



İTÜ Computer Engineering Department
16.03.2017
Assoc.Prof. Feza BUZLUCA
Asst. Prof. Sanem KABADAYI

COMPUTER ARCHITECTURE 1ST MIDTERM
BİLGİSAYAR MİMARİSİ 1nci YILIÇI SINAVI

QUESTION 1: (35 Points)

SORU 1:

a. (10p)

We construct the pipeline using digital combinational digital units of different types with propagation delays of 10ns, 15ns, and 20ns. The delay of each register is 5 ns. Hence, the minimum duration of the slowest segment can be $20+5=25$ ns. Therefore $t_{p(min)}=25$ ns.

Speedup for $n \rightarrow \infty$ **$S=t_n/t_p$ $S=100/25= 4$.**

İş hattı, yayılma gecikmeleri 10ns, 15ns, 20ns olan farklı tipte kombinezonsal sayısal elemanlar kullanılarak oluşturulmuştur. Her bir saklayıcının gecikmesi 5 ns'dir. Buna göre, en yavaş segmentin en düşük süresi $20+5=25$ ns. olabilir. Bu nedenle $t_{p(min)}=25$ ns.

*$n \rightarrow \infty$ için hızlanma **$S=t_n/t_p$ $S=100/25= 4$.***

b. (10p)

$T_2 = 150$ ns, $t_p=25$ ns. Remember; after the first element, on each clock cycle a new result is obtained at the output. The remaining 98 elements will be processed in $98 \times 25 = 2450$ ns. Total time for 100 elements:

$$T_{100} = 150 + 98 \times 25 = 150 + 2450 = 2600 \text{ ns}$$

Alternative solution: It is not necessary but you can also calculate the number of layers k first.

$$T_2 = (k+2-1) \times 25 = 150 \text{ ns} \rightarrow k=5 \rightarrow T_{100} = (5+100-1) \times 25 = 2600 \text{ ns}$$

$T_2 = 150$ ns, $t_p=25$ ns. Hatırlatma; ilk elemandan sonar her sat darbesinde çıkışta yeni bir sonuç oluşur. Kalan 98 elemanın işlenmesi $98 \times 25 = 2450$ ns sürer. 100 eleman için toplam süre:

$$T_{100} = 150 + 98 \times 25 = 150 + 2450 = 2600 \text{ ns}$$

Diğer bir çözüm: Gerekli olmamakla beraber, önce katman sayısı k hesaplanabilir.

$$T_2 = (k+2-1) \times 25 = 150 \text{ ns} \rightarrow k=5 \rightarrow T_{100} = (5+100-1) \times 25 = 2600 \text{ ns}$$

c. (15p)

Since both pipelines are designed using same type of digital units, the period of the clock cycle t_p for the pipeline P_B cannot be shorter than 25 ns. When the maximum speed of clock signals are same, **we prefer P_A** because

- **Duration for T_1** is longer in P_B .
- Because of extra registers and connections, **cost** and **power** consumption of P_B are higher.

*Her iki iş hattı da aynı tipte elemanlar kullanılarak tasarlandığından P_B iş hattının saat işareti periyodu t_p 25 ns'den daha kısa olamaz. Saat işaretlerinin maksimum hızları aynı olması durumunda **P_A iş hattı tercih edilir, çünkü***

- **T_1 süresi P_B iş hattında daha uzundur.**
- **Ek saklayıcılar nedeniyle P_B iş hattının maliyeti ve güç tüketimi daha yüksektir.**

QUESTION 2: (30 Points)

SORU 2: (30 Puan)

a) **Note:** Prediction mechanisms are applied only to conditional branch instructions. For an unconditional branch instruction (BRA), which branches always, a prediction is not necessary.

Dikkat: Öngörü yöntemleri sadece koşullu dallanma komutlarında uygulanırlar. Her zaman dallanan koşulsuz dallanma (BRA) komutunda zaten öngörü gereksiz ve anlamsızdır.

BRZ OUT: If the initial decision is to take the branch ($p = 1$), there will be a misprediction in the first iteration. Then, p will be changed to 0. In the last iteration, there will be a misprediction because $p = 0$. There are mispredictions in the first and last iterations. Other predictions are correct.

İlk ve son yinelemelerde yanlış öngörü olur; diğer öngörüler doğrudur.

Correct (Doğru): 8

Incorrect (yanlış): 2

BRZ OUT: If the initial decision is not to take the branch ($p = 0$), there will be a misprediction only in the last iteration. Other predictions are correct.

Sadece son yinelemede yanlış öngörü olur; diğer öngörüler doğrudur.

Correct (Doğru): 9

Incorrect (yanlış): 1

CONCLUSION: We would prefer “not taking the branch” ($p = 0$). The number of mispredictions is 1.

SONUC: Başlangıç durumu olarak “dallanma yok” ($p = 0$) tercih edilir. Hata sayısı 1 olur.

b)

BRZ OUT: If the initial value of the prediction bits is 11 (possibility to take the branch is VERY high), there will be mispredictions in the first and second iterations, because the prediction is changed after two mispredictions. There will also be a misprediction in the last iteration. Other predictions are correct.

Birinci ve ikinci yinelemelerde yanlış öngörü olur, çünkü bu yöntemde karar iki hatadan sonra değişir. Ek olarak son yinelemede de yanlış öngörü olur; diğer öngörüler doğrudur.

Correct (Doğru): 7

Incorrect (yanlış): 3

BRZ OUT: If the initial value of the prediction bits is 10 (possibility to take the branch is high), there will be mispredictions in the first iteration. There will also be a misprediction in the last iteration. Other predictions are correct.

Birinci ve son yinelemelerde yanlış öngörü olur; diğer öngörüler doğrudur.

Correct (Doğru): 8

Incorrect (yanlış): 2

BRZ OUT: If the initial decision is not to take the branch (00 or 01), there will be a misprediction only in the last iteration. Other predictions are correct.

Sadece son yinelemede yanlış öngörü olur; diğer öngörüler doğrudur.

Correct (Doğru): 9

Incorrect (yanlış): 1

CONCLUSION: We would prefer “not taking the branch” (00 or 01). The number of incorrect predictions is 1.

SONUC: Başlangıç durumu olarak “dallanma yok” (00 veya 01) tercih edilir. Hata sayısı 1 olur.

QUESTION 3: (35 Points)

SORU 3:

a. (15 p)

| | | | | | | | | | | | | | | | | | | | |
|---------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R1 <-- 10 (Counter) | F | O | A | W | | | | | | | | | | | | | | | |
| R2 <-- R3 | | F | O | A | W | | | | | | | | | | | | | | |
| R2 <-- R2 - 1 | | | | | | F | O | A | W | | | | | | | | | | |
| R3 <-- R3 + R4 | | | | | | | F | O | A | W | | | | | | | | | |
| R4 <-- R4 + 1 | | | | | | | | F | O | A | W | | | | | | | | |
| R1 <-- R1 - 1 | | | | | | | | F | O | A | W | | | | | | | | |
| Branch if not zero | | | | | | | | | | | | F | O | A | W | | | | |
| LOOP | | | | | | | | | | | | | | | | F | O | A | W |

| | | | |
|-------|-----|-------------|-----------------------|
| ... | ADD | R0, #10, R1 | ; R1 <-- 10 (Counter) |
| LOOP: | ADD | R0, R3, R2 | ; R2 <-- R3 |
| | SUB | R2, #1, R2 | ; R2 <-- R2 - 1 |
| | NOP | | |
| | NOP | | |
| | ADD | R3, R4, R3 | ; R3 <-- R3 + R4 |
| | ADD | R4, #1, R4 | ; R4 <-- R4 + 1 |
| | SUB | R1, #1, R1 | ; R1 <-- R1 - 1 |
| | NOP | | |
| | NOP | | |
| | BNZ | LOOP | ; Branch if not zero |
| | NOP | | |
| | NOP | | |
| | NOP | | |
| | ADD | R0, 0, R2 | ; R2 <-- 0 |
| | ADD | R0, 0, R3 | ; R3 <-- 0 |
| | ADD | R0, 0, R4 | ; R4 <-- 0 |

Reason for NOP instructions between R1-1 and BNZ:
The operation R1-1 modifies the flags in the condition code register (CCR) and determines the condition for the instruction BNZ.

The CCR is also modified (written to) in the W layer. The instruction BNZ will get this value in the O layer. Therefore, operand fetch operation of the instruction BNZ must come after the write operation of the instruction SUB R1, #1, R1.

R1-1 işlemi ile BNZ: komutu arasındaki NOP komutlarının gerekliliği:

R1-1 işlemi durum saklayıcısındaki (CCR) bayrakları değiştirir ve BNZ komutunun koşulunu belirler.

CCR'ye de W katmanında yazılır. BNZ komutu bu değeri O katmanında alır. Bu nedenle, BNZ komutunun operand alma işlemi, SUB R1, #1, R1 komutunun yazma işleminden sonra olmalıdır.

b) (20 p)

There are different possible solutions. **The penalty is at least 3 clock cycles.**

Farklı olası çözümler mümkündür. Ceza en az 3 saat çevrimi olur.

One of the possible solutions:

Olası çözümlerden biri:

| | | | |
|-------|-----|-------------|-----------------------|
| ... | ADD | R0, #10, R1 | ; R1 <-- 10 (Counter) |
| LOOP: | ADD | R0, R3, R2 | ; R2 <-- R3 |
| | ADD | R3, R4, R3 | ; R3 <-- R3 + R4 |
| | SUB | R1, #1, R1 | ; R1 <-- R1 - 1 |
| | ADD | R4, #1, R4 | ; R4 <-- R4 + 1 |
| | SUB | R2, #1, R2 | ; R2 <-- R2 - 1 |
| | BNZ | LOOP | ; Branch if not zero |
| | NOP | | |
| | NOP | | |
| | NOP | | |
| ... | | | |

Some of the common mistakes:

- You cannot move R1-1 under the BNZ.
- There must be at least 2 instructions between R1-1 and ADD R0, #10, R1. Otherwise, there is another conflict on R1.
- You cannot move ADD R0, 0, R2 or other instructions after the BNZ into the loop.

Bazı sık yapılan hatalar:

- R1-1 komutu, BNZ komutuna altına taşınamaz.
- R1-1 ile ADD R0, #10, R1 arasında en az iki komut olmalı, yoksa R1 nedeniyle yeni bir çatışma oluşur.
- ADD R0, 0, R2 veya BNZ komutundan sonra gelen diğer komutlar döngünün içine taşınamaz.