

***Θοδωρής Αράπης***

***El18028***

**Άσκηση 3.1**

**1.**

|  |  |  |
| --- | --- | --- |
| ***TAG*** | ***INDEX*** | ***OFFSET*** |

Έχουμε block μεγέθους 32 bytes, άρα το offset θα αποτελείται από:

Η cache έχει χωρητικότητα 128 bytes. Έχουμε μνήμη άμεσης απεικόνισης οπότε:

Οπότε, για την ετικέτα:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Επαναλήψεις** | **x[i]** | **y[i]** | **x[i+2]** | **x[i]** |
| *1η* (i = 0) | *miss* | *miss* | *miss* | ***hit*** |
| *2η* (i = 1) | ***hit*** | *miss* | *miss* | ***hit*** |
| *3η* (i = 2) | ***hit*** | *miss* | *miss* | *miss* |
| *4η* (i = 3) | ***hit*** | *miss* | ***hit*** | *miss* |
| *5η* (i = 4) | ***hit*** | *miss* | *miss* | ***hit*** |
| *6η* (i = 5) | ***hit*** | *miss* | *miss* | ***hit*** |
| *7η* (i = 6) | ***hit*** | *miss* | *miss* | *miss* |
| *8η* (i = 7) | ***hit*** | *miss* | ***hit*** | *miss* |

**2.**

*Πίνακας 1*

Ο παραπάνω πίνακας προκύπτει ως εξής:

Έχουμε τις διευθύνσεις

και

Τα στοιχεία των πινάκων είναι στοιχεία κινητής υποδιαστολής διπλής ακρίβειας και μεγέθους 8 bytes το καθένα.

Συνεπώς θα ισχύει:

Και

Παρατηρούμε δηλαδή ότι ανά τέσσερα στοιχεία οι πίνακες έχουν ίδιο , συνεπώς θα λάβει μέρος η ακόλουθη διαδικασία:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***INDEX*** | **V** | ***DATA*** | | | |  |
| **00** | **0** |  |  |  |  | |
| **01** | **0** |  |  |  |  | |
| **10** | **0** |  |  |  |  | |
| **11** | **0** |  |  |  |  | |

*Πίνακας της cache*

***1η Επανάληψη (i = 0):***

Ο πολλαπλασιαστέος ελέγχεται αν υπάρχει στην cache. Δεν υπάρχει οπότε έχουμε *miss* και τοποθετείται στην πρώτη γραμμή της cache (λαμβανόμενο από την κύρια μνήμη) το block ().

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***INDEX*** | **V** | ***DATA*** | | | |  |
| **00** | **1** |  |  |  |  | |
| **01** | **0** |  |  |  |  | |
| **10** | **0** |  |  |  |  | |
| **11** | **0** |  |  |  |  | |

Στη συνέχεια, Ο πολλαπλασιαστέος ελέγχεται αν υπάρχει στην cache. Δεν υπάρχει οπότε έχουμε *miss* και τοποθετείται στην πρώτη γραμμή της cache (λαμβανόμενο από την κύρια μνήμη) το block ().

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***INDEX*** | **V** | ***DATA*** | | | |  |
| **00** | **1** |  |  |  |  | |
| **01** | **0** |  |  |  |  | |
| **10** | **0** |  |  |  |  | |
| **11** | **0** |  |  |  |  | |

Ύστερα, Ο όρος ελέγχεται αν υπάρχει στην cache. Δεν υπάρχει οπότε έχουμε *miss* και τοποθετείται στην πρώτη γραμμή της cache (λαμβανόμενο από την κύρια μνήμη) το block ().

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***INDEX*** | **V** | ***DATA*** | | | |  |
| **00** | **1** |  |  |  |  | |
| **01** | **0** |  |  |  |  | |
| **10** | **0** |  |  |  |  | |
| **11** | **0** |  |  |  |  | |

Τέλος, το αποτέλεσμα θα γραφτεί στο το οποίο βρίσκεται στην cache άρα έχουμε ***hit****.*

***2η Επανάληψη (i = 1):***

Ακολουθώντας την ίδια διαδικασία, εν συντομία, έχουμε:

***hit*** στο , αφού υπάρχει στην cache από την προηγούμενη επανάληψη

*miss* στο και τοποθέτηση του block στην cache (αντικαθιστά το block )

*miss* στο και τοποθέτηση του block στην cache (αντικαθιστά το block )

***hit*** στο , αφού υπάρχει στην cache από το προηγούμενο

***3η Επανάληψη (i = 2):***

*Όμοια:*

***hit*** στο , αφού υπάρχει στην cache από την προηγούμενη επανάληψη

*miss* στο και τοποθέτηση του block στην cache (αντικαθιστά το block )

*miss* στο και τοποθέτηση του block στην cache στην δεύτερη γραμμή ():

*miss* στο και τοποθέτηση του block στην cache (αντικαθιστά το block )

