

Αναφορά Εργαστηρίου 5

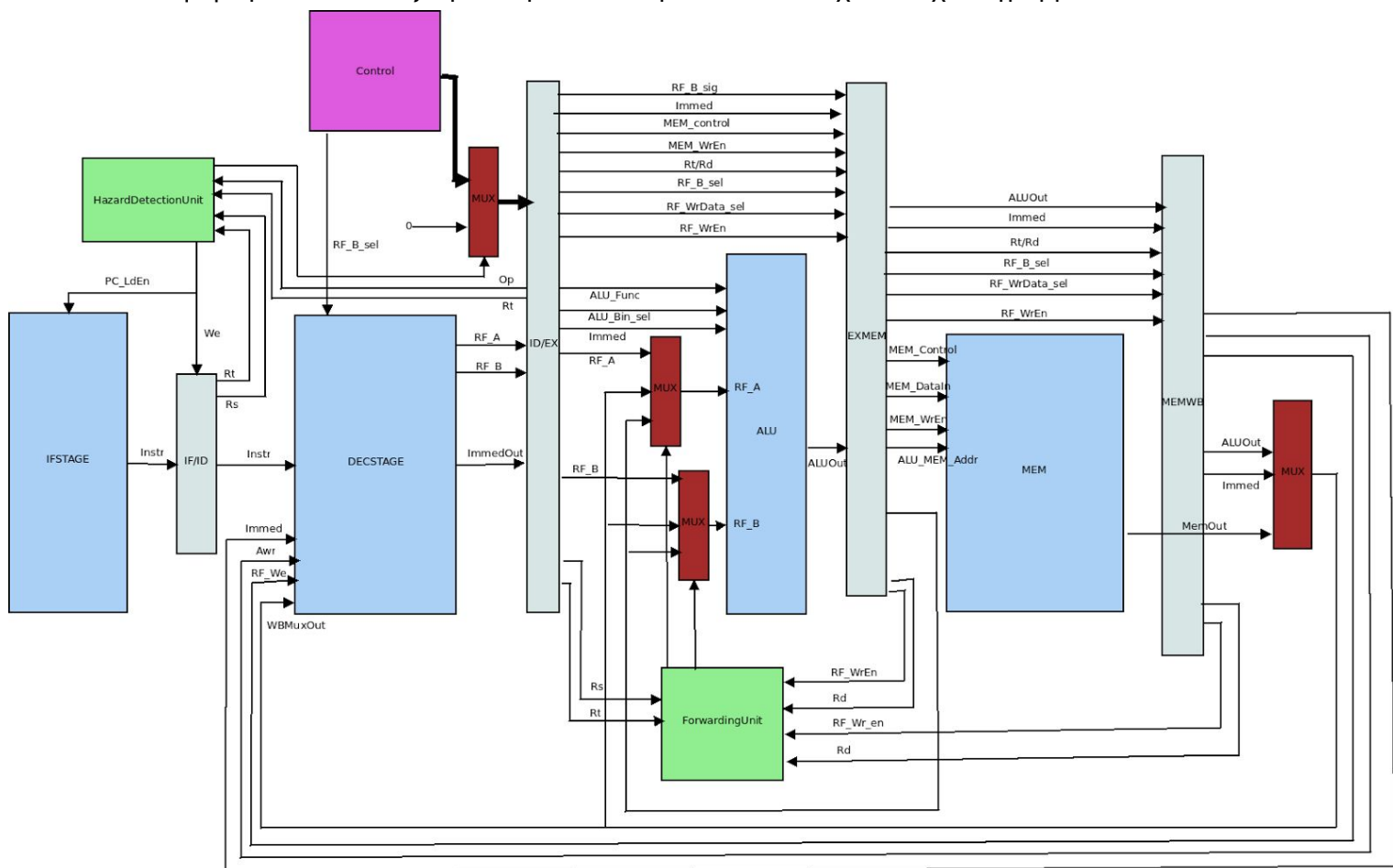
Οργάνωση Υπολογιστών

Ομάδα Εργασίας: Κριθαράκης Εμμανουήλ, Φωτάκης Τζανής

Κωδικός Ομάδας: LAB31231483

Προεργασία

Ως προεργασία του συγκεκριμένου εργαστηρίου είχε ζητηθεί όπως είναι φυσικό ο κώδικας των διαφόρων modules που περιγράφονται παρακάτω αλλά και σχεδιαγράμματα που παρουσιάζουν την συνδεσμολογία μεταξύ των διαφόρων ζητούμενων modules που κληθήκαμε να αναπτύξουμε. Παρακάτω παρατίθενται τα σχετικά σχεδιαγράμματα.



Περιγραφή της Άσκησης

Για το συγκεκριμένο εργαστήριο ζητήθηκε να σχεδιαστεί ένας pipeline processor βασισμένος στο instruction set που περιέχει τις εξής εντολές: **add,li,sw,lw**.

Στην προσπάθεια για επιτυχή λειτουργία οφείλαμε να αντιμετωπίσουμε τα διάφορα hazards της σχεδίασης αυτής όπως είναι τα structural και data hazards. Τα structural hazards στην πραγματικότητα δεν αποτέλεσαν ποτέ πρόβλημα καθώς εξ αρχής θεωρήσαμε pipeline register με 5 διαφορετικά στάδια. Συγκεκριμένα, τα στάδια είναι το IFSTAGE, DECSTAGE, ALUSTAGE, MEMSTAGE και WRITEBACK για κάθε εντολή. Απόκοτο αυτής της υλοποίησης είναι οι πόροι του επεξεργαστή πχ η ALU να μην χρησιμοποιείται από 2 εντολές ταυτόχρονα. Οφείλουμε να σημειώσουμε ότι τα control hazards δεν αποτέλεσαν ζήτημα για το εν λόγω instruction set καθώς η θεώρηση ότι αποφασίζω στο writeback stage δεν προκάλεσε παραβίαση των hazards αυτών. Συνεπώς, τα μόνα hazards που έπρεπε να αντιμετωπίσουμε ήταν τα data hazards (κίνδυνοι χρήσης σημάτων πριν είναι έτοιμα). Για την επίλυση των κινδύνων αυτών χρησιμοποιήθηκαν οι τεχνικές του forwarding και του stall που αναφέρονται παραπάνω.

