

BÖLÜM 4

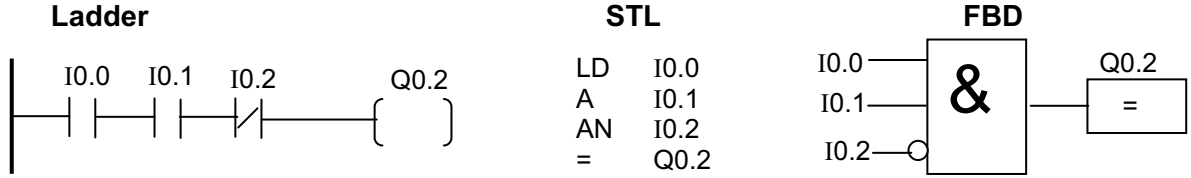
PLC' DE PROGRAMLAMA MANTIĞI

4.1 - Giriş : PLC' de programlamayı 3 değişik şekilde yapmak mümkündür.

- 1 - Ladder diyagram (merdiven diyagramı veya kontak plan ile yapılan programlama)
- 2 - Komut listesi ile yapılan programlama (STL = Statement list editör)
- 3 - Fonksiyon blok diyagramı ile programlama (FBD = Function Block Diagram)

Bu programlar arasında elektrikçilerin en çok kullandığı program ladder programıdır. Bu programda devrenin çalışmasını izlemek daha kolaydır. Bu program gerçek bağlantıyı verir. Elektronik mesleğinden olanlar daha çok fonksiyon blok diyagramına yatkındırlar. Bilgisayarlılar ve mühendisler ise komut listesi programına yatkındırlar. Ancak şu da bilinmelidir ki ladder diyagramda hazırlanan bir projeyi program, STL ve FBD yazılımlarına kendiliğinden çevirmektedir. Program hangi yöntemle yazılırsa yazılsın diğer yazılım şekillerini bilgisayar ekranından görmek mümkündür. Ancak bu uygulama her marka PLC' de yoktur.

Örnek :



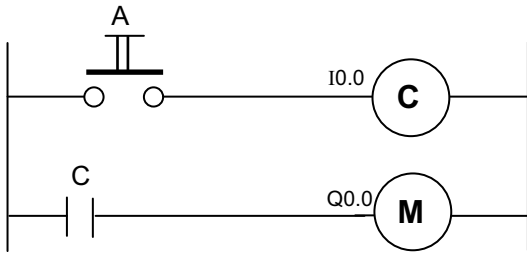
LOAD (LD) = yüklemek
= aktarmak-

1. nesil PLC' lerde (MicroWIN 16 kullanıldığında) ana program, koşulsuz bir END (STL' de MEND) komutu ile sonlandırılır.

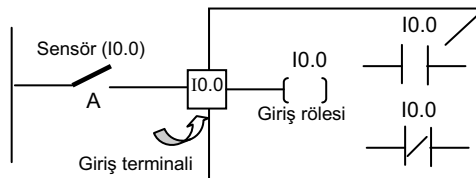
2. nesil PLC' lerde (Micro WIN 32 kullanıldığında) ana programın sonunda END komutuna gerek yoktur (kullanılırsa hata verir).

Örnek :

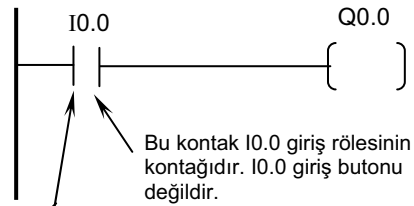
a) Klasik kumanda devresi



A butonuna basılınca C kontaktörü çalışır. Açık kontağını kapatarak motoru çalıştıracak olan M kontaktörü enerjilenir ve motor çalışır.



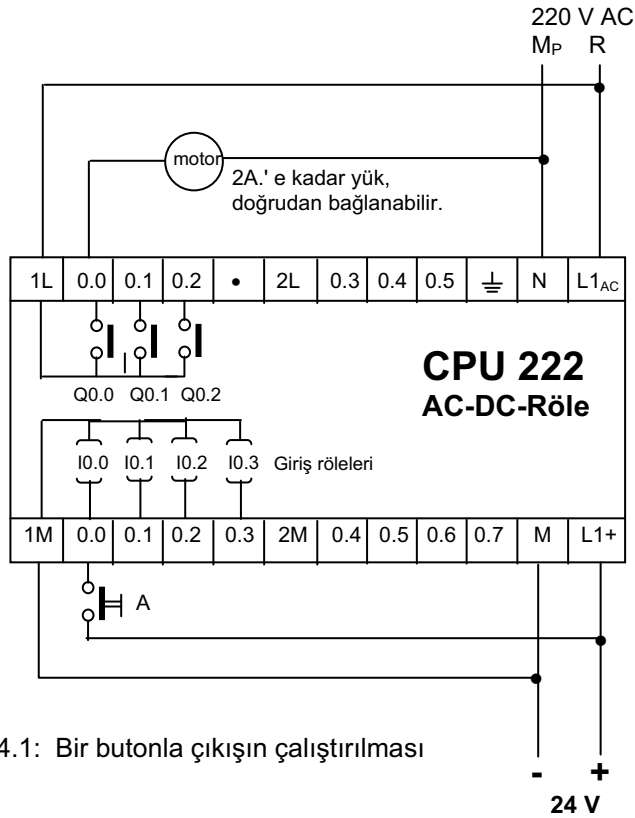
b) Ladder devresi



Bu kontak I0.0 giriş rölesinin kontağıdır. I0.0 giriş butonu değildir.

I0.0 butonuna basılınca I0.0 giriş rölesi enerjilenir. I0.0 giriş rölesi kontağını kapatarak Q0.0 çıkış rölesini enerjiler. Q0.0 çıkış verir ve alıcı çalışır. I0.0' daki sinyal kesilirse Q0.0 ' ın da çıkışı kesilir.

PLC bağlantısı :



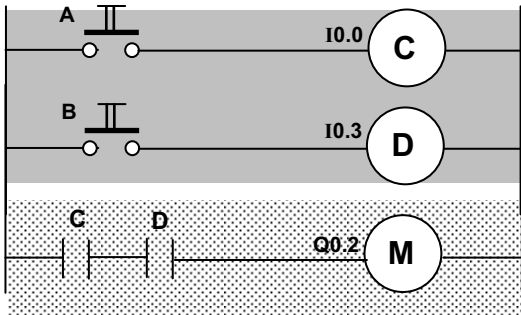
UYARI !
CPU 222 DC-DC-DC
Modellerde 24 V DC
uygulanmalıdır.

Şekil 4.1: Bir butonla çıkışın çalıştırılması

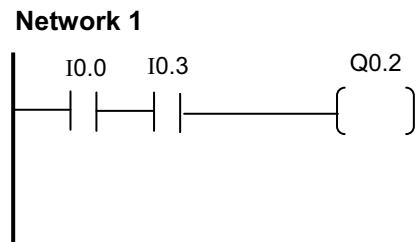
Sensörler ve genişletme
Modülleri için
(**Buraya gerilim vermeyiniz**)

4.2 - VE mantığı :

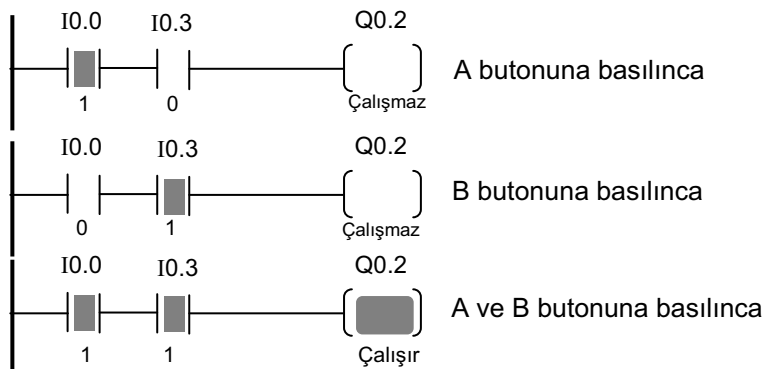
a) Klasik kumanda devresi



b) Ladder devresi



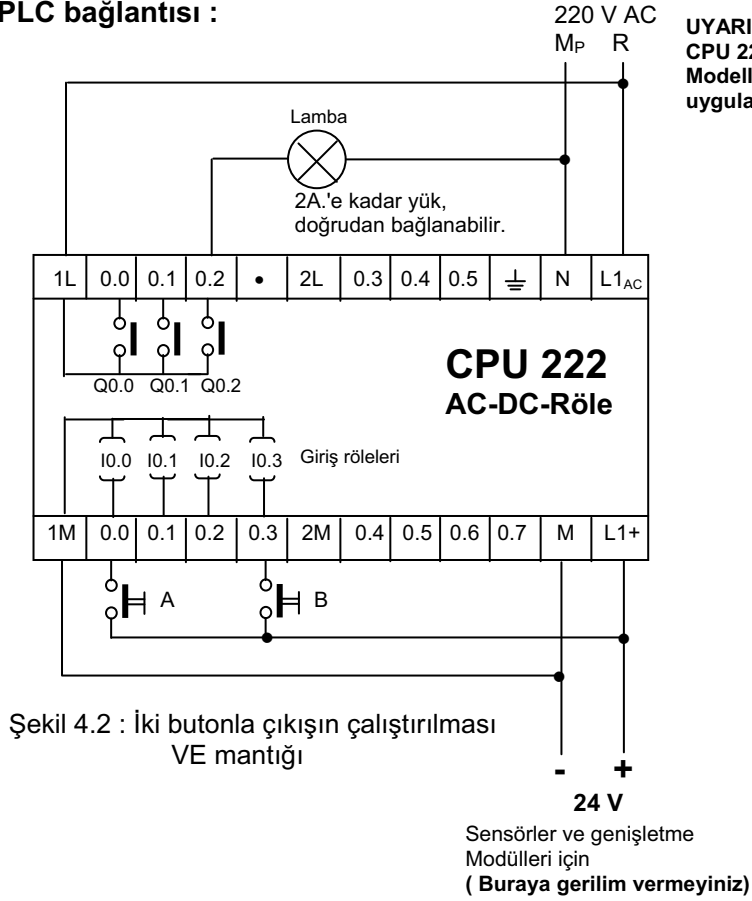
Ladder çalışmada aşağıdaki ihtimaller söz konusudur.



