

YILDIZ TEKNİK ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

Mikroişlemci Kontrollü
Storagescope ve Multimetre

Şenol Özbaş

Yüksek Lisans Tezi

23
YILDIZ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

**MİKROİŞLEMCİ KONTROLLU
STORAGESCOPE VE MULTIMETRE**

YÜKSEK LİSANS TEZİ

ELEK. VE HAB. MÜH. ŞENOL ÖZBAŞ

İSTANBUL

1985

YILDIZ ÜNİVERSİTESİ
GENEL KİTAPLIĞI

Kot : R 152
Alındığı Yer : Fen Bil. Ens. 23
Tarih : 25.3.1987
Fatura : 1000 TL.
Fiatı : 1/1
Ayniyat No : 44759
Kayıt No : 621.3
UDC : 378.242
Ek :

YILDIZ
ÜNİVERSİTESİ
KÜTÜP

YILDIZ ÜNİVERSİTESİ

D.B. No 42593

YILDIZ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

MİKROİŞLEMCİ KONTROLLU

STORAGESCOPE VE MULTIMETRE

YÜKSEK LİSANS TEZİ

ELEK. VE HAB. MÜH. ŞENOL ÖZBAŞ

İSTANBUL

1985

İÇİNDEKİLER

-Mikroişleyicilerin ve mikrobilgisayarların tanıtımı

Mikroişleyicilerin genel yapısı

Mikroişleyicilerin çevre birimleri

-6502 mikroişleyicisi

Yazılım özellikleri

Donanım özellikleri

-Mikrobilgisayarlarda giriş/çıkış kontrolü

-6522 Versatile interface adapter

-ORIC-1 mikrobilgisayarı için giriş/çıkış özellikleri

-ORIC-1 için joystick arabirimi

-ORIC-1'e analog giriş

ADC 0808 Analog/Digital dönüştürücü

Analog girişten direnç ve sıcaklık ölçümü

-ZX-81 mikrobilgisayarına analog giriş/çıkış

Yankı ,harmonizör ve storagescope uygulaması

Özet:

Bilgisayar projelerinin pekçok mükafatlarından biride bir mikrobilgisayarı dış dünyaya bağlamayla mümkündür. Bir mikrobilgisayarın sayasal dünyasını analog işaretlerin gerçek dünyasına bağlamak için bir arabirim gereklidir.

Bu arabirim bilgisayarın anlayabileceği sayısal işareti gerçelime dönüştürme yada gerilimi sayısal işarete çevirmek için kullanılır. Bu tez'de arabirimlerin genel kullanımlarını ve onların multimetre ve hafızalı osiloskop olarak kullanımını etraflıca anlattık.

Summary:

Connecting one's microcomputer to the outside world is probably one of the most rewarding computer projects. To link the digital world of the microcomputer to the real world of analogue signals an interface is required.

This interface is used either to turn digital signal into some meaningful voltage or to convert a voltage into digital signal so that the computer can interpret it. In this thesis we describe general facilities of the interfaces and it's use as multimeter and storagescope.

Mikroişleyicilerin ve Mikrobilgisayarların

Tanıtımı

Mikroişleyici (microprocessor) ,genellikle tek bir tümleşik devreye konmuş oldukça karmaşık yapıya sahip sayısal işlem elemanıdır.Yapısında ,yazmaçlar(registers), bilgi akış yolları (data paths) ve ikili kod (binary) halinde gelen komutları tanıyan ve ona göre işlem işleyen denetim birimlerinden oluşmuştur.Söz konusu ikili kod halindeki komut "makina dili" (machine language) ile yazılmış komuttur.Bu dil mikroişleyiciyi yapan firma tarafından o mikroişleyiciye özgü olarak tanımlanmıştır.

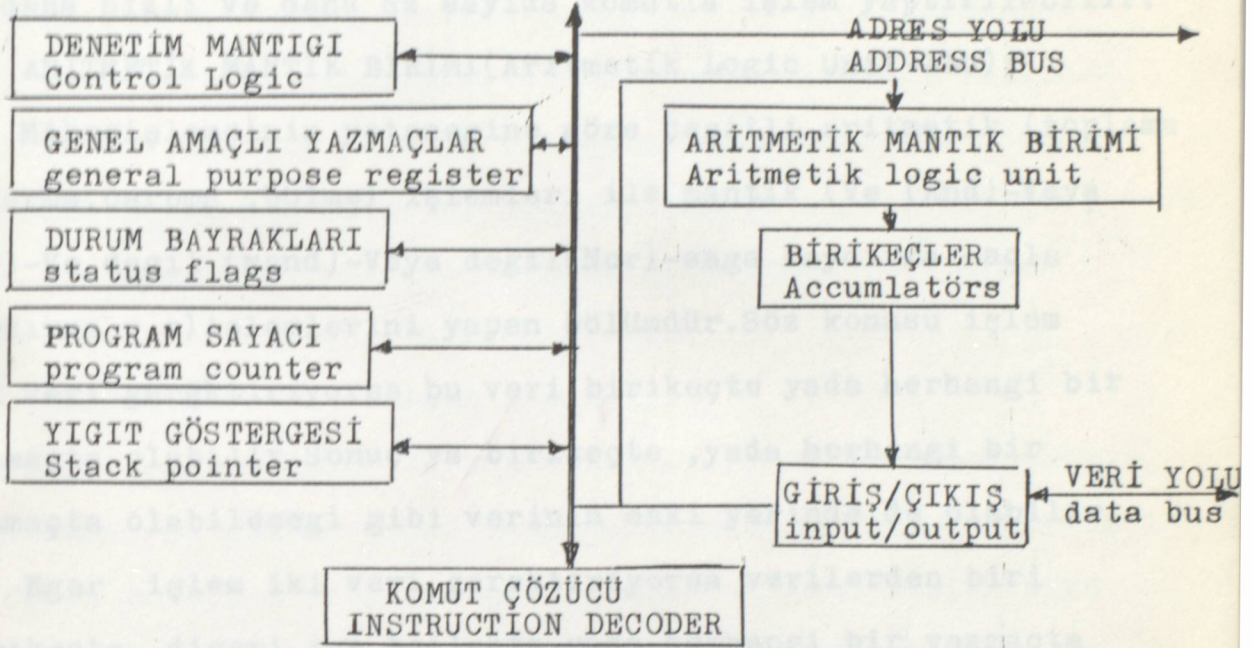
Mikroişleyiciyi taşıyan devre ,yukardakilere ilave olarak sistemin çalışmasında en önemli rolü oynayan zamanlama (timing) biriminide içerir.Bu(clock) saat darbeleri bazılarında dışarıdan verilebildiği gibi bazılarında da tümleşik devre (integrated circuit) içinde bulunur.

Mikroişleyicilerle birlikte sınırlı kapasitede yazılıp okunabilen bellek(random access memory) kısaca RAM ve yalnız okunabilir bellek (read only memory)yani ROM da bulunabilir

Mikrobilgisayarlar , içerisinde bir veya birkaç mikroişleyi ,giriş/çıkış birimlerini (input/output units) yazılıp okunabilen ve yalnızca okunabilen bellekler bulunduran bir sistemdir.Bu birimler tek tek elektronik kartlar üzerine yerleştirilmiş olabileceği gibi tek bir tümleşik devre içine yerleştirilmiş mikrobilgisayarlar (single chip microcomputer) da kullanıma sunulmaktadır.Fiatlarının her geçen gün düşmesine rağmen daha yetenekli mikroişleyiciler ve mikrobilgisayarlar tümleşik devre olarak geliştirilmektedir.

Mikroişleyicilerin Genel Yapısı

Bir miroişleyicinin ana yapısı aşağıda blok olarak gösterilmiştir.



BİRİKEÇLER (Accumulators): Bu yazmaçlar, aritmetik yada mantık işlemi yapılacak verilerden birini saklar. İşlemin sonucu gene birikeçte saklanır. Bazı mikroişleyicilerde hiç birikeç olmadığı gibi bazılarında da birkaç tane vardır.

DURUM BAYRAKLARI (Status Flags): Aritmetik yada mantık işleminin sonucuna göre konum alan 1 bit (ikilik) yazmaçlardır.

Birikeçteki işlem sonucuna göre (flag)ler 1 yada 0 olurlar. Buradaki 1 bit'lik bilgilerden bazıları sıfır (zero), negatif (negative) ki işlem sonucunun sıfır yada negatif olduğunu gösterirler, elde (carry) aritmetik işlem sonucunda elde 'yi gösterir, (overflow) taşma gibi bilgileri gösterir.

Programcı yaptırdığı işlemin sonucuna bakmadan, durum bayraklarından yararlanarak program akışını belirler. Örneğin yapılan bir işlemin sonucu sıfır ise programın bir başka bölümüne, sıfır ise bir başka bölümüne geçilebilir.

GENEL AMAÇLI YAZMAÇLAR (General Purpose Registers):

Geçici verilerin saklanması, adreslemede v.b işlemlerde kullanılabilen yazmaçlardır. Bu sayede mikroişleyiciye daha hızlı ve daha az sayıda komutla işlem yaptırılabilir.

ARİTMETİK MANTIK BİRİMİ (Aritmetik Logic Unit-ALU):

Mikroişlemcinin yeteneğine göre çeşitli aritmetik (toplama çıkarma, çarpma, bölme) işlemleri ile mantık (Ve (And)-Veya (Or)-Ve değil (Nand)-Veya değil (Nor)-sola kaydırma -sola kaydırma v.s) işlemlerini yapan bölümdür. Söz konusu işlem tek veri gerektiriyorsa bu veri birikeşte yada herhangi bir yazmaçta olabilir. Sonuç ya birikeşte, yada herhangi bir yazmaçta olabileceği gibi verinin eski yerinde de olabilir.

Eğer işlem iki veri gerektiriyorsa verilerden biri birikeşte, diğeri ise bellekte yada herhangi bir yazmaçta olabilir. Sonuç ise birikeşte elde edilebilir.

PROGRAM SAYACI (Program Counter):

Bir program bellekte ard arda işlem kodları (mikroişleyicinin tipine göre 1,2,4,8,16,32 bit'lik) olarak bulunur. Bellekteki her satır ise numaralanmıştır. Bu numaralar ADRES adını alır. Bir program, belli adresten başlayarak sıra ile işlenir. Bu sıra ise program sayacı yazmacında saklanır.

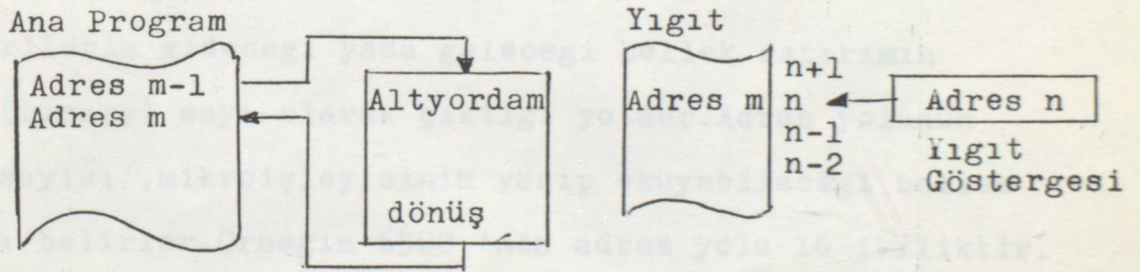
Her işlemde önce sayaç (counter) adres yoluna (address bus) 'a çıkarılır ve o adresteki komut işleyiciye alınır. Komutun tipine göre program sayacı 1 artar, yada belli bir sayıda artar/eksilir. Böylece alınmış komuttan sonra aranacak komutun adresini içerir. Alınmış komut işlendikten sonra program sayacı adres yoluna yeni adresi salar. Bu işlemler ard arda program boyunca kendiliğinden sürer.

DİZİN YAZMACI (Index register):

Bellek alanı içinde herhangi bir satıra ulaşmayı sağlayan bir adresleme yazmacıdır. Program sayacını değiştirmeden programcının istediği satırın yazılmasını veya okummasını sağlayabilir. Bazı mikroişleyicilerde dizin yazmacı bulunmaz, bazılarında ise bir yada birkaç tane bulunabilir. Bu yazmaç belleğin belli bir adresini içeriyorsa, bu değeri bir artırarak veya bir azaltarak belleğin istenen bölümlerine (veri almak veya yazmak amacıyla) ulaşılabilir.

YIĞIT GÖSTERGESİ (Stack pointer):

Yığıt, alt yordamlara (subroutines) yada kesme (interrupt) programlarına atlandığında, atlamadan önceki program sayacının değerini saklayan, böylece geri dönüş yapmayı belirlemeye imkan sağlayan bellek alanıdır. Yığıt geri dönüş adresinden başka bilgilerde içerebilir. Yığıt'ta sıranan hangi dönüş adresinde olduğunu yığıt göstergesi gösterir. Yığıt'a ilk giren bilgi yada adres en son alınır (first-in-last-out FILO)



Eğer alt programdan tekrar bir alt programa atlanılırsa yeni dönüş adresi yığıtta $n-1$ 'inci adreste saklanır ve yığıt göstergesinde $n-1$ 'inci adresin değeri bulunur.

İÇ DENETİM MANTIGI (Internal Control Logic):

Mikroişleyicinin tümleşik devre olarak en karmaşık bölümüdür. Bu bölüm, program sayacının gösterdiği adresten gelen komutun çözülmesini yapar. Komutun ne yaptırmak istediği burada ortaya çıkar.

Gerekli işlemlerin yapılması için zamanlama birimine gerekli uyarıları gönderir.İç denetim mantık alanı kombinasyonel devrelerden oluşmuştur.Zamanlama birimi ile iç denetim mantığının oluşturduğu ardışıl dizge o komutun işlenmesini ,sonucun elde edilmesini ve bir sonraki komutun çağrılması için gerekli işlemleri yaptırır.

Mikro işleyicinin çevre ile bağlantıları :

Baglantı yolları kısaca yol (bus-path) olarak anılacaktır.

Yollar sayısal imlerin gidip geldiği elektriksel bağlantılardır.Sayısal imler genellikle paralel olmakla birlikte seri (ardarda) 'da olabilir.Başlıca üç tip yol vardır.

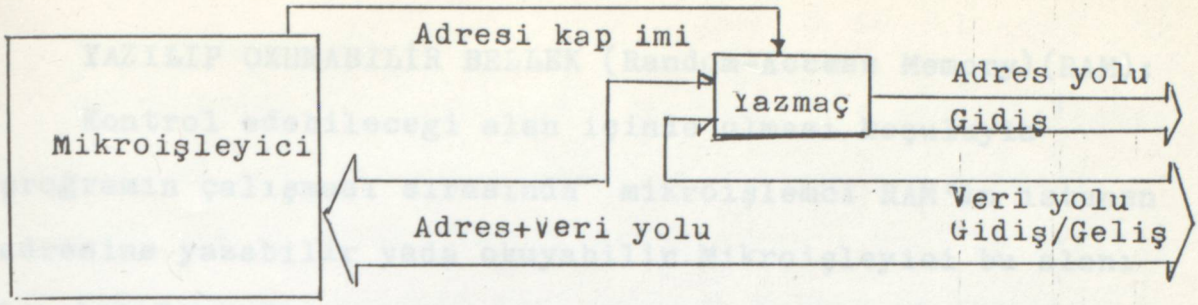
1-Veri Yolu (Data Bus):

Verilerin işleyiciye girip çıktığı yoldur.Bu yolun ikili (bit) sayısı o işleyicinin kaç ikilik olduğunu belirler Örneğin MC 6502 ,MC6800 ,Z80 8 ikilik ,TI-99 16 ikilik işleyicilerdir.

2-Adres Yolu (Address Bus):

Verilerin gideceği yada geleceği bellek satırının ikili (binary) sayı olarak çıktığı yoldur.Adres yolunun ikili sayısı ,mikroişleyicinin yazıp okuyabileceği bellek alanını belirler.Örneğin 6502 'nin adres yolu 16 ikiliktir.

Dolayısı ile 2^{16} satır adresleyebilir.Her satır 8 ikilik olmalıdır.(veri yolu için).Bazı mikroişleyicilerde adres ve veri yollarının bir bölümü veya tümü ortaktır.Bu tip mikroişleyicilerde adres ve veri yollarının kullanım olayı sıra ile yapılmaktadır.Yola önce adres salınır,bu adres adres yolunu ayırmak için bir yazmaca alınır.Ardından veri ,ortak yolda gider yada gelir.



Veri yolunun iki yönlü olmasına karşı adres yolu tek yönlüdür.

3- Denetim yolu (Control Bus):

Mikroişleyicilerin dışarıyla ve doğru olarak veri alış verişi sağlanan iki yönlü yoldur. Bu yol, mikroişleyiciye tipine göre Read/Write (yaz/oku), clock, synchronisation, interrupt (program kesme) v.s imleri taşır.

Yukarıda açıklanan üç tip bağlantı ile bir mikroişleyici bir yazılıp okunabilir bellek (RAM), bir yalnız okunabilir bellek (ROM), bir giriş/çıkış birimi (PIO) bir bilgisayarı oluşturur.

YALNIZ OKUNABİLİR BELLEK (Read-only memory) (ROM):

Mikroişleyicinin işleyeceği programların saklandığı bellek ortamıdır. Kur (RESET) işlemi ile mikroişleyici komutu ROM'dan alır, işlediği bir sonraki komuta geçer. Programı oluşturan komut dizilerinin dışında değişmez veriler tablolar halinde gene ROM içinde saklanır. Komutların ROM'dan alınması işlemi, mikroişleyicinin denetim yoluna saldırdığı imler ve adres yoluna saldırdığı adres sayesinde olur. Komut alma (Instruction fetch) mikroişleyicinin yapısından ileri gelir.

Bu olay dizisi kullanıcıyı yada programcıyı ilgilendirmez.

Kullanıcı veya programcı programın herhangi bir yolla ROM içine makine dili olarak yazılmış olmasını sağlamalıdır.

YAZILIP OKUNABİLİR BELLEK (Random-Access Memory)(RAM):

Kontrol edebileceği alan içinde olması koşuluyla programın çalışması sırasında mikroişlemci RAM'in istenen adresine yazabilir yada okuyabilir. Mikroişleyici bu alanı bir karalama defteri olarak kullanabilir, geçişi veri yada işlem sonuçlarını programın akışı içinde daha sonra kullanılmak üzere RAM'a yazılacaktır. Bunun dışında RAM büyük veri dizilerinin saklanabileceği ortamdır. Bu veri dizileri bir program olabilir. O zaman mikroişleyici uygun uyarılar verilirse bu programa atlayıp işleyebilir.

GİRİŞ/ÇIKIŞ BİRİMİ (Input/Output Unit):

Dizgenin dış dünya özellikle kullanıcı yada başka aygıtlarla bağlantısı bu birim vasıtası ile yapılır. Veriler çevre aygıtların cinsine göre çeşitli standartlara uygun olarak paralel yada seri olarak girip çıkabilir. Çevre aygıtlar yazıcı (printer), gösterici (display), çeşitli sürücü veya alıcı devreler olabilir.

6502 MİKROİŞLEYİCİSİ

Bütün küçük bilgisayarların herhangi bir programı çalıştı-
tırabilmesi için bir mikroişlemciye ihtiyaçları vardır

6502 'de bu mikroişleyicilerden biridir. 6502 ,8 ikillik
(bit'lik) gruplar halinde veri, bunları işleyecek komutları
anlayan 8 bit'lik bir mikroişleyicidir.

(8 bit'lik bir gruba 1 BYTE denir.)

6502 ,1 kilobyte 1024 byte olmak üzere 64KByte
olarak kısaca söylenebilen 65536 byte'lık bilgiye ulaşılabilir.

Böylece 64K 'lık adresleme bölgesine ulaşılabilir.

Tez çalışmasının gerçekleştirildiği ORIC-1 mikrobilgisayarı
da 6502 mikroişleyicisini kullanmaktadır. Bu bilgisayarda
64K'lık adres bölgesinin tamamı kullanılmaktadır ve bunun
en üstten 16K 'lık bölgesi Basic komutlarının çözümlendiği
makina dili programın saklandığı bölgedir. Bu program
ROM içinde bulunmaktadır. Burada saklanan (interpreter)
yorumlayıcı program Basic dilinde yazılmış program satırlarını
okuyan ve her bir satırdaki komutları düzenleyen geniş
bir makina dili programdır.

6502 'nin adresleyebildiği sahanın yorumlayıcıdan arda
kalan 3 çeyreklik (yani 48K'lık) kısmı yazılıp okunabilen
bellegi Basic programlar , makina dili programlar ve
sistem değişkenleri ile ekranı saklamak için tahsis edilmiştir.

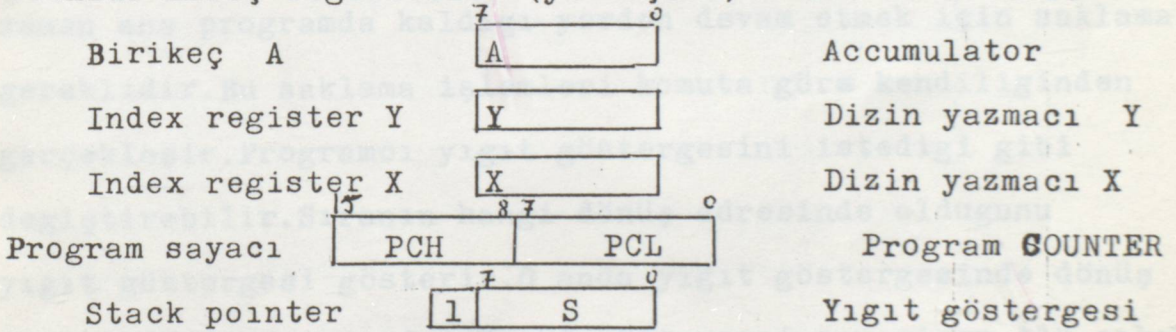
6502 Yazılım Özellikleri :

Bir mikrobilgisayarın donanımı artık standartlaşmış olmakla birlikte mikrobilgisayar yazılımları çok farklılık gösterir. Yazılım onu gerçekleştirene göre değişir. Ayrıca yazılım aracılığı ile aynı bilgisayar değişik işlerde kullanılabilir. Örneğin bir bilgisayarla matematiksel işlemler yapabileceğiniz gibi aynı bilgisayarla fiziksel büyüklüklerin otomatizasyonu ile ilgili problemlere değişik çözümler getirebilirsiniz.

Şimdi 6502 içindeki yazmaçların durumunu gözden geçirelim, 6502 ' içinde üç tür yazmaç (register) vardır.

İlk tip data transferi için kullanılır ve matematik işlemlerin sonuçlarını tutar. İkinci tipler ise bu matematik işlemlerin sonucu olarak 6502'nin dahili durumunu gösterir.

Üçüncü tip ise programın akış pozisyonunu saklar. Özel olanlar hariç registerler (yazmaçlar) 8 bit'liktirler.



A Birikeci (Accumulator A):

İşlem sonuçları bu yazmaçta elde edilebildiği gibi bu yazmaçlara alınan veriler üzerinde sağa sola kayma tümlleme v.s çeşitli işlemler yaptırılabilir. Eski veri kaybolur yerinde işlenmiş yeni veri alır. Bellekle ve giriş/çıkış birimiyle veri alış verişi bu birikeç aracılığı ile yapılabilir.

Dizin Yazmaçları (Index registers):

X ve Y olmak üzere iki dizin yazmacı olan 6502 'de bunlar bellekteki veriye ulaşılacağı zaman adrese ofset temin etmek için kullanılırlar. Aynı zamanda çevrim sayıcı olarakda kullanılabilirler.

Program sayacı (Program counter):

Çalışmakta olan programda hangi bellek satırında hangi komutun işleneceğini gösterir. Programcı program sayacını doğrudan etkileyemez. Komuta göre program sayacı işlenecek komutun adresi ile dolar ve o komutu çözümleyip işlemek üzere mikroişleyiciye aldırır. (PC) Program sayacı 6502 64K' lık adresleme kapasitesine sahip olduğundan 16 bitten adresi saklar.

Yığıt göstergesi (Stack Pointer):

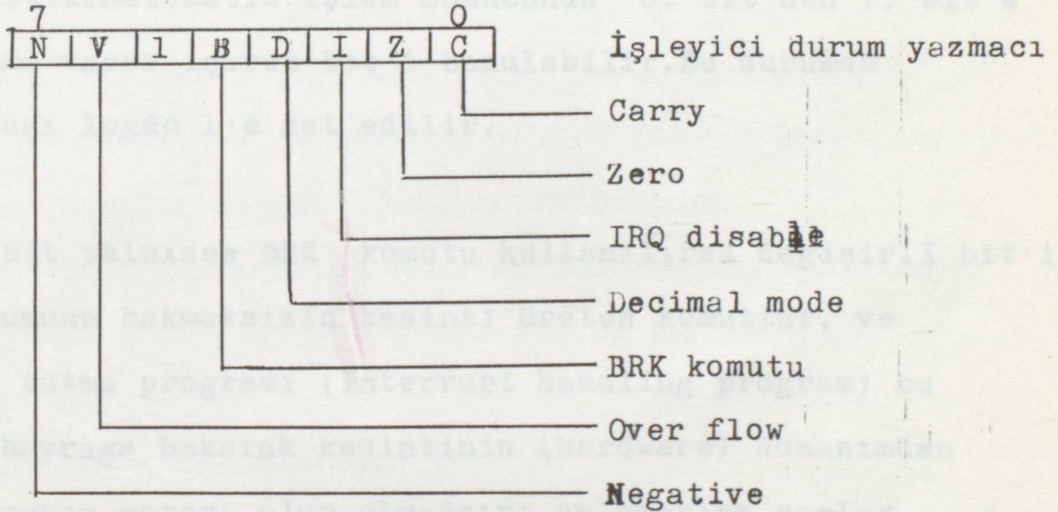
Dönüş adresinin saklandığı RAM alanıdır. Dönüş adreslerini altıyordamlara ve kesme (interrupt) programlarına atlandığı zaman ana programda kaldığı yerden devam etmek için saklamak gereklidir. Bu saklama işlemleri komuta göre kendiliğinden gerçekleşir. Programcı yığıt göstergesini istediği gibi değiştirebilir. Sıranın hangi dönüş adresinde olduğunu yığıt göstergesi gösterir. 0 anda yığıt göstergesinde dönüş adresinin adresi vardır. Yığıt göstergesi son giren ilk çıkar prensibi ile data depolamak için kullanılır. Bir ucu sabit byte listesidir. Yığıta yerleştirilen son byte listenin açık ucundan en üstte gelecek şekilde yerleştirilir. Ana kullanımı daha öncede anlatıldığı gibi program sayacını saklamaktır. Byte'lar birbir peşine sıralanarak saklanır.

(SP) yığıt göstergesi gerçekte 16 bit'lidir. Bellekte belirli adresleri işaret eder. Fakat daha az ağırlıklı 8 bitini b z değiştirebilmekteyiz. Daha ağırlıklı 8 bit'i ise önceden belirlenmiş bir bellek bölgesine ulaşılsını

(Stack) Yığıt'a bir byte yerleşeceği zaman yığıt göstergesinin içediği bir azaltılır.Yığıt doldugu zaman ise tersi olur.Bunun için Reset olduktan sonra yığıt göstergesi maksimum degeete yani FFH degerine set edilir.

(PC) program sayacı bir kesinti istegi geldiginde yada altprogram çağrıldığında program sayacının o andaki degeri stack 'e saklanır.Bu saklama sırasında PC iki byte'a ayrılır, önce biri sonra öteki yığıt'a 8 'bit'lik paketler halinde saklanır.Kesinti veya altprogramdan geri dönülürken eski program sayacının degeri yığıt'dan geri çekilir.Orjinal program kaldığı yerden devam eder.

Durum Yazmacı (Condition Code Register):



İşleyici durum yazmacında (processor status register) 8 bit'inden 7si kullanılır 5 no'lu bit yapısından dolayı 1 konumunda bulunmaktadır.Bir komutun işlenmesinden sonra sonuçla ilgili bazı özellikler durum yazmacına kendiliginden yazılır.Bu bilgilerini ve programcı tarafından koşullu sapmalarda (Conditional branch) kullanılması sağlar.Bu durumu düzenleme ve test etmek için bu bit'leri izleyen komutlar vardır.Herbir bit (ikillik) aşağıdaki gibi tanımlanmıştır.

N

Bir logic (mantık) işleminden sonra birikec'in (Accumulator) 7 'inci bit'ini kopyalar ve buna negatif flag (bayrak) denir. (Two's complement number) İkisinin tümleyeni sayılar kullanıldığında sayının işareti birikecin 7'inci bit'inde saklanır. 7'inci bit sıfırsa birikeçteki sayı pozitif , bir ise negatiftir. Sayının negatif yada pozitif olup olmadığı birikecin sadece bir bit'inin test edilmesi ile kolayca anlaşılır.

V

Birikecin 7'inci bit'i işaret bit'i olarak kullanıldığını söylemiştik. Matematik işlem sonucunda 6. bit'den 7. bit'e bir taşma varsa işaret bit'i bozulabilir. Bu durumda V bayrağı logic 1'e set edilir.

B

Bu bit yalnızca BRK komutu kullanılırsa değişir. I bit'i_ nin durumuna bakmaksızın kesinti üreten komuttur, ve kesinti tutma programı (Interrupt handling program) bu (flag) bayraga bakarak kesintinin (hardware) donanımdan yada program gereği olup olmadığını anlamasına sağlar.

D

Bu bit set edildiğinde 6502 BCD modda çalışmaya başlar.

Birikeçte toplama yada çıkarma yapılacağı zaman içerikler iki BCD sayı olarak işlem görür. Digit'ler BCD sahanın dışına çıktığında birikec'in içeriğini bu hataya karşı otomatik olarak ayarlar. D bayrağı 0 ise 6502 standart binary modda çalışır.

I

Bu bayrak IRQ girişinden gelen kesintileri örtmek için kullanılır. Bu bit 1'ise böyle kesintilere aldırmayacaktır.

I, $\emptyset$ 'a gelene kadar kesinti istegi geciktirilecektir.

IRQ, NMI yada BRK komutu I bit'ini 1'e çeker.

I bayrağı BRK komutu yadaaaNMI kesintisini yasaklamayabilir ve program kontrolu altında bu flag 1 yada $\emptyset$ yapılır.

Z

Bu bayrak logic işlem yada data (veri) transferinin sonucu sıfırsa logic 1'e çekilir. Z 'yi sıfır yada bir yapan komut yoktur. Fakat birikece sıfır yada sıfırdan farklı bir sayı yüklenerek zero flag değiştirilebilir.

C

Bu bit 'in iki farklı kullanımı vardır. Öncelikle birikeç'te aritmetik işlemlerin sonucu olarak elde yada borç olup olmadığını gösterir. (carry/borrow)

İkinci kullanımı birikeç'in döndürülmesiyada ötelenme işlemlerinde birikecin en uçtaki bitini buraya getirip saklanmasıdır.

Komut Dizisinin Kullanımı:

Mikroişleyicinin işleyeceği komutların bir listesi ekte verilmiştir. Bu listede komutların sembolik yazılımları (MNEMONIC) ile adresleme tiplerine göre komutların onaltılı (hexadecimal) kodları gösterilmiştir. Ayrıca her adresleme tipi için o komutun kaç sekizliden (byte) oluştuğu ve kaç CLOCK periyodu içinde işlendiği belirtilmiştir. Ayrıca her komutun durum yazmacında (status register) hangi bit'leri etkilediği gösterilmiştir.

Assembler (derleyici) dili programlama:

ORIC ve benzeri mikrobilgisayarlarda pekçok kişi yüksek seviyeli programlama dili olan Basic 'de program yazar.Yüksek seviyeli diller bilgisayarlarda kolay programlama yapabilmek için geliştirilmişlerdir.İngilizce'de bulunan kelimelere benzer komutlar kullanılarak bu kolaylık sağlanmıştır.Programcı bayraklar (flags) ikillikler(bits) in ve bunların nerede bulunduğunu bilmeyebilir ve bu onun çalışmasını engellemez.Mamafi yüksek seviyeli dillerin dezavantajı çok yavaş çalışmalarıdır.

6502'nin makina dilindeki kendi komutlarına OP CODE 'lar (operation codes) denir.Herbir opcode 6502 'ye ne yapacağını söyler.Opcod'e'lar çok basit ve çok kısa zamanda programın çalışmasını sağlar.Daha kolay anlaşılabilmesi için (mnemonic) hatırlatıcılarla verilir.

Örneğin LDA H200 'ün opcode'u onaltı tabanına (Hexadecimal) A9 olupbunu izleyen iki byte'la yine hex 200 adresini gösterir.

Bunun anlamı 200 hex adresindeki lbyte'lık bilgiyi birikece yükle (Load the Accumulator) işlemini göstermektir.

Programlar bu hatırlatıcılar kullanılarak yazılabilir.

Böyle programlara assembler language ile yazılmıştır denir,tabi mnemonic'lerle program yazabilmek içinde bir assembler/disassembler işletim programına sahip olmak gerekirkki bu pğramda makina dili yazılıp çalıştırılan bir programdır.

Derleyici dilde program kullanmanınBasic kullanmaya göre bazı avantaj ve dezavantajları vardır.Avantajı bellekte daha az yertutması ve çok daha hızlı çalışmasıdır.Ana dezavantajı programı yazması daha zordur ve yanlış birşey yapıldığı zaman hata mesajı vermez.Bu durumda makinanın kontrolü kaybolur.

Programcı hangi adresleme tipini seçeceğini ,kullanacağı komutun gerektiğinde ne kadar zaman alacağını ,o komutun durum yazmacında nasıl etkilendiğini görerek ve bilerek komutları arka arkaya sıralar ve programı yazar.Bu yazılımda en önemli konulardan biri de ara sonuçların kaybolmaması sorunudur.Örneğin bir işlemin sonucu bir yazmaçta elde edilmiş ise ve daha sonraki komutlarada o yazmaç başka işlemlerde kullanılıyorsa ara sonuç kaybolacaktır.Kaybolmaması istenen ara sonuçlar daha sonra kullanılmak üzere RAM'de bilinen bir adrese aktarılır.

Komut listesinde her komutun karşı tarafında ,o komutun hangi işlemi yaptığı gösterilmiştir.Programcı bunları bilmek durumundadır.Mikroişleyiciye yaptırılacak işlem karmaşık ise uygun komutları seçmek gerekir.Yoksa program çok uzun olabilir, ve yanlışlık oranı artar.

Kaynak Program:

Mikroişleyiciye yaptırılacak işin önce akış çizelgesi yapılmalıdır.Bu akış çizelgesini gerçekleştirmek için uygun komutları arka arkaya sıralamak gerekir.Yazılımda sembolik yazılım (MNEMONIC) ve sözde komutlar kullanılırsa "kaynak program" (source program) elde edilir.Programı önce kaynak program olarak yazmanın yararı ,irdelemelerin ve yanlışları düzeltmenin daha kolay olmasıdır.Program yazımında böylece kullanılan dil "derleyici dil" (Assembler language) adını alır.En çok bilinen kaynak program yazılımlarından birincisi LINE ASSEMBLER diğeri ise TWO PASS ASSEMBLER diye adlandırılanlardır.İkincisi birincisine oranla daha yeteneklidir.Birinde satır satır mnemonic'ler tercüme edilir,ve etiket denilen birazdan açıklanacak kısım yoktur.

Kaynak program (Source program); 4 bölgeden oluşur.

1-Etiket (LABEL):Programın program parçasının yada alt programın işlemidir.Sonuçta etiket bulunduğu yerin adresi olarak ortaya çıkacaktır.

2-İşleç (OPERATÖR):İşlemi yaptıran komutun (instruction) adıdır.

3-İşlenen (OPERAND):İşleme giren veriyi tanımlayan addır.

4-Açıklama (COMMENT): Gerektiğinde yazılan programın daha iyi anlaşılabilmesi için yapılan açıklamalar söz konusudur.

Bir örnekle açıklarsak

Etiket	İşleç	İŞLENEN	Açıklamalar
BAŞLA	LDX	VERI	Veri süreyi ayarlar.
DÖNGÜ	DEX		IX bir eksiltilyor.
	BNE	DÖNGÜ	sonuç sıfır değilse dön
	RTS		sonuç sıfırsa bun altprogramdan ana program dönülür.

Yukarıda bir zamanlama programı verilmiştir.VERI programcı tarafından tanımlanan bir adrestir.Bu adresteki data X yazmacına alınır.İşlenen olarak döngü ,etiket döngünün satırındaki DEX komutunun program belleğinde bulunduğu yerin adresidir.İşlemsonucu sıfır olmadığı sürece program sayacı BNE komutundan sonraki adres yerine DEX 'in bulunduğu adresle dolacaktır.Böylece program X sıfırlanana kadar devam edecektir.

Kaynak programı ,mikroişleyicinin anlayacağı -komut listesinde bulunan -ikili kodlara çevirmek gerekir.Bu iş ya derleyici program (ASSEMBLER PROGRAM) ile donatılmış bir bilgisayarla yapılır yada programcıtarafından komut listesine bakılarak ve gerekle işlemler yapılarak gerçekleştirilir.

Programcı tarafından yapılacak çeviride programın fazla uzun olmaması gerekir.

Yoksa yanlışlık oranı artar ve çeviri işlemi çok zaman alır.Kaynak programdan elde edilen komut kodlarının oluşturduğu program " amaç program" adını alır.Program belegine amaç program yerleştirilecektir.Bu program onaltılı (HEX) sayılarla kolayca gösterilebilmektedir.Böylece sembolik komutlarla yazılmış program ,bellege "makina dili" ne çevrilmiş olarak doldurulur ve işlenir.

Adresleme Tipleri (Addressing modes):

Yazılımda komutların işleyeceği verileri ve adreslerini iyi belirlemeye gerekir.OPCODE (işleç) ,OPERAND 'ı (işleneni) nereden alıp nereden işleyecektir? Bu sorunun yanıtı adresleme tipinin seçimindedir.

Programın çalışmasında elde edilen veriler ara yada kesin sonuçlar ya mikroişleyicinin yazmaçlarından birinde yada bellekte biryerlerde bulunacaktır.Programın akışı içinde veri dolaşımını uygun adresleme tipini seçerek sağlamak gerekir.6502 programcısının tanıklanmış adresleme tipleri aşağıda tanımlanmıştır.

1-Kapalı Adresleme (Implied Addressing):

Dahili yazmaçlarda çalışan tek byte'lık komutların kullandığı bir adresleme tipidir.Bu komutlar veri (data) ya ihtiyaç göstermez.

2-Hemen adresleme (Immediate addressing):

Opcode (işleç)'i bir byte'lık operand (işlenen) izler.

Opcoda 'un belirlediği yazmaça bir byte veri yükleme yada mantık ,aritmetik işlemlerinde register'ın(yazmacın) içerigini etkilemede kullanılır.Daha açık bir söyleyişle 6502 register'lerine direkt olarak ulaşp üzerinde hemen işlem yapılmasını sağlayan adresleme tipidir.

3-Mutlak adresleme (Absolute addressing):

Bu adresleme tip daima üç byte' gerektirir. İlki opcode bunu adresin daha az ağırlıklı byte'ı sonrada daha ağırlıklı byte takip eder. Adres byte'ların sırası bütün adresleme tiplerinde hep aynıdır. Birikeç yada yazmaçlara adresi iki byte olarak verilen bellek bölgesinden veri nakletmektir.

4-Zero-Page Addressing (Sıfırıncı sayfada adresleme):

Sıfırıncı sayfada adresleme mutlak adreslemeye çok benzer fakat burada daha ağırlıklı byte daima sıfırdır ve yazılmaz. Bunun için opcode 'dan sonra sadece bir byte olarak az ağırlıklı adres byte'ı verilir. Bunun iki faydası vardır bellekten iki yerine bir byte harcanmış olmaktadır aynı zamanda clock cycle (saat darbesi periyodu) zaman kazanılmış olur.

5-Birikeç Adresleme (Accumulator addressing):

Birikecin kendisini döndürmek , kaydırmak istediğimiz komutlarda bu tip adresleme kullanılır.

6-Bağıl yada Göreli adresleme (Relative addressing):

Bu tip adreslemede yalnızca şartlı dallanma (conditional branch) komutları için kullanılır. Opcode'u izleyen byte offset'i belirler. Bu bir byte ile programın akış pozisyonundan 126 geriye 129 ileriye dallanma yapılabilir.

7-Dizinli Adresleme (Indexed Addressing):

Dizin Yazmacı (index register) kullanılarak yapılan adresleme tipidir.

Mutlak, X(yada Y) (Absolute, X(orY)):

Opcode 'dan sonra iki byte mutlak adres ile verilir ancak bu adrese dizin yazmacının içeriği toplanarak gerçek adrese ulaşılır. 256 byte uzunluktaki tablolara ulaşmak için dizin yazmacı ile birlikte kullanılabilir. Arguman genellikle tablonun ilk elemanının adresini gösterir.

Sıfırıncı sayfa, X(yada Y) (Zero page, X(or Y)):

Mutlak dizinli adresleme ile aynı şekilde çalışır yalnız mutlak adresleme yerine sıfırıncı sayfada adresleme kullanılır.

Bu modu kullanan komutlar toplam iki byte uzunluğunda olup belleğin ilk 256 byte 'ındaki tablolarla sınırlıdır.

8-Dolaylı Adresleme (Indirect Addressing):

Mutlak adreslemede olduğu gibi toplam üç byte 'dan oluşur. Opcode'dan sonra gelen (önce az ağırlıklı adres byte'ı) adres bir sonraki adres 'in gösterdiği veriler opcode'un kullanacağı gerçek adresi gösterirler.

9-Dizin yazmaçları kullanılarak dolaylı adresleme:

Indirect addressing using the Index Registers:

Son iki adresleme tipi yine dizin yazmaçlarını kullanır.

Dolaylı dizinlenmiş adresleme (Indirect indexed addressing):

(Indirect), Y şeklinde gösterilir. Burada operand'da verilen adres bir byte 'dır. Dolayısı ile sıfırıncı sayfadan verilen adres yazmacının içerdiği adresi ekleyecek böylece bulunan adrese ve bir sonrakine gidip gerçekten kullanacağı adresin MSB ve LSB (Most/Least Significant Bit) değerini bulur.

Dizinlenmiş dolaylı adresleme (Indexed Indirect Addressing):

(Indirect), XX formunda gösterilir. İki byte'lık operandın gösterdiği adres ve bir sonraki adresteki data'lar dizinlenecek adresi gösterir yazmacın içerdiği ile bu adres toplanarak gerçek adres bulunur.

6502 donanımı (Hardware):

Donanım açısından özellikleri yapımçı firma tarafından (data sheet) veri sayfalarından bulunabilir. Ekte bu da vardır.

Bu bölümde 6502'nin fiziksel yönden bacak bağlantılarını ve onların nasıl kullanılacağını tanımlayacağız.

Adres Yolu (Address bus):

Bunlar 6502 nin ulaşmak istediği byte' ları seçmek için kullanılır.En az ağırlıklısı A₀ olmak üzere A₁₅ 'e kadar 16 tanedir.Herbirinin 0 yada 1 olma durumuna göre $2^{16}=65536$ farklı adres seçilebilir.6502 'nin adres uçlarının hepsi çıkışıdır, bir başka cihaz tarafından kontrol edilemez.

Veri Yolu (Data Bus):

6502 içindeki yazmaçlara gelen veya giden komutlar ve data 'lar 8 ucu kullanırlar.Herbir data line tristate (üç durumludur) olup logic 0 yada 1 olarak çıkış olabileceği gibi giriş olarakda çalışabilirler.Veritransferi yalnızca ϕ_2 clock işareti 1 durumundayken gerçekleşir,bu işaret 0 olduğunda ise veri yolu yüksek empedans gösterir.(H-Z)

Read/Write (R/W) (oku /yaz imi):

6502 'ye data girileceğini yada çıkılacağını çevre cihazlara bildirmek için 6502 tarafından üretilen bir çıkıştır.

Bu uç logic 1 iken işleyicinin data okuyacağını 0 iken data yazacağını gösterir.Boşta iken 1 dir.R/w ucu ϕ_2 clock sinyali low iken konum değiştirir.

ϕ_0 Clock girişi:

Bütün mikroişleyiciler kendi içindeki olayları senkronize edebilmek için clock girişine ihtiyaçları vardır.(Bazılarında tümleşik devrelerinin içine yerleştirilmiş clock generatörleri vardır.)6502 nin clock girişi normalde 1MHz'dir diğer modeli ki 6502A 'da ise 2MHz 'lık saat darbeleri girilir.

Bu clock girişinden elde edilen saat darbeleri ile birbiri ile zıt fazlı ϕ_1 ve ϕ_2 clock işaretleri dışarı gönderilir

6502 senkron bir mikroişleyici olarak tasarlanmıştır bu nedenle veri transferi yalnızca 62 saat darbelerinin yalnızca 1 konumunda olduğu süre içinde gerçekleşir.

Reset (Kur imi):

6502'ye güç verildiğinde bu giriş kullanılarak işlemci ilk şartlanır.(Initialisation)Güç verilir verilmez bu giriş bir periyod 0 'da tutulur ve sistem kendi kararlılığını sağlar.Sonra logic 1'e çekilerek ilksartlanma sağlanır işlemci 6 clock periyodu bekler ve FFFC ve FFFD adreslerine gidip buradaki verileri program sayacına yerleştirir.

Böylece program sayacındaki adresten itibaren gelen program çalışmaya başlar.Reset iminden sonra bütün yazmaçlar sıfırlanmış olarak kullanıma sunulur.

Interrupt Request (IRQ):

Çalışmakta olan bir program işlemciye giren bu uc sayesinde kesilebilir.Ancak bu kesme işlemi durum yazmacında (status register) I bayrağı 0 ve bu giriş de 0 'a çekilmiş ise gerçekleşir.6502 komut akışını durdurur I bayrağını otomatik olarak set eder, ve program FFFE ve FFFF adreslerinde tutulan adresten devam eder.

Non-Maskable Interrupt (NMI):

IRQ 'ya benzer fakat I flag ne olursa olsun kesintiye sebep olur.Bu giriş logic 1'den 0'a geçtiğinde işlemci yaptığı işi bırakır ve FFFA ile FFFB adreslerinin gösterdiği adresteki programı çalıştırmaya başlar.

IRQ 'da bir kere sıfıra çekildikten sonra sizin kesinti istegini ortadan kaldırmanızı bekliyordu.NMI'de ise yalnızca high seviyeden low seviyeye geçişlerde etkilendiğinden peşinden tekrar NMI ucunu kullanmak istiyorsak önce 1'e sonra tekrar 0 'a çekmemiz gerekir.

Ready (RDY):

Bu giriş yavaş belleklerle (slow memory) 6502 'yi senkronize etmek için kullanılır. Ø2 low iken yalnızca değişir. Ø2 high seviyeye gittiği zaman RDY low ise bir sonraki makina periyodu gecikecektir. Eğer hızlı bellekler kullanılıyorsa bu giriş sürekli 1 konumda tutulmalıdır.

Synchronisation Signal(SYNC):

Bu çıkış opcodeu yerine getireceği zaman 1' konumuna geçer.

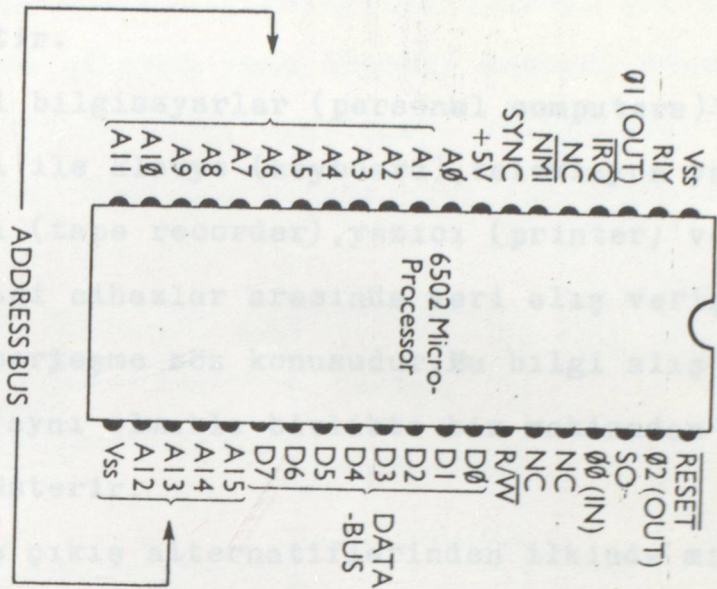
Set Overflow (SO):

Bu giriş 1'den 0 konumuna gelince durum yazmacındaki V bayragını set eder. Bu işlem aritmetik komutlarla ve yazılımla halledilebildiğinden fazlaca bir üstünlük getirmez dolayısı ile bu uç bazı bilgisayarlarda logic 1'e çekili olarak durur.

Besleme uçları (VCC -VSS):

Normal işletim gerilimi ve tümleşik devreye güç temini için %5 toleranslı 5,0VDC gerilim gerekir. VCC +7,0V 'dan fazla ,VSS' de -0,5V' dan az olmamalıdır.

Fig. 10.1: The 6502 Micro-Processor



*these control lines are available at the expansion bus, and can be utilized for user designed peripherals.

- D0 – D7 : DATA-BUS
- A0 – A15 : ADDRESS-BUS
- $\overline{R/W}$: READ/WRITE
- $\overline{IRQ}$: INTERRUPT REQUEST, ACTIVE IF THE "I" REGISTER IS AT "0" & $\overline{IRQ}$ IS PULLED LOW.
- $\overline{RESET}$: INITIALIZES THE 6502 FROM A POWER DOWN CONDITION.

BIT	LEVEL	POWER	DECIMAL
A ₀	1	2 ⁰	1
A ₁	1	2 ¹	2
A ₂	1	2 ²	4
A ₃	1	2 ³	8
A ₄	1	2 ⁴	16
A ₅	1	2 ⁵	32
A ₆	1	2 ⁶	64
A ₇	1	2 ⁷	128
A ₈	1	2 ⁸	256
A ₉	1	2 ⁹	512
A ₁₀	1	2 ¹⁰	1024
A ₁₁	1	2 ¹¹	2048
A ₁₂	1	2 ¹²	4096
A ₁₃	1	2 ¹³	8192
A ₁₄	1	2 ¹⁴	16384
A ₁₅	1	2 ¹⁵	32768

ADDRESS LINES
SPECIFYING MEMORY

TOTAL = 65,535 + 1 = 65,536 MEMORY LOCATIONS

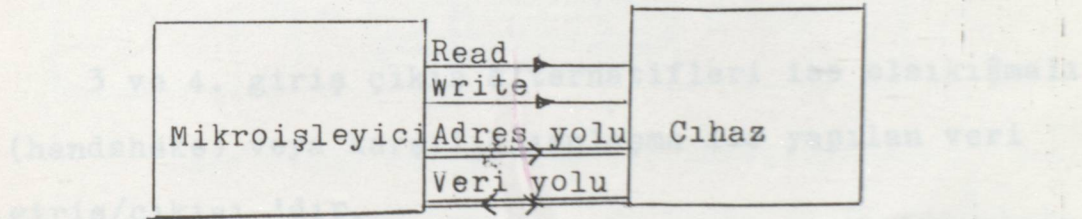
ALL LINES AT 0 : MEMORY
LOCATION 00000

Mikrobilgisayarlarda Giriş/Çıkış Kontrolu:

Mikrobilgisayarların en temel fonksiyonu dışarıdaki bir cihazdan bilgi alıp işlemek ve işlenmiş bu bilgiyi dışarı iletmektir.

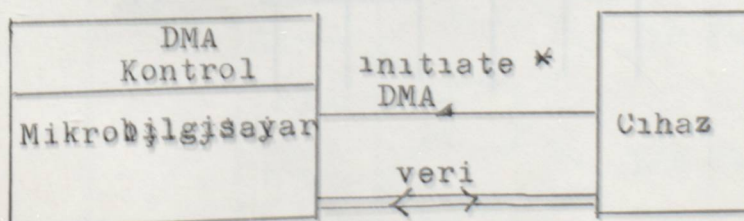
Tüm kişisel bilgisayarlar (personal computers)'da mikroişleyicisi ile klavye (keyboard), televizyon yada monitor kaset kaydedici (tape recorder), yazıcı (printer) ve disk drive gibi harici cihazlar arasında veri alış verişi diğer bir anlamda haberleşme söz konusudur. Bu bilgi alış verişi prensip olarak aynı olmakla birlikte bir makinadan diğerine farklılıklar gösterir.

Bilgi giriş çıkış alternatiflerinden ilkinde mikrobilgisayar ve karşısındaki cihaz arasındaki etkileşimi merkezi işletme birimi (Central Process Unit (CPU)) kontrol eder.



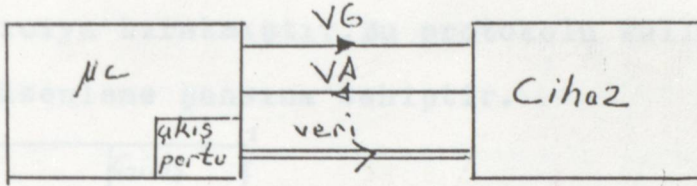
CPU örneğin bir display'i kontrol eder. (LED'lerden oluşan bir panelde yürüyen yazıların çıkarılması gibi.)

İkinci giriş çıkış alternatifinde CPU tarafından olaya bakıldığında olay otomatiktir. Ancak cihaz tarafından bakılacak olursa veri transferi cihazın kumandasında gerçekleşir. Cihaz veri transferine hazır olduğu zaman bu transferin esaslarını belirler ve CPU'nun ortaklığını dikkate almadan işlemini görür.

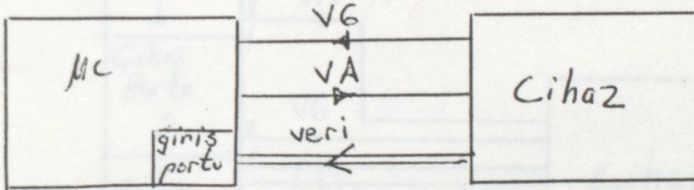


DMA (Direct Memory Access) controller:

CPU 'yu işe karıştırmadan bellek gözlerine ulaşmayı sağlar. Genellikle mikrobilgisayarın dışındaki bir cihazdan RAM'e veri dizisi girmeyi yada okumayı kontrol etmek için kullanılır. Tipik bilgisayar uygulaması ; bilgisayar ve manyetik disk veri depolama birimi arasında kayıt transferini düzenler.



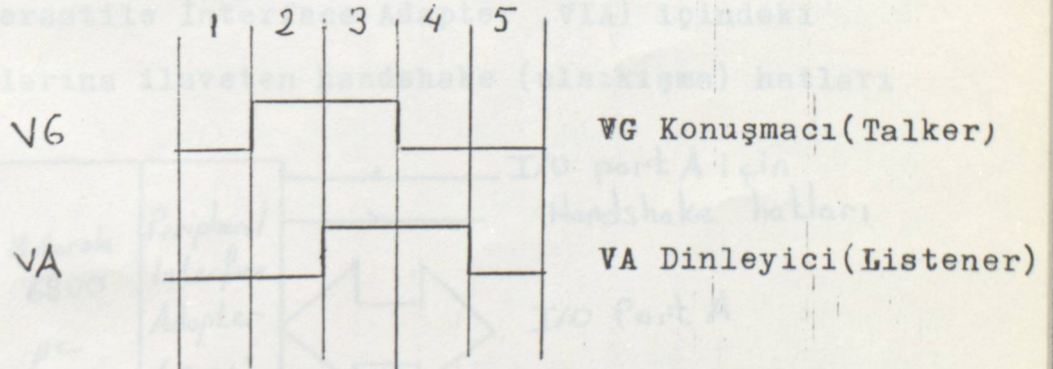
VG = Veri gönderdim.
VA = Veriyi aldım.



3 ve 4. giriş çıkış alternatifleri ise el sıkışmalı (handshake) veya karşılıklı anlaşma ile yapılan veri giriş/çıkışı 'dır.

Bilgisayar ve cihaz arasında her bir byte bilgi transferi için peşpeşe adımlarla el sıkışma olayı gerçekleşir.

Bu veri alış verişi aşağıdaki gibi 5 adımla gerçekleşir.



1) Her iki tarafta veri giriş-çıkışı için hazır.

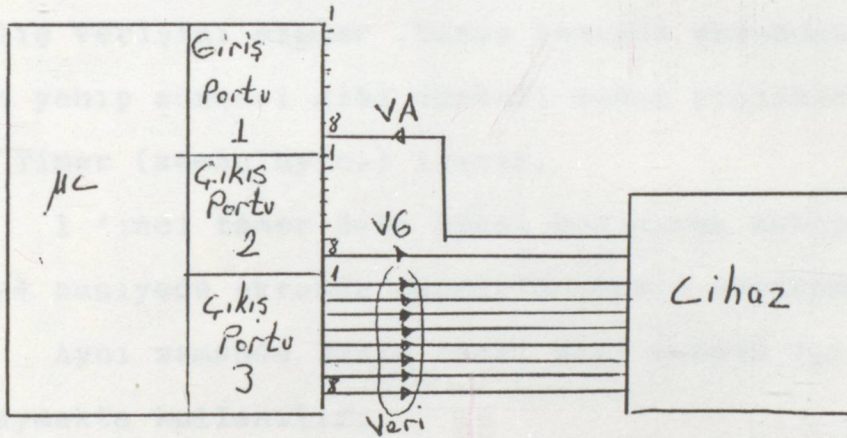
2) Konuşmacı mesajı gönderir. $VG=1$

3) $VA=1$ yapıp dinleyici mesajı aldığını belirtir.

4) Konuşmacı veri aldıysa bir sonrakini gönderebileceğini söyler.

5) Dinleyicide bir sonraki mesajı alabileceğini belirtir.

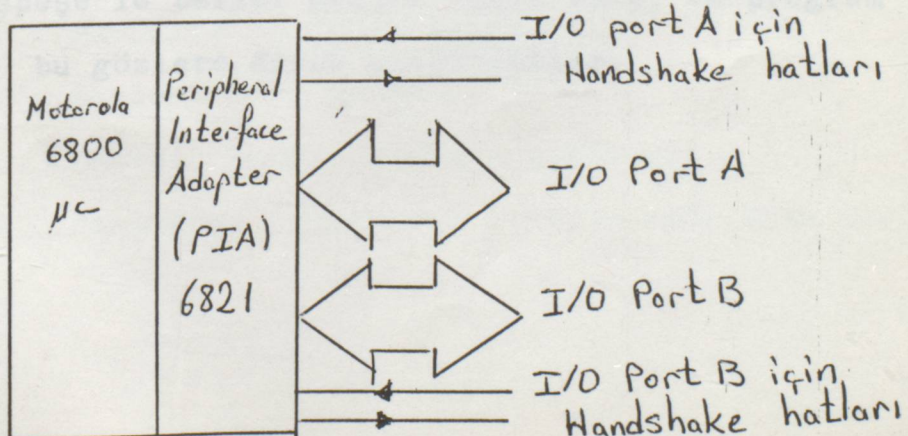
Bazı mikrobilgisayarlar DA ve DG el sıkışma mesajlarını kullanıcıya bırakmıştır. Bu protokolü kullanıcı istediği gibi düzenleme şansına sahiptir.



Elsıkışma hatlarının bir düzenlemesi

Diger mikrolarda ise her I/O portu kendi handshake hatlarına sahiptir.

Bunlara örnek olarak 6800 ailesinden 6821, 6502 ailesinden 6522 ticari adları ile bilinen (Peripheral Interface Adapter, PIA) yada (Versatile Interface Adapter, VIA) içindeki iki I/O portlarına ilaveten handshake (elsıkışma) hatları vardır.



6522 Versatile Interface Adapter (VIA)

6522 VIA genel amaçlı bir arabirim ünitesidir. 6502 ve 6800 ailesi mikroişleyicilerinin diğer birimlerle haberleşebilmesini mümkün kılan bir ara birimdir. Kullanım yerlerinden birkaçı (Keyboard) klavyeyi tarar, (printer) yazıcı ve ses üreticisine veri gönderir kaset sistemini kontrol eder. Gerekirse dışarıdan başka bilgisayarlar ile bilgi alış verişinde bulunur.

6522 2 paralel veri uçlarından 8'er bit'lik veri alış verişini sağlar. Bunun yanında ekrandaki cursor (benek) in yanıp sönmesi gibi düzenli zaman aralıklarını temin eden 2 Timer (zamanlayıcı) içerir.

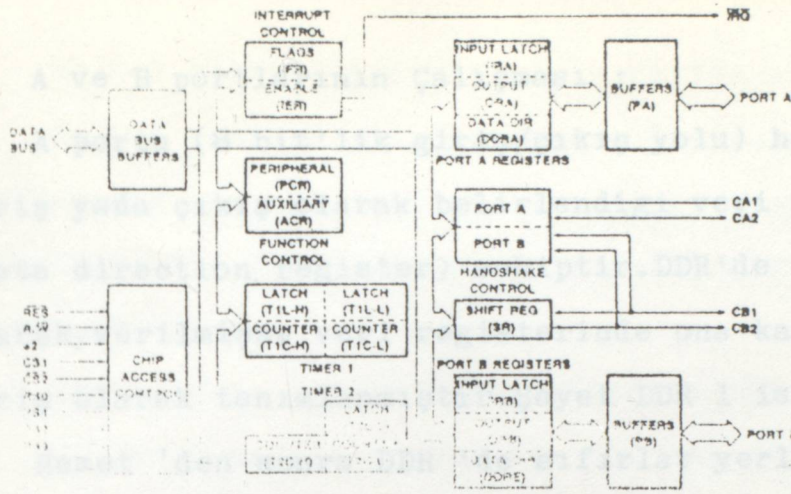
1 'inci timer daha genel kullanıma sahip olanıdır. Her 1/4 saniyede ekranda cursor'ün yakıp söndürmekte kullanılır.

Aynı zamanda Basic 'deki Wait komutu için zamanı saymakta kullanılır.

2'inci timer kasetten veri yüklemesi yapıldığı zaman kullanılır. Zaman periyodunu set etme ve sonrada kaset giriş durumunu test eder 2. timer'ın zamanlamasını değiştirerek kasetten farklı hızlarda veri okuyup yazılabilir.

6522 nin bir başka özelliği içinde bir shift register (ötelenebilir yazmaç) bulunmasıdır. Bu ötelebilir yazmaçın önemi seri bilgi transferinde kullanılabilmesindedir.

6522 peşpeşe 16 bellek gözünü işgal eder, ve program tarafından bu gözlerle direk ulaşılabilir.



6522 blok şeması

6522'nin iç yazmaç durumu :

Aşağıdaki tablo ile 16 yazmacının ortak kısaltmalar kullanılarak özeti verilmiştir.

R0	IORB	B portunun giriş/çıkış için data registeri
R1	IORA	Handshake ile A portu için giriş/çıkış reg.
R2	DDRB	B portunun data yön registeri
R3	DDRA	A portunun data yön registeri
R4	T1C-L	1.timer counter '1 oku yada lache yaz(Low-byte)
R5	T1C-H	1.timer counter' 1 oku yada latch'e yaz(High-byte)
R6	T1L-L	1.timer latch byte 'a ulaş.(low byte)
R7	T1L-H	1.timer latch byte' a ulaş.(high byte)
R8	T2C-L	2.timer counter'1 oku yada latch'e yaz.(low-byte)
R9	T2C-H	2.timer counter'byte'a ulaş.(high-byte)
RA	SR	Seri giriş/çıkış için shift register
RB	ACR	Auxillary Kontrol Register
RC	PCR	Perpheral kontrol Register
RD	IFR	Kesinti Flag Register
RE	IER	Kesinti Enable(mümkün) Register
RF	IORA	Handshake olmaksızın A portu için data registeri

A ve B portlarının Çalışması :

A portu (8 bit'lik giriş/çıkış yolu) herbir ucunun giriş yada çıkış olarak belirlendiği veri yön yazmaçlarına (data direction register) sahiptir.DDR'de bir bit'isıfır olarak verilmişse veri registerinde ona karşı gelen bit giriş olarak tanımlanmıştır.Şayet DDR 1 ise çıkış olacaktır.

Reset 'den sonra DDR 'de sıfırlar yerleşir böylece port'un bütün bitleri giriş olacaktır.Giriş olarak tanımlanmış port'un bir bit'ine yazmanın bu ucun çıkış seviyelerine etkisi yoktur.Çıkış olarak set edildiği zaman veri yazmacı bu uçtaki çıkış seviyelerini kontrol eder.Giriş olarak set edilen bir bitten okuma yapılacağı zaman akım değeri uygulanarak bu pin okunabilir.Çıkış olarak tanımlandığında bir bit'den veri geri okumayla bu bit'e yazma aynı şey değildir.Bunun için daima I/O pin(uc) 'undaki değer okunur, çıkış buffer(tampon)'a giden değer değil.

B port'u da A ile aynı şekilde çalışır.Yalnız çıkışa set edilmiş bir bitten geri okunacağı zaman farklılık gösterir.B port'unun bit'leri I/O pin'deki değeri okumak yerine çıkış buffer'a giden değer okunur.

1. Alan genişletme yarıçapı kullanılarak interfacing için konfigüre edilebilir.

2. Alan ise printer port'a kullanılarak diğer iletebilir.

Ayrıca 8250 H ve 8255 H arasında kalan boş bellek alanı herhangi bir bellek gibi PEEK ve POKE komutlarına tabi olarak giriş yada çıkış alanı olarak kullanılabilir.

Bu alan yanında herhangi bir bellek alanına yerleştirilen veri giriş çıkış olarak kullanılabilir.

ORIC-1 mikrobilgisayarı için giriş/çıkış özellikleri:

ORIC-1 ,harici olayları uygun iletişim teknikleri kullanarak izlemek için ve bazı basit kontrol elektronigi deneyleri ile ilgilenenler için esnek giriş/çıkış imkanları verir.

Giriş/Çıkış için tanımlanabilecek iki alan vardır.

- 1)Üçüncü page(sayfanın)'in üst kısmı(3FF H'in altı)
- 2)Üçüncü sayfanın ilk bölümü ki 300 H ile 30F H arasında kalan bellek alanıdır ve ORIC içinde bu alan 6522 VIA (Versatile Interface Adapter) tümleşik devresinin 16 yazmacı tarafından kullanılmaktadır.İki tane 8 bit'lik giriş yada çıkış olarak tanımlanabilen port'a sahip olan bu çevre ara biriminin ilk 8 bit'i Sound Chip(Ses Tümleşik devresi)tarafından ses üretmek için kullanıldığı gibi printer (yazıcı) çıkışı içinde aynı zamanda kullanılmaktadır.

Bu kullanımlarının yanında birde ikinci port'tan 3 bit'in de birlikte kullanımıyla keyboard(klavye)'u taramak ve hangi tuşa basıldığını bulmak için de kullanılmaktadır.

VIA anlatıldığı gibi pek çok aktiviteden sorumlu olduğundan tabiki çok dikkatli kullanılmalıdır.

1. alan genişletme yarığı kullanılarak interfacing için konfigüre edilmelidir.

2. alan ise printer port'u kullanılarak dışarı ile haberleşilebilir.

Ayrıca BFEO H ve BFFF H arasında kalan boş bellek alanı herhangi dolu bellek gözü gibi PEEK ve POKE komutlarına hakim olarak giriş yada çıkış alanı olarak kullanılabilir.

Bunlar yanında herhangi bir bellek alanına yerleştirilen harici cihazlarda okunup yazılabilen bellekler gibi gözükebilir.

Bu olayı genişletme yarığında bulabileceğimiz $\overline{MAP}$ ucu logic 0 'a çekilerek gerçekleştirilebilir. $\overline{MAP}$ ucu Low 'a çekilince bu bellek bölgesindeki RAM disable (devre dışı) bırakılmış olur. Bu uç yardımıyla aynı zamanda ROM 'da devre dışı bırakılabilir.

3. sayfanın üst kısmı:

Bu alan arabirim yerleştirmek için en uygun yerdir.

30F H adresinin üzerini ve 3FF H adresine kadar olan bölgeyi kullanmak için gene genişletme yarığında bulunan (I/O Control) ucunu 0 'a çekmek yeterlidir. Dolayısı ile 6522 VIA da devre dışı kalır. 300 H ve 3FF H arasında bir adres ,adres yolunda görüldüğü zaman ULA (Uncommitted Logic Array) genişletme yarığına I/O ucu ile low konuma çekerek 3. sayfanın seçildiğini söyler. Fakat 3. sayfanın başını printer ve ses devresine çıkış veren VIA kullanmış olduğundan I/O Control ucu bu adreslerin kullanım amacını bozar. Bunun için haricten kullanılacak arabirimler için 310 H ve 3FF H arasına yerleştirilmesinde fayda vardır.

3. sayfanın alt kısmı: (300 H ile 30F H arası)

Bu printer arabiriminin kullandığı alanı kullanmak için bir takım hilelere baş vurmamak gerekir. Bu bellek alanının gerçek kontrolü ORIC içindeki 6522 VIA tarafından yapılmaktadır.

300 H ile 30F H arasında VIA 'nın 16 register'i vardır.

Bunların hepside giriş çıkış işlemi ile ilgili değildir, fakat bunlar için ayrıntılı bilgi data sheet (veri sayfası) lerde bulunabileceği gibi bu konu ile ilgili bir bölümde bu çalışma içinde bulunmaktadır. Bu bölümde en çok kullandığımız ilk dört yazmacın ve en sonuncusunun işlevlerini ve nasıl kullanılacağını anlatacağız.

DATA-LINES										BIT	
	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D	D	D ₄		
	1	1	1	1	1	1	1	1	1	BINARY LEVEL	
	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰	2 ⁴	POWER OF 2	
128 =	64	32	16	8	4	2	1			DECIMAL	

DECIMAL	BINARY	HEX
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

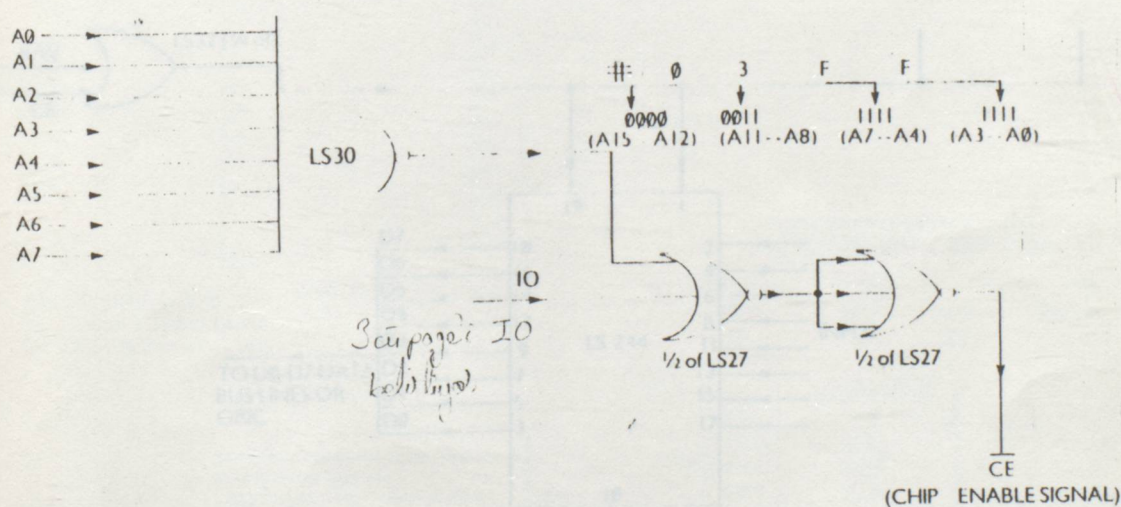
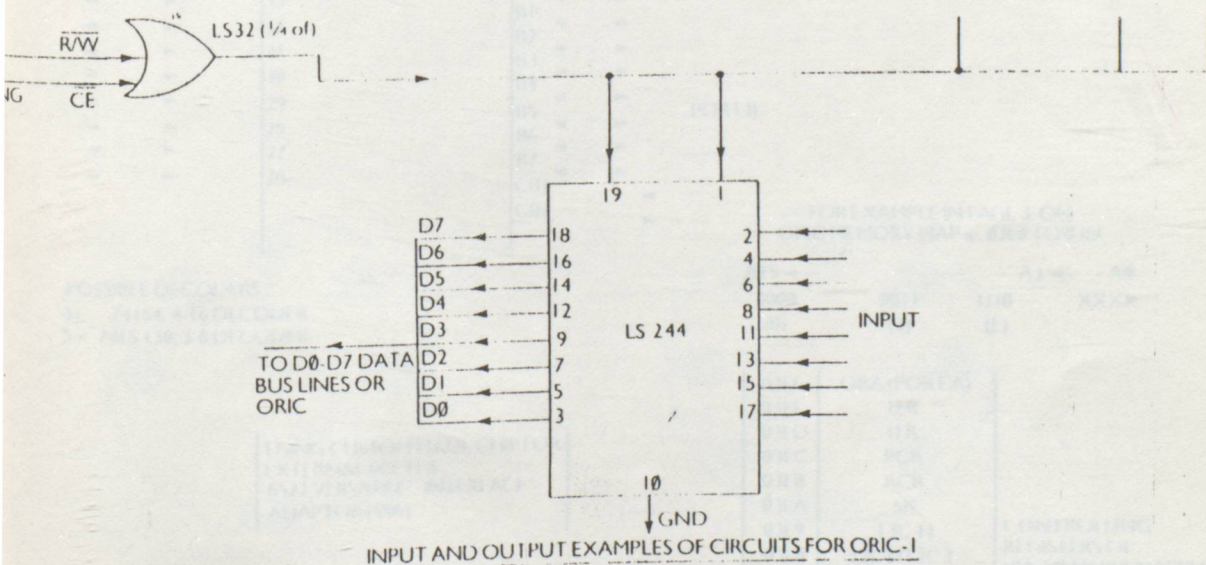
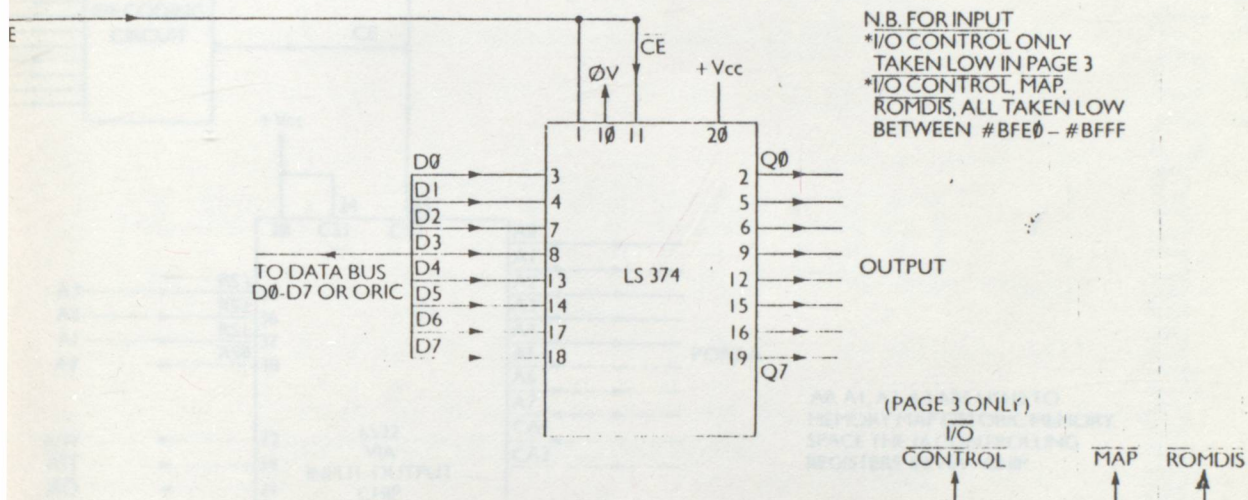
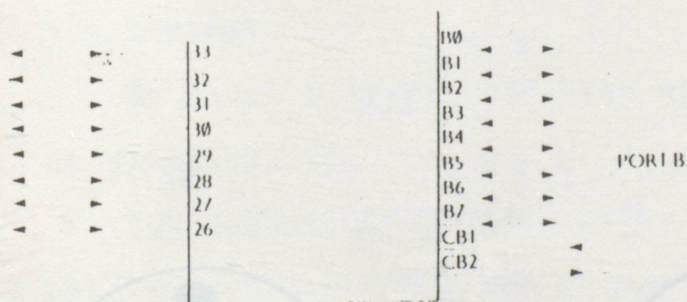
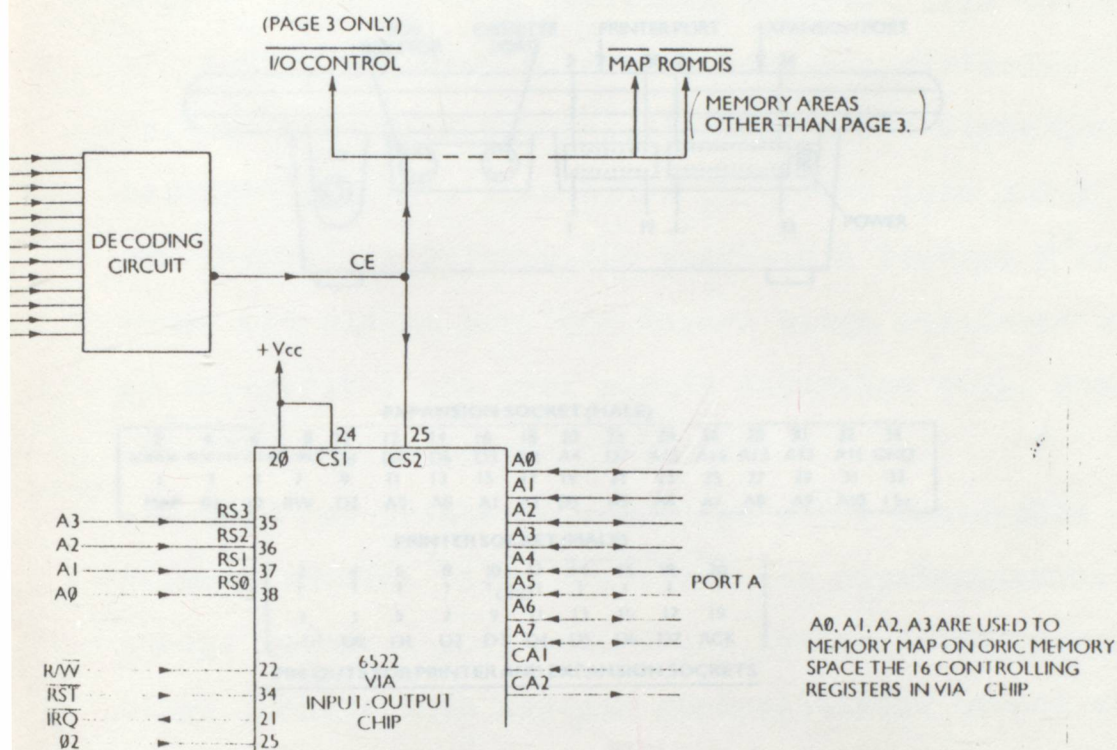


Fig. 10.5
ABSOLUTE DECODING OF
LOCATION 03FE (TOP OF PAGE 3)

ROM DECODING CIRCUIT





POSSIBLE DECODERS
4x 74154, 4-16 DECODER
5x 74LS 138, 3-8 DECODER

(USING CUSTOM MADE CHIP FOR
EXTERNAL I/O THE
6522 VERSATILE INTERFACE
ADAPTOR (VIA))

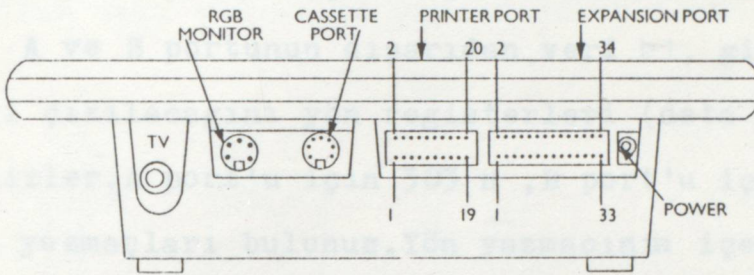
FOR EXAMPLE IN PAGE 3 ON
ORIC MEMORY MAP i.e. 03E0 TO 03E1

A15 ←	A3 ←	A0
0000	0011	1110
(0)	(3)	(F)
		XXXX

03EF	ORA (PORT A)
03EF	IER
03ED	IFR
03EC	PCR
03EB	ACR
03EA	SR
03E9	T3C.H
03E8	T2L.L/T2C.L
03E7	T1L.H
03E6	T1L.L
03E5	T1C.H
03E4	T0L.L/T0C.L
03E3	DDRA
03E2	DDRB
03E1	ORA (PORT A)
03E0	ORB (PORT B)

CONTROLLING
REGISTERS OF
VIA, MEMORY MAPPED
ONTO PAGE 3 MEMORY
SPACE

Appendix 11 ATMOS I/O Connections



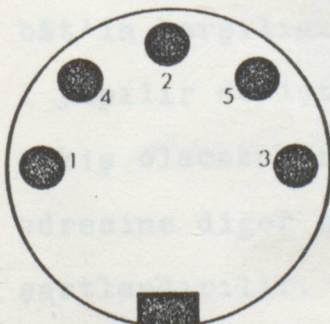
EXPANSION SOCKET (MALE)

2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34
ROMDP	RESET	CONT	IRQ	Dd	D1	D6	D3	D4	A4	D7	A15	A14	A13	A12	A11	GND
1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33
MAP	O2	IO	RW	D2	A3	A0	A1	A2	D5	A5	A6	A7	A8	A9	A10	15v

PRINTER SOCKET (MALE)

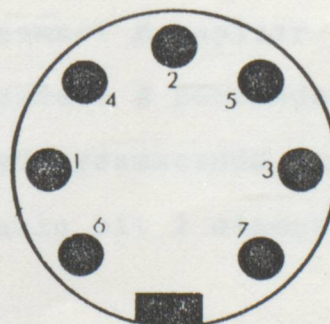
2	4	6	8	10	12	14	16	18	20
1	1	1	1	1	GRD	1	1	1	1
1	3	5	7	9	11	13	15	17	19
DIR	D0	D1	D2	D3	D4	D5	D6	D7	ACK

PIN-OUTS FOR PRINTER AND EXPANSION SOCKETS



RGB OUTPUT

1. RED
2. GREEN
3. BLUE
4. SYNC
5. GROUND



CASSETTE & SOUND

1. TAPE OUT
2. GROUND
3. TAPE IN
4. SOUND
5. RELAYS
6. RELAYS
7. RELAYS

6522 tümleşik devresi A ve B portu denilen 2 takım 8 bit'lik (data line) veri yoluna sahiptir. Bu veri yollarından B portu 300 H adresinden ,A port'uda 301 H adresinden dışarı ile veri alış verişinde bulunmak için kullanılır.

A ve B portunun dışarıdan veri mi, girileceğini ya da veri çıkılacağını yön registerleri (data direction registers) belirler. A port'u için 303 H ,B port'u için 302 H 'de bu yön yazmaçları bulunur. Yön yazmacının içeriği 0 ise bunun ait olduğu port'dan dışarı bilgi çıkılacak anlamına gelir, içerik 1 ise aynı port'a dışarıdan bilgi girileceğini belirler, ve portu ona göre şartlar.

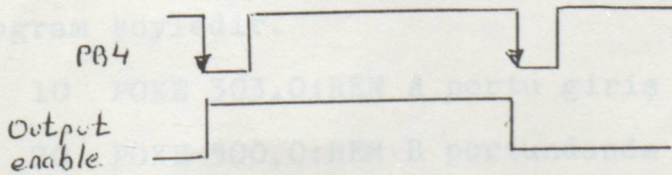
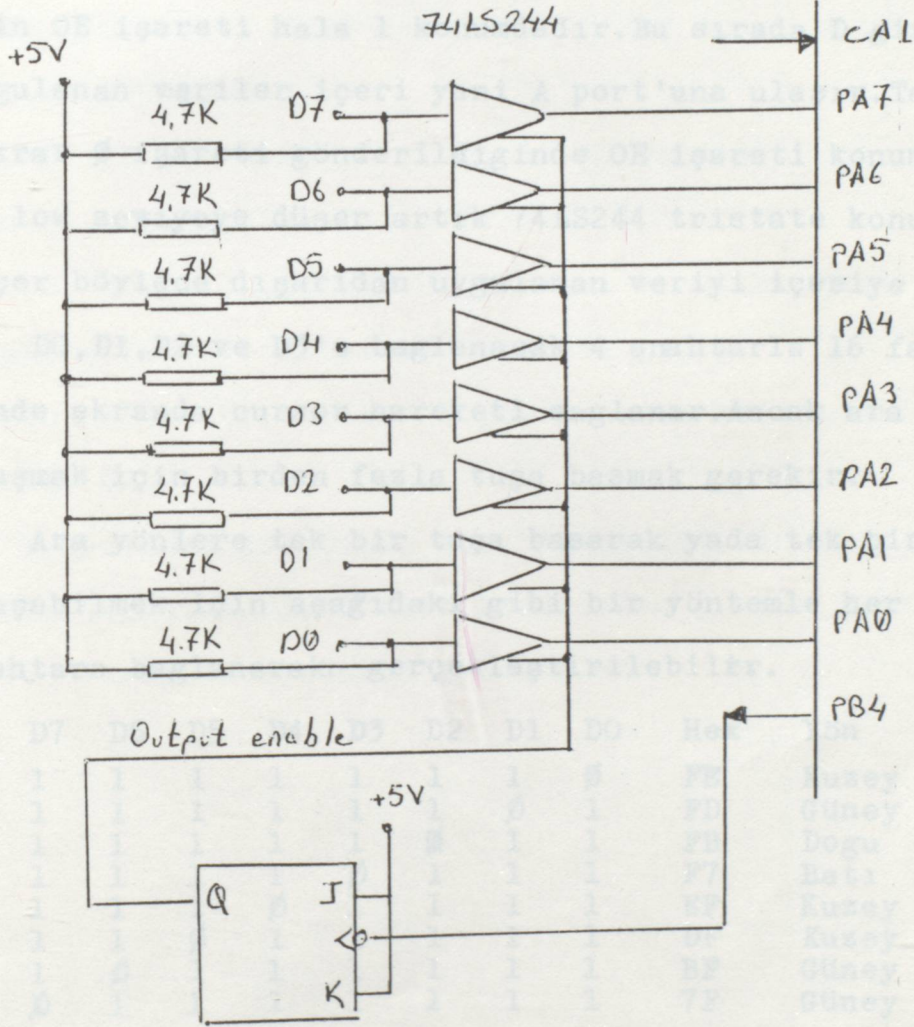
ORIC 'de printer (yazıcı)'ya çıkış Centronics olarak bilinen paralel bilgi transferi ile yapılmaktadır. Yazıcı çıkışı içinde A port'unun 8 ucu kullanılmıştır. Paralel bilgi transferinde mesajın gönderildiğini Strobe ucu ile karşıdaki cihaza belirtmek gerekir. Bu işlemde B portunun 4 nolu ucu ile sağlanır. Karşıdaki cihazında bu bilgiyi aldığını gösterdiği Acknowledge işaretide kontrol yazmacı ile tanınır.

Bu genel bilgiyi verdikten sonra bunların nasıl kullanıldığını örnekleyelim.

Yön yazmacında port'a giriş yapılacaksa giriş yapılacak bit'in karşılığındaki yön yazmacı 0 yapılır ,çıkış olacaksa 1 yapılır demiştik. Örneğin sadece B portundan 4 nolu bit çıkış olacaksa B port'unun yön yazmacında yani 302 H adresine diğer bitler 0 ,4 nolu bit 1 olacak şekilde yazmaç şartlandırılır.

Buradan hemen kullanılabilir bir sonuç çıkarmak istersek kolayca açıklanan özelliklerden faydalanıp kendimize kumanda kolu (joystick) arabirimi yapabiliriz. Bu aynı zamanda mikro'ya dış dünyasından bilgi girişi uçları olarakda kullanılabilir.

ORIC printer portu



30: WIRE:REM ORIC Grafik moduna geçti.

40: X=100:Y=90

50: A=PERK(301)

60: IF A= PE THEN Y=Y+1:REM Kuzey

PB4 port'una bir kere 0 yazılınca peşinden gelen clock ile tekrar 1 konumuna geçecek şekilde düzeleme yapılmıştır. Biz belli bir süre dışarıdan sürekli bilgi girişi olsun istiyorsak PB4 den bir kere 0 gönderdiğimizde output enable işareti 1 konumuna geçecektir PB4 tekrar ~~hiç~~ konuma geçse bile JK FF düşen kenarla tetiklendiği için OE işareti hala 1 konumdadır. Bu sırada D girişlerine uygulanan veriler içeri yani A port'una ulaşır. Tabiki PB4'e tekrar 0 işareti gönderildiğinde OE işareti konum değiştirir ve low seviyeye düşer artık 74LS244 tristate konumuna geçer böylece dışarıdan uygulanan veriyi içeriye aktarmaz.

D0, D1, D2 ve D3'e bağlanacak 4 anahtarla 16 farklı yönde ekranda cursor hareketi sağlanır. Ancak ara yönlere ulaşmak için birden fazla tuşa basmak gerekir.

Ara yönlere tek bir tuşa basarak yada tek bir anahtarla ulaşabilmek için aşağıdaki gibi bir yöntemle her bit bir anahtara bağlanarak gerçekleştirilebilir.

D7	D6	D5	D4	D3	D2	D1	D0	Hex	Yön
1	1	1	1	1	1	1	0	FE	Kuzey
1	1	1	1	1	1	0	1	FD	Güney
1	1	1	1	1	0	1	1	FB	Doğu
1	1	1	1	0	1	1	1	F7	Batı
1	1	1	0	1	1	1	1	EF	Kuzey doğu
1	1	0	1	1	1	1	1	DF	Kuzey batı
1	0	1	1	1	1	1	1	BF	Güney doğu
0	1	1	1	1	1	1	1	7F	Güney batı

8 anahtarlı joystick yapılmışsa bunun için gerekli program şöyledir.

```

10 POKE 303,0:REM A portu giriş olarak şartlandı
20 POKE 300,0:REM B portundanda 0 gönderildi.
30 HIRES:REM ORIC Grafik moduna geçti.
40 X=100:Y=90
50 A=PEEK( 301)
60 IF A= FE THEN Y=Y+1:REM Kuzey

```


ORIC-1 içinin Kumanda Kolu ara birimi (Joystick interface):

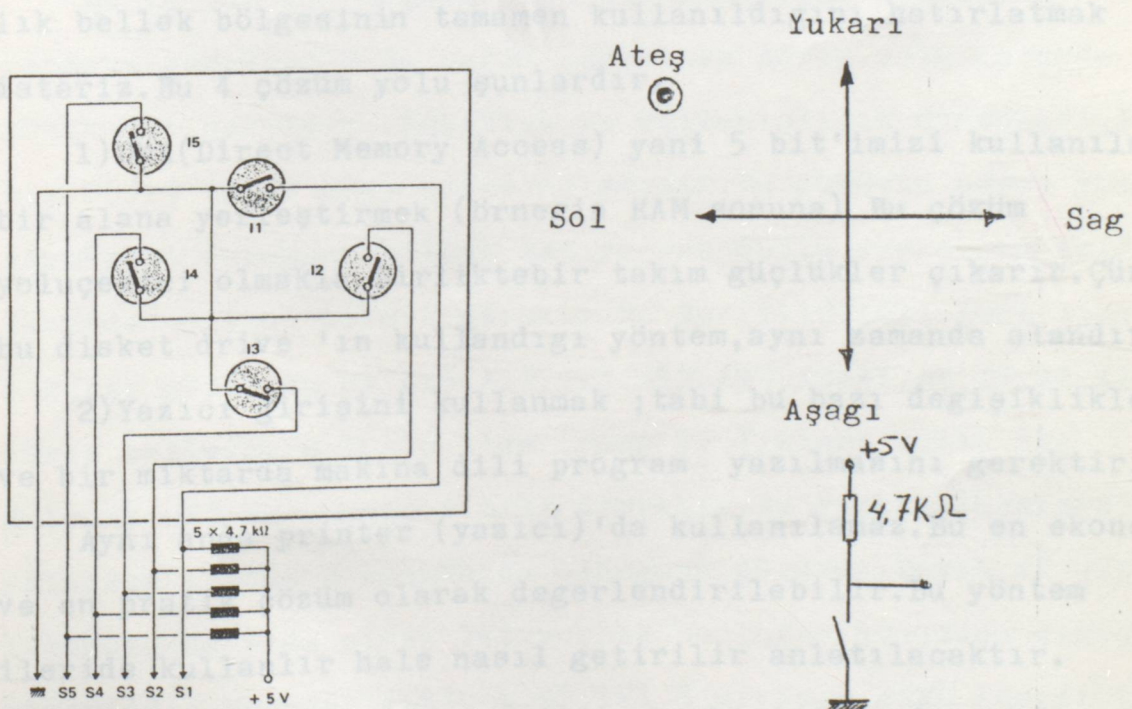
Alışlagelmiş mikrobilgisayarların en önemli kullanım alanı tartışma götürmez ki bir şekilde, oyundur.

ORIC-1, ve diğer benzer mikrolar bu ilginç kabiliyetlere ve ilginç fiatlara sahip olmasına rağmen oyun alanında genellikle kullanılan arabirimlerden yoksundurlar.

Mikro'nuzdan bir otomatik kontrol cihazı olarak istifade edebiliyorsanız bu arabirim ileride sözünü edeceğimiz çok değişik uygulamalar içinde gereklidir. Burada size en ince ayrıntılarına kadar anlatacağımız ara birim bilgisayara bir noktanın hareketi üstüne bilgiler verir.

Ayrıca komple bir yazılımınız varsa bilgisayarı "yüksek mikro sistemler" le donatabilirsiniz.

Elektriksel bir bakış açısından bir kumanda kolu (Joystick) kolun hareketine göre açılıp kapanan 5 anahtar olarak görülebilir.



Mesela kol yukarıda ise I1 kapalı olacak, eger kol aşağıda ise ve sagda ise I2 ve I3 kapalı olacaktır.

Eger her anahtarın bir ucu topraga diğeri ise 4.7KOhm üzerinden 5V 'a bağlı olduğu kabul edilirse S çıkışı anahtar kapalı olduğu zaman 0V ,açık olduğu zaman 5V'luk gerilime sahip olacaktır.

İşte biz bilgisayarın belleğine sokacağımız ,ikilik düzende 5 bilgiyi böyle elde edebiliriz.Burada aynı yöntemin yani anahtarın iki durumunun digital (sayısal) bilgilere çevirmekte kullanıldığını belirtelim.(Tuş takımında)

Interface (Arabirim):

ORIC-1'in kalbinin 6502 mikroişleyicisi olduğunu biliyoruz.Bu Z80 'e ters olarak giriş çıkış bölgeleri tanımlamamıştır.O zaman bizim yararlanabileceğimiz bir bellek bölgesi tanımlamamız gerekir,ki 6502 joystick'ten alınan 5 veriyi anlayabilsin.

Burada 4 çözüm yolu vardır.Bunları açıklarken 64KByte lık bellek bölgesinin tamamen kullanıldığını hatırlatmak isteriz.Bu 4 çözüm yolu şunlardır.

1)DMA(Direct Memory Access) yani 5 bit'imizi kullanılmayan bir alana yerleştirmek (örneğin RAM sonuna) Bu çözüm yolu çekici olmakla birlikte bir takım güçlükler çıkarır.Çünkü bu disket drive 'ın kullandığı yöntem, aynı zamanda alandır.

2)Yazıcı girişini kullanmak ;tabi bu bazı değişiklikler ve bir miktarda makina dili program yazılmasını gerektirir.

Aynı anda printer (yazıcı)'da kullanılamaz.Bu en ekonomik ve en pratik çözüm olarak değerlendirilebilir.Bu yöntem ileride kullanılır hale nasıl getirilir anlatılacaktır.

3)Bir RAM byte 'ının üzerine arabirimi kaydırmakda kolay ve ucuz çözümlerden biridir.Fakat çıkış konnektörünün üzerinde RAMDIS(RAM DISCONNECT)çeşitinden bir çıkış olmalıdır.

4) Aynı şeyi ROM'la denemek ; Herkes bilirki ROM monitor programı içerir,ve bir byte'ı ortadan kaldırmak yada değiştirmek "yasaklanmıştır".Ama biz ROM'da hiçbir işe yaramayan byte'lar bulabiliriz.(Mesela copyright içeren bazı byte'lar)

Aşağıdaki programı denersek

```
10 FOR X=60301 TO 60340
```

```
20 PRINT CHR$(PEEK(X));
```

```
30 NEXT X
```

"Software by Peter Halford and Andy Brown" mesajı ile karşılaşacaksınız.

İşte arabirim için kullanılabilecek 40 byte ortaya böylece çıkmış oldu.Bu sefer ROMDIS sinyali çıkış konnektör_ünde kesin gerektirir.Şimdi 60301 ve 60340 arasında bir adres seçip onun kodunu çözmeliyiz.(Decode operation)

Belli bir bellek bölgesini adres yolu üzerinden seçip kullanmaya decoding işlemi diyoruz.

Bu iş tek bir koşulu olan bir mantıksal fonksiyon yapmayı gerektirir.Bu yerin adresinin Adres yolu üstünde gözükmesi olayıdır .Rahatlık ve kullanışlılık nedenleriyle biz 60319 (BB9F Hex) adresini seçtik.Bu adresin adres yolunda gözükmesi durumunu bilmek istiyorsak 60319 sayısının ikili düzende eşitini bilmek gerekir.Bu da (1110.1011.1001.1111) e karşı gelir.Bu da A15,A14,A13,A12,A90,A8A7,A4,A3,A2,A1,A0 adres tellerinin logic 1'de (yani 5V),diger uçlar ise 0 (yani 0V) durumunda olması demektir.Bu adresi incelersek yalnızca 4 ucun 0 konumunda olduğunu görürüz.Demek ki hep 1 konumunda elde etmek istiyorsak bunların konumunu değiştirmek yeterlidir.Bunuda 74LS30 tipi iki 8 girişli NAND kapısı ile gerçekleştirebiliriz.Bu kapının çıkışı yalnızca tüm girişler 1 olduğunda 0 olur.Böylece bu adresimizin yol üzerindeki

varlığını iki 74LS30 devresinin çıkışında low duruma çevirdik.Şimdi yalnızca bu iki durumu birleştirmek kalıyor.

Bunuda bir OR kapısı ile gerçekleştiririz.Böylece adres decoding işlemi tamamlandı.Artıkadres yolunda ~~EB9F~~ H adresi belirince OR kapısının çıkışında logic 0 değeri görülecektir.Bu da veri yoluna ulaşım devremizin kumandasına yarayacak ve arabirimimizin sınırı olacaktır.

Adresimiz seçildiğine göre artık bu bellek alanının boşaltılması için ORIC-1'in ROM 'unun seçilmemesi gerekir

Bu seçilmeme durumu adres çözücü (decoder) çıkışının ORIC 'in genişletme yarığı (expansion bus) daki 2 nolu girişe bağlanması ~~çünkü~~ROMDIS girişine bağlanması ile temin edilir.Bu bir veya (OR) kapısı ile gerçekleşir.

ROMDIS girişine logic 0 uygulanırsa dahili ROM seçilmemiş olur ve bize boş bir bellek sahası (space memory) açılmış olur.Artık veri yolu üzerine bilgi girmek için hemen herşey hazırdır.Fakat burada çok önemli bir şey daha vardır.Mikroişleyici veri yolu üzerindeki verileri ne zaman okuyacaktır?Bu veri okuma yazma ile ilgili bir çıkış ucu ile (yani Read/Write (oku/yaz)) halledilir.ORIC'de bu çıkış 6502 verileri okudugu zaman 1 ,yazdığı zaman 0 dir.

Biz 6502'ye arabirimden gelen bilgileri okutmalıyız.Bu nedenle veri yoluna dışardan giriş mikroişleyici okuma durumundayken olmalıdır.Böylece yeni bir koşul gelmiş olurki buda R/W işaretinin evrigini alarak çözücü devrenin çıkışı ile veya kapısından geçirirerek sağlanır.Son olarakda clock işaretiyle senkroizasyon sağlanmalıdır.6502'de daha öncede söyledigimiz gibi veri transferi yalnızca 02 clock işareti 1 durumundayken gerçekleşir,0 ise veri yolu yüksek empedans gösterir.Buda adres kod çözümünün doğru olması ve R/W işaretinin read (okuma) konumunda olması hali gerçekleştikten sonra

son koşul olan clock işaretinin 1 konumunda olduğunun da kontrolü tamamlandıktan sonra veri yoluna bilgi yollanacaktır. Veri yoluna giriş izni işareti (CS) ni hazırlamış olduk.

Özet olarak CS (Chip Select) ancak aşağıdaki üç koşul doğrulandığında aktif (yani 0)'da olacaktır.

1) Adres yolunda =EB9F Hex

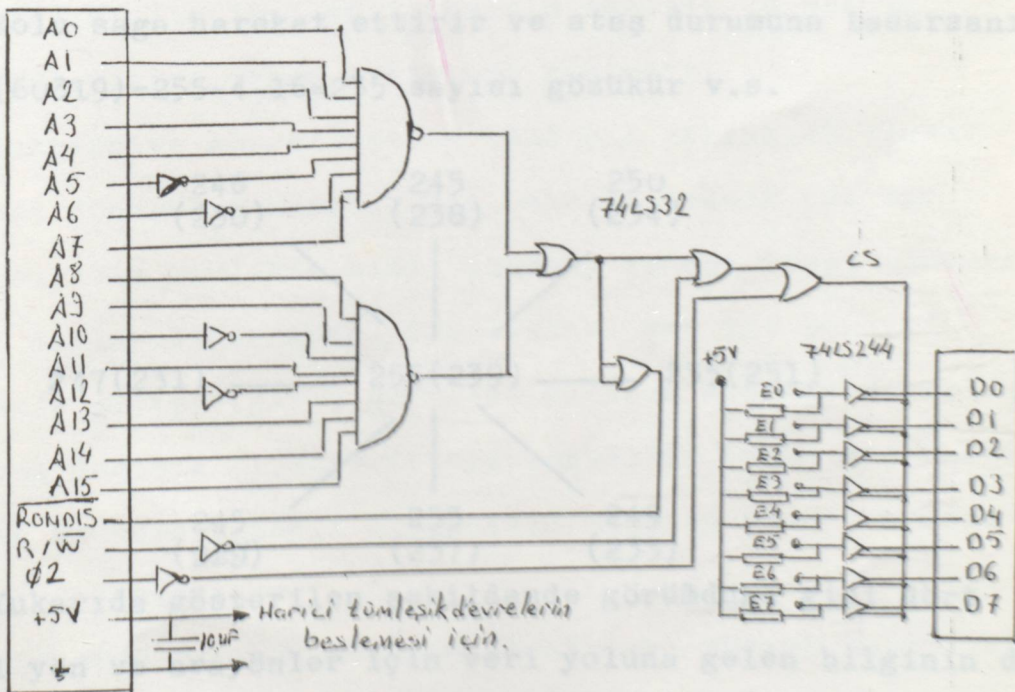
2) R/W=1

3) $\phi 2 = 1$

Veri yoluna giriş 74LS244 tipi TTL, 8 tane tampon (bufer) (üçkonumlu) (tri-state) ile sağlanır. Bu 8 giriş 4,7KOhm'luk dirençler yardımıyla +5V'a çekilmiştir. Bu arada ORIC'e bağlandığında 8 girişin ne durumda olduğunu bilmek için PRINT PEEK(60319) yazmak yeterlidir.

ORIC'in
Genişletme
Yarığı

6/6 74LS04 74LS30



EB9F adresini dekode (çözümleyen) edip dışarıdan bağlanan cihazdan veri alma devresi.

Arabirimin basit bir uygulaması :

Bir önceki sayfada bulacağınız devre ile en basit olarak bu bölümün başında anlatılan kumanda kolunun (joystick) bağlanması ve yazılımının verilmesi ile bu oğaya yeni bir bakış açısı gelecektir. Kumanda kolunun çalışması için program yazmak için onu oluşturan 5 anahtarın veri yolu üzerindeki etkisini incelemek gerekir. Bu organizasyon çoğunlukla şöyledir.

Bit 0:Yukarı (1)

Bit 1:Aşagı (2)

Bit 2:Sag (4)

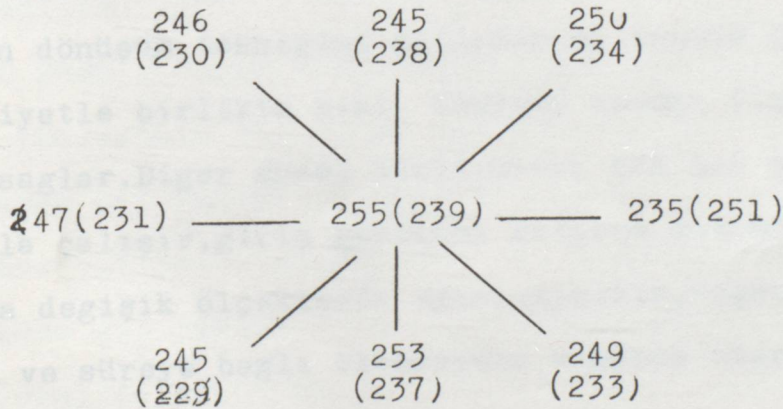
Bit 3:Sol (8)

Bit 4:Ateş (16)

Bit 5,6 ve 7:+5V

Eger kumanda kolu herhangi bir yönde yada ateş etme pozisyonunda degilse ki boştaadır. Bu durumda PEEK(60319)=255 'dir.

Kolu saga hareket ettirir ve ateş durumuna basarsanız PEEK(60319)=255-4-16=235 sayısı gözükür v.s.



Yukarıda gösterilen şekildende görüldüğü gibi dört temel yön ve arayönler için veri yoluna gelen bilginin dntaban_ındaki karşılıkları ve aynı şartlarda ateş tuşuna basılması hali için gegerler verilmiştir. Bunlar birer basic satırında kontrol edilerek ekranda gereken ve istenilen yerlere cursor(benek) hareket ettirilebilir.

ORIC-1'e Analog Giriş:

Bir mikrobilgisayarın digital dünyası ile analog işaretlerin gerçek dünyası arasında bağlantı kurabilmek için bir arabirime (interface) ihtiyaç vardır. Bilgisayarında yardımı ile bu arabirim akım yada gerilimi digital (sayısal) işarete çevirir, yada digital işareti gerilime dönüştürebilir.

Sözü geçen dönüşümü yapan hazır tümleşik devreler vardır. Düşük fiyata sahip bu devrelerden Ferranti ZN426E D/A (Digital /Analog Dönüştürücü), ZN427E ise A/D (Analog/Digital dönüştürücü) isimleri ile satılmaktadır.

Biz bu bölümde analog işaretlerin digital'e dönüşmesi ile ilgileneceğiz. Endüstriyel standart adı ADC 0808 yada ADC 0809 A/D dönüştürücünün kullanılması yukarıda sözü edilen ADC (Analog Digital Converter) 'lerden daha yetenekli olduğu için tercih edilmiştir.

ADC 0808, National Semiconductor yapımı olup CMOS teknolojisi ile imal edilmiştir. 8 kanallı, 8 bit ve mikroişleyiciye uygun çıkışlara sahiptir.

Successive Approximation (Ardışıl Yaklaşım) olarak bilinen dönüşüm tekniğini kullanır. Bu teknik iyi bir hassasiyetle birlikte hızlı dönüşüm zamanı (fast conversion time) sağlar. Diğer güzel özellikleri tek bir pozitif kaynakla çalışır, giriş gerilimi sıfırın altında ve üstünde kolayca değişik ölçeklerde ayarlanabilir, düşük güç harcaması olması ve süreye bağlı olmaksızın monoton olarak hassas olarak çalışmasıdır.

ADC 0808 'in ana dönüşüm elemanları yüksek kararlılığa sahip karşılaştırıcı (comparator), 256K gerilim bölücü devre, analog anahtarlama ağacı (switch tree), ve bunların yanında hepsinden önemlisi ardışıl yaklaşım yazmacları (successive approximation registers) dır. Ayrıca dönüştürücünün girişinde 3 bit adres çözücü (decoder) ile 8 farklı analog

ADC 0808 / 0809 8 bit, 8 kanallı mikroişlemcilere uyabilen Analog/Digital dönüştürücü:

Genel tanımı:

ADC 0808 ve ADC0809 mikroişlemcilere uygun kontrol mantığı , 8 kanal çoğullayıcısı ve 8 bit hassasiyetle analog işareti digital'e çeviren CMOS türü bir tümleşik devredir.

Dönüşümü ardışıl yaklaşım denilen (succesive aproximation) tekniği kullanarak gerçekleştirir.

Dönüştürücü ; bir yüksek, empedans bölücü kararlılığına sahip karşılaştırıcı (comparator) , bir ardışıl yaklaşım yazmacı ve birde analog anahtar ağacıyla 256K gerilim bölücüye sahiptir.

8 kanal çoğullayıcı ile 8 tek uclu analog işaretlerden herhangi birine direk olarak ulaşılabilir.

Tümleşik devre harici sığır ve tam ölçek ayarlarına ihtiyaç göstermeksizin hassas olarak çalışır.

TTL-Tristate (üç durumlu) çıkışları, çoğullayıcı adres girişleri ile (decode) seçilerek mikroişlemciye kolayca ilave edilir.

ADC 0808 , diğer A/D dönüşüm tekniklerinin en fazla aranan yönlerinin bir araya getirilmesi ile tasarlanmıştır.

Yüksek hız, yüksek hassasiyet, sıcaklığa minimum bağımlılık uzun süre çalışmada kararlılık ve tekrarlanabilirlik hassasiyeti ve minimum güç tüketimi bu ideal özelliklerinden birkaçıdır.

Bu özellikler otomotive uygulamaları , kullanıcıya makina kontrolü ve işlem yapabilme gibi imkanlar verir.

Özellikleri:

Çözümleyicilik: 8 bit

Toplam ayarlanamayan hata: $\pm 1/2\text{LSB}$

Kaybolmayan çıkış kodları

Dönüşüm zamanı 100mikrosaniye (10KHz)

Tek besleme kaynağı +5Vdc

Bütün mikroişlemcilere kolayca bağlanabilirlik

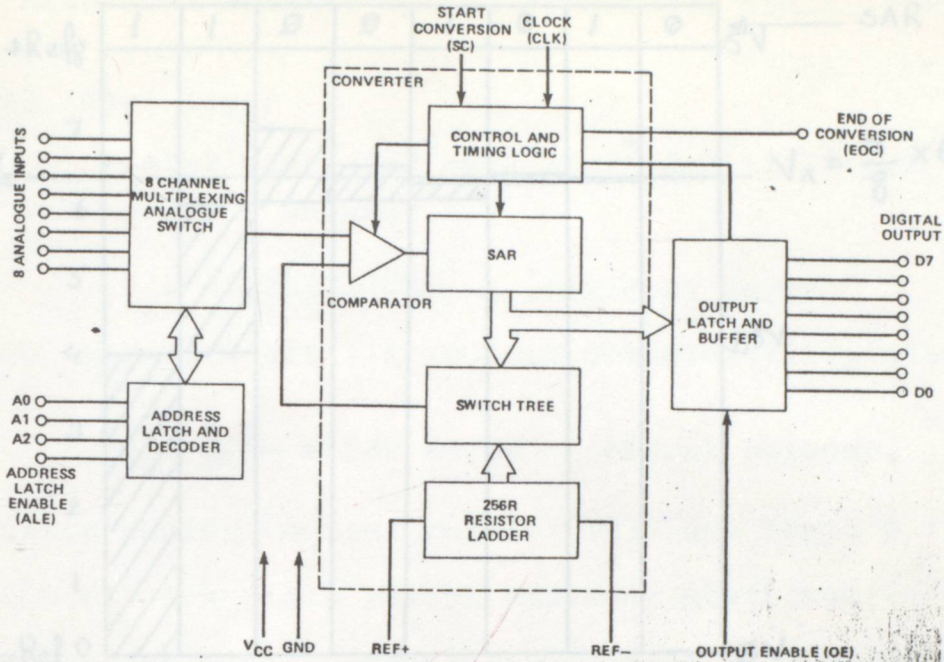
Çıkışların TTL logic seviyelerine uygunluğu

0 ile 5V arasında giriş gerilimi seviyeleri

Çalışma sıcaklığı -40 ile +85 derece arası

Düşük güç sarfıyatı :15mW

Çıkışlar tri-state durumuna kilitli



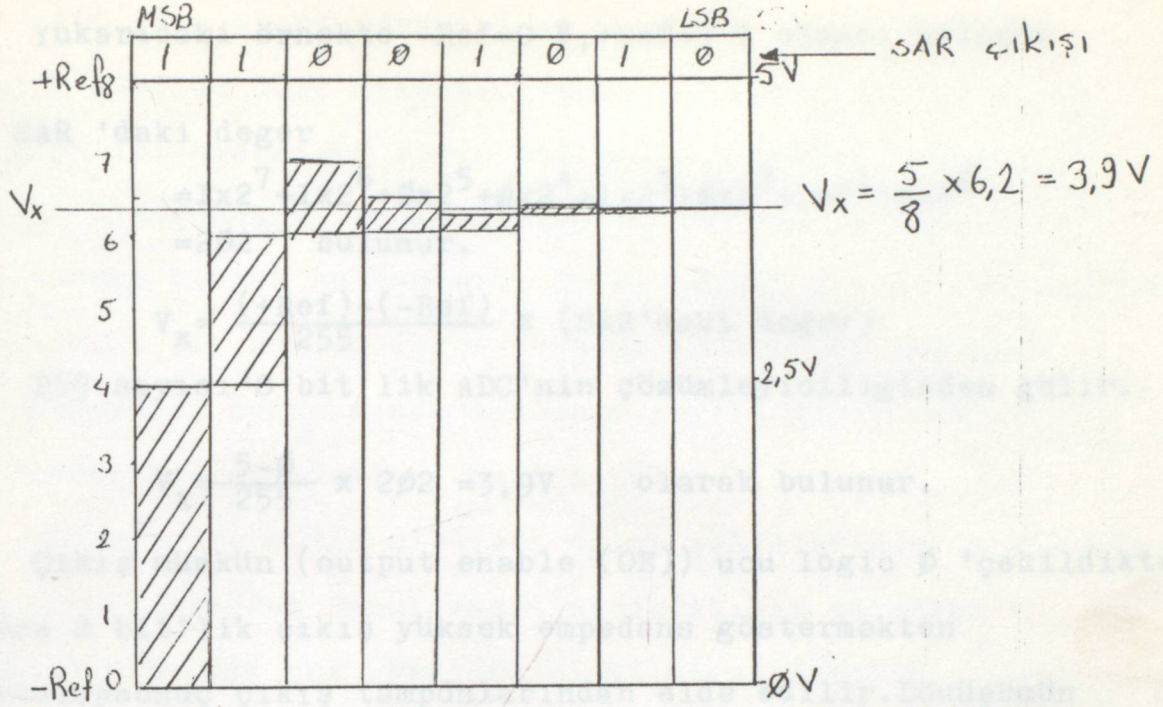
Internal block diagram of the ADC0808.

Dönüştürücünün üç temel elemanı var, ardışıl yaklaşım yazmacı (SAR), karşılaştırıcı (comparatör) ve direnç merdiveni ve anahtar ağacı dizisidir.

Bunlara ek olarak 8 analog girişinden birini seçmeye yarayan multiplexer zamanlama ve kontrol mantığı digital veri çıkmasını sağlayan "tristate" veri tamponları vardır.

SAR dönüşüme başla (start conversion) darbesinin pozitif kenarı ile reset'lenir. Eger tamamlanmamış bir dönüşüm süresince başka bir SC işareti uygulanırsa o ana kadar yapılan dönüşüm yeni istegin himayesi altında gerçekleşecektir. Dönüşüm bitti (End Of Conversion)(EOC) çıkışı dönüşüme başla girişine bağlanırsa periyodik dönüşüm zamanlaması sağlanmış olur.

Multiplexer tarafından seçilen analog giriş işareti karşılaştırıcıya gelir. Karşılaştırıcı dönüşümün tekrarı ve hassasiyetinden sorumludur. $\frac{1}{256}$ tanımlanan kararlılık ve hassasiyete ulaşabilmek için "chopper stabilized" tekniği kullanılır. Karşılaştırıcının diğer girişi anahtar ağacındadır ve giriş işaretinin bir önceki yaklaşımına eşittir.



Karşılaştırıcı V_x bilinmeyen gerilimi yukarıdaki gibi yarıya bölme metodu ile yaklaşılarak bulur. Önce MSB bit bulunur. Bilinmeyen gerilim 2,5V veya + ve - referansların tam orta değerinden büyükse MSB biti 1 ,küçükse 0 olarak karşılaştırıcının çıkışından SAR'a yerleştirilir. Bu ilk iterasyon sonucu yazmaçtaki digital degere göre gerilim bölücü dirençler vasıtası ile bilinmeyen gerilime yarıya bölme yöntemi ile yaklaşılar. Böyle 8 iterasyon sonunda 1/2 bit (LSB) hassasiyetle bilinmeyen değerin karşılığı bulunur.

$$\frac{V_{in}}{V_{(+ref)} - V_{(-ref)}} = \frac{D_x}{D_{max} - D_{min}}$$

Bilinmeyen gerilim ile digital değer arasında yukarıdaki gibi bir bağıntı vardır. Burada

V_{in} : Giriş gerilimi

V_{ref} : Referans gerilim

D_x : Girişe karşı gelen digital değer

$D_{max, min}$: Digital skalanın sınırları

yukarıdaki örnekte -Ref=0 V,+Ref=5 V olması halinde

SAR 'daki deger

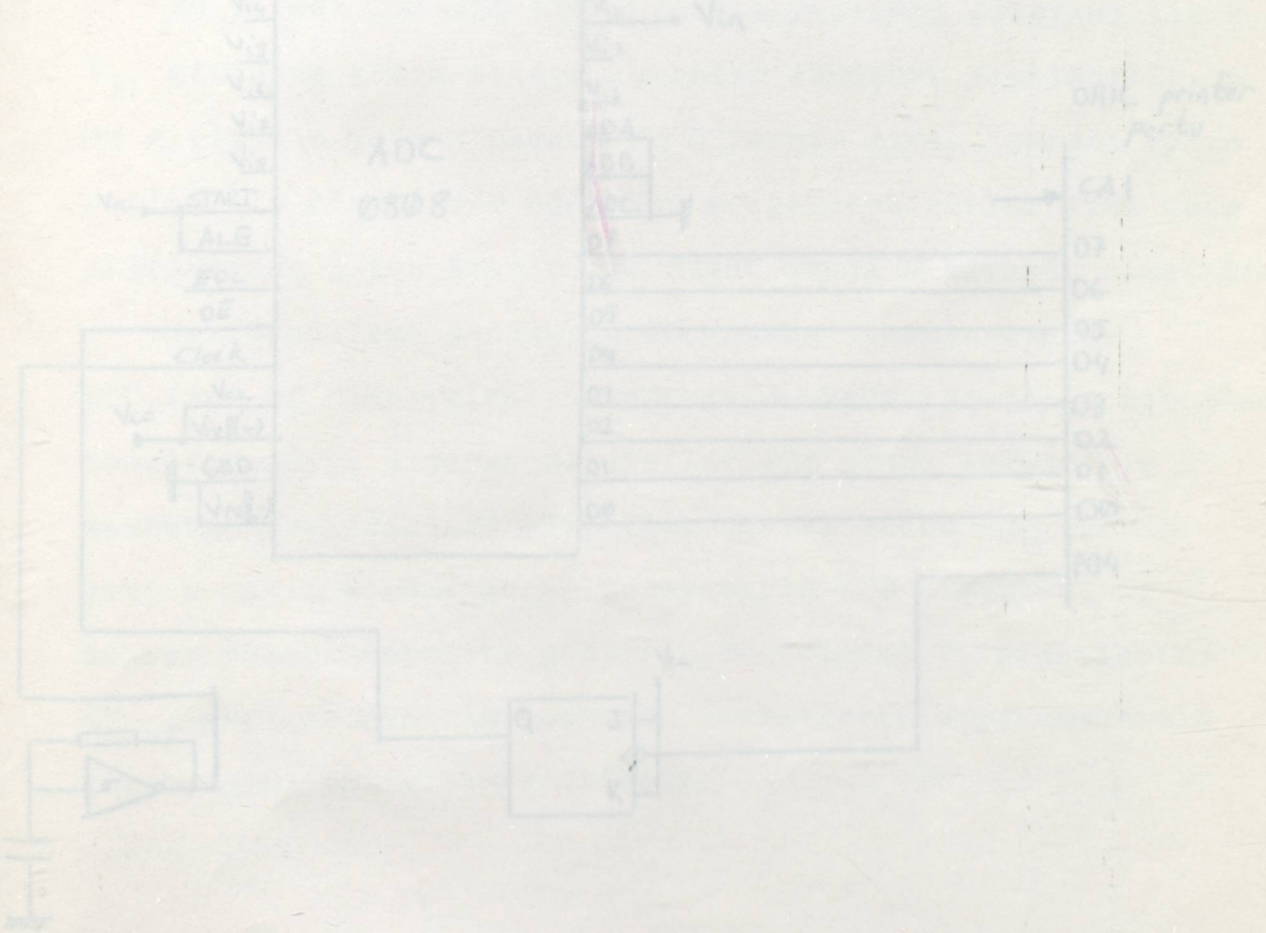
$$=1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 \\ =202 \text{ bulunur.}$$

$$V_x = \frac{(+\text{Ref}) - (-\text{Ref})}{255} \times (\text{SAR'daki deger})$$

255 sayısı 8 bit'lik ADC'nin çözümleyiciliğinden gelir.

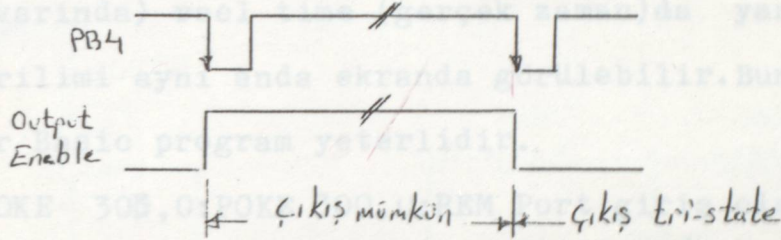
$$V_x = \frac{5-0}{255} \times 202 = 3,9V \text{ olarak bulunur.}$$

Çıkış mümkün (output enable (OE)) ucu logic 0 'çekildikten sonra 8 bit'lik çıkış yüksek empedans göstermekten kurtulup sonuç çıkış tamponlarından elde edilir. Dönüşümün tüm zamanlaması doğrudan (min 10KHz ,max 1,28MHz) clock girişi ile kontrol edilir.



Devrenin çalışması şöyledir:

B portunun 4 nolu ucu printer soketi üzerinde STROBE ismi ile adlandırılmıştır. Bu uc sürekli logic 1 konumda durur. Öric içindeki monitör program buucun yalnızca 1 periyod için sıfır konumuna çekilmesine izin verir. Böylece JK FF çıkışında low logic seviye high konuma geçer. Bunun anlamı ise ADC'nin D0-D7 veri uçlarının yüksek empedans göstermekten kurtulup aktif konuma geçmesi demektir.



Bu sırada ADA-ADB-ADC uçları L yapılmış dolayısı ile V_{il} girişine gelen analog işaretin dönüşümü yapılabilir. Bu üç girişin L yada H olmasıyla 8 farklı giriş işareti zaman paylaşımlı olarak A/D dönüşümü gerçekleştirirler. Aynı anda ALE(Address Latch Enable) ve START ucuda H konumda olmalıdır.

Her örnekleme sonucunda dönüşümün tamamlandığını EOC (End Of Conversion) ucundanda dönüşüm işleminin tamamlanıp tamamlanmadığı dışarıya bildirilir. PB4 ucuna tekrar bir L darbe gönderildiğinde Flip Flop çıkışı konum değiştirir yani H dan L konuma geçer. Dolayısıyla ADC çıkışları tekrar yüksek empedans gösterir. Son olarak V_i girişlerine $\pm V_{ref}$ gerilim sınırını aşmayan işaretlerin uygulanmasının gerektiğini hatırlatmak isteriz.

Devrenin çalışmasında açıkladıktan sonra analog işaretin bilgisayarın bellek gözüne yerleşmesi ve bunun osiloskop ekrananda görüldüğü gibi izlenmesi için bir programın çalıştırılması gereklidir. Analog işaretin hızla örneklenip belleğe yerleştirilmesi için daha hızlı çalışması nedeni ile makina dili program tarafından bu kısım gerçekleştirilir. Ekranda gösterim işleminde ise hız açısından pek önemli olmayabilir bu nedenle gösterim programı Basic yazılmıştır. Eğer giriş işaretinin frekansı çok düşükse (10Hz civarında) reel time (gerçek zaman) da yani analog giriş gerilimi aynı anda ekranda görülebilir. Bunun için basit bir Basic program yeterlidir.

```
10 POKE 300,0:POKE 300,0:REM Port giriş olarak tanımlandı  
PB4'den 0 gönderildi.
```

```
20 HIRES :REM Ekran grafik modunu 199-239 pixele ayarladı.
```

```
30 FOR X=1 TO 239
```

```
40 L=PEEK( 301):REM 301 adresindeki data L değişkenine  
Eşitlendi.
```

```
50 Y=(191-(190/255)*L)
```

```
60 CURSET X,Y,1
```

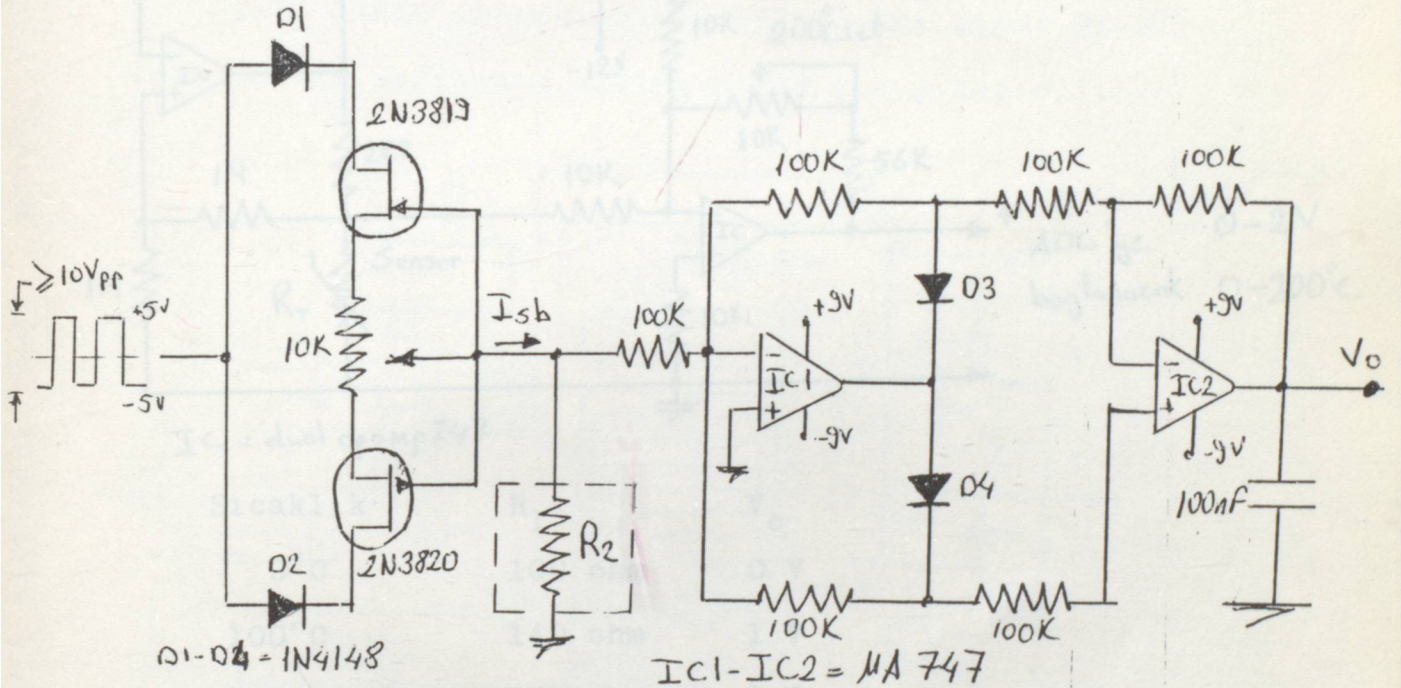
```
70 NEXT X
```

```
80 POKE 300,0:POKE 303,255:REM PB4 'den ikinci 0 gönderildi  
Port tekrar çıkış yapıldı.
```

REM konularak yapılan açıklamaların bulunduğu Basic satırları anlaşılacağı gibi önemli olup bu satırlar üzerine diğer satırları değiştirerek daha komple bir program yazılabilir. Bundan sonrası tamamen kullanıcıya açık olup amacına göre programını yazıp dilediği yerde bu arabirimi kullanabilir.

Analog girişten direnç ölçümü:

Bilgisayarımıza ADC ile analog sinyalleri digitale çevirerek girdikten sonra bunu pek çok yolla işleme imkanına sahip oluruz. Örneğin değerini bilmediğimiz bir dirençin değerini ölçen bir alet kolaylıkla bu arabirimimiz sayesinde yapılabilir. ADC 'ye dirençle orantılı bir gerilim girmek gerekir. Bunun içinde bir direnç/gerilim dönüştürücüye ihtiyaç vardır. Bu dönüştürücü aşağıdaki gibi tasarlanmıştır.



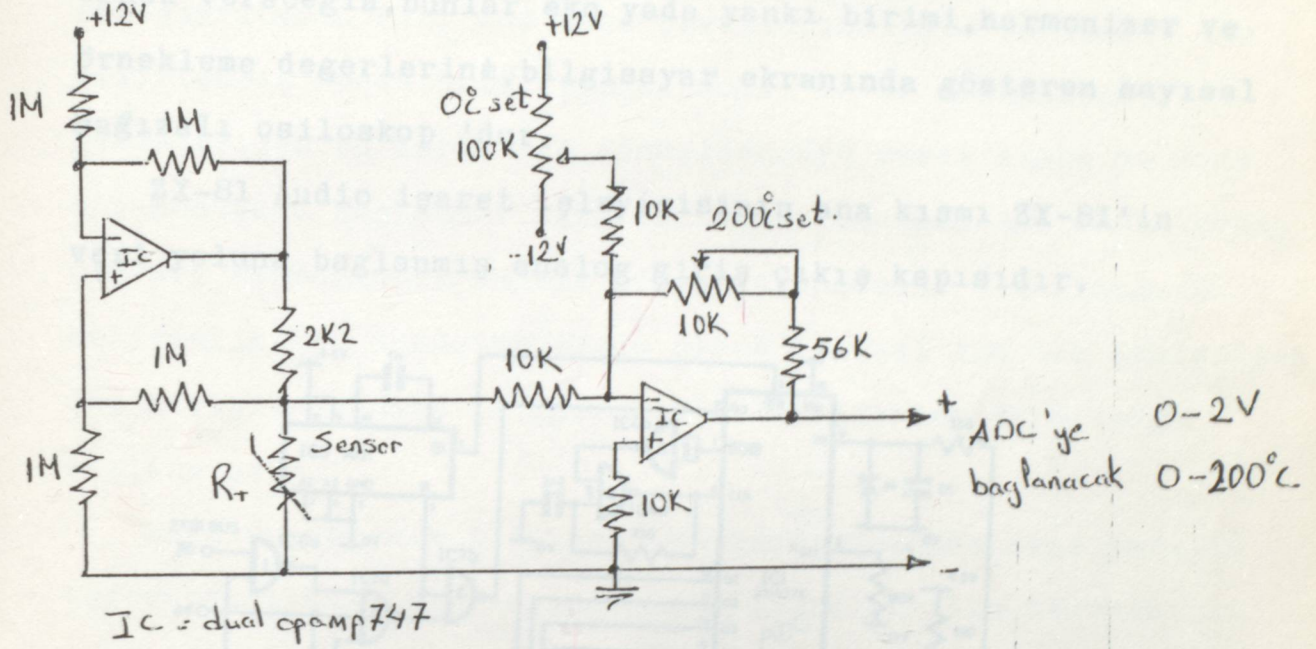
R_2 direncine sabit akım üretmek için complimenter JFET'ler kullanılmıştır. IC1 ve IC2 doğrultucu olarak çalışır ve

$$V_o = I_{sb} \times R_2$$

ifadesi ile belirlenen çıkış gerilimini temin ederler.

Sıcaklık ölçümü:

Arabirimimizle kolayca 0-200 derece arasında sıcaklık ölçer bir alet yapabiliriz. Bunun için temel malzemeler bir dual opamp birde sensör olarak PRT-100 platinum direnç yada onun yerini tutabilecek 1N4148 diodudur.



Sıcaklık	R ₁	V _o
0°C	100 ohm	0 V
100°C	140 ohm	1 V
200°C	180 ohm	2 V

İlk opamp sabit akım kaynağı ikinciside amplifikatör olarak çalışır. Bizim kullandığımız ADC' nin resolüsyonu 10mV dur. Burada verilen sıcaklığı gerilime dönüştüren biriminde 1 derece sıcaklık artışında çıkış geriliminin değişimi 10mV olduğundan yaptığımız ölçümlerin hassasiyeti 1 dereceyi sezebilecek doğruluktadır.

Sensörün ön kalibrasyonu -40° ile +400° derece arasında lineer bir direnç olarak yapılmıştır. İstenirse bilinen iki farklı dirençle sensör yer değiştirerekde kalibre edilebilir.

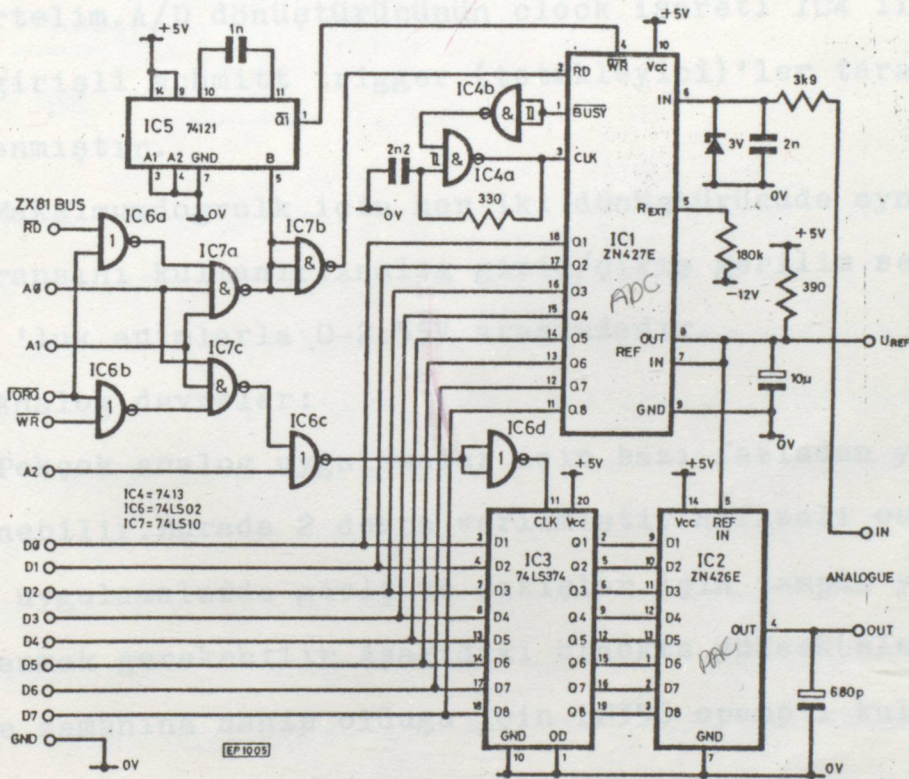
Önce 100 ohm' luk direnç kullanarak sıfıra set edici pot ile sıfır ayarı yapılır. (R_T=100 ohm , 0° C' de)

Sonrada 200° C için kazancı değiştiren ikinci pot ile

ZX-81 mikrobilgisayarına Analog giriş/çıkış:

Bu bölümde alışılmamış audio (ses) efektlerivermek için ZX-81 'in audio işaret işleyicisinin kalbi olarak nasıl kullanılacağını anlatacağız. Burada üç uygulama alanını örnek vereceğiz, bunlar eko yada yankı birimi, harmonizer ve örnekleme değerlerinin bilgisayar ekranında gösteren sayısal hafızalı osiloskop 'dur.

ZX-81 Audio işaret işleyicisinin ana kısmı ZX-81'in veri yoluna bağlanmış analog giriş çıkış kapısıdır.



8 bit ayrıştırma ile kaliteli ses üretmek için 80KHz üstünde bir örnekleme hızı temin eden ZN426 D/A dönüştürücü, ve ZN427 A/D dönüştürücüleri kullanılmıştır. Dahili I/O işlemleri ile çakışmaları önlemek için Z80 'in 3 nolu I/O portunu kullanılmıştır. IC6 ve IC7 I/O işlemlerini çözümler ve IC7b ve IC6c kapılarının çıkışlarında dışardan bir veri içeri girmesininmi yoksa dışarı veri çıkışınınmi istendiğini dönüştürücülere söyler.

Çıkış istegi darbesi, ZX-81'in veri yoluna bağlı olan 8 bit'lik latch'de o an veri yolundaki bilgiyi kilitlen

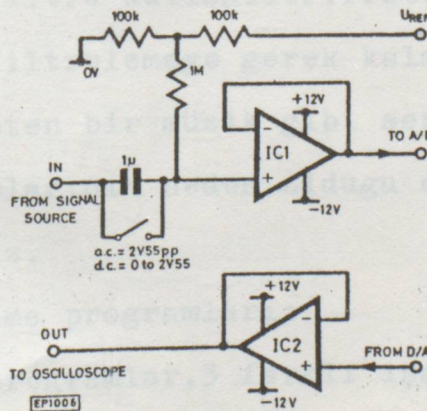
Bu devrenin çıkışlarında D/A dönüştürücüyü sürer.

Giriş istemi darbesi ki buda IC7b kapısı çıkışında oluşturul verinin ZX-81 tarafından okunmasını sağlayarak A/D dönüştürücünün (IC1'in) data çıkışlarına izin verir. IC5 ile belirlenmiş monostable (tek kararlı) devre tarafından geciktirilen bu darbe A/D dönüştürücüyü reset etmek ve yeni bir dönüşümü başlatmak için kullanılır. Burada A/D dönüştürücü den veri okumanın bir önceki giriş işlemlerine ait olduğunu belirtelim. A/D dönüştürücünün clock işareti IC4 ile gösterilen iki girişli schmitt trigger (tetikleyici)'ler tarafından sağlanmıştır.

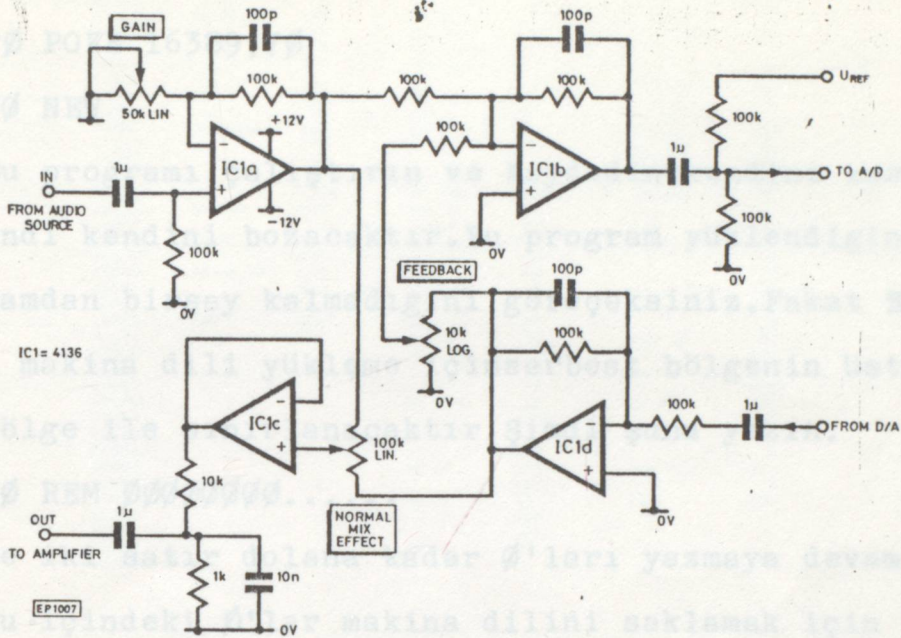
Maksimum doğruluk için her iki dönüştürücüde aynı gerilim referansını kullanır. Analog giriş/çıkış gerilim seviyesi 10mV 'luk adımlarla 0-2.55V arasındadır.

Analog devreler:

Pekçok analog uygulamalar için bazı fazladan yükselticiler istenebilir. Burada 2 devre verilmiştir. Hafızalı osiloskop gibi uygulamalarda giriş ve çıkışlar için tampon yükseltici kullanmak gerekebilir. Aşağıdaki örnekte yüksek (slew rate) düşme zamanına sahip olduğu için LF356 opamp'ı kullanılmıştır.



The diagram shows a four-channel audio mixer circuit using the IC1 4136. The circuit includes four op-amp channels (IC1a, IC1b, IC1c, IC1d) with various feedback and input components like resistors, capacitors, and potentiometers. It features a 'GAIN' control, a 'FEEDBACK' control, and a 'NORMAL MIX EFFECT' control. The output is connected to an amplifier and an A/D converter.



Profesyonel analog-digital sistemler,örnekleme işleminin oluşturduğu distorsiyondan kaçınmak için yüksek dereceli alçak geçiren filtre kullanılır.Pratikte burada verilen basit devrelerfiltrelemeye gerek kalmadan çalışır;elektro gitar veya teypten bir müzik gibi ses kaynakları için yansıma frekanslarının neden olduğu distorsiyonlardan hiçbiri duyulmaz.

Aşğıdaki programlar,3 farklı işaret işleme uygulaması için ses işaret işleme devresini kullanırlar.Bunlar 16K'lık RAM'i olan Z81 'le kullanılmak üzere tasarlanmıştır.

Makina dili için bellekte yer ayırmak için aşağıdakileri yazın.

1Ø SAVE "SET"

2Ø POKE 16389,7Ø

3Ø NEW

Bu programı çalıştırın ve kaydedin;kendini kasete kaydedecek ve kendi kendini bozacaktır.Bu program yüklendiğinde programdan birşey kalmadığını göreceksiniz.Fakat BASIC alanı makina dili yükleme için serbest bölgenin üstünde 1,5K'lık bir bölge ile sınırlanacaktır.Şimdi şunu yazın.

1Ø REM ØØØØØØØØØØ.....

ve iki satır dolana kadar Ø'ları yazmaya devam edin.REM komutu içindeki Ø'lar makina dilini saklamak için kullanılır.

Sonra makina dili programları yazmak için poke deyimi ile bir çevrim oluşturun.Her uygulama için yazılacak sayılar aşağıdaki tablo ile assembly dili mnemonic'leriyle birlikte verilmiştir.Bununla birlikte istenen parametreleri hesaplayan Basic programda makina dili program içine Poke ile bu parametreleri yerleştirir.Makina dili çevrimlerden N tuşuna basılarak çıkılabilir ve basic'e geçilebilir.

	HARMONIZER:	DIGITAL STORAGE OSCILLOSCOPE
...string of poked characters...	10 REM ...string of poked characters...	10 REM ...string of poked characters...
A=16514	20 CLS	20 FAST
T "DELAY TIME:...(MILLISEC)," (MAX 500)"	30 FAST	30 PRINT "DC TRIGGERING LEVEL:...", "10 TO 2.05 VOLT"
T P	40 LET A=16514	40 INPUT U
>500 THEN GOTO 60	50 PRINT "DELAY RANGE:...(MILLISEC)," (MAX 500)"	50 LET U=255-INT(1000U+0.5)
P= 128-INT(P/B.61+0.5)	60 INPUT P	60 POKE 16515,U
A+2,P	70 IF P>500 THEN GOTO 60	70 GOSUB 300
E A+21,P	80 LET P=128-INT(P/B.61+0.5)	80 GOTO 170
P=USR A	90 POKE A+4,P	90 LET L=USR 16514
Ø Ø	100 POKE A+7,P	100 LET L=USR 16518
	110 POKE A+28,P	110 LET K8=INKEY\$
	120 POKE A+35,P	120 IF K8="" THEN GOTO 100
	130 CLS	130 IF K8="N" THEN GOTO 170
	140 PRINT "DOWN:Ø UP:1"	140 IF K8="S" THEN GOSUB 400
	150 INPUT P	150 IF K8="B" THEN GOSUB 500
	160 IF P=Ø THEN GOTO 200	160 GOTO 100
	170 IF P=1 THEN GOTO 400	170 CLS
	180 GOTO 130	180 PRINT "MENU OPTIONS:..." B: SAMPLE..." D: DISPLAY..." E: EXIT"
	200 POKE A+21,Ø	190 PRINT "IN DISPLAY MODE:..." S: SHIFT "<..." B: SHIFT ">"
	210 GOSUB 600	200 PRINT " N: MENU"
	220 LET X=INT(256/P+0.5)	210 PAUSE 33000
	230 GOTO 800	220 POKE 16437,255
	400 POKE A+21,19	230 LET I8=INKEY\$
	410 GOSUB 600	240 IF I8="B" THEN GOTO 90
	420 LET X=INT(256*(P-1)+0.5)	250 IF I8="" THEN STOP
	430 GOTO 800	260 GOSUB 300
	600 PRINT "TRANSPOSE:...(SEMITONES)"	270 GOTO 100
	610 INPUT P	300 CLS
	620 LET P=(2**((1/12))*P)	310 PRINT "SAMPLE LENGTH:...(T(215 MILLISEC))"
	630 RETURN	320 INPUT T
	800 IF X>255 THEN LET X=255	330 LET T=INT(1/3.712+0.5)
	810 POKE A+1,X	340 LET T=70
	820 LET P=USR A	350 IF T>128 THEN GOTO 300
	830 GOTO Ø	360 GOSUB 400
		370 RETURN
		400 IF T<70 THEN RETURN
		410 LET T=T-1
		420 GOSUB 400
		430 RETURN
		500 IF T>128 THEN RETURN
		510 LET T=T-1
		520 GOSUB 400
		530 RETURN
		600 POKE 16540,T
		610 POKE 16547,T+T
		620 RETURN

Data noted by * will be changed by POKE statements.

ADDRESS:	ECHO:	HARMONIZER:	STORAGE SCOPE:
16514	ld hl,NN 33	ld c,N 14	ld c,N 14
16515	0	1718	1008
16516	728	ld de,NN 17	ld b,N 6
16516	ld b,N 6	0	1
16517	3	1208	dj nz 16
16518	ld a, (hl) 126	ld hl,NN 33	-2
16519	out N,a 211	0	in a,N 219
16520	3	1208	3
16521	in a,N 219	in a,N 219	add c 129
17522	3	3	jr nc 48
16523	ld(hl,a) 119	ld(hl,a) 119	-9
16524	dj nz 16	inc hl 35	ld hl,NN 33
16526	-2	ld a,(de) 26	0
16527	inc de 19	out N,a 211	70
16528	inc hl 35	3	in a,N 219
16529	dec h 37	ld a,b 120	3
16530	inc h 36	add c 129	ld(hl,a) 119
16531	jp p,NN 242	ld b,a 71	inc hl 35
16532	133	jr nc 48	ld a,h 124
16533	64	1	cp N 254
16534	ld h,N 38	inc de 19	128
16535	728	inc de 198	jr nz 32
16536	in a,N 219	dec d 21	-9
16537	2	inc d 20	ret 201
16538	cp N 254	jp p,NN 242	ld hl,NN 33
16539	119	159	0
16540	jp nz,NN 194	64	708
16541	133	ld d,N 22	ld a, (hl) 126
16542	64	1208	out N,a 211
16543	ret 201	dec h 37	3
16544		inc h 36	inc hl 35
16545		jp p,NN 242	ld a,h 124
16546		138	cp N 254
16547		64	1288
16548		ld h,N 38	jr nz 32
16549		1208	-9
16550		in a,N 219	ld a,N 62
16551		2	0
16552		cp N 254	out N,a 211
16553		119	3
16554		jp nz 194	ret 201
16555		138	
16556		64	
16557		ret 201	
16558			

Yankı programı:

Sese işaret işlemcisini yankı birimi gibi kullanan program bir önceki sayfada gösterilmiştir. Programın çalışmasını anlamak için saat kadranı üzerine yerleştirilen bellek hücrelerini düşünün. Saatin yelkovanı sabit hızla dönmektedir.

yelkovan bir hücreyi gösterdiğinde bu hücre içindeki sayı önce D/A dönüştürücüye aktarılır, A/D dönüştürücüden okunan değer bu hücreye yerleştirilir. Her örnek böylece çevrim içindeki hücre sayısı kadar ve örnekleme hızına bağlı olarak bir süre gecikir. Örnekleme hızı 30KHz'de sınırlıdır, fakat çevrim içindeki hücre sayısı 500milisaniyeye kadar farklı geçikme için değiştirilebilir.

Harmonizör programı:

Bu programla giriş işaretinin gerçek zaman adımı değişimi oluşturulur ve yankı programı gibi çalışır. Farklı hızlarda dönen iki yelkovanlı bir saat düşünün yelkovanların biri her hücreden sayıları yazar ve D/A dönüştürücüye aktarır, diğeri sayıları A/D dönüştürücüden hücrelere yazar. İki hızın oranı girişin çıkışa adım oranını belirler.

Hafızalı scope programı:

Bu programla dalga şeklini okuma ve seçilmiş bölümleri tarayan sayısal hafızalı scope olayı gerçekleştirilmiştir.

Basic program bir tetikleme seviyesi ve zaman penceresi seçimini sağlar. Giriş tetikleme seviyesine erişince örnekleme 69KHz'lik bir örnekleme hızı ile başlar. 14,5K lık uygun bellek 215ms'de dolar. Bundan sonra gösterme modu başlar ve zaman penceresi istenen bölümü göstermek için hareket eder. Bu tür bir osiloskop konuşma ve müzik gibi dalga şekillerini göstermekte ve araştırmakta idealdir.

Paydalanılan dergiler:

Microsystems, Oric Owner, Elektor, Your computer, Wireless world, Practical Electronics, Radio and Electronics, Electronics Today International

Paydalanılan Kataloglar:

Motorola data sheets

Rockwell data sheets

National semiconductor application notes

Siemens microcomputer application notes

Intel microcomputer application notes

KAYNAKLAR

PEATMAN ,J.B.,Microcomputer-Based Design.7 th printing
McGraw Hill book company 1982

MAUNDER ,B.,The Oric-1 Companion ,Linsac Inc. 1983

RODNAY ,Z and Lesea ,A.,Microcomputer interfacing
techniques,Sybex book company.1982

RODNAY ,Z.,Advanced 6502 programming.Sybex book company.1982

WHEWELL ,L.,Oric Advanced User Guide.Adder publishing.1984

Faydalanılan dergiler:

Microsystems,Oric Owner,Elektor,Your computer,

Wireless World,Practical Electronics,Radio and Electronics
world,Electronics Today International

Faydalanılan Kataloglar:

Motorola data sheets

Rockwell data sheets

National Semiconductor aplication notes

Siemens microcomputer components data catalog 1984

Intel microcomputer components catalog

