

57466

YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Y.Ü. YÜKSEKÖĞRETİM KURULU
DOKÜMAN

PLC VE ENDÜSTRİYEL
TESİSLERDE UYGULAMALARI

Elek. Müh. Hilmi DİNÇ
F.B.E. Elektrik Mühendisliği Anabilim Dalında
hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Prof. Dr. Halit PASTACI

İSTANBUL - 1996

İÇİNDEKİLER

İÇİNDEKİLER	1
TÜRKÇE ÖZET	5
İNGİLİZCE ÖZET	6
GİRİŞ	7
I) PLC'ye Giriş	8
I-1) Merkezi İşlem Ünitesi	9
I-1-1) CPU işlemcisi	9
I-1-2) Ana Bellek	10
I-2) Giriş/Çıkış Modülleri	11
I-2-1) Giriş Modülü	11
I-2-2)Çıkış	12
I-2-3) I/Q İşlemleri	12
I-3) Güç Kaynağı	13
I-4) Programlayıcı	13
II) Lojik Sistemler	13
III) PLC'ler Arası İletişim	15
IV) Simatic - S5 Yazılım Program	16
IV-1) Online Programlama	16
IV-2) Offline	16
IV-3)Simatic S5 Programlamasına giriş	16
IV-3-1) Program Yapısı	16
IV-3-1-1) Organizasyon Blokları	18
IV-3-1-2) Program Blokları	18
IV-3-1-3) Fonksiyon Blokları	18
IV-3-1-4) Veri Blokları	20
IV-3-1-5) Sıra Blokları	20
IV-3-2) Veri Yapısı Blokları	20
IV-3-3) Merdiven Program Mantığı	21
IV-3-4) İfade Listesi “ “ (STL)	21

IV-3-5) Program Dökümanteri	21
V)Dijital Arabirim Elemanlarına Giriş	22
V-1) Giriş	22
V-2) Dijital Giriş Elemanları	22
V-2-1) Mekanik Anahtarlar	22
V-2-2) Transistör Anahtarlar	23
V-2-3) Yaklaşım Anahtarlar	24
V-2-4) Opto Elektronik Dedektörler	25
V-2-5)Kodlayıcılar Dedektörler	26
V-2-6) Sıcaklık Anahtarları	27
V-2-7) Basınç Anahtarları	28
V-3) Dijital Çıkış Elemanları	28
V-3-1) Selenoidler	29
V-3-2) Kontaktörler	29
V-3-3) Katı hal Rölesi	30
VI) Analog Arabirim Elemanları	30
VI-1) Giriş	30
VI-2) Dijital /Analog Dönüştürücüler (DAD)	31
VI-3) Analog/Dijital “ (ADD)	32
VI-4) Çoklayıcılar	33
VI-5) Aradevreler	33
VI-6) Analog Dönüştürücüler	36
VI-6-1) Potansiyometreler	36
VI-6-2)Doğrusal Değişebilir Diferensiyel Transformatör	36
VI-6-3) Termokulp	37
VI-6-4)Genleşme ölçer	38
VII) Simatic S5 Programı Ana Menüsü	39
VIII) Komutlar	47
VIII-1) Basit İşlemler	48
VIII-1-1) Boolean lojik İşlemleri	48
VIII-1-2) AND İşlemi	49

VIII-1-3) OR İşlemi	49
VIII-1-4) OR işleminden önce AND işlemi olması hali	49
VIII-1-5) AND işleminden önce OR işlemi olması hali	50
VIII-1-6) SET ve RESET işlemleri	50
VIII-1-7) Yükleme ve Transfer	52
VIII-1-8) TIMER işlemleri	55
VIII-1-8-1) Pulse Timer	59
VIII-1-8-2) Genişletilmiş Pulse Timer	60
VIII-1-8-3) On-Delay Timer	60
VIII-1-8-4) Depolanmış on-delay ve reset timer	61
VIII-1-8-5)Off-delay timer	62
VIII-1-9) Sayıcı İşlemleri	62
VIII-1-10) Karşılaştırma İşlemleri	66
VIII-1-11) Aritmetik İşlemleri	67
VIII-1-12) Blok Çağırma İşlemleri	68
VIII-1-13) Diğer İşlemleri	72
VIII-2) İlave İşlemler	73
VIII-2-1) CPU-103 için yükleme işlemleri	73
VIII-2-2) CPU-103 enable işlemi	74
VIII-2-3) CPU-103 bit-test işlemi	75
VIII-2-4) Dijital lojik işlemler	76
VIII-2-5) Kaydırma İşlemleri	79
VIII-2-6) Değişim İşlemleri	80
VIII-2-7) Azaltma/arttırma	81
VIII-2-8) Disable/Enable Kesintisi	83
VIII-2-9) “Do” İşlemi	83
VIII-2-10) aTLAMA İşlemleri	86
VIII-2-11) Yer değiştirme İşlemleri	87
VIII-2-12) Set/Reset İşlemleri	88
VIII-2-13) Yükleme ve Transfer İşlemleri	88
VIII-2-14)Timer ve Sayıcı İşlemleri	90

VIII-2-15) “Do” İşlemi	91
VIII-3) Sistem İşlemleri	92
VIII-3-1) Set İşlemi	92
VIII-3-2)Yükleme ve transfer işlemleri	93
VIII-3-3) Aritmetik İşlemleri	95
VIII-3-4) Diğer İşlemler	95
VIII-3-5) Şart Kodu üretme	96
VIII-4) Basit programlar	98
VIII-4-1) Geçici-kontak röle/kenar değerlendirilmesi.	98
VIII-4-2) Binary scaler / binary bölücü	98
VIII-4-3) Saat/Saat Pulse jeneratörü	99
IX) Uygulamalar	101
IX-1) Bir motora start/stop butonlarıyla direkt yok verme	101
IX-2) Bir motora yıldız/üçgen yolverme	103
IX-3) İki bayanın karıştırılması için hazırlanmış düzeneğin PLC ile kontrolü	104
IX-4) Yaya geçitlerinde bulunan trafik ışıklarının PLC ile kontrolü	106
IX-5) Bir konveyör üzerindeki 100 adet şişeyi sayan düzeneğin PLC ile kontrolü	108
IX-6)Dokuz katlı bir binadaki hava damperlerinin PLC ile kontrolü	109
IX-7) Yedeklemeli olarak çalışan bir trafo merkezinin PLC ile kontrolü	113
SONUÇLAR	118
YARARLANILAN KAYNAKLAR	119
ÖZGEÇMİŞ	120

ÖZET

Programlanabilir lojik kontrol cihazı (PLC), belleğinde saklanmış lojik programı kullanarak çeşitli kontrol fonksiyonlarını gerçekleştiren elektronik bir cihazdır. PLC, bilgisayar kullanmayı biraz bilen kişiler tarafından dahi programlanabilir. Lojik programın değişebilir olması sebebiyle kontrol edilen sistemde istenilen çalışma şekilleri kolayca elde edilebilir.

Birinci bölümde, PLC tanımı yapılmakta ve CPU, giriş/çıkış modülleri, güç kaynağı ve PLC programlayıcısı hakkında temel bilgiler verilmiştir. İkinci bölümde, PLC'nin çalışmasını sağlayan lojik sistemler tanıtılmıştır. Üçüncü bölümde PLC'ler arası iletişimin nasıl sağlandığı ve arabirim elemanları hakkında bilgiler verilmiştir. Dördüncü bölümde, Simatic-S5 serisinde program yazmak için gerekli bilgiler verilmiştir. Beşinci bölümde, PLC'ler kullanılarak bir sistemin kontrol edilmesi için gerekli olan dijital giriş/çıkış cihazlarının prensip şemaları ve çalışma prensipleri üzerinde durulmuştur. Altıncı bölümde, Analog / Dijital dönüştürücüler (ADD) ve Dijital - Analog dönüştürücüler hakkında bilgiler verilmiştir. Yedinci bölümde, Simatic S5 serisinin Dos altında çalışan program menüsü tanıtılmıştır. Sekizinci bölümde PLC konutları ve programda nasıl kullanılacakları anlatılmıştır. Dokuzuncu bölümde ise modern tesislerde ve endüstride kullanılan PLC uygulamaları ile ilgili örnekler verilmiştir.

Son bölümdeki uygulamalar merdiven ve ifade listesi programları ile yazılmış ve programın çalışması hakkında bilgiler verilmiştir. En son uygulamada ise bir tesis besleyen enerji merkezinin PLC kullanılarak kontrol eden program yazılmıştır.

SUMMARY

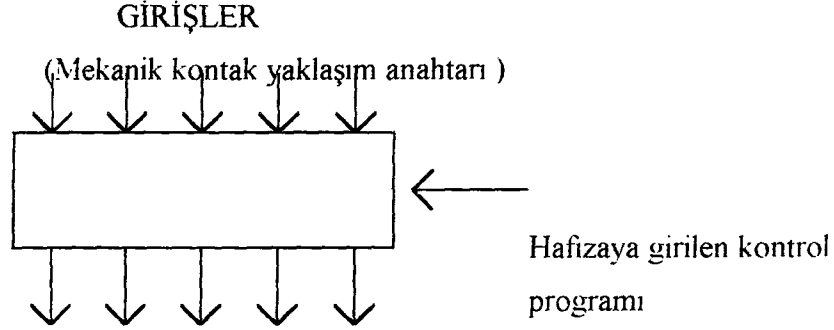
A programmable logic controller is an electrical device that carries out different control functions using the logic program stored in the memory. PLC can be programmed by a person who has a little knowledge about computers. The operating systems which are necessary in the controlling system can be easily available, because of the changeability of the logic program.

In the first chapter, it is mentioned what PLC is and it is given some basic knowledge about CPU, input/output modules, power supply and programming device. In the second chapter, it is mentioned the logic systems which permit the operating of PLC. In the third section, some information about the communication interfaces and how the transmission can be between PLC's are given. In the fourth chapter, the necessary information to write a program using Simatic S5 series program are given. In the fifth chapter the digital input/output devices, which are necessary to control an automation system using PLC, and their operating principles are mentioned. In the sixth chapter some information about Analog/Digital converter (ADD) and Digital/Analog converter (DAD) are given. In the seventh chapter, Simatic S5 series's program menu which uses Dos, is noticed. In the eighth chapter, the operations of PLC and how they are used in the program are mentioned. In the ninth chapter, some examples about the applications which are used in the modern installations and the industry are given.

The programmes of the applications in the last chapter are written using the ladder and STL program and some information about the operating of the program and some information about the operating of the program are given. In the last program, which controls a power centre supplying an installation, is written by using PLC program.

GİRİŞ

Lojik , sıralama, zamanlama, sayma ve aritmetik işlemler gibi makinaları kontrol eden özel fonksiyonları gerçekleştirmek ve komutları bellekte saklamak için programlanabilir bir belleği kullanan dijital elektronik cihaza PLC denir.



PLC, belleğinde saklanmış lojik programlarını kullanarak makina ve fiziksel cihazları kontrol eder. Bu programlar giriş cihazları vasıtasıyla PLC'ye iletilen işarete (örneğin; anahtar pozisyonu, sıcaklık veya mekanik hareket vb.) cevap olarak uygulanacak talimatları içerir. PLC içine saklanmış bu program mantığıyla röle veya motor kontrol cihazları gibi çıkış cihazlarında kullanılacak fiziksel işlemin durumunu kontrol eder.

PLC Karakteristikleri

- Bilgisayar programcısı olmayan kişiler içinde kullanımı oldukça kolaydır.
- Oldukça düşük fiyat ve yekpare gövde.
- Kesikli ve analog çıkış imkanı .
- Değişken programlama kabiliyeti.
- Devre işleminin görsel izlenebilmesi.
- Yüksek hızlı katı hal işletimi ve yüksek güvenilirlik.
- Devre işletiminin yazılı çıktısı.

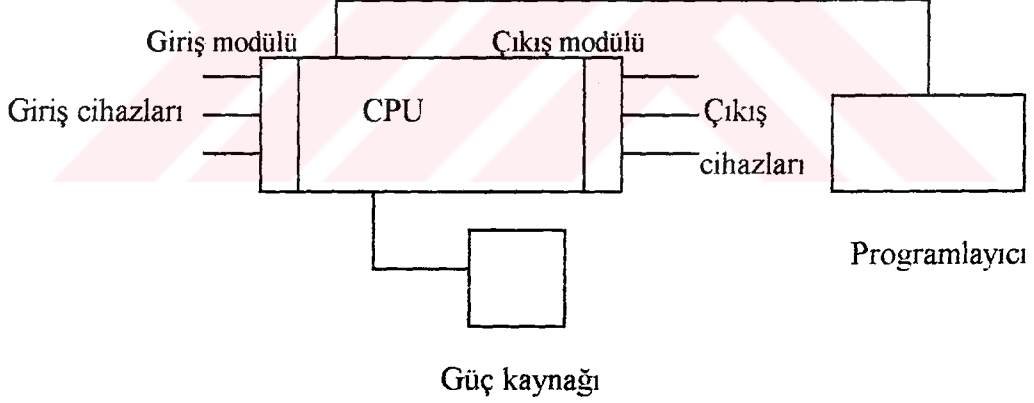
- İstenmeyen program değişikliğinin önlenmesi için güvenlik kilidi.
- Devrenin kolayca değişimi.
- Simulasyon modunda pilot çalıştırma
- Sabit program kullanımının ekonomik olmayışı.
- Doğal şartlara karşı hassasiyet.

1-) PLC'ye Bakış

Programlanabilir kontrolörlerin çoğu aşağıdaki 5 ana bileşenden oluşur:

- 1-) Merkezi işlem ünitesi (CPU)
- 2-) Giriş Modülü
- 3-) Çıkış Modülü
- 4 -) Güç Kaynağı
- 5 -) Programlayıcı

Bu bileşenler aşağıdaki Şekil -1' de gösterilmiştir.



Şekil 1-2-) Tipik PLC sistemi

I-1) Merkezi işlem ünitesi (CPU):

CPU programlanabilir kontrolörün çalışmasını kontrol eder. CPU kullanıcı programını ve sayısal veri değerlerini saklayan belleği ve programı uygulayan , aritmetik ve lojik işlemlerini yapan Mikroişlemci devresini içerir.

CPU işlemci bölümü aritmetik lojik ünitesi (ALU) ve kontrol ünitesinden oluşur. Ana bellek bölümü ise sistem programları , kullanıcı programları, ve sistem verisini saklayan kısımdır.



Şekil -2 CPU'nun bileşenleri

I-1-1) CPU işlemcisi: CPU'nun işlemci bölümü kontrol ünitesi ve Aritmetik lojik ünitesi (ALU) 'dan oluşmuştur.

a-) Kontrol ünitesi: Kontrol ünitesi program talimatlarını düzenleyen , herbir talimatı yorumlayan ve işlemi yapmak için ALU'ya gerekli bilgiyi gönderen temel bilgisayar fonksiyonlarını içeren ünedir.

b-) Aritmetik mantık ünitesi (ALU): Bütün lojik işlemleri ve matematik hesaplamaları yapar.

I-1-2)ANA BELLEK: CPU'nun ana belleği sistem programlarını, kullanıcı programlarını ve sistem verisini saklamak için kullanılır. CPU'da bilgileri saklamak için temel olarak iki tip bellek vardır.

a-)Silinebilir bellek: Silinebilen bellek yalnızca CPU için gerekli güç sağlanabiliyorken işlev görür. CPU kapatıldığı zaman bu bellek içerikleri kaybolur.Bu yüzden ; sık olarak silinebilen bellek pil gibi güç kaynakları ile beslenir.

b-) Silinemeyen bellek: Silinemeyen bellek sabit bir güç kaynağına gerek duymaksızın içerikleri saklar ve sürekli olan program ve verileri saklamak için kullanılırlar.Silinmeyen bellek içerikleri genellikle özel dizayn edilmiş cihazlar kullanılarak değiştirilebilir.

BELLEK TİPLERİ :

Aşağıda beş temel bellek tipi listelenmiştir.

a-) RAM: RAM silinebilen bir bellektir ve geçici uygulama programlarını veya uygulama programlarının bir bölümünü saklamak için kullanılır. RAM'daki verilere program talimatlarını uygulamak için gerekli zamanın miktarını azaltarak direkt olarak ulaşılabilir. RAM belleği genellikle pil ile desteklenir.

b-) ROM: ROM silinemeyen ve kararlı tip bellektir. ROM'la program kaydedildiği zaman, çoğu halde değiştirilemez. ROM tipik olarak değiştirilemeyen sistem programlarını saklamak için kullanılır.

c-) PROM: Silinemeyen tipli bir bellektir.Bazen tam olarak düzeltilebilen bir uygulama programı veya bir RAM programı için bir destek olarak kullanılabilir.

d-) EPROM: EPROM silinemeyen bir bellek tipi olup belirli bir zaman için ultraviyole ışığa maruz bırakılarak silinebilir. EPROM'u yeniden programlamak için belleğim tüm içerikleri ilk olarak silinmelidir.

e-) EEPROM: EEPROM silinemeyen bir bellek tipidir ve EPROM'a benzer olarak dizayn edilmiş bir cihazı kullanarak çabucak silinebilir.

BELLEĞİN KULLANIMI :

CPU'nun ana belleği aşağıdaki bilgileri saklamak için kullanılır.

-Sistem programları: Sistem programları sürekli olarak işlemcinin çalışmasını kontrol eden kontrol programlarıyla yüklenir.Sistem programı hardware’i kontrol eder, ve kullanıcı programlarındaki talimatları yerine getirir.

-Kullanıcı programları: Kullanıcı programları kullanıcı tarafından yazılır ve çalıştırılmak üzere CPU belleğine transfer edilir. CPU belleğinin en büyük bölümünü kullanıcı program kaplar.

-Sistem verisi: Sistem verisi “sabit “ değerleri içerir ve programlar tarafından kullanılır.Bu veri bellekte aşağıda belirtilmiş bölgelerde saklanır.

• / Depolama Sahası (Storage Area): Sistem veya kullanıcı programlarda tarif edilen timer veya sayıcı değerlerini depolamak için kullanılır.

• / Giriş işlemcisi görüntü tablosu (PIII) ,giriş modül durumunu gösterir.

• / Çıkış işlemcisi görüntü tablosu (PIQ) ,çıkış modül istenilen durumunu gösterir.

I-2) GİRİŞ ve ÇIKIŞ MODÜLLERİ :

Giriş ve çıkış (I/Q) modülleri,PLC ve işlemciler veya PLC’yi kontrol için programlanmış cihazlar arasındaki bağlantıyı temin ederler.I/Q modülleri PLC ve I/Q cihazları tarafından tanımlanmış çeşitli şekillerde bilgileri transfer edip gerekli dönüşümleri yaparlar.

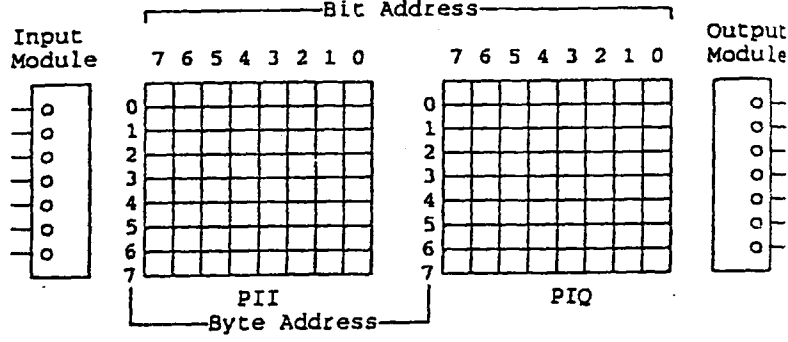
I-2-1)Giriş Modülü= Giriş modülü; giriş cihazları (pushbutton,keyboard vb.) ve programlanabilir kontrolör arasında fiziki bir bağlantı veya ara birim elemanı olarak iş görür.

Giriş cihazına bağlandığı giriş modül ucuna göre bir adres verilir.Bu uç üzerinden cihazın durumu PLC’ye iletilir.

Giriş cihazlarının durum değişikliklerini görmek için giriş modülü periyodik olarak taranır.Bu bilgi ise PII tablosunda saklanır. PII tablosu CPU’nun ana hafızasının sistem veri alanına yerleştirilir.

Eğer giriş modülündeki bir bit “on “ olursa PII tablosundaki uygun bölge (yer) “1” değerini alacaktır.Benzer olarak giriş modülündeki bir bit “OFF “olursa, PII tablosundaki uygun bölge “ 0 “ değerini alacaktır.Program işlerken ,CPU giriş cihazının durumu için PII tablosuna göre hareket eder.

Aşağıdaki şekil I/Q modülleri ve PII, PIQ tabloları arasındaki ilişkiyi göstermektedir.



Şekil -3) PII- PIQ Tabloları Arasındaki İlişki

I-2-2) Çıkış Modülü= Çıkış modülü çıkış cihazları (sınır anahtarları, motor starterleri, uyarı cihazları vb.) ve programlanabilir kontrolörler arasında bir fiziki veya ayırıcı birimdir.

Giriş cihazlarına benzer olarak, çıkış cihazları da bağlandıkları çıkış modülleri terminallerine göre adreslenirler. Cihazın durumu bu terminaller üzerinde, PLC'den alınan bilgiye göre değişir.

Belirli bir cihaz için çıkış durumu program tarafından belirlendiği zaman, bu durum ilk olarak CPU'nun ana belleğinin sistem verisi bölgesindeki PIQ tablosuna yazılır. Program tamamlandığı zaman, çıkış modülüne bağlanmış cihazların durumu PIQ tablosunda belirtilmiş değerlere göre değişir.

Eğer PIQ tablosunda bir bit "1" değerini içeriyorsa, çıkış modülündeki uygun adres "ON" olur. Benzer olarak, eğer PIQ tablosundaki bir bit "0" değerini içeriyorsa, çıkış modülündeki uygun adres "OFF" olur.

I-2-3) I/Q İşlemleri= Bir kullanıcı program çalışırken, PLC aşağıdaki sırayla I/Q modüllerin kullanır.

- PII ve PIQ tabloları resetlenir ve çıkış modüllerindeki uçlar boştur.
- İşlemci giriş bit'lerinin durumlarını okumak için giriş modülünün terminallerini tarar.
- İşlemci bir zamanda PII değerlerini tarayarak PIQ'ya gerekli değerleri yazarak kullanıcı programı çalıştırır.

-Tüm program tamamlandıktan sonra, PIQ'deki her bir bit'in durumu çıkış modülündeki uçlara transfer edilir. Bu işlemler gerekli çıkış cihazlarına "ON" değerini verir. Gerekli tüm çıkış cihazları servise alındıktan sonra periyot tekrar eder.

I-3) GÜC KAYNAĞI=

Güç kaynağı PLC işlemcisi için 5V DC güç temin eder. Güç kaynağı işlemci devreleri tarafından kullanılan AC hat gerilimini DC gerilime çevirir. Güç kaynağı aynı zamanda giriş-çıkış modülleri için devreleri için gerekli gücü temin eder.

I-4)PROGRAMLAYICI=

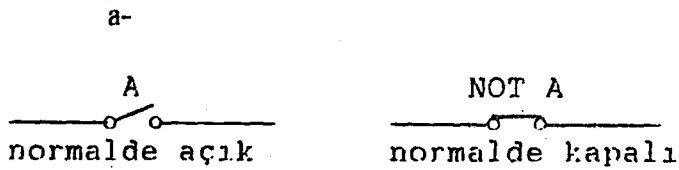
Programlama cihazı teknik olarak programlanabilir kontrolörün bir bölümü değildir. Bu cihaz kullanıcı programını hazırlamak ve işlemcisi belleğine bu programı transfer etmek için kullanılır.

Bazı PLC'ler bu cihazları kullanırken ,simetrik 55'e uygun bir IBM şahsi bilgisayarıyla çalıştırılabilir.

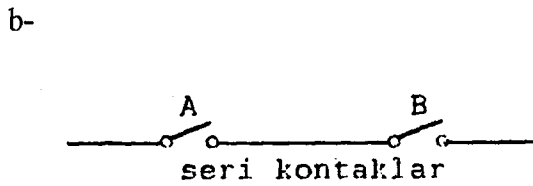
II-LOJİK SİSTEMLER

Lojik sistemler iki durumlu olarak çalışırlar. Bu iki durum basit bir anahtarın açık veya kapalı pozisyonu gibidir. Örneğin ,bir pnomatik valfin açık veya kapalı olması.

Lojik sistemlerde Boolean cebiri kullanılmaktadır. Boolean cebri terimleri ; basit bir anahtarın açık-kapalı veya sıfır-bir konumlarına karşılık gelir. Eğer anahtarı " A " ile isimlendirirsek, anahtar kapalı iken $A=1$ ve anahtar açıkken $A=0$ yazmak mümkündür. " A " harfi bir Boolean değişkeni olarak isimlendirilir.

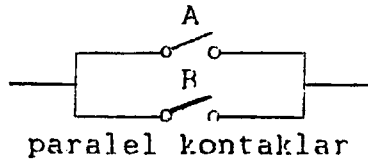


A	NOT A
0	1
1	0



A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

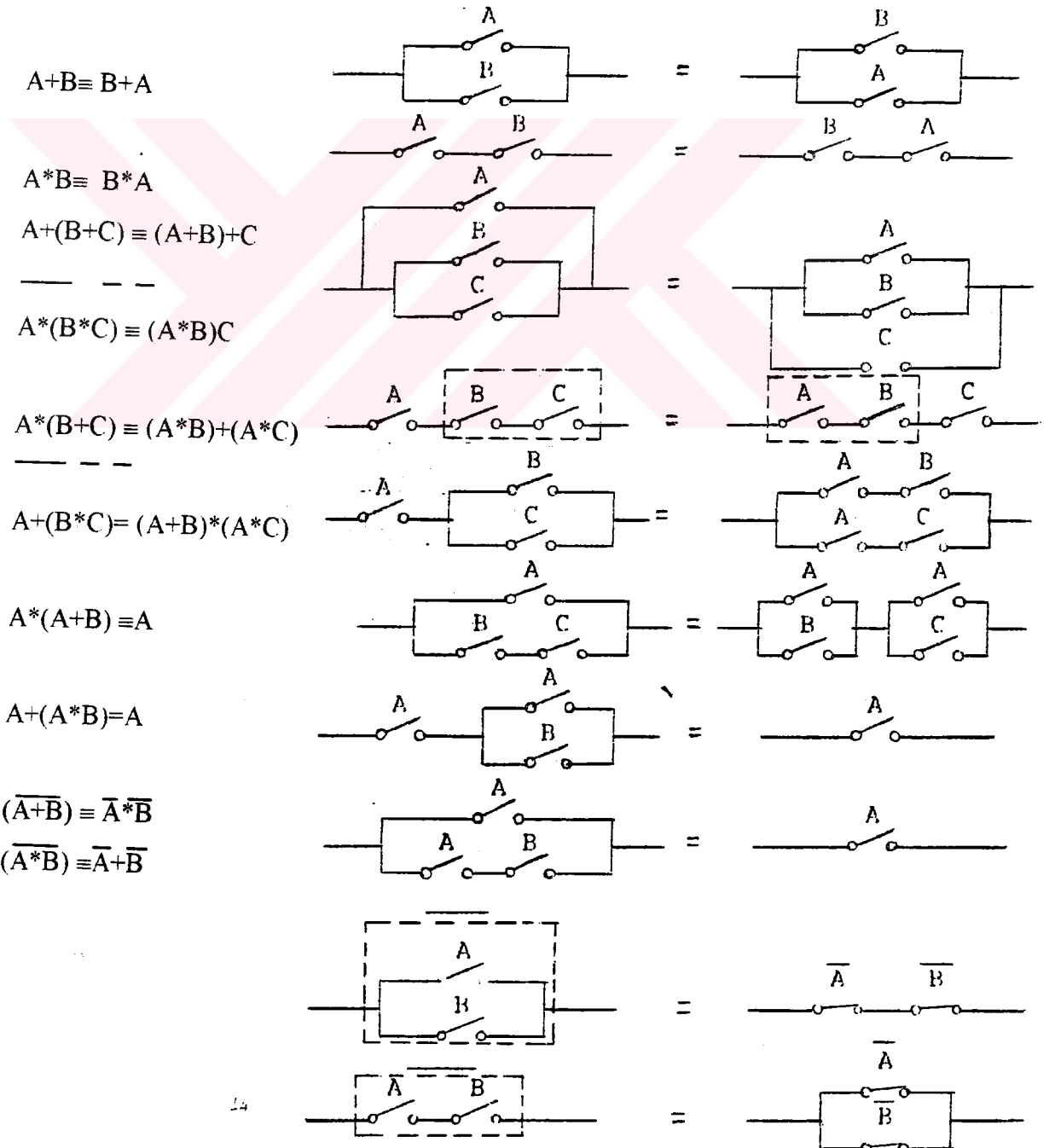
c-



A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

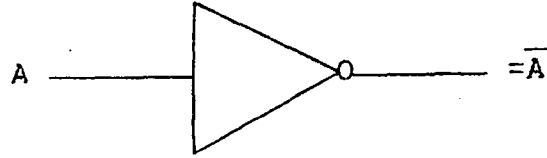
Şekil-4) Lojik (a) değil, (b) ve, (c) veya fonksiyonları.

Boolean cebiri yasaları, anahtar devreleri için alternatif formları bulmamıza yardım eder. Alternatif formları, programı kolaylaştırmak veya gereksiz anahtarları elimine etmek için kullanılır.

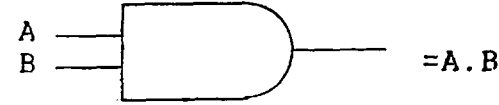


Dijital elektronikte kapı olarak adlandırılan lojik elemanlardan kurulu entegre devreler.

DEĞİL



AND (VE)



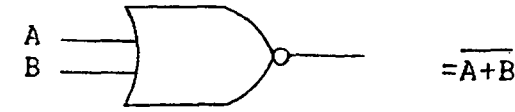
NAND (VE DEĞİL)



OR (VEYA)



NOR (VEYA DEĞİL)

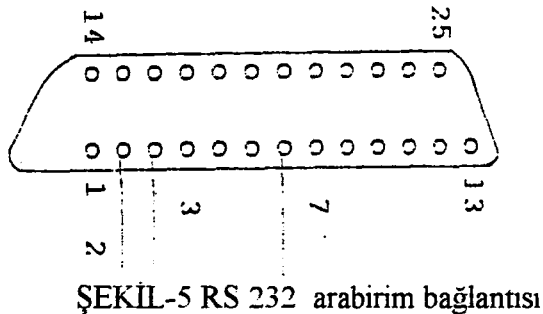


III-PLC'LER ARASINDA İLETİŞİM

Otomatik kontrol sistemlerinde kullanılan PLC ve diğer cihazlar arasında bilgi iletişimi sağlanmalıdır.

Paralel iletişim arabirimleri, verileri iletmek için genellikle 8bit'li bir paralel taşıma hattı kullanılırlar. Bunlar, verilerin kısa mesafelerde hızla iletilmesi için kullanılırlar. Çok kullanılan iki paralel iletişim arabirimi "Centronics" ve "IEEE-488" dir.

Seri iletişim arabirimi, bir bit veya veriyi aynı anda alıp iletebilir. Seri iletişim arabirimleri verileri uzak mesafelere iletmek için kullanılmaktadır. En genel amaçlı standart seri bağlantı RS232'dir. RS232 arabirimi aşağıdaki şekil-4'te görüldüğü gibi 25 uçlu bir fiştir. Minimum kablo bağlantısı 2,3, ve 7 No'lu uçları kullanarak oluşur. Lojik-1 '12v' ile Lojik-0 ise '+12V' ile temsil edilir. İletişim mesafesi yaklaşık 15 metredir.



ŞEKİL-5 RS 232 arabirim bağlantısı

Uç	Tanımlama
1	Koruma toprağı
2	Aktarılan veri (TX)
3	Alınan veri (RX)
4	Göndermek için isteme (RTS)
5	Göndermek için temizleme (CTS)
6	Veri seti hazır (DSR)
7	İşaret toprağı
8	Alınan işareti izleme
20	Veri terminali hazır (DTR)
22	Gösterge ses alarmı

IV -SİMATİK-5 YAZILIM PROGRAMI

IV -1)ONLINE PROGRAMLAMA

Online programlama seçeneği PLC ile direk bir iletişim ağı kurmaya izin verir.Bu şekilde program çalışırken programı kontrol edip gerekli değişiklikleri yapmayı mümkün kılar.Böylece PLC ve programın çalışması aynı anda kontrol edilebilir.

IV-2)OFFLINE PROGRAMLAMA

Offline program seçeneği .PLC'nin çalışmasından bağımsız fakat PLC ile program alış-verişi içinde bir programlamaya izin verir.Offline programı ile PLC'yi direk olarak etkilemeksizin program yazılabilir,değiştirilebilir veya belgelenebilir.

IV -3)SİMATİK -S5 PROGRAMLAMASINA GİRİŞ

Bu bölümde , aşağıda belirtilen temel program unsurlarını inceleyeceğiz.

- Program yapısı ⇒ Programlama tarzını ve çeşitli işlemciler tarafından desteklenen blok tiplerini açıklar.
- Data (veri) yapısı ⇒ Çeşitli işlemciler tarafından desteklenen veri tipleri,veri alanları ve veri formatlarını açıklar.
- Merdiven programlama ⇒Merdiven programlamanın temel eleman ve şartlarını açıklar.
- İfade listeli programlama ⇒Bu programlamanın temel eleman ve şartlarını açıklar.
- Belgeleme : Simatic S5 yazılımıyla kapsanmış veri seçeneklerini listeler.

IV-3-1)PROGRAM YAPISI

Program belirli işleri yapan veri ve mantık bloklarına bölünmüştür.Temel olarak 5 tip blok kullanılır.

- Organizasyon bloğu (OB): Genel olarak ,organizasyon blokları kullanıcı tarafından programlanır ve belirtilmiş şartlar altında sistem tarafından çağrılır.Organizasyon bloklar için ilk çağrılan bloktur.

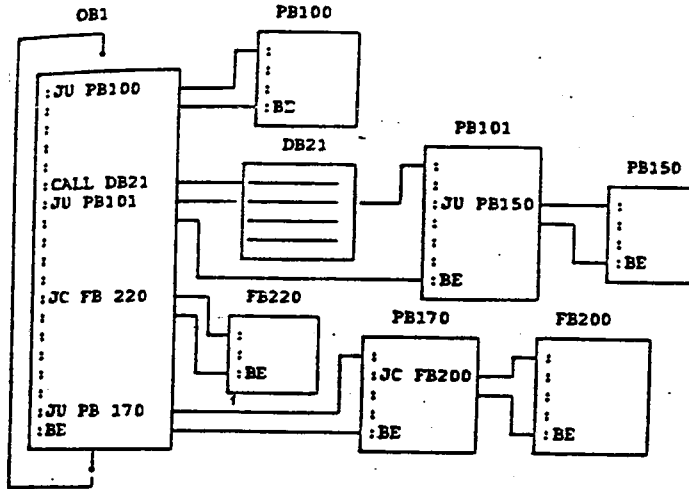
- Program Bloğu (PB): Program blokları işlemci tarafından uygulanacak olan,kullanıcı tarafından tanımlanmış mantığı içerir.Bu bloklardaki mantık “ifade” (statement) olarak tanımlanmış talimat serileri şeklindedir.Herbir ifade blok sonuna gelene kadar yada bir hata oluşana kadar işlem görür.

- Fonksiyon Bloğu (FB): Program bloklarına benzer olarak fonksiyon blokları da program talimatları içerir.Fonksiyon blokları tekrarlı veya seri aksiyonları veya hesaplamaları yapmak için kullanılır.

- Veri Bloğu (DB): Veri blokları program tarafından kullanılan veriyi saklar.Bu veriler hesaplama sonuçları, I/Q durumları,sabitler vb. olabilir.Verdi blokları ifade talimatı içermezler.

- Sıra Blokları (SB): Sıra blokları programlama sıralayıcısı için kullanılmış program bloklarının özel bir şeklidir.

Aşağıdaki şekil-5 program çevrimini göstermektedir.Sistem OB1’i çağırır.Bu blokta diğer blokların çağrılmasını yönetir.Verdi blokları diğer bir veri bloğu çağrılana kadar aktif kalır.Diğer bloklara “Jump “ komutu ile ulaşılır.



Şekil- 6 Örnek program yapısı

IV-3-1-1) ORGANİZASYON BLOKLARI : Organizasyon blokları (OBS) herhangi bir kullanıcı programda temel elemandır ve sistem programı ve kullanıcı programı arasında arabirim olarak iş görür.

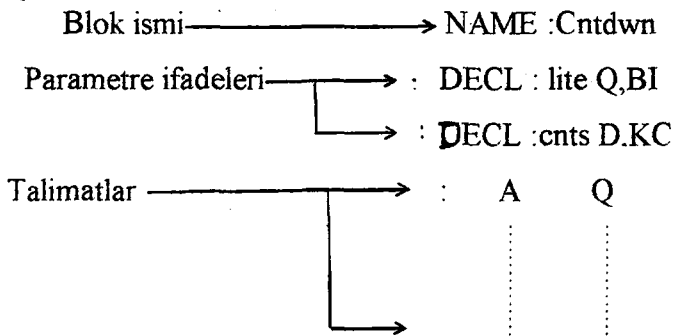
Organizasyon bloklarının sayısı ve tipi kullanılan PLC'ye bağlıdır. Genel olarak kullanıcı tarafından belirtilmiş durumlar meydana geldiğinde çalıştırılmak için programlanır ve sistem tarafından çağrılır.

