

# ΨΗΦΙΑΚΑ ΣΥΣΤΗΜΑ-HW1

ΕΡΓΑΣΤΗΡΙΑΚΕΣ ΑΣΚΗΣΕΙΣ

ΟΝΟΜΑ: Αγγελική

ΕΠΩΝΥΜΟ: Στρατάκη

ΑΕΜ: 10523

ΗΜΕΡΟΜΗΝΙΑ: ΙΑΝΟΥΑΡΙΟΣ 2024

ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ 2023-2024

## ΑΣΚΗΣΗ 1:

Για την άσκηση αυτή έχουμε το αρχείο `alu.v`, όπου υλοποιεί να μοντέλο για μια Μονάδα Λογικών Αριθμητικών (ALU - Arithmetic Logic Unit), εκτελώντας διάφορες αριθμητικές και λογικές λειτουργίες βασιζόμενη σε μια 4-bit είσοδο ελέγχου (`alu_op`).

## ΑΣΚΗΣΗ 2:

Για την άσκηση αυτή έχουμε τρία αρχεία:

### 1) `cacl.v` :

Το αρχείο αυτό περιγράφει έναν ψηφιακό υπολογιστή (calculator) με δυνατότητα πρόσθεσης, αφαίρεσης και επιλογής λειτουργίας μέσω κουμπιών. Τα αποτελέσματα εμφανίζονται στα LEDs.

### 2) `cacl_eng.v` :

Το αρχείο αυτό περιγράφει έναν κωδικοποιητή λειτουργιών (encoder). Ο κωδικοποιητής αυτός μετατρέπει την κατάσταση των τριών κουμπιών (`btnl`, `btnc`, `btnr`) σε έναν 4-bit κωδικό (`alu_op`), ο οποίος χρησιμοποιείται στον ψηφιακό υπολογιστή για τον έλεγχο της λειτουργίας της ALU. Κάθε bit του `alu_op` αντιπροσωπεύει μια συγκεκριμένη λειτουργία, και οι λογικές εξισώσεις έχουν σχεδιαστεί έτσι ώστε να επιτυγχάνουν τον καθορισμό τους με βάση την κατάσταση των κουμπιών.

Με βάση τα σχήματα 2, 3, 4, 5 θα έχουμε τα εξής:

$$\text{alu\_op}[0] = ((\sim \text{btrn}) \& \text{btnl}) \mid ((\text{btnl} \wedge \text{btnc}) \& \text{btrn})$$

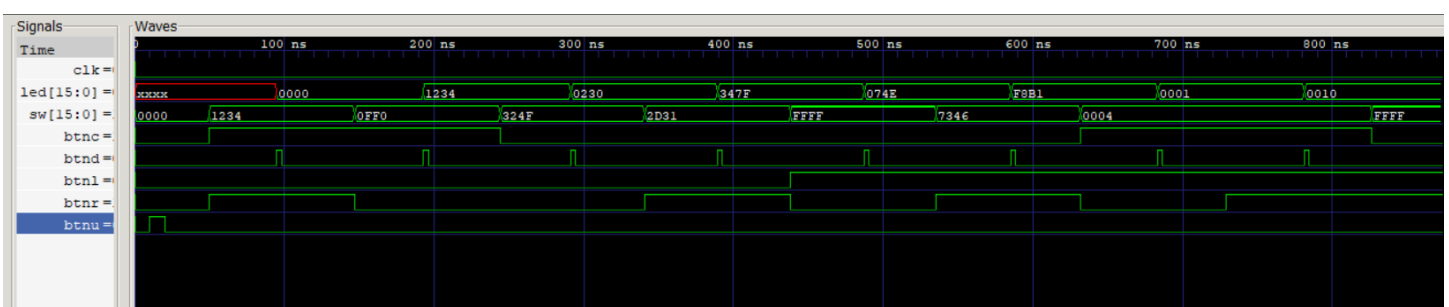
$$\text{alu\_op}[1] = (\text{btrn} \& \text{btnl}) \mid ((\sim \text{btnl}) \& (\sim \text{btnc}))$$

$$\text{alu\_op}[2] = ((\text{btrn} \& \text{btnl}) \mid (\text{btrn} \wedge \text{btnl})) \& (\sim \text{btnc})$$

$$\text{alu\_op}[3] = (((\sim \text{btrn}) \& \text{btnc}) \mid (\sim (\text{btrn} \wedge \text{btnc}))) \& \text{btnl}$$

### 3) `cacl_tb.v` :

Το αρχείο αυτό περιγράφει ένα testbench (περιβάλλον ελέγχου) για τον ψηφιακό υπολογιστή (calc). Το testbench παράγει ένα αρχείο VCD (simulation.vcd) που μπορεί να χρησιμοποιηθεί για την προβολή και ανάλυση των αλλαγών σήματος κατά την προσομοίωση. Επίσης, δοκιμάζει διάφορες λειτουργίες του ψηφιακού υπολογιστή με τη χρήση της υπορουτίνας `test`. Έτσι, δημιουργούμε την εξής κυματομορφή:



### ΑΣΚΗΣΗ 3:

Για την άσκηση αυτή έχουμε το αρχείο `regfile.v`, το οποίο υλοποιεί αρχείο καταχωρητών (register file) παρέχοντας έναν μηχανισμό εγγραφής/ανάγνωσης για έναν πίνακα καταχωρητών, με έλεγχο για να αποφευχθεί η εγγραφή στις ίδιες διευθύνσεις που διαβάζονται.

### ΑΣΚΗΣΗ 4:

Για την άσκηση αυτή έχουμε το αρχείο `datapath.v`, που αφορά τον ψηφιακό δρόμο δεδομένων (datapath) ενός υπολογιστή. Ο κώδικας περιγράφει τον τρόπο λειτουργίας του `datapath` και του συστήματος εκτέλεσης εντολών, συμπεριλαμβανομένων των βασικών συνιστωσών του, όπως ο προγραμματιστικός μετρητής (PC), το αρχείο καταχωρητών, η ALU, κ.ά.

### ΑΣΚΗΣΗ 5:

Για την άσκηση αυτή έχουμε δύο αρχεία:

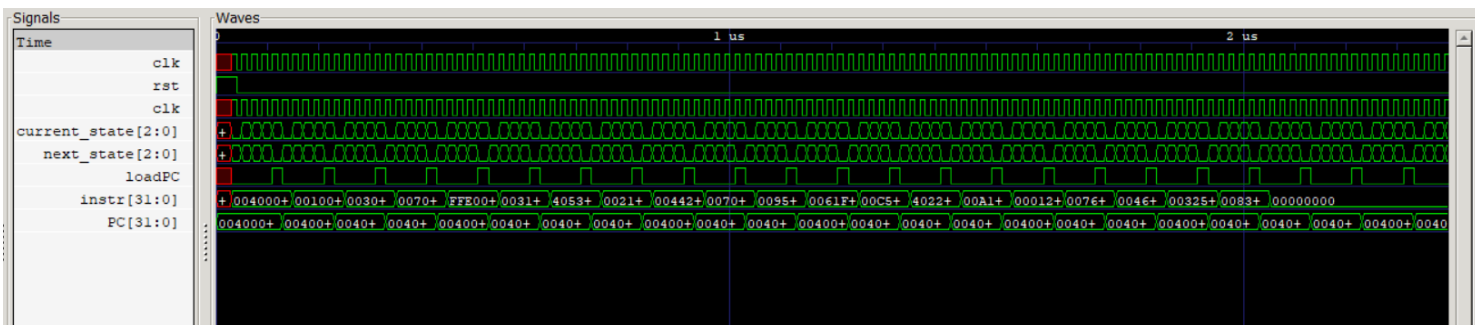
1) `top_proc.v` :

Ο κώδικας αυτού του αρχείου περιγράφει έναν συνολικό ψηφιακό υπολογιστή με μια σειρά από στοιχεία όπως η μονάδα δεδομένων (datapath) και ένας αυτόματος ελεγκτής κατάστασης (FSM - Finite State Machine). Ο αυτόματος ελεγκτής κατάστασης αναλαμβάνει να διαχειριστεί τη ροή των εντολών και την αλλαγή καταστάσεων του συστήματος εκτέλεσης. Συνολικά, ο κώδικας παρέχει μια περιγραφή ενός απλού υπολογιστή RISC με στοιχεία εκτέλεσης και ελέγχου κατάστασης.

2) `top_proc_tb.v` :

Το αρχείο αυτό περιγράφει ένα testbench (περιβάλλον ελέγχου) του top-level-module, Ορίζει ένα κύκλο ρολογιού (clk), ένα σήμα επαναφοράς (rst), καθώς και ορισμένα άλλα σήματα εισόδου και εξόδου που συνδέονται με τον `top_proc`. Επιπλέον, δύο ακόμη μνήμες (RAM και ROM) προσομοιώνονται με τα `ram.v` και `rom.v` αντίστοιχα. Οι μνήμες αυτές χρησιμοποιούνται για την προσομοίωση της RAM (DATA MEMORY) και της ROM (INSTRUCTION MEMORY) που θα

χρησιμοποιούνται σε έναν πραγματικό υπολογιστή. Έτσι, δημιουργούμε την εξής κυματομορφή:



Επιπλέον, το σχηματικό διάγραμμα από το FSM της άσκησης είναι το εξής:

