Εργασία στο Μάθημα “Αρχιτεκτονική Υπολογιστών

2022-2023

Γεώργιος Δάλλας
ΑΕΜ:4116

imm1 = 66 imm2 = 1

Οπότε το πρόγραμμα είναι το εξής:

```
        addi $2, $zero, 66
       add $1, $zero, $zero
Loop: lw $4, 400($1)
        lw $5, 800($1)
        xor $5, $5, $4
        sw $5, 600($2)
        addi $1, $1, 4
        subi $2, $2, 1
        bne $2, $zero, Loop
```

Ζητούμενα:

1. **Πιθανοί κίνδυνοι:**

   a. Η εντολή **addi $1, $zero, $zero** πρέπει να ολοκληρώσει το στάδιο WB ώστε να χρησιμοποιηθεί παρακάτω από την εντολή **lw $4, 400($1), η $1** στη **πρώτη επανάληψη**. Η λύση είναι να κάνει stall (φυσαλίδα) μέχρι η εντολή **addi $1, $zero, $zero** να φτάσει στο στάδιο WB.

   b. Η εντολή **lw $5, 800($1) μέσα στο loop** δεν μπορεί να διαβάσει τον $1, μέχρι η **lw $4, 400($1)** να φτάσει στο στάδιο WB. Η λύση και εδώ είναι το stall (φυσαλίδες), μέχρι η ανάγνωση του καταχωριτή να είναι διαθέσιμη.

   c. Η εντολή **xor $5, $5, $4 μέσα στο loop** δεν μπορεί να έχει πρόσβαση στον $5 μέχρι η **lw $5, 800($1)** να φτάσει στο στάδιο WB και να καταγράψει την νέα τιμή του $5. Ακριβώς όπως πάνω η λύση είναι το stall (φυσαλίδες), μέχρι η ανάγνωση του καταχωριτή να είναι διαθέσιμη.

   d. Η **sw $5, 600($2) μέσα στο loop** δεν μπορεί να έχει πρόσβαση στον $5 μέχρι η εντολή **xor $5, $5, $4** να φτάσει στο στάδιο WB. Η λύση για άλλη μια φορά είναι η δημιουργία φυσαλίδων.

   e. Η **bne $2, $zero, Loop μέσα στο loop** δεν μπορεί να χρησιμοποιήσει τον $2 μέχρι η **subi $2, $2, 1** που τον ορίζει να φτάσει στο σημείο WB. Εώς τότε κάνει stall(φυσαλίδες).

   Στο διάγραμμα χρονισμού απεικονίζονται με Φ οι φυσαλίδες που θα δημιουργούνται κάθε φορά που φτάνουν σε αυτή την εντολή, με Φ οι φυσαλίδες που ισχύουν μόνο την πρώτη φορά (στον κύκλο 5 με 7), ενώ με Φ οι φυσαλίδες που δημιουργούνται από την εντολή άλματος υπό συνθήκη.

   Το διάγραμμα χρονισμού είναι το παρακάτω:

2.
Η επανάληψη θα γίνει 66 φορές.
Η πρώτη προσπέλαση θα έχει 3 κύκλους καθυστέρηση (Φ), και οι υπόλοιπες 12 (Φ) αν δεν λάβουμε υπόψιν μας την ειδική λειτουργία των εντολών αλμάτων υπό συνθήκη.

Κάθε φορά που η λούπα φτάνει στην εντολή **bne $2, $zero, Loop** , εντοπίζει εντολή άλματος υπό συνθήκη (στο στάδιο ID), και ο επεξεργαστής κάνει stall την σωλήνωση μέχρι και το σημείοο EX της εντολής. Άρα κάθε φορά που ισχύει η συνθήκη προστίθενται stall για **3 κύκλους**. Άρα θα προστεθούν **65 φορές** (Αφού στην πρώτη προσπέλαση της λούπας δεν γίνεται προσθήκη).
Εν τέλει, έχουμε:
Κύκλοι = **2*6**(εντολές εκτός λούπας * στάδια) **+23*66 + 3** (3 κύλοι stall στην πρώτη προσπέλαση) **+ 65*3** (66 επαναλήψεις αλλά στην πρώτη δεν προστίθεται η καθυστέρηση της ειδικής λειτουργία της εντολής άλματος υπό συνθήκη) **+1** (η τελευταία εντολή bne θα πάρει έναν ακόμη κύκλο)
= 1729