

Αναφορά Εργαστηρίου 5

Κωδικός Ομάδας **LAB31231465**

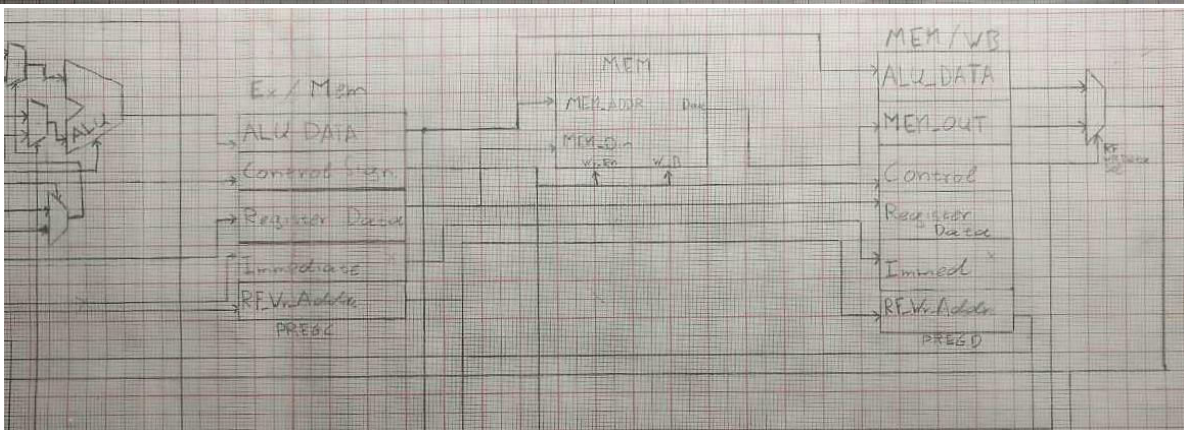
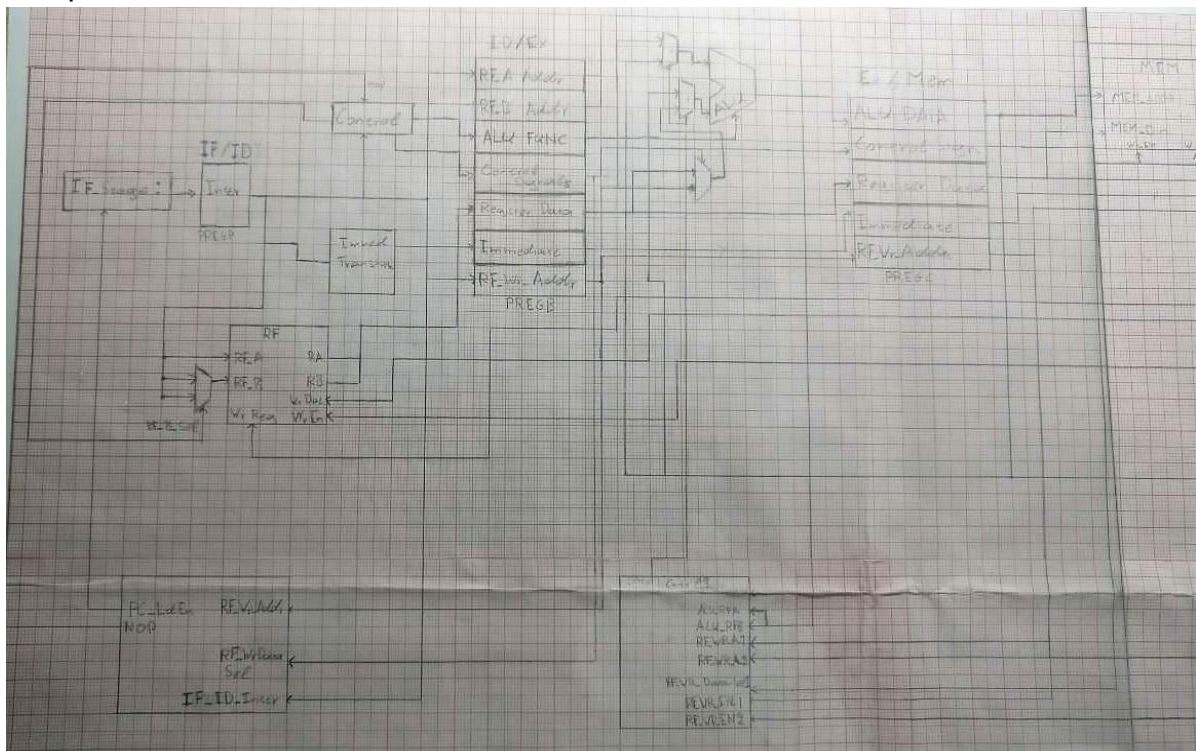
Διαλεκτάκης Γιώργος

Βαϊλάκης Απόστολος Νικόλαος

Προεργασία

Ως προεργασία του 5ου εργαστηρίου ζητήθηκε ένα σχηματικό διάγραμμα του Datapath του Pipeline επεξεργαστή που υλοποιήσαμε καθώς και ο κώδικας σε VHDL για την υλοποίησή του.

Datapath :



Περιγραφή Άσκησης

Σκοπός της 5ης εργαστηριακής άσκησης ήταν να υλοποιήσουμε έναν pipeline επεξεργαστή ο οποίος να χειρίζεται Data Hazards. (Αυτό επιτυγχάνεται είτε με forwarding είτε με stall.) Για να το πετύχουμε αυτό, χρησιμοποιήσαμε το 3ο εργαστήριο, αφαιρώντας την μονάδα ελέγχου και προσθέτοντας μία καινούργια που να αποσκοπεί στη σωστή λειτουργία του Pipeline επεξεργαστή. Επίσης προσθέσαμε pipeline καταχωρητές έτσι ώστε η έξοδος της κάθε βαθμίδας να γίνεται είσοδος στην επόμενη. Πιο συγκεκριμένα δημιουργήσαμε τους **καταχωρητές IF/ID, ID/EX, EX/MEM και MEM/WB** στους οποίους αποθηκεύαμε πληροφορία από το προηγούμενο Stage με σκοπό να την χρησιμοποιήσουμε σε επόμενο.

Η νέα μονάδα ελέγχου βγάζει πλέον ως έξοδο όλα τα σήματα ελέγχου ασύγχρονα τα οποία μεταφέρονται στους pipeline registers οι οποίοι τα βγάζουν σύγχρονα (αφού είναι registers) και έτσι κάθε σήμα χρησιμοποιείται στον κατάλληλο κύκλο, σε κάθε stage.

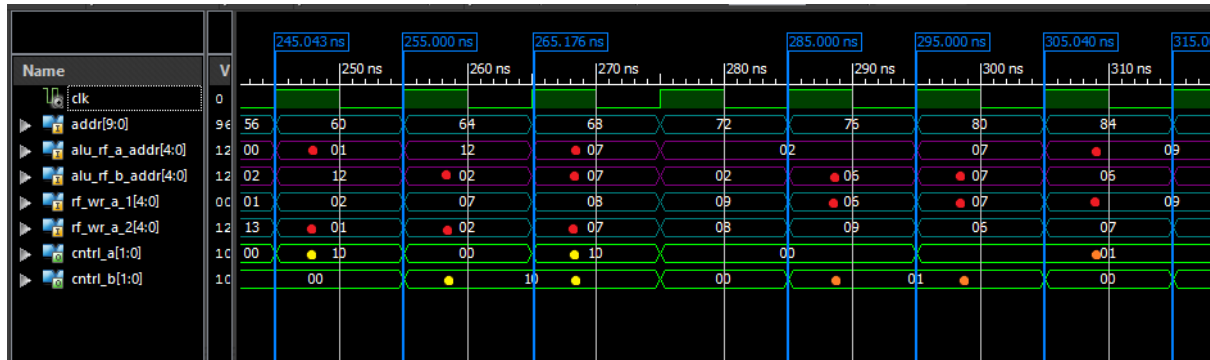
Ένας από τους τρόπους αντιμετώπισης των **Data Hazards** είναι το **forwarding**.

Για το σκοπό αυτό προσθέσαμε **2 πολυπλέκτες** στην είσοδο της ALU οι οποίοι επιλέγουν αν τα δεδομένα που εισάγουμε σε αυτήν έρχονται από την κανονική ροή δεδομένων ή από forward δηλαδή από τον καταχωρητή **EX/MEM** ή **MEM/WB**. Για τον έλεγχο αυτό δημιουργήσαμε ένα module το **FWD_CONTROL** το οποίο προβλέπει αν χρειάζεται να υπάρξει προώθηση δεδομένων. Έχει 2 σήματα εξόδου, το **Control_A** και το **Control_B** τα οποία ελέγχουν την πρώτη και τη δεύτερη είσοδο της ALU αντίστοιχα. Το **Control_A** γίνεται 01 όταν ο **MEM/WB.Rd** ισούται με τον **ID/EX.Rs** και έχουμε Write Enable ενεργό ενώ το **Control_B** γίνεται 01 όταν μιλάμε για τον **Rt** αντί για τον **Rs** κι έτσι **έχουμε προώθηση ενός κύκλου**. Το **Control_A** γίνεται 10 όταν ο **EX/MEM.Rd** ισούται με τον **ID/EX.Rs** και έχουμε Write Enable ενεργό. Αντίστοιχα για τον **Rt** και **Control_B**. Έτσι μιλάμε για **προώθηση δύο κύκλων**. Ανάλογα με κάθε σήμα ελέγχου τα δεδομένα που έρχονται στις εισόδους της ALU μπορούν να προέρχονται από το **ID/EX register (no forwarding)**, από το **EX/MEM register (single cycle forwarding)** ή από το **MEM/WB register (double cycle forwarding)**.

Ο άλλος τρόπος αντιμετώπισης των Data Hazards είναι το **Stall**. Στην περίπτωση που έχουμε την εντολή **lw** και ο **ID/EX.Rd** ισούται με τον **IF/ID.Rs** ή **IF/ID.Rt** τότε κάνουμε το **Pc_Ld_En = '0'** και εισάγουμε στον pipeline μία εντολή **NOP** γνωστή και ως **Bubble**. Τα παραπάνω εκτελούνται μέσω ενός module που δημιουργήσαμε, με όνομα **HDU**.

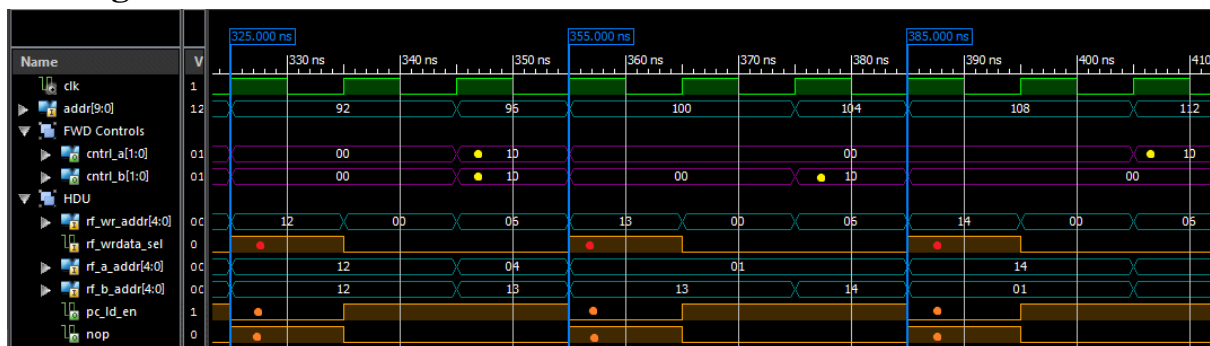
Κυματομορφές :

Forwarding



Παραπάνω βλέπουμε δύο είδη forwarding, double cycle και single cycle. Αρχικά έχουμε τρία διαδοχικά forward τυπου double cycle (κίτρινες κουκίδες). Αυτό φαίνεται απο το σήμα `cntrl` που πηγαίνει στους πολυπλέκτες που βρίσκονται μπροστά από τις εισόδους A και B της ALU. Μπορούμε ακόμα να δούμε οτι με κοκκινες κουκίδες τα σήματα σε κάθε εντολή τα οποία είναι όμοια, στα double cycle forwards οι κουκίδες αυτές βρίσκονται στο `rf_wr_a_2`, εξού και η επιλογή αυτού του είδους forwarding. Ακόμη μπορούμε να δούμε απο τις κόκκινες κουκίδες ποιές απο τις δύο εισόδους της ALU προκαλούν το forwarding, έχοντας το κατάλληλο σήμα ελέγχου στον αντίστοιχο πολυπλέκτη. Στο τρίτο μάλιστα forward βλέπουμε οτι αυτό γίνεται ταυτόχρονα και στις δύο εισόδους της ALU. Τα επόμενα forward που γίνονται είναι single cycle και χρησιμοποιούν την ίδια λογική εμφανίζοντας αυτήν την φορά διαφορετικό control όπου χρειάζεται.

Stalling



Παραπάνω βλέπουμε τρία διαδοχικά stall (πορτοκαλί κουκίδες) τα οποία και χρησιμοποιούν την ίδια λογική του single cycle forwarding ελέγχοντας ταυτόχρονα το σήμα `rf_wrdata_sel` (κοκκινες κουκίδες) το οποίο δείχνει οτι πρόκειται για πράξη μνήμης, και οτι δεν υπάρχουν ακόμα τα κατάλληλα δεδομένα. Για το stalling χρησιμοποιούνται τα δύο σήματα `pc_ld_en` και `nop`. Το πρώτο σταματάει το IFSTAGE για έναν κύκλο και το δεύτερο εμφανίζει bubble (`nop`) στο pipeline. Μετά το κάθε Stall βλέπουμε (με κίτρινες κουκίδες) και το αντίστοιχο double cycle forward το οποίο και πλέον έχει τα ζητούμενα δεδομένα απο την μνήμη.

Συμπεράσματα

Αφού φέραμε εις πέρας και το 5ο εργαστήριο κατανοήσαμε τις διαφορές που έχει ένας multi cycle επεξεργαστής με έναν pipeline επεξεργαστή κυρίως σε υλοποίηση και φυσικά σε απόδοση αφού μπορεί να εκτελεί παράλληλα έως και πέντε εντολές εξοικονομώντας έτσι αρκετό χρόνο, αφού ο τρόπος δεν μειώνει το CPI αλλά αυξάνει το throughput.