: Number of I/O points (binary) • Number of interrupt inputs is too high. Bits 13 to 15: 001 Bits 00 to 12: Number of Interrupt Inputs (binary) Bits 00 to 12: All zeroes • A Slave Unit's unit number is duplicated or a C500 Slave Unit has more than 320 I/O points. Bits 13 to 15: 010 Bits 00 to 12: Slave Unit's unit number (binary) • The unit number of an I/O Interface (excluding Slave Racks) is duplicated. Bits 13 to 15: 011 Bits 00 to 12: Unit number (binary) • A Master Unit's unit number is duplicated or outside of the allowed setting range. Bits 13 to 15: 100 Bits 00 to 12: Master Unit's unit number (binary) • The number of Expansion Racks is too high. Bits 13 to 15: 101 Bits 00 to 12: Number of Expansion Racks (binary) • C200H Special I/O Unit or Remote I/O was not recognized. Bits 13 to 15: 110
I/O Table Setting Error	80E0 hex	— (Not fixed.)
Program Error	80F0 hex	<ul style="list-style-type: none"> • Bits 08 to 15: Error Cause Bit 15: UM overflow error Bit 14: Illegal instruction error Bit 13: Differentiation overflow error Bit 12: Task error Bit 11: No END error Bit 10: Illegal access error Bit 09: Indirect DM/EM BCD error Bit 08: Instruction error • Bits 00 to 07: Invalid
Cycle Time Overrun Error	809F hex	— (Not fixed.)

پاک کردن خطاهای جدی (FALS(007) سیستم

پاک کردن خطاهای جدی تولید شده توسط سیستم

دو روش برای پاک کردن خطاهای جدی که توسط سیستم و دستور (FALS(007) ایجاد می گردد وجود دارد:

- روشن و خاموش نمودن PLC
- با روشن نگه داشتن PLC این خطای سیستم زمانی که خطای دیگری روی دهد پاک می شود.

پاک کردن خطاهای جدی تعریف شده توسط کاربر

برای از بین بردن خطاهای جدی ایجاد شده با استفاده از دستور FALS توسط کاربر ابتدا باید علت ایجاد خطا را از بین برده و سپس با خاموش و روشن کردن PLC و یا با استفاده از یک تجهیز جانبی برنامه نویسی این خطا را پاک نمود.

6-2-3- فلگ های مربوط به دستورالعمل FALS(007)

نام	نشانه	عملکرد
Error Flag	ER	زمانی که مقدار N در محدوده مجاز تعیین شده (1 تا 0511 دسیمال) نباشد روشن می شود. زمانی که خطایی جدی توسط سیستم ایجاد گردد ولی کد خطا و یا جزئیات خطا صحیح نباشد این فلگ روشن می شود. در بقیه موارد خاموش می باشد.

در جدول زیر Word ها و فلگ های مرتبط در منطقه AR نشان داده شده است.

- Word ها و فلگ های منطقه AR برای خطاهای تعریف شده توسط کاربر

نام	آدرس	عملکرد
FALS Error Flag	A401.06	زمانی که خطایی توسط دستور FALS(007) ایجاد می گردد روشن می شود.

- Word ها و فلگ های منطقه AR برای خطاهای ایجاد شده توسط سیستم

نام	آدرس	عملکرد
شماره FAL/FALS های تولید شده توسط سیستم	A529	زمانی که خطای سیستم توسط دستور FALS(007) روی می دهد یک شماره FAL/FALS مجازی مورد استفاده قرار می گیرد. شماره FAL/FALS مجازی را باید در این حافظه قرار داد.

- Word ها و فلگ های منطقه AR برای خطاهای ایجاد شده توسط سیستم و خطاهای تعریف شده توسط کاربر

نام	آدرس	عملکرد
ناحیه Error Log	A100 تا A199	ناحیه Error Log شامل کد خطای روی داده تاریخ و زمان رویداد 20 خطای آخر می باشد این ناحیه شامل خطاهای رویداده توسط دستور FALS(007) نیز می باشد.
کد خطا	A400	زمانی که خطایی روی می دهد کد این خطا در حافظه A400 ذخیره می گردد. کد خطا برای شماره FAL بین 0001 تا 01FF عددی مابین C101 تا C2FF می باشد. اگر دو یا چند خطا به طور همزمان روی دهد کد خطایی که از اهمیت بیشتری برخوردار است در A400 نوشته می شود.

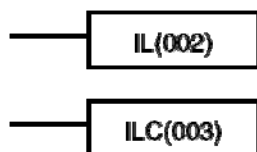
7- دستورات کنترل ترتیب

در این بخش به توضیح دستورالعملهای که برای کنترل ترتیب در برنامه به کار می روند می پردازیم.

کد دستورالعمل	دستورالعمل	نام دستورالعمل
002	IL	Interlock
003	ILC	Interlock Clear
004	JMP	Jump
005	JME	Jump End

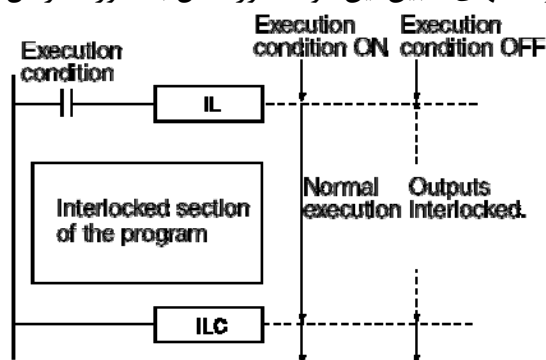
7-1-1- دستورالعملهای Interlock و Interlock Clear (IL و ILC)

این دستورالعمل زمانی که شرط عملکرد دستور IL(002) خاموش باشد همه خروجی های مابین دو دستورالعمل IL(002) و ILC(003) را قفل می نماید. این دو دستورالعمل معمولاً به صورت دوتایی و با هم مورد استفاده قرار می گیرند.



7-1-1-1- شرح عملکرد دستور IL(002) و ILC(003)

زمانی که شرط عملکرد ورودی دستورالعمل IL(002) خاموش باشد تمام خروجی های مربوط به تمام دستورالعملهای مابین IL(002) تا ILC(003) قفل خواهند شد. زمانی که شرط عملکرد ورودی دستورالعمل IL(002) خاموش باشد دستورالعملهای مابین این دو دستورالعمل به صورت نرمال عمل خواهند نمود.



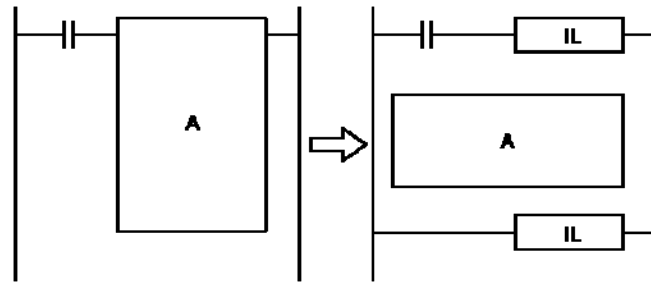
جدول زیر چگونگی عملکرد خروجی های مختلف را که زمانی که مابین دو دستور IL(002) و ILC(003) قرار می گیرند نشان می دهند.

عملکرد خروجی	دستورالعمل	
خاموش	بیت های تعیین شده بوسیله OUT، OUTN، OUT یا OUTB(534)	
خاموش (Reset)	فلگ اتمام کار	TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540), TMHHX(552), TIML(542), and TIMXL(553)
به مقدار SV بازگردانده می شود. (Reset)	PV	
به حالت پیشین خود برگردانده می شوند.	بیت ها و Word های مشخص شده بوسیله بقیه دستورالعملها	

نکته: بیت ها و Word های به کار رفته توسط سایر دستورالعملها مانند دستورالعملهای TTIM(087), TTIMX(555), MTIM(543), MTIMX(554), SET, RSET, CNT, CNTX(546), CNTR(012), CNTRX(548), SFT, KEEP(011) در حالت قبلی خود باقی می مانند.

اگر بیتی در قسمت قفل شده از برنامه توسط دستورالعمل IL(002) وجود دارد که باید در حالت اینترلاک روشن بماند می توان این بیت را درست قبل از IL(002) با استفاده از دستورالعمل SET روشن نمود.

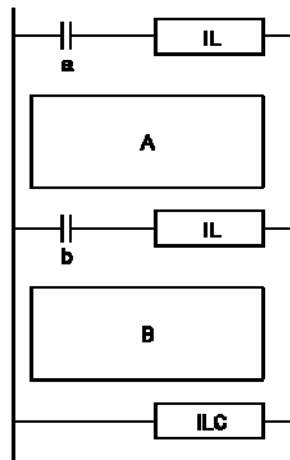
استفاده از این دستورالعملها بسیار مؤثر می باشد، زمانی که چندین فرآیند مختلف با استفاده از یک شرط عملکرد انجام می گیرند، با استفاده از IL(002) و ILC(003) حجم کمتری از حافظه برنامه نویسی اشغال می گردد.



7-1-2- نکات قابل توجه در مورد دستورالعمل IL(002) و ILC(003)

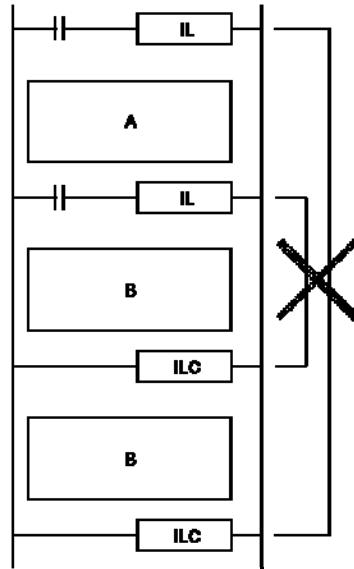
زمانی که یک بخش از برنامه با استفاده از این دستورالعملها اینترلاک می گردد، این عمل در زمان دوره برنامه هیچ تأثیری نخواهد داشت، به این دلیل که برنامه های موجود در این بخش به صورت داخلی اجرا می گردند. نحوه عملکرد دستورالعملهای DIFU(013) و DIFD(014) و همچنین دستورالعملهای که در آن از لبه استفاده شده است زمانی که در بخشی از برنامه که بین دو دستورالعمل IL(002) و ILC(003) قرار می گیرند، منحصرأ به حالت شرط عملکرد دستورالعمل وابسته نمی باشند. تغییرات در شرط عملکرد دستوراتی همچون DIFU(013) و DIFD(014) و همچنین دستورالعملهای که در آن از لبه استفاده شده است زمانی که این دستورات در بخشی از برنامه که توسط دستورات Interlock قفل شده است نوشته شده اند و شرط عملکرد دستور IL(002) خاموش باشد ثبت نمی گردد.

به طور کلی دو دستورالعمل IL(002) و ILC(003) با هم مورد استفاده قرار می گیرند، ولی می توان از چند دستور IL(002) به صورت همزمان با یک دستور ILC(003) همچنان که در نمودار زیر دیده می شود، استفاده نمود. اگر این دو دستورالعمل با هم مورد استفاده قرار نگیرند، در هنگام بررسی برنامه خطایی توسط سیستم ایجاد می گردد که البته این خطا باعث ایست عملکرد و اجرای برنامه نخواهد شد.



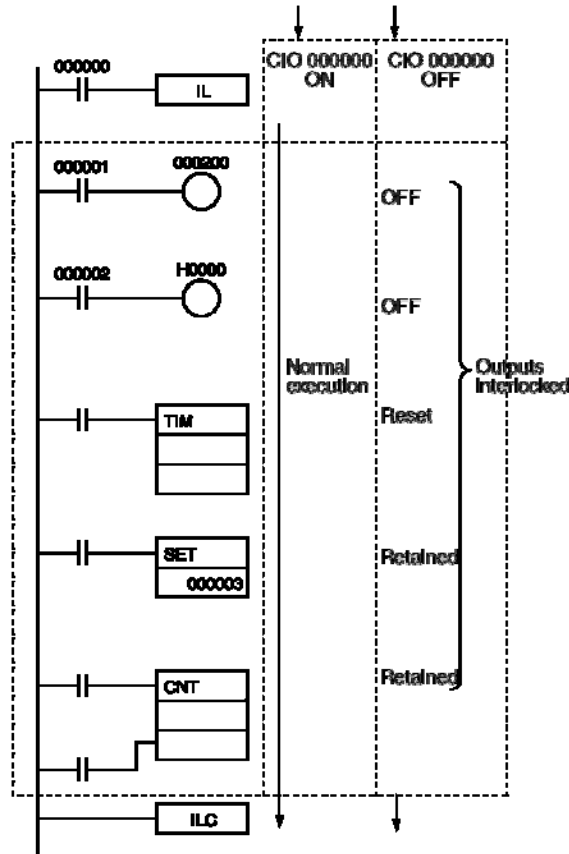
Execution condition		Program section	
a	b	A	B
OFF	ON	Interlocked	Interlocked
OFF	OFF	Interlocked	Interlocked
ON	OFF	Not interlocked	Interlocked
ON	ON	Not interlocked	Not interlocked

همچنان که در ادامه دیده می شود دستور IL(002) و ILC(003) را نمی توان به صورت تو در تو مورد استفاده قرار داد.



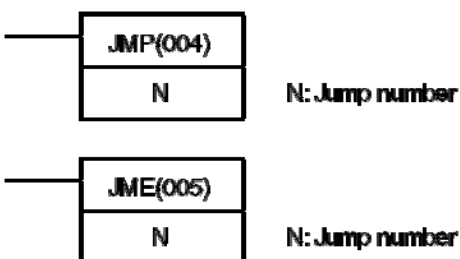
7-1-3- مثال از عملکرد دستور IL(002) و ILC(003)

در این مثال زمانی که CIO0.00 خاموش باشد، تمام خروجی ها بین دو دستورالعمل IL(002) و ILC(003) قفل خواهند شد. ولی زمانی که CIO0.00 روشن باشد دستورالعمل به صورت معمول اجرا می گردد.



2-7-2- دستورالعملهای پرش و انتهای پرش (Jump and Jump End-JMP,JME)

در این دستورالعمل زمانی که شرط عملکرد دستور خاموش باشد برنامه از محلی که دستور JMP(004) نوشته شده است به اولین دستور JME(005) با همان شماره پرش خواهد کرد. دستورالعملهای JMP(004) و JME(005) باهم مورد استفاده قرار می گیرند.



2-7-1- مشخصات عملگرهای دستور JMP(004) و JME(005)

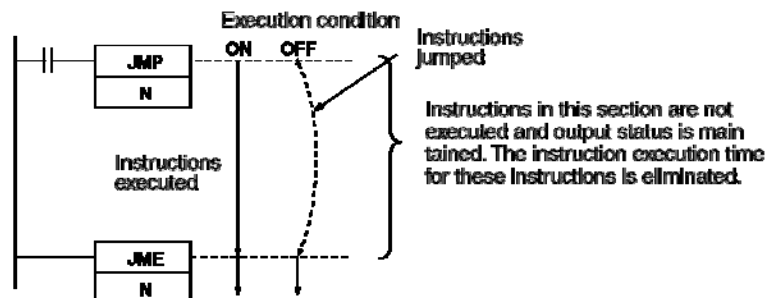
شماره JUMP را باید عددی مابین 0000 تا 03FF (0 تا 1023 دسیمال) انتخاب نمود.

فضای حافظه	N	
	JMP(004)	JME(005)
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 #03FF(Binary) 0 to 1023 (Decimal)	#0000 to #03FF(Binary) 0 to 1023 (Decimal)
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	---

7-2-2- شرح عملکرد دستور JMP(004) و JME(005)

زمانی که شرط عملکرد دستورالعمل JMP(004) روشن باشد هیچ پرشی در برنامه صورت نخواهد پذیرفت و برنامه به صورت معمول اجرا خواهد شد.

ولی زمانی که شرط عملکرد دستورالعمل خاموش باشد، اجرای برنامه به اولین دستور JME(005) با همان شماره پرش خواهد کرد، در نتیجه دستورالعمل های به کار رفته شده بین دستورهای JMP(004) و JME(005) اجرا نمی گردند و حالت خروجی های مابین این دو دستورالعمل ثابت باقی خواهد ماند.



به دلیل این که تمام دستورالعمل های بین JMP(004) و JME(005) زمانی که شرط عملکرد آن خاموش باشد نادیده گرفته خواهند شد در نتیجه زمان دوره اجرای برنامه به اندازه دستورالعملهایی که اجرا نشده اند کاهش خواهد یافت.

در جدول زیر انواع دستورالعملهای پرش آمده است و همچنین مقایسه این دستورالعملها نیز شرح داده شده اند.

مورد	JMP(004) JME(005)	CJP(510) JME(005)	CJPN(511) JME(005)	JMP0(515) JME0(516)
شرط عملکرد دستور	خاموش	روشن	خاموش	خاموش
تعداد مجاز	مجموعاً 1024			بدون محدودیت
دستورالعملهای پردازش شده در زمان پرش	اجرا نمی شود.			پردازش NOP(000)
زمان اجرای دستورالعملها در زمان پرش	---			همانند دستور NOP(000)
حالت خروجی ها (بیت ها و Word ها)	بیت ها و Word ها حالت پیشین خود را حفظ می نمایند.			
حالت تایمرها در زمان پرش	تایمرها به شمارش ادامه می دهند.			
عملکرد در برنامه	همیشه پرش	پرش در زمان روشن بودن	پرش در زمان خاموش بودن	مجاز نمی باشد.

7-2-3- فلگ های مربوط به دستورالعمل JMP(004) و JME(005)

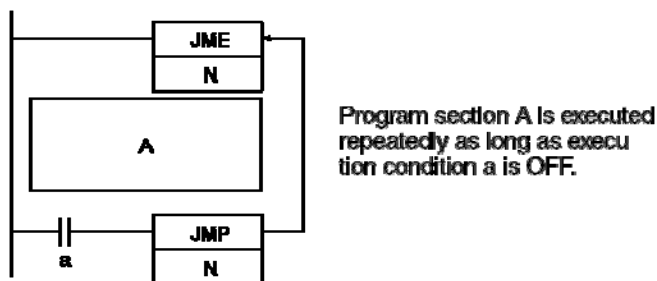
نام	نشانه	عملکرد
Error Flag	ER	زمانی که مقدار N در محدوده مجاز تعیین شده (0 تا 03FF) نباشد روشن می شود. زمانی که در برنامه دستور JMP(005) نوشته شده باشد ولی دستور JME(006) متناظر با آن وجود نداشته باشد. زمانی که در برنامه دستور JMP(005) در یک Task نوشته شده باشد ولی دستور JME(006) متناظر با آن در آن Task وجود نداشته باشد. در بقیه موارد خاموش می باشد.

7-2-4- نکات قابل توجه در مورد دستورالعمل JMP(004) و JME(005)

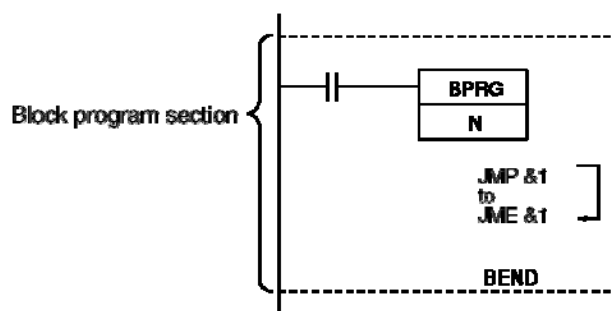
تمام خروجی‌ها در زمان استفاده از دستورالعمل JMP(004) حالت پیشین خود را حفظ می‌نمایند، اما تایمرها به شمارش خود ادامه می‌دهند به دلیل این که مقدار PV تایمرها حتی در صورت اجرا نشدن دستورالعمل نیز تجدید خواهد شد.

زمانی که از یک یا چند دستورالعمل JME(005) با یک شماره استفاده شود تنها دستورالعملی که از آدرس پایین‌تری در برنامه برخوردار باشد در نظر گرفته می‌شود و دستورالعملی که در مکان بالاتری از نظر حافظه برنامه واقع باشند نادیده گرفته می‌شوند.

زمانی که JME(005) جلوتر از دستور JMP(004) در برنامه مورد استفاده قرار گیرد، در بعضی موارد دستورالعمل‌های مابین JME(005) و JMP(004) بارها و بارها تا زمانی که شرط عملکرد دستورالعمل خاموش گردد، اجرا می‌گردند. در صورتی که شرط عملکرد دستورالعمل JMP(004) خاموش نگردد زمان دوره برنامه طولانی شده و خطای "طولانی شدن زمان دوره" ایجاد خواهد شد.



در برنامه‌های بلوکی صرفنظر از حالت شرط عملکرد دستورالعمل JMP(004)، همواره دستورالعمل‌های مابین JMP(004) و JME(005) نادیده گرفته خواهند شد.



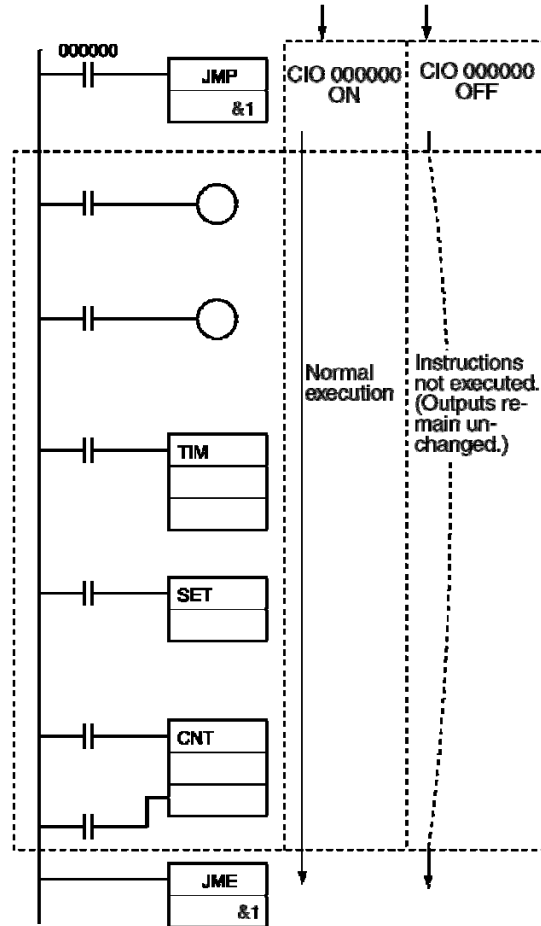
دستور JMP(004) و JME(005) را باید به صورت جفت در یک Task مورد استفاده قرار داد زیرا استفاده از این دستورالعمل‌ها در بین Task‌های مختلف مجاز نمی‌باشد. اگر دستور JME(005) در یک Task مورد استفاده قرار گیرد در صورتی که دستور JMP(004) با همان شماره در آن Task وجود نداشته باشد، خطایی توسط برنامه ایجاد خواهد شد.

عملکرد دستورالعمل‌هایی همچون DIFU(013) و DIFD(014) زمانی که بین دستورالعمل‌های JMP(004) و JME(005) مورد استفاده قرار می‌گیرند منحصر به شرط عملکرد آنها بستگی نخواهد داشت. زمانی که دستورالعمل‌هایی همچون DIFU(013) و DIFD(014) در بخشی از برنامه بعد از دستورالعمل JMP(004) مورد

استفاده قرار می گیرند بدون در نظر گرفتن حالت شرط عملکرد در هنگام پرش از این بخش از برنامه، شرط عملکرد این دستورات تنها قبل از پرش و بعد از پرش مورد مقایسه قرار خواهد گرفت.

7-2-5- مثال از عملکرد دستورات JMP(004) و JME(005)

در مثال زیر زمانی که CIO0.00 خاموش باشد، دستورات عملیاتی موجود مابین JMP(004) و JME(005) اجرا نخواهند شد و تمام خروجی ها حالت نهایی خود را حفظ خواهند نمود. زمانی که CIO0.00 روشن باشد دستور عملیاتی به صورت نرمال اجرا می گردند.



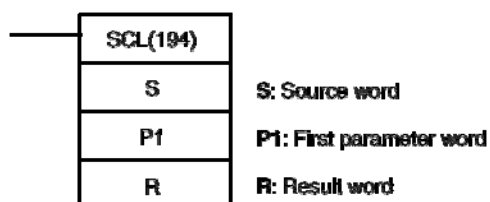
8- دستورات مقیاس کردن

در این بخش به شرح دستورات عملیاتی مربوط به مقیاس نمودن داده ها که جزء دستورات کنترل داده ها می باشند پرداخته می شود.

کد دستورالعمل	دستور العمل	نام دستور العمل
194	SCL	Scaling
486	SCL2	Scaling 2
487	SCL3	Scaling 3

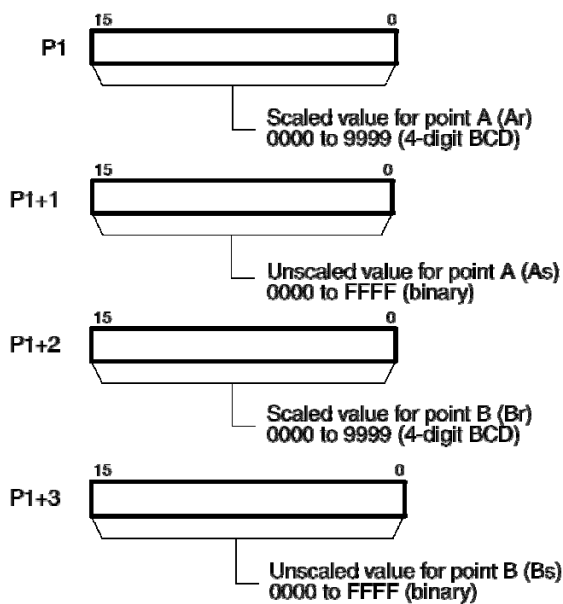
8-1-1- دستورالعمل مقیاس گذاری (Scaling-SCL)

این دستورالعمل با در نظر گرفتن تابع خطی مشخص، داده باینری بدون علامت را به داده BCD بدون علامت تبدیل می نماید.



8-1-1- مشخصات عملگرهای دستور SCL(194)

محتوای چهار Word که از اولین آدرس مشخص شده به عنوان Parameter Word آغاز می گردد به صورت زیر می باشد:



نکته : حافظه های P1 تا P1+3 باید در یک ناحیه باشند.

فضای حافظه	S	P1	R
CIO Area	CIO 0000 to CIO 6143	CIO 0000 to CIO 6140	CIO 0000 to CIO 6143
Work Area	W000 to W511	W000 to W508	W000 to W511
Holding Bit Area	H000 to H511	H000 to H508	H000 to H511
Auxiliary Bit Area	A000 to A959	A000 to A956	A448 to A959
Timer Area	T0000 to T4095	T0000 to T4092	T0000 to T4095
Counter Area	C0000 to C4095	C0000 to C4092	C0000 to C4095
DM Area	D00000 to D32767	D00000 to D32764	D00000 to D32767
EM Area without bank	E00000 to E32767	E00000 to E32764	E00000 to E32767
EM Area with bank	En_00000 to En_32767 (n = 0 to C)	En_00000 to En_32764 (n = 0 to C)	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	---		
	DR0 to DR15	---	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		

8-1-2- شرح عملکرد دستور SCL(194)

دستورالعمل SCL(194) برای تبدیل داده باینری بدون علامت که در Word مرجع (S) وجود دارد به داده BCD به کار می رود. این داده BCD تبدیل شده در Word نتیجه (R) قرار می گیرد و این تبدیل با توجه به تابع خطی با نقاط تعیین شده (As,Ad) و (Bs,Bd) انجام می پذیرد. آدرس اولین Word از حافظه که شامل داده های مربوط به نقاط (As,Ad) و (Bs,Bd) می باشد در P1 معین می گردد. این نقاط با استفاده از دو داده (Bs و As) قبل از مقیاس کردن و همچنین دو داده (Bd و Ad) بعد از مقیاس کردن تعریف می گردند. برای این تبدیل از معادلات زیر استفاده می شود.

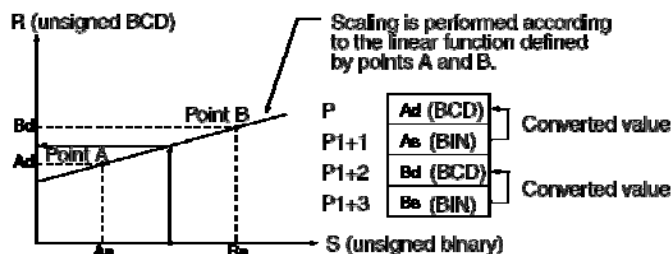
$$R = Bd - \frac{(Bd - Ad)}{BCD \text{ conversion of } (Bs - As)} \times BCD \text{ conversion of } (Bs - S)$$

شیب خط به صورت زیر محاسبه می گردد.

$$R = Bd - \frac{(Bd - Ad)}{BCD \text{ conversion of } (Bs - As)}$$

نقاط A و B برای تعریف یک خط با شیب مثبت و یا منفی به کار برده می شود. استفاده از شیب منفی مقیاس گذاری معکوس را فعال خواهد کرد.

نتیجه به مقدار نزدیکترین عدد صحیح گرد خواهد شد. اگر نتیجه از 0000 کوچکتر شود عدد 0000 به عنوان نتیجه در نظر گرفته می شود. اگر نتیجه بزرگتر از 9999 باشد، عدد 9999 به عنوان نتیجه در نظر گرفته می شود.



از این دستورالعمل می توان برای مقیاس کردن مقدار های تبدیل شده سیگنالهای ورودی آنالوگ با توجه به پارامترهای مورد نظر کاربر استفاده نمود. به عنوان مثال اگر ولتاژ 0 تا 5 ولت به صورت یک سیگنال ورودی آنالوگ به یکی از ورودی های آنالوگ متصل باشد این داده ها در PLC به داده ای باینری بین 0000 تا 0FA0 تبدیل می گردد، این داده ها را می توان با استفاده از دستورالعمل SCL(194) به داده ای BCD مابین 50 تا 200 درجه سانتیگراد تبدیل نمود.

دستورالعمل SCL(194) داده های باینری بدون علامت را به داده BCD بدون علامت تبدیل می نماید. برای تبدیل یک مقدار منفی باید قبل از استفاده از دستورالعمل SCL(194) این حداکثر مقدار منفی را در برنامه اضافه نمود.

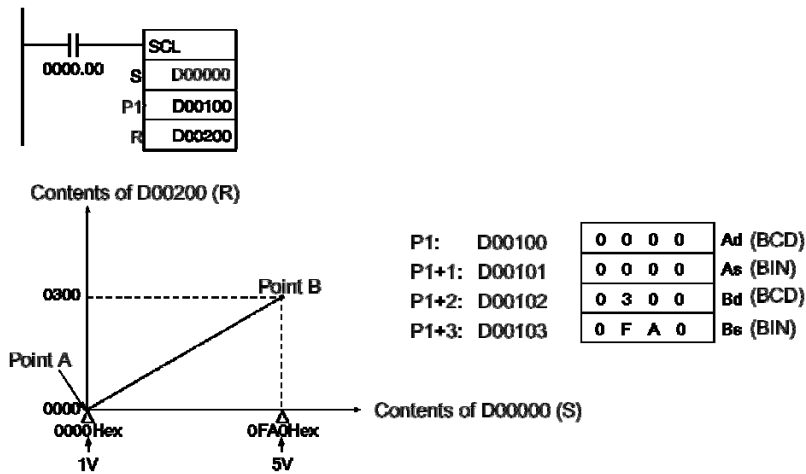
امکان ایجاد نتیجه منفی و انتقال آن به Word نتیجه توسط دستور SCL(194) وجود ندارد. اگر نتیجه مقداری منفی شود عدد 0000 به عنوان نتیجه به در R ریخته می شود.

3-1-8- فلگ های مربوط به دستورالعمل SCL(194)

نام	نشانه	عملکرد
Error Flag	ER	روشن در صورتی که محتوای P1 و P1+2 به فرمت BCD نباشد. روشن در صورتی که محتوای P1+1 و P1+3 برابر باشد. در بقیه موارد خاموش.
Equal Flag	=	روشن زمانی که نتیجه 0 باشد در بقیه موارد خاموش.

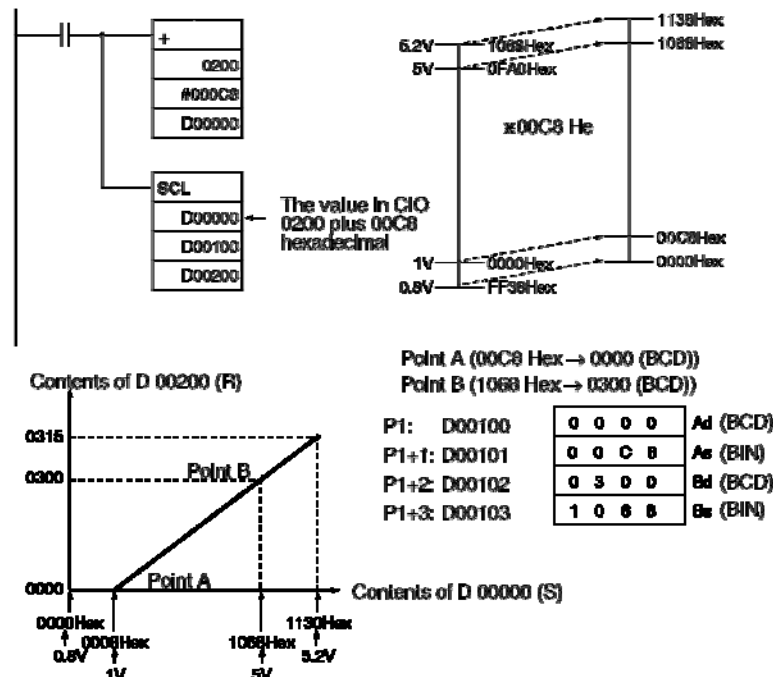
4-1-8- مثال از عملکرد دستور SCL(194)

در این مثال فرض بر این است که سیگنال آنالوگ ورودی 1 تا 5 ولت به داده باینری بین 0000 تا 0FA0 هگزادسیمال تبدیل شده و در حافظه D0 قرار گرفته است. دستور SCL(194) برای تبدیل این داده باینری به داده ای BCD مابین 0000 تا 0300 در حافظه D200 به کار رفته است.



مقادیر منفی

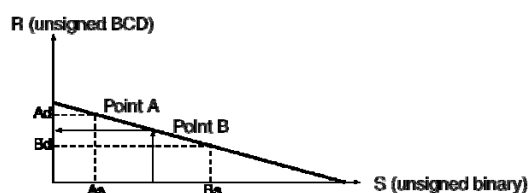
اگر مقدار واقعی سیگنال ورودی آنالوگ 0.8 تا 5.2 ولت باشد، این سیگنال به داده های باینری بین FF38 تا 1068 تبدیل می گردد. از آنجایی که دستورالعمل SCL(194) فقط برای مقیاس کردن داده های باینری مابین 0000 تا FFFF مورد استفاده قرار می گیرد در چنین مواردی استفاده مستقیم از دستورالعمل SCL(194) برای مقیاس کردن داده های علامت دار کمتر از 1V (به عنوان مثال داده های مابین FF38 تا FFFF هگزادسیمال) ممکن نمی باشد. در عمل در این موارد باید عدد 00C8 را به تمام داده ها اضافه نمود در نتیجه این اضافه کردن قبل از مقیاس کردن FF38 به 0000 تبدیل خواهد شد. در مثال زیر چگونگی این امر مشاهده می گردد.



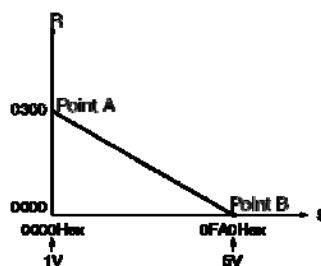
در این مثال مقدار های مابین 0000 تا 00C8 هگزادسیمال به داده های منفی تبدیل خواهد شد. از آنجایی که دستور (SCL194) تنها مقادیر BCD بدون علامت را در حافظه قرار می دهد و مقادیر کمتر از 0000 را به صورت 0000 نمایش می دهد. در این مثال نیز زمانی که محتوای D0 مقادیر مابین 0000 تا 00C8 باشد مقدار نتیجه به صورت 0000 در حافظه مربوطه قرار می گیرد.

مقیاس معکوس

زمانی که $As < Bs$ و $Ar > Br$ باشد، مقیاس معکوس مورد استفاده قرار می گیرد.

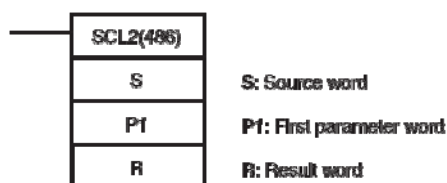


از مقیاس گذاری معکوس برای تبدیل داده به صورت معکوس استفاده می گردد به عنوان مثال زمانی که بخواهیم سیگنال ورودی آنالوگ مابین 1 تا 5V (0000 تا 0FA0 هگزادسیمال) را به 0300 تا 0000 تبدیل نماییم باید از مقیاس گذاری معکوس استفاده نمود.



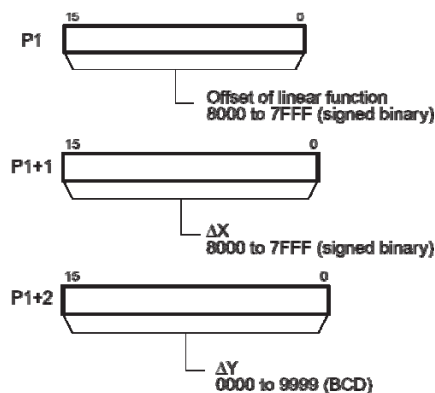
2-8- دستورالعمل مقیاس گذاری 2 (SCL2-Scaling 2)

این دستورالعمل داده های علامت دار باینری را به داده های علامت دار BCD با توجه به تابع خطی تعیین شده، تبدیل می نماید. در تعریف تابع خطی تبدیل می توان آفست نیز در نظر گرفته می شود.



8-2-1- مشخصات عملگرهای دستور SCL2(486)

در زیر محتوای سه Word از حافظه که از اولین پارامتر Word (P1) آغاز می گردد نشان داده شده است.



نکته: حافظه های P1 تا P1+2 باید در یک ناحیه باشند.

فضای حافظه	S	P1	R
CIO Area	CIO 0000 to CIO 6143	CIO 0000 to CIO 6141	CIO 0000 to CIO 6143
Work Area	W000 to W511	W000 to W509	W000 to W511
Holding Bit Area	H000 to H511	H000 to H509	H000 to H511
Auxiliary Bit Area	A000 to A959	A000 to A957	A448 to A959
Timer Area	T0000 to T4095	T0000 to T4093	T0000 to T4095
Counter Area	C0000 to C4095	C0000 to C4093	C0000 to C4095
DM Area	D00000 to D32767	D00000 to D32765	D00000 to D32767
EM Area without bank	E00000 to E32767	E00000 to E32765	E00000 to E32767
EM Area with bank	En_00000 to En_32767 (n = 0 to C)	En_00000 to En_32765 (n = 0 to C)	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	---		
	DR0 to DR15	---	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		

8-2-2- شرح عملکرد دستور SCL2(486)

دستورالعمل SCL2(486) برای تبدیل داده باینری علامت دار موجود در S به داده BCD علامت دار مورد استفاده قرار می گیرد (داده BCD شامل مقدار قدمطلق و همچنین فلگ Carry که نشان دهنده علامت است می باشد) و این نتیجه BCD در حافظه تعیین شده به عنوان نتیجه R قرار می گیرد. مقیاس گذاری با استفاده

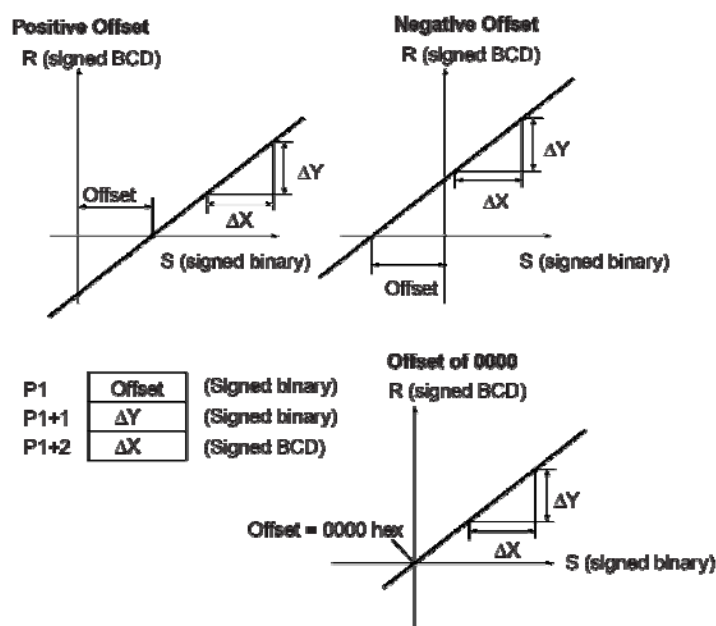
از این دستورالعمل با توجه به تابع خطی معینی که با توجه به شیب $(\Delta X, \Delta Y)$ و همچنین آفست تعیین می گردد انجام می گیرد. مقادیر ΔX ، ΔY و همچنین آفست در حافظه های P1 تا P1+2 تعیین می گردد. علامت نتیجه این مقیاس گذاری با استفاده از فلگ Carry تعیین می گردد (روشن: منفی، خاموش: مثبت). از معادله زیر برای تبدیل استفاده می گردد:

$$R = \frac{\Delta Y}{\text{BCD conversion of } \Delta X} \times ((\text{BCD conversion of } S) - (\text{BCD conversion of offset}))$$

شیب خط $\Delta Y/\Delta X$ می باشد.

آفست و شیب خط می تواند مقداری مثبت، صفر و یا مقداری منفی باشد. استفاده از مقادیر منفی امکان مقیاس گذاری معکوس را فراهم می آورد.

نتیجه به نزدیکترین مقدار صحیح گرد خواهد شد. نتیجه موجود در R مقدار گرد شده نتیجه خواهد بود و علامت توسط فلگ Carry نمایش داده خواهد شد. پس نتیجه عددی BCD مابین 9999- تا 9999 خواهد بود. اگر نتیجه مقداری بیشتر از 9999 باشد مقدار 9999 را به عنوان خروجی خواهیم داشت.



دستورالعمل SCL2(486) را می توان برای مقیاس کردن مقدار تبدیل شده سیگنالهای ورودی آنالوگ با توجه به پارامترهای تعیین شده توسط کاربر به کار برد. به عنوان مثال اگر سیگنال ورودی آنالوگ 1 تا 5 ولت به یکی از ورودی های آنالوگ متصل گردد و به صورتی داده ای باینری بین 0000 تا 0FA0 هگزادسیمال در حافظه نمایش داده شود می توان این داده را با استفاده از دستورالعمل SCL2(486) به داده ای BCD مابین 100- تا 200 تبدیل نمود.

SCL2(486) داده باینری علامت دار به داده BCD علامت دار تبدیل می نماید. در نتیجه مقادیر منفی را نیز می توان مستقیماً در S قرار داد. نتیجه این مقیاس در R و علامت آن با استفاده از Carry تعیین می گردد.

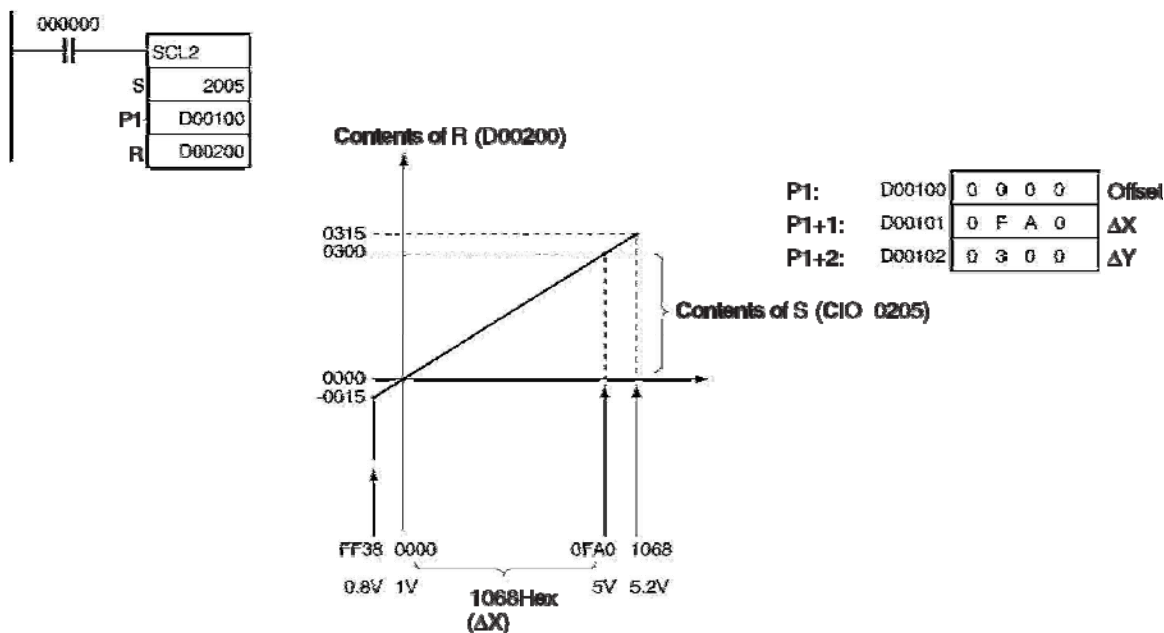
8-2-3- فلگ های مربوط به دستورالعمل SCL2(486)

نام	نشانه	عملکرد
Error Flag	ER	روشن در صورتی که محتوای P1+1 (ΔX) صفر باشد. روشن در صورتی که محتوای P1+2 (ΔY) داده ای BCD نباشد. در بقیه موارد خاموش.
Equal Flag	=	روشن زمانی که نتیجه 0 باشد در بقیه موارد خاموش.
Carry Flag	CY	روشن زمانی که نتیجه منفی باشد. خاموش در صورتی که نتیجه مثبت و یا صفر باشد.

8-2-4- مثال از عملکرد دستور SCL2(486)

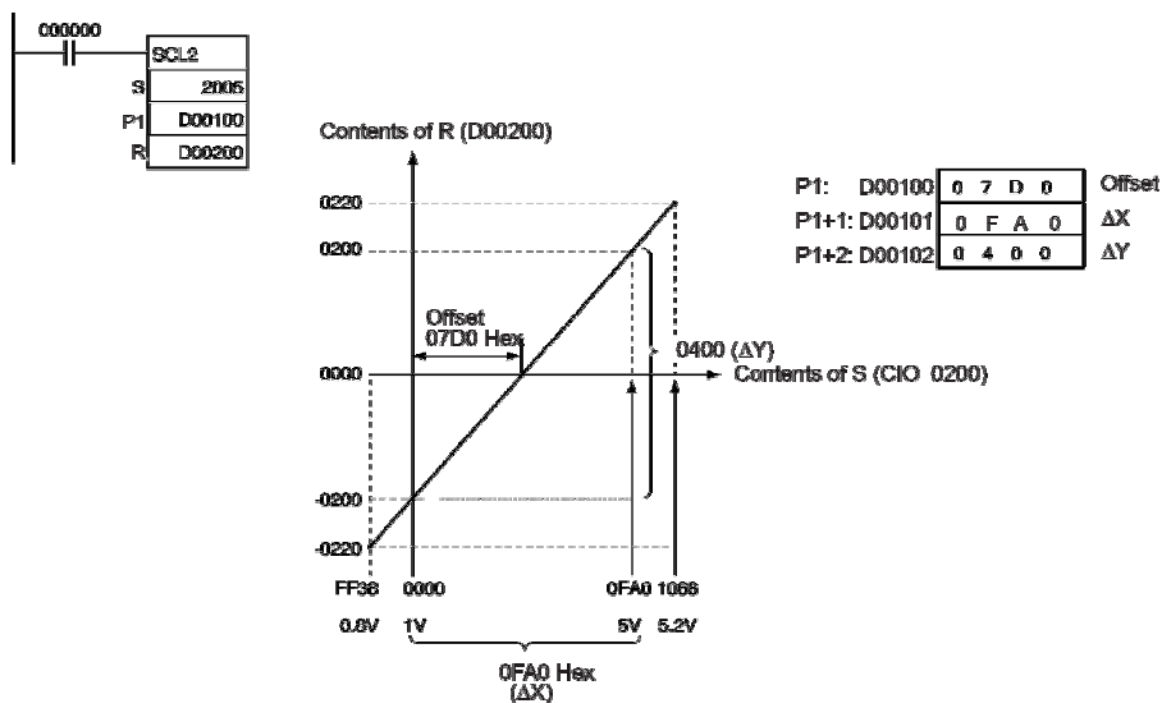
مقیاس کردن سیگنال ورودی آنالوگ بین 1 تا 5 ولت بین 0 تا 300

در مثال زیر فرض بر این است که سیگنال ورودی آنالوگ بین 1 تا 5V به داده ای باینری بین 0000 تا 0FA0 تبدیل شده و در حافظه CIO2005 قرار گرفته است. دستورالعمل SCL2(486) برای تبدیل داده باینری موجود در CIO2005 به داده ای BCD مابین 0 تا 300 مورد استفاده قرار می گیرد. در این مثال زمانی که CIO0.00 روشن می گردد محتوای CIO2005 با توجه به تابع خطی تعیین شده توسط $\Delta X(0FA0)$ ، $\Delta Y(300)$ و $Offset(0)$ مقیاس می گردد. مقادیر پارامتر در D100 تا D102 تعیین شده و نتیجه در D200 ریخته می شود.



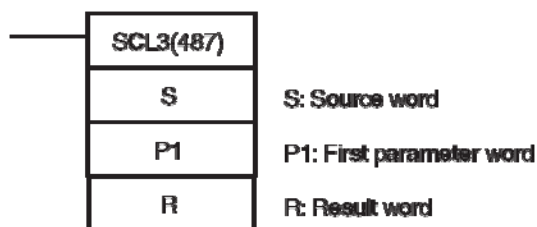
مقیاس کردن سیگنال ورودی آنالوگ بین 1 تا 5 ولت بین 200- تا 200

در مثال زیر فرض بر این است که سیگنال ورودی آنالوگ بین 1 تا 5V به داده ای باینری بین 0000 تا 0FA0 تبدیل شده و در حافظه CIO2005 قرار گرفته است. دستورالعمل SCL2(486) برای تبدیل داده باینری موجود در CIO2005 به داده ای BCD مابین 200- تا 200 مورد استفاده قرار می گیرد. در این مثال زمانی که CIO0.00 روشن می گردد محتوای CIO2005 با توجه به تابع خطی تعیین شده توسط $\Delta X(0FA0)$ ، $\Delta Y(400)$ و Offset(07D0) مقیاس می گردد. مقادیر پارامتر در D100 تا D102 تعیین شده و نتیجه در D200 ریخته می شود.



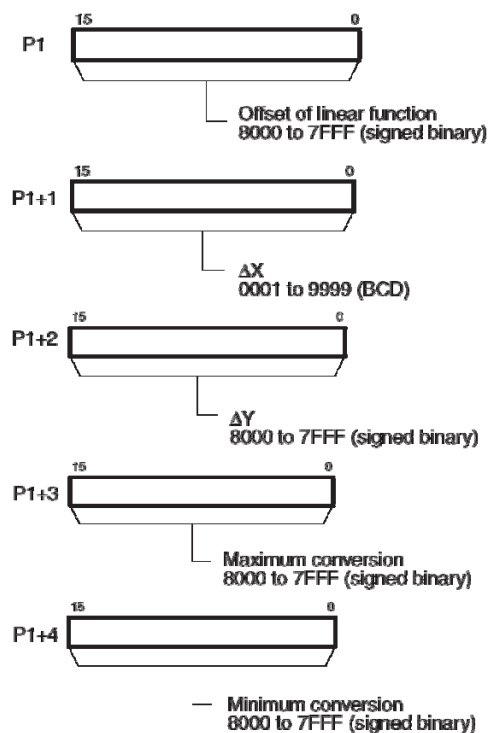
3-8- دستورالعمل مقیاس گذاری 3 (SCL3-Scaling 3)

این دستورالعمل داده های علامت دار BCD را به داده های علامت دار باینری با توجه به تابع خطی تعیین شده، تبدیل می نماید. در تعریف تابع خطی تبدیل می توان مقدار آفست نیز در نظر گرفته می شود.



8-3-1- مشخصات عملگرهای دستور SCL3(487)

در زیر محتوای پنج Word از حافظه که از اولین پارامتر Word (P1) آغاز می گردد نشان داده شده است.



نکته: حافظه های P1 تا P1+4 باید در یک ناحیه باشند.

فضای حافظه	S	P1	R
CIO Area	CIO 0000 to CIO 6143	CIO 0000 to CIO 6139	CIO 0000 to CIO 6143
Work Area	W000 to W511	W000 to W507	W000 to W511
Holding Bit Area	H000 to H511	H000 to H507	H000 to H511
Auxiliary Bit Area	A000 to A959	A000 to A955	A448 to A959
Timer Area	T0000 to T4095	T0000 to T4091	T0000 to T4095
Counter Area	C0000 to C4095	C0000 to C4091	C0000 to C4095
DM Area	D00000 to D32767	D00000 to D32763	D00000 to D32767
EM Area without bank	E00000 to E32767	E00000 to E32763	E00000 to E32767
EM Area with bank	En_00000 to En_32767 (n = 0 to C)	En_00000 to En_32763 (n = 0 to C)	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	---		
	DR0 to DR15	---	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		

8-3-2- شرح عملکرد دستور SCL3(487)

دستورالعمل SCL3(487) برای تبدیل داده BCD علامت دار موجود در S به داده باینری علامت دار مورد استفاده قرار می گیرد (داده BCD شامل مقدار قدرمطلق و همچنین فلگ Carry که نشان دهنده علامت است می باشد) و این نتیجه باینری در حافظه تعیین شده به عنوان نتیجه R قرار می گیرد. مقیاس گذاری با استفاده از این دستورالعمل با توجه به تابع خطی معینی که با توجه به شیب $(\Delta X, \Delta Y)$ ، آفست و همچنین مقدار حداقل و حداکثر باینری تعیین می گردد، انجام می گیرد. مقادیر ΔX ، ΔY ، آفست و همچنین مقدار حداکثر و حداقل داده تبدیل شده باینری در حافظه های P1 تا P1+4 تعیین می گردد. علامت نتیجه این مقیاس گذاری با استفاده از فلگ Carry تعیین می گردد (روشن: منفی، خاموش: مثبت). برای خاموش و روشن نمودن فلگ Carry می توان از دستورالعملهای STC(040) و CLC(041) استفاده نمود.

از معادله زیر برای تبدیل استفاده می گردد:

$$R = \frac{\Delta Y}{\text{Binary conversion of } \Delta X} \times ((\text{Binary conversion of } S) + (\text{Offset}))$$

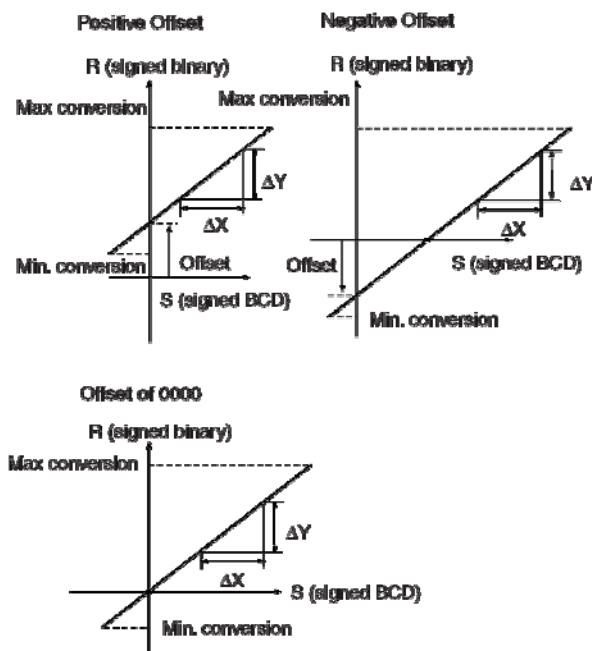
شیب خط $\Delta Y/\Delta X$ می باشد.

آفست و شیب خط می تواند مقداری مثبت، صفر و یا مقداری منفی باشد. استفاده از مقادیر منفی امکان مقیاس گذاری معکوس را فراهم می آورد.

نتیجه به نزدیکترین مقدار صحیح گرد خواهد شد.

محتوای S به صورت یک مقدار قدرمطلق BCD در نظر گرفته خواهد شد و علامت آن به صورت فلگ Carry در نظر گرفته خواهد شد. پس محتوای مرجع S عددی BCD مابین 9999- تا 9999 خواهد بود.

اگر نتیجه کوچکتر از مقدار حداقل تعیین شده باشد، حداقل مقدار به عنوان نتیجه در نظر گرفته خواهد شد و اگر نتیجه بزرگتر از مقدار حداکثر تعیین شده باشد، حداکثر مقدار به عنوان نتیجه در نظر گرفته خواهد شد.



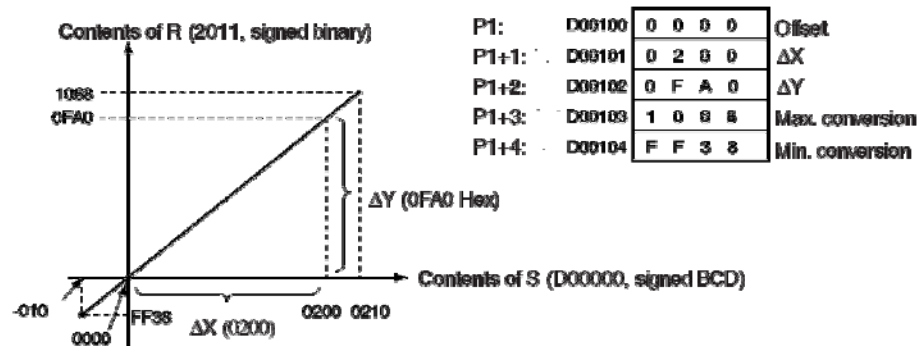
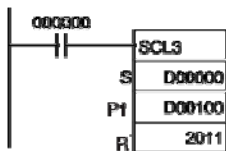
دستورالعمل SCL3(487) را می توان برای مقیاس کردن داده های تعیین شده توسط کاربر به سیگنال خروجی آنالوگ به کار برد. به عنوان مثال با استفاده از دستور SCL3(487) می توان داده BCD مابین 0 تا 200 را به داده باینری مابین 0000 تا 0FA0 هگزادسیمال تبدیل نموده و به صورت سیگنال آنالوگ خروجی 1 تا 5V به خروجی آنالوگ منتقل نمود.

8-3-3- فلگ های مربوط به دستورالعمل SCL3(487)

نام	نشانه	عملکرد
Error Flag	ER	روشن در صورتی که محتوای S عددی BCD نباشد. روشن در صورتی که محتوای P1+1 (ΔX) بین 1 تا 9999 نباشد. در بقیه موارد خاموش.
Equal Flag	=	روشن زمانی که نتیجه 0 باشد در بقیه موارد خاموش.
Negative Flag	N	روشن زمانی که بارزترین بیت R یک باشد. در بقیه موارد خاموش.

8-3-4- مثال از عملکرد دستور SCL3(487)

زمانی که عددی مابین 0 تا 200 به صورت یک سیگنال خروجی آنالوگ مقیاس می گردد (به عنوان مثال به سیگنال خروجی 1 تا 5V تبدیل می گردد). داده علامت دار BCD مابین 0 تا 200 به داده ای باینری و علامت دار مابین 0000 تا 0FA0 به عنوان یک خروجی آنالوگ تبدیل می گردد. در مثال زیر زمانی که CIO 0.00 روشن می گردد، محتوای حافظه D0 با توجه به تابع خطی تعریف شده توسط کاربر ΔX(0200)، ΔY(0FA0) و Offset(0) مقیاس می گردد. این مقادیر و پارامترها در D100 تا D102 تعیین می گردد. علامت داده BCD موجود در D0 با استفاده از فلگ Carry تعیین می گردد. نتیجه این مقیاس در CIO2011 قرار می گیرد.



9- Subroutine ها

9-1- فراخوانی Subroutine (SBS)

این دستورالعمل Subroutine با شماره معین را فراخوانی نموده و برنامه آن را اجرا می نماید.



9-1-1- مشخصات عملگرهای دستور SBS(091)

N: شماره Subroutine

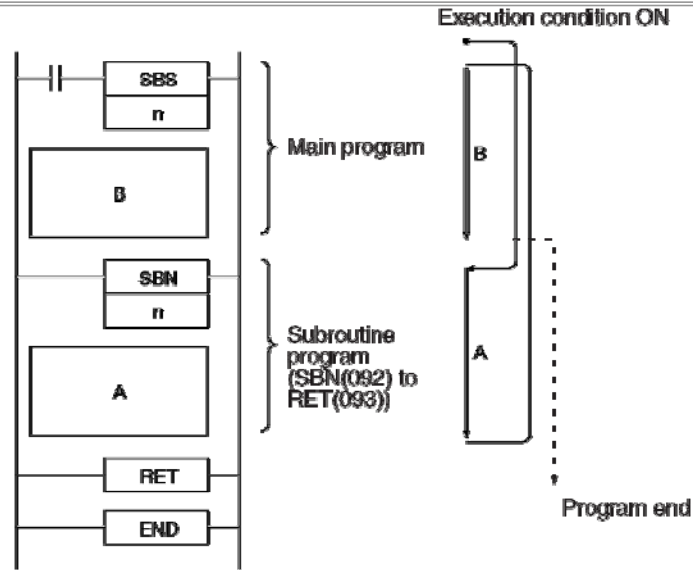
مشخص کننده شماره Subroutine بین 0 تا 1023 دسیمال می باشد.

فضای حافظه	N
CIO Area	---
Work Area	---
Holding Bit Area	---
Auxiliary Bit Area	---
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	0 to 1023 (Decimal)
Data Registers	---
Index Registers	---
Indirect addressing using Index Registers	---

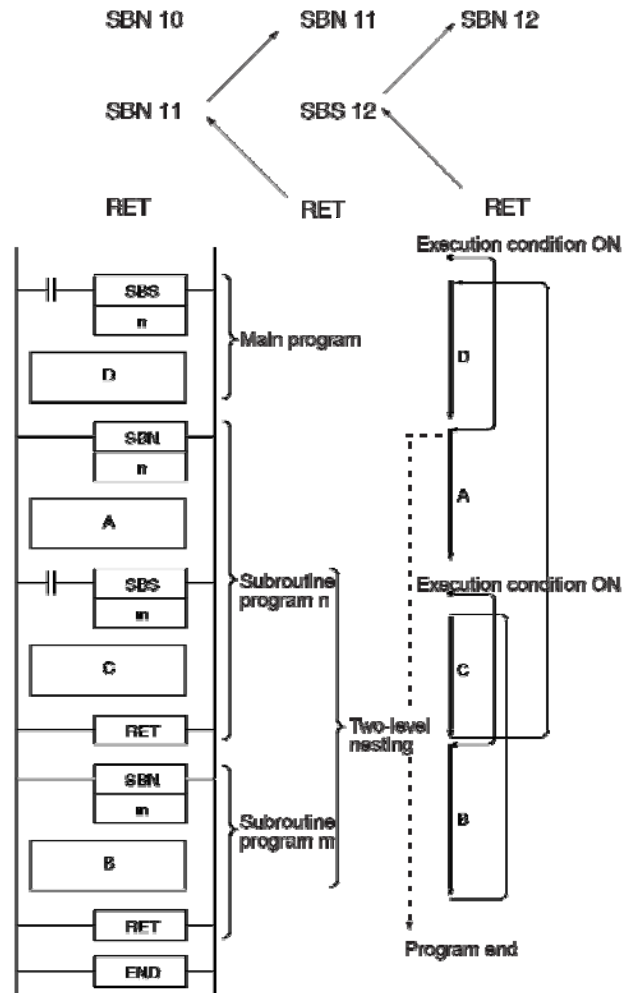
نکته : در مورد PLC های مدل CJ1M-CPU11، CJ1M-CPU21 شماره Subroutine باید مابین 0 & 255 دسیمال انتخاب گردد.

9-1-2- شرح عملکرد دستور SBS(091)

دستورالعمل SBS(091) زیربرنامه ای را با شماره معین فراخوانی می نماید. این زیربرنامه، برنامه ای است که مابین دو دستورالعمل SBN(092) و RET(093) نوشته شده باشد. زمانی که اجرای زیربرنامه انجام شد خواندن برنامه از دستور بعد از SBS(091) ادامه پیدا خواهد کرد.



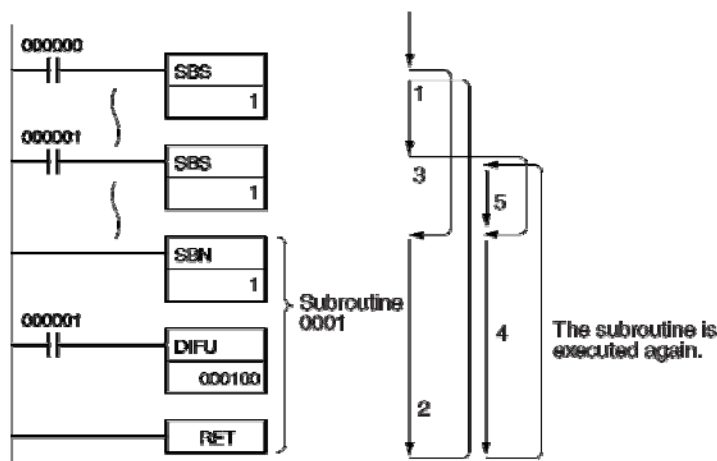
می توان این زیربرنامه ها را تا 16 سطح مختلف به صورت تو در تو استفاده نمود. این استفاده تو در تو به منظور فراخواندن یک زیربرنامه در زیر برنامه دیگر می باشد که در نمودار زیر استفاده تو در تو از زیر برنامه ها تا سه سطح نمایش داده شده است.



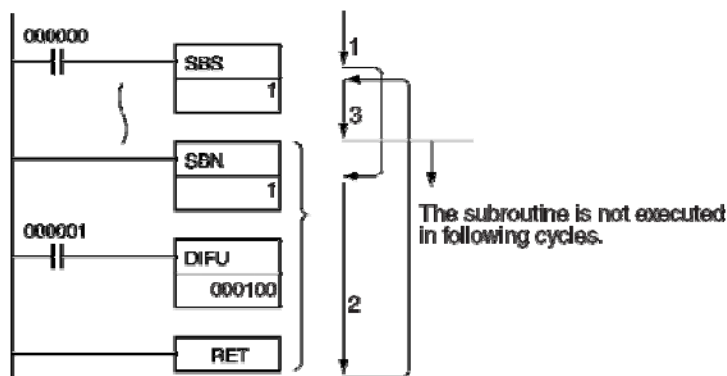
نکته: یک زیر برنامه را می توان بیش از یک بار در برنامه فراخوانی نمود.

9-1-3 Subroutine و دستورات لبه

نکات قابل توجه در مورد دستورات لبه همچون دستورات DIFU(013) و DIFD(014) و یا دستورات عملیاتی مختلف به صورت لبه^۱ بالارونده زمانی که در یک زیربرنامه مورد استفاده قرار می گیرد در ادامه شرح داده می شود. چگونگی عملکرد دستورات لبه در یک زیربرنامه هنگامی که این زیربرنامه چندین بار در یک سیکل برنامه اجرا می گردد غیر قابل پیش بینی می باشد. به عنوان مثال در برنامه زیر زمانی که ورودی CIO0.00 روشن می باشد زیربرنامه شماره 1 اجرا شده و CIO1.00 روشن می گردد، اگر CIO0.01 نیز روشن باشد این زیر برنامه دوباره اجرا خواهد شد و در این بار اجرای برنامه CIO1.00 بدون در نظر گرفتن حالت CIO0.01 خاموش خواهد شد.



همچنین در صورتی که در یک سیکل با فراخوانی یک زیر برنامه خروجی دستورات لبه روشن گردد اگر این زیر برنامه در سیکل های بعدی فراخوانی نگردد این خروجی همچنان روشن خواهد ماند و خاموش نخواهد شد.



در این مثال زمانی که CIO0.00 روشن باشد زیربرنامه 1 فراخوانی شده و در صورتی که CIO0.01 از حالت خاموش به روشن تغییر حالت می دهد خروجی CIO1.00 روشن خواهد شد. اگر در سیکل بعدی برنامه CIO0.00 خاموش شده باشد و این زیربرنامه فراخوانی نگردد خروجی دستورات لبه روشن باقی خواهد ماند.

9-1-4- فلگ های مربوط به دستورالعمل SBS(091)

نام	نشانه	عملکرد
Error Flag	ER	زمانی که استفاده تو در تو از زیربرنامه ها از 16 سطح بیشتر گردد روشن می شود. زمانی که زیربرنامه با شماره مشخص شده موجود نباشد. زمانی که زیر برنامه ای که در حال اجرا است فراخوانی گردد روشن می شود. زمانی که زیربرنامه معین در همان Task فراخوانی شده تعریف نشده باشد روشن می گردد. در بقیه موارد خاموش می باشد.

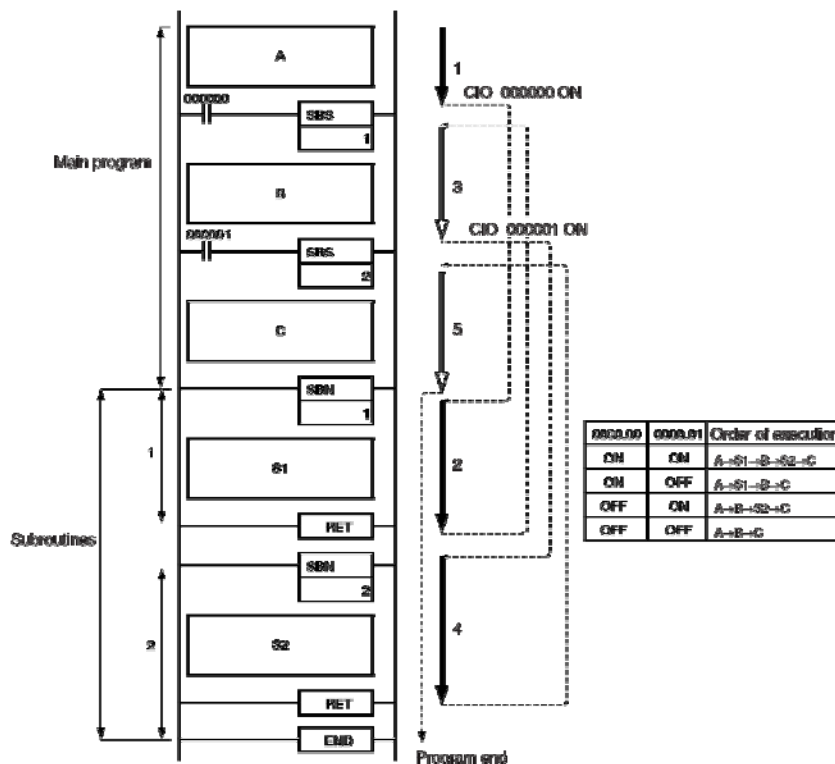
9-1-5- نکات قابل توجه در مورد دستورالعمل SBS(091)

دستورالعمل SBS(091) و دستورالعمل SBN(092) متناظر با آن برای تعریف زیربرنامه باید در یک Task نوشته شوند. اگر SBN(092) با شماره متناظر در همان Task نوشته نشده باشد خطایی روی خواهد داد. زیربرنامه ها زمانی که بین دستورالعملهای IL(002) و ILC(003) قرار می گیرند هیچ عملیات خاصی انجام نمی دهند و مانند دستور NOP(000) رفتار خواهند کرد.

9-1-6- مثال هایی از عملکرد دستور SBS(091)

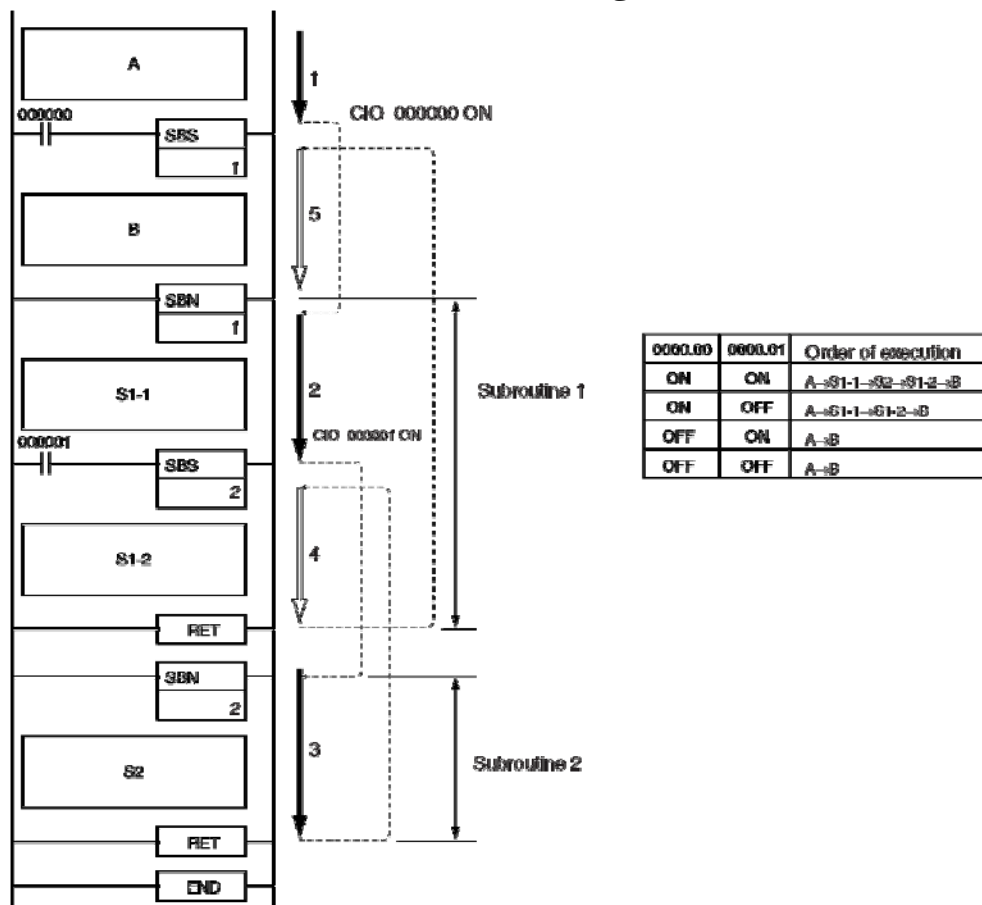
مثال 1: زیربرنامه های ترتیبی (بدون تداخل)

در مثال زیر زمانی که CIO0.00 روشن باشد زیربرنامه شماره 1 اجرا می گردد و بعد از اجرای این زیربرنامه اجرای برنامه به دستورالعمل بعد از SBS(091) با شماره 1 برمی گردد. زمانی که CIO0.01 روشن باشد زیر برنامه شماره 2 اجرا می گردد و بعد از اجرای این زیربرنامه اجرای برنامه به دستورالعمل بعد از SBS(091) با شماره 2 برمی گردد.



مثال 2: زیربرنامه های تو در تو

در مثال زیر زمانی که CIO0.00 روشن باشد زیر برنامه شماره 1 اجرا می گردد و زمانی که CIO0.01 روشن باشد زیر برنامه شماره 2 اجرا می گردد و بعد از اجرای زیر برنامه شماره 2 برنامه به خط بعد از SBS(091) با شماره 2 بر خواهد گشت. زمانی که اجرای زیر برنامه شماره 1 به پایان رسید اجرای برنامه به دستور بعد از SBS(091) با شماره یک بر می گردد.


9-2-2- تعریف کردن Subroutine (SBN)

این دستورالعمل ابتدای زیر برنامه با شماره معین را نشان می دهد. این دستورالعمل همواره با دستور RET(093) برای مشخص نمودن ناحیه زیر برنامه مورد استفاده قرار می گیرد.


9-2-1- مشخصات عملگرهای دستور SBN(092)

N: شماره Subroutine

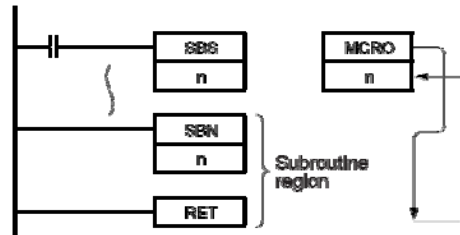
مشخص کننده شماره Subroutine بین 0 تا 1023 دسیمال می باشد.

فضای حافظه	N
CIO Area	---
Work Area	---
Holding Bit Area	---
Auxiliary Bit Area	---
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	0 to 1023 (Decimal)
Data Registers	---
Index Registers	---
Indirect addressing using Index Registers	---

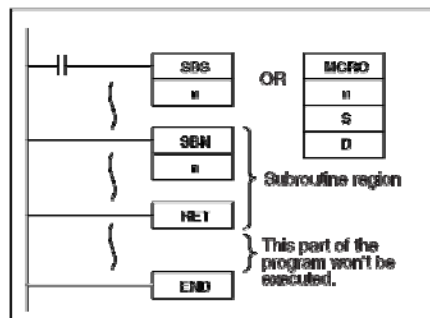
نکته : در مورد PLC های مدل CJ1M-CPU11، CJ1M-CPU21 شماره Subroutine باید مابین 0 تا 255 & دسیمال انتخاب گردد.

9-2-2- شرح عملکرد دستور (SBN(092)

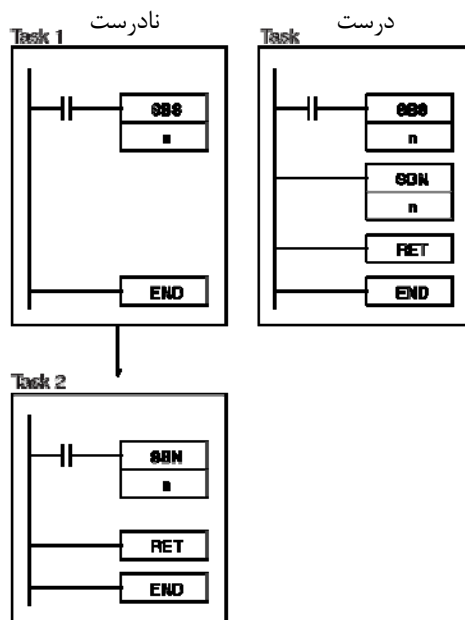
دستورالعمل (SBN(092 ابتدا و شروع یک زیر برنامه با شماره معین را نشان می دهد. انتهای زیر برنامه با دستورالعمل (RET(093 معین می گردد. قسمتی از یک برنامه را که با دستورالعمل (SBN(092 آغاز شده باشد مشخص کننده زیر برنامه می باشد. این بخش از برنامه فقط در صورتی که با استفاده از دستور (SBS(091 فراخوانی شود اجرا می گردد.



زمانی که زیر برنامه ای فراخوانی نشود و در نتیجه اجرا نگردد این زیر برنامه مانند NOP(00 رفتار خواهد نمود. این زیر برنامه ها معمولاً در انتهای برنامه و قبل از دستور END(001 در هر Task باید قرار داد. اگر بخشی از برنامه بعد از محدوده زیر برنامه قرار گیرد این بخش از برنامه نادیده گرفته می شود.



نکته: باید از تعریف زیربرنامه در همان Task فراخوانی آن توسط دستور SBS(091) اطمینان حاصل نمود. یک زیربرنامه تعریف شده در یک Task را نمی توان در Task دیگری فراخوانی نمود مگر اینکه این زیربرنامه در یک Task وقفه تعریف شده باشد.



3-9- بازگشت Subroutine (RET)

این دستورالعمل نشان دهنده انتهای یک زیربرنامه می باشد که همواره با دستور SBN(092) مورد استفاده قرار می گیرد.



9-3-1- شرح عملکرد دستور RET(093)

دستور RET(093) انتهای یک زیربرنامه را که با دستور SBN(092) آغاز می گردد را نشان می دهد. زمانی که اجرای برنامه به دستور RET(093) رسید برنامه به دستور بعد از دستور SBS(091) که این زیربرنامه را فراخوانی نموده است باز می گردد.

زمانی که زیربرنامه ای فراخوانی نمی گردد دستورالعملهای این زیر برنامه مانند NOP(000) رفتار خواهد نمود.



10- آدرس دهی غیر مستقیم

10-1- مقدمه ای بر آدرس دهی غیر مستقیم

یکی از روش های کارا و مورد استفاده در طراحی های پیچیده برای ذخیره سازی داده ها آدرس دهی غیر مستقیم می باشد.

آدرس دهی غیر مستقیم در حافظه Data Memory, EM و IR با استفاده از اشاره گر های نرم افزاری (Pointer) قابل اجرا می باشد. یک اشاره گر مانند یک آدرس عمل خواهد نمود. به عنوان مثال با قرار دادن یکی از حافظه های DM به عنوان اشاره گر، محتوای داخل آن حافظه به صورت یک آدرس در نظر گرفته خواهد شد این آدرس یکی از حافظه های منطقه DM با شماره ای برابر با محتوای اشاره گر خواهد بود. در واقع اشاره گر به آدرسی که داده ها باید در آن ذخیره گردند اشاره می کند. با استفاده از این روش داده ها را می توان دریافت و در حافظه مورد نظر قرار داد.

در مثال زیر، حافظه D50 به عنوان اشاره گر در نظر گرفته شده است. که چگونگی عملکرد این اشاره گر در زیر نشان داده شده است.

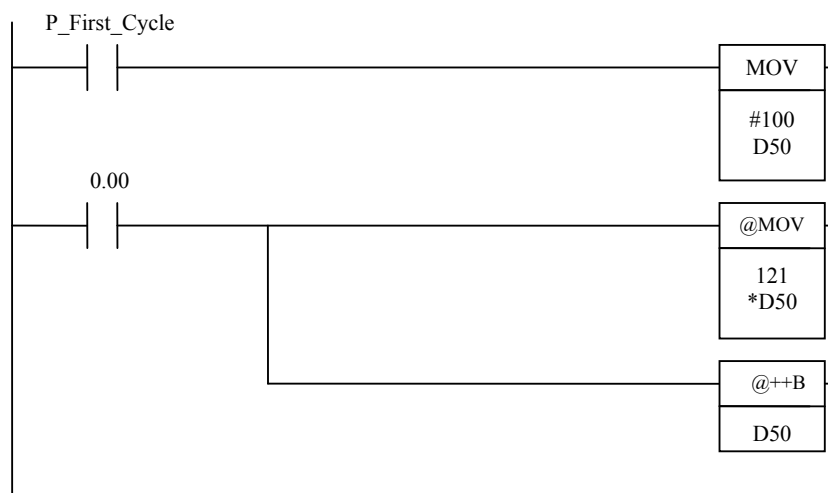
D50	0	1	0	0	→	1	3	A	B	D100
	0	1	0	1	→	2	F	3	D	D101
	0	1	0	2	→	1	7	2	9	D102
						6	3	4	8	D103
						2	C	D	5	D104
						3	4	7	2	D105
						2	A	9	7	D106
						F	4	B	3	D107
						6	9	7	1	D108
						6	0	3	9	D109

باید به این نکته توجه نمود که برای تعیین آدرس D50 به عنوان اشاره گر باید قبل از آن از "*" استفاده شود. زمانیکه از *D50 در دستورالعمل ها استفاده می گردد علامت "*" نشان دهنده آدرس دهی غیر مستقیم می باشد.



در مثال بالا، زمانی که بیت 0.00 روشن می شود، محتوای حافظه CIO121 به حافظه ای که اشاره گر *D50 به آن اشاره می کند منتقل می شود. مقدار موجود در حافظه CIO121 در D50 کپی نمی گردد. در این مثال به دلیل اینکه حافظه D50 محتوای عدد 0100 می باشد زمانی که بیت ورودی 0.00 روشن می گردد محتوای CIO121 در حافظه D100 که به آن اشاره شده است، کپی می شود. اگر محتوای حافظه D50 عدد 0101 باشد، مقدار موجود در CIO121 در D101 کپی می گردد.

استفاده از آدرس دهی غیر مستقیم یکی از روشهای مؤثر در کاهش حجم برنامه می باشد که شاید در مثال پیش به وضوح دیده نشود. برای استفاده از آدرس دهی غیر مستقیم باید چگونگی استفاده از آن در برنامه نردبانی بیان گردد.



در برنامه بالا که از آدرس دهی غیرمستقیم استفاده شده است، در ابتدا حافظه D50 با فعال شدن فلگ First Cycle با عدد 0100 بار می گردد. زمانی که ورودی 0.00 روشن می گردد محتوای CIO121 به حافظه DM که اشاره گر *D50 به آن اشاره می کند (D100) منتقل می شود.

سپس محتوای حافظه D50 افزایش یافته و از 100 به 101 تبدیل می شود. این بدان معناست که زمانی که برای دومین بار بیت 0.00 فعال می گردد محتوای CIO121 به حافظه D101 منتقل شده و همچنین محتوای D50 از 101 به 102 تبدیل می شود. به همین ترتیب، هر بار فعال شدن 0.00 باعث می شود که محتوای CIO121 در حافظه جدیدی از منطقه DM ذخیره گردد.

علاوه بر استفاده از این روش برای ذخیره کردن داده ها در ناحیه DM می توان از این روش برای خواندن داده ها از این منطقه از حافظه نیز استفاده نمود.

10-1-1- آدرس دهی غیر مستقیم در ناحیه DM

حافظه های ناحیه DM را می توان به دو روش به صورت غیرمستقیم آدرس دهی نمود: روش باینری و روش BCD.

آدرس دهی غیرمستقیم به شیوه باینری (@D):

زمانی که "@" را قبل از آدرس های ناحیه DM قرار می دهیم، محتوای این حافظه به صورت باینری در نظر گرفته شده و این حافظه به صورت غیر مستقیم به حافظه ای با شماره این آدرس باینری اشاره خواهد نمود. ناحیه DM (D0 تا D32767) را می توان با استفاده از این شیوه و داده های باینری بین 0000 تا 7FFF به صورت غیر مستقیم آدرس دهی نمود.



آدرس دهی غیرمستقیم به شیوه BCD (*D):

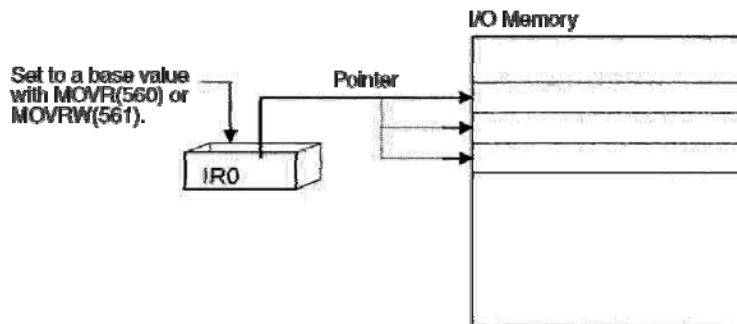
زمانی که "*" را قبل از آدرس های ناحیه DM قرار می دهیم، محتوای این حافظه به صورت BCD در نظر گرفته شده و این حافظه به صورت غیر مستقیم به حافظه با شماره این آدرس BCD اشاره خواهد نمود. فقط بخشی از ناحیه DM (D0 تا D9999) را می توان با استفاده از این شیوه و اعداد بین 0000 تا 9999 به صورت غیر مستقیم آدرس دهی نمود.



10-1-2- آدرس دهی غیر مستقیم در ناحیه IR

وقتی از ناحیه حافظه IR به صورت یک عملگر و با پیشوند "I" در دستورالعمل ها استفاده می شود این حافظه به صورت یک اشاره گر عمل خواهد نمود در نتیجه دستورالعمل مورد نظر محتوای حافظه IR را به عنوان آدرس در نظر خواهد گرفت نه خود آن آدرس را. به صورت کلی منطقه Index Register اشاره گر ورودی/خروجی ها می باشد.

- همه حافظه های ورودی/خروجی (به غیر از حافظه های IR، DR و فلگ های شرطی) را می توان با استفاده از ورودی/خروجی های PLC تعیین نمود. در این روش نیازی به تعیین ناحیه حافظه نمی باشد.
 - علاوه بر روش اصلی آدرس دهی غیر مستقیم در ناحیه IR، این آدرس ها می توانند از یک عدد ثابت و یا از محتوای ناحیه DR، به صورت افزایشی یا کاهشی تأثیر بپذیرند. این روش را می توان در برنامه های حلقه برای خواندن و یا نوشتن داده ها زمانیکه آدرس با هر بار اجرای حلقه یک واحد افزایش یا کاهش پیدا می کند، مورد استفاده قرار داد.
- با در نظر گرفتن آفست و یا تغییرات افزایشی/کاهشی محتوای اولیه ناحیه IR را می توان با استفاده از دستورالعمل MOV (560) و یا MOVW (561) بار نمود و سپس به صورت اشاره گر در دستورالعمل های مختلف مورد استفاده قرار داد.

