

Αναφορά Εργαστηριακής άσκησης 1

Ομάδα LAB31239665

Καλογεράκης Στέφανος AM:2015030064

Πίσκοπος Διονύσης AM:2015030115

Σκοπός εργαστηριακής άσκησης

Σκοπός της πρώτης εργαστηριακής άσκησης ήταν η υπενθύμιση των γνώσεων της προχωρημένης λογικής σχεδίασης, της γλώσσας περιγραφής υλικού VHDL όπως και η περαιτέρω εξοικείωση με το λογισμικό Xilinx.

Προεργασία

Σαν παραδοτέα/προεργασία του εργαστηρίου ζητήθηκε και στα δύο μέρη της άσκησης τόσο ο πηγαίος κώδικας σε VHDL, όσο και κυματομορφές προσομοίωσης. Επιπλέον, στο Β μέρος ζητήθηκε ένα σχηματικό διάγραμμα της συνδεσμολογίας του αρχείου καταχωρητών (Βλέπε Παράρτημα).

Περιγραφή

Η πρώτη άσκηση χωρίστηκε σε δύο μέρη υλοποίησης. Αρχικά, στο μέρος Α σχεδιάστηκε και υλοποιήθηκε μια μονάδα υπολογισμού λογικών και αριθμητικών πράξεων (**ALU**) ενώ στην συνέχεια στο μέρος Β σχεδιάστηκε ένα αρχείο καταχωρητών (**Register File**).

Μέρος Α

Στο Α μέρος υλοποίησης όπως προαναφέρθηκε, υλοποιήσαμε μια μονάδα υπολογισμού λογικών και αριθμητικών πράξεων ALU στόχος της οποίας είναι η εκτέλεση πράξεων. Τέτοιες πράξεις είναι η πρόσθεση, η αφαίρεση, λογικές πράξεις AND, OR, NOT και κάποιες ολισθήσεις αριθμητικές, λογικές και κυκλικές. Κατά την υλοποίηση μας, η κάθε πράξη είναι και ένα διαφορετικό module και όλα μαζί συνδέονται στο top level.

Στο top-level η επιλογή της κάθε πράξης γίνεται μέσω του πεδίου Op, το οποίο λειτουργεί σαν κωδικός και σχεδιάστηκε βάσει των προδιαγραφών της εκφώνησης. Αξίζει να σημειωθεί ότι τα σήματα εισόδου της ALU είναι αριθμοί σε 2's complement. Ακόμα, υπάρχουν και τρία βοηθητικά σήματα εξόδου 1 bit(Ovf, Cout, Zero) τα οποία ενεργοποιούνται(γίνονται '1') σε συγκεκριμένες περιπτώσεις.

Συγκεκριμένα:

1. **Σήμα Zero:** Γίνεται '1' όταν όλα τα bits του Out(έξοδος) είναι μηδενικά
2. **Σήμα Cout:** Γίνεται '1' όταν το αποτέλεσμα της πρόσθεσης ή της αφαίρεσης δύο αριθμών δεν χωράει στα 32bits εξόδου που έχουμε διαθέσιμα.
3. **Σήμα Onf:** Γίνεται '1' όταν το αποτέλεσμα κατά την πρόσθεση ή την αφαίρεση δύο θετικών αριθμών είναι αρνητικό ή αντίστοιχα το αποτέλεσμα δύο αρνητικών αριθμών είναι θετικό. **Παρατήρηση: Να σημειωθεί ότι δεν πρέπει να ταυτίζουμε το σήμα Cout της προηγούμενης περίπτωσης, με αυτό του Onf καθώς μελετάμε εισόδους οι οποίες είναι σε μορφή 2's complement**

Μέρος Β

Στο δεύτερο μέρος της εργαστηριακής άσκησης, καλούμαστε να υλοποιήσουμε το register file component του επεξεργαστή. Πιο συγκεκριμένα, για την επίτευξη του προαναφερθέν σκοπού, χρειάστηκε να δημιουργήσουμε τα components που αποτελούν το top level του FR, που περιγράφονται παρακάτω:

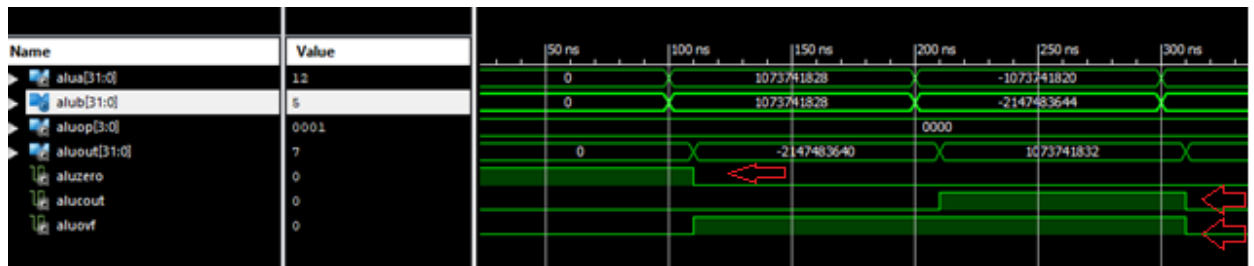
- 1 decoder 5 σε 32 με σκοπό την εύρεση του καταχωρητή που θέλουμε να γράψουμε.
- 32 καταχωρητές με εισόδους ρολογιού και σήματος Enable εκάστως.
- 2x32 πολυπλέκτες 32 σε 1 για την δημιουργία των 2 συστημάτων πολυπλεξίας, που θα δίνουν τα αποτελέσματα των registers που επιθυμούμε να διαβάσουμε.

Για την υλοποίηση του ερωτήματος **Β** με τα παραπάνω components, χρειάστηκε αρχικά να δημιουργήσουμε τα σήματα Enable των καταχωρητών, προέκυπταν από το αποτέλεσμα της διεύθυνσης που έπαιρνε ο decoder με το σήμα Wr_EN του RF μέσω μιας πύλης AND. Στην συνέχεια, αφότου έχουμε δημιουργήσει τους καταχωρητές μας, δημιουργούμε για κάθε σύστημα πολυπλεξίας (MUX1 και MUX2 από το σχήμα) 32 πολυπλέκτες 32 σε 1. Με αυτό τον τρόπο, αντιστοιχούμε το n-οστό bit από κάθε καταχωρητή (για n από 0 έως 31) στο n-οστό πολυπλέκτη του κάθε συστήματος. Έτσι, καταφέρνουμε να δώσουμε ένα 32-bit αποτέλεσμα στην έξοδο του συστήματος με βάση το κοινό select (Σήμα Ard1 για το σύστημα MUX1 και Ard2 για το MUX2 αντίστοιχα) που θα διαθέτουν οι 32 πολυπλέκτες κάθε συστήματος πολυπλεξίας. Τέλος, για να συνδέσουμε και τους 32 καταχωρητές με τα 2 συστήματα πολυπλεξίας, δημιουργούμε στο top level έναν πίνακα 32x32 με την κάθε σειρά να αντιστοιχεί στα 32 bit του κάθε καταχωρητή, ενώ η κάθε στήλη είναι το bit του κάθε καταχωρητή, από το MSB στο LSB. Η προηγούμενη δημιουργία του πίνακα σημάτων, μας βοηθάει στην κατανόηση αντιστοιχίας των σημάτων των καταχωρητών με τα συστήματα πολυπλεξίας.

