



COMPUTER ARCHITECTURE 1ST MIDTERM
BİLGİSAYAR MİMARİSİ 1NCİ YILIÇI SINAVI

QUESTION 1: (45 Points)

a)

i. Always predict "not taken"

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
L2	FD	FO	EX	WR											
L3		FD	FO	EX	WR										
L4			FD	FO	EX	WR									
L5				FD	FO	EX	WR								
L6					FD	FO	-								
L7	Instruction after L6 also enter the pipeline; it is only fetched.														
L2								FD	FO	EX	WR				
L3									FD	FO	EX	WR			
L4										FD	FO	EX	WR		
L5											FD	FO	EX	WR	
L6												FD	FO	EX	WR

Misprediction detected

PC is updated.

The pipeline stalled and emptied.
L6 and L7

Branch penalty: 3 cycles.

10 Points

ii. Dynamic prediction with one bit, and the initial decision is to take the branch

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
L2	FD	FO	EX	WR											
L3		FD	FO	EX	WR										
L4			FD	FO	EX	WR									
L5				FD	FO	EX	WR								
L2					FD	FO	EX	WR							
L3						FD	FO	EX	WR						
L4							FD	FO	EX	WR					
L5								FD	FO	EX	WR				
L2									FD	FO	-				
L3										FD	FO	-			
L6												FD	FO	EX	WR

Address comes from BHT.

Prediction is correct.

Branch address has been already taken
from the branch history table. PC is not up-
dated again.

The pipeline is stalled
and emptied. L2 and L3

Misprediction detected

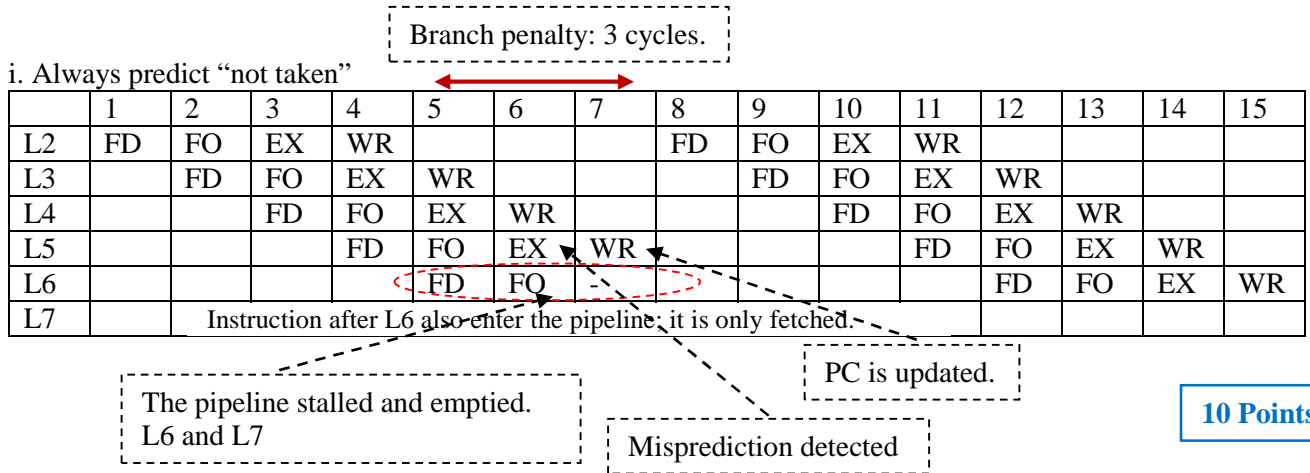
Branch penalty: 3 cycles.

10 Points

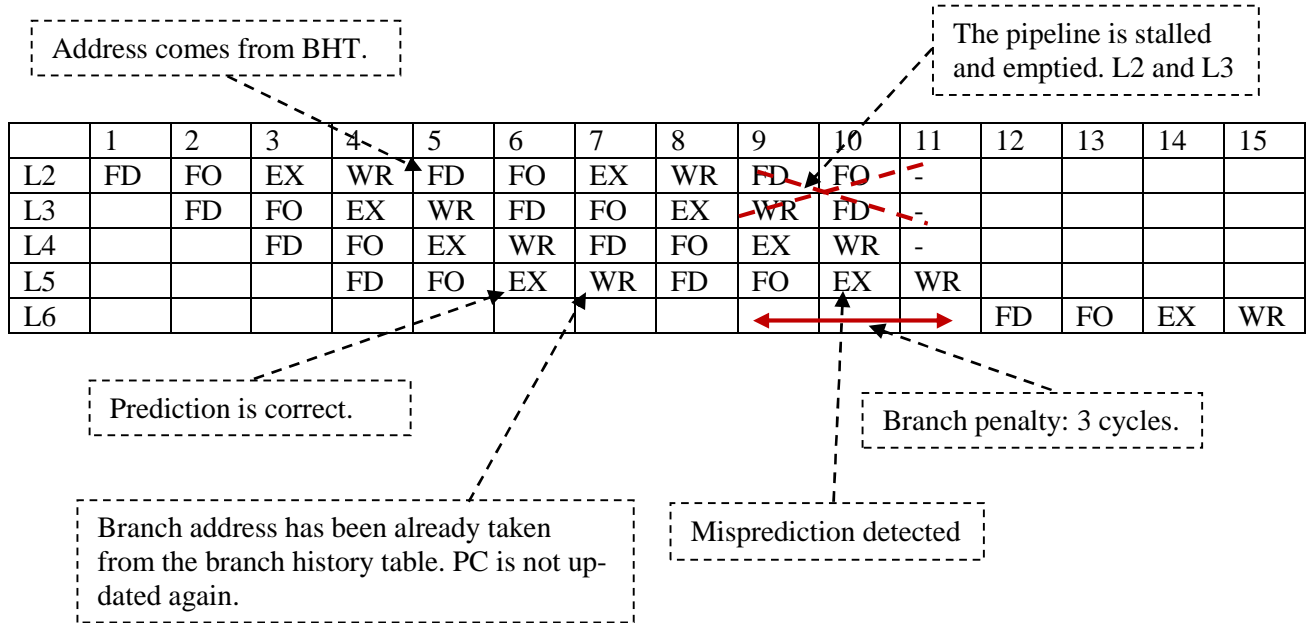
In this case (ii), at clock cycle 10 the pipeline control unit can detect that the branch will not be taken and it can take instruction L6 into the pipeline at clock cycle 11, because the address of the next instruction is in the PC. If the control unit had this capability the branch penalty would be only 2 clock cycles. Both answers are accepted.

*Bu durumda (ii) iş hattı denetim birimi 10. saat çevriminde dallanmanın olmayacağı anlaşıldıktan sonar 11. Saat çevriminde sonraki komutu (L6) iş hattına alabilir, çünkü bu komutun adresi PC'de yer almaktadır. Dene-
tim birimi bu şekilde tasarlandığıysa dallanma cezası sadece 2 saat çevrimi olur. Her iki yanıt da kabul edilmiştir.*

a) Same diagrams can be also drawn as shown below.
Aynı diyagramlar aşağıda gösterildiği gibi de çizilebilir.



ii. Dynamic prediction with one bit, and the initial decision is to take the branch



In this case (ii), at clock cycle 10 the pipeline control unit can detect that the branch will not be taken and it can take instruction L6 into the pipeline at clock cycle 11, because the address of the next instruction is in the PC. If the control unit had this capability the branch penalty would be only 2 clock cycles. Both answers are accepted.

Bu durumda (ii) iş hattı denetim birimi 10. saat çevriminde dallanmanın olmayacağı anlaşıldıktan sonar 11. Saat çevriminde sonraki komutu (L6) iş hattına alabilir, çünkü bu komutun adresi PC'de yer almaktadır. Denetim birimi bu şekilde tasarlandıysa dallanma cezası sadece 2 saat çevrimi olur. Her iki yanıt da kabul edilmiştir.

b) Branch penalties are 3 clock cycles for both cases as shown on the diagrams.

5 Points

c) There are 4 instruction in the loop. If the loop runs 10 times 40 instructions will run. After the loop instruction at line 6 will also run. So there are 41 instructions to run.

There are 4 stages in the pipeline. Total time for 41 instructions without any hazards: $4 + 40 = 44$ cycles.

One branch penalty = 3 cycles. Total time: $44 + 3 = 47$ clock cycles.

10 Points

d) Without the pipeline: $41 * 60$ ns. (41 instructions. Each instruction is completed in 60 ns)

With the pipeline: $47 * 20$ ns. (47 clock cycles. Period of clock is 20 ns)

Speed up = $41 * 60 / 47 * 20 = 41 * 3 / 47 \approx 2,6$

5 Points

e) Max. speed up = $60 / 20 = 3$

5 Points



İTÜ Bilgisayar Mühendisliği Bölümü
12.03.2015
Süre: 1 saat 45 dakika

Numara:
Ad Soyad:
İmza:

BİLGİSAYAR MİMARİSİ 1nci YILIÇI SINAV

SORU 2: (30 Puan)

Bir RISC MİB 3 katmanlı bir iş hattına sahiptir:

F: (Fetch and decode instruction) Komutu al ve çöz

A: (Operand (register) read, ALU operation, and write the result of ALU to the register) Operandları oku, ALU işlemini yap, sonucu saklayıcılara yaz

M: (Memory access if necessary (from register to memory or from memory to register)) Eğer gerekli ise bellek erişimi yap (saklayıcıdan belleğe veya bellekten saklayıcıya)

a) Yanda verilen programı inceleyerek tanıtılan iş hattının zaman diyagramını çizin.

Tüm veri bağımlılığı ve dallanma sorunlarını NOP komutu ekleyerek çözünüz.

b) Dallanma sorununu gidermek için iş hattının performansını düşürmeyen yazılım çözümlerini yazınız. Verilen programın algoritmasını değiştirmeyiniz.

```
...
LD 0(R10), R1 ; R1 <-- M[R10]
LD 0(R11), R2 ; R2 <-- M[R11]
ADD R1, R2, R3 ; R3 <-- R1+R2
ADD R0, R3, R4 ; R4 <-- R3
ST 0(R12), R3 ; M[R12] <-- R3
BA L2 ; Branch always
L1: ADD R0, 0, R4 ; R4 <-- 0
...
L2: ADD R0, 0, R3 ; R3 <-- 0
```

KOMUT KÜMESİ:

LD	X(Rs), Rd	$Rd \leftarrow M[Rs + X]$	Load
ST	X(Rs), Rd	$M[Rs + X] \leftarrow Rd$	Store
ADD	Ri, Rj, Rd	$Rd \leftarrow Ri + Rj$	
BA	Y	$PC \leftarrow PC + Y$	Branch always (relative)

Cözüm:

	1	2	3	4	5	6	7	8	9	10	11
1	LD 0(R10), R1	F	A	M							
2	LD 0(R11), R2		F	A	M						
3	ADD R1, R2, R3			F	A	M					
4	ADD R0, R3, R4				F	A	M				
5	ST 0(R12), R3					F	A	M			
6	BA L2						F	A	M		
7	L1: ADD R0, 0, R4							F	A	M	
8	...										
9	L2: ADD								F	A	M

*1 Veri bağımlılığı
R2 yeni değeri almamış
*2 Dallanma sorunu
R4 sıfırlanır oysa
Sıfırlanmaması lazım
NOP'lu çözüm

① 2 ile 3' satır arasında
NOP konulur
② 6 ile 7 satır arasında
NOP konulur

1. Çözüm
LD 0(R10), R1
LD 0(R11), R2
ADD R1, R2, R3
ADD R0, R3, R4
BA L2
ST 0(R12), R3
L1: ADD R0, 0, R4
L2: ADD R0, 0, R3

2. Çözüm
LD 0(R10), R1
LD 0(R11), R2
ADD R1, R2, R3
ADD R0, R3, R4
ST 0(R12), R3
BA L2
ADD R0, 0, R3
L1: ADD R0, 0, R4
L2: --- Sonradan komut

3. Çözüm (opsiyonel)
LD 0(R10), R1
LD 0(R11), R2
ADD R1, R2, R3
BA L3
ADD R0, R3, R4
L1: ADD R0, 0, R4
L3: ST 0(R12), R3
L2: ADD R0, 0, R3



İTÜ Bilgisayar Mühendisliği Bölümü
12.03.2015
Süre: 1 saat 45 dakika

Numara:
Ad Soyad:
İmza:

BİLGİSAYAR MİMARİSİ 1nci YILIÇI SINAV

SORU 3: (25 Puan)

Komut iş hatlarında oluşan sorunları çözmek için kullanılan donanım tabanlı ve yazılım (derleyici) tabanlı çözümlerin isimlerini aşağıdaki tablolarda uygun kutulara yazınız. Her kutuya sadece bir çözümün adını yazınız. Eğer böyle bir çözüm yoksa ilgili kutuya "YOK" yazınız.

Çözüm:

a) Operand bağımlılığı ile ilgili çözümler:

	Donanım Tabanlı	Yazılım (derleyici) Tabanlı
Performansı düşürmeyen	Operand forwarding (bypassing)	Changing the order of instructions if possible
Performansı düşüren	Pipe line Stall (Hardware Inter lock)	Using NOP instructions

b) Dallanma problemi ile ilgili çözümler:

	Donanım Tabanlı	Yazılım (derleyici) Tabanlı
Performansı düşürmeyen	Target Instruction Prefetch	Changing the order of instructions if possible
Performansı düşüren	Flush & Stall Branch Prediction	Using NOP instructions

c) Sadece donanım tabanlı çözümleri kısaca açıklayınız. (Please refer to lecture notes.)

Operand forwarding ALU'nun çıkışından girişine doğrudan bir bağlantı oluşturur. ALU'da oluşturulan sonuç hedef yarıya yazılacak aynı zamanda bir sonraki işlemde kullanılabileceği için ALU'nun girişine verilir.

Instruction Stall, (Donanım kilitleme)

Bu donanım komutları sıra ile izler. Veri bağımlılığına sebep olacak komutun iş hatına girmesini bir veya birkaç periyot geciktirir. Dallanmalar için benzer şekilde izleyen komutları bekletir.

Target Instruction Prefetch (Önceden komut alma)

Koşullu dallanmalar da hem dallanmayı izleyen komutlar, hem de dallanmanın hedefi komutlar (target instructions) aynı iş hatına alınıp, koşul gerçekleşmez ise izleyen komuttan geçileşirse hedef komuttan devam edilir.

Branch Prediction

Dallanma öngörüsü statik veya dinamik olabilir. Statikte her koşul gerçekleşmiş gibi davranılır. Dinamik öngöründe daha önceki dallanmaların yönüne göre öngörü yapılır. Bunun için dallanma hedef tablosu ve dallanma bayrağı (bayrakları) tutulur.