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***INDEX*** | **V** | ***DATA*** | | | |  |
| **00** | **1** |  |  |  |  | |
| **01** | **1** |  |  |  |  | |
| **10** | **0** |  |  |  |  | |
| **11** | **0** |  |  |  |  | |

***4η Επανάληψη (i = 3):***

Έχουμε:  
***hit*** στο , αφού υπάρχει στην cache από την προηγούμενη επανάληψη

*miss* στο και τοποθέτηση του block στην cache (αντικαθιστά το block )

***hit*** στο , αφού υπάρχει στην cache από το προηγούμενο

*miss* στο και τοποθέτηση του block στην cache (αντικαθιστά το block )

***5η Επανάληψη (i = 4):***

***hit*** στο , αφού υπάρχει στην cache από την προηγούμενη επανάληψη

*miss* στο και τοποθέτηση του block στην cache (αντικαθιστά το block ) ()

*miss* στο και τοποθέτηση του block στην cache (αντικαθιστά το block )

***hit*** στο , αφού υπάρχει στην cache από το προηγούμενο

Παρατηρούμε ότι, με εξαίρεση την πρώτη επανάληψη, το “μοτίβο” των ***hits*** και *misses* επαναλαμβάνεται κάθε τέσσερις επαναλήψεις (οι τέσσερις επαναλήψεις υπογραμμισμένες με κίτρινο χρώμα στον πίνακα 1).

Συνεπώς από την 2η επανάληψη μέχρι και την 125η (124 επαναλήψεις) το μοτίβο αυτό θα επαναληφθεί για φορές.

Για τις επαναλήψεις 126, 127 και 128 θα ισχύει ότι και για τις επαναλήψεις 5, 6, 7.

Έχουμε:

1η επανάληψη 1 ***hit***

2η-4η επαναλήψεις 7 ***hits***

5η-7η επαναλήψεις 5 ***hits***

Οπότε συνολικά ο αριθμός των ***hits*** θα είναι:

**3.** Μπορούμε να βρούμε τον αριθμό των *misses* με παρόμοιο τρόπο με το ερώτημα 2. Ένας άλλος τρόπος είναι ο ακόλουθος:

Έχουμε ***hits***. Σε κάθε επανάληψη έχουμε τέσσερις προσπελάσεις στην cache και έχουμε συνολικά επαναλήψεις. Οπότε ο αριθμός των ***hits*** και *misses* στο πρόγραμμα είναι προσπελάσεις. Οπότε ο αριθμός των *misses* είναι:

**4.** Το ποσοστό ευστοχίας θα είναι:

**5.** Θα αυξηθούν τα hits και συνάμα θα μειωθούν τα misses.

**5.1)**

|  |  |  |
| --- | --- | --- |
| ***TAG*** | ***INDEX*** | ***OFFSET*** |

Έχουμε block μεγέθους 32 bytes, άρα το offset θα αποτελείται από:

Θα έχουμε 4 γραμμές, (όπως και στο Direct Mapping) και επιπλέον:

Οπότε, για την ετικέτα:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Επαναλήψεις** | **x[i]** | **y[i]** | **x[i+2]** | **x[i]** |
| *1η* (i = 0) | *miss* | *miss* | ***hit*** | ***hit*** |
| *2η* (i = 1) | ***hit*** | ***hit*** | ***hit*** | ***hit*** |
| *3η* (i = 2) | ***hit*** | ***hit*** | *miss* | ***hit*** |
| *4η* (i = 3) | ***hit*** | ***hit*** | ***hit*** | ***hit*** |
| *5η* (i = 4) | ***hit*** | *miss* | ***hit*** | ***hit*** |
| *6η* (i = 5) | ***hit*** | ***hit*** | ***hit*** | ***hit*** |
| *7η* (i = 6) | ***hit*** | ***hit*** | *miss* | ***hit*** |
| *8η* (i = 7) | ***hit*** | ***hit*** | ***hit*** | ***hit*** |

**5.2)**

*Πίνακας 2*

Όμοια με το ερώτημα 2.:

Και

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***INDEX*** | **V** | ***DATA*** | | | |  |
| **0** | **0** |  |  |  |  | |
| **0** |  |  |  |  | |
| **1** | **0** |  |  |  |  | |
| **0** |  |  |  |  | |

*Πίνακας της cache*

***1η Επανάληψη (i = 0):***

Ο πολλαπλασιαστέος ελέγχεται αν υπάρχει στην cache. Δεν υπάρχει οπότε έχουμε *miss* και τοποθετείται στην πρώτη γραμμή της cache (λαμβανόμενο από την κύρια μνήμη) το block ().

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***INDEX*** | **V** | ***DATA*** | | | |  |
| **0** | **1** |  |  |  |  | |
| **0** |  |  |  |  | |
| **1** | **0** |  |  |  |  | |
| **0** |  |  |  |  | |

Στη συνέχεια, Ο πολλαπλασιαστέος ελέγχεται αν υπάρχει στην cache. Δεν υπάρχει οπότε έχουμε *miss* και τοποθετείται στην δεύτερη γραμμή της cache (λαμβανόμενο από την κύρια μνήμη) το block ().

