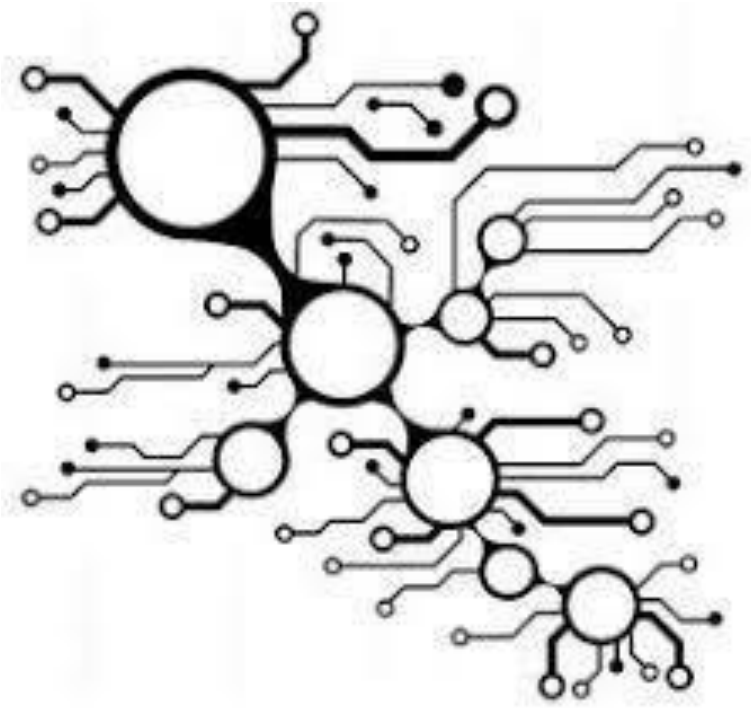
ΟΡΓΑΝΩΣΗ ΥΠΟΛΟΓΙΣΤΩΝ

Αναφορά Εργασίας 1



ΣΟΦΟΚΛΗΣ ΦΙΛΑΡΕΤΟΣ ΓΑΒΡΙΗΛΙΔΗΣ

Α.Μ. 2014030062

ΕΙΣΑΓΩΓΗ

Πρώτη φάση

Στην πρώτη φάση ζητήθηκε η δημιουργία μίας μονάδας, η οποία εκτελεί αριθμητικές καιλογικές πράξεις, της ALU καθώς και η δημιουργία ενός αρχείου καταχωρητών (Register File – RF).

Δεύτερη φάση

Σε αυτή την φάση της εργασίας στόχος είναι ο ορισμός της αρχιτεκτονικής συνόλου εντολών (Instruction Set Architecture ή ISA) και η σχεδίαση των βαθμίδων ανάκλησης εντολών (Instruction Fetch ή IF) , αποκωδικοποίησης εντολών(Instruction Decoding ή DEC), εκτέλεσης εντολών (Execution ή EX), πρόσβασης μνήμης (Memory ή MEM).

Τρίτη φάση

Στόχος της τρίτης φάσης είναι η ολοκλήρωση και ο έλεγχος του datapath ,η δημιουργία ενός controller για τα σήματα ελέγχου καθώς και η επικοινωνία μεταξύ αυτών με σκοπό την σχεδίαση ενός επεξεργαστή ενός κύκλου.

ΥΠΟΠΟΙΗΣΗ

Πρώτη φάση

ALU

Για την υλοποίηση ALU δημιούργησα ένα component που έχει ως εισόδους δύο 32 bits αριθμούς (σε συμπλήρωμα ως προς 2) και μία 4 bits για τον κωδικό της πράξης .Στις εξόδους εμφανίζονται το Out (το αποτέλεσμα της πράξης), το Zero (το οποίο λειτουργεί ως flag όταν το Out είναι ένα σήμα με 32 μηδενικά) ,το Cout (το οποίο αφορά το κρατούμενο της πρόσθεσης ή αφαίρεσης) και τελευταία το Ovf για την υπερχείλιση των πράξεων. Όσον αφορά την διεκπεραίωση των πράξεων έφτιαξα δύο ξεχωριστά module με Adder και Subber.

REGISTER FILE

Για την υλοποίηση του RF κατασκεύασα ένα component που έχεις ως

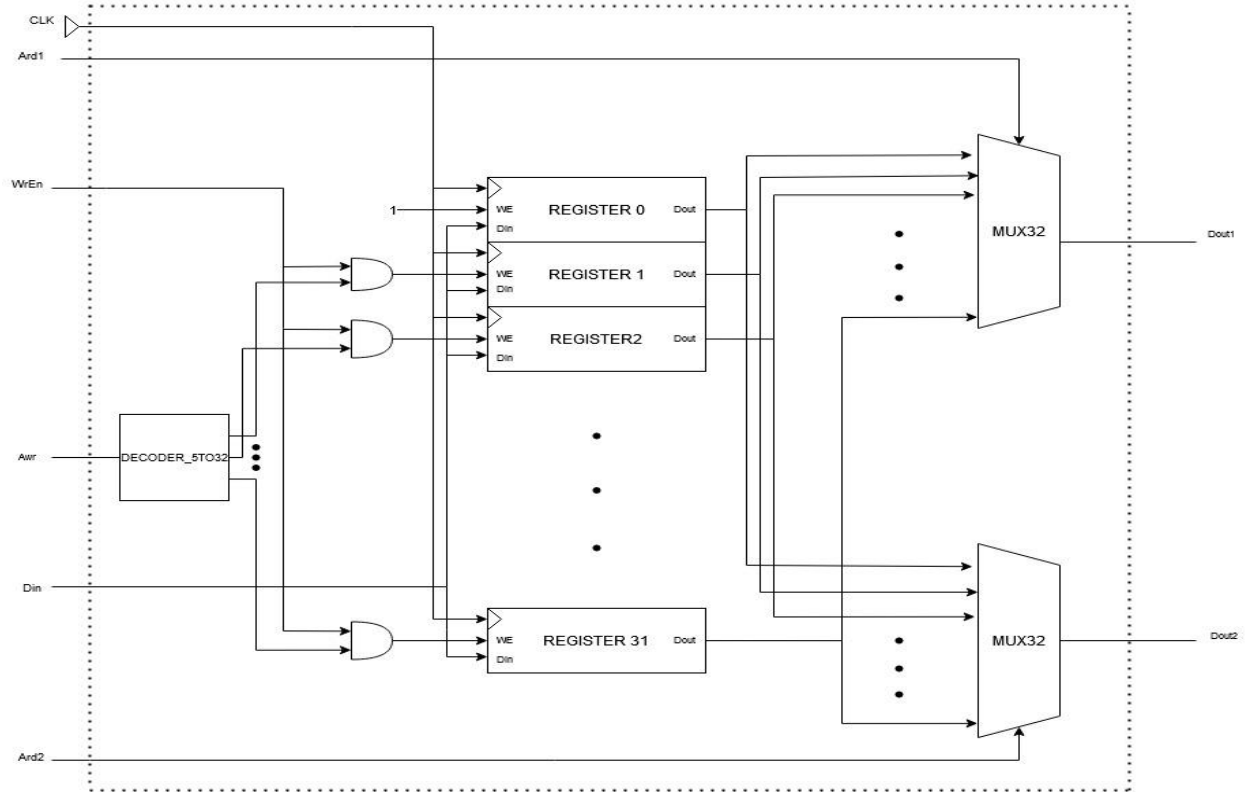
εισόδους 5bits Ard1 και Addr2 (για την επιλογή της εξόδου) ,το WrEn για τον έλεγχο της εγγραφής,Awr , Din(η τιμή που θα καταχωρηθεί στον καταχωρητή) και τέλος οι έξοδοι του συστήματος είναι δύο Dout1 και Dout2 32bit έκαστος.

Αρχικά δημιούργησα έναν register(στον οποίο το αποτέλεσμα καθυστερεί 10 ns) με την οποία κατασκεύασα πρώτα τον REG0 και έπειτα τους υπόλοιπους 31 με την χρήση της for-loop . Έπειτα σχεδιάσα έναν πολυπλέκτη που παίρνει 32 αριθμούς με 32 bit ώστε να διαλέξω το σωστό αποτέλεσμα σαν έξοδο.
Σε αυτή την υλοποίηση δημιούργησα ένα τύπο μεταβλητής (σε πακέτο σε μία βιβλιοθήκη) όπου είναι ένας πίνακας 32 θέσεων από vectors 32 θέσεων .

```
package myDataTypes is
type array32       is array (31 downto 0)      of STD_LOGIC_VECTOR (31 downto 0)
end package ;
```

!!!Έχω βάλει τα αρχεία της βιβλιοθήκης - πακέτου στους φακέλους που χρειάζεται
!!!Σε περίπτωση π βγάλει error απλώς χρειάζεται να κάνει compile πρώτα το myDataTypes.vhd πριν τρέξετε τα .vhd που καλούν την βιβλιοθήκη-πακέτο!!!!

Τελικά όλα συνδέθηκαν στο top level RF όπως φαίνεται στο παρακάτω block diagram.



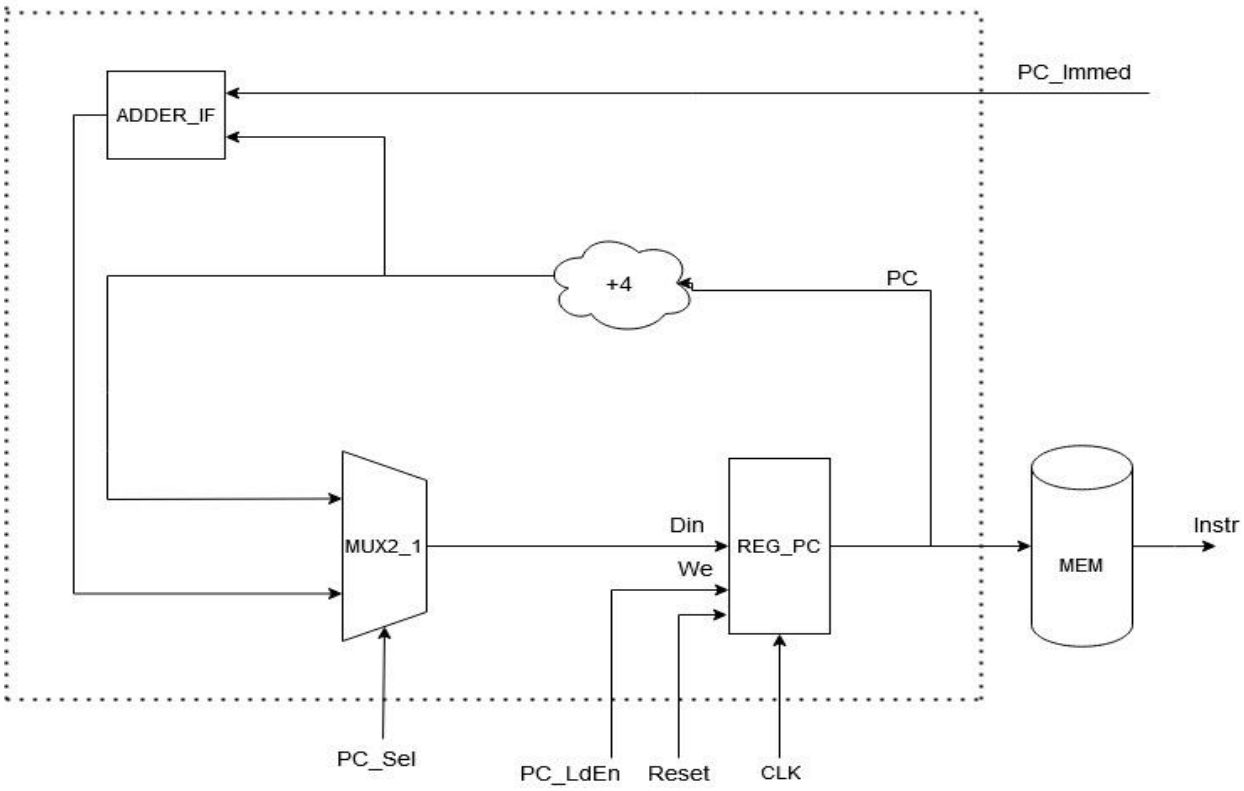
Δεύτερη φάση

Το πρώτο βήμα ήταν η μελέτη της κωδικοποίησης των εντολών CHARIS .Με αυτό τον τρόπο έπρεπε να αναγνωρίσω και να αποκωδικοποιήσω κάθε εντολή. Οι μαθηματικές και οι λογικές πράξεις καθώς και κάποιες εντολές ολίσθησης, είναι R τύπου και διακρίνονται από το function που έχουν. Τις υπόλοιπες εντολές που είναι τύπου I και J τις διακρίνω με το αν χρειάζεται να κάνουμε SignExtend ,το Immed ή ZeroFill το Immed στους αντίστοιχους καταχωρητές

Στο δεύτερο μέρος της φάσης ασχολήθηκα με την υλοποίηση της κύριας μνήμης. Για τον συγκεκριμένο λόγο χρησιμοποιήσα το κώδικα από την εκφώνηση και υλοποιησα τη μνήμη που ζητείται. Σε αυτή την μνήμη στις θέσεις από 0-1023 είναι για εντολές και υπόλοιπες για δεδομένα.

Όσο αναφορά την υλοποίηση βαθμίδας ανάκλασης εντολών (IF) ζητήθηκε να υλοποιήσω ένα module με το οποίο θα ελέγχω ποιες εντολές θα ανακτώ από την μνήμη μου.Το συγκεκριμένο σύστημα αποτελείται από έναν των 32 bits (PC - Programm counter).Μετά από κάθε εκτέλεση εντολής ο PC αυξάνεται κατά 4 και έπειτα από ένα κύκλο θα δείχνει στην επόμενη θέση της μνήμης. Σε αυτό το στάδιο δημιούργησα και έναvadder στον οποίο προστίθεται η διεύθυνση του PC, αυξημένη κατά 4, μαζί με μια είσοδο, την PC_Immed η οποία είναι 32 bits. Με αυτό τον τρόπο όταν έρχεται μια εντολή branch τότε προστίθεται στην διεύθυνση του PC η τιμή Immediate της εντολής branch και να κατευθύνεται στην επιθυμητή διεύθυνση της μνήμης. Και στο τέλος για τον έλεγχο της πληροφορίας που θα περάσει στον PC.

Παρακάτω φαίνεται το block diagram του IFSTAGE



Για την αποκωδικοποίηση των εντολών (Instruction Decoding ή DEC), ζητήθηκε να υλοποιήσω ένα module με το οποίο θα αποκωδικοποιώ τις εντολές που παίρνω από την μνήμη και ελέγχω ποια δεδομένα εισέρχονται στους καταχωρητές της REGISTER FILE. Σαν πρώτο βήμα είναι ο κατάλληλος διαχωρισμός της εντολής που ανακτώ από την μνήμη σε 4 διαφορετικά σήματα.

Σημά1(25-21 bits):ορίζει ποιον καταχωρητή να διαβάζω από την 1η θύρα ανάγνωσηςτης REGISTER FILE.

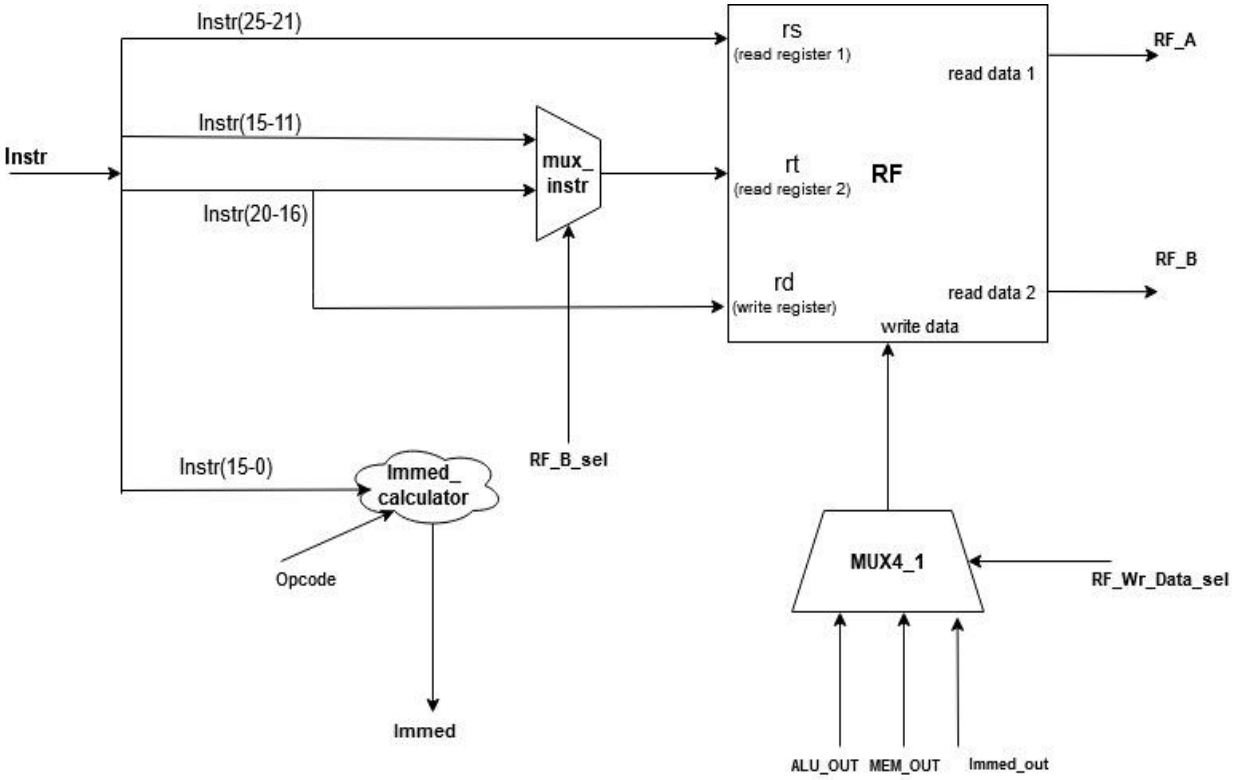
Σήμα2(15-11 bits):ορίζει ποιον καταχωρητή να διαβάζω από την 2η θύρα ανάγνωσηςτης REGISTER FILE.

Σήμα3(20-16 bits):ορίζει ποιον καταχωρητή της REGISTER FILE να γράψω ή να διαβάσω.

Σήμα 4(15-0 bits): Με αυτό και με Immed_calculator που έφτιαξα παράγω το Immed που είναι 32 bits.Στο Immed_Calculator συμπεριέλαβα όλες τις περιπτώσεις στο πινακάκι που δόθηκε στην εκφώνηση και σύμφωνα με το opcode κάθε εντολής ρυθμίζω αν χρειάζεται sign extend (Immed) ή zerofill

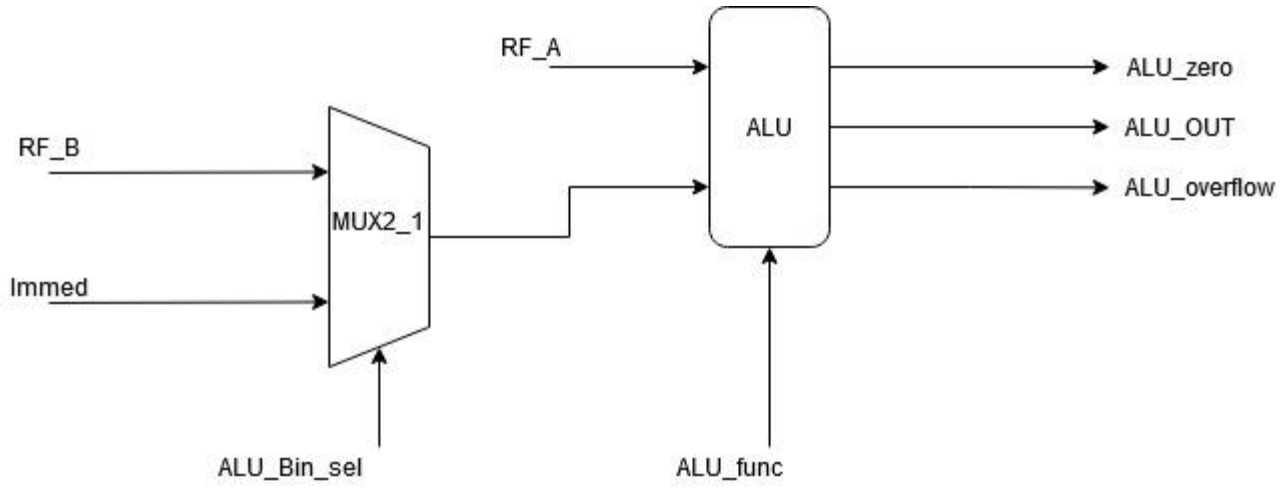
(Immed).

Το block diagram του DECSTAGE είναι :



Για την σχεδίαση της βαθμίδας EX είναι μία προέκταση της ALU με ένα πολυπλέκτη. Σε αυτόν τον πολυπλέκτη επιλέγω ανάλογα με την εντολή τον 2ο αριθμό μετον οποίο θέλω να κάνουμε πράξη. Αν είναι τύπου R η εντολή θα περνά στην είσοδο στην ALU το αποτέλεσμα από ένα καταχωρητή διαφορετικά περνάω το Immed.

Οι συνδέσεις υλοποιήθηκαν σύμφωνα με το block diagram του EXSTAGE:



Τρίτη φάση

Στην τρίτη φάση υλοποίησα το Datapath του επεξεργαστή συνδέοντας όλα τα Modules που υλοποίησα στην δεύτερη φάση, δηλαδή IFSTAGE ,DECSTAGE , EXSTAGE .Η MEMSTAGE άλλαξε σε κοινή μνήμη και βγήκε εκτός Datapath . Στην συνέχεια δημιούργησα στο Control που ελέγχει την κωδικοποιημένη εντολή Instr και δίνει τις κατάλληλες τιμές στα σήματα εισόδου που χρειάζονται τα υποκυκλώματα του Datapath για να υλοποιήσουν τις κατάλληλες πράξεις κάθε φορά. Τέλος υλοποιήθηκε το PROC_SC ,το τελικό και ανώτερο module του επεξεργαστή όπου ενώνονται τα datapath control και η κοινή πλέον μνήμη.

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 !