

YILDIZ TEKNİK ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

Mikroişlemci Kontrollü Elektriksel
İşaret Sentezi

Selman Tanyıldız

Yüksek Lisans Tezi

Ref
EHM
223
1986

YILDIZ UNIVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MİKROİŞLEMCİ KONTROLLU ELEKTRİKSEL
İŞARET SENTEZİ

YÜKSEK LİSANS TEZİ
ELEKTRONİK VE HAB. MÜH. SELMAN TANYILDIZ

İSTANBUL 1986

YILDIZ UNİVERSİTESİ
GENEL KİTAPLIĞI

Kot : R 152
Alındığı Yer : Fen Bil. Ens 33
Tarih : 10.4.1987
Fatura :
Fiatı : 1000 TL.
Ayniyat No : 1/1
Kayıt No : 44769
UDC : 001.64
Ek : 378.242

YILDIZ
UNİVERSİTESİ
KUTUPHANE

YILDIZ ÜNİVERSİTESİ

A-ÖZET FEN BİLİMLERİ ENSTİTÜSÜ

B-SUMMARY

C-GİRİŞ

MİKROİŞLEMCİ KONTROLLU ELEKTRİKSEL

İŞARET SENTEZİ

D-SANAYSEL ÇALIŞMA

II-Kullanılan Sistemin Tanıtılması

II-1)Sistem Özellikleri

II-2)8085 Mikroislemcisi

II-3)8085 Giriş Çıkış Devreleri

II-4)8085 Sayısal Zamanlayıcı Devresi

II-5)8255 Programlanabilir Çökyönlü

YÜKSEK LİSANS TEZİ

ELEKTRONİK VE HAB. MÜH. SELMAN TANYILDIZ

Temel Alt-programlar

III-kullanılan Mikroislemcili Sistemin

Çalıştırılması

III-1)Alt-programlarda Yapılan

Değişiklikler

III-2)8085 Haberleşme Kavramı

III-3)Terminal Haberleşme Programı

III-4)Mikrobilgisayar Terminal Programı

İSTANBUL 1986

İÇİNDEKİLER

A-ÖZET

B-SUMMARY

C-GİRİŞ

I-Mikroişlemci Kontrollü Elektirksel

İşaret Sentez Yolları.....1

D-DENEYSEL ÇALIŞMA

II-Kullanılan Sistemin Tanıtılması.....7

II-1)Sistem Özellikleri.....8

II-2)Z80 Mikroişlemcisi.....9

II-3)Z80 Paralel Giriş Çıkış Devresi.....15

II-4)Z80 Sayıcı Zamanlayıcı Devresi.....24

II-5)8251 Programlanabilir Çokyönlü

Gönderici Alıcı.....30

II-6)Monitör Programda Yer Alan

Temel Altprogramlar.....36

III-Kullanılan Mikroişlemcili Sistemin

Geliştirilmesi.....38

III-1)Altprogramlarda Yapılan

Değişiklikler.....39

III-2)Seri Haberleşme Kavramı.....40

III-3)Terminal Haberleşme Programı.....43

III-4)Mikrobilgisayar Terminal Programı.....49

ÖZET

IV-Karakter Üretim Kavramı.....	54
IV-1)Osiloskop Ekranında Karakter Üretilmesi.....	57
IV-2)Televizyon Ekranında Karakter Üretilmesi.....	62
IV-3)Matris Noktalar İle Karakter Üretimi.....	66
E-SONUÇ	
V-Mikroişlemci Belleğinde İşaret Sentezi Yapılarak Karakter Üretimi.....	71
F-KAYNAKLAR.....	78
G-ÖZGEÇMİŞ.....	79

ÖZET

Mikroişlemciler, haberleşme sistemleri ve endüstriyel kontrol sistemleri gibi uygulamalarda çokça kullanılırlar. Bu sistemlerde , mikroişlemci kontrollu elektriksel işaret sentezi temel uygulamalardan biridir. Mikroişlemci temelli sistem fonksyonları programlama ile kolayca değiştirilebilir. Bu nedenle , elektriksel işaret sentezi bir programlama kavramı olmaktadır.

Sayısal ve örneksel işaretler, mikroişlemci ile sentezlenebilir. Örneksel işaretin sentezlenebilmesi için , ilkönce sayısal işaret üretilmelidir. Daha sonra bu sayısal işaret, sayısal örneksel dönüştürücü kullanılarak örneksel işaret haline getirilir.

Karakter üretimi, bir elektriksel işaret sentez konusudur. Karakter bilgisini göstermek için , bu veri elektriksel işaretlere dönüştürülmelidir.

Pratikte, mikroişlemci kontrollu elektriksel işaret sentezi uygulamaları, fiziksel bir büyüklüğü üretmek için tasarlanmış cihazlarda kullanılır.

SUMMARY

The Microprocessors are widely used in application as communication systems and industrial control systems. The electrical signal synthesis controlled by microprocessors is one of the basic applications in these systems. The microprocessor based system functions are easily changed by programming. Thus, the synthesis of electrical signal becomes a programming concept.

Both digital and analog signals can be synthesized by the microprocessors. First, the digital signal must be generated to synthesis an analog signal. Then, this signal is converted to the analog signal by using a digital to analog converter.

The character generating is a subject of the electrical signal synthesis. The data which contains a character information must be converted to electrical signals for displaying.

In practice, the applications of electrical signal synthesis controlled by microprocessors are used in the devices that are designed to generate a physical quantity.

I-MİKROİŞLEMCI KONTROLLÜ ELEKTRİKSEL İŞARET

SENTEZ YOLLARI

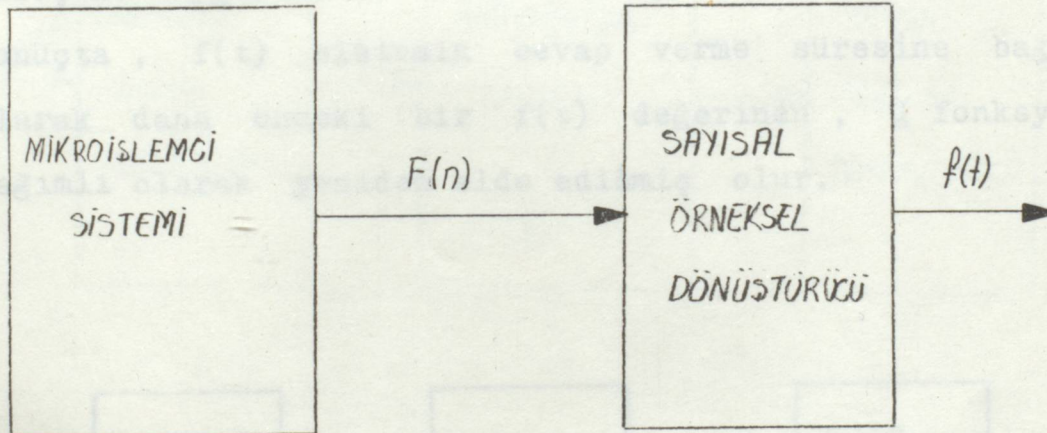
Elektronik kontrol ve haberleşme sistemlerinde belirli özelliklere sahip işaretlerin üretilmesi gereği ortaya çıkmaktadır. Sentezlenmek istenen işaret , sürekli zaman düzleminde örneksel bir işaret veya ayrık zaman düzleminde sayısal bir işaret olabilir. Mikroişlemci , yürüttüğü program ve çevrebirimleri ile olan veri iletişimi sayesinde istenen işaretleri , belli zamanlama sınırları içinde kalmak şartı ile sentezleyebilir.

Mikroişlemci , sayısal bir yapıya sahip olması nedeni ile , sadece sayısal işaretlerin sentezlemesini yapabilir. Eğer örneksel işaretler elde edilmek istenirse , istenen örneksel işaretler ile ilişkili sayısal işaretler üretilmeli , ve daha sonra sayısal-örneksel dönüşüm yapılmalıdır.

Aşağıda , mikroişlemci kontrollü elektriksel işaret sentezleme yollarından bazıları anlatılmıştır.

a-Doğrudan Sayısal-Örneksel Dönüştürme:

Mikroişlemci bellek veya giriş-çıkış çevrebirimleri haritasında , belirli bellek veya bellek gözlerine yüklenen veri , sayısal değeri ile orantılı olarak gerilim veya akım değerine dönüştürülür. Böylelikle , mikroişlemciye bağlı sayısal örneksel dönüştürücü yardımı ile , sayısal ortamdan örneksel ortama geçiş sağlanmış olur.



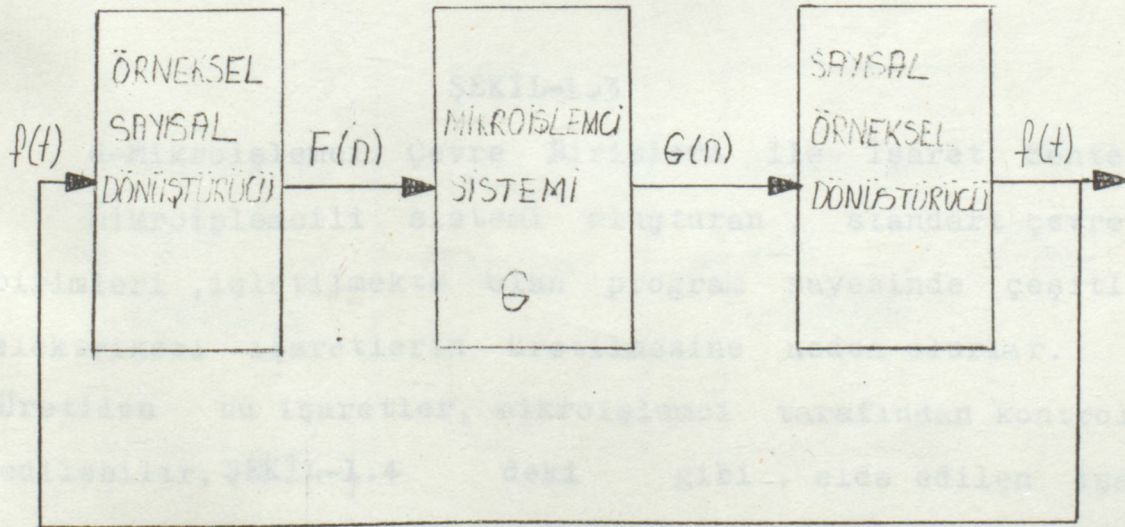
ŞEKİL-1.1

ŞEKİL-1.1 de görüldüğü gibi , mikroişlemci tarafından üretilen $F(n)$ sayı dizisi , sayısal örneksel dönüştürücüye uygulanarak , $f(t)$ fonksyonu elde edilir. Elde edilen bu fonksyon , istenen $f(t)$ fonksyonuna n zaman aralığının t 'ye karşılık gelen değerlerinde eşittir. Sayısal örneksel dönüştürücünün karakteristiğine bağlı olarak , çıkış $f(t)$ işaretini $F(n)$ dizisinin

fonksiyonudur. $F(n)$ nin üretilmesi mikroişlemci programına bağlıdır. Bu sebeple üretilen $f(t)$ işareti mikroişlemci kontrollüdür.

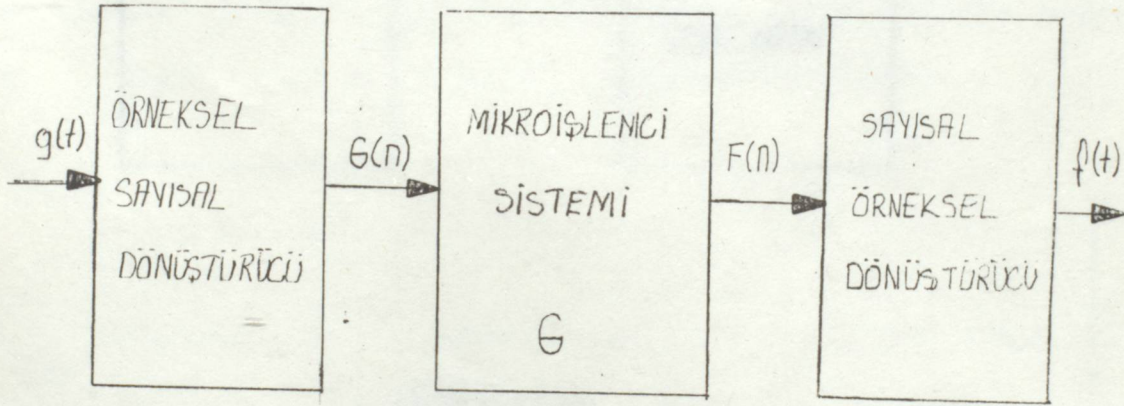
b-Geribeslemeli Doğrudan Sayısal-Örneksel Dönüştürme :

Şekil-1.2 de görüldüğü gibi, üretilen $f(t)$ fonksyonu örneksel-sayısal dönüşüm yoluyla $F(n)$ dizisine dönüştürülerek mikroişlemciye gönderilir, Mikroişlemcili sistemin sağladığı G sayısal fonksyonuna bağlı olarak $G(n)$ dizisi üretilir. $G(n)$ değeri sayısal-örneksel dönüştürücü girişine uygulanarak $f(t)$ fonksyonu üretilmiş olur. Sonuçta , $f(t)$ sistemin cevap verme süresine bağlı olarak daha önceki bir $f(t)$ değerinden , G fonksyonuna bağımlı olarak yeniden elde edilmiş olur.



ŞEKİL-1.2

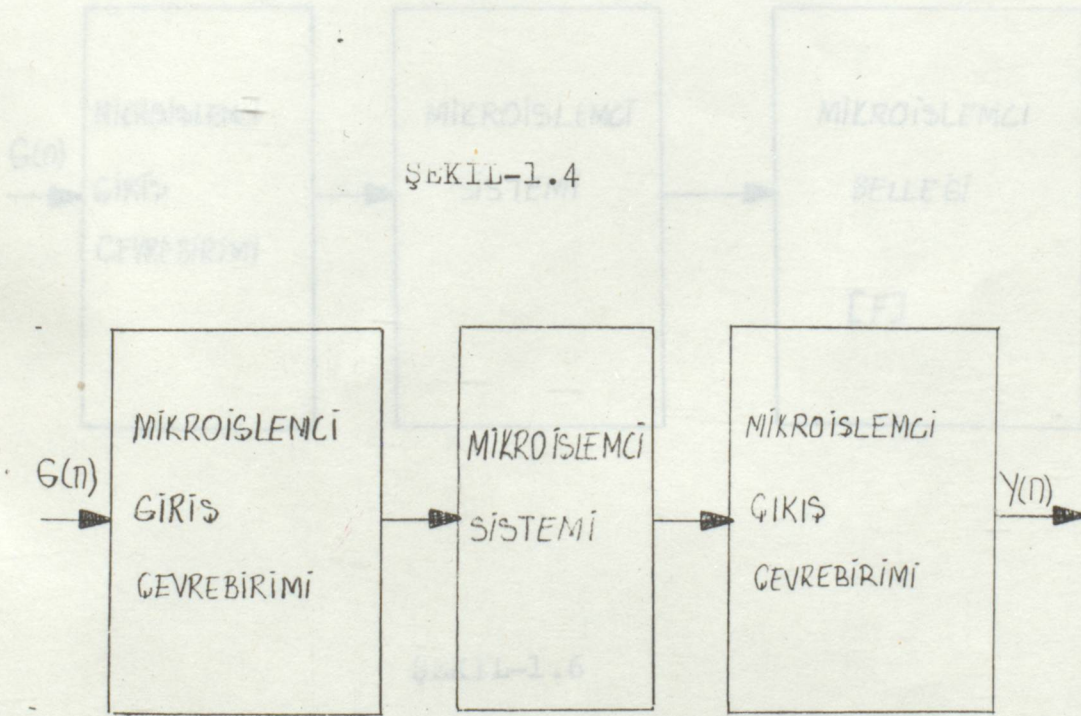
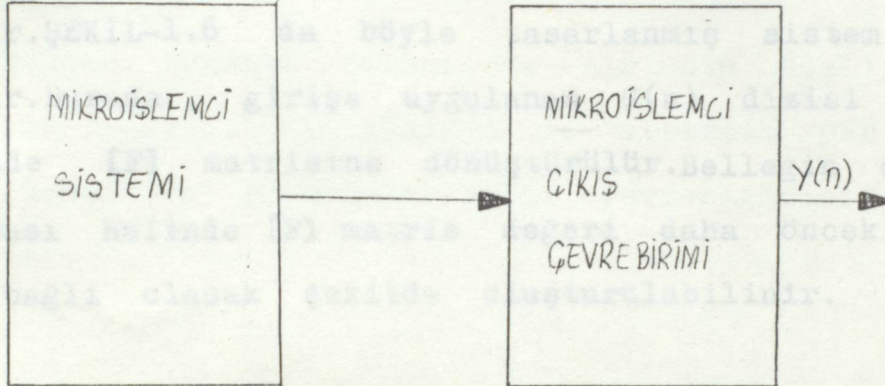
c-İşaret İzleyerek Doğrudan Sayısal-Örneksel Dönüşüm:
 ŞEKİL-1.3 deki gibi , üretilen $f(t)$ işareti $F(n)$ dizisine , $F(n)$ dizisi G fonksyonu ve $G(n)$ dizisine , $G(n)$ dizisi de $g(t)$ işaretine bağımlıdır. Böylelikle elde edilen işaretin değişimi girişte yer alan $g(t)$ nin izlenmesi ile sağlanmış olur.



ŞEKİL-1.3

d-Mikroişlemci Çevre Birimleri ile İşaret Sentezi:
 Mikroişlemcili sistemi oluşturan , standart çevre birimleri , işletilmekte olan program sayesinde çeşitli elektriksel işaretlerin üretilmesine neden olurlar. Üretilen bu işaretler, mikroişlemci tarafından kontrol edilebilir, ŞEKİL-1.4 deki gibi , elde edilen işaret sayısal anlam taşımaya karşılık , örneksel ortamda sürekli işaretler olarak düşünülebilir. Çıkış işaret ve-ya işaretlerinin değişimi mikroişlemcinin yürütmekte

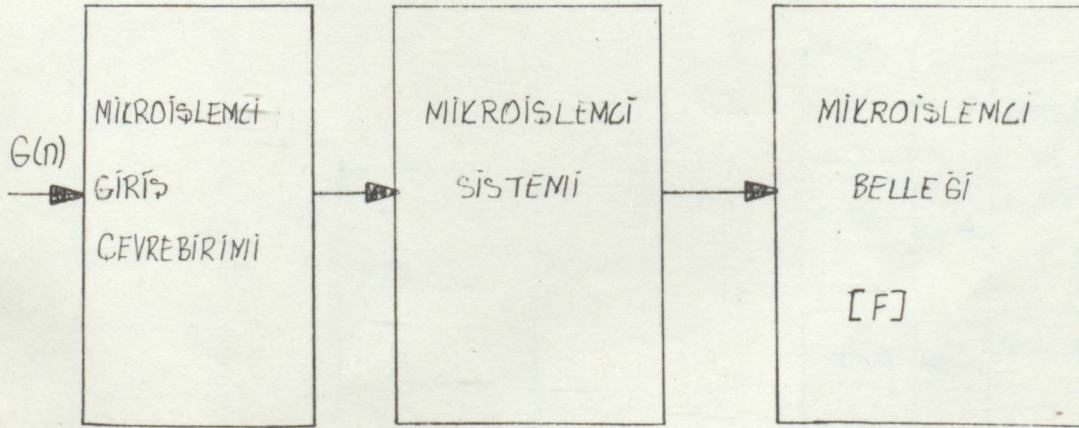
olduğu programa bağlıdır. ŞEKİL-1.5 ise en çıkışta elde edilen $Y(n)$, giriş çevre birimine uygulanan $G(n)$ işareti ve mikroişlecinin yürütmekte olduğu programın bir sonucudur.