c) STL (Komut listesi)
Network 1

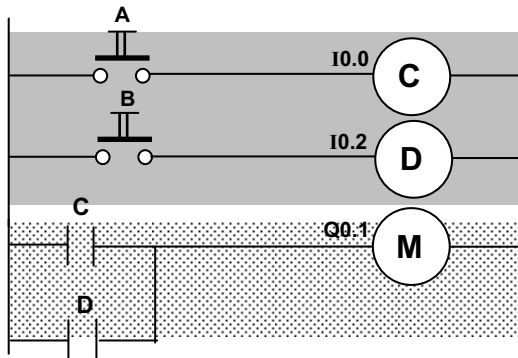
LD 10.0
A 10.3 A → AND
= Q0.2

PLC bağlantısı :

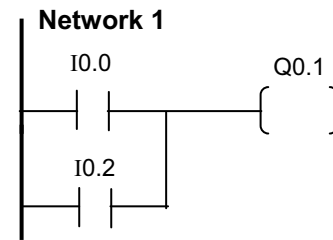


4.3 - VEYA mantığı :

a) Klasik kumanda devresi

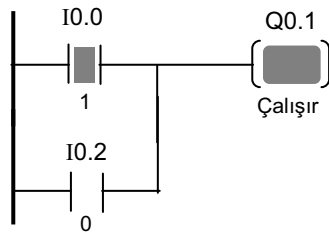


b) Ladder devresi

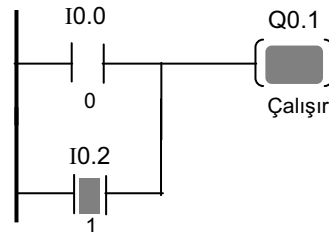


c) STL (Komut listesi)
Network 1
 LD I0.0
 O I0.2 O → OR
 = Q0.1

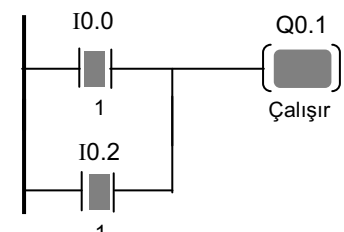
Ladder çalışmada aşağıdaki ihtimaller söz konusudur.



A butonuna basılınca

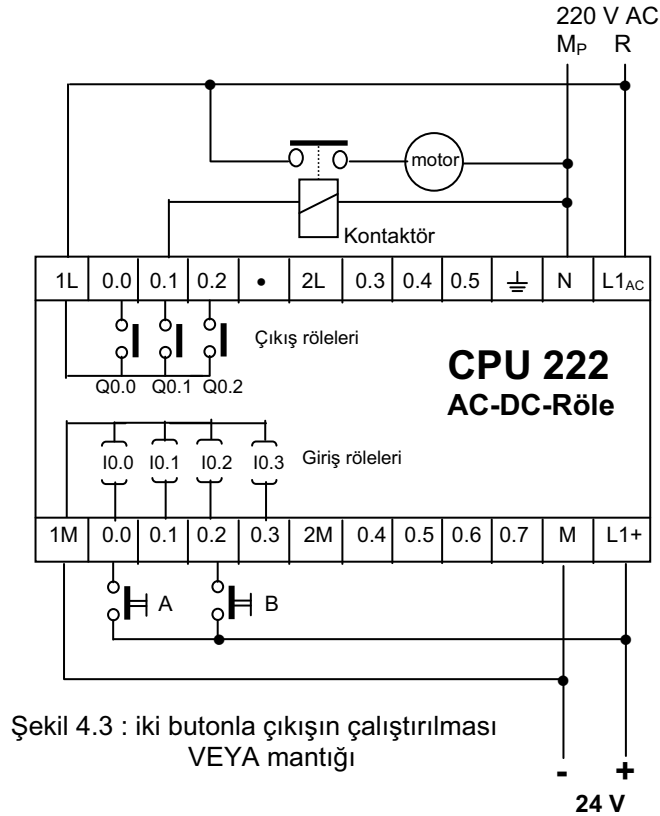


B butonuna basılınca



A ve B butonuna basılınca

PLC bağlantısı :



UYARI !
CPU 222 DC-DC-DC
Modellerde 24 V DC
uygulanmalıdır.

2A.' den büyük yükler, röle veya kontaktör üzerinden sürülmelidir.

Sensörler ve genişletme
Modülleri için
(**Buraya gerilim vermeyiniz**)

VEYA mantığında, örneğin A butonuna basılırsa I0.0 giriş rölesi aktif olur (enerjilenir). CPU, verilen komuta göre hangi çıkış rölesini aktif yapacağına karar verir. Burada I0.0 giriş rölesinin Q 0.1 çıkış rölesini aktif yapması (çalıştırması) programlanmıştır. Dolayısıyla I0.0 giriş rölesi aktif (1) olduğunda Q0.1 çıkış rölesi çalışır. Q0.1 çıkış rölesi kontağını kapatarak kontaktörün çalışmasını sağlar. Kontaktör kontağını kapatarak motoru çalıştırır.

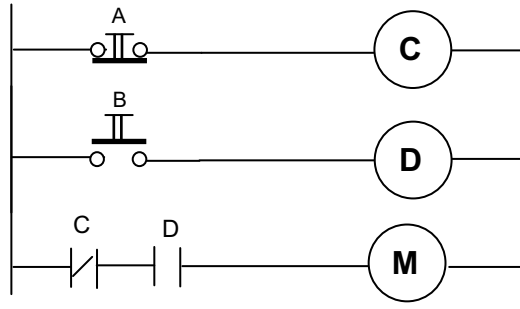
A butonu serbest bırakıldığında I0.0 giriş rölesi pasif olur (çalışmaz). Buna bağlı olarak Q0.1 çıkış rölesinin enerjisi kesilir. Q0.0 çıkış rölesi kontağını açar kontaktör ve motor çalışmaz.

Aynı çalışma şekli B butonu için de geçerlidir. Ancak B butonuna basıldığında I0.2 giriş rölesi enerjilenir ve I0.2 giriş rölesi, Q0.1 çıkış rölesini çalıştırır. Q0.1 çıkış rölesinin çalışmasıyla yukarıda anlatılan olaylar tekrarlanır.

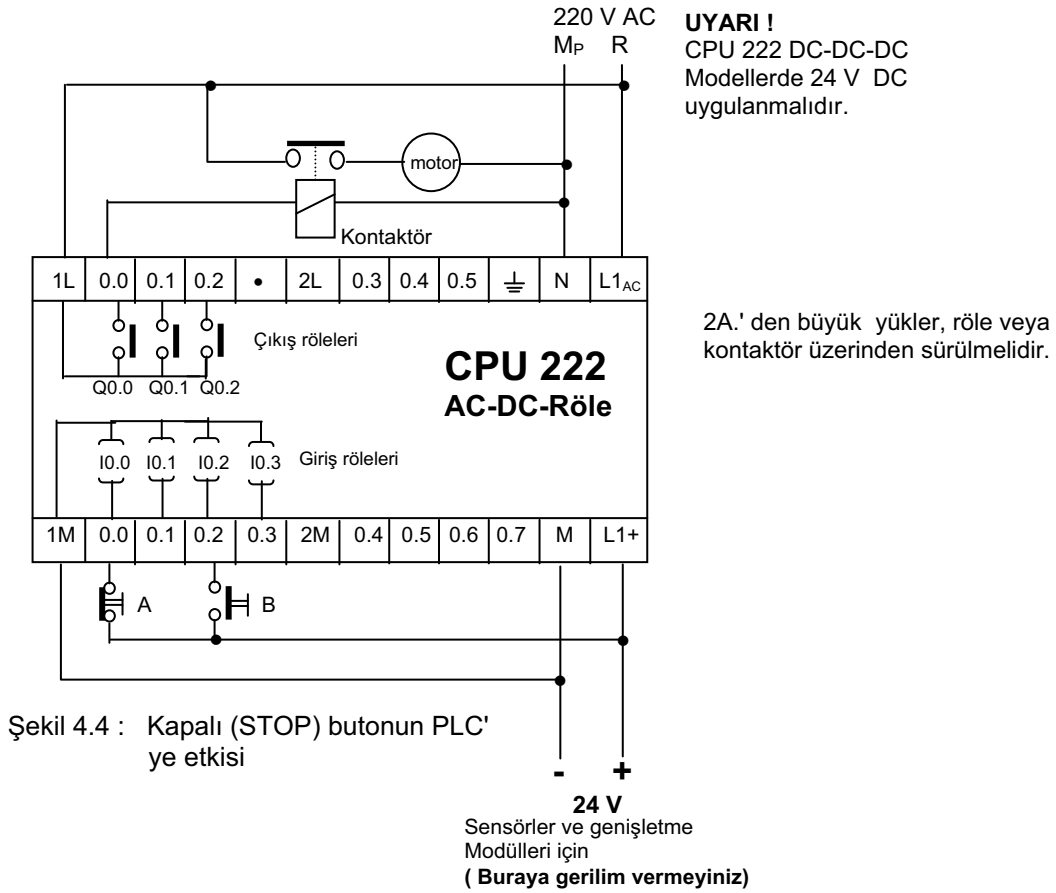
VE mantığında ise A ve B butonlarının her ikisine aynı anda basılmadığı sürece çıkış rölesi ve yük çalışmaz.

4.4 - Stop butonunun (kapalı kontağın) PLC' ye etkisi :

a) Klasik kumanda devresi :

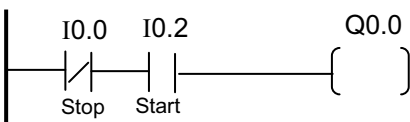


b) PLC bağlantısı :



Şekil 4.4' deki PLC bağlantısında A butonu kapalı olduğundan I0.0 giriş rölesi kullanıcının herhangi bir müdahalesi olmadan enerjilenir. I0.0 giriş rölesi, ladder devresindeki kapalı kontağını açar. Bu durumda B butonuna (start) basılırsa I0.2 giriş rölesi enerjilenir ve kontağını kapatır. Ancak I0.0 giriş rölesinin kontağı açık olduğundan Q0.0 çıkış rölesi enerjilenemez. Dolayısıyla kumanda sistemi, kumanda mantığına göre doğru çalışmamış olur (ladder devresi a).

