

7ο Εργαστήριο Αρχιτεκτονικής Η/Υ:

Κρυφές μνήμες (cache)

Απαντήσεις

Α. Ευθυμίου

Στις εκτελέσεις του προγράμματος θα γίνονται αλλαγές στις εξής παραμέτρους¹ του.

arraySize - το μέγεθος του πίνακα array σε λέξεις. Αντιστοιχεί στον καταχωρητή s0.

option - Ελέγχει τί θα κάνει ο εσωτερικός βρόχος: 0 μηδενίζει τα στοιχεία του πίνακα και 1 τα αυξάνει κατά 1. Αντιστοιχεί στον καταχωρητή s1.

stepSize - το βήμα με το οποίο προσπελάζεται ο πίνακας array. Αντιστοιχεί στον καταχωρητή s2.

repCount - ο αριθμός επαναλήψεων του εσωτερικού βρόχου. Αντιστοιχεί στον καταχωρητή s3.

1 Μέρος Α: Αντιστοίχιση γραμμών σε θέσεις της cache

Οι παράμετροι του cache.s: arraySize=32, option=0, stepSize=1, repCount=1, padding=0.

Cache configuration: direct mapped (2^N Ways 0), 8 γραμμές (2^N Lines 3), μέγεθος γραμμής: 2 λέξεις (2^N Words/Line 1).

Εκτελέστε το πρόγραμμα μέχρι την πρώτη προσπέλαση μνήμης και συμπληρώστε τις πληροφορίες που ζητούνται στο αριστερό μισό (με επικεφαλίδα direct mapped) του πίνακα απαντήσεων 1 στην γραμμή για padding 0 και την πρώτη προσπέλαση: Αν η κρυφή μνήμη ευστόχησε (Hit ή Miss), τον αριθμό του block που προσπελάστηκε (0 το πρώτο μέχρι 7 το τελευταίο) και το tag της γραμμής (φαίνεται σε ξεχωριστή στήλη σε κάθε cache block). Για το tag χρησιμοποιείτε μόνο δεκαεξαδικούς αριθμούς.

Συνεχίστε την εκτέλεση μέχρι και την δεύτερη προσπέλαση μνήμης και συμπληρώστε ξανά τις αντίστοιχες πληροφορίες στον πίνακα.

Με padding 0 η πρώτη προσπέλαση γίνεται στη διεύθυνση 0x10000000. Με άδεια κρυφή μνήμη, προκαλείται αστοχία και προσκομίζεται ολόκληρη η γραμμή: και το word 0 και το word 1 στη γραμμή με index 0. Το tag είναι 0x400000 που προκύπτει από την αρχική διεύθυνση αλλά με "κομμένα" τα 6 λιγότερο σημαντικά bit. Το 6 προκύπτει από τα 3 bits για index (το 2^N lines το έχουμε βάσει 3) και τα 3 bits block offset. Το block offset προκύπτει από το μέγεθος της γραμμής, αλλά σε bytes: το 2^N words/line το έχουμε βάσει 1, άρα 2 λέξεις των 4 bytes η κάθε μία, 8 bytes, και επειδή $8 = 2^3$, 3 bits για block offset.

Η δεύτερη προσπέλαση γίνεται στην διεύθυνση 0x10000004, που αντιστοιχεί στη word 1 της ίδιας γραμμής. Αν αναλύσετε τη διεύθυνση σε πεδία (το κάνει ο Ripes για σας στο cache tab στο επάνω μέρος της εικονας όταν εκτελείται η εντολή) θα δείτε ότι η διεύθυνση είναι ίδια με την προηγούμενη εκτος από 1 bit στο πεδίο block offset. Επομένως έχουμε ευστοχία.

Αλλάξτε το padding σε 4. Δείτε και καταγράψτε στις αντίστοιχες γραμμές του πίνακα ότι ζητείται για τις δύο πρώτες προσπελάσεις του προγράμματος.

¹Προσοχή να μην συγχέονται με τις παραμέτρους της κρυφής μνήμης!

Τώρα η πρώτη προσπέλαση γίνεται στη διεύθυνση 0x10000004. Πάλι συμβαίνει αστοχία γιατί η κρυφή μνήμη είναι άδεια. Αλλά ξεκινάμε τώρα από την δεύτερη λέξη (word 1) της γραμμής. Η δεύτερη προσπέλαση γίνεται στη διεύθυνση 0x10000008, που αντιστοιχεί σε άλλη γραμμή: αν αναλύσετε τη διεύθυνση σε πεδία, θα δείτε ότι το index είναι τώρα 1, συνεπώς η γραμμή αυτή αντιστοιχεί στην δεύτερη θέση της κρυφής μνήμης. Συνεπώς έχουμε πάλι αστοχία και αυτό είναι διαφορετικό από όταν το padding ήταν 0. Στη συνέχεια της εκτέλεσης έχουμε παρόμοια συμπεριφορά με την περίπτωση όπου το padding ήταν 0.

Επαναλάβετε με padding 8 αυτή τη φορά και συμπληρώστε τον πίνακα.

Τώρα η πρώτη προσπέλαση γίνεται στη διεύθυνση 0x10000008 που αντιστοιχεί στην Word 0 της δεύτερης γραμμής της κρυφής μνήμης (index = 1). Η συνέχεια είναι όπως όταν το padding ήταν 0, απλά ξεκινάμε μια θέση κάτω στην κρυφή μνήμη.

Padding	Mem access	Direct Mapped			Fully Associative		
		Hit?	Αριθμός block	Tag	Hit?	Αριθμός block	Tag
0	1st	Miss	0	400000	Miss	0	2000000
	2nd	Hit	0	400000	Hit	0	2000000
4	1st	Miss	0	400000	Miss	0	2000000
	2nd	Miss	1	400000	Miss	1	2000001
8	1st	Miss	1	400000	Miss	0	2000001
	2nd	Hit	1	400000	Hit	0	2000001

Πίνακας 1: Πίνακας απαντήσεων πρώτου μέρους

Αλλάξτε την οργάνωση σε Fully Associative, με αλγόριθμο αντικατάστασης (Replacement Policy) LRU, και τις υπόλοιπες παραμέτρους ίδιες (συνολικό αριθμό γραμμών 8, μέγεθος γραμμής 2 λέξεις). Προσοχή δεν υπάρχει απευθείας τέτοια επιλογή στο cache configuration του Ripes. Πρέπει να σκεφτείτε πως θα θέσετε τις τιμές στα (2^N Lines, 2^N Ways) ώστε να το πετύχετε. Επαναλάβετε τις τρεις παραπάνω παραλλαγές του padding (0, 4, 8). Αυτή τη φορά συμπληρώνετε το δεξί μισό του πίνακα απαντήσεων (με επικεφαλίδα Fully Associative).

Cache configuration: fully associative (2^N Lines 0), 8 γραμμές (2^N Ways 3), μέγεθος γραμμής: 2 λέξεις (2^N Words/Line 1). Προσοχή αφού είναι fully associative έχουμε ένα μόνο σετ, συνεπώς, (2Lines 0). Για να έχουμε τον ίδιο συνολικό αριθμό γραμμών, αυξάνουμε τα ways: 2^N Ways 3.

Παρατηρήστε ότι τώρα το tag είναι διαφορετικό, σε σχέση με την οργάνωση direct mapped και το ίδιο padding, παρόλο που η διεύθυνση είναι η ίδια. Δεν υπάρχει πεδίο index στη διεύθυνση, έτσι το tag έχει 3 bits περισσότερα από ότι στην direct mapped οργάνωση. Αυτό θα το δείτε και όταν το padding αλλάξει. Κατά τα άλλα η κρυφή μνήμη συμπεριφέρεται παρόμοια.

Με padding 4, πάλι, αρχικά, γίνονται προσπελάσεις σε δύο διαφορετικές γραμμές κρυφής μνήμης, με δύο συνεχόμενες αστοχίες. Μετά η συμπεριφορά είναι παρόμοια, αλλά προσέξτε ότι τώρα κάθε γραμμή έχει διαφορετικό tag σε αντίθεση με την direct-mapped οργάνωση όπου κάποιες γραμμές (με διαφορετικό index) είχαν το ίδιο tag.

Με padding 8, προσέξτε ότι στην πρώτη προσπέλαση πάλι η πρώτη γραμμή της κρυφής μνήμης χρησιμοποιήθηκε. Στην fully-associative μνήμη υπάρχει ένα μόνο σετ (με index 0). Μέσα στο σετ η πρώτη ελεύθερη θέση συμπληρώνεται (ή η λιγότερο πρόσφατα χρησιμοποιημένη, αν όλα τα ways του σετ είναι κατελειμένα).

2 Μέρος B: Επίδραση μεγέθους γραμμής

Βάλτε τις παρακάτω τιμές παραμέτρων στο cache.s: arraySize 128, option 1, stepSize 1, repCount 1, και padding 0. Ρυθμίστε την κρυφή μνήμη ως εξής: οργάνωση Direct Mapped,

αριθμός γραμμών 8, μέγεθος γραμμής 2 λέξεις. Τρέξετε το πρόγραμμα και καταγράψτε το μοτίβο ευστοχιών αστοχιών ως μια ακολουθία από γράμματα M - miss, H - hit. Η απάντησή σας θα πρέπει να είναι το συντομότερο μοτίβο που επαναλαμβάνεται, για παράδειγμα το "HHMHHM" θα πρέπει να γραφεί ως "HHM". Καταγράψτε και το τελικό ποσοστό ευστοχίας του προγράμματος.

Σκεφτείτε πώς το μοτίβο μπορεί άμεσα να σας δώσει το τελικό ποσοστό ευστοχίας.

Μοτίβο	Ποσοστό ευστοχίας
MHHH	75%

Πίνακας 2: Direct mapped, 8 γραμμές, 2 λέξεις ανά γραμμή

Το μοτίβο ωφείλεται στο ότι η πρώτη προσπέλαση, ανάγνωση, σε μια (νέα) γραμμή κρυφής μνήμης προκαλεί αστοχία. Η επόμενη προσπέλαση είναι εγγραφή στην ίδια διεύθυνση και είναι εύστοχη. Ακολουθούν δύο προσπελάσεις, ανάγνωση - εγγραφή, στην δεύτερη λέξη της ίδιας γραμμής. Αφού η γραμμή είναι ήδη στην κρυφή μνήμη και οι δύο ευστοχούν.

Αλλάξτε το μέγεθος γραμμής σε 4 λέξεις, αλλά για να διατηρηθεί η συνολική χωρητικότητα της cache ίδια, μειώστε τον αριθμό γραμμών σε 4. Βρείτε και γράψτε το νέο μοτίβο και το ποσοστό ευστοχίας.

Μοτίβο	Ποσοστό ευστοχίας
MHHHHHHH	87.5%

Πίνακας 3: Direct mapped, 4 γραμμές, 4 λέξεις ανά γραμμή

Cache configuration: direct mapped (2^N Ways 0), μέγεθος γραμμής: 4 λέξεις (2^N Words/Line 2). Για να υπάρχουν συνολικά 8 γραμμές, πρέπει ο αριθμός των σετ να γίνει ο μισός 2^N Lines 2.

Τώρα η γραμμή μεγάλωσε και έχουμε μία αστοχία ακολουθούμενη από 7 εύστοχες προσπελάσεις.

Σκεφτείτε σε ποιό από τα δύο είδη τοπικότητας αναφορών μνήμης οφείλεται κάθε ευστοχία του μοτίβου. Αντιγράψτε το μοτίβο που βρήκατε στο τελευταίο ερώτημα, αλλάζοντας κάθε H (hit) είτε σε T (temporal), για χρονική τοπικότητα αναφοράς, είτε σε S (spatial), για χωρική τοπικότητα. Αφήστε τα M ως είναι.

Μοτίβο
MTSTSTST

Πίνακας 4: Μοτίβο με είδη τοπικότητας αναφορών

Όταν γίνεται η εγγραφή έχει ήδη προηγηθεί προσπέλαση στην ίδια διεύθυνση για ανάγνωση. Επομένως όλες οι εγγραφές ευστοχούν και αυτό οφείλεται στην χρονική τοπικότητα. Όλες εκτός της πρώτης ανάγνωσης σε κάθε γραμμή ευστοχούν γιατί έχουν τοποθετηθεί εκ των προτέρων στην κρυφή μνήμη επειδή βρίσκονται κοντά στην διεύθυνση της πρώτης ανάγνωσης που αστόχησε. Επομένως οι ευστοχίες αυτές εκμεταλλεύονται την χωρική τοπικότητα.

Το παραπάνω ποσοστό ευστοχίας δεν θα άλλαζε αν είχαμε μεγαλύτερο repCount ακόμα και αν η οργάνωση γινόταν fully associative. Ο λόγος είναι ότι η κρυφή μνήμη είναι πολύ μικρή για το μέγεθος του πίνακα. Οι γραμμές που έχουν προσκομιστεί παλιότερα αντικαθίστανται από άλλες γραμμές. Έτσι όταν αρχίσει μια επόμενη επανάληψη του εξωτερικού βρόχου, δεν θα υπάρχουν άλλο στην κρυφή μνήμη

οι πρώτες γραμμές του πίνακα. Συνεπώς θα προκαλούνται ξανά αστοχίες στην πρώτη προσπέλαση κάθε γραμμής. Ενώ οι αστοχίες στην πρώτη επανάληψη του εξωτερικού βρόχου, όταν το `repCount` είναι 0, είναι υποχρεωτικές, καθώς είναι η πρώτη φορά που το πρόγραμμα ζητάει τις γραμμές, οι αστοχίες στις επόμενες επαναλήψεις είναι αστοχίες χωρητικότητας. Μόνο μια μεγαλύτερη κρυφή μνήμη θα μπορούσε να τις αντιμετωπίσει.

3 Μέρος Γ: Επίδραση `repCount`

Βάλτε τις παρακάτω τιμές παραμέτρων στο `cache.s`: `arraySize 32`, `option 1`, `stepSize 4`, `repCount 1`, και `padding 0`. Ρυθμίστε τον `cache simulator` ως εξής: οργάνωση `Direct Mapped`, αριθμός γραμμών 16, μέγεθος γραμμής 2 λέξεις. Καταγράψτε τον αριθμό προσπελάσεων που συμβαίνουν σε μία επανάληψη του **εξωτερικού βρόχου** και το τελικό ποσοστό ευστοχίας (στο τέλος της εκτέλεσης όλου του προγράμματος).

Αριθμ. προσπελάσεων μνήμης	Ποσοστό ευστοχίας
8	50%

Πίνακας 5: `repCount 1`

Καθώς το `repCount` είναι 1, μόνο μία επανάληψη συμβαίνει στον εξωτερικό βρόχο.

Το ενδιαφέρον εδώ είναι ότι λόγω του `stepSize 4` κάθε γραμμή προσπελαύνεται μόνο 2 φορές: μια ανάγνωση που αστοχεί και μια εγγραφή (στην ίδια λέξη) που ευστοχεί. Επιπλέον πάλι λόγω της τιμής του `stepSize`, οι μισές γραμμές της κρυφής μνήμης δεν χρησιμοποιούνται.

Αλλάξτε το `repCount` σε 2 και επαναλάβετε το πείραμα. Καταγράψτε το νέο (τελικό) ποσοστό ευστοχίας.

Ποσοστό ευστοχίας
75%

Πίνακας 6: `repCount 2`

Σκεφτείτε τί θα συμβεί αν συνεχίσετε να αυξάνετε το `repCount`. Συμπληρώστε το μικρότερο δυνατό μοτίβο που επαναλαμβάνεται, καταγράφοντας και το είδος τοπικότητας αναφορών μνήμης σε κάθε ευστοχία, **από την δεύτερη επανάληψη και έπειτα**. (Όπως παραπάνω, γράψτε το μοτίβο, αλλά στις ευστοχίες, γράψτε S ή T, ενώ τυχόν αστοχίες συμβολίζονται με M.)

Μοτίβο
T

Πίνακας 7: Μοτίβο με είδη τοπικότητας αναφορών για τη δεύτερη επανάληψη και έπειτα

Μετά την πρώτη επανάληψη του εξωτερικού βρόχου, όλες οι λέξεις που προσπελαύνει το πρόγραμμα έχουν τοποθετηθεί στην κρυφή μνήμη. Επομένως όλες οι προσπελάσεις από την δεύτερη επανάληψη και έπειτα ευστοχούν. Το ποσοστό ευστοχίας θα συνεχίσει να αυξάνεται προσεγγίζοντας το 100% όσο αυξάνεται το `repCount`. Καθώς οι προσπελάσεις αφορούν λέξεις που έχουν προσπελαστεί στο παρελθόν το είδος τοπικότητας που εκμεταλευόμαστε είναι χρονική.

Συμπληρώστε τον μαθηματικό τύπο που δίνει το ποσοστό αστοχιών σε σχέση με το `repCount`. (Στο αρχείο απαντήσεων) γράψτε το σαν να το γράφατε π.χ. σε Java. Μπορείτε να το επιβεβαιώσετε με μερικά πειράματα, αλλά προσέξτε ότι ο Ripes παρουσιάζει το hit rate (στο πλαίσιο `statistics`) ενώ εδώ ζητείται το miss rate.

Ποσοστό αστοχιών σε σχέση με το repCount
$1/(2*\text{repCount})$

Πίνακας 8: repCount 2

Όλες οι προσπελάσεις είναι $2*(\text{size}/\text{stepSize})*\text{repCount}$ και από αυτές αστοχούν μόνο οι μισές της πρώτης (εξωτερικής) επανάληψης: $\text{size}/\text{stepSize}$. Με διαίρεση προκύπτει ο τύπος του πίνακα.

4 Μέρος Δ: Επίδραση stepSize και associativity

Βάλτε τις παρακάτω τιμές παραμέτρων στο cache.s: arraySize 128, option 1, stepSize 16, repCount 4, και padding 0. Ρυθμίστε τον cache simulator ως εξής: οργάνωση Direct Mapped, αριθμός γραμμών 16, μέγεθος γραμμής 4 λέξεις. Καταγράψτε τον αριθμό προσπελάσεων της πρώτης επανάληψης του εξωτερικού βρόχου και το ποσοστό ευστοχίας. Επιπλέον, γράψτε το μοτίβο προσπελάσεων στην μνήμη της πρώτης επανάληψης του εξωτερικού βρόχου, αλλά κάθε σε αστοχία θα γράψετε το είδος της αστοχίας που συμβαίνει: Υ-υποχρεωτική, Σ - σύγκρουσης, Χ - Χωρητικότητα και Ε για ευστοχία. Χρησιμοποιούμε Ελληνικούς χαρακτήρες εδώ γιατί στα Αγγλικά όλα τα είδη αστοχιών ξεκινούν με C! **Προσοχή** ενώ προηγουμένως είχε ζητηθεί το συντομότερο δυνατό μοτίβο, εδώ ζητείται πλήρης καταγραφή - χαρακτηρισμός όλων των προσπελάσεων της πρώτης επανάληψης του εξωτερικού βρόχου.

Αριθμ. προσπελάσεων μνήμης	Ποσοστό ευστοχίας	Μοτίβο
16	50%	ΥΕΥΕΥΕΥΕΥΕΥΕΥΕΥΕ

Πίνακας 9: Πρώτη επανάληψη του εξωτερικού βρόχου

Όλες οι αναγνώσεις αστοχούν υποχρεωτικά καθώς, λόγω του μεγάλου stepSize, "πέφτουν" σε διαφορετικές γραμμές της κρυφής μνήμης. Η εγγραφή που ακολουθεί κάθε ανάγνωση ευστοχεί.

Συνεχίστε την προσομοίωση και ολοκληρώστε την **δεύτερη** επανάληψη του εξωτερικού βρόχου. Σημειώστε το συνολικό ποσοστό ευστοχίας από την αρχή της εκτέλεσης μέχρι αυτό το σημείο της εκτέλεσης. (Όχι μέχρι το τέλος του προγράμματος.) Γράψτε το μοτίβο ολόκληρης της **δεύτερης** επανάληψης, όπως και προηγουμένως.

Ποσοστό ευστοχίας	Μοτίβο
50%	ΣΕΣΕΣΕΣΕΣΕΣΕΣΕΣΕ

Πίνακας 10: Δεύτερη επανάληψη του εξωτερικού βρόχου

Το μοτίβο εξακολουθεί να είναι αστοχία για την ανάγνωση και ευστοχία για την εγγραφή. Όμως τώρα η αστοχία χρεώνεται ως σύγκρουση καθώς στην πρώτη εξωτερική επανάληψη όλες οι ζητούμενες λέξεις είχαν προσπελαστεί. Οι αστοχίες είναι σύγκρουσης γιατί αναγκάζομαστε να διώξουμε γραμμές (που ξαναχρειάζονται αργότερα) επειδή έρχονται άλλες με το ίδιο index.

Συνεχίστε την προσομοίωση μέχρι να τερματίσει το πρόγραμμα και παρατηρήστε το τελικό ποσοστό ευστοχίας. Αν αλλάζατε το repCount σε μεγαλύτερο αριθμό, θα βελτιωνόταν το τελικό ποσοστό ευστοχίας;

Το μοτίβο παραμένει το ίδιο ανεξάρτητα από το repCount. Το ποσοστό ευστοχίας παραμένει στο 50%.