ŞEKİL-1.5

e-Mikroişlemci Belleğinde İşaret Sentezi:

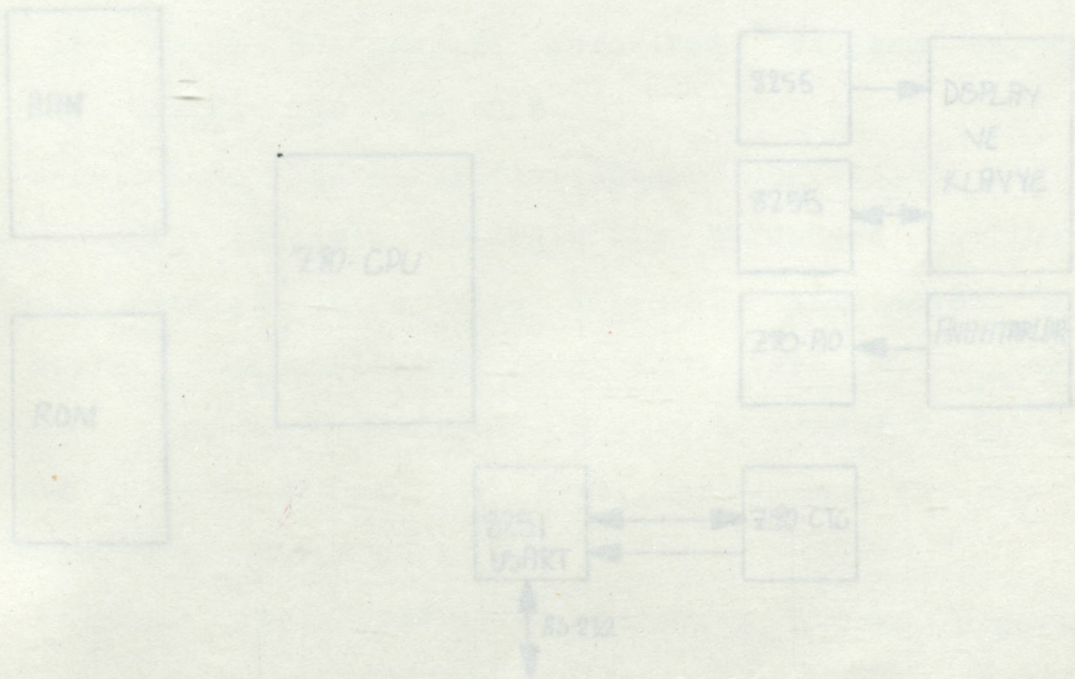
Mikroişlemci programı ile, bellekte bir işaret dizisi veya diğer fiziksel işaretlere dönüşebilen bir veri matrisi oluşturulabilir. Oluşan bu matrisin her elamanı yürütülmekte olan programa göre değişim gösterir. ŞEKİL-1.6 da böyle tasarlanmış sistem gösterilmiştir. Burada, girişe uygulanan $G(n)$ dizisi program sayesinde $[F]$ matrisine dönüştürülür. Belleğin okunabilir olması halinde $[F]$ matris değeri daha önceki değerine bağlı olacak şekilde oluşturulabilir.



ŞEKİL-1.6

II-KULLANILAN SİSTEMİN TANITILMASI:

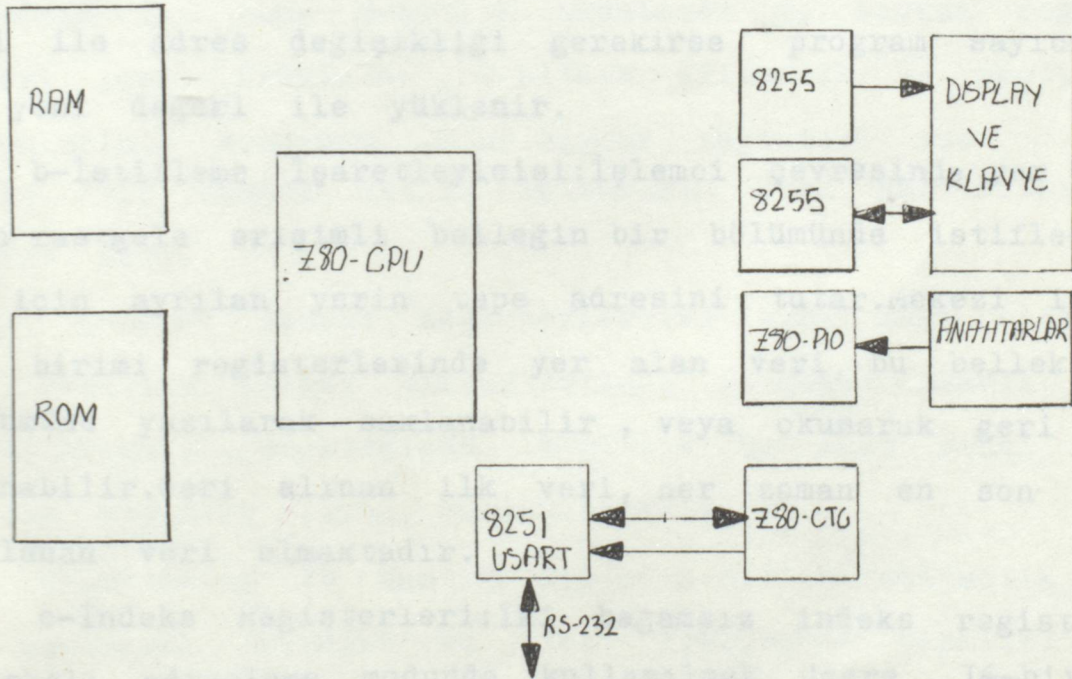
Bu çalışmada , MPF-IP Z80 temelli mikrobilgisayarı kullanılmıştır. Sistemde yapılan bazı değişiklikler sayesinde bilgisayarın çıkış özellikleri geliştirilmiştir. devresi , kontrol devrelerinden oluşmaktadır. Bundan sonraki bölümlerde sistemin özellikleri , yapılan yazılım değişikliği ve sistemde yer alan mikroişlemci ve programlanabilir çevrebirimleri anlatılmıştır. çevrebirim , 7 adet seri giriş/çıkış çevrebirimi ve 1 adet sayısal zamanlayıcı devresi ile donatılmıştır.



II-1-Sistem Özellikleri:

Aşağıdaki şekilde gösterildiği gibi , sistem , miroişlemci , yalnız okunur bellek , seri giriş/çıkış çevrebirimi , paralel giriş/çıkış çevrebirimleri , sayıcı /zamanlayıcı devresi , kontrol devrelerinden oluşmuştur.

Sistem , 26K Byte yalnız okunur bellek (ROM) , 4K Byte rastgele erişimli bellek (RAM) , 3 adet paralel giriş/çıkış çevrebirimi , 7 adet seri giriş/çıkış çevrebirimi ve 1 adet sayıcı/zamanlayıcı devresi ile donatılmıştır.



II-2-Z80 Mikroişlemcisi

Register Yapısı:

Merkezi İşlem Birimi Registerleri:Z80 mikroişlemcisi , toplam 208 bitlik okunabilir yazılabilir bellek içerir.Bu bellek 16 adet 8-bit register ve 4 adet 16-bit register şeklinde oluşturulmuştur.Altı genel amaçlı register ayrı iki küme olarak inşa edilmiştir.

Özel Amaçlı Registerler:

a-Program Sayıcısı:Program sayıcısı, bellekten alınıp okunan komutun 16-bitlik adresini tutar.Program sayıcısı içeriği adres hatlarına yüklendikten hemen sonra otomatik olarak bir artırılır.Eğer , komutun işlenmesi ile adres değişikliği gerekirse program sayıcısı yeni değeri ile yüklenir.

b-İstifleme İşaretleyicisi:İşlemci çevresinde yer alan rastgele erişimli belleğin bir bölümünde istifleme için ayrılan yerin tepe adresini tutar.Mekezi işlem birimi registerlerinde yer alan veri, bu bellek bölümüne yazılarak saklanabilir , veya okunarak geri alınabilir.Geri alınan ilk veri, ner zaman en son saklanan veri olmaktadır.

c-İndeks Registerleri:İki bağımsız indeks registeri indeksli adresleme modunda kullanılmak üzere , 16-bit taban adresini tutar.

d-Kesinti Sayfası Adres Registeri:Z80 mikroişlemcisi kesinti olayının neden olduğu programa dallanırken

bu register 8-bitlik yüksek dereceli adresi , kesintiye neden olan çevre birimi ise 8-bitlik düşük dereceli adresi sağlar.

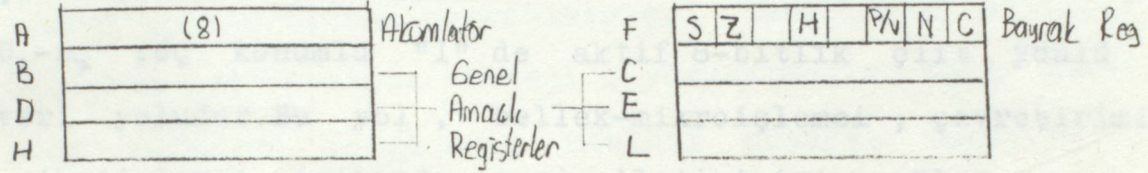
e-Bellek Tazeleme Registeri: Bu register sayesinde dinamik belleklerin kullanılması kolaylaşır. 8-bitlik registerin 7 biti, her komutun alınıp gelinmesinden sonra arttırılır. "LD R,A" komutu işlenmesi sonunda değeri ne ise 8. bit o değerde kalacaktır. Merkezi işlem birimi, bellekten alıp gelinmiş bir komutu işlerken , tazeleme sayıcısındaki veri düşük dereceli 8-bit adres hattına yollanır.

f-Akümlatör ve Bayrak Registerleri: Z80 birbirinden bağımsız iki tane 8-bitlik akümlatör ve bayrak registeri içerir. Akümlatör , 8-bitlik aritmetik ve mantık işlemlerinin sonucunu tutar. Bayrak registeri ise sekiz veya onaltı bitlik işlemlerin özel şartlarını işaret eder.

g-Genel Amaçlı Registerler: İki ayrı küme halinde , 8-bitlik ikiye birleştirilmiş 6 genel amaçlı registerler , gerektiğinde 16-bitlik üç adet genel amaçlı register olarak kullanılabilir.

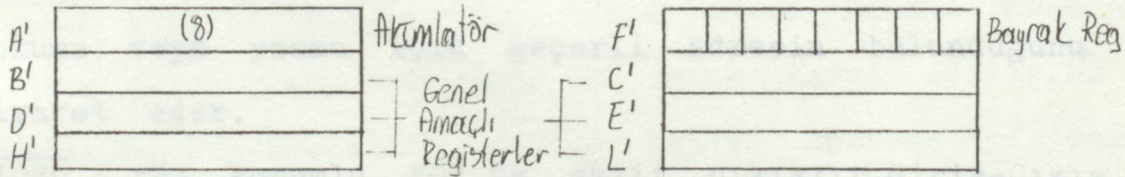
h-Aritmetik Ve Mantık Birimi: 8-bitlik aritmetik ve mantıksal işlemler bu birim içerisinde yapılır. Yapılan fonksiyon türleri: toplama , çıkartma , mantıksal ve , veya , özellikli veya , karşılaştırma , sağa-sola kaydırma , sağa-sola döndürme , arttırma , azaltma , bitlere "0" veya "1" yapma ve test etme olarak

sayılabilir. Z80 mikroişlemcisinin iç yapısı genel register düzeni ile ŞEKİL-2.1 de verilmiştir.



Register Köme Anahtarları

- ☐ AF veya A'F' seçer
- ☐ BC, DE, HL veya B'C'D'E'H'L' seçer



IX (16) } İndeks Registerleri

IY (16)

PC (16) Program Sayıcısı

SP (16) İstifleme İşaretleyicisi

IFF ☐ Kesinti Mücadelesi R (7) Bellek Terzedeme Registeri

IM (2) Kesinti Modu I (8) Kesinti Sayıları Adres Registeri

ŞEKİL-2.1

Z80-Mikroişlemcisi Tümdevre İşaret Ucları:

A_0-A_{15} :Üç konumlu "1" de aktif , 16-bitlik adres yoludur. 64 KByte kadar bellek , 256 Byte kadar giriş-çıkış çevrebirimini adresler.

D_0-D_7 :Üç konumlu "1" de aktif 8-bitlik çift yönlü veri yoludur. Bu yol , bellek-mikroişlemci , çevrebirimi -mikroişlemci arasında veri iletişimini sağlar.

$\overline{M}_1$: "0" da aktif çıkıştır. Komutun yürütülmesi sırasında mikroişlemci OP-kodunun alınıp gelinme sürecinde olduğunu gösterir.

$\overline{MREQ}$ Üç konumlu "0" da aktif çıkıştır. Adres yolunda , okuma veya yazma için geçerli adresin bulunduğunu işaret eder.

$\overline{IORQ}$:Üç konumlu "0" da aktif çıkıştır. Giriş-Çıkış birimleri için, okuma veya yazma durumunda adres yolu alt 8-bitinde geçerli değerin bulunduğunu gösterir.

$\overline{RD}$:Üç konumlu "0" da aktif çıkıştır. Bu çıkış merkezi işlem biriminin bellekten veya çevrebiriminden veri okuma isteğini belirtir.

$\overline{WR}$:Üç konumlu "0" da aktif çıkıştır. Merkezi işlem biriminin veri yoluna , yazılması istenen verinin gönderildiğini gösterir.

$\overline{HALT}$: "0" da aktif çıkıştır. Merkezi işlem biriminin HALT komutunu yürüttüğünü ve bu komuttan kurtulmak için kesinti beklediğini gösterir.

$\overline{WAIT}$: "0" da aktif giriştir. Mikroişlemciye adreslenen verinin transfer için hazır olmadığını belirtir. Merkezi

işlem birimi , bu giriş aktif olduğu sürece bekleme konumunda kalır.

$\overline{INT}$: "0" da aktif giriştir. Bu girişe uygulanacak işaret giriş-çıkış çevrebirimleri tarafından üretilir. Üretilen bu kesinti isteği , yürütülmekte olan komut sona erdikten sonra eğer kesinti müsdesi varsa ve $\overline{BUSRQ}$ işareti aktif değilse kabul edilir.

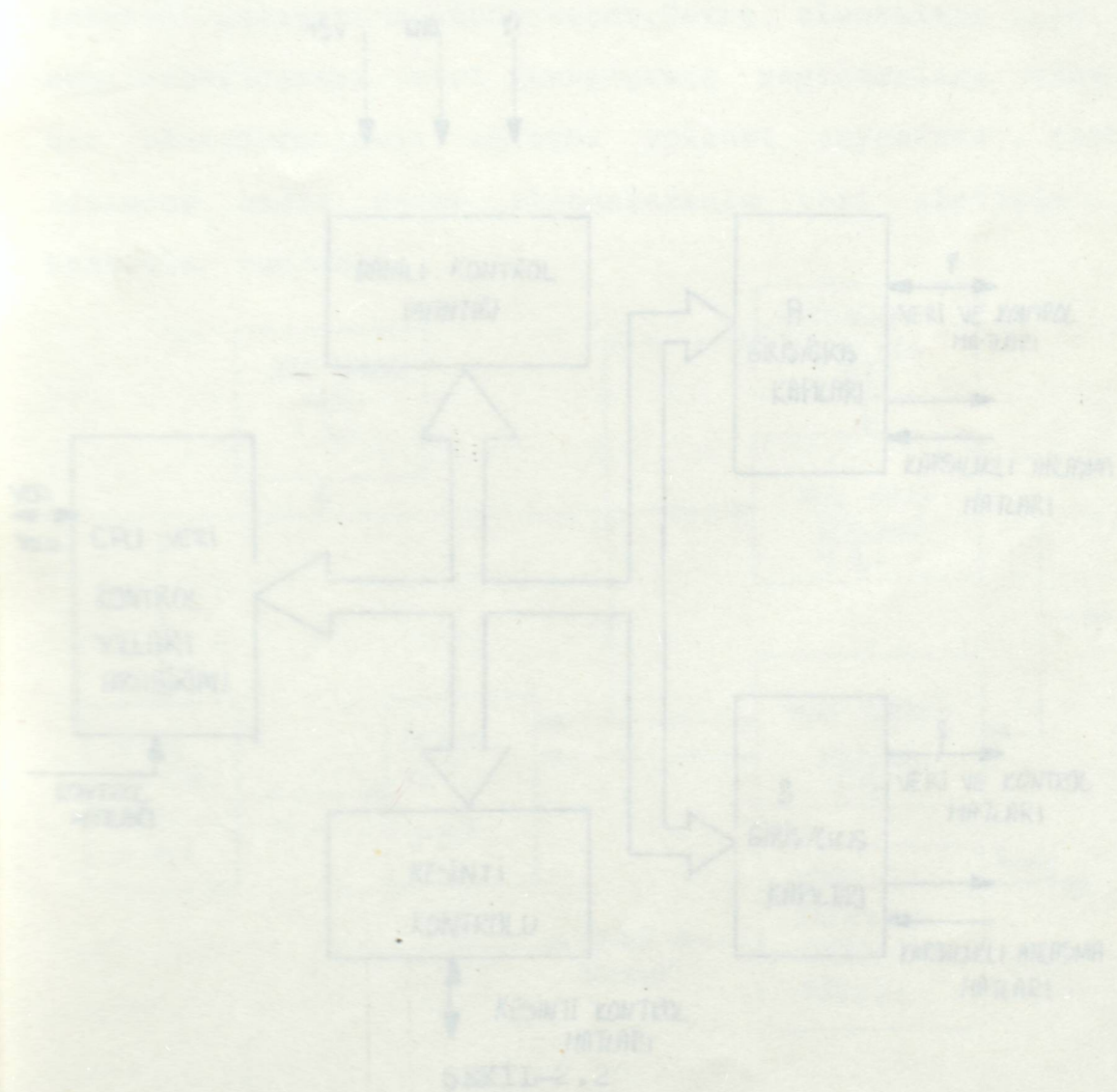
$\overline{NMI}$: "0" a düşerken aktif olan kenar tetiklemeli giriştir. Engellenemiyen kesinti isteği , diğer kesinti isteklerinden daha fazla önceliğe sahiptir. Yürütülmekte olan komutun bitmesinden hemen sonra, kesinti müsadesi olup olmadığına bakmadan, bu istek kabul edilir. Bu kesinti, mikroişlemciyi 0066H bellek gözünden itibaren yerleşmiş programın yürütmesine neden olur. Program sayıcısı değeri , daha sonra kullanılabilmesi için istif belleğine atılır.

$\overline{RESET}$: "0" da aktif giriştir. Program sayıcısını , I ve R registerlerini sıfırlar. Kesinti müsadde flip-flop-unu ve kesinti modunu "0" a şartlandırır.

$\overline{BUSRQ}$: "0" da aktif giriştir. Merkezi işlem birimi , adres , veri ve kontrol yolunun ortak yoldan ayırıp bağlı çevrebirimlerinin bu mikroişlemci dışındaki donanımla kullanılabilmesini sağlar.

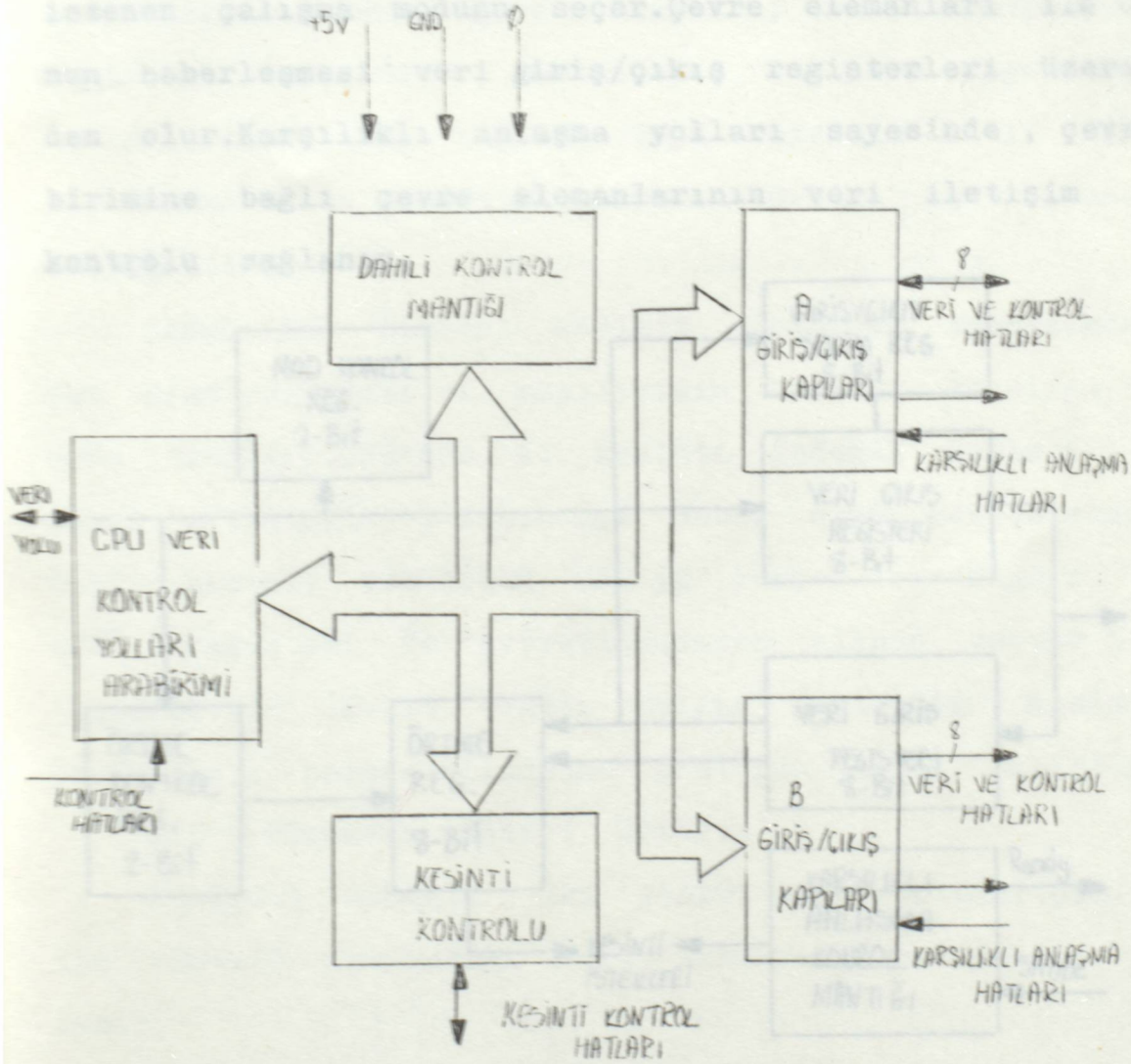
$\overline{BUSA\overline{K}}$: "0" da aktif çıkıştır. Mikroişlemci adres , veri ve kontrol yollarının yüksek empedans konumunda olduğu ve bu yolları talep eden donanımın bu yolları kullana-

bileceğini gösterir. Giriş/Çıkış Devresi
 0 :TTL seviyesinde , tek fazlı clock (saat) darbeleri
 girişidir.
 CPU veri ve kontrol arabirimi , dahili kontrol
 mantığı , paralel giriş/çıkış kapıları ve kesinti
 kontrol mantığından oluşmuştur. Dahili kontrol mantığı
 , CPU veri yolu ile A ve B paralel giriş/çıkış
 kapıları arasında eşzamanlılık sağlar.



11-3-Z80 Paralel Giriş/Çıkış Devresi (PIO)

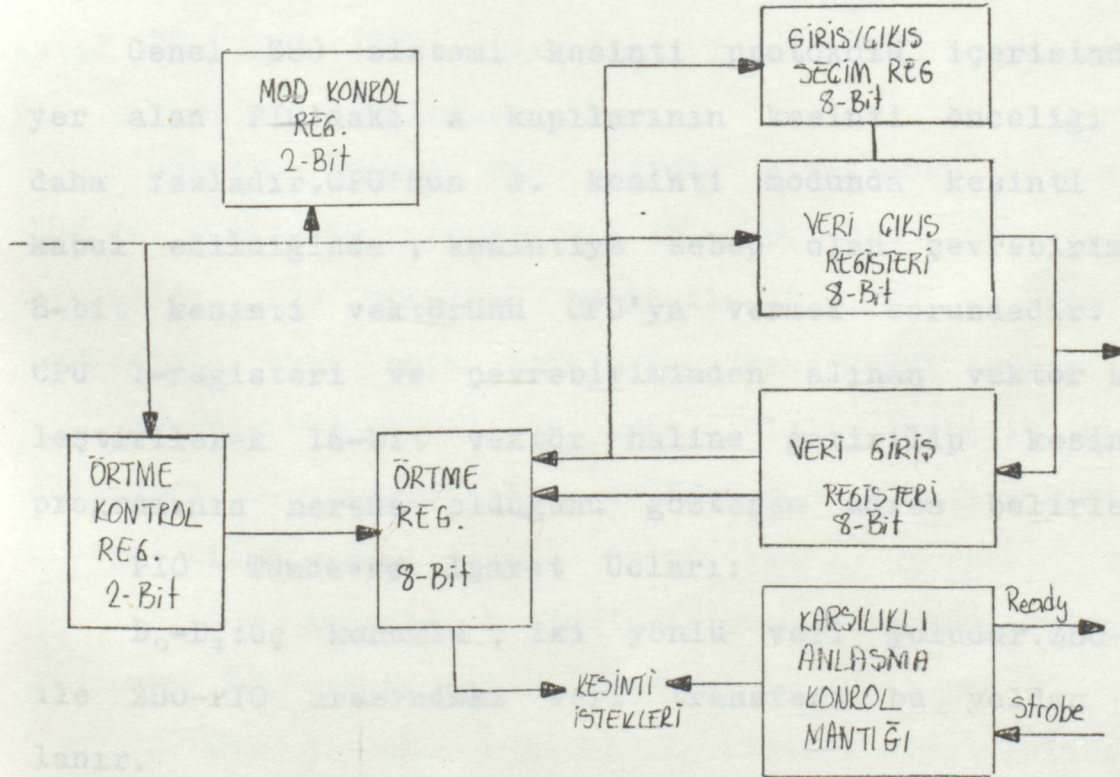
ŞEKİL-2.2de görüldüğü gibi Z80-PIO'nun iç yapısı, CPU veri ve kontrol arabirimi, dahili kontrol mantığı, paralel giriş/çıkış kapıları ve kesinti kontrol mantığından oluşmuştur. Dahili kontrol mantığı, CPU veri yolu ile A ve B paralel giriş/çıkış kapıları arasında eşzamanlılık sağlar.



ŞEKİL-2.2

Paralel giriş çıkış devresinde yer alan kapılar ŞEKİL-2.3de gösterildiği gibi , 8-bit veri giriş registeri , 8-bit veri çıkış registeri , 2-bit mod kontrol registeri , 8-bit örtme registeri , 8-bit giriş/çıkış seçme registeri ve 2-bit örtme kontrol registerinden oluşmuştur.

CPU 2-bitlik mod kontrol registerini yükleyerek istenen çalışma modunu seçer. Çevre elemanları ile CPU nun haberleşmesi veri giriş/çıkış registerleri üzerinden olur. Karşılıklı anlaşma yolları sayesinde , çevre birimine bağlı çevre elemanlarının veri iletişim kontrolü sağlanır.



ŞEKİL-2.3

A kapılarına ait registerler seçilmiş olur. Genellikle 8-bitlik örtme registeri ve 8-bitlik giriş/çıkış seçim registerleri sadece bit kontrol modunda kullanılır. Bu modda, sekiz çevrebirimi kapısının herbiri giriş veya çıkış olarak tanımlanabilir. Ayrıca, örtme registeri kullanılarak hangi bitlerin kesintiye sebep olup olmayacağı belirlenebilir. Örtme kontrol registeri ise kesinti işaretinin aktif olacağı mantık seviyesini belirler. Buna ilave olarak, örtme kontrol registeri sayesinde, örtülmeyen kapılara uygulanan kesinti işaretinin birinin veya hepsinin aktif olması konumlarında kesinti olayının CPU'ya aktarılması seçilebilir.

Genel Z80 sistemi kesinti protokolü içerisinde yer alan PIO'daki A kapılarının kesinti önceliği daha fazladır. CPU'nun 2. kesinti modunda kesinti kabul edildiğinde, kesintiye sebep olan çevrebirimi 8-bit kesinti vektörünü CPU'ya vermek zorundadır. CPU I-registeri ve çevrebiriminden alınan vektör birleştirilerek 16-bit vektör haline getirilip, kesinti programının nerede olduğunu gösteren adres belirlenir.

PIO Tümdevre İşaret Ucları:

D_0-D_7 : Üç konumlu, iki yönlü veri yoludur. Z80-CPU ile Z80-PIO arasındaki veri transferi bu yoldan sağlanır.

B/A SEL: "I" de aktif giriştir. PIO ve CPU arası veri transferinde "I" konumu ile B, "O" konumu ile

A kapılarına ait registerler seçilmiş olur. Genellikle A_0 adres hattı ile irtibatlanır.

$C/D\ SEL: "I"$ de aktif giriştir. CPU ile PIO arasındaki veri transferinin tipini tanımlar. CPU yazma konumunda ve $"I"$ seviyesinde seçilen kapılara ait bir emir gönderildiğini belirler. $"O"$ konumunda ise sadece veri transferinin olduğu anlaşılır. Genellikle adres yolunun A_4 biti ile birleştirilir.

$\overline{CE}: "O"$ da aktif giriştir. CPU'nun okuma veya yazma için bu çevrebirimini seçtiğini gösterir.

$\emptyset$: Sistemin saat darbeleri girişidir.

$\overline{MI}$: CPU'dan alınan $"O"$ da aktif giriştir. PIO'nun kesinti mantığı ile CPU'yu eşzamanlı kılar. Ayrıca, aktif $\overline{RD}$ veya $\overline{IORQ}$ işaretleri bulunmadan uygulanırsa PIO sıfırlanma konumuna girer.

$\overline{IORQ}$: Z80-CPU'nun aynı isimli çıkışından alınan $"O"$ da aktif bu giriş, $B/A\ SEL$, $C/D\ SEL$, $\overline{CE}$ ve $\overline{RD}$ işaretleri ile birlikte kullanılarak, CPU ve PIO arasındaki veri veya emirlerin transferini sağlar.

$\overline{RD}$: Z80-CPU'dan alınan aynı isimli işaretin uygulandığı bu girişin $"O"$ seviyesinde olması, PIO'dan CPU'ya bilgi yollanmasına neden olur.

IEI : Z80 kesinti sistemi içinde oluşturulabilen zincirin PIO için giriş ucudur.

IEO : Z80 kesinti sistemi içerisinde oluşturulabilen zincirin PIO için çıkış ucudur.

$\overline{INT}$: $"O"$ da aktif bu giriş, CPU'ya kesinti isteğini işaret eder.

A_0-A_7 ve B_0-B_7 : A ve B kapıları giriş /çıkış bilgi yoludur. Çevre elemanları ile iki yönlü iletişim sağlar.

$\overline{A\ STB}$ ve $\overline{B\ STB}$: Çevre elemanlarından alınan "0" da aktif , A veya B kapıları yetkilendirme darbesi girişidir.

A RDY ve B RDY: Kapılar ile çevre elemanları arasında veri gönderilebileceğini veya alınabileceğini gösteren "1" de aktif çıkıştır.

M1	M0	X	X	1	1	1	1
----	----	---	---	---	---	---	---

Mod 0

Lokasyon

Mod 1

D1

D0

Mod

0

0

Çıkış

0

0

1

Giriş

1

1

0

Bilgi

2

1

1

Kontrol

3

Mod 0 seçildiğinde , PIO'nun çıkış registerine yüklenen veri, çevre elemanına iletilmek üzere giriş /çıkış veri yoluyla iletilir. Çıkış registerindeki bilgi , CPU tarafından yeni bilgi yazılarak değiştirilir. Çıkış registerındaki bilgi okunmak istendiğinde CPU giriş komutunu yürütmelidir.

PIO'nun Programlanması:

PIO'nun A kapıları dört ayrı modda çalışabilir. Bunlar , çıkış modu (Mod 0) , giriş modu (Mod 1) giriş/çıkış modu (Mod 2) ve kontrol modu (Mod 3) olarak adlandırılır. B kapıları ise, mod 2 dışındaki modlarda çalıştırılabilir. Bu modları seçen kontrol kelimesi formatı aşağıda gösterilen şekilde olmalıdır:

D7	D6	D5	D4	D3	D2	D1	D0
M1	M0	X	X	1	1	1	1
Mod Kelimesi		Kullanılmaz!		Mod Kelimesinin olurgunu belirtir.			

D7	D6	MOD
0	0	Çıkış 0
0	1	Giriş 1
1	0	İki yönlü 2
1	1	Kontrol 3

Mod 0 seçilmesi ile , PIO'nun çıkış registerine yüklenen veri, çevre elemanına iletilmek üzere giriş /çıkış veri yoluna iletilir. Çıkış registerindeki bilgi , CPU tarafından yeni bilgi yazılarak değiştirilir. Çıkış registerindeki bilgi okunmak istendiğinde CPU giriş komutunu yürütmelidir.

ve böylece çevre elemanına yeni verinin hazır olduğunu anlatır. Bu sinyal , çevre elemanından STB işareti alana kadar "I" seviyesinde kalır.

Mod 1'in seçilmesi ile kapılar giriş olarak şartlandırılır. RDY sinyali , giriş registerine bilgi yollanmasını çevre elemanına bildirir. Çevre elemanı STB sinyalinin pozitif darbe tetiklemesi ile birlikte veriyi yollar. Daha sonra RDY sinyalinin aktif durumu kaldırılmış olur.

Mod 2'de , dört karşılıklı anlaşma yolları da kullanılarak iki yönde veri iletişimi sağlanır. Sadece A kapıları mod 2 için kullanılabilir. Bu moda A kapıları anlaşma yolları çıkış kontrolü ve B kapıları anlaşma yolları giriş kontrolü sağlamak üzere kullanılır.

Mod 3'ü seçen kontrol kelimesi ardından , giriş /çıkış yollarının veri akış yönünü belirleyen kontrol kelimesi formatı aşağıdaki gibi gönderilmelidir:

D7	D6	D5	D4	D3	D2	D1	D0
I/O ₇	I/O ₆	I/O ₅	I/O ₄	I/O ₃	I/O ₂	I/O ₁	I/O ₀

"I" e şartlanan bitlere karşılık gelen yollar giriş , "0" a şartlanan bitlere karşılık gelen yollar ise çıkış olarak tanımlanır.

Kesinti kontrol kelimesi, A ve B kapılarına ait registerlere aşağıdaki formatta yüklenmelidir.

D7	D6	D5	D4	D3	D2	D1	D0
Kesinti Mikadisi	VE/VEYA	"1"/"0"	Örtme şartlama	0	1	1	1

Sadece Mod 3 de
kullanılır

Kesinti Kontrol Kelimesi
Olduğunu Belirtir

Eğer D7=I yapılırsa , bu kapılar kesintiye sebep olabilir. Aksi durumda , çevre elemanlarından gelen kesinti isteği değerlendirilmez. D6 , D5 ve D4 bitleri sadece mod 3'te kullanılır.

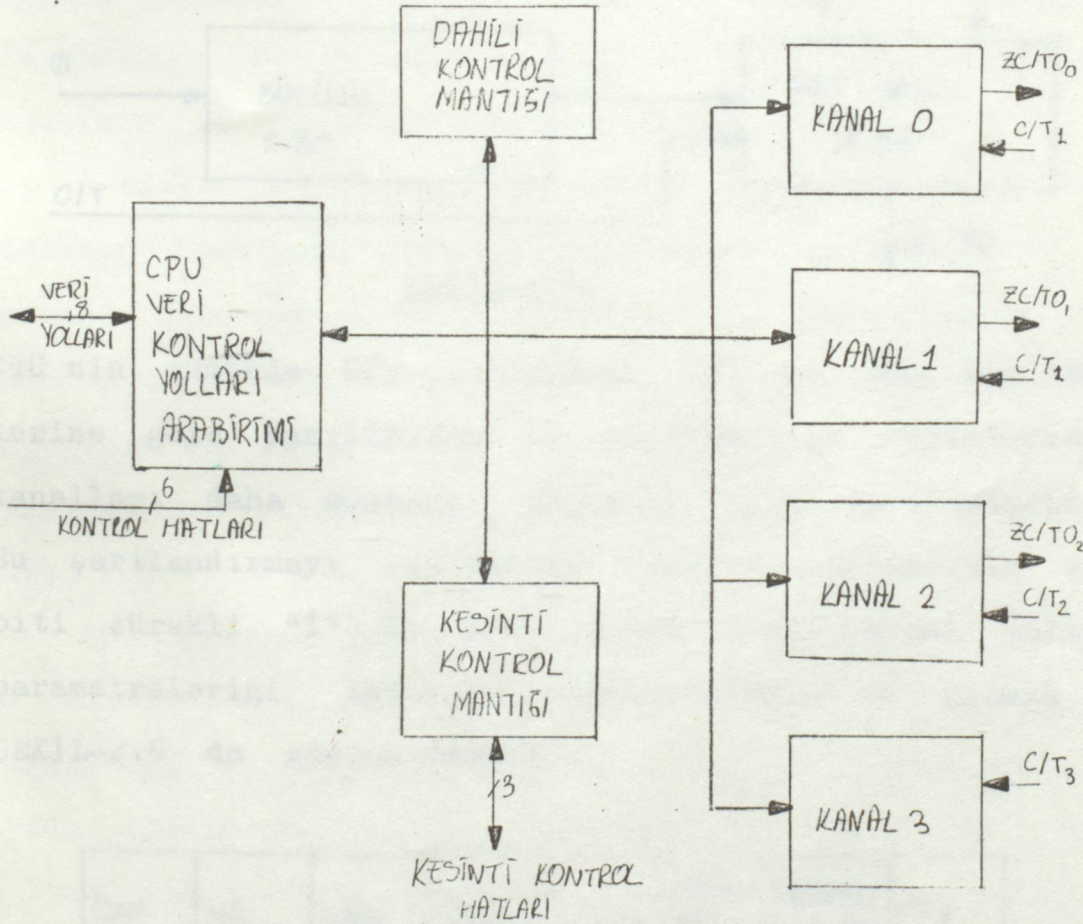
D6=I ise , kesinti isteğinin olabilmesi için seçilen bitlerin tümünün istenen seviyeye sahip olması gerekir. D6=0 durumunda ise , seçilen bitlerin birinin istenen seviyede olması kesinti isteğine neden olur.

D5=I durumunda , giriş/çıkış yolunun "I" seviyesi , D5=0 durumunda ise "0" seviyesi kesinti isteği sağlamak amacı ile izlenir.

D4=I yapılarak hangi bitlerin kesinti isteğini sağlayacakları belirlenir. Bunu sağlamak için aşağıdaki formatta örtme kelimesi gönderilir.

II-4-Z80 Sayıcı Zamanlayıcı Devresi (CTC)

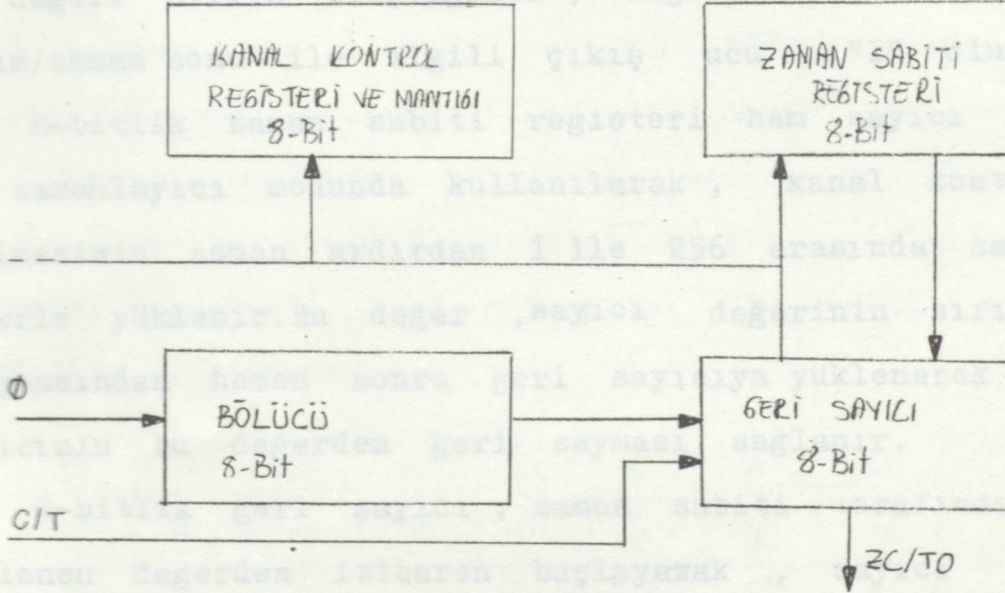
Z80 merkezi işlem birimli sistemlerde (CPU) bu zamanlayıcı ve sayıcı devresi yoluyla zamanlama ve sayma fonksiyonları yerine getirilebilir. Devre, ŞEKİL-2.4 deki gibi, Z80-CPU veri ve kontrol arabirimi, dahili kontrol mantığı, kesinti kontrol mantığı ve birbirinden bağımsız dört adet sayıcı, zamanlatıcı kanalından meydana gelmiştir.



ŞEKİL-2.4

Kanal mantığı yapısı, iki register, iki sayıcı ve kontrol mantığından oluşmuştur. ŞEKİL-2.5 gösterildi-

gi gibi , registerler 8-bitlik zaman sabiti registeri ve kanal kontrol registeridir.Sayıcılar ise , 8-bitlik okunabilir geri sayıcı ve 8-bitlik bölücüdür.



ŞEKİL-2.5

CTC nin tümünde CPU tarafından CSI ve CSO nin değerlerine göre yazılabilen bu dört kontrol registerin kanalları daha sonraki işlemler için şartlandırır. Bu şartlandırmayı sağlayacak kontrol kelimesinin son biti sürekli "1" de olup diğer 7-bit kanal çalışma parametrelerini taşır.Bu parametrelerin ne olduğu ŞEKİL-2.6 da gösterilmiştir.

Kesinti Maddesi	MOD	SAHA	EĞİM	TETİLEME	ZAMAN SABİTİ YÜCLEME	YEMİDEN BAŞLANMA	1
--------------------	-----	------	------	----------	----------------------------	---------------------	---

ŞEKİL-2.6

Sadece zamanlayıcı modunda kullanılabilen bölücü yardımı ile sistem saati (Ø) 16 ya veya 256 ya bölünerek geri sayıcının girişine uygulanır. Geri sayıcı değeri sıfıra ulaştığında, ilgili kanalın sıfır sayım/zaman sonu ile ilgili çıkış ucu "I" olur.

8-bitlik zaman sabiti registeri hem sayıcı hem de zamanlayıcı modunda kullanılarak, kanal kontrol kelimesinin hemen ardından 1 ile 256 arasında seçilen değerle yüklenir. Bu değer, sayıcı değerinin sıfıra ulaşmasından hemen sonra geri sayıcıya yüklenerek satıcının bu değerden geri sayması sağlanır.

8-bitlik geri sayıcı, zaman sabiti tarafından yüklenen değerden itibaren başlayarak, sayıcı modunda her harici saat darbesi ile zamanlayıcı modunda ise bölücünün çıkışından alınan darbe ile değerini bir azaltır. 0, 1, 2 kanallarında sıfıra ulaşıldığında, karşılık gelen ZC/TO çıkışından darbe elde edilir.

Z80 sistemi kesinti protokolünde CTC nin kesinti kontrol mantığı, sistemdeki önceliklere uygunluğu sağlar. IEI ve IEO işaret hattı, sistem protokolüne uygun zincir bağlantısı sağlar. CPU'ya en yakın zincir halkası durumundaki birim en fazla önceliğe sahiptir. CTC içinde ise bu öncelik sırası 0 dan 3 e doğrudur. CTC kanalları geri sayıcı sıfıra ulaştığında kesinti sağlamak üzere programlanabilir.

Kabul edilen kesinti isteğinden sonra , CTC kesinti kontrol mantığı CTC içindeki kesinti isteyen kanalın önceliğini belirler.Daha sonra , eğer CTC'nin EI girişi aktif ise 8-bitlik kesinti vektörü sistemin veri yolunda yer alır.Bu vektörün yüksek dereceli 5-biti, CTC şartlandırma işleminden sonra yazılmış olan değerdir.Ardından gelen iki bit , en yüksek önceliğe sahip kesinti isteğine karşılık gelecek şekilde aşağıdaki gibi kodlanarak CTC kesinti kontrol mantığını şartlandırır.

KESİNTİ VEKTÖRÜ							
D7	D6	D5	D4	D3	D2	D1	D0
V7	V6	V5	V4	V3	X	X	0
					0	0	Kanal 0
					0	1	Kanal 1
					1	0	Kanal 2
					1	1	Kanal 3

Buradaki en düşük dereceli bit sürekli "0" dır.

Bu kesinti vektörü , kesinti programları adreslerinin saklı olduğu tablonun ilgili bellek gözünü işaret eder.Vektör, düşük dereceli 8-biti gösterir.Bu sırada CPU I registerinin içeriğini okuyarak 16-bit tablo işaretleyicisinin yüksek dereceli 8-bitini bulur. Bu , işaretleyici kesinti programının bulunduğu belleğin düşük dereceli 8-bitini , işaretlenen yerin bir sonraki gözü yüksek dereceli adreslerini içeren yeri gösterir.Aşağıda kesinti başlangıç işaretleyicisinin nelerden oluştuğu gösterilmiştir.

I REGISTERİ İÇERİŞİ	ÇEVRE BİRİMİNDEN ALINAN 7 BİT	0
---------------------	----------------------------------	---

CTC Tümdevre İşaret Uçları:

DO-D7 :DO en az ağırlıklı bit olmak üzere, CPU ile CTC arasındaki veri transferinin sağlandığı iki yönlü ve üç konumlu veri yoludur.

CSI-CSO:"I" de aktif, kanalların seçimini sağlayan giriştir.Aşağıda gösterildiği gibi, dört bağımsız kanala erişimi sağlar.

	CS1	CS0
Ch 0	0	0
Ch 1	0	1
Ch 2	1	0
Ch 3	1	1

$\overline{CE}$: "0" da aktif giriştir.CTC'nin okunma veya yazılma amacı ile CPU tarafından seçildiğini gösterir.

$\emptyset$:Sistem saat darbeleri girişidir.

$\overline{MI}$: "0" da aktif giriştir.CPU'nun aynı isimli çıkışı ile birleştirilir.

$\overline{IORQ}$:CPU'nun giriş/çıkış çevrebirimleri için sağladığı aynı isimli işaretin uygulanacağı "0" da aktif giriştir.

$\overline{RD}$:CPU'dan alınan "0" da aktif giriştir.RD IORQ ve $\overline{CE}$ işaretleri ile veri, kanal kontrol registerleri ile CPU'nun haberleşmesini düzenler.CTC'ye yazma konumunda, $\overline{IORQ}$ ve $\overline{CE}$ 'nin aktif $\overline{RD}$ 'nin ise aktif

olmaması gerekir.CTC'yi okuma durumunda ise, bu üç işaretin tümünün aktif olması ile geri sayıcıdaki değer veri yoluna gönderilir.

IEI, : "I" de aktif giriştir.Diğer çevrebirimleri ile oluşturulacak kesinti sırasında, zincir şeklindeki protokolün gerçekleşmesini sağlar.

$\overline{INO}$: "0" de aktif çıkıştır.CTC kanalının kesinti yapacak şekilde programlanması durumunda , o kanaldaki sayıcının sıfıra erişmesi ile birlikte bu çıkış aktif olur.

IEO : "I" de aktif çıkıştır.IEI ile birlikte sistemin tümünde zincir protokolünü oluşturur.

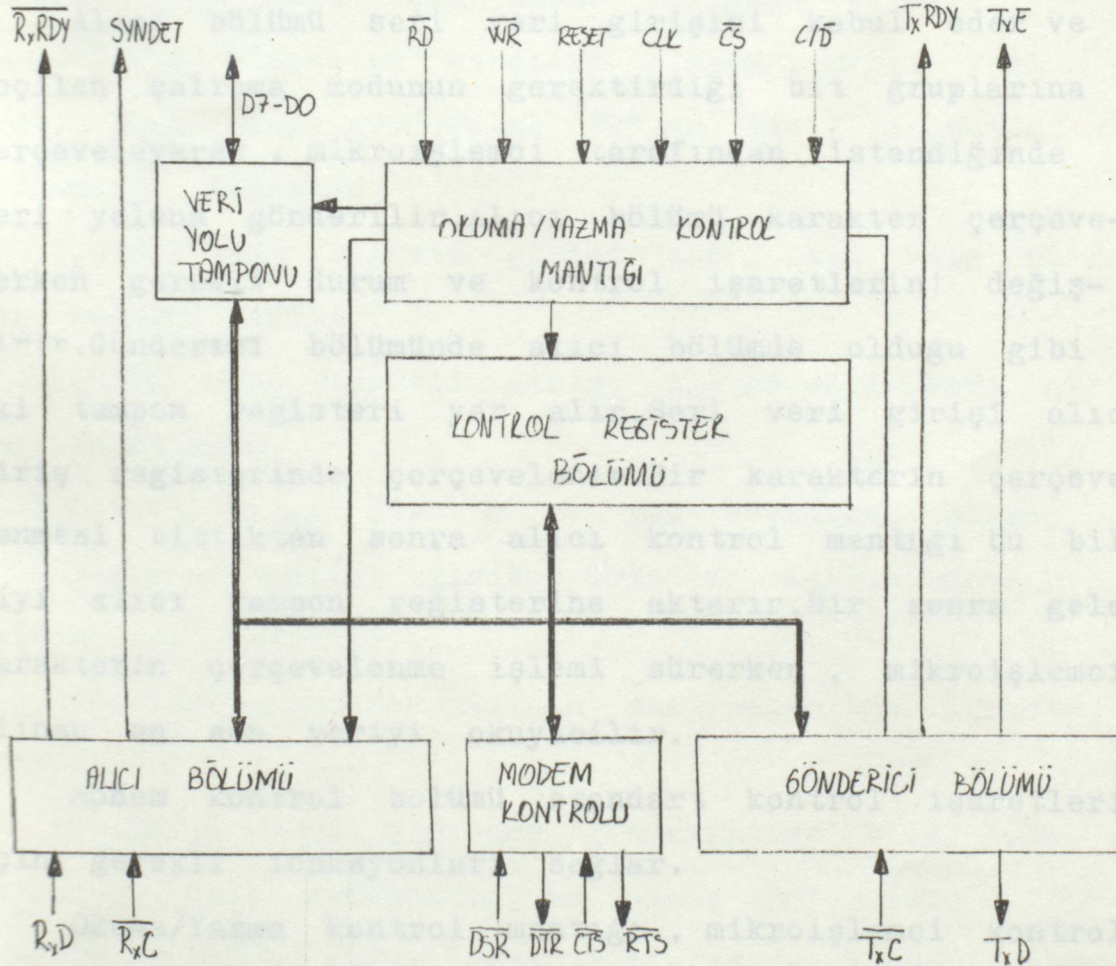
RESET : "0" da aktif giriştir.Bu girişin aktif olması ile kanal sayıcılarının sayması durur ve bütün registerler sıfırlanır.Bu anda IEO çıkışı IEI girişinin aynısıdır ve veri yolu yüksek empedans durumundadır.
CLK/TRG3-CLK/TRG0:Hangi konumda aktif olacağı programlanabilen harici saat darbeleri ve zamanlayıcı tetikleme girişleridir.

$\overline{ZC}/T02-\overline{ZC}/T00$: "0" da aktif çıkıştır.CTC'de yer alan 0,1,2 kanallarına karşılık gelmek üzere üç adettir. Sayıcı veya zamanlayıcı konumunda , geri sayıcı sıfırlandığında çıkışta "I" de aktif darbe görülür.

II-5) 825I-Programlanabilir Çok Yönlü Gönderici-Alıcı:

Aşağıdaki ŞEKİL-2.7de gösterildiği gibi , devre kontrol registerleri , gönderici/alıcı , modem kontrol ve okuma yazma bölümlerinden oluşmuştur.

Gönderici bölümü , mikroişlemci veri yolundan aldığı 8-bitlik bilgiye seri gönderilme için uygun format ilave eder.Mikroişlemci diğer paralel bilgiyi gönderébilsin diye , gönderici registeri boşaltılırken uygun kontrol işaretleri verilir.



ŞEKİL- 2.7

Gönderilmekte olan bilgi , gönderme bölümünde gönderici çıkış registeri ve gönderici tampon registeri olmak üzere , iki kere tamponlanmıştır. Veri , mikroişlemci tarafından gönderici tampon registerine yazılır. Gönderici çıkış registerindeki veri boşalır boşalmaz , gönderici tampon registeri gönderici çıkış registerine aktarılır. Böylelikle , gönderici çıkış registerinden bilgi gönderilirken mikroişlemci bir sonra gönderilmek istenen bilgiyi gönderici tampon registerine yükleyebilir.

Alıcı bölümü seri veri girişini kabul eder ve seçilen çalışma modunun gerektirdiği bit gruplarına çerçeveyeyleyerek , mikroişlemci tarafından istendiğinde veri yoluna gönderilir. Alıcı bölümü karakter çerçevelerken gerekli durum ve kontrol işaretlerini değiştirir. Gönderici bölümünde alıcı bölümde olduğu gibi iki tampon registeri yer alır. Seri veri girişi alıcı giriş registerinde çerçevelenir. Bir karakterin çerçevelenmesi bittikten sonra alıcı kontrol mantığı bu bilgiyi alıcı tampon registerine aktarır. Bir sonra gelen karakterin çerçevelenme işlemi sürerken , mikroişlemci alınan en son veriyi okuyabilir.

Modem kontrol bölümü standart kontrol işaretleri için gerekli fonksyonları sağlar.

Okuma/Yazma kontrol mantığı , mikroişlemci kontrol ve adres yolları ile arabirim işlevini görür.

825I-USART Tümdevre İşaret Ucları:

$\overline{CS}$: Bu giriş "0" uygulandığında çevrebirimi mikroişlemci tarafından seçilmiş olur.

$C/\overline{D}$: Giriş "0" iken veri registeri , "1" iken kontrol ve durum registeri adreslenmiş olur.

$\overline{WR}$: Adreslenen registre veri yolundaki bilgiyi yükler.

$\overline{DSR}$: Karşı veri birimi bu giriş "0" uygulayarak gönderme veya alma için hazır olduğunu bildirir.

$\overline{CTS}$: RTS sinyaline karşılık olarak , karşı veri birimi tarafından gönderilen "0" da aktif işaretin uygulandığı giriştir.

T_xC : Bu saat işareti girişi ile , seri veri çıkış hızı kontrol edilir.

R_xC : Bu saat işareti girişi ile de seri veri giriş hızı denetlenir.

R_xD : Seri bilgi USART'a bu yol üzerinden girer.

$\overline{RESET}$: Bu kontrol girişi USART'ın içerisindeki registerleri sıfırlar.

CLK : Mikroişlemciden alınan ana zamanlama işareti girişidir.

$\overline{DTR}$: Karşı veri birimine gönderilecek bilgi gönderilmeye hazır olduğunda , bu çıkış "0" olur.

RTS : 825I'den karşı veri birimine veri yollanırken her yeni karakterden önce bu çıkış "0" olur.

T_xRDY : Bu kontrol sinyali , gönderici tampon regis-

terinden: gönderici çıkış registerine veri transfer edilirken "1" olur.

T_xE : Bu sinyal çıkışı, gönderici çıkış registeri tamamıyla boşalınca "1" olur.

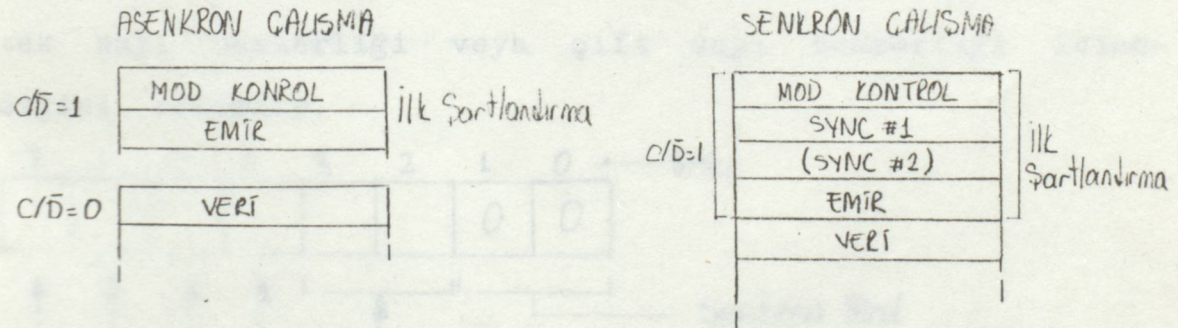
T_xD : Seri veri bu çıkıştan yollanır.

R_xRDY : Alıcı registerindeki bilgi tamamıyla gerçekleştirildiğinde bu çıkış "0" olur.

DO-D7: Mikroişlemci ile USART'ın arasında 8-bitlik paralel veri alışverişini sağlar.

İlk Şartlandırma:

USART sistem RESET işareti veya yeni giriş çıkış işlemine başlanırken şartlandırılabilir. Şekil-2.8 gibi mikroişlemci tarafından sağlanan iki, üç veya dört kelime yardımı ile şartlandırma yapılır.



ŞEKİL-2.8

Asenkron çalışmada mod kontrol kelimesini emir kelimesi takip eder. Senkron çalışmada ise, mod kontrol kelimesini bir veya iki SYNC karakteri ve ardından emir kelimesi izler.

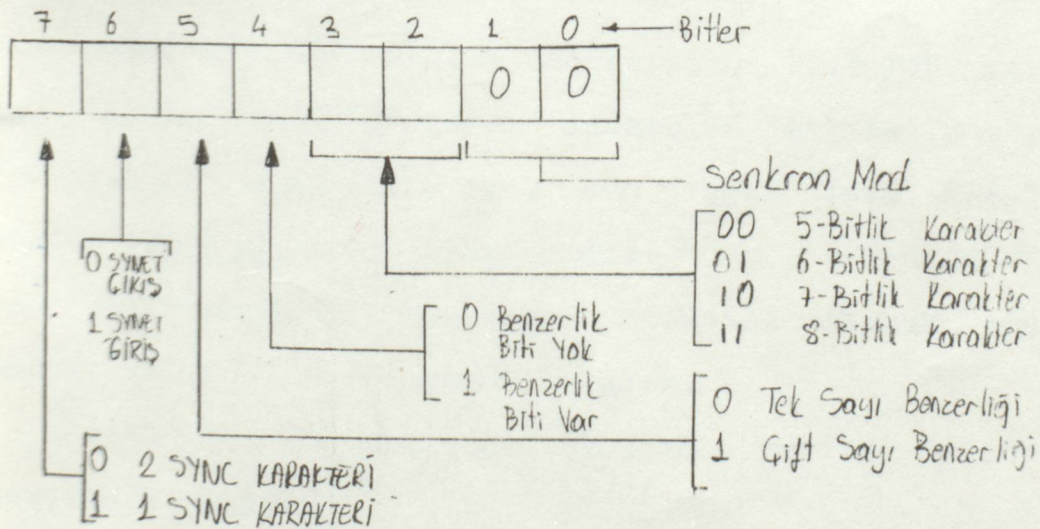
Mod Kontrol Kodları:

ŞEKİL-2.9da senkron , ŞEKİL-2.10 da asenkron mod kontrol kodları gösterilmiştir.

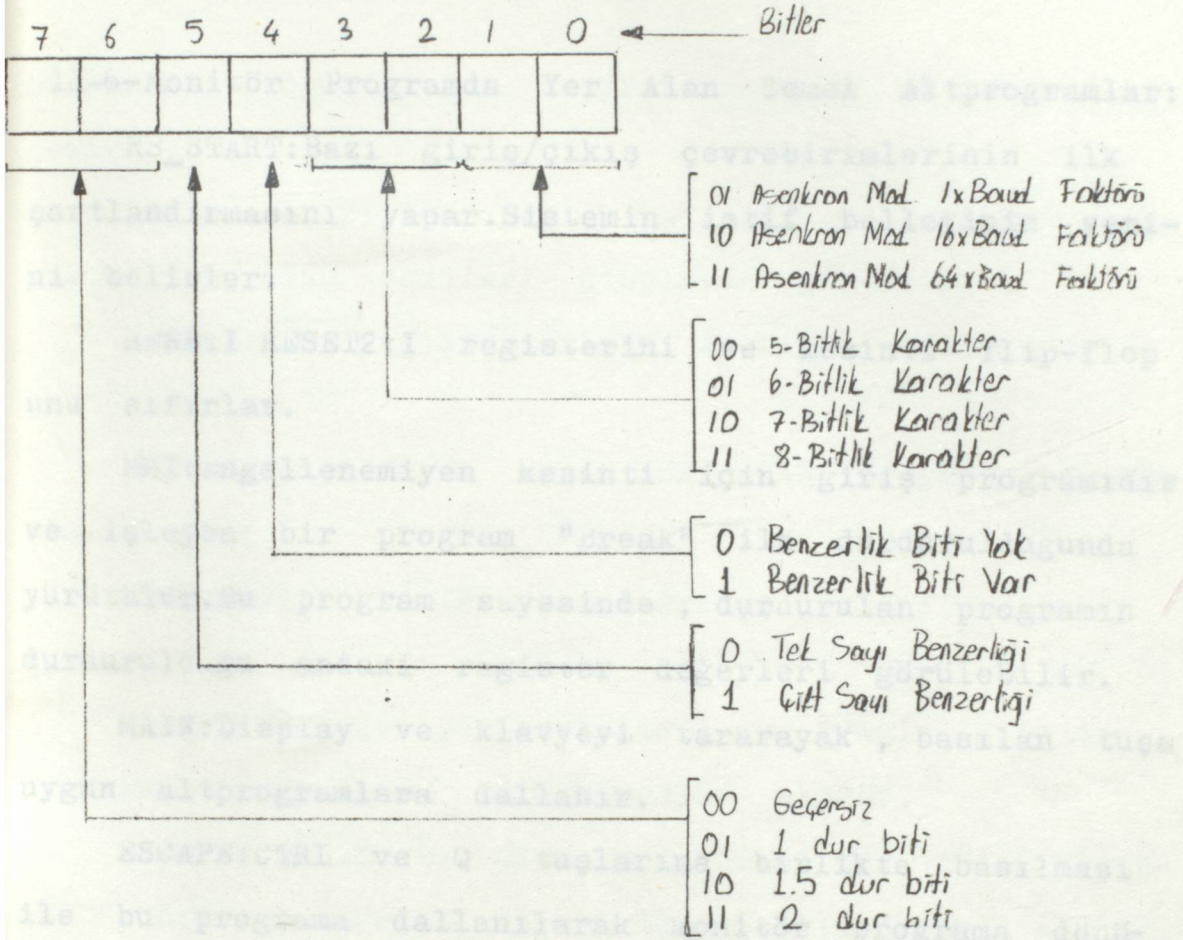
Kontrol kodunun 0. ve 1. bitleri , seçilen senkron veya asenkron çalışma modunu belirtir. Bu bitlerin sıfırdan farklı olması asenkron çalışmayı ve alıcı -gönderici saat darbeleri ile veri transferi baud hızı arasındaki oranı gösterir. Asenkron seri bilgiler saat darbesinde , her 16. saat darbesinde veya her 64. saat darbesinde gönderilip alınabilir.

Senkron ve asenkron modlarda , 2. ve 3. kontrol bitleri her veri karakterinin kaç bit olacağını tain eder.

Senkron ve asenkron modlarda 4. ve 5. bitler, benzerlik bitinin olup olmayacağını; eğer olacaksa tek sayı benzerliği veya çift sayı benzerliği istendiğini belirtir.



ŞEKİL-2.9



ŞEKİL-2.10

II-6-Monitör Programda Yer Alan Temel Altprogramlar:

RS_START:Bazı giriş/çıkış çevrebirimlerinin ilk şartlandırmasını yapar.Sistemin istifade belleğinin yerini belirler. Alan verileri displayde gösterir.

RESET1 RESET2:I registerini ve kesinti flip-flopunu sıfırlar.

NMI:Engellenemiyen kesinti için giriş programıdır ve işleyen bir program "Break" ile durdurulduğunda yürütülür.Bu program sayesinde , durdurulan programın durdurulduğu andaki register değerleri görülebilir.

MAIN:Display ve klavyeyi tarararak , basılan tuşa uygun altprogramlara dallanır.sini sağlar.

ESCAPE:CTRL ve Q tuşlarına birlikte basılması ile bu programa dallanılarak monitör programa dönlür. FILLDA:Verilen bellek gözünden ayar veri ile

KEYEXEC:Alınan tuşa göre uygun program bölümlerine gönderme altprogramıdır.asına yeni bir veri gir-

DEASM:Verilen bellek adresinden itibaren yerleşmiş makina dili programı assambler diline çevirir.

BASICZ BASICC:8 k byte ROM işgal eden MPF-IP BASIC yorumlayıcısı programıdır.ına kadar klavye ve

EDIT:Assambler programın etiketli olarak ASCII dosya halinde yazılmasını sağlar.

ASM:EDIT ile yazılan programı makina diline çevirmeyi sağlar.

LASM:Verilen bellek gözünden itibaren yazılan

assambler programı, her satırın sonunda makina diline çevirir.

MEMEXC:Belirtilen bellek adresinden itibaren bellekte yer alan verileri displayde gösterir.

REGEXC:CPU registerlerinin içeriğini displayde gösterir.

LOAD:Teyp'ten daha önce saklanmış dosyanın okunmasını sağlar.

DUMP:Teyp'e bellek dosyalarının aktarılmasını sağlar.

GOEXEC:Belleğe yerleştirilmiş programın monitör programdan çıkılarak yürütülmesini sağlar.

STEP:Yürütülmekte olan programın her emir kodundan sonra durdurarak, adım adım işletilmesini sağlar.

FILLDA:Belirtilen bellek gözlerini aynı veri ile doldurmayı sağlar.

INSET:İki veri bloğu arasına yeni bir veri girmeyi sağlar.

DELETE:Bellekten , belirtilen adresteki veriyi çıkarıp daha sonraki verileri bir geri kaydırır.

SCAN SCAN2:Bir tuşa basılınca kadar klavye ve displayi tarar.Basılan tuşun karşılığını akümülatöre yükler.

III-KULLANILAN MİKROİŞLEMCİLİ SİSTEMİN GELİŞTİRİLMESİ

Kullanılan sistemin RS-232 C hattı üzerinden terminal arabirimi gerçekleştirilerek , monitör programının daha kullanışlı hale gelmesi sağlanmıştır. Bu işlem, sistemde yer alan paralel giriş/çıkış , seri giriş/çıkış ve sayıcı/zamanlayıcı çevrebirimlerinin programlanması ile gerçekleştirilmiştir.

Bundan sonraki aşamada , mikrobilgisayarın terminal olarak programlanıp, sistemle arabiriminin sağlanması amaçlanmıştır. Böylece , kullanılan sistemin özellikleri geliştirilerek çok yönlü kullanım imkânı elde edilmiştir.

Aşağıdaki bölümlerde , seri haberleşme kavramı , terminal ve mikrobilgisayar haberleşme programı anlatılmıştır.

ve klavyeden veri okuma programı yer almaktadır. GÖRÜŞ komut satırı yukarıda anlatılan biçimde değiştirilmiştir. Buna göre :

<u>ADRES</u>	<u>KOMUT</u>
601F	JP 0E900H

şeklini almıştır.

Daha sonra anlatılacak terminal haberleşme programı ile , monitör programı içerisinde sistemin özelliklerinin terminal tarafından kullanılması sağlanmıştır.

III-1-Altprogramlarda Yapılan Değişiklikler:

Sistemde yer alan 20 karakterlik display ve standart olmayan klavye yerine , çok yönlü özellikleri olan terminallerle çalışma zorunluluğu nedeni ile monitör programda aşağıda belirtilen değişiklikler yapılmıştır.

Display'e veri gönderen ve klavyeden veri alan SCAN2 altprogramında:

<u>ADRES</u>	<u>KOMUT</u>
024F	LD HL,OFIFEH

program satırı aşağıdaki şekilde değiştirilmiştir:

<u>ADRES</u>	<u>KOMUT</u>
024F	JP OB990H

Böylece , bu altprogram çağrıldığında ,OB990 adresine atlanıp orada yer alan programın yürütülmesi sağlanacaktır.

Aynı şekilde , DEASM programı içinde display'e veri gönderme ve klavyeden veri okuma programı yer almaktadır.60IF komut satırı yukarıda anlatılan biçimde değiştirilmiştir.Buna göre :

<u>ADRES</u>	<u>KOMUT</u>
60IF	JP OE900H

şeklini almıştır.

Daha sonra anlatılacak terminal haberleşme programı ile , monitör programı içerisinde sistemin özelliklerinin terminal tarafından kullanılması sağlanmıştır.

III-2-Seri Haberleşme Kavramı:

Seri veri karakterleri , beş , altı , yedi veya sekiz uzunluğa sahip veri birimleri halinde gönderilir. Veri uzunluğunun değişken olmasının sebebi , önce-leri yapılan haberleşme kodlarının ASCII olmayışından kaynaklanmaktadır.

Seri bilgi yollama ve almada iki önemli sorun vardır. Birincisi , işaret durumunu asenkron değiştirdiği için , bir veri bitine karşılık gelen zaman aralığında örneklemenin neye göre yapılacağının bilinmeyişidir. Diğeri ise , seri bitlerin paralel bitlere doğru olarak çevrevelenmesinin sağlanmasıdır. Seri veri akışı sürerken , ard arda iki veri biti arasında , bunların tek bir karakterin üyeleri olup olmadığını belirten doğrudan bir yol yoktur.

Seri veri işaretini örneklemedeki sorun saat işareti kullanılarak giderilmiştir. Böylece , seri veri akışı sırasında bit seviyeleri örnekleme noktalarında okunur. Göndermede ve almada , iki farklı frekansa sahip ve mikroişlemci saatinden ayrı olabilen , saat darbeleri kullanılarak , bu işaretlere eşlik eden veri işaretlerinin doğrulukla örneklenebilmesi sağlanır.

Veri iletiminin her iki ucunda kullanılan saat darbelerinin senkron olması gerekir. Bu eşzamanlılık ya senkron veya asenkron seri veri transfer yöntemi ile sağlanabilir.

Senkron veri iletiminde , eşlik eden saat darbelerinin her aktif geçişinde geçerli bir veri biti iletilmelidir. Her iki uçtaki saat işareti frekansı kabul edilebilir ölçülerde hatalı olarak aynı olduğu taktirde , gönderilen bu veri alıcı bölümde doğru olarak elde edilecektir.

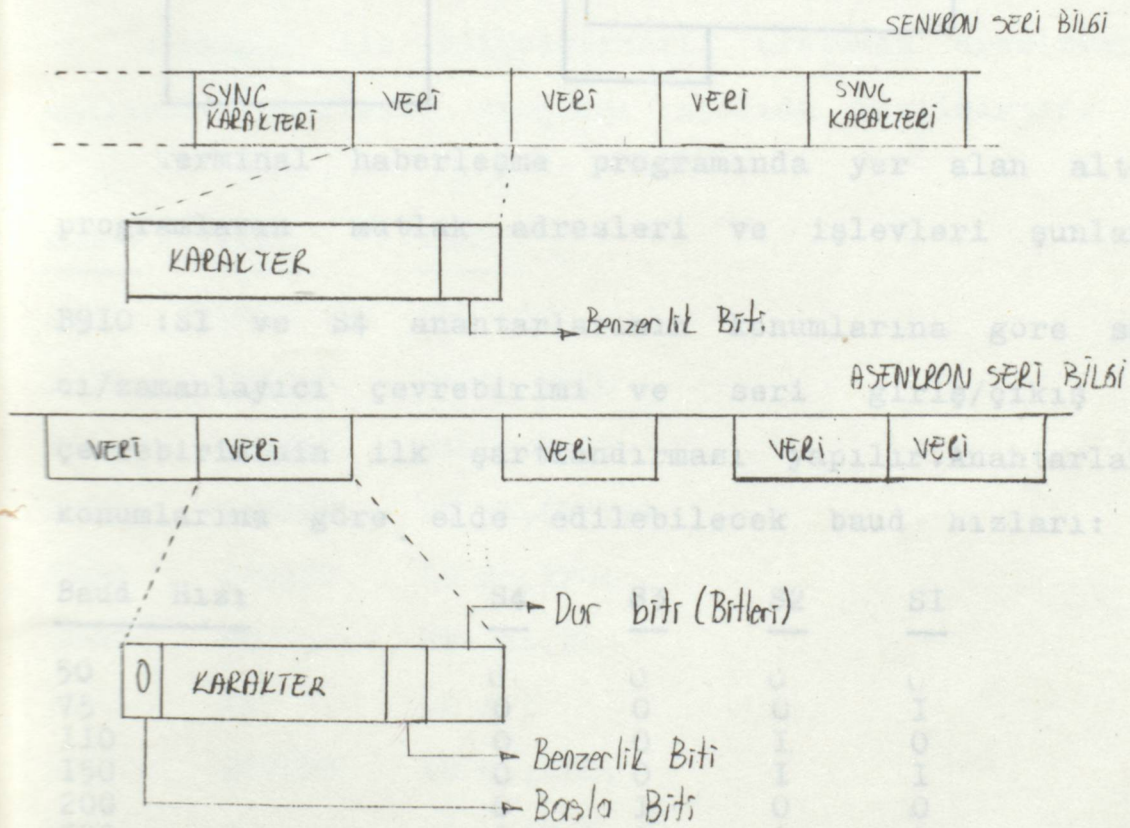
Senkron veri dizisinde , karakter sınırları seri bilginin her bloğunun peşinden gelen , daha önceden saptanmış bit dizisi ile ayrit edilir. Bu bit dizisi SYNC karakteri adını alır. Her veri bloğunun peşinde bir veya iki SYNC karakteri bulunur. Böylelikle , alıcı devre SYNC karakterini yakalayabilir ve karakter sınırlarını kurabilir. SYNC karakteri ardından gelen ilk veri biti yeni karakterin ilk bitidir.

Senkron seri veri göndermede , gönderici devre her göndermeyi saat darbelerinin aktif geçişlerinde geçerli bitlerle sağlamak zorundadır. Eğer geçerli veri hazır değilse , SYNC karakteri sürekli olarak gönderilir. Bu işlem gönderilecek geçerli veri hazır oluncaya kadar devam eder. Veri karakterlerinin alıcı bölümde birleştirilebilmesi için , seri veri alındığında SYNC karakterinin tanımlanıp ayıklanması gerekir.

Asenkron veri dizisinde , sadece mevcut olan karakterler gönderilir. Geçerli veri iletiminde , iki karakter arasında veri yolu "H" konumunda tutulur. Her asenkron veri karakteri , bir başla-biti ve 1,1.5

veya 2 dur-biti ile gerçekleştirilir. Başla-biti "0" seviyesinde, dur-biti "1" seviyesindedir. Başla ve dur-bitleri arasında 5, 6, 7 veya sekiz bitlik veri; ve eğer seçilmişse bir benzerlik biti bulunur. Seri veri gönderilmeye başlandığında, önce başla-bitinin ardından karakterin yollanması gereklidir.

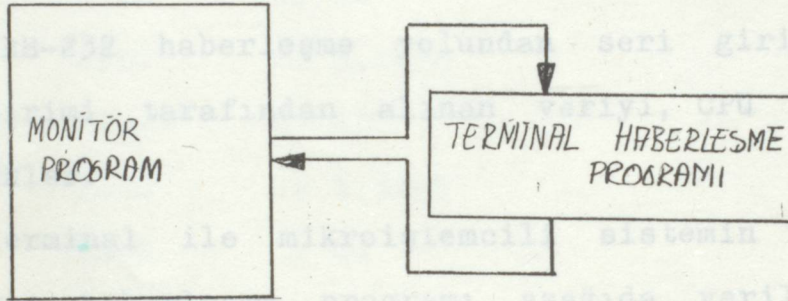
Senkron ve asenkron veri dizileri yapısı ŞEKİL-3.1 de gösterilmiştir.



ŞEKİL-3.1

III-3-Terminal Haberleşme Programı:

Sistem monitör altprogramlarında yapılan değişiklikler sayesinde , monitör program dışına çıkmak mümkün olmuştur. Böylelikle , yazılan haberleşme programına geçilerek aşağıda görülen işlem akışı sağlanmıştır.



Terminal haberleşme programında yer alan alt-programların mutlak adresleri ve işlevleri şunlardır:

B910 :SI ve S4 anahtarlarının konumlarına göre sayıcı/zamanlayıcı çevrebirimi ve seri giriş/çıkış çevrebiriminin ilk şartlandırması yapılır. Anahtarların konumlarına göre elde edilebilecek baud hızları:

Baud Hızı	S4	S3	S2	SI
50	0	0	0	0
75	0	0	0	1
110	0	0	1	0
150	0	0	1	1
200	0	1	0	0
300	0	1	0	1
600	0	1	1	0
1200	0	1	1	1
2400	1	0	0	0
4800	1	0	0	1
9600	1	0	1	0

şeklinde olacaktır.Burada , 0 kapalı , I açık anahtar konumlarını ifade eder.

B975 :Terminalin kursor satırını siler.

B373 :280 CPU'nun A registerindeki veriyi seri giriş/çıkış çevrebirimi üzerinden RS-232 haberleşme yoluna gönderir.

B36A :RS-232 haberleşme yolundan seri giriş/çıkış çevrebirimi tarafından alınan veriyi, CPU A registerine yükler.

terminal ile mikroişlemcili sistemin arabirimini sağlayan haberleşme programı aşağıda verilmiştir.

<u>ADRES</u>	<u>KOD</u>	<u>KOMUT</u>
E800	DD	POP IX
E802	CD10B9	CALL B910
E805	CD75B9	CALL B975
E808	2A82FF	LD HL,(FF82)
E80B	0104FF	LD BC,FF04
E80E	ED	SBC HL,BC
E810	45	LD B,L
E811	2104FF	LD HL,FF04
E814	7E	LD A,(HL)
E815	CD73B3	CALL B373
E818	23	INC HL
E819	10F9	DJNZ E814
E81B	CD6AB3	CALL B36A

<u>ADRES</u>	<u>KOD</u>	<u>KOMUT</u>
E81E	FE0D	CP OD
E820	2826	JR Z, E848
E822	FE08	CP 08
E824	282D	JR Z, E853
E826	FE2B	CP 2B
E828	2B2C	JR Z, E856
E82A	FE2D	CP 2D
E82C	2830	JR Z, E866
E82E	FE0I	CP 0I
E830	2834	JR Z, E866
E832	FE02	CP 02
E834	2830	JR Z, E866
E836	FE03	CP 03
E838	282C	JR Z, E866
E83A	FE05	CP 05
E83C	2828	JR Z, E866
E842	FE06	CP 06
E844	CCOOEA	CALL Z, EA00
E847	C9	RET
E848	CD73B3	CALL B373
E84B	3EOA	LD A, OA
E84D	CD73B3	CALL B373
E850	3EOD	LD A, OD
E852	C9	RET
E853	3E5F	LA A, 5F

<u>ADRES</u>	<u>KOD</u>	<u>KOMUT</u>
E855	09	RET
E856	3E0A	LD A,0A
E858	0D73B3	CALL B373
E85B	3E69	LD A,69
E85D	09	RET
E85E	3E0A	LD A,0A
E860	0D73B3	CALL B373
E863	3E5E	LD A,5E
E865	09	RET
E866	0685	LD B,85
E868	2175E8	LD HL,E875
E86B	7E	LD A,(HL)
E86C	0D73B3	CALL B373
E86F	23	INC HL
E870	10F9	DJNZ E86B
E872	031B	JP E81B
E900	DB6C	IN A,(6C)
E902	CB07	BIT 7,A
E904	CA00E8	JP Z,E800
E907	21F1FE	LD HL,FEF1
E906	032260	JP 6022

Programda yer alan altprogramların listesi aşağıda verilmiştir:

<u>ADRES</u>	<u>KOD</u>	<u>KOMUT</u>
B36A	DB6I	IN A, (6I)
B36C	CB0I	BIT I, A
B36E	28FA	JR Z, B36A
B370	DB60	IN A, (60)
B372	C9	RET
B373	F5	PUSH AF
B374	DB6I	IN A, (6I)
B376	CB00	BIT 0, A
B378	28F9	JR Z, B374
B37A	FI	POP AF
B37B	D360	OUT (60), A
B37D	C9	RET
B910	DB6C	IN A, (6C)
B912	E60F	AND OF
B914	7B	LD E, A
B915	FE05	CP 05
B917	3E07	LD A, 07
B919	3802	JR C, B91D
B91B	3E47	LD A, 47
B91D	D366	OUT (66), A
B91F	217EB3	LD HL, B37E
B922	1600	LD D, 00
B924	18	ADD HL, DE
B925	3E	LD A, (HL)

ADRES	KOD	KOMUT
B926	D366	OUT (66),A
B928	2I89B3	LD HL,B389
B92B	0606	LD B,06
B92D	4E	LD C,(HL)
B92E	23	INC HL
B92F	EDA3	OUTI
B93I	20FA	JP NZ,B92D
B933	C9	RET
B990	DB6C	IN A,(6C)
B992	CBO7	BIT 7,A
B994	CA00E8	JP Z,E800
B997	2IFIFE	LD HL,FEFL
B99A	C35202	JP 0252

III-4-Mikrobilgisayar Terminal Programı:

Aşağıda listesi verilen program sayesinde , CP/M 2.2 temelli mikrobilgisayarın klavye ve ekranı terminal olarak kullanılabilir. Mikrobilgisayarın TTY tanımlı RS-232 arabirimi üzerinden mikroişlemcili sisteme bağlanarak , karşılıklı veri transferi sağlanabilir.

Mikrobilgisayar işletim sisteminde bu program yürütüldükten sonra , TTY bağlı sisteme terminal olarak çalışacaktır. Program (Control-A) tuşu ile durdurularak tekrar mikrobilgisayar işletim sistemine dönülebilir.

0100	TERMINAL	ORG	100H
0001 =	CNTRLA	EQU	I
0100 AF	MAIN	XRA	A
0101 328701		STA	ECHO
0104 CD8B01		CALL	SCANOPTS
0107 CA1401		JZ	MAINI
010A 23		INX	H
010B 7E		MOV	A,M
010C FE45		CPI	'E'
010E C21401		JNZ	MAINI
0111 328701		STA	ECHO
0114 CDAC01	MAINI	CALL	GTIOBYTE
0117 328801		STA	SVOIBYTE
011A 3EFF		MVI	A,OFFH

011C 32890I
 011F CDACOI MAIN2
 0122 E6FC
 0124 F60I
 0126 CDBIOI
 0129 CDB70I
 012C 32890I
 012F B7
 0130 CA560I
 0133 3A890I
 0136 FEOI
 0138 CA560I
 013B 3A870I
 013E B7
 013F CA480I
 0142 3A890I
 0145 CDBEOI
 0148 CDACOI MAIN3
 014B E6FC
 014D CDBIOI
 0150 3A890I
 0153 CDBEOI
 0156 CDACOI MAIN4
 0159 E6FC
 015B CDBIOI
 015E CDB70I

STA CRTCHAR
 CALL GTIOBYTE
 ANI OCH
 ORI I
 CALL PTIOBYTE
 CALL DIRIN
 STA CRTCHAR
 ORA A
 JZ MAIN4
 LDA CRTCHAR
 CPI CNTRLA
 JZ MAIN4
 LDA ECHO
 ORA A
 JZ MAIN3
 LDA CRTCHAR
 CALL DIROUT
 CALL GTIOBYTE
 ANI OFCH
 CALL PTIOBYTE
 LDA CRTCHAR
 CALL DIROUT
 CALL GTOIBYTE
 ANI OFCH
 CALL PTIOBYTE
 CALL DIRIN

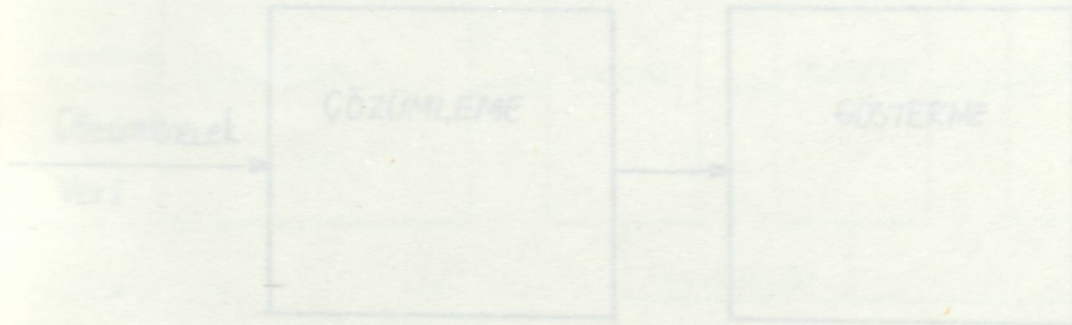
0161 328A0I	STA	TTYCHAR
0164 B7	ORA	A
0165 CA780I	JZ	MAIN5
0168 CDAC0I	CALL	GTIOBYTE
016B E6FC	ANI	OFCH
016D F60I	ORI	I
016F CDB10I	CALL	PTIOBYTE
0172 3A8A0I	LDA	TTYCHAR
0175 CDBE0I	CALL	DIROUT
0178 3A890I MAIN5	LDA	CRTCHAR
017B FE0I	CPI	CNTRLA
017D C21F0I	JNZ	MAIN2
0180 3A880I	LDA	SVIOBYTE
0183 CDB10I	CALL	PTIOBYTE
0186 C9	RET	
0187 00 ECHO	DB	\$-\$
0188 00 SVIOBYTE	DB	\$-\$
0189 00 CRTCHAR	DB	\$-\$
018A 00 TTYCHAR	DB	\$-\$
018B 218000 SCANOPTS	LXI	H,80H
018E 7E	MOV	A,M
018F B7	ORA	A
0190 C8	RZ	
0191 47	MOV	B,A
0192 3E5B	MVI	A,['
0194 23	INX	H

0I95 BE	SCANOPT1	CMP	M
0I96 C29B0I		JNZ	SCANOPT2
0I99 060I		MVI	B,I
0I9B 23	SCANOPT2	INX	H
0I9C 05		DCR	B
0I9D C2950I		JNZ	SCANOPT1
0IA0 2B		DCX	H
0IA1 06FF		MVI	B,OFFH
0IA3 BE		CMP	M
0IA4 CAA90I		JZ	SCANOPT3
0IA7 0600		MVI	B,0
0IA9 78	SCANOPT3	MOV	A,B
0IAA B7		ORA	A
0IAB C9		RET	
005 =	BDOS	EQU	5
0IAC 0E07	GTIOBYTE	MVI	C,7
0IAE C30500		JMP	BDOS
0IBI 0E08	PTIOBYTE	MVI	C,8
0IB3 5F		MOV	E,A
0IB4 C30500		JMP	BDOS
0IB7 IEFF	DIRIN	MVI	E,OFFH
0IB9 0E06		MVI	C,6
0IBB C30500		JMP	BDOS
0IBE 0E06	DIROUT	MVI	C,6
0IC0 E67F		ANI	7FH
0IC2 5F		MOV	E,A

0103 030500 KAKTER ÜRETM KAVRMI: JMP BDOS

0106 END

Sayısal sistemlerde , geçitli yollardan elde edilen verilerin değerlendirilebilmesi için , okunabilir alfa- nümerik karakterlere dönüştürülmesi gerekmektedir. Bu işlemin gerçekleştirilmesi için , Şekil-4.1'de anlatılan işlem takip edilmelidir.Özellikle verinin hangi kar- aktere karşılık geldiği düşünülmeli , ardından oluşu- lanan verinin gösterilmesi sağlanmalıdır.



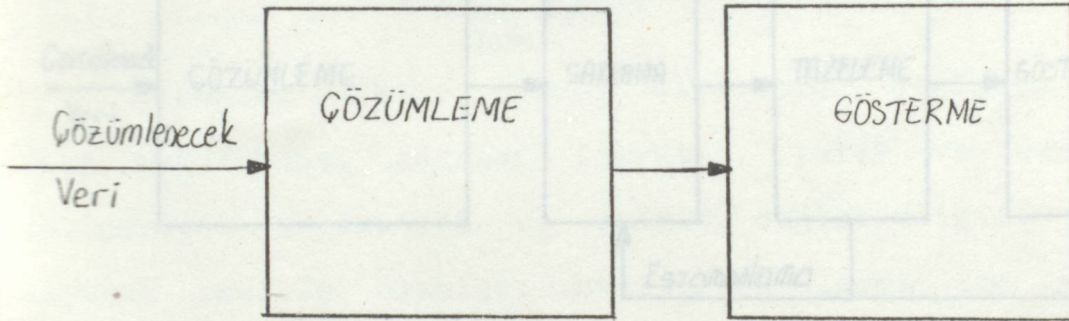
Şekil-4.1

Kullanılan displayın özelliğine göre , karakter- lere göre işleminde bazı farklılıklar oluşur. Eğer , displayın sürekli sürekli yenilenmesi gerekiyorsa , bu yenilenen verinin veya gösterilecek verinin- ni sürekli olarak göstermesi gerekir.Çözümlenecek verinin tak- rarı durumunda , her yenileme işleminde önce- ki verinin işlenmesi gerekmektedir.(Şekil-4.2)

Gösterilecek verinin saklanması durumunda taze- lenen verinin her gösterme devrinde tekrarlanması.(Şekil- 4.3)

IV-KARAKTER ÜRETME KAVRAMI:

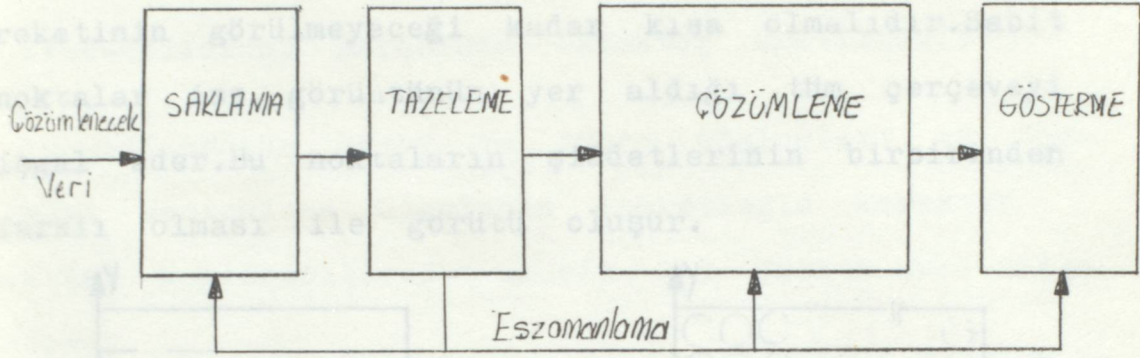
Sayısal sistemlerde , çeşitli yollardan elde edilen verilerin değerlendirilebilmesi için , okunabilir alfa-nümerik karakterlere dönüştürülmesi gerekmektedir.Bu işlemin gerçekleşebilmesi için , ŞEKİL-4.1de anlatılan işlem takip edilmelidir.Uncelikle verinin hangi karaktere karşılık geldiği çözümlenmeli , ardından çözümlenen verinin gösterilmesi sağlanmalıdır.



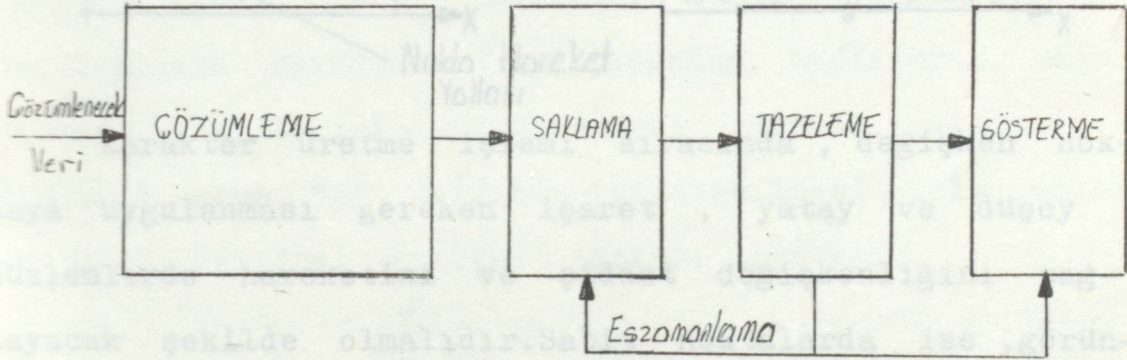
ŞEKİL-4.1

Kullanılan displayin özelliğine göre , karakter üretme işleminde bazı farklılıklar gözükür.Eğer , displayin sürekli sürekli tazelenmesi gerekiyorsa , ya çözümlenecek verinin veya gösterilecek verinin bellekte saklanması gerekir.Çözümlenecek verinin saklanması durumunda , her tazeleme işleminden önce çözümleme işlemi tekrarlanmalıdır.(ŞEKİL-4.2)

Gösterilecek verinin saklanması durumunda tazeleme işlemi her gösterme devrinde tekrarlanır.(ŞEKİL-4.3)



ŞEKİL-4.2

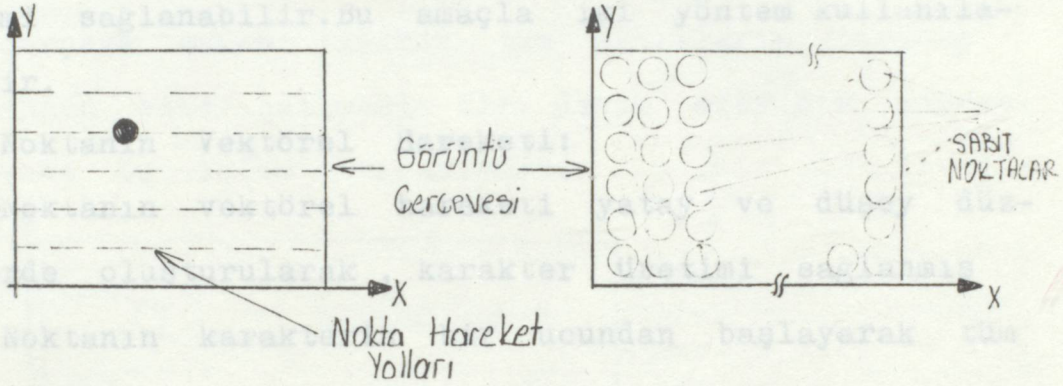


ŞEKİL-4.3

Çözümünecek verinin saklanması ile gösterme devri başlamadan önce çözümleme işlemi yapılmakta ve zaman harcanmaktadır. Gösterilecek verinin saklanması ile , daha fazla belleğe gerek duyulmaktadır.

Gösterme işlemi , hareketli nokta veya sabit noktalar ile olmak üzere iki şekilde sağlanabilir. Hareketli nokta , gösterilmesi gereken tüm görüntüyü değişken şiddette ve bir çerçeve zamanı içerisinde tarıyarak oluşturur. Bir çerçeve zamanı , noktanın ha-

rekentinin görülmeyeceği kadar kısa olmalıdır. Sabit noktalar ise görüntünün yer aldığı tüm çerçeveyi işgal eder. Bu noktaların şiddetlerinin birbirinden farklı olması ile görüntü oluşur.



Karakter üretme işlemi sırasında , değişken noktaya uygulanması gereken işaret , yatay ve düşey düzlemlerde hareketini ve şiddet değişkenliğini sağlayacak şekilde olmalıdır. Sabit noktalarda ise , görüntü her bir noktanın şiddetinin belirlenmesi ile sağlanır.

Uygulamada değişken noktanın hareketi ile katot ışınları tüpün ekranında görüntü elde edilir. Bunun sonucu olarak , osiloskop ekranında veya televizyon ekranında karakter üretmek mümkün olur. Ayrıca , matris şeklinde dizili lampa veya LED'lerle sabit noktalar elde edilir ve bu noktalara işaret uygulanması ile karakter üretimi sağlanabilir.

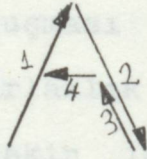
Bundan sonraki bölümlerde , osiloskop , TV ve matris LED'lerle karakter üretme yöntemleri anlatılmıştır.

IV-1-Osiloskop Ekranında Karakter Üretilmesi:

Osiloskopun yatay ve düşey düzlemleri ve parlaklık kontroluna uygulanan işaretlerle karakter üretimi sağlanabilir. Bu amaçla iki yöntem kullanılabılır.

Noktanın Vektörel Hareketi:

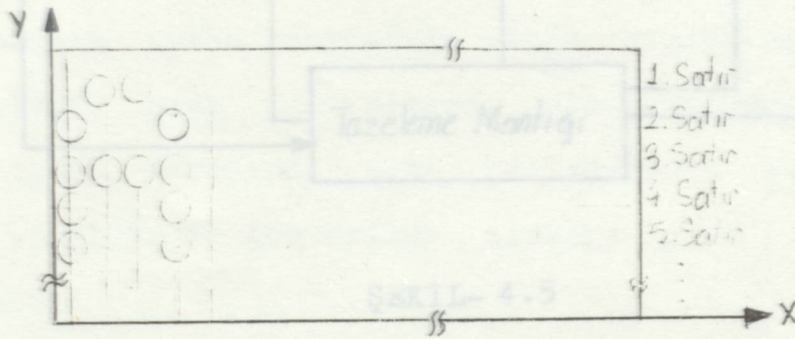
Noktanın vektörel hareketi yatay ve düşey düzlemlerde oluşturularak , karakter üretimi sağlanmış olur. Noktanın karakterin bir ucundan başlayarak tüm karakter çizgilerini takip etmesi gerekir. Aşağıda görüldüğü gibi "A" harfinin çizilebilmesi için noktanın dört vektörel hareket yapması gerekir.



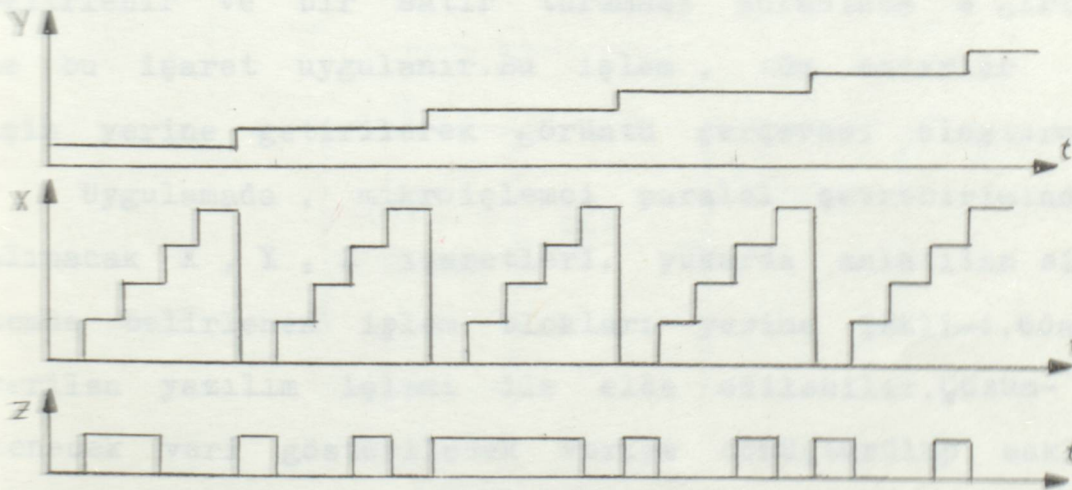
Bir sonraki karakteri çizmek için , noktanın parlaklığı minimuma indirilerek diğer karakterin başlama noktasına gelinir. Böylelikle tüm karakterler bir çerçeve zamanı içerisinde elde edilir.

Bu yöntemle bütün karakterlerin aynı parlaklıkta görülmesi zordur. Ayrıca , karakterleri oluşturan vektörel hareketin her karakter için farklı olması nedeni ile , uygulanacak işaretlerin arasında benzerlik yoktur. Bu nedenle , düzensiz bellek yerleşimine gerek duyulur.

Nokta Satır Taraması: Şekil-4.5'deki sistem kullanılarak nokta satır boyunca hareket ederken, şiddetini değiştirerek peşpeşe gelen karakterlerin aynı nokta satırını oluşturur. Bu işlem aşağıda gösterildiği gibi bir çerçeve zamanı içinde tüm satırların taranıp görüntünün elde edilmesi ile devam eder. Bir sonraki çerçevede bu işlem tekrarlanır.

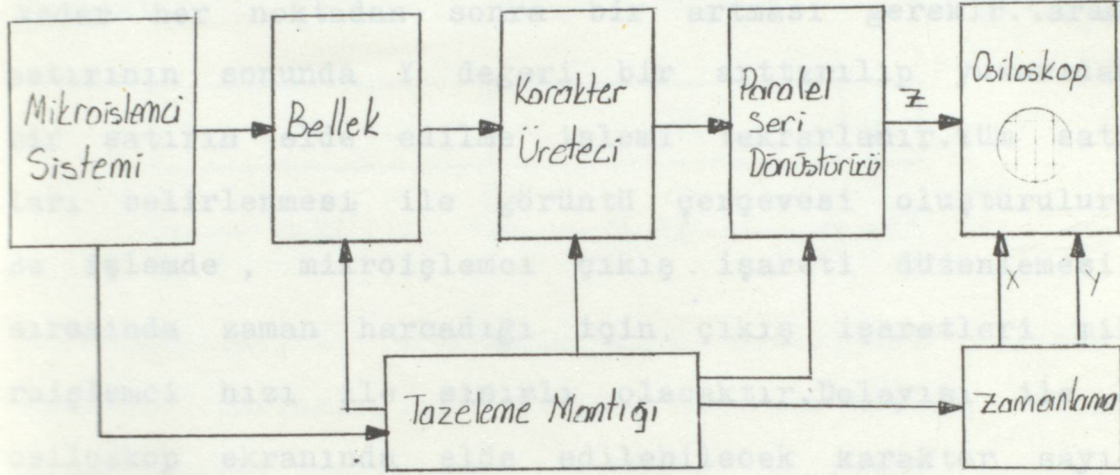


Satırların oluşması yatay X , ve dikey Y düzlemleri ile Z parlaklık işaretinin değiştirilmesi ile mümkündür. Buna ilişkin işaretler Şekil-4.4de gösterilmiştir.



ŞEKİL-4.4

Yukarıda anlatılan yöntem ŞEKİL-4.5deki sistem kullanılarak gerçekleştirilebilir.

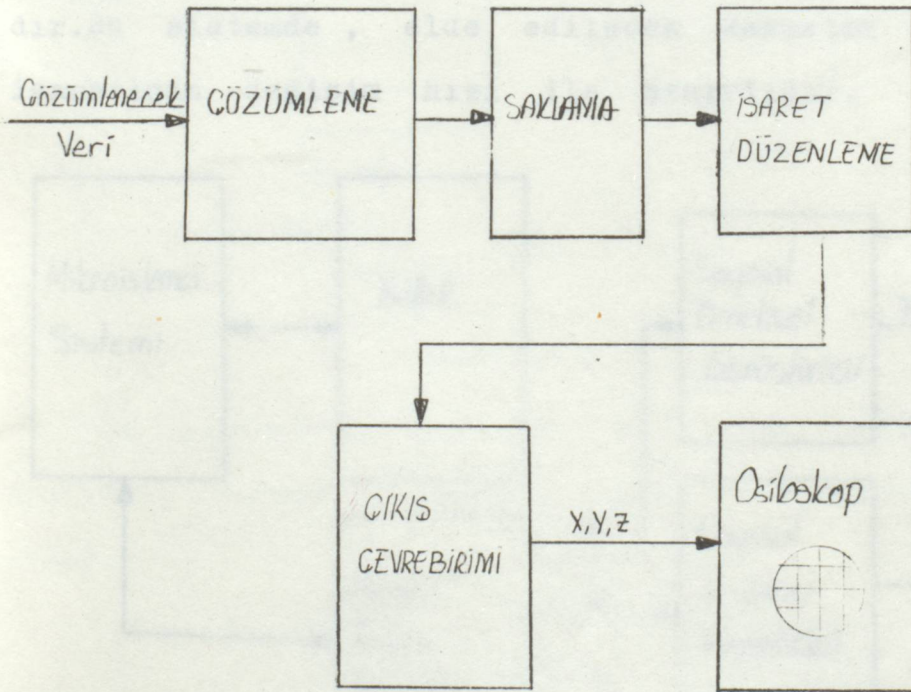


ŞEKİL- 4.5

Belleğe yerleştirilen çözümlenecek verinin karakter satırında yer alan bilgiler , gösterme mantığı tarafından belirlenip çözümleme işlemi uygulanır. Çözümleme işleminde , ilk olarak birinci nokta satırı belirlenir ve bir satır taraması süresince Z girişine bu işaret uygulanır.Bu işlem , tüm satırlar için yerine getirilerek görüntü çerçevesi oluşturulur.

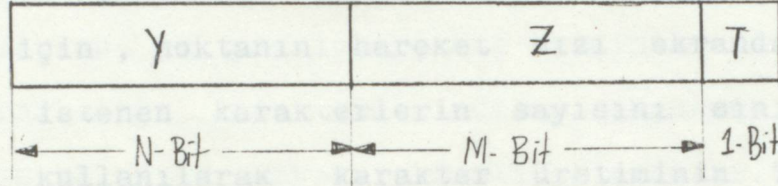
Uygulamada , mikroislemci paralel çevrebiriminden alınacak X , Y , Z işaretleri, yukarda anlatılan sistemde belirlenen işlem blokları yerine ŞEKİL-4.6da verilen yazılım işlemi ile elde edilebilir.Çözümlenecek veri gösterilecek veriye dönüştürülüp saklandıktan sonra , Y'ye uygulanacak işaretin değeri belirlenir ve çıkışa yollanır.Ardından bellekte yer alan

gösterilecek verinin nokta satırı Z'yi belirlemek üzere çıkışa yollanır. Bu sırada, A'in satır sonuna kadar her noktadan sonra bir artması gerekir. Arama satırının sonunda Y değeri bir arttırılıp yukarıdaki bir satırın elde edilme işlemi tekrarlanır. Tüm satırları belirlenmesi ile görüntü çerçevesi oluşturulur. Bu işlemde, mikroişlemci çıkış işareti düzenlemesi sırasında zaman harcadığı için, çıkış işaretleri mikroişlemci hızı ile sınırlı olacaktır. Dolayısı ile, osiloskop ekranında elde edilebilecek karakter sayısı ŞEKİL-4.7'de gösterilen sisteme göre daha az olacaktır.

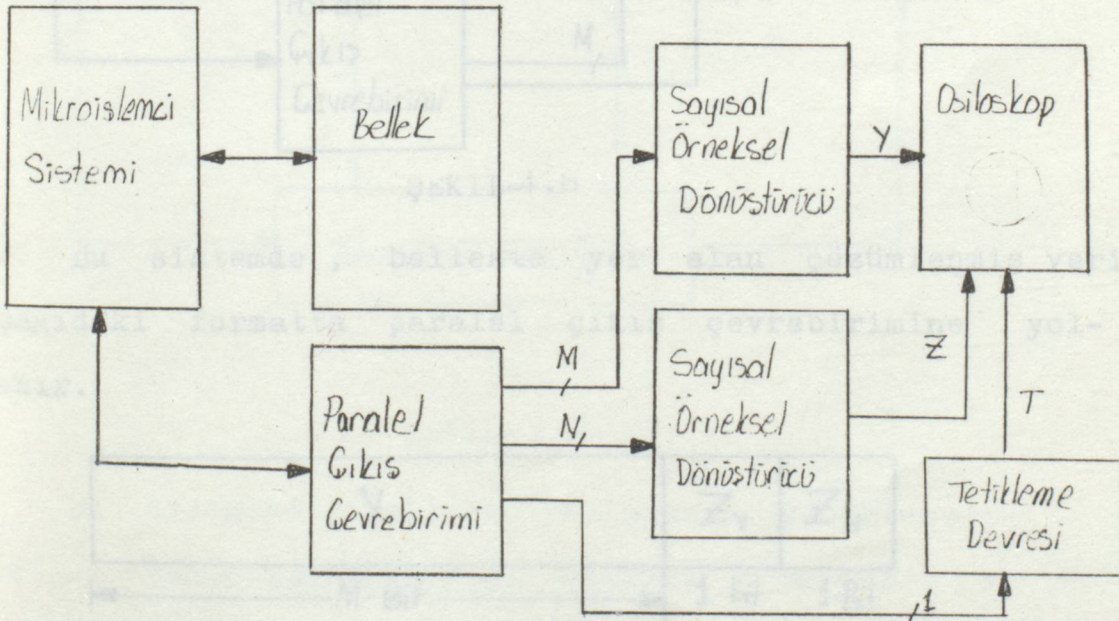


ŞEKİL-4.6

Paralel çıkış çevrebirimi çıkış kelimesi formatı ŞEKİL-4.7'de gösterilen sistemde aşağıdaki biçimde olacaktır.

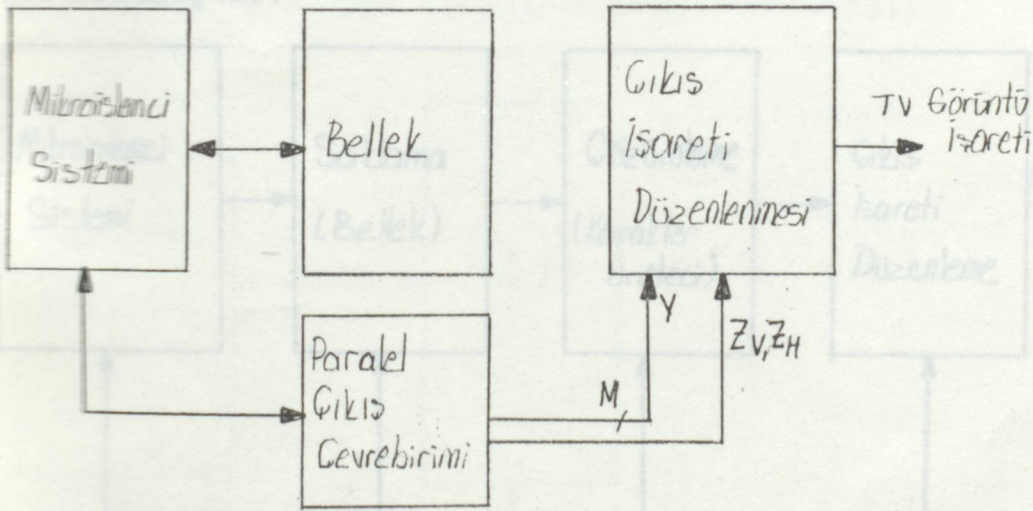


Bu formatta , Y taranmakta olan satırı , Z bulunan noktanın parlaklığını , T ise satırın sonuna gelinip gelinmediğini belirler. Osiloskopun A exeni boyunca ışın tarama işleminden faydalanılıp , her noktanın bu exendeki yerinin belirlenmesine gerek kalmamaktadır. Bu sistemde , elde edilecek karakter sayısı Z işaretinin değişim hızı ile sınırlıdır.



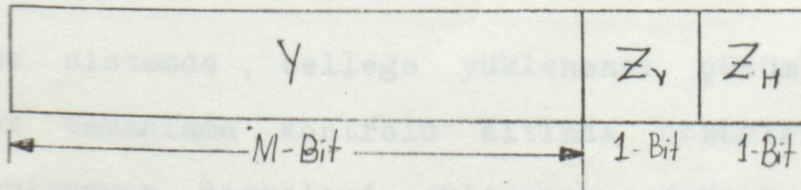
ŞEKİL-4.7

14-2-Televizyon Ekranında Karakter Üretilmesi: Çıkma-
 11. TV ekranında, hareketli noktanın satır boyunca
 taranması sırasında şiddetinin değişmesi ile görüntü
 oluşturulur. Çerçeve süresi ve satır sayısı standart
 olduğu için, noktanın hareket hızı ekranda elde
 edilmek istenen karakterlerin sayısını sınırlar. Mikro-
 işlemci kullanılarak karakter üretiminin TV ekranın-
 da nasıl sağlanabileceği ŞEKİL-4.8'deki sistemde gös-
 terilmiştir. Aynı harcanadığı sistem ŞEKİL-4.9'da



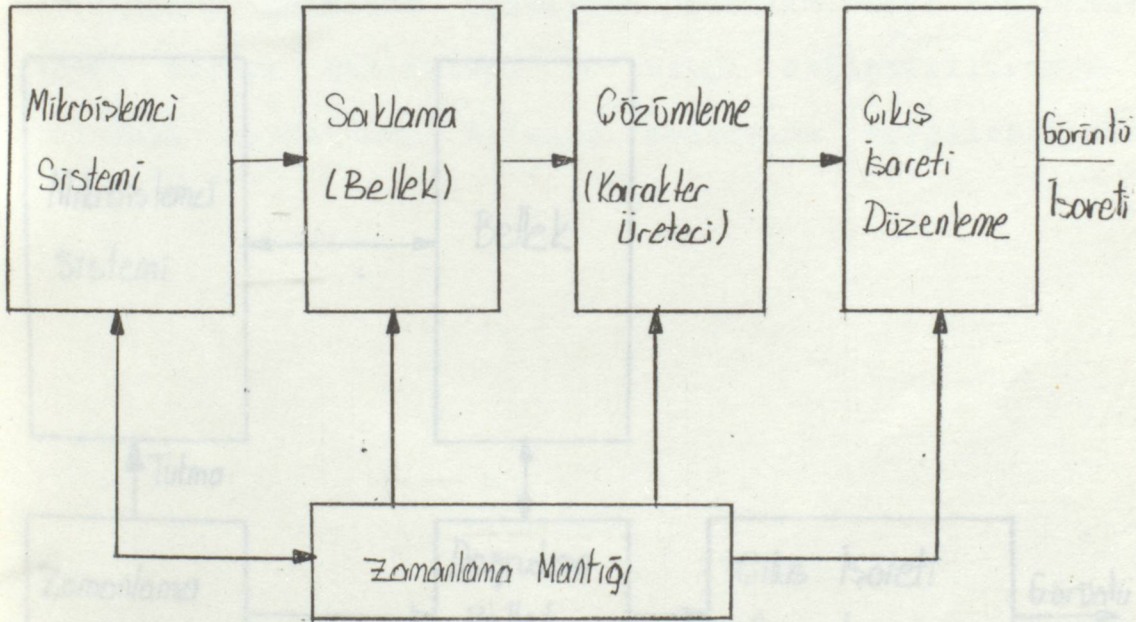
ŞEKİL-4.8

Bu sistemde, bellekte yer alan çözümlenmiş veri
 aşağıdaki formatta paralel çıkış çevrebirimine yol-
 lanır.



Bu şekildeki bilgi paketinde , Y bir satırda çıkması gereken noktaları Z_Y ve Z_H yatay ve düşey taramalar için gerekli senkronizasyon işaretinin olup olmadığını belirler. Mikroişlemci , veriyi çözümlerken , düzenlerken ve paralel çıkış birimine yollarken zaman harcadığı için elde edilen görüntü işaretinde yer alan karakter sayısı sınırlı olacaktır.

Mikroişlemcinin veri düzenleme , çözümleme ve aktarma ile zaman harcamadığı sistem ŞEKİL-4.9da gösterilmiştir.

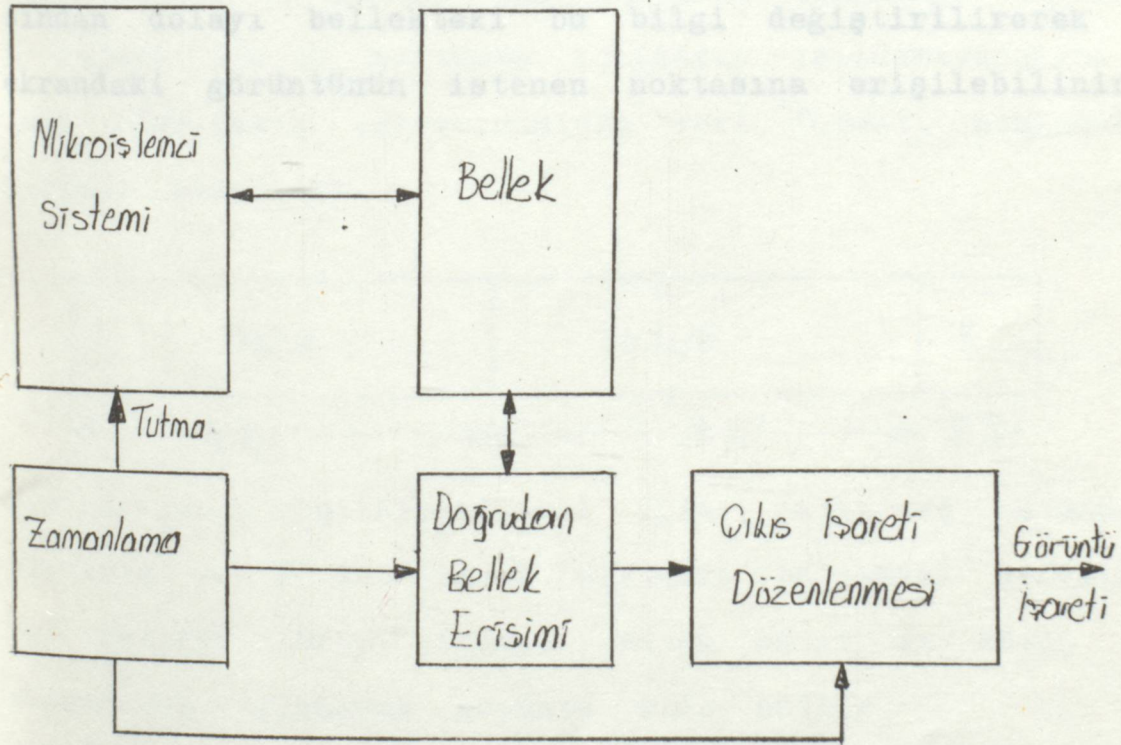


ŞEKİL- 4.9

Bu sistemde , belleğe yüklenen çözümlenecek veriler zamanlama kontrolü altında çözümlenir ve senkronizasyon darbeleri eklenerek görüntü işareti

elde edilir. Bellekte saklı bilginin her çerçeve süresince okunması görüntünün devamlılığını sağlar. Ayrıca , alan taramalarına karşılık gelen gösterilecek veriler farklı çözümlenmiş olabilir. Bu şekilde karakteri oluşturan görüntünün netliği ve keskinliği bir çerçeve zamanı içerisinde iki ayrı alan taraması yapılarak sağlanmış olur.

Mikroişlemcili sistemde donanımı kısıtlayıp , yazılıma ağırlık vererek karakter üretimi sağlanabilen diğer bir yol ŞEKİL-4.10da verilmiştir.

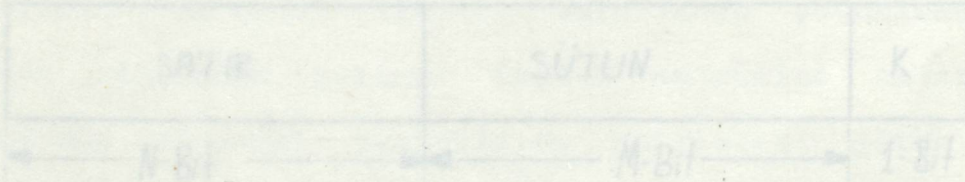


ŞEKİL- 4.10

IV-3-Matrix Noktalar ile Karakter Üretimi:

Belleğe saklanan çözümlenmiş veri , her çerçeveye süresince doğrudan bellek erişim (DMA) çevrebirimi sayesinde , bellekte yer alan veriler mikroişlemci hızından bağımsız olarak okunur.Bu okumanın hızı zamanlama devresi ile belirlenir.Okunan bilgi seri biçime dönüştürülüp senkronizasyon darbeleri eklenerek görüntü işareti elde edilir.Tek alan taraması kullanıldığı takdirde mikroişlemci diğer alan süresince tutma konumundan kurtulur.

Bu sistemin ŞEKİL-4.9da anlatılan sisteme göre üstünlüğü , çözüleme işleminin yazılım ile sağlanmasından dolayı bellekteki bu bilgi değiştirililerek ekrandaki görüntünün istenen noktasına erişilebilir.



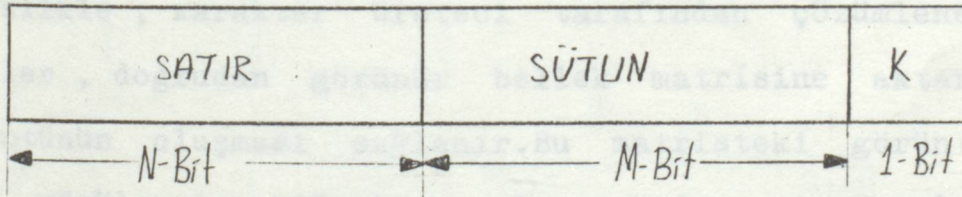
Bir noktanın belirlenebilmesi için , satır ve sütun belirlenmiş ve K noktası aktif seviyede bulunması gerekir. Bir çerçeve adresi içinde , bütün satır ve sütun elemanları taranarak görüntü elde edilir.

Ağer ŞEKİL-4.10deki sistemde her matrix elemanı son değerini koruyabiliyor ise , matrixin bellek özelliği nedeni ile çözüleme işlemine gerek kalmaz. Bu durumda , ŞEKİL-4.10da anlatılan sistem elde edilir.

IV-3-Matris Noktalar İle Karakter Üretimi:

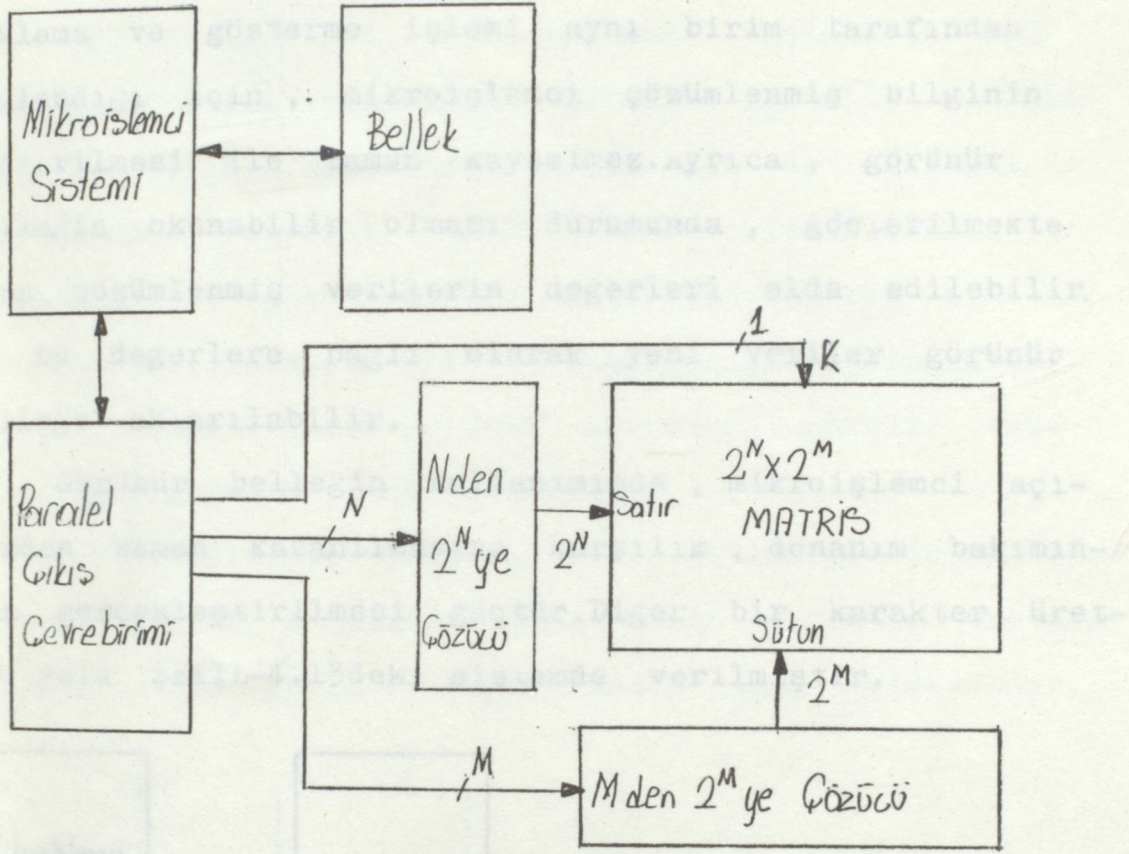
Karakteri oluşturacak noktaların çözümleme yolu ile bellekte belirlenmesinden sonra , diğer yöntemlerde olduğu gibi görünebilir duruma getirilmesi gerekir. Matrisin hangi noktalarının ışıldıyacağı bellekte yer alan bilgi seviyesi ile belirlenir. Nokta matrisi, lamba veya ışık yayan diyot (LED) kullanılarak görünebilirliği sağlanır.

ŞEKİL-4.11 deki sistemde çözümlenmiş veri , mikroişlemci çevrebiriminden N ve M'den 2^N ve 2^M çözücülere yollanır. Matristeki N değeri hareketli noktanın bulunduğu satırı M değeri ise noktanın bulunduğu sütunu K değeri ise bu noktanın ışıldayıp ışıldamıyacağını tain eder. Çıkış çevrebiriminin veri formatı aşağıdaki biçimde olmalıdır.



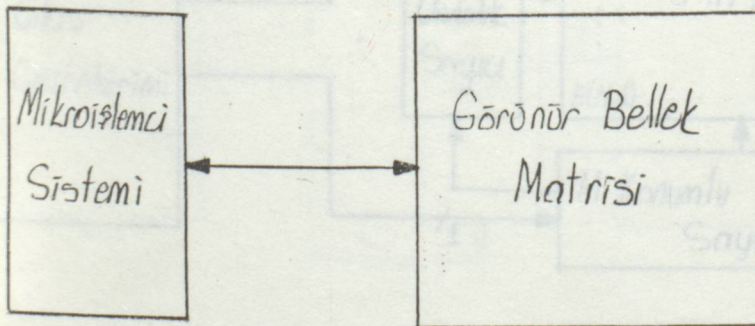
Bir noktanın ışıldıyabilmesi için , satır ve sütunun seçilmiş ve K'nin aktif seviyede bulunması gerekir. Bir çerçeve süresi içinde , bütün satır ve sütun elemanları taranarak görüntü elde edilir.

Eğer ŞEKİL-4.11deki sistemde her matris elemanı son değerini koruyabiliyor ise , matrisin bellek özelliği nedeni ile tazeleme işlemine gerek kalmaz. Bu durumda , ŞEKİL-4.12de anlatılan sistem elde edilir.



ŞEKİL-4.11

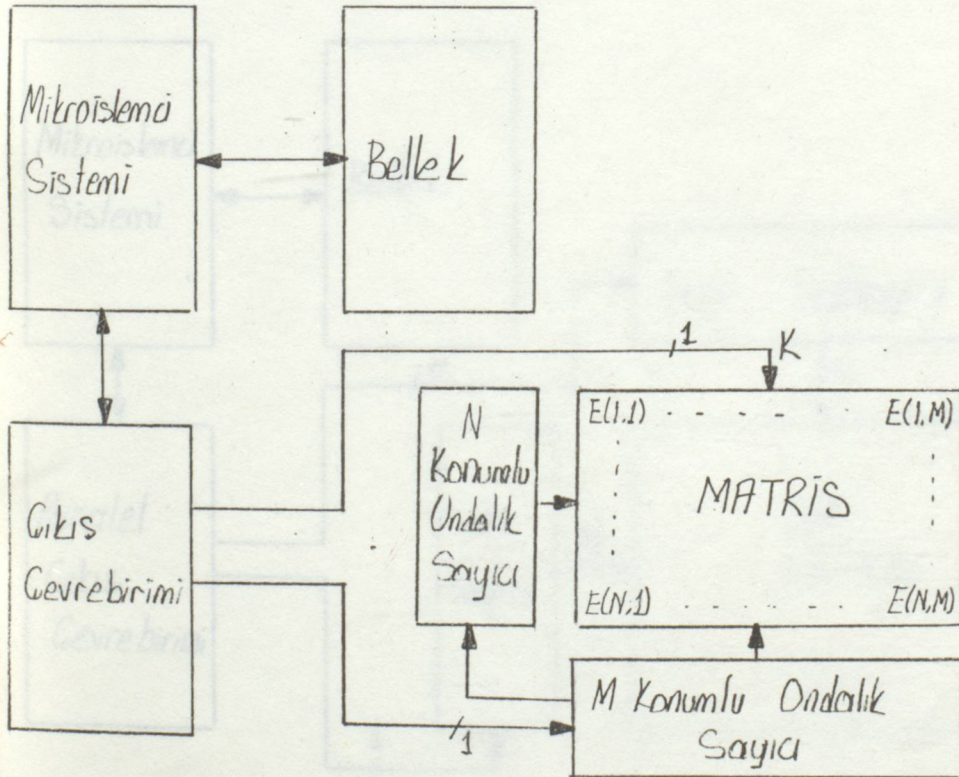
Böylelikle , karakter üretici tarafından çözümlenen veriler , doğrudan görünür bellek matrisine aktarılıp görüntünün oluşması sağlanır.Bu matristeki görüntü , yeni çözülenmiş bilgiler gelene kadar aynı kalır.



ŞEKİL-4.12

Saklama ve gösterme işlemi aynı birim tarafından sağlandığı için , mikroişlemci çözümlenmiş bilginin gösterilmesi ile zaman kaybetmez. Ayrıca , görünür belleğin okunabilir olması durumunda , gösterilmekte olan çözümlenmiş verilerin değerleri elde edilebilir, ve bu değerlere bağlı olarak yeni veriler görünür belleğe aktarılabilir.

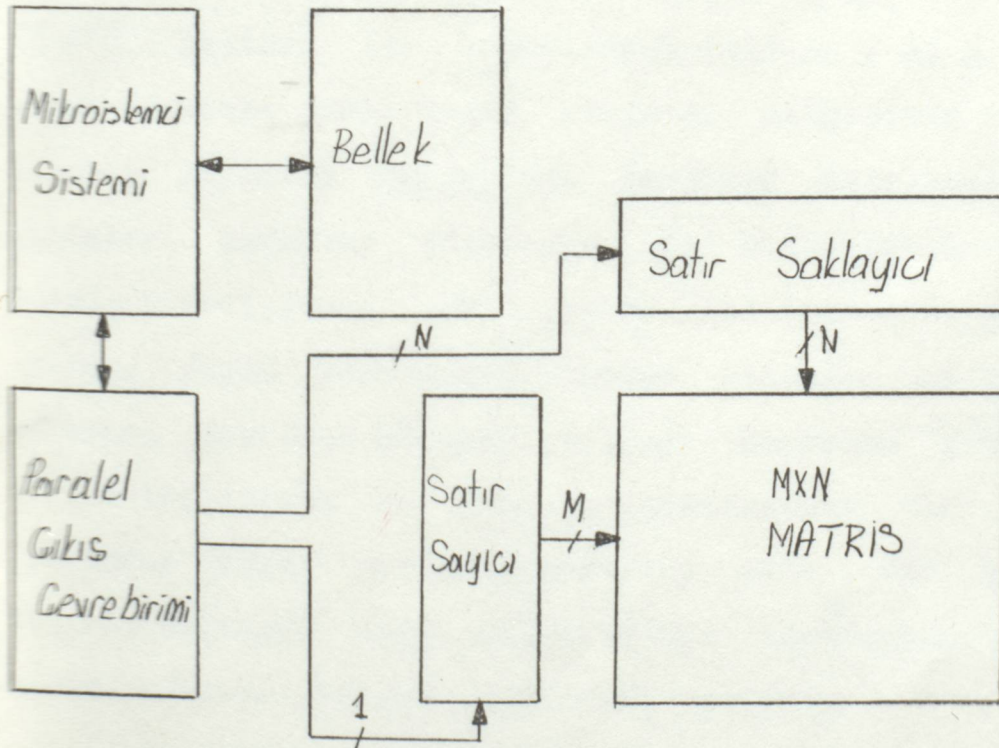
Görünür belleğin kullanımında , mikroişlemci açısından zaman kazanılmasına karşılık , donanım bakımından gerçekleştirilmesi güçtür. Diğer bir karakter üretme yolu ŞEKİL-4.13deki sistemde verilmiştir.



ŞEKİL- 4.13

Bu sistemde , I,I elemanından başlanarak I,M elemanına kadar ışıldaması gereken noktalar K ile belirlenir.Ardından, satır sayıcı bir arttırılarak bir sonraki satırda matris satır elemanlarının belirlenmesi işlemi yerine getirilir.En son N,M elemanının belirlenmesi ile çerçeve tamamlanmış olur. Böylece oluşturulan görüntü çerçevesi sürekli taze- lenerek görüntü elde edilir.

Çevrebiriminin paralel veri yüklediği düşünülürse , bir çerçeve zamanında daha fazla karakter üretile- bilmesi için ŞEKİL-4.14deki sistem uygun olmaktadır.



ŞEKİL- 4.14

Daha önce anlatılan sistemdeki gibi matris satırı sürekli arttırılarak ve o satıra ait noktalar belirlenerek görüntü çerçevesi oluşturulur. Bu sistemde aktif olan satırın tüm elemanları aynı anda belirlenir. Satırda yer alan maksimum sütun sayısı paralel çıkış çevrebirimi giriş/çıkış veri yolu genişliği ile sınırlıdır.

Şekil-3.1'de, tek bir karakter için verilen devre yapısının temelini oluşturan 4004/4009 saklayıcı devresi görülmektedir. Hangi saklayıcının seçileceği NAND kapıları ile gerçekleştirilmektedir. Bu kapı girişi, mikro-öğrenci adres yolu üzerinden alınan A3-A0 ve A1-A0 hatları ile belirlenmektedir. A0, A1 + A2 adres hatları belirlenen karakter matrisinin satır-herhangi seçimini sağlar. Bu karakter matrisinin D girişleri karakter şemaları ile birleştirilerek ortak bir D-W mikroöğrenci veri yoluyla iletilecektir. Mikro-öğrenci ya da komutunda, seçilen karakter matris satırına 8-bitlik paralel bilgiye doğrudan yerleştirilir. Birleştirilen bu veri 8 çıkışlarına yeni veri çıkışına kadar dağıtılır. Buna göre, bir karakterin belirlenmesi için mikroöğrenci tarafından toplam 8-bitlik bilgi işlemi yapılması gerekir.

Mikroöğrenci adres yolundan alınan verilerin belirli bir karakter matrisine seçilerek diğer karakterlerin çıkışı engellenebilir.

V-MİKROİŞLEMCİ BELLEĞİNDE İŞARET SENTEZİ YAPILARAK KARAKTER ÜRETİMİ:

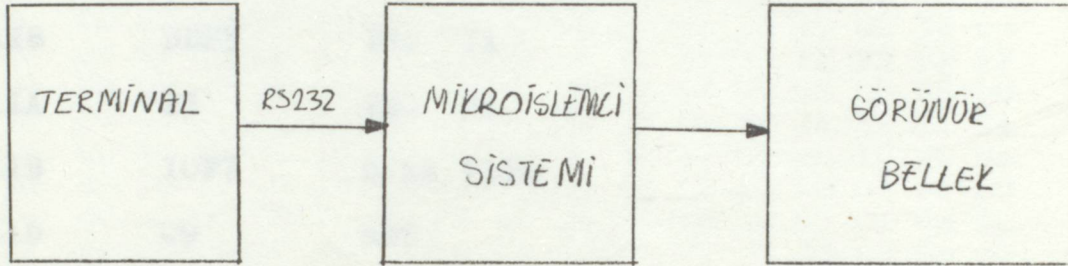
Mikroişlemcili sistemin bellek haritasında doğrudan yer alacak şekilde görünür bellek düzenlenmiştir. Her karakter 8x8 'lik 64 nokta matrisinden oluşmaktadır. Bu nedenle , gösterilecek karakter 8-byte' lık yer işgal etmektedir.

ŞEKİL-5.1 de , tek bir karakter için verilen devrenin temelini MCI40099 saklayıcı tümleşik devresi teşkil etmektedir. Hangi saklayıcının seçileceği NAND kapısı ile şartlandırılmaktadır. Bu kapı girişi, mikroişlemci adres yolu üzerinden alınan A_3-A_{10} ve $\overline{A_3-A_{10}}$ hatları ile belirlenmektedir. A_0 , A_1 , A_2 adres hatları belirlenen karakter matrisinin satırlarının seçimini sağlar. Tüm karakter matrisinin D girişleri karakter sütunları ile birleştirilerek ortak D_0-D_7 mikroişlemci veri yoluna irtibatlanmıştır. Mikroişlemci yazma konumunda, seçilen karakter matris satırına 8-bitlik paralel bilgiyi doğrudan yerleştirilir. Yerleştirilen bu veri Q çıkışlarını yeni veri yüklenene kadar şartlandırır. Buna göre , bir karakterin belirlenebilmesi için mikroişlemci tarafından toplam sekiz yükleme işlemi yapılması gerekir.

Mikroişlemci adres yolundan alınan verilerin farklı karakter matrisini seçmesi sağlanarak diğer karakterlerin oluşması sağlanabilir.

ŞEKİL-5.1

Sonuç olarak ŞEKİL-5.2 deki sistem oluşturularak terminalden alınan seri veri mikroişlemcili sistem tarafından çözümlenip , doğrudan görünür belleğe yerleştirilmektedir.



ŞEKİL- 5.2

Bu sistemde çözümleme işlemi yazılım yolu ile sağlanmaktadır.Çözümlemeyi sağlayacak program aşağıda verilmiştir:

ADRES	KOD	KOMUT
F100	2190E9	LD HL,B990
F103	D60E	SUB OE
F105	5F	LD E,A
F106	D8	RET C
F107	I600	LD D,00
F109	0603	LD B,03
F10B	CB23	SLA E
F10D	CB12	RL D
F10F	10FB	DJNZ F10B
F111	I9	ADD HL,DE

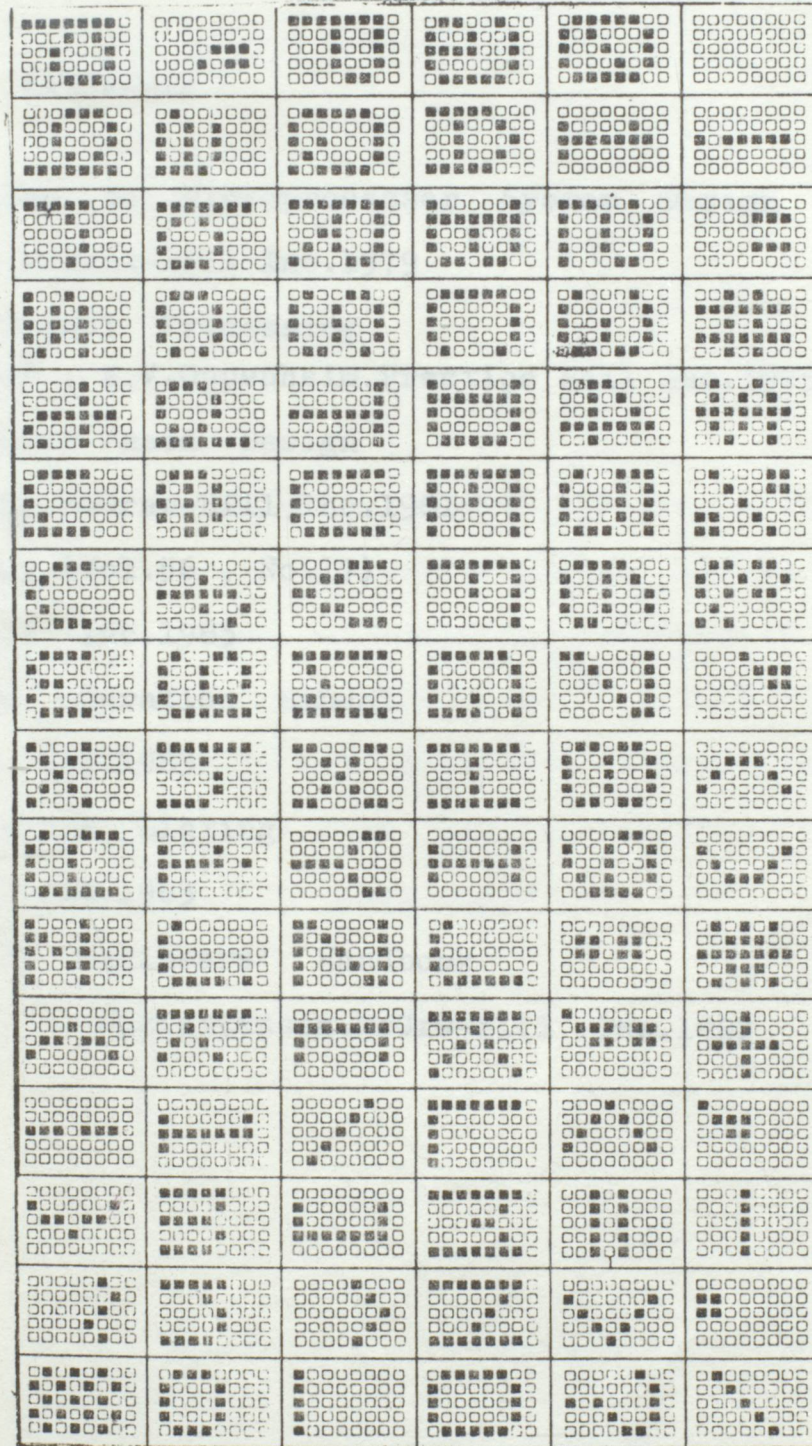
ADRES	KOD	KOMUT
FII2	0608	LD B,08
FII4	7E	LD A,(HL)
FII5	DD7700	LD (IX+00),A
FII8	DD23	INC IX
FIIA	23	INC HL
FII B	IOF7	DJNZ FII4
FII D	C9	RET

Yukarıdaki programın karakter çözümlemede kullandığı tablo aşağıda verilmiştir. Bu tablonun hangi karakterlere karşılık geldiği ŞEKİL-5.3de görülmektedir.

ADRES	VERI															
B9A0	1C	00	1C	22	22	3E	22	22	00	1C	22	20	20	2A	1C	00
B9B0	1C	00	1C	20	20	26	22	1C	08	00	1C	08	08	08	1C	00
B9C0	22	00	1C	22	22	22	22	1C	00	1C	22	20	1C	02	0A	3C
B9D0	22	00	22	22	22	22	22	1C	FF	FF	FF	FF	FF	FF	FF	FF
B9E0	00	1C	00	1C	02	1E	22	1E	00	00	1C	22	20	2A	1C	08
B9F0	18	00	1A	26	1A	02	22	1C	00	00	00	18	08	08	08	1C
BA00	00	22	00	1C	22	22	22	1C	00	00	1E	20	1C	02	3C	08
BA10	00	22	00	22	22	22	26	1A	FF	FF	FF	FF	FF	FF	FF	FF
BA20	00	00	00	00	00	00	00	00	08	08	08	08	08	00	08	00
BA30	00	14	14	14	00	00	00	00	00	14	14	3E	14	3E	14	14
BA40	00	1C	2A	18	1C	0A	3C	08	00	3C	32	04	08	10	26	06
BA50	00	10	20	28	10	2A	24	1A	00	18	18	10	20	00	00	00
BA60	00	04	08	10	10	10	08	04	00	10	08	04	00	04	08	10
BA70	00	08	2A	1C	3E	1C	2A	08	00	00	08	08	3E	08	08	00
BA80	00	00	00	00	18	18	10	20	00	00	00	00	3E	00	00	00
BA90	00	00	00	00	00	00	18	18	00	00	02	04	08	10	20	00
BAA0	00	1C	22	26	27	32	22	1C	00	01	18	08	08	08	08	1C
BAB0	00	1C	22	02	1C	20	20	3E	00	1C	22	02	00	02	22	1C
BAC0	00	04	0C	14	24	3E	04	04	00	3E	20	3C	02	02	22	1C
BAD0	00	0C	10	20	3C	22	22	1C	00	3E	02	04	08	10	20	20
BAE0	00	1C	22	22	1C	22	22	1C	00	1C	22	22	1E	02	04	18
BAF0	00	00	18	18	00	18	18	00	18	18	18	10	20	00	18	18
BB00	00	04	08	10	20	10	08	04	00	00	00	3E	00	3E	00	00
BB10	00	10	08	04	02	04	08	10	00	1C	22	02	04	08	00	08
BB20	00	1C	22	02	1A	2A	2A	1C	00	08	14	22	22	3E	22	22
BB30	00	3C	12	12	1C	12	12	3C	00	1C	22	20	20	20	22	1C
BB40	00	3C	12	12	12	12	12	3C	00	3E	20	20	38	20	20	3E
BB50	00	3E	20	20	38	20	20	20	00	1E	20	20	26	22	22	1E
BB60	00	22	22	22	3E	22	22	22	00	1C	08	08	08	08	08	1C
BB70	00	02	02	02	02	02	22	1C	00	22	24	28	30	28	24	22
BB80	00	20	20	20	20	20	20	3E	00	22	36	2A	22	22	22	22
BB90	00	22	32	2A	26	22	22	22	00	1C	22	22	22	22	22	1C
BBA0	00	3C	22	22	3C	20	20	20	00	1C	22	22	22	2A	24	1A
BBB0	00	3C	22	22	3C	28	24	22	00	1C	22	20	1C	02	22	1C
BBC0	00	3E	08	08	08	08	08	08	00	22	22	22	22	22	22	1C
BBD0	00	22	22	22	22	14	14	08	00	22	22	22	2A	36	22	22
BBE0	00	22	22	14	08	14	22	22	00	22	22	14	08	08	08	08
BBF0	00	3E	02	04	08	10	20	3E	00	1C	10	10	10	10	10	1C
BC00	00	00	20	10	08	04	02	00	00	1C	04	04	04	04	1C	00
BC10	00	08	14	22	00	00	00	00	00	00	00	00	00	00	00	3E
BC20	00	0C	0C	08	05	00	00	00	00	00	00	1C	02	1E	22	1E
BC30	00	20	20	2C	32	22	32	2C	00	00	00	1C	22	20	22	1C
BC40	00	02	02	1A	26	24	24	1A	00	00	00	1C	22	3E	20	1C
BC50	00	04	0A	08	1C	08	08	08	00	1A	26	26	1A	02	22	1C
BC60	00	20	20	2C	32	22	22	22	00	08	00	18	08	08	08	1C
BC70	00	02	00	02	02	02	22	1C	00	20	20	24	28	30	28	24
BC80	00	18	08	08	08	08	08	1C	00	00	00	34	2A	2A	2A	2A
BC90	00	00	00	2C	32	22	22	22	00	00	00	1C	22	22	22	1C

ADRES	VERI														
BCAO	00	2C	32	22	32	2C	20	20	00	1A	26	22	26	1A	02 02
BCBO	00	00	00	2C	32	20	20	20	00	00	00	1E	20	1C	02 3C
BCCO	00	08	08	3E	08	08	0A	04	00	00	00	22	22	22	26 1A
BCDO	00	00	00	22	22	22	14	08	00	00	00	22	22	2A	2A 14
BCEO	00	00	00	22	14	08	14	22	00	22	22	22	1E	02	22 1C
BCFO	00	00	00	3E	04	08	10	3E	00	04	08	08	10	08	08 04
BD00	00	08	08	08	00	08	08	08	00	10	08	08	04	08	08 10
BD10	00	2A	10	04	00	00	00	00	00	00	14	2A	14	2A	14 2A

BA20



BDLF

24KIL-5.3

KAYNAKLAR

1-Introduction to Microprocessor System Design

H.Garland.McGraw-Hill Inc.1979

2-Digital Signal Processing

R.W.Schafer , A.V.Oppenheim.Prentice Hall Inc.1975

3-Microcomputer-Based Design

J.B.Peatman.McGraw-Hill,Inc.1981

4-Electronics (Vol.56 , No.17)

McGraw-Hill Inc.1983

5-IOM-MPF-IP Operation Manual

Multitech Inc.1983

6-MPF-IP Source Listing

Multitech Inc.1983

7-Electronics Engineers' Handbook

D.G. Fink, D. Christiansen.McGrw-Hill Inc.1982

ÖZGEÇMİŞ

1962 yılında Trabzon'da doğdum. İlk ve orta öğrenimimi İstanbul'da tamamladım. Yıldız Üniversitesi Elektronik ve Haberleşme Mühendisliği Bölümünden 1983 yılında mezun oldum. Aynı yıl , Yıldız Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Haberleşme Bölümü Yüksek Lisans sınıfına girdim. Halen bu sınıfta kayıtlıyım.