

Spim Cache

**Εισαγωγή στη χρήση του
προσομοιωτή**

Αρχιτεκτονική Υπολογιστών II



Spim Cache

- Ελεύθερο Λογισμικό, επέκταση του PCSpim
- Δημιουργήθηκε στο Πολυτεχνείο της Βαλένθια, στην Ισπανία
- Συμβατό με το λειτουργικό σύστημα Windows, τελευταία έκδοση 080605 (2008)
- Διαθέσιμο στην ιστοσελίδα:
 - <http://www.disca.upv.es/spetit/spim.htm>
- Προσομοιώνει τις κρυφές μνήμες (εντολών & δεδομένων) του επεξεργαστή MIPS
- Βασίζεται στον προσομοιωτή SPIM



Πλεονεκτήματα & μειονεκτήματα

- Υποστηρίζει:
 - *Hardvard Architecture*
 - Διαφορετικά μεγέθη *Cache* (4) & *Block* (3)
 - Διαφορετικούς τύπους *Mapping* (*direct-associative*)
 - Διαφορετικές πολιτικές εγγραφής (4)
 - Δύο διαφορετικούς τύπους επανεγγραφής
 - Αρκετά φιλικό περιβάλλον διεπαφής
- Μείον:
 - Δεν υπάρχει ρολόι
 - Όχι τόσο φιλικό περιβάλλον διεπαφής



Από τι αποτελείται

Στο PCSpim-Cache θα δείτε:

- μπάρα μενού (Menu Bar)
- μπάρα εργαλείων (Toolbar - τα βασικά)
- έξι (6) σταθερά και αμετάβλητα υποπαράθυρα:
 - i. Καταχωρητών (εκτός PC, EPC, Cause)
 - ii. Κώδικα – εντολών προς εκτέλεση
 - iii. Κύριας μνήμης (Data, Stack & Kernel data ξεχωριστά)
 - iv. Κρυφής μνήμης εντολών
 - v. Κρυφής μνήμης δεδομένων
 - vi. Μηνυμάτων κατάστασης προσομοιωτή
- Status Bar (Μπάρα κατάστασης)



Παράδειγμα - screenshot

PCSpim-Cache

File Simulator Window Help CacheSimulation

Μπάρα μενού
Μπάρα εργαλείων

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

Καταχωρητές

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

[0x00400000] 0x340c0001 ori \$12, \$0, 1 ; 6: li \$t4, 1
[0x00400004] 0x3c081001 lui \$8, 4097 [num] ; 7: la \$t0, num
[0x00400008] 0x8d090000 lw \$9, 0(\$8) ; 9: lw \$t1, 0(\$t0)
[0x0040000c] 0x212a0002 addi \$10, \$9, 2 ; 10: addi \$t2, \$t1, 2
[0x00400010] 0x3c011001 lui \$1, 4097 [res] ; 11: la \$t0, res
[0x00400014] 0x34280004 ori \$8, \$1, 4 [res]
[0x00400018] 0xad0a0000 sw \$10, 0(\$8) ; 12: sw \$t2, 0(\$t0)

Μνήμη εντολών

DATA
[0x10000000]...[0x10010000] 0x00000000 0x00000000 0x00000000 0x00000000
[0x10010000] 0x0000000d 0x00000000 0x00000000 0x00000000
[0x10010010]...[0x10040000] 0x00000000

Μνήμη δεδομένων

STACK
[0x7ffffeffc] 0x00000000

Set V Tag (h) Instructions (h) Acc

0 0
1 0
2 0
3 0
4 0
5 0
6 0
7 0

Κρυφή μνήμη εντολών

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

Set V Tag (h) Data (h) Acc

0 0
1 0
2 0
3 0
4 0
5 0
6 0
7 0

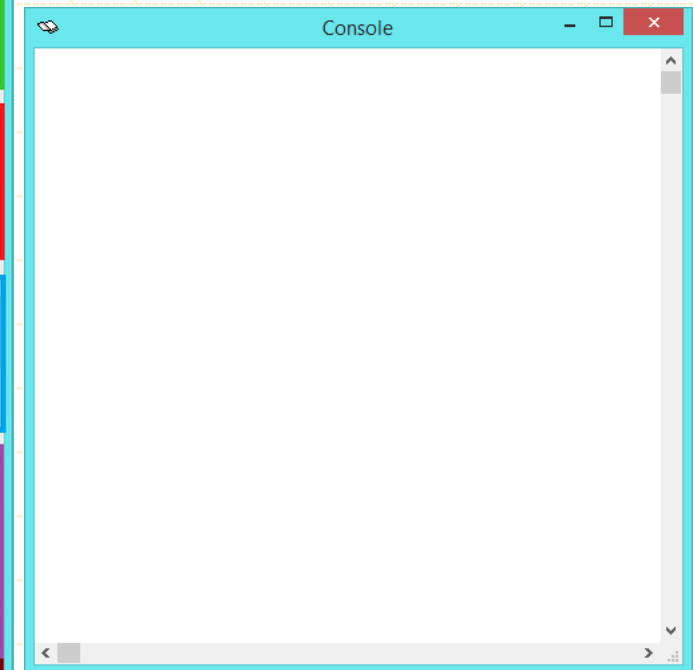
Κρυφή μνήμη δεδομένων

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

Memory and registers cleared and the simulator reinitialized.
F:\architectoniki2\spim-cache\exampleCache.s successfully loaded

Μηνύματα κατάστασης προσομοιωτή

For Help, press F1 Μπάρα κατάστασης PC=0x00000000 EPC=0x00000000 Cause=0x00000000 DELAY LD



Παράθυρα (1) - Καταχωρητές

- Αναπαράσταση:
 - Καταχωρητών διευθύνσεων, status, HI, LO
 - Γενικού σκοπού (General Registers) με aliases & χωρίς
 - Μονής & διπλής υποδιαστολής (Floating Point Registers)

```
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) = 7ffffeffc
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
```



Παράθυρα (2) - Κώδικας

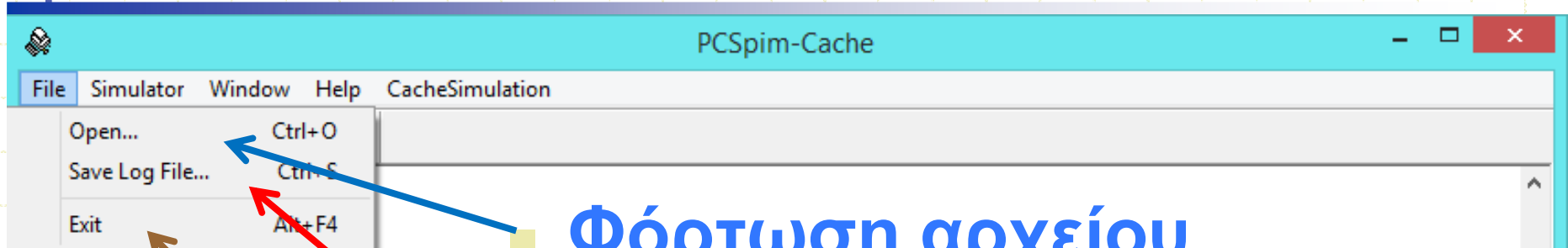
- Αναπαράσταση 4 στηλών της μνήμης εντολών, δείχνοντας από αριστερά προς τα δεξιά:
 1. την διεύθυνση
 2. την hex αναπαράσταση της εντολής σε κώδικα μηχανής
 3. την μεταφρασμένη εντολή *assembly*
 4. την αρχική εντολή *assembly*

```
<
[0x00400000] 0x340c0001 ori $12, $0, 1           ; 6: li $t4,1
[0x00400004] 0x3c081001 lui $8, 4097 [num]         ; 7: la $t0,num
[0x00400008] 0x8d090000 lw $9, 0($8)       ; 9: lw $t1,0($t0)
[0x0040000c] 0x212a0002 addi $10, $9, 2    ; 10: addi $t2,$t1,2
[0x00400010] 0x3c011001 lui $1, 4097 [res] ; 11: la $t0,res
[0x00400014] 0x34280004 ori $8, $1, 4 [res]
[0x00400018] 0xad0a0000 sw $10, 0($8)      ; 12: sw $t2,0($t0)
[0x0040001c] 0x216b0001 addi $11, $11, 1   ; 13: addi $t3,$t3,1
[0x00400020] 0x516cffff beql $11, $12, -24 [label1-0x00400020]; 14: beq $t3,$t4,label1
[0x00400024] 0x3402000a ori $2, $0, 10     ; 15: li $v0,10 #au revoir...
[0x00400028] 0x0000000c syscall            ; 16: syscall
```

KERNEL



Μενού File (Menu Bar-1)



Φόρτωση αρχείου

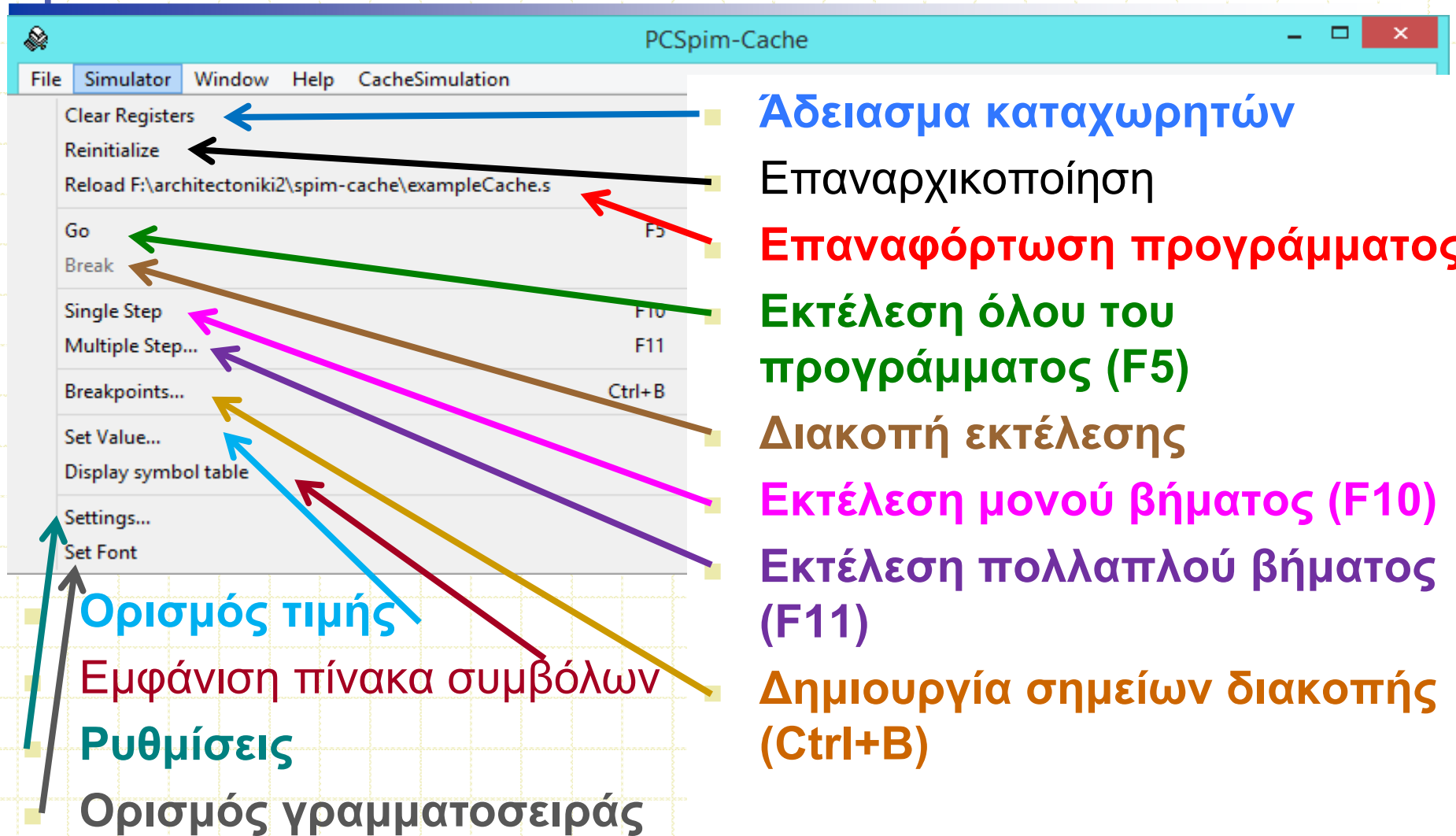
Επιλέγουμε το αρχείο .s & **Open**

**Αποθήκευση αρχείου
καταγραφής**

Έξοδος



Μενού Simulator (Menu Bar-2)

