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

Αναφορά Εργαστηριακής Άσκησης 3

Αριθμός Ομάδας:66

Λαμπράκης Μιχάλης 2020030077
Δήμας Χρήστος 2021030183

1.Σκοπός της Άσκησης

Σκοπός της 3ης εργαστηριακής άσκησης είναι η σχεδίαση ενός επεξεργαστή pipeline. Σαν βάση για την σχεδίαση χρησιμοποιήθηκε ο multi-cycle επεξεργαστής που σχεδιάστηκε στην προηγούμενη εργασία και προστέθηκαν σε αυτόν καταχωρητές για τα ενδοιάμεσα στάδια καθώς και τα σήματα ελέγχου (control). Επιπλέον σχεδιάστηκαν τα κατάλληλα units για την αντιμετώπιση των datahazards με forwarding και με stalls.

2.Περιγραφή της Σχεδίασης

Το κύριο χαρακτηριστικό του pipeline επεξεργαστή είναι ότι εκτελεί πολλές εντολές ταυτόχρονα σε έναν κύκλο του ρολογιού, σε αντίθεση με τους single και multi cycle επεξεργαστές οπου για να ξεκινήσει μια νέα εντολή πρέπει πρώτα να έχει ολοκληρωθεί η τρέχουσα.

Η αρχή της λειτουργίας του pipeline επεξεργαστή μοιάζει πολύ με αυτή του multi-cycle ως προς το ότι βασίζεται στον διαχορισμό της εκτέλεσης σε στάδια τα όποία είναι όμοια με τα στάδια του multi-cycle:

IF_stage / DEC_stage / EXEC_stage / MEM_stage / WB_stage

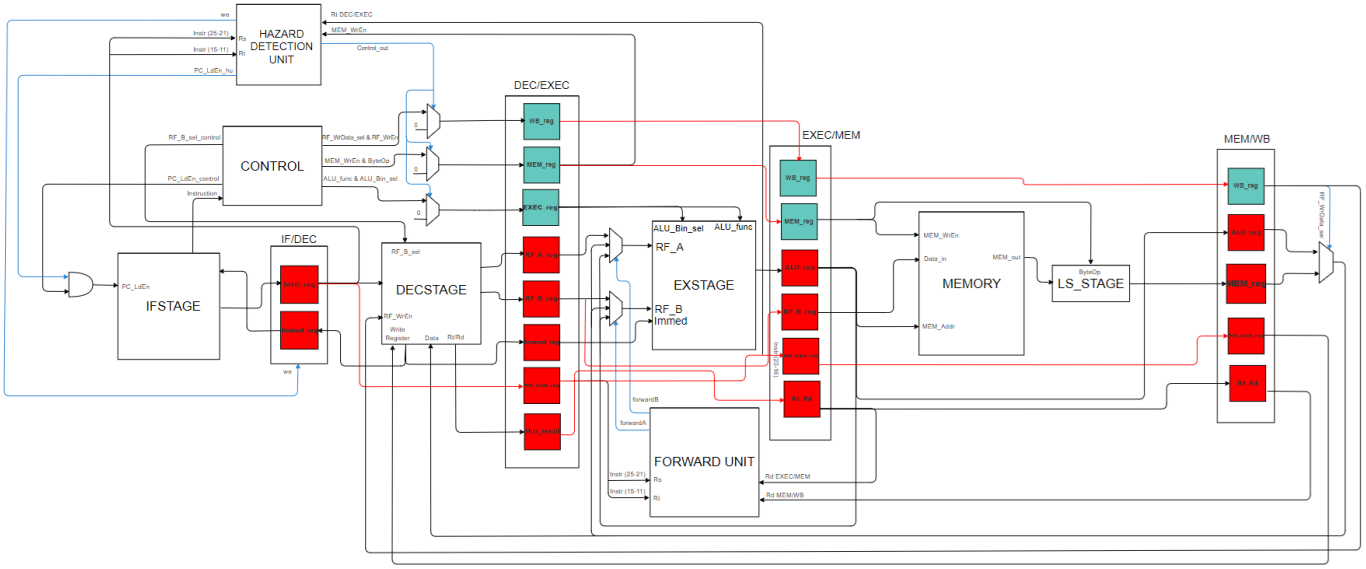
Για να γίνει το παραπάνω, τοποθετήθηκαν καταχωρητές μεταξύ των 5 σταδίων που αποθηκεύουν όλες τις εξόδους που παράγονται από το κάθε στάδιο και περνάνε σαν εισόδους στο επόμενο. Ακόμη, τοποθετήθηκαν καταχωρητές και για τα control σήματα ώστε να μη χάνεται η πληροφορία που χρειάζονται οι εντολές για να λειτουργήσουν όταν περνάνε από το ένα στάδιο στο επόμενο. Όλοι οι απαραίτητοι καταχωρητές έχουν ομαδοποιθεί στα εξής modules (για λόγους ιεραρχικής σχεδίασης):

IF_DEC / DEC_EX / EX_MEM / MEM_WB

Για την αντιμετώπιση των datahazards δημιουργήθηκαν το **Forwarding Unit** και το **Hazard Detection Unit**. Στην περίπτωση του Forwarding Unit, hazard μπορεί να προκύψει όταν η τιμή ενός καταχωρητή του Register File χρειαστεί σε μια επόμενη εντολή (αφού οι εντολές εκτελούνται ταυτόχρονα, δε θα έχουν προλάβει να ολοκληρωθούν οι 5 κύκλοι εκτέλεσης). Αφού πργαμοτοποιηθούν (σύμφωνα με την θεωρία) οι απαραίτητοι έλεγχοι και επιβεβαιωθεί ότι υπάρχει hazard, δίνουμε την κατάλληλη εισόδο στον Register File χωρίς να περιμένουμε αυτή να βγει από το WB_stage.

Στην περίπτωση του **Hazard Detection Unit**, επιτυγχάνεται η αντιμετώπιση των datahazards μέσω stalls. Σε αυτή την περίπτωση, όταν ένας καταχωρητης παίρνει τιμή από την μνήμη, πρέπει αναγκαστικά να περάσει και από τα 5 στάδια εκτέλεσης, οπότε αμα χρησιμοποιείται από μια επόμενη εντολή, αυτή πρέπει να περιμένει. Μόλις ανισχνευθεί hazard, παγώνουν τα IF_stage και DEC_stage απενεργοποιώντας τα write enable των κατάλληλων καταχωρητών και εισάγοντας μηδενικά στην βαθμίδα σημάτων ελέγχου του DEC_EX.

Ακολουθεί Block Diagram της ολοκληρωμένης σχεδίασης:



3.Συμπεράσματα

Σε αυτή την εργασία κατασκευάστηκε ένας pipeline επεξεργαστής. Συνδυάζοντας τις 2 πρώτες εργασίες, καταλήξαμε σε έναν επεξεργαστή που εκτελεί πολλές εντολές ταυτόχρονα άρα είναι και πιο αποτελεσματικός. Η σχεδιάση του έγινε αρκετά πολύπλοκη και απαιτήθηκε πολύ καλή οργάνωση του κώδικα και συνεργασία μεταξύ μας.