YILDIZ TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

Programlanır Denetleyici Tasarımı ve Simülasyonu

Ekrem Özorbeyi

Yüksek Lisans Tezi

YILDIZ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

PROGRAMLANIR DENETLEYİCİ TASARIMI VE SİMÜLASYONU

YÜKSEK LİSANS TEZİ

MÜH. EKREM ÖZORBEYİ

İSTANBUL 1988

YILDIZ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

PROGRAMLANIR DENETLEYİCİ TASARIMI VE SİMÜLASYONU

YÜKSEK LİSANS TEZİ

MÜH. EKREM ÖZORBEYİ

İSTANBUL 1988

Bu tez çalışmasının hazırlanması sırasında, fikir ve eleştirilerini esirgemeyen tez yürütücüm, değerli hocam Sayın Doç. Dr. Sezgin ALSAN'a ve Yıldız Üniversitesi Elektronik-Haberleşme Müh. Bölümü öğretim üyelerine en derin teşekkürlerimi sunarım.

İÇİNDEKİLER

ÖZET

Bu tez çalışması, endüstride denetimin sağlandığı özel amaçlı mikrobilgisayarların çalışma sistemini, tasarımını ve simülasyonunu içermektedir.

Birinci ve ikinci bölümde, konuya ilişkin sistem giriş-çıkışları, sayıcı ve zamanlayıcı yapılarından genel olarak sözedilmiştir.

Bu tür aygıtlarda denetim, oluşturulan denetleme programları aracılığıyla gerçekleştirildiğinden, üçüncü bölümde Programlanır Denetleyici yazılım komutları anlatılmıştır.

Bölüm dört, Z80 CPU kullanılarak, tasarlanan programlanır denetleyici yapısını ve özel işlev komutlarını içermektedir. Aynı bölüm içerisinde, Programlanır Denetleyici işletim sistemi programı ve özel işlev alt programları teker teker anlatılmıştır.

Son bölümde ise tasarlanan sistemin simülasyonunu sağlamak amacıyla, Z80 CPU temelli 6128 Amstrad mikrobilgisayarı üzerinde gerçekleştirilen çalışma anlatılmıştır.

SUMMARY

In this research the working system of the special aimed microcomputer, design and the simulation have been worked out which provide the control in the industry.

In the first and second chapters, input and output - the system - counter and timer have been mentioned being related to the subject.

Programmable controller software instructions have been discussed in the third chapter. Because, in such kind of devices the control can be provided with the help of controlled programs.

The forth chapter consists of the function instructions and the building of programmable controller, designed by using the Z80 CPU. In the same chapter, progrmammeble controller system software and subroutines of function instructions are discussed in detail.

Consequently, in the last chapter in order to provide the simulation of the designed siystem, the work realized on the 6128 Amstrad microcomputer based Z80 CPU are explained.

1- GİRİŞ

Günümüzde PROGRAMLANIR DENETLEYİCİLER endüstrinin bir çok koluna girmiş ve yaygınlık kazanmıştır. Daha önceden, denetleyici sistemler, röle ve kontaktörlerle tasarlanan konvansiyonel yolla oluşturulmaktaydı. Konvansiyonel sistemin yaygınlık kazanmasına rağmen, aşağıdaki olumsuz etkilerden dolayı, yerini Programlanır denetleyicilere (P.D. şeklinde gösterilecektir) bıraktı.

1- Çalışma hızının çok düşük olması.
2- Fiziki boyutların çok büyük olması.
3- Sağladığı olanakların kısıtlı olması.
4- Elektro-mekanik yapısından dolayı bol arıza yapması.
5- Enerji tüketiminin fazla olması.
6- Varolan sistem üzerinde değişiklik yapma olanağının çok az olması.
7- Denetlenecek giriş-çıkış sayısının artmasıyla birlikte sistem donanımının karmaşık hale gelmesi

Bu olumsuz sebepler ve teknolojinin gelişmesi, konvansiyonel sistemin yerini, elektronik sabit devrelere bırakmış, yarı iletken malzemeler, denetleme devrelerinde kullanılmaya başlanmıştır. Bu gelişmeler, çalışma hızının azalmasını ve çalışma güvenliğini sağlamış, fakat yinede yetersiz kalmıştır. Bir sonraki aşamada, mikroişlemcilerin gelişmesi, araştırmacıları bu alandada mikroişlemci temelinde çalışmaya yöneltti. Bu gelişmeler 70'li yılların ortasında başlayıp, 80'li yılların başlarında büyük hız ve aşama kaydetmiştir.

Mikroişlemci ile tasarlanan denetleyicilerde, temel yazılıma dayanmaktadır. Denetleme için kullanılan elemanların tümü, yazılımla oluşturulmuş, sistem giriş-çıkışları modüller halinde donanım ile sağlanmıştır. Konvansiyonel yöntem ile oluşturulan denetleme devrelerinde, röle kontakları işleve göre seri veya paralel olarak birbirlerini tamamlar. Oluşturulan denetleme devresinde, kontakların seri ya da paralel gelmesi, P.D. komutlarında lojik işlemlere (VE VEYA gibi) dönüştürülür.

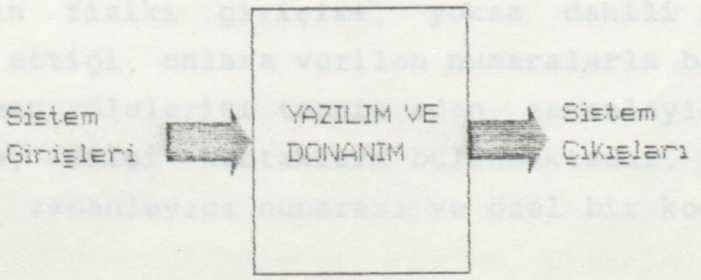
Böylelikle konvansiyolen sistemle oluşturulan devre yapısı, P.D. komutları aracılığıyla yazılıma dönüştürülebilir. Daha önceden oluşturulan devre yapıları P.D.'lerde değerlendirilebilir. Oluşan bu bağla, eski sistem ile çalışan kişiler, Programlanır denetleyiciler ile çalışmaya çok çabuk uygunluk gösterebilirler.

Bu tez çalışmasında genel P.D. komutları incelenerek, Z80 CPU ile P.D. sistem yazılımı oluşturulup, gereken donanım tasarlanmıştır. Z80 CPU temelli 6128 Amstrad mikrobilgisalyar üzerinde simülasyonu sağlanmıştır.

2. PROGRAMLANIR DENETLEYİCİ

2.1 GENEL BİLGİLER

Genel anlamda denetleme işlemlerinde üç kısımdan söz edilebilir. Giriş bilgileri, girişteki bilgileri değerlendirecek ve istenilen işlevleri sağlayacak yazılım ve donanım, işlemler sonucu oluşacak çıkış bilgileri (Şekil 2.1.1) Giriş bilgileri, sınır anahtarları (limit switiches), basçek butonlar (push buttons), seviye duyargaları (level sensors), basınç ve sıcaklık duyargarından elde edilen var-yok veya değişken işaretler olabilmektedir. Bu giriş bilgilerini değerlendirecek, istenilen işlevi sağlayacak kısım ise yazılımla oluşturulmuştur. Denetleme programının yazılması için, belirli P.D. komutları bulunmaktadır. Bu komutlar, oluşturulan devre şemasında (ladder diagram), kontakların durumlarına göre lojik ifadeler (VE VEYA gibi), Zamanlayıcılar (Timer), Sayıcılar (Counter) ve bazı aritmetik, lojik özel işlevleri içerir. Bunlar yazılımla oluşturulduğu için, kullanıcının isteğine göre, özel birtakım işlevlerde ilave edilebilir.



Şekil 2.1.1

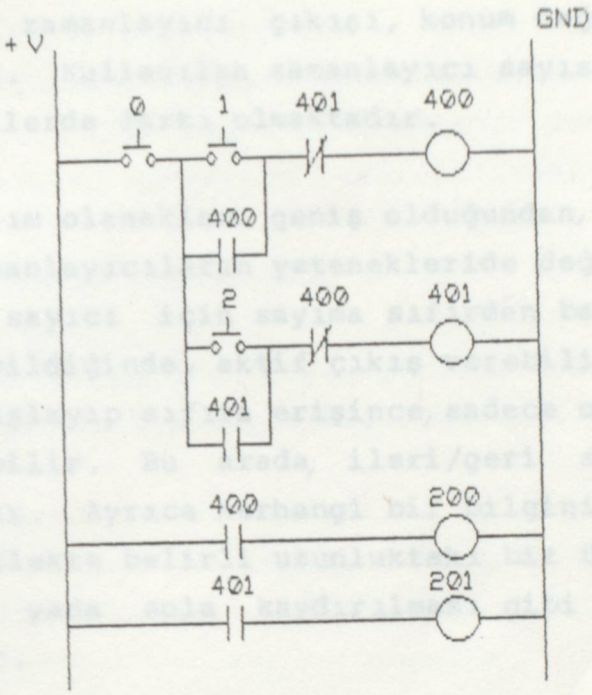
Denetleme işlemi için oluşturulan devre şemasında, kontakların seri veya paralel olması, P.D. komutlarında karşılık bulurlar. Dolayısıyla devre şeması (ladder diagram) PD program listesine dönüştürülür. Bu program, girişler izlenerek, PD üzerinden koşturulur. Her çevrimde, gi-

rişlerin durumuna göre değişen işlem sonuçları, çıkışları, etkiler. Program bitince, tekrar başa dönülerek işlemler yeniden yapılır. Bu işlem kör bir çevrim halinde devam eder.

2.2. GİRİŞ VE ÇIKIŞLAR

Herhangi bir denetleme işleminin yapılabilmesi için, belirli giriş bilgilerine ihtiyaç vardır. Bu giriş bilgileri, denetleme sisteminde değerlendirilir ve işlem sonucu, sistem çıkışlarına yansıtılarak denetleme sağlanır.

Programlanır denetleyici yazılımında, fiziki girişler, kontaklar şeklinde ifade edilir. Yine yazılım içerisinde, kontak çoğaltma röleleri, özel amaçlı röleler ve kontaklarına ihtiyaç duyulur. Bu tür rölelere, dahili röle (Internal Relay) adı verilir. Program içerisinde yer alan kontağın fiziki girişimi, yoksa dahili röle kontağınımı temsil ettiği, onlara verilen numaralarla belirlenir. Ayrıca zaman rölelerini temsil eden, zamanlayıcıların ve sayıcıların, bilgi kontakları bulunmaktadır. Bunlarda, ilgili sayıcı zamanlayıcı nunarası ve özel bir kodla belirlenmektedir.



Şekil 2.2.1

Şekil 2.2.1'de verilen devre grafik gösteriminde (ladder diagram), 0,1,2 nolu fiziki girişlerini, 200 ve 201 nolu çıkışlar ise fiziki çıkışları göstermektedir. Dahili röle ve kontaklarını 400 ve 401 numaralar temsil etmektedir. Fiziki giriş-çıkış ve dahili röle sayısı, P.D. tasarımına göre değişmektedir. Yine giriş ve çıkış gerilim değerleri, farklı P.D.'lerde farklı olmaktadır. Örneğin IDEC IZUMI firmasının ürettiği FA1-JUNIOR P.D.'sinde 5A 110V röle çıkışı, 0,5A/24V dc, 20mA/24V, 50 mA/24V 1A/24V DC gerilim değerlerinde transistor çıkışı bulunmaktadır.

2.3. ZAMANLAYICILAR VE SAYICILAR

Denetleme işlemlerinde kullanılan zaman röleleri, P.D.'lerde yazılımla oluşturulan zamanlayıcılara karşılık düşmektedir. Ayrıca yazılım olanaklarından yararlanılarak,

kullanıcıya, çeşitli özellikte sayıcılar sunulmuştur. Zamanlayıcılarda, zamanlama süresi farklı değerlerde olabilmektedir. Seçilen değere erişildiğinde, yazılım tarafından oluşturulan zamanlayıcı çıkışı, konum değiştirerek bilgi üretmektedir. Kullanılan zamanlayıcı sayısı ve olanakları farklı P.D.'lerde farkı olmaktadır.

Yazılım olanakları geniş olduğundan, tasarlanan sayıcı ve zamanlayıcıların yetenekleride değişmektedir. Örneğin, bir sayıcı için sayıma sıfırdan başlayıp, seçilen değere erişildiğinde, aktif çıkış verebilir. Veya seçilen değerden başlayıp sıfıra erişince, sadece o aralıkta, aktif çıkış verebilir. Bu arada, ileri/geri sayıcı türleride bulunmaktadır. Ayrıca herhangi bir bilginin, seçilmiş özel kontak (bellekte belirli uzunluktakı bit üzerinde) bölgesinde, sağa yada sola kaydırılması gibi özel yetenekler sağlanmıştır.

2.4. P.D. ÖZEL İŞLEVLERİ

Yazılım olanakları kullanılarak çeşitli özel işlevler gerçekleştirilmektedir. Bunlar arasında toplama, çıkarma, bölme gibi aritmetik işlemler, veri karşılaştırma artırma, azaltma, veri kaydırma, sayıcı sayılmış değeri karşılaştırma, harici veri göstericisine değer aktarma vb.g. işlevler bulunmaktadır. Bellekte belirli bir bölge bu işlemler için ayrılmıştır. Bu bölgeye veri kayıtcıları (Data register) bölgesi adı verilmektedir.

Örneğin FA1-JUNIOR P.D.'seninde 16 bit'lik yüz adet veri kayıtçısı tanımlanmıştır. İlk iki veri kayıtçısı, ana kayıtçılardır. Aritmetik işlemler bu kayıtçılar üzerinde gerçekleşmektedir. Ayrıca ihtiyaçlar sonucu ortaya çıkacak bazı özel işlevler, tasarım sırasında ilave edilebilir.

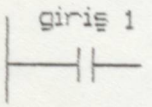
3. PROGRAMLANIR DENETLEYİCİ KOMUTLARI

3.1. GİRİŞ

Programlanır denetleyici komutları, farklı standartlara göre, isim ve gösterim olarak farklılık göstermektedir. Bu tez çalışmasında kullanılan komutlarda, IDEC IZIMU Japon firmasının FA1-JUNIOR Programlanır denetleyicisinde kullandığı, Amerikan notasyonu temel alınmıştır.

3.2 LOD-SATIR BAŞLANGICI

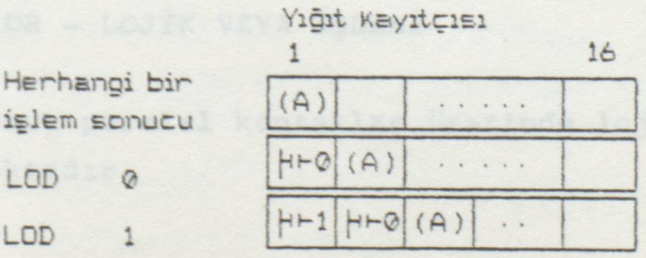
Bu komut bir satırın başlangıcında kullanılır. Bir satır kesinlikle LOD komutuyla başlamalıdır.



Komut	Data
LOD	1

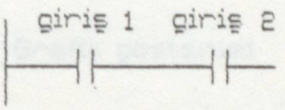
Grafik gösterimi Yazılımı

Bu komutla, numarası belirli olan giriş yada dahili röle bellekte bulunduğu gözden okunarak, yığıt kayıtçısına aktarılır. Bundan sonraki işlem komutu için, durum, yağıt kayıtçısında saklanır. ANDLOD, ORLOD, CNT işlemleri için arka arkaya gelen LOD komutlarıyla, kontakların durumları, yığıt kayıtcısına ardışıl olarak saklanır. Bu komut, sayıcı ve zamanlayıcı kontaklarıyla birlikte de kullanılabilir.



3.3 AND - LOJİK VE İŞLEMİ

Bu komut seri olarak bağlanmış kontaklar arasında lojik VE (And) işlemi yapmaktadır.



Grafik gösterimi

Komut	Data
LOD	1
AND	2

Yazılımı

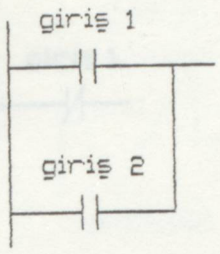
Giriş 1	Giriş 2	Sonuç
ON	ON	ON
ON	OFF	OFF
OFF	ON	OFF
OFF	OFF	OFF

Doğruluk tablosu

Bu komut, birden fazla seri kontaklar, sayıcı, zamanlayıcı, kaydırma kayıtçısı kontakları üzerinde kullanılabilir.

3.4. OR - LOJİK VEYA İŞLEMİ

Bu komut paralel kontaklar üzerinde lojik VEYA (OR) işlemi yapmaktadır.



Grafik gösterimi

Komut	Data
LOD	1
OR	2

Yazılımı

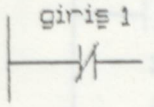
Giriş 1	Giriş 2	İşlem Sonucu
ON	ON	ON
ON	OFF	ON
OFF	ON	ON
OFF	OFF	OFF

Doğruluk tablosu

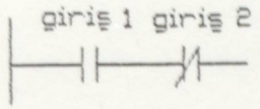
OR komutu, ardışıl olarak birden fazla kullanılabilir. Sayıcı, zamanlayıcı, kaydırma kayıtçısı kontakları üzerinde de tanımlıdır.

3.5. NOT - EVRİK İŞLEMİ

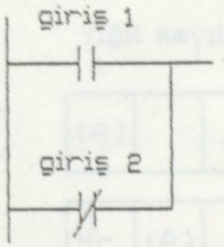
Bu komut yardımcı bir komuttur. Okunan giriş durumunun evriğini almak için AND, OR, LOD komutlarıyla birlikte kullanılır.



Komut	Data
LODNOT	1



LOD	1
ANDNOT	2



LOD	1
ORNOT	2

Grafik gösterimi Yazılımı

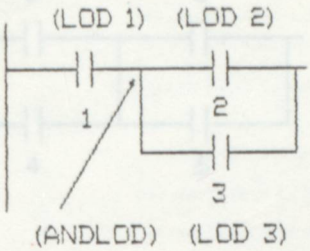
Giriş	Çıkış
ON	OFF
OFF	ON

Doğruluk tablosu

Ayrıca, yukarıdaki komut çiftleri, sayıcı, zamanlayıcı kaydırma kontakçısı, üzerinde de kullanılabilir.

3.6. ANDLOD İŞLEMİ

ANDLOD komutu, seri LOD komutuyla başlayan, kontakları birleştirmek için kullanılır.



Grafik gösterimi

Komut	Data
LOD	1
LOD	2
OR	3
ANDLOD	

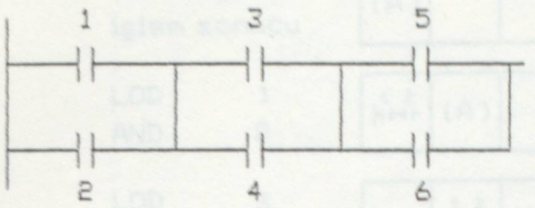
Yazılımı

Yığıt Kayıtçısı

	1				16
Herhangi bir işlem sonucu	(A)			...	
LOD 1	1	(A)		...	
LOD 2 OR 3	2	1	(A)	...	
ANDLOD	1 2 3	(A)		...	

Yukarıdaki işlemde, 1 numaralı giriş durumu okunur, yığıt kayıtçısına sürülür. Bir sonraki satırda bulunan 2 numaralı girişte okunarak yığıt kayıtçısına konur. Sonraki satırda bulunan 3 numaralı kontakla, 2 numaralı kontak OR işlemi yapılarak, sonuç yığıt kayıtcısına konur. ANDLOD komutu ile, elde edilen sonuç ve 1 numaralı giriş, AND işlemine sokulur ve sonuç, yığıt kayıtçısının ilk bitine konur. Bir çıkıştan önce birden fazla ANDLOD komutu kullanılabilir. ANDLOD sayısı, LOD sayısının bir eksiğine eşit olmalıdır.

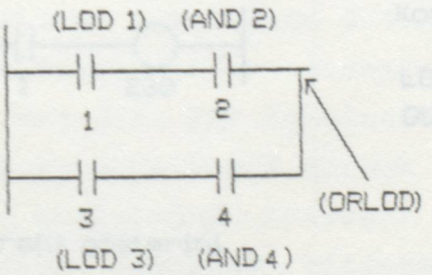
ÖRNEK



	LOD	1
	OR	2
ANDLOD	LOD	3
	OR	4
ANDLOD	LOD	5
	OR	6
	ANDLOD	
	ANDLOD	

3.7. ORLOD İŞLEMİ

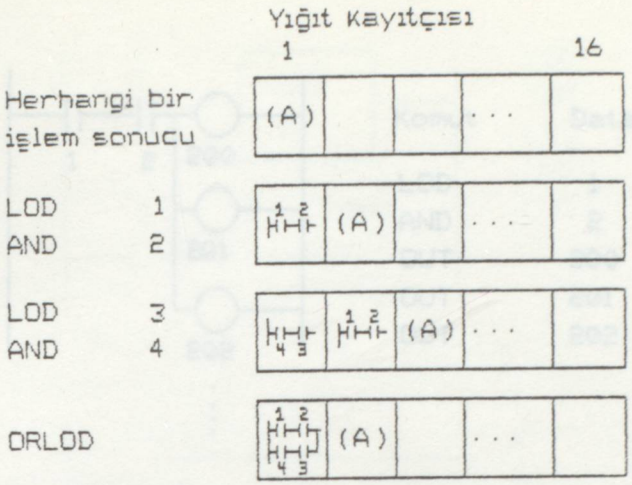
ORLOD komutu, LOD komutuyla başlayan, paralel satırları birbirine bağlamak için kullanılır.



Komut	Data
LOD	1
AND	2
LOD	3
AND	4
ORLOD	

Grafik gösterimi

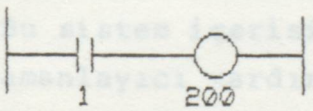
Yazılımı



Bu komuta gelindiğinde, yığıt kayıtçısında bulunan son iki bit üzerinde lojik OR işlemi yapılarak, sonuç yığıt kayıtçısının son bit'ine kaydedilir. Bir çıkıştan önce birden fazla ORLOD komutu kullanılabilir.

3.8. OUT - ÇIKIŞ KOMUTU

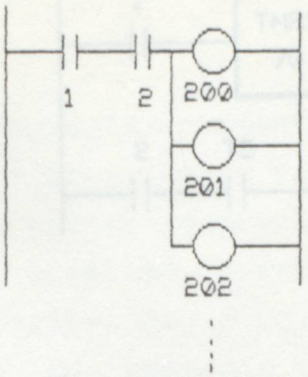
OUT komutu, bu komuta gelininceye kadar, gerçekleşen işlem sonucunu, numarası verilen çıkışa aktarır.



Komut	Data
LOD	1
OUT	200

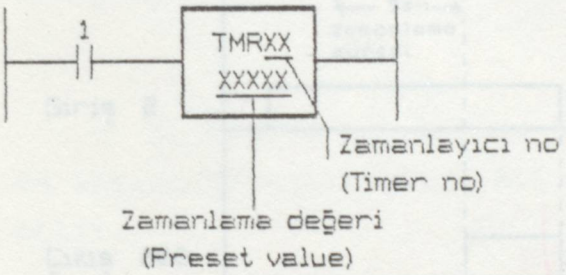
Grafik gösterimi — Yazılımı

OUT Komutu, fiziki çıkış, dahili ve özel röleler için geçerlidir. Aynı düğüm noktasına, birden fazla çıkış bağlanabilir. Program içerisinde, aynı numaralı çıkış kontakları, birden fazla kullanılabilir.



Komut	Data
LOD	1
AND	2
OUT	200
OUT	201
OUT	202

3.9 TMR - ZAMANLAYICI (TIMER)



Grafik gösterimi

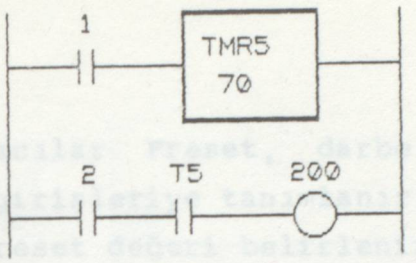
LOD 1
TMR XX XXXXX

Zamanlama değeri

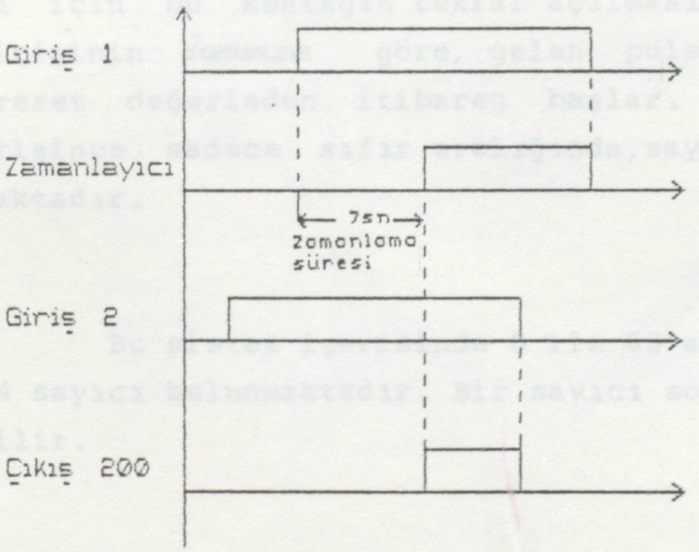
Zamanlayıcı no

Yazılımı

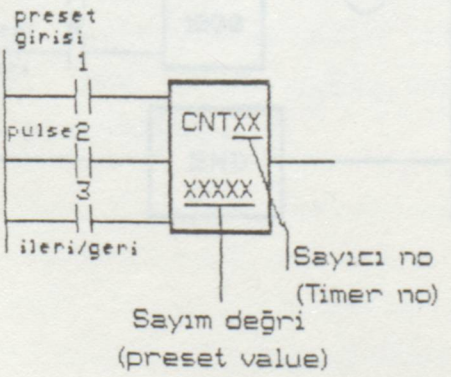
Bu sistem içerisinde 0 dan 79'a kadar numaralı, 80 tane zamanlayıcı vardır. Zamanlayıcılarda, temel zaman süresi 100 msn dir. TMR komutunda verilen zamanlama değeri 100 msn ile çarpılarak gerçek zaman değeri elde edilir. Zamanlayıcı, girişindeki kontağın kapanmasıyla birlikte çalışmaya başlar. Zamanlama bitinceye kadar bu kontak kapalı kalmalıdır. Programda verilen zamanlama değerine erişildiğinde, zamanlıyıcı çıkışı aktif olmaktadır. Girişindeki kontak kapalı kaldığı sürece çıkış aktif olmaktadır. Zamanlayıcı çıkışının pasif olması için, giriş kontağının açılması gerekir.




```
LOD     1
TMR     5     70
LOD     2
ANDT    5
OUT     200
```



3.9. CNT - SAYICI (COUNTER)



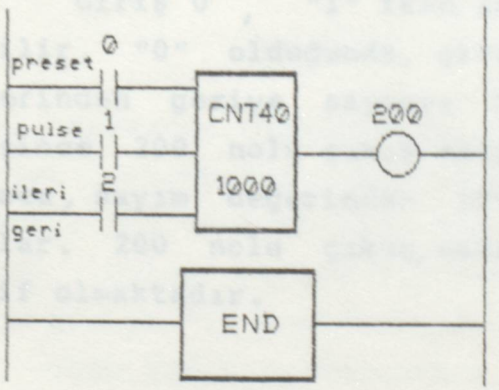

```
LOD     1
LOD     2
LOD     3
CNT     XX    XXXXX
```

Sayım değeri

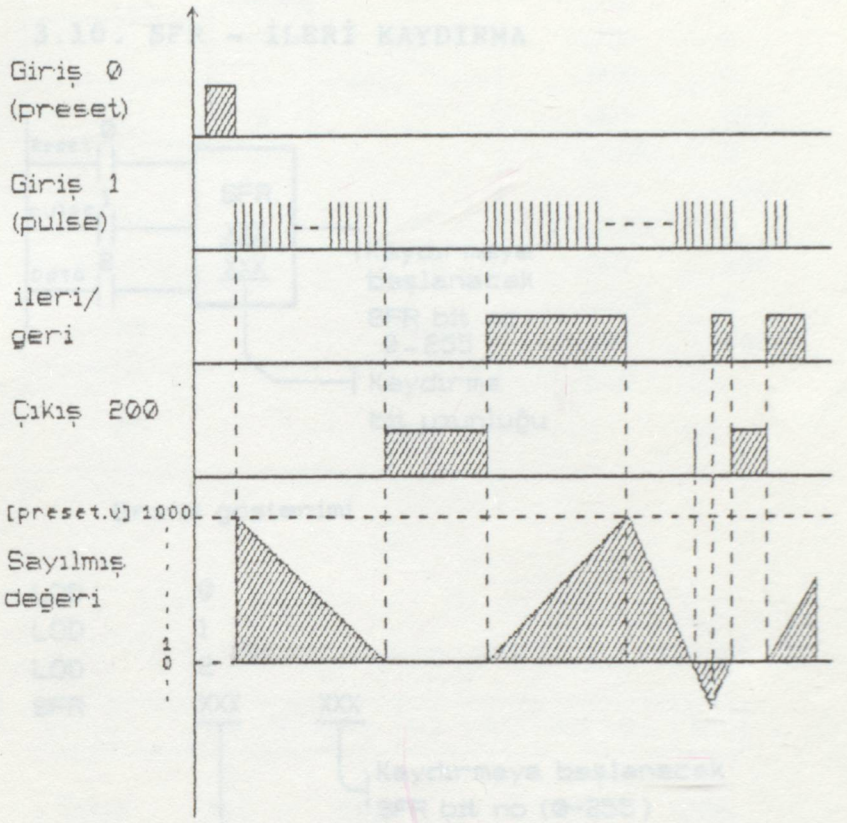
Sayıcı no

Sayıcılar Preset, darbe (pulse), ileri/geri (up/down) girişleriye tanımlanır. Yazılımında, sayıcı numarası ve preset değeri belirlenir. Preset girişinin kapanmasıyla birlikte, preset değeri sayıcının ilişkin gözüne yerleştirilir. Sayım için gelecek darbelerin, geçerli olması için bu kontağın tekrar açılması gerekir. İleri/geri girişinin konumuna göre, gelen pulse'lerle sayım işlemi, preset değerinden itibaren başlar. Sayım işlemi, sıfıra erişince, sadece sıfır aralığında, sayıcı çıkışı aktif olmaktadır.

Bu sistem içerisinde 0 ile 63 arasında numaralanmış 64 sayıcı bulunmaktadır. Bir sayıcı sonuna çıkış bağlanabilir.

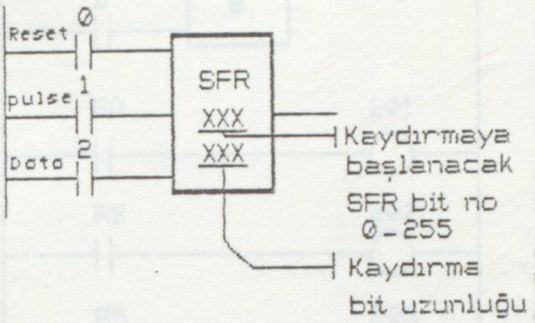



```
LOD   0
LOD   1
LOD   2
CNT   40   1000
OUT   200
END
```



Giriş 0 , "1" iken, sayım değeri sayıcıya yerleştirilir. "0" olduğunda, giriş 1'e gelen darbelerle sayma değerinden geriye saymaya başlanır. Sayım değeri sıfıra erişince 200 nolu çıkış aktif olur. 2 numaralı giriş "1" olunca, sayım değerinden itibaren, yukarıya doğru saymaya başlar. 200 nolu çıkış, sadece sayım değeri sıfır olunca aktif olmaktadır.

3.10. SFR - İLERİ KAYDIRMA



Grafik gösterimi

```
LOD     0
LOD     1
LOD     2
SFR     XXX     XXX
```

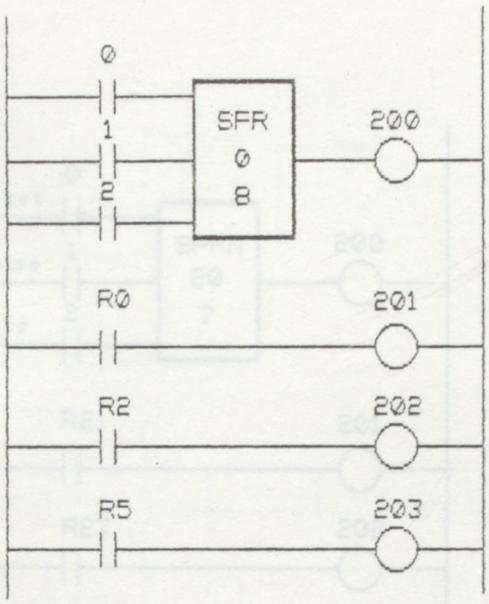
Kaydırmaya başlanacak SFR bit no (0-255)

Kaydırma bit uzunluğu

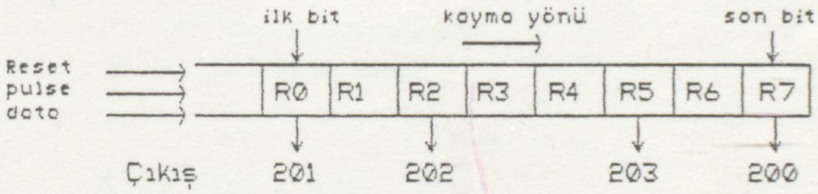
Yazılımı

Kaydırma kayıtçısı, topmam 256 kontağı temsil eden bit'ten oluşmuştur. Kaydırma işlemi, belirlenen başlangıç numarası ve uzunluk üzerinde gerçekleşir. SFR reset, veri (data), pulse ve SFR yazılım komutuyla programlanabilir. Reset girişi "1" seviyesindeyken, SFR kontaklarını (bit'lerin) tümü sıfır durumundadır. Reset girişi sıfır olduktan sonra, pulse ucuna gelen her çıkan kenar darbeyle, data girişindeki bilgi, yazılımla belirlenen numaralar üzerinde kaymaya başlar. Kaydırma kayıtçısı kontakları, fiziki giriş, dahili röle kontakları gibi program içerisinde kullanılır.

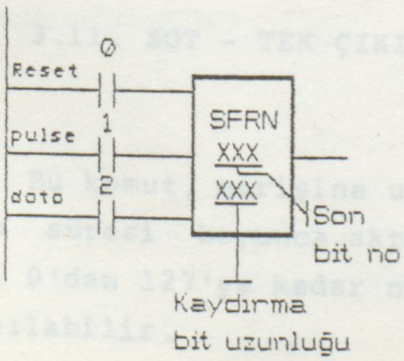
ÖRNEK




```
LOD     0
LOD     1
LOD     2
SFR     0       8
OUT     200
LODR    0
OUT     201
LODR    2
OUT     202
LODR    5
OUT     203
END
```



SFR komutuyla soldan sağa kaydırma gerçeklenir. Ters yönde kaydırma işlemi için SFRNOT komutu kullanılarak gerçeklenebilir. SFRNOT reset, veri (data), pulse girişleri ve SFRNOT yazılım, komutuyla gerçeklenebilir.




```
LOD     0
LOD     1
LOD     2
SFR     XXX     XXX
```

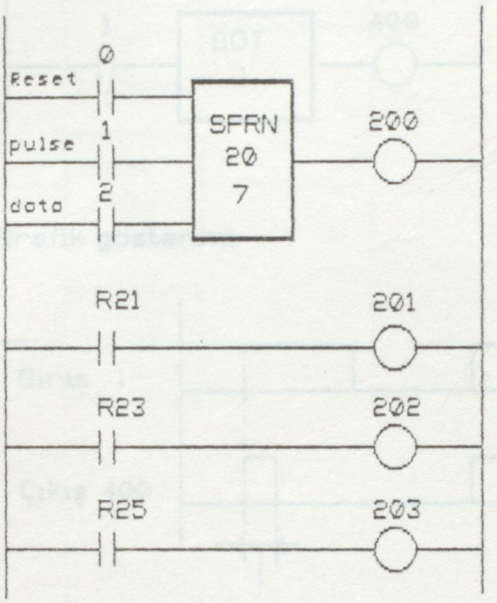
Kaydırma bit uzunluğu

Son bit no

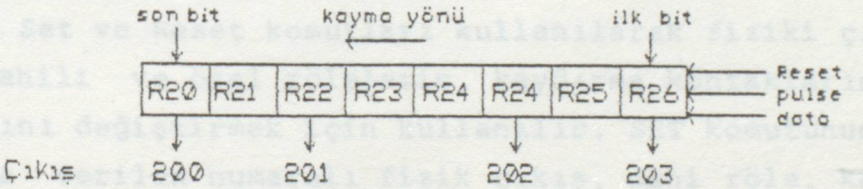
Grafik gösterimi

Yazılımı

ÖRNEK

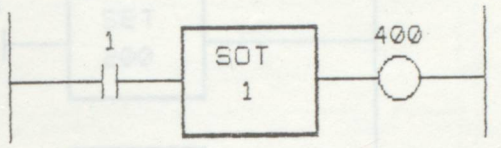



```
LOD     0
LOD     1
LOD     2
SFR     20     7
OUT     200
LODR    21
OUT     201
LODR    23
OUT     202
LODR    25
OUT     203
END
```



3.11. SOT - TEK ÇIKIŞ (SINGILE OUT)

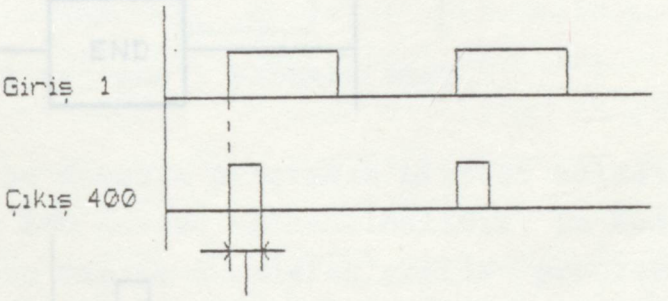
Bu komut, girişine uygulanan darbe ile sadece bir çevrim süresi boyunca,aktif çıkış elde etmek için kullanılır. 0'dan 127'ye kadar numaralandırılmış 128 SOT işlemi kullanılabilir.




```
LOD   1
SOT   1
OUT   400
```

Grafik gösterimi

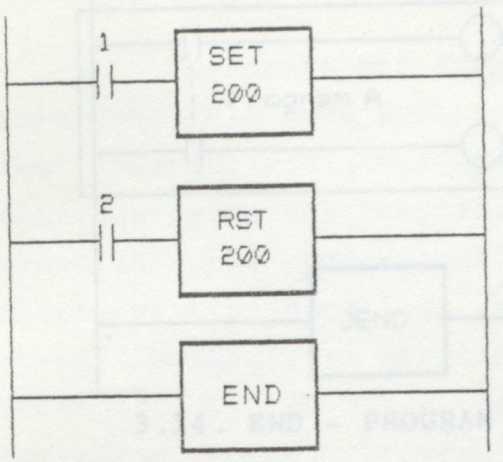
Yazılımı



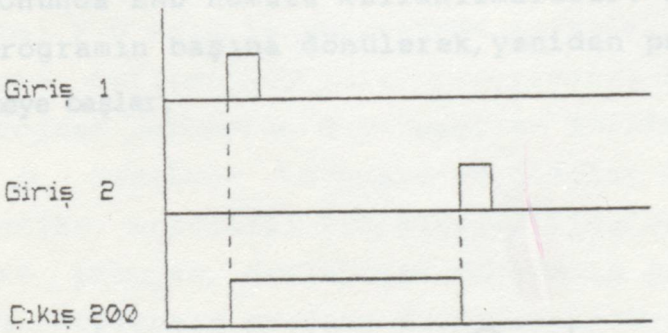
Bir çevrimsüresi
(1 scan time)

3.12 SET - RST

Set ve Reset komutları kullanılarak fiziki çıkışların dahili ve özel rölelerin, kaydırma kontaklarının durumlarını değiştirmek için kullanılır. SET komutunun yazılımında verilen numaralı fizik çıkış, dahi röle, kaydırma kayıtcısı kontağı aktif ("1") seviyeye getirilebilir. RST komutu kullanılarak ilgili kontağı pasif ("0") seviyeye getirmek mümkündür.

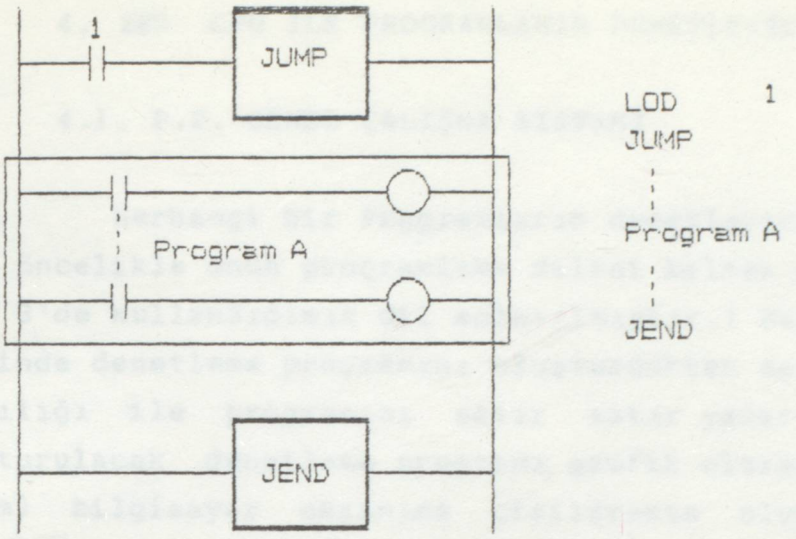



```
LOD   1
SET   200
LOD   2
SET   200
END
```



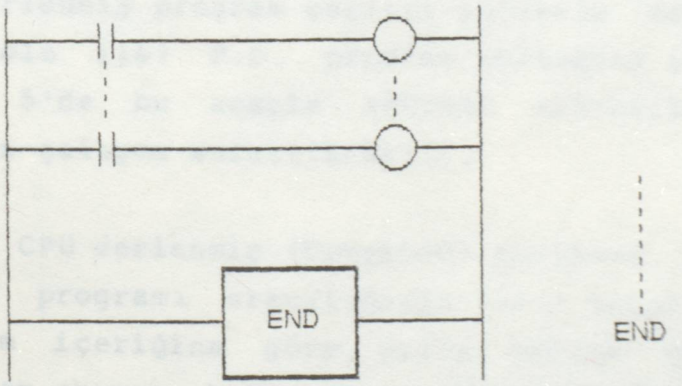
3.13 JUMP - JEND

Bu komut ikilisi program içerisindeki bir program parçasının işlenmesi yada işlenmemesine karar verir. JUMP komutundan önce gelen kontak kapalı ise program akışı değişerek JEND komutunun bulunduğu satıra gelinir ve arada kalan program parçası işlenmez. Eğer JUMP komutundan önce gelen kontak açık ise atlama işlemi gerçekleşmez ve bütün program işlenir.



3.14. END - PROGRAM SONU

Bu komutla, programın bittiği anlaşılır. Her program sonunda END Komutu kullanılmalıdır. Bu komut görüldüğünde programın başına dönülerek, yeniden program satırları işlenmeye başlar.



4. Z80 CPU İLE PROGRAMLANIR DENETLEYİCİ TASARIMI

4.1. P.D. GENEL ÇALIŞMA SİSTEMİ

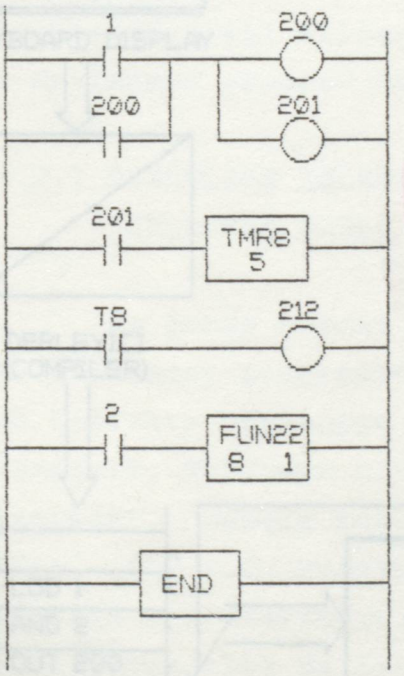
Herhangi bir Programların denetleyiciyi kullanmak için öncelikle onun programlama dilini bilmek gerekir. (Bölüm 3'de kullandığımız dil anlatılmıştır.) Kullanıcı, kağıt üzerinde denetleme programını oluşturduktan sonra, Keyboard aracılığı ile programını satır satır yazar. Örnek 4.1.1. Oluşturulacak denetleme programı, grafik olarak (ladder diagram) bilgisayar ekranına çizilerekte oluşturulabilir. Örnek 4.1.2. Ekrandaki şekil algılanarak, derlenip program belleğine konur. Program mantığını oluşturmak ve programı yazmaya ilk aşama diyebiliriz. Bundan sonraki aşamada ise yazılan programı yada çizilen grafik programı, (Ladder diagram) P.D işletim sisteminde anlamlı hale getirmektir. Bunun için derleme (Compile) programları kullanılır. Oluşturulan program derlenerek P.D. program belleğine yerleştirilir. Her P.D. üzerinde keyboard ve display olmayabilir. Bazı programlar aracılığı ile, kişisel bilgisayarlar üzerinden yazılan program, derlenerek derlenmiş program elde edilebilir. Bu derlenmiş program çeşitli yollarla (örneğin seri iletişim yolu ile) P.D. program belleğine yerleştirilebilir. Bölüm 5'de bu amaçla AMSTRAD mikrobilgisayarı üzerinde yapılan çalışma anlatılacaktır.

CPU derlenmiş (Compiled) programı, yazılan işletim sistem programı aracılığıyla satır satır işlemeye başlar. Program içeriğine göre, giriş bellek bölgesinden ilgili bilgiler okunur, komut alt programlarında işlevler gerçeklenir ve ilgili çıkışlar uyarılır. Program akışında END komutuna rastlanınca programın başına dönülerek, bu iş kör bir çevrim halinde devam eder. Programın birkez dönme süresine çevrim süresi (SCAN TIME) denir. Giriş-çıkış sayısının artması, programın uzunluğu, çevrim süresinin artmasıyla yakından ilgilidir. Bu değer FAI-JUNIOR İÇİN 32 ms/1K steps dir.

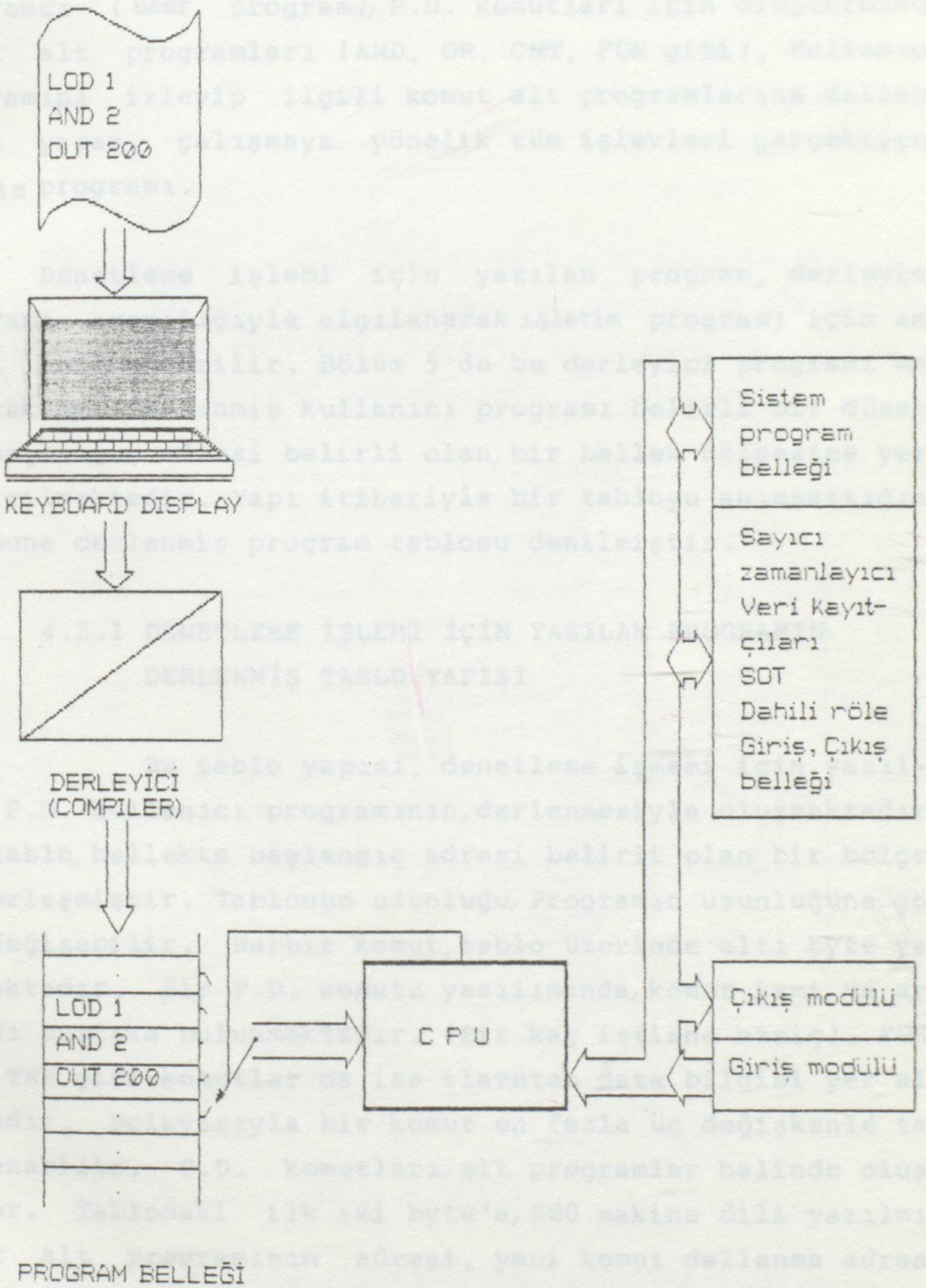
ÖRNEK 4.1.1

```
10 LOD 1
20 OR 200
30 OUT 200
40 OUT 201
50 LOD 201
60 TMR 8 5
70 LODTMR 8
80 OUT 212
90 LOD 2
100 FUN22 81
110 END
```

ÖRNEK 4.1.2



P.D. ilk açıldığında bütün sayıcılar, zamanlayıcılar veri kayıtçıları dahili röleler, özel bayrak röleleri sıfırlanır. Bütün fiziki girişler okunur ve giriş bellek (input memory area) bölgesine yerleştirilir. Program işlemeye başlar, çevrim sonunda fiziki çıkışlar uyarılır. Bu işlemler her çevrimde tekrarlanır.


LOD 1
AND 2
OUT 200
KEYBOARD DISPLAY
DERLEYİCİ
(COMPILER)
LOD 1
AND 2
OUT 200
PROGRAM BELLEĞİ
C P U
Sistem
program
belleği
Sayıcı
zamanlayıcı
Veri kayıt-
çıları
SOT
Dahili röle
Giriş, Çıkış
belleği
Çıkış modülü
Giriş modülü

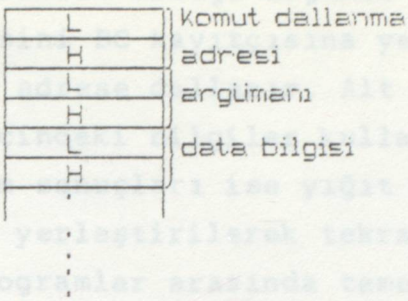
4.2 PROGRAMLANIR DENETLEYİCİ YAZILIMI

Programlanır denetleyici yazılımını üç kısma ayırabiliriz. Denetleme sistemi için oluşturulmuş kullanıcı programcı (user program), P.D. komutları için oluşturulmuş komut alt programları (AND, OR, CNT, FUN gibi), Kullanıcı programını izleyip ilgili komut alt programlarına dallandırma yapan, çalışmaya yönelik tüm işlevleri gerçekleyen işletim programı.

Denetleme işlemi için yazılan program, derleyici programı aracılığıyla algılanarak işletim programı için anlamlı hale getirilir. Bölüm 5'de bu derleyici programı anlatacaktır. Derlenmiş kullanıcı programı belirli bir düzende, başlangıç adresi belirli olan, bir bellek bölgesine yerleştirilmektedir. Yapı itibariyle bir tabloyu anımsattığından buna derlenmiş program tablosu denilmiştir.

4.2.1 DENETLEME İŞLEMİ İÇİN YAZILAN PROGRAMIN DERLENMİŞ TABLO YAPISI

Bu tablo yapısı, denetleme işmemi için yazılmış P.D. kullanıcı programının, derlenmesiyle oluşmaktadır. Bu tablo, bellekte başlangıç adresi belirli olan bir bölgeye yerleşmiştir. Tablonun uzunluğu, Programın uzunluğuna göre değişebilir. Herbir komut, tablo üzerinde altı byte yer tutmaktadır. Bir P.D. komutu yazılımında, komut ismi ve argümanı mutlaka bulunmaktadır. (Bir kaç istisna hariç). FUN, CNT, TMR gibi komutlar da ise ilaveten data bilgisi yer almaktadır. Dolayısıyla bir komut en fazla üç değişkenle tanımlanabilir. P.D. komutları, alt programlar halinde oluşmuştur. Tablodaki ilk iki byte'a, Z80 makina dili yazılmış komut alt programının adresi, yani komut dallanma adresi yerleştirilir. İkinci iki byte'a ise, argüman bilgisi, üçüncü iki byte'a ise, eğer varsa data bilgisi konur. Şekil 4.2.1.1.



Şekil 4.2.1.1

Bu yapının anlaşılması amacıyla aşağıda bir örnek verilmiştir. (Örnek 4.2.1.1.) Yazılan program, derleme programı aracılığıyla derlenmiş program tablosuna dönüştürülür. Data gözü, TMR, FUN, CNT komutlarında dolmaktadır.

```
LOD     1
LOD     715
LOD     7
CNT     40    20
OUT     200
END
```

# 8000	#80	#8012	#48
	#64		#61
	#01	CNT 40	#34
LOD 1	#00	20	#00
	#00		#14
	#00		#00
#8006	#80	#8018	#C2
	#64		#64
	#CB		#CB
LOD 715	#02	OUT 200	#00
	#00		#00
	#00		#00
#800C	#80	#801E	#00
	#64		#00
	#07		#00
LOD 7	#00	END	#00
	#00		#00
	#00		#00

ÖRNEK 4.2.1.1.

4.2.2. KOMUTLAR İÇİN OLUŞTURULMUŞ ALT PROGRAMLAR

Alt programlar, herbir komut için o komutun işlevine göre oluşturulmuş programları içerir. İşletim programı, der-

lenmiş program tablosundan aldığı argüman bilgisini, HL kayıtçısına, data bilgisini BC kayıtçısına yerleştirir ve ilk iki byte'ta bulunan adrese dallanır. Alt programdan HL ve eğer gerekiyorsa BC içindeki bilgiler kullanılır. Alt programlarda yapılan işlem sonuçları ise yığıt kayıtçısının (DE kayıtçısı) son bitine yerleştirilerek tekrar işletim programına dönülür. Alt programlar arasında temel lojik komutları, sayıcılar, zamanlayıcılar, özel işlev programları bulunmaktadır. Tablo 4.2.2.1.'de alt programlar ve onların dallanma adresleri verilmiştir.

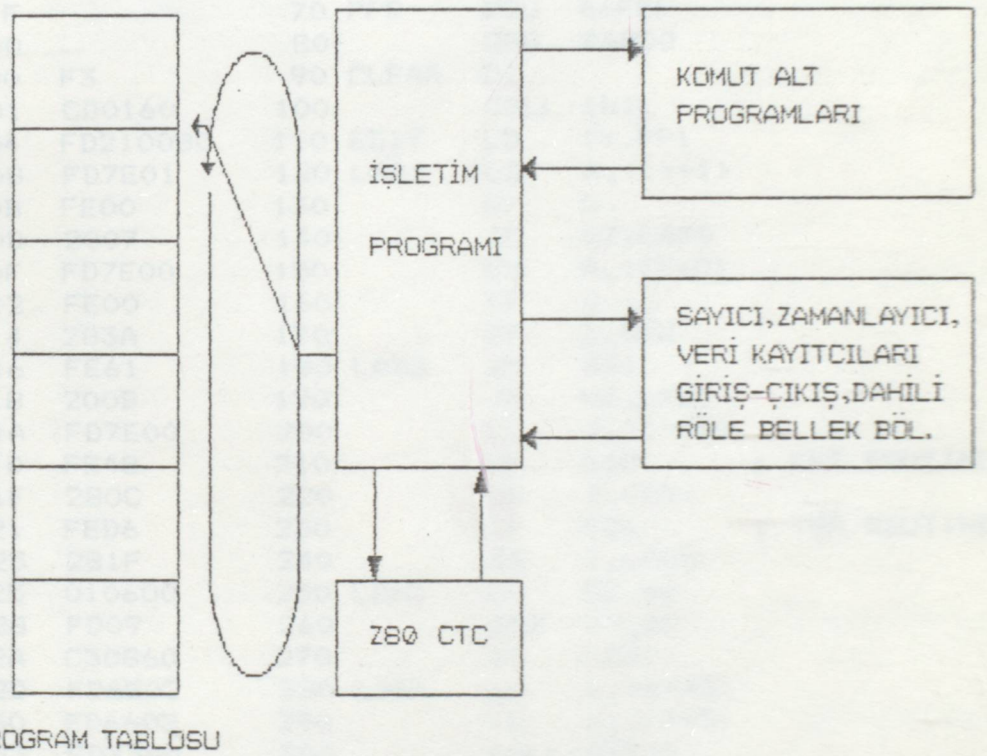
KOMUT İSMİ	CALL ADRESİ
INITIAL	#6100
CNT	#6148
TMR	#61D6
SOT	#6252
SFRN	#6272
SFR	#62FD
LODNOT	#6479
LOD	#6480
ANDNOT	#648A
AND	#6491
ANDLOD	#649A
ORNOT	#64A6
OR	#64AD
ORLOD	#64B6
OUT	#64C2
MCS	#64D1
MCR	#64E0
SET	#64EB
RES	#64F3
SETR	#64FB
RESR	#6503
LODCNT	#652F
LODNOTCNT	#6535
ANDCNT	#653C
ANDNOTCNT	#6542
ORCNT	#6549
ORNOTCNT	#654F
LODTMR	#6560
LODNOTTMR	#6566
ANDTMR	#656D
ANDNOTTMR	#6573
ORTMR	#657A
ORNOTTMR	#6580
LODR	#6587
LODNOTR	#658D
ANDR	#6594
ANDNOTR	#659A
ORR	#65A1
ORNOTR	#65A7
END	#0
JMP	#1
JEND	#2
DISP	#65D2

KOMUT İSMİ	CALL ADRESİ
FUN1	#67A0
FUN2	#67AD
FUN3	#67BA
FUN4	#67EC
FUN5	#6815
FUN6	#683D
FUN7	#6875
FUN8	#68A8
FUN9	#68E0
FUN10	#68F7
FUN11	#6901
FUN12	#6925
FUN13	#6936
FUN14	#6940
FUN15	#6964
FUN16	#6977
FUN17	#698A
FUN18	#699D
FUN19	#69B7
FUN20	#69E9
FUN21	#69F6

4.2.3. P.D. İŞLETİM PROGRAMI

Alete enerji verildiği anda, işletim programı çalışmaya başlar. Öncelikle bütün sayıcı, zamanlayıcılar, data kayıtçıları, dahili röleleri ve fiziki çıkışların konumlarını sıfırlar (pasif hale getirir). Derlenmiş programın yerleştiği tablo, ilk adresinden başlayarak taranır. Eğer programda sayıcı ve zamanlayıcı varsa, preset değerlerini, numarası verilen sayıcı ve zamanlayıcının ilgili bellek gözüne yerleştirir.

Daha sonra, tablonun ilk adresinden başlayarak, tablodaki argüman değerini (2 iki byte) HL kayıtçısına, data değerini (3. iki byte) BC kayıtçısına koyarak ilk iki byte'taki komut adresine dallandırır. Tabloda herbir satır, altı byte yerleştiğinden, bir sonraki komut için, daha önceki adrese altı ilave edilip, tablodaki yeri bulunur. Tablonun ilk adresi, programın başında IY kayıtçısına yerleştirilmektedir. Tabloyu tarama anında, dallanma adresi için, sıfır değerine rastlanınca, tablonun başına dönülerek işlemeye yeniden başlar. Ayrıca özel zaman röleleri yine bu program içerisinde oluşturulur. Zamanlayıcılar için oluşturulacak zaman temel süresi, Z80 CTC kullanılarak oluşturulmuştur. CTC'nin başlangıçta şartlanması yine işletim programı aracılığıyla yapılmaktadır.



İŞLETİM PROGRAMI ÇALIŞMA SİSTEMİ

P.D İŞLETİM PROGRAMI

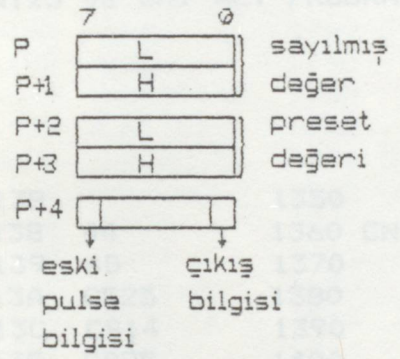
```
8000            10 PP1     EQU  #B000
7000            20 PP2     EQU  #7000
7060            30 PP3     EQU  #7060
71A0            40 PP4     EQU  #71A0
7420            50 PP5     EQU  #7420
74A0            60 PP6     EQU  #74A0
6FFF            70 PPD     EQU  #6FFF
6000            80         ORG  #6000
6000  F3        90 CLEAR   DI
6001  CD0160   100         CALL INIL
6004  FD210080 110 EDIT    LD   IY,PP1
6008  FD7E01   120 LAB1    LD   A,(IY+1)
600B  FE00     130         CP   0
600D  2007     140         JR   NZ,LABB
600F  FD7E00   150         LD   A,(IY+0)
6012  FE00     160         CP   0
6014  283A     170         JR   Z,RUN
6016  FE61     180 LABB    CP   #61
6018  200B     190         JR   NZ,LAB3
601A  FD7E00   200         LD   A,(IY+0)
601D  FE48     210         CP   #48      ; CNT ROUTINE LOW ADDR
601F  280C     220         JR   Z,LAB4
6021  FED6     230         CP   #D6      ; TMR ROUTINE LOW ADDR
6023  281F     240         JR   Z,LAB5
6025  010600   250 LAB3    LD   BC,#6
6028  FD09     260         ADD  IY,BC
602A  C30860   270         JP   LAB1
602D  FD6E02   280 LAB4    LD   L,(IY+2)
6030  FD6603   290         LD   H,(IY+3)
6033  CD3360   300         CALL CNTX5
6036  23       310 LAB6    INC  HL
6037  23       320         INC  HL
603B  FD4E04   330         LD   C,(IY+4)
603B  71       340         LD   (HL),C
603C  23       350         INC  HL
603D  FD4605   360         LD   B,(IY+5)
6040  70       370         LD   (HL),B
6041  C32560   380         JP   LAB3
6044  FD6E02   390 LAB5    LD   L,(IY+2)
6047  FD6603   400         LD   H,(IY+3)
604A  CD4A60   410         CALL TMX5
604D  C33660   420         JP   LAB6
6050  CD5060   430 RUN     CALL INPUT
```

```
6053  3EFB      440        LD   A,#FB    ; CTC HIGH ADD
6055  DBF2      450        IN   A,(#F2)
6057  CB07      460        RLC  A
6059  CB07      470        RLC  A
605B  CB07      480        RLC  A
605D  E608      490        AND  #8
605F  47        500        LD   B,A
6060  3A5970    510        LD   A,(#7059)    ; PP2+59. BYTE FOR FL
6063  E6F7      520        AND  #F7
6065  B0        530        OR   B
6066  325970    540        LD   (#7059),A
6069  FD210080  550        LD   IY,PP1       ; SOURCE PROG MEM ADD
606D  110000    560        LD   DE,#0
6070  D9        570        EXX
6071  210000    580        LD   HL,0
6074  110100    590        LD   DE,1
6077  D9        600        EXX
6078  C38060    610        JP   LAB7
607B  010600    620 LAB8   LD   BC,#6
607E  FD09      630        ADD  IY,BC
6080  FD7E01    640 LAB7   LD   A,(IY+1)
6083  FE00      650        CP   #0
6085  2027      660        JR   NZ,LAB9
6087  FD7E00    670        LD   A,(IY+0)
608A  FE02      680        CP   #02
608C  284C      690        JR   Z,LAB10
608E  FE01      700        CP   #1
6090  2834      710        JR   Z,LAB11
6092  FE00      720        CP   #0
6094  2018      730        JR   NZ,LAB9
6096  3A5770    740        LD   A,(#7057) ; CHECK OF FLAG 703 STA
6099  CB07      750        RLC  A
609B  D2A860    760        JP   NC,LAB13
609E  211970    770        LD   HL,#7019  ; OUTPUTS MEMORY CLEAR
60A1  AF        780        XOR  A
60A2  0610      790        LD   B,16
60A4  77        800 LAB14  LD   (HL),A
60A5  23        810        INC  HL
60A6  10FC      820        DJNZ LAB14
60A8  CDA860    830 LAB13  CALL OUTPT
60AB  C35060    840        JP   RUN
60AE  217B60    850 LAB9   LD   HL,LAB8
60B1  E5        860        PUSH HL
60B2  FD6E00    870        LD   L,(IY+0)
60B5  FD6601    880        LD   H,(IY+1)
60B8  E5        890        PUSH HL
60B9  FD6E02    900        LD   L,(IY+2)
60BC  FD6603    910        LD   H,(IY+3)
60BF  FD4E04    920        LD   C,(IY+4)
60C2  FD4605    930        LD   B,(IY+5)
60C5  C9        940        RET
```

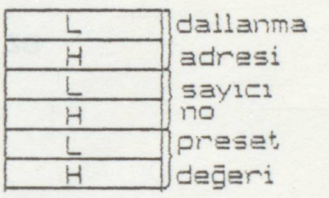
```
60C6  CB43        950 LAB11  BIT  0,E
60C8  2810        960        JR   Z,LAB10
60CA  FD09        970 LAB12  ADD  IY,BC
60CC  FD7E01      980        LD   A,(IY+1)
60CF  FE00        990        CP   #0
60D1  20F7       1000        JR   NZ,LAB12
60D3  FD7E00     1010        LD   A,(IY+0)
60D6  FE02       1020        CP   #2
60D8  20F0       1030        JR   NZ,LAB12
60DA  FD09       1040 LAB10  ADD  IY,BC
60DC  C38060     1050        JP   LAB7
6100             1060        ORG  #6100
6100  210061     1070 INIL   LD   HL,PP2    ;SYSTEM VARIABLES MEM AREA
6103  010005     1080        LD   BC,#500   ;LENGH OF SYSTEM VARIABLES
6106  3600       1090        LD   (HL),#0
6108  54         1100        LD   D,H
6109  5D         1110        LD   E,L
610A  13         1120        INC  DE
610B  EDB0       1130        LDIR
610D  06FB       1140        LD   B,#FB   ;INITIAL OF CTC FOR TIMER
610F  0EF0       1150        LD   C,#F0
6111  3E07       1160        LD   A,#07
6113  ED79       1170        OUT  (C),A
6115  3EFA       1180        LD   A,#FA
6117  ED79       1190        OUT  (C),A   ;CH0
6119  0EF1       1200        LD   C,#F1
611B  3E57       1210        LD   A,#57
611D  ED79       1220        OUT  (C),A
611F  3E64       1230        LD   A,#64
6121  ED79       1240        OUT  (C),A   ;CH1
6123  0EF2       1250        LD   C,#F2
6125  3E57       1260        LD   A,#57
6127  ED79       1270        OUT  (C),A
6129  3EFF       1280        LD   A,#FF
612B  ED79       1290        OUT  (C),A   ;CH2
612D  0EF3       1300        LD   C,#F3
612F  3E57       1310        LD   A,#57
6131  ED79       1320        OUT  (C),A
6133  3EFF       1330        LD   A,#FF
6135  ED79       1340        OUT  (C),A   ;CH3
6137  C9         1350        RET
```

4.2.4. SAYICI VE ZAMANLAYICI YAPISI

Yazılımla oluşturulan, sayıcı ve zamanlayıcıların her birisi bellekte beş byte yerleşmektedir. Sayıcılar için ilk iki byte sayılmış değer (counted volue), ikinci iki byte preset değeri, son byte'ta ise pulse ve çıkış bilgisinin bulunduğu veriler bulunmaktadır.



CNT Bellek yerleşim düzeni



Program tablosunda
CNT yerleşim düzeni

Sayıcılar, bellek haritasında #7060 adresinden itibaren yerleştirilmiştir. İşletim programı tarafından, tablodan alınan sayıcı numarası, HL kayıtçısına konarak CNTX5 alt programına dallanılır. Bu alt program aracılığı ile bellek üzerindeki adresi bulunur. Reset, pulse yukarı aşağı sayım bilgileri, LOD işlemiyle yığıt kayıtçısına konulmuştur. Reset bilgisi "1" ise preset değeri sayılmış değer gözönüne konur, beşinci bayttaki pulse ve çıkış bilgisi sıfırlanır. Daha sonraki çevrimde pulse bilgisi okunur. Sayım için bir darbenin oluştuğunu anlamak için beşinci bayttaki eski pulse bilgisiyle karşılaştırılır. Darbenin geldiği tesbit edilince, yukarı aşağı bilgisine göre, sayılmış değer artılırır ya da azaltılır. Sayılmış değer sıfıra erişince, çıkış bilgisi "1" konumuna set edilir. Sıfıra erişmemişse, çıkış bilgisi "0" konumunda tutulur.

Zamanlayıcıların yapısıda sayıcıların yapısına benzemektedir. Zamanlayıcılarda bellek üzerinde beş byte'a yerleşmiştir. İlk iki byte'ta CTC'den alınan değer (ilk time değeri) vardır. İkinci iki byte'ta ise preset değeri vardır. Son byte ise çıkış durumu ve çalışma mod bilgisi bulunmaktadır.

CNTX5 VE CNT ALT PROGRAMLARI

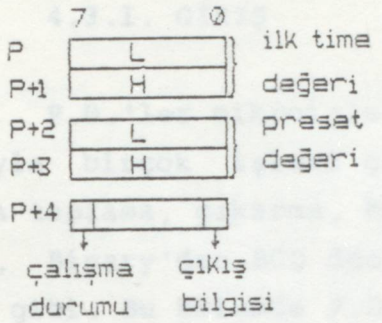
```
6138              1350          ORG   #6138
6138  44          1360 CNTX5    LD    B,H
6139  4D          1370          LD    C,L
613A  CB25        1380          SLA   L
613C  CB14        1390          RL    H
613E  CB25        1400          SLA   L
6140  CB14        1410          RL    H
6142  09          1420          ADD   HL,BC
6143  014361      1430          LD    BC,PP3     ; ILK COUNTER MEMORY
6146  09          1440          ADD   HL,BC
6147  C9          1450          RET
6148  CD3861      1460 CNT      CALL  CNTX5      ; CNT COMMAND
614B  E5          1470          PUSH  HL
614C  DDE1        1480          POP   IX
614E  CB53        1490          BIT   2,E
6150  2813        1500          JR    Z, CNS1
6152  DD7E02      1510          LD    A,(IX+2)
6155  DD7700      1520          LD    (IX+0),A
6158  DD7E03      1530          LD    A,(IX+3)
615B  DD7701      1540          LD    (IX+1),A
615E  DD360400    1550          LD    (IX+4),#0
6162  C39661      1560          JP    CNS3
6165  CB4B        1570 CNS1     BIT   1,E
6167  282D        1580          JR    Z,CNS3
6169  DDCB047E    1590          BIT   7,(IX+4)
616D  2036        1600          JR    NZ,CNS5
616F  CB43        1610          BIT   0,E
6171  2013        1620          JR    NZ,CNS2
6173  DD7E00      1630          LD    A,(IX+0)
6176  D601        1640          SUB   #1
6178  DD7700      1650          LD    (IX+0),A
617B  DD7E01      1660          LD    A,(IX+1)
```

```
617E              1660          ORG  #617E
617E  DE00        1670          SBC  A,#0
6180  DD7701      1680          LD   (IX+1),A
6183  C39661      1690          JP   CNS3
6186  DD7E00      1700  CNS2    LD   A,(IX+0)
6189  C601        1710          ADD  A,#1
618B  DD7700      1720          LD   (IX+0),A
618E  DD7E01      1730          LD   A,(IX+1)
6191  CE00        1740          ADC  A,#0
6193  DD7701      1750          LD   (IX+1),A
6196  CB4B        1760  CNS3    BIT  1,E
6198  2007        1770          JR   NZ,CNS4
619A  DDCB04BE    1780          RES  7,(IX+4)
619E  C3A561      1790          JP   CNS5
61A1  DDCB04FE    1800  CNS4    SET  7,(IX+4)
61A5  DD7E00      1810  CNS5    LD   A,(IX+0)
61A8  FE00        1820          CP   #0
61AA  2007        1830          JR   NZ,CNS6
61AC  DD7E01      1840          LD   A,(IX+1)
61AF  FE00        1850          CP   #0
61B1  2807        1860          JR   Z,CNS7
61B3  DDCB04B6    1870  CNS6    RES  0,(IX+4)
61B7  C3BE61      1880          JP   CNS8
61BA  DDCB04C6    1890  CNS7    SET  0,(IX+4)
61BE  DD7E04      1900  CNS8    LD   A,(IX+4)
61C1  CB1F        1910          RR   A
61C3  CB13        1920          RL   E
61C5  C9          1930          RET
```

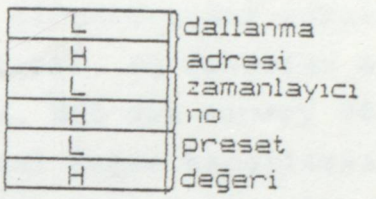
TMX5 VE TMR ALT PROGRAMLARI

```
61C6  44          1940  TMX5    LD   B,H
61C7  4D          1950          LD   C,L
61C8  CB25        1960          SLA  L
61CA  CB14        1970          RL   H
61CC  CB25        1980          SLA  L
61CE  CB14        1990          RL   H
61D0  09          2000          ADD  HL,BC
61D1  01D161      2010          LD   BC,PP4    ;ILK TIMER MEMORY
61D4  09          2020          ADD  HL,BC
61D5  C9          2030          RET
61D6  CDC661      2040  TMR     CALL TMX5      ; TMR  COMMAND
61D9  E5          2050          PUSH HL
61DA  DDE1        2060          POP  IX
61DC  CB43        2070          BIT  0,E
61DE  2866        2080          JR   Z,TMRS5
61E0  DDCB047E    2090          BIT  7,(IX+4)
61E4  202A        2100          JR   NZ,TMRS2
```

```
61E6  06FB       2110          LD   B,#FB
61E8  0EF3       2120          LD   C,#F3
61EA  ED78       2130          IN   A,(C)
61EC  DD7701     2140          LD   (IX+1),A
61EF  0EF2       2150          LD   C,#F2
61F1  ED78       2160          IN   A,(C)
61F3  DD7700     2170          LD   (IX+0),A
61F6  0EF3       2180          LD   C,#F3
61F8  ED78       2190          IN   A,(C)
61FA  DDBE01     2200          CP   (IX+1)
61FD  280A       2210          JR   Z,TMRS1
61FF  DD7701     2220          LD   (IX+1),A
6202  0EF2       2230          LD   C,#F2
6204  ED78       2240          IN   A,(C)
6206  DD7700     2250          LD   (IX+0),A
6209  DDCB04FE   2260 TMRS1    SET  7,(IX+4)
620D  C31662     2270          JP   TMRS3
6210  DDCB0446   2280 TMRS2    BIT  0,(IX+4)
6214  2034       2290          JR   NZ,TMRS6
6216  06FB       2300 TMRS3    LD   B,#FB
6218  0EF2       2310          LD   C,#F2
621A  ED78       2320          IN   A,(C)
621C  0EF3       2330          LD   C,#F3
621E  ED40       2340          IN   B,(C)
6220  4F         2350          LD   C,A
6221  DD6E00     2360          LD   L,(IX+0)
6224  DD6601     2370          LD   H,(IX+1)
6227  A7         2380          AND  A
6228  ED42       2390          SBC  HL,BC
622A  DD4E02     2400          LD   C,(IX+2)
622D  DD4603     2410          LD   B,(IX+3)
6230  A7         2420          AND  A
6231  ED42       2430          SBC  HL,BC
6233  280A       2440          JR   Z,TMRS4
6235  D23F62     2450          JP   NC,TMRS4
6238  DDCB0486   2460          RES  0,(IX+4)
623C  C34A62     2470          JP   TMRS6
623F  DDCB04C6   2480 TMRS4    SET  0,(IX+4)
6243  C34A62     2490          JP   TMRS6
6246  DD360400   2500 TMRS5    LD   (IX+4),#0
624A  DD7E04     2510 TMRS6    LD   A,(IX+4)
624D  CB1F       2520          RR   A
624F  CB13       2530          RL   E
6251  C9         2540          RET
```



TMR Bellek yerleşim düzeni



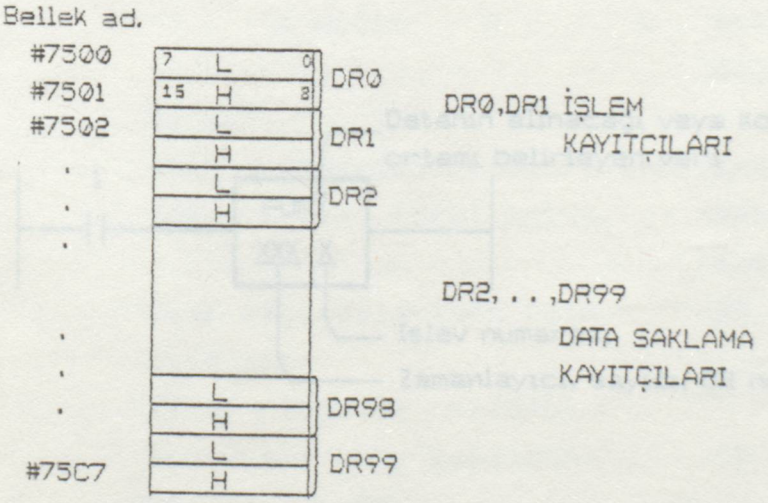
Program tablosunda
TMR yerleşim düzeni

Tabloda, numarası bulunan zamanlayıcının, bellekteki yerini bulmak için TMX5 alt programı kullanılır. Zamanlayıcı alt progrmanına dallanıldığında, işletim programı tarafından zamanlayıcı numarası, HL Kayıtçısı içerisine konulmuştur. Bundan yararlanılarak, ilgili zamanlayıcının bellekteki yeri tesbit edilmiş olur. Zamanlayıcı komutundan önce bulunan kontak açıksa, zamanlama işlemi başlamaz. Ve zamanlayıcı çıkışı pasiftir. (Çıkış bilgisi sıfırdır). DE kayıtcısının son bitinden, kontağın kapandığı tesbit edilince, CTC'den alınan değer, ilk time volue için ayrılan ilk iki byte'a, ikinci iki byte'a ise preset değeri yerleştirilir. Her çevrimde, CTC'den alınan değer, ilk time değerinden çıkarılır. Fark preset değerine espit ya da büyükse çıkış aktif konuma çekilir. Şayet, zamanlayıcıdan önce gelen kontak açılırsa, zamanlama işlemi yarıda kesilecek, çıkış bilgisi sıfırda kalacaktır. Kontak kapandığı anda işlemler yeniden başlayacaktır. Zamanlayıcılar için ayrılan belleğin başlangıç adresi, Hex 71AO en son adres ise Hex 741F dir. Bu aralıkta 128 zamanlayıcı tanımlıdır.

4.3. P.D. ÖZEL İŞLEV KOMUTLARI

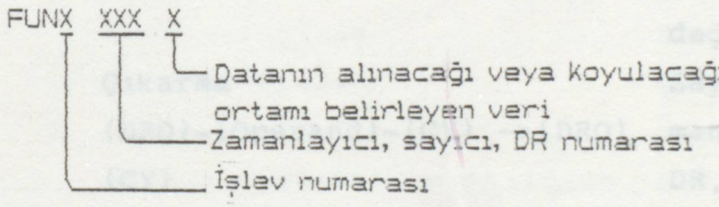
4.3.1. GİRİŞ

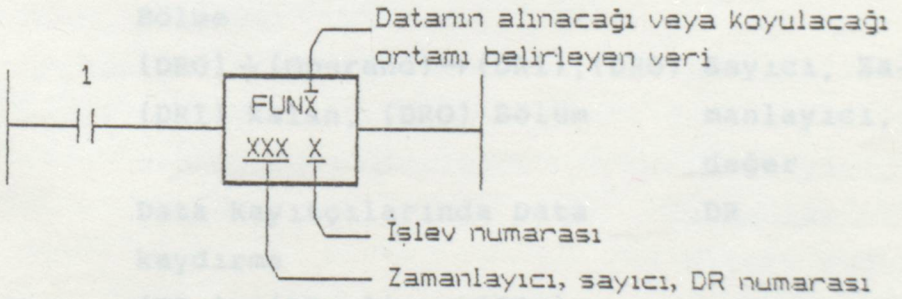
P.D.'ler mikroişlemci temelli olduklarından yazılım yoluyla birçok işlevi gerçekleyebiliriz. Bu işlevler arasında toplama, çıkarma, bölme, çarpma, BCD'den-binary dönüşümü, Binary'den-BCD dönüşümü, sayısal değer karşılaştırma v.b gibi. Bu bölümde P.D. için oluşturulmuş özel işlev komutları (Function comands) teker teker anlatılacaktır. Hesaplamaların yapılması, sonuçların depolanması ya da data saklanması için bellekte belirli bir bölge ayrılmıştır. Bu ayrılan bölgede, 0'dan 99'a kadar numaralandırılmış Veri Kayıtçıları (Data registers) bulunmaktadır. Herbir veri kayıtçısı 16 bit'liktir. Dolayısıyla bellekte iki byte yer tutmaktadırlar. İşlemler çoğunlukla sıfır nolu veri kayıtçısı (DRO diye gösterilecektir) ve DR1 üzerinde yapılmaktadır. DRO ve DR1 işlemlerin yapıldığı, diğer kayıtçıları ise dataların saklandığı bölge olarak isimlendirebiliriz.



DR'lerin bellekte yerleşme düzeni

Veri kayıtçıları üzerinde aritmetik işlemler yapıldığı gibi, farklı işlevlerde gerçeklenmektedir. Kayıtçılar, arasında data kaydırma, zamanlayıcı, sayıcı değerlerini DRO' daki veriyle karşılaştırma v.b gibi. Her özel işlev komutu üç veriyle belirlenir. FUN1, FUN2, yazılımında, numaralar hangi işlevin yapılacağı gösterir. Yapılacak işlevin datasının alınacağı ortamın (zamanlayıcı, sayıcı, data kayıtçısı, dahili röle, fiziki giriş ya da çıkışlar) numarasını ikinci veri, ortamın türünü belirleyecek bilgi ise üçüncü veriyle elde edilir.





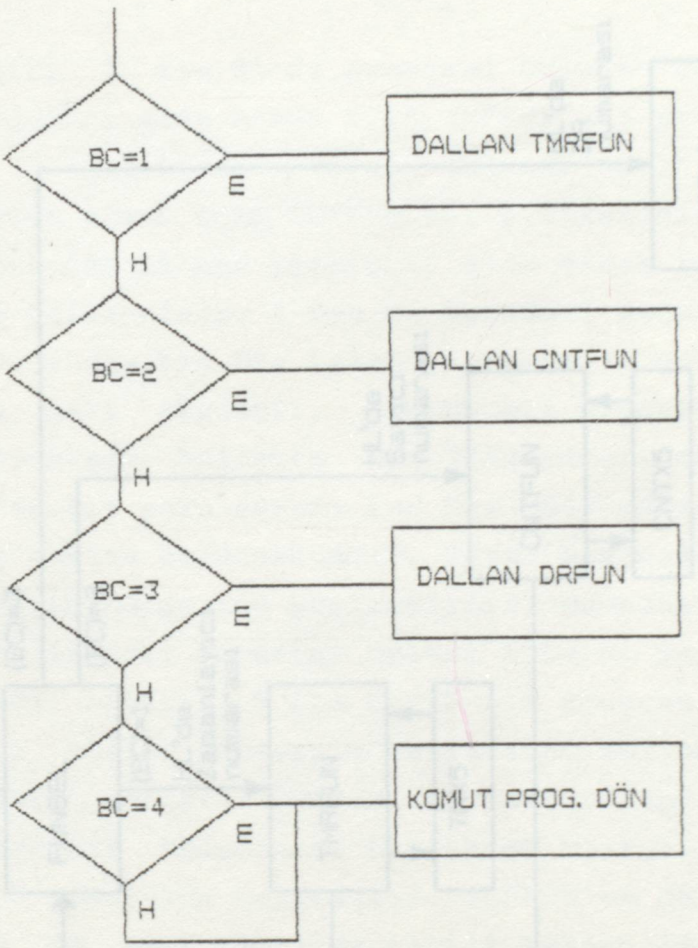
İşlev komutlarının Grafik olarak gösterimi

Aşağıda işlev numaraları ve görevleri gösterilmiştir.

İŞLEV NO	İŞLEVİ	ETKİN OLDUĞU ORTAM
FUN 1	DRO içindeki ikili sayıyı BCD koduna çevirme	-
FUN 2	DRO içindeki BCD kodundaki sayıyı ikili sayı sistemine çevirme	-
FUN 3	4-Dijit karşılaştırma (DRO) $\gtreqless$ (operand) sonuca göre > → 710 = → 711 < → 712 Dahili rölelerine set etmek	Zamanlayıcı sayıcı Değer
FUN 4	Toplama (DRO)+(Operand)+(CY) → (DRO), (CY)	Sayıcı, zamanlayıcı DR, değer
FUN 5	Çıkarma (DRO)-(Operand)-(CY) → (DRO), (CY)	Sayıcı, Zamanlayıcı DR, değer
FUN 6	Çarpma (DRO)x(Operand) → (DR1), (DRO) 4 diğit H L	Sayıcı, Zamanlayıcı DR, değer
FUN 7	Bölme (DRO) ÷ (Operand) → (DR1),(DRO) (DRI) kalan, (DRO) Bölüm	Sayıcı, Zamanlayıcı, DR değer
FUN 8	Data Kayıtçılarında Data kaydırma (DRm) → (DRm+1) ··· → (DRn)	DR
FUN 9	BCD digit sola kaydırma (DRI DRO) ← 00	DR
FUN 10	DRO ← Data yükleme 16 bit	Zamanlayıcı, DR, değer

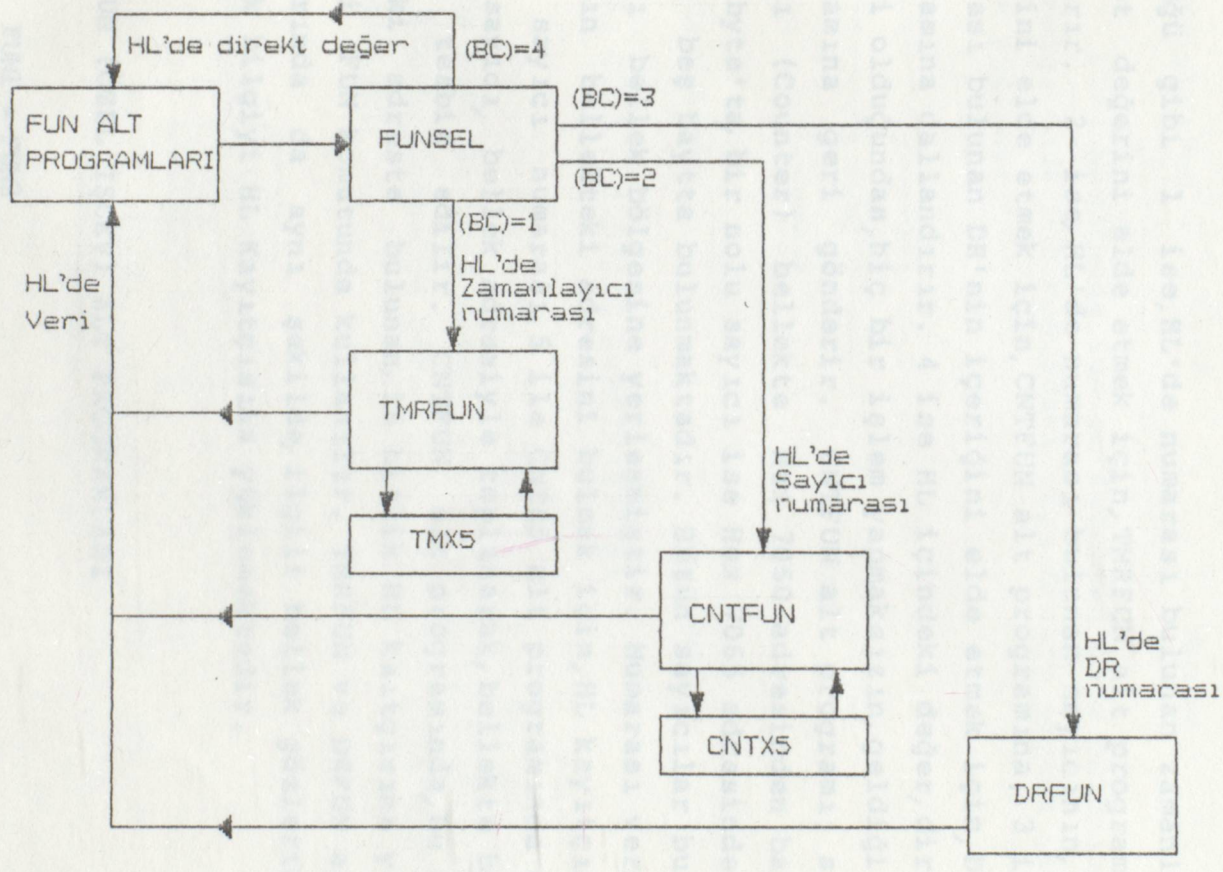
İŞLEV NO	İŞLEVİ	ETKİN OLDUĞU ORTAM
FUN 11	DRO ← Data yükleme 8 bit	girişler
FUN 12	DRO ← Data yükleme indirekt (DRO) ← (Operand + (DR1))	DR
FUN 13	DRI Data yükleme 16 bit	Zamanlayıcı, sayıcı DR, değer
FUN 14	DRI ← Data yükleme 8 bit	Girişler
FUN 15	(DRO) ← (DRO+1)	DR
FUN 16	(DR1) ← (DRI-1)	DR
FUN 17	(DRO) → (Operand)	Zamanlayıcı Sayıcı
FUN 18	Sayıcı değerinin eşitliğini tesbit	Sayıcı
FUN 19	Sayıcı değerinin büyüklüğünü tesbit	Sayıcı
FUN 20	(DRO) → (DR) 16 bit yükleme	DR
FUN 21	(DRO) → (DR) 16 bit yükleme	DR
FUN 22	Harici data display yükleme	DR, sayıcı, zamanlayıcı direkt değer

4.3.2. FUN KOMUTLARINDA KULLANILAN BAZI ALT PROGRAMLAR



FUNSEL AKIŞ DİYAGRAMI

Yapılacak işlevin ne alacağı, datanın nereden alınıp nereye konacağı, FUN komutlarının yazılımından anlaşılmaktadır. Örneğin FUN1, 20 3 yazılımında FUN1, işlevin ne olacağı, 20 datanın alınacağı ya da konacağı ortamın numarası, 3 ise ortamın türünü (Zamanlayıcı, sayıcı, DR, ya da direkt değer) belirler. FUNSEL programına dallanıldığında, BC kayıtçısında ortamın türünü belirleyecek bilgi, HL Kayıtçısında ise datanın alınacağı ya da konacağı ortamın numarası bulunmaktadır. FUNSEL alt programı, BC kayıtçısının içindeki değeri 1,2,3,4 sayılarıyla karşılaştırılır. Şemada gö-



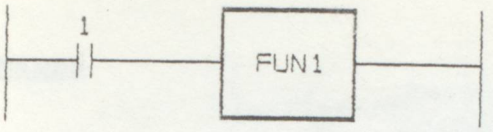
FUNSEL ÇALIŞMA SİSTEMİ

rüldüğü gibi 1 ise, HL'de numarası bulunan zamanlayıcının preset değerini elde etmek için, TMRFUN alt programına dallandırır. 2 ise, HL'de numarası bulunan sayıcının, sayılmış değerini elde etmek için, CNTFUN alt programına, 3 ise, HL'de numarası bulunan DR'nin içeriğini elde etmek için, DRFUN alt programına dallandırır. 4 ise HL içindeki değer, direkt data değeri olduğundan, hiç bir işlem yapmaksızın, geldiği FUN alt programına geri gönderir. CNTFUN alt programı; sıfırında sayıcı (Counter) bellekte Hex 7060 adresinden başlayarak beş byte'ta, bir nolu sayıcı ise Hex 7065 adresinden başlayarak beş baytta bulunmaktadır. Bütün sayıcılar bu düzenle sayıcı bellek bölgesine yerleşmiştir. Numarası verilen sayıcının bellekteki adresini bulmak için, HL kayıtçısı içindeki sayıcı numarası, 5 ile CNTX5 alt programında çarpılıp, ilk sayıcı, bellek adresiyle toplanarak, bellekte bulunduğu adres tesbit edilir. CNTFUN alt programında, bu adres ve sonraki adreste bulunan, 16 bitlik HL kaıtçısına yüklenir. İlgili FUN komutunda kullanılır. TMRFUN ve DRFUN alt programlarında da aynı şekilde, ilgili bellek gözlerindeki 16 bitlik bilgiyi HL Kayıtçısına yüklemektedir.

4.4 FUN (ÖZEL İŞLEV) ALT PROGRAMLARI

4.4.1 FUN1 - FUN2

Fun 1 komutunda bu komuttan önce bulunan kontak kapandığı anda DRO içindeki 16-bitlik binary sayı BCD (Binary coded decimal) sayıya çevrilerek DRO'a yüklenmektedir.

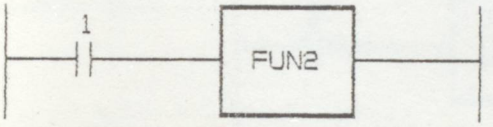



```
LOD    1
FUN1
```

Grafik gösterimi

Yazılımı

FUN2 komutunda ise bu komuttan önce bulunan kontak kapandığı anda BCD kodundaki sayı binary sayıya çevrilerek DRO'a yüklenir.



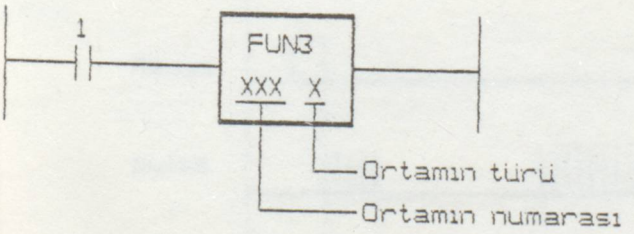

```
LOD    1
FUN2
```

Grafik gösterimi

Yazılımı

4.4.2. FUN3 - 4 DİJİT KARŞILAŞTIRMA

Bu komutta, DRO içindeki sayı ile sayıcı, zamanlayıcı, DR içeriği veya direkt değer ile karşılaştırılarak, büyük, küçük, durumuna göre, 710, 711, 712 nolu özel röleler set edilir.



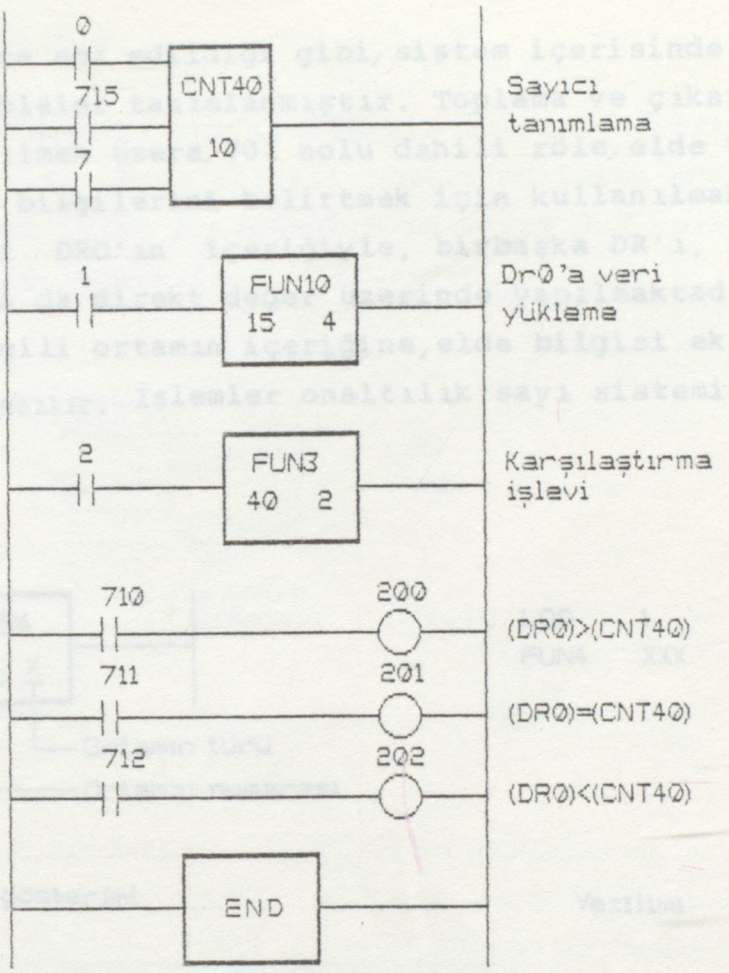

```
LOD    1
FUN3   XXX   X
```

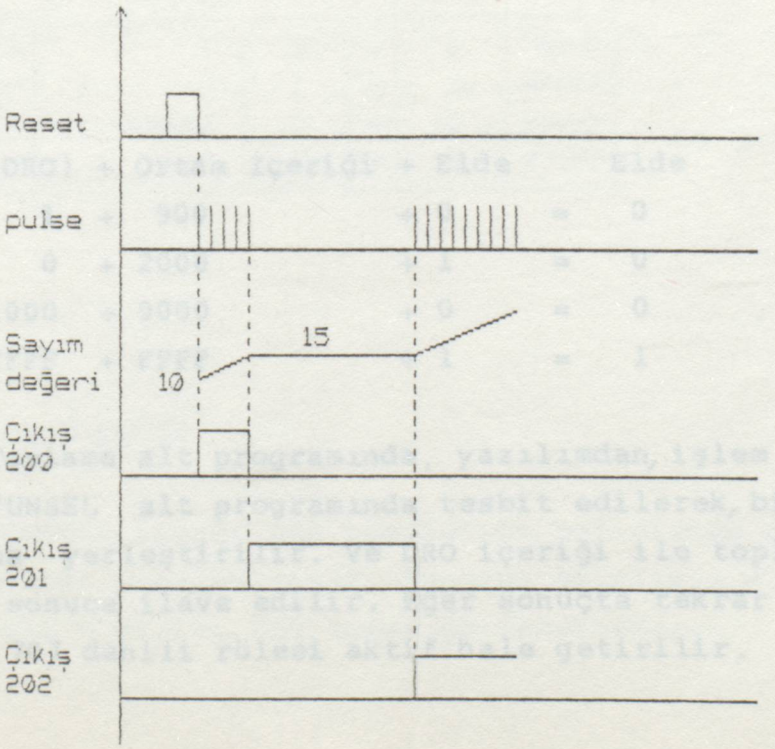
Grafik gösterimi

Yazılımı

DRO içindeki sayısal değer, ortamdaki değerden büyük ise 710, eşit ise 711, küçük ise 712 nolu özel röleler aktif duruma getirilir.

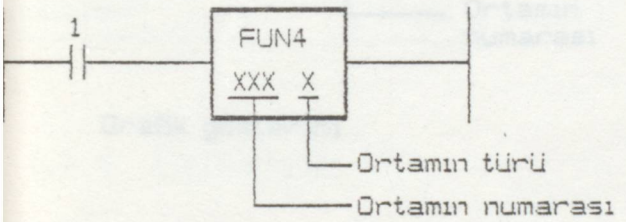
ÖRNEK


0
715
7
CNT40
10
Sayıcı tanımlama
1
FUN10
15 4
Dr0'a veri yükleme
2
FUN3
40 2
Karşılaştırma işlevi
710
200
(DR0)>(CNT40)
711
201
(DR0)=(CNT40)
712
202
(DR0)<(CNT40)
END


Reset
pulse
Sayım
değeri
10
15
Çıkış
200
Çıkış
201
Çıkış
202

4.4.3. FUN 4 TOPLAMA

Daha önce söz edildiği gibi, sistem içerisinde bazı özel dahili röleler tanımlanmıştır. Toplama ve çıkarma işleminde kullanılmak üzere, 707 nolu dahili röle, elde ve borç (Carry-Borrow) bilgilerini belirtmek için kullanılmaktadır. Toplama işlemi DRO'ın içeriğiyle, birbaşka DR'ı, sayıcı, zamanlayıcı ya da direkt değer üzerinde yapılmaktadır. DRO içeriği ile ilgili ortamın içeriğine, elde bilgisi eklenerek sonuç DOR'a yazılır. İşlemler onaltılık sayı sisteminde yapılmaktadır.



Grafik gösterimi

```
LOD    1
FUN4   XXX   X
```

Yazılımı

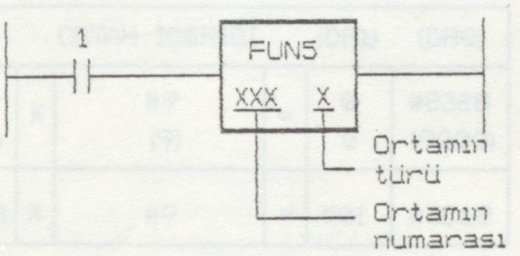
ÖRNEK

(DRO)	+	Ortam İçeriği	+	Elde		Elde	(DRO)
1	+	900	+	0	=	0	901
0	+	2000	+	1	=	0	2001
1000	+	9000	+	0	=	0	A000
FFFF	+	FFFF	+	1	=	1	FFFF

Toplama alt programında, yazılımdan, işlem yapılacak ortam FUNSEL alt programında tesbit edilerek, bilgi HL kayıtçısına yerleştirilir. Ve DRO içeriği ile toplanır. Elde bilgisi sonuca ilave edilir. Eğer sonuçta tekrar elde oluşuyorsa 707 dahili rölesi aktif hale getirilir.

4.4.4. FUN5 ÇIKARMA

FUN5 komutunda, DRO içeriğiyle belirlenen ortamın içeriği çıkarılır. Borç bilgisi tesbit edilerek sonuçtan çıkarılır. Eğer işlem sonucunda borç varsa 707, nolu dahili röle aktif hale getirilir.



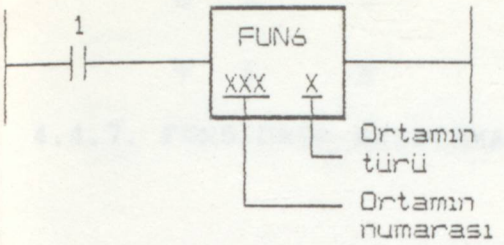

```
LOD     1
FUN5    XXX     X
```

Grafik gösterimi Yazılımı

(DRØ)		ORTAM İÇERİĞİ	BORC		BORC	(DRØ)
#9ØØ	-	1	Ø	=	Ø	#8FF
Ø	-	Ø	1	=	1	#FFFF
#AØØØ	-	#9ØØØ	Ø	=	Ø	#1ØØØ
Ø	-	1	Ø	=	1	#FFFF

4.4.5. FUN 6 ÇARPMA

Çarpma işlemi, DRO içeriğiyle, belirlenen ortam içeriği arasında yapılmaktadır. Sonucun ağırlıklı 4 diğiti DRI, ağırlıksız 4 digiti, DRO içine yazılmaktadır.




```
LOD    1
FUN6   XXX    X
```

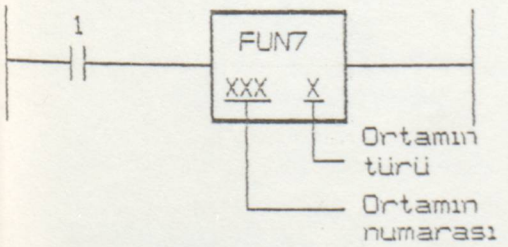
Grafik gösterimi

Yazılımı

(DR0)		ORTAM İÇERİĞİ		(DR1)	(DR0)
#3E8 (1000)	*	#9 (9)	=	0 0	#2328 (9000)
#2328	*	#9	=	#01	#3C68

4.4.6. FUN7 BÖLME

Bölme işlemi, DRO içeriği ile, belirlenen ortam içeriği arasında yapılmaktadır. Bölünen = DRO, Bölen = belirlenen ortam, Bölüm = DRO, kalan - DRI Bölme işlemi sonucu DRO yazılmaktadır. Bütün işlemlerde DR'ların içeriği tam sayı olarak değerlendirilmektedir. Bu sebepten Bölme tamsayı olarak yapılıp kalan DRI'e yazılmaktadır. Bölme Programında, FUNSEL programı yardımı ile ortam tesbiti yapılıp, DRO, HL kayıtçısı içindeki değere bölünür.




```
LOD    1
FUN7   XXX    X
```

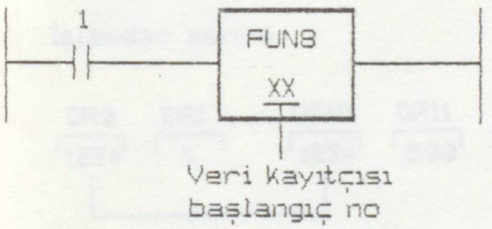
Grafik gösterimi

Yazılımı

(DRO)	ORTAM İÇERİĞİ	(DRØ)	(DR1)
8	÷ 2	4	Ø
9	÷ 2	4	1

4.4.7. FUN8 DATA KAYDIRMA

Bu komut yazılımında, numarası belirlenen DR'dan başlayıp, DRI içeriğiyle belirlenen uzunluk ile sonlanan kayıtçı bölgesi üzerinde, dataların kaydırılması söz konusudur. ilk numaralı kayıtçı içeriğine DRO içeriği aktarılmaktadır.



Grafik gösterimi

```
LOD   1
FUN8  XX
```

Yazılımı

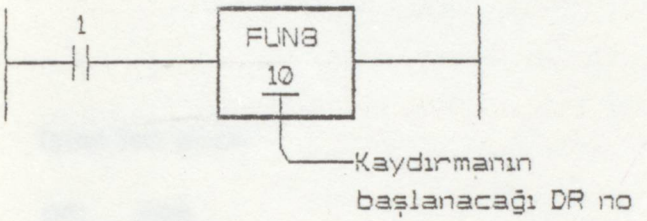
DRO: Giriş bilgisi
DRI: Kaydırma Uzunluğu

ÖRNEK

İşlemden önce

DR0	DR1	...	DR10	DR11	DR12	DR13	....	DR14
1234	4		500	600	700	800		

İşlem

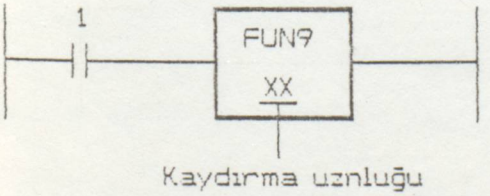


İşlemden sonra

DR0	DR1	...	DR10	DR11	DR12	DR13	....	DR14
1234	4		1234	500	600	700		
						800		

4.4.8. FUN9 BCD DİJİT SOLA KAYDIRMA

FUN9 komutunda, DRO, DRI üzerinde sola dijit kaydırma işlemi yapılmaktadır. Sola kaç dijik kaydırılacağı komut yazılımında verilmektedir.




```
LOD    1
FUN9   XX
```

Grafik gösterimi Yazılımı

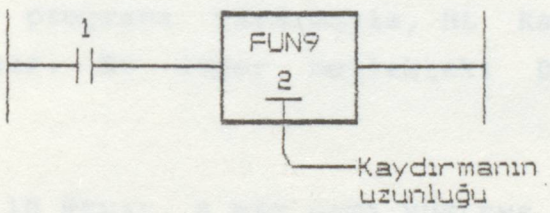
Kaydırma uzunluğu, 1 ile 7 arasında olabilir. Belirlenen uzunluk kadar, sıfır bilgisi DRO içine konur. DRO'dan taşan bilgiler DRI içine aktarılmaktadır.

ÖRNEK

İşlemden önce

DR1 DR0
0001 2345

İşlem



LOD 1
FUN9 2

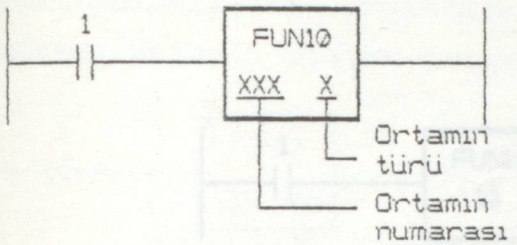
İşlemden sonra

DR1 DR0
0123 4500
00 00

LOD 1
FUN8 10

4.4.9 FUN 10 16-BİT DATA YÜKLEME (DRO'A)

Bu komut ile, DRO'a 16-bit data yüklemek sözkonusudur. Yüklenecek bilgi, sayıcı, zamanlayıcı, DR Dataları veya direk data değeri olabilir.




```
LOD     1
FUN10   XXX     X
```

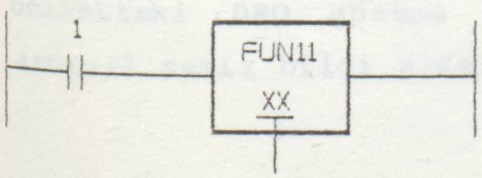
Grafik gösterimi

Yazılımı

Bu programda yazılım ile belirlenen ortamdan, FUNSEL alt programı yardımıyla, HL Kayıtçısı içinde data değeri alınır. Bu değer bellekteki DRO gözönü yerleştirilir.

4.4.10 FUN11 8 BİT DATA YÜKLEME

FUN11 komutta komut yazılımıyla verilen giriş numarasından başlayarak, ardışıl sekiz fizik giriş, dahili röle veya çıkış DRO'ın ağırlıksız sekiz bitine yerleştirilir. Bu işlem sonucunda DRO'ın ağırlıklı sekiz biti aynen korunmaktadır. DRO'a 16 giriş yüklenmek istenirse, önce ağırlıksız 8 bit yüklenir FUN9 aracılığıyla 2 dijit kaydırılıp daha sonra ağırlıklı 8-bit yüklenir.



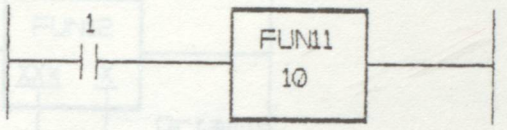

```
LOD     1
FUN11   XXX     X
```

Yüklenecek ardışıl fiziki giriş-çikiş veya dahili rölenin ilk numarası

Grafik gösterimi

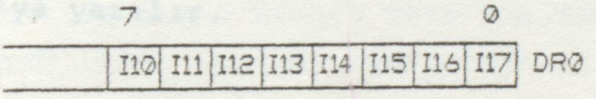
Yazılımı

ÖRNEK




```
LOD    1
FUN11  10
```

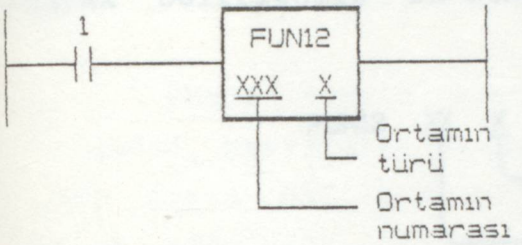
1 Nolu kontak kapandıktan sonra 10 nolu girişten başlayarak ardışıl 8 giriş DRO'ın ağırlıksız sekiz bitine yerleşir.



Fiziki girişler, dahili röleler, fiziki çıkışlar bellekte Hex 7000 adresinden başlayarak yerleştirilmiştir. INPSB programı aracılığıyla, bu bölge içerisinde herhangi fiziki giriş, çıkışı ya da dahili röle durumu tesbit edilebilir. Bu alt program; compile tablosundan, işletim programı aracılığıyla, HL Kayıtcısı içine konmuş fizik giriş-çıkış ya da dahili röle numarasıyla INPSB alt programına dallandırılır. INPSB programı dönüşünde, ilgili ... konumu A kayıtçısının ağırlıksız son bitinde elde edilir. Bu bilgi bellekteki DRO gözüne yazılır. Bu işlem HL artırılarak, ardışıl sekiz bilgi elde edilinceye kadar devam ettirir.

4.4. 11. FUN12 16 BIT DOLAYLI YOLLA DATA YÜKLEME

Bu komut DRO'a, sayıcı, zamanlayıcı ve DR 16 bit data yüklemekte kullanılmaktadır. Datanın alınacağı ortamın numarası dolaylı olarak belirlenir.



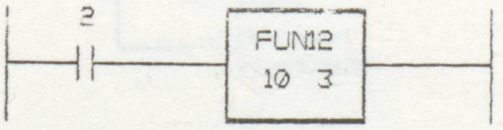

```
LOD     1
FUN12   XXX     X
```

Grafik gösterimi

Yazılımı

Burada, datanın alınacağı ortamın numarası, yazılımda verilen değer ile DRI içeriği toplanarak elde edilir. Datanın alınacağı ortamın türüne göre numarası dolaylı olarak belirlenmiş sayıcı, zamanlayıcı ya da DR'dan alınan 16 bitlik veri DRO'ya yazılır.

ÖRNEK

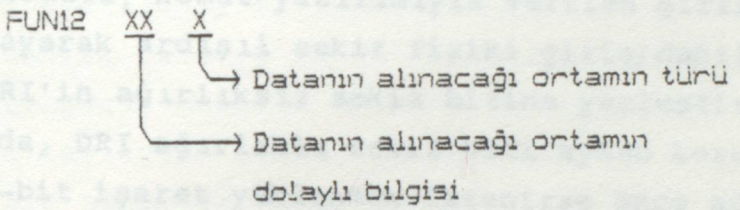



```
LOD     2
FUN12   10      3
```

10+(DR1) = 10+5 = 15

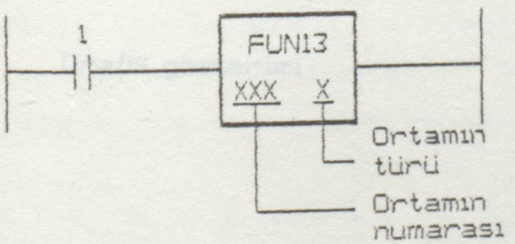
DR0	DR1	DR15	
59	5	500	İşlemden önce
500	5	500	İşlemden sonra

Yukarıdaki örnekte, 2 numaralı kontak kapandığı zaman, yazılımda belirlenen dolaylı ortam numarası 10, ile DR1'in içeriği toplanarak, gerçek ortam numarası belirlenir. Ortamın türünü belirleyecek bilgi 3 olarak verildiğinden, bilginin DR'lerden alınacağını belirlemiştir. Numarası dolaylı olarak belirlenmiş DR'den, 16 bitlik bilgi DRO'a yazılır.



4.4.12 FUN13 16-BİT DATA (DRI'E) YÜKLEME

Bu komut ile, DR1'e 16-bit data yüklemek sözkonusudur. Yüklenecek bilgi sayıcı, zamanlayıcı, DR değerleri ya da direkt data değeri olabilir.




```
LOD     1
FUN13   XXX     X
```

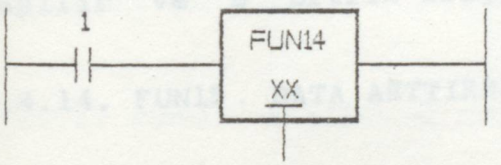
Grafik gösterimi

Yazılımı

Bu programda, yazılım ile belirlenen ortamdan, FUNSEL alt programı yardımıyla HL kayıtçısı içinde data değeri alınır. Bu değer HL kayıtçısından bellekteki DR1 gözüne yerleştirilir.

4.4.13. FUN14. 8 BİT DATA YÜKLEME

FUN14 komutu, komut yazılımıyla verilen giriş numarasından, başlayarak ardışıl sekiz fiziki giriş, dahili röle veya çıkış DRI'in ağırlıksız sekiz bitine yerleştirir. Bu işlem sonucunda, DRI ağırlıklı sekiz biti aynen korumaktadır. DRI'e 16-bit işaret yüklenmek istenirse önce ağırlıksız 8 bit yüklenir. FUN9 aracılığıyla 2 diğit kaydırılıp daha sonra ağırlıklı 8 bit yüklenir.

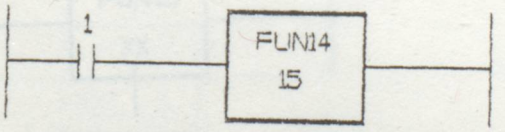



```
LOD     1
FUN14   XX
```

Yüklenecek ardışıl fiziki giriş-çikiş veya dahili rölenin ilk numarası

Grafik gösterimi

Yazılımı




```
LOD     1
FUN14   15
```

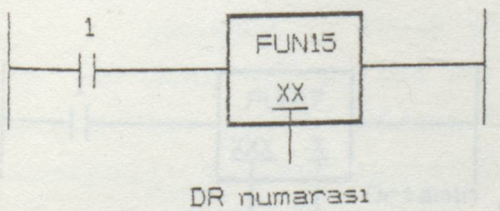
1 Nolu kontak kapandıktan sonra, 15 nolu girişten başlayarak ardışıl sekiz işaret DRO'ın ağırlıksız sekiz bitine yerleştirilir.

7							0	
I15	I16	I17	I18	I19	I20	I21	I22	DR1

Fiziki girişler, dahili röleler, fizik çıkışlar bellekte Hex 7000 adresinden başlayarak yerleştirilmiştir. INPSB programı aracılığıyla, bu bölge içerisinde herhangi fiziki giriş, dahili röle ya da fiziki çıkışın durumu elde edilir. Bu alt program; compile tablosundan işletim programı aracılığıyla HL kayıtçısı içine konmuş fiziki giriş, çıkış ya da dahili röle numarasıyla, INPSB alt programına dallandırır. INPSB programı dönüşünde, ilgili numaranın konumu, A kayıtçısının ağırlıksız bitinde elde edilir. HL bir arttırılarak, bu işlem ardışıl sekiz bilgi elde edilinceye kadar yapılır ve 8 bitlik bilgi bellekteki DRI gözüne yazılır.

4.4.14. FUN15 DATA ARTTIRMA

FUN15 komutu, komut yazılımında numarası belirlenmiş data kayıtçısının içeriğine bir ilave eder ve sonucu aynı data kayıtçısına yazar. Eğer kayıtçı içeriği max değerde ise, bu komut kullanıldıktan sonra kayıtçı içeriği sıfır olur. Elde işaretini temsil eden 707 numaraları dahili röle bu sonuçtan etkilenmez.



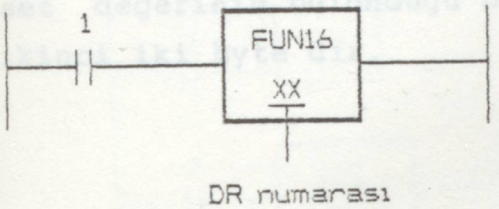
Grafik gösterimi

```
LOD     1
FUN15   XX
```

Yazılımı

4.4.15 FUN16 . DATA AZALTMA

FUN16 komutu, komut yazılımında numarası belirlenmiş data kayıtçısının içeriğini bir azaltır ve sonucu aynı data kayatçısına yazar. Eğer kayıtçı içeriği sıfır değerinde ise, bu komut kullanıldıktan sonra kayıtçı içreiği max değere (Hex FFFF) set olur. Borç işaretini temsil eden 707 numaralı dahili röle bu sonuçtan etkilenmez.

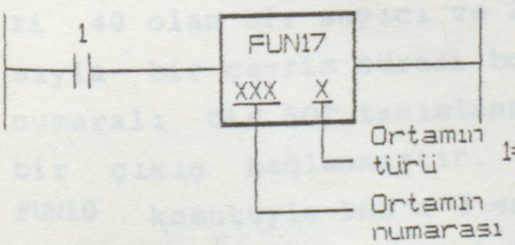



```
LOD     1
FUN16   XX
```

Grafik gösterimi Yazılımı

4.4.16 FUN17 SAYICI VE ZAMANLAYICILARA DATA YÜKLEME

Bu komut ile, sayıcı ve zamanlayıcıların pereset değerlerinin bulunduğu bellek gözöne DRO'ın içeriği yerleştirilir. DRO'ın içeriği, sayıcımı yoksa zamanlayıcımı preset değerine aktarılacağı, bu komut yazılımında belirlenir. Daha önceki özel işlev komutlarında olduğu gibi, ortamın türünü belirleyen veri 1 ise zamanlayıcı, 2 ise sayıcı olduğunu işaret eder.

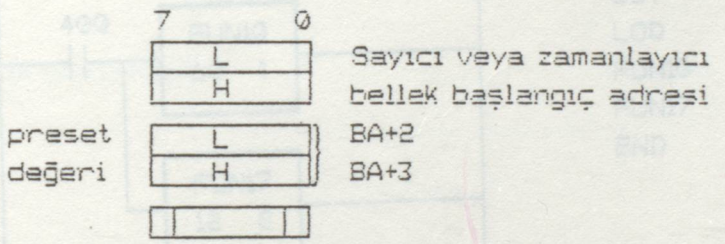


1=Zamanlayıcı , 2=Sayıcı

```
LOD     1
FUN17   XXX   X
```

Grafik gösterimi Yazılımı

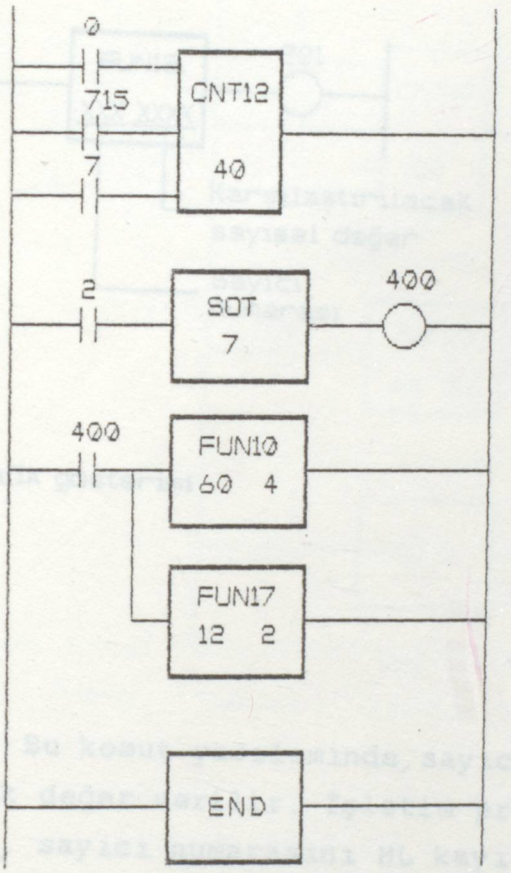
Bu alt programda komut yazılımından, derlenmiş, program tablosuna yerleşmiş ortam türü, FUNSL alt programında tesbit edilir. FUNSL alt programı, BC kayıtçısının içeriği 1 ise TMX5 alt programına, 2 ise CNTX5 alt programına dallandırır. Bu alt programda, HL kayıtçısı içindeki sayıcı ya da zamanlayıcı numarasından, bellekteki sayıcı ya da zamanlayıcı adresi tesbit edilir. Bu adresler, numarası belirlenen sayıcı ya da zamanlayıcının bulunduğu ilk adreslerdir. Preset değerinin bulunduğu belle gözü, bu adresten sonra gelen ikinci iki byte dır.



DRO'ın içeriği, tesbit edilmiş, preset değeri bellek gözüne yerleştirilir.

Aşağıdaki örnek programda, numarası 12, preset değeri 40 olan bir sayıcı ve 2 nolu fiziki girişin aktif olmasıyla bir çevrim süresi boyunca aktif çıkış vermesi için 7 numaralı bir SOT tanımlanmıştır. SOT çıkışına, 400 numaralı bir çıkış bağlanmıştır. 400 numaralı kontak aktif olunca FUN10 komutuyla DRO'a 7 sayısal değeri yüklenir. Aynı kontağa bağlı FUN17 özel işlevinde 12 numaralı sayıcıya DRO'ın içeriği yüklenir. 12 nolu sayıcıya reset işareti gelince, sayım değeri, preset değerine yüklenen DRO'ın içeriğinden başlar.

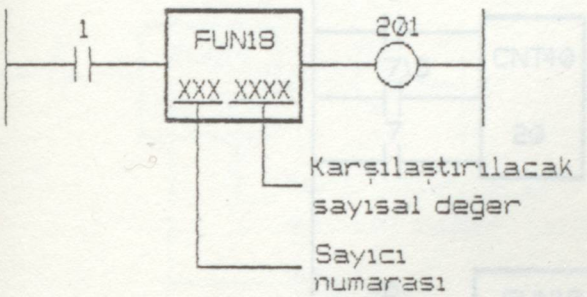
ÖRNEK




```
LOD     0
LOD     715
LOD     7
CNT     12      40
LOD     2
SOT     7
OUT     400
LOD     400
FUN10   60      4
FUN17   12      2
END
```

4.4.17 FUN 18 SAYICILARDA SAYILMIŞ DEĞER EŞİTLİK KONTROLÜ

Bu komutla, herhangi bir sayıcının sayılmış değeri, komut yazılımında verilen, sayısal değerle eşitliği kontrol edilir.



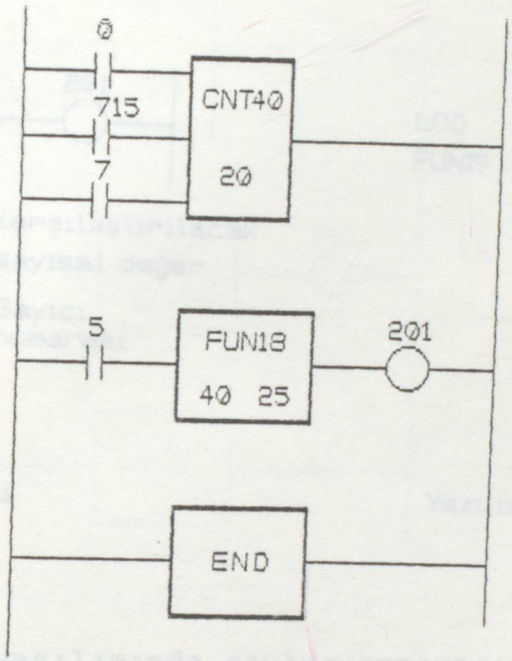

```
LOD     1
FUN18   XXX     X
```

Grafik gösterimi

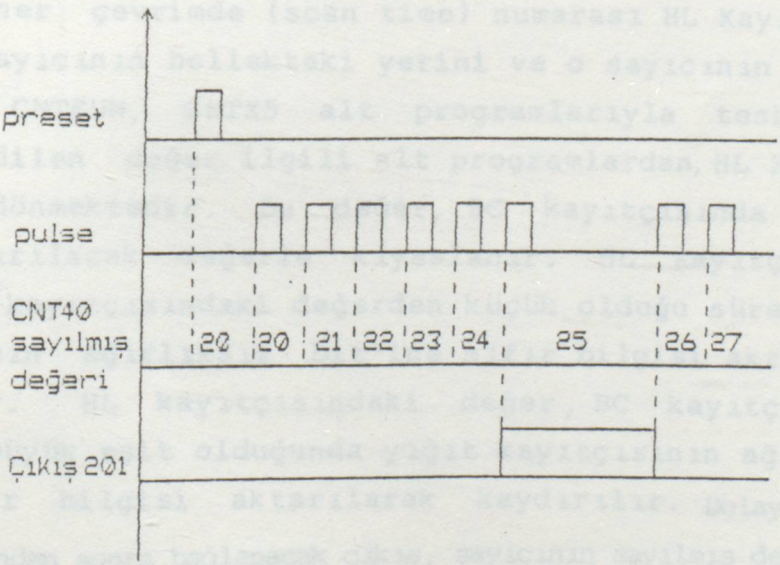
Yazılımı

Bu komut yazılımında, sayıcı numarası ve karşılaştırılacak değer verilir. İşletim programı compile tablosundan aldığı, sayıcı numarasını HL kayıtçısına, karşılaştırılacak değeri ise BC kayıtçısına yerleştirir. Bu alt program, her çevrimde, numarası HL kayıtçısında bulunan sayıcının bellekteki yerini ve sayıcının sayılmış değerini CNTFUN, CNTX5 alt programları aracılığıyla tesbit ederek, BC kayıtçısında bulunan değerle karşılaştırılır. Eğer bu iki değer arasında eşitlik varsa, yığıt kayıtçısının sıfır numaralı bitini set eder. Eşitlik yoksa yığıt kayıtçısının (DE kayıtçısı) ağırlıksız bitini sıfırlar. Dolayısıyla FUN18'den sonra bağlanacak herhangi bir çıkış sadece sayılmış değer ve yazılımda verilen değerlerin eşitliği süresince aktif olacaktır.

ÖRNEK

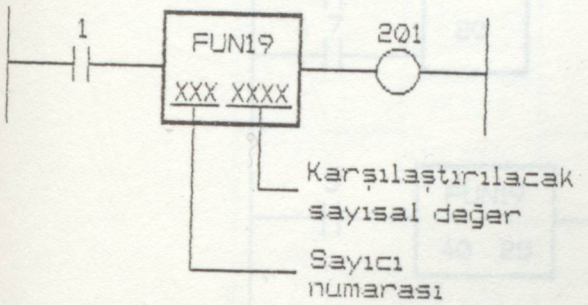



```
LOD     0
LOD     715
LOD     7
CNT     40      20
LOD     5
FUN18   40      25
OUT     201
END
```



4.4.18. FUN19 SAYICILARDA SAYILMIŞ DEĞER BÜYÜK-KÜÇÜK KONTROLÜ

Bu komutla herhangi bir sayıcının sayılmış değeri, komut yazılımında verilen sayısal değerden büyüklüğü ya da küçüklüğü kontrol edilir.



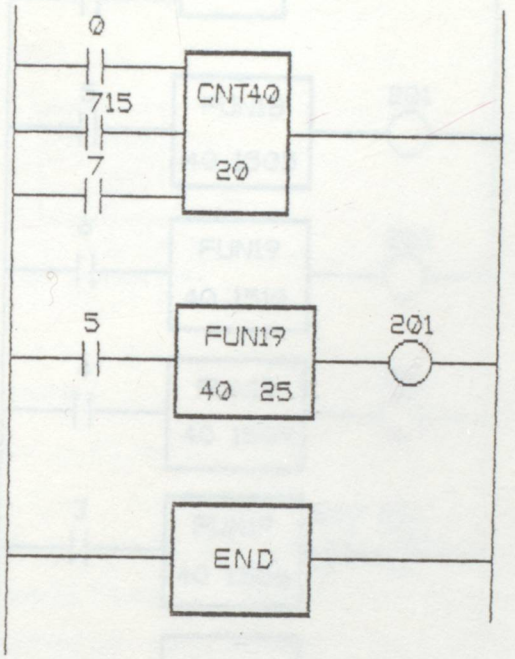

```
LOD     1
FUN19   XXX     X
```

Grafik gösterimi

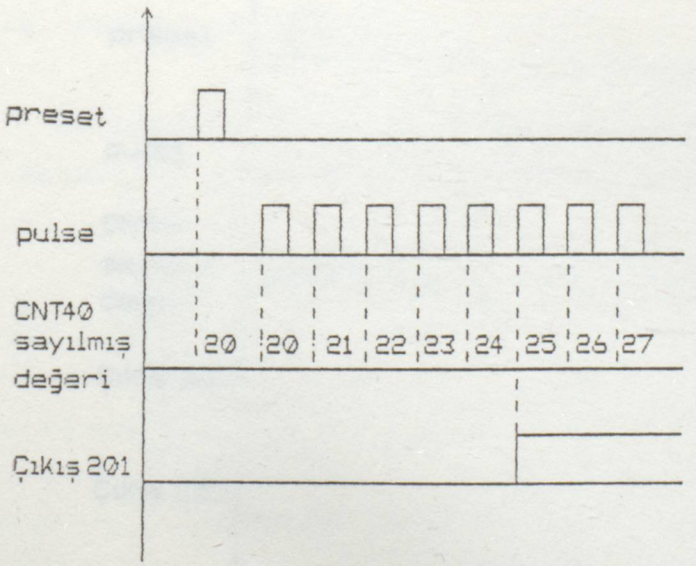
Yazılımı

Bu komut yazılımında, sayıcı numarası ve karşılaştırılacak değer verilir. İşletim programı, derlenmiş program tablosundan aldığı, sayıcı numarasını HL kayıtçısına, karşılaştırılacak deri ise BC kayıtçısına yerleştir. Bu alt program her çevrimde (scan time) numarası HL Kayıtçısında bulunan sayıcının bellekteki yerini ve o sayıcının sayılmış değerini CNTFUN, CNTX5 alt programlarıyla tesbit eder. Tesbit edilen değer, ilgili alt programlardan, HL Kayıtçısı içinde dönmektedir. Bu değer, BC kayıtçısında bulunan karşılaştırılacak değerle kıyaslanır. HL Kayıtçısındaki değer, BC kayıtçısındaki değerden küçük olduğu sürece yığıt kayıtçısının ağırlıksız bit'ine sıfır bilgisi aktarılarak kaydırılır. HL kayıtçısındaki değer, BC kayıtçısındaki değerden büyük eşit olduğunda yığıt kayıtçısının ağırlıksız bitine bir bilgisi aktarılarak kaydırılır. Dolayısıyla FUN19 komutundan sonra bağlanacak çıkış, sayıcının sayılmış değeri kontrol edilecek değerden büyük yada eşit olduğu durumlarda aktif, küçük olduğu durumda ise pasif olacaktır.

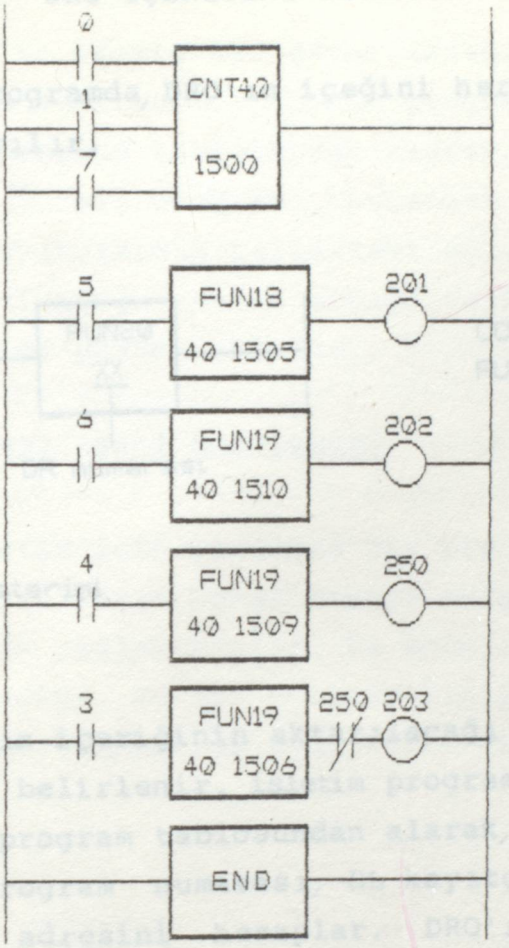
ÖRNEK



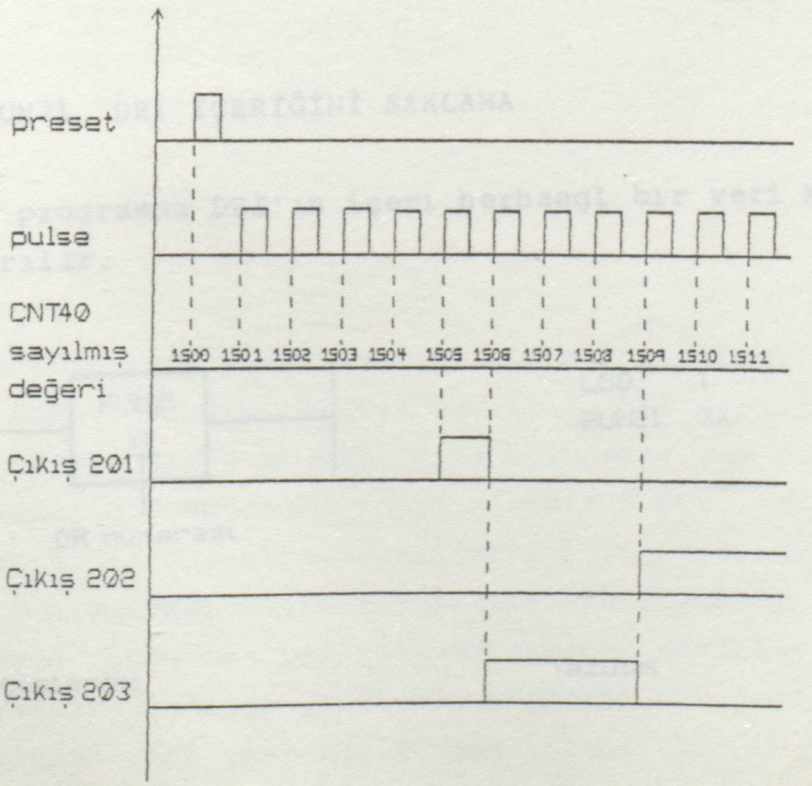

```
LOD      0
LOD      715
LOD      7
CNT      40      20
LOD      5
FUN19    40      25
OUT      201
END
```



ÖRNEK

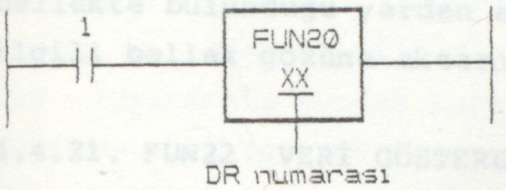



```
LOD      0
LOD      1
LOD      7
CNT      40      1500
LOD      5
FUN18    40      1505
OUT      201
LOD      6
FUN19    40      1510
OUT      202
LOD      4
FUN19    40      1509
OUT      250
LOD      3
FUN19    40      1506
ANDNOT   250
OUT      203
END
```



4.4.19 FUN20 DRO İÇERİĞİNİ SAKLAMA

Bu programda, DRO'ın içeğini herhangi bir veri kayıtcısına aktarılır.



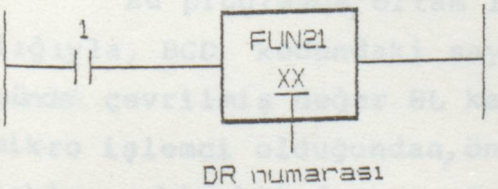

```
LOD    1
FUN20  XX
```

Grafik gösterimi Yazılımı

DRO'ın içeriğinin aktarılacağı veri kayıtçısı, komut yazılımında belirlenir. işletim programı, belirlenen numarayı derlenmiş program tablosundan alarak, HL kayıtçısına kaydeder. Bu program numarası, HL kayıtçısında bulunan DR'nin bellekteki adresini hesaplar. DRO'ın bellekteki yerinden aldığı veriyi, adresi hesaplanmış, ilgili bellek gözüne aktarır.

4.4.20. FUN21 DRI İÇERİĞİNİ SAKLAMA

Bu programda DRI'ın içeri herhangi bir veri kayıtçısına aktarılır.



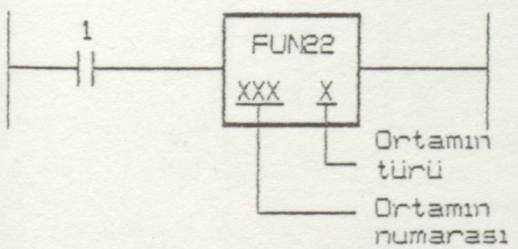

```
LOD    1
FUN21  XX
```

Grafik gösterimi Yazılımı

DRI'in içeriğinin aktarılacağı veri kayıtçısı, komut yazılımında belirlenir. işletim programı, belirlenen numarayı derlenmiş program tablosundan alarak, HL kayıtcısına yerleştirir. FUN21 alt programı, numarası HL Kayıtçısında bulunan, veri kayıtçısının bellekteki adresini hesaplar. DRI'ın bellekte bulunduğu yerden aldığı veriyi, adresi belirlenmiş ilgili bellek gözüne aktarır.

4.4.21. FUN22 VERİ GÖSTERGESİ

Denetim için yazılmış bir programda, sayıcıların ya da zamanlayıcıların hangi değere eriştiğini görmek, bir çok yönden fayda sağlayacaktır. Bu amaçla FUN22 alt programı oluşturulmuştur. Bu alt program ilgili ortamın içeriği BCD koduna çevirerek, harici 4-dijitlik veri gösterisine aktarmaktadır.



Grafik gösterimi

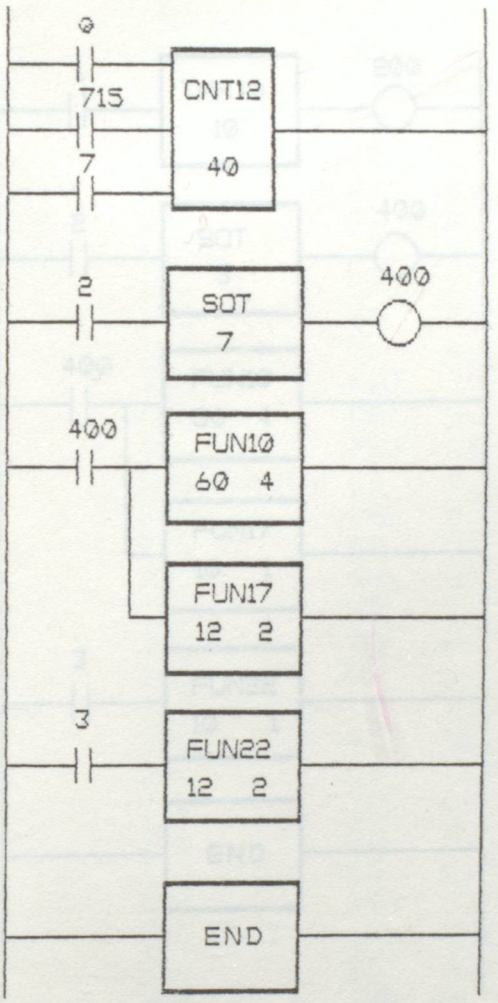
```
LOD     1
FUN22   XXX   X
```

Yazılımı

Bu programda, ortam içeriği BDC alt programı aracılığıyla, BCD kodundaki sayıya çevrilir. BDC programı dönüşünde çevrilmiş değer HL kayıtçısı içindedir. Z80 8-bitlik mikro işlemci olduğundan, önce ağırlıksız 8 bit, sonra ağırlıklı sekiz bit dışarı atılır. (Harici veri göstericisinin donanımı bir sonraki kısımda anlatılacaktır). FUN22 komutu aracılığıyla numarası belirlenmiş sayıcı, zamanlayıcı veya veri kayıtçısı içeriği programın herhangi bir yerinde, dışarıda bulunan veri göstergelerinde izlenebilir.

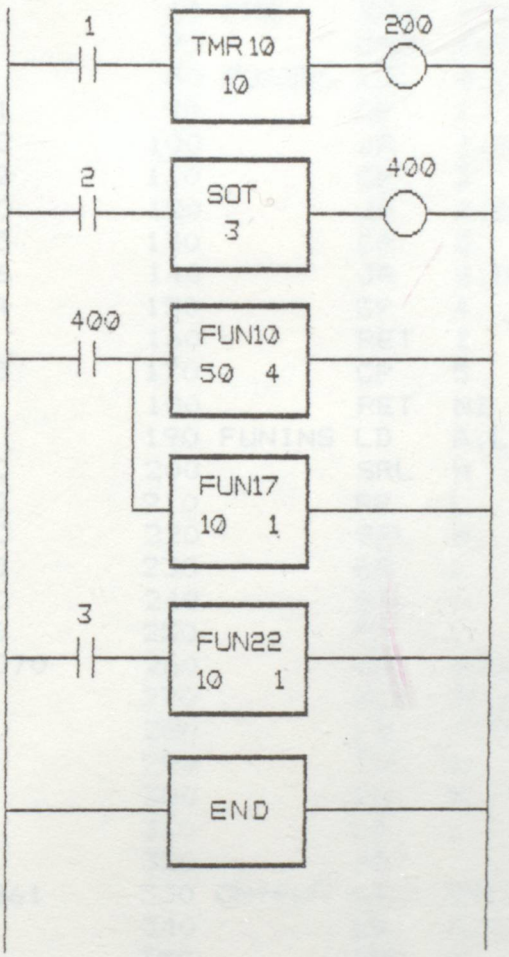
Örnek 1'de verilen programda, preset değeri 40 olan 12 numaralı sayıcı tanımlanmıştır. 2 numaralı kontak açık iken, 3 numaralı kontak kapatılarak FUN22 aracılığı ile 12 numaralı sayıcının sayılmış değeri harici veri göstericisinde izlenebilir. 2 numaralı kontak kapandığı andan itibaren, 12 nolu sayıcının preset değeri gözüne, DRO'da bulunan 60 değeri yüklenir. Sayıcıya uygulanacak Preset işaretiyle sayıcı bu değerden itibaren saymaya başlayacaktır. Bu olaylar 3 numaralı kontak kapalı kaldığı sürece FUN22 aracılığıyla harici veri göstergesinde izlenebilir. Örnek 2'de ise bir zamanlayıcının zamanlama değerinin izlenmesi için gereken bir program parçası verilmiştir.

ÖRNEK 1




```
LOD     0
LOD     715
LOD     7
CNT     12      40
LOD     2
SOT     7
OUT     400
LOD     400
FUN10   60      4
FUN17   12      2
FUN22   12      2
END
```

ÖRNEK 2




```
LOD     1
TMR     10      10
OUT     200
LOD     2
SOT     3
OUT     400
LOD     400
FUN10   50      4
FUN17   10      1
LOD     3
FUN22   10      1
END
```

4.5. P.D. ÖZEL İŞLEV ALT PROGRAMLARI

```
7000              10 PP2     EQU  #7000
6138              20 CNTX5   EQU  #6138
61C6              30 TMX5    EQU  #61C6
642E              40 INPSB   EQU  #642E
7500              50 PP7     EQU  #7500
7502              60 PP8     EQU  #7502
6700              70         ORG  #6700
6700  79          80 FUNSEL  LD   A,C
6701  FE01        90         CP   1
6703  282C       100         JR   Z,TMRFUN
6705  FE02       110         CP   2
6707  2820       120         JR   Z,CNTFUN
6709  FE03       130         CP   3
670B  2856       140         JR   Z,DRFUN
670D  FE04       150         CP   4
670F  C8         160         RET  Z
6710  FE05       170         CP   5
6712  C0         180         RET  NZ
6713  7D         190 FUNINS  LD   A,L
6714  CB3C       200         SRL  H
6716  CB1D       210         RR   L
6718  CB3C       220         SRL  H
671A  CB1D       230         RR   L
671C  CB3C       240         SRL  H
671E  CB1D       250         RR   L
6720  010070     260         LD   BC,PP2
6723  09         270         ADD  HL,BC
6724  7E         280         LD   A,(HL)
6725  23         290         INC  HL
6726  66         300         LD   H,(HL)
6727  6F         310         LD   L,A
6728  C9         320         RET
6729  CD3861     330 CNTFUN  CALL CNTX5
672C  7E         340         LD   A,(HL)
672D  23         350         INC  HL
672E  66         360         LD   H,(HL)
672F  6F         370         LD   L,A
6730  C9         380         RET
6731  CDC661     390 TMRFUN  CALL TMX5
6734  E5         400         PUSH HL
6735  DDE1       410         POP  IX
6737  DDCB047E   420         BIT  7,(IX+4)
673B  2004       430         JR   NZ,TTMR
673D  210000     440         LD   HL,0
6740  C9         450         RET
6741  DDCB0446   460 TTMR    BIT  0,(IX+4)
6745  2807       470         JR   Z,TMRFN
6747  DD6E02     480         LD   L,(IX+2)
674A  DD6603     490         LD   H,(IX+3)
674D  C9         500         RET
```

```
674E  06FB      510 TMRFN  LD   B,#FB
6750  0EF2      520        LD   C,#F2
6752  ED78      530        IN   A,(C)
6754  0EF3      540        LD   C,#F3
6756  ED40      550        IN   B,(C)
6758  4F        560        LD   C,A
6759  DD6E00    570        LD   L,(IX+0)
675C  DD6601    580        LD   H,(IX+1)
675F  A7        590        AND  A
6760  ED42      600        SBC  HL,BC
6762  C9        610        RET
6763  CB25      620 DRFUN  SLA  L
6765  CB14      630        RL   H
6767  010075    640        LD   BC,PP7
676A  09        650        ADD  HL,BC
676B  7E        660        LD   A,(HL)
676C  23        670        INC  HL
676D  66        680        LD   H,(HL)
676E  6F        690        LD   L,A
676F  C9        700        RET
6770  D5        710 BDC    PUSH DE     ; BINARY TO BCD CONVERTION
6771  110000    720        LD   DE,0   ; (HL)BIN = (HL)BCD
6774  0E10      730        LD   C,#10
6776  CB15      740 BDCS   RL   L
6778  CB14      750        RL   H
677A  7B        760        LD   A,E
677B  8F        770        ADC  A,A
677C  27        780        DAA
677D  5F        790        LD   E,A
677E  7A        800        LD   A,D
677F  8F        810        ADC  A,A
6780  27        820        DAA
6781  57        830        LD   D,A
6782  0D        840        DEC  C
6783  20F1      850        JR   NZ,BDCS
6785  6B        860        LD   L,E
6786  62        870        LD   H,D
6787  D1        880        POP  DE
6788  C9        890        RET
6789  D5        900 DBC    PUSH DE     ; BCD TO BINARY CONVERTION
678A  110000    910        LD   DE,0   ; (HL)BCD = (HL)BIN
678D  0E10      920        LD   C,#10
678F  CB15      930 DBCS   RL   L
6791  CB14      940        RL   H
6793  7B        950        LD   A,E
6794  8F        960        ADC  A,A
6795  5F        970        LD   E,A
6796  7A        980        LD   A,D
6797  8F        990        ADC  A,A
6798  57       1000        LD   D,A
6799  0D       1010        DEC  C
679A  20F3     1020        JR   NZ,DBCS
679C  6B       1030        LD   L,E
679D  62       1040        LD   H,D
679E  D1       1050        POP  DE
679F  C9       1060        RET
```

```
67A0  CB43       1070 FUN1   BIT  0,E
67A2  C8         1080        RET  Z
67A3  2A0075     1090        LD   HL,(PP7)
67A6  CD7067     1100        CALL BDC
67A9  220075     1110        LD   (PP7),HL
67AC  C9         1120        RET
67AD  CB43       1130 FUN2   BIT  0,E
67AF  C8         1140        RET  Z
67B0  2A0075     1150        LD   HL,(PP7)
67B3  CD8967     1160        CALL DBC
67B6  220075     1170        LD   (PP7),HL
67B9  C9         1180        RET
67BA  CB43       1190 FUN3   BIT  0,E
67BC  C8         1200        RET  Z
67BD  CD0067     1210        CALL FUNSEL
67C0  ED4B0075   1220        LD   BC,(PP7)
67C4  A7         1230        AND  A
67C5  ED42       1240        SBC  HL,BC
67C7  F5         1250        PUSH AF
67C8  210070     1260        LD   HL,PP2
67CB  015800     1270        LD   BC,88
67CE  09         1280        ADD  HL,BC
67CF  F1         1290        POP  AF
67D0  2812       1300        JR   Z,F3S2
67D2  3808       1310        JR   C,F3S1
67D4  CB86       1320        RES  6,(HL)
67D6  CBBE       1330        RES  7,(HL)
67D8  23         1340        INC  HL
67D9  CBC6       1350        SET  0,(HL)
67DB  C9         1360        RET
67DC  CBF6       1370 F3S1   SET  6,(HL)
67DE  CBBE       1380        RES  7,(HL)
67E0  23         1390        INC  HL
67E1  CB86       1400        RES  0,(HL)
67E3  C9         1410        RET
67E4  CBB6       1420 F3S2   RES  6,(HL)
67E6  CBFE       1430        SET  7,(HL)
67E8  23         1440        INC  HL
67E9  CB86       1450        RES  0,(HL)
67EB  C9         1460        RET
67EC  CB43       1470 FUN4   BIT  0,E
67EE  C8         1480        RET  Z
67EF  D5         1490        PUSH DE
67F0  CD0067     1500        CALL FUNSEL
67F3  E5         1510        PUSH HL
67F4  215870     1520        LD   HL,#7058
67F7  5E         1530        LD   E,(HL)
67F8  ED4B0075   1540        LD   BC,(PP7)
67FC  7B         1550        LD   A,E
67FD  E608       1560        AND  #8
67FF  2801       1570        JR   Z,F4S1
6801  37         1580        SCF
```

```
6802  E1      1590 F4S1   POP  HL
6803  D1      1600        POP  DE
6804  ED4A    1610        ADC  HL,BC
6806  220075  1620        LD   (PP7),HL
6809  215870  1630        LD   HL,#7058
680C  DA1268  1640        JP   C,F4S2
680F  CB9E    1650        RES  3,(HL)
6811  C9      1660        RET
6812  CBDE    1670 F4S2   SET  3,(HL)
6814  C9      1680        RET
6815  CB43    1690 FUN5   BIT  0,E
6817  C8      1700        RET  Z
6818  D5      1710        PUSH DE
6819  CD0067  1720        CALL FUNSEL
681C  E5      1730        PUSH HL
681D  215870  1740        LD   HL,#7058
6820  5E      1750        LD   E,(HL)
6821  2A0075  1760        LD   HL,(PP7)
6824  7B      1770        LD   A,E
6825  E608    1780        AND  8
6827  2801    1790        JR   Z,F5S1
6829  37      1800        SCF
682A  C1      1810 F5S1   POP  BC
682B  D1      1820        POP  DE
682C  ED42    1830        SBC  HL,BC
682E  220075  1840        LD   (PP7),HL
6831  215870  1850        LD   HL,#7058
6834  DA3A68  1860        JP   C,F5S2
6837  CB9E    1870        RES  3,(HL)
6839  C9      1880        RET
683A  CBDE    1890 F5S2   SET  3,(HL)
683C  C9      1900        RET
683D  CB43    1910 FUN6   BIT  0,E
683F  C8      1920        RET  Z
6840  D5      1930        PUSH DE
6841  CD0067  1940        CALL FUNSEL
6844  44      1950        LD   B,H
6845  4D      1960        LD   C,L
6846  210000  1970        LD   HL,0
6849  110000  1980        LD   DE,0
684C  3E10    1990        LD   A,16
684E  29      2000 MULO   ADD  HL,HL
684F  EB      2010        EX   DE,HL
6850  ED6A    2020        ADC  HL,HL
6852  EB      2030        EX   DE,HL
6853  E5      2040        PUSH HL
6854  2A0075  2050        LD   HL,(PP7)
6857  CB15    2060        RL   L
6859  CB14    2070        RL   H
685B  220075  2080        LD   (PP7),HL
685E  E1      2090        POP  HL
685F  3004    2100        JR   NC,MUL1
6861  09      2110        ADD  HL,BC
6862  3001    2120        JR   NC,MUL1
6864  13      2130        INC  DE
```

```
6865  3D          2140 MUL1   DEC  A
6866  20E6        2150        JR   NZ,MULO
6868  220075      2160        LD   (PP7),HL
686B  210075      2170        LD   HL,PP7
686E  23          2180        INC  HL
686F  23          2190        INC  HL
6870  73          2200        LD   (HL),E
6871  23          2210        INC  HL
6872  72          2220        LD   (HL),D
6873  D1          2230        POP  DE
6874  C9          2240        RET
6875  CB43        2250 FUN7   BIT  0,E
6877  C8          2260        RET  Z
6878  D5          2270        PUSH DE
6879  CD0067      2280        CALL FUNSEL
687C  44          2290        LD   B,H
687D  4D          2300        LD   C,L
687E  ED5B0075    2310        LD   DE,(PP7)
6882  210000      2320        LD   HL,0
6885  AF          2330        XOR  A
6886  3E10        2340        LD   A,16
6888  CB13        2350 DIV1   RL   E
688A  CB12        2360        RL   D
688C  ED6A        2370        ADC  HL,HL
688E  ED42        2380        SBC  HL,BC
6890  3001        2390        JR   NC,DIV2
6892  09          2400        ADD  HL,BC
6893  3F          2410 DIV2   CCF
6894  3D          2420        DEC  A
6895  20F1        2430        JR   NZ,DIV1
6897  EB          2440        EX   DE,HL
6898  ED6A        2450        ADC  HL,HL
689A  220075      2460        LD   (PP7),HL
689D  210075      2470        LD   HL,PP7
68A0  23          2480        INC  HL
68A1  72          2490        LD   (HL),D
68A2  7B          2500        LD   A,E
68A3  320275      2510        LD   (PP8),A
68A6  D1          2520        POP  DE
68A7  C9          2530        RET
68A8  CB43        2540 FUN8   BIT  0,E
68AA  C8          2550        RET  Z
68AB  3A0275      2560        LD   A,(PP8);Uzunluk hatasi
68AE  85          2570        ADD  A,L
68AF  D664        2580        SUB  #64
68B1  F0          2590        RET  P
                  2600 ;************************
68B2  7D          2610        LD   A,L
68B3  D603        2620        SUB  3
68B5  F8          2630        RET  M
                  2640 ;************************
```

```
68B6  D5          2650        PUSH DE
68B7  3A0275      2660        LD   A,(PP8)
68BA  3D          2670        DEC  A     ;Shift Uzunluk
68BB  85          2680        ADD  A,L
68BC  6F          2690        LD   L,A
68BD  2600        2700        LD   H,0
68BF  CB25        2710        SLA  L
68C1  CB14        2720        RL   H
68C3  010075      2730        LD   BC,PP7
68C6  09          2740        ADD  HL,BC
68C7  ED4B0275    2750        LD   BC,(PP8)
68CB  0B          2760        DEC  BC
68CC  CB21        2770        SLA  C
68CE  CB10        2780        RL   B
68D0  54          2790        LD   D,H
68D1  5D          2800        LD   E,L
68D2  13          2810        INC  DE
68D3  2B          2820        DEC  HL
68D4  EDB8        2830        LDDR
68D6  210075      2840        LD   HL,PP7
68D9  1B          2850        DEC  DE
68DA  EDA0        2860        LDI
68DC  EDA0        2870        LDI
68DE  D1          2880        POP  DE
68DF  C9          2890        RET
68E0  CB43        2900 FUN9   BIT  0,E
68E2  C8          2910        RET  Z
68E3  45          2920        LD   B,L;**** KAC DIGIT SHIFT OLACAGI L'DE*
68E4  3E00        2930 F9     LD   A,0
68E6  210075      2940        LD   HL,PP7
68E9  ED6F        2950        RLD
68EB  23          2960        INC  HL
68EC  ED6F        2970        RLD
68EE  23          2980        INC  HL
68EF  ED6F        2990        RLD
68F1  23          3000        INC  HL
68F2  ED6F        3010        RLD
68F4  10EE        3020        DJNZ F9
68F6  C9          3030        RET
68F7  CB43        3040 FUN10  BIT  0,E;  DATA LOAD (16 BIT)
68F9  C8          3050        RET  Z
68FA  CD0067      3060        CALL FUNSEL
68FD  220075      3070        LD   (PP7),HL
6900  C9          3080        RET
6901  CB43        3090 FUN11  BIT  0,E
6903  C8          3100        RET  Z
6904  DDE5        3110        PUSH IX
6906  C5          3120        PUSH BC
6907  DD210075    3130        LD   IX,PP7
690B  0608        3140        LD   B,8
690D  DDE5        3150 CANN   PUSH IX
690F  C5          3160        PUSH BC
6910  E5          3170        PUSH HL
6911  CD2E64      3180        CALL INPSB
6914  E1          3190        POP  HL
6915  C1          3200        POP  BC
```

```
6916  DDE1      3210        POP  IX
6918  CB2F      3220        SRA  A
691A  DDCB0016  3230        RL   (IX+0)
691E  23        3240        INC  HL
691F  10EC      3250        DJNZ CANN
6921  C1        3260        POP  BC
6922  DDE1      3270        POP  IX
6924  C9        3280        RET
6925  CB43      3290 FUN12  BIT  0,E;  DATA LOAD (Indirect 16 bit)
6927  C8        3300        RET  Z
6928  ED4B0275  3310        LD   BC,(PP8)
692C  09        3320        ADD  HL,BC
692D  CD6367    3330        CALL DRFUN
6930  220075    3340        LD   (PP7),HL
6933  C9        3350        RET
6934  CB43      3360 FUN13  BIT  0,E;  DATA LOAD (16 BIT DR1)
6936  C8        3370        RET  Z
6937  CD0067    3380        CALL FUNSEL
693A  220275    3390        LD   (PP8),HL
693D  C9        3400        RET
693E  CB43      3410 FUN14  BIT  0,E
6940  C8        3420        RET  Z
6941  DDE5      3430        PUSH IX
6943  C5        3440        PUSH BC
6944  DD210275  3450        LD   IX,PP8
6948  0608      3460        LD   B,8
694A  DDE5      3470 CAN    PUSH IX
694C  C5        3480        PUSH BC
694D  E5        3490        PUSH HL
694E  CD2E64    3500        CALL INPSB
6951  E1        3510        POP  HL
6952  C1        3520        POP  BC
6953  DDE1      3530        POP  IX
6955  CB2F      3540        SRA  A
6957  DDCB0016  3550        RL   (IX+0)
695B  23        3560        INC  HL
695C  10EC      3570        DJNZ CAN
695E  C1        3580        POP  BC
695F  DDE1      3590        POP  IX
6961  C9        3600        RET
6962  CB43      3610 FUN15  BIT  0,E;DATAINCREMENT
6964  C8        3620        RET  Z
6965  CB25      3630        SLA  L
6967  CB14      3640        RL   H
6969  010075    3650        LD   BC,PP7
696C  09        3660        ADD  HL,BC
696D  4E        3670        LD   C,(HL)
696E  23        3680        INC  HL
696F  46        3690        LD   B,(HL)
6970  03        3700        INC  BC
6971  70        3710        LD   (HL),B
6972  2B        3720        DEC  HL
6973  71        3730        LD   (HL),C
6974  C9        3740        RET
```

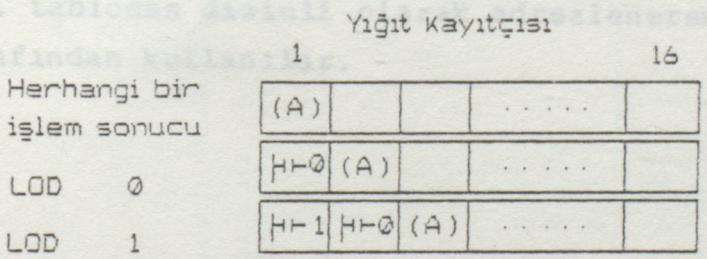
```
6975  CB43      3750 FUN16  BIT  0,E;DATADECREMENT
6977  C8        3760        RET  Z
6978  CB25      3770        SLA  L
697A  CB14      3780        RL   H
697C  010075    3790        LD   BC,PP7
697F  09        3800        ADD  HL,BC
6980  4E        3810        LD   C,(HL)
6981  23        3820        INC  HL
6982  46        3830        LD   B,(HL)
6983  0B        3840        DEC  BC
6984  70        3850        LD   (HL),B
6985  2B        3860        DEC  HL
6986  71        3870        LD   (HL),C
6987  C9        3880        RET
6988  CB43      3890 FUN17  BIT  0,E
698A  C8        3900        RET  Z
698B  CDCF69    3910        CALL FUNSL
698E  E5        3920        PUSH HL
698F  DDE1      3930        POP  IX
6991  2A0075    3940        LD   HL,(PP7)
6994  DD7502    3950        LD   (IX+2),L
6997  DD7403    3960        LD   (IX+3),H
699A  C9        3970        RET
699B  CB43      3980 FUN18  BIT  0,E;  CNT DEG. ESITLIGI
699D  C8        3990        RET  Z
699E  C5        4000        PUSH BC
699F  CD2967    4010        CALL CNTFUN
69A2  C1        4020        POP  BC
69A3  37        4030        SCF
69A4  3F        4040        CCF
69A5  ED42      4050        SBC  HL,BC
69A7  C2B069    4060        JP   NZ,FC11
69AA  37        4070        SCF
69AB  CB13      4080        RL   E
69AD  CB12      4090        RL   D
69AF  C9        4100        RET
69B0  CB23      4110 FC11   SLA  E
69B2  CB12      4120        RL   D
69B4  C9        4130        RET
69B5  CB43      4140 FUN19  BIT  0,E;  CNT DEG. BUYUKLUGU
69B7  C8        4150        RET  Z
69B8  C5        4160        PUSH BC
69B9  CD2967    4170        CALL CNTFUN
69BC  C1        4180        POP  BC
69BD  37        4190        SCF
69BE  3F        4200        CCF
69BF  ED42      4210        SBC  HL,BC
69C1  F2C969    4220        JP   P,FC21
69C4  CB23      4230        SLA  E
69C6  CB12      4240        RL   D
69C8  C9        4250        RET
69C9  37        4260 FC21   SCF
69CA  CB13      4270        RL   E
69CC  CB12      4280        RL   D
69CE  C9        4290        RET
```

```
69CF  79          4300 FUNSL  LD   A,C
69D0  FE01        4310        CP   1
69D2  CAC661      4320        JP   Z,TMX5
69D5  FE02        4330        CP   2
69D7  CA3861      4340        JP   Z,CNTX5
69DA  C9          4350        RET
69DB  CB25        4360 DRFUN1 SLA  L
69DD  CB14        4370        RL   H
69DF  010075      4380        LD   BC,PP7
69E2  09          4390        ADD  HL,BC
69E3  73          4410        LD   (HL),E
69E4  23          4420        INC  HL
69E5  72          4430        LD   (HL),D
69E6  C9          4440        RET
69E7  CB43        4450 FUN20  BIT  0,E;D.STORE 16.BIT (DR0)
69E9  C8          4460        RET  Z
69EA  D5          4461        PUSH DE
69EB  ED5B0075    4470        LD   DE,(PP7)
69EF  CDDB69      4490        CALL DRFUN1
69F2  D1          4491        POP  DE
69F3  C9          4500        RET
69F4  CB43        4510 FUN21  BIT  0,E;D.STORE 16.BIT (DR1)
69F6  C8          4520        RET  Z
69F7  D5          4521        PUSH DE
69F8  ED5B0275    4530        LD   DE,(PP8)
69FC  CDDB69      4550        CALL DRFUN1
69FF  D1          4551        POP  DE
6A00  C9          4560        RET
6A01  CB43        4570 FUN22  BIT  0,E;DATA DISPLAY(DYNAMIC DISPLAY)
6A03  C8          4580        RET  Z
6A04  D5          4590        PUSH DE
6A05  2A0075      4600        LD   HL,(PP7)
6A08  CD7067      4610        CALL BDC
6A0B  06FB        4620        LD   B,#FB
6A0D  0EF4        4630        LD   C,#F4
6A0F  ED69        4640        OUT  (C),L
6A11  23          4650        INC  HL
6A12  ED61        4660        OUT  (C),H
6A14  D1          4670        POP  DE
6A15  C9          4680        RET
```

4.5. Z80 CPU'LU PD DONANIM YAPISI

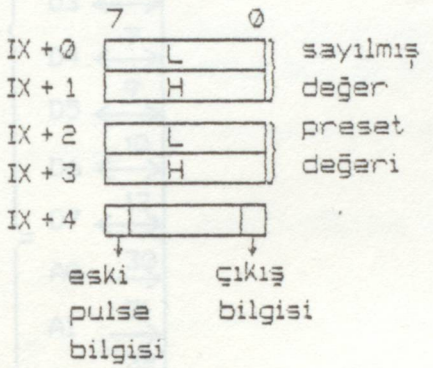
4.5.1. Z80 CPU KAYITÇILARI VE GÖREVLERİ

Z80 CPU içerisinde, sekiz tane 8 bitlik kayıtçı bulunmaktadır. Bunlar A,B,C,D,E,H,L ve F kayıtçılarıdır. Z80 CPU komutlarının hemen hemen hepsinin üzerinde gerçekleşen A kayıtçısı, Akümülatör (Accumulator) ismini alır. F kayıtçısı ise işlem sonuçlarına göre etkilenen bayrakların (Flags) saklandığı kayıtçıdır. Diğer kayıtçılar ise genel amaçlı kayıtçılardır. Z80 CPU'da kayıtçılar birleşik olarak kullanılabilir. Bunlar AF, BC, DE ve HL Çiftlerdir. Böylelikle 16 bitlik işlemlerin yapılması olanaklı olmuştur. PD yazılımında genellikle kayıtçı çiftleri kullanılmıştır. Ayrıca IX,IY, SP, PC diye adlandırılmış 16 bitlik kayıtçılarda bulunmaktadır. 8-bitlik kayıtçıların görüntüleri, 8 bitlik I (Interrupt), R (Memory Refresh) kayıtçılarıda bulunmaktadır. PD yazılımında, DE kayıtçı çifti yığıt kayıtçısı olarak kullanılmıştır. LOD komutuyla başlayan satırda ilgili kontağın durumu DE kayıtçısının son bitine yerleştirilmektedir. Bundan sonra gelecek LOD komutlarında ilgili kontakların durumu, DE sola kaydırılarak son bitine tekrar yüklenir. ANDLOD ya da ORLOD komutlarıyla geri çekilerek gerekli işlem yapılır.



A kayıtçısının yetenekleri fazla olduğundan işlem kayıtçısı olarak kullanılmıştır. Lojik işlemler E kayıtçısı ve A kayıtçısı arasında yapılır, işlem sonucu DE kayıtçısının son bitine kaydedilir.

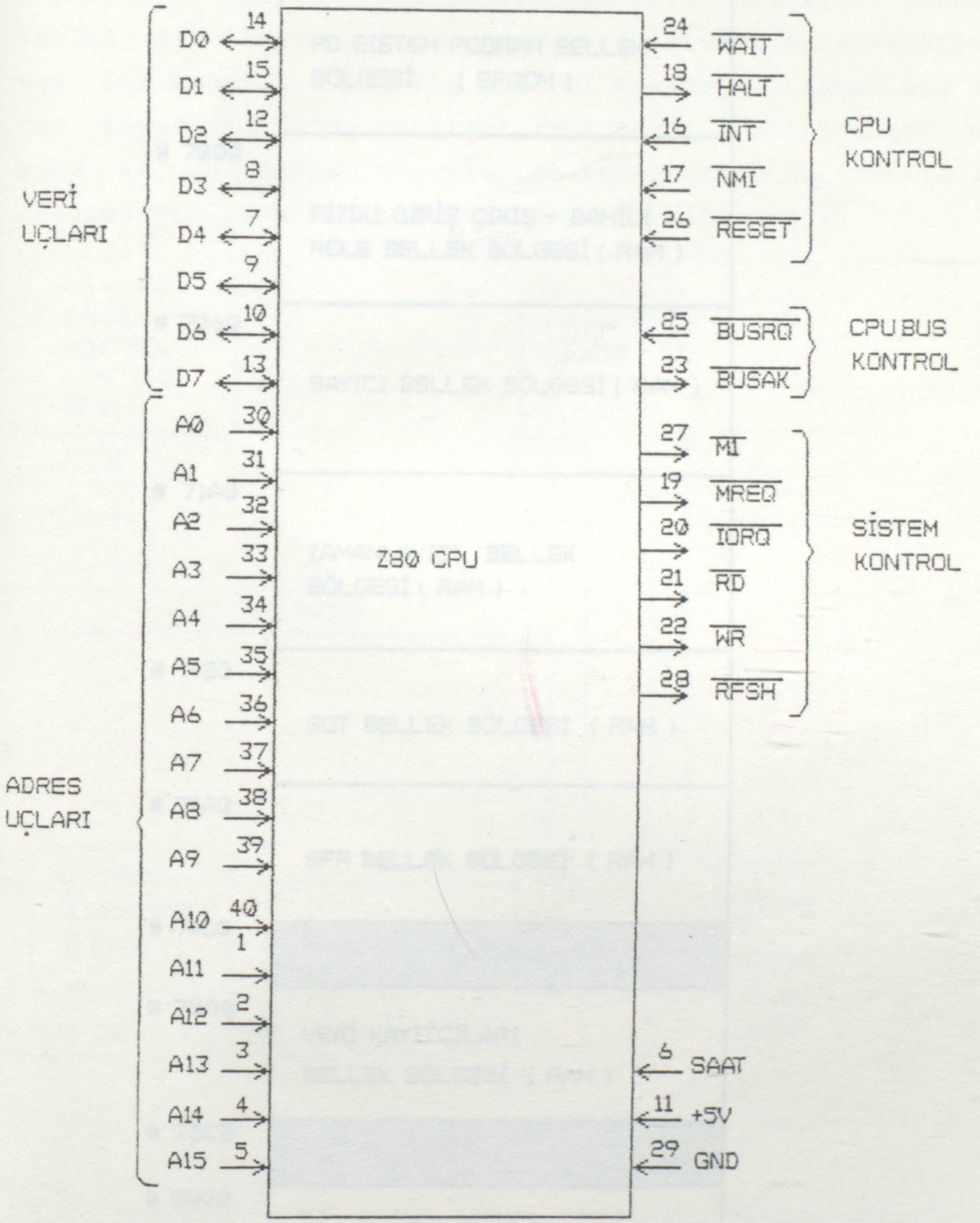
Sayıcıların ve zamanlayıcıların bellekteki bulundukları adresler ise alt programlarda hesaplanarak 16 bitlik IX kayıtçısına yüklenir. IX kayıtçısı dizinli (Index) olarak kullanıldığından, preset değeri bellek gözü, sayılmış değer bellek gözü, durum bellek gözü adresleri bulunurken kolaylık sağlamaktadır.



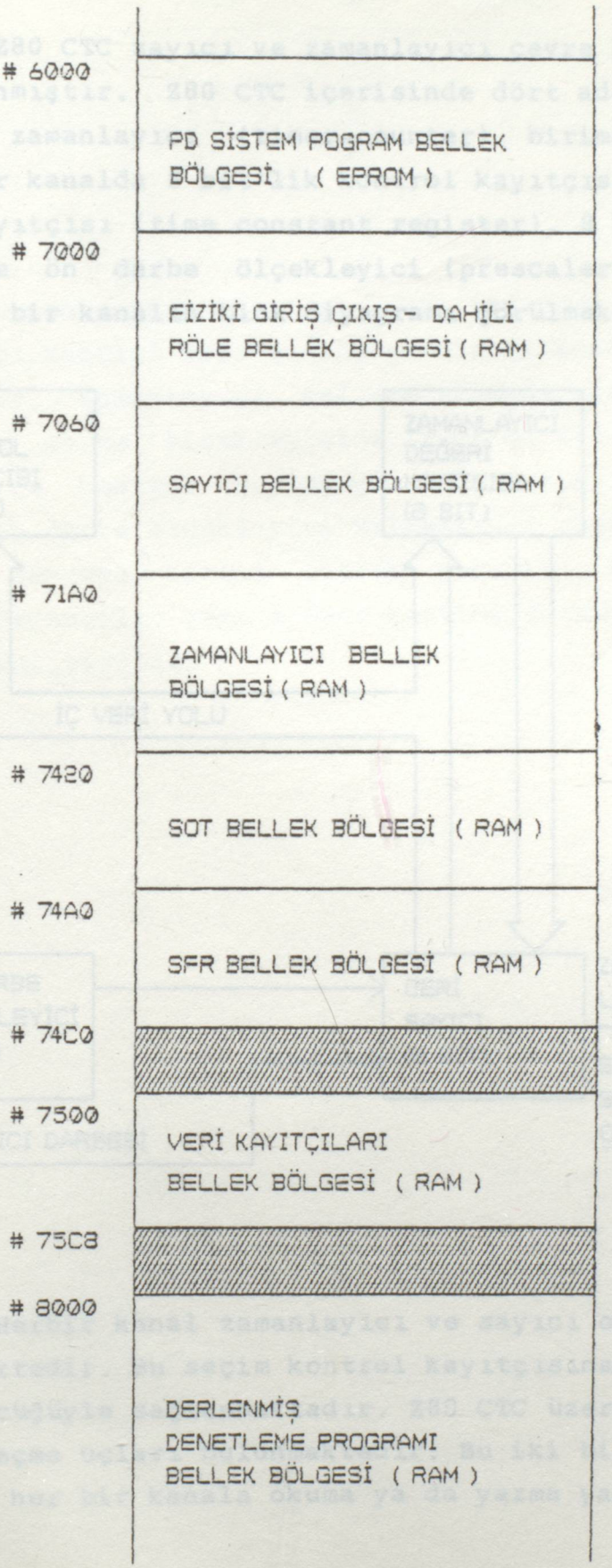
(IX) = #7065

Yukarıda 1 numaralı sayıcının bellekteki yapısı ve IX kayıtçısına göre dizinlenmesi gösterilmiştir.

HL ve BC 16 bitlik kayıtçıları ise yardımcı alt programlara (FUNSEL, CNTX5, TMX5) ortam numarası veya ortam türü göndermede ve dönüşte programa ortamın datasını getirmede kullanılır. IY kayıtçısı ise Derlenmiş programın yerleştiği tablodan dizinli olarak adreslenerek işletim programı tarafından kullanılır.

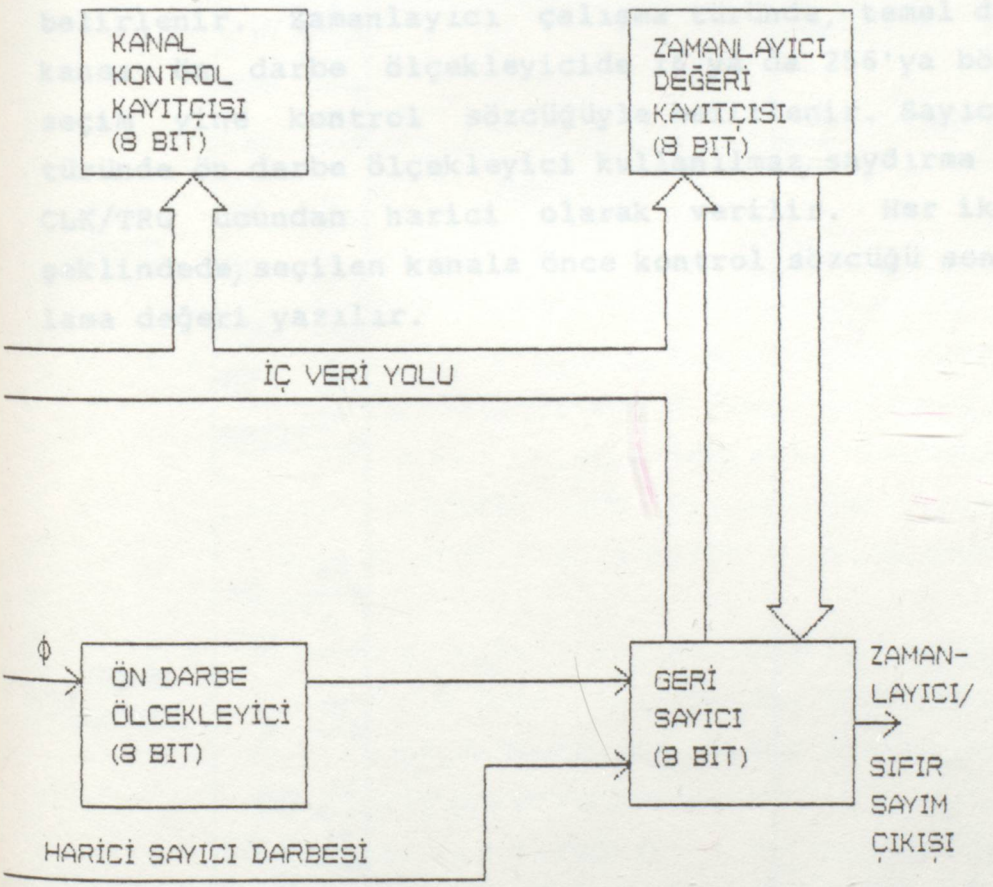

VERİ UÇLARI
D0 14
D1 15
D2 12
D3 8
D4 7
D5 9
D6 10
D7 13
ADRES UÇLARI
A0 30
A1 31
A2 32
A3 33
A4 34
A5 35
A6 36
A7 37
A8 38
A9 39
A10 40
A11 1
A12 2
A13 3
A14 4
A15 5
Z80 CPU
24 WAIT
18 HALT
16 INT
17 NMI
26 RESET
CPU KONTROL
25 BUSRQ
23 BUSAK
CPU BUS KONTROL
27 M1
19 MREQ
20 IORQ
21 RD
22 WR
28 RFSH
SİSTEM KONTROL
6 SAAT
11 +5V
29 GND

4.5.2. PD BELLEK HARİTASI



4.5.3. Z80 CTC VE KULLANILMASI

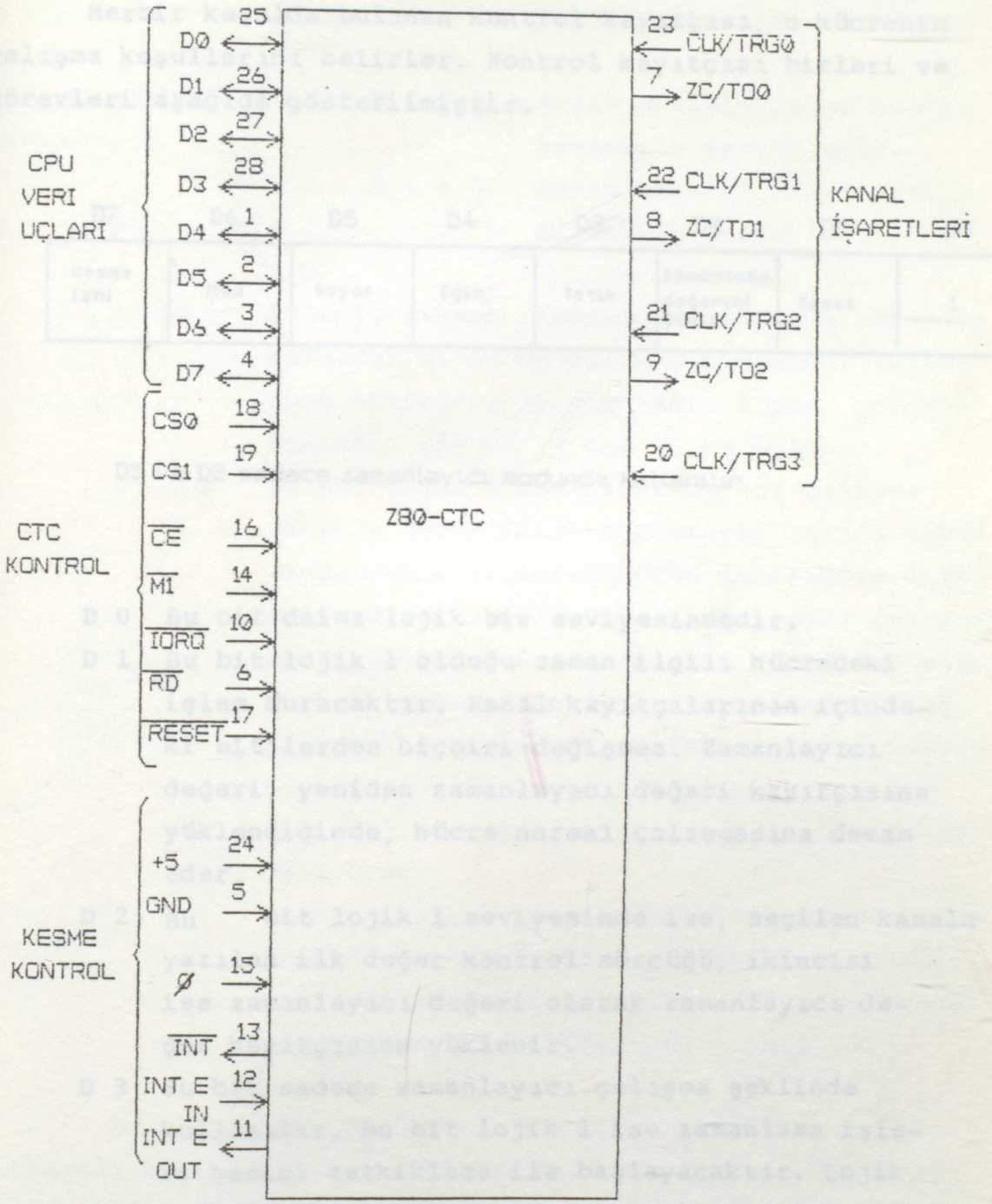
Z80 CTC sayıcı ve zamanlayıcı çevre birimi olarak tasarlanmıştır. Z80 CTC içerisinde dört adet sekiz bit'lik sayıcı zamanlayıcı (timer-counter) birimi bulunmaktadır. Her bir kanalda 8 bit'lik kontrol kayıtçısı, zamanlama değer kayıtçısı (time constant register), 8 bit'lik geri sayıcı ve ön darbe ölçekleyici (prescaler) bulunmaktadır. Aşağıda bir kanalın blok diyagramı görülmektedir.



Herbir kanal zamanlayıcı ve sayıcı olarak çalıştırılabilmektedir. Bu seçim kontrol kayıtçısına yazılacak kontrol sözcüğüyle sağlanmaktadır. Z80 CTC üzerinde CSO ve kanal seçme uçları bulunmaktadır. Bu iki bit'lik binary adres ile her bir kanala okuma ya da yazma yapılabilir.

CS1	CS0	KANAL
0	0	0
0	1	1
1	0	2
1	1	3

Hücrenin zamanlayıcımı yoksa sayıcımı olarak çalıştırılacağı, kontrol kayıtçısına yazılacak kontrol sözcüğüyle belirlenir. Zamanlayıcı çalışma türünde, temel darbe frekansı ön darbe ölçekleyicide 16 ya da 256'ya bölünür. bu seçim yine kontrol sözcüğüyle belirlenir. Sayıcı çalışma türünde ön darbe ölçekleyici kullanılmaz, saydırma darbeleri CLK/TRG ucundan harici olarak verilir. Her iki çalışma şeklindede, seçilen kanala önce kontrol sözcüğü sonra zamanlama değeri yazılır.


CPU
VERI
UÇLARI
D0 25
D1 26
D2 27
D3 28
D4 1
D5 2
D6 3
D7 4
CS0 18
CS1 19
CTC
KONTROL
CE 16
M1 14
IORQ 10
RD 6
RESET 17
KESME
KONTROL
+5 24
GND 5
Ø 15
INT 13
INT E IN 12
INT E OUT 11
Z80-CTC
23 CLK/TRG0
7 ZC/TO0
22 CLK/TRG1
8 ZC/TO1
21 CLK/TRG2
9 ZC/TO2
20 CLK/TRG3
KANAL
İŞARETLERİ

4.5.3.1 KONTROL KAYITCISI

Herbir kanalda bulunan kontrol kayıtçısı, o hücrenin çalışma koşullarını belirler. Kontrol kayıtçısı bitleri ve görevleri aşağıda gösterilmiştir.

D7	D6	D5	D4	D3	D2	D1	D0
Kesme izni	Mod	Boyut	Eğim	Tetik	Zamanlama değerini yükle	Reset	1

D5 ve D2 sadece zamanlayıcı modunda kullanılır

D 0 Bu bit daima lojik bir seviyesindedir.

D 1 Bu bit lojik 1 olduğu zaman ilgili hücredeki işlem duracaktır. Kanal kayıtçılarının içindeki bit'lerden hiçbiri değişmez. Zamanlayıcı değeri, yeniden zamanlayıcı değeri kayıtçısına yüklendiğinde, hücre normal çalışmasına devam eder.

D 2 Bu bit lojik 1 seviyesinde ise, seçilen kanala yazılan ilk değer kontrol sözcüğü, ikincisi ise zamanlayıcı değeri olarak zamanlayıcı değer kayıtçısına yüklenir.

D 3 Bu bit sadece zamanlayıcı çalışma şeklinde kullanılır. Bu bit lojik 1 ise zamanlama işlemi harici tetkikleme ile başlayacaktır. Lojik 0 konumunda ise zamanlama işlemi zamanlayıcı değeri kayıtçısı yüklenir yüklenmez başlar.

D 4 Bu bit harici CLK/TRG girişinin hangi kenarda aktif olacağını belirler.

Zamanlayıcı çalışma şekli	D 4 = 1	Zamanlamayı pozitif kenar başlatır.
	D 4 = 0	Zamanlamayı negatif kenar başlatır.
Sayıcı çalışma şekli	D 4 = 1	Sayım pozitif kenar tetiklemesiyle gerçekleşir.
	D 4 = 0	Sayım negatif kenar tetiklemesiyle gerçekleşir.

D 5 Bu bit, sadece zamanlayıcı çalışma şeklinde kullanılır. Bu ön darbe ölçekleyici faktörünü belirler. Bu bit lojik 1 ise giriş frekansı 256'ya, 0 ise 16 ya bölünür.

D 6 Bu bit lojik 1 ise kanal sayıcı çalışma şekline set edilir. Böylelikle sayıcı sayım darbelerini harici CLK/TRG girişinden alır. Lojik 0 seviyesinde ise zamanlayıcı çalışma şekli seçilir ve geri sayıcı sayım darbelerini ön darbe ölçekleyici çıkışından alır. Böylelikle zamanlama periodu, sistem darbe periodu (x) ön darbe ölçekleyici faktörü (16/256) (x) zamanlayıcı değeri, çarpımına eşit olur.

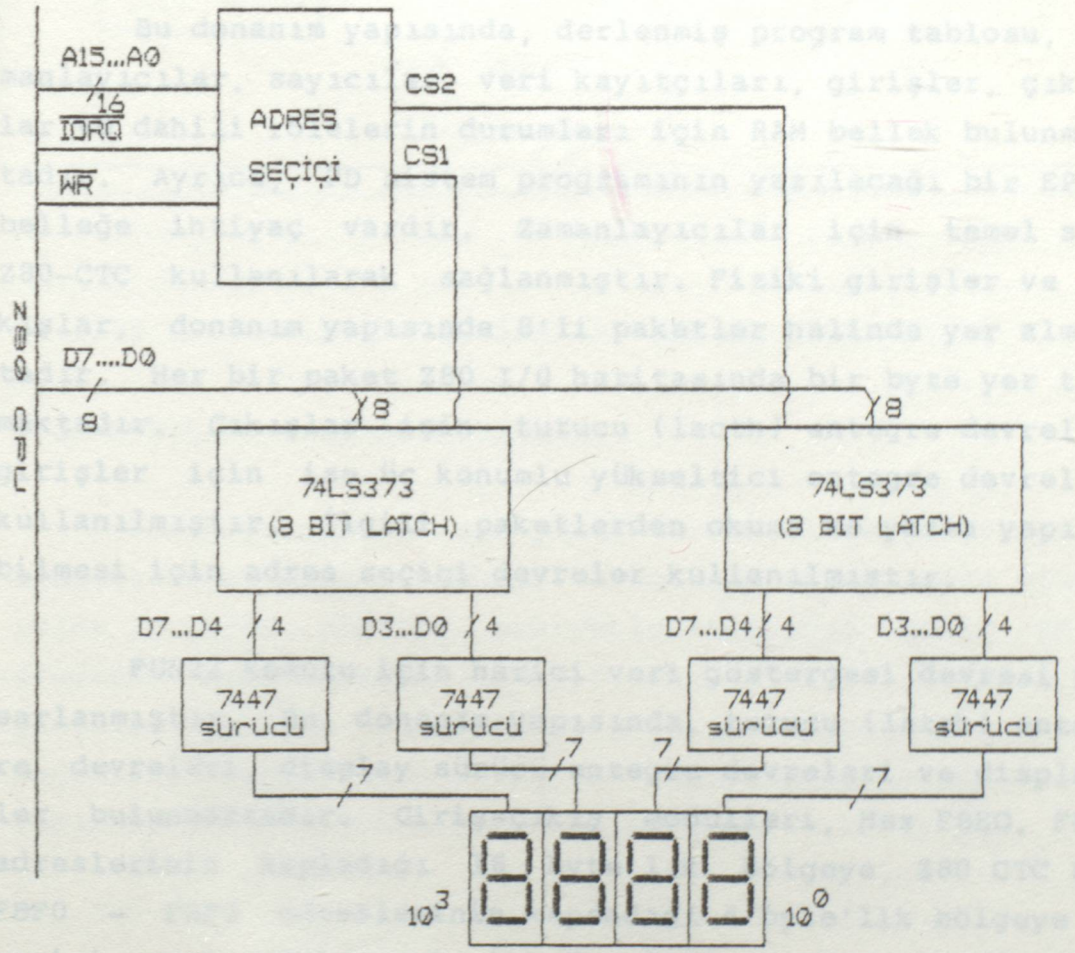
D 7 Bu bit lojik 1 ise, kesme (interrupt) işlemini sağlayacak yapı seçilmiş olur. Kesme vektörü sayım değeri sıfıra erişince kesme vektörü kayıtçısından, Z80'e yazılır.

Bu uygulamada 0 numaralı kanal zamanlayıcı, 1,2,3, numaralı kanal ise sayıcı olarak kullanılmıştır. CTC hex FBFO, FBF1, FBF2, FBF3 adreslerine yerleştirilmiştir. Sıfır numaralı kanal için ön darbe ölçekleyici 16 olarak seçilmiştir. Amstrad 4 MHz frekansla çalışmaktadır. Bu frekans temel alınarak PD zamanlayıcıları için, 100 msn'lik temel zaman süresi (time base) üretilmiştir. Bunun için 0 numaralı kanal zamanlayıcı olarak kullanılmış, sistem çalışma frekansı 4000'e bölünmüştür.

Bu çıkış 1 numaralı kanalda 100'e bölünerek 0,1 sn' lik işaret elde edilmiştir. 2 ve 3 numaralı kanallar ardışı bağlanarak, yüklenen maksimum değerden başlayarak 100 msn darbelerle geri sayıcı olarak kullanılmıştır. TMR alt programında, 2 ve 3 numaralı sayıcılar okunarak zamanlayıcı süresi tesbit edilir.

4.5.4. HARİCİ VERİ GÖSTERGESİ DONANIMI

Bu donanım, FUN22 komutu için oluşturulmuştur. Kısım 4.4.21 de anlatıldığı gibi zamanlayıcı, sayıcı ve veri kayıtçıları 16 bit'tir (2byte) Z80 CPU 8 bit olduğundan ortam



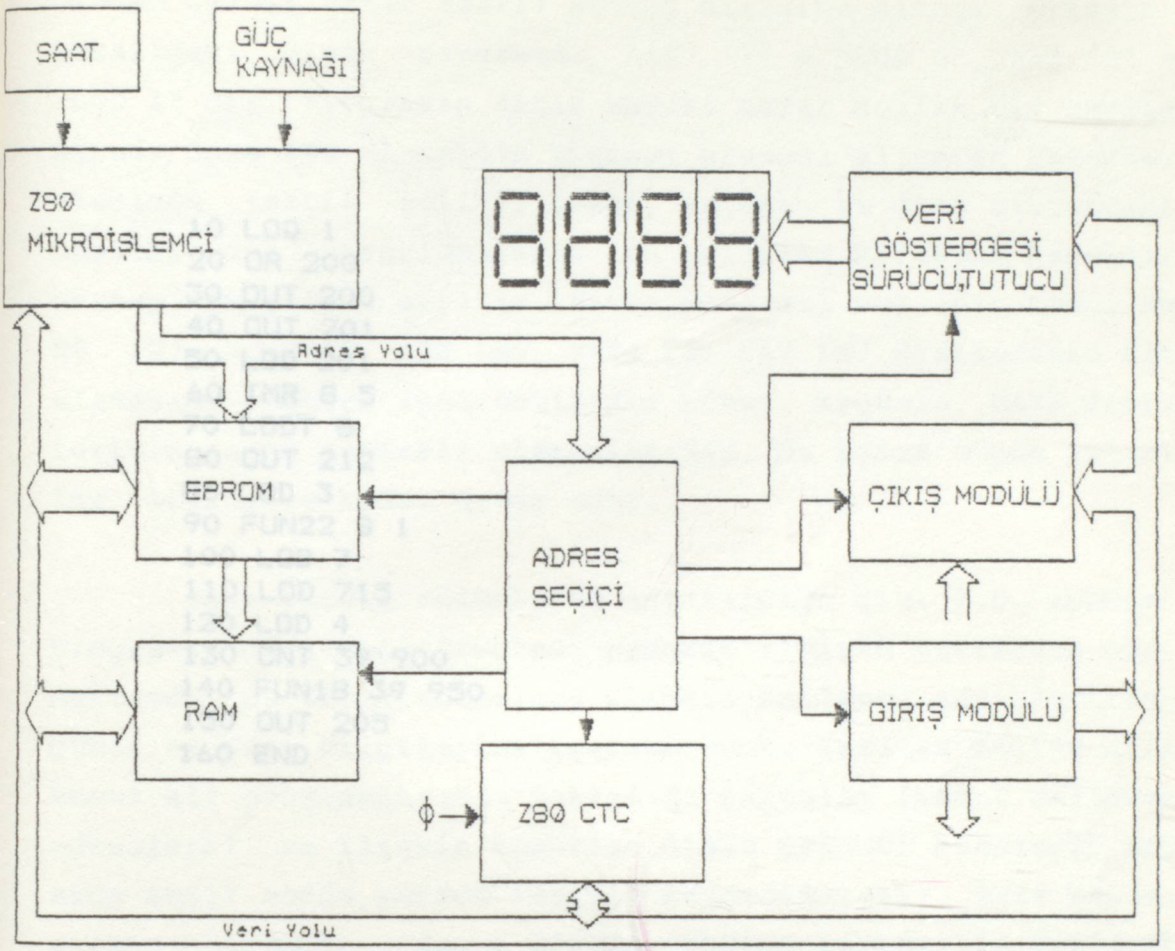
içeriği iki parça halinde göstergeye yansıtılmaktadır. Z80 giriş-çıkış (I/O map) haritasında 2 byte yer tutan donanım, Hex FAE0 ve FAE1 adreslerine yerleşmiştir. Hex FAE0 adresinden ortam içeriğinin ağırlıksız byte'ı ve hex FAE1 adresinden ortam içeriğinin ağırlıklı byte'ı atılmaktadır. Atılan bu datalar, tutucu (LACTH) entegrelerinde saklanarak gösterge sürücülere verilir. Kullanılan 8 bit'lik 2 tutucu (TTL 74LS373) için, 7AE0 ve 7AE1 adreslerinden, seçme (Chip Select) işareti üretilmiştir.

Donanımda herbir dijit için ayrı ayrı sürücüler kullanılmıştır. Programın (dolayısıyla çevrim süresi) uzaması için veri göstergeleri tarama usulü tasarlanmamıştır.

4.5.5. PD DONANIMI

Bu donanım yapısında, derlenmiş program tablosu, zamanlayıcılar, sayıcılar, veri kayıtçıları, girişler, çıkışlar ve dahili rölelerin durumları için RAM bellek bulunmaktadır. Ayrıca, PD sistem programının yazılacağı bir EPROM belleğe ihtiyaç vardır. Zamanlayıcılar için temel süre Z80-CTC kullanılarak sağlanmıştır. Fiziki girişler ve çıkışlar, donanım yapısında 8'li paketler halinde yer almaktadır. Her bir paket Z80 I/O haritasında bir byte yer tutmaktadır. Çıkışlar için tutucu (lacth) entegre devreler, girişler için ise üç konumlu yükseltici entegre devreleri kullanılmıştır. İlgili paketlerden okuma ve yazma yapılabilmesi için adres seçici devreler kullanılmıştır.

FUN22 komutu için harici veri göstergesi devresi tasarlanmıştır. Bu donanım yapısında, tutucu (latch) entegre devreleri, display sürücü entegre devreleri ve display'ler bulunmaktadır. Giriş-Çıkış modülleri, Hex FBE0, FBEF adreslerinin kapladığı 16 byte'lık bölgeye, Z80 CTC Hex FBF0 - FBF3 adreslerinin kapsadığı 4 byte'lik bölgeye ve harici veri göstergesi ise Hex FAE0 ve FAEI adreslerinin oluşturduğu 2 byte'lik I/O bölgesine yerleştirilmiştir.



5. 6128 AMSTRAD MİKROBİLGİSAYARI İLE PD SİMÜLASYONU

5.1 DENETLEME PROGRAMI YAZILIMI

Bu kısımda, P.D için yazılmış denetleme programının oluşturulması anlatılacaktır. İstenilen işleve göre oluşturulan program, Amstrad mikrobilgisayarının basic editor programı kullanılarak, tıpkı bir basıc program yazarmış gibi yazılır. Yazılan bu program basic yorumlayıcı için anlamlı değildir. Oluşturulan bu program, basic programlama diliyle yapılan, P.D. derleyici programı tararından işlenir.

```
10 LOD 1
20 OR 200
30 OUT 200
40 OUT 201
50 LOD 201
60 TMR 8 5
70 LODT 8
80 OUT 212
85 LOD 3
90 FUN22 8 1
100 LOD 7
110 LOD 715
120 LOD 4
130 CNT 39 900
140 FUN18 39 950
150 OUT 205
160 END
```

Program yazılırken yukarıda görüldüğü gibi komutlar argümanlar ve data değerleri arasında boşluklar bırakılır. Uygun yazılım şekliyle oluşturulan program, ASCII modda diske FILE dosya ismiyle kaydedilir. Bu işlem için SAVE "FILE", A Basıc komutu kullanılır.

5.2. PD DERLEYİCİ PROGRAMI

Bu program, yazılarak diske ASCII modda kaydedilmiş, denetleme programından yararlanarak, daha önce sözü edilen tablo yapısını oluşturmaktadır. Önce Oluşturulan makina dili P.D. sistem programı diskten okunarak, Amstrad bilgisayarın belleğine Hex 6000 adresinden başlayarak yerleştirir. Daha sonra ASCII modda diske yüklenmiş, PD denetleme programı, OPENIN "FILE" komutuyla açılan dosyadan INPUT AS$ (I) komutuyla bir çevrim içinde satır okunur.

Okunan herbir satır AS$(I) string dizisine atanır. Örneğin yukarıdaki örnek programda, AS$ (1) = "LOD O", AS$ (2) = "LOD 1" dir. Programın satır sayısı kadar açılan, bir çevrim içinde, önce AS$ dizisinin birinci elemanı alınarak karekter uzunluğu tesbit edilir. Komut, argüman ve data arasındaki boşluklardan yararlanılarak bu bilgiler, karekter, karekter okunup parçalara ayrılır. Satır numarası NUM (N), komut MN E$ (N), argüman ARG (N), data ise DAT (N) dizilerinin ilk elemanına atanır. Aynı satırdaki komut, argüman, data dizilerinin aynı numaralı elemanlarıdır. Bu işlem bütün satırlar bitinceye kadar devam eder.

Daha önceki bölümlerde anlatıldığı gibi P.D. işletim programı, derlenmiş program, program tablosu üzerinden dönmekteydi. Bu tablo komutlara ilişkin dallanma adresleri, argüman ve data bilgilerini içermekteydi. Yazılan makina dili komut alt programlarının başladığı adresler (komut dallanma adresleri) ve ilişkin komutlar, diske OPENOUT komutuyla, ard arda ASCII modda, MNEMON ismiyle kaydedilmiştir. İşte kaydedilmiş bu ASCII dosya OPENIN "MNEMON komutu ile açılarak INPUT#9, A$(S), B(S) basıc komutuyla bir çevrim halinde okunur. A$(S) dizisinde komut, B(S) disinde ise onlara ilişkin dallanma adresleri bulunmaktadır.

FILE dosyasından elde edilen MNE$ (N) dizisi elemanları, teker, teker, MNEMON dosyasından elde edilen A$ (S) dizisi elemanlarıyla karşılaştırılır. Eşitliğin sağlandığı anda o komuta ait dallanma adreside B (S) dizisinin aynı numaralı elemanı içinde bulunmaktadır. POKE basıc. komutu aracılığıyla, tablonun başladığı ilk adres olan, Hex 8000 adresli bellek gözüne B (S) değeri yerleştirilir. Bu değer 2 byte olduğundan, Hex 8000 ve Hex 8001 adresine yerleştirilir. Hex 8002 ve Hex 8003 adresli bellek gözüne aynı numaralı ARG (S), Hex 8003 ve Hex 8004 adresli bellek gözüne ise aynı numaralı DAT (S) dizisi elemanları yerleştirilir. MNE$ (N) dizisin bütün elemanları bitinceye kadar, bu karşılaş-

tırma ve yerleştirme işlemi devam eder. Böylelikle yazılan PD programı, derlenerek, tablo şekline dönüştürülür. CALL basıc komutu aracılığıyla daha önce Amstrad belleğine yerleştirilen PD sistem programına dallanılır ve PD denetleme programı çalışmaya başlar.

P.D DERLEYICI PROGRAMI

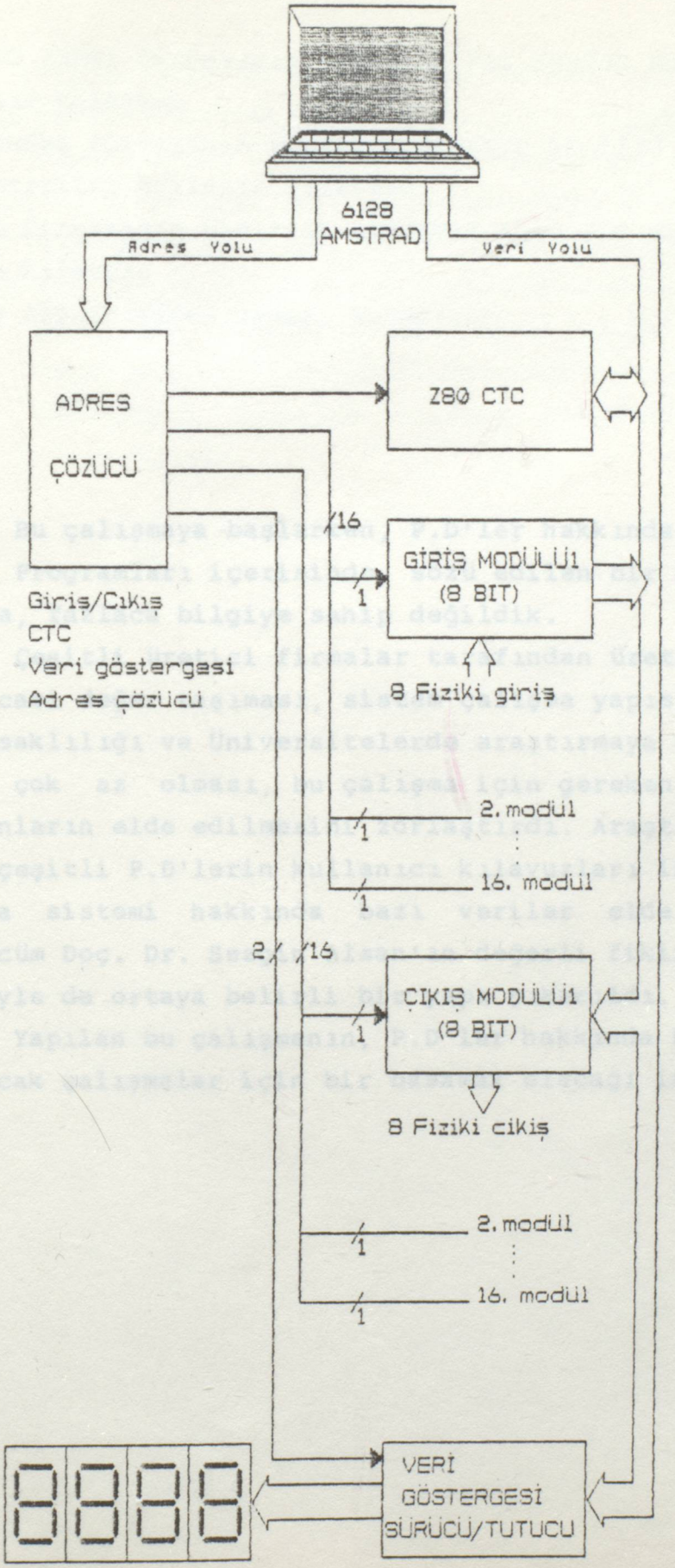
```
10 MODE 0
20 LOCATE 9,12:PRINT"PLC
30 FOR I=&FBE0 TO &FBEF
40 OUT (I),0
50 NEXT
60 BORDER 9
70 MEMORY &3FFF:POT=0
80 OUT &FAE1,0:OUT &FAE0,0
90 LOAD"PLCS.BIN"
100 LOAD"FUNYENI.BIN"
110 MODE 2
120 PRINT CHR$(7)
130 LOCATE 68,21:PRINT"LOAD
140 I=0:T$="":DIM AS$(100),MNE$(100),NUM(100)
,ARG(100),DAT(100),A$(100),B(100)
150 OPENIN "FILE
160 INPUT #9,AS$(I)
170 IF EOF THEN CLOSEIN:GOTO 190
180 I=I+1:GOTO 160
190 FOR N=0 TO I
200 AS$(N)=AS$(N)+" "
210 P=LEN( AS$(N))
220 K=1:T$=""
230 E$=MID$(AS$(N),K,1)
240 IF E$<>" " THEN 260
250 IF T$<>"" THEN  NUM(N)=VAL (T$):T$="":GOT
0 270 ELSE K=K+1:GOTO 230
260 T$=T$+E$:K=K+1:GOTO 230
270 IF K>P THEN 420
280 E$=MID$(AS$(N),K,1)
290 IF E$<>" " THEN 310
300 IF T$<>"" THEN  MNE$(N)=T$:T$="":GOTO 320
 ELSE K=K+1:GOTO 270
310 T$=T$+E$:K=K+1:GOTO 270
320 IF K>P THEN 420
330 E$=MID$(AS$(N),K,1)
340 IF E$<>" " THEN 360
350 IF T$<>"" THEN ARG(N)=VAL(T$):T$="":GOTO
370 :ELSE K=K+1:GOTO 320
360 T$=T$+E$:K=K+1:GOTO 320
370 IF K>P THEN 420
380 E$=MID$(AS$(N),K,1)
390 IF E$<>" " THEN 410
400 IF T$<>"" THEN DAT(N)=VAL(T$):GOTO 420 :E
LSE K=K+1:GOTO 370
410 T$=T$+E$:K=K+1:GOTO 370
```

```
420 NEXT
430 PRINT CHR$(7)
440 LOCATE 68,21:PRINT"COMPILE
450 S=0
460 OPENIN"MNEMON"
470 INPUT #9,A$(S),B(S)
480 IF EOF THEN CLOSEIN:GOTO 500
490 S=S+1:GOTO 470
500 FOR K=0 TO I
510 FOR YT=1 TO S
520 IF MNE$(K)=A$(YT) THEN 540
530 NEXT:CLS: PRINT" YAZILIM YANLISI ..." :GO
TO 740
540 NEXT K
550 FOR K=0 TO I
560 UP=K*6+&8000
570 GOSUB 710
580 GOSUB 690
590 POKE (UP),ADL:POKE (UP+1),ADH
600 ADD=ARG(K)
610 GOSUB 690
620 POKE (UP+2),ADL:POKE (UP+3),ADH
630 ADD=DAT(K)
640 GOSUB 690
650 POKE (UP+4),ADL:POKE (UP+5),ADH
660 NEXT
670 LOCATE 68,21:PRINT"SCAN     "
680 CALL &6000
690 ADH=INT(ADD/256):ADL=ADD-ADH*256
700 RETURN
710 FOR PO=1 TO S
720 IF A$(PO)=MNE$(K) THEN ADD=B(PO):RETURN

730 NEXT: PRINT" YAZILIM YANLISI ..." :RETURN
740 STOP
```

5.3. SİMÜLASYON İÇİN YAPILAN DONANIM

Amstrad mikrobilgisayarının Expansion soketine bağlanan donanımda, Z80-CTC, giriş-çıkış modülleri ve harici data göstergeleri bulunmaktadır. Bu donanım yapıları daha önceki bölümlerde anlatılmıştır. Hazırlanan donanım blok şema olarak şekil 5.3.1. de gösterilmiştir.


6128
AMSTRAD
Adres Yolu
Veri Yolu
ADRES
ÇÖZÜCÜ
Z80 CTC
16
GİRİŞ MODÜLÜ1
(8 BIT)
1
Giriş/Cıkış
CTC
Veri göstergesi
Adres çözücü
8 Fiziki giriş
2. modül
16. modül
2
16
ÇIKIŞ MODÜLÜ1
(8 BIT)
8 Fiziki cikiş
2. modül
16. modül
VERİ
GÖSTERGESİ
SÜRÜCÜ/TUTUCU

Bu çalışmaya başlarken, P.D'ler hakkında, Yüksek Lisans Programları içerisinde sözü edilen bir kaç noktanın dışında, fazlaca bilgiye sahip değildik.

Çeşitli üretici firmalar tarafından üretilen P.D'lerin ticari değer taşıması, sistem çalışma yapısı hakkındaki bilgi saklılığı ve Üniversitelerde araştırmaya ayrılan fonların çok az olması, bu çalışma için gereken materyel ve dökümanların elde edilmesini zorlaştırdı. Araştırmada, üretilen çeşitli P.D'lerin kullanıcı kılavuzları incelenerek, çalışma sistemi hakkında bazı veriler elde edildi. Tez yürütücüm Doç. Dr. Sezgin Alsan'ın değerli fikir ve eleştirileriyle de ortaya belirli bir yapı çıkarıldı.

Yapılan bu çalışmanın, P.D'ler hakkında bundan sonra yapılacak çalışmalar için bir basamak olacağı inancındayım.

KAYNAKÇA

1- IDEC IZUMI Firmasının ürettiği FA1 JUNIOR PD'sinin kullanım kılavuzu
2- SIEMENS firmasının ürettiği SIMATIC 55-1014 Programmable controller kullanım kılavuzu
3- AEG Firmasının ürettiği LOGISTAT A020 PLC cihazı kullanım kılavuzu
4- Z80 APPLICATIONS James. W.Coffron

ÖZGEÇMİŞ

1963 yılında Kars-Arpaçay'da doğdum. İlk, Orta ve Lise öğrenimimi Kars'da tamamladım. 1982 yılında başladığım, Yıldız Üniversitesi elektronik ve Haberleşme Mühendisliği bölümünden 1986 yılı yaz döneminde mezun oldum. Aynı yıl Yıldız Üniversitesi Fen Bilimleri Enstitüsünün açtığı Yüksek Lisans sınavını kazanarak, Elektronik bölümünde Yüksek Lisans eğitimine başladım. 25.3.1987 tarihinden buyana Yıldız Üniversitesi Elektronik ve Haberleşme Mühendisliği Bölümü Elektronik Anabilim dalında araştırma görevlisi olarak çalışmaktayım.