

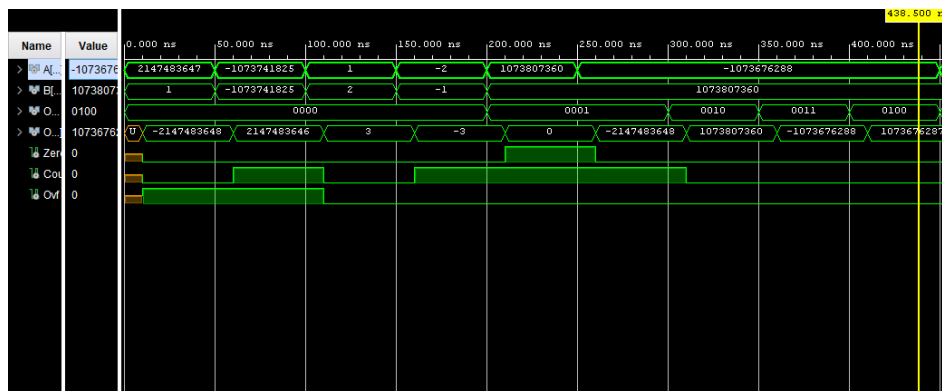
ΑΝΑΦΟΡΑ ΑΣΚΗΣΕΩΝ 1,2,3

• ΣΧΕΔΙΑΣΗ ΤΗΣ ΜΟΝΑΔΑΣ ΑΡΙΘΜΗΤΙΚΩΝ ΠΡΑΞΕΩΝ

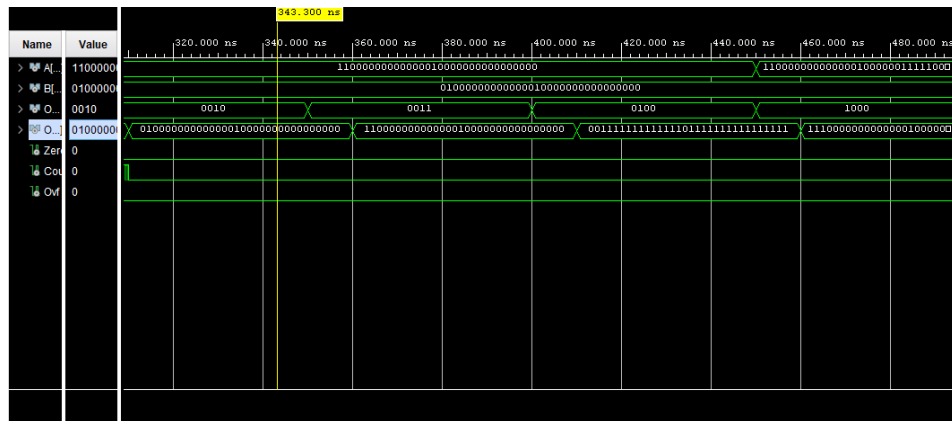
1^η φάση

Αρχικά ,υλοποιήσαμε την ALU η οποία παίρνει ως εισόδους τα σήματα A,B και ένα σήμα Op που μας δηλώνει την αριθμητική ή λογική πράξη που θέλουμε να κάνουμε .Ως έξοδο η μονάδα έχει τα σήματα OUT, το αποτέλεσμα της αριθμητικών ή λογικών πράξεων, το zero που γίνεται 1 όταν το OUT έχει την τιμή μηδέν ,το cout όταν υπάρχει κρατούμενο και τέλος το onf όταν υπάρχει υπερχείλιση στη πράξη. Κάνοντας το testbench για να ελέγξουμε το κύκλωμα πρόεκυψαν οι παρακάτω κυματομορφές:

Εδώ κυρίως παρατηρούμε την πρόσθεση και την αφαίρεση:



Ενώ εδώ τις υπόλοιπες λογικές πράξεις:



Έχει την καθυστέρηση στην έξοδο που έχει ζητηθεί 10 ns

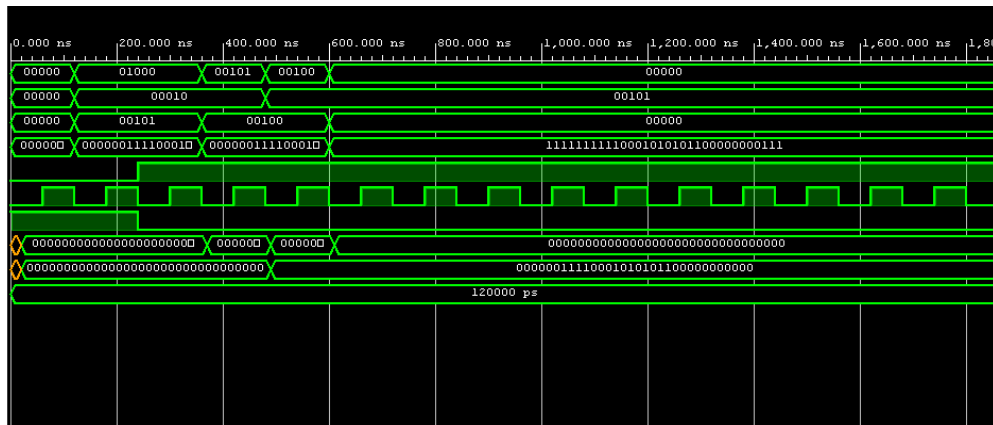
• Σχεδίαση αρχείου καταχωρητών

Αρχικά φτιάξαμε έναν καταχωρητή 32 bit ,έχει εισόδους τα δεδομένα εισόδου και ένα σήμα wren που είναι ο επιλογέας για το αν θα γράψει τα δεδομένα εισόδου,clk,rst. Και μοναδική έξοδο το Dataout που είναι η τιμή που είναι αποθηκευμένη στο register . Τον

χρησιμοποιήσαμε 32 από αυτούς τους καταχωρητές για τη παράγωγή του register file μας ακολουθώντας την συνδεσμολογία του block diagram που μας δίνετε στην εκφώνηση.

Ένας από τους 32 καταχωρήσεις περνούν είσοδο Din όταν το WrEN είναι 1 και το ποιος το καθορίζει ο decoder που έχουμε. Έχουμε επίσης τα σήματα Ard1 και Ard2 που καθορίζουν από ποιο καταχωρητή πρέπει να διαβάσει και τα αποτελέσματα βγαίνουν στο Dout1 και στο Dout2. Ο καταχωρητής μηδέν είναι αρχικοποιημένος ξεχωριστά και βγάζει πάντα 0.

Επίσης έγινε test bench για να δούμε αν λειτουργούν σωστά:



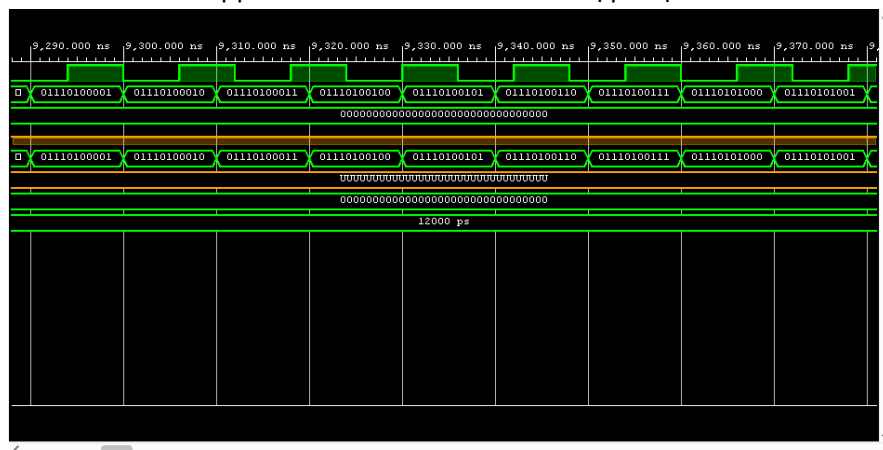
Έχει την καθυστέρηση στην έξοδο που έχει ζητηθεί.