Organizasyon bloğu-1 (OB1) işlem çevrimi tarafından saklanır ve herbir işlemci çevriminde sistem programı tarafından çağrılır. Diğer organizasyon blokları kontrolör hataları, çevrim startı ve batarya hatalarında kullanılır. Bu bloklar merdiven veya ifade listesi ile programlanabilir.

IV-3-1-2) PROGRAM BLOKLARI ⇒ Program blokları lojik yapısını içerir. Program blok ifadeleri sıra ile bir blok çağrılana veya "BE" ifadesine varılana kadar çalışır. Program blokları maximum 256 adettir. (PB0-PB255) Fakat bu sayı kullanılan PLC'ye göre daha az olabilir. Bu bloklar merdiven veya ifade listesi şeklinde programlanabilir.

IV-3-1-3) FONKSİYON BLOKLARI: ⇒ Fonksiyon blokları tekrarlı ve karışık olayları ve hesaplamaları yapmak için kullanılır. Bazı işlemciler aynı zamanda genişletilmiş fonksiyon blokları (FX) içerebilir. Bu bloklar ek program esnekliği temin eder ve fonksiyon bloklara benzer olarak çağrılır.

Fonksiyon blokları blok başlığı ve blok çatisi olmak üzere iki bölümden oluşur.



Blok ismi : Herbir fonksiyon bloğa bir isim verilir. Bu isim 1~8 karakterli olabilir.

Parametre ifadeleri: Blokta kullanılan deęişken ve resmi operatörlerin bir listesidir. Herbir parametre "DECL" talimatı kullanılarak elde edilir.

DECL : Parametre isim, parametre tipi

veya DECL : paramtere isim, parametre tipi, Data tipi

veya DECL : parametre isim, parametre tipi ,Data formatı

Parametre ismi: 1~ 4 karakter uzunluęunda ve alfabetik bir karakter ile başlamalıdır

Parametre tipi: Herbir parametre ařaęıdaki parametre tiplerinden biriyle tanımlanır.

I = Giriř

Q = Çıkıř

D = Veri

B = Blok

T = Tineer

C = Sayıcı

Veri tipi: Eęer giriş veya çıkıř parametre tipleri belirlediyseniz ařaęıdaki veri tiplerinden hangisini kullanacaęınızı belirtmelisiniz.

BI =Bit

BY =Byte

W =Kelime

D =Çift kelime

Veri formatı: Eęer veri tipi saptandıysa, ařaęıda belirtilmiř veri formatlarından biri belirtilmelidir.

KM =BINARY

KH =HEXADEIMAN

KY =Z BITS

KS =ASCII TEXT

KF =Sabit nokta

KT =Timer deęer

KC =Sayıcı deęeri

IV-3-1-4)VERİ BLOKLARI: Veri blokları (DBs),program tarafından kullanılan verileri depolar ve gösterir.Bazı işlemciler genişletilmiş veri blokları (OX) de içerebilir.Verii işlemci belleğinde daima binary şeklinde saklanır.

Toplam 225 adet veri bloğu vardır.Bu sayı PLC tipine göre azalabilir. ”DBO” kullanıcı tarafından oluşturulmuş bloklar için bir adres listesine sahiptir.Bu blok otomatik olarak sistem programı tarafından oluşturulur ve sistem kullanımı için saklanır.

IV-3-1-5)SIRA BLOKLARI: Sıra blokları, sıra kaskadlarını organize eden sistem fonksiyon bloklarından çağrılır.Sistem fonksiyon blokları sıra kontrol fonksiyonları ile önceden programlanabilir.

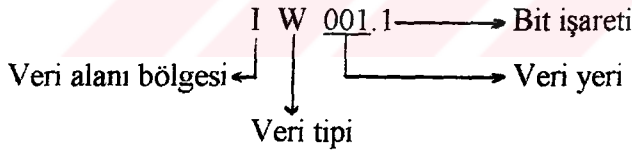
Sıra kontrolü için fonksiyon blokları herbir sıra bloğun lojik işlem sonucunu (RLO) kullanır.

a-Komut bölümü: Bu bölüm RLO’ya bağlı olan ifadeler ile ilk olarak programlanır ve adım tarafından temin edilmiş aksiyomları yerine getirir.

b-Operasyon bölümü: Bu bölüm gelecek sıra bloğun yapılıp yapılmadığını belirten bir RLO üretir.

IV-3-2)VERİ YAPISI

Tipik bir veri aşağıdaki gibidir.

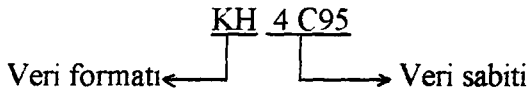


Veri bölgesi: Bir veya iki mnemonic karakter olarak veri bölgesini belirtir.

Veri tipi: Verinin uzunluğunu belirtir.(Byte: B veya Y), (Word =W gibi)

Veri yeri :Veri bölgesindeki verinin yerini belirtir.(adres,bit değer,sayı vb.)

Bit işareti: Veri bölgesindeki bir bit’ı belirtir.Maximum bit değeri “0~31” arasındadır.Verii aynı zamanda programa sabit veri şeklinde direk olarak girilebilir.



Veri formatı: Değeri okumak için kullanılmış formatı belirtir.

Veri sabiti: Kullanılan değeri belirtir. Bu değer belirtilmiş veri formatı için geçerli olmalıdır.

IV-3-3)MERDİVEN DİYAGRAM MANTIĞI

Merdiven diyagram mantığı ,bir programlama dilidir. Bu dilde girişler ve çıkışlar arasındaki ilişki bir kablolama diyagramı şeklinde temsil edilir. Merdiven diyagramı parçalara (segment) bölünmüştür. Herbir parça bir veya daha fazla elektriki çıkışa giden yatay güç hatlarını gösteren bir yol veya yol ağı içerir. Herbir parça maximum 16 paralel çıkışa sahip olabilir.

IV-3-4)İFADE LİSTESİ MANTIĞI

Bu programlama tipi program talimatlarını oluşturmak için grafik semboller yerine mnemonic ifadeler kullanır. İfade listesi gösterimi grafik şekillerden daha basit dil elemanları kullanması sebebiyle, bu program daha sağlıklı, daha az belleğe gerek duyan ve genel olarak daha hızlı çalışan programlama tipidir.

İfade listesi talimatları aşağıdaki elemanlardan oluşur.

İşlem: Talimatlar ile yapılmış aksiyomu belirtir. Mnemonic veya aritmetik semboller şeklinde olabilir.

İşlemci: İşlem tarafından kullanılan veridir. İşlemci veri tipini gösteren bir mnemonic sembol ve onu takiben veri adresini gösteren bir sayıdan oluşur.

IV-3-5)PROGRAM DÖKÜMANLARI

Simatik S5 yazılım blok ve parça (segment) bilgileri ve sembolik programlamayı içeren belge niteliklerini temin eder.

<u>İfade liste dökümanı</u>	<u>Merdiven dökümanı</u>
Blok başlık mevzuu(metni)	
SEGMENT 1	
Segment başlık metni	
İsim: Örnek	
Tabaka: JU PSI :ifade yorumu	
: Bel	

V) DİJİTAL ARABİRİM ELEMANLARI

V-1) GİRİŞ

Bu bölüm, bir işlemde veya makinede kullanılan kontrol elemanlarının yapısını ve çalışmasını içerir. Basit bir anahtar dijital giriş elemanına bir örnektir. Röleler, selenoidler ve motorlar gibi işletme araçları anahtarlarla kontrol edilmektedir.

Uzunluk, sıcaklık ve zaman gibi birçok nicelikler doğal olarak analog büyüklüklerdir. Dijital dedektörler olarak isimlendirilen birçok eleman analog miktarları ölçerler fakat dijital çıkış üretirler. Kodlayıcılar, lineer yerdeğiştirmeyi veya dönüş hızı gibi analog büyüklükleri ölçen elemanlardır.

Analog sinyallerin, CPU tarafından anlaşılabilmesi için dijital büyüklüklere dönüştürülmesi gerektiğinden özel devreler gerektirirler.

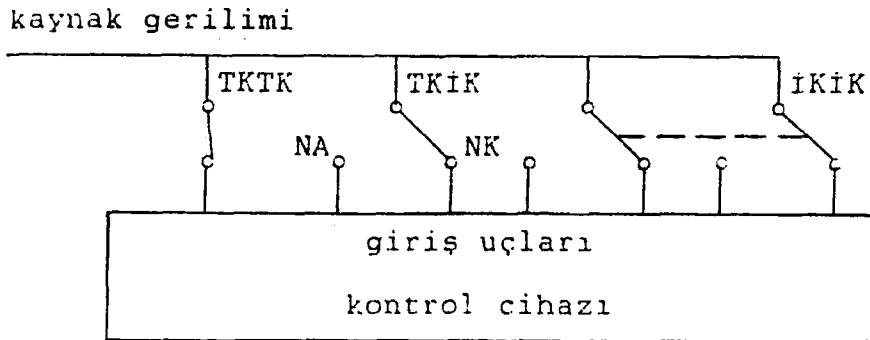
V-2) Dijital Giriş Elemanları

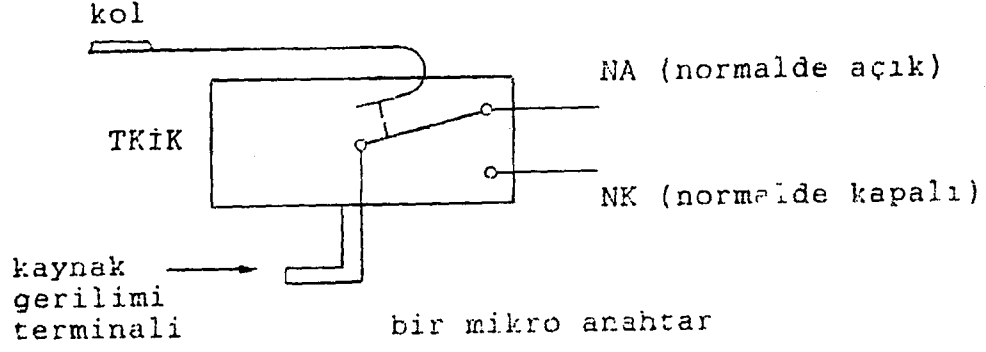
Dijital dedektör veya dönüştürücüler (transducer) akım beslemeli veya sönümlü olarak PLC uçlarına bağlanır. Akım beslemeli eleman, giriş ucuna akım sağlayan, akım sönümlü ise giriş ucundan akımı 0V seviyesine çeken elemandır.

V-2-1) Mekanik Anahtarlar

En basit dijital elemanlar mekanik anahtarlardır. Bunlarla kutup sayılarına ve kullanılan pozisyon sayısına bağlı olarak çeşitli düzenlemeler mümkündür.

Basit bir aç/kapa anahtar tek kutup tek kontak (TKTK) şeklindedir. Bir mikro anahtar tek kutup iki kontak (TKİK) şeklindedir. Şekil 7'de bu tip anahtarların devreye bağlanması görülmektedir.

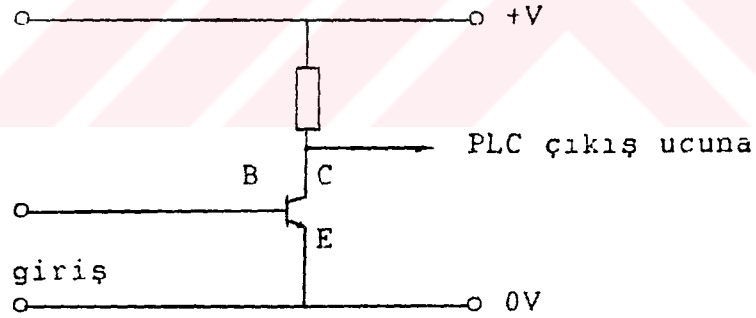




Şekil 7) Anahtar Bağlantıları

V-2-2) Transistör Anahtarları

Şekil 8'de, NPN transistörlü bir anahtarlama devresi görülmektedir. İdeal olarak baz-emiter birleşimi kesime kutuplandığında kollektör-emiter arası açık devredir. Baz-emiter birleşimi iletme kutuplandığında ise kollektör-emiter arası kısa devredir. Bir transistörün ideal anahtarlama, kollektör akımının maksimum doyma değerine ulaşması için baz akımının yeterince büyük olmasını gerektirir. Doyumda ise kollektör-emiter arası gerilim oldukça küçük bir değere düşer.



B=Baz

E=Emiter

C=Kollektör

Şekil 8) NPN transistör anahtarı

V-2-3)Yaklaşım Anahtarları (Proximity Switch)

Yaklaşım anahtarları, fiziksel bir kontak kurmadan bir objenin varlığını denetleyen kontaklı katı hal anahtarlarıdır. Mekanik anahtarların tersine hareketli parçaları yoktur. Bunlar, mekanik anahtarların çalışmasını etkileyen yağlı ve tozlu ortamlarda kullanılmaya müsaittir. Endüktif ve kapasitif olmak üzere iki çeşit yaklaşım anahtarı vardır.

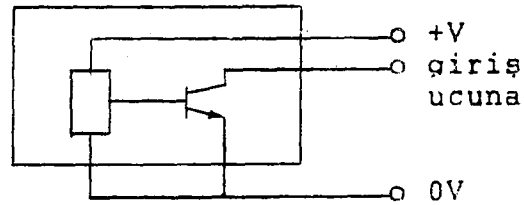
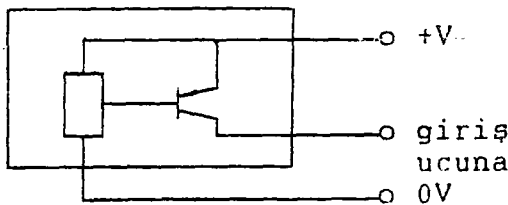
Endüktif yaklaşım anahtarları, yalnızca metal cisimleri denetlerler. Algılama mesafesine giren metal bir objenin, anahtar içindeki bobinin endüktansını değiştirmesi prensibi ile çalışır. Eğer metal bir malzeme, yaratılan elektromagnetik alan içine girerse, endüksiyon yasasına göre malzeme içinde endüksiyon akımları oluşur ve osilatör devresinden enerji çeker.

Kapasitif yaklaşım anahtarları da, bir kapasitörün elektrik alanına yaklaşan bir cismin neden olduğu kapasite değişimini sezebildiği için metal veya metal olmayan cisimlerin denetiminde kullanılır. Kapasitif yaklaşım anahtarları, dielektrik katsayısına bağlı olarak iletken olan veya olmayan tüm malzemeleri algılayabilmektedir. Endüktif yaklaşım anahtarlarındaki gibi cismin hareket etmesi çalışmasını etkilemez.

Doğru gerilim yaklaşım anahtarları Şekil 9'daki gibi NPN ve PNP transistörle kullanılır. Yaklaşım anahtarının algılama mesafesi yaklaştırılan cismin kütlesi kadar malzeme özelliklerine de bağlıdır.

PNP

NPN



algılama elemanı

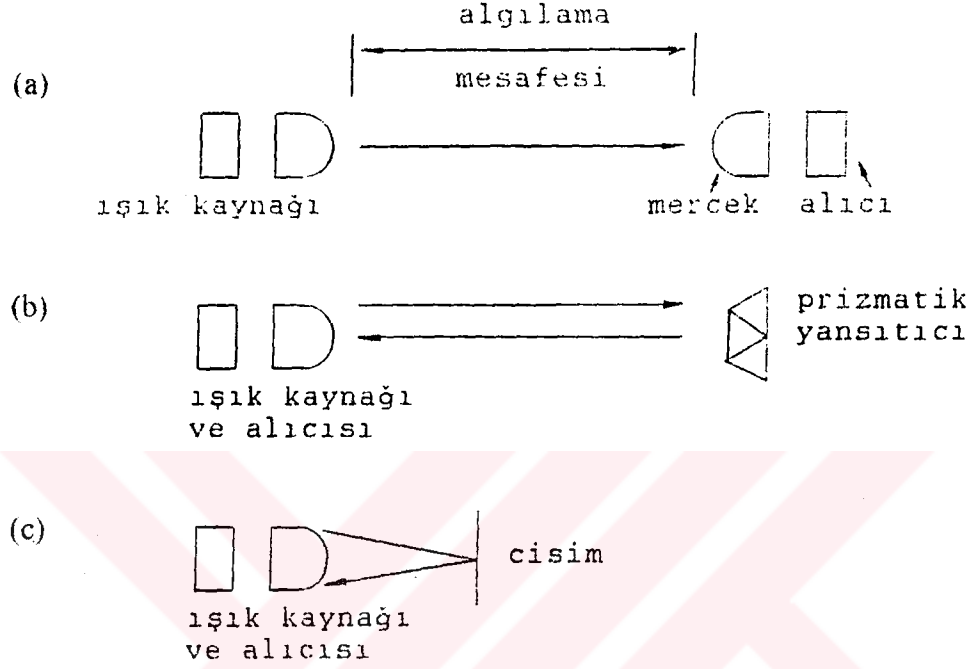
algılama elemanı

+V=Pozitif d.c kaynak gerilimi

Şekil 9) D.C yaklaşım anahtarları

V-2-4)Opto-Elektronik Dedektörler

Opto-elektronik dedektörler,bir ışık kaynağı ve ışık alıcısından oluşan elemanlardır.Bunlar bir ışık hüzmesinin yansımaları veya kesintisini denetleyen elemanlardır.Şekil 10'da görüldüğü gibi karşılıklı,yansıtıcı ve cisimden yansımaları olmak üzere üç çeşit opto-elektronik dedektör vardır.



Şekil 10) (a) Karşılıklı,(b)yansıtıcı,(c)cisimden yansımaları opto-elektronik dedektörler

-Karşılıklı: Işık kaynağı ve alıcı olmak üzere iki ayrı üniteden oluşur.Işık hüzmesini kesen bir obje alıcı içindeki anahtar tetikler.Bu eleman,100 metre mesafeye kadar uzun mesafelerde kullanılır fakat saydam elemanların algılanmasında kullanılmaz.

-Yansıtıcı: Bu tipte ise ışık kaynağı ve alıcısı aynı elemanın içindedir.kaynaktan üretilen ışık hüzmesi,ışık hüzmesini alıcıya geri yansıtan prizmatik yansıtıcıya yöneltilir.Kesilen ışık hüzmesi alıcıyı tetikler.

Geri yansımaları dedektör de denilen bu eleman 5 metreye kadar menzile sahiptir.Polarizasyon filtresi,yanlış tetikleme neden olan kaçak yansımaları önlemek için kullanılır.

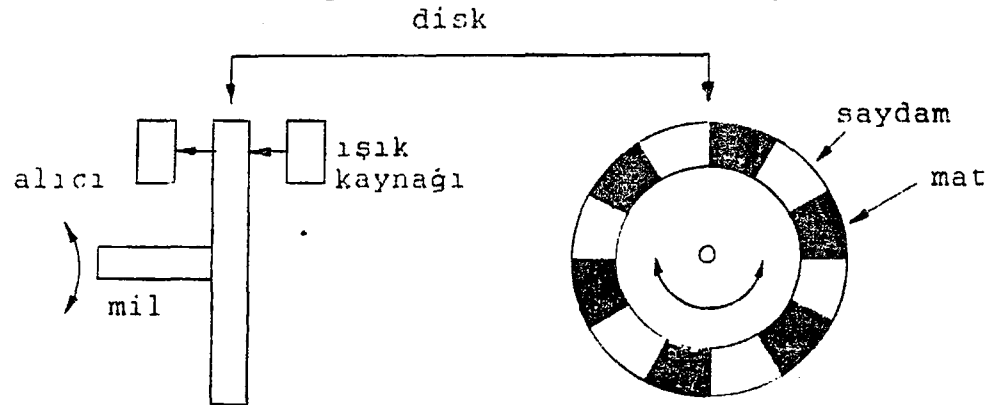
-Cisimden yansımali: Bu elemanlar,cisimin yüzey yansımasi esasina göre çalışır.Genelde algılama mesafeleri kısadır(2 metre).Bu uzaklık cismin yüzey rengine,dokusuna ve yansıtıcılığına bağlıdır.

V-2-5) Kodlayıcılar (Encoders)

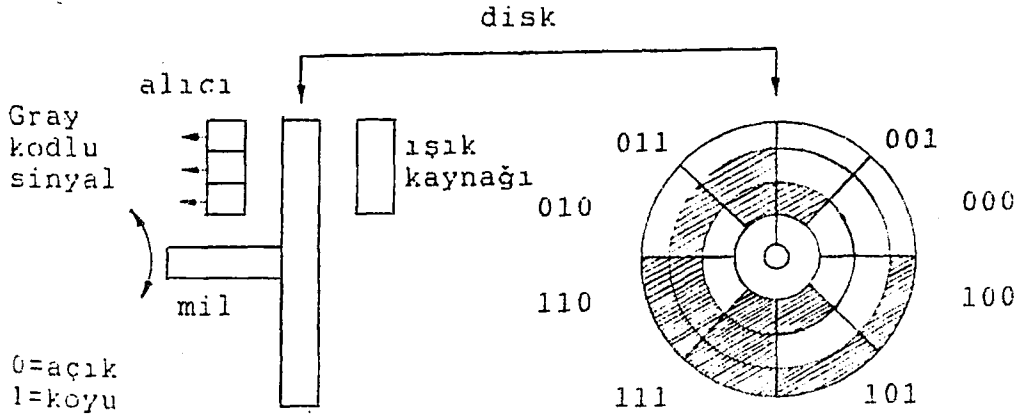
Kodlayıcılar,dönüş hareketi ve doğrusal yerdeğiřtirmeyi dijital sinyallere çeviren elemandır.Doğrusal kodlayıcı,doğrusal yerdeğiřtirmeyi,mil kodlayıcıları da dönüş hızını ölçerler.Mutlak ve artımlı olarak ikiye ayrılırlar.Artımlı (incremental) kodlayıcılar,pozisyonu bir referans noktası ile karşılaştırarak ölçerler.Kesin (absolute) kodlayıcılar,pozisyonlara uyarlanan sayı üretirler.

Artımlı mil kodlayıcısının bir örneđi Şekil 11'de görölmektedir.Bu kodlayıcı,işık kaynađı ve bir ışık alıcısı arasına yerleřtirilen,üzeri eşit aralıklarla saydam ve koyu renklerle bölümlendirilmiş diskten oluşur. Mil döndüğünde,disk de döner ve ışık hüzmelerini açıp kapadıđı için ışık alıcısında bir seri sinyal üretir.Üretilen sinyal referans ile karşılaştırılarak pozisyon ölçölür.Birçok artımlı mil kodlayıcıları, birbirleri ile 90° farklı fazlarda iki sinyal üretebilmesi için yerleřtirilmiş iki opto-elektronik eleman kullanırlar.İkinci sinyalin lojik durumu birinci ile karşılaştırılarak dönüş yönü belirlenebilir.Eđer disk,bir motor tarafından döndürölüyor ise sinyal hızı motor hızı ile orantılı olarak deđişmektedir.

Mutlak mil kodlayıcılarına bir örnek Şekil 12'de görölmektedir.Kodlayıcı,bir ışık kaynađı ve üç alıcı arasına yerleřtirilmiş bir diskten oluşur.Disk,herbir mil pozisyonu için farklı kod üretmesi için saydam ve koyu bölmelere ayrılmıştır.Disk,normal olarak Gray kodu adı verilen bir kod ile işaretlenmektedir.Aşğıdaki tablo Gray kodu ikili sayı dönüşümünü göstermektedir.



Şekil 11) Artımlı mil kodlayıcısı



Şekil 12)3-Bit'lik Gray kodlu mutlak mil kodlayıcısı

Doğrusal kodlayıcılar, açık ve koyu renkler ile işaretlenmiş şeritler kullanırlar. İşletimi diğerlerine benzerdir.

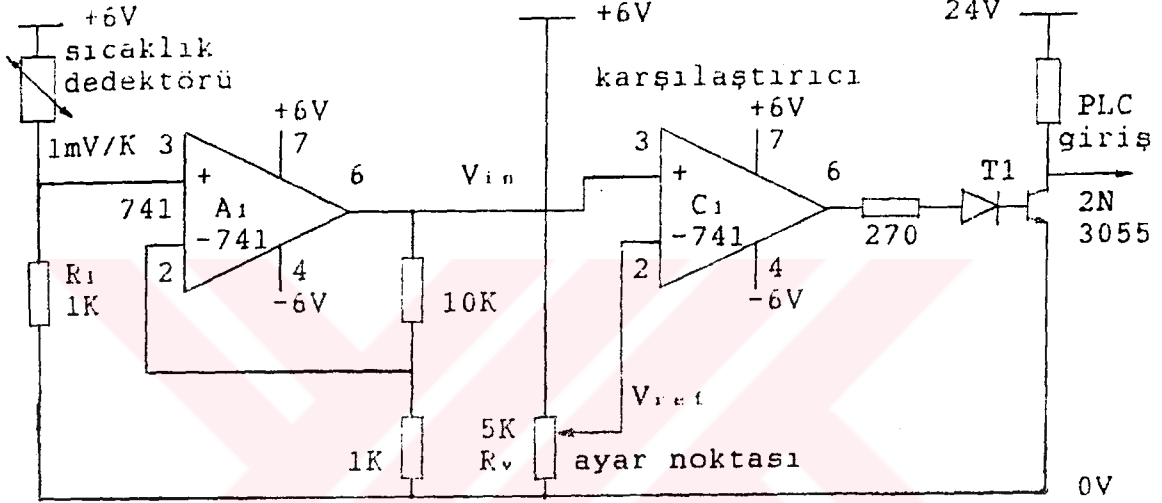
Tablo. İkili ve Gray Kodları

İkili Kod	Gray Kodu
0000	0000
0001	0001
0010	0011
0011	0010
0100	0110
0101	0111
0110	0101
0111	0100

V-2-6)Sıcaklık Anahtarları

Sıcaklık anahtarları, özel bir geçiş sıcaklığı civarında hızla direnç değiştiren yarıiletken elemanlardır. Örneğin, böyle bir eleman 75° C geçiş sıcaklığına kadar $100K \Omega$ gibi yüksek bir direnç gösteriyorsa bu sıcaklık üzerinde ısıtıldığında eleman direnci 100Ω gibi bir değere düşer. Eleman yüksek direnç gösterdiğinde açık devre durumdadır ve küçük bir akım çeker. Uçları arasındaki direnç düştüğünde iletme geçer. Sıcaklık anahtarları, güç yarıiletkenleri, trafo ve motor gibi ısınan elemanları yüksek ısıdan korumak için kullanılır. Bu elemanlar ,sıcaklık alarm girişlerinde kullanılmaktadır.

Termokupl ve yarıiletken sıcaklık dedektörleri, bir karşılaştırıcı yükselteci ile birlikte, dijital sıcaklık anahtarı olarak kullanılır. Şekil 13'de, bir sıcaklık dedektörünün nasıl bir sıcaklık anahtarı olarak kullanıldığı görülmektedir. Sıcaklık dedektörü akımı, sıcaklık oranında geçirir. Elemanın sıcaklık katsayısı $1\mu\text{ A/K}$ değerindedir. Devrede R_1 direnci kullanılarak dedektör akımı gerilime çevrilir. Bu gerilim A_1 tarafından yükseltilir ve diğer C_1 yükselticisini besler. Sıcaklık giriş gerilimi, (V_{in}) ayarlı değerden veya referans girişinden (V_{ref}) daha büyükse veya eşitse karşılaştırıcı çıkışta lojik "1" üretir. Karşılaştırıcı çıkışı, T1 transistörünü anahtarlar bu da PLC gitiş ucuna bir gerilim verir. Anahtarlama sıcaklığı R_v ayarlı direnci ile ayarlanır.



Şekil 13) Sıcaklık anahtarı devresi

V-2-7) Basınç Anahtarları

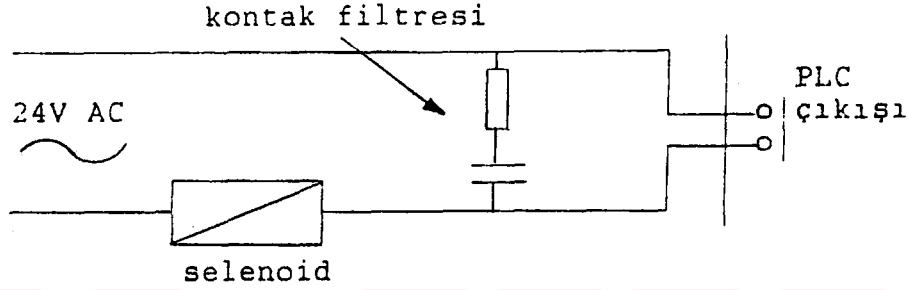
Basınç anahtarları, belirli bir geçiş basıncında açılıp kapanan elemanlardır. Sıvı basıncı oranında hareket eden diyaframın, bir mikroanahtar işletmesiyle çalışır. Sonuçta basınç anahtarı da normal anahtar gibi devreye bağlanır.

V-3) DİJİTAL ÇIKIŞ ELEMANLARI

Dijital çıkışlar, sırası işlem değişkenlerini ve makine hareketlerini denetleyen işletme mekanizmalarını kontrol eder. Bu işletme mekanizmaları, motorlar, katı hal röleleri, kontaktörler ve selenoidlerdir.

V-3-1) Selenoidler

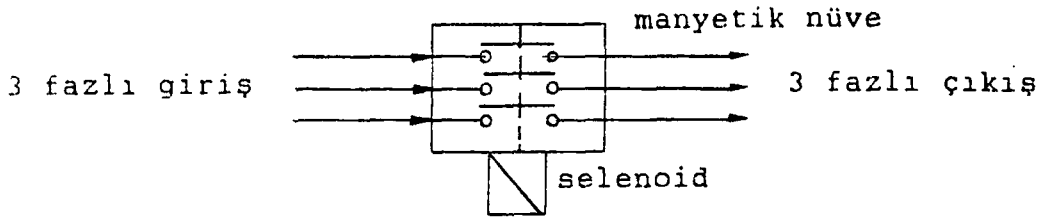
Selenoid, elektrik sinyalini mekanik harekete çeviren bir elektromagnetik elemandır. Aslında bir bobin ve hareketli bir parçadan oluşur. Enerjilenen bobin, parçanın hareket etmesine sebep olur. Hareket uzaklığına darbe ismi verilir. Tipik darbe uzunluğu elemana uygulanan kuvvete göre 4-20 mm arasında değişir. İşletme gerilimleri, 12V DC, 24 V DC /AC ve 240V AC'dir. Şekil 14'te 24V AC bir selenoidin PLC röleli çıkış ucuna bağlantısı görülmektedir. Bir RC filtre devresi de bir reaktif yük anahtarlandığında ortaya çıkan gürültüyü bastırmak için röle kontaklarına bağlanır. Bu, ayrıca rölenin kontak ömrünü de artırır.



Şekil 14) Bir selenoidin PLC çıkışına bağlantısı

V-3-2) Kontaktör

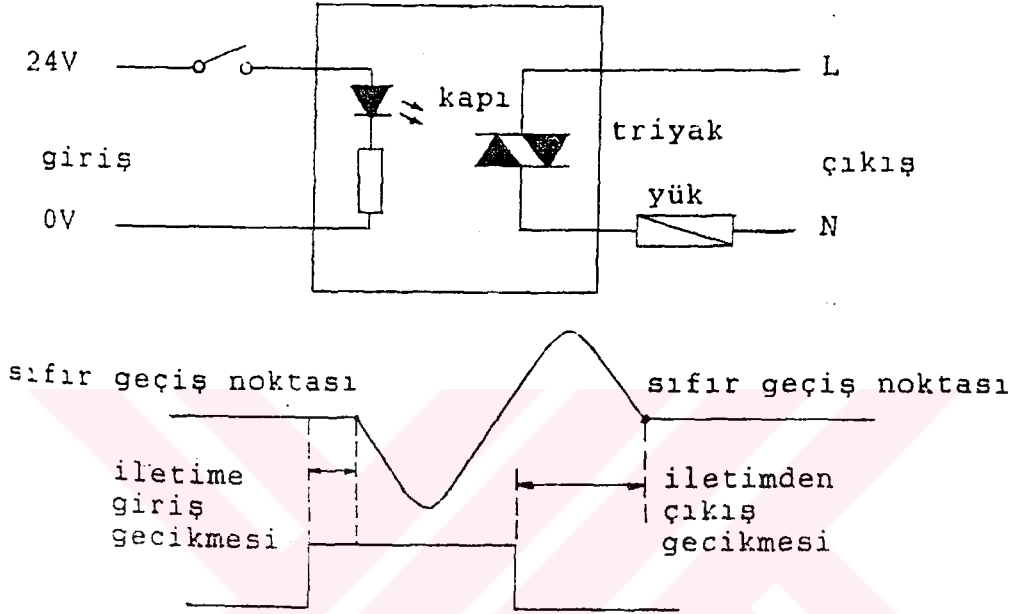
Bir kontaktör, elektriksel veya mekanik olarak işletilebilen bir yüksek akım anahtarıdır. Şekil 15'te bir kontaktör görülmektedir. Bu bir selenoid, manyetik nüve etkilenerek yüksek akıma dayanabilen kontakları kapatır. Kontaktörler bir ve üç fazlı kaynakları anahtarlama için endüstride oldukça yaygın olarak kullanılır.



Şekil 15) Üç fazlı kontaktör

V-3-3)Kati Hal Rölesi

Bir kati hal rölesi,normal röle ile aynı özellikleri gösterir fakat hareketli parçası yoktur.Şekil 16'daki gibi optik olarak izoleli bir triyaktan meydana gelir.Kati hal rölesinde bir led ile tetiklenebilmesi için triyağın kapı ucu optik olarak izole edilmiştir.Kati hal röleleri sıfır geçiş devreleri ile birleştirilerek şebeke anahtarlamalarında kullanılırlar.



Şekil 16)Sıfır geçiş noktasında çalışan kati hal rölesi

VI) ANALOG ARABİRİM ELEMANLARI

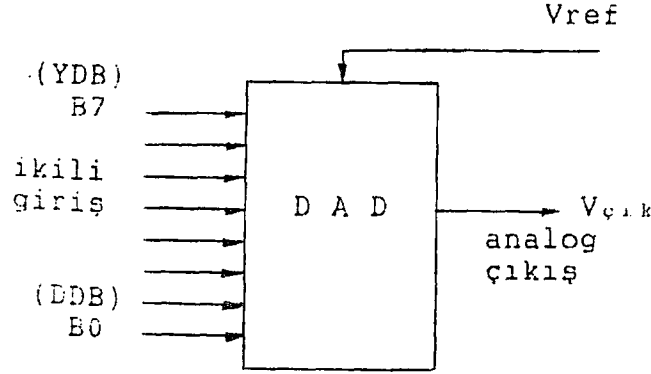
VI-1)Giriş

Daha önceki bölümde incelendiği gibi dijital elemanlar açık ve kapalı olmak üzere iki kesikli seviyeye sahiptirler.Analog sinyaller ise bunlardan farklı olarak seviyeleri değişebilen sürekli sinyallerdir.

PLC'ler dijital cihazlardır.Analog sinyalleri kullanmak için,analog-dijital dönüştürücüler,dijital-analog dönüştürücüler,çoklayıcılar ve tekleyiciler gibi özel arabirimler gerektirmektedirler.

VI-2)Dijital Analog Dönüştürücüler (DAD)

Dijital analog dönüştürücü (DAD).Şekil 17'deki gibi bir dijital girişten analog bir çıkış üretir.DAD'nün bütün tiplerinin analog gerilimi,bir referans geriliminden (V_{ref}) sağlanır. DAD'ye girilen ikili kodlu sinyal, çıkışta temsil edilen V_{ref} geriliminin kesirlerini belirler. Bir DAD çıkışı, sürekli bir sinyal değil fakat bir seri kesikli gerilim seviyesidir.



Şekil 17) 8- Bitlik DAD

Örneğin,şekil 17'de gösterilen 8-bitlik DAD'nün B0ve B7 bitleri "0" veya "1" değerlerini alabilirler ve çıkış ise aşağıdaki gibi verilmektedir.

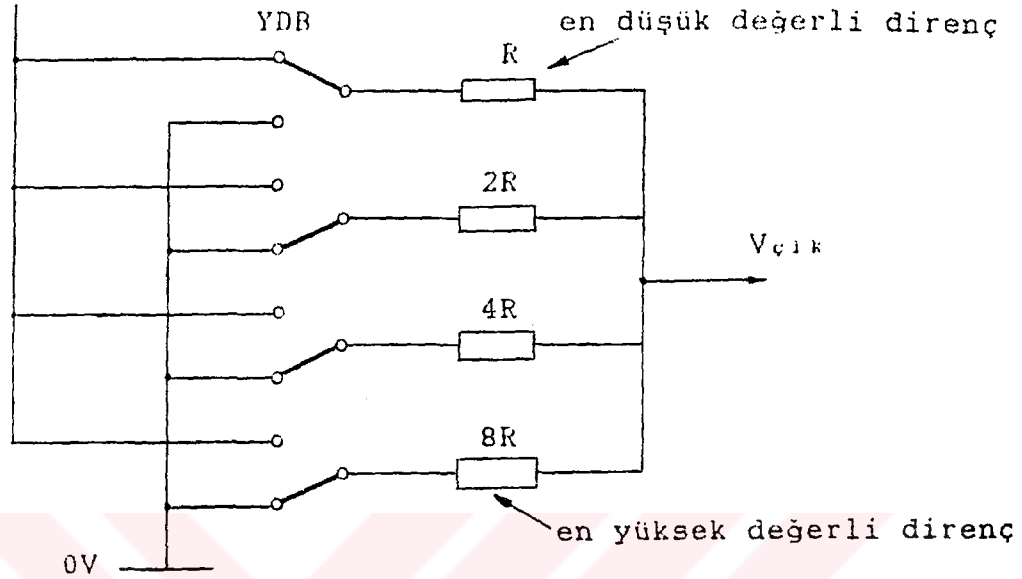
$$V_{çık} = V_{ref} \cdot \left(\frac{B7}{2} + \frac{B6}{4} + \frac{B5}{8} + \frac{B4}{16} + \frac{B3}{32} + \frac{B2}{64} + \frac{B1}{128} + \frac{B0}{256} \right)$$

B7,yüksek değerli bit (YDB) B0'da düşük değerli bittir.(DDB)

Referans gerilimi, (V_{ref}) 10V olan 8-bitlik bir DAD düşünülürse .00000001 ikili girişi, en düşük kesirli çıkışı (örneğin, 10/256 volt) üretir.00000010 iki giriş değeri 10/128 voltluk bir sonraki kesiri üretir. İkili giriş kullanılarak 256 kesirli analog seviye üretilebilir.N-bitlik bir DAD'nün gerilim ayarı maksimum işletme gerilimini 2^{N-1} 'e bölerek bulunabilir. 2^{N-1} çarpanı seviyeler arası basamak sayısını temsil eder.10 volt bir referans gerilimi ile 8 bitlik bir DAD,10/255 volt kararlılıkla çalışır.

DAD'ün hızını,girişteki bir değişiklikten sonra değerinin kararlı bir hale kadar geçen süre belirler.DAD'ün diğer önemli parametreleri linearite ve doğruluktur.Linearite,ikili girişe göre belirlenen çıkış hattından sapma miktarıdır.Doğruluk ise DAD gerçek çıkışı ile istenen çıkış arasındaki farktır.

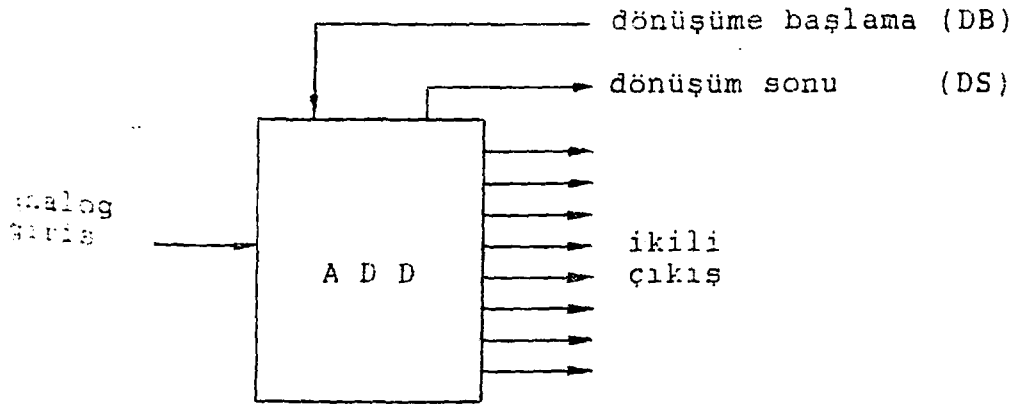
Bir DAD'ün işletme prensibi,Şekil 18'deki gibidir.DAD'ler V_{ref} geriliminden analog çıkış üretmek için ikili giriş tarafından anahtarlanan bir grup ağırlıklı direnç kullanılır.En ağırlıklı direnç analog çıkışta en küçük belirli değeri üretir.



Şekil 18)DAD'ün çalışma prensibi

VI-3)Analog Dijital Dönüştürücüler

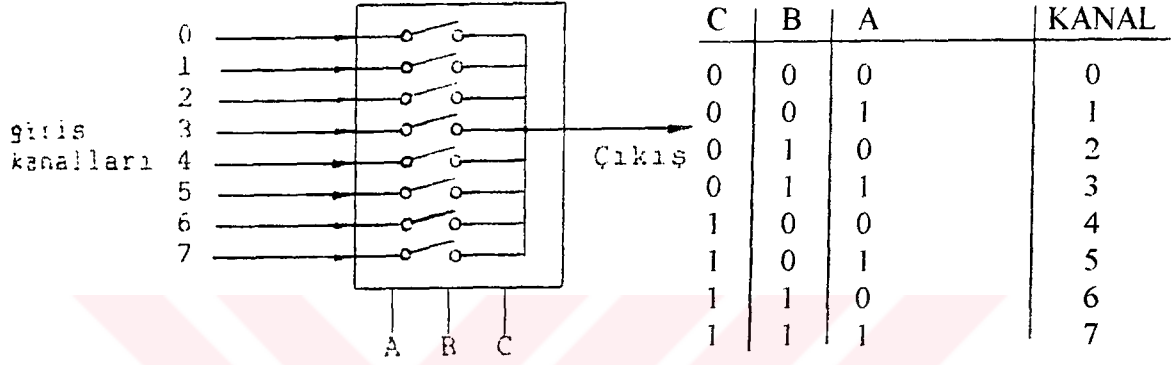
Analog dijital dönüştürücüler (ADD), bir analog girişten dijital bir çıkış üretirler.ADD'ler, dönüşüme başlama (DB) ve dönüşüm sonu (DS) uçlarını içerirler.DB dadesi uygulandığında,ADD o anda analog girişi eşdeğer ikili sinyale çevirir.ADD,daha sonra dönüşümün bittiğini belirten DS sinyalini üretir.ADD'nin ana parametreleri,kararlılık, doğruluk,linearite ve hızdır.



Şekil 19)8-Bitlik analog dijital dönüştürücü

VI-4)Coklayıcılar (Multiplexers)

Çoklayıcılar,birçok sinyalin tek bir hattı paylaşmasını mümkün kılarlar.Şekil 20’de birçok çoklayıcının blok diyagramı görülmektedir.Bu şekilde her bir giriş kanalı, çoklayıcının içindeki anahtarlardan biri kapatıldığında çıkış hattına bağlanır.Pratikte eleman içindeki anahtarlar A,B ve C hatları tarafından kontrol edilen bir grup transistördür.A,B ve C hatlarına girilen bir ikili sayı çıkışta anahtarlanacak kanalı belirler.Tekleyiciler (De-multiplexers), çoklayıcıların tersine çalışan elemanlardır .



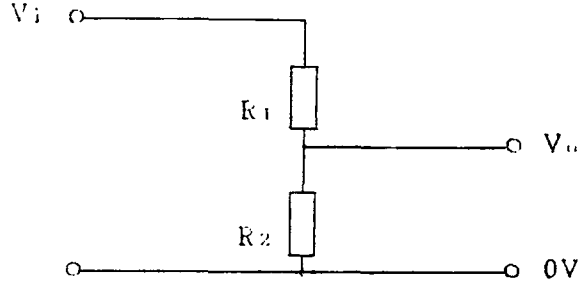
Şekil 20) Çoklayıcı (Multiplexer)

VI-5) Aradevrelere

Analog sinyalleri iletmenin genel kuralı, gerilim devriyelerini eşlemek ve kaynak devresi empedansını yük devresi empedansına eşit veya yakın olmasını sağlamaktır.Yük devrsine optimum güç transferi için empedans adaptasyonu zorunludur.Gerilim seviyelerini eşlemek, bizi gerilim seviyesini artırmaya veya azaltmaya belkide iki kutuplu gerilimi tek kutuplu gerilime dönüştürmeye zorlar.Empedans değerini denetlemek için empedans değıştirici bir devreye gerek vardır.

Bir gerilim bölücü, Şekil 21’deki gibi giriş sinyalinin gerilim seviyesini düşürmek için kullanılan seri bağlı direnç grubudur. V_0 çıkış gerilimi, aşağıda formülde verildiği gibi V_i giriş geriliminden daha düşüktür.

$$V_0=V_i \cdot R_2 / (R_1 + R_2)$$



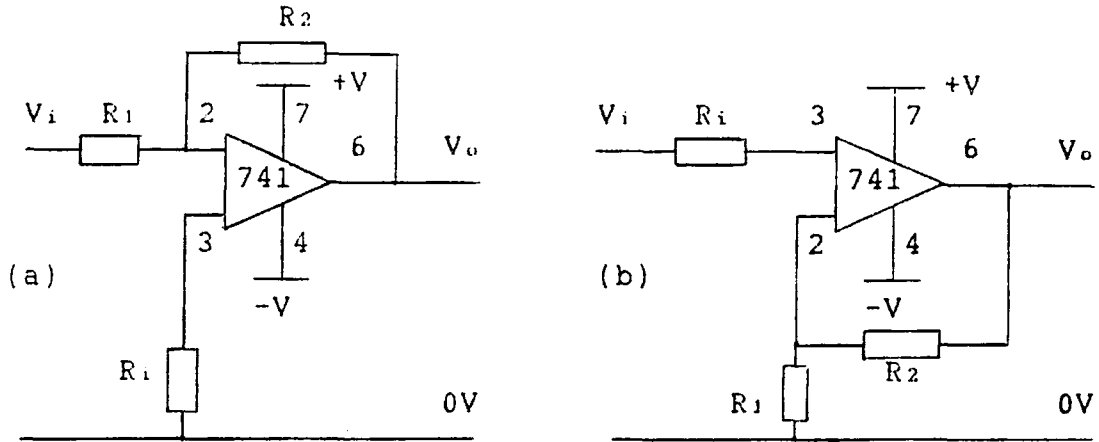
Şekil 21) Gerilim bölücü

Gerilim seviyesini yükseltmek için eviren veya evirmeyen yükselteçler kullanılabilir. 741 işlemsel kuvvetlendirici (op-amp) kullanılarak eviren veya evirmeyen yükselteç devresi Şekil 22'de görülmektedir. Çıkış geriliminin (V_o), giriş gerilimine (V_i) oranına yükseltecin kazancı denir. Eviren yükselteçler $-R_2/R_1$ lik kazançla sahip olduklarından negatif çıkış üretirler. Evirmeyen yükselteçlerin kazancı, $(R_2+R_1)/R_1$ dir. Devrelerdeki R_i direnci, işlemsel kuvvetlendiricinin her iki girişinin 0V da aynı direnci göstermesini sağlar ve aşağıdaki formülle hesaplanır.

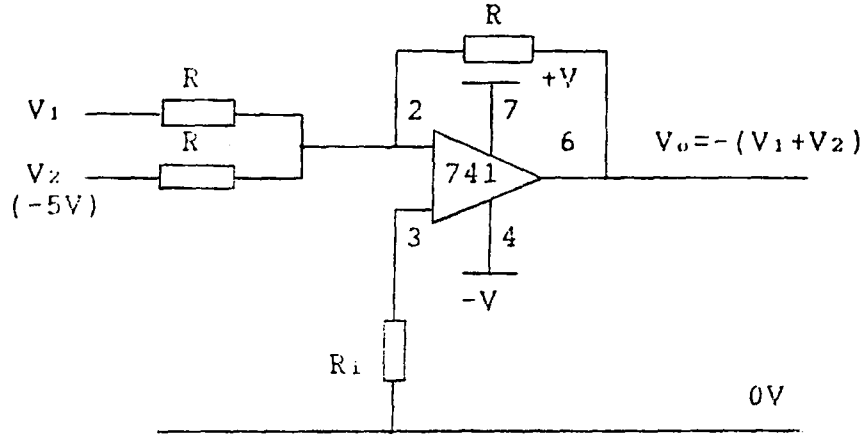
$$R_i = R_2 * R_1 / (R_1 + R_2)$$

Bir toplam yükseltici, iki kutuplu gerilimi tek kutuplu gerilime dönüştürmek için kullanılır. Şekil 23'de, 741 işlemsel kuvvetlendiricisi ile yapılan bir toplam yükseltici devresini göstermektedir. V_o çıkış gerilimi şöyle verilir.

$$V_o = -(V_1 + V_2)$$



Şekil 22) (a) Eviren, (b) evirmeyen yükselteç



V1	V2	V0 = -(V1 + V2)
-5V	-5V	+10V
0V	-5V	+ 5V
+5V	-5V	0V

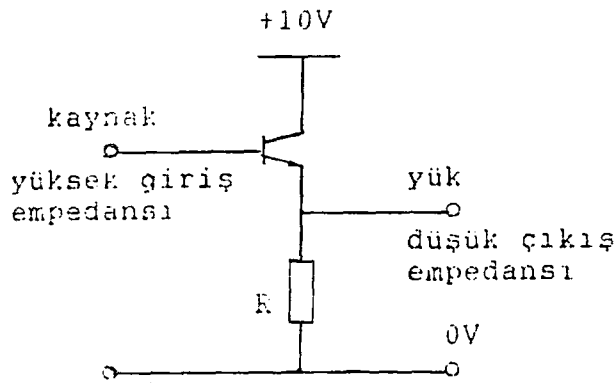
iki kutuplu
giriş

tek kutuplu
çıkış

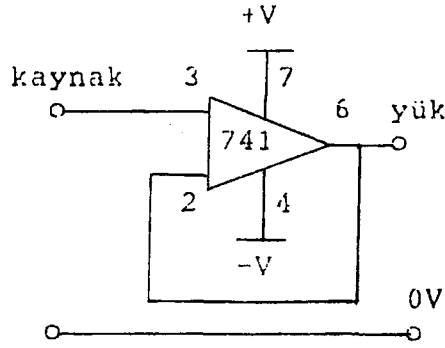
Şekil23) Toplam yükseltici

Eğer yükseltici, iki kutuplu gerilimi (bipolar) tek kutuplu gerilime (unipolar) dönüştürmek için kullanılıyorsa girişlerden biri uygun bir negatif gerilimde tutulmalıdır. Örneğin ; V2 -5V da tutulursa toplam işlemi, V1 gerilimini 0V ile 10V arasında tek kutuplu gerilime dönüştürür (Şekil 24).

Emiter takipçisi ve gerilim takipçisi devre, yük ve kaynak arasında bir empedans sağlar. Kaynak, bir yükseltici veya dijital analog dönüştürücü olabilir. Şekil 24'de emiter çıkışlı ve gerilim takipçisi devreler görülmektedir. Bu devrelerde çıkış gerilimi, giriş gerilimini takip eder fakat giriş empedansı çıkış empedansından oldukça büyüktür. Böylece bu devreler kaynak empedansını düşürürler.



(a)



(b)

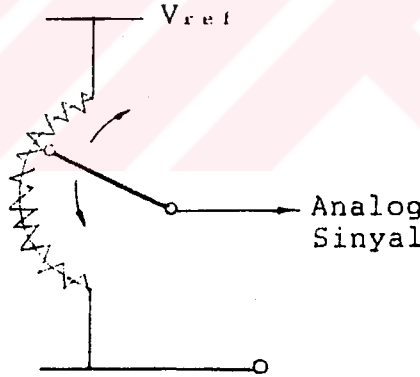
Şekil 24)Empedans değıştiren (a) emiter takipçisi,(b) gerilim takipçisi,devreler

VI-6) Analog Dönüştürücüler

Konum değışikliği,sıcaklık ve basınç gibi fiziksel büyüklükler, dönüştürücülerle analog gerilime veya akıma çevirilir.Aşağıda bazı dönüştürücü tipleri açıklanmaktadır.

VI-6-1) Potansiyometreler

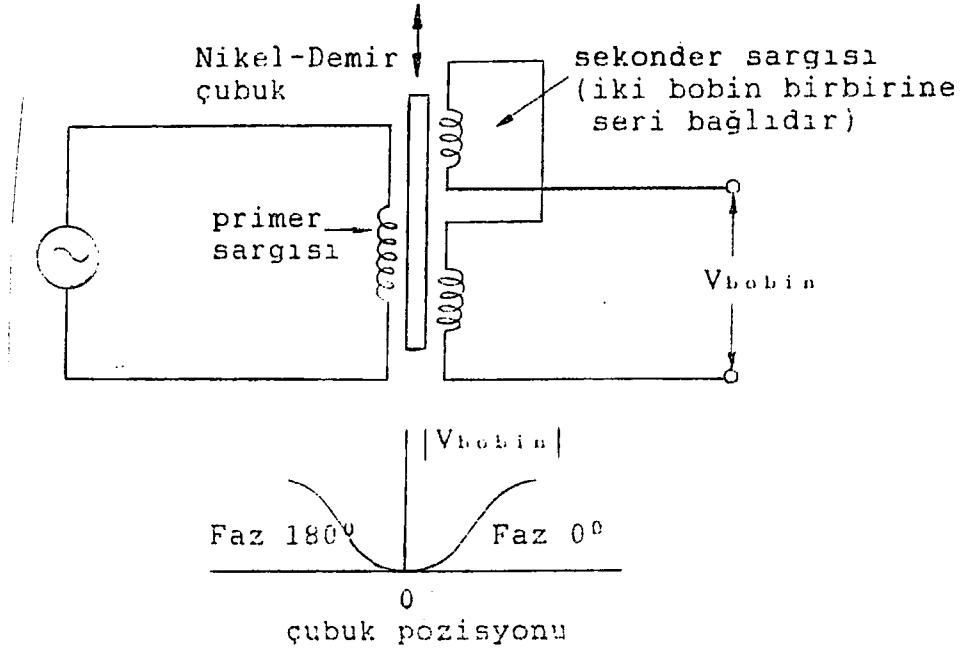
Bir ADD'ye analog giriş sağlamanın en basit yolu, Şekil 25'teki gibi bir potansiyometre devresi kullanmaktır.Potansiyometrenin orta ucu,bağlı bulunduğu kontağın fiziksel konumunu bir gerilim değerine çevirir.Doğrusal ve dönel potansiyometreler, düşük fiyatlı dönüştürücüler olarak çok kullanılmaktadır.



Şekil 25) Dönel pozisyon potansiyometresi

VI-6-2) Doğrusal Değişebilir Diferansiyel Transformatör

Doğrusal değışebilir diferansiyel transformatör (DDDT), bir konum dönüştürücüdür.Bu eleman,Şekil 26'daki gibi primer ve sekonder bobinleri arasında serbestçe hareket edebilen nikel-demir bir çubuktan oluşur.

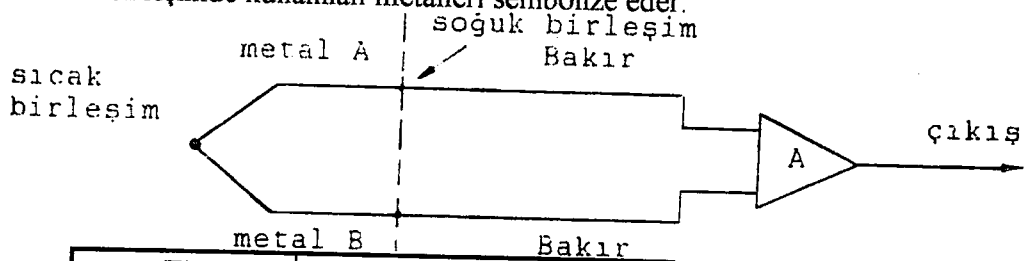


Şekil 26) DDDT

Primer bobini, sekonder bobinin her iki yarısında da gerilim endüklensin diye alternatif gerilim ile beslenir. Yukarı ve aşağı doğru hareket eden çubuk, sekonder sargıdaki gerilimi ve fazını değiştirir. Nüve, merkeze doğru hareket ettiğinde girişe (primer sargısı) göre çıkışın faz açısı 180° değişir. Böylece istene her nüve konumu ile bir fazdedektörü oluşturulur.

VI-6-3) Termokupl

Termokupl, Şekil 27'deki gibi düzenlenen iki benzer olmayan telden meydana gelir. Bu birleşim, ısıtıldığında termoelektrik etki ile gerilim üretilir. Termokupl tipleri harflerle gösterilir. Bu harfler birleşimde kullanılan metalleri sembolize eder.

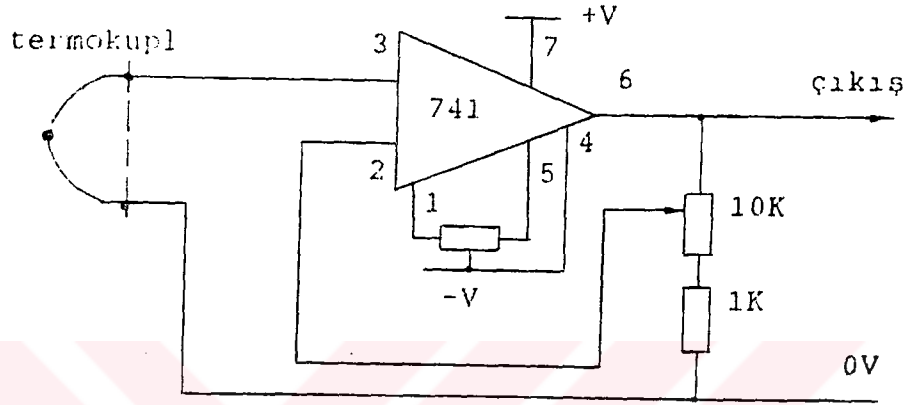


Tip	Metal A/Metal B
E	Krom/Konstantan
J	Demir/Konstantan
K	Krom/Alüminyum
T	Bakır/Konstantan

Şekil 27) Termokupl

Termokupllar, doğrusal olmayan elemanlardır yani çıkış gerilimleri sıcaklıkla orantılı olarak artmaz. Gerçekte bir termokupl, çıkış gerilimini sıcaklığa çevrimin gösterem kalibrasyon tablosuna göre beslenir.

Termokupldan üretilen gerilim bir ADD ünitesi girişine uygulanmadan önce yükseltilmelidir. 741 op-amp ile basit bir yükseltici devresi, Şekil 28'de görülmektedir.



Şekil 28) Termokupl yükseltici devresi

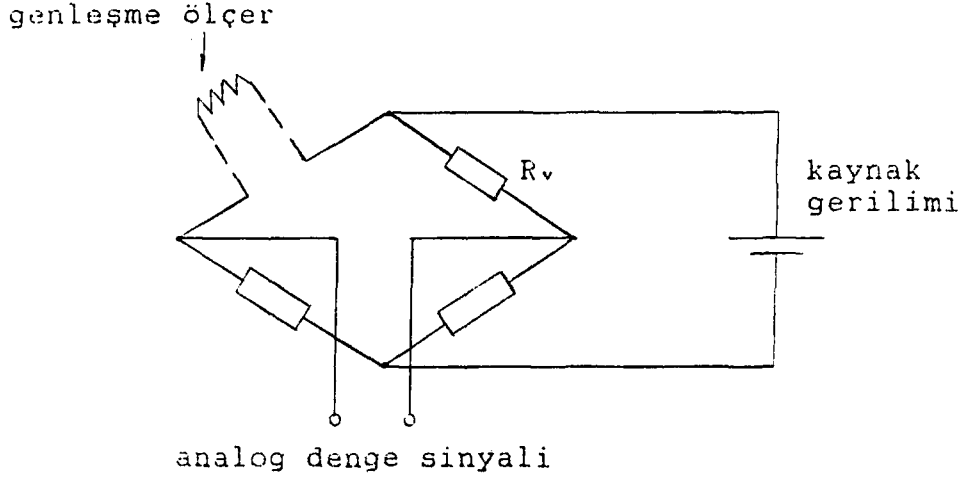
VI-6-4) Genleşme Ölçer

Genleşme ölçer, uzatıldığında veya sıkıştırıldığında direnci değişen elemandır. G, kalibre faktörü olarak isimlendirilirse, direnç değişimi ($\Delta R/R$) ile basınç değişimi ($\Delta L/L$) arasındaki ilişki aşağıdaki gibi verilmektedir.

$$G = \frac{\Delta R/R}{\Delta L/L}$$

Kalibre faktörü (G), metal alaşımli genleşme ölçer için yaklaşık 2 ve yarıiletken genleşme ölçer için ise 100Ω civarındadır. Basınç direkt olarak yukarıdaki eşitliği kullanarak bulmak mümkün olduğu halde, pratikte Şekil 29'da dengeleme köprü devresi kullanılır. Köprünün analog çıkışı, alete herhangi bir baskı uygulamadığında R_x ayarlı direnci kullanılarak sıfırlanır. Alete bir baskı uygulandığında, köprünün dengesi bozulduğundan köprü devresinde bir gerilim oluşur. Bu gerilim, genellikle ayarlanmış değer ile karşılaştırılabilmesi için yükseltilecek ADD girişine geri beslenir. Büyük kalibre faktörü nedeniyle yarı iletken genleşme

ölçerler metal tipleri ile karşılaştırıldığında daha geniş sinyaller üretirler. Buna rağmen sıcaklık değişimlerine karşı daha hassastır.



Şekil 29) Genleşme ölçer köprüsü

VII-) PROGRAM ANA MENÜSÜ

Ana menü

(F1) Online menüsü =

(F1) Program menüsü → a) Blok ve segmentleri oluşturma, silme, kopyalama
b) Bir segmentteki program talimatlarını oluşturma ve değiştirme

-(F1) ARAMA → Bellekteki blok listesindeki bir bloğu belirli bir segment veya adres için araştırır.

→ (F1) → Mevcut bloğa dönülür

→ (F4) → Blok işaretleme penceresini çağırır.

→ (F10) → Belirlenmiş blokları aramaya başlar.

-(Shift F1) Blok arama → Ekrandaki mevcut bloktan diğer bir bloğa geçiş yapmak için kullanılır.

-(F2) SEGMENT OLUŞTURMA → Mevcut blok içine program yazmak ve saklamak için boş bir segmenti çağırır.

-(F10) Segmenti girme:

→ (F1) İlave etme → Mevcut bloğun sonuna segmenti girer

→ (F2) Segment girme penceresinde belirtilen yere yeni segmenti sokar

→ (F3) Yeni segmenti segment girme penceresinde belirtilmiş segmentin yerine yerleştirir.

-(Shift F2) Blok oluşturma → Belirtilmiş tip ve numaralı bloğu oluşturur.

- (F10)_ Bloğu girer
- (F3) YAZMA (İfade listesi) → Belirli bir segmentte ifade listesi ile program talimatlarını yazmaya yarar.
 - (F1) Yardım → Elde edilebilecek tüm ifade listesi talimatlarını içeren yardım menüsün çağırır.
 - (F10)_ Segmenti girme
- (F3) YAZMA (Merdiven diyagramında) → Program talimatlarını merdiven diyagramı ile yazmaya yarar.
 - (F1)_ Normalde açık kontak
 - (Shift F1)_ Taramalar arasında bir değişikliğe göre RLO set veya reset eder.
 - (F2)_ Normalde kapalı kontak
 - (F3)_ Çıkış rölesi
 - (Shift F3) Kıyaslama
 - (F1)_ $A=B$
 - (F2)_ $A \neq B$
 - (F3)_ $A < B$
 - (F4)_ $A \leq B$
 - (F5)_ $A > B$
 - (F6)_ $A \geq B$
 - (F8)_ Math toggle
 - (F9)_ Yatay bir dal oluşturur
 - (F10)_ Bellekteki herhangi bir değişikliği programa girmeyi sağlar
- (F4) TIMER
 - (F1)_ Pulse timer
 - (F2)_ Genişletilmiş pulse timer
 - (F3)_ Gecikmeli timer (Mantık doğru)
 - (F4)_ Gecikme süresi saklı timer
 - (F5)_ Gecikmeli timer (Mantık yanlış)
 - (F6)_ Sayıcıyı durdurur ve mevcut timeri present değerine set eder
 - (F7)_ Belli talimat rölelerini girer
 - (F8)_ Seçilmiş röleleri siler
 - (F10)_ Bellekte oluşan değerleri girer
- (Shift F4) Sayıcı
 - (F1)_ Değeri birer birer arttırarak sayma
 - (F2)_ Değeri birer birer azaltarak sayma
 - (F5)_ Sayıcının ilk değerini set eden talimatı girer
 - (F6)_ Saymayı durdurur ve sayıcıyı ilk değerine set eder

- (F7)_ Seçilen bobini girer
- (F8)_ Seçilen bobini siler
- (F10)_ Bellekteki herhangi bir değişikliği girer
- (F5) Taşı (F5) → Talimatı yükler ve bir yerden bir yere transfer eder
 - Taşı _ (F1) → Veriyi bir yerden başka bir yere nakleder
 - Ani giriş (F3) → I/Q modülündeki bir giriş adresinden belirli bir hedefe veriyi direk olarak taşır
 - Ani çıkış (F4) → I/Q modülündeki bir çıkış adresine veriyi belirli bir yerden direk olarak taşır
 - Bobin yerleştirme (F7) → Seçmeli talimat bobinini girer
 - Bobin silme (F8) → Seçmeli talimat bobinini siler
 - Segmenti gir (F10) → Bellekteki herhangi bir değişikliği girer
- (Shift F5) Dönüştür → Bu talimat veriyi bir formattan diğer birine çevirir
 - (F1) → BCD ⇒ BIN } BCD değeri binary değere çevirir
 - (F2) → BIN ⇒ BCD } Binary değeri BCD değere çevirir
 - (F3) → FLT ⇒ FIX } Fleting (değişen) nokta değerini sabit nokta değerine çevirir
 - (F4) → FIX ⇒ Flt } Sabit bir nokta değerini fleting nokta değerine çevirir.
 - (F5) → Bire tümlleme → Belirli değeri bire tümlleme işlemini yapar.
 - (F6) → İkiye tümlleme → Belirli değeri ikiye tümlleme işlemini yapar.
 - (F7) → Bobine yeleştirme → Seçilen talimat bobinini girer.
 - (F8) → Bobin silme → Seçilen talimat bobinini siler
 - (F10) → Segmenti gir → Bellekteki herhangi bir değişikliği girer.
- (F6) → Logic → Karma verme işini yapar.
 - (F1) → Ve → İki değer için “ve” işlemini yapar.
 - (Shift F1) → Ve değil → İki değer için “ve değil” işlemini yapar.
 - (F2) → Veya → İki değer için “veya” işlemini yapar.
 - (Shift F2) → Veya değil → İki değer için “veya değil” işlemini yapar.
 - (F3) → XOR → İki değer için “XOR” işlemini yapar.
 - (F7) → Bobini gir → Seçilen talimat bobinini girer
 - (F8) → Bobini sil → Seçilen talimat bobinini siler

- ↳ (F10) → Segmenti gir → Bellekteki herhangi deęişiklikleri girer.
- (Shift F6) → Bit işlemleri → Kayıt işlemlerini yapmaya yarar
 - (F1) → Sola koyma → Kayıttaki her bir biti belirli sayıda sola kaydırır.
 - (F2) → Sağa işaretle → Kayıttaki her bir biti en sağdaki bitin haricinde sağa kaydırır
 - (F3) → İşaretsiz sağa → Kayıttaki her bir biti belirli bir sayıda sağa kaydırır
 - (F4) → Sola döndür → Kayıttaki her bir biti belirli bir sayıda sola döndürür
 - (F5) → Sağa döndür → Kayıttaki her bir biti belirli bir sayıda sağa döndürür
 - (F7) → Bobini gir → Seçilen talimat bobinini girer.
 - (F8) → Bobini sil → Seçilen talimat bobinini siler.
 - (F10) → Segmenti gir → Bellekteki herhangi bir deęişikliği girer.
- (F7) → Matematik işlemleri → Dört işlemi yapar
 - (F1) → Ekle
 - (F2) → Çıkar
 - (F3) → Çarp
 - (F4) → Böl
 - (F7) → Bobini gir
 - (F8) → Bobini sil
 - (F9) → Çift kelime işlemcisi kullanıldığında; sabit ve floating nokta arasındaki akış talimatını girer
 - (F10) → Segmenti gir → Bellekteki herhangi bir deęişikliği girer.
- (F8) → Yatay dallama yapar
- (F9) → Dikey dallama yapar
- (Shift F9) → Özel işlemler → Belirli bir bloęa ve bloklararası geçiše müsaade eder
 - (F1) → Atla
 - (F2) → Geçici sonlama
 - (F7) → Bobini gir
 - (F8) → Bobini sil
 - (F10) → Segmenti gir
- (F4) Segment silme → Mevcut bloktaki seçilmiş segment veya grupların silinmesini sağlar
 - (F1) → Teyid etme
 - (F2) → İptal etme
- (Shift F4) Blok silme → Seçilmiş blok veya blok gruplarının silinmesini sağlar

- (F1) → Hafızadaki mevcut bütün organizasyon bloklarını işaretler
- (F2) → Hafızadaki mevcut bütün program bloklarını işaretler
- (F3) → Hafızadaki mevcut bütün fonksiyon bloklarını işaretler
- (F4) → Hafızadaki mevcut bütün veri bloklarını işaretler
- (F5) → Hafızadaki mevcut bütün sıra (sequence) bloklarını işaretler
- (F6) → Hafızadaki mevcut bütün blokları işaretler
- (F7) → Hafızadaki mevcut işaretlenmiş bütün blokları işaretsizleştirir
- (F10) → İşaretlendirilmiş bütün blokları siler
- (F5) → Blokları kopyalar → Seçilen bir bloğun tüm içeriklerini kopyalar
- (F7) → Zum yapma → Eğer program ekranında atlama veya çağırma talimatları bulunuyorsa bu bloklara girmek için bu komut kullanılır
- (F8) → Zumdan çıkma → Eğer zum yapıp içine girilen bloktan çıkmak istiyorsanız, bu komut kullanılır
- F2 → Veri ekranı → Online programlamada; bu komut ile online kaydedici verileri ve I/Q durumlarını kontrol etmek ve verileri işlemci kaydedicisine girmeye izin verir. Offline programda ise; bir disket dosyası içine saklanacak veya daha sonra PLC'ye yüklenecek olan veri tabakalarını oluşturmaya yarar.
- (F1) → Sonraki adres → Aynı adres tipinde bir sonraki adrese yükler. IB000 adresinde iseniz (F1) tuşuna bastığınızda IB001 ekrana eklenir.
- (F2) → Önceki adres → Aynı adres tipinde bir sonraki adrese yükler. Eğer IB008 adresinde iseniz (F1) tuşuna bastığınızda IB007 ekrana eklenir.
- (F3) → Veri yazma → Adresler ekranda gösterildiğinde, bu komutu kullanarak belirli bir veriyi girebilirsiniz. Timer, sayıcı, sistem verisi veya bit adresleri bu komut kullanılarak girilmez.
- (F8) → Cursor'un bulunduğu kolondaki bütün data verilerini netleştirir.
- (F9) → Ekrandaki tüm veri girişlerini netleştirir.
- (F10) → Veri formatı → Veri format menüsünü gösterir.
 - (F1) → Ekrandaki değer formatını işaretlenmiş desimale değiştirir.
 - (F2) → Ekrandaki değer formatını işaretlenmemiş desimale değiştirir.
 - (F3) → Ekrandaki değer formatını hexadesimale değiştirir.

- (F4) → Ekrandaki değerin formatını ASCII gösterimine değıştirir.
- (F5) → Ekrandaki değerin formatını Binary'e değıştirir.
- (F6) → Ekrandaki değerin formatını floting noktaya değıştirir.

F3 → Disk işlemleri: Online ve offline modu gelişmiş disk işlem seçeneklerini uygulayıcıya sunar.

- (F1) → Yükle → Dosyaları disketten belleğe yükler
- (F2) → Sakla → Bu seçeneği kullanarak program bilgileri online ve offline bellekteki belirli bir adrese yüklenilir.
- (F3) → Birleştir → Seçilmiş dosya veya blokları bu seçenek kullanılarak online veya offline bellekteki mevcut yerlerde birleştirebilirsiniz.
 - (F4) → Blokları işaretle
 - (F8) → Online /offline görüntüleme
 - (F10) → Başla → Traffer başlar
- (F4) → Bir programda yapılan değışiklikler bu haliye saklandığı zaman yazılım programı otomatik olarak değışiklerin yapılmadan önceki dosyaları yedek dosya olarak saklar. Eğer ilk programı çağırarak isterseniz (F4)'e basmanız gerekir.
- (F5) → Dosyaları silme → Disk işlemi ekranında gösterilmiş directory'den disk dosyalarını silmek istediğinizde "F5"i kullanınız.
- (F6) → Dosyaları kopyalama → Disk işlemleri ekranında gösterilmiş directory'den disk dosyalarını kopyalamak için "F6"yi kullanınız.
- (F7) → Dosya ismi değıştirme → Disk işlemleri ekranında gösterilmiş directory'den disk dosyalarının ismini değıştirmek için "F7"yi kullanınız.
- (F9) → Sürücü ve directory'ü değıştirme → Disk işlemleri ekranında gösterilmiş mevcut sürücü veya directory'i değıştirmek için disk menüsünden "F9"u seçiniz.
- (F10) → "DIR" tuşu

(F5) → PLC İşlemleri → Online programında bu seçenek PLC ile direk bir iletişim kurar. Böylece program çalışırken programı kontrol edip değışiklik yapılabilir.

- (F1) → (Cold Çalış) → Eğer işlemci bir hata yüzünden durdurlursa, soğuk çalış işlemciyi çalıştırmadan önce kayıtlı verisi netleştirir (açıklar).
- (F2) → (Warm Çalış) → Eğer işlemci bir hata olmaksızın durursa (örneğin bir elektrik kesintisi sebebiyle) kayıtlı

veri sıcak çalışma ile saklanır ve kesinti noktasında işlem kaldığı yerden devam eder.

→ (F3) → DUR → İşlemciyi durdurur.

→ (F4) → HATA ARAMA VE DÜZELTME → Online modunda programda iken; PLC işlemleri menüsünden "F4" seçilirse, program ve PLC durumları hakkında detaylı bilgiler alınabilir.

→ (F1) → Kontrol bitleri → İster işlemci çalışıyor isterse bir hata nedeniyle durmuş olsun "F1" seçilerek işlemcinin kontrol bitlerinin durumları gözlenir.

→ (F2) → Kesintiye listeleme → Bir hata durumunda program durumu hakkındaki bilgiler Istack ismi altında PLC'nin işlemci belleğinde muhafaza edilir ve bu bilgi "F2" ile çağrılır.

→ (F1) → Yapılan hata sayısı fazla ve bu hata sayısı toplam hata sayısından daha az ise bu tuş hata serisindeki bir sonraki hata bilgisini ekrana getirir.

→ (F2) → Hata sayısı 1'den daha büyük ise "F2" tuşu hata serisinde bir önceki hata bilgisini ekrana getirir.

→ (F3) → BİT SONUÇLARI → Hata olduğu zaman programın durumu hakkındaki bilgileri ekrana getirir.

→ (F4) → HATA BİTLERİ → İşlemcinin durduğu noktaya kadar olan hataların tümü hakkındaki bilgiyi ekrana getirir.

→ (F5) → GİZLİ BİTLER → Bitlerin durumlarına göre sistemi korumak veya durdurmaya müsaade eden saklı bit durumları hakkındaki bilgiyi ekrana getirir.

→ (F10) → HATA ATLAMASI → Hata bir program talimatı sebebiyle oluştuysa. bu seçenek

segmen yazıcısını çağırır ve segmenti ile bu hata sırasını ekranda gösterir.

→ (F3) → Hata olduğunda, programın blok yapısı hakkındaki bilgi PLC'nin işlemci belleğinde tutulur ve bu seçenekle bu bilgiler ekrana çağrılır.

→ (F1) → Bir sonraki sayfa çağrılır.

→ (F2) → Bir önceki sayfa çağrılır.

(F6) → Cross-Reference → Bu seçenek ya bütün program ya da seçilmiş bloklarda, blok ve adreslerin kullanılış listesini hazırlar.

→ (F1) → İçinde adres, blok ve segmentlerin bulunduğu listeyi ekrana çağırır.

→ (F4) → Blokları işaretle → Belirli blokları seçmek için bu tuşu kullanın.

→ (F5) → BIT → Bu seçenek bit adreslerinin tip ve sıralarını seçmek için kullanılır.

→ (F6) → BYTE → Bu seçenek byte adreslerini, tip ve sıralarını seçmek için kullanılır.

→ (F7) → KELİME → Bu seçenek kelime adreslerinin tip ve sıralarını seçmek için kullanılır.

→ (F8) → ÇİFT KELİME → Bu seçenek çift kelime adreslerinin tip ve sıralarını seçmek için kullanılır.

→ (F2) → Cross-referans için seçilmiş her bir bloğu ekrana getirir.

→ (F1) → Belirlenmiş blokların her biri listelenmiş şekilde ekrana getirilir.

→ (F2) → Belirlenmiş blokların her biri bu belirlenmiş bloğu çağırarak için programlanmış blokların bir listesini yapar.

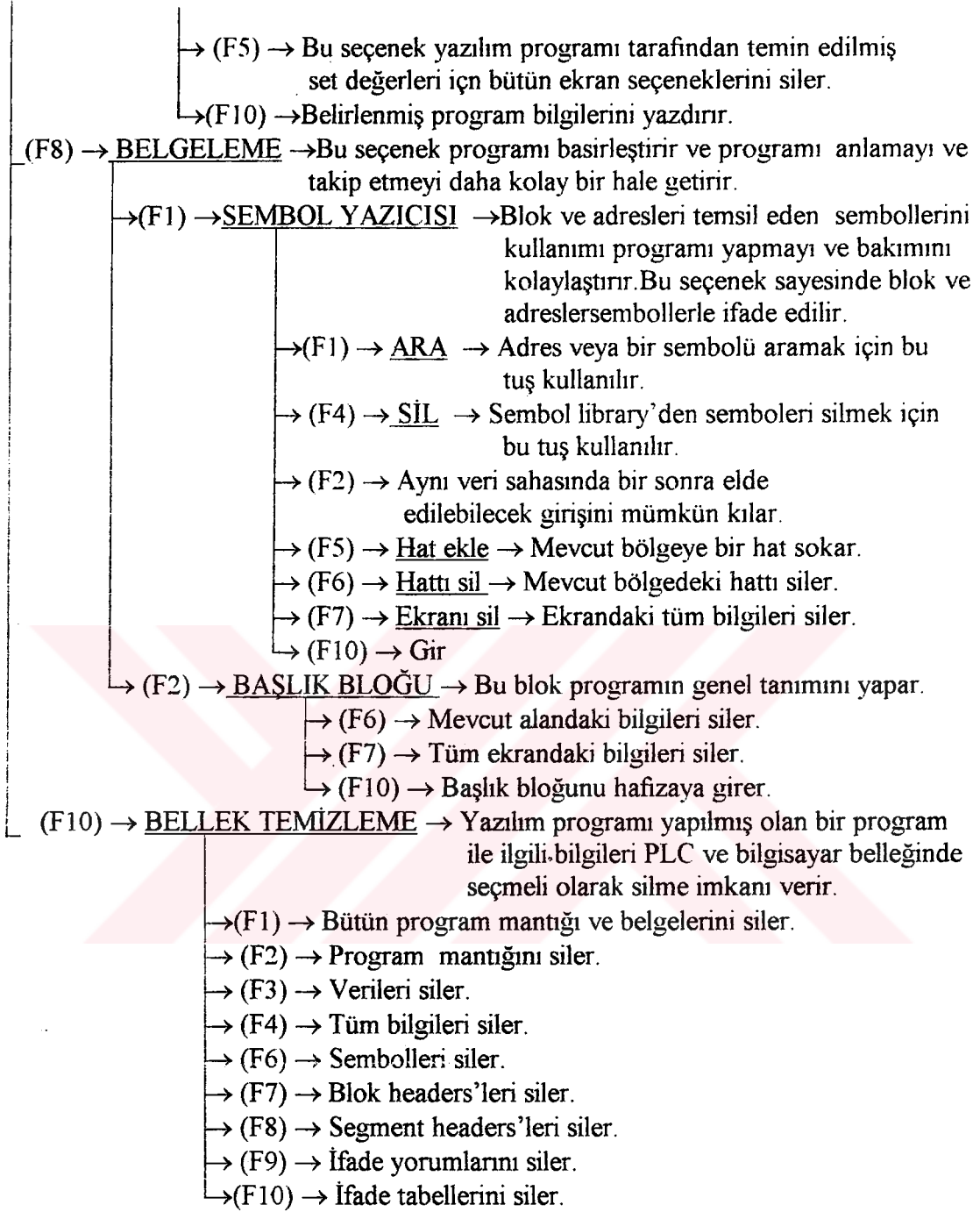
→ (F3) → Bu seçenek programda bulunan tüm blokların bir ağaç diyagramını oluşturur.

→ (F4) → Bu seçenek cross-referans da kullanılacak belirli blokları seçmek için kullanılır.

(F7) → YAZIM PROGRAMI → Program ve dökümanlarınızı yazdırmak için bu tuşu kullanın.

→ (F1) → Pencere seçimi → Bu seçenekle yazım programına girdiğinizde ekranda gözüken 3 adet pencereden birini seçebilirsiniz.

→ (F4) → Blokları işaretle → Yazım işleminde kullanılacak belirlenmiş blokları işaretler ve yazıma hazır hale getirir.



VIII- KOMUTLAR

STEP-5 programı dili aşağıda belirtilmiş 3 işlem tipine sahiptir.

VIII-1- Basit işlemler: Organizasyon, program, sequence ve fonksiyon bloklarında işlem görebilen fonksiyonlardır.

VIII-2- İlave işlemler: Yer değiştirme, test ve değişim işlemleri gibi kompleks fonksiyonlardır ve yalnızca STL ile programlanabilir.

VIII-3- Sistem işlemleri: İşletim sistemine direk olarak bağlanmaya yararlar ve yalnızca STL ile programlanabilirler.

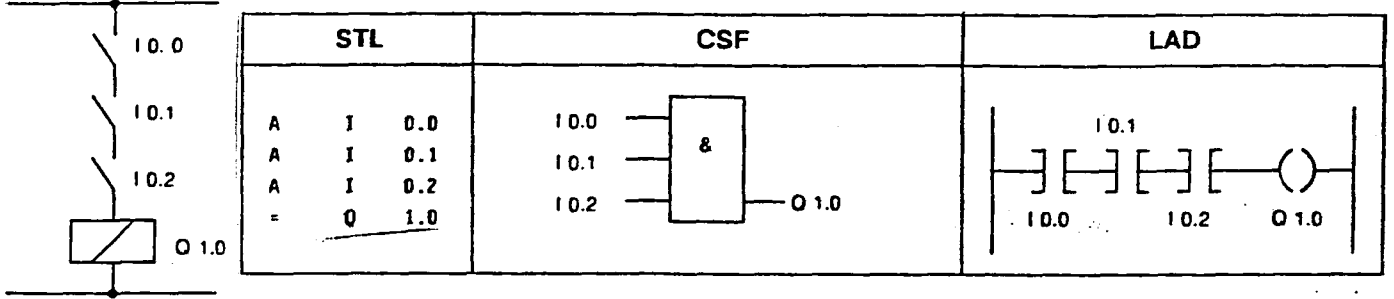
VIII-1-) BASİT İŞLEMLER

VIII-1-1-) BOOLEAN LOJİK İŞLEMLERİ

İşlem	İşlemci	Açıklama
0		<u>Logic OR vasıtasıyla AND işlemlerini birleştirir:</u> Logic OR vasıtasıyla bir önceki RLO ile bir sonraki AND Logic işlemi (RLO) nin sonucunu birleştirir.
A(<u>Logic AND vasıtasıyla parantez içindeki ifadeyi birleştirir:</u> Logic AND vasıtasıyla bir önceki RLO ile parantez içindeki ifadenin RLO'sunu birleştirir.
O(<u>Logic OR vasıtasıyla parantez içindeki ifadeyi birleştirir:</u> Logic OR vasıtasıyla bir önceki RLO ile parantez içindeki ifadenin RLO'sunu birleştirir.
)		<u>Parantez kapama:</u> Parantez içindeki ifadeyi bu işaretle bitirir.
A	<input type="checkbox"/>	<u>"1" ve logic AND vasıtasıyla RLO'lu birleşme için tarama işlemcisi:</u> işlemci işaret durumu "1" ise sonuç "1"dir. Bu sonuç logic "OR" vasıtasıyla işlemcideki RLO ile birleştirilir.
0	<input type="checkbox"/>	<u>"1" ve logic OR vasıtasıyla RLO'lu birleşme için tarama işlemcisi:</u> işlemci işaret durumu 1 ise sonuç "1"dir.
AN	<input type="checkbox"/>	<u>"0" ve logic AND vasıtasıyla RLO'lu birleşme için tarama işlemcisi:</u> işlemci işaret durumu 0 ise sonuç "0"dir.
ON	<input type="checkbox"/>	<u>"0" ve logic OR vasıtasıyla RLO'lu birleşme için tarama işlemcisi:</u> işlemci işaret durumu 0 ise sonuç "1"dir.
<u>ID</u>	<u>PARAMETRE</u>	<u>CPU 100</u>
I		0.0 127,7
Q		0.0 127,7
F		0.0 127,7
T		0.0 15
C		0.0 15

VIII-1-2-) AND İŞLEMİ:

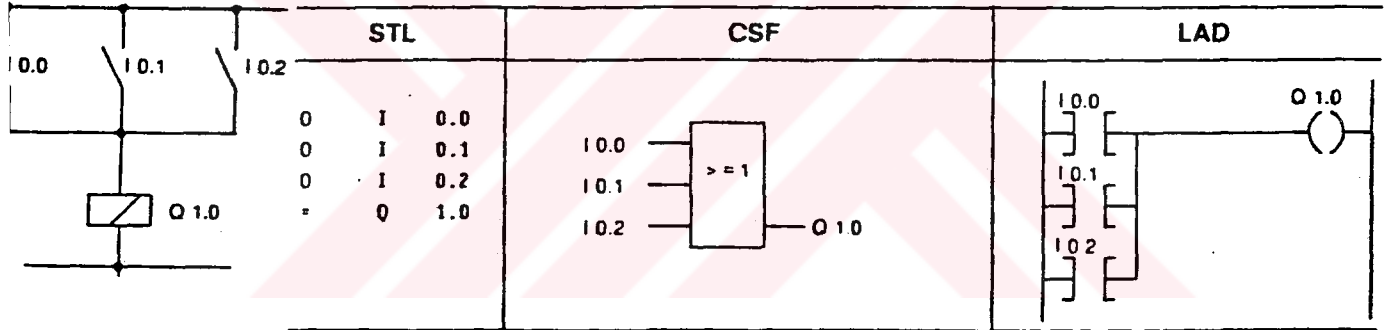
AND işlemi farklı şartların olup olmadığını görmek için tarama yapar.



Yukarıdaki devrede 3 girişte "1" olduğu zaman Q1.0 çıkışı "1" olur. Eğer girişlerden yalnızca bir tanesi bile "0" olsa, Q1.0 çıkışı "0" olur.

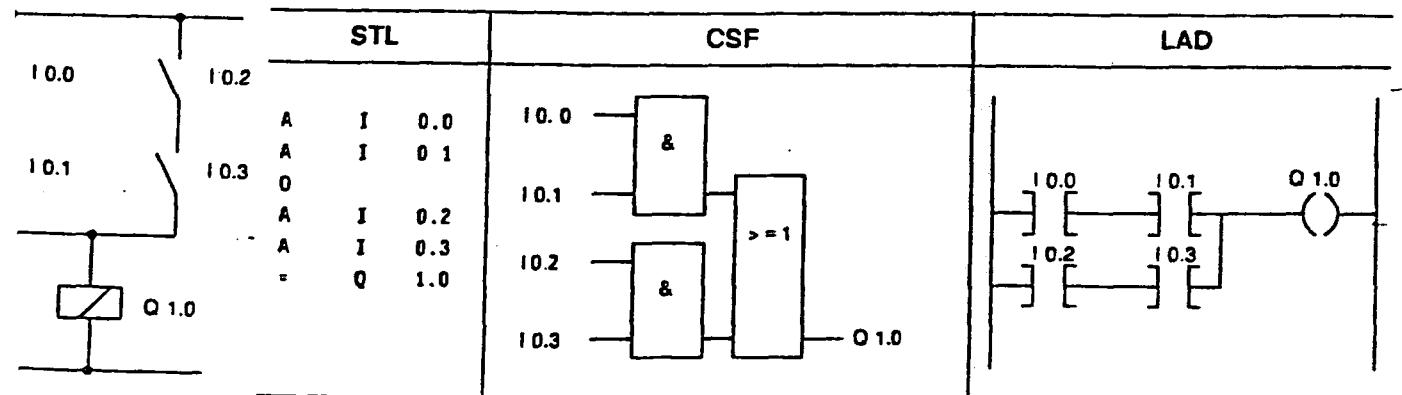
VIII-1-3-) OR İŞLEMİ

OR işlemi 2 veya daha şartlardan birin olup olmadığını görmek için tarama yapar.



Yukarıdaki devrede girişlerden en az biri dahi "1" olsa Q1.0 çıkışının "0" olması için tüm girişlerin "0" olması gerekir.

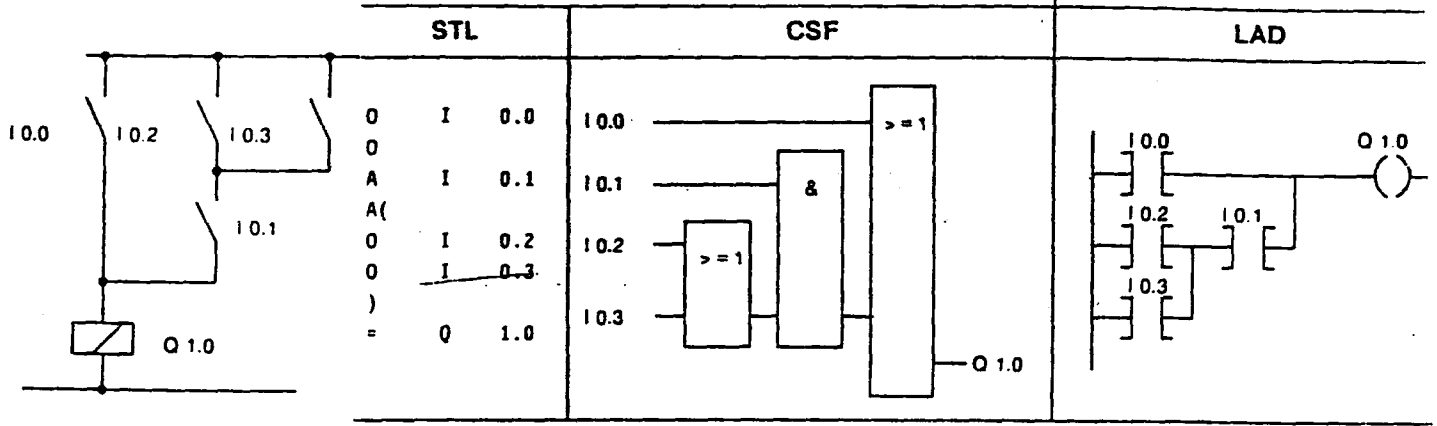
VIII-1-4-) OR İşleminde Önce AND İşlemi Olması Hali



Q1.0 çıkışının "1" olması için AND şartlarından en az birinin sağlanması gereklidir. Her iki AND şartı da sağlanmıyorsa Q1.0 çıkışı "0" olur.

VIII-1-5-) AND İşleminin Önce OR Olması Hali

Örnek 1-



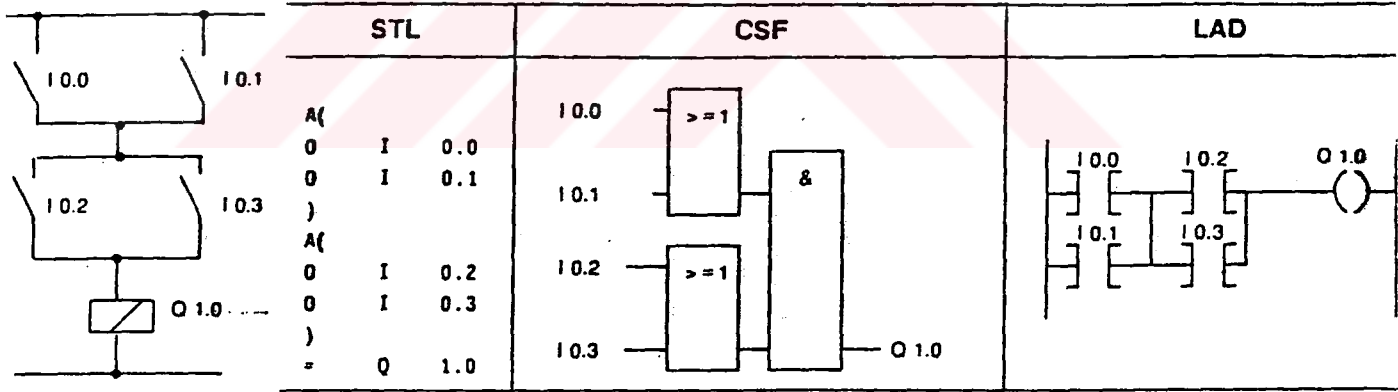
Aşağıdaki şartların biri yerine getirildiğinde Q1.0 çıkışı "1" olur.

- I 0.0 "1" ise;

- I 0.1 veya I 0.2 ya da I 0.3 = "1" ise;

AND şartlarından hiçbiri yerine getirilmezse Q 1.0 çıkışı "0" olur.

Örnek 2-



Her iki "OR" şartı da yerine getirilirse Q1.0 "1" olur.

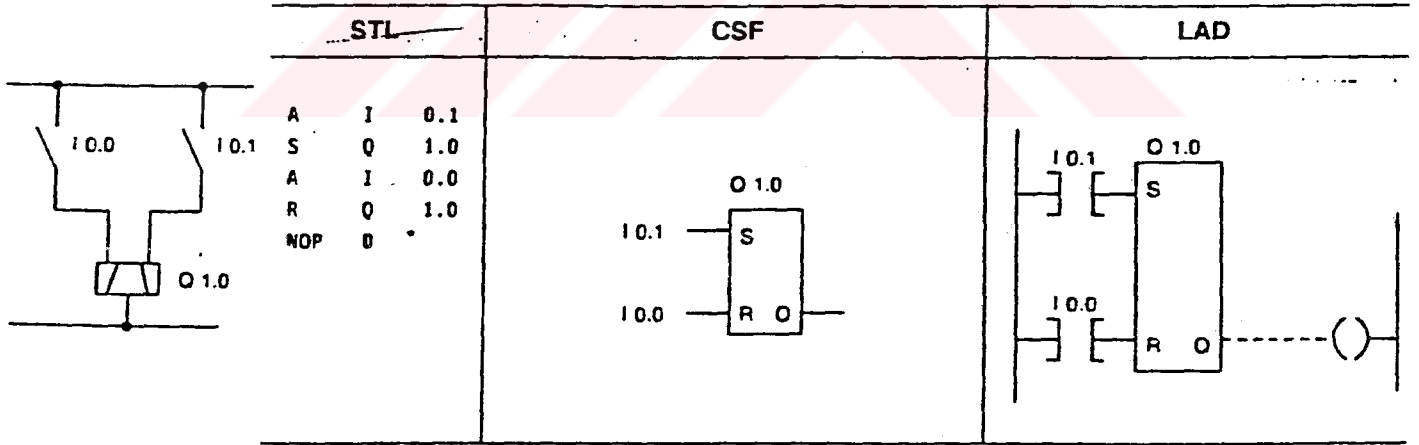
OR şartlarından en az 1'i yerine getirilmezse çıkış "0" olur.

VIII-1-6-) SET VE RESET İŞLEMLERİ

Set/reset işlemleri; işlemcide şekillenmiş RLO'yu saklar. Saklanmış RLO adreslenmiş işlemcinin işaret durumunu temsil eder. Saklama dinamik veya statik olabilir.

İŞLEM	İŞLEMCİ	AÇIKLAMA
S		Set: Program RLO=1 ile tarandığı zaman; işaret durumu adreslenmiş işlemciye "1" olarak aktarılır. Bu durumda RLO bu durumdan etkilenmez.
R		Reset: Program RLO=1 ile tarandığı zaman; işaret durumu adreslenmiş işlemciye "0" olarak aktarılır. Bu durumda RLO bu durumdan etkilenmez.
=		İç Programın her tarama sırasında adreslenmiş işlemciye RLO'nun mevcut değeri aktarılır.
<u>ID</u> I Q F	<u>Parametre</u>	CPU 100 0.00-127.7 0.00- 127.7 0.00- 127.7

Örnek 1) Flip- Flop (Reset baskın) çıkış sinyali için



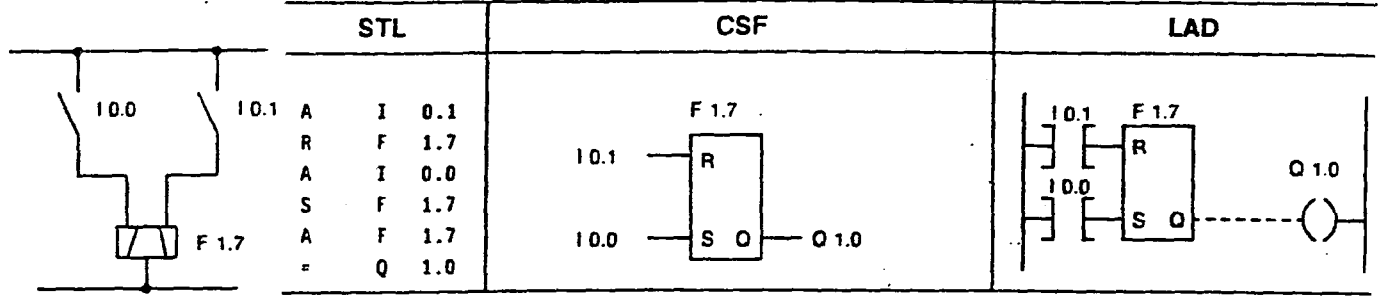
-I 0.1 girişine "1" değeri gelince flip-flop'ın Q 1.0 set edilir. (Q 1.0 "1" olur). Eğer I 0.1 girişindeki işaret "0" ile değişirse; Q 1.0 durumu değişmez.

- I 0.0 girişine değeri gelince flip-flop'ın Q 1.0 reset edilir. (Q 1.0 "0" olur).

- Eğer SET (I 1.0 girişi) ve RESET (I 0.0 girişi) aynı anda uygulanırsa; son olarak programlanmış tarama işlemi (Bu durumda; bu işlem A I 0.0'dır) . Programın kalan işlem süresinde bu etkinin altındadır.

- Bu programda reset edilen Q 1.0 çıkışı etkindir.

Örnek 2- Flaglar ile 8S Flip-Flop (Set baskındır)



- I0.0 girişine "1" işareti geldiğinde flip-flop F1.7 set edilir. (İşaret durumu "1")Eğer I0.0'daki işaret "0" a değişirse F1.7 flag'ının işaret durumu sabit kalır.

-I0.1 girişine bir işareti geldiğinde flip-flop F1.7 reset edilir. (İşaret durumu "0")Eğer I0.1'deki işaret "0" a değişirse F1.7 flag'ının işaret durumu değişmez.

-Eğer her iki giriş de "1" değer alırsa, flip-flop set edilir.(set baskındır.)

-Flag'ın işaret durumu her periyotta taranır ve Q1.0'a transfer edilir.

VIII-1-7-YÜKLEME VE TRANSFER İŞLEMLERİ

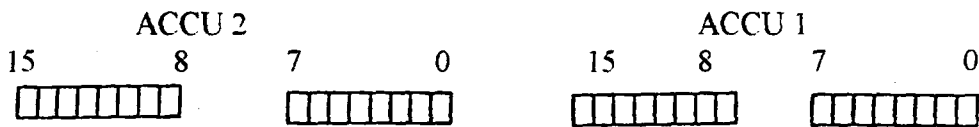
Aşağıdaki işlemler için yükleme ve transfer işlemleri kullanınız.

*Değişik işlem alanları arasında bilgi değişimi.

*İleri işlemler için zaman ve sayma değerlerini hatırlama.

*Program işlemi için sabitleri yükleme.

Bilgi akümülatör (ACCU1 ve ACCU2) üzerinden direkt olarak iletilir.Akümülatörler geçici bilgi saklama işini yapan özel kayıt bölgeleridir.Herbiri 16 bit uzunluğundadır.

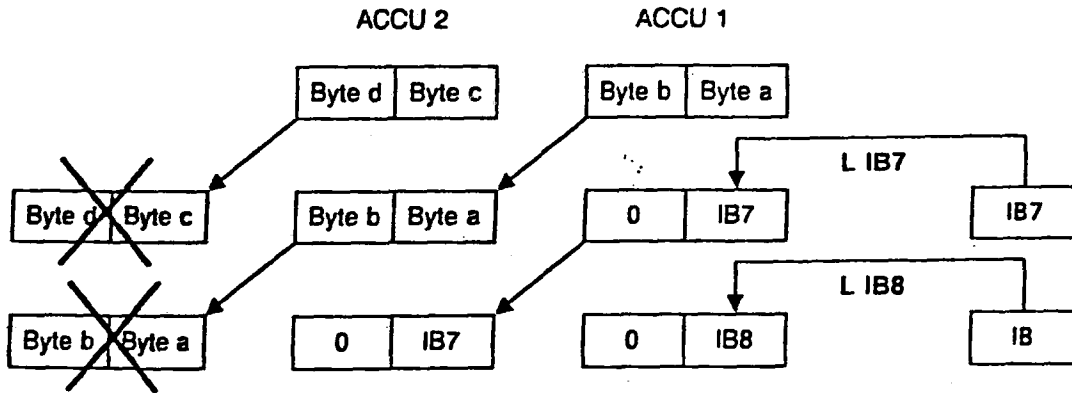


İŞLEM	İŞLEMCİ	AÇIKLAMA	
L	<input type="checkbox"/>	<input type="checkbox"/>	<u>Yükle:</u> RLO'ya bakmaksızın işlemci içerikleri ACCU1 içine kopya edilir. RLO ise bundan etkilenmez.
T	<input type="checkbox"/>	<input type="checkbox"/>	<u>Transfer et:</u> RLO'ya bakmaksızın ACCU1'deki içerik bir işlemciye transfer edilir. RLO ise bundan etkilenmez.
ID			CPU 100 IB 0 127 IW 0 127 0 127
LD	<input type="checkbox"/>	<input type="checkbox"/>	<u>BCD olarak yükleme:</u> RLO'ya bakmaksızın Binary'ye olan zamanları ve sayıları (timer ve counter) ACCU1'e BCD kodunda yükler.
ID	Tc		Parametre CPU 100 0 -15 0 -15

YÜKLEME İŞLEMİ:

Yükleme süresince bilgi bir bellek alanından (örneğin PII'dan) ACCU 1'e kopyalanır. ACCU 1'in ilk içeriği ACCU 2'te kaydırılır. ACCU 2'nin orijinal içeriği ise kaybolur.

Örneğin: Ardarda gelen iki byte (IB7 ve IB8) PII'dan akümülatöre yüklenir. Yükleme ise PII'yı değiştirmez.

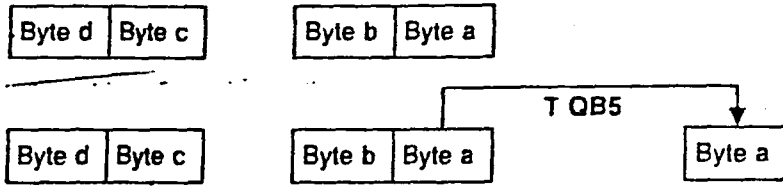


Transfer İşlemi :

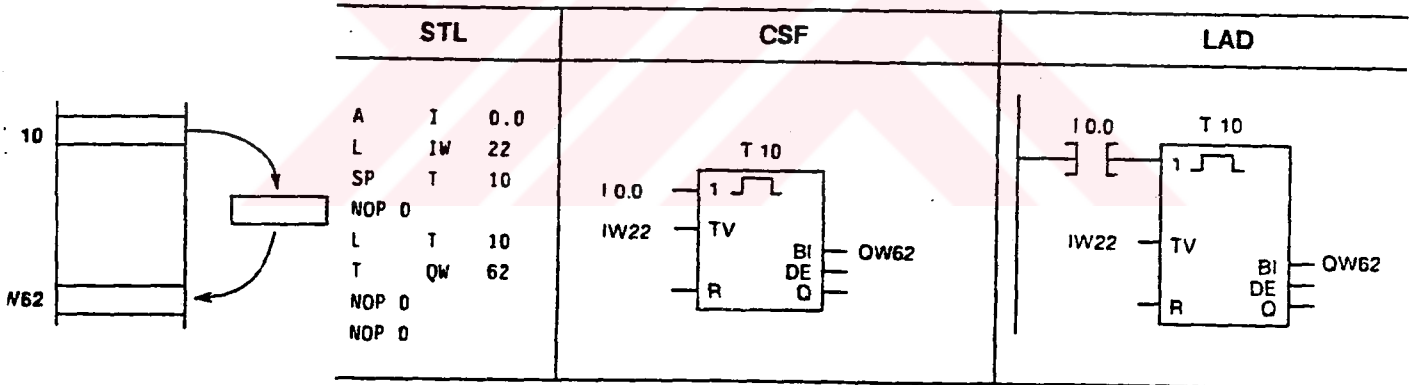
Transfer süresince, transfer edilecek bilgi ACCU 1`den adreslenmiş bellek bölgesine (örneğin: PI Q) kopyalanır. Bu transfer ACCU 1`in içeriğini etkilemez.

ACCU 2

ACCU 1



Örnek: Bir zamanı yükleme ve transfer etme.

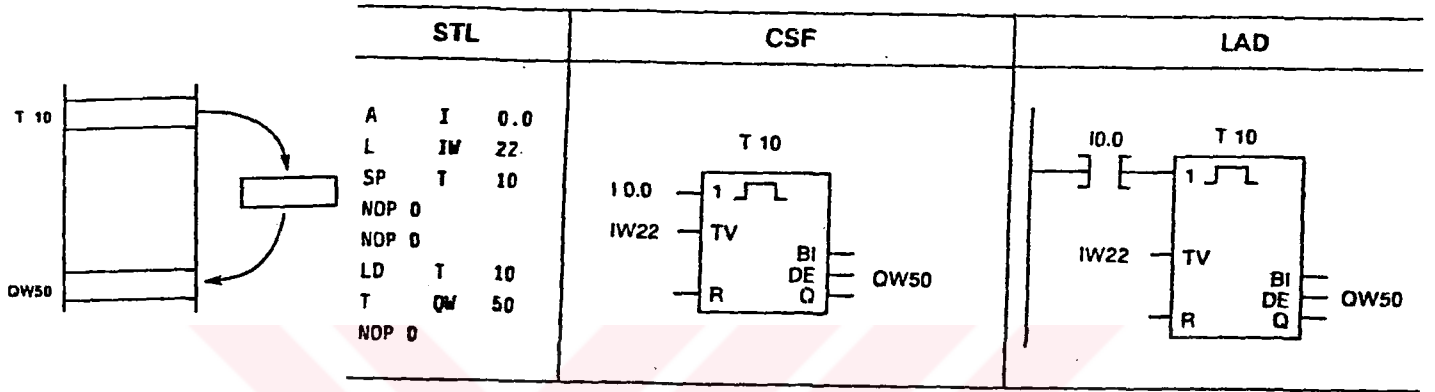


Grafik giriş süresince QW 62 zamanlayıcının BI çıkışına uygulanır. Programlayıcı otomatik olarak kontrol programındaki uygun yükleme ve transfer işlemini depolar. Böylece T 10 ile adreslenmiş bellek bölgesinin içeriği ACCU1'e yüklenir. Daha sonra; akümülatörün içeriği

QW 62 ile adreslenmiş prosöre transfer edilir. Bu örnekte binary kodunda T 10 timer'ini görebilirsiniz.

BI ve DE digital çıkışlardır. BI çıkışındaki zaman binary kodundadır, DE çıkışındaki zaman ise BCD kodundadır.

Örnek : Kodlanmış bir zamanı yükleme ve transfer etme.



T 10 ile adreslenmiş bellek içeriği BCD kodunda akümülatöre yüklenir. Sonra bir transfer işlemi akümülatör içeriğini QW 50 tarafından adreslenmiş proces image bellek bölgesine transfer eder. Bir kodlama işlemi bir timer veya sayıcı bölgesinin DE çıkışına bir adres vererek LAD ve CSF grafik gösteriliş şeklinde indirek olarak temsil etmek mümkündür. Bununla beraber; bu işlem STL ile ayrı bir ifade (statement) ile gösterilir.

VIII-1-8-)TIMER İŞLEMLERİ :

İŞLEM	İŞLEMCİ	AÇIKLAMA
SP	-	<u>Pulse timer:</u> RLO'nun yükselen kenarında timer çalıştırılır. RLO "0" olduğu zaman timer "0" a set edilir. Tarama işaret durumu "1" olduğu zaman sonuçlanır.
		<u>Genişletilmiş pulse timer:</u>

SE	□	□	Timer RLO'nun yükselen kenarında çalıştırılır. RLO'nun "0" olmasından timer etkilenmez. İşaret durumu "1"de kalır.
SD	□	□	<u>On-delay timer:</u> RLO'nun yükselen kenarında çalıştırılır. RLO'nun "0" olduğu zaman timer "0"a set edilir. Timer süresi bitince ve RLO hala girişte ise işaret durumu "1" olduğu zaman tarama sonuçlanır.
SS	□	□	<u>Depolanmış on-delay timer:</u> Timer RLO'nun yükselen kenarında çalıştırılır. RLO'nun "0" olmasında timer etkilenmez. Timer'in süresi bittiği zaman işaret durumu "1" durumunda tarama sonuçlanır. "R" işlemi ile timer resetlendiği zaman işaret durumu "0" olur.
SF	□	□	<u>Off-delay timer:</u> Timer RLO'nun düşen kenarında çalıştırılır. RLO "1" olduğu zaman, timer ilk değerine set edilir. Girişteki RLO "1" veya timer çalıştığı sürece işaret durumu "1"de tarama sonuçlanır.
R	□	□	<u>Reset timer:</u> Timer RLO "1" olduğu sürece ilk değerine resetlenir. RLO "0" olduğunda, timer bundan etkilenmez. Timer resetlendiği veya henüz çalıştırılmadığı sürece tarama işaret durumu "0"da sonuçlanır.
ID			Parametre CPU 100
T			0-15

Bir Zamanı Yükleme :

Timer işlemleri iç timer'leri çağırır.

Bir timer işlemi başlatıldığı zaman, ACCU1'deki kelime bir zaman değeri olarak kullanılır. Bu yüzden ilk olarak akümülatördeki zaman değerleri belirtilmelidir.

Bir timer'i aşağıdaki data tiplerin herhangi biri ile yükleyebilirsiniz.

KT sabit zaman değeri

veya

DW veri kelimesi

IW giriş kelimesi

QW çıkış kelimesi

FW flag kelimesi

Bu veri tipleri BCD kodunda olmalıdır.

Sabit Bir Zaman Değerini Yükleme :

Aşağıdaki örnek 40 s sürekli bir zamanı nasıl yüklenebileceğini gösterir.

L KT 40.2

Zaman Bazı Tablosu :

Baz	0	1	2	3
Çarpan	0.01 s	0.1s	1s	10s

Örnek : KT 40.2 40x1 s 'ye tekabül eder.

Telörans : Zaman teloransı bazına eşdeğerdir.

Örnek	İşlemci	Zaman Aralığı
4Ds'lik bir	KT 400.1	400*0.1s -39.9s ↔ 40s
Zamanı set etmek	KT 40.2	40*1s -39s ↔ 40s
mümkündür	KT 4.3	4*10s -30s ↔ 40s

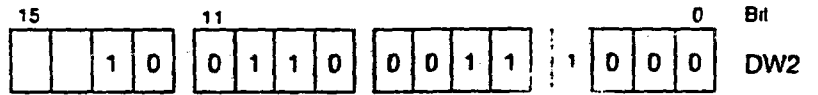
Not : Daima en küçük zaman bazını kullanınız.

Zamanı Bir Giriş,Cıkış,Flag veya Veri Kelimesi Olarak Yükleme :

Yükleme ifadesi ⇒ L DW 2

638 s'lik bir zaman BCD kodunda DW 2 veri kelimesi olarak depolanmıştır. Bu zaman değeri için 14 ve 15 byte'lar gereksizdir.

Zaman Bazı Tablosu :



Baz	0.0	0.1	10	11
Çarpan	0.01s	0.1s	1s	10s

DW 2 veri kelimesini yazmak için kontrol programını da kullanabilirsiniz.

Örnek : DB 3 veri bloğunun veri kelimesi DW 2'ye 270x100ms'lik değeri yüklenir.

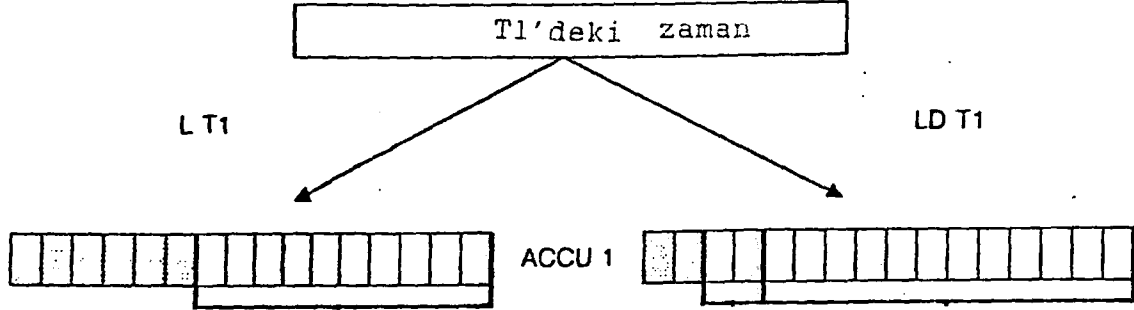
C DB 3

L KT 270.1

T DW 2

Mevcut Zaman.1'in Cıkışı :

ACCU 1'in içine mevcut zamanı yerleřtirmek için bir yük işlemi kullanabilirsiniz.



Bir Timer'i Çalıştırma :

Programlanabilir kontrolörlerde , timer programı taramasını asenkron olarak yapar. Set edilmiş zaman,program tarama periyodu süresince bitirilebilir (durdurulabilir). Bu gelecek zaman taraması tarafından değerlendirilir. En kötü durumda; tam bir program tarama periyodu bu değerlendirmeden önce geçilebilir. Sonuç olarak, timer kendi kendine aktif olmamalıdır.

Örnek :

Şematik Gösteriliş	Açıklama
<p>Program</p> <p>T17'nin işareti</p> <p>0 1</p> <p>L KT 100.0</p> <p>SP T 17</p> <p>A T 17</p> <p>= Q 1.0</p> <p>1s - n · tp</p>	<p>T17* Timer'i çalıştırılması sebebiyle şema olarak "n" + 1" işlem periyodunu gösterir. "= Q 1.0 ifadesinden hemen sonra timer süresi bitmesine rağmen Q 1.0'a çıkışı şart edilmiş vaziyette kalır. Bir sonraki program tarama periyodu gelene kadar değişim farkedilemez.</p> <p>* KT100.0 1 saniyeye eşittir.</p>

Aşağıdaki kurallar timer'lara uygulanır.

* Reset timer'i hariç; bir kenar deęişimi olduęu zaman bütün timer işlemleri yalnızca başlatılır (start edilir). R10 "0" ve "1" arasında deęişir.

* Çalışma başladıktan sonra; yüklenmiş zaman "0" 'a varıncaya kadar zaman bazına azaltılır.

* Timer çalışırken eđer kenar deęişimi var ise; timer ilk deęerine resetlenir ve yeniden start edilir.

* Timer'in işaret durumu Boolean mantık işlemleriyle taranabilir.

VIII-1-8-1)PULSE; TIMER :

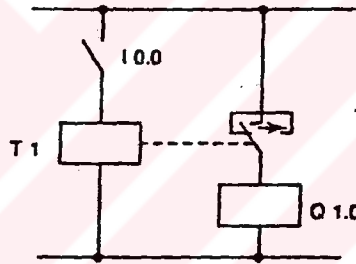
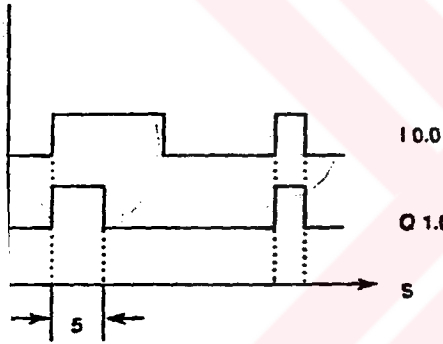
Örnek: I0.0 girişindeki işaret durumu 0'dan 1'e deęiştiiği zaman Q 1.0 çıkışı set edilir.

Bununla birlikte çıkış 5s'den daha uzun set olarak kalmamalıdır.

Zaman diyagramı

Devre diyagramı

STL

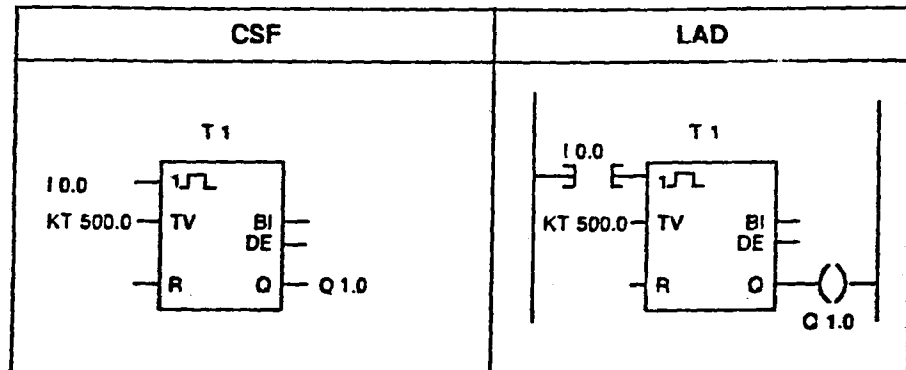


```

A      I      0.0
L      KT     500.0
SP     T      1
NOP    0
NOP    0
NOP    0
A      T      1
=      Q      1.0

```

TI: NO kontaklı zaman rölesi



NOT : Zaman telöransı zaman bazına eşittir. Mümkünse daima en küçük zaman bazını kullanınız.

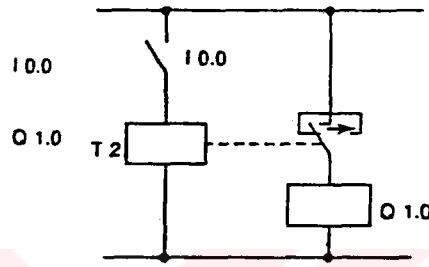
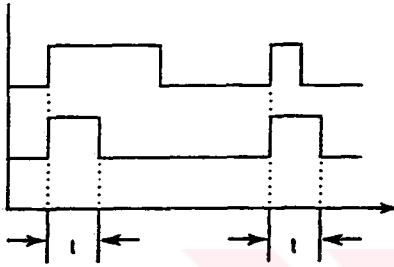
VIII 8-1-2-)Genişletilmiş Pulse Timer:

Örnek : I 0.0 girişindeki işaret 0'dan 1'e değiştiği zaman Q 1.0 çıkışı belirlibir zaman için set edilir. Zaman IW 16'da gösterilir.

Zaman diyagramı

Devre diyagramı

STL

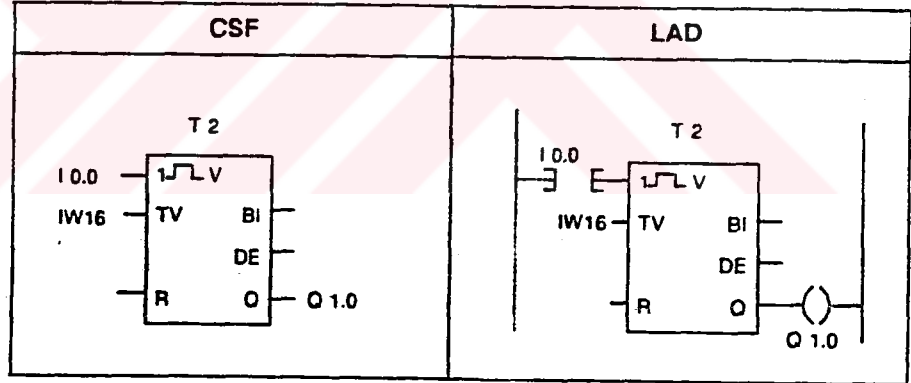


```

A      I      0.0
L      IW     16
SE     T      2
NOP    0
NOP    0
NOP    0
A      T      2
=      Q      1.0

```

T2: Pulse şekilli zaman rölesi



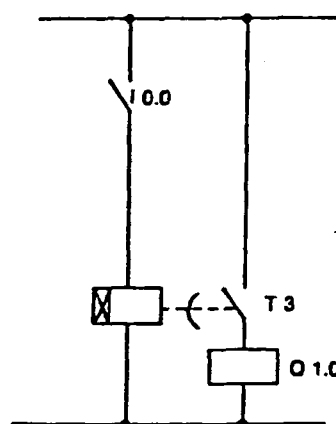
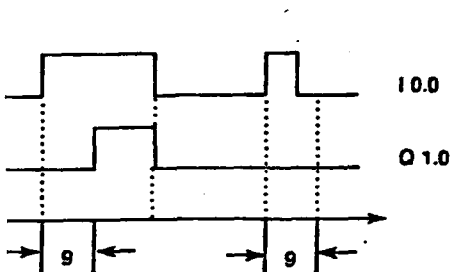
VII-1-8-3-)ON - DELAY TIMER :

Örnek : Q 1.0 çıkışı I 0.0 girişi "1" olduktan itibaren 9s' ye set edilir ve girişteki işaret "1" değerini koruduğu sürece set edilmiş olarak kalır.

Zamanlama diyagramı

Devre diyagramı

STL

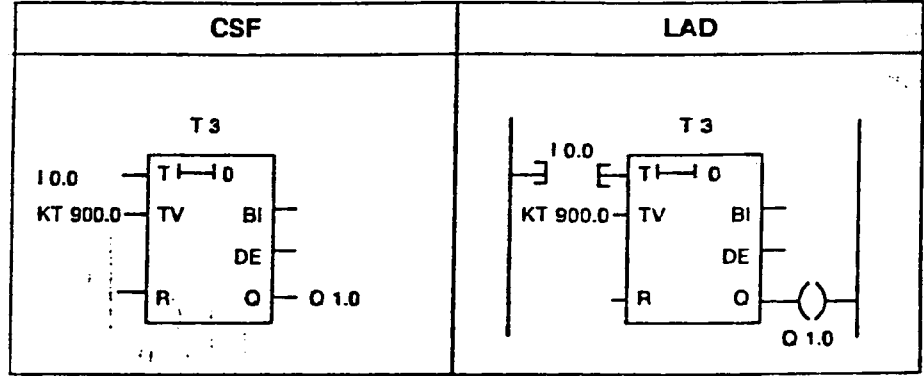


```

A      I      0.0
L      KT     900
SD     T      3
NOP    0
NOP    0

```

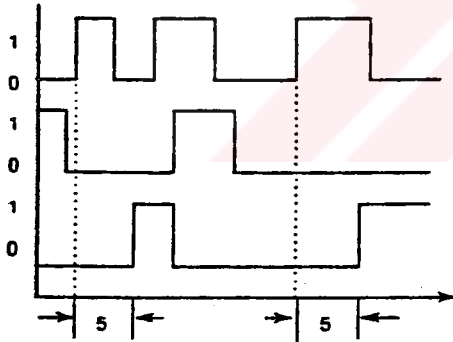
NOP 0
A T 3
= Q 1.0



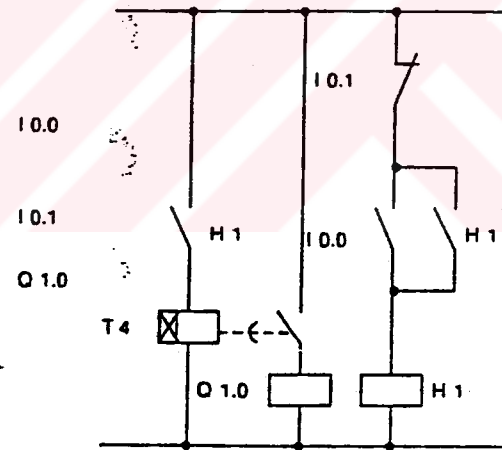
VIII-1-8-4-)DEPOLANMIŞ ON - DELAY VE RESET TIMER :

I 0.0 "1" olduktan 5s sonra Q 1.0 çıkışı set edilir. I 0.0 girişindeki işaret durumundaki değişiklikler çıkışı etkilemez. I 0.1 girişi TU timri'ni ilk dğerine resetler ve Q1.0 çıkışını 0'a set eder.

Zaman diyagramı



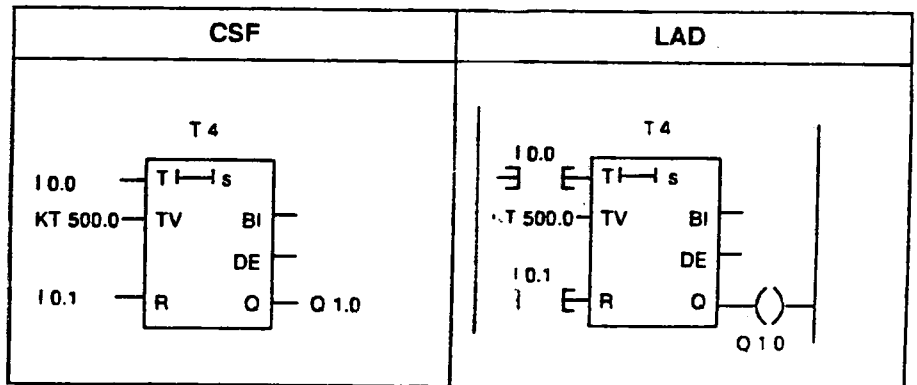
Devre diyagramı



H1: Yardımcı röle

STL

A I 0.0
L KT 500.0
SS T 4
A T 0.1
R T 4
NOP 0
NOP 0

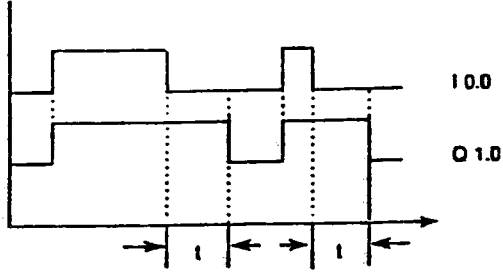


A T 4
 = Q 1.0

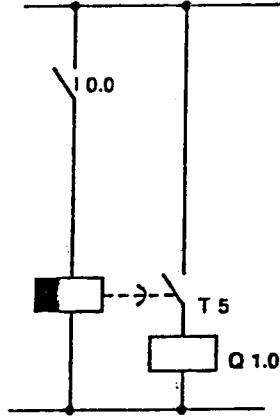
VIII-1-8-5)OFF - DELAY TIMER :

Örnek : I 0.0 girişi resetlendiği zaman belirli bir "t" gecikmesinden sonra Q 1.0 "0"la set edilir.FW 14'deki değer gecikme zamanı belirtir.

Zamanlama diyagramı

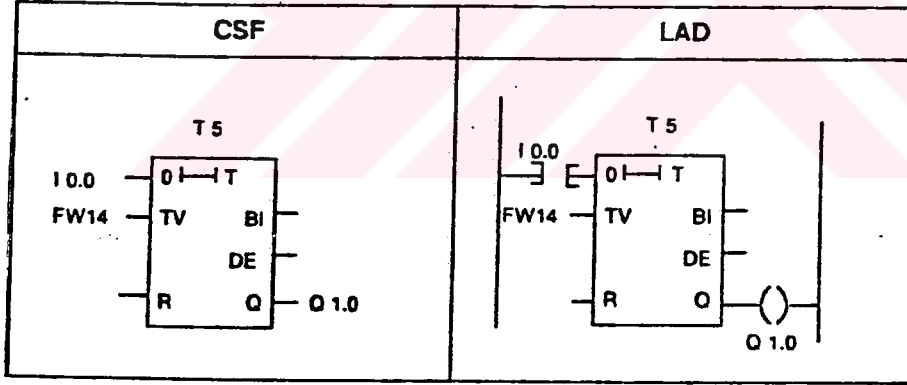


Devre diyagramı



STL

A	I	0.0
L	FW	14
SF	T	5
NOP	0	
NOP	0	
NOP	0	
A	T	5
=	Q	1.0



VIII-1-9-) SAYICI İŞLEMLERİ :

Programlanabilir kontrolörler sayma işlemini yapmak için satma işlemlerini kullanırlar. Sayıcılar yukarı veya aşağı sayabilirler. Sayma aralığı 0'dan 999'a kadardır.

İŞLEM	İŞLEMCİ		AÇIKLAMA
S	<input type="checkbox"/>	<input type="checkbox"/>	<u>Set sayıcısı:</u> Sayıcı RLO'nun yükselen kenarında set edilir.
R	<input type="checkbox"/>	<input type="checkbox"/>	<u>Reset sayıcısı:</u> Sayıcı RLO "1" olduğu sürece "0" a reset edilir.
CU	<input type="checkbox"/>	<input type="checkbox"/>	<u>Yukarı sayma:</u> Sayma RLO'nun her yükselen kenarında "1" artırılır. RLO "0" olduğu zaman sayma etkilenmez..
CD	<input type="checkbox"/>	<input type="checkbox"/>	<u>Aşağı sayma:</u> Sayma RLO'nun her yükselen kenarında "1" azaltılır. RLO "0" olduğu zaman sayma etkilenmez..
<u>ID</u> C			<u>Parametre</u> CPU 100 0-15

Bir Sayı Yükleme :

Sayı işlemleri iç sayıcıları çağırır. Bir sayıcı set edildiği zaman ACCU1'deki kelime bir sayı olarak kullanır. Bu yüzden ilk önce akümülatördeki sayıları depolamalısınız.

Bir sayıyı aşağıdaki ve tiplerinden herhangi biriyle yükleyebilirsiniz.

KC sabit sayı

DW veri kelimesi

IW giriş kelimesi

QW çıkış kelimesi

FW flag kelimesi

Bu kelimeler için veri BCD kodunda olmalıdır.

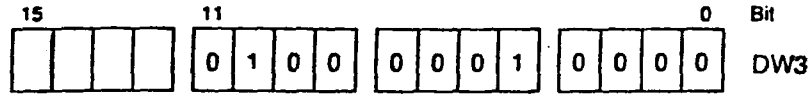
Sabit Bir Sayıyı Yükleme :

Aşağıdaki örnek 38 sayısını nasıl yüklediğini gösterir.

Bir Sayıyı Giriş,Cıkış,flag veya Veri Kelimesi olarak Yükleme:

Yükleme ifadesi : L DW 3

410 sayısı BCD kodunda DW 3 veri kelimesine yüklenir.12~15 byte'ları sayı için geçersizdir.

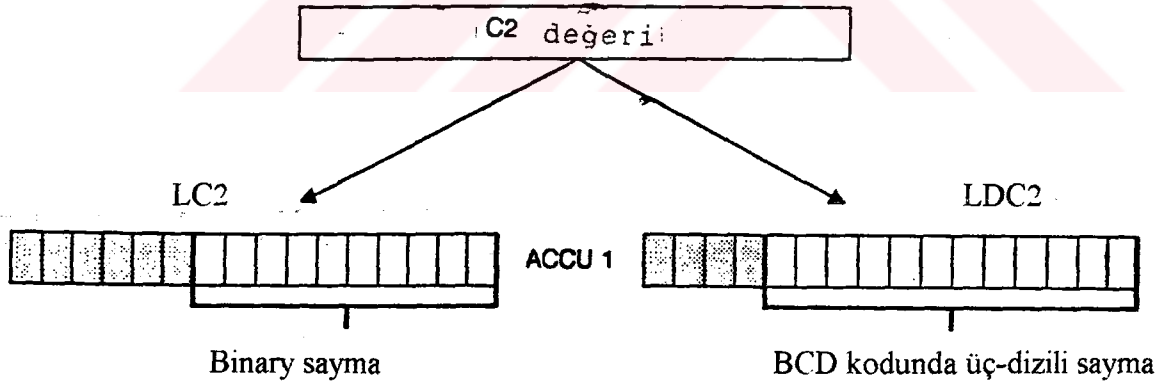


Sayıyı Tarama :

Sayıci durumunu taramak için mantık işlemlerini kullanınız. Sayı sıfır olmadığı sürece, tarama sonucu "1" dir.

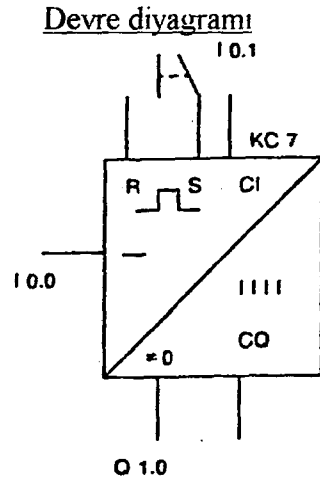
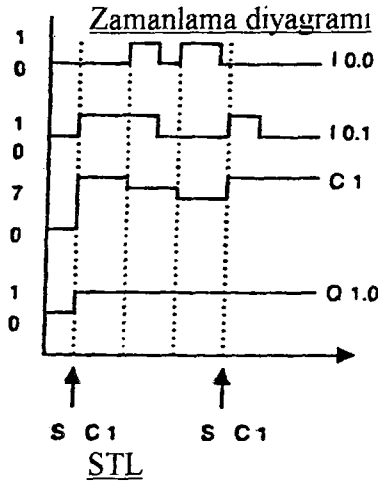
Mevcut Sayıcı Durumunu Cıkarma :

Mevcut sayıcı durumunu ACCU 1'in için yerleştirmek için bir yükleme işlemini kullanabilirsiniz. "BCD'de yükle" işlemleri bir dijital ekran çıkartır.



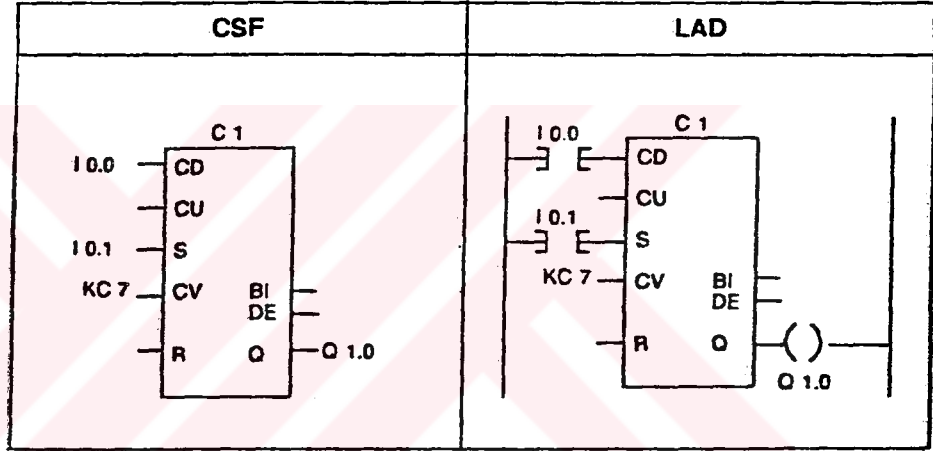
Bir Sayıcıyı Set Etme ve Aşağı Saydırma :

Örnek : I 0.1 girişi set edildiği zaman sayıcı-1 7 sayısına set edilir. Q 1.0 çıkışı şimdi "1" dir. I 0.0'ın darbesinde sayı "1" azalır. Sayı "0" olduğu zaman çıkış "0" a set edilir.



STL

A	I	0.0
CD	C	1
NOP	0	
A	I	1
L	KC	7
S	C	1
NOP	0	
NOP	0	
NOP	0	
A	C	1
=	Q	1.0

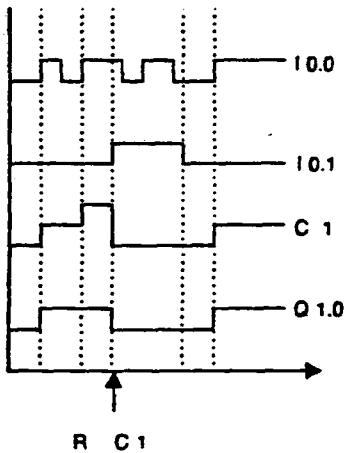


Bir Sayıcıyı Resetleme ve Yukarı Savma :

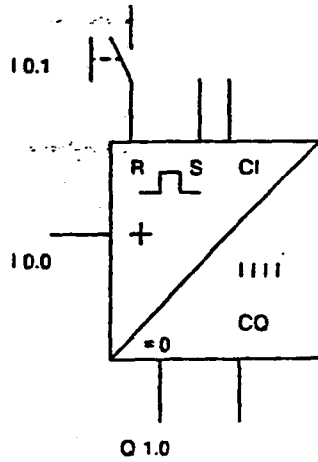
Örnek : I 0.0 girişine darbe geldiğinde sayıcı-1'deki sayı "1" arttırılır. I 0.1 girişi "1" olduğu sürece sayıcı "0" için resetlenir.

A C 1 işlemi sayı "0" olmadığı sürece Q 1.0 çıkışına "1" işaret durumuyla sonuçlanır.

Zaman diyagramı

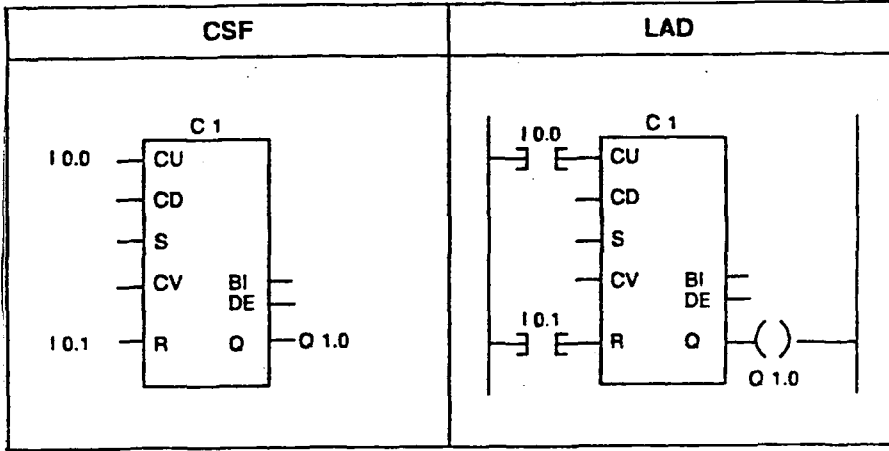


Devre diyagramı



STL

A	I	0.0
CU	C	1
NOP	0	
NOP	0	
NOP	0	
A	I	0.1
R	C	1
NOP	0	



NOP 0
A C 1
= Q 1.0

VIII-1-10-)Karşılaştırma İşlemleri :

Karşılaştırma işlemleri iki akümülatörün içeriklerini karşılaştırır. Karşılaştırma akümülatörlerin içeriklerini değiştirmez.

İŞLEM	İŞLEMCİ	AÇIKLAMA
!=F		<u>Eşit için karşılaştırma:</u> İki akümülatörün içeriği bit(örnekleri) olarak yorumlanır ve eşit olduğu görmek için taranır.
><F		<u>Eşit değil için karşılaştırma:</u> İki akümülatörün içeriği bit(örnekleri) olarak yorumlanır ve eşit olmadığını görmek için taranır.
>F		<u>Daha büyük için karşılaştırma:</u> İki akümülatörün içerikleri sabit sayılar olarak yorumlanır ve ACCU-2'deki işlemcinin ACCU-1'deki işlemciden daha büyük olduğunu görmek için karşılaştırılır.
>= F		<u>Daha büyük veya eşit için karşılaştırma:</u> İki akümülatörün içerikleri sabit sayılar olarak yorumlanır ve ACCU-2'deki işlemcinin ACCU-1'deki işlemciden daha büyük veya eşit olduğunu görmek için karşılaştırılır.
<F		<u>Daha küçük karşılaştırma:</u> İki akümülatördeki içerikleri sabit sayılar olarak yorumlanır ve ACCU-2'deki işlemcinin ACCU-1'deki işlemciden daha küçük olduğunu görmek için karşılaştırılır.
<= F		<u>Daha küçük veya eşit için karşılaştırma:</u> İki akümülatördeki içerikleri sabit sayılar olarak yorumlanır ve ACCU-2'deki işlemcinin ACCU-1'deki işlemciden daha küçük

veya eşit olduğunu görmek için karşılaştırılır.

Karşılaştırma İşlemlerinin Çalışması (Modelleşmesi)

İki işlemciyi karşılaştırmak için bu işlemleri iki akümülatörün içine sırasıyla yüklenir. İşlemlerin yerine getirilmesi R 10'dan bağımsızdır.

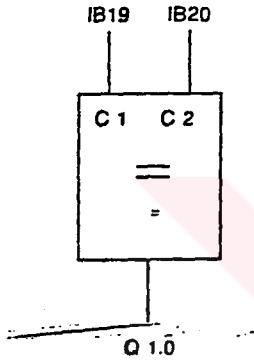
Sonuç binary'dir ve karşılaştırma sonucu tamamsa RLO "1" diğer durumda ise "0" 'dır.

Örnek : IB 13 ve IB 20 giriş değerleri karşılaştırılacaktır. Eğer iki değer eşitse, Q 1.0 çıkışı set edilecektir.

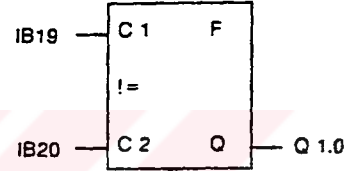
Devre diyagramı

STL

CSF/LAD



```
L IB 19
L IB 20
! = F
= Q 1.0
```



VIII-1-11-)ARİTMETİK İŞLEMLER :

Aritmetik işlemler sabit sayılar halinde akümülatör içeriklerini yorumlar ve gerekeni yaparlar. Sonuç ACCU 1'de depolanır.

İşlem	İşlemci	Açıklama
+F		<u>Toplama:</u> İki akümülatörün içerikleri (toplanır) birbirine eklenir.
-F		<u>Çıkarma :</u> Accu-1'in içerikleri Accu-2'nin içeriklerinden çıkarılır.

Aritmetik İşlemleri Çalıştırma (Modelleme) :

Aritmetik işlem yapılmadan önce her iki işlemci akümülatörlere yüklenmelidir.

Aritmetik işlemler R 10'dan bağımsız yapılıır. Sonuç daha ileri "işlemler" için ACCU 1'de elde edilebilir. ACCU 2 içerikleri değiştirilmez.

STL	AÇIKLAMA		
L C3	Sayıcı -3'ün değeri Accu-1'e yüklenir.		
L C1	Sayıcı-1'in değeri Accu-1'e yüklenir. Accu-1'in ilk içeriği Accu-2'ye kaydırılır.		
+F	İki akümülatörün içerikleri 16 bitlik sayı olarak yorumlanır ve toplanır.		
T QW12	Sonuç (Accu-1'in içeriği) QW12 çıkış kelimesine transfer edilir.		
Sayısal Örnek			
876	15	0	
+	00000011	01101100	+ F ACCU 2
668	00000010	10011100	ACCU 1
=	00000110	00001000	ACCU 1
1544			

VIII-1-12-) BLOK ÇAĞIRMA İŞLEMLERİ

Blok çağırma işlemleri yapılanmış (oluşturulmuş) programın sırasını belirtir.

İŞLEM	İŞLEMCİ		AÇIKLAMA
JU	<input type="checkbox"/>	<input type="checkbox"/>	<u>Şartsız sıçrama :</u> Program taraması RLO'ya bakmaksızın başka bir bloktan devam eder. RLO etkilenmez.
JC	<input type="checkbox"/>	<input type="checkbox"/>	<u>Şartlı sıçrama :</u> RLO "1" olduğu zaman program, başka bir bloktan diğer bir bloğa geçer. Diğer yandan program taraması aynı blokta devab eder.RLO "1" e set edilir.
ID OB PB FB SB			<u>Parametre</u> CPU 100 0-63 0-63 0-63
C	<input type="checkbox"/>	<input type="checkbox"/>	<u>Bir veri bloğunu çağırma</u> Veri bloğu RLO'ya bakmaksızın aktif hale gelir. Program taraması kesilmez. RLO etkilenmez.
G	<input type="checkbox"/>	<input type="checkbox"/>	<u>Bir veri bloğu oluşturma ve silme</u> RLO'ya bakmaksızın RAM içinde veriyi saklamak için bir alan tesis edilir.

ID	Parametre	CPU 100 2-63***
BE		<u>Blok sonu:</u> -Mevcut bloğu RLO'ya bakmaksızın sonlandırır. -Program taraması çağırmanın meydana geldiği blokta devam eder.
BEU		<u>Şartsız blok sonu:</u> -Mevcut bloğu RLO'ya bakmaksızın sonlandırılır. -Program taraması çağırmanın meydana geldiği blokta devam eder.
BEC		<u>Şartlı blok sonu:</u> - RLO="1" olduğu zaman mevcut blok sonlandırılır. -Program taraması çağırmanın meydana geldiği blokta devam eder. -Blok değişimi süresince RLO "1" kalır. -Eğer RLO "0" ise, işlem yapılmaz. -RLO "1" e set edilir ve program taraması devam eder.

SARTSIZ BLOK CAĞIRMA :

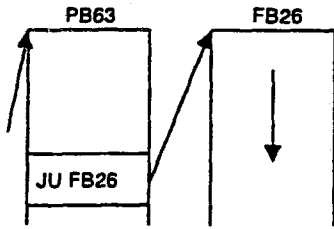
Bir blok diğer bir blok içinde katlara bakmaksızın çağrılır.

Örnek: Özel bir fonksiyon içinde F626 programlanmıştır. Bu blok program da (PB63) birkaç bölgede çağrılır ve aktif hale gelir

Program sırası

STL

Açıklama



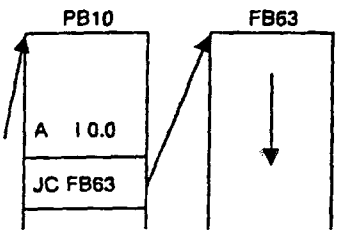
JU FB26

PB63 program bloğundaki "JU FB26" ifadesi FB26 fonksiyon bloğunu çağırır.

SARTLI CAĞIRMA-IC

Bir evvelki şart yerine geldiğinde (RLO="1") bir blok diğer bir bloğun içinde çağrılır.

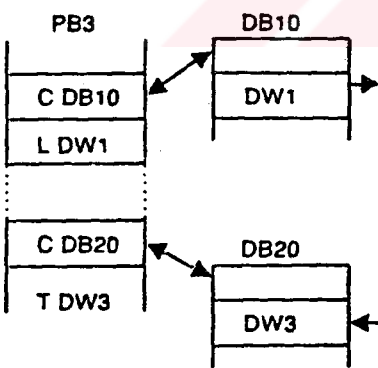
ÖRNEK: Özel bir fonksiyon FB63 içinde programlanmıştır. Bu blok PB10 içinde belirli şartlar sağlandığında çağrılır ve aktif hale gelir.

<u>Program sırası</u>	<u>STL</u>	<u>Açıklama</u>
	S F 1.0 A I 0.0 JC FB63	PB10 program bloğu içindeki "JC FB63" ifadesi I 0.0 giriş "1" olduğu zaman FB63 fonksiyon bloğunu çağırır.

VERİ BLOĞUNU CAĞIRMA "CDB"

Veri blokları daima şartsız olarak çağırılır. Bu işlem yeni veri blokları üretmez. Çağrılmış bloklar program taramasından önce programlanmalı veya oluşturulmalıdır.

ÖRNEK: PB63 program bloğu DB10 veri bloğu içinde DW1 veri kelimesi olarak programlanmış bilgiye ihtiyaç duymaktadır. Diğer bir veri (örneğin ,aritmetik bir işlemin sonucu) DB20 veri bloğunda DW3 veri kelimesi olarak saklanmıştır.

<u>Program sırası</u>	<u>STL</u>	<u>Açıklama</u>
	C DB 10 L DW 1 C DB20 T DW 3	DB10 veri bloğundaki DW1 veri kelimesindeki bilgi akümülatöre yüklenir. ACCU-1'deki içerikler DB20 bloğundaki DW3 veri kelimesine yüklenir.

VERİ BLOĞU ÜRETME VEYA SİLME

“GDBx” ifadeleri veri bloğu çağırır. Bunun yerine bu ifade yeni bir blok üretir. Eğer bu veri bloğunda bir veri kullanmak istiyorsanız, “CDB” ifadesiyle bu ifadeyi çağırırız.

Veri bloğu üretme :

<u>Örnek</u>	<u>STL</u>	<u>Açıklama</u>
Programlayıcının yardımı olmaksızın 128 veri kelimesi bir veri bloğu üretir	L KF+1237 G DB 5	+127 sabit sayısı ACCU-1'e yüklenir. Aynı zamanda ACCU-1'deki eski içerik ACCU-2'ye kaydırılır. Veri bloğu-5 PLC RAM'ında 128 veri kelimesi (0000) uzunluğunda üretilir ve blok adres listesine girilir. Daha sonra “GÜB5” işlemi aktif olur ve eğer ACCU-'deki içerik “0” değilse etkilemez.

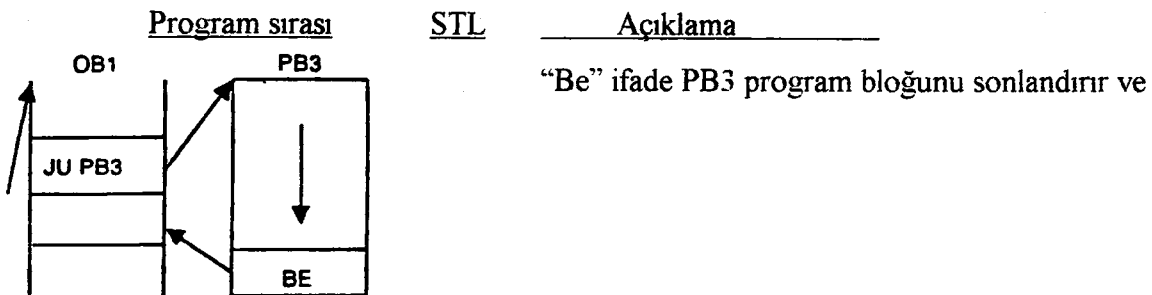
Veri bloğu silme

<u>Örnek</u>	<u>STL</u>	<u>Açıklama</u>
İhtiyaç duyulmayan bir veri bloğunu silmek	L KF +D G DB 5	+0 sabit sayısı ACCU-1'eyüklenir. Aynı zamanda ACCU-2'ye kaydırılır ve PLC RAM'ında olması gereken veri bloğu-5 geçersiz sayılır ve blok adres listesinden çıkarılır.

BLOK SONU “BE”

“BE” işlemi bloğu sonlandırır. Veri bloklarını sonlandırmaya gerek yoktur. “BE” daima bir bloktaki en son ifadedir.

Örnek : PB3 program bloğunu “BE” ifadesiyle sonlandırılması.

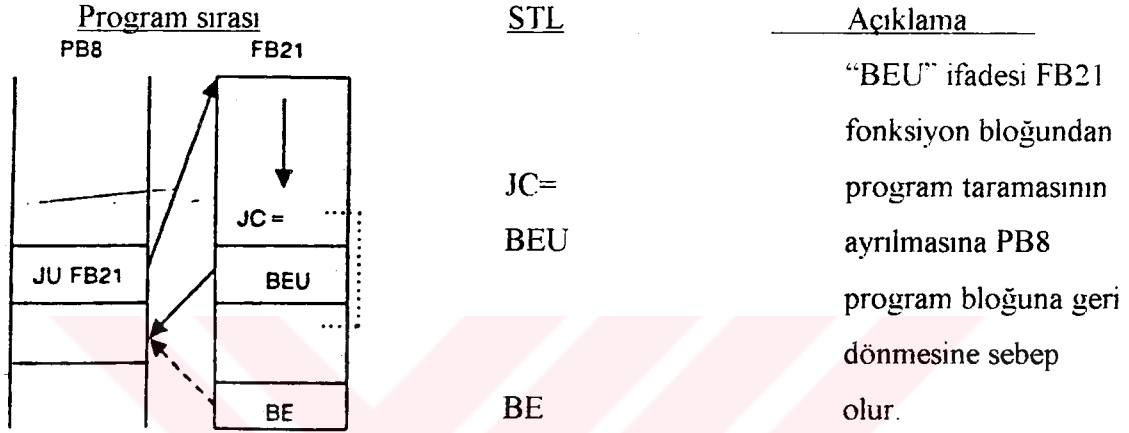


program taramasının OBI organizasyon bloğuna
BE dönmesine sebep olur.

SARTSIZ BLOK SONU "BEU"

"BEU" işlemi bir blok için bir geri gitmeye sebep olur. Bununla beraber sıçrama işlemi fonksiyon bloktaki "BEU" işlemi ile by-pass edilebilir.

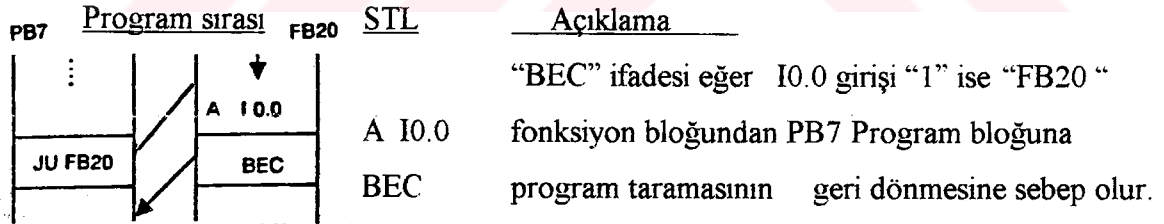
Örnek :FB21 fonksiyon bloğunun taramasını RLO'ya bakmaksızın sonlandırılması.



SARTLI BLOK SONU "BEC"

"BEC" işlemi bir blok içinde bir önceki şart yerine geldiğinde (RLO=1) geri dönüş yapılmasına sebep olur. Lineer program taraması RLO=1 ile devam eder.

Örnek :FB20 fonksiyon bloğunun taramasının RLO=1 ise sıralandırılması



VIII-1-13-)DİĞER İŞLEMLER

İŞLEM	İŞLEMCİ	AÇIKLAMA
STP		Program taramasının sonunda durma :(OB1'de) Mevcut program taramasını sonlandırır. PIQ okunur. Sonra PLC STOP moduna geçer.
NOPO		NO işlemi : RAM'daki 16 bit "0" set edilir.
NOPI		NO işlemi : RAM'daki 16 bit "1" set edilir.

BLD			<u>Ekran üretme işlemi:</u> "BLD" programcı için ekran üretme işlemi yapar.
ID			<u>Parametre</u> 130,131,132,133,255

"DUR" işlemi

Bu işlem (STP) programlanabilir kontrolörü "dur" moduna geçirir. Bu kritik zamanlı sistemler ve PLC hataları meydana geldiğinde gerekebilir.

Bu ifade kullanıldıktan sonra, kontrol programı RLO'ya bakmaksızın sonuna kadar taranır. Daha sonra PLC ID hatası "STS" ile "DUR" moduna geçer. PLC tekrar mod seçici (DUR-ÇALIŞ) veya programlayıcı ile tekrar çalıştırabilirsiniz.

"NOP"(işlem yok)

Bu işlem bellek bölgelerini rezerve eder.

VIII-2-1) İLAVE İŞLEMLER

İlave işlemler işlem setini genişletir. Bununla birlikte bütün bloklarda programlanabilen basit işlemlere kıyasla ilave işlemler aşağıda belirtilmiş olan sınırlamalara sahiptir.

-Bu işlemler yalnızca fonksiyon bloklar içinde programlanabilir.

-Bu işlemler yalnızca STL'de yazılabilir.

VIII-2-1) CPU -103 için yükleme işlemi

İlave yükleme işlemi bilgiyi akümülatöre kopyalar.

İŞLEM	İŞLEMCİ	AÇIKLAMA
L	□ □	<u>Yükleme:</u> Sistem verisindeki bir kelime RLO'ya bakmaksızın ACCU-1'e yüklenir.
ID RS		<u>Parametre</u> 0-255

ÖRNEK

Sistem verisi üzerinden
SINECLI bus işlemi parametrelerini
set etmek için programlayıcı ve

STL

L RS 57

Açıklama

Programlayıcı ve ilave
sayılı ACCU 1'i yükleme

SD 57'de ilave numaraları ACCU-

1'in içine girilmelidir

VIII-2-2-) CPU-103 için Enable İşlemi

Kenar değişimi olmaksızın dahi aşağıdaki işlemi çalıştırmak için enable işlemini (FR) kullanabilirsiniz.

- Bir timer'i çalıştırma.
- Bir sayıcıyı set etme.
- Yukarı ve aşağı sayma.

İŞLEM	İŞLEMCI	AÇIKLAMA	
FR	<input type="checkbox"/>	<input type="checkbox"/>	<u>Enable timer sayıcı:</u> Timer ve sayıcı RLO'nun yükselen kenarında "enable" edilebilir. Bu işlem bir timer'i tekrar çalıştırır, bir sayıcıyı set eder veya "Başka" işlemin de RLO=1 ise sayıcının yukarı veya aşağıya saymasına sebep olur.
<u>ID</u>	<u>Parametre</u>		
T		0-127	
C		0-127	

ÖRNEK

I 0.0 girişi T2 timer'inin genişletilmiş pulse'li olarak (pulse genişliği 50S) çalıştırır. Pulse süresince bu timer Q 1.0'ı set eder.

Eğer Q 1.0 çıkışı tekrarlı olarak resetleniyorsa, timer de tekrarlı olarak yeniden çalıştırılmalıdır.

STL

```
A I 0.0      Genişletilmiş pulse'li pulse'li
L K 500.1    olarak T2 timer'ini çalıştırır.
SE T 2      Q 1.0 çıkışı 50S için set edilir.
A T 2
= Q 1.0
A Q 1.1      Eğer Q 1.1 çıkışı I 0.0 girişi set
FR T 2      edildiği sürece (RLO'nun pozitif
BE          kenar değişimi) set edilirse, T2
           timer'i tekrar çalıştırılır. Bu yüzden
           0.1 çıkışı tekrar çalıştırma zamanında
           set olarak kalır veya resetlenir.
```

Eğer I 0.0 girişi Q 1.0 çıkışının kenar değişimi sürsince set edilirse, time tekrar çalıştırılmaz.

VIII-2-3-) CPU 103 İÇİN BİT TEST İŞLEMLERİ:

Bit test işlemleri bit digital işlemcileri tekrar, bit test işlemcileri daima lojik işlemin başında olmalıdır.

İŞLEM	İŞLEMCİ	AÇIKLAMA
TB	<input type="checkbox"/>	İşaret durum "1" için bir biti test etme: Tekil bit RLO'ya bakmaksızın taranır. Bitin işaret durumuna göre RLO etkilenir.
TBN	<input type="checkbox"/>	İşaret durumu "0" için bir biti test etme: Tekil bit RLO'ya bakmaksızın taranır. RLO bitin işaret durumuna göre RLO etkilenir.
SU	<input type="checkbox"/>	Bir biti şartsız olarak set etme: Adreslenmiş bit RLO'ya bakmaksızın 1'e set edilir. RLO etkilenmez.
RU	<input type="checkbox"/>	Bir biti şartsız olarak reset etme: Adreslenmiş bit RLO'ya bakmaksızın 0'a set edilir. RLO etkilenmez.
ID		Parametre
T		0.0 - 127.15
C		0.0 - 127.15
D		0.0 - 255.15
RS		0.0 - 255.15

İşlem	TB		TBN	
Gösterilmiş işlemcideki bitin işaret durumu	0	1	0	1
Lojik işlemin sonucu	0	1	1	0

ÖRNEK	STL	AÇIKLAMA
Parça eşyaları sayan bir fotoelektrik barier I 0.0 girişinde tesis edilmiştir. Her bir 100 parça sonunda program FB5 veya FB&'ya atlayacaktır. 800 parça sonunda sayıcı 10 otomatik	C DB 10 A I 0.0 CU C 10 A I 0.1 L KC 000 S C 10 O I 0.2	Veri bloğu 10'u çağır. İ 0.1 girişi sayıcı 10'u "0" sabitiyle yükler. I 0.0'daki her bir pozitif kenar değişimiyle sayıcı "1" azaltılır. Sayıcı I 0.2 girişi veya F 5.2 bayrağı tarafında resetlenir. Sayıcının mevcut sayısı BCD

olarak resetlenecek ve	O	F	5.2	kodunda DW 12 veri kelimesinde
sayma yeniden	R	C	10	saklanır.
başlayacaktır.	LD	C	10	
	T	DW	12	
	TBN	D	12.8	DW 12 veri kelimesindeki 8 NO.lu
	JC	FB	5	Bit sıfır olur olmaz, program işlemcisi
				FB5 fonksiyon bloğuna atlar.
				DW 12 veri kelimesindeki 8 NO.lu
	TB	D	12.8	bit "1" olur olmaz, program taraması
	JC	FB	6	FB6 fonksiyon bloğuna atlar.
				DW 12 veri kelimesinin 11 NO.lu
				biti "1" olur olmaz (sayı "800" olduğu
	TB	D	12.11	zaman), F5.2 bayrağı şartlı olarak set
	=	F	5.2	edilir.

Parça eşyaları sayan bir	:	A	I	0.3	I 0.4 girişi sayıcı-20'nin sayısı "0"
fotoelektrik bariyer I 0.3	:	CU	CI	2	sabitiyle yüklenir. Sayı I 0.3 girişindeki
girişinde tesis edilmiştir.	:	A	I	0.4	her bir pozitif kenar değişimiyle "1"
Her bir 256 parçadan sonra	:	L	KC	000	azaltılır. Eğer sayı 256=100H (bit -8
sayıcının sıfır olduğu	:	S	C	20	"1" olduğu an)'e varırsa, program
farkedilir ve sa ve sayma					taraması "FULL" etiketine (label) atlar.
yeniden başlatılır.	:	TB	C	2 0.8	Dha sonra blok sonlandırılır.
	:	JC	=	FULL	C-20 sayıcının 8 NO'lu biti şartsız
	:	BE			olarak sıfıra set edilir. Sonra sayı tekrar
					OOOH'a geri döner.

FULL: RU C 208

: BE

VIII-2-4-) DİJİTAL MANTIK İŞLEMLERİ

Dijital lojik işlemleri her iki akümülatörün içeriklerini bit bit lojik olarak birleştirir.

İşlem	İşlem	Mana (Açıklama)
AW		AND işlemi vasıtasıyla bit bit birleştirme.
OW		OR işlemi vasıtasıyla bit bit birleştirme.
XOW		XOR işlemi vasıtasıyla bit bit birleştirme.

Dijital Lojik İşlemlerini Metodlama:

Dijital bir lojik işlemi RLO'ya bakmaksızın işleme sokulur ve RLO'dan etkilenmez. Aritmetik işlemlerin sonuçlarına göre durumların kodlarını set eder.

Ariemetik işlemin sonucu Accu-1'de elde edilir. Accu-2'nin içeriği etkilenmez.

STL

L IW 92

L KH OOFF

AW

T QW 82

AÇIKLAMA

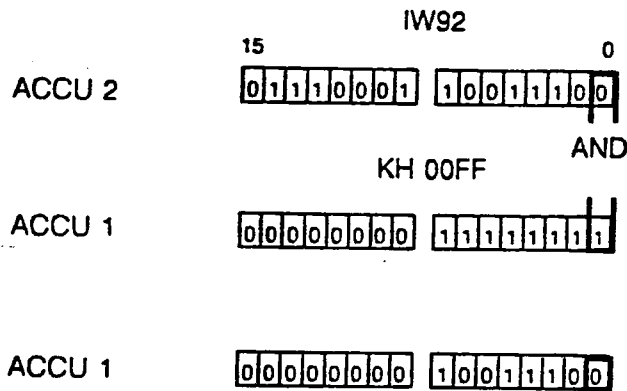
- IW 92 giriş kelimesini Accu-1'e yükle.

- Sabiti Accu-1'e yükle, Accu-1'in bir önceki içeriğini Accu-2'ye kaydır.

- Her iki akümülatörün içeriklerini bit bit lojik ile birleştir.

- Sonuç içeriğini Accu-1'den QW 82 çıkış kelimesine transfer et.

SAYISAL ÖRNEK



IW 92 giriş kelimesindeki 8-en yüksek dereceli biti 0'a set et.

Her iki kelimeyi bit bit karşılaştır. Eğer bitlerin her ikisi de 1 ise; sonuç biti 1'e set et.

ÖRNEK:

STL

L IW 36

- Accu-1'e IW 36 giriş kelimesini yükle.

L KH OOFF

- Accu-1'e bir sabit bit yükle, Accu-1'in bir önceki içeriğini

OW

Accu-2'ye kaydır.

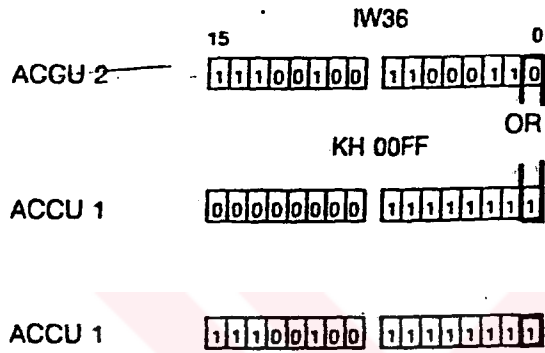
T IW 36

- Her iki akümülatör içeriğini lojik OR ile bit bitbirleştir.

- Sonucu (Accu-1'in içeriğini) IW 36 girişine transfer et.

AÇIKLAMA

SAYISAL ÖRNEK



IW 36 giriş kelimesindeki 8 düşük dereceli bitleri 1'a set et. Her iki kelimeyi bit bit kıyasla.

Eğer uygun bitlerden herhangi biri 1'e eşit ise; sonuç kelimesindeki bu biti 1'e set et.

ÖRNEK:

STL

L IW 70

- Accu-1'e IW 70 giriş kelimesini yükle.

L IW 6

- Accu-1'e IW 6 giriş kelimesini yükle. Accu-1'in bir önceki içeriğini Accu-2'ye kaydır.

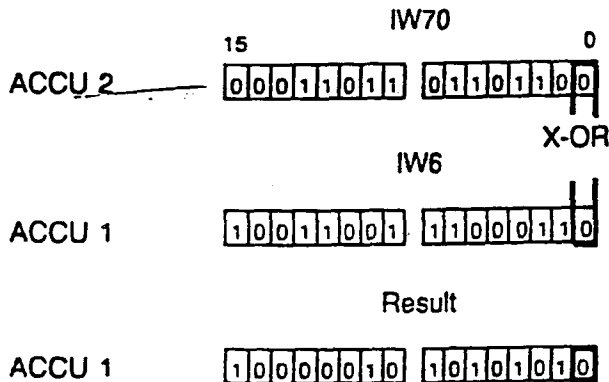
XOW

T QW 86

- Her iki akümülatörün içeriklerini lojik XOR ile bit bit birleştir.

- Sonucu (Accu-1'in içeriği) QW 86 çıkış kelimesine transfer et.

SAYISAL ÖRNEK



IW 70 ve IW 6 giriş kelimelerinin eşit olup olmadığını görmek için inceleme.

Eğer Accu-1 ve Accu-2'de uygun bitler eşit değilse; ancak bu durumda sonuç biti

1'e set edilir.

VIII-2-5-) KAYDIRMA İŞLEMLERİ:

Kaydırma işlemleri Accu-1'deki bir bit modelini kaydırır. Accu-2'nin içerikleri bundan etkilenmez. Kayma Accu-1'in içeriklerinin ikinin kuvvetleriyle çarpar veya böler.

İŞLEM	İŞLEMCİ	AÇIKLAMA
SLW	☐	<u>Sola kaydırma:</u> Accu-1'deki bit örneği sola kaydırılır.
SRW	☐	<u>Sağa kaydırma:</u> Accu-1'deki bit örneği sağa kaydırılır.
		<u>Parametre</u> 0 - 15

Bir kaydırma işleminin yapılması:

Kaydırma işlemlerinin yapılması ŞARTSIZ'dir. Bu işlemde RLO etkilenmez. Bununla birlikte; kaydırma işlemleri şart kodlarını set ederler.

Sırasıyla; dışarı kaydırılan son bitin durumu sıçrama fonksiyonlarıyla taranabilir.

Kaydırma işlemi parametresi, Accu-1'in içeriklerini sağa veya sola kaydıracak olan bit pozisyonlarının sayısını gösterir. Kaydırma süresince boşaltılmış olan bit pozisyonlarına sıfır yüklenir.

Accu-1'den kaydırılan bit içerikleri kaybolur.

"0"lı bir kaydırma işlemi "NOP" işlemine benzer olarak ele alınabilir.

Kaydırma işlemi yapmadan önce, işleme girecek olan işlemciyi Accu-1'e yükleyiniz. Değiştirilmiş olan işlemci ancak bu işlemle elde edilebilir.

ÖRNEK:

	<u>STL</u>	<u>AÇIKLAMA</u>
L DW	2	- IW 2 veri kelimesinin içeriklerini Accu-1'e yükle.
SLW	3	- Accu-1'deki model biti üç pozisyon sola kaydır.
TDW	3	- Sonucu (Accu-1'in içeriği) DW 3 veri kelimesine transfer et.

Sayısal olarak:

ACCU 1 15 464₁₀ (DW2) 0
 000000001 11010000

464₍₁₀₎ değeri DW 2 veri kelimesinde depolanmıştır. Bu değeri $2^3 = 8$ 'le çarpınız. Bunu Accu-1'deki mode bit üç pozisyon sola kaydırarak yapabilirsiniz

← SLW 3

ACCU 1 15 3712₁₀ 0
 00001110 10000000

ÖRNEK:

STL

AÇIKLAMA

L IW 124

- IW 124 giriş kelimesinin değerini Accu-1'e yükle.

SRW 3

- Accu-1'deki mode biti 4 pozisyon sağa kaydır.

T QW 126

- Sonucu (Accu-1 içeriği) QW 126 çıkış kelimesine transfer et.

Sayısal olarak:

ACCU 1 15 352₁₀ (IW124) 0
 00000001 01100000

352₍₁₀₎ değeri IW 124'de saklanmıştır. Accu-1'deki uygun mode biti 352₍₁₀₎'u 24 = 16'ya bölmek için 4 pozisyon sağa kaydırabilirsiniz

SRW 4 →

ACCU 1 15 22₁₀ 0
 00000000 00010110

VIII-2-6-) DEĞİŞİM (DÖNÜŞÜM) İŞLEMLERİ

Değişim işlemleri Accu-1'deki değeri değiştirir.

İŞLEM	İŞLEMCİ	AÇIKLAMA
CFW		<u>Bir'in tümleri:</u> ⇒ Accu-1'in içeriğini bit bit değiştirir.
CSW		<u>İki'nin tümleri:</u> ⇒ Accu-1'in içeriğini bit bit değiştirir. Daha sonra 0001 _H eklenir.

Bu işlemler RLO'dan bağımsızdır ve RLO'yu etkilemez. CSW işlemi şart kodlarını set eder.

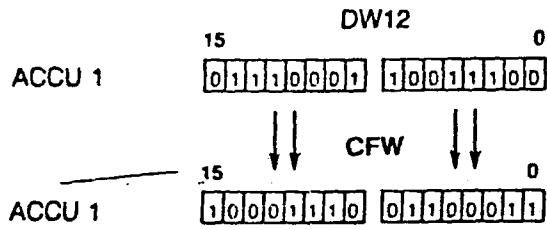
ÖRNEK:

STL

AÇIKLAMA

- L DW 12 - DW 12 veri kelimesinin içeriğini Accu-1'e yükle.
CFW - Accu-1'deki bilerin hepsini değiştir.
T QW 20 - Accu-1'in yeni içeriklerini QW 20 çıkışı kelimesine transfer et.

Sayısal olarak:



Bir sistemde normalde açık kontaklar normalde kapalı kontaklarla yer değiştirilebilir. Eğer DW 12 kelimesindeki bilgi bir önceki etkisinde tutulacaksa; DW 12 değiştirilmelidir.

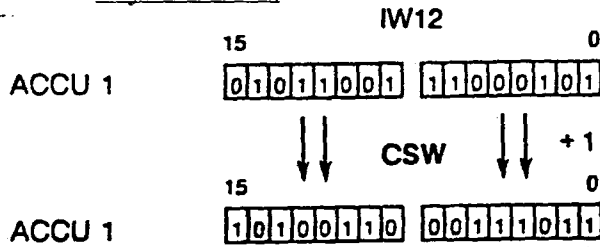
ÖRNEK:

STL

AÇIKLAMA

- L IW 12 - IW 12 giriş kelimesini Accu-1'e yükle.
CSW - Bütün bitleri değiştir ve "1" ekle
T DW 10 - Değiştirilmiş kelimeyi DW 100 veri kelimesine transfer et.

Sayısal olarak:



IW 12 giriş kelimesindeki değerlerin negatif değerini elde ediniz

VIII-2-7-) AZALTMA/ ARTIRMA İŞLEMLERİ

Azaltma /artırma işlemleri Accu-1'deki yüklenmiş veriyi değiştirir.

İŞLEM	İŞLEMCİ	AÇIKLAMA
		Azaltma:

D		⏏	Akümülatörün içeriklerini azaltma.
I		⏏	<u>Artırma:</u> Akümülatörün içeriklerini artırma. Accu-1'in içeriği parametrede gösterilmiş sayı tarafından ya artırılır ya da azaltılır. İşlem şartsız olarak yerine getirilir.
		<u>Parametre</u>	0 - 255

Azaltma ve artırma işlemleri RLO'dan bağımsızdır ve RLO'yu veya şart kodlarını etkilemez.

Parametre ACC-1'in içeriğini geçiştirecek olan değeri gösterir.

İşlemlerde ondalık değerler kullanılır fakat sonuç ACCU-1'e binary formda depolanılır.

Değişiklikler yalnızca akümülatördeki düşük dereceli bayt ile ilgilidir.

<u>ÖRNEK</u>	<u>STL</u>	<u>AÇIKLAMA</u>
Hexadimol sabiti	C DB 6	-DB6 Veri bloğunaçağır.
1010H'ı 16 of'tır ve DW8 veri kelimesinde depola	L KH 1010 I 16	-1010H hexadesimol sabiti ACCU-1'yükler.
Artırılmış sonuçtan 33 çıkar ve yeni sonucu DW9 veri kelimesine depola.	T DW 8	-ACCU-1'in düşük dereceli baytını "16" artır.1020H sonucuna ACCU-1'e yerleştir.
	D 33	-ACCU-1'in içeriğini (1020H) DW8 veri kelimesine transfer et. Artırılan sonuç ACCU-1'de olduğundan direk olarak 33 azaltılabilir.
	T DW 9	-Sonuç FFFH olacaktır. -Bunula beraber, ACCU-1'in yüksek dereceli biti düşük dereceli bayt ile beraber azaltılamayacağından ACCU-2deki sonuç 10FFH olur. ACCU-1'in içeriğine DW9'a (10FFH) transfer et.

VIII-2-8-)DISABLE/ENABLE KESİNTİSİ

Disable/enable kesinti işlemleri kesme sürücülü ve zaman kontrollü program taramasını etkiler.Bu işlemler prosesi veya zaman kesintilerini bir ifade veya blok dizisinin işlenmesiyle girişim uğramasına engel olur.

İŞLEM	İŞLEMÇİ	AÇIKLAMA+
IA		Disable kesinti
RA		Enable kesinti

Bu işlemleri RLO'ya bağlı değildir ve RLO şart kodlarını etkilemez.Bir IA ifadesi işleme girdikten sonra kesintiler hemen yapılır."RA" ifadesi ise IA'nın etkisini iptal eder.

ÖRNEK

STL

AÇIKLAMA

Bir belirli program bölümünde disable kesinti yapma ve sonra enable kesintili ile devam etme

= Q 1
IA
A I 0.0
JU FB 3

Disable kesinti
Eğer bir kesinti meydana gelirse,IA ve RA arasındaki program bölümü kesintiye bakmaksızın taranır

RA

Enable kesinti.
Meydana gelmiş kesintiler RA işleminden sonra işlem gör

VIII-2-9-) DO İŞLEMİ

“DO” işlemi step-5 işlemlerini indekslenmiş işlemler halinde işlem görmesini sağlar.Bu ise kontrol programının işlemesi süresince bir işlemcinin parametresini değiştirmeye müsaade eder.

“DO İFADELERİ”

Do bayrak kelimesi veya veri kelimesi-x RLO'dan etkilenmeyen iki-kelimeli bir ifadedir.”Do” aşağıdaki iki ifadeden oluşur.

* Birinci ifade “DO” işlemi ve bir bayrak kelimesi/veri kelimesinden oluşur.

* İkinci ifadede ise işlemi ve işlemci tanımlayıcısını belirtir.Bu işlemci tanımlayıcısı işlemek istediğimiz kontrol programını belirtir.

Kontrol programı bayrak veya veri kelimesinde saklanmış olan parametre ile birlikte çalışır.Bu parametre birinci ifadede tanımlanmıştır.Eğer binary işlemlerini, girişleri,çıkışları veya bayrakları indekslemek istiyorsanız bu kelimenin en yüksek baytı bit adresine girmelisiniz.Alçak bayt adresine bayt adresini girebilirsiniz.Diğer herhangi bir durumda yüksek bayt “0” olmalıdır.

Aşağıdaki işlemleri “DO” ifadesiyle birleştirebilirsiniz.

İşlemler	Açıklamalar
A ¹ ,AN, O,ON	Boolean logic işlemleri
S, R,=	Set/Reset işlemleri
FR T,RT, SF T, SD T, SP T, SE T	Timer Q işlemleri
FR C, RC, SC, CD C, CU U	Sayıcı işlemleri
L, LD, D	Yükleme ve transfer işlemleri
JU=, JC=JZ=, JN=, JP=, JM, JO=	Atlama işlemleri
SLW, SRW	Kayma işlemleri
D, I	Azalma ve artırma işlemleri
C, DB, JU, JC, TNB	Blok çağırma işlemleri

Aşağıdaki şekil veri kelimesi içeriğinin sıradaki ifade parametresini nasıl belirlendiğini (tespit edildiğini) gösterir.

	DBG	FBX	Asıl program
		: C DB 6	: C DB 6
DW12	KH=0108	: DO DW 12	: A I 8.1
		: A I 0.0	
DWI3	KH=0001	: DO DW 13	: FR T 1
		: FR T 0	

ÖRNEK: Bu örnek yeni parametreleri her bir program taramasında nasıl üretildiğini gösterir.

<u>ÖRNEK</u>	<u>STL</u>	<u>AÇIKLAMA</u>
DW20'de DW100'e kadar olan veri işlemlerini "0" işaret durumuyla set etme .Veri kelimelerinin parametresi için indeks kaydı DWI'dir.	: C DW 202 : L KB 20 : T DW 1 F1 :L KH 0 :DO DW1 : T DW 0 :L DW 1 :L KB 1 :+F :T DW 1 :L KB 100 :<=F :JC=F1	<ul style="list-style-type: none">• Veri bloğu DB 202'yi çağır• 20 sabit sayısını ACCU-1'e yükle• ACCU-1'in içeriklerini DW1 veri kelimesine transfer et.• ACCU-1'e "0" hexadesomol sabitini yükle• DW1 veri kelimesine DO işlemini uygula• ACCU-1'deki içerikleri veri kelimesine transfer et. Bu veri kelimesinin adresi Dw1 veri kelimesine saklanmıştır.• ACCU-1'e Dw1 veri kelimesini yükle.• ACCU-1'e "1" sabit sayısını yükle.Dw1'daki veri kelimesi ACCU-2'ye kaydırılır.• ACCU-1'e ACCU-2'yi ekle ve sonucu ACCU-1'de sakla (veri kelime adresi daha yüksektir.).• Dw1 veri kelimesine (yeni veri kelime adresi) ACUU-1'in içeriklerini transfer et• 100 sabit sayısı ACCU-1'e yüklenir ve yeni veri kelime adresi ACCU-2'ye kaydedilir.• ACCU'ları küçük eşit için kıyasla: ACCU2≤ACCU1• Eğer ACCU-2 ≤ACCU1 ise şartlı olarak F1 etikeni atla.

VIII-2-10)ATLAMA İŞLEMLERİ

İŞLEM	İŞLEMCİ	AÇIKLAMA
JU=	<input type="checkbox"/>	<u>Şartsız atlama:</u> Şartsız atlama ;şartlardan bağımsız olarak yapılır.
JC=	<input type="checkbox"/>	<u>Şartlı atlama :</u> Şarth atlama RLO "1" olduğu zaman yapılır.RLO "0" ise ifade yapılmaz ve RLO "1" e set edilir.
JZ=	<input type="checkbox"/>	<u>Sonuç "0" ise atlama:</u> Şartlı atlama RLO "1" olduğu zaman yapılır.RLO "0" ise ifade yapılmaz ve RLO "1" e set edilir.
JN=	<input type="checkbox"/>	<u>Sonuç "0" değilse atlama:</u> Atlama $CC1 \neq CC0$ olduğunda yapılır. RLO değişmez..
JP=	<input type="checkbox"/>	<u>Sonuç pozitif ise atlama:</u> Atlama $CC1=1$ ve $CC0=0$ ise yapılır.RLO değiştirilmez.
JM=	<input type="checkbox"/>	<u>Sonuç negatif ise atlama:</u> Atlama $CC1=0$ ve $CC0=1$ ise yapılır.RLO değiştirilmez.
JO=	<input type="checkbox"/>	<u>Aşırı akıma atlama :</u> Atlama aşırı akım meydana gelirse yapılır.Diğer yandan
ID Atlama etiketi(4 karaktere kadar)		

Sembolik bir atlama hedefi (atlama etiketi) daima atlama işleminin yanında girilmelidir.Bu atlama etiketi (4) karaktere kadar olmalıdır.Birinci karakter bir alfabe harfi olmalıdır.

Programlama yaparken aşağıdaki seçenekleri göz önünde bulundurunuz.

- Kesin atlama mesafesi program hafızasındaki +127 veya -128 kelime azmayabilir.Boel ifade iki kelime olabilir.(Örneğin ;bir sabiti yükleme)Uzun bir atlama için ;bir ileri (gelişmiş mesafe kullanınız.)

- Atlamalar yalnızca bir blok içinde yapılabilir.

- Segmen sınırları üzerinde atlamaya müsaade edilmez.("BLO 255)

- Atlama etkileri yalnızca CPU 102 için ;taramaların bir serin 5'in başlangıcında set edilebilir.

ÖRNEK

STL

AÇIKLAMA

Eğer IW1 giriş kelimesi bitlerin hiçbiri set edilemez ise, program taraması “AN 1” etiketine atlar. IW1 giriş kelimesi ve QW3 çıkış kelimesi uyuyorsa, program “AN 0” etiketin geri artar. Diğer yandan IW1 giriş kelimesi ve DW12 veri kelimesi kıyaslanır. Eğer IW1 giriş kelimesi DW12 veri kelimesinden daha büyük veya daha küçük ise, program taraması “DEST” etiketine atar.

ANO: L IW1
:IL KH 0000
:+F
:JZ=ANI
:AI 0.0
ANI: L IW 1
:L QW 3
:X QW
:JN=ANO
:L IW1
:L DW 12
:>F
:JC=DEST

DEST: A IO1

ACCU-1’e IW1 giriş kelimesini yükle. Eğer ACCU-1’in içeriği “0” eşit ise “AN1” etiketine atla. Diğer durumda sıradaki ifade “IO.0” işleme girer.

