



ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΥΠΟΛΟΓΙΣΤΩΝ
ΠΜΣ: «Επιστήμη και Τεχνολογία της Πληροφορικής και των
Υπολογιστών»

ΜΑΘΗΜΑ: «Προχωρημένα Θέματα Αρχιτεκτονικής
Υπολογιστών» - Χειμερινό εξάμ. 2019-20 - Δημήτρης Κεχαγιάς

ΕΡΓΑΣΙΑ 1: Μνήμη cache χρησιμοποιώντας τον προσομοιωτή PCSpim-Cache

ΜΕΡΟΣ Α: Γνωριμία με τον προσομοιωτή PCSpim-Cache

- Είναι ένας παραθυρικός προσομοιωτής για τη μελέτη του μηχανισμού της κρυφής μνήμης (μνήμη cache)
- Βασίζεται στον προσομοιωτή PCSpim, ο οποίος τρέχει προγράμματα συμβολικής γλώσσας του επεξεργαστή MIPS32
- Μπορείτε να τον κατεβάσετε από την ιστοσελίδα:
<http://www.disca.upv.es/spetit/spim.htm>

Το περιβάλλον του PCSPIM-CACHE:

- Μπάρα των μενού
- Ένα κεντρικό παράθυρο με 6 εσωτερικά υποπαράθυρα:
 1. Των καταχωρητών
 2. Της μνήμης εντολών (του κώδικα)
 3. Της μνήμης δεδομένων
 4. Της μνήμης cache εντολών
 5. Της μνήμης cache δεδομένων
 6. Των μηνυμάτων
- Status line (Γραμμή κατάστασης)

PCSPIM-CACHE – Screenshot:

The screenshot shows the PCSPIM-CACHE simulator interface. Red brackets and arrows point to various sections with labels in Greek:

- Καταχωρητές** (Registers): Points to the top section showing PC, Status, HI, LO, and General Registers (R0-R25).
- Μνήμη εντολών (κώδικας)** (Instruction Memory): Points to the instruction list starting with [0x00400000] lui \$1, 4096 [Array_A] ; 7: la \$2, Array_A.
- Μνήμη δεδομένων** (Data Memory): Points to the data section starting with [0x10000000] ... [0x10000480] 0x00000000.
- Μνήμη cache εντολών** (Instruction Cache): Points to the table with columns Set, V, Tag (h), Instructions (h), and Acc.
- Μνήμη cache δεδομένων** (Data Cache): Points to the table with columns Set, V, Tag (h), Data (h), and Acc.
- Μηνύματα** (Messages): Points to the bottom section showing memory and registers cleared and the simulator reinitialized.

Το παράθυρο του κώδικα:

```

[0x00400000] 0x3c011000 lui $1, 4096 [Array_A] ; 7: la $2, Array_A
[0x00400004] 0x34220480 ori $2, $1, 1152 [Array_A]
[0x00400008] 0x34060000 ori $6, $0, 0 ; 8: li $6, 0
[0x0040000c] 0x34040008 ori $4, $0, 8 ; 9: li $4, 8
[0x00400010] 0x8c450000 lw $5, 0($2) ; 10: lw $5, 0($2)
[0x00400014] 0x00c53020 add $6, $6, $5 ; 11: add $6, $6, $5
[0x00400018] 0x20420004 addi $2, $2, 4 ; 12: addi $2, $2, 4
[0x0040001c] 0x2084ffff addi $4, $4, -1 ; 13: addi $4, $4, -1
[0x00400020] 0x0004082a slt $1, $0, $4 ; 14: bgt $4, $0, loop
[0x00400024] 0x1420ffff bne $1, $0, -20 [loop-0x00400024]
    
```

- Διευθύνσεις
- Εντολές σε γλώσσα μηχανής
- Μεταφρασμένες εντολές assembly
- Αρχικές εντολές assembly

Το παράθυρο των καταχωρητών:

- Καταχωρητές γενικού σκοπού: **R0-R31**
- Καταχωρητές μονής και διπλής κινητής υποδιαστολής: **(FP0-FP31), (FP0,FP2..FP30)**
- **PC, EPC, Cause, BadVAddr, Status, HI, LO**

PC = 00000000 EPC = 00000000 Cause = 00000000 BadVAddr= 00000000
Status = 3000ff10 HI = 00000000 LO = 00000000

General Registers

R0 (r0) = 00000000 R8 (t0) = 00000000 R16 (s0) = 00000000 R24 (t8) = 00000000
R1 (at) = 00000000 R9 (t1) = 00000000 R17 (s1) = 00000000 R25 (t9) = 00000000
R2 (v0) = 00000000 R10 (t2) = 00000000 R18 (s2) = 00000000 R26 (k0) = 00000000
R3 (v1) = 00000000 R11 (t3) = 00000000 R19 (s3) = 00000000 R27 (k1) = 00000000
R4 (a0) = 00000000 R12 (t4) = 00000000 R20 (s4) = 00000000 R28 (gp) = 10008000
R5 (a1) = 00000000 R13 (t5) = 00000000 R21 (s5) = 00000000 R29 (sp) = 7fffeffc
R6 (a2) = 00000000 R14 (t6) = 00000000 R22 (s6) = 00000000 R30 (s8) = 00000000
R7 (a3) = 00000000 R15 (t7) = 00000000 R23 (s7) = 00000000 R31 (ra) = 00000000
FIR = 00009800 FCSR = 00000000 FCCR = 00000000 FEXR = 00000000
FENR = 00000000

Double Floating Point Registers

FP0 = 0.000000 FP8 = 0.000000 FP16 = 0.000000 FP24 = 0.000000
FP2 = 0.000000 FP10 = 0.000000 FP18 = 0.000000 FP26 = 0.000000
FP4 = 0.000000 FP12 = 0.000000 FP20 = 0.000000 FP28 = 0.000000
FP6 = 0.000000 FP14 = 0.000000 FP22 = 0.000000 FP30 = 0.000000

Single Floating Point Registers

FP0 = 0.000000 FP8 = 0.000000 FP16 = 0.000000 FP24 = 0.000000
FP1 = 0.000000 FP9 = 0.000000 FP17 = 0.000000 FP25 = 0.000000
FP2 = 0.000000 FP10 = 0.000000 FP18 = 0.000000 FP26 = 0.000000
FP3 = 0.000000 FP11 = 0.000000 FP19 = 0.000000 FP27 = 0.000000
FP4 = 0.000000 FP12 = 0.000000 FP20 = 0.000000 FP28 = 0.000000
FP5 = 0.000000 FP13 = 0.000000 FP21 = 0.000000 FP29 = 0.000000
FP6 = 0.000000 FP14 = 0.000000 FP22 = 0.000000 FP30 = 0.000000
FP7 = 0.000000 FP15 = 0.000000 FP23 = 0.000000 FP31 = 0.000000

Το παράθυρο της μνήμης δεδομένων:

DATA

[0x10000000]...[0x10000480] 0x00000000
 [0x10000480] 0x00000001 0x00000001 0x00000001 0x00000001
 [0x10000490] 0x00000002 0x00000002 0x00000002 0x00000002
 [0x100004a0]... [0x10000cc0] 0x00000000
 [0x10000cc0] 0x00000003 0x00000003 0x00000003 0x00000003
 [0x10000cd0] 0x00000004 0x00000004 0x00000004 0x00000004
 [0x10000ce0]...[0x10040000] 0x00000000

STACK

[0x7ffffc]

KERNEL DATA

[0x90000000]...[0x90010000]

- Διευθύνσεις
- Δεδομένα

Το παράθυρο της cache εντολών:

- Εντολές
- Στατιστικά

The screenshot shows the PCSim-Cache simulator interface. The top section displays system status and registers. The middle section shows the memory layout with addresses and values. The bottom section shows the instruction cache and data cache statistics.

System Status:

- PC = 00000000, EPC = 00000000, Cause = 00000000, BadVAddr = 00000000
- Status = 3000ff10, HI = 00000000, LO = 00000000

General Registers:

- R0 (r0) = 00000000, R8 (t0) = 00000000, R16 (s0) = 00000000, R24 (t8) = 00000000
- R1 (at) = 00000000, R9 (t1) = 00000000, R17 (s1) = 00000000, R25 (t9) = 00000000

Memory Layout:

Address	Value
[0x10000000]...[0x10000480]	0x00000000
[0x10000480]	0x00000001 0x00000001 0x00000001 0x00000001
[0x10000490]	0x00000002 0x00000002 0x00000002 0x00000002
[0x100004a0]...[0x10000cc0]	0x00000000
[0x10000cc0]	0x00000003 0x00000003 0x00000003 0x00000003
[0x10000cd0]	0x00000004 0x00000004 0x00000004 0x00000004
[0x10000ce0]...[0x10040000]	0x00000000

Instruction Cache:

Set	V	Tag (h)	Instructions (h)	Acc
8	1	8000	slt \$1, \$0, \$4	
9	1	8000	bne \$1, \$0, -20	
10	1	8000	NULL	miss
11	0			
12	0			
13	0			

Data Cache:

Set	V	Tag (h)	Data (h)	Acc
5	1	200009	00000002	
6	1	200009	00000002	
7	1	200009	00000002	miss
8	0			
9	0			
10	0			

Cache Statistics:

- Instruction Cache:** Accesses:53 Hits:42 Hit Rate:0.792453
- Data Cache:** Accesses:8 Hits:0 Hit Rate:0.000000 Misses: Compulsory:8 Conflict:0 Capacity:0

Footer:

C:\Documents and Settings\Dimitris Kehagias\Επιφάνεια εργασίας\Simulators\SPIM Cache\sum.s successfully loaded

For Help, press F1

PC=0x00000000 EPC=0x00000000 Cause=0x00000000

Το παράθυρο της cache δεδομένων:

- Δεδομένα
- Στατιστικά

PC = 00000000 EPC = 00000000 Cause = 00000000 BadVAddr= 00000000
Status = 3000f110 HI = 00000000 LO = 00000000
General Registers
R0 (r0) = 00000000 R8 (t0) = 00000000 R16 (s0) = 00000000 R24 (t8) = 00000000
R1 (at) = 00000000 R9 (t1) = 00000000 R17 (s1) = 00000000 R25 (t9) = 00000000

[0x00400000] 0x3c011000 lui \$1, 4096 [Array A] ; 7: la \$2, Array A
[0x00400004] 0x34220480 ori \$2, \$1, 1152 [Array_A]
[0x00400008] 0x34050000 ori \$6, \$0, 0 ; 8: li \$6, 0
[0x0040000c] 0x34040008 ori \$4, \$0, 8 ; 9: li \$4, 8
[0x00400010] 0x8c450000 lw \$5, 0(\$2) ; 10: lw \$5, 0(\$2)

DATA
[0x10000000] ... [0x10000480] 0x00000000
[0x10000480] 0x00000001 0x00000001 0x00000001 0x00000001
[0x10000490] 0x00000002 0x00000002 0x00000002 0x00000002
[0x100004a0] ... [0x10000c00] 0x00000000

Set V Tag (h) Instructions (h) Acc
8 1 8000 slt \$1, \$0, \$4
9 1 8000 bne \$1, \$0, -20
10 1 8000 NULL miss
11 0
12 0

Instruction Cache Accesses:53 Hits:42 Hit Rate:0.792453

Set V Tag (h) Data (h) Acc
5 1 200009 00000002
6 1 200009 00000002
7 1 200009 00000002 miss
8 0
9 0
10 0

Data Cache Accesses:8 Hits:0 Hit Rate:0.000000 Misses: Compulsory:8 Conflict:0 Capacity:0

C:\Documents and Settings\Dimitris Kehagias\Επιφάνεια εργασίας\Simulators\SPIM Cache\sum.s successfully loaded

For Help, press F1 PC=0x00000000 EPC=0x00000000 Cause=0x00000000

Το μενού File:

- Φόρτωση αρχείου (επιλέγουμε το αρχείο με επέκταση.s)
- Αποθηκεύουμε το αρχείο log
- Έξοδος

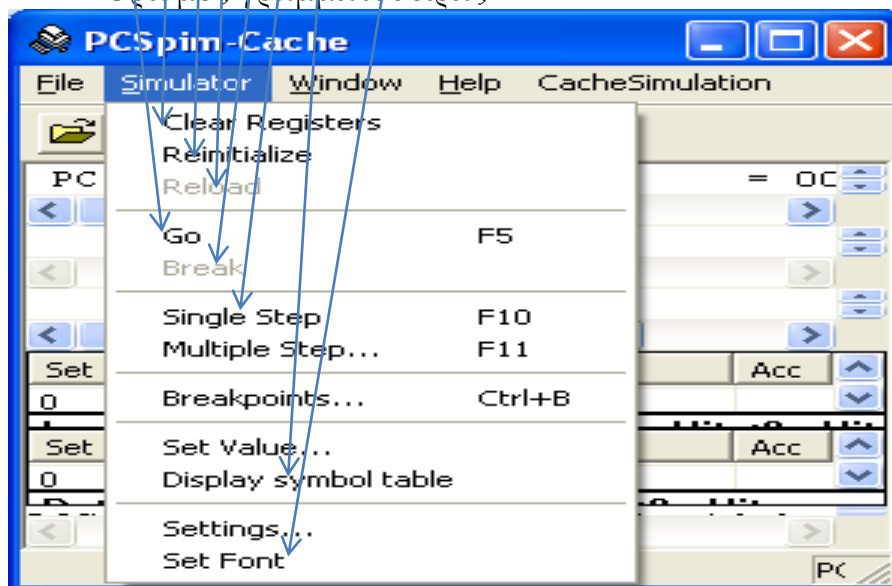
File Simulator Window Help CacheSimulation

Open Ctrl+O
Save Log File... Ctrl+S
Exit Alt+F4

PC=0x00000000 EPC=0x00000000 Cause=0x00000000

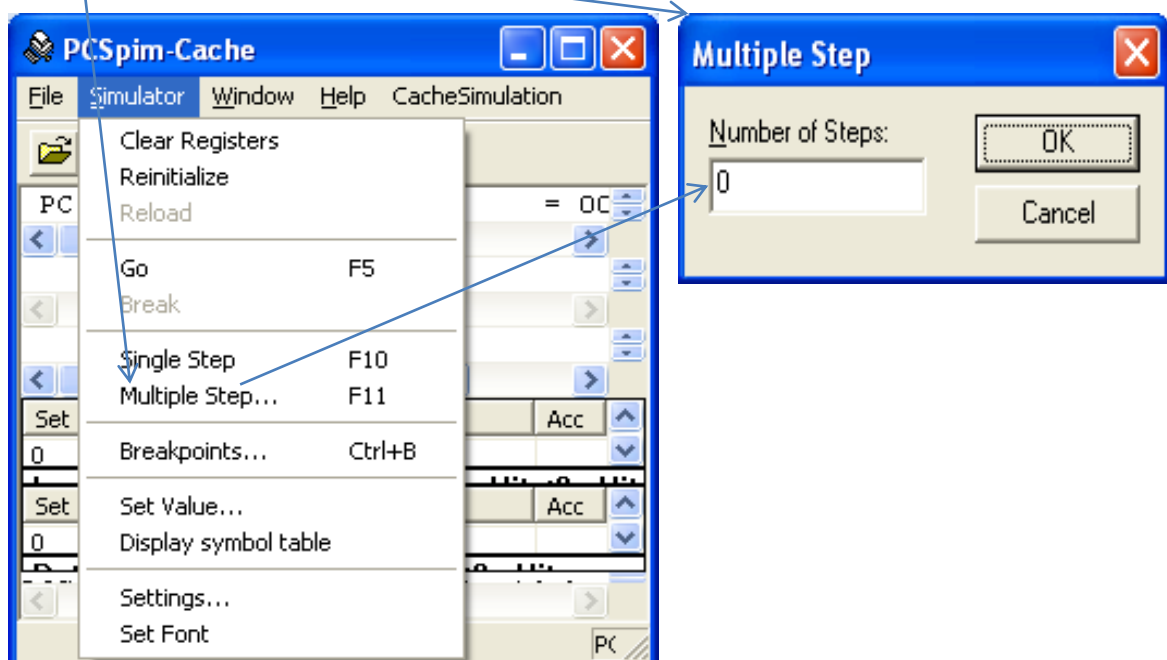
Το μενού Simulator(1/5):

- Μηδενισμός περιεχομένων καταχωρητών
- Μηδενισμός περιεχομένων καταχωρητών, μνήμης και επαναρχικοποίηση του προσομοιωτή
- Επαναφόρτωση προγράμματος
- Εκτέλεση όλου του προγράμματος
- Διακοπή εκτέλεσης
- Εκτέλεση βήμα-βήμα
- Εμφάνιση συμβολικού πίνακα
- Ορισμός γραμματοσειράς



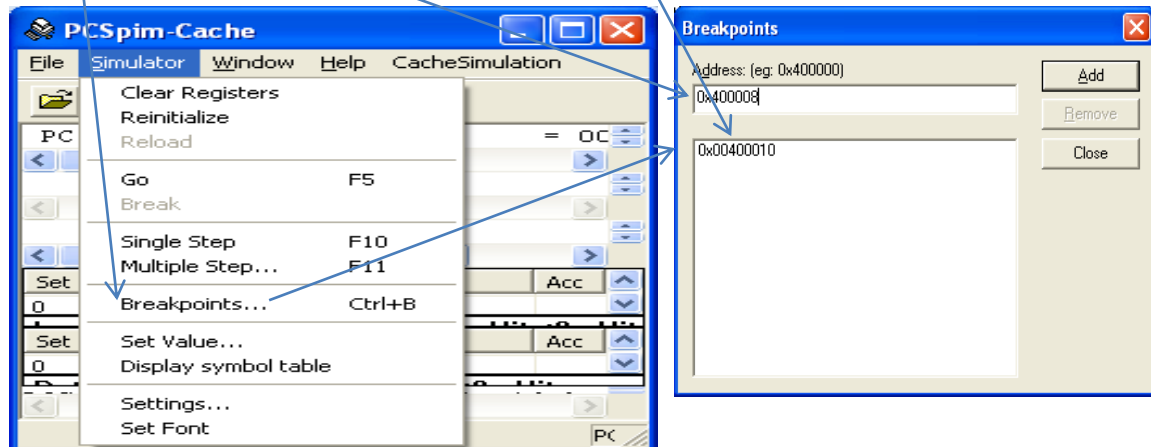
Το μενού Simulator(2/5):

Εκτέλεση προγράμματος με πολλαπλά βήματα, τα οποία ορίζουμε στο πλαίσιο που εμφανίζεται.



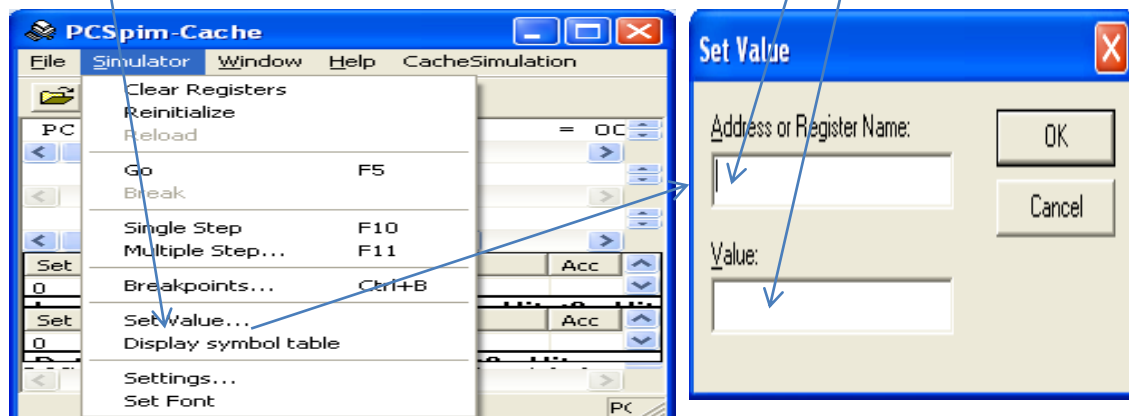
Το μενού Simulator(3/5):

Ορίζουμε σημεία διακοπής μέσω του πλαισίου που εμφανίζεται και της επιλογής add. Μπορούμε να διαγράψουμε ένα σημείο διακοπής επιλέγοντάς το από τη λίστα και την επιλογή remove



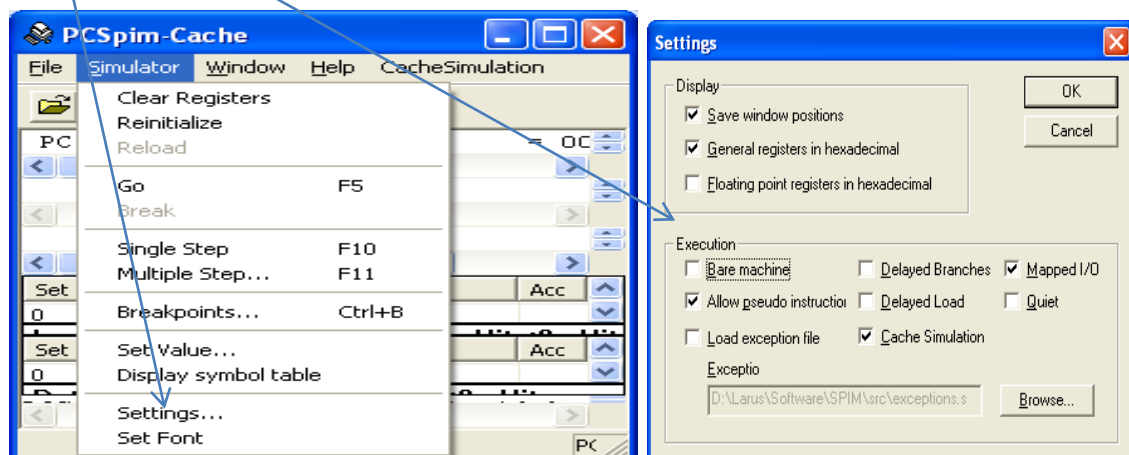
Το μενού Simulator(4/5):

Ορίζουμε τιμές σε καταχωρητές ή θέσεις μνήμης. Δίνουμε το όνομα του καταχωρητή ή της θέσης μνήμης και την αντίστοιχη τιμή.



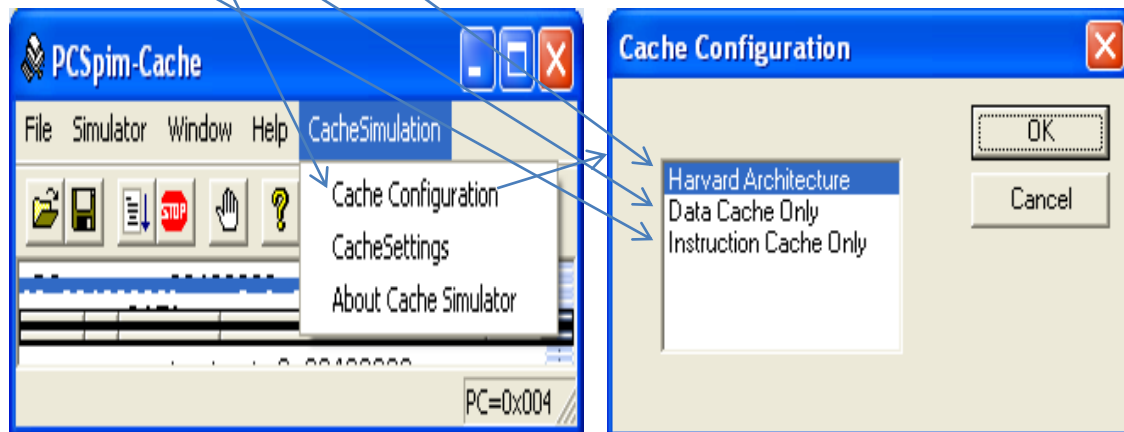
Το μενού Simulator(5/5):

Ρυθμίσεις. Οι βασικές ρυθμίσεις εκτέλεσης του προσομοιωτή.



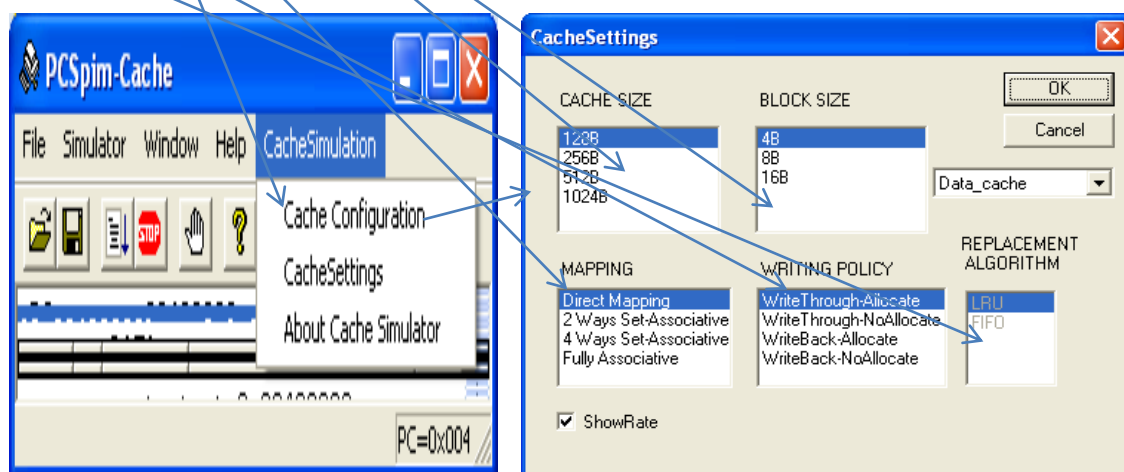
Το μενού CacheSimulation(1/2):

- Διαμόρφωση της μνήμης cache
- Προσομοίωση της cache δεδομένων και της cache εντολών (αρχιτεκτονική harvard)
- Προσομοίωση μόνο της cache δεδομένων
- Προσομοίωση μόνο της cache εντολών



Το μενού CacheSimulation(2/2):

- Ρυθμίσεις της μνήμης cache
- μέγεθος cache
- μέγεθος block
- οργάνωση cache
- πολιτική εγγραφών
- αλγόριθμος αντικατάστασης block



Παράδειγμα: Πρόγραμμα sum.s(1/8):

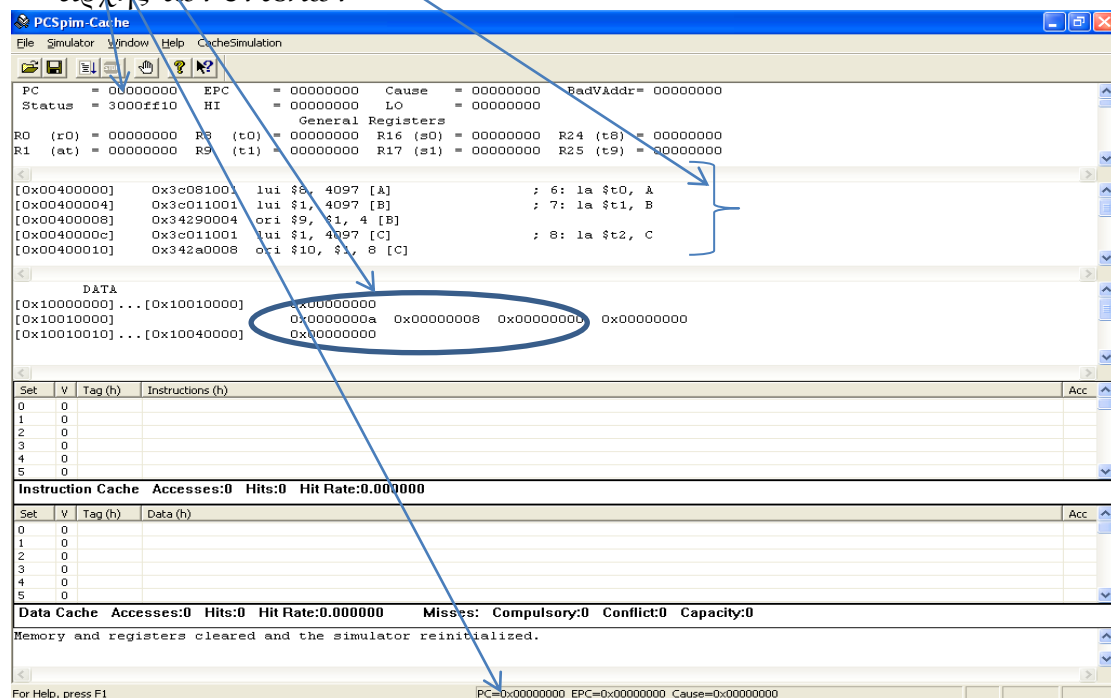
- Χρησιμοποιείστε έναν text editor (π.χ. Notepad) και δημιουργήστε το αρχείο sum.s με το εξής περιεχόμενο:

```
.data
A: .word 10
B: .word 8
C: .word 0
.text
    la $t0, A
    la $t1, B
    la $t2, C
main:
    lw $t3, 0($t0)
    lw $t4, 0($t1)
    add $t5,$t3, $t4
    sw $t5, 0($t2)
    li $v0, 10
    syscall
```

- ❖ Το πρόγραμμα αυτό υπολογίζει το άθροισμα δύο ακεραίων A και B από τη μνήμη και αποθηκεύει το αποτέλεσμα στη θέση μνήμης C
- **Φορτώστε το πρόγραμμα sum.s στον προσομοιωτή PCSpim-Cache και τρέξτε το με την επιλογή “Single Step”.**

Παράδειγμα: Πρόγραμμα sum.s(2/8):

- Οι εντολές (είναι αποθηκευμένες στη μνήμη από τη διεύθυνση: 0x00400000)
- Δεδομένα
- Ο PC ισούται με μηδέν. Πρέπει να αρχικοποιηθεί με τη διεύθυνση αρχής των εντολών



Παράδειγμα: Πρόγραμμα sum.s(3/8):

- Αρχικοποιούμε τον PC=0x400000

The screenshot shows the PCSpin-Cache simulator interface. The main window displays the initial state of the processor, including the PC (00000000), EPC (00000000), Cause (00000000), BadVAddr (00000000), Status (3000ff10), HI (00000000), LO (00000000), and General Registers (R0-R17). The instruction cache and data cache are both empty. A 'Set Value' dialog box is open, allowing the user to set the PC value to 0x400000. The dialog box has a title bar 'Set Value' and buttons for 'OK' and 'Cancel'. The 'Address or Register Name' field is set to 'PC' and the 'Value' field is set to '0x400000'.

Παράδειγμα: Πρόγραμμα sum.s(4/8):

- Η πρώτη εντολή

The screenshot shows the PCSpin-Cache simulator interface after the first instruction has been executed. The PC is now 00400000. The instruction cache has been accessed (Accesses:1) and the hit rate is 0.000000. The instruction cache table shows the first instruction (0x00400000) with tag 0 and hit rate 0.000000. The data cache remains empty. The status bar at the bottom shows the PC value (0x00400000) and the date (Πέμπτη, 19 Μαρτίου 2015).

Παράδειγμα: Πρόγραμμα sum.s(5/8):

- Η πρώτη εντολή προσκομίζεται στην cache εντολών.

The screenshot shows the PCSpm-Cache simulator interface. The top section displays system status: PC = 00400000, EPC = 00000000, Cause = 00000000, BadVAddr = 00000000, Status = 3000ff10, HI = 00000000, LO = 00000000. Below this, the General Registers (R0-R25) are listed with their current values. The instruction stream shows the first instruction: [0x00400000] 0x3c081001 lui \$8, 4097 [A] ; 6: la \$t0, A. The instruction cache table shows Set 0 with V=1, Tag(h)=8000, and the instruction lui \$8, 4097. The cache status is: Instruction Cache Accesses:1 Hits:0 Hit Rate:0.000000. The data cache is empty. A message at the bottom states: Memory and registers cleared and the simulator reinitialized.

Παράδειγμα: Πρόγραμμα sum.s(6/8):

- Αστοχία στην cache δεδομένων, λόγω της εντολής

The screenshot shows the PCSpm-Cache simulator interface at a later stage. The PC is now 00400014. The instruction stream shows the instruction [0x00400014] 0x8d0b0000 lw \$11, 0(\$8) ; 10: lw \$t3, 0(\$t0). The instruction cache table shows Set 5 with V=1, Tag(h)=8000, and the instruction lw \$11, 0(\$8). The cache status is: Instruction Cache Accesses:6 Hits:0 Hit Rate:0.000000. The data cache table shows Set 0 with V=0, Tag(h)=0, and Data(h)=0. The cache status is: Data Cache Accesses:1 Hits:0 Hit Rate:0.000000 Misses: Compulsory:1 Conflict:0 Capacity:0. A red circle highlights the 'miss' status in the data cache table. A blue arrow points from the instruction 'lui \$11, 0(\$8)' to the data cache miss.

Παράδειγμα: Πρόγραμμα sum .s(7/8):

- Μεταφέρεται το δεδομένο

PCSpim-Cache simulator showing the execution of the sum.s(7/8) program. The Instruction Cache is highlighted, showing a miss for the instruction at address 0x00400014. A blue arrow points from the list item 'Μεταφέρεται το δεδομένο' to the Instruction Cache entry.

Set	V	Tag (h)	Instructions (h)	Acc
3	1	8000	lui \$1, 4097	
4	1	8000	ori \$10, \$1, 8	
5	1	8000	lw \$11, 0(\$8)	
6	0			
7	0			
8	0			

Instruction Cache Accesses:6 Hits:0 Hit Rate:0.000000

Set	V	Tag (h)	Data (h)	Acc
0	1	200200	0000000a	miss
1	0			
2	0			
3	0			
4	0			
5	0			

Data Cache Accesses:1 Hits:0 Hit Rate:0.000000 Misses: Compulsory:1 Conflict:0 Capacity:0

For Help, press F1 PC=0x00400014 EPC=0x00000000 Cause=0x00000000

Παράδειγμα: Πρόγραμμα sum .s(8/8):

- Ολοκληρώθηκε το πρόγραμμα
- Στατιστικά

PCSpim-Cache simulator showing the execution of the sum.s(8/8) program. The Instruction Cache is highlighted, showing a miss for the instruction at address 0x00400024. A blue arrow points from the list item 'Στατιστικά' to the Instruction Cache entry.

Set	V	Tag (h)	Instructions (h)	Acc
8	1	8000	sw \$13, 0(\$10)	
9	1	8000	ori \$2, \$0, 10	
10	1	8000	8000	miss
11	0			
12	0			
13	0			

Instruction Cache Accesses:11 Hits:0 Hit Rate:0.000000

Set	V	Tag (h)	Data (h)	Acc
0	1	200200	0000000a	
1	1	200200	00000008	
2	1	200200	00000012	miss
3	0			
4	0			
5	0			

Data Cache Accesses:3 Hits:0 Hit Rate:0.000000 Misses: Compulsory:3 Conflict:0 Capacity:0

For Help, press F1 PC=0x00000000 EPC=0x00000000 Cause=0x00000018

ΜΕΡΟΣ Β: ΑΣΚΗΣΗ

Δίνεται το ακόλουθο τμήμα κώδικα:

```
.data 0x10000480
Array_A: .word 1,1,1,1,2,2,2,2
          .data 0x10000CC0
Array_B: .word 3,3,3,3,4,4,4,4
          .text
          .globl start
start:    li $8, N
ext_loop: la $2, Array_A
          la $3, Array_B
          li $6, 0
          li $4, 8
loop:    lw $5, 0($2)
          lw $7, 0($3)
          add $6, $6, $5
          add $6, $6, $7
          addi $2, $2, 4
          addi $3, $3, 4
          addi $4, $4, -1
          bgt $4, $0, loop
          addi $8, $8, -1
          bgt $8, $0, ext_loop
          .end
```

1. Τρέξτε τον προσομοιωτή PCSpim-Cache:
(α) Διαμορφώστε τη μνήμη cache ώστε να προσομοιώνεται τόσο η cache δεδομένων όσο και η cache εντολών (αρχιτεκτονική Harvard).
(β) Ρυθμίστε τόσο την cache δεδομένων όσο και την cache εντολών ως εξής: μέγεθος cache=128B, μέγεθος block=16B, οργάνωση (mapping)=four-way set associative, πολιτική εγγραφών (writing policy)=Write Through Allocate, αλγόριθμος αντικατάστασης block=LRU.
2. Πόσα blocks(Sets) δημιουργούνται στη cache εντολών και στη cache δεδομένων; (παρατηρήστε τα αντίστοιχα παράθυρα του προσομοιωτή). Εξηγήστε την απάντησή σας.
3. Τρέξτε τον ανωτέρω κώδικα βήμα-βήμα στον προσομοιωτή PCSpim-Cache, με N=5. Εξηγήστε τον τρόπο που διαμορφώνονται τα παράθυρα των εντολών μετά την εκτέλεση κάθε βήματος. Δηλαδή, εξηγήστε την ένδειξη miss ή hit, γιατί η συγκεκριμένη

εντολή μεταφέρθηκε στο συγκεκριμένο block και την τιμή της ετικέτας (Tag) της συγκεκριμένης εντολής.

4. Τρέξτε τον ανωτέρω κώδικα στον προσομοιωτή PCSpim-Cache, με $N=5, 10$ και 100 . Για κάθε N καταγράψτε το hit rate και καταλήξτε σε συμπεράσματα σχετικά με την επίδραση της χρονικής (temporal) τοπικότητας στην απόδοση της cache. Εξηγείστε τα είδη αστοχιών που συμβαίνουν.