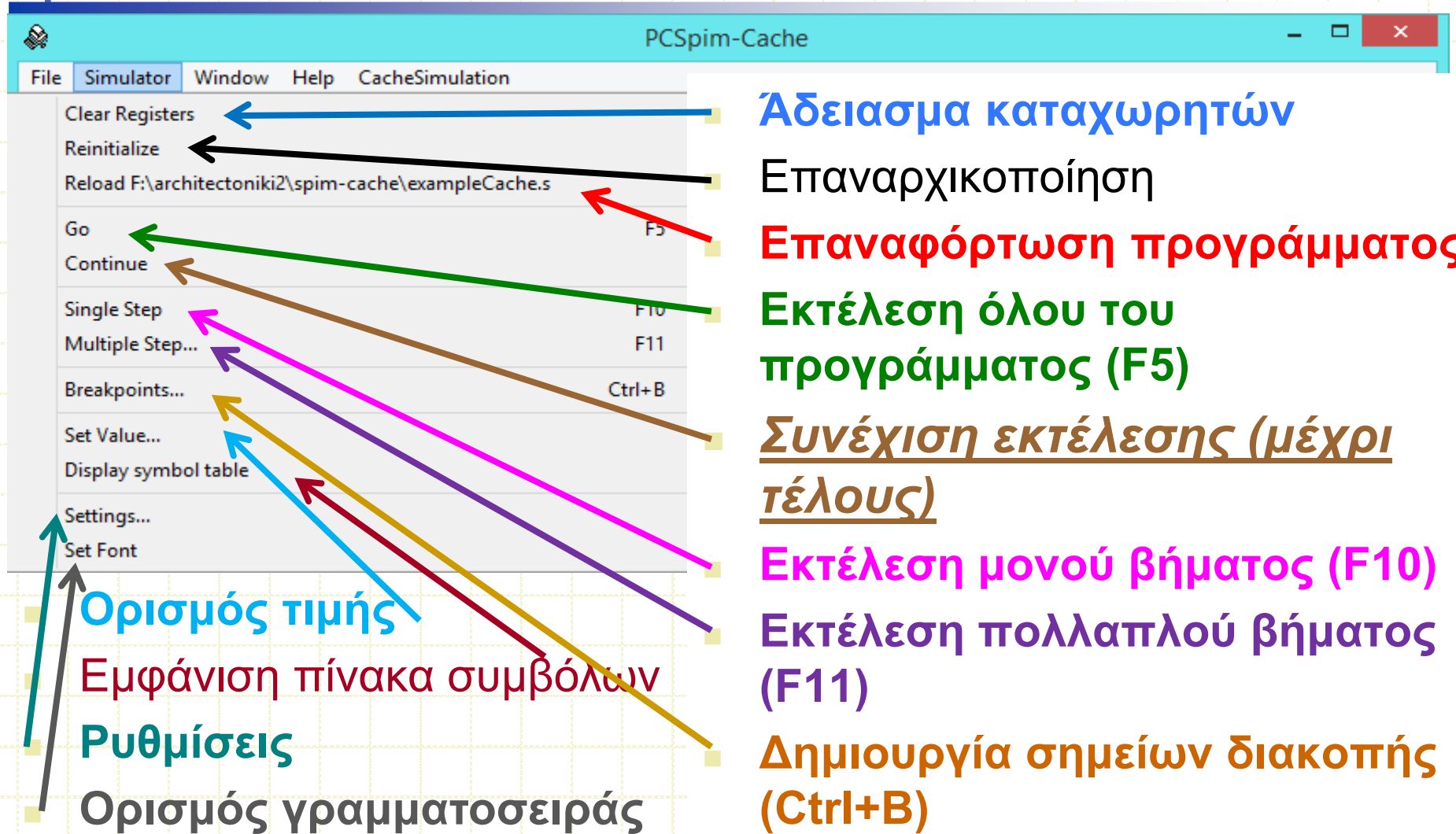


The screenshot shows the 'Simulator' menu of the PCSpim-Cache application. The menu items and their corresponding functions, as indicated by colored arrows, are:

- Clear Registers** (Blue arrow): Άδειασμα καταχωρητών
- Reinitialize** (Black arrow): Επαναρχικοποίηση
- Reload F:\architectoniki2\spim-cache\exampleCache.s** (Red arrow): Επαναφόρτωση προγράμματος
- Go** (Green arrow, F5): Εκτέλεση όλου του προγράμματος (F5)
- Break** (Brown arrow, Ctrl+B): Διακοπή εκτέλεσης
- Single Step** (Magenta arrow, F10): Εκτέλεση μονού βήματος (F10)
- Multiple Step...** (Purple arrow, F11): Εκτέλεση πολλαπλού βήματος (F11)
- Breakpoints...** (Orange arrow): Δημιουργία σημείων διακοπής (Ctrl+B)
- Set Value...** (Light blue arrow): Ορισμός τιμής
- Display symbol table** (Dark red arrow): Εμφάνιση πίνακα συμβόλων
- Settings...** (Teal arrow): Ρυθμίσεις
- Set Font** (Dark grey arrow): Ορισμός γραμματοσειράς



Μενού Simulator (Menu Bar-3)

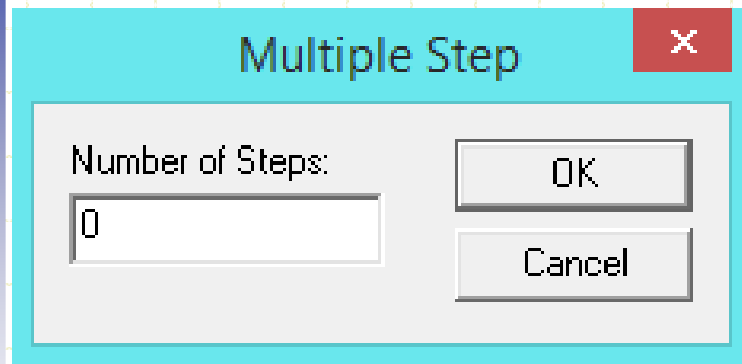


The screenshot shows the 'Simulator' menu of the PCSpim-Cache application. The menu items and their corresponding annotations are as follows:

- Clear Registers**: Αδειασμα καταχωρητών (Blue arrow)
- Reinitialize**: Επαναρχικοποίηση (Black arrow)
- Reload F:\architectoniki2\spim-cache\exampleCache.s**: Επαναφόρτωση προγράμματος (Red arrow)
- Go**: Εκτέλεση όλου του προγράμματος (F5) (Green arrow)
- Continue**: Συνέχιση εκτέλεσης (μέχρι τέλους) (Brown arrow)
- Single Step**: Εκτέλεση μονού βήματος (F10) (Magenta arrow)
- Multiple Step...**: Εκτέλεση πολλαπλού βήματος (F11) (Purple arrow)
- Breakpoints...**: Δημιουργία σημείων διακοπής (Ctrl+B) (Orange arrow)
- Set Value...**: Ορισμός τιμής (Light blue arrow)
- Display symbol table**: Εμφάνιση πίνακα συμβόλων (Dark red arrow)
- Settings...**: Ρυθμίσεις (Teal arrow)
- Set Font**: Ορισμός γραμματοσειράς (Black arrow)



Μενού Simulator (Menu Bar-4)

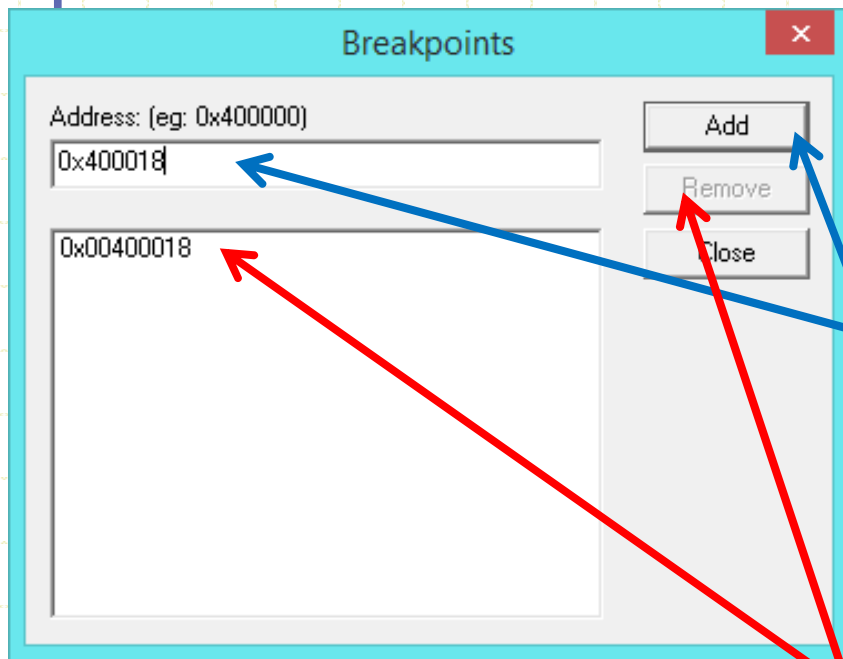


Επιλέγοντας από την μπάρα του μενού ***Simulator→Multi-Step...*** ή ***F11***, & θα εμφανιστεί το παραπάνω παράθυρο.

Αλλάζοντας την τιμή μεταβάλλουμε το βήμα της πολλαπλής εκτέλεσης.



Μενού Simulator (Menu Bar-5)



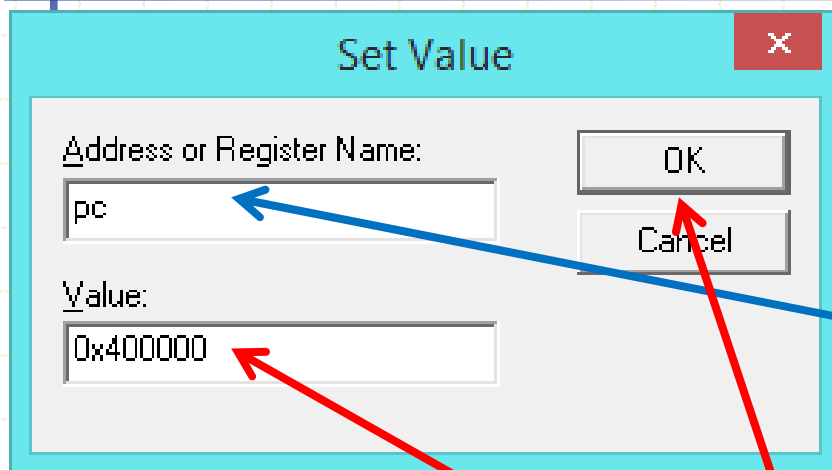
Επιλέγοντας από την μπάρα του μενού **Simulator** → **Breakpoints...** ή **Ctrl+B**, & θα εμφανιστεί το διπλανό παράθυρο.

Δίνοντας την διεύθυνση μνήμης της εντολής στην οποία θέλουμε να δημιουργήσουμε σημείο διακοπής & πατώντας **“Add”** ολοκληρώνουμε τη δημιουργία.

Αντίθετα αν από το κάτω μέρος του παραθύρου επιλέξουμε ένα υπάρχον σημείο & πατήσουμε **“Remove”**, τότε το απομακρύνουμε.



Μενού Simulator (Menu Bar-6)



Set Value

Address or Register Name:

pc

Value:

0x400000

OK

Cancel

Επιλέγοντας από την μπάρα του μενού **Simulator→Set value...** θα εμφανιστεί το διπλανό παράθυρο.

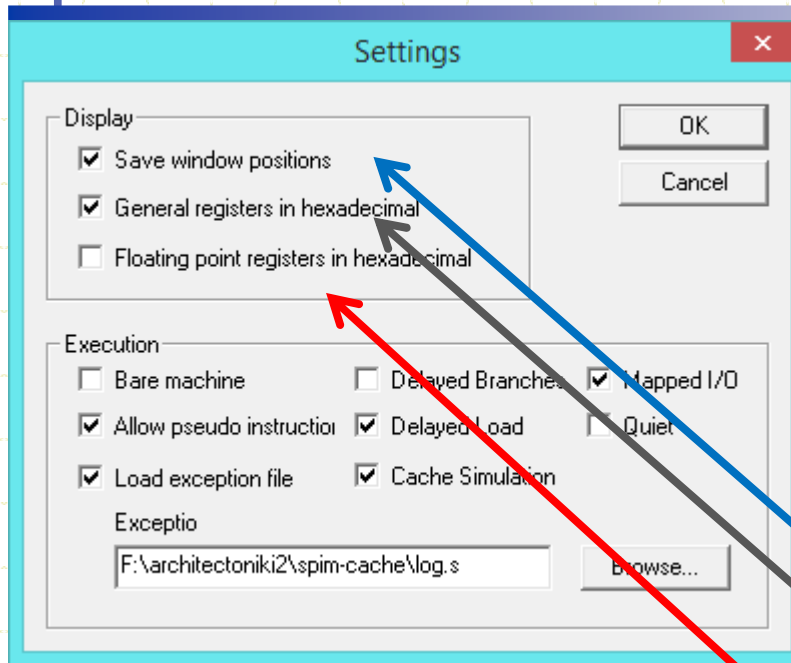
Πάνω δίνουμε τον καταχωρητή ή την διεύθυνση μνήμης της οποίας θέλουμε να αλλάξουμε την τιμή.

Αντίθετα, στο κάτω μέρος του παραθύρου δίνουμε την τιμή που θέλουμε να πάρει ο αντίστοιχος καταχωρητής ή η θέση μνήμης.

Για να ολοκληρωθεί η διαδικασία πατάμε **“Ok”**.



Μενού Simulator (Menu Bar-7)



Επιλέγοντας από την μπάρα του μενού **Simulator** → **Settings...** θα εμφανιστεί το διπλανό παράθυρο στο οποίο μπορούμε να ρυθμίσουμε διάφορες παραμέτρους σχετικά με την εμφάνιση & εκτέλεση του προγράμματος.

Ρυθμίσεις εμφάνισης:

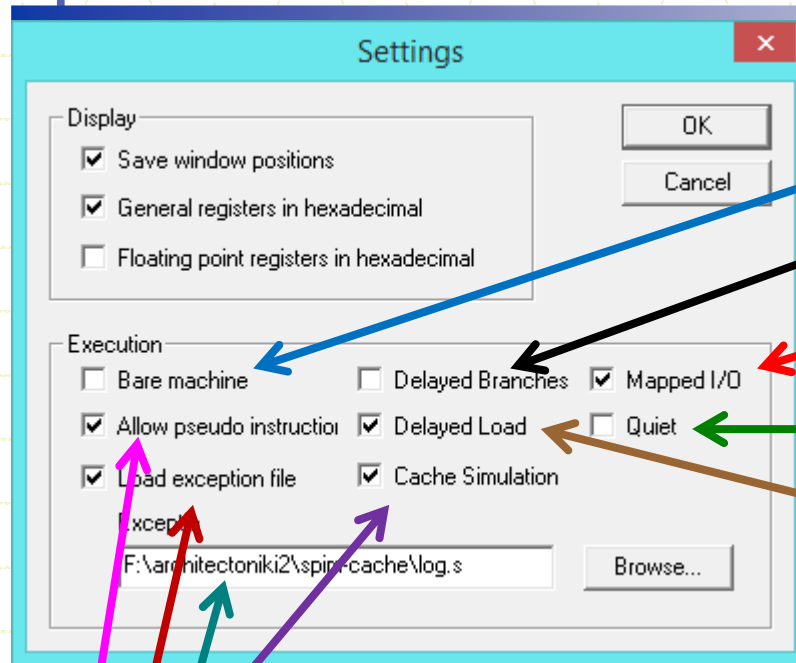
Αποθήκευση θέσεων παραθύρων

Οι γενικοί καταχωρητές σε δεκαεξαδικές τιμές

Οι καταχωρητές κινητής υποδιαστολής σε δεκαεξαδικές τιμές



Μενού Simulator (Menu Bar-8)



Ρυθμίσεις εκτέλεσης:

Bare machine

Καθυστέρηση διακλαδώσεων

Mapped I/O

Ήσυχη εκτέλεση

Μεγαλύτερη καθυστέρηση
στις εντολές load

Να επιτρέπεται η χρήση ψευδοεντολών

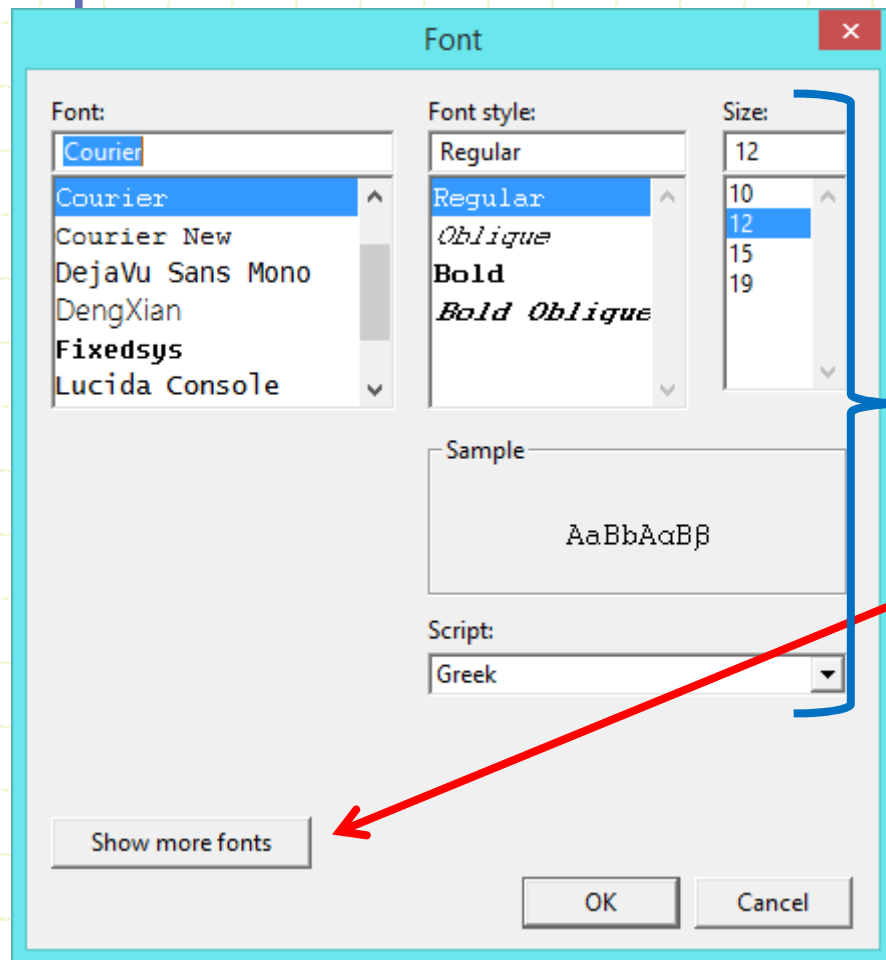
Προσομοίωση της κρυφής μνήμης

Φόρτωση αρχείου εξαιρέσεων

Το αρχείο εξαιρέσεων



Μενού Simulator (Menu Bar-9)



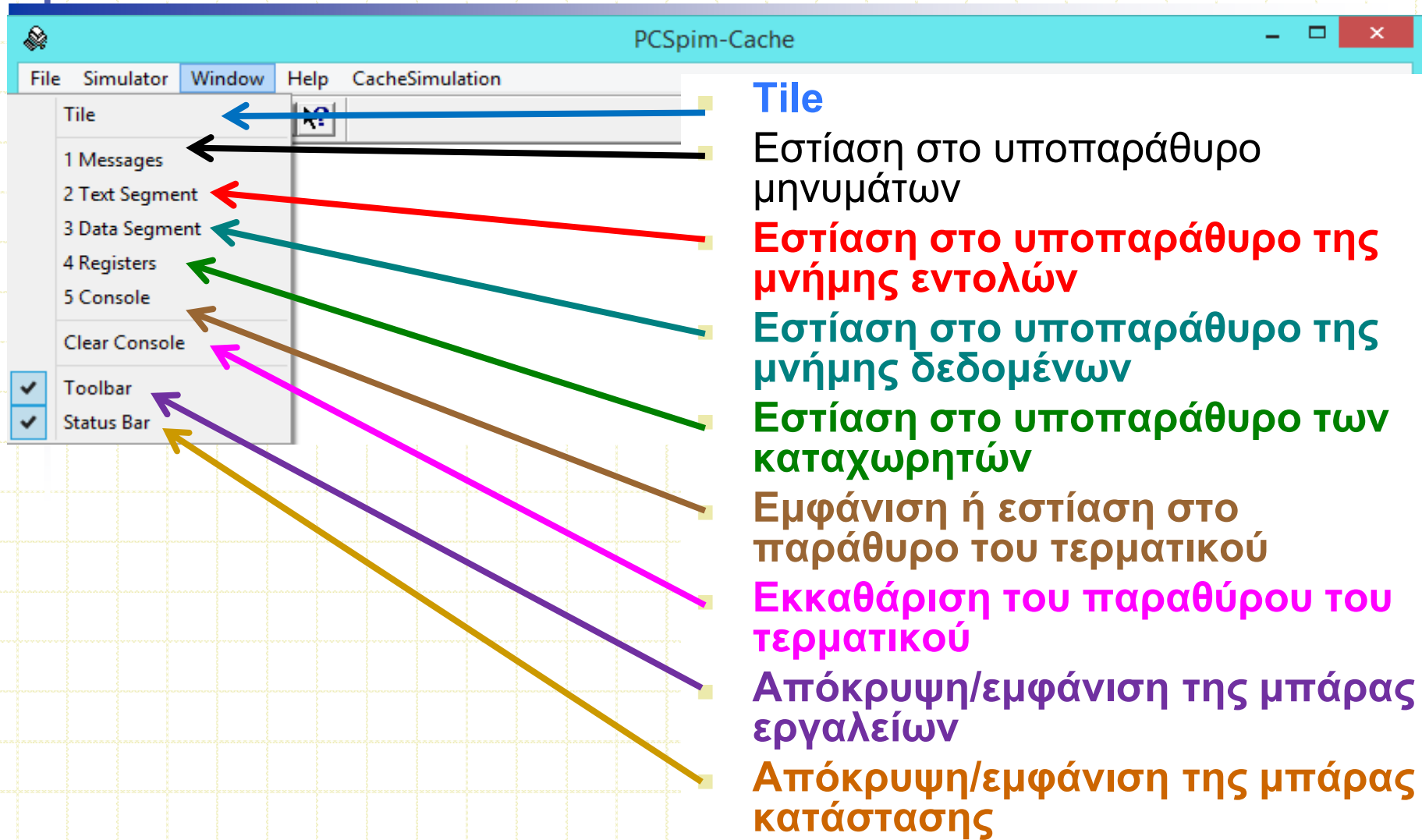
Επιλέγοντας από την μπάρα του μενού **Simulator** → **Set Font...** θα εμφανιστεί το διπλανό παράθυρο.

