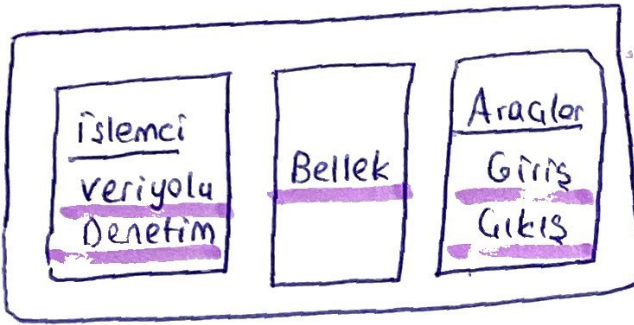


Moore Yasası → Her 18 ayda bir transistör sayısının 2 katına çıkması

Güç Duvarı

Clock rate ↑ Power ↓
Soğutma ihtiyacı ↑
Enerji tüketimi ↑

Ana Bileşen



(Intel core, x86, System)
CISC

(Complex Instruction Set Comp.)
→ Çok sayıda karmaşık
buyruk

→ Buyruk boyutları
değişken

→ Karmaşık donanım

→ Belleğe erişim
herhangi bir buyruk
tarafından gerçekleştirilebilir.

→ Çok sayıda
adresleme kipi

(MIPS, ARM, RISC-V)
RISC

(Reduced Instruction Set Computer)
→ Az sayıda basit büyük

→ Buyruk boyutları sabit

→ Karmaşık derleyici

→ Yalnızca load ve store buyrukları belleğe erişim gerçekleştirir.

→ Az sayıda adresleme kipi

* Bellekte bulunan veri üzerinde işlem yapmak için
→ Veri aktarma buyrukları (data transfer instructions)

Küçükü basta → sayının en küçük, en anlamsız,
en sağdaki baytının en küçük adresli bellek konumuna
yazılması (RISC'in tercihi)

Büyükü basta → Sayının en büyük, en anlamlı, en soldaki
baytının en küçük adresli bellek konumuna yazılması

Von Neumann Mimarisi

→ Aynı fiziksel bellek uzayında
hem buyruklar hem de veri

→ Buyruklar ve veriler aynı
hattı kullanır.

→ işlemci buyruklara ve
veriye aynı anda erişemez.

→ 1 buyruğun tamamlanması
2 çevrim sürer.

→ Denetim biriminin maliyeti
daha düşüktür.

Harvard Mimarisi

→ Veri ve buyruklar farklı
fiziksel bellek uzayında

→ Veri hattı ve buyruk
hattı ayrıdır.

→ işlemci buyruklara ve
veriye aynı anda erişebilir.

→ 1 buyruğun tamamlan-
ması 1 çevrim sürer.

→ Denetim biriminin maliyeti
daha yüksektir.

* Kodun belleğe saçılması (code spilling)

↳ Tüm değişkenler her zaman yazmaqlarda bulunamaz

↳ Yazmaqlara erişim, belleğe erişimden hızlıdır.

Program Yığıtı → Öğelerden son gelenin ilk işlem göreceği biçimde üst üste yığıldığı varsayılan veri yapısı
↳ Yığıt bellekte tutulur. Yığıtın bellekteki adresi yığıt işaretçisi (stack pointer) ile belirtilir.

MIPS { Push → Yığıtın en üstüne veri eklemek
 Pop → Yığıttan en üstteki veriyi çıkarmak

RISC-V Adresleme Kipleri

- 1) Anlık Adresleme (Immediate Addressing)
- 2) Yazmaç Adresleme (Register Addressing)
- 3) Eklemeli Adresleme (Base Addressing) → yazmaça anlık değer eklenir.
- 4) Göreceli Adresleme (PC-relative Addressing)
↳ Dallanma adresi; Sayacı + anlık değer

İşlemci Tasarım Aşamaları

- 1) Büyük kümesi mimarisi belirlenmesi
- 2) BKM gereksinimleri belirlenmesi
- 3) Veriyolu oluşturulması
- 4) Denetim birimlerinin oluşturulması

* Durağan (Static) Buyruk → Bellekte duran buyruklar

* Dinamik Buyruk → İşlemci içerisinde işlem gören buyruklar

Bilgisayar Mimarisi Tasarım İlkeleri

- 1) Olağan durumu hızlandır.
- 2) Yalınlık düzenden gelir.
- 3) Küçük olan hızlıdır.
- 4) İyi tasarım ödünleşme gerektirir.

Boru Hattında Olusan Sorun

1) Yapi Sorunu (Structural Hazard) → Yeterince donanim kaynagi bulunmadigi durum

↳ Kaynak artırilarak gozulebilir (Bellek sayisi artırilir)

* Tek bellek olsaydi (buyruk + veri) getir asamasi ve bellek asamasi arasinda yapı sorunu olurdu.

Gözüm = Kapi eklemek, ayri bellek birimleri

2) Veri Sorunu (Data Hazard) → gereken verinin hazır olmaması

* veri bagimliliği - RAW (read after write)

Gözüm = Yazılım → veri bagimliliği olmayacak sekilde kod üretmek X zordur.

Donanim → Veri yönlendirmesi

* Yükle-kullan (load-use) veri bagimliliği (Bellek ↔ Göz)

↳ Zamanda geri gidilemez, 1 vuruşluk gecikme kaçınılmaz!

↳ Boru hattı duraklaması (pipeline stall)

3) Denetim Sorunu (Control Hazard) → işlemciye getirilen buyruğun yürütülmesi gereken buyruk olmadığı durum.

* Dallanma buyrukları geldiğinde boru hattı duraklatılırsa 2 saat vuruşu gecikme olur.

Gözüm = 1) Geciktirilmiş Dallanmalar (yazılım)

→ Dallanma sonucundan bağımsız buyruklar dallanma sırasında oluşan boşluğa yerleştirilebilir.

2) Dallanma Öngörüsü (donanim)

a) Durağan Dallanma Öngörücüleri (hep atlar ya da atlamaz)

b) Devingen Dallanma Öngörücüleri

→ Bir önceki dallamanın atlayıp atlamadığına bakarak tahmin (Last value prediction)

→ Gift doruklu öngörücü

→ Dallanma Hedef Ara Belleği (Branch Target Buffer)

* Bir önceki hedef adresi tutan bir önbellektir. (Her dallanma için)

→ Yerel geçmişe dayalı öngörücüler (local history)

* Her dallanmanın kendisine ait geçmiştir.

→ Genel geçmişe dayalı öngörücüler (global history)

* Program içindeki dallanmaların beraber geçmiştir.

→ Yerel ve genel geçmişi beraber kullanmak (G-Share)

→ Birden çok öngörücüyü beraber kullanmak

* En iyi öngörücüyü seçen dallanma öngörücüne turnuva öngörücüsü (tournament predictors) denir.

* RISC-V'te beklenmedik bir değişime sebep olan olay;

→ Bir buyruktan kaynaklanıyorsa → Olağandışı durum (exception)

→ İşlemcinin dışından kaynaklanıyorsa → Kesme (interrupt)

Olağandışı Durumların Üstesinden Gelme

1) Duruma yol açan buyruğun adresi SEPC (Supervisor Exception Program Center) yazmasına kaydedilir.

2) İşletim sistemi durumun üstesinden gelir.

→ programı sonlandırır

→ program kaldığı yerden devam eder. (SEPC'ye sıplar.)

★ İki bilgiye ihtiyaç var:

1) Duruma sebep olan buyruk (SEPC'de kayıtlı)

2) Durumun oluşma sebebi

→ Durumun sebebi SCAUSE (Supervisor Exception Cause Register) yazmasına kaydedilir.

* RAM → Random Access Memory → önbellek

* SRAM (Durağan Rastgele Erişimli Bellek) → erişim daha hızlı, daha büyük, daha fazla transistör, fiyatları daha yüksek

* DRAM (Devrinen Rastgele Erişimli Bellek) ★

→ Elektrik kesildiğinde içerik kaybedilir. → Ana bellek

* DRAM, aynı satıra gelen istekleri ara bellekten karşılar.
(Row-buffer)

→ erişim zamanını kısaltır, daha hızlı okunur.

* Program verisini bellekte paralellik oluşturacak şekilde saklamaya binleştirme denir.

Flash Bellek
Manyetik Teker (Disk) } Kalıcı bellek

Yerellik

1) Zamanda Yerellik → Aynı veriye, aynı adrese belirli bir süre arafında erişilmesi

2) Alanda Yerellik → Birbirine yakın bellek adreslerine yazılmış verilere erişilmesi (diziler)

Önbellek Temel Kavramlar

1) Verinin bulunması (hit)

2) Verinin bulunmaması (miss)

3) Bulma zamanı

4) Bulamama gecikmesi → sonraki aşama belleğe geçip arama

Doğrudan Eşlemeli Önbellek

→ Bir veri bellekte yalnızca bir yerde olabilir

* İki adres aynı satıra denk geldiğinde sürekli birbirlerinin verisini önbellekten çıkarırlar. → Pinpon Etkisi

→ önbellek boyutu artırılarak giderilebilir.

Tam ilişkili Önbellek

→ veri bellekte istenilen yere kaydedilir.

* Pinpon etkisi ortadan kalkar.

Kümüli ilişkili önbellek

→ Bir satırda birden fazla verinin adreslenmesi

Doğrudan Eşlemeli

→ Okumak istenen verinin nerede bulunacağı bellidir.
↳ erişim süresi kısadır.

→ Bellekte tutulan etiket daha kısadır.

→ 1 tane karşılaştırmacıya ihtiyaç duyar.

Tam ilişkili

→ Veri istenilen yere yazılabilir.
↳ Pinpon etkisi ortadan kalkar.

→ Adresin büyük bir bölümü etiket olarak saklanmak zorundadır.

→ Her satır için bir karşılaştırmacıya ihtiyaç duyar.

Yazma İşlemi

* Verinin ne zaman ana belleğe yazılacağını tasarım belirler.

1) Doğrudan Yazma (write through) → belleğin tüm aşamalarına o anda yazma → uzun sürer ve performansı düşürür.

2) Sonradan Yazma (writeback) → sadece ön belleğe yazı, belleğe yazma işlemini sonraya bırak. → hızlı ve karmaşıktır.

* Ön bellekte olmayan bir adrese yazıldığında onun ön belleğe getirilip getirilemeyeceği;

1) Yaz ve yerini ayır (write allocate) → yazma işlemi ana belleğe yapılır ve veri ön belleğe getirilir.

2) Yaz ve yerini ayırma (no write allocate) → Yazma işleminin sonra veri ön belleğe yazılmaz.

Sanal Bellek

1) Sistemde tüm programların gereksinimlerini karşılayacak kadar büyük bir bellek olmayabilir.

2) Programcı programın çalışacağı sistemdeki bellek boyutunu bilemez.

3) Programcı kendi programı yanında hangi programların çalışacağını bilemez.

* İkincil bellek ile ana bellek arasındaki veri akışı sayfalar aracılığıyla yapılır.

→ Belleğe erişim için sanal adreslerin fiziksel adreslere dönüştürülmesi gerekir. (Adres dönüşümü / page translation)

* Eşlesmeler sayfa tablosu (page table) isimli bir yapıda bellekte tutulur.

* Dönüşüm için işlemcide tutulan önbellege Etkin Sayfalar Önbellesi (Translation Lookaside Buffer) denir.

→ Büyük ve veri için farklı önbellekler bulunur.

Sayfa Hatası (Page Fault) → denetim işletim sisteminde

- 1) Fiziksel sayfa ikincil bellektedir.
- 2) Fiziksel sayfa henüz atanmamıştır.

* Önceden hangi sayfalar ikincil belleğe alındı?

Bu sayfalar için ayrılmış bölüm (bellekte) → takas dosyası (swap space)

* Döngü Açılması (Loop Unrolling)

- 1) Devingen buyruk sayısının azalması
 - 2) Dallanma buyruklarının kaldırılması
 - 3) Durağan buyruk sayısının artması
- ↳ Kodun büyümesi
- } +
} -