

Αναφορά Εργαστηρίου 1

Κωδικός Ομάδας **LAB31231465**

Διαλεκτάκης Γιώργος

Βαϊλλάκης Απόστολος Νικόλαος

Προεργασία

Αρχικά να αναφέρουμε ότι δεν μας ζητήθηκε να παραδώσουμε κάποιο κυκλωματικό διάγραμμα ως προεργασία στα πλαίσια του 1^{ου} εργαστηρίου, παρά μόνο τον πηγαίο κώδικα σε VHDL και τις κυματομορφές του.

Περιγραφή Άσκησης

Σκοπός της άσκησης ήταν η σχεδίαση σε Γλώσσα Περιγραφής Υλικού VHDL μιας μονάδας υπολογισμού λογικών και αριθμητικών πράξεων (ALU) και ενός αρχείου καταχωρητών (Register File) καθώς και η προσομοίωσή τους με τη βοήθεια των εργαλείων της Xilinx.

ΠΡΩΤΟ ΜΕΡΟΣ:

Στο πρώτο μέρος του εργαστηρίου όπως ήδη αναφέραμε, υλοποιήσαμε την μονάδα **ALU** η οποία εκτελεί αριθμητικές πράξεις όπως πρόσθεση, αφαίρεση μεταξύ δύο αριθμών σε 2's complement, αλλά και λογικές πράξεις όπως AND, OR, NOT όπως επίσης διάφορες ολισθήσεις αριθμητικές, λογικές και κυκλικές.

Τα σήματα του κυκλώματος είναι αριθμοί σε **2's complement**, επίσης υπάρχει ο κωδικός πράξης **Op** που ελέγχει την λειτουργία της ALU. Τέλος, υπάρχουν και τα σήματα **Zero**, **Cout** και **Ovf**.

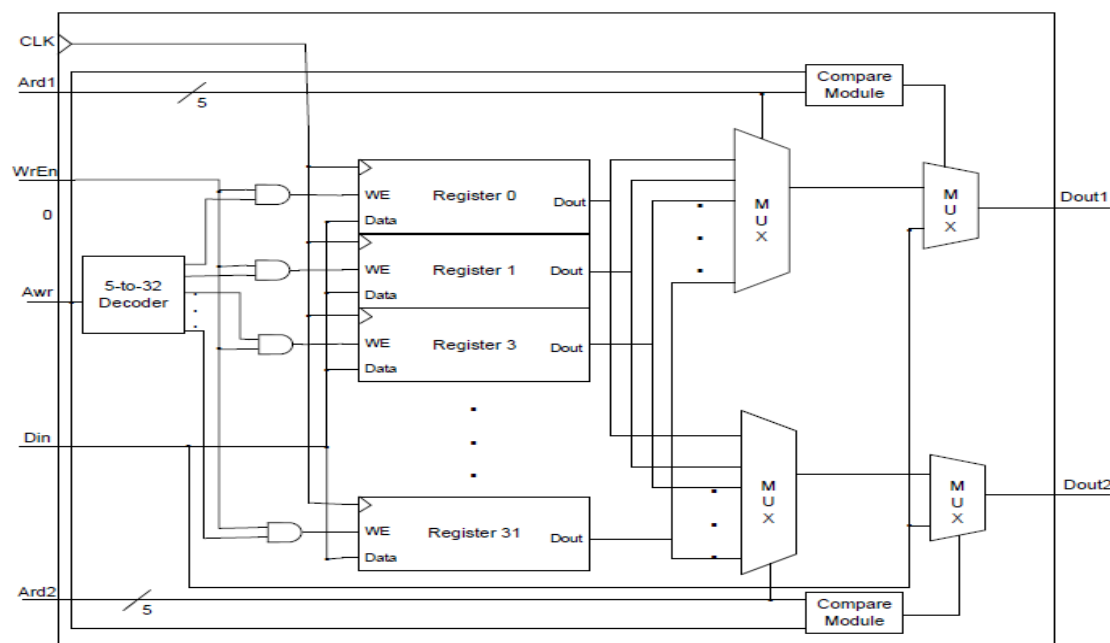
- Το σήμα Zero ενεργοποιείται όταν όλα τα bits του αποτελέσματος μίας οποιασδήποτε πράξης είναι '0'.
- Το σήμα Cout ενεργοποιείται όταν το αποτέλεσμα της πρόσθεσης ή της αφαίρεσης 2 αριθμών δεν χωράει σε πλάτος 32-bits. Πρέπει να τονίσουμε ότι μπορεί να υπάρξει κρατούμενο εξόδου(Cout) χωρίς να υπάρχει υπερχειλίση(Overflow) καθώς μιλάμε για αριθμούς σε format 2's complement.

- Υπερχείλιση μπορεί να υπάρξει μόνο όταν η ALU εκτελεί πρόσθεση ή αφαίρεση και σε καμία άλλη περίπτωση. Το σήμα *Onf* θα ενεργοποιηθεί όταν οι δύο αριθμοί που προστίθενται ή αφαιρούνται είναι και οι δύο θετικοί και βγάλουν αρνητικό αποτέλεσμα ή όταν είναι και οι δύο αρνητικοί και έχουν θετικό αποτέλεσμα.

ΔΕΥΤΕΡΟ ΜΕΡΟΣ:

Στο δεύτερο μέρος του εργαστηρίου αρχικά, υλοποιήσαμε έναν καταχωρητή των 32 bits ο οποίος όταν το σήμα ενεργοποίησης εγγραφής(**WE**) γίνει '1' φέρνει στην έξοδό του(**DataOut**) τα δεδομένα που έχει στην είσοδό(**DataIn**).

Στη συνέχεια χρησιμοποιήσαμε τον καταχωρητή αυτόν 32 φορές με σκοπό την παραγωγή ενός αρχείου καταχωρητών(**Register File**) που περιγράφεται στο



παρακάτω κυκλωματικό διάγραμμα.

Όπως φαίνεται στο παραπάνω σχήμα, υπάρχουν 2 διευθύνσεις ανάγνωσης **Ard1** και **Ard2** καθώς διαβάζουμε πάντα δύο-δύο τους καταχωρητές και μία διεύθυνση εγγραφής **Awr** για τον καταχωρητή που θέλουμε να γράψουμε. Το σήμα **WrEn** μας “λέει” πότε να γράψουμε στους καταχωρητές. Το σήμα **DataIn** είναι τα δεδομένα προς εγγραφή, ενώ το **DataOut1** και **DataOut2** τα δεδομένα που έχει ο πρώτος και ο δεύτερος καταχωρητής που διαβάζουμε αντίστοιχα.

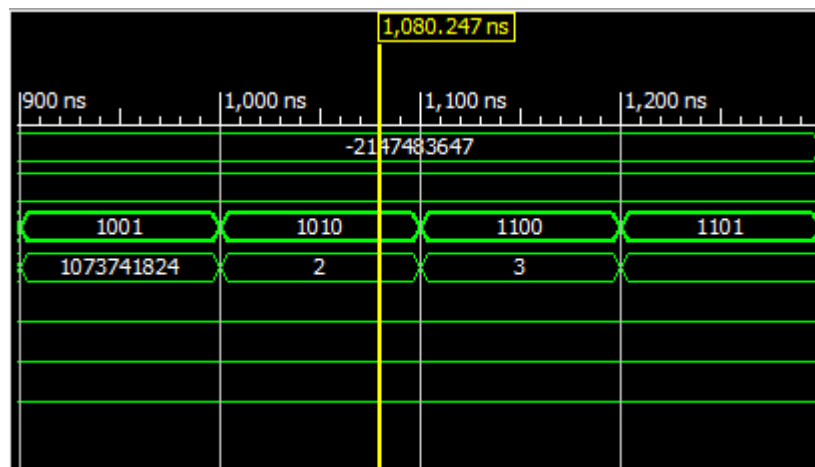
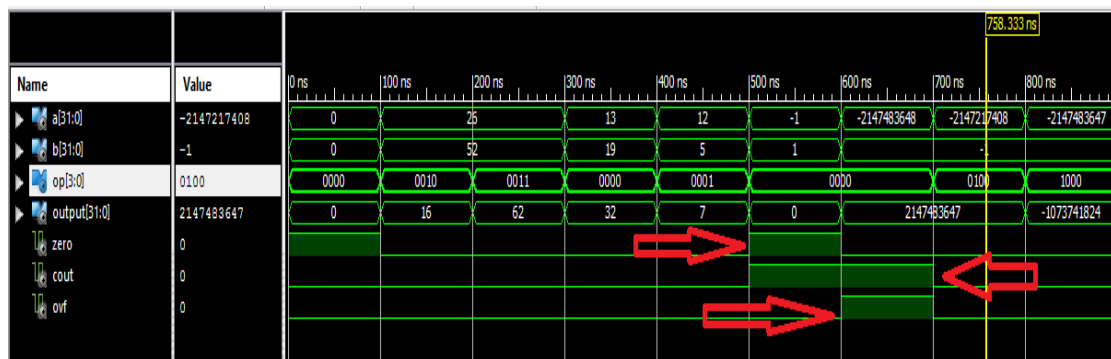
Πρέπει να τονίσουμε ότι με σκοπό να μην αλλάζει ποτέ τιμή ο καταχωρητής **R0** και να έχει πάντα την τιμή '0', τον αρχικοποιήσαμε στο '0' και επιπλέον θέσαμε σε ένα

εσωτερικό σήμα(**REGWENO**), που αποτελεί το σήμα ενεργοποίησης εγγραφής στον πρώτο καταχωρητή, την τιμή '0'.

Τέλος, όσον αφορά τα **Compare Modules**, αυτά ελέγχουν αν γράφουμε και ταυτόχρονα διαβάζουμε από τον ίδιο καταχωρητή, έτσι ώστε σε περίπτωση που συμβαίνει αυτό, τα δεδομένα που έχουμε στην είσοδο να περνάνε κατευθείαν στην έξοδο χωρίς να περιμένουν το ρολόι, γεγονός που καθιστά το κύκλωμά μας πιο γρήγορο. Να σημειωθεί ότι έχουμε συμπεριλάβει ως συνθήκη στην υλοποίηση του Compare Module, η 2η είσοδος να είναι διάφορη του "00000" που αναφέρεται στην διεύθυνση εγγραφής στον R0 ο οποίος έχει πάντα την τιμή μηδέν και δεν μπορούμε να γράψουμε σε αυτόν. Διαφορετικά, το κυκλώμα σε αυτό το σημείο θα "Latch-αρε".

Κυματομορφές

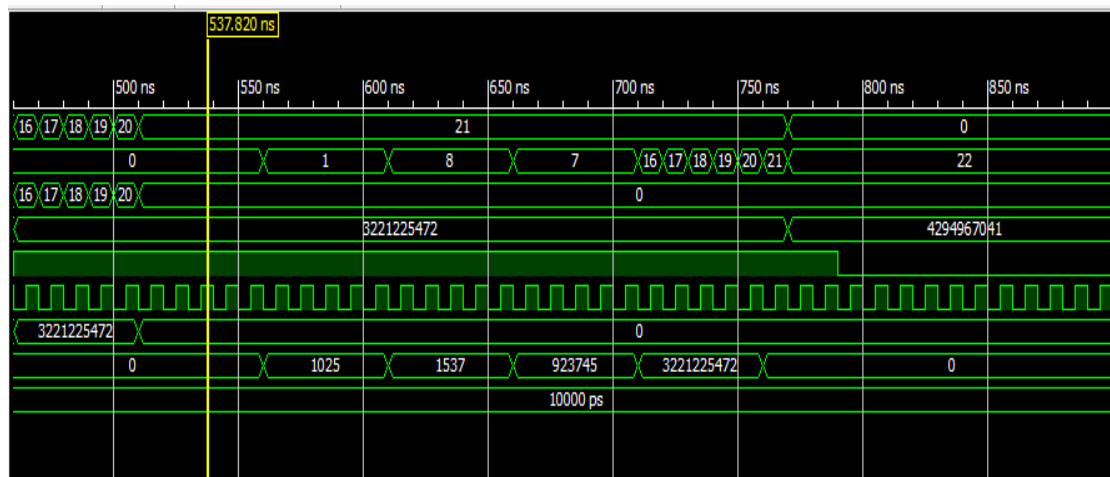
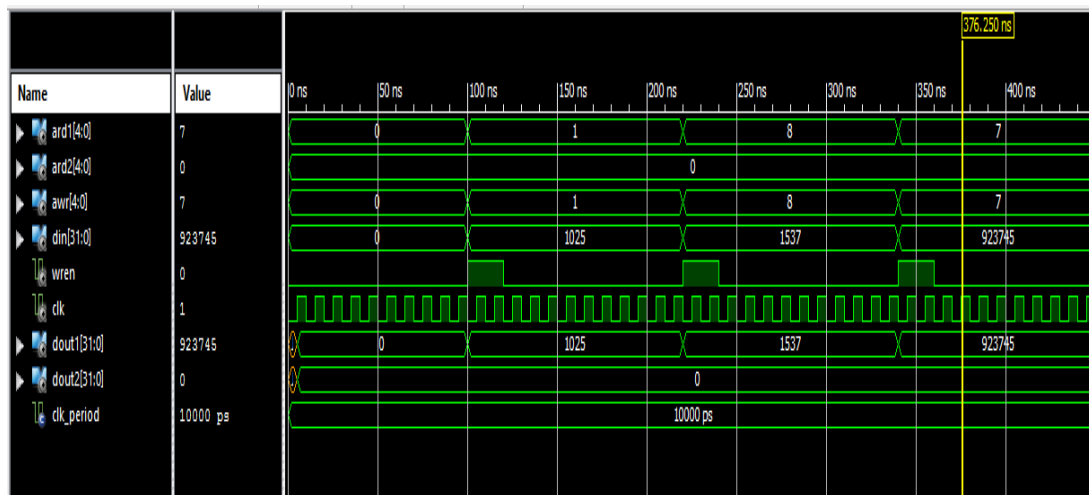
ALU:



Στις παραπάνω κυματομορφές που αφορούν την προσομοίωση της ALU παρατηρούμε με την σειρά τις εξής πράξεις: And, Or, πρόσθεση, αφαίρεση, ξανά πρόσθεση του 1 με το -1 όπου περιμένουμε και Zero και κρατούμενο εξόδου, έπειτα μία ακόμα πρόσθεση όπου περιμένουμε υπερχείλιση καθώς προσθέτουμε 2 αρνητικούς αριθμούς και το αποτέλεσμα είναι ένας θετικός, Not και τις ολισθήσεις: sra, srl, sll, rl, rr.

Τα κόκκινα βέλη δείχνουν τις αλλαγές των σημάτων Zero, Cout και Onf.

REGISTER FILE:



Στις παραπάνω κυματομορφές παρατηρούμε την λειτουργία του αρχείου καταχωρητών. Αρχικά, γράφουμε σε κάποιους καταχωρητές, στη συνέχεια τους διαβάζουμε για να δούμε αν όντως έχουν τα δεδομένα που τους βάλαμε στην αρχή. Επίσης, ελέγξαμε ότι τα write και read παίρνουν μόνο έναν κύκλο ρολογιού για να

εκτελεστούν. Τέλος, προσπαθούμε να γράψουμε τον καταχωρητή R0, ωστόσο παρατηρούμε ότι δεν αλλάζει τιμή αλλά παραμένει στο '0'.

Συμπεράσματα

Στο εργαστήριο αυτό θυμηθήκαμε πόσο χρήσιμη είναι μία γλώσσα περιγραφής υλικού όπως η VHDL και πήραμε μία γεύση του τι πρόκειται να ακολουθήσει!