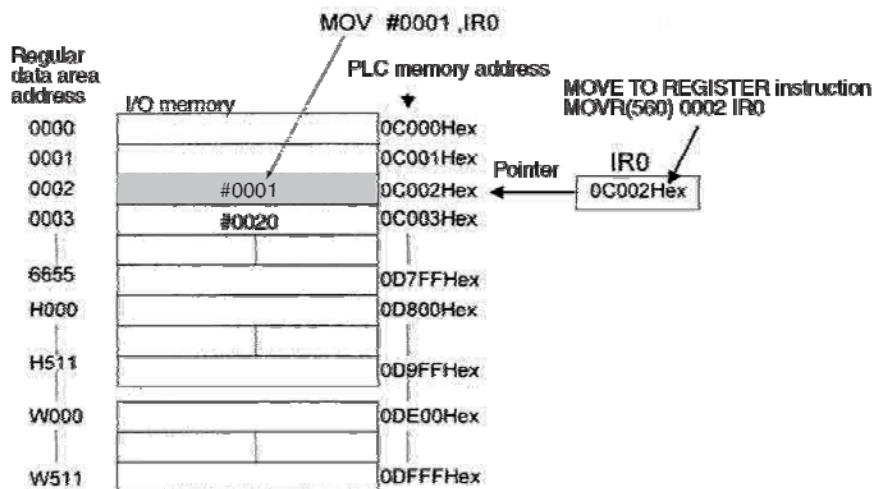


مثال:

در مثال زیر چگونگی ذخیره نمودن محتوای یک حافظه از PLC مانند CIO0002 در یکی از حافظه های ناحیه IR (IRO) نشان داده شده است. در این مثال یکی از حافظه های IR در یکی از دستورالعمل ها و همچنین روش افزایش خودکار مورد استفاده قرار گرفته است.

MOVR(560) 0002 IR0
 MOV(021) #0001 ,IR0
 MOV #20 +1,IR0

آدرس حافظه CIO 0002 در IR0 ذخیره می شود.
 عدد #0001 در حافظه PLC که توسط IR0 به آن اشاره شده ریخته می شود
 محتوای IR0 خوانده شده، یک واحد افزایش می یابد و عدد #20 در آن حافظه که به آن اشاره شده ریخته می شود.



11- توابع وقفه

11-1- نگاهی کلی به توابع وقفه (مدل CJ1M)

عملکرد PLC ها به طور معمول به صورت سیکلی انجام می پذیرد (نظارت بر پردازش ← اجرای برنامه ← بازنگری و تجدید ورودی و خروجی ها ← خدمات سرویس های جانبی) توسط Task هایی که در مراحل اجرای برنامه به صورت دوره ای انجام می شوند. توابع وقفه برای ایجاد وقفه ای موقتی در پردازش سیکلی برنامه و اجرای یک برنامه ویژه با رویداد شرایط از پیش تعریف شده مورد استفاده قرار می گیرند.

11-1-1- انواع توابع وقفه

ورودی های وقفه (حالت مستقیم)

زمانی که یکی از ورودی های وقفه موجود روی CPU از حالت خاموش به روشن (یا روشن به خاموش) تغییر حالت می دهد Task وقفه متناظر با آن ورودی وقفه اجرا می گردد. Task های وقفه از شماره 140 الی 143 به چهار ورودی وقفه موجود روی CPU اختصاص داده شده است.

ورودی های وقفه (حالت شمارشی)

در این روش پالسهای ورودی مربوط به یکی از ورودی های وقفه موجود روی CPU شمارش می شود زمانی که این شمارش به تعداد از پیش تعیین شده رسید، Task وقفه متناظر با آن ورودی اجرا می گردد. حداکثر فرکانس پاسخ ورودی وقفه در این حالت 1kHz می باشد.

وقفه های برنامه ریزی شده

این تابع Task های وقفه را در فواصل زمانی خاص که توسط تایمر موجود در CPU اندازه گیری می شود، اجرا می کند. این فاصله زمانی را می توان با ضربی از 10ms، 1ms و یا 0.1ms تنظیم نمود. حداقل فاصله زمانی 0.5ms می باشد. Task وقفه شماره 2و3 به وقفه های برنامه ریزی شده اختصاص داده شده است.

وقفه های شمارنده پرسرعت

این تابع پالسهای ورودی پرسرعت موجود روی CPU را شمرده و زمانی که تعداد شمرده شده با تعداد از پیش تعریف شده برابر می گردد و یا از مقدار از پیش تعیین شده کمتر می شود (یک مقدار ثابت یا محدوده ای برای مقایسه) Task وقفه مورد نظر را اجرا می نماید، یکی از Task های وقفه مابین 0 و 255 را می توان با استفاده از دستورالعمل خاص به این ورودی پر سرعت اختصاص داد.

وقفه های خارجی

وقتی از کارتهای وقفه و یا کارتهای ورودی/خروجی ویژه استفاده شود، این امکان وجود دارد که یکی از Task های وقفه بین 0 تا 255 را با استفاده از دستورالعمل خاص به آنها اختصاص داد.

وقفه قطع برق

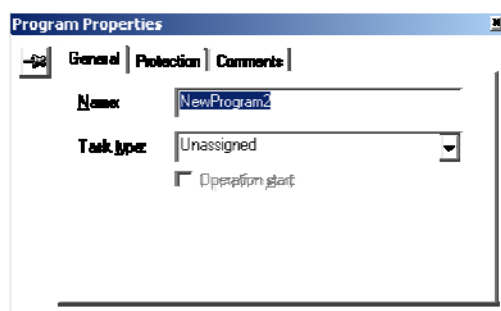
وقتی برق PLC قطع می شود در صورت تعریف این وقفه اجرا می شود.

11-1-2- ایجاد Task وقفه

1. در فضای کاری پروژه *New PLC1* را انتخاب و روی آن راست کلیک نمایید. در منوی باز شده *Insert Program* را انتخاب نمایید. یک برنامه جدید با نام *New Program 2 (Unassigned)* در فضای کاری مربوط به پروژه ایجاد می شود.

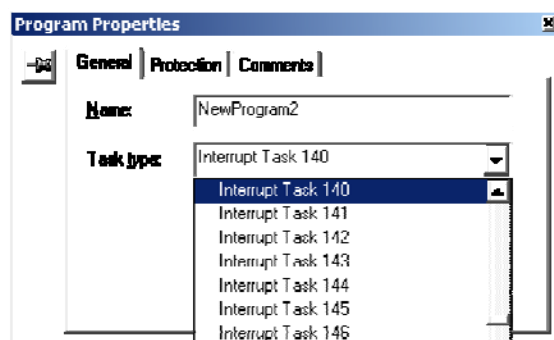


2. روی *New Program 2 (Unassigned)* راست کلیک کرده و از منوی باز شده *Properties* را انتخاب نمایید.



3. در پنجره باز شده که نمایش دهنده مشخصات برنامه است در قسمت *Task Type* نوع Task را انتخاب نمایید.

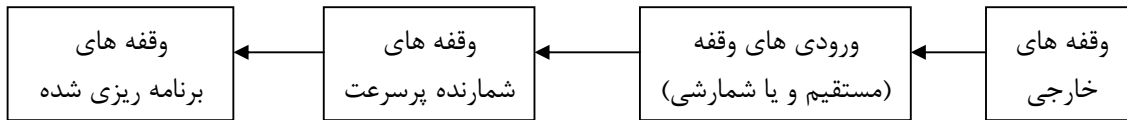
در این مثال Task وقفه شماره 140 به *New Program 2* اختصاص داده شده است.



با بستن این پنجره بعد از انتخاب نوع Task می توان برنامه ای را که به عنوان Task شماره 140 اجرا می گردد را ایجاد نمود.
 برنامه ای مستقل می توان برای هر یک از Task های وقفه تعریف کرد و در انتهای آن باید دستور END(001) را وارد نمود.

11-1-3- الویت Task های وقفه

ورودی های وقفه (مستقیم و یا شمارشی)، وقفه های شمارنده های پرسرعت و همچنین وقفه های برنامه ریزی شده همه دارای الویت یکسان می باشند. زمانی که Task وقفه A (به عنوان مثال Task متناظر با یک ورودی وقفه) اجرا می گردد، اگر Task وقفه B (به عنوان مثال Task وقفه برنامه ریزی شده) فراخوانی شود پردازش Task A متوقف نخواهد شد. پردازش Task B بعد از اتمام پردازش Task A آغاز می گردد.
 در صورتی که وقفه های مختلف دقیقاً در یک زمان روی دهد الویت اجرای آنها به صورت زیر خواهد بود:



اگر دو وقفه از یک نوع به صورت همزمان روی دهد Task وقفه با شماره کوچکتر زودتر اجرا می گردد.
نکته: زمانی که در برنامه ای احتمال وقوع وقفه های مختلف به صورت همزمان وجود داشته باشد این وقفه ها به ترتیبی که در بالا دیده می شود اجرا می گردد، نکته حائز اهمیت این که از وقوع شرایط وقفه تا اجرای برنامه متناظر با آن وقفه زمانی به طول می انجامد به ویژه این امکان وجود دارد که Task های برنامه ریزی شده در زمان از پیش تعیین شده اجرا نگردد، در نتیجه برنامه را باید به صورتی طراحی نمود که در صورت لزوم تداخلی در اجرای Task های وقفه ایجاد نگردد.

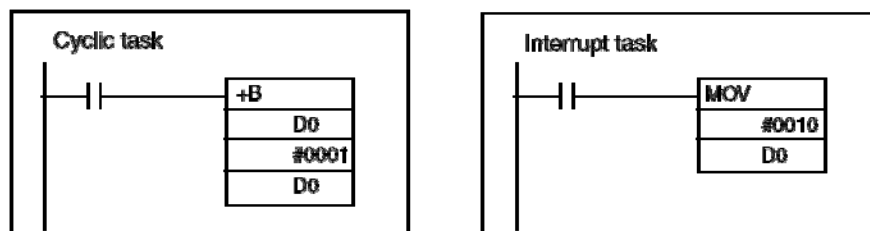
11-1-4- پردازش دوتایی توسط Task های دوره ای و Task های وقفه

اگر حافظه ای هم توسط Task های سیکلی و هم Task های وقفه پردازش شود، برای غیر فعال نمودن وقفه باید از پوشش وقفه (Intrrupt Mask) استفاده نمود.

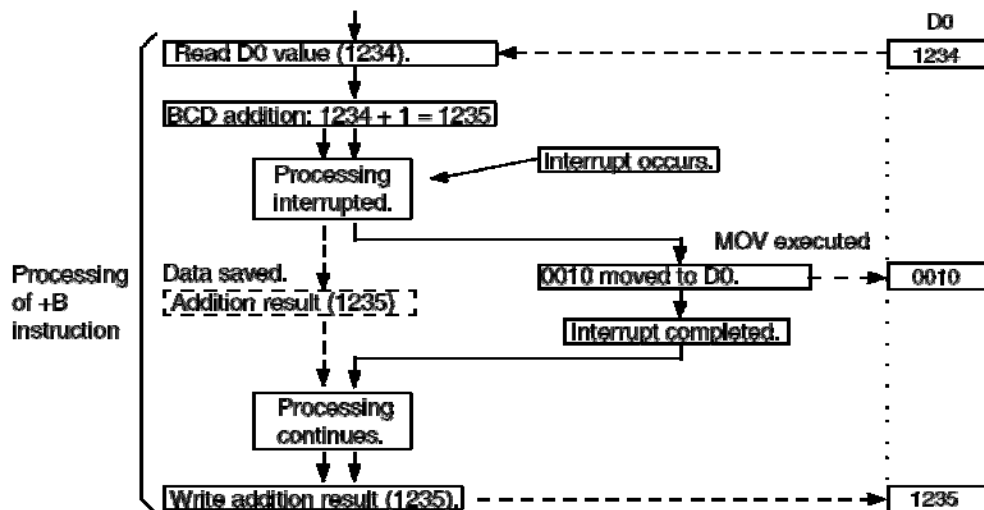
زمانی که وقفه ای روی می دهد، اجرای برنامه سیکلی متوقف می گردد، این وقفه حتی در طول اجرای یک دستورالعمل در Task های سیکلی نیز روی خواهد داد در اینصورت قسمتی از پردازش که انجام شده است ذخیره می گردد. زمانی که پردازش Task وقفه کامل شود، پردازش Task سیکلی ادامه خواهد یافت و داده ها به حالت ذخیره شده (قبل از اجرای Task وقفه) بازگردانده خواهد شد. اگر حافظه ای توسط دستورالعمل های به کار رفته شده در Task وقفه تغییر نماید این تغییر بعد از برگرداندن داده ها به حالت ذخیره شده در Task های سیکلی تأثیر خود را از دست خواهد داد.

برای جلوگیری از ایجاد وقفه در اجرای دستورالعمل در هنگام پردازش، باید دستورالعمل DI(693) برای غیرفعال نمودن وقفه، درست قبل از دستورالعمل و همچنین دستورالعمل EI(694) برای فعال نمودن دوباره وقفه درست بعد از دستورالعمل وارد شود.

- در مثال زیر پردازش دوتایی توسط یک Task وقفه، که در هنگام پردازش دستورالعمل +B ایجاد وقفه نموده است و حافظه مشترکی را تحت تأثیر قرار داده نشان داده شده است.

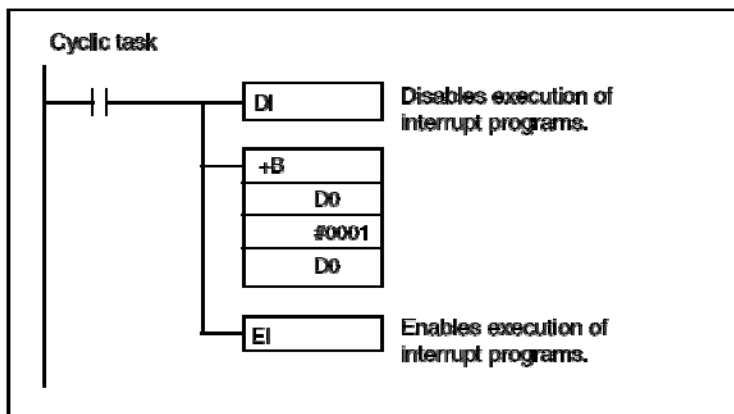


چگونگی پردازش:

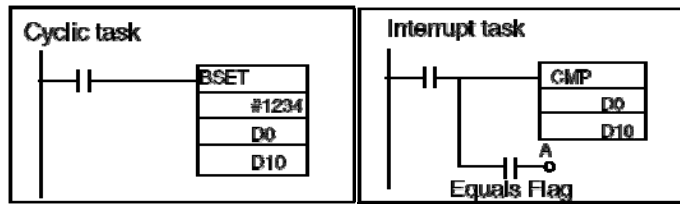


در زمان پردازش دستورالعمل +B وقفه روی می دهد، بنابراین نتیجه دستورالعمل بدون اینکه در آدرس مقصد (D0) نوشته شود به صورت موقتی ذخیره می گردد. در طول اجرای Task وقفه عدد #0010 به آدرس D0 منتقل می گردد، ولی زمانی که به پردازش Task دوره ای بازمی گردد نتیجه ذخیره شده دستورالعمل +B در آدرس D0 نوشته می شود. در نتیجه اجرای Task وقفه هیچ تأثیری در برنامه نخواهد داشت.

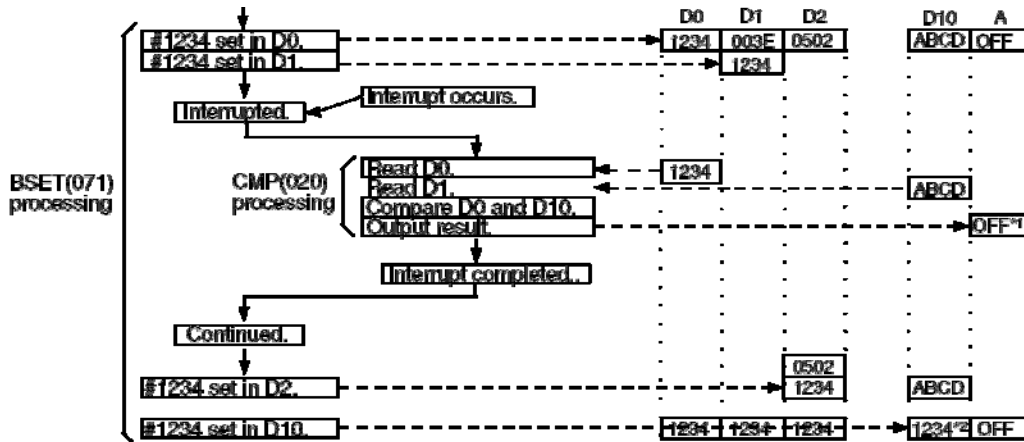
جلوگیری از پردازش دوتایی



- در مثال زیر پردازش دوتایی توسط یک Task وقفه نشان داده شده است، وقفه در زمان پردازش دستور BSET که در حال نوشتن مقداری ثابت در یک بلوک از حافظه می باشد، روی می دهد این پردازش دوتایی باعث نتیجه نادرست دستور مقایسه در Task وقفه می شود.

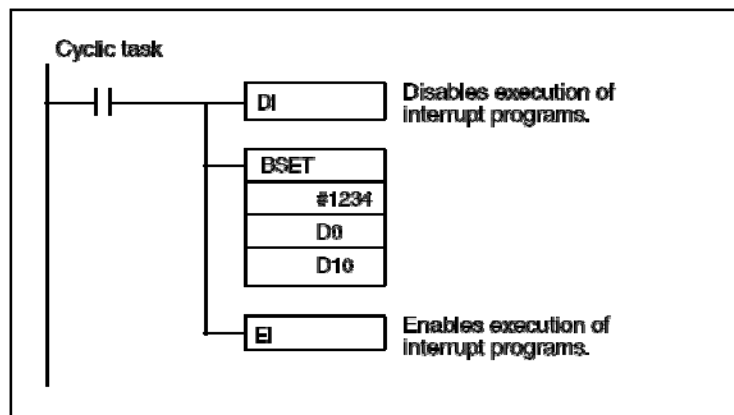


چگونگی پردازش:



در طول اجرای دستورالعمل BSET(071) قبل از نوشته شدن عدد #1234 در D10 و اجرای Task وقفه مقدار D0 با D10 برابر نبوده در نتیجه اگر Task وقفه در این میان پردازش گردد به دلیل برابر نبودن محتوای دو آدرس D0 و D10 خروجی A خاموش باقی می ماند (*1). در انتها (*2) با اینکه محتوای هر دو آدرس D0 و D10 توسط دستور BSET مساوی شده است ولی نتیجه درستی از این مقایسه توسط خروجی A به دست نیامده است.

جلوگیری از پردازش دوتایی



11-2-1- ورودی وقفه (حالت مستقیم)

این تابع یک Task وقفه را در صورتی که سیگنال ورودی متناظر با آن (لبه بالارونده و یا پایین رونده) دریافت شود اجرا می نماید.

11-2-1-1- بیت‌های ورودی وقفه و ترمینالهای مختص آن

شکل زیر بیت‌های مربوط به ورودی های وقفه و همچنین ترمینالهای مورد استفاده به عنوان ورودی وقفه را در واحد CPU به مدل CJ1M-CPU21,22,23 نمایش می دهد.

چهار بیت از آدرس CIO 2960.00 تا CIO2960.03 برای ورودی های وقفه به کار برده می شود.

تنظیمات توابع ورودی در قسمت تنظیمات PLC

به طور معمول بیت‌های CIO 2960.00 تا CIO2960.03 به عنوان ورودی های معمولی مورد استفاده قرار می گیرند. زمانی که از این ورودی های به عنوان ورودی وقفه استفاده می شود باید تنظیمات مربوط به این ورودی ها در قسمت تنظیمات نرم افزار CX-Programmer انجام شود.

شماره Task	تنظیمات ورودی	ترمینال ورودی	
		Bit	Word
Task وقفه 140	ورودی وقفه 0	00	CIO2960
Task وقفه 141	ورودی وقفه 1	01	
Task وقفه 142	ورودی وقفه 2	02	
Task وقفه 143	ورودی وقفه 3	03	

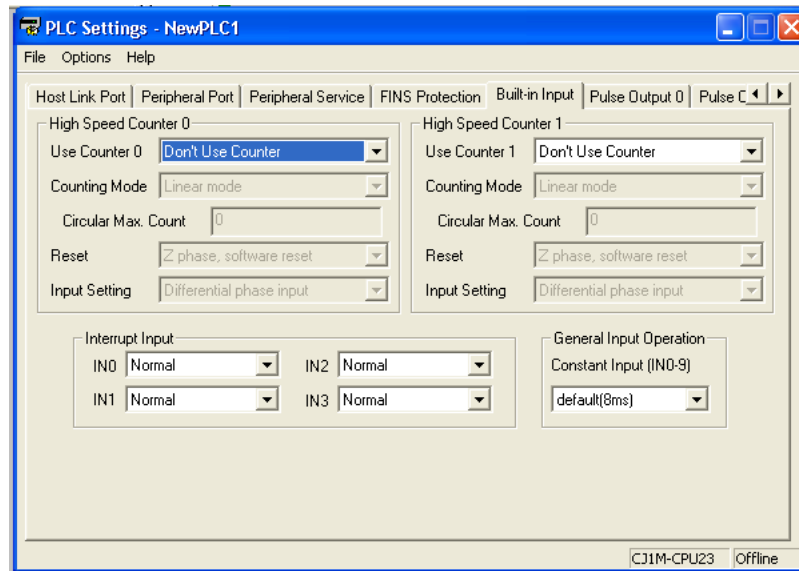
11-2-2- شیوه عملکرد

- مشخص نمودن ورودی های وقفه مورد استفاده و همچنین شماره Task متناظر با آن.
- سیم بندی ورودی ها
- استفاده از نرم افزار CX-Programmer برای انتخاب ورودی های وقفه در تنظیمات PLC.
- نوشتن برنامه در Task وقفه متناظر با ورودی وقفه.
- استفاده از دستور MSKS(690) برای مشخص کردن لبه بالارونده و یا پایین رونده.
- استفاده از دستور العمل MSKS(690) برای فعال کردن ورودی های وقفه (به حالت مستقیم).



11-2-3- تنظیمات PLC

برای مشاهده تنظیمات مربوط به ورودی های وقفه، در قسمت تنظیمات PLC در نرم افزار CX-Programmer باید Built-in Input را انتخاب نمود. با تغییر هر ورودی به ورودی وقفه در قسمت پایینی این پنجره آن ورودی به عنوان ورودی وقفه تلقی خواهد شد.

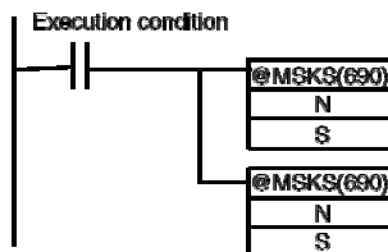


نکته: (1) تنظیمات ورودی مربوط به IN0 تا IN3 متناظر با ورودی های وقفه از شماره 0 تا 3 می باشد. (2) زمانی که یک ورودی به صورت معمول مورد استفاده قرار می گیرد نوع آن ورودی را باید Normal انتخاب شود.

11-2-4- نوشتن برنامه نردبانی

تنظیمات MSKS(690)

دستور MSKS(690) باید به ترتیب استفاده ورودی های وقفه اجرا شود. تنظیمات دستور MSKS(690) تنها با یکبار اجرا شدن آن فعال خواهد شد در نتیجه به طور کلی این دستورالعمل را با استفاده از شرط لبه بالارونده فقط به اندازه یک زمان دوره (Cycle Time) اجرا می نمایند. دستورالعمل MSKS(690) شامل دو تابع زیر می باشد که هر دو آنها با هم مورد استفاده قرار می گیرند. اگر ورودی وقفه ای به صورت لبه بالارونده مورد استفاده قرار گیرد، می توان دستورالعمل MSKS(690) ابتدایی را حذف نمود. چون ورودی ها پیش فرض به صورت لبه بالارونده در نظر گرفته می شوند.



1. تعیین کننده لبه بالارونده و یا پایین رونده ورودی وقفه.
2. فعال کردن و یا غیر فعال کردن ورودی وقفه.

عملکرد MSKS(690)

2. فعال کردن/غیرفعال کردن ورودی وقفه		1. لبه بالارونده و یا لبه پایین رونده		شماره Task وقفه	شماره ورودی وقفه
S	N	S	N		
غیرفعال / فعال	شماره ورودی وقفه	شرط عملکرد	شماره ورودی وقفه		
#0000 : فعال کردن وقفه	6	#0000 : لبه بالارونده #0001 : لبه پایین رونده	10	140	ورودی وقفه 0
	7		11	141	ورودی وقفه 1
غیرفعال کردن وقفه	8		12	142	ورودی وقفه 2
	9		13	143	ورودی وقفه 3

نوشتن برنامه Task وقفه

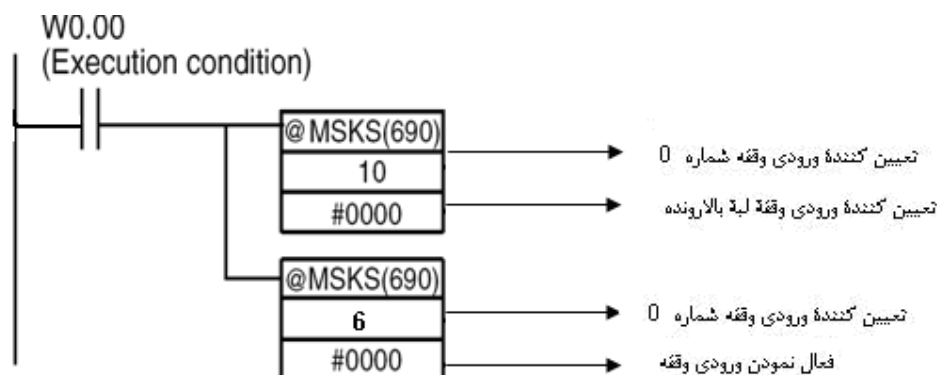
برنامه متناظر با هر ورودی وقفه در Task های وقفه شماره 140 تا 143 ایجاد می گردد همچنین باید توجه نمود که در انتهای برنامه های هر Task دستورالعمل (001)END را قرار داد.

11-2-5- تنظیمات ورودی های وقفه و عملکرد آنها

در این مثال چگونگی اجرای Task وقفه شماره 140 زمانی که ورودی CIO 2960.00 روشن می شود، نمایش داده شده است.

تنظیمات:

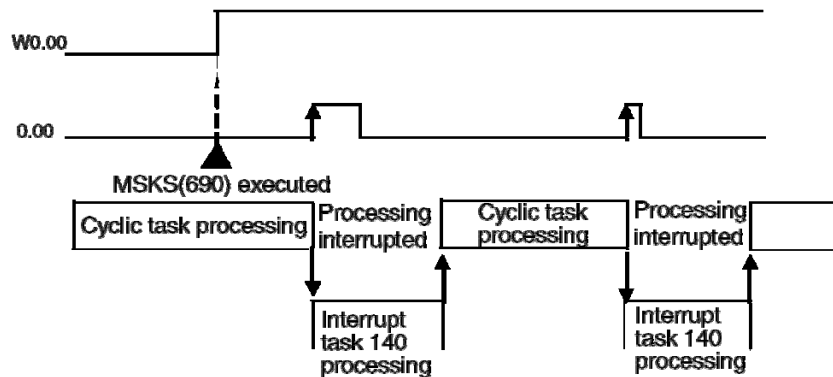
1. عنصر ورودی مورد نظر را به ورودی 2960.00 متصل نمایید.
2. با استفاده از نرم افزار CX-Programmer ورودی 2960.00 را به عنوان ورودی وقفه تنظیم نمایید.
3. با استفاده از نرم افزار CX-Programmer برنامه مربوط به Task وقفه متناظر با ورودی وقفه 2960.00 را ایجاد نمایید. (Task 140)
4. با استفاده از نرم افزار CX-Programmer دستور MSKS(690) را در برنامه بنویسید.



عملکرد

زمانی که شرط عملکرد W 0.00 روشن می شود، دستورالعمل MSKS(690) برای فعال کردن ورودی وقفه CIO 2960.00 به صورت لبه بالارونده اجرا می گردد.

اگر CIO2960.00 از حالت خاموش به روشن (لبه بالارونده) تغییر وضعیت دهد، پردازش Task سیکلی که در حال اجرا می باشد متوقف شده و پردازش Task وقفه با شماره 140 آغاز می گردد. زمانی که پردازش Task وقفه به پایان میرسد، پردازش Task سیکلی متوقف شده ادامه می یابد.



توجه: ورودی هایی که به عنوان ورودی معمولی (Normal Input) و یا ورودی های واکنش-سریع (Quick-Response Input) مورد استفاده قرار گرفته اند را نمی توان به صورت ورودی وقفه استفاده نمود.

11-3- ورودی وقفه (حالت شمارشی)

11-3-1- نگاه کلی

این تابع لبه بالارونده یا پایین رونده سیگنالهای ورودی را شمرده و زمانی که این شمارش به مقدار از پیش تعیین شده (Set Value) برسد، Task وقفه مربوطه را اجرا می نماید.

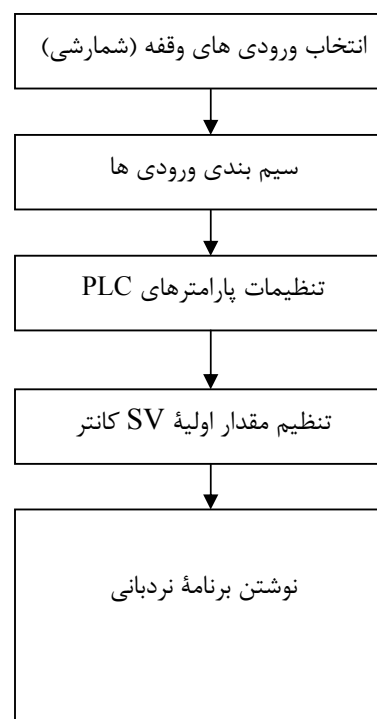
- ورودی های وقفه (شمارشی) از همان ترمینالهای مربوط به ورودی وقفه (مستقیم) استفاده می نماید.
- شمارنده ورودی وقفه می تواند به صورت افزایشی و یا کاهش با استفاده از دستورالعمل MSKS(690) شمارش را انجام دهد.
- ورودی های وقفه (شمارشی) از همان Task های ورودی های وقفه مستقیم (Task شماره 140 تا 143) استفاده می نمایند.
- حداکثر فرکانس پاسخ برای این ورودی ها در حالت شمارشی 5kHz می باشد.

رابطه بین بیت‌های ورودی، شماره Task و شمارنده

Word های شمارنده		توابع		بیت‌های ورودی
PV	SV	شماره Task وقفه	شماره ورودی وقفه	واحد CPU
A536	A532	140	ورودی وقفه 0	2960.00
A537	A533	141	ورودی وقفه 1	2960.01
A538	A534	142	ورودی وقفه 2	2960.02
A539	A535	143	ورودی وقفه 3	2960.03

عملکرد

- مشخص نمودن ورودی های وقفه مورد استفاده و همچنین شماره Task متناظر با آن.
- سیم بندی ورودی های وقفه.
- استفاده از نرم افزار CX-Programmer برای انتخاب ورودی های وقفه در تنظیمات PLC.
- تنظیم مقدار SV شمارنده در ناحیه AR متناظر با آن شمارنده.
- نوشتن برنامه در Task وقفه متناظر با ورودی وقفه.
- استفاده از دستور MSKS(690) برای مشخص کردن لبه بالارونده و یا پایین رونده. استفاده از دستور العمل MSKS(690) برای فعال کردن ورودی های وقفه (شمارشی).



نکته: تابع ورودی وقفه (شمارشی) یکی از توابع ورودی های وقفه می باشد که یک وقفه را بر پایه شمارش پالس اجرا می نماید. در صورتی که فرکانس پالس ورودی زیاد باشد اجرای وقفه به دفعات تکرار خواهد شد که از اجرای معمول Task های سیکلی جلوگیری خواهد نمود. در این حالت یا زمان دوره PLC زیاد شده که باعث ایجاد خطا میگردد و یا پالسهای ورودی با فرکانس بالا خوانده نمی شوند.

حداکثر فرکانس پالس ورودی در حالت شمارشی 1kHz می باشد. حتی ممکن است در مواردی که از پالس ورودی با حداکثر فرکانس استفاده می شود نیز عملکرد سیستم تحت تأثیر قرار گیرد. بنابراین باید صحت عملکرد سیستم قبل از استفاده از پالسهای ورودی با فرکانس حداکثر بررسی گردد.

11-3-2- تنظیمات PLC

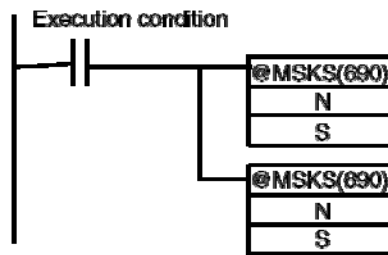
شیوه عملکرد برای استفاده از نرم افزار CX-Programmer جهت انجام تنظیمات ورودی های وقفه (حالت شمارشی) مشابه با انجام تنظیمات ورودی های وقفه (حالت مستقیم) می باشد.

11-3-3- نوشتن برنامه نردبانی

تنظیمات MSKS(690)

دستور MSKS(690) باید به ترتیب استفاده ورودی های وقفه اجرا شود. تنظیمات دستور MSKS(690) تنها با یکبار اجرا شدن آن فعال خواهد شد در نتیجه به طور کلی این دستورالعمل را با استفاده از شرط لبه بالارونده فقط به اندازه یک زمان دوره (Cycle Time) اجرا می نمایند.

دستورالعمل MSKS(690) شامل دو تابع زیر می باشد که هر دو آنها با هم مورد استفاده قرار می گیرند. اگر ورودی وقفه ای به صورت لبه بالارونده مورد استفاده قرار گیرد، می توان دستورالعمل MSKS(690) ابتدایی را حذف نمود. چون ورودی ها پیش فرض به صورت لبه بالارونده در نظر گرفته می شوند.



1. تعیین کننده لبه بالارونده و یا پایین رونده ورودی وقفه.

2. فعال کردن و یا غیر فعال کردن ورودی وقفه.

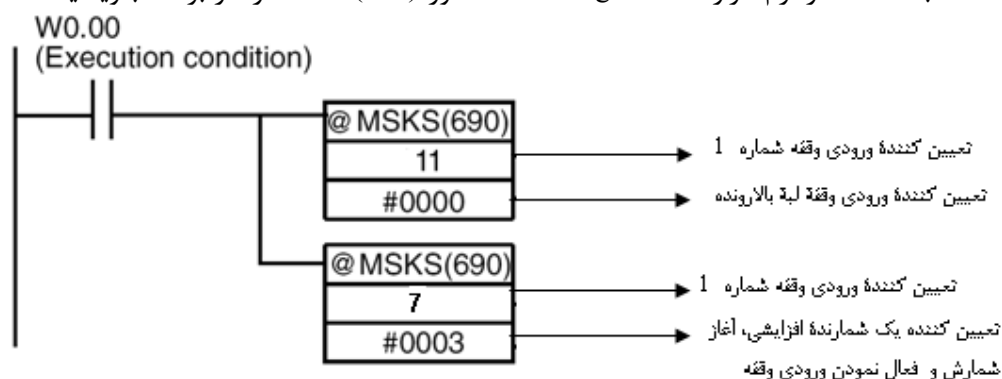
عملکرد MSKS(690)

2- فعال کردن / غیرفعال کردن ورودی وقفه		1- لبه بالا رونده یا پایین رونده		شماره Task وقفه	شماره ورودی وقفه
S	N	S	N		
فعال / غیرفعال	شماره ورودی وقفه	شرط عملکرد	شماره ورودی وقفه	140	ورودی وقفه 0
: #0000 فعال کردن وقفه	6	: #0000 لبه بالا رونده	10	141	ورودی وقفه 1
: #0001 غیر فعال کردن وقفه	7	: #0001 لبه پایین رونده	11	142	ورودی وقفه 2
	8		12	143	ورودی وقفه 3
	9		13		

11-3-4- تنظیمات ورودی های وقفه و عملکرد آنها

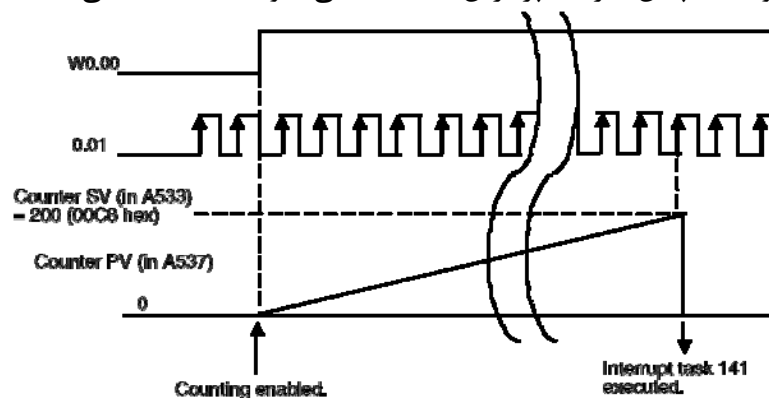
در این مثال چگونگی اجرای Task وقفه با شماره 141 زمانی که 200 بار پالسهای ایجاد شده توسط ورودی CIO 2960.01 شمرده شود، نمایش داده شده است. (شمارنده به صورت افزایشی کار می کند).
تنظیمات:

1. عنصر ورودی مورد نظر را به ورودی 2960.01 متصل نمایید.
2. با استفاده از نرم افزار CX-Programmer ورودی 2960.01 را به عنوان ورودی وقفه تنظیم نمایید.
3. با استفاده از نرم افزار CX-Programmer برنامه مربوط به Task وقفه متناظر با ورودی وقفه 2960.01 را ایجاد نمایید. (Task 141)
4. با استفاده از نرم افزار CX-Programmer مقدار SV کانتر را در آدرس A533 مقدار #00C8 Hex (معادل 200 دسیمال) قرار دهید.
5. با استفاده از نرم افزار CX-Programmer دستور MSKS(690) را در برنامه بنویسید.



عملکرد

زمانی که شرط عملکرد W0.00 روشن می شود، دستورالعمل MSKS(690) برای فعال کردن ورودی وقفه در حالت شمارشی اجرا می شود.
اگر CIO 2960.01 ، برای 200 مرتبه از حالت خاموش به روشن (لبه بالارونده) تغییر وضعیت دهد، پردازش Task سیکلی که در حال اجرا می باشد متوقف شده و پردازش Task وقفه با شماره 141 آغاز می گردد. زمانی که پردازش Task وقفه به پایان میرسد، پردازش Task سیکلی متوقف شده ادامه می یابد.

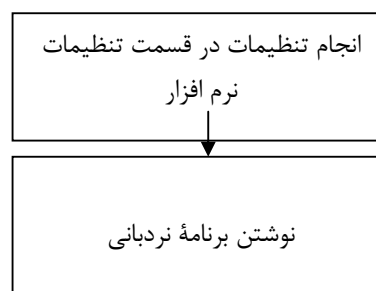


11-4-4- وقفه های برنامه ریزی شده

این تابع Task وقفه برنامه ریزی شده را در فاصله های زمانی مشخص اجرا می نماید. فاصله های زمانی توسط ساعت موجود در CPU محاسبه می شود. Task وقفه شماره 2 به وقفه های برنامه ریزی شده اختصاص داده شده است.

11-4-1- عملکرد

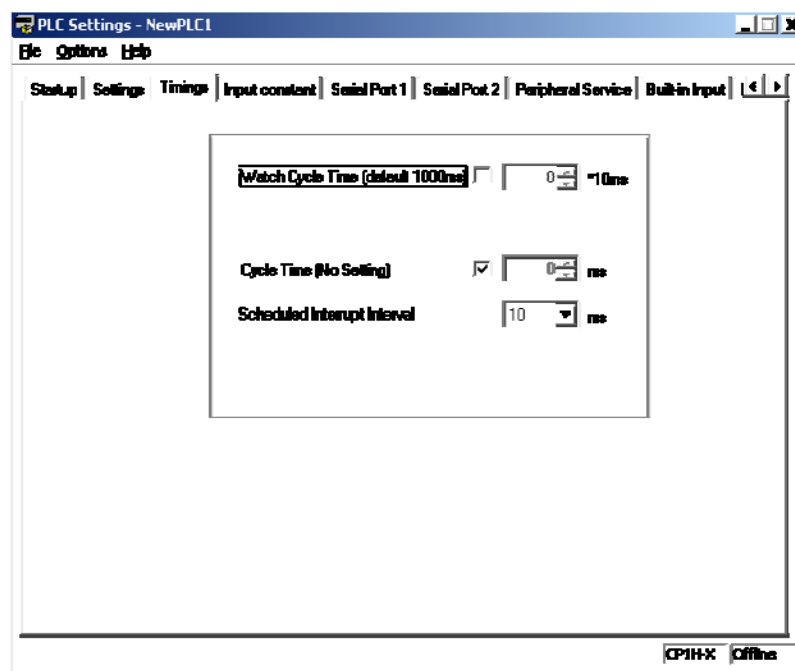
- انجام تنظیمات مربوط به زمان Task های وقفه برنامه ریزی شده را در قسمت تنظیمات نرم افزار CX-Programmer .
- نوشتن برنامه مربوط به Task وقفه برنامه ریزی شده را در TASK وقفه شماره 2. (Scheduled Interrupt Task)
- استفاده از دستور MSKS(690) برای تعیین مقدار SV تایمر.



11-4-2- تنظیمات PLC

در قسمت Setting نرم افزار CX-Programmer و در بخش *Timing* گزینه *Scheduled Interrupt Interval* فاصله زمانی انجام Task وقفه برنامه ریزی شده معین می گردد. این زمان را می توان 10ms، 1ms و یا 0.1ms تعیین نمود. فاصله زمانی برای اجرای Task وقفه حاصل ضرب SV در دستور MSKS(690) در زمان تعیین شده در تنظیمات می باشد.

تنظیمات مربوط به فاصله زمانی Task های وقفه برنامه ریزی شده

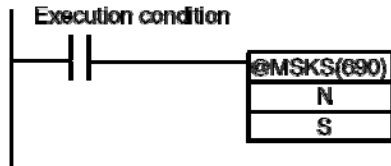


- نکته:**
- 1) باید فاصله زمانی Task وقفه را بیشتر از زمان مورد نیاز برای اجرای Task وقفه انتخاب نمود.
 - 2) اگر فاصله زمانی تنظیم شده برای Task وقفه کوتاه باشد این Task به دفعات اجرا می گردد، این امر باعث افزایش زمان دوره گشته و در نتیجه در پردازش Task سیکلی تاثیر خواهد گذاشت.
 - 3) اگر Task وقفه دیگری (از نوع ورودی وقفه، وقفه های خارجی و ...) در حال اجرا باشد و Task وقفه برنامه ریزی شده روی دهد در این صورت این Task تا کامل نشدن پردازش Task های وقفه دیگر اجرا نخواهد شد.
- وقتی در برنامه ای از Task های وقفه مختلف استفاده می شود باید برنامه را طوری طراحی نمود که تا حد ممکن از تداخل Task ها جلوگیری نمود. حتی اگر دو وقفه مختلف در یک زمان روی دهد، Task برنامه ریزی شده در ادامه برنامه باقی خواهد ماند و هر بار در فاصله زمانی مشخص فراخوانی خواهد شد حتی اگر در اجرای Task برنامه ریزی قبلی تأخیر ایجاد شده باشد.

11-4-3- نوشتن برنامه نردبانی

تنظیمات (MSKS(690)

دستور MSKS(690) باید برای استفاده از Task های وقفه برنامه ریزی شده به صورت صحیح مورد استفاده قرار گیرد. تنظیمات مورد نظر توسط MSKS(690) با یکبار فعال کردن آن انجام می شود به همین علت به عنوان شرط عملکرد این دستور کافی است از یک لبه بالا رونده استفاده نمود.