■ Ρύθμιση γραμματοσειρών

■ Προβολή περισσότερων γραμματοσειρών, δηλ. το σύνολο γραμματοσειρών του συστήματος



Μενού Window (Menu Bar-10)

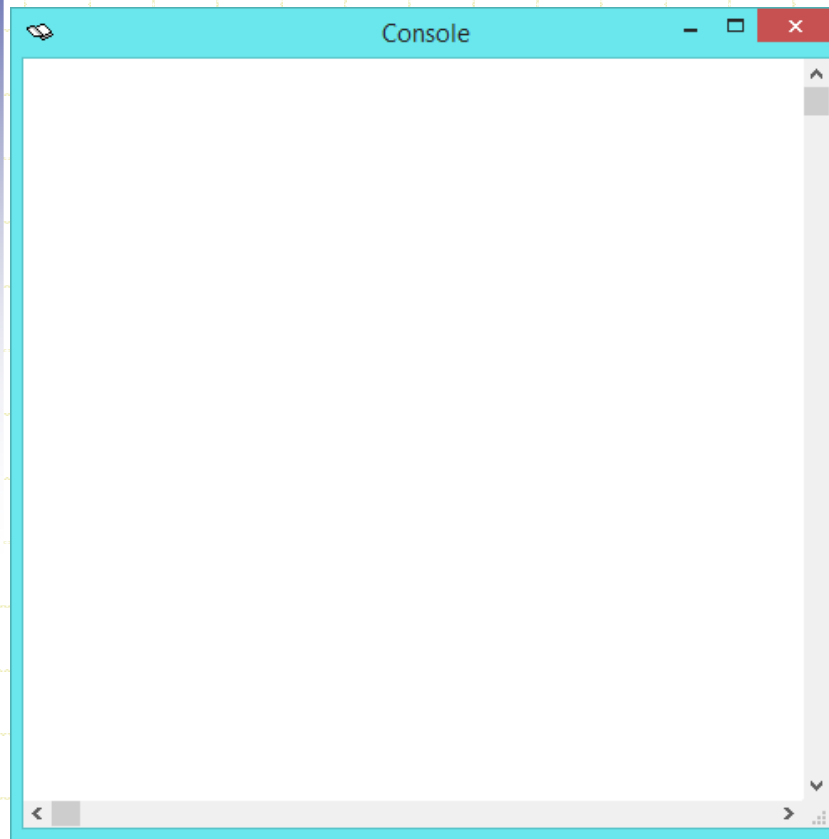


The screenshot shows the 'Window' menu of the PCSpim-Cache application. The menu items are: Tile, 1 Messages, 2 Text Segment, 3 Data Segment, 4 Registers, 5 Console, Clear Console, Toolbar (checked), and Status Bar (checked). Colored arrows point from each item to a corresponding text description on the right:

- Tile** (blue arrow): Εστίαση στο υποπαράθυρο μηνυμάτων
- 1 Messages** (black arrow): Εστίαση στο υποπαράθυρο της μνήμης εντολών
- 2 Text Segment** (red arrow): Εστίαση στο υποπαράθυρο της μνήμης δεδομένων
- 3 Data Segment** (teal arrow): Εστίαση στο υποπαράθυρο των καταχωρητών
- 4 Registers** (green arrow): Εμφάνιση ή εστίαση στο παράθυρο του τερματικού
- 5 Console** (brown arrow): Εκκαθάριση του παραθύρου του τερματικού
- Clear Console** (magenta arrow): Απόκρυψη/εμφάνιση της μπάρας εργαλείων
- Toolbar** (purple arrow): Απόκρυψη/εμφάνιση της μπάρας κατάστασης
- Status Bar** (yellow arrow):



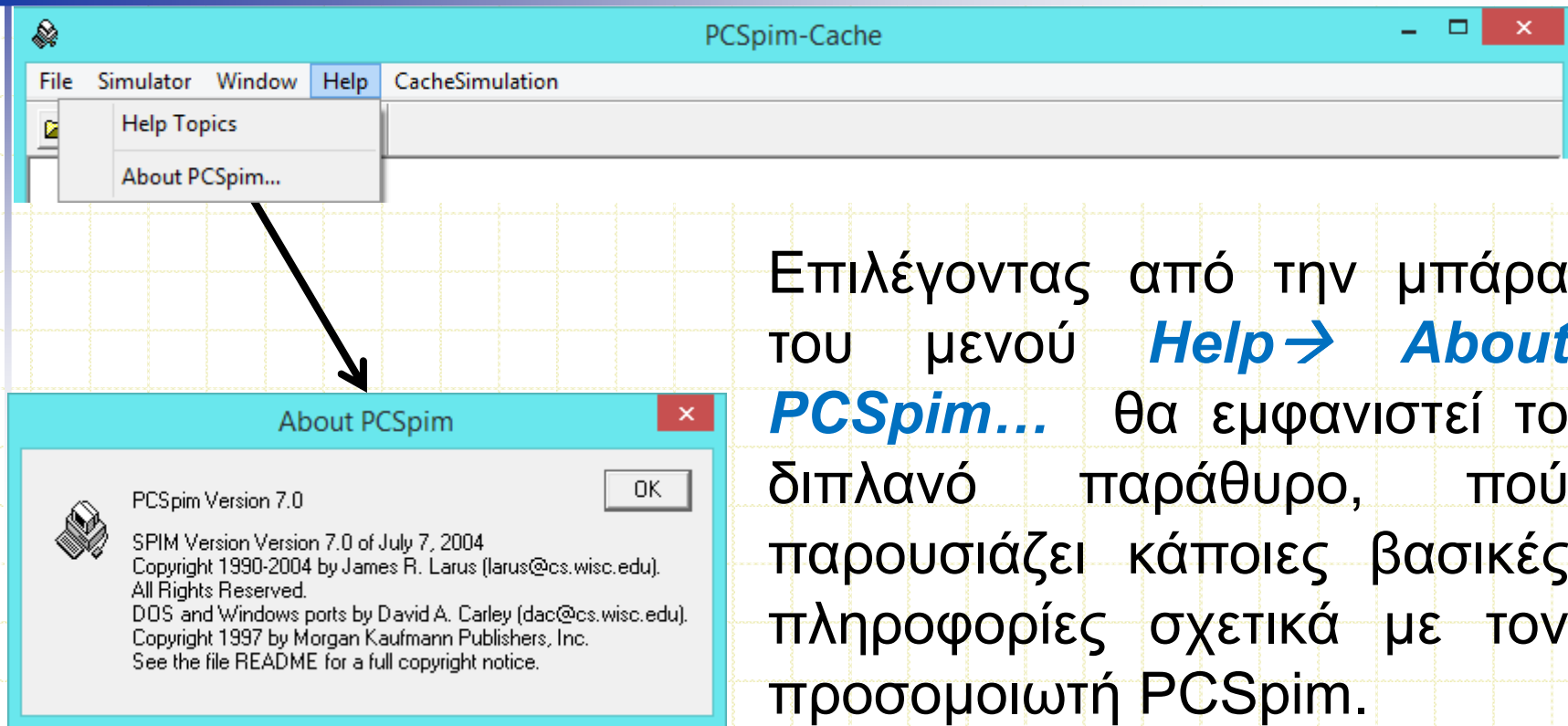
Μενού Window (Menu Bar-11)



Επιλέγοντας από την μπάρα του μενού **Window** → **Console** θα εμφανιστεί το διπλανό παράθυρο, πού είναι το παράθυρο του Τερματικού, που αποτελεί το βασικό Input/Output του προσομοιωτή.



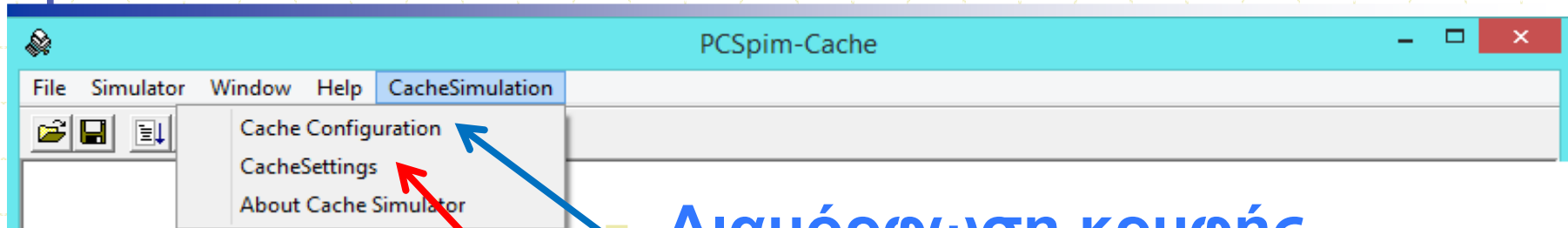
Μενού Help (Menu Bar-12)



Επιλέγοντας από την μπάρα του μενού **Help** → **About PCSpim...** θα εμφανιστεί το διπλανό παράθυρο, πού παρουσιάζει κάποιες βασικές πληροφορίες σχετικά με τον προσομοιωτή PCSpim.



Μενού CacheSimulator (Menu Bar-13)



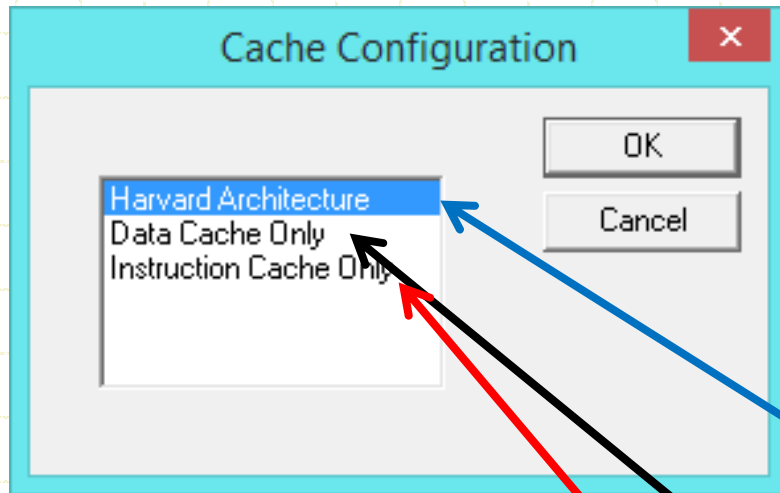
Διαμόρφωση κρυφής
μνήμης

Ρυθμίσεις κρυφής μνήμης

Σχετικά με τον
Προσομοιωτή Κρυφών
Μνημών



Μενού CacheSimulator (Menu Bar-14)



Επιλέγοντας από την μπάρα του μενού **CacheSimulator** → **Cache Configuration** θα εμφανιστεί το διπλανό παράθυρο.

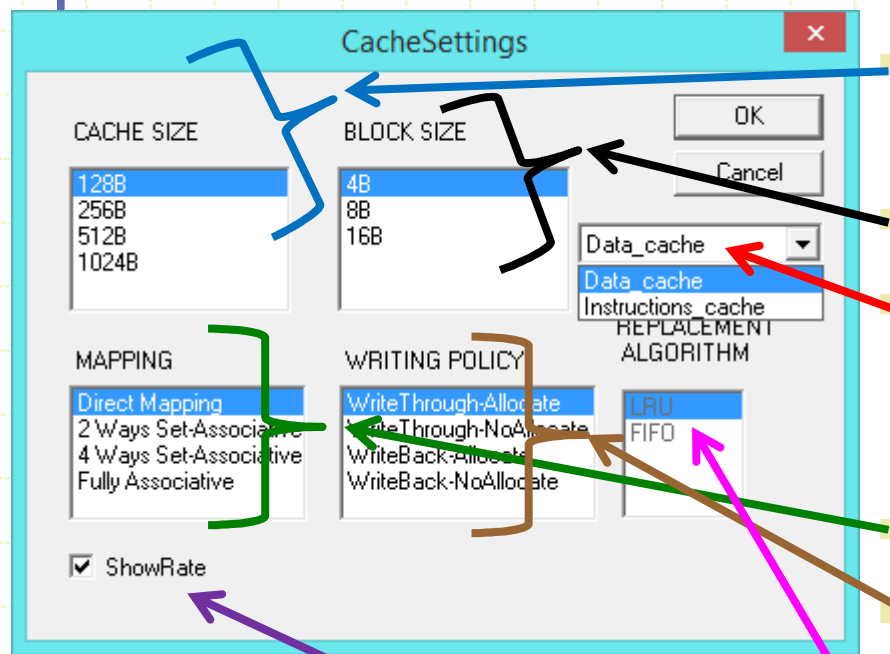
Χρήση αρχιτεκτονικής Harvard

Προσομοίωση μόνο της κρυφής μνήμης δεδομένων

Προσομοίωση μόνο της κρυφής μνήμης εντολών



Μενού CacheSimulator (Menu Bar-15)



Ρύθμιση μεγέθους της κρυφής μνήμης

Ρύθμιση του μεγέθους του block

Εναλλαγή ρυθμίσεων μεταξύ κρυφής μνήμης δεδομένων & εντολών

Επιλογές χαρτογράφησης

Πολιτική εγγραφών στην κρυφή μνήμη

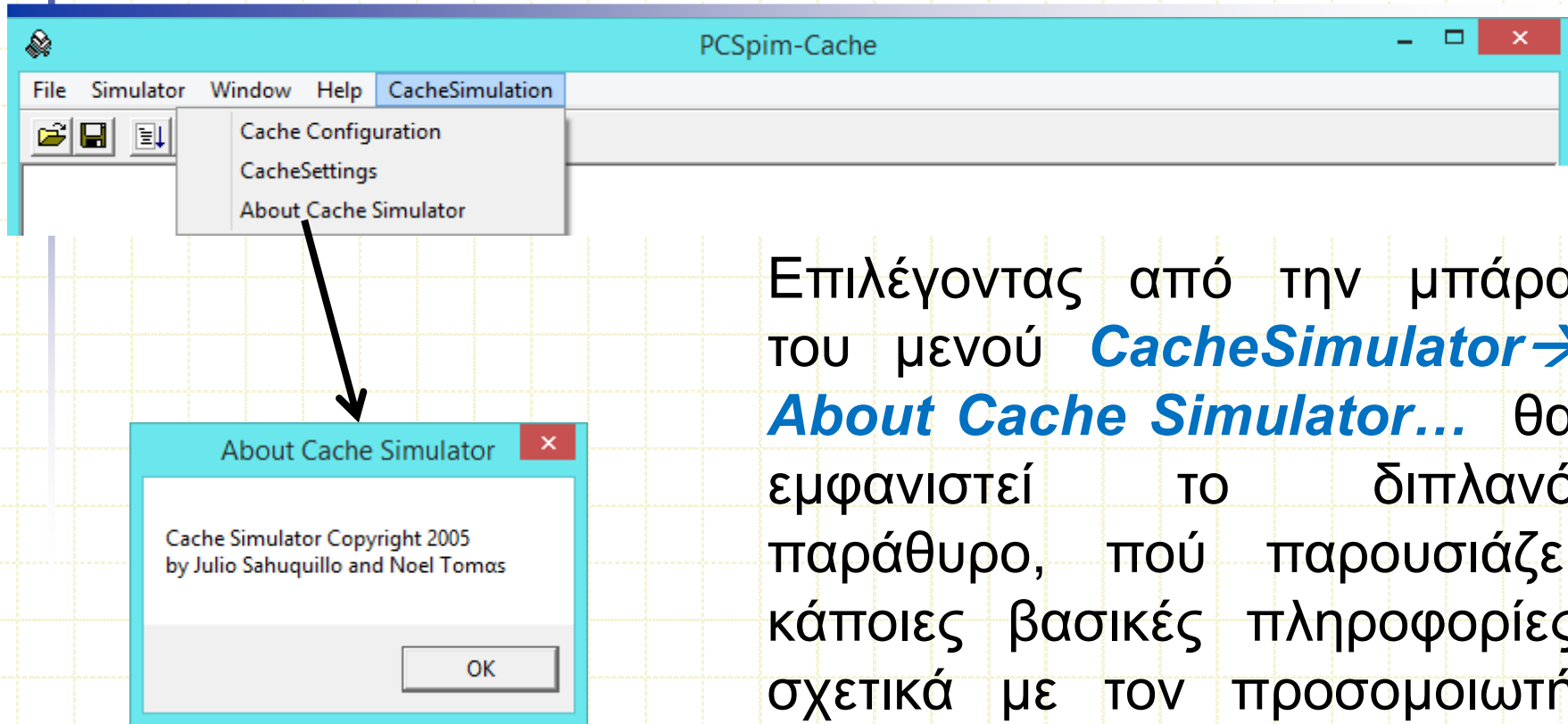
Εν χρήσει αλγόριθμος αντικατάστασης εγγραφών

Εμφάνιση ρυθμού

Επιλέγοντας από την μπάρα του μενού **CacheSimulator** → **Cache Settings** θα εμφανιστεί το παραπάνω παράθυρο.



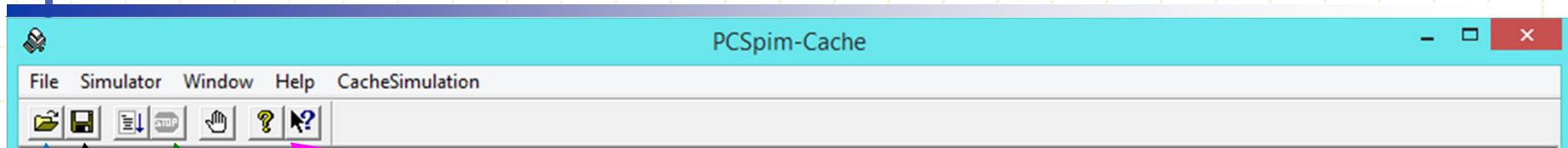
Μενού CacheSimulator (Menu Bar-16)



Επιλέγοντας από την μπάρα του μενού **CacheSimulator** → **About Cache Simulator...** θα εμφανιστεί το διπλανό παράθυρο, πού παρουσιάζει κάποιες βασικές πληροφορίες σχετικά με τον προσομοιωτή κρυφών μνημών.



Γραμμή εργαλείων (Toolbar)



Βοήθεια

Δημιουργία σημείου διακοπής εκτέλεσης

Διακοπή εκτέλεσης

Εκτέλεση προγράμματος (F5)

Αποθήκευση αρχείου

Άνοιγμα αρχείου



Γραμμή κατάστασης (Status Bar)

For Help, press F1

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



Η γραμμή κατάστασης βρίσκεται χαμηλά & συνήθως δείχνει αριστερά “For Help, press F1”, στο κέντρο τους καταχωρητές PC, EPC & Cause με τις αντίστοιχες δεκαεξαδικές τιμές τους.

Όμως κατά την διάρκεια εκτέλεσης ενός προγράμματος παρέχει χρήσιμες πληροφορίες σχετικά με την τρέχουσα κατάσταση του προσομοιωτή.



Σημεία διακοπής - Breakpoints

The screenshot shows the PCSpim Cache simulator interface. The assembly code window displays the following instructions:

```
[0x00400008] 0x8d090000 lw $9, 0($8) ; 9: lw $t1,0($t0)
[0x0040000c] 0x212a0002 addi $10, $9, 2 ; 10: addi $t2,$t1,2
[0x00400010] 0x3c011001 lui $1, 4097 [res] ; 11: la $t0,res
[0x00400014] 0x34280004 ori $8, $1, 4 [res]
[0x00400018] 0x00000000 break $1
[0x0040001c] 0x216b0001 addi $11, $11, 1 ; 13: addi $t3,$t3,1
[0x00400020] 0x516cfff9 beql $11, $12, -28 [label1-0x00400020]; 14: beq $t3,$t4,label1
```

A blue arrow points from the text "break \$1" in the code to the "break \$1" instruction in the assembly list. Another blue arrow points from the text "break \$1" in the assembly list to the "break \$1" instruction in the assembly list.

Below the assembly code, there is a text box with the following text:

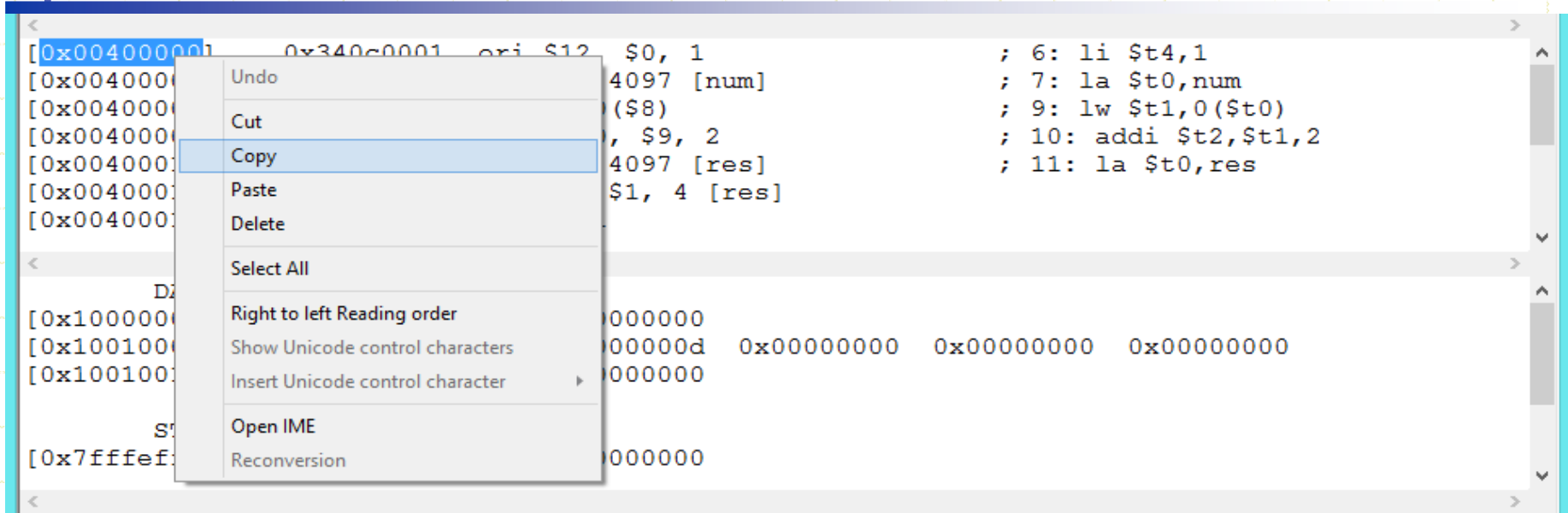
Για να εισάγουμε ένα σημείο διακοπής κάπου στον κώδικα του Επιλέγοντας από την μπάρα του μενού **Simulator** → **Breakpoints...** θα εμφανιστεί το διπλανό παράθυρο.

The "Breakpoints" dialog box is shown, with the "Address: (eg: 0x400000)" field containing "0x400018". The "Add" button is highlighted.

- Βάζουμε την διεύθυνση μνήμης της εντολής στο πάνω κενό & πατάμε **"Add"**. Τότε θα προστεθεί στο κάτω μέρος του παραθύρου η πλήρης HEX διεύθυνση της εντολής στην οποία θέλουμε το σημείο διακοπής, & στον κώδικα θα προστεθεί μία εντολή **"break \$1"**.



Παράθυρα (1)



- Στα υποπαράθυρα έχουμε την δυνατότητα με δεξί κλικ να κάνουμε διάφορες λειτουργίες, όπως το να αντιγράψουμε την διεύθυνση μνήμης μίας εντολής. Μπορούμε δηλαδή να χειριστούμε μόνο οπτικά το περιεχόμενό τους σαν text, όχι όμως & να αλλάξουμε τις τιμές. Αυτό γίνεται με την επιλογή **“Set value”**.



Παράθυρα – Κρυφής μνήμης εντολών (2)

Set	V	Tag (h)	Instructions (h)	Acc
8	1	8000	beql \$11, \$12, -28	
9	1	8000	ori \$2, \$0, 10	
10	1	8000	8000	m...
11	0			
12	0			
13	0			
14	0			
15	0			

Instruction Cache Accesses:18 Hits:6 Hit Rate:0.333333

Εντολές

Miss

Στατιστικά κρυφής μνήμης εντολών:

- Αριθμός προσβάσεων (Accesses)
- Ευστοχίες (Hits)
- Ρυθμός Ευστοχίας (Hit Rate)



Παράθυρα – Κρυφής μνήμης δεδομένων (3)

Set	V	Tag (h)	Data (h)	Acc
0	1	200200	000000d	
1	1	200200	000000f	hit
2	0			
3	0			
4	0			
5	0			
6	0			
7	0			

Εντολές (black arrow pointing to the Data column)

Hit (blue arrow pointing to the 'hit' status in the Acc column)

Data Cache Accesses:4 Hits:2 Hit Rate:0.500000 Misses: Compulsory:2 Conflict:0 Capacity:0

Στατιστικά κρυφής μνήμης δεδομένων:

- Αριθμός προσβάσεων (Accesses)
- Ευστοχίες (Hits)
- Ρυθμός Ευστοχίας (Hit Rate)
- Αστοχίες (Misses)
 1. Υποχρεωτικές (Compulsory)
 2. Συγκρούσεων (Conflicts)
 3. Χωρητικότητας (Capacity)



Που γράφουμε τον κώδικά μας;

- **Μόνο σε text editor** (Notepad στα Windows, ή κάποιον άλλο Notepad++ κτλ)
- **Όχι σε επεξεργαστή κειμένου (Word)**
 - Έχει ειδικούς χαρακτήρες
- Τα αρχεία assembly MIPS έχουν κατάληξη **.s**
- Χρησιμοποιώντας έναν text editor δημιουργήστε το αρχείο “exampleCache.s” με το παρακάτω περιεχόμενο:



Πρόγραμμα exampleCache.s (1)

```
.data
num:  .word 13
res:  .word 0

.text
li $t4,1
la $t0,num

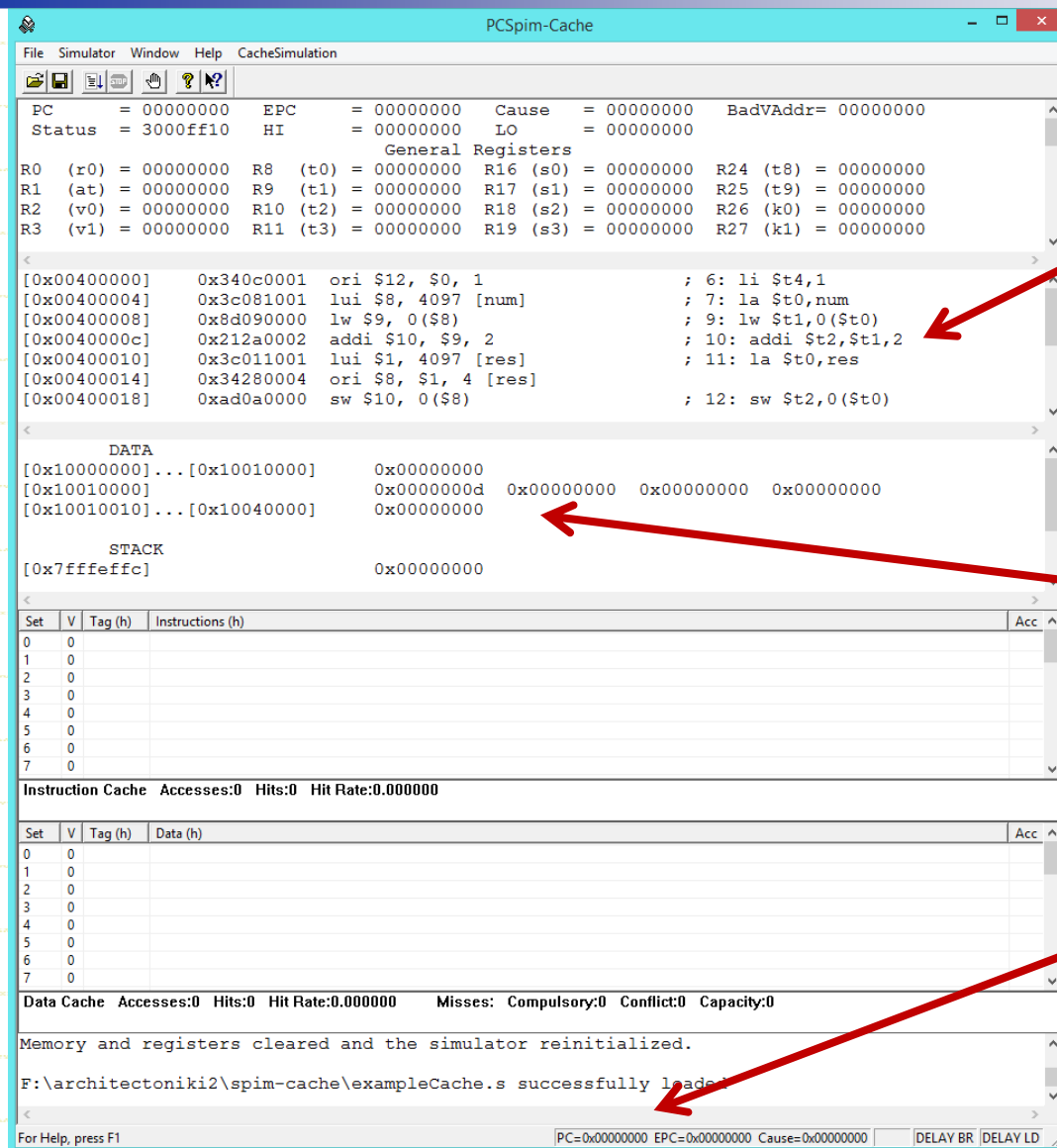
main:
lw $t1,0($t0)
addi $t2,$t1,2
la $t0,res
sw $t2,0($t0)
addi $t3,$t3,1
beq $t3,$t4,label1
li $v0,10 #au revoir...
syscall
```

Αυτό το πρόγραμμα
επαναληπτικά γράφει &
διαβάζει από την μνήμη.

Επιλέξτε “***File/Open***” & εντοπίστε
τον φάκελο στον οποίο έχετε
αποθηκεύσει το αρχείο
“***exampleCache.s***” που
δημιουργήσατε.



Πρόγραμμα exampleCache.s (2)



The screenshot shows the PCSpim-Cache simulator interface. At the top, the title bar reads "PCSpim-Cache". Below it is a menu bar with "File", "Simulator", "Window", "Help", and "CacheSimulation". A toolbar with various icons is below the menu. The main display area is divided into several sections:

- Registers:** A table showing the state of various registers. PC is 00000000, EPC is 00000000, Cause is 00000000, and BadVAddr is 00000000. Status is 3000ff10, HI is 00000000, LO is 00000000. General Registers R0 through R31 are listed with their values.
- Instructions:** A list of instructions with their addresses and assembly code. For example, [0x00400000] 0x340c0001 ori \$12, \$0, 1 ; 6: li \$t4,1.
- DATA:** A section showing memory addresses and their corresponding data values. For example, [0x10000000]...[0x10010000] 0x00000000.
- STACK:** A section showing the stack pointer and its value. For example, [0x7fffffc] 0x00000000.
- Cache Tables:** Two tables showing the state of the Instruction Cache and Data Cache. Both have 8 sets, each with a valid bit (V), tag (h), and data (h). The Instruction Cache has a hit rate of 0.000000. The Data Cache has a hit rate of 0.000000 and a capacity of 0.
- Status Bar:** At the bottom, it shows "Memory and registers cleared and the simulator reinitialized." and "F:\architectoniki2\spim-cache\exampleCache.s successfully loaded".

οι εντολές έχουν μεταφραστεί & φορτωθεί στη μνήμη εντολών, ξεκινώντας πάντα από την θέση "0x00400000"

οι τιμές των δεδομένων έχουν φορτωθεί στη μνήμη δεδομένων.

ο PC έχει την τιμή "0" ενώ πρέπει να πάρει την τιμή "0x400000" για μπόρέσει το πρόγραμμα να εκτελεστεί σωστά.



Πρόγραμμα exampleCache.s (3)

The screenshot shows the PCSpim-Cache simulator interface. At the top, the title bar reads "PCSpim-Cache". Below it is a menu bar with "File", "Simulator", "Window", "Help", and "CacheSimulation". A toolbar with various icons is below the menu bar. The main window is divided into several sections:

- Registers:** A table showing the state of various registers. The PC register is highlighted with a value of 00000000.
- General Registers:** A table showing the state of registers R0 through R31. R0 (r0) is 00000000, R1 (at) is 00000000, R2 (v0) is 00000000, R3 (v1) is 00000000, R8 (t0) is 00000000, R9 (t1) is 00000000, R10 (t2) is 00000000, R11 (t3) is 00000000, R16 (s0) is 00000000, R17 (s1) is 00000000, R18 (s2) is 00000000, R19 (s3) is 00000000, R24 (t8) is 00000000, R25 (t9) is 00000000, R26 (k0) is 00000000, and R27 (k1) is 00000000.
- Instructions:** A list of instructions being executed. The first instruction is "ori \$12, \$0, 1" at address 0x00400000. The second instruction is "lui \$8, 4097 [num]" at address 0x00400004. The third instruction is "lw \$9, 0(\$8)" at address 0x00400008. The fourth instruction is "addi \$10, \$9, 2" at address 0x0040000c. The fifth instruction is "lui \$1, 4097 [res]" at address 0x00400010. The sixth instruction is "ori \$8, \$1, 4 [res]" at address 0x00400014. The seventh instruction is "sw \$10, 0(\$8)" at address 0x00400018.
- DATA:** A table showing the state of data memory. The first row is [0x10000000]...[0x10010000] with a value of 0x00000000. The second row is [0x10010000] with a value of 0x0000000d. The third row is [0x10010010]...[0x10040000] with a value of 0x00000000.
- STACK:** A table showing the state of the stack. The first row is [0x7fffffc] with a value of 0x00000000.
- Instruction Cache:** A table showing the state of the instruction cache. The first row is Set 0 with a value of 0. The second row is Set 1 with a value of 0. The third row is Set 2 with a value of 0. The fourth row is Set 3 with a value of 0. The fifth row is Set 4 with a value of 0. The sixth row is Set 5 with a value of 0. The seventh row is Set 6 with a value of 0. The eighth row is Set 7 with a value of 0.
- Data Cache:** A table showing the state of the data cache. The first row is Set 0 with a value of 0. The second row is Set 1 with a value of 0. The third row is Set 2 with a value of 0. The fourth row is Set 3 with a value of 0. The fifth row is Set 4 with a value of 0. The sixth row is Set 5 with a value of 0. The seventh row is Set 6 with a value of 0. The eighth row is Set 7 with a value of 0.
- Memory and registers cleared and the simulator reinitialized.**
- F:\architectoniki2\spim-cache\exampleCache.s successfully loaded**

A "Set Value" dialog box is open in the foreground. It has a title bar "Set Value" and a close button. It contains two input fields: "Address or Register Name:" with the value "pc" and "Value:" with the value "0x400000". There are "OK" and "Cancel" buttons.

