

# EGE ÜNİVERSİTESİ

## EGE MYO

---

MEKATRONİK PROGRAMI

# PROGRAMLANABİLİR DENETLEYİCİLER

---

PROGRAM BUYRUKLARI

# PROGRAM BUYRUKLARI

---

END Buyrukları

NOP Buyrukları

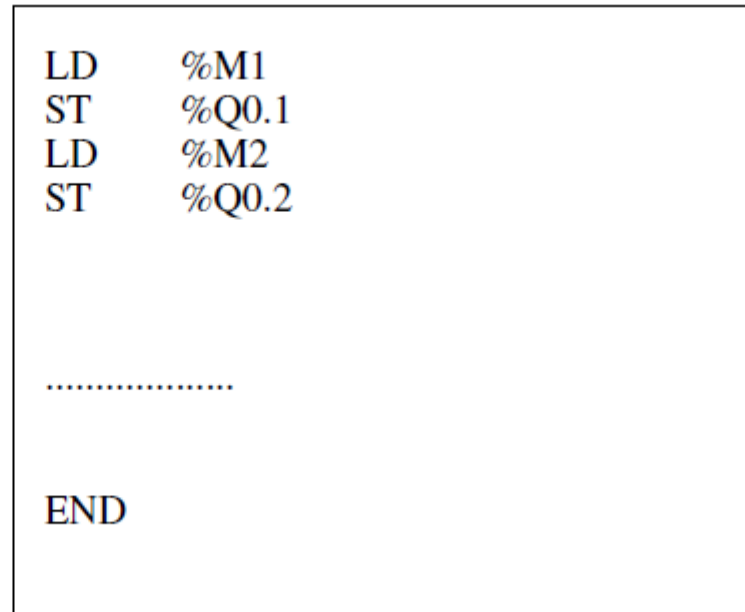
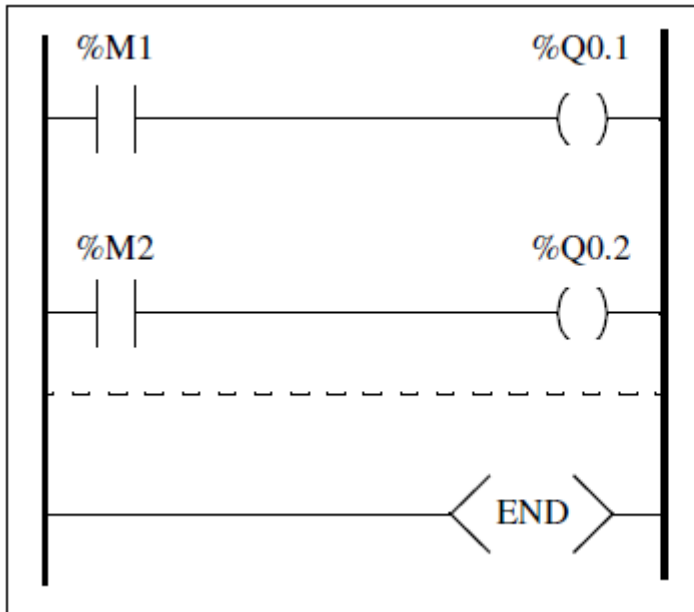
Jump (Atlama) Buyrukları

