Αρχιτεκτονική Υπολογιστών 2020-2021

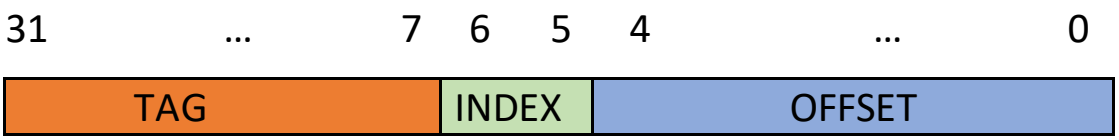
3η Σειρά Ασκήσεων

Βικέντιος Βιτάλης el18803

Άσκηση 3.1

1)Το μέγεθος του κάθε block είναι 32 bytes, δηλαδή 32*8 = 256 bits. Συμπερασματικά, γνωρίζοντας πως 2^5=32, θα χρειαστούμε 5 bits OFFSET. Για την κύρια μνήμη η οποία έχει μέγεθος 128 bytes και αποτελείται από (128/32) 4 blocks θα χρειαστούμε 2 INDEX bits. Τέλος, οι διευθύνσεις που χρησιμοποιούνται είναι μεγέθους TAG (32-5-2) = 25 bits.

Διαγραμματικά:



2,3) Κατά την εκτέλεση του προγράμματος θα γίνουν 128 επαναλήψεις λόγω του i. Κάθε στοιχείο καταλαμβάνει 8 διαδοχικές θέσεις στην μνήμη και διευθυνσιοδοτείται ανά byte. Μπορούν να αποθηκευτούν μέχρι και 4 στοιχεία ανά block. Τα 4 στοιχεία που αποθηκεύονται, όμως, δεν είναι τυχαία, αλλά αποτελούνται από το 1 στοιχείο που θα παρουσιάσει miss/hit όταν χρειαστεί να διαβαστεί ή να γραφεί και τα 3 γειτονικά του, που χαρακτηρίζονται από το ίδιο index. Το πρώτο στοιχείο του πίνακα χ βρίσκεται στην 0x0000A100 = 0d41216, το δεύτερο στην 0x0000A108 = 0d41224 κλπ. Το τελευταίο του στοιχείο βρίσκεται στην 0x0000A510 = 0d42256. Ανά 4 στοιχεία του x, αλλάζει κυκλικά το index τους. Για παράδειγμα τα στοιχεία x[0], x[1], x[2], x[3] έχουν index 00, τα x[4], x[5], x[6], x[7] έχουν index 01. Ομοίως, το πρώτο στοιχείο του y βρίσκεται στην 0xA080C200 = 0d2692792832, το δεύτερο στην 0xA080C208 = 0d2692792840 κλπ. , και το τελευταίο στην 0xA080C600 = 0d2692793856. Ανά 4 στοιχεία του y, αλλάζει το index και ακολουθεί το ίδιο μοτίβο με τον πίνακα χ. Δηλαδή, τα στοιχεία y[0], y[1], y[2], y[3] έχουν index 00 και τα y[4], y[5], y[6], y[7] έχουν index 01 κλπ.

Έχοντας όλα τα παραπάνω κατά νου, εξετάζουμε την εξής περίπτωση:

Για παράδειγμα, αν χρειαστεί να διαβαστεί το x[0] και παρουσιάσει miss, τότε στο block με index 00 θα

αποθηκευτούν τα στοιχεία x[0], x[1], x[2], x[3] τα οποία χαρακτηρίζονται όλα από index 00. Αν, χρειαστεί να διαβαστεί το x[7] και παρουσιάσει miss, τότε στο block με index 01 θα αποθηκευτούν τα στοιχεία x[4], x[5], x[6], x[7] τα οποία χαρακτηρίζονται όλα από index 01. Έτσι τα στοιχεία των δύο πινάκων ανταγωνίζονται μεταξύ τους ανά τετράδες για το αντίστοιχό τους block: τα x[0], x[1], x[2], x[3] θα ανταγωνιστούν με τα y[0], y[1], y[2], y[3] για το block με index 00. Δύο τετράδες στοιχείων του ίδιου πίνακα, π.χ. η x[0], x[1], x[2], x[3] και η x[4], x[5], x[6], x[7] δεν θα ανταγωνιστούν ποτέ μεταξύ τους για το ίδιο block, αφού χαρακτηρίζονται από διαφορετικό index, δηλαδή αποθηκεύονται σε διαφορετικά block.

Προκύπτει ο πίνακας:

i	x[i]	y[i]	x[i+2]	x[i]				
0	[0](00)	[0](00)	[2](00)	[0](00)	m	m	m	h
1	[1](00)	[1](00)	[3](00)	[1](00)	h	m	m	h
2	[2](00)	[2](00)	[4](01)	[2](00)	h	m	m	m
3	[3](00)	[3](00)	[5](01)	[3](00)	h	m	h	m
4	[4](01)	[4](01)	[6](01)	[4](01)	h	m	m	h
5	[5](01)	[5](01)	[7](01)	[5](01)	h	m	m	h
6	[6](01)	[6](01)	[8](10)	[6](01)	h	m	m	m
7	[7](01)	[7](01)	[9](10)	[7](01)	h	m	h	m
8	[8](10)	[8](10)	[10](10)	[8](10)	h	m	m	h
9	[9](10)	[9](10)	[11](10)	[9](10)	h	m	m	h
10	[10](10)	[10](10)	[12](11)	[10](10)	h	m	m	m
11	[11](10)	[11](10)	[13](11)	[11](10)	h	m	h	m
12	[12](11)	[12](11)	[14](11)	[12](11)	h	m	m	h
13	[13](11)	[13](11)	[15](11)	[13](11)	h	m	m	h

Παρατηρούμε μοτίβο miss-hit να δημιουργείται από την 3η επανάληψη και μετά και για κάθε 4 επαναλήψεις (i >= 2). Βλέπουμε ότι μέχρι την 128η επανάληψη, δηλαδή για i = 127 θα έχουν λάβει χώρα 31 τέτοιες τετράδες και η 32η τετράδα θα έχει σταματήσει στη μέση.

Σε κάθε τετράδα, έχουμε 7 hits και 9 misses, άρα:

3 hits στις 2 πράσινες γραμμές +

31 * 7 hits σε κάθε τετράδα +

3 hits που αντιστοιχούν στην τελευταία τετράδα η οποία σταματάει στις 2 πρώτες γραμμές της.

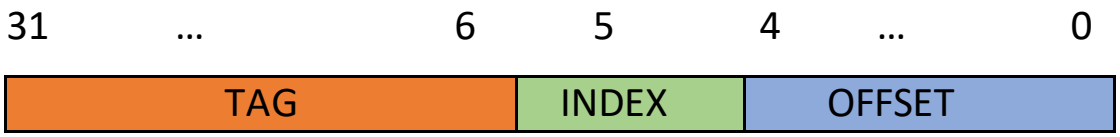
Συνολικά *223 hits*, και 5 + 31 * 9 + 5 = *289 misses.*

4) Hit Ratio

hit% = $hits/\ hits+\ misses\ *$ 100= 223 / 512 $*$ 100 = 43.55 %

5) Με 2-way set cache με LRU πολιτική αντικατάστασης, γνωρίζουμε ότι ο αριθμός των block και το μέγεθος κάθε block παραμένουν ανεπηρέαστοι. Είχαμε 4 block, τώρα θα έχουμε 2 sets, με 2 block των 32 byte έκαστο. Συνεπώς, χρειαζόμαστε μόνο ένα bit για το πεδίο INDEX, και πάλι 5 bits για το πεδίο OFFSET και 32 – 5 – 1 = 26 bits για το πεδίο TAG.

Σχηματικά:



Το INDEX κάθε στοιχείου πίνακα αλλάζει κάθε 4 στοιχεία, δηλαδή τα x[0], x[1], x[2], x[3] έχουν INDEX 0. Τα x[4], x[5], x[6], x[7] έχουν INDEX 1 κλπ. Όμοια για τα στοιχεία του y. Επίσης, αφού το 6ο bit κάθε διεύθυνσης είναι το INDEX πλέον, τα στοιχεία των πινάκων με παλιό INDEX 00 ή 01 θα προορίζονται για το set με INDEX 0, ενώ τα στοιχεία των πινάκων με παλιό INDEX 10 ή 11 θα προορίζονται για το set με INDEX 1. Δύο

διαδοχικές τετράδες στοιχείων του ίδιου πίνακα δεν θα ανταγωνίζονται ποτέ μεταξύ τους εξαιτίας των διαφορετικών INDEX και επιπλέον ο ανταγωνισμός μεταξύ των τετράδων στοιχείων x και y θα είναι μικρότερος, καθώς τώρα μπορούν να αποθηκευτούν μέχρι και 2 τετράδες σε κάθε set.

Εκτέλεση του βρόχου εκ νέου:

i	x[i]	y[i]	x[i+2]	x[i]				
0	[0](0)	[0](0)	[2](0)	[0](0)	m	m	h	h
1	[1](0)	[1](0)	[3](0)	[1](0)	h	h	h	h
2	[2](0)	[2](0)	[4](1)	[2](0)	h	h	m	h
3	[3](0)	[3](0)	[5](1)	[3](0)	h	h	h	h
4	[4](1)	[4](1)	[6](1)	[4](1)	h	m	h	h
5	[5](1)	[5](1)	[7](1)	[5](1)	h	h	h	h
6	[6](1)	[6](1)	[8](0)	[6](1)	h	h	m	h
7	[7](1)	[7](1)	[9](0)	[7](1)	h	h	h	h
8	[8](0)	[8](0)	[10](0)	[8](0)	h	m	h	h
9	[9](0)	[9](0)	[11](0)	[9](0)	h	h	h	h
10	[10](0)	[10](0)	[12](1)	[10](0)	h	h	m	h
11	[11](0)	[11](0)	[13](1)	[11](0)	h	h	h	h
12	[12](1)	[12](1)	[14](1)	[12](1)	h	m	h	h
13	[13](1)	[13](1)	[15](1)	[13](1)	h	h	h	h

Παρατηρούμε μοτίβο να λαμβάνει χώρα μετά από την 3η επανάληψη και κάθε 4 επαναλήψεις.

Δηλαδή:

6 + 31 * 14 + 7 = *447 hits* και 2 + 31 * 2 + 1 = *65 hits.*

Το ποσοστό ευστοχίας είναι πλέον αυξημένο:

Hit % = $hits \ / \ hits + misses * 100$ = 447 / 512 $*$ 100 = 87.3 %

Άσκηση 3.2

1)Parity missing bits:

- 1100 ⊕ 1111 ⊕ 1000 = 1011, STRIPE0
- 0010 ⊕ 0001 ⊕ 1001 = 1010, STRIPE1
- 1010 ⊕ 1111 ⊕ 1011 = 1110, STRIPE2
- 0111 ⊕ 0011 ⊕ 1101 = 1001, STRIPE3

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

2) XOR ανάμεσα στο παλιό και νέο περιεχόμενο του (STRIPE2, DISK3) και τα προηγούμενα parity bits:

$$1011 \oplus 0100 \oplus 1110 = 0001$$

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

3)Ανάκτηση του STRIPE1, DISK3, χρησιμοποιούμε τα περιεχόμενα των άλλων δίσκων:

$$0010 \oplus 0001 \oplus 1001 = 1010$$