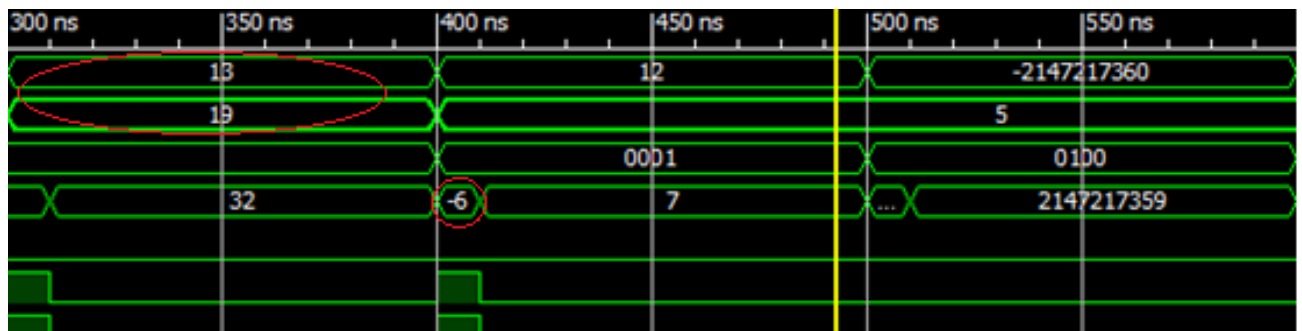
Κυματομορφές-Προσομοίωση

ALU

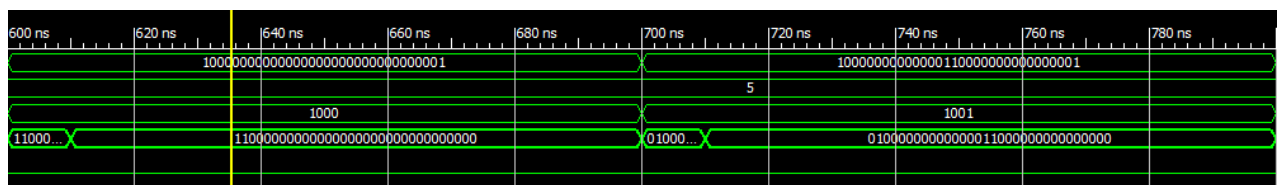
Όπως προαναφέρθηκε το πιο ενδιαφέρον κομμάτι ήταν κατά την πρόσθεση και τη αφαίρεση τα σήματα Zero, Onf, Cout. Παρακάτω φαίνονται και οι τρεις αυτές περιπτώσεις (εξηγήθηκε σε προηγούμενο μέρος η λειτουργία τους) που επαληθεύουν τις υποθέσεις μας.



Αξίζει να σημειωθεί μια ενδιαφέρουσα συμπεριφορά που εμφανίζεται κατά την διάρκεια των 10ns καθυστέρησης μέχρι το να εξάγουμε το τελικό αποτέλεσμα. Συγκεκριμένα βλέπουμε ότι σε όλες τις μεταβάσεις έχουμε την εφαρμογή της καινούργιας επιλεγμένης πράξης με την προηγούμενη είσοδο όπως χαρακτηριστικά φαίνεται και από την παρακάτω εικόνα.

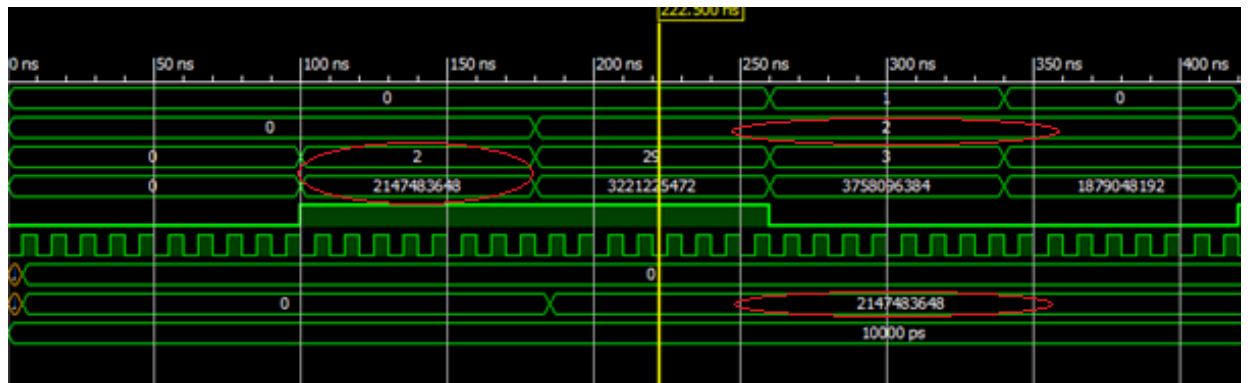


Παραθέτουμε ακόμη ενδεικτικά κυματομορφές για την αριθμητική και την λογική δεξιά ολίσθηση αντίστοιχα

