



Bellek

Prof. Dr. Eşref ADALI

Doç. Dr. Şule Gündüz Öğüdücü

Sürüm-B



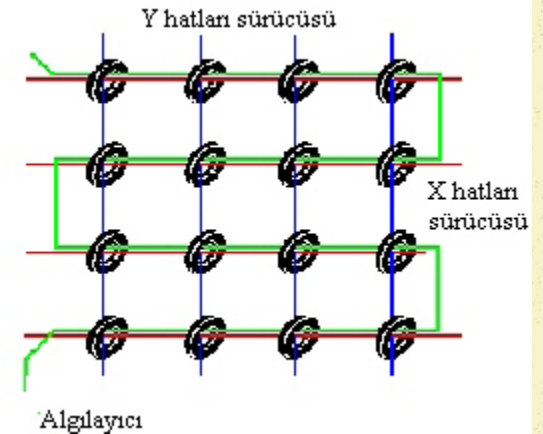
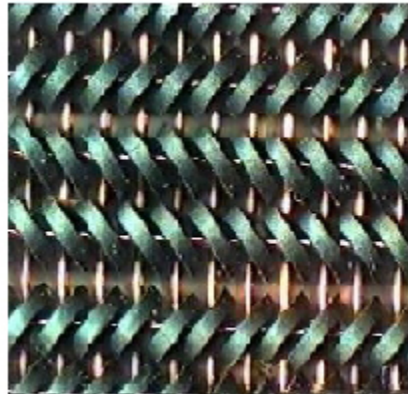
Konular



- Bellek tanımı
- Saltoku bellekler
- Oku/Yaz Bellekler
- Bellek Düzeni
- Belleğe Erişim
- Bellek Tasarımı

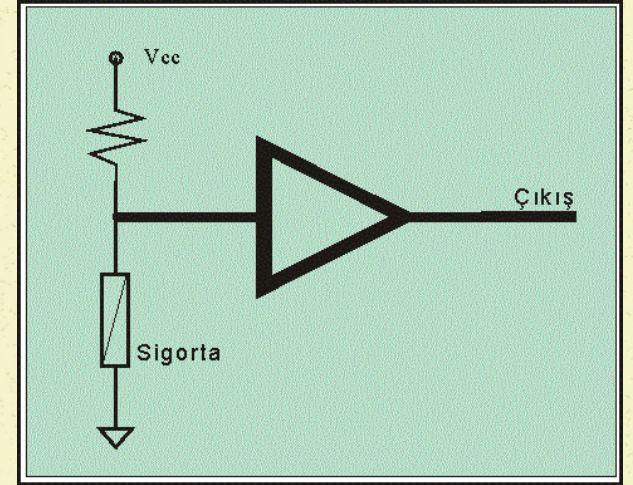
Bellek Tanımı

- Bilgisayarın çalışmasına yön verecek programı ve programın üzerinde çalışacağı verilerin saklandığı yere bellek diyoruz.
- Bilgisayarın ilk dönemlerinde, bellek olarak mekanik çarklar ve delikli kartlar kullanılırken elektronik bilgisayarlarda, önceleri röleler ve daha sonraları çekirdek bellekler kullanılmıştır. Yakın zamanda kullanılan bellekler ise yarıiletken teknolojisi ile üretilmektedir.
- Çekirdek bellekler, simit biçimindeki manyetik halkaların içinden geçirilen akımla kutuplanması ve daha sonra, yardımcı sargılarla kutuplanma yönünün öğrenilmesi ilkesine göre çalışmaktadır.
- Çekirdek belleğin iyi yönü ise, belleğe yazılmış olan bilgilerin saklanması için enerji gerektirmemesidir. Bu nedenle, bilgisayar kapatılmış olsa bile bellekteki bilgiler korunabilir.



Saltoku Bellekler

- **ROM (Saltoku Bellek) :** Bu bellek türüne bilgi yazılması, belleğin üretimi sırasında gerçekleştirilmektedir. ROM türü belleklerin üretimi için gerekli maskelerin hazırlanma maliyeti oldukça yüksektir. Bu nedenle çok sayıdaki üretimler için uygundur.
- **PROM (Programlanabilir Saltoku Bellek) :** Üretildikleri an bütün gözeleri (en küçük bellek birimi) 0 veya 1 ile yüklü belleklerdir. Her bellek gözesi içinde bir sigorta bulunmaktadır. Bu sigortalar, özel bir yöntem ve aygıt aracılığı ile atırılabilir. Bir gözenin sigortasının atmış olması, o gözenin konumunun değişmesi demektir. Sözgelimi tüm gözeleri 0 olan bir belleğin istenen gözeleri 1 konumuna getirilerek programlanmış olur. Ancak 1 konumuna getirilmiş gözenin tekrar 0 konumuna dönme şansı yoktur. Bir başka deyişle atmış bir sigortanın yenilenmesi söz konusu değildir.



Saltoku Bellekler

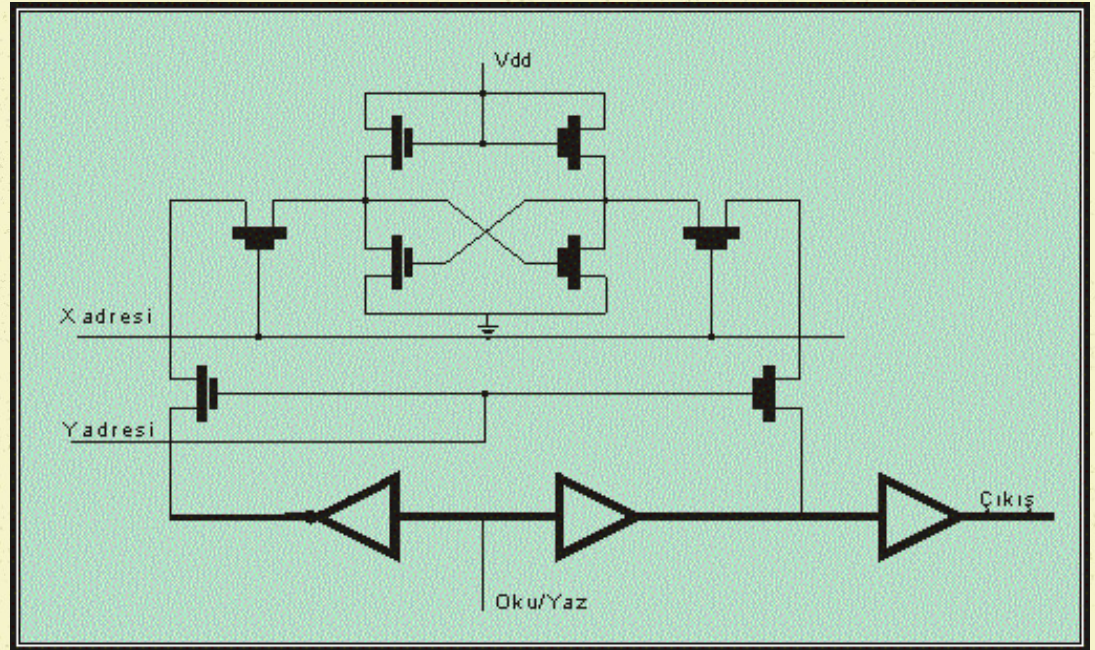
- **EPROM (Silinebilir Programlanabilir Saltoku Bellek) :** EPROM'lar üretildiklerinde tüm bellek gözeleri 1 konumundadır. 1 konumunda olan gözelerden istenenler, özel yöntemler ve aygıtlarla 0 konumuna geçirilebilir. Mor ötesi ışığın yarıiletken üzerine belli bir süre tutulması sonunda, tüm gözeler 1 konumuna gelirler. Böylece EPROM silinmiş olur. EPROM'u silmek için EPROM'un gövdesi üzerinde bir pencere bulunmaktadır. Aydınlatmanın tek bir göze için yapılamaması nedeniyle, tek bir göz veya gözenin 1 konumuna getirilmesi olanağı yoktur. Ancak tüm bellek silinebilir.
- **EEPROM (Elektrikle Silinebilir Programlanabilir Saltoku Bellek) :** Silinebilir ve programlanabilir belleklerin en gelişmiş olanı, elektriksel olarak silinebilen salt oku belleklerdir. Bu belleklerde, bellek gözlerine istenen bir değer yazılabilir ve yazılan bu bilgi yeni bir yazmaya kadar kalır. Bellek gözesine yazılan bilgi 0 ve 1'lerden oluşabilir. Başka bir deyişle, bir gözenin içeriği 0 ya da 1 konumuna getirilebilir. EEPROM'lara veri yazılmasında da özel yöntem ve aygıtlardan yararlanılır.

Statik Oku/Yaz Bellek

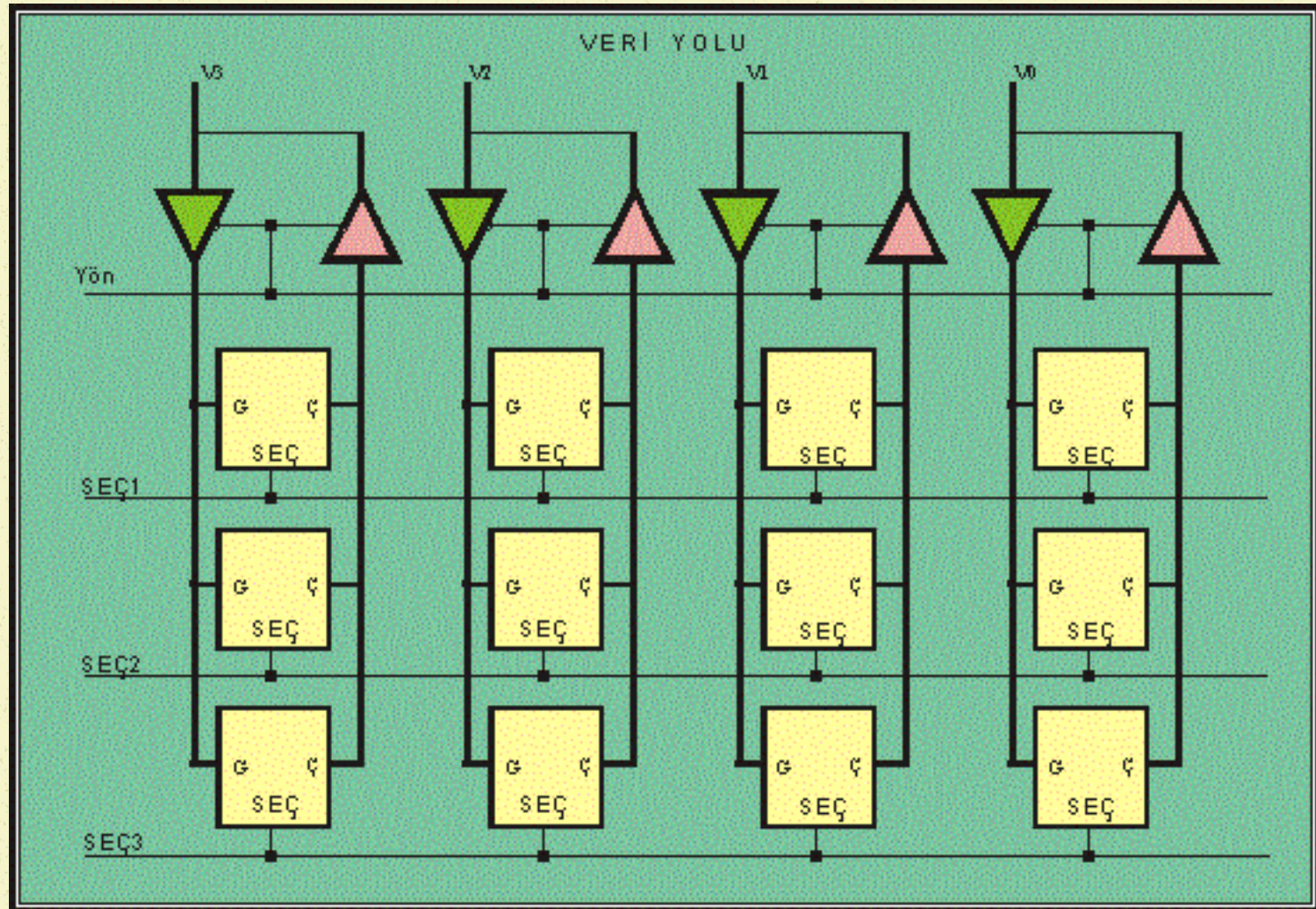
Bilgisayar içindeki kullanıcı programının yazılacağı veya verilerin yazılacağı bellek türü Oku/Yaz bellektir. Oku/Yaz türü belleklerin statik ve dinamik olarak iki türü vardır.

Statik Oku/Yaz bellekler :
Statik Oku/Yaz belleğin her bir gözesi aslında bir flip-flop tur. Statik Oku/Yaz belleği besleyen enerji kesildiğinde, bellek içindeki tüm veriler kaybolmaktadır.

Statik bellek gözesi



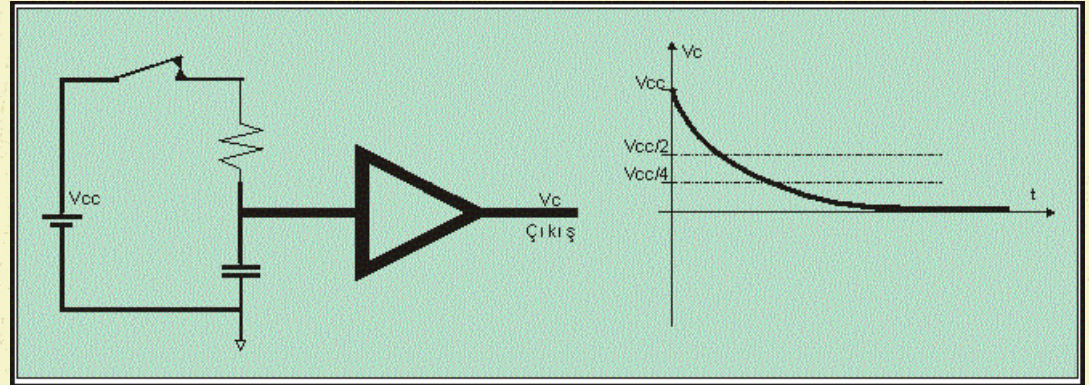
Statik Oku/Yaz Bellek



Dinamik Oku/Yaz Bellek

Dinamik bellek gözesi

- **Dinamik Oku/Yaz bellekler :**
Dinamik bellek gözesi temelde bir kapasite ve bir sürücünden oluşur. Sürücü, genelde tek bir tranzistörle gerçekleştirir.



- Bir gözeyi 1 konumuna getirmek için kapasitenin doldurulması gerekir. Dolan kapasite, kaçak akımlar nedeniyle zaman içinde boşalır; kapasite uçlarındaki gerilimin zamana bağlı olarak azalır.
- Eğer kapasitenin boşalma karakteristiği bilinir ve kapasite uçlarındaki gerilim yarı veya dörtte bir değerine inmeden okunursa 1 olarak algılanabilir. Eğer bir gözenin 1 olduğu algılanırsa kapasite yeniden doldurulur, 0 olduğu algılanırsa doldurulmaz. Böylece bellekte bilgilerin saklanması sağlanmış olur. Bu işleme dinamik belleğin tazelenmesi denir.

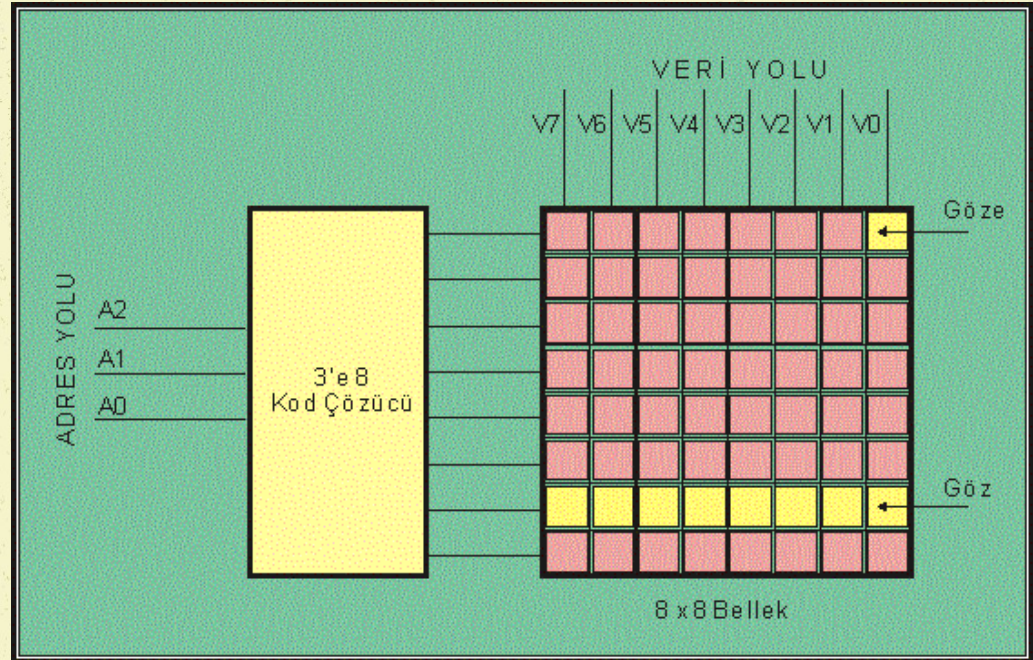
Bellek Düzeni

- Belleklerin cinsi ne olursa olsun, kullanım açısından aynı özeliği gösterirler. Belleğin temel birimi **göze**'dir. Gözeler yan yana konarak bir bellek **gözü** 'ü oluşturur. Bellek gözlerinin üst üste konmasıyla da **bellek** oluşur.

Bellek gözeleri yan yana konarak bir gözünü ve gözler üst üste konarak belleği oluşturur. Belleğe, $N \times M$ boyutlu bir matris gibi bakabiliriz. N satır sayısına, dolayısıyla bellek gözü sayısına, M ise sütun sayısına, dolayısıyla bir gözdeki göze sayısına karşı düşmektedir.

Belleğin aynı sütunda bulunan tüm gözelerinin giriş ve çıkış uçları birbirine bağlıdır. Dolayısıyla, belleğin göz sayısı ne olursa olsun veri bağlantı ucu, bir gözdeki göze sayısına eşittir.

Tek boyutlu adresleme



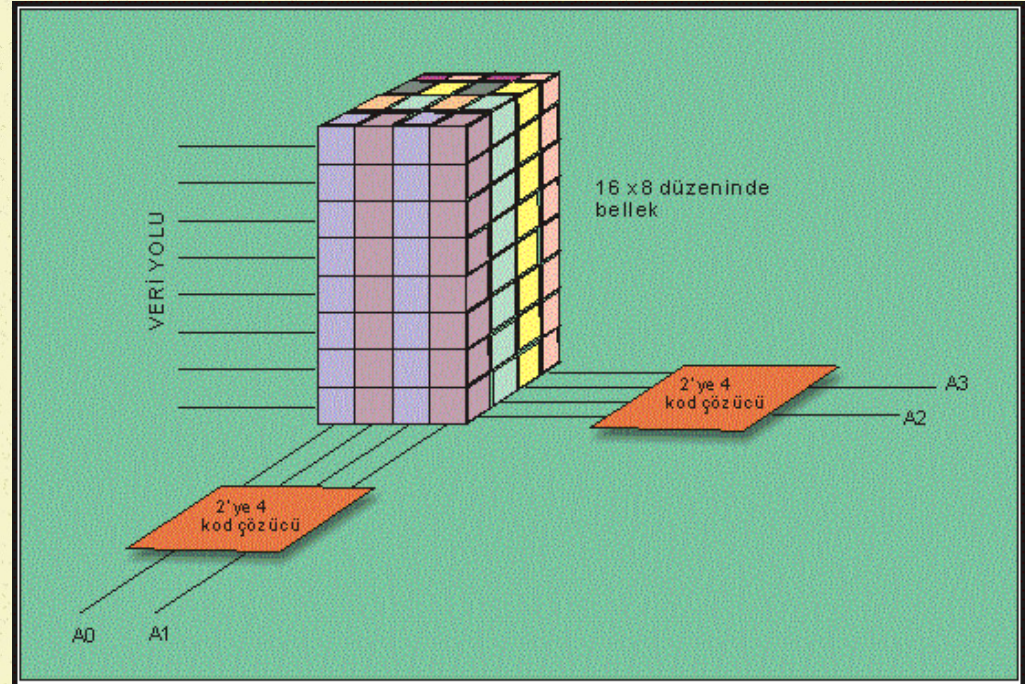
İki Boyutlu Adresleme

İki boyutlu adreslemenin sağladığı kolaylık şekilde görülmektedir. 32 gözünü olan belleğe erişmek için tek boyutlu adresleme yöntemi kullanıldığında 4x32 kod çözücü kullanılır; dolayısıyla, bellekteki gözleri seçmek için 32 bağlantı gerekir.

İki boyutlu adresleme yönteminde iki tane 2x4 kod çözücü kullanılır; dolayısıyla bellekteki gözleri seçmek için 8 bağlantı yeterlidir.

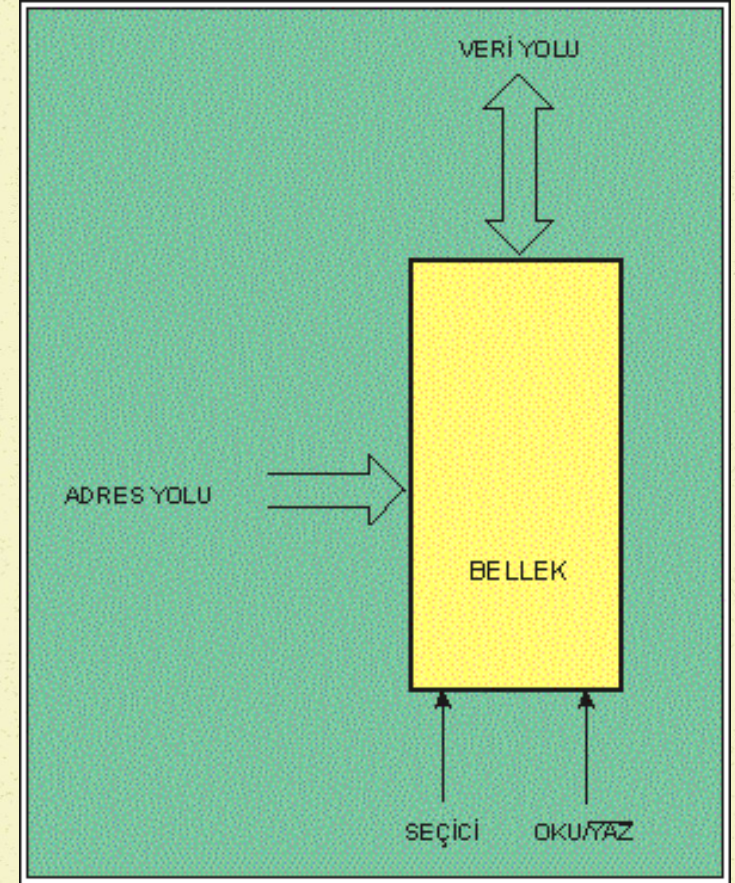
1024 gözünü olan bir belleği iki boyutlu adreslemek için iki tane 5x32 kod çözücü kullanılır. Bu durumda toplam 64 hat yeterlidir. de gösterilen bu bağlantılar ise şunlardır:

İki boyutlu adresleme



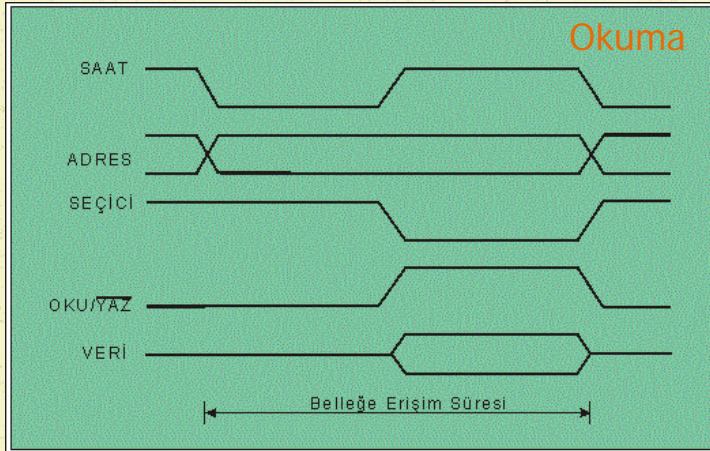
Belleğin Dış Görünümü

- **Veri Yolu** : Bellek gözelerinin, bir diğer ortama bağlantısını sağlar. Oku/Yaz türü belleklerde, veri alışverişi iki yönlü, salt oku belleklerde sadece okuma yönündedir. Veri yolundaki hat sayısı, bellek gözündeki göze sayısına eşittir.
- **Adres Yolu** : Bellek gözlerinin seçilmesi için gerekli olan, ikilik düzende oluşturulmuş adres bilgisini taşır.
- **Oku/Yaz** : Bellek ile veri yolu arasındaki veri akışının belirlenmesini sağlar. Okuma konumunda (1 konumu), adres yolunun belirlediği bellek gözünün içeriği veri yoluna alınır. Yazma konumunda (0) ise, veri yolu üzerindeki veriler adres yolunun belirlediği bellek gözüne yazılır.
- **Seçici** : Bir bellek kırımlığının seçilebilmesi için bu girişin etkin hale getirilmesi gerekir. Seçici girişler genellikle 0 'da etkin olurlar. Bellek kırımlığının seçilmemesi durumunda, belleğin veri giriş-çıkış uçları üçüncü konuma geçerler. Başka bir deyişle, bellek kendisini bağlı olduğu veri yolundan yalıtır.



Belleğe Erişim

- Bir bellek gözünün içeriğinin okunması veya bir bellek gözüne veri yazılması süresine, **belleğe erişim süresi** denilmektedir. Bu özellik, bilgisayarın çalışma hızını doğrudan etkilediği için, belleğin en önemli özeliği sayılabilir.

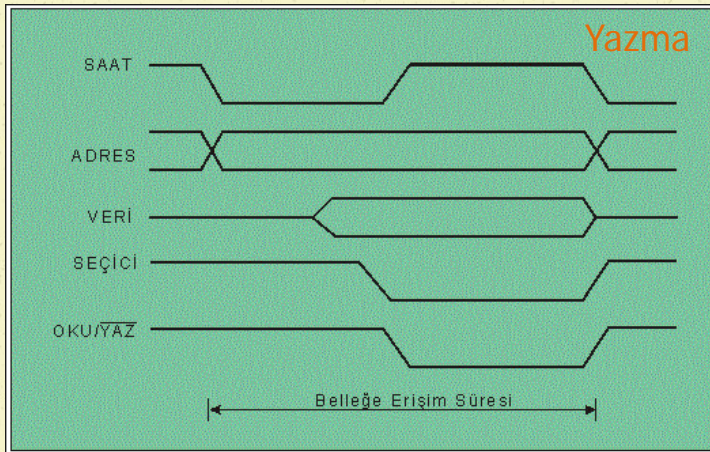


OKUMA

- Saat işaretinin 1 den sıfıra inmesiyle, adres bilgileri adres yoluna konur.
- Adres yolu üzerinde adres bilgisinin oluşmasının ardından kırmık seçici canlanır; kod çözücü belirlenen bellek gözünü seçer.
- Bellek gözünün seçilmesinin ardından oku emrinin gönderilmesi ile, seçilen bellek gözünün içeriği veri yoluna aktarılır.

YAZMA

- Saat işaretinin 1 den sıfıra inmesiyle, adres bilgileri adres yoluna konur.
- Belleğe gönderilecek veri veri yoluna konur.
- Adres yolu üzerinde adres bilgisinin oluşmasının ardından kırmık seçici canlanır; kod çözücü belirlenen bellek gözünü seçer.
- Yaz emrinin ardından, veri yolundaki veri seçilen bellek gözüne yazılır.

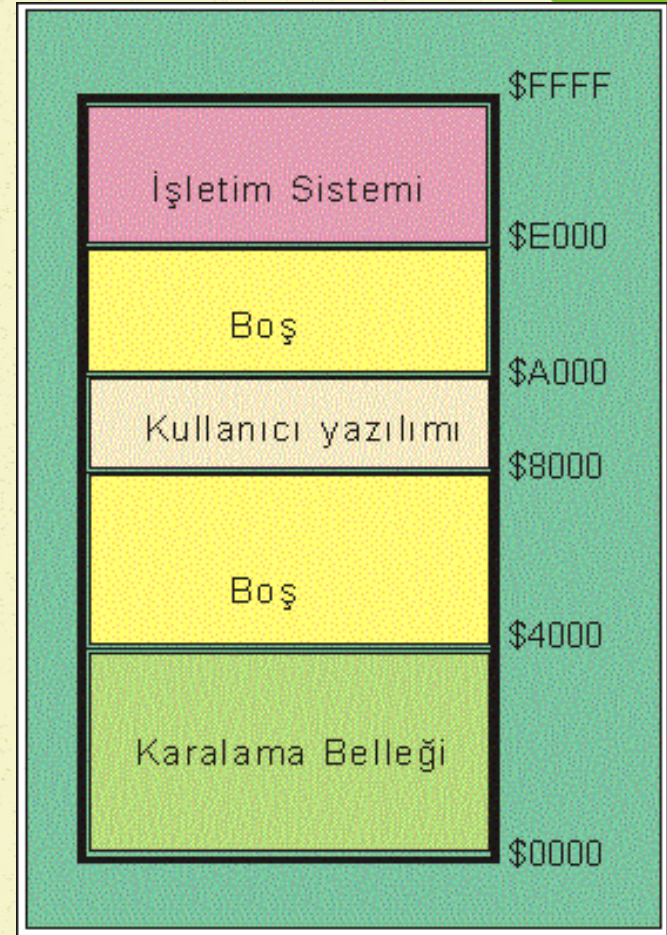


Bellek Tasarımı

- Bilgisayarın yeteneğini arttırmak, belli ölçüde, bellek kapasitesini artırmaya bağlıdır. Bellek kırmıklarının sığaları ne kadar artarsa artsın, bir bilgisayarın belleği tek bir bellek kırmığı ile gerçekleştirilemez. Gerekli olan bellek birden fazla bellek kırmığının birbirine eklenmesi ile sağlanacaktır. Ekleme boyuna ve enine gerçekleştirilebilir:

Boyuna Genişletme

- 8 bitlik mikroişlemcilerde adres yolu, genellikle, 16 bitlidir. Bu nedenle bu sınıf mikroişlemciler en çok 64 K bellek gözünü adresleyebilir. 16, 32 veya 64 bitlik mikroişlemcilerde ise adres yolu 24 bit veya daha fazladır.
- Bir bilgisayarda, MIB'in adresleyebileceği kadar belleğin bulunması her zaman gerekli değildir. Kullanılacak belleklerin türleri ve hangi adreste hangi tür belleğin bulunacağı belirlenir. Bir bilgisayar belleğinde, hangi adresler arasında bellek bulunduğu, kullanılan belleğin cinsi ve kullanım amacını belirtmek için **bellek haritası** denilen çizim kullanılır.

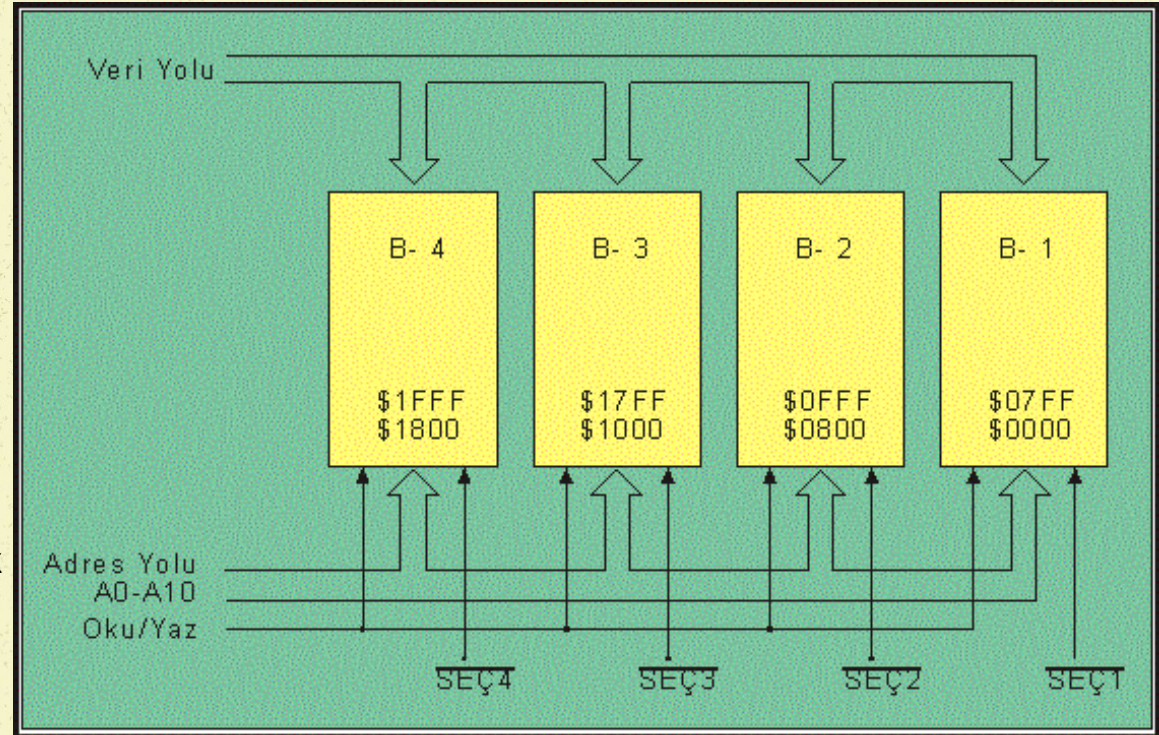


Boyuna Bellek Geniřletme -I

- 8-bitlik veri yolu ve 16 bitlik adres yolu olan bir mikrořlemci için \$0000 adresinden bařlayan ve kesintisiz olarak \$1FFF adresine kadar devam eden Oku/Yaz türü bellek tasarlanacaktır. Bu tasarımda kullanılacak bellek kırmıkları 2Kx8 boyutundadır.
- 1.Adım :** Verilerden anlařılacađı üzere, istenen toplam bellek sığası 8Kx8 dir. Buna göre eldeki bellek kırmıklarından dört adet kullanmak gerekir. Her bellek kırmıđına düşecek adres alanları ise řöyle olacaktır:
 - Birinci bellek kırmıđı \$0000 - \$07FF
 - İkinci bellek kırmıđı \$0800 - \$0FFF
 - Üçüncü bellek kırmıđı \$1000 - \$17FF
 - Dördüncü bellek kırmıđı \$1800 - \$1FFF
- Böylece \$0000 - \$1FFF aralıđında sürekli olan bir bellek elde edilecektir.
- 2.Adım :** Dört bellek kırmıđının veri yolları, adres yolları ve Oku/Yaz giriřleri birbirine bađlanır. Belleklerin veri yollarının birbirine bađlanması sırasında, her bellek kırmıđının aynı sırada bulunan veri hatları birbirine bađlanır ve sonunda mikrořlemcinin veri hattı ile birleřtirilir.
- 2K' lık bir bellek içindeki gözleri adresleyebilmek için (211 = 2048) 11 adres hattının kullanılması gerekir. Bir bařka deyiře, A0' dan A10' a kadar olan adresler, 2 K' lık bir belleđin gözlerinin adreslenmesi için gereklidir. Bu nedenle, her bellek kırmıđının adres uçları aynı sırada olmak üzere birbirine bađlanır.
- Bellek kırmıklarında bulunan Oku/Yaz giriřleri birbirine bađlanır ve sonunda, MİB'in Oku/Yaz çıkıřına bađlanır.

Boyuna Bellek Geniřletme -II

- **3. Adım :** Veri yolları, adres yolları ve Oku/Yaz giriřleri ortak baėlanan drt bellek kırıėı indeki bellek gzlerini, birbirini izleyecek biimde dzenlemek iin kırıık seicisinden yararlanılmaktadır.
- Kırıık seici, bir kırıık indeki tm gzlerin seilip seilmemesini belirleyen bir giriřtir.
- Kırıık seici sayesinde, \$0000 - \$07FF adresleri arasındaki bir adres geldiėinde sadece birinci bellek kırıėı seilebilir. Benzer řekilde \$0800 - \$0FFF arasındaki bir adreslemede ancak ikinci bellek kırıėı seilebilir. Kırıık seicinin, uygun adres aralıėında canlandırılması iřlemi bellekler iin kullanılmayan adres hatları kullanılarak saėlanır.



Boyuna Bellek Geniřletme -III

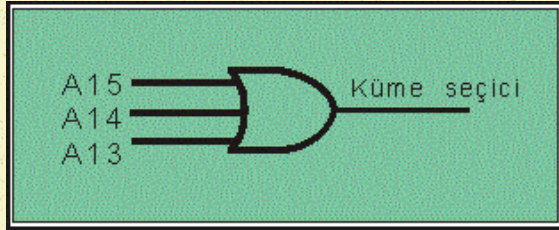
	Adres hatları					
	A15	A14	A13	A12	A11	A10A0
1. Bellek	0	0	0	0	0	Bellek içindeki, bellek gözlerinin adreslenmesi için kullanılır.
2. Bellek	0	0	0	0	1	
3. Bellek	0	0	0	1	0	
4. Bellek	0	0	0	1	1	

Tabloya göre řu sonuçlara varılır:

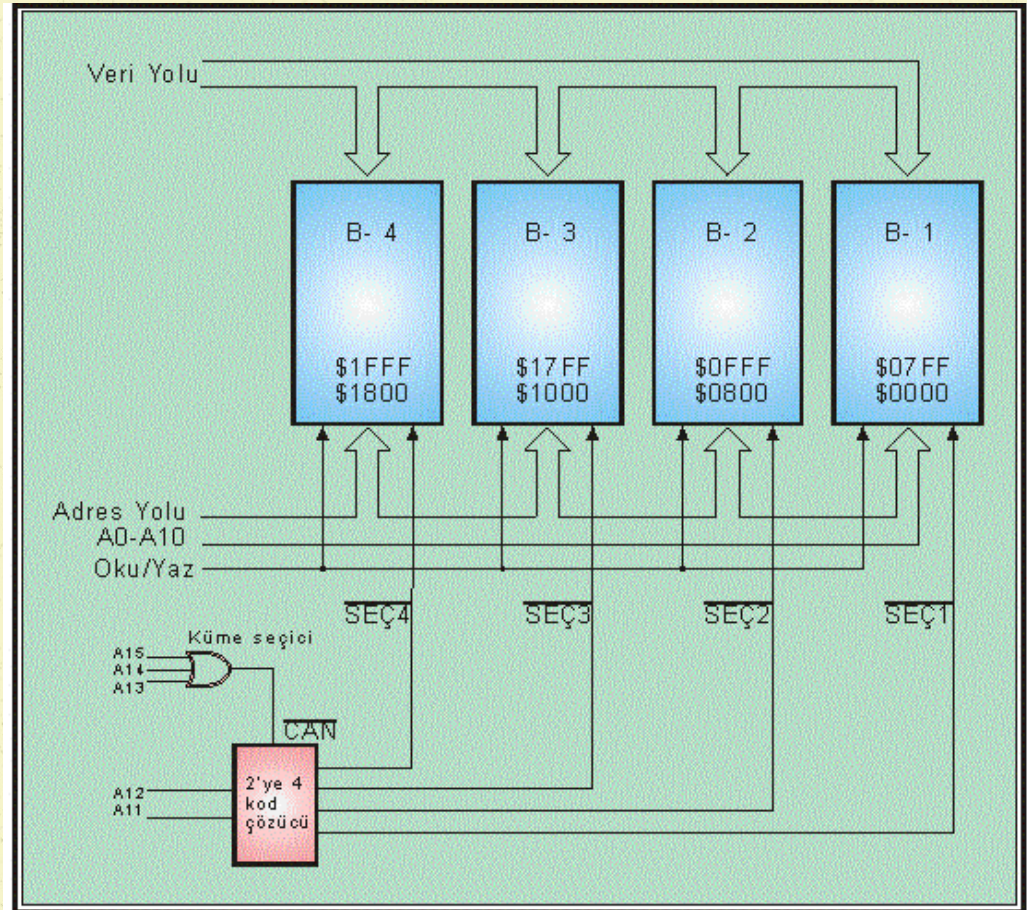
- Bu bellek kümesi için A15, A14, A13 hep 0 olarak kalmaktadır.
- A12 ve A11 dörk bellek kırmığının seçimini yapmaktadır.

Boyuna Bellek Geniřletme -IV

4. Adım : Kırmık seçici devrenin tasarlanmasında ilk adım, bu dört bellek kırmığı için sabit kalan A15, A14 ve A13 hatlarının durumlarını belirleyecek devrenin tasarlanmasıdır. Bu amaçla bir VEYA kapısı kullanılabilir. 8Kx8 lik belleğin seçimini sağlayan bu tasarıma, Küme Seçici Devre denilmektedir.



Kırmıkların seçimi, daha önce değinildiği gibi A12 ve A11 tarafından yapılmaktadır. Bu amaçla 2X4 kod çözücü kullanılabilir. Kod çözücü , küme seçici çıkışı ile etkin hale getirilecektir. Kod çözücünün çıkışları, belleklerin seçme girişlerine bağlanır.



Enine Bellek Geniřletme -I

- Statik belleklerin bir gözünde genellikle 8 göze bulunmasına karşın, dinamik belleklerin bir gözünde bir göze bulunmaktadır. Örneğin 8Kx1 düzenindeki kırmıklar kullanarak 8Kx8' lik bellek gerçeklemek için yapılması gereken işlemler şunlardır:
- **1. Adım :** Tasarım için gerekli olan 8 belleğin adres uçları birbirine aynı sırada bulunanlar birbirine karşı gelecek biçimde bağlanır. Yani A0' lar kendi aralarında A1' ler kendi aralarında birbirine bağlanırlar. Böylece A0' dan A12 ye kadar tüm adres yolları birbiri ile birleştirilerek ortak adres hatları oluşturulur.
- **2. Adım :** Veri yolunun oluşturulmasında ise farklı bir yol izlenir. Sekiz bellek kırmığı içinde her kırmığın veri yolu, bir veri hattına karşı düşürülür. Böylece sekiz bellek kırmığı sekiz veri hattına bağlanmış olur.
- **3. Adım :** Tüm belleklerin Oku/Yaz girişleri ve kırmık seçicileri kendi aralarında bağlanarak tek bir giriş haline getirilir.
- Enine genişletme yoluyla elde edilmiş bellekler üst üste eklenerek boyuna da büyütülebilirler.

Enine Bellek Geniřletme -II

