Bilgisayar Mimarileri

Boru Hattı Sorunları

Kitaptaki İlgili Bölümler:

- Computer Organization and Design: The Hardware Software Interface [RISC-V Edition] David A. Patterson, John L. Hennessy
 - 4. Bölüm (4.5 4.7)

Boru Hattı Yöntemi

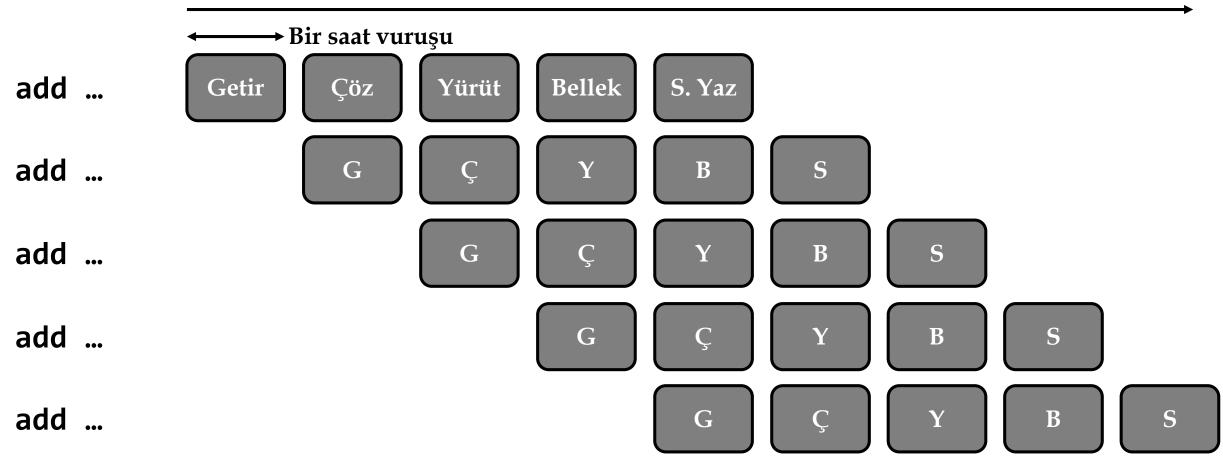
Boru Hattı

Ana fikir:

- 1. Buyruk yürütme döngüsünü farklı parçalara böl.
- 2. Her parçada bir buyruk çalıştırmak için yeterli sayıda kaynak ekle.
- 3. Her parçada farklı bir buyruk çalıştır.

Boru Hattı

Zaman



Tüm aşamalar ayrı bir buyruk tarafından kullanılıyor.

Pentium III-IV Boru Hattı

Basic Pentium III Processor Misprediction Pipeline

1 2 3 4 5 6 7 8 9 10
Fetch Fetch Decode Decode Decode Rename ROB Rd Rdy/Sch Dispatch Exec

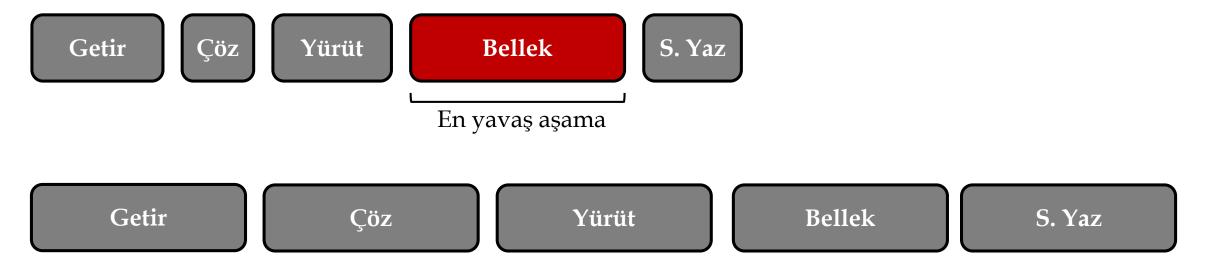
Basic Pentium 4 Processor Misprediction Pipeline

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
TC N	xt IP	TC F	etch	Drive	Alloc	Ren	ame	Que	Sch	Sch	Sch	Disp	Disp	RF	RF	Ex	Flgs	Br Ck	Drive	

Core i7 işlemcilerde ise 14 ile 20 aşama olduğu tahmin ediliyor.

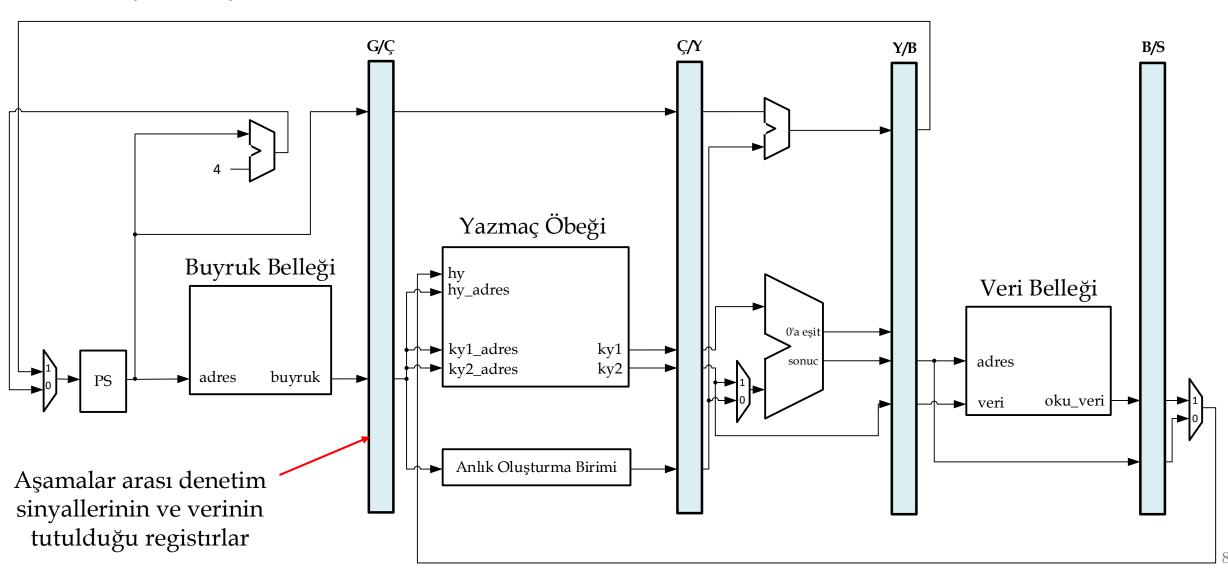
Boru Hattı Saat Vuruş Sıklığı

Boru hattı saat vuruş sıklığını en yavaş aşama belirler.

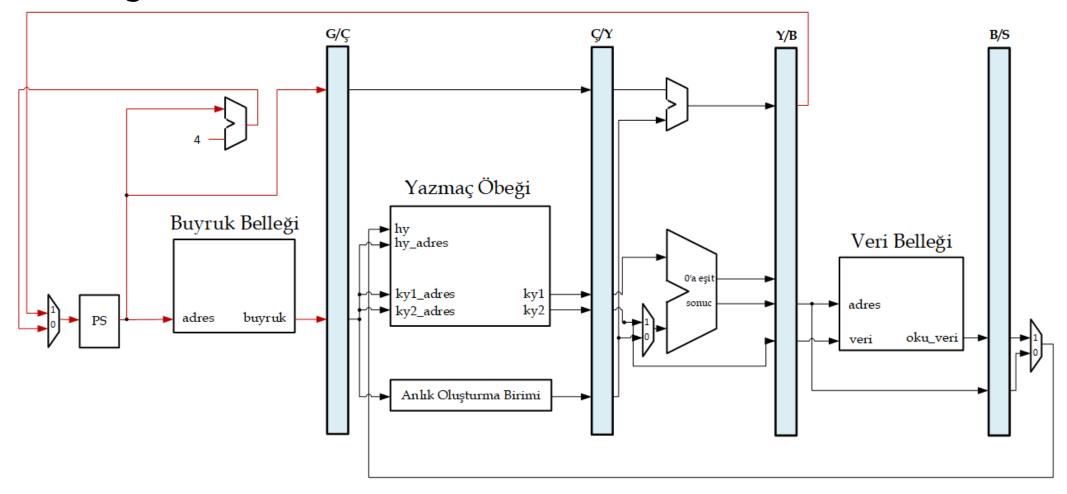


İşlemcimizin Boru Hattı Uygulayan Hali

*it/çek buyrukları yok.

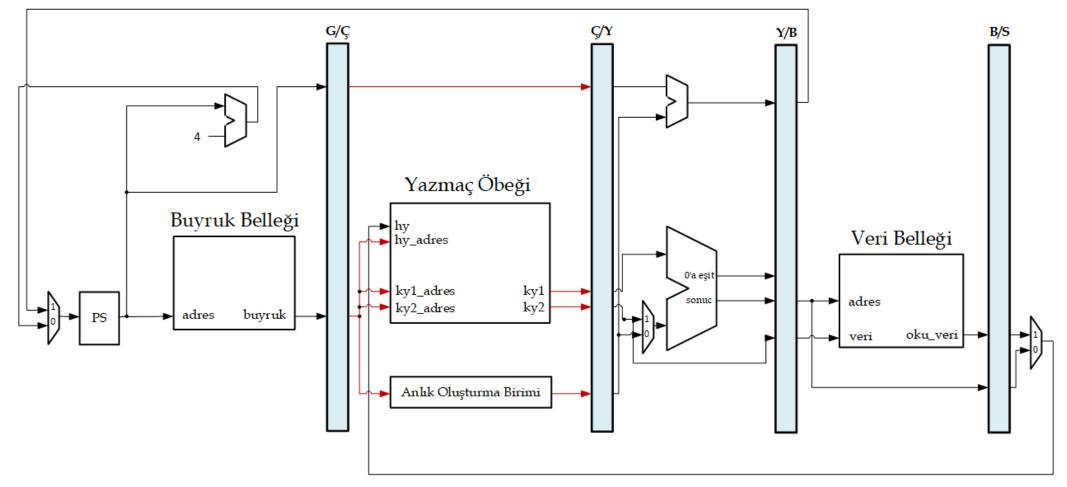


Getir Aşaması



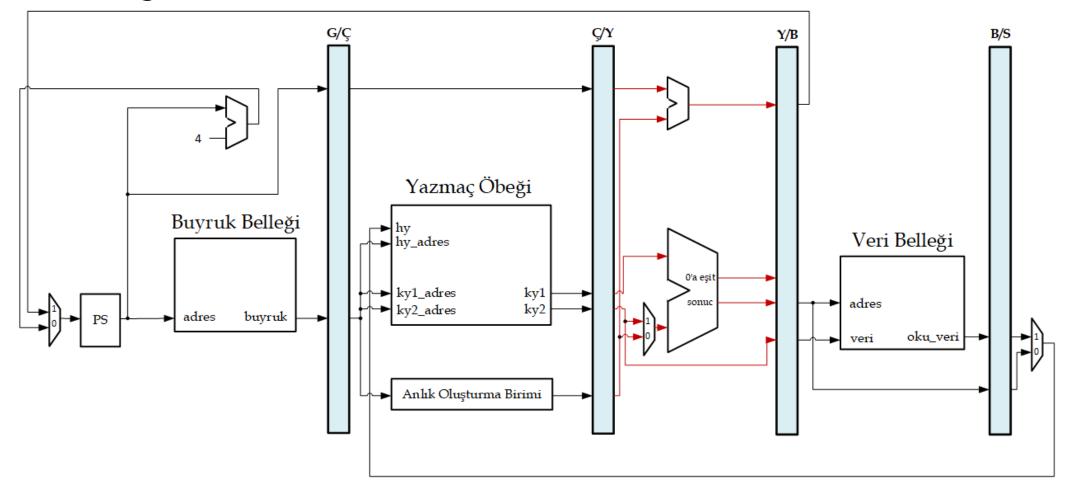
- Program sayacı belirlenmesi.
- Komut belleğine erişim.

Çöz Aşaması



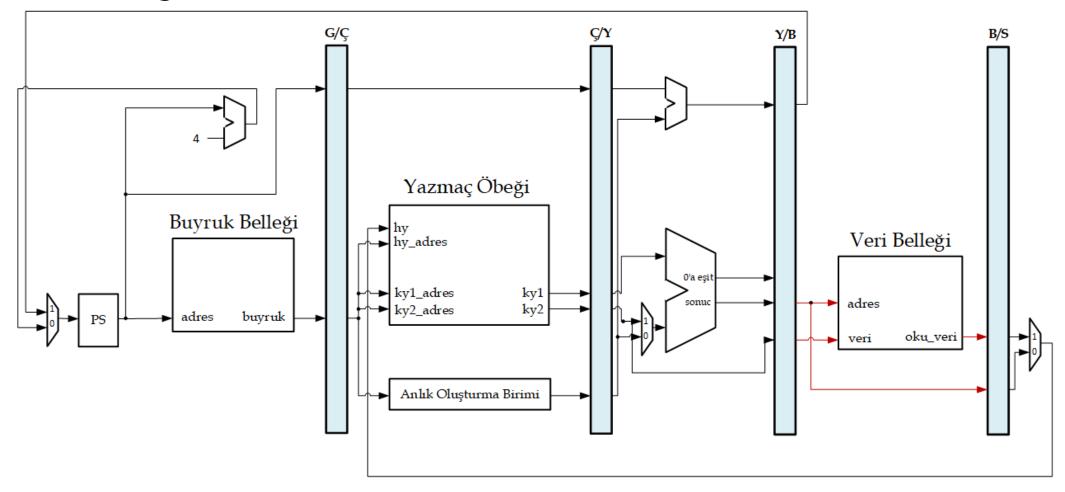
- Komutun çözülmesi.
- Register değerlerinin okunması.
- Anlık değerin oluşturulması.

Yürüt Aşaması



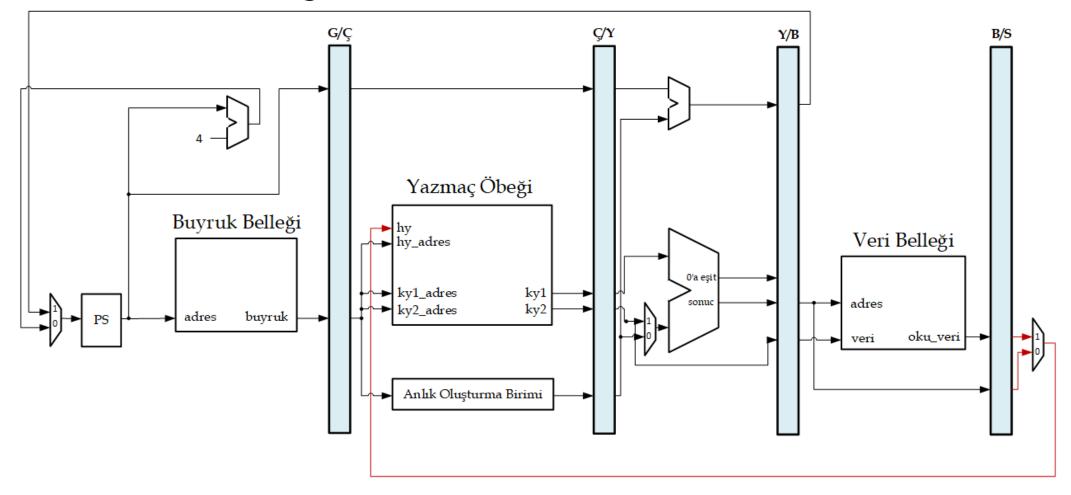
- Aritmetik işlemin yapılması.
- Program sayacının hesaplanması (dallanmalar).

Bellek Aşaması



• Veri belleğine erişim.

Sonucu Yaz Aşaması



• İşlem sonucunun register öbeğine yazılması.

Boru Hattında Oluşan Sorun

Boru Hattı Sorunu (-ing. pipeline hazard):

Boru hattında bir sonraki komutun bir sonraki saat vuruşunda yürütülemediği durum.

Üç farklı boru hattı sorunu vardır:

- 1. Yapı Sorunu (-ing. Structural Hazard)
- 2. Veri Sorunu (-ing. Data Hazard)
- 3. Denetim Sorunu (-ing. Control Hazard)

Yapı Sorunu

Yapı Sorunu (-ing. Structural Hazard):

Aynı anda tek kaynağa iki birimin birden erişmek istemesi nedeniyle ortaya çıkar. Kısacası bir saat vuruşunda yürütülmesi gereken komutlar için **yeterince donanım kaynağı** bulunmadığı durum. **Bu durumda iki komut da yürütülemez**.

Yapı Sorunu

Eğer RISC-V işlemcimizde tek bellek birimi olsaydı **komutlar ve veriler aynı** bellek biriminde bulunurdu.

• Getir aşaması ile Bellek aşaması arasında yapı sorunu olurdu.



Yapı Sorunu

Tek bellek birimi olan işlemcide **Getir** aşaması ile **Bellek** aşaması arasındaki **yapı sorunu** nasıl çözülebilir?

- Birleşik bellek modülüne birden fazla okuma yapabilecek şekilde yeni bağlantı noktaları ekleyerek.
 - Register öbeği gibi. Register öbeğinden aynı saat vuruşunda iki veri okuyabiliyoruz.
- Komutları ve verileri ayrı bellek birimlerine yerleştirerek.

Veri Sorunu

Veri Sorunu (-ing. Data Hazard):

Bir saat vuruşunda yürütülmesi gereken komutlar için **gereken verinin hazır olmadığı** durum. **Bu durumda komut yürütülemez**.

• Bir topla komutunun yazdığı registırı okuyan çıkar komutu gelirse:



topla'nın x19'a yazacağı burada anlaşılır. topla x19'a yazax19'a yazıldıktan bir için çıkar x19'u sonraki çöz okuyamaz aşamasında x19 okunur.

Veri Sorunu

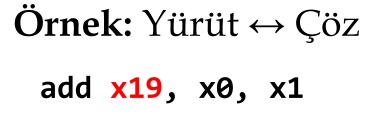
Veri sorunları nasıl çözülür?

- Yazılım: Derleyici veri bağımlılığı olmayacak şekilde kod üretmeye çalışabilir.
 - **Genellikle** iyi sonuçlar vermez. Çok aşamalı (20+) boru hattı olan bir işlemciye **veri sorunu olmayan kod üretmek zordur**.
- Donanım: Veri yönlendirmesi.
 - Ana fikir: Veri bağımlılıklarının çözümlenmesi için buyrukların tamamlanmasını beklememiz gerekmez. Bazı aşamaların sonunda hazır olan veri (Örn. yürüt aşamasında AMB'nin çıkışı) diğer aşamaların belli kısımlarına (Örn. çöz aşamasında işlenenlerin okunması) yönlendirilebilir.

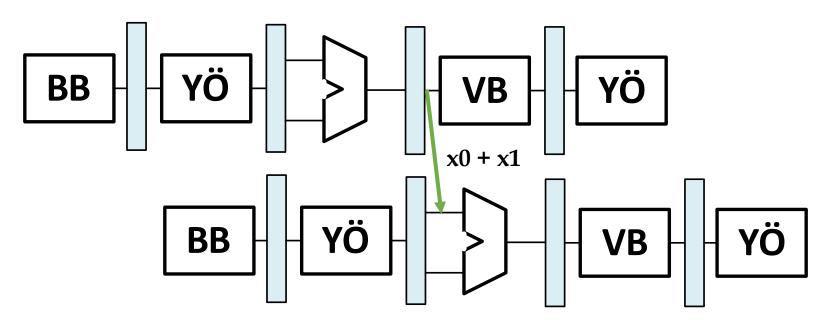
Veri Yönlendirmesi

Yürüt aşamasından **Çöz** aşamasına veri yönlendirmesi yapılırsa art arda gelen R-tipi buyruklar seri olarak yürütülebilirler.

BB: Buyruk Belleği, YÖ: Yazmaç Öbeği, VB: Veri Belleği

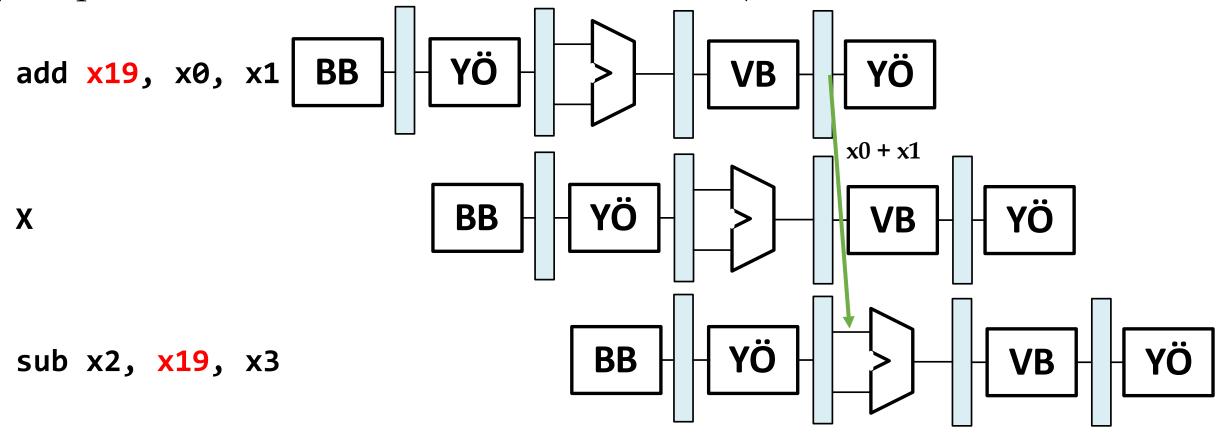


sub x2, x19, x3

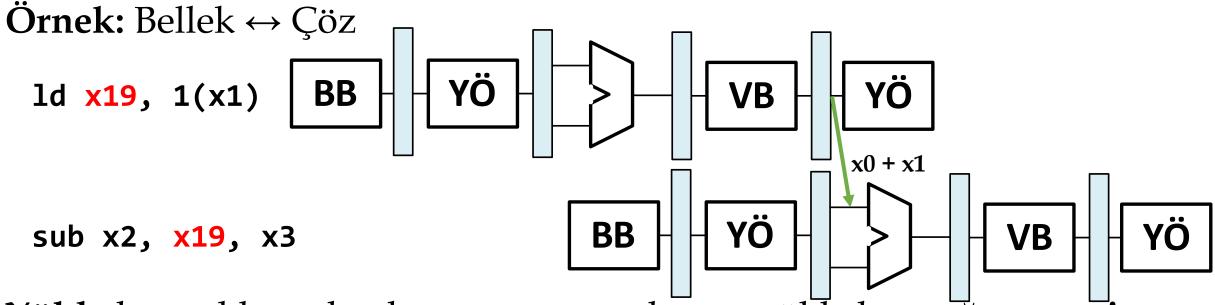


Veri Yönlendirmesi

Yürüt aşamasını geçen buyruğun hesapladığı veri de yönlendirilebilir (hesaplanan veri aşamalar ilerledikçe saklanır).



Veri Yönlendirmesi



Yükle buyruklarından hemen sonra gelen ve yükle buyruğuna veri bağımlılığı olan buyruklar yönlendirme olsa bile bir saat vuruşu geç yürütülürler (yükle-kullan (-ing. load-use) veri bağımlılığı).

- Çünkü: Bellek aşamasından bir sonraki buyruğun çöz aşamasına yönlendirme yapılamaz (zamanda geri gitmemiz gerekir).
- Bu durum **boru hattı duraklaması** (-ing. pipeline stall) olarak adlandırılır.

Denetim Sorunu

Denetim Sorunu (-ing. Control Hazard): İşlemciye getirilen buyruğun yürütülmesi gereken buyruk olmadığı durum. Bu durumda buyruk yürütülemez.

Dallanma buyrukları boru hattında denetim bağımlılıklarına sebep olur.

• Çünkü: Dallanma buyruklarının yönü (atlar-atlamaz) yürüt aşamasında belli olur.

```
1000:
bl x0, x1, #A BB
```

Karşılaştırma sonucu yürüt aşamasında belli olur

Denetim Sorunları Masraflıdır

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

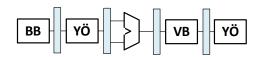
SPECint2006'deki **devingen buyrukların** %17'si dallanma buyruğudur. İşlemcinin dallanma buyrukları geldiğinde 2 saat vuruşu durakladığı durumda başarımı duraklamadığı durumdaki başarımına göre nasıl değişir?

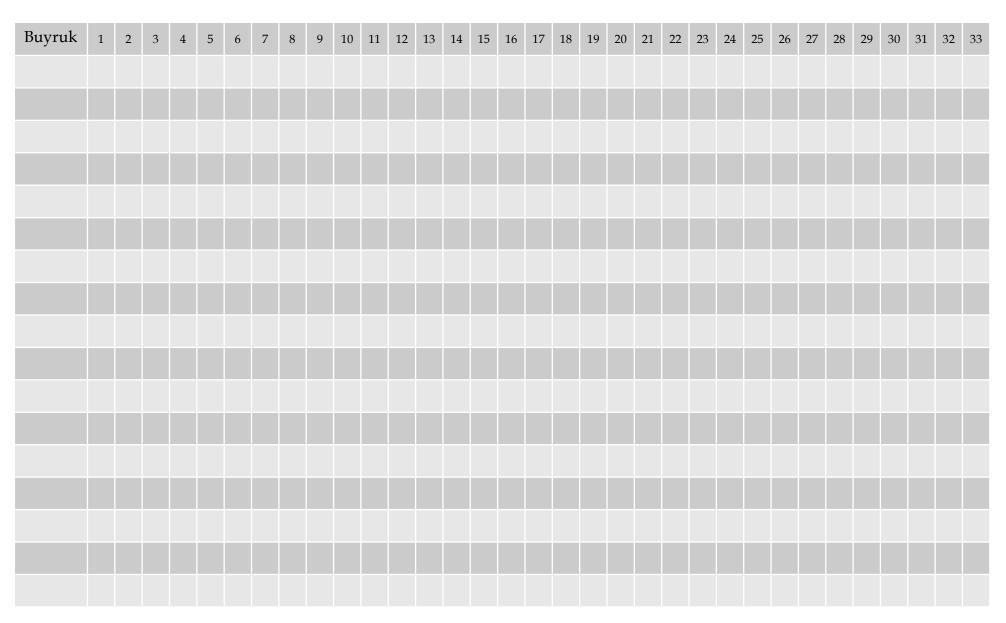
$$BB\zeta_{duraklama} \stackrel{\cong}{=} \frac{1}{17}$$

 $BB\zeta_{durakla} \stackrel{\cong}{=} 1 + 2 * \frac{17}{100} = 1,34$

Örnek

addi x9, x0, #0
addi x10, x0, #1
addi x11, x0, #A
lb x12, 0(x11)
sb x12, 1(x11)
addi x11, x11, #1
addi x9, x9, #1
ble x9, x10, #-16





Özet

- Boru Hattı Yöntemi
- Boru Hattı Bağımlılıkları
 - Yapısal Bağımlılıklar
 - Veri Bağımlılıkları
 - Denetim Bağımlılıkları

Sonraki Ders

• Dallanma Öngörüsü 🗲 Denetim Bağımlılıklarının Çözümü