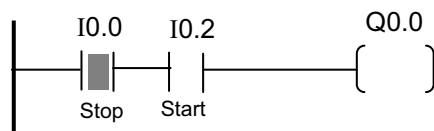
Network 1



(a)

c) Ladder devresi

Network 1



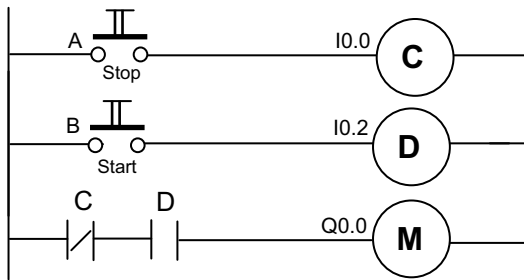
(b)

Ladder devresi yukarıda b' deki gibi tasarlanırsa, yani I0.0 açık olarak kullanılırsa, PLC'ye enerji verildiğinde I0.0 giriş rölesi enerjilenir ve ladder devresindeki I0.0 kontağını kapatır. B butonuna basıldığında I0.2 giriş rölesi enerjilenir ve Q0.0 çıkış rölesi çalışır. B butonuna basılı iken A butonuna basıldığında I0.0 giriş rölesinin enerjisi kesilir. Ladder devresindeki I0.0 kontağını açar, Q0.0 çıkış rölesinin çalışması durur. Burada dikkat edilirse kapalı kontaklı STOP butonu (A), ladder devresinde açık kontak olarak kullanıldı. Bu durum kumanda mantığına aykırı olabilir. Bu nedenle:

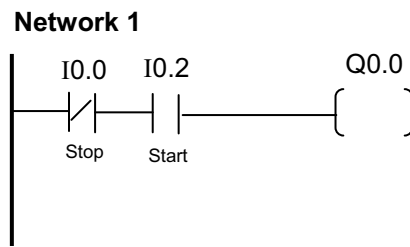
PLC girişine bağlanacak olan tüm sensörlerin (start butonu, stop butonu , aşırı akım rölesi kontağı, ısı sensörü, ışık sensörü, optik sensör, manyetik sensör,vb), açık kontaklı kullanılması kumanda mantığına daha uygun olur.

Ancak sistemin durdurma butonuna basıldığı halde durdurulamaması halinde hayati ve maddi tehlikeler söz konusu olduğunda stop için kapalı buton kullanılabilir.

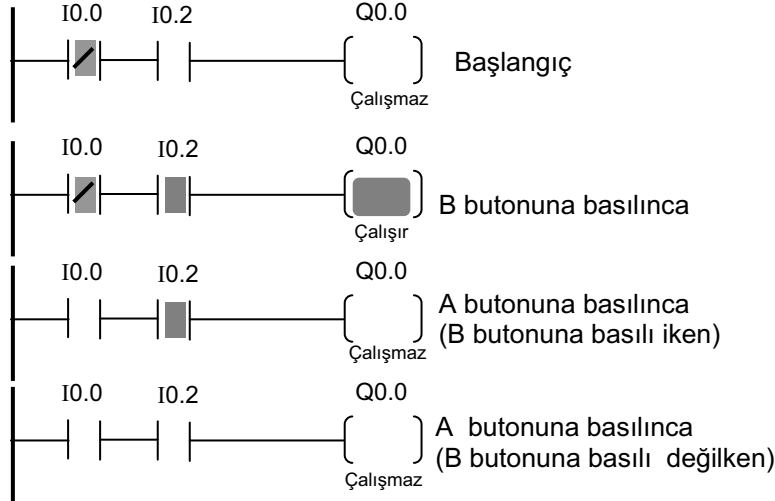
A ve B butonları açık kontak olarak kullanılacak olursa :



a) Klasik kumanda devresi



b) Ladder devresi



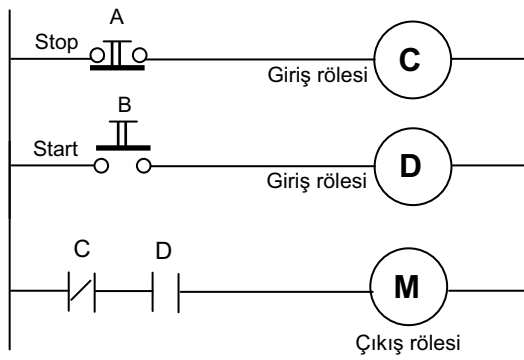
c) STL (Komut listesi) Network 1

LDN I0.0
A I0.2 A → AND
= Q0.0

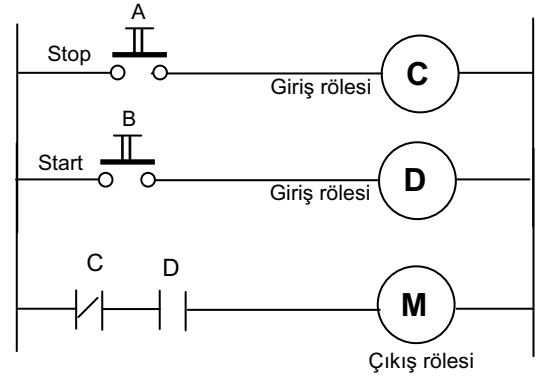
Şekil 4.5 :

A (stop) ve B (start) butonları açık kontaklı kullanıldığında, ilk anda I0.0 giriş rölesinin kapalı kontağı aktif (1) durumdadır. Yani enerji geçirir durumdadır. B (start) butonuna basıldığında I0.2 giriş rölesi enerjilenir. I0.2 giriş rölesi, kontağını kapatır ve Q0.0 çıkış rölesi enerjilenir (aktif olur). B butonuna basılı iken A (stop) butonuna basılırsa I0.0 giriş rölesi enerjilenerek ladder devresindeki I0.0 kapalı kontağını açar. Bu durumda Q0.0 çıkış rölesinin enerjisi kesilir (pasif olur). B butonuna basılı değilken A butonuna basılırsa, ladder devresinde her iki kontak da açık olacağından çıkış rölesi çalışmaz.

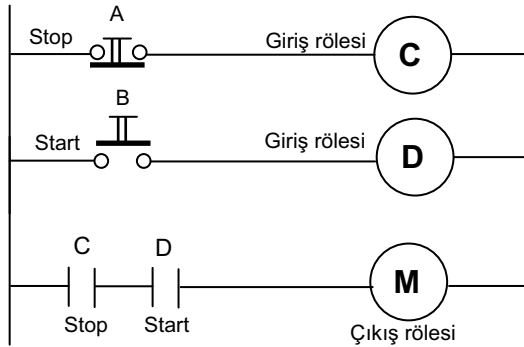
Bu durum aşağıdaki şekilde daha anlaşılır şekilde ifade edilebilir.



Devreye enerji verildiği an C kontaktörü enerjilenir ve kontağını açar. B butonuna basılırsa D kontaktörü enerjilenir ve kontağını kapatır. Ancak M kontaktörü enerjilenemez. Çünkü C kontağı açılmıştır. M kontaktörünün çalışması için A butonuna da basılmalıdır. **Bu durum, klasik kumanda mantığına uygun değildir.** Aşağıdaki devre ile bu sakınca ortadan kaldırılabilir. Ancak :

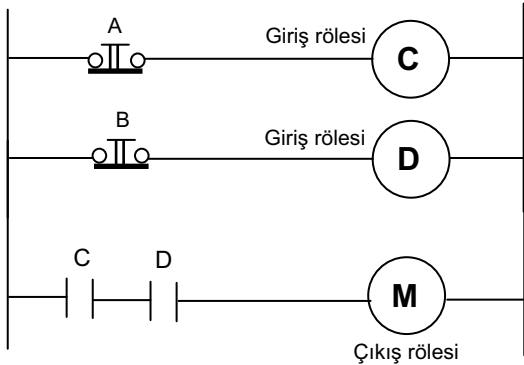


Enerji verildiği an M devresindeki C kontağı aktif, D kontağı pasif durumda olduğundan M kontaktörü çalışmaz. B (start) butonuna basılırsa D kontaktörü enerjilenir M devresindeki kontağını kapatır ve M kontaktörü enerjilenir. B butonuna basılı iken A (stop) butonuna basılırsa C kontaktörü enerjilenir ve M devresindeki kontağını açar. Böylece M kontaktörünün enerjisi kesilir. Butonlar her ikisi de serbest bırakılırsa şekildeki durum oluşur. **Bu durum, klasik kumanda mantığına uygundur.**



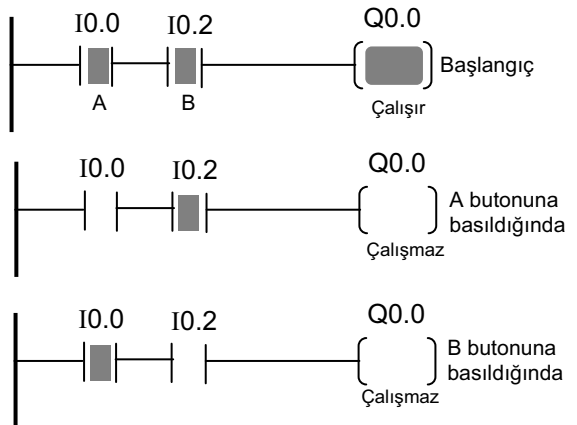
Biz kumanda mantığında durdurma işlemleri için sürekli kapalı kontak kullanıyorduk. Burada ise M devresinde durdurma işlemi için açık kontak kullanılmıştır. Bu kural burada ihlal edilmektedir. Dolayısıyla proje tasarımcısı için karışıklıklara yol açabilir.

İki stop (kapalı) butonunun PLC'ye etkisi :



a) Klasik kumanda devresi

Network 1



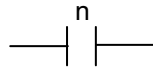
b) Ladder devresi

PLC' de durdurma işlemleri için STOP butonunu kapalı kontaklı olarak kullanmak yerine START butonu gibi açık kontaklı kullanmak daha uygundur. Çünkü kapalı kontaklı STOP butonu bağlı bulunduğu giriş rölesine sürekli enerji verir. Ancak açık kontaklı buton, giriş rölesine sadece butona basılınca enerji verir.

Durdurma butonunun açık kontaklı kullanılmasının sakıncası ise durdurma (STOP) butonuna gelen iletkenler koptuğunda ya da butona basıldığı halde kontaklarının kapanmaması halinde durdurma işleminin gerçekleştirilememesidir. Bu durum bazı tesislerde hayati ve maddi tehlike yaratabilir. Proje tasarlanırken bu sakıncanın tedbiri alınmalıdır.

4.5 - BİR KOMUTUN YAPISININ İNCELENMESİ :

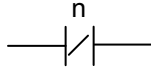
Normalde Açık Kontak



Operantlar : n (bit) I, Q, M, SM, S, T, C, V

Açıklama : n adresinin değeri "1" olduğu zaman normalde açık kontak kapanır. Normalde açık bir kontak kapandığında enerji akışı olur. Seri bağlandığı zaman, normalde açık kontak, takip eden Ladder elemanı ile lojik olarak AND' lenmiş olur. Paralel bağlandığında ise OR mantığı kullanılmış olur.

Normalde Kapalı Kontak



Operantlar : n (bit) I, Q, M, SM, S, T, C, V

Açıklama : n adresinin değeri "0" olduğunda normalde kapalı kontak kapanır "1" olduğunda ise kontak açıktır. Güç akışı kontak kapandığında olur. Seri bağlandığı zaman, normalde kapalı kontak, takip eden ladder elemanı ile lojik olarak AND' lenmiş olur. Paralel bağlandığında ise OR mantığı kullanılmış olur.

Yükle (STL)

Format : LD n

Operantlar : n I, Q, M, SM, S, T, C, V (bit)

Açıklama : Bu komut, n adresindeki bitin değerini lojik yığının en üstüne kopyalar. Yığındaki diğer bit değerleri bir aşağıya kayar.

LD → Hat açma ve hatta açık kontak bağlama işlemi



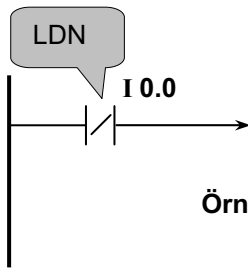
Yükle Değil (STL)

Format : LDN n

Operantlar : n (bit) I, Q, M, SM, S, T, C, V

Açıklama : Bu komut, n adresindeki bitin değerinin **değilini** lojik yığının en üstüne kopyalar. Yığındaki diğer bit değerleri bir aşağıya kayar.

LDN → Hat açma ve hatta kapalı kontak bağlama işlemi



Örnek : LDN I 0.0 → (0.0 dan 7.7 ye kadar)
(oktal sayı sistemi)

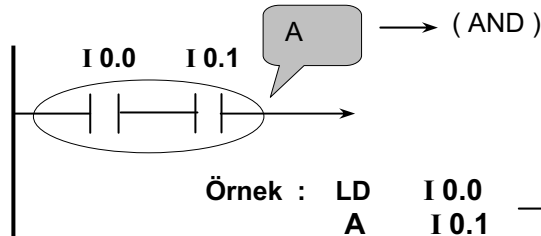
And (STL)

Format : A n

Operantlar : n (bit) I, Q, M, SM, S, T, C, V

Açıklama : Bu komut, süregelen lojik sonucuya n adresindeki bit değerini lojik olarak AND ler. Sonuç, yeni süregelen lojik sonuç olarak lojik yığının en üstüne yazılır.

A → Açık kontakları seri bağlama işlemi



Örnek : LD I 0.0
A I 0.1 → (0.0 dan 7.7 ye kadar)
(oktal sayı sistemi)

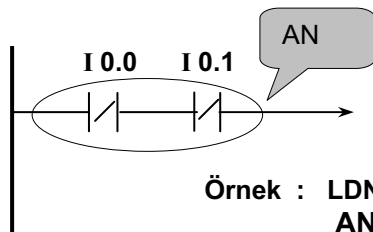
And Değil (STL)

Format : AN n

Operantlar : n (bit) I, Q, M, SM, S, T, C, V

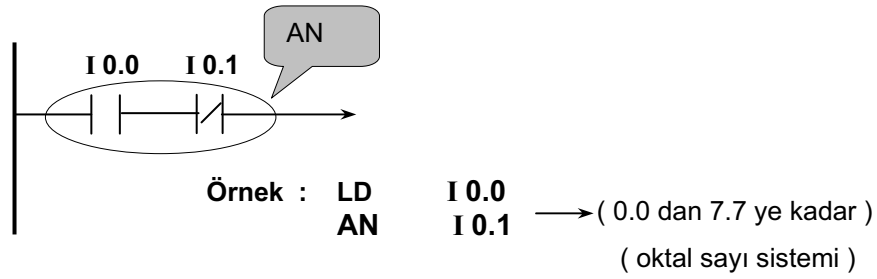
Açıklama : Bu komut süregelen lojik sonucuya n adresindeki bit değerini lojik olarak AND NOT lar. Sonuç, yeni süregelen lojik sonuç olarak lojik yığının en üstüne yazılır.

AN → Kapalı kontakları seri bağlama işlemi

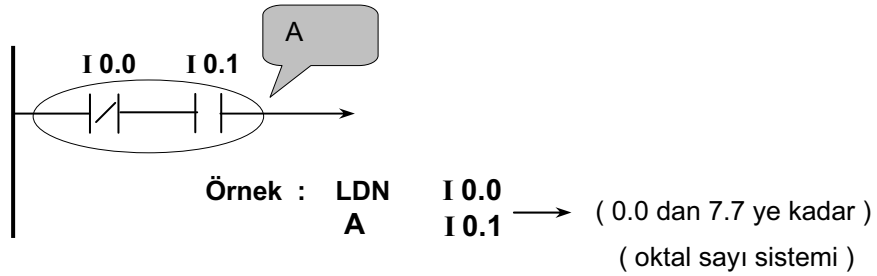


Örnek : LDN I 0.0
AN I 0.1 → (0.0 dan 7.7 ye kadar)
(oktal sayı sistemi)

AN → Açık kontak ile kapalı kontağı seri bağlama işlemi



A → Kapalı kontak ile açık kontağı seri bağlama işlemi



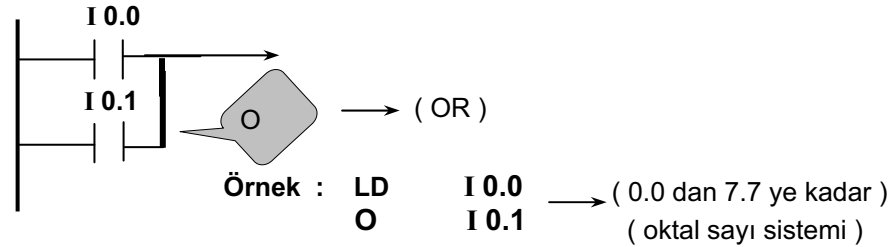
Or (STL)

Format : O n

Operantlar : n (bit) I, Q, M, SM, S, T, C, V

Açıklama : Bu komut lojik yığının en üstteki bitini n adresindeki bitle lojik olarak OR lar. Sonuç lojik yığının en tepe değeri olur.

O → Açık kontakları paralel bağlama işlemi



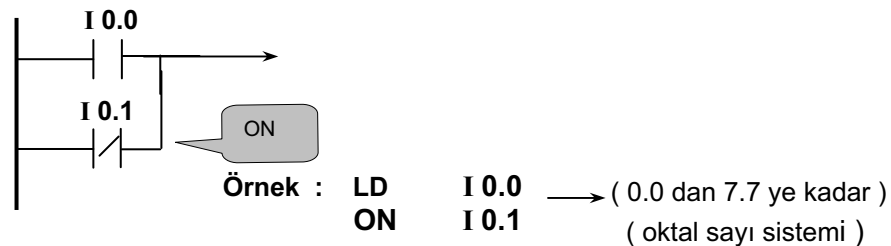
Or Değil (STL)

Format : ON n

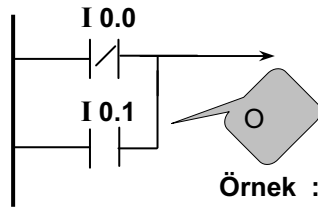
Operantlar : n (bit) I, Q, M, SM, S, T, C, V

Açıklama : Bu komut n adresindeki bitin değeriyle lojik yığının en üst değeri arasında lojik OR NOT işlemi yapar. Sonuç lojik yığının tepe değerine yazılır.

ON → Açık kontak ile kapalı kontağı paralel bağlama işlemi



O → **Kapalı kontak ile açık kontağı paralel bağlama işlemi**



Örnek : LDN I 0.0
O I 0.1 → (0.0 dan 7.7 ye kadar)
(oktal sayı sistemi)

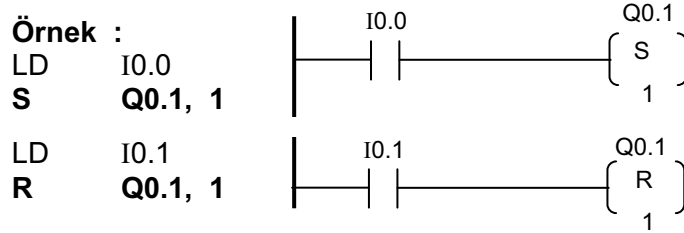
Set

Sembol : ——— S BIT
[S]
N

Operantlar : S_BIT (bit) : I, Q, M, SM, S, T, C, V

N (bayt) : IB, QB, MB, SMB, SB,
VB, AC, Sabit sayılar, * VD, *AC

Açıklama : S_BIT adresinden başlayan N adet biti set eder (1 yapar). Set edilebilecek nokta sayısı 1- 255 arasındadır (CPU 210 hariç). N, kaç bitin SET edileceğini ifade eder (burada sadece Q0.1 biti).



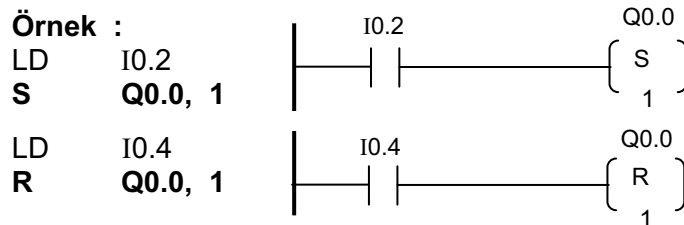
Reset

Sembol : ——— R BIT
[R]
N

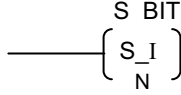
Operantlar : R_BIT (bit) : I, Q, M, SM, S, T, C, V

N (bayt) : IB, QB, MB, SMB, SB,
VB, AC, Sabit sayılar, * VD *AC

Açıklama : R_BIT adresinden başlayarak N adet biti reset eder. (0 yapar) Reset edilebilecek nokta sayısı 1 - 255 arasındadır (CPU 210 hariç). Eğer R_BIT bir zaman rölesini (T) veya sayıcıyı (C) gösteriyorsa, hem T/C çıkış biti, hem de anlık sayma değeri reset edilir. N, kaç bitin RESET edileceğini ifade eder (burada sadece Q0.0 biti).



