BİLGİSAYAR MİMARİSİ VE ORGANİZASYONU

Ders Kodu : BM303
L+T+P : 0-3-1
AKTS : 4

Dersi Veren :
Prof. Dr. Ahmet Akbaş

Mühendislik Fakültesi
Bilgisayar Müh. Bölümü

BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

2025-26 GYY DERSİ

BM303

BİLGİSAYAR MİMARİSİ ve ORGANİZASYONU

Prof. Dr. A. Akbaş

Ders İçeriği :

- Temel bellek devreleri
- Ana bellek organizasyonu
- Bellek teknolojisi
- Doğrudan bellek erişimi
- Etkin bellek erişim süresini azaltan önbellek
- Ana belleğin görünür boyutunu artıran sanal bellek
- İkincil depolama için kullanılan manyetik ve optik diskler

11.nci HAFTA DERSİ

1- Bilgisayar Sistemlerine Giriş (Fonksiyonel Birimler, Sistem Mimarisi, Performans, Teknolojik Arka Plan);

2- Programcı Bakış Açısıyla bir Sistem Turu;

3- Bilgisayar Sistemlerinde Bilginin (information) Temsili ve Manüplasyonu;

4- Komut Seti Mimarisi, Programların Makine Düzeyinde Temsili-I (RISC ve CISC mimariler);

5- Komut Seti Mimarisi, Programların Makine Düzeyinde Temsili-II (IA-32 ve x86-64 mimarileri);

6- İşlemci ve Giriş/Çıkış Birimleri Arasında Veri Transferi (Program Kontrollü G/Ç, Kesmeli G/Ç);

7- Yazılım (Program Hazırlama ve Çalıştırma Süreçleri, C ve Assembly Dilleri Arasındaki Etkileşim, İşletim Sistemleri);

8- İşlemci (CPU) Birimi (Fonksiyonel Birimler, Komut Getirme vr Yürütme Adımları, Kontrol Sinyalleri, HDL);

9- Program Performansının Optimizasyonu (Komut Düzeyinde Paralelleme, Eş-zamanlılık, Süperskaler İşlemciler);

10- Giriş/Çıkış Organizasyonu (Yol Yapısı, Yol Hakimliği, Arayüz Devreleri, Ara-bağlantı Standartları);

11- Bellek Sistemi (Temel Bellek Devreleri, Ana Bellek Organizasyonu, Bellek Teknolojileri);

12- Aritmetik (Tam Sayı ve Kayan Noktalı Sayılar için Aritmetik İşlem Devreleri);

13- Paralel Çalışma ve Performans (Çoklu İş Parçaları, Vektörel Süreçler, Paylaşımlı Bellekli Çoklu İşlemciler);

14- ARM İşlemcili SoC Örnekleri.

BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

2025-26 GYY DERSİ
BM303
BİLGİSAYAR MİMARİSİ ve ORGANİZASYONU
Prof. Dr. A. Akbaş

ENG. TERMINOLOGY & ACRONYMS

TEMEL KAVRAMLAR

Programlar ve üzerinde çalıştıkları veriler bilgisayarın belleğinde tutulur. Bu derste, belleğin nasıl çalıştığına bakacağız.

Programların yürütme hızı büyük ölçüde işlemci ile bellek arasındaki komut ve veri aktarma hızına bağlıdır. Ayrıca, büyük miktarda veri işleyen programlar için yeterli bellek kapasitesi de sağlanmalıdır.

Buna göre, belleğin hızlı, büyük ve ucuz olması beklenir. Ancak, bu üç gereksinimi aynı anda karşılamak mümkün değildir. Artan hız ve boyut, maliyet artışını gerektirir. Belleğin etkin hızını ve boyutunu iyileştirirken maliyeti makul seviyede tutan yapılar geliştirilmelidir.

Bir bilgisayarın belleği, **önbellek**, **ana bellek** ve **ikincil depolama birimleri**nin hiyerarşik düzeniyle oluşur. Bu derste, bu birimleri uygulamak için yaygın olarak kullanılan bileşenleri ve organizasyonları ele alacağız. Bu kapsamda, **disk** gibi bir **G/Ç** cihazı ile ana bellek arasında işlemcinin minimum düzeyde katılımıyla veri aktarmak için kullanılan Doğrudan Bellek Erişimi (**DMA**), bellek verilerine erişim sürelerinin **önbellek**ler aracılığıyla nasıl azaltılabileceği konuları yer almaktadır.

Son olarak, belleğin etkin boyutunu artırmak için ikincil depolama cihazlarının büyük depolama kapasitesinden yararlanan **sanal bellek** kavramını ele alacağız. Bazı temel kavramları sunarak başlayalım:

Herhangi bir bilgisayarda kullanılabilecek belleğin maksimum boyutu adresleme haritası (şeması) ile belirlenir. Örneğin, **16 bit** adresler üreten bir bilgisayar $\mathbf{2^{16} = 64K}$ (kilo) bellek konumuna kadar adresleme kapasitesine sahiptir.

Komutları **32 bit adres**ler üreten makineler $\mathbf{2^{32}=4G}$ (giga) konuma kadar bellek kullanabilirken, **64 bit adres**lere sahip makineler $\mathbf{2^{64}=16E}$ **(exa)** ya da $\mathbf{16 \times 10^{18}}$ konuma kadar erişebilir. Konum sayısı bilgisayarın adres alanının boyutunu temsil eder.

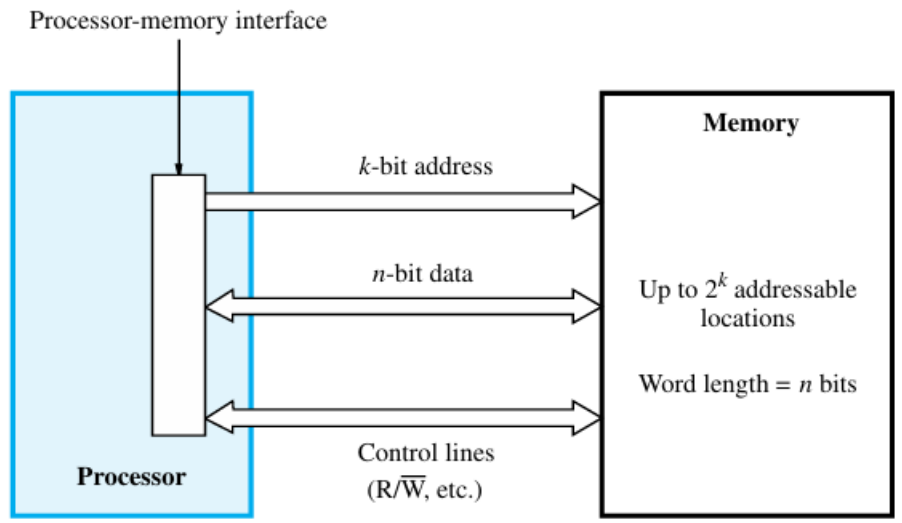
Bellek genellikle sözcük uzunluğundaki miktarlarda veri depolamak ve almak için tasarlanmıştır. Örneğin, komutları **32 bit** adresler üreten bayt adreslenebilir bir bilgisayar sisteminde işlemciden bellek birimine **32 bit**lik bir adres gönderildiğinde, yüksek değerli **30 bit** hangi sözcüğe erişileceğini belirler. Bayt boyutlu bir veri belirtilirse, adresin düşük sıralı **2 bit**i etkin bayt konumunu belirtir.

İşlemci ile bellek arasındaki bağlantı, **adres**, **veri** ve **kontrol** hatlarıyla sağlanır.

TEMEL KAVRAMLAR

İşlemci, adres hatlarını veri aktarım işleminde yer alan bellek konumunu belirtmek için, veri hatlarını da veriyi aktarmak için kullanır. Kontrol hatları, bir Okuma veya Yazma işlemini, aktarılan bir sözcüğün bayt ya da başka bir boyutta olduğunu kodlayan bilgileri ve gerekli zamanlama bilgileri ile beraber bellek tarafından istenen işlemin ne zaman tamamladığını belirtmek için kullanılır.

İşlemci-bellek arayüzü belleğin cevabını aldığında, istenen bellek işleminin tamamlandığını belirten idahili kontrol sinyali üretilir. Onaylandığında, işlemci yürütme dizisindeki bir sonraki adıma geçer.



Ref. [1]

Bellek birimlerinin hızının bir ölçüsü, bir veri sözcüğünü aktarma işleminin başlatılması ile sonlanması arasında geçen süredir. Buna **bellek erişim süresi** denir. Bir diğer ölçü ise, iki ardışık bellek işleminin başlatılması arasında gereken minimum zaman gecikmesi olan **bellek çevrim süresi**dir (örneğin iki Okuma işlemi arasındaki süre). Çevrim süresi, bellek biriminin uygulama ayrıntılarına bağlı olarak genellikle erişim süresinden biraz daha uzundur.

Bir bellek birimi, herhangi bir konuma erişim süresi, konumun adresinden bağımsız ise **rastgele erişimli bellek** (**RAM**) olarak adlandırılır. Bu tür bellek birimleri manyetik ve optik diskler gibi seri veya kısmen seri erişim depolama cihazlarından ayrılır. İkincil cihazların erişim süresi, verilerin adresine veya konumuna bağlıdır.

Bilgisayar bellekleri yarı iletken teknolojisi ile üretilir. Bu tür belleklerin iç yapısı ve çalışması ile ilgili temel kavramlar ve belleğin etkin hızı ve boyut artırma yöntemleri aşağıda sunulmuştur.

BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

2025-26 GYY DERSİ
BM303
BİLGİSAYAR MİMARİSİ ve ORGANİZASYONU
Prof. Dr. A. Akbaş

ENG. TERMINOLOGY & ACRONYMS

TEMEL KAVRAMLAR

Önbellek ve Sanal Bellek :

Bir bilgisayar sisteminin işlemci birimi, komutları ve verileri ana bellekten alınabilecekleri hızdan daha yüksek hızla işleyebilir. Bu nedenle, bellek erişim süresi sistemin bir darboğazıdır. Bu süreyi azaltmanın bir yolu **önbellek** kullanmaktır. Önbellek, daha büyük ve daha yavaş olan ana bellek ile işlemci arasındaki küçük ve hızlı bir bellektir. Bir programın o an etkin olan kısımları ve verilerini tutar.

Sanal bellek, bellek organizasyonuyla ilgili diğer bir önemli kavramdır. Bu teknikle, bir programın yalnızca etkin kısımları ana bellekte saklanır ve geri kalanı çok daha büyük ikincil depolama cihazında saklanır. Programın segmentleri, uygulama programı için şeffaf bir şekilde ana bellek ile ikincil depolama cihazı arasında ileri geri aktarılır. Sonuç olarak, uygulama programı bilgisayarın **fiziksel ana belleğ**inden çok daha **büyük** bir bellek görür.

Blok Aktarımları :

Yukarıdaki tartışma, verilerin ana bellek ile önbellek arasında ve ana bellek ile disk arasında sıkça hareket ettiğini göstermektedir. Bu aktarımlar her seferinde tek sözcük yerine onlarca, yüzlerce veya binlerce sözcük içeren bitişik bloklar halinde yapılır. Ana bellek ile grafik ekran veya Ethernet arayüzü gibi yüksek hızlı cihazlar arasındaki veri aktarımları da büyük veri blokları içerir.

Bu nedenle, ana belleğin performansı için kritik bir parametre, veri bloklarını yüksek hızda okuma veya yazma yeteneğidir. Bu, bellek teknolojisini ve bellek sisteminin organizasyonunu tartışırken tekrar tekrar karşılaşacağımız önemli bir hususdur.

Çeşitli hızlarda yarı-iletken rastgele erişimli bellekler (**RAM**'ler) mevcuttur. Döngü süreleri **100 ns** ila **10 ns**'nin altındaki sürelere kadar değişebilir. Bu derste, bu belleklerin temel özelliklerini ele alacağız. Bellek hücrelerinin bir çip içinde düzenlenme biçimini tanıtarak başlayalım:

Bellek Yongalarının İç Düzeni :

Bellek hücreleri genellikle her hücrenin bir bit bilgi depolayabildiği bir dizi biçiminde düzenlenir. Olası bir düzende her hücre satırı bir bellek sözcüğünü oluşturur. Bir satırın tüm hücreleri, çip üzerindeki adres kod çözücüsü tarafından yönlendirilen ve sözcük satırı olarak adlandırılan ortak bir hatta bağlanır. Her sütundaki hücreler iki bit hattıyla bir **Algıla/Yaz** (**Sense/Write**) devresine bağlanır.

BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

2025-26 GYY DERSİ
BM303
BİLGİSAYAR MİMARİSİ ve ORGANİZASYONU
Prof. Dr. A. Akbaş

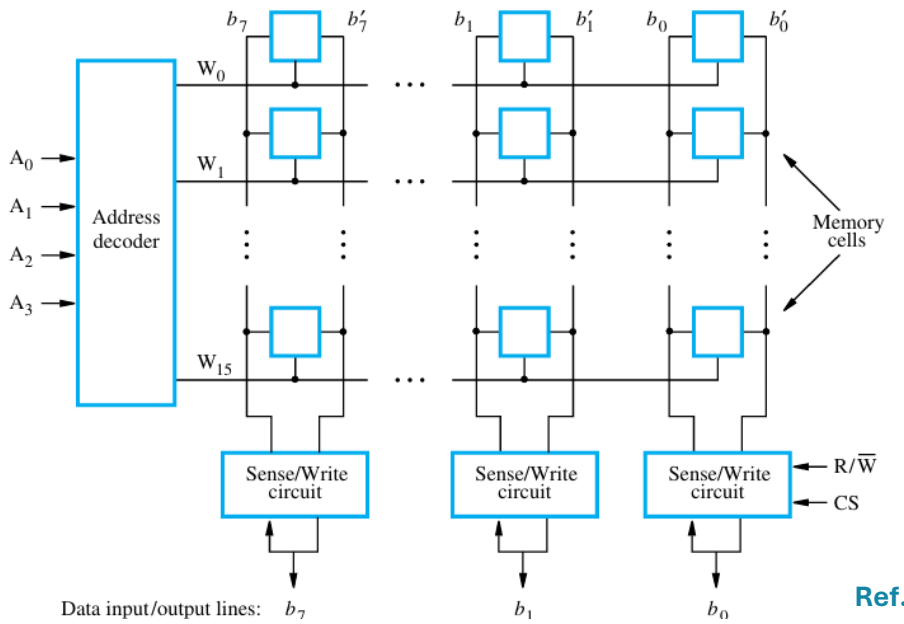
ENG. TERMINOLOGY & ACRONYMS

YARI-İLETKEN RAM BELLEKLER

Bellek Yongalarının İç Düzeni :

Algıla/Yaz devreleri çipin veri giriş/çıkış hatlarına bağlanır. Okuma işlemi esnasında, bu devreler bir sözcük satırı tarafından seçilen hücrelerde depolanan bilgileri algılar veya okur ve bu bilgileri çıkış veri satırlarına yerleştirir. Yazma işlemi esnasında, **Algıla/Yaz** devreleri giriş verilerini alır ve bunları seçili kelimenin hücrelerine depolar.

Şekil, her biri **8 bit**lik **16 sözcük**ten oluşan çok küçük bir bellek devresi örneğidir. Bu **16 × 8** bir bellek yapısıdır.



Ref. [1]

Her **Algıla/Yaz** devresinin veri girişi ve veri çıkışı, bir bilgisayarın veri hatlarına bağlanabilen tek bir çift yönlü veri hattına bağlanır. İki kontrol hattı, **R/W** ve **CS** sağlanır. **R/W** (Oku/Yaz) girişi, gerekli işlemi belirtir ve **CS** (Çip Seçimi) girişi, çok çipli bir bellek sisteminde belirli bir çipi seçer. Şekildeki bellek devresi **128** bit depolar ve **adres**, **veri** ve **kontrol** hatları için **14** harici bağlantı gerektirir. Ayrıca güç kaynağı ve toprak bağlantıları için iki hat daha gerekir.

Şimdi **1K** (**1024**) bellek hücresi olan bir devreyi ele alalım. Bu devre, toplam **19** harici bağlantı gerektiren **128 × 8** bellek olarak düzenlenebilir. Alternatif olarak, aynı sayıda hücre **1K×1** biçiminde düzenlenebilir. Bu durumda, **10** bitlik bir adrese ihtiyaç duyulur, ancak yalnızca bir veri hattı vardır ve bu da **15** harici bağlantıyla sonuçlanır.

Bir sonraki şekil böyle bir düzenlemeyi göstermektedir. Gerekli **10** bitlik adres, hücre dizisi için satır ve sütun adreslerini oluşturmak üzere her biri **5** bitlik iki gruba ayrılır. Bir satır adresi, hepsine paralel olarak erişilen **32** hücrelik bir satırı seçer. Ancak, sütun adresine göre bu hücrelerden yalnızca biri harici veri hattına bağlanır.

BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

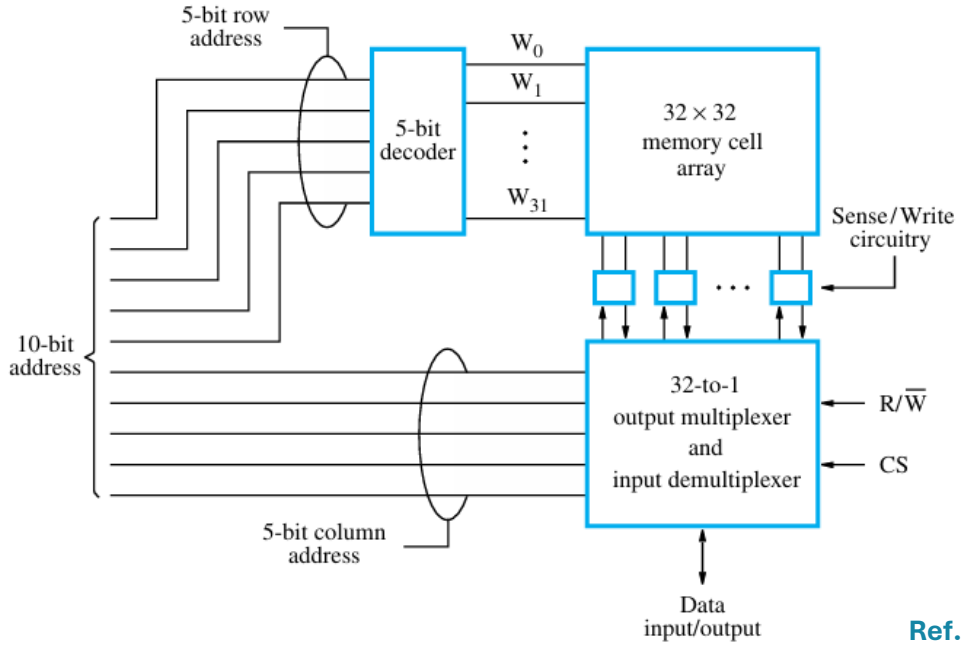
2025-26 GYY DERSİ
BM303
BİLGİSAYAR MİMARİSİ ve ORGANİZASYONU
Prof. Dr. A. Akbaş

ENG. TERMINOLOGY & ACRONYMS

YARI-İLETKEN RAM BELLEKLER

Bellek Yongalarının İç Düzeni:

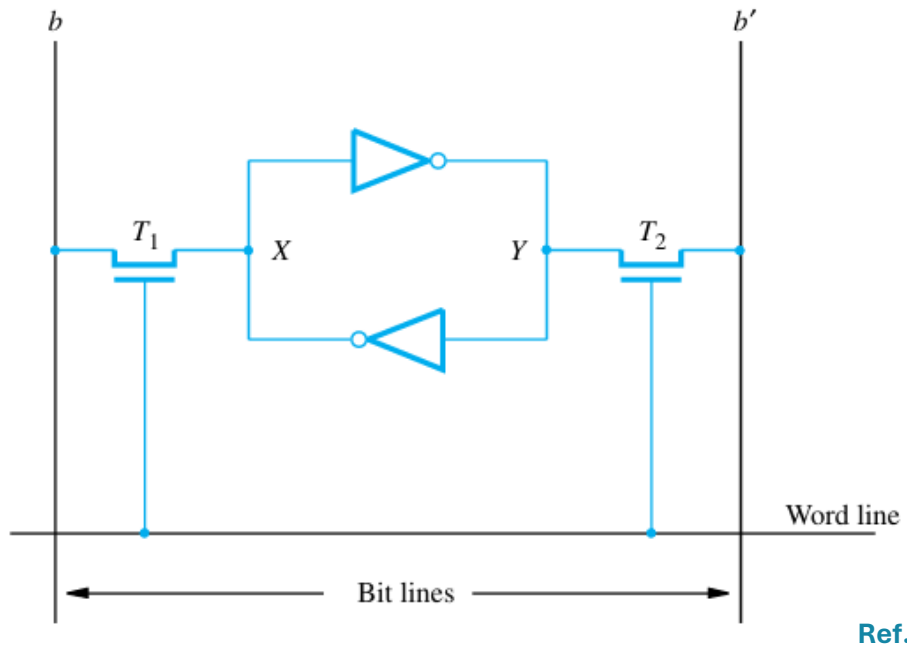
Ticari olarak satılan bellek yongaları, bu şekillerde gösterilen örneklerden çok daha fazla sayıda bellek hücresi içerir. Büyük çipler esasen Şekildeki ile aynı organizasyona sahiptir, ancak daha büyük bir bellek hücre dizisi kullanır ve daha fazla harici bağlantıya sahiptir. Örneğin, **1G bit**lik bir çip **256M × 4** organizasyona sahip olabilir, bu durumda **28 bit**lik bir adrese ihtiyaç duyulur ve çiple bellek arasında **4 bit** aktarılır.



Ref. [1]

Statik Bellekler :

Güç uygulandığı sürece durumlarını koruyabilen devrelerden oluşan belleklerdir. Şekilde, **statik** bir **RAM** (**SRAM**) hücresinin nasıl uygulanabileceği görülüyor. İki invertör, bir kilit (latch) oluşturmak için çapraz bağlanır. Kilit, **T1** ve **T2** transistörleri tarafından iki bit hattına bağlanır. Bu transistörler, kelime hattının kontrolü altında açılıp kapatılabilen anahtarlar gibi davranır.



Ref. [1]

BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

2025-26 GYY DERSİ
BM303
BİLGİSAYAR MİMARİSİ ve ORGANİZASYONU
Prof. Dr. A. Akbaş

ENG. TERMINOLOGY & ACRONYMS

YARI-İLETKEN RAM BELLEKLER

Statik Bellekler :

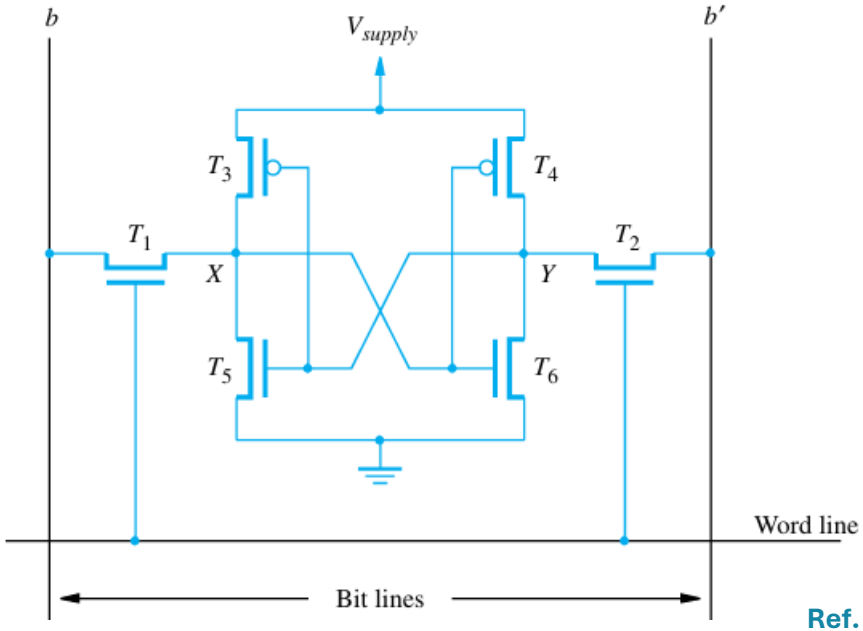
Sözcük hattı Toprak (ground) seviyesinde olduğunda, transistörler kapatılır ve kilit durumunu korur. Örneğin, **X** noktası mantıksal **1** ve **Y** noktası mantıksal **0** değerinde ise, sözcük hattındaki sinyal Toprak seviyesinde olduğu sürece bu durum korunur. Bu durumun **1** değerini temsil ettiğini varsayalım.

Okuma İşlemi : **SRAM** hücresinin durumunu okumak için, **T1** ve **T2** anahtarlarını kapatmak üzere sözcük hattı etkinleştirilir. Hücre **1** durumundaysa, bit hattı **b** yüksek ve bit hattı **b'** düşük seviyededir. Hücre **0** durumundaysa bunun tersi geçerlidir. Bu nedenle, **b** ve **b'** her zaman birbirinin tümleyenidir. İki bit hattının sonundaki **Algıla/Yaz** devresi durumlarını izler ve buna göre karşılık gelen çıktıyı ayarlar.

Yazma İşlemi : Bir Yazma işlemi esnasında **Algıla/Yaz** devresi durumlarını algılamak yerine bit hatları **b** ve **b'**yi çalıştırır. Uygun değeri bit hattı **b** ve tümleyenini **b'**ye yerleştirir ve sözcük hattını etkinleştirir. Bu, hücreyi sözcük hattı devre dışı bırakıldığında hücrenin koruduğu karşılık gelen duruma zorlar.

CMOS Hücresi :

Statik bellek hücresinin **CMOS** ile gerçekleştirilen şeklinde (**T3-T5**) ve (**T4-T6**) transistör çiftleri kilitin invertörleridir. Hücre durumu, öncekindeki gibi okunur veya yazılır. Örneğin, durum **1**'de, **X** noktasındaki voltaj, **T3** ve **T6** transistörleri açıkken, **T4** ve **T5** kapalıyken yüksek tutulur. **T1** ve **T2** açıksa, bit hatları **b** ve **b'** sırasıyla yüksek ve düşük seviyelerde olacaktır.



Ref. [1]

BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

2025-26 GYY DERSİ
BM303
BİLGİSAYAR MİMARİSİ ve ORGANİZASYONU
Prof. Dr. A. Akbaş

ENG. TERMINOLOGY & ACRONYMS

YARI-İLETKEN RAM BELLEKLER

CMOS Hücresi :

Hücrenin durumunu koruması için sürekli güce ihtiyaç vardır. Güç kesilirse, hücrenin içerikleri kaybolur. Güç geri geldiğinde, kilit kararlı bir duruma yerleşir, ancak hücrenin kesintiden önceki durumuyla aynı olması gerekmez. Bu nedenle, **SRAM**'lerin **uçucu bellekler** olarak bilinir, güç kesildiğinde içerikleri kaybolur.

CMOS SRAM'lerin en büyük avantajlarından biri, yalnızca hücreye erişildiğinde akım aktığı için çok düşük güç tüketmeleridir. Aksi takdirde, her invertördeki **T1**, **T2** ve bir transistör kapatılır ve besleme voltajı ile toprak arasında sürekli bir elektriksel yol olmadığından emin olunur.

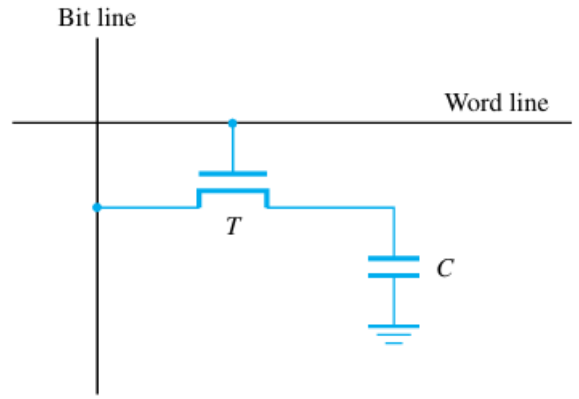
Statik RAM'lere çok hızlı bir şekilde erişilebilir. Ticari olarak satılan çiplerde birkaç nanosaniye mertebesinde erişim süreleri bulunur. **SRAM**'ler, hızın kritik öneme sahip olduğu uygulamalarda kullanılır.

Statik RAM'ler hızlıdır, ancak her hücresi birkaç transistörle oluşturulur. Daha ucuz ve daha yüksek yoğunluklu **RAM**'ler için daha basit hücreler gereklidir. Ancak, bu hücreler, sıkça Okunup Yazılmadıkça durumlarını uzun süre koruyamazlar.

Dinamik Bellekler :

Bu tür hücreleri kullanan belleklere **dinamik RAM** (**DRAM**) denir. Bilgi, dinamik bir bellek hücresinde bir kapasitör üzerinde yük olarak depolanır, ancak bu yük yalnızca onlarca milisaniye boyunca korunabilir. Hücrenin bilgiyi daha uzun süre depolaması için, içeriği kapasitör yükünü tam değerine geri yükleyerek periyodik olarak yenilenmelidir. Bu, hücre içeriği okunduğu veya içine yeni bilgi yazıldığında gerçekleşir.

Bir **C** kapasitörü ve bir transistör **T**'den oluşan dinamik bir bellek hücresi Şekilde gösterilmiştir. Bu hücrede bilgi depolamak için, transistör **T** açılır ve bit hattına uygun bir voltaj uygulanır. Bu, kapasitöre yük depolar.



Ref. [1]

BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

2025-26 GYY DERSİ
BM303
BİLGİSAYAR MİMARİSİ ve ORGANİZASYONU
Prof. Dr. A. Akbaş

ENG. TERMINOLOGY & ACRONYMS

YARI-İLETKEN RAM BELLEKLER

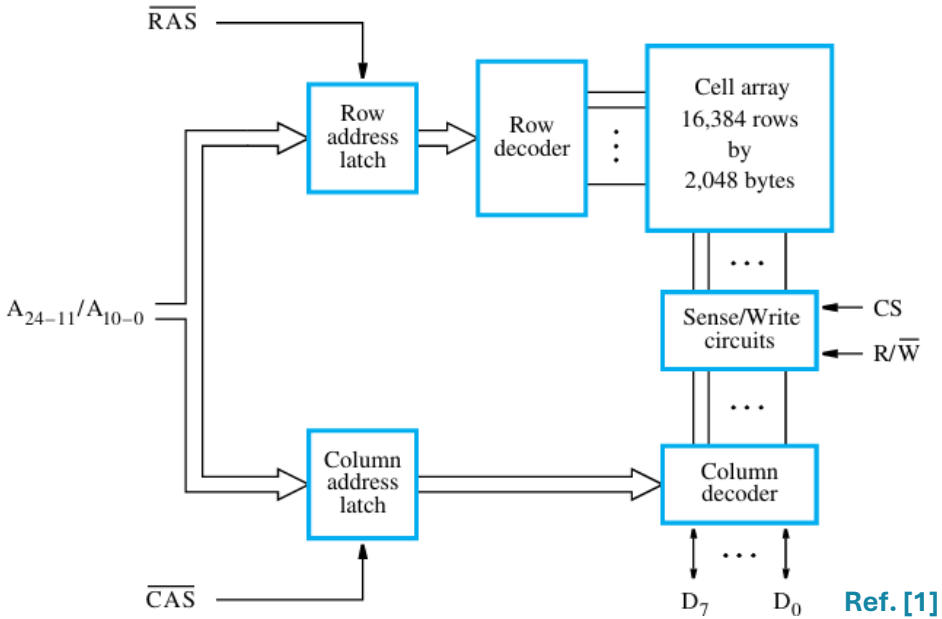
Dinamik Bellekler :
Transistör kapatıldıktan sonra, kondansatör deşarj olmaya başlar. Bunun nedeni, transistörün kapatıldıktan sonra pikoamper cinsinden ölçülen küçük miktarda akımı iletmeye devam etmesidir. Bu nedenle, hücrede depolanan bilgi, yalnızca kondansatördeki yük belirli bir eşik değerinin altına düşmeden önce okunursa doğru şekilde alınabilir.

Bir Okuma işlemi esnasında, seçilen bir hücredeki transistör açılır. Bit hattına bağlı bir algılama amplifikatörü, kondansatörde depolanan yükün eşik değerinin üstünde mi yoksa altında mı olduğunu algılar.

Yük eşiğin üstündeyse, algılama amplifikatörü bit hattını mantık değeri **1**'i temsil eden tam voltaja sürer. Sonuç olarak, kondansatör mantık değeri **1**'e karşılık gelen tam yüke yeniden şarj edilir. Algılama amplifikatörü, kondansatördeki yükün eşik değerinin altında olduğunu algılarsa, kondansatörü tamamen boşaltmak için bit hattını toprak seviyesine çeker.

Satır sözcüğü bir satırdaki tüm hücreler için ortak olduğundan, seçili satırdaki tüm hücreler aynı anda okunur ve yenilenir.

32M×8 olarak yapılandırılmış **256 Megabit** bir **DRAM** çipinde hücreler **16K × 16K** dizi biçiminde düzenlenmiştir. Her satırdaki **16.384** hücre **2.048 8**'li gruba bölünerek **2.048 bayt** veri oluşturur. Bu nedenle, bir satırı seçmek için **14** adres biti ve seçili satırdaki **8** bitlik bir grubu belirtmek için **11** bit daha gerekir. Bu bellekteki bir bayta erişmek için toplamda **25** bitlik bir adres gerekir. Adresin yüksek sıralı **14** biti ve düşük sıralı **11** biti, sırasıyla bir baytın satır ve sütun adresleridir.



Ref. [1]

BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

2025-26 GYY DERSİ
BM303
BİLGİSAYAR MİMARİSİ ve ORGANİZASYONU
Prof. Dr. A. Akbaş

ENG. TERMINOLOGY & ACRONYMS

YARI-İLETKEN RAM BELLEKLER

Dinamik Bellekler :
Harici bağlantılar için gereken pin sayısını azaltmak için, satır ve sütun adresleri **14** pin üzerinde çoklanır. Okuma veya Yazma işlemi sırasında, önce satır adresi uygulanır. **Satır Adresi Strobu** (**RAS**) adı verilen bir giriş kontrol hattındaki sinyal darbesine cevap olarak satır adresi kilidine yüklenir. Bu, seçili satırdaki tüm hücrelerin okunup yenilendiği bir Okuma işleminin başlatılmasına neden olur.

Satır adresi yüklendikten kısa bir süre sonra, sütun adresi adres pinlerine uygulanır ve **Sütun Adresi Strobu** (**CAS**) adı verilen ikinci bir kontrol hattının kontrolü altında sütun adresi kilidine yüklenir. Bu kilitteki bilgiler çözülür ve uygun **8 Algıla/Yaz** devresi grubu seçilir. **R/W** kontrol sinyali bir Okuma işlemi gösteriyorsa, seçili devrelerin çıkış değerleri veri hatları **D7–0**'a aktarılır.

Bir Yazma işlemi için, **D7–0** hatlarındaki bilgiler seçili devrelere aktarılır ve ardından ilgili **8** sütundaki seçili hücrelerin içeriklerinin üzerine yazmak için kullanılır. Ticari **DRAM** yongalarında, **RAS** ve **CAS** kontrol sinyallerinin düşük olduğunda etkin olduğu belirtilmelidir. Bu nedenle, bu sinyaller yüksekten düşüğe değiştiğinde adresler kilitlenir.

RAS ve **CAS** sinyalleri, bu gerçeği belirtmek için diyagramlarda üstü çizgili olarak gösterilir. **DRAM**'ın çalışma zamanlaması bu sinyaller tarafından kontrol edilir. Bu sinyaller, işlemci bir Okuma veya Yazma komutu verdiğinde çipin dışındaki bir bellek denetleyici devresi tarafından üretilir. Bir Okuma işlemi sırasında, çıktı verileri belleğin erişim süresine eşdeğer bir gecikmeden sonra işlemciye aktarılır. Bu tür belleklere **asenkron DRAM** denir. Bellek denetleyicisi ayrıca, daha sonra açıklayacağımız gibi, bellek yongalarında depolanan verileri yenilemekten de sorumludur.

Hızlı Sayfa Modu : Şekildeki **DRAM**'a erişildiğinde, seçili satırdaki **16.384** hücrenin tümünün içerikleri algılanır, ancak veri hatlarına yalnızca **8 bit** yerleştirilir, **D7–0**. Bu bayt, sütun adresi, bitler **A10–0** tarafından seçilir. Devreye yapılan basit bir ekleme, satırı yeniden seçmek zorunda kalmadan aynı satırdaki diğer baytlara erişmeyi mümkün kılar. Her algılama yükselteci aynı zamanda bir kilit görevi görür. Bir satır adresi uygulandığında, seçili satırdaki tüm hücrelerin içerikleri ilgili kilitlere yüklenir.

YARI-İLETKEN RAM BELLEKLER

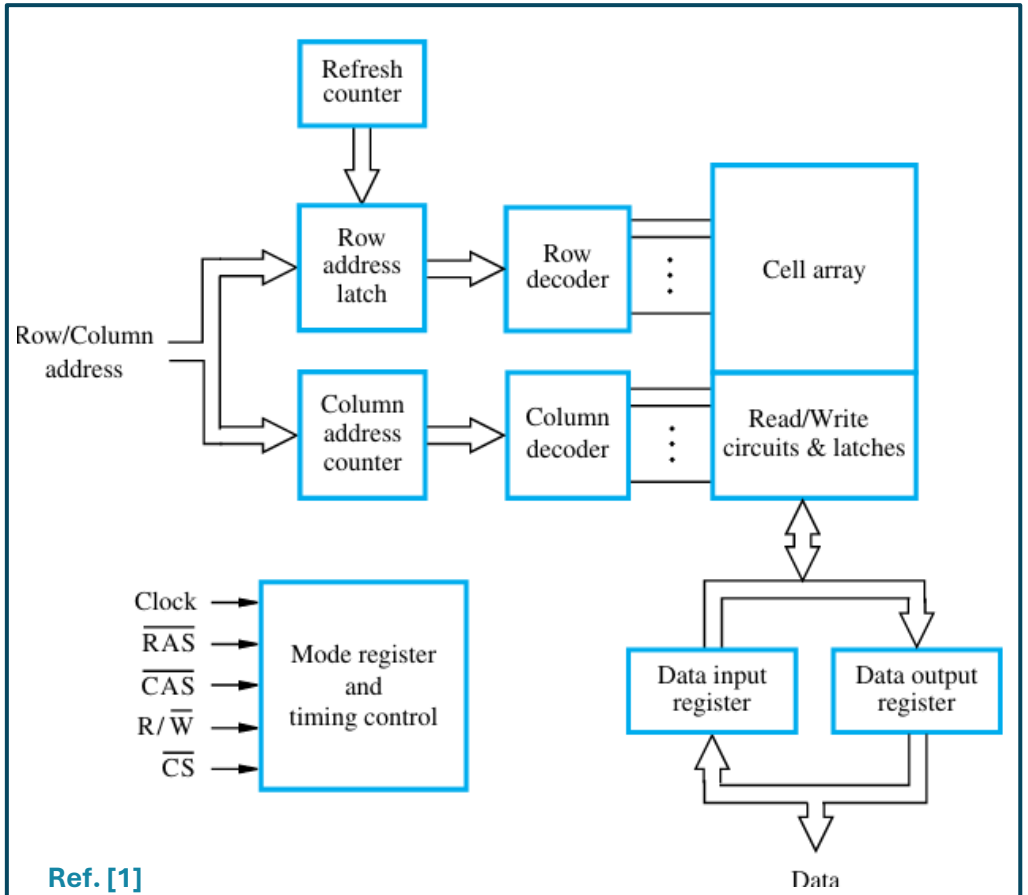
Hızlı Sayfa Modu: Daha sonra, farklı baytları veri hatlarına yerleştirmek için yalnızca farklı sütun adresleri uygulamak gerekir. Bu düzenleme çok kullanışlı bir özelliğe yol açar. Seçili satırdaki tüm baytlar, ardışık **CAS** sinyallerinin kontrolü altında ardışık bir sütun adresi dizisi uygulanarak sıralı bir şekilde aktarılabilir. Böylece, bir **veri bloğu**, rastgele adresleri içeren aktarımlar için elde edilebilenden çok daha hızlı aktarılabilir. Blok transfer kabiliyetine **hızlı sayfa modu** özelliği denir. (Büyük bir veri bloğuna genellikle **sayfa** olarak adlandırılır.)

Daha önce ana bellek işlemlerinin büyük çoğunluğunun blok transferlerini içerdiği belirtilmişti. Hızlı sayfa modunda elde edilebilen daha hızlı oran, **dinamik RAM**'leri bu ortama özellikle uygun hale getirir.

Senkron (Eşzamanlı) DRAM'ler

1990'ların başlarında, bellek teknolojisindeki gelişmeler, çalışması bir saat sinyaliyle senkronize edilen **DRAM**'lerin ortaya çıkmasına neden oldu. Bu tür bellekler, **senkron DRAM**'ler (**SDRAM**'ler) olarak bilinir. Yapıları Şekilde gösterilmiştir. Hücre dizisi, **asenkron DRAM**'lerdekiyle aynıdır. **SDRAM**, çipe birçok yararlı özellik katan kontrol devresini dahil eden bir saat sinyalinin kullanılmasıdır.

Örneğin, **SDRAM**'lerde, yenileme için seçilecek satırların adreslerini sağlayan bir yenileme sayacıyla birlikte yerleşik **yenileme devreleri** bulunur. Bu tür bellek çiplerinin dinamik yapısı kullanıcı için görünmezdir.



Ref. [1]

BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

2025-26 GYY DERSİ
BM303
BİLGİSAYAR MİMARİSİ ve ORGANİZASYONU
Prof. Dr. A. Akbaş

ENG. TERMINOLOGY & ACRONYMS

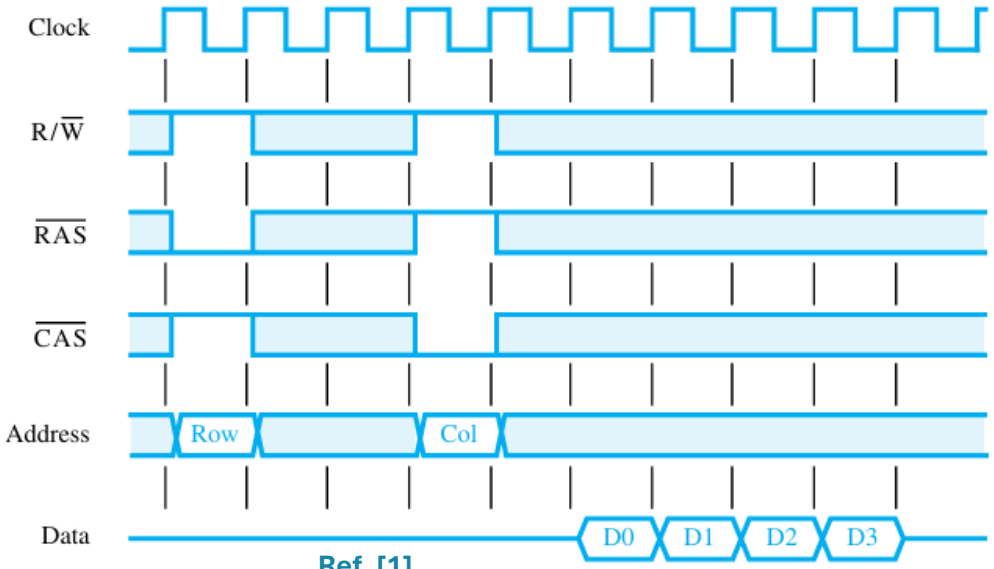
YARI-İLETKEN RAM BELLEKLER

Senkron DRAM'lerın adres ve veri bağlantıları, kaydediciler vasıtasıyla tamponlanabilir. Dahili olarak, **Sense/Write** amplifikatörleri, **asenkron DRAM**'larda olduğu gibi kilit olarak işlev görür. Bir Okuma işlemi, seçilen satırdaki tüm hücrelerin içeriklerini bu kilitlere yükler. Seçilen sütunun kilitlerindeki veriler veri kaydedicisine aktarılır ve böylece veri çıkış pinlerinde kullanılabilir hale gelir. Arabellek kaydedicileri, büyük veri bloklarını çok yüksek hızda aktarırken kullanışlıdır. Harici bağlantılar çipin dahili devrelerinden izole edilerek, kaydedici giriş veya çıkı için yeni bir erişim işlemi başlatılabilir.

SDRAM'ların, kontrol bilgilerini bir mod kaydedicisine yazarak seçilebilen birkaç farklı çalışma modu vardır. Örneğin, farklı uzunluklarda patlama işlemleri belirtilebilir. Ardışık sütunları seçmek için **CAS** hattında harici olarak üretilen darbeler sağlamak gerekli değildir. Gerekli kontrol sinyalleri, bir sütun sayacı ve saat sinyali ile dahili olarak üretilir. Yeni veriler, her saat darbesinin yükselen kenarında veri hatlarına yerleştirilir.

Şekil, uzunluğu **4** olan tipik bir patlama okumasının zamanlama diyagramını gösteriyor. Burada ilk olarak, satır adresi RAS sinyalinin kontrolü altında kilitlenir. Bellek, seçilen satırı etkinleştirmek için 5 veya 6 saat döngüsü (şekilde 2) alır.

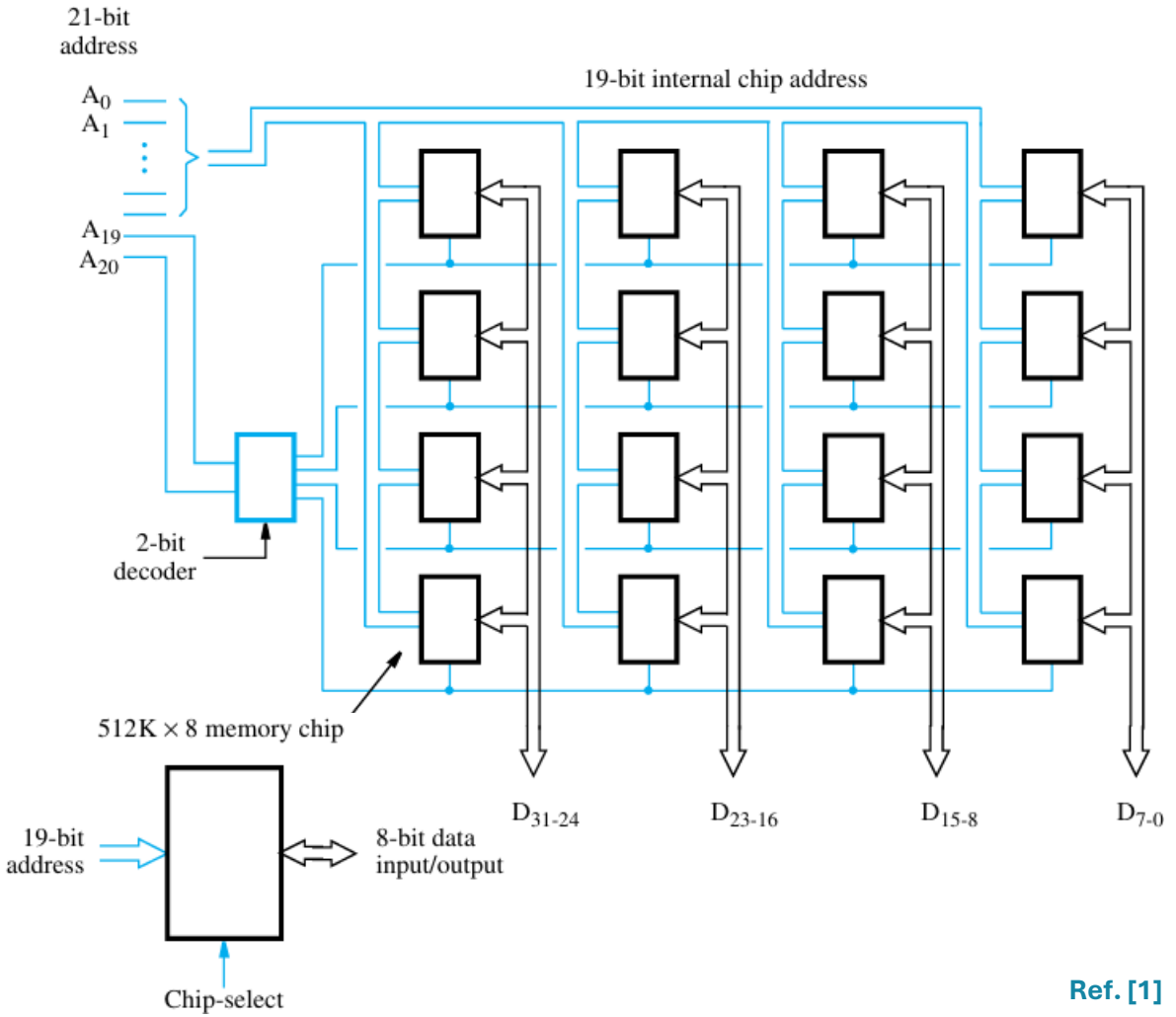
Ardından, sütun adresi CAS sinyalinin kontrolünde kilitlenir. Bir saat döngüsü gecikmeden sonra, ilk veri biti seti veri hatlarına yerleştirilir. **SDRAM**, seçilen satırdaki bir sonraki üç bit setine erişmek için sütun adresini otomatik olarak artırır ve bunlar bir sonraki **3** saat döngüsünde veri hatlarına yerleştirilir.
Senkron DRAM'ler, ihtiyaç duyulan tüm kontrol sinyalleri çipin içinde üretildiği için verileri çok yüksek bir hızda iletebilir. **1990**'larda **133 MHz**, günümüzde **1GHz**'i aşabilen saat hızlarıyla çalışmaktadır.



Ref. [1]

DAHA BÜYÜK KAPASİTELİ BELLEKLERİN YAPISI

512K x 8 bit statik bellek çiplerini kullanan **2M x 32 bit** bellek modülü



Ref. [1]

Salt Okunur Bellekler:

Hem statik hem de dinamik **RAM** yongaları uçucudur, yani yalnızca güç açıkken bilgileri tutarlar. Güç kapatıldığında depolanan bilgileri tutan bellek aygıtları gerektiren birçok uygulama vardır. Örneğin, işletim sistemini bir sabit diskten ana belleğe yükleme önyükleme sürecini başlatmak için kullanılacak küçük bir programı böyle bir belleğe depolamanın gerekliliğini açıklar.

Gömülü uygulamalar başka bir önemli örnektir. Birçok gömülü uygulama sabit disk kullanmaz ve yazılımlarını depolamak için uçucu olmayan bellekler gerektirir. Farklı türde uçucu olmayan bellekler geliştirilmiştir. Genellikle, içerikleri yukarıda tartışılan uçucu muadilleriyle aynı şekilde okunabilir. Ancak, bilgileri uçucu olmayan bir belleğe yerleştirmek için özel bir yazma işlemi gerekir.

Normal çalışması yalnızca depolanan verileri okumayı içerdiğinden, bu tür bir belleğe **salt okunur bellek** (**ROM**) denir.

Ref. [1]

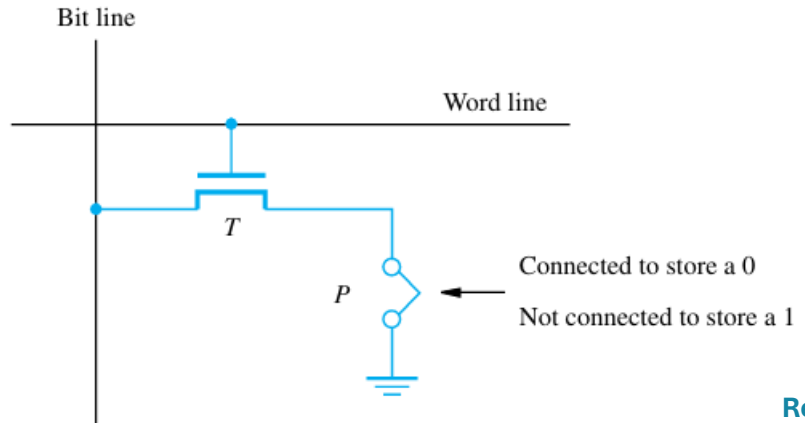
BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

2025-26 GYY DERSİ
BM303
BİLGİSAYAR MİMARİSİ ve ORGANİZASYONU
Prof. Dr. A. Akbaş

ENG. TERMINOLOGY & ACRONYMS

SALT OKUNUR BELLEKLER

ROM : Bir belleğe, üretim sırasında yalnızca bir kez bilgi yazılabildiğinde, salt okunur bellek veya **ROM** denir.



Ref. [1]

PROM : Bazı ROM tasarımları, verilerin kullanıcı tarafından yüklenmesine izin verir ve böylece **programlanabilir** bir **ROM** (**PROM**) sağlar. Programlanabilirlik, Şekildeki **P** noktasına bir sigorta yerleştirilerek elde edilir.

EPROM : Başka bir **ROM** çipi türü daha da yüksek bir kolaylık seviyesi sağlar. Depolanan verilerin silinmesine ve içine yeni verilerin yazılmasına olanak tanır.

EEPROM : Silinebilir **PROM**'un bir diğer türü elektrikle programlanabilir, silinebilir ve yeniden programlanabilir. Bu tür bir çipe elektrikle silinebilir PROM veya **EEPROM** denir. Silinmesi için çıkarılması gerekmez.

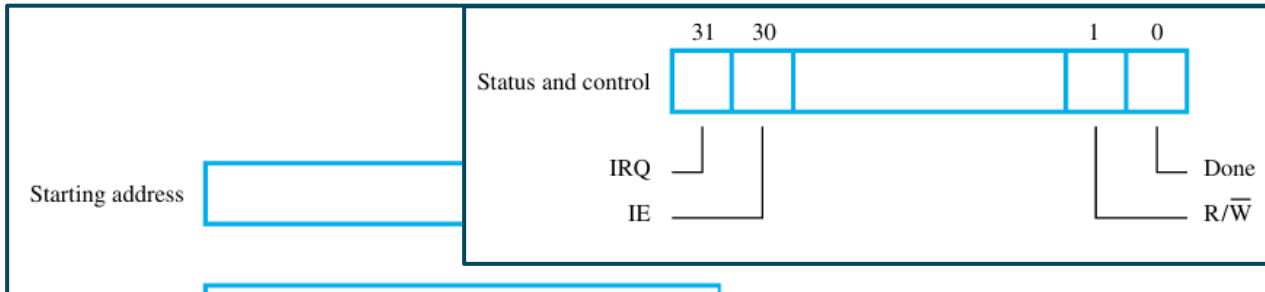
Flash Bellek : **EEPROM** teknolojisine benzer bir yaklaşım, flash bellek cihazlarının ortaya çıkmasına neden olmuştur. Bir flash hücre, tıpkı bir **EEPROM** hücresi gibi, sıkışmış yük tarafından kontrol edilen tek bir transistöre dayanır. Yine bir **EEPROM** gibi, tek bir hücrenin içeriğini okumak mümkündür. Temel fark, bir flash cihazda, yalnızca tüm bir hücre bloğunun yazılmasının mümkün olmasıdır. Yazmadan önce, bloğun önceki içerikleri silinir. Flash cihazlar daha yüksek yoğunluğa sahiptir, bu da daha yüksek kapasiteye ve bit başına daha düşük maliyete yol açar. Tek bir güç kaynağı voltajı gerektirirler ve çalışmalarında daha az güç tüketirler.

Flash Kartlar : Daha büyük bir bellek modülü küçük bir karta flash çipleri takmaktır. Bu tür kartların, çeşitli ürünlerde kullanılabilir hale getiren standart bir arayüzü vardır. Bir kart, kolayca erişilebilen bir yuvaya takılır.

DOĞRUDAN BELLEK ERİŞİMİ

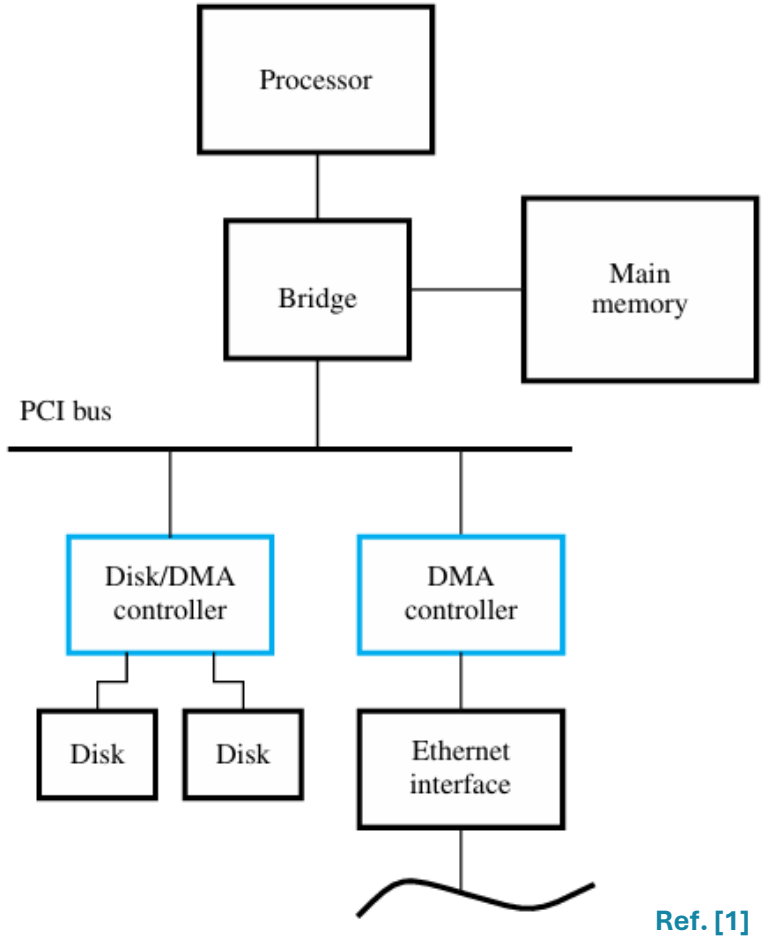
Flash Kartlar : USB arayüzlü flash kartlar yaygın olarak kullanılır ve genellikle bellek anahtarları olarak bilinir. Çeşitli bellek boyutları vardır. Daha büyük kartlar **32 GB**'a kadar tutabilir. Bir dakikalık müzik, **MP3** kodlama biçimi kullanılarak yaklaşık **1 MB** belleğe depolanabilir. Bu nedenle, **32 GB**'lık bir flash kart yaklaşık **500 saatlik müzik** depolayabilir.

Flash Sürücüler : Daha büyük flash bellek modülleri, sabit **disk sürücüleri**nin yerini almak üzere geliştirilmiştir ve bu nedenle **flash sürücüler** olarak adlandırılırlar. Bunlar, standart disk sürücü yuvalarına takılabilecekleri noktaya kadar sabit diskleri tam olarak taklit edecek şekilde tasarlanmıştır. Ancak, flash sürücülerin depolama kapasitesi önemli ölçüde daha düşüktür: 64, 128, 256 GB mertebelerinde. Buna karşılık, sabit disklerin kapasiteleri bir **terabayt**ı aşar. Ayrıca, disk sürücülerin bit başına maliyeti daha azdır.



Ref. [1]

Bir bilgisayar sisteminde DMA kontrolörlerinin kullanımı ve tipik bir DMA'nın kaydedicileri



Ref. [1]

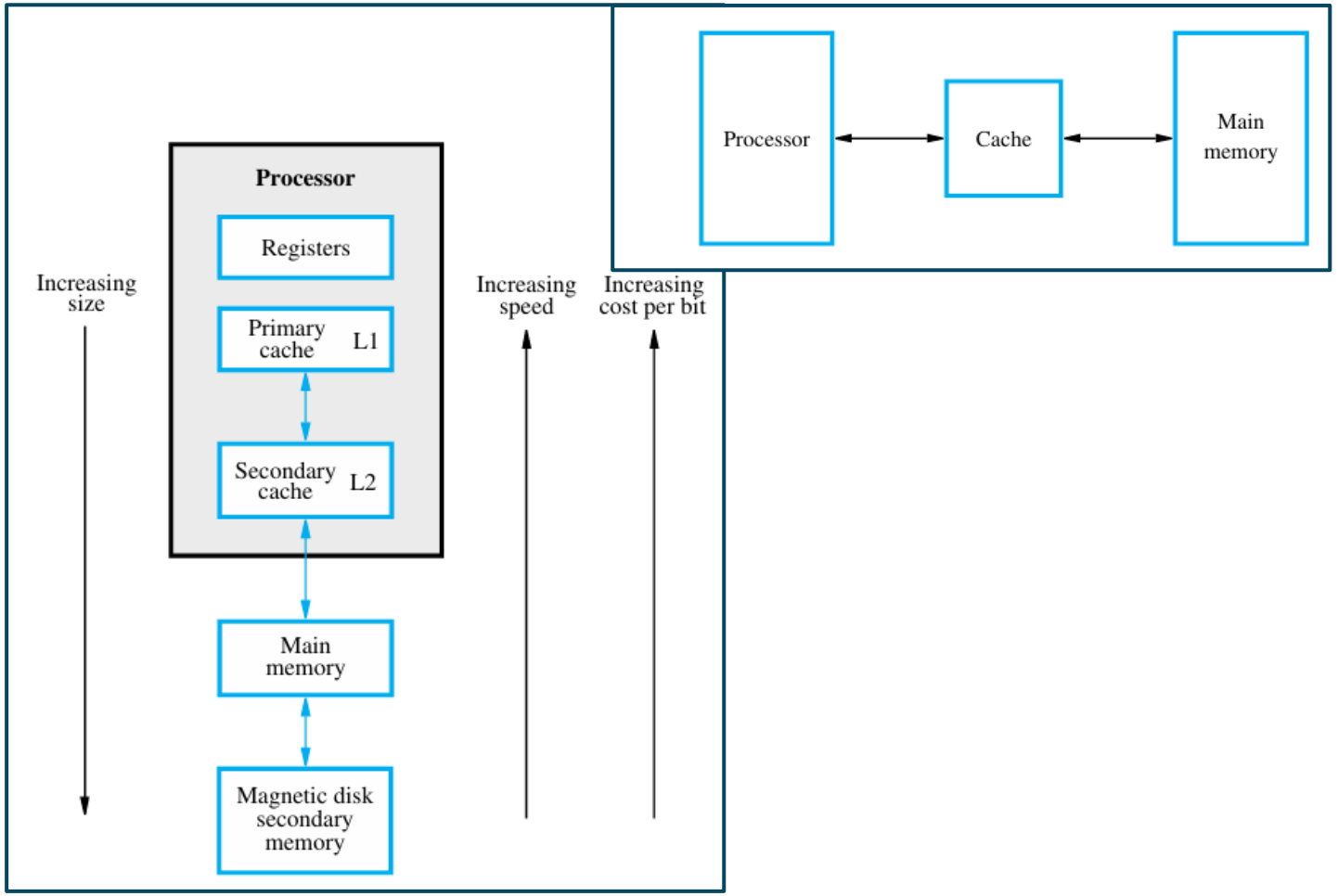
BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

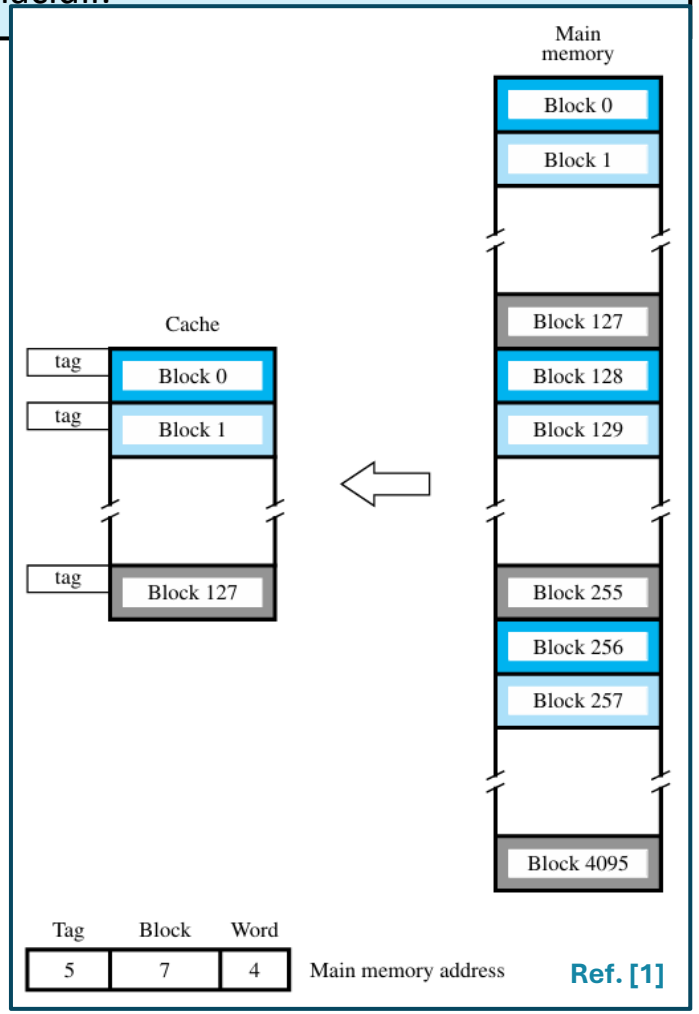
2025-26 GYY DERSİ
BM303
BİLGİSAYAR MİMARİSİ ve ORGANİZASYONU
Prof. Dr. A. Akbaş

ENG. TERMINOLOGY & ACRONYMS

BELLEK HİYERARŞİSİ, ÖNBELLEKLER ve BELLEK HARİTALAMA FONKSİYONLARI

Program yürütme sürecinde bellek erişiminin hızı son derece önemlidir. Şekildeki hiyerarşik bellek sisteminin çalışmasını yönetmenin anahtarı, kullanılmak üzere olan komutları ve verileri işlemciye mümkün olduğunca yakın bir yere getirmektir. Bu, önbellek belleklerinin kullanılmasının temel amacıdır.



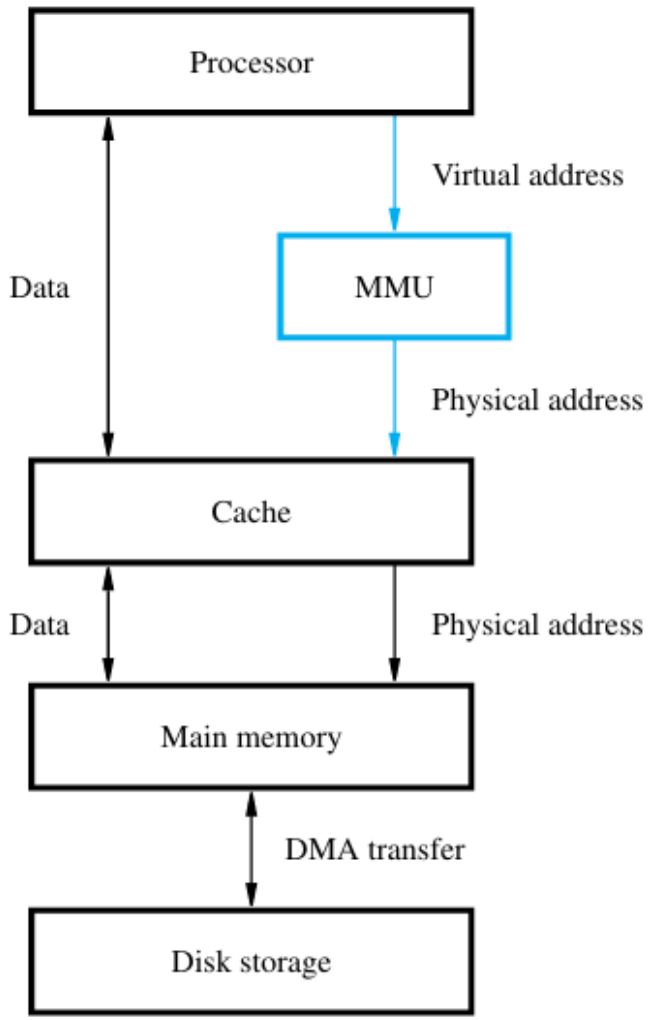


Ref. [1]

SANAL BELLEK ORGANİZASYONU

Çoğu modern bilgisayar sisteminde, fiziksel ana bellek işlemcinin adres alanı kadar büyük değildir. Örneğin, **32 bit adres**ler veren bir işlemcinin **4 GB** adreslenebilir alanı vardır. **32 bit işlemci**li tipik bir bilgisayardaki ana belleğin boyutu **1 GB** ile **4 GB** arasında değişebilir. Bir programın tümü belleğe sığmazsa, o anda yürütülmeyen parçaları genellikle bir diskte (ikincil depolama cihazı) saklanır. Bu parçalar yürütme için gerekli olduğunda, önce ana belleğe getirilmeleri ve bellekte bulunan diğer parçaların yerini almaları gerekir. Bu eylemler, **sanal bellek** olarak bilinen bir şema kullanılarak **işletim sistemi** tarafından **otomatik olarak** gerçekleştirilir. Uygulama programcılarının kullanılabilir ana belleğin getirdiği sınırlamaların farkında olmaları gerekmez. İşlemcinin tüm adres alanını kullanarak programları hazırlarlar.

Sanal bellek sisteminde, programlar ve dolayısıyla işlemci, kullanılabilir fiziksel ana bellek alanından bağımsız bir adres alanındaki komutlara ve verilere başvurur. **İşlemcinin komutlar veya veriler için verdiği ikili adreslere sanal** veya **mantıksal adres**ler denir. Bu adresler, donanım ve yazılım eylemlerinin bir kombinasyonu tarafından **fiziksel adres**lere çevrilir. Sanal bir adres, o anda fiziksel bellekte bulunan bir program veya veri alanına atıfta bulunuyorsa, ana bellekteki uygun konumun içeriklerine hemen erişilir. Aksi takdirde, başvurulan adresin içerikleri, kullanılmadan önce bellekte uygun bir konuma getirilmelidir.



Ref. [1]

BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

2025-26 GYY DERSİ
BM303
BİLGİSAYAR MİMARİSİ ve ORGANİZASYONU
Prof. Dr. A. Akbaş

ENG. TERMINOLOGY & ACRONYMS

SANAL BELLEK ORGANİZASYONU : SAYFALAMA SİSTEMİ

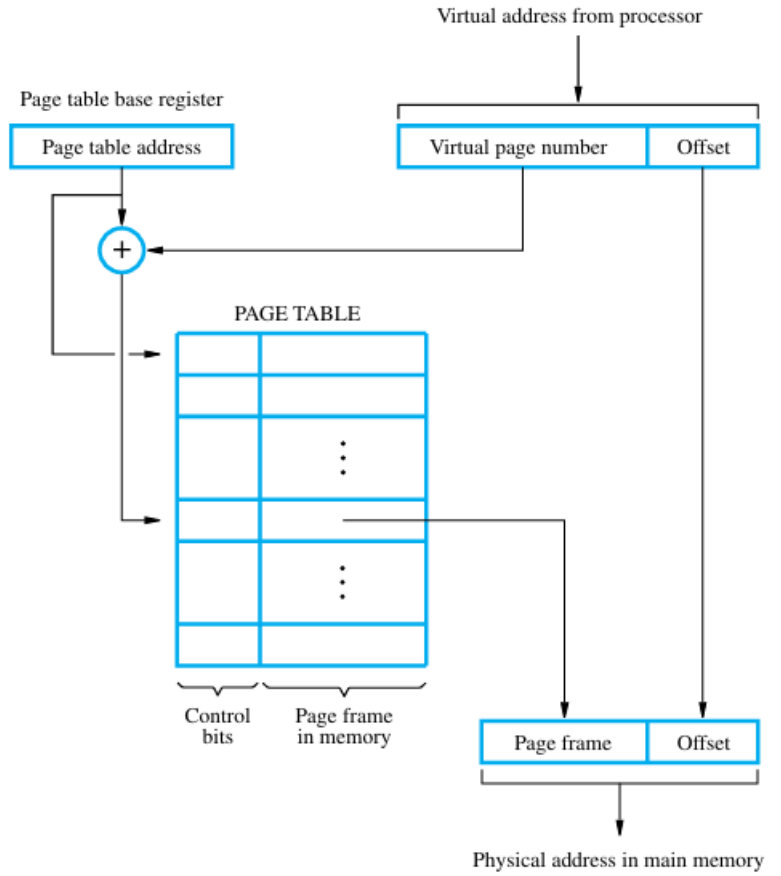
Bir sayfalama sistemi, fiziksel belleği (ana bellek) **çerçeve** adı verilen bloklara böler. Bir çerçevenin boyutu 2'nin bir kuvvetidir ve sistemler **512** bayttan **64K**'ya kadar boyutlarda çerçeveler uygulamıştır. Tipik bir boyut **4K**'dır (**32 bit** veya **4 bayt** sözcük boyutuna sahip bir sistemde **1024** adres).

Bir sayfalama sistemi, mantıksal belleği çerçevelerle aynı boyuttaki "sayfalara" böler. İşletim sistemi bir işleme bellek ayırması gerektiğinde, bir veya daha fazla boş çerçeve seçer ve işlem sayfalarını bir "**sayfa tablosu**" aracılığıyla çerçevelere eşler.

Bir işlemin sayfa tablosu, işlemin mantıksal adres alanındaki her sayfa için bir girişe sahiptir. İşlem belleğe erişim gerektirdiğinde, sistem hangi sayfaya başvurduğunu belirlemek için mantıksal adresi inceler.

Daha sonra sistem, işlemin fiziksel bellekteki hangi çerçeveye başvurduğunu öğrenmek için o mantıksal sayfanın sayfa tablosu girişine danışır. Son olarak, sistem uygun çerçevedeki uygun fiziksel bellek konumuna referansı yürütür.

Çerçeve ve sayfa boyutunu 2'nin kuvveti yapmak işlemi hızlandırır, çünkü bu mantıksal adresin rahatlıkla iki alana, bir "**sayfa numarası**" ve bir "**ofset**"e bölünebileceği anlamına gelir. Sayfa numarası işlem sayfa tablosunda hızlı bir indeks haline gelebilir ve ofset basitçe sayfa/çerçeve içindeki referansın adresi olarak kalır.



Ref. [1]

SANAL BELLEK ORGANİZASYONU : SAYFALAMA SİSTEMİ

Örneğin, **32** bitlik bir sistemin (adresler 32 bit uzunluğundadır) **4K** çerçeve boyutu varsa, bu sayfa içindeki ofset için adresin **12** bitinin gerekli olduğu anlamına gelir (bir sayfada 2^{12} **= 4096 bayt**). Adresin düşük sıralı **12 biti ofset** olur ve yüksek sıralı **20 biti sayfa numarası** olur.

İşlemin **1424** bellek konumuna referans verdiğini varsayalım. Bu sayı **4096**'dan az olduğundan, referans ilk sayfa olan **0** sayfasınadır. **32** bitlik ikili sayı olarak gösterilen **1424** sayısı aşağıdaki gibidir (okunabilirlik için adres baytlara bölünmüştür ve sayfa numarası alanı altı çizilidir):

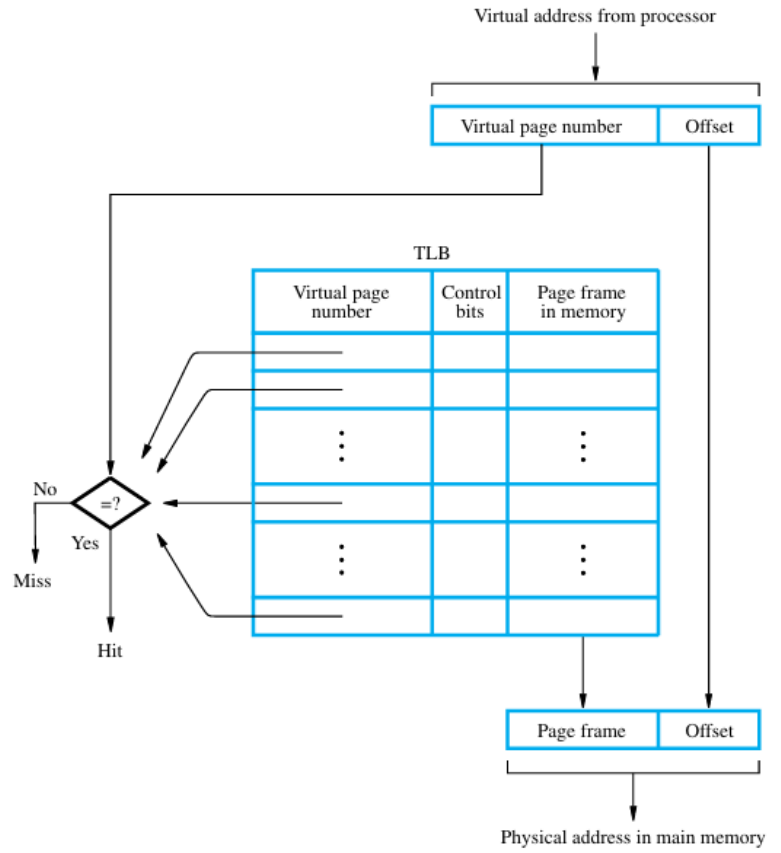
00000000 00000000 00000101 10010000

Yüksek düzeyli **20** bitin hepsinin sıfır olduğunu ve bu mantıksal adresin **0. sayfada**, yani ilk sayfada olduğunu unutmayın.

Bunun yerine işlemin **10304** bellek konumuna başvurduğunu varsayalım. Sayfa boyutu **4096** olduğundan, bu başvuru üçüncü sayfadaki, **2. sayfadaki bir konuma**dır (ilk sayfanın 0. sayfa olduğunu unutmayın). İkili sistemde **10304** sayısı şu şekildedir:

00000000 00000000 00101000 01000000

Yüksek sıralı **20 bit** artık ikili sayı **2**'yi oluşturur, bu nedenle bu referans sayfa **2**'ye aittir. Sayfa içindeki ofset, düşük sıralı **12** bit tarafından **2112** olarak verilir (**11** ve **6** bitlerinden **2048** ve **64**'ün toplamı; unutmayın, ilk bit **0** bitidir). Sayfa numarasını (**2**) sayfa boyutuyla (**4096**) çarptığınızda **8192** elde edersiniz ve **2112** ofsetini eklediğinizde **10304** adresi elde edilir.



Ref. [1]

SANAL BELLEK ORGANİZASYONU : SAYFALAMA SİSTEMİ

Şimdi bunun bir sayfa tablosuyla nasıl çalıştığına bakalım. Sayfa adresi sayfa tablosunda bir dizin haline gelir.

Sayfa tablosundaki giriş fiziksel çerçeve numarasını içerir. Sistem, fiziksel çerçeve numarası olan sayfa tablosu girişinin içeriğini adresin sayfa alanıyla değiştirir ve sonuç tam bir fiziksel adres olur.

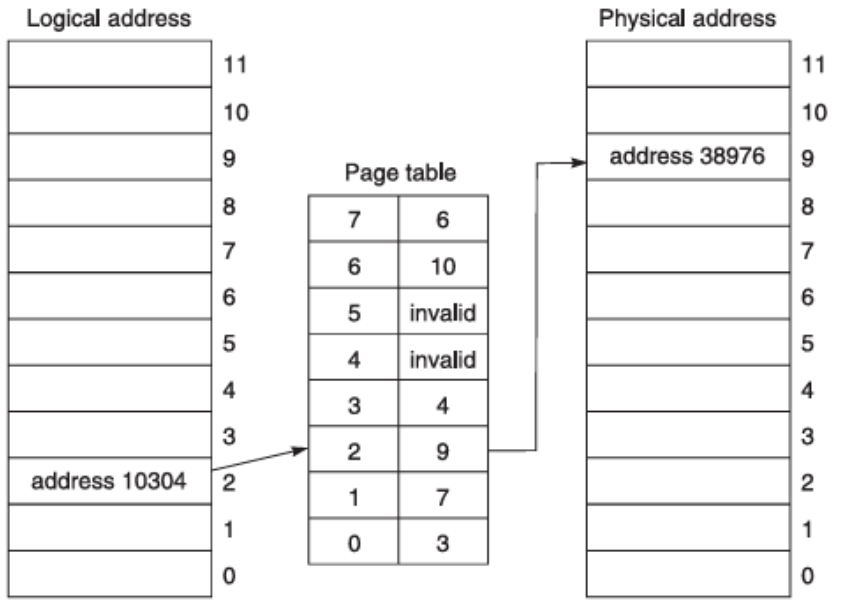
İşlemin **2. sayfası** (üçüncü sayfası) için ayrılan çerçevenin **9. çerçeve** olduğunu varsayalım. **2. sayfa** için sayfa tablosu girişi şu şekilde olur (çerçeve adresi için yalnızca **20** bit olduğunu unutmayın):

00000000 00000000 1001

Şimdi **10304** konumuna ait bellek referansı, **9** numaralı çerçeve adresinin **2** numaralı sayfa adresiyle değiştirilmesiyle sonuçlanacak ve ortaya çıkan fiziksel adres **9** numaralı çerçevede bulunan bu adres olacaktır:

00000000 00000000 10011000 01000000

Fiziksel adres **10.** çerçevede **38976**'dır (tekrar ediyorum, ilk çerçeve **0** numaralı çerçevedir). Şekil, bir işlem için sayfa tablosunun bir temsilini gösterir. İşlem için ayrılan bellek sayfalarının bitişik olması gerekmediğini unutmayın. İşletim sistemi, işlemin sayfa tablosuna çerçeveyi ekleyerek herhangi bir boş bellek çerçevesini işleme atayabilir.



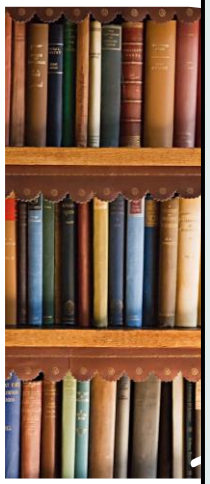
Ref. [1]

BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

2025-26 GYY DERSİ
BM303
BİLGİSAYAR MİMARİSİ ve ORGANİZASYONU
Prof. Dr. A. Akbaş

Kaynaklar

REFERANSLAR



1- Hamacher, C., Vranesic, Z., Zaky, S. and Manjikian, N. (2012). *Computer Organization and Embedded Systems.* McGraw Hill, New York. ISBN 978–0–07–338065–0

2- Bryant, R.E. and O'Hallaron, D.R. (2016). *Computer Systems, A Programmer's Perspective.* Pearson Education Limited, Malaysia. ISBN 10: 1-292-10176-8

3- Mono, M.M. (2001). *Bilgisayar Sistem Mimarisi*. Literatür Yayınları, İstanbul. ISBN 975-843-31-5

4- Stallings, W. (2005). *Computer Organisation and Architecture, Designing for Performance.* Pearson, Prentice Hall, NJ. ISBN 0-13-607373-5

5- Abd-El-Barr, M. and El-Rewini, H. (2005). *Fundamentals of Computer Organization and Architecture.* Wiley-Interscience, Hoboken-New Jersy. ISBN 0-471-46741-3

6- Reynolds, C. and Tymann, P. (2008). *Principles of Computer Science.* McGraw-Hill Schaum's Outline Series, New York. 0-07-151037-0

7- https://www.wikipedia.org v.b internet siteleri

11.nci HAFTA
DERSİNİN SONU