



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
Ακ. έτος 2020-2021, 5ο εξάμηνο, ΣΗΜΜΥ

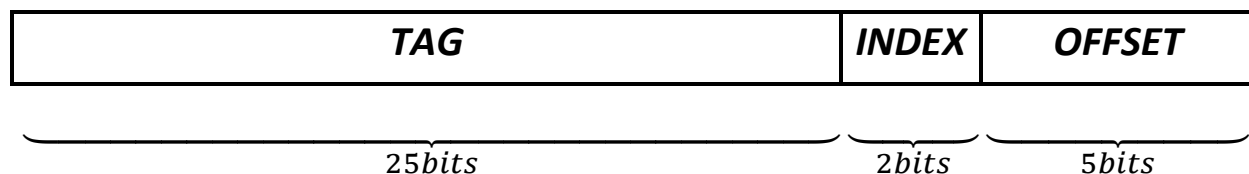
ΤΜΗΜΑ 1ο (Α - ΚΑΣ)
3^η ΣΕΙΡΑ ΑΣΚΗΣΕΩΝ ΣΤΗΝ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΠΟΛΟΓΙΣΤΩΝ
Ημερομηνία παράδοσης: 14/02/2021

Θοδωρής Αράπης

EI18028

Άσκηση 3.1

1.



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

$$\mathbf{OFFSET} = \log_2(32) = 5 \text{ bits}$$

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

$$128 \text{ bytes} = (\text{γραμμές}) \cdot (\text{blocks}) = 2^{\mathbf{INDEX}} \cdot 2^{\mathbf{OFFSET}} = 2^{\mathbf{INDEX}} \cdot 32 \Rightarrow$$

$$\mathbf{INDEX} = \log_2\left(\frac{128}{32}\right) = 2 \text{ bits}$$

Οπότε, για την ετικέτα:

$$\mathbf{TAG} = 32 - \mathbf{OFFSET} - \mathbf{INDEX} \Rightarrow$$

$$\mathbf{TAG} = 25 \text{ bits}$$

2.

Επαναλήψεις	x[i]	y[i]	x[i+2]	x[i]
1 ^η (i = 0)	<i>miss</i>	<i>miss</i>	<i>miss</i>	<i>hit</i>
2 ^η (i = 1)	<i>hit</i>	<i>miss</i>	<i>miss</i>	<i>hit</i>
3 ^η (i = 2)	<i>hit</i>	<i>miss</i>	<i>miss</i>	<i>miss</i>
4 ^η (i = 3)	<i>hit</i>	<i>miss</i>	<i>hit</i>	<i>miss</i>
5 ^η (i = 4)	<i>hit</i>	<i>miss</i>	<i>miss</i>	<i>hit</i>
6 ^η (i = 5)	<i>hit</i>	<i>miss</i>	<i>miss</i>	<i>hit</i>
7 ^η (i = 6)	<i>hit</i>	<i>miss</i>	<i>miss</i>	<i>miss</i>
8 ^η (i = 7)	<i>hit</i>	<i>miss</i>	<i>hit</i>	<i>miss</i>

Πίνακας 1

Ο παραπάνω πίνακας προκύπτει ως εξής:

Έχουμε τις διευθύνσεις

0x0000A100 → x[0] και 0xA080C200 → y[0]

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

Συνεπώς θα ισχύει:

$$\begin{array}{l}
 0x0000A100 \rightarrow x[0] \\
 0x0000A108 \rightarrow x[1] \\
 0x0000A110 \rightarrow x[2] \\
 0x0000A118 \rightarrow x[3]
 \end{array}
 \left. \vphantom{\begin{array}{l} 0x0000A100 \rightarrow x[0] \\ 0x0000A108 \rightarrow x[1] \\ 0x0000A110 \rightarrow x[2] \\ 0x0000A118 \rightarrow x[3] \end{array}} \right\} INDEX = 00,
 \quad
 \begin{array}{l}
 0x0000A120 \rightarrow x[4] \\
 0x0000A128 \rightarrow x[5] \\
 0x0000A130 \rightarrow x[6] \\
 0x0000A138 \rightarrow x[7]
 \end{array}
 \left. \vphantom{\begin{array}{l} 0x0000A120 \rightarrow x[4] \\ 0x0000A128 \rightarrow x[5] \\ 0x0000A130 \rightarrow x[6] \\ 0x0000A138 \rightarrow x[7] \end{array}} \right\} INDEX = 01$$

Και

$$\begin{array}{l}
 0xA080C200 \rightarrow y[0] \\
 0xA080C208 \rightarrow y[1] \\
 0xA080C210 \rightarrow y[2] \\
 0xA080C218 \rightarrow y[3]
 \end{array}
 \left. \vphantom{\begin{array}{l} 0xA080C200 \rightarrow y[0] \\ 0xA080C208 \rightarrow y[1] \\ 0xA080C210 \rightarrow y[2] \\ 0xA080C218 \rightarrow y[3] \end{array}} \right\} INDEX = 00,
 \quad
 \begin{array}{l}
 0xA080C220 \rightarrow y[4] \\
 0xA080C228 \rightarrow y[5] \\
 0xA080C230 \rightarrow y[6] \\
 0xA080C238 \rightarrow y[7]
 \end{array}
 \left. \vphantom{\begin{array}{l} 0xA080C220 \rightarrow y[4] \\ 0xA080C228 \rightarrow y[5] \\ 0xA080C230 \rightarrow y[6] \\ 0xA080C238 \rightarrow y[7] \end{array}} \right\} INDEX = 01$$

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

INDEX	V	DATA			
00	0				
01	0				
10	0				
11	0				

Πίνακας της cache

1^η Επανάληψη (i = 0):

$$x[0] = x[0] * y[0] - x[2]$$

Ο πολλαπλασιαστέος $x[0]$ ελέγχεται αν υπάρχει στην cache. Δεν υπάρχει οπότε έχουμε **miss** και τοποθετείται στην πρώτη γραμμή της cache (λαμβάνόμενο από την κύρια μνήμη) το block $x[0] - x[3]$ (**INDEX** = 00).

INDEX	V	DATA			
00	1	$x[0]$	$x[1]$	$x[2]$	$x[3]$
01	0				
10	0				
11	0				

Στη συνέχεια, Ο πολλαπλασιαστέος $y[0]$ ελέγχεται αν υπάρχει στην cache. Δεν υπάρχει οπότε έχουμε **miss** και τοποθετείται στην πρώτη γραμμή της cache (λαμβάνόμενο από την κύρια μνήμη) το block $y[0] - y[3]$ (**INDEX** = 00).

INDEX	V	DATA			
00	1	$y[0]$	$y[1]$	$y[2]$	$y[3]$
01	0				
10	0				
11	0				

Ύστερα, Ο όρος $x[2]$ ελέγχεται αν υπάρχει στην cache. Δεν υπάρχει οπότε έχουμε **miss** και τοποθετείται στην πρώτη γραμμή της cache (λαμβανόμενο από την κύρια μνήμη) το block $x[0] - x[3]$ (**INDEX** = 00).

INDEX	V	DATA			
00	1	$x[0]$	$x[1]$	$x[2]$	$x[3]$
01	0				
10	0				
11	0				

Τέλος, το αποτέλεσμα θα γραφτεί στο $x[0]$ το οποίο βρίσκεται στην cache άρα έχουμε **hit**.

2^η Επανάληψη ($i = 1$):

$$x[1] = x[1] * y[1] - x[3]$$

Ακολουθώντας την ίδια διαδικασία, εν συντομία, έχουμε:

hit στο $x[1]$, αφού υπάρχει στην cache από την προηγούμενη επανάληψη

miss στο $y[1]$ και τοποθέτηση του block $y[0] - y[3]$ στην cache (αντικαθιστά το block $x[0] - x[3]$)

miss στο $x[3]$ και τοποθέτηση του block $x[0] - x[3]$ στην cache (αντικαθιστά το block $y[0] - y[3]$)

hit στο $x[1]$, αφού υπάρχει στην cache από το προηγούμενο

3^η Επανάληψη ($i = 2$):

$$x[2] = x[2] * y[2] - x[4]$$

Όμοια:

hit στο $x[2]$, αφού υπάρχει στην cache από την προηγούμενη επανάληψη

miss στο $y[2]$ και τοποθέτηση του block $y[0] - y[3]$ στην cache (αντικαθιστά το block $x[0] - x[3]$)

miss στο $x[4]$ και τοποθέτηση του block $x[4] - x[7]$ στην cache στην δεύτερη γραμμή (**INDEX** = 01):

miss στο $x[2]$ και τοποθέτηση του block $x[0] - x[3]$ στην cache (αντικαθιστά το block $y[0] - y[3]$)

INDEX	V	DATA			
00	1	$x[0]$	$x[1]$	$x[2]$	$x[3]$
01	1	$x[4]$	$x[5]$	$x[6]$	$x[7]$
10	0				
11	0				

4^η Επανάληψη ($i = 3$):

$$x[3] = x[3] * y[3] - x[5]$$

Έχουμε:

hit στο $x[3]$, αφού υπάρχει στην cache από την προηγούμενη επανάληψη

miss στο $y[3]$ και τοποθέτηση του block $y[0] - y[3]$ στην cache (αντικαθιστά το block $x[0] - x[3]$)

hit στο $x[5]$, αφού υπάρχει στην cache από το προηγούμενο

miss στο $x[3]$ και τοποθέτηση του block $x[0] - x[3]$ στην cache (αντικαθιστά το block $y[0] - y[3]$)

5^η Επανάληψη ($i = 4$):

$$x[4] = x[4] * y[4] - x[6]$$

hit στο $x[4]$, αφού υπάρχει στην cache από την προηγούμενη επανάληψη

miss στο $y[4]$ και τοποθέτηση του block $y[4] - y[7]$ στην cache (αντικαθιστά το block $x[4] - x[7]$) (**INDEX** = 01)

miss στο $x[6]$ και τοποθέτηση του block $x[4] - x[7]$ στην cache (αντικαθιστά το block $y[4] - y[7]$)

hit στο $x[4]$, 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 2^η επανάληψη μέχρι και την 125^η (124 επαναλήψεις) το μοτίβο αυτό θα επαναληφθεί για $\frac{124}{4} = 31$ φορές.

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

Έχουμε:

1^η επανάληψη \rightarrow 1 **hit**

2^η-4^η επαναλήψεις \rightarrow 7 **hits**

5^η-7^η επαναλήψεις \rightarrow 5 **hits**

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

$$HITS = 1 \cdot 1 + 31 \cdot 7 + 1 \cdot 5 = 223 \text{ hits}$$

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

Έχουμε 223 **hits**. Σε κάθε επανάληψη έχουμε τέσσερις προσπελάσεις στην cache και έχουμε συνολικά 128 επαναλήψεις. Οπότε ο αριθμός των **hits** και **misses** στο πρόγραμμα είναι $4 \cdot 128 = 512$ προσπελάσεις. Οπότε ο αριθμός των **misses** είναι:

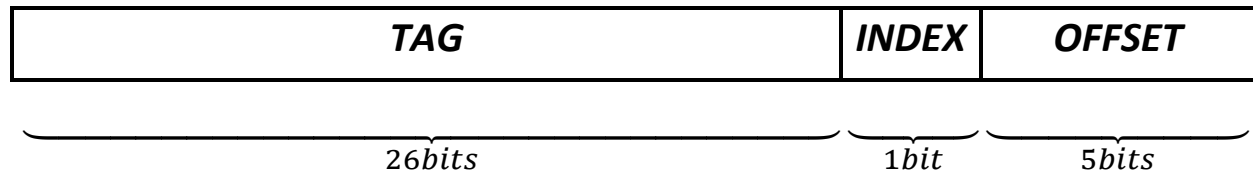
$$MISSES = 512 - 223 = 289 \text{ misses}$$

4. Το ποσοστό ευστοχίας θα είναι:

$$\text{hit ratio}\% = \frac{\text{ευστοχίες}}{\text{προσπελάσεις}} = \frac{223}{512} \cdot 100\% = 0.44\%$$

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

5.1)



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

$$\textbf{OFFSET} = \log_2(32) = 5 \text{ bits}$$

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

$$(\text{Number of Sets}) = \frac{(\text{γραμμές})}{(\text{set})} = \frac{4}{2} = 2^{\textbf{INDEX}} \text{ sets} \Rightarrow$$

$$\textbf{INDEX} = \log_2(2) = 1 \text{ bit}$$

Οπότε, για την ετικέτα:

$$\textbf{TAG} = 32 - \textbf{OFFSET} - \textbf{INDEX} \Rightarrow$$

$$\textbf{TAG} = 26 \text{ bits}$$

5.2)

Επαναλήψεις	x[i]	y[i]	x[i+2]	x[i]
1 ^η (i = 0)	<i>miss</i>	<i>miss</i>	<i>hit</i>	<i>hit</i>
2 ^η (i = 1)	<i>hit</i>	<i>hit</i>	<i>hit</i>	<i>hit</i>
3 ^η (i = 2)	<i>hit</i>	<i>hit</i>	<i>miss</i>	<i>hit</i>
4 ^η (i = 3)	<i>hit</i>	<i>hit</i>	<i>hit</i>	<i>hit</i>
5 ^η (i = 4)	<i>hit</i>	<i>miss</i>	<i>hit</i>	<i>hit</i>
6 ^η (i = 5)	<i>hit</i>	<i>hit</i>	<i>hit</i>	<i>hit</i>
7 ^η (i = 6)	<i>hit</i>	<i>hit</i>	<i>miss</i>	<i>hit</i>
8 ^η (i = 7)	<i>hit</i>	<i>hit</i>	<i>hit</i>	<i>hit</i>

Πίνακας 2

Όμοια με το ερώτημα 2.:

$$\left. \begin{array}{l} 0x0000A100 \rightarrow x[0] \\ 0x0000A108 \rightarrow x[1] \\ 0x0000A110 \rightarrow x[2] \\ 0x0000A118 \rightarrow x[3] \end{array} \right\} INDEX = 0, \quad \left. \begin{array}{l} 0x0000A120 \rightarrow x[4] \\ 0x0000A128 \rightarrow x[5] \\ 0x0000A130 \rightarrow x[6] \\ 0x0000A138 \rightarrow x[7] \end{array} \right\} INDEX = 1$$

Και

$$\left. \begin{array}{l} 0xA080C200 \rightarrow y[0] \\ 0xA080C208 \rightarrow y[1] \\ 0xA080C210 \rightarrow y[2] \\ 0xA080C218 \rightarrow y[3] \end{array} \right\} INDEX = 0, \quad \left. \begin{array}{l} 0xA080C220 \rightarrow y[4] \\ 0xA080C228 \rightarrow y[5] \\ 0xA080C230 \rightarrow y[6] \\ 0xA080C238 \rightarrow y[7] \end{array} \right\} INDEX = 1$$

INDEX	V	DATA			
0	0				
	0				
1	0				
	0				

Πίνακας της cache

1^η Επανάληψη (i = 0):

$$x[0] = x[0] * y[0] - x[2]$$

Ο πολλαπλασιαστέος $x[0]$ ελέγχεται αν υπάρχει στην cache. Δεν υπάρχει οπότε έχουμε *miss* και τοποθετείται στην πρώτη γραμμή της cache (λαμβάνόμενο από την κύρια μνήμη) το block $x[0] - x[3]$ ($INDEX = 0$).

INDEX	V	DATA			
0	1	$x[0]$	$x[1]$	$x[2]$	$x[3]$
	0	$y[0]$	$y[1]$	$y[2]$	$y[3]$
1	0				
	0				

Στη συνέχεια, Ο πολλαπλασιαστέος $y[0]$ ελέγχεται αν υπάρχει στην cache. Δεν υπάρχει οπότε έχουμε *miss* και τοποθετείται στην δεύτερη γραμμή της cache (λαμβάνόμενο από την κύρια μνήμη) το block $y[0] - y[3]$ (**INDEX** = 0).

INDEX	V	DATA			
0	1	$x[0]$	$x[1]$	$x[2]$	$x[3]$
	1	$y[0]$	$y[1]$	$y[2]$	$y[3]$
1	0				
	0				

Ύστερα, Ο όρος $x[2]$ ελέγχεται αν υπάρχει στην cache. Υπάρχει, οπότε έχουμε *hit*.

Τέλος, το αποτέλεσμα θα γραφτεί στο $x[0]$ το οποίο βρίσκεται στην cache άρα έχουμε πάλι *hit*.

2^η Επανάληψη ($i = 1$):

$$x[1] = x[1] * y[1] - x[3]$$

Ακολουθώντας την ίδια διαδικασία, εν συντομία, έχουμε:

hit στο $x[1]$, αφού υπάρχει στην cache από την προηγούμενη επανάληψη

hit στο $y[1]$, αφού υπάρχει στην cache από την προηγούμενη επανάληψη

hit στο $x[3]$, αφού υπάρχει στην cache από την προηγούμενη επανάληψη

hit στο $x[1]$, αφού υπάρχει στην cache

3^η Επανάληψη ($i = 2$):

$$x[2] = x[2] * y[2] - x[4]$$

Όμοια:

hit στο $x[2]$, αφού υπάρχει στην cache

hit στο $y[2]$, αφού υπάρχει στην cache

miss στο $x[4]$ και τοποθέτηση του block $x[4] - x[7]$ στην cache στην τρίτη γραμμή (**INDEX** = 1):

hit στο $x[2]$, αφού υπάρχει στην

INDEX	V	DATA			
0	1	$x[0]$	$x[1]$	$x[2]$	$x[3]$
	1	$y[0]$	$y[1]$	$y[2]$	$y[3]$
1	1	$x[4]$	$x[5]$	$x[6]$	$x[7]$
	0				

4^η Επανάληψη ($i = 3$):

$$x[3] = x[3] * y[3] - x[5]$$

Έχουμε:

hit στο $x[3]$, αφού υπάρχει στην cache

hit στο $y[3]$, αφού υπάρχει στην cache

hit στο $x[5]$, αφού υπάρχει στην cache

hit στο $x[3]$, αφού υπάρχει στην cache

5^η Επανάληψη ($i = 4$):

$$x[4] = x[4] * y[4] - x[6]$$

hit στο $x[4]$, αφού υπάρχει στην cache

miss στο $y[4]$ και τοποθέτηση του block $y[4] - y[7]$ στην cache στην τέταρτη γραμμή (**INDEX** = 1)

hit στο $x[6]$, αφού υπάρχει στην cache

hit στο $x[4]$, αφού υπάρχει στην cache

Στην έβδομη επανάληψη, θα έχουμε *miss* στο $x[8]$, οπότε με την LRU (Least Recently Used), θα αντικατασταθεί το block $y[0] - y[3]$ (Έχει την περισσότερη ώρα να χρησιμοποιηθεί, αφού τελευταία φορά χρησιμοποιήθηκε το block $x[0] - x[3]$ από το πρώτο set) με το block $x[8] - x[11]$ (**INDEX** = 0).

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 2^η επανάληψη μέχρι και την 125^η (124 επαναλήψεις) το μοτίβο αυτό θα επαναληφθεί για $\frac{124}{4} = 31$ φορές.

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

Έχουμε:

1^η επανάληψη $\rightarrow 2$ *hit*

2^η-4^η επαναλήψεις $\rightarrow 14$ *hits*

5^η-7^η επαναλήψεις $\rightarrow 11$ *hits*

Οπότε συνολικά ο αριθμός των *hits* θα είναι:

$$\mathbf{HITS} = 1 \cdot 2 + 31 \cdot 14 + 1 \cdot 11 = 447 \mathbf{hits}$$

5.3) Όπως και στο 3. ερώτημα θα ισχύει:

Έχουμε 447 *hits*. Θα έχουμε πάλι 512 προσπελάσεις. Οπότε ο αριθμός των *misses* είναι:

$$\mathbf{MISSES} = 512 - 447 = 65 \mathbf{misses}$$

5.4) Το ποσοστό ευστοχίας θα είναι:

$$\mathit{hit\ ratio\%} = \frac{\mathit{ευστοχίες}}{\mathit{προσπελάσεις}} = \frac{447}{512} \cdot 100\% = 87.3\%$$

(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:

$$S0.D3 = (1100)XOR(1111)XOR(1000) = 1011$$

$$S1.D2 = (0010)XOR(0001)XOR(1001) = 1010$$

$$S2.D1 = (1010)XOR(1111)XOR(1011) = 1110$$

$$S3.D0 = (0111)XOR(0011)XOR(1101) = 1001$$

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:

$$S2.D1' = (\text{παλιά τιμή block})XOR(\text{νέα τιμή block})XOR(\text{παλιά ισοτιμία}) \Rightarrow$$

$$S2.D1' = (1011)XOR(0100)XOR(1110) = 0001$$

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**:

$$S1.D3 = (0010)XOR(0001)XOR(1010) = 1001$$