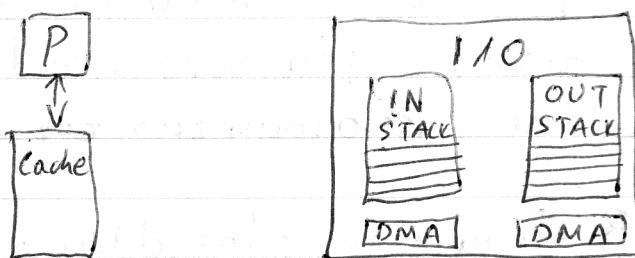


Άριθμος 14 - Οργάνωση Υποδοχής
 Οντότυπο: Αναγνωστικός Εμπαρουντ
 AM:3618

Άριθμος 14.1



a) Ο επεξεργαστής P, θα διαβάσει τα δεδομένα στην μνήμη περιττών, που έχει miss στο Cache, διαδικασία υπάρχει αντιγράφο της διαχείρισης γειτίας στη μνήμη Cache. Σε κάθε addrs περιττών, ο επεξεργαστής θα ξεχινή τα δεδομένα από Main Memory.

B) Δεδομένου ότι η μνήμη Cache είναι write-through και δεδομένα στην ορίσμενη περιοχή (OUT), θα είναι ανανεωμένα και οχι ένα παλιότερο αντιγράφο. Μόδις διεύρυνσης δοιών μία μεταφορά DMA, τα δεδομένα είναι ανανεωμένα, αρα και δεδομένα. Και επίσης η περιττώσης θα πρέπει να υπάρχει ένα bit εδήποτε πως οντώς έχουν ανανεωθεί τα δεδομένα. Όπως το status, της γύρας αριθμητής 13.

γ) Στην περιττώση που έχουμε μία write-back μνήμη cache, θα έχουμε δεδομένα δεδομένα στη main memory, ή όχι μία addrs γειτία πάντα να ανταπαγγέλει την γειτία που δεδουλύεται στη cache και ορίζει να ανταπαγγέλει τα δεδομένα στη cache. Σε καθειδία περιττώσης, θα κατατίθουν δεδομένα δεδομένα στην γύρων γέιτια.

Άσκηση 14.3

- a) Υποθέτουμε πως οι κρυφές μνήμες είναι write-back.
- i) Η sharedVar είναι σε μνήμη και έχει τιμή 0.
 - ii) Ο επεξεργαστής A, δημιουργεί την sharedVar, ανάγνωση cache και αδρόχα. Είστε ποιποι είναι sharedVar ανοίγει μνήμη και ενδογέρμονται αυτήν. Μετά τον ενδογέρμοντα περιβλήτη sharedVar έχει αρχίσει την τιμή 0 σε μνήμη.
 - iii) Ο επεξεργαστής B, δημιουργεί την sharedVar και ενημερώνει ακρίβως στην τηλεοπτική βίντα 2(ii).
 - iv) Ο επεξεργαστής A εκχωρεί την sharedVar, που έχει σε μνήμη cache, την τιμή 1. Η περιβλήτης sharedVar σε μνήμη (Main), δούλεψε την write-back, διεύθυντας ακόμη την τιμή 0.
 - v) Ο επεξεργαστής B διαθέτει την περιβλήτη sharedVar ανοίγει μνήμη που έχει την τιμή 0 και δημιουργεί ενδογέρμοντα αυτήν. Κι αυτό γιατί ο επεξεργαστής A, σε χρήση της επεξεργαστικής μνήμης είναι sharedVar ανοίγει μνήμη cache και δεν αναρτεί την τιμή σε μνήμη (Main).
- b)
- i) Η sharedVar είναι σε μνήμη (main) και έχει την τιμή 0.
 - ii) Ο επεξεργαστής A δημιουργεί την sharedVar ανοίγει cache και αδρόχα. Ως στην περιβλήτη αυτή σε μνήμη (main) και αρχίζει την κανονική ενδογέρμηση μνήμης, την ανοδικεύει σε μνήμη (main). Στην κανονική μνήμη στην περιβλήτη S.
 - iii) Ο επεξεργαστής B δημιουργεί την sharedVar και μεταφέρει από την περιβλήτη A στην περιβλήτη B την επεξεργαστική μνήμη cache. Τα αποτελέσματα αυτού του περιπτώσεων είναι στην περιβλήτη B την αρχή της κανονικής ενδογέρμησης μνήμης και δημιουργεί την ανοδικεύουσα μνήμη cache περιβλήτη S.

→ πιστώ iv) και v)

Άσκηση 14.4

a) lw \$t0, 120(\$0)
 addi \$t0, \$t0, 1
 sw \$t0, 128(\$0)
 lw \$t0, 124(\$0)
 addi \$t0, \$t0, 2
 sw \$t0, 132(\$0)

b) lw \$t0, 120(\$0)
 lw \$t1, 124(\$0)
 addi \$t0, \$t0, 1
 addi \$t1, \$t1, 2
 sw \$t0, 128(\$0)
 sw \$t1, 132(\$0)

Όταν ερεδιούμε το πρόγραμμα α) με την κλασική pipeline της S-Bαθμίδων, χρειάζονται 2 κύκλοι ποδογραφίας, δηλαδί 2 έπαντας lw και addi. Με την αναδιαράθη του προγράμματος (b), δεν χρειάζονται οι 2 κύκλοι, καθώς δεν υπάρχει πιά εξαρτησία μεταξύ των εποντών, addi χρησιμοποιεί 1 extra register, τον \$t1, για να κρατήσουμε την τιμή του b και να αδηδάνετε στον \$t0 να εχουμε την τιμή του a).

γ)
 lw \$t0, 0(\$12)
 addi \$t0, \$t0, 1
 sw \$t0, 0(\$14)
 lw \$t0, 0(\$13)
 addi \$t0, \$t0, 2
 sw \$t0, 0(\$15)

Για να κάνουμε instruction scheduling θα χρειαζεί να ξέρουμε αν οι pointers px και pb έχουν διαφορετική ή ίδιας τύπου. Αν οι pointers px και pb είναι διαφορετικοί τότε θα γίνει 6ωρο instruction scheduling και το πρόγραμμα θα εκτελεστεί σταυρό (B). Αν δεν θα προσπαθήσει μία τιμή από την δεύτερη που φαίνεται στον καταχωριστή πα και θα την βάσεις καταχωρίστηκε στον καταχωριστή πα και στην δεύτερη που δείχνει ο καταχωριστής pb και θα την βάσεις είναι καταχωρίστηκε στον t1. Μετά τους κατατυλόντας υποτομήματα, θα αναδικείται την αρχική

μένη την του το σε δίδια μνήμης που διχτυώνει ο καταχωριστής ρχ καταγνωρίζει τη την του τη σε δίδια μνήμης που διχτυώνει ο καταχωριστής ρχ. Άνοι pointers είναι σίστοι τούτε σε μη παραπάνω διαδικασία, αρχικά το το θα έχει τη την ανοιχτή δίδια μνήμης που διχτυώνει το πα σε w το +1, θα έχει την (ίσως διαφορετική) ανοιχτή δίδια μνήμης που διχτυώνει το ρχ. Επιτά μετα τους unaligned memory. Θα έχουμε datatypes την ρχ, την καρβ. Έτσι θα προβλαδίσει το πρόγραμμα να αποδημεύει το σε μια dǎdoς διεύθυνση σε μνήμη και το ίδιο θα βγει με το +1. και το γρ. Εφόσον ο compiler δεν μπορεί να εδεχθεί την την την ρχ και ρb, δεν γίνεται instruction scheduling κατά το compile και αφνούριε το hardware, να κάνει διεύθυνσης αν μπορεί.

8) Η δεύτερη ενοτητή load θα περιγράψει ότι τα ρχ και ρb έχουν την ιδία την σε 1% των περιπτώσεων. Έτσι θα έχουμε εξαρετή ενοτητήν περισσότερο την 2^{10} λιών και την 1^{10} λιών. Σε αυτήν την περιπτώση, δηλαδή το 99% των περιπτώσεων, οι ρχ και ρb έχουν διαφορετική την και είναι η δεύτερη load είναι ανιχαρτική να εγκεκληθεί. Το hardware θα καταφέρει να βιδιώνει το πρόγραμμα, αφού μπορεί να βιδεύει την την την pointers ρχ και ρb. Και που δεν μπορούνε να ταρι ο compiler κατά το compile time..

Θα χαρτεί χρόνος δόμων addend εξαρτημένων καθε φορά που υπάρχει ενοτητής εξαρτησεων από την προηγούμενη την σε κάθε page fault και miss. Αν δε χαρτεί χρόνος οταν δεν υπάρχει - υπάρχουν addend εξαρτημένων μεταξύ την προεδρίας

Άσκηση 14.5

- a) Μιας και ο επεξεργασμός δεν έχει multithreading αλλά έχει το instruction scheduling και το out-of-order, ο επεξεργασμός δαπέχει $80 - 8 = 72$ μ.
- b) Το νήπιο A δεν θα τρέξει πιο γρήγορα μιας κομμάτια τους 8 μνους καταφέρει να κερδίσει με χρή τα δύο ανάγκην από την load, θα αρχίσει να εκτελεί εποδει το νήπιο B, με χρή των 6 μνους που κανοία αποκτιανεί για πάντη το πρόγραμμα για διαφορούς αδιόρθωτος δόγματος.
- c) Τώρα ους έχουμε multithreading δεν θα χαρίζει κανένας κύριος ποδόρος, μιας και μεταξύ miss του νήπιου A, θα καταφέρει να τρέξει για αδιόρθωτο και επειδή δεν θα ρέψει το νήπιο B για το ουδέτερο 8 μνού, θα λαμβάνει τον χρόνο το νήπιο A θα εξυπάρει ανάρμονη το load και θα διατηρήσει με την εκτέλεση των εισοδών του νήπιου B το νήπιο A θα είναι έτοιμο να γνωρίσει την εκτέλεση των εισοδών του.
- 14.3iv) O επεξεργασμός A αρράβει την αριθμητική της sharedVal, την αρράβει σε καραβάνη M και ενημερώνει μέσω του bus για την αρράβηση αυτή. O επεξεργασμός B θα δέχει το μήπιο μέσω του bus και θα αρράβει την καραβάνη της sharedVal στην cache του ανοίγει σε 1 (Invalid).

143) Ο επεξεργαστής προβάδισε διαθέσιμη την
sharedVal αλλά είναι σε κατάσταση Ι και δε γίνεται
miss. Ο αριθμός παραγόμενης στο bus και του ήτοι
αναρμόνιος ο επεξεργαστής Α με την τιμή της
sharedVal νωρίς είναι 11. Ο επεξεργαστής Β, δεκτεί
ταυτόχρονα την θέση και διαπιστώνει ότι
ταυτίζεται με την sharedVal στην παραγόμενη cache του.