تعیین کننده Task برنامه ریزی شده شماره 0 (Task وقفه شماره 2) تعیین کننده فاصله زمانی اجرای Task برنامه ریزی شده

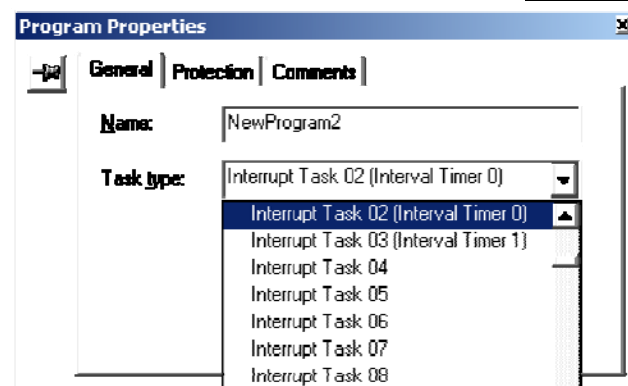
عملکرد (MSKS(690)

فاصله زمانی وقفه		عملگر	
فاصله زمانی برنامه ریزی شده	واحد زمانی تنظیم شده در تنظیمات PLC	S	N
		زمان وقفه	شماره Task وقفه برنامه ریزی شده
10 تا 99,990 میلی ثانیه	10ms	#0000 تا #270F (0000 تا 9999)	وقفه برنامه ریزی شده شماره 0 (Task وقفه شماره 2) Reset Start :14 Start without Reset 4
1 تا 9,999 میلی ثانیه	1ms		وقفه برنامه ریزی شده شماره 1 (Task وقفه شماره 3) Reset Start :15 Start without Reset 5
0.5 تا 999.9 میلی ثانیه	0.1ms		

نوشتن برنامه Task وقفه

برنامه وقفه برنامه ریزی شده در Task وقفه شماره 2 (وقفه برنامه ریزی شده شماره 0) که در فاصله زمانی مشخص اجرا می گردد، ایجاد می شود. همچنین باید توجه نمود که در انتهای برنامه دستورالعمل END(001) را قرار داد.

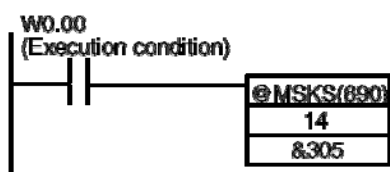
انتخاب Task وقفه برنامه ریزی شده



11-4-4- تنظیمات ورودی های وقفه و عملکرد آنها

در این مثال چگونگی اجرای Task وقفه در فاصله زمانی مشخص 30.5 ms نشان داده شده است. تنظیمات:

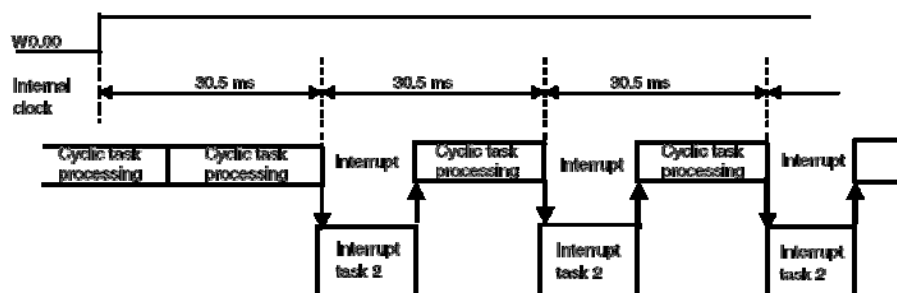
1. با استفاده از نرم افزار CX-Programmer واحد زمانی وقفه برنامه ریزی شده را 0.1ms انتخاب می شود.
2. با استفاده از نرم افزار CX-Programmer برنامه وقفه مربوط به Task وقفه شماره 2 نوشته می شود.



تعیین کننده Task برنامه ریزی شده شماره 0 (Reset Start) تنظیم فاصله زمانی 30.5 میلی ثانیه
 $(305 * 0.1 = 30.5)$ میلی ثانیه

عملکرد

با روشن شدن شرط عملکرد W0.00 دستورالعمل MSKS(690) برای فعال نمودن وقفه برنامه ریزی شده اجرا می گردد. تایمر Reset شده و محاسبه زمان آغاز می گردد. در نتیجه Task برنامه ریزی شده شماره 2 هر 30.5 میلی ثانیه اجرا می گردد.



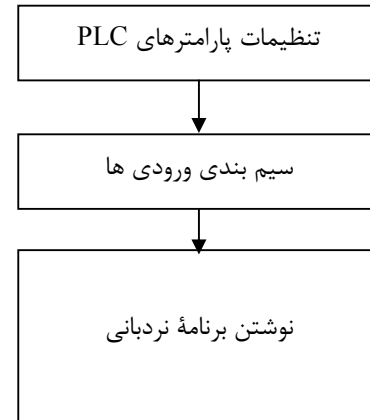
11-6- وقفه های شمارنده های پرسرعت

این تابع Task وقفه مشخصی (0 تا 255) را زمانی که ارزش فعلی (PV) شمارنده های پرسرعت موجود روی PLC و یا CPU برابر با مقدار از پیش تعیین شده (یک مقدار ثابت) باشد و یا در یک محدوده تعیین شده قرار بگیرد (مقایسه محدوده)، اجرا می نماید.

- دستور CTBL(882) برای تعریف جدول مقایسه استفاده می شود.
- هر دو دستور CTBL(882) و INI(880) برای آغاز مقایسه به کار برده می شوند.
- دستور INI(880) برای توقف مقایسه مورد استفاده قرار می گیرد.

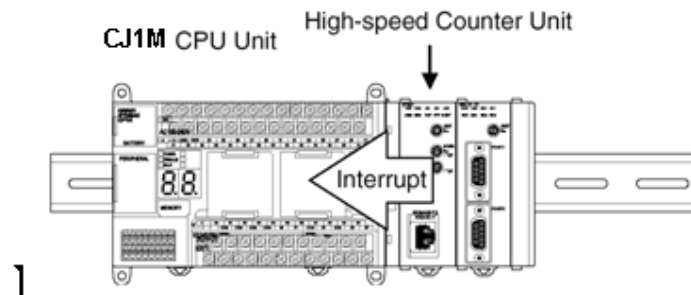
11-6-1- عملکرد

- انجام تنظیمات مربوط به شمارنده های پرسرعت با استفاده از نرم افزار CX-Programmer.
- انجام سیم بندی ورودی های پرسرعت.
- نوشتن برنامه مربوط به Task وقفه.
- استفاده از CTBL(882) جهت تعیین شماره شمارنده پرسرعت و جدول مقایسه همچنین ایجاد جدول مقایسه.



11-7- وقفه های خارجی

یک وقفه خارجی باعث پردازش یک وقفه توسط CPU، در جواب یکی از ورودی های وقفه موجود در CPU Bus می باشد. دریافت این وقفه ها توسط واحد CPU همواره فعال می باشد. وقفه های خارجی نیاز به تنظیمات خاصی در CPU ندارند، همچنین با استفاده از برنامه می توان هر یک از Task های وقفه را به این ورودی های وقفه خارجی اختصاص داد.



نکته: زمانی که یک وقفه هم به عنوان Task وقفه خارجی (0 تا 255) و هم به عنوان یک Task برنامه ریزی شده (2) و یا به عنوان یک وقفه ورودی پرسرعت (0 تا 255) به کار برده شود، این Task علاوه بر این که در صورت وقوع وقفه های خارجی اجرا می گردد در سایر وقفه ها نیز اجرا خواهد شد. ولی به طور کلی استفاده از Task وقفه با یک شماره برای وقفه های مختلف پیشنهاد نمی شود.

11-8- وقفه قطع برق:

این تابع Task وقفه مربوط به قطع برق را اجرا می کند.

11-8-1- شرح عملکرد:

برنامه مورد نظر در Task مربوط به این وقفه (Task1) نوشته میشود. بعد از خاموش شدن PLC این برنامه اجرا می شود.

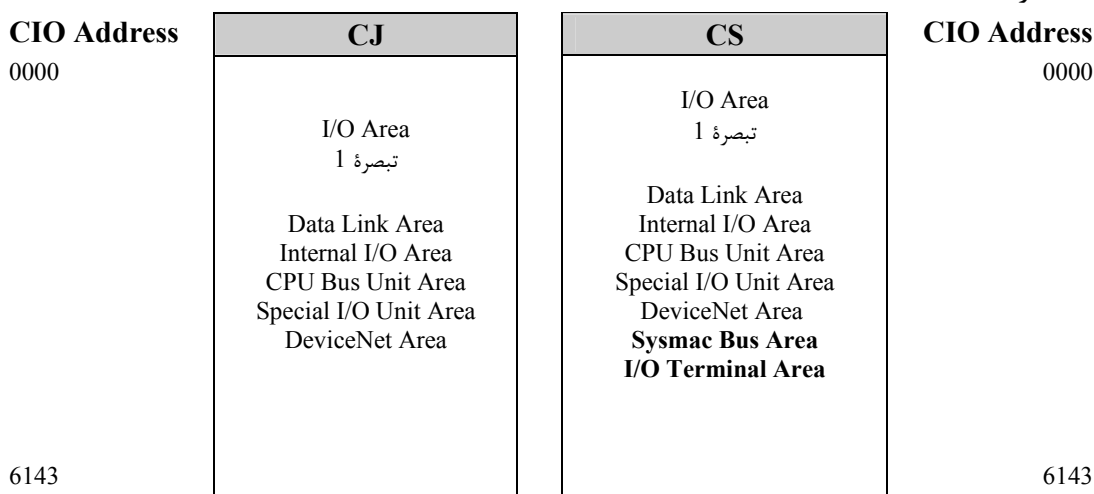
12- مروری دوباره بر فضای حافظه (یادآوری)

1-12-1- تعریف فضای داده ها

در زیر فضای داده ها در حافظه ورودی/خروجی به تفصیل توضیح داده خواهد شد.

12-1-1- ناحیه CIO (Common Input/Output Area)

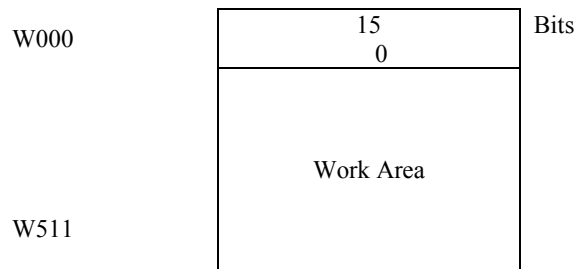
هنگام وارد کردن یک آدرس از منطقه CIO حافظه PLC نیازی به وارد کردن حروف اختصاری CIO در ابتدای آدرس نمی باشد. منطقه حافظه CIO عموماً برای تبادل و دوباره سازی (Refreshing) اطلاعات بین PLC و بخشهای ورودی و خروجی استفاده می شود. منطقه CIO از 6144Word از 0000 تا 6143 (از بیت 0000.00 تا 6143.15) تشکیل شده است. در این منطقه از حافظه Word هایی که به بخشهای ورودی و خروجی اختصاص داده نشده باشند را می توان در برنامه به عنوان Word ها و بیت های کاربردی و مجازی استفاده کرد.



- می توان با انجام تنظیمات خاص و تعیین آدرس ابتدایی هر Rack از PLC از آدرسهای CIO0000 تا CIO0999 برای بخشهای ورودی و خروجی استفاده کرد. این تنظیمات برای مشخص کردن آدرس ابتدایی هر Rack توسط نرم افزار CX-Programmer و در قسمت جدول ورودی/خروجی (I/O Table) انجام می گیرد. برای تعیین آدرس ابتدایی اولین Rack می توان از CIO0000 تا CIO0900 استفاده کرد.
- بخشی از منطقه حافظه CIO که "بدون کاربرد" می باشد را می توان به عنوان حافظه کار در برنامه نویسی استفاده کرد. البته در آینده با اضافه شدن توابع جدید به PLC ها بیت هایی از منطقه CIO که هم اکنون بدون استفاده است نیز بکار گرفته خواهند شد، به همین دلیل در ابتدا استفاده از بیتهای منطقه W(Work Area) همواره توصیه می شود.

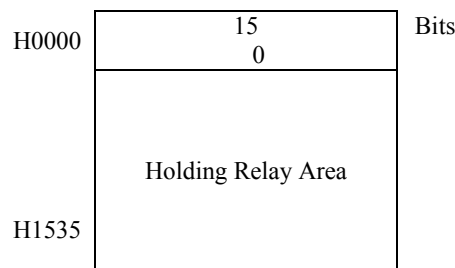
12-1-2- ناحیه W (Work Area)

زمانی که از آدرسهای منطقه کار PLC استفاده می شود باید قبل از آدرس حرف "W" را برای مشخص کردن منطقه حافظه قرار داد. منطقه 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 حافظه PLC ها اختصاص داده خواهد شد به همین دلیل استفاده از این منطقه به عنوان حافظه مجازی قبل از استفاده از بیتهای بدون کاربرد منطقه CIO توصیه می شود. اگر این حافظه برای برنامه نویسی کافی نبود می توان از حافظه CIO (CIO1200 تا CIO1499 و همچنین CIO3800 تا CIO6143) استفاده کرد.



12-1-3- ناحیه H (Holding Area)

زمانی که از آدرسهای منطقه Holding استفاده می شود باید قبل از آدرس حرف "H" را برای مشخص کردن این منطقه 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 از مد Program به مد Run و یا مد Monitor حفظ خواهد شد. محتوای این منطقه در صورت وجود باتری حفظ می شود. منطقه Holding Area از 1536Word که از H000 تا H1535 (بیتهای H000.00 تا H1535.15) می باشد، تشکیل شده است. می توان از این منطقه به هر صورت مورد نیاز، به صورت کنتاکتهای بسته و باز در برنامه استفاده کرد.

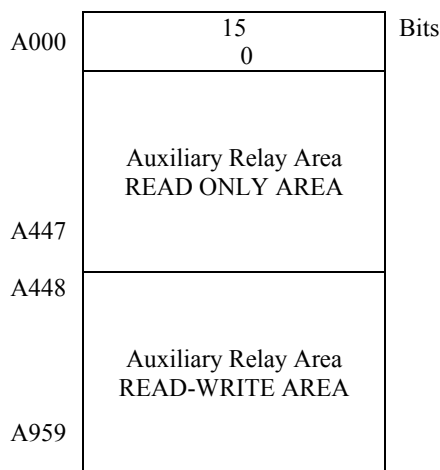


12-1-4- ناحیه A (Auxiliary Area)

زمانی که از آدرسهای منطقه Auxiliary استفاده می شود باید قبل از آدرس حرف "A" را برای مشخص کردن این منطقه حافظه قرار داد. منطقه Auxiliary شامل فلگهای از پیش تعیین شده و بیتهای کنترلی است که برای نشان دادن سلامت CPU، و ماژول های ورودی و خروجی خاص و همچنین مسیر ارتباطی CPU و باتری به کار می روند.

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

CJ	CS
آدرسهای فقط خواندنی در منطقه Auxiliary قابل کنترل به صورت مستقیم توسط برنامه نیستند.	آدرسهای فقط خواندنی در منطقه Auxiliary قابل کنترل به صورت مستقیم توسط برنامه و یا اختصاص دادن به ماژول های ویژه C200H نیستند.



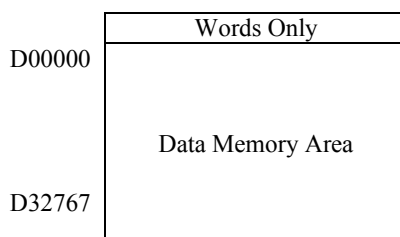
12-1-5- ناحیه D (Data Memory Area)

زمانی که از آدرسهای منطقه Data Memory استفاده می شود باید قبل از آدرس حرف "D" را برای مشخص کردن این منطقه 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 می باشد، استفاده می شود. منطقه D معمولا برای ذخیره کردن اطلاعات همچون مقادیر ثابت، مقادیر ثبت شده 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.

ولی این بیتها را نمی توان با جبر (Force) روشن و یا خاموش کرد.



12-1-6- ناحیه E (Extended Data Memory Area)

زمانی که از آدرسهای منطقه Extended Data Memory استفاده می شود باید قبل از آدرس حرف "E" را برای مشخص کردن این منطقه 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 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.

این منطقه از بانکهای حافظه تشکیل شده است، که هر کدام از این بانکها شامل 32,768Word می باشد. حافظه 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 می باشد، 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.

منطقه E معمولا برای ذخیره کردن اطلاعات همچون مقادیر ثابت، مقادیر ثبت شده 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.

ولی این بیتها را نمی توان با جبر (Force) روشن و یا خاموش کرد.

مشخص کردن آدرسهای ناحیه E

برای مشخص کردن آدرسهای منطقه E دو روش وجود دارد: مشخص کردن آدرس و همچنین بانک در یک زمان و یا مشخص کردن آدرس در بانک جاری. 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

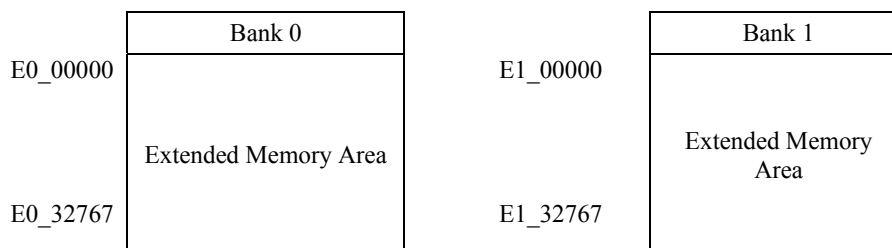
در این روش شماره بانک مورد نظر باید بعد از "E" ذکر شود، مانند E2_00010 که مشخص کننده آدرس 00010 از بانک 2 در منطقه E می باشد.

2. مشخص کردن آدرس در همان بانک جاری

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

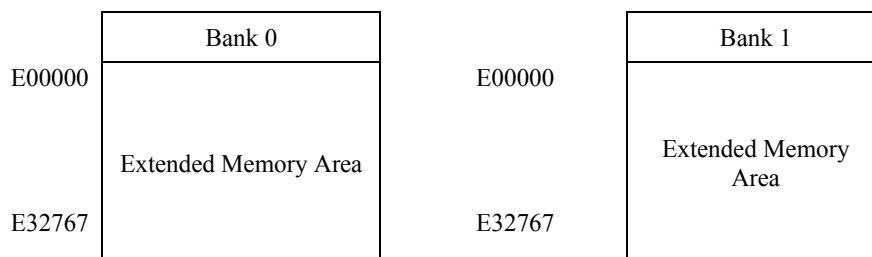
بانک جاری زمانی که مد PLC از مد Program به مد Run/Monitor تغییر کند به 0 تبدیل می شود مگر اینکه بیت A500.12 که بیت نگهدارنده حافظه IO می باشد روشن باشد.

مشخص کردن بانک و آدرس مورد نظر در آن بانک



مشخص کردن آدرس در بانک جاری

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



12-1-7- ناحیه تایمر (Timer Area)

دو ناحیه برای اطلاعات تایمر وجود دارد: فلگ اتمام کار تایمرها (Completion Flag) و همچنین مقدار حال حاضر تایمرها (Present Value). هنگام مشخص کردن فلگ اتمام کار تایمرها باید حرف "T" را قبل از آدرس قرار داد. 4096 تایمر با شماره تایمر T0000 تا T4095 وجود دارد. که این 4096 تایمر (T0000 تا T4095) برای تمامی دستورات تایمر است، که شامل دستورات TIM، TIMH(015)، TMHH(540)، TTIM(087)، TIMW(813) و TMHW(815) می باشد. برای این دستورات عملها فلگ اتمام کار و همچنین مقدار حال حاضر تایمر با شماره تایمر مشخص می گردد. (دستورهای TIML(542) و MTIM(543) از این شماره تایمرها استفاده نمی کنند).

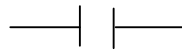
زمانیکه شماره تایمر در دستوری با عملگر بیتی به عنوان عملگر استفاده می شود، این داده به عنوان فلگ اتمام کار تایمر در نظر گرفته می شود. ولی زمانیکه شماره تایمر در دستوری با عملگر Word استفاده شود، این داده به عنوان مقدار حال حاضر تایمر که خود داده ای با فرمت Word است، در نظر گرفته می شود. فلگ اتمام کار تایمر را می توان به دفعات مورد نیاز در برنامه به صورت کنتاکت باز و یا بسته استفاده کرد و همچنین مقدار حال حاضر تایمر را هم می توان به صورت یک داده Word به دفعات در دستورات عملهای مختلف استفاده کرد.

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

فلگ اتمام کار تایمر

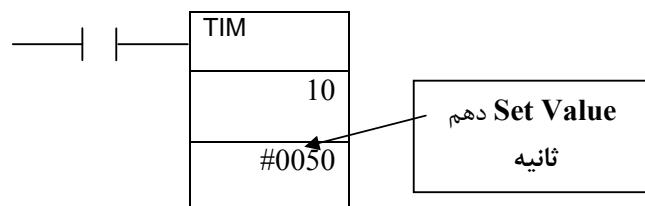
این فلگ به صورت بیتی فراخوانده می شود. این فلگ زمانی که زمان مشخص شده برای تایمر به پایان رسید و یا به عبارتی زمانی که کار تایمر به اتمام رسید روشن خواهد شد.

T0010



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 دهنده زمان باقی مانده تایمر می باشد. تایمرها برای ایجاد فاصله زمانی قبل از اجرای مرحله بعدی برنامه به کار می روند. تا این امکان را به وجود آورند که قبل از شروع مرحله بعدی مرحله قبلی به پایان رسیده باشد.



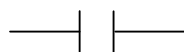
8-1-12- ناحیه کانتر (Counter Area)

دو ناحیه برای اطلاعات کانتر وجود دارد: فلگ اتمام کار کانترها (Completion Flag) و همچنین مقدار حال حاضر 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 این 4096 کانتر (C0000 تا C4095) برای تمامی دستورات کانتر است، که شامل دستورات CNT، CNTR(012) و CNTW(814) می باشد. برای این دستورات عملها فلگ اتمام کار و همچنین مقدار حال حاضر کانتر با شماره کانتر مشخص می گردد. زمانیکه شماره کانتر در دستوری با عملگر بییتی به عنوان عملگر استفاده می شود، این داده به عنوان فلگ اتمام کار کانتر در نظر گرفته می شود. ولی زمانیکه شماره کانتر در دستوری با عملگر Word استفاده شود، این داده به عنوان مقدار حال حاضر کانتر که خود داده ای با فرمت Word است، در نظر گرفته می شود. استفاده از دو کانتر مختلف با یک شماره کانتر در برنامه به هیچ عنوان توصیه نمی گردد، این امر باعث عملکرد نادرست کانتر هنگام عملکرد همزمان آنان خواهد شد. (اگر از دو یا چند کانتر مختلف با یک شماره کانتر در برنامه استفاده شود در هنگام بررسی برنامه توسط نرم افزار به عنوان خطا در نظر گرفته می شود).

فلگ اتمام کار کانتر

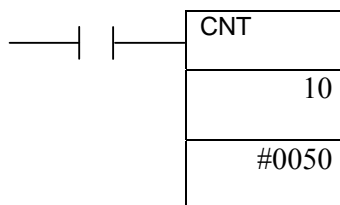
این فلگ به صورت بییتی فراخوانده می شود. این فلگ زمانی که شمارش تعداد پالسهای مشخص شده برای کانتر به پایان رسید و یا به عبارتی زمانی که کار کانتر به اتمام رسید روشن خواهد شد.

