

**Rasgele Erişimli Bellek
Arşivini Tasarımı**

Armağan Özkaya

Yüksek Lisans Tezi

YILDIZ ÜNİVERSİTESİ
GENEL KİTAPLIĞI

R 368

Kot : 44
Alındığı Yer : Fen Bilimleri Bostb
Tarih : 13/05/1991
Fatura :
Fiatı : 3500 TL
Ayniyat No : 1/3
Kayıt No : 47645
UDC : 001.64 378.242
Ek :

YILDIZ ÜNİVERSİTESİ

D.B. No. 45372

YILDIZ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

RASGELE ERİŞİMLİ BELLEK ARABİRİMİ TASARIMI

YÜKSEK LİSANS TEZİ
MÜH. ARMAĞAN ÖZKAYA

İSTANBUL 1987

ONSÖZ ;

Bilim ve teknoloji alanında yüzyıllar boyu devam eden gelişme, özellikle 19. ve 20. yüzyıllarda yapılan önemli aşamalarla büyük bir hız kazanmıştır.

Tekniğin gelişmesinde iki farklı aşama göze çarpar. Oldukça uzun bir tarihe sahip olan birinci aşamada daha çok bedeni yeteneklerin, birinciye göre yeni sayılan ikinci aşamada ise beyin gücünün yerini alabilecek araçların geliştirilmesi ile uğraşılmıştır. Bu araçların gerek işlem, gerekse yapım esasları bazı teorik ilkelere dayanır. Bunlar arasında özellikle bilişim teorisi çok geniş konulu bir bilim dalı olarak son 25 - 30 yılda gelişmiş olup, geniş anlamda bilginin ölçülmesi, işlenmesi, saklanması ve iletilmesi gibi olayları inceler.

19. yüzyılda başlayan endüstrileşme hareketinin doğal bir sonucu olarak, insan ile ilgili sosyolojik ve teknolojik sistemlerin eylemleri daha karmaşık bir durum göstermeye başlamıştır. Bilgi toplama metodları gelişmiş, hızla toplanan bilgilerin çokluğu karşısında bunları zamanında işlemek için insanın yetersiz kaldığı görülmüştür. Bu nedenle bilişim araçları teorik bir araştırma konusu olmaktan çıkıp, pratik amaçlara hizmet eden araçlar sınıfına girmiştir.

1642 yılında Pascal'ın en ilkel hesap makinasını gerçekleştirmesi bugünkü bilgisayar teknolojisine atılan ilk adım olmuştur. Pascal'dan sonra 1833 de Charles Babbage'in önerdiği ancak o zaman gerçekleştirilemeyen bir makina, sabit bir dizge ve bunu yönlendirecek bir bellek kavramı ortaya koyması nedeniyle bugünkü bilgisayarların öncüsü sayılmaktadır. Babbage'in önerip gerçekleştiremediği bu makina ancak 1937 de Harvard Üniversitesi'nde gerçekleştirilmiştir.

Bu tarihler arasında daha birçok teknolojik gelişme oldu. 1854 yılında George Boole, Düşünce Kanunlarını yayınladı. Bu, bilgisayarların gelişmesinde önemli bir aşamadır. İlk elektronik bilgisayar düşüncesi 1919 yılından sonra ortaya atıldı. Bunun nedeni de, bu tarihte Eccles ve Jordan'ın "flip-flop" u bulmuş olmalarıdır. 2. Dünya Savaşı bilgisayar teknolojisine büyük katkıda bulundu. 1946 Şubat ayında Pennsylvania Üniversitesi'nde ENIAC (Electronic Numerical Integrator and Calculator) adı verilen makina gerçekleştirildi. 100 KW. güç harcayan ENIAC'ın 18800 tüpü, elle lehirlenmiş yarım milyondan fazla bağlantısı vardı. 30 metre boy ve yaklaşık 30 ton ağırlığa sahipti. Sayıların işleniş onluk

düzende idi ve toplama işlemi 200 μ s., çarpma işlemi 2300 μ s. süre alıyordu.

ENIAC' ın yapımı sürerken Dr.Von Neumann ve arkadaşları EDVAC (Electronic Discrete Variable Automatic Computer) adını verdikleri 4096 bellek gözü içeren bilgisayarı gerçeklediler. Bu bilgisayardaki veri ve program saklamanın bellekte yapılması görüşü, günümüzde hala kullanılmaktadır.

1953 te, bu zamana kadar bilgisayar konusuyla ilgilenme yen IBM, deneme amacıyla bu piyasaya girmiş ve ilk olarak IBM 701 i piyasaya sürmüştür. Bunu 1955 te IBM 704 izledi. IBM 704 ile yüksek düzeyde bir dille programlama kavramı getirilmiştir. Bu makinada kullanılan programlama dili FORTRAN' dı.

1960 da transistörlerin, 1966 da ise tümdevrelerin yapılmaya başlanması sonucu, devrelerin güvenilirlikleri artarken maliyetleri düşmüştür. 1970 lere gelindiğinde ilerleyen yarıiletken teknolojisi Merkezi İşlem Birimini (MIB) tek kırkık içine sığdırabilecek düzeye gelmişti. İlk mikroişlemci Intel 4004 4 bitlik olup, 1971 yılında piyasaya sürüldü. Aynı firma 1973 yılında 8 bitlik 8080' i piyasaya sürdü. Bu akıma diğer tümdevre yapımcıları da katıldılar ve Motorola firması 6800 ü üretti. 1980 lere gelindiğinde üç mikroişlemci, I 8085, M 6802, Z 80 endüstri standardı olarak yerlerini aldılar.

1980 de 16 bitlik mikroişlemciler devri başlamış ve bu alanda başı çeken üretimler I 8086, I 8088, M 68000, Z 8000 olmuştur.

Günümüzde mikroişlemciler teknolojinin hemen tüm alanlarında kullanılmakta, karmaşık işlemlerin hızla, ucuz maliyetlerle yapılmasını sağlamaktadırlar. Bu çalışmada da mikroişlemci , çevrebirimleri ile birlikte en büyük görevi üstlenmektedir.

Çalışmamda destek olan hocam Doç.Metin Yücel ve I.T.Ü araştırma görevlilerinden Y.Müh.B.Tevfik Akgün'e teşekkür ederim.

Armağan Özkaya

Istanbul , Temmuz-1987

5.2.1.1	Arabirim Sistemi Bellek Düzeni	...	38
5.2.1.2	Arabirim Sistemi Çevre Birimleri ve Kullanımı	39
5.2.2	Mıknatıslı Şerit Mekanizması Kontrol Düzeni	..	44
5.2.2.1	Sarma ve Okuma/Yazma Kontrolü	44
5.2.2.2	Devir Hızı ve Kararlılığının Kontrolü		47
5.2.2.3	Kaset Varlığı, Yazma Koruması ve Konum Bilgisi Sezme	49
5.2.2.4	Bilgisayar ve Okuma/Yazma Amplifikatörleri ile İletişim	50
5.2.3	Veri Yazma ve Okuma Düzeni	52
5.2.3.1	Veri Kodlama Yöntemi	52
5.2.3.2	Veri Yazma Devresi ve Yöntemi	...	53
5.2.3.3	Verinin Okunması ve Kontrolü	55
5.3	Kaset Paylaşımı ve Formatlama	56
5.4	İşletim Sistemi	62
5.4.1	B"Kaset Adı" (Başla) Komutu	62
5.4.2	L (Listele) Komutu	63
5.4.3	D"Dosya Adı" (Oku) Komutu	63
5.4.4	Y"Dosya Adı" (Yaz) Komutu	63
5.4.5	S"Dosya Adı" (Sil) Komutu	64
5.4.6	T (Düzenle) Komutu	64
5.4.7	K"Dosya Adı" (Kopyala) Komutu	65
5.4.8	F"Kaset Adı" (Formatla) Komutu	65
5.5	Sistem Yazılımı	66
5.5.1	Arabirim Yazılımı	66
5.5.2	Bilgisayar Yazılımı	80
6.	SONUC	81
EKLER			
1.	MPF.1P Sistemi Donanım Şeması	83
2.	MPF.1P Sistemi Giriş/Çıkış Kartı Donanım Şeması	88
3.	Arabirim Sistemi Besleme Kaynakları	90
KAYNAKLAR			
.....			
91			
ÖZGEÇMİŞ			
.....			
92			

ÖZET ;

Bilgisayarlar büyük miktarlarda verileri saklamak amacıyla magnetik saklama ortamlarını kullanırlar. Bu amaç için mıknatıslı şerit düzenleri kullanan sistemlerde, erişim olayının ardışıl olması kötüdür.

Bu çalışmada bir ses kasetindeki veri dosyalarına rasgele erişebilen bir sistem tasarlanmak istenmiştir. Böylece ses kaseti, tıpkı bir disket gibi kullanılmış olacaktır.

Sistemin anlatımına girilmeden önce, yarıiletken ve magnetik veri saklama ortamları, magnetik ortamlarda veri saklamada kullanılan kayıt yöntemleri, veri saklamada ve iletiminde kullanılan kodlama yöntemleri konularında genel bilgiler verilmiştir.

Tasarlanan sistemde kasetin tek yüzüne yüklenebilecek mutlak veri miktarı 1535.68 KB. dır.Yükleme işlemi sırasında verinin kodlanması, gerçek veri miktarını her yüz için 629100 karakter olarak düşürür. Her kaset yüzü 8 KB.lık 72 bölüme ve kaset adı, liste yapısı, boş bölümler yapısı için kullanılan bölümlere ayrılmıştır. Böylece her yüzde, tek kanal için 576 KB. veri yazma imkanı vardır.

Çift kanal ve çift kaset yüzü kullanıldığında veri kapasitesi $4 \times 576 \text{ KB.} = 2304 \text{ KB.}$ olacaktır.

Oluşturulan sistemin donanımı çift kanal için gerçekleştirilmesine rağmen, işletim sistemi programları tek kanal kullanımı için yazılmıştır.

SUMMARY ;

To store large quantities of data in computers, use is made of magnetic storage units. In systems where magnetic tape storage is employed, sequential access is considered unsuitable.

In this study, it was aimed to design a system which can provide random access to data files in audio cassettes. Thereby an audio cassette would be used just like a discette.

Before describing the system, general information is given on data storage media of the semiconductor and magnetic types, recording methods used for data storage in magnetic media and coding methods used for data storage and transmission.

The system which has been devised under this study provides an absolute data loading capacity of 1535.88 KB on one side of the cassette. The real data capacity reduces to 629100 characters/side due to data coding that has to be carried out during the loading process. On each side of the cassette, there are 72 sections intended to accommodate 8 KB each, plus 3 sections assigned for the cassette name, file layouts and the empty sections list. This means a data loading possibility of 576 KB per channel per side.

Therefore, when two sides and two channels are used, the loading capacity will be $4 \times 576 \text{ KB} = 2304 \text{ KB}$.

• Although the system hardware was developed for a double channel application, the operating system programs were written out for the use of a single channel only.

1 - GİRİŞ

Bilindiği gibi tüm mıknatıslı şerit düzenleri ardışıl erişimli (sequential) çalışırlar. Bu durum özellikle yavaş sistemlerde büyük sıkıntı oluşturur. Aynı zamanda mıknatıslı şerit düzenlerinde erişim olayının olduğu kadar, okuma ve yazma işlemlerinin yavaş olması da büyük sıkıntı kaynağıdır.

Veri güvenliğinin sağlanması için belli veri yazma yöntemlerinin kullanılması ve veri yazma hızının düşük tutulması nedeniyle, mıknatıslı şeritlerin veri kapasitesi de sınırlanmaktadır. Örnek olarak bu sistemin tasarımında kullanılan MPF.1P mikrobilgisayarında Multitech firması tarafından oluşturulan veri yükleme yöntemi ile 1 sekizlik veri 60 ms. de yüklenmektedir. Böylece bir kasete (60 dk.) ancak 58.6 KB. lık veri yükleme imkanı vardır.

Tüm bu dezavantajları nedeniyle disket birimleri doğal olarak mıknatıslı şerit birimlerine tercih edilmektedir.

Bir mıknatıslı şerit biriminde uygun erişim, yazma ve okuma hızı, veri kapasitesi ile birlikte disket birimlerine tercih edilebilecek maliyet fiyatı sağlanabilirse, böyle bir birimin kullanımı mümkün olabilir.

Mıknatıslı şerit ile kullanıcı bilgisayar arasında bu işlemleri yapan bir "mıknatıslı şerit (kaset) sürücüsü" tasarlanabilir. Böyle bir birimin tasarımında, gerekli çevre birimlerini içeren bir mikroişlemci sistemi kullanılmalıdır. Bu amaçla eğitim amacıyla hazırlanmış olan MPF.1P mikrobilgisayarı kullanılmıştır. Z 80 mikroişlemci temelli olup, giriş/çıkış kartında paralel ve seri iletişim, sayma ve zamanlama işlemleri için çevre birimleri vardır.

Mıknatıslı şerit mekanizmasının elektriksel kontrollü olması gerektiğinden, tüm işlemleri rölelerle yapılan bir kaset mekanizması kullanılmıştır.

2 - VERİ SAKLAMA ORTAMLARI

Bellekler bilgisayar sistemlerinin vazgeçilmez bölümüdür. Bellek erişim hızı, kapasitesi ve maliyeti sistem için uygun bellek seçiminde temel etkindir.

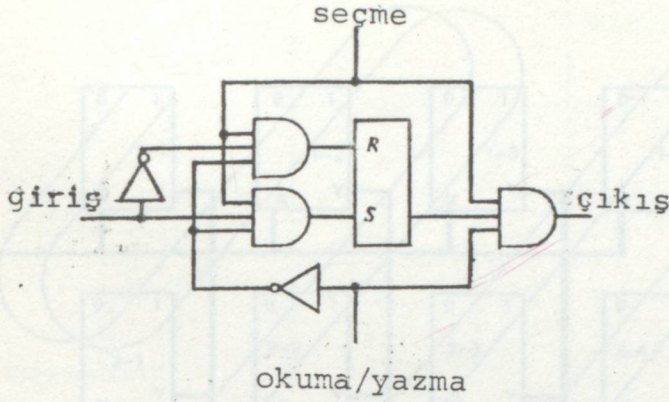
2.1 - Yarıiletken Ortamlar

Yarıiletken bellekler LSI (Large Scale Integrated) teknolojisi ile yapılmış ucuz, kullanışlı, hızlı belleklerdir. Çeşitli amaç ve tiplerde olabilirler. Bellek düzenlemesi sekizli veya sözcük düzeninde olmasına karşın bellek büyüklüğü genellikle bit sayısı ile ölçülür. Bir bellek w sözcüğe ve her sözcük b bite sahipse, kapasitesi $w*b$ bit olacaktır.

OYB (Rasgele Erişimli Bellek - Oku Yaz Bellek) : Tüm bellek gözlerine erişimler aynı süreleri alırlar. Diğer belleklerde bu süreler verinin fiziksel yerine bağlıdır. Temel dezavantajları verinin kalıcı olmamasıdır. OY bellekler statik ve dinamik çalışabilirler. Statik OYB'lerde bir flip-flop bir bitlik bellektir. Eşdeğer lojik devresi Şekil-2.1 de görülmektedir. Seçme girişi göze veri yazmaya yada okumaya izin verir. Yazma / Okuma girişi ile işlem türü seçilir. Bu giriş "0" olduğunda bilgi girişindeki veri flip-flop çıkışına transfer edilir.

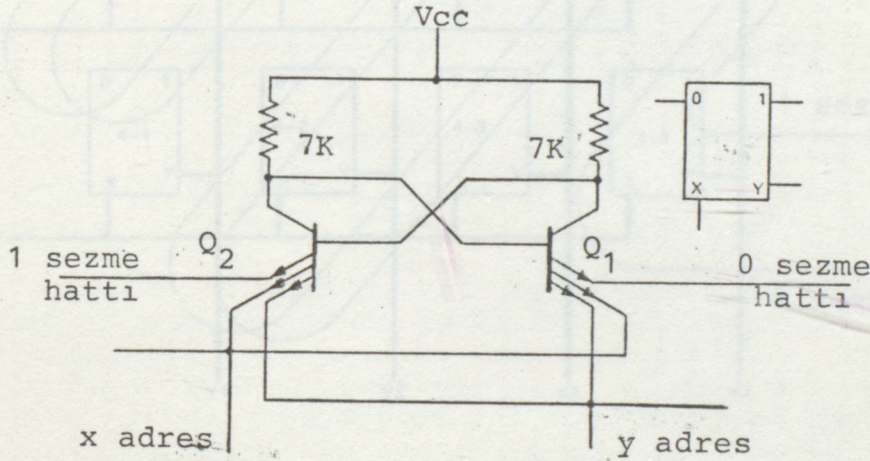
Dinamik OYB'lerde şarj edilmiş bir kapasite elemanı aynı görevi görür. Bu yüzden bu tür bellekler belli aralıklarla tazelenecek şekilde tasarlanırlar. OY bellekler bipolar, MOSFET (p veya n tip) elemanlardan ve I^2L (Integrated Injection Logic) teknolojisi ile yapılırlar. Bipolar bellekler 20 ns. ve MOS bellekler 200 ns. civarında bellek erişim süresine sahiptirler. I^2L bellekler yaklaşık bipolar bellekler kadar hız ve MOS bellekler kadar tümleştirme yoğunluğu sağlarlar. I^2L ve MOS bellekler aynı zamanda oldukça düşük güç harcamaları açısından avantajlıdırlar. Bipolar bellekler az veri kapasitesi ve yüksek hız isteyen sistemlerde veya özel uygulamalarda, MOS bellekler ise yüksek kapasiteli ana belleklerde kullanılırlar.

Bipolar belleklere bir örnek olarak 74S201 (256*1 Memory Array) tümdevresi gösterilebilir. Tipik erişim zamanı 42 ns. ve güç tüketimi 1.8 mW/bit dir. Intel 8102 (1K*1 MOS Memory tümdevresi) MOS OY belleklere örnek olarak verilebilir. Erişim süresi 450 ns. ve güç tüketimi 7 μ W/bit dir.



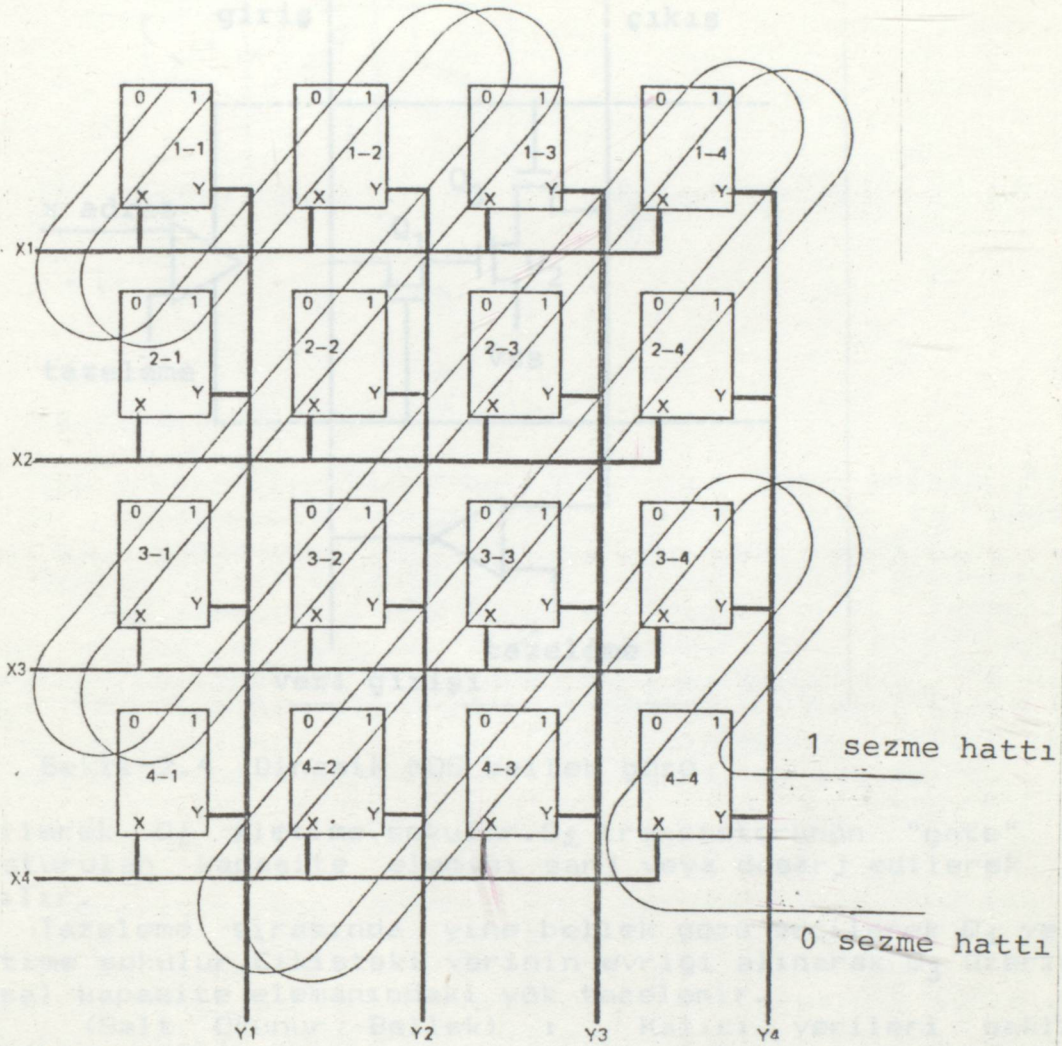
Şekil-2.1 Bir bitlik bellek gözü

Şekil-2.2 de bir bitlik bipolar OY bellek gözünün yapısı verilmiştir.



Şekil-2.2 Bipolar OYB bellek gözü

Şekil-2.3 de ise bu yapıda 16*1 bellek elemanının düzenlenmesi görülmektedir. Göze yazılan veriye bağlı olarak transistörlerin bir tanesi iletimde, diğeri kesimdedir. Örneğin; Bu bellek gözünün seçildiğini ve "1" yazılmak istendiğini düşünelim. X ve Y seçme hatları "H" ve 1 sezme hattı da "L" olacaktır. Böylece Q₁ iletime geçecek, kollektor gerilimi düşecek, bu nedenle Q₂ kesimde kalacaktır. Okuma işlemi sırasında 0 veya 1 sezme hatlarının aktif olma durumuna göre çıkışta veri görülecektir. Başka bir bellek gözünün seçilmesi durumunda seçme hatları "L" olacak, ancak Q₂ yine kesimde, Q₁ iletimde kalacaktır.

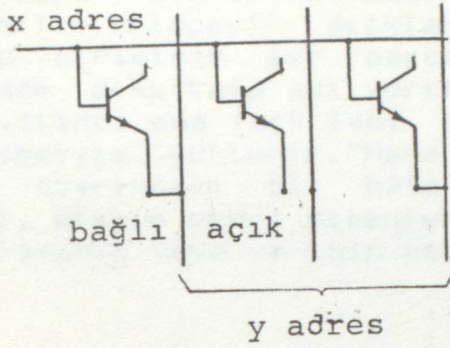


Sekil-2.3 16*1 bellek elemanı düzeni

Dinamik belleklere örnek olarak M41163 verilebilir. 16K*1 şeklinde düzenlenmiştir. Aktif durumda 26 $\mu\text{W}/\text{bit}$, bekleme (standby) durumunda 1.3 $\mu\text{W}/\text{bit}$ güç harcaması vardır. 200 ns. lik erişim hızına sahiptir ve verilerin her 2 ms. de bir tazelenmesi gerekir. [1]

Bir dinamik bellek gözü Şekil-2.4 de gösterildiği gibi üç tane MOS transistörden oluşur. Eleman azlığı aynı büyüklükteki bir yongaya daha çok bellek gözünün sığdırılmasını sağlar. Ancak verilerin birkaç ms. aralıklarla tazelenmesi gerekir.

Devrede veri Q_2 transistörünün "gate" inde bir yük birikimi olarak saklanır. Verinin yazılması için bu bellek gözü



Sekil-2.5 Bipolar SOB yapısı

MOS yalnız okunabilir bellekler daha düşük erişim hızına ve güç harcamasına sahiptirler. (Örnek olarak bu değerler Intel 8316 tümdevresi için 500 ns. ve 10 μ W /bit dir.)

Verisi kızılötesi ışınlarla silinip tekrar programlanabilen yüksek kapasiteli MOS SOB ler yapılmıştır. İstenildiği kadar yeniden programlama işlemi yapılabilir. MOSFET transistörlerin çalışmasına izin veren yada vermeyen bir "floating gate" ile verinin saklanması sağlanır. "Gate" elektriksel olarak bağımsızdır ve eğer şarj edilmişse, transistörde "drain" ile "gate" arasında kalıcı bir kanal sağlanır.

2.2 - Magnetik Ortamlar

Genellikle tüm mikroişlemci sistemlerinde ikincil bir veri saklama birimine ihtiyaç duyulur. Bu tür sistemler dışında ki sistemlerde SOB birimlerinde saklanan sabit program parçaları veya kullanıcı tarafından yazılarak OYB birimlerine transfer edilen programlar sistemi sınırlar. Her iki yöntemde de yeni program geliştirme veya ticari programların kullanımı durumunda sistem verimliliği düşer.

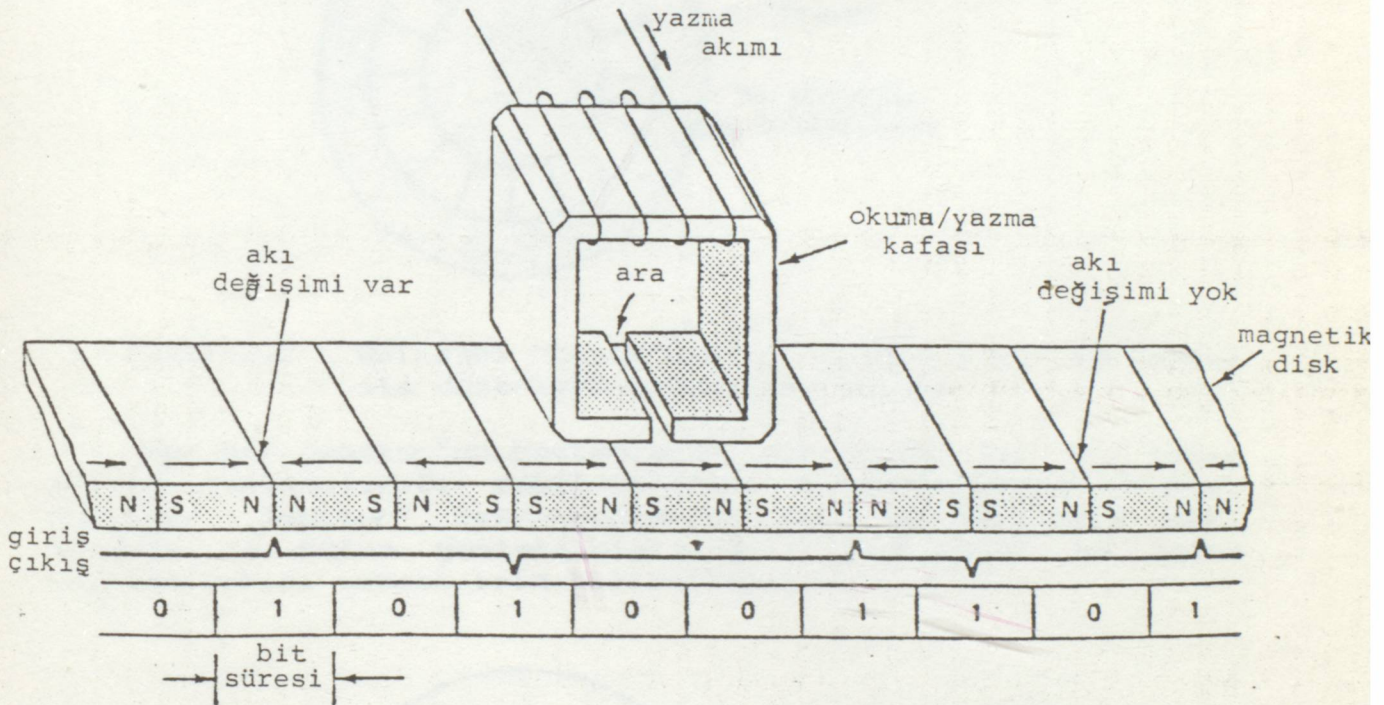
Bu sebeplerden dolayı magnetik saklama ortamları kullanılırlar.

2.2.1 - "Floppy" ve "Rigid" Disk Teknolojileri

Bu birimler 100 KB. dan 60 MB. a kadar saklama kapasiteli ve veri dosyalarının saklanması, işlenmesi, aranması için elverişli birimlerdir.

Bilgisayar sistemlerinde "floppy" disk sürücülerini ve "rigid" disk sürücülerini olmak üzere iki tip disk sürücüsü vardır. İkisi arasındaki prensip farkı kullanılan iletişim ortamıdır. Bir "floppy" disk demir-oksit kaplanmış ince bir Mylar diskten yapılmıştır. Diskin esnek olması nedeniyle "floppy" ismi verilmiştir.

"Hard" disklerde demir-oksit kaplanmış bir metal disk kullanılır."Floppy" disklerin aksine "hard" diskler disk sürücü biriminin bir parçasıdır ve hareketsizdir. Her iki sistemde disk kafa adı verilen bir magnetik eleman etrafında döner. İkinci ana fark ise; bir "floppy" diskte kafa, diskin sürülmesiyle yüklenir. "Hard" diskte kafa disk yüzeyinin 20-50 µinch üzerindeki bir hava yastığında kayar. Bu toleransın sebebi, diskin çevre etkenlerden korunmasıdır. Toz parçacıkları bile kafanın veya verinin bozulmasına sebep olur. [2]

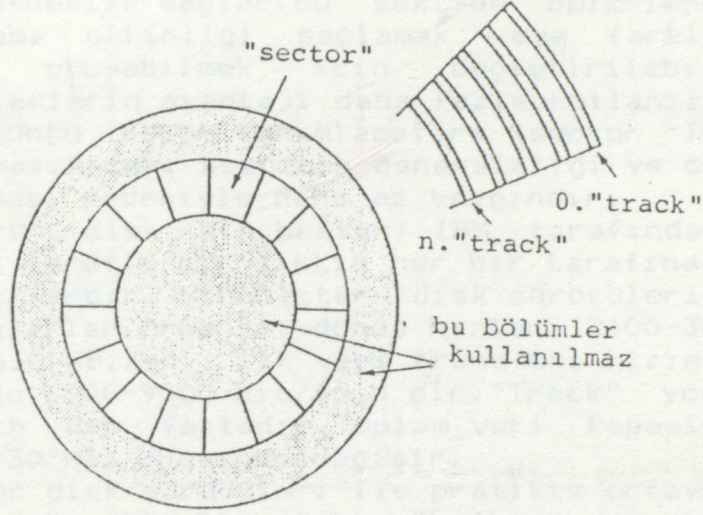


Şekil-2.6 Bir "floppy" veya "hard" disk ile okuma/yazma kafasının konumu.

Her iki tipte de kayıt tekniği benzerdir. Şekil-2.6 da gösterildiği gibi lojik "1" değeri okuma/yazma kafasından akan akımın yönünün terslenmesi ile yazılır. Bu durum demir-oksit üzerindeki magnetik akımın terslenmesini doğurur. Akı yönünün değişmemesi lojik "0" değerini gösterir. Her bir bitlik göz ince bir mıknatıs gibi düşünüldüğünde, iki tane N veya S kutbu bulunduğu zaman bir akı değişimi, dolayısıyla lojik "1" değeri yüklendiği anlaşılır. Disk içeriği okunurken düzenli aralıklarla örnekleme yapılır ve her akı değişimi için lojik "1" gönderilir.

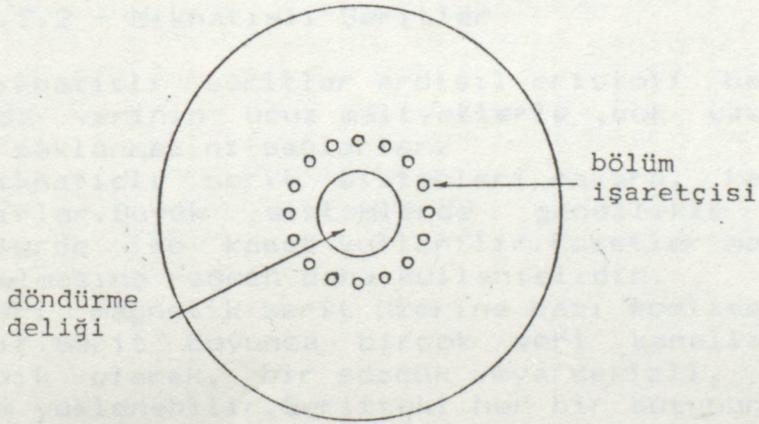
Disk kafa etrafında dönerken "track" denen dar, dairesel bir düzlem taranır. Kafanın disk yarıçapı boyunca hareketi ile

çok sayıda "track" tanımlanabilir.Şekil-2.7 Veri genellikle tüm bir "track" boyunca değilde,her bir "track" ın bölünmesi ile oluşan "sector"lere yüklenir.



Şekil-2.7 Dairesel "track"ların "sector"lere bölünmesi ile düzenlenmiş bir disk yüzeyi

Her bir "sector"ün başlangıcını belirlemek için iki temel teknik kullanılır.Bir sabit bölünmüş (hard-sectored) diskte, delikler "sector" başlangıçlarını belirler.Şekil-2.8 deki örnekte 16 bölüm gösterilmiştir.Bir foto-sezici bu bölüm işaretçilerini sezmek için kullanılabilir.



Şekil-2.8 Bir sabit bölünmüş (hard-sectored) disk ve bölüm işaretçileri

Daha yaygın bir teknik de, her bir "sector" başlangıcına "sector ID" leri yazmaktır. Bu, özel bir format programının, her yeni diske veri yüklenmeden önce kullanılmasını gerektirir. Yöntem sabit bölmeleme yönteminden daha karmaşıktır. Fakat daha fazla esneklik sağlar. Bu şekilde bölünmüş diskler azami depolama etkinliği sağlamak veya farklı "sector" formatlarını okuyabilmek için değiştirilebilirler. Sabit bölünmüş disklerin avantajı daha fazla kullanılabilir bölge sağlamasıdır. Çünkü bölüm tanımlamaları (sector ID) gereksizdir, kullanılmaz. Ancak, sistemin esnekliği ve özel disklere ihtiyaç duyulması nedeniyle daha az yaygındır.

İlk "hard" disk (Winchester) IBM tarafından geliştirilmiştir ve çift taraflı bir diskin her bir tarafına 30 MB. veri yükleme imkanı vardır. Winchester disk sürücüleri daha yüksek performans sağlarlar. Örneğin, dönüş hızları 2400-3600 dev./dk. dir. Böylece 610 KB./sn. lik veri transferi hızına ulaşırlar. Veri yoğunluğu 6000-9000 bit/inch dir. "Track" yoğunluğu ise 500 track/inch den fazladır. Toplam veri kapasitesi sürücü tipine göre 5-30 MB. arasında değişir.

Winchester disk sürücüleri ile pratikte ortaya çıkan bir problem vardır. Bu tip bir diskin "backup" ını almak için 60 dan fazla "floppy" disk gerekir. Problemin çözümüne yaklaşımlardan biri, bir kartuş sistemi kullanmaktır. 0.25 inch lik mıknatıslı şeritler kullanan bir sistem 75 MB. veri saklayabilir ve 1 MB./sn. lik transfer hızına sahiptir. Böyle bir kartuş taşınabilir ve emniyetle kullanılabilir.

Bazı firmalar taşınabilir Winchester birimleri üretmişlerdir. Örnek olarak bir birimde sabit 5 MB. Winchester disk 5 MB. taşınabilir kartuş bulunur. Bu sistemde 10 MB. veri saklama olanağı vardır ve kartuş aynı zamanda "backup" amacıyla kullanılabilir. [2]

2.2.2 - Mıknatıslı Şeritler

Mıknatıslı şeritler ardışıl erişimli belleklerdir. Büyük miktarda verinin ucuz maliyetlerle, çok uzun süre, kalıcı olarak saklanmasını sağlarlar.

Mıknatıslı şerit sistemleri makara, kaset veya kartuş kullanırlar. Büyük sistemlerde genellikle makara, mini sistemlerde ise kaset kullanılır. Kasetler makaralardan daha yavaş olmasına rağmen daha kullanışlıdır.

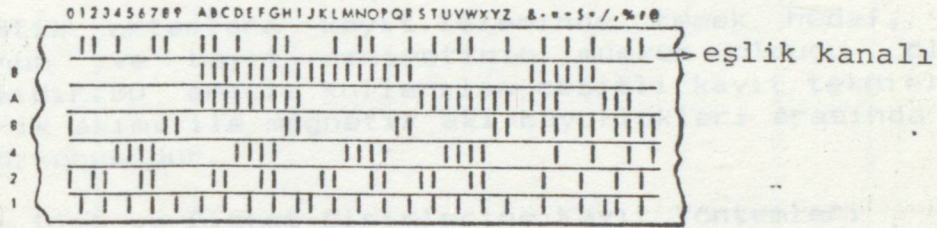
Veri magnetik şerit üzerine bazı kodlama yöntemleri ile yüklenir. Şerit boyunca birçok veri kanalları oluşturulabilir. Tipik olarak, bir sözcük veya sekizli, şerite bir sütun boyunca yüklenebilir. Şeritteki her bir sütunun eşlik kontrolü veya bir kanal boyunca belli sayıda bitin eşlik kontrolü şeklinde iki tip eşlik kontrolü yapılabilir.

Şerit formatlı veya formatsız olabilir. Formatlı şeritlerde veri yerleştirmek için zamanlama veya referans

işaretleri vardır. Formatsız şeritlerde böyle bir referans işareti yoktur ve veri bloğunun başlangıcına başlama işareti konarak veri yüklenir.

Mıknatıslı şerit sistemlerinin özellikleri değişiktir. Başlama ve durma zamanları 1-20 ms. , şerit hızları 18.75-900 inch/sn. olabilir. [3]

Toplam uzunluğu 1800-2400 feet olan mıknatıslı şeritler üzerine 30×10^6 bitden fazla veri yüklenebilir. Karakter yoğunluğu 800-4800 karakter/inch dir. 1 inch genişliğindeki şeritlere 7 yada 8 satır halinde veri yüklemek alışılmış yöntemdir.



Şekil-2.9 Karakterlerin mıknatıslı şerit üzerine yazımı

Şekil-2.9 de 7 bit olarak yüklenmiş karakterler içeren bir mıknatıslı şerit parçası görülmektedir. Kontrol kanalındaki bitler, çift eşlik biti olarak kullanılmıştır. Aynı zamanda bloktaki verilerin toplamı blok sonuna kontrol sözcüğü olarak yazılır.

Bazı mıknatıslı şerit sürücüleri, yüksek çalışma hızı, hızlı başlama ve durma, yüksek okuma ve yazma güvenilirliği gibi özelliklere sahiptir. Sürücü motor sürekli hareket halindedir. Böylece motorun ivmelenme durumu ortadan kaldırılarak şeritin yüksek hızla hareketi sağlanır. Bu birimlerde iki tür yazma/okuma kafası kullanılır. Tek aralıklı kafada bir anda sadece yazma veya okuma yapılabilir, çift aralıklı kafada ise yazılan veri aynı zamanda okunabilir. Bu yazılan verinin kontrolü için bir kolaylıktır.

Yüklenen bloklar arasında yaklaşık 0.75 inch lik bir aralık bırakılır. Bu boşluk şeritin durma ve yerleşme süresini karşılamak içindir. [1]

3 - MAGNETİK ORTAMLARDA VERİ SAKLAMADA KULLANILAN KAYIT YÖNTEMLERİ

Magnetik ortamlara kayıt işleminde temel hedef, veri yoğunluğunun ve kayıt emniyetinin mümkün olduğu ölçüde artırılmasıdır. Bu amaçla kullanılan çeşitli kayıt tekniklerinin de elektrik akımı ile magnetik akı büyüklükleri arasında bir dönüşüm söz konusudur.

3.1 - Disk ve Disket Birimlerine Kayıt Yöntemleri

Disk ve disket birimlerine lojik işaretler doğrudan gerilim seviyeleri olarak yüklenemezler. Ancak lojik "1" seviyesi bir akı değişimi olarak yüklenebilir. Akı değişimi olmaması lojik "0" değerine karşı düşer.

Magnetik disklerde, UART (Universal Asynchronous Receiver Transmitter) birimlerinde kullanılan başlangıç bitinin sezilmesi ve verinin bit süresinin ortasında örneklenmesi yöntemi kullanılamaz. Bu yöntemle akı değişiminin sezilmesi mümkün olmayabilir. Bu sebeple disk sürücülerinde, veri grubuna bir de saat darbesi eklenir. Bu "pencere"lerde oluşan bir akı değişimi dikkate alınır. Saat ve veri bilgisinin kodlanması için temel iki yöntem vardır ;

3.1.1 - FM (Frekans Modülasyonu) Yöntemi

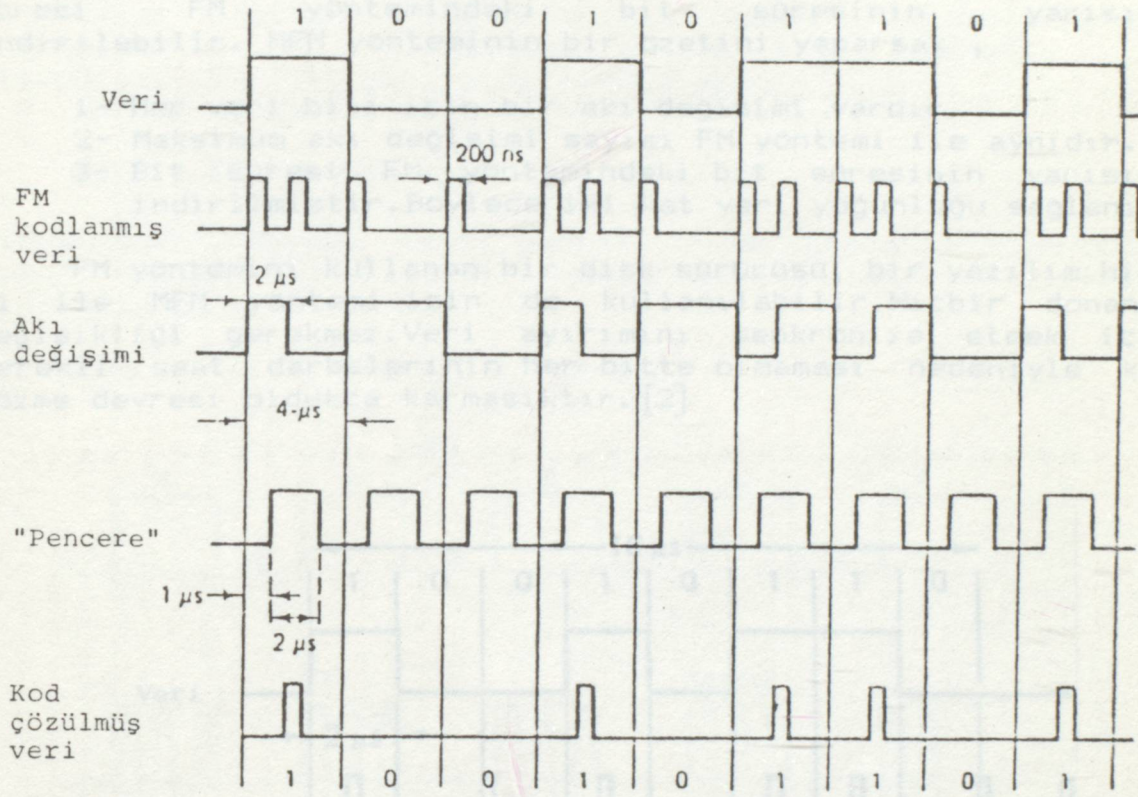
Bu yöntemde yüklenen her her bir veri biti için bir saat darbesi vardır. Şekil-3.1 de gösterildiği gibi her bit bir saat darbesi ile başlamak ve eğer lojik "1" yüklenecekse bit süresinin (4 μ s.) ortasında bir darbe yer almalıdır. Darbenin olmaması lojik "0" değerini gösterir.

Lojik "1" ve lojik "0" değerlerinin farklı frekanslarda kodlanması nedeniyle bu teknik FM (frekans modülasyonu) yöntemi olarak isimlendirilir. Yöntemin önemi, farklı disk sürücülerinde kullanılması durumunda ortaya çıkar. Motor hızlarındaki küçük farklılıklar bu yöntemle önemini yitirir. Çünkü saat işareti veri grubundan elde edilebilir.

Kod çözme işlemi oldukça basittir. Prensipte olarak 1 μ s. lik saat darbesinden sonra 2 μ s. lik bir tek-atım tetiklenir. Bu "pencere" ile bir flip-flop'a izin verilir. 2 μ s.

lik pencere süresi ile olası bit kaymaları karşılanır.

Tek-atım kullanılmasının emniyetsiz olması nedeniyle birçok FM veri ayırıcıları senkron sayıcı veya faz kilitli çevrim (PLL) teknikleri kullanmaktadırlar.



Sekil-3.1 FM kodlama tekniği

3.1.2 - MFM (Düzenlenmiş-Modified-FM) Yöntemi

FM yönteminde bir bit süresi boyunca saat darbesi ve veri biti için olmak üzere iki akı değişimi oluşabilir. Saat darbesinin kullanışlı veri bilgisi taşımaması nedeniyle akı değişiminin yarısı israf edilmektedir. MFM yöntemi ile her veri biti için bir akı değişimi ile tam etkinlik sağlanır. Böylece yükleme süresi de yarıya indirilmiş olur.

MFM yöntemi için temel kurallar ;

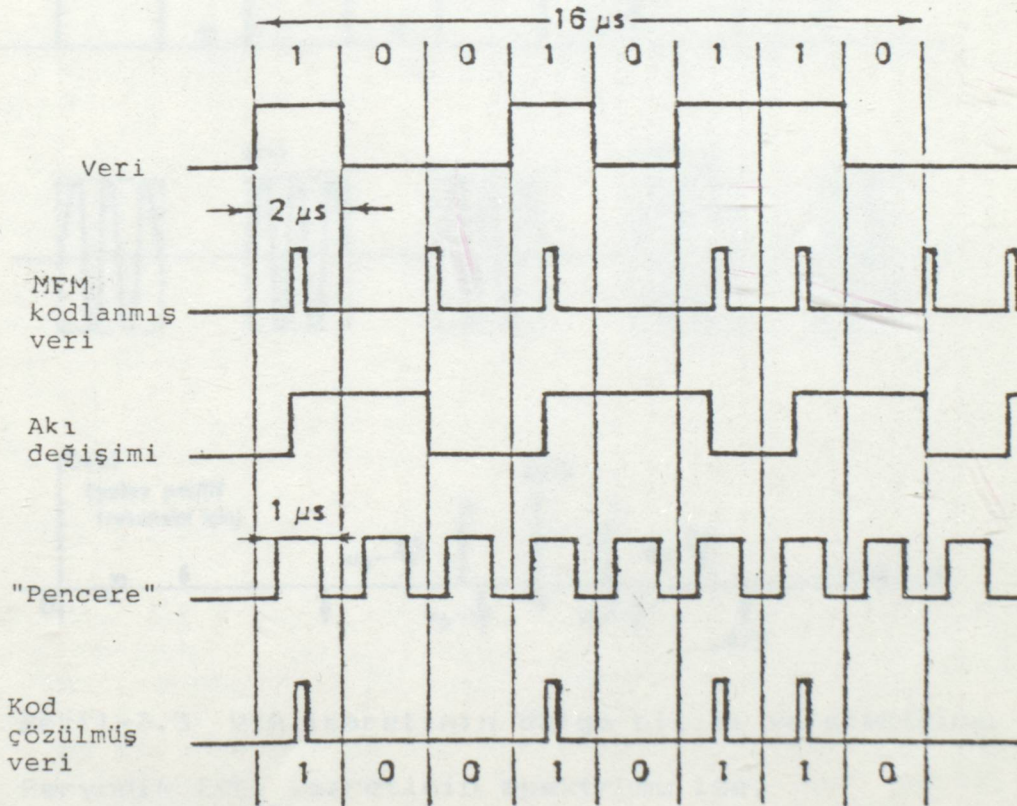
- 1- FM yönteminde olduğu gibi lojik "1" değeri bit süresinin ortasında bir darbe olarak yazılır.
- 2- Bir önceki bitin lojik "1" olması durumunda lojik "0" değeri için bir darbe gönderilmez. Diğer tüm durumlarda

lojik "0" degeri, bit süresinin başlangıcında bir darbe olarak yazılır.

Şekil-3.2 de gösterildiği gibi MFM yöntemi özel saat darbeleri içermez. Gerekli akı değişimi artırılmaksızın, bit süresi FM yöntemindeki bit süresinin yarısına indirilebilir. MFM yönteminin bir özetini yaparsak,

- 1- Her veri biti için bir akı değişimi vardır.
- 2- Maksimum akı değişimi sayısı FM yöntemi ile aynıdır.
- 3- Bit süresi FM yöntemindeki bit süresinin yarısına indirilmiştir. Böylece iki kat veri yoğunluğu sağlanır.

FM yöntemini kullanan bir disk sürücüsü, bir yazılım hilesi ile MFM yöntemi için de kullanılabilir. Hiçbir donanım değişikliği gerekmez. Veri ayırımını senkronize etmek için gerekli saat darbelerinin her bitte olmaması nedeniyle kod çözme devresi oldukça karmaşıktır. [2].



Şekil-3.2 MFM kodlama tekniği

3.2 - Miknatıslı Serit Birimlerine Kayıt Yöntemleri

3.2.1 - Var-Yok Anahtarlama Yöntemi (Amplitude Shift Keying-ASK-)

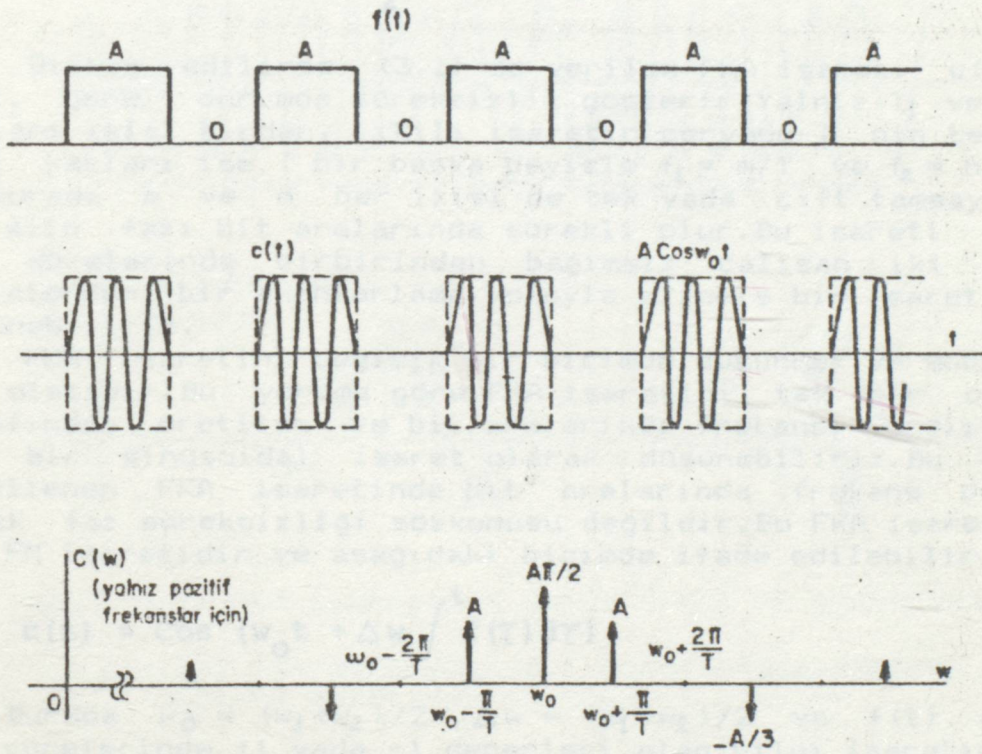
İkili bilgi işaretlerinin genlik modülasyonuna uygulanmasıdır. 0-1 dizisinden oluşan $f(t)$ işareti, bir sinüsoidal işaretin genliğini modüle eder.

Module edilmiş işaret,

$$c(t) = f(t) * \cos \omega_0 t$$

biçimindedir. Burada $f(t)$, Şekil-3.3 de gösterilen, periyodik kabul edilen 0-1 dizisidir. Module edilmiş işaretin frekans spektrumu şu biçimdedir ;

$$C(\omega) = 1/2 * [F(\omega + \omega_0) + F(\omega - \omega_0)]$$



Şekil-3.3 VYA işaretinin dalga biçimi ve spektrumu

Periyodik $f(t)$ işaretinin spektrumu ise,

$$F(\omega) = A * \pi \sum_{n=-\infty}^{\infty} \text{Sa}(n\pi/2) * \delta(\omega - n\pi/T)$$

olup, 0-1 dizisinde bir bit süresi T dir. Module edilmiş işaretin dalga şekli ve spektrumu Şekil-3.3 de verilmiştir.VYA da genliğin, ikili dizinin "0" yada "1" olmasına bağlı olarak 0 ve A arasında kaydırılması nedeniyle bu anahtarlama türü genlik kaydırmalı anahtarlama (GKA) olarak da adlandırılır.

3.2.2 - Frekans Kaydırmalı Anahtarlama Yöntemi (Frequency Shift Keying-FSK-)

İkili bilgi işaretlerinin frekans modülasyonuna uygunlanmasıdır.0-1 dizisinden oluşan f(t) işareti, bir sinüsoidal işaretin frekansını modüle eder.Bit süresi T olmak üzere k. bit döneminde FKA işareti şu biçimdedir.

$$c(t) = \begin{cases} A * \cos w_1 t & 1 \text{ için} \\ A * \cos w_2 t & 0 \text{ için} \end{cases} \quad k*T \leq t < (k+1)*T \quad (3.1)$$

Dikkat edilirse (3.1) de verilen FKA işareti c(t) nin fazı, genel durumda süreksizlik gösterir.Yalnız T₁ ve T₂ periyotları ikisi birden, ikili işaretin periyodu T nin tek yada çift katları ise (bir başka deyişle f₁ = m/T ve f₂ = n/T ise ve burada m ve n her ikisi de tek yada çift tamsayı ise) işaretin fazı bit aralarında sürekli olur.Bu işareti değişik bit sürelerinde birbirinden bağımsız çalışan iki değişik osilatörden, bir anahtarlama yoluyla alınmış bir işaret olarak düşünebiliriz.

FKA işaretini değişik bir biçimde düşünmek ve modellemek de olasıdır.Bu yoruma göre FKA işaretini tek bir osilatör tarafından üretilen ve bit aralarında frekansı değiştirilen tek bir sinüsoidal işaret olarak düşünebiliriz.Bu biçimde modellenen FKA işaretinde bit aralarında frekans değişir, ancak faz süreksizliği söz konusu değildir.Bu FKA işareti özel bir FM işaretidir ve aşağıdaki biçimde ifade edilebilir ;

$$c(t) = \cos (w_0 t + \Delta w \int_0^t f(\tau) d\tau) \quad (3.2)$$

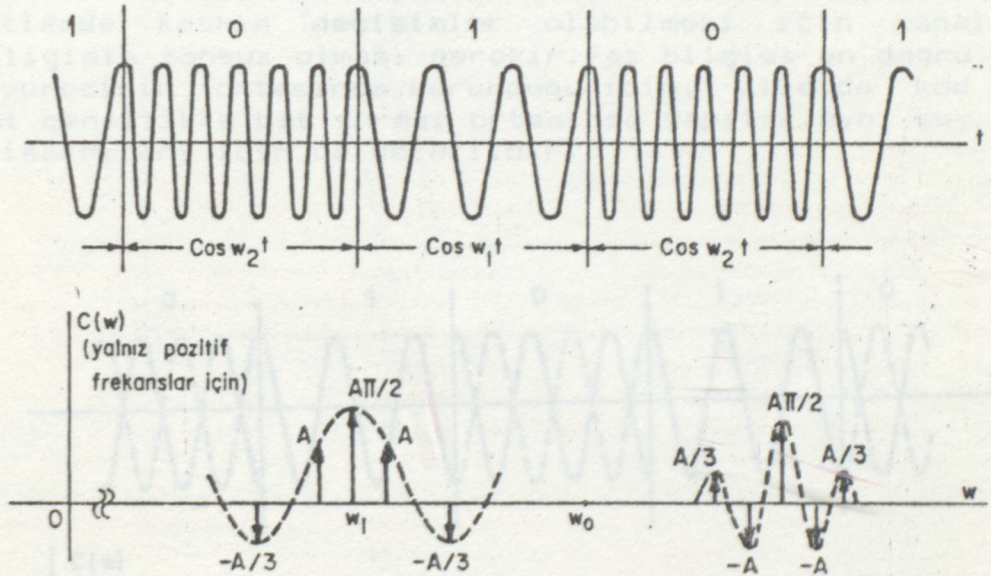
Burada w₀ = (w₁+w₂)/2 , Δw = (w₁-w₂)/2 ve f(t) değişik bit sürelerinde +1 yada -1 değerleri alan bilgi işaretidir. Bu tür FKA işareti, (3.2) deki gibi bir FM işareti olarak yorumlandığında w₀ taşıyıcı frekansı ve Δw frekans sapmasıdır. Bu iki FKA işaretinin birbirinin aynı olmadığı açıktır. Ancak birinci tür FKA da bit aralarında faz sürekli ise ikinci tür FKA ile aynı olur.

(3.1) de verilen birinci tür FKA işareti biri w₁ , diğeri w₂ frekansında iki VYA işaretinin toplamı olarak düşünülebilir.Bu nedenle bu işaretin spektrumu da biri w₁ de, diğeri w₂

de iki VYA işareti spektrumunun toplamıdır. Yalnız VYA işaretlerinden birisi zaman bölgesinde T kadar kaydırılmıştır ;

$$C(\omega) = \frac{A \cdot \pi}{2} \sum_{n=-\infty}^{\infty} \text{Sa}\left(\frac{n \cdot \pi}{2}\right) \left\{ \left[\delta\left(\omega + \omega_1 - \frac{n \cdot \pi}{T}\right) + \delta\left(\omega - \omega_1 - \frac{n \cdot \pi}{T}\right) \right] + (-1)^n \cdot \text{Sa}\left(\frac{n \cdot \pi}{2}\right) \left[\delta\left(\omega + \omega_2 - \frac{n \cdot \pi}{T}\right) + \delta\left(\omega - \omega_2 - \frac{n \cdot \pi}{T}\right) \right] \right\}$$

Bit aralarında faz süreksizliği olmayan bir FKA işareti dalga şekli ve spektrumu Şekil-3.4 de verilmiştir. İkili işaretin band genişliği $B = k/T$ varsayılırsa FKA işaretin band genişliği $2\Delta f + 2B$ olur.



Şekil-3.4 FKA işaretinin dalga şekli ve spektrumu

3.2.3 - Faz (Evre) Kaydırmalı Anahtarlama Yöntemi (Phase Shift Keying-PSK-)

İkili bilgi işaretlerinin faz modülasyonuna uygulanmasıdır. Taşıyıcı fazı ikili işarete bağlı olarak 180° kaydırılır. "1" seviyesinde, modüle eden işaret $\text{Cos } \omega_0 t$, "0" seviyesinde ise $\text{Cos } (\omega_0 t + \pi)$ dir. FKA işaretinin zaman bölgesi gösterimi şöyledir ;

$$c(t) = \text{Cos } (\omega_0 t + (\pi \text{ yada } 0)) = \mp \text{Cos } \omega_0 t \quad k \cdot T \leq t < (k+1) \cdot T$$

EKA işareti 0-1 dizisine bağlı olarak üretilen çift

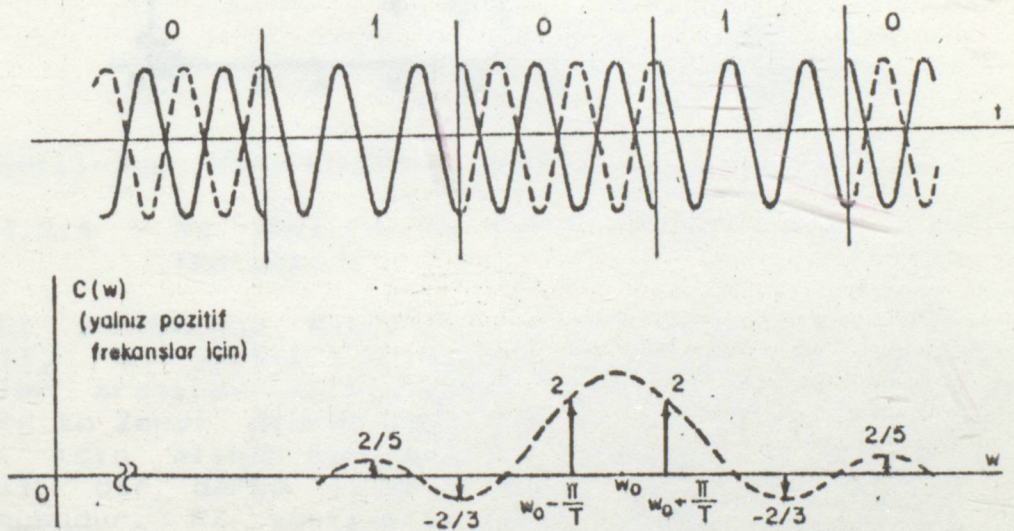
kutuplu bir dizi ($\bar{1}$ dizisi) ile taşıyıcının çarpımına eşittir. Bu nedenle EKA işareti,

$$c(t) = [2*f(t)-1] * \text{Cos } w_0 t$$

Buna göre çift kutuplu dizinin spektrumunun w_0 a taşıması ile EKA işaretinin spektrumu şu şekilde ifade edilebilir;

$$C(w) = \pi \sum_{\substack{n=-\infty \\ n \neq 0}}^{\infty} \text{Sa} \left(\frac{n*\pi}{2} \right) \left[\delta(w+w_0 - \frac{n*\pi}{T}) + \delta(w-w_0 - \frac{n*\pi}{T}) \right]$$

EKA işaretinin dalga biçimi ve spektrumu Şekil-3.5 de çizilmiştir. "0" dan "1" e veya "1" den "0" a bir geçiş olduğunda, bit süresinin başında yada sonunda oluşan süreksiz faz geçişleri iletim sırasında düzeltilir. Kanal çıkışındaki işaretlerde keskin değişimler olabilmesi için kanal band genişliğinin sonsuz olması gerekir. Faz bilgisi en doğru olarak bit süresinin ortasında korunduğu için, alıcıda kod çözme işlemi genellikle bit süresi ortasında yapılır. Aynı şey VYA ve FKA işaretleri için de geçerlidir.



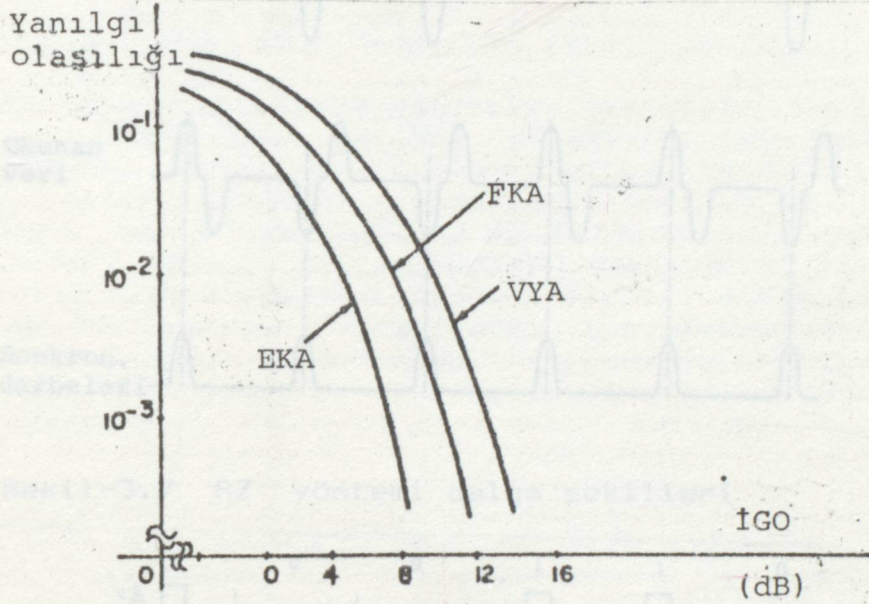
Şekil-3.5 EKA işaretinin dalga şekli ve spektrumu

VYA, FKA ve EKA yöntemlerinden hangisi tercih edilmelidir? Modüle edilmiş her işarete kanal boyunca toplamsal Gauss gürültüsünün eklendiği varsayıldığında her yöntem için bir yanığı olasılığı hesaplanabilir. A modüle eden işaretin

genliği, N gürültü gücü olmak üzere işaret-gürültü oranı,

$$1GO = A^2 / (2*N)$$

olacaktır. Şekil-3.6 da yanılgi olasılığı eğrileri 1GO na bağıli olarak çizilmiştir. Belirli bir yanılgi olasılığı için FKA yön temindeki işaret-gürültü oranı EKA yöntemine göre 3 dB. daha iyi, VYA yöntemine göre ise 6 dB. daha iyidir. [5].

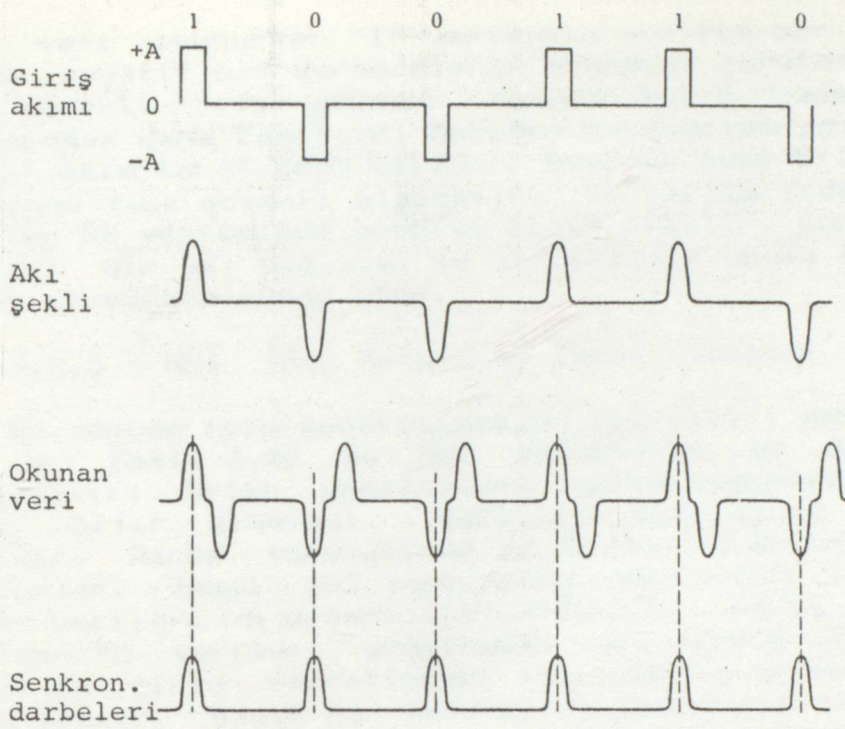


Şekil-3.6 VYA, FKA, EKA yöntemlerinin yanılgi olasılıkları

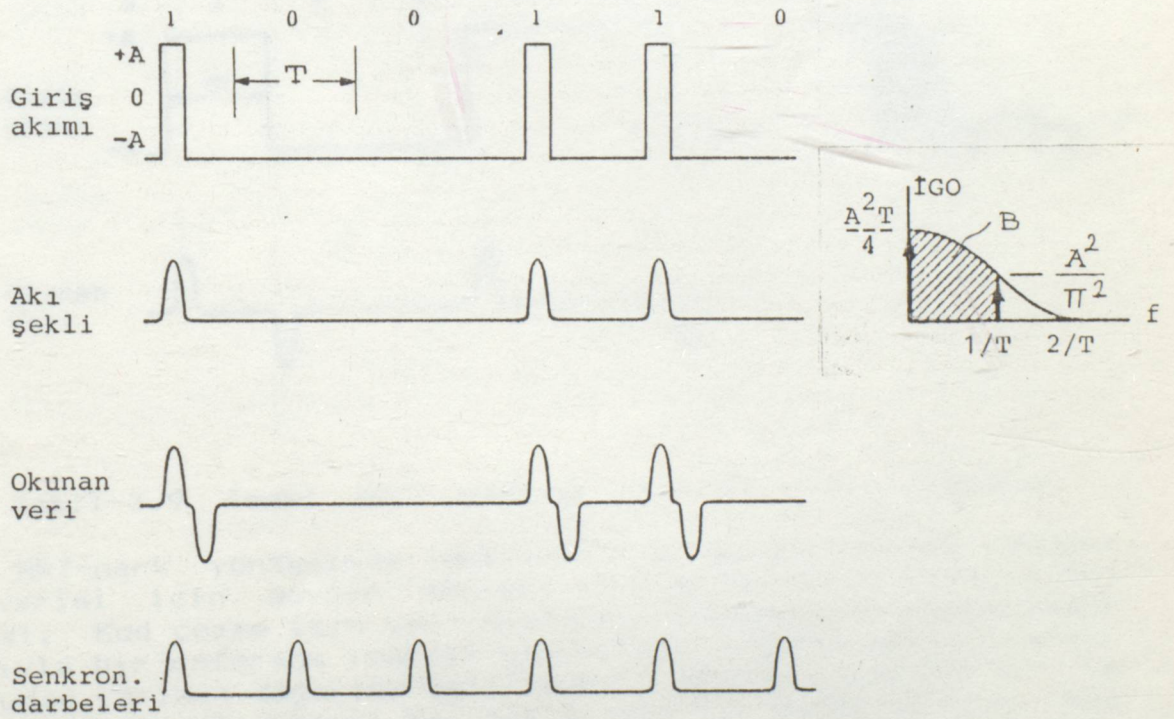
3.2.4 - RZ (Return to Zero) ve RB (Return to Bias) Yöntemleri

RZ yönteminde Şekil-3.7 de görüldüğü gibi "1" verisi pozitif, "0" verisi ise negatif bir darbe ile kaydedilir. Darbeler arasında akımın değeri sıfır olduğu için yöntem RZ (Return to Zero) olarak isimlendirilir. Okuma kafasından "1" değeri için alınan her pozitif darbeyi, hemen arkasından negatif bir darbe izler. Aynı durum "0" değeri için sözkonusudur. RZ yönteminin temel zorluğu peşpeşe iki darbe içinden "1" verisi için pozitif darbeyi, "0" verisi için negatif darbeyi sezmeğdir. Bu amaçla senkronizasyon darbeleri veriyle birlikte kaydedilir. RZ yönteminin bir olumsuz yönü de eski kayıtların üzerine yapılan veri kayıtlarında ortaya çıkar. Yeni veri yüklenmeden önce eski kayıtların silinmesi ve verinin mutlaka senkronizasyon darbeleri ile kaydedilmesi gereklidir.

RB yönteminde ise kafa akımı "1" değerleri haricinde negatiftir. Akı şeklinde her "1" verisi için bir darbe vardır.



Şekil-3.7 RZ yöntemi dalga şekilleri

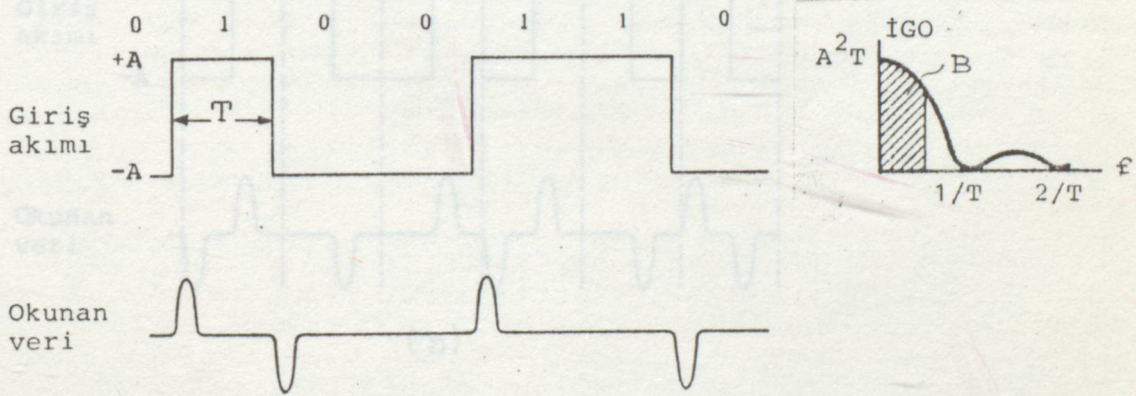


Şekil-3.8 RB yöntemi dalga şekilleri ve IGO'nun değişimi

Ancak veri okunurken "1" verisinin pozitif bir darbe ve onu izleyen negatif bir darbe olarak okunduğu görülür. "0" verisi için sürekli sıfır okunur. Yeniden kayıt yapma işlemi RZ yönteminden daha basittir. Önceden kaydedilen pozitif darbeler negatif akım tarafından silinir. Böylece yeni bilginin senkro nizasyonu daha güvenli olacaktır. "0" verilerinden oluşan uzun darbeler RB yönteminde problem oluşturabilir. Çünkü sadece "1" verileri bir akı değişimi ve dolayısıyla okuma kafasında bir işaret oluşumuna sebep olur.

3.2.5 - NRZ (Non Return to Zero) Yöntemi

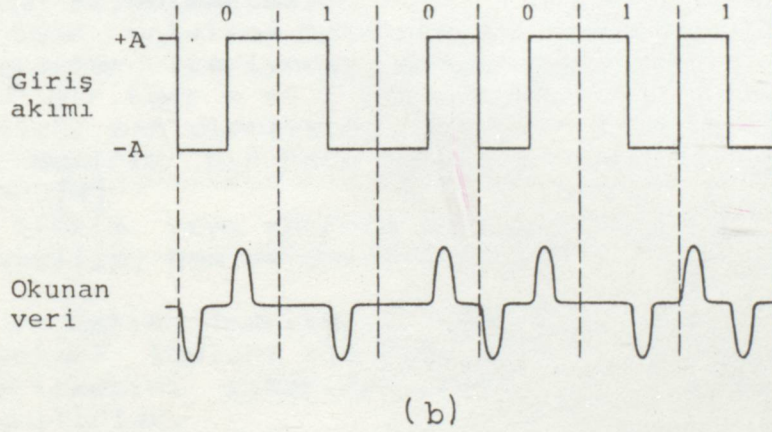
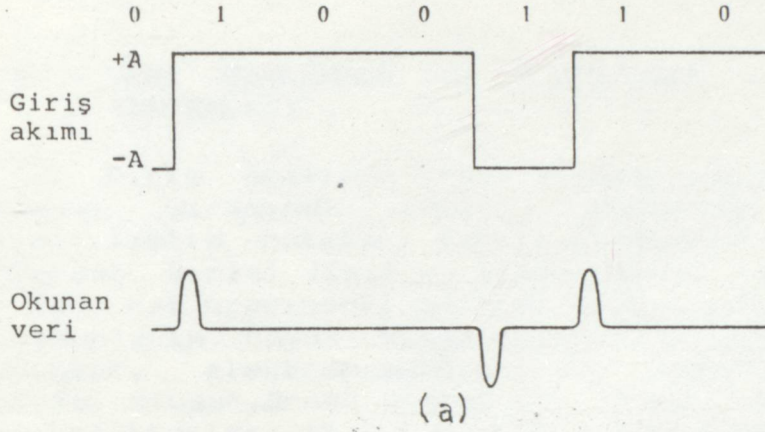
Bu yöntem için senkronizasyon işaretleri gerekmez. Şekil-3.9 ve Şekil-3.10 da NRZ yönteminin üç değişik şekli verilmiştir. Ortak özellikleri giriş işaretlerinin hiçbir zaman sıfır olmaması, farkları ise giriş işaretlerinin şeklidir. Darbe süreleri RZ ve RB yöntemlerinden daha uzun tutulurlar. Temel NRZ yönteminde (Şekil-3.9) "1" verileri +A ve "0" verileri -A değeri ile yazılırlar. Giriş işareti sadece "1" ve "0" verileri arasındaki geçişlerde değişir. Okuma sırasında giriş işaretindeki her seviye değişimi için bir darbe alınır. Basit bir kod çözücü ile giriş işareti yeniden elde edilebilir.



Şekil-3.9 Temel NRZ yöntemi ve IGO'nun değişimi

NRZ-mark yönteminde (Şekil-3.10.a) giriş işareti sadece "1" verisi için seviye değiştirir (+A dan -A ya veya -A dan +A ya). Kod çözme için veri dizisinin başlangıcını belirlemek amacıyla bir referans işareti gereklidir. Çünkü giriş işareti doğrudan verinin değerini belirlemez. Okunan işarete "1" ve "0" verileri arasındaki her geçiş için bir darbe vardır. Kod çözme işleminde kod çözücü bir ilk duruma şartlandırılmalıdır.

NRZ-biphase mark yönteminde ise (Şekil-3.10.b) , "0" verisi $-A$ ile $+A$ arasında geçişlerle belirtilir. Okunan işaretle diğer iki yönteme kıyasla daha fazla darbe vardır ve kod çözme işlemi daha zordur. Ancak uzun sabit veri dizilerinin oluşturduğu problemler bu yöntemde görülmez.



Şekil-3.10 NRZ-mark ve NRZ-biphase mark yöntemleri

4 - VERİ SAKLAMADA VE İLETİMİNDE KULLANILAN KODLAMA YÖNTEMLERİ

İki birim arasında veri iletimi yada bir ortama veri yüklenmesi sırasında verinin bütünlüğü ve doğruluğu önemlidir. İletim hataları farklı etkenlere bağlıdır. En önemli etkenlerden birisi (ısı ve elektriksel) gürültü kaynaklarıdır. Bir hat üzerindeki gürültü darbelerinin tipik süresi 10 ms.dir. Bunların büyük kısmı güç hatlarındaki ani artışlar, yıldırımlar, elektromagnetik ve elektromekanik araçlar nedeniyle oluşur. Örnek olarak bir iletim hattında 10 bitten oluşan karakterler, 15 karakter/sn. hızla gönderildiğinde 6850 karakterde 1 karakter geçersiz okunmakta, 1467 de 1 karakter de tamamen kaybolmaktadır.

Eğer ortalama hata miktarı karakter başına e ise ve blok n karakter içeriyorsa bloğun başarıyla iletilme olasılığı $(1-e)^n$ dir. Eğer $e \ll 1$ ise yaklaşık olarak bir bloktaki hata olasılığı $e*n$ olacaktır. 8 bitlik n tane veri içeren bir blok için ampirik bir hata olasılığı formülü $10^{-4}*n$ olarak bulunmuştur. [4]

İletim veya yükleme sırasındaki hatalar sezilebilir veya hem sezilip, hem de düzeltilebilir ;

1- Eşlik (parity), checksum, CRC (Cyclic Redundancy Character) kodları ile hata sezilir ve alıcı verinin tekrar gönderilmesini ister. Genellikle çift yönlü haberleşmelerde kullanılırlar.

2- Hamming, düzenlenmiş (modified) Hamming, syndrome yöntemleri ile hata sezilir ve düzeltilir. Veri iletiminin tekrar istenemeyeceği tek yönlü haberleşmelerde tercih edilirler.

4.1 - Hata Sezen Kodlar

4.1.1 - Eşlik (Parity) Kontrolü

Tek bitlik hataların sezilmesi için en basit yöntemdir. İletilecek her sözcüğe bir eşlik biti eklenir. Sözcükte "1" olan bitlerin sayısının tek veya çift olması durumuna göre eşlik biti belirlenir.

Örnek ; Aşağıdaki veriler çift eşlik biti ile kodlanmıştır ve en anlamlı bit eşlik bitidir. Bu verilerin hangisinde hata vardır?

D1_h = 11010001
36_h = 00110110
E5_h = 11100101

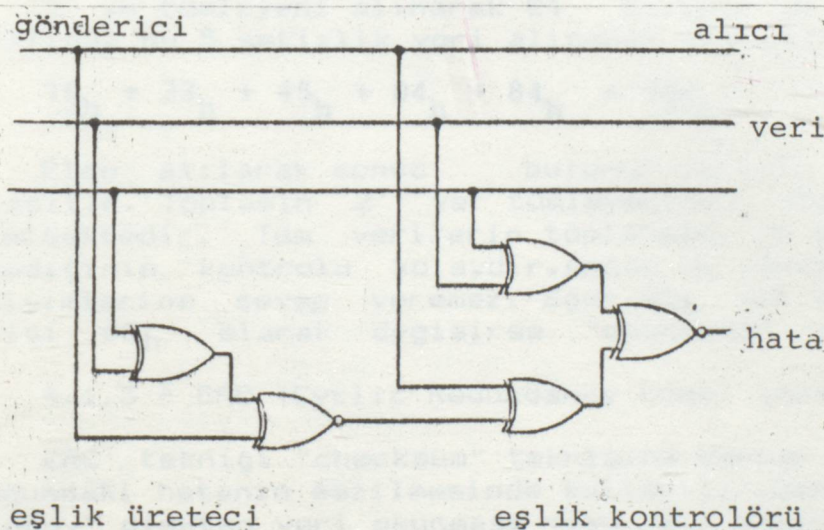
Sadece E5_h verisinde tek sayıda "1" bulunduğu görülmektedir. Bu sebeple E5_h verisi yanlış alınmıştır.

Tüm UART birimleri otomatik olarak eşlik biti oluşturacak şekilde tasarlanmıştır. Alınan veride bir hata sezildiğinde PE (parity error) bayrağı set edilir. Eger arzu edilirse alıcı, verinin yeniden gönderilmesini isteyebilir.

Bu yöntemle çoklu bit hatalarının sezilmesi mümkün değildir. Yukarıdaki örnekte D1_h verisi D2_h olarak alınmışsa iki bitte hata oluşacak fakat eşlik biti yine doğru olacaktır.

Buna rağmen seri haberleşme kanallarında eşlik bitinin kullanımı sınırlanmaz. Aksine hemen hemen tüm ana bilgisayar üreticileri merkezi işlem birimi ile bellek arasındaki veri transferlerinde eşlik bitini kullanırlar.

Eşlik üretici ve kontrolörü devreleri dar-VEYA kapıları ile oluşturulabilir. Şekil-4.1 deki örnekte 3 bitlik mesaj tek eşlik biti ile iletilir. Eger alınan veride çift eşlik oluşursa eşlik kontrolörünün çıkışı "1" olacaktır.



Şekil-4.1 Tek eşlik biti ile hata sezimi

4.1.2 - Checksum Yöntemi

Eşlik kontrolünün temel dezavantajı her veri sözcüğü için ek bit gerektirmesidir. Bu ise iletim hızını düşürür ve veri saklandığı zaman ek bellek gözleri gerektirir. Bu sebeplerle "checksum" veri bloklarının transferinde kullanılan bir kontrol yöntemi olmuştur.

"Checksum" bir sekizlik veridir ve veri bloğunda son olarak gönderilir. Alıcı, alınan veri bloğunu ("checksum" da dahil olmak üzere) toplar. "Checksum" sekizlisi ile karşılaştırır. Eğer sonuç farklı ise veri tekrar gönderilir.

Bir avantajı da eşlik kontrolüne nazaran daha az masraflı olmasıdır. Örnek olarak 256 sekizlik blokta, 1 sekizli veya % 0.4 oranında bellek hata kontrolüne ayrılır. Eğer eşlik kontrolü kullanılsaydı 256 bit veya % 12.5 oranında bellek gerekliydi. Ayrıca veri bloğunda çeşitli etkenlerle oluşan herhangi bir hata "checksum" ın değişmesine neden olur. Oysa eşlik kontrolü sadece bir sekizlik veri içindir.

"Checksum" nasıl hesaplanır? Tüm veri bloğunun toplamı bir sekizli olmalıdır. Bunu yapmak için tüm veriler toplanır ve oluşan elde ihmal edilir. Toplamın 2'ye tümleyeni alınarak son veri olarak gönderilir.

Örnek ; $10_h, 23_h, 45_h$ ve 04_h verilerinin "checksum" verisini hesaplayalım ,

$$10_h + 23_h + 45_h + 04_h = 7C_h$$

2'ye tümleyeni alınarak 84 bulunur ve gönderilir. Alıcı tarafında bu 5 sekizlik veri alınarak toplanır.

$$10_h + 23_h + 45_h + 04_h + 84_h = 100_h$$

Elde atılarak sonuç bulunur. Verinin doğru alındığı anlaşılır. Toplamın 2'ye tümleyeninin alınmasının sebebi görülmektedir. Tüm verilerin toplanması ve sonucun 00 olup olmadığının kontrolü kolaydır. Ancak bu yöntem de çoklu bit değişimlerine cevap veremez. Eğer 45_h verisi 44_h ve 04_h verisi 05_h olarak değişirse "checksum" yine geçerlidir.

4.1.3 - CRC (Cyclic Redundancy Code) Yöntemi

CRC tekniği "checksum" tekniğine benzer olarak bir veri bloğundaki hatanın sezilmesinde kullanılır. Genel kullanımı bir "floppy" diskten veri okunması veya veri yazılması ile senkron veri haberleşmesidir. Hata sezmede etkili bir yöntemdir. Ancak veri miktarının diğer kodlama türlerinden daha fazla olması dezavantajıdır.

Veri bloğu belli uzunlukta ikili polinomdur ve $P(x)$ ile adlandırılır. Örneğin, 10101 verisi polinom olarak $x^4 + x^2 + 1$ şeklinde gösterilir. Polinomun derecesi sözcükteki bit

sayısından bir eksiktir. Veri iletiminin sonunda veri sabit bir ikili polinoma bölünür, bu polinoma generatör polinomu $G(x)$ denir. Kalan ikili polinom da, blok kontrol polinomu $R(x)$ olarak isimlendirilir ve verinin sonuna eklenir. Alıcı uçta, alınan veri bloğu aynı generatör polinomuna bölünür. Eğer kalan olursa hata oluşmuştur. Verinin tekrar gönderilmesi istenir.

Bu yöntemde kullanılan ikili polinomlar aşağıdaki gibi tanımlanabilir,

- $M(x)$; m bitlik asıl veri polinomu
- $G(x)$; generatör polinomu
- $T(x)$; t bitlik iletilen veri polinomu
- $E(x)$; hata polinomu

Polinomların toplanması ve çıkarılması, modulo-2 aritmetiği ve dar-VEYA işlemleri ile sağlanır. Örnek olarak toplama işlemi,

$$\begin{array}{r} x^4 \quad +x^2+x \\ x^3+x^2+x \\ \hline x^4+x^3 \end{array}$$

Çarpma ve x^5+x^3+x polinomunun x^2+x+1 ile bölümü ise şekilde yapılır.

$$\begin{array}{r} x^4+x^2+x \\ x^3+x^2+x \\ \hline x^7 \quad +x^5+x^4 \\ x^6 \quad +x^4+x^3 \\ \hline x^5 \quad +x^3+x^2 \\ \hline x^7+x^6 \quad +x^2 \end{array}$$

ve

$$\begin{array}{r} x^3+x^2+x \\ x^2+x+1 \overline{) x^5+x^3+x} \\ \underline{x^5+x^4+x^3} \\ x^4 \\ \underline{x^4+x^3+x^2} \\ x^3+x^2+x \\ \underline{x^3+x^2+x} \\ 0 \end{array}$$

şeklinde yapılır.

Kodlanan veriyi elde etmek için ilk olarak asıl veri c bit sola kaydırılır. c ise eklenecek kontrol bitlerinin

sayısıdır ve generatör polinomunun derecesine eşittir. Kaydırılan veri $M(x) \cdot x^c$ ile gösterilsin. Bu polinom generatör polinomuna bölünür. Bölme işleminde artan polinomla kaydırılan veri toplanarak iletilen polinom elde edilir. Bu polinom generatör polinomuna tam olarak bölünebilir. Böylece ,

$$\frac{M(x) \cdot x^c}{G(x)} = Q(x) + \frac{R(x)}{G(x)}$$

Burada $Q(x)$ bölüm ve $R(x)$ artan polinomlar olup, derecesi generatör polinomundan bir derece düşüktür. $G(x)$ ile çarparak,

$$M(x) \cdot x^c = Q(x) \cdot G(x) + R(x)$$

ve modulo-2 ye göre toplama ile çıkarma eşdeğer olduğundan,

$$T(x) = M(x) \cdot x^c + R(x) = Q(x) \cdot G(x)$$

bulunur. Buradan $T(x)$ polinomunun $G(x)$ ile tam olarak bölünebileceği ve sonucun $Q(x)$ olacağı görülür.

Örnek olarak,

1- Veri 101101 , dolayısıyla $M(x) = x^5 + x^3 + x^2 + 1$ ve generatör polinomu $G(x) = x^4 + x^3 + 1$ olsun. $G(x)$ 5 bit içerir ve derecesi 4 dür. Dolayısıyla $M(x)$ 4 defa kaydırılır.

$$M(x) \cdot x^4 = (x^5 + x^3 + x^2 + 1) \cdot (x^4) = x^9 + x^7 + x^6 + x^4$$

2- Bu çarpımın generatör polinomuna bölümü kalanı verir.

$$\begin{array}{r}
 + 1 \quad + x^2 \\
 + 1 \quad + x^2 \\
 \hline
 x^9 + x^7 + x^6 + x^4 \\
 x^9 + x^8 + x^5 \\
 \hline
 x^8 + x^7 + x^6 + x^5 + x^4 \\
 x^8 + x^7 + x^4 \\
 \hline
 x^6 + x^5 \\
 x^6 + x^5 + x^2 \\
 \hline
 + x^2 \text{ kalan}
 \end{array}$$

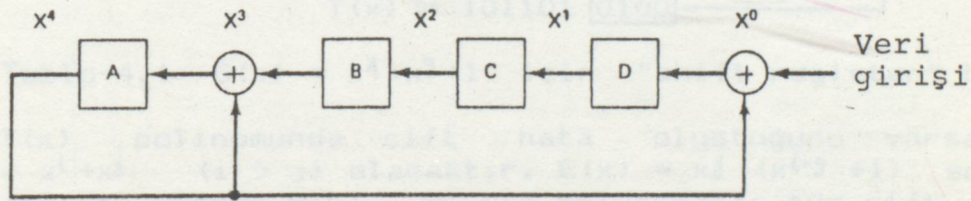
Kalan $R(x) = x^2$, $M(x) \cdot x^4$ polinomuna eklenerek iletilen polinom bulunur.

Böylece iletilen kod sözcüğü , 1011010100 olacaktır. Alıcıda bu verinin aynı polinoma bölünmesi sonucunda artan oluşmazsa veri doğru alınmıştır. Kontrol edilirse,

$$\begin{array}{r} x^4+x^3 \quad +1 \\ \hline x^9 \quad +x^7+x^6 \quad +x^4 \quad +x^2 \\ x^9+x^8 \quad \quad \quad +x^5 \\ \hline x^8+x^7 \quad \quad \quad +x^4 \\ \hline x^6+x^5 \quad \quad \quad +x^2 \\ \hline \end{array}$$

0 artan yok.

Şekil-4.2 de $G(x) = x^4+x^3+1$ için iletilecek veriyi bulan bir kodlayıcı blok şeması verilmiştir. Tablo-4.1 incelendiğinde şema kolayca anlaşılabilir. Aynı devre alınan verinin kod çözümünde de kullanılabilir. Saklayıcı kalan veriyi içerir. Eger hata varsa kalan sıfırdan farklı olur. Kalan veride her "1" değeri bir bitlik hatayı gösterir.



□ "Shift register"

⊕ Dar-VEYA

Şekil-4.2 $G(x) = x^4+x^3+1$ için $T(x)$ polinomunun bulunması

Gönderilen $T(x)$ polinomu yerine $T(x)+E(x)$ polinomunun alındığını farzedelim. $E(x)$ deki her bir bit, yanlış alınan bir bitlik veriye karşı düşer. Alıcı tarafta bu veri $G(x)$ e

bölündüğünde $G(x)$ in içerdiği terimlere karşı düşen hatalar farkedilemez.

$E(x) = x^i$ olduğunu farzedelim. "i" hata oluşan biti gösterir. Eğer $G(x)$ iki veya daha fazla terime sahipse $E(x)$ polinomunu bölemeyecek, böylece tüm hatalar sezilebilecektir. "r" tane kontrol bitine sahip bir polinomla en fazla "r" tane hata sezilebilir.

"Shift" No.	A	B	C	D	
	0	0	0	0	ilk durum
1	0	0	0	1	} veri girişi
2	0	0	1	0	
3	0	1	0	1	
4	1	0	1	1	
5	1	1	1	1	} 4 tane 0 girişi (c)
6	0	1	1	0	
7	1	1	0	0	
8	0	0	0	1	
9	0	0	1	0	
10	0	1	0	0	

kaydırılan veriye kalan eklenir

$M(x) = 101101$
 $T(x) = 101101$ 0100

Tablo-4.1 $G(x) = x^4+x^3+1$ için "shift register" bitleri

$T(x)$ polinomunda çift hata olduğunu varsayalım, $E(x) = x^i + x^j$ ($i > j$) olacaktır. $E(x) = x^j (x^{i-j} + 1)$ şeklinde yazılabilir. Eğer $G(x)$, x ile bölünemezse tüm çift sayıda hatalar sezilebilir. Ancak $G(x)$ in $x + 1$ polinomunu bölememe şartı vardır. Bunun için de k değeri $(1-j)$ değerinden (maksimum mesaj uzunluğundan) büyük olmalıdır. Örnek olarak $x^{15} + x^{14} + 1$ polinomu, $x^k + 1$ polinomunu ancak $k \geq 32768$ değeri için bölebilir.

Eğer hatalı bitler tek sayıda ise $E(x)$ tek sayıda terim içerir. ($x^5 + x^2 + 1$ gibi) Modulo-2 aritmetiginde $(x+1)$ in kuvveti olan tek sayıda terime sahip hiç bir polinom yoktur. $G(x)$ in $(x+1)$ in kuvveti yapılmasıyla tek sayıdaki hataların tümü yakalanabilir. Geçersiz bir verinin doğru kabul edilme olasılığı $1/2^{r-1}$ dir.

Uluslararası standart olarak üç polinom kullanılır ;

$$\text{CRC-12} = x^{12} + x^{11} + x^2 + x + 1$$

$$\text{CRC-16} = x^{16} + x^{15} + x^2 + 1$$

$$\text{CRC-CCIT} = x^{16} + x^{12} + x^5 + 1$$

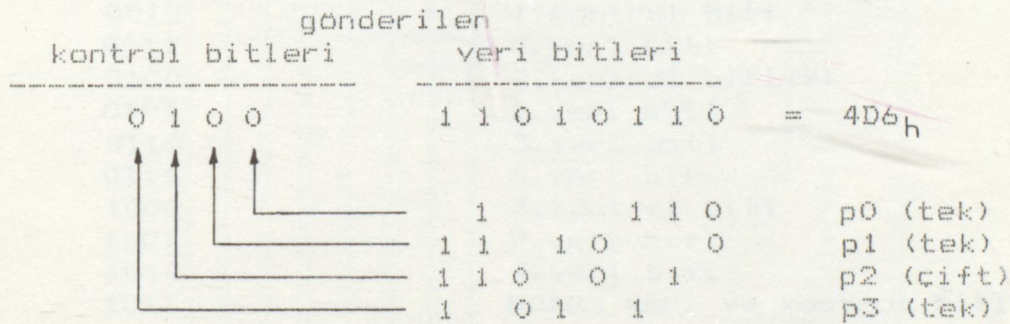
Bu üç polinom da $(x + 1)$ in kuvvetidir. CRC-12, 6 bitlik karakter uzunluğu için kullanılır. Diğerleri 8 bitlik karakterler içindir. CRC-16 ve CRC-CCITT 18 bit ve daha uzun iletim hatalarını % 99.998 oranında, 17 bitlik hataları % 99.997 oranında, 16 bitlik hataların ise tümünü yakalar. [4]

4.2 - Hata Sezen ve Düzelten Kodlar

4.2.1 - Hamming Kodu

1950 de matematikçi Richard Hamming yalnızca hata sezmekle kalmayıp, aynı zamanda düzelten bir kodlama türü buldu. Hamming Kodu olarak isimlendirilen bu kodlama türü bugün kullanılan bütün türlerin temelini teşkil eder.

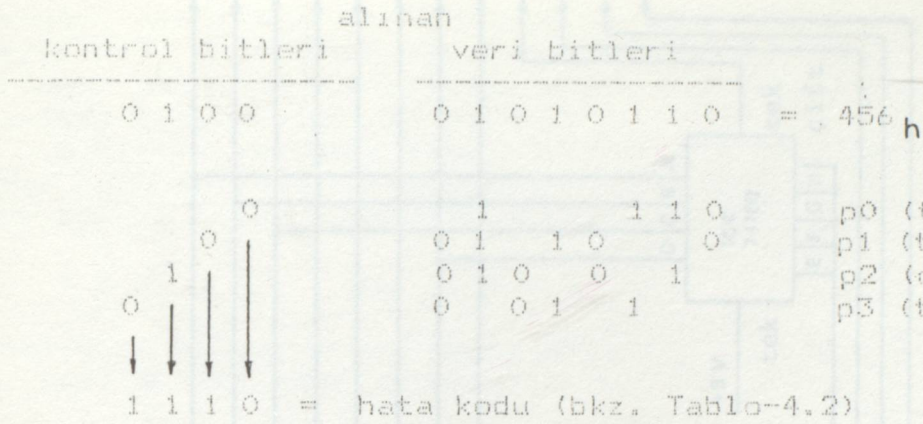
Hata düzeltme görevini her bir veri sözcüğü ile beraber gönderilen eşlik bitleri yapar. Aşağıda bu teknik 8 bitlik sözcük için gösterilmiştir. p0, p1, p2, p3 ile gösterilen 4 tane eşlik biti kullanılmıştır.



Her eşlik biti farklı veri bitlerinin oluşturduğu bir kümeyi test eder. Ayrıca her veri biti en az iki eşlik bitine ait küme içinde olmalıdır. Bu örnekte D6_h verisi, eşlik bitleri ile beraber 4D6_h olarak 12 bit halinde gönderilmektedir.

Aşağıda ise alınan verinin nasıl test edildiği gösterilmektedir. 4 eşlik biti yeniden üretilir, fakat bu kez gönderici tarafın ürettiği eşlik bitleri de göz önüne alınır. Örnek olarak en az anlamlı eşlik biti verinin 0,1,2,8 numaralı bitlerinin kontrolü ile hesaplanır.

12 bit kodlanmış veri çıkışı
0 0 0 0 1 1 0 1 0 1 1 0



Verinin 7. bitinin yanlış alındığını görülmektedir. Alıcı tarafta üretilen 4 eşlik biti bir hata kodu gösterir. Tablo-4.2 de oluşabilecek 16 hata için hata kodları verilmiştir. 0000 hata kodu verinin hatasız alındığını gösterir. Bu örnekte 1110 hata kodu 7. bitin düzeltilmesi gerektiğini gösterir.

hata kodu	hatalı bit
0000	veri doğru alındı
0001	0.kontrol biti
0010	1.kontrol biti
0011	0.veri biti
0100	2.kontrol biti(*)
0101	1.veri biti
0110	3.veri biti
0111	6.veri biti
1000	3.kontrol biti
1001	2.veri biti
1010	4.veri biti
1011	bütün veri ve kontrol bitleri "0"
1100	5.veri biti
1101	birden fazla bitte hata var
1110	7.veri biti
1111	birden fazla bitte hata var

(*) ; bütün veri ve kontrol bitleri "1"

Tablo-4.2 8 bit Hamming Kod'u için hata kodları

Bu kodlama tarzı çoklu iletim hatalarını sezebilmesine rağmen düzeltemez. Ancak tek bitlik hataların düzeltilmesi mümkündür. Şekil-4.3 de 4 tane 74180 (parity generator/checker) tamdevresi ile gerçekleştirilen bir Hamming kodlama

devresi verilmiştir. Bu devre ile 70 ns.de tüm eşlik bitlerinin üretilmesi mümkündür. Mikroişlemci ile sistem belleği arasında kullanılmak üzere tasarlanmış hata sezme ve düzeltme tümdevreleri de vardır.

m bitlik veriye c tane kontrol biti eklendiğini ve t tane bitin gönderildiğini düşünelim. ($t = m + c$) c tane kontrol biti ile 2 tane hata sezilebilir. Böylece ,

$$t \leq 2^c - 1 \quad \text{ve} \quad t + 1 \leq 2^c$$

olmalıdır. Kolaylık sağlanması amacıyla $t + 1 = 2^c$ kabul edilirse ve veri m bit içeriyorsa ,

$$\begin{aligned} t &= m + c \\ m + c + 1 &= 2^c \\ m &= 2^c - c - 1 \end{aligned}$$

olacaktır. Buradan m bitlik verinin kodlanması için gerekli bit sayısı bulunabilir.

4.2.2 - Düzenlenmiş (Modified) Hamming Kodu

Eğer Hamming Kodu, hata kodu hatalı bitin numarasını verecek şekilde düzenlenirse "Düzenlenmiş Hamming Kodu" elde edilir. Bu durumda hata kodu, kodlanmış sözcüğün en az anlamlı bitinden başlamak üzere hatalı bitin numarasını verir. Örnek olmak üzere verinin 4 bit uzunluğunda olduğunu düşünelim. $m = 4$ ve $c = 3$ olmak üzere kodlanmış sözcük 7 bit uzunluğunda olacaktır.

m3	m2	m1	c2	m0	c1	c0	bit gösterimi
7	6	5	4	3	2	1	bit numarası

Bu örnekte olduğu gibi daha uzun veriler için de kontrol bitleri 2^0 , 2^1 , 2^2 , -----, 2^n bit numaralarında yer almalıdırlar. Veri bitleri kontrol bitleri ile beraber çift eşlik (tek eşlik kontrolü de kullanılabilir) kontrolüne sokulduğunda ;

$$\begin{aligned} c0 \oplus m0 \oplus m1 \oplus m3 &= 0 \\ c1 \oplus m0 \oplus m2 \oplus m3 &= 0 \\ c2 \oplus m1 \oplus m2 \oplus m3 &= 0 \end{aligned}$$

koşulları sağlanmalıdır. Buradan kontrol bitlerinin ,

$$\begin{aligned}
c_0 &= m_0 \oplus m_1 \oplus m_3 \\
c_1 &= m_0 \oplus m_2 \oplus m_3 \\
c_2 &= m_1 \oplus m_2 \oplus m_3
\end{aligned}$$

olması gerektiği bulunur.

Örnek olarak $m_3m_2m_1m_0 = 1101$ verisinin Hamming Kodu'nu çift eşlik kontrolü ile bulalım ;

$$\begin{aligned}
c_0 &= 1 \oplus 0 \oplus 1 = 0 \\
c_1 &= 1 \oplus 1 \oplus 1 = 1 \\
c_2 &= 0 \oplus 1 \oplus 1 = 0
\end{aligned}$$

ve kodlanmış sözcük $m_3m_2m_1c_2m_0c_1c_0 = 1100110$ olacaktır.

Bu kod sözcüğünün alıcı tarafta tek bitlik hata ile 1101110 olarak alındığını düşünelim. Yine çift eşlik kontrolü ile ,

$$\begin{aligned}
p_0 &= c_0 \oplus m_0 \oplus m_1 \oplus m_3 = 0 \oplus 1 \oplus 0 \oplus 1 = 0 \\
p_1 &= c_1 \oplus m_0 \oplus m_2 \oplus m_3 = 1 \oplus 1 \oplus 1 \oplus 1 = 0 \\
p_2 &= c_2 \oplus m_1 \oplus m_2 \oplus m_3 = 1 \oplus 0 \oplus 1 \oplus 1 = 1
\end{aligned}$$

bulunur. Böylece $p_2p_1p_0 = 100$ olduğundan 4. bitte hata olduğu sezilerek alınan sözcük 1100110 şeklinde düzeltilir.

4.2.3 - "Syndrome" Yöntemi

Veri bitlerinin sayısının k , eşlik bitlerinin sayısının ise r tane olduğunu düşünelim. Kodlanmış sözcük $k + r$ bit uzunluğunda olacaktır.

$$a_1a_2a_3 \text{ ----- } a_kc_1c_2 \text{ ----- } c_r$$

Burada a_i , veri sözcüğünün i . biti , c_j ise kontrol bitlerinin j . biti olacaktır. Eşlik bitleri aşağıdaki lineer eşitlikleri sağlayacak biçimde seçilirler ,

$$0 = h_{11}.a_1 \oplus h_{12}.a_2 \oplus \text{-----} \oplus h_{1k}.a_k \oplus 1.c_1 \oplus 0.c_2 \oplus \text{-----} \oplus 0.c_r$$

$$0 = h_{21}.a_1 \oplus h_{22}.a_2 \oplus \text{-----} \oplus h_{2k}.a_k \oplus 0.c_1 \oplus 1.c_2 \oplus \text{-----} \oplus 0.c_r$$

$$0 = h_{r1}.a_1 \oplus h_{r2}.a_2 \oplus \text{-----} \oplus h_{rk}.a_k \oplus 0.c_1 \oplus 0.c_2 \oplus \text{-----} \oplus 1.c_r$$

Burada hij katsayıları 0 veya 1 olup sabit değerlerdir. Yukardaki eşitlik matris gösterilimi ile yazılırsa ,

$$\bar{H} = \begin{bmatrix} h_{11} & h_{12} & \text{-----} & h_{1k} & 1 & 0 & 0 & \text{-----} & 0 \\ h_{21} & h_{22} & \text{-----} & h_{2k} & 0 & 1 & 0 & \text{-----} & 0 \\ \text{-----} & \text{-----} & \text{-----} & \text{-----} & \text{-----} & \text{-----} & \text{-----} & \text{-----} & \text{-----} \\ \text{-----} & \text{-----} & \text{-----} & \text{-----} & \text{-----} & \text{-----} & \text{-----} & \text{-----} & \text{-----} \\ h_{r1} & h_{r2} & \text{-----} & h_{rk} & 0 & 0 & 0 & \text{-----} & 1 \end{bmatrix}, \quad \bar{T} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_k \\ c_1 \\ \vdots \\ c_r \end{bmatrix}$$

böylece $\bar{H} \cdot \bar{T} = 0$ yazılabilir. Alınan sözcük \bar{R} , gönderilen kodlanmış sözcüğe eşit olmayabilir. Eğer $\bar{H} \cdot \bar{R} \neq 0$ ise en az bir hatanın olduğu anlaşılır.

\bar{E} hata matrisi olmak üzere, $\bar{R} = \bar{T} + \bar{E}$ şeklinde yazılabilir. Alınan sözcükte hata oluşmamışsa $\bar{E} = 0$ olacaktır. Örnek olarak verinin 11111 olarak gönderilmesine rağmen 01101 olarak alındığını düşünelim ,

$$\bar{R} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} \quad \bar{T} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad \bar{E} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

olur. Hatanın hangi bitte olduğunu bulmak için

$$\bar{S} = \bar{H} \cdot \bar{R} = \bar{H} \cdot \bar{T} + \bar{H} \cdot \bar{E} = \bar{H} \cdot \bar{E}$$

şeklinde bir vektör tanımlanabilir.Çünkü $\bar{H} \cdot \bar{T} = 0$ dir ve hata oluşmadığı zaman $\bar{H} \cdot \bar{E} = 0$ olacaktır.Sadece bir tek hata oluştuğunda \bar{E} matrisinde, karşı düşen satır elemanı 1 diğerleri 0 olur. Böylece ,

$$\bar{S} = \begin{bmatrix} h_{11} & h_{12} & \text{-----} & h_{1k} & 1 & 0 & \text{-----} & 0 \\ h_{21} & h_{22} & \text{-----} & h_{2k} & 0 & 1 & \text{-----} & 0 \\ \text{-----} & \text{-----} & \text{-----} & \text{-----} & \text{-----} & \text{-----} & \text{-----} & \text{-----} \\ \text{-----} & \text{-----} & \text{-----} & \text{-----} & \text{-----} & \text{-----} & \text{-----} & \text{-----} \\ h_{r1} & h_{r2} & \text{-----} & h_{rk} & 0 & 0 & \text{-----} & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ 1 \\ \vdots \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \bar{H} \\ \text{matrisinin} \\ \text{hatalı} \\ \text{sütunu} \end{bmatrix}$$

\bar{S} vektörü "syndrome" olarak isimlendirilir. Bir tıp terimi olup bir hastalığı karakterize eden belirtiler grubuna verilen isimdir.

Hatanın belirlenebilmesi için \bar{H} matrisinin tüm sütunları birbirinden farklı ve hiç bir sütundaki elemanların, hepsinin 0 olmaması gerekmektedir.

Örnek ; \bar{H} matrisinin aşağıda verilen şekilde olduğunu ve 1000 verisinin kodlanarak gönderileceğini varsayalım.

$$\bar{H} = \begin{array}{c} \begin{array}{ccccccc} a1 & a2 & a3 & a4 & c1 & c2 & c3 \end{array} \\ \hline \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

$$\bar{H} \cdot \bar{T} = \bar{H} \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ c1 \\ c2 \\ c3 \end{bmatrix} = 0 \text{ ve } \begin{bmatrix} 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus c1 \oplus 0 \oplus 0 \\ 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus c2 \oplus 0 \\ 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus c3 \end{bmatrix} = 0$$

ve buradan $c1 = c2 = c3 = 1$ bulunur. Kodlanmış sözcük 1000111 dir. Bu sözcüğün 1010111 olarak alındığını düşünelim ,

$$\bar{S} = \bar{H} \cdot \bar{R} = \bar{H} \cdot \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

"Syndrome" \bar{H} matrisinin açanca sütununa eşit olduğundan açanca bit $a3$ un düzeltilmesi gerektiği anlaşılır.

5 - RASGELE ERİŞİMLİ BELLEK ARABİRİMİ SİSTEMİ

5.1 - Sistem Tasarım Kriterleri

5.1.1 - Sistem Donanımı Tasarım Kriterleri

Sistem donanımının tasarımında mıknatıslı şerit mekanizmasının çalışma düzeni, arabirimde kullanılan MPF - 1P mikrobilgisayarının çevre birimlerinin sağladığı olanaklar göz önünde tutulmuştur. Mekanizmanın tamamen elektriksel kontrollü olması, kaset varlığını, konumunu ve yazma korumasını sezme olanaklarını vermesi basit devreler ile mekanizma kontrolünün yapılmasına olanak sağlamıştır.

Mekanizma kontrolü, kaset varlığı ve yazma koruması sezme işlemleri PIA (Paralel İletişim Arabirimi) aracılığı ile mikroişlemci tarafından yapılmaktadır. Kaset konumu, belirleme ve istenen programın bulunması ise SZD (Sayıcı/Zamanlayıcı Devresi) ile yapılmaktadır. Mıknatıslı şerit üzerine yazılacak verinin bilgisayardan seri olarak alınıp kodlanması ve yazılması, verinin kasetten okunup çözülmesi amacıyla SIA (Seri İletişim Arabirimi) kullanılmaktadır.

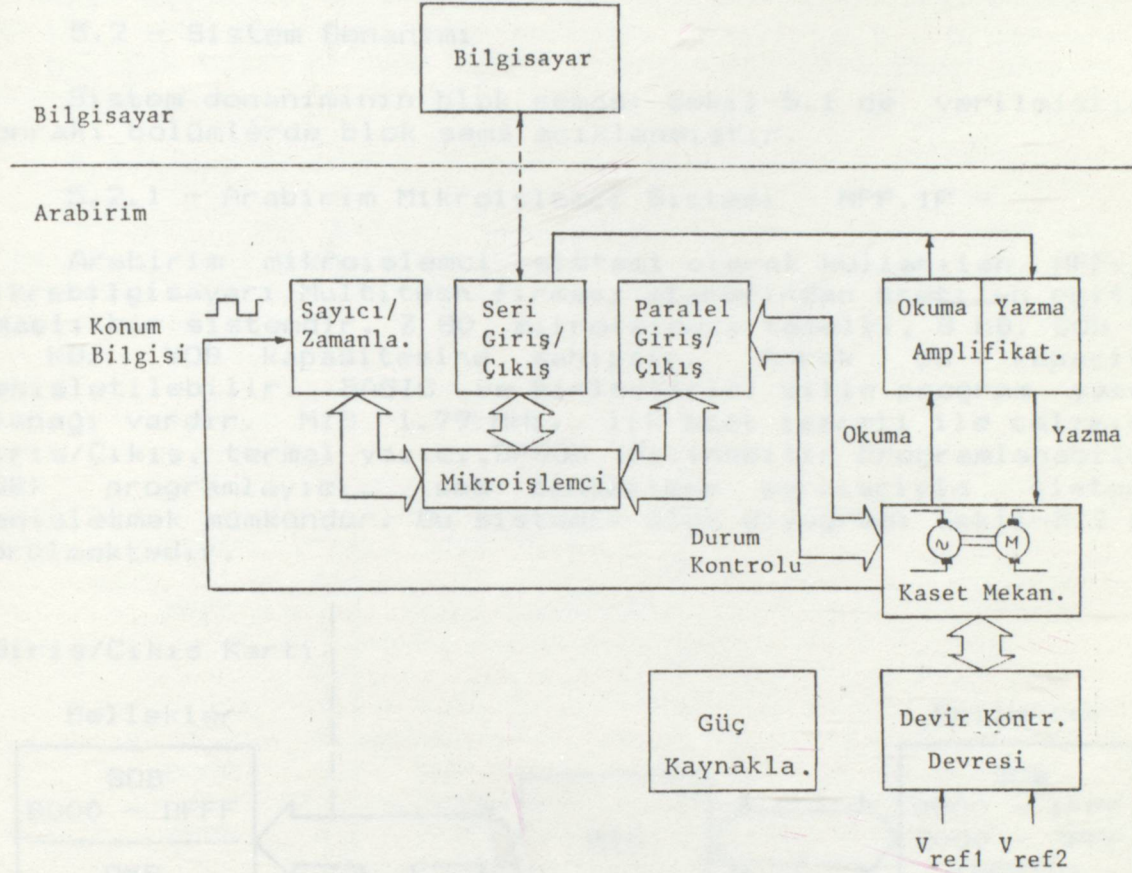
Sistemin donanım yapısını diğer bölümlerini devir kontrol devresi ve amplifikatörler teşkil ederler. Önceleri gerçekleştirilmesi düşünülen çift hızda okuma ve yazma işlemleri nedeniyle devir kontrol devresi çift hızda çalışacak şekilde tasarlanmıştır. Bu devre kasette oluşabilecek sıkışmaların yada değişik kaset özelliklerinin yazma ve okuma işlemi sırasında sorun oluşturmaması için düşünülmüştür.

Yazma ve okuma amplifikatörleri de çift kanal yazma ve okuma yapabilecek şekilde düzenlenmiştir. Yazma amplifikatörlerinin, mıknatıslı şerit üzerine daha iyi kayıt yapabilmek amacıyla NZR yöntemiyle kayıt yapabilecek yetenekte olması düşünülmüştür. Yaygın olarak kullanılan bir amplifikatör devresi ile okuma amplifikatörleri oluşturulmuştur.

5.1.2 - Sistem Yazılımı Tasarım Kriterleri

Sistem yazılımının oluşturulmasında sistem donanımının özellikleri temel etkindir. Çünkü sistemin çalışması büyük

ölçüde, kullanılan mikrobilgisayarın çevre birimlerine bağlıdır. Bunlara hükmetmek ise yazılım ile mümkündür.



Sekil-5.1 Arabirim sistemi blok diyagramı

Verinin kodlanması ve okunan verinin kod çözülmesi de yazılım ile yapılmaktadır. Çevre birimlerinin değişik çalışma durumları şartlandırılması ve çalıştırılması için altprogramlar yazılmıştır. Bunlar, işletim sistemi programlarının çeşitli yerlerinde çağrılmaktadırlar. İşletim sistemi programlarının daha hızlı çalışması ve daha az bellek harcaması için mümkün olduğu kadar yapısal yazılmasına çalışılmıştır.

Bazı bellek gözleri belli işlemlerin yapılıp yapılmadığını kontrol amacıyla kullanılmıştır. Veri kodlanması ve kod çözülmesinde, okunan bazı verilerin beklenen veriler olup olmadığının kontrolunda gerekli tablolar ve liste yapılarının saklanması için ayrılmış bellek gözleri önemli yer işgal ederler.

Bu çalışmada bilgisayar tarafında yazılması gereken

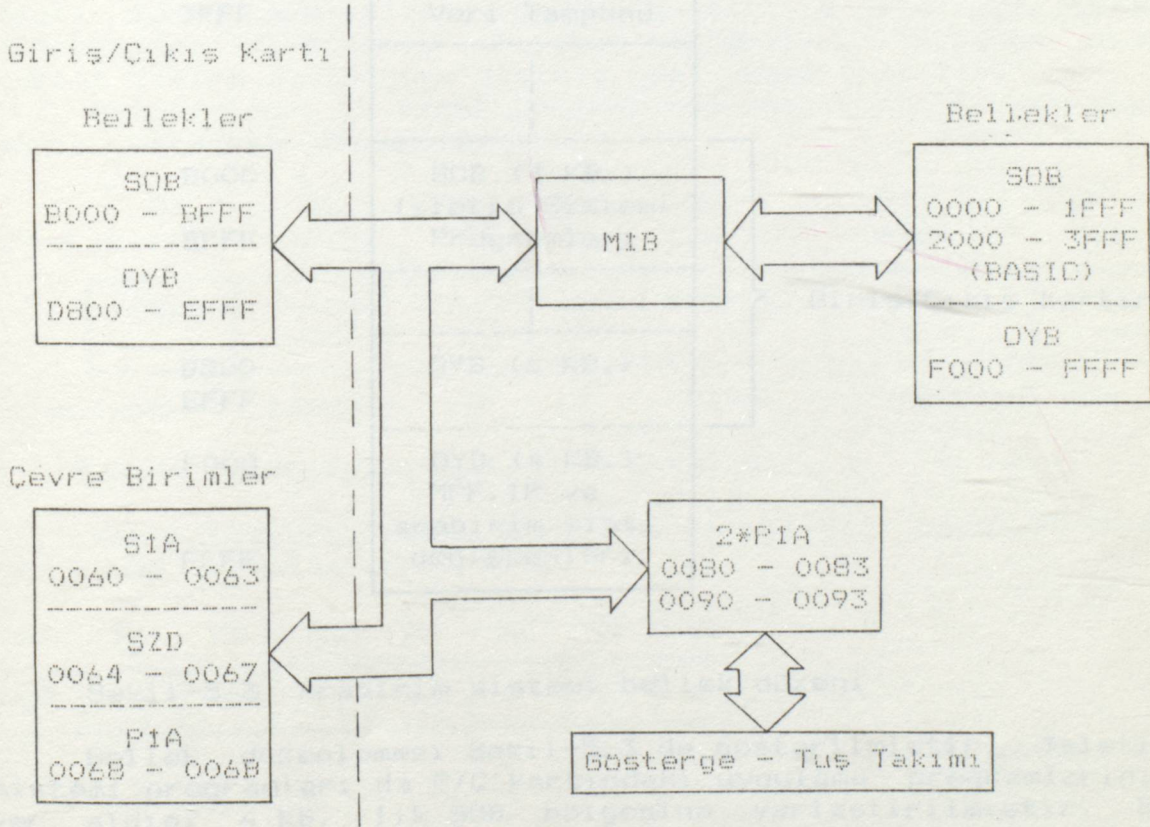
işletim sistemi programları yoktur. Sistemin çalışmasının gerçek olarak denenebilmesi için bu programların da yazılıp, bu sistemi kullanacak olan bilgisayara yüklenmesi gerekir.

5.2 - Sistem Donanımı

Sistem donanımının blok şeması Şekil-5.1 de verilmiştir. Sonraki bölümlerde blok şema açıklanmıştır.

5.2.1 - Arabirim Mikroişlemci Sistemi - MPF.1P -

Arabirim mikroişlemci sistemi olarak kullanılan MPF.1P mikrobilgisayarı Multitech firması tarafından üretilen eğitim amaçlı bir sistemdir. Z 80 mikroişlemci temelli, 8 KB. SOB ve 4 KB. YOB kapasitesine sahiptir. Ancak bu kapasite genişletilebilir. BASIC ve birleştirici dille program yazma olanığı vardır. MIB 1.79 MHz. lik saat işareti ile çalışır. Giriş/Çıkış, termal yazıcı, SPSOB (Silinebilir Programlanabilir SOB) programlayıcı, ses sentezleme kartlarıyla sistemi genişletmek mümkündür. Bu sistemin blok diyagramı Şekil-5.2 de görülmektedir.



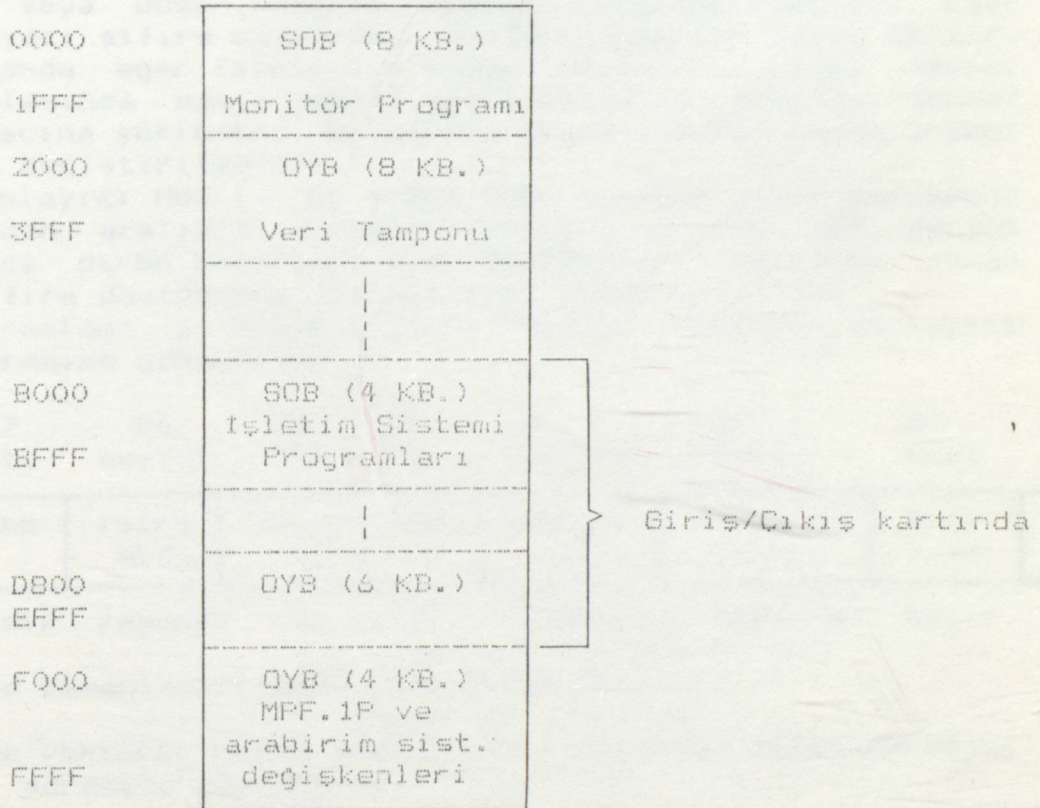
Şekil-5.2 MPF.1P mikrobilgisayarının blok diyagramı

Giriş/Çıkış kartı üzerinde PIA, SIA ve SZD çevrebirim leriyle 6 KB. lık ek YOB olanağı ve bazı uygulama programlarının saklanması için 4 KB. lık SOB olanağı vardır.

20 karakter kapasiteli gösterge ve 49 tuş kapasiteli tuş takımı, kullanıcı ile ilişkisini sağlar. Tasarlanan sistemde, MPF.1P' in monitör programındaki bazı alt programlar, kullanıcıyı uyararak veya kullanıcının bazı işlemleri yapmasını istemek için kullanılmıştır.

5.2.1.1 - Arabirim Sistemi Bellek Düzeni

MPF.1P' in bellek düzeninde 2000 - 3FFF adresleri arasında yer alan SOB, her 8 KB. lık program verisinin saklanması için çıkarılmıştır. Bu adreslere 8 KB. lık YOB yerleştirilmiştir.



Sekil-5.3 Arabirim sistemi bellek düzeni

Bellek düzenlemesi Sekil-5.3 de gösterilmiştir. İşletim sistemi programları da G/C kartındaki uygulama programlarının yer aldığı 4 KB. lık SOB bölgesine yerleştirilmiştir. Bu programlar B000 adresinden başlar. F000 - FFFF adresleri

arasında ise MPF.1P ve tasarlanan arabirim sisteminin deęişkenleri vardır. Bunlardan en fazla bellek gözünü FB10 - FAFB adresleri arasında yer alan "liste yapısı tamponu" tutar. Bellek düzeninin donanım şeması Ek-1.a da görülebilir.

5.2.1.2 - Arabirim Sistemi Çevrebirimleri ve Kullanımı

** SZD (Sayıcı/Zamanlayıcı Devresi) ; Z B0 mikroişlemci sistemine uyumludur. Sayıcı ve zamanlayıcı fonksiyonlarına sahip 4 kanallı vardır. Her bir kanal sayıcı veya zamanlayıcı olarak programlanabilir. Kanal-0 dan kanal-3 e doğru kesme önceliğine sahip kesme düzeni vardır.

Sayıcı Mod ; Bu modda SZD, CLK/TRG girişindeki işaretin kenarlarını sayar. Her bir kenardan sonra, sistem saatinin yükselen kenarı ile aşağı sayacı bir eksiltir. Saymanın yükselen veya düşen kenarla olması programlanabilir. Eğer aşağı sayacı sıfıra ulaşırsa, ZC/TO çıkışı bir darbe üretir. Aynı zamanda eğer izinli ise kesme üretilir. Aşağı sayıcı sıfıra ulaştınca zaman sabiti saklayıcısının içeriği tekrar aşağı sayacına yüklenir. Bu saklayıcının içeriği sayma işlemi sırasında değiştirilebilir.

Zamanlayıcı Mod ; Bu modda SZD, sistem saati periyoduna bağlı zaman aralıklı darbeler üretir. 1 ile 256 periyot uzunluğunda darbe katarları elde edilebilir. Yüklenen zaman sabiti sıfıra düştüğünde (izinli ise) kesme üretilir.

Programlama ; Kanal kontrol sözcüğü kontrol saklayıcı sınıfın adresine gönderilir.

D7 D6 D5 D4 D3 D2 D1 D0
1 → izinli sayıcı 256 CLK/TRG evet evet

kesme	işlem modu	bölme *	eğim	tetickl. *	zaman s. izler	sıfır lama	1
-------	------------	---------	------	------------	----------------	------------	---

0 → izinsiz zamanl. 16 otom. hayır hayır

* ; Sadece zamanlayıcı modu için önemlidir.

Kesme vektörü, kesme modu-2 de çalışıldığı zaman herhangi bir kanal adresine gönderilir.

D7	D6	D5	D4	D3	D2	D1	D0
V7	V6	V5	V4	V3	*	*	0

0 0 = kanal-0 (en öncelikli kesme)

0 1 = kanal-1

1 0 = kanal-2

1 1 = kanal-3 (en az öncel. kesme)

* ; SZD tarafından eklenir.

** P1A (Paralel İletişim Arabirimi) ; Dış birimlerle mikroislemcinin kontrolunda çift yönlü paralel veri alışverişini sağlar. İki bağımsız giriş/çıkış iskelesi vardır. İskele seçme ucuyla A ve B iskeleleri seçilir. Kontrol/veri işareti ile gönderilen verinin kontrol veya işlem verisi olduğu bildirilir. Her bir iskele için kullanılan iki el sıkışma hattı ile kesme izni varsa kesme hizmet programına gidilerek el sıkışmalı veri alışverişi sağlanır. 4 farklı moda çalışma olanağı vardır ;

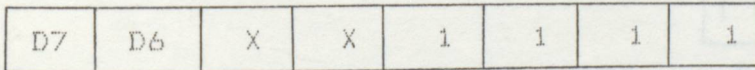
Mod	İşlem
0	8 bit çıkış
1	8 bit giriş
2	8 bit çift yönlü iletişim (*)
3	bit programlama (**)

(*) ; El sıkışma hatlarına bağlı olarak
(**) ; Her bir bit giriş veya çıkış olarak programlanabilir.

Programlama ; Her iskeleye ait kontrol saklayıcısına, kontrol sözcüğünün yüklenmesi ile iskele programlanabilir. 4 temel kontrol sözcüğü vardır ;

- 1 - Mod kontrol sözcüğü
- 2 - Kesme kontrol sözcüğü
- 3 - Kesme izni kontrol sözcüğü
- 4 - Kesme vektörü kontrol sözcüğü

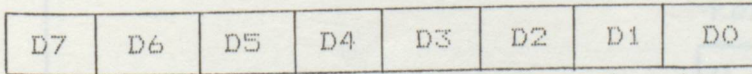
İlk Sekizli



Mod Kontrol Sözcüğü

0	0	=	mod	0
0	1	=	mod	1
1	0	=	mod	2
1	1	=	mod	3

İkinci Sekizli (mod 3 için)



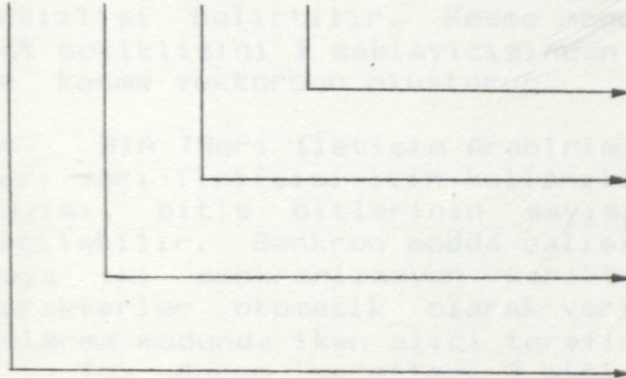
G/Ç Saklayıcısı K. Sözc.

0	=	çıkış için program.
1	=	giriş için program.

Mod kontrol sözcüğü ile iskelenin hangi modda kullanılacağı belirtilir. Eğer mod 3 seçilmişse G/Ç saklayıcısı kontrol sözcüğü hangi bitlerin giriş veya çıkış olacağı belirtir.

İlk Sekizli

D7	D6*	D5*	D4*	0	1	1	1
----	-----	-----	-----	---	---	---	---



KESME KONTROL SÖZCÜĞÜ

0 = maske sözcüğü izlemez
1 = maske sözcüğü izler

0 = giriş bitleri 0 aktif
1 = giriş bitleri 1 aktif

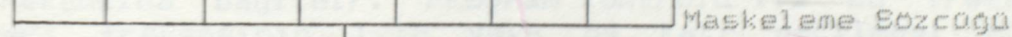
0 = VEYA işlemi
1 = VE işlemi

0 = kesme izni yok
1 = kesme izni var

* ; Sadece mod 3 için kullanılırlar.

İkinci Sekizli (D4=1 ise)

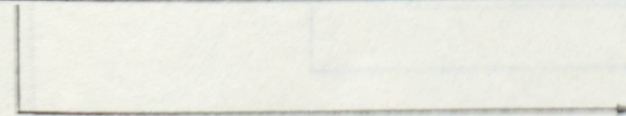
D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----



0 = maskesiz
1 = maskeli

Kesme kontrol sözcüğünde, D4 maske sözcüğünün izleyip izlemediğini belirtir. D5 ile giriş bitlerinin hangi seviyede aktif olduğu seçilir. D6 biti 0 ise bitlerden en az biri aktif, 1 ise hepsi aktif olduğunda kesme üretilir. D7 biti ile kesmeye izin verilir veya verilmez. Maskeleme sözcüğünde 0 olan bitler kesme üretmek için kullanılabilir.

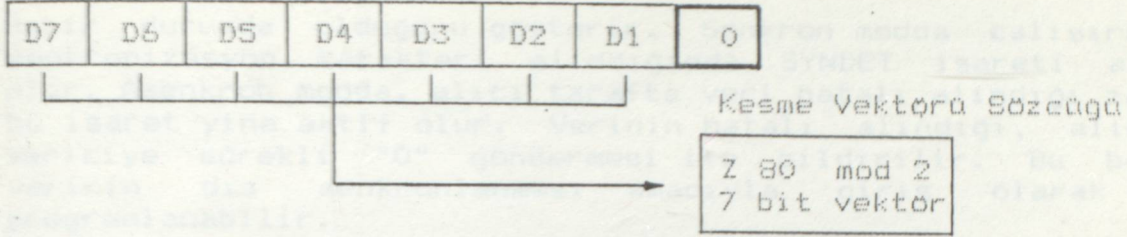
D7	X	X	X	0	0	1	1
----	---	---	---	---	---	---	---



KESME İZİN SÖZCÜĞÜ

0 = kesme izinsiz
1 = kesme izinli

Kesme izin kontrol sözcüğü ilgili iskelenin herhangi bir yolla kesme üretmeye izinli olup olmadığını belirtir.



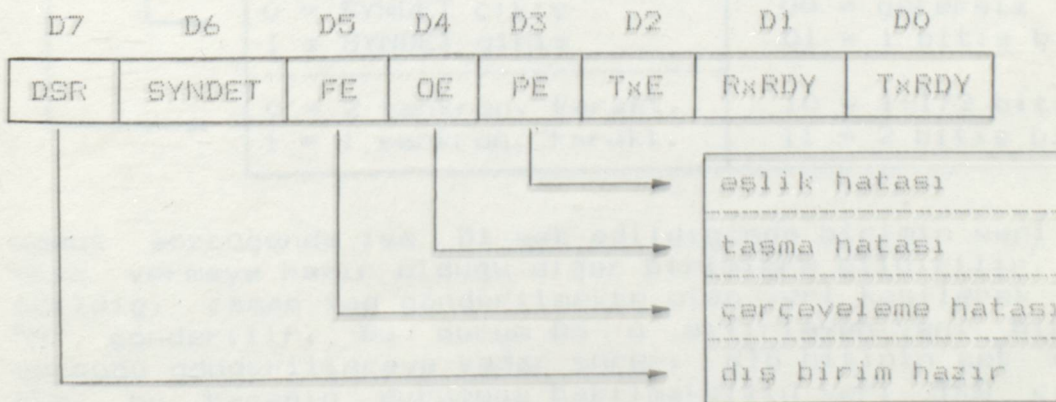
Kesme vektörü kontrol sözcüğü ile kesme vektörünün alt sekizlisi belirtilir. Kesme modu 2 de MIB kesme vektörünün üst sekizlisini I saklayıcısından alır. Bu iki sekizli beraber ce kesme vektörünü oluşturur.

** SIA (Seri İletişim Arabirimi) ; Senkron veya asenkron seri veri iletişimi için kullanılabilir. Veri sözcüğündeki bit sayısı, bitiş bitlerinin sayısı ve eşlik biti program ile seçilebilir. Senkron modda çalışırken her veri çerçevesi bir veya iki senkronizasyon karakteri ile belirlenebilir. Bu karakterler otomatik olarak verici tarafından üretilir veya avlanma modunda iken alıcı tarafından beklenir.

Tüm durum işaretleri 8 bitlik bir durum saklayıcısından izlenebilir. Bu saklayıcıda eşlik hatası, çerçeveleme hatası, ve taşma hatasını gösteren bayraklar vardır. Sadece okunabilir özelliğe sahiptir.

Saat frekansı veri hızıyla ilgili değildir. Ancak maksimum veri hızının en az 30 katı olmalıdır. Genellikle sistem saatine bağlanır. Gönderilen verinin hızı TxC girişinde ki, alınan verinin hızı da RxC girişindeki saat işaretlerinin frekansına bağlıdır. Program kontrolü ile bu frekansların, veri frekansının 1,16 veya 64 katı seçilmesi mümkündür. Böylece saat frekansı değiştirilmeden veri hızı değiştirilebilir.

Dış kontrol amacıyla kullanılan 4 durum kontrol işareti vardır. Bu işaretler de durum saklayıcısından izlenebilir. Durum saklayıcısı ,

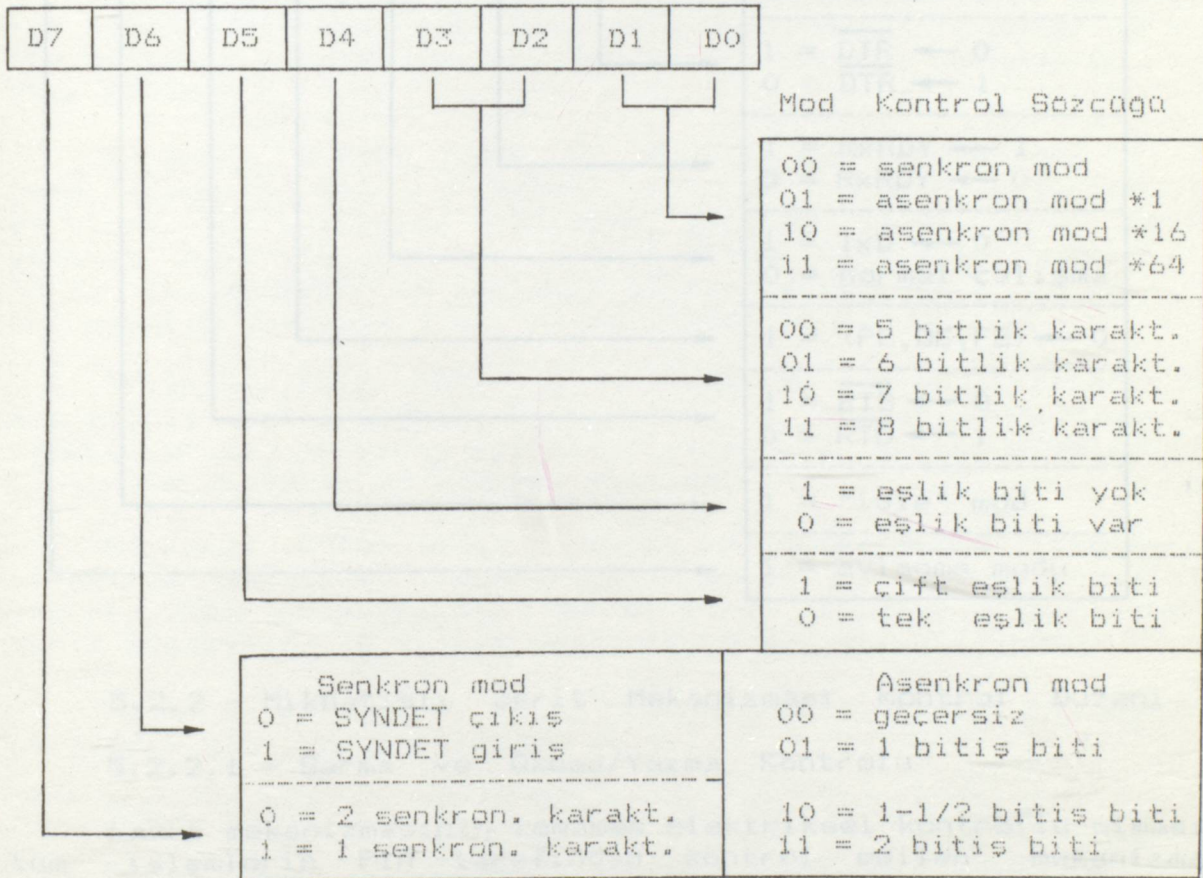


şeklindedir. TxE verici tamponunun boş olduğunu ve hiçbir verinin iletilmediğini, TxRDY ve RxRDY ise verici ve alıcının

hazır durumda olduğunu gösterir. Senkron modda çalışırken, senkronizasyon karakteri alındığında SYNDET işareti aktif olur. Asenkron modda, alıcı tarafta veri hatalı alındığı zaman bu işaret yine aktif olur. Verinin hatalı alındığı, alıcını vericiye sürekli "0" göndermesi ile bildirilir. Bu bacak verinin dış senkronlanması amacıyla giriş olarak da programlanabilir.

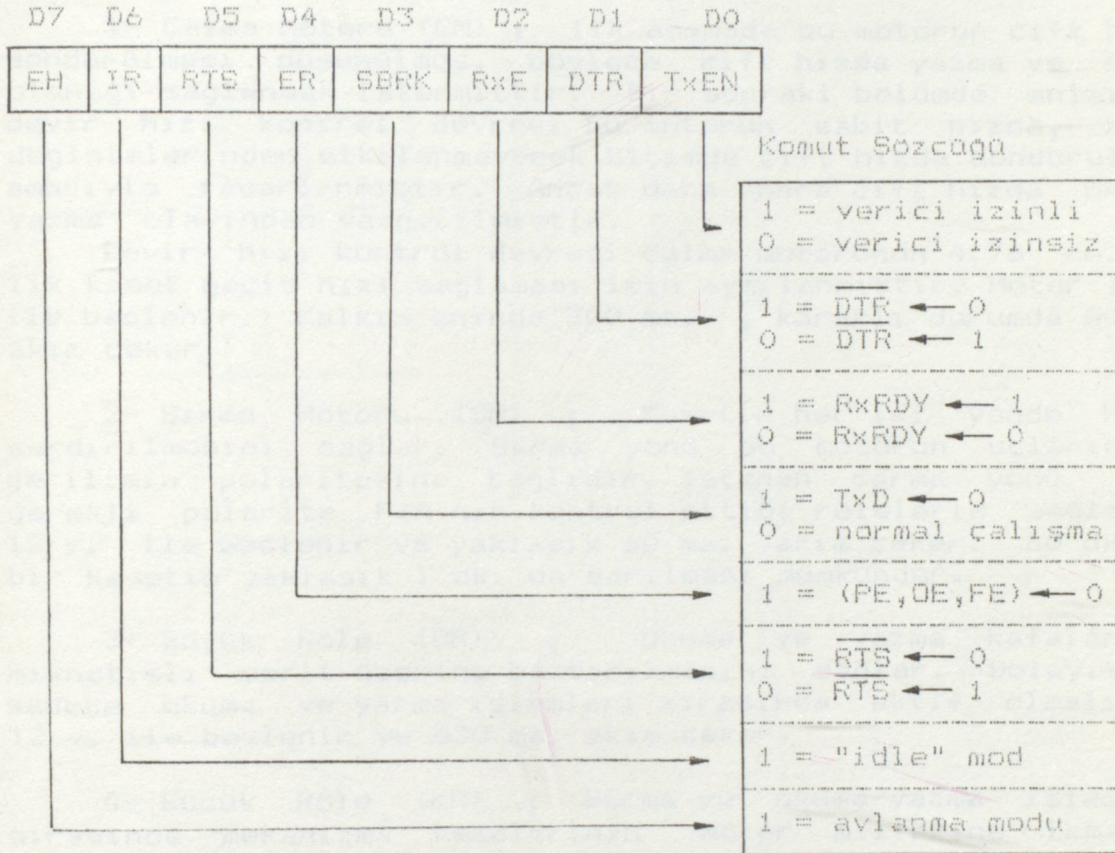
Durum saklayıcısında tüm bitler ilgili durumlar oluştuğu zaman set edilirler. Durum saklayıcısı MIB tarafından okunduğu zaman tüm bitler sıfırlanırlar. Hata bitleri aynı zamanda komut sözcüğündeki ER biti ile sıfırlanırlar.

Programlama ;



Komut sözcüğünde ise D1 set edildiğinde birimin veri almaya veya vermeye hazır olduğu diğer birimlere bildirilir. D3 set edildiği zaman ise gönderilmekte olan veri kesilerek sürekli "0" gönderilir. Bu durum D3'ü sıfırlayan yeni bir komut sözcüğü gönderilinceye kadar sürer. RTS bitinin set edilmesi ile, bu bacağın durumuna bakılmaksızın veri TxD çıkışından gönderilebilir. Bu birimi sıfırlamak, tüm işlemleri durdurmak için D6 set edilir. Yeni bir işlem için birim tekrar şartlanmalıdır.

D7 nin set edilmesi ile senkron çalışmada senkronizasyon karakterlerini yakalamak için avlanma moduna girilir. Bu bit yeni bir komut sözcüğü ile sıfırlanabilir, D6 nin set edilmesiyle sıfırlanabilir yada senkronizasyon karakterleri yakalandığında sıfırlanır.



5.2.2 - Mıknatıslı Şerit Mekanizması Kontrol Düzeni

5.2.2.1 - Sarma ve Okuma/Yazma Kontrolü

Kaset mekanizmasının tamamen elektriksel kontrollü olması tüm işlemlerin PIA tarafından kontrol edilen mekanizma röleleri yardımıyla yapılmasını sağlar. Mekanizma üzerindeki iki rölenin ve iki D.C. motorun konumlarına göre sağa veya sola hızlı sarma ve çalma (sağa normal hızda dönüş - 4.76 cm/sn. - ve okuma/yazma kafalarının aktif olması durumu) işlemleri yapılır.

Mekanizma röleleri doğrudan PIA tarafından sürülemeyeceğinden açık kollektörlü tümleme kapıları tarafından sürülen röleler kullanılmıştır. Ayrıca okuma işlemi sırasında yazma amplifikatörü çıkışının yazma kafasından ayrılması gerekir. Bu

işlem yapılmazsa kasetteki veri bozulacaktır. Yazma amplifikatörü ile yazma kafası arasında kullanılan bir röle ile bu sorun çözülmüştür. Kullanılan röleler tek yönlü çift kutuplu rölelerdir.

Böylece mekanizma kontrolü için 4 değişken yazılabilir ;

1- Çalma Motoru (ÇM) ; İlk aşamada bu motorun çift hızda döndürülmesi düşünülmüş, böylece çift hızda yazma ve okuma olanığı sağlanmak istenmiştir. Bir sonraki bölümde anlatılan devir hızı kontrol devresi bu motorun sabit hızda, kaset değişimlerinden etkilenmeyecek biçimde çift hızda döndürülmesi amacıyla tasarlanmıştır. Ancak daha sonra çift hızda okuma/yazma olayından vazgeçilmiştir.

Devir hızı kontrol devresi çalma motorunun 4.76 cm./sn. lik kaset geçiş hızı sağlaması için ayarlanmıştır. Motor 12 v. ile beslenir. Kalkış anında 300 ma. , kararlı durumda 80 ma. akım çeker.

2- Sarma Motoru (SM) ; Kasetin her iki yönde hızlı sarılmasını sağlar. Sarma yönü bu motorun uçlarındaki gerilimin polaritesine bağlıdır. İstenen sarma yönü için gerekli polarite PIA'nın kontrol ettiği rölelerle sağlanır. 12 v. ile beslenir ve yaklaşık 60 ma. akım çeker. 60 dk.lık bir kasetin yaklaşık 1 dk. da sarılması mümkündür.

3- Büyük Röle (BR) ; Okuma ve yazma kafalarının miknatıslı şerit üzerine bastırılmasını sağlar. Dolayısıyla sadece okuma ve yazma işlemleri sırasında aktif olmalıdır. 12 v. ile beslenir ve 530 ma. akım çeker.

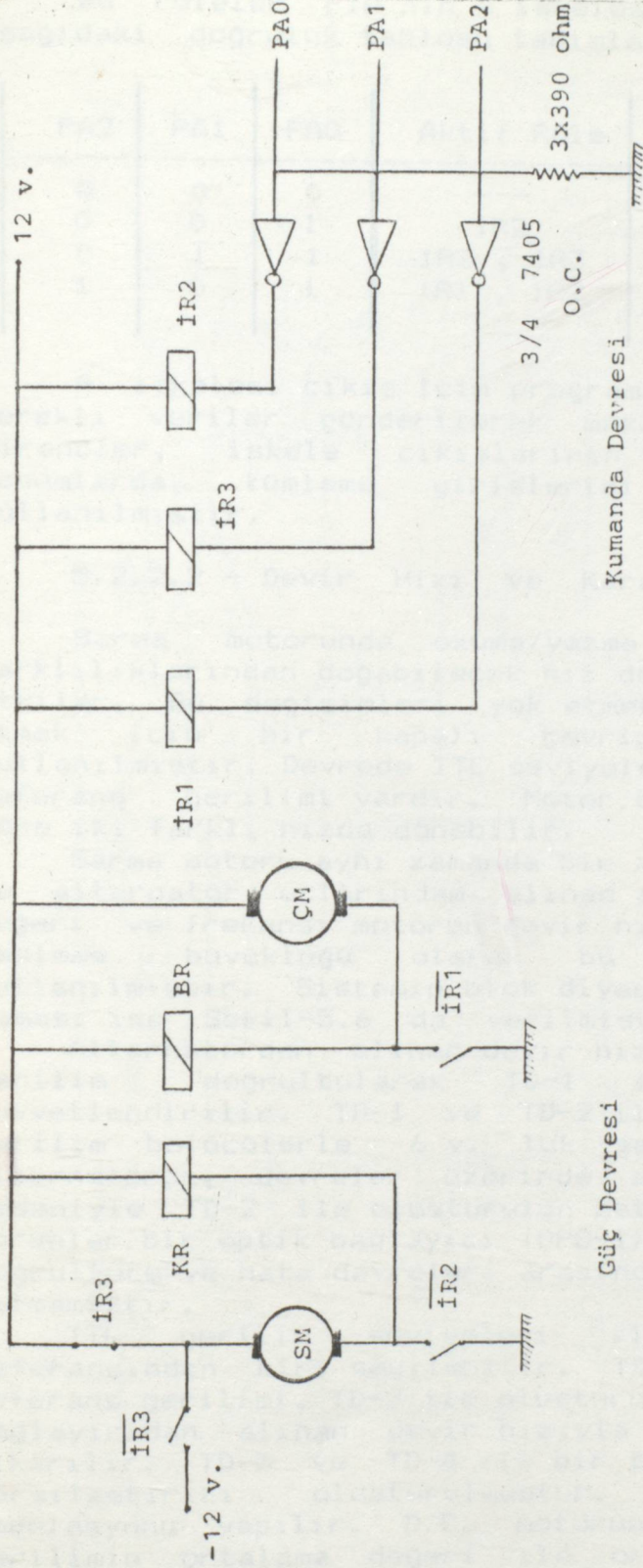
4- Küçük Röle (KR) ; Sarma ve okuma/yazma işlemleri sırasında mekanizma keçelerinin motor millerine temasını sağlar. Hem okuma/yazma, hem de sarma işlemleri sırasında aktif olmalıdır. 12 v. ile beslenir ve 270 ma. akım çeker.

Böylece çalma ve sarma işlemlerinin lojik ifadeleri aşağıdaki gibi yazılabilirler ;

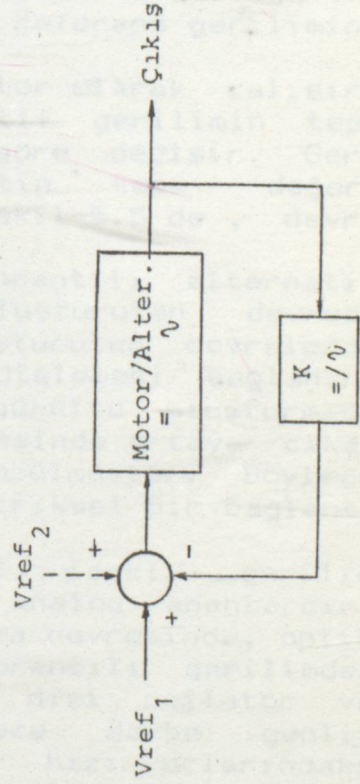
Sağa Sarma	SR = SM(-12 v.).KR
Sola Sarma	SL = SM(+12 v.).KR
Çalma	ÇA = SM(+12 v.).ÇM.KR.BR = ÇM.SL.BR

Bu ifadeleri sağlayan güç ve kumanda devreleri Şekil-5.4 de verilmiştir. Kullanılan röleler ve görevleri ise ;

- İşlem Rölesi 1 (IR1) : Diğer işlem röleleriyle birlik te çalma işleminin yapılmasını sağlar.
- İşlem Rölesi 2 (IR2) : IR3 ile birlikte sola ve sağa sarma işlemlerini yapar.
- İşlem Rölesi 3 (IR3) : IR2 ile birlikte sağa ve sola sarma işlemlerini yapar.



Şekil-5.4 Sarma ve okuma/yazma işlemlerinin sağlanması



Şekil-5.5 Motor devir hızı kontrol devresi blok diyagramı

Bu röleler P1A nin A iskelesi ile sürülürler. Böylece aşağıdaki doğruluk tablosu tanımlanabilir ;

PA2	PA1	PA0	Aktif Röle	İşlem	İsk. Verisi (hex)
0	0	0	---	dur	00
0	0	1	1R2	SL	01
0	1	1	1R2 , 1R3	SR	03
1	0	1	1R1 , 1R2	CA	05

A iskelesi çıkış için programlanıp, istenen işlem için gerekli veriler gönderilerek mekanizma kontrolü yapılır. Dirençler, iskele çıkışlarının yüksek empedans olduğu konumlarda, tümleme girişlerini "0" yapmak amacıyla kullanılmıştır.

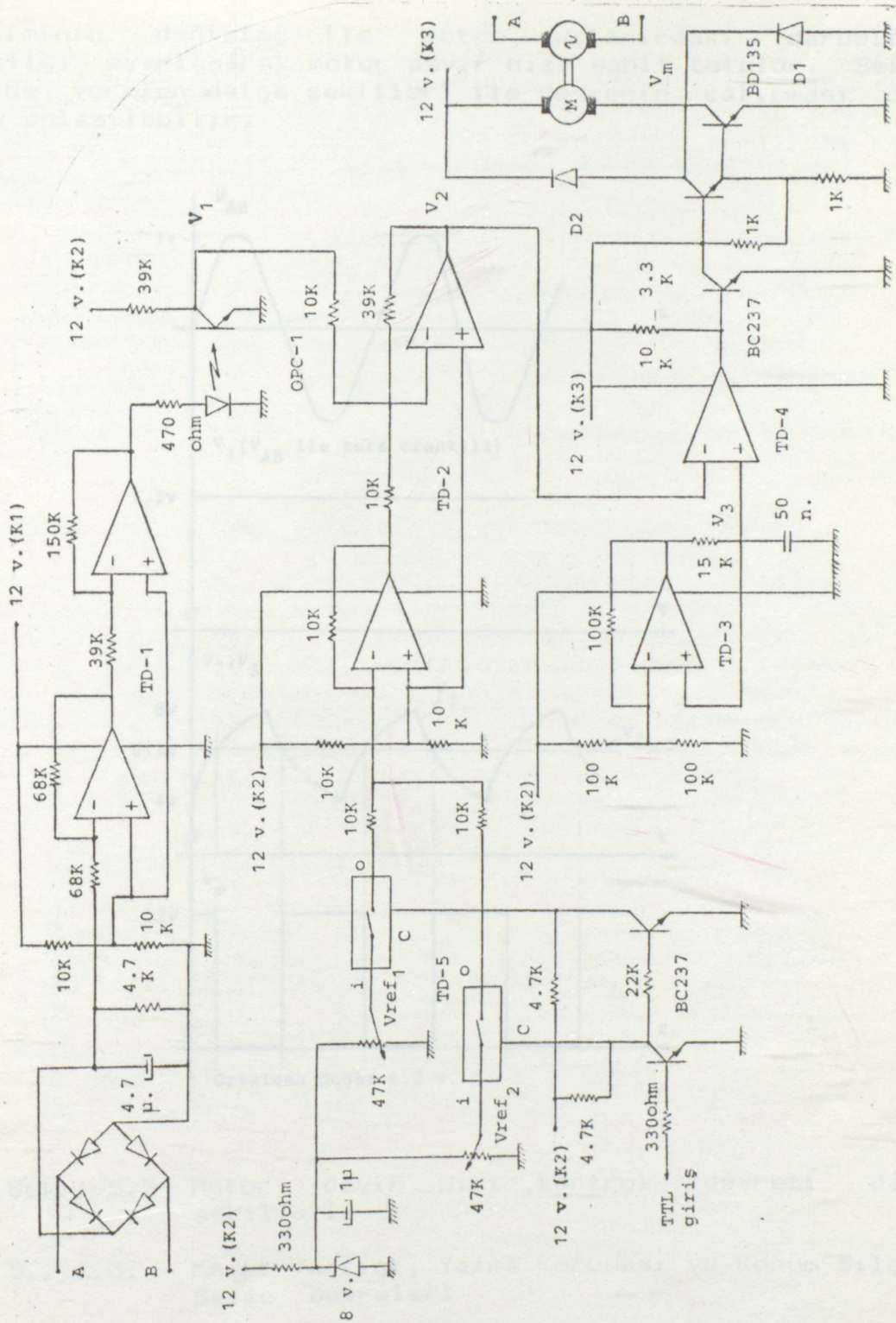
5.2.2.2 - Devir Hızı ve Kararlılığının Kontrolü

Sarma motorunda okuma/yazma işlemi sırasında kaset farklılıklarından doğabilecek hız değişimleri veri güvenliğini etkiler. Bu değişimleri yok etmek ve sabit devir hızı elde etmek için bir kapalı çevrim-oran kontrol devresi kullanılmıştır. Devrede TTL seviyeleriyle kontrol edilebilen iki referans gerilimi vardır. Motor seçilen referans gerilimine göre iki farklı hızda dönebilir.

Sarma motoru aynı zamanda bir alternatör olarak çalışır. Bu alternatör uçlarından alınan alternatif gerilimin tepe değeri ve frekansı motorun devir hızına göre değişir. Geri besleme büyüklüğü olarak bu işaretin tepe değeri kullanılmıştır. Sistemin blok diyagramı Şekil-5.5 de , devre şeması ise Şekil-5.6 da verilmiştir.

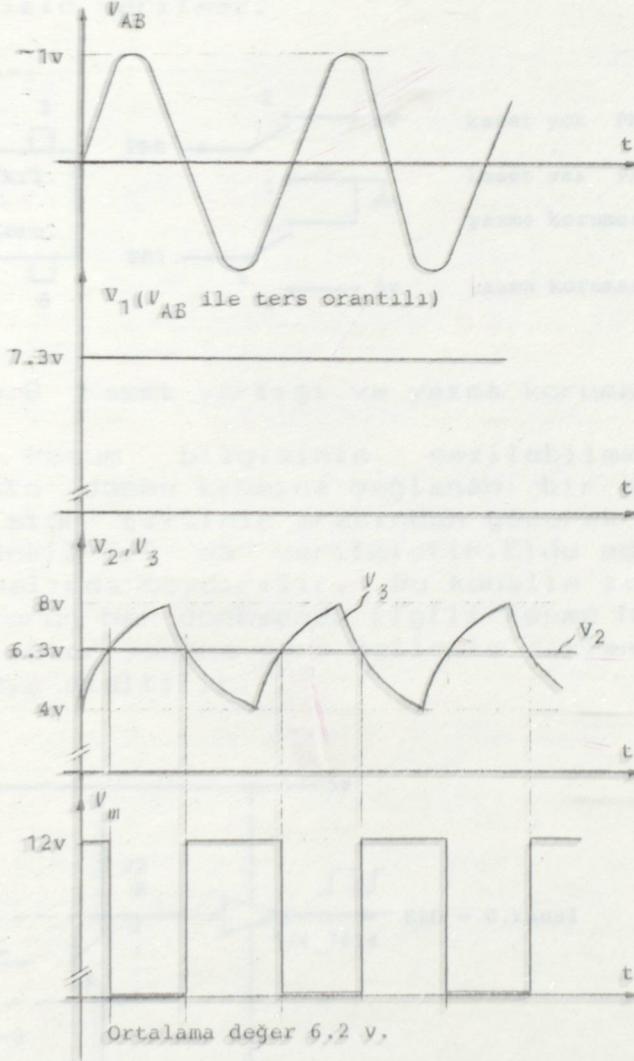
Alternatörden alınan devir hızıyla orantılı alternatif gerilim doğrultularak TD-1 ile oluşturulan devrede kuvvetlendirilir. TD-1 ve TD-2 ile oluşturulan devrelerde gerilim bölücülerle 6 v. luk seviye ötelemesi sağlanır. Alternatörün, devreler üzerinde aşırı gürültü oluşturması nedeniyle TD-2 ile oluşturulan hata devresinde ortaya çıkan sorunlar bir optik bağlayıcı (OPC-1) ile çözülmüştür. Böylece doğrultucu ve hata devreleri arasında elektriksel bir bağlantı kalmamıştır.

TTL gerilim seviyeleri ile iki farklı gerilim referansından biri seçilebilir. TD-5 bir analog anahtardır. Referans gerilimi, TD-2 ile oluşturulan hata devresinde, optik bağlayıcıdan alınan devir hızıyla ters orantılı gerilimden çıkarılır. TD-3 ve TD-4 ile bir testere dişi osilatör ve karşılaştırıcı oluşturulmuştur. Böylece darbe genlik modülasyonu yapılır. D.C. motorun devir hızı uçlarındaki gerilimin ortalama değeri ile orantılı olduğundan, hata



Şekil-5.6 Motor devir hızı kontrol devresi

geriliminin deęişimi ile motor uçlarındaki darbelerin genişliği ayarlanarak motor devir hızı sabit tutulur. Şekil-5.7 de verilen dalga şekilleri ile devrenin çalışması daha kolay anlaşılabilir.

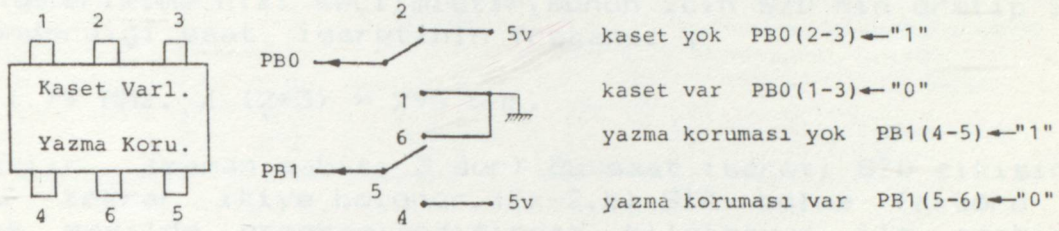


Şekil-5.7 Motor devir hızı kontrol devresi dalga şekilleri

5.2.2.3. - Kaset Varlığı, Yazma Koruması ve Korun Bilgisi Sezme Devreleri

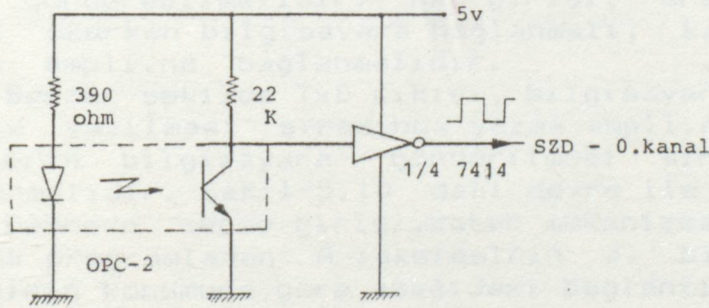
Arabirimde veri yazmak yada okumak amacıyla kaset bulunup bulunmadığı, kullanılan kasette yazma koruması olup olmadığı sezilebilir. Bu işlemler mekanizma üzerinde bulunan ye

sözedilen durumlara göre konum deęiřtiren bir anahtar takımı yardımıyla yapılır. Bu anahtar takımına uygun gerilim seviyeleri verimistir ve çıkıř uçları PIA nın B iskelesine baęlıdır. (Şekil-5.8) İřletim sistemi programlarının bařında B iskelesi okunur. Mekanizmada kaset bulunmadığı sezilirse uyarı verilir. Aynı şekilde yazma koruması varsa kasete yeni veri yazılmasına izin verilmez.



Şekil-5.8 Kaset varlığı ve yazma koruması sezme

Kaset konum bilgisinin sezilebilmesi için kaset mekanizmasının dönen kısmına baęlanan bir eleman, bir foto diyot-transistör çiftinin arasından geçerek darbeler üretir. Bu devre Şekil-5.9 da verilmiştir. Elde edilen darbeler SZD nin 0. kanalında saydırılır. Bu kanalın zaman sabiti 1 dir. Böylece motorun her dönüşünde ilgili kesme hizmet programına gidilerek, motor yönüne göre bellekte saklanan konum bilgisi artırılır veya azaltılır.



Şekil-5.9 Konum bilgisi sezme devresi

5.2.2.4 - Bilgisayar ve Okuma/Yazma Amplifikatörleri ile İletişim

Arabirim mikroişlemcisinin, kullanıcı bilgisayar ve amplifikatörler ile iletişimi SIA aracılığı ile ve asenkron olarak yapılmaktadır. Haberleşme hızlarını, SIA nın RxC ve TxC

saat işaretlerinin frekansları belirler. MPF.1F sisteminde SIA nin bu iki girişi kısa devre edilerek SZD nin 2.kanalına (ZC/TD çıkışına) bağlanmıştır. Sistem saati bir flip-flop tarafından ikiye bölünür.Bu işaret SZD nin 2. kanalında sayılır.Zaman sabitinin her sıfıra inişinde ZC/TD çıkışı, dolayısıyla SIA nin RxC ve TxC girişleri aktif olur.

Bu kanal iki durum için programlanır ;

1- Bilgisayar ile arabirim arasında haberleşme için 9600 baud haberleşme hızı seçilmiştir.Bunun için SZD nin ürettiği SIA ne gönderdiği saat işaretinin frekansı ,

$$1.79 \text{ MHz.} / (2*3) = 298 \text{ KHz.}$$

olacaktır. (zaman sabiti 3 dür) Bu saat işareti SZD çıkışında tekrar tekrar ikiye bölünür.(Ek-2.b) SIA bölme faktörü 16 olacak şekilde programlandığından bilgisayar ile arabirim arasındaki haberleşme hızı ,

$$hba = 298 \text{ KHz.} / (2*16) = 9323 \text{ bit/sn.} \quad \text{olacaktır.}$$

2- Amplifikatörlerle arabirim arasında haberleşme için SZD zaman sabiti 4 ve SIA bölme faktörü 16 seçilmiştir. Böylece iki birim arasında haberleşme hızı ,

$$hka = 1.79 \text{ MHz.} / (2*4*2*16) = 6992 \text{ bit/sn.} \quad \text{olur.}$$

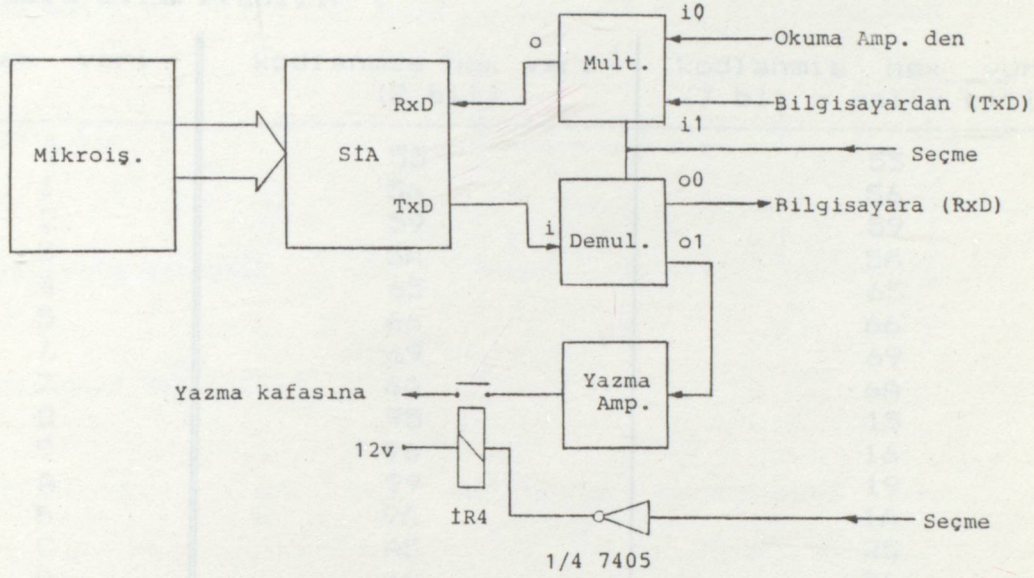
SIA nin seri veri girişinin (RxD) hem okuma amplifika-töründen, hemde bilgisayardan veri okuması gereklidir. Aynı şekilde seri veri çıkışı (TxD), yazma ampli.ne ve bilgisayara veri gönderebilmelidir. RxD girişi, arabirim bilgisayardan komut okurken bilgisayara bağlanmalı, kasetten veri okurken okuma ampli.ne bağlanmalıdır.

Benzer şekilde TxD çıkışı, bilgisayardan okunan verilerin kasete yazılması sırasında yazma ampli.ne, kasetten alınan verilerin bilgisayara gönderilmesi sırasında bilgisayara bağlanmalıdır. Şekil-5.10 daki devre ile bu sorun çözülmüştür.

Devrede seçme girişi,zaten mekanizma kontrolü için çıkış olarak programlanan A iskelesinin 3. bitine bağlanır. Seçme girişinin konumuna göre aşağıdaki bağlantılar sağlanır ;

Seçme Girişi (PA3)	RxD	TxD
0	okuma ampl.	bilgisayara (RxD)
1	bilgisayardan (TxD)	yazma ampl.

Ayrıca seçme girişi "0" olduğu zaman yazma amp.nün çıkışı bir röle yardımıyla yazma kafasından ayrılır. Böylece okuma işlemi sırasında yazma ampli.nün çıkışındaki gerilimin veriyi bozması önlenir.



Sekil-5.10 Veri akış yönlerinin seçilmesi

5.2.3 - Veri Yazma ve Okuma Düzenleri

5.2.3.1 - Veri Kodlama Yöntemi

İlk aşamada verinin yalın şekliyle SIA tarafından eşlik biti ile birlikte yazma amplifikatörüne gönderilmesi düşünülmüştür. Ancak uzun sabit veri dizilerinin okunması sırasında amplifikatörlerde sorunlar ortaya çıkmış, verinin doğru okunması mümkün olmamıştır. Bu nedenle, uzun sabit veri dizilerinin değişken dizilere dönüştürülmesi ihtiyacı doğmuştur. Bu amaçla bir çeşit sayısal evre kaydarmalı anahtarlama yöntemi kullanılmıştır.

Her bir bitlik veri iki bit halinde gönderilir. "0" verileri i0 ve "1" verileri o1 değerleri ile kodlanırlar. Böylece veri dizisi en fazla iki bit süresince aynı seviyede kalabilir. Bu yöntem kullanıldığı zaman bazı kolaylıklar sağladığı görülmüştür. Örneğin, 4 bitlik verilerin kodlanması ile oluşan 8 bitlik paketlerde mutlaka 4 bit "1" ve 4 bit "0" olmak zorundadır. Her bir 8 bitlik paket, ilk 7 bit veri ve son bit eşlik biti olarak yorumlandığında, eşlik bitinin çift eşlik kuralına uyduğu görülür. Böylece her 4 bitlik veri 10 bitlik paket halinde (7 bit kodlanmış veri,

eşlik biti, başlama ve bitiş bitleri) gönderilir. Kodlama işlemini yazılım ile yapmak kolaydır. Bu işlem için hex sayıların kodlanmış karşılıklarını içeren 16 elemanlı bir tablo kullanılır. Şekil-5.11 de 8A_h verisinin kodlanması gösterilmiştir. A_h verisinin kodlanmış karşılığı 99_h olmaktadır. Üst bit eşlik biti olarak yorumlandığı zaman 19_h olacaktır. Benzer işlemler hex sayılar için yapılırsa şöyle bir tablo çıkarılabilir ;

hex veri	kodlanmış hex veri (8 bit)	kodlanmış hex veri (7 bit + eşlik biti)
0	55	55
1	56	56
2	59	59
3	5A	5A
4	65	65
5	66	66
6	69	69
7	6A	6A
8	95	15
9	96	16
A	99	19
B	9A	1A
C	A5	25
D	A6	26
E	A9	29
F	AA	2A

Kodlama programında 8 bitlik verinin önce alt 4 bitini, sonra üst 4 bitini veri olarak göz önüne alınıp tabloda karşılıkları bulunur ve peşpeşe gönderilir. SIA çift eşlik biti üretecek şekilde programlanmalıdır.

8 bitlik verinin iki paket halinde gönderilmesi, veri hızı ve kapasitesinin yarıya düşmesi gibi çok büyük bir sakınca getirmiştir.

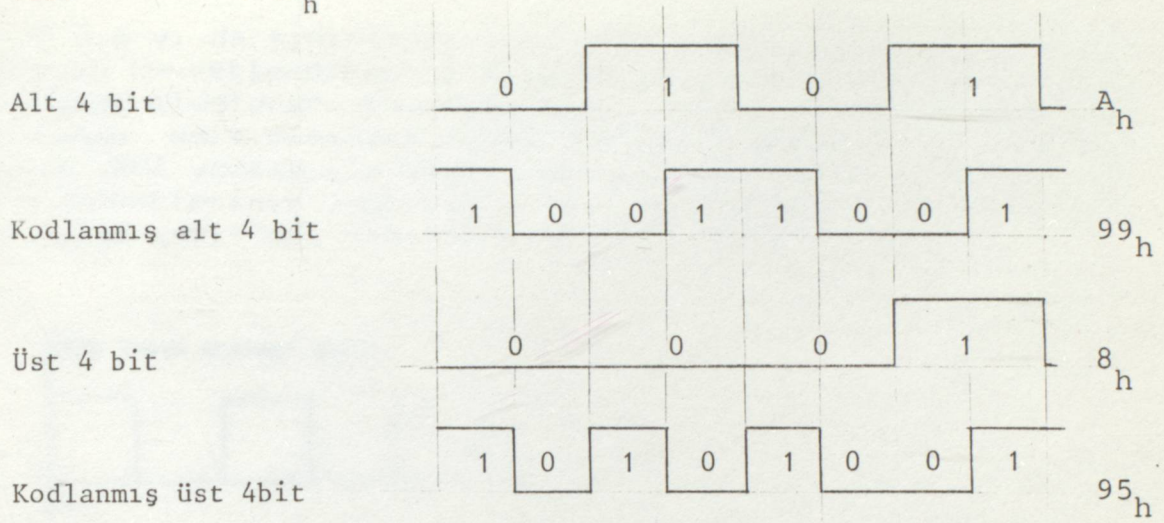
5.2.3.2. - Veri Yazma Devresi ve Yöntemi

Yazma amplifikatörünün tasarımında ilk amaçlanan, alışılmış kaset sistemlerinden daha yüksek veri kayıt hızı, ve güvenliği sağlamaktır. İlave olarak TTL seviyelerinin, FKA yada diğer yöntemlere gerek duymadan doğrudan kaydedilmesi düşünülmüştür.

Devre NRZ yöntemiyle kayıt yapacak şekilde tasarlanmıştır. Yazma kafasının akımı hiçbir zaman sıfır olmaz. Bu durum birbiriyle evrik çalışan iki işlemsel yükseltici ile sağlanır. (Şekil-5.12)

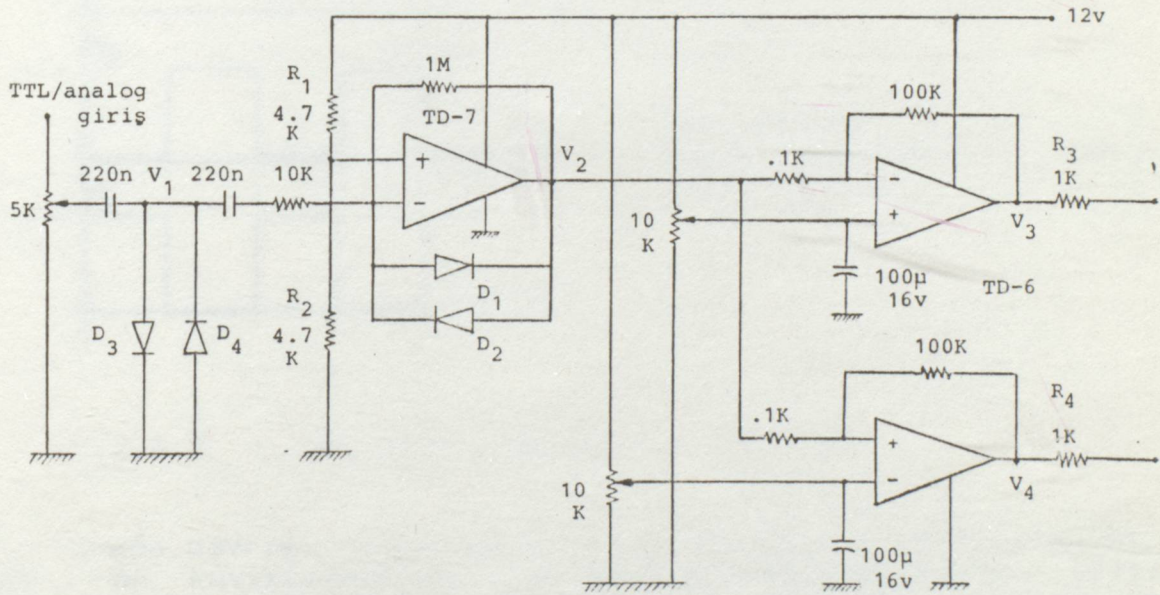
Analog veya TTL giriş işareti d.c. kuplajlanır ve D3, D4 diyotları tarafından ∓ 0.6 v. a kenetlenir. Bu işaret ilk

Veri : 8A_h



Kodlanmış veri: 9599_h

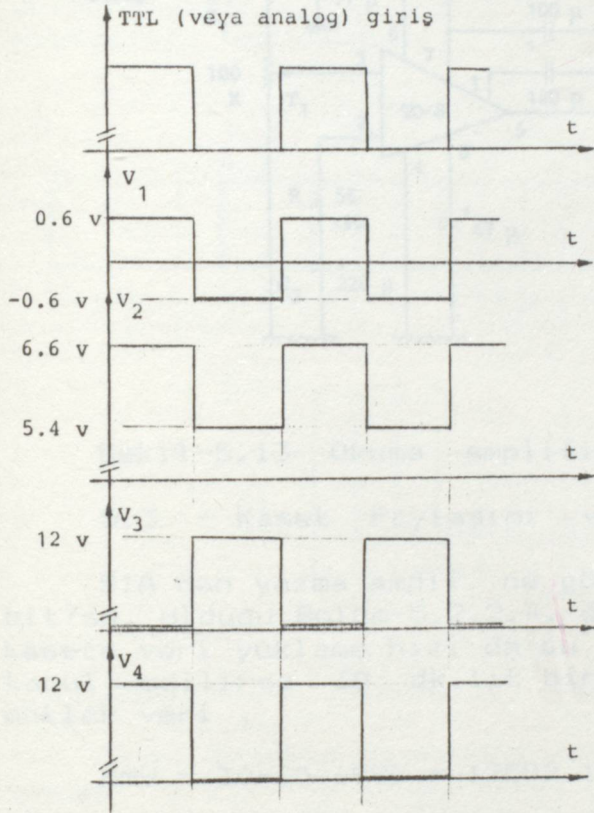
Şekil-5.11 8A hex verisinin kodlanması



Şekil-5.12 Yazma amplifikatörü devre şeması

eviren ampli. de kuvvetlendirilir. R1, R2 dirençleriyle oluşturulan gerilim bölücüleri ile çıkış işareti 6 v. seviyesinde salınır. D1 ve D2 diyotları, çıkış gerilimini

6 v. \mp 0.6 v. da sınırlandırılır. Bu işaret sonraki eviren ve evirmeyen kuvvetlendirici çıkışlarında birbirinin evriği olan iki işaret oluşturur. Kuvvetlendirici çıkış gerilimlerinin hiç bir zaman eşit olmaması nedeniyle kafa akımı sıfır olamaz. Böylece NRZ yöntemiyle kayıt yapılmış olur. Kafa akımı R ve R dirençlerinin değerleri ile orantılıdır. Yazma amplifikatör dalgı şekilleri Şekil-5.13 de verilmiştir.

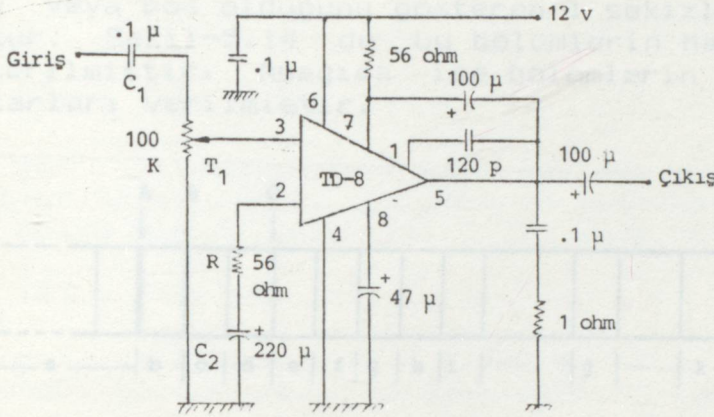


Şekil-5.13
Yazma amplifikatör dalgı şekilleri

5.2.3.3 - Verinin Okunması ve Kontrolü

Okuma devresi olarak oldukça yaygın olarak kullanılan bir ses ön kuvvetlendirici ve güç kuvvetlendiricisi çifti kullanılmıştır. Ön kuvvetlendirici çıkış gerilimi ayarlanarak güç kuvvetlendiricisinin çıkışının doymaya gitmesi ve çıkış geriliminin 0 veya 12 v. olması sağlanır. Bu işaret bir gerilim bölücü ile bölünerek toplayıcı makas (demultiplexer) girişine, dolayısıyla SIA nin RxC girişine uygulanır. Okunan verinin çift eşlik bitine sahip olması gerekir, çünkü SIA 7 bit veri ve çift eşlik biti olarak programlanmıştır. Eger hatalı okuma yapılırsa kaset bölüm başına alınarak tekrar

okuma yapılır. Okuma ampli. devresi Şekil-5.13 de verilmiştir. Okuma kafasından gelen giriş işaretinin d.c. seviyesi C1 tarafından kaldırılır. T1 ile giriş işaretinin genliği ayarlanır. Kazanç R ve C2 değerlerine bağlıdır. Bu ampli. ler iki kanal için tasarlanmış ancak tek kanal için kullanılmıştır.



Şekil-5.13 Okuma amplifikatörü devre şeması

5.3 - Kaset Paylaşımı ve Formatlama

SIA dan yazma ampli. ne gönderilen seri veri hızının 6992 bit/sn. olduğu Bölüm-5.2.2.4. den hatırlanacaktır. Dolayısıyla kasete veri yükleme hızı da bu olacaktır. Bu hız 6990 bit/sn kabul edilirse 30 dk.lık bir kaset yüzüne yüklenecek max. mutlak veri ,

$$MMV = 30 \cdot 60 \cdot 6990 = 12582.10 \quad \text{bit}$$

olacaktır. Her 4 bitlik veri kodlanarak 10 bit olarak gönderildiğinden max. paket sayısı ,

$$MPS = MMV/10 = 1258200 \quad \text{bit}$$

olur ve her karakter (8 bit ASCII kodda) 2 paket olarak yazıldığından max. karakter sayısı ,

$$MKS = MPS/2 = 629100 \quad \text{karakter}$$

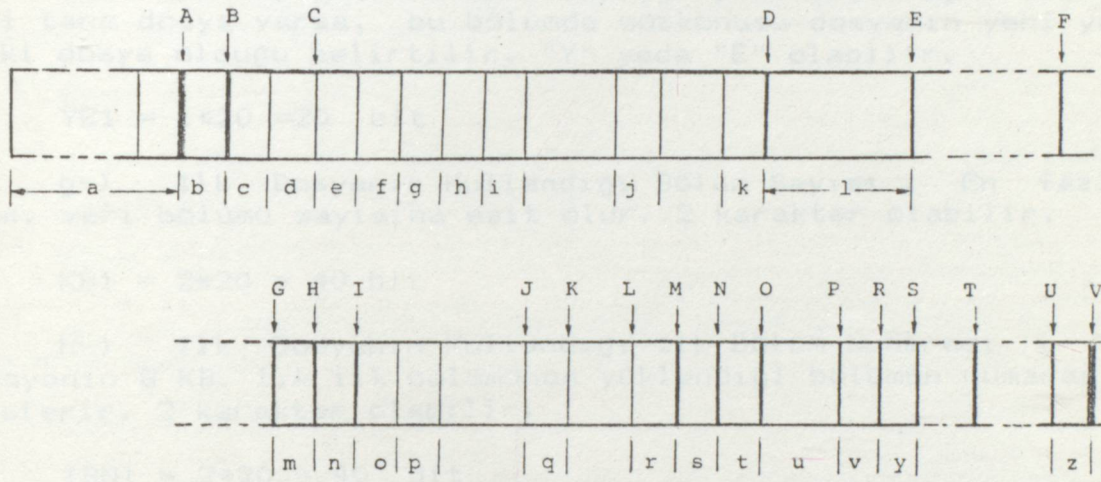
olacaktır. Böylece mutlak veri miktarı ,

$$MVM = MMV/(1024 \cdot 8) = 1535.88 \quad \text{KB.} \quad \text{olur.}$$

Gerçek veri miktarı ise ,

GVM = MKS/1024 = 614.35 KB. olur.

Kaset başlangıcında, kasete ve kasette bulunan dosyalara ait bilgileri içeren bir bölüm ayrılmıştır. Bu bölümde kaset adı, kasette program var-yok bayrağı, dosya adları, dosyaların yüklendiği bölümlerin numaraları ve henüz boş olan bölümlerin numaraları bulunur. Kasette aynı isimde iki dosyanın bulunabileceği düşünülduğünden bunların adlarının yanlarına dolu veya boş olduğunu gösteren 1 sekizlik veri (bayrak) konmuştur. Şekil-5.14 de bu bölümlerin hangi sırada olduğu gösterilmiştir. Aşağıda ise bölümlerin açıklamaları ve veri miktarları verilmiştir.



Şekil-5.14 Miknatıslı şeritin paylaşımı

a-) Kasetin başındaki veri yazılamayacak şerit bölgesini geçmek ve veri yüklenebilecek bölgeye gelmek için başlangıçtan itibaren 5 sn. lik gecikme vardır. Bu gecikme en fazla ,

$$BG = 6990 * 5 = 34950 \text{ bit yer kaplar.}$$

b-) Başlama Formatı ; Kaset üzerinde 16 tane hex sayının dışında veri bulunamaz. Bu zorunluluk kodlamadan dolayı gelir. (Bölüm-5.2.3.1.) Bu sebepten dolayı formatlama verilerinin bu 16 veri dışında seçilmesi emniyetli olur. Başlama formatı, 255 tane BE_h , ve sonra 1 tane CC_h verisinden oluşur. Bu format bir liste yada veri bloğunun başlayacağını gösterir.

$$BF1 = (255+1)*10 = 2560 \text{ bit}$$

c-) Kaset Adı ; En fazla 8 karakterden oluşur. Kullanılan kasetin ismini belirtir. Kullanıma başlamadan önce

istenen kaset olup, olmadığı bu bölgeden kontrol edilir.

$$KA = 8*20 = 160 \text{ bit}$$

d-) Var-Yok Bayrağı ; Kasete yüklenmiş dosya olup olmadığını gösterir. "V" yada "Y" olabilir.

$$VYB = 1*20 = 20 \text{ bit}$$

e-) İlk Dosyanın Adı ; En fazla 8 karakter olabilir.

$$DA1 = 8*20 = 160 \text{ bit}$$

f-) İlk Dosya Yeni-Eski Bayrağı ; Kasette aynı isimde iki tane dosya varsa, bu bölümde sözkonusu dosyanın yeni yada eski dosya olduğu belirtilir. "Y" yada "E" olabilir.

$$YE1 = 1*20 = 20 \text{ bit}$$

g-) İlk Dosyanın Kullandığı Bölüm Sayısı ; En fazla, max. veri bölümü sayısına eşit olur. 2 karakter olabilir.

$$KB1 = 2*20 = 40 \text{ bit}$$

h-) İlk Dosyanın Kullandığı İlk Bölüm Numarası ; İlk dosyanın 8 KB. lık ilk bölümünün yüklendiği bölümün numarasını gösterir. 2 karakter olabilir.

$$IBN1 = 2*20 = 40 \text{ bit}$$

i-) İlk Dosyanın Kullandığı İkinci Bölüm Numarası ; Eger ilk dosya 8 KB. dan büyükse, ikinci 8 KB. lık bölümünün yüklendiği bölüm numarasını verir.

$$IBN2 = 2*20 = 40 \text{ bit}$$

j-) İlk Dosyanın Kullandığı n. Bölüm Numarası ; Eger ilk dosya (n-1)*8 KB. dan büyükse, n. 8 KB.lık bölümün yüklendiği bölüm numarasını verir.

$$IBNN = 2*20 = 40 \text{ bit}$$

k-) İlk Dosyanın Kullandığı Son Bölüm Numarası ; İlk dosyanın kullandığı son bölümün numarasını verir.

$$IBNS = 2*20 = 40 \text{ bit}$$

l-) İkinci Dosyaya Ait Bilgilerin Bulunduğu Bölüm ; e, f, g, h, i, j, k bölümlerine benzer amaçlar için kullanılır. Farkı, 2. dosyaya ait bilgileri içermesidir.

m-) Dosya Bilgilerini Sonlandırma Formatı ; Dosya bilgileri önce 255 tane CC ,sonra 1 tane BB verisi ile sonlandırılır. Bu veriler sonlandırma formatını oluşturur.

$$SF1 = (255+1)*10 = 2560 \text{ bit}$$

n-) Boş Bölümler Yapısı Başlangıç Formatı ; Kaset üzerindeki, henüz veri yüklenmemiş bölüm numaralarını içeren bölümün başlangıcını belirtir. b bölümündeki formatlama tipiyle aynıdır.

$$BF2 = (255+1)*10 = 2560 \text{ bit}$$

o-) İlk Boş Bölümün Numarası ; Kaset üzerindeki ilk boş bölümün numarasını verir.

$$BBN1 = 2*20 = 40 \text{ bit}$$

p-) İkinci Boş Bölümün Numarası ; Kaset üzerindeki ikinci boş bölümün numarasını verir.

$$BBN2 = 2*20 = 40 \text{ bit}$$

q-) n. Boş Bölümün Numarası ; Kaset üzerindeki n. boş bölümün numarasını verir.

$$BBNN = 2*20 = 40 \text{ bit}$$

r-) Son Boş Bölümün Numarası ; Kaset üzerindeki son boş bölümün numarasını verir.

$$BBNS = 2*20 = 40 \text{ bit}$$

s-) Boş Bölümler Yapısı Sonlandırma Formatı ; Bu format m bölümündeki sonlandırma formatıyla aynıdır. Boş bölümler yapısını sonlandırır.

$$SF2 = (255+1)*10 = 2560 \text{ bit}$$

t-) Dosya Verileri Başlangıç Formatı ; Dosya verilerinin bu formatlama verilerinden sonra başlayacağını gösterir. Diğer başlangıç formatlarıyla aynıdır.

$$BF3 = (255+1)*10 = 2560 \text{ bit}$$

u-) İlk Veri Bölümü ; Dosya verilerinin saklandığı ilk bölümdür. 8 KB. kapasitesindedir.

$$VB1 = 8*1024*20 = 16384 \text{ bit}$$

v-) İlk Veri Bölümü Sonlandırma Formatı ; Diğer sonlandırma formatlarıyla aynıdır. İlk veri bölümünü sonlandırır.

$$SF3 = (255+1)*10 = 2560 \text{ bit}$$

y-) İkinci Veri Bölümü Başlangıç Formatı ; Diğer başlangıç formatlarıyla aynıdır. İkinci veri bölümünü başlatır.

$$BF4 = (255+1)*10 = 2560 \text{ bit}$$

z-) Son Veri Bölümü Sonlandırma Formatı ; Son veri bölümünü sonlandırır. Diğer sonlandırma formatlarıyla aynıdır.

$$SFS = (255+1)*10 = 2560 \text{ bit}$$

olarak yazılabilir. Kaset üzerinde işaretlenen noktalar ise ;

- A-) Kaset kullanımı başlangıç noktası
- B-) Kaset başlama formatı başlangıç noktası
- C-) Kasete ait verilerin bitiş noktası
- D-) İlk dosyaya ait verilerin bitiş noktası
- E-) İkinci dosyaya ait verilerin bitiş noktası
- F-) n. dosyaya ait verilerin bitiş noktası
- G-) Son dosyaya ait verilerin bitiş noktası
- H-) Boş bölümler yapısı başlangıç formatının başlangıç noktası
- I-) Boş bölümler yapısı başlangıç noktası
- J-) n. boş bölüm numarası başlangıç noktası
- K-) n. boş bölüm numarası bitiş noktası
- L-) Son boş bölüm numarası başlangıç noktası
- M-) Boş bölümler yapısı sonlandırma formatı başlangıç noktası
- N-) Dosya verileri başlangıç formatının başlangıç noktası
- O-) İlk veri bölümü başlangıç noktası
- P-) İlk veri bölümü bitiş noktası
- R-) İkinci veri bölümü başlama formatının başlangıç noktası
- S-) İkinci veri bölümü başlangıç noktası
- T-) n. veri bölümü başlangıç noktası
- U-) Son veri bölümü bitiş noktası
- V-) Kaset kullanım bitiş noktası

Görüldüğü gibi dosyalara ait verilerin bulunduğu bölüm, boş bölüm numaralarını gösteren bölüm ve tüm veri bölümleri birbirlerinden sonlandırma ve başlama formatlarıyla ayrılırlar. Kaset kullanımının başlama noktası (A noktası) 5 sn. lik gecikme nedeniyle tüm kasetlerde aynı olur. Boş bölümler yapısının başlangıç noktası (H-I noktaları) kasetin

bilinen bir devir sayısı kadar döndürülmesi ile bulunabilir. Aynı durum diğer veri bölümlerinin başlangıç noktaları (0, S, T, ...) için de geçerlidir. Ancak mekanizmadaki devir kaymalarını karşılamak amacıyla bu devir sayıları olması gerekenden birkaç devir sayısı eksik tutulur.

Kullanılabilecek max. veri bölümü sayısını bulmak için ilk olarak N noktasına kadar harcanabilecek max. veri miktarını ile veri bölümleri arasında formatlama için harcanabilecek veri miktarını bulmak gerekir. N noktasına kadar harcanabilecek max. veri miktarı, max. veri bölümü sayısı MBS olmak üzere ;

$$\begin{aligned}MVN &= BG+2*BF+KA+VYB+(DAX+YEX+KBSX+BBN+MBS*BNS)*MBS+2*SF \\MVN &= 34950+4*2560+160+20+(160+20+40+40+MBS*40)*MBS \\MVB &= 34950+10240+180+(260+MBS*40) * MBS \\MVB &= 45370 + 260*MBS +40*(MBS**2)\end{aligned}$$

olur. Veri bölümleri arasındaki formatlama verilerinin harcanabileceği veri miktarı ise ;

$$\begin{aligned}FVM &= (BFX+SFx)*MBS \\FVM &= (2560+2560)*MBS = 5120*MBS\end{aligned}$$

olacaktır. Böylece veri bölümleri dışındaki bölümler için harcanan toplam veri miktarı ;

$$TVM = MVN+FVM = 45370+5380*MBS+40*(MBS**2)$$

bulunur. Sistemde 2000 - 3FFF adreslerine yerleştirilen (Bölüm-5.2.1.1) 8 KB. lık veri tamponu nedeniyle, program verilerinin saklanması için gerçek veri miktarı 8 KB. lık bölümlere ayrılırsa, TVM gözönüne alınmadan , max. 8 KB.lık bölüm sayısının ;

$$MBS = 614.35 / 8 = 76.79$$

olabileceği görülür. 72 bölüm kullanılması ve artan yerlerin veri bölümleri dışındaki bölümlere ayrılması düşünülürse,

$$\begin{aligned}TVM &= 45370+5380*72+40*(72**2) \\TVM &= 45370+387360+207360 = 640090 \text{ bit} \\TVM &= 640090 / 10 = 64009 \text{ paket} \\TVM &= 64009 / (2*1024) = 31.25 \text{ KB.}\end{aligned}$$

olacaktır. Böylece kasetin bir yüzüne en fazla 72*8 KB. = 576 KB. lık veri yükleme olanağı vardır. Kasette,

$$\begin{aligned}GVM - TVM - 576 \text{ KB.} &= 614.35 \text{ KB} - 31,25 \text{ KB.} - 576 \text{ KB.} \\&= 7.1 \text{ KB.}\end{aligned}$$

kullanılmayan veri bölgesi kalacaktır. Bir veri bölümünün okunma süresi, $t_{vb} = (8 \times 1024 \times 20) / 6690 = 24.5$ sn. süre alır. Örnek olarak, kaset ortasında yer alan 8 KB. lık bir bölüm 30 sn. (erişim süresi) + 24.5 sn. = 55 sn. de okunabilir.

5.4. - İşletim Sistemi

Sistemin kullanımına başlamadan önce, kullanıcı bilgi sayara, arabirimle iletişimi sağlayan programı yüklemek gereklidir. Çünkü "kaset işletim sistemi"nin tüm komutları, kullanıcı bilgisayar tarafından arabirime gönderilir. Arabirim gönderilen bu komutları değerlendirerek ilgili programı yürütür.

Arabirim açıldığında SIA bilgisayarla haberleşmeye şartlanarak (9600 baud) kullanıcıya "HAZIRIM" mesajı verilir ve komut beklenir. Komut okuma altprogramında bilgisayardan okunan komutta hata varsa "PE/FE * TEKRARLAYINIZ" mesajı verilir. Kullanıcı komutu tekrarlar. Komut hatasız alınmışsa komut ve kaset/dosya adı INSBUF da saklanır. Sonra hangi komut olduğu bulunur ve ilgili programa dallerilir.

5.4.1 - B"Kaset Adı" (Başla) Komutu

Kullanıma bu komutla başlamak zorunludur. Bu komut alınınca PIA B iskelesi giriş olarak programlanır. Kaset varlık bilgisi okunur. Mekanizmada kaset yoksa "KASET YOK" mesajı verilir. Mekanizmaya kaset konmadan diğer işlemlere geçilmez. Kasetin bulunduğu sezildiğinde PIA A iskelesinden sola sarma verisi atılarak kaset başa alınır. SIA 6992 bit/sn. için şartlandırılır. Motor çalma verisi PIA A iskelesinden atılarak okuma başlatılır. 5 sn. lik gecikme ile A noktasına gelinir. Bu noktaya gelindiğinde kaset konum belirleme görevini yapan SZD 0. kanal şartlandırılarak devir sayma işlemi başlatılır.

Kasetten önce liste yapısı ve sonra boş bölümler yapısı SIA tarafından okunur ve LISBUF, EPABUF bölgelerinde saklanır. Okuma işlemi sırasında aynı zamanda verinin kod çözülmesi de yapılır. Liste yapısının ve boş bölümler yapısının bittiği, sonlandırma formatlarından anlaşılır. Kasete ait verilerin uzunluğu ,

$$KVU = KA + VYB = 8 + 1 = 9 \text{ sekizli,}$$

liste yapısı uzunluğu ,

$$\begin{aligned} LYU &= (DAX + YEX + KBSX + MBS * ENX) * MBS \\ &= (8 + 1 + 2 + 72 * 2) * 72 = 11160 \text{ sekizli,} \end{aligned}$$

boş bölümler yapısı uzunluğu ,

$$BBU = BBN * MBS = 2 * 72 = 144 \text{ sekizli}$$

olacaktır. Böylece bilgisayarın göndereceği verinin uzunluğu,

$$KVU+LYU+BBU = 9+11160+144 = 11313 \text{ sekizli}$$

olur. Bu blok 8 KB. dan fazla olduğundan artan bölümü D800 adresinden başlayan DYB bölgesine yüklenir.

Eğer okuma işlemi sırasında hatalı veri alınırsa kaset tekrar başa alınarak yeniden okuma yapılır. INSBUF ile LISBUF bölgelerindeki kaset adları karşılaştırılır. Kullanılan kaset istenen kaset değilse veya kaset boş ise uyarı verilerek, istenen kaset ise "İŞLEM TAMAMLANDI" mesajı verilerek yeni komut beklenir.

5.4.2 - L (Listele) Komutu ;

SiA bilgisayarla haberleşme için (9600 baud) şartlan dırılır. LISBUF bayrağına bakılarak liste yapısının kasetten LISBUF bölgesine yüklenip, yüklenmediği anlaşılır. Eğer yüklenmemişse "HATALI BAŞLAMA" mesajı verilir. Yüklenmişse, LISBUF bölgesindeki liste yapısı bilgisayara gönderilir. Bilgisayar bu liste yapısını listeler. Kaset yalnızca formatlanmış ve hiçbir dosya yüklenmemişse dosya var-yok bayrağından anlaşılabilceğinden ($\neq 11_h$) kullanıcıya mesaj yazılır. Bilgisayar bu komutu gönderdikten sonra veri okuma konumunda beklemelidir.

5.4.3 - O"Dosya Adı" (Oku) Komutu

LISBUF bayrağı yardımıyla LISBUF bölgesinde liste yapısının bulunup bulunmadığına bakılır. Eğer liste yapısı varsa dosya adı liste yapısından bulunur. Dosyanın kullandığı bölüm sayısı ve bölüm numaraları başka bir bölgeye aktarılır. Her bölüm okunduktan sonra kullanılan bölüm sayısı eksiltilir. İlk bölüm numarasına göre bölümün yeri bir tablodan bulunarak, mekanizmanın bulunduğu konuma göre sağa veya sola sarma komutu PIA A iskelesinden atılır. Bu bölüme gelindiği zaman çalma konumuna geçilir. Bölüm okunarak LISBUF bölgesine yerleştirilir. Sonra SiA 9600 baud için şartlandırılır. Dosya verileri bilgisayara gönderilir. İkinci bölüm numarası okunur ve bilgisayara gönderilir. Dosyanın kullandığı tüm bölümler okununcaya kadar bu işlem devam eder. Her bölümün okunması sırasında hata oluşursa bölümün başına gidilerek bölüm yeniden okunur.

Bilgisayar komut gönderdikten sonra seri veri okuma konumuna geçmeli ve okuduğu verileri bellekte saklamalıdır.

5.4.4 - Y"Dosya Adı" (Yaz) Komutu

Liste yapısı bilgisayarda saklanmaktadır. Yeni bir dosya yazılacağı zaman, dosya ismi, yeni-eski bayrağı (eğer aynı

isimde başka dosya varsa "Y", eski dosyanın yeni-eski bayrağı "E" yapılmalıdır), dosya uzunluğundan kullanacağı bölüm sayısı, boş bölümler yapısından yüklenecek bölüm numaraları düzenlenir. Liste ve boş bölüm yapıları bu şekilde düzenlendikten sonra, bu komut arabirime gönderilir ve bekleme konumuna geçilir. Arabirim B iskelesini okuyarak yazma koruması olup olmadığına bakar. Yazma koruması varsa uyarı verir, yoksa kaseti başa alır. 5 sn. lik gecikme ile yine A noktasına gelinir. Kullanıcıya yazma işlemi için nazır olduğu bildirilir.

Kullanıcının yapacağı bir işlemle (örneğin, basılacak bir tuşla) kaset, liste yapısı ve boş bölümler yapısına ait düzenlenmiş veriler arabirime gönderilir. Bu veriler LISBUF bölgesinde ve D800 adresinden başlayan OYB bölgesinde saklanır. Arabirim A noktasından başlayarak başlama formatını yazar. Sonra kasete ait bilgiler ve liste yapısı yazılır. Liste yapısı sonlandırma formatı ve boş bölümler başlangıç formatından sonra boş bölümler yapısı ve boş bölümler sonlandırma formatı yüklenir. LISBUF bölgesinden dosyanın kullandığı bölüm sayısı ve bölüm numaraları başka bir bellek bölgesine taşınır. İlk bölüm numarasına göre bölümün başlangıcına gidilir. Başlama formatı yazılır. Bilgisayardan 8 KB.lık bölüm verisi alınarak LISBUF bölgesine, daha sonra sözkonusu bölüme yüklenir. Bölüm sonlandırma formatı yazılır. İkinci bölüm numarasına göre bölümün yeri bulunarak başlangıcına gidilir. Bu bölümün başlangıç formatı yazılır. Aynı şekilde bilgisayardan 8 KB. lik ikinci bölüm verisi alınarak LISBUF bölgesine, buradanda ilgili bölüme yüklenir. Dosyanın kullandığı tüm bölümler bitinceye kadar bu olay devam eder. İşlem tamamlandı mesajı verilir ve yeni komut beklenir.

5.4.5 - S"Dosya Adı" (Sil) Komutu

Liste yapısı bilgisayarda saklanmaktadır. Silinecek dosyanın liste yapısındaki yeri bulunur. Kullandığı bölümlerin numaraları boş bölümler yapısına eklenir. Böylece bu bölümler daha sonraki yazma işlemlerinde kullanılabilir.

Liste yapısındaki silinecek dosya ile ilgili bölümlere FF_h verisi yerleştirilir. Liste yapısında kullanılmayan tüm bölgelerde bu veri bulunmak zorundadır.

Eğer silinmesi gereken birden fazla dosya varsa her seferinde bu komut kullanılarak liste yapısı ve boş bölümler yapısı değiştirilir ancak bu komutla kaset üzerindeki verilerde değişiklik olmaz. Bu değişikliği yapmak için T (düzenleme) komutu kullanılmalıdır.

5.4.6 - T (Düzenle) Komutu

Bu komut S"Dosya Adı" komutu kullanıldığı zaman liste yapısı ve boş bölümler yapısında oluşan değişiklikleri kaset üzerine kaydetmek için kullanılır.

Komut alındığında arabirim veri okuma konumuna geçer. Mevcut liste yapısı ve boş bölümler yapısı bilgisayardan okunarak LISBUF ve D800 adresinden başlayan OYB bölgesinde saklanır. 5 sn. lik gecikme ile A noktasına gelinir. Başlama formatı yazılır. Kaset verileri, liste yapısı yüklenir. Liste yapısı sonlandırma formatı ve boş bölümler yapısı başlangıç formatı yüklendikten sonra boş bölümler yapısı yazılır. Boş bölümler yapısı sonlandırma formatı da yüklenerek işlem tamamlandı mesajı verilir.

5.4.7 - K"Dosya Adı" (Kopye) Komutu

Bu komutun yürütülmesi iki aşamada olur. İlk olarak O komutunda olduğu gibi dosya, kasetten bilgisayara okunur. Sonra Yaz komutu ile dosya kaset üzerine yazılır.

Eğer bir kasetteki dosya başka bir kasete kopya edilecekse ilk olarak kopye edilecek dosyanın bulunduğu kaset mekanizmaya konmalıdır. Bunun için kullanıcıya uyarı verilir. Kasetin mekanizmaya konduğu sezilince kaset başa alınır. Başla komutunda olduğu gibi kasete ait veriler, liste yapısı ve boş bölümler yapısı okunur. O komutuna ait program yürütülerek dosya verilerini içeren bölümler tek tek okunarak bilgisayara gönderilir. Sonra kasetin değiştirilmesi için kullanıcıya uyarı verilerek beklenir. Dosyanın kopye edileceği kasetin, mekanizmaya konduğu kullanıcı tarafından arabirime bildirilir. Kaset başa alınır. Liste yapısı ve boş bölümler yapısı okunur ve bilgisayara gönderilir. Kopye edilecek dosyanın adı, yeni- eski bayrağı, dosya uzunluğundan kullandığı bölüm sayısı, boş bölüm yapısından kullandığı bölüm numaraları belirlenir. Bunlara göre liste yapısı ve boş bölümler yapısı yeniden düzenlenir. Sonra Y komutuna ait program yürütülerek kaset verileri, liste yapısı, boş bölümler yapısı kasete yüklenir.

5.4.8 - F"Kaset Adı" (Formatla) Komutu

İlk olarak kullanılan bir kaset formatlanmadan kullanılamaz. Bunun için kaset mekanizmaya konur. Bu komut arabirime gönderilir. Kasetin mekanizmaya konduğu sezildiği zaman kaset başa alınır. PIA A iskelesinden calma verisi atılır. 5 sn. lik gecikme programı ile A noktasına gelinmesi beklenir. SFA 6992 bit/sn. için şartlandırılır. Başlama formatı yazılır. Kaset adı ve dosya var-yok bayrağı (59_h - ASCII "Y") ve liste yapısı (FF_h olarak) yüklenir. Liste yapısı sonlandırma formatı ve arkasından boş bölümler yapısı başlangıç formatı yazılır. Boş bölümler yapısı (FF_h olarak) yazıldıktan sonra sonlandırma formatı da yüklenerek işlem tamamlandı mesajı verilir.

Başlangıç formatı 255 tane BB_h ve sonra 1 tane CC_h verisinden, sonlandırma formatı ise 255 tane CC_h ve sonra 1 tane BB_h verisinden oluşur.

5.5 - Sistem Yazılımı

5.5.1 - Arabirim Yazılımı

```
      ; ***** İŞLETİM SİSTEMİ PROGRAMLARI *****  
  
      ORG  B000H      ; Programlar B000H adresinden  
                      ; başlar.  
BEGIN   : XOR A  
          LD (LISBUF),A ; LISBUF sıfırlanır.  
          CALL INFCOM   ; 9600 baud için SZD ve SiA  
                      ; şartlandırma  
          LD HL,READYM  ; Hazır mesajı  
L11     : LD D,50H      ; Gösterge katsayısı  
          CALL PRTMES   ; Mesaj yazma altprogr.  
L4      : PUSH DE  
          CALL SCAN1    ; Mesaj tekrarlanır.  
          POP DE  
          DEC D  
          JR NZ,L4  
L5      : CALL READ     ; Komut okuma altprog.  
JMP4    : JR Z,NOERR   ; Okumada hata var mı?  
          LD HL,ERRMS  
          JR L11  
NOERR   : LD A,(INSBUF) ; Okumada hata yok.  
          CP "B"       ; Başla komutumu ?  
          JP Z,START  
          CP "D"       ; Oku komutumu ?  
          JP Z,READDA  ; Listele komutumu ?  
          CP "L"       ; Yaz komutumu ?  
          CP "Y"       ; Sil komutumu ?  
          CP "S"       ; Kopya et komutumu ?  
          CP "P"       ; Düzenle komutumu ?  
          CP "T"       ;  
          CP "F"       ;  
          JP Z,FORMAT  
READ3   : LD HL,ERRMS1 ; Hata mesajı  
          CALL PRTMES  
          CALL SCAN1  
          LD HL,INSBUF  
          LD B,08H  
          IN A,(URTCNT)  
          BIT 1,A      ; Komut okundumu ?
```



```
JR NZ,READ3
CALL READ2      ; SIA dan veri okunur
JR JMP4

; ***** Başla Komutu Programı *****

START : LD A,4FH      ; B iskelesi giriş olarak pro.
        OUT (PBCONT),A
L1 : IN A,(PBDATA)   ; B iskelesinden okuma
        BIT 0,A
        JR NZ,JMP1   ; Kaset kontrolü
        LD HL,NOCAMS ; Kaset yok mesajı
        CALL PRTMES
        CALL SCAN1   ; Mesaj yazılır
        JR L1
JMP1 : CALL CASBEG   ; Kaset başa alınır.
        LD A,0EH     ; Motor durur,seçme g. aktif
        OUT (PADATA),A
        LD HL,CURTI2 ; 6992 bit/sn. için SZD ve
        ; SIA şartlandırma
        CALL INFCAS
        LD A,05H
        OUT (PADATA),A ; Motor çalma konumunda
        CALL DEL5SE   ; 5 sn. gecikme
        CALL COUNTI   ; Sayıcı şartlandırma
        LD HL,LISBUF
L12 : CALL RECAS     ; Kasetten veri okuma altpr.
        JR Z,END      ; Okuma hatasız tamamlandı mı ?
        CALL CASBEG   ; Hatalı okuma, kaset başa
        ; alınır
        POP HL
        JR L12
END : POP IX         ; Yığın düzeltilir
        LD (LISEND),HL ; Liste yapısı sonu saklanır
        LD HL,EPABUF  ; Boş bölümler liste yapısı
        ; tampon adresi
END1 : CALL RECAS    ; Boş bölümler liste yapısı
        JR Z,ENDSTA  ; hatasız okundumu ?
        CALL CASBEG  ; Okumada hata var
        POP HL       ; Tampon adresi çekilir
        JR END1
ENDSTA : XOR A
        OUT (PADATA),A ; Motor durur
        LD (LISFLS),A  ; Liste yapısı geçerli
        CALL INFCOM   ; 9600 baud için şartlandırma
        LD A,(LISBUF+9)
        CP 11H        ; Kasette dosya varmı ?
        JR NZ,EMPMS   ; Kaset boş mesajı için
        LD IY,INSBUF+2 ; Dosya var
        LD IX,LISBUF
```



```
CALL COMP ; istenen kasetmi ?
NMBREAD JR Z,JMP5
LD HL,DIFCAS ; istenen kaset degil
JR L14
JMP5 : LD A,03H
OUT (PADATA),A ; Motor sarma konumunda
L18 : LD A,(TUR1)
CP 02H ; 2 devir oldumu ?
JR NZ,L18
OKMSG : LD HL, OKEYMS ; istenen kaset ve islem tamam
; landi mesaji
L14 : CALL PRMES
L13 : CALL SCAN1
IN A,(URTCNT) ; Yeni komut beklenir
BIT 1,A
JR Z,L13
LD HL,INSBUF ; Yeni komut alindi
LD B,0BH
CALL READ2 ; Komut okunur
JR JMP4
EMPMS : LD HL,EMPMSG
JR L14
; ***** Oku Komutu Programi *****
READDA : XOR A
LD (SECCOU),A ; Bölüm sayacı sıfırlanır
IN A,(PBDATA)
BIT 0,A ; Kaset varmı ?
JR NZ,REJMP1
LD HL,NOCAMS ; Kaset yok mesaji
JR L14
REJMP1 : LD A,LISFLG ; Liste yapısı geçerlimi ?
CP 00
JR Z,REJMP2
LD HL,REBEGM ; Yeniden başla mesaji
JR L14
REJMP2 : CALL NASEAR ; Kaset ismi aranır
JR Z,FOUND
LD HL,NOFIMS
JR L14
FOUND : CALL CONVR1 ; Okunacak bölüm sayısı bulunur
; ve bölüm numaraları saklanır
LD DE,SECCOU+2 ; Bölüm numarası için
REJMP4 : CALL CONVR2 ; Bölüm adresi bulunur
REJMP3 : CALL LOCFIN ; Bölüm yeri bulunur
LD HL,LISBUF
CALL RECAS ; Kasetten veri okunur
JR Z,NBREAD ; Hatalı okuma varmı ?
```



```
POP DE ; Bölüm numarası çekilir
JR REJMP3 ; Bölüm yeniden okunur
NBREAD : XOR A
        OUT (PADATA),A ; Motor durur, bilgisayara
        ; veri gönderme konumu
        CPL
        LD (LISFLB),A ; Liste yapısı geçersiz
        CALL OUTCOM ; Veriler gönderilir
        LD HL,SECCOU
        DEC (HL) ; Bölüm sayacı eksiltilir
        JR Z,OKMSG ; Bölümler tükendimi ?
        LD DE,(SECCOU+1) ; Yeni bölüm numarası için
        JR REJMP4 ; Yeni bölüm okunur

; ***** Yaz Komutu Programı *****

WRITE : XOR A
        LD (SECCOU),A ; Bölüm sayacı sıfırlanır
        LD A,(EPAFLB)
        CP 44H ; Kaset dolumu ?
        JR NZ,JMP11
        LD HL,CAFUMS ; Kaset dolu mesajı
        JR L14 ; Yeni komut beklenir
JMP11 : IN A,(PBDATA)
        BIT 1,A ; Yazma koruması varmı ?
        JR NZ,JMP12
        LD HL,WPMSG ; Yazma koruması mesajı
JMP12 : JR L14
        CALL INFCOM ; Bigisayarla haberleşme için
        LD HL,LISBUF ; Liste yapısını okumak için
WRTJ5 : LD B,01
        CALL READ1 ; 1 karakter okunur
        DEC HL
        LD A,(HL)
        CP 0CCH ; Liste yapısı sonumu ?
        JR NZ,WRTJ5
        CALL NASEAR ; Dosya ismi aranır
        CALL CONVR1 ; Kullanılan bölüm sayısı
        ; bulunur, bölüm num. saklanır
        CALL CASBEG ; Kaset başa alınır
        LD HL,CURT12
        CALL INFCAS ; SIA ve SZD şartlandırma
        LD A,0DH ; Motor çalma konumunda, seçme
        ; ucu aktif
        OUT (PADATA),A
        CALL DELSSE ; 5 SN. gecikme
        CALL FORBEG ; Başlangıç formatı
        LD HL,LISBUF
        LD BC,LISLEN
```



```
CALL REPT ; Liste yapısı verileri yazılır
CALL FOREND ; Sonlandırma formatı
CALL FORBEG ; Başlangıç formatı
LD HL,EPABUF
LD BC,72
CALL REPT ; Boş bölümler yapısı yazılır
CALL FOREND ; Sonlandırma formatı
LD DE,SECCOU+2
WRTJB : CALL CNVR2 ; Bölüm numarası ve adr.bulunur
CALL REFORM ; Başlangıç formatı okunur
POP IX ; Yığın düzeltilir
LD A,B
OUT (PADATA),A ; Motor durur, seçme ucu aktif
LD A,OFFH
LD (LISFLB),A ; Liste yapısı geçersiz
CALL INFCOM ; Bilgisayarla haberleşme için
LD HL,CONMSG
WRTJ7 : CALL PRMES ; Veri isteği mesajı
WRTJ6 : CALL SCAN1
IN A,(URCNT) ; Veri okundumu ?
BIT 1,A
JR Z,WRTJ6
LD HL,LISBUF ; Veri okundu
LD C,1FH
WRTJ5 : LD B,OFFH ; 8 KB. lık veri okunur
CALL READ1
JR Z,WRTJ4
LD HL,REDAMS ; Hatalı okuma var
JR WRTJ7 ; Verinin tekrarı için
WRTJ4 : DEC C
JR NZ,WRTJ5
LD HL,CURT12
CALL INFCAS ; Yükleme için şartlandırma
LD HL,LISBUF
LD BC,1FFFH ; 8 KB. veri yazılır
CALL REPT ; Veri kodlanır ve yazılır
CALL FOREND ; Sonlandırma formatı yazılır
LD HL,SECCOU
DEC (HL) ; Bölüm sayısı eksiltilir
JR Z,OKMSG ; Bölümler tükendimi ?
LD DE,(SECCOU+1)
JR WRTJB ; Diğer bölümler için
; ***** Listeleme Komutu Programı *****
LIST : CALL INFCOM ; 9600 baud için şartlandırma
LD A,(LISFLB) ; Liste yapısı bayrağı
CP OFFH ; Liste yapısı geçerlimi ?
JR Z,INSERR ; Geçersizse hata mesajı ver
LD IX,LISBUF
```



```

; ***** Altprogramlar *****
; INFCOM - 9600 baud için SIA ve SZD şartlandırma
; INFCAS - 4992 bit/sn. için şartlandırma
INFCOM ; LD HL,CURTI1
INFCAS ; LD B,6
INIT ; LD C,(HL)
; INC HL
; OUTI
; JR NZ,INIT
; RET

```

```

; DEL5SE - 5 sn. başlangıç gecikmesi
DEL5SE ; LD BC,OFFFHH ; ilk zaman sabiti
; LD D,8 ; ikinci zaman sabiti
L16 ; DEC C
; JR NZ,L16
; DEC B
; JR NZ,L16
; DEC D
; JR NZ,L16
; RET ; Gecikme biter

```

00

```

; CASBEG - Kaseti başa alır
CASBEG ; XOR A
; LD (TUR2),A ; Tur sayacı 1 sıfırlanır
; INC A
; LD (TUR1),A ; Tur sayacı 2 = 1
; LD HL,CPAINI
; LD B,5
; CALL INIT ; SZD ve FIA şartlandırma
; LD HL,WAITMS ; Bekle mesajı
; CALL PRMES
L2 ; CALL SCAN1
; IN A,(CCONTO) ; Sayıcı değeri okunur
; LD B,A ; B de saklandı
; LD DE,OFFFHH ; Gecikme başlangıcı
L3 ; DEC E
; JR NZ,L3
; DEC D
; JR NZ,L3
; IN A,(CCONTO) ; Sayıcı değeri tekrar okundu
; CP B
; JR NZ,L2 ; Eski değerle aynı mı ?
; RET ; Eski değerle aynı değil
; kaset başa alındı

```



```
CONVR1      ; CONVR1 - Kullanılan bölüm sayısını bulur ve
            ; bölüm numaralarını saklar
CONVR1      : INC HL                ; Kullanılan bölüm sayısı
            ; alt dörtliğe için
            LD A,(HL)
            SUB 30H
            LD B,A
            INC HL
            LD A,(HL)              ; Kullanılan bölüm sayısı
            SUB 30H                ; üst dörtliğe için
            SLA A
            SLA A
            SLA A
            SLA A                  ; ASCII-onaltılı dönüşüm
            ADD A,B
            LD (SECCOU),A          ; Kullanılan bölüm sayısı
            ; saklanır
            LD DE,SECCOU+2
            LD BC,144              ; Kullanılan bölüm numaraları
            LDIR                  ; saklanır
            RET

CONVR2      ; CONVR2 - Bölüm numarasını ve adresini bulur
CONVR2      : LD A,(DE)            ; Bölüm numarası için
            SUB 30H                ; ASCII-onaltılı dönüşüm
            LD B,A                  ; Alt dörtlük B saklayıcısında
            INC DE
            LD (SECCOU+1),DE       ; DE saklanır
            LD A,(DE)              ; Üst dörtlük için
            SUB 30H
            SLA A
            SLA A
            SLA A
            SLA A                  ; Üst dörtlük A da
            ADD A,B                ; Bölüm numarası A da
            LD E,A
            LD D,PARTAB            ; Bölüm yerleri tablosu
            SLA E                  ; Bölüm yeri DE saklayıcısında
            PUSH DE

REFORM      ; REFORM - Bölüm Başlangıç formatı okur
REFORM      : CALL LOCFIN          ; Bölümün yeri için
            LD A,05H
            OUT (PADATA),A         ; Çalma konumu
            LD HL,CURT12
            CALL INFCAS            ; Kasetten veri okuma konumu
FORM1       : LD B,01              ; Bir karakter okuma için
            LD HL,WRTBUF
            CALL READ1             ; Başlangıç formatı için
```



```

; JR Z,FORM2 ; Veri hatasız okundumu ?
; POP DE
READ ; JR REFORM
FORM2 : DEC HL
READ1 : LD A,(HL) ; Okunan veri A da
; CP OBBH ; ilk formatlama verisini ?
; JR Z,FORM1
READ2 : CP OCCH ; Son formatlama verisini ?
; JR NZ,FORM1
; RET

; COUNTI - Konum bilgisi için SZDO şartlandırma
COUNTI : DI ; MIB kesme izinsiz
; XDR A
; OUT (CCONTO),A ; Kesme vektörü
; LD A,7FH ; SZD kesme izinsiz, zaman
; OUT (CCONTO),A ; sabiti bekler
; LD A,0AH
; OUT (CCONTO),A ; Zaman sabiti 10
; LD A,INTVEH ; Kesme vektörü üst sekizlisi
; LD I,A
; IM 2 ; Kesme modu 2
; LD A,0F9H
; OUT (CCONTO),A ; SZD kesme izni, normal
; ; çalışma verisi
; EI ; MIB kesme izni
; RET

; FORBEG - Bölüm başlangıç formatını yükler
; FOREND - Bölüm sonlandırma formatını yükler
FORBEG : LD B,OFFH ; 255 tane veri yüklenecek
WAIT1 : IN A,(URTCNT)
; BIT 0,A ; SIA hazır mı ?
; JR NZ,WAIT1
; LD A,OBBH ; ilk formatlama verisi
; OUT (URTD A),A ; Veri atılır
; DJNZ WAIT1
; LD A,OCCH ; ikinci formatlama verisi
; JR FOR1

FOREND : LD B,OFFH
WAIT2 : IN A,(URTCNT)
; BIT 0,A
; JR NZ,WAIT2
; LD A,OCCH ; ilk formatlama verisini ?
; OUT (URTD A),A ; Veri atılır
; DJNZ WAIT2 ; 255 tane veri atılır
; LD A,OBBH ; Son formatlama verisi
FOR1 : OUT (URTD A),A
; RET
```



```

; READ - Bilgisayardan komut okur
; READ1 - Bilgisayardan veya kasetten veri okur
READ      : LD HL,INSBUF
          : LD B,0BH           ; 11 karakter okunur
READ1     : IN A,(URTCNT)
          : BIT 1,A
          : JR Z,READ1       ; Veri okundumu ?
READ2     : BIT 3,A
          : RET NZ           ; FE varmi ?
          : BIT 4,A
          : RET NZ           ; FE varmi ?
          : BIT 5,A
          : RET NZ           ; DE varmi ?
          : IN A,(URTDATA)   ; Veri okunur
          : LD (HL),A
          : INC HL
          : DJNZ READ1      ; Tüm veriler okundumu ?
          : RET

; COMP - Kaset yada dosya isimlerini karşılaştırır
; LD B,S           ; S karakter karşılaştırılır
COMP      : LD A,(IY)
L12       : CP (IX)
          : RET NZ           ; Aynı değilse Z=1
          : INC IX
          : INC IY
          : DJNZ L12
          : RET             ; Aynı ise Z=0

; OUTCOM - Bilgisayara seri veri çıkışı için
; CALL INFCOM     ; 9600 baud için şartlandırma
OUTCOM    : LD HL,LISBUF
          : LD C,URTDATA    ; Çıkış adresi
WAIT1     : IN A,(URTCNT)
          : BIT 0,A         ; SIA hazır mı ?
          : JR Z,WAIT1
          : LD A,OCCH
          : CP (HL)         ; Son verimi ?
          : RETZ            ; Son veri ise döngü
          : OUTI
          : JR WAIT1

; LOCFIN - Bölüm yerini bulur
; INC DE           ; Bölüm adresi DE de
; LD IX,TUR2      ; Kaset konumu için
LOCFIN    : LD A,(DE)
L17       : CP (IX)
```



```

                                JP PE,REPT1
                                RET
CHRWR1   : LD E,A                ; Veriye göre tablo adresi
WAIT3    : IN A,(URTCNT)         ; alt sekizlisi E de
                                BIT 0,A                ; SIA hazır mı ?
                                JR NZ,WAIT3
                                LD A,(DE)             ; Kodlanmış veri A da
                                OUT (URTDA),A         ; Veri atılır
                                RET

REPT3    : REPT3 - Formatlama için liste yapısını yükler
          : IN A,(URTCNT)
          BIT 0,A                ; SIA hazır mı ?
          JR NZ,RET3
          LD A,2AH               ; F verisinin kodlanmış karşılı.
          OUT (URTDA),A
          CPI                    ; BC sıfır oldumu ?
          JP PE,REPT3
          RET

RECAS    : RECAS - Kasetten veri okur ve kod çözer
L6        : PUSH HL              ; Tampon adresi saklanır
          : LD B,1               ; 1 karakter okunacak
          CALL READ1
          RET NZ                 ; Hatalı okunmuşsa geri döner
          DEC HL
          LD A,(HL)             ; Okunan veri A da
          CP 0BBH               ; İlk formatlama verisimi !?
          JR Z,L6
          CP 0CCH               ; İkinci formatlama verisimi ?
          JR NZ,L6
          LD D,TABBES          ; Tablo adresi üst sekizlisi
L8        : LD C,2               ; 2 paket okunacak
L7        : LD B,1               ; 1 paket daha oku
          CALL READ1
          RET NZ                 ; Okumada hata var mı ?
          DEC HL
          LD A,(HL)
          CP 0CCH               ; Sonlandırma verisimi ?
          RET Z                 ; Bellekte saklanır
          LD E,A               ; Tablo adresi alt sekizlisi
          LD A,(DE)            ; Verinin kod çözülmesi
          ; karşılığı A da
          DEC HL
          RRD                   ; Dörtlük düzenlemesi yapılır
          INC HL
          DEC C
          JR NZ,L7             ; İki pakette okundumu ?
```



```
INC HL
JR LB           ; iki paket daha oku
RET

; ***** Kesme Hizmet Programı *****
; Adresi INTVEH 00 ve INTVEH 01 gözlerinde
PUSH AF
PUSH IY        ; AF ve IY saklanır
LD IY,TUR1
IN A,(PADATA)  ; Motor yönünü belirlemek için
AND 6
JR Z,JMP7
INC (IY)       ; Sağa dönüş
JR NZ,JMP8     ; TUR1 doldumu ?
INC IY
INC (IY)
JMP8          : POP AF
               ; AF ve IY çekilir
               EI
               RETI
JMP7          : DEC (IY)       ; Sola dönüş
               ; TUR1 sıfır mı ?
               INC IY
               DEC (IY)       ; TUR2 eksiltilir
               JR JMP8

; ***** Program Parametreleri *****

CURTI1        : DEFB CCONT2
               DEFB 47H       ; Sayıcı modu,zaman sa. bekler
               DEFB CCONT2
               DEFB 03H       ; 9600 baud için zaman sabiti
UARTI1        : DEFB URTCNT
               DEFB 0         ; SIA sıfırlanır
               DEFB URTCNT
               DEFB 40H       ; SIA "idle" modda
               DEFB URTCNT
               DEFB 7EH       ; 1 stop bit, çift eşlik, 8 bit
               DEFB URTCNT    ; lik karakter, *16 faktör
               DEFB 37H       ; Hata bayrakları sıfırlanır
URTCNT        : EQU 61H       ; SIA kontrol saklayıcısı adr.
CCONT2        : EQU 66H       ; SZD 2. kanal adresi
CURTI2        : DEFB CCONT2
               DEFB 47H       ; SZD sayıcı modunda
               DEFB CCONT2
               DEFB 04H       ; 6992 bit/sn. için şartlan.
               DEFB URTCNT
```



```

DEFB 0
DEFB URTCNT
DEFB 40H
DEFB URTCNT
DEFB 7AH ; 1 stop bit, çift eşlik, 7
DEFB URTCNT ; bitlik karakter, *16 faktör
DEFB 37H ; Hata bayrakları sıfırlanır
CPAINI : DEFB CCONTO
DEFB 7FH ; SZD sıfırlanır, zaman sabiti
DEFB CCONTO ; beklenir
DEFB 0AH ; Zaman sabiti 10
DEFB CCONTO
DEFB 79H ; Çalışma izni
DEFB PACONT
DEFB 0FH ; PA giriş
DEFB PADATA
DEFB 01H ; Sola sarma verisi
PACONT : EQU 6AH ; A iskelesi kontrol sak. adr.
PADATA : EQU 68H ; A iskelesi veri sak. adresi
PBCONT : EQU 6BH ; B iskelesi kontrol sak. adr.
PBDATA : EQU 69H ; B iskelesi veri sak. adresi
CCONTO : EQU 64H ; SZD 0. kanal adresi
URTD A : EQU 60H ; SIA veri saklayıcısı adresi
INSBUF : EQU 0F000H ; Komut tamponu başl. adresi
LISBUF : EQU 2000H ; Bölüm tamponu başl. adresi
TUR1 : EQU 0F010H ; Devir sayısı sayacı 1
TUR2 : EQU 0F011H ; Devir sayısı sayacı 2
LISFLG : EQU 0F013H ; Liste yapısı bayrağı
EPABUF : EQU 0F020H ; Boş bölümler yapısı tamponu
EMPLEN : EQU 144 ; Boş bölümler yapısı uzunluğu
LISLEN : EQU 3FFFH ; Liste yapısı uzunluğu
INTVEH : EQU 2EH ; Kesme vektörü üst sekizlisi
SECCDU : EQU 0F00EH ; Bölüm sayacı
HBTAB : EQU 0FAH ; Kodlama tablosu adresi
; üst sekizlisi
TABBEG : EQU 0F200H ; Kod çözme tablosu başl.adresi
PARTAB : EQU 0F300H ; Bölüm adresleri tablosu
SCAN1 : EQU 029BH ; MPF.1P monitor
FRTMES : EQU 0886H ; programlarından
READYM : DEFM "HAZIRIM"
DEFB 0DH
ERRMS : DEFM "PE/FE * TEKRARLAYINIZ"
DEFB 0DH
ERRMS1 : DEFM "LUTFEN TEKRARLAYINIZ"
DEFB 0DH
NOCAMS : DEFM "KASET YOK"
DEFB 0DH
DIFCAS : DEFM "ISTENEN KASET DEGIL."
DEFB 0DH
OKEYMS : DEFM "ISLEM TAMAMLANDI"

```


EMPMSS : DEFB ODH
DEFM "KASSETTE DOSYA YOK"
DEFB ODH
REBEGM : DEFM "BASLA KOMUTUNU YURUT"
DEFB ODH
NOFIMS : DEFM "ARANAN DOSYA YOK"
DEFB ODH
WPMSG : DEFM "YAZMA KORUMASI VAR"
DEFB ODH
CONMSG : DEFM "DEVAM EDINIZ"
DEFB ODH
REDAMS : DEFM "VERIYI TEKRARLAYINIZ"
DEFB ODH
INERMS : DEFM "HATALI BASLAMA"
DEFB ODH
CAFUMS : DEFM "KASET DOLU"
DEFB ODH
WAITMS : DEFM "LUTFEN BEKLEYINIZ"
DEFB ODH

5.5.2 - Bilgisayar Yazılımı

İşletim sisteminin anlatımında görüldüğü gibi arabirimle bilgisayar arasındaki iletişim, işletim sisteminin temelini oluşturur. Bunun için 9600 baud sabit hızında asenkron seri veri iletişimi kullanılır.

Komutlar bilgisayara 10 karakter olarak gönderilir. "Oku" ve "Listele" komutları gönderildiği zaman bilgisayar liste yapısını okumak için seri veri alma konumunda beklemelidir. Arabirimin gönderdiği liste yada dosya verilerini okuyarak belikte saklamalıdır. "Listele" komutunda okunan liste yapısının anında veya okuma işlemi bittikten sonra listelenmesi yapılmalıdır.

"Yaz" komutunda yazılacak dosyanın ismi, yeni-eski bayrağı kullanacağı bölüm sayısı, yükleme yapılacak bölümler, boş bölümler yapısı yüklendikten sonra dosya verileri gönderilmelidir. "Sil" komutunda bilgisayarda saklanan liste yapısında, silinecek dosya ile ilgili tüm gözlemlere FF verisi yüklenir. Ayrıca boş bölümler yapısına dosyanın kullanmakta olduğu bölümler eklenir. "Düzenle" komutunda, liste yapısı ve boş bölümler yapısı arabirime gönderilerek yüklenmesi sağlanır.

Kopya etme işleminde kopya edilecek dosyaya ait veriler okunarak saklanır. Dosyanın kopya edileceği kasetten liste yapısı ve boş bölümler yapısı okunur, yeni dosya için düzenlenir ve kasete tekrar yüklenir. Yapılan düzenlemeye uygun olarak yeni dosya verileri de yazılır. Formatlama işleminde sadece kasetin adını göndermek yeterlidir.

İşte, bilgisayar yazılımı bu işlevleri sağlayacak şekilde düzenlenmelidir.

6 - SONUÇ

Bir ses kasetinin rasgele erişimli bir bellek olarak kullanımının mümkün olduğu görülmüştür. Böyle bir bellek, alışılmış kaset birimlerinden daha hızlı ve kullanışlı olacaktır. Ancak sistemin tam anlamıyla gerçekleştirilmesi için oldukça ayrıntılı ve geniş kapsamlı bir çalışma gerekir.

Sistem tasarımı sırasında ortaya çıkan sorunların çözümü, istenmeyen bazı sonuçlar doğurmuştur. Örnek olarak, sabit, uzun süreli verilerin yüklenmesinde ve okunmasında oluşan hatalar, verinin kodlanması ile giderilmiştir. Fakat verinin iki paket olarak yüklenmesi kaset kapasitesinin yarıya düşmesine sebep olmuştur. Daha uygun bir yükleme metodunun bulunarak, verinin doğru yüklenmesi ve okunması sağlanırsa kaset kapasitesi artırılabilir.

NRZ yöntemiyle yapılan kayıt büyük emniyet sağlamıştır. Çünkü kafa akımının hiçbir zaman sıfır olmaması sürekli olarak kasete veri yüklemek demektir. Bu durum eski kayıtlardan doğan sakıncaları ortadan kaldırır.

MPF.1P sisteminin bellek kısıtlaması nedeniyle 8 KB. lık bölüm kapasitesinin 12 KB. olarak artırılmasının daha uygun olacağı görülmüştür. Çünkü 11 KB. dan fazla olan liste yapısının okunup saklanması sorun çıkarmıştır. 12 KB. lık bölüm kapasitesi için gerekli bellek bölgesi, MPF.1P sistemine bellek eklenerek oluşturulabilir.

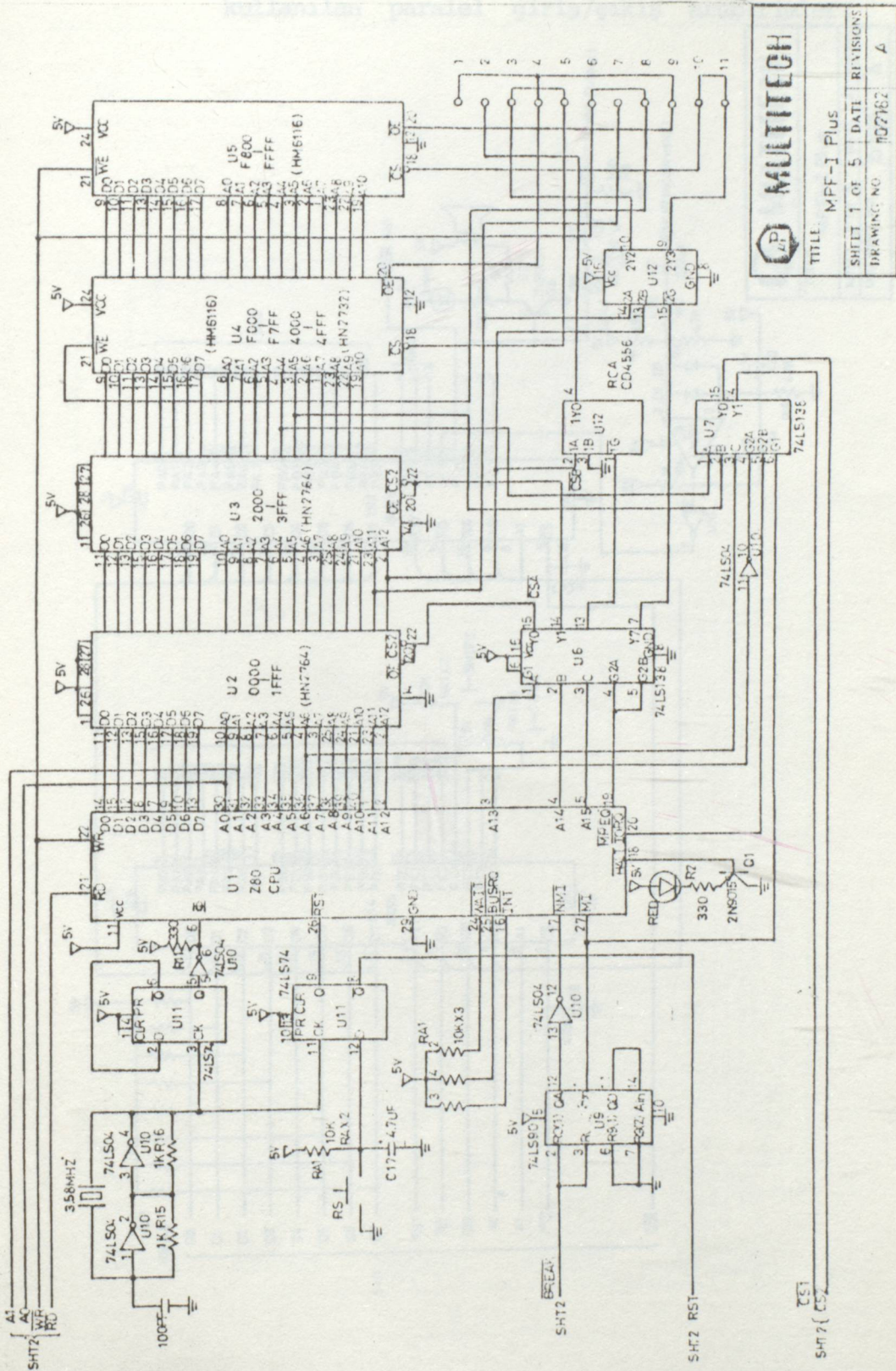
Sistemin çalışması için arabirimle bilgisayar arasındaki veri iletişimi uygun şekilde sağlanmalı ve kullanıcı bilgisayar programları yazılmalıdır.

Ek - 1.3 MPF.1P sisteminin KİS ve bellek şeması

EKLER ;

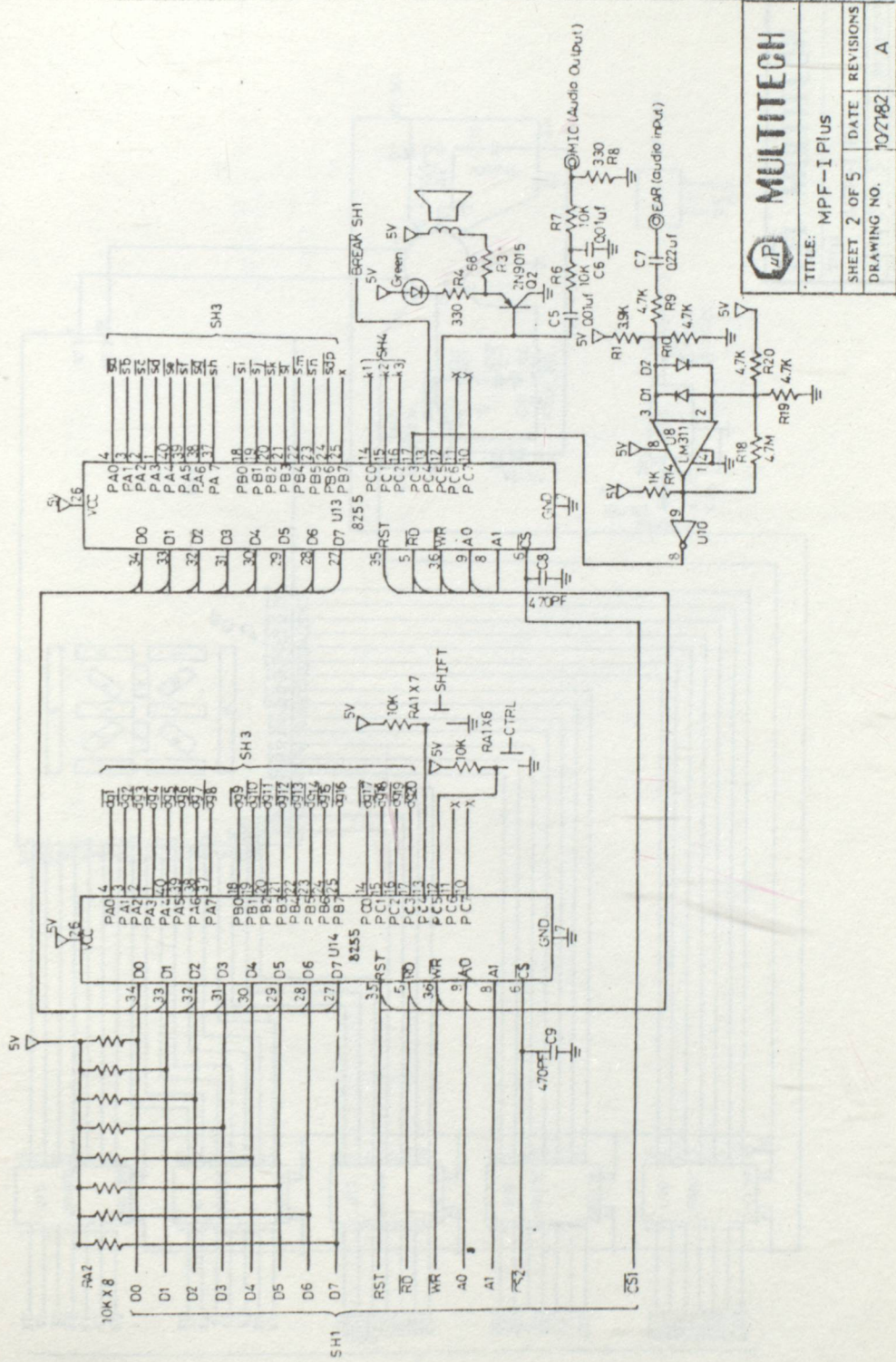
1. MPF.1P Sistemi Donanım Şeması
2. MPF.1P Sistemi Giriş/Çıkış Kartı Donanım Şeması
3. Arabirim Sistemi Besleme Kaynakları

Ek - 1.a MPF.1P sisteminin MIB ve bellek düzenlemesi



		TITLE: MPF-1 Plus	
		SHEET 1 OF 5	DATE: REVISIONS
DRAWING NO: 102162		4	

Ek - 1.b MPF.1P sisteminin gösterge ve tuş takımı için kullanılan paralel giriş/çıkış arabirimleri

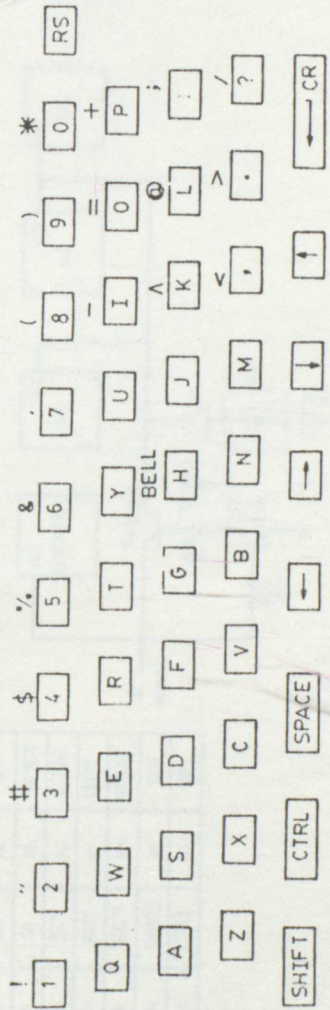
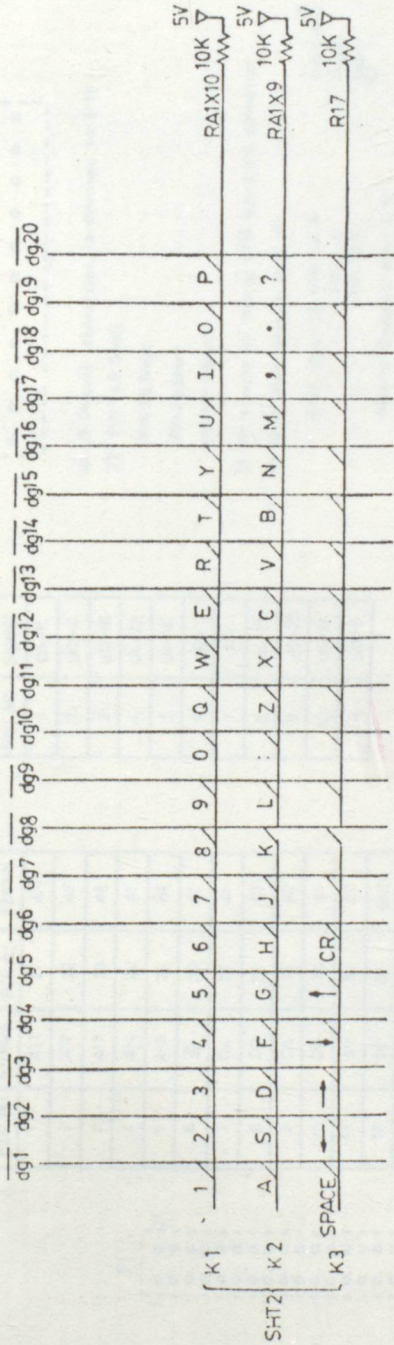


MULTITECH

TITLE: MPF-I Plus

SHEET	DATE	REVISIONS
2 OF 5		
DRAWING NO.	107182	A

Ek - 1.d MPF.1P sisteminin tuş takımı düzeni

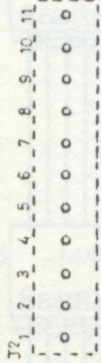


MULTITECH

TITLE: MPF-1 PLUS

SHEET 4 OF 5	DATE	REVISIONS
DRAWING NO. 102182		A

Ek - 1.e MPF.1P sisteminin besleme ve diğer kartlara bağlantı düzeni



U4, U5 Default connection is destined for 6116

J2: PIN 1,4,9 Short

PIN 3,5 Short

PIN 6,8 Short

PIN 10,11 Short

If user's want to change 6116 into 5516 connection for lower power battery back up

first: Cut J2 PIN 1,4,9

PIN 3,5

PIN 10,11

Second: Connect PIN 1,5,10 PIN 3,4 PIN 9,11

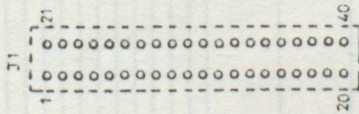


J2 PIN FUNCTION

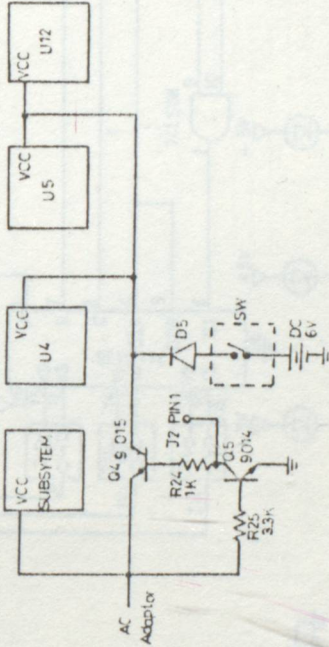
PIN NO	SIGNAL
1	D5-C
2	U17-4
3	U17-10
4	U4-20
5	U4-16
6	WR
7	A11
8	U4-21
9	U5-20
10	U5-16
11	U12-9

J1 PIN FUNCTION

PIN NO	SIGNAL	PIN NO	SIGNAL
1	A11	21	A10
2	A12	22	A9
3	A13	23	A8
4	A14	24	A7
5	A15	25	A6
6	$\overline{A6}$	26	A5
7	D4	27	A4
8	D3	28	A3
9	D5	29	A2
10	D6	30	A1
11	5V	31	A0
12	D2	32	GND
13	D7	33	REFSH
14	DC	34	TR1
15	D1	35	RESET
16	INT	36	BUSRC
17	INT	37	WALT
18	HALT	38	BUSAR
19	MREQ	39	WR
20	TORQ	40	RD



SW	Description
ON	Auto battery back up
OFF	No battery back up



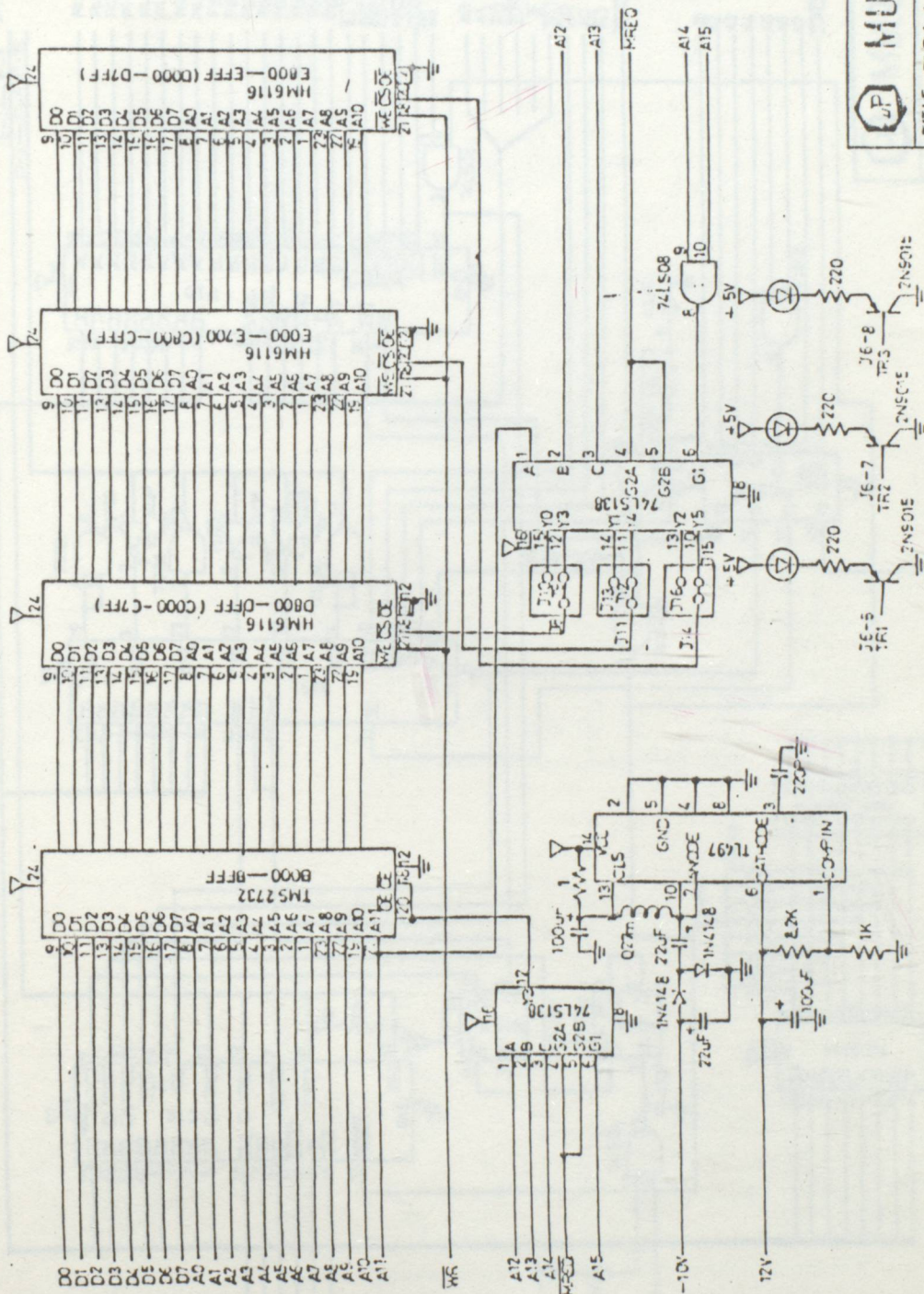
MULTITECH

TITLE: MPF-1 Plus

SHEET 5 OF 5 DATE: 10/21/82 REVISIONS: A

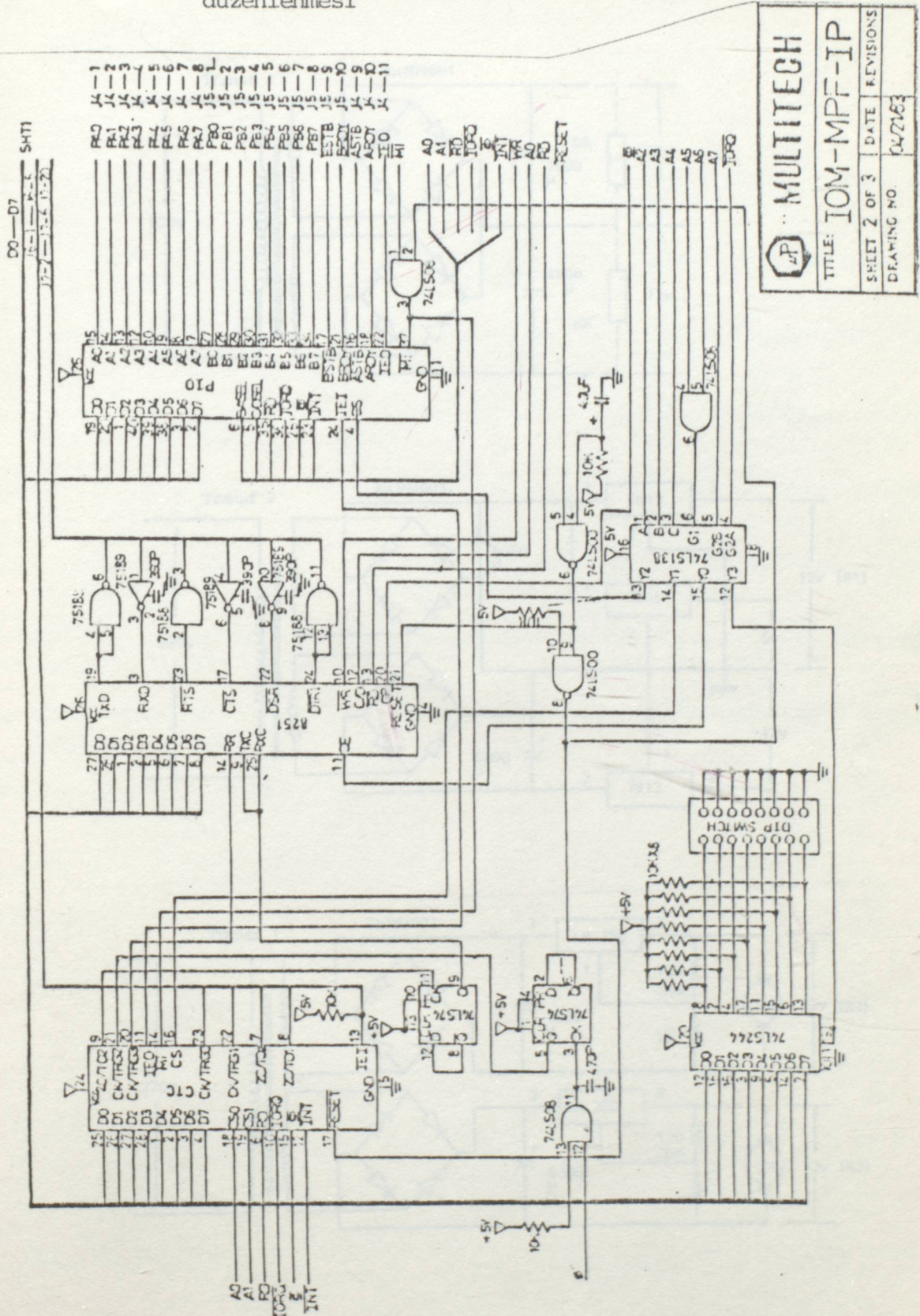
DRAWING NO: 102762

Ek - 2.a MPF.1P sisteminin giriş/çıkış kartı bellek düzeni



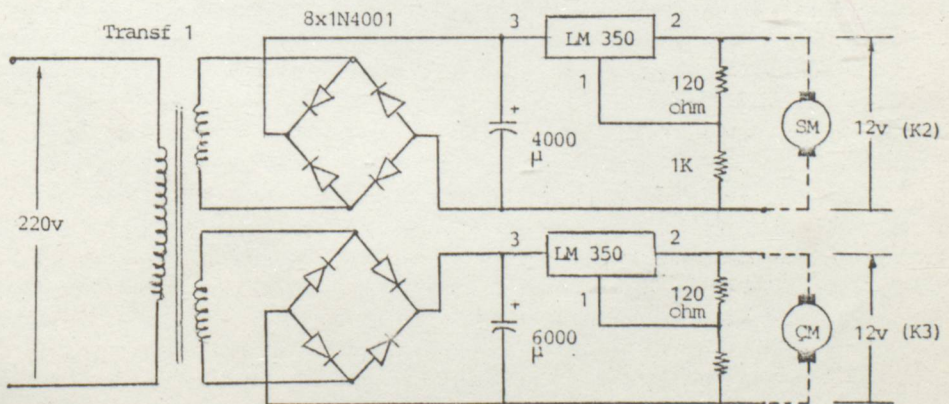
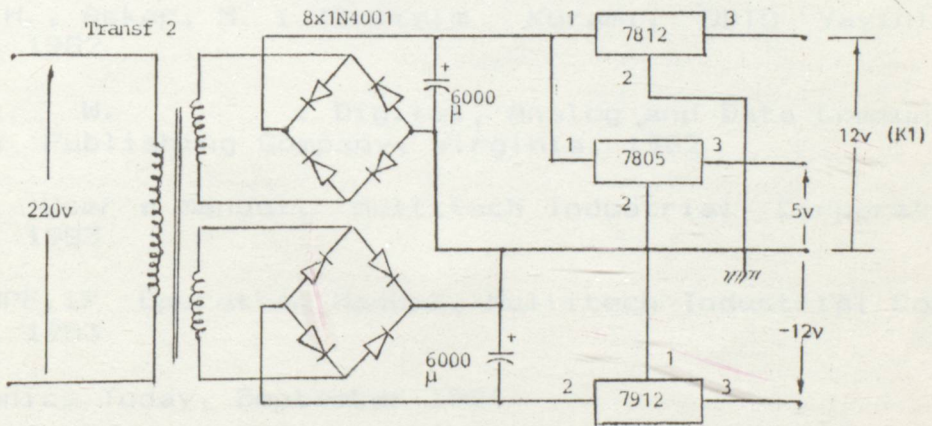
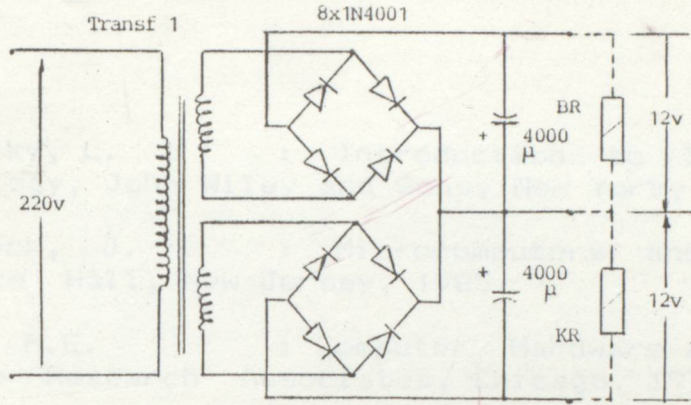
	MULTITECH	
	TITLE IOM-MPF-IP	
	SHEET 1 OF 3	DATE
	DRAWING NO 04/2163	REVISIONS

Ek - 2.b MPF.1P sisteminin giriş/çıkış kartı arabirimlerinin düzenlenmesi



	TITLE: IOM-MPF-1P	
	SHEET 2 OF 3	DATE REVISIONS
DRAWING NO. 042163		

Ek - 3 Arabirim sistemi besleme kaynakları



KAYNAKLAR ;

- 1- Nashelsky, L. : Introduction to Digital Computer Technology, John Wiley and Sons, New York, 1977
- 2- Uffenbeck, J. : Microcomputers and Microprocessor, Prentice Hall, New Jersey, 1985
- 3- Sloan, M.E. : Computer Hardware and Organization, Science Research Associates, Chicago, 1976
- 4- Tanenbaum, S.A. : Computer Networks, Prentice Hall, Englewood Cliffs, 1981
- 5- Derin, H., Askar, M. : İletişim Kuramı, ODTÜ Yayınları, Ankara, 1982
- 6- Sinnema, W. : Digital, Analog and Data Communica. Preston Publishing Company, Virginia, 1982
- 7- MPF.1P User's Manual, Multitech Industrial Corporation, Taiwan, 1983
- 8- IDM - MPF.1P Operation Manual, Multitech Industrial Corp., Taiwan, 1983
- 9- Electronics Today, September 1984

ÖZGEÇMİŞ ;

1963 yılında Seydişehir' de doğdum. İlk öğrenimimi Konya' da tamamladım. 1980 yılında Seydişehir Endüstri Meslek Lisesi Elektrik Bölümü' nden mezun olarak Yıldız Üniversitesi Elektronik ve Haberleşme Mühendisliği Bölümü'nde okumaya hak kazandım . Şubat-1985' de Elektronik ve Haberleşme Mühendisi olarak mezun oldum ve Yıldız Üniversitesi Fen Bilimleri Enstitüsü' nde yüksek lisans öğrenimine başladım.

Ekim-1985' de Yıldız Üniversitesi Bilgisayar Bilimleri Mühendisliği Bölümü, Bilgisayar Donanımı Anabilimdalı' nda araştırma görevlisi olarak çalışmaya başladım ve halen bu görevimi sürdürmekteyim.