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***INDEX*** | **V** | ***DATA*** | | | |  |
| **0** | **1** |  |  |  |  | |
| **1** |  |  |  |  | |
| **1** | **0** |  |  |  |  | |
| **0** |  |  |  |  | |

Ύστερα, Ο όρος ελέγχεται αν υπάρχει στην cache. Υπάρχει, οπότε έχουμε ***hit***.

Τέλος, το αποτέλεσμα θα γραφτεί στο το οποίο βρίσκεται στην cache άρα έχουμε πάλι ***hit****.*

***2η Επανάληψη (i = 1):***

Ακολουθώντας την ίδια διαδικασία, εν συντομία, έχουμε:

***hit*** στο , αφού υπάρχει στην cache από την προηγούμενη επανάληψη

***hit*** στο , αφού υπάρχει στην cache από την προηγούμενη επανάληψη

***hit*** στο , αφού υπάρχει στην cache από την προηγούμενη επανάληψη

***hit*** στο , αφού υπάρχει στην cache

***3η Επανάληψη (i = 2):***

*Όμοια:*

***hit*** στο , αφού υπάρχει στην cache

***hit*** στο , αφού υπάρχει στην cache

*miss* στο και τοποθέτηση του block στην cache στην τρίτη γραμμή ():

***hit*** στο , αφού υπάρχει στην

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***INDEX*** | **V** | ***DATA*** | | | |  |
| **0** | **1** |  |  |  |  | |
| **1** |  |  |  |  | |
| **1** | **1** |  |  |  |  | |
| **0** |  |  |  |  | |

***4η Επανάληψη (i = 3):***

Έχουμε:  
***hit*** στο , αφού υπάρχει στην cache

***hit*** στο , αφού υπάρχει στην cache

***hit*** στο , αφού υπάρχει στην cache

***hit*** στο , αφού υπάρχει στην cache

***5η Επανάληψη (i = 4):***

***hit*** στο , αφού υπάρχει στην cache

*miss* στο και τοποθέτηση του block στην cache στην τέταρτη γραμμή ()

***hit*** στο , αφού υπάρχει στην cache

***hit*** στο , αφού υπάρχει στην cache

Στην έβδομη επανάληψη, θα έχουμε *miss* στο , οπότε με την LRU (Least Recently Used), θα αντικατασταθεί το block (Έχει την περισσότερη ώρα να χρησιμοποιηθεί, αφού τελευταία φορά χρησιμοποιήθηκε το block από το πρώτο set) με το block ().

Παρατηρούμε πάλι ότι, με εξαίρεση την πρώτη επανάληψη, το “μοτίβο” των ***hits*** και *misses* επαναλαμβάνεται κάθε τέσσερις επαναλήψεις (οι τέσσερις επαναλήψεις υπογραμμισμένες με κίτρινο χρώμα στον πίνακα 2).

Συνεπώς από την 2η επανάληψη μέχρι και την 125η (124 επαναλήψεις) το μοτίβο αυτό θα επαναληφθεί για φορές.

Για τις επαναλήψεις 126, 127 και 128 θα ισχύει ότι και για τις επαναλήψεις 5, 6, 7.

Έχουμε:

1η επανάληψη 2 ***hit***

2η-4η επαναλήψεις 14 ***hits***

5η-7η επαναλήψεις 11 ***hits***

Οπότε συνολικά ο αριθμός των ***hits*** θα είναι:

**5.3)** Όπως και στο 3. ερώτημα θα ισχύει:

Έχουμε ***hits***. Θα έχουμε πάλι προσπελάσεις. Οπότε ο αριθμός των *misses* είναι:

**5.4)** Το ποσοστό ευστοχίας θα είναι:

(Σχεδόν διπλάσιο)

**Άσκηση 3.2**

**1.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **DISK0** | **DISK1** | **DISK2** | **DISK3** |
| **STRIPE0** | 1100 | 1111 | 1000 | 1011 |
| **STRIPE1** | 0010 | 0001 | 1010 | 1001 |
| **STRIPE2** | 1010 | 1110 | 1111 | 1011 |
| **STRIPE3** | 1001 | 0111 | 0011 | 1101 |

Θα ισχύει:

**2.** Αφού γίνει η εγγραφή της νέας τιμής “0100” στο **STRIPE2** του **DISK3**, θα πρέπει να αλλάξει και το αντίστοιχο block ισοτιμίας του **STRIPE2.**

Θα ισχύει:

**3.** Εφόσον καταστραφεί ο **DISK3**, ο ελεγκτής RAID 5 θα εξυπηρετήσει τις αιτήσεις για πληροφορίες που είχε ο κατεστραμμένος δίσκος, χρησιμοποιώντας όλους τους άλλους δίσκους καθώς και τις τιμές της ισοτιμίας. Επομένως για το **STRIPE1**: