



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

Μάθημα: Αρχιτεκτονική Υπολογιστών

Ονοματεπώνυμο: Ειρήνη Δόντη

A.M: 03119839

3^η Σειρά Γραπτών Ασκήσεων

5^ο Εξάμηνο

Τμήμα 1^ο

Αθήνα

2021 - 2022

Άσκηση 1

Δίνεται η κάτωθι ακολουθία 32-bit διευθύνσεων μνήμης:

3, 180, 43, 2, 191, 88, 190, 14, 181, 44, 186, 253.

Έχουμε στη διάθεση μας 3 επιλογές κρυφής μνήμης άμεσης απεικόνισης, μεγέθους 8 words.

Για την πρώτη επιλογή C1 (word blocks: 1 και Χρόνος Προσπέλασης: 2 κύκλοι), δημιουργούμε τον παρακάτω πίνακα:

DEC	BIN	TAG	INDEX	TAG10	INDEX10	HIT/MISS
3	00000011	00000	011	0	3	M
180	10110100	10110	100	22	4	M
43	00101011	00101	011	5	3	M
2	00000010	00000	010	0	2	M
191	10111111	10111	111	23	7	M
88	01011000	01011	000	11	0	M
190	10111110	10111	110	23	6	M
14	00001110	00001	110	1	6	M
181	10110101	10110	101	22	5	M
44	00101100	00101	100	5	4	M
186	10111010	10111	010	23	2	M
253	11111101	11111	101	31	5	M

Miss rate: 100%

Average Memory Access Time = $12 \cdot [\text{access time}] + 12 \cdot [\text{miss stall time}] = 12 \cdot 2 + 12 \cdot 25 = 324$ cycles.

Για την δεύτερη επιλογή C2 (word blocks: 2 και Χρόνος Προσπέλασης: 3 κύκλοι), δημιουργούμε τον παρακάτω πίνακα:

DEC	BIN	TAG	INDEX	TAG10	INDEX10	HIT/MISS
3	00000011	00000	01	0	1	M
180	10110100	10110	10	22	2	M
43	00101011	00101	01	5	1	M
2	00000010	00000	01	0	1	M
191	10111111	10111	11	23	3	M
88	01011000	01011	00	11	0	M
190	10111110	10111	11	23	3	H
14	00001110	00001	11	1	3	M
181	10110101	10110	10	22	2	H
44	00101100	00101	10	5	2	M
186	10111010	10111	01	23	1	M
253	11111101	11111	10	31	2	M

Miss rate: $10/12 = 83,333\%$

Average Memory Access Time = $12 * [\text{access time}] + 10 * [\text{miss stall time}] = 12 * 3 + 10 * 25 = 286 \text{ cycles.}$

Για την τρίτη επιλογή C3 (word blocks: 4 και Χρόνος Προσπέλασης: 5 κύκλοι), δημιουργούμε τον παρακάτω πίνακα:

DEC	BIN	TAG	INDEX	TAG10	INDEX10	HIT/MISS
3	00000011	00000	0	0	0	M
180	10110100	10110	1	22	1	M
43	00101011	00101	0	5	0	M
2	00000010	00000	0	0	0	M
191	10111111	10111	1	23	1	M
88	01011000	01011	0	11	0	M
190	10111110	10111	1	23	1	H
14	00001110	00001	1	1	1	M
181	10110101	10110	1	22	1	H
44	00101100	00101	1	5	1	M
186	10111010	10111	0	23	0	M
253	11111101	11111	1	31	1	M

Miss rate: $10/12 = 83,333\%$

Average Memory Access Time = $12 * [\text{access time}] + 10 * [\text{miss stall time}] = 12 * 5 + 10 * 25 = 310 \text{ cycles.}$

Παρατηρούμε ότι η μνήμη cache που βελτιστοποιεί το σύστημα είναι η C2, γιατί δημιουργεί μικρότερη Average Memory Access Time έναντι των άλλων μνημών cache. Με άλλα λόγια, επιλέξαμε τη C2 μνήμη cache, διότι έχει περισσότερα Hits και μικρότερο χρόνο προσπέλασης έναντι των άλλων μνημών (θεωρώντας ότι η miss stall time είναι σταθερή).

Άσκηση 2

a) Offset:

32 bytes block size = $2^5 \rightarrow 5$ bits

Index:

blocks (1 ανά γραμμή) = χωρητικότητα/(bytes*(1-way)) ή

$512/32 = 16$ ομάδες = $2^4 \rightarrow 4$ bits

Οπότε, Tag = $32 - 5 - 4 = 21$ bits

TAG	INDEX	OFFSET
21	4	5

X \rightarrow 0xFD00A000 \rightarrow 1111 1101 0000 0000 1010 0000 0000 0000

Y \rightarrow 0xA080C000 \rightarrow 1010 0000 1000 0000 1100 0000 0000 0000

Z \rightarrow 0xF1001000 \rightarrow 1111 0001 0000 0000 0001 0000 0000 0000

Δημιουργούμε το ανάλογο διάγραμμα που φαίνεται παρακάτω:

CACHE INDEX	X	Y	Z
0000	X[0]/X[1]/X[2]/ X[3]	Y[0]/Y[1]/Y[2]/ Y[3]	Z[0]/Z[1]/Z[2]/ Z[3]
0001	X[4]/X[5]/X[6]/ X[7]	Y[4]/Y[5]/Y[6]/ Y[7]	Z[4]/Z[5]/Z[6]/ Z[7]
0010	X[8]/X[9]/X[10]/ X[11]	Y[8]/Y[9]/Y[10]/ Y[11]	Z[8]/Z[9]/Z[10]/ Z[11]
....
....
1111	X[60]/X[61]/X[62]/ X[63]	Y[60]/Y[61]/Y[62]/ Y[63]	Z[60]/Z[61]/Z[62]/ Z[63]

Με περισσότερη λεπτομέρεια, ο πίνακας διευθύνσεων για τον πίνακα X είναι:

Στοιχεία Πίνακα X	tag	index	offset
X[0]-X[3]	1111 1101 0000 0000 1010 000	0000	0 0000
X[4]-X[7]	1111 1101 0000 0000 1010 000	0001	0 0000
X[8]-X[11]	1111 1101 0000 0000 1010 000	0010	0 0000
X[12]-X[15]	1111 1101 0000 0000 1010 000	0011	0 0000
X[16]-X[19]	1111 1101 0000 0000 1010 000	0100	0 0000
X[20]-X[23]	1111 1101 0000 0000 1010 000	0101	0 0000
X[24]-X[27]	1111 1101 0000 0000 1010 000	0110	0 0000
X[28]-X[31]	1111 1101 0000 0000 1010 000	0111	0 0000
X[32]-X[35]	1111 1101 0000 0000 1010 000	1000	0 0000
X[36]-X[39]	1111 1101 0000 0000 1010 000	1001	0 0000
X[40]-X[43]	1111 1101 0000 0000 1010 000	1010	0 0000
X[44]-X[47]	1111 1101 0000 0000 1010 000	1011	0 0000
X[58]-X[51]	1111 1101 0000 0000 1010 000	1100	0 0000
X[52]-X[55]	1111 1101 0000 0000 1010 000	1101	0 0000
X[56]-X[59]	1111 1101 0000 0000 1010 000	1110	0 0000
X[60]-X[63]	1111 1101 0000 0000 1010 000	1111	0 0000

Παρόμοια ισχύουν και για τους πίνακες Z και Y.

b)

Δημιουργούμε τον παρακάτω πίνακα για να αποφανθούμε για τον αριθμό hits και misses για όλη την εκτέλεση του παραπάνω κώδικα:

i	i	i	X[i]	X[i]	X[i]	Y[i]	Y[i]	Y[i]	Z[i]	Z[i]	Z[i]
0	22	44	X[0]:M	X[22]:M	X[44]:M	Y[0]:M	Y[22]:M	Y[44]:M	Z[0]:M	Z[22]:M	Z[44]:M
1	23	45	X[1]:M	X[23]:M	X[45]:M	Y[1]:M	Y[23]:M	Y[45]:M	Z[1]:M	Z[23]:M	Z[45]:M
2	24	46	X[2]:M	X[24]:M	X[46]:M	Y[2]:M	Y[24]:M	Y[46]:M	Z[2]:M	Z[24]:M	Z[46]:M
3	25	47	X[3]:M	X[25]:M	X[47]:M	Y[3]:M	Y[25]:M	Y[47]:M	Z[3]:M	Z[25]:M	Z[47]:M
4	26	48	X[4]:M	X[26]:M	X[48]:M	Y[4]:M	Y[26]:M	Y[48]:M	Z[4]:M	Z[26]:M	Z[48]:M
5	27	49	X[5]:M	X[27]:M	X[49]:M	Y[5]:M	Y[27]:M	Y[49]:M	Z[5]:M	Z[27]:M	Z[49]:M
6	28	50	X[6]:M	X[28]:M	X[50]:M	Y[6]:M	Y[28]:M	Y[50]:M	Z[6]:M	Z[28]:M	Z[50]:M
7	29	51	X[7]:M	X[29]:M	X[51]:M	Y[7]:M	Y[29]:M	Y[51]:M	Z[7]:M	Z[29]:M	Z[51]:M
8	30	52	X[8]:M	X[30]:M	X[52]:M	Y[8]:M	Y[30]:M	Y[52]:M	Z[8]:M	Z[30]:M	Z[52]:M
9	31	53	X[9]:M	X[31]:M	X[53]:M	Y[9]:M	Y[31]:M	Y[53]:M	Z[9]:M	Z[31]:M	Z[53]:M
10	32	54	X[10]:M	X[32]:M	X[54]:M	Y[10]:M	Y[32]:M	Y[54]:M	Z[10]:M	Z[32]:M	Z[54]:M
11	33	55	X[11]:M	X[33]:M	X[55]:M	Y[11]:M	Y[33]:M	Y[55]:M	Z[11]:M	Z[33]:M	Z[55]:M
12	34	56	X[12]:M	X[34]:M	X[56]:M	Y[12]:M	Y[34]:M	Y[56]:M	Z[12]:M	Z[34]:M	Z[56]:M
13	35	57	X[13]:M	X[35]:M	X[57]:M	Y[13]:M	Y[35]:M	Y[57]:M	Z[13]:M	Z[35]:M	Z[57]:M
14	36	58	X[14]:M	X[36]:M	X[58]:M	Y[14]:M	Y[36]:M	Y[58]:M	Z[14]:M	Z[36]:M	Z[58]:M
15	37	59	X[15]:M	X[37]:M	X[59]:M	Y[15]:M	Y[37]:M	Y[59]:M	Z[15]:M	Z[37]:M	Z[59]:M
16	38	60	X[16]:M	X[38]:M	X[60]:M	Y[16]:M	Y[38]:M	Y[60]:M	Z[16]:M	Z[38]:M	Z[60]:M
17	39	61	X[17]:M	X[39]:M	X[61]:M	Y[17]:M	Y[39]:M	Y[61]:M	Z[17]:M	Z[39]:M	Z[61]:M
18	40	62	X[18]:M	X[40]:M	X[62]:M	Y[18]:M	Y[40]:M	Y[62]:M	Z[18]:M	Z[40]:M	Z[62]:M
19	41	63	X[19]:M	X[41]:M	X[63]:M	Y[19]:M	Y[41]:M	Y[63]:M	Z[19]:M	Z[41]:M	Z[63]:M
20	42		X[20]:M	X[42]:M		Y[20]:M	Y[42]:M		Z[20]:M	Z[42]:M	
21	43		X[21]:M	X[43]:M		Y[21]:M	Y[43]:M		Z[21]:M	Z[43]:M	

Τα πρώτα στοιχεία των εκάστοτε πινάκων εμφανίζουν compulsory miss, γιατί τότε ξεκινά η διαδικασία.

Επειδή όλοι οι πίνακες ξεκινούν την αποθήκευση του $X[0], Y[0], Z[0]$ σε INDEX 0000, τότε για όλες τις τιμές των $X[i], Y[i], Z[i]$ $i = 0, 1, \dots, 63$, τότε θα έχουμε μόνο κατάσταση miss. Αυτό συμβαίνει διότι η έναρξη όλων των πινάκων σε INDEX 0000 θα φέρει σε ανταγωνισμό όλους τους εμπλεκόμενους πίνακες για τις θέσεις της μνήμης cache.

Misses = $3 \times (63 + 1) = 192$ και Hit = 0.

Οπότε, miss rate = 100% και hit rate = 0%.

c)

Αντικαθιστούμε τη μνήμη με μία συσχέτισης δύο δρόμων, έχοντας ίδιο συνολικό αριθμό και μέγεθος block δεδομένων.

Offset:

32 bytes block size = $2^5 \rightarrow 5$ bits

Index:

blocks (2 ανά γραμμή) = χωρητικότητα / ($2 \times \text{bytes} \times (1\text{-way})$) ή

$512 / [2 \times 32] = 8$ ομάδες = $2^3 \rightarrow 3$ bits

Οπότε, Tag = $32 - 3 - 5 = 24$ bits

TAG	INDEX	OFFSET
24	3	5

$X \rightarrow 0xFD00A000 \rightarrow 1111\ 1101\ 0000\ 0000\ 1010\ 0000\ 0000\ 0000$

$Y \rightarrow 0xA080C000 \rightarrow 1010\ 0000\ 1000\ 0000\ 1100\ 0000\ 0000\ 0000$

$Z \rightarrow 0xF1001000 \rightarrow 1111\ 0001\ 0000\ 0000\ 0001\ 0000\ 0000\ 0000$

Δημιουργούμε το ανάλογο διάγραμμα που φαίνεται παρακάτω:

IND EX	X	X	Y	Y	Z	Z
000	X[0]/X[1]/X[2]]/X[3]	X[32]/X[33]/ X[34]/X[35]	X[0]/X[1]/X[2]]/X[3]	X[32]/X[33]/ X[34]/X[35]	X[0]/X[1]/X[2]]/X[3]	X[32]/X[33]/ X[34]/X[35]
001	X[4]/X[5]/X[6]]/X[7]	X[36]/X[37]/ X[38]/X[39]	X[4]/X[5]/X[6]]/X[7]	X[36]/X[37]/ X[38]/X[39]	X[4]/X[5]/X[6]]/X[7]	X[36]/X[37]/ X[38]/X[39]
010	X[8]/X[9]/X[10]]/X[11]	X[40]/X[41]/ X[42]/X[43]	X[8]/X[9]/X[10]]/X[11]	X[40]/X[41]/ X[42]/X[43]	X[8]/X[9]/X[10]]/X[11]	X[40]/X[41]/ X[42]/X[43]
011	X[12]/X[13]/ X[14]/X[15]	X[44]/X[45]/ X[46]/X[47]	X[12]/X[13]/ X[14]/X[15]	X[44]/X[45]/ X[46]/X[47]	X[12]/X[13]/ X[14]/X[15]	X[44]/X[45]/ X[46]/X[47]
100	X[16]/X[17]/ X[18]/X[19]	X[48]/X[49]/ X[50]/X[51]	X[16]/X[17]/ X[18]/X[19]	X[48]/X[49]/ X[50]/X[51]	X[16]/X[17]/ X[18]/X[19]	X[48]/X[49]/ X[50]/X[51]
101	X[20]/X[21]/ X[22]/X[23]	X[52]/X[53]/ X[54]/X[55]	X[20]/X[21]/ X[22]/X[23]	X[52]/X[53]/ X[54]/X[55]	X[20]/X[21]/ X[22]/X[23]	X[52]/X[53]/ X[54]/X[55]
110	X[24]/X[25]/ X[26]/X[27]	X[56]/X[57]/ X[58]/X[59]	X[24]/X[25]/ X[26]/X[27]	X[56]/X[57]/ X[58]/X[59]	X[24]/X[25]/ X[26]/X[27]	X[56]/X[57]/ X[58]/X[59]
111	X[28]/X[29]/ X[30]/X[31]	X[60]/X[61]/ X[62]/X[63]	X[28]/X[29]/ X[30]/X[31]	X[60]/X[61]/ X[62]/X[63]	X[28]/X[29]/ X[30]/X[31]	X[60]/X[61]/ X[62]/X[63]

Δημιουργούμε τον παρακάτω πίνακα, όμοια με πριν:

Επειδή επαναλαμβάνεται το ίδιο μοτίβο κάθε φορά, θα εκτελέσουμε τον αλγόριθμο μέχρι ένα σημείο και μετά θα υπολογίσουμε τα hit και misses για τον συνολικό αλγόριθμο.

i	X	Y	Z
0	X[0](000) M	Y[0](000) M	Z[0](000) M
1	X[1](000) M (R(Y[0]))	Y[1](000) M (R(Z[0]))	Z[1](000) M (R(X[1]))
2	X[2](000) H	Y[2](000) H	Z[2](000) H
3	X[3](000) H	Y[3](000) H	Z[3](000) H
4	X[4](001) M	Y[4](001) M	Z[4](001) M
5	X[5](001) M (R(Y[4]))	Y[5](001) M (R(Z[4]))	Z[5](001) M (R(X[5]))
6	X[6](001) H	Y[6](001) H	Z[6](001) H
7	X[7](001) H	Y[7](001) H	Z[7](001) H
8	X[8](010) M	Y[8](010) M	Z[8](010) M
9	X[9](010) M (R(Y[8]))	Y[9](010) M (R(Z[8]))	Z[9](010) M (R(X[9]))
10	X[10](010) H	Y[10](010) H	Z[10](010) H
11	X[11](010) H	Y[11](010) H	Z[11](010) H

12	X[12](011) M	Y[12](011) M	Z[12](011) M
13	X[13](011) M (R(Y[12]))	Y[13](011) M (R(Z[12]))	Z[13](011) M (R(X[13]))
14	X[14](011) H	Y[14](011) H	Z[14](011) H
15	X[15](011) H	Y[15](011) H	Z[15](011) H

Παρατηρούμε ότι, ανά 2 γραμμές, υπάρχουν hits και τα υπόλοιπα είναι misses.

Οπότε, για όλον τον κώδικα, υπάρχουν ίσο αριθμό hits και misses.

$$\text{Misses} = \frac{3 \times (63+1)}{2} = \text{Hit} = 96.$$

Οπότε, miss rate = 50% και hit rate = 50%.