



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΩΝ & ΥΛΙΚΟΥ
ΗΡΥ 415 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΠΟΛΟΓΙΣΤΩΝ

Χειμερινό Εξάμηνο 2017-8

Εργαστήριο 3: Tomasulo + Reorder Buffer + Ld/St Queue

ΕΚΔΟΣΗ 1.1, Δ. Πνευματικάτος

Απαιτούμενα:

Καλή κατανόηση της VHDL και του αλγορίθμου Tomasulo, Εργαστήρια #1, #2.

Περιγραφή:

Προσθέστε τη λειτουργικότητα της επικύρωσης εντολών σε σειρά χρησιμοποιώντας ένα Reorder Buffer (υποχρεωτικό), και ικανότητα πρόσβασης μνήμης μέσω μιας ενοποιημένης ουράς πρόσβασης μνήμης Ld/St (προαιρετικό, +20%).

Reorder Buffer:

Αντικαταστήστε τη Register File (και το Register Result Status) με ένα Reorder Buffer. Η Register File θα είναι πλέον οργανωμένη ως δομή μνήμης χωρίς συσχετιστικό ψάξιμο. Το ρόλο αυτό θα αναλάβει ο Reorder Buffer που οργανώνεται σαν μια ουρά η οποία γεμίζει κατά την έκδοση μιας εντολής, και αδειάζει σειριακά όταν η εντολή στην κεφαλή της ουράς έχει ολοκληρωθεί.

Δείξτε τα απαραίτητα πεδία της κάθε καταχώρησης. Συμπεριλάβετε το PC της κάθε εντολής, και ένα Exception Status (οι λειτουργικές μονάδες που έχετε υλοποιήσει δεν παράγουν εξαιρέσεις αλλά στη γενική περίπτωση υποστηρίζονται εξαιρέσεις).

1. Δώστε το ενημερωμένο σχηματικό διάγραμμα του συστήματος, και του Reorder Buffer.
2. Δώστε το διάγραμμα χρονισμού του.
3. Ολοκληρώστε και ελέγξτε το Reorder Buffer.
4. Περιγράψτε τη στρατηγική και τα αποτελέσματα του ελέγχου.

Ουρά Πρόσβασης Μνήμης (Load/Store Queue), προαιρετικό, +20%:

Προσθέστε μια επιπλέον λειτουργική μονάδα, την πρόσβαση μνήμης μέσω μιας ενοποιημένης ουράς πρόσβασης μνήμης (Load/Store Queue). Η δομή μοιάζει αρκετά με τον Reorder Buffer αφού και αυτή γεμίζει και αδειάζει σαν ουρά. Όμως η ουρά Load/Store διατηρεί πεδία διευθύνσεων μνήμης και συγκρίνει για εξαρτήσεις μνήμης.

Θεωρήστε ότι οι προσβάσεις μνήμης χρησιμοποιούν μόνο καταχωρητή βάσης (δηλαδή το Offset είναι πάντα 0 (π.χ. lw \$2, 0(\$5), sw \$7, 0(\$13)), και έτσι δε χρειάζεται πράξη και η διεύθυνση είναι έτοιμη μόλις διαβαστεί από την Register File ή από το CDB σε περίπτωση εξάρτησης. Όσο μια διεύθυνση δεν έχει υπολογιστεί ακόμα θεωρείται «άγνωστη» διεύθυνση.

Η ουρά ελέγχει για εξαρτήσεις μέσω μνήμης: ένα load εξαρτάται από προηγούμενα store στην ίδια ή σε άγνωστη διεύθυνση, και ένα store εξαρτάται από προηγούμενα load ή store στην ίδια ή σε άγνωστη διεύθυνση.

Οι εντολές load και store είναι έτοιμες όταν έχουν διαθέσιμη τη διεύθυνση μνήμης και στην περίπτωση των stores και την τιμή που θα αποθηκεύσουν, και δεν έχουν εξάρτηση μέσω μνήμης με κάποια άλλη πρόσβαση μνήμης.

Η ουρά δίνει προτεραιότητα στα Loads, και αν υπάρχει εκδίδει προς την μνήμη το πρώτο έτοιμο load. Όταν δεν υπάρχει έτοιμο Load εκδίδει προς την μνήμη το πρώτο έτοιμο store.

Η μνήμη είναι pipelined και έχει καθυστέρηση 3 κύκλους συνδέεται δε με την μονάδα ουράς Load/Store με μία πόρτα.

Διεξαγωγή:

1. Δείξτε τα απαραίτητα πεδία της κάθε καταχώρησης
2. Δώστε το ενημερωμένο σχηματικό διάγραμμα του συστήματος, και της Ουράς Πρόσβασης Μνήμης.
3. Δώστε το διάγραμμα χρονισμού του.
4. Ολοκληρώστε και ελέγξτε το Reorder Buffer.
5. Περιγράψτε τη στρατηγική και τα αποτελέσματα του ελέγχου