C0010



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.



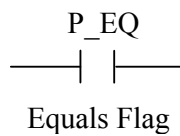
9-1-12- فلگهای وضعیت

این فلگها شامل فلگهای محاسباتی مانند فلگ خطا (Error Flag) و فلگ مساوی (Equal Flag) می باشد، که نتیجه 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 منطقه 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 استفاده می شود. حالت این فلگها از نتیجه اجرا دستورات در برنامه تأثیر می پذیرد و این فلگها دارای دسترسی فقط خواندنی هستند و نمی توان آنها را مستقیماً با استفاده از دستور خاصی فرمان داد.

حالت تمام فلگهای وضعیت، زمانی که برنامه بین Task های مختلف سوییچ می کند باز نشانده می شود. در نتیجه حالت فلگهای مانند ER و AER فقط در همان Task که استفاده می شود برقرار است. همچنین این فلگها را نمی توان با استفاده از جبر (Force) روشن و خاموش کرد.

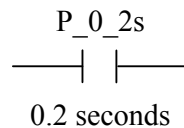


در CX-Programmer فلگهای وضعیت نیز مانند بقیه نمادها با "P_" آغاز می شوند.

12-1-10- پالسهای زمانی

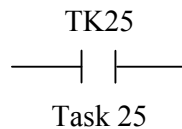
پالسهای زمانی توسط تایمر داخلی CPU روشن و خاموش می شود. این بیتها را با استفاده از نمادها به جای آدرس در برنامه استفاده می کنند.

پالسهای زمانی فقط خواندنی هستند، این پالسها را نمی توان با استفاده از دستورات فرمان داد. این پالسها همگی در شروع عملکرد PLC بازنشانده می شوند.



12-1-11- ناحیه TK (Task Flag Area)

این ناحیه از حافظه از TK00 تا TK31 است که مربوط به Task های سیکلی می باشد. این فلگ در صورتی که Task مورد نظر با همان شماره قابل اجرا و یا در حالت Run باشد روشن و در صورتی که Task مورد نظر قابل اجرا نباشد و یا در حالت انتظار باشد خاموش می شود.



این فلگها فقط نشان دهنده حالت Task های سیکلی بوده و از Task های وقفه ای تأثیر نمی پذیرد.

12-1-12- ناحیه IR (Index Register)

این 16 ثبات شاخص (IR00 تا IR15) برای ذخیره کردن آدرسهای حافظه PLC برای آدرس دهی غیر مستقیم حافظه به کار برده می شود (آدرسهای مطلق حافظه در RAM). هر کدام از ثباتهای شاخص قابلیت حفظ یکی از آدرس های PLC را دارند که این آدرس ها به حافظه های مطلق PLC اشاره می کنند. این منطقه از حافظه را می توان به صورت جداگانه در یک Task و یا، برای PLC های مدل CJ1H در همه Task ها استفاده کرد.

12-1-13- ناحیه DR (Data Register)

منطقه ثبات داده از حافظه (DR00 تا DR15) با منطقه IR با هم استفاده می شوند. وقتی داده ای در حافظه DR درست قبل از وارد کردن داده ای در حافظه IR با همان شماره وارد شود، محتوای DR به محتوای IR برای مشخص کردن آدرس مورد نظر در PLC اضافه می شود. از آنجایی که محتوای حافظه DR به صورت باینری علامت دار است، لذا محتوای IR می تواند حافظه می تواند به صورت جداگانه در هر Task به کار برده شود. به آدرسهای بالاتر و یا پایین تر تبدیل شود. این منطقه از برای وارد کردن داده های مورد نیاز در منطقه DR کار برده شود و یا برای به نویسی استفاده کرد.

بیت های این منطقه از حافظه را نمی توان با جبر می توان از دستورالعملهای معمول برنامه

12-2- نقشه حافظه

(Force) روشن و یا خاموش کرد

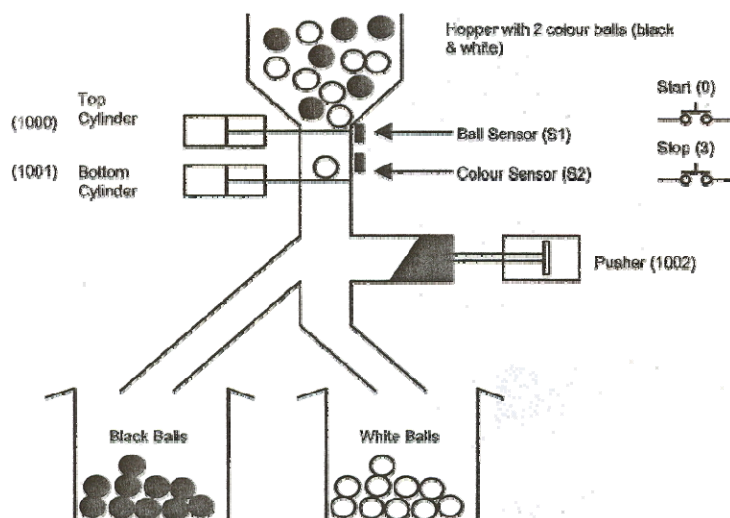
TK00 – TK31	Task Flag Area
A000 – A959	Auxiliary Area
T0000 – T4095	Timer Completion Flags
C0000 – C4095	Counter Completion Flags
CIO 0000 – CIO 6143	Core I/O Area
H000 – H511	Holding Area
W000 – W511	Work Area
T0000 – T4095	Timer PVs
C0000 – C4095	Counter PVs
D00000 – D32767	Data Memory Area
E0_00000 – En32767	Banks of Extended Memory



13- مثالهای کاربردی

13-1- مکانیزم دسته بندی توپ ها

در این مثال، سیستم توپهای مشکی و سفید را در دو محفظهٔ مختلف دسته بندی می نماید. کلید Start برای شروع عملکرد سیستم می باشد، سنسور توپ (Ball Sensor) وجود توپ در مخزن را حس می کند. سولنوئید بالایی توپ را برای تشخیص رنگ توسط سنسور رنگ (Colour Sensor) رها می کند، در این مرحله قبل از باز شدن سولنوئید پایین و افتادن توپ در محفظهٔ مورد نظر رنگ آن معین می گردد.

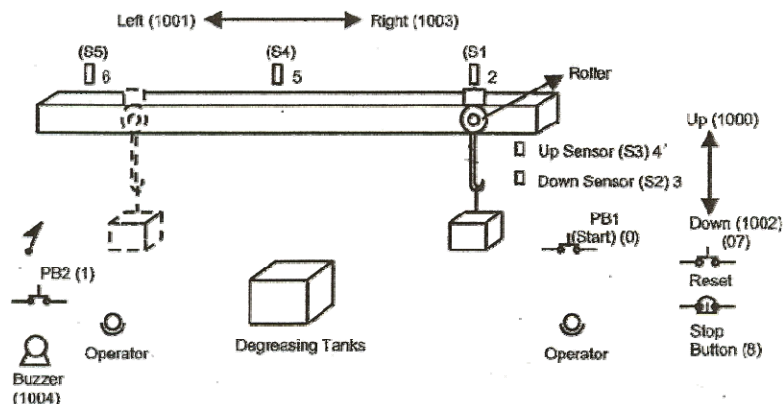


ورودی	عنصر
0.00	شستی Start
0.01	سنسور توپ (S1)
0.02	سنسور رنگ (S2)
0.03	شستی Stop

خروجی	عنصر
1.00	سیلندر بالایی
1.01	سیلندر پایینی
1.02	Pusher

13-2- کنترل جرثقیل سقفی برای عملکرد روغن زدایی

در این مثال قطعه مورد نظر باید قبل از فرستاده شدن به مرحله بعدی در تانک مخصوص روغن زدایی گردد.



زمانی که PB1 فشرده می شود، غلتک در جهت مناسب برای بالا بردن چنگک جرثقیل تا سنسور S3 حرکت می کند. سپس به سمت چپ حرکت کرده تا به محل سنسور S4 برسد. در این هنگام متوقف شده و قطعه را تا تانک روغن زدایی پایین می آورد. زمانی که به سنسور S2 رسید، قطعه در این تانک به مدت 20 ثانیه قرار می گیرد. بعد از اتمام 20 ثانیه قطعه بالا برده شده و به سمت چپ تا سنسور S5 انتقال پیدا می کند. سپس پایین آمده تا به سنسور S2 می رسد در این مکان زنگ به صدا در خواهد آمد. اپراتور قطعه را برداشته و با فشردن PB2 جرثقیل به مکان اولیه خود برخواهد گشت. در هر لحظه شستی Stop باعث توقف جرثقیل خواهد شد با خارج شدن از حالت Stop جرثقیل به ادامه عملکرد خود می پردازد. زمانی که عملکرد باید از ابتدا آغاز گردد باید شستی Reset فشرده شود.

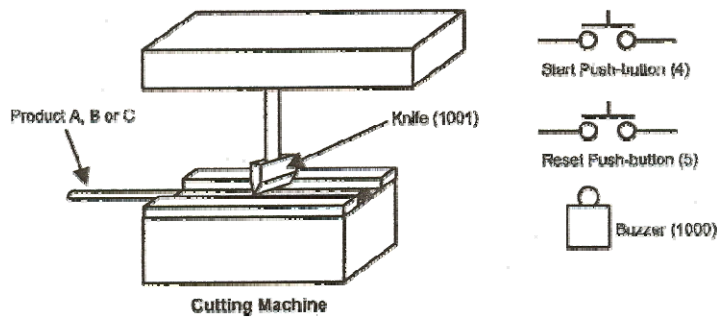
ورودی	عنصر
0.00	شستی Start (PB1)
0.01	شستی بازگشت (PB2)
0.02	سنسور 1 (S1)
0.03	سنسور پایین (S2)
0.04	سنسور بالا (S3)
0.05	سنسور روغن زدایی (S4)
0.06	سنسور انتهای (S5)
0.07	شستی Reset (RST)
0.08	شستی Stop

خروجی	عنصر
1.00	موتور بالا
1.01	موتور چپ
1.02	موتور پایین
1.03	موتور راست
1.04	زنگ



13-3- محاسبه عمر تیغه برش

در این مثال یک تیغه برای برش سه محصول مختلف A و B و C مورد استفاده قرار می گیرد. این تیغه باید بعد از برش 1000 قطعه A و یا 500 قطعه B و یا 100 قطعه C تعویض گردد. اما این محصولات به صورت درهم در روی خط قرار می گیرند. زمانی که عمر تیغه برش تمام شود زنگ به صدا در خواهد آمد. سه سنسور مختلف برای تشخیص نوع محصول به کار برده شده است و برای شروع دوباره عملکرد تیغه بعد از تعویض آن از یک شستی (Reset) استفاده می شود.



ورودی	عنصر
0.00	محصول A
0.01	محصول B
0.02	محصول C
0.03	شستی Start
0.04	شستی Reset

خروجی	عنصر
1.00	Buzzer
1.01	تیغه برش

14- طبقه بندی دستورالعملها با توجه به عملکرد

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

Classification	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	DIFFERENTIATE UP	DIFD	DIFFERENTIATE 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 OPERATION	—	—
	Interlock	IL	INTERLOCK	ILC	INTERLOCK CLEAR	MILH	MULTI-INTERLOCK DIFFERENTIATION HOLD
		MILR (See note 1.)	MULTI-INTERLOCK DIFFERENTIATION RELEASE	MILC (See note 1.)	MULTI-INTERLOCK CLEAR	—	—
	Jump	JMP	JUMP	JME	JUMP END	CJP	CONDITIONAL JUMP
		CJPN	CONDITIONAL 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 ^A	Timer (with timer numbers)	TIMX	TIMER	TIMHX	HIGH-SPEED TIMER	TMHXX	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	Time comparison	—	—	
	Data comparison (Condition Flags)	CMP	UNSIGNED COMPARE	CMP _L	DOUBLE UNSIGNED COMPARE	CPS	SIGNED BINARY COMPARE	
		CPS _L	DOUBLE SIGNED BINARY COMPARE	ZCP ^A	AREARANGE COMPARE	ZCPL ^A	DOUBLE AREARANGE 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	MOV _L	DOUBLE MOVE	MVN	MOVE NOT	
		MVNL	DOUBLE MOVE NOT	—	—	—	—	
	Bit/digit	MOVB	MOVE BIT	MOV _D	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	MOV _R	MOVE TO REGISTER	MOV _{RW}	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	FXL	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 RADIAN	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 ⁶	LD, AND, OR + =, <, <=, >, >= + F	Symbol comparison (single-precision floating point)	FSTR ⁶	FLOATING-POINT TO ASCII	FVAL ⁶	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 ^c	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 EXPLICIT	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 <i>bit_address</i>	Conditional END	EXIT NOT <i>bit_address</i>	Conditional END NOT	<i>input_condition</i> EXIT	Conditional END	
	IF branch processing	IF <i>bit_address</i>	CONDITIONAL BLOCK BRANCHING	IF NOT <i>bit_address</i>	CONDITIONAL BLOCK BRANCHING (NOT)	ELSE	CONDITIONAL BLOCK BRANCHING (ELSE)	
		IEND	CONDITIONAL BLOCK BRANCHING END	---	---	---	---	
	WAIT	WAIT <i>bit_address</i>	ONE CYCLE AND WAIT	WAIT NOT <i>bit_address</i>	ONE CYCLE AND WAIT NOT	<i>input_condition</i> 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 <i>bit_address</i>	LOOP BLOCK END	LEND NOT <i>bit_address</i>	LOOP BLOCK END NOT	
		<i>input_condition</i> LEND	LOOP BLOCK END	---	---	---	---	
Text string processing instructions	---	MOV\$	MOV STRING	+\$	CONCATE-NATE 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	---	---	

ASCII Code

DEC	OCT	HEX	BIN	Symbol	DEC	OCT	HEX	BIN	Symbol
0	000	00	00000000	NUL	44	054	2C	00101100	,
1	001	01	00000001	SOH	45	055	2D	00101101	-
2	002	02	00000010	STX	46	056	2E	00101110	.
3	003	03	00000011	ETX	47	057	2F	00101111	/
4	004	04	00000100	EOT	48	060	30	00110000	0
5	005	05	00000101	ENQ	49	061	31	00110001	1
6	006	06	00000110	ACK	50	062	32	00110010	2
7	007	07	00000111	BEL	51	063	33	00110011	3
8	010	08	00001000	BS	52	064	34	00110100	4
9	011	09	00001001	HT	53	065	35	00110101	5
10	012	0A	00001010	LF	54	066	36	00110110	6
11	013	0B	00001011	VT	55	067	37	00110111	7
12	014	0C	00001100	FF	56	070	38	00111000	8
13	015	0D	00001101	CR	57	071	39	00111001	9
14	016	0E	00001110	SO	58	072	3A	00111010	:
15	017	0F	00001111	SI	59	073	3B	00111011	;
16	020	10	00010000	DLE	60	074	3C	00111100	<
17	021	11	00010001	DC1	61	075	3D	00111101	=
18	022	12	00010010	DC2	62	076	3E	00111110	>
19	023	13	00010011	DC3	63	077	3F	00111111	?
20	024	14	00010100	DC4	64	100	40	01000000	@
21	025	15	00010101	NAK	65	101	41	01000001	A
22	026	16	00010110	SYN	66	102	42	01000010	B
23	027	17	00010111	ETB	67	103	43	01000011	C
24	030	18	00011000	CAN	68	104	44	01000100	D
25	031	19	00011001	EM	69	105	45	01000101	E
26	032	1A	00011010	SUB	70	106	46	01000110	F
27	033	1B	00011011	ESC	71	107	47	01000111	G
28	034	1C	00011100	FS	72	110	48	01001000	H
29	035	1D	00011101	GS	73	111	49	01001001	I
30	036	1E	00011110	RS	74	112	4A	01001010	J
31	037	1F	00011111	US	75	113	4B	01001011	K
32	040	20	00100000		76	114	4C	01001100	L
33	041	21	00100001	!	77	115	4D	01001101	M
34	042	22	00100010	"	78	116	4E	01001110	N
35	043	23	00100011	#	79	117	4F	01001111	O
36	044	24	00100100	\$	80	120	50	01010000	P
37	045	25	00100101	%	81	121	51	01010001	Q
38	046	26	00100110	&	82	122	52	01010010	R
39	047	27	00100111	'	83	123	53	01010011	S
40	050	28	00101000	(84	124	54	01010100	T
41	051	29	00101001)	85	125	55	01010101	U
42	052	2A	00101010	*	86	126	56	01010110	V
43	053	2B	00101011	+	87	127	57	01010111	W

ASCII Code

DEC	OCT	HEX	BIN	Symbol	DEC	OCT	HEX	BIN	Symbol
88	130	58	01011000	X	133	205	85	10000101	...
89	131	59	01011001	Y	134	206	86	10000110	†
90	132	5A	01011010	Z	135	207	87	10000111	‡
91	133	5B	01011011	[136	210	88	10001000	^
92	134	5C	01011100	\	137	211	89	10001001	%
93	135	5D	01011101]	138	212	8A	10001010	Š
94	136	5E	01011110	^	139	213	8B	10001011	<
95	137	5F	01011111	˘	140	214	8C	10001100	œ
96	140	60	01100000	`	141	215	8D	10001101	
97	141	61	01100001	a	142	216	8E	10001110	ž
98	142	62	01100010	b	143	217	8F	10001111	
99	143	63	01100011	c	144	220	90	10010000	
100	144	64	01100100	d	145	221	91	10010001	‘
101	145	65	01100101	e	146	222	92	10010010	’
102	146	66	01100110	f	147	223	93	10010011	“
103	147	67	01100111	g	148	224	94	10010100	”
104	150	68	01101000	h	149	225	95	10010101	•
105	151	69	01101001	i	150	226	96	10010110	–
106	152	6A	01101010	j	151	227	97	10010111	—
107	153	6B	01101011	k	152	230	98	10011000	~
108	154	6C	01101100	l	153	231	99	10011001	™
109	155	6D	01101101	m	154	232	9A	10011010	š
110	156	6E	01101110	n	155	233	9B	10011011	>
111	157	6F	01101111	o	156	234	9C	10011100	œ
112	160	70	01110000	p	157	235	9D	10011101	
113	161	71	01110001	q	158	236	9E	10011110	ž
114	162	72	01110010	r	159	237	9F	10011111	ÿ
115	163	73	01110011	s	160	240	A0	10100000	
116	164	74	01110100	t	161	241	A1	10100001	ı
117	165	75	01110101	u	162	242	A2	10100010	ı
118	166	76	01110110	v	163	243	A3	10100011	£
119	167	77	01110111	w	164	244	A4	10100100	ı
120	170	78	01111000	x	165	245	A5	10100101	¥
121	171	79	01111001	y	166	246	A6	10100110	ı
122	172	7A	01111010	z	167	247	A7	10100111	§
123	173	7B	01111011	{	168	250	A8	10101000	¨
124	174	7C	01111100		169	251	A9	10101001	©
125	175	7D	01111101	}	170	252	AA	10101010	ª
126	176	7E	01111110	~	171	253	AB	10101011	«
127	177	7F	01111111		172	254	AC	10101100	¬
128	200	80	10000000	€	173	255	AD	10101101	
129	201	81	10000001		174	256	AE	10101110	®
130	202	82	10000010	,	175	257	AF	10101111	—
131	203	83	10000011	f	176	260	B0	10110000	°
132	204	84	10000100	„	177	261	B1	10110001	±

ASCII Code

DEC	OCT	HEX	BIN	Symbol	DEC	OCT	HEX	BIN	Symbol
178	262	B2	10110010	²	217	331	D9	11011001	Û
179	263	B3	10110011	³	218	332	DA	11011010	Ü
180	264	B4	10110100	´	219	333	DB	11011011	Û
181	265	B5	10110101	µ	220	334	DC	11011100	Ü
182	266	B6	10110110	¶	221	335	DD	11011101	Ý
183	267	B7	10110111	·	222	336	DE	11011110	Þ
184	270	B8	10111000	,	223	337	DF	11011111	ß
185	271	B9	10111001	¹	224	340	E0	11100000	à
186	272	BA	10111010	º	225	341	E1	11100001	á
187	273	BB	10111011	»	226	342	E2	11100010	â
188	274	BC	10111100	¼	227	343	E3	11100011	ã
189	275	BD	10111101	½	228	344	E4	11100100	ä
190	276	BE	10111110	¾	229	345	E5	11100101	å
191	277	BF	10111111	¿	230	346	E6	11100110	æ
192	300	C0	11000000	À	231	347	E7	11100111	ç
193	301	C1	11000001	Á	232	350	E8	11101000	è
194	302	C2	11000010	Â	233	351	E9	11101001	é
195	303	C3	11000011	Ã	234	352	EA	11101010	ê
196	304	C4	11000100	Ä	235	353	EB	11101011	ë
197	305	C5	11000101	Å	236	354	EC	11101100	ì
198	306	C6	11000110	Æ	237	355	ED	11101101	í
199	307	C7	11000111	Ç	238	356	EE	11101110	î
200	310	C8	11001000	È	239	357	EF	11101111	ï
201	311	C9	11001001	É	240	360	F0	11110000	ð
202	312	CA	11001010	Ê	241	361	F1	11110001	ñ
203	313	CB	11001011	Ë	242	362	F2	11110010	ò
204	314	CC	11001100	Ì	243	363	F3	11110011	ó
205	315	CD	11001101	Í	244	364	F4	11110100	ô
206	316	CE	11001110	Î	245	365	F5	11110101	õ
207	317	CF	11001111	Ï	246	366	F6	11110110	ö
208	320	D0	11010000	Ð	247	367	F7	11110111	÷
209	321	D1	11010001	Ñ	248	370	F8	11111000	ø
210	322	D2	11010010	Ò	249	371	F9	11111001	ù
211	323	D3	11010011	Ó	250	372	FA	11111010	ú
212	324	D4	11010100	Ô	251	373	FB	11111011	û
213	325	D5	11010101	Õ	252	374	FC	11111100	ü
214	326	D6	11010110	Ö	253	375	FD	11111101	ý
215	327	D7	11010111	×	254	376	FE	11111110	þ
216	330	D8	11011000	Ø	255	377	FF	11111111	ÿ



تهران 15167، خیابان وزرا، روبروی خیابان سی و پنجم، شماره 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