Η δομή ενός pipeline processor αποτελείται από τα 5 στάδια που αναφέρθηκαν παραπάνω με μια σημαντική διαφοροποίηση από προηγούμενους processors που έχουμε υλοποιήσει στο παρελθόν (πχ multicycle processor). Ανάμεσα στα διάφορα στάδια υπάρχουν καταχωρητές ονόματι pipeline registers, οι οποίοι είναι απαραίτητοι στην μεταβίβαση των control signals κάθε εντολής από επίπεδο σε επίπεδο. Ανάλογα το επίπεδο εισόδου και εξόδου ονομάζονται κατάλληλα. Συγκεκριμένα οι καταχωρητές είναι οι IFID, IDEX, EXMEM ΚΑΙ MEMWB αντίστοιχα. Το control σε σύγκριση με έναν multicycle processor είναι σε μεγάλο βαθμό διαφορετικό. Δεν αποτελείται πλέον από μια fsm που παράγει ανάλογα το stage της και κατάλληλα σήματα αλλά παρέχει ότι control signals χρειάζεται μια εντολή (συνεπώς δέχεται ως input το Instr) έπειτα από το fetching από το rom.data που δίνεται. Τα σήματα αυτά εισάγονται στους pipeline registers και σε κάθε κύκλο μέχρι την ολοκλήρωση της εντολής ανάλογα το stage που βρίσκεται η εντολή χρησιμοποιεί τα signals που χρειάζεται. Στο τελευταίο stage χρησιμοποιούνται και τα τελευταία signals. Η πορεία των signals φαίνεται ξεκάθαρα στο σχήμα του processor. Όπως αναφέρθηκε παραπάνω για την αντιμετώπιση των data hazards χρειάστηκαν κάποια units στο datapath του processor. Αυτά παρουσιάζονται παρακάτω.

Forwarding unit

Υπάρχουν φορές που κατά την υλοποίηση ενός rom.data μέσω pipeline register κάποια από τις επόμενες εντολές να χρειαστεί πληροφορία που εκτιμήθηκε υπολογίστηκε και αποθηκεύτηκε σε προηγούμενη κατάσταση. Ένας από τους κινδύνους αποτελεί να μην έχει προλάβει να γραφτεί το αποτέλεσμα από το writeback stage στους καταχωρητές και επόμενες εντολές να λάβουν την παλιά λανθασμένη τιμή. Το πρόβλημα αυτό το λύνει το forwarding με το σκεπτικό ότι μόλις έχω γνώση της πληροφορίας που θα αποθηκευτεί στο writeback stage αν μια επόμενη εντολή την χρειαστεί να μπορεί ο processor να την δώσει. Το forwarding εμφανίζεται τόσο σε 2 καταστάσεις, ενός ή δύο κύκλων. Συγκεκριμένα στο forwarding one cycle αυτό που γίνεται είναι να εισάγονται στο forwarding unit η διεύθυνση του καταχωρητή Rd και το Rf_wr_en από τον pipeline register EXMEM καθώς και οι αντίστοιχες IDEX_rt και IDEX_rs και εφόσον είναι ίδιες με μια εξ αυτών τότε σημαίνει ότι

θέλω την νέα τιμή που υπάρχει στο EXMEM_rd να εισαχθεί είτε στον πρώτο τελεστή της alu είτε στον δεύτερο. Για το forwarding two cycles η λογική είναι ίδια με την διαφορά ότι η τιμή του καταχωρητή rd λαμβάνεται από τον επόμενο pipeline register τον MEMWB(όπως φυσικά και το Rf_wr_en).

HazardDetection unit

Το συγκεκριμένο unit είναι χρήσιμο για την υλοποίηση του stall. Το stall είναι απαραίτητο για την αντιμετώπιση των data hazards 3 cycles. Συγκεκριμένα, χρησιμοποιείται όταν η προηγούμενη εντολή είναι η lw (στο υπάρχων instruction set) και ο καταχωρητής που λαμβάνει την τιμή από την memory χρησιμοποιείται από την επόμενη εντολή. Πρέπει να είμαστε σε θέση να εγγυηθούμε ότι η επόμενη εντολή θα λάβει την σωστή τιμή. Άρα με το unit αυτό φορτώνουμε την επόμενη εντολή της στον register IFID και κόβουμε το write_enable του με σκοπό να δημιουργηθούν bubbles στην διαδοχή των εντολών έτσι ώστε να διαβεβαιωθούμε ότι ο καταχωρητής έλαβε την σωστή τιμή για να χρησιμοποιήσει η επόμενη εντολή της lw. Αφού λάβει την τιμή ο register αυτός ενεργοποιούμε το PC_Ld_en να συνεχίσει να φέρνει εντολές που θα γράφονται στον IFID pipeline register του οποίου ενεργοποιείται ξανά το write enable για να δέχεται καινούργιες εντολές.

Πολλές φορές υπάρχει η ανάγκη για τον συνδυασμό των forwarding και stall, πράγμα που υλοποιείται απλά κατά την ταυτόχρονη λειτουργία τους αφού τα δύο modules είναι ανεξάρτητα μεταξύ τους.

Το bonus μέρος του εργαστηρίου υλοποιήθηκε, δυστυχώς, κατά το ήμισυ αναπτύσσοντας τα σήματα του controller για όλες τις εντολές του CHARIS 4 εκτός από αυτές των branch.

Κυματομορφές

