

MİKROİŞLEMCİLER DERS NOTU

Kaynaklar :

Mikroişlemcilere Giriş: Assembler ile Yazılım ve Arayüz, *Mehmet Bodur, EMO Yayınları*

THE INTEL MICROPROCESSORS, Eighth Edition, *BARRY B. BREY*

MİKROİŞLEMCİ VE MİKRODENETLEYİCİLER, *MEB Yayınları*, MEGEP sitesinden

İnternet üzerinden konulara göre arama yaparak birçok dökümana ulaşabilirsiniz.

GİRİŞ

Mikroişlemci, bir işlemi kontrol etmek için ya da cihazları açıp-kapamak için kullanılan bir programlanabilir lojik cihaz olarak tanımlanabilir. Diğer taraftan, Mikroişlemci, temelde bilgisayarın tüm işlemleri yapmasını sağlayan, halk tabiri ile bilgisayarın beyni olarak nitelendirilebilecek **hesaplama**, **karar verme** ve **yönetim** mekanizmasıdır. Mikroişlemciler bilgisayar programlarının yapmak istediği bütün işlemleri yerine getirdiği için, çoğu zaman merkezi işlem ünitesi (CPU-Central Processing Unit) olarak da adlandırılır.

Önceleri ancak haberleşme, uzay ve havacılık gibi alanlarda kullanım bulan **mikroişlemci** kavramı hızla bilgisayar donanımı, endüstriyel kontrol sistemleri gibi alanlarda geniş uygulama alanı buldu. İşlemci üretim teknolojisinin gelişmesi ve ucuzlamasıyla günümüzde mikroişlemci kullanılmayan alan kalmadı, hemen hemen hayatımızın her yerinde yer almaktadır; bilgisayarlar, tabletler, cep telefonları, akıllı sistemler, CNC tezgâhları, vb.

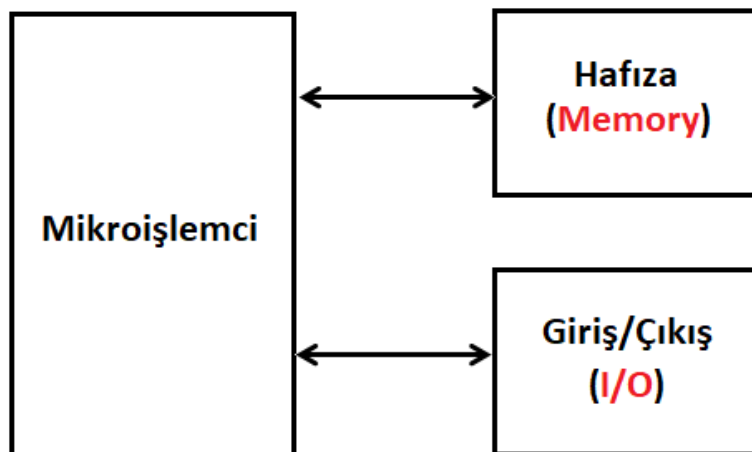
Boyutları çok küçük olmasına rağmen içerisinde yüz binlerce, milyonlarca hatta milyarlarca elektronik devre elemanı bulunduran mikroişlemci, **bit** denilen ikili (binary) sayılarla (1 ve 0) haberleşir. Mikroişlemciler, **makine dili** (*machine language*) adı verilen ikili formdaki sabit bir komut setine sahiptir. Fakat 0 ve 1'ler ile işlem yapmak çok zordur, çünkü bit sayısı arttıkça, kontrol edebilmemiz zorlaşır. Komutların ikilik kodlarını sürekli bir tablodan bakmamız gerekir. Bu nedenlerden dolayı bu ikilik komutlar (binary instruction) **mnemonics** adı verilen kısaltma isimlerle adlandırılırlar. Mnemonicsler **assembly dilini** oluştururlar. Bu dersimizde 8086 mikroişlemcinin assembly dili anlatılacaktır.

CPU	Yıl	İlave Komut	Komut Sayısı	Hız (MHz)
8086	1977		81	10
80186	1982	6	87	25
80286	1982	16	103	25
80386	1985	31	134	40
80486	1989	3	137	100

CPU	Transistör Sayısı	Yıl	Firma
Intel 4004	2300	1971	Intel
Intel 8008	2500	1972	Intel
Intel 8080	4500	1974	Intel
Intel 8088	29,000	1979	Intel
Intel 80286	134,000	1982	Intel
Intel 80386	275,000	1985	Intel
Intel 80486	1,200,000	1989	Intel
Pentium	3,100,000	1993	Intel
AMD K5	4,300,000	1996	AMD
Pentium II	7,500,000	1997	Intel
AMD K6	8,800,000	1997	AMD
Pentium III	9,500,000	1999	Intel
AMD K6-III	21,300,000	1999	AMD
AMD K7	22,000,000	1999	AMD
Pentium 4	42,000,000	2000	Intel
Barton	54,300,000	2003	AMD
AMD K8	105,900,000	2003	AMD
Itanium 2	220,000,000	2003	Intel
Itanium 2 with 9MB cache	592,000,000	2004	Intel
Cell	241,000,000	2006	Sony/IBM/Toshiba
Core 2 Duo	291,000,000	2006	Intel
Core 2 Quad	582,000,000	2006	Intel
G80	681,000,000	2006	NVIDIA
POWER6	789,000,000	2007	IBM
Dual-Core Itanium 2	1,700,000,000	2006	Intel
Quad-Core Itanium Tukwila (processor) ^[1]	2,000,000,000	2008	Intel
8-Core Xeon Nehalem-EX	2,300,000,000	2010	Intel
10-Core Xeon Westmere-EX	2,600,000,000	2011	Intel
62-Core Xeon Phi	5,000,000,000	2012	Intel

Mikroişlemciler

Bir mikroişlemci çok amaçlı programlanabilir bir lojik cihazdır. **Hafıza** (memory) olarak isimlendirilen bir depolama aygıtından ikilik komutları (binary instruction) okur, ikilik bilgiyi **giriş** (input) olarak kabul eder ve bu bilgiyi çözerek, ilgili komuta göre işler ve sonucu **çıkış** (output) olarak üretir. Tipik bir programlanabilir makine; mikroişlemci, hafıza ve giriş/çıkış (I/O) olmak üzere en az 3 bileşenden oluşur.



Programlanabilir bir makine (mikrobilgisayar)

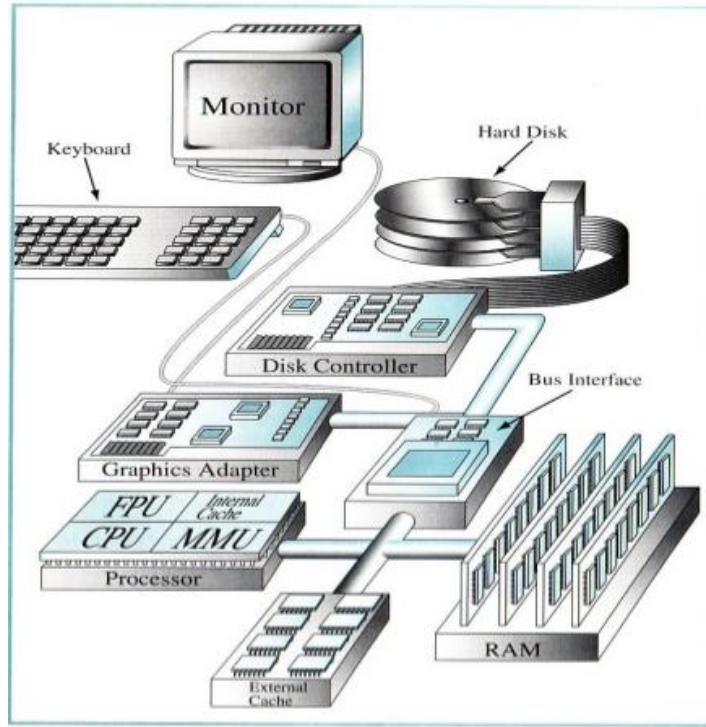
Bu üç bileşen, verilen bir görevi yerine getirebilmek için beraber çalışırlar ve birbirini etkilerler. Böylece bir sistem meydana getirirler. Bu sistemin fiziksel bileşenleri **donanım** (hardware) olarak adlandırılır. Mikroişlemcinin bir görevi yerine getirebilmesi için yazılan komutlar grubuna **program** ve programlar grubuna da **yazılım** (software) adı verilir.

Mikroişlemci uygulamaları temel olarak iki grupta toplanır;

- 1- Yeniden programlanabilir sistemler (Reprogrammable Systems)
- 2- Gömülü sistemler (Embedded Systems)

Yeniden Programlanabilir Sistemler

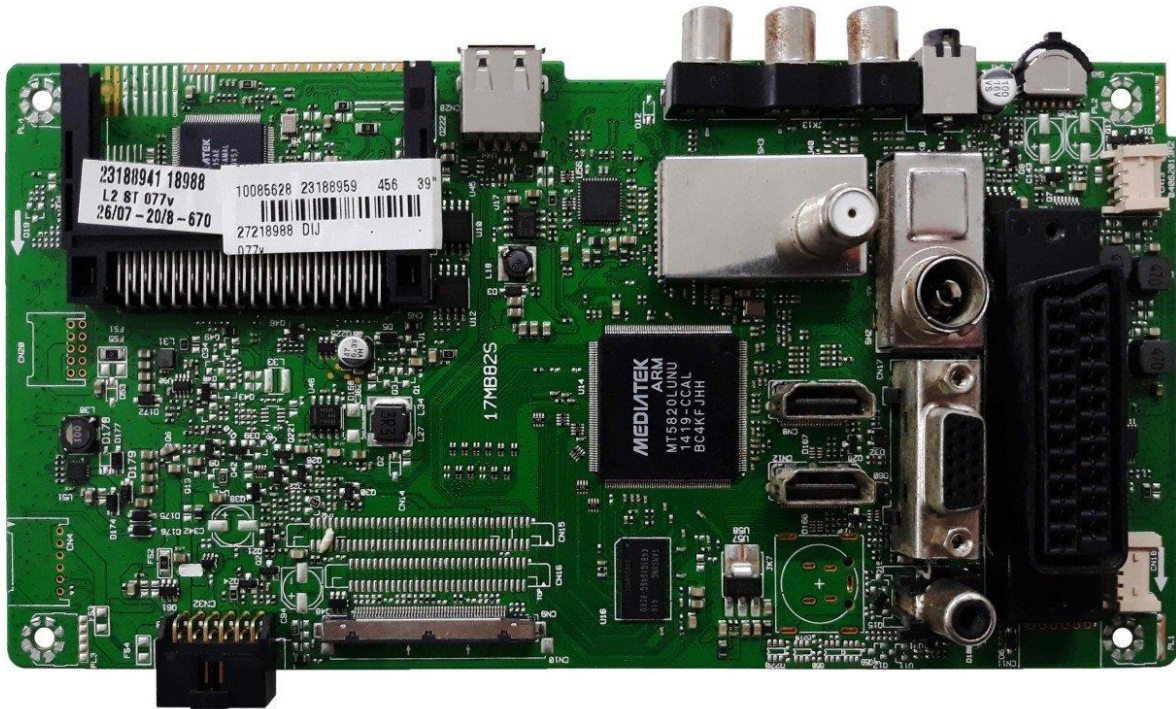
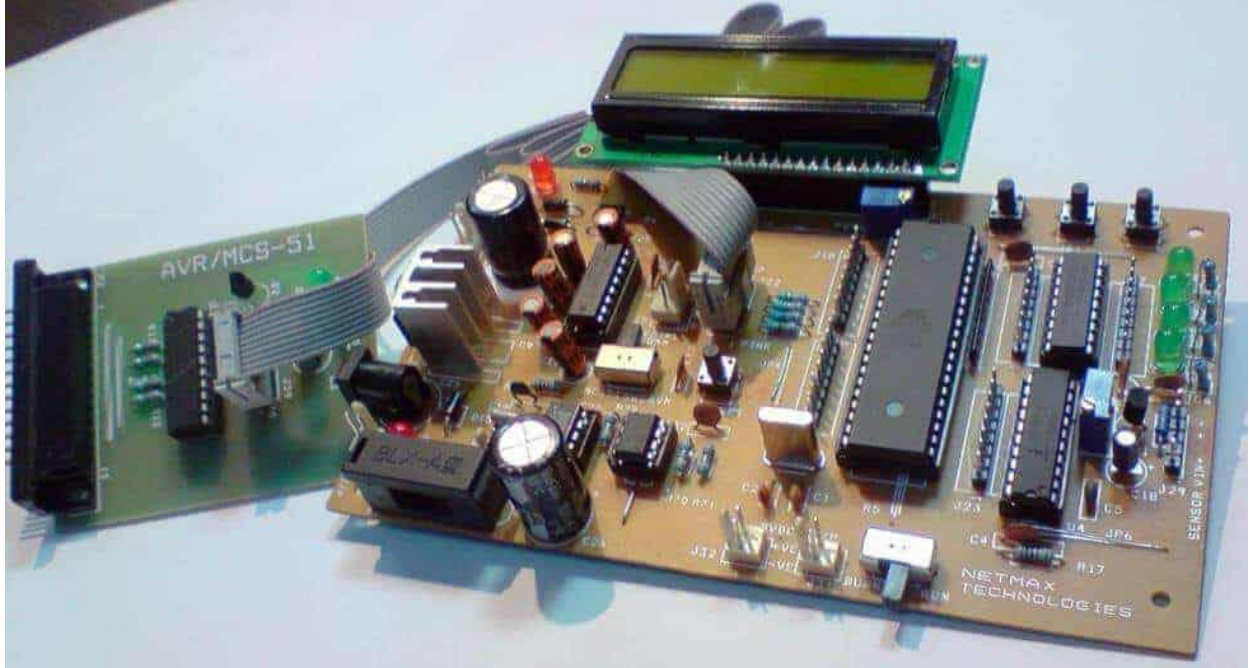
Mikrobilgisayarlar gibi yeniden programlanabilir sistemlerde, mikroişlemci, hesaplama ve bilgi işleme (data processing) işlemleri için kullanılır. Bu sistemler, çok büyük miktardaki bilgiyi işleyebilen genel amaçlı mikroişlemcileri, bilgi depolama cihazlarını (hard disk, vb.) ve giriş çıkış cihazlarını (klavye, monitör, yazıcı, vb.) ihtiva ederler. Örnek olarak kişisel bilgisayar (PC)



PC Bir Bilgisayar Temel Bileşenleri

Gömülü Sistemler

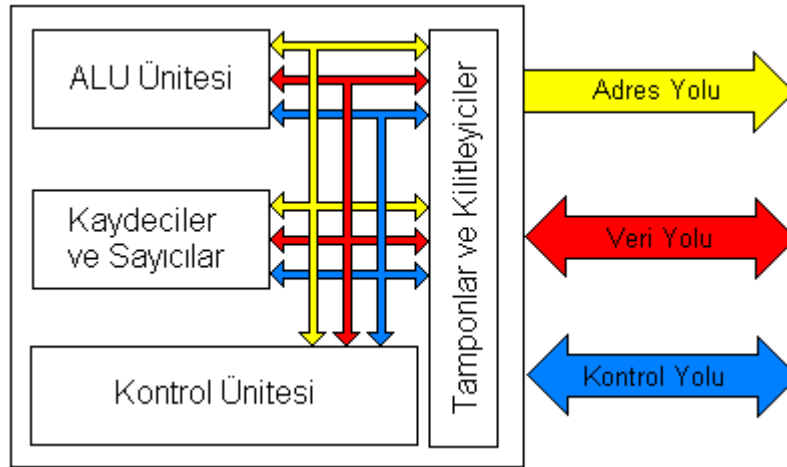
Bu sistemlerde, mikroişlemci bir ürünün parçasıdır ve kullanıcının tekrar programlamasına müsaade edilmez. Örneğin fotokopi makinası, çamaşır makinası, otomobil kumanda panelleri. Bu sistemlerde genellikle bütün bileşenleri bir chip üzerinde bulunduran **mikrodenetleyiciler** kullanılır.



TV Anakartı

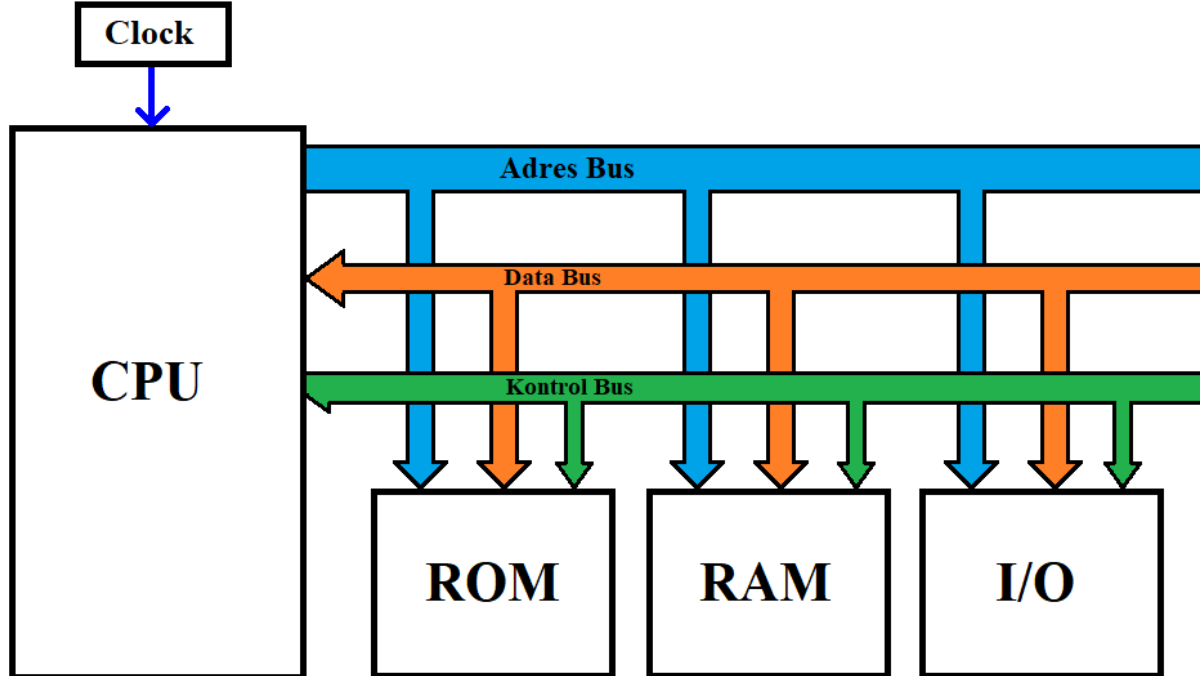
Mikroişlemci ve Mikrobilgisayar

Bir mikroişlemcinin üzerinde mantık kapılarının bileşiminden oluşan ve çeşitli işlevleri yürütmeye yarayan birimler bulunur. Bu birimler arasında **aritmetik-mantık birimi (ALU)**, **kontrol birimi**, **girdi-çıkı** ve küçük bir miktar **bellek (cache)** vardır.



Mikroişlemci iç blok diyagramı

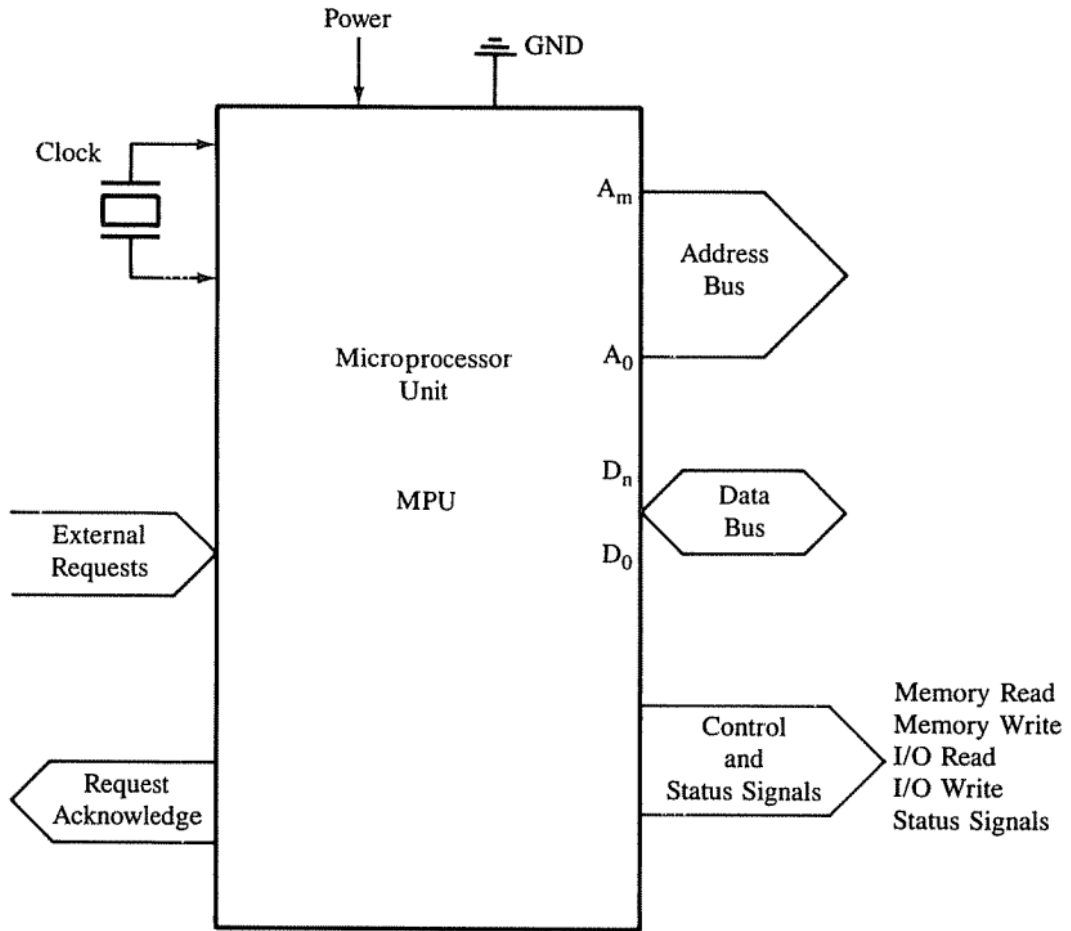
Mikrobilgisayarlar kullanım amacına göre uygun bir donanım içerirler. Tasarımdaki önemli sorunlar, kullanılacak olan bellek elemanlarının, Giriş/Çıkış kapılarının vs. amaca uygun biçimde adreslenebilmesidir. Açıkçası RAM/ROM ve değişik amaçlı Giriş/Çıkış kapılarına bellek haritasında uygun adreslerin karşı düşürülmesi sağlanmalıdır.



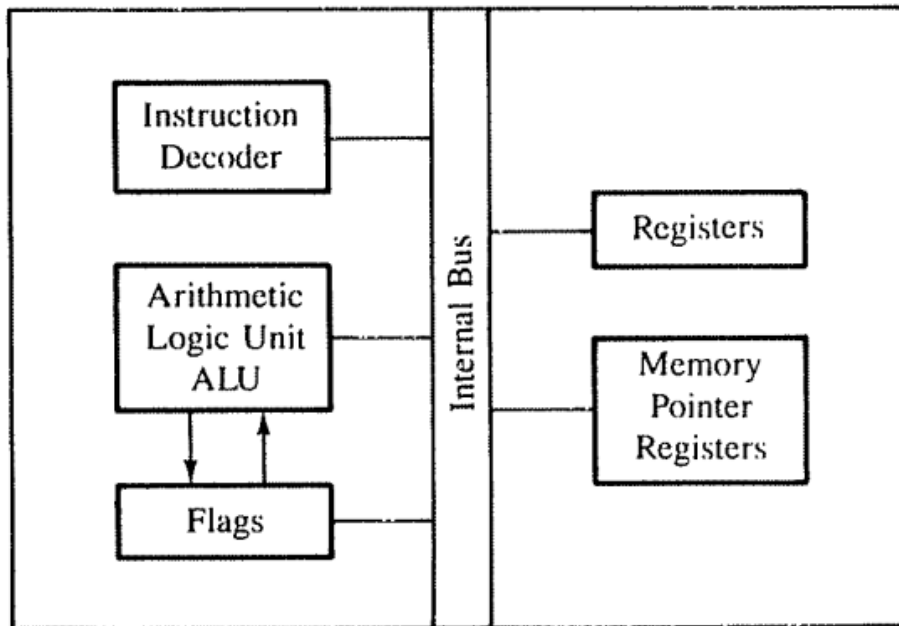
Mikrobilgisayar ve bus yapısı

Yukarıdaki şekilde basit bir mikrobilgisayar sisteminin yapısı verilmiştir.

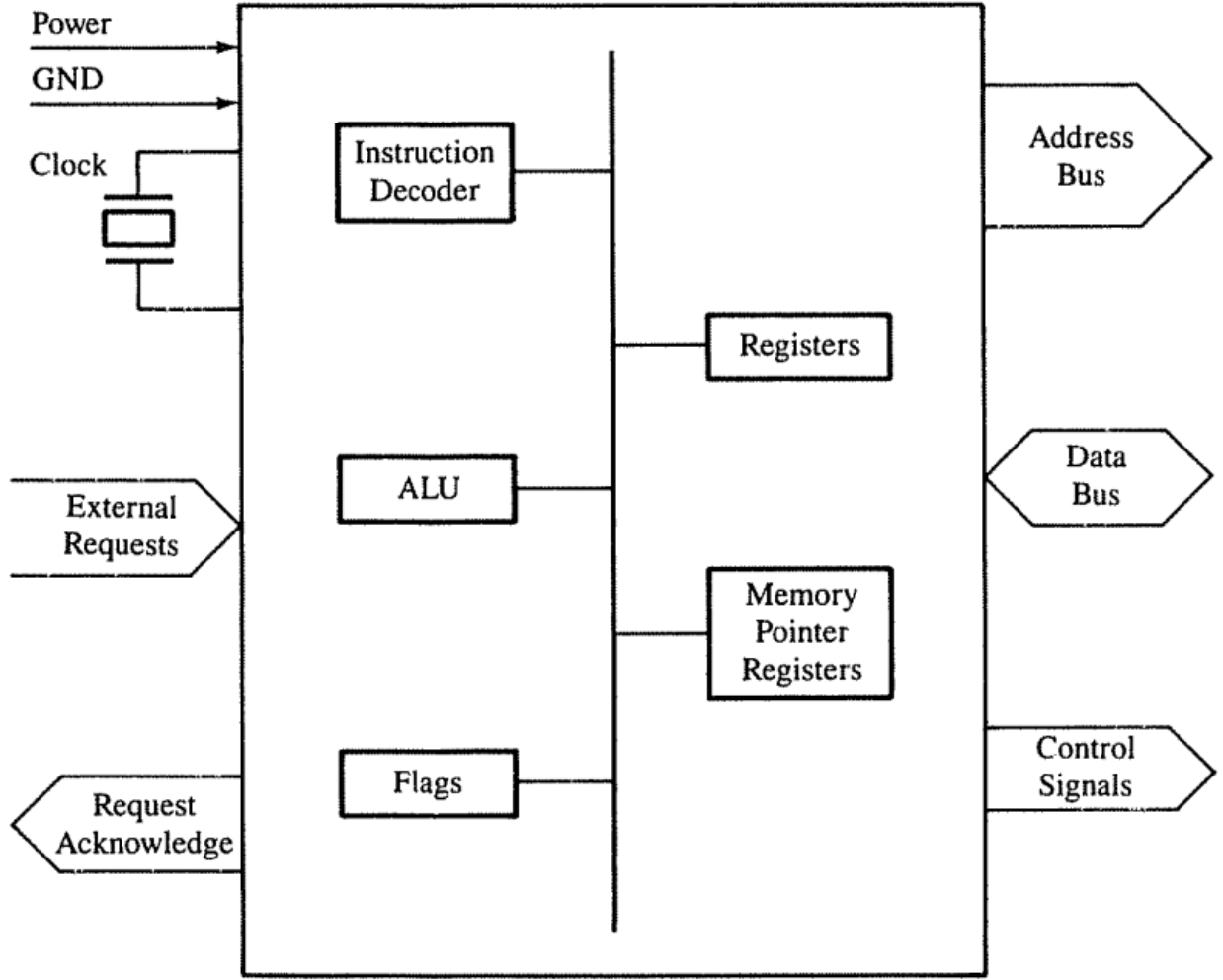
Genel Mikroişlemci Yapısı



Mikroişlemci için gerekli dış sinyaller



uP'nin verileri işleme gereksinimlerinin bu açıklaması, F2'de gösterilen basitleştirilmiş bir blok diyagramda özetlenebilir.



Mikroişlemcinin özetlenmiş dış ve iç blok diyagramı

Bellek ve G/Ç cihazı ile **harici** iletişim kurmak için MPU aşağıdakilere sahip olmalıdır:

- 1- Bir bellek kaydedici ve bir G/Ç'nin adresini göndermek için **Adres yolu**.
- 2- MPU ile bellek ve G/Ç cihazları arasında veri aktarımı için **Veri yolu**.
- 3- İşlemlerini tanımlamak ve zamanlama sağlamak için **kontrol sinyalleri**.
- 4- MPU işlemlerini kesmek için **Harici İstek** sinyal hatları.
- 5- Çevre birimlerinin isteklerine yanıt vermek için **İstek Onay** sinyalleri.
- 6- Devreleri çalıştırmak için **güç**, zamanlama için **saat sinyalleri**.

Verileri **dahili** olarak işlemek için MPU aşağıdakileri içermelidir:

1. Alınan ikili bilgilerin kodunu çözmek için **Komut Kodçözücü** (Instruction Decoder)
2. İkili verileri depolamak için **Kaydedici** (Register)
3. Bellek kaydedicilerini adreslemek için **Bellek İşaretçi Kaydediciler** (Memory Pointer Reg)
4. Aritmetik ve mantık işlemlerini gerçekleştirmek için **ALU**.
5. Karar verme için veri koşullarını gösteren **bayraklar** (Flags) (flip-floplar).

TERİMLER

CPU (CENTRAL PROCESSING UNIT): Merkezi İşlem Birimi; programla sürülen saat modlu ardışıl bir devre olup görevleri:

- 1) Program emirlerini almak, çözümlemek ve uygulamak
- 2) BUS' larla donatılmış sistemi denetlemektir. En basit bir bilgisayar, bir CPU ve bir bellekten oluşur. BUS, CPU ve Bellek arasındaki etkileşimleri sağlar.

BUS: CPU ve diğer birimler arasındaki adres, veri ve kontrol sinyallerinin etkileşimini sağlayan iletken hatlardır, yollardır. Adres, Veri ve Kontrol Yolu olmak üzere üç farklı BUS vardır.

Adres BUS (Adres Yolu)

Verinin okunacağı veya verinin yazılacağı adres bölgesini temsil eden bilgilerin taşınmasında kullanılan hatlar, **adres yolu** olarak isimlendirilir. Adres yolu, tek yönlüdür ve paralel iletişim sağlayacak yapıdadır. Bir tek BUS bir CPU tarafından sürülüp birçok cihaz tarafından kullanılabilir. Tüm elemanlara aynı hat ile bağlıdır. Yani işlem yapılacak çipi seçmek için kullanılırlar.

Mikroişlemcinin adresleme kapasitesini belirler. Yani kullanılabilecek *hafıza (ram) miktarını* belirler. 2^n ; **n** adres hat sayısıdır.

20 bit adres hattı varsa $2^{20} = 1$ MB hafızayı destekler.

30 bit adres hattı varsa $2^{30} = 1$ GB hafızayı destekler.

Data BUS (Veri Yolu)

Mikroişlemciden çevre birimlerine ya da bu birimlerden işlemciye veri aktarmada kullanılan çift yönlü hatlar, **veri yolu** olarak isimlendirilir.

Mikroişlemciler veri yolu genişliği göre isimlendirilirler; 8 bit, 16 bit, 32 bit, 64 bit işlemci vb. Veri yolu genişliği, mikroişlemcinin aynı anda işleyebileceği veri sayısını belirler.

Kontrol BUS (Kontrol Yolu)

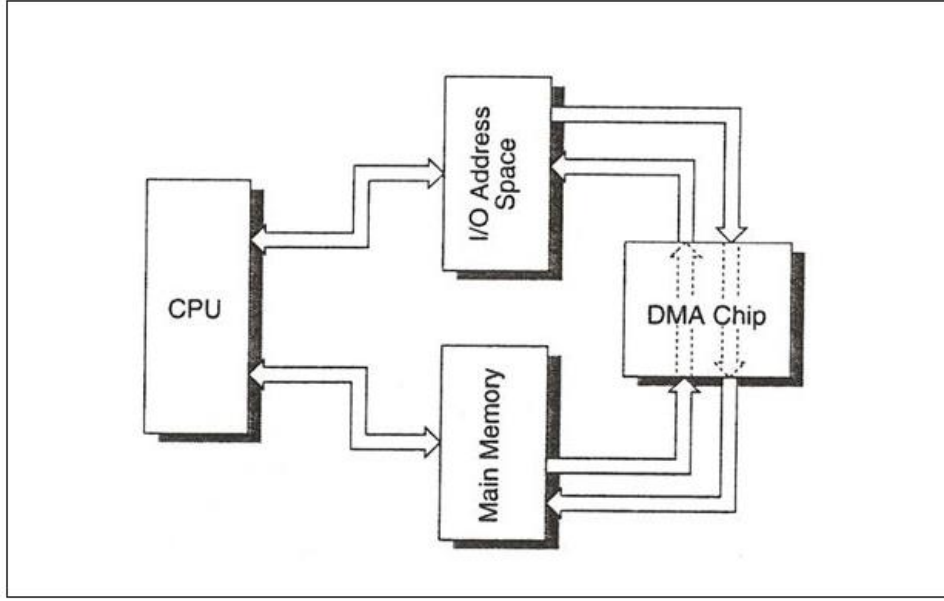
Mikroişlemcili sistemde bulunan birimler arasındaki ilişkiyi düzenleyen sinyallerin (yazma, okuma, interrupt, vb.) iletilmesi amacıyla kullanılan hatlar **kontrol yolu** olarak adlandırılır

GİRİŞ/ÇIKIŞ ARAYÜZ ELEMANLARI: Sistem ile dış dünya arasındaki etkileşimin artırılması sisteme bazı eklemelerin yapılması ile olabilir. Harici cihazlar (klavye, fare, yazıcı, çizici, disk, vs.) bu tip eklemelerdir. Bu tip cihazların sisteme doğrudan bağlanmasının doğurduğu sakıncaları ortadan kaldırmak için arayüz elemanları kullanılır. Aksi takdir harici cihazlar hatların kullanımında karışıklıklara neden olabilirler. PIA (paralel interface acces) paralel arayüz elemanı, SIO (serial input output) seri giriş çıkış elemanı, giriş/çıkış arayüz elemanlarına örnek olarak verilebilir.

KAYDEDİCİLER (Registers): Bir programın çalıştırılması esnasında, gerekli bilgilerin CPU tarafından kullanılabilmesini sağlayan geçici, dahili hafıza elemanlarıdır. Kısacası tüm bilgiler kaydedicilere gelir, CPU tarafından işlenir, ilgili yerlere buradan gönderilir.

SAYICILAR (Counters): Program komut (instruction) adresi ve yığın (stack) adresi gibi bilgileri saklayan geçici, dahili hafıza elemanlarıdır.

DMA (DIRECT MEMORY ACCES): Doğrudan hafıza erişimidir. DMA denetleyici çipi tarafından yapılır. Hızlı data transferi gerektiğinde kullanılır. CPU ile yaptığı bir dizi haberleşme sonucunda CPU'yu devre dışı bırakıp bus'ları teslim alır ve harici cihaza daha hızlı erişilmesini sağlar.



APU (ARITMETIC PROCESSING UNIT): Aritmetik işlemci birimi. CPU'lar sadece alınan emirlerin icrası ile ilgilenirler. Aritmetik işlemciler ise emirlerin icrası sırasında gerekli olan aritmetik işlemleri yapabilen co-processor'lerdir. CPU'ya yardımcı olduklarından hızı arttırırlar.

CO-PROCESSOR (YARDIMCI İŞLEMCI): Birincil işlemcinin (CPU) işlevlerini tamamlamak için kullanılan bir bilgisayar işlemcisidir. Kendi belleği vardır. Yardımcı işlemci tarafından gerçekleştirilen işlemler, kayan nokta aritmetiği, grafikler, sinyal işleme, dizi işleme, kriptografi veya çevresel aygıtlarla I / O arabirimi olabilir.



80486 işlemcilerden sonra matematik işlemci CPU içine yerleştirildi.

ALU (ARITMETIC LOGIC UNIT): CPU içinde bulunur ve emirlerle ilgili aritmetik (toplama, çarpma, kuvvet alma, vb.) ve lojik (AND, OR, büyüktür, vb.) işlemleri icra ederler. Sonuçlarını kaydedicilere gönderir.

KONTROL BİRİMİ: Kontrol emir kaydedicinin içeriği ile sürülür ve emir kaydedicisinin kodu bu birim tarafından çözülür. Yani komutların çözümlendiği birim.

Hafıza ve I/O işlemlerini tanımlamak ve zamanlama sağlamak için kontrol sinyalleri üreten birimdir.

Bellekten alınan emir kodları CPU' nun içinde bulunan Kontrol Birimi Ünitesinde daha küçük ve basit mikro emir kodlarına çözülür ve ne iş için kullanılacağı anlaşılır. Emir kodları bu şekilde basit mikro emir kodları yardımıyla icra edilir.

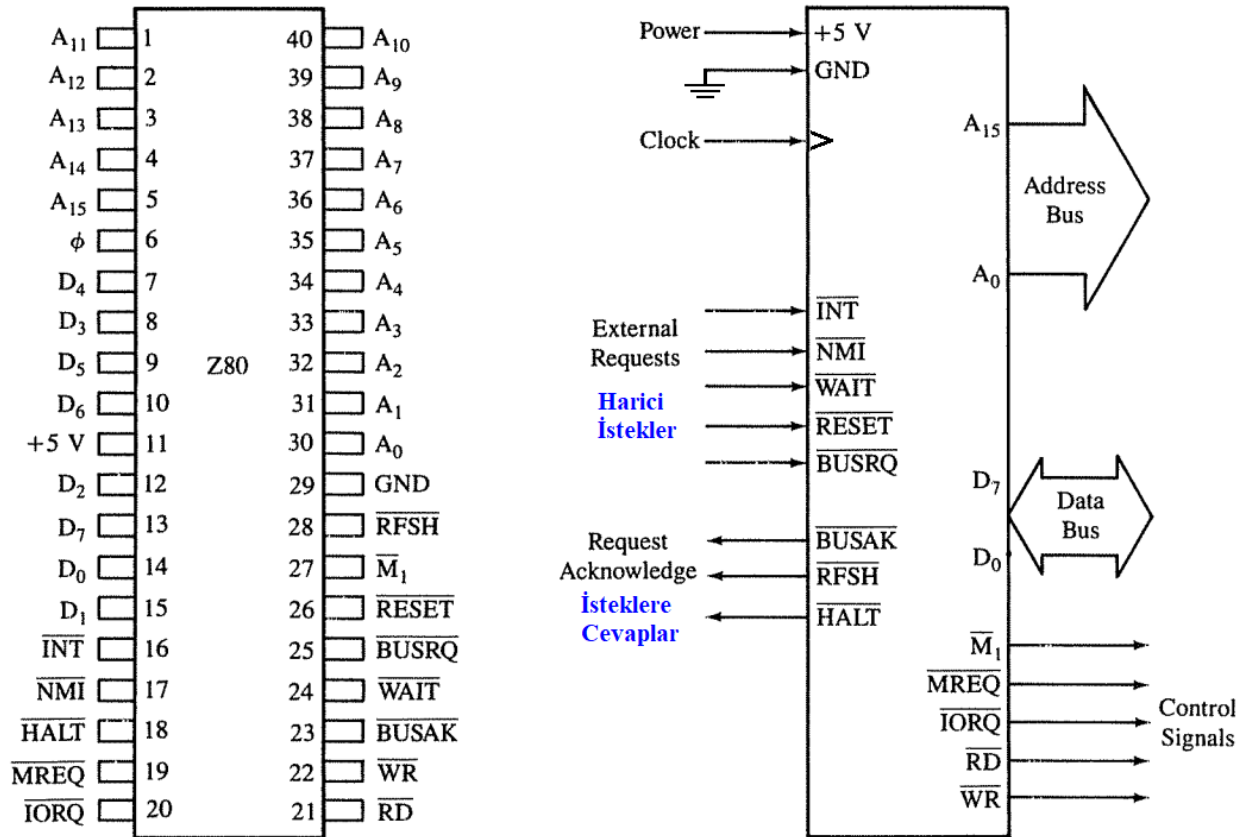
Program belleğe ilk erişimde her zaman ilk kodun emir olduğu varsayılır ayrıca her bir emrin icrası tamamlandıktan sonra, program bellekteki bir sonraki kod yine CPU tarafından emir kodu olarak algılanır.

Yani CPU her emrin icrasından sonra tekrar emir bekler.

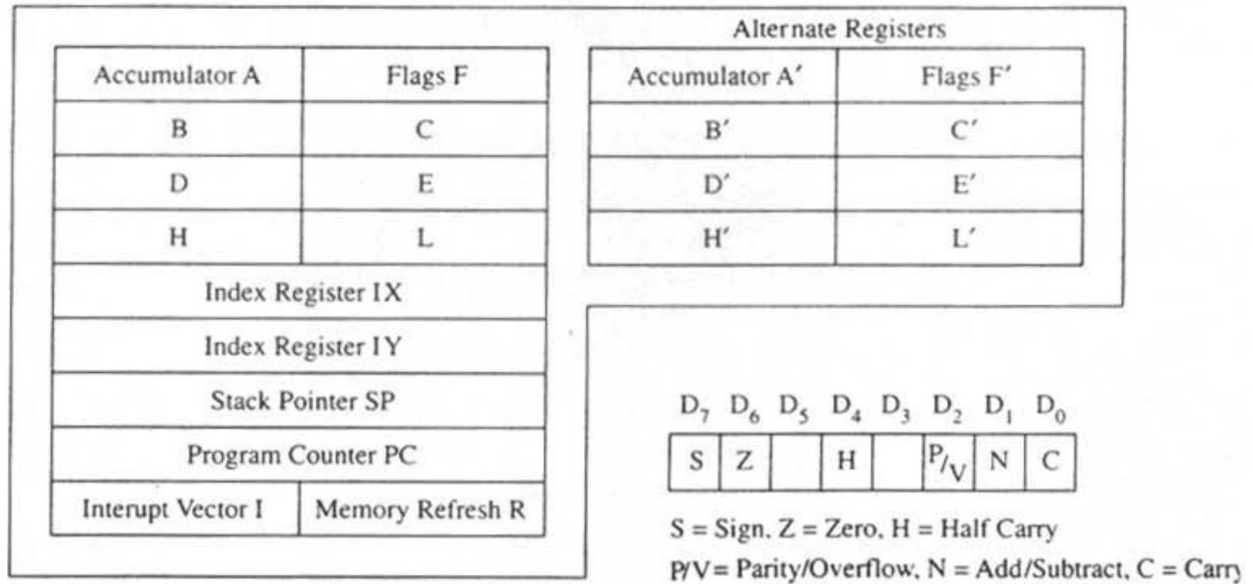
İntel mikroişlemcilerinin Gelişim Süreci

İşlemci Serisi	Veri Yolu	Adres Yolu	Hafıza Büyüklüğü
8048	8		2K internal
8051	8		8K internal
8085A	8	16	64K
8086	16	20	1M
8088	8	20	1M
80186	16	20	1M
80188	8	20	1M
80286	16	24	16M
80386EX	16	26	64M
80386DX	32	32	4G
80486DX/DX2	32	32	4G + 8K cache
80486SX	32	32	4G + 8K cache
80486DX4	32	32	4G + 16 cache
Pentium	64	32	4G + 16K cache
Pentium Pro	64	36	64G + 16K L1 cache + 256K L2 cache
Pentium II	64	36	64G + 32K L1 cache + 256K L2 cache
Pentium III	64	36	64G + 32K L1 cache + 256K L2 cache
Pentium 4	64	36	64G + 32K L1 cache + 512K L2 cache

Z80 Mikroişlemci



Z80 mikroişlemcinin pinleri ve lojik sinyal durumları



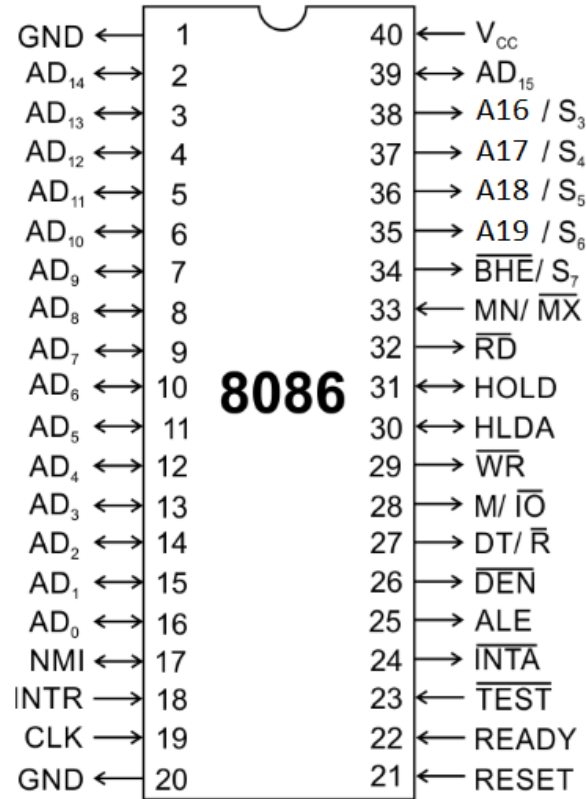
Z80'in 16 Bit Kaydediciler ve Bayrak bit durumları

8 bir (1 bayt) data hattı ve 16 bir adres hattı mevcuttur. $2^{16} = 64 \text{ kB}$ hafıza adresleyebilir.

8086 Mikroişlemci

Zilog tarafından geliştirilmiş olan Z80 (1974), INTEL tarafından geliştirilmiş olan 8080 (1974) 8-bit veri yollu CPU'lar bugünkü CPU'ların temelini oluşturmuşlardır. INTEL firması öne çıkarak, 16-bit veri yoluna geçiş noktasında 8088 ve 8086 (1977) olarak sunulmuş iki işlemci, 70'li yılların sonlarında yeni geliştirilmekte olan kişisel bilgisayarlarda kullanılmaya başlandı. Bu teknolojinin bilgisayar endüstrisinin devi IBM tarafından kabul görmesiyle birlikte **80x86** işlemciler kişisel bilgisayar endüstrisinde özel bir yere sahip oldular.

8086 işlemciler henüz entegre paketleme teknolojisinin en fazla 40-uçlu paketlere imkân sağlayabildiği dönemde geliştirildiğinden 16-bit veri genişliği için özel **çoklamalı uç** donanımı oluşturmak gerekti.



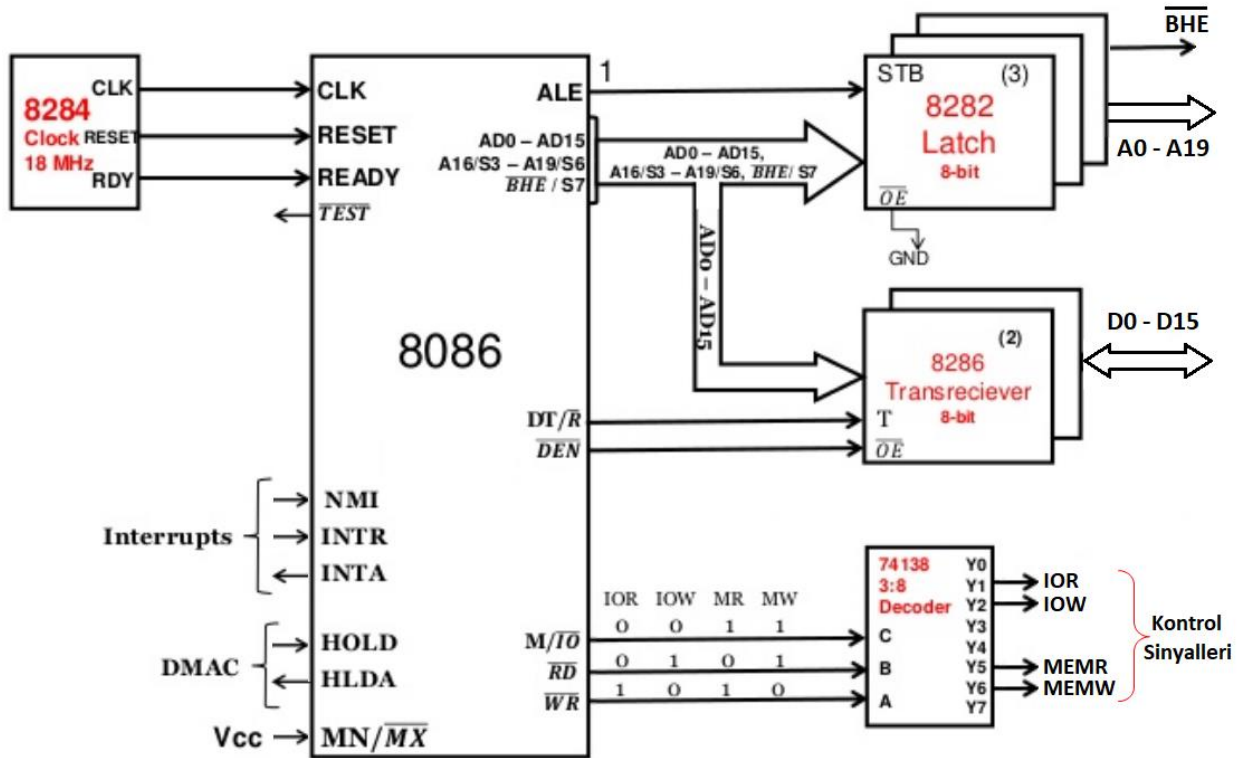
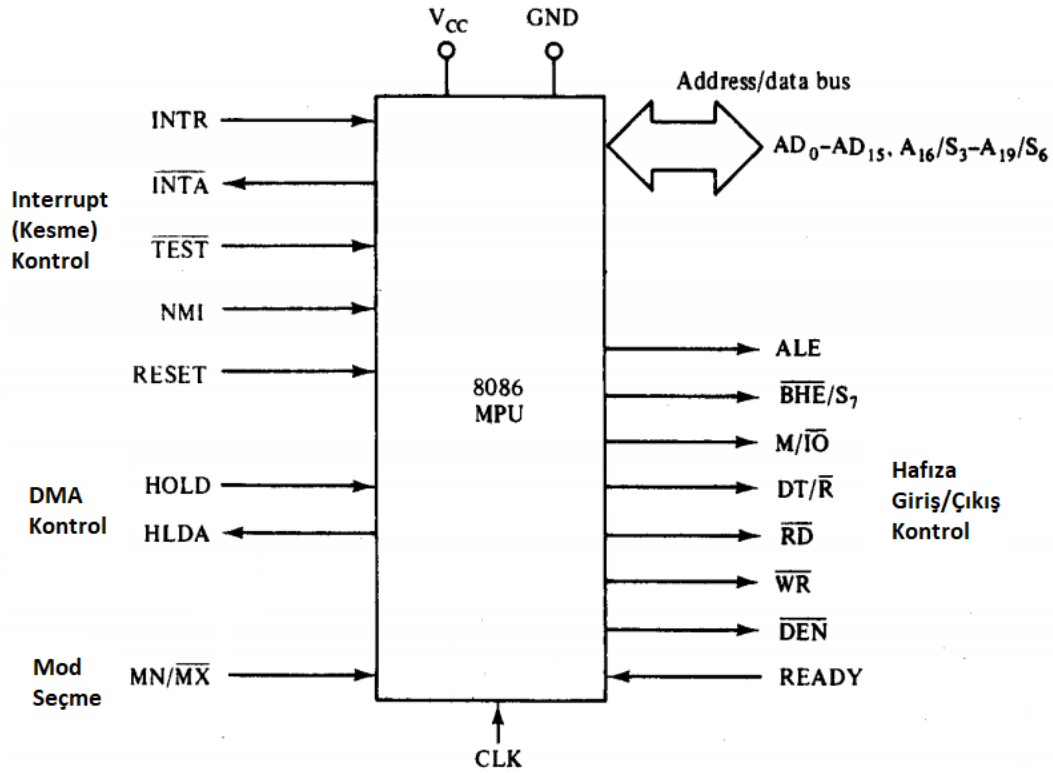
8086 CPU Pin Diyagramı

8086 İşlemcinin başta gelen giriş ve çıkış uçları şunlardır:

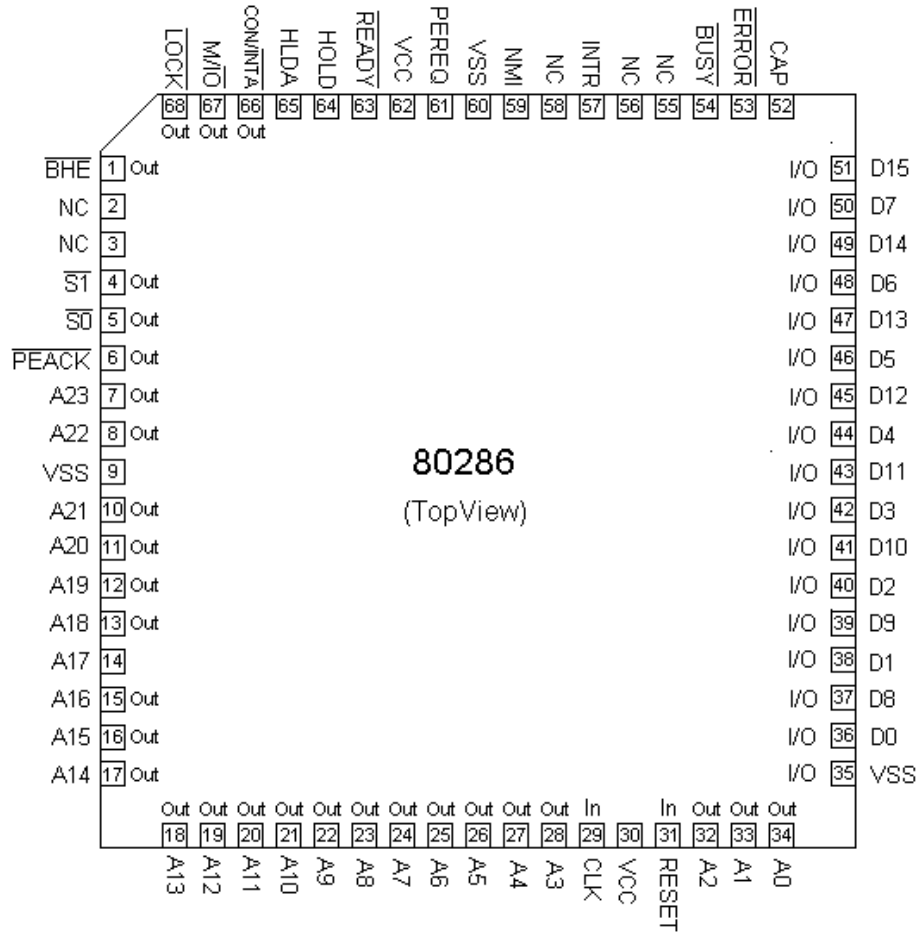
AD0-AD15 hem adres hem de veri iletmek amacıyla kullanılır. Veri iletirken giriş ya da çıkış olabilmesine karşın adres iletirken daima çıkıştır. A16-A19 çıkış uçları, AD0-AD15'e ilaveten 20 bit adres genişliğini oluşturur.

16 bir (2 bayt = Word) data hattı ve 20 bir adres hattı mevcuttur. $2^{20} = 1 \text{ MW}$ hafıza adresleyebilir.

Pinler:



8086 Çevre Bağlantı Konfigürasyonu



80286 Pin Konfigürasyonu (68 pin)

Pinlerin Görevleri:

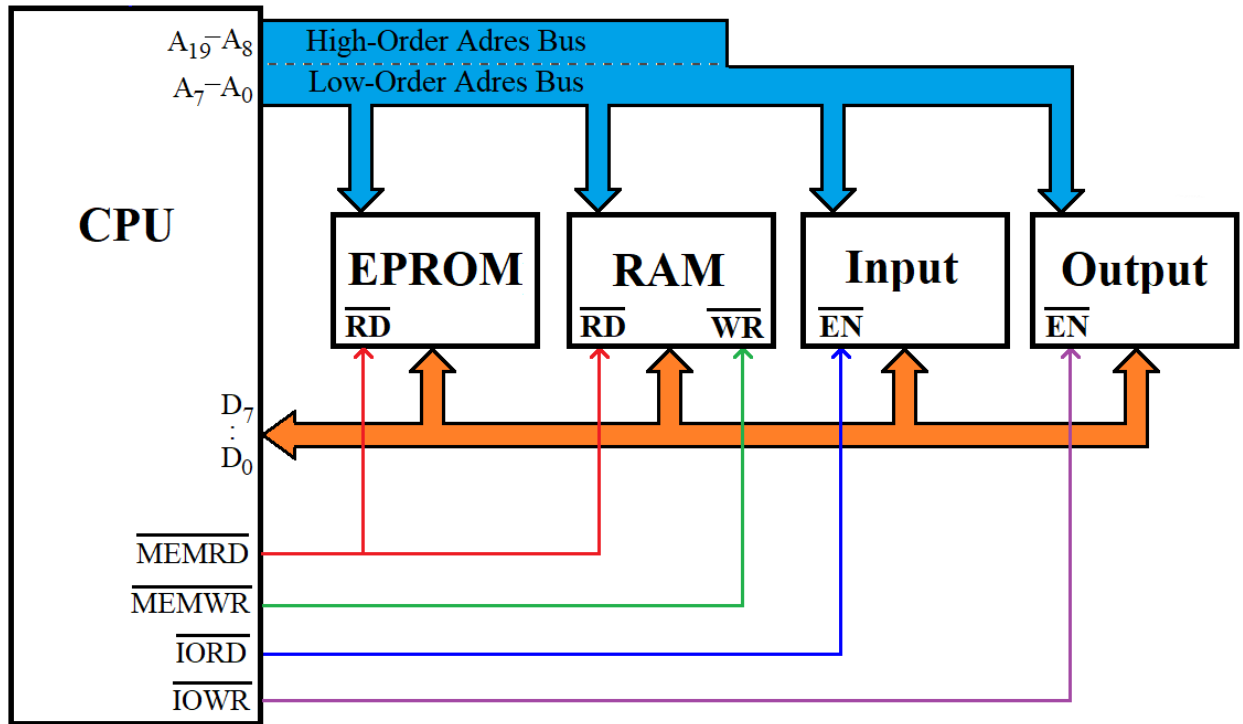
- **ALE** - Address Latch Enable (Adres Tutma Aktif) : Lojik 1 seviyesi ile işlemcinin Adres / Veri yolunda adres bilgisinin olduğunu belirtir.
- **$\overline{\text{BHE}}$ / S_7** - Bus High Enable (Yol Yüksek Aktif) : 8086 işlemcisinde, bir okuma veya yazma çevriminde, veri yolunun üst 8- bit'inin ($\text{D}_{15} - \text{D}_8$) aktif olduğunu dış lojiğe bildirir.
- **$\text{M} / \overline{\text{IO}}$** - Memory or Input / Output (Hafıza veya Giriş/ Çıkış) : Lojik 0 olduğunda mikroişlemcinin bir I/O cihazı ve lojik 1 olduğunda hafıza modülü üzerinde işlem yaptığını dış lojiğe bildirir.
- **$\text{DT} / \overline{\text{R}}$** - Data Transmit / Receive (Veri Gönderme/ Alma) : Lojik 1 olduğunda mikroişlemcinin dışarıya veri yolladığında ve lojik 0 olduğunda ise, mikroişlemcinin dışardan veri okuduğunu belirtir. Bu sinyal dış veri yolu buffer'larını aktif etmede kullanılır.
- **$\overline{\text{RD}}$** - Okuma (Read) : Lojik 0 olduğunda mikroişlemcinin hafızadan veya bir giriş cihazından veri okuduğunu dış lojiğe bildirir.
- **$\overline{\text{WR}}$** - Yaz (write) : Bu sinyal lojik 0 olduğunda mikroişlemcinin bir hafıza veya I/O cihazına bir veri yazdığını belirtir.
- **$\overline{\text{DEN}}$** - Data Bus Enable (Veri Yolu Aktif) : Harici veri yol buffer'larını aktif etmede kullanılır.
- **READY** - Hazır : Mikroişlemcinin bir giriş sinyali olup işlemcinin zamanlamasına bekleme durumları (wait states) ekleme için kullanılabilir. Eğer bu giriş, lojik 0 yapılırsa, mikroişlemci bekleme durumuna girer ve boşta kalır. Bekleme durumu istenmiyor ise normalde bu giriş lojik 1 durumunda kalmalıdır.
- **INTR** - Interrupt Request (Kesme isteği) : Donanım kesmesi için kullanılır. Kesme bayrağı (IF) 1 olduğunda INTR girişi lojik 1 yapılırsa, x86 işlemcisi bir dış donanım kesmesi almış olur ve bunun neticesi, o an yürütülen komut tamamlandıktan sonra, kesme tasdik (INTA) çevrimine girer.
- **$\overline{\text{INTA}}$** - Interrupt Acknowledge (kesme onayı) : INTR girişine işlemcinin cevap sinyalidir. Bu sinyal genelde bir kesme isteğine karşılık olarak, bir kesme vektör numarasının dışardan okunmasında, dış lojik tarafından kullanılır.
- **$\overline{\text{TEST}}$** - Test : WAIT komutu tarafından test edilen bir giriştir. Eğer TEST lojik 0 ise, WAIT komutu bir NOP gibi davranır. Eğer TEST lojik 1 ise, WAIT komutu TEST girişinin lojik 0 olmasını bekler. Bu uç 8087 nümerik işlemcisiyle beraber kullanılır.
- **NMI** - Nonmaskable Interrupt (Maskelenemeyen kesme) : INTR gibi, bir dış donanım kesme girişidir. Bu kesmenin oluşması için Kesme bayrağı (IF) bit'inin 1 olması gerekmez.
- **RESET** -Reset : Bu giriş, en az 4 saat periyodu lojik 1 seviyesinde tutulduğunda mikroişlemci kendisini sıfırlar ve işlemci **FFFF0h** adresindeki komuttan itibaren çalışmaya başlar. Ayrıca IF bayrağı da sıfırlanır.
- **$\text{MN} / \overline{\text{MX}}$** - Minumum / Maximum mode : İşlemcinin minumum veya maximum modlarından birinde çalışmasını seçme girişidir. Bu girişin 1 (+5V) olması durumunda, işlemci minumum modunda, toprağa (0V) bağlanması durumunda maximum modda çalışır.
- **HOLD** - Hold (tutma) : Doğrudan hafızaya erişim (Direct Memory Access - DMA) tek

girişidir. Eğer bu giriş lojik 1 yapılırsa, mikroişlemci çalışmasını durdurur ve adres, veri ve kontrol yolunu yüksek empedans durumuna koyar. Bu girişin lojik 0 olma durumunda işlemci normal çalışmasına devam eder.

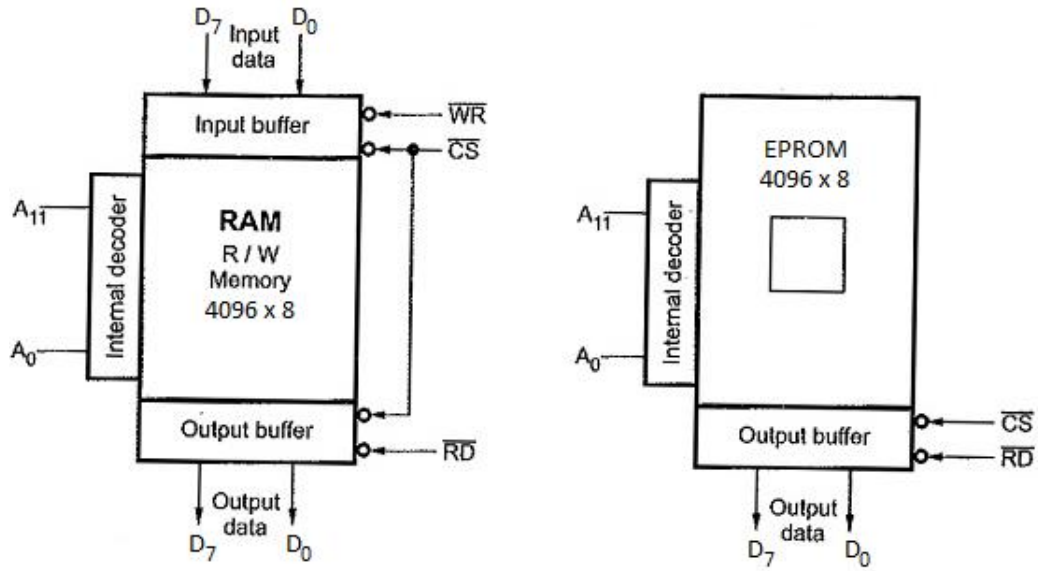
- **HLDA** - Hold Acknowledge (Tutma Tasdik) : İşlemcinin tutma durumuna girdiğini belirtir.
- **CLK** - Clock (saat) : Mikroişlemcinin temel zamanlama sinyalini sağlar.
- **VCC** - Power supply (Besleme kaynağı) : +5V besleme
- **GND** - Ground (Toprak) : İşlemci iki toprak girişine sahip olup her ikisinin de toprağa bağlanması gerekmektedir.

Haberleşme

Verilen bir adresteki elemanın (RAM, ROM, PIO, SIO) seçimi uygun donanımla yapılmaktadır. Adres bilgisi bir kod çözücünün girişine uygulanmakta ve kod çözücünün çıkışı adres bölgesinde kullanılacak olan RAM'ı veya ROM'u seçmektedir. Arayüz elemanının seçimi ise benzer biçimde olmaktadır. Genelde arayüz elemanındaki tüm registerlerin (yazmaç veya kaydedici) seçimi için adres hattının en düşük seviyeli hatlarından (Low-Order) yararlanılır. Adres hattının en yüksek seviyeli hatları (High-Order) ise arayüz elemanını aktif yapacak işareti üretmek için diğer uygun kontrol işaretleri ile birlikte, bir kod çözücünün girişine uygulanır. CPU tarafından üretilen okuma/yazma (R/W) işareti R/W bacağına sahip tüm cihazlara ortak bir hat üzerinden bağlanır. Data Bus ve Adres Buslar tüm elemanlar ve CPU tarafından ortak olarak kullanılmaktadır. CPU'nun data busına bağlanan tüm elemanların herhangi bir data çarpışmasına yol açmaması için Yüksek-Empedans (üç konumlu/three state) özelliğine sahip olmaları gerekmektedir.

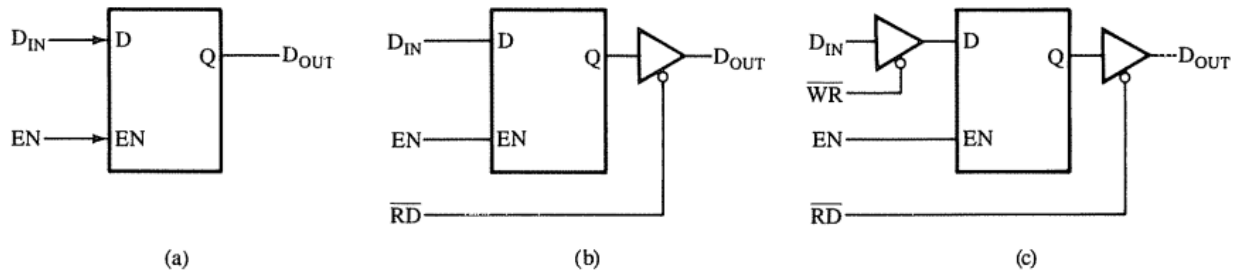


Adres ve Data hatları ile birlikte Kontrol sinyallerinin ilgili çiplere göre bağlantıları

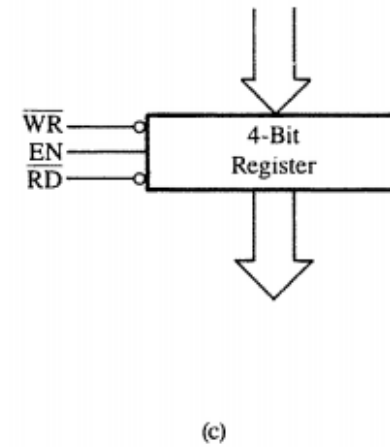
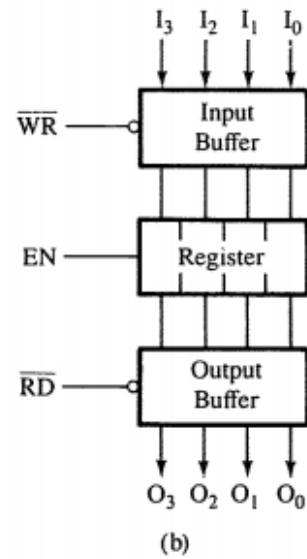
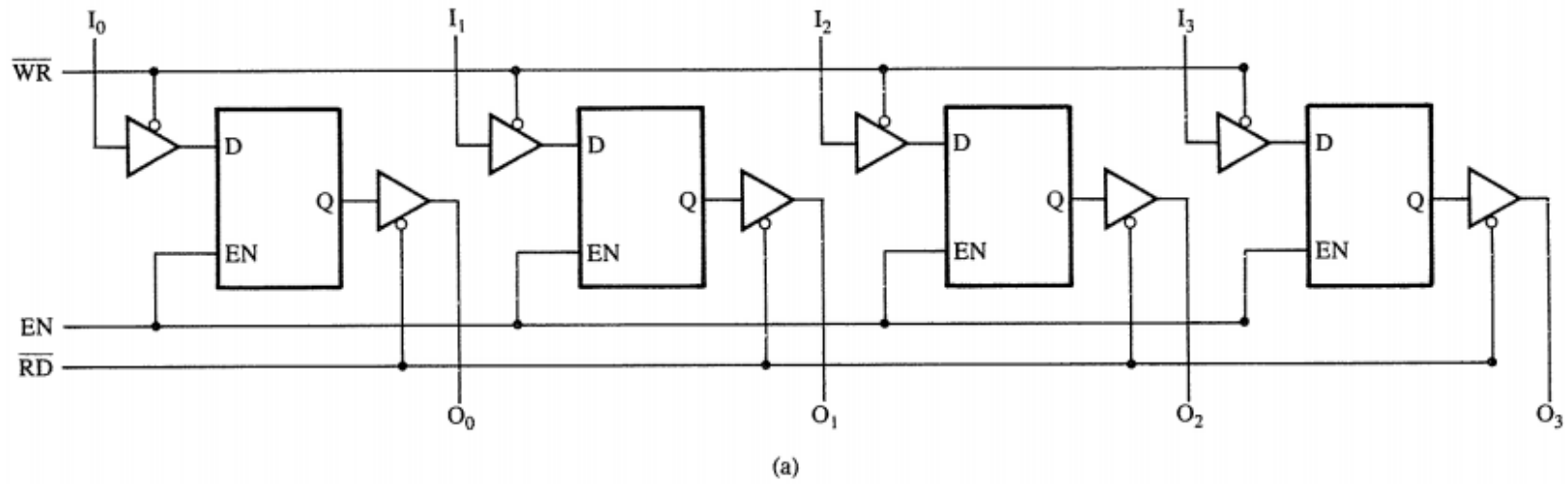


RAM ve EPROM'un fiziksel bağlantı şeması

Hafıza Elemanlarının İç Yapıları

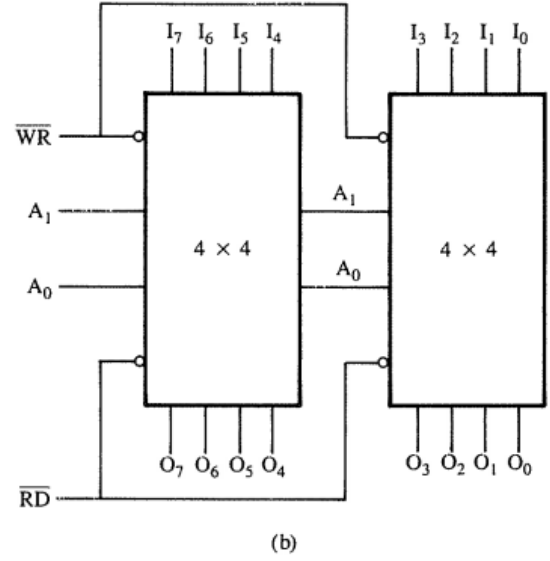
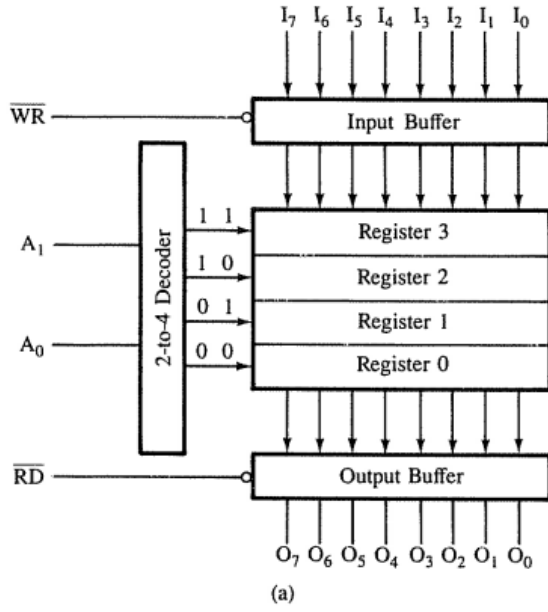


Depolama Elemanı olarak Latches (Tutucu)

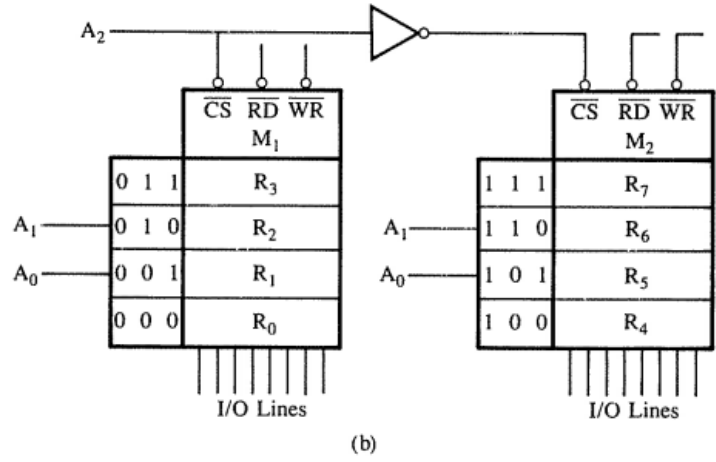
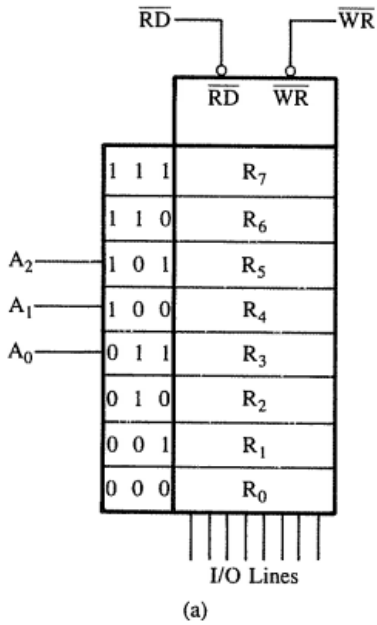


(a,b) 4 Bit Depolama Elemanı Olarak 4 Latche (Tutucu)

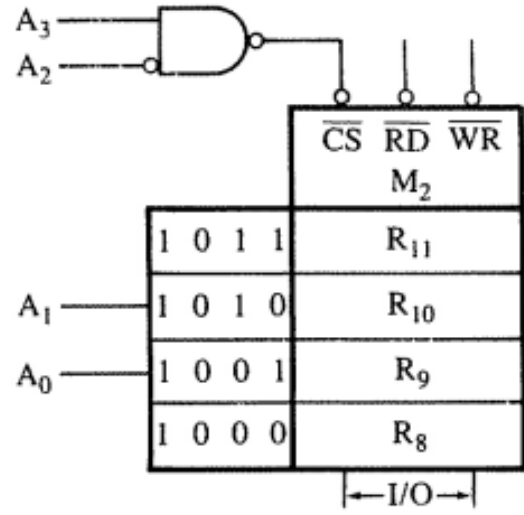
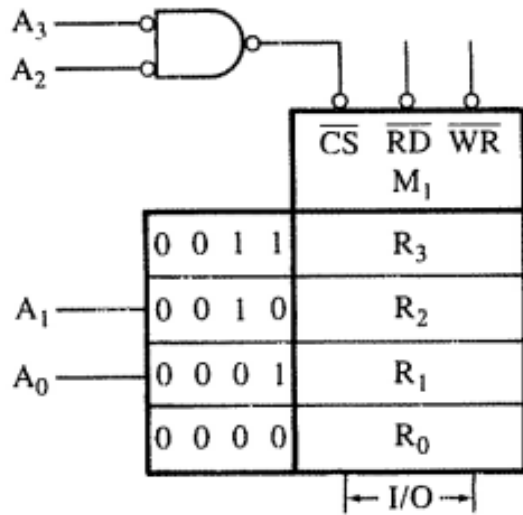
(c) 4 Bit Kaydedici Blok Diyagramı



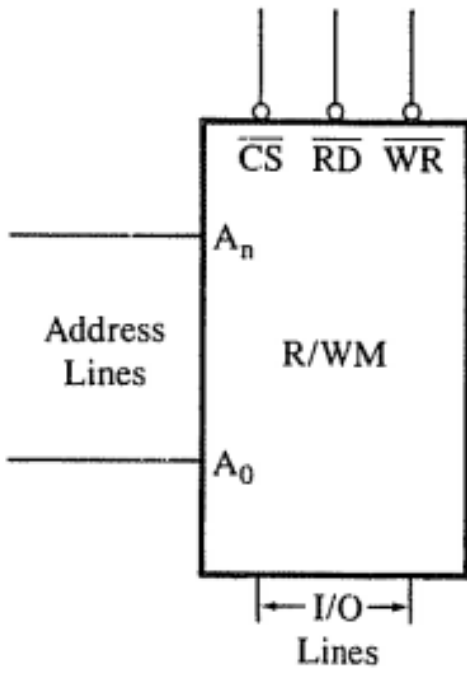
(a) 4x8 Bit Kaydedici (b) 2 adet 4x4 Bit Kaydedici ile 4x8 Bit Kaydedici



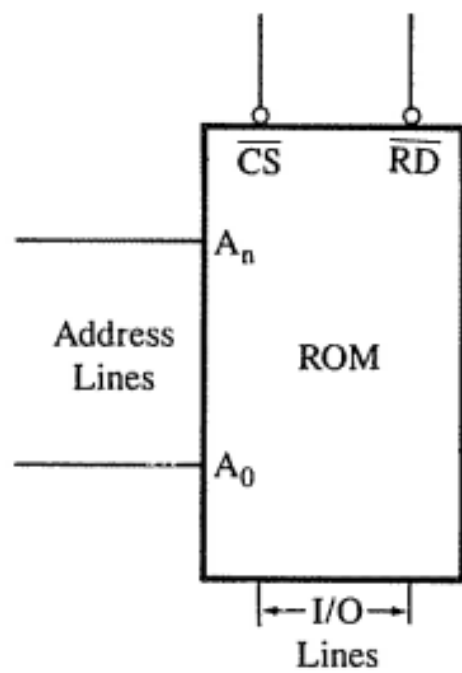
(a) 8 Kaydediciyle Hafıza Çipi (8x8 Bit) (b) 2 adet 4 Kaydedici ile Hafıza Çipi (8x8 Bit)



4 Adres hattı (A₀, A₁, A₂, A₃) ile 8 Kaydediciyi Adresleme



(a)

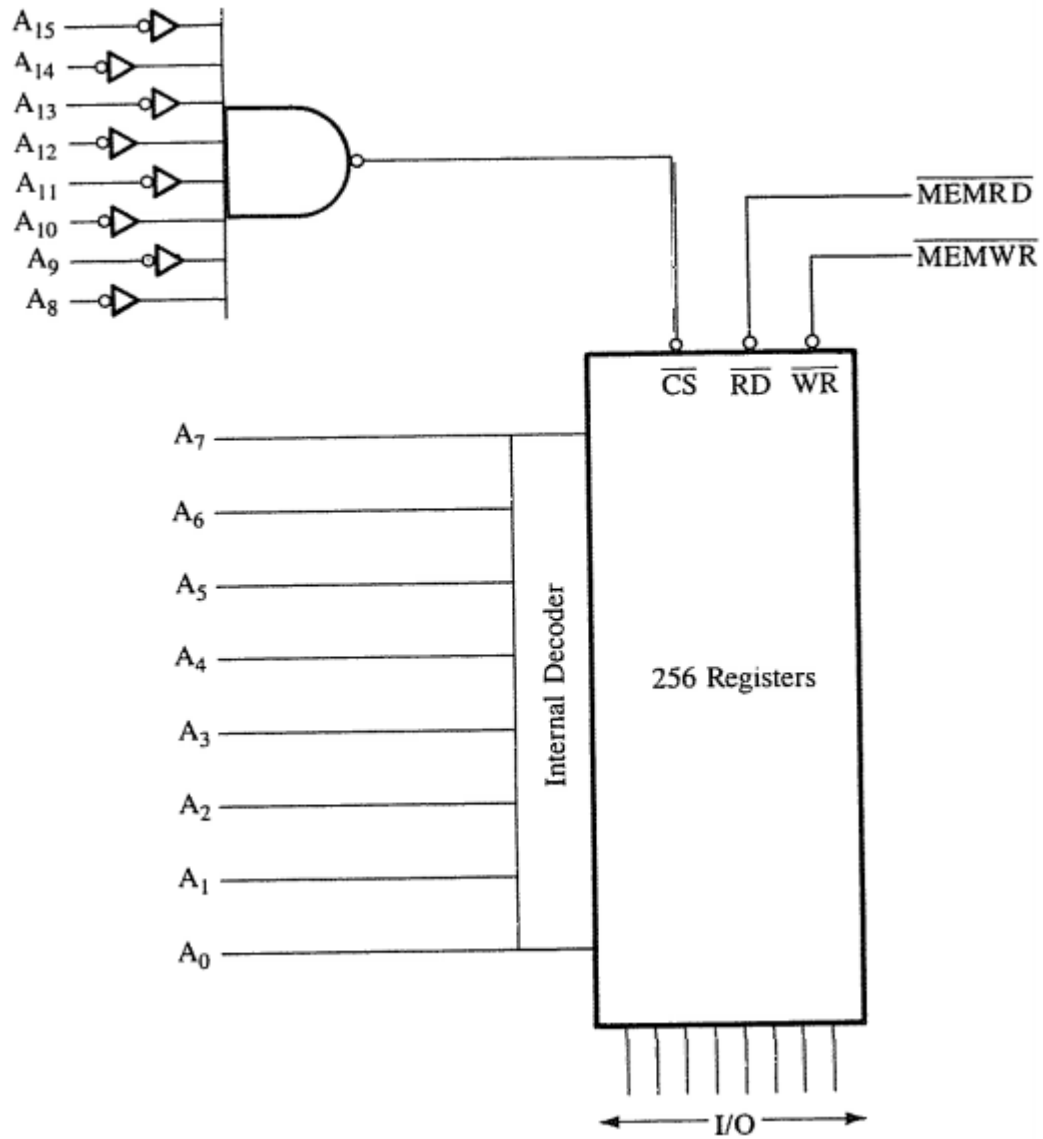


(b)

(a) R/W Hafıza Modeli (RAM)

(b) ROM Hafıza Modeli

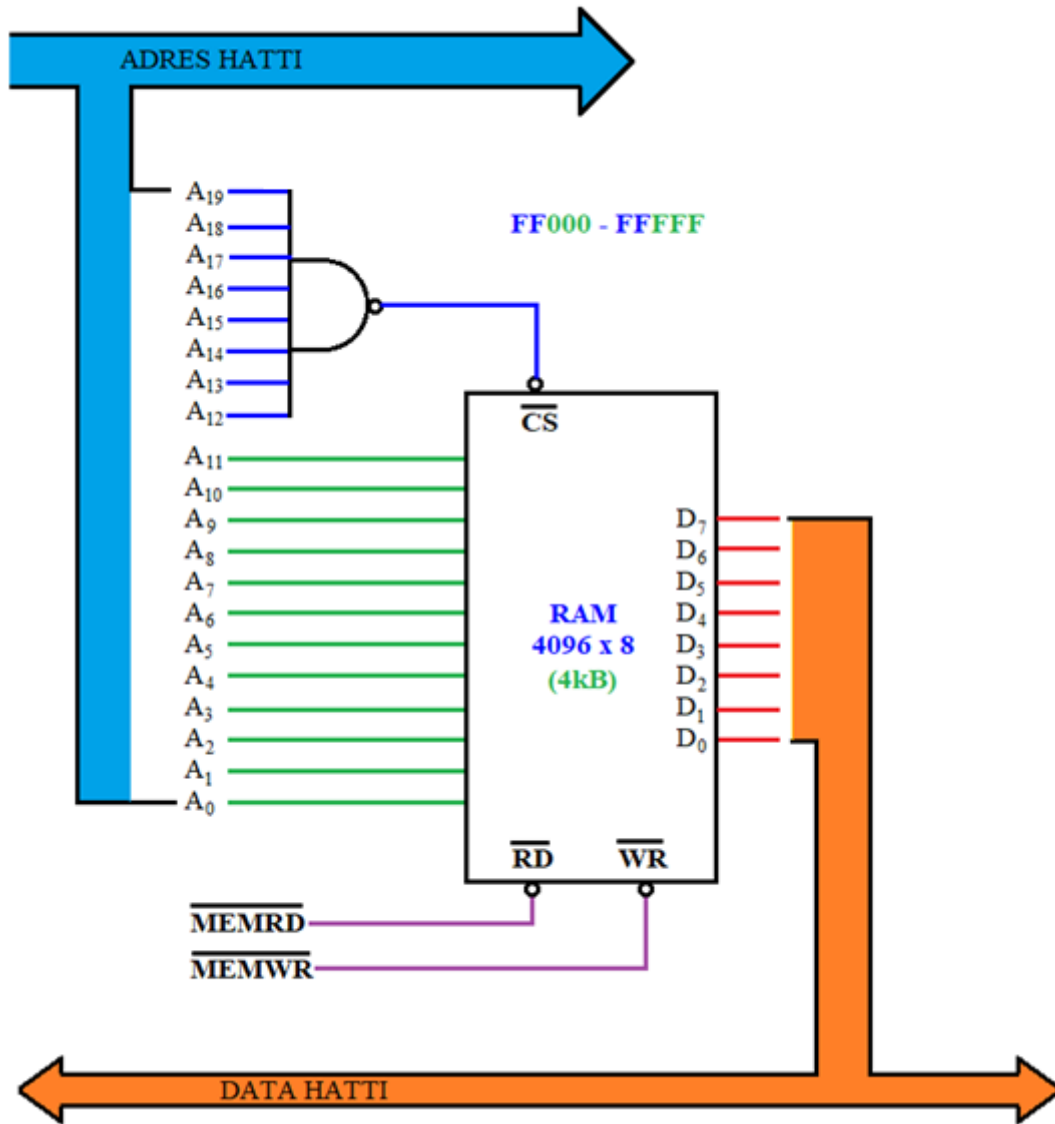
Hafıza Haritalama (Memory Map)



Hafıza Haritası : Hafızanın 256 Baytı. **0000 – 00FF**

A ₁₅	A ₁₄	A ₁₃	A ₁₂	A ₁₁	A ₁₀	A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	= 0000 _H	
								↓							↓		
								1	1	1	1	1	1	1	1	1	= 00FF _H
Chip Enable or Chip Select								Register Select									

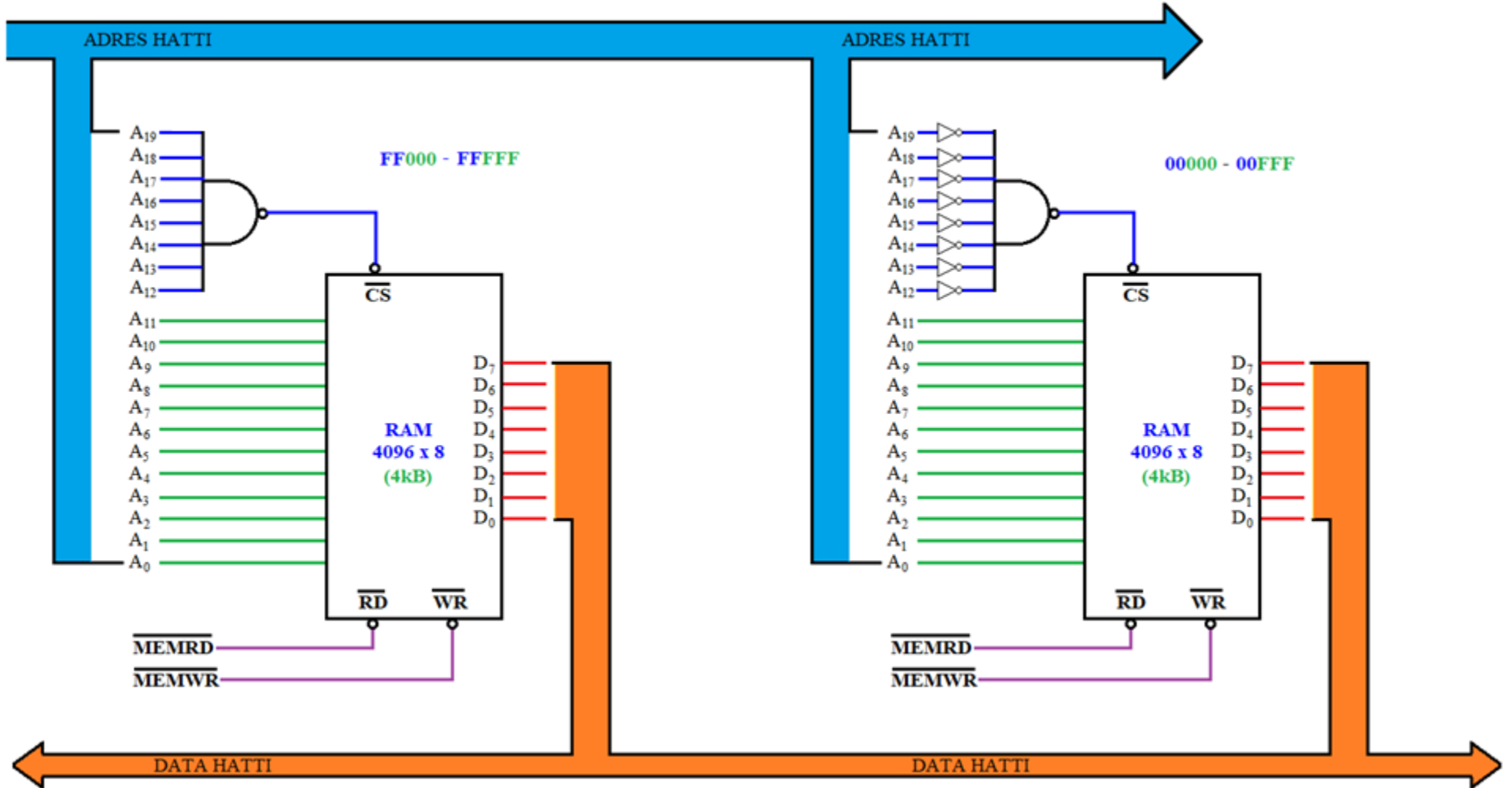
Örnek: Hafızanın FF000 adresinden başlayarak 4kB RAM hafızayı yerleştirerek kontrol sinyallerini oluşturunuz.



RAM belirli bir adrese bağlanması

Şekilde görüldüğü gibi bu kapasite arttırma işleminde RAM ve ROM belleğin databus'ı CPU'nun databus'ına bağlanır. CPU' nun yüksek seviyeli adres bitleri ise gerekirse bazı kontrol hatları ile birlikte bir kod çözücüye uygulanır ve kod çözücünün çıkışından üretilen işaretle istenen RAM veya ROM seçilir. Seçilen bellek elemanının ulaşılacak istenen hücresi de düşük seviyeli adres hatlarının bellek elemanının ilgili adres bacaklarına uygulanması ile sağlanır.

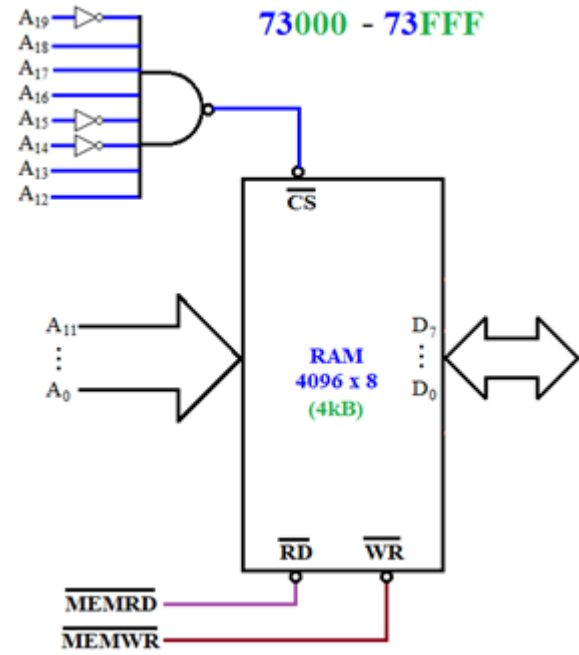
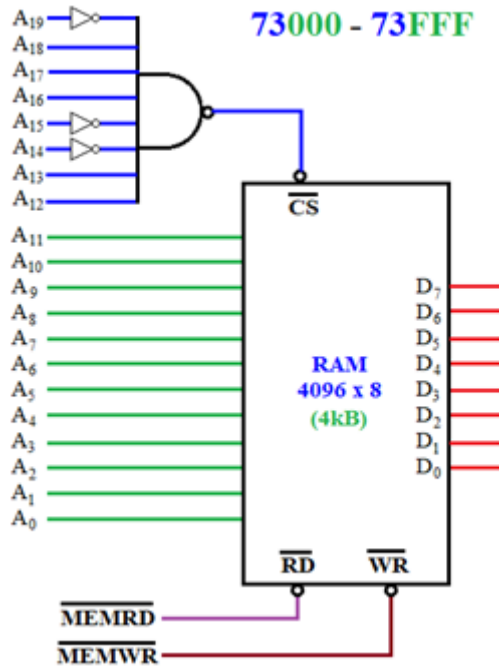
Örnek: Hafızanın 00000 ve FF000 adresinden başlayarak 1'er adet 4kB RAM hafızayı yerleştirerek kontrol sinyallerini oluşturunuz.



Örnek : 73000h ile 73FFFh adresleri arasında gelecek şekilde 4kB RAM bağlantısını yapınız.

RAM'ın adres hat sayısını $4k = 2^n$ formülü ile bulabiliriz. Burada $k = 1024 = 2^{10}$ dur. $2^n = 4096 = 4 \times 2^{10}$ olur. Buradan $n = 12$ bulunur. Yani A_0 ile A_{11} arası hatlar RAM'ın iç seçme hatlarıdır. 20 bit adres hattımız olduğu için geriye kalan 8 bit ($A_{12} - A_{19}$) adres hattımız ise RAM'ı aktif etmek için \overline{CS} ucuna bağlanmalıdır.

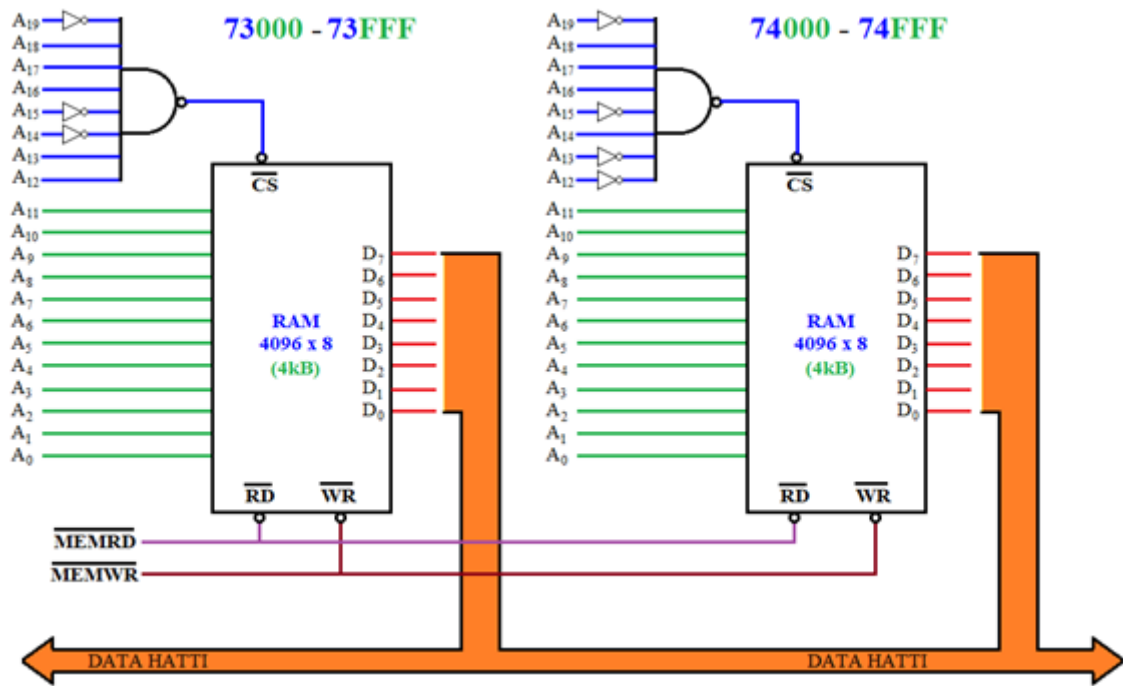
HEX Adr	A ₁₉	A ₁₈	A ₁₇	A ₁₆	A ₁₅	A ₁₄	A ₁₃	A ₁₂	A ₁₁	A ₁₀	A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀
73000	0	1	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
73FFF	0	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1



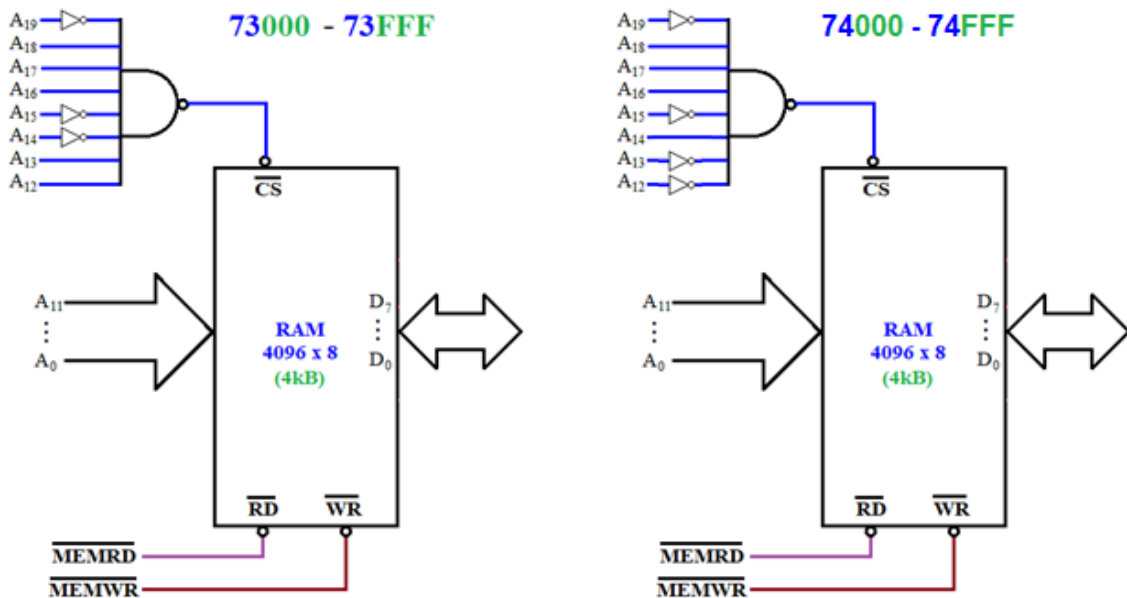
Örnek : 73000h ile 74FFFh adresleri arasında gelecek şekilde 2 adet 4kB RAM (toplam 8kB) bağlantısını yapınız.

RAM'ın adres hat sayısını $4k = 2^n$ formülü ile buluruz. Buradan $n = 12$ bulunur. Yani A_0 ile A_{11} arası hatlar RAM'ın iç seçme hatlarıdır. 20 bit adres hattımız olduğu için geriye kalan 8 bit (A_{12} - A_{19}) adres hattımız ise RAM'ları aktif etmek için \overline{CS} uçlarına bağlanmalıdır.

HEX Adr	A ₁₉	A ₁₈	A ₁₇	A ₁₆	A ₁₅	A ₁₄	A ₁₃	A ₁₂	A ₁₁	A ₁₀	A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀
73000	0	1	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
73FFF	0	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
74000	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
74FFF	0	1	1	1	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1



73000 ile 74FFF adresleri arasında 2 adet 4kB (8kB) RAM bağlantısı



Yukarıdaki örnekte RAM adreslerini farklı bir yoldan da bulabiliriz.

Çözüm : RAM'larımız; $4k = 2^n$ den $n = 12$ bit (hat) **000 – FFF** arası çalışır demek

$$\left. \begin{array}{l} 73000 + 000 = 73000 \\ 73000 + FFF = 73FFF \end{array} \right\} 73000 - 73FFF \text{ arası } \mathbf{1. RAM}$$

73FFF'den bir sonraki adres : $73FFF+1 = 74000h$ olur. Bununla FFF'yi toplarsak;

$$74000 - 74FFF \text{ arası } \mathbf{2. RAM}$$

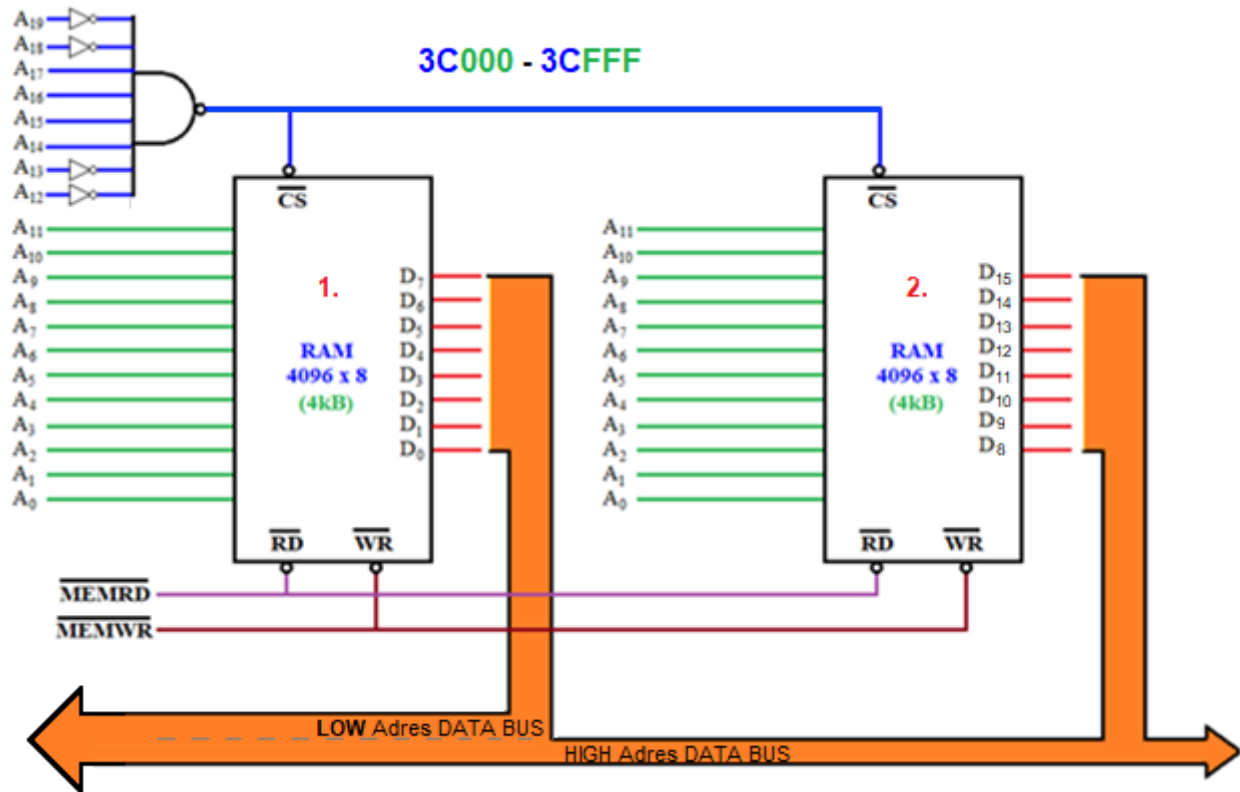
Dolayısıyla 2 tane 4kB RAM, **73000h – 74FFFh** adresleri arasına yerleşmiş olur.

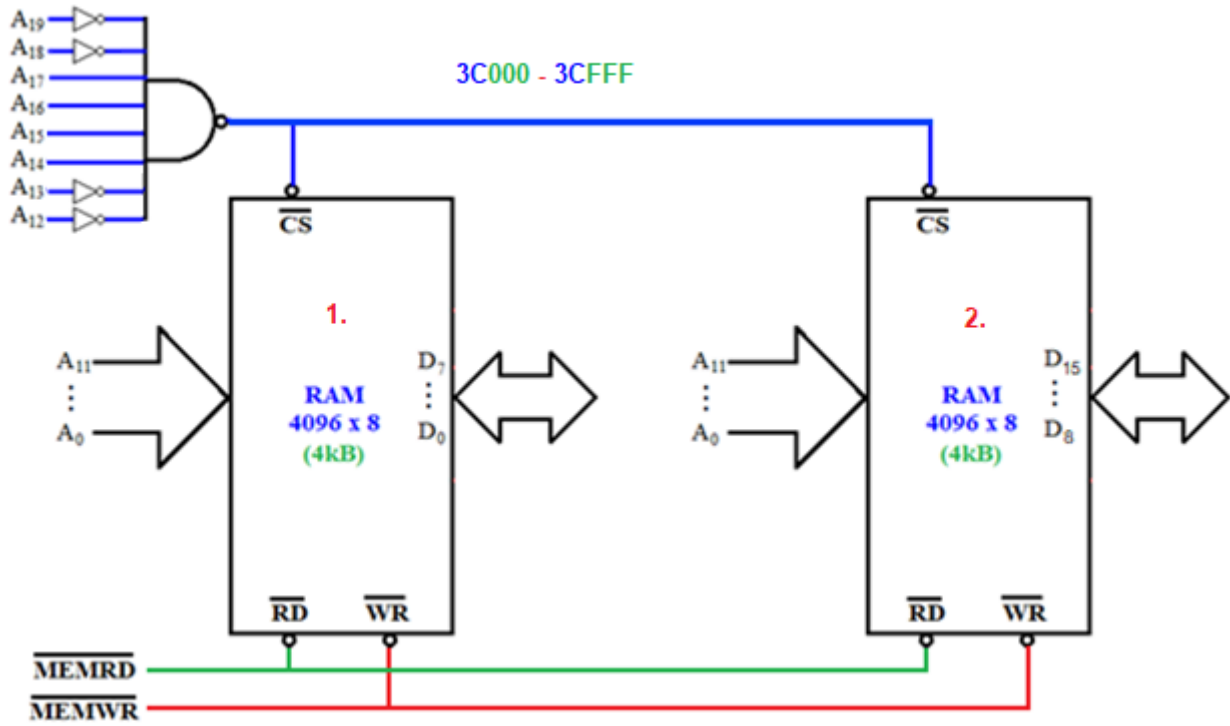
Örnek : 3C000h adresinden başlamak üzere 2 adet (8 bitlik) 4kB RAM kullanarak 1 adet (16 bitlik) 4kW (kilo Word) RAM bağlantısını yapınız.

Çözüm : RAM'larımız; $4k = 2^n$ den $n = 12$ bit (hat: A_0 ile A_{11}) **000 – FFF** arası çalışır demektir. Bu durumda RAM adresimiz; 3C000 – 3CFFF aralığındadır. RAM'larımız 8 bit (Bayt) ama bizden 16 bit (word) data hattı isteniyor. O yüzden RAM'lar paralel bağlanmalıdır. 1. RAM Low (düşük seviyeli) ($D_0 - D_7$) Data Bus'a, 2. RAM High (yüksek seviyeli) ($D_8 - D_{15}$) Data Bus'a bağlanmalıdır.

Üst seviyeli 3C (A_{12} ile A_{19}) değeri chip select (CS) için kullanılmalıdır.

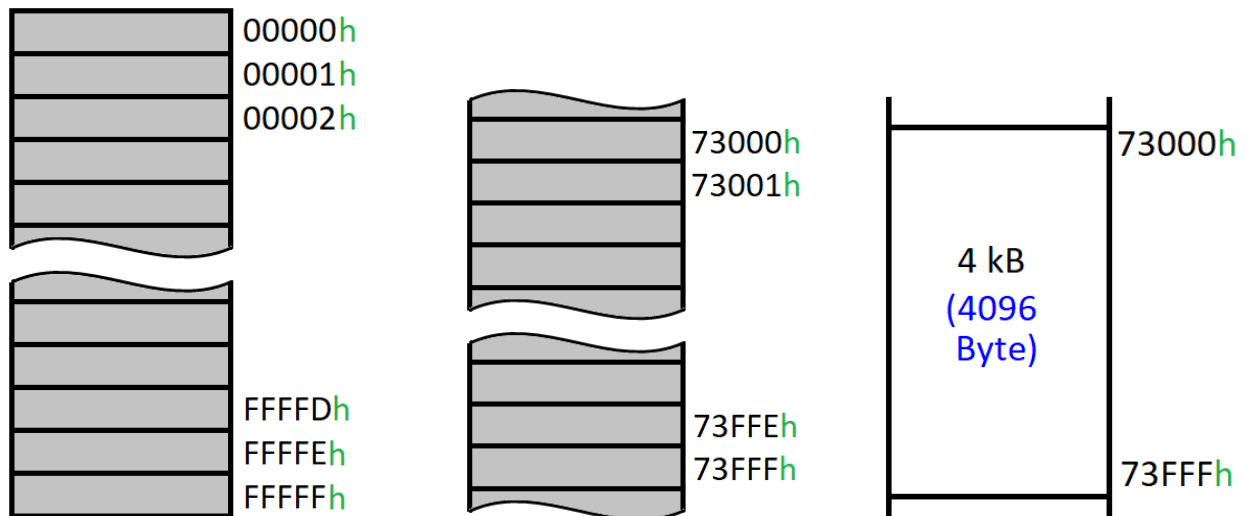
	A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}
3C	0	0	1	1	1	1	0	0





Ödev : Hafızanın 40000h adresinden başlayarak 2 tane 8kB RAM (toplam 16kB) yerleştiriniz. Adres bağlantılarını gösteriniz.

Hafızanın şematik gösterimi

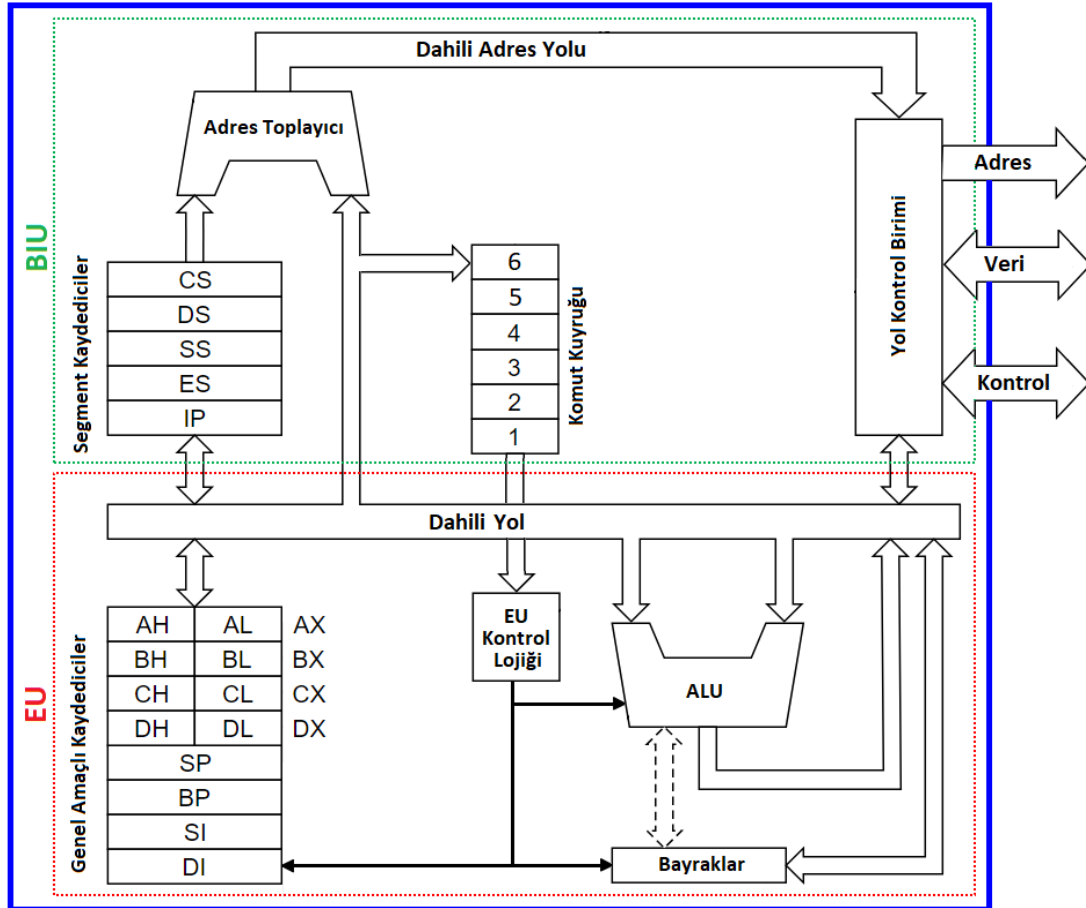


İç Mimari:

Aşağıda x86 ailesinin 16-bit çekirdek mimarisinin basit bir gösterimi verilmiştir. Buna göre mikroişlemci iki temel ayrı çalışma birimine sahiptir; Yürütme birimi (**Execution Unit-EU**) ve Yol arabirimi (**Bus Interface Unit-BIU**).

EU: komutları yorumlamakta ve yürütmektedir.

BIU: ise yol işlemlerini (işlem kodu okuma, operand okuma ve giriş/çıkış cihazlarıyla haberleşme gibi) yerine getirir.



8086 Mikroişlemci İç Mimarisi

Yürütme Birimi:

EU, komut çözme ve komutlar yürütme için bir kontrol birimine, ALU' ye, genel amaçlı saklayıcılara (AX,BX,CX,DX), işaretçi (BP, SP) ve indis (DI, SI) saklayıcılar ile bayrak (Flags) saklayıcısına sahiptir. Kontrol birimi, BIU tarafından komut kuyruğuna (Instruction queue) sırasıyla yerleştirilen makine dilindeki komutların yorumlanmasını, kodunun çözülmesi ve yürütülmesi için gereken ve işlemleri kontrol eder. EU hafızadan bir operanda ihtiyaç duyarsa veya yazmak isterse bu işi BIU' ya yönlendirir. Bu sırada BIU için gereken fiziksel hafıza adresi hesaplama işlemlerini de sağlar.

Yol Arabirimi:

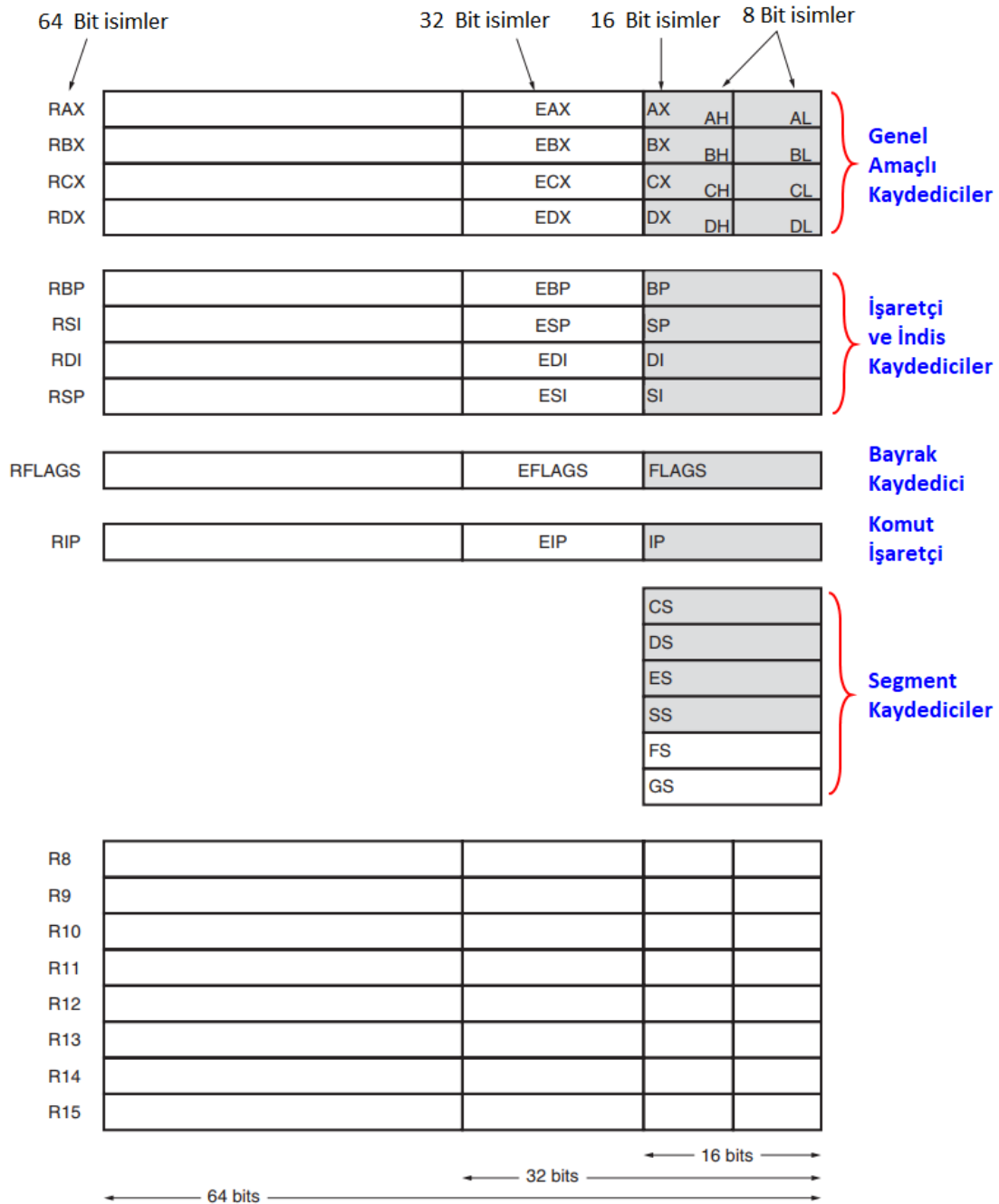
BIU; bütün dış yol işlemlerini kontrol eden bir kontrol birimine, EU' nun yürüteceği komut byte' larını tutan komut kuyruğuna, fiziksel hafıza adresleri üretmek için bir toplayıcıya, 4 segment saklayıcısına (CS, DS, SS, ES), komut işaretçisine (IP) sahiptir. Anlaşılacağı gibi **BIU hafıza ve Giriş/çıkış işlemleri dahil bütün dış yol işlemlerinden sorumludur**. BIU komut kuyruğuna yerleştirmek üzere örneğin 8086 mikroişlemci için 6 byte komut kodunu önden okuyabilir. Komut kodunun bu şekilde önden okunması **BIU ve EU birimlerinin paralel çalışmasına imkân verir**.

Önceden okunan komutlar icra edilirken yenilerinin bu esnada okunabilmesi işlemcinin performansını artırır. Bu tarz çalışan mikroişlemci mimarisine **işhatlı mimari** (*pipelined architecture*) adı verilir.

Kaydediciler (Registers):

Aşağıda Intel x86 ailesinde yer alan mikroişlemcilerin temel kaydedicileri göstermektedir. Gri olarak boyanmış alanlar 16 bitlik (8086, 80186 ve 80286) işlemcilerde bulunur, **E** ile başlayan 32-bit kaydediciler, 80386, 80486 ve Pentium gibi işlemcilerde bulunur. **R** ile başlayan 64-bit kaydediciler, Core Duo, i3, i5, i7 gibi işlemcilerde bulunur.

Bir CPU'da kaydediciler veri uzunluğu kadar olmak zorundadır.



x86 Kaydedicileri (Registers)

Genel Amaçlı Kaydediciler:

Bu grupta yer alan kaydediciler, programcı tarafından çok değişik amaçlarla (matematik işlemler, mantıksal işlemler, veri saklama, vb.) kullanılabilir. Daha sonraki bölümlerde açıklanacak bazı komutlar, bu kaydedicileri özel işlemlerde kullandıkları için, her birine bu işlemleri belirten bir isim verilmiştir;

AX: Akümülatör (accumulator),

BX: Taban (base),

CX: Sayaç (counter)

DX: Veri (data)

olarak isimlendirilir. Genel amaçlı kaydedicilerden her biri, **8-bit** (AH, AL, BH, BL, CH, CL), **16-bit** (AX, BX, CX ve DX), **32-bit** (EAX, EBX, ECX ve EDX) veya **64-bit** (RAX, RBX, RCX ve RDX) olarak kullanılabilir. Kaydediciler, assembly dilde, iki veya üç harfli gösterimlerle belirtilir. Örneğin, akümülatör; RAX, EAX, AX, AH veya AL ile ifade edilir. Genel-amaçlı kaydedicilerin temel fonksiyonları aşağıda sırasıyla sunulacaktır. Daha sonraki bölümlerde ise kullanımları örneklerle anlatılacaktır.

İşaretçi ve İndis Kaydedicileri

İşaretçi ve indis kaydediciler (BP, SP, DI, SI), genel amaçlı olarak da kullanılabilir. Genellikle, hafızada yer alan operand'lara erişimde indis veya işaretçi olarak kullanılır. Yani ikincil adres tutucu olarak kullanılırlar.

BP: Taban İşaretçisi (base pointer),

SP: Yığın İşaretçisi (stack pointer),

DI: Hedef İndis (destination index),

SI: Kaynak İndis (source index),

Segment Kaydedicileri

Mikroişlemci'deki diğer kaydedicilerle (BX, BP, SP, DI, SI) beraber hafıza **adresleri üretmede** kullanılır.

CS: Kod Segment (code segment). Kod segment, hafızanın, programları ve alt programları tutan bir bölümdür. Program kodunun başlangıç taban adresini saklar. Okunacak komutun offset adresi **IP** kaydedici tarafından belirlenir.

DS: Veri Segment (data segment). Bir program tarafından kullanılan verilerin bulunduğu hafıza alanıdır. Bu veri alanının başlangıç taban adresini saklar. Bu alana yazılacak veya okunacak verinin offset adresi, **BX**, **DI** ve **SI** kaydedicileri tarafından belirlenir.

SS: Yığın Segment (stack segment). Yığın için kullanılan hafıza alanının başlangıç taban adresini saklar. Yığın segment'ine yazılacak veya okunacak verinin offset adresi, **SP** ve **BP** kaydedicileri tarafından belirlenir.

ES: Ek Segment (extra segment). Bazı string komutlarında kullanılan ek veri alanının başlangıç taban adresini saklar. Bu alana yazılacak veya okunacak verinin offset adresi, **DI** ve **SI** kaydedicileri tarafından belirlenir.

IP (Instruction Pointer-Komut İşaretleyicisi)

Her zaman mikroişlemci tarafından yürütülecek bir sonraki komutu adreslemede kullanılır ve kod segmentin offset adresini tutar. IP içeriği, Kod Segment içeriği ile toplanmasıyla, bir sonraki komutun gerçekte olan fiziksel hafıza adresi bulunur.

FLAGS (Bayrak Kaydedici)

Bayraklar, mikroişlemcinin çalışmasını esnasında olan olay ile ilgili uyarı bitleridir. Bayrak bit'leri, birçok aritmetik ve lojik komut yürütüldükten sonra değişmektedir. Bayrakların bazıları, mikroişlemcinin çalışmasını kontrol etmek amacıyla da kullanılır. Aşağıda bayraklar ve fonksiyonları kısaca anlatılmaktadır.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				O	D	I	T	S	Z		A		P		C

- **C (Carry)** : Bir işaretli sayının aritmetik işleminde, toplamadan sonraki eldeyi (carry) veya çıkarmadan sonraki ödöncü (borrow) belirtir.
- **P (Parity)** : Tek eşlik (odd parity) işlemi, lojik 0; çift eşlik (even parity) işlemi, lojik 1 ile gösterilir. Eşlik, tek veya çift olarak belirtilen, bir byte veya word'deki 1 'lerin sayısıdır. Örneğin, ikili olarak belirtilen bir sayı (00110100) 3 tane 1 'e sahipse, o sayı Tek Eşlik'tir. Eğer bir sayı 1'lere sahip değilse (bütün bit'ler "0"), o sayı Çift Eşlik olarak kabul edilir.
- **A (Auxiliary carry)** : Yapılan bir aritmetik işlem sonucunda, 4. bitten 5. bite, toplamadan sonraki eldeyi ve çıkarmadan sonraki ödöncü belirtir. Bu özel bayrak bit'i **BCD bir toplama veya çıkarma işleminden sonra** , AL'nin değerini ayarlamak için, DAA ve DAS komutları tarafından test edilir.
- **Z (Zero)** : Bir aritmetik ve lojik işlem sonucunun sıfır olduğunu belirtir. Eğer Z=1 ise, sonuç sıfırdır ve eğer Z=0 ise sonuç sıfır değildir.
- **S (Sign)** : Bir toplama veya çıkarma işleminden sonra, sonucun aritmetik işaretini belirtir. Eğer sonuç negatif ise S=1, sonuç pozitif ise S=0 olur.
- **T (Trap)** : Eğer Trap bayrağı 1'lenmiş ise, tümdevre hata takip (debugging) işlemi devreye girer. Yani adım adım çalıştırmak için kullanılır
- **I (Interrupt/Kesme)** : Mikroişlemci tümdevresinin kesme isteği giriş bacağı INTR, harici kesme isteği (Interrupt Request) işlemini kontrol eder. Eğer I=1 ise, INTR bacağı aktif yapılır (enabled) ve eğer I=0 ise, INTR bacağı pasif (disabled) olur. Aktif olma durumunda, mikroişlemci tümdevre INTR ucu üzerinden gelen kesmelere cevap verir. Aksi durumda, gelen kesme istekleri mikroişlemci tarafından ihmal edilir, yani cevap verilmez. 'I' bayrak bit'inin durumu, STI (Set I Flag) ve CLI (Clear I Flag) komutlarıyla kontrol edilir.
- **D (Direction/Yön)** : String komutları yürütülürken, DI ve/veya SI kaydedicilerinin artırılması veya azaltılması işlemlerinin seçimini kontrol eder. Eğer D=1 ise, kaydediciler otomatik olarak azaltılır ve eğer D=0 ise, kaydediciler otomatik olarak artırılır. D bayrağı STD (Set Direction) komutuyla 1 'lenir ve CLD (Clear Direction) komutuyla 0'lanır.
- **O (Overflow)** : Taşma (overflow), işaretli sayıların toplandığında veya çıkartıldıklarında oluşan bir durumdur. Taşma, işlem sonucunun hedef saklatıcıya sığmadığını gösterir. örneğin, 8-bit saklayıcılarda işaretli sayı aritmetiğinde, 7Fh (+127) sayısı 01h ile toplandığında sonuç 80H (-128) olur. Bu sonuç işaretli toplama için O- bayrağı ile belirtilen bir taşma durumudur.

Fiziksel ve Offset Adres

8086 mikroişlemci 20 bit adres hattına sahiptir. Ama kaydedicilerimiz 16 bitlik yapıdadır. Bu durumda 20 bit adresleme imkânı direk olarak yoktur. Bu nedenle iki tane 16 bitlik kaydedici kullanılarak 20 bit gerçek adres elde edilebilmektedir.

Segment kaydediciler (CS, DS, SS, ES), 16 bit olduğu için, hafızayı $2^{16} = 64$ kB'lık bölümlere (Segment) ayırır. İşaretçi ve indis kaydediciler ise bu Segment içerisindeki alanları (offset) seçmek için kullanılır. 1 MB'lık hafızada toplam 16 adet Segment bulunur. Her bir Segment **X0000-XFFFF** aralığında olur. burada X ise 0,1,2, ... , E, F olabilir.

Gerçek Adres	Segment Adresi
FFFF	15. Segment
F0000 EFFF	14. Segment
E0000 2FFF	2. Segment
20000 1FFF	1. Segment
10000 0FFF	0. Segment
00000	

20 bitlik gerçek adres ise;

Segment kaydedicideki adres **10h ile çarpılır** (yani sağına bir “0” eklenir) ve ofset kaydedicideki adres ile toplanır. Bu işlemi BIU içerisindeki “Adres Toplayıcı” bölümü yapmaktadır.

Örnek : CS = 1000h ve IP = 3A07h olsun. Gerçek adres;
 $1000 \text{ h} \times 10 \text{ h} = 10000 \text{ h}$
 $10000 \text{ h} + 3A07 \text{ h} = \mathbf{13A07h}$ olarak bulunur

CS:IP şeklinde gösterilir, **1000:3A07** ve **13A07h** anlamına gelir

AFFFFh	Yığın segmenti	
A0000h		SS=A000
8FFFFh	Ekstra segment	
80000h		ES=8000
3FFFFh	Veri segmenti	
30000h		DS=3000
1FFFFh	Kod segmenti	
10000h		CS=1000

Segment kaydediciler ile adreslenen bir hafıza haritası örneği

Segment kaydedicileri ofset kaydediciler ile birlikte yazmak gerekirse;

CS:IP

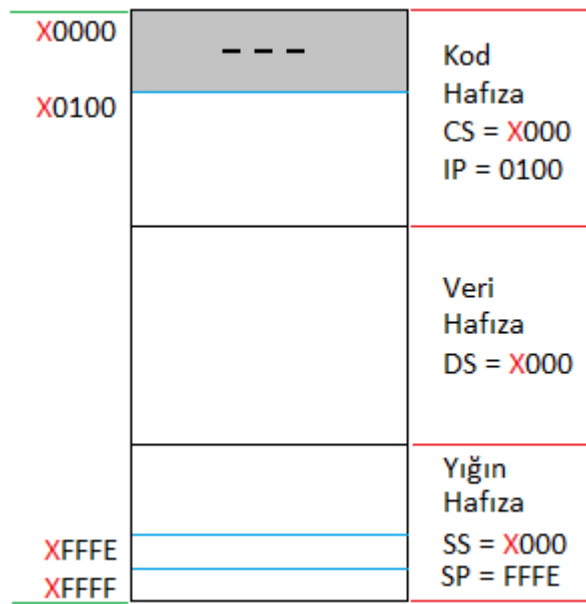
DS:BX veya DS:DI veya DS:SI

SS:SP veya SS:BP

ES:DI veya ES:SI

bu şekilde yazılabilir. Bunlar kullanılan komutlara göre otomatik olarak seçilirler.

64 kB'tan küçük programların, kod, veri ve yığın hafızaları tek bir segmente (X0000) konur. Yani CS, DS, SS ve ES'de aynı adres vardır. Kod (program) o segmentin ilk adresinden itibaren yerleştirilir. Yığın (stack) hafıza o segmentin son adresinden itibaren geriye doğru kullanılarak gelir. Veri hafızası ise program kodların bittiği adresten hemen sonra başlatılır.



Yığın (Stack) Hafıza

Sıralı erişim sağlayan, LIFO (Last Input First Output) sisteminde çalışan hafıza türüdür. Buradaki bilgiler 16 bittir. Çünkü burada kaydedicilerin değerleri saklanır. **Alt programlı veya interrupt (kesme) ile çalışıldığı zaman CPU'nun ana programa dönüşünü sağlamak ve işlemi kaldığı durumdan tekrar devam ettirebilmesi için bazı veri veya kayıtçıların tutulması** gerekir. Alt program veya interrupt' dan önce gerekli bilgiler (AX, BX, CS, DS....gibi) yığın belleğe atılır, alt program veya interruptlı çalışma bitince daha sonrada tekrar bu bilgiler yığından alınarak (konulan sıranın tersi sırada alınmalıdır, LIFO'nun özelliği) ana programın çalışmasına kalınan yerden devam etmesi sağlanır. POP ve PUSH komutları ile veri konup alınabilir.

BİLGİSAYAR MİMARİLERİ

1. Mikroişlemci Yapısal Tasarım Mimarisi
2. Mikroişlemci Komut Seti Mimarisi

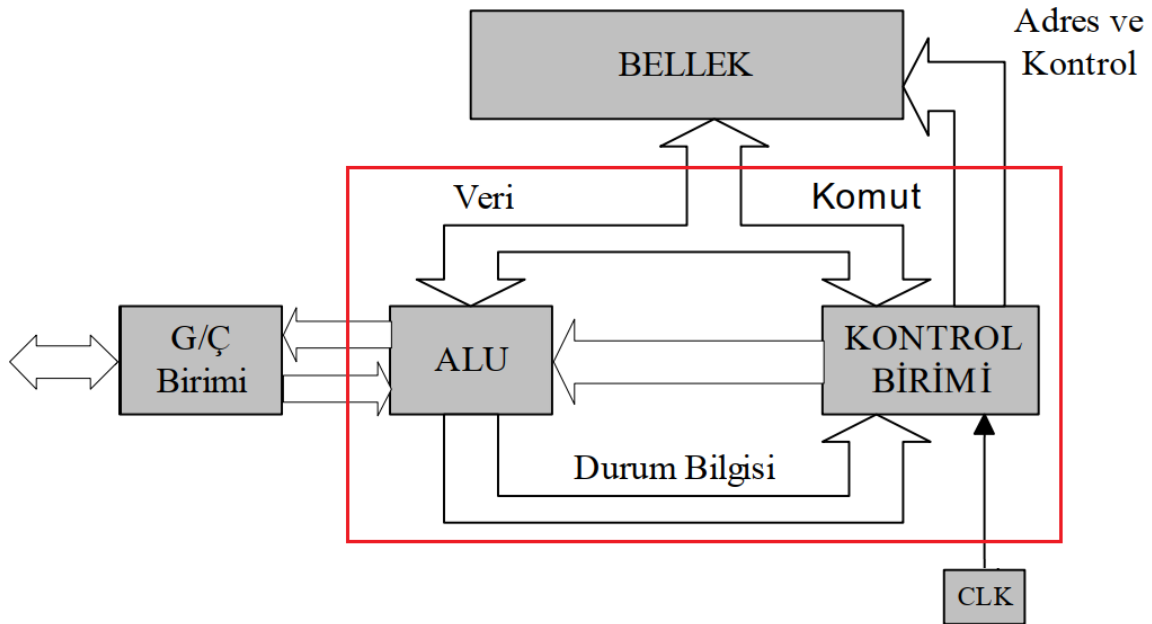
Mikroişlemci Yapısal Tasarım Mimarisi

Bilgisayarın yüklenen tüm görevleri çok kısa zamanda yerine getirmesinde yatan ana unsur bilgisayarın tasarım mimarisidir. Bir mikroişlemci, mimari yetenekleri ve tasarım felsefesiyle şekillenir;

1. Von Neuman Mimarisi
2. Harvard Mimarisi

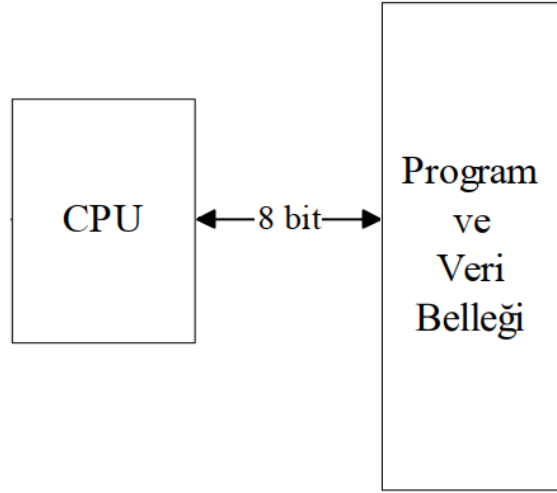
Von Neuman Mimarisi

Bilgisayarlarda ilk kullanılan mimaridir. İlk bilgisayarlar Von Neuman yapısından yola çıkılarak geliştirilmiştir. Geliştirilen bu bilgisayar beş birimden oluşmaktaydı. Bu birimler; aritmetik ve mantıksal birim, kontrol birim, bellek, giriş-çıkış birimi ve bu birimler arasında iletişimi sağlayan yollardan oluşur



Von Neuman Mimarili Bilgisayar Sistemi

Bu mimaride veri ve komutlar bellekten tek bir yoldan mikroişlemciye getirilerek işlenmektedir. Program ve veri aynı bellekte bulunduğu için, komut ve veri gerektiğinde aynı iletişim yolunu kullanmaktadır. Bu durumda, komut için bir al-getir saykılı, sonra veri için diğer bir al-getir saykılı gerekmektedir.



Von Neuman Mimari Blok Diyagramı

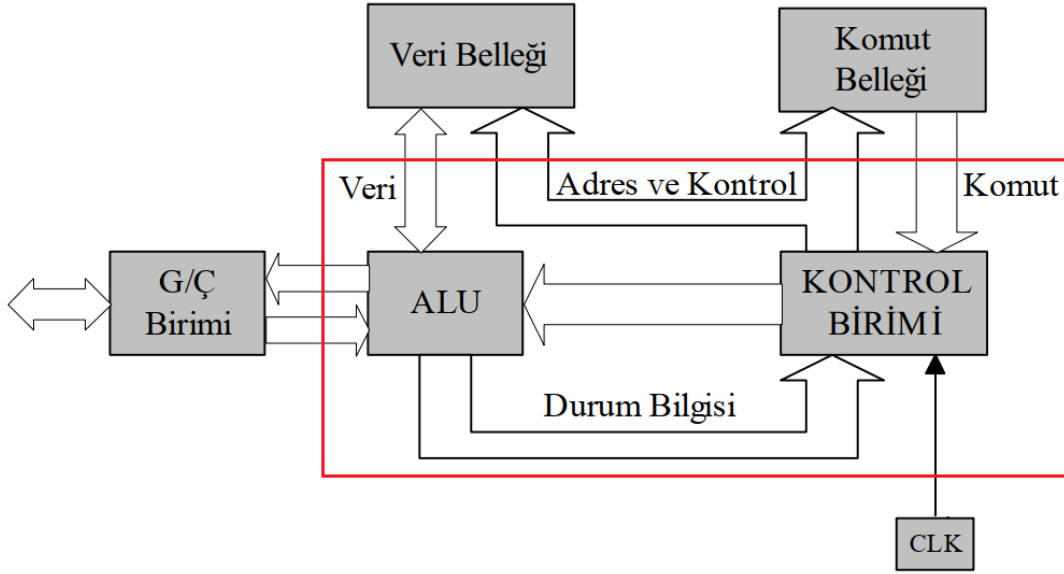
Von Neuman mimarisine sahip bir bilgisayar aşağıdaki sıralı adımları gerçekleştirir.

1. Program sayıcısının gösterdiği adresten (bellekten) komutu al-getir.
2. Program sayıcısının içeriğini bir artır.
3. Getirilen komutun kodunu kontrol birimini kullanarak çöz. Kontrol birimi, bilgisayarın geri kalan birimlerine sinyal göndererek bazı operasyonlar yapmasını sağlar.
4. 1. adıma geri dönlür.

Von Neuman mimarisinde, veri bellekten alınıp işledikten sonra tekrar belleğe gönderilmesinde çok zaman harcanır. Bu mimari yaklaşıma sahip olan bilgisayarlar günümüzde, verilerin işlenmesinde, bilginin derlenmesinde ve sayısal problemlerde olduğu kadar endüstriyel denetimlerde başarılı bir şekilde kullanılmaktadır. Genellikle **CPU**'larda bu yapı kullanılır.

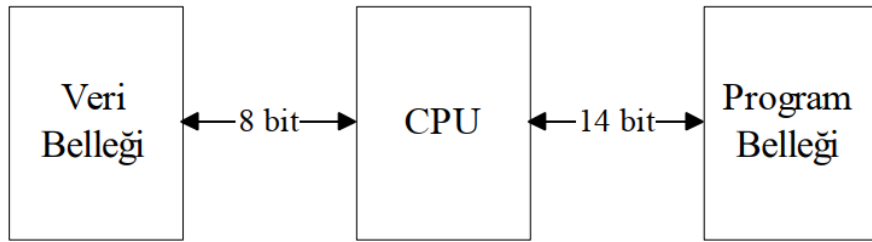
Harvard Mimarisi

Harvard mimarili bilgisayar sistemlerinin Von Neuman mimarisinden farkı veri ve komutların ayrı ayrı belleklerde tutulmasıdır. Buna göre, veri ve komut aktarımında iletişim yolları da birbirinden bağımsız yapıda bulunmaktadır.



Harvard Mimarili bilgisayar sistemi

Komutla birlikte veri aynı saykılta farklı iletişim yolundan ilgili belleklerden alınıp işlemciye getirilebilir. Getirilen komut işlenip ilgili verisi veri belleğinden alınırken sıradaki komut, komut belleğinden alınıp getirilebilir. Bu önden alıp getirme işlemi, dallanma haricinde hızı iki katına çıkarabilmektedir.



Harvard Mimarisi Blok Diyagramı

Bu mimari günümüzde daha çok sayısal sinyal işlemcilerinde (DSP) kullanılmaktadır. Bu mimaride program içerisinde döngüler ve zaman gecikmeleri daha kolay ayarlanır. Von Neuman yapısına göre daha hızlıdır. Özellikle PIC **mikrodenetleyicilerinde** bu yapı kullanılır.

Mikroişlemci Komut Tasarım Mimarileri

1. CISC (Complex Instruction Set Computer) Mimarisi
2. RISC (Reduced Instruction Set Computer) Mimarisi

CISC (Complex Instruction Set Computer) Mimarisi

CISC (*Karmaşık Komut Seti Bilgisayarı*) mimarisi, programlanması kolay ve etkin bellek kullanımı sağlayan tasarım felsefesinin bir ürünüdür. İşlemci üzerinde performans düşüklüğü ve işlemcinin karmaşık bir hale gelmesine neden olsa da yazılımı basitleştirmektedir. Bu mimarinin en önemli iki özelliği, değişken uzunluktaki komutlar diğeri ise karmaşık komutlardır. Değişken ve karmaşık uzunluktaki komutlar bellek tasarrufu sağlar. Karmaşık komutlar birden fazla komutu tek bir hale getirirler. Karmaşık komutlar aynı zamanda karmaşık bir mimariyi de oluşturur. Mimarideki karışıklık işlemcinin performansını da doğrudan etkilemektedir. Bu sebepten dolayı çeşitli istenmeyen durumlar ortaya çıkabilir. CISC komut seti mümkün olabilen her durum için bir komut içermektedir. CISC mimarisinde yeni geliştirilen bir mikroişlemci eski mikroişlemcilerin assembly dilini desteklemektedir.

CISC mimarisi çok kademeli işleme modeline dayanmaktadır. İlk kademe, yüksek seviyeli dilin yazıldığı yerdir. Sonraki kademeyi ise makine dili oluşturur. Burada yüksek seviyeli dilin derlenmesi ile bir dizi komutlar makine diline çevrilir. Bir sonraki kademede makine diline çevrilen komutların kodları çözülerek, mikro-kodlara çevrilir. En son olarak da işlenen kodlar gerekli olan görev yerlerine gönderilir.

CISC Mimarisinin Avantajları

- Mikroprogramlama assembly dilinin yürütülmesi kadar kolaydır ve sistemdeki kontrol biriminden daha ucuzdur.
- Yeni geliştirilen mikrobilgisayar bir öncekinin assembly dilini desteklemektedir.
- Verilen bir görevi yürütmek için daha az komut kullanılır. Böylece bellek daha etkili kullanılır.
- Mikroprogram komut kümeleri, yüksek seviyeli dillerin yapılarına benzer biçimde yazıldığından derleyici karmaşık olmak zorunda değildir.

CISC Mimarisinin Dezavantajları

- Gelişen her mikroişlemci ile birlikte komut kodu ve yonga donanımı daha karmaşık bir hale gelmiştir.
- Her komutun çevirim süresi aynı değildir. Farklı komutlar farklı çevrim sürelerinde çalıştıkları için makinenin performansını düşürecektir.
- Bir program içerisinde mevcut komutların hepsi kullanılamaz.
- Komutlar işenirken bayrak bitlerinin dikkat edilmesi gerekir. Buda ek zaman süresi demektir. Mikroişlemcinin çalışmasını etkilemektedir.

RISC (Reduced Instruction Set Computer) Mimarisi

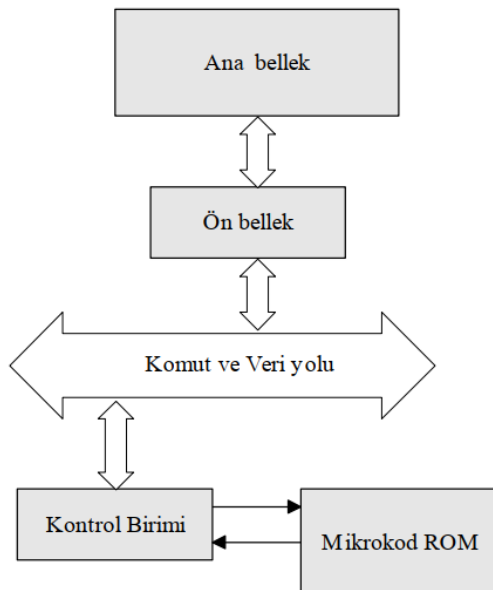
RISC (*Azaltılmış Komut Seti Bilgisayarı*) mimarisi IBM, Apple ve Motorola gibi firmalarca sistematik bir şekilde geliştirilmiştir. RISC mimarisinin taraftarları, bilgisayar mimarisinin gittikçe daha karmaşık hale geldiğini ve hepsinin bir kenara bırakılıp en başta yeniden başlamak fikrindeydiler. 70’li yılların başında IBM firması ilk RISC mimarisini tanımlayan şirket oldu. Bu mimaride bellek hızı arttığından ve yüksek seviyeli diller assembly dilinin yerini aldığından, CISC’in başlıca üstünlükleri geçersiz olmaya başladı.

RISC’in felsefesi üç temel prensibe dayanır.

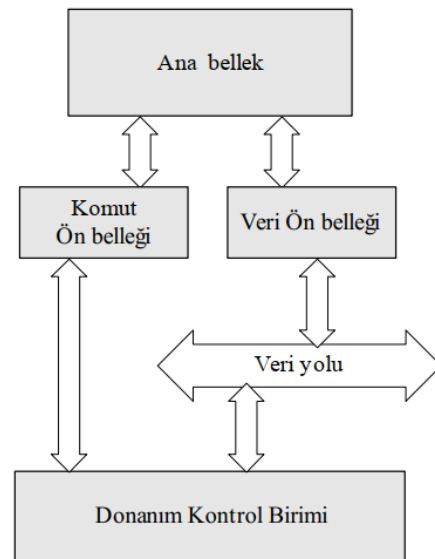
- Bütün komutlar tek bir çevrimde çalıştırılmalıdır: Her bir komutun farklı çevrimde çalışması işlemci performansını etkileyen en önemli nedenlerden biridir. Komutların tek bir çevrimde performans eşitliğini sağlar.
- Belleğe sadece “load” ve “store” komutlarıyla erişilmelidir. Eğer bir komut direkt olarak belleği kendi amacı doğrultusunda yönlendirilirse onu çalıştırmak için birçok saykıl geçer. Komut alınıp getirilir ve bellek gözden geçirilir. RISC işlemcisiyle, belleğe yerleşmiş veri bir kaydediciye yüklenir, kaydedici gözden geçirilir ve son olarak kaydedicinin içeriği ana belleğe yazılır.
- Bütün icra birimleri mikro-kod kullanmadan donanımdan çalıştırılmalıdır. Mikro-kod kullanımı, dizi ve benzeri verileri yüklemek için çok sayıda çevrim demektir. Bu yüzden tek çevrimli icra birimlerinin yürütülmesinde kolay kullanılmaz.

RISC mimarisi küçültülen komut kümesi ve azaltılan adresleme modları sayısı yanında aşağıdaki özelliklere sahiptir.

- Bir çevrimlik zamanda komut işleyebilme
- Aynı uzunluk ve sabit formatta komut kümesine sahip olma
- Ana belleğe sadece “load” ve “store” komutlarıyla erişim; operasyonların sadece kaydedici üzerinde yapılması
- Bütün icra birimlerinin mikro-kod kullanmadan donanımsal çalışması
- Yüksek seviyeli dilleri destekleme
- Çok sayıda kaydediciye sahip olması



Mikro-kod denetimli CISC mimarisi



Donanım denetimli RISC mimarisi

RISC Mimarisinin Üstünlükleri

- **Hız:** Azaltılmış komut kümesi, kanal ve süperskalar tasarıma izin verildiğinden RISC mimarisi CISC işlemcilerin performansına göre 2 veya 4 katı yüksek performans gösterirler.
- **Basit donanım:** RISC işlemcinin komut kümesi çok basit olduğundan çok az yonga uzayı kullanılır. Ekstra fonksiyonlar, bellek kontrol birimleri veya kayan noktalı aritmetik birimleri de aynı yonga üzerine yerleştirilir.
- **Kısa Tasarım Zamanı:** RISC işlemciler CISC işlemcilere göre daha basit olduğundan daha çabuk tasarlanabilirler.

RISC Mimarisinin Eksiklikleri:

CISC tasarım stratejisinden RISC tasarım stratejisine yapılan geçiş kendi problemlerinde beraberinde getirmiştir. Donanım mühendisleri kodları CISC işlemcisinden RISC işlemcisine aktarırken anahtar işlemleri göz önünde bulundurmak zorundadırlar.

CISC ve RISC Tabanlı İşlemcilerin Karşılaştırılması

CISC ve RISC tabanlı işlemcilerin karşılaştırılmasında iki önemli faktör farklılıklarını ortaya çıkarmada yeterlidir.

Hız: Genelde RISC çipleri kanal tekniği kullanarak eşit uzunlukta segmentlere bölünmüş komutları çalıştırmaktadır. Kanal tekniği komutları kademeli olarak işler ki bu RISC'in bilgi işlemini CISC'den daha hızlı yapmasını sağlar RISC işlemcisinde tüm komutlar 1 birim uzunlukta olup kanal tekniği ile işlenmektedir. Bu teknikte bazıları hariç komutlar, her bir basamağında aynı işlemin uygulandığı birimlerden geçerler.

Transistör Sayısı: CISC mimarisinde kullanılan transistör sayısı RISC'e nazaran daha fazladır. Transistör sayısının bir yerde çok olması fazla yerleşim alanı ve ayrıca fazla ısı demektir. Bundan dolayı da fazla ısı üretimi soğutma olayını gündeme getirmektedir. CISC tabanlı Pentium işlemcilerde karışık ısı dağıtıcısı veya soğutma fanlar kullanılmaktadır.

RISC mimarisindeki önemli üstünlüklere karşı bazı mahzurları ortaya çıkmaktadır. RISC mimarisi, CISC'in güçlü komutlarından yoksundur ve aynı işlemi yapmak için daha fazla komut işlenmesini gerektirir. Bundan dolayı da RISC'in bant genişliği artar. Bu sistemde güçlü komutların yokluğu ikinci bir yardımcı işlemciyle ya da işlemci içinde oluşturulacak ayrı bir pipeline bölümüyle giderilebilir. Komut ön-belleğinin kullanılması yüksek komut alıp getirme işlemini azaltmaktadır. RISC mimarisi diğerine nazaran daha kompleks yazılımlara ihtiyaç duyar.

RISC (Hard-wired Control Unit)	CISC (Microprogrammed Control Unit)
Hızlı	Nispeten yavaş
Ucuz	Pahalı
Yeniden dizayn zor	Esnek
Daha az komut (instruction)	Daha fazla komut (instruction)
Daha fazla saklayıcı bellek (register)	Daha az saklayıcı bellek (register)

EPIC (Explicitly Parallel Instruction Computing) Mimarisi

Bu mimari RISC ve CISC mimarisinin üstün yönlerinin bir arada bulunduğu 64 bitlik komut kümesi olan bir mimari türüdür. EPIC mimarisi, işlemcinin hangi komutların paralel çalışabildiğini denetlemesi yerine, EPIC derleyicisinden açık olarak hangi komutların paralel çalışabildiğini bildirmesini ister. Çok uzun komut kelimesi (VLIW (Very Long Instruction Word)) kullanan bilgisayarlar, yazılımın paralelliğine ilişkin kesin bilgi sağlanan mimari örneklerdir. EPIC varolan VLIW mimarisinin dallanma sorunlarını çözmeye çalışarak daha ötesine gitmeyi hedeflemektedir. Derleyici programdaki paralelliği tanımlar ve hangi işlemlerin bir başkasından bağımsız olduğunu belirleyerek donanıma bildirir. EPIC mimarisinin ilk örneği, IA-64 mimarisine dayalı Itanium işlemci ailesidir.

EPIC Mimarisin Üstünlükleri

- Paralel çalıştırma (çevrim başına birden çok komut çalıştırma)
- Tahmin kullanımı
- Derleme anında paralelizmi tanıyan derleyiciler
- Büyük bir kaydedici kümesi
- Dallanma tahmini ve bellek gecikmesi problemlerine karşı üstün başarı
- Gelişme ile birlikte eskiye karşı uyumluluk