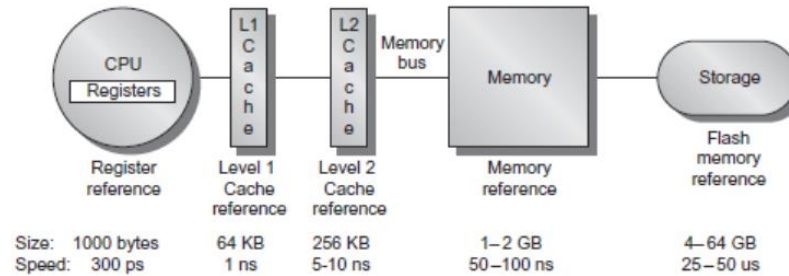


Bellek Performansı

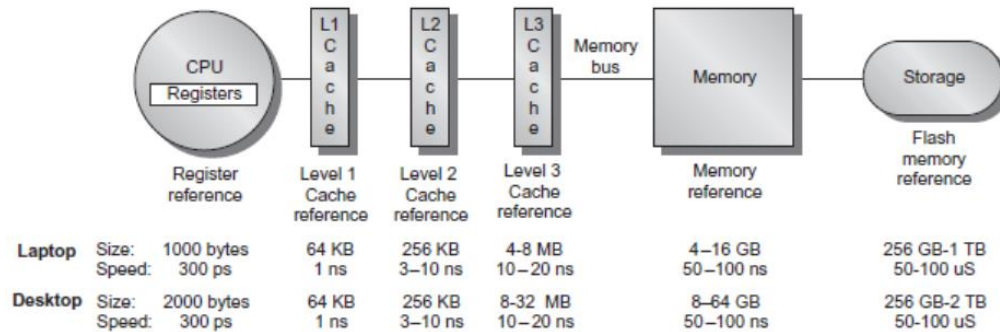
5. HAFTA

- Programcılar, düşük gecikmeye sahip sınırsız miktarda bellek ister
- Hızlı bellek teknolojisi, yavaş bellekten bit başına çok daha pahalıdır
- Çözüm: bellek sisteminin hiyerarşik organizasyonu
 - Tüm adreslenebilir bellek alanı en büyük ve en yavaş bellektedir
 - İşlemciye doğru adım adım ilerlerken her biri altındaki belleğin bir alt kümesini içeren giderek daha küçük ve daha hızlı bellekler
- Zamansal ve uzaysal yerellik, neredeyse tüm referansların daha küçük belleklerde bulunabilmesini mümkün kılar.
 - İşlemciye sunulan büyük, hızlı bir bellek gibi davranır

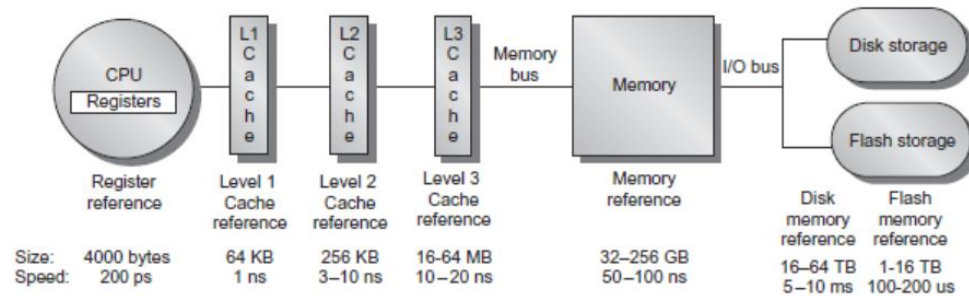
Bellek Hiyerarşisi



(A) Memory hierarchy for a personal mobile device

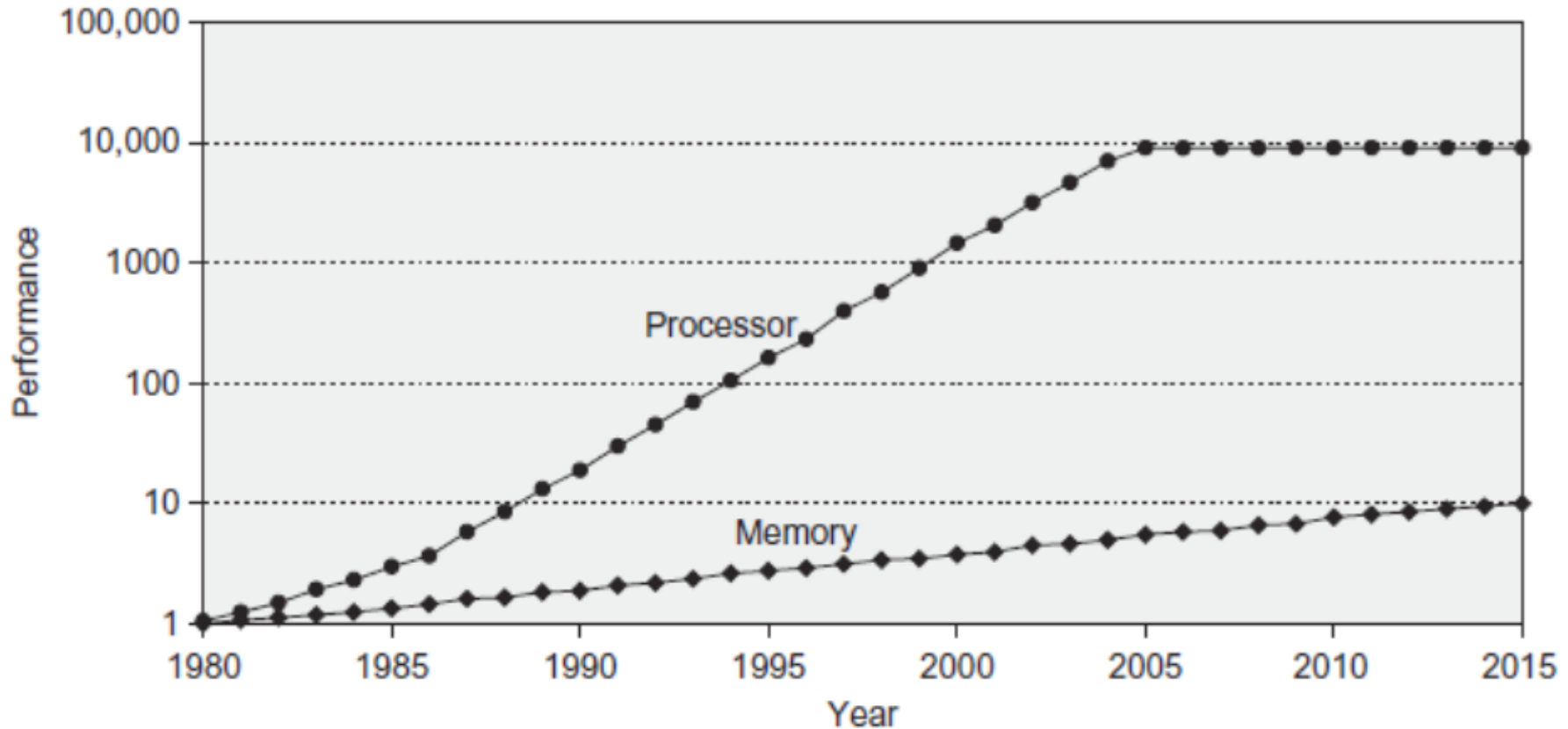


(B) Memory hierarchy for a laptop or a desktop



(C) Memory hierarchy for server

Bellek Performans Boşluğu



Level	1	2	3	4	5
Name	registers	cache	main memory	solid state disk	magnetic disk
Typical size	< 1 KB	< 16MB	< 64GB	< 1 TB	< 10 TB
Implementation technology	custom memory with multiple ports CMOS	on-chip or off-chip CMOS SRAM	CMOS SRAM	flash memory	magnetic disk
Access time (ns)	0.25 - 0.5	0.5 - 25	80 - 250	25,000 - 50,000	5,000,000
Bandwidth (MB/sec)	20,000 - 100,000	5,000 - 10,000	1,000 - 5,000	500	20 - 150
Managed by	compiler	hardware	operating system	operating system	operating system
Backed by	cache	main memory	disk	disk	disk or tape

Bellek Hiyerarşisi Tasarımı

- Bellek hiyerarşisi tasarımı, son zamanlardaki çok çekirdekli işlemcilerle daha önemli hale geliyor:
 - Bant genişliği çekirdek sayısıyla büyür:
 - Intel Core i7, çevirm başına çekirdek başına iki referans oluşturabilir
 - Dört çekirdek ve 3.2 GHz saat hızı
 - 25,6 milyar 64 bit veri referansı/saniye +
 - 12,8 milyar 128 bit talimat referansı/saniye
 - = 409,6 GB/sn!
 - DRAM bant genişliği bunun yalnızca % 8'idir (34,1 GB/sn)
 - Gereklilikler:
 - Çok bağlantı noktalı, ardışık düzen önbellekleri
 - Çekirdek başına iki seviye önbellek
 - Çip üzerinde üçüncü düzey önbellek

Performans ve Güç

- Üst düzey mikroişlemciler > 10 MB çip üzerinde önbelleğe sahiptir
 - Büyük miktarda alan ve güç - bütçe tüketir

Bellek Hiyerarşisinin Temelleri

- Önbellekte bir kelime bulunmadığında, bir *iskalama-miss* meydana gelir:
 - Daha yüksek gecikme gerektiren, hiyerarşide daha düşük seviyeden kelime getirilir
 - Alt seviye başka bir önbellek veya ana bellek olabilir
 - *Blok* içinde yer alan diğer kelimeler de getirilir
 - Yerellik prensibinden yararlanır
 - Adrese göre belirlenen, kendi *kümesi* içindeki herhangi bir konuma önbelleğe blok yerleştirilir
 - blok adresi MOD önbellekteki set sayısı

Bellek Hiyerarşisinin Temelleri

- n küme \Rightarrow n -yollu ilişkisel küme
 - Doğrudan eşlenmiş önbellek \Rightarrow küme başına bir blok
 - Tam ilişkisel \Rightarrow bir küme
- Önbelleğe yazma: iki strateji
 - *Baştan sona yazma*
 - Alt hiyerarşi seviyelerini hemen güncelle
 - *Geriye doğru yazma*
 - Yalnızca güncellenmiş bir blok değiştirildiğinde alt hiyerarşi seviyelerini güncelle
 - Her iki strateji de yazma işlemlerini asenkron yapmak için *yazma tamponunu kullanır*

Bellek Hiyerarşisinin Temelleri

- Iskalama oranı
 - Bir ıskalama ile sonuçlanan önbellek erişiminin bir oranı
- Iskalama nedenleri
 - Zorunlu
 - Bir bloğa ilk referans
 - Kapasite
 - Bloklar atılır ve daha sonra geri alınır
 - Çakışma
 - Program, önbellekte aynı konuma eşlenen farklı bloklardan birden fazla adrese tekrar tekrar başvuru yapar

Ön Bellek Performansı

$$\frac{\text{Misses}}{\text{Instruction}} = \frac{\text{Miss rate} \times \text{Memory accesses}}{\text{Instruction count}} = \text{Miss rate} \times \frac{\text{Memory accesses}}{\text{Instruction}}$$

$$\text{Average memory access time} = \text{Hit time} + \text{Miss rate} \times \text{Miss penalty}$$

- Spekülatif ve çok iş parçacıklı işlemciler, bir ıskalama sırasında diğer talimatları yürütebilir.
 - İskalamaların performansa olumsuz etkisini azaltır

$$\text{CPU execution time} = (\text{CPU clock cycles} + \text{Memory stall cycles}) \times \text{Clock cycle time}$$

$$\text{Memory stall cycles} = \text{Number of misses} \times \text{Miss penalty}$$

$$= \text{IC} \times \frac{\text{Misses}}{\text{Instruction}} \times \text{Miss penalty}$$

$$= \text{IC} \times \frac{\text{Memory accesses}}{\text{Instruction}} \times \text{Miss rate} \times \text{Miss penalty}$$

$$\begin{aligned} \text{Memory stall clock cycles} = & \text{IC} \times \text{Reads per instruction} \times \text{Read miss rate} \times \text{Read miss penalty} \\ & + \text{IC} \times \text{Writes per instruction} \times \text{Write miss rate} \times \text{Write miss penalty} \end{aligned}$$

Örnek

- Assume that 33% of the instructions in a program are data accesses. The cache hit ratio is 97% and the hit time is one cycle, but the miss penalty is 20 cycles. What is the memory stall cycles?

Örnek

- Assume that 33% of the instructions in a program are data accesses. The cache hit ratio is 97% and the hit time is one cycle, but the miss penalty is 20 cycles. What is the memory stall cycles?

$$\begin{aligned}\text{Memory stall cycles} &= \text{Memory accesses} \times \text{Miss rate} \times \text{Miss penalty} \\ &= 0.33 I \times 0.03 \times 20 \text{ cycles} \\ &= 0.2 I \text{ cycles}\end{aligned}$$

- If I instructions are executed, then the number of wasted cycles will be $0.2 \times I$.

This code is 1.2 times slower than a program with a “perfect” CPI of 1!

Örnek 1

- Talimat önbellek ıskat oranının %2 olduğunu ve veri ön bellek ıskat oranının % 4 olduğunu varsayalım. İşlemci hiçbir bellek bekletmesi olmaksızın 2 CPI değerine sahipse ve tüm ıskalar için ıskat cezası 100 çevrim ise ıskat yapmayan mükemmel bir ön bellek ile bir işlemcinin ne kadar hızlı çalışacağını belirleyiniz (tüm yükleme ve saklama frekansı %36 olarak varsayalım).

Örnek 2

- Talimat önbellek ıskı oranının %2 olduğunu ve veri ön bellek ıskı oranının % 4 olduğunu varsayalım. İşlemci hiçbir bellek bekletmesi olmaksızın 2 CPI değerine sahipse ve tüm ıskalar için ıskı cezası 100 çevrim ise ıskı yapmayan mükemmel bir ön bellek ile bir işlemcinin ne kadar hızlı çalışacağını belirleyiniz (tüm yükleme ve saklama frekansı %36 olarak varsayalım).

$$\text{Talimat ıskı çevrimi} = I \times 0,02 \times 100 = 2 \cdot I$$

$$\text{Veri ıskı çevrimi} = I \times 0,36 \times 0,04 \times 100 = 1,44 \cdot I$$

$$\text{Bellek bekleme çevrimleri} = 2I + 1,44I = 3,44I$$

$$\text{Toplam CPI} = 2 + 3,44 = 5,44$$

$$\frac{\text{Beklemeli CPU süresi}}{\text{Mükemmel önbellekte CPU süresi}} = \frac{I \times \text{CPI bekleme} \times \text{saat çevrimi}}{I \times \text{CPI mükemmel} \times \text{saat çevrimi}} = \frac{5,44}{2} = 2,72 \text{ kat}$$

Örnek 2

- 1ns saat çevrim sürelili, 20 ıska cezalı, komut başına 0,05 ıska oranlı ve 1 çevrim önbellek erişim sürelili bir işlemci için ortalama bellek erişim süresini bulunuz.

Örnek 2

- 1ns saat çevrim sürelili, 20 ıska cezalı, komut başına 0,05 ıska oranlı ve 1 çevrim önbellek erişim sürelili bir işlemci için ortalama bellek erişim süresini bulunuz.

$$\begin{aligned}\text{Ortalama bellek erişim süresi} &= \text{isabet süresi} + \text{ıska oranı} \times \text{ıska cezası} \\ &= 1 + 0,05 \cdot 20 \\ &= 2 \text{ çevrim}\end{aligned}$$

$$= 2 \text{ çevrim} \times 1\text{ns} = 2\text{ns dir}$$

Örnek 3

Example Assume we have a computer where the cycles per instruction (CPI) is 1.0 when all memory accesses hit in the cache. The only data accesses are loads and stores, and these total 50% of the instructions. If the miss penalty is 50 clock cycles and the miss rate is 1%, how much faster would the computer be if all instructions were cache hits?

Answer First compute the performance for the computer that always hits:

$$\begin{aligned}\text{CPU execution time} &= (\text{CPU clock cycles} + \text{Memory stall cycles}) \times \text{Clock cycle} \\ &= (\text{IC} \times \text{CPI} + 0) \times \text{Clock cycle} \\ &= \text{IC} \times 1.0 \times \text{Clock cycle}\end{aligned}$$

Now for the computer with the real cache, first we compute memory stall cycles:

$$\begin{aligned}\text{Memory stall cycles} &= \text{IC} \times \frac{\text{Memory accesses}}{\text{Instruction}} \times \text{Miss rate} \times \text{Miss penalty} \\ &= \text{IC} \times (1 + 0.5) \times 0.01 \times 50 \\ &= \text{IC} \times 0.75\end{aligned}$$

where the middle term $(1 + 0.5)$ represents one instruction access and 0.5 data accesses per instruction. The total performance is thus

$$\begin{aligned}\text{CPU execution time}_{\text{cache}} &= (\text{IC} \times 1.0 + \text{IC} \times 0.75) \times \text{Clock cycle} \\ &= 1.75 \times \text{IC} \times \text{Clock cycle}\end{aligned}$$

The performance ratio is the inverse of the execution times:

$$\begin{aligned}\frac{\text{CPU execution time}_{\text{cache}}}{\text{CPU execution time}} &= \frac{1.75 \times \text{IC} \times \text{Clock cycle}}{1.0 \times \text{IC} \times \text{Clock cycle}} \\ &= 1.75\end{aligned}$$

The computer with no cache misses is 1.75 times faster.

Effective Access Time

- The performance of hierarchical memory is measured by its *effective access time* (EAT).
- EAT is a weighted average that takes into account the hit ratio and relative access times of successive levels of memory.
- The EAT for a two-level memory is given by:

$$\text{EAT} = H \times \text{Access}_C + (1-H) \times \text{Access}_{MM}$$

where H is the cache hit rate and Access_C and Access_{MM} are the access times for cache and main memory, respectively.

Örnek

- Consider a system with a main memory access time of 200ns supported by a cache having a 10ns access time and a hit rate of 99%.
- Suppose access to cache and main memory occurs concurrently. (The accesses overlap.)

Örnek

- Consider a system with a main memory access time of 200ns supported by a cache having a 10ns access time and a hit rate of 99%.
- Suppose access to cache and main memory occurs concurrently. (The accesses overlap.)
- The EAT is:

$$0.99(10\text{ns}) + 0.01(200\text{ns}) = 9.9\text{ns} + 2\text{ns} = 11,9\text{ns}.$$

Örnek

- The access speed for main memory is **60 ns**
- The cache memory is 10 ns.
- If the hit ratio is 0.9 (means 9 out of 10, it will find the data or code in the fast memory (cache)), determine the effective access time.

Örnek

- The access speed for main memory is **60 ns**
- The cache memory is 10 ns.
- If the hit ratio is 0.9 (means 9 out of 10, it will find the data or code in the fast memory (cache)), determine the average memory access time.
- Effective access time = hit ratio x cache memory speed + (1 – hit ratio) x main memory speed
- Average access time = $0.9 \times 10 + 0.1 \times 60 = \mathbf{15\ ns}$
- Without cache memory, it takes **60 ns**. With cache memory, it takes **15ns**. You can see the benefit. In this case, you might ask, why don't we replace by cache memory? Money!!!!

Bellek Teknolojisi ve Optimizasyonlar

- Performans metrikleri
 - Gecikme önbellek problemidir
 - Bant genişliği, çok işlemci ve G/Ç'yi ilgilendiren bir konudur
 - Erişim süresi
 - Okuma isteği ile istenen kelimenin gelmesi arasındaki süre
 - Çevrim süresi
 - Belleğe yapılan istekler arasındaki minimum süre
- SRAM belleğinin gecikme süresi düşüktür, önbellek için kullanılır
- Yüksek bant genişliği için DRAM çiplerini birden fazla blok halinde düzenleme, ana bellek için kullanılır

Bellek Teknolojisi

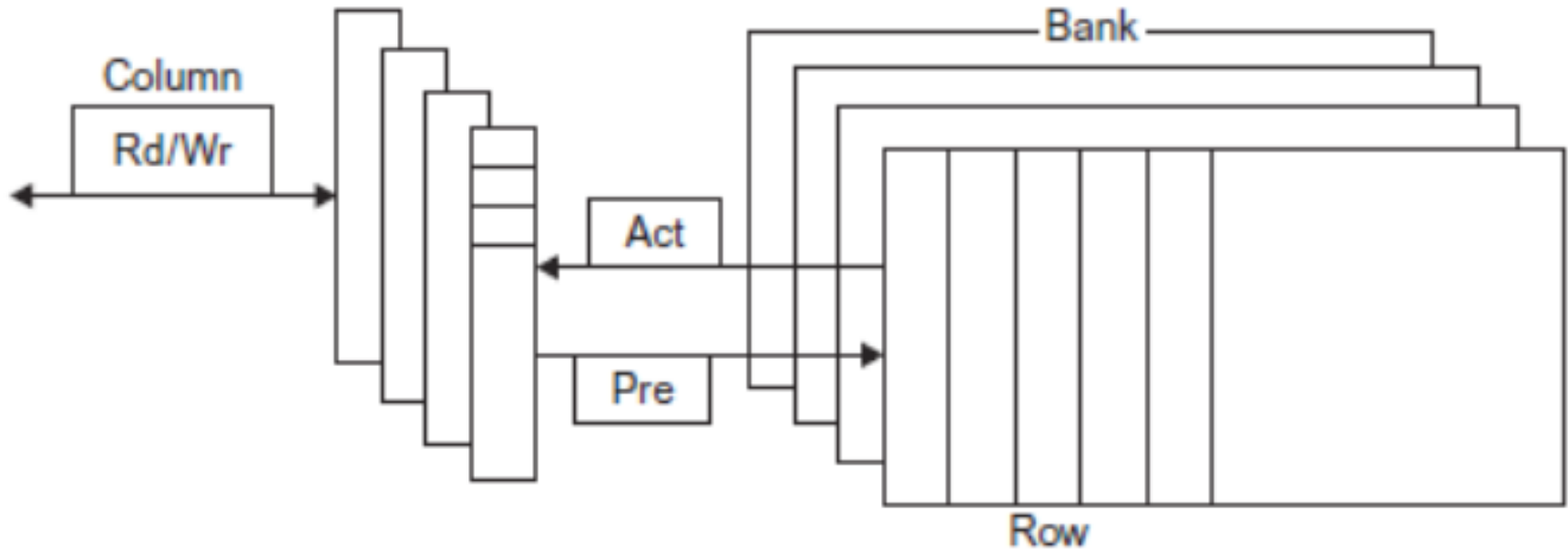
■ SRAM

- Biti tutmak için düşük güç gerektirir
- 6 transistör/bit gerektirir

■ DRAM

- Okunduktan sonra yeniden yazılmalıdır.
- Ayrıca periyodik olarak yenilenmelidir
 - Her ~ 8 ms'de bir (kabaca zamanın %5'i)
 - Her satır aynı anda yenilenebilir
- Bir transistör/bit
- Adres satırları çoğullanır:
 - Adresin üst yarısı: satır erişim strobu (RAS)
 - Adresin alt yarısı: sütun erişim strobu (CAS)

DRAM'ın İç Organizasyonu



Bellek Teknolojisi

- Amdahl:
 - Bellek kapasitesi, işlemci hızıyla doğrusal olarak büyümelidir
 - Ne yazık ki, bellek kapasitesi ve hızı işlemcilere ayak uyduramadı
- Bazı optimizasyonlar:
 - Aynı satıra birden çok erişim
 - Senkron DRAM
 - DRAM arayüzüne saat eklendi
 - Önce kritik kelime ile seri çekim modu
 - Daha geniş arayüzler
 - Çift veri hızı (DDR)
 - Her DRAM cihazında birden fazla bank

Bellek Optimizasyonları

Production year	Chip size	DRAM type	Best case access time (no precharge)			Precharge needed
			RAS time (ns)	CAS time (ns)	Total (ns)	Total (ns)
2000	256M bit	DDR1	21	21	42	63
2002	512M bit	DDR1	15	15	30	45
2004	1G bit	DDR2	15	15	30	45
2006	2G bit	DDR2	10	10	20	30
2010	4G bit	DDR3	13	13	26	39
2016	8G bit	DDR4	13	13	26	39

Bellek Optimizasyonları

Standard	I/O clock rate	M transfers/s	DRAM name	MiB/s/DIMM	DIMM name
DDR1	133	266	DDR266	2128	PC2100
DDR1	150	300	DDR300	2400	PC2400
DDR1	200	400	DDR400	3200	PC3200
DDR2	266	533	DDR2-533	4264	PC4300
DDR2	333	667	DDR2-667	5336	PC5300
DDR2	400	800	DDR2-800	6400	PC6400
DDR3	533	1066	DDR3-1066	8528	PC8500
DDR3	666	1333	DDR3-1333	10,664	PC10700
DDR3	800	1600	DDR3-1600	12,800	PC12800
DDR4	1333	2666	DDR4-2666	21,300	PC21300

	2000	2003	2007	2014	2020
	DDR	DDR2	DDR3	DDR4	DDR5
Power Consumption(V)	2.5	1.8	1.35/1.5	1.2	1.1
Transfer Rate(MT/s)	200-400	400-800	800-1866	1866-3200	4800-6400
Peak Transfer Rate(MB/s)	3200	6400	14928	25600	84000
I/O Bus Clock (MHz)	100-200	200-400	400-933	933-1600	1600 – 4800
DIMM Pins	184	240	240	288	288
SO-DIMM Pins	200	200	204	260	262

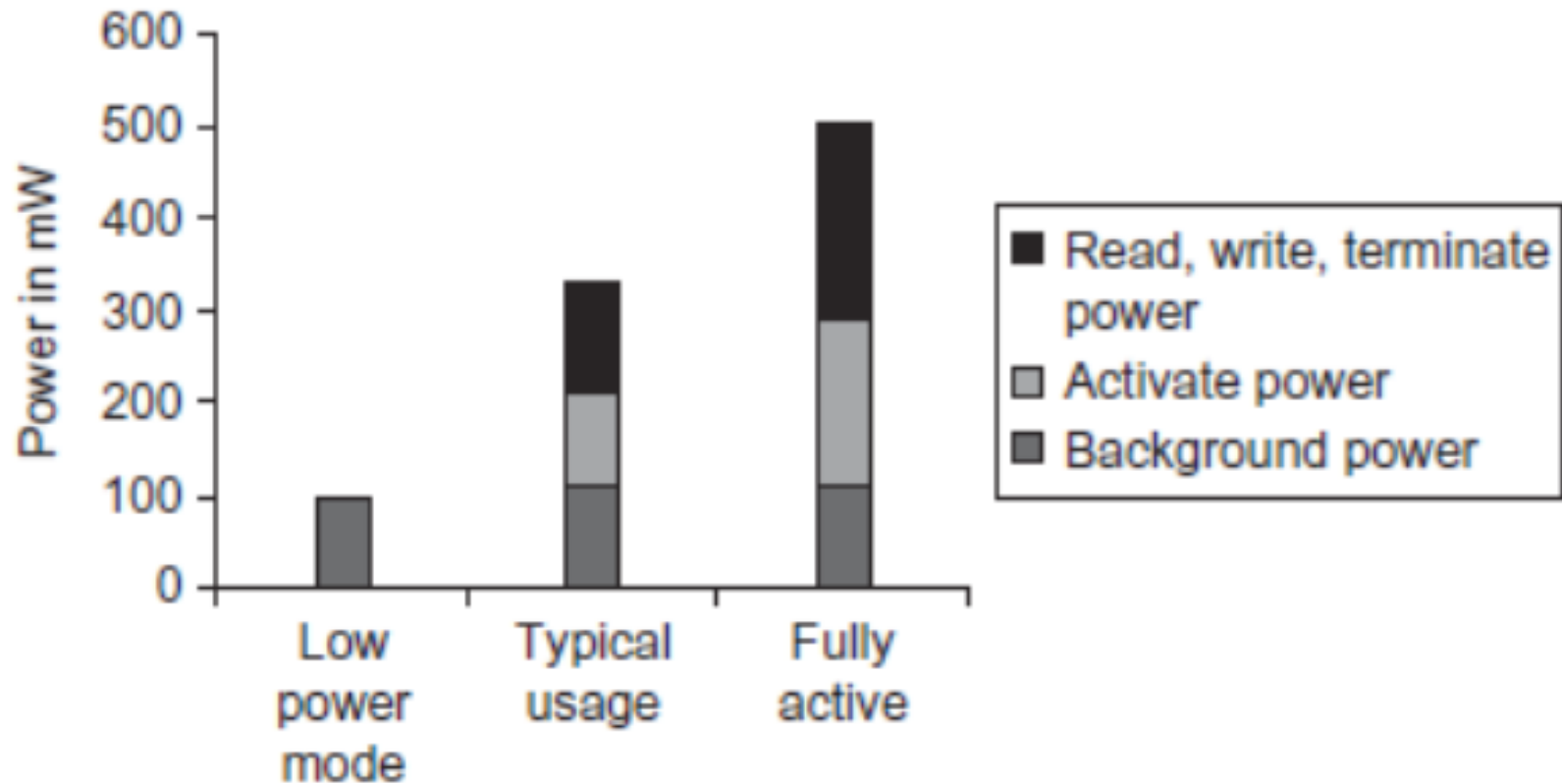
Bellek Optimizasyonları

- **DDR:**
 - **DDR2**
 - Daha düşük güç (2,5 V -> 1,8 V)
 - Daha yüksek saat hızları (266 MHz, 333 MHz, 400 MHz)
 - **DDR3**
 - 1,5 V
 - 800 MHz
 - **DDR4**
 - 1-1.2V
 - 1333 MHz
 - **DDR5**
 - 1.1V
- **GDDR5, DDR3 tabanlı grafik belleğidir**

Bellek Optimizasyonları

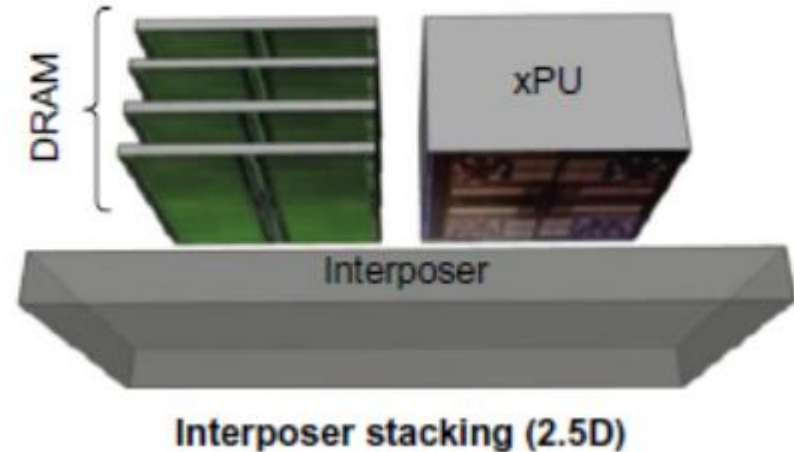
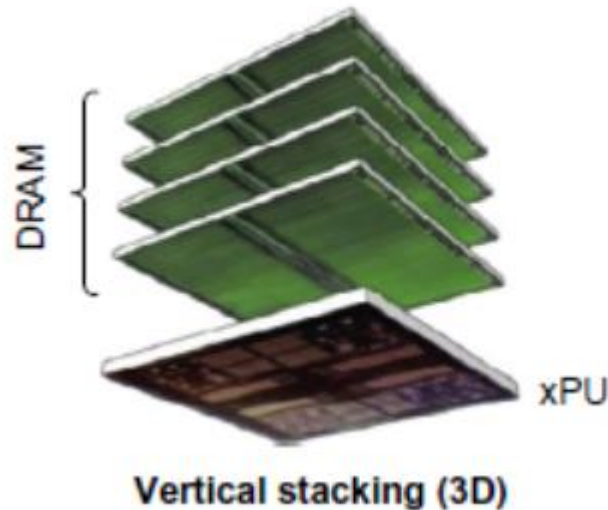
- SDRAM'lerde gücün azaltılması:
 - Düşük voltaj
 - Düşük güç modu (saati yok sayar, yenilemeye devam eder)
- Grafik belleği:
 - DDR3'e karşı DRAM başına 2-5 X bant genişliği
 - Daha geniş arayüzler (16'ya karşı 32 bit)
 - Daha yüksek saat hızı
 - Soket yerine lehimleme ile bağlanan DIMM modülleri

Bellek Güç Tüketimi



Yığılmış/Gömülü DRAM'ler

- İşlemci ile aynı pakette yığılmış DRAM'ler
 - Yüksek Bant Genişliği Belleği (HBM)



Flash Bellek

- EEPROM Türü
- Türler: NAND (daha yoğun) ve NOR (daha hızlı)
- NAND Flaş:
 - Okumalar sıralıdır, tüm sayfayı okur (.5 ila 4 KiB)
 - İlk bayt için 25 us, sonraki baytlar için 40 MiB/s
 - SDRAM: İlk bayt için 40 ns, sonraki baytlar için 4,8 GB/sn
 - 2 KiB aktarımı: SDRAM için 75 uS vs 500 ns, 150X daha yavaş
 - Manyetik diskten 300 ila 500X daha hızlı

NAND Flash Bellek

- Üzerine yazılmadan önce (bloklar halinde) silinmelidir
- Uçucu değildir, sıfır güç
- Sınırlı sayıda yazma döngüsü (~100.000)
- 2\$/GiB, SDRAM için 20-40\$/GiB ve manyetik disk için 0.09 GiB ile karşılaştırıldığında
- Faz Değişimi/Memrister Belleği
 - Muhtemelen yazma performansında 10 kat ve okuma performansında 2 kat iyileştirme

Bellek Güvenilirliği

- Hafıza kozmik ışınlar karşı hassastır
- *Yazılım hataları* : dinamik hatalar
 - Hata düzeltme kodları (ECC) tarafından tespit edilir ve düzeltilir
- *Zor hatalar* : kalıcı hatalar
 - Arızalı satırları değiştirmek için yedek satırları kullanır
- Chipkill : RAID benzeri bir hata kurtarma tekniği

Gelişmiş Optimizasyonlar

- İsabet Sayısını düşürme
 - Küçük ve basit birinci seviye önbellekler
 - Yol tahmini
- Bant genişliğini artırma
 - Sıralı önbellekler, çok banklı önbellekler, bloke olmayan önbellekler
- Iskalama cezasını azaltma
 - Önce kritik kelime, yazma arabelleklerini birleştirme
- Iskalama oranını azaltma
 - Derleyici optimizasyonları
- Parallelleştirme yoluyla ıskalama cezasını veya ıskalama oranını azaltma
 - Donanım veya derleyici prefetch

Sanal Bellek ve Sanal Makineler

- Sanal bellek ile koruma
 - Prosesleri kendi bellek alanlarında tutar
- Mimari Rolü
 - Kullanıcı modu ve süpervizör modu sağlar
 - CPU durumunun belirli yönlerini korur
 - Kullanıcı modu ve süpervizör modu arasında geçiş yapmak için mekanizmalar sağlar
 - Bellek erişimlerini sınırlamak için mekanizmalar sağlar
 - Adresleri dönüştürmek için TLB sağlar

Sanal makineler

- İzolasyon ve güvenliği destekler
- Bir bilgisayarı birçok kullanıcı arasında paylaşma
- İşlemcileri ek yükü kabul edilebilir hale getirir
- Farklı ISA'ların ve işletim sistemlerinin kullanıcı programlarına sunulmasına izin verir
 - “Sistem Sanal Makineleri”
 - SVM yazılımına "sanal makine monitörü" veya "hipervizör" denir.
 - Monitörün altında çalışan bireysel sanal makinelere "misafir VM'ler" denir.

VMM gereksinimleri

- Konuk yazılımı şunları yapmalıdır:
 - Yerel donanım üzerinde çalışıyormuş gibi davranır
 - Gerçek sistem kaynaklarının tahsisini değiştirememek
- VMM, konukları "bağlam değiştirme" yapabilmelidir
- Donanım aşağıdakilere izin vermelidir:
 - Sistem ve kullanım işlemci modları
 - Sistem kaynaklarını tahsis etmek için ayrıcalıklı talimat alt kümesi

VM'lerin Sanal Belleğe Etkisi

- Her konuk işletim sistemi kendi sayfa tablolarını tutar
 - VMM, fiziksel ve sanal bellek arasına “gerçek bellek” adı verilen bir bellek düzeyi ekler.
 - VMM, konuk sanal adreslerini fiziksel adreslerle eşleyen gölge sayfa tablosunu korur
 - VMM’in konuğun kendi sayfa tablosunda yaptığı değişiklikleri algılamasını gerektirir
 - Sayfa tablosu işaretçisine erişim ayrıcalıklı bir işlemse doğal olarak gerçekleşir