

Αναφορά Εργαστηριακής άσκησης 5

Ομάδα LAB31239665

Καλογεράκης Στέφανος AM:2015030064

Πίσκοπος Διονύσης AM:2015030115

Σκοπός εργαστηριακής άσκησης

Σκοπός της πέμπτης και τελευταίας εργαστηριακής άσκησης ήταν η υλοποίηση ενός pipeline επεξεργαστή ο οποίος να μπορεί να διαχειριστεί και ορισμένες περιπτώσεις από Data Hazards με τις μεθόδους forwarding και stall.

Προεργασία

Σαν παραδοτέα/προεργασία του εργαστηρίου ζητήθηκε τόσο ο πηγαίος κώδικας όλων των καινούργιων components που υλοποιήθηκαν όσο και η υλοποίηση ενός σχηματικού διαγράμματος του ολοκληρωμένου datapath με όλες τις συνδέσεις των καινούργιων components. Ακόμα μας ζητήθηκε η προσομοίωση συγκεκριμένων σημάτων που βασίστηκε σε ορισμένα αρχεία εισόδου.

Περιγραφή

Όπως προαναφέρθηκε και απο τον σκοπό της εργαστηριακής άσκησης σκοπός μας ήταν να υλοποιήσουμε έναν pipeline επεξεργαστή. Για να πετύχουμε αυτό οφείλαμε να πραγματοποιήσουμε αρκετές νέες λειτουργίες και modules σε σύγκριση με το προηγούμενο εργαστήριο μιας και ο multicycle επεξεργαστής πραγματοποιούνταν με μια διαφορετική λογική.

Πιο συγκεκριμένα, προκειμένου να πετύχουμε την δομή ενός pipeline επεξεργαστή, δημιουργήσαμε 4 καινούργιους registers μεταξύ όλων των σταδίων με αντίστοιχες ονομασίες IF_ID Register, ID_EX Register, EX_MEM Register, MEM_WB Register στους οποίους αποθηκεύαμε πληροφορία από το προηγούμενο στάδιο για να αξιοποιηθεί στο επόμενο. Όλα τα σήματα που είναι χρήσιμα για επόμενα στάδια περνούν από τα στάδια που χρειάζονται προκειμένου να διατελέσουν την εκάστοτε λειτουργία τους.

Να σημειώσουμε επίσης ότι διαφοροποιήσαμε σε αρκετά μεγάλο βαθμό το control unit αφαιρώντας την fsm που χρησιμοποιήσαμε στην περίπτωση του multicycle επεξεργαστή, ενώ προσθέσαμε τα απαραίτητα σήματα εισόδου και εξόδου προκειμένου να λειτουργεί με σωστό τρόπο ο pipeline επεξεργαστής.

Σαν απαραίτητη λειτουργικότητα του εργαστηρίου, έπρεπε να σχεδιάσουμε έναν επεξεργαστή pipeline, ικανός να υλοποιεί τις εντολές: add, li, sw, lw

Επιπρόσθετα, γνωρίζουμε από την θεωρία του μαθήματος η σχεδίαση ενός τέτοιου επεξεργαστή συνοδεύεται από μια σειρά από hazards τα οποία ως σχεδιαστές οφείλουμε να έχουμε συνυπολογίσει κατά την σχεδίαση μας. Παράδειγμα τέτοιων hazards, είναι τα data hazard και structural hazards. Προκειμένου, να επιτύχουμε την αντιμετώπιση τους σχεδιάσαμε δύο καινούργια components στο σύστημα μας, τα οποία παρουσιάζονται παρακάτω

Forward Unit

Το συγκεκριμένο module, συναντάται σε περιπτώσεις που κάποια από τις επόμενες εντολές χρειάζεται πληροφορία που υπολογίστηκε σε κάποια προηγούμενη κατάσταση με αποτέλεσμα η μνήμη να μην έχει ενημερωθεί με την καινούργια τιμή και κάποια από τις επόμενες εντολές λάβει λανθασμένη τιμή. Η λογική του forwarding που πραγματοποιείται στο module αυτό, δίνει την δυνατότητα στον επεξεργαστή, μόλις αποθηκεύει την καινούργια τιμή να μπορεί να την δώσει αν χρειαστεί σε επόμενη εντολή.

Στα πλαίσια δημιουργίας του συγκεκριμένου module, προσθέσαμε πολυπλέκτες στην είσοδο της ALU, οι οποίοι αποφασίζουν αν τα δεδομένα έρχονται από την κανονική ροή ή από το forward. Ο έλεγχος πραγματοποιείται στο component ForwardUnit, το οποίο παράγει δύο σήματα εξόδου 2 bit, Control_A και Control_B τα οποία ελέγχουν ανάλογα με την περίπτωση τις εισόδους της ALU.

Stall Unit

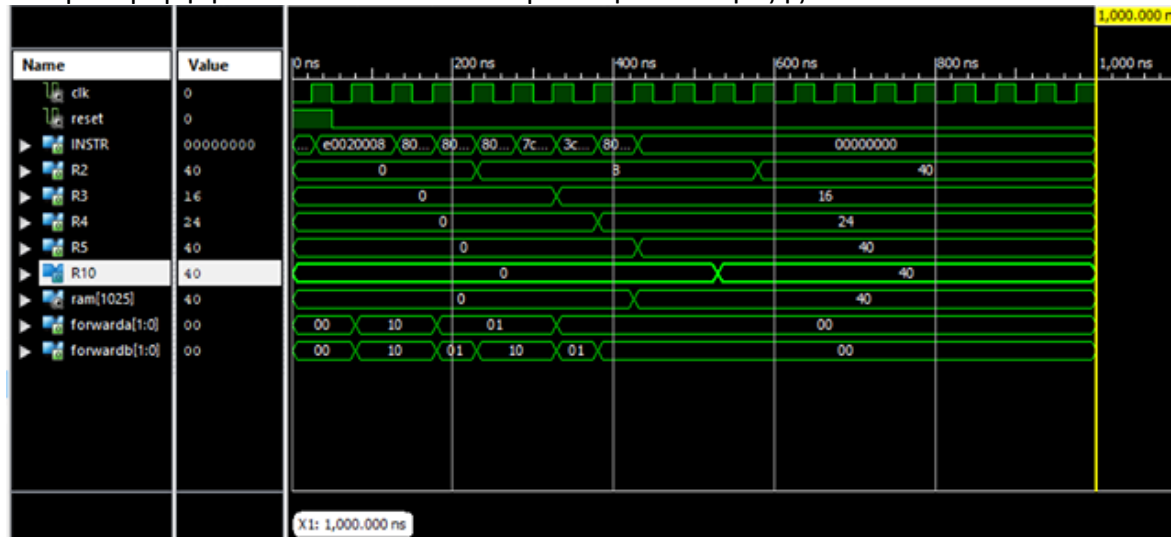
Σε αυτό το module, αντιμετωπίζεται hazard που συναντάται κατά τις εντολές lw, lb (μας ενδιαφέρει η lw στο συγκεκριμένο εργαστήριο). Το πρόβλημα που ενδέχεται να συναντηθεί είναι όταν προηγούμενη εντολή είναι η lw και ο καταχωρητής λάβει κάποια τιμή από την μνήμη να χρησιμοποιείται από επόμενη εντολή αφού δεν μπορούμε να εγγυηθούμε ότι η επόμενη εντολή θα λάβει την τιμή που πρέπει. Σύμφωνα και με την θεωρία του μαθήματος ο πιο απλός τρόπος υλοποίησης είναι με την εισαγωγή κατάστασης που δεν πραγματοποιείται κάποια λειτουργία. Αυτή τη μέθοδο, την γνωρίζουμε και ως bubble.

Για την επίτευξη του συγκεκριμένου αποτελέσματος, αναγκαζόμαστε στιγμιαία να διακόψουμε την ροή του pipeline, κόβοντας το σήμα write enable του καταχωρητή, IF_ID Register. Αφού ο καταχωρητής, ενημερωθεί επιτυχώς ενεργοποιούμε τον καταχωρητή PC_LDEn να κάνει fetch καινούργια εντολή, ενώ ενεργοποιείται και πάλι το write enable με την ροή να συνεχίζεται κανονικά

Κυματομορφές-Προσομοίωση

Forward Unit

Η κυματομορφή του forward unit που προεκυψε είναι η εξής



Οι εντολές που πραγματοποιήθηκαν κατά σειρά είναι

- li r2, 8
- add r3,r2,r2
- add r4, r3, r2
- add r5, r4, r3
- sw r5, 4(r0)
- lw r10, -4(r2)
- add r2, r4,r3

Το παραπάνω παράδειγμα λειτουργείας υλοποιήθηκε, με τα πρότυπα των διαλέξεων όπου συναντάται η περίπτωση του forwarding και μια αλληλουχία των απαιτούμενων πράξεων που δείχνουν την σωστή λειτουργία του. Παρατηρούμε επίσης ότι η κάθε εντολή πραγματοποιείται 5 κύκλους με αρχή το fetch της εντολής δείχνοντας ότι διέρχεται κάθε φορά από όλους τους καταχωρητές.