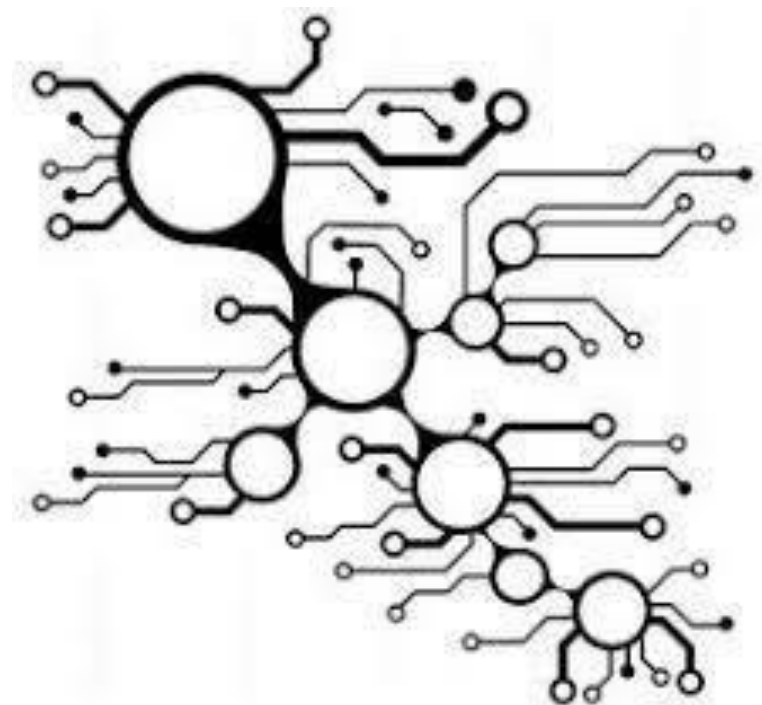


# ΟΡΓΑΝΩΣΗ ΥΠΟΛΟΓΙΣΤΩΝ

Αναφορά Εργασίας 2



ΣΟΦΟΚΛΗΣ ΦΙΛΑΡΕΤΟΣ ΓΑΒΡΙΗΛΙΔΗΣ

A.M. 2014030062

# ΕΙΣΑΓΩΓΗ

## Τέταρτη φάση

Στην τέταρτη φάση ζητήθηκε η μετατροπή του επεξεργαστή ενός κύκλου που σχεδιάσατε στην Εργασία#1 σε επεξεργαστή πολλαπλών κύκλων.

## Πέμπτη φάση

Στην Πέμπτη φάση ζητήθηκε να υλοποιήσουμε έναν **pipeline** επεξεργαστή μετατρέποντας τον κώδικα των προηγούμενων φάσεων

# ΥΛΟΠΟΙΗΣΗ

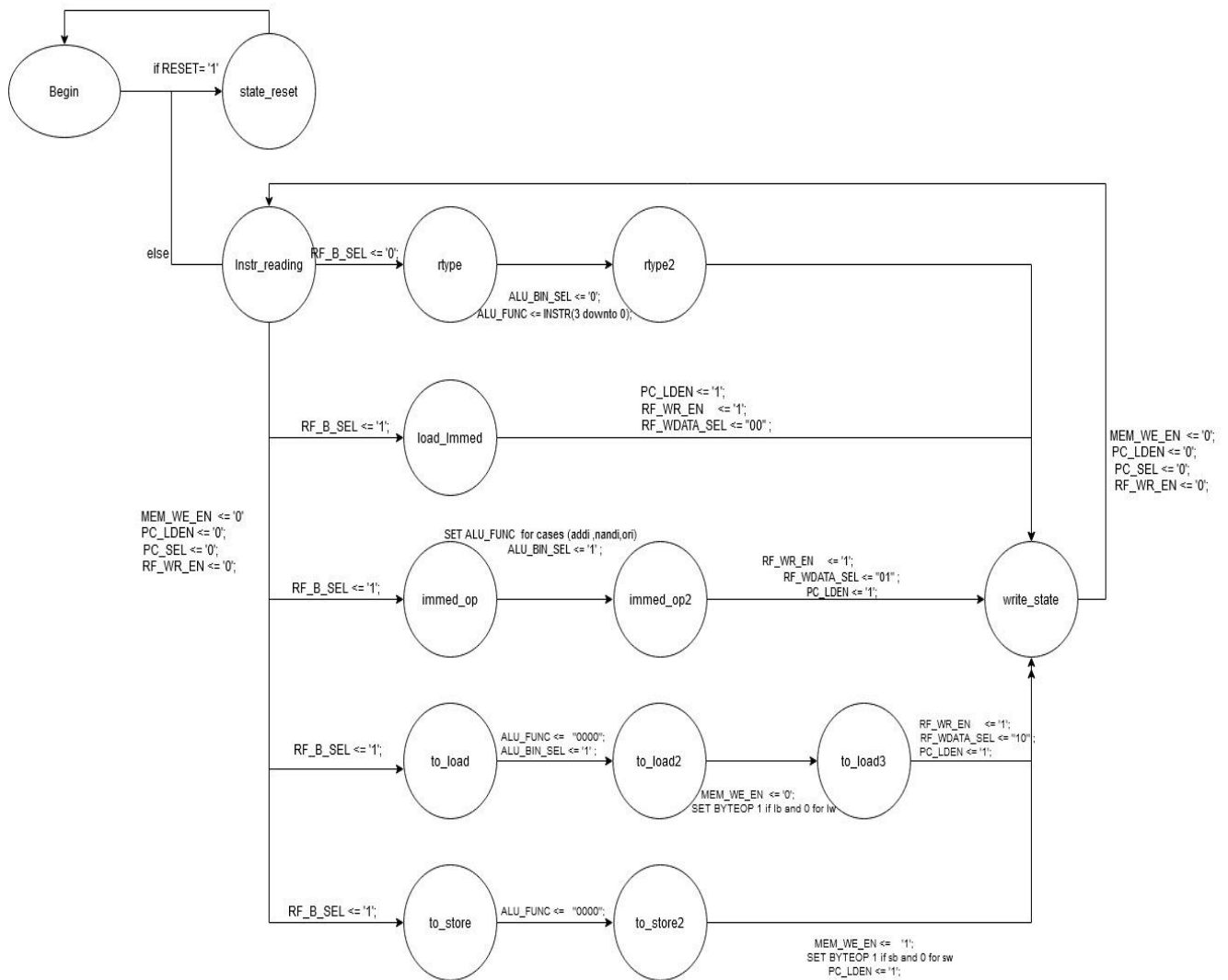
## Τέταρτη φάση

Για την υλοποίηση του **multicycle processor** χρειάστηκαν οι εξής αλλαγές από τον single cycle processor :

- 1) Αλλαγή ολόκληρου του Control module στο οποίο δημιούργησα μια Mealy FSM όπως φαίνεται στο παρακάτω διάγραμμα :

Η επιλογή των καταστάσεων έγινε σύμφωνα με την κατηγορία πράξεων που παίρνει το Instruction code.

Σε κάθε κατάσταση γίνονται οι απαραίτητοι ορισμοί στις εξόδους της μονάδας ελέγχου ,όπως φαίνεται και στο διάγραμμα , ώστε να λειτουργεί σωστά ο επεξεργαστής πολλών κύκλων



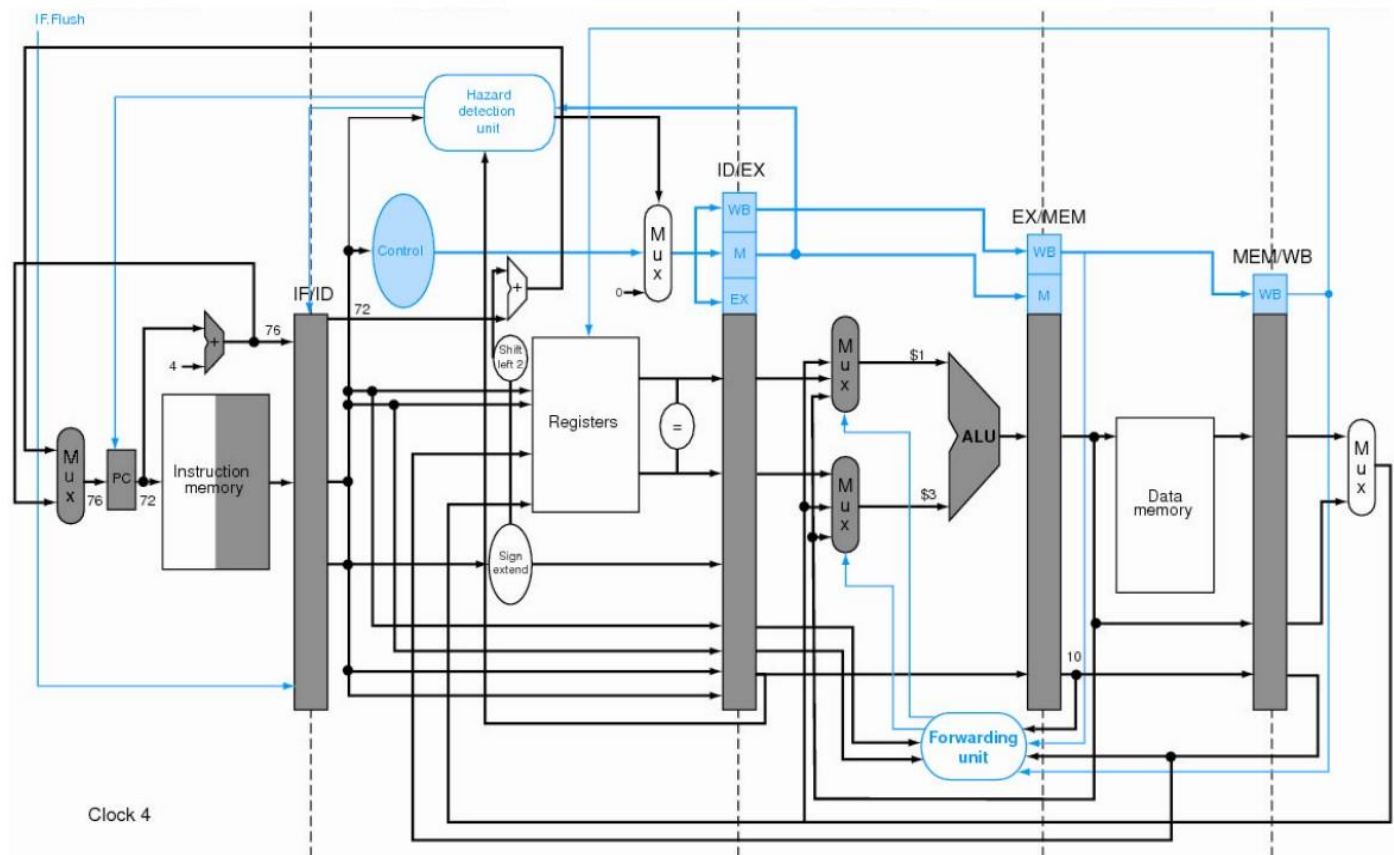
- 2) Μικρή αλλαγή στο **Datapath** . Συγκεκριμένα προστέθηκαν 2 registers που στην ουσία συγχρονίζουν την ασύγχρονη **ALU**. Η λειτουργία τους είναι απλή απλώς παίρνουν τα **RF\_A** και **RF\_B** από το **DecStage** και τα ενώνουν μούχρονα πλέον, με το **ExStage**

## Πέμπτη φάση

Αρχικά , για την υλοποίηση του **pipeline processor** χρειάστηκε να προσθέσω δυο compare modules και δυο mux2\_1 στο **DecStage** .Αυτό είναι απαραίτητο στην περίπτωση που ο καταχωρητής εγγραφής Awr είναι ίδιο με κάποιον από

τους καταχωρητές ανάγνωσης Ard1 ,Ard2 .Σε αυτή την περίπτωση πρέπει να προωθήσουμε το Datain στο Dataout του DecStage ώστε να διαβαστεί σωστά ο καταχωρητής ,αφού η τιμή εγγραφής του θα περαστεί κανονικά μετά από ένα κύκλο.

Στην συνέχεια όλη η υλοποίηση του επεξεργαστή pipeline γίνεται στη αλλαγή του Datapath σύμφωνα με το παρακάτω διάγραμμα των διαλέξεων :



Αρχικά δημιουργήθηκαν οι registers-buffers που προωθούν τα απαραίτητα δεδομένα από το κάθε stage στο επόμενο , δηλαδή IF\_DEC\_REGISTER , DEC\_EX\_REGISTER ,EX\_MEM\_REGISTER, MEM\_WB\_REGISTER.

Για την αντιμετώπιση των Data Hazards αρχικά υλοποιήθηκε το FORWARD\_UNIT όπου στην ουσία ελέγχει ασύγχρονα αν τα σήματα των καταχωρητών ανάγνωσης και εγγραφής πριν την

ALU ισούται με τα αντίστοιχα, ένα και δύο κύκλους μετά, που δεν έχουν προλάβει ακόμα να γραφτούν σωστά . Έτσι ανάλογα την περίπτωση το FORWARD\_UNIT στέλνει το σωστό σήμα select στους 2 πολυπλέκτες 4\*1 που προσθέσαμε πριν την ALU για να πάρει τις σωστές τιμές , δηλαδή αν θα πάρει τις τιμές του καταχωρητές RF\_A/RF\_B από το DEC\_EX\_REGISTER ή από ένα κύκλο μετά από το EX\_MEM\_REGISTER ή από δυο κύκλους μετά από το MEM\_WB\_REGISTER.

Τέλος υλοποιήθηκε το STALL\_UNIT που στην ουσία καθυστερεί ένα κύκλο στην περίπτωση που υπάρχει rtype εντολή μετά από lw/lb που έχει για καταχωρητή ανάγνωσης τον καταχωρητή εγγραφής της lw/lb .Αυτό γίνεται μηδενίζοντας τα RF\_WR\_EN και MEM\_WR\_EN στο DEC\_EX\_REGISTER και το PC\_LD\_EN στο IF\_STAGE.

## ΠΡΟΒΛΗΜΑΤΑ

Το πρόβλημα που δεν μπόρεσα να λύσω είναι το ότι όποτε πατούσα να σωθεί η κυματομορφή (wcfg) έκλεινε απευθείας το isim. Για αυτό το λόγο στον φάκελο με τις κυματομορφές έβαλα screenshot των simulations που έτρεξα !