ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΠΟΛΟΓΙΣΤΩΝ

(2020-2021)

3η Σειρά Ασκήσεων

Ονοματεπώνυμο:

- Χρήστος Τσούφης

Α.Μ.:

- 03117176

Στοιχεία Επικοινωνίας:

- el17176@mail.ntua.gr

Μέρος Α

Δίνεται επεξεργαστής με ένα επίπεδο κρυφής μνήμης (L_1) με μέσο χρόνο πρόσβασης στη μνήμη 3.5 κύκλους ρολογιού, όπου το 98% των προσβάσεων είναι επιτυχείς και εξυπηρετούνται σε 1 κύκλο.

$t_{μέσο_1} = t_{hit_{L1}} + rate_{miss_{L1}} * t_{miss_{L1}}$

Εδώ, $t_{μέσο_1}$= 3.5 κύκλοι , $t_{hit_{L1}}$= 1 κύκλος , $rate_{miss_{L1}} = \frac{100-98}{100}$

Επομένως, $\boldsymbol{t_{miss_{L1}}}$**= 125 κύκλοι** και είναι ο χρόνος που απαιτείται για την μεταφορά δεδομένων από και προς την μνήμη εάν δεν βρεθούν στην cache.

Ζητείται να προστεθεί ένα δεύτερο επίπεδο κρυφής μνήμης (L_2) ώστε η επιτάχυνση (speedup) του μέσου χρόνου πρόσβασης στη μνήμη να είναι τουλάχιστον ίση με 2.5. Αν το hit rate αυτού του 2ου επιπέδου είναι 88.8%, ποιο το μέγιστο κόστος πρόσβασης (κύκλοι) σε αυτό;

$t_{μέσο_2} = t_{hit_{L1}} + rate_{miss_{L1}} * t_{miss_{L1}} = 1 + \frac{100-98}{100} * t_{miss_{L1}}$

$t_{miss_{L1}} = t_{hit_{L2}} + rate_{miss_{L2}} * t_{miss_{L2}} = t_{hit_{L2}} + \frac{100-88.8}{100} * 125$

Speedup = $\frac{t_{μέσο_1}}{t_{μέσο_2}} \geq 2.5$

Δηλαδή, $t_{μέσο_2} \leq 1.4$ cc, που σημαίνει ότι στην χειρότερη περίπτωση $t_{μέσο_2} = 1.4$ cc.

Ακόμη, θα είναι απαραίτητη η προσπέλαση στην μνήμη αν το στοιχείο δεν βρεθεί στην cache L2. Οπότε, προκύπτει ότι $\boldsymbol{t_{hit_{L2}}}$ **= 6 κύκλοι** , στην χειρότερη περίπτωση.

Μέρος Β

Δίνεται ο παρακάτω κώδικας **C**:

```
int i,j;
double A[16][8], B[16][8];
for (i = 0; i < 8; i++) {
    for (j = 0; j < 8; j++) {
        if (i < 4)
            A[i+4][j] = A[i+4][j] + A[i][j] + B[i][j];
        else
            A[i][j] = A[i][j] + A[i+1][j] + B[i][j];
    }
}
```

- Έστω ότι το στοιχείο A[0][0] βρίσκεται στο σετ 0.
  Τότε, το στοιχείο A[0][4] είναι 4στοιχεία*8bytes/στοιχείο = $(32)_d$ = 00100000. Άρα βρίσκεται στο σετ 1.
  Το A[1][0] είναι 8στοιχεία*8bytes/στοιχείο = $(64)_d$ = 01000000. Άρα βρίσκεται στο σετ 2 που φαίνεται από το index του 010.
  Συνολικά υπάρχουν 8 στο πλήθος sets (0 – 7), οπότε το A[2][0] θα απεικονίζεται δύο sets μετά το A[1][0] δηλαδή στο set 4 και αυτό.
  Ομοίως, το A[3][0] στο set 6 κ.ο.κ..
- Το B[0][0] βρίσκεται 16*8 στοιχεία*8bytes/στοιχείο = 1.024 = 10000000000 άρα στο σετ 0, όπου 16 είναι οι γραμμές του πίνακα A.
- Για την cache memory, θα ισχύει ότι:
  - Η μνήμη περιέχει #blocks = $\frac{512}{32}$ = 16 blocks.
  - Χρησιμοποιείται 2-way set associative cache με πολιτική αντικατάστασης LRU και ίδια κατά τα άλλα μεγέθη. Έτσι, set size = $\frac{blocks}{2} = \frac{16}{2}$ = 8 set.
  - Σε κάθε block της cache θα απεικονίζονται #στοιχείων ανά block = $\frac{32}{8}$ = 4 του πίνακα (πίνακας αποθηκευμένος κατά γραμμές).
  - Block offset = $\log_2 32$ = 5 bits
  - Index = $\log_2 8$ = 3 bits

Παρατήρηση:

Όταν i = 0, 4, 8, 12, 16: A[i][0] – A[i][3] set 0 & A[i][4] – A[i][7] set 1

Όταν i = 1, 5, 9, 13, 17: A[i][0] – A[i][3] set 2 & A[i][4] – A[i][7] set 3

Όταν i = 2, 6, 10, 14, 18: A[i][0] – A[i][3] set 4 & A[i][4] – A[i][7] set 5

Όταν i = 3, 7, 11, 15, 19: A[i][0] – A[i][3] set 6 & A[i][4] – A[i][7] set 7

Όμοια και για τον πίνακα B.

A) Τότε

i = 0 **j = 0…7**	**j = 0**	**j = 1**	**j = 2**	**j = 3**	**j = 4**	**j = 5**	**j = 6**	**j = 7**
read A[i+4][j]	M	H	H	H	M	H	H	H
read A[i][j]	M	M	M	M	M	M	M	M
read B[i][j];	M	M	M	M	M	M	M	M
write A[i+4][j]	M	M	M	M	M	M	M	M
i = 4 **j = 0...7**	**j = 0**	**j = 1**	**j = 2**	**j = 3**	**j = 4**	**j = 5**	**j = 6**	**j = 7**
read A[i][j]	H	H	H	H	H	H	H	H
read A[i+1][j]	H	H	H	H	H	H	H	H
read B[i][j];	M	H	H	H	M	H	H	H
write A[i][j]	H	H	H	H	H	H	H	H

Όταν i = 0, … , 3: 26 misses & 6 hits
Όταν i = 4, 5, 6: 2 misses & 30 hits
Όταν i = 7: 4 misses & 28 hits. Θα είναι 2 περισσότερα misses λόγω του A[8][0] – A[8][3] και A[8][4] – A[8][7].
Συνολικά, θα είναι 4*26 + 3*2 + 4 = 114 misses & 4*6 + 3*30 + 28 = 142 hits.

B) Εδώ, χρησιμοποιείται 4-way set associative cache με πολιτική αντικατάστασης LRU.

Set size = $\frac{blocks}{2} = \frac{16}{4}$ = 4 sets.

Παρατήρηση:

Όταν i = 0, 2, 4, 6, 8, 10, 12, 14, 16: A[i][0] – A[i][3] set 0 & A[i][4] – A[i][7] set 1

Όταν i = 1, 3, 5, 7, 9, 11, 13, 15: A[i][0] – A[i][3] set 2 & A[i][4] – A[i][7] set 3

i = 0 j = 0...7	**j = 0**	**j = 1**	**j = 2**	**j = 3**	**j = 4**	**j = 5**	**j = 6**	**j = 7**
read A[i+4][j]	M	H	H	H	M	H	H	H
read A[i][j]	M	H	H	H	M	H	H	H
read B[i][j];	M	H	H	H	M	H	H	H
write A[i+4][j]	H	H	H	H	H	H	H	H
i = 4 j = 0...7	**j = 0**	**j = 1**	**j = 2**	**j = 3**	**j = 4**	**j = 5**	**j = 6**	**j = 7**
read A[i][j]	H	H	H	H	H	H	H	H
read A[i+1][j]	H	H	H	H	H	H	H	H
read B[i][j];	M	H	H	H	M	H	H	H
write A[i][j]	M	H	H	H	M	H	H	H

Όταν i = 0, 1, 2, 3, 6: 26 hits & 6 misses
Όταν i = 4, 5: 28 hits & 4 misses
Όταν i = 7: 28 hits & 4 misses
Συνολικά, θα είναι 5*26 + 2*28 + 28 = 214 hits & 5*6 + 2*4 + 4 = 42 misses.
Συνεπώς, θα αντικατασταθεί η 2-way με μια 4-way set associative.