Απάντηση:
Όχι

Αν είχαμε την παράμετρο option του προγράμματος στην τιμή 0, ποιό θα ήταν το τελικό ποσοστό ευστοχίας;

Ποσοστό ευστοχίας
0%

Πίνακας 11: Ποσοστό ευστοχίας με option 0

Γίνεται μόνο μία προσπέλαση ανά λέξη και κάθε λέξη είναι σε διαφορετική γραμμή (σε μία εξωτερική επανάληψη). Επομένως συμβαίνουν μόνο αστοχίες.

Επαναφέρετε τις παραμέτρους του cache.s στις αρχικές του μέρους Γ: arraySize 128, option 1, stepSize 16, repCount 4, και padding 0. Ρυθμίστε την cache ώστε να ακολουθεί την Fully Associative οργάνωση, με όλες τις υπόλοιπες παραμέτρους ίδιες (αριθμός γραμμών 16, μέγεθος γραμμής 4 λέξεις). Τρέξτε τον **πρώτο εξωτερικό** βρόχο και παρατηρήστε πώς τοποθετούνται οι γραμμές δεδομένων στην cache. Καταγράψτε το πλήθος των θέσεων (γραμμών) της cache που χρησιμοποιούνται, και το ποσοστό ευστοχίας. Το μοτίβο προσπελάσεων (ακολουθία ευστοχιών - αστοχιών και το είδος των αστοχιών) είναι ίδιο με την Direct Mapped οργάνωση για την πρώτη (εξωτερική) επανάληψη;

Cache configuration: fully associative (2^N Lines 0), 16 γραμμές (2^N Ways 4), μέγεθος γραμμής: 4 λέξεις (2^N Words/Line 2).

Αριθμός θέσεων	Ποσοστό ευστοχίας	Ίδιο μοτίβο;
8	50%	Ναι

Πίνακας 12: Fully associative, πρώτη επανάληψη εξωτερικού βρόχου

Με fully-associative οργάνωση οι γραμμές τοποθετούνται σε ελεύθερα ways και επειδή η κρυφή μνήμη έχει αρκετή χωρητικότητα δεν γεμίζει και δεν έχουμε αντικαταστάσεις γραμμών.

Συνεχίστε την προσομοίωση και ολοκληρώστε την **δεύτερη** επανάληψη του εξωτερικού βρόχου. Σημειώστε το συνολικό ποσοστό ευστοχίας μέχρι αυτό το σημείο της εκτέλεσης. Το μοτίβο προσπελάσεων (ακολουθία ευστοχιών-αστοχιών και το είδος των αστοχιών) είναι ίδιο με την δεύτερη επανάληψη της Direct Mapped οργάνωσης;

Συνολικό ποσοστό ευστοχίας	Ίδιο μοτίβο με DM;
75%	Όχι

Πίνακας 13: Fully associative, δεύτερη επανάληψη εξωτερικού βρόχου

Τώρα έχουμε συνέχεια ευστοχίες από την δεύτερη (εξωτερική) επανάληψη και έπειτα. Συνεπώς το μοτίβο είναι διαφορετικό από την αντίστοιχη περίπτωση με οργάνωση direct-mapped. Οι ευστοχίες εκμεταλλεύονται (μόνο) την χρονική τοπικότητα αναφορών.

Συνεχίστε την προσομοίωση μέχρι να τερματίσει το πρόγραμμα και παρατηρήστε το τελικό ποσοστό ευστοχίας. Αν αλλάζατε το repCount σε μεγαλύτερο αριθμό, θα βελτιωνόταν το ποσοστό ευστοχίας;

Απάντηση:
Ναι

Όλες οι προσπελάσεις μετά την πρώτη επανάληψη είναι εύστοχες, επομένως όσο μεγαλύτερο είναι το repCount τόσο μεγαλύτερο το τελικό ποσοστό ευστοχίας (θα τείνει προς το 100%).

Βρείτε την μικρότερη associativity (αριθμός ways στον cache simulator) που να δίνει το ίδιο ποσοστό ευστοχίας με την Fully Associative, χωρίς να αλλάξετε άλλες παραμέτρους ούτε στο πρόγραμμα ούτε στην κρυφή μνήμη.

Απάντηση:
8

Βλέπουμε ότι 8 από τα 16 ways χρησιμοποιούνται. Επομένως αν έχουμε associativity (αριθμός ways) ίση με 8 θα έχουμε το ίδιο ποσοστό ευστοχίας με την fully-associative.