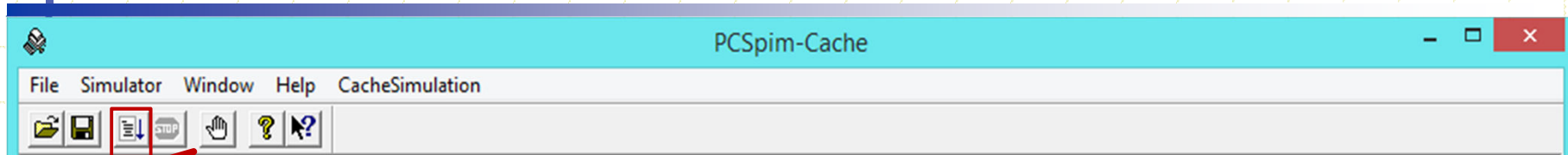
οι εντολές έχουν μεταφραστεί & φορτωθεί στη μνήμη εντολών, ξεκινώντας πάντα από την θέση "0x00400000"

οι τιμές των δεδομένων έχουν φορτωθεί στη μνήμη δεδομένων.

ο PC έχει την τιμή "0" ενώ πρέπει να πάρει την τιμή "0x400000" για μπόρέσει το πρόγραμμα να εκτελεστεί σωστά.



Πρόγραμμα exampleCache.s (4)

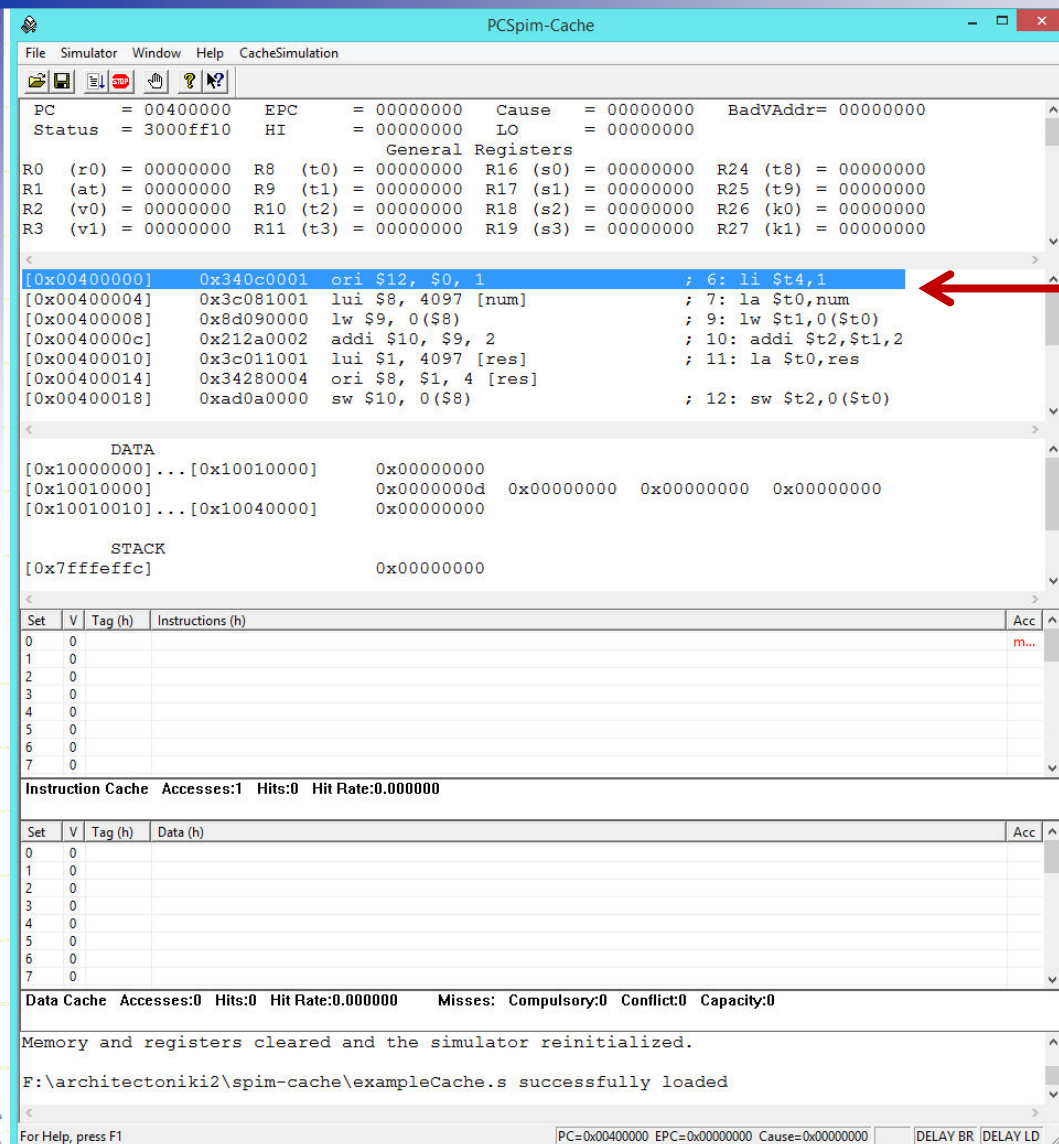


Πατώντας ***Simulator/Execute*** έχουμε τις παρακάτω τρεις επιλογές:

1. Άμεση εκτέλεση & ολοκλήρωση του προγράμματος, πατώντας **Go** ή **F5**, ή πατώντας το αντίστοιχο εικονίδιο στην μπάρα εργαλείων.
2. Εκτέλεση βήμα-βήμα, προχωρώντας την προσομοίωση κατά έναν κύκλο ρολογιού, πατώντας **Single Step** ή **F10**.
3. Εκτέλεση πολλαπλών βημάτων, προχωρώντας την προσομοίωση κατά έναν αριθμό κύκλων ρολογιού, πατώντας **Multi Step** ή **F11**.



Πρόγραμμα exampleCache.s (5)

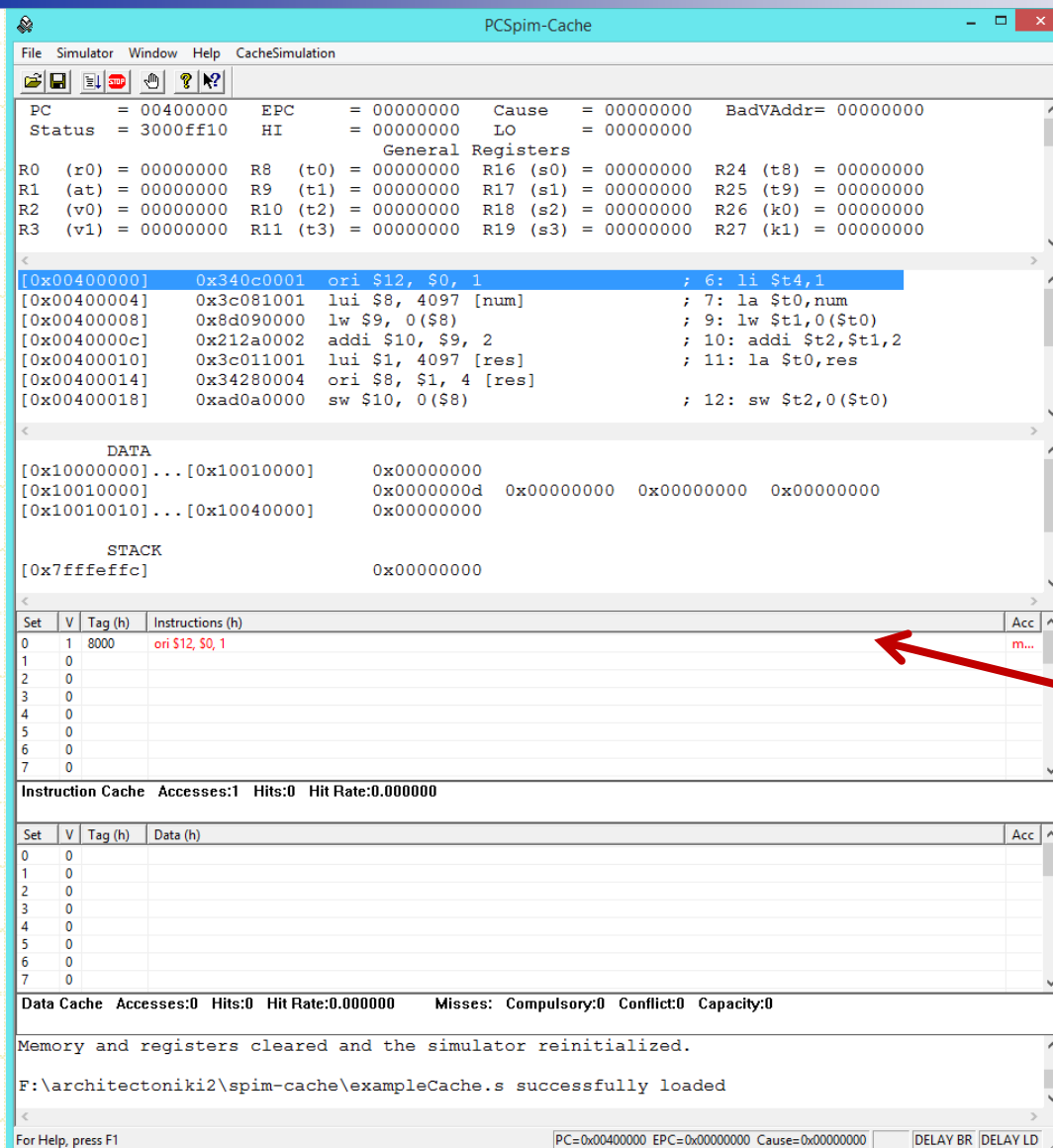


```
File Simulator Window Help CacheSimulation
[Icons]
PC = 00400000 EPC = 00000000 Cause = 00000000 BadVAddr= 00000000
Status = 3000ff10 HI = 00000000 Lo = 00000000
General Registers
R0 (r0) = 00000000 R8 (t0) = 00000000 R16 (s0) = 00000000 R24 (t8) = 00000000
R1 (at) = 00000000 R9 (t1) = 00000000 R17 (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
[0x00400000] 0x340c0001 ori $t2, $0, 1 ; 6: li $t4,1
[0x00400004] 0x3c081001 lui $8, 4097 [num] ; 7: la $t0,num
[0x00400008] 0x8d090000 lw $9, 0($8) ; 9: lw $t1,0($t0)
[0x0040000c] 0x212a0002 addi $t0, $9, 2 ; 10: addi $t2,$t1,2
[0x00400010] 0x3c011001 lui $1, 4097 [res] ; 11: la $t0,res
[0x00400014] 0x34280004 ori $8, $1, 4 [res] ; 12: sw $t2,0($t0)
[0x00400018] 0xad0a0000 sw $t0, 0($8)
DATA
[0x10000000]...[0x10010000] 0x00000000
[0x10010000] 0x0000000d 0x00000000 0x00000000 0x00000000
[0x10010010]...[0x10040000] 0x00000000
STACK
[0x7ffffc] 0x00000000
Set V Tag (h) Instructions (h) Acc
0 0
1 0
2 0
3 0
4 0
5 0
6 0
7 0
Instruction Cache Accesses:1 Hits:0 Hit Rate:0.000000
Set V Tag (h) Data (h) Acc
0 0
1 0
2 0
3 0
4 0
5 0
6 0
7 0
Data Cache Accesses:0 Hits:0 Hit Rate:0.000000 Misses: Compulsory:0 Conflict:0 Capacity:0
Memory and registers cleared and the simulator reinitialized.
F:\architectoniki2\spim-cache\exampleCache.s successfully loaded
For Help, press F1 PC=0x00400000 EPC=0x00000000 Cause=0x00000000 DELAY BR DELAY LD
```

η 1^η εντολή είναι
έτοιμη να εκτελεστεί



Πρόγραμμα exampleCache.s (6)



The screenshot shows the PCSpim-Cache simulator interface. At the top, the title bar reads "PCSpim-Cache". Below it is a menu bar with "File", "Simulator", "Window", "Help", and "CacheSimulation". A toolbar with various icons is located below the menu bar. The main display area is divided into several sections:

- Registers:** A table showing the state of various registers. PC is 00400000, EPC is 00000000, Cause is 00000000, and BadVAddr is 00000000. Status is 3000fff10, HI is 00000000, LO is 00000000. General Registers R0 through R31 are listed with their values.
- Instructions:** A list of instructions with their addresses and assembly code. The first instruction at address 0x00400000 is highlighted in blue: `[0x00400000] 0x340c0001 ori $t2, $0, 1 ; 6: li $t2, 1`.
- DATA:** A section showing memory data at addresses 0x10000000 through 0x1000000f.
- STACK:** A section showing stack memory at address 0x7ffffeffc.
- Instruction Cache:** A table with columns Set, V, Tag (h), Instructions (h), and Acc. The first row shows Set 0, V 1, Tag 8000, and Instructions `ori $t2, $0, 1`. A red arrow points to this row.
- Data Cache:** A table with columns Set, V, Tag (h), Data (h), and Acc. It is currently empty.
- Status:** A section showing "Accesses:1 Hits:0 Hit Rate:0.000000", "Misses: Compulsory:0 Conflict:0 Capacity:0", and "Memory and registers cleared and the simulator reinitialized."
- Footer:** A status bar at the bottom shows "F:\architectoniki2\spim-cache\exampleCache.s successfully loaded" and "For Help, press F1".

η 1^η εντολή έχει φορτωθεί στην κρυφή μνήμη εντολών, & επειδή είναι η πρώτη προσπάθεια γίνεται “miss”.



Πρόγραμμα exampleCache.s (7)

```
PCSpim-Cache
File Simulator Window Help CacheSimulation
[Icons]
PC = 0040000c EPC = 00000000 Cause = 00000000 BadVAddr= 00000000
Status = 3000ff10 HI = 00000000 LO = 00000000
General Registers
R0 (r0) = 00000000 R8 (t0) = 10010000 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
[0x00400000] 0x340c0001 ori $12, $0, 1 ; 6: li $t4,1
[0x00400004] 0x3c081001 lui $8, 4097 [num] ; 7: la $t0,num
[0x00400008] 0x8d090000 lw $9, 0($8) ; 9: lw $t1,0($t0)
[0x0040000c] 0x212a0002 addi $10, $9, 2 ; 10: addi $t2,$t1,2
[0x00400010] 0x3c011001 lui $1, 4097 [res] ; 11: la $t0,res
[0x00400014] 0x34280004 ori $8, $1, 4 [res]
[0x00400018] 0xad0a0000 sw $10, 0($8) ; 12: sw $t2,0($t0)
DATA
[0x10000000]...[0x10010000] 0x00000000
[0x10010000] 0x0000000d 0x00000000 0x00000000 0x00000000
[0x10010010]...[0x10040000] 0x00000000
STACK
[0x7fffffc] 0x00000000
Set V Tag (h) Instructions (h) Acc
0 1 8000 ori $12, $0, 1
1 1 8000 lui $8, 4097
2 1 8000 lw $9, 0($8)
3 0
4 0
5 0
6 0
7 0
Instruction Cache Accesses:3 Hits:0 Hit Rate:0.000000
Set V Tag (h) Data (h) Acc
0 1 200200 0000000d m...
1 0
2 0
3 0
4 0
5 0
6 0
7 0
Data Cache Accesses:1 Hits:0 Hit Rate:0.000000 Misses: Compulsory:1 Conflict:0 Capacity:0
[0x00400008] 0x8d090000 lw $9, 0($8) ; 9: lw $t1,0($t0)
[0x00400008] 0x8d090000 lw $9, 0($8) ; 9: lw $t1,0($t0)
[0x00400008] 0x8d090000 lw $9, 0($8) ; 9: lw $t1,0($t0)
For Help, press F1 PC=0x0040000c EPC=0x00000000 Cause=0x00000000 DELAY BR DELAY LD
```

εδώ έχουμε ένα
“miss” στην κρυφή
μνήμη δεδομένων.



Πρόγραμμα exampleCache.s (8)

```
PCSpim-Cache
File Simulator Window Help CacheSimulation
[Icons]
PC = 00400010 EPC = 00000000 Cause = 00000000 BadVAddr= 00000000
Status = 3000ff10 HI = 00000000 LO = 00000000
General Registers
R0 (r0) = 00000000 R8 (t0) = 10010004 R16 (s0) = 00000000 R24 (t8) = 00000000
R1 (at) = 10010000 R9 (t1) = 0000000d R17 (s1) = 00000000 R25 (t9) = 00000000
R2 (v0) = 00000000 R10 (t2) = 0000000f R18 (s2) = 00000000 R26 (k0) = 00000000
R3 (v1) = 00000000 R11 (t3) = 00000001 R19 (s3) = 00000000 R27 (k1) = 00000000

[0x00400008] 0x8d090000 lw $9, 0($8) ; 9: lw $t1,0($t0)
[0x0040000c] 0x212a0002 addi $10, $9, 2 ; 10: addi $t2,$t1,2
[0x00400010] 0x3c011001 lui $1, 4097 [res] ; 11: la $t0,res
[0x00400014] 0x34280004 ori $8, $1, 4 [res]
[0x00400018] 0xad0a0000 sw $10, 0($8) ; 12: sw $t2,0($t0)
[0x0040001c] 0x216b0001 addi $11, $11, 1 ; 13: addi $t3,$t3,1
[0x00400020] 0x116cfff9 beq $11, $12, -28 [label1-0x00400020]; 14: beq $t3,$t4,label1

DATA
[0x10000000]...[0x10010000] 0x00000000
[0x10010000] 0x0000000d 0x00000002 0x00000000 0x00000000
[0x10010010]...[0x10040000] 0x00000000

STACK
[0x7ffffefc] 0x00000000

Set V Tag (h) Instructions (h) Acc
2 1 8000 lw $9, 0($8)
3 1 8000 addi $10, $9, 2
4 1 8000 lui $1, 4097
5 1 8000 ori $8, $1, 4
6 1 8000 sw $10, 0($8)
7 1 8000 addi $11, $11, 1
8 1 8000 beq $11, $12, -28
9 0

Instruction Cache Accesses:11 Hits:1 Hit Rate:0.090909
Set V Tag (h) Data (h) Acc
0 1 20020 0000000d
1 1 20020 00000002 hit
2 0
3 0
4 0
5 0
6 0
7 0

Data Cache Accesses:3 Hits:1 Hit Rate:0.333333 Misses: Compulsory:2 Conflict:0 Capacity:0
[0x00400020] 0x116cfff9 beq $11, $12, -28 [label1-0x00400020]; 14: beq $t3,$t4,label1
[0x00400008] 0x8d090000 lw $9, 0($8) ; 9: lw $t1,0($t0)
[0x0040000c] 0x212a0002 addi $10, $9, 2 ; 10: addi $t2,$t1,2

For Help, press F1 PC=0x00400010 EPC=0x00000000 Cause=0x00000000 DELAY BR DELAY LD
```

εδώ έχουμε “hit”
τόσο στην κρυφή
μνήμη δεδομένων,
όσο & στην κρυφή
μνήμη εντολών.



Πρόγραμμα exampleCache.s (9)

```
PCSpim-Cache
File Simulator Window Help CacheSimulation

PC = 00000000 EPC = 00000000 Cause = 00000018 BadVAddr= 00000000
Status = 3000ff12 HI = 00000000 LO = 00000000

General Registers
R0 (r0) = 00000000 R8 (t0) = 10010004 R16 (s0) = 00000000 R24 (t8) = 00000000
R1 (at) = 10010000 R9 (t1) = 00000002 R17 (s1) = 00000000 R25 (t9) = 00000000
R2 (v0) = 0000000a R10 (t2) = 0000000f R18 (s2) = 00000000 R26 (k0) = 00000000
R3 (v1) = 00000000 R11 (t3) = 00000002 R19 (s3) = 00000000 R27 (k1) = 00000000

[0x00400010] 0x3c011001 lui $1, 4097 [res] ; 11: la $t0,res
[0x00400014] 0x34280004 ori $8, $1, 4 [res] ; 12: sw $t2,0($t0)
[0x00400018] 0xad0a0000 sw $10, 0($8) ; 13: addi $t3,$t3,1
[0x0040001c] 0x216b0001 addi $11, $11, 1 ; 14: beq $t3,$t4,label1
[0x00400020] 0x116cfff9 beq $11, $12, -28 [label1-0x00400020]; 15: li $v0,10 #au revoir...
[0x00400024] 0x3402000a ori $2, $0, 10 ; 16: syscall
[0x00400028] 0x0000000c syscall ; 16: syscall

DATA
[0x10000000]...[0x10010000] 0x00000000
[0x10010000] 0x0000000d 0x0000000f 0x00000000 0x00000000
[0x10010010]...[0x10040000] 0x00000000

STACK
[0x7ffffeffc] 0x00000000

Set V Tag (h) Instructions (h) Acc
8 1 8000 beq $11, $12, -28
9 1 8000 ori $2, $0, 10
10 1 8000 8000 m...
11 0
12 0
13 0
14 0
15 0

Instruction Cache Accesses:18 Hits:6 Hit Rate:0.333333

Set V Tag (h) Data (h) Acc
0 1 200200 0000000d
1 1 200200 0000000f hit
2 0
3 0
4 0
5 0
6 0
7 0

Data Cache Accesses:4 Hits:2 Hit Rate:0.500000 Misses: Compulsory:2 Conflict:0 Capacity:0

[0x00400020] 0x116cfff9 beq $11, $12, -28 [label1-0x00400020]; 14: beq $t3,$t4,label1
[0x00400024] 0x3402000a ori $2, $0, 10 ; 15: li $v0,10 #au revoir...
[0x00400028] 0x0000000c syscall ; 16: syscall

For Help, press F1 PC=0x00000000 EPC=0x00000000 Cause=0x00000018 DELAY BR DELAY LD
```

Ολοκλήρωση του προγράμματος.

Στο κάτω μέρος των παραθύρων βλέπουμε τα στατιστικά για τις κρυφές μνήμες & δεδομένων αντίστοιχα.



Πρόγραμμα exampleCache.s (10)

Μπορούμε τώρα να δοκιμάσουμε την εκτέλεση του προγράμματος με διάφορες διαμορφώσεις & ρυθμίσεις των κρυφών μνημών.



Περισσότερα...

- Στις σημειώσεις του μαθήματος
‘Αρχιτεκτονική Υπολογιστών II’
- Email επικοινωνίας:
 - xergias@di.uoa.gr

