



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ  
ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΩΝ & ΥΛΙΚΟΥ  
ΗΡΥ 312: ΟΡΓΑΝΩΣΗ ΥΠΟΛΟΓΙΣΤΩΝ  
ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2019-2020

## Εργασία #2: Δημιουργία επεξεργαστή πολλαπλών κύκλων και μετατροπή του σε pipeline

Ομάδα εργασίας LAB31243764

Γκογκολάκη Ελένη 2012030071

Ραφαήλ Τσιριβάκος 2013030199

### 4η φάση: «Δημιουργία επεξεργαστή πολλαπλών κύκλων»

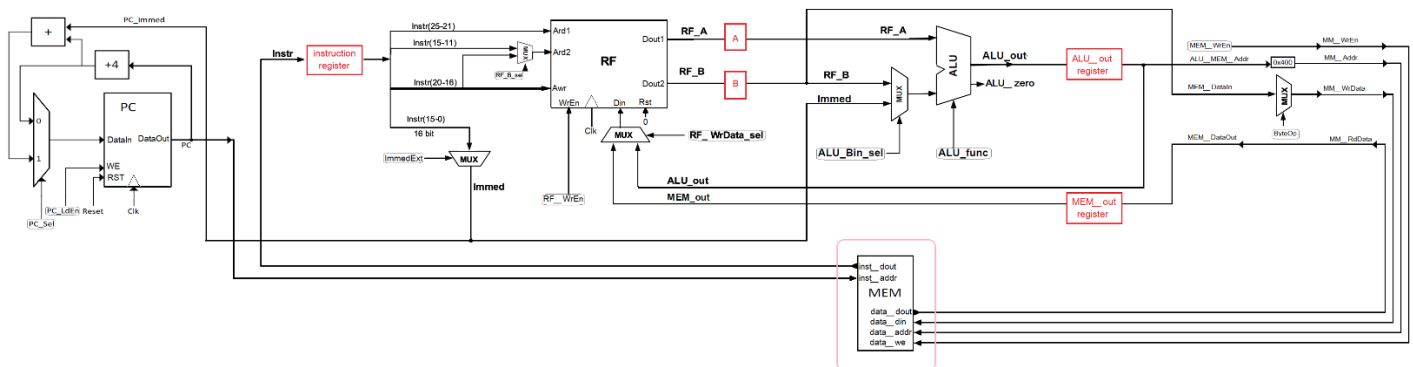
#### Σκοπός

Ο σκοπός της 4ης φάσης είναι να μετατρέψουμε τον επεξεργαστή ενός κύκλου που σχεδιάσαμε στην Εργασία#1 σε επεξεργαστή πολλαπλών κύκλων.

#### Διεξαγωγή

