# Οργάνωση Υπολογιστών ~ ΗΡΥ 312

#### Προεργασία 3ου Εργαστηρίου

Ομάδα Εργασίας: LAB31239629

Κονιδάρη Ηρώ A.M. 2012030049 Μάνεσης Αθανάσιος A.M. 2014030061

### Σκοπός της Άσκησης

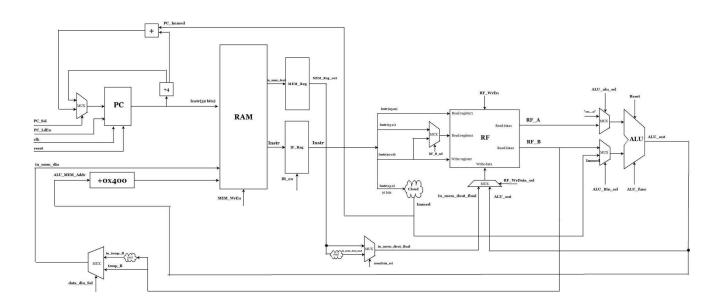
Σκοπός αυτής της άσκησης ήταν η ολοκλήρωση και ο έλεγχος του datapath του δεύτερου εργαστηρίου.

#### Περιγραφή/Υλοποίηση της Άσκησης

Χρησιμοποιήσαμε τις μονάδες που υλοποιήσαμε στο προηγούμενο εργαστήριο και ενώνοντας τα modules δημιουργήσαμε το πλήρες datapath. Οι διεπαφές του datapath είναι τα control σήματα του. Στο bonus κομμάτι της άσκησης μας ζητείται να υλοποίησουμε την αυτόματη παραγωγή των control σημάτων συναρτήσει του Instr που βρίσκεται στην μνήμη.

Στο testbench του datapath τρέξαμε και τα δύο ενδεικτικά προγράμματα που μας δόθηκαν και προσθέσαμε και τις εντολές not, ror. Όλες οι εντολές αποκωδικοποιήθηκαν και εκτελέστηκαν σωστά.

Παρακάτω ακολουθεί το block-diagram του Datapath. Εκτός από τα modules του δεύτερου εργαστηρίο προστέθηκαν, δύο καταχωρητές IF και MEM που κρατάνε τις εντολές και τα δεδομένα της μνήμης, ένα module που κρατάει τα 8-LSB και κάνει zero filling τα υπόλοιπα bit ενός 32bit αριθμού, που μας χρησιμεύει στις εντολές Iw, sw και πολυπλέκτες ώστε να κάνουμε control αν θελουμε τα 8LSB ή ολο τον 32bit αριθμό ή στην περίπτωση Ii,lui εντολής να περνάμε στον Α τελεσταίο της ALU τον αριθμό 0 και όχι το RF\_A. Τα επιπλέον control σήματα που χρησιμοποιούμε είναι τα DATA\_din\_sel, MEMdata\_sel, ALU\_alu\_sel, IR\_EN.



## Κυματομορφές



Το πρόγραμμα που εκτελείται στο παραπάνω simulation είναι το εξής:

addi r5,r0,8

ori r3,r0,ABCD

sw r3,4(r0)

lw r10,-4(r5)

lb r16,4(r0)

nand r4,r10,r16

not r1, r5

ror r2,r4

bne r5,r5,8

b -2

add r1,r2,r3

Παρατηρούμε την ορθή λειτουργία του κώδικα στο testbench. Όλες οι εντολές αποκωδικοποιούνται σωστά και εγγράφονται στην Register File ή στην μνήμη τα αποτελέσματα τους. Συγκεκριμένα στο σημείο των branches μπορούμε να δούμε και το infinite loop όπου το PC ανανεώνεται σωστά και επιστρέφει στην εντολή bne.

#### **BONUS: CONTROL**

Υλοποιήσαμε σε ένα βασικό επίπεδο το control της cpu, αλλά δεν προλάβαμε να το τεστάρουμε επαρκώς ώστε να τηρεί της προϋποθέσεις ενός multicycle MIPS processor. Παρόμοια διαδικασία κάναμε και στο testbench του datapath, καθώς η εντολή δεν εκτελείται σε ένα κύκλο αλλά πρώτα γίνεται το instuction fetch και μετά το decode. Προφανώς για λόγους μεγέθους του testbench δεν δείξαμε όλα τα stages της εκτέλεσης μια εντολής σε multicycle. Θα χρειαζόταν επίσης να προσθέσουμε επιπλέον καταχωρητές ώστε να έχουμε στον επόμενο παλμό του ρολογιού το αποτέλεσμα της προηγούμενης βαθμίδας. Το να εκτελούμε ένα stage την φορά θα μας βοηθήσει στο να κάνουμε pipelining ρίχνοντας τον συνολικό χρόνο που χρειάζονται πολλαπλές εντολές για να εκτελεστούν, αφού τα stages ανάμεσα στις εντολές μπορούν πλέον να επικαλύπτονται.

Οργάνωση Υπολογιστών ~ ΗΡΥ 312