

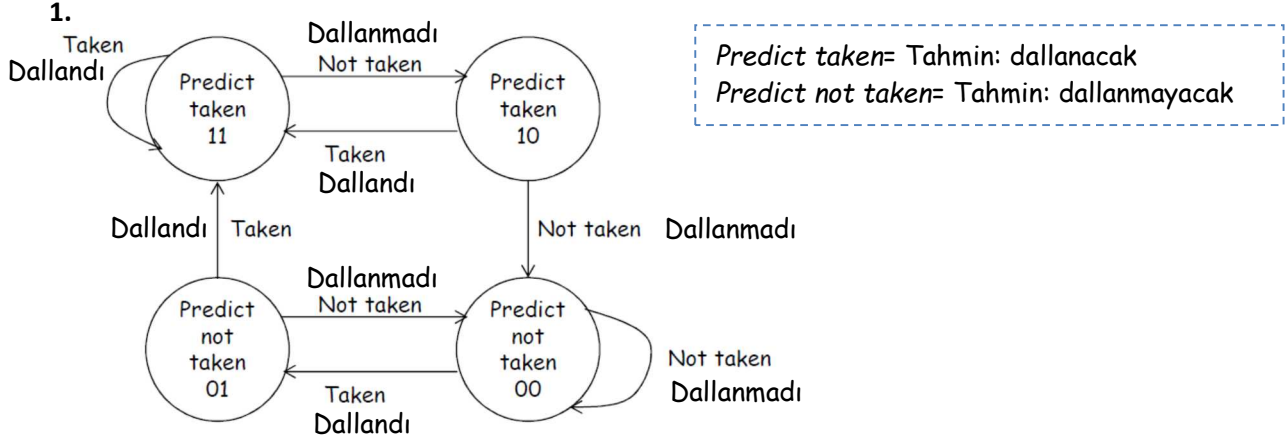


20.03.2014

COMPUTER ARCHITECTURE 1ST MIDTERM SOLUTIONS
BİLGİSAYAR MİMARİSİ 1nci YILIÇI SINAV ÇÖZÜMLERİ

QUESTION 1: (35 Points)

SORU 1: (35 Puan)



2. H : Hit / Tahmin doğru, P : Penalty / Ceza:

a.

i. T F T F T F T F
H P H P H P H P
4 x 2 = 8 clock cycles

ii. T T T T F F F F
H H H H P P P P
4 x 2 = 8 clock cycles

iii. T T T T T T T F
H H H H H H H P
1 x 2 = 2 clock cycles

b.

i. T F T F T F T F
P P P P P P P P
8 x 2 = 16 clock cycles

ii. T T T T F F F F
P H H H P H H H
2 x 2 = 4 clock cycles

iii. T T T T T T T F
P H H H H H H P
2 x 2 = 4 clock cycles

c.

i. T F T F T F T F
H P H P H P H P
4 x 2 = 8 clock cycles

ii. T T T T F F F F
H H H H P P H H
2 x 2 = 4 clock cycles

iii. T T T T T T T F
H H H H H H H P
1 x 2 = 2 clock cycles

QUESTION 2: (30 Points)**SORU 2: (30 Puan)****a) (5 points / 5 puan)**

	1	2	3	4
LD X(Rs), Ri	FD	OA	WR	ME
ADD Ri, Rj, Rk		FD	OA	WR

Memory access, Ri is loaded.
Bellek erişimi, Ri yüklenir.

Operand fetched. Ri is read.
Operand alma, Ri okunur.

The operand fetch operation (OA) must be performed after memory access (ME). Therefore, second instruction (ADD) must wait 2 clock cycles. Maximum penalty is 2 clock cycles.

Operand alma işlemi (OA) bellek erişiminden (ME) sonra yürütülmelidir. Bu nedenle ikinci komut (ADD) 2 saat çevrimi bekletilmelidir. En büyük ceza 2 saat çevrimidir.

b) (10 points / 10 puan)

Clock cycles (Adımlar)	1	2	3	4	5	6	7	8	9
Instructions (Komutlar)	1	2	3	4	5	6	7	8	9
LD 0(R10), R1	FD	OA	WR	ME					
LD 0(R11), R5		FD	OA	WR	ME				
ADD R1, R2, R3			FD	OA	WR				
ADD R0, R3, R4				FD	OA	WR			
ADD R5, R6, R7					FD	OA	WR		
BA NEXT						FD	OA	WR	
ADD R0, 0, R7							FD	OA	WR

Data conflict, R1.
Veri bağımlılığı, R1.

Data conflict, R3.
Veri bağımlılığı, R3.

PC is updated in WR. ADD instruction has been already fetched.
PS WR'de güncellenir. ADD komutu alınmış oldu.

Solving problems using NOP instructions.

Problemlerin NOP komutları ile çözülmesi.

```
LD 0(R10), R1      ; R1 <-- M[R10]]
LD 0(R11), R5      ; R5 <-- M[R11]]
NOP
ADD R1, R2, R3      ; R3 <-- R1+R2
NOP
ADD R0, R3, R4      ; R4 <-- R3
ADD R5, R6, R7      ; R7 <-- R5 + R6
BA NEXT            ; Branch always
NOP
NOP
ADD R0, 0, R7       ; R7 <-- 0
```

...

NEXT:

c) (15 points / 15 puan)

Solving the branch problem without decreasing the performance of the pipeline.

Dallanma komutu ile ilgili sorunun iş hatındaki performansı düşürmeden çözülmesi.

```

LD    0(R10), R1      ; R1 <-- M[R10])
LD    0(R11), R5      ; R5 <-- M[R11])
NOP
ADD   R1, R2, R3      ; R3 <-- R1+R2
BA    NEXT           ; Branch always
ADD   R0, R3, R4      ; R4 <-- R3
ADD   R5, R6, R7      ; R7 <-- R5 + R6
ADD   R0, 0, R7       ; R7 <-- 0

```

...

NEXT:

Moving 2 instruction under the branch instruction solves the branch problem and also one of the data conflicts.

İki komutu dallanma komutunun altına taşımak hem dallanma problemini hem de veri bağımlılığı problemlerinden birini çözmektedir.

QUESTION 3: (35 Points)**SORU 3: (35 Puan)****Asynchronous with handshaking**a) El sıkışmalı bellek adres aları

Birinci 4K (Tahve Gft) A00000 - A00FFF

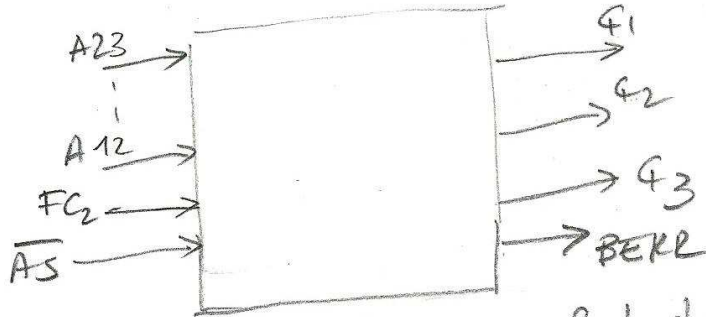
İkinci 4K (Tahve Gft) A01000 - A01FFF

Uyarmalı senkron bellek adres aları**Synchronous with strobe**

8K (Tahve Gft) B00000 - B01FFF

Kod gözütü Doğruluk Tablosu**Truth table of the address decoder**

\overline{AS}	F2	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	G1	G2	G3	BER
0	1	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0
0	1	1	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0
0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0
0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1

Kod Gözütü Seması**Block diagram of the address decoder**

Dövre seması izleyen sayfada dır

The circuit is in the next page.

b) 50ns