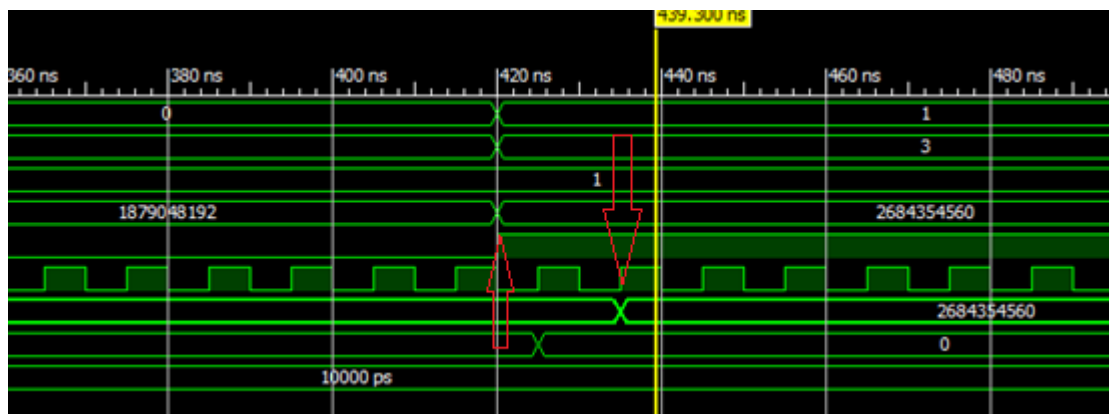


Register File

Κατά την εκτέλεση του Register File σε top level παρατηρούμε ενδεικτικά την λειτουργία όπως φαίνεται παρακάτω. Δεν υπάρχουν πολλές ακραίες περιπτώσεις να ελεγχθούν πραγματοποιούμε όμως κάποιες διαφορετικές για να σιγουρευτούμε ότι η σύνδεση έχει πραγματοποιηθεί σωστά.



Πιο συγκεκριμένα, γράφουμε στον καταχωρητή 2 και στην συνέχεια διαβάζουμε με επιτυχία όπως βλέπουμε τα δεδομένα. Αντίστοιχα έχουν γίνει κάποιες ακόμα δοκιμές.



Επίσης όπως φαίνεται και από το χαρακτηριστικό στιγμιότυπο παραπάνω λειτουργούν και οι καθυστερήσεις όπως θα θέλαμε. Η περίπτωση που παραθέτουμε σαν παράδειγμα είναι η περίπτωση εγγραφής. Αξίζει να σημειωθεί ότι οι καθυστερήσεις έχουν τεθεί από τις προδιαγραφές της εκφώνησης.

Σφάλμα κατά την αποστολή κώδικα

Κατά την αποστολή του κώδικα, σημειώθηκε λάθος ονομασία του decoder5x32.vhd σε decoder5x8.vhd , με αποτέλεσμα το top level του register_file.vhd να μην αναγνωρίζει ως component τον decoder που είχαμε δημιουργήσει. Το σφάλμα ανιχνεύτηκε αμέσως και η σωστή ονομασία του αρχείου περιλήφθηκε στην εξέταση

του εργαστηρίου, προσδίδοντας πλήρη αποτελεσματικότητα στον κωδικά του δεύτερου μέρους της άσκησης.

Παράρτημα

Επισυνάπτεται παρακάτω το σχηματικό διάγραμμα συνδεσμολογίας αρχείου καταχωρητών

