



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
www.cslab.ece.ntua.gr

1 Ιουλίου 2005

ΠΡΟΗΓΜΕΝΑ ΘΕΜΑΤΑ ΟΡΓΑΝΩΣΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

2η Σειρά Ασκήσεων και η Λύση

Οι κρυφές μνήμες χωρίς κλειδώματα (lockup-free caches) σχεδιάστηκαν για να επιταχύνουν το ρυθμό εκτέλεσης των εντολών στον επεξεργαστή, επικαλύπτοντας το χειρισμό των αστοχιών κρυφής μνήμης (cache misses) με την επεξεργασία άλλων εντολών ή άλλων αστοχιών κρυφής μνήμης. Στο πρόβλημα αυτό υποθέστε ότι η ιεραρχία μνήμης διαθέτει κρυφή μνήμη ενός επιπέδου, με μέγεθος cache line 64bytes και διάδρομο επικοινωνίας κύριας μνήμης με την κρυφή μνήμη (memory bus) 16bytes. Υποθέστε, επίσης, ότι ο χρόνος εύρεσης των ζητούμενων δεδομένων στη μνήμη είναι $t_{\text{miss}} = 5$ cycles, και ο χρόνος μεταφοράς δεδομένων στη κρυφή μνήμη είναι 1 cycle/16 bytes, δηλαδή για την μεταφορά ολόκληρης της cache line: $t_{\text{transfer}} = 4$ cycles. Τέλος, δίνεται ότι στο σύστημα 1 word = 4 bytes.

α) Εξετάζουμε την εκτέλεση του εξής κώδικα:

```
for (i = 0; i < 1000; i++) {  
    a += A[i] + B[i];  
}
```

Θεωρούμε ότι οι τιμές των μεταβλητών $4 \cdot i$, a βρίσκονται στους καταχωρητές $\$s4$, $\$t0$ και οι διευθύνσεις των πινάκων A και B στους $\$s1$ και $\$s2$. Το περιεχόμενο του καταχωρητή $\$s4$ είναι αρχικά 0. Ο αντίστοιχος κώδικας σε assembly είναι ο εξής:

```
Loop:      add    $t1, $s1, $s4  
           add    $t2, $s2, $s4  
           lw     $t3, 0($t1)  
           lw     $t4, 0($t2)  
           add    $t0, $t0, $t3  
           add    $t0, $t0, $t4  
           addi   $s4, $s4, 4  
           bne   $s4, $s5, Loop  
  
Exit:
```

Στον πίνακα που ακολουθεί φαίνεται η ροή εκτέλεσης του παραπάνω βρόχου, υποθέτοντας ότι κανένα από τα στοιχεία των πινάκων A και B δεν βρίσκονται ήδη στην κρυφή μνήμη (m = miss latency, t = transfer latency, A = lw from A , B = lw from B , a = add/addi, b = branch). Επιπλέον, διαθέτουμε απλή κρυφή μνήμη.

cycles										1										
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
array A				m	m	m	m	m	t	t	t	t								
array B														m	m	m	m	m	t	t
Execute	a	a	A											B						
cycles	2									3										
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
array A																				
array B	t	t																		
Execute			a	a	a	b	a	a	A	B	a	a	a	b	a	a	A	B	a	...

Πόσοι κύκλοι απαιτούνται για την εκτέλεση των 1000 επαναλήψεων του `for loop`;

β) Θεωρείστε ότι ο σχεδιασμός της κρυφή μνήμης επιτρέπει την προώθηση στον επεξεργαστή της ζητούμενης λέξης αμέσως μόλις αφιχθεί στην κρυφή μνήμη, χωρίς να χρειάζεται να ολοκληρωθεί η μεταφορά ολόκληρης της cache line. Ωστόσο, μόνο μία αστοχία μπορεί να βρίσκεται σε εκκρεμότητα. Εξετάστε δύο ενδεχόμενα:

- Πρέπει να ολοκληρωθεί η μεταφορά ολόκληρης της cache line στην κρυφή μνήμη. Μόνο τότε μπορεί να αποσταλεί η επόμενη αίτηση αστοχίας στη μνήμη.
- Αρκεί να αφιχθεί η ζητούμενη από τον επεξεργαστή λέξη, για να υπάρχει η δυνατότητα αποστολής της επόμενης αίτησης αστοχίας στη μνήμη.

Φτιάξτε από ένα διάγραμμα χρονισμού για κάθε μία από τις περιπτώσεις (βi) και (βii), όπως αυτό του παραπάνω πίνακα. Πόσοι κύκλοι απαιτούνται για την εκτέλεση των 1000 επαναλήψεων του `for loop`; Στην περίπτωση (βi) υπάρχει τρόπος να μειώσουμε τον αριθμό των κύκλων που απαιτούνται για την εκτέλεση του βρόχου με βελτιστοποίηση του κώδικα;

γ) Θεωρείστε ότι ο σχεδιασμός της κρυφή μνήμης επιτρέπει την προώθηση στον επεξεργαστή της ζητούμενης λέξης αμέσως μόλις αφιχθεί στην κρυφή μνήμη, χωρίς να χρειάζεται να ολοκληρωθεί η μεταφορά ολόκληρης της cache line, και επιπλέον την εκκρεμότητα πολλών αστοχιών. Φτιάξτε ένα διάγραμμα χρονισμού για αυτόν τον σχεδιασμό της κρυφής μνήμης, όπως αυτό του παραπάνω πίνακα. Πόσοι κύκλοι απαιτούνται για την εκτέλεση των 1000 επαναλήψεων του `for loop`;

Λύση

α) Υποθέτοντας ότι κανένα από τα στοιχεία των πινάκων A και B δεν βρίσκονται ήδη στην κρυφή μνήμη, συνεπάγεται ότι κατά την πρώτη αναφορά σε ένα από τα στοιχεία μίας cache line θα πραγματοποιείται cache miss. Οπότε, θα φορτώνεται μία ολόκληρη cache line στην cache. Όπως φαίνεται από το τμήμα του κώδικα που εξετάζουμε, γίνεται σειριακή προσπέλαση σε στοιχεία μονοδιάστατων πινάκων (τα οποία προφανώς είναι αποθηκευμένα σε γειτονικές θέσεις μνήμης). Άρα, κατά την σειριακή προσπέλαση καθενός από τους πίνακες πραγματοποιείται 1 miss, και στη συνέχεια $(κ-1)$ hits, όπου $κ$ ο αριθμός των λέξεων που χωρούν σε μία cache line.

$$κ = \frac{\text{μέγεθος cache line}}{\text{μέγεθος λέξης}} = \frac{64\text{bytes}}{4\text{bytes / word}} = 16\text{words}$$

Δηλαδή πραγματοποιείται 1 miss, 15 hits, 1 miss, 15 hits... 1 miss, 7 hits.

Υποθέτουμε δηλαδή, ότι το πρώτο στοιχείο του πίνακα A είναι αποθηκευμένο στην πρώτη θέση μίας cache line. Ομοίως για τον πίνακα B. Επιπλέον, οι δύο cache lines που περιέχουν τα στοιχεία των δύο διαφορετικών πινάκων και προσπελούνται εναλλακτικά, δεν συγκρούονται στην cache.

Επομένως στις 1000 επαναλήψεις του βρόχου θα έχουμε $\left\lceil \frac{1000}{16} \right\rceil = 63$ misses και $(1000-63) = 937$ hits.

Σε περίπτωση που το πρώτο στοιχείο καθενός από τους δύο πίνακες δεν είναι ευθυγραμμισμένο στην αρχή μίας cache line, στην χειρότερη περίπτωση θα είχαμε 64 misses και 936 hits. Για παράδειγμα, μία από αυτές τις περιπτώσεις είναι: 1 miss, 6 hits, 1 miss, 15 hits... 1 miss, 15 hits, 1 miss.

Όπως φαίνεται στον πίνακα της εκφώνησης, στην περίπτωση μίας επανάληψης του βρόχου όπου οι αναφορές στη μνήμη είναι misses, η διάρκεια εκτέλεσης είναι 26 κύκλοι. Στην περίπτωση που οι αναφορές στη μνήμη είναι hits, η διάρκεια εκτέλεσης είναι 8 κύκλοι.

Συνολικά: $26 \times 63 + 8 \times 937 = 9134$ κύκλοι εκτέλεσης

H $26 \times 64 + 8 \times 936 = 9152$ κύκλοι εκτέλεσης.

βι) Αν η ζητούμενη λέξη προωθείται στον επεξεργαστή αμέσως μόλις αφιχθεί στην κρυφή μνήμη, αλλά πρέπει να ολοκληρωθεί η μεταφορά ολόκληρης της cache line στην κρυφή μνήμη για να επιλυθεί η εκκρεμότητα, το σχήμα εκτέλεσης φαίνεται στον παρακάτω πίνακα.

Δηλαδή, μετά την μεταφορά της ζητούμενης λέξης στον επεξεργαστή (μετά τον πρώτο κύκλο t), μπορεί να συνεχιστεί κανονικά η εκτέλεση των εντολών που ακολουθούν ενώ ταυτόχρονα μεταφέρονται οι υπόλοιπες λέξεις της cache line στην cache (βλ. επικάλυψη κύκλων t με τους κύκλους aab). Όταν όμως διαγνωσθεί μία εντολή προσπέλασης της μνήμης, η οποία είναι miss, θα πρέπει να περιμένουμε να επιλυθεί η προηγούμενη αστοχία (να ολοκληρωθεί η μεταφορά της cache line στην cache: να ολοκληρωθούν και οι 4 κύκλοι t) για να προωθηθεί η επόμενη αστοχία στην μνήμη (βλ. τι συμβαίνει με την εντολή ανάγνωσης B).

cycles											1									
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
array A				m	m	m	m	m	t	t	t	t								
array B													m	m	m	m	m	t	t	t
Execute	a	a	A							B									a	a
cycles	2										3									
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
array A																				
array B	t																			
Execute	a	b	a	a	A	B	a	a	a	b	a	a	A	B	a	a	a	b	a	...

Στην περίπτωση μίας επανάληψης του βρόχου όπου οι αναφορές στη μνήμη είναι misses, η διάρκεια εκτέλεσης είναι 22 κύκλοι. Στην περίπτωση που οι αναφορές στη μνήμη είναι hits, ομοίως με το ερώτημα (α), η διάρκεια εκτέλεσης είναι 8 κύκλοι.

Συνολικά: $22 \times 63 + 8 \times 937 = 8882$ κύκλοι εκτέλεσης

H $22 \times 64 + 8 \times 936 = 8896$ κύκλοι εκτέλεσης.

Μπορούμε όμως να πραγματοποιήσουμε την εξής αναδιάταξη κώδικα:

```

Loop:   add  $t1, $s1, $s4
        lw   $t3, 4($t1)
        add  $t2, $s2, $s4
        add  $t0, $t0, $t3
        lw   $t4, 0($t2)
        addi $s4, $s4, 4
        add  $t0, $t0, $t4
        bne  $s4, $s5, Loop
    
```

