



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΩΝ & ΥΛΙΚΟΥ
ΕΡΓΑΣΤΗΡΙΑΚΕΣ ΑΣΚΗΣΕΙΣ ΓΙΑ ΤΟ ΜΑΘΗΜΑ:
Οργάνωση Υπολογιστών
HPY 302
<http://www.mhl.tuc.gr>
ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2024

Εργαστηριακή Άσκηση 3

Αριθμός Ομάδας: 56

Ονοματεπώνυμο 1: Χαϊντούντη Μαρία

Ονοματεπώνυμο 2: Χειλαδάκης Νικόλαος

Σκοπός της άσκησης: Ο σκοπός της άσκησης αυτής είναι Μετατροπή του επεξεργαστή ενός κύκλου σε επεξεργαστή pipeline.

Περιγραφή της Συγδίασης: Για την υλοποίηση της φάσης αυτής χρησιμοποιήθηκε η μονάδα ελέγχου του επεξεργαστή ενός κύκλου ως σημείο εκκίνησης, όπως προτείνεται από την εκφώνηση. Κυρίως αλλαγές έγιναν στο κομμάτι του «Datapath».

Η βασική δομή παραμένει ίδια, δηλαδή το στάδιο IF είναι υπεύθυνο για την αναγνώριση των εντολών από τη μνήμη με βάση την τρέχουσα τιμή του «Program Counter», το DEC στάδιο αποκωδικοποιεί την εντολή και διαβάζει τις απαιτούμενες τιμές από τους καταχωρητές, στο EXEC η ALU πραγματοποιεί τις αριθμητικές και λογικές πράξεις και στο MEM στάδιο εκτελείται η προσπέλαση μνήμης για φόρτωση(load) ή αποθήκευση δεδομένων(store). Τέλος, γίνεται WriteBack, όπου τα αποτελέσματα επεξεργασίας είτε από το στάδιο EXEC, είτε από το στάδιο MEM γράφονται στους καταχωρητές μέσω του πολυπλέκτη «tux2to1».

Για την υλοποίηση του pipeline επεξεργαστή, όμως, δημιουργήθηκαν νέα modules απαραίτητα για την ορθή του λειτουργία. Πιο συγκεκριμένα, δημιουργήθηκαν τρία module που το καθένα απεικονίζει και ένα διαφορετικό στάδιο του pipeline. Ο «DEC_EX_pipeline» αποθηκεύει τα δεδομένα εντός του σταδίου και εξασφαλίζει ότι θα παραμείνουν σταθερά κατά τη διάρκεια του pipeline. Ο «EX_MEMORY_pipeline» είναι υπεύθυνος για την αποθήκευση και προώθηση σημαντικών ελέγχων και δεδομένων που χρειάζονται για την εκτέλεση και πρόσβαση στη μνήμη κατά την εκτέλεση των εντολών και τέλος ο «MEM_WB_pipeline» αναλαμβάνει την εκτέλεση των δραστηριοτήτων εγγραφής. Δημιουργήθηκε, ακόμα, ένας «register15bit», ο οποίος επιτρέπει τη διαχείριση των 15-bit σημάτων, όπως κωδικών εγγραφέων, και χρησιμοποιείται για να διατηρεί τα σημαντικά δεδομένα κατά τη μετάδοση τους μέσω των διάφορων σταδίων του Datapath. Ο πολυπλέκτης «tux3to1» χρησιμοποιείται για την επιλογή της εισόδου της ALU, ανάλογα με τα σήματα ελέγχου από τη μονάδα «Forwarding», στην οποία πότε εμφανίζονται καθυστερήσεις στην παροχή δεδομένων

από το στάδιο MEM και χρησιμοποιεί τις τεχνικές προώθησης για να διορθώσει αυτές τις καθυστερήσεις. Προστέθηκαν, επιπλέον, τέσσερις καταχωρητές «Dec_rd_rs_rt_reg», «Exec_rd_rs_rt_reg», «Mem_rd_rs_rt_reg», «Exec_Mem_datain_reg», οι οποίοι πρώτοι τρείς αποθηκεύουν τα δεδομένα των καταχωρητών από το αντίστοιχο στάδιο rd, rs, rt για να εξασφαλιστεί ότι οι πράξεις θα γίνουν μεταξύ των σωστών καταχωρητών και όχι μεταξύ άλλων που μπορεί να έρθουν από άλλους κύκλος και ο τέταρτος αποθηκεύει την είσοδο της μνήμης, για να την καθυστερήσει για έναν κύκλο προκειμένου να επιτευχθεί σωστός συγχρονισμός. Τέλος, έγινε η υλοποίηση του «Hazard Detection Unit» που εντοπίζει και διαχειρίζεται τα hazards, τα οποία είναι προβλήματα που προκύπτουν, όταν οι εντολές στο pipeline έχουν αλληλεξαρτήσεις. Εάν ανιχνευθεί κάποιο πρόβλημα, τότε όταν είναι μηδέν τα σήματα ελέγχου «PC_LdEn», «IF_ID_Write», «control_signal», παγώνει το πρόγραμμα έως ότου λυθεί ο κίνδυνος. Για το «πάγωμα» της λειτουργίας του προγράμματος χρησιμοποιούνται «stalls», για τα οποία ο πολυπλέκτης «mux_stalling» επιλέγει είτε να σταματήσει το pipelining διαλέγοντας μια «stalled» κατάσταση, είτε επιτρέπει την κανονική λειτουργία επιλέγοντας τα πραγματικά σήματα ελέγχου που έρχονται από το Control.

Παρακάτω ακολουθεί απεικόνιση του συνολικού συστήματος με την σύνδεση του datapath με το control unit:

