Εργασία 2

Ομάδα LAB31244236

Περάκης Γεώργιος 2016030188
Τζώρτζη Μαρία Ελένη 2016030140

Σκοπός εργαστηριακής άσκησης

Σκοπός της δεύτερης εργασίας είναι η εφαρμογή των γνώσεων για την σχεδίαση και υλοποίηση ενός επεξεργαστή πολλών κύκλων και ενός pipeline επεξεργαστή. Χρησιμοποιήθηκαν modules που κατασκευάστηκαν στην προηγούμενη εργασία καθώς και υλικό από τις διαλέξεις και το φροντιστήριο του μαθήματος.

4η Φάση

CONTROL

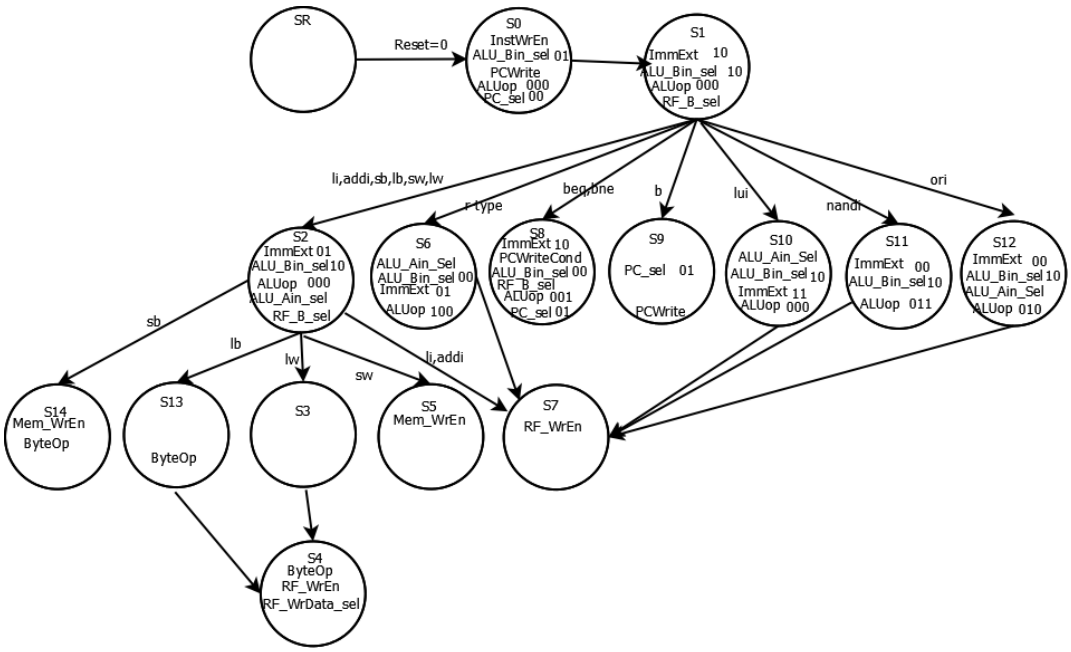
Για το control χρησιμοποιήθηκαν δυο modules, το main control και το ALU control.

Main_Control

Αυτό το module είναι σύγχρονο και αποτελείται από μια fsm.Στο main control της πρώτης εργασία προστέθηκαν επιπλέον έξοδοι για τον έλεγχο του DATAPATH. Αυτές είναι:

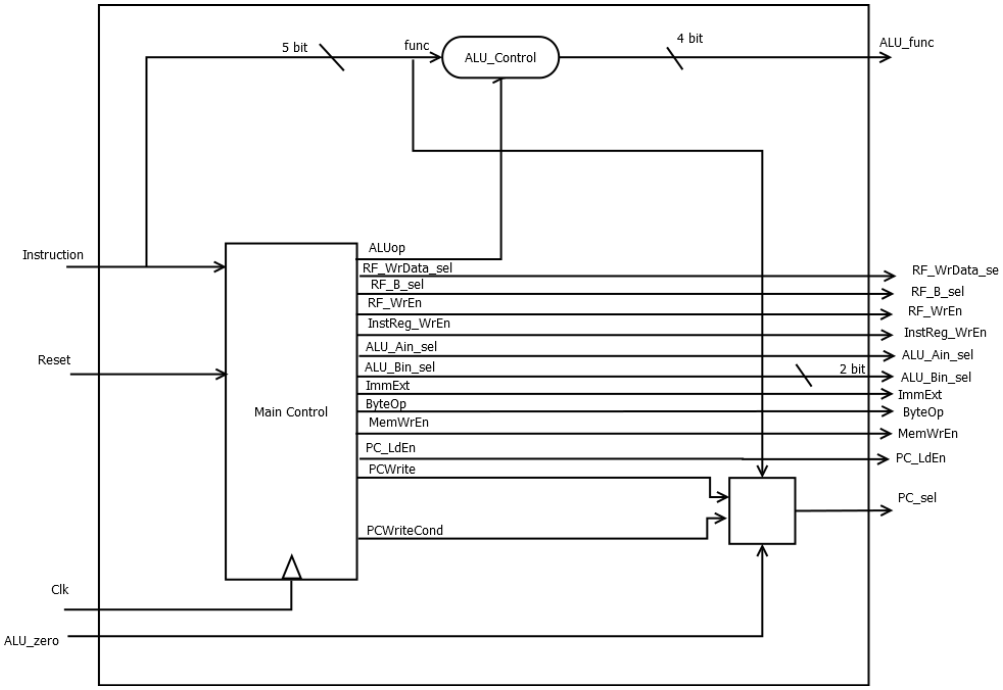
- η έξοδος για να ελέγχεται η είσοδος A της ALU (ALU_Ain_Sel) ,ώστε να μπορεί να γίνεται η επιλογή ανάμεσα στον program counter και την έξοδο A του αρχείου καταχωρητών,
- το σήμα InstReg_WrEn, που ελέγχει τον καταχωρητή που αποθηκεύει την τρέχουσα εντολή,
- τα σήματα PCWrite και PCWriteCond τα οποία χρησιμοποιούνται σε συνδυασμό με το σήμα zero της ALU για τον έλεγχο της επόμενης τιμής του Program counter (PC_sel).

Η fsm του main control αποτελείται από 16 καταστάσεις, αρχικοποιείται με Reset στην κατάσταση SR και ύστερα από ένα κύκλο ρολογιού μεταβαίνει στην κατάσταση S0. Με την ολοκλήρωση μιας εντολής, η μηχανή επιστρέφει αυτόματα στην κατάσταση S0. Με την ενεργοποίηση του Reset η μηχανή μεταβαίνει στην κατάσταση SR. Τα παραπάνω δεν φαίνονται στο σχήμα για λόγους απλούστευσης και κατανόησης. Για την κάθε εντολή αναλύθηκε το register transfer της και επομένως όλα τα σήματα που χρειάζεται για να λειτουργήσει.



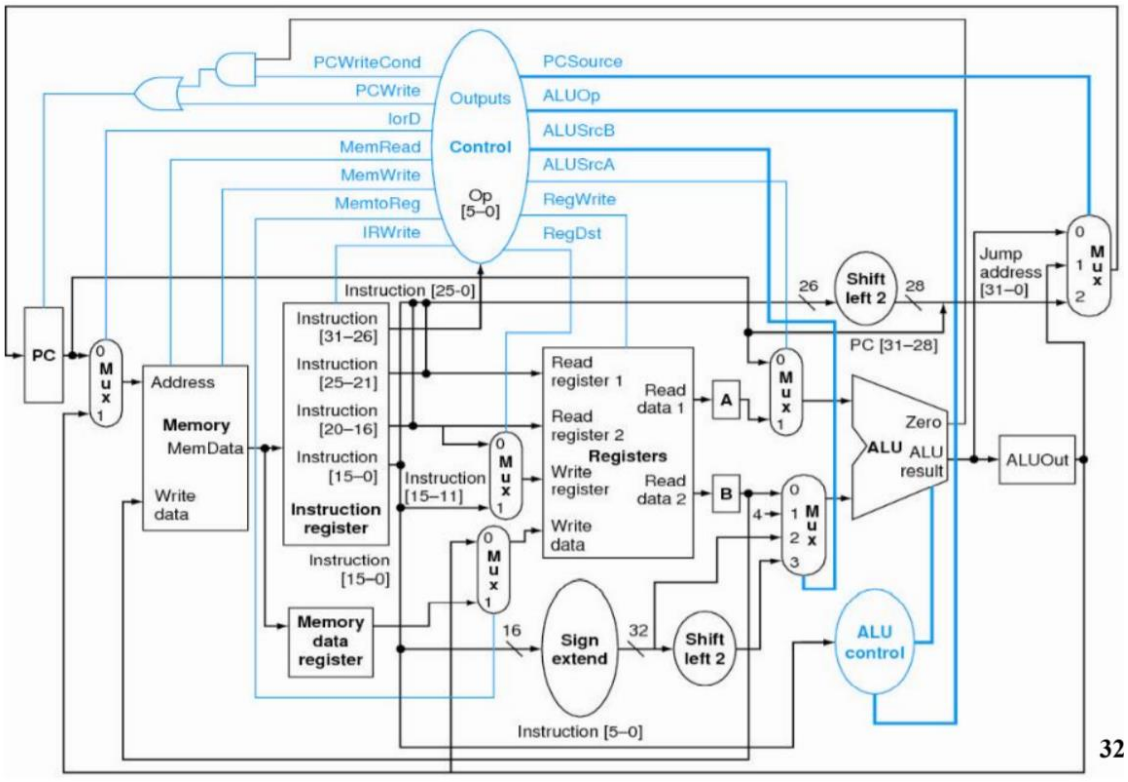
ALU_Control

Αυτό το module είναι το ίδιο που χρησιμοποιήθηκε στην προηγούμενη φάση.Ακολουθεί block diagram του control.



DATAPATH

Στο Datapath προστέθηκαν οι καταχωρητές που αποθηκεύουν: το instruction, τα δεδομένα που διαβάζονται από την μνήμη, τις δυο εισόδους και το αποτέλεσμα της ALU. Για τη δημιουργία του Datapath της εργασίας δεν δημιουργήθηκε block diagram, καθώς αυτό σχεδιάστηκε με βάση το αντίστοιχο της διάλεξης. Το μόνο διαφορετικό στην σχεδίαση μας είναι στο EXSTAGE, όπου βρίσκεται ένα module (extender_shift) .Το παραπάνω, έχει ως είσοδο όλο το instruction το οποίο ολισθαίνει αριστερά κατά 2 θέσεις τροφοδοτώντας τον πολυπλέκτη με την διεύθυνση jump.



EXSTAGE:

Οι αλλαγές που έγιναν σε αυτό το module είναι:

- η προσθήκη ενός πολυπλέκτη 2 προς 1 για την επιλογή εισόδου A προς την ALU.
- Η προσθήκη πολυπλέκτη 4 προς 1 για την επιλογή του B της ALU.Οι τρεις επιλογές του πολυπλέκτη είναι: το A από το αρχείο καταχωρητών, το 4 που χρησιμοποιείται για τον υπολογισμό του επόμενου program counter και ο immediate .

IFSTAGE

Αυτό το module αποτελείται από έναν πολυπλέκτη και τον καταχωρητή PC. Ο πολυπλέκτης επιλέγει την επόμενη τιμή του PC ανάμεσα σε τρεις πιθανές εισόδους: Το αποτέλεσμα της ALU, την έξοδο του καταχωρητή του αποτελέσματος της ALU και το jump address (έξοδος του extender_shift).

Τα modules MEMSTAGE, DECSTAGE, RAM δεν αλλάξαν για την τέταρτη φάση της εργασίας. Επίσης οι καταχωρητές του datapath που χρησιμοποιούνται για την αποθήκευση τιμών έχουν Write enable πάντα ενεργοποιημένο, εκτός του καταχωρητή για το instruction, που ελέγχεται από το control.

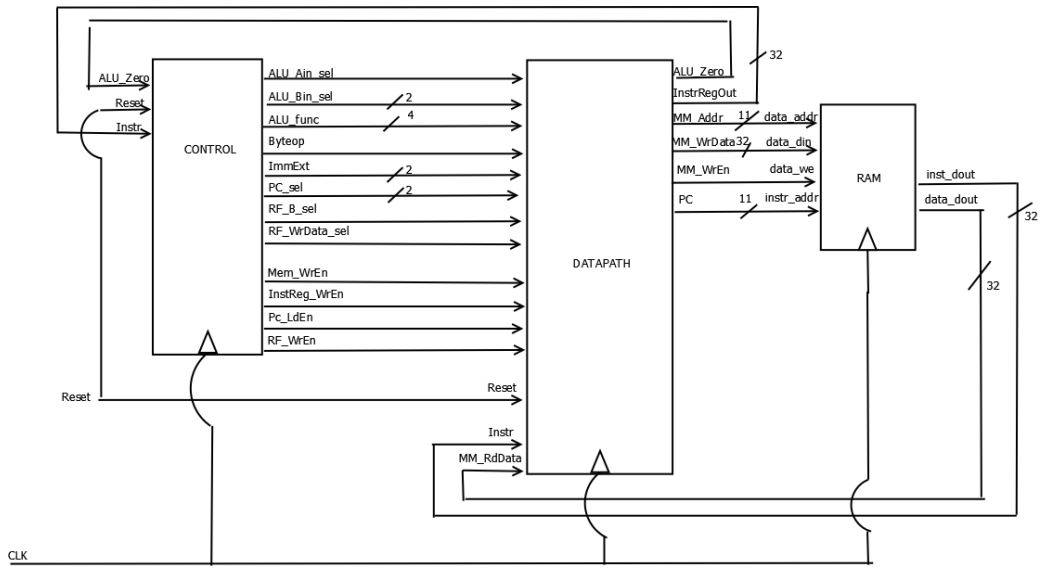
Για τον έλεγχο του Datapath_MC χρησιμοποιήθηκε το testbench Datapath_MC_test στο οποίο προσομοιώνονται οι κύκλοι της fsm με τα κατάλληλα σήματα ελέγχου.

MULTICYCLE PROCESSOR

Για την δημιουργία του multicycle processor συνενώθηκαν τα components: datapath, control και ram, στο toplevel PROCESSOR_MC. Για τον έλεγχο του processor ξεκινήσαμε με κύκλο ρολογιού 200 ns. Αφού ελέγχθηκαν όλες οι εντολές, η περίοδος ρολογιού μειώθηκε στα 22ns που είναι και το critical path του επεξεργαστή μας.

Για να ελεγχθεί η ορθή λειτουργία του επεξεργαστή, χρησιμοποιήθηκαν τρία .data αρχεία : το rep1(πρόγραμμα αναφοράς 1),το rep2(πρόγραμμα αναφοράς 2),και το rom_all_instr, το οποίο περιέχει όλα τα instructions που υποστηρίζονται από τον επεξεργαστή.

Στη συνέχεια φαίνεται το block diagram του top-level του επεξεργαστή, όπου και συνδέονται τα DATAPATH, CONTROL και RAM.



5η Φάση

CONTROL

Για το control χρησιμοποιήθηκε το ALU control από την 4η φάση.

Main_Control

Το main control βασίστηκε σε αυτό του single cycle επεξεργαστή. Σε αυτό προστέθηκαν το PCWriteCondBeq και το PCWriteCondBne τα οποία ενεργοποιούνται μόνο όταν το opcode υποδεικνύει branch equal ή branch not equal. Επιπλέον, προστέθηκαν τα write enable των pipeline καταχωρητών και το σήμα immcalc.Το τελευταίο ελέγχει ένα πολυπλέκτη στο Exstage, ο οποίος βρίσκεται ενδιάμεσα στην ALU και τον πολυπλέκτη που ελέγχει την είσοδο B της ALU. Αυτός ο πολυπλέκτης επιλέγει ανάμεσα στον immediate και την έξοδο του προηγουμένου πολυπλέκτη.

Control Pipeline Registers

Προστέθηκαν τα modules IDEX-C, EXMEM-C, MEMWB-C τα οποία περιέχουν τους pipeline καταχωρητές του control, οι οποίοι μεταφέρουν-καθυστερούν τα σήματα ελέγχου του control, έως ότου καταναλωθούν από το DATAPATH.

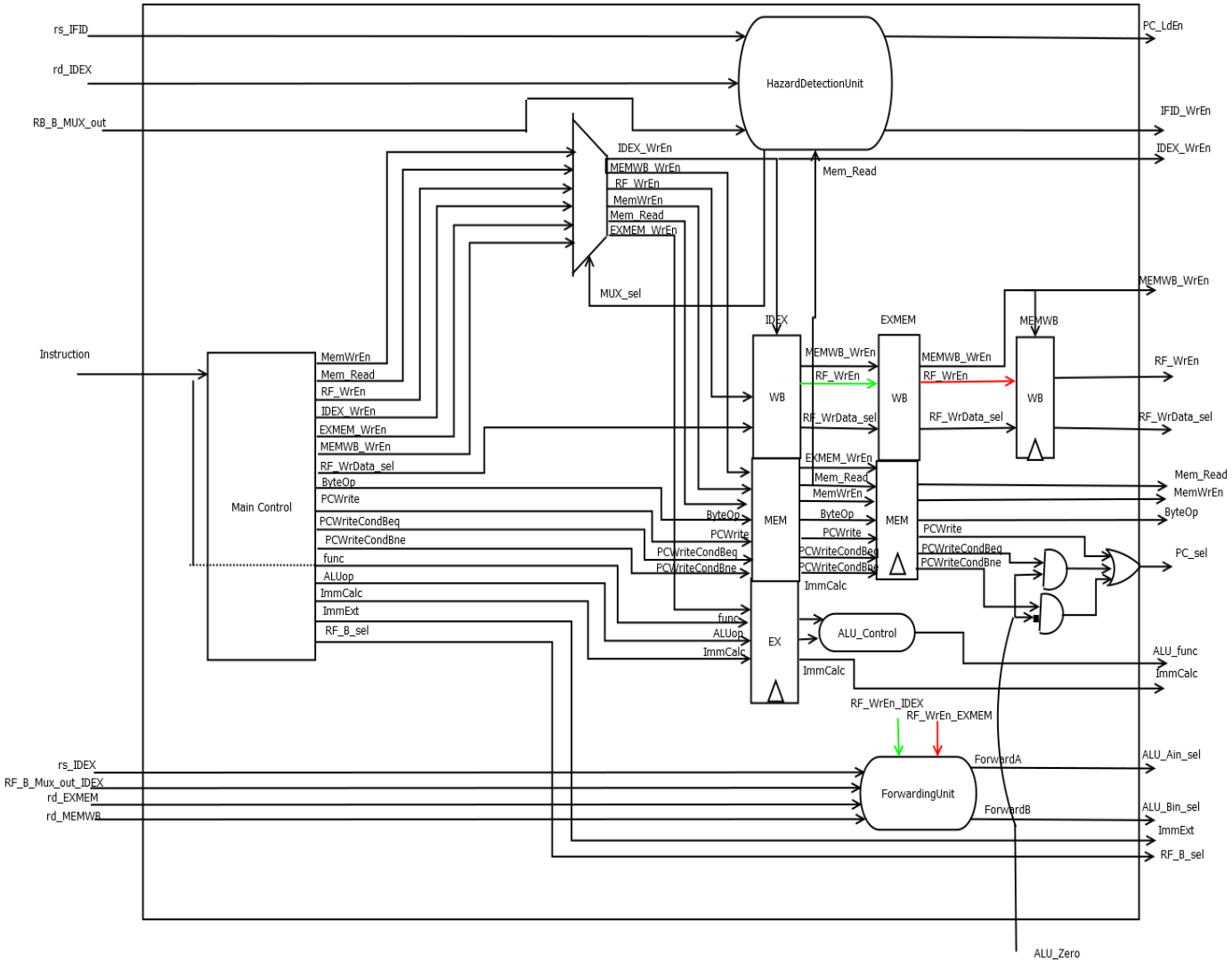
Forwarding Unit

Το Forwarding Unit δημιουργήθηκε και επαληθεύτηκε η λειτουργία του μέσω των προγραμμάτων αναφοράς. Για την δημιουργία του module προσαρμόστηκε ο κώδικας που βρίσκεται στις διαλέξεις του μαθήματος.

Hazard Detection Unit

Το Hazard Detection Unit είναι υπεύθυνο για τα stalls, ώστε να αντιμετωπιστούν data hazards του pipeline επεξεργαστή. Αυτό έχει εξόδους που ελέγχουν τα enable του IFID Register και του PC Register, καθώς και τον πολυπλέκτη που επιτρέπει να εισάγονται τα nop στα στάδια του επεξεργαστή. Ελέγχοντας τον επεξεργαστή με το αρχείο rom_hazards.data, φαίνεται ότι δουλεύει σωστά το stalling, καθώς παρατηρούνται οι αναμενόμενες τιμές στους καταχωρητές του αρχείου καταχωρητών.

Η αλληλουχία των εντολών που χρησιμοποιήθηκαν για τον έλεγχο του Hazard Detection Unit βρίσκεται στο rom_hazard.data.Σε αυτό το σημείο πρέπει να σημειωθεί ότι ο έλεγχος των δύο παραπάνω μονάδων έγινε με τον συνολικό έλεγχο του επεξεργαστή. Παρακάτω φαίνεται το block diagram του control.



DATAPATH

IFSTAGE

Σε αυτό το στάδιο επαναφέρθηκαν ο πολυπλέκτης 2 προς 1 για την επιλογή του επόμενου program counter και ο αθροιστής του PC με το τέσσερα. Το στάδιο αυτό δέχεται ως είσοδο την δεύτερη είσοδο του πολυπλέκτη, δηλαδή το PC+Immediate, το οποίο υπολογίζεται στο EXSTAGE.

DECSTAGE

Σε αυτό το στάδιο μεταφέρθηκε ο extender του Immediate ώστε να μην μεταφέρεται το σήμα ImmExt στους pipeline καταχωρητές του control. Δεν υπήρξε κάποια άλλη διαφοροποίηση σε σχέση με το DECSTAGE των προηγούμενων φάσεων.

EXSTAGE:

Σε αυτό το στάδιο υπάρχουν δυο πολυπλέκτες οι οποίοι ελέγχουν την έξοδο που οδηγείται στην ALU, οι οποίοι ελέγχονται από το forwarding unit του control. Είσοδοι τους είναι:

- η έξοδος από το αρχείο καταχωρητών (RF_A ,RF_B)
- η έξοδος του Writeback stage, δηλαδή ο υπολογισμός της ALU δύο εντολές πριν
- η έξοδος του exmem register, δηλαδή ο προηγούμενος υπολογισμός της ALU.

Η έξοδος του πολυπλέκτη που αφορά την είσοδο B της ALU, εισέρχεται σε έναν δεύτερο πολυπλέκτη. Αυτός οδηγείται από το σήμα ελέγχου immcalc και έχει δεύτερη είσοδο τον Immediate που έρχεται από τον idex καταχωρητή. Έτσι, αν η εντολή έχει δεύτερο τελεσταίο τον immediate επιλέγεται αυτός. Στο Exstage, επιπλέον, βρίσκεται το module που προσθέτει τον ολισθιμένο κατά 2 immediate με τον program counter .

MEMSTAGE

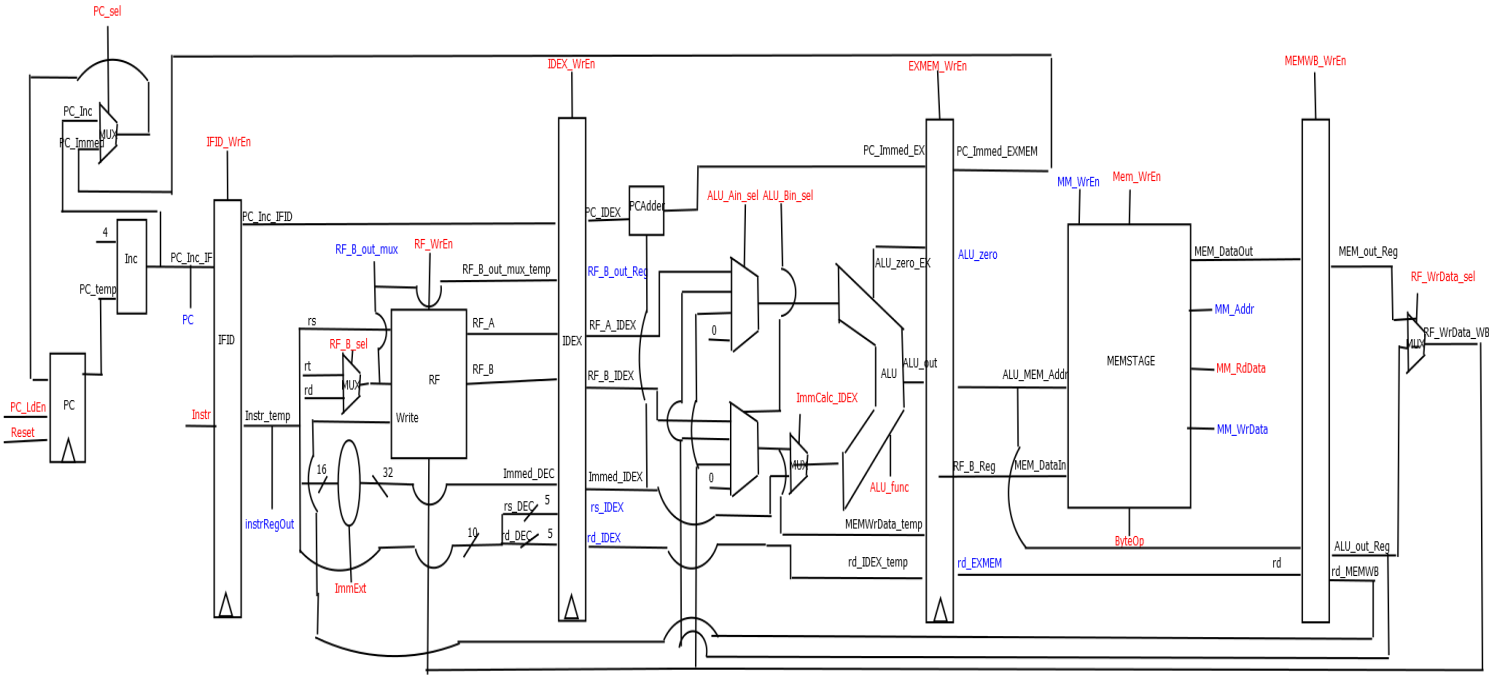
Στο datapath επαναχρησιμοποιήθηκε το MEMSTAGE της 4^{η} φάσης.

WBSTAGE

Αυτό το module περιέχει μόνο έναν πολυπλέκτη που επιλέγει τι θα εγγραφεί στο αρχείο καταχωρητών: το αποτέλεσμα της ALU ή η έξοδος της μνήμης.

DATAPATH PIPELINE REGISTERS

Στο datapath περιέχονται όλοι οι ενδιάμεσοι καταχωρητές που χρειάζονται για την υλοποίηση της pipeline λειτουργικότητας. Από αυτούς εξάγονται τα κατάλληλα σήματα προς το Control για την λειτουργία του.Παρακάτω φαίνεται το block diagram του datapath.



PIPELINE PROCESSOR

Για την δημιουργία του PROCESSOR PIPELINE συνενώθηκαν τα components datapath, control και ram. Έγινε εκτενές testing με τα .data αρχεία: rep1, rom_hazard, rom_no_hazard. Διαπιστώθηκε ότι οι εντολές του Charis εκτελούνται ορθά, αντιμετωπίζοντας τα data hazards που προκύπτουν. Να σημειωθεί ότι το rom_hazard.data είναι το αρχείο που περιέχει add, li, sw και lw, όπως ζητήθηκε στην εκφώνηση. Το documentation του ονομάζεται rom_hazard_assembly.txt