Altyordam (Subroutine) Buyrukları

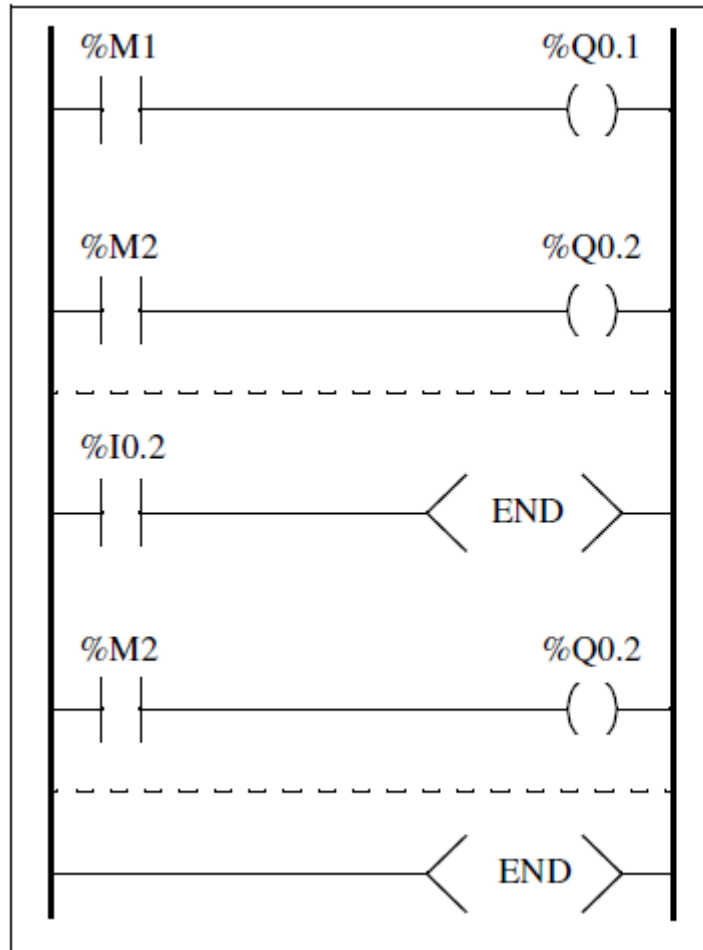
# END Buyrukları

- **Açıklama** End buyrukları, bir program taraması icrasının sonunu tanımlar.
- **END, ENDC ve ENDCN**
- Üç farklı end buyruğu mevcuttur:
  - **END**: programın koşulsuz bitimi
  - **ENDC**: Bir önceki test buyruğunun Boole sonucu 1 ise, programın bitimidir.
  - **ENDCN**: Bir önceki test komutunun Boole sonucu 0 ise, programın bitimidir.
- Varsayılan olarak, program bitimi etkinleştirildiği zaman çıkışlar güncellenir ve sonraki tarama başlatılır.
- Tarama periyodik ise, periyot bitimine ulaşıldığı zaman çıkışlar güncellenir ve sonraki tarama başlatılır.

- **Örnekler** Koşulsuz bir END buyruğu örneği.



- Koşullu bir END buyruğu örneği.



```
LD   %M1
ST   %Q0.1
LD   %M2
ST   %Q0.2
```

.....

```
LD   %I0.2
ENDC
LD   %M2
ST   %Q0.2
```

→ %I0.2 = 1 ise,  
program taramasının  
bitimi

%I0.2 = 0 ise, yeni  
END komutuna kadar  
program taramasını  
devam ettirir

.....

```
END
```

# NOP Buyruđu

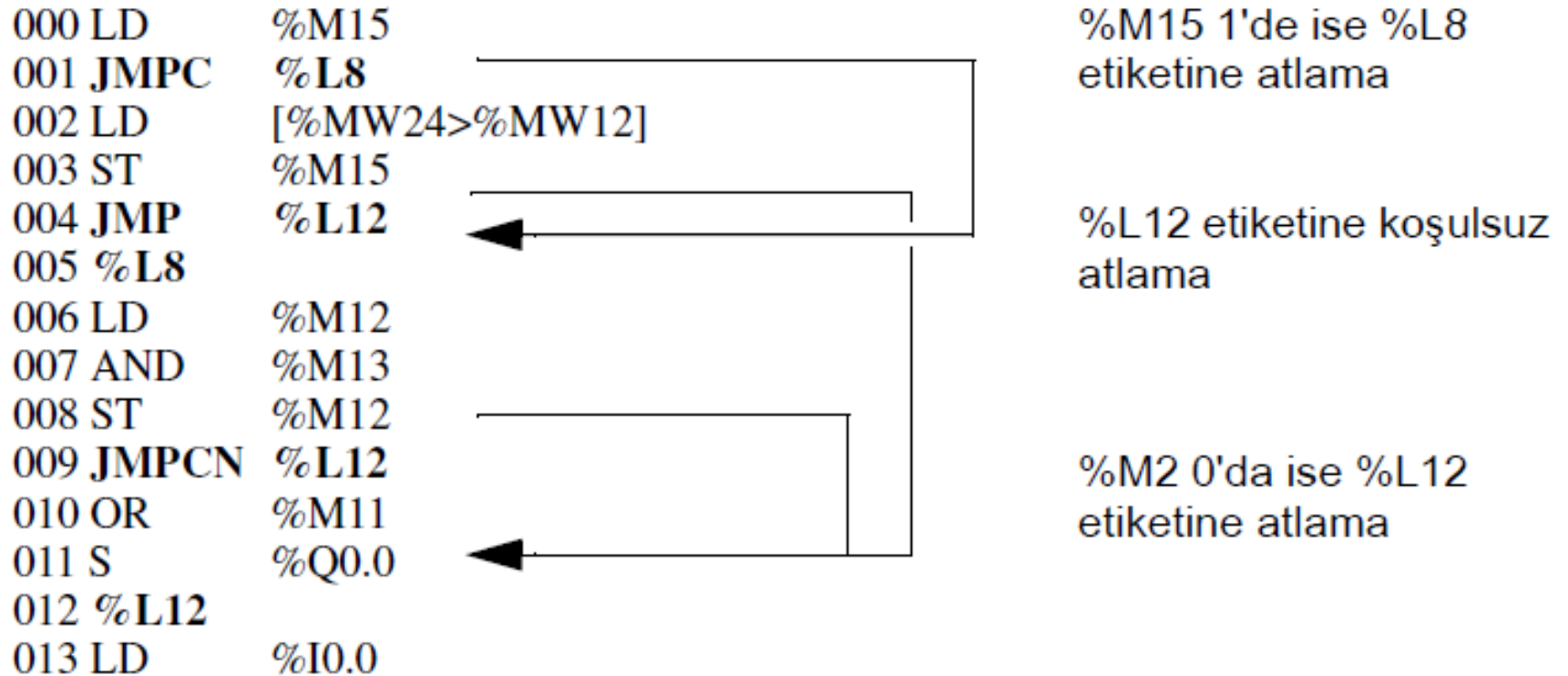
- NOP buyruđu, herhangi bir işlem gerçekleştirmez. Bir programda satırları (lines) "ayırarak" için kullanınız, böylece daha sonra, satır numaralarını değiştirmeksizin komutları yerleştirebilirsiniz.

