

R
152
71

YILDIZ TEKNİK ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

Röleli Loj. Dev. İlg. Mik. İle Gerç

Yüksek Lisans Tezi

Seyfi Sarısel

1986

Ref
EHM
229
1986

152
71

Elk Hab
1500
3500

YILDIZ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

RÖLELİ LOJİK DEVRE İŞLEVLERİNİN
MİKROBİLGİSAYAR İLE GERÇEKLENMESİ

Yüksek Lisans Tezi
Müh. Seyfi SARİSEL

KONYA 1986

YILDIZ ÜNİVERSİTESİ
GENEL KİTAPLIĞI

Kot : R 152
Alındığı Yer : Fen Bil. Ens. 71
Tarih : 9.12.1988
Fatura :
Fiatı : 3500 TL.
Ayniyat No : 1/21
Kayıt No : 45765
UDC : 001.64
Ek : 373.242



27-
YILDIZ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

RÖLELİ LOJİK DEVRE İŞLEVLERİNİN
MİKROBİLGİSAYAR İLE GERÇEKLENMESİ

Yüksek Lisans Tezi

Müh. Seyfi SARISEL

ÖZET

Konumuz röleli lojik devre işlevlerinin mikrobilgisayar ile gerçekleşmesidir. Bu nedenle birinci bölümde bu konu üzerinde durulmuştur. İkinci bölümde, ileriki konularda anlatılacak olan çalışmalarımızın daha iyi anlaşılabilmesi için röleli lojik devrelerde kullanılan elemanlar anlatılmıştır. Bu elemanların sembolleri, yapıları ve çalışma prensipleri verilmiştir. Üçüncü bölümde, bu elemanlardan oluşan ve konumuza temel oluşturabilecek bazı röleli lojik devreler incelenmiştir. Bu devrelerin lojik ifadelerinin çıkarılması verilmiştir. Üçüncü bölümün sonunda rölelerle yapılmış kompleks bir devre olan toplamalı bir asansörün devre şemaları ve çalışma prensipleri anlatılmıştır. Dördüncü bölümde röleli lojik devrelerin mikrobilgisayara uygulanabilmesi için yazılan program verilmiştir. Bu programın hangi temeller üzerine dayandığı, bölümlerinin amaç ve çalışmaları, giriş, çıkış birimlerinin ne olacağı anlatılmıştır. Konunun sonunda verilen bir örnekle programın kullanılma ve çalışma yönünden daha iyi anlaşılması düşünülmüştür. Beşinci bölümde daha karmaşık röleli lojik devrelerden oluşan asansörler için yazılan basic programlar verilmiştir. Bu programların yazılmasındaki düşünce tarzı, amacı ve uygulanması konuları işlenmiştir.

Complicated relay logic circuits had been given in the fifth part. Thought style, intent and application items of these programs had been explained.

SUMMARY

Our them is realizing the relay logic circuit functions with a microcomputer. Because of this reason this them is dwelled on in the first part. In the second part, the elements which are used in the relay logic circuits had been explained to make the works which would be explained in the ahead them more understandable. The symbols, structures and the working principles had been given. In the third part, some relay logic circuits which can create a foundation to our them and comes into existence from these elements have been examined minutely. The logic expressions of these circuits had been given. At the ending of the third part the circuit diagram and the working principle of an elevator which is a complex circuit made of relays had been explained. In the fourth part some programs had been given to apply the relay logic circuits to a microcomputer. It had been explained which foundations this program bases on, intends and working of the parts and what the input and the output units will be and how to use them. The program is thought to be understood better in respect of using and working with the example given at the ending of the them. Some basic programs written for the elevators which comes into existence from more complicated relay logic circuits had been given in the fifth part. Thought style, intend and application them of these programs had been explained.

3.4 Devre Yolu Belirleme Programı	12
3.5 His Motors Yoldas Uygulama His Motors Programı	13
3.6 Toggala Kumandali His Motoru Uygulama	14
3.6.1 Giriş	14
3.6.2 Devrede Kullanılan Elemanlar	14
3.6.3 Devrenin Çalışması	16
4. BULUNAN LOGİK DEVRELERİN ARIKORU KULLANILAN UYGULAMA	
4.1 Giriş	19

İ Ç İ N D E K İ L E R

1. GİRİŞ	1
2. RÖLELİ DEVRELERDE KULLANILAN ELEMAN VE SEMBOL LERİ	2
2.1 Giriş	2
2.2 Butonlar	2
2.3 Mikro Anahtarlar	2
2.4 Röleler	3
2.5 Kontaktörler	4
2.6 Aşırı Akım Röleleri	5
2.6.1 Manyetik Aşırı Akım Röleleri	5
2.6.2 Termik Aşırı Akım Röleleri	6
2.7 Manyetik Açma Röleleri	6
2.8 Zaman Röleleri	7
2.8.1 Düz Zaman Röleleri	7
2.8.2 Ters Zaman Röleleri	7
2.8.3 Pistonlu Zaman Röleleri	8
2.8.4 Motorlu Zaman Röleleri	8
2.8.5 Termik Zaman Röleleri	8
2.8.6 Doğru akım Zaman Röleleri	9
3. RÖLELİ LOJİK DEVRELERİN İNCELENMESİ	10
3.1 Giriş	10
3.2 Ani Temaslı Butonlarla Yapılan kumanda devresi	10
3.3 Röleli Uzaktan Kumanda Devresi	11
3.4 Dönüş Yönü Değiştirme Devresi	12
3.5 Bir Motora Yıldız-Üçgen Yol Verme Devresi	13
3.6 Toplama Kumandalı Bir Hızlı Asansör	14
3.6.1 Giriş	14
3.6.2 Devrede Kullanılan Elemanlar	14
3.6.3 Devrenin Çalışması	16
4. RÖLELİ LOJİK DEVRELERİN MİKROBİLGİSAYARA UYGULANMASI	19
4.1 Giriş	19

4.2	Programın Amacı	20
4.3	Programın İncelenmesi	21
4.3.1	Programın Blok Şeması	21
4.3.2	Programdaki Değişken Ve Dizilerin Anlamları	21
4.3.3	Lojik İfadelerin Programa Aktarılmasında Kullanılan İfadeler	22
4.3.4	Programın Bölümlerinin İncelenmesi	22
4.3.4.1	Lojik İfadelerin Bilgisayara girilmesi	22
4.3.4.2	Yazılan satırların görülmesi bölümü	23
4.3.4.3	Edit İşlemi	23
4.3.4.4	Save ve Load İşlemleri	24
4.3.4.5	Compile Etme Bölümü	24
4.3.4.6	Tuş Tanımlama Bölümü	27
4.3.4.7	Run Etme Bölümü	27
4.3.5	Assambler Alt Programın İncelenmesi	28
4.4	Programın Kullanılması	29
5.	İKİ YÖNDE TOPLAMALI BİR ASANSÖRÜN BASIC PROGRAMI	38
5.1	Giriş	38
5.2	Programın İncelenmesi	38
5.2.1	Programın Akış Şeması	38
5.2.2	Programdaki Değişken ve Dizilerin anlamları	38
5.2.3	Programın Bölümlerinin İncelenmesi	40
5.2.3.1	Girişlerin Taranması	40
5.2.3.2	Çıkışların Görülmesi	40
5.2.3.3	Asansörün Hareket Yönünün Tayini	40
5.2.3.4	Asansörün Durdurulması	41
6.	DUBLEKS ASANSÖRÜN BASIC PROGRAMI	46
6.1	Giriş	46
6.2	Programın İncelenmesi	46

6.2.1	Programın Akış Şeması	46
6.2.2	Programdaki Değişken ve dizilerin Anlamları	46
6.2.3	Girişlerin Taranması Bölümü	47
6.2.4	Kabinlerin Hareket Yönlerinin Tayini ve Programın Tamamı	49
7.	KAYNAKLAR	57
8.	ÖZGEÇMİŞ	58

1.GİRİŞ

Elektronik ve mikrokompüter teknolojisi, bugün büyük uygulamalarıyla endüstriye girmiştir.Tesislerin emniyeti, güvenilirliği, üretim hızı, mamüllerin kalitesi otomasyon ile sağlanmaktadır.Tesislerde geniş amaçlı olarak röle ve kontaktör gibi elektromekanik devre elemanları ile yapılan proses kontrolleri, teknolojinin gelişmesi ile artık yetersiz kalmakta, bu gibi elemanların sayılarının artmasında tesis kontrol edilemeyecek şekilde karmaşık donanımlar oluşturmaktadır.Öte yandan bu konvansiyonel teknikle yapılan kontrol sistemleri, montajı ve konstrüksiyon masrafları orta büyüklükteki tesislerde bile aşırı boyutlara ulaşmaktadır.Bu konuda yapılan araştırmalar, uygulayıcıları programlanabilir otomasyon cihazlarının yaygın biçimde bu alanda da kullanmaya yöneltmiştir.Böylece kontrol ile ilgili kriter ve komutların, bu amaçla yapılan elektrik tesisatından bağımsız olması sağlanmıştır.

Programlanabilir otomasyon cihazları, endüstriye büyük bir yenilik getirmekte, konvansiyonel röle ve kontaktör tekniği ile yapılan proses kontroluna göre kesin üstünlükleri nedeni ile büyük avantaj sağlamaktadır.

Bu tip tesisat ve kontrol yöntemleri, örneğin gıda,şeker, petrokimya,çimento fabrikaları, demir-çelik, maden endüstrisi, meşrubat sanayii, tekstil ile bütün takım tezgahları,otomatik ambalajlama, diğer seri üretim yapan tesisler ve otomatik çalışan bantlarda büyük önem taşırlar. Bu tip sistemlerde emniyetli kontrol çok önemli bir rol oynar.

Bu nedenle röleli lojik devrelerin mikrobilgisayarla gerçekleştirilmesi konusu üzerinde yapmış olduğum çalışmaların programlanabilir kontrol sistemlerine bir ilave olacağını zannetmekteyim.

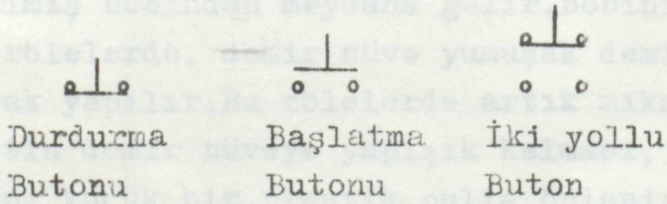
2. RÖLELİ DEVRELERDE KULLANILAN ELEMAN VE SEMBOLLERİ

2.1 GİRİŞ

Bir röleli lojik devrenin çalışma sistemini anlayabilmemiz için o devrede bulunan elemanların sembollerini, görev ve çalışma prensiplerini bilmemiz gerekmektedir. Harf ve rakamları bilmeden okuyup yazılamıyacağı gibi sembollerini bilmeden bir devrenin incelenmesi ve üzerinde gelişme yapılabilmesi mümkün değildir. Bu sebeple aşağıda röleli devrelerde kullanılan elemanların çalışmaları, yapıları ve sembollerini anlatacağız.

2.2 BUTONLAR

Röle ve kontaktörleri çalıştıran veya durduran kumanda elemanlarına buton adı verilir. Butonlar yapılarına göre durdurma, başlatma, iki ve daha fazla yollu olmak üzere üç kısma ayrılırlar.

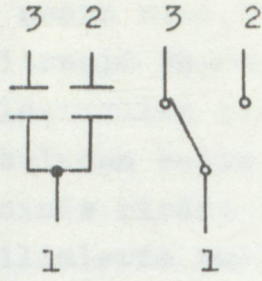


Şekil 2.1

Şekil 2.1 de sembolleri verilen butonlar çalışma şekillerine görede ani temaslı ve kalıcı olmak üzere iki kısma ayrılırlar. Kalıcı buton basıldığında konum değiştirir ve o konumda kalır, ikinci bir basma veya başka bir kumanda tarafından normal hale döndürülür. Ani temaslı butona basıldığında buton konum değiştirir, serbest bırakıldığında otomatik olarak normal konumuna döner.

2.3 MİKRO ANAHTARLAR

Yapısı ve sembolü Şekil 2.2 de görülen ve yapıları küçük olan kumanda elemanlarına mikro anahtar adı verilir. Yapılarının ufak olması nedeniyle zaman rölelerinde, proğ-

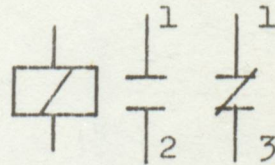
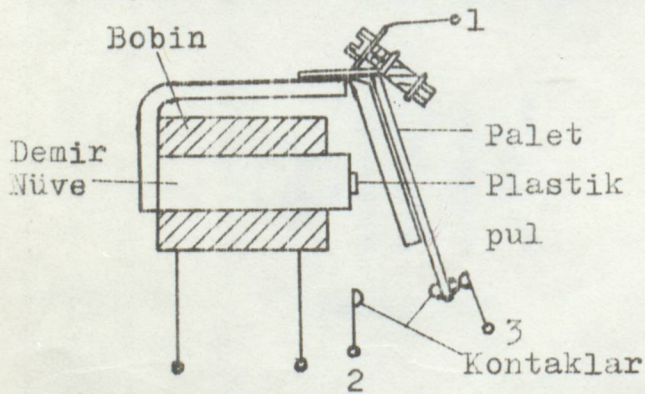


Şekil 2.2

ram şalterlerinde, basınç anahtarlarında vb. pek çok yerlerde kullanılır. Mikro anahtarlar türlerine göre bir veya daha fazla kontaklı olabilirler.

2.4 RÖLELER

Konumuzun esasını teşkil eden ve Şekil 2.3 de görülen ufak güçteki elektromagnetik anahtarlara röle adı verilir. Röleler elektromagnetik, palet ve kontaklar olmak üzere üç kısımdan oluşurlar. Elektromagnetik demir nüve üzerine sarılmış bobinden meydana gelir. Bobini doğru akıma bağlanan rölelerde, demir nüve yumuşak demirden ve bir parça olarak yapılır. Bu rölelerde artık mıknatısiyet nedeniyle paletin demir nüveye yapışık kalması, nüvenin ön yüzüne konmuş küçük bir plastik pulla önlenir. Bobini alternatif



Şekil 2.3

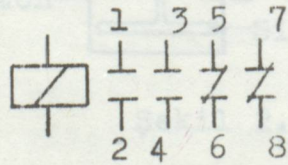
akıma bağlanan rölelerde, demir nüve sac paketinden yapılır. Alternatif akımın değer ve yön değiştirmesi, rölelerde

titreşime neden olur. Bobini alternatif akıma bağlanan bir rölenin titreşim yapması, demir nüvenin ön yüzünde acılmış oyuga yerleştirilen bir bakır halkayla önlenir. Demir nüve üzerinde bulunan bobin bir veya daha fazla sargıdan oluşur. Röle bobininde birden fazla sargının bulunması rölenin değişik gerilimlerde kullanılmasını sağlar. Röledeki kontaklar palet aracılığı ile açılır ve kapanırlar. Normal durumda palet yay ve yer çekimi nedeniyle, demir nüveden uzakta bulunur. Rölelerde normalde açık ve normalde kapalı olmak üzere iki çeşit kontak vardır. Şekil 2.3 de verilen rölenin bobinine bir gerilim uygulandığında, röle enerjilenir ve paletini çeker. Palet üzerinde bulunan 1-3 nolu kontak açılır ve 1-2 nolu kontak kapanır. Bobinin akımı kesildiğinde röle üzerinde bulunan yay paletin demir nüveden uzaklaşmasını sağlar. Bu durumda kontaklar ilk konumlarına dönerler.

Günümüzde elektronik kumanda yanında halen röleli devrelerde fazlalıkla kullanılmaktadır.

2.5 KONTAKTÖRLER

Şekil 2.4 de yapısı ve sembolü görülen büyük güçteki elektromagnetik anahtarlara kontaktör adı verilir. kontaktörlerin yapı ve çalışma prensibi rölelerde olduğu gibidir.



Şekil 2.4

Aralarındaki fark kontaktörlerin daha büyük güçlerdeki devreleri kontrol etmek amacıyla kullanılmasıdır.

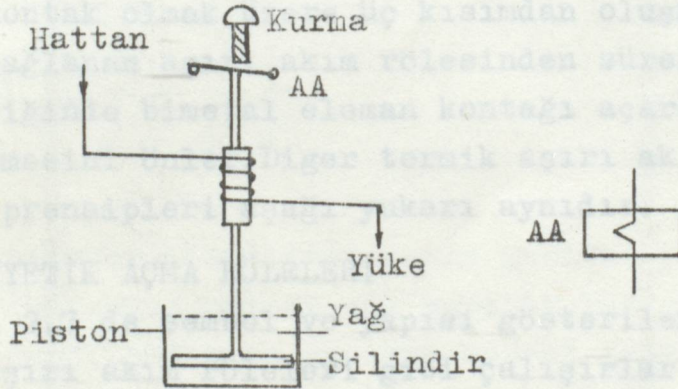
Kontaktörlerin yapılarının basit ve kolay kullanılabilir olmaları geniş bir kullanım alanı oluşturmuştur.

2.6 AŞIRI AKIM RÖLELERİ

Aşırı akımların elektrik cihazlarına vereceği zararı önlemek için kullanılan elemanlara aşırı akım rölesi denir. Devrelere seri bağlanan bu elemanlar herhangi bir sebepten fazla akım çekildiğinde devreyi açarak cihazın korunmasını sağlar. Akım röleleri cinslerine göre devreyi açtıktan sonra ya manuel olarak yada otomatik olarak normal konuma dönerler. Aşırı akım röleleri çalışma prensiplerine göre birkaç sınıfa ayrılırlar.

2.6.1 Manyetik Aşırı Akım Röleleri

Şekil 2.5 de sembol ve yapısı görülen ve akımın manyetik etkisiyle çalışan aşırı akım rölelerine manyetik aşırı akım rölesi adı verilir. Devreye seri olarak bağlanan aşırı akım rölesinden ani yüksek akım geçtiğinde (örneğin ilk anma akımları) rölenin çalışıp devreyi açması ,yağ dolu silindir içinde hareket eden bir pistonla önlenir.

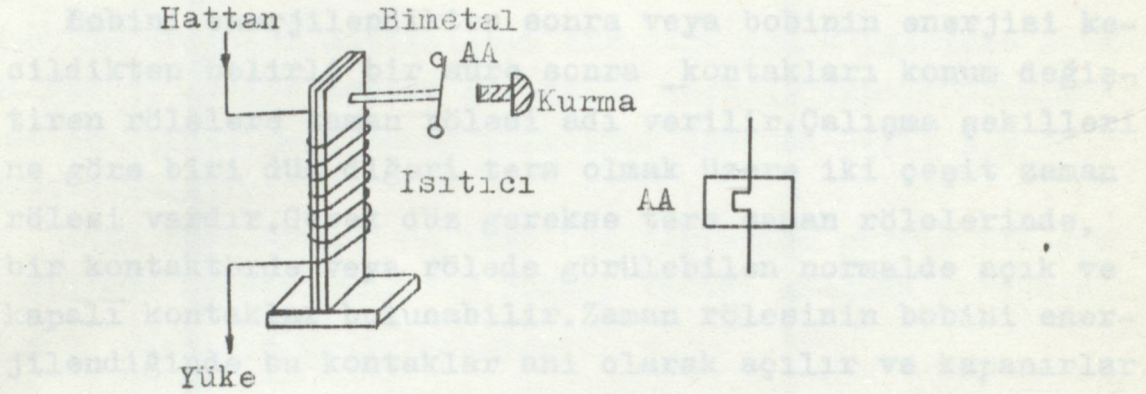


Şekil 2.5

Aşırı akım rölesinin bobininden normal değerin üzerinde bir akım geçtiğinde bobin nüveyi yukarı çeker. Silindir içinde bulunan piston nedeniyle demir nüvenin hareketi yavaş olur. Eğer akım kısa süreli ise rölenin kontağı hemen açılmaz. Eğer bobinden geçen akım normal değerine düşmesse bir süre sonra kontak açılarak devre korunmuş olur. Bu tür rölelere koruma röleleri de denir.

2.6.2 Termik Aşırı Akım Röleleri

Yükten geçen akımın yarattığı ısı etkisiyle çalışan aşırı akım rölelerine termik aşırı akım rölesi adı verilir. Termik aşırı akım rölelerinin endirek ısıtmalı, direkt ısıtmalı ve ergiyici alaşımlı olmak üzere çeşitleri vardır.

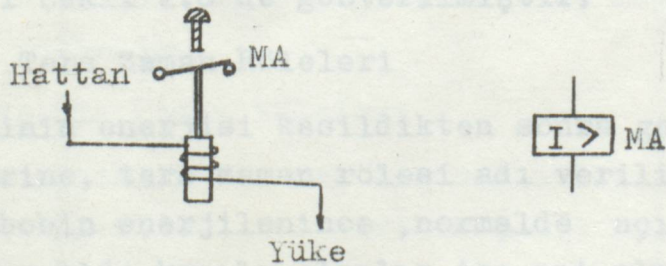


Şekil 2.6

Şekil 2.6 da endirekt ısıtmalı termik aşırı akım rölesinin yapısı ve sembolü verilmiştir. Bu röle ısıtıcı, bimetal ve kontak olmak üzere üç kısımdan oluşmuştur. Yüke seri olarak bağlanan aşırı akım rölesinden sürekli bir aşırı akım geçtiğinde bimetal eleman kontağı açarak yükten aşırı akım geçmesini önler. Diğer termik aşırı akım rölelerinin çalışma prensipleri aşağı yukarı aynıdır.

2.7 MANYETİK AÇMA RÖLELERİ

Şekil 2.7 de sembol ve yapısı gösterilen bu koruma röleleri aşırı akım röleleri gibi çalışırlar. Önceden belirttiğimiz gibi aşırı akım röleleri büyük akımlarda ve gecikmeli olarak çalışırlar. Manyetik açma röleleri ise normalden aşırı olan ani akımlarda devreyi açarak cihazları korurlar.



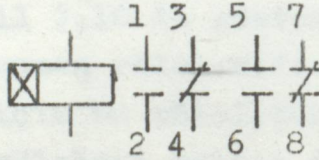
Şekil 2.7

Manyetik açma röleleri aşırı akım röleleriyle beraber ve yine devreye seri olarak kullanılırlar. Manyetik açma rölelerinin ayarlandığı değrin altındaki aşırı akımlarda aşırı akım rölesi görev yaparak devreyi korurlar.

2.8 ZAMAN RÖLELERİ

Bobini enerjilendikten sonra veya bobinin enerjisi kesildikten belirli bir süre sonra kontakları konum değiştiren rölelere zaman rölesi adı verilir. Çalışma şekilleri ne göre biri düz diğeri ters olmak üzere iki çeşit zaman rölesi vardır. Gerek düz gerekse ters zaman rölelerinde, bir kontaktörde veya rölede görülebilen normalde açık ve kapalı kontaklar bulunabilir. Zaman rölesinin bobini enerjilendiğinde bu kontaklar ani olarak açılır ve kapanırlar. enerjisi kesildiğinde ise ani olarak normal konumlarına dönerler.

2.8.1 Düz Zaman Röleleri

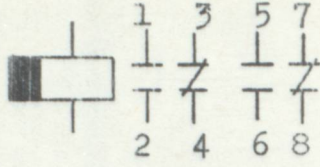


Şekil 2.8

Bobini enerjilendikten sonra gecikme yapan zaman rölelerine düz zaman rölesi adı verilir. Düz zaman rölelerinde bobin enerjilendikten gecikme süresi kadar sonra rölenin normalde açık olan kontakları kapanır, kapalı olanlar ise açılırlar. bobinin enerjisi kesilir kesilmez kontaklar ani olarak eski konumlarına dönerler. Düz zaman rölesinin sembolleri Şekil 2.8 de gösterilmiştir.

2.8.2 Ters Zaman Röleleri

Bobinin enerjisi kesildikten sonra gecikme yapan zaman rölelerine, ters zaman rölesi adı verilir. Ters zaman rölesinde bobin enerjilenince, normalde açık kontaklar kapanır, normalde kapalı olanlar ise ani olarak açılırlar.



Şekil 2.9

Bobinin enerjisi kesildikten bir süre sonra (gecikme süresi kadar) evvelce açılan kontaklar kapanır kapalı olanlar ise açılırlar. Ters zaman rölesinin sembolü Şekil 2.9 da gösterilmiştir.

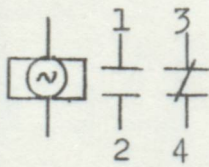
Zaman rölelerinin çalışma prensiplerine görede bir çok çeşitleri vardır. Bunların birkaç tanesi aşağıda verilmiştir.

2.8.3 Pistonlu Zaman Röleleri

Gecikme işlevinin mekaniksel olarak bir piston ile yapıldığı zaman röleleridir.

2.8.4 Motorlu Zaman Röleleri

Şekil 2.10 da gösterilen ve gecikmesi bir motor ile sağlanan zaman röleleridir. Her çalışmasında aynı süreyi elde etmek için bu rölelerde senkron motor kullanılır. Motorlu zaman röleleri yalnız düz zaman rölesi olarak yapılırlar.

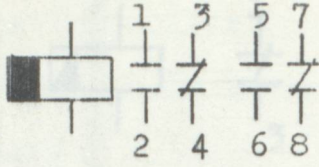


Şekil 2.10

Gecikme süreleri diğer çeşitlere göre daha geniş sınırdadır ve daha kesindir.

2.8.5 Termik Zaman Röleleri

Zaman gecikmesinin ısı ile sağlandığı zaman rölelerine termik zaman rölesi adı verilir. Şekil 2.11 de sembolü verilen röle ısıtıcı, bimetal ve kontaklardan oluşmuştur.



Şekil 2.9

Bobinin enerjisi kesildikten bir süre sonra (gecikme süresi kadar) evvelce açılan kontaklar kapanır kapalı olanlar ise açılırlar. Ters zaman rölesinin sembolü Şekil 2.9 da gösterilmiştir.

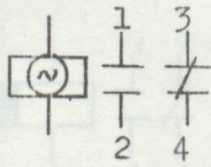
Zaman rölelerinin çalışma prensiplerine görede bir, çok çeşitleri vardır. Bunların birkaç tanesi aşağıda verilmiştir.

2.8.3 Pistonlu Zaman Röleleri

Gecikme işlevinin mekaniksel olarak bir piston ile yapıldığı zaman röleleridir.

2.8.4 Motorlu Zaman Röleleri

Şekil 2.10 da gösterilen ve gecikmesi bir motor ile sağlanan zaman röleleridir. Her çalışmasında aynı süreyi elde etmek için bu rölelerde senkron motor kullanılır. Motorlu zaman röleleri yalnız düz zaman rölesi olarak yapılırlar.

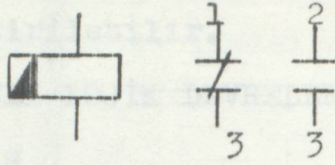


Şekil 2.10

Gecikme süreleri diğer çeşitlere göre daha geniş sınırdadır ve daha kesindir.

2.8.5 Termik Zaman Röleleri

Zaman gecikmesinin ısı ile sağlandığı zaman rölelerine termik zaman rölesi adı verilir. Şekil 2.11 de sembolü verilen röle ısıtıcı, bimetal ve kontaklardan oluşmuştur.

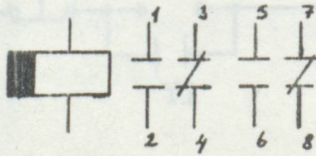


Şekil 2.11

Isıtıcının akımı bir direnç ile sınırlandırılmıştır. Röle devreye girdikten bir süre sonra bimetal, ısınn etkisiyle bir kontağı açıp diğerini kapatır. Termik zaman röleleri basit yapıları ve ucuz olmaları nedeniyle kumanda devrelerinde sıklıkla kullanılırlar. Zaman ayarları tam olmakla birlikte sınırlama direncinin değerini değiştirmek suretiyle değiştirilebilir.

2.8.6 Doğru Akım Zaman Röleleri

Bir röle ile yardımcı elemanlardan oluşan ve yalnız doğru akımla çalışan zaman röleleridir. Yapıları basit ve kullanımları kolay olan pekçok doğru akım zaman rölesi vardır. Şekil 2.12 de verilen kondansatörlü zaman röleside bunlardan biridir. Bu zaman rölesi bir doğru akım rölesi ve buna paralel bağlanan bir kondansatörden oluşmuştur.



Şekil 2.12

Bu röle enerjilendiğinde normalde kapalı olan kontaklar açılır açık olanlar ise kapanırlar. Aynı esnada kondansatör besleme gerilimine şarj olur. Rölenin enerjisi kesildiğinde şarj olan kondansatör röle bobininden boşalınca kadar kontaklar konum değiştirmez. Böylece bir gecikme sağlanmış olur. Görüldüğü gibi ters zaman rölesi olarak çalışır. Kondansatörün değerini değiştirilmele gecikme süresi değiştirilebilir.

şırlar.Kondansatörün değerini değiştirmekle gecikme süresi değiştirilebilir.

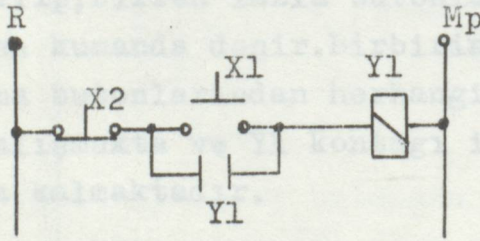
3. RÖLELİ LOJİK DEVRELERİN İNCELENMESİ

3.1 GİRİŞ

Bu bölümde röleli lojik devrelerin mikrobilgisayara uygulanmasında yardımcı olmak üzere bazı örnekler verilecektir.Önceki bölümde anlatmış olduğumuz röle ve diğer elemanlarla oluşturulan ladder diagramlarını(röleli devre şemaları) inceleyerek, bunların çalışma şekilleri ve lojik ifadelerinin çıkarılması anlatılacaktır.

3.2 ANİ TEMASLI BUTONLARLA YAPILAN KUMANDA DEVRESİ

Şekil 3.1 de ani temaslı butonlarla yapılan bir kumanda devresi verilmiştir.Bu devrede X1 başlatma butonuna basıldığında Y1 rölesi enerjilenir ve Y1 kontağı(Mühürleme kontağı) kapanır.X1 başlatma butonu bırakılsa bile Y1 rölesi çalışmasına devam eder.Başlatma butonuna sürekli basmak gerekmez.X2 durdurma butonuna basıldığında Y1 rölesinin enerjisi kesilerek röle ilk konumuna döner.



Şekil 3.1

Şekilde görülen devrede X1 başlatma ve X2 durdurma butonlarını birer lojik giriş ifadesi , Y1 rölesinide çıkış ifadesi olarak kabul edersek aralarında şu şekilde bir lojik bağlantı kurulabilir.

$$Y1 = (X1 + Y1) \cdot \overline{X2} \quad \text{veya}$$

$$Y1 = (X1 \text{ OR } Y1) \text{ AND NOT } X2$$

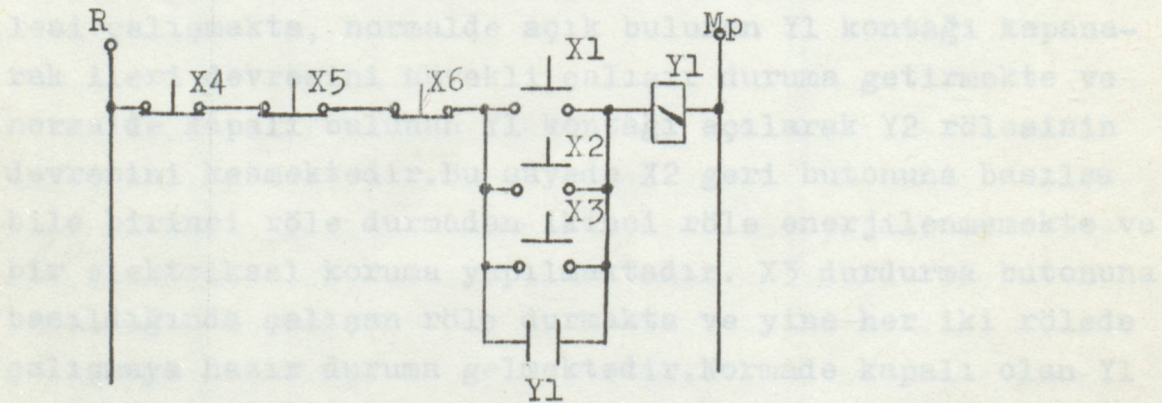
Burada butonlara basıldığında lojik 1 değerini alacakları kabul edilerek bu ifadeler yazılmıştır. Buna göre bu devrenin doğruluk tablosu;

X1	X2	Y1	Y1
0	0	0	0
0	0	1	Tanımsız
0	1	0	0
0	1	1	Tanımsız
1	0	0	0
1	0	1	1
1	1	0	Tanımsız
1	1	1	1

Şeklinde olacaktır. Yazılan ifade ve doğruluk tablosunda görüleceği gibi Y1 çıkışının yine kendi ifadesi içerisinde kullanılması lojik olarak bazı tanımsız durumlara neden olmaktadır.

3.3 RÖLELİ UZAKTAN KUMANDA DEVRESİ

Şekil 3.2 de görülen birden fazla ani temaslı butonla çalıştırılıp, birden fazla butonla durdurulabilen sistemlere uzaktan kumanda denir. Birbirine paralel bağlanmış olan çalıştırma butonlarından herhangi birine basıldığında Y1 rölesi çalışmakta ve Y1 kontağı ile kilitlenerek çalışma konumunda kalmaktadır.



Şekil 3.2

Birbirine seri olarak bağlanmış olan ve normalde kapalı bulunan durdurma butonlarından her hangi birisine basıldığında Y1 rölesinin enerjisi kesilerek sistem ilk durumuna döner. Buna göre sistemin lojik ifadesi

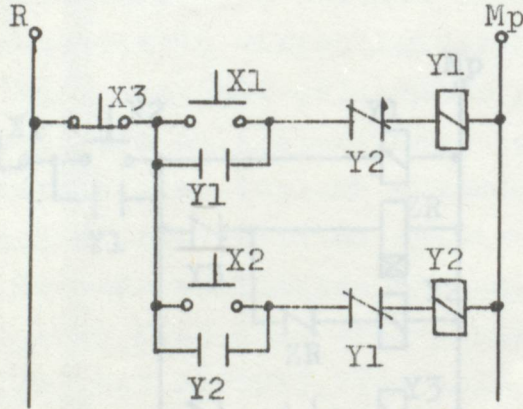
$$Y1 = (Y1 + X1 + X2 + X3) \cdot \overline{X4} \cdot \overline{X5} \cdot \overline{X6} \quad \text{veya}$$

$$Y1 = (Y1 \text{ OR } X1 \text{ OR } X2 \text{ OR } X3) \text{ AND NOT } X4 \text{ AND NOT } X5 \text{ AND NOT } X6$$

olacaktır.

3.4 DÖNÜŞ YÖNÜ DEĞİŞTİRME DEVRESİ

Şekil 3.3 de görülen ladder diagramında kullanılan devre genellikle elektrik motorlarının çalışma yönünü değiştirmekte kullanılır.



Şekil 3.3

Bu devrede X1 ileri butonuna basıldığında Y1 ileri rölesi çalışmakta, normalde açık bulunan Y1 kontağı kapanarak ileri devresini sürekli çalışır duruma getirmekte ve normalde kapalı bulunan Y1 kontağı açılarak Y2 rölesinin devresini kesmektedir. Bu sayede X2 geri butonuna basılsa bile birinci röle durmadan ikinci röle enerjilenmemekte ve bir elektrikselsel koruma yapılmaktadır. X3 durdurma butonuna basıldığında çalışan röle durmakta ve yine her iki rölede çalışmaya hazır duruma gelmektedir. Normalde kapalı olan Y1 ve Y2 kontakları sayesinde bir röle çalışırken diğerinin çalışması engellenerek bir motor kumandası sırasında güç

devresinin kısa devre olması önlenmiş olur. Sistemin lojik ifadesi ise

$$Y1 = (X1 + Y1) \cdot \overline{X3} \cdot \overline{Y2} \text{ veya}$$

$$Y1 = (X1 \text{ OR } Y1) \text{ AND NOT } X3 \text{ AND NOT } Y2$$

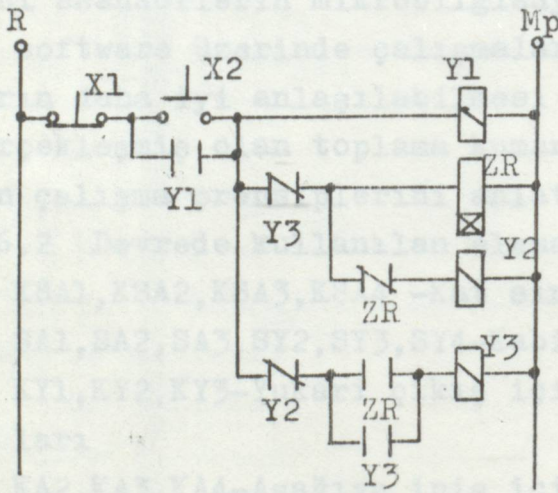
$$Y2 = (X2 + Y2) \cdot \overline{X3} \cdot \overline{Y1} \text{ veya}$$

$$Y2 = (X2 \text{ OR } Y2) \text{ AND NOT } X3 \text{ AND NOT } Y1$$

şeklinde olacaktır.

3.5 BİR MOTORA YILDIZ-ÜÇGEN YOL VERME

Şekil 3.4 de üç fazlı bir asekron motora otomatik yıldız-üçgen yol vermede kullanılan ladder diagramı verilmiştir. Bu örneği zaman rölesinin çalışmasını anlatmak için vermektediriz.



Şekil 3.4

Bu devrede X2 başlatma butonuna basıldığında Y1 ve Y2 kontaktörleri enerjilenerek motor yıldız bağlantıda çalışmaya başlar. Aynı anda zaman röleside enerjilenir. Gecikme süresi sonunda ZR kontaklarının durumu değişir. Böylece Y3 kontaktörü enerjilenir ve Y2 kontaktörü devre dışı olarak motor üçgen çalışmaya başlar. Bu esnada ZR röleside devre dışı olur. X1 durdurma butonuna basıldığında Y1 ve Y3 kontaktörlerinin enerjisi kesilerek motor durur.

Devrenin lojik ifadesi

$Y1 = \text{NOT } X1 \text{ AND } (X2 \text{ OR } Y1)$

$ZR = \text{NOT } X1 \text{ AND } (X2 \text{ OR } Y1) \text{ AND NOT } Y3$

$Y2 = \text{NOT } X1 \text{ AND } (X2 \text{ OR } Y1) \text{ AND NOT } Y3 \text{ AND NOT } ZR$

$Y3 = \text{NOT } X1 \text{ AND } (X2 \text{ OR } Y1) \text{ AND NOT } Y2 \text{ AND } (ZR \text{ OR } Y3)$

olacaktır.

3.6 TOPLAMA KUMANDALI VE BİR HIZLI ASANSÖR

3.6.1 Giriş

Günümüzde röleli lojik devrelerin en çok kullanıldıkları yerlerden biride asansörlerdir. Asansörlerin bir hızlı, iki hızlı, elle kumandalı, otomatik bir kumandalı, toplama kumandalı, grup kumandalı ve bunların kombinezonları olan pek çok çeşiti vardır. Bu nedenle ileriki konularda görüleceği gibi asansörlerin mikrobilgisayar ile kumandası için gerekli li software üzerinde çalışmalar yapılmıştır. İleriki konuların daha iyi anlaşılabilmesi için röleli lojik devreyle gerçekleşmiş olan toplama kumandalı ve bir hızlı asansörün çalışma prensiplerini anlatacağız.

3.6.2 Devrede kullanılan elemanlar

KSA1, KSA2, KSA3, KSA4 -Kat sınır anahtarları

SA1, SA2, SA3, SY2, SY3, SY4-Kabin hareketini izleme röleleri

KY1, KY2, KY3-Yukarı çıkış için katlardaki çağırma butonları

KA2, KA3, KA4-Aşağıya iniş için katlardaki çağırma butonları

RY2, RY3, RY4-Yukarı çıkış için kat röleleri

RA1, RA2, RA3-Aşağıya iniş için kat röleleri

KK1, KK2, KK3, KK4-Kat kapı kontakları

KKK-Kabin kapı kontağı

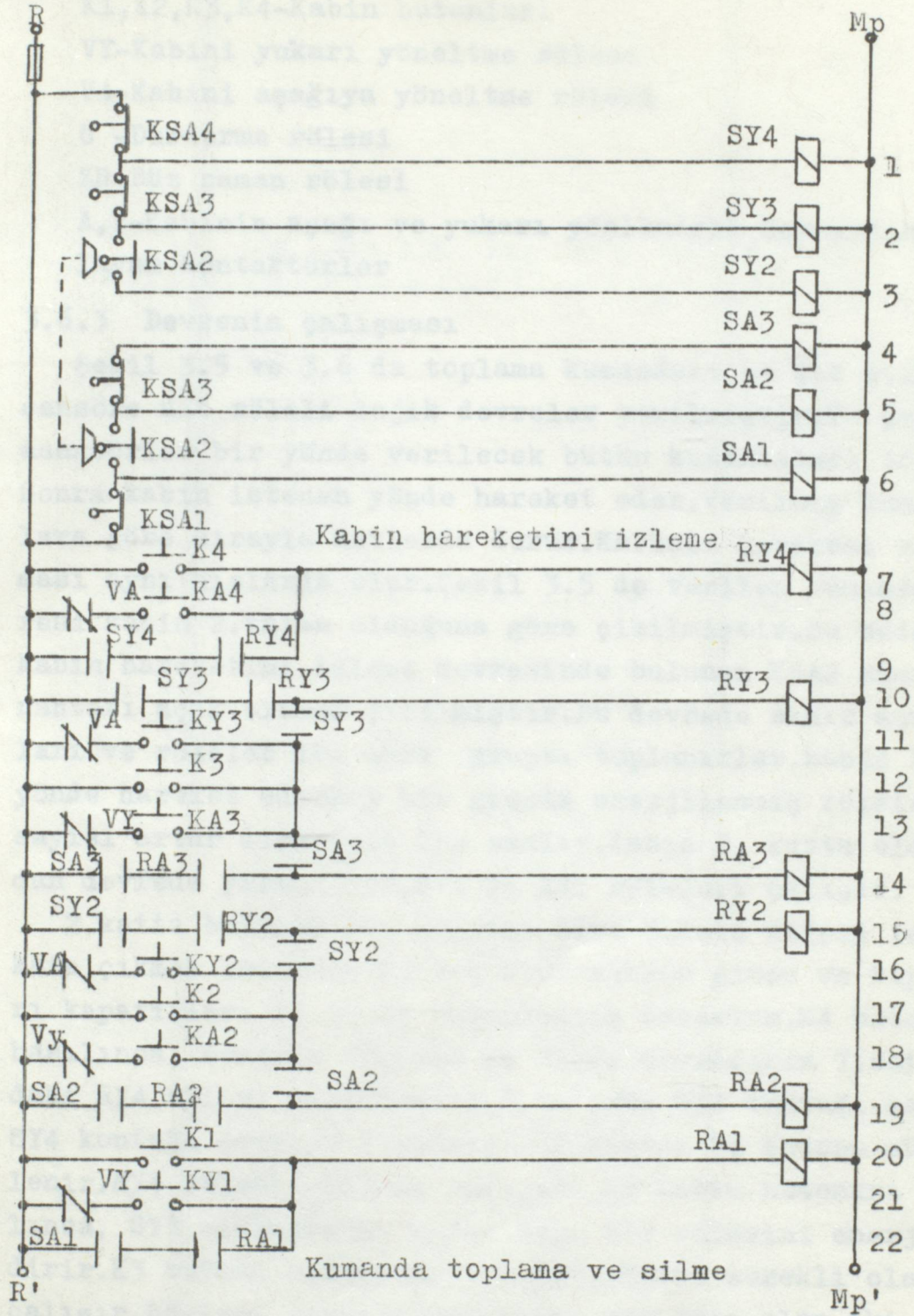
HGK-Halat gevşeme kontağı

HDK-Hız düzeneç kontağı

FSR-Faz sırası rölesi

İB-İmdat butonu

YR-Kumanda kesme rölesi



Şekil 3.5

K1,K2,K3,K4-Kabin butonları

VY-Kabini yukarı yöneltme rölesi

VA-Kabini aşağıya yöneltme rölesi

U -Durdurma rölesi

ZR-Düz zaman rölesi

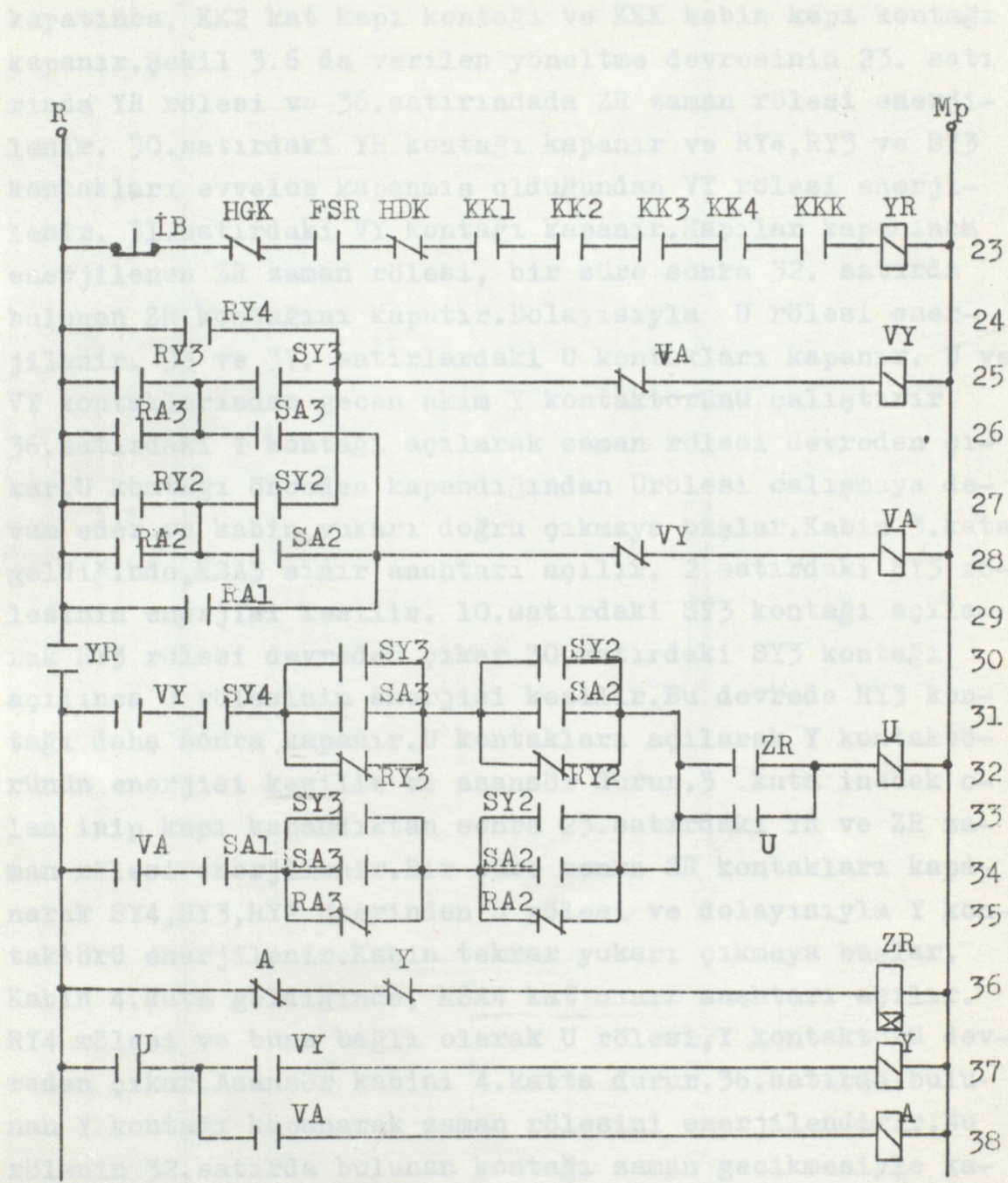
A,Y-Kabinin aşağı ve yukarı yönlerdeki hareketini sağlayan kontaktörler

3.6.3 Devrenin çalışması

Şekil 3.5 ve 3.6 da toplama kumandalı ve bir hızlı asansöre ait röleli lojik devreler verilmiştir.Bu çeşit asansörler bir yönde verilecek bütün kumandaları toplar. Sonra kabin istenen yönde hareket eder.Verilmiş kumandalara göre sırayla katlarda durur.Kabinin hareketi ve durması aynı hızlarda olur.Şekil 3.5 de verilen kumanda devresi kabin 2.katta olduğuna göre çizilmiştir.Bu nedenle kabin hareketini izleme devresinde bulunan KSA2 sınır anahtarı açık olarak çizilmiştir.Bu devrede sınır anahtarları ve röleler iki ayrı grupta toplanırlar.kabin bir yönde hareket ederken bir grupta enerjilenmiş rölelerin sayısı artar diğerinde ise azalır.Kabin 2. katta olduğundan devrede yalnız SY4,SY3 ve SA1 röleleri çalışır.

2.katta bulunan iki kişiden biri 3.kata diğeri ise 4. kata çıkmak isterse, bu iki kişi kabine girer ve kapıları kapatırlar. K3 ve K4 butonlarına basarlar.K4 butonuna basılınca, kumanda toplama ve silme devresinin 7.satırındaki RY4 rölesi enerjilenir.9.satırda RY4 kontağı kapanır. SY4 kontağı evvelce kapanmış olduğundan K4 butonu mühürlenir.RY4 rölesi sürekli çalışır. K3 kabin butonuna basılınca, SY3 kontağından geçen akım RY3 rölesini enerjilendirir.K3 butonu mühürlenir ve RY3 rölesi sürekli olarak çalışır.Böylece asansör kendisine verilmiş olan iki kumandayı toplamış olur.

Kabin 2.katta olduğundan , kabin hareketini izleme devresinde SY4,SY3 ve SA1 röleleri sürekli olarak çalışırlar. 3. ve 4. kata çıkmak isteyen kişiler kabine girip kapıları



Yöneltilme Devresi

Şekil 3.6

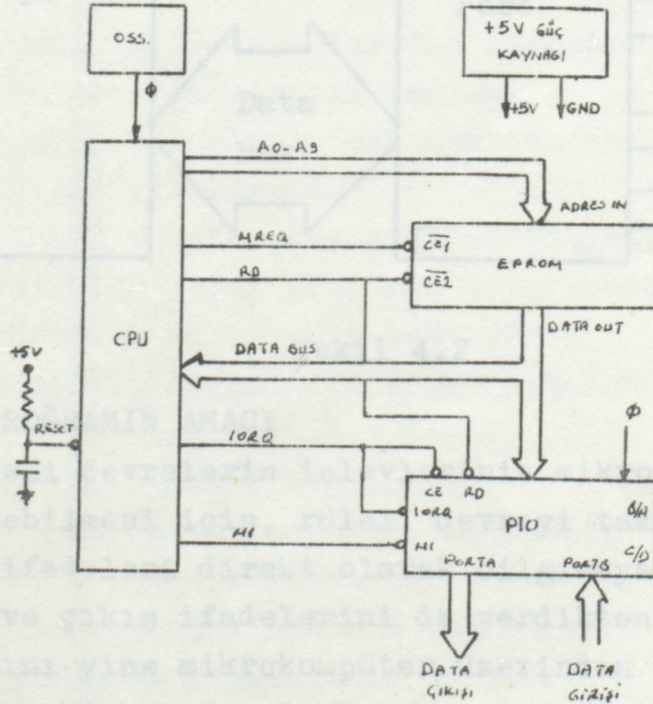
kapatınca, KK2 kat kapı kontağı ve KKK kabin kapı kontağı kapanır.Şekil 3.6 da verilen yöneltme devresinin 23. satırında YR rölesi ve 36.satırındada ZR zaman rölesi enerjilenir. 30.satırdaki YR kontağı kapanır ve RY4,RY3 ve SY3 kontakları evvelce kapanmış olduğundan VY rölesi enerjilenir. 31.satırdaki VY kontağı kapanır.Kapılar kapanınca enerjilenen ZR zaman rölesi, bir süre sonra 32. satırda bulunan ZR kontağını kapatır.Dolayısıyla U rölesi enerjilenir. 33 ve 37. satırlardaki U kontakları kapanır. U ve VY kontaklarından geçen akım Y kontaktörünü çalıştırır. 36.satırdaki Y kontağı açılarak zaman rölesi devreden çıkar.U kontağı önceden kapandığından Urölesi çalışmaya devam eder.ve kabin yukarı doğru çıkmaya başlar.Kabin 3.kata geldiğinde,KSA3 sınır anahtarı açılır. 2.satırdaki SY3 rölesinin enerjisi kesilir. 10.satırdaki SY3 kontağı açılarak RY3 rölesi devreden çıkar.30.satırdaki SY3 kontağı açılınca U rölesinin enerjisi kesilir.Bu devrede RY3 kontağı daha sonra kapanır.U kontakları açılarak Y kontaktörünün enerjisi kesilir ve asansör durur.3 .kata inecek olan inip kapı kapandıktan sonra 23.satırdaki YR ve ZR zaman rölesi enerjilenir.Bir süre sonra ZR kontakları kapanarak SY4,RY3,RY2 üzerinden U rölesi ve dolayısıyla Y kontaktörü enerjilenir.Kabin tekrar yukarı çıkmaya başlar. Kabin 4.kata geldiğinde, KSA4 kat sınır anahtarı açılır. RY4 rölesi ve buna bağlı olarak U rölesi,Y kontaktörü devreden çıkar.Asansör kabini 4.katta durur.36.satırda bulunan Y kontağı kapanarak zaman rölesini enerjilendirir.Bu rölenin 32.satırda bulunan kontağı zaman gecikmesiyle kapanır.ZR kontağı kapanıncaya kadar asansör çalışmaz.4 kata gelen kişi de dışarıya çıkma olanağını bulur.

Eğer bu komutlardan biri kabin içinden diğeri dışarıdan veya her ikiside dışarıdan verilmiş olsaydı asansör yine yukarıda açıklandığı gibi çalışırdı.Bu örneklerdende görüldüğü gibi hareket yönündeki kumandalar değerlendirilmektedir.

4.RÖLELİ LOJİK DEVRELERİN MİKROBİLGİSAYARA UYGULANMASI

4.1 GİRİŞ

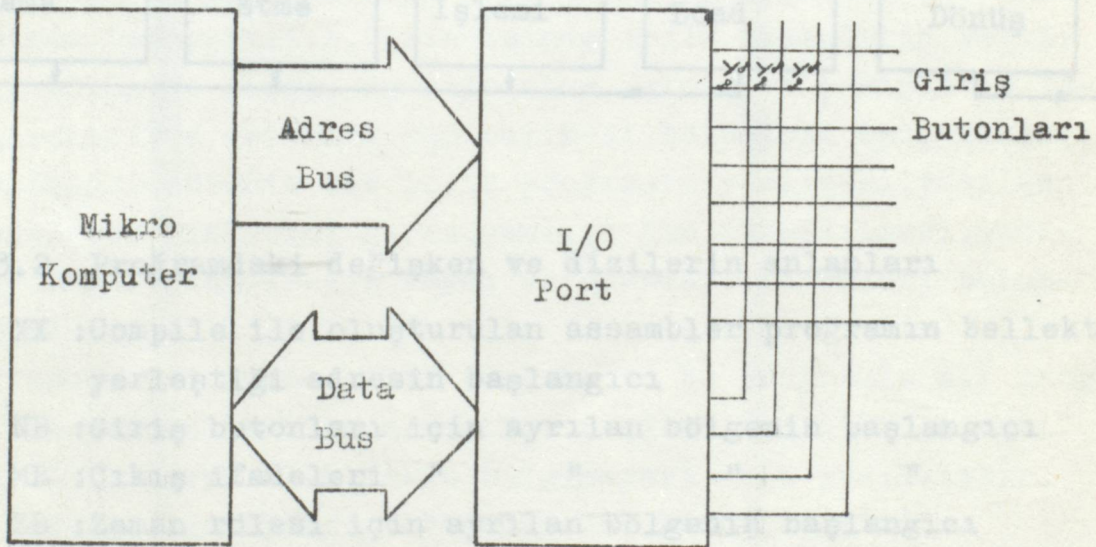
Bir önceki bölümde anlattığımız gibi röleler ve röleli devreler elektromekanik kökenli sistemlerdir. Rölelerin kontaktlarının açma kapama sayıları yüksek olmasına rağmen, mekaniksel olduklarından sınırlıdır. Röleli sistemler, elektronik sistemlere göre dış etkilere daha kolay etkilenebilirler. Harcadıkları enerji ve kapladıkları alan daha büyüktür. Yukarıda dört katlı bir bina için anlattığımız bir hızlı toplama kumandalı asansörde onyediyet adet röle kullanılmıştır. Yani her kat için ortalama dört röle kullanılmaktadır. Buda ilave her kat için yaklaşık dört röle ilave edilmesi demektir. Halbuki tasarlamış olduğumuz mikroislemcili daha genel bir ifadeyle mikrobilgisayarlı bir sistemde ilave bir hardware devre olmadan, birkaç rölenin oluşturduğu devreden birkaç yüz rölenin oluşturduğu devrelerin işlevleri software ile kolayca gerçekleştirilebilir.



ŞEKİL 4.1

Şekil 4.1 de görülen devre şeması direkt olarak mikro işlemci ile yapılan bir sistemi göstermektedir. Bu basit sistem bir mikroişlemci, bir giriş çıkış portu, bir osilatör, besleme kaynağı ve gerekli programı içeren bellek ünitesinden oluşmuştur. Bu devre basit yapısıyla pek çok rölenin yapabileceği işleri yapabilir. Fakat buradaki zorluk kullanılan mikroişlemcinin assambler dilini bilmek ve her yeni lojik devre için programın yenilenmesi gerekliliğidir.

Bizim çalışmalarımız daha komplike olan mikrokomputer ve basic dili ile dir. Basit devresi 4.2 de görülen devre ve yazmış olduğumuz programın yardımıyla birçok değişik röleli devrenin simülasyonu yapılabilir.



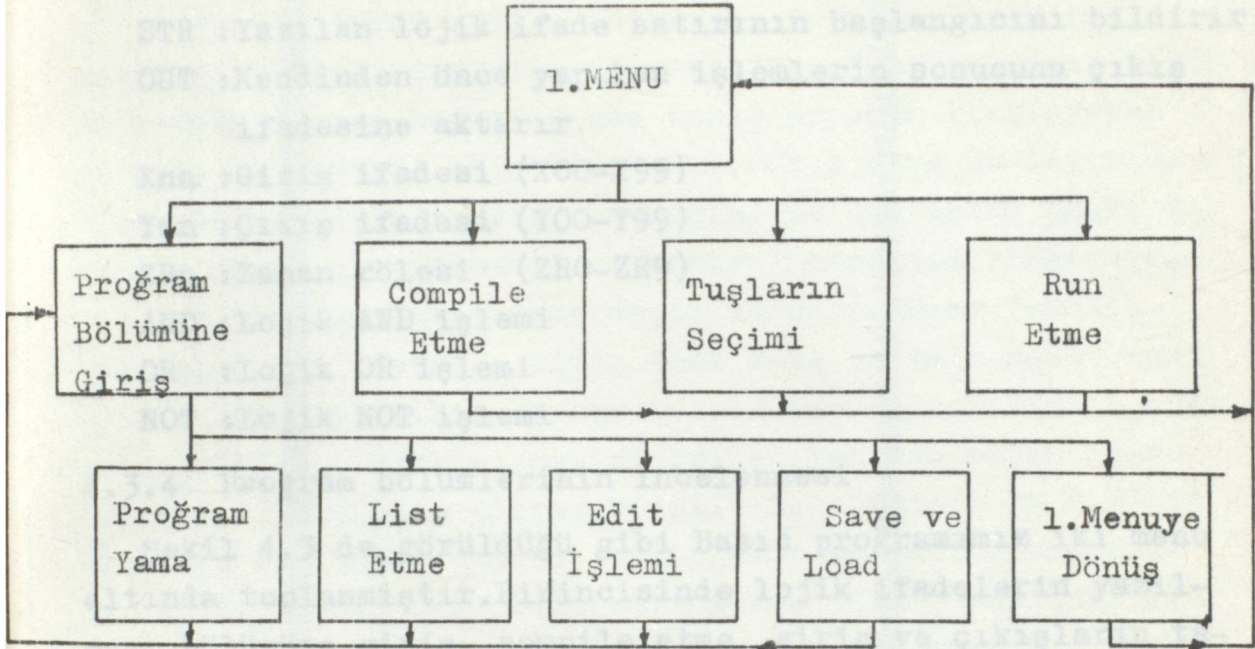
Şekil 4.2

4.2 PROGRAMIN AMACI

Röleli devrelerin işlevlerinin mikrobilgisayarla gerçekleştirilebilmesi için, röleli devreyi takip ederek yazılan lojik ifadeleri direkt olarak bilgisayara girmek, gerekli giriş ve çıkış ifadelerini de verdikten sonra devrenin çalışmasını yine mikrokomputer üzerinden izleyebilmek için yazılmış bir programdır.

4.3 PROGRAMIN İNCELENMESİ

4.3.1 Programın blok şeması



4.3.2 Programdaki değişken ve dizilerin anlamları

- XX :Compile ile oluşturulan assambler programın bellekte yerleştiği adresin başlangıcı
- NB :Giriş butonları için ayrılan bölgenin başlangıcı
- MB :Çıkış ifadeleri " " " "
- ZB :Zaman rölesi için ayrılan bölgenin başlangıcı
- I :Girilen lojik ifadenin satır numarası
- LB :Gerekli adreslerin low byte'ı
- HB :Gerekli adreslerin hight byte'ı
- INM :En büyük giriş indisi
- CIKM :En büyük çıkış indisi
- ZRM :En büyük zaman rölesi indisi
- PR\$(X):Lojik ifadelerin bulunduğu dizi
- LPR(X):Her lojik satırın uzunlukları
- TM(X) :Zaman rölelerinin süreleri

4.3.3 Lojik ifadelerin programa aktarılmasında kullanılan ifadeler

STR :Yazılan lojik ifade satırının başlangıcını bildirir

OUT :Kendinden önce yapılan işlemlerin sonucunu çıkış ifadesine aktarır.

Xnn :Giriş ifadesi (X00-X99)

Ynn :Çıkış ifadesi (Y00-Y99)

ZRn :Zaman rölesi (Z00-ZR9)

AND :Lojik AND işlemi

OR :Lojik OR işlemi

NOT :Lojik NOT işlemi

4.3.4 Program bölümlerinin incelenmesi

Şekil 4.3 de görüldüğü gibi Basic programımız iki menu altında toplanmıştır.Birincisinde lojik ifadelerin yazılması bölümüne giriş, compile etme, giriş ve çıkışların tanımlanması ve devrenin run edilmesi bölümleri bulunmaktadır.İkinci bölümde ise lojik programın yazılması,yazılan ifadelerin listesini görme,yanlışlığın düzeltilmesi,yazılan ifadelerin diskete kaydı ve okunması bölümleri bulunmaktadır.

Yazmış olduğumuz basic program ve bu programda alt program olarak kullanılan assambler program Z80 mikroişlemci temelli Amstrad 6128 mikro bilgisayarı için yazılmıştır. Bu nedenle Assambler programlar Z80 komutlarıyladır.

4.3.4.1 Lojik ifadelerin bilgisayara girilmesi bölümü

Yazmış olduğumuz program run edildiğinde,

f0 :STR

f1 :NOT

f2 :AND

f3 :OR

f4 :OUT

olarak deyimler dörder karakter uzunluğunda fonksiyon tuşlarına atanmaktadır.Compile işlemi sırasında lojik ifade dizileri dörder karakter olarak işleme tabi tutulmaktadır.

Bunedenle işlemlerin, giriş ve çıkış ifadelerinin dörder karakter uzunluğunda olmaları gerekmektedir.

Programda INPUT PR\$(I) satırıyla yazılan lojik ifade satırları PR\$(I) dizisine alınmaktadır. Yazılan satır bitip de return tuşuna basıldığında basic program ilerleyerek yazılan bu satırda AND, OR, NOT, STR, OUT, X, Y ve ZR deyimlerinin doğru yerleşip yerleşmediğini ve bunlardan başka tanımsız deyim olup olmadığını kontrol etmektedir. Eğer yanlış yerleşim veya tanımsız deyim varsa program "yanlış başlangıç var", "yanlış bitiş var" veya "hatalı komut var" uyarılarını yapar. Daha sonra tekrar başa dönerek yine aynı satırı doğru olarak girmemizi bekler. Yazılan satır doğru ise PR\$(I) dizisine alınır. Compile işlemi için bu dizinin uzunluğu (karakter sayısı) belirlenir ve satır numarası bir arttırılarak yeni bir satır girme işlemi beklenir. Eğer yazdığımız lojik satırlar bitti ise bizden yeni bir satır girmemizi bekleyen programa SON komutu yazılarak menu ye dönlür.

4.3.4.2 Yazılan satırların görülmesi bölümü

1960-2020 satır numaraları arasında olan bu bölümde önceden PR\$(I) dizisine girmiş olduğumuz satırları ekrana print ederek görmemiz sağlanır. Program herbir satırı bastıktan sonra her hangi bir tuşa basmamızı bekler. Böylece teker teker satırları kontrol etme olanağını verir. Satırlar bittikten sonra yine herhangi bir tuşa basıldığında menüye dönlür.

4.3.4.3 Edit işlemi

1910-1950 satırlar arasında olan bu bölümde INPUT" kaçıncı satır"; I komutu ile hangi satırı düzeltmek istediğimiz sorulur. Bunu girdikten sonra hatalı satırda ekrana basılarak bizden yeniden doğru işlemi girmemiz beklenir. Girilen satır yazılım yönünden doğruluğu kontrol edildikten sonra menüye dönlür.

4.3.4.4 Save ve Load işlemleri

Yazmış olduğumuz lojik ifade satırlarını disket üzerinde saklamak için `INPUT"FILE ISMINI GIRINIZ";FL$` satırı ile file ismi girildikten sonra bu ismin 8 karakterden uzun olup olmadığı kontrol edilir. Eğer uzun ise yeniden bir file ismi girmemiz beklenir. İsim geçerli ise disket üzerinde bu isimde bir file açılarak I, INM, CIKM, ZRM değişkenleri ve PR\$(X), LPR(X), XX\$(X), YY\$(X), TM(X) dizileri save edilir. Bu değişkenleri disketten yüklemek için yine file ismini vererek yüklemeyi yapabiliriz. 1750-1900 satırları arasında yazılmış bu bölümde işlem bittikten sonra otomatik olarak menu ye dönmektedir.

4.3.4.5 COMPILE etme bölümü

Programımızın en önemli olan bu bölümünde, PR\$(I) dizisindeki lojik ifadeler compile edilmekte yani bir çevirme işlemine tabi tutulmaktadırlar. Bir string dizisindeki komut ifadelerini bir basic satırı haline getirmek zor olmakla birlikte kolay bile olsa sonuçta oluşan yine bir basic program olacaktır. Bu durumda direkt olarak bir basic satırı yazmakla böyle bir işlem arasında pek bir fark olmayacaktır.

Bilindiği gibi mikroişlemciler yalnız assambler dili ile çalıştıklarından basic programlar bir basic compiler aracılığı ile işletilmektedir. Bu nedenle assambler dilde yazılmış programlara göre oldukça yavaştırlar. Bu yüzden programımızın compile bölümünde AND, OR, NOT komutlarını içeren string lojik ifadeler, bu işlemlerin assambler dilde yapılacağı bir şekile çevrilerek bellekte belirlenen bir bölgeye yerleştirilmektedir. Yani basic program halinde olan compile bölümü, girilen lojik ifadeye göre değişen assambler programlar üretmektedir.

Yukarıda belirtilen işlemi yapabilmek için belleği dört ayrı bölgeye ayırdık. Bu bölge adresleri ve içerikleri şekil 4.4 de gösterilmiştir

Ana Program		Girişler		Normal Çıkışlar	Zaman rölesi	
4000	X1	5800	Y1	6000	ZR1	6200
4001	$\overline{X1}$	5801	$\overline{Y1}$	6001	ZR1	6201
4002	X2	5802	Y2	6002	$\overline{ZR1}$	6202
4003	$\overline{X2}$	5803	$\overline{Y2}$	6003	ZR2	6203
.						
.						

Şekil 4.4

Girişler X00,X01,...Xnn olarak ani temaslı butonlar şeklinde düşünülmüş olup, giriş için kullanılan her bir buton için bellekte ikişer byte lık yer kullanılmaktadır.Şekil 4.4 İlk byte o butonun normalde açık, ikincisi ise normalde kapalı kontakları için kullanılmakta ve normalde ilk byte lojik 0, ikinci byte ise lojik 1 olarak bir alt program ile şartlandırılmaktadır.

Çıkışlar iki farklı şekilde olup normal röleler için Y00,Y01,...Ynn çıkış ifadeleri olmak üzere yine,kullanılan her bir çıkış için bellekte ikişer byte lık yerler ayrılmaktadır.Ayrılan ilk byte rölenin normalde açık, ikinci byte ise normalde kapalı olan kontakları için kullanılmaktadır.Röle enerjilendiğinde ilk byte lojik 1, ikinci byte ise onun evriği lojik 0 ,enerjisi kesildiğinde ise ilk byte lojik 0 dolayısıyla ikinci byte lojik 1 olmaktadır.Bu şartlandırma işlemleri Assambler dilde yazılmış bir alt program tarafından yapılmaktadır.

İkinci farklı çıkış olan zaman rölelerinin ZR1,ZR2,...ZRn devrede kullanılan her biri için üçer byte lık yer kullanılmaktadır.Bunlardan ilki zaman rölesi enerjilendiğinde lojik 1,enerjisi kesildiğinde lojik 0 olmaktadır.İkinci byte zaman rölesinin normalde açık olan kontaklarını, üçüncü byte normalde kapalı kontakları için kullanılmaktadır.Zaman rö-

lesi enerjilendikten gecikme süresi kadar sonra kontakları konum değiştirmekte yani ikinci byte lojik 1, üçüncü byte lojik 0 olmaktadır. Enerjisi kesilir kesilmez kontaklar ilk konumlarına dönerler. Yani ikinci byte lojik 0, üçüncü byte lojik 1 olur.

Compile sırasında lojik işlem satırları dörder karakter olarak kontrol edilerek gelen komuta göre assambler program karşılıkları önceden belirlenen adrese yerleştirilirler. Bu bölümün çalışmasını bir örnekle anlatmak daha kolay olacaktır.

$$Y01 = (X01 + Y01) \cdot \overline{X02}$$

şeklinde bir lojik ifadenin nasıl compile edileceğini ve compile edildikten sonra meydana gelen assambler programı gösterelim.

Bu ifadeyi programa girmek için ,

STR X01 OR Y01 AND NOT X02 OUT Y01

şekline dönüştürmemiz gerekmektedir. STR yeni bir satırın başladığını OUT sonucun Y01 'e aktarılacağını göstermektedir. Compile sırasında STR komutu görüldüğünde, bu satır ilk program satırı olduğundan #4000 adresine #3A, X01 girişi görüldüğünde #4001 ve #4002 adreslerine #5800+01 adresi, OR işlemi görüldüğünde #4006 adresine #B6, Y01 çıkışı görüldüğünde #4003 adresine #21, #4004 ve #4005 adreslerine #6000+01 AND işlemi görüldüğünde #400A adresine #A6, NOT komutu görüldüğünde bundan sonra gelecek giriş veya çıkış elemanının indisi bir arttırılır, X02 girişi görüldüğünde #4007 adresine #21, #4008 ve #4009 adreslerine #5800+03 ve OUT Y01 komutları görüldüğünde #400B adresine #32, #400C ve #400D adreslerine #6000+01 adresi yerleştirilir. Bu satırdan sonraki satırdaki SON komutuyla da #400E adresine #C9 konulur. Bu işlemler sonunda meydana gelen assambler program aşağıda verilmiştir.


```

4000 3A 01 58 LD A,(5800+01)
4003 21 01 60 LD HL,(6000+01)
4006 B6      OR HL
4007 21 03 58 LD HL,(5800+03)
400A A6      AND HL
400B 32 01 60 LD (6000+01),A
400E C9      RET

```

Oluşan bu assamler programdanda görüldüğü gibi kontakların konumları A ve HL registerlerine yerleştirilerek aralarında gerekli işlemler yapılmaktadır.

4.3.4.6 Tuş tanımlama bölümü

Programın bu bölümünde, yazılan lojik ifadenin içerisinde geçen girişler bilgisayar üzerindeki tuşlar olarak tanımlanmaktadır.Çıkışların tanımlanması bölümünde ekranda görmek istediğimiz çıkışlara herhangi bir string ifade girmemiz gerekmektedir.Eğer devrede zaman rölesi kullanılmışsa bu rölelerin gecikme süreleri yine bu bölümün sonunda Sn olarak girilmektedir.

4.3.4.7 Run etme bölümü

Bu bölümde önceden yazılmış ve compile edilmiş olan röleli lojik devre çalıştırılmakta ve belirlenen çıkışlar ekranda izlenebilmektedir.

Giriş olarak tanımlanan tuşlardan herhangi birine basıldığında o tuşun ait olduğu girişin ilk byte'ına lojik 1 seviyesi yüklenmekte, Assambler alt program çağırılarak gerekli şartlandırmalar yapılmaktadır.Daha sonra ana program çağırılarak işletilmektedir.

Zaman rölesinin gecikme işlemi ve çalışmasıda bu bölümde olmaktadır.Zaman rölesi enerjilendiğinde basic program içinde o rölenin gecikme süresi başlar ve bu sürenin sonunda bu rölenin kontakları olan bellek gözüne lojik 1 yerleştirilir.Daha sonra alt program ve anaprogram tekrar işletilir.Sonuçta tanımlanmış olan çıkışlar ekrandan izlenerek devrenin çalışması gerçekleşmiş olur.

4.3.5 Assambler alt programın incelenmesi

Aşağıda verilmiş olan bu program, giriş ve röle kontaktarı olarak kullanılan alanın şartlandırma işlemini yapmaktadır. Yani bir giriş butonuna basıldığında ilk byte in lojik 0 ikinci byte in lojik 1 olması, herhangi bir röle enerjilendiğinde açık olan kontağının kapanması, kapalı olanın açılması işlemi bu alt program tarafından yapılmaktadır. Bu sebeple alt program ana programdan önce ve sonra iki kere çağırılmaktadır. Bu yüzden giriş butonlarına basıldığında bu butona ait bellek gözü lojik 1 olmakta ana program çalıştıktan sonra çağırılan alt programla tekrar lojik 0 olmaktadır.

```

3900 06 INM+1      LD B, INM+1
3902 0E CIKM+1     LD C, CIKM+1
3904 16 ZRM+1     LD D, ZRM+1
3906 F5           PUSH AF
3907 C5           PUSH BC
3908 E5           PUSH HL
3909 21 08 58     LD HL, #5800
390C 36 00        jpl LD(HL), 0
390E 23          INC HL
390F 36 FF        LD(HL), #FF
3911 23          INC HL
3912 10 F8        DJNZ jpl
3914 41          LD B, C
3915 21 00 60     LD HL, #6000
3918 7E          jp2 LD A, (HL)
3919 2F          CPL
391A 23          INC HL
391B 77          LD(HL), A
391C 23          INC HL
391D 10 F9        DJNZ jp2
391F 42          LD B, D
3920 21 00 62     LD HL, #6200

```



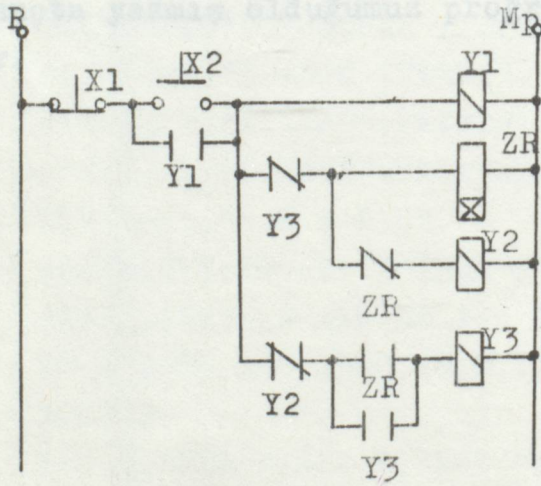
```

3923 7E      jp3  LD A,(HL)
3924 2F      CPL
3925 23      INC HL
3926 77      LD (HL),A
3927 23      INC HL
3928 23      INC HL
3929 10 F8   DJNZ jp3
392B F1      POP AF
392C C1      POP BC
392D E1      POP HL
392E C9      RET

```

4.4 PROĞRAMIN KULLANITIMASI

Proğramın kullanılmasını bir örnek ile açıklayalım. Bunun için Bölüm 3.5 de verilen bir motora yıldız-üçgen yol verme devresinin uygulamasını yapalım.



Şekil 4.5

Şekil 4.5 de verilen bu devrenin lojik ifadesi

$$Y1 = (X2 + Y1) \cdot \overline{X1}$$

$$ZR = (X2 + Y1) \cdot \overline{X1} \cdot \overline{Y3}$$

$$Y2 = Y1 \cdot \overline{Y3} \cdot \overline{ZR}$$

$$Y3 = (Y3 + ZR) \cdot Y1 \cdot \overline{Y2}$$

olarak çıkarılmıştı. Şimdi proğramı çalıştıralım ve çıkan

ilk menüden program ,daha sonra program yazma bölümünü seçelim.Burada bizden lojik ifadeler girmemiz beklenecektir. Buraya

```
STR X02 OR Y01 AND NOT X01 OUT Y01
STR X02 OR Y01 AND NOT X01 AND NOT Y03 OUT ZR1
STR Y01 AND NOT Y03 AND NOT ZR1 OUT Y02
STR Y03 OR ZR1 AND Y01 AND NOT Y02 OUT Y03
SON
```

Yazdığımızda bu devrenin lojik ifadesi programa girilmiş ve menüye dönmüş olur.İlk menüden Compile bölümünü seçerek Compile ettikten sonra devre çalışmaya hazır duruma gelmektedir.Tuş seçme işlemi ile giriş ve görmek istediğimiz çıkışlar belirlendikten sonra Run bölümü seçilerek devrenin çalışması ekran üzerinden izlenebilir.

Sonuçta yazmış olduğumuz programın tamamı aşağıda verilmiştir.


```

10  DIM XX$(99),YY$(99),X$(99),Y$(99),TM(99),LPR(100),
    PR$(100)
20  XX=84000:NB=85800:MB=86000:ZB=86200
30  MODE 1
40  LOCATE 13,24:PRINT"MENU DEN SECİN"
50  WINDOW#1,10,35,8,20
60  PRINT #1,"1.PROGRAM":PRINT#1
70  PRINT #1,"2.COMPILE":PRINT#1
80  PRINT #1,"3.TUS TANIMLAMA":PRINT#1
90  PRINT #1,"4.RUN"
100  A$=INKEY$:IF A$="" THEN 100
110  A=VAL(A$):IF A<1 OR A>4 THEN 100
120  ON A GOTO 130,450,1010,1200
130  MODE 1:LOCATE 13,24:PRINT"MENU DEN SECİN"
140  WINDOW#1,10,35,8,20
150  PRINT #1,"1.PROGRAM YAZMA":PRINT#1
160  PRINT #1,"2.PROGRAM LISTESI":PRINT#1
170  PRINT #1,"3.EDIT İŞLEMİ":PRINT#1
180  PRINT #1,"4.LOAD":PRINT#1
190  PRINT #1,"5.SAVE":PRINT#1
200  PRINT #1,"6.M E N U"
210  A$=INKEY$:IF A$="" THEN 210
220  A$=VAL(A$):IF A>6 OR A<1 THEN 210
230  ON A GOTO 240,1960,1910,1750,1590,20
240  MODE 2
250  KEY 0,"STR ":KEY 1,"NOT ":KEY 2,"AND ":KEY 3,"OR "
    :KEY 4,"OUT "
260  LOCATE 1,24:PRINT"f0:STR  f1:NOT  f2:AND  f3:OR
    f4:OUT  GİRİŞLER:Xnn  ÇIKIŞLAR:Ynn  ZAMAN RL:ZRn"
270  LOCATE 1,1
280  IF I=0 THEN 320
290  FOR X=0 TO I-1
300  PRINT X;"INCI SATIR ";PR$(X)
320  NEXT

```



```

320 PRINT I;"INCI SATIR";
330 INPUT PR$(I)
340 LPR(I)=LEN(PR$(I))
350 IF PR$(I)="SON" THEN 130
360 IF LEFT$(PR$(I),3)<>"STR" THEN PRINT"YANLIŞ BAŞLAN-
    GIC VAR":GOTO 320
370 BS=ABS(LPR(I),6)
380 IF MID$(PR$(I),BS,3)<>"OUT" THEN PRINT"YANLIŞ BİTİŞ
    VAR":GOTO 320
390 FOR X=1 TO LPR(I) STEP 4
400 KN$=MID$(PR$(I),X,4):KN1$=MID$(PR$(I),X,1):KN2$=MID$
    (PR$(I),X,2)
410 IF KNS<>"AND " AND KNS<>"OR " AND KNS<>"NOT " AND
    KNS<>"STR " AND KNS<>"OUT " AND KN1S<>"X" AND KN1S
    <>"Y" AND KN2S<>"ZR" THEN PRINT"HATALI KOMUT VAR":
    GOTO 320
420 NEXT
430 IF SR=1 THEN I=SI:SR=0:GOTO 130
440 I=I+1:GOTO 320
450 CLS:PRINT"COMPILE"
460 FOR X=0 TO I
470 FOR Y=1 TO LPR(X) STEP 4
480 A$=MID$(PR$(X),Y,3)
490 IF A$="OUT" THEN FL=1
500 IF A$="STR" THEN FLG=1
510 IF LEFT$(A$,1)<>"X" THEN 530
520 B$=MID$(A$,2,2):IN=VAL(B$):XX$(IN)=A$:A$="X":IF IN
    INM THEN INM=IN:GOTO 550
530 IF LEFT$(A$,1)<>"Y" THEN 550
540 BS=MID$(A$,2,2):CIK=VAL(B$):YY$(CIK)=A$:A$="Y":IF
    CIK CIKM THEN CIKM=CIK
550 IF LEFT$(A$-2)<>"ZR" THEN 570
560 B$=MID$(A$,3,1):ZR=VAL(B$):TM(ZR)=1:A$="ZR":IF ZR
    ZRM THEN ZRM=ZR

```



```

570 GOSUB 680
580 NEXT Y
590 FL=0
600 NEXT X
610 POKE &3900,&6:POKE &3901,&INM+1:POKE &3902,&E:POKE
    &3903,CIKM+1:POKE &3904,&16:POKE &3905,ZRM+1
620 FOR H=&3906 TO &392E
630 READ R:POKE H,R
640 NEXT
650 RESTORE
660 DATA &F5,&C5,&E5,&21,&0,&58,&36,&0,&23,&36,&FF,&23,
    &10,&F8,&41,&21,&0,&60,&7E,&2F,&23,&77,&23,&10,&F9,
    &42,&21,&0,&62,&7E,&2F,&23,&77,&23,&23,&10,&F8,&F1,&C1
    ,&E1,&C9
670 GOTO 20
680 IF A$="STR" THEN POKE XX,&3A:XX=XX+1:RETURN
690 IF FLG=0 THEN 790
700 IF A$("<")"X" THEN 730
710 NN=NB+2xIN:HB=(NN AND FFOO)/&100:LB=NN AND &FF
720 POKE XX,LB:XX=XX+1:POKE XX,HB:XX=XX+1:FLG=0:NB=&5800
    :MB=&6000:ZB=&6200:RETURN
730 IF A$("<")"Y" THEN 760
740 MM=MB+2xCIK:HB=(MM AND FFOO)/&100:LB=MM AND &FF
750 POKE XX,LB:XX=XX+1:POKE XX,HB:XX=XX+1:FLG=0:NB=&5800
    :MB=&6000:ZB=&6200:RETURN
760 IF A$("<")"ZR" THEN 790
770 ZZ=ZB+3xZR:HB=(ZZ AND FFOO)/&100:LB=ZZ AND &FF
780 POKE XX,LB:XX=XX+1:POKE XX,HB:XX=XX+1:FLG=0:NB=&5800
    :MB=&6000:ZB=&6200:RETURN
790 IF A$="NOT" THEN NB=NB+1:MB=MB+1:ZB=ZB+1:RETURN
800 IF A$="SON" THEN POKE XX,C9:XX=XX+1:GOTO 610
810 IF A$("<")"X" THEN 840
820 NN=NB+2xIN:HB=(NN AND FFOO)/&100:LB=NN AND &FF
830 POKE XX,&21:XX=XX+1:POKE XX,LB:XX=XX+1:POKE XX,HB:

```



```

      XX=XX+2:NB=&5800:MB=&6000:ZB=&6200:RETURN
840  IF FL=1 THEN 910
850  IF A$<>"Y" THEN 880
860  MM=MB+2*CIK:HB=(MM AND FFOO)/&100:LB=MM AND &FF
870  POKE XX,&21:XX=XX+1:POKE XX,LB:XX=XX+1:POKE XX,HB:
      XX=XX+2:NB=&5800:MB=&6000:ZB=&6200:RETURN
880  IF A$<>"ZR" THEN 910
890  ZZ=ZB+3*ZR:HB=(ZZ AND FFOO)/&100:LB=ZZ AND &FF
900  POKE XX,&21:XX=XX+1:POKE XX,LB:XX=XX+1:POKE XX,HB:
      XX=XX+2:NB=&5800:MB=&6000:ZB=&6200:RETURN
910  IF A$="AND" THEN XX=XX+3:POKE XX,&A6:XX=XX-3:RETURN
920  IF A$="OR" THEN XX=XX+3:POKE XX,&B6:XX=XX-3:RETURN
930  IF A$="OUT" THEN POKE XX,&32:XX=XX+1:RETURN
940  IF FL=0 THEN RETURN
950  IF A$<>"Y" THEN 980
960  MM=MB+2*CIK:HB=(MM AND &FFOO)/&100:LB=MM AND &FF
970  POKE XX,LB:XX=XX+1:POKE XX,HB:XX=XX+1:NB=&5800:MB=&
      6000:ZB=&6200:RETURN
980  IF A$<>"ZR" THEN RETURN
990  ZZ=ZB+3*ZR+2:HB=(ZZ AND &FFOO)/&100:LB=ZZ AND &FF
1000 POKE XX,LB:XX=XX+1:POKE XX,HB:XX=XX+2:NB=&5800:MB=
      &6000:ZB=&6200:RETURN
1010 MODE 2:PRINT"GIRISLERIN TANIMLANMASI"
1020 FOR N=1 TO INM
1030 IF XX$(N)="" THEN 1060
1040 PRINT XX$(N);"-";:INPUT X$(N)
1050 IF X$(N)="M" THEN PRINT" M:MENU (BASKA BIR TUS GIRI
      NIZ):GOTO 1040
1060 NEXT
1070 PRINT"CIKISLARIN TANIMLANMASI"
1080 FOR N=1 TO CIKM
1090 IF YY$(N)="" THEN 1120
1100 PRINT YY$(N);"-";:INPUT Y$(N)
1110 IF Y$(N)="M" THEN PRINT" M:MENU (BASKA BIR TUS GI-

```



```

RINIZ)":GOTO 1100
1120 NEXT
1130 PRINT"ZAMAN ROLELERININ SURELERINI VERINIZ (Sn)":
PRINT
1140 FOR N=0 TO ZRM
1150 IF TM(N)=0 THEN 1180
1160 PRINT N;"INCI ROLENIN SURESI";:INPUT T
1170 TM(N)=300xT
1180 NEXT
1190 GOTO 20
1200 MODE 2
1210 FOR X=1 TO 99:IF X$(X)="" THEN 1230
1220 PRINT" X";X;"-";X$(X);
1230 NEXT
1340 PRINT" M:MENU"
1250 C$=INKEY$:IF C$<>"" THEN 1340
1260 FOR X=0 TO ZRM:IF T(X)=0 THEN 1280
1270 IF TIME T(X)=TM(X) THEN 1300
1280 NEXT
1290 GOTO 1250
1300 T(X)=0:NN=&6200+3xX:POKE NN,&FF
1310 CALL &3900
1320 CALL &4000
1330 GOTO 1430
1340 IF C$="M" THEN 20
1350 CALL &3900
1360 FOR Q=1 TO INM
1370 IF X$(Q)=C$ THEN 1390
1380 NEXT
1390 NN=&5800+2xQ:POKE NN,&FF:POKE NN+1,0
1400 CALL &4000
1410 CALL &3900
1420 CALL &4000
1430 LOCATE 1,4

```



```

1440 FOR X=0 TO ZRM
1450 S=PEEK(&6200+3xX):IF S=&FF THEN 1480
1460 ZRL=PEEK(&6202+3xX)
1470 IF ZRL=&FF THEN T(X)=TIME
1480 NEXT X
1490 FOR X=0 TO ZRM:ZRL=PEEK(&6202+3xX)
1500 IF ZRL=0 THEN POKE (&6200+3xX),0:CALL &4000
1510 NEXT X
1520 FOR N=0 TO CIKM
1530 IF Y$(N)="" THEN 1560
1540 SN=PEEK(&6000+2xN):IF SN=255 THEN SN=1
1550 PRINT"Y";N;"-";SN
1560 NEXT N
1570 CALL &3900
1580 GOTO 1250
1590 CLS:ON ERROR GOTO 1710:PRINT"SAVE ICIN":INPUT"FILE
ISMINI VERINIZ";FL$
1600 IF LEN(FL$)>8 THEN 1590
1610 OPENOUT FL$
1620 WRITE #9,I,INM,CIKM,ZRM
1630 FOR X=0 TO I
1640 WRITE #9,PR$(X),LPR(X)
1650 NEXT X
1660 FOR X=0 TO 99
1670 WRITE #9,XX$(X),YY$(X),TM(X)
1680 NEXT X
1690 CLOSEOUT
1700 GOTO 130
1710 PRINT"HATALI FILE ISMI":PRINT
1720 LOCATE 10,23:PRINT"HERHANGI BIR TUSA BASIN"
1730 IF INKEY$="" THEN 1730
1740 GOTO 130
1750 CLS:PRINT"LOAD ICIN":INPUT"FILE ISMINI GIRINIZ";FL$
1760 ON ERROR GOTO 1870:IF LEN(FL$)>8 THEN 1750

```


5. İKİ YÜZDE TOPLANALI BİR ASANSÖRÜN BASIC PROGRAMI

```

1770 OPENIN FL$
1780 INPUT # 9,I,INM,CIKM,ZRM
1790 FOR X=0 TO I
1800 INPUT # 9,PR$(X),LPR(X)
1810 NEXT
1820 FOR X=0 TO 99
1830 INPUT # 9,XX$(X),YY$(X),TM(X)
1840 NEXT
1850 CLOSEIN
1860 GOTO 130
1870 PRINT"HATALI FILE ISMI":PRINT
1880 LOCATE 10,23:PRINT"HERHANGI BIR TUSA BASIN"
1890 IF INKEY$="" THEN 1890
1900 GOTO 130
1910 MODE 2:SR=1:SI=I:PRINT"KACINCI SATIR";:INPUT I
1920 PRINT PR$(I)
1930 PRINT"GEREKLI DEGISIKLIGI GIRIN"
1940 PRINT I;"INCI SATIR";:INPUT PR$(I)
1950 GOTO 340
1960 MODE 2
1970 FOR X=0 TO I:IF PR$(X)="" THEN 2000
1980 PRINT X;" ";PR$(X)
1990 IF INKEY$="" THEN 1990
2000 NEXT
2010 IF INKEY$="" THEN 2010
2020 GOTO 130

```

Programın akış şeması Şekil 5.1 de verilmiştir. Şemadan da görüldüğü gibi program üç ana bölüme ayrılmıştır.

5.2.3 Programdaki Değişken ve Dizielerin Anlamları

1. Kabinin bulunduğu katı bildiren sayı

2. Asansörün hareketinin başlangıcı

3. Kabinin girildiği kabin çekillerinin orijinal numarası

5. İKİ YÖNDE TOPLAMALI BİR ASANSÖRÜN BASIC PROGRAMI

5.1 GİRİŞ

Bu bölümde iki yönde toplamalı bir asansör, mikrobilgisayar üzerinde simülasyon şeklinde gerçekleştirilmiştir. Bu programı hazırlamakta amacımız, asansör gibi karmaşık röleli devreye sahip olan düzenlerin basic programlarla sağlanmasıdır. Programı anlatmadan evvel bu tip asansörün nasıl çalıştığını anlatalım.

İki yönde toplamalı asansörde, bir aşağı yönlü birde yukarı yönlü olmak üzere iki çağırma butonu bulunur. Asansör yukarı çıkarken yukarı yönlü çağrılara, aşağı inerken aşağı yönlü çağrılara cevap verir. Hareket yönüne ters olan komutları kaydederek gittiği yöndeki hareketi son bulduktan sonra yerine getirir. Kabin herhangi bir komut üzerine katta durduğunda, bir süre bekler ve bu sırada gelen komutları, geldiği ve gideceği yöne göre değerlendirerek yerine getirir.

Yapmış olduğumuz programda, programın çalışmasının daha iyi görülebilmesi için, kabin içi ve katlardaki çağırma butonları bilgisayarın üzerindeki tuşlar olarak, kabinin hareketini sağlayan motorun durumu ve kabinin konumu ekran üzerinde şekil ve ifadelerle gösterilmiştir. Programda ufak bir değişiklikle, giriş ve çıkışların portlardan yapılması ile mikrokompüter ve bu program gerçek bir asansöre kumanda edebilir.

5.2 PROGRAMIN İNCELENMESİ

5.2.1 Programın Akış Şeması

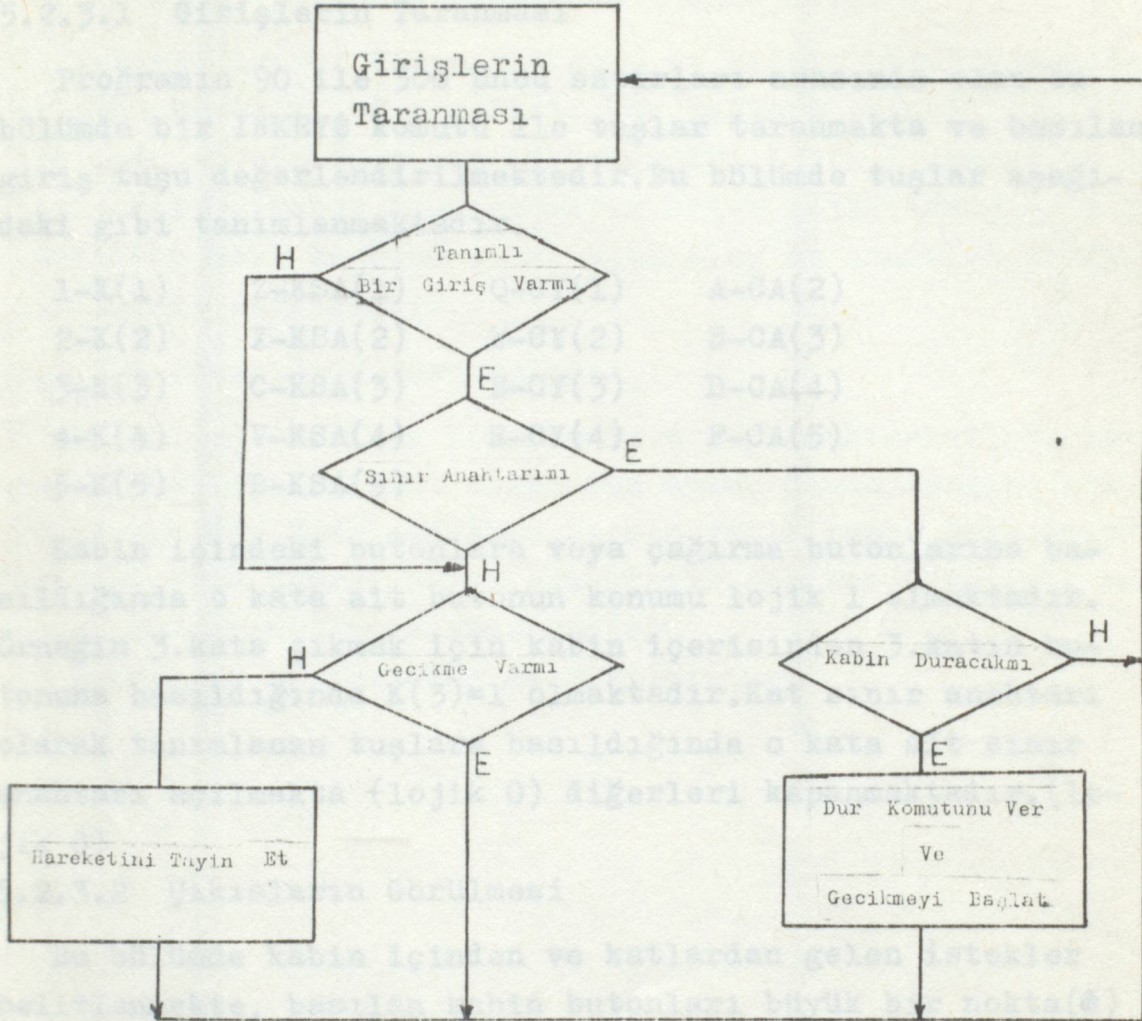
Programın akış şeması Şekil 5.1 de verilmiştir. Şemadan da görüldüğü gibi program üç ana bölümden oluşmuştur.

5.2.2 Programdaki Değişken ve Dizilerin Anlamları

N :Kabinin bulunduğu katı bildiren sayı

T :Gecikme süresinin başlangıcı

SN:Ekran çizdirilen kabin şekillerinin orijin numarası



Şekil 5.1

KA1,KA2,KA3,KA :Kabinin aşağı inip inmeyeceğini belirleyen sayılar.

KY1,KY2,KY3,KY :Kabinin yukarı çıkıp çıkmayacağını belirleyen sayılar.

K(N) :Kabin içindeki butonların konumları

CA(N) :Aşağı yöndeki çağırma komutları

CY(N) :Yukarı yöndeki çağırma komutları

KSA(N):Kat sınır anahtarlarının konumları

5.2.3 Programın Bölümlerinin İncelenmesi

5.2.3.1 Girişlerin Taranması

Programın 90 ile 300 üncü satırları arasında olan bu bölümde bir INKEY\$ komutu ile tuşlar taranmakta ve basılan giriş tuşu değerlendirilmektedir. Bu bölümde tuşlar aşağıdaki gibi tanımlanmaktadır.

1-K(1)	Z-KSA(1)	Q-CY(1)	A-CA(2)
2-K(2)	X-KSA(2)	W-CY(2)	S-CA(3)
3-K(3)	C-KSA(3)	E-CY(3)	D-CA(4)
4-K(4)	V-KSA(4)	R-CY(4)	F-CA(5)
5-K(5)	B-KSA(5)		

Kabin içindeki butonlara veya çağırma butonlarına basıldığında o kata ait butonun konumu lojik 1 olmaktadır. Örneğin 3.kata çıkmak için kabin içerisinden 3.katın butonuna basıldığında $K(3)=1$ olmaktadır. Kat sınır anahtarı olarak tanımlanan tuşlara basıldığında o kata ait sınır anahtarı açılmakta (lojik 0) diğerleri kapanmaktadır. (lojik 0)

5.2.3.2 Çıkışların Görülmesi

Bu bölümde kabin içinden ve katlardan gelen istekler belirlenmekte, basılan kabin butonları büyük bir nokta (●) yukarı çıkma istekleri yukarı yönde bir ok (↑), aşağı inme istekleri aşağı yönde bir ok (↓) işaretleriyle ekrana istekğin geldiği katın hizasına basılmaktadır.

5.2.3.3 Asansörün hareket yönünün tayini

Bu bölümde asansörün konumuna göre yapabileceği hareketlerin lojik ifadeleri bulunmaktadır. Asansörün hareketi için verilen komutlar bu lojik ifadelerde yerlerine konmaktadır. Sonuçta çıkan lojik konuma göre asansörün hareket yönü ve şekli belirlenmektedir. Her tuş tarama işleminden sonra bu bölüme gelinerek kabinin hareketi sürekli olarak izlenmektedir. Örnek olarak kabin içerisinden yukarı çıkma isteğinin kabul edilip edilmediği aşağıdaki lojik ifade ile

$KYL = KSA(2) \cdot KSA(3) \cdot KSA(4) \cdot KSA(5) \cdot K(2) + KSA(3) \cdot KSA(4) \cdot KSA(5) \cdot K(3) + KSA(4) \cdot KSA(5) \cdot K(4) + KSA(5) \cdot K(5)$

Kabinin ikinci katta olduğunu ve hiçbir komut almadığını düşünürsek $KYL=0$ olacaktır. Üst katlara çıkmak için kabin içinden bir tuşa basıldığında KYL lojik 1 olacak ve kabin yukarı çıkmaya başlayacaktır. Diğer lojik ifadelerin çıkarılmasındaki düşünce tarzı aynı olup bu ifadeler programın içinde verilmiştir.

5.2.3.4 Asansörün durdurulması

Bu bölümde verilen komutlarla sınır anahtarlarının konumu karşılaştırılarak asansörün durup durmayacağına karar verilmektedir. Eğer asansöre birden fazla komut girilmişse hareket yönü tekrar belirlenmektedir. Gecikme süresinin sonunda bu hareket uygulanmaktadır.

Programın tamamı aşağıda verilmiştir.

```

140 IF AS="4" THEN K(4)=-1:GOTO 300
150 IF AS="5" THEN K(5)=-1:GOTO 300
160 B=DATE 32,24
170 IF AS="2" THEN KSA(1)=0:KSA(2)=-1:KSA(3)=-1:KSA(4)=
-1:KSA(5)=-1:N=1:SN=8:PRINT" 2. K A T'TA":GOTO 730
180 IF AS="1" THEN KSA(1)=-1:KSA(2)=0:KSA(3)=-1:KSA(4)=
-1:KSA(5)=-1:N=2:SN=96:PRINT" 1. K A T'TA":GOTO 730
190 IF AS="0" THEN KSA(1)=-1:KSA(2)=-1:KSA(3)=0:KSA(4)=
-1:KSA(5)=-1:N=3:SN=165:PRINT" 2.2 A T'TA":GOTO 730
200 IF AS="V" THEN KSA(1)=-1:KSA(2)=-1:KSA(3)=-1:KSA(4)=
0:KSA(5)=-1:N=4:SN=244:PRINT" 3. K A T'TA":GOTO 730
210 IF AS="3" THEN KSA(1)=-1:KSA(2)=-1:KSA(3)=-1:KSA(4)=
-1:KSA(5)=0:N=5:SN=323:PRINT" 4. K A T'TA":GOTO 730
220 IF AS="6" THEN CY(1)=-1:GOTO 300
230 IF AS="7" THEN CY(2)=-1:GOTO 300
240 IF AS="8" THEN CY(3)=-1:GOTO 300
250 IF AS="9" THEN CY(4)=-1:GOTO 300
260 IF AS="A" THEN CY(2)=-1:GOTO 300
270 IF AS="B" THEN CY(3)=-1:GOTO 300

```



```

230 IF A$="D" THEN CA(4)=-1:GOTO 300
10 MODE 2
20 FOR X=2 TO 397 STEP 79
30 ORIGIN 130,X:DRAW 100,0,13:ORIGIN 130,X=2:DRAW 100,0,
  13
40 NEXT X
50 I=5:FOR Q=1 TO 16 STEP 5:I=I+1:LOCATE 8,Q:PRINT I;".
  KAT":NEXT Q
60 LOCATE 8,21:PRINT" Z. KAT"
70 LOCATE 32,23:PRINT" K A B I N"
80 DIM K(10)
90 A$=INKEY$
100 IF A$="K" THEN T=TIME
110 IF A$="1" THEN K(1)=-1:GOTO 300
120 IF A$="2" THEN K(2)=-1:GOTO 300
130 IF A$="3" THEN K(3)=-1:GOTO 300
140 IF A$="4" THEN K(4)=-1:GOTO 300
150 IF A$="5" THEN K(5)=-1:GOTO 300
160 LOCATE 32,24
170 IF A$="Z" THEN KSA(1)=0:KSA(2)=-1:KSA(3)=-1:KSA(4)=
  -1:KSA(5)=-1:N=1:SN=8:PRINT" Z. K A T'TA":GOTO 730
180 IF A$="X" THEN KSA(1)=-1:KSA(2)=0:KSA(3)=-1:KSA(4)=
  -1:KSA(5)=-1:N=2:SN=86:PRINT" 1.K A T'TA":GOTO 730
190 IF A$="C" THEN KSA(1)=-1:KSA(2)=-1:KSA(3)=0:KSA(4)=
  -1:KSA(5)=-1:N=3:SN=165:PRINT" 2.K A T'TA":GOTO 730
200 IF A$="V" THEN KSA(1)=-1:KSA(2)=-1:KSA(3)=-1:KSA(4)=
  0:KSA(5)=-1:N=4:SN=244:PRINT" 3.K A T'TA":GOTO 730
210 IF A$="B" THEN KSA(1)=-1:KSA(2)=-1:KSA(3)=-1:KSA(4)=
  -1:KSA(5)=0:N=5:SN=323:PRINT" 4.K A T'TA":GOTO 730
220 IF A$="Q" THEN CY(1)=-1:GOTO 300
230 IF A$="W" THEN CY(2)=-1:GOTO 300
240 IF A$="E" THEN CY(3)=-1:GOTO 300
250 IF A$="R" THEN CY(4)=-1:GOTO 300
260 IF A$="A" THEN CA(2)=-1:GOTO 300
270 IF A$="S" THEN CA(3)=-1:GOTO 300

```



```

230 IFA$="D" THEN CA(4)=-1:GOTO 300
290 IF A$="F" THEN CA(5)=-1:GOTO 300
300 REM PRINT
310 FOR X=1 TO 5:READ P:IF K(X)=0 THEN LOCATE 29,P:PRINT
    " ":GOTO 330
320 LOCATE 29,P:PRINT CHR$(231)
330 NEXT
340 DATA 23,18,13,8,3
350 FOR X=1 TO 4:READ M:IF CY(X)=0 THEN LOCATE 29,M:PRINT
    " ":GOTO 370
360 LOCATE 29,M:PRINT CHR$(240)
370 NEXT
380 DATA 22,17,12,7
390 FOR X=2 TO 5:READ H:IF CA(X)=0 THEN LOCATE 29,H:PRINT
    " ":GOTO 410
400 LOCATE 29,H:PRINT CHR$(241)
410 NEXT
420 RESTORE
430 DATA 19,14,9,4
440 IF T=900 TIME THEN 90
450 IF KA=0 AND KY=0 AND CA(N)=-1 THEN CA(N)=0
460 IF KA=0 AND KY=0 AND CY(N)=-1 THEN CY(N)=0
470 IF KA=0 AND KY=0 AND K(N)=-1 THEN K(N)=0
480 KY1=KSA(2) AND KSA(3) AND KSA(4) AND KSA(5) AND K(2)
    OR KSA(3) AND KSA(4) AND KSA(5) AND K(3) OR KSA(4)
    AND KSA(5) AND K(4) OR KSA(5) AND K(5)
490 KY2=KSA(2) AND KSA(3) AND KSA(4) AND KSA(5) AND CY(2)
    OR KSA(3) AND KSA(4) AND KSA(5) AND CY(3) OR KSA(4)
    AND KSA(5) AND CY(4)
500 KY3=(KSA(2) AND KSA(3) AND KSA(4) AND KSA(5) AND CA(2)
    OR KSA(3) AND KSA(4) AND KSA(5) AND CA(3) OR KSA(4)
    AND KSA(5) AND CA(4) OR KSA(5) AND CA(5)) AND NOT(KY1
    OR KY2)
510 KA1=KSA(1) AND K(1) OR KSA(1) AND KSA(2) AND K(2) OR
    KSA(1) AND KSA(2) AND KSA(3) AND K(3) OR KSA(1) AND

```



```

KSA(2) AND KSA(3) AND KSA(4) AND K(4)
520 KA2=KSA(1) AND KSA(2) AND CA(2) OR KSA(1) AND KSA(2)
AND KSA(3) AND CA(3) OR KSA(1) AND KSA(2) AND KSA(3)
AND KSA(4) AND CA(4)
530 KA3=(KSA(1) AND CY(1) OR KSA(1) AND KSA(2) AND CY(2)
OR KSA(1) AND KSA(2) AND KSA(3) AND CY(3) OR KSA(1)
AND KSA(2) AND KSA(3) AND KSA(4) AND CY(4)) AND NOT
(KA1 OR KA2)
540 KY=(KY1 OR KY2 OR KY3) AND NOT KA
550 KA=(KA1 OR KA2 OR KA3) AND NOT KY
560 IF KY<>0 THEN LOCATE 35,1:PRINT"YUKARI CIKIYOR":GOTO
90
570 IF KA<>0 THEN LOCATE 35,1:PRINT"ASAGI INIYOR ":GOTO
90
580 GOTO 90
590 IF T=900 TIME THEN 90
600 IF KA<>0 THEN 670
610 IF KSA(N)<>(K(N) OR CY(N)) THEN K(N)=0:CY(N)=0:LOCATE
35,1:PRINT"ASANSOR STOP "T=TIME
620 IF CA(N)=0 THEN 90
630 KY1=KSA(2) AND KSA(3) AND KSA(4) AND KSA(5) AND K(2)
OR KSA(3) AND KSA(4) AND KSA(5) AND K(3) OR KSA(4)
AND KSA(5) AND K(4) OR KSA(5) AND K(5)
640 KY2=KSA(2) AND KSA(3) AND KSA(4) AND KSA(5) AND CY(2)
OR KSA(3) AND KSA(4) AND KSA(5) AND CY(3) OR KSA(4)
AND KSA(5) AND CY(4)
650 CA(N)=0:KY3=(KSA(2) AND KSA(3) AND KSA(4) AND KSA(5)
AND CA(2) OR KSA(3) AND KSA(4) AND KSA(5) AND CA(3)
OR KSA(4) AND KSA(5) AND CA(4) OR KSA(5) AND CA(5))
AND NOT(KY1 OR KY2)
660 IF (KY1 OR KY2 OR KY3)=0 THEN LOCATE 35,1:PRINT"ASAN
SOR STOP ":T=TIME:GOTO 90 ELSE CA(N)=-1:GOTO 90
670 IF KSA(N)<>(K(N) OR CA(N)) THEN K(N)=0:CA(N)=0:LOCATE
35,1:T=TIME:PRINT"ASANSOR STOP "
680 IF CY(N)=0 THEN 90

```



```

690 KA1=KSA(1) AND K(1) OR KSA(1) AND KSA(2) AND K(2) OR
    KSA(1) AND KSA(2) AND KSA(3) AND K(3) OR KSA(1) AND
    KSA(2) AND KSA(3) AND KSA(4) AND K(4)
700 KA2=KSA(1) AND KSA(2) AND CA(2) OR KSA(1) AND KSA(2)
    AND KSA(3) AND CA(3) OR KSA(1) AND KSA(2) AND KSA(3)
    AND KSA(4) AND CA(4)
710 CY(N)=0:KA3=(KSA(1) AND CY(1) OR KSA(1) AND KSA(2)
    AND CY(2) OR KSA(1) AND KSA(2) AND KSA(3) AND KSA(4)
    AND CY(4)) AND NOT(KA1 OR KA2)
720 IF (KA1 OR KA2 OR KA3)=0 THEN LOCATE 35,1:PRINT"ASAN
    SOR STOP ":T=TIME:GOTO 90 ELSE CY(N)=-1:GOTO 90 .
730 ORIGIN 150,SX:DRAW 0,65,0:DRAW 50,65,0:DRAW 50,0,0:
    DRAW 0,0,0:ORIGIN 170,SX+10:DRAW 0,40,0:DRAW 10,40,0
    :DRAW 10,0,0:DRAW 0,0,0
740 ORIGIN 150,SN:DRAW 0,65,9:DRAW 50,65,9:DRAW 50,0,9:
    DRAW 0,0,9:ORIGIN 170,SN+10:DRAW 0,40,9:DRAW 10,40,9
    DRAW 10,0,9:DRAW 0,0,9
750 SX=SN
760 GOTO 590

```


6. DUBLEKS ASANSÖRÜN BASIC PROGRAMI

6.1 GİRİŞ

Bu son bölümde daha karmaşık sisteme sahip olan dubleks asansörlerin Basic programla gerçekleşmesi verilmektedir. Yine programı anlatmadan önce dubleks asansörlerin çalışma prensiplerini verelim.

Bu tip asansörlerde iki adet toplama kumandalı asansör beraber çalışırlar. Her kabinde her kat için birer kat butonu bulunur. Her katta ise yukarı ve aşağı yönler için birer adet çağırma butonu bulunur. Kabinlerdeki kat butonları yalnız bulunduğu kabine kumanda ederler. Katlardaki çağırma butonları ise her iki kabinin kumandası için ortak olarak kullanılırlar. Örnek olarak 1. ciktan kalkan kabinlerden birisi 7. diğeri 8. kata çıkarken 4. kattan yukarı çıkmak için bir komut geldiğinde kabinlerden birisi 4. katta durur. Buradaki kişi bindikten sonra yukarı doğru hareket eder. Bu katta duran kabin bu komutu siler. Diğer kabin bu katta durmadan yoluna devam eder.

Yazmış olduğumuz programda birinci asansör ikinci asansöre göre daha öncelikli olarak düşünülmüştür. Örneğin iki kabinde dururken gelen herhangi bir komutu ilk asansör cevaplamaktadır. Bunun nedenide bu tip asansörlerin tam bir lojik ifadesi çıkarılamamaktadır. Çünkü gelen istekler belli bir kurala göre değil rastgele olmaktadır. Her iki asansörün konumu, hareket yönleri, gelen isteklerin kabinlere göre konumu ve yönleri rastgele olduğundan bu asansörler belli bir kurala göre değil mantığa en uygun olan şekilde çalışmaktadırlar.

6.2 PROGRAMIN İNCELENMESİ

6.2.1 Programın akış şeması

Programın akış şeması Şekil 6.1 de verilmiştir.

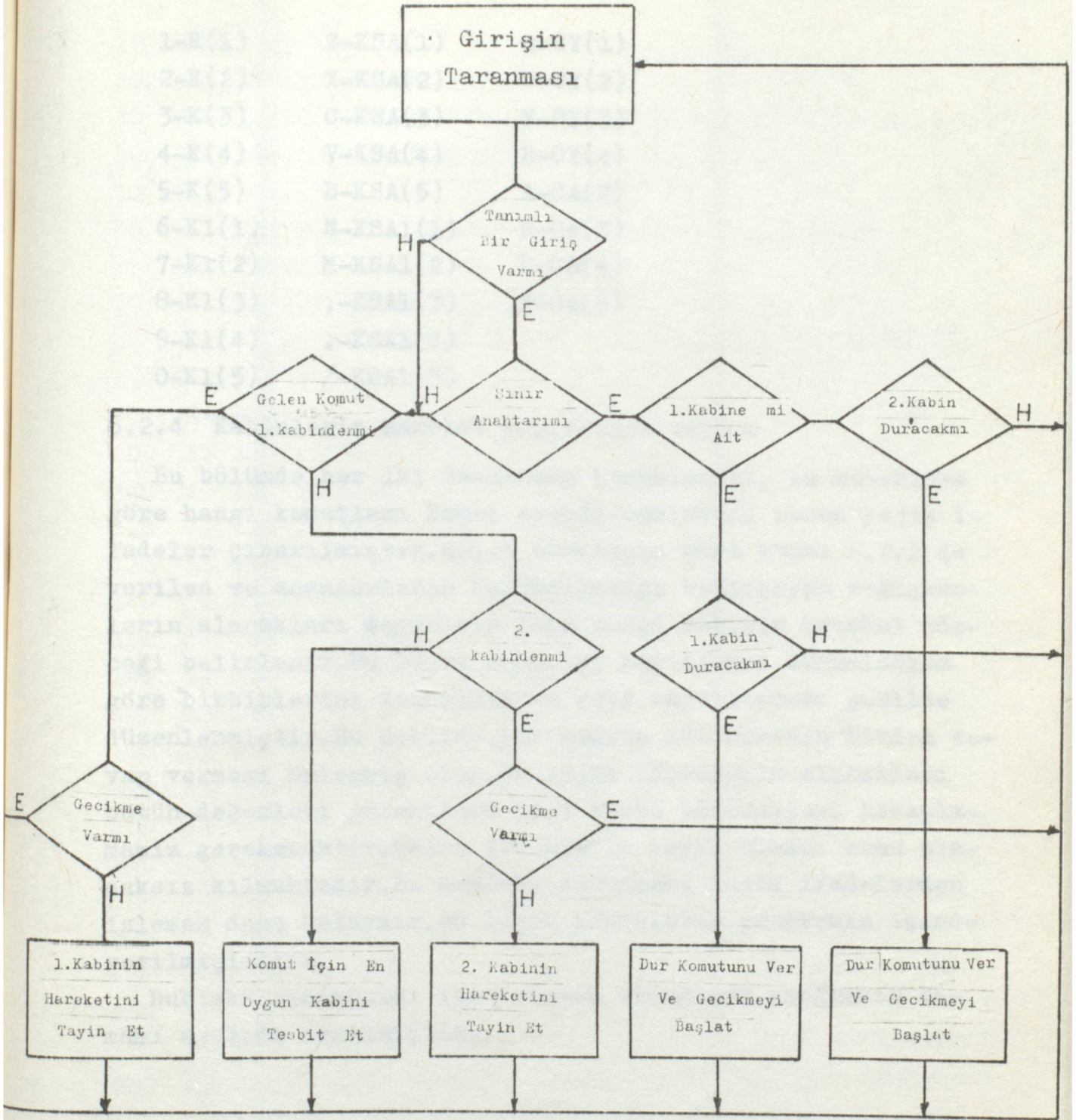
6.2.2 Programdaki değişken ve dizilerin anlamları

N : 1. kabinin yerini gösteren sayı

N :2.Kabinin yeri
 SN :1.Kabin şeklinin orijin numarası
 DN :2. " " " "
 KY11 :1.Kabinden yukarı çıkma komutunun olup olmadığı
 KY12 :1.Kabinin yukarısından yukarıya çıkma isteğinin olup olmadığı
 KY13 :1.Kabinin yukarısından aşağı inme isteğinin olup olmadığı
 KA11 :1.Kabinden aşağı inme komutunun olup olmadığı
 KA12 :1.Kabinin aşağısından aşağıya inme isteğinin olup olmadığı
 KA13 :1.Kabinin aşağısından yukarı çıkma isteğinin olup olmadığı
 KY21 :2.Kabinden yukarı çıkma komutunun olup olmadığı
 KY22 :2.Kabinin yukarısından yukarıya çıkma isteğinin olup olmadığı
 KY23 :2.Kabinin yukarısından aşağı inme isteğinin olup olmadığı
 KA21 :2.Kabinden aşağı inme komutunun olup olmadığı
 KA22 :2.Kabinin aşağısından aşağıya inme isteğinin olup olmadığı
 KA23 :2.Kabinin aşağısından yukarı çıkma isteğinin olup olmadığı
 K(N),K1(N) :1 ve 2.Kabin içindeki butonların konumları
 CA(N) :Aşağı yönde çağırma komutları
 CY(N) :Yukarı " " "
 KSA(N),KSA1(N) :1. ve 2. asansörlerin katsınır anahtarlarının konumları

6.2.3 Girişlerin taranması bölümü

Bu bölümde önceki programda olduğu gibi giriş durumlarının belirlenmesi için tuşlar taranmaktadır. Basılan tuşlar değerlendirilerek işlem bölümüne gidilmektedir. Burada tuşlar aşağıdaki gibi tanımlanmıştır.



Şekil 6.1

1-K(1)	Z-KSA(1)	Q-CY(1)
2-K(2)	X-KSA(2)	W-CY(2)
3-K(3)	C-KSA(3)	E-CY(3)
4-K(4)	V-KSA(4)	R-CY(4)
5-K(5)	B-KSA(5)	A-CA(2)
6-K1(1)	N-KSA1(1)	S-CA(3)
7-K1(2)	M-KSA1(2)	D-CA(4)
8-K1(3)	, -KSA1(3)	F-CA(5)
9-K1(4)	. -KSA1(4)	
0-K1(5)	/ -KSA1(5)	

6.2.4 Kabinlerin hareket yönlerinin tayini

Bu bölümde her iki asansörün konumlarını, bu konumlara göre hangi komutları kabul edebileceklerini veren lojik ifadeler çıkarılmıştır. Gelen komutlara göre bölüm 6.2.2 de verilen ve asansörlerin hareketlerini belirleyen değişkenlerin alacakları değerlere göre hangi kabinin hareket edeceği belirlenir. Bu lojik konumlar kabinlerin durumlarına göre birbirlerini tamamlayacak veya engelleyecek şekilde düzenlenmiştir. Bu şekilde bir komuta iki kabinin birden cevap vermesi önlenmiş olur. Bu lojik ifadelerin alacakları bütün değerleri görebilmek için bütün ihtimalleri hesaplamamız gerekecektir. Gelen isteklerin keyfi olması bunu olanaksız kılmaktadır. Bu nedenle durumları lojik ifadelerden izlemek daha kolaydır. Bu lojik ifadelerde programın içinde verilmişlerdir.

Dubleks asansörler için yazmış olduğumuz programın tamamı aşağıda verilmiştir.


```

10  MODE 2
20  FOR X=2 TO 397 STEP 79
30  ORIGIN 80,X: DRAW 370,0,13: ORIGIN 80,X+2: DRAW 100,0,13
40  ORIGIN 350,X+2: DRAW 100,0,13
50  NEXT
60  I=5: FOR Q=1 TO 16 STEP 5: I=I+1: LOCATE 2,Q: PRINT I; ".
    KAT": NEXT
70  LOCATE 2,21: PRINT " Z.KAT"
80  LOCATE 25,23: PRINT "1.K A B I N": LOCATE 59,23: PRINT "2.
    K A B I N"
90  A$=INKEY$
100 IF A$="1" THEN K(1)=-1: GOTO 400
110 IF A$="2" THEN K(2)=-1: GOTO 400
120 IF A$="3" THEN K(3)=-1: GOTO 400
130 IF A$="4" THEN K(4)=-1: GOTO 400
140 IF A$="5" THEN K(5)=-1: GOTO 400
150 IF A$="6" THEN K1(1)=-1: GOTO 400
160 IF A$="7" THEN K1(2)=-1: GOTO 400
170 IF A$="8" THEN K1(3)=-1: GOTO 400
180 IF A$="9" THEN K1(4)=-1: GOTO 400
190 IF A$="0" THEN K1(5)=-1: GOTO 400
200 LOCATE 25,24
210 IF A$="Z" THEN KSA(1)=0: KSA(2)=-1: KSA(3)=-1: KSA(4)=
    -1: KSA(5)=-1: N=1: SN=7: PRINT " Z.K A T 'TA": GOTO 1180
220 IF A$="X" THEN KSA(1)=-1: KSA(2)=0: KSA(3)=-1: KSA(4)=
    -1: KSA(5)=-1: N=2: SN=86: PRINT "1.K A T 'TA": GOTO 1180
230 IF A$="C" THEN KSA(1)=-1: KSA(2)=-1: KSA(3)=0: KSA(4)=
    -1: KSA(5)=-1: N=3: SN=165: PRINT "2.K A T 'TA": GOTO 1180
240 IF A$="V" THEN KSA(1)=-1: KSA(2)=-1: KSA(3)=-1: KSA(4)=
    0: KSA(5)=-1: N=4: SN=244: PRINT "3.K A T 'TA": GOTO 1180
250 IF A$="B" THEN KSA(1)=-1: KSA(2)=-1: KSA(3)=-1: KSA(4)=
    -1: KSA(5)=0: N=5: SN=323: PRINT "4.K A T 'TA": GOTO 1180
260 LOCATE 59,24
270 IF A$="N" THEN KSA1(1)=0: KSA1(2)=-1: KSA1(3)=-1: KSA1(
    4)=-1: KSA1(5)=-1: M=1: DN=7: PRINT " Z.K A T 'TA: GOTO 1220

```



```

280 IF A$="M" THEN KSA1(1)=-1:KSA1(2)=0:KSA1(3)=-1:KSA1(4)
    =-1:KSA1(5)=-1:M=2:DN=86:PRINT " 1.K A T 'TA":GOTO
    1220
290 IF A$=", " THEN KSA1(1)=-1:KSA1(2)=-1:KSA1(3)=0:KSA1(4)
    =-1:KSA1(5)=-1:M=3:DN=165:PRINT " 2. K A T 'TA" :GOTO
    1220
300 IF A$="." THEN KSA1(1)=-1:KSA1(2)=-1:KSA1(3)=-1:KSA1
    (4)=0:KSA1(5)=-1:M=4:DN=244:PRINT " 3. K A T 'TA":
    GOTO 1220
310 IF A$="/" THEN KSA1(1)=-1:KSA1(2)=-1:KSA1(3)=-1:KSA1
    (4)=-1:KSA1(5)=0:M=5:DN=323:PRINT "4. K A T 'TA ":
    GOTO 1220
320 IF A$="Q" THEN CY(1)=-1:GOTO 400
330 IF A$="W" THEN CY(2)=-1:GOTO 400
340 IF A$="E" THEN CY(3)=-1:GOTO 400
350 IF A$="R" THEN CY(4)=-1:GOTO 400
360 IF A$="A" THEN CA(2)=-1:GOTO 400
370 IF A$="S" THEN CA(3)=-1:GOTO 400
380 IF A$="D" THEN CA(4)=-1:GOTO 400
390 IF A$="F" THEN CA(5)=-1:GOTO 400
400 FOR X=1 TO 5:READ P:IF K(X)=0 THEN LOCATE 22,P:PRINT
    " ":GOTO 420
410 LOCATE 22,P:PRINT CHR$(231)
420 NEXT
430 DATA 23,18,13,8,3
440 FOR X=1 TO 5:READ P:IF K1(X)=0 THEN LOCATE 56,P:PRINT
    " ":GOTO 460
450 LOCATE 56,P:PRINT CHR$(231)
460 NEXT
470 DATA 23,18,13,8,3
480 FOR X=1 TO 4:READ XM:IF CY(X)=0 THEN LOCATE 22,XM:
    PRINT " ":LOCATE 56,XM:PRINT " ":GOTO 500
490 LOCATE 22,XM:PRINT CHR$(240):LOCATE 56,XM:PRINT CHR$
    (240)
500 NEXT

```



```

510 DATA 22,17,12,7
520 FOR X=2 TO 5:READ H:IF CA(X)=0 THEN LOCATE 22,H:
    PRINT "  ":LOCATE 56,H:PRINT "  ":GOTO 540
530 LOCATE 22,H:PRINT CHR$(241):LOCATE 56,H:PRINT CHR$(
    241)
540 NEXT
550 DATA 19,14,9,4
560 RESTORE
570 IF KA=0 AND KY=0 AND KA1=0 AND KY1=0 AND (CA(N)=-1
    OR CA(M)=-1) THEN CA(N)=0:CA(M)=0
580 IF KA=0 AND KY=0 AND KY1=0 AND KA1=0 AND (CY(N)=-1
    OR CY(M)=-1) THEN CY(N)=0:CY(M)=0
590 IF KA=0 AND KY=0 AND K(N)=-1 THEN K(N)=0
600 IF KA1=0 AND KY1=0 AND K1(M)=-1 THEN K1(M)=0
610 IF T1=1200 TIME THEN 740
620 Y12=KSA(2) AND KSA(3) AND KSA(4) AND KSA(5):Y13=KSA(3)
    AND KSA(4) AND KSA(5):Y14=KSA(4) AND KSA(5):Y15=KSA(5)
630 A11=KSA(1):A12=KSA(1) AND KSA(2):A13=KSA(1) AND KSA(2)
    AND KSA(3):A14=KSA(1) AND KSA(2) AND KSA(3) AND KSA(4)
640 KY11=(Y12 AND K(2) OR Y13 AND K(3) OR Y14 AND K(4) OR
    Y15 AND K(5)) AND NOT(KA12 OR KA13 OR KA)
650 KY12=(Y12 AND CY(2) OR Y13 AND CY(3) OR Y14 AND CY(4))
    AND NOT(KY22 OR KY13 OR KA23 OR KA)
660 KY13=(Y12 AND CA(2) OR Y13 AND CA(3) OR Y14 AND CA(4)
    OR Y15 AND CA(5)) AND NOT(KY11 OR KY12 OR KY23 OR KA22
    OR KA)
670 KA11=(A11 AND K(1) OR A12 AND K(2) OR A13 AND K(3) OR
    A14 AND K(4)) AND NOT(KY12 OR KY13 OR KY)
680 KA12=(A12 AND CA(2) OR A13 AND CA(3) OR A14 AND CA(4))
    AND NOT(KA22 OR KY23 OR KA13 OR KY)
690 KA13=(A11 AND CY(1) OR A12 AND CY(2) OR A13 AND CY(3)
    OR A14 AND CY(4)) AND NOT(KA11 OR KA12 OR KA23 OR
    KY22 OR KY)
700 KY=(KY11 OR KY12 OR KY13) AND NOT KA
710 KA=(KA11 OR KA12 OR KA13) AND NOT KY

```



```

720 IF KY<>0 THEN LOCATE 24,3:PRINT"1.ASANSOR":LOCATE 24,
330 4:PRINT"YUKARI ":LOCATE 24,5:PRINT"CIKIYOR ":GOTO
740
730 IF KA<>0 THEN LOCATE 24,3:PRINT"1.ASANSOR":LOCATE 24,
340 4:PRINT"ASAGI ":LOCATE 24,5:PRINT"INIYOR "
740 IF T2=1200 TIME THEN 90
750 Y22=KSA1(2) AND KSA1(3) AND KSA1(4) AND KSA1(5):Y23=
350 KSA1(3) AND KSA1(4) AND KSA1(5):Y24=KSA1(4) AND KSA1
360 (5):Y25=KSA1(5)
760 A21=KSA1(1):A22=KSA1(1) AND KSA1(2):A23=KSA1(1) AND
370 KSA1(2) AND KSA1(3):A24=KSA1(1) AND KSA1(2) AND KSA1
380 (3) AND KSA1(4)
770 KY21=(Y22 AND K1(2) OR Y23 AND K1(3) OR Y24 AND K1(4)
OR Y25 AND K1(5)) AND NOT(KA22 OR KA23 OR KA1)
780 KY22=(Y22 AND CY(2) OR Y23 AND CY(3) OR Y24 AND CY(4))
AND NOT(KY12 OR KY23 OR KA13 OR KA1)
790 KY23=(Y22 AND CA(2) OR Y23 AND CA(3) OR Y24 AND CA(4)
OR Y25 AND CA(5)) AND NOT(KY21 OR KY22 OR KY13 OR
390 KA12 OR KA1)
800 KA21=(A21 AND K1(1) OR A22 AND K1(2) OR A23 AND K1(3)
OR A24 AND K1(4)) AND NOT(KY22 OR KY23 OR KY1)
810 KA22=(A22 AND CA(2) OR A23 AND CA(3) OR A24 AND CA(4))
AND NOT(KA12 OR KY13 OR KA23 OR KY1)
820 KA23=(A21 AND CY(1) OR A22 AND CY(2) OR A23 AND CY(3)
OR A24 AND CY(4)) AND NOT(KA21 OR KA22 OR KY12 OR
400 KY1)
830 KY1=(KY21 OR KY22 OR KY23) AND NOT KA1
840 KA1=(KA21 OR KA22 OR KA23) AND NOT KY1
850 IF KY1 0 THEN LOCATE 58,3:PRINT"2. ASANSOR":LOCATE
58,4:PRINT"YUKARI ":LOCATE 58,5:PRINT"CIKIYOR ":
GOTO 90
860 IF KA1 0 THEN LOCATE 58,3:PRINT"2.ASANSOR":LOCATE
58,4:PRINT"ASAGI ":LOCATE 58,5:PRINT"INIYOR ":
GOTO 90
870 GOTO 90

```



```

880 IF KA<>0 THEN 960
890 IF KSA(N)<>K(N) THEN K(N)=0:T1=TIME:LOCATE 24,3:PRINT
    "1.ASANSOR":LOCATE 24,4:PRINT"STOP      ":LOCATE 24,5:
    PRINT"      "
900 IF KSA(N)<>(CY(N) AND KY12) THEN CY(N)=0:T1=TIME:LO
    CATE 24,3:PRINT"1.ASANSOR":LOCATE 24,4:PRINT"STOP      "
    :LOCATE 24,5:PRINT"      "
910 IF CA(N)=0 THEN 90
920 Y12=KSA(2) AND KSA(3) AND KSA(4) AND KSA(5):Y13=KSA(3)
    AND KSA(4) AND KSA(5):Y14=KSA(4) AND KSA(5):Y15=KSA(5)
930 KY11=(Y12 AND K(2) OR Y13 AND K(3) OR Y14 AND K(4) OR
    Y15 AND K(5)) AND NOT(KA12 OR KA13 OR KA)
940 KY12=(Y12 AND CY(2) OR Y13 AND CY(3) OR Y14 AND CY(4))
    AND NOT(KY22 OR KY13 OR KA23 OR KA)
950 IF (KY11 OR KY12 OR KY23 OR KA22)=0 THEN LOCATE 24,3:
    PRINT"1.ASANSOR":LOCATE 24,4:PRINT"STOP      ":LOCATE
    24,5:PRINT"      ":CA(N)=0:T1=TIME:KA=-1:GOTO 90
    ELSE GOTO 90
960 IF KSA(N)<>K(N) THEN K(N)=0:T1=TIME:LOCATE 24,3:PRINT
    "1.ASANSOR":LOCATE 24,4:PRINT"STOP      ":LOCATE 24,5:
    PRINT"      "
970 IF KSA(N)<>(CA(N) AND KA12) THEN CA(N)=0:T1=TIME:LOCA
    TE 24,3:PRINT"1.ASANSOR":LOCATE 24,4:PRINT"STOP      ":
    LOCATE 24,5:PRINT"      "
980 IF CY(N)=0 THEN 90
990 A11=KSA(1):A12=KSA(1) AND KSA(2):A13=KSA(1) AND KSA(2)
    AND KSA(3):A14=KSA(1) AND KSA(2) AND KSA(3) AND KSA(4)
1000 KA11=(A11 AND K(1) OR A12 AND K(2) OR A13 AND K(3)
    OR A14 AND K(4)) AND NOT(KY12 OR KY13 OR KY)
1010 KA12=(A12 AND CA(2) OR A13 AND CA(3) OR A14 AND CA(4))
    AND NOT(KA22 OR KY23 OR KA13 OR KY)
1020 IF (KA11 OR KA12 OR KA23 OR KY22)=0 THEN LOCATE 24,3
    :PRINT"1.ASANSOR":LOCATE 24,4:PRINT"STOP      ":LOCATE
    24,5:PRINT"      ":CY(N)=0:T1=TIME:KY=-1:GOTO 90
    ELSE GOTO 90

```



```

1030 IF KA1<>0 THEN 1110
1040 IF KSA1(M)<>K1(M) THEN K1(M)=0:T2=TIME:LOCATE 58,3:
PRINT"2.ASANSOR":LOCATE 58,4:PRINT"STOP ":LOCATE
58,5:PRINT"
1050 IF KSA1(M)<>(CY(M) AND KY22) THEN CY(M)=0:T2=TIME:
LOCATE 58,3:PRINT"2.ASANSOR":LOCATE 58,4:PRINT"STOP
":LOCATE 58,5:PRINT"
1060 IF CA(M)=0 THEN 90
1070 Y22=KSA1(2) AND KSA1(3) KSA1(4) AND KSA1(5):Y23=KS
A1(3) AND KSA1(4) AND KSA1(5):Y24=KSA1(4) AND KSA1(
5):Y25=KSA1(5)
1080 KY21=(Y22 AND K1(2) OR Y23 AND K1(3) OR Y24 AND K1(
4) OR Y25 AND K1(5)) AND NOT(KA22 OR KA23 OR KA1)
1090 KY22=(Y22 AND CY(2) OR Y23 AND CY(3) OR Y24 AND CY(
4)) AND NOT(KY12 OR KY23 OR KA13 OR KA1)
1100 IF (KY21 OR KY22 OR KY13 OR KA12)=0 THEN T2=TIME:
LOCATE 58,3:PRINT"2.ASANSOR":LOCATE 58,4:PRINT"STOP
":LOCATE 58,5:PRINT" ":CA(M)=0:KA=-1:GOTO
90 ELSE GOTO 90
1110 IF KSA1(M)<>K1(M) THEN K1(M)=0:T2=TIME:LOCATE 58,3:
PRINT"2.ASANSOR":LOCATE 58,4:PRINT"STOP ":LOCATE
58,5:PRINT"
1120 IF KSA1(M)<>(CA(M) AND KA22) THEN CA(M)=0:T2=TIME:
LOCATE 58,3:PRINT"2.ASANSOR":LOCATE 58,4:PRINT"STOP
":LOCATE 58,5:PRINT"
1130 IF CY(M)=0 THEN 90
1140 A21=KSA1(1):A22=KSA1(1) AND KSA1(2):A23=KSA1(1) AND
KSA1(2) AND KSA1(3):A24=KSA1(1) AND KSA1(2) AND KSA
1(3) AND KSA1(4)
1150 KA21=(A21 AND K1(1) OR A22 AND K1(2) OR A23 AND K1(
3) OR A24 AND K1(4)) AND NOT(KY22 OR KY23 OR KY1)
1160 KA22=(A22 AND CA(2) OR A23 AND CA(3) OR A24 AND CA(
4)) AND NOT(KA12 OR KY13 OR KA23 OR KY1)
1170 IF (KA21 OR KA22 OR KA13 OR KY12)=0 THEN T2=TIME:
LOCATE 58,3:PRINT"2.ASANSOR":LOCATE 58,4:PRINT"STOP

```



```
      ":LOCATE 58,5:PRINT"          ":CY(M)=0:KY=-1:GOTO 90
      ELSE GOTO 90
1180  ORIGIN 95,SX:DRAW 0,65,0:DRAW 50,65,0:DRAW 50,0,0:
      DRAW 0,0,0:ORIGIN 115,SX+10:DRAW 0,40,0:DRAW 10,40,0
      :DRAW 10,0,0:DRAW 0,0,0
1190  ORIGIN 95,SN:DRAW 0,65,9:DRAW 50,65,9:DRAW 50,0,9:
      ORIGIN 115,SN+10:DRAW 0,40,9:DRAW 10,40,9:DRAW 10,0,
      9:DRAW 0,0,9
1200  SX=SN
1210  GOTO 880
1220  ORIGIN 365,DX:DRAW 0,65,0:DRAW 50,65,0:DRAW 50,0,0:
      DRAW 0,0,0:ORIGIN 385,DX+10:DRAW 0,40,0:DRAW 10,40,0
      :DRAW 10,0,0:DRAW 0,0,0
1230  ORIGIN 365,DN:DRAW 0,65,9:DRAW 50,65,9:DRAW 50,0,9:
      DRAW 0,0,9:ORIGIN 385,DN+10:DRAW 0,40,9:DRAW 10,40,9
      :DRAW 10,0,9:DRAW 0,0,9
1240  DX=DN
1250  GOTO 1030
```


KAYNAKLAR

1. Schaefer P. "Process Control Computer Technology ", Siemens AG,Heyden 1978
2. A.E. Knowlton "Standart Handbook for Electirical Engineering",Mc Graw Hill Book Company.
3. Eveleigh " Introduction to Control şystems Design ", Mc Graw Hill Book Company.
4. Irving L. Kosov " Control of Electric Machines ",Pren tice Hall Series
5. " Z80 CPU " , Zilog Publications
6. "CPC 464 Firmware ",Locomotive Software And Amstrad Consumer Electronics Plc.
7. " Amstrad CPC 6128 User Instructions",Amstrad Consu mer Electronics Plc.

ÖZGEÇMİŞ

1963 Yılında Konyada doğdum. İlk öğrenimimi Karapınar yavuzselim ilk okulunda, orta öğrenimimi Cihanbeyli lisesi orta bölümünde, liseyi de Konya Gazi lisesinde iyi derece ile 1980 yılında tamamladım. Aynı yıl girmiş olduğum üniversite sınavlarında İ.D.M.M. Akademisinin elektrik bölümünü kazandım. İkinci yıldan sonra yapılan opsiyon değişikliği ile 1984 yılında Yıldız Üniversitesi Elektronik ve Haberleşme mühendisliği bölümünden mezun oldum. Aynı dönem yine Yıldız Üniversitesi Elektronik ve Haberleşme bölümünde master yapmaya başladım. Bir süre Yıldız Üniversitesi Elektrik bölümünde araştırma görevlisi olarak bulunduktan sonra halen Konya'da serbest olarak çalışmaktayım.