

Αναφορά 2ης Εργασίας

Δημιουργία επεξεργαστή πολλών κύκλων

Μπέτσιος Θωμάς

A.M. 2013030104

Σκοπός της εργαστηριακής άσκησης

Σκοπός της εργαστηριακής άσκησης είναι η συνέχεια της 1ης εργαστηριακής άσκησης με την ολοκλήρωση ενός επεξεργαστή πολλών κύκλων. Για το σκοπό αυτό σχεδίασα αρχικά τις αλλαγές στο Datapath και μετά προχώρησα στην μονάδα ελέγχου (Control).

Περιγραφή

Αρχικά θα ήθελα να επισημάνω ότι δεν κατάφερα να διορθώσω το 1^ο μέρος με τον επεξεργαστή ενός κύκλου, το οποίο στο test του Datapath μου έβγαζε ένα σήμα με την τιμή 'XXXX' ενώ τα test της κάθε μίας βαθμίδας έβγαζαν σωστά αποτελέσματα. Προσπάθησα αρκετά και ξόδεψα χρόνο αλλά τελικά δεν κατάφερα να βρω και να διορθώσω το λάθος. Πολύ πιθανό είναι να έχει ξεφύγει καταλάθος κάποιο σήμα στο port map του Datapath, το οποίο επιρρεάζει αρκετά.

Για την 4^η φάση:

Για την υλοποίηση της 4^{ης} φάσης και την μετατροπή του επεξεργαστή ενός κύκλου σε multi cycle processor χρειάστηκε να εισάγω ανάμεσα στις ξεχωριστές βαθμίδες του Datapath καταχωρητές που θα καθήστερουν την έξοδο για έναν κύκλο ρολογιού πριν γίνει είσοδος στην επόμενη. Επίσης χρειάστηκε να υλοποιήσω και 4 πολυπλέκτες (δύο των 4 bit και δύο των 2 bit) και να τους συνδέσω στο port map του Datapath, έναν για την επιλογή **instruction ή data** ανάμεσα στον Program Counter και την μνήμη (2 bit), έναν για την επιλογή **program Counter ή καταχωρητή A** της πρώτης εισόδου της ALU (2 bit), έναν για την **επιλογή της δεύτερης εισόδου της ALU**, (4 bit) τον οποίο τον υλοποίησα μέσα στο ExStage αντικαθιστώντας τον προηγούμενο Mux (2 bit) και τέλος έναν Mux (4 bit) για την επιλογή του σωστού Program Counter που πλέον ο υπολογισμός του γίνεται μέσα στην ALU.

Στην συνέχεια προσέθεσα τα νέα σήματα ελέγχου (IRWrite, IorD, ALU_Ain_Sel, PC_Source και τροποποίησα το ALU_Bin_Sel από 1 σε 2 bit) για τους καταχωρητές και πολυπλέκτες που χρειάζονταν στο Control path το οποίο το είχα ήδη υλοποιήσει με FSM από την προηγούμενη εργασία.

Συμπεράσματα

Σ' αυτήν την εργασία μάθαμε πως λειτουργεί ένας επεξεργαστής πολλών κύκλων και το πώς βελτιώνει την γενικότερη απόδοση του επεξεργαστή μας, γιατί πλέον ο κύκλος του ρολογιού δεν κρατά όσο η πιο αργή εντολή (αυτή με το critical path), η κάθε βαθμίδα βγάζει την έξοδο της και στον επόμενο κύκλο γίνεται είσοδος στην επόμενη μικραίνοντας έτσι τον κύκλο του ρολογιού και αυξάνοντας την απόδοση.

Για την 5^η φάση: Έκανα μια προσπάθεια κατανόησης της θεωρίας και υλοποίησης αλλά επειδή δεν προλάβαινα και δεν βγήκε και κάποιο σημαντικό αποτέλεσμα δεν έστειλα κώδικα για αυτήν την φάση