

**ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΜΠ**

“Αρχιτεκτονική Υπολογιστών”

Ορφανουδάκης Φίλιππος 03113140

1η Άσκηση

$x = 0$, επομένως πρότυπο κινητής υποδιαστολής 27 bit , με εκθέτη μήκους 6 bit και κλάσμα 20 bits.

Άρα έχουμε το εξής:

πρόσημο-s	εκθέτης-e	κλάσμα-f
1 bit	6 bit	20 bit

α) Για να έχουμε την αναπαράσταση για κάθε περίπτωση πρέπει αρχικά να έχουμε την πόλωση η οποία υπολογίζεται ως εξής $p = 2^{(e-1)} - 1 = 31 = 11111$.

Προκύπτει το εξής :

ΣΥΝΘΗΚΗ	ΤΙΜΗ
$0 < e < 63$	$(-1)^s * 2^{(e-31)} * 1,f$
$e=0, f \neq 0$	$(-1)^s * 2^{(-30)} * 0,f$
$e=0, f=0$	0


```

8.  sub $t4, $t3, $t2
9.  bne $t4, $zero, L

```

Έπειτα στον κάθετο άξονα τοποθετούμε τις εντολές και στον οριζόντιο τον αριθμό των κύκλων.

α) Χωρίς προώθηση

	1	2	3	4	5	6	7	8	9	10	11	12	13
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	-	+
7											IF	-	-
8													
9													

	14	15	16	17	18	19	20	21	22	23	24	25	26
1													
2													
3													
4													
5													
6	EX	MEM	WB										
7	ID	EX	MEM	WB									
8	IF	ID	-	+	EX	MEM	WB						

9		IF	-	-	ID	-	+	EX	MEM	WB			
2			-	-		-	-	-	IF	ID	EX	MEM	WB

Παρατηρούμε ότι εμφανίζονται καθυστερήσεις στο διάγραμμα μας . Οι λόγοι εμφάνισης καθυστέρησης είναι για την αποφυγή 3 κινδύνων :

- Κίνδυνος δομής
- Κίνδυνος δεδομένων
- Κίνδυνος ελέγχου

ΚΑΘΥΣΤΕΡΗΣΕΙΣ

- Εντολή 3: Κίνδυνος δεδομένων καθώς χρειαζόμαστε τον t0
- Εντολή 4: Κίνδυνος δομής λόγω καθυστέρησης της προηγούμενης εντολής
- Εντολή 5: Κίνδυνος δεδομένων καθώς χρειαζόμαστε τον t1
- Εντολή 6:
 - Κίνδυνος δομής λόγω καθυστέρησης της προηγούμενης εντολής
 - Κίνδυνος δεδομένων καθώς χρειαζόμαστε τον t0
- Εντολή 7: Κίνδυνος δομής λόγω καθυστέρησης της προηγούμενης εντολής
- Εντολή 8: Κίνδυνος δεδομένων καθώς χρειαζόμαστε τον t2
- Εντολή 9:
 - Κίνδυνος δομής λόγω καθυστέρησης της προηγούμενης εντολής
 - Κίνδυνος δεδομένων καθώς χρειαζόμαστε τον t2
- Εντολή 2: Κίνδυνος ελέγχου καθώς πρέπει να παρθεί η απόφαση ποια θα είναι η επόμενη εντολή

Κάθε φορά που έχει + αντί για - είναι ο δεύτερος μισός κύκλος της ID που γίνεται η ανάγνωση του καταχωρητή , αφού έχει γίνει η εγγραφή του στο πρώτο μισό κύκλο της WB.

Το loop θα ολοκληρωθεί όταν ο t4 ισούται με το μηδέν , όμως η αποπάνω εντολή λέει ότι ο $t4 = t3 - t2$, απο την 1η εντολή βλέπουμε ότι $t3 = t2 + 100$ και μέσα στη loop $t2 = t2 + 4$. Άρα $t2 + 100 - 4*i - t2 = 0 \Rightarrow i = 25$ φορές θα γίνει η loop.

αρα $1 + 25*(21-1) + 2 = 503$ κύκλοι για την εκτέλεση του κώδικα.

β) Με όλα τα σχήματα προώθησης.

	1	2	3	4	5	6	7	8	9	10	11	12	13
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
9											IF	ID	EX

	14	15	16	17	18	19	20
1							
2							
3							
4							
5							

6							
7							
8							
9	MEM	WB					
2	IF	ID	EX	MEM	WB		

Στο παραπάνω διάγραμμα θεωρούμε ότι δεν έχει εφαρμοστεί καμία τεχνική για βελτιστοποίηση του branch με είτε με συγκριτή είτε με πρόβλεψη των 2 bit.

Πάλι θα χρειαστούμε 25 επαναλήψεις της loop , άρα:
 $1+13*25+2 = 328$ κύκλους για την εκτέλεση του κώδικα.

Οι προωθήσεις φαίνονται με βέλη .

γ)

Αν στον αρχικό μου κώδικα αλλάξω την 3η με την 4η εντολή έχω το εξής αποτέλεσμα:

```

1.  addi $t3, $t2, 100
L:  2.  lw $t0, 0($t2)
3.  lw $t1, 4($t2)
4.  addi $t0, $t0, 1
5.  sub $t0, $t0, $t1
6.  sw $t0, 0($s0)
7.  addi $t2, $t2, 4
8.  sub $t4, $t3, $t2
9.  bne $t4, $zero, L

```

	1	2	3	4	5	6	7	8	9	10	11	12	13
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

	12	13	14	15	16	17	18						
2	IF	ID	EX	MEM	WB								

Οι προωθήσεις φαίνονται και με βέλη αλλά και στις πρώτες 2 περιπτώσεις που είχαμε προώθηση 2 εντολές παρακάτω δεν μπορούσε να σχεδιαστεί αλλιώς και φαίνεται με το ίδιο χρώμα δηλαδή έχω από το 2 MEM → 4 EX και 3 MEM → 5 EX.

Τώρα θα έχω $1+9*25+2 = 228$ κύκλους για την εκτέλεση του κώδικα.