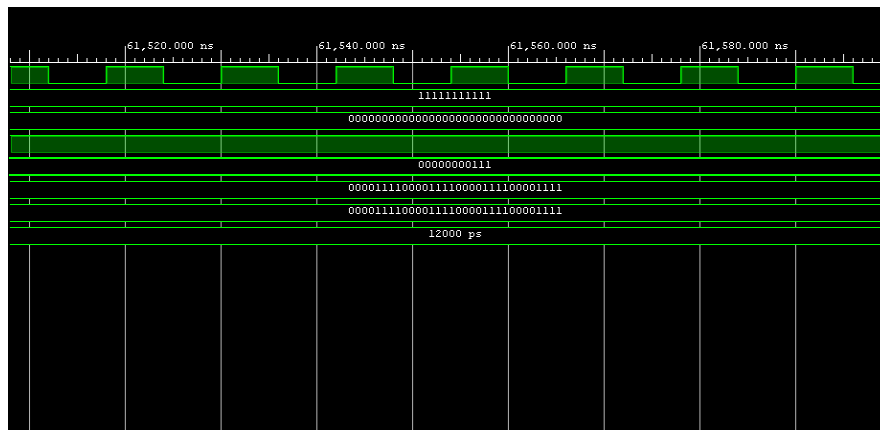
2^η φάση

• Παραγωγή Μνήμης 1024x32

Χρησιμοποιήσαμε τον κώδικα που μας δόθηκε για να κατασκευάσουμε μια μνήμη RAM 1024 θέσεων των 32 bits. Η μνήμη αυτή ανοίγει το αρχείο rom.data που έχει μέσα τις εντολές. Το οποίο εμείς τροποποιήσαμε για τις ανάγκες της άσκησης.

Ενδεικτικά ένα κομμάτι του test bench κάπου στη μέση





Και στο τέλος όταν δίνουμε

```
data_we<='1';
```

```
data_addr<="00000000111";
```

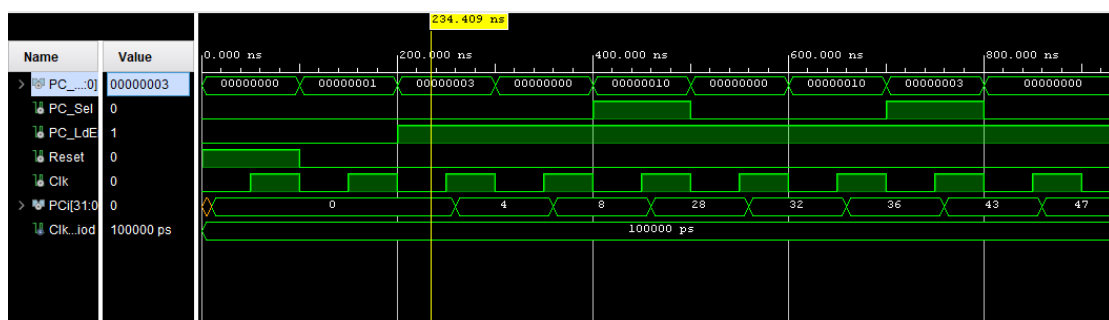
```
data_din<="00001111000011110000111100001111";
```

- Υλοποίηση της IFSTAGE μοναδας

Το IFSTAGE διαχειρίζεται τον program counter και αναλαμβάνει την ενημέρωσή του για να πάμε στην επόμενη εντολή αν δεν υπάρχει branch, και αντίστοιχα στη μεταβολή του αν υπάρχει.

Αναλυτικότερα, ifstage βαθμίδα, έχει έναν πολυπλέκτης(mux) που δέχεται δύο εισόδους των 32 bits την PC+4 και την PC+4+PC_Immed, επιλέγει ποια θα χρησιμοποιήσει ανάλογα με το PC_Sel. Η βαθμίδα PC είναι ένας καταχωρητής των 32bits με Reset, Clock και Enable. Η βαθμίδα +4 (add4_label) προσθέτει 4 στην έξοδο PC για να πάει στην επόμενη διεύθυνση, ενώ η βαθμίδα + (addImmed_label) προσθέτει στο PC+4 το PC_Immed (για εντολές branch).

Και το testbench του:



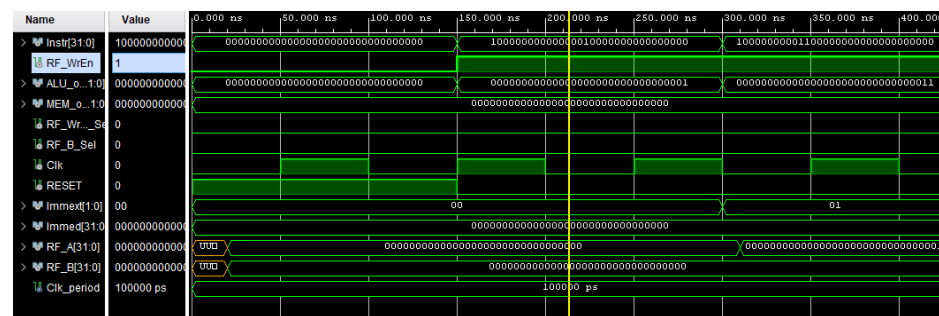
Βλέπουμε ότι περνούμε αρκετές περιπτώσεις για να καλύψουμε ένα εύρος περιπτώσεων και βγάξουν σωστά αποτελέσματα(Πχ. :όταν $Pc_sel=1$ τότε $pc=pc_{prev}+4+pc_immed=8+4+16=28$ όπως βλέπουμε στο 500.000ns)

- **Βαθμίδα αποκωδικοποίησης εντολών (DECODE)**

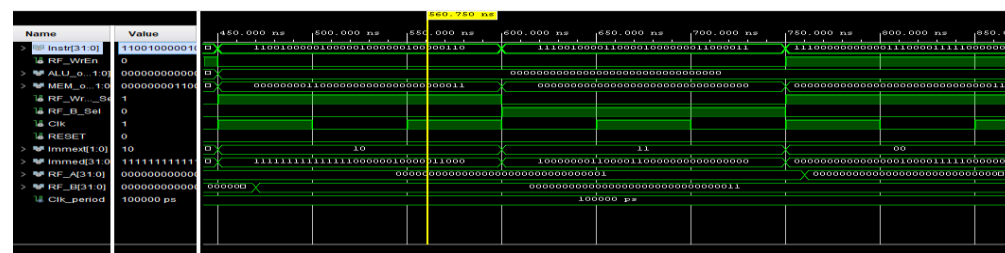
Σ' αυτό το κομμάτι χρησιμοποιήσαμε ένα αντίγραφο του αρχείου καταχωρητών (RF – Register File) από το 1ο μέρος για να σχεδιάσουμε και να υλοποιήσουμε μια βαθμίδα αποκωδικοποίησης εντολών. Το DECSTAGE λειτουργεί ως interface για τους καταχωρητές μας, αναλαμβάνει την εγγραφή καθώς και την ανάγνωση από τους καταχωρητές μας. Η εγγραφή μπορεί να είναι είτε από την ALU είτε από την μνήμη το οποίο αποφασίζεται από ένα πολυπλέκτη.

Το συννεφάκι(transform32) ,που φαίνεται στο block diagram ,παίρνει ως είσοδο τον Immed(16bits) και το OpCode(6bits) και βγάζει ως έξοδο Immed(32bits). Ανάλογα με το Immextμετατρέπει την είσοδο Immed(16bits) σε έξοδο 32bits είτε με zero fill, είτε με shift left 16 και zero fill, είτε με sign extend, είτε με sign extend και shift left 2. Ο 1ος πολυπλέκτης(mux2_1) δέχεται 2 εισόδους(5 bits η κάθε μία) και επιλέγει μία από τις 2 ως έξοδο(5 bits) ανάλογα με το RF_B_sel. Ο 2ος πολυπλέκτης(mux2_2) δέχεται 2 εισόδους(32 bits η κάθε μία) και επιλέγει μία από τις 2 ως έξοδο(32 bits) ανάλογα με το RF_WrData_sel.

Testbench:



Το αρχείο καταχωρητών δέχεται ως είσοδο την έξοδο της ALU(ALU_out), γιατί το RF_B_Sel είναι 0, και αφού το RF_WrEn είναι 1, τότε καταχωρεί την είσοδο στον Register1.

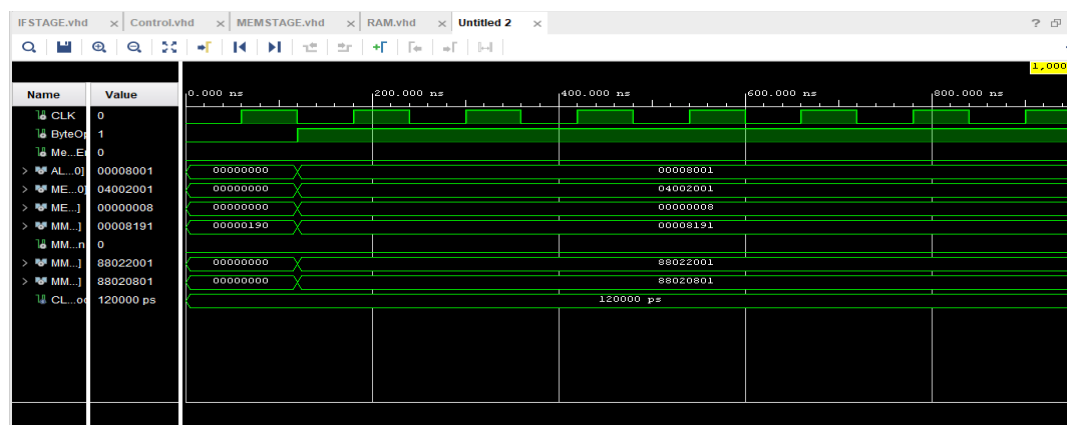


Εδώ αλλάξαμε το OpCode και παρατηρούμε ότι στα 15downto0 κάνει zero fill, όπως φαίνεται στην έξοδο Immed(32bits). Επίσης βάλαμε ο πολυπλέκτης να επιλέγει ως είσοδο την έξοδο της MEM (MEM_out) , γιατί το RF_B_Sel είναι 1, αλλά δεν την γράφει σε κάποιο καταχωρητή, αφού το RF_WrEn είναι 0 και με την ίδια λογική ελέγχουμε και τις άλλες περιπτώσεις.

- **Βαθμίδα εκτέλεσης εντολών (EX)**

Στο κομμάτι αυτό χρησιμοποιήσαμε την ALU που είχαμε σχεδιάσει από το 1ο εργαστήριο έτσι ώστε να σχεδιάσουμε και να υλοποιήσουμε μια βαθμίδα εκτέλεσης αριθμητικών και λογικών εντολών.

Testbench:



3^η φάση

Το datapath του επεξεργαστή μας καταμερίστηκε στις 'αυτοτελείς' λογικές μονάδες του οι ποιες είναι: το ifstage, το decstage, το exstage, το memstage.

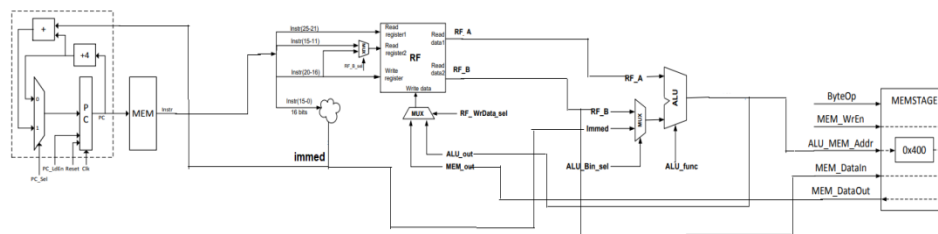
Ifstage: Υπεύθυνο για κατάλληλη αύξηση του PC.

Decstage: Υπεύθυνο για την αποκωδικοποίηση της εντολής, δηλαδή προσδιορισμός source και output registers άλλα και υπολογισμός σωστού immediate.

Exstage: Υπεύθυνο για την χρήση της ALU και, δηλαδή την επιλογή ALU source.

Memstage: Υπεύθυνο για την μνήμη. Ποιο συγκεκριμένα, τον υπολογισμό διευθύνσεων, επεξεργασία πράξεων σε επίπεδο byte, και enable.

Σχήμα:



Control είναι για να ελέγχουμε τα διάφορα σήματα που υπάρχουν στο datapath

Στο proc_cs αρχείο γίνεται η σύνδεση του control του datapath και της μνημης (ram)

Παράγει τα σήματα ελέγχου για τα υπόλοιπα modules. παίρνει σαν εισόδους το instruction, το zero και clk,rst και ανάλογα με το instruction υπολογίζει με πολυπλέκτες το opcode της ALU, τον επιλογέα δεύτερης εισόδου της ALU, τον επιλογέα για αλλαγή του immediate, τον επιλογέα για το αν θα κάνει branch ο PC ή όχι, τον επιλογέα ενεργοποίησης εγγραφής στη μνήμη, τον επιλογέα μεταξύ lw/sw lb/sb, τον επιλογέα για το δεύτερο καταχωρητή της ALU, και τον επιλογέα εγγραφής καταχωρητή.

Τελικό test bench:

