

Bilgisayar Mimarisi

Boru Hattı
İş Hattı
Pipe Line

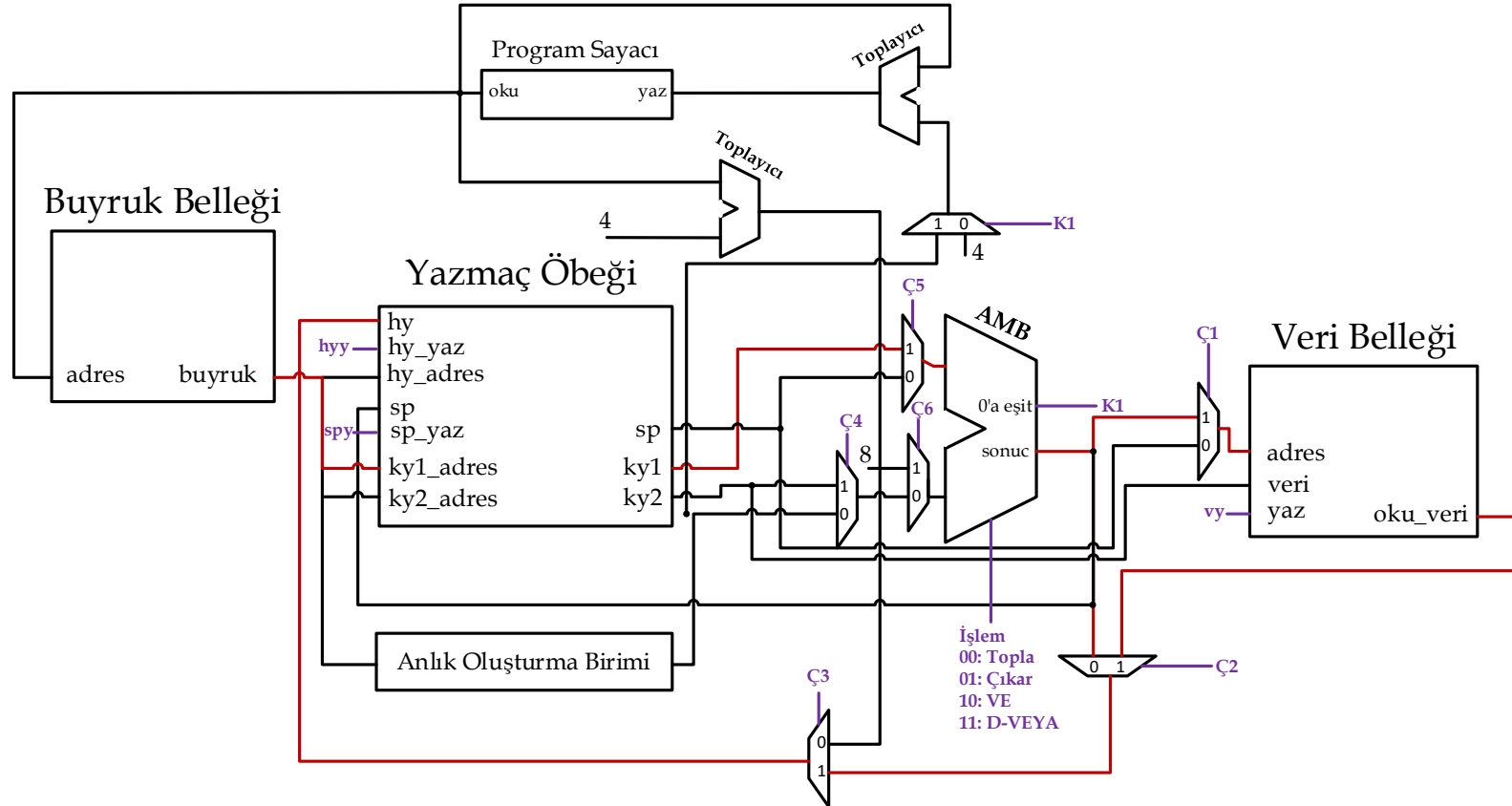
İnönü Üniversitesi
Bilgisayar Mühendisliği

Tek Palslik İşlemciler

İşlemcilerde saat frekansını devredeki **kritik yol** belirler.

İşlemcimizde kritik yol: *ld buyruğu*.

Komut oku → Register oku → ALU → Bellekten oku → Registıra yaz

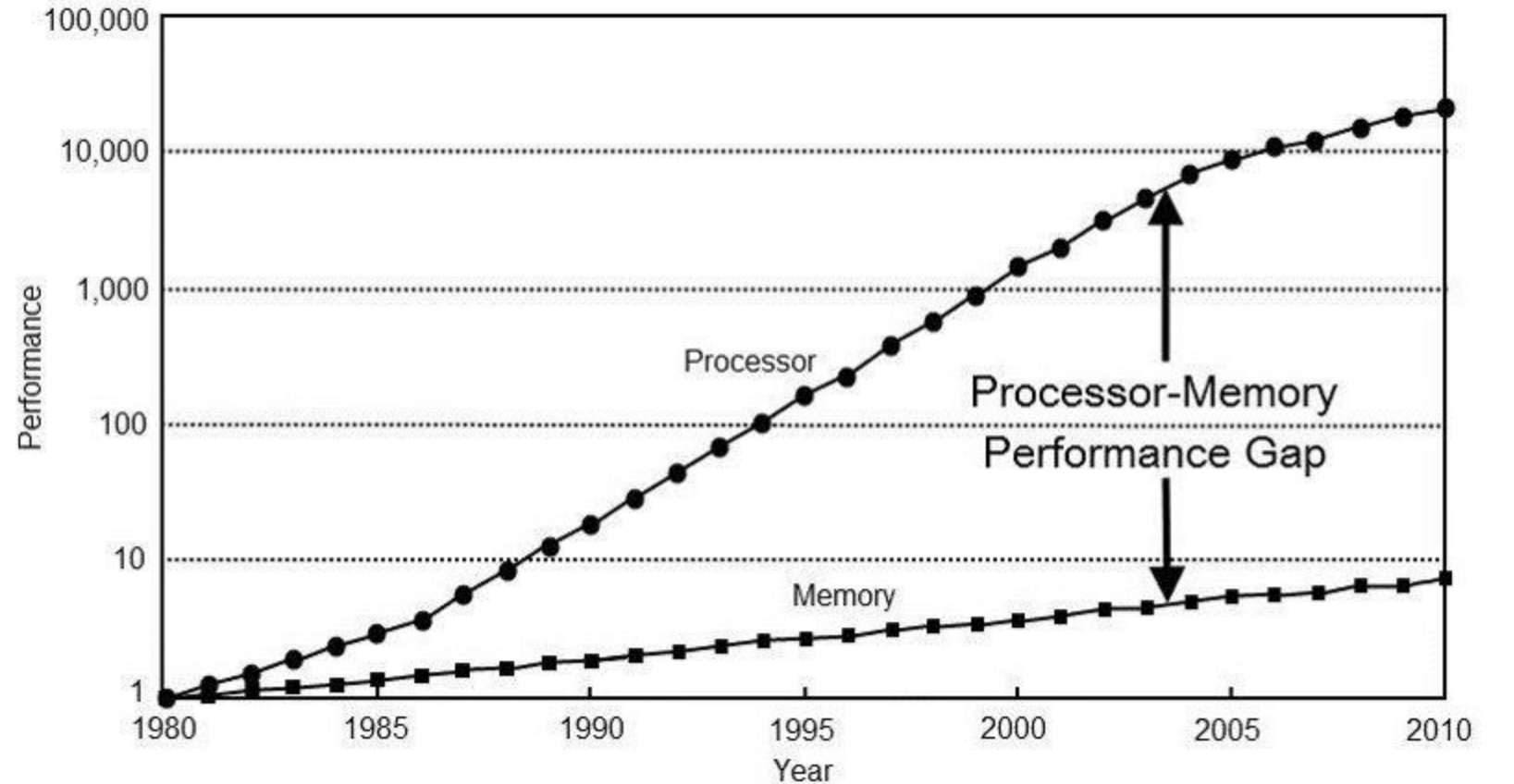


Tek Palslik İşlemciler

Bellek gecikmeleri:

1. Sabit olmayabilir.
2. İşlemci içi gecikmelere kıyasla çok uzun.

Bellek
Duvarı



Tek Palslik İşlemciler

“Olağan durumu hızlandır” ilkesi kullanılarak tek palslik işlemcilerin başarımı **artırılamaz**.

→ Saat frekansını artırabilmek için en kötü durumu hızlandırmamız gerekiyor.

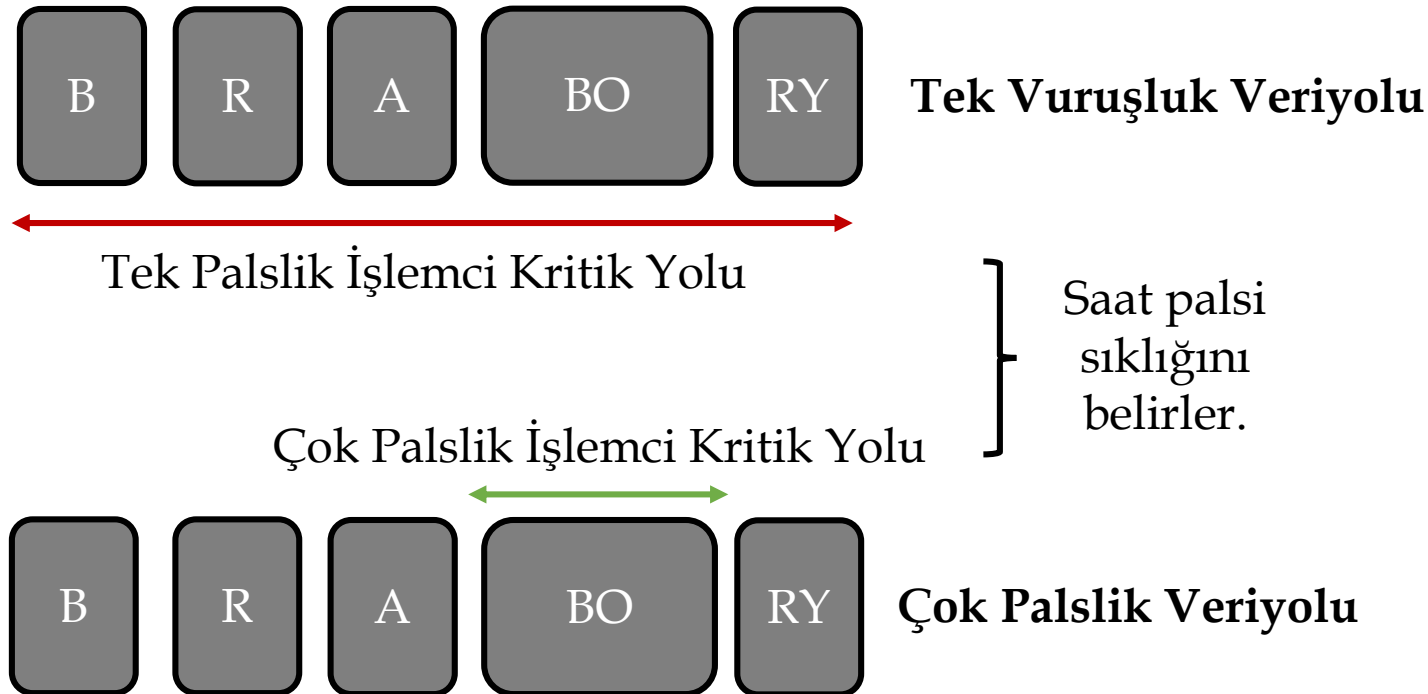
Günümüzde tek palslik işlemcilerin yerini çok palslik işlemciler almıştır.

Çok Palislik (Vuruşlu) İşlemciler

Kritik yol: *ld* buyruğu.

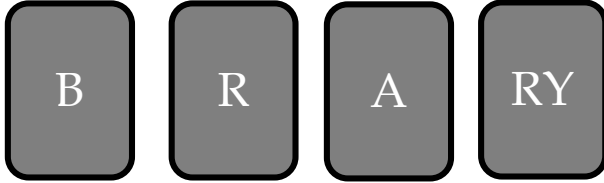
Komut oku → Register oku → ALU → Bellekten oku → Registıra yaz
(B) (R) (A) (BO) (RY)

Bu aşamaların her biri bir saat vuruşunda gerçekleşebilir.



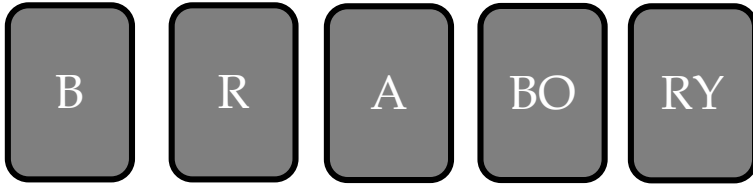
Çok Palslik İşlemciler

add ...



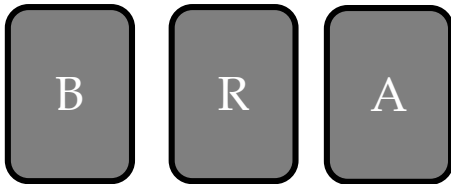
topla komutu bellekten okumaz

ld ...



yükle komutu tüm aşamalardan geçer

beq ...



dallanma komutları bellekten okumaz,
register öbeğine yazmaz.

Örnek

addi x9, x0, #0

addi x10, x0, #0

addi x11, x0, #A

lb x12, 0(x11)

sb x12, 1(x11)

addi x11, x11, #1

addi x9, x9, #1

ble x9, x10, #-16 // lb'ye zıpla

Çok palslik işlemcide:

-> addi 4 pals

-> sb 4 pals

-> lb 5 pals

-> ble 3 pals sürüyor.

Tek palslik işlemcinin saat frekansı çok palslik işlemcinin **dörtte biri** olduğuna göre bu iki işlemcinin başarımlarının oranı nedir?

Örnek

addi x9, x0, #0

addi x10, x0, #5

addi x11, x0, #A

lb x12, 0(x11)

sb x12, 1(x11)

addi x11, x11, #1

addi x9, x9, #1

ble x9, x10, #-16 // lb'ye zıpla

Çok palslik işlemcide:

-> addi 4 pals

-> sb 4 pals

-> lb 5 pals

-> ble 3 pals sürüyor.

Tek palslik işlemcinin saat frekansı çok palslik işlemcinin **dörtte biri** olduğuna göre bu iki işlemcinin başarımlarının oranı nedir?

Örnek

100.000.000 tane A tipi, 200.000.000 tane B tipi komuttan oluşan program K1 ve K2 işlemcilerinde çalıştırılıyor. K1 işlemcisinin saat frekansı 100 MHz'dir ve K1 tek palsliktir. K2 işlemcisinin saat frekansı 400 MHz'dir ve K2 çok palsliktir. K2 işlemcisinde A tipi komutlar 6 çevrimde, B tipi komutlar ise 3 çevrimde tamamlandığına göre K1 ve K2'den hangisi daha hızlıdır?

Tek mi Çok mu?

100.000.000 tane A tipi, 200.000.000 tane B tipi komuttan oluşan program K1 ve K2 işlemcilerinde çalıştırılıyor. K1 işlemcisinin saat frekansı **100 MHz'dir** ve K1 **tek palsliktir**. K2 işlemcisinin saat frekansı **400 MHz'dir** ve K2 **çok palsliktir**. K2 işlemcisinde **A tipi komutlar 6 çevrimde**, **B tipi komutlar ise 3 çevrimde** tamamlandığına göre K1 ve K2'den **hangisi daha hızlıdır?**

$$Yürütme\ Zamanı = \frac{(BBÇ \times Buyruk\ Sayısı)}{Saat\ Sıklığı}$$

$$Yürütme\ Zamanı_{K1} = \frac{300.000.000}{100 \times 10^6} = \frac{300}{100} = 3\ saniye$$

$$Yürütme\ Zamanı_{K2} = \frac{100.000.000 \times 6 + 200.000.000 \times 3}{400 \times 10^6} = \frac{1200}{400} = 3\ saniye$$

İşlemci Tasarım İlkeleri

1- Olağan durumu hızlandır.

İt ve çek (Push Pull) komutları: Program yığıtına veri yazıldıktan hemen sonra yığıt işaretçisi azaltılır, yığıttan veri okunduktan hemen sonra yığıt işaretçisi artırılır. İt ve çek komutları ile *iki çevrimde* yapacağımız işi *tek çevrime* indirdik.

2- Yalınlık düzenden gelir.

VAX mimarisinin CMPCx buyruğu bellekteki iki karakter dizisini karşılaştırır: “Eğer bir dizi diğerinden daha uzunsa kısa olana baytlar eklenerek diziler eşit boyuta getirilir. Karşılaştırma tüm baytlar incelenene kadar veya bir eşitsizlik bulunana kadar devam eder. Koşul kodlarını (denetim sinyalleri) son karşılaştırma belirler. ”

RISC-V *ld* buyruğu bellekten bir çift sözcük okur ve register öbeğine yazar.

Sizce hangi mimari daha düzenli? Hangi komutun donanım uygulaması daha yalın?

3- Küçük olan hızlıdır.

Çok palslik işlemci: İşlemci veriyolu aşamalara bölündüğünde kritik yol küçüleceğinden saat frekansı artırılabilir.

İşlemci Tasarım İlkeleri

4- İyi tasarım bedel ödeme gerektirir.

Çek buyruğu: Çek buyruğu aynı anda iki registıra yazılmasını gerektirir. Bu sebepten **tasarımı karmaşıktırır**. Çek komutu aynı zamanda olağan durumu hızlandırarak **başarımı artırır**.

Boru Hattı (İş Hattı) Yöntemi

Pipe LineManagement

Boru Hattı - amařır rneęi

Boru Hattı yntemi kullanmadan amařır yıkamak iin:

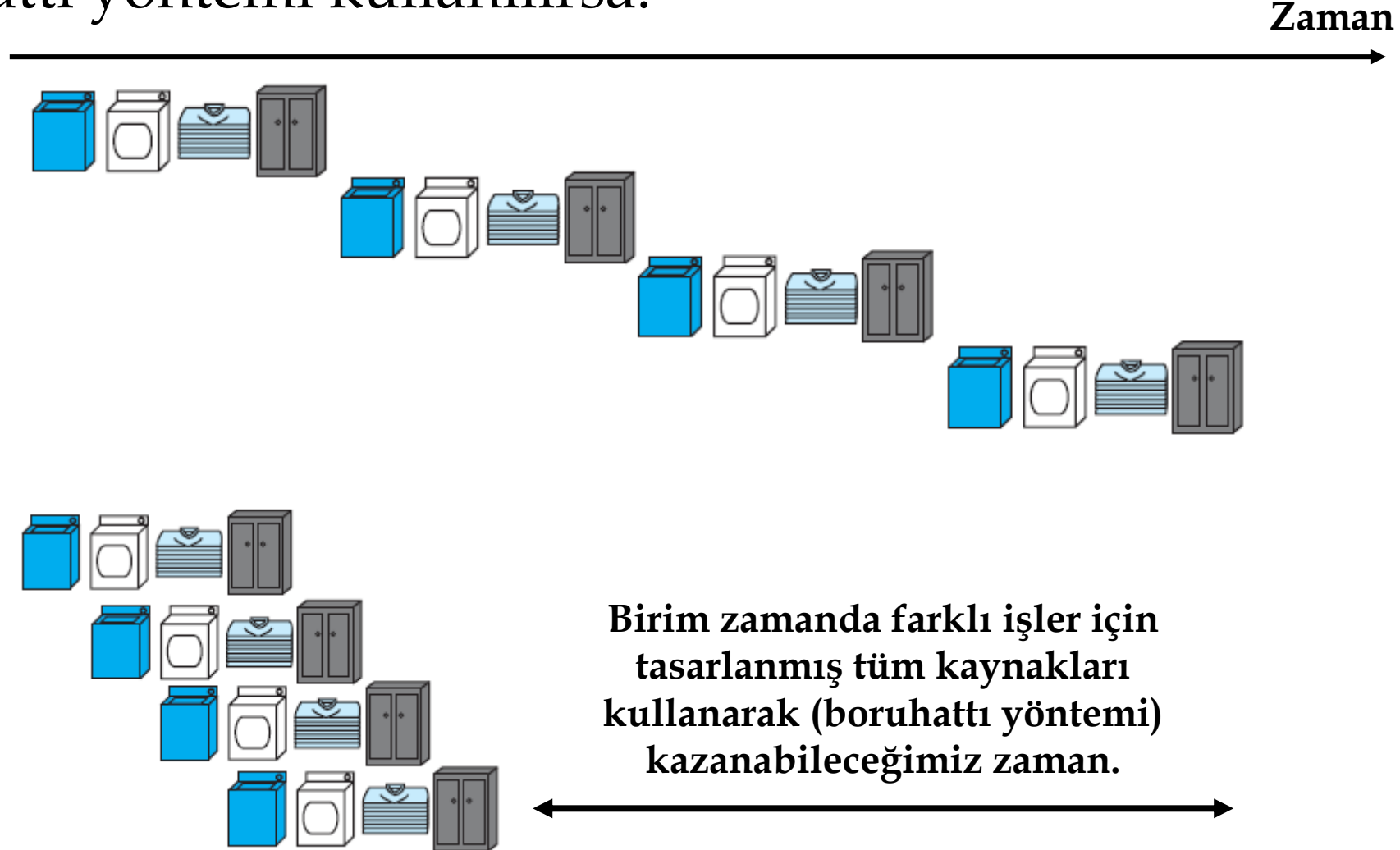
- 1- Kirli amařırlar amařır makinesine koyulur.
- 2- amařır makinesinin iři bitince amařırlar kurutucuya koyulur.
- 3- Kurutucunun iři bittięinde amařırlar katlanır.
- 4- Katlama iři bittięinde amařırlar dolaba koyulur.

Drt kez amařır yıkanması gerektięinde:



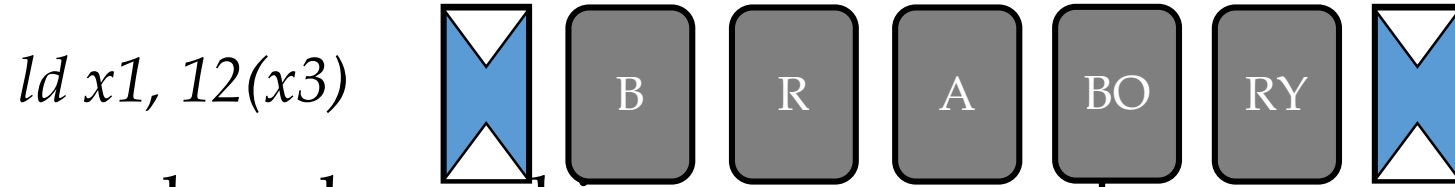
Boru Hattı - amařır rneęi

Boru Hattı yntemi kullanılırsa:



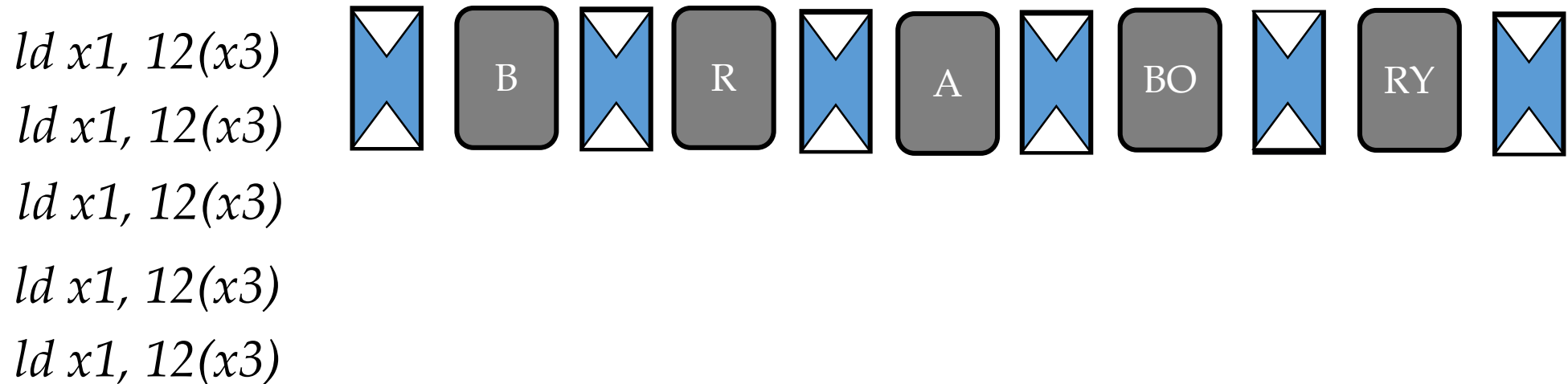
Boru Hattı

Çok palslik işlemcideki verimsizlik:



Birim zamanda yalnızca bir aşama meşgul.

Boru Hattı yöntemi uygulanırsa:



RISC-V İşlemcileri İçin Uygun Boru Hattı

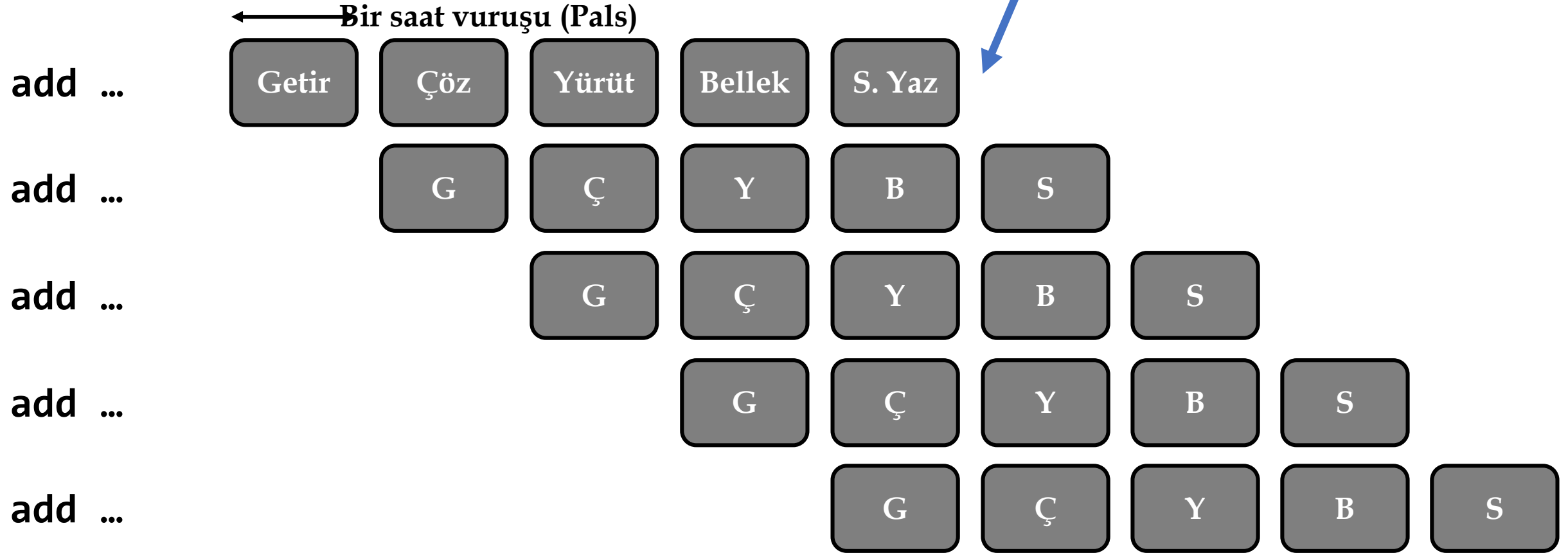
RISC-V komutları beş aşamada tamamlanır.

1. Komutları bellekten getir [**Getir**] [Fetch].
 2. Komutları çöz ve register değerlerini oku [**Çöz**] [Decode].
 3. İşlemi yürüt ya da adres hesapla [**Yürüt**] [Execute].
 4. Bellekteki bir işlenene eriş [**Bellek**] [Store].
 5. Register öbeğine sonucu yaz [**Sonucu Yaz**].
- Her zaman olmayabilir. Örn: **beq** buyruğu belleğe erişmez, register öbeğine yazmaz.

5 Aşamalı Boru Hattı

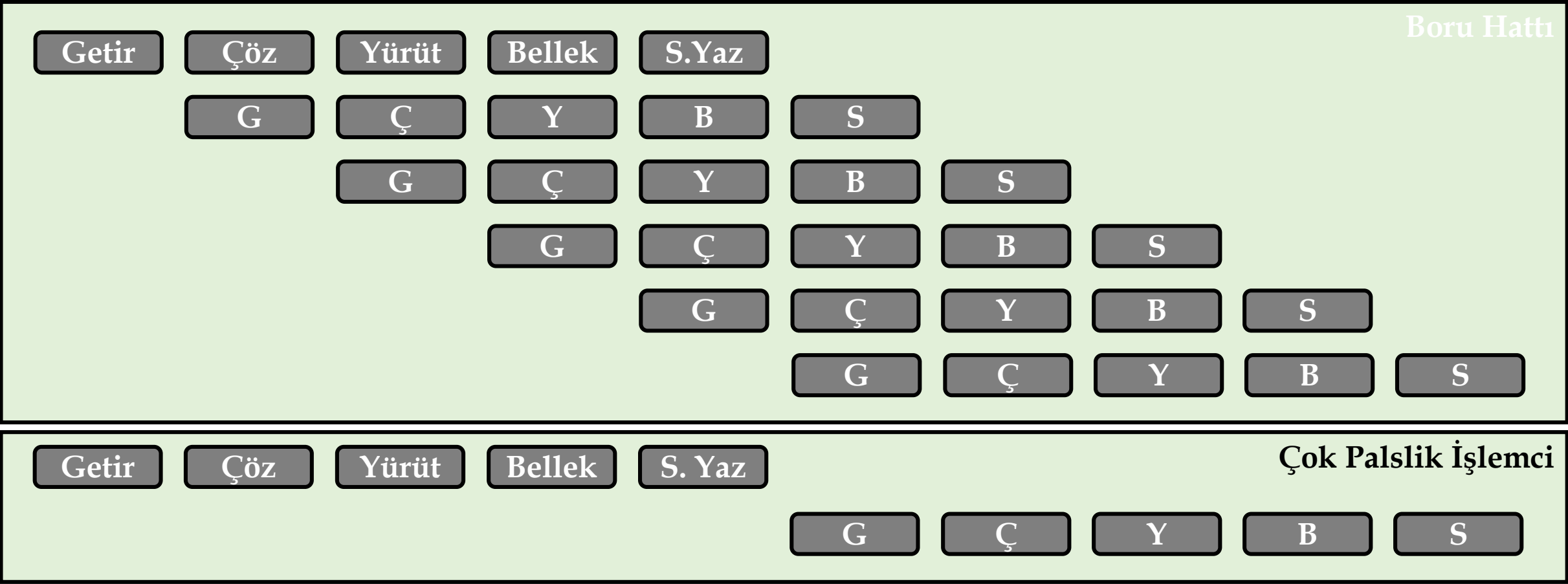
Boru Hattının
Dolma Zamanı

Zaman



**Tüm aşamalar bir
komut tarafından
kullanılıyor.**

Boru Hattı Yöntemi ve Çok Palslik İşlemci



$BBC_{BoruHattı} = 1/0$ $BBC_{BoruHattı} = 2/0$ $BBC_{BoruHattı} = 3/0$ $BBC_{BoruHattı} = 4/0$ $BBC_{BoruHattı} = 5/1$ $BBC_{BoruHattı} = 6/2$ $BBC_{BoruHattı} = 7/3$ $BBC_{BoruHattı} = 8/4$ $BBC_{BoruHattı} = 9/5$ $BBC_{BoruHattı} = 10/6 = 1.7$

$BBC_{ÇokVuruşlu} = 1/0$ $BBC_{ÇokVuruşlu} = 2/0$ $BBC_{ÇokVuruşlu} = 3/0$ $BBC_{ÇokVuruşlu} = 4/0$ $BBC_{ÇokVuruşlu} = 5/1$ $BBC_{ÇokVuruşlu} = 6/1$ $BBC_{ÇokVuruşlu} = 7/1$ $BBC_{ÇokVuruşlu} = 8/1$ $BBC_{ÇokVuruşlu} = 9/1$ $BBC_{ÇokVuruşlu} = 10/2 = 5$

Özet

- İşlemci Veriyolu
- İşlemci Denetimi
- Tek Vuruşluk İşlemciler
- Çok Vuruşluk İşlemciler
- İşlemci Tasarım İlkeleri
- Boru Hattı Yöntemine Giriş

Sonraki Ders

- Boru Hattı Başarım Hesabı
- Boru Hattında Oluşan Bağımlılıklar
 - Veri bağımlılıkları
 - Denetim bağımlılıkları