

56057

T.C.

İSTANBUL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

PLC İLE ELEKTRONİK DEVRELERİN

OTOMATİK KONTROLÜ

YÜKSEK LİSANS TEZİ

Atilla ÖZMEN

Elektronik Mühendisliği Anabilim Dalı

(Elektronik Programı)

Danışman : Prof. Dr. Ergür TÛTÛNCÛOĐLU

MAYIS - 1996

ÖNSÖZ

PLC' ler endüstriyel otomasyon ve kontrol sistemleri için geliştirilmiş olan modüler giriş/çıkış birimi ile merkezi işlem biriminden meydana gelmiş mikrobilgisayarlardır. Bu çalışmada Siemens S5 90U PLC' si kullanılarak analog dijital çevirici tasarımı anlatılmış ve aynı PLC ile verilen bir örnek problem üzerinde kombinasyonel lojik devre tasarımına değinilmiştir. Analog dijital çevirici tasarımı programlama, iki ayrı metot da yapılmıştır.

Bu çalışmanın PLC konusunda çalışacak arkadaşlara yararlı olmasını umut eder ve böylesine güncel bir konuda tez almamı sağlayan ve çalışmalarım sırasında yardımlarını esirgemeyen, değerli Hocam Prof. Dr. Ergür TÛTÛNCÛOĐLU' na teşekkür ederim.



İÇİNDEKİLER

ÖNSÖZ.....	I
İÇİNDEKİLER.....	II
ÖZ / ABSTRACT.....	V
I. GİRİŞ.....	1
1.1. Programlanabilir Lojik Kontrolödiciler.....	1
1.2. Tarihçe.....	2
1.3. PLC' lerin Üstünlükleri.....	3
II. PROGRAMLANABİLİR LOJİK KONTROLÖDİCİLER' İN YAPISI	4
2.1. Merkezi İşlem Birimi.....	5
2.2. Hafıza	7
2.2.1. Hafıza Çeşitleri.....	9
2.2.2. Hafıza Haritaları.....	10
2.3. Giriş Çıkış Birimleri.....	11
2.3.1. Giriş Birimleri.....	11
2.3.2. Çıkış Birimleri	12
2.4. Analog Giriş/Çıkış Birimleri	13
2.4.1. Analog Giriş.....	14
2.4.2. Analog Çıkış.....	14
2.5. Özel Giriş/Çıkış Birimleri.....	15
III. PLC İLE PROGRAMLAMA	17
3.1. Giriş	17
3.2. Tarama Zamanı.....	18
3.3. Programlama Çeşitleri.....	18
IV. STEP 5 PROGRAMLAMA DİLİ.....	19
4.1. Giriş.....	19
4.2. Programlamada Kullanılan Bloklar	19
4.2.1. Organizasyon Blokları	19

4.2.2. Program Blokları.....	20
4.2.3. Fonksiyon Blokları	20
4.2.4. Data Blokları.....	20
4.2.5. Sıralı Bloklar	20
4.3. Step 5 Programlama Dilinde Kullanılan Operandlar	21
4.4. Step 5 Programlama Dilinde Kullanılan Komutlar ve Programlama	23
4.4.1. Lojik İşlemler	24
4.4.1.1. AND İşlemi.....	24
4.4.1.2. OR İşlemi.....	24
4.4.1.3. OR' dan Önce AND İşlemi.....	25
4.4.1.4. AND' den Önce OR İşlemi.....	25
4.4.2. RLO ve Akümülatör.....	26
4.4.3. Set Reset işlemleri	27
4.4.4. Zamanlayıcılar.....	27
4.4.4.1. Pulse Timer (SP).....	29
4.4.4.2. Extended Pulse Timer (SE)	30
4.4.4.3. On Delay Timer (SR)	30
4.4.4.4. Stored On Delay Timer (SS)	31
4.4.4.5. Off Delay Timer (SF).....	31
4.4.5. Sayıcılar	32
4.4.6. Bit Test İşlemleri.....	33
4.4.6.1 Bit Test İşlemlerinde Kullanılan Komutlar.	33
4.4.7. Byte ve Word Komutları,	33
4.4.7.1. Yükleme Komutları.....	33
4.4.7.2. Transfer Komutları.....	34
4.4.7.3. Karşılaştırma Komutları.....	34
4.4.7.4. Aritmetik İşlemlerde Kullanılan Komutlar.....	35
4.4.7.5. Lojik Komutlar (AW, OW, XOW)	35
4.4.7.6. Kaydırma Komutları.....	36
4.4.7.7. Tümlleme İşlemleri.....	36
4.4.8. Diğer Komutlar	37
4.4.8.1. Arttırma ve Azaltma Komutları	37
4.4.8.2. Dalkanma Komutları	37

V. ANALOG / SAYISAL ÇEVİRİCİLER	38
5.1. Analog ve Sayısal İşaretler.....	38
5.2. Paralel Karşılaştırıcı veya Flaş A/D Çeviriciler	39
5.3. Tek Eğimli A/D Çeviriciler	40
5.4. Sayıcı Türden A/D Çeviriciler	42
5.5. İzlemeli Türden A/D Çeviriciler.....	43
5.6. Ardışıl Yaklaşımlı A/D Çeviriciler.....	44
VI. BULGULAR	45
5.1. PLC İle ADC uygulaması.....	45
5.1.1. Devrenin Ardışıl Yaklaşımlı ADC Olarak Çalışacak Şekilde PLC' nin Programlanması	47
5.1.2. Devrenin Sayıcı Türden ADC Olarak Çalışacak Şekilde PLC'nin Programlanması.....	50
5.2. PLC ile Kombinasyonel Lojik Devre Tasarımı.....	52
5.2.1. Problem	53
VII. TARTIŞMA VE SONUÇ	55
VIII. ÖZET / SUMMARY	57
IX. KAYNAKLAR	59
X. EKLER	60
X1. ÖZGEÇMİŞ	79

ÖZ

PLC İLE ELEKTRONİK DEVRELERİN OTOMATİK KONTROLÜ

Çalışmanın ilk bölümlerinde PLC' lerin yapıları ele alınarak röleli kontrol panellerine göre üstünlüklerinden bahsedilmiştir. Uygulamalara kadar olan bölümlerinde ise PLC ile programlama anlatılarak STEP 5 programlama dili ayrıntılı olarak incelenmiştir.

Çalışmanın uygulamalar bölümünde, iki farklı uygulama ele alınmış ve ilk uygulamada SIEMENS S5 90U PLC' si ile analog/dijital çevirici gerçekleştirilerek sadece PLC yazılımının değiştirilmesiyle, gerçekleştirilen devrenin "ardışıl yaklaşımlı" ve "sayıcı türden" olmak üzere iki farklı şekilde çalışması sağlanmıştır. Diğer uygulamada ise yine aynı PLC kullanılarak kombinyasyonel lojik devre tasarımı anlatılmıştır.

Analog/Dijital çevirici için LADDER programlama metodu ile STL metodu kullanılarak, her iki metodun birbirlerine karşı avantajlarının ortaya konmasına çalışılmıştır.

ABSTRACT

AUTOMATIC CONTROL OF ELECTRONIC CIRCUITS WITH PLC

At the beginning of this study, the structures of the PLCs are mentioned and their advantages to the relay controlled panels are explained. Before the chapter of applications the PLC programming methods are presented and STEP5 programming language is introduced in detail.

In the chapter of applications, first of all two different types of analog to digital converters, namely "Successive Approximation" and "Counter" are realized by changing only the program lists for SIEMENS S5 90U PLC. In the other application a combinational logic circuit realization with the same PLC is given.

For the ADC, the LADDER programming method and the STL programming method are used to compare in order to observe the advantages and disadvantages to each other.

I. GİRİŞ

1.1 Programlanabilir Lojik Kontrolödiciler

Bir üretim veya kontrol işleminde başarıya ulaşabilmek için yapılacak olan işlemlerin belli sraya göre ve zamanına uygun olarak yapılması gerekmektedir. Bunu sağlamak için de sistemdeki değişkenlerin, hangi işlemi ne zaman ve nasıl yapması gerektiğinin belirlenmesi için, sürekli gözetim altında tutulması gerekmektedir. Süreç denetimi adı verilen bu işlem, insanlar tarafından gerçekleştirilebileceği gibi, otomatik olarak da gerçekleştirilebilir. Otomatik süreç denetimi sırasında asıl olan, herhangi bir değişkenin değerine göre çeşitli mantıksal değerlendirmeler sonucunda kontrol sırasında kullanılan çeşitli giriş/çıkış elemanlarının açılıp kapatılmasıdır. İşte, bu süreç denetimi sırasında, bir mikroişlemciyle, modüler bir giriş/çıkış sisteminden oluşan ve bu yapıyla küçük bir bilgisayarı andıran PLC' ler oldukça sık kullanılan kontrol elemanları arasındadırlar.

Geleneksel kontrol panellerinde yüzlerce ve çoğu zaman binlerce elektromekanik geciktiriciler kullanılır ve ayrıca bu geciktiricilerin birbirleriyle olan bağlantı kabloları da binlerce metrelik karmaşık bir yapı oluşmasına neden olurdu. Bu bahsettiğimiz kontrol panelleri, bir veya birkaç tane rölenin bozulmasına kadar kararlı bir şekilde çalışırlardı ki; bozulan bir rölenin teşhis edilmesi de çoğu zaman oldukça güçlü. Üretim işleminin tamamen değiştirilmesi durumunda ise kontrol panelinde de buna uygun değişimler yapılırdı. Bunun yanı sıra farklı bir işlem için elde bulunan panelin yeniden çalıştırılmak üzere adapte edilmesi işlemi de oldukça pahalıydı. Bu yüzden eski panelin yeniden adapte edilmesi yerine daha yeni başka bir panelle değiştirilirdi. İşte, bu olumsuz yönlerden dolayı daha kararlı, hata teşhisi daha kolay yapılabilecek ve oldukça esnek olan, yazılıma dayalı lojik kontrol sistemlerinin kullanılması gereği ortaya çıktı. Böylece PLC konusundaki ilk düşünceler ortaya atılmış oldu. PLC konusundaki bu düşünceler ilk olarak Detroit-Michigan'daki bir otomobil fabrikasında ortaya atılmıştır. Bu sistemlerde yeni kontrol işlemlerini sağlamak için yazılımı değiştirmek yeterliydi ve bu da kontrol panelini değiştirmekten çok daha kolaydı.

Detroit'deki bu kullanımdan sonra programlanabilir denetleyicilerin kullanımı endüstrinin diğer alanlarında da hızlı bir şekilde yayılmaya başladı. Mikroişlemcilere girişle birlikte, programlanabilir kontrolödiciler üretime ilişkin çok miktardaki dataları toplayıp hesaplayarak kontrol işlemini gerçekleştirecek duruma geldiler. Daha sonra bir programlanabilir

denetleyici kullanan kiři için sofistike metotlar geliřtirildi. Byylece sistem ve kullanıcısı srekli bir etkileřim iinde bulunabilmekteydi. Byylece, eřitli cihazların bir ekran aracılıęı ile, alıřlagelmiř rle sembolleri kullanarak programlanabilmesi mmkn oldu.

1.2 Tarihe

1960'lı yıllarda kontrol iřlemiyle sorunlarla karřılařılmaya kadar elektromekanik letler dzenli olarak kullanılmıyordu. oęunlukla rleler olarak bilinen bu aletlerin binlercesi gerek seri retiminde ve gerekse yalnız makineler de kullanılıyordu. Bu rlelerin oęu, tařıma endstrisinde, otomotiv endstrisinde kullanılmıyordu. Yine bunların binlercesi panellere ve kontrol kabinlerine monte edilmiřlerdi ki; kontrol zmlerini retebilmek iin yzlerce metre uzunluęundaki kablolarla birbirlerine baęlı durumdaydılar. Ancak rle sayısının artması durumunda, szgelimi 300-500 veya daha fazla rlenin olması durumunda gvenilirlik ve sreklilik aısından rleleri panellere monte etmek byk zorluklara neden olmaktaydı. Bir bařka sorun da bu iřin maliyetiydi. Tek bir rle kendi bařına olduka ucuz olmasna karřın bunların monte edilmesi maliyeti bir hayli ykselebilirirdi. Baęlantı kabloları ve iřilikle birlikte her bir rlenin fiyatı 30 ve 40 dolar arasmda deęiřebilmekteydi. stelik bu yetmiyormuř gibi retim prosesinin deęiřtirilmesine ihtiya duyulması durumunda, kontrol panellerinde modifikasyonlar yapmak gerekiyordu ki; bu iři rlelerin bulunduęu bir panel zerinde yapmak (kabloların yeniden baęlanması) olduka byk bir efor gerektirirdi. Bunun dıřında bu deęiřmeler bazen yetersiz kalırdı. Bunun iřięi altında bir kontrol panelinin yeni prosese uygun yeni bir kontrol paneliyle deęiřtirilmesi iřlemi sık sık grlen bir durumdu. Buna nceden kestirilemeyen durumları ve deęiřimin maliyetini de ekleyebiliriz.

İřte, tm bunlar 1960'lı yılların sonunda ilk programlanabilir lojik denetleyicilerinin temellerini atmıřtır. Bu ilk PLC sistemi, Amerika'daki byk otomotiv reticilerinin ihtiyalarına zel cevaplar verecek Őekilde geliřtirildi ve ancak bunların sayesinde karřık kontrol panelleri ortadan kaldırılabildi, eleman deęiřtirmeye ve yeniden baęlantıya ihtiya duymadan deęiřiklikler lojik olarak kolayca yapılılabildi ve bir problem ortaya ıktıęında bunun teřhisi ve giderilmesi olduka kolaylařtı.

Gemiř 15 yıl ierisinde retilen PLC'lerin retim iřlerini ve geliřimlerini incelemek olduka ilgintir. 1980' li yıllarla birlikte PLC' lerde birtakım yeni teknolojik geliřmeler gzlenmiřtir. Bunlar arasmda PLC' lerin daha hızlı bir tarama yapılabilmesini, ok az sayıda rle kullanan sistemler iin dřk fiyatlı PLC' lerin piyasaya srlmesini (bu durum basit kontrol sistemlerinde de PLC kullanımına sebep olmuřtur), PLC' ler iin mikroiřlemci kullanan

akıllı giriş/çıkış birimlerinin kullanılmasını sayabiliriz. Bunun dışında yazılım konusunda da PLC' lerin programlanması için yüksek seviyeli dillerin kullanılmasını sayabiliriz.

1.3 PLC' lerin Üstünlükleri

Tüm bu yazdıklarımıza göre PLC' lerin avantajlarını şu şekilde sıralayabiliriz :

1) Geçmiş yıllarda her bir makine veya kontrol edilecek sistem kendine ait bir kontrolediciye ihtiyaç gösterirken, günümüzde PLC' ler sayesinde bu durum ortadan kalkmış ve sadece bir tek PLC ile birden fazla sistem kontrol edilebilecek duruma gelmiştir.

2) Röleli karmaşık kontrol panellerinde kontrol edilen sisteme ait program değişikliği yeniden bağlantıya ihtiyaç gösterdiği için büyük zaman ve emek gerektirirken, PLC' li bir sistemde yazılımın değiştirilmesi bu iş için yeterlidir.

3) Teknolojik gelişmelerle birlikte PLC' ler daha da fonksiyonel hale gelerek ucuzlamaktadırlar. Günümüzde birkaç yüz dolara içerisinde yüzlerce röle, zamanlayıcı ve sayıcı bulunan bir PLC satın almak mümkündür.

4) Bir kontrol sistemi için yazılmış bir program laboratuvar ortamında gerçek ortamdaymış gibi test edilebilirken, klasik röleli kontrol panellerinin en iyi test ortamı yine kullanılacakları yer olan gerçek ortamlardır.

5) PLC' ler röleli kontrol panellerine nazaran daha çok çalışabilir durumda kalabilirler. Herhangi bir bozulma anında PLC' deki teşhis programları tamir için ortalama zamanı oldukça düşürür.

6) PLC' ler çok az yer kaplar ve daha az enerji tüketirler.

7) PLC' ler bu avantajlarının yanı sıra az sayıda denetim işleminin yapıldığı durumlarda başlangıçta daha fazla yatırım gerektirirler. Ancak klâsik röleli kontrol panellerine göre üst düzeyde bir kontrol işlemini gerçekleştirmeleri açısından oldukça avantajlıdır.

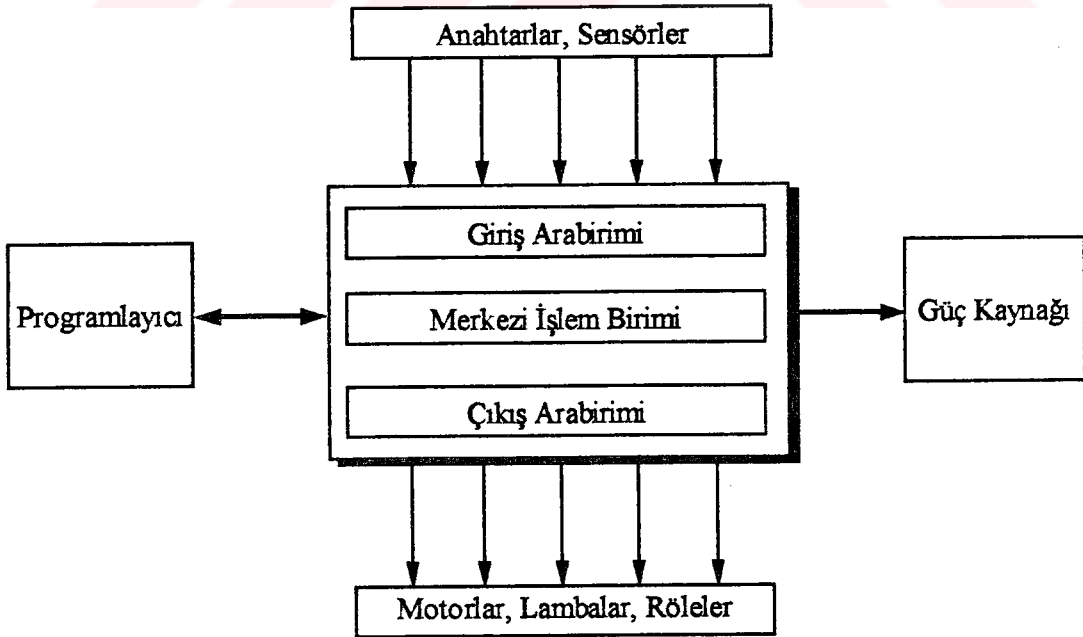
II. PROGRAMLANABİLİR LOJİK KONTROLEDİCİLERİN YAPISI

PLC' ler en genel anlamda belli sayıda giriş ve çıkışlara sahip olan ve girişlerin çeşitli kombinasyonel durumlarına göre, istenilen çıkışları almamızı sağlayan birer mikrobilgisayar şeklinde düşünülebilirler. Veya başka bir tanıma göre bir PLC, makine ve süreçleri denetlemek için sıralama, mantık, zamanlama, sayma ve aritmetik gibi bazı işlemleri gerçekleştirmemizi sağlayan ve bunun için kullanılacak olan komutların depolandığı programlanabilir bir belleği olan sayısal bir elektronik araçtır.

PLC' lerin hemen hemen tümü aşağıda sayılmış olan beş ana kısımdan meydana gelmişlerdir, bunlar :

- Merkezi İşlem Birimi (CPU : Central Processing Unit)
- Giriş Arabirimi
- Çıkış Arabirimi
- Güç Kaynağı
- Programlayıcı

Bu beş ana parçanın yer aldığı tipik bir PLC sisteminin blok diyagramı Şekil 2.1' de gösterilmiştir.



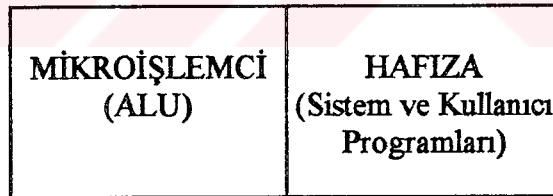
Şekil 2.1 Tipik Bir PLC Sistemi

Yukarıda Tipik bir yapısı gösterilen PLC' lerin bazılarında giriş ve çıkış arabirimlerinin yanı sıra iletişim için de arabirimler bulunmaktadır. Ayrıca yazılan programı yedeklemek ve başka bir PLC' ye aktarmak için EEPROM modülü, enerji kesilmeleri durumunda PLC' yi besleyebilecek yedek güç kaynakları, giriş-çıkış sayılarını arttırmak amacıyla genişleme yuvaları gibi elemanlarda PLC' lerin temel yapılarına dahil edilebilirler. Buradaki PLC' nin yapılarından bazıları donanım birimleri olmakla birlikte, bazıları ise o PLC' ye ait olan yazılım veya programlarının işlevsel özelliklerini yansıtır.

Her PLC' nin üretici firması tarafından yazılmış bulunan bir işletim sistemi programı bulunmaktadır. ROM' da saklı bulunan bu işletim sistemi PLC' lerin üretim aşamasında sisteme yüklenir. PLC' lerin çoğunun işletim sistemi birbirlerine oldukça benzemektedirler. PLC'nin çalışma özelliklerini belirleyen bu işletim sistemi, kullanıcı programlarını yürütür, sistemin hatalı çalıştığı durumları belirleyerek gerekli hata mesajlarını üretir, PLC' lerin çevre birimleriyle haberleşmelerini düzenler.

2.1 Merkezi İşlem Birimi (CPU)

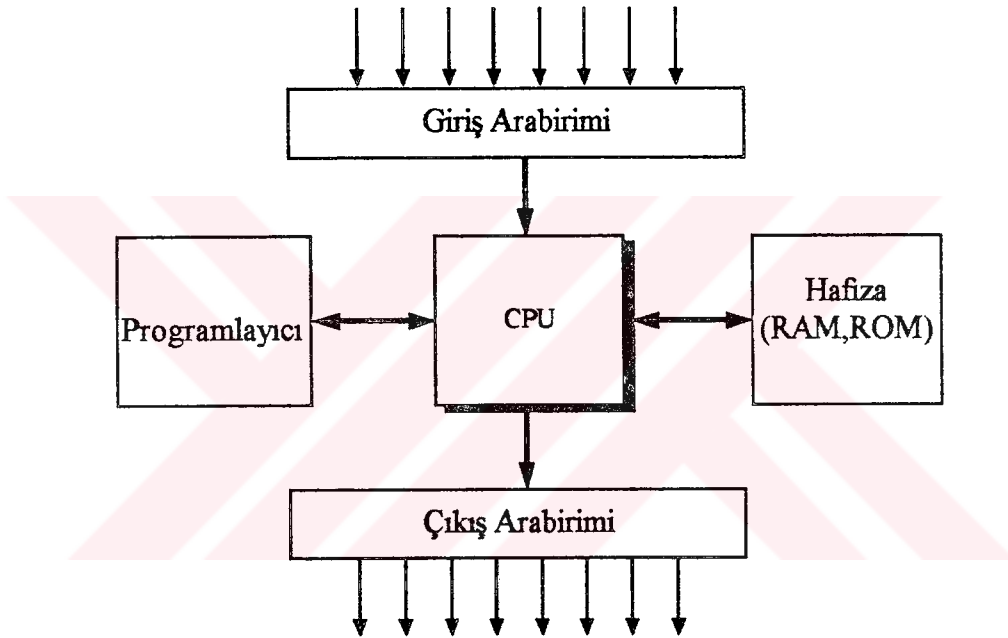
Sistemin Beyni durumundaki Merkezi İşlem Birimi en genel anlamda PLC' nin çalışmasını kontrol eder. İki ayrı kısımdan meydana gelir, bunlar: Mikroişlemci (veya mikrokontroller) ve hafıza bölgesidir (Şekil 2.2).



Şekil 2.2 Merkezi İşlem Birimi

CPU, programın çalışmasının her anında giriş ve çıkış birimlerini kontrol ederek bütün mantık ve hesaplama işlemlerini yapar. Bunun yanı sıra hafıza, giriş-çıkış birimleriyle haberleşmeyi sağlar. CPU' daki mikroişlemci entegre bir devre yongası üzerine sıkıştırılan çeşitli devre elemanlarından meydana gelmiştir. İlk defa 1970 yılında geliştirilen mikroişlemciler daha sonraki yıllarda orta boyutlu entegrasyon (MSI) ' un geliştirilmesiyle yonga üzerindeki eleman sayısı ve dolayısıyla işlemcilerin gücü artmıştır. Bunu, her IC üzerinde 5000 ila 10000 transistörün yer aldığı büyük boyutlu entegrasyon ve 1980' lerin başında her yonga üzerinde 600000 elemanın yerleştirildiği çok büyük boyutlu (VLSI) entegrasyon takip

etmiştir. Son yıllardaki gelişmelerle bu sayılar daha da artmıştır. Mikroişlemcilerdeki bu gelişmelere paralel olarak PLC' ler de her geçen gün daha fonksiyonel özellikler kazanmaktadırlar. PLC' lerde mikroişlemciler yerine "mikrokontroller" adı verilen elemanlar da kullanılmaktadır. Bu elemanların mikroişlemcilerden farkı, işlemcinin bellek ve giriş-çıkış birimleriyle aynı yonga üzerinde bulunmasıdır. Bir mikroişlemci, veri saklamak için belli bir büyüklükteki rasgele erişimli bellek (RAM), komutları saklamak için belli büyüklükteki bir programlanabilir salt okunur bellek (PROM), bir güç kaynağı, çeşitli çevre birimleri ile haberleşmeyi gerçekleştirmek için arabirim devresi ile birlikte mikroişlemciye dayalı bir sistemi oluşturabilmektedir (Şekil 2.3)

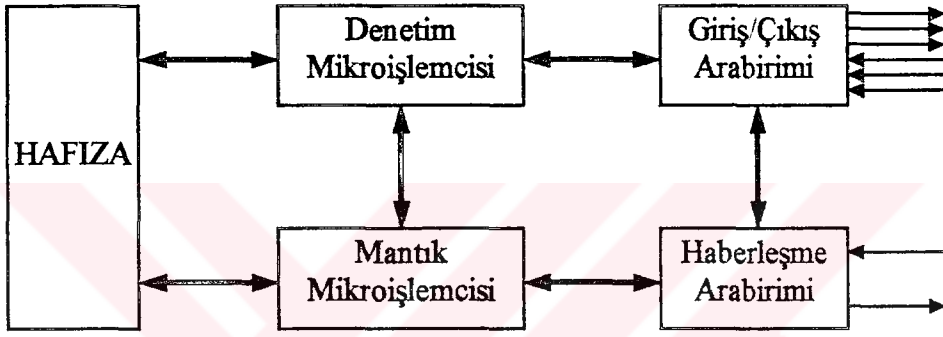


Şekil 2.3 Basit Bir Mikroişlemci Sisteminin Blok Diyagramı

Sistem, mikroişlemci ve program hafızasının ortaklaşa etkileşimi ile çalışır. Devreye güç verildiği andan itibaren işlemci bellekte saklı bulunan ilk komutu okur ve gereken işlemi yapar. Bu işlemi bitirdikten sonra tekrar belleğe geri dönen işlemci, ikinci komutu okur ve ona göre davranır. Bu durum böylece, yapılmak istenen iş tamamlanmaya kadar sürüp gider. İki ayrı PLC üreticisi firma aynı üretmiş oldukları PLC' lerinde aynı işlemciyi kullanabilirler. Ancak her iki firmanın farklı işletim sistemleri tasarlamalarından dolayı üretmiş oldukları PLC' lerde farklı özelliklere ve yeteneklere sahip olacaklardır. İşletim sistemleri daha önce belirtildiği gibi üretim aşamasında PLC' lere yüklenilir ve kullanıcılara açık değildir.

Günümüzde PLC' lerde artık birden fazla mikroişlemci kullanılmaktadır. Bu şekilde sistemin görevi birden fazla mikroişlemci arasında paylaşılmaktadır. Bu çoklu işlemci

yaklaşımında her bir mikroişlemcinin üstlendiği bir görev vardır. Sözgelimi Şekil 2.4'te böyle bir yaklaşıma sahip iki mikroişlemci kullanan bir PLC sisteminin blok diyagramı görülmektedir. Bu sistemde birinin mantık işlemlerini yürüttüğü, diğerinin ise denetim işlemlerini yürüttüğü iki ayrı mikroişlemci vardır. Denetim mikroişlemcisi, iletişim arabirimi ve operatörle haberleşme ve etkileşim gibi karmaşık hesap ve veri ile ilgili işlemlerle ilgilenirken, mantık mikroişlemcisi ise aritmetik ve lojik işlemlerin yanı sıra, zamanlama ve sayma gibi mantık işlemlerini yerine getirmektedir. Programın herhangi bir anında kendi kapasitesinin üzerinde bir görevle karşılaştığı takdirde bu işlemci sorunu denetim mikroişlemcisine devrederek görevine kaldığı yerden devam eder.

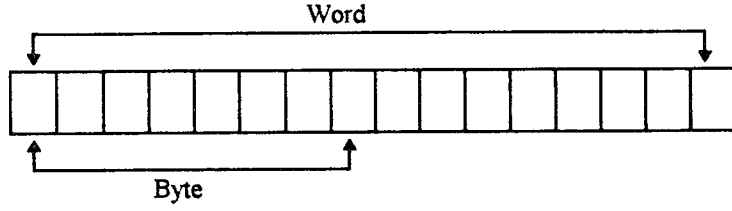


Şekil 2.4 İki Mikroişlemcili PLC'nin Blok Diyagramı

2.2 Hafıza

Sistem programlarını, kullanıcı programlarını ve sistem datalarını saklamak için kullanılan hafızalar iki ana grup altında toplanılabilir. Bunlar : Uçucu ve Kalıcı hafızalardır. İsimlerinden de anlaşılacağı gibi uçucu hafızalardaki bilgiler güç kaynağının kapatılması durumunda, başka bir deyişle beslemenin kesilmesi durumunda kaybolurlar. Bu tür hafızalarda bunu önlemenin yolu belleği bir pil kullanarak beslemektir. Kalıcı hafızalarda ise bilgiler güç kaynağından bağımsızdır ve beslemenin kesilmesi durumunda dahi kaybolmazlar.

PLC hafızası bit, bayt ve word formatları şeklinde ifade edilir. Bit tek bir saklama elemanıdır ve yalnızca "0" veya "1" değerlerinden herhangi birini alabilir. Bir bayt sekiz bitten, bir word ise onaltı bit veya iki bayttan meydana gelir (Şekil 2.5).



Şekil 2.5 *Byte ve Word*

Orta ve büyük ölçekli PLC' lerde hafıza kapasiteleri bir büyüklükten başka bir büyüklüğe değiştirilebilir. Ancak küçük ölçekli PLC' lerde hafızanın büyüklüğü Kbyte olarak ifade edilir. Buradaki Kbyte 1024 Byte' m karşılığıdır ve K harfi ile ifade edilir. Sözelimi 2K' lık bir bellek 2048 byte, 1024 word veya 16384 bitlik bir bilgi depolayabilir.

Çeşitli bilgi türleri özelliklerine göre değişik bellek çeşitlerine ihtiyaç gösterirler. PLC' lerde çeşitli bellek türlerine ihtiyaç gösteren dört ayrı çeşit bilgi yer almaktadır.

Uygulama Bilgisi (Application Memory) Lojik hafıza olarak da adlandırılır. Günlük veya başka bir deyişle halihazırda çalışmakta bulunan kontrol mantığının saklı olduğu programdır. PLC bu tür bilgiyi kullanarak gerekli kontrol işlemlerini gerçekleştirir. Sistem kullanıcısı tarafından oluşturulur. Herhangi bir uygulama için gerçekleştirilen bir program bu tür bilgi için örnek olarak verilebilir.

Veri Tablosu Bilgisi (Data Table Memory) değişken bilgi olarak da adlandırılabilir. Bir uygulamanın kullanmış olduğu herhangi bir matematiksel hesaplamalar sırasında zamanlayıcı ve sayıcının değerlerini içerir. Bunun dışında sistem giriş ve çıkış bilgileri (on-off durumu) de bu tür bilginin kapsadığı alanlar arasına girmektedir.

Çalıştırılan (Executive) Bilgiler sistemin genel amaçlı veya başka bir deyişle denetim tabanını oluşturan bilgilerin bulunduğu yerdir. Her tarama sırasında çevre ile iletişimin sağlandığı bilgiler, yazılan kontrol programının yorumlandığı ve çalıştırıldığı (işletim sistemi bilgileri) bilgiler ve genel amaçlı sistemin kontrol altında tutulduğu bilgiler bu tür arasına girmektedir.

Yazboz Alanı (Scratch Pad) isminden de anlaşılacağı gibi bir geçici bilgi depolama alanıdır. Bazı hesaplamaların ara basamaklarında birtakım ara bilgiler bu tür alanlar arasında sayılabilirler. Bazı sistemlerde bu tür bilgiler PLC' nin bazı istatistiksel bilgilerini de içerirler, sözelimi hafızanın büyüklüğü ve ne kadarının kullanıldığı, flag sayısı gibi.

İşte bu tür bilgilerin ihtiyaç göstermiş oldukları bellek çeşitleri de farklılık göstermektedir. Şimdi bunları sırasıyla inceleyelim:

2.2.1 Hafıza Çeşitleri

Yalnızca Okunabilen Bellek (Read Only Memory: ROM). Mikroişlemcili tabanlı sistemlerde ilk kullanılan ve uçucu olmayan bellek çeşitleri arasındadır. ROM bellek isminden de anlaşılacağı gibi sadece okunabilen bir bellek türüdür. PLC üreticileri ROM bellekleri çalıştırılabilir programları saklamak amacıyla kullanırlar. Bunun sebebi bu tür programların daha sonraki aşamalarda bir düzeltmeye ihtiyaç göstermemeleridir. Görüldüğü gibi ROM' lar işletim sistemleri gibi sonradan herhangi bir değişim yapılmayacak yazılımlar için oldukça ideal bellek çeşitleri arasındadırlar. Bipolar veya MOS teknolojisi kullanılarak üretilen bu belleklere bilgiler üretim aşamasında yakılarak saklanır. Dolayısıyla elektriksel gürültülerden ve besleme geriliminin kaybolmasından etkilenmezler.

Rasgele Erişimli Bellek (Random Access Memory: RAM). Uçucu bellek türleri arasında yer alan RAM belleklerin ROM belleklere göre avantajları okunabilmelerinin yanı sıra üzerlerine bilgi de yazılabilmeleridir. Bu sebepten dolayı bu tür bellekler yaz/oku bellek olarak da adlandırılırlar. Güç kaynağının kesilmesiyle birlikte RAM belleklerdeki bilgiler de kaybolur. Günümüzde RAM' lar CMOS teknolojisiyle üretilmiş olup düşük miktarda güç harcarlar. PLC kullanan sistemlerin çoğu bir prosesi yürütmeye yönelik uygulama programlarını RAM belleklerde saklamaktadırlar. Hızlı olmaları, programların kolaylıkla yaratılabilmelerine, değiştirilebilmelerine ve veri girişine olanak sağlamaları dolayısıyla diğer bellek türlerine göre avantajlıdırlar.

Programlanabilir Yalnız Okunabilen Bellek (Programmable ROM: PROM). ROM bellek türünde olup kullanıcı tarafından bazı özel aletlerle programlanabilme özelliğine sahiptir. Sadece bir defa programlanabilme özelliğine sahip olan bu bellekler, PLC' lerde bazen ve o da genellikle RAM için sürekli depolama kaynağı olarak kullanılırlar.

Silinebilir Programlanabilir Yalnızca Okunabilen Bellek (Erasable PROM: EPROM). Özel bir PROM türüdür. Bu bellek türündeki bilgiler istenildiği zaman kullanıcı tarafından silinip yeniden programlanabilir. Bu silme işinin gerçekleştirilmesi entegre devre üzerinde açılmış bulunan bir pencereye belli bir süreyle uygulanan morötesi ışık sayesinde olur. Yeniden programlama için ise "Eprom Programlayıcı" adlı özel bir programlama cihazı kullanılır. EPROM kalıcı olması istenen uygulama programlarında, çevrim sırasında herhangi bir bilginin yazılmasını gerektirmeyecek durumlarda kullanılabilir.

Elektriksel Olarak İçeriği Değiştirilebilen Bellek (Erasable Alterable ROM: EAROM). ROM bellek türünde olan bu tür bellekler PROM' lara benzer. Ancak bu belleklerin içerikleri morötesi ışık yerine entegre devreye belirli değerlerde bir gerilim uygulanarak değiştirilir veya

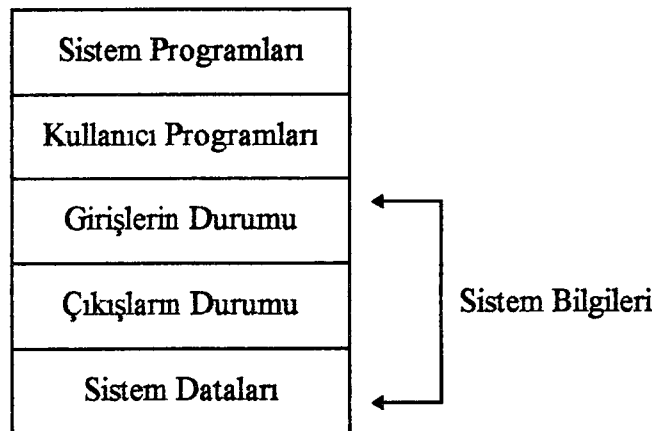
silinir. RAM tipi belleklerde kalıcı bilgi depolama olanağı sağlayan bir destek elemanı olarak yararlanılabilir. Uygulama programlarının saklanması ise pek kullanılmaz.

Elektriksel Olarak Silinebilen PROM (Electrically Erasable PROM: EEPROM).

Birçok PLC ile birlikte sağlanan programlama cihazları yardımıyla tekrar tekrar programlanabilme özelliğine sahiptirler. Bu yüzden yeni PLC' lerde uygulama programlarını saklandığı yer olarak kullanılmaktadırlar. Bunun dışında yaz/boz alanı, data tablosu hafızası olarak da kullanılırlar. Tekrar programlanmasının sınırlı bir sayıya sahip olması (bir baytın silme/yazma operasyonu yaklaşık olarak 10000 düzeyinde sınırlıdır) ve silme/yazma işlemlerinin özellikle çevrim gerektiren programlarda kendini belli edecek kadar uzun süre alması bu bellek türünün dezavantajları arasında yer almaktadır. Ancak buna rağmen birçok PLC üreticisinin uygulama programlarını saklamak amacıyla kullandığı bellek türü arasındadır.

2.2.2 Hafıza Haritaları

PLC' lerdeki hafızalar hafıza haritaları adı verilen bir şekilde organize edilmişlerdir. Örneğin belli bir hafıza büyüklüğüne sahip bir PLC' de hafızanın bir kısmı işletim sistemine ayrılmıştır ve kullanıma kapalıdır. İşte bir hafızanın hangi kısmının, hangi tür fonksiyonel işlemler için ayrıldığını gösteren ve ne kadarının programcının kullanımına açık olduğunu gösteren haritalara hafıza haritaları adı verilir. PLC üreticileri hafıza haritalarını oluştururken birbirlerinden pek farklı olmayan teknikler kullanırlar. Bir üreticide sabit olan hafıza bölgesi başka bir üreticide değişken olabilmektedir. PLC' lerde hafıza Şekil 2.6' da görülebileceği gibi genel olarak üç ana gruba bölünür :



Şekil 2.6 Hafıza Haritası

Sistem veya işletim sistemi programları CPU' nun çalışmasını kontrol eden ve hafızada kalıcı olarak saklı bulunan programlardır. İşletim sistemi donanımı kontrol ettiği gibi kullanıcı programlarının çalıştırılmasını da sağlar. Hafızanın bu bölgesi kullanıcıya kapalıdır.

Kullanıcı programları bir kullanıcı tarafından yazılan ve çalıştırılmak üzere CPU' nun hafızasına aktarılan programları oluştururlar. Hafızanın en büyük kısmı kullanıcı programlarının saklanması için ayrılmış durumdadır.

Sistem bilgileri kullanıcı tarafından kullanılan ve atanan bazı değerleri içerdiği gibi, donanım bilgilerini de içermektedir. Burası hafızada üç ana bölüme ayrılır.

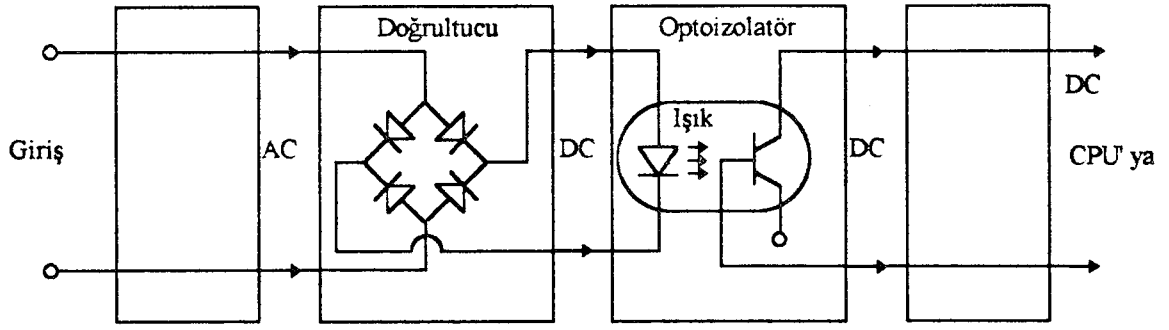
- **Saklama alanı** : Sistem veya kullanıcı tarafından tanımlanan ve kullanılan zamanlayıcı, sayıcı, flag ve benzeri bilgilerin bulunduğu alandır.
- **Girişlerin durumu** : PLC girişlerinin durumunun saklandığı alandır. PLC' nin her taramasından sonra girişlerin durumu bu hafıza bölgesine (1 ve 0 şeklinde) yazılır. Hafızada bu kısım için ayrılan yer PLC' lerdeki giriş sayısına göre belirlenir.
- **Çıkışların durumu** : Girişlerin ve uygulama programının taramasından sonra elde edilen lojik çıkış, hafızanın bu bölgesine yerleştirilerek çıkış modüllerine aktarılır.

2.3 Giriş / Çıkış Birimleri

Giriş ve Çıkış Birimlerinin fonksiyonu, PLC' lerin kontrol edilecek dış aletlerle ve işlemlerle bağlantılarını sağlamaktır. İlk başlarda sadece ayrık giriş/çıkış birimleri varken, günümüzde ise hemen hemen her türlü giriş/çıkış elemanının CPU ile bağlantısını sağlayabilecek arabirimler vardır. Hatta kimi giriş/çıkış arabirimlerine CPU' nun görevini hafifletmesi ve yapılan işlemi hızlandırması açısından zekada eklenmiştir.

2.3.1 Giriş Birimleri

Giriş modülü elektronik olarak dört temel işlevi yerine getirir. Birincisi girişte işaret bulunup bulunmadığını algılar. Giriş işaretleri kontrol edilen işlem içinde hangi anahtar, sensör ya da başka bir işaretin kapalı ya da açık olduğunu belirtir. İkinci kısımda, giriş tarafından fark edilen bu işaret bir çeviriciden geçirilir. Bu çevirici bir AC-DC çevirici olabileceği gibi bir DC-DC çevirici de olabilir. İşaret bu şekilde bir sonraki elektronik modül tarafından kullanılabilir bir duruma getirilir. Çeviriciden gelen işaret bir izolatör yardımıyla izole edildikten sonra CPU' ya gönderilir. Son kısımda ise elektronik devre PLC' nin CPU' su tarafından algılanabilen bir işaret üretir ve CPU' ya gönderir. Bu anlatılan işlemler Şekil 2.7' de görülebilir.



Şekil 2.7 Giriş Arabirimi

Şekil 2.7' de görülen yalnız bir giriş durumu içindir. Diğer girişler, yine bu girişlere benzer devre kullanılır. İlk bloğu meydana getiren sensör doğrudan çeviriciye bağlanmıştır. İkinci blok sensörden gelen işareti alır. Bu blok AC giriş gerilimleri için bir çeviricidir ve içerdiği bir doğrultucu yardımıyla girişindeki işareti çıkışındaki devre tarafından kullanılacak bir seviyeye getirilmesini sağlar. DC giriş işaretleri için ise DC-DC çeviricilere ihtiyaç vardır.

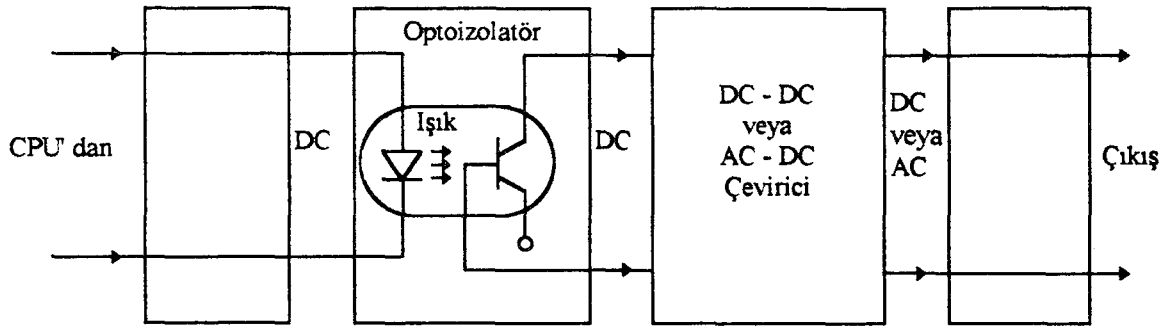
Çeviricinin çıkışı doğrudan PLC' nin CPU' suna verilmez. Aksi takdirde giriş işaretindeki büyük dalgalanmalar ve bozulmalar CPU' ya ulaşabilirdi. Sözgelimi doğrultucu devrenin bir bozulma sonucu devre dışı kaldığını farz edersek, PLC' nin CPU' suna giriş gerilimi seviyesinde (24V dc veya 220V ac...) bir işaret meydana gelecekti. Bu da 5V ile çalışması gereken CPU' ya zarar verecektir. İşte izolasyon CPU' yu bu tür bozulmalara karşı korumaktadır.

İzolasyon Şekil 2.7' de gösterildiği gibi genellikle bir optoizolatör kullanılarak gerçekleştirilir. Böylece giriş modülünde bir işaret olup olmaması durumu bir ışık hüzmesi ile CPU' ya iletilir. Bu durumda PLC girişindeki her türlü bozulma veya dalgalanma CPU' dan yalıtılmış olacaktır.

Çıkış bloğu aracılığıyla CPU' ya gönderilen işaret diğer tüm modüllerden gelen işaretlerle birlikte ikilik kodlanmış olarak hafızaya almır gerçekleştirilecek olan kontrol sisteminde gereken şekilde kullanılır.

2.3.2 Çıkış Birimleri

PLC' nin çıkış modülleri, giriş modüllerinin tam tersi bir biçimde çalışırlar. CPU' dan alınan DC işaret, DC veya AC işaret olarak kullanılabilir. Çıkış gerilimleri seviyelerine çevrilir. Bu şekildeki bir çıkış modülünün blok diyagramı Şekil 2.8' de gösterilmiştir.



Şekil 2.8 Çıkış Arabirimi

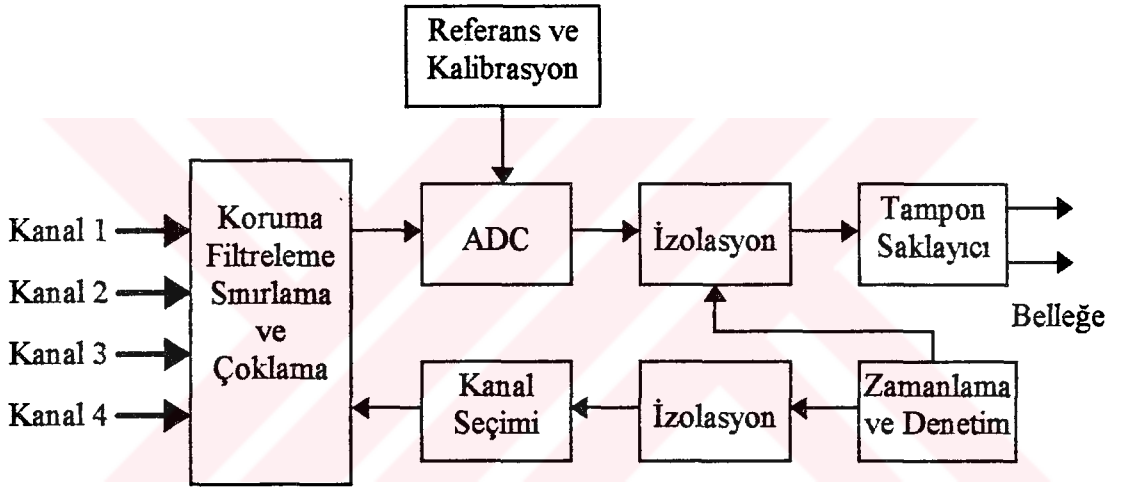
Tarama sonrasında CPU' dan alınan işaret izolasyon devresine uygulanır. Bu şekilde çıkış cihazlarından bir kararsız durumun veya gerilim dalgalanmalarının CPU' ya doğru akması ve dolayısıyla CPU' ya zarar vermesi engellenmiş olur. İzolatör çıkışı daha sonra bir anahtarlama veya röle devresine aktarılır. AC işaret anahtarlama genellikle bir triyak kullanılarak sağlanır. Çıkış modülü PLC' lerin yapısına ve modellerine bağlı olarak AC veya DC işaret olabileceği gibi bir röle çıkışı (s5 90U da olduğu gibi) da olabilir.

2.4 Analog Giriş/Çıkış Birimleri

PLC' lerde ayrı giriş/çıkış birimlerinin yanı sıra analog işaretlerin alınır kullanılmama da izin veren analog giriş/çıkış birimleri de vardır. Analog giriş/çıkış birimleri genellikle hem tek kutuplu (sadece pozitif gerilimleri kabul eden) ve hem de çift kutuplu (pozitif ve negatif gerilimleri kabul eden) bir şekilde imal edilirler. PLC' nin girişlerindeki akım ve gerilimler algılanarak program tarafından kullanılmak üzere A/D çeviricilere uygulanır. Program tarafından, dijitale çevrilmiş olan bu değerler işlenerek, tekrar çeşitli çıkış elemanlarını kontrol etmek veya sürmek amacıyla D/A çeviriciler yardımıyla analog işarete çevrilerek çıkışa verilirler. Giriş ve çıkışlardaki bu dönüştürmeler sırasında önemli olan bu modüllerin çözünürlüğüdür. Sözgelimi 8 bitlik bir çeviriciye sahip giriş/çıkış modülleri 256' da 1' lik bir çözünürlüğe sahiptir. Böyle bir modül 10V' luk bir işareti $10/256 = 0.04V$ ' luk bir hassasiyetle işleyebilir. Bunun yanı sıra 10 bitlik bir çevirici bulunduran bir modül ise $2^{10}=1024$ ' de 1 ve 12 bitlik bir çevirici bulunduran bir modül ise $2^{12}=4096$ ' da 1' lik bir çözünürlüğe sahiptir. Görüldüğü gibi çeviricilerin bit sayısı arttıkça işlenen analog işarettaki hata payı da azalmaktadır.

2.4.1 Analog Giriş

Tipik bir analog giriş birimi Şekil 2.9' da gösterilmiştir. Burada kullanılan bir çoklayıcı ile birden fazla girişin kullanılabilceği görülmektedir. Çoklayıcı bunun yanı sıra doğru olmayan polaritedeki veya büyüklükteki işaretlerin A/D çeviriciye ulaşmasını engeller. CPU ile modül arasındaki izolasyon optoizolatörlerle sağlanır. Birim içerisindeki zamanlama ve denetim kısmı kanal seçimini ve verinin tampon saklayıcıya yazılışını denetler. Böylece saklayıcıya veri yazılması ile CPU' nun buradan veri okumasının aynı anda meydana gelmesi engellenmiş olur. Tampon saklayıcıdaki bilgi CPU' nun tarama hızına bağlı olmayıp A/D çeviricinin çevrim hızına bağlıdır. A/D çeviricinin kalibrasyonu birim içerisinde üretilen bir referans gerilimi ile yapılır. Bu gerilim kalibrasyon işlemi kimi PLC' lerde elle yapılırken, bazılarında ise yazılımla gerçekleştirilebilmektedir.

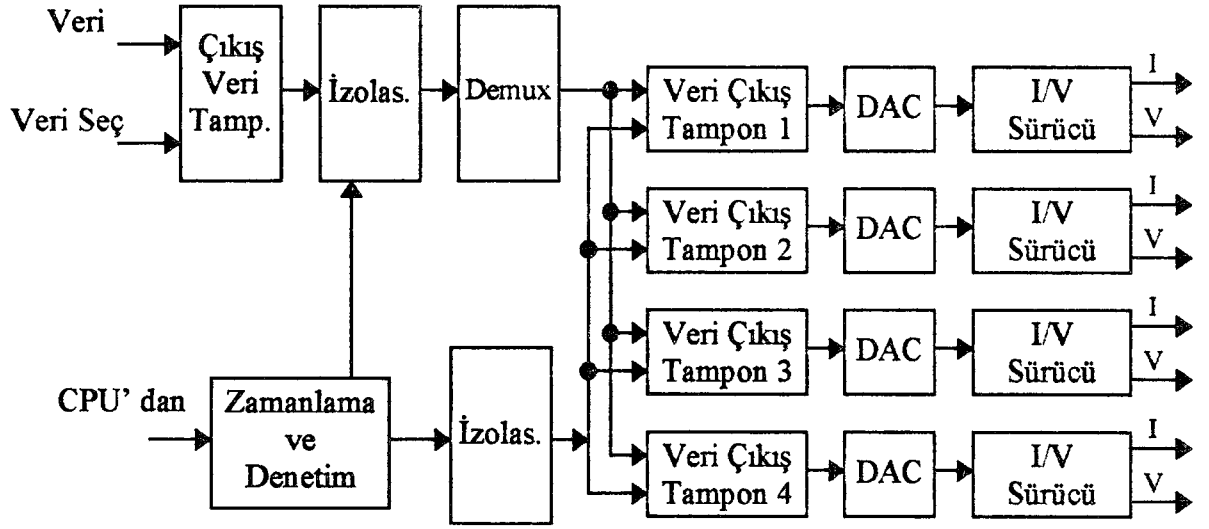


Şekil 2.9 4 Kanallı Bir Analog Giriş Modülünün Yapısı

2.4.2 Analog Çıkış

Şekil 2.10' da yapısı görülen analog çıkış birimi kontrol programından gelen ikili işaretleri analog gerilim veya analog akıma çevirerek çıkış elemanlarının kontrolünü sağlar. Kullanılan veri uzunluğu genellikle 10 bit veya daha azdır. Şekil 2.10' daki birim 4 ayrı datayı işleyebilecek kapasitededir.

Bellekten gelen dijital veri izolasyon işleminden sonra dört ayrı kanala çoğullandırılır. Veri çıkış tamponlarındaki bu işaretler D/A çeviricilerden geçirilerek akım ve gerilim sürücüsü olarak çıkış elemanlarını kontrol etmek üzere çıkışa verilir.



Şekil 2.10 Dört Kanallı Bir Analog Çıkış Birimi

Görüldüğü gibi PLC' lerdeki analog giriş/çıkış birimleri, ADC veya DAC içermelerinin yanı sıra, kanal seçici olarak kullanılan multiplexer kullanmaları bakımından ayrık giriş/çıkış birimlerinden ayrılmaktadırlar.

2.5 Özel Giriş Çıkış Birimleri

PLC' lerde standart giriş/çıkış birimlerinin yanı sıra, kullanıcının ihtiyaç duyduğu özel işlemler için kullanılan giriş/çıkış modülleri de vardır. Aşağıda sayacağımız kimi uygulamalarda PLC ve giriş/çıkış modülleri yetersiz kalabildiğinden özel giriş/çıkış modülleri kullanılır.

- Sadece birkaç milisaniyelik gecikmelerin kabul edilebileceği, çok hızlı değişimlerin yer aldığı uygulamalarda
- Yoğun ASCII veri transferi gerektiren uygulamalarda
- Yoğun matematiksel işlemler gerektiren uygulamalarda
- İleri düzeyde hassas konum denetimi gibi karmaşık süreç denetimi gerektiren uygulamalarda,

özel olarak tasarlanmış giriş/çıkış birimleri kullanılır. Bu tür giriş/çıkış birimleri genellikle akıllı birimler olup PLC CPU' su ile sadece bilgi alışverişini yapmak amacıyla haberleşirler. Zamanlama ve denetim işlerini CPU' nun tarama hızından bağımsız olarak kendi içlerinde gerçekleştirirler.

Günümüzde bu tür birimler sadece orta ve büyük ölçekli PLC' lerde kullanılmakla birlikte, zamanla bu tür birimlerin her PLC modeli için yaygın olarak kullanılması beklenmektedir. Şimdi bu tür özel giriş/çıkış modüllerinden bazılarını kısaca inceleyelim.

Hızlı Giriş Birimi : Bu birim PLC' nin girişindeki çok hızlı (PLC' nin tarama hızından daha küçük) işaret değişimlerini algılamak için kullanılır. PLC' nin girişindeki 50-100ms' lik darbelerin tarama hızı daha yüksek olan PLC tarafından algılanması imkansızdır. İşte bu modül bir darbe genişleticisi gibi çalışarak darbe süresini tarama süresi kadar arttırır veya CPU' nun kabul etmesine bağlı olarak taramayı keserek girişin CPU tarafından algılanmasını sağlar.

Programlanabilen Birimler : Bu tür giriş/çıkış birimleri, PLC tarafından gerçekleştirilmesi mümkün olmayan bazı matematiksel veya mantık işlemlerinin yerine getirilmesini sağlarlar.

PID Denetleyici Birimleri : Ayrık giriş/çıkış modüllerinin açık/kapalı şeklindeki denetimlerinin yetersiz kaldığı ve ileri düzeyde denetim algoritmasının kullanılması gerektiği durumlarda kullanılan bir modüldür.

Bunların yanı sıra termokuplör giriş birimi, ASCII giriş birimi ve hareket denetleyicileri gibi özel giriş/çıkış birimlerini sayabiliriz.

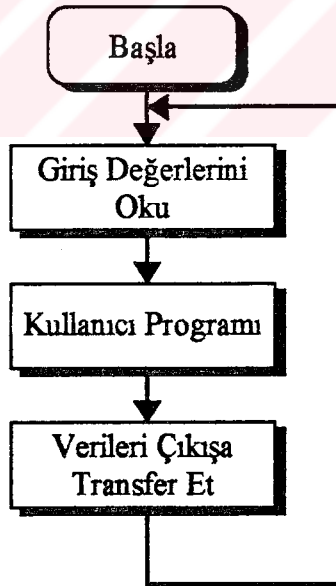


III. PLC İLE PROGRAMLAMA

3.1 Giriş

PLC' ler geliştirildikleri günden itibaren birbirlerinden farklı birçok programlama dili kullanmışlardır. Başlarda, her üretici geliştirdiği PLC modeli için farklı bir programlama dili kullanmaktaydı. Ancak son yıllarda PLC programlama dillerinde IEC 1131-3 adlı bir standart getirildi. Bu standart, daha önce PLC' lerde kullanılan dillerin belli bir uyum içerisinde kullanılmasını amaçlamasının yanı sıra, üreticilerin kullanacakları programlama dilinin yapısını kendi kendilerine belirlemelerine izin vermiyor ve daha önce kullanılan dillerin bu standartlara uyarlanmasına olanak sağlıyor.

PLC programlanmasında program hafızasında bulunan kullanıcı programı , ilk komuttan itibaren başlayarak sırasıyla bütün komutların yürütülmesini sağlar. Programın son kısmı da tarandıktan sonra, program tekrar başa dönerek taramaya devam eder. Şekil 3.1' de genel bir programın yürütülmesine ilişkin akış diyagramı çizilmiştir.



Şekil 3.1 Programın İşlenmesi

Akış diyagramında dikkat edilecek olursa program sürekli bir çevrim içerisindedir. Yani, son komut işlendikten sonra program başa dönerek çalışmaya devam etmektedir. Her

hangi bir kesme anında program kesmenin meydana geldiği durumda, istenilen işi yaptıktan sonra başa dönerek çalışmasına kaldığı yerden devam eder.

PLC' de program çalışmaya başlar başlamaz çizilen akış diyagramından da görülebileceği gibi şu işlemler gerçekleşir.

- Giriş değerleri okunarak hafızaya alınır ve bir sonraki çevrime kadar bu değerler değişmez.
- İstenilen kontrol işleminin gerçekleştirilmesini sağlayacak olan kullanıcı programı adım adım işlenir. Bir önceki adımda hesaplanan ara değerler ve girişte hafızaya alınan değerler kullanılarak çıkış için gerekli hesaplamalar yapılır. Bu arada meydana gelen bir kesme isteği yerine getirildikten sonra program çalışmaya kaldığı yerden devam eder.
- Kullanıcı programının yürütülmesi tamamlandıktan sonra çıkış için oluşturulan bu değerler çıkış birimine transfer edilerek yeni bir çevrim için program başa döner.

3.2 Tarama Zamanı

PLC' lerde bir çevrimin tamamlanması için geçen süre olarak tanımlanır. Bir PLC' nin tarama zamanı PLC' nin giriş/çıkış sayısına, program uzunluk ve karmaşıklığına ve ayrıca merkezi işlem biriminin çalışma frekansına bağlıdır. PLC' lerde genel olarak tarama zamanı birkaç ms' den 300ms' ye mertebelerine kadar çıkabilmektedir. Tarama hızı ise 1024 byte (1Kbyte) lık bir program parçasını taramak için geçen süre olarak ifade edilir.

3.3 Programlama Çeşitleri

PLC' lerde programlama genel olarak iki kısım altında toplanabilir. Bunlardan birincisi kullanım açısından kullanıcıya kolaylıklar sağlayan grafik temelli programlama, ikincisi ise metin tabanlı olan ve kısa ve esnek programlar yapılmasını sağlayan programlama yöntemleridir.

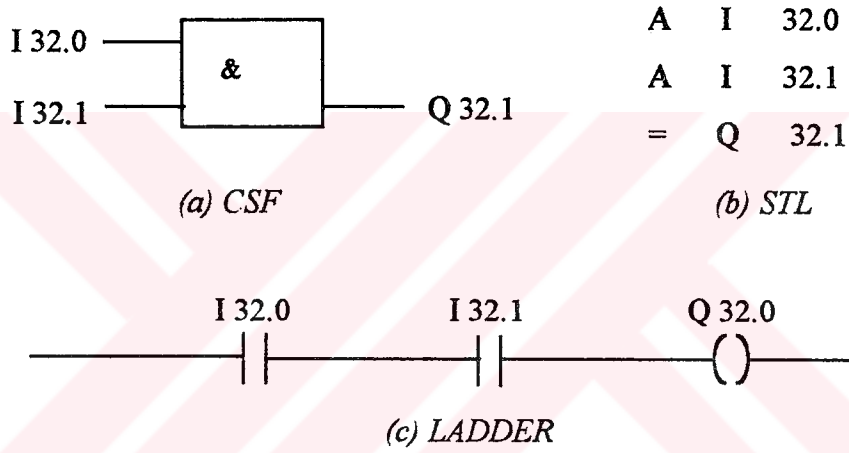
Grafiksel programlama yöntemlerinden olan Ardışık Fonksiyon Tablosu (Step 5' te CSF) şeklindeki programlamada, her basamak, yapılması gereken bir işlemi gösterirken basamaklar arasındaki bağlantı diğer basamağa geçmek için gereken şartı gösterir. Bir diğer grafiksel programlama yöntemlerinden olan Merdiven Diyagramı yönteminde ise düşünülen yapı röleler ile gerçekleştirilecekmiş gibi ele alınır ve yazılımla gerçekleştirilir. Metin tabanlı programlamadan ise, daha çok grafik tabanlı programların desteklenmesinde yararlanır.

Bu tür programlamalara ilişkin örnekler Step 5 programlama dilinde Bir sonraki bölümde ayrıntılı olarak incelenecektir.

IV. STEP 5 PROGRAMLAMA DİLİ

4.1 Giriş

Step 5 programlama dili değişik otomasyon fonksiyonlarını formüle etmek için SIMATIC S5 programlanabilir kontrolcileriyle birlikte kullanılır. Step 5' de bulunan üç çeşit programlama metodu programlamayı kolaylaştırır ve ayrıca programlama dilini öğrenmeye ve kullanmaya oldukça yardım eder. Bu üç çeşit programlama şunlardır : (1) Ladder diyagram (LAD), (2) Control System Flowchart (CSF) ve (3) Statment List (STL). Şimdi basit bir AND işlemiyle her üçüne kısa bir örnek verelim.



Şekil 4.1 Step 5 Programlamasına Basit Bir Örnek

Simatic S5 programlaması programlamayı kolaylaştırması açısından çeşitli bloklara bölünür. Aynı şekilde programlama kendi arasında segmentlere bölünür.

Programlamada kullanılan blokları kısaca şu şekilde gösterebiliriz :

4.2 Programlamada Kullanılan Bloklar

4.2.1 Organizasyon Blokları (OB) :

Genel anlamda kullanıcı tarafından programlanan ve önceden tanımlanmış koşullar altında sistem tarafından çağrılan, programın ana bloğunu oluşturan kısımdır. Tüm diğer bloklar bu blok içinde yazılmış olan programlar tarafından çağrılır ve kullanılır. Programlama yapılırken organizasyon bloğu mutlaka bulunmalıdır.

4.2.2 Program Blokları (PB) :

Program blokları asıl fonksiyonların yazıldığı ve programın en büyük parçasını oluşturan kısımlarıdır.

4.2.3 Fonksiyon Blokları (FB) :

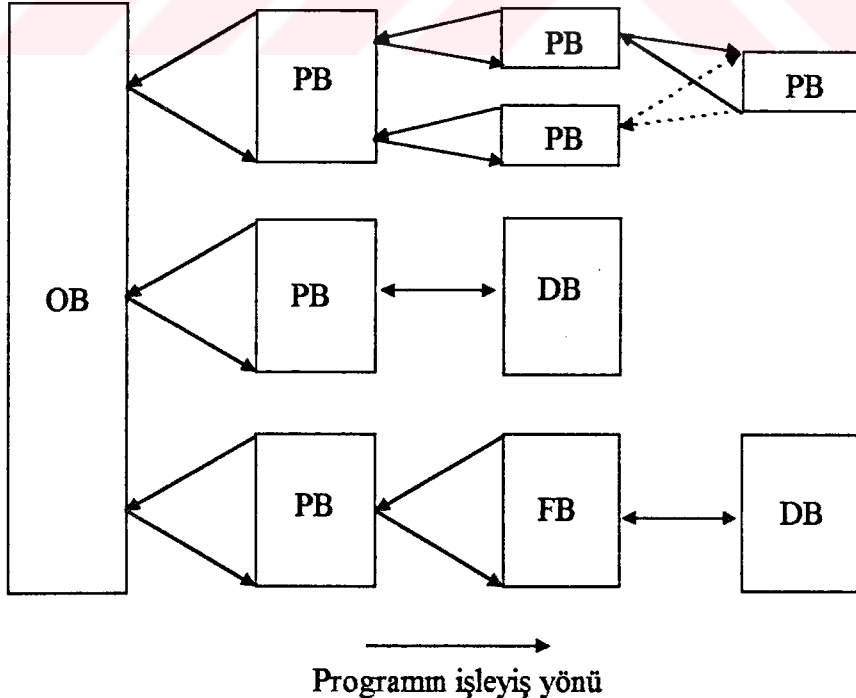
Birtakım özel aritmetik ve karmaşık işlemlerin yapılmasını sağlayan bloklardır. Yani program blokları içinde gerçekleştirilemeyen matematiksel ve lojik işlemlerin yapılması sırasında kullanılırlar. Aynı şekilde programlama blokları gibi komut listesinden oluşurlar ve programlama bloğu içerisinde yazılabilecek her türlü komutu kabul ederler.

4.2.4 Data Blokları (DB) :

Data blokları program tarafından kullanılan bilgileri saklamak ve görüntülemek için kullanılır. Bu bilgiler çeşitli hesaplamaların sonuçları mesaj yazıları, giriş-çıkış durumları, vb... olabilir. Bu bloklar çalıştırılabilen komutlar içermezler (sadece programda kullanılmak üzere birtakım sayılardan oluşmuşlardır).

4.2.5 Sıralı Bloklar (SB) :

Sıralama işlemlerinde kullanılırlar ve programlama bloklarının özel bir durumudurlar.

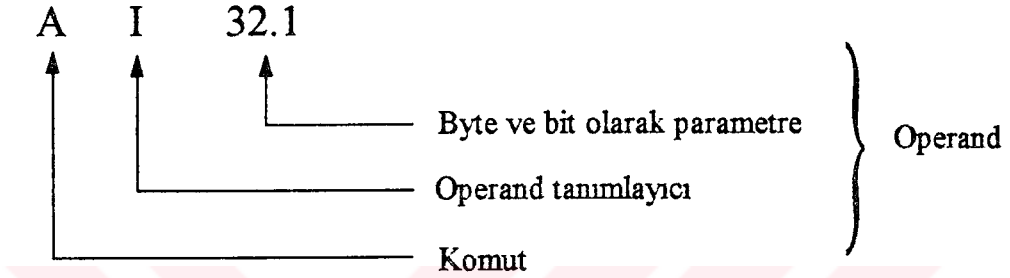


Şekil 4.2 Step 5' te Programlamanın Blok Olarak Gösterimi

Genel olarak bir programın bloklar arasındaki işleyişi grafiksel olarak yukarıdaki şekilde ifade edilebilir (Şekil 4.2).

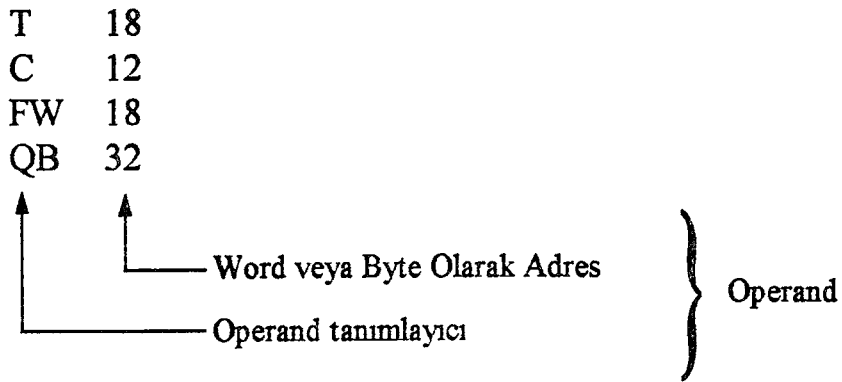
4.3 Step 5 Programlama Dilinde Kullanılan Operandlar

Program yazımı sırasında (LAD, CSF veya STL olabilir) kullanılan komutlar programın yapısına göre değişik operandlar alır ki ; bu operandlar da iki kısımdan meydana gelir : Bunlar (1) operandın tümünü gösteren operand tanımlayıcı ve (2) adresleri içeren parametrelerdir. Bunu kısaca gösterelim (Şekil 4.3).



Şekil 4.3 *Operand*

Programlamada dikkat edilecek olursa parametre kısmı bir nokta ile birbirinden ayrılmış iki parçadan meydana gelmiştir. Burada noktanın solunda bulunan kısım adresi gösterirken, sağında bulunan kısım ise adresin hangi biti olduğunu göstermektedir. Adresler komutun yapısına bağlı olarak bir byte'ı gösterebileceği gibi, bir "word" (iki byte) ü de gösterebilirler. Programlamada adreslerin word olarak kullanılması durumlarında operandın sonuna "W" harfi gelirken, byte olarak kullanılmaları durumlarında ise B harfi gelmektedir. Zamanlayıcı (T) ve sayıcılar (C) ise programlamada daima word adreslerini kullanmaktadırlar. Bu yüzden tek harf ile ifade edilirler(sonlarına "W" harfi almazlar). Bunları kısaca gösterelim :



Şekil 4.4 *Operandın Programlamada Kullanılması*

Tüm girişler (I), çıkışlar (Q) ve bayraklar (F) byte ve word adreslemesine sahip olmakla birlikte bit adresleme işlemlerinde de kullanılabilirler. Datalar (D) ise sadece word ve bit adresleme işlemlerinde kullanılırlar.

Step 5 programlama dilinde aşağıdaki operandlar kullanılır.

Girişler (Inputs) I : Programlanabilir kontrolediciyle, yapılan program arasında bir arabirim oluştururlar. Word olarak IW, byte olarak IB şeklinde ifade edilirler

Çıkışlar (Outputs) Q : Programlanabilir kontrolediciyle yapılan program arasında bir arabirim oluştururlar. Word (ikibyte) olarak QW, byte olarak QB şeklinde gösterilirler.

Bayraklar (Flags) F : İkili lojik işlemlerin ara sonuçlarını saklamak için kullanılırlar. Word ve byte olarak gösterimleri sırasıyla FW ve FY şeklindedir. burada dikkat edilecek olursa Flag'ların byte olarak gösterimi FB şeklinde olmayıp FY şeklinde olmasıdır. Bunun sebebi programlama sırasında Flag'ların FB şeklinde gösterilen Fonksiyon bloklarıyla karışmamasıdır.

Datalar D : Byte ve Word olarak yapılan ara işlemlerin sonuçlarını saklamak için kullanılırlar. Programlamada word olarak DW şeklinde gösterilebilen Data'lar ayrıca Word'ün sağ ve sol byte'mı ifade etmek üzere şu şekilde de kullanılabilirler :

DL : DW' nin sol Byte'ı

DL : DW' nin sağ byte'ı

Zamanlayıcılar (Timers) T : Zamanlama işlemlerini yerine getirirler. Sadece word olarak ifade edilebilirler. ileride ayrıntılı olarak ele alınacaklardır.

Sayıclar (Counters) C : Sayma işlemlerini yerine getirirler. Zamanlayıcılar gibi sadece word olarak değer alabilen sayıcılarda daha sonra ayrıntılı olarak inceleneceklerdir.

Arabirimler (Peripherals) P : Harici birimleri doğrudan referans alarak bu birimlerle olan işlemlerde kullanılırlar.

Sabitler (Constans) K : Programda doğrudan kullanılmak istenen değerler için gereken işlemlerde kullanılırlar. Sabitler programlamada sekiz değişik biçimde ele alınıp programlanabilirler. Şimdi bunları kısaca görelim :

KB 0 ila 255 : Byte olarak programlamada kullanılabilir.

KF -32768 ila 32767 : İşaretili sayı olarak programlamada kullanılabilir.

KY 0 ila 255, 0 ila 255 : İki byte olarak programlamada kullanılabilir.

KH 000 ila FFF : Heksadesimal olarak programlamada kullanılabilir.

KM 0000000000000000 ila 1111111111111111 : İkili olarak gösterilip programlamada kullanılabilir.

KS ASCII : ASCII karakter olarak ifade edilebilir.

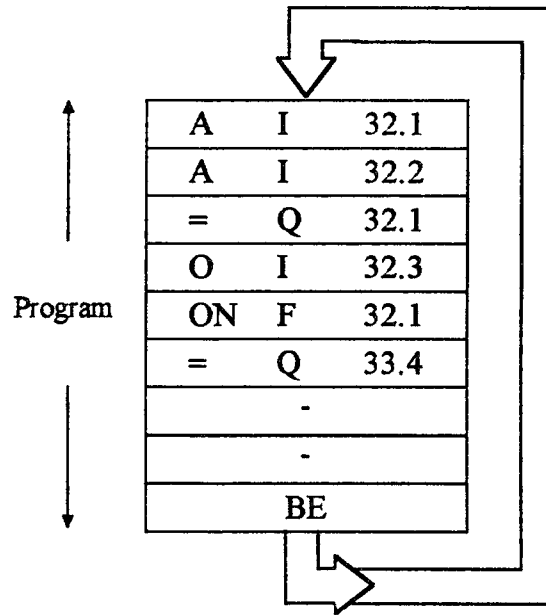
KT 000.0 ila 999.3 : Zaman sabiti olarak programlamada kullanılabilir.

KC 000 ila 999 : Sayıcı (counter) sabiti olarak programlamada kullanılabilir.

4.4 Step 5 Programlama Dilinde Kullanılan Komutlar ve Programlama

Step 5 ile programlamanın üç değişik şekilde yapılabileceğini belirtmiştik. Bunlardan CSF ve LAD programlamaları kişiye grafiksel olarak programlama yapma imkanı sağlarken, STL (liste yöntemiyle yapılan programlama) ise kısa ve esnek programlar yapılmasını sağlamaktadırlar. Step 5 ile yapılan programlama, herhangi bir mikroişlemci veya bilgisayar programından farklı olarak sürekli bir tarama içerisinde (Şekil 4.5).

Organizasyon bloğunun ilk satırından başlayan tarama, programın yapısına göre diğer bloklara dallandıktan sonra tekrar başa dönerek devam eder, bu da, programın sürekli bir çevrim içerisinde olduğunu gösterir.



Şekil 4.5 Step 5 Programının Çalışması

Şimdi burada step 5’de programlamanın nasıl yapıldığına ve ne tür komutların kullanıldığına bakalım.

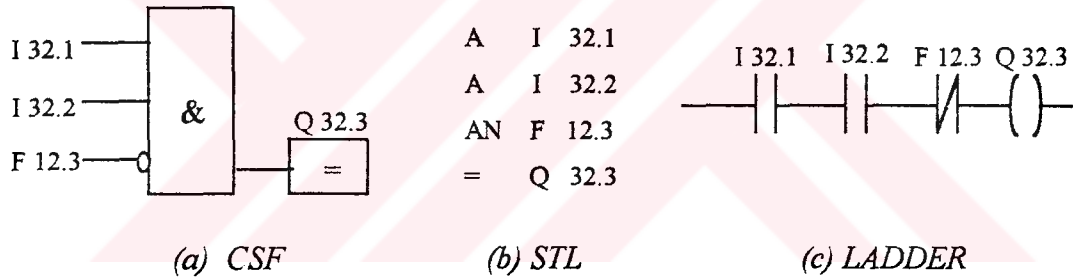
Step 5’ de kullanılan komutları aşağıdaki şekilde gruplayabiliriz.

- Lojik işlemler (AND, OR, EXOR, NAND, NOR ...)
- Set / Reset işlemleri
- Zamanlayıcı İşlemleri
- Sayıcı işlemleri
- Bit test işlemleri
- Word ve Byte İşlemleri

4.4.1 Lojik İşlemler

4.4.1.1 AND İşlemi

Lojik olarak AND işleminin gerçekleştirilmesini sağlar. CSF programlamasında “&” sembolü ile temsil edilir.



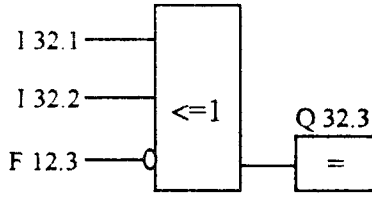
Şekil 4.6 CSF, STL ve LADDER’ da AND İşleminin Gösterimi

Şekil 4.6’ da girişlerden I 32.1 ve I 32.2’ in “1” olması ve F 12.3’ ün “0” olması durumlarında çıkış Q 32.3 “1” olmaktadır.

4.4.1.2 OR İşlemi

OR işleminin gerçekleştirilmesini sağlar. CSF programlamasında <=1 sembolü ile temsil edilir.

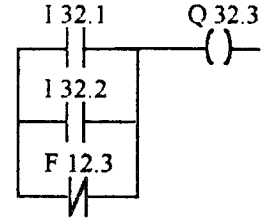
Şekil 4.7’ de I 32.1’in “1” veya I 32.2’ nin “1” veya F 12.3 ‘ün “0” olması durumunda Q32.3 çıkışı “1” olmaktadır.



(a) CSF

O I 32.1
 O I 32.2
 ON F 12.3
 = Q 32.3

(b) STL



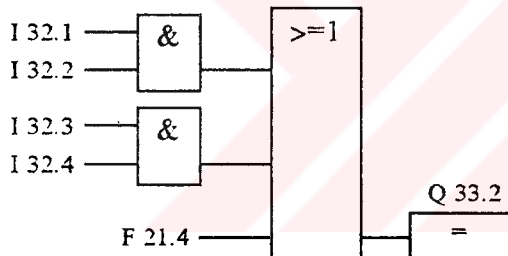
(c) LADDER

Şekil 4.7 Step 5' te OR İşlemi

Bu her iki komutun ayrı ayrı kullanılmalarının dışında AND' den önce OR veya OR' dan önce AND şeklindeki kullanılmalarının nasıl olduğuna bakalım.

4.4.1.3 OR' dan Önce AND İşlemi

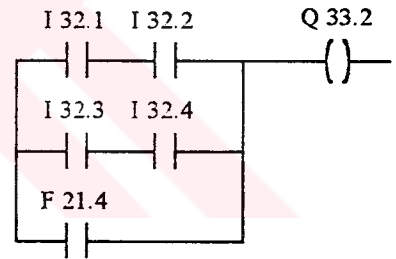
AND ve OR işlemlerinin bu kombinasyonlarında OR işleminden önce AND işleminin nasıl yapıldığı Şekil.4.8' de gösterilmiştir.



(a) CSF

A I 32.1
 A I 32.2
 O
 A I 32.3
 A I 32.4
 O F 21.4
 = Q 33.2

(b) STL



(c) LADDER

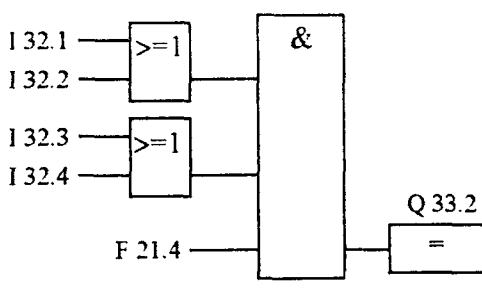
Şekil 4.8 OR' dan Önce AND İşlemi

Şekilde dikkat edilecek olursa yukarıda her üç şekilde de gösterilen programda şu işlem yapılmaktadır : (I32.1 AND I 32.2) OR (I 32.3 AND I 32.4) OR F 21.4 = Q 33.2.

4.4.1.4 AND' den önce OR İşlemi

AND ve OR işlemlerinde AND işleminden önce OR işleminin nasıl yapıldığını gösteren program CSF, STL ve LADDER olarak Şekil 4.9' da verilmiştir.

Şekil 4.9' da step 5 programlama diline göre her üç şekilde de yazılan bu programda şu işlem yapılmaktadır : (I32.1 OR I32.2) AND (I32.3 OR I32.4) OR F21.4 = Q33.2



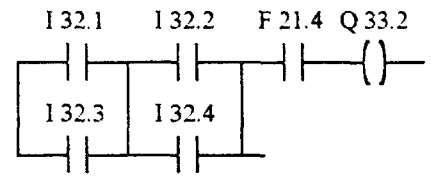
(a) CSF

```

A(
O 1 32.1
O 1 32.2
)
A(
O 1 32.3
O 1 32.4
)
A F 21.4
= Q 33.2

```

(b) STL



(c) LADDER

Şekil 4.9 AND' den Önce OR İşlemi

Step 5 programlama dilinde Set/Reset işlemlerine geçmeden önce değinilmesi gereken iki önemli konu vardır ki; bunlar; Programların çalıştırılması sırasında sık sık kullanılan akümülatör, diğeri de RLO (Result of Logic Operation) dur. Bunları sırasıyla inceleyelim.

4.4.2 RLO ve Akümülatör

Step 5' te yazılan programların işlenmesi sırasında programda herhangi bir anda meydana gelen bir duruma göre programın herhangi bir noktasına dallanılmak istenebilir. Sözelimi program sırasında istenilen bir durumun sağlanıp sağlanmadığı kontrol edilip ona göre bir işlem yapılmak istenildiği takdirde ne yapılacaktır? İşte bu durumda programda işlenen her satırın durumunu bit olarak saklayan RLO kullanılmaktadır. RLO program sırasında bir komutun işlenmesinde "1" veya "0" değeri alabilen bir durum olarak ifade edilebilir. Şimdi buna bazı örnekler verelim :

A I 22.3 : I 22.3 "1" için taranır ve işlemdeki "RLO" ile AND işlemine sokulur.

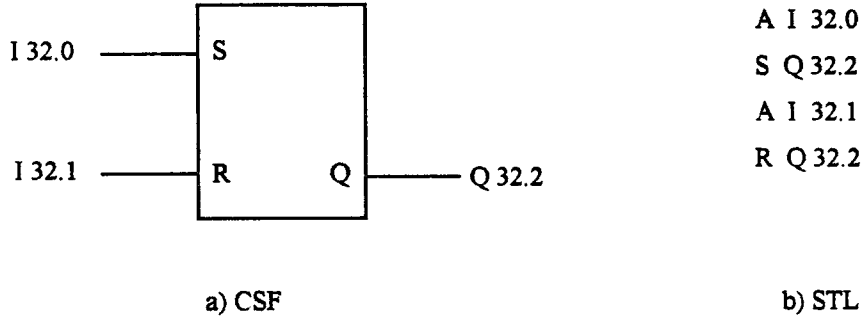
ON F 18.3 : F 18.3 "0" için taranır ve işlemdeki RLO ile OR işlemine sokulur.

Koşullu işlemler RLO işleminin sonucuna göre çalıştırılan işlemler olarak ifade edilebilirler. Burada bilinmesi gereken bu koşullu işlemlerin RLO' ya etki etmeyeceğidir. Yani tarama durumlarında (scan statement) RLO elde edilir ve koşullu durumlarda bu RLO kullanılır.

Akümlatör : RLO gibi çalışan ancak bilgi kapasitesi byte ve word mertebesinde olan aritmetik ve lojik işlemlerde kullanılan bir saklayıcıdır. Daha sonraki bölümlerde örneklerle ayrıntılı olarak incelenecektir.

4.4.3 Set Reset İşlemleri

Set-reset işlemlerinin yapılmasını sağlayan RS FF'ler kullanılır. Burada RLO (Result of Logic Operation)'nun program içindeki herhangi bir andaki değişimi, RS Flip-Flop'lar yardımıyla programın herhangi bir anındaki RLO'dan bağımsız olarak saklanabilir. Buradaki RS FF'ler "set üstünlüklü" ve "reset üstünlüklü" olmak üzere iki kısımda incelenebilirler. Ancak buna değinmeden önce CSF ve STL'de RS FF'lerin nasıl gösterildiğine bir bakalım.



Şekil 4.10 RS FF' in CSF ve STL Olarak Gösterimi

Şekil 4.10' da dikkat edilecek olursa RS FF'in set girişi I 32.0'a, reset girişi I 32.1' e ve Q çıkışı ise Q 32.2'ye bağlı durumdadır. Burada I 32.0 girişinin lojik 0'dan lojik 1'e değişmesi durumunda RS FF'in Q çıkışı (Q 32.2) set edilir ve bu durum program boyunca I 32.1'in "1" olmasına (RS FF resetlenene) kadar devam eder.

Set veya reset üstünlüklü olmak üzere iki kısımda incelenebilen RS FF'ler den set üstünlüklü olanın her iki girişine aynı anda "1" gelmesi durumunda Flip-Flop set edilir. Aynı şekilde her iki girişin aynı anda "0" olması durumunda çıkışta herhangi bir değişim meydana gelmez.

Reset üstünlüklü RS FF'de ise durum bunun tam tersidir. Yani bu tür Flip-Flop'da her iki girişin aynı anda "0" olmasıyla birlikte üstünlük "reset"de olduğu için FF reset edilir. Burada her iki girişin aynı anda "1" olması çıkışlara etki etmez.

4.4.4 Zamanlayıcılar

PLC'lerde bir işlemi belli bir süre kullanarak yapmamızı sağlayan zamanlayıcılar Step 5 programlama dili yardımıyla beş değişik şekilde programlanabilirler. Her zamanlayıcı RLO'nun "0" dan "1" e değişmesiyle birlikte (start girişi buna göre tanımlanmışsa) çalışmaya başlar. Zamanlayıcıların çalışma sürelerini zamanlayıcılarda kullanılan zaman sabitleri belirler. A.B

şeklindeki birbirinden noktayla ayrılan iki sayıdan oluşan zaman sabitinin sağ tarafındaki kısım (B) zaman tabanına karşılık gelirken, sol tarafındaki kısım (A) ise zamanı göstermektedir ki, bu gösterilen zaman, zaman tabanının değişimine göre değişim göstermektedir. Buradaki gerçek zaman A ile B'nin çarpımı şeklinde ifade edilir.

A.B Şeklindeki zaman sabitinde A 0-999 arası bir sayıyı gösterirken, B 0-3 değerlerine karşılık gelmektedir. Zaman tabanının her değerine karşılık gelen zamanlar şu şekildedir:

0 : 10 Milisaniye

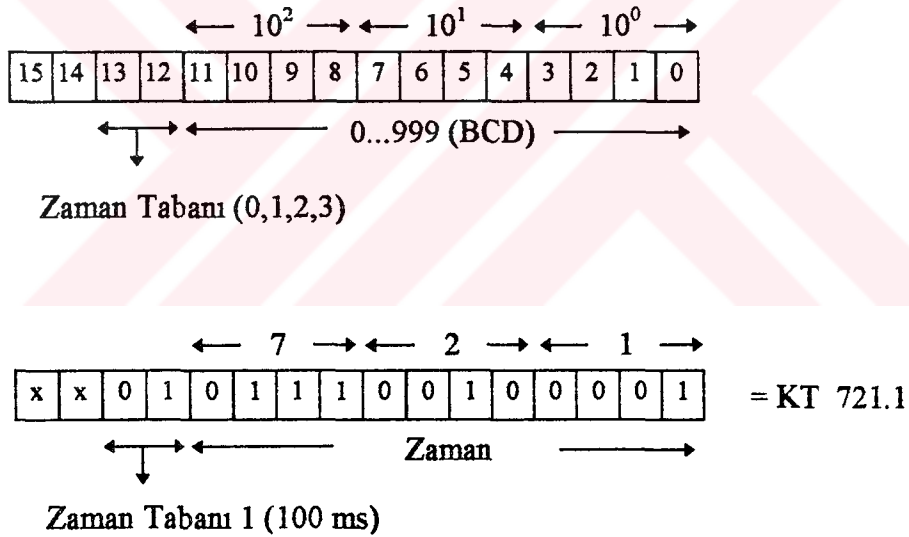
1 : 100 Milisaniye

2 : 1 Saniye

3 : 10 Saniye

Örneğin zaman sabiti $862.0 = 862 \times 10 \text{ ms} = 8620 \text{ ms}'ye$ veya $721.2 = 721 \times 1 \text{ Sn} = 721 \text{ Sn}'ye$ karşılık gelmektedir.

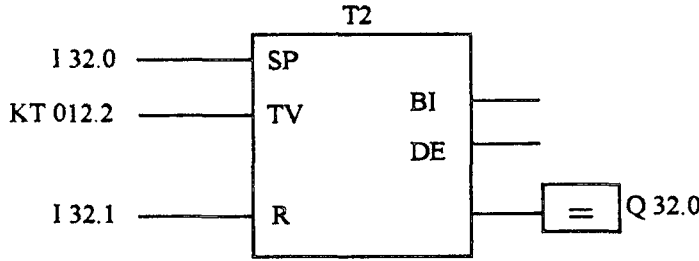
Bir zamanlayıcı çalıştırılmaya başlamadan önce yapılacak işlem için belirlenen zaman sabiti akümülatör 1'e yüklenmelidir. Bu işlem L KT A.B şeklinde gerçekleştirilebilir. (Buradaki KT zamanlayıcı sabitine karşılık gelmektedir). 16 Bitlik (word) bir sayıya karşılık gelen zamanlayıcı sabitinin ikili olarak formatı ise aşağıdaki şekildedir (Şekil 4.11).



Şekil 4.11 Zamanlayıcı Sabitinin Formatı

Step 5'de 5 değişik şekilde zamanlayıcıları programlayabileceğimizi belirtmiştik. Bunlar: (1) Pulse Timer (SP), (2) Extended Pulse Timer (SE), (3) Switch-On Delay Timer (SR), (5) Stored Switch On Delay Timer (SS), (5) Switch Off Delay Timer (SF).

Zamanlayıcıların STL, CSF ve Ladder'da programlanması aşağıdaki şekilde gösterilmiştir (Şekil 4.12).



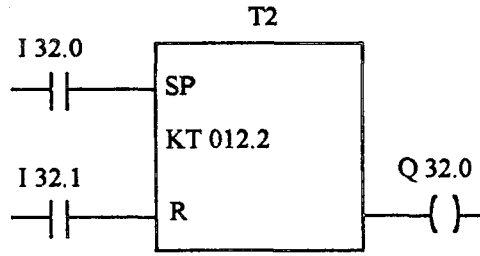
(b) CSF

```

A I 32.0
L KT 012.2
SP T2
A I 32.1
R T2
A T1
= Q 32.0

```

(a) STL



(c) LADDER

Şekil 4.12 Zamanlayıcı Gösterimi

Şekil 4.12' de gösterilen zamanlayıcı örneğinde SP (Pulse Timer) zamanlayıcısı kullanılmıştır. Buradaki örnekte SP'ler diğer zamanlayıcı türlerine karşılık gelecek şekilde değiştirilebilirler.

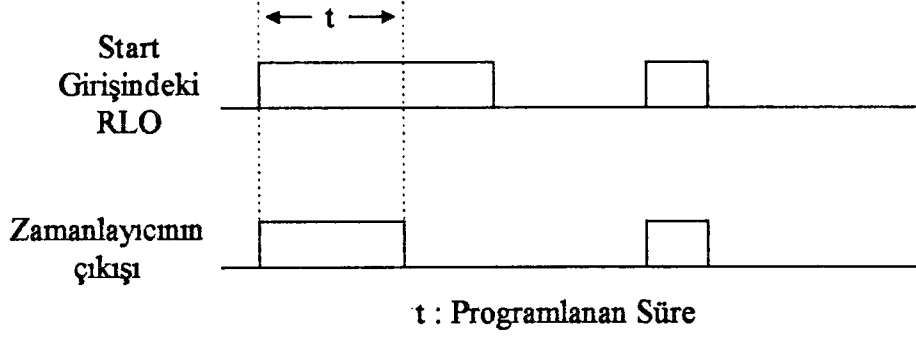
Burada I 32.0'm "1" olmasıyla birlikte zamanlayıcı çalışmaya başlar. Aynı şekilde I 32.1'in "1" olmasıyla birlikte zamanlayıcı resetlenir (ilk durumuna döner). T2 ikinci zamanlayıcı gösterirken (programlamada birbirinden bağımsız birden fazla zamanlayıcı kullanılabilir), KT 12.2'de zamanlayıcının 12 x 1 = 12 Sn çalışacağını göstermektedir.

Yine dikkat edilecek olursa CSF'de reset aynı blok üzerinde gösterilebilirken, ladder'da ayrı bir segment olarak gösterilmesi gerekmektedir. (CSF'deki BI ve DE çıkışları sırasıyla Binary ve Decimal çıkışlara karşılık gelmektedirler).

4.4.4.1 Pulse Timer (SP)

Bu tür zamanlayıcılarda zamanlayıcı girişindeki RLO'nun "0" dan "1" e değişmesiyle birlikte zamanlayıcı çalışmaya başlar ve girişi "1" olur. Girilen zaman sabiti süresi sonunda giriş hala "1" durumundaysa çıkış "0" olur. Ancak belirlenen süre geçmeden zamanlayıcının girişi "0" a düşerse zamanlayıcının çalışması durur ve çıkışta "0" a düşer. Bu durumda

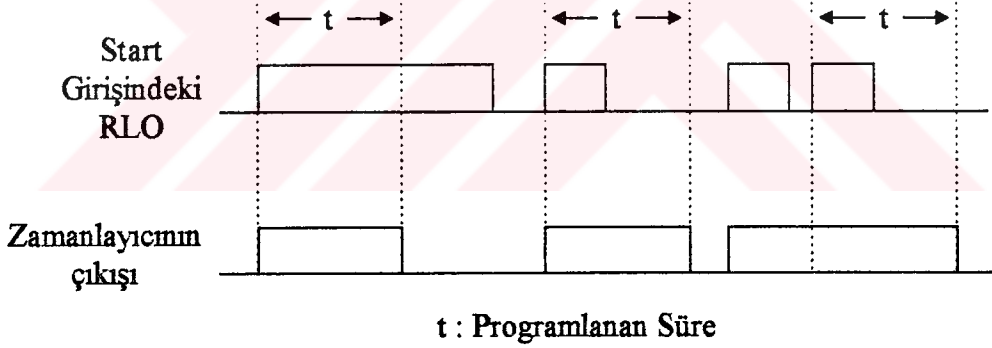
zamanlayıcının tekrar çalışabilmesi için girişin "1" yapılması gerekir. Bu zamanlayıcının karakteristiği Şekil 4.13' de gösterilmiştir.



Şekil 4.13 Pulse Timer

4.4.4.2 Extended Pulse Timer (SE)

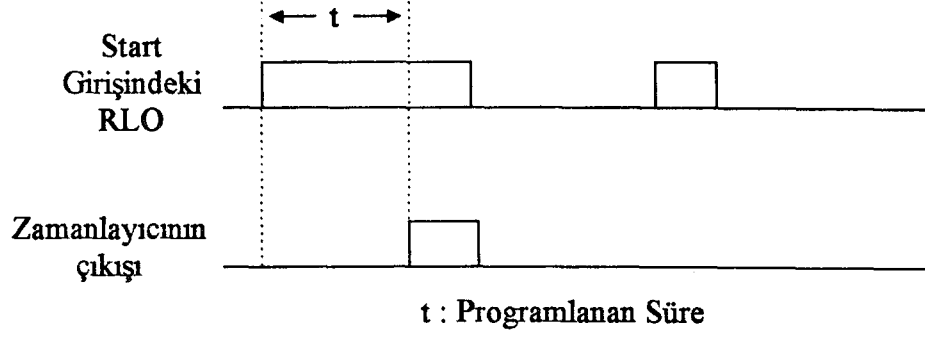
Girişindeki RLO'nun "0" dan "1" e geçmesiyle, çıkışı da (Q) aynı anda "1" olarak çalışmaya başlayan zamanlayıcının, belirlenen t süresi sonunda çıkışı sıfıra düşer. Bu sürede zamanlayıcının girişi "0" a düşse dahi belirlenen süre dolmadan çıkış "0" a düşmez (zamanlayıcı çalışmaya devam eder) (Şekil 4.14).



Şekil 4.14 Extended Pulse Timer

4.4.4.3 On Delay Timer (SR)

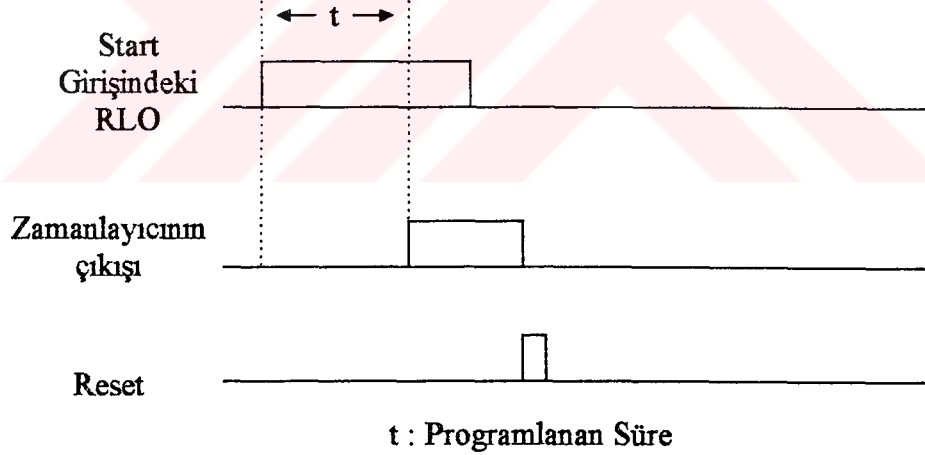
Girişindeki RLO'nun "0" dan "1" e değişmesiyle zamanlayıcı çalışmaya başlar. Programlanan t süresi sonunda zamanlayıcının çıkışı "1" konumuna geçer ve bu durum zamanlayıcının girişinin sıfıra düşmesine kadar devam eder (Şekil 4.15).



Şekil 4.15 On Delay Timer

4.4.4.4 Stored On Delay Timer (SS)

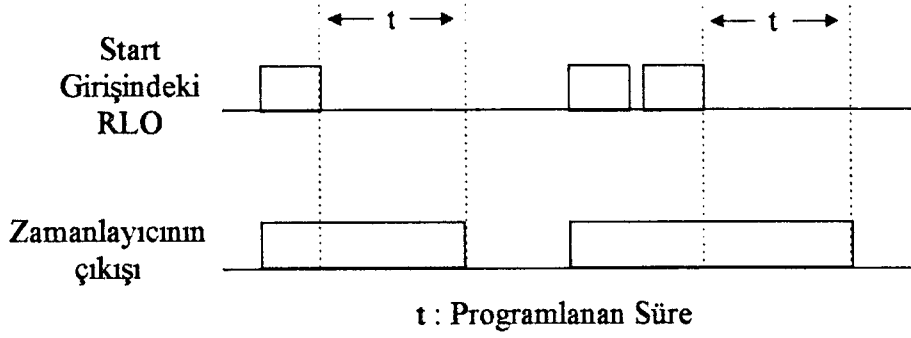
Zamanlayıcı girişindeki RLO'nun "0" dan "1" e değişmesiyle zamanlayıcı çalışmaya başlar. Programlanan süre sonunda zamanlayıcının çıkışı "1" olur. Bu zamanlayıcının "on delay" zamanlayıcısından farkı, programlama süresi içinde zamanlayıcının girişi "0" a düşse dahi, zamanlayıcının çalışmaya devam etmesidir. Ayrıca bu zamanlayıcının tekrar çalışabilmesi için mutlaka reset edilmesi gerekmektedir. Zamanlayıcının karakteristiği Şekil 4.16' da gösterilmiştir.



Şekil 4.16 Stored On Delay Timer

4.4.4.5 Off Delay Timer (SF)

Zamanlayıcı girişindeki RLO'nun "1" den "0" a geçmesiyle birlikte zamanlayıcı çalışmaya başlar ve programlanan süre sonunda zamanlayıcının çıkışı "0" olur. Karakteristiği Şekil 4.17' de gösterilmiştir.

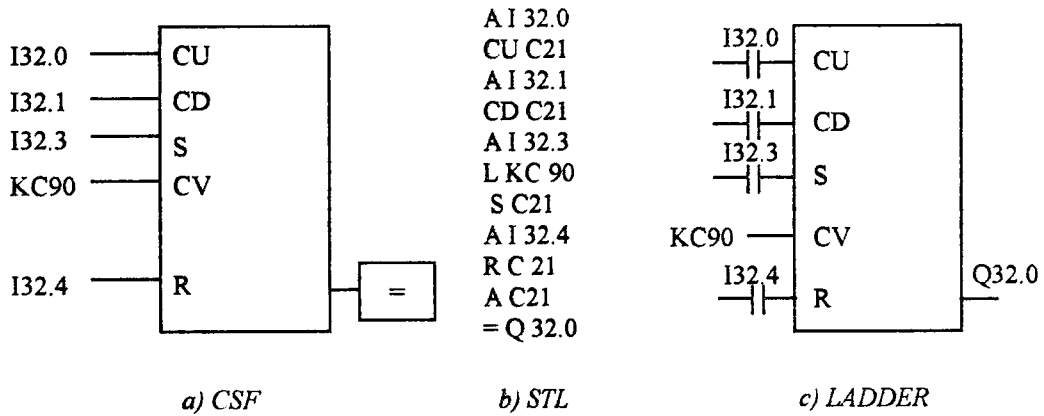


Şekil 4.17 Off Delay Timer

4.4.5 Sayıcılar

Step 5 programlama dilinde birbirinden bağımsız olarak birden fazla sayıcı programlaması yapılabilir. Programlamada Cn (counter) harfi ile ifade edilir ki; buradaki n kaçınıcı sayıcı olduğunu göstermektedir.

Sayıclar set girişindeki RLO'nun "0" dan "1" e deęişmesiyle set edilirler (bu sırada KC şeklinde ifade edilen sayıcı sabiti akümülatör l'e yüklenmiş olur). CU (count up : yukarı say) ve CD (count down : aşağı say) komutları sayıcının aşağı ve yukarı saymalarını sağlarlar. Bu iki komut RLO'nun "0" dan "1" e deęişmesine baęlı olarak sayma (arttırma ve azaltma) işlemlerini gerçekleştirirler. Bunun dışında set (S) komutu sayıcıya 0-999 arasında istenilen bir sayma deęerini set etmemizi sağlarken; Reset komutu da sayıcıyı sıfırlamamızı sağlar. CPU'nun yapısına baęlı olarak çeşitli sayılarda sayıcı kullanabiliriz. CSF, Ladder ve STL'de sayıcılara bir örnek Şekil 4.18' de gösterilmiştir.



Şekil 4.18 Sayıcı Gösterimi

Programda dikkat edilecek olursa I 32.0 "1" olduğu zaman, sayıcının değeri 1 artmakta; I 32.1 "1" olduğu zaman ise 1 azalmaktadır. CSF ve Ladder programlamalarında ise istenilen sayma değeri (burada 90) set girişiyle sayıcıya alınmaktadır. STL' de ise I 32.3'ün "1" olmasıyla ("0" dan "1" e geçmesiyle) sayıcının içeriğine bakılmaksızın değeri 90'a set edilecektir.

4.4.6 Bit Test İşlemleri

Bu tür işlemlerde herhangi bir operandın (Bit olarak) durumunu test etmek amacıyla çeşitli komutlar kullanılır. Bit test işlemlerinin yapılmasında zamanlayıcılar, datalar ve sayıcılar operand olarak kullanılır.

4.4.6.1 Bit Test İşlemlerinde Kullanılan Komutlar.

SU : Koşulsuz olarak bit'i "1" yap (Set bit unconditionally) (SU D 12.7 :DW12'nin 7. bitini koşulsuz olarak "1" yapar.

RU : Bit' i koşulsuz olarak "0" yap (Reset bit unconditionally). (RU D 103.7 : DW103'ün 7. bitini "1" yapar.

TB : Bit'in "1" olup olmadığını test eder (Test bit for "1"). (TB D18.3 : DW18'in 3. bitinin "1" olup olmadığını test eder ve bir olması durumunda RLO "1" olur).

TBN : Bit'in "0" olup olmadığını test eder. (Test bit for "0"). (TBN D17.2 : DW17'nin 2. bitinin "0" olup olmadığını test eder. Sıfır olması durumunda RLO "1" olur.

Step 5 programlama dilinde bu türdeki bit işlemleri dışında byte ve word işlemleri yapmamızı sağlayan komutlar vardır. Bunlar arasında transfer, yükleme, karşılaştırma ve aritmetik işlem komutlarını sayabiliriz.

4.4.7 Byte ve Word Komutları

4.4.7.1 Yükleme Komutları (L)

Kısaca L harfi ile gösterilen Load ile operandlardaki bilgiler (Q, I, F, ...) akümülatöre yüklenir. Şimdi bu işleme kısa bir örnek verelim :

L IB32 : IB32'nin içeriğini akümülatör1'e yükle

L IB42 : IB42'nin içeriğini akümülatör1'e yükle

Buradaki ikinci komutun işlenmesinden sonra akümülatör1'deki değer otomatik olarak akümülatör2'ye kayacaktır. Yani bu iki satır işlendiği zaman akümülatör1'de IB42' nin içeriği, akümülatör2'de de IB32' nin içeriği bulunacaktır.

4.4.7.2 Transfer Komutları (T)

Step 5'deki transfer komutlarıyla akümülatörde bulunan bilgiler operandlara transfer edilir. Kısaca T harfi ile gösterilir. Transfer işlemlerinde şu operandlar kullanılır: I (Girişler IW, IB), Q (çıkışlar QB, QW), Flaglar (FY,FW) ve Datalar (DW:DL,DR). Transfer işlemlerine kısa bir örnek verelim :

T IB32 : Akümülatör1'in değerini IB32'ye transfer et.

T QB30 : Akümülatörün değerini QB30'a transfer et.

Bu örnekler daha da çoğaltılabilir, ancak burada dikkat edilmesi gereken transfer işlemlerinde sabit değerlerin kullanılmayacağıdır. Sözgelimi T KH 00FF yanlış bir kullanım olacaktır.

4.4.7.3 Karşılaştırma Komutları

Bu komutlar akümülatör1 ve akümülatör2'deki iki sayıyı karşılaştırmak için kullanılırlar. Karşılaştırma işleminin sonucuna göre RLO'nun değeri "1" veya "0" olur. Karşılaştırma işlemlerinde kullanılan komutlar şunlardır:

! = F : Eşit olma durumu için karşılaştır.

> < F : Eşit olmama durumu için karşılaştır.

> F : Büyük olma durumu için karşılaştır.

> = F : Büyük eşit olma durumu için karşılaştır.

< F : Küçük olma durumu için karşılaştır.

< = F : Küçük eşit olma durumu için karşılaştır.

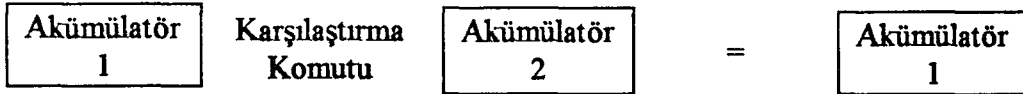
Bu karşılaştırma işlemlerine bir örnek verelim:

L IW32 : IW32'nin içeriğini akümülatör1'e yükle

L FW18 : FW18'in içeriğini akümülatör1'e yükle (Bu sırada akümülatör1'deki değer akümülatör2'ye geçecektir).

< F : Her iki değeri karşılaştır.

=F 0.0 : Karşılaştırma sonucu doğruysa (akümülatör1'deki değer, akümülatör2'deki değerden küçük ise) F 0.0 bitini "1" yap. STL ile yazılmış bu programın çalışması aşağıdaki şekilde gösterilmiştir (Şekil 4.19).



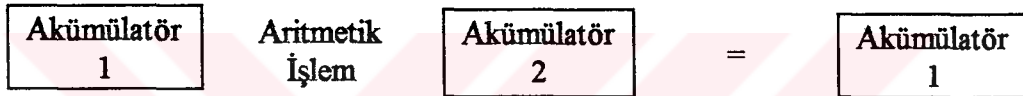
Şekil 4.19

4.4.7.4 Aritmetik İşlemlerde Kullanılan Komutlar

Bu tür komutlar iki tane sayıyı birbirinden çıkarmak veya birbirleriyle toplamak için kullanılırlar. Burada üzerlerinde işlem yapılan sayılar, akümülatör1 ve akümülatör2'deki sayılardır ve sonuç ise akümülatör1'e kaydedilir. Aritmetik işlemlerde kullanılan komutlar şunlardır :

- +F : Toplama işlemlerinde kullanılır.
- F : Çıkarma işlemlerinde kullanılır.

Yapılan işlem ise şöyle ifade edilebilir (Şekil 4.20)



Şekil 4.20

Şimdi her ikisine kısa bir örnek verelim:

L IW32

L FW18

+F : IW32 ve FW18'deki değerler toplanır

L DW18

-F : ve böylece elde edilen ara sonuçtan DW18'in içeriği çıkarılır. Sonuç akümülatör1'de saklanır.

Fonksiyon bloğu kullanılarak bu tür işlemler Ladder ve CSF olarak da yapılabilir.

4.4.7.5 Lojik Komutlar (AW, OW, XOW)

Akümlatör1 ve akümülatör2'deki sayıları lojik olarak AND, OR ve EXOR işlemlerine tabi tutarlar. Sonuç akümülatör1'de saklanır..

AW : AND işlemini yapar.

OW : OR işlemini yapar

XOW : EXOR işlemini yapar ve tüm bu işlemler akümülatör1 ve akümülatör2 sayıları arasında gerçekleştirilir. Buna bir örnek verelim :

L IW32

L FW18

AW : IW32 ve FW18'in içerikleri lojik olarak And'lenir

T QW32 : elde edilen sonuç QW32'ye aktarılır.

yukarıdaki örnekte QW32'nin içeriğini F857 ve FW18'in içeriğini de F549 olarak ele alırsak AND işlemi Şekil 4.21' de gösterildiği gibi gerçekleştirilir.

1 1 1 1 1 0 0 0 0 1 0 1 0 1 1 1

AND

1 1 1 1 1 0 0 0 0 1 0 1 0 1 1 1



1 1 1 1 1 0 0 0 0 1 0 1 0 1 1 1

Şekil 4.21 AND İşlemi

4.4.7.6 Kaydırma İşlemleri

Bu tür işlemlerde akümülatör1'deki sayıyı istenilen miktarlarda sağa ve sola kaydırmak için iki farklı komut kullanılır. Bunlar :

SLW (0-15) : Sola kaydır (Shift Left)

SRW (0-15) : Sağa kaydır (Shift Right)

Örnek :	Komut	Akümülatör1
	L KH 0F0F	0000111100001111
	SLW 4	1111000011110000

Akümülatör1'e alınan 0F0F değeri ikinci satırda 4 bit sola kaydırılmaktadır ve sonuç yine akümülatör1'dedir.

4.4.7.7 Tümlleme İşlemleri

Bu işlemlerle akümülatör1'deki bir sayının 2'ye ve 1'e tümlleme işlemleri gerçekleştirilir. Elde edilen sayı yine akümülatör1'dedir. Bu işlemlerde kullanılan komutlar :

CFW : Akümülatör1'deki sayının 1'e tümleyenini alır.
CSW : Akümülatör1'deki sayının 2'ye tümleyenini alır.

Örnek:	Komut	Akümlatör1
	L IW32	0100110001101101
	CFW	1011001110010010

1'e Tümlleme işleminde her bit'in ayrı ayrı tersi alınır. 2'ye tümlleme işleminde ise 1'e tümlenen sayıya 1 eklenir. Sözelimi IW32'deki sayının 2'ye tümleyeni almırsa 1011001110010011 sayısı elde edilir.

4.4.8 Diğer Komutlar

4.4.8.1 Arttırma ve Azaltma Komutları

Akümlatör1'deki sayının belli miktarlarda artırılıp, azaltılmasını sağlayan komutlardır. Bunlar :

D 0-255 : Akümülatör1'deki değeri 0-255 değeri kadar azaltır.
I 0-255 : Akümülatör1'deki değeri 0-255 değeri kadar arttırır.

4.4.8.2 Dallanma Komutları

Dallanma komutları iki kısımda ele alınabilir. (1) Çeşitli bloklara yapılan dallanmalar. (2) Program içindeki herhangi bir satıra koşullu veya koşulsuz olarak yapılan dallanmalar. Örneğin *JU Pbn* derken program koşulsuz olarak bu noktada n. program bloğuna dallanacaktır. Burada kullanılan komutlar şunlardır :

JU (PB,FB, . . .) : Koşulsuz olarak bloklara dallan.

JC (PB,FB, . . .) : Bir koşula bağlı olarak bloklara dallan.

Program sırasında kullanılan ve bir satırdan başka bir satıra dallanmamızı sağlayan komutlar ise şunlardır :

JU : Koşulsuz dallan

JC : Koşullu dallan

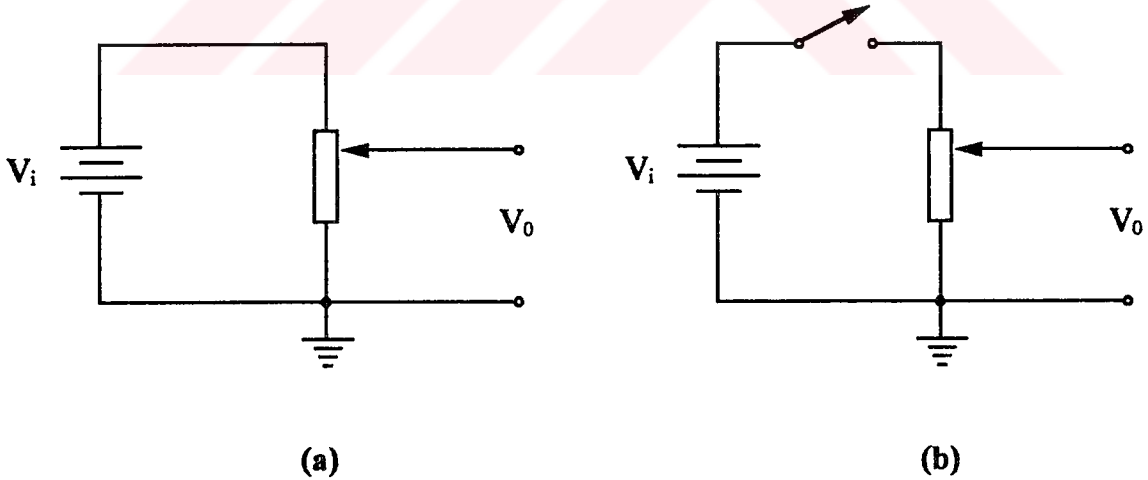
Bu komutlar kullanılırken ise dallanılmak istenen satıra bir isim verilerek kullanılır. sözelimi SON etiketli satıra dallamak istediğimizde kullanacağımız komut $JU = SON$ şeklinde olacaktır.

V. ANALOG / SAYISAL ÇEVİRİCİLER

5.1 Analog ve Sayısal (Dijital) İşaretler

Elektronikte iki temel büyüklük vardır, bunlar : Analog ve sayısal büyüklüklerdir. Analog büyüklükler bir aralıkta herhangi bir değer alabilirken, sayısal büyüklükler ise aynı aralıkta ancak belirli bazı değerleri alabilirler. Örneğin Şekil 5.1’de gösterilen devrede çıkış gerilimi 0-5V arasında istenilen bir değere ayarlanabilir ve bir analog büyüklüktür. Bunun yanı sıra basınç, sıcaklık, ağırlık gibi birçok büyüklük de analog büyüklükler arasında sayılabilir. Örneğin hareketli sargılı bir voltmetre, iğnesinin sonsuz sayıda farklı noktayı gösterebilmesinden dolayı analog bir cihazdır.

Dijital bir sinyal ise sadece iki farklı durumla ifade edilebilir. Başka bir deyişle sadece belli değerleri alabilir. Şekil 5.1b’de V_0 çıkış büyüklüğü dijital bir büyüklüğü ifade etmektedir. Burada anahtarın kapalı olması durumuna göre çıkış ya vardır veya yoktur. Başka bir ifadeyle “0” veya “1” dir. Bir kare dalganın yükselme ve düşme sürelerinin çok ani olduğu kabul edilirse kare dalga dijital işarete örnek olarak gösterilebilir. Sayısal devreye en iyi örnek isminden de anlaşılacağı gibi sayısal bir voltmetre gösterilebilir. Çünkü bu voltmetre üzerindeki göstergesi sadece sonlu sayıda değerleri gösterebilmektedir (3.92, 8.65...vb).



Şekil 5.1 Analog ve Sayısal İfade Eden Devreler

(a) Analog çıkışlı Devre

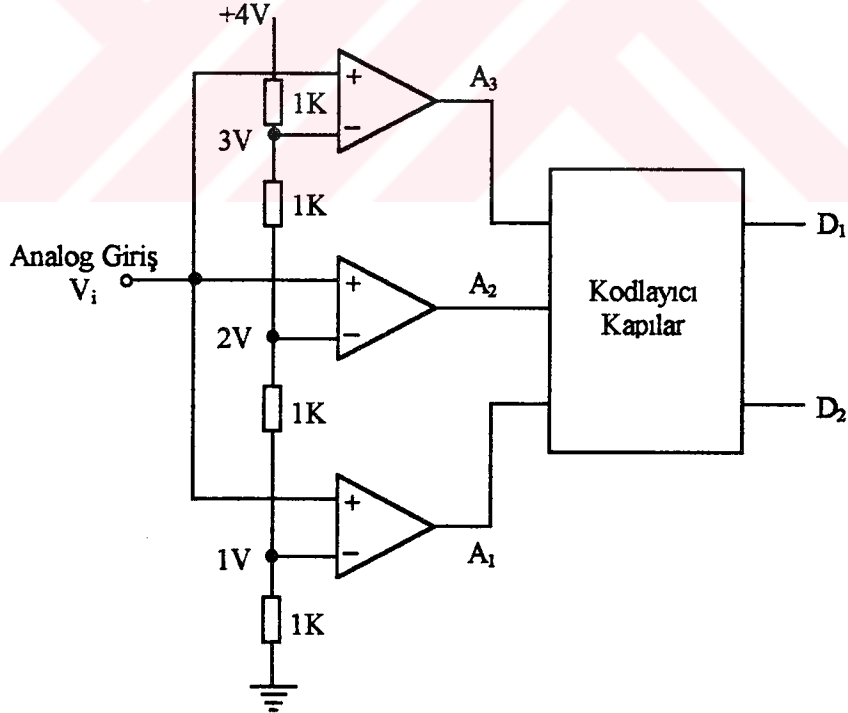
(b) Sayısal Çıkışlı Devre

Bunun yanı sıra birçok bilgisayar ve hesap makineleri de 0 ve 5V gibi sayısal işaretlerle çalışır. Bu şekildeki bir bilgisayar ve hesap makinelerinde bu iki değer ikili kodlar halinde saklanıp işlenerek birçok farklı değer elde etmemizi sağlayabilirler. Sözelimi 8 bitlik bir bilgisayarda $2^8 = 256$ farklı sayıda işaret gönderilebilir. Bir analog büyüklük hafızada saklanmak istendiğinde genellikle dijital bir büyüklüğe çevrilir. Bu iş içinde analog-dijital çeviriciler kullanılır. Bu çevrim yapılmadan önce işaret analog olduğundan bir aralıkta herhangi bir değeri alabilirken çevrim yapıldıktan sonra dijital dönen işaret yine aynı aralıkta, ancak sınırlı sayıda değerler alabilecektir.

Çeşitli tiplerde A/D çeviriciler vardır. Şimdi bunlardan en çok kullanılanlarından bazılarını değinelim.

5.2 Paralel Karşılaştırmalı veya Flaş A/D Çeviriciler

Teorik olarak en kolay ve en hızlı A/D çeviricilerindendir. Gerilim bölücülerinden ve karşılaştırmacılardan meydana gelen bu A/D çeviricinin 3-bitlik bir basit uygulaması Şekil 5.2'de gösterilmiştir. Şekilden de görülebileceği gibi dirençlerden meydana gelen gerilim bölücü üç karşılaştırmacının eşik gerilimlerini ayarlamaktadır. Bölücünün üstündeki gerilim ise anlamlı



Şekil 5.2 Paralel Karşılaştırmalı A/D Çevirici

maksimum giriş gerilimini ifade etmektedir. Buradaki karşılaştırıcıların pozitif girişlerindeki gerilimlerin, negatif girişlerinden daha büyük olmaları durumunda çıkışları sayısal olarak "1" dir. Bunun tam tersi durumunda ise yani, pozitif girişlerindeki gerilimlerin, negatif girişlerindeki gerilimlerden küçük olması durumunda ise karşılaştırıcıların çıkışları sayısal olarak "0" olacaktır. Şekil 4.2'de örnek olarak verilen devrenin çıkışları Tablo 1'de gösterilmiştir.

Burada dikkat edilecek olursa çıkış geriliminin 1V dan az olması durumunda hiçbir karşılaştırıcının çıkışı "1" olmayacaktır. Eğer sadece en düşük eşiğe sahip karşılaştırıcının çıkışı "1" ise buradan bu karşılaştırıcının giriş geriliminin 1-2V arasında olduğu anlaşılır. Buna benzer şekilde giriş geriliminin (analog giriş işaretinin) seviyesine göre diğer karşılaştırıcılarda "1" veya "0" seviyelerine geçerler. Bu durum Şekil 5.3'de gösterilmiştir.

V_i (V)	A1	A2	A3	D1	D0
0 - 1	0	0	0	0	0
1 - 2	1	0	0	0	1
3 - 4	1	1	0	1	0
4 - 5	1	1	1	1	1

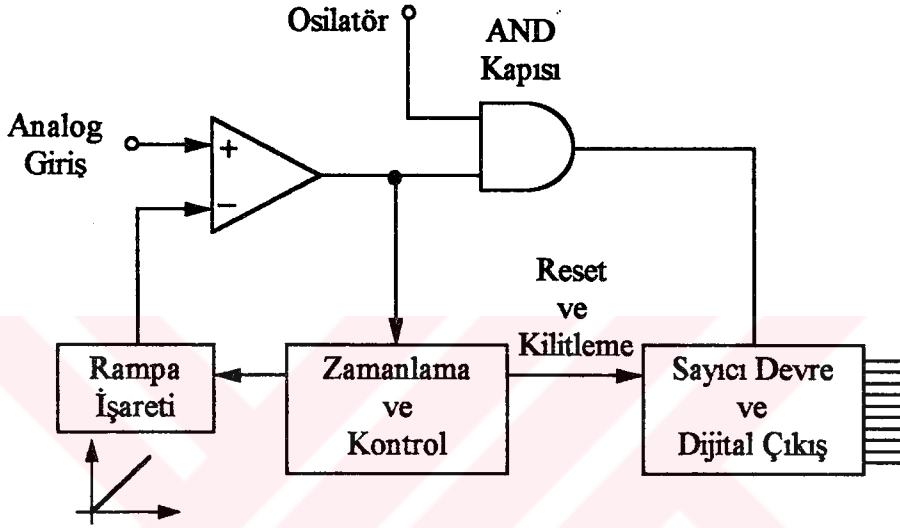
Şekil 5.3 Giriş Gerilimleri İçin Çıkışlar

Bu tür A/D çeviriciler hızlı olmalarına karşın bitlerin sayısı arttırıldığında, yüksek sayıda karşılaştırıcılara ihtiyaç duymalarından dolayı sorun yaratmaktadırlar. Sözgelimi 4-bitlik bir A/D için 15 karşılaştırıcı, 3-bitlik bir A/D çevirici için 7 karşılaştırıcı gerekmektedir. Bir N-bitlik çevirici ise 2^N-1 tane karşılaştırıcı gerektirir ki; buradan 8-bitlik bir çevirici için $2^8-1 = 255$ karşılaştırıcı gerektiği ortaya çıkar. Bu da bu tip A/D çeviricilerin çok fazla sayıda karşılaştırıcıya ihtiyaç göstermelerinden dolayı büyük bir dezavantaja sahip olduklarını göstermektedir

5.3 Tek Eğimli A/D Çevirici

Tek eğimli A/D karşılaştırıcı Şekil 5.4'te gösterildiği gibi, bir rampa üretici, ikilik sayıcı ve karşılaştırıcıdan meydana gelmiştir. Çevrim başlamadan önce rampa kaynağı ve sayıcı devresi kontrol devresi tarafından sıfıra resetlenir. Bu ilk durumdan sonra karşılaştırıcının artı girişine dijital olarak çevrilmek istenen analog işaret uygulanır ve karşılaştırıcının diğer girişini oluşturan rampa

çıkışı sıfır olduğundan karşılaştırıcının çıkışı "1" konumundadır. Bu durumda karşılaştırıcının çıkışıyla birlikte bir AND kapısına uygulanmış bulunan saat darbesi sayıcıya ulaşarak çalışmasını sağlar ve aynı zamanda kontrol elemanı da karşılaştırıcının çıkışının "1" olduğunu görerek rampa kaynağını da çalıştırır. Rampa gerilimi belli süre sonunda giriş gerilimini (analog işaret) aşarak karşılaştırıcının çıkışının "0" olmasına neden olur. İşte bur durumda AND kapısının girişlerinden biri "0" olacağından saat işaretinin sayıcı devresine gitmesi engellenir ve dolayısıyla sayıcı durur. Böylece analog giriş işaretinin dijital karşılığı sayıcı devresinin çıkışlarında ortaya çıkar.



Şekil 5.4 Tek Eğimli A/D çevirici

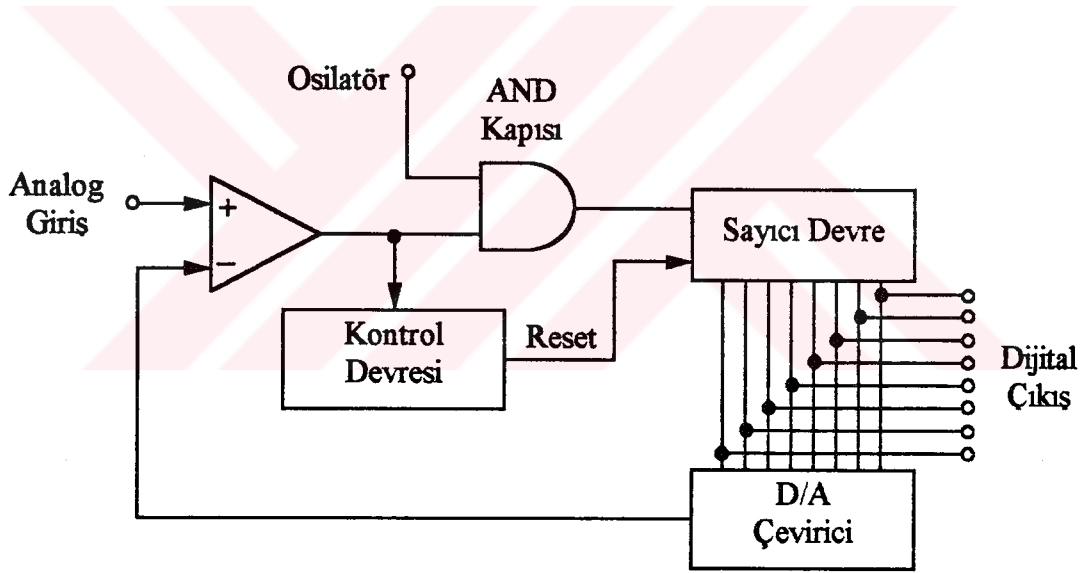
Buradaki tek eğimli A/D çeviricilerin yanı sıra çift eğimli A/D çeviriciler de vardır. Bu tür çeviriciler yapı olarak tek eğimli A/D çeviricilere benzemekle birlikte içerdiği bir entegratör devresiyle tek eğimli A/D çeviricilere nazaran daha kararlıdır. Bunun sebebi Tek eğimli A/D çeviricilerde zaman, sıcaklık veya giriş gerilimi duyarlılığının rampa kaynağında değişikliklere sebep olmasıdır.

Bu tür A/D çeviricilerin yanı sıra yapılarında D/A çeviriciler içeren A/D çeviricilerde, çok kullanılan A/D çeviriciler arasındadırlar. Bunlar arasında sayıcı tipten, izlemeli tipten ve ardışıl yaklaşımli olanlarını sayabiliriz.

5.4 Sayıcı Türden A/D Çevirici

Bu tür A/D çeviriciler Şekil 5.5'te de görülebileceği gibi karşılaştırma devresi, sayıcı devresi, kontrol devreleri ve D/A çeviricilerden meydana gelmişlerdir (Şekil 5.5).

Devrenin ilk durumunda sayıcı devresi kontrol devresi tarafından resetlenir. Böylece bu durumda analog işarete çevrilerek karşılaştırıcının eksi girişine uygulanan gerilim sıfırdır. Bu işlemlerden sonra dijitalle çevrilmek istenen analog işaret karşılaştırıcının artı girişine uygulanır. Karşılaştırıcının eksi girişi sıfır olduğundan çıkışı "1" olur ve saat darbesi saat darbesinin AND kapısı üzerinden sayıcıya ulaşmasına izin verir. Bu durumdan sonra sayma işlemine geçen sayıcı devresinin çıkışı bir D/A çevirici yardımıyla analog işarete çevrilerek, dijital işarete çevrilmek istenen analog işaretle karşılaştırılır. Bu işaretin karşılaştırıcının artı girişine uyguladığımız işareten büyük olduğu anda ise karşılaştırıcının çıkışı "0" olarak sayma işleminin durmasını sağlar. Böylece sayıcının çıkışındaki değer bizim dijital işarete çevirmek istediğimiz analog işaretimizin (karşılaştırıcının artı girişine uyguladığımız işaretin) dijital karşılığıdır.

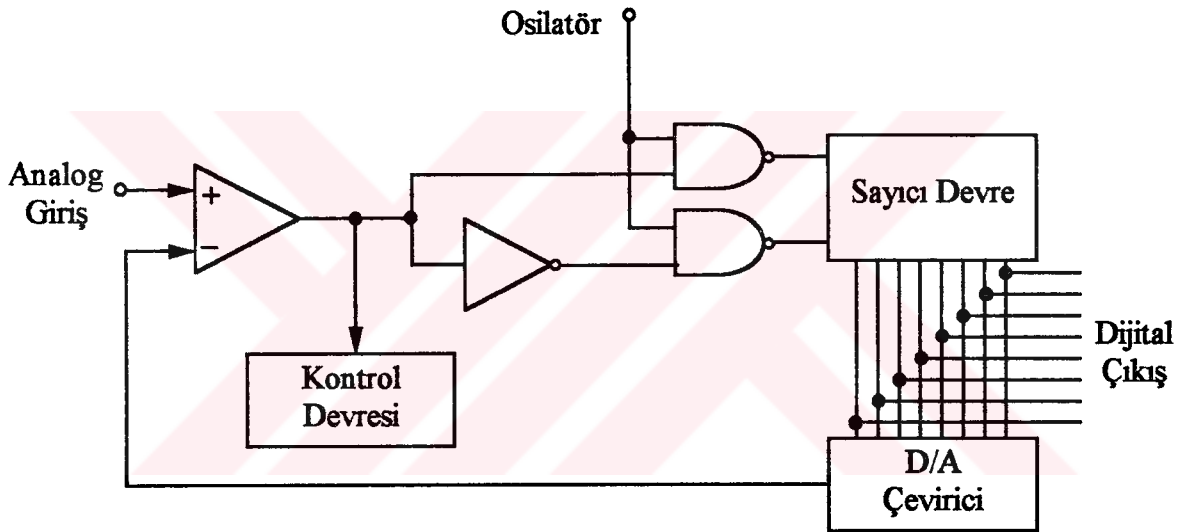


Şekil 5.5 Sayıcı Türden A/D Çevirici

Bu türdeki A/D çeviricinin dezavantajları D/A çeviriciye ihtiyaç göstermesi ve her çevrimde sayma işleminin sıfırdan başlamasının gerekmesidir. Bu tür A/D çeviriciye benzer başka bir çevirici ise "izlemeli A/D çevirici" adı verilen A/D çeviricidir. Şimdi bunu inceleyelim.

5.5 İzlemeli Türden Analog Dijital Çevirici

Bu tür A/D çevirici bir önceki A/D çevirici tipine benzemekle birlikte ondan daha hızlıdır. Bunun sebebi bu A/D çeviricide sayıcı olarak hem aşağı ve hem de yukarı sayabilen bir sayıcının kullanılmış olmasıdır (Şekil 5.6). Dikkat edilecek olursa bir önceki “sayıcı türden A/D çevirici” tipinde sadece yukarı sayabilen bir sayıcı kullanılmıştır. Buradaki osilatör karşılaştırıcının çıkışındaki NAND kapıları yukarı veya aşağı sayma girişlerine bağlanır. Çevrimin başındaki herhangi bir giriş gerilimi için D/A çeviricinin çıkışı “0”dır ve dolayısıyla karşılaştırıcının çıkışı sayısal olarak “1”dir. Bu da osilatörün yukarı sayma girişine bağlanması demektir. Bu durumda sayıcı D/A çeviricinin çıkışı dijitale çevrilecek olan analog işaretten büyük oluncaya kadar yukarı saymaya devam eder.



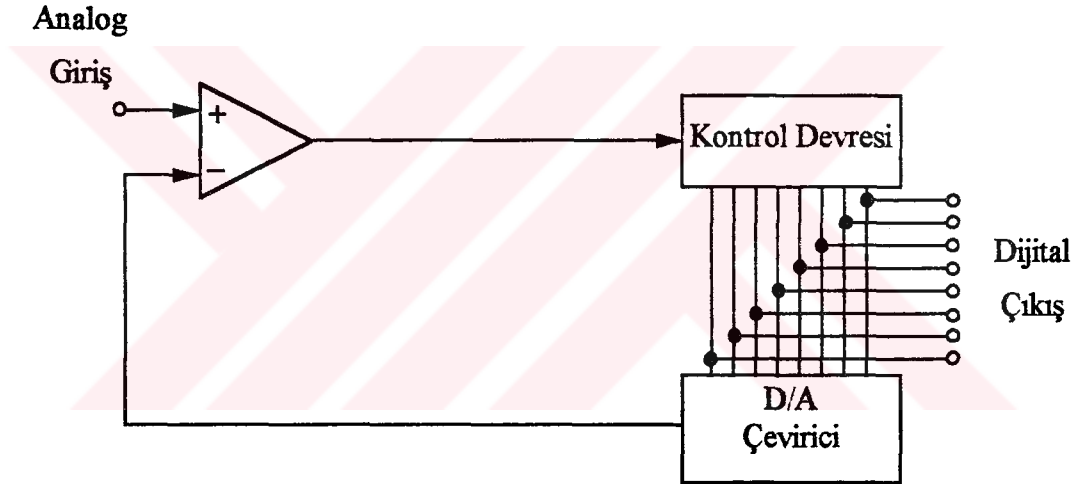
Şekil 5.6 İzlemeli Türden A/D Çevirici

Bu durumdan sonra karşılaştırıcının çıkışı “0” seviyesine düşer ve dolayısıyla osilatör sayıcının aşağı sayma (CD) girişine bağlanmış olur. Ancak, eğer analog giriş işaretimiz değişmemişse, sayıcının aşağı saymasıyla birlikte D/A çeviricinin karşılaştırıcının eksi girişine uygulanmış olan çıkışı, analog işaret seviyesinin altında kalacağından karşılaştırıcının çıkışı tekrar “1” olur ve sayma işlemi bu kez yukarıya doğru tekrarlanır. Görülüyor ki, bu tip bir A/D çeviricide sabit bir analog işaretin dijitale çevrilmek istenmesi durumunda çıkış işaretinin en küçük değerlikli çıkış biti salınım yapacaktır. Bu da bu tip sayıcılar için bir dezavantajdır. Ancak sinüsoidal herhangi

bir işaret sayısallaştırılırken bir önceki sayıcı tipten A/D çeviricide olduğu gibi sayma işlemi her seferinde sıfırdan başlamak yerine, bir örnekten diğerine geçerken sadece birkaç sayım yukardan veya aşağıdan başlaması bu tip çeviriciler için bir avantajdır. Bu yapıdaki başka bir çevirici tipi de “ardışıl yaklaşımlı A/D çevirici” adı verilen çeviricidir. Bunun sayıcı türden olan A/D çeviricilerine karşı en önemli üstünlükleri salınım yapmamaları ve daha hızlı olmalarıdır. Şimdi böyle bir A/D çeviriciyi daha ayrıntılı olarak inceleyelim.

5.6 Ardışıl Yaklaşımlı A/D Çevirici

D/A Çevirici kullanan başka bir A/D çeviricidir. Ancak bu türden bir A/D çeviricinin D/A çevirici kullanan sayıcı türündeki diğer A/D çeviricilerine üstünlüğü N saat darbesiyle N-bitlik bir çözünürlüğe ulaşmasıdır. Sözelimi 8-bitlik bir A/D çeviricide 8 saat darbesi sonunda analog işaretimiz dijitale çevrilmiş olacaktır.



Şekil 5.7 Ardışıl Yaklaşımlı ADC

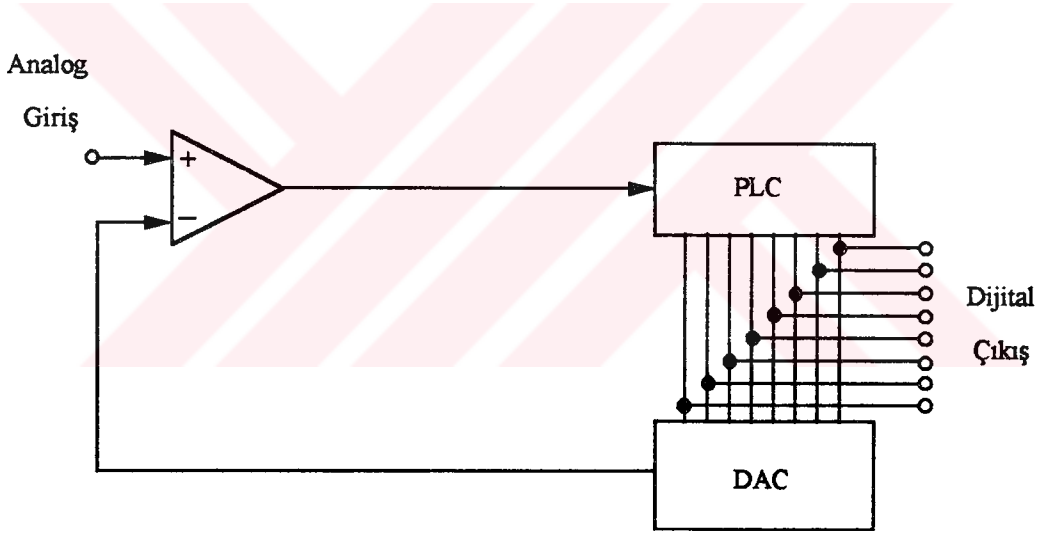
Şekil 5.7’ de gösterilen “Ardışıl Yaklaşımlı A/D Çevirici” de ilk durumda kontrol devresi tarafından dijital çıkışın MSB bit’ i (en yüksek değerlikli bit) “1” yapılır. Bu durumda D/A çevirici tarafından analog işarete çevrilen çıkış, karşılaştırıcının eksi girişine uygulanır ve analog girişle karşılaştırılır. Analog girişin büyük olması durumunda karşılaştırıcının çıkışı “1” olur ve DAC’ ın çıkışının en büyük değerlikli bu biti “1” olarak kalır, aksi durumda “0” yapılır. Daha sonra ikinci en yüksek değerlikli bit 1 yapılır ve karşılaştırıcının çıkışına göre “1” olarak bırakılır veya “0” a getirilir. Böylece bu işlem çıkışın en küçük değerlikli bitine kadar devam eder ve sonuçta çıkışta elde edilen sayı Analog Giriş’in sayısal karşılığı olur.

VL BULGULAR

İlk olarak S5 90U PLC' si kullanılarak "ardışıl yaklaşımlı" ve "sayıcı türden" analog/dijital çevirici gerçekleştirilmiş ve daha sonra PLC ile kombinasyonel lojik devre tasarımı anlatılmıştır. Yazılan programlar EKLER kısmındadır.

6.1 PLC ile ADC Uygulaması

Burada 5. bölümde genel amaçları ve çalışma prensipleri tanıtılmış olan ADC' lerden "Ardışıl Yaklaşımlı ADC" ve "Sayıcı Türden ADC" nin kontrol elemanı olarak PLC kullanılması durumunda çalışmaları anlatılacaktır. Bu şekildeki bir ADC' nin blok diyagramı aşağıdaki şekildeki gibidir (şekil 6.1).



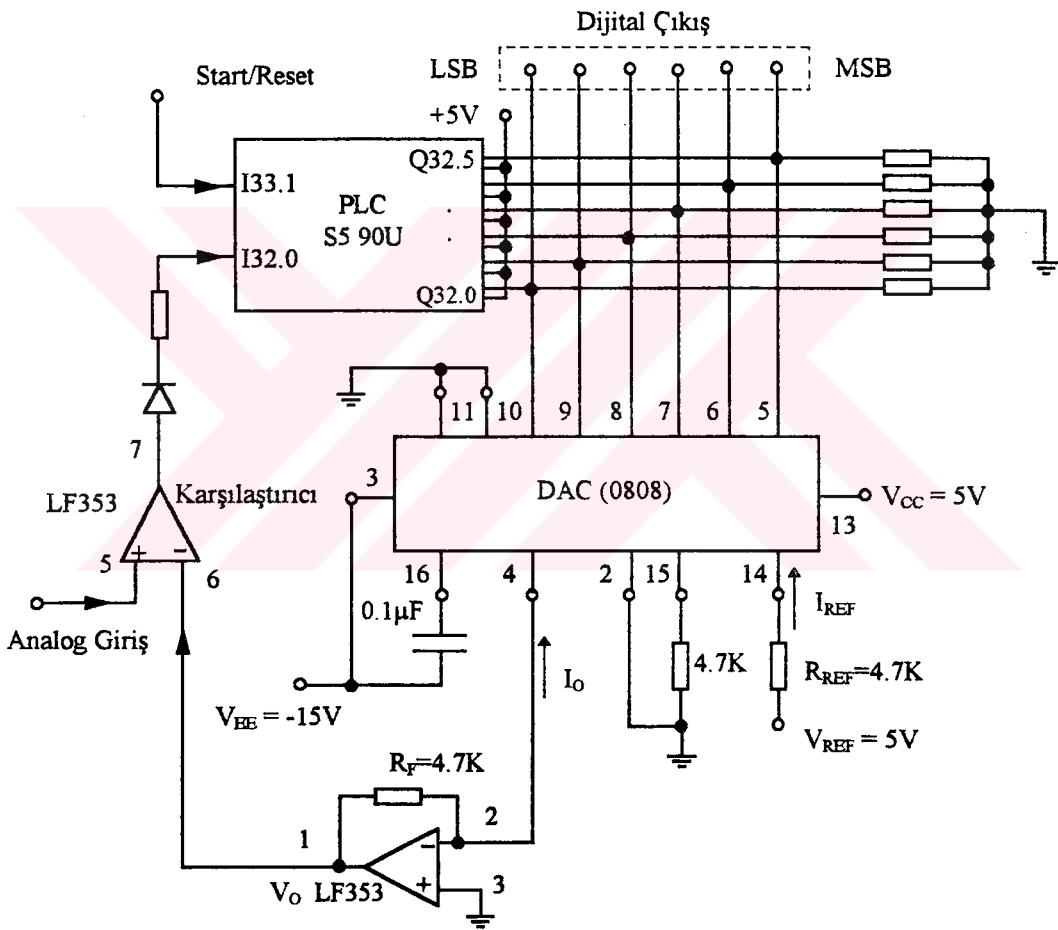
Şekil 6.1 Kontrol Devresi Olarak PLC Kullanan ADC

Blok diyagramı Şekil 6.1'de gösterilen ADC PLC'nin programlanmasına bağlı olarak gerek ardışıl yaklaşımlı ve gerekse sayıcı türden olarak çalıştırılabilir. Bu kısımdaki uygulamada Şekil 6.2 'de gösterilen devre kurulmuş ve Step 5 programlama dili kullanılarak PLC, devrenin her iki türden ADC olarak çalışmasını sağlayacak şekilde programlanmıştır.

Devrede kontrol elemanı olarak 6 adet çıkış (röle çıkışı) ve 10 adet girişe sahip s5 90U PLC' si kullanılmıştır. Uygulamada seçilen PLC'nin 6 tane çıkışa sahip olmasından dolayı gerçekleştirilen ADC'de 6 bitlik olmuş ve dolayısıyla devrede kullanılan 0808 8 bitlik DAC

entegresinin de 6 bit olarak çalışmasını sağlamak amacıyla en küçük değerlikli iki biti sürekli sıfır seviyesinde tutulmuştur.

S5 90U' nun 10 girişinden sadece ikisi kullanılarak bunlardan, I32.0' a karşılaştırıcının çıkışı uygulanmış ve I33.1 ise Start/Reset düğmesi olacak şekilde programlanmıştır. Q32.0 ile Q32.5 arasındaki 6 tane çıkış ise DAC'ın girişlerini ve çevrim sonunda ise dijital çıkışları oluşturmaktadır. Devre, I33.1'in "1" konumuna getirilip tekrar "0" konumuna geçmesiyle birlikte çalışmaya başlar. Ancak bu işlem PLC'nin tarama zamanından daha kısa sürede gerçekleşmemesine dikkat edilmesi gerekir. Aksi takdirde PLC I33.1'deki bu değişimi algılayamayacak ve sistem çalışmayacaktır.



Şekil 6.2 Kontrol Devresi Olarak PLC Kullanan ADC

Analog işaretin dijital işarete çevrilme süreci içinde (start verilmiş ancak dijital çevirme işlemi tamamlanmamış durumda) I33.1'e darbe verilmesiyle birlikte çevrim yeniden başlayacaktır. Ancak, çevrim işlemi bittikten sonra devre, ilk baştaki konumuna dönecek ve

analogtan dijital çevrim işleminin tekrar başlatılması (I33.1 ile start işareti gönderilerek) gerekecektir.

DAC'm çıkışı olan I_O , akım referansı I_{REF} ile dijital giriş oranlı olarak bir akım çekme özelliğine sahiptir. Buna göre çıkış akımını aşağıdaki şekilde ifade edebiliriz.

$$K = \left(\frac{Q_{32.5}}{2} + \frac{Q_{32.4}}{4} + \frac{Q_{32.3}}{8} + \frac{Q_{32.2}}{16} + \frac{Q_{32.1}}{32} + \frac{Q_{32.0}}{64} \right) \quad \text{olmak üzere,}$$

$$I_O = I_{REF} \cdot K = \frac{V_{REF}}{R_{REF}} \cdot K \quad \text{olur.}$$

Bu şekilde gösterilen çıkış akımı daha sonra birim kazançlı bir opamp devresinden geçirilerek V_O çıkış gerilimine dönüştürülür. Buna göre V_O çıkış gerilimi ;

$$V_O = I_O \cdot R_F = V_{REF} \cdot \frac{R_F}{R_{REF}} \cdot K \quad R_{REF} = R_F \quad \text{seçilerek}$$

$$V_O = V_{REF} \cdot K \quad \text{olarak elde edilir.}$$

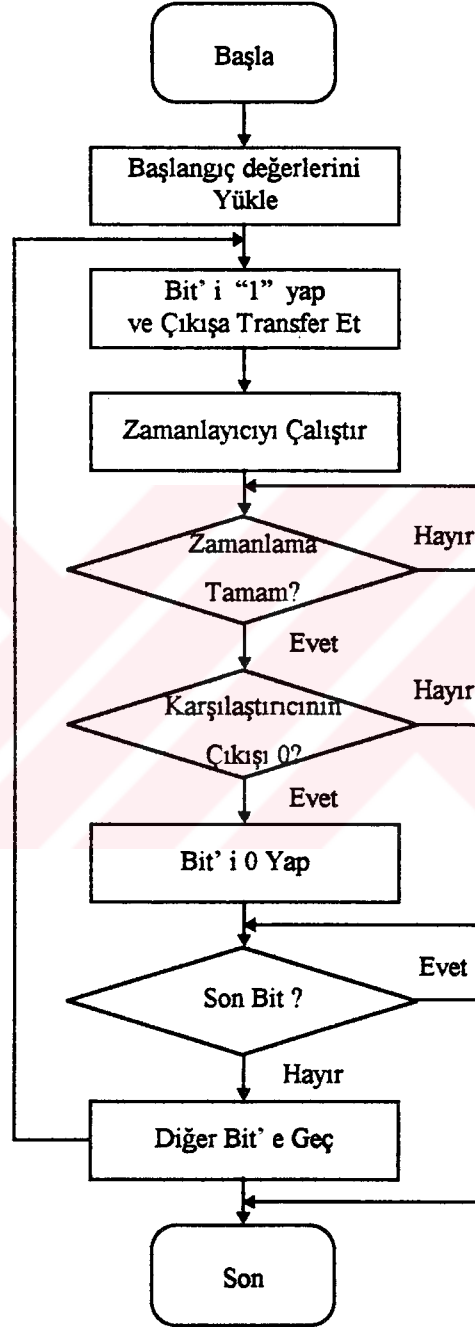
Karşılaştırıcının eksi girişinin artı girişinden büyük olması durumunda, PLC' nin girişinde (I32.0) "0" seviyesinin (0-5V) oluşmasını sağlamak amacıyla bir diyottan yararlanılmıştır.

6.1.1 Devrenin Ardışıl Yaklaşımlı ADC Olarak Çalışacak Şekilde PLC' nin Programlanması

Devrenin çalıştırılmasıyla birlikte, PLC tarafından en yüksek değerlikli bit 1 konumuna getirilerek DAC' m çıkışı giriş işaretiyle (Analog İşaret' le) karşılaştırılır. Giriş işaretinin büyük olması durumunda PLC' nin girişlerinden birine uygulanmış olan karşılaştırıcının çıkışı "1" seviyesine geçer. Bu girişin yüksek olduğunu gören PLC ,(DAC çıkışı henüz giriş işareti seviyesine ulaşmadığından) en yüksek değerlikli bu bit' i "1" konumunda bırakır. Ters durumda, yani DAC çıkışının giriş işaretinden büyük olması durumunda, karşılaştırıcının konumu "0" seviyesine geçer ve PLC tarafından algılanan bu durum en yüksek değerlikli bit' in tekrar "0" seviyesine çekilmesini sağlar. Bu işlem, sırasıyla en düşük değerlikli bit' e kadar

devam eder. Böylece, dijital çıkış sayısı kadar bir çevrim sonunda, analog giriş işaretimiz dijitale çevrilmiş olur.

Çalışma prensibi kısaca bu şekilde olan ardışıl yaklaşımlı ADC' de PLC' nin programlanmasına geçilmeden önce program akış diyagramı şu şekilde oluşturulmuştur.



Şekil 6.3

Program üç bloktan meydana gelmiştir. Bunlardan OB1 program çalıştırılır çalıştırılmaz programın ilk işlendiği bloktur. Programda bu blok dikkat edilecek olursa , sadece FB1

bazı işlemleri yapmamıza olanak vermemesinden dolayıdır. Sözelimi programda kullanılan MOVE komutu sadece Fonksiyon bloklarında kullanılabilir. Programı segment segment açıklamaya geçmeden önce programın genel olarak ne yaptığına bir bakalım.

Program çalıştırılır çalıştırılmaz ilk adımda FW15' e 0000000000100000 ve FW10' a ise 0000000000000000 değeri yüklenir ve OR işlemine sokularak çıkışa verilirler. Daha sonra çevrim boyunca karşılaştırıcının çıkışının sıfır olduğu durumlarda FW15 bir bit sağa çevrilerek bu işlem tekrarlanır (FW15 bir bit sağa çevrildikten sonra FW10 ile OR işlemine sokularak sonuç çıkışa verilir). Ancak çevrim sırasında karşılaştırıcının çıkışı "1" ise FW10' un o bite karşı düşen bit' i "1" yapıldıktan sonra işleme devam edilir. Programı bu şekilde açıkladıktan sonra segment segment açıklamasına geçebiliriz.

Segment 1 : I33.1' in 1 olmasıyla birlikte F5.0' ı set eder.

Segment 2 : I33.1' in 1 olmasıyla birlikte zamanlayıcıyı sıfırlar.

Segment 3 : I33.1' in 1 olmasıyla birlikte FW10' a 0000000000000000 değerini yükler.

Segment 4 : I33.1' in 1 olmasıyla birlikte FW15' e 0000000000100000 değerini yükler.

Segment 5 : I33.1' in olma veya F5.0' m "0" olma durumlarını F5.1' de saklar. (bu durum I33.1' in "0" veya F5.0' m "1" olması durumlarında çevrimin başlaması için kullanılmıştır dolayısıyla F5.0' m sıfır olması durumu çevrim devam edebilir anlamındadır).

Segment 6 : Bekleme süresi dolmuşsa (zamanlayıcı durmuşsa) ve F5.0 "0" sa FW10 ile FW15 Or işlemine sokularak sonuç çıkışa (QW31 : QB31+QB32) transfer edilir.

Segment 7 : Çevrim devam işareti varsa ve zamanlanan süre dolmuşsa zamanlayıcıyı yeniden çalıştırır.

Segment 8 : çevrim devam işareti varsa, zamanlanan süre dolmuşsa ve karşılaştırıcının çıkışının bağlı olduğu I32.0 "1" ise FB2' ye dallanılmasını sağlar.

Segment 9 : Son bitin işlenip işlenmediğini FW15' in değerine bakarak kontrol eder ve istenen durum gerçekleşmişse F5.0' ı reset eder ve çevrim bir daha ki start işaretine kadar bekler.

Segment 10 : FW15' i bir bit sağa kaydırır.

FB2 : Programdaki 2.Fonksiyon Bloğunda, (FB2) FW15 ile FW10 OR işlemine sokularak çıkış FW10 değerine yazılmaktadır.

6.1.2 Devrenin Sayıcı Türden ADC Olarak Çalışacak Şekilde, PLC' nin Programlanması

Programda PLC bir sayıcıyı gibi çalıştırılarak çıkışı DAC' m çıkışlarına uygulanmakta ve sayma işleminin her adımında DAC' m çıkışı ile analog giriş karşılaştırılmaktadır. İlk durumda analog giriş büyük olduğundan, karşılaştırıcının çıkışı normalde "1" dir. Sayma işlemi sırasında karşılaştırıcının çıkışının "0" olduğu anda sayıcı çıkışının analog giriş işaretimize ulaştığı anlaşılıp çevrim kesilir. Böylece analog giriş işaretimizin dijital karşılığı PLC' nin çıkışlarında elde edilmiş olur.

Program sırasında bir sayıcı kullanılarak sıfırdan itibaren saydırılmış ve değeri doğrudan çıkışa transfer edilerek DAC' m girişlerine uygulanmıştır. Her çevrim sırasında DAC' m çıkışı ile analog giriş karşılaştırılmış ve ilk durumda "1" olan karşılaştırıcının çıkışının "0" a düşmesiyle program kesilmiştir. Bu arada karşılaştırıcının değeri bir arttırıldıkta hemen sonra çıkışa transfer edilmemiş, arada bir zamanlayıcı kullanılarak çevrim işleminin yavaşlatılması sağlanmıştır. Böylece ADC' nin çevrim hızının, zamanlayıcının değeri değiştirilerek istenildiği gibi değiştirilebilmesine imkan tanınmıştır.

Programın segment-segment açıklaması şu şekildedir :

Segment 1 : I33.1' in "1" olmasıyla F5.0 set edilir. Bu çevrimin başlatılması için atılan ilk adımdır ve çevrimin kesilmesi F5.0' m reset edilmesine kadar devam edecektir.

Segment 2 : I33.1' in 1 olmasıyla birlikte yeni bir çevrime hazırlık için Counter 1 sıfırlanır.

Segment 3 : I33.1' in "1" olmasıyla zamanlayıcı sıfırlanır.

Segment 4 : I33.1' in "0" veya F5.0' m "1" olma durumları, bir sonraki segmentlerde "çevrim devam edebilir" işareti olarak F5.1' de saklanır. Böylece bundan sonraki segmentler F5.0' m "0" olması durumlarında işleme sokulacaklardır.

Burada dikkat edilecek olursa I33.1' "1" olduktan sonra "0" a düşmeden çevrimin başlatılmadığıdır.

Bundan sonraki segmentlerdeki işlemler zamanlanan süre dolmuşsa (bu durum zamanlayıcının çıkışının saklandığı F5.2 değerine bakılarak anlaşılmaktadır. Zamanlayıcının çalışması durumunda bu değer "1" iken, zamanlama işlemi bittikten sonra sıfıra düşmektedir) ve çevrimin devam etmesi gerektiğini belirten F5.1 "0" ise gerçekleşmektedir. buna göre :

Segment 5 : sayıcının değeri çıkışa transfer edilmektedir. (sayıcının 16 bitlik formatta olmasından dolayı QW31' e transfer edilmiştir.)

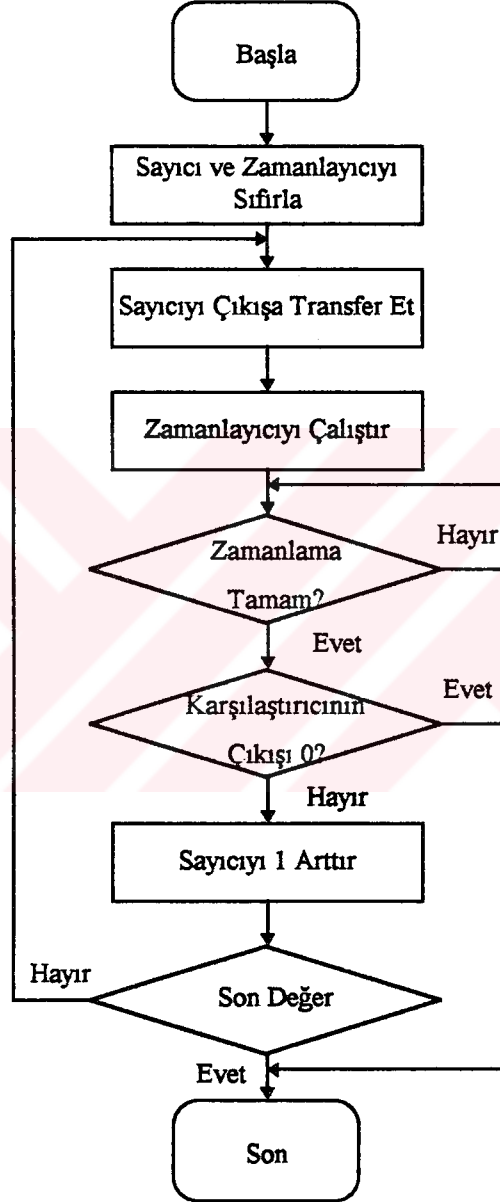
Segment 6 : Zamanlayıcı çalıştırılmaktadır.

Segment 7 : Karşılaştırıcının çıkışına bakılmakta "0" a düştüğü anda F5.0 reset edilerek çevrim kesilmektedir.

Segment 8 : Sayıcının değeri bir arttırılmaktadır.

Segment 9 : Sayıcının son değerinden sonra F5.0 reset edilerek çevrim bitirilmektedir.

Programın blok diyagramı Şekil 6.4' de gösterilmiştir.



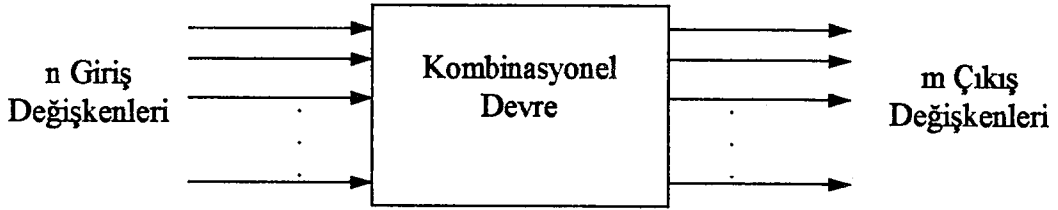
Şekil 6.4

NOT : Program iki fonksiyon bloğundan meydana gelmiştir. Bir önceki ADC uygulamasında olduğu gibi bütün işlemler sadece bir fonksiyon bloğu içerisinde

gerçekleştirilmiş olup, OB1 (Organizasyon bloğu) yalnızca FB1' e (Fonksiyon Bloğu) dallanılmak için kullanılmıştır.

6.2 PLC ile Kombinyonel Lojik Devre Tasarımı

Bu bölümde PLC kullanılarak kombinyonel lojik devrelerin tasarımı anlatılacaktır. Kombinyonel devrede belli bir andaki çıkışlar, önceki girişlere bakılmaksızın doğrudan doğruya o andaki girişlerin birleşimiyle belirlenen mantık kapılarından oluşur. Şekil 6.5' te böyle bir devrenin blok diyagramı verilmiştir.



Şekil 6.5 Kombinyonel Devre

Şekil 6.5' teki blok diyagramda görüldüğü gibi PLC kombinyonel devre olarak kullanılacaktır. Kombinyonel devre uygulamaları için de PLC olarak S5 90U kullanılacaktır.

Kombinyonel devrede n sayısı kadar girişi olan değişkenler için 2^n tane olası giriş değeri bileşimi vardır. Mümkün olan her giriş bileşimi için yalnız ve yalnız bir olası çıkış bileşimi vardır. Her biri bir çıkış değişkeni olmak üzere m tane Boole fonksiyonu tanımlanabilir. Buradaki her çıkış fonksiyonu n kadar giriş değişkeniyle ifade edilir.

Başlangıçta sözel olarak ifade edilen problem aşağıdaki yolun sırasıyla izlenmesi sonunda PLC'ye uygulanacak duruma gelir. Buna göre kombinyonel devrenin tasarlanması için aşağıdaki yol izlenir.

- Kaç giriş ve çıkışa ihtiyaç gösterildiği belirlenir.
- Giriş ve çıkış değişkenleri arasında istenilen ilişkiye göre doğruluk tablosu çıkartılır.
- Her bir çıkış için gerekiyorsa Karnaugh diyagramından da yararlanılarak fonksiyon sadeleştirilir.
- Mantık Şeması çizilerek PLC'ye uygulanabilir duruma getirilir.

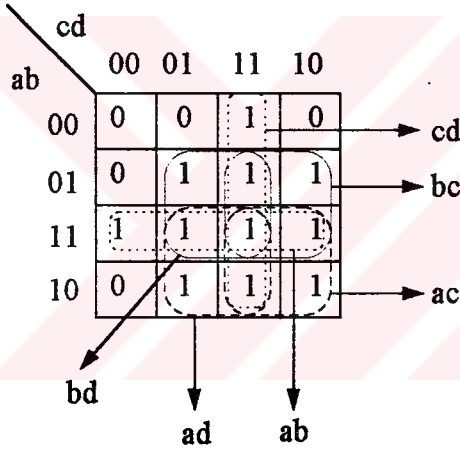
Şimdi bu sıraya uyarak kombinyonel devreye ilişkin problem ele alıp PLC'ye uygulayalım.

6.2.1 Problem

Dört giriş ve bir çıkışa sahip olan bir kontrol devresinde girişlerden en az ikisinin kapalı olması durumunda çıkışın "1" olması isteniyor. Bu durumda kombinyonel devreyi tasarlayarak PLC'ye uygulayalım.

a, b, c, d Giriş değişkenleri ve q çıkış değişkeni olarak seçilmek üzere, problemimize ait doğruluk tablosu, yandaki tablodaki gibi oluşturulur. Bu aşamadan sonra çıkış değişkenlerinin giriş değişkenlerinin birer fonksiyonu olarak yazılması gerekir. Buna göre doğruluk tablosundan minimum elemana sahip lojik fonksiyonu bulmak için Karnaugh diyagramından şu şekilde yararlanılır.

a	b	c	d	q
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1



Karnaugh diyagramından çıkış fonksiyonumuz

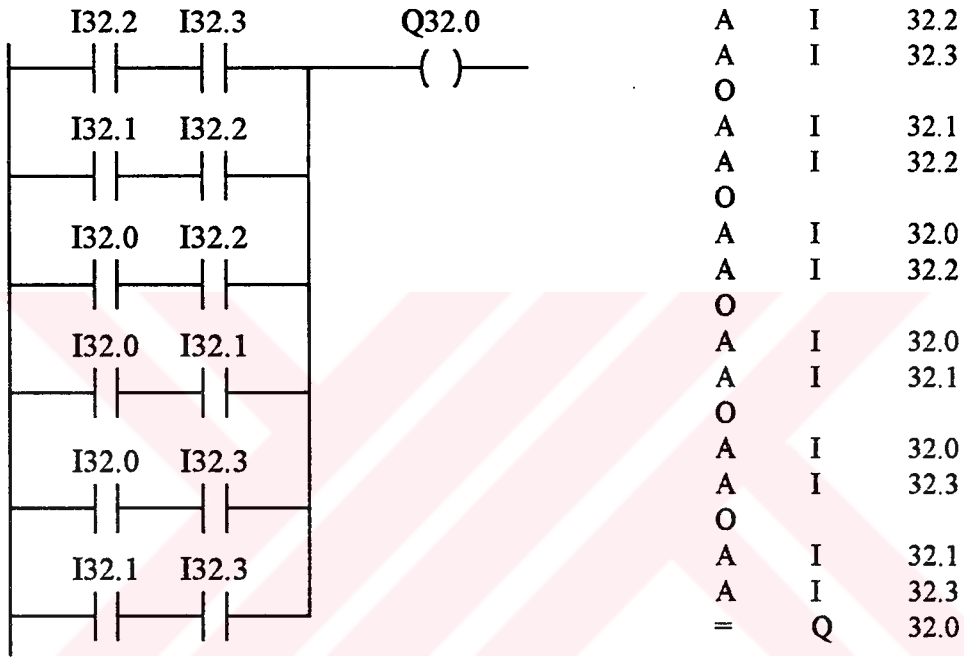
$$q = cd + bc + ac + ab + ad + bd$$

olarak elde edilir. Buradan 90U' nun 10 girişinden ilk 4 girişi olan I32.0, I32.1, I32.2, I32.3 sırasıyla a, b, c, d girişlerine, ve yine 90U' nun 6 çıkışından biri olan Q32.0 kombinyonel devremizin q çıkışına karşılık getirilerek lojik fonksiyonumuz şu şekilde yeniden düzenlenebilir:

$$Q32.0 = (I32.2).(I32.3) + (I32.1).(I32.2) + (I32.0).(I32.2) + (I32.0).(I32.1) + (I32.0).(I32.3) + (I32.1).(I32.3)$$

90U' nun giriş ve çıkışlarına göre tanımlanmış olan bu fonksiyon Step 5 programlama dilinde (LADDER ve STL ile) Şekil 6.6' daki gibi oluşturulabilir.

LADDER ve STL ile oluşturulmuş olan bu kombinyonel devrede I32.0, I32.1, I32.2,

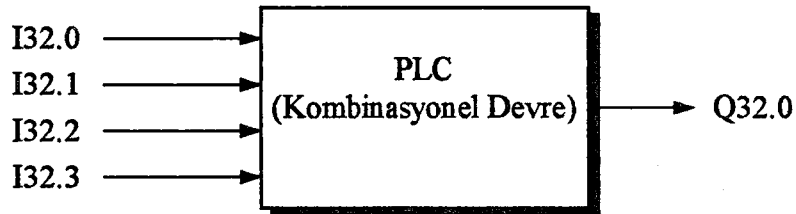


(a) LADDER

(b) STL

Şekil 6.6

I32.3 girişlerinden herhangi ikisinin "1" olması durumunda Q32.0 çıkışı "1" konumuna geçecektir. Son haliyle kombinyonel devre Şekil 6.7' deki gibidir.



Şekil 6.7 Kombinyonel Devrenin PLC ile Oluşturulması

VII. TARTIŞMA VE SONUÇ

Endüstriyel otomasyon alanında yaygın olarak kullanılan PLC' lerde giriş/çıkış birimleri PLC CPU' sunun dış dünya ile bağlantısını sağlamakta kullanılırlar. PLC' ler için ilk yıllarda sadece ayrık (discrete) açık/kapalı elemanlar için arabirimler bulunmaktaydı ki; bu tür arabirimler kontrol işlemleri için oldukça sınırlıydı.

En yaygın olarak kullanılan arabirim çeşidi olan ayrık giriş/çıkış arabirimi, CPU' nun "1" veya "0" türündeki çıkışlara sahip giriş elemanları ile veya "1" veya "0" türündeki çıkışlarla kontrol edilebilecek çıkış elemanları ile bağlantısını sağlar. Bu giriş elemanlarına örnek olarak puşbuton, fotoelektrik sensör ve seviye anahtarlarını gösterebiliriz. Aynı şekilde ayrık giriş/çıkış arabirimleri ile denetlenebilecek çıkış elemanlarına kontrol rölelerini, fanları veya alarmları örnek olarak verebiliriz.

PLC' lerin zamanla gelişip daha çok iş yapabilecek yeteneklere (çeşitli aritmetik işlemler, sayma ve zamanlama şeklinde özellikler kazanmaları gibi) kavuşmalarıyla birlikte, ayrık türdeki giriş/çıkış arabirimlerinin yanısıra, farklı türlerdeki veri giriş/çıkışlarını da sağlayabilecek arabirimlerin yapılması gerekliliği ortaya çıktı. Bu şekilde farklı verileri de kabul edebilen giriş/çıkış arabirimlerinin yapılmasıyla, PLC' lerin çok daha değişik kontrol uygulamalarında kullanılmaları da sağlanmış oldu. Sözgelimi PLC' lerde analog giriş çıkış birimlerinin kullanılmalarıyla birlikte, analog kontrol işlemlerinde kullanılan birçok elemanın PLC' lerle doğrudan bağlantısı sağlanmış oldu.

Tezin ilk kısmında, kontrol elemanı olarak PLC kullanan ve dijital/analog çevirici içeren bir ADC gerçekleştirilmiştir. Böylece, bu uygulamada kullanılan ve sadece ayrık giriş/çıkışlara sahip olan S590U PLC' sinde analog giriş kabul edebilecek duruma getirilmesi sağlanmıştır. Ancak burada kullanılan PLC' nin altı çıkışa sahip olması dolayısıyla gerçekleştirilen ADC' nin çözünürlüğü de 2^6 da sınırlı kalmıştır ve bunun dışında, kullanılan PLC' nin gerek çevrim hızının düşük olması ve gerekse çıkış ve girişlerdeki 0→1 ve 1→0 geçişlerinin oldukça uzun zaman alması ADC' nin çevrim hızını da oldukça yavaşlatmıştır.

PLC ile gerçekleştirilen ADC' de tek bir devre kurularak, sadece PLC yazılımının değiştirilmesiyle devrenin farklı iki türden (ardışıl yaklaşımlı ve sayıcı) ADC olarak çalışması sağlanmıştır. Bu şekilde kurulan farklı iki türden ADC' nin çalışmalarının, kullanılan

programdaki zamanlayıcının değeri değiştirilerek (özellikle ADC' nin çevrim hızını azaltarak), kolaylıkla incelenebilmelerine olanak sağlanmıştır.

Ardışıl yaklaşımlı ADC' de girişteki analog işaret altı bitlik bir çevrim sonunda dijital işarete çevrilirken, sayıcı türden ADC' de ise ancak 2^6 bitlik bir çevrimden sonra analog işaretimiz dijitale çevrilebilmektedir.

EKLER kısmında verilen programlarda LADDER ile yazılmış programlar, daha sonra STL' ye convert edilmiştir ve ayrıca program, sadece STL ile de yazılarak bu programın LADDER' a göre daha esnek ve kısa olduğu gözlenmiştir.

Tezde bunun dışında PLC kullanılarak kombinasyonel devre tasarımı nasıl gerçekleştirildiği de anlatılmıştır. Böyle bir tasarımda sözlü olarak ifade edilen problemde kurulacak kontrol sistemine göre, gerekli giriş/çıkış sayıları belirlendikten sonra, kombinasyonel lojik devre tasarım metodu kullanılarak problemin PLC' ye nasıl uygulanması gerektiği gösterilmiştir.

Böylece, bir PLC ile gerçekleştirilmek istenen kontrol için yazılacak PLC programının, kombinasyonel lojik devre tasarım yöntemi kullanılarak minimum seviyede tutulması sağlanabilir. Bu şekilde kullanılan programın tarama zamanı düşürülebileceği gibi, PLC' nin program için ayrılan hafızasında en verimli şekilde kullanılması da sağlanmış olur.

VIII. ÖZET

Tezin giriş bölümünde genel olarak PLC'lerin hangi ihtiyaçlardan dolayı ortaya çıktıklarından bahsedilmiş ve zaman içindeki tarihsel değişimlerine değinilmiştir. Mikroişlemci alanındaki gelişmelerle birlikte, PLC'lerin kazandıkları yetenekler ve dolayısıyla klasik röleli kontrol panellerine göre elde ettikleri üstünlükler, yine bu bölümün konuları arasında yer almıştır.

Tezin uygulamalara kadar olan bölümlerinde ise PLC'lerin yapıları ve programlanmaları ele alınmıştır. PLC'lerin yapılarının anlatıldığı bölümde, genel olarak bir PLC'nin blok diyagramı çizilerek, PLC'yi oluşturan modüller tek tek ele alınarak incelenmiş ve PLC'lerin analog giriş/çıkış modüllerine de değinilmiştir. PLC'lerde programlanmanın anlatıldığı bölümlerde, programlama mantığının nasıl olması gerektiğine değinildikten sonra, tezdaki uygulamaların geliştirildiği S5 90U PLC'sinin programlama dili olması dolayısıyla, STEP 5 programlama dili, tezin bir başka bölümü olarak ele alınmış ve ayrıntılı bir şekilde incelenmiştir. Bu bölümde, özellikle metin tabanlı programlama yapılmasına olanak sağlayan STL ile grafik tabanlı programlama yapılmasına olanak sağlayan CSF ve LADDER programlama metotları ayrıntılı bir şekilde ele alınarak, programlamada her bir metodun diğerine göre avantajlarının ortaya çıkarılmasına çalışılmıştır.

Uygulamalardan önceki son kısımda, uygulamalarda büyük yer tutması dolayısıyla Analog/Dijital çeviriciler ele alınmış ve bunun yanı sıra analog ve dijital işaret kavramları da açıklanarak çeviricilerin daha iyi kavratılması amaçlanmıştır.

Tezin uygulamalar bölümünde "PLC ile ADC" ve "PLC ile Kombinasyonel Devre" olmak üzere, iki farklı uygulama ele alınmıştır. PLC ile ADC uygulamasında, on giriş ve altı çıkışa sahip S5 90U PLC'si kullanılmış ve kurulan bir devreye karşın, sadece PLC yazılımının değiştirilmesiyle, devrenin "ardışıl yaklaşım" ve "sayıcı türden" olmak üzere iki farklı türde ADC olacak şekilde çalışması gözlenmiştir. Yazılımlar metin ve grafik tabanlı programların birbirlerine karşı avantajlarının daha iyi anlaşılması açısından STL ve LADDER ile ayrı ayrı yazılmıştır. Diğer uygulamada ise kombinasyonel devre tasarım metodunun PLC'ye uygulanması ele alınarak, PLC ile tasarlanacak bir kontrol probleminin, verilen bir örnekle PLC'ye nasıl uygulandığı gösterilmiştir.

SUMMARY

AUTOMATIC CONTROL OF ELECTRONIC CIRCUITS WITH PLC

At the beginning, the requirements that caused PLCs to be used widely are considered and historical developments are explained. With the developments in the field of microprocessor technology, the capabilities of PLCs and because of these, the advantages of them to the classical relay control panels are mentioned.

Before the chapters of applications, the structures of the PLCs and how they are programmed are described. In the chapter that the structures of the PLCs are described, the block diagram of a PLC is drawn and the modules that construct PLC are examined one by one and analog input/output modules are explained. In the sections about the PLC programming, the algorithms of programming are presented in general and STEP 5 - which is the programming language of S5 90U PLC - is examined in detail. In this chapter, text oriented programming structure STL and graphical oriented programming structure CSF and LADDER methods are studied and their advantages and disadvantages are determined.

In the last section before the applications chapter, because of their importance, analog and digital signal concepts are explained and the analog to digital converters are described.

In the applications chapter, two different applications namely "ADC with PLC" and "Combinational circuits with PLC" are mentioned. In the "ADC with PLC", S5 90U PLC having 10 inputs and 6 outputs is used. Hence, by changing only the program list of PLC the operation of the circuit as a "Successive Approximation" and as a "Counter" are observed. The software are written in both STL and LADDER to show the advantages and disadvantages of them to each other. In the other application, the method of designing combinational circuit is covered with an example.

IX. KAYNAKLAR

- [1] SİMKO SANAYİ VE TİCARET A.Ş. (1993) : Simatic S5 Seminer Notları
- [2] SIEMENS (1991) : Simatic S5 Step[®]5 Ladder 90 User's Guide
- [3] SIEMENS (1991) : Simatic S5 Step[®]5 AG 90 Package Manual
- [4] Siemens Simatic S5 S5-90U, S5-95U, S5-100U and S5-95F Programmable Controllers Catalog (1994)
- [5] WEBB J. W. : Programmable Logic Controllers Principles and Applications (ISBN 0-02-946411-0)
- [6] IAN G. W. : Programmable Controllers Operation and Application (ISBN 0-13-730037-9)
- [7] Prof.Dr. TÜTÜNCÜOĞLU E. (1987) : Mikroişlemciler ve Bilgisayarlar (Uludağ Üniversitesi Mühendislik Fakültesi)
- [8] Doç.Dr. KAYNAK O. (1988) : Programlanabilir Denetleyiciler (Boğaziçi Üniversitesi Mühendislik Fakültesi)
- [9] JOHNSON D.G. (1987) : Programmable Controllers for Factory Automation (ISBN 0-8247-7674-7)
- [10] Doç.Dr. KURTULAN S. (1996) : Programlanabilir Lojik Kontrolörler ve Uygulamaları
- [11] OTTER J. D. (1994) : Programlanabilir Mantık Denetleyicileri (M.E.B. ISBN 975-11-0874-8)
- [12] HALL D. V. (1994) : Mikroişlemciler ve Sayısal Sistemler (M.E.B. ISBN 975-11-0858-6)
- [13] MANO M. M. (1994) : Sayısal Tasarım (M.E.B. ISBN 975-11-0875-6)

X EKLER

EK-1 : Ardiřıl yaklařımlı ADC iin yazılan LADDER program (61-65)

EK-2 : Sayıcı trden ADC iin yazılan LADDER program. (66-68)

EK-3 : Ardiřıl yaklařımlı ADC iin yazılan LADDER programın STL' ye convert edilmiř durumu (69-72)

EK-4 : Sayıcı trden ADC iin yazılan LADDER programın STL' ye convert edilmiř durumu (73-74)

EK-5 : Ardiřıl yaklařımlı ADC iin doęrudan STL ile yazılan program (75-76)

EK-6 : Sayıcı trden ADC iin doęrudan STL ile yazılan program (77-78)



EK-1

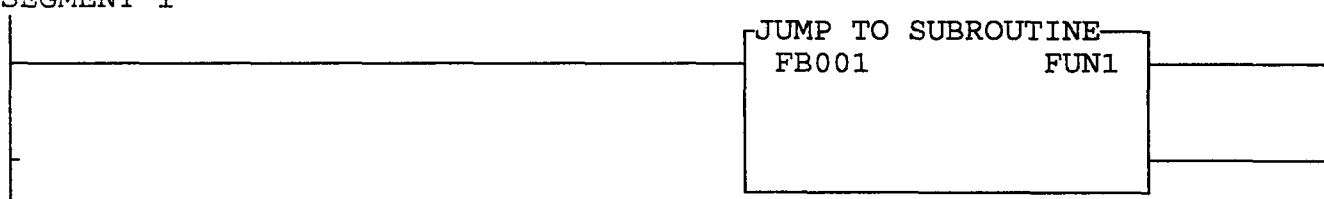
ARDISIL YAKLASIMLI ADC (LADDER)

LADDER 90	SIEMENS AG
Copyright 1991	All Rights Reserved
Date: May 14, 1996	Time: 03:15:52 PM

SEGMENT LOGIC

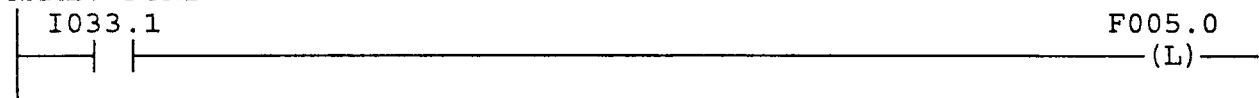
OB001

SEGMENT 1



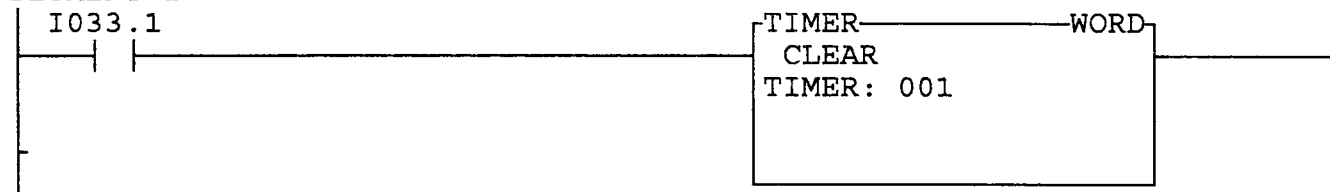
FB001

SEGMENT 1
NAME: FUN1



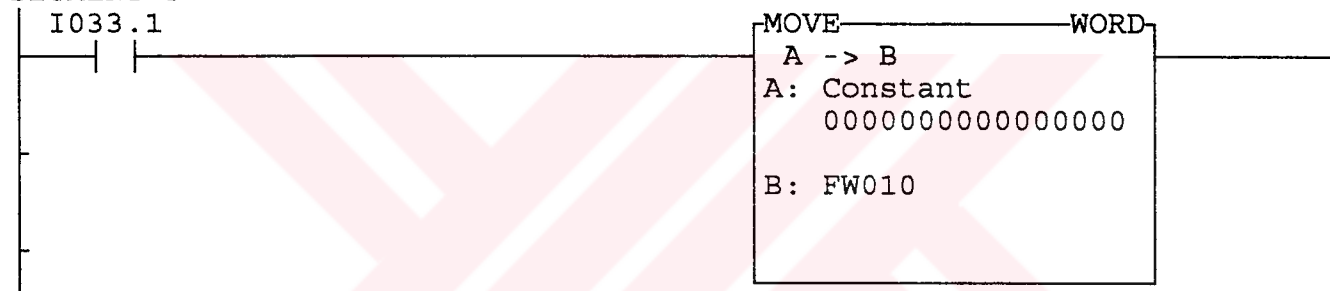
FB001

SEGMENT 2



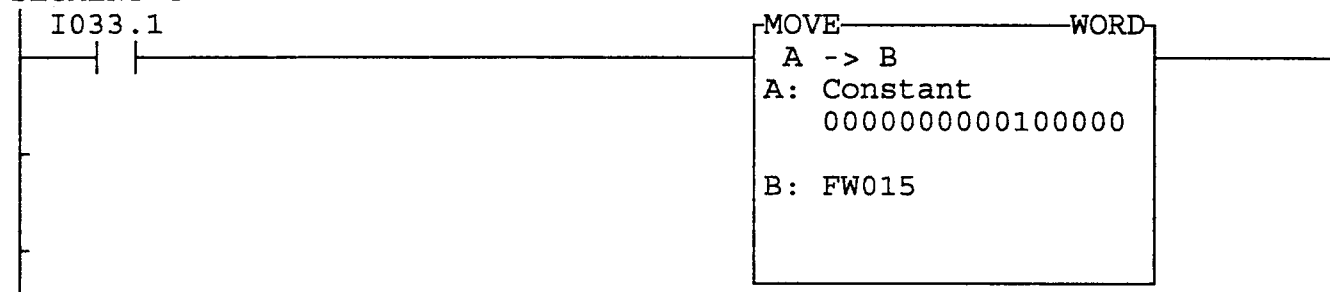
FB001

SEGMENT 3



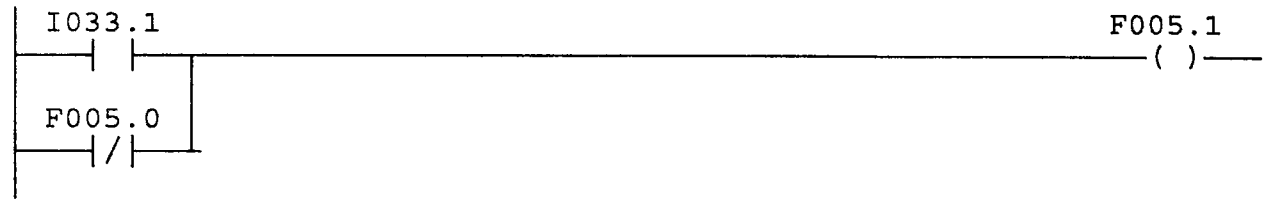
FB001

SEGMENT 4



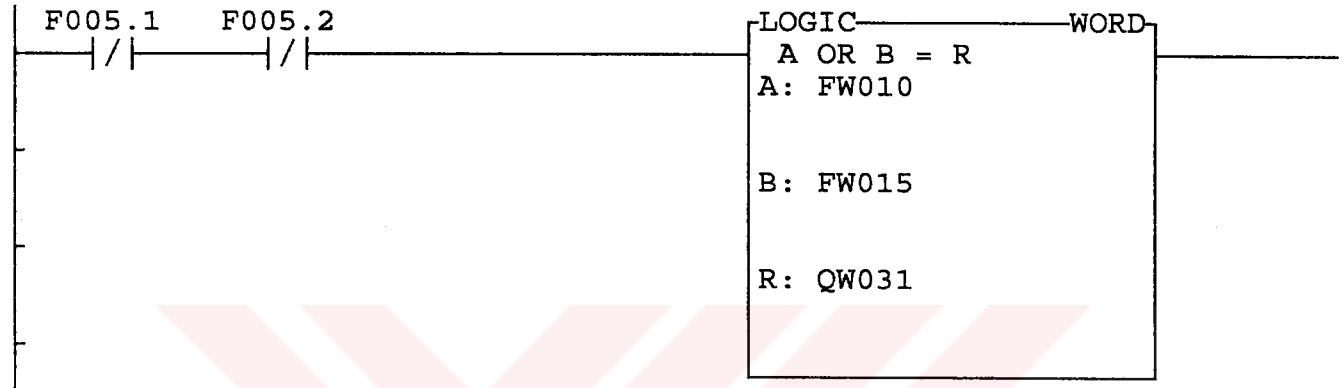
FB001

SEGMENT 5



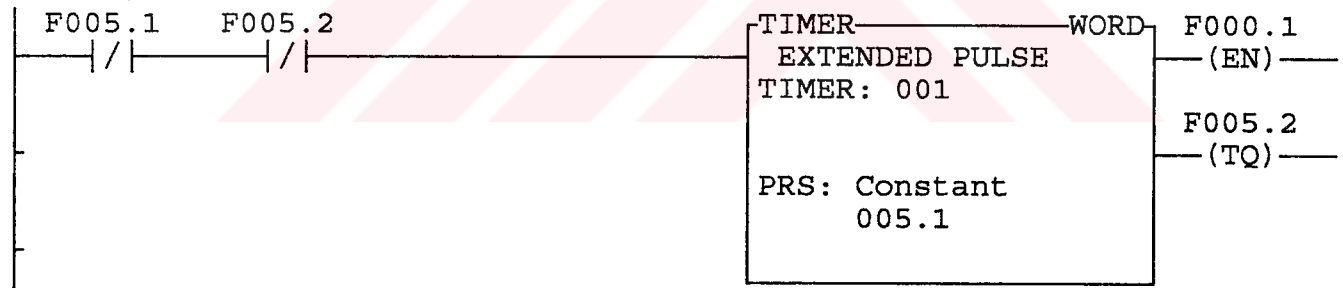
FB001

SEGMENT 6



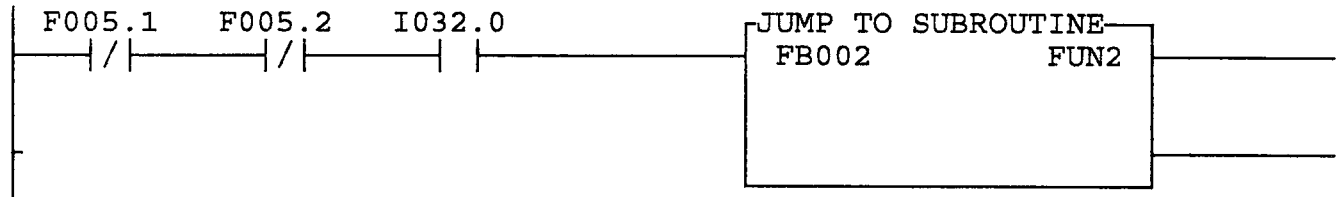
FB001

SEGMENT 7



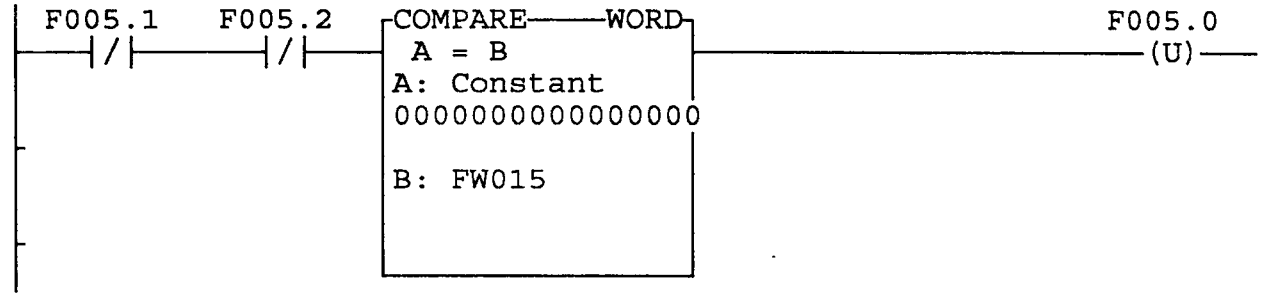
FB001

SEGMENT 8



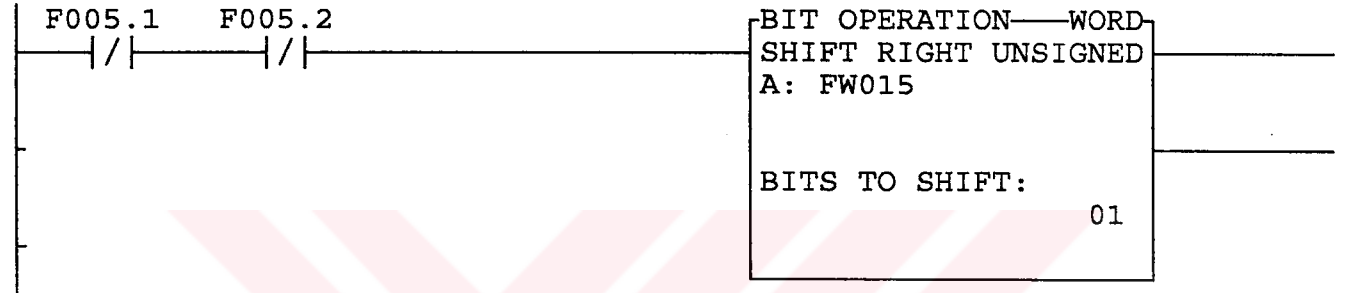
FB001

SEGMENT 9



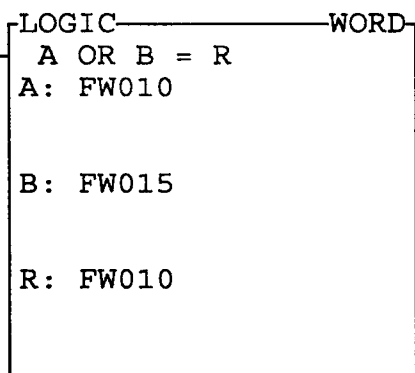
FB001

SEGMENT 10



FB002

SEGMENT 1
NAME: FUN2



EK-2

SAYICI TURDEN ADC (LADDER)

LADDER 90

SIEMENS AG

Copyright 1991

All Rights Reserved

Date: May 14, 1996

Time: 03:22:45 PM

SEGMENT LOGIC

OB001

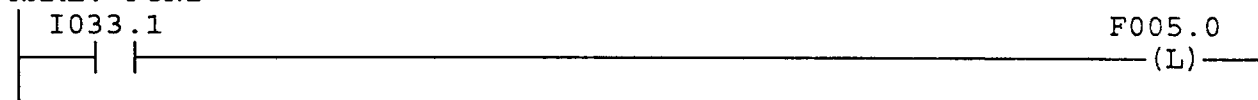
SEGMENT 1

JUMP TO SUBROUTINE
FB001 FUN1



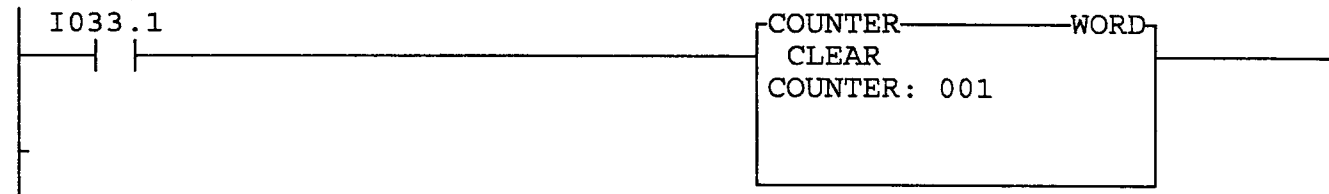
FB001

SEGMENT 1
NAME: FUN1



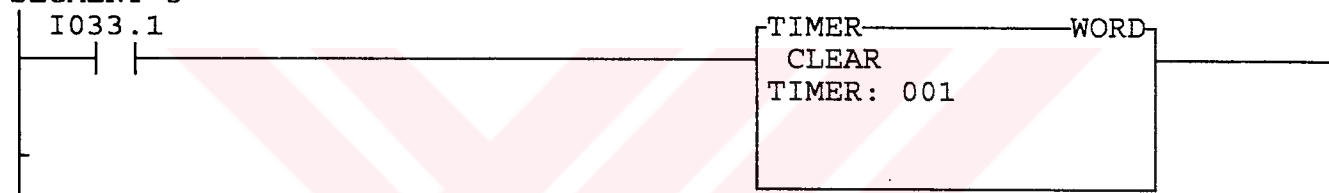
FB001

SEGMENT 2



FB001

SEGMENT 3



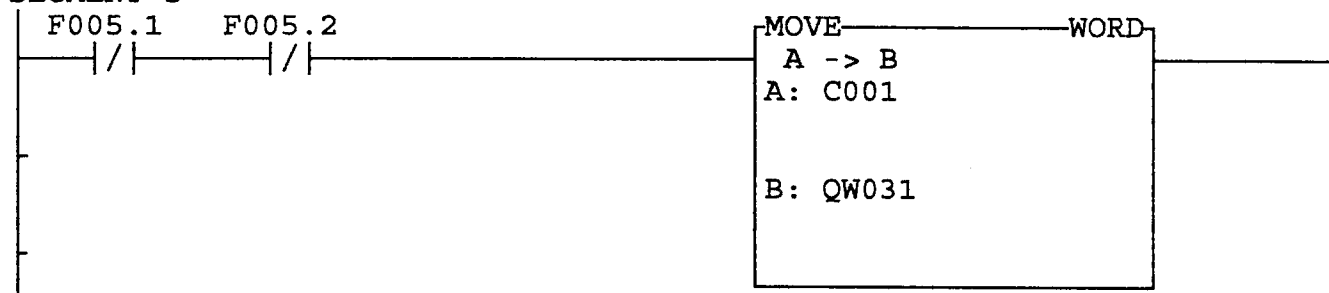
FB001

SEGMENT 4



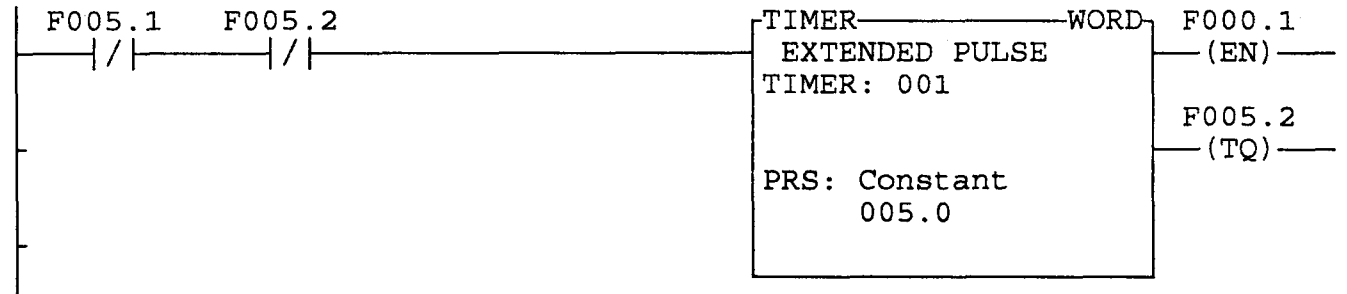
FB001

SEGMENT 5



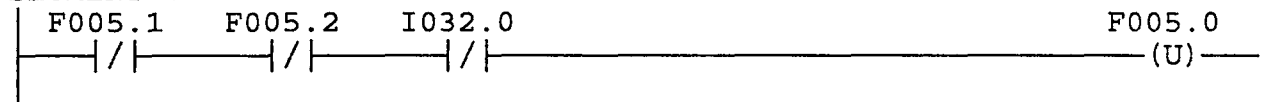
FB001

SEGMENT 6



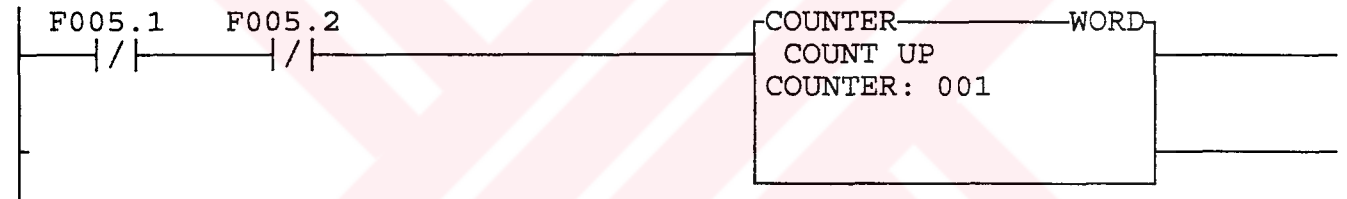
FB001

SEGMENT 7



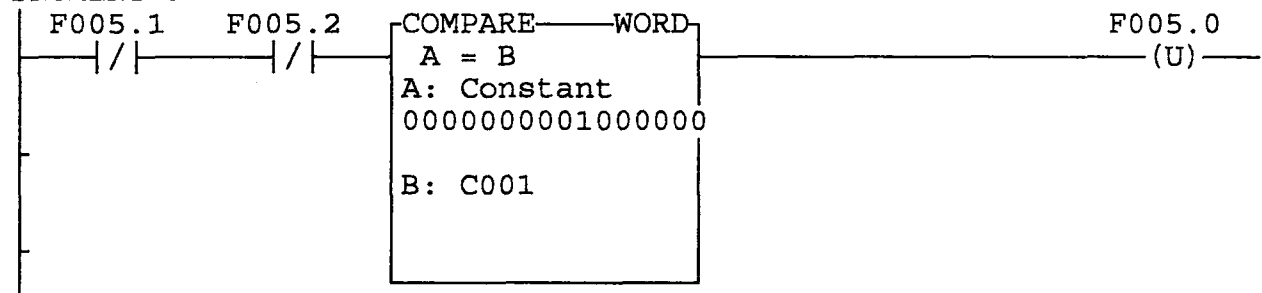
FB001

SEGMENT 8



FB001

SEGMENT 9



ARDISIL YAKLASIMLI ADC (CONVERT TO STL)

LADDER 90

SIEMENS AG

Copyright 1991

All Rights Reserved

Date: May 14, 1996

Time: 10:17:35 PM

SEGMENT LOGIC

OB001

SEGMENT 1

: JU FB1

NAME: FUN1

: BE



FB001

SEGMENT 1

NAME: FUN1

: A I33.1

: S F5.0

SEGMENT 2

: A I33.1

: R T1

SEGMENT 3

: A I33.1

: JC =M1

: JC =M2

M1 : L KM 000000000000000000

: T FW10

M2 : NOP 0

SEGMENT 4

: A I33.1

: JC =M3

: JC =M4

M3 : L KM 00000000000100000

: T FW15

M4 : NOP 0

SEGMENT 5

: O I33.1

: ON F5.0

: = F5.1

SEGMENT 6

: AN F5.1

: AN F5.2

: JC =M5

: JC =M6

M5 : L FW10

: L FW15

: OW

: T QW31

M6 : NOP 0

SEGMENT 7

: AN F5.1

: AN F5.2

: = F0.1

: L KT 5.1

: SE T1

: A T1

: = F5.2

SEGMENT 8

: AN F5.1

: AN F5.2

: A I32.0

: JC FB2

NAME: FUN2

:

SEGMENT 9

: AN F5.1

: AN F5.2

: A(

```
      : JC  =M7
      : JU  =M8
M7    : L   KM 000000000000000000
      : L   FW15
      : !=F
M8    : )
      : R   F5.0
SEGMENT 10
      : AN  F5.1
      : AN  F5.2
      : JC  =M9
      : JC  =M10
M9    : L   FW15
      : SRW 1
      : T   FW15
M10   : NOP 0
      : BE
```



FB002

SEGMENT 1

NAME: FUN2

: L FW10
: L FW15
: OW
: T FW10
: BE



SAYICI TURDEN ADC (CONVERT TO STL)

LADDER 90

SIEMENS AG

Copyright 1991

All Rights Reserved

Date: May 14, 1996

Time: 10:23:47 PM

SEGMENT LOGIC

OB001

SEGMENT 1

: JU FB1

NAME: FUN1

: BE



FB001

```

SEGMENT 1
NAME: FUN1
  : A   I33.1
  : S   F5.0
SEGMENT 2
  : A   I33.1
  : R   C1
SEGMENT 3
  : A   I33.1
  : R   T1
SEGMENT 4
  : O   I33.1
  : ON  F5.0
  : =   F5.1
SEGMENT 5
  : AN  F5.1
  : AN  F5.2
  : JC  =M1
  : JC  =M2
M1     : L   C1
       : T   QW31
M2     : NOP 0
SEGMENT 6
  : AN  F5.1
  : AN  F5.2
  : =   F0.1
  : L   KT 5.0
  : SE  T1
  : A   T1
  : =   F5.2
SEGMENT 7
  : AN  F5.1
  : AN  F5.2
  : AN  I32.0
  : R   F5.0
SEGMENT 8
  : AN  F5.1
  : AN  F5.2
  : CU  C1
SEGMENT 9
  : AN  F5.1
  : AN  F5.2
  : A(
  : JC  =M3
  : JU  =M4
M3     : L   KM 0000000001000000
       : L   C1
       : !=F
M4     : )
       : R   F5.0
       : BE

```


EK-5

ARDISIL YAKLASIMLI ADC (STL)

LADDER 90

SIEMENS AG

Copyright 1991

All Rights Reserved

Date: May 13, 1996

Time: 03:19:21 PM

SEGMENT LOGIC

OB001

SEGMENT 1

: JU FB1

NAME: FUN1

:

: BE



FB001

SEGMENT 1

NAME: FUN1

```

: A   I33.1
: S   F5.0
: A   I33.1
: ON  F5.0
: JC  =M1
: A   F7.0
: JC  =M2
:
: R   F7.0
: L   KM 000000000000000000
: T   FY0
: L   KM 00000000000100000
: T   FY1
:
M6   : L   FY1
      : L   FY0
      : OW
      : T   QB32
:
M2   : AN  F7.0
      : L   KT 5.1
      : SE  T1
      : A   T1
      : =   F7.0
      :
      : A   F7.0
      : JC  =M3
      :
      : AN  I32.0
      : JC  =M4
      :
      : L   FY1
      : L   FY0
      : OW
      : T   FY0
:
M4   : L   KM 000000000000000000
      : L   FY1
      : !=F
      : JC  =M5
      :
      : L   FY1
      : SRW 1
      : T   FY1
      : JU  =M6
:
M5   : R   F5.0
M1   : R   F7.0
:
M3   : BE

```

EK-6

SAYICI TURDEN ADC (STL)

LADDER 90

SIEMENS AG

Copyright 1991

All Rights Reserved

Date: May 13, 1996

Time: 03:25:48 PM

SEGMENT LOGIC

OB001

SEGMENT 1
: JU FB1
NAME: FUN1
:
: BE



FB001

SEGMENT 1

NAME: FUN1

```

      : A   I33.1
      : S   F5.0
      : A   I33.1
      : ON  F5.0
      : JC  =M1
      : A   F7.0
      : JC  =M2
      :
      : R   C1
      :
M5    : L   C1
      : T   QB32
      :
M2    : AN  F7.0
      : L   KT 2.2
      : SE  T1
      : A   T1
      : =   F7.0
      :
      : AN  F7.0
      : CU  C1
      :
      : A   F7.0
      : JC  =M3
      :
      : AN  I32.0
      : JC  =M4
      :
      : L   KM 00000000001000000
      : L   C1
      : !=F
      : JC  =M4
      :
      : JU  =M5
      :
M4    : R   F5.0
M1    : R   F7.0
      :
M3    : BE

```

XI. ÖZGEÇMİŞ

22.09.1971 Tarihinde Batman'da doğdu. 1989 yılında Pendik Lisesi'nden mezun oldu. Aynı yıl girdiği üniversite sınavında İstanbul Üniversitesi Elektronik Mühendisliği Bölümü'ne girmeye hak kazandı. Bu bölümden mezun olduğu 1993 yılında, İstanbul Üniversitesi Fen Bilimleri Enstitüsü Elektronik Mühendisliği Anabilim Dalı'nda Yüksek Lisans programına kayıt yaptırdı. Halen aynı üniversitenin Elektrik-Elektronik Mühendisliği Bölümü Elektrik Makinaları Anabilim Dalı'nda Araştırma Görevlisi olarak görev yapmaktadır.

