



الجمهورية العربية السورية
جامعة دمشق
كلية التطبيقية
قسم الميكاترونكس

المتحكمات المنطقية القابلة للبرمجة

PLC



الدكتور المهندس

سامر حسام الدين

جامعة دمشق

كلية الهندسة الميكانيكية والكهربائية

قسم التصميم الميكانيكي

2020 -2019

الفهرس:

الصفحة	الموضوع
1	مقدمة.....
2	الفصل الأول: العمليات المنطقية.....
2	1-1 الوظائف المنطقية الأساسية.....
5	2-1 عمليات منطقية أخرى.....
6	3-1 تشكيل مفاتيح الوظائف.....
9	4-1 تبسيط الوظائف المنطقية.....
10	5-1 مخطط Karnaugh-Veitch.....
13	الفصل الثاني: مكونات PLC.....
13	1-2 مجالات استعمال الـ PLC.....
13	2-2 دورة عمل PLC.....
16	3-2 وحدات التخزين في الذواكر.....
17	4-2 عناصر التحكم المتقدمة (Advanced control elements).....

مقدمة:

طور جهاز التحكم المنطقي القابل للبرمجة الأول من قبل مجموعة من المهندسين في شركة جنرال موتور عام 1968 عندما كانت الشركة تبحث عن بديل عن نظم التحكم بالحاكمات المعقدة. نظام التحكم الجديد يجب أن يلبي المتطلبات التالية:

*برمجة بسيطة

*إمكانية تغيير البرنامج بدون تغيير في بنية النظام

*أصغر، أرخص وموثوق أكثر من نظم التحكم بالحاكمات.

*بسيط وتكاليف صيانتها منخفضة.

تطور لاحقاً النظام الأمر الذي سمح باتصال بسيط بالإشارات الثنائية. المتطلبات الأساسية كانت كيف يمكن التعامل مع الإشارات بالشكل المحدد في برنامج التحكم. مع النظم الجديدة أصبح ممكناً ولأول مرة توضيح الإشارات على الشاشة وتخزينها في الذاكرة الإلكترونية.

منذ ذلك الوقت ثلاثة عقود قد مرت وشوط كبير قد أنجز في تطوير الإلكترونيات الدقيقة لم يحصل فيه ولو توقف قليل في نظم المتحكمات القابلة للبرمجة على سبيل المثال حتى ولو في تحسين البرمجيات وتخفيض حدود سعة الذاكرة الابتدائية التي تبقى ممثلة بمفتاح مهم من أجل المبرمج. في هذه الأيام يعد ذلك صعباً في الأعمال ذات الأهمية.

علاوة على ذلك مجال الوظائف نما بشكل كبير جداً. في السنوات الخمسة عشر الماضية العمليات التصورية (visualization)، العمليات المنطقية، أو حتى استخدام الـ PLC كمتحكم كان يعتبر شيئاً خيالياً. هذه الأيام شكل دعم هذه الوظائف الأجزاء الذكية لأنواع أجهزة PLC المختلفة، الصفحات اللاحقة من هذه المقدمة التمهيدية توجز التصميم الأساسي لـ PLC مع استعمالاته ومهامه الأساسية الحالية.



الفصل الأول

العمليات المنطقية

العمليات المنطقية

1-1 الوظائف المنطقية الأساسية :

كما شرحنا في الفقرة السابقة أي جهاز كمبيوتر وبشكل مكافئ أي جهاز PLC يعمل مستخدماً الترقيم أو النظام الرقمي للأساس 2 . هذا ينطبق أيضاً على النظام الثماني والست عشري . هذه المتغيرات بشكل فردي يمكنها أن تأخذ فقط القيمتين 0-1 . شكلت الخوارزميات الخاصة لتكون قادرة على الربط بين المتغيرات بما يسمى الجبر المنطقي الذي يكون واضحاً عندما يمثل في الاسطوانات الكهربائية .

1- النفي (وظيفة NOT)

زر الضاغط الظاهر بالمثل يمثل بشكل طبيعي التماس المغلق عندما يكون غير مشغل، المصباح H1 يكون مضاء أما في الحالة غير المشغلة يكون H1 مطفاً .

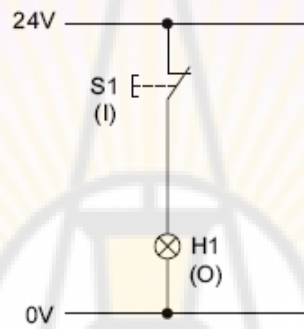


Fig. B3.1:
Circuit diagram

زر الضاغط S1 يعمل كإشارة دخل ، المصباح يشكل الخرج . الحالة المشغلة يمكن أن تشكل في جدول الحقيقة :

I	O
0	1
1	0

Fig. B3.2:
Truth table

المعادلة المنطقية تكون كالتالي: $\bar{I} = O$ (read: Not I equals O)

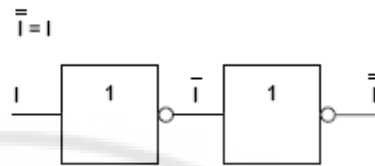
الرموز المنطقية :

Fig. B3.3:
NOT function



إذا كان هناك نفيين متعاقبين عند ذلك يلغي أحدهما الآخر .

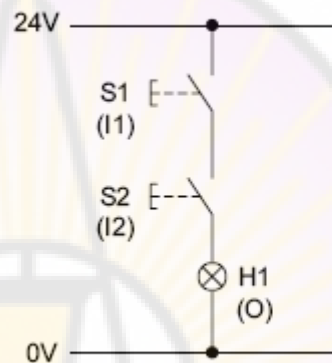
Fig. B3.4:
2 logic
NOT functions



2- تابع الوصل (وظيفة AND)

إذا جمع تماس مفتوح على التسلسل يكون المصباح منار فقط إذا كان كلا مفتاحي التماس مشغلين .

Fig. B3.5:
Circuit diagram



I1	I2	O
0	0	0
0	1	0
1	0	0
1	1	1

Fig. B3.6
Truth table

جدول الحقيقة المأخوذ في حالة الوصل. يتخذ الخرج القيمة 1 فقط إذا كان الدخل الأول والثاني يساوي 1. هذه تدعى عملية AND التي تمثل كمعادلة بالشكل التالي: $I1 \wedge I2 = O$

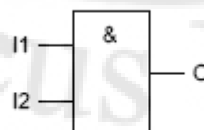


Fig. B3.7
AND function

بالإضافة إلى ذلك الخوارزمية التالية مخصصة لأجل تابع الوصل

$$a \wedge 0 = 0$$

$$a \wedge 1 = a$$

$$a \wedge \bar{a} = 0$$

$$a \wedge a = a$$

3- الفصل (وظيفة OR)

أحد التوابع المنطقية الأساسية هو OR . إذا كان اثنان من توابع التماس المفتوح موصولين على التوازي ، يكون المصباح مضاء عندما واحد منهما أو واحد من مفتاحي التماس موصول .

Fig. B3.8:
Circuit diagram

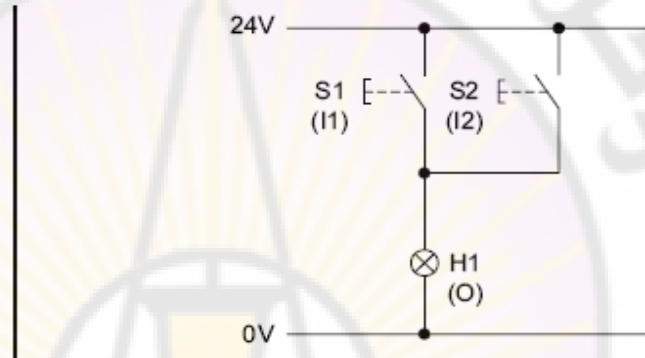
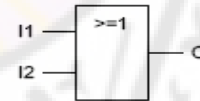


Fig. B3.9
Truth table

I1	I2	O
0	0	0
0	1	1
1	0	1
1	1	1

Fig. B3.10:
OR function



العملية المنطقية تكتب على شكل المعادلة التالية :

$$I1 \vee I2 = O$$

الخوارزمية التالية تقدم شرح لعملية OR

$$b \vee 0 = b$$

$$b \vee 1 = 1$$

$$b \vee b = b$$

$$b \vee \bar{b} = 1$$

2-1 عمليات منطقية أخرى :

المفهوم الكهربائي لعمليات (NOT)، (AND)، (OR) شرح في الفقرات السابقة كل من هذه العمليات يمكن أن يشرح أيضاً من خلال الهواء المضغوط أو الإلكترونيات . الجبر المنطقي أيضاً يقدم شرحاً للعمليات المنطقية التالية .

الجدول التالي يزودنا بتوضيح عن :

Name	Equation	Truth table	log. symbols	pneumatic realisation	elektr. realisation	elektron. realisation															
Identity	$I = O$	<table><tr><td>I</td><td>O</td></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	I	O	0	0	1	1													
I	O																				
0	0																				
1	1																				
Negation	$\bar{I} = O$	<table><tr><td>I</td><td>O</td></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	I	O	0	1	1	0													
I	O																				
0	1																				
1	0																				
Conjunction	$I1 \wedge I2 = O$	<table><tr><td>I1</td><td>I2</td><td>O</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	I1	I2	O	0	0	0	0	1	0	1	0	0	1	1	1				
I1	I2	O																			
0	0	0																			
0	1	0																			
1	0	0																			
1	1	1																			
Disjunction	$I1 \vee I2 = O$	<table><tr><td>I1</td><td>I2</td><td>O</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	I1	I2	O	0	0	0	0	1	1	1	0	1	1	1	1				
I1	I2	O																			
0	0	0																			
0	1	1																			
1	0	1																			
1	1	1																			

Name	Equation	Truth table	log. symbols	pneumatic realisation	elektr. realisation	elektron. realisation															
Antivalence (exclusive OR)	$\frac{I1 \wedge \overline{I2} \vee \overline{I1} \wedge I2}{I1 \wedge I2 = 0}$	<table><tr><th>I1</th><th>I2</th><th>O</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	I1	I2	O	0	0	0	0	1	1	1	0	1	1	1	0				
I1	I2	O																			
0	0	0																			
0	1	1																			
1	0	1																			
1	1	0																			
Equivalence	$\frac{I1 \wedge I2 \vee \overline{I1} \wedge \overline{I2}}{I1 \wedge I2 = 0}$	<table><tr><th>I1</th><th>I2</th><th>O</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	I1	I2	O	0	0	1	0	1	0	1	0	0	1	1	1				
I1	I2	O																			
0	0	1																			
0	1	0																			
1	0	0																			
1	1	1																			
NAND	$\overline{I1 \wedge I2} = 0$	<table><tr><th>I1</th><th>I2</th><th>O</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	I1	I2	O	0	0	1	0	1	1	1	0	1	1	1	0				
I1	I2	O																			
0	0	1																			
0	1	1																			
1	0	1																			
1	1	0																			
NOR	$\overline{I1 \vee I2} = 0$	<table><tr><th>I1</th><th>I2</th><th>O</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	I1	I2	O	0	0	1	0	1	0	1	0	0	1	1	0				
I1	I2	O																			
0	0	1																			
0	1	0																			
1	0	0																			
1	1	0																			

3-1 تشكيل مفاتيح الوظائف :

اشتقاق المعادلات المنطقية يتم من جدول الحقيقة ، غالباً تكون العمليات المنطقية المشروحة في الفقرة السابقة غير كافية لتوضيح حالات تقنية التحكم . في أغلب الأحيان هناك مزيج من العمليات المختلفة الاتصال المنطقي لذلك شكل المعادلة المنطقية يمكن أن يؤسس بسهولة من خلال جدول الحقيقة ، المثال التالي يمكن أن يوضح ذلك :

مهمة محطات الفرز :

أجزاء مختلفة تستعمل للبناء الداخلي لهذه المطابخ (المحطات) لنحصل على نظام إنتاج آلي (آلات الثقب والصهر) . الأبواب والجدران معدة من أجل أنماط محددة من المطابخ المزودة بآلات ثقب مختلفة حيث الحساسات معدة للتحري عن الثقب.

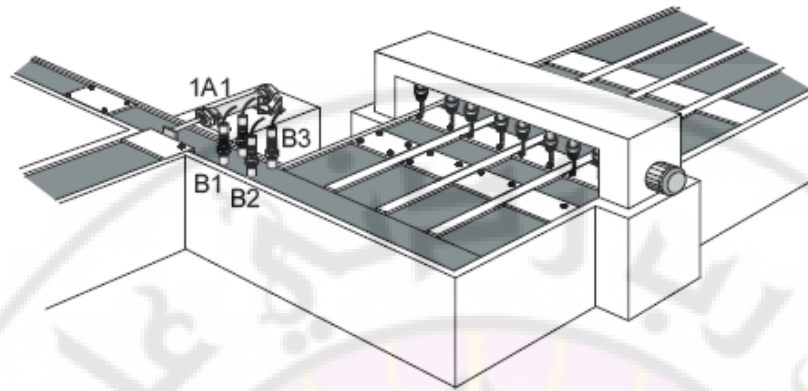
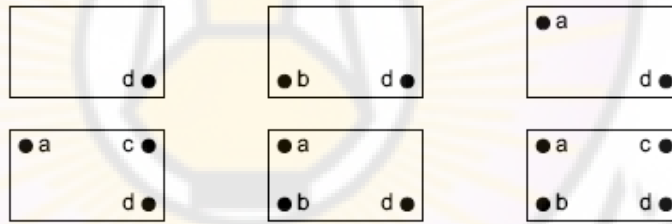


Fig. B3.11:
Sorting station

الأجزاء مع نماذج الثقوب التالية هي لأجل أنماط المطابيح القياسية. هذه الأجزاء تستخدم من خلال الاسطوانة ثنائية التأثير 1.0 .

Fig. B3.12:
Hole pattern parts



بفرض أن أداة حفر الثقب تقرأ كإشارة 1 ينتج لدينا جدول الحقيقة التالي :

Fig. B3.13
Truth table

a	b	c	d	y
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

خياران متوفران كي تشتق المعادلة المنطقية من جدول الحقيقة والذان يقودان إلى صيغتين مختلفتين ولكن بنتيجة واحدة طبعاً إذا كانت الظروف نفسها .

الشكل القياسي ، الفصل:

في شكل الفصل القياسي أي عملية وصل (AND) للمداخل المتنوعة تعطي نتيجة 1 ، تقدم كعملية فصل (OR) . حالة الإشارة 0 المداخل المتنوعة تقدم كعملية نفي ومع حالة الإشارة 1 يكون لدينا نفي النفي .

في المثال المعطى العملية المنطقية تظهر كالتالي:

$$y = (\bar{a} \wedge \bar{b} \wedge \bar{c} \wedge d) \vee (\bar{a} \wedge b \wedge \bar{c} \wedge d) \vee (a \wedge \bar{b} \wedge \bar{c} \wedge d) \vee (a \wedge \bar{b} \wedge c \wedge d) \vee (a \wedge b \wedge \bar{c} \wedge d) \vee (a \wedge b \wedge c \wedge d)$$

الشكل القياسي للوصل:

في الشكل القياسي للوصل كل فصل (OR) للمداخل المتنوعة يقود للنتيجة 0 تنفذ مثل عملية الوصل (AND) . على خلاف مع الحالة القياسية للفصل في هذا المثال تكون المداخل متنوعة منفية في حالة الإشارة 1 ونفي النفي في حالة الإشارة 0.

$$y = (a \vee b \vee c \vee d) \wedge (a \vee b \vee \bar{c} \vee d) \wedge (a \vee b \vee \bar{c} \vee \bar{d}) \wedge (a \vee \bar{b} \vee c \vee d) \wedge (a \vee \bar{b} \vee \bar{c} \vee d) \wedge (a \vee \bar{b} \vee \bar{c} \vee \bar{d}) \wedge (\bar{a} \vee b \vee c \vee d) \wedge (\bar{a} \vee b \vee \bar{c} \vee d) \wedge (\bar{a} \vee \bar{b} \vee c \vee d) \wedge (\bar{a} \vee \bar{b} \vee \bar{c} \vee d)$$

4-1 تبسيط الوظائف المنطقية :

كلتا المعادلتين للمثال المعطى تعطي اختلافاً بالطول والكبر المأخوذة بالشكل القياسي للوصل تكون أطول في هذا المثال . الذي يوضح المعايير لاستعمال حالتى الوصل أو الفصل هو الشكل الأقصر للمعادلة إذاً في هذه الحالة نأخذ حالة الفصل (OR) .

$$y = (\bar{a} \wedge \bar{b} \wedge \bar{c} \wedge d) \vee (\bar{a} \wedge b \wedge \bar{c} \wedge d) \vee (a \wedge \bar{b} \wedge \bar{c} \wedge d) \vee (a \wedge \bar{b} \wedge c \wedge d) \vee (a \wedge b \wedge \bar{c} \wedge d) \vee (a \wedge b \wedge c \wedge d)$$

هذا التعبير يمكن أن يبسط بمساعدة الخوارزميات المنطقية .
القواعد الأكثر أهمية للجبر المنطقي تعطى كالتالي:

$$a \vee 0 = a \quad a \wedge 0 = 0$$

$$a \vee 1 = 1 \quad a \wedge 1 = a$$

$$a \vee a = a \quad a \wedge a = a$$

$$a \vee \bar{a} = 1 \quad a \wedge \bar{a} = 0$$

قانون التبادل:

$$a \vee b = b \vee a \quad a \wedge b = b \wedge a$$

قانون التجميع :

$$a \vee b \vee c = a \vee (b \vee c) = (a \vee b) \vee c$$

$$a \wedge b \wedge c = a \wedge (b \wedge c) = (a \wedge b) \wedge c$$

قانون التوزيع

$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c) \quad a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$$

قانون De morgan

$$\overline{a \vee b} = \bar{a} \wedge \bar{b} \quad \overline{a \wedge b} = \bar{a} \vee \bar{b}$$

قانون التخفيض

$$a \vee \bar{a} \wedge b = a \vee b$$

بالتطبيق على المثال السابق نصل إلى النتيجة التالية :

$$\begin{aligned}
 y &= \overline{a}bcd \vee a\overline{b}cd \vee ab\overline{c}d \vee abcd \vee ab\overline{c}d \vee abcd \\
 &= \overline{a}bcd \vee a\overline{b}cd \vee ab\overline{c}d \vee abcd \vee abd(\overline{c} \vee c) \\
 &= \overline{a}cd(\overline{b} \vee b) \vee abd(\overline{c} \vee c) \vee abd \\
 &= \overline{a}cd \vee abd \vee abd \\
 &= \overline{a}cd \vee ad(\overline{b} \vee b) \\
 &= (\overline{a}c \vee a)d \\
 &= (\overline{c} \vee a)d \\
 &= \overline{c}d \vee ad
 \end{aligned}$$

من أجل التوضيح في عملية and رمز الوصل يحذف في التعابير الفردية .
 القاعدة الأساسية في التبسيط هي في التحليل إلى عوامل والتخفيض إلى معايير محددة ، على أية حال هذا يتطلب خبرة كبيرة من الممارسة العملية ومعرفة عميقة بالخوارزميات المنطقية . المعيار الآخر للتبسيط سيشرح في الفقرة القادمة .

5-1 مخطط Karnaugh-Veitch :

في حالة مخططات Karnaugh-Veitch (مخطط KV) جدول الحقيقة يحول إلى قيم

Fig. B3.14:
Truth table

a	b	c	d	y	No.
0	0	0	0	0	1
0	0	0	1	1	2
0	0	1	0	0	3
0	0	1	1	0	4
0	1	0	0	0	5
0	1	0	1	1	6
0	1	1	0	0	7
0	1	1	1	0	8
1	0	0	0	0	9
1	0	0	1	1	10
1	0	1	0	0	11
1	0	1	1	1	12
1	1	0	0	0	13
1	1	0	1	1	14
1	1	1	0	0	15
1	1	1	1	1	16

مجموعة من 16 خيار مخصص متوفر في هذا المثال وفقاً له يجب أن يتوفر في جدول القيم 16 مربع مخصص

Fig. B3.15:
Value table

	\overline{cd}	\overline{cd}	$c\overline{d}$	cd
\overline{ab}	1	2	3	4
\overline{ab}	5	6	7	8
$a\overline{b}$	9	10	11	12
ab	13	14	15	16

تحول نتائج جدول القيم إلى جدول KV تبعاً للجدول المعروض في الأسفل . في المبدأ ، يمكن أن يكون التمثيل في المخطط بطريقة الوصل (AND) أو الفصل (OR) . المثال التالي على أية حال سيدرس بطريقة الفصل

Fig. B3.16:
Value table

	\overline{cd}	\overline{cd}	$c\overline{d}$	cd
\overline{ab}	0	1	0	0
\overline{ab}	0	1	0	0
$a\overline{b}$	0	1	0	1
ab	0	1	0	1

الخطوة التالية تشمل دمج الحالات التي أخذت القيمة 1 وأدخلت في جدول القيم . هي تعامل في مربعات خاصة طالما نراعي القواعد التالية:

- حالات الدمج والجمع في مخطط KV يجب أن تكون على شكل مستطيل أو مربع
 - عدد حالات الدمج يجب أن يكون من الشكل 2^x
- النتائج تدرج كالتالي:

Fig. B3.17:
Value table

	\overline{cd}	\overline{cd}	$c\overline{d}$	cd
\overline{ab}	0	1	0	0
\overline{ab}	0	1	0	0
$a\overline{b}$	0	1	0	1
ab	0	1	0	1

Y1 Y2

تختار القيم المتغيرة للكتل المهيأة وذلك بحالة الدمج (or) كالتالي:

$$y_1 = \bar{c}d$$

$$y_2 = acd$$

$$y = \bar{c}d \vee acd$$

$$= (\bar{c} \vee ac) \wedge d$$

$$= (\bar{c} \vee a) \wedge d$$

$$= \bar{c}d \vee ad$$

طبيعياً لا يحدد مخطط KV لـ 16 مربع فقط مثلاً 5 متغيرات سوف تعطي عدد من المربعات تساوي $2^5=32$ مربع وست متغيرات تعطي 64 مربع أي 2^6

الفصل الثاني

مكونات PLC

1-2 مجالات استعمال الـ PLC

كل نظام أو آلة لديه متحكم يعتمد على أحد الأنماط التقنية المستخدمة. المتحكمات يمكن أن تقسم إلى: الهواء المضغوط – الهيدروليكية – الكهربائية – والمتحكمات الإلكترونية. عادةً يمكن أن تجتمع أنماط تقنية مختلفة أثناء الاستعمال، علاوة على ذلك يجب أن نفرق بين الأجزاء الكهربائية القابلة للبرمجة (مثال: التوصيل السلكي في الآلات الكهربائية أو المكونات الإلكترونية) والمتحكمات المنطقية القابلة للبرمجة. تستخدم الحالة الأولى في الحالات الابتدائية حيث أي إعادة برمجة من قبل المستخدم تكون خارج المهمة وحجم العمل يجيز التطوير فقط في المتحكمات خاصة. الاستخدام النموذجي لمتحكمات كهذه يمكن أن يوجد في آلات الغسل الأوتوماتيكية – كاميرات الفيديو والسيارات حتى لو لم يسمح حجم العمل في التطوير في المتحكمات الخاصة أو إذا كان لدى المستخدم القدرة على تسهيل عملية التبسيط أو استقلالية تغيير البرامج أو برمجة المؤقت والعداد، إذاً استخدام المتحكم المركزي أو العام أثناء كتابة البرامج وتنصيبه على الذاكرة الإلكترونية يكون الخيار الأفضل.

يمكن أن يمثل PLC بالمتحكم العام. إنه يمكن أن يستخدم في استعمالات مختلفة، وبواسطة البرنامج المنصب على الذاكرة يمنح المستخدم إمكانيات بسيطة للتغيير وتوسيع وتحسين عمليات التحكم المهمة الأساسية للـ PLC تتضمن ربط إشارات الدخل تبعاً للبرنامج المحدد إذا كان "true" إلى مفاتيح المخارج المطابقة. الجبر المنطقي يشكل الأساس الرياضي للعملية والتي تتميز بدقة الحالتين المعروفتين لأحد المتغيرات 0 و 1 ((انظر المقطع 3)) بناءً على ذلك يمكن أن يأخذ الخرج فقط هاتين الحالتين. على سبيل المثال أي محرك يمكن أن يكون إما مطفأ أو مشغل عند التحكم به. هذه الأداة التي تقوم بهذه الوظيفة تأخذ اسم الـ PLC: المتحكم المنطقي القابل للبرمجة. حيث تقوم المداخل والمخارج بوظيفة مشابهة لوظيفة الحواكم الكهربائية أو مفاتيح التحكم بصمامات الهواء المضغوط.

البرنامج يخزن على الذاكرة الإلكترونية. على أية حال مهام PLC ازدادت بسرعة إلى مهام المؤقت والعداد، الذاكرة يمكن إعدادها وإعادة تهيئتها. عمليات الكمبيوتر الرياضية يمكن أن تنفذ كل التتابع الممثلة رياضياً والتي أصبح من الممكن تنفيذها على جميع أجهزة الـ PLC الحالية تقريباً.

2-2 دورة عمل PLC:

وهي التي يحددها نظام التشغيل المخزن في الذاكرة الثابتة وتقوم بتنفيذ برنامج المستثمر وتكون دورة العمل هذه متكررة مادام PLC في حالة تشغيل (run) كما يوضح الشكل (1-50) ونجد أن هذه العملية تمر بثلاث مراحل هي:

1- مسح المداخل: حيث يقرأ PLC المعلومات من كل وحدات الدخل والقادمة من الحساسات والمفاتيح الموصولة على المداخل وينسخها إلى جزء من الذاكرة يسمى صورة الدخل.

2- تنفيذ البرنامج: يقوم بتنفيذ برنامج المستخدم لمرة واحدة من بدايته إلى نهايته يفحص معلومات صورة الدخل ويحدد ما على PLC عمله ومن ثم يقوم بتخريج المعطيات إلى المشغلات ولكن معلومات الخرج هذه لا تؤثر على حالة المخارج وإنما يحتفظ بها جزء من الذاكرة يسمى صورة الخرج.

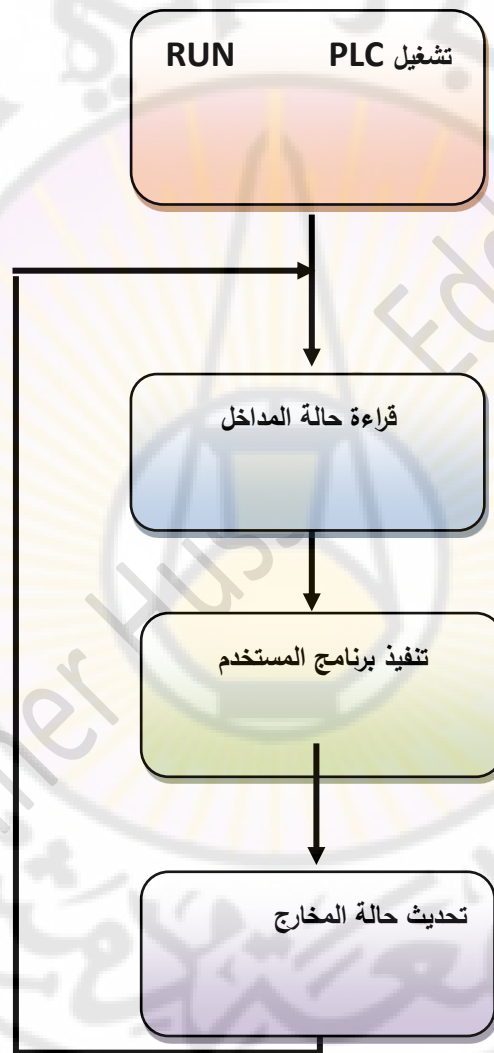
3- تحديث حالة المخارج: نسخ المعلومات من منطقة صورة الخرج إلى وحدات الخرج. هذا ويعتمد زمن دورة المسح على عدة عوامل:

1-طول برنامج المستخدم.

2-سرعة المعالج.

3-عدد المداخل والمخارج المستخدمة (وجود كتل توسع).

ويكون عادةً زمن المسح من 0.1 ms وحتى عشرات الميلي ثانية وذلك لـ 1000 تعليمة.



الشكل (2-1) دورة العمل في جهاز PLC

برامج معالجة البيانات التقليدية تتم معالجتها مرة واحدة فقط من الأعلى إلى الأسفل وبعد ذلك تنتهي ، وبما يخالف ذلك فإن برنامج PLC يعالج دورياً وباستمرار .

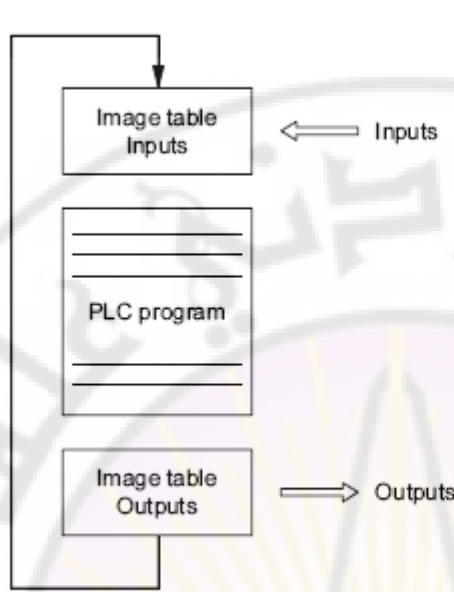


Fig. B4.5:
Cyclical processing of
a PLC program

مميزات المعالجة الدورية هي:

- 1- فور تنفيذ البرنامج للمرة الأولى فإنه يقفز أوتوماتيكياً إلى البداية ويتم إعادة المعالجة.
 - 2- قبل أن يتم معالجة السطر الأول من البرنامج بمعنى آخر في بداية الدورة يتم تخزين حالة المدخل في جدول الصورة. صورة المعالجة هي منطقة من الذاكرة منفصلة ويتم الوصول إليها أثناء الدورة وبالتالي حالة المدخل تبقى ثابتة أثناء الدورة حتى ولو تغيرت فيزيائياً.
 - 3- لا يتم فوراً إعداد أو إعادة إعداد ما هو شبيه بالمدخل والمخارج أثناء الدورة ولكن يتم تخزين الحالة (status) مباشرة في صورة مخرجات المعالجة وفقط في نهاية الدورة يتم تبديل كل الإخراجات فيزيائياً وفقاً للحالات المنطقية المخزنة في الذاكرة. معالجة سطر البرنامج من خلال وحدة التحكم المركزية لـ PLC تأخذ زمناً يعتمد على PLC وهذه العملية تتراوح ما بين عدة ميكرو ثانية وعدة ميلي ثانية.
- والزمن الذي يحتاجه PLC من أجل تنفيذ البرنامج لمرة واحدة بما ذلك زمن التحديث والإخراج من صورة المعالجة يصطلح بتسميته زمن الدورة، وكلما كان البرنامج أطول كلما طال الزمن اللازم لـ PLC لمعالجة سطر البرنامج الفردي وكلما كانت الدورة أطول وتبلغ فترات الزمن الحقيقي لذلك ما بين 1-100 ميلي ثانية

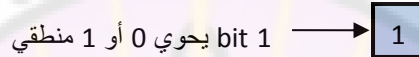
تسلسل المعالجة الدورية للبرنامج PLC باستخدام صورة المعالجة هي التالي:

- إشارات الإدخال الأقصر من الزمن الدوري لا يمكن تمييزها.

- في بعض الحالات قد يوجد زمن تأخير بزمن دورتين بين حدوث إشارة الإدخال والتفاعل المطلوب لإخراج هذه الإشارة.
- بما أن الأوامر تتم معالجتها بالتسلسل بالتالي قد يكون السلوك الخاص لتسلسل برنامج PLC مسألة (رئيسية).
- في بعض التطبيقات من الضروري ارتباط المداخل والمخارج أثناء دورة العمل بشكل مباشر كمدخل للحصول والوصول للمعلومة. هذا النمط من معالجة البرنامج من خلال تمريره في صورة المعالجة هو أيضا مدعوم من بعض أنظمة PLC.

3-2 وحدات التخزين في الذاكر :

البت Bit: وهي خانة واحدة تحوي قيمة ثنائية يمكن أن تكون 0 أو 1 .



البايت Byte: وهي عبارة عن ثمانية خانات أي (8 Bit) .



Byte = 8 Bit

الكلمة Word: وهي عبارة عن 2 Byte أي 16 Bit .



Byte n + 1

Byte n

2 أي 4 byte .

Word n

الكلمة المزدوجة Double Word

Double Word n



Byte n

Byte n + 1

Byte n + 2

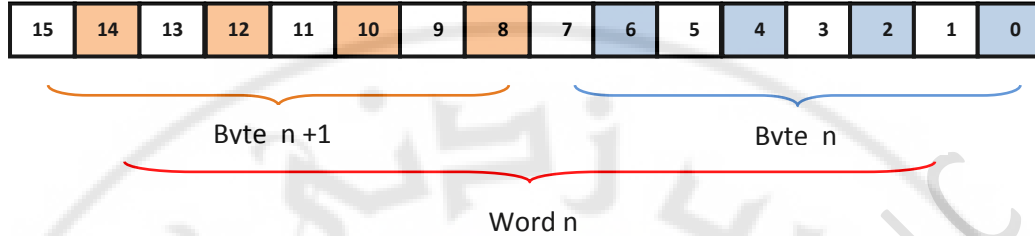
Byte n + 3

Word n

Word n + 1

المعطيات التي يتم التعامل معها في الذاكرة:

العدد الصحيح Integer number : وهو عبارة عن 16 Bit وهي أعداد لها إشارة في المدى من 32768 - إلى 32767 + .



ملاحظة: البت رقم 15 تكون إشارته كآتي:

0 يعني أن إشارة العدد الصحيح موجبة.

1 يعني أن إشارة العدد الصحيح سالب

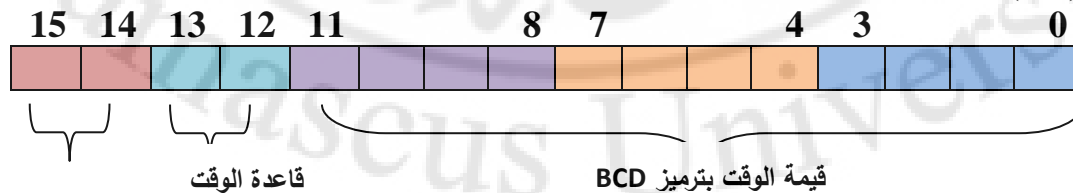
العدد الصحيح المزدوج Double Integer number : وهو عبارة عن 32 Bit وهي أعداد لها إشارة في المدى من 2147483648 - إلى 2147483647 + وهنا البت رقم 31 يدل على إشارة العدد الصحيح



العدد الحقيقي Real number : وهو عبارة عن 32 Bit وتسمى أيضاً بالأعداد ذات الفاصلة العائمة وهي في المدى السالب من 38 - 3.402823E إلى 38 + 1.175495E - والمدى الموجب من 38 - 1.175495E إلى 38 + 3.402823E .

4-2 عناصر التحكم المتقدمة (Advanced control elements) :

1. **المؤقتات (Timers):** تعتبر المؤقتات من أهم الوظائف في أجهزة PLC فتستخدم مثلاً لتأخير عمل المحرك لفترة زمنية معينة أو لتشغيل مضخة لفترة زمنية معينة. يتعلق عدد المؤقتات الموجودة في جهاز PLC بإمكانية وحدة المعالجة المركزية.
- قاعدة التوقيت:** يقوم المؤقت بحجز (16 Bit = 1 Word) من الذاكرة وذلك لقيمة الزمن التي نريد العمل عندها، يتم استخدام ترميز BCD في تحميل قيمة الوقت ويكون على الشكل (1-3) :



الشكل (2-2)

15 , 14 Bits : تبقى دائماً في حالة الصفر ويتم تجاهلها عند عمل المؤقت .
13 , 12 Bits : تشير إلى قاعدة التوقيت ويتم ترميز قاعدة التوقيت حسب الجدول الآتي :

Time base	Binary code for the time base
10 ms	00
100 ms	01
1s	10
10 s	11

11 to 8 Bits : تشير إلى قيمة المئات .

7 to 4 Bits : تشير إلى قيمة العشرات .

3 to 0 Bits : تشير إلى قيمة الواحدات .

تحميل الوقت: يتم تحديد الوقت بالطريقة الآتية:

S5T # a H – bb M – cc S – ddd MS

حيث: a : ساعة ، bb : دقيقة ، cc : ثواني ، ddd : ميلي ثانية
وأقصى مدة توقيت يمكن تحميلها هي : 2 H – 46 M – 30 S ويتم تحديد قاعدة التوقيت تلقائياً.

1-1-2-المؤقت النبضي Pulse Timer:

S : مدخل تفعيل المؤقت.

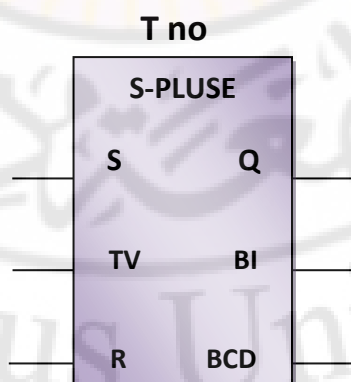
TV : مدخل تعريف قيمة الوقت.

R : مدخل تصفير المؤقت.

Q : مخرج يمثل حالة المؤقت.

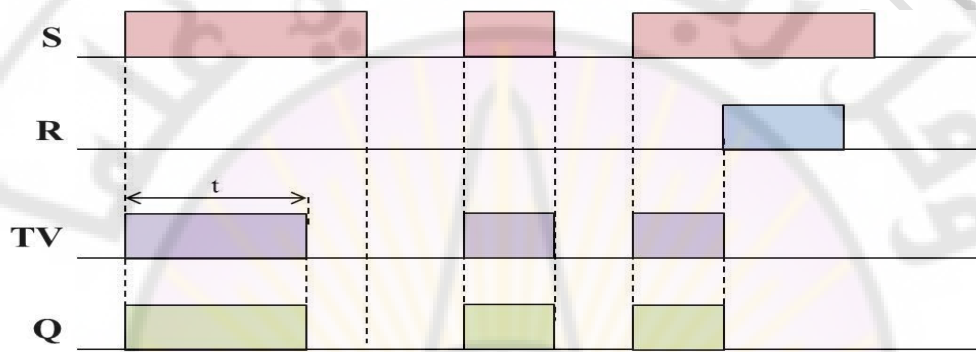
BI : قيمة الوقت الباقي وتمثل بعدد صحيح.

BCD : قيمة الوقت الباقي وتمثل بعدد ثنائي (ترميز BCD).



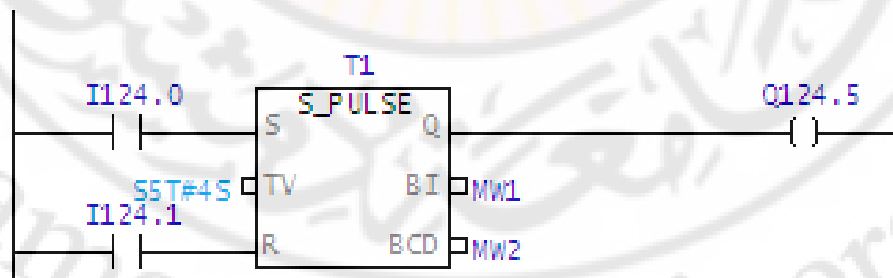
الشكل (2-3)

يمثل الشكل (2-3) رمز المؤقت النبضي، عند الجبهة الصاعدة للمدخل S (تغيير الإشارة من 0 إلى 1 منطقي) سيتحول المخرج Q إلى حالة (1 منطقي) ويبدأ المؤقت بعد قيمة الزمن المحدد على المدخل TV بشرط أن يستمر المدخل S على الحالة 1 وأن يكون المدخل R في حالة 0. سيتحول المخرج Q إلى الحالة التوقف (0 منطقي) عندما ينتهي المؤقت من عد الزمن المحدد أو عندما يصبح المدخل R في حالة 1 منطقي (الجبهة الصاعدة). إذا كان المؤقت في حالة عمل وتحول المدخل R من 0 إلى 1 منطقي (الجبهة الصاعدة) فإن المؤقت سيتوقف عن العمل وستتم إعادة عد الزمن من البداية، ويبين الشكل (2-4) المخطط الزمني لآلية عمل هذا المؤقت.



الشكل (2-4)

مثال :

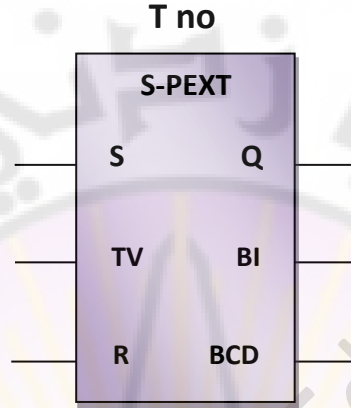


الشكل (2-5)

يظهر في الشكل (2-5) دائرة تستخدم المؤقت النبضي حيث عند تغيير الإشارة على المدخل 124.0 من 0 إلى 1 فإن المؤقت يبدأ بدورة الزمن المحدد والذي هو 4S بشرط أن يستمر

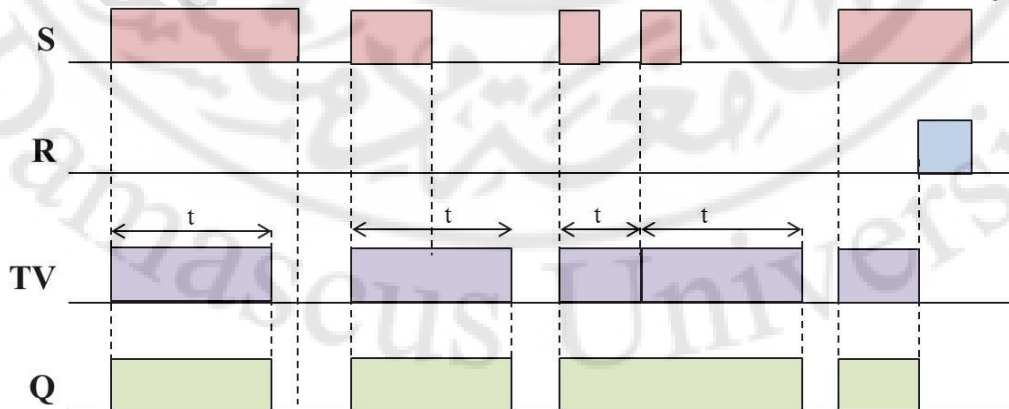
المدخل 124.0 على الحالة 1 ، لكن إذا كان المؤقت يعمل وتحول المدخل 124.0 من الحالة 1 إلى الحالة 0 أو تحول المدخل 124.1 من الحالة 0 إلى الحالة 1 فإن المؤقت سيتوقف عن العمل ويبقى المخرج 124.5 Q في حالة عمل مادام المؤقت في حالة عمل . سيتم تخزين قيمة الوقت الباقي في الذاكرة MW1 على شكل عشري وفي الذاكرة MW2 على شكل ثنائي (ترميز BCD) .

2-1-1- المؤقت ذو النبضة الممتدة : Extended Pulse Timer



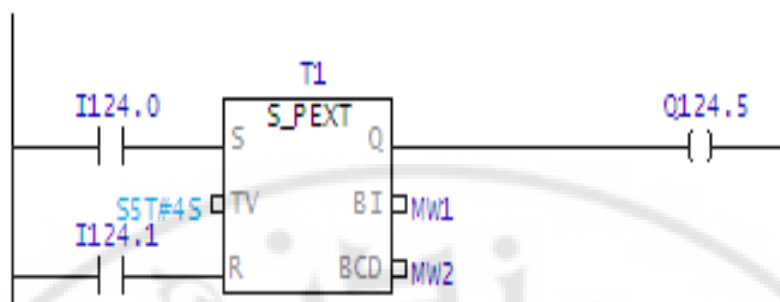
الشكل (6-2)

يمثل الشكل (6-2) رمز المؤقت النبضي ذو النبضة الممتدة، عند الجبهة الصاعدة للمدخل S (تغيير الإشارة من 0 إلى 1 منطقي) سيتحول المخرج Q إلى حالة (1 منطقي) ويبدأ المؤقت بعد قيمة الزمن المحدد على المدخل TV بشرط أن يبقى المدخل R في حالة 0. لن يتأثر المخرج Q إذا تغيرت الإشارة على المدخل إلى 0 منطقي (عكس المؤقت السابق). سيتحول المخرج Q إلى الحالة التوقف (0 منطقي) عندما ينتهي المؤقت من عد الزمن المحدد أو عندما يصبح المدخل R في حالة 1 منطقي (الجبهة الصاعدة). إذا كان المؤقت في حالة عمل وتحول المدخل R من 0 إلى 1 منطقي فإن المؤقت سيتوقف عن العمل ويعيد عد الزمن من البداية. إذا كان المؤقت في حالة عمل وتحول المدخل S إلى 0 منطقي ثم عاد إلى حالة 1 منطقي فإن المؤقت سيقوم بإعادة عد الزمن من البداية، يبين الشكل (7-2) المخطط الزمني لآلية عمل هذا المؤقت.



الشكل (7-2)

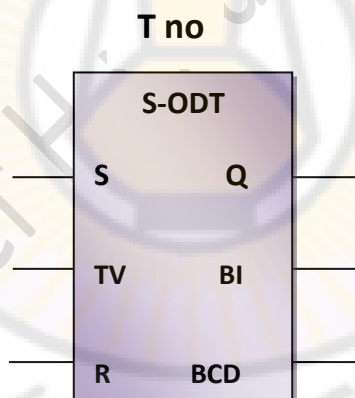
مثال :



الشكل (8-2)

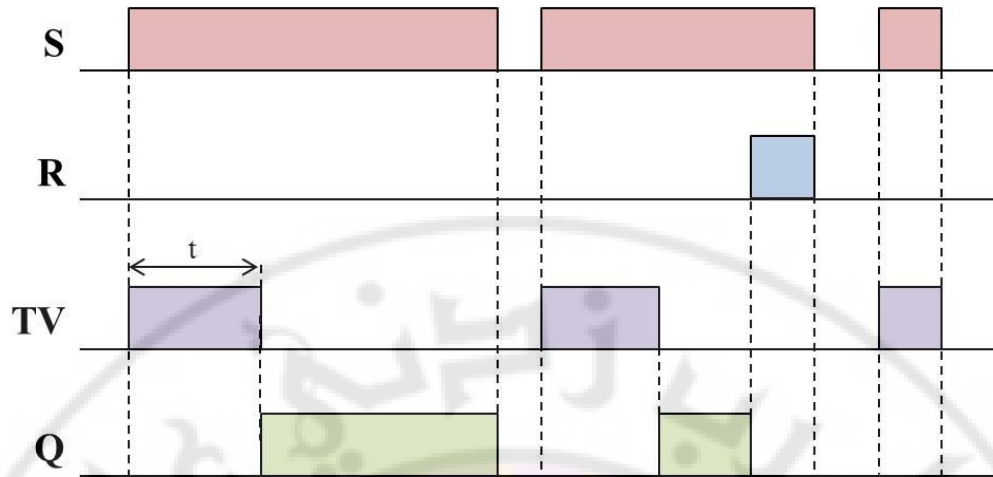
يظهر في الشكل (8-3) دائرة تستخدم المؤقت ذو النبضة الممتدة، فعند تغيير الإشارة على المدخل 124.0 من 0 إلى 1 فإن المؤقت يبدأ بدورة الزمن المحدد والذي هو 4S دون أن يتأثر بتحول المدخل 124.0 من الحالة 1 إلى الحالة 0 ، ولكن إذا كان المؤقت يعمل وتحول المدخل 124.0 من الحالة 0 إلى الحالة 1 مرة أخرى ، فإن المؤقت سيحسب الزمن من جديد، إذا تحول المدخل 124.1 من الحالة 0 إلى الحالة 1 فإن المؤقت سيتوقف عن العمل. سيتم تخزين قيمة الوقت الباقي في الذاكرة MW1 على شكل عشري وفي الذاكرة MW2 على شكل ثنائي (ترميز BCD).

3-1-1- مؤقت تأخير الوصل On - Delay Timer :



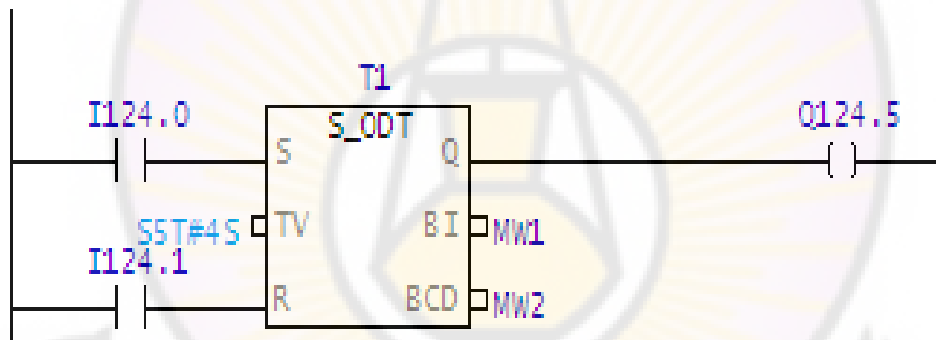
الشكل (9-2)

يمثل الشكل (9-2) رمز مؤقت تأخير ،وصف، عد ،جبهة الصاعدة للمدخل S (تغيير الإشارة من 0 إلى 1 منطقي) سيبدأ المؤقت بعد قيمة الزمن المحدد على المدخل TV بشرط أن يبقى المدخل R في حالة 0 منطقي والمدخل S بحالة 1 منطقي . يتحول المخرج Q إلى حالة 1 منطقي عندما ينتهي المؤقت من عد الزمن المحدد. يتحول المخرج Q إلى حالة 0 منطقي عند تحول المدخل R إلى 1 منطقي (الجبهة الصاعدة) أو عندما يتحول المدخل S إلى حالة 0 منطقي (الجبهة الهابطة). إذا كان المؤقت في حالة عد الزمن وتحول المدخل S إلى 0 منطقي فإن المؤقت سيتوقف عن عد الزمن، يبين الشكل (9-3) المخطط الزمني لآلية عمل هذا المؤقت.



الشكل (10-2)

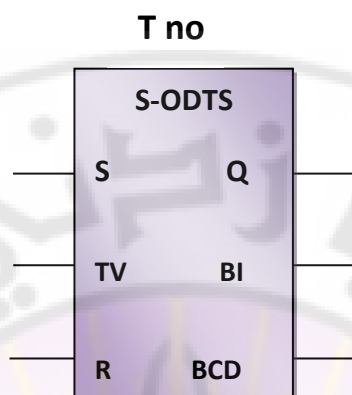
مثال :



الشكل (11-2)

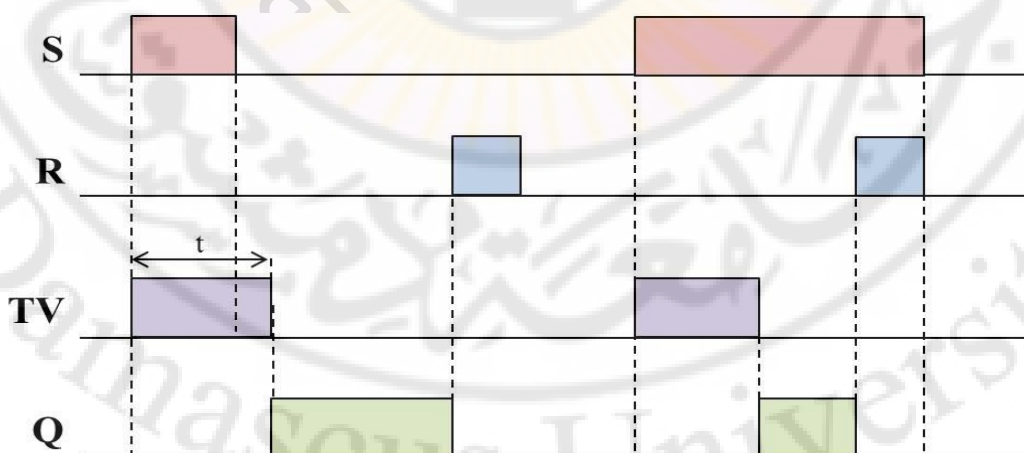
يظهر في الشكل (11-2) دائرة تستخدم مؤقت تأخير الوصل، إذا تغيرت الإشارة على المدخل 124.0 من 0 إلى 1 فإن المؤقت يبدأ بدورة الزمن المحدد والذي هو 4S، وعند الانتهاء من حساب الزمن المحدد فإن المخرج 124.5 يصبح على الوضع 1 بشرط أن يستمر المدخل 124.0 على الحالة 1، ولكن إذا كان المؤقت يعمل وتحول المدخل 124.0 من الحالة 1 إلى الحالة 0 فإن المؤقت سيتوقف عن العمل ويصبح المخرج 124.5 على الوضع 1. إذا تحول المدخل 124.1 من الحالة 0 إلى الحالة 1 والمؤقت يعمل فإنه سيتم حساب التوقيت من البداية.

4-1-1 - مؤقت تأخير الوصل الثابت :Retentive On - Delay Timer



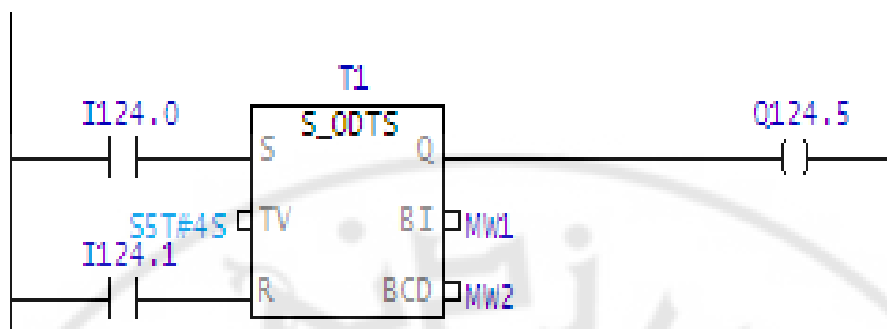
الشكل (12-2)

يمثل الشكل (12-2) رمز مؤقت تأخير الوصل الثابت، عند الجبهة الصاعدة للمدخل S (تغيير الإشارة من 0 إلى 1 منطقي) يبدأ المؤقت بعدد قيمة الزمن المحدد على المدخل TV بشرط أن يبقى المدخل R في حالة 0 ولن يتأثر بتحول المدخل S إلى حالة 0 منطقي. يتحول المخرج Q إلى الحالة 1 منطقي عندما ينتهي المؤقت من عد الزمن المحدد. يتحول المخرج Q إلى حالة 0 منطقي عندما يتحول المدخل R إلى 1 منطقي (الجبهة الصاعدة). إذا كان المؤقت في حالة عمل وتحول المدخل S إلى 0 منطقي ثم عاد إلى حالة 1 منطقي فإن المؤقت سيقوم بإعادة عد الزمن من البداية، يبين الشكل (13-2) المخطط الزمني لآلية عمل هذا المؤقت.



الشكل (13-2)

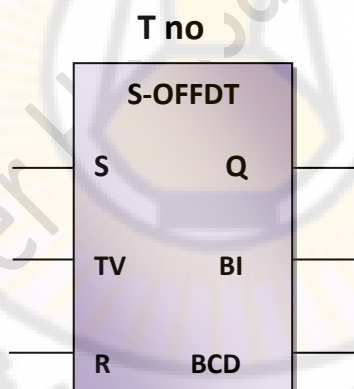
مثال



الشكل (14-2)

يظهر في الشكل (14-2) دائرة تستخدم مؤقت تأخير الوصل، إذا تغيرت الإشارة على المدخل 124.0 من 0 إلى 1 فإن المؤقت يبدأ بدورة الزمن المحدد والذي هو 4S وعند الانتهاء من حساب الزمن المحدد فإن الخرج 124.5 يصبح على الوضع 1 ولا يتأثر بتحول المدخل 124.0 من الحالة 1 إلى الحالة 0 ولكن إذا كان المؤقت يعمل وتحول المدخل 124.0 أو المدخل 124.1 من الحالة 0 إلى الحالة 1 فإنه سيتم حساب التوقيت من البداية.

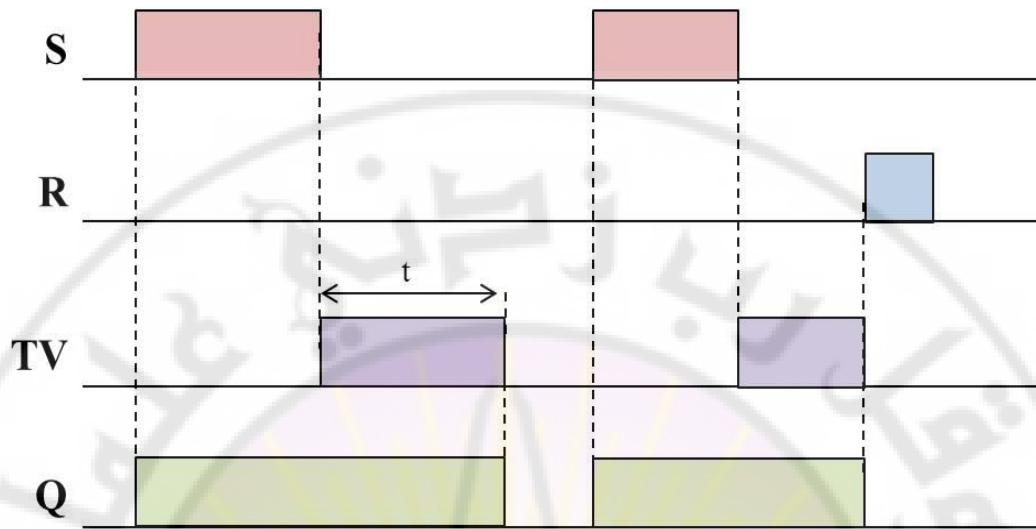
5-1-1- مؤقت تأخير الفصل Off - Delay Timer :



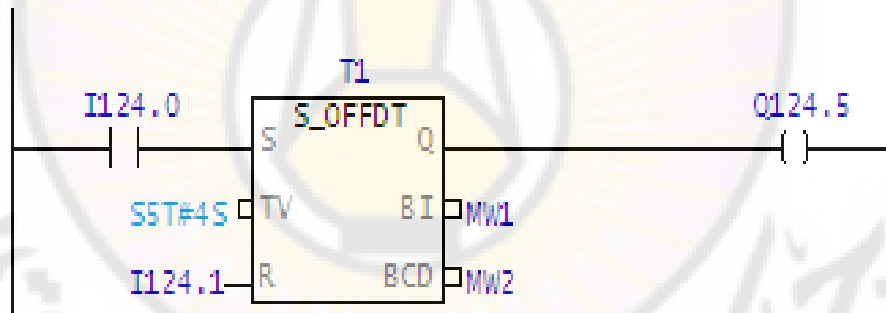
الشكل (15-2)

يمثل الشكل (15-2) رمز مؤقت تأخير الفصل، يتحول المخرج Q إلى الحالة 1 منطقي عند تغير الإشارة على المدخل S إلى 1 منطقي (الجهة الصاعدة). عند تغيير الإشارة على المدخل S إلى 0 منطقي (الجهة الهابطة) يبدأ المؤقت بعدد قيمة الزمن المحدد على المدخل TV بشرط أن يبقى المدخل R في حالة 0 منطقي. يتحول المخرج Q إلى حالة 0 منطقي عندما ينتهي المؤقت من عدد قيمة الزمن المحدد أو عندما يتحول المدخل R إلى 1 منطقي (الجهة الصاعدة).

إذا كان المؤقت في حالة عد الزمن المحدد وتحول المدخل R إلى 1 منطقي، فإن المؤقت سيقوم بإعادة تأهيل عداد الزمن (التصفير)، يبين الشكل (16-2) المخطط الزمني لآلية عمل هذا المؤقت.



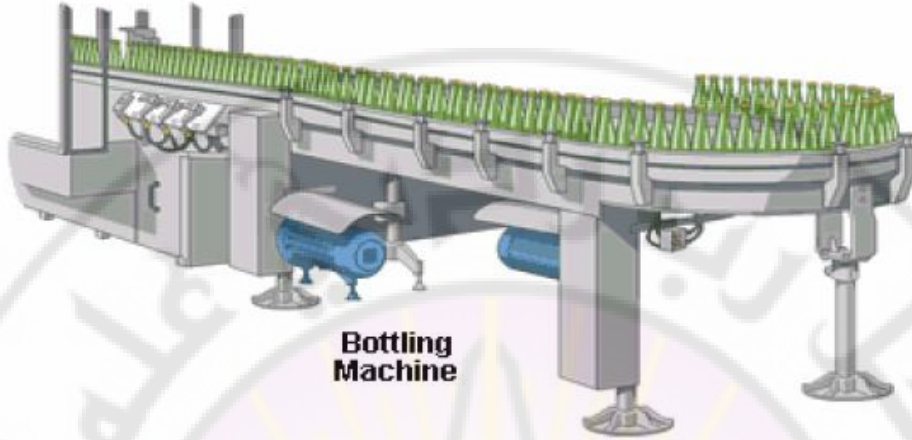
الشكل (16-2)



الشكل (17-2)

يظهر في الشكل (17-2) دائرة تستخدم مؤقت تأخير الفصل، فعند تغيير الإشارة على المدخل 124.0 من 1 إلى 0 فإن المؤقت يبدأ بدورة الزمن المحدد والذي هو 4S، يكون المخرج 124.5 على الوضع 1 عندما يكون المدخل 124.0 على الوضع 1 أو عندما يكون المؤقت في حالة عمل وعند انتهاء المؤقت من دورة الزمن المحدد فإن المخرج 124.5 سيكون في حالة 0 منطقي، إذا تحول المدخل 124.1 من 0 إلى 1 والمؤقت يعمل فإن المؤقت سيتوقف عن العمل ويصبح المخرج 124.5 في حالة 0.

2-1- العدادات (Counters) :



تستخدم العدادات لمعرفة عدد مرات حدوث ظاهرة معينة ضمن نظام التحكم ويتم ذلك عن طريق حساسات مناسبة تلائم العملية الإنتاجية وتستخدم العدادات بكثرة في عمليات التعبئة على خطوط الإنتاج. تستطيع العدادات في جهاز PLC نوع Siemens العد من 0 إلى 999 ولذلك فهي تقوم بحجز 1 Word من الذاكرة لتخزين قيمة العدّ فيها، ويعتمد عدد العدادات التي يمكن استخدامها في جهاز PLC حسب مواصفات وحدة المعالجة المركزية (CPU).

يتم تحميل قيمة العدّ حسب الشكل الآتي:
حيث:

VALUE: هي قيمة العدّ مثل: C # 60.

1-2-1- العداد التصاعدي التنازلي Up / Down Counter :

C no : رقم العداد المستخدم من الذاكرة .

CU : مدخل العدّ التصاعدي.

CD : مدخل العدّ

التنازلي.

تحميل أمر العداد.

مدخل تعريف قيمة العدّ.

مدخل تصفير العداد.

مخرج يمثل حالة العداد.

CV : مخرج يمثل قيمة العدّ الحالية وتكون بصيغة العدّ الستة عشري.

مخرج يمثل قيمة العدّ الحالية وتكون بصيغة العدّ الثنائي (ترميز BCD).

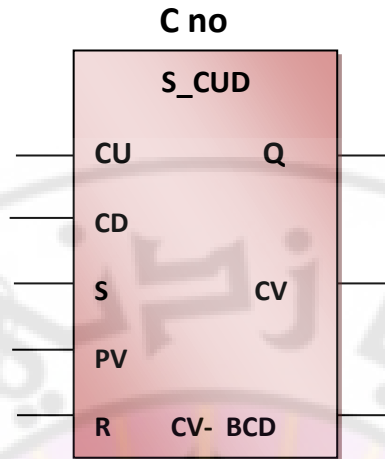
S : مدخل

PV :

R

Q :

BCD



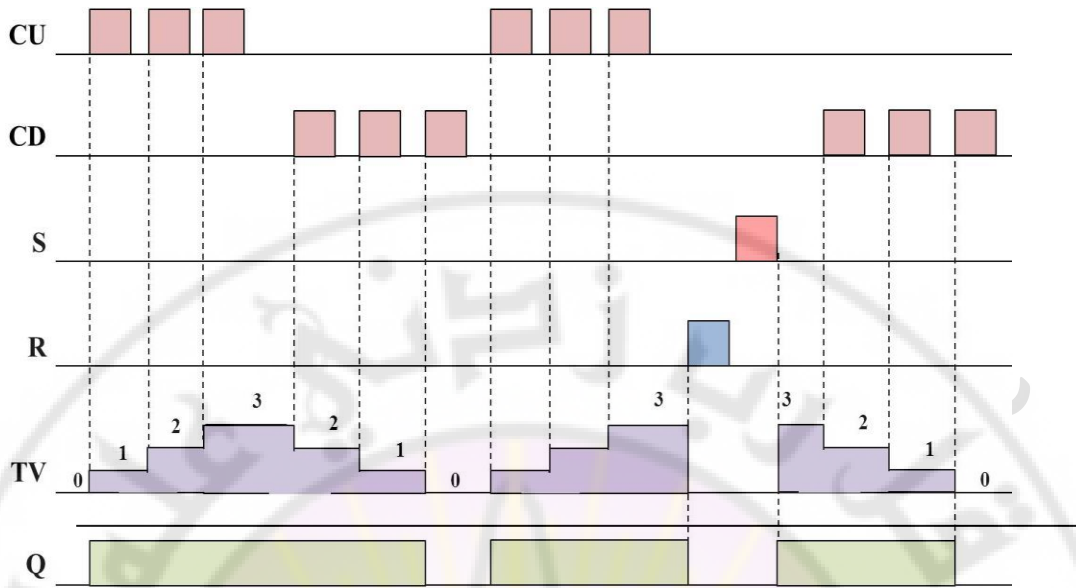
الشكل (2-18)

يمثل الشكل (2-18) رمز العداد التصاعدي التنازلي، عند تغيير الإشارة على المدخل S إلى الحالة 1 منطقي (الجهة الصاعدة) سيتم تحميل العداد C1 بقيمة العدّ المحددة على المدخل PV، وعند تغيير الحالة على المدخل CU إلى 1 منطقي (الجهة الصاعدة) فإن محتوى العداد سيزداد بمقدار واحد إلا إذا كانت قيمة العداد 999 فإنه سيتوقف عن العدّ، عند تغير الحالة على المدخل CD إلى 1 منطقي (الجهة الصاعدة) فإن محتوى العداد سينقص بمقدار واحد إلا إذا كانت قيمة العداد 0 فإنه سيتوقف عن العدّ، وإذا تغيرت الحالة على المدخل R إلى 1 منطقي (الجهة الصاعدة) سيتم إعادة ضبط قيمة العداد على الصفر.

يكون الخرج Q في الحالة 1 منطقي إذا كانت قيمة العداد أكبر من صفر.

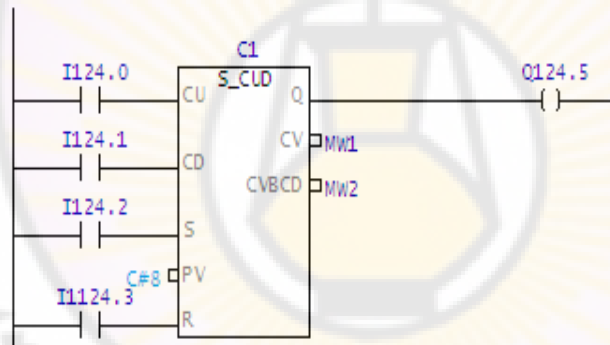
يكون الخرج Q في الحالة 0 منطقي إذا كانت قيمة العداد تساوي صفر.

يبين الشكل (3-18) المخطط الزمني لآلية عمل هذا العداد.



الشكل (19-2)

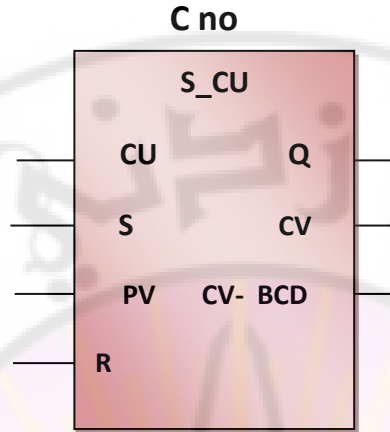
مثال :



الشكل (20-2)

يظهر في الشكل (20-2) دائرة تستخدم عداداً تصاعدياً تنازلياً، عند تغيير الإشارة على المدخل 124.2 من الحالة 0 إلى الحالة 1 فإنه سيتم تحميل العداد بقيمة العد المحددة والتي هي 8. عند تغيير الحالة على المدخل 124.0 من 0 إلى 1 فإن محتوى العداد سيزداد بمقدار واحد. عند تغيير الحالة على المدخل 124.1 من 0 إلى 1 فإن محتوى العداد سينقص بمقدار واحد. إذا تغيرت الحالة على المدخل 124.3 من 0 إلى 1 فإنه سيتم إعادة ضبط قيمة العداد على الصفر، يكون الخرج Q124.5 في الحالة 1 طالما أن قيمة العداد لا تساوي 0.

2-2-1- العداد التصاعدي Up Counter :



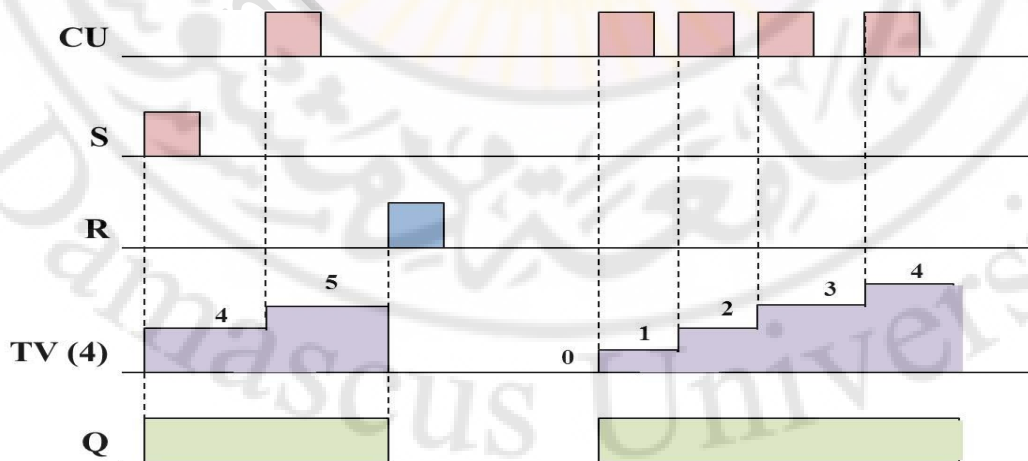
الشكل (21-2)

يمثل الشكل (21-2) رمز العداد التصاعدي، عند تغير الإشارة على المدخل S إلى الحالة 1 منطقي (الجهة الصاعدة) فإنه سيتم تحميل العداد C1 بقيمة العد المحددة على المدخل PV، وعند تغيير الحالة على المدخل CU إلى 1 منطقي سيزداد محتوى العداد بمقدار واحد إلا إذا كانت قيمة العداد 999، إذا تغيرت الحالة على المدخل R إلى 1 منطقي سيتم إعادة ضبط قيمة العداد على الصفر.

يكون الخرج Q في الحالة 1 منطقي إذا كانت قيمة العداد أكبر من صفر.

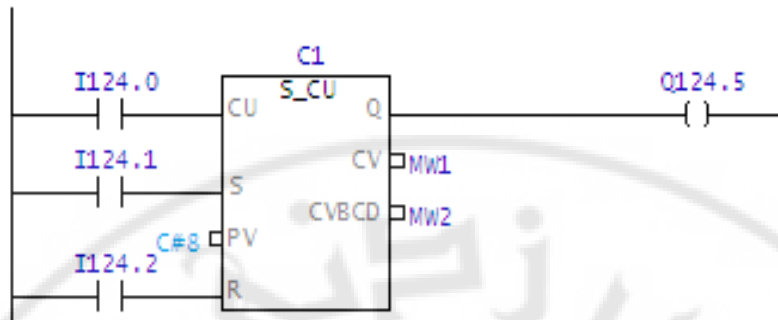
يكون الخرج Q في الحالة 0 منطقي إذا كانت قيمة العداد تساوي صفر.

يبين الشكل (22-2) المخطط الزمني لآلية عمل هذا العداد.



الشكل (22-2)

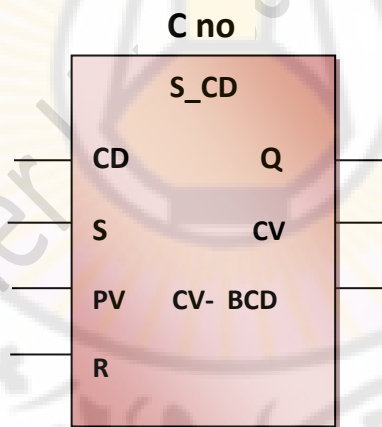
مثال :



الشكل (23-2)

يظهر في الشكل (23-2) دائرة تستخدم عداداً تصاعدياً، عند تغيير الإشارة على المدخل 124.1 من الحالة 0 إلى الحالة 1 فإنه سيتم تحميل العداد بقيمة العد المحددة والتي هي 8، وعند تغيير الحالة على المدخل 124.0 من 0 إلى 1 فإن محتوى العداد سيزداد بمقدار واحد إلا إذا كانت قيمة العداد 999، إذا تغيرت الحالة على المدخل 124.2 من 0 إلى 1 فإنه سيتم إعادة ضبط قيمة العداد على الصفر، يكون الخرج Q124.5 في الحالة 1 طالما أن قيمة العداد لا تساوي 0.

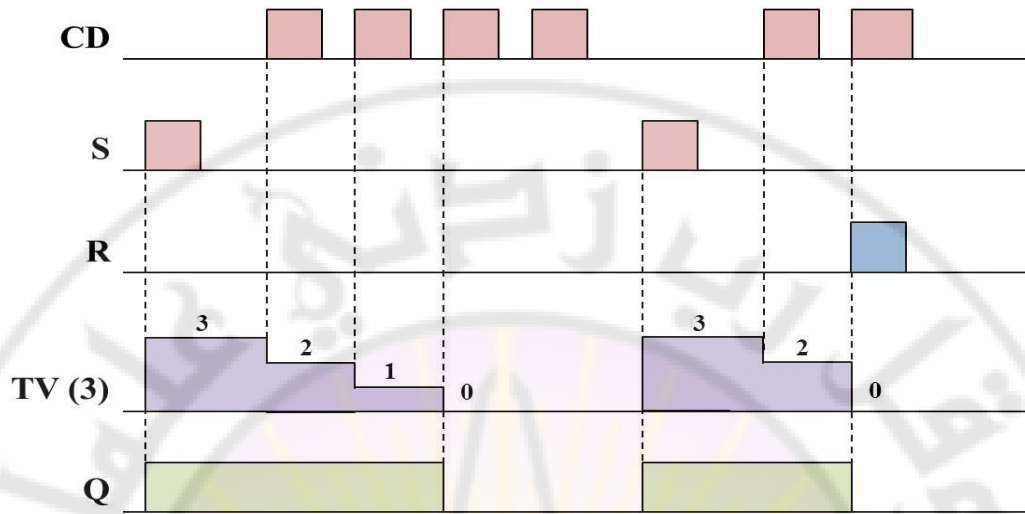
3-2-1 العداد التنازلي Down Counter:



الشكل (24-2)

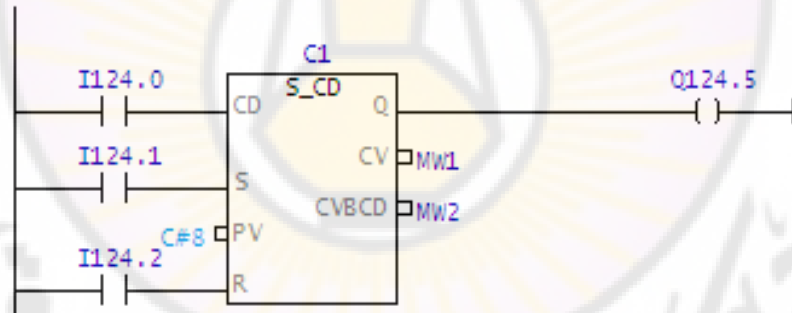
يمثل الشكل (24-2) رمز العداد التنازلي، عند تغيير الإشارة على المدخل S إلى الحالة 1 منطقي يتم تحميل العداد C1 بقيمة العد المحددة على المدخل PV، عند تغيير الحالة على المدخل CD إلى 1 منطقي فإن محتوى العداد ينقص بمقدار واحد إلا إذا كانت قيمة العداد 0 فإنه سيتوقف عن العد، إذا تغيرت الحالة على المدخل R من 0 إلى 1 فإنه سيتم إعادة ضبط قيمة العداد على الصفر. يكون الخرج Q في الحالة 1 منطقي إذا كانت قيمة العداد أكبر من صفر.

يكون الخرج Q في الحالة 0 منطقي إذا كانت قيمة العداد تساوي صفر.
يبين الشكل (24-3) المخطط الزمني لآلية عمل هذا العداد.



الشكل (25-2)

مثال :



الشكل (26-2)

يظهر في الشكل (26-2) دائرة تستخدم عداداً تنازلياً، عند تغيير الإشارة على المدخل 124.1 من الحالة 0 إلى الحالة 1 فإنه سيتم تحميل العداد بقيمة العد المحددة والتي هي 8، وعند تغيير الحالة على المدخل 124.0 من 0 إلى 1 فإن محتوى العداد سينقص بمقدار واحد إلا إذا كانت قيمة العداد 0، إذا تغيرت الحالة على المدخل 124.2 من 0 إلى 1 فإنه سيتم إعادة ضبط قيمة العداد على الصفر، يكون الخرج Q124.5 في الحالة 1 طالما أن قيمة العداد لا تساوي 0.

3-1 أوامر المقارنة (Comparison Instructions):

قد نحتاج في بعض الأحيان للمقارنة بين قراءة معينة تأتي من حساس وبين قيمة ثابتة، ومثال ذلك إذا أردنا المقارنة بين درجة حرارة فرن والتي سيقوم بقراءتها حساس الحرارة الموجود في الفرن وبين درجة الحرارة الذي يطلب عمل الفرن عندها، لذلك تحوي أجهزة PLC أوامر

المقارنة التي نستطيع من خلالها المقارنة بين عددين صحيحين أو بين عدد حقيقيين وهي على عدة أشكال حسب الآتي:

رمز عملية المقارنة	شكل عملية المقارنة
(القيمة 2) == (القيمة 1)	إذا كانت القيمة الأولى مساوية للقيمة الثانية
(القيمة 2) < > (القيمة 1)	إذا كانت القيمة الأولى لا تساوي القيمة الثانية
(القيمة 2) > (القيمة 1)	إذا كانت القيمة الأولى أكبر من القيمة الثانية
(القيمة 2) < (القيمة 1)	إذا كانت القيمة الأولى أصغر من القيمة الثانية
(القيمة 2) >= (القيمة 1)	إذا كانت القيمة الأولى أكبر أو تساوي القيمة الثانية
(القيمة 2) <= (القيمة 1)	إذا كانت القيمة الأولى أصغر أو تساوي القيمة الثانية

نستطيع من خلال هذه الأوامر المقارنة بين عددين صحيحين، وكما سبق معنا فإن العدد الصحيح يقوم بحجز 16 Bit من الذاكرة أي Word والذي يكون مجال الأعداد فيه من -32768 إلى +32767 أي يمكننا المقارنة بين 2 Word، كما نستطيع المقارنة بين عددين صحيحين مزدوجين فيقوم بحجز 32 Bit من الذاكرة أي 2Word والذي يكون مجال الأعداد فيه من -2147483648 إلى +2147483647، أما إذا أردنا المقارنة بين عددين حقيقيين فيقوم بحجز 32 Bit من الذاكرة، وسنتناول في بحثنا هذا المقارنة بين عددين صحيحين فقط.

1-3-1 المقارنة بين عددين صحيحين (Compare two integer) :

أ- عملية المقارنة يساوي (EQ_I) :

ويرمز له كما في الشكل (3-26) .

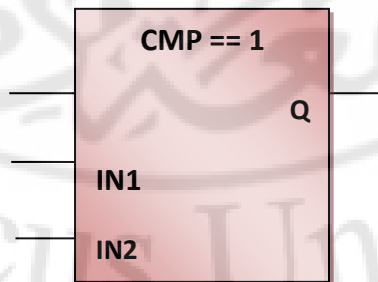
حيث: CMP = 1 : نمط عملية المقارنة .

IN1 : مدخل القيمة الأولى .

IN2 : مدخل القيمة الثانية .

Q : مخرج نتيجة المقارنة.

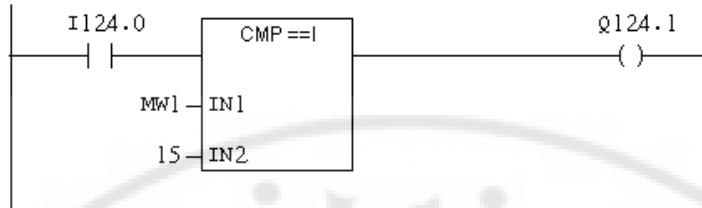
أما المدخل الغير معنون يستخدم لتفعيل التعليلة.



الشكل (2-27)

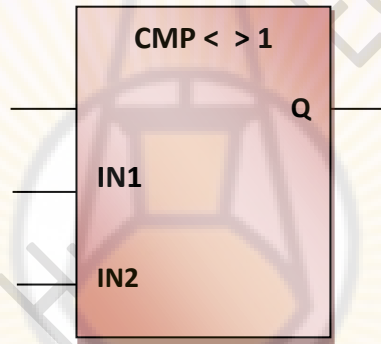
يكون المخرج Q في حالة 1 منطقي إذا كانت القيمة على المدخل IN1 تساوي القيمة على المدخل IN2.

