Άσκηση Pipeline 1

Δεδομένα

Έχουμε ένα loop...

```
Rep:
  lw $2,100($3)
  sub $2,$2,$5
  sw $2,100($3)
  sub $3,$3,$6
  sub $1,$1,$7
  bne $1,$0, Rep
Exit:
```

Άσκηση Pipeline 1

Δεδομένα

Έχουμε ένα loop...

```
Rep:
  lw $2,100($3)
  sub $2,$2,$5
  sw $2,100($3)
  sub $3,$3,$6
  sub $1,$1,$7
  bne $1,$0, Rep
Exit:
```

και αυτή την αρχική κατάσταση στους καταχωρητές

$1: 500
$7: 5

- Δεν υπάρχει cache miss
- Cache hit σε 1cc
- branches γίνονται resolve στο MEM stage

Άσκηση Pipeline 1

Ζητούμενο Α : Για το 1ο LOOP (μέχρι και το lw του 2ου LOOP)

Να δείξετε τα **διάφορα στάδια του pipeline** (Διάγραμμα χρονισμού) που περνάνε οι εντολές.
Υποδείξτε και **εξηγείστε** τα **πιθανά hazards** που μπορούν να προκύψουν κατά την εκτέλεση, καθώς και τον τρόπο που αντιμετωπίζονται.

Κύκλος	1	2	3	4	5	6	7
Εντολή 1	IF	ID	EX	MEM	WB		
Εντολή 2		IF	ID	EX	MEM	WB	
Εντολή 3			…	…	…	…	…
…							

Άσκηση Pipeline 1

<u>Ζητούμενο Β</u> : Πόσοι κύκλοι απαιτούνται συνολικά για να ολοκληρωθεί ο βρόχος (για όλες τις επαναλήψεις του, όχι μόνο για την 1η);

```
Rep:
  lw $2,100($3)
  sub $2,$2,$5
  sw $2,100($3)
  sub $3,$3,$6
  sub $1,$1,$7
  bne $1,$0, Rep
Exit:
```

Κύκλος	1	2	3	4	5	6	7
Εντολή 1	IF	ID	EX	MEM	WB		
Εντολή 2		IF	ID	EX	MEM	WB	
Εντολή 3			…	…	…	…	…
…							

Άσκηση Pipeline 1 – Απάντηση

```
Rep:                          $1: 500
  lw $2,100($3)               $7: 5
  sub $2,$2,$5                Δεν υπάρχει cache miss
  sw $2,100($3)               cache hit σε 1cc
  sub $3,$3,$6
  sub $1,$1,$7  ⟶     $1 = 500, 495, 490, 485, ...
  bne $1,$0, Rep
Exit:
```

Ο βρόχος θα εκτελεστεί για 500 / 5 = 100 επαναλήψεις.

Το διάγραμμα χρονισμού του pipeline για τη χρονική διάρκεια που ζητείται είναι το ακόλουθο:

Κύκλος	1	2	3	4	5	6	7	8	9	10	11	12	13
`lw $2,100($3)`	IF	ID	EX	MEM	WB								
`sub $2,$2,$5`		IF	ID			EX	MEM	WB					
`sw $2,100($3)`			IF			ID			EX	MEM	WB		
`sub $3,$3,$6`						IF			ID	EX	MEM	WB	
`sub $1,$1,$7`									IF	ID	EX	MEM	WB
`bne $1,$0, Rep`										IF	ID		
`lw $2,100($3)`													

Κύκλος	14	15	16	17	18	19	20
`lw $2,100($3)`							
`sub $2,$2,$5`							
`sw $2,100($3)`							
`sub $3,$3,$6`							
`sub $1,$1,$7`							
`bne $1,$0, Rep`	EX	MEM	WB				
`lw $2,100($3)`			IF	ID	EX	MEM	WB

Ή πιο συμπιεσμένο για να χωράει σε μια σελίδα ☺

CC	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
lw	IF	ID	EX	M	WB															
sub		IF	ID			EX	M	WB												
sw			IF			ID			EX	M	WB									
sub						IF			ID	EX	M	WB								
sub									IF	ID	EX	M	WB							
bne										IF	ID			EX	M	WB				
lw																IF	ID	EX	M	WB

Άσκηση Pipeline 1 – Hazards

CC	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
lw	IF	ID	EX	M	WB															
sub		IF	ID			EX	M	WB												
sw			IF			ID			EX	M	WB									
sub						IF			ID	EX	M	WB								
sub									IF	ID	EX	M	WB							
bne										IF	ID			EX	M	WB				
lw																IF	ID	EX	M	WB

```
Rep:
   lw $2,100($3)
   sub $2,$2,$5
   sw $2,100($3)
   sub $3,$3,$6
   sub $1,$1,$7
   bne $1,$0, Rep
Exit:
```

Stalls στους κύκλους 4,5:

Ο καταχωρητής **$2** για την εντολή **sub $2,$2,$5** (η ανάγνωση του οποίου γίνεται στο στάδιο ID) γίνεται διαθέσιμος στο τέλος του κύκλου 5 (στάδιο WB) από την εντολή **lw**.

Άσκηση Pipeline 1 – Hazards

CC	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
lw	IF	ID	EX	M	WB															
sub		IF	ID			EX	M	WB												
sw			IF			ID			EX	M	WB									
sub						IF			ID	EX	M	WB								
sub									IF	ID	EX	M	WB							
bne										IF	ID			EX	M	WB				
lw																IF	ID	EX	M	WB

```
Rep:
   lw $2,100($3)
   sub $2,$2,$5
   sw $2,100($3)
   sub $3,$3,$6
   sub $1,$1,$7
   bne $1,$0, Rep
Exit:
```

Stalls στους κύκλους 7,8:

Ο $2 για την εντολή sw $2,100($3) γίνεται διαθέσιμος στο τέλος του κύκλου 8.

Άσκηση Pipeline 1 – Hazards

CC	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
lw	IF	ID	EX	M	WB															
sub		IF	ID			EX	M	WB												
sw			IF			ID			EX	M	WB									
sub						IF			ID	EX	M	WB								
sub									IF	ID	EX	M	WB							
bne										IF	ID			EX	M	WB				
lw																IF	ID	EX	M	WB

```
Rep:
   lw $2,100($3)
   sub $2,$2,$5
   sw $2,100($3)
   sub $3,$3,$6
   sub $1,$1,$7
   bne $1,$0, Rep
Exit:
```

Stalls στους κύκλους 12,13:

Ο $1 για την εντολή bne $1,$0,Rep γίνεται διαθέσιμος στο τέλος του κύκλου 13.

Άσκηση Pipeline 1 – Hazards

CC	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
lw	IF	ID	EX	M	WB															
sub		IF	ID			EX	M	WB												
sw			IF			ID			EX	M	WB									
sub						IF			ID	EX	M	WB								
sub									IF	ID	EX	M	WB							
bne										IF	ID			EX	M	WB				
lw																IF	ID	EX	M	WB

Το δεύτερο στιγμιότυπο της εντολής lw $2,100($3)
αρχίζει να εκτελείται από τον κύκλο 16,
διότι η απόφαση για την διακλάδωση ελήφθη στον κύκλο 15.

Άσκηση Pipeline 1 – Υπολογισμός Χρόνου

CC	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
lw	IF	ID	EX	M	WB															
sub		IF	ID			EX	M	WB												
sw			IF			ID			EX	M	WB									
sub						IF			ID	EX	M	WB								
sub									IF	ID	EX	M	WB							
bne										IF	ID			EX	M	WB				
lw																IF	ID	EX	M	WB

15 κύκλοι ρολογιού

Για τα πρώτα 99 loops έχουμε 99 × 15cc= 1485cc.
Για την 100η επανάληψη έχουμε 16cc.

Συνολικά απαιτούνται 1485cc + 16cc = **1501cc** για την εκτέλεση του βρόχου.

Άσκηση Pipeline 2

Για την ίδια ακολουθία εντολών, δείξτε και εξηγείστε τον χρονισμό του pipeline, θεωρώντας τώρα ότι **υπάρχει σχήμα προώθησης**. Θεωρείστε ότι οι αποφάσεις για τις διακλαδώσεις λαμβάνονται στο στάδιο MEM.

Πόσοι κύκλοι απαιτούνται συνολικά για να ολοκληρωθεί ο βρόχος;

```
Rep:
  lw $2,100($3)
  sub $2,$2,$5
  sw $2,100($3)
  sub $3,$3,$6
  sub $1,$1,$7
  bne $1,$0, Rep
Exit:
```

Άσκηση Pipeline 2 – Χωρίς/Με Προώθηση

CC	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
lw	IF	ID	EX	M	WB															
sub		IF	ID			EX	M	WB												
sw			IF			ID			EX	M	WB									
sub						IF			ID	EX	M	WB								
sub									IF	ID	EX	M	WB							
bne										IF	ID			EX	M	WB				
lw																IF	ID	EX	M	WB

CC	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
lw	IF	ID	EX	M	WB															
sub		IF	ID		EX	M	WB													
sw			IF		ID	EX	M	WB												
sub					IF	ID	EX	M	WB											
sub						IF	ID	EX	M	WB										
bne							IF	ID	EX	M	WB									
lw											IF	ID	EX	M	WB					

Άσκηση Pipeline 2 – Hazards

```
Rep:
  lw $2,100($3)
  sub $2,$2,$5
  sw $2,100($3)
  sub $3,$3,$6
  sub $1,$1,$7
  bne $1,$0,Rep
Exit:
```

Στον κύκλο 4 υπάρχει stall. **Γιατί**;

CC	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
lw	IF	ID	EX	MEM	WB										
sub		IF	ID		EX	MEM	WB								
sw			IF		ID	EX	MEM	WB							
sub					IF	ID	EX	MEM	WB						
sub						IF	ID	EX	MEM	WB					
bne							IF	ID	EX	MEM	WB				
lw											IF	ID	EX	MEM	WB

Άσκηση Pipeline 2 – Hazards

```
Rep:
  lw $2,100($3)
  sub $2,$2,$5
  sw $2,100($3)
  sub $3,$3,$6
  sub $1,$1,$7
  bne $1,$0,Rep
Exit:
```

Η τιμή της θέσης μνήμης 100($3), που θα αποθηκευτεί στον $2, δεν μπορεί να είναι διαθέσιμη πριν το στάδιο MEM.

Όταν όμως θα γίνει διαθέσιμη στο τέλος του κύκλου αυτού θα προωθηθεί στις εισόδους της ALU, ώστε να εκτελεστεί η εντολή sub $2,$2,$5 χωρίς να περιμένουμε να γραφτεί η τιμή στον $2.

Αποφεύγεται το stall που είχαμε στην προηγούμενη περίπτωση στην κύκλο 5 ☺

CC	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
lw	IF	ID	EX	MEM	WB										
sub		IF	ID		EX	MEM	WB								
sw			IF		ID	EX	MEM	WB							
sub					IF	ID	EX	MEM	WB						
sub						IF	ID	EX	MEM	WB					
bne							IF	ID	EX	MEM	WB				
lw											IF	ID	EX	MEM	WB

Το ίδιο ισχύει και για τα stalls που είχαμε στους κύκλους 7, 8

CC	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
lw	IF	ID	EX	M	WB															
sub		IF	ID			EX	M	WB												
sw			IF			ID			EX	M	WB									
sub						IF			ID	EX	M	WB								
sub									IF	ID	EX	M	WB							
bne										IF	ID			EX	M	WB				
lw																IF	ID	EX	M	WB

CC	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
lw	IF	ID	EX	M	WB															
sub		IF	ID		EX	M	WB													
sw			IF		ID	EX	M	WB												
sub					IF	ID	EX	M	WB											
sub						IF	ID	EX	M	WB										
bne							IF	ID	EX	M	WB									
lw											IF	ID	EX	M	WB					

Το ίδιο ισχύει και για τα stalls που είχαμε στους κύκλους 12, 13.

CC	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
lw	IF	ID	EX	M	WB															
sub		IF	ID			EX	M	WB												
sw			IF			ID			EX	M	WB									
sub						IF			ID	EX	M	WB								
sub									IF	ID	EX	M	WB							
bne										IF	ID			EX	M	WB				
lw																IF	ID	EX	M	WB

CC	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
lw	IF	ID	EX	M	WB															
sub		IF	ID		EX	M	WB													
sw			IF		ID	EX	M	WB												
sub					IF	ID	EX	M	WB											
sub						IF	ID	EX	M	WB										
bne							IF	ID	EX	M	WB									
lw											IF	ID	EX	M	WB					

Άσκηση Pipeline 2 – Υπολογισμός Χρόνου

CC	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
lw	IF	ID	EX	M	WB															
sub		IF	ID		EX	M	WB													
sw			IF		ID	EX	M	WB												
sub					IF	ID	EX	M	WB											
sub						IF	ID	EX	M	WB										
bne							IF	ID	EX	M	WB									
lw											IF	ID	EX	M	WB					

10 κύκλοι ρολογιού

Για τα πρώτα 99 loops έχουμε 99 × 10cc = 990cc.
Για την 100η επανάληψη έχουμε 11cc.

Συνολικά απαιτούνται 990cc + 11cc = **1001cc** για την εκτέλεση του βρόχου.

Χωρίς την προώθηση χρειαστήκαμε **1501cc.**

Άσκηση Pipeline 3

Δεδομένα

Έχουμε ένα loop…

```
LOOP: LW    $t0,  0($t3)
      ADDI  $t2,  $t0, 0
      LW    $t1,  4($t3)
      ADD   $t2,  $t2,   $t1
      SW    $t2,  0($t3)
      ADD   $t2,  $t0,   $t0
      SW    $t2,  128($t3)
      ADDI  $t3,  $t3, 8
      SUBI  $t9,  $t9, 4
      BNEZ  $t9,  LOOP
EXIT:
```

Άσκηση Pipeline 3

Δεδομένα

Έχουμε ένα loop…

```
LOOP: LW    $t0,  0($t3)
      ADDI  $t2,  $t0, 0
      LW    $t1,  4($t3)
      ADD   $t2,  $t2,   $t1
      SW    $t2,  0($t3)
      ADD   $t2,  $t0,   $t0
      SW    $t2,  128($t3)
      ADDI  $t3,  $t3, 8
      SUBI  $t9,  $t9, 4
      BNEZ  $t9,  LOOP
EXIT:
```

και αυτή την αρχική κατάσταση

$t9 = 256

- Δεν υπάρχει cache miss
- Cache hit σε 1cc
- branches γίνονται resolve στο MEM stage

Άσκηση Pipeline 3

```
LOOP: LW    $t0,  0($t3)
      ADDI  $t2,  $t0, 0
      LW    $t1,  4($t3)
      ADD   $t2,  $t2,   $t1
      SW    $t2,  0($t3)
      ADD   $t2,  $t0,   $t0
      SW    $t2,  128($t3)
      ADDI  $t3,  $t3, 8
      SUBI  $t9,  $t9, 4
      BNEZ  $t9,  LOOP
EXIT:
```

$t9 = 256

→ $t9 = 256, 252, ..., 0

Ο βρόχος θα εκτελεστεί για 256 / 4 = 64 φορές.

Άσκηση Pipeline 3

1ο Ζητούμενο : Για το 1ο LOOP (μέχρι και το lw του 2ου LOOP)

Να δείξετε τα **διάφορα στάδια του pipeline** (διάγραμμα χρονισμού) που περνάνε οι εντολές. Υποθέστε ότι η αρχιτεκτονική δε διαθέτει σχήμα προώθησης.

Κύκλος	1	2	3	4	5	6	7
Εντολή 1	IF	ID	EX	MEM	WB		
Εντολή 2		IF	ID	EX	MEM	WB	
Εντολή 3			...	...	...	...	...
...							

Άσκηση Pipeline 3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
`LW $t0, 0($t3)`	F																							
`ADDI $t2, $t0, 0`																								
`LW $t1, 4($t3)`																								
`ADD $t2, $t2, $t1`																								
`SW $t2, 0($t3)`																								
`ADD $t2, $t0, $t0`																								
`SW $t2, 128($t3)`																								
`ADDI $t3, $t3, 8`																								
`SUBI $t9, $t9, 4`																								
`BNEZ $t9, LOOP`																								
`LW $t0, 0($t3)`																								

Άσκηση Pipeline 3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
`LW $t0, 0($t3)`	F	D																						
`ADDI $t2, $t0, 0`		F																						
`LW $t1, 4($t3)`																								
`ADD $t2, $t2, $t1`																								
`SW $t2, 0($t3)`																								
`ADD $t2, $t0, $t0`																								
`SW $t2, 128($t3)`																								
`ADDI $t3, $t3, 8`																								
`SUBI $t9, $t9, 4`																								
`BNEZ $t9, LOOP`																								
`LW $t0, 0($t3)`																								

Άσκηση Pipeline 3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
`LW $t0, 0($t3)`	F	D	X																					
`ADDI $t2, $t0, 0`		F	D																					
`LW $t1, 4($t3)`			F																					
`ADD $t2, $t2, $t1`																								
`SW $t2, 0($t3)`																								
`ADD $t2, $t0, $t0`																								
`SW $t2, 128($t3)`																								
`ADDI $t3, $t3, 8`																								
`SUBI $t9, $t9, 4`																								
`BNEZ $t9, LOOP`																								
`LW $t0, 0($t3)`																								

Άσκηση Pipeline 3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
LW $t0, 0($t3)	F	D	X	M																				
ADDI $t2, $t0, 0		F	D	-																				
LW $t1, 4($t3)			F	-																				
ADD $t2, $t2, $t1																								
SW $t2, 0($t3)																								
ADD $t2, $t0, $t0																								
SW $t2, 128($t3)																								
ADDI $t3, $t3, 8																								
SUBI $t9, $t9, 4																								
BNEZ $t9, LOOP																								
LW $t0, 0($t3)																								

Άσκηση Pipeline 3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
LW $t0, 0($t3)	F	D	X	M	W																			
ADDI $t2, $t0, 0		F	D	-	-																			
LW $t1, 4($t3)			F	-	-																			
ADD $t2, $t2, $t1																								
SW $t2, 0($t3)																								
ADD $t2, $t0, $t0																								
SW $t2, 128($t3)																								
ADDI $t3, $t3, 8																								
SUBI $t9, $t9, 4																								
BNEZ $t9, LOOP																								
LW $t0, 0($t3)																								

Άσκηση Pipeline 3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
`LW $t0, 0($t3)`	F	D	X	M	W																			
`ADDI $t2, $t0, 0`		F	D	-	-	X																		
`LW $t1, 4($t3)`			F	-	-	D																		
`ADD $t2, $t2, $t1`						F																		
`SW $t2, 0($t3)`																								
`ADD $t2, $t0, $t0`																								
`SW $t2, 128($t3)`																								
`ADDI $t3, $t3, 8`																								
`SUBI $t9, $t9, 4`																								
`BNEZ $t9, LOOP`																								
`LW $t0, 0($t3)`																								

Άσκηση Pipeline 3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
`LW $t0, 0($t3)`	F	D	X	M	W																			
`ADDI $t2, $t0, 0`		F	D	-	-	X	M																	
`LW $t1, 4($t3)`			F	-	-	D	X																	
`ADD $t2, $t2, $t1`						F	D																	
`SW $t2, 0($t3)`							F																	
`ADD $t2, $t0, $t0`																								
`SW $t2, 128($t3)`																								
`ADDI $t3, $t3, 8`																								
`SUBI $t9, $t9, 4`																								
`BNEZ $t9, LOOP`																								
`LW $t0, 0($t3)`																								

Άσκηση Pipeline 3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
LW $t0, 0($t3)	F	D	X	M	W																			
ADDI $t2, $t0, 0		F	D	-	-	X	M	W																
LW $t1, 4($t3)			F	-	-	D	X	M																
ADD $t2, $t2, $t1						F	D	-																
SW $t2, 0($t3)							F	-																
ADD $t2, $t0, $t0																								
SW $t2, 128($t3)																								
ADDI $t3, $t3, 8																								
SUBI $t9, $t9, 4																								
BNEZ $t9, LOOP																								
LW $t0, 0($t3)																								

Άσκηση Pipeline 3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
LW $t0, 0($t3)	F	D	X	M	W																			
ADDI $t2, $t0, 0		F	D	-	-	X	M	W																
LW $t1, 4($t3)			F	-	-	D	X	M	W															
ADD $t2, $t2, $t1						F	D	-	-															
SW $t2, 0($t3)							F	-	-															
ADD $t2, $t0, $t0																								
SW $t2, 128($t3)																								
ADDI $t3, $t3, 8																								
SUBI $t9, $t9, 4																								
BNEZ $t9, LOOP																								
LW $t0, 0($t3)																								

Άσκηση Pipeline 3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
LW $t0, 0($t3)	F	D	X	M	W																			
ADDI $t2, $t0, 0		F	D	-	-	X	M	W																
LW $t1, 4($t3)			F	-	-	D	X	M	W															
ADD $t2, $t2, $t1						F	D	-	-	X	M	W												
SW $t2, 0($t3)							F	-	-	D	-	-	X	M	W									
ADD $t2, $t0, $t0										F	-	-	D	X	M	W								
SW $t2, 128($t3)													F	D	-	-	X	M	W					
ADDI $t3, $t3, 8														F	-	-	D	X	M	W				
SUBI $t9, $t9, 4																	F	D	X	M	W			
BNEZ $t9, LOOP																		F	D	-	-	X	M	W
LW $t0, 0($t3)																								F

Total Cycles = **63*23+24 = 1473**

Άσκηση Pipeline 3

2ο Ζητούμενο : Για το 1ο LOOP (μέχρι και το lw του 2ου LOOP)

Να δείξετε τα **διάφορα στάδια του pipeline** (διάγραμμα χρονισμού) που περνάνε οι εντολές. Υποθέστε τώρα ότι η αρχιτεκτονική **διαθέτει σχήμα προώθησης**.

Κύκλος	1	2	3	4	5	6	7
Εντολή 1	IF	ID	EX	MEM	WB		
Εντολή 2		IF	ID	EX	MEM	WB	
Εντολή 3			...	...	...	...	...
...							

Άσκηση Pipeline 3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
`LW $t0, 0($t3)`	F															
`ADDI $t2, $t0, 0`																
`LW $t1, 4($t3)`																
`ADD $t2, $t2, $t1`																
`SW $t2, 0($t3)`																
`ADD $t2, $t0, $t0`																
`SW $t2, 128($t3)`																
`ADDI $t3, $t3, 8`																
`SUBI $t9, $t9, 4`																
`BNEZ $t9, LOOP`																
`LW $t0, 0($t3)`																

Άσκηση Pipeline 3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
`LW $t0, 0($t3)`	F	D														
`ADDI $t2, $t0, 0`		F														
`LW $t1, 4($t3)`																
`ADD $t2, $t2, $t1`																
`SW $t2, 0($t3)`																
`ADD $t2, $t0, $t0`																
`SW $t2, 128($t3)`																
`ADDI $t3, $t3, 8`																
`SUBI $t9, $t9, 4`																
`BNEZ $t9, LOOP`																
`LW $t0, 0($t3)`																

Άσκηση Pipeline 3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
`LW $t0, 0($t3)`	F	D	X													
`ADDI $t2, $t0, 0`		F	D													
`LW $t1, 4($t3)`			F													
`ADD $t2, $t2, $t1`																
`SW $t2, 0($t3)`																
`ADD $t2, $t0, $t0`																
`SW $t2, 128($t3)`																
`ADDI $t3, $t3, 8`																
`SUBI $t9, $t9, 4`																
`BNEZ $t9, LOOP`																
`LW $t0, 0($t3)`																

Άσκηση Pipeline 3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
`LW $t0, 0($t3)`	F	D	X	M												
`ADDI $t2, $t0, 0`		F	D	-												
`LW $t1, 4($t3)`			F	-												
`ADD $t2, $t2, $t1`																
`SW $t2, 0($t3)`																
`ADD $t2, $t0, $t0`																
`SW $t2, 128($t3)`																
`ADDI $t3, $t3, 8`																
`SUBI $t9, $t9, 4`																
`BNEZ $t9, LOOP`																
`LW $t0, 0($t3)`																

Άσκηση Pipeline 3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
`LW $t0, 0($t3)`	F	D	X	M	W											
`ADDI $t2, $t0, 0`		F	D	-	X											
`LW $t1, 4($t3)`			F	-	D											
`ADD $t2, $t2, $t1`					F											
`SW $t2, 0($t3)`																
`ADD $t2, $t0, $t0`																
`SW $t2, 128($t3)`																
`ADDI $t3, $t3, 8`																
`SUBI $t9, $t9, 4`																
`BNEZ $t9, LOOP`																
`LW $t0, 0($t3)`																

Άσκηση Pipeline 3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
`LW $t0, 0($t3)`	F	D	X	M	W											
`ADDI $t2, $t0, 0`		F	D	-	X	M										
`LW $t1, 4($t3)`			F	-	D	X										
`ADD $t2, $t2, $t1`					F	D										
`SW $t2, 0($t3)`						F										
`ADD $t2, $t0, $t0`																
`SW $t2, 128($t3)`																
`ADDI $t3, $t3, 8`																
`SUBI $t9, $t9, 4`																
`BNEZ $t9, LOOP`																
`LW $t0, 0($t3)`																

Άσκηση Pipeline 3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
LW $t0, 0($t3)	F	D	X	M	W											
ADDI $t2, $t0, 0		F	D	-	X	M	W									
LW $t1, 4($t3)			F	-	D	X	M									
ADD $t2, $t2, $t1					F	D	-									
SW $t2, 0($t3)						F	-									
ADD $t2, $t0, $t0																
SW $t2, 128($t3)																
ADDI $t3, $t3, 8																
SUBI $t9, $t9, 4																
BNEZ $t9, LOOP																
LW $t0, 0($t3)																

Άσκηση Pipeline 3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
`LW $t0, 0($t3)`	F	D	X	M	W											
`ADDI $t2, $t0, 0`		F	D	-	X	M	W									
`LW $t1, 4($t3)`			F	-	D	X	M	W								
`ADD $t2, $t2, $t1`					F	D	-	X								
`SW $t2, 0($t3)`						F	-	D								
`ADD $t2, $t0, $t0`								F								
`SW $t2, 128($t3)`																
`ADDI $t3, $t3, 8`																
`SUBI $t9, $t9, 4`																
`BNEZ $t9, LOOP`																
`LW $t0, 0($t3)`																

Άσκηση Pipeline 3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
`LW $t0, 0($t3)`	F	D	X	M	W											
`ADDI $t2, $t0, 0`		F	D	-	X	M	W									
`LW $t1, 4($t3)`			F	-	D	X	M	W								
`ADD $t2, $t2, $t1`					F	D	-	X	M							
`SW $t2, 0($t3)`						F	-	D	X							
`ADD $t2, $t0, $t0`								F	D							
`SW $t2, 128($t3)`									F							
`ADDI $t3, $t3, 8`																
`SUBI $t9, $t9, 4`																
`BNEZ $t9, LOOP`																
`LW $t0, 0($t3)`																

Άσκηση Pipeline 3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
LW $t0, 0($t3)	F	D	X	M	W											
ADDI $t2, $t0, 0		F	D	-	X	M	W									
LW $t1, 4($t3)			F	-	D	X	M	W								
ADD $t2, $t2, $t1					F	D	-	X	M	W						
SW $t2, 0($t3)						F	-	D	X	M						
ADD $t2, $t0, $t0								F	D	X						
SW $t2, 128($t3)									F	D						
ADDI $t3, $t3, 8										F						
SUBI $t9, $t9, 4																
BNEZ $t9, LOOP																
LW $t0, 0($t3)																

Άσκηση Pipeline 3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
LW $t0, 0($t3)	F	D	X	M	W											
ADDI $t2, $t0, 0		F	D	-	X	M	W									
LW $t1, 4($t3)			F	-	D	X	M	W								
ADD $t2, $t2, $t1					F	D	-	X	M	W						
SW $t2, 0($t3)						F	-	D	X	M	W					
ADD $t2, $t0, $t0								F	D	X	M					
SW $t2, 128($t3)									F	D	X					
ADDI $t3, $t3, 8										F	D					
SUBI $t9, $t9, 4											F					
BNEZ $t9, LOOP																
LW $t0, 0($t3)																

Άσκηση Pipeline 3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
LW $t0, 0($t3)	F	D	X	M	W											
ADDI $t2, $t0, 0		F	D	-	X	M	W									
LW $t1, 4($t3)			F	-	D	X	M	W								
ADD $t2, $t2, $t1					F	D	-	X	M	W						
SW $t2, 0($t3)						F	-	D	X	M	W					
ADD $t2, $t0, $t0								F	D	X	M	W				
SW $t2, 128($t3)									F	D	X	M				
ADDI $t3, $t3, 8										F	D	X				
SUBI $t9, $t9, 4											F	D				
BNEZ $t9, LOOP												F				
LW $t0, 0($t3)																

Άσκηση Pipeline 3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
LW $t0, 0($t3)	F	D	X	M	W											
ADDI $t2, $t0, 0		F	D	-	X	M	W									
LW $t1, 4($t3)			F	-	D	X	M	W								
ADD $t2, $t2, $t1					F	D	-	X	M	W						
SW $t2, 0($t3)						F	-	D	X	M	W					
ADD $t2, $t0, $t0								F	D	X	M	W				
SW $t2, 128($t3)									F	D	X	M	W			
ADDI $t3, $t3, 8										F	D	X	M			
SUBI $t9, $t9, 4											F	D	X			
BNEZ $t9, LOOP												F	D			
LW $t0, 0($t3)																

Άσκηση Pipeline 3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
LW $t0, 0($t3)	F	D	X	M	W											
ADDI $t2, $t0, 0		F	D	-	X	M	W									
LW $t1, 4($t3)			F	-	D	X	M	W								
ADD $t2, $t2, $t1					F	D	-	X	M	W						
SW $t2, 0($t3)						F	-	D	X	M	W					
ADD $t2, $t0, $t0								F	D	X	M	W				
SW $t2, 128($t3)									F	D	X	M	W			
ADDI $t3, $t3, 8										F	D	X	M	W		
SUBI $t9, $t9, 4											F	D	X	M		
BNEZ $t9, LOOP												F	D	X		
LW $t0, 0($t3)																

Άσκηση Pipeline 3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
`LW $t0, 0($t3)`	F	D	X	M	W											
`ADDI $t2, $t0, 0`		F	D	-	X	M	W									
`LW $t1, 4($t3)`			F	-	D	X	M	W								
`ADD $t2, $t2, $t1`					F	D	-	X	M	W						
`SW $t2, 0($t3)`						F	-	D	X	M	W					
`ADD $t2, $t0, $t0`								F	D	X	M	W				
`SW $t2, 128($t3)`									F	D	X	M	W			
`ADDI $t3, $t3, 8`										F	D	X	M	W		
`SUBI $t9, $t9, 4`											F	D	X	M	W	
`BNEZ $t9, LOOP`												F	D	X	M	
`LW $t0, 0($t3)`																

Άσκηση Pipeline 3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
`LW $t0, 0($t3)`	F	D	X	M	W											
`ADDI $t2, $t0, 0`		F	D	-	X	M	W									
`LW $t1, 4($t3)`			F	-	D	X	M	W								
`ADD $t2, $t2, $t1`					F	D	-	X	M	W						
`SW $t2, 0($t3)`						F	-	D	X	M	W					
`ADD $t2, $t0, $t0`								F	D	X	M	W				
`SW $t2, 128($t3)`									F	D	X	M	W			
`ADDI $t3, $t3, 8`										F	D	X	M	W		
`SUBI $t9, $t9, 4`											F	D	X	M	W	
`BNEZ $t9, LOOP`												F	D	X	M	W
`LW $t0, 0($t3)`																F

Total Cycles = **63*15+16 = 961**

Άσκηση Pipeline 3

3ο Ζητούμενο : Για το 1ο LOOP (μέχρι και το lw του 2ου LOOP)

Προσπαθήστε να πετύχετε καλύτερη απόδοση τροποποιώντας τον κώδικα, χωρίς όμως να αλλάξετε τη σημασιολογία του προγράμματος.

Κύκλος	1	2	3	4	5	6	7
Εντολή 1	IF	ID	EX	MEM	WB		
Εντολή 2		IF	ID	EX	MEM	WB	
Εντολή 3			…	…	…	…	…
…							

Άσκηση Pipeline 3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
LW $t0, 0($t3)	F	D	X	M	W											
ADDI $t2, $t0, 0		F	D	-	X	M	W									
LW $t1, 4($t3)			F	-	D	X	M	W								
ADD $t2, $t2, $t1					F	D	-	X	M	W						
SW $t2, 0($t3)						F	-	D	X	M	W					
ADD $t2, $t0, $t0								F	D	X	M	W				
SW $t2, 128($t3)									F	D	X	M	W			
ADDI $t3, $t3, 8										F	D	X	M	W		
SUBI $t9, $t9, 4											F	D	X	M	W	
BNEZ $t9, LOOP												F	D	X	M	W
LW $t0, 0($t3)																F

κι εδώ

εδώ

Θέλουμε να τα αποφύγουμε!

Μπορούν να μετακινηθούν

Total Cycles = **63*15+16 = 961**

Άσκηση Pipeline 3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
LW $t0, 0($t3)	F	D	X	M	W											
ADDI $t3, $t3, 8		F	D	X	M	W										
ADDI $t2, $t0, 0			F	D	X	M	W									
LW $t1, **-4**($t3)				F	D	X	M	W								
SUBI $t9, $t9, 4					F	D	X	M	W							
ADD $t2, $t2, $t1						F	D	X	M	W						
SW $t2, **-8**($t3)							F	D	X	M	W					
ADD $t2, $t0, $t0								F	D	X	M	W				
SW $t2, **120**($t3)									F	D	X	M	W			
BNEZ $t9, LOOP										F	D	X	M	W		
LW $t0, 0($t3)														F	D	X

Total Cycles = **63*13+14 = 833**

Άσκηση Pipeline 4 (2010-2011)

Δεδομένα

Έχουμε ένα loop…

```
Loop: lw    $1, 0($2)
      addi  $1, $1, 1
      sw    $1, 0($2)
      addi  $2, $2, 4
      sub   $4, $3, $2
      bne   $4, $0, Loop
Exit:
```

Άσκηση Pipeline 4 (2010-2011)

Δεδομένα

Έχουμε ένα loop...

```
Loop: lw    $1, 0($2)
      addi  $1, $1, 1
      sw    $1, 0($2)
      addi  $2, $2, 4
      sub   $4, $3, $2
      bne   $4, $0, Loop
Exit:
```

και αυτή την αρχική κατάσταση στους καταχωρητές

$3 = $2 +320

- Δεν υπάρχει cache miss
- Cache hit σε 1cc
- branches γίνονται resolve στο MEM stage

Άσκηση Pipeline 4 – Απάντηση

```
Loop: lw    $1, 0($2)
      addi  $1, $1, 1
      sw    $1, 0($2)
      addi  $2, $2, 4
      sub   $4, $3, $2
      bne   $4, $0, Loop
Exit:
```

$3 = $2 + 320

→ $4 = 316, 312, 308, ..., 0

Ο βρόχος θα εκτελεστεί για 320 / 4 = 80 επαναλήψεις.

Άσκηση Pipeline 4

<u>Ζητούμενο Α</u> : Για το 1ο LOOP (μέχρι και το lw του 2ου LOOP)

Να δείξετε τα **διάφορα στάδια του pipeline** (Διάγραμμα χρονισμού) που περνάνε οι εντολές. Υποθέστε ότι η αρχιτεκτονική δε διαθέτει σχήμα προώθησης.

Κύκλος	1	2	3	4	5	6	7
Εντολή 1	IF	ID	EX	MEM	WB		
Εντολή 2		IF	ID	EX	MEM	WB	
Εντολή 3			…	…	…	…	…
…							

Άσκηση Pipeline 4 – Ζητούμενο Α

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
`lw $1,0($2)`	IF	ID	X	M	W																	
`addi $1,$1,1`		IF	ID	-	-	X	M	W														
`sw $1,0($2)`			IF	-	-	ID	-	-	X	M	W											
`addi $2,$2,4`						IF	-	-	ID	X	M	W										
`sub $4,$3,$2`									IF	ID	-	-	X	M	W							
`bne $4,$0,Loop`										IF	-	-	ID	-	-	X	M	W				
`lw $1,0($2)`																		IF	ID	X	M	W

Total Cycles = **79*17+18 = 1361**

Άσκηση Pipeline 4

<u>Ζητούμενο Β</u> : Για το 1ο LOOP (μέχρι και το lw του 2ου LOOP)

Να δείξετε τα **διάφορα στάδια του pipeline** (Διάγραμμα χρονισμού) που περνάνε οι εντολές. Υποθέστε ότι η αρχιτεκτονική διαθέτει τώρα σχήμα προώθησης.

Κύκλος	1	2	3	4	5	6	7
Εντολή 1	IF	ID	EX	MEM	WB		
Εντολή 2		IF	ID	EX	MEM	WB	
Εντολή 3			…	…	…	…	…
…							

Άσκηση Pipeline 4 – Ζητούμενο Β

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
`lw $1,0($2)`	IF	ID	X	M	W										
`addi $1,$1,1`		IF	ID	-	X	M	W								
`sw $1,0($2)`			IF	-	ID	X	M	W							
`addi $2,$2,4`					IF	ID	X	M	W						
`sub $4,$3,$2`						IF	ID	X	M	W					
`bne $4,$0,Loop`							IF	ID	X	M	W				
`lw $1,0($2)`											IF	ID	X	M	W

Total Cycles = **79*10+11 = 801**

Άσκηση Pipeline 4

<u>Ζητούμενο Γ</u> : Για το 1ο LOOP (μέχρι και το lw του 2ου LOOP)

Προσπαθήστε να πετύχετε καλύτερη απόδοση τροποποιώντας τον κώδικα, χωρίς όμως να αλλάξετε τη σημασιολογία του προγράμματος!

Κύκλος	1	2	3	4	5	6	7
Εντολή 1	IF	ID	EX	MEM	WB		
Εντολή 2		IF	ID	EX	MEM	WB	
Εντολή 3			…	…	…	…	…
…							

Άσκηση Pipeline 4 – Ζητούμενο Β

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
`lw $1,0($2)`	IF	ID	X	M	W										
`addi $1,$1,1`		IF	ID	-	X	M	W								
`sw $1,0($2)`			IF	-	ID	X	M	W							
`addi $2,$2,4`					IF	ID	X	M	W						
`sub $4,$3,$2`						IF	ID	X	M	W					
`bne $4,$0,Loop`							IF	ID	X	M	W				
`lw $1,0($2)`											IF	ID	X	M	W

Άσκηση Pipeline 4 – Ζητούμενο Γ

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
`lw $1,0($2)`														
`addi $2,$2,4`														
`addi $1,$1, 1`														
`sw $1,`**`-4($2)`**														
`sub $4,$3,$2`														
`bne $4,$0,Loop`														
`lw $1,0($2)`														

Άσκηση Pipeline 4 – Ζητούμενο Γ

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
`lw $1,0($2)`	IF	ID	X	M	W									
`addi $2,$2,4`		IF	ID	X	M	W								
`addi $1,$1, 1`			IF	ID	X	M	W							
`sw $1,-4($2)`				IF	ID	X	M	W						
`sub $4,$3,$2`					IF	ID	X	M	W					
`bne $4,$0,Loop`						IF	ID	X	M	W				
`lw $1,0($2)`										IF	ID	X	M	W

Total Cycles = **79*9+10 = 721**

Άσκηση Pipeline 5 (2011-2012)

Δεδομένα

Έχουμε ένα loop…

```
Loop: lw    $1, 0($2)
      addi  $1, $1, 1
      lw    $5, 4($2)
      add   $1, $1, $5
      sw    $1, 0($2)
      addi  $2, $2, 4
      sub   $4, $3, $2
      bne   $4, $0, Loop
Exit:
```

Άσκηση Pipeline 5 (2011-2012)

Δεδομένα

Έχουμε ένα loop...

```
Loop: lw    $1, 0($2)
      addi  $1, $1, 1
      lw    $5, 4($2)
      add   $1, $1, $5
      sw    $1, 0($2)
      addi  $2, $2, 4
      sub   $4, $3, $2
      bne   $4, $0, Loop
Exit:
```

και αυτή την αρχική κατάσταση στους καταχωρητές

$3 = $2 +220

- Δεν υπάρχει cache miss
- Cache hit σε 1cc
- branches γίνονται resolve στο MEM stage

Άσκηση Pipeline 5 – Απάντηση

```
Loop: lw    $1, 0($2)
      addi  $1, $1, 1
      lw    $5, 4($2)
      add   $1, $1, $5
      sw    $1, 0($2)
      addi  $2, $2, 4
      sub   $4, $3, $2
      bne   $4, $0, Loop
Exit:
```

$3 = $2 + 220

→ $4 = 216, 212, 208, ..., 0

Ο βρόχος θα εκτελεστεί για 220 / 4 = 55 επαναλήψεις.

Άσκηση Pipeline 5

<u>Ζητούμενο Α</u> : Για το 1ο LOOP (μέχρι και το lw του 2ου LOOP)

Να δείξετε τα **διάφορα στάδια του pipeline** (Διάγραμμα χρονισμού) που περνάνε οι εντολές. Υποθέστε ότι η αρχιτεκτονική δε διαθέτει σχήμα προώθησης.

Κύκλος	1	2	3	4	5	6	7
Εντολή 1	IF	ID	EX	MEM	WB		
Εντολή 2		IF	ID	EX	MEM	WB	
Εντολή 3			…	…	…	…	…
…							

Άσκηση Pipeline 5 – Ζητούμενο Α

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
`lw $1,0($2)`	F	D	X	M	W																	
`addi $1,$1,1`		F	D	-	-	X	M	W														
`lw $5,4($2)`			F	-	-	D	X	M	W													
`add $1,$1,$5`						F	D	-	-	X	M	W										
`sw $1,0($2)`							F	-	-	D	-	-	X	M	W							
`addi $2,$2,4`										F	-	-	D	X	M	W						
`sub $4,$3,$2`													F	D	-	-	X	M	W			
`bne $4,$0,Loop`														F	-	-	D	-	-	X	M	W
`lw $1,0($2)`																						F

Total Cycles = **54*21+22 = 1156**

Άσκηση Pipeline 5

<u>Ζητούμενο Β</u> : Για το 1ο LOOP (μέχρι και το lw του 2ου LOOP)

Να δείξετε τα **διάφορα στάδια του pipeline** (Διάγραμμα χρονισμού) που περνάνε οι εντολές. Υποθέστε ότι η αρχιτεκτονική διαθέτει τώρα σχήμα προώθησης.

Κύκλος	1	2	3	4	5	6	7
Εντολή 1	IF	ID	EX	MEM	WB		
Εντολή 2		IF	ID	EX	MEM	WB	
Εντολή 3			…	…	…	…	…
…							

Άσκηση Pipeline 5 – Ζητούμενο Β

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
`lw $1,0($2)`	F	D	X	M	W													
`addi $1,$1,1`		F	D	-	X	M	W											
`lw $5,4($2)`			F	-	D	X	M	W										
`add $1,$1,$5`					F	D	-	X	M	W								
`sw $1,0($2)`						F	-	D	X	M	W							
`addi $2,$2,4`								F	D	X	M	W						
`sub $4,$3,$2`									F	D	X	M	W					
`bne $4,$0,Loop`										F	D	X	M	W				
`lw $1,0($2)`														F	D	X	M	W

Total Cycles = **54*13+14 = 716**

Άσκηση Pipeline 5

<u>Ζητούμενο Γ</u> : Για το 1ο LOOP (μέχρι και το lw του 2ου LOOP)

Προσπαθήστε να πετύχετε καλύτερη απόδοση τροποποιώντας τον κώδικα, χωρίς όμως να αλλάξετε τη σημασιολογία του προγράμματος!

Κύκλος	1	2	3	4	5	6	7
Εντολή 1	IF	ID	EX	MEM	WB		
Εντολή 2		IF	ID	EX	MEM	WB	
Εντολή 3			…	…	…	…	…
…							

Άσκηση Pipeline 5 – Ζητούμενο Β

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
`lw $1,0($2)`	F	D	X	M	W													
`addi $1,$1,1`		F	D	-	X	M	W											
`lw $5,4($2)`			F	-	D	X	M	W										
`add $1,$1,$5`					F	D	-	X	M	W								
`sw $1,0($2)`						F	-	D	X	M	W							
`addi $2,$2,4`								F	D	X	M	W						
`sub $4,$3,$2`									F	D	X	M	W					
`bne $4,$0,Loop`										F	D	X	M	W				
`lw $1,0($2)`														F	D	X	M	W

Άσκηση Pipeline 5 – Ζητούμενο Γ

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
`lw $1,0($2)`	F	D	X	M	W													
`lw $5,4($2)`		F	D	X	M	W												
`addi $1,$1,1`			F	D	X	M	W											
`add $1,$1,$5`				F	D	X	M	W										
`sw $1,0($2)`					F	D	X	M	W									
`addi $2,$2,4`						F	D	X	M	W								
`sub $4,$3,$2`							F	D	X	M	W							
`bne $4,$0,Loop`								F	D	X	M	W						
`lw $1,0($2)`												F	D	X	M	W		

Total Cycles = **54*11+12 = 606**

Απόδοση – Άσκηση 6

Καλείστε να συμμετέχετε στην κατασκευή ενός καινούριου mp3 player και αναλαμβάνετε τη σχεδίαση του επεξεργαστή που χρησιμοποιεί το datapath του MIPS που είδαμε στο μάθημα. Μετά από μετρήσεις, ανακαλύπτετε τα παρακάτω για τα διαφορετικά κομμάτια του datapath:

- Οι αναγνώσεις/εγγραφές στο Register File απαιτούν 2ns.
- Οι λειτουργίες της ALU απαιτούν 3ns.
- Οι προσβάσεις στη μνήμη (αναγνώσεις/εγγραφές) απαιτούν 12ns.
- Οι εντολές άλματος υπό συνθήκη επιλύονται (και θέτουν τη σωστή τιμή στο PC) στο στάδιο EX.

Οι προγραμματιστές σας ενημερώνουν ότι το λογισμικό του συστήματος έχει τα εξής χαρακτηριστικά:

- 20% των εντολών είναι εντολές ADD
- 15% των εντολών είναι εντολές NAND
- 35% των εντολών είναι εντολές BEQ
- 15% των εντολών είναι εντολές LW
- 15% των εντολών είναι εντολές SW
- Το 30% των εντολών LW ακολουθείται από μια εξαρτώμενη εντολή (RAW hazard)

Αναλύστε τις τρεις παρακάτω πιθανές διαφορετικές υλοποιήσεις του επεξεργαστή:

- Single-cycle processor
- Multi-cycle processor
- Pipelined processor, όπου η προσθήκη των pipeline registers (IF/ID, ID/EX κτλ) έχει σαν αποτέλεσμα να απαιτούνται 2ns για την ανάγνωση των δεδομένων τους και 1 ns για την εγγραφή τους. Ταυτόχρονα, η εκτέλεση μιας εντολής άλματος υπό συνθήκη έχει ως αποτέλεσμα την εισαγωγή 2.65 stalls κατά μέσο όρο στο pipeline.

Για κάθε μία από τις παραπάνω υλοποιήσεις υπολογίστε τα εξής:

(i) CPI
(ii) Τη μέγιστη συχνότητα του ρολογιού του επεξεργαστή (σε MHz)
(iii) Το χρόνο εκτέλεσης (σε ms) ενός προγράμματος με 100000 εντολές.

Απόδοση – Άσκηση 6

	Single-cycle	Multi-cycle	Pipelined
CPI	1	Οι ADD, NAND, SW χρειάζονται 4 κύκλους. Οι BEQ 3 και η LW 5. Έτσι για το CPI έχουμε: $(0.2 + 0.15 + 0.15) * 4 + 0.35 * 3 + 0.15 * 5 = 3{,}8$	Το CPI ενός pipelined επεξεργαστή χωρίς εξαρτήσεις είναι 1. Εδώ stalls εισάγουν τα branches καθώς και το ποσοστό των εντολών LW που ακολουθούνται από εξαρτώμενη εντολή και άρα εμφανίζουν RAW hazard. Καθώς υπάρχουν σχήματα προώθησης τα RAW hazards αυτά εισάγουν 1 stall κάθε φορά. Έτσι θα έχουμε για το CPI: $1 + 0.35 * 2.65 + 0.15 * 0.3 * 1 = 1.9725$

Απόδοση – Άσκηση 6

	Single-cycle	Multi-cycle	Pipelined
f_{MAX}	Ο κύκλος θα είναι ίσος με αυτόν της πιο αργής εντολής, δηλαδή 12+2+3+12+2 = 31ns. Άρα η μέγιστη συχνότητα θα είναι: $\frac{1}{31 * 10^{-9}} = \frac{1000}{31} MHz$ $\approx 32.26 MHz$	Ο κύκλος θα είναι ίσος με το χρόνο που απαιτεί το πιο αργό στάδιο, δηλαδή 12ns. Άρα η μέγιστη συχνότητα θα είναι: $\frac{1}{12 * 10^{-9}} = \frac{1000}{12} MHz$ $\approx 83.33 MHz$	Ο κύκλος θα είναι ίσος με το χρόνο που απαιτεί το πιο αργό στάδιο, δηλαδή 2+12+1 = 15ns. Άρα η μέγιστη συχνότητα θα είναι: $\frac{1}{15 * 10^{-9}} = \frac{1000}{15} MHz$ $\approx 66.67 MHz$

Απόδοση – Άσκηση 6

	Single-cycle	Multi-cycle	Pipelined
t_{exec}	$t_{exec} = CPI * instr * t_{cycle}$ $= 1 * 10^5 * 31 * 10^{-9}$ $= 3.1ms$	$t_{exec} = CPI * instr * t_{cycle}$ $= 3.8 * 10^5 * 12 * 10^{-9}$ $= 4.56ms$	$t_{exec} = CPI * instr * t_{cycle}$ $= 1.9725 * 10^5 * 15 * 10^{-9}$ $= 2.958ms$