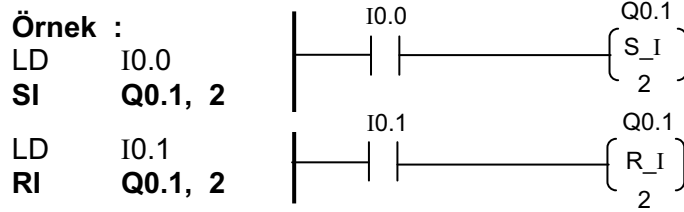
Anında Set :

Sembol : 

Operantlar : S_BIT (bit) : Q

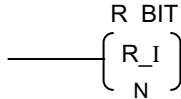
N (bayt) : IB, QB, MB, SMB, SB,
VB, AC, Sabit sayılar, * VD, *AC

Açıklama : S_BIT adresinden başlayan N adet biti set eder (1 yapar). Program taramasının bitmesini beklemeyiz. Set edilebilecek nokta sayısı 1- 64 arasındadır (CPU 210 hariç). "I" anında yapılan işlemi gösterir. Yeni değer hem fiziksel çıkışa anında (komut çalıştıktan hemen sonra) yazılır. Sadece çıkış kütüğüne yazan normal SET komutundan farkı budur. N, kaç bitin SET edileceğini ifade eder (burada Q0.1 biti ve Q0.2 biti).



Bu komut, değişim hızı PLC nin döngü (tarama) hızından daha fazla olan sinyallerin okunması ve sonucun tarama bitmeden çıkış birimine aktarılması amacı ile kullanılır.

Anında Reset

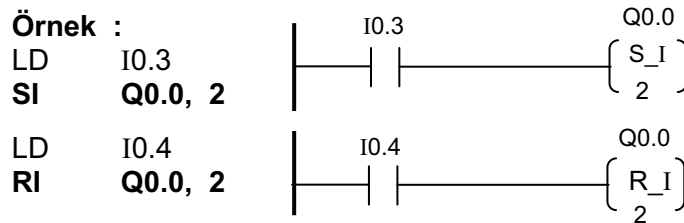
Sembol : 

Operantlar : R_BIT (bit) : Q

N (bayt) : IB, QB, MB, SMB, SB,
VB, AC, Sabit sayılar, * VD *AC

Açıklama : R_BIT adresinden başlayarak N adet biti reset eder. (0 yapar) Program taramasının bitmesini beklemeyiz. reset edilebilecek nokta sayısı 1- 64 arasındadır (CPU 210 hariç). "I" anında yapılan işlemi gösterir. Yeni değer hem fiziksel çıkışa anında (komut çalıştıktan hemen sonra) yazılır. Sadece çıkış kütüğüne yazan normal RESET komutundan farkı budur.

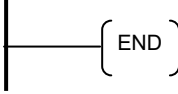
Eğer R_BIT bir zaman rölesini (T) veya sayıcıyı (C) gösteriyorsa, hem T/C çıkış biti, hem de anlık sayma değeri reset edilir. N, kaç bitin RESET edileceğini ifade eder (burada Q0.0 biti ve Q0.1 biti).



Bu komut, değişim hızı PLC nin döngü (tarama) hızından daha fazla olan sinyallerin okunması ve sonucun tarama bitmeden çıkış birimine aktarılması amacı ile kullanılır. Bu tür komutların interrupt alt programlarında kullanılması gerekir.

Girişler için LDI, AND işlemleri için ANI, OR işlemleri için ORI komutları da kullanılabilir.

Son

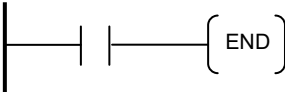
Sembol : 

Operantlar : yok

Açıklama : Koşulsuz son, kullanıcı programını bitirmek için kullanılır. Her zaman için ana programın son devresini oluşturur. Bir programda mutlaka koşulsuz son bulunmalıdır.

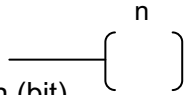
Uyarı! : Bu komut, MicroWIN 32 programında kullanılmaz.

Koşullu son

Sembol : 

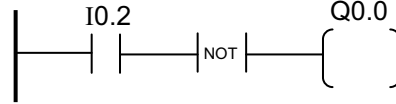
Açıklama : Koşullu son, önündeki kontak aktif ise programı sonlandırır. Bir programda Koşullu Son bulunmayabilir veya birden çok bulunabilir. Ancak bir programda bir defa koşulsuz son kullanılabilir.

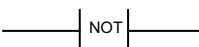
Çıkış

Sembol : 
Operantlar : n (bit) I, Q, M, SM, S, T, C, V

Açıklama : Bobinin bağlı olduğu hatta enerji geldiği zaman çıkış rölesi çeker ve n adresi 1 olur. DEĞİL (güç akışını ters çevir) kontağı bir çıkış bobininin önüne yerleştirilerek ters çevrilmiş bir çıkış oluşturulabilir.

Güç Akışını Ters Çevir



Sembol : 


Operantlar : Yok

IO.2 kontağı açıksa Q0.0 çıkış rölesi çalışır. IO.2 kontağı kapalı ise Q0.0 çıkış rölesi çalışmaz. Yani ters işlem yapar.

Açıklama : NOT (Güç akışını ters çevirme) kontağına enerji akışı gelirse durur, gelmezse güç akışı yaratır. Lojik olarak DEĞİL işlemini yapar. **Özel Kısıtlamalar :** NOT kontağı güç hattına bağlanamaz.

NOT kontağı, başka bir kola bağlantı sağlamak için bir kola paralel bağlanamaz

Pozitif Yükselen Kenar

Sembol : 

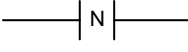
Operantlar : Yok

Açıklama : 0' dan 1 konumuna her geçişte, sadece bir tarama için güç akışına izin verir. Her CPU için pozitif ve negatif yükselen kenar kullanımının bir sınırı vardır. Maksimum sayılar aşağıda gösterilmiştir.

CPU 212	CPU 214-215-216-221-222 -224-226	CPU 210
128	256	32

Özel Kısıtlamalar : Pozitif yükselen kenar kontağı güç hattına bağlanamaz. Pozitif yükselen kenar kontağı başka bir kola bağlantı sağlamak için bir kola paralel bağlanamaz.

Negatif Yükselen Kenar

Sembol : 

Operantlar : Yok

Açıklama : 1' den 0 konumuna her geçişte, sadece bir tarama için güç akışına izin verir. Her CPU için pozitif ve negatif yükselen kenar kullanımının bir sınırı vardır. Maksimum sayılar aşağıda gösterilmiştir.

CPU 212
128

CPU 214-215-216-221-222-224-226
256

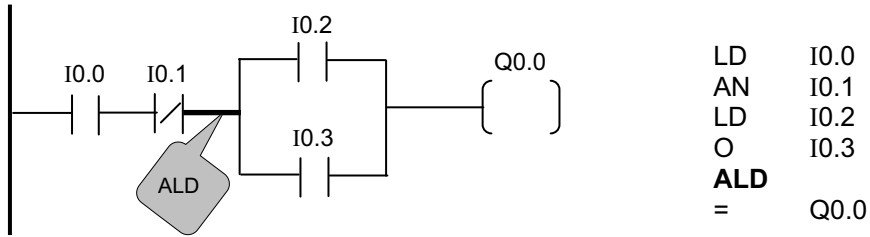
CPU 210
32

Özel Kısıtlamalar : Negatif yükselen kenar kontağı güç hattına bağlanamaz. Negatif yükselen kenar kontağı başka bir kola bağlantı sağlamak için bir kola paralel bağlanamaz.

And Yükle

Format : ALD

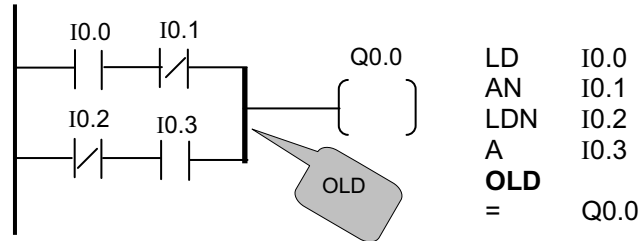
Bu komut, lojik yığının en üst ve bir altındaki bitlerinin lojik olarak AND' lenmesini sağlar. Sonuç yığının en üstüne yazılır ve yığının derinliği bir azalır.



Or Yükle

Format : OLD

Bu komut, lojik yığının birinci ve ikinci bitleri arasında OR işlemi yapar. Sonuç lojik yığının tepesine yazılır. İşlem sonucunda lojik yığının derinliği bir azalır.



Lojik Kesme

Format : LPP

Bu komut lojik yığının en tepesindeki bit değerini atar. İkinci bit değeri yığının tepesine yükselir ve diğer değerler bir yukarı kayar. En son çıkış kolu oluştur anlamındadır (**Logic Pop** kelimelerinin kısaltılmışı şeklidir).

Lojik İtme

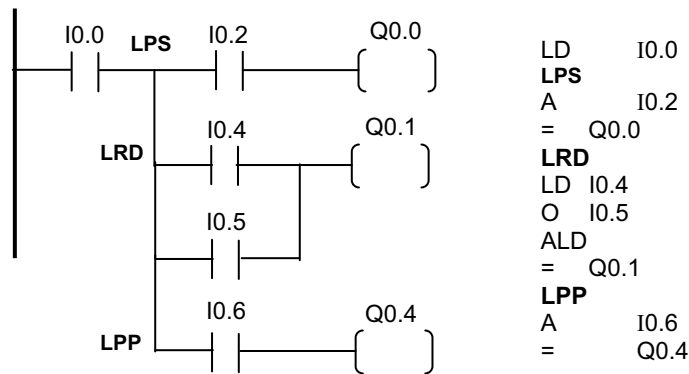
Format : LPS

Bu komut, lojik yığının tepesindeki değeri kopyalayarak üste yazar ve yığındaki diğer bitleri bir aşağı kaydırır. Yığının en altındaki bit itilerek atılır ve yok olur. Ara kol oluştur anlamındadır (**Logic Push** kelimelerinin kısaltılmış şeklidir).

Lojik Okuma

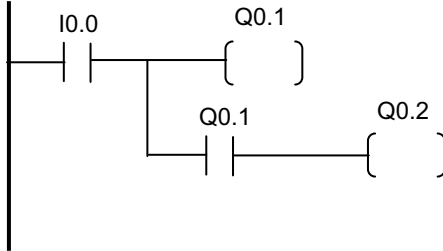
Format : LRD

Bu komut, lojik yığının ikinci bitini birincinin yerine kopyalar. Yığın kesilmez ya da itilmez (kayma olmaz), ancak ilk değer kopyalama nedeniyle yok olur. Ara kolu çıkışa oluştur anlamındadır (**Logic Read** kelimelerinin kısaltılmış şeklidir).



Örnek :

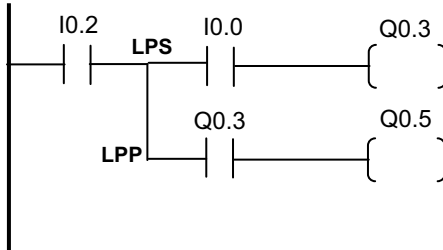
Network 1



Komut listesi (STL):

```
Network 1
LD I0.0
= Q0.1
A Q0.1
= Q0.2
```

Network 1



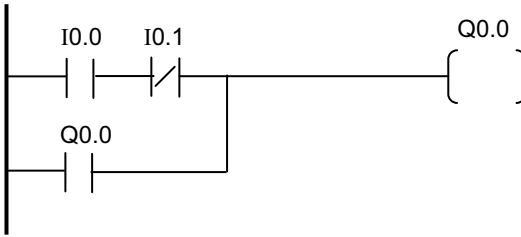
Komut listesi (STL):

```
Network 1
LD I0.2
LPS
A I0.0
= Q0.3
LPP
A Q0.3
= Q0.5
```

4.6 - Karışık devre yazılımları :

Örnek 1:

Network 1



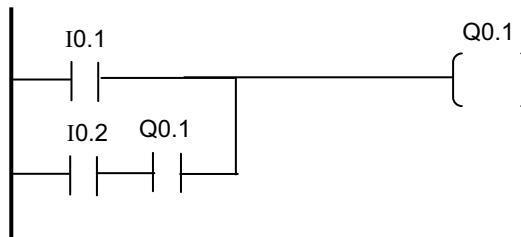
Ladder devresi

STL (Komut Listesi)

Network 1
LD I0.0
AN I0.1
O Q0.0
= Q0.0

Örnek 2 :

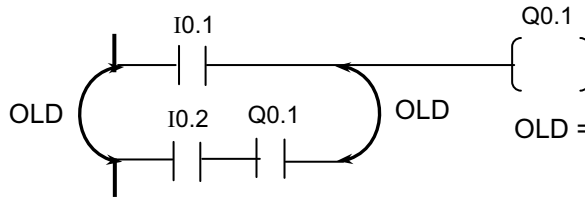
Network 1



Ladder devresi

STL (Komut Listesi)

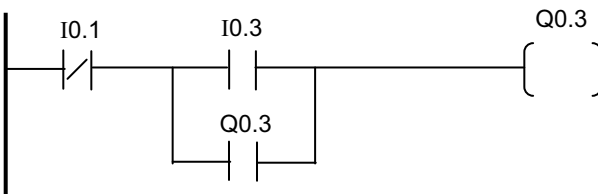
Network 1
LD I0.1
LD I0.2
A Q0.1
OLD
= Q0.1



OLD = Grupları paralel bağlama komutudur.

Örnek 3 :

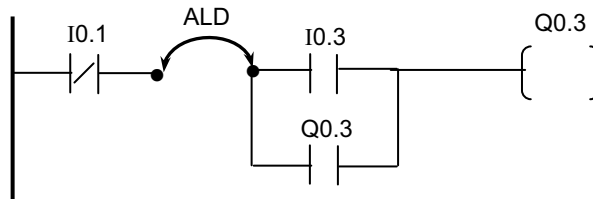
Network 1



Ladder devresi

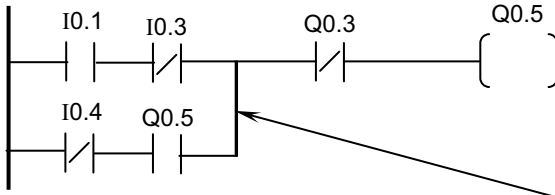
STL (Komut Listesi)

Network 1
LDN I0.1
LD I0.3
O Q0.3
ALD
= Q0.3



ALD = Grupları seri bağlama komutudur.

**Örnek 4 :
Network 1**

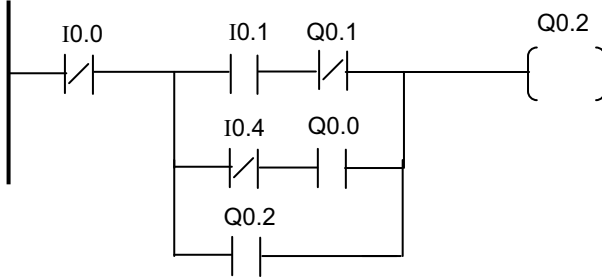


Ladder devresi

STL (Komut Listesi)

```
Network 1
LD    I0.1
AN    I0.3
LDN   I0.4
A      Q0.5
OLD
AN    Q0.3
=      Q0.5
```

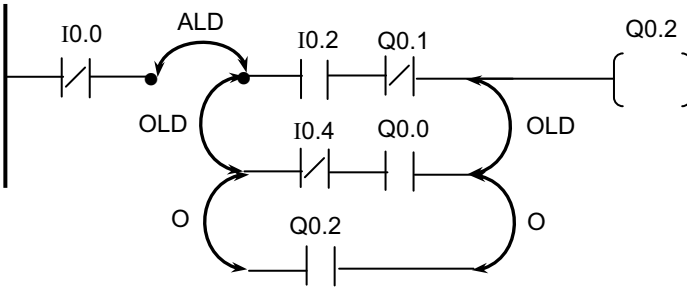
**Örnek 5 :
Network 1**



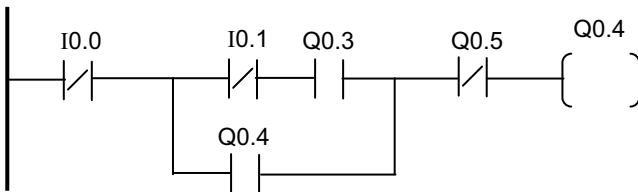
Ladder devresi

STL (Komut Listesi)

```
Network 1
LDN   I0.0
LD     I0.1
AN     Q0.1
LDN   I0.4
A      Q0.0
OLD
O      Q0.2
ALD
=      Q0.2
```



Örnek 6 :

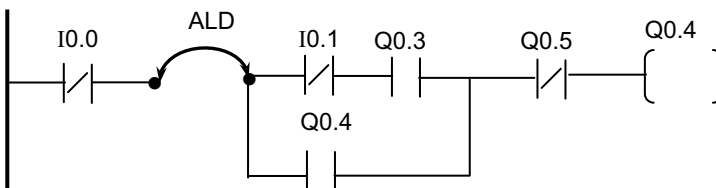


STL (Komut Listesi)

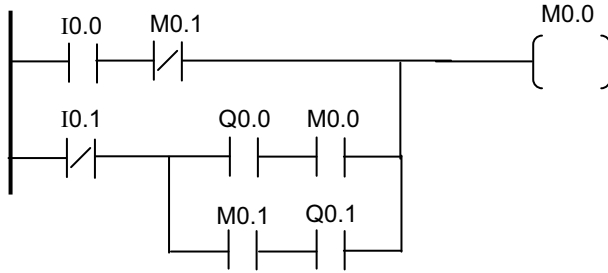
```
Network 1
LDN   I0.0
LDN   I0.1
A      Q0.3
O      Q0.4
ALD
AN    Q0.5
=      Q0.4
```

Network 1

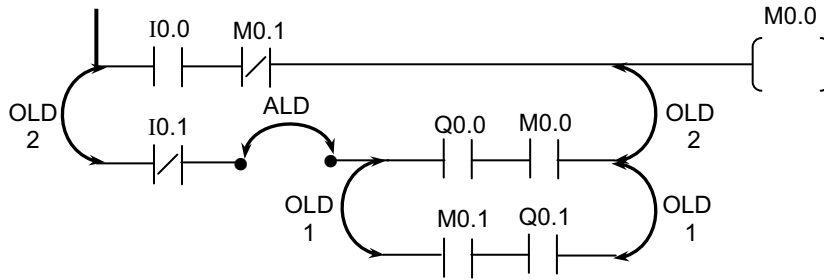
Ladder devresi



**Örnek 7 :
Network 1**



Ladder devresi



STL (Komut Listesi)

Network 1

```
LD      I0.0
AN      M0.1
LDN     I0.1
LD      Q0.0
A       M0.0
LD      M0.1
A       Q0.1
```

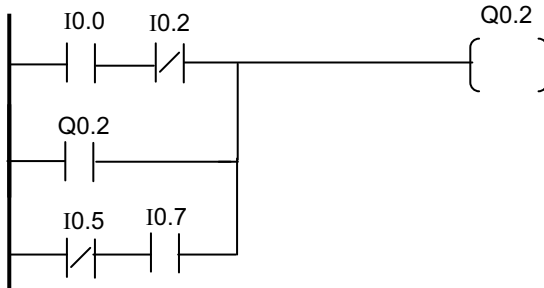
(1)OLD

ALD

(2)OLD

= M0.0

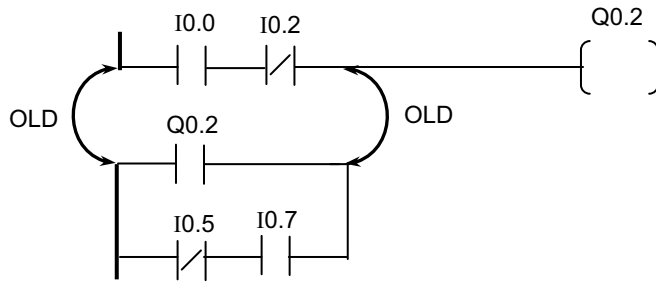
**Örnek 8 :
Network 1**



STL (Komut Listesi)

Network 1

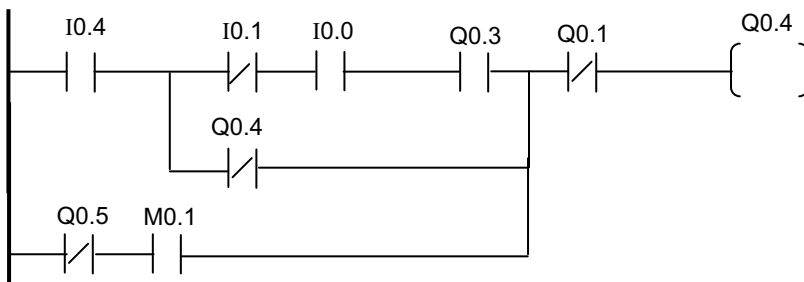
```
LD      I0.0
AN      I0.2
LDN     I0.5
A       I0.7
O       Q0.2
LDN     I0.5
A       I0.7
OLD     Q0.2
=       Q0.2
```

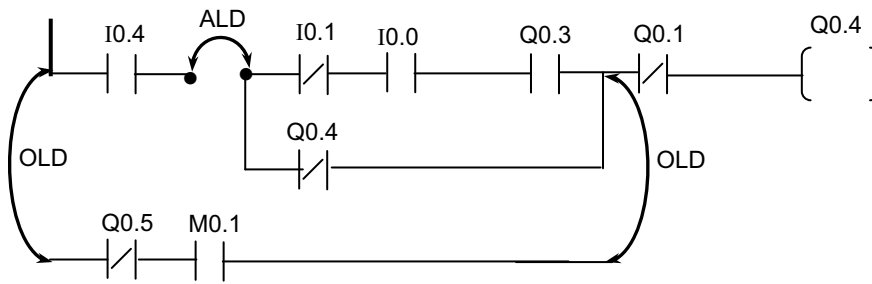


Örnek 9 :

Ladder devresi

Network 1



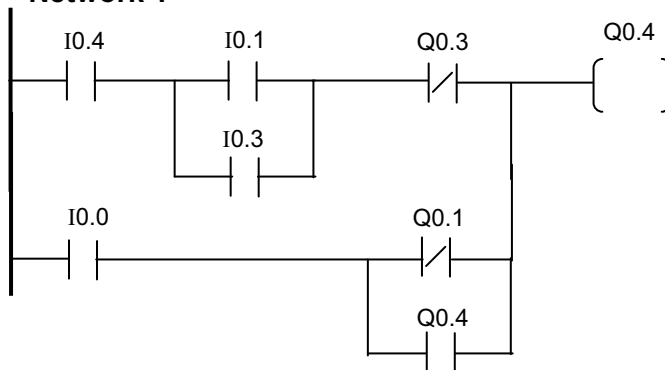


STL (Komut Listesi)

Network 1

```
LD    I0.4
LDN   I0.1
A      I0.0
A      Q0.3
ON     Q0.4
ALD
LDN    Q0.5
A      M0.1
OLD
AN     Q0.1
=      Q0.4
```

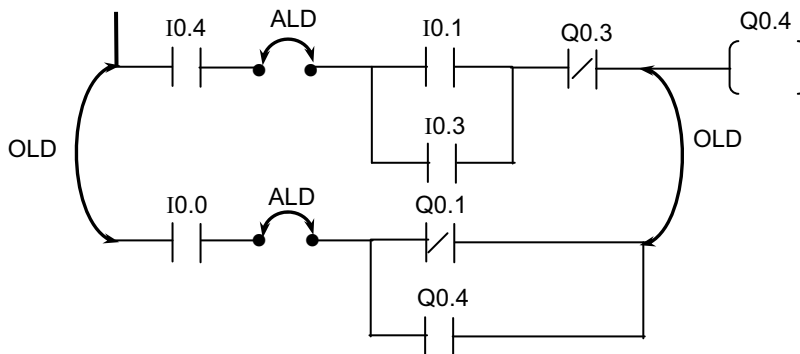
Örnek 10 : Network 1



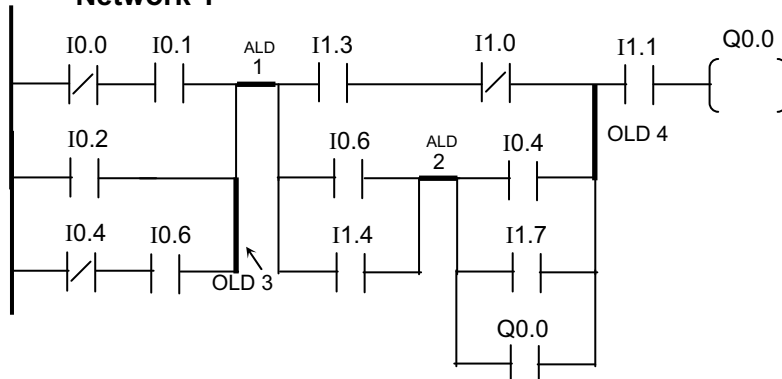
STL (Komut Listesi)

Network 1

```
LD    I0.4
LD    I0.1
O      I0.3
ALD
AN     Q0.3
LD    I0.0
LDN    Q0.1
O      Q0.4
ALD
OLD
=      Q0.4
```



Örnek 11 : Network 1

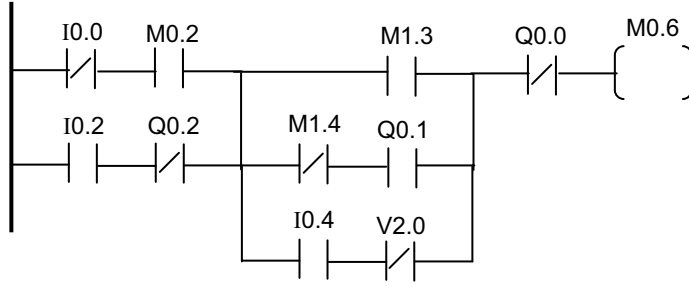


STL (Komut Listesi)

Network 1

```
LDN   I0.0
A      I0.1
O      I0.2
LDN   I0.4
A      I0.6
(3) OLD
LD     I1.3
AN     I1.0
LD     I0.6
O      I1.4
LD     I0.4
O      I1.7
O      Q0.0
```

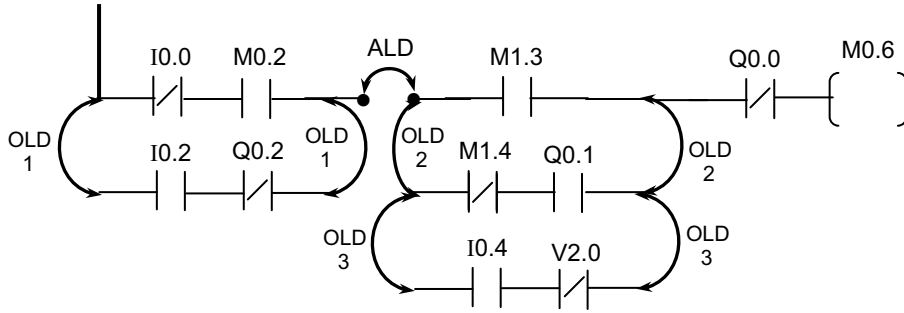
**Örnek 12 :
Network 1**



**STL (Komut
Listesi)**

Network 1

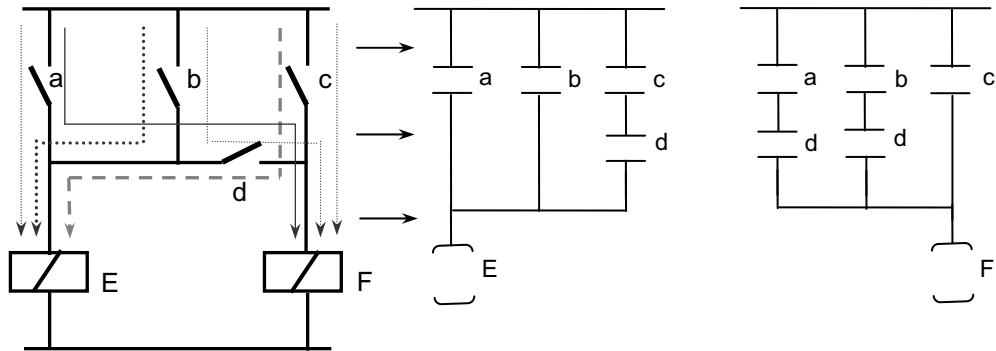
```
LDN I0.0
A M0.2
LD I0.2
AN Q0.2
(1) OLD
LD M1.3
LDN M1.4
A Q0.1
(2) OLD
LD I0.4
AN V2.0
(3) OLD
ALD
AN Q0.0
= M0.6
```



4.7 - Köprü devresi :

Kontaktörlü uygulamadan PLC uygulamasına geçiliyorsa büyük olasılıkla ladder gösterim şekline çevrilemeyecek bazı sviç (sensör) kombinasyonlarıyla karşılaşılır. Bunlardan biri de köprü devresidir. Hem basit hem de karmaşık köprü devreleri için çözüm açıklaması aşağıda gösterilmiştir.

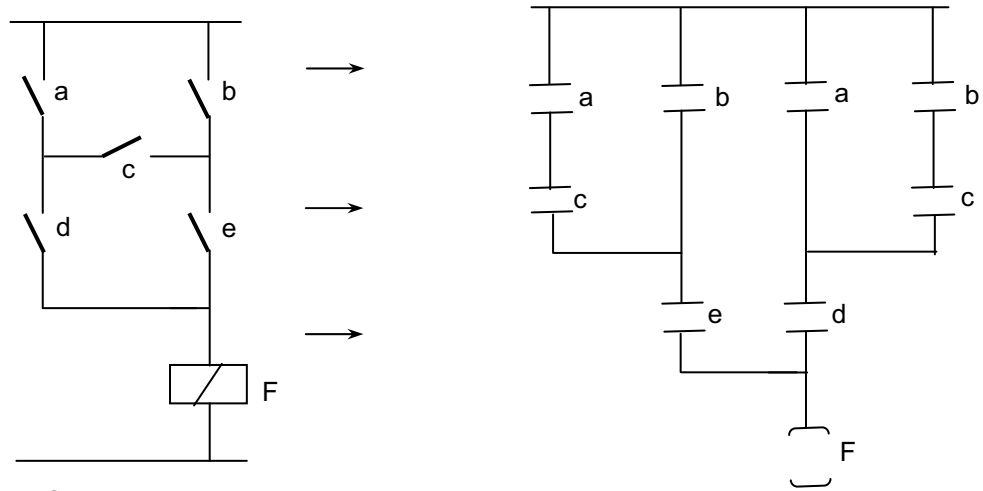
1) Basit köprü devresi



Şekil 4.6 : Basit köprü devresi ve ladder devresi

Görülen köprü devresi iki devre halinde "ladder"a çevrilmiştir. İki farklı olası akım yolu basitçe ayrılmıştır. Karşılaştırma kolaylığı açısından ladder gösterimi dikey olarak verilmiştir.

2) Karmaşık köprü devresi

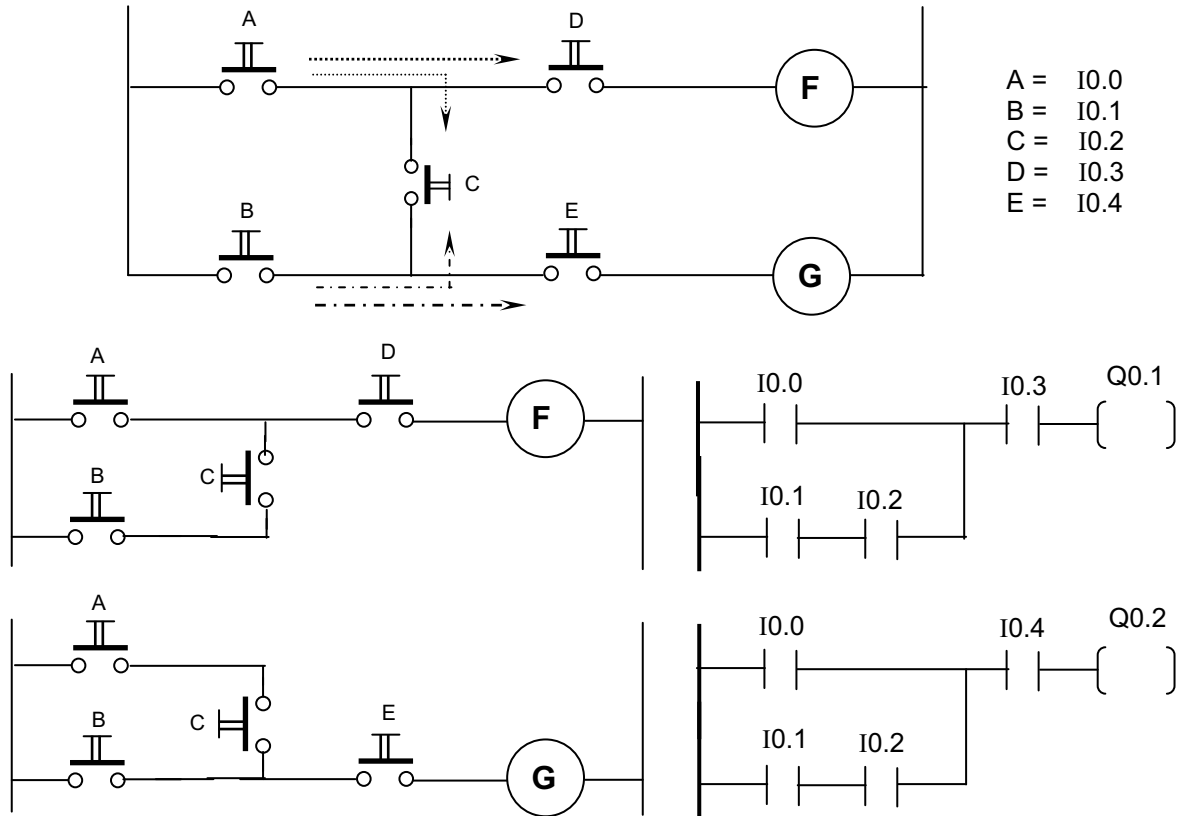


Şekil 4.7 : karmaşık köprü devresi ve ladder devresi

İki farklı olası akım yolu tekrar dönüştürülmüş ve birleştirilmiştir. Bir yanda a ve c, b' ye paraleldir, diğer yanda b ve c, a' ya paraleldir. Karşılaştırma kolaylığı açısından ladder gösterimi dikey olarak verilmiştir.

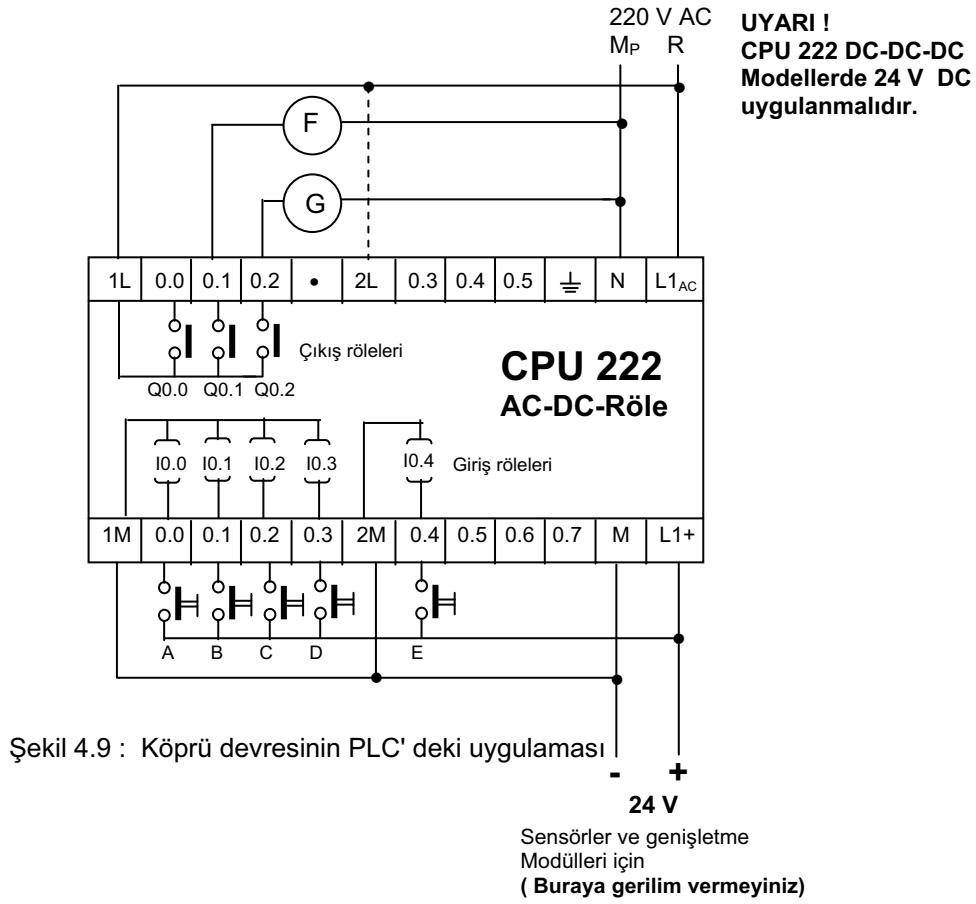
Yeni proje tasarımlarında mümkün olduğu kadar köprü devresi kullanmaktan kaçınılmalıdır. Başlangıçtan itibaren “ladder” mantığı ile düşünülmelidir.

Örnek :



Şekil 4.8 : karmaşık köprü devresi ve ladder devresi

PLC bağlantısı :

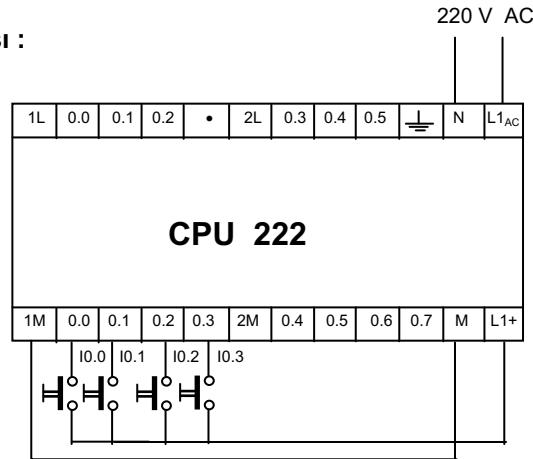


LPS, LDR ve LPP uygulaması ile ilgili örnek problem :

Bir tezgahta 3 adet selenoid valf ve 3 adet start butonu bulunmaktadır.

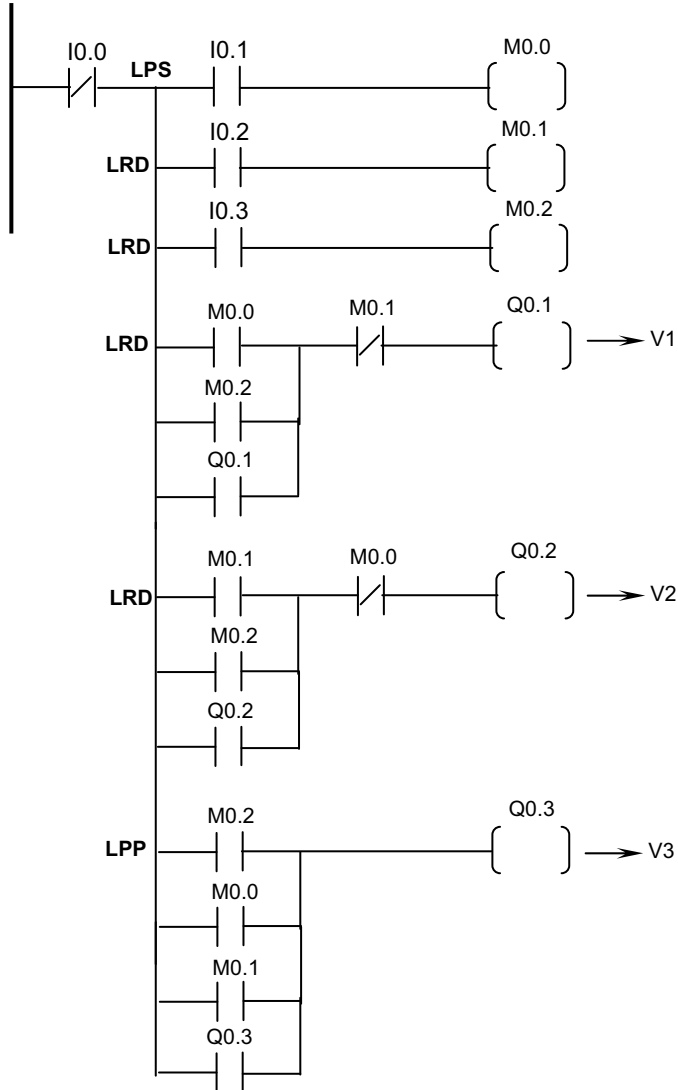
1. starta basıldığında 1 ve 3 nolu valfler sürekli olarak çalışacaktır.
 2. starta basıldığında 2 ve 3 nolu valfler sürekli devreye girmektedir.
 3. starta basıldığında valflerin üçü de sürekli çalışacaktır.
- Çalışma sırasında hangi start butonuna basılıyorsa o start butonuna ait valfler devreye girecektir. Ayrıca sistemi istenildiği anda durdurabilecek stop butonu kullanılacaktır.

PLC bağlantısı :



Ladder devresi :

Network 1



Komut listesi :

Network 1
 LDN I0.0
 LPS
 A I0.1
 = M0.0
 LRD
 A I0.2
 = M0.1
 LRD
 A I0.3
 = M0.2
 LRD
 LD M0.0
 O M0.2
 O Q0.1
 ALD
 AN M0.1
 = Q0.1
 LRD
 LD M0.1
 O M0.2
 O Q0.2
 ALD
 AN M0.0
 = Q0.2
 LPP
 LD M0.2
 O M0.0
 O M0.1
 O Q0.3
 ALD
 = Q0.3

Şekil 4.11: Ladder devresi