IW1 giriş kelimesi ve QW3 çıkış kelimesini kıyaslayınız. Eğer eşit iseler ACCU-1’deki bireysel(individual) bitleri set ediniz

Eğer ACCU-1’in içerikleri “0” değilse “ANO” etiketine atla. Diğer durumda sıradaki ifadeler işleme girecektir.

Eğer RLO “1”e eşit ise, “DEST” etiketine atla. Eğer RLO=0 ise sıradaki ifadeyi işleme sokun.

VIII-2-11-) YERDEĞİŞTİRME SLEMLERİ

Bu parametreler programda açık, kesin işlemciler olarak işgörürler. Bu işleyiş için özel işlemler gereklidir. Bununla beraber bu özel işlemler yerdeğıştirmesiz işlemlerin etkilerinden daha değışik bir etki göstermezler. Bu işlemlerin özet tanımı ve örnekler aşağıdadır.

Binary logic işlemleri

İŞLEM	İŞLEMCİ	AÇIKLAMA
A=	□	<u>AND işlemi:</u> “1” için kesin (formal) işlemciyi tarar.
AN=	□	<u>AND işlemi:</u> “0” için kesin işlemciyi tarar.
O	□	<u>OR işlemi:</u> “1” için kesin işlemciyi tarar.
ON=	□	<u>OR işlemi:</u>

		"0" için kesin işlemciyi tarar.		
Kesin işlemci		<u>İzin verilmiş asıl işlemci</u>	Parametre tipi	Veri tipi
		Binary formda adreslenmiş girişler, çıkışlar ve bayraklar. Zamanlayıcı ve sayıcılar	I,Q,F T,C	B1

VIII-2-12-)Set/Reset işlemleri:

İŞLEM	İŞLEMCİ	AÇIKLAMA		
S =	—	Bir kesin işlemciyi (binary) set eder.		
R8 =	—	Bir kesin işlemciyi (binary) reset eder.		
= =	—	<u>Aktarma :</u> RLO kesin işlemciye aktarılır.		
Kesin işlemci		İzin verilmiş asıl işlemci	Parametre tipi	Veri tipi
		Binary formda adreslenmiş girişler ,çıkışlar ve bayraklar.	I,O,F	B1

ÖRNEK:FB30 OB1'deki parametrelere kaydırılır

OB1'deki Çağırma	FB30'daki program	İşleyecek program
:JU FB 30	:A =ON1	: A I 0.0
İ SİM: BİRLEŞTİR	:AN =ON2	: AN I 0.1
ON1:IO.0	:O =ON3	: O I 0.2
ON2:IO.1	:5 =NOT5	: S Q 1.2
ON3:IO.2	:= =OFF1	:= Q 1.0
VAL1:IO.3	:A =VAL1	:A I 0.3
OFF1:Q1.0	:A =ON2	:A I 0.1
OFF2:Q 1.1	:ON =ON3	:ON I 0.2
MOT5:Q1.2	:RB =MOT5	:R Q 1.2
:BE	:= =OFF2	:= Q 1.1
	:BE	

VIII-2-13-)Yükleme ve transfer işlemleri

İŞLEM	İŞLEMCİ	AÇIKLAMA		
L=	<input type="checkbox"/>	Bir kesin işlemciyi yükle		
LD=	<input type="checkbox"/>	BCD kodunda bir kesin işlemciyi yükle.		
LW=	<input type="checkbox"/>	Bir kesin işlemcinin bit modelini yükle.		
T=	<input type="checkbox"/>	Bir kesin işlemciyi transfer et.		
Kesin işlemci		İzin verilmiş Asıl işlemciler	Parametre tipi	Veri tipi
For L=		Binary formda adreslenmiş giriş, çıkış, bayraklar Veri Timer ve sayıcılar	I,Q,F PW*PY* DW,DR,OL T,C	BY,W
For LD=		Timer ve sayıcılar	T,C	
For LW=		Bit modeli	D	KF,KH:KM ,KY,KS,KT ,KC
For T=		Binary formda adreslenmiş giriş, çıkışlar , veri (OW,DR,DL) ve bayraklar	I,Q,DW,DR ,DL;F, PW*PY*	BY,W

ÖRNEK:FB34 PB1'deki parametre kaydırılır.

PB1'i çağırır.	FB34'teki program	İşleyecek program
:JU FB 34İ	:A =IO	:A I 90
İSİM :YÜKLE/TRANSFER ET	:L =L12	:L FW 10
IO:I O.O	:S C 6	:S C 6
I1:I O.1	:A =I1	:A I 0.1
L1:FW 10	:LW =LW1	:L KC 140
LW1:KC 1A0	:S C 7	:S C 7
LC1:C 7	:A I 0.2	:A I 2
T1:Q W 4	:C C 6	:CU C 6
LW2:KC 160	:C C 7	:CU C 7
:BE	: =LC	:LD C 7
	:T =T1	:T QW 4
	:A I 0.3	:A I 0.3
	:R C 6	:R C 6
	:R C 7	:R C 7
	:LW =LW2	:L KC 160
	:LD =LC1	:LD C 7
	:!=F	:!=F

:R	C	7	:R	C	7
:B			:BE		

VIII-2-12-) TIMER VE SAYICI İŞLEMLERİ

İŞLEM	İŞLEMÇİ	AÇIKLAMA
FR=	<input type="checkbox"/>	Bir cold çalıştırma için kesin işlemciyi elde etme
RD=	<input type="checkbox"/>	Kesin işlemciyi (Dijital) resetleme
SP=	<input type="checkbox"/>	Akümülatörde saklanmış değeri kullanan bir kesin işlemci olarak belirlenmiş pals timerini çalıştırır.
SD=	<input type="checkbox"/>	Akümülatörde saklanmış değeri kullanan bir kesin işlemci olarak belirlenmiş geçirmeli timeri çalıştırır.
SEC=	<input type="checkbox"/>	Akümülatörde saklanmış değeri kullanan kesin işlemci olarak belirlenmiş genişletilmiş pulse timeri çalıştırır veya akümülatördeki sayıyı kullanılarak bir kesin işlemci olarak bir sayıcı set eder.
SSU=	<input type="checkbox"/>	Akümülatörde saklanmış değeri kullanan bir kesin işlemci olarak belirlenmiş saklı-gecikmeli timeri çalıştırır veya bir kesin işlemci olarak belirlenmiş sayıcıyı yukarı doğru sayacak şekilde çalıştırır.
SFD=	<input type="checkbox"/>	Akümülatörde saklanmış değeri kullanan bir kesin işlemci olarak belirlenmiş off-delay timeri çalıştırır veya bir kesin işlemci olarak belirlenmiş bir sayıcıyı aşağı sayacak şekilde çalıştırır.
Kesin işlemci		İzin verilmiş asıl işlemci
		Parametre tipi
		Veri tipi
		Timerler ve sayıcılar
		T,C ¹

Timer ve sayıcıları belirleme

Temel işlemlere benzer olarak bir zaman veya sayıyı kesin işlemci olarak belirlenebilir. Bu durumda ister bir işlemci kelimesini yerleştirmek isterse bir sabit olarak belirlensin. Aşağıdaki gibi ayırt edilmelidir.

* İşlemci kelimeleri I,Q parametre tipi "W" veri tipi olabilir. Bunları akümülatörün içine yerleştirmek için "L=" işlemini kullanınız.

* Sabitler "D" parametre tipi ve KT veya KC veri tipi olabilir. Kesin işlemcileri akümülatöre yüklemek için "LW=" yi kullanınız.

Aşağıdaki örnekler timer ve sayıcı işlemlerinin nasıl çalıştığını gösterir.

ÖRNEK=1

Fonksiyon Bloğunu Çağırma	(FB32) Fonksiyon bloğundaki program	İşlenen program
:JU FB 32 İSİM :TIME I5:I 0.0 I6:I 0.1 TIM5:T 5 TIM6:T 6 OFF:Q 1.0 7 :BE	:AN =I5 :A =I6 :L KT 005.2 :SFO =TIM5 :A =I5 :AN =I6 :L KT 005.2 :SSU =TIM6 :A =TIM5 :O =TIM6 := =OFF6 :A I 02 :RD =TIM5 :RD =TIM6 :BE	:AN I 0.0 :A I 0.1 :L KT 5.2 :SF T 5 :A I 0.0 :AN I 0.1 :L KT 5.2 :SS T 6 :A T 5 :O T 6 := Q 1.0 :A I 02 :R T 5 :R T 6 :BE

ÖRNEK=2

Fonksiyon Bloğunu Çağırma	(FB33)Fonksiyon bloğundaki program	İşlenen program
:JU FB 33 İSİM :SAYI I2: I 0.0 I3: I 0.1 I4: I 0.2 CNT5: C 5 OFF3: Q 1.0 :BE	:A =I2 :L KC 017 :SEC =CNT5 :A =I3 :SSU =CNT5 :A =I4 :SFD =CNT5 :A =CNT5 := =OFF3 :A I 0.3 :RD =CNT5 :BE	:A I 0.0 :L KC 0.17 :S C 5 :A I 0.1 :CU C 5 :A I 0.2 :CD C 5 :A C 5 := Q 1.0 :A I 0.3 :R C 5 :BE

VII-2-15-)"DO İSLEMİ

İŞLEM	İŞLEMCİ	AÇIKLAMA
-------	---------	----------

DO =	<input type="checkbox"/>	Kesin işlemciyi işleme: Yer değiştirilmiş bloklar şartsız olarak çağırılır.	
Kesin işlemci		İzin verilmiş asıl işlemci	Parametre tipi
		DB, PB, SB, FB ¹	B

ÖRNEK:

Fonksiyon Bloğunu Çağırma	FB35 Fonksiyon bloğundaki program	İşlenmiş program
STL	:DO =D5	:C DBI 5
:JU FB 35	:L =DW2	:L DW 2
NAME :DO	:DO =D6	:C DB 6
D5 : DB5	:T =DW1	:T DW 1
DW2: DW2	:T =Q4	:T QW 4
D6: DB6	:DO =MOT5	:JU FB 36
DW1: DW1	:BE	:BE
Q4: QW4		
MOT: FB36		
:BE		

VIII-3-) SİSTEM İŞLEMLERİ

Sistem işlemleri ve ilave işlemler aşağıdaki sınırlamalara sahiptirler.

* Bu işlemleri yalnızca fonksiyon bloklarda programlayabilirsiniz.

* Bu işlemleri yalnızca STL metod gösterimi ile programlayabilirsiniz.

Sistem işlemleri sistem verilerine bir pencere açması sebebiyle; yalnızca sistem bilgisine sahip kullanıcılar bu işlemleri kullanmalıdır. Eğer sistem işlemlerini programlamak istiyorsanız, programcı preset menüsündeki “ SYS:OPS.Y”i seçmelisiniz.

VIII-3-1-) SET İŞLEMLERİ

İlave bit işlemlerine benzer olarak; set işlemleri de bireysel (individual) bitleri değiştirebilir.

İŞLEM	İŞLEMCİ	AÇIKLAMA
SU	<input type="checkbox"/>	<u>Şartsız olarak bit set etme:</u> Sistem veri alanındaki belirli bir bit 1'e set edilir.
RU	<input type="checkbox"/>	<u>Şartsız olarak bit reset etme:</u>

		Sistem veri alanındaki belirli bir bit "0" a set edilir.
ID RS	Parametre	0.0 - 255.15

Set işlemlerinin işlenmesi RLO'dan bağımsızdır.

VIII-3-2-) YÜKLEME VE TRANSFER İŞLEMLERİ

Yükleme ve transfer işlemlerini programlanabilir kontrolörün giriş program belleğini adreslemek için kullanılır. Bu işlemler başlıca işlemciler tarafında adreslenemeyen bellek bölgeleri ve akümülatör arasındaki veri değişimi için kullanılır.,

İŞLEM	İŞLEMCİ	ÇIKIŞ	AÇIKLAMA
LIR		<input type="checkbox"/>	<u>Kayı indirekt olarak yükleme:</u> Bir bellek kelimesinin içeriği belirlenmiş kayda (ACCU 1,2) yüklenir. Adres ACCU-1'dedir.
TIR		<input type="checkbox"/>	<u>Kayı indirekt transfer etme:</u> Belirtilmiş kaydın içeriği bir bellek bölgesine transfer edilir. Adres ACCU-1'dedir.
			Parametre 0 (ACCU-1 için), 2 (ACCU-2 için)
TNB		<input type="checkbox"/>	<u>Bir veri bölgesini transfer etme:</u> Bir bellek alanı bölge olarak program belleğine transfer edilir. Son adres hedef bölgesi: ACCU 1 Son adres kaynak bölgesi: ACCU-2
T	<input type="checkbox"/>	<input type="checkbox"/>	<u>Transfer etme:</u> Bir kelime sistem veri alanına transfer edilir.
ID RS	Parametre		0 - 255

Kayı içeriklerini yükleme ve transfer etme:

Her iki akümülatör kayıtlar olarak adreslenir. Her bir kayıt 16 bit genişliğindedir. "LIR" ve "TIR" işlemleri veri kelimeleri olarak iletilmesi sebebiyle; kat-yıtlar çiftler halinde adreslenir. Yüklenen ve transfer edilen kayıt içerikleri RLO'dan bağımsızdır.

STL	AÇIKLAMA
L KH 6100 LIR 0	ACCU-1'e 6100 _H adresi yüklenir. Bilgi ACCU-1'e 6100 _H adresli bellek bölgesinden yüklenir.

Alan transferi metodlaması:

Alan transferi RLO'dan bağımsız olarak yapılır. Parametre transfer edilecek veri bölgesinin uzunluğunu gösterir. Alan 255 bayt uzunluğuna kadar olabilir. Kaynak olanın adresi ACCU-2'dedir. Hedef bölgesinin adresi ise ACCU-1'dedir.

ÖRNEK	Gösteriliş
12- baytlık bir veri alanını FOA2 _H adresinden EE90 _H adresine transfer etme.	
STL	Açıklama
:L KH FOA2 :L KH EE90 :TNB 12	Kaynak alanın son adresini ACCU-1'e yükle. Hedef alanın son adresini ACCU-1'e yükle. Kaynak adresi ACCU-2'ye kaydırılır. Veri alanını hedef alana transfer et.

Sistem Veri Alanını Transfer Etme:

Örnek: Her bir mod "STOP"tan "RW"a değiştirilmesinden sonra tarama monitör zamanı 100 ms'ye set edin. Bu zamanı sistem veri kelimesi -96'daki 10 ms'nin çarpımları şeklinde programlayabilirsiniz. Örnek için aşağıdaki fonksiyon bloğu OB21'den çağrılabilir.

STL	Açıklama
FB 11 L KF +10	Blok sayı ve tipi Faktör -10'lu ACCU-1'e yükle.,

T BE	RS	96	Bu deęeri sistem veri kelimesi -96'ya transfer et.
---------	----	----	--

VIII-3-3-) Aritmetik İşlemler

Bir aritmetik işlem belirlenmiş bir deęer tarafından ACCU-1'in içerięini deęiştirir. Parametre bu deęeri pozitif veya negatif ondalık sayı olarak temsil eder. Aşağıdaki tablo "ADD" işleminin başlıca niteliklerini gösterir.

İŞLEM	İŞLEMCİ	
ADD	<input type="checkbox"/>	<input type="checkbox"/> <u>Bit sabiti toplama:</u> Bayt veya kelime sabitlerini toplama.
ID		Parametre
BF		- 128: + 127
KF		- 32768: + 32767

İşleme:

Aritmetik bir işlem RLO'dan bağımsız olarak yapılır. Bu işlem RLO'yu veya şart kodlarını etkilemez.

Negatif bir parametre girerek çıkartma yapılabilir.

Sonuç 16 bitle temsil edemese bile; ACCU-2 iş görmez.

Örnek	STL	Açıklama
1020 _H sabitini 33 azaltın ve sonucu FW28 bayrak kelimesinde saklayın. Daha sonra sonuca 256 sabitini ekleyin ve toplamı FW30 bayrak kelimesinde saklayın.	L KH 1020 ADD BF -33 T FW 28 ADD KF 256 T FW 30	<ul style="list-style-type: none"> • -1020_H sabiti ACCU-1'e yüklenir. • -33_{OD} sabiti ACCU içerięine eklenir. • Yeni ACCU içerięleri (OFFF_H), FW28 bayrak kelimesinde saklanır. • 256_{OD} sabiti son sonuca eklenir. • Yeni ACCU içerięleri (10FF_H), FW30 veri kelimesinde saklanır.

VIII-3-4-) Diğer İşlemciler

İŞLEM	İŞLEMCİ	AÇIKLAMA
TAK		<u>Akümülatör içerięlerini deęiştirmek:</u> RLO'ya bakmaksızın ACCU-1 ve ACCU-2 içerięleri yer deęiştirir. RLO ve şart kodları bundan etkilenmez.
STS		<u>Hemen Durdurma:</u>

PLC RLO'ya bakmaksızın STOP moduna geçer.

“STS” işleminin işlemesi (çalışması):

“STS” işlemi yapıldığı zaman programlanabilir kontrolör hemen “STOP” moduna geçer. Bu noktada program işlemesi sonlandırılır. “ STOP” durumu yalnızca mansel olarak (mode seçici ile) veya programlayıcı fonksiyon “PC START” ile iptal edilebilir.

VIII-3-5-) ŞART KODU ÜRETİMİ

Programlanabilir kontrolörlerin işlemcisi aşağıdaki 3 şart koduna sahiptir:

* CC 0

* CC 1

* OV

Aşağıdaki işlemler şart kodlarını etkiler

* Karşılaştırma işlemleri

* Aritmetik işlemleri

* Kayma işlemleri

* Bazı değiştirme işlemleri

Şart kodlarının durumu çeşitli atlama işlemleri için bir şart temsil eder.

Karşılaştırma İşlemleri İçin Şart Kodu Üretme:

Karşılaştırma işlemlerinin yapılması şart kodları CC0 ve CC1'i set eder. Overflow şart kodu ise bundan etkilenmez. Karşılaştırma işlemleri RLO'yu etkiler. Karşılaştırma işlemi yapıldığında RLO “1”dir. Bu karşılaştırma işleminden sonra “JC” şartlı atlama işlemini kullanımına izin verilir.

ACCU-1'in içerikleriyle kayıtlanmış ACCU-2 içerikleri	Şart kodları			Mümkün Atlama İşlemleri
	CC 1	CC 0	OV	
Eşit	0	1		JZ
Daha Küçük	0	1		JN, JM
Daha Büyük	1	0		JN, JP

Aritmetik İşlemleri İçin Şart Kodları Üretme:

Aritmetik işlemlerin yapılması aritmetik işlemlerin sonucuna göre bütün şart kodlarını set eder.

Aritmetik işlem yapıldıktan sonraki sonuç	Şart kodları			Mümkün Atlama İşlemleri
	CC 1	CC 0	OV	
< - 32768	1	0	1	JN, JP, JO,
- 32768: -1	0	1	0	JN, JM
0	0	0	0	JZ
+1: + 32767	1	0	0	JN, JP
> + 32767	0	1	1	JN, JM, JO
(-) 65536*	0	0	1	JZ, JO

Dijital Lojik İçin Şart Kodu Üretme:

Dijital lojik işlemleri CC 0 ve CC 1'i set eder. Bu işlemler overflow şart kodunu etkilemez. İşlem yapıldıktan sonra set etme ACCU içeriklerinden bağımsızdır.

ACCU'ların içerikleri	Şart kodları			Mümkün Atlama İşlemleri
	CC 1	CC 0	OV	
Sıfır (KH =0000)	0	0		JZ
Sıfır değil	1	0		JN, JP

Kayma İşlemleri İçin Şart Kodu Üretme:

Kayıdırma işlemlerinin yapılması CC 0 ve CC 1'i set eder. Bu durum overflow şart kodunu etkilemez. Kod set etme dışarı kaydırılmış son bitin durumuna bağlıdır.

Dışarı kaydırılmış son bitin değeri	Şart kodları			Mümkün Atlama İşlemleri
	CC 1	CC 0	OV	
"0"	0	0		JZ
"1"	1	0		JN, JP

Değiştirme İşlemleri İçin Şart Kodu Üretme:

İki'nin tümününin teşekkülü bütün şart kodlarını set eder. Şart kodlarının durumları değiştirme fonksiyonlarının sonucuna göre şekillenir.

Aritmetik işlem	Şart kodları	Mümkün
-----------------	--------------	--------

yapılmasında sonraki sonuç				Atlama İşlemleri
	CC 1	CC 0	OV	
- 32768*	0	1	1	JN, JM, JO,
- 32767: -1	0	1	0	JN, JM
0	0	0	0	JZ
+1:+ 32767	1	0	0	JN,JP

VIII-4-) BASİT PROGRAMLAR

VIII-4-1-) Geçici-Kontak Röle/ Kenar değeri

Örnek	Devre Diyagramı
<p>I 0.0 girişindeki işaretin her bir yükselen kenarında, "AND" şartı "AI 0.0 ve AN F 64.0" yerine getirilir. RLO "1"dir.</p> <p>Bu F 64.0 ve F 2.0 bayrakları set eder. ("Kenar bayrakları")</p> <p>Bir sonraki işleme periyodunda AND şartı "A I 0.0 ve AN F 64.0" F64.0 bayrağı set edilemediği sebebiyle yapılamaz. 2.0 bayrağı resetlenir.</p> <p>Bu yüzden; F 2.0 bayrağı yalnızca program run için "1"dir.</p> <p>I 0.0 girişindeki anahtar devre dışı bırakıldığı zaman, F 64.0 bayrağı resetlenir.</p> <p>Bu resetleme I 0.0 girişindeki işaretin bir sonraki yükselen kenarının değeri için gerekli yolu hazırlar.</p>	

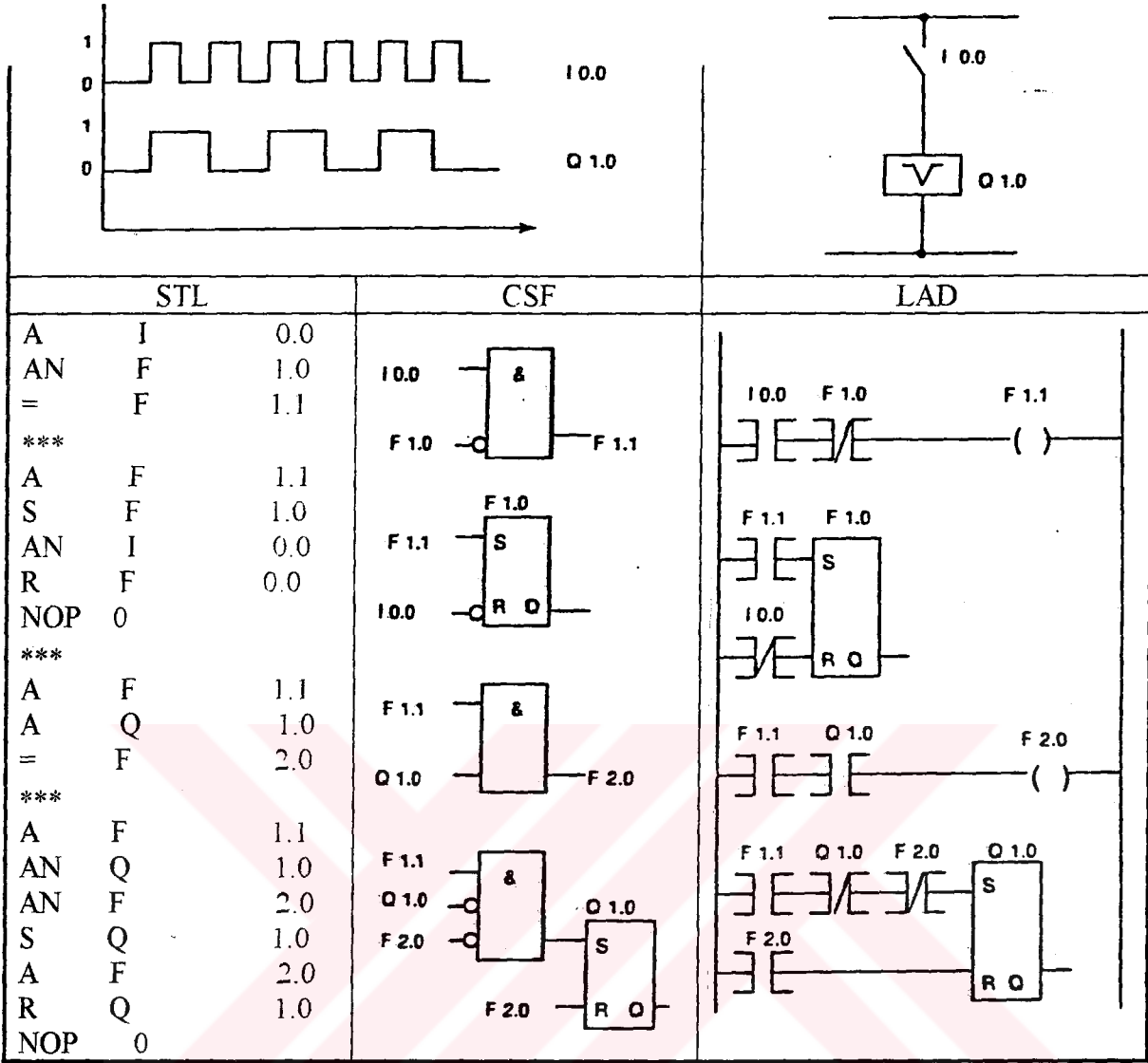
STL	CSF	LAD
A I 0.0 AN F 64.0 = F 2.0 AN I 0.0 R F 64.0 NOP 0		

VIII-4-2-) Binary Scaler/ Binary Bölücü:

Bu bölüm binary scaler' i nasıl programlayacağımızı anlatır.

Örnek= Binary scaler (Q1.0 çıkışı) I 0.0 işaret durumu 0'dan 1'e çıkışta (yükselen kenar) durumunu değiştirir. Bu yüzden giriş frekansının yarısı flip-flop'un çıkışında görülür.

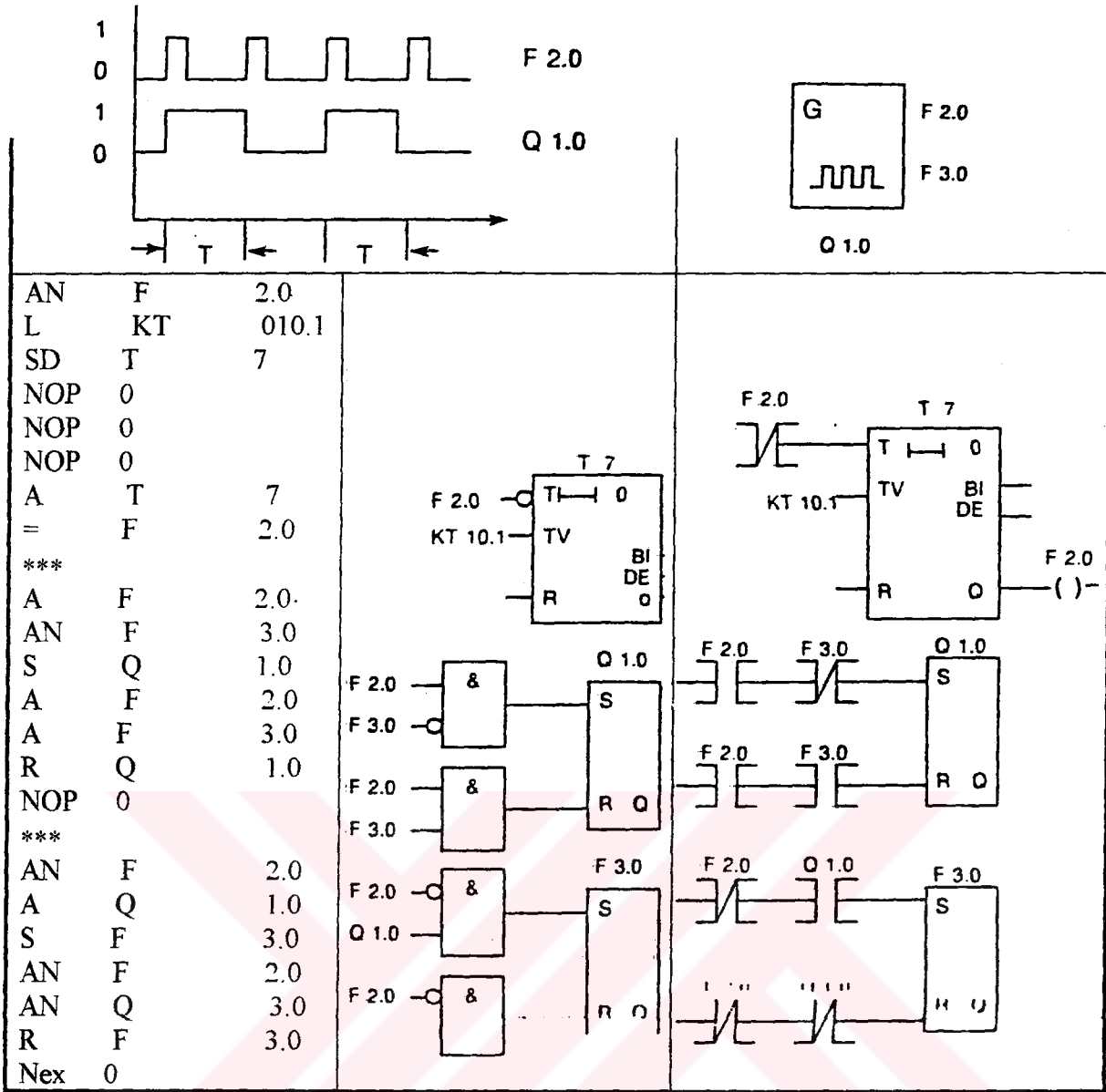
Zamanlama Diyagramı	Devre Diyagramı



VIII-4-3-) Saat/ Saat Pulse Jeneratörü

Örnek: Bir saat-pulse jeneratörü bir binary scaler tarafından takip edilen bir self-clocking timer kullanarak oluşturulabilir. F 20 bayrağı T7 timer'inin aşağıya doğru çalıştığı her seferde tekrar start eder. Örneğini F 2.0 bayrağı timerin aşağıya çalıştığı her seferde bir periyod için "1"dir. Binary scaler'e uygulanmış F 2.0 bayrağının pulse'leri Q 1.0 çıkışında pulse iş faktörü 1.1'li bir pulse treni oluşturur. Bu pulse treninin periyodu self-clocking timer'in zaman değerinin 2 katı kadardır.

Zamanlama Diyagramı	Devre Diyagramı



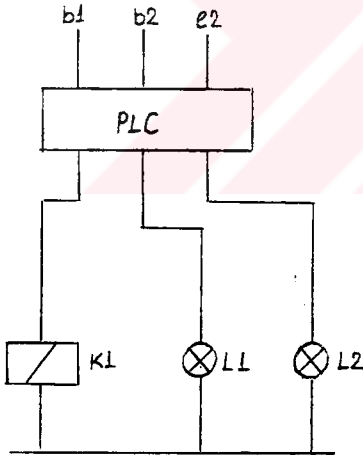
NOT: CSF ve LAD olarak çıkış ancak STL programlandığında segment sınırlar "***" girilirse alınabilir.

Bu PLC'lerin kontrol uygulamalarını içermektedir. Bir kontrol problemini çözmek için üç aşama takip edilir.

- Kontrol edilecek sistemin analizi.
- Kontrol elemanlarının tespit edilmesi.
- Kontrol programının yazılması.

Yukarıda belirtilmiş olan aşamaları takip ederek aşağıdaki uygulamalar Siemens -57 PLC'si ile yapılmıştır.

Uygulama 1) Bir motora start/stop butunlarıyla direkt yol verme



- b1 : Açma Butonu
- b2: Kapama Butonu
- K1: Motor Kontaktörü
- e2: Motor Koşuma Rölesi (Termik)
- L1: Çalışıyor Lambası
- L2: Arıza Lambası

Bu motoru kontrol etmek için aşağıda belirlenmiş giriş/çıkış noktasına gerek vardır.

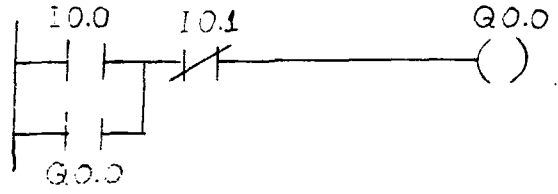
GİRİŞLER

- I.0.0="b1" açma butonu
- I.0.1="b2" kapama butonu
- I.0.2="e2" Termik arıza kapalı kontağı

ÇIKIŞLAR

- Q0.0="K1" Motor Kontaktör
- Q0.1="L1" Çalışıyor lambası
- Q0.2."L2" Arıza lambası.

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



Şebeke 2
 LD Q0.0
 = Q0.1



Şebeke 3
 LD I0.2
 = Q0.2



Şebeke 4
 MEND



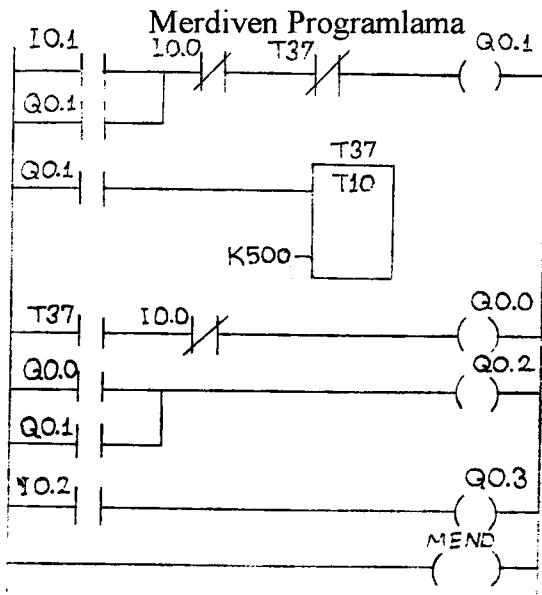
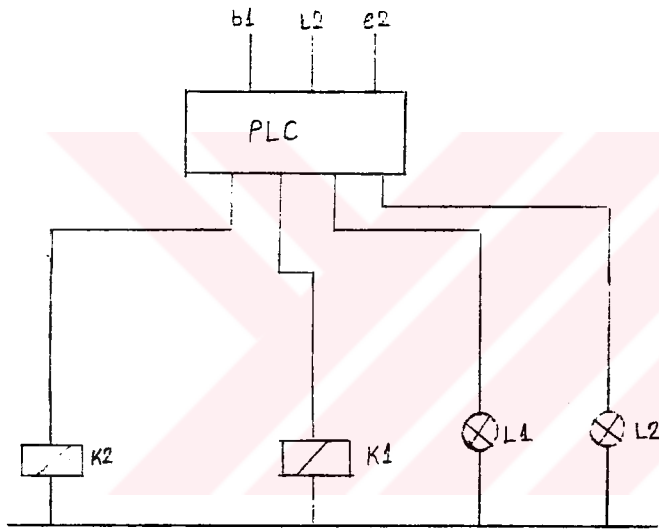
Uygulama 2) Bir motora yıldız/üçgen yolverme. Sistemde bulunan elemanları PLC'ye bağlanacakları giriş/çıkış noktalarına göre listelemesi aşağıda verilmiştir.

GİRİŞLER

I.0.0="b1" stop butonu
I.0.1="b2" start butonu
I.0.2="e2" Termik arıza

ÇIKIŞLAR

Q0.0= Yıldız çalışma kontaktörü
Q0.1=Üçgen çalışma kontaktörü
Q0.2."L1"Çalışıyor lambası.
Q0.3= "L2" Arıza lambası



STL programlama

Şebeke 1

```
LD I0.1
O Q0.1
AN I0.0
AN T37
= Q0.1
```

Şebeke 2

```
LD Q0.1
TON T37.500
```

Şebeke 3

```
LD T37
AN I0.0
= Q0.0
```

Şebeke 4

```
LD Q0.0
O Q0.4
= Q0.2
```

Şebeke 5

```
LD I0.2
= Q0.3
```

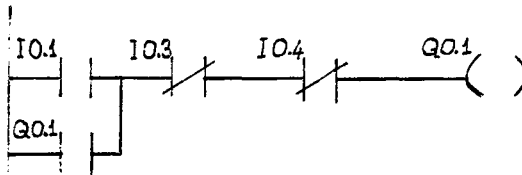
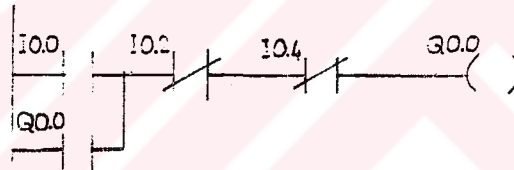
Şebeke 6

```
MEND
```


Uygulama 3) Bu uygulamada, bir boyanın deęişik renklerinin yapılması problemi üzerinde durulacaktır. Aşağıdaki şekilde gösterilen karıştırma tankına, üst tarafın iki alt tarafına ise bir adet boru hattı bağlanmaktadır. İki boru hattından iki deęişik boya ve alttaki boru hattından isi karışımı bitirilmiş boya tranfer edilir. Bunun için elimizde ki cihaz listesi ve giriş çıkış konumları aşağıda belirtilmiştir.

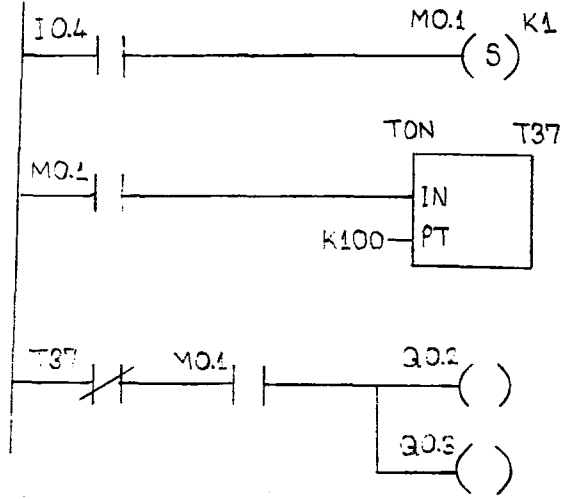
I.0.0=Pompa-1 start butonu	Q0.0= Pompa-1 kontaktörü
I.0.1=Pompa-2 start butonu	Q0.1=Pompa-2 kontaktörü
I.0.2=Pompa-1 stop butonu	Q0.2=Mixer motoru kontaktörü
I.0.3=Pompa-2 stop butono	Q0.3= Buhar vanası kontaktörü
I0.4= Yüksek seviye	Q0.4= Boşaltma vanası kontaktörü
I0.5= Düşük seviye	Q0.5= Boşaltma pompası

1 ve 2 nolu pompaların çalışma/durma ladder diyagram aşağıdaki gibidir.



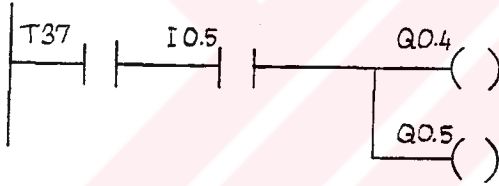
Görüldüğü gibi pompalar I0.0 ve I0.1 stop butonlarına basıldığı zaman çalışırlar. Durlmaları için ise ya I0.2, I0.3 butonlarına basılmalı ya da I0.4 yüksek seviye switch inde “dur” sinyali gelmesi gerekir.

Adım 2) Yüksek seviye switch’inden “doldu” işareti geldiğinde; boyaların “t” süresince ısıtılıp karıştırılmasına sıra gelir. “t” süresini 20 s’ye olacak ladder diyagramı aşağıdaki gibi yazılır.

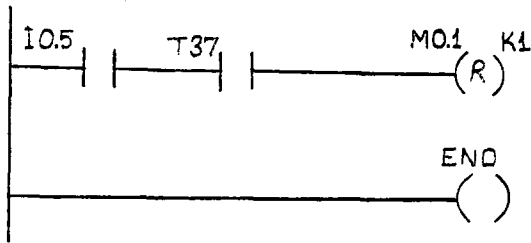


10 saniye süresince Miker motoru çalıştırılıp buhar vanası da açık bırakılır.

Adım 3) "10" saniye dolduğunda karışımı drain edilmelidir. Bu boşaltma düşük seviye swichlinden "dur" işareti gelene kadar devam eder. Bunun da ladder diyagramı aşağıdadır.



Adım 4) Son adımda ise program sonlandırılır.



Programın STL ile yazılımı ise aşağıdaki gibidir.

Şebeke 1
LD I0.0
O Q0.0
AN I0.2
AN I0.4
= Q0.0

Şebeke 2
LD I 0.1
O Q 0.1
AN I0.3
AN I0.4
= Q0.1

Şebeke 3
LD I0.4
S M0.1,1

Şebeke 4
LD M0.1
TON T37, 100

Şebeke 5
LDN T 37
A M0.1
= Q0.2
= Q 0.3

Şebeke 6
LD T37
AN I0.5
= Q0.4
=Q 0.5

Şebeke 7
LD I0.5
A 137
Z M0.1, 1

Şebeke8
MEND.

Uygulama 4)

Bu uygulamada yaya geçitlerinde bulunan trafik ışıklarının bantrolünün PLC ile yapılışı üzerinde durulacaktır. Programın çalışması şöyle açıklanabilir.

Adım -1) Yaya karşıdan karşıya geçmek için butonu bastığında araçlar için sarı ışık 10 saniye yaracaktır.

Adım 2) Araçlara kırmızı ışık 30 saniyelik süre için yanacaktır.

Adım-3) Yayalara yeşil ışık 30 saniyelik süre için yanacaktır.

Adım 4) Yayalara 30 saniye sonra kırmızı ışık yanarlar.

Adım 5) Yayalara 30 saniye sonra kırmızı ışık yandıktan 10 saniye sonra araçlara 10 saniye süre için sarı ışık yanacak.

Adım 6) Bütün bu süre sonunda yayalara yeşil ışık yanacak.

I0.0 = Yayaların karşıdan karşıya geçmek için bastıkları switch

Q0.0= Yeşil trafik ışığı

Q0.1= Sarı trafik ışığı

Q0.2= Kırmızı trafik ışığı

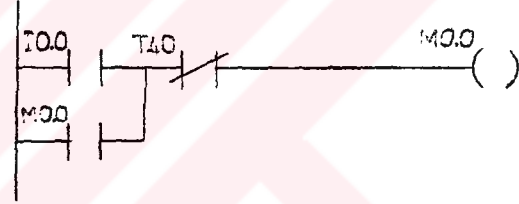
Q0.3= Yayalar için yeşil ışık

Q0.4= Yayalar için kırmızı ışık

STC

Şebeke 1

LD I0.0
O M0.0
AN T4.0
= M0.0



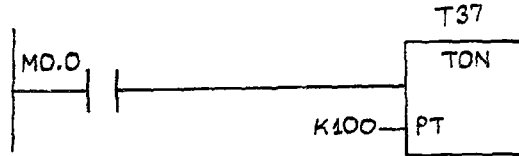
Şebeke 2

LND M0.0
TON Q0.0



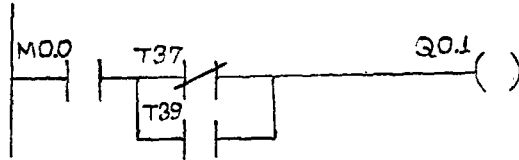
Şebeke 3

LD M0.0
TON T37.100



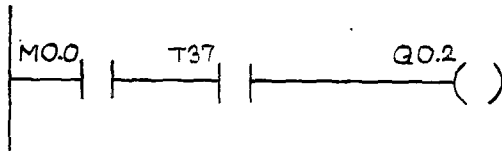
Şebeke 4

LD M0.0
LDN T37
O T37
ALO
= Q0.1

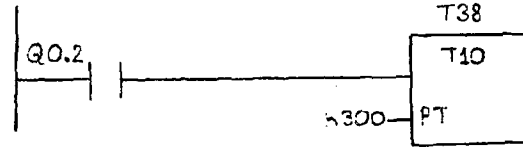


Şebeke 5

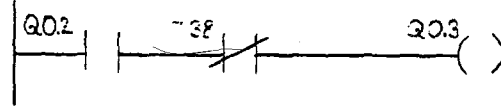
LD M0.0
A T37
= Q0.0



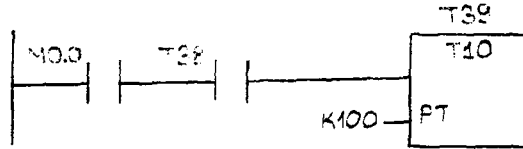
Şebeke 6
LD Q0.2
TON T38,300



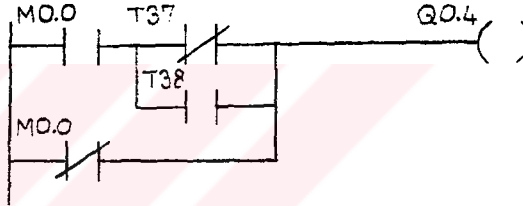
Şebeke 7
LD Q0.2
AN T38
= Q0.3



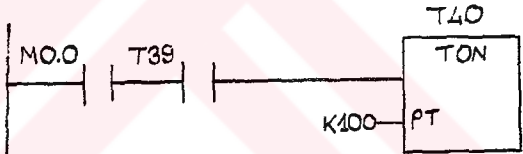
Şebeke 8
LD M0.0
A T38
TON T39,100



Şebeke 9
LD M0.0
LDN T37
Q T38
ALD
ON M0.0
= Q0.4



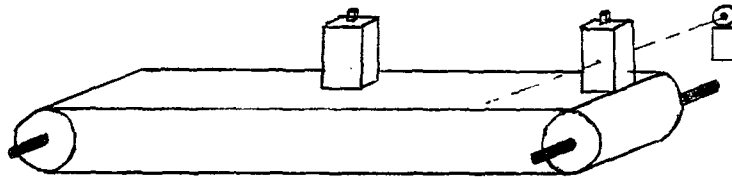
Şebeke 10
LD M0.0
A T39
TON T40, 100



Şebeke 11
MEND



Uygulama 5) Bu uygulamada bir konvayer üzerinde giden 100 adet şişeyi sayan bir sistemin programı üzerinde durulacaktır. Şişe sayısı 100'e ulaştığında motor otomatikman duracaktır.



Yukarıdaki şekilden yola çıkarak giriş ve çıkışlar belirlenirse;

I0.0= Konteyner motors start butons = b1

I0.1 = Fotosel 1-şişeyi gördü

I0.2 = Konteyner motoru stop butonu = b2

I0.3= Termik arıza

Q0.0= Konteyner motoru kontaktörü

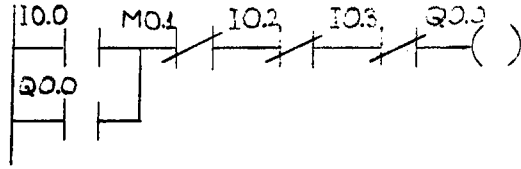
Q0.1 = Şişe sayısı 100 oldu lambasını yak.

Q0.2= Termik arıza lamsanı yak.

Bu bilgilerle PLC programını LAD ve STL olarak yazılırsa.

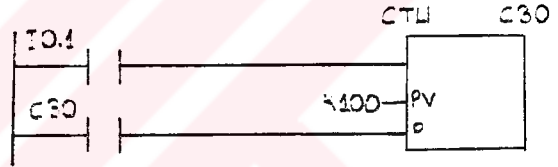
Şebeke 1

```
LD I0.0
O Q0.0
AN M0.1
AN I0.2
AN I0.3
= Q0.0
```



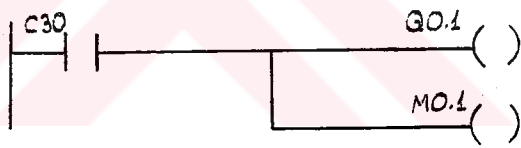
Şebeke 2

```
LD I0.1
LD C01
CTU CO1,100
```



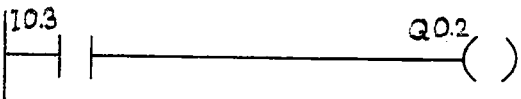
Şebeke 3

```
LD C01
= Q0.1
= M0.1
```



Şebeke 4

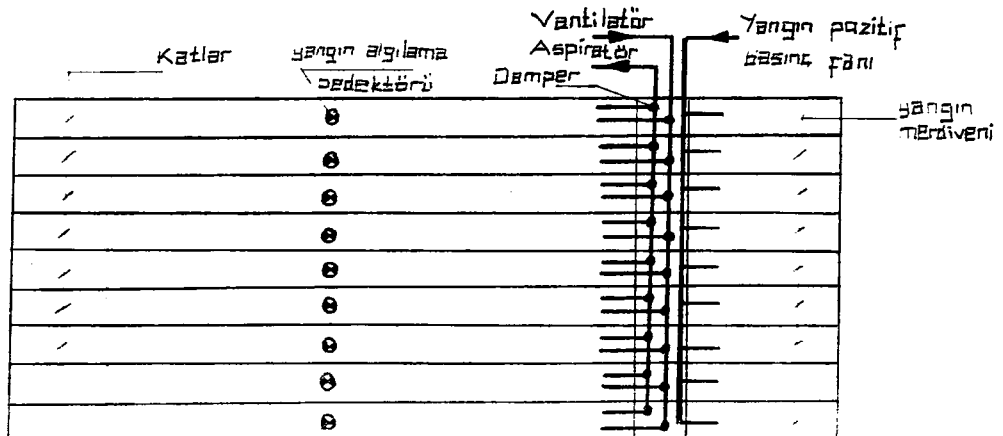
```
LD I0.3
= Q0.2
```



Uygulama 6)



Bu uygulamada 9 katlı bir otelin havalandırma sisteminin bir yangın anında nasıl davranacağı üzerinde durulacaktır. Sistem şeması aşağıda gösterilmiştir.



Yangın algılama dedektörleri 9 kata 8-zone olarak sistemlenmişlerdir. Her katta taze hava ve egzoz sistemlerinde otomatik kumanda edilebilir damperr motorları kullanılmıştır. Buna göre sistemin yangın anında çalışması şöyle incelenebilir.

Herhangi bir kattaki yangın algılama zonunda bir yangın ihbarı alındığında o kata ait damper direkt olarak kapatılıp servis merdivenini basınçlandırarak gen dağıtılarak katta kalanların bu merdiven sayesinde tahliye edilebilmeleri mümkün kılınacaktır. Ve yangın ihbarı verildiği anda taze hava ve egzoz fonları da otomatikman kapatılacaktır. Buna göre giriş çıkış noktaların yazarsak;

I0.0= Ventilator motoru start butonu = b1

I0.1= Aspirator motoru start butonu=b3

I0.3= Yangın pozitif basınç fon motoru start butonu = 65

I0.4= Ventilator motoru stop butonu = b2

I0.5= Aspirator moturu stop butonu = b4

I0.6= Yangın pozitif basınç fon motors stop butonu = b6

I0.7 = 1.Kat yangın zonu

I1.0=2.Kat yangın zonu

I1.1 = 3.Kat yangın zonu

I1.2= 4. Kat yangın zonu

I1.3 =5. Kat yangın zonu

I1.4= 6.Kat yangın zonu

I1.5 = 7.Kat yangın zonu

I1.6= 8.Kat yangın zonu

I1.7= 9.Kat yangın zonu

Q0.0= Ventilator motoru kontaktörü

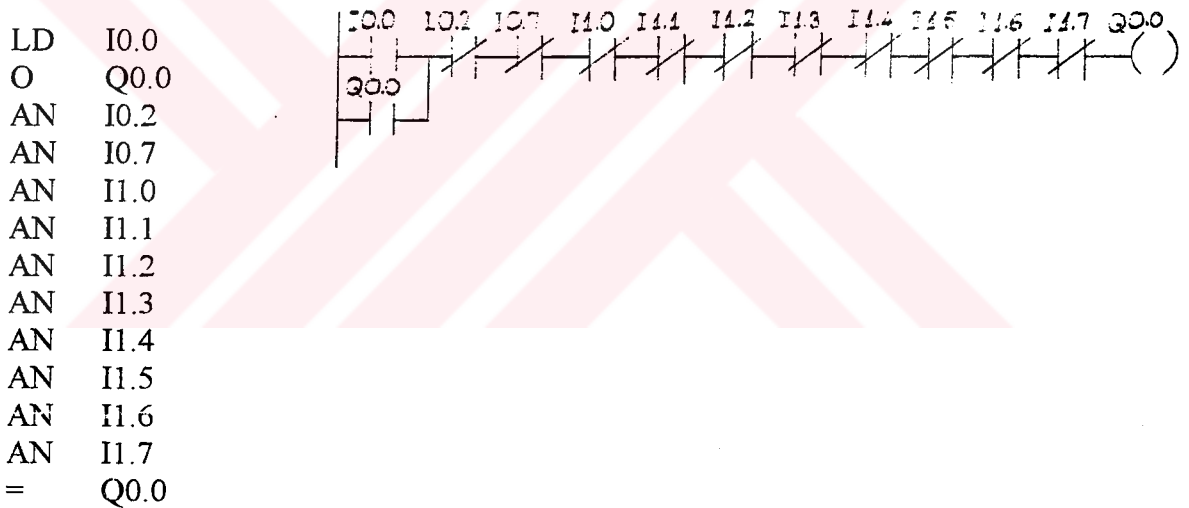
Q0.1= Aspirator motoru kontaktörü

Q0.2= Yangın pozitif basınç fanı motoru kontaktörü

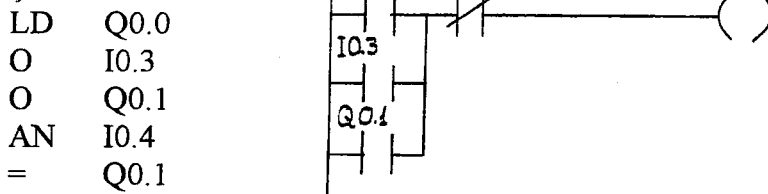
- Q0.3= 1. Kat damper motoru
- Q0.4= 2. Kat damper motoru
- Q0.5= 3. Kat damper motoru
- Q0.6= 4. Kat damper motoru
- Q0.7= 5. Kat damper motoru
- Q1.0= 6. Kat damper motoru
- Q1.1= 7. Kat damper motoru
- Q1.2= 8. Kat damper motoru
- Q1.3= 9. Kat damper motoru

STL

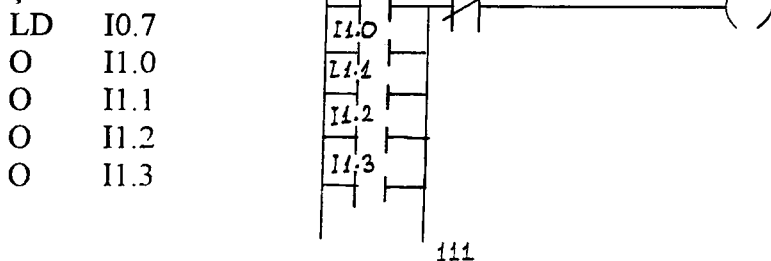
Şebeke 1



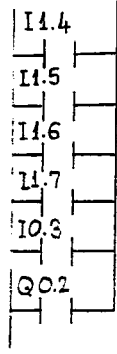
Şebeke 2



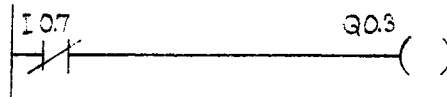
Şebeke 3



O I1.4
 O I1.5
 O I1.6
 O I1.7
 O I0.3
 O Q0.2
 AN I0.6
 = Q0.2



Şebeke 4
 AN I0.7
 = Q0.3



Şebeke 5
 AN I0.1
 = Q0.4



Şebeke 6
 AN I1.1
 = Q0.5



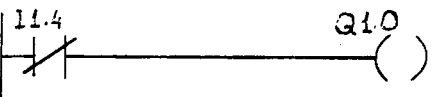
Şebeke 7
 AN I1.2
 = Q0.6



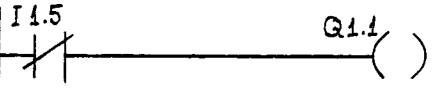
Şebeke 8
 AN I1.3
 = Q0.7



Şebeke 9
 AN I1.4
 = Q1.0



Şebeke 10
 AN I1.5
 = Q1.1



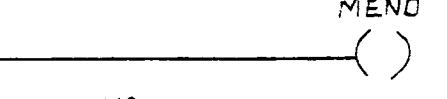
Şebeke 11
 AN I1.6
 = Q1.2



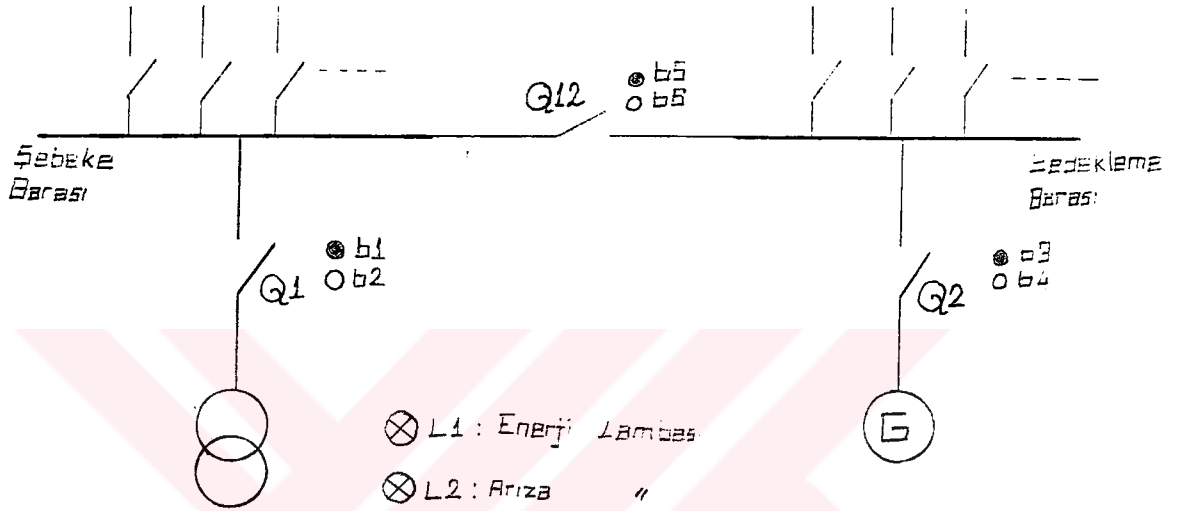
Şebeke 12
 AN I1.7
 = Q1.3



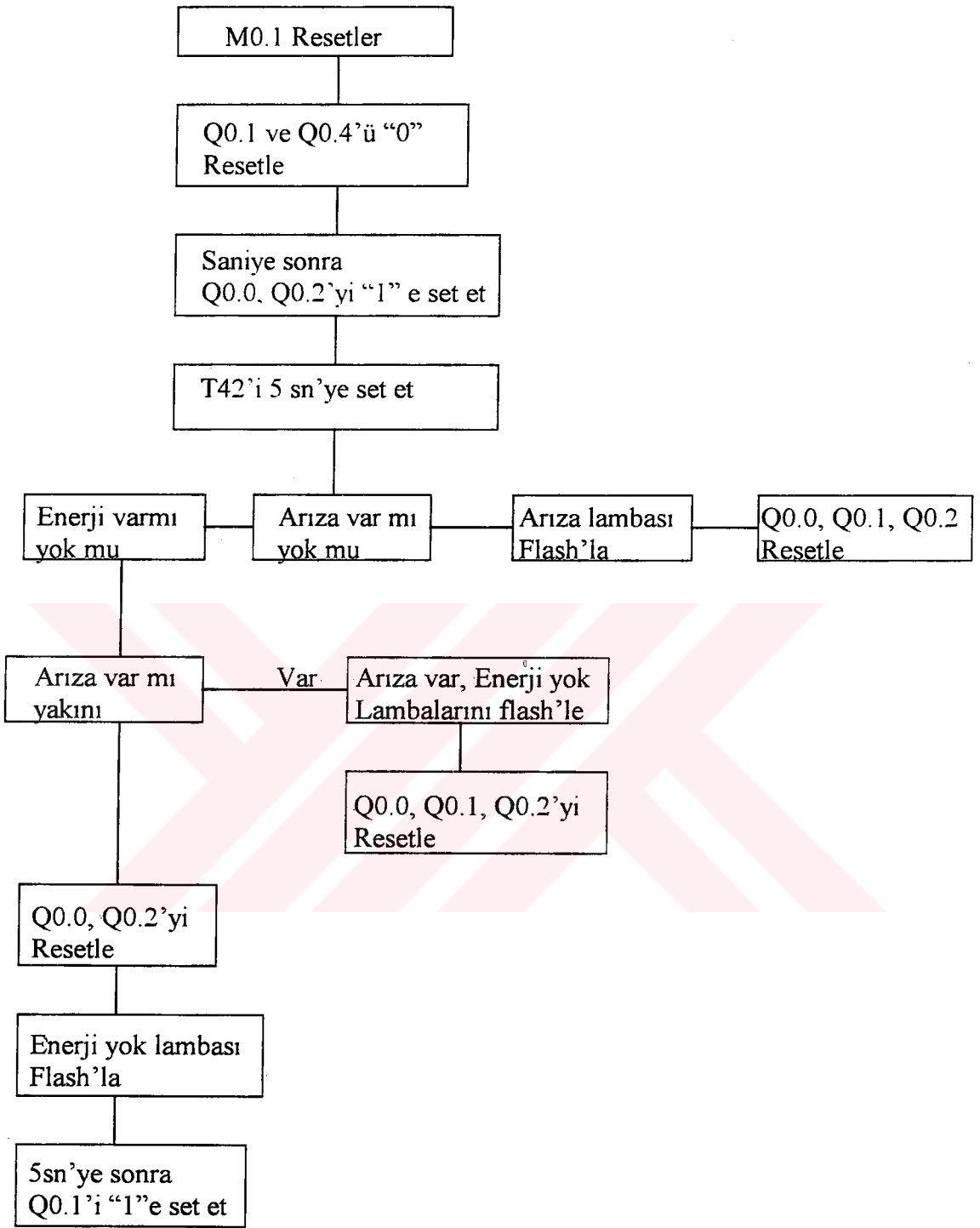
Şebeke 13
 MEND.



Uygulama 7) Bu uygulamada bir enerji besleme merkezinde (sanayi merkezi, bina, fabrika vb) yedekli çalışmanın PLC ile kontrolü üzerinde durulacaktır. Sistemin prensip şeması aşağıdaki gibidir.



Şemadan görüldüğü gibi Q1, Q2, Q12 kompakt şalterlerdir ve açma/kapama bobinleri sayesinde devreye alınıp, çıkarılabilirler. Enerji varken Q1 ve Q12 şalterleri devrededir. Ve tüm sistem trafo üzerinden beslenir. Enerji kesildiği anda L1 enerji yok lambası yanar ve Q1, Q12 şalterleri devreden çıkar. Aynı anda jeneratör devreye çalışarak “t” saniye sonra yedeklenecek gücü üzerine alır. Herhangi bir arıza anında tüm şalterler devre dışı bırakılıp “L2” lambası arıza meydana geldiğini belirtir. Sistemle ilgili bu kısa bilgiler verildikten sonra sistemin çalışma prensibinin akış şeması yazılabilir.



Yukarıdaki şemadan hareket ederek sistemin PLC program Simatic S7200 ile yazırsa;

GİRİŞLER

I0.0 = "11" Enerji Var/yok bilgisi
I0.1 = "12" Arıza Var/yok bilgisi
I0.2 = "b1" Butonu
I0.3 = "b2" "
I0.4 = "b3" "
I0.5 = "b4" "
I0.6 = "b5" "
I0.7 = "b6" "

ÇIKIŞLAR

Q0.0 = Q1 Şalteri
Q0.1 = Q2 "
Q0.2 = Q12 "
Q0.3 = Enerji yok
Q0.4 = Arıza var

STL Program

Şebeke 1
LD I0.0
JMP K1

Şebeke 2
LD I0.1
JMP K2

Şebeke 3
LDN I0.0
LDN I0.3
O I0.2
ALD
S M0.2 K1
TON T42 K50

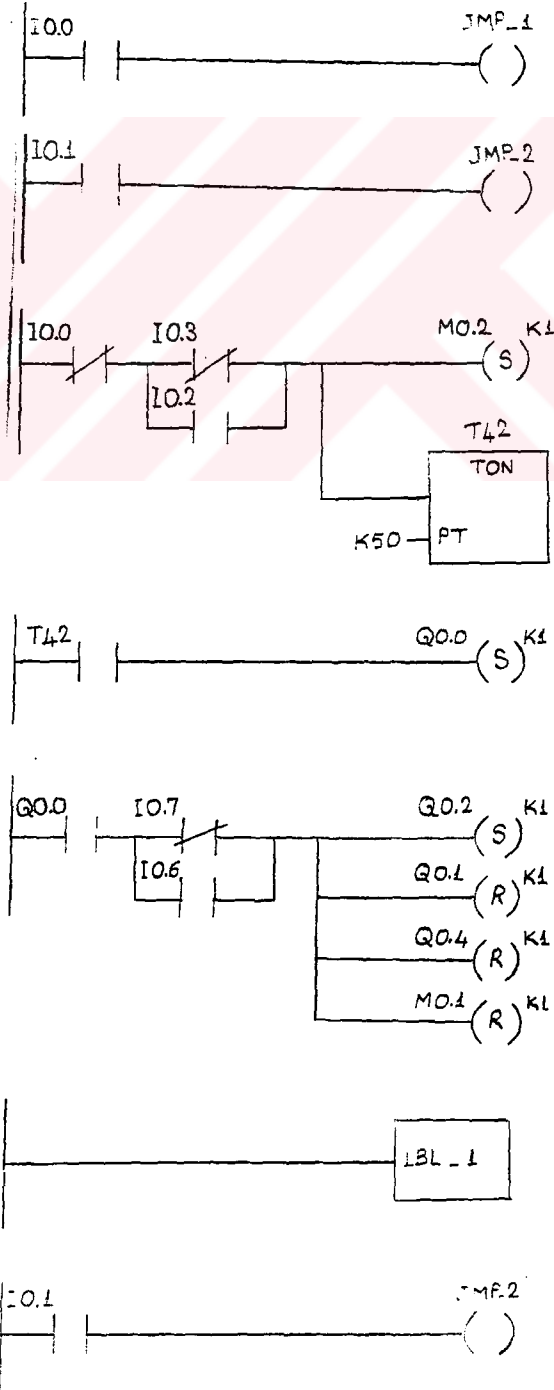
Şebeke 4
LD T42
S Q0.0 K1

Şebeke 5
LD Q0.0
LDN I0.7
O I0.6
ALD
S Q0.2 K1
R Q0.1 K1
R Q0.4 K1
R M0.1 K1

Şebeke 6
LBL K1

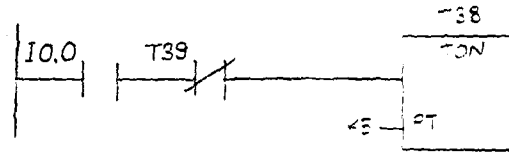
Şebeke 7
LD I0.1
MP K2

Merdiven Programı



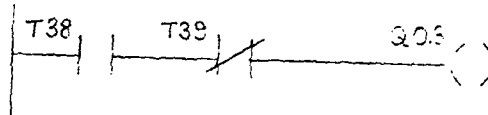
Şebeke 8

LD I0.0
AN T39
TON T38 K5



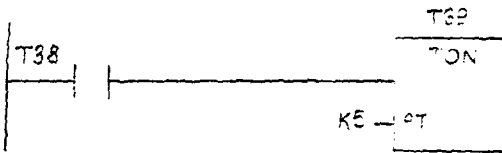
Şebeke 9

LD T38
AN T39
= Q0.3



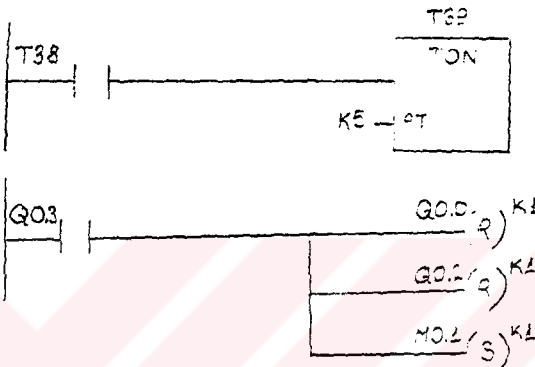
Şebeke 10

LD T38
TON T39 K5



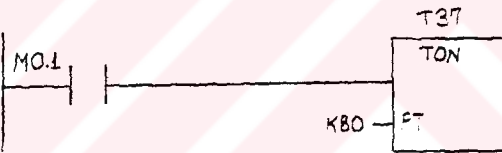
Şebeke 11

LD Q0.3
R Q0.0 K1
R Q0.2 K1
S M0.1 K1



Şebeke 12

LD M0.1
TON T37 K80



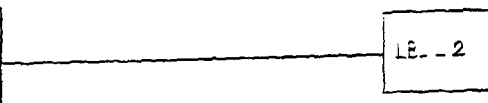
Şebeke 13

LD T37
S Q0.1 K1



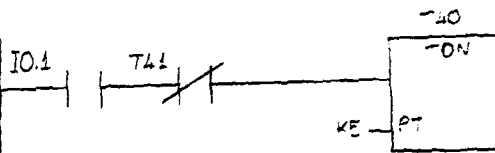
Şebeke 14

LBL K2



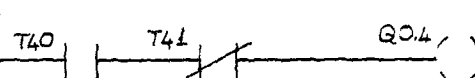
Şebeke 15

LD I0.1
AN T41
TON T40 K5



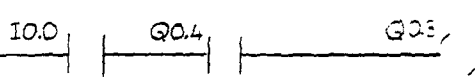
Şebeke 16

LD T40
AN T41
= Q0.4



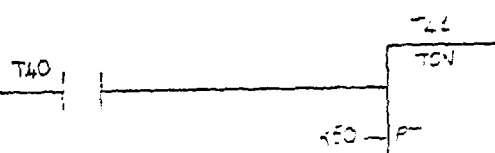
Şebeke 17

LD I0.0
A Q0.4
= Q0.3

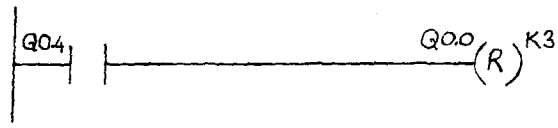


Şebeke 18

LD T40
TON T41 K50



Şebeke 19
LD Q0.4
R Q0.0 K3



Şebeke 20
MEND



SONUÇLAR

Programlanabilir kontrol cihazı (PLC), belleğinde saklı bir program ve çeşitli giriş/çıkış elemanları ile bir makina veya sürecin denetlemesini sağlayan katı hal elemanıdır. Yeni teknolojilerin endüstriyel tesislere ve modern tesislere sağladığı bir kolaylık olan bu cihazlar, sağladıkları esnek denetim, kolay tesis edilme, az yer kaplama, kolaylıkla değişen konumlara uyabilme ve bakım kolaylığı gibi birçok yararları nedeniyle hemen hemen her sahada, çok yaygın bir şekilde kullanılmaktadır. Ülkemiz de özellikle yeni kurulan modern tesislerde bu cihazlar yer almakta, daha eski tesislerde ise eski elektromagnetik kontrol elemanları yerine bunların kullanılması planlanmaktadır.

Yeni tesislerde ve endüstride çalışan mühendis ve yöneticisi, otomize edilen imalat ve süreç kontrol sistemlerinin bütün elemanlarına ilişkin çalışma ve uygulama üzerinde detaylı bilgiye gerek duyar. Bu tez ile PLC alanında bu gereksinim karşılanamaya çalışılmaktadır. Modern bilgisayar kontrollü düzenlerin bir çoğu, büyük ihtimalle, sistemi oluşturan teknolojik yapıya ilişkin ayrıntılı bilgisi olmayan bir eleman tarafından çalıştırılır. Bu hususta PLC'ler, yardımcısız kullanım açısından önemli bir yer tutmaktadır.

Bu çalışmada, esas olarak PLC'ler tanıtılmakta, giriş/çıkış arabirim devreleri ile kullanılan arabirim elemanları açıklanmakta ve kesitli tesislerde karşılaşılabilecek sistemlerin PLC ile kontrolünün nasıl yapıldığını gösteren örneklere yer verilmiştir.

YARARLANILAN KAYNAKLAR

1. Crispin, A.J, 1990, “Programmable Logic Controllers and Their Engineering Applications”, McGraw-Hill.
2. Simatic , 1995, “SIMATIC S5, S5904, S5-100u Programmable Controllers”, Siemens Catalog.
3. Simatic, 1995, “SIMATIC S7, S7200 Programmable Controllers” Siemens Catalog.
4. Brayn, L.A., 1988, “Programmable Controllers Theory and Implementation”, Industrial Text.
5. Warnock, I.G., 1988, “Programmable Controllers Operation and Application”, McGraw-Hill.
6. Voltom Teknik Elektrik, 1993, “Yaklaşım Anahtarları ve Opto-Elektronik Sensörleri”, Seminer Notları.
7. Greshock, K., W.J., 1993. “Industrial Control Electronics”, MacMillan Publishing, ISBN: 0-02-424864.
8. Kaynak , O., 1988, “Programlanabilir Denetleyiciler”, Boğaziçi Üniversitesi Müh. Fak.

ÖZGEÇMİŞ

1971 yılında Adapazarı'nda doğdu. 1988 yılında girdiği Y.T.Ü. Mühendislik Fakültesi Elektrik Mühendisliği bölümünden 1992 yılında Elektrik Mühendisi olarak mezun oldu. Aynı yıl Y.T.Ü. Fen Bilimleri Enstitüsü Elektrik Mühendisliği Bölümü'nde yüksek lisans programına girmeye hak kazandı. 1 yıllık master ingilizce hazırlık programını bitirdikten sonra yüksek lisans çalışmalarına başladı. Halen özel bir firmada Elektrik kontrol mühendisi olarak çalışmaktadır.