مثال :



الشكل (28-2)

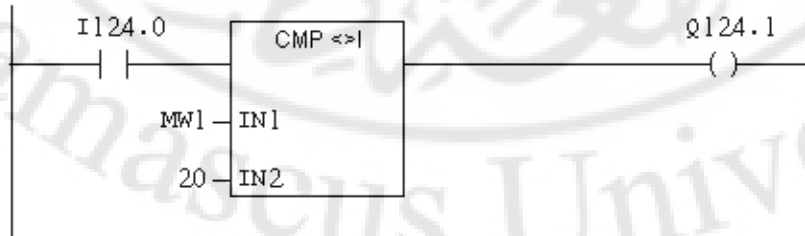
يظهر في الشكل (28-2) دائرة مقارن تستخدم مقارن لعملية (يساوي)، عند تفعيل المدخل I124.0 ستفعل عملية المقارنة، فإذا كانت القيمة المخزنة في الذاكرة MW1 تساوي 15 سيتم تفعيل المخرج Q124.1. عند عدم تساوي القيمة الموجودة في الذاكرة MW1 سيتم إيقاف تفعيل المخرج Q124.1. ب- عملية المقارنة لا يساوي (NE_I) :



الشكل (29-2)

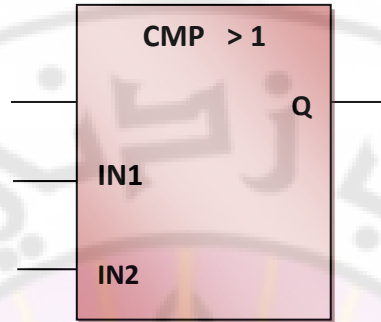
يرمز له كما في الشكل (29-2)، يكون المخرج Q في حالة 1 منطقي إذا كانت القيمة على المدخل IN1 لا تساوي القيمة على المدخل IN2.

مثال :



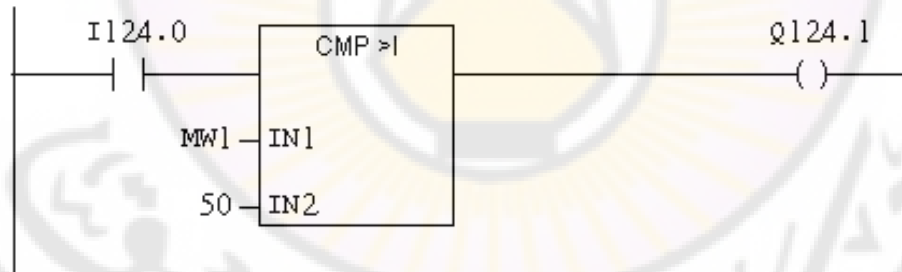
الشكل (30-2)

يظهر في الشكل (29-3) دائرة مقارن تستخدم لعملية (لا يساوي)، عند تفعيل المدخل I124.0 سيتم تفعيل عملية المقارنة فإذا كانت القيمة المخزنة في الذاكرة MW1 لا تساوي 20 سيتم تفعيل المخرج Q124.1.
إذا كانت قيمة الذاكرة MW1 تساوي 20 سيتم إيقاف تفعيل المخرج Q124.1.
ت- عملية المقارنة أكبر من (GT_I) :



الشكل (31-2)

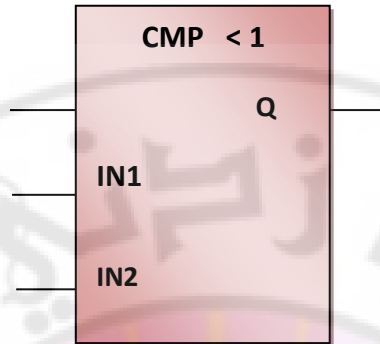
يرمز له كما في الشكل (31-2) ، يكون المخرج Q في حالة 1 منطقي إذا كانت القيمة على المدخل IN1 أكبر من القيمة على المدخل IN2 .
مثال :



الشكل (32-2)

يظهر في الشكل (32-2) دائرة مقارن تستخدم لعملية (أكبر من)، عند تفعيل المدخل I124.0 فستفعل عملية المقارنة فإذا كانت الذاكرة MW1 أكبر من 50 فإنه سيتم تفعيل المخرج Q124.1.
إذا كانت قيمة الذاكرة MW1 أصغر أو تساوي 50 سيتم إيقاف تفعيل المخرج Q124.1

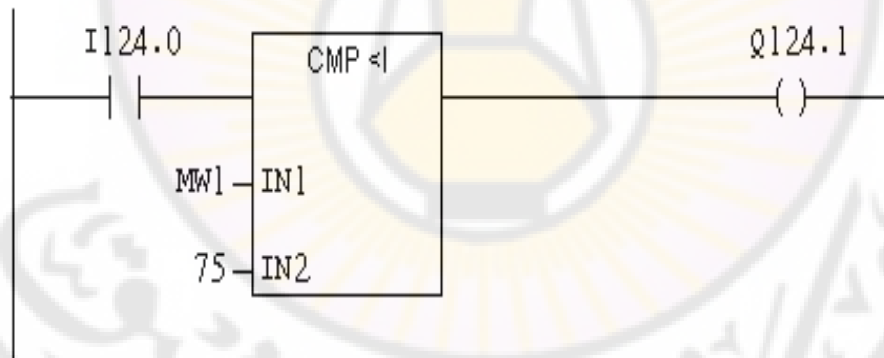
ث- عملية المقارنة أصغر من (LT_I) :



الشكل (2-33)

يرمز له كما في الشكل (2-33)، يكون المخرج Q في حالة 1 منطقي إذا كانت القيمة على المدخل IN1 أصغر من القيمة على المدخل IN2.

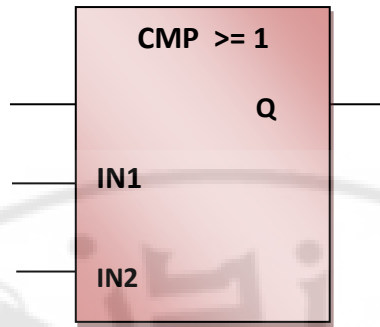
مثال :



الشكل (2-34)

يظهر في الشكل (2-34) دائرة مقارن تستخدم لعملية (أصغر من)، عند تفعيل المدخل I124.0 فستفعل عملية المقارنة فإذا كانت القيمة المخزنة في الذاكرة MW1 أصغر من 75 فإنه سيتم تفعيل المخرج Q124.1. إذا كانت قيمة الذاكرة MW1 أكبر أو تساوي 75 سيتم إيقاف تفعيل المخرج Q124.1 .

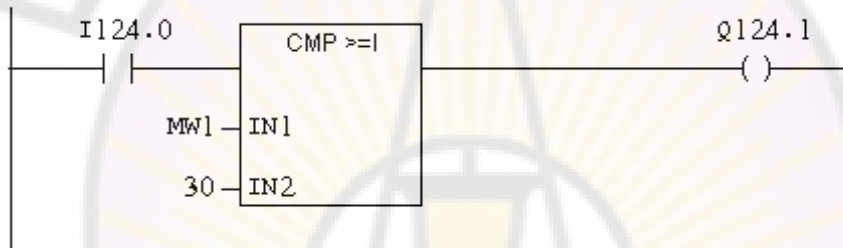
ج- عملية المقارنة أكبر أو تساوي (GE_I) :



الشكل (35-2)

يرمز له كما في الشكل (35-2)، يكون المخرج Q في حالة 1 منطقي إذا كانت القيمة على المدخل IN1 أكبر أو تساوي القيمة على المدخل IN2.

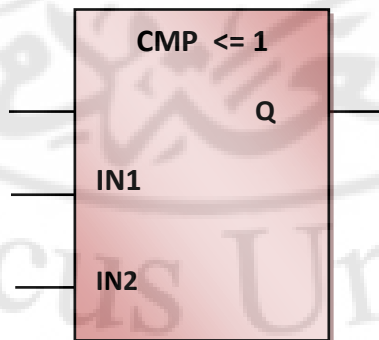
مثال :



الشكل (36-2)

يظهر في الشكل (36-2) دائرة مقارن تستخدم لعملية (أكبر أو يساوي)، عند تفعيل المدخل I124.0 فستفعل عملية المقارنة فإذا كانت القيمة المخزنة في الذاكرة MW1 أكبر أو تساوي 30 فإنه سيتم تفعيل المخرج Q124.1. إذا كانت قيمة الذاكرة MW1 أصغر من 30 فإنه سيتم إيقاف تفعيل المخرج Q124.1.

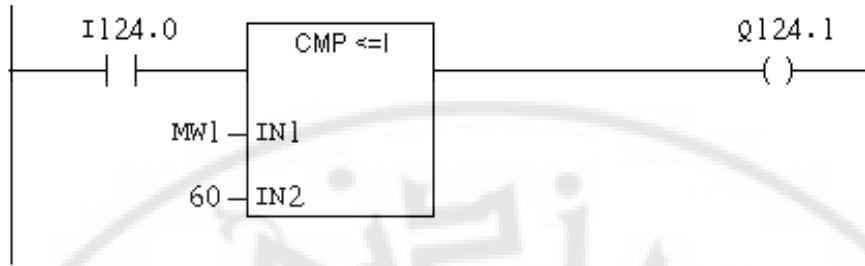
ج- عملية المقارنة أصغر أو تساوي (LE_I) :



الشكل (37-2)

يرمز له كما في الشكل (36-3)، يكون المخرج Q في حالة 1 منطقي إذا كانت القيمة على المدخل IN1 أصغر أو تساوي القيمة على المدخل IN2 .

مثال •



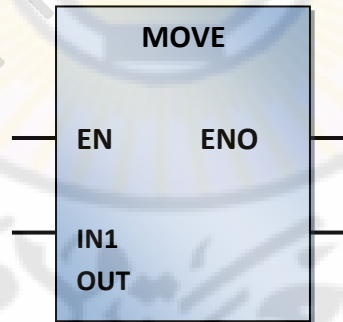
الشكل (38-2)

يظهر في الشكل (37-3) دائرة مقارن تستخدم لعملية (أصغر أو يساوي)، عند تفعيل المدخل I124.0 فستفعل عملية المقارنة فإذا كانت القيمة المخزنة في الذاكرة MW1 أصغر أو تساوي 60 سيتم تفعيل المخرج Q124.1 .
إذا كانت قيمة الذاكرة MW1 أكبر من 60 فإنه سيتم إيقاف تفعيل المخرج Q124.1

4-1- تعليمة النقل (MOVE) :

يرمز له كما في الشكل (39-2) ، حيث :

- EN : مدخل تفعيل التعليمة.
- IN1 : القيمة أو الموقع المراد النسخ منه (المصدر) .
- ENO : مخرج يعمل عند تنفيذ العملية .
- OUT : الموقع المراد النسخ إليه (الوجهة) .



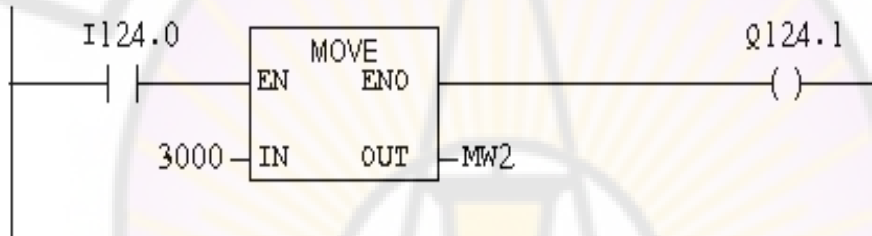
الشكل (39-2)

تستخدم هذه التعليمة لتحميل موقع من الذاكرة بقيمة معينة أو لنسخ قيمة من موقع في الذاكرة إلى موقع آخر وتعتبر التعليمة MOV أساسية عند التعامل مع مداخل أو مخرجات تماثلية (analog)، يمكن نسخ المعطيات الآتية: (Byte – Word – Double Word)، فيما يلي نبين كيفية نسخ أنواع مختلفة من المعطيات.

مثال : Double Word : 0101010101 00001111 11110000
 11111111
 الحالة بعد عملية النقل أو النسخ :
 0101010101 00001111 11110000 : Double word
 : 11111111 Byte
 0101010101
 0101010101 00001111 : Word

مثال Byte :
 الحالة بعد عملية النقل أو النسخ : 00001111
 00001111 : Byte
 00001111 00000000 : Word
 00001111 00000000 00000000 00000000 : Double word

مثال :



الشكل (40-2)

يظهر في الشكل (40-2) دائرة تستخدم عملية النقل، عند تفعيل المدخل I124.0 فإنه سيتم نسخ القيمة 3000 إلى موقع الذاكرة MW2 وعندما تفعيل عملية النسخ سينقل المخرج Q124.5 إلى الحالة 1.