Κατά αυτόν τον τρόπο γλιτώνουμε τους δύο κύκλους καθυστέρησης μετά από κάθε εντολή lwA η οποία καταλήγει σε αστοχία. Επομένως, στην περίπτωση μίας επανάληψης του βρόχου όπου οι αναφορές στη μνήμη είναι misses, η διάρκεια εκτέλεσης είναι 20 κύκλοι. Και οι συνολικοί χρόνοι εκτέλεσης είναι:

$$20 \times 63 + 8 \times 937 = 8756 \text{ κύκλοι εκτέλεσης}$$

$$H \quad 20 \times 64 + 8 \times 936 = 8768 \text{ κύκλοι εκτέλεσης.}$$

βii) Αν η ζητούμενη λέξη προωθείται στον επεξεργαστή αμέσως μόλις αφιχθεί στην κρυφή μνήμη, και αρκεί να αφιχθεί η ζητούμενη από τον επεξεργαστή λέξη για να επιλυθεί η εκκρεμότητα, το σχήμα εκτέλεσης φαίνεται στον παρακάτω πίνακα:

											1											
cycles	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9		
array A				m	m	m	m	m	t	t	t	t										
array B											m	m	m	m	m	t	t	t	t			
Execute	a	a	A							B									a	a	a	b
											2											
cycles	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9		
array A																						
array B																						
Execute	a	a	A	B	a	a	a	b	a	a	A	B	a	a	a	b	a	a	A	...		

Στην περίπτωση μίας επανάληψης του βρόχου όπου οι αναφορές στη μνήμη είναι misses, η διάρκεια εκτέλεσης είναι 22 κύκλοι. Στην περίπτωση που οι αναφορές στη μνήμη είναι hits, ομοίως με το ερώτημα (α), η διάρκεια εκτέλεσης είναι 8 κύκλοι.

$$\text{Συνολικά: } 20 \times 63 + 8 \times 937 = 8756 \text{ κύκλοι εκτέλεσης}$$

$$H \quad 20 \times 64 + 8 \times 936 = 8768 \text{ κύκλοι εκτέλεσης.}$$

γ) Αν η ζητούμενη λέξη προωθείται στον επεξεργαστή αμέσως μόλις αφιχθεί στην κρυφή μνήμη, και επιπλέον επιτρέπεται η εκκρεμότητα πολλών αστοχιών, το σχήμα εκτέλεσης είναι το εξής:

Δηλαδή, μπορούν πλέον να επικαλύπτονται οι χρόνοι ανάγνωσης από τη μνήμη, όχι όμως οι χρόνοι μεταφοράς των δεδομένων στην cache μέσω του memory bus. Κάτι τέτοιο εξάλλου θα ήταν αδύνατο, εκτός και αν διπλασιαζόταν το εύρος του memory bus. Στην περίπτωσή μας (όπου το memory bus έχει σταθερό εύρος), θεωρούμε τη βέλτιστη τεχνική μεταφοράς των δεδομένων: τα δεδομένα όλων των αστοχιών που εκκρεμούν μεταφέρονται εναλλακτικά (interleaved way), έτσι ώστε τα ζητούμενα δεδομένα να φτάνουν στον επεξεργαστή όσο το δυνατόν γρηγορότερα. Με τον επιλεγμένο τρόπο οι εξαρτήσεις δεδομένων καθυστερούν την εκτέλεση του προγράμματος κατά τον ελάχιστο δυνατό χρόνο.

											1										
cycles	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	
array A				m	m	m	m	m	t		t		t		t						
array B					m	m	m	m	m	t		t		t		t					
Execute	a	a	A	B							a	a	a	b	a	a	A	B	a	a	a
											2										
cycles	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	
array A																					
array B																					
Execute	b	a	a	A	B	a	a	a	b	a	a	A	B	a	a	a	b		a	...	

Στην περίπτωση μίας επανάληψης του βρόχου όπου οι αναφορές στη μνήμη είναι misses, η διάρκεια εκτέλεσης είναι 13 κύκλοι. Στην περίπτωση που οι αναφορές στη μνήμη είναι hits, ομοίως με το ερώτημα (α), η διάρκεια εκτέλεσης είναι 8 κύκλοι.

Συνολικά: $13 \times 63 + 8 \times 937 = 8315$ κύκλοι εκτέλεσης

Η $13 \times 64 + 8 \times 936 = 8320$ κύκλοι εκτέλεσης.