



**Σχολή
Ηλεκτρολόγων Μηχανικών
και
Μηχανικών Υπολογιστών**

τμήμα κ. Κοζύρη

Δεύτερη Σειρά Αρχιτεκτονικής Υπολογιστών (ακ.έτος 2018-2019)

Κόσυβας Δημήτριος
(el14828@central.ntua.gr)

ΑΜ:03114828
Εξάμηνο: 9ο

1)

Διάγραμμα Χρονισμού (χωρίς προώθηση)

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	27	28	29	30
LOOP:lw \$t1, 0(\$t0)	IF	ID	EX	M1	M2	WB																								
add \$t2,\$t1,\$t4		IF	ID	-	-	-	EX	M1	M2	WB																				
sw \$t2, 0(\$t3)			IF	-	-	-	ID	-	-	-	EX	M1	M2	WB																
add \$t3,\$t2,\$t6				-	-	-	IF	-	-	-	ID	EX	M1	W2	WB															
addi \$t0, \$t0, 4				-	-	-		-	-	-	IF	ID	EX	M1	M2	WB														
lw \$t5, 0(\$t0)				-	-	-		-	-	-		IF	ID	-	-	-	EX	M1	M2	WB										
add \$t7, \$t7, \$t2				-	-	-		-	-	-			IF	-	-	-	ID	EX	M1	M2	WB									
lw \$t6, 0(\$t5)				-	-	-		-	-	-				-	-	-	IF	ID	-	-	EX	M1	M2	WB						
add \$t4,\$t7,\$t3				-	-	-		-	-	-				-	-	-		IF	-	-	ID	EX	M1	M2	WB					
addi \$t9, \$t9, -4				-	-	-		-	-	-				-	-	-			-	-	IF	ID	EX	M1	M2	WB				
bnez \$t9, LOOP				-	-	-		-	-	-				-	-	-			-	-		IF	ID	-	-	-	EX	M1	M2	WB
lw \$t1, 0(\$t0)				-	-	-		-	-	-				-	-	-			-	-				-	-	-				IF

Εντοπισμός και αντιμετώπιση πιθανών κινδύνων που μπορεί να προκύψουν κατά την εκτέλεση :

- Stall στους κύκλους 4,5,6 : Ο καταχωρητής \$t1 για την εντολή add \$t2,\$t1,\$t4 του οποίου η ανάγνωση γίνεται στο στάδιο ID γίνεται διαθέσιμος στο τέλος του σταδίου WB από την εντολή lw \$t1, 0(\$t1) → RAW DATA HAZARD.
- Stall στους κύκλους 8,9,10 : Ο καταχωρητής \$t2 για την εντολή sw \$t2 ,0(\$t3) του οποίου η ανάγνωση γίνεται στο στάδιο ID γίνεται διαθέσιμος στο τέλος του σταδίου WB από την εντολή add \$t2,\$t1,\$t4 → RAW DATA HAZARD.
- Stall στους κύκλους 14,15,16 : Ο καταχωρητής \$t0 για την εντολή lw \$t5,0(\$t0) του οποίου η ανάγνωση γίνεται στο στάδιο ID γίνεται διαθέσιμος στο τέλος του σταδίου WB από την εντολή add \$t0,\$t0,4 → RAW DATA HAZARD
- Stall στους κύκλους 19,20 : Ο καταχωρητής \$t5 για την εντολή lw \$t6,0(\$t5) του η ανάγνωση γίνεται στο στάδιο ID γίνεται διαθέσιμος στο τέλος του σταδίου WB από την εντολή lw \$t5,0(\$t0) → RAW DATA HAZARD
- Stall στους κύκλους 24,25,26 : Ο καταχωρητής \$t9 για την εντολή bnez \$t9,LOOP του οποίου η ανάγνωση γίνεται στο στάδιο ID γίνεται διαθέσιμος στο τέλος του σταδίου WB από την εντολή addi \$t9,\$t9,-4 → RAW DATA HAZARD

Τα stalls συμβολίζονται με παύλες στο διάγραμμα .Το δεύτερο στιγμιότυπο της εντολής lw \$t1,0(\$t0) αρχίζει να εκτελείται στον κύκλο 30 καθώς η απόφαση για τη διακλάδωση ελήφθη στον κύκλο 29. Επομένως αφού η αρχική τιμή του \$t9 είναι 64 , ο βρόχος θα εκτελεστεί $64 / 4 = 16$ φορές .

Για τα πρώτες 15 επαναλήψεις απαιτούνται $15 * 29 \text{ cc} = 435 \text{ cc}$

Για την 16η επανάληψη έχουμε 30 cc..

Συνολικά απαιτούνται $(435 + 30) \text{ cc} = 465 \text{ cc}$ για την εκτέλεση του βρόχου.

2)

Διάγραμμα Χρονισμού (με προώθηση)

CC	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
LOOP:lw \$t1, 0(\$t0)	IF	ID	EX	M1	M2	WB													
add \$t2,\$t1,\$t4		IF	ID	-	-	EX	M1	M2	WB										
sw \$t2, 0(\$t3)			IF	-	-	ID	EX	M1	M2	WB									
add \$t3, \$t2, \$t6				-	-	IF	ID	EX	M1	M2	WB								
addi \$t0, \$t0, 4				-	-		IF	ID	EX	M1	M2	WB							
lw \$t5, 0(\$t0)				-	-			IF	ID	EX	M1	M2	WB						
add \$t7, \$t7, \$t2				-	-				IF	ID	EX	M1	M2	WB					
lw \$t6, 0(\$t5)				-	-					IF	ID	EX	M1	M2	WB				
add \$t4,\$t7,\$t3				-	-						IF	ID	EX	M1	M2	WB			
addi \$t9, \$t9, -4				-	-							ID	EX	M1	M2	WB			
bnez \$t9, LOOP				-	-								IF	ID	EX	M1	M2	WB	
lw \$t1, 0(\$t0)				-	-														IF

- Στους κύκλους 4,5 γίνεται stall διότι η τιμή του καταχωρητή \$t1 στην add \$t2,\$t1,\$t4 δεν μπορεί να είναι διαθέσιμη πριν το στάδιο M2.
- Στον κύκλο 12 γίνεται stall γιατί η τιμή της θέσης μνήμης 0(\$t5) που θα αποθηκευτεί στον καταχωρητή \$t6 δεν μπορεί να είναι διαθέσιμη πριν το στάδιο M2.

Τα forwards συμβολίζονται με τα βέλη στο διάγραμμα .Το δεύτερο στιγμιότυπο της εντολής lw \$t1,0(\$t1) εκτελείται στον κύκλο 19 αφού η απόφαση για τη διακλάδωση ελήφθη στον κύκλο 18 . Στα πρώτα 15 loops απαιτούνται $15 * 18 = 270 \text{ cc}$
 Στην 16η επανάληψη έχουμε 19 cc.
 Άρα συνολικά απαιτούνται $270 + 19 = 289 \text{ cc}$ για την εκτέλεση του βρόχου.

3)

a) Χρησιμοποιούμε πάλι forwarding και παράλληλα κάνουμε αναδιάταξη στον κώδικα για να πετύχουμε βέλτιστη απόδοση.

```

LOOP:  lw $t1,0($t0)
        addi $t0,$t0,4
        addi $t9,$t9,-4
        add $t2,$t1,$t4
        sw $t2,0($t3)
        add $t3,$t2,$t6
        lw $t5,0($t0)
        add $t7,$t7,$t2
        add $t4,$t7,$t3
        lw $t6,0($t5)
        bnez $t9,LOOP

```

Ο κώδικας μετά την αναδιάταξη και παρακάτω το διάγραμμα χρονισμού

CC	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
LOOP:lw \$t1,0(\$t0)	IF	ID	EX	M1	M2	WB										
addi \$t0,\$t0,4		IF	ID	EX	M1	M2	WB									
addi \$t9,\$t9,-4			IF	ID	EX	M1	M2	WB								
add \$t2,\$t1,\$t4				IF	ID	EX	M1	M2	WB							
sw \$t2,0(\$t3)					IF	ID	EX	M1	M2	WB						
add \$t3,\$t2,\$t6						IF	ID	EX	M1	M2	WB					
lw \$t5,0(\$t0)							IF	ID	EX	M1	M2	WB				
add \$t7,\$t7,\$t2								IF	ID	EX	M1	M2	WB			
add \$t4,\$t7,\$t3									IF	ID	EX	M1	M2	WB		
lw \$t6,0(\$t5)										IF	ID	EX	M1	M2	WB	
bnez \$t9,LOOP											IF	ID	EX	M1	M2	WB
lw \$t1,0(\$t0)																IF

Για τα πρώτα 15 loops απαιτούνται $15 \times 15 \text{ cc} = 225 \text{ cc}$

Για την 16η επανάληψη έχουμε 16 cc.

Οπότε χρειάζονται $225 + 16 = 241 \text{ cc}$ για την εκτέλεση του βρόχου.

b)

Στη συνέχεια γίνεται επανένωση των σταδίων M1 και M2 σε ένα και κατασκευάζουμε το διάγραμμα χρονισμού σύμφωνα με τον αρχικό κώδικα που μας είχε δοθεί (χωρίς αναδιάταξη).

CC	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
LOOP:lw \$t1, 0(\$t0)	IF	ID	EX	MEM	WB											
add \$t2,\$t1,\$t4		IF	ID	-	EX	MEM	WB									
sw \$t2, 0(\$t3)			IF	-	ID	EX	MEM	WB								
add \$t3, \$t2, \$t6					IF	ID	EX	MEM	WB							
addi \$t0, \$t0, 4						IF	ID	EX	MEM	WB						
lw \$t5, 0(\$t0)							IF	ID	EX	MEM	WB					
add \$t7, \$t7, \$t2								IF	ID	EX	MEM	WB				
lw \$t6, 0(\$t5)									IF	ID	EX	MEM	WB			
add \$t4,\$t7,\$t3										IF	ID	EX	MEM	WB		
addi \$t9, \$t9, -4											IF	ID	EX	MEM	WB	
bnez \$t9,LOOP												IF	ID	EX	MEM	WB
lw \$t1, 0(\$t0)																IF

- Stall στο κύκλο 4 γιατί η θέση μνήμης 0(\$t0) που θα αποθηκευτεί στον καταχωρητή \$t1 δεν μπορεί να είναι διαθέσιμη πριν το MEM στάδιο.

Συγκρίνοντας τα διαγράμματα χρονισμού παρατηρούμε ότι και στα δύο απαιτείται ο ίδιος χρόνος δηλαδή 241 cc .Αλλά με την αναδιάταξη του κώδικα αλλάζει η σειρά των εντολών και τελικά το πρόγραμμά μας. Συνεπώς πιστεύω ότι ο καλύτερος τρόπος με τον οποίο θα επιτύχουμε βέλτιστη επίδοση χωρίς την αλλοίωση του προγράμματος είναι με τη διαίρεση του σταδίου MEM σε 2 .