# Jump (Atlama) Buyrukları

- **Açıklama** Jump buyrukları bir programın icrasının hemen kesilmesine ve %Li (i = 0-15) etiketini taşıyan program satırından sonraki satırdan devam edilmesine neden olur.
- **JMP, JMPC ve JMPCN**
- Üç farklı Jump buyruğu mevcuttur:
  - JMP: koşulsuz program atlama
  - JMPC: Bir önceki mantığın Boole sonucu 1 ise program atlama
  - JMPCN: Bir önceki mantığın Boole sonucu 0 ise program atlama



## • Jump buyruklarına örnekler.



## Ana noktalar

- Jump buyruklarının, parantezler arasında bulunmasına izin verilmez ve bunlar AND(, OR(, ve bir kapa parantez komutu ")" arasına yerleştirilmemelidir.
- Etiket, yalnızca bir LD, LDN, LDR, LDF veya BLK buyruğundan önce yerleştirilmelidir.
- %Li etiketinin etiket numarası, program içinde sadece bir kere tanımlanmalıdır.
- Program atlaması, akış-aşağı veya akış-yukarı olan bir programlama satırına gerçekleştirilir. Atlama akış-yukarı olduğunda program tarama süresine dikkat edilmelidir. Aşılan tarama zamanı, watchdog zamanlayıcının süresinin sona ermesine neden olabilir.

# Altyordam (Subroutine) Buyrukları

- **Açıklama** Altyordam buyrukları, bir programın bir altyordamı gerçekleştirmesine ve sonrasında ana programa geri dönmesine neden olur.
- **SRn, SRn: ve RET**
- Altyordamlar üç adımdan oluşur:
  - **SRn** buyruğu, bir önceki Boole buyruğunun sonucu 1 ise, SRn etiketi ile referanslanmış altyordamı çağırır.
  - Altyordam, **SRn** etiketiyle referanslanır:  
TWDLCAA10DRF, TWDLCAA16DRF için n=0-15; diğer tüm denetleyiciler için n=0-63 arasında değer alabilir.
  - Bir altyordamın bitimine yerleştirilen **RET** buyruğu, program akışını ana programa geri döndürür.

- Altyordam buyruklarına örnekler.

```
000 LD      %M15
001 AND     %M5
002 ST      %Q0.0
003 LD      [%MW24>%MW12]
004 SR8
005 LD      %I0.4
006 AND     M13
007 -
008 -
009 -
010 END

011 SR8:
012 LD      1
013 IN      %TM0
014 LD      [%TM0.Q]
015 ST      %M15
016 RET
```



## Ana noktalar

- Bir altyordam, başka bir altyordamı çağırmamalıdır.
- Altyordam buyruklarının, parantezler arasında bulunmasına izin verilmez ve bunlar AND(, OR(, ve bir kapa parantez buyruğu)" arasına yerleştirilmemelidir.
- Etiket, sadece bir LD veya bir Boole denkleminin (veya rung) başlangıcını işaret eden bir BLK buyruğundan önce yerleştirilebilir.
- Bir altyordam çağrısını takip eden bir atama buyruğu bulunmamalıdır çünkü altyordam, Boole akümülatörün içeriğini değiştirebilir. Bu yüzden geri dönüşte, akümülatör çağrı öncesindeki değerinden farklı bir değere sahip olabilir. Aşağıdaki örneği inceleyiniz.

- Bir altyordam programlama örneği.

