



2η ΑΣΚΗΣΗ ΣΤΗΝ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΠΟΛΟΓΙΣΤΩΝ

Ακ. έτος 2022-2023 5ο Εξάμηνο, Σχολή ΗΜ&ΜΥ

Ονοματεπώνυμο : Κουστένης Χρίστος

A.M : el20227

Ο κώδικας που δίνεται είναι ο ακόλουθος:

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

1)

Διάγραμμα Χρονισμού χωρίς Forwarding

	Clock Cycles																										
Instructions		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	E	M	W																					
	2		IF	ID			E	M	W																		
	3			IF			ID			E	M	W															
	4						IF			ID	E	M	W														
	5									IF	ID			E	M	W											
	6										IF			ID			E	M	W								
	7													IF			ID	E	M	W							
	8																IF	ID	E	M	W						
	9																	IF	ID			E	M	W			
1																							IF	ID	E	M	W

- Stall στους cc 4, 5 λόγω RAW dependency (μεταξύ των εντολών 1, 2) στον \$t1
- Stall στους cc 7, 8 λόγω RAW dependency (μεταξύ των εντολών 2, 3) στον \$t3
- Stall στους cc 11, 12 λόγω RAW dependency (μεταξύ των εντολών 4, 5) στον \$t5



- IV. Stall στους cc 14, 15 λόγω WAW dependency (μεταξύ των εντολών 5, 6) στον \$t6
 V. Stall στους cc 19, 20 λόγω RAW dependency (μεταξύ των εντολών 8, 9) στον \$t9
 VI. Η επόμενη lw \$t1, 0(\$t2) (εντολή 1.) θα ξεκινήσει να εκτελείται από τον cc 22 αφού η απόφαση για διακλάδωση ελήφθη στον cc 21 (στάδιο EX)

\$t9 = 0x400
 $(0x400)_{16} \rightarrow (1024)_{10}$

Ο βρόχος θα επαναληφθεί $1024/4 = 256$ φορές

Total cc = $255 * 1 + 23 = 5378$ cc

2)

Διάγραμμα Χρονισμού με Forwarding

		Clock Cycles																		
Instructions		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
	1	IF	ID	E	M	W														
	2		IF	ID		E	M	W												
	3			IF		ID		E	M	W										
	4					IF		ID	E	M	W									
	5							IF	ID		E	M	W							
	6								IF		ID	E	M	W						
	7										IF	ID	E	M	W					
	8											IF	ID	E	M	W				
	9												IF	ID	E	M	W			
	1															IF	ID	E	M	W

Στο παραπάνω διάγραμμα χρονισμού έχουμε όλες τις διαφορετικές προωθήσεις(με διαφορετικά χρώματα) που αναχαίτισαν αναμενόμενα Stalls.

Ο βρόχος θα επαναληφθεί 256 φορές όμως με τη χρήση forwarding παρατηρούμε ότι θα χρειαστούν συνολικά :

Total cc = $255 * 14 + 16 = 3586$ cc

3)

Δίνονται οι παρακάτω καθυστερήσεις κάθε σταδίου στον κύκλο σωλήνωσης :

IF : 320 ps

ID : 300 ps

EX: 340 ps



MEM : 500ps

WB : 200ps

Κάθε στάδιο θα διαρκεί όσο το πιο χρονοβόρο στάδιο του κύκλου. Το Memory Access στάδιο είναι το πιο χρονοβόρο με διάρκεια 500ps και με την προσθήκη του χρόνου που απαιτείται για την ανανέωση των καταχωρητών μεταξύ των σταδίων (20ps) έχουμε μέγιστο χρόνο υλοποίησης ενός σταδίου ίσο με 520ps. Άρα

Διάρκεια κύκλου pipeline = $520 \times 5 = 2600\text{ps}$

Total Run Program Time = $520 \times 3586 = 1864720\text{ps} = 1.864720\mu\text{s} = 0.001864720\text{ms}$

4) Μας συμφέρει να σπάσουμε το πιο χρονοβόρο στάδιο ώστε να μειωθεί ο μέγιστος χρόνος διακπεραίωσης κάθε σταδίου του κύκλου σωλήνωσης, άρα σπάμε το Memory Access σε στάδιο M1 και M2 διάρκειας 250ps το καθένα. Τώρα μέγιστης χρονικής διάρκειας στάδιο γίνεται το Execution με διάρκεια $340 + 20 = 360\text{ps}$. Επομένως, ο νέος συνολικός χρόνος ολοκλήρωσης ενός κύκλου σωλήνωσης είναι $360 \times 6 = 2160\text{ps} < 2600\text{ps}$. Άρα πετύχαμε τον σκοπό μας μειώνοντας τη συνολική διάρκεια του κύκλου. Οι κύκλοι που απαιτούνται τώρα για την εκτέλεση του κώδικα θα είναι

Total cc = $14 \times 255 + 17 = 3587\text{cc}$. Κάθε clock cycle θα διαρκεί 360ps. Οπότε

Total Run Program Time = $3587 \times 360 = 1291320\text{ps} = 1.291320\mu\text{s} = 0.001291320\text{ms}$

Διάγραμμα Χρονισμού με Forwarding μετά από σπάσιμο σταδίου

		Clock Cycles																			
Instructions		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	1	IF	ID	E	M1	M2	W														
	2		IF	ID	E	M1	M2	W													
	3			IF	ID	E	M1	M2	W												
	4				IF	ID	E	M1	M2	W											
	5					IF	ID	E	M1	M2	W										
	6						IF	ID	E	M1	M2	W									
	7							IF	ID	E	M1	M2	W								
	8								IF	ID	E	M1	M2	W							
	9									IF	ID	E	M1	M2	W						
1															IF	ID	E	M1	M2	W	

Να σημειωθεί ότι έχουμε εξετάσει το ευνοϊκό σενάριο στο οποίο οι προωθήσεις της πρώην MEM γίνονται μετά τη M1 και όχι μετά την M2.



5)

Διάγραμμα Χρονισμού με Forwarding στον αναδιαταγμένο κώδικα

		Clock Cycles															
Instructions		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	1	IF	ID	E	M	W											
	8		IF	ID	E	M	W										
	2			IF	ID	E	M	W									
	7				IF	ID	E	M	W								
	4					IF	ID	E	M	W							
	3						IF	ID	E	M	W						
	5							IF	ID	E	M	W					
	6								IF	ID	E	M	W				
	9									IF	ID	E	M	W			
	1												IF	ID	E	M	W

Ακολουθεί ο νέος αναδιαταγμένος κώδικας με αρίθμηση που ακολουθεί την αρχική για να γίνεται εύκολα η σύγκριση:

```

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

```

Total cc = 11*255+13 = 2818cc

ClockCycle time = 520ps(πριν το σπάσιμο θεώρηση)

Total Run Program Time = 1465360ps = 1.465360μs = 0.001465360ms

6) Προφανώς η περίπτωση που εξετάζεται στο πρώτο ερώτημα δηλαδή η σωλήνωση χωρίς προωθήσεις είναι η λιγότερο αποδοτική. Όπως έχει δειχθεί παραπάνω στην εκάστοτε απο τις ακόλουθες περιπτώσεις έχουμε :



$T1 = 1864720ps$ = ο χρόνος υλοποίησης του προγράμματος στην απλή περίπτωση που απλά εφαρμόζουμε προωθήσεις.

$T2 = 1291320ps$ = ο χρόνος υλοποίησης του προγράμματος στην περίπτωση που σπάμε το Memory access στάδιο σε δύο ισότιμα και στη συνέχεια χρησιμοποιούμε προωθήσεις.

$T3 = 1465360ps$ = χρόνος υλοποίησης του προγράμματος στην περίπτωση που αναδιατάσσουμε τον κώδικα για να αποφύγουμε stalls και κάνουμε χρήση σχημάτων προώθησης κατά την εκτέλεση του.

Ισχύει **$T2 < T3 < T1$** . Επομένως, η καλύτερη επίδοση επιτυγχάνεται όταν σπάμε το πιο χρονοβόρο στάδιο(Memory access) στον κύκλο σωλήνωσης σε δύο ίσα μισής καθυστέρησης και στη συνέχεια εφαρμόζουμε όλα τα δυνατά σχήματα προώθησης κατά την εκτέλεση του προγράμματος.