
Caches

Γεώργιος Κυριακόπουλος – el18153

1. Για το μέσο χρόνο πρόσβασης της μνήμης L1 έχουμε:

$$AMAT_{L1} = 3.5 = 1 \cdot hit_{L1} + MR_{L1}MP_{L1} \Rightarrow$$

$$3.5 = 1 + 0.02MP_{L1} \Rightarrow$$

$$0.02MP_{L1} = 2.5 \Rightarrow$$

$$MP_{L1} = 125 \text{ κύκλοι}$$

Για το speedup λόγω του δευτέρου επιπέδου κρυφής μνήμης έχουμε:

$$\frac{AMAT_{L1}}{AMAT_{L2}} \geq 2.5 \Rightarrow$$

$$\frac{3.5}{1 \cdot hit_{L1} + MR_{L1}(1 \cdot hit_{L2} + MR_{L2}MP_{L2})} \geq 2.5 \Rightarrow$$

$$1.4 \geq 1 + 0.02hit_{L2} + 0.28 \Rightarrow$$

$$0.02hit_{L2} \leq 0.12 \Rightarrow$$

$$hit_{L2} \leq 6$$

Επομένως το μέγιστο κόστος πρόσβασης του δευτέρου επιπέδου είναι 6 κύκλοι.

2. Α) Έχουμε μία 2-way associative cache, με χρήση LRU ως κανόνα επίλυσης των conflicts, η οποία είναι write-allocate, επομένως, σε περίπτωση write-miss, φορτώνει και στην cache τα δεδομένα και γράφει σε αυτή, μεγέθους 512B. Έχουμε, επίσης μέγεθος block 32B και στοιχεία πίνακα double μεγέθους 8B.

Ως αποτέλεσμα, έχουμε $512/32=16$ blocks και αφού έχουμε 2-way associative οργάνωση, 8+8 blocks χωρισμένα σε 2 set. Επίσης το κάθε block μας χωράει $32/8=4$ στοιχεία του πίνακα.

Επίσης ακολουθούμε για παράδειγμα την παρακάτω μέθοδο αποθήκευσης σύμφωνα και με την εκφώνηση (λόγω της συνεχόμενης δήλωση, τα στοιχεία 80-157 πάνε στην θέση του αριθμού τους -80 και ομοίως ότι ισχύει για τα στοιχεία του a ισχύουν και για του b).

Σημείωση: για ευκολία καταγραφής, όπου $a[i][j]=a_{ij}$ και με Bold είναι το LRU στο κάθε set.

0	a00	a01	a02	a03	a40	a41	a42	a43
1	a04	a05	a06	a07	a44	a45	a46	a47
2	a10	a11	a12	a13	a50	a51	a52	a53
3	a14	a15	a16	a17	a54	a55	a56	a57
4	a20	a21	a22	a23	a60	a61	a62	a63
5	a24	a25	a26	a27	a64	a65	a66	a67
6	a30	a31	a32	a33	a70	a71	a72	a73
7	a34	a35	a36	a37	a74	a75	a76	a77

Η τελική απάντηση για τον αριθμό των hits και misses βρίσκεται στη σελίδα 18.

Για $i=0, j=0$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	-	-	-	-	-	-	-	-
1	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-

Read a40 – Compulsory Miss

Read a00 – Compulsory Miss

Read b00 – Compulsory Miss

Write a40 – Conflict Miss

0	a40	a41	a42	a43	b00	b01	b02	b03
1	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-

Για $i=0, j=1$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	a40	a41	a42	a43	b00	b01	b02	b03
1	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-

Read a41 – Hit

Read a01 – Conflict Miss

Read b01 – Conflict Miss

Write a41 – Conflict Miss

0	a40	a41	a42	a43	b00	b01	b02	b03
1	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-

Όμοια με το $i=0, j=1$ είναι και τα $i=0, j=2$ και $i=0, j=3$ και η cache παραμένει σε αυτήν την κατάσταση για το index 0 μέχρι να πάμε στο $i=4$.

Για **i=0, j=4**: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	a40	a41	a42	a43	b00	b01	b02	b03
1	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-

Read a44 – Compulsory Miss

Read a04 – Compulsory Miss

Read b04 – Compulsory Miss

Write a44 – Conflict Miss

0	a40	a41	a42	a43	b00	b01	b02	b03
1	a44	a45	a46	a47	b04	b05	b06	b07
2	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-

Για $i=0, j=5$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	a40	a41	a42	a43	b00	b01	b02	b03
1	a44	a45	a46	a47	b04	b05	b06	b07
2	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-

Read a45 – Hit

Read a05 – Conflict Miss

Read b05 – Conflict Miss

Write a45 – Conflict Miss

0	a40	a41	a42	a43	b00	b01	b02	b03
1	a44	a45	a46	a47	b04	b05	b06	b07
2	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-

Όμοια με το $i=0, j=5$ είναι και τα $i=0, j=6$ και $i=0, j=7$ και η cache παραμένει σε αυτήν την κατάσταση για το index 1 μέχρι να πάμε στο $i=4$. Άρα για $i=0$ έχουμε συνολικά 6 hit και 26 misses = 32 accesses ($8*4$).

Για $i=1, j=0$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	a40	a41	a42	a43	b00	b01	b02	b03
1	a44	a45	a46	a47	b04	b05	b06	b07
2	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-

Read a50 – Compulsory Miss

Read a10 – Compulsory Miss

Read b10 – Compulsory Miss

Write a50 – Conflict Miss

0	a40	a41	a42	a43	b00	b01	b02	b03
1	a44	a45	a46	a47	b04	b05	b06	b07
2	a50	a51	a52	a53	b10	b11	b12	b13
3	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-

Για $i=1, j=1$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	a40	a41	a42	a43	b00	b01	b02	b03
1	a44	a45	a46	a47	b04	b05	b06	b07
2	a50	a51	a52	a53	b10	b11	b12	b13
3	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-

Read a51 – Hit

Read a11 – Conflict Miss

Read b11 – Conflict Miss

Write a51 – Conflict Miss

0	a40	a41	a42	a43	b00	b01	b02	b03
1	a44	a45	a46	a47	b04	b05	b06	b07
2	a50	a51	a52	a53	b10	b11	b12	b13
3	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-

Όμοια με το $i=1, j=1$ είναι και τα $i=1, j=2$ και $i=1, j=3$ και η cache παραμένει σε αυτήν την κατάσταση για το index 2 μέχρι να πάμε στο $i=4$.

Για $i=1, j=4$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	a40	a41	a42	a43	b00	b01	b02	b03
1	a44	a45	a46	a47	b04	b05	b06	b07
2	a50	a51	a52	a53	b10	b11	b12	b13
3	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-

Read a54 – Compulsory Miss

Read a14 – Compulsory Miss

Read b14 – Compulsory Miss

Write a54 – Conflict Miss

0	a40	a41	a42	a43	b00	b01	b02	b03
1	a44	a45	a46	a47	b04	b05	b06	b07
2	a50	a51	a52	a53	b10	b11	b12	b13
3	a54	a55	a56	a57	b14	b15	b16	b17
4	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-

Για $i=1, j=5$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	a40	a41	a42	a43	b00	b01	b02	b03
1	a44	a45	a46	a47	b04	b05	b06	b07
2	a50	a51	a52	a53	b10	b11	b12	b13
3	a54	a55	a56	a57	b14	b15	b16	b17
4	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-

Read a55 – Hit

Read a15 – Conflict Miss

Read b15 – Conflict Miss

Write a55 – Conflict Miss

0	a40	a41	a42	a43	b00	b01	b02	b03
1	a44	a45	a46	a47	b04	b05	b06	b07
2	a50	a51	a52	a53	b10	b11	b12	b13
3	a54	a55	a56	a57	b14	b15	b16	b17
4	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-

Όμοια με το $i=1, j=5$ είναι και τα $i=1, j=6$ και $i=1, j=7$ και η cache παραμένει σε αυτήν την κατάσταση για το index 3 μέχρι να πάμε στο $i=4$. Άρα για $i=1$ έχουμε συνολικά 6 hit και 26 misses = 32 accesses ($8 \cdot 4$). Όμοια για $i=2, i=3$ θα έχουμε 6 hit και 26 misses για το καθένα. Επομένως για $i=0-3$ έχουμε 24 hit και 104 misses = 128 accesses ($8 \cdot 4 \cdot 4$).

Για $i=4, j=0$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	a40	a41	a42	a43	b00	b01	b02	b03
1	a44	a45	a46	a47	b04	b05	b06	b07
2	a50	a51	a52	a53	b10	b11	b12	b13
3	a54	a55	a56	a57	b14	b15	b16	b17
4	a60	a61	a62	a63	b20	b21	b22	b23
5	a64	a65	a66	a67	b24	b25	b26	b27
6	a70	a71	a72	a73	b30	b31	b32	b33
7	a74	a75	a76	a77	b34	b35	b36	b37

Read a40 – Hit

Read a50 – Hit

Read b40 – Compulsory Miss

Write a40 – Hit

0	a40	a41	a42	a43	b40	b41	b42	b43
1	a44	a45	a46	a47	b04	b05	b06	b07
2	a50	a51	a52	a53	b10	b11	b12	b13
3	a54	a55	a56	a57	b14	b15	b16	b17
4	a60	a61	a62	a63	b20	b21	b22	b23
5	a64	a65	a66	a67	b24	b25	b26	b27
6	a70	a71	a72	a73	b30	b31	b32	b33
7	a74	a75	a76	a77	b34	b35	b36	b37

Για $i=4, j=1$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	a40	a41	a42	a43	b40	b41	b42	b43
1	a44	a45	a46	a47	b04	b05	b06	b07
2	a50	a51	a52	a53	b10	b11	b12	b13
3	a54	a55	a56	a57	b14	b15	b16	b17
4	a60	a61	a62	a63	b20	b21	b22	b23
5	a64	a65	a66	a67	b24	b25	b26	b27
6	a70	a71	a72	a73	b30	b31	b32	b33
7	a74	a75	a76	a77	b34	b35	b36	b37

Read a41 – Hit

Read a51 – Hit

Read b41 – Hit

Write a41 – Hit

0	a40	a41	a42	a43	b40	b41	b42	b43
1	a44	a45	a46	a47	b04	b05	b06	b07
2	a50	a51	a52	a53	b10	b11	b12	b13
3	a54	a55	a56	a57	b14	b15	b16	b17
4	a60	a61	a62	a63	b20	b21	b22	b23
5	a64	a65	a66	a67	b24	b25	b26	b27
6	a70	a71	a72	a73	b30	b31	b32	b33
7	a74	a75	a76	a77	b34	b35	b36	b37

Όμοια με το $i=4, j=1$ είναι και τα $i=4, j=2$ και $i=4, j=3$ και η cache παραμένει σε αυτήν την κατάσταση.

Για $i=4, j=4$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	a40	a41	a42	a43	b40	b41	b42	b43
1	a44	a45	a46	a47	b04	b05	b06	b07
2	a50	a51	a52	a53	b10	b11	b12	b13
3	a54	a55	a56	a57	b14	b15	b16	b17
4	a60	a61	a62	a63	b20	b21	b22	b23
5	a64	a65	a66	a67	b24	b25	b26	b27
6	a70	a71	a72	a73	b30	b31	b32	b33
7	a74	a75	a76	a77	b34	b35	b36	b37

Read a44 – Hit

Read a54 – Hit

Read b44 – Compulsory Miss

Write a44 – Hit

0	a40	a41	a42	a43	b40	b41	b42	b43
1	a44	a45	a46	a47	b44	b45	b46	b47
2	a50	a51	a52	a53	b10	b11	b12	b13
3	a54	a55	a56	a57	b14	b15	b16	b17
4	a60	a61	a62	a63	b20	b21	b22	b23
5	a64	a65	a66	a67	b24	b25	b26	b27
6	a70	a71	a72	a73	b30	b31	b32	b33
7	a74	a75	a76	a77	b34	b35	b36	b37

Για $i=4, j=5$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	a40	a41	a42	a43	b40	b41	b42	b43
1	a44	a45	a46	a47	b44	b45	b46	b47
2	a50	a51	a52	a53	b10	b11	b12	b13
3	a54	a55	a56	a57	b14	b15	b16	b17
4	a60	a61	a62	a63	b20	b21	b22	b23
5	a64	a65	a66	a67	b24	b25	b26	b27
6	a70	a71	a72	a73	b30	b31	b32	b33
7	a74	a75	a76	a77	b34	b35	b36	b37

Read a45 – Hit

Read a55 – Hit

Read b55 – Hit

Write a45 – Hit

0	a40	a41	a42	a43	b40	b41	b42	b43
1	a44	a45	a46	a47	b44	b45	b46	b47
2	a50	a51	a52	a53	b10	b11	b12	b13
3	a54	a55	a56	a57	b14	b15	b16	b17
4	a60	a61	a62	a63	b20	b21	b22	b23
5	a64	a65	a66	a67	b24	b25	b26	b27
6	a70	a71	a72	a73	b30	b31	b32	b33
7	a74	a75	a76	a77	b34	b35	b36	b37

Όμοια με το $i=4, j=5$ είναι και τα $i=4, j=6$ και $i=4, j=7$ και η cache παραμένει σε αυτήν την κατάσταση. Άρα για $i=4$ έχουμε συνολικά 30 hit και 2 misses = 32 accesses ($8*4$).

Όμοια για $i=5, i=6$ θα έχουμε 30 hit και 2 misses για το καθένα. Επομένως για $i=4-6$ έχουμε 90 hit και 6 misses = 96 accesses ($8*4*3$). Σύνολο τώρα, 114 hit και 110 misses.

Για $i=7, j=0$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	a40	a41	a42	a43	b40	b41	b42	b43
1	a44	a45	a46	a47	b44	b45	b46	b47
2	a50	a51	a52	a53	b50	b51	b52	b53
3	a54	a55	a56	a57	b54	b55	b56	b57
4	a60	a61	a62	a63	b60	b61	b62	b63
5	a64	a65	a66	a67	b64	b65	b66	b67
6	a70	a71	a72	a73	b30	b31	b32	b33
7	a74	a75	a76	a77	b34	b35	b36	b37

Read a70 – Hit

Read a80 – Compulsory Miss

Read b70 – Compulsory Miss

Write a70 – Hit

0	a80	a81	a82	a83	a40	a41	a42	a43
1	a44	a45	a46	a47	b04	b05	b06	b07
2	a50	a51	a52	a53	b10	b11	b12	b13
3	a54	a55	a56	a57	b14	b15	b16	b17
4	a60	a61	a62	a63	b20	b21	b22	b23
5	a64	a65	a66	a67	b24	b25	b26	b27
6	a70	a71	a72	a73	b70	b71	b72	b73
7	a74	a75	a76	a77	b34	b35	b36	b37

Για $i=7, j=1$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	a80	a81	a82	a83	a40	a41	a42	a43
1	a44	a45	a46	a47	b04	b05	b06	b07
2	a50	a51	a52	a53	b10	b11	b12	b13
3	a54	a55	a56	a57	b14	b15	b16	b17
4	a60	a61	a62	a63	b20	b21	b22	b23
5	a64	a65	a66	a67	b24	b25	b26	b27
6	a70	a71	a72	a73	b70	b71	b72	b73
7	a74	a75	a76	a77	b34	b35	b36	b37

Read a71 – Hit

Read a81 – Hit

Read b71 – Hit

Write a71 – Hit

0	a80	a81	a82	a83	a40	a41	a42	a43
1	a44	a45	a46	a47	b04	b05	b06	b07
2	a50	a51	a52	a53	b10	b11	b12	b13
3	a54	a55	a56	a57	b14	b15	b16	b17
4	a60	a61	a62	a63	b20	b21	b22	b23
5	a64	a65	a66	a67	b24	b25	b26	b27
6	a70	a71	a72	a73	b70	b71	b72	b73
7	a74	a75	a76	a77	b34	b35	b36	b37

Όμοια με το $i=7, j=1$ είναι και τα $i=7, j=2$ και $i=7, j=3$ και η cache παραμένει σε αυτήν την κατάσταση.

Για $i=7, j=4$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	a80	a81	a82	a83	a40	a41	a42	a43
1	a44	a45	a46	a47	b04	b05	b06	b07
2	a50	a51	a52	a53	b10	b11	b12	b13
3	a54	a55	a56	a57	b14	b15	b16	b17
4	a60	a61	a62	a63	b20	b21	b22	b23
5	a64	a65	a66	a67	b24	b25	b26	b27
6	a70	a71	a72	a73	b70	b71	b72	b73
7	a74	a75	a76	a77	b34	b35	b36	b37

Read a74 – Hit

Read a84 – Compulsory Miss

Read b74 – Compulsory Miss

Write a74 – Hit

0	a80	a81	a82	a83	a40	a41	a42	a43
1	a84	a85	a86	a87	a44	a45	a46	a47
2	a50	a51	a52	a53	b10	b11	b12	b13
3	a54	a55	a56	a57	b14	b15	b16	b17
4	a60	a61	a62	a63	b20	b21	b22	b23
5	a64	a65	a66	a67	b24	b25	b26	b27
6	a70	a71	a72	a73	b70	b71	b72	b73
7	a74	a75	a76	a77	b74	b75	b76	b77

Για $i=7, j=5$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	a80	a81	a82	a83	a40	a41	a42	a43
1	a84	a85	a86	a87	a44	a45	a46	a47
2	a50	a51	a52	a53	b10	b11	b12	b13
3	a54	a55	a56	a57	b14	b15	b16	b17
4	a60	a61	a62	a63	b20	b21	b22	b23
5	a64	a65	a66	a67	b24	b25	b26	b27
6	a70	a71	a72	a73	b70	b71	b72	b73
7	a74	a75	a76	a77	b74	b75	b76	b77

Read a75 – Hit

Read a85 – Hit

Read b75 – Hit

Write a75 – Hit

0	a80	a81	a82	a83	a40	a41	a42	a43
1	a84	a85	a86	a87	a44	a45	a46	a47
2	a50	a51	a52	a53	b10	b11	b12	b13
3	a54	a55	a56	a57	b14	b15	b16	b17
4	a60	a61	a62	a63	b20	b21	b22	b23
5	a64	a65	a66	a67	b24	b25	b26	b27
6	a70	a71	a72	a73	b70	b71	b72	b73
7	a74	a75	a76	a77	b74	b75	b76	b77

Όμοια με το $i=7, j=5$ είναι και τα $i=7, j=6$ και $i=7, j=7$ και η cache παραμένει σε αυτήν την κατάσταση. Άρα για $i=7$ έχουμε συνολικά 28 hit και 4 misses = 32 accesses ($8*4$).

Τελικό σύνολο, δηλαδή, 142 hit και 114 misses = 256 accesses ($8*8*4$).

B) Έχουμε μία 4-way associative cache, με χρήση LRU ως κανόνα επίλυσης των conflicts, η οποία είναι write-allocate, επομένως, σε περίπτωση write-miss, φορτώνει και στην cache τα δεδομένα και γράφει σε αυτή, μεγέθους 512B. Έχουμε, επίσης μέγεθος block 32B και στοιχεία πίνακα double μεγέθους 8B.

Ως αποτέλεσμα, έχουμε $512/32=16$ blocks και αφού έχουμε 4-way associative οργάνωση, $4+4+4+4$ blocks χωρισμένα σε 4 set. Επίσης το κάθε block μας χωράει $32/8=4$ στοιχεία του πίνακα.

Επίσης ακολουθούμε για παράδειγμα την παρακάτω μέθοδο αποθήκευσης σύμφωνα και με την εκφώνηση (λόγω της συνεχόμενης δήλωση, τα στοιχεία 80-157 πάνε στην θέση του αριθμού τους -80 και ομοίως ότι ισχύει για τα στοιχεία του a ισχύουν και για του b).

Σημείωση: για ευκολία καταγραφής, όπου $a[i][j]=a_{ij}$ και με Bold είναι το LRU στο κάθε set.

0	a00	a01	a02	a03	a20	a21	a22	a23	a40	a41	a42	a43	a60	a61	a62	a63
1	a04	a05	a06	a07	a24	a25	a26	a27	a44	a45	a46	a47	a64	a65	a66	a67
2	a10	a11	a12	a13	a30	a31	a32	a33	a50	a51	a52	a53	a70	a71	a72	a73
3	a14	a15	a16	a17	a34	a35	a36	a37	a54	a55	a56	a57	a74	a75	a76	a77

Η τελική απάντηση για τον αριθμό των hits και misses και τον εάν θα διαλέγαμε την αλλαγή σε 4-way associative cache βρίσκεται στη σελίδα 43.

Για $i=0, j=0$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

[illegible]

Read a40 – Compulsory Miss

Read a00 – Compulsory Miss

Read b00 – Compulsory Miss

Write a40 – Hit

[illegible]

Για $i=0, j=1$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	a00	a01	a02	a03	b00	b01	b02	b03	a40	a41	a42	a43	-	-	-	-
1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Read a41 – Hit

Read a01 – Hit

Read b01 – Hit

Write a41 – Hit

0	a00	a01	a02	a03	b00	b01	b02	b03	a40	a41	a42	a43	-	-	-	-
1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Όμοια με το $i=0, j=1$ είναι και τα $i=0, j=2$ και $i=0, j=3$ και η cache παραμένει σε αυτήν την κατάσταση.

Για $i=0, j=4$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

[illegible]

Read a44 – Compulsory Miss

Read a04 – Compulsory Miss

Read b04 – Compulsory Miss

Write a44 – Hit

[illegible]

Για $i=0, j=5$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	a00	a01	a02	a03	b00	b01	b02	b03	a40	a41	a42	a43	-	-	-	-
1	a04	a05	a06	a07	b04	b05	b06	b07	a44	a45	a46	a47	-	-	-	-
2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Read a45 – Hit

Read a05 – Hit

Read b05 – Hit

Write a45 – Hit

0	a00	a01	a02	a03	b00	b01	b02	b03	a40	a41	a42	a43	-	-	-	-
1	a04	a05	a06	a07	b04	b05	b06	b07	a44	a45	a46	a47	-	-	-	-
2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Όμοια με το $i=0, j=5$ είναι και τα $i=0, j=6$ και $i=0, j=7$ και η cache παραμένει σε αυτήν την κατάσταση. Άρα για $i=0$ έχουμε συνολικά 26 hit και 6 misses = 32 accesses ($8*4$).

Για $i=1, j=0$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

[illegible]

Read a50 – Compulsory Miss

Read a10 – Compulsory Miss

Read b10 – Compulsory Miss

Write a50 – Hit

[illegible]

Για $i=1, j=1$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	a00	a01	a02	a03	b00	b01	b02	b03	a40	a41	a42	a43	-	-	-	-
1	a04	a05	a06	a07	b04	b05	b06	b07	a44	a45	a46	a47	-	-	-	-
2	a10	a11	a12	a13	b10	b11	b12	b13	a50	a51	a52	a53	-	-	-	-
3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Read a51 – Hit

Read a11 – Hit

Read b11 – Hit

Write a51 – Hit

0	a00	a01	a02	a03	b00	b01	b02	b03	a40	a41	a42	a43	-	-	-	-
1	a04	a05	a06	a07	b04	b05	b06	b07	a44	a45	a46	a47	-	-	-	-
2	a10	a11	a12	a13	b10	b11	b12	b13	a50	a51	a52	a53	-	-	-	-
3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Όμοια με το $i=1, j=1$ είναι και τα $i=1, j=2$ και $i=1, j=3$ και η cache παραμένει σε αυτήν την κατάσταση.

Για $i=1, j=4$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	a00	a01	a02	a03	b00	b01	b02	b03	a40	a41	a42	a43	-	-	-	-
1	a04	a05	a06	a07	b04	b05	b06	b07	a44	a45	a46	a47	-	-	-	-
2	a10	a11	a12	a13	b10	b11	b12	b13	a50	a51	a52	a53	-	-	-	-
3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Read a54 – Compulsory Miss

Read a14 – Compulsory Miss

Read b14 – Compulsory Miss

Write a54 – Hit

0	a00	a01	a02	a03	b00	b01	b02	b03	a40	a41	a42	a43	-	-	-	-
1	a04	a05	a06	a07	b04	b05	b06	b07	a44	a45	a46	a47	-	-	-	-
2	a10	a11	a12	a13	b10	b11	b12	b13	a50	a51	a52	a53	-	-	-	-
3	a14	a15	a16	a17	b14	b15	b16	b17	a54	a55	a56	a57	-	-	-	-

Για $i=1, j=5$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	a00	a01	a02	a03	b00	b01	b02	b03	a40	a41	a42	a43	-	-	-	-
1	a04	a05	a06	a07	b04	b05	b06	b07	a44	a45	a46	a47	-	-	-	-
2	a10	a11	a12	a13	b10	b11	b12	b13	a50	a51	a52	a53	-	-	-	-
3	a14	a15	a16	a17	b14	b15	b16	b17	a54	a55	a56	a57	-	-	-	-

Read a55 – Hit

Read a15 – Hit

Read b15 – Hit

Write a55 – Hit

0	a00	a01	a02	a03	b00	b01	b02	b03	a40	a41	a42	a43	-	-	-	-
1	a04	a05	a06	a07	b04	b05	b06	b07	a44	a45	a46	a47	-	-	-	-
2	a10	a11	a12	a13	b10	b11	b12	b13	a50	a51	a52	a53	-	-	-	-
3	a14	a15	a16	a17	b14	b15	b16	b17	a54	a55	a56	a57	-	-	-	-

Όμοια με το $i=1, j=5$ είναι και τα $i=1, j=6$ και $i=1, j=7$ και η cache παραμένει σε αυτήν την κατάσταση. Άρα για $i=1$ έχουμε συνολικά 26 hit και 6 misses = 32 accesses ($8 \cdot 4$).

Για $i=2, j=0$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	a00	a01	a02	a03	b00	b01	b02	b03	a40	a41	a42	a43	-	-	-	-
1	a04	a05	a06	a07	b04	b05	b06	b07	a44	a45	a46	a47	-	-	-	-
2	a10	a11	a12	a13	b10	b11	b12	b13	a50	a51	a52	a53	-	-	-	-
3	a14	a15	a16	a17	b14	b15	b16	b17	a54	a55	a56	a57	-	-	-	-

Read a60 – Compulsory Miss

Read a20 – Compulsory Miss

Read b20 – Compulsory Miss

Write a60 – Hit

0	a20	a21	a22	a23	b20	b21	b22	b23	a40	a41	a42	a43	a60	a61	a62	a63
1	a04	a05	a06	a07	b04	b05	b06	b07	a44	a45	a46	a47	-	-	-	-
2	a10	a11	a12	a13	b10	b11	b12	b13	a50	a51	a52	a53	-	-	-	-
3	a14	a15	a16	a17	b14	b15	b16	b17	a54	a55	a56	a57	-	-	-	-

Για $i=2, j=1$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	a20	a21	a22	a23	b20	b21	b22	b23	a40	a41	a42	a43	a60	a61	a62	a63
1	a04	a05	a06	a07	b04	b05	b06	b07	a44	a45	a46	a47	-	-	-	-
2	a10	a11	a12	a13	b10	b11	b12	b13	a50	a51	a52	a53	-	-	-	-
3	a14	a15	a16	a17	b14	b15	b16	b17	a54	a55	a56	a57	-	-	-	-

Read a61 – Hit

Read a21 – Hit

Read b21 – Hit

Write a61 – Hit

0	a20	a21	a22	a23	b20	b21	b22	b23	a40	a41	a42	a43	a60	a61	a62	a63
1	a04	a05	a06	a07	b04	b05	b06	b07	a44	a45	a46	a47	-	-	-	-
2	a10	a11	a12	a13	b10	b11	b12	b13	a50	a51	a52	a53	-	-	-	-
3	a14	a15	a16	a17	b14	b15	b16	b17	a54	a55	a56	a57	-	-	-	-

Όμοια με το $i=2, j=1$ είναι και τα $i=2, j=2$ και $i=2, j=3$ και η cache παραμένει σε αυτήν την κατάσταση.

Για $i=2, j=4$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	a20	a21	a22	a23	b20	b21	b22	b23	a40	a41	a42	a43	a60	a61	a62	a63
1	a04	a05	a06	a07	b04	b05	b06	b07	a44	a45	a46	a47	-	-	-	-
2	a10	a11	a12	a13	b10	b11	b12	b13	a50	a51	a52	a53	-	-	-	-
3	a14	a15	a16	a17	b14	b15	b16	b17	a54	a55	a56	a57	-	-	-	-

Read a64 – Compulsory Miss

Read a24 – Compulsory Miss

Read b24 – Compulsory Miss

Write a64 – Hit

0	a20	a21	a22	a23	b20	b21	b22	b23	a40	a41	a42	a43	a60	a61	a62	a63
1	a24	a25	a26	a27	b24	b25	b26	b27	a44	a45	a46	a47	a64	a65	a66	a67
2	a10	a11	a12	a13	b10	b11	b12	b13	a50	a51	a52	a53	-	-	-	-
3	a14	a15	a16	a17	b14	b15	b16	b17	a54	a55	a56	a57	-	-	-	-

Για $i=2, j=5$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	a20	a21	a22	a23	b20	b21	b22	b23	a40	a41	a42	a43	a60	a61	a62	a63
1	a24	a25	a26	a27	b24	b25	b26	b27	a44	a45	a46	a47	a64	a65	a66	a67
2	a10	a11	a12	a13	b10	b11	b12	b13	a50	a51	a52	a53	-	-	-	-
3	a14	a15	a16	a17	b14	b15	b16	b17	a54	a55	a56	a57	-	-	-	-

Read a65 – Hit

Read a25 – Hit

Read b25 – Hit

Write a65 – Hit

0	a20	a21	a22	a23	b20	b21	b22	b23	a40	a41	a42	a43	a60	a61	a62	a63
1	a24	a25	a26	a27	b24	b25	b26	b27	a44	a45	a46	a47	a64	a65	a66	a67
2	a10	a11	a12	a13	b10	b11	b12	b13	a50	a51	a52	a53	-	-	-	-
3	a14	a15	a16	a17	b14	b15	b16	b17	a54	a55	a56	a57	-	-	-	-

Όμοια με το $i=2, j=5$ είναι και τα $i=2, j=6$ και $i=2, j=7$ και η cache παραμένει σε αυτήν την κατάσταση. Άρα για $i=2$ έχουμε συνολικά 26 hit και 6 misses = 32 accesses ($8*4$). Ομοίως για $i=3$ έχουμε 26 hit και 6 misses = 32 accesses ($8*4$). Άρα, μέχρι τώρα έχουμε 104 hits και 24 misses = 128 accesses ($8*4*4$).

Για $i=4, j=0$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	a20	a21	a22	a23	b20	b21	b22	b23	a40	a41	a42	a43	a60	a61	a62	a63
1	a24	a25	a26	a27	b24	b25	b26	b27	a44	a45	a46	a47	a64	a65	a66	a67
2	a30	a31	a32	a33	b30	b31	b32	b33	a50	a51	a52	a53	a70	a71	a72	a73
3	a34	a35	a36	a37	b34	b35	b36	b37	a54	a55	a56	a57	a74	a75	a76	a77

Read a40 – Hit

Read a50 – Hit

Read b40 – Compulsory Miss

Write a40 – Hit

0	b20	b21	b22	b23	b40	b41	b42	b43	a40	a41	a42	a43	a60	a61	a62	a63
1	a24	a25	a26	a27	b24	b25	b26	b27	a44	a45	a46	a47	a64	a65	a66	a67
2	a30	a31	a32	a33	b30	b31	b32	b33	a50	a51	a52	a53	a70	a71	a72	a73
3	a34	a35	a36	a37	b34	b35	b36	b37	a54	a55	a56	a57	a74	a75	a76	a77

Για $i=4, j=1$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	b20	b21	b22	b23	b40	b41	b42	b43	a40	a41	a42	a43	a60	a61	a62	a63
1	a24	a25	a26	a27	b24	b25	b26	b27	a44	a45	a46	a47	a64	a65	a66	a67
2	a30	a31	a32	a33	b30	b31	b32	b33	a50	a51	a52	a53	a70	a71	a72	a73
3	a34	a35	a36	a37	b34	b35	b36	b37	a54	a55	a56	a57	a74	a75	a76	a77

Read a41 – Hit

Read a51 – Hit

Read b41 – Hit

Write a41 – Hit

0	b20	b21	b22	b23	b40	b41	b42	b43	a40	a41	a42	a43	a60	a61	a62	a63
1	a24	a25	a26	a27	b24	b25	b26	b27	a44	a45	a46	a47	a64	a65	a66	a67
2	a30	a31	a32	a33	b30	b31	b32	b33	a50	a51	a52	a53	a70	a71	a72	a73
3	a34	a35	a36	a37	b34	b35	b36	b37	a54	a55	a56	a57	a74	a75	a76	a77

Όμοια με το $i=4, j=1$ είναι και τα $i=4, j=2$ και $i=4, j=3$ και η cache παραμένει σε αυτήν την κατάσταση.

Για $i=4, j=4$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	b20	b21	b22	b23	b40	b41	b42	b43	a40	a41	a42	a43	a60	a61	a62	a63
1	a24	a25	a26	a27	b24	b25	b26	b27	a44	a45	a46	a47	a64	a65	a66	a67
2	a30	a31	a32	a33	b30	b31	b32	b33	a50	a51	a52	a53	a70	a71	a72	a73
3	a34	a35	a36	a37	b34	b35	b36	b37	a54	a55	a56	a57	a74	a75	a76	a77

Read a44 – Hit

Read a54 – Hit

Read b44 – Compulsory Miss

Write a44 – Hit

0	b20	b21	b22	b23	b40	b41	b42	b43	a40	a41	a42	a43	a60	a61	a62	a63
1	b24	b25	b26	b27	b44	b45	b46	b47	a44	a45	a46	a47	a64	a65	a66	a67
2	a30	a31	a32	a33	b30	b31	b32	b33	a50	a51	a52	a53	a70	a71	a72	a73
3	a34	a35	a36	a37	b34	b35	b36	b37	a54	a55	a56	a57	a74	a75	a76	a77

Για $i=4, j=5$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	b20	b21	b22	b23	b40	b41	b42	b43	a40	a41	a42	a43	a60	a61	a62	a63
1	b24	b25	b26	b27	b44	b45	b46	b47	a44	a45	a46	a47	a64	a65	a66	a67
2	a30	a31	a32	a33	b30	b31	b32	b33	a50	a51	a52	a53	a70	a71	a72	a73
3	a34	a35	a36	a37	b34	b35	b36	b37	a54	a55	a56	a57	a74	a75	a76	a77

Read a45 – Hit

Read a55 – Hit

Read b45 – Hit

Write a45 – Hit

0	b20	b21	b22	b23	b40	b41	b42	b43	a40	a41	a42	a43	a60	a61	a62	a63
1	b24	b25	b26	b27	b44	b45	b46	b47	a44	a45	a46	a47	a64	a65	a66	a67
2	a30	a31	a32	a33	b30	b31	b32	b33	a50	a51	a52	a53	a70	a71	a72	a73
3	a34	a35	a36	a37	b34	b35	b36	b37	a54	a55	a56	a57	a74	a75	a76	a77

Όμοια με το $i=4, j=5$ είναι και τα $i=4, j=6$ και $i=4, j=7$ και η cache παραμένει σε αυτήν την κατάσταση. Άρα για $i=4$ έχουμε συνολικά 30 hit και 2 misses = 32 accesses ($8*4$). Ομοίως για $i=5$ έχουμε 30 hit και 2 misses = 32 accesses ($8*4$). Άρα, μέχρι τώρα έχουμε 164 hits και 28 misses = 196 accesses ($8*4*6$).

Για $i=6, j=0$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	b20	b21	b22	b23	b40	b41	b42	b43	a40	a41	a42	a43	a60	a61	a62	a63
1	b24	b25	b26	b27	b44	b45	b46	b47	a44	a45	a46	a47	a64	a65	a66	a67
2	b30	b31	b32	b33	b50	b51	b52	b53	a50	a51	a52	a53	a70	a71	a72	a73
3	b34	b35	b36	b37	b54	b55	b56	b57	a54	a55	a56	a57	a74	a75	a76	a77

Read a60 – Hit

Read a70 – Hit

Read b60 – Compulsory Miss

Write a60 – Hit

0	b60	b61	b62	b63	b40	b41	b42	b43	a40	a41	a42	a43	a60	a61	a62	a63
1	b24	b25	b26	b27	b44	b45	b46	b47	a44	a45	a46	a47	a64	a65	a66	a67
2	b30	b31	b32	b33	b50	b51	b52	b53	a50	a51	a52	a53	a70	a71	a72	a73
3	b34	b35	b36	b37	b54	b55	b56	b57	a54	a55	a56	a57	a74	a75	a76	a77

Για $i=6, j=1$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	b60	b61	b62	b63	b40	b41	b42	b43	a40	a41	a42	a43	a60	a61	a62	a63
1	b24	b25	b26	b27	b44	b45	b46	b47	a44	a45	a46	a47	a64	a65	a66	a67
2	b30	b31	b32	b33	b50	b51	b52	b53	a50	a51	a52	a53	a70	a71	a72	a73
3	b34	b35	b36	b37	b54	b55	b56	b57	a54	a55	a56	a57	a74	a75	a76	a77

Read a61 – Hit

Read a71 – Hit

Read b61 – Hit

Write a61 – Hit

0	b60	b61	b62	b63	b40	b41	b42	b43	a40	a41	a42	a43	a60	a61	a62	a63
1	b24	b25	b26	b27	b44	b45	b46	b47	a44	a45	a46	a47	a64	a65	a66	a67
2	b30	b31	b32	b33	b50	b51	b52	b53	a50	a51	a52	a53	a70	a71	a72	a73
3	b34	b35	b36	b37	b54	b55	b56	b57	a54	a55	a56	a57	a74	a75	a76	a77

Όμοια με το $i=6, j=1$ είναι και τα $i=6, j=2$ και $i=6, j=3$ και η cache παραμένει σε αυτήν την κατάσταση.

Για $i=6, j=4$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	b60	b61	b62	b63	b40	b41	b42	b43	a40	a41	a42	a43	a60	a61	a62	a63
1	b24	b25	b26	b27	b44	b45	b46	b47	a44	a45	a46	a47	a64	a65	a66	a67
2	b30	b31	b32	b33	b50	b51	b52	b53	a50	a51	a52	a53	a70	a71	a72	a73
3	b34	b35	b36	b37	b54	b55	b56	b57	a54	a55	a56	a57	a74	a75	a76	a77

Read a64 – Hit

Read a74 – Hit

Read b64 – Compulsory Miss

Write a64 – Hit

0	b60	b61	b62	b63	b40	b41	b42	b43	a40	a41	a42	a43	a60	a61	a62	a63
1	b64	b65	b66	b67	b44	b45	b46	b47	a44	a45	a46	a47	a64	a65	a66	a67
2	b30	b31	b32	b33	b50	b51	b52	b53	a50	a51	a52	a53	a70	a71	a72	a73
3	b34	b35	b36	b37	b54	b55	b56	b57	a54	a55	a56	a57	a74	a75	a76	a77

Για $i=6, j=5$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	b60	b61	b62	b63	b40	b41	b42	b43	a40	a41	a42	a43	a60	a61	a62	a63
1	b64	b65	b66	b67	b44	b45	b46	b47	a44	a45	a46	a47	a64	a65	a66	a67
2	b30	b31	b32	b33	b50	b51	b52	b53	a50	a51	a52	a53	a70	a71	a72	a73
3	b34	b35	b36	b37	b54	b55	b56	b57	a54	a55	a56	a57	a74	a75	a76	a77

Read a65 – Hit

Read a75 – Hit

Read b65 – Hit

Write a65 – Hit

0	b60	b61	b62	b63	b40	b41	b42	b43	a40	a41	a42	a43	a60	a61	a62	a63
1	b64	b65	b66	b67	b44	b45	b46	b47	a44	a45	a46	a47	a64	a65	a66	a67
2	b30	b31	b32	b33	b50	b51	b52	b53	a50	a51	a52	a53	a70	a71	a72	a73
3	b34	b35	b36	b37	b54	b55	b56	b57	a54	a55	a56	a57	a74	a75	a76	a77

Όμοια με το $i=6, j=5$ είναι και τα $i=6, j=6$ και $i=6, j=7$ και η cache παραμένει σε αυτήν την κατάσταση. Άρα για $i=6$ έχουμε συνολικά 30 hit και 2 misses = 32 accesses ($8*4$).

Άρα, μέχρι τώρα έχουμε 194 hits και 30 misses = 224 accesses ($8*4*7$).

Για $i=7, j=0$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	b60	b61	b62	b63	b40	b41	b42	b43	a40	a41	a42	a43	a60	a61	a62	a63
1	b64	b65	b66	b67	b44	b45	b46	b47	a44	a45	a46	a47	a64	a65	a66	a67
2	b30	b31	b32	b33	b50	b51	b52	b53	a50	a51	a52	a53	a70	a71	a72	a73
3	b34	b35	b36	b37	b54	b55	b56	b57	a54	a55	a56	a57	a74	a75	a76	a77

Read a70 – Hit

Read a80 – Compulsory Miss

Read b70 – Compulsory Miss

Write a70 – Hit

0	b60	b61	b62	b63	a80	a81	a82	a83	a40	a41	a42	a43	a60	a61	a62	a63
1	b64	b65	b66	b67	b44	b45	b46	b47	a44	a45	a46	a47	a64	a65	a66	a67
2	b70	b71	b72	b73	b50	b51	b52	b53	a50	a51	a52	a53	a70	a71	a72	a73
3	b34	b35	b36	b37	b54	b55	b56	b57	a54	a55	a56	a57	a74	a75	a76	a77

Για $i=7, j=1$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	b60	b61	b62	b63	a80	a81	a82	a83	a40	a41	a42	a43	a60	a61	a62	a63
1	b64	b65	b66	b67	b44	b45	b46	b47	a44	a45	a46	a47	a64	a65	a66	a67
2	b70	b71	b72	b73	b50	b51	b52	b53	a50	a51	a52	a53	a70	a71	a72	a73
3	b34	b35	b36	b37	b54	b55	b56	b57	a54	a55	a56	a57	a74	a75	a76	a77

Read a71 – Hit

Read a81 – Hit

Read b71 – Hit

Write a71 – Hit

0	b60	b61	b62	b63	a80	a81	a82	a83	a40	a41	a42	a43	a60	a61	a62	a63
1	b64	b65	b66	b67	b44	b45	b46	b47	a44	a45	a46	a47	a64	a65	a66	a67
2	b70	b71	b72	b73	b50	b51	b52	b53	a50	a51	a52	a53	a70	a71	a72	a73
3	b34	b35	b36	b37	b54	b55	b56	b57	a54	a55	a56	a57	a74	a75	a76	a77

Όμοια με το $i=7, j=1$ είναι και τα $i=7, j=2$ και $i=7, j=3$ και η cache παραμένει σε αυτήν την κατάσταση.

Για $i=7, j=4$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	b60	b61	b62	b63	a80	a81	a82	a83	a40	a41	a42	a43	a60	a61	a62	a63
1	b64	b65	b66	b67	b44	b45	b46	b47	a44	a45	a46	a47	a64	a65	a66	a67
2	b70	b71	b72	b73	b50	b51	b52	b53	a50	a51	a52	a53	a70	a71	a72	a73
3	b34	b35	b36	b37	b54	b55	b56	b57	a54	a55	a56	a57	a74	a75	a76	a77

Read a74 – Hit

Read a84 – Compulsory Miss

Read b74 – Compulsory Miss

Write a74 – Hit

0	b60	b61	b62	b63	a80	a81	a82	a83	a40	a41	a42	a43	a60	a61	a62	a63
1	b64	b65	b66	b67	a84	a85	a86	a87	a44	a45	a46	a47	a64	a65	a66	a67
2	b70	b71	b72	b73	b50	b51	b52	b53	a50	a51	a52	a53	a70	a71	a72	a73
3	b74	b75	b76	b77	b54	b55	b56	b57	a54	a55	a56	a57	a74	a75	a76	a77

Για $i=7, j=5$: (αρχικά η κατάσταση της cache πριν αυτό το στάδιο, έπειτα τα read/write με τα αντίστοιχα τους hit/miss και μετά η τελική κατάσταση της cache)

0	b60	b61	b62	b63	a80	a81	a82	a83	a40	a41	a42	a43	a60	a61	a62	a63
1	b64	b65	b66	b67	a84	a85	a86	a87	a44	a45	a46	a47	a64	a65	a66	a67
2	b70	b71	b72	b73	b50	b51	b52	b53	a50	a51	a52	a53	a70	a71	a72	a73
3	b74	b75	b76	b77	b54	b55	b56	b57	a54	a55	a56	a57	a74	a75	a76	a77

Read a75 – Hit

Read a85 – Hit

Read b75 – Hit

Write a75 – Hit

0	b60	b61	b62	b63	a80	a81	a82	a83	a40	a41	a42	a43	a60	a61	a62	a63
1	b64	b65	b66	b67	a84	a85	a86	a87	a44	a45	a46	a47	a64	a65	a66	a67
2	b70	b71	b72	b73	b50	b51	b52	b53	a50	a51	a52	a53	a70	a71	a72	a73
3	b74	b75	b76	b77	b54	b55	b56	b57	a54	a55	a56	a57	a74	a75	a76	a77

Όμοια με το $i=7, j=5$ είναι και τα $i=7, j=6$ και $i=7, j=7$ και η cache παραμένει σε αυτήν την κατάσταση. Άρα για $i=7$ έχουμε συνολικά 28 hit και 4 misses = 32 accesses ($8*4$).

Τελικό σύνολο έχουμε 222 hits και 34 misses = 256 accesses ($8*8*4$).

Hit rate 4-way = $222/256 = 86.718 \%$

Hit rate 2-way = $142/256 = 55.468 \%$

Παρατηρούμε ότι ο αριθμός των hits αυξήθηκε, επομένως και ο αριθμός των misses μειώθηκε. Επομένως, θα αντικαθιστούσαμε την 2-way associative με την 4-way associative, καθώς όπως είδαμε και παραπάνω έχουμε υψηλότερο hit rate %, κάτι που είναι φυσικά επιθυμητό και μάλιστα βασικό ζητούμενο.