2η Άσκηση στην Αρχιτεκτονική Υπολογιστών
Pipelined Datapath

Ανδρέας Στάμος
Αριθμός Μητρώου: *03120****

Δεκέμβριος 2022

Περιεχόμενα

1 Ερώτημα 1

Στον pipelined MIPS οι κίνδυνοι δεδομένων που υπάρχουν είναι μόνο το Read after Write. Οι υπόλοιποι δεν μπορούν να προκύψουν αφού Write υπάρχει μόνο στο WB και το Write γίνεται πάντα μετά το Read (που γίνεται στο ID). Για το Read after Write ο pipelined MIPS το αντιμετωπίζει είτε με προωθήσεις είτε σταματώντας την εκτέλεση εντολών ή/και εισάγοντας καθυστερήσεις. Υπάρχουν επιπλέον και κίνδυνοι ελέγχου όπου η τιμή του PC δεν έχει καθοριστεί ακόμα. Η παρούσα αρχιτεκτονική επιλέγει απλά να καθυστερήσει μέχρι το στάδιο MEM, όπου πλέον έχει υπολογιστεί η νέα τιμή του PC. Σημειώνεται ότι, το να εισάγαμε τις 2 επόμενες εντολές στο pipeline (υποθετώντας ότι δεν θα γίνει branch) θα είχε πολύ μικρό κόστος, αφού η μόνη διαφορά θα ήταν πως αν η διακλάδωση είναι να γίνει θα πρέπει να μηδενιστούν οι καταχωρητές if/id και id/ex. Και για το stall σήματα που πάνε προς αυτούς, ο μηδενισμός της επίτρεψης εγγραφής, οπότε το υλικό δεν θα κόστιζε σημαντικά παραπάνω για την πρόβλεψη μη λήψης διακλάδωσης.

No\cc	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
1	IF	ID	EX	MEM	WB																					
2		IF	ID	ID	ID	EX	MEM	WB																		
3			IF	IF	IF	ID	ID	ID	EX	MEM	WB															
4						IF	IF	IF	ID	EX	MEM	WB														
5									IF	ID	ID	ID	EX	MEM	WB											
6										IF	IF	IF	ID	ID	ID	EX	MEM	WB								
7													IF	IF	IF	ID	EX	MEM	WB							
8																IF	ID	EX	MEM	WB						
9																	IF	ID	ID	ID	EX	MEM	WB			
next																						IF	ID	EX	MEM	WB

Σχήμα 1: Διάγραμμα χρονισμού για pipelined MIPS 5 σταδίων χωρίς προώθηση. Με κόκκινο χρώμα σημειώνονται τα στάδια όπου γίνεται stall. (υπενθυμίζεται ότι stall συμβαίνει θέτοντας την επίτρεψη εγγραφής των ενδιάμεσων καταχωρητών και του PC σε 0, οπότε επαναλαμβάνεται ξανά το ίδιο στάδιο)

Το διάγραμμα χρονισμού φαίνεται στην εικόνα 1. Παρατηρούμε ότι απαιτούνται 21cc για κάθε επανάληψη του βρόχου. Επιπλέον στο τέλος του βρόχου απαιτούνται 2cc ακόμα προκειμένου να ολοκληρωθεί και η τελευταία εντολή του βρόχου. Σημειώνεται, όμως, πως στην πραγματικότητα αυτοί οι 2cc στο τέλος του βρόχου, στην

πραγματικότητα δεν αντιστοιχούν σε πραγματική καθυστερήση, καθώς δεν εισάγουν κάμια εξάρτηση δεδομένων, οπότε θα μπορούσε ταυτόχρονα να εκτελείται στο pipeline οποιαδήποτε επόμενη εντολή προβλέπει το υπόλοιπο πρόγραμμα ή το λειτουργικό σύστημα.

Αφου αρχικά $\$t9 = 0x400$ και $\$t9- = 0x4$ ανά βρόχο αυτό σημάινει ότι θα γίνουν $n = 0x100 = 256$ επαναλήψεις. Συνεπώς απαιτούνται συνολικά $256 \cdot 21 + 2 = 5378cc$ για την εκτέλεση του βρόχου.

2 Ερώτημα 2

No\cc	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	IF	ID	EX	MEM	WB														
2		IF	IF	ID	EX	MEM	WB												
3				IF	ID	ID	EX	MEM	WB										
4					IF	IF	ID	EX	MEM	WB									
5							IF	ID	ID	EX	MEM	WB							
6								IF	IF	ID	EX	MEM	WB						
7										IF	ID	EX	MEM	WB					
8											IF	ID	EX	MEM	WB				
9												IF	ID	EX	MEM	WB			
next															IF	ID	EX	MEM	WB

Σχήμα 2: Διάγραμμα χρονισμού για pipelined MIPS 5 σταδίων με όλες τις δυνατές προωθήσεις. Με κόκκινο χρώμα σημειώνονται τα στάδια όπου γίνεται stall. (υπενθυμίζεται ότι stall συμβαίνει θέτοντας την επίτρεψη εγγραφής των ενδιάμεσων καταχωρητών και του PC σε 0, οπότε επαναλαμβάνεται ξανά το ίδιο στάδιο). Επίσης με βελάκι σημειώνονται οι προωθήσεις που συμβαίνουν.

Το διάγραμμα χρονισμού φαίνεται στην εικόνα 2. Παρατηρούμε ότι απαιτούνται 14cc για κάθε επανάληψη του βρόχου. Επιπλέον στο τέλος του βρόχου απαιτούνται 2cc ακόμα προκειμένου να ολοκληρωθεί και η τελευταία εντολή του βρόχου.

Γίνονται, όπως εξήγηθηκε πριν, $n = 0x100 = 256$ επαναλήψεις. Συνεπώς απαιτούνται συνολικά $256 \cdot 14 + 2 = 3586cc$ για την εκτέλεση του βρόχου.

3 Ερώτημα 3

Ο κύκλος πρέπει να διάρκει (τουλάχιστον) όσο το βραδύτερο στάδιο του pipeline, δηλαδή πρέπει να διαρκεί $500 + 20 = 520ps$. Αυτό ισοδυναμεί με συχνότητα ρολογιού $f = 1.923GHz$.

4 Ερώτημα 4

Εφόσον η διάρκεια του ρολογίου καθορίζεται από το βραδύτερο στάδιο, μπορεί να μειωθεί μόνο αν μειωθεί και η διάρκεια του βραδύτερου σταδίου. Συνεπώς διάσπαμε στην μέση το στάδιο MEM (για πάραδειγμα σε ένα στάδιο RECEIVE DATA και ένα στάδιο SEND DATA). Τα στάδια αυτά διαρκούν έκαστο $\frac{500ps}{2} = 250ps$. Συνεπώς πλέον το βραδύτερο στάδιο είναι το EX. Άρα η (ελάχιστη) διάρκεια του ρολογιού είναι $340 + 20 = 360ps$ που αντιστοιχεί σε συχνότητα ρολογίου $f = 2.777GHz$.

Όποια και αν είναι σημασιολογικά η διάσπαση του σταδίου MEM στα ίσης διάρκειας MEM1 και MEM2, θεωρούμε ότι τα δεδομένα μπορούν να προωθηθούν μόνο στο τέλος του 2ου σταδίου MEM2.

No\cc	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	IF	ID	EX	MEM1	MEM2	WB																	
2		IF	ID	ID	ID	EX	MEM1	MEM2	WB														
3			IF	IF	IF	ID	ID	ID	EX	MEM1	MEM2	WB											
4						IF	IF	IF	ID	EX	MEM1	MEM2	WB										
5									IF	ID	ID	ID	EX	MEM1	MEM2	WB							
6										IF	IF	IF	ID	EX	MEM1	MEM2	WB						
7													IF	ID	EX	MEM1	MEM2	WB					
8														IF	ID	EX	MEM1	MEM2	WB				
9															IF	ID	EX	MEM1	MEM2	WB			
next																		IF	ID	EX	MEM1	MEM2	WB

Σχήμα 3: Διάγραμμα χρονισμού για pipelined MIPS 6 σταδίων (διάσπαση MEM σε MEM1 και MEM2) με όλες τις δυνατές προωθήσεις. Με κόκκινο χρώμα σημειώνονται τα στάδια όπου γίνεται stall. (υπενθυμίζεται ότι stall συμβαίνει θέτοντας την επίτρεψη εγγραφής των ενδιάμεσων καταχωρητών και του PC σε 0, οπότε επαναλαμβάνεται ξανά το ίδιο στάδιο). Επίσης με βελάκι σημειώνονται οι προωθήσεις που συμβαίνουν.

Το διάγραμμα χρονισμού φαίνεται στην εικόνα 3. Παρατηρούμε ότι απαιτούνται 17cc για κάθε επανάληψη του βρόχου. Επιπλέον στο τέλος του βρόχου απαιτούνται 3cc ακόμα προκειμένου να ολοκληρωθεί και η τελευταία εντολή του βρόχου.

Γίνονται, όπως εξήγηθηκε πριν, $n = 0x100 = 256$ επαναλήψεις. Συνεπώς απαιτούνται συνολικά $256 \cdot 17 + 3 = 4355cc$ για την εκτέλεση του βρόχου.

Η επίδοση μετράται με τον χρόνο εκτέλεσης. Πριν την διάσπαση του MEM η εκτέλεση διαρκεί $3586cc \cdot 520ps = 1.8647\mu s$, ενώ μετά την διάσπαση διαρκεί $4355cc \cdot 360ps = 1.5678\mu s$. Πρόκειται για επιτάχυνση περίπου 16%.

5 Ερώτημα 5

Στην pipelined αρχιτεκτονική με προωθήσεις καθυστερήσεις προκύπτουν όταν στην αμέσως επόμενη εντολή από μια lw χρησιμοποιείται το αποτέλεσμα της lw (προκύπτουν και στις διακλάδωσεις, αλλά αυτό δεν βελτιώνεται απλά με αναδιατάξη). Επόμενώς στόχος είναι να όπου συμβαίνει αυτό να τοποθετήσουμε μια άσχετη εντολή από κάπου άλλου, μετά την lw, διατηρώντας όμως τις εξαρτήσεις δεδομένων.

Ο κώδικας που πρόκυπτει είναι:

```
1  loop:
2  lw $t1, 0($t2)
3  addi $t2, $t2, -4
4  lw $t3, 0($t1)
5  addi $t9, $t9, -4
6  lw $t5, 100($t3)
7  add $t4, $t3, $t3
8  add $t6, $t5, $t4
9  sw $t6, 0($t2)
10 bnez $t9, loop
```

Παρατήρουμε ότι η βελτιστοποίηση είναι βέλτιστη, δηλαδή δεν υπάρχουν καθόλου stalls.

No\cc	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	IF	ID	EX	MEM	WB											
2		IF	ID	EX	MEM	WB										
3			IF	ID	EX	MEM	WB									
4				IF	ID	EX	MEM	WB								
5					IF	ID	EX	MEM	WB							
6						IF	ID	EX	MEM	WB						
7							IF	ID	EX	MEM	WB					
8								IF	ID	EX	MEM	WB				
9									IF	ID	EX	MEM	WB			
next												IF	ID	EX	MEM	WB

Σχήμα 4: Διάγραμμα χρονισμού για pipelined MIPS 5 σταδίων με όλες τις δυνατές προωθήσεις στον αναδιατεταγμένο κώδικα. Με κόκκινο χρώμα σημειώνονται τα στάδια όπου γίνεται stall. (υπενθυμίζεται ότι stall συμβαίνει θέτοντας την επίτρεψη εγγραφής των ενδιάμεσων καταχωρητών και του PC σε 0, οπότε επαναλαμβάνεται ξανά το ίδιο στάδιο). Επίσης με βελάκι σημειώνονται οι προωθήσεις που συμβαίνουν.

Το διάγραμμα χρονισμού φαίνεται στην εικόνα 4. Παρατηρούμε ότι απαιτούνται 11cc για κάθε επανάληψη του βρόχου. Επιπλέον στο τέλος του βρόχου απαιτούνται 2cc ακόμα προκειμένου να ολοκληρωθεί και η τελευταία εντολή του βρόχου.

Γίνονται, όπως εξήγηθηκε πριν, $n = 0x100 = 256$ επαναλήψεις. Συνεπώς απαιτούνται συνολικά $256 \cdot 11 + 2 = 2818cc$ για την εκτέλεση του βρόχου. Ο χρόνος εκτέλεσης είναι $2818cc \cdot 520ps = 1.4654\mu s$

6 Ερώτημα 6

Χωρίς τις προωθήσεις απαιτούνται $5378cc \cdot 520ps = 2.797\mu s$, με τις προωθήσεις απαιτούνται $3586cc \cdot 520ps = 1.8657\mu s$, με διάσπαση του MEM σε 2 στάδια και προωθήσεις απαιτούνται $4355 \cdot 360ps = 1.5678\mu s$, ενώ τέλος με προωθήσεις και βέλτιστη αναδιάταξη των εντολών απαιτούνται $2818cc \cdot 520ps = 1.4654\mu s$. Βέλτιστη επίδοση έχουμε όταν γίνονται προωθήσεις και αναδιάταξη των εντολών.

Το συμπέρασμα που εξάγεται είναι ότι ο ελάχιστος χρόνος εκτέλεσης προκύπτει όταν έχουμε συνδυασμό hardware και software βελτιστοποιήσεων.