

Άσκηση 1^η – Μέρος Α

1^ο Ζητούμενο: Δίνεται το παρακάτω πρόγραμμα γραμμένο σε C, καθώς και η αντίστοιχη μετάφραση του σε assembly MIPS. Συμπληρώστε τα κενά. Σας υπενθυμίζουμε ότι ο καταχωρητής \$zero είναι πάντα μηδέν.

```
int fib (int n) {
    if (n < 2)
        return 1;
    else
        return fib(n-1) + fib(n-2);
}
```

```
fib:   addi   $sp, $sp, -12
       sw    $ra, 8($sp)
       sw    $s0, 4($sp)
       slti  $t0, $a0, 2
       beq  $t0, $zero, else
       addi  $v0, &zero, 1
       addi  $sp, $sp, 12
       jr   $ra
else:  addi  $a0, $a0, -1
       sw    $a0, 0($sp)
       jal  fib
       add  $s0, $v0, $zero
       lw   $a0, 0($sp)
       addi $a0, $a0, -1
       jal  fib
       add  $v0, $v0, $s0
       lw   $s0, 4($sp)
       lw   $ra, 8($sp)
       addi $sp, $sp, 12
       jr   $ra
```

2^η Άσκηση

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:
```

2^η Άσκηση

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

2^η Άσκηση

```
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
```

\$t9 = 256

EXIT:

→ \$t9 = 256, 252, ..., 0

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

2^η Άσκηση – 1^ο ζητούμενο

1^ο Ζητούμενο : Για το 1^ο LOOP (μέχρι και το lw του 2^{ου} 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		
...							

2^η Άσκηση – 1^ο ζητούμενο

	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)																								

2^η Άσκηση – 1^ο ζητούμενο

	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)																								

2^η Άσκηση – 1^ο ζητούμενο

	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)																								

2^η Άσκηση – 1^ο ζητούμενο

	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)																								

2^η Άσκηση – 1^ο ζητούμενο

	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)																								

2^η Άσκηση – 1^ο ζητούμενο

	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)																								

2^η Άσκηση – 1^ο ζητούμενο

	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)																								

2^η Άσκηση – 1^ο ζητούμενο

	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)																								

2^η Άσκηση – 1^ο ζητούμενο

	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)																								

2^η Άσκηση – 1^ο Ζητούμενο

	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				
SUBT \$t9, \$t9, 4																	F	D	X	M	W			
RNEZ \$t9, LOOP																	F	D	-	-	X	M	W	
LW \$t0, 0(\$t3)																								F

Total Cycles = $63 \cdot 23 + 24 = 1473$

2^η Άσκηση – 2^ο ζητούμενο

2^ο Ζητούμενο : Για το 1^ο LOOP (μέχρι και το lw του 2^{ου} 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		
...							

2^η Άσκηση – 2^ο ζητούμενο

	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)																

2^η Άσκηση – 2^ο ζητούμενο

	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)																

2^η Άσκηση – 2^ο ζητούμενο

	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)																

2^η Άσκηση – 2^ο ζητούμενο

	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)																

2^η Άσκηση – 2^ο ζητούμενο

	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)																

2^η Άσκηση – 2^ο ζητούμενο

	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)																

2^η Άσκηση – 2^ο ζητούμενο

	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)																

2^η Άσκηση – 2^ο ζητούμενο

	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)																

2^η Άσκηση – 2^ο ζητούμενο

	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)																

2^η Άσκηση – 2^ο ζητούμενο

	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)																

2^η Άσκηση – 2^ο ζητούμενο

	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)																

2^η Άσκηση – 2^ο ζητούμενο

	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)																

2^η Άσκηση – 2^ο ζητούμενο

	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)																

2^η Άσκηση – 2^ο ζητούμενο

	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)																

2^η Άσκηση – 2^ο ζητούμενο

	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)																

2^η Άσκηση – 2^ο ζητούμενο

	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 = $63 \cdot 15 + 16 = 961$

2^η Άσκηση – 3^ο ζητούμενο

3^ο Ζητούμενο : Για το 1^ο LOOP (μέχρι και το lw του 2^{ου} 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		
...							

2^η Άσκηση – 2^ο ζητούμενο (επανάληψη)

	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 = $63 \cdot 15 + 16 = 961$



2^η Άσκηση – 3^ο ζητούμενο

	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 = $63 \cdot 13 + 14 = 833$

Άσκηση 3^η – Μέρος Α

Δίνεται η παρακάτω ακολουθία προσπελάσεων

Διεύθυνση (hex)	Αποτέλεσμα
0x0DB	Miss
0x0C8	Miss
0x0DC	Hit

- Μήκος διεύθυνσης 9 bits
- Συνολικό μέγεθος tag array 48 bits
- Ελάχιστη μονάδα δεδομένων που μπορεί να διευθυνσιοδοτηθεί το 1 byte
- Πολιτική αντικατάστασης LRU
- Αρχικά η cache είναι άδεια

Άσκηση 3^η – Μέρος Α (i)

(i) Βρείτε το μέγεθος της cache

- Tag array 48 bits
- | | |
|--------------------|---|
| 48 bits x 1 block | X |
| 24 bits x 2 blocks | X |
| 16 bits x 3 blocks | X |
| 12 bits x 4 blocks | X |
| 8 bits x 6 blocks | X |
| 6 bits x 8 blocks | |
| 4 bits x 12 blocks | X |
| 3 bits x 16 blocks | |
| 2 bits x 24 blocks | X |
| 1 bit x 48 block | X |

Άσκηση 3^η – Μέρος Α (i)

- Για tag μήκους 6 bits και 8 blocks (index = 2)

	TAG	I
0x0D8 : miss →	0 1101	1000
0x0C8 : miss →	0 1100	1000
0x0DC : hit →	0 1101	1100

} ≠

- Για tag μήκους 3 bits και 16 blocks (index = 3)

	TAG	IND
0x0D8 : miss →	0 1101	1000
0x0C8 : miss →	0 1100	1000
0x0DC : hit →	0 1101	1100

} ✓

- Συνολικά 16 blocks μεγέθους 8 bytes (?) = 128 bytes cache

Άσκηση 3^η – Μέρος Α (ii)

- 0x151 → 1 0101 0001 → miss
- 0x155 → 1 0101 0101
- 0x020 → 0 0010 0000
- 0x191 → 1 0001 0001
- 0x1d4 → 1 1101 0100
- 0x153 → 1 0101 0011
- 0x123 → 1 0010 0011
- 0x021 → 1 0010 0001

set	tag
000	
001	
010	101
011	
100	
101	
110	
111	

Άσκηση 3^η – Μέρος Α (ii)

- 0x151 → 1 0101 0001 → miss
- 0x155 → 1 0101 0101 → hit
- 0x020 → 0 0010 0000
- 0x191 → 1 0001 0001
- 0x1d4 → 1 1101 0100
- 0x153 → 1 0101 0011
- 0x123 → 1 0010 0011
- 0x021 → 1 0010 0001

set	tag
000	
001	
010	101
011	
100	
101	
110	
111	

Άσκηση 3^η – Μέρος Α (ii)

- 0x151 → 1 0101 0001 → miss
- 0x155 → 1 0101 0101 → hit
- 0x020 → 0 0010 0000 → miss
- 0x191 → 1 0001 0001
- 0x1d4 → 1 1101 0100
- 0x153 → 1 0101 0011
- 0x123 → 1 0010 0011
- 0x021 → 1 0010 0001

set	tag
000	
001	
010	101
011	
100	000
101	
110	
111	

Άσκηση 3^η – Μέρος Α (ii)

- 0x151 → 1 0101 0001 → miss
- 0x155 → 1 0101 0101 → hit
- 0x020 → 0 0010 0000 → miss
- 0x191 → 1 0001 0001 → miss
- 0x1d4 → 1 1101 0100
- 0x153 → 1 0101 0011
- 0x123 → 1 0010 0011
- 0x021 → 1 0010 0001

set	tag
000	
001	
010	101 100
011	
100	000
101	
110	
111	

Άσκηση 3^η – Μέρος Α (ii)

- 0x151 → 1 0101 0001 → miss
- 0x155 → 1 0101 0101 → hit
- 0x020 → 0 0010 0000 → miss
- 0x191 → 1 0001 0001 → miss
- 0x1d4 → 1 1101 0100 → miss
- 0x153 → 1 0101 0011
- 0x123 → 1 0010 0011
- 0x021 → 1 0010 0001

set	tag
000	
001	
010	111 100
011	
100	000
101	
110	
111	

Άσκηση 3^η – Μέρος Α (ii)

- 0x151 → 1 0101 0001 → miss
- 0x155 → 1 0101 0101 → hit
- 0x020 → 0 0010 0000 → miss
- 0x191 → 1 0001 0001 → miss
- 0x1d4 → 1 1101 0100 → miss
- 0x153 → 1 0101 0011 → miss
- 0x123 → 1 0010 0011
- 0x021 → 1 0010 0001

set	tag
000	
001	
010	111 101
011	
100	000
101	
110	
111	

Άσκηση 3^η – Μέρος Α (ii)

- 0x151 → 1 0101 0001 → miss
- 0x155 → 1 0101 0101 → hit
- 0x020 → 0 0010 0000 → miss
- 0x191 → 1 0001 0001 → miss
- 0x1d4 → 1 1101 0100 → miss
- 0x153 → 1 0101 0011 → miss
- 0x123 → 1 0010 0011 → miss
- 0x021 → 1 0010 0001

set	tag
000	
001	
010	111 101
011	
100	000 100
101	
110	
111	

Άσκηση 3^η – Μέρος Α (ii)

- 0x151 → 1 0101 0001 → miss
- 0x155 → 1 0101 0101 → hit
- 0x020 → 0 0010 0000 → miss
- 0x191 → 1 0001 0001 → miss
- 0x1d4 → 1 1101 0100 → miss
- 0x153 → 1 0101 0011 → miss
- 0x123 → 1 0010 0011 → miss
- 0x021 → 1 0010 0001 → hit

set	tag
000	
001	
010	111 101
011	
100	000 100
101	
110	
111	

Άσκηση 3^η – Μέρος Β

Δίνεται το ακόλουθο κομμάτι κώδικα:

```
int i,j;
double A[8][16], B[16][16], C[8][16];

for (i = 0; i < 8; i++)
    for (j = 0; j < 16; j++)
        A[i][j] = A[i][j] + C[i][j] * B[i+8][j%8];
```

- ✓ *Κάθε στοιχείο του πίνακα είναι 8 bytes*
- ✓ *1 επίπεδο κρυφής μνήμης, direct-mapped, write-allocate, 64 blocks μεγέθους 32 bytes*
- ✓ *Αρχικά η cache είναι άδεια*
- ✓ *Όλες οι μεταβλητές αποθηκεύονται σε καταχωρητές εκτός από τα στοιχεία των πινάκων*
- ✓ *Οι πίνακες αποθηκεύονται στη μνήμη κατά γραμμές και είναι ευθυγραμμισμένοι*
- ✓ *ο A στη θέση μνήμης 0x00000140, ο B στην 0x8000f940 και ο C στην 0x08008140*
- ✓ *Η σειρά με την οποία γίνονται οι αναφορές στην μνήμη είναι A, C, B, A*

Άσκηση 3^η – Μέρος Β (i)

1^ο Ζητούμενο: Βρείτε το συνολικό αριθμό hits και misses για όλη την εκτέλεση του παραπάνω κώδικα. Υποδείξτε τον τύπο των misses.

- 1 block = 32 bytes
- 1 στοιχείο = 8 bytes
- πίνακας αποθηκευμένος κατά γραμμές



σε 1 block της cache θα απεικονίζονται 4 διαδοχικά στοιχεία του πίνακα, π.χ. $A[i][j], A[i][j+1], A[i][j+2], A[i][j+3]$

32 bytes block size = $2^5 \rightarrow$ 5 bits offset
64 blocks \rightarrow 64 sets = $2^6 \rightarrow$ 6 bits index

όλα στο set 10

$A[0][0] \rightarrow 0x0000\ 0140 = 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0001\ 0100\ 0000$
 $B[0][0] \rightarrow 0x8000\ F940 = 1000\ 0000\ 0000\ 0000\ 1111\ 1001\ 0100\ 0000$
 $C[0][0] \rightarrow 0x0800\ 8140 = 0000\ 1000\ 0000\ 0000\ 1000\ 0001\ 0100\ 0000$

$B[8][0] \rightarrow$ 32 sets πιο κάτω από το $B[0][0]$

Άσκηση 3^η – Μέρος Β (i)

A[0][0] compulsory miss

i=0, j=0

...				
10	A[0][0]	A[0][1]	A[0][2]	A[0][3]
11				
12				
13				
...				
42				
43				
44				
45				
...				

Άσκηση 3^η – Μέρος Β (i)

$i=0, j=0$

A[0][0] compulsory miss

C[0][0] compulsory miss

...				
10	C[0][0]	C[0][1]	C[0][2]	C[0][3]
11				
12				
13				
...				
42				
43				
44				
45				
...				

Άσκηση 3^η – Μέρος Β (i)

$i=0, j=0$

A[0][0] compulsory miss

C[0][0] compulsory miss

B[8][0] compulsory miss

...				
10	C[0][0]	C[0][1]	C[0][2]	C[0][3]
11				
12				
13				
...				
42	B[8][0]	B[8][1]	B[8][2]	B[8][3]
43				
44				
45				
...				

Άσκηση 3^η – Μέρος Β (i)

$i=0, j=0$

A[0][0] compulsory miss
C[0][0] compulsory miss
B[8][0] compulsory miss
A[0][0] conflict miss

...				
10	A[0][0]	A[0][1]	A[0][2]	A[0][3]
11				
12				
13				
...				
42	B[8][0]	B[8][1]	B[8][2]	B[8][3]
43				
44				
45				
...				

Άσκηση 3^η – Μέρος Β (i)

$i=0, j=0$

A[0][0] compulsory miss
C[0][0] compulsory miss
B[8][0] compulsory miss
A[0][0] conflict miss

A[0][1] hit

...				
10	A[0][0]	A[0][1]	A[0][2]	A[0][3]
11				
12				
13				
...				
42	B[8][0]	B[8][1]	B[8][2]	B[8][3]
43				
44				
45				
...				

Άσκηση 3^η – Μέρος Β (i)

i=0, j=0
A[0][0] compulsory miss
C[0][0] compulsory miss
B[8][0] compulsory miss
A[0][0] conflict miss

i=0, j=1
A[0][1] hit
C[0][1] conflict miss

...				
10	C[0][0]	C[0][1]	C[0][2]	C[0][3]
11				
12				
13				
...				
42	B[8][0]	B[8][1]	B[8][2]	B[8][3]
43				
44				
45				
...				

Άσκηση 3^η – Μέρος Β (i)

i=0, j=0
A[0][0] compulsory miss
C[0][0] compulsory miss
B[8][0] compulsory miss
A[0][0] conflict miss

i=0, j=1
A[0][1] hit
C[0][1] conflict miss
B[8][1] hit

...				
10	C[0][0]	C[0][1]	C[0][2]	C[0][3]
11				
12				
13				
...				
42	B[8][0]	B[8][1]	B[8][2]	B[8][3]
43				
44				
45				
...				

Άσκηση 3^η – Μέρος Β (i)

i=0, j=0
 A[0][0] compulsory miss
 C[0][0] compulsory miss
 B[8][0] compulsory miss
 A[0][0] conflict miss

i=0, j=1
 A[0][1] hit
 C[0][1] conflict miss
 B[8][1] hit
 A[0][1] conflict miss

...				
10	A[0][0]	A[0][1]	A[0][2]	A[0][3]
11				
12				
13				
...				
42	B[8][0]	B[8][1]	B[8][2]	B[8][3]
43				
44				
45				
...				

Άσκηση 3^η – Μέρος Β (i)

	A[0][0]	compulsory miss
i=0, j=0	C[0][0]	compulsory miss
	B[8][0]	compulsory miss
	A[0][0]	conflict miss
	A[0][1]	hit
i=0, j=1	C[0][1]	conflict miss
	B[8][1]	hit
	A[0][1]	conflict miss
	A[0][2]	hit
i=0, j=2	C[0][2]	conflict miss
	B[8][2]	hit
	A[0][2]	conflict miss
	A[0][3]	hit
i=0, j=3	C[0][3]	conflict miss
	B[8][3]	hit
	A[0][3]	conflict miss

...				
10	A[0][0]	A[0][1]	A[0][2]	A[0][3]
11				
12				
13				
...				
42	B[8][0]	B[8][1]	B[8][2]	B[8][3]
43				
44				
45				
...				

Άσκηση 3^η – Μέρος Β (i)

	A[0][4]	compulsory miss
i=0, j=4	C[0][4]	compulsory miss
	B[8][4]	compulsory miss
	A[0][4]	conflict miss
	A[0][5]	hit
i=0, j=5	C[0][5]	conflict miss
	B[8][5]	hit
	A[0][5]	conflict miss
	A[0][6]	hit
i=0, j=6	C[0][6]	conflict miss
	B[8][6]	hit
	A[0][6]	conflict miss
	A[0][7]	hit
i=0, j=7	C[0][7]	conflict miss
	B[8][7]	hit
	A[0][7]	conflict miss

...				
10	A[0][0]	A[0][1]	A[0][2]	A[0][3]
11	A[0][4]	A[0][5]	A[0][6]	A[0][7]
12				
13				
...				
42	B[8][0]	B[8][1]	B[8][2]	B[8][3]
43	B[8][4]	B[8][5]	B[8][6]	B[8][7]
44				
45				
...				

Άσκηση 3^η – Μέρος Β (i)

i=0, j=8
 A[0][8] compulsory miss
 C[0][8] compulsory miss
 B[8][0] hit ← Ήδη στην cache
 A[0][8] conflict miss

i=0, j=9
 A[0][9] hit
 C[0][9] conflict miss
 B[8][1] hit
 A[0][9] conflict miss

i=0, j=10
 A[0][10] hit
 C[0][10] conflict miss
 B[8][2] hit
 A[0][10] conflict miss

i=0, j=11
 A[0][11] hit
 C[0][11] conflict miss
 B[8][3] hit
 A[0][11] conflict miss

...				
10	A[0][0]	A[0][1]	A[0][2]	A[0][3]
11	A[0][4]	A[0][5]	A[0][6]	A[0][7]
12	A[0][8]	A[0][9]	A[0][10]	A[0][11]
13				
...				
42	B[8][0]	B[8][1]	B[8][2]	B[8][3]
43	B[8][4]	B[8][5]	B[8][6]	B[8][7]
44				
45				
...				

Άσκηση 3^η – Μέρος Β (i)

	A[0][12]	compulsory miss
i=0, j=12	C[0][12]	compulsory miss
	B[8][4]	hit ← Ήδη στην cache
	A[0][12]	conflict miss
	A[0][13]	hit
i=0, j=13	C[0][13]	conflict miss
	B[8][5]	hit
	A[0][13]	conflict miss
	A[0][14]	hit
i=0, j=14	C[0][14]	conflict miss
	B[8][6]	hit
	A[0][14]	conflict miss
	A[0][15]	hit
i=0, j=15	C[0][15]	conflict miss
	B[8][7]	hit
	A[0][15]	conflict miss

...				
10	A[0][0]	A[0][1]	A[0][2]	A[0][3]
11	A[0][4]	A[0][5]	A[0][6]	A[0][7]
12	A[0][8]	A[0][9]	A[0][10]	A[0][11]
13	A[0][12]	A[0][13]	A[0][14]	A[0][15]
...				
42	B[8][0]	B[8][1]	B[8][2]	B[8][3]
43	B[8][4]	B[8][5]	B[8][6]	B[8][7]
44				
45				
...				

Άσκηση 3^η – Μέρος Β (i)

αναφορές στη μνήμη

αποτέλεσμα

i = 0:

A[0][0]	C[0][0]	B[8][0]	A[0][0]	m	m	m	m
A[0][1]	C[0][1]	B[8][1]	A[0][1]	h	m	h	m
A[0][2]	C[0][2]	B[8][2]	A[0][2]	h	m	h	m
A[0][3]	C[0][3]	B[8][3]	A[0][3]	h	m	h	m
A[0][4]	C[0][4]	B[8][4]	A[0][4]	m	m	m	m
A[0][5]	C[0][5]	B[8][5]	A[0][5]	h	m	h	m
A[0][6]	C[0][6]	B[8][6]	A[0][6]	h	m	h	m
A[0][7]	C[0][7]	B[8][7]	A[0][7]	h	m	h	m
A[0][8]	C[0][8]	B[8][0]	A[0][8]	m	m	h	m
A[0][9]	C[0][9]	B[8][1]	A[0][9]	h	m	h	m
A[0][10]	C[0][10]	B[8][2]	A[0][10]	h	m	h	m
A[0][11]	C[0][11]	B[8][3]	A[0][11]	h	m	h	m
A[0][12]	C[0][12]	B[8][4]	A[0][12]	m	m	h	m
A[0][13]	C[0][13]	B[8][5]	A[0][13]	h	m	h	m
A[0][14]	C[0][14]	B[8][6]	A[0][14]	h	m	h	m
A[0][15]	C[0][15]	B[8][7]	A[0][15]	h	m	h	m

38 misses
26 hits

Άσκηση 3^η – Μέρος Β (i)

Συνολικά

Accesses : 64 * 8 = 512
Misses : 38 * 8 = 304
Hits : 26 * 8 = 208

Άσκηση 3^η – Μέρος Β (ii)

2^ο Ζητούμενο: Σας ζητούν να βελτιώσετε την απόδοση του κώδικα επιλέγοντας μια από τις παρακάτω λύσεις:

- i. Αντικατάσταση της cache με μια άλλη ίδιου μεγέθους, 2-way associative, με ίδιο μέγεθος block που να χρησιμοποιεί την LRU πολιτική αντικατάστασης.
- ii. Χρήση της τεχνικής merging arrays για τους πίνακες A και C. Ο καινούριος πίνακας θα ξεκινά από τη διεύθυνση **0x00000140** ενώ ο πίνακας B και πάλι από τη διεύθυνση **0x8000f940**.

Ποια από τις 2 επιλογές θα ακολουθούσατε; Δικαιολογήστε την απάντησή σας υπολογίζοντας το καινούριο miss rate. Για την περίπτωση των merging arrays δείξτε και τον καινούριο κώδικα.

Άσκηση 3^η – Μέρος Β (ii)

- i. Αντικατάσταση της cache με μια άλλη ίδιου μεγέθους, 2-way associative, με ίδιο μέγεθος block που να χρησιμοποιεί την LRU πολιτική αντικατάστασης.

$A[0][0]$, $B[0][0]$, $C[0][0]$ αποθηκεύονται όλα στο ίδιο block (10^0) της cache και κάθε block χωρά 4 στοιχεία του πίνακα. Το $B[8][0]$ αποθηκεύεται πλέον στο ίδιο set με το $B[0][0]$.

Άσκηση 3^η – Μέρος Β (ii)

αναφορές στη μνήμη

αποτέλεσμα

i = 0:

A[0][0]	C[0][0]	B[8][0]	A[0][0]	m	m	m	m
A[0][1]	C[0][1]	B[8][1]	A[0][1]	h	m	m	m
A[0][2]	C[0][2]	B[8][2]	A[0][2]	h	m	m	m
A[0][3]	C[0][3]	B[8][3]	A[0][3]	h	m	m	m
A[0][4]	C[0][4]	B[8][4]	A[0][4]	m	m	m	m
A[0][5]	C[0][5]	B[8][5]	A[0][5]	h	m	m	m
A[0][6]	C[0][6]	B[8][6]	A[0][6]	h	m	m	m
A[0][7]	C[0][7]	B[8][7]	A[0][7]	h	m	m	m
A[0][8]	C[0][8]	B[8][0]	A[0][8]	m	m	h	h
A[0][9]	C[0][9]	B[8][1]	A[0][9]	h	h	h	h
A[0][10]	C[0][10]	B[8][2]	A[0][10]	h	h	h	h
A[0][11]	C[0][11]	B[8][3]	A[0][11]	h	h	h	h
A[0][12]	C[0][12]	B[8][4]	A[0][12]	m	m	h	h
A[0][13]	C[0][13]	B[8][5]	A[0][13]	h	h	h	h
A[0][14]	C[0][14]	B[8][6]	A[0][14]	h	h	h	h
A[0][15]	C[0][15]	B[8][7]	A[0][15]	h	h	h	h

30 misses
34 hits

Θέμα Β – 2^ο Ζητούμενο

- i. Αντικατάσταση της cache με μια άλλη ίδιου μεγέθους, 2-way associative, με ίδιο μέγεθος block που να χρησιμοποιεί την LRU πολιτική αντικατάστασης.

Συνολικά

Accesses : $64 * 8 = 512$
Misses : $30 * 8 = 240$
Hits : $34 * 8 = 272$

Θέμα Β – 2^ο Ζητούμενο

- ii. Χρήση της τεχνικής merging arrays για τους πίνακες A και C. Ο καινούριος πίνακας θα ξεκινά από τη διεύθυνση **0x00000140** ενώ ο πίνακας B και πάλι από τη διεύθυνση **0x8000f940**.

```
struct merged {
    double a;
    double b;
} merged_AC;

int i, j;
merged_AC AC[8][16];
double B[16][16];

for (i=0; i < 8; i++)
    for (j=0; j < 16; j++)
        AC[i][j].a = AC[i][j].a + AC[i][j].c * B[i+8][j%8];
```

AC[0][0], B[0][0] αποθηκεύονται όλα στο ίδιο block (10^ο) της cache και κάθε block χωρά 2 και 4 στοιχεία του κάθε πίνακα αντίστοιχα. Το B[8][0] αποθηκεύεται 32 sets πιο κάτω από το B[0][0].

Άσκηση 3^η – Μέρος Β (ii)

αναφορές στη μνήμη

αποτέλεσμα

i = 0:

AC[0][0].a	AC[0][0].c	B[8][0]	AC[0][0].a	m	h	m	h
AC[0][1].a	AC[0][1].c	B[8][1]	AC[0][1].a	h	h	h	h
AC[0][2].a	AC[0][2].c	B[8][2]	AC[0][2].a	m	h	h	h
AC[0][3].a	AC[0][3].c	B[8][3]	AC[0][3].a	h	h	h	h
AC[0][4].a	AC[0][4].c	B[8][4]	AC[0][4].a	m	h	m	h
AC[0][5].a	AC[0][5].c	B[8][5]	AC[0][5].a	h	h	h	h
AC[0][6].a	AC[0][6].c	B[8][6]	AC[0][6].a	h	m	h	h
AC[0][7].a	AC[0][7].c	B[8][7]	AC[0][7].a	h	h	h	h
AC[0][8].a	AC[0][8].c	B[8][0]	AC[0][8].a	m	h	h	h
AC[0][9].a	AC[0][9].c	B[8][1]	AC[0][9].a	h	h	h	h
AC[0][10].a	AC[0][10].c	B[8][2]	AC[0][10].a	m	h	h	h
AC[0][11].a	AC[0][11].c	B[8][3]	AC[0][11].a	h	h	h	h
AC[0][12].a	AC[0][12].c	B[8][4]	AC[0][12].a	m	h	h	h
AC[0][13].a	AC[0][13].c	B[8][5]	AC[0][13].a	h	h	h	h
AC[0][14].a	AC[0][14].c	B[8][6]	AC[0][14].a	h	m	h	h
AC[0][15].a	AC[0][15].c	B[8][7]	AC[0][15].a	h	h	h	h

10 misses
54 hits

Άσκηση 3^η – Μέρος Β (ii)

αναφορές στη μνήμη

αποτέλεσμα

i = 7:

AC[7][0].a	AC[7][0].c	B[15][0]	AC[7][0].a	m	h	m	h
AC[7][1].a	AC[7][1].c	B[15][1]	AC[7][1].a	h	h	h	h
AC[7][2].a	AC[7][2].c	B[15][2]	AC[7][2].a	m	h	h	h
AC[7][3].a	AC[7][3].c	B[15][3]	AC[7][3].a	h	h	h	h
AC[7][4].a	AC[7][4].c	B[15][4]	AC[7][4].a	m	h	m	h
AC[7][5].a	AC[7][5].c	B[15][5]	AC[7][5].a	h	h	h	h
AC[7][6].a	AC[7][6].c	B[15][6]	AC[7][6].a	h	h	h	h
AC[7][7].a	AC[7][7].c	B[15][7]	AC[7][7].a	h	h	h	h
AC[7][8].a	AC[7][8].c	B[15][0]	AC[7][8].a	m	h	m	m
AC[7][9].a	AC[7][9].c	B[15][1]	AC[7][9].a	h	h	m	m
AC[7][10].a	AC[7][10].c	B[15][2]	AC[7][10].a	m	h	m	h
AC[7][11].a	AC[7][11].c	B[15][3]	AC[7][11].a	h	h	h	h
AC[7][12].a	AC[7][12].c	B[15][4]	AC[7][12].a	m	h	h	h
AC[7][13].a	AC[7][13].c	B[15][5]	AC[7][13].a	h	h	h	h
AC[7][14].a	AC[7][14].c	B[15][6]	AC[7][14].a	h	m	h	h
AC[7][15].a	AC[7][15].c	B[15][7]	AC[7][15].a	h	h	h	h

Νέα misses εδώ λόγω conflict!

15 misses
49 hits

...			
2	AC[7][0] B[14][0]		
...			
6	AC[7][8] B[15][0]	AC[7][8] B[15][1]	AC[7][9] B[15][2] AC[7][9] B[15][3]
7	AC[7][10] B[15][4]	AC[7][10] B[15][5]	AC[7][11] B[15][6] AC[7][11] B[15][7]
8	AC[7][12] B[15][8]	AC[7][12] B[15][9]	AC[7][13] B[15][10] AC[7][13] B[15][11]
9	AC[7][14] B[15][12]	AC[7][14] B[15][13]	AC[7][15] B[15][14] AC[7][15] B[15][15]
10	AC[0][0]		
...			
18	AC[1][0]		
...			
26	AC[2][0]		
...			
34	AC[3][0]		
...			
42	AC[4][0] B[8][0]		
...			
54	B[11][0]		
...			
58	AC[6][0] B[12][0]		
...			
62	B[13][0]		
63			

Θέμα Β – 2^ο Ζητούμενο

- ii. Χρήση της τεχνικής merging arrays για τους πίνακες A και C. Ο καινούριος πίνακας θα ξεκινά από τη διεύθυνση **0x00000140** ενώ ο πίνακας B και πάλι από τη διεύθυνση **0x8000f940**.

Συνολικά

Accesses : $64 * 8 = 512$
Misses : $10 * 7 + 15 = 85$
Hits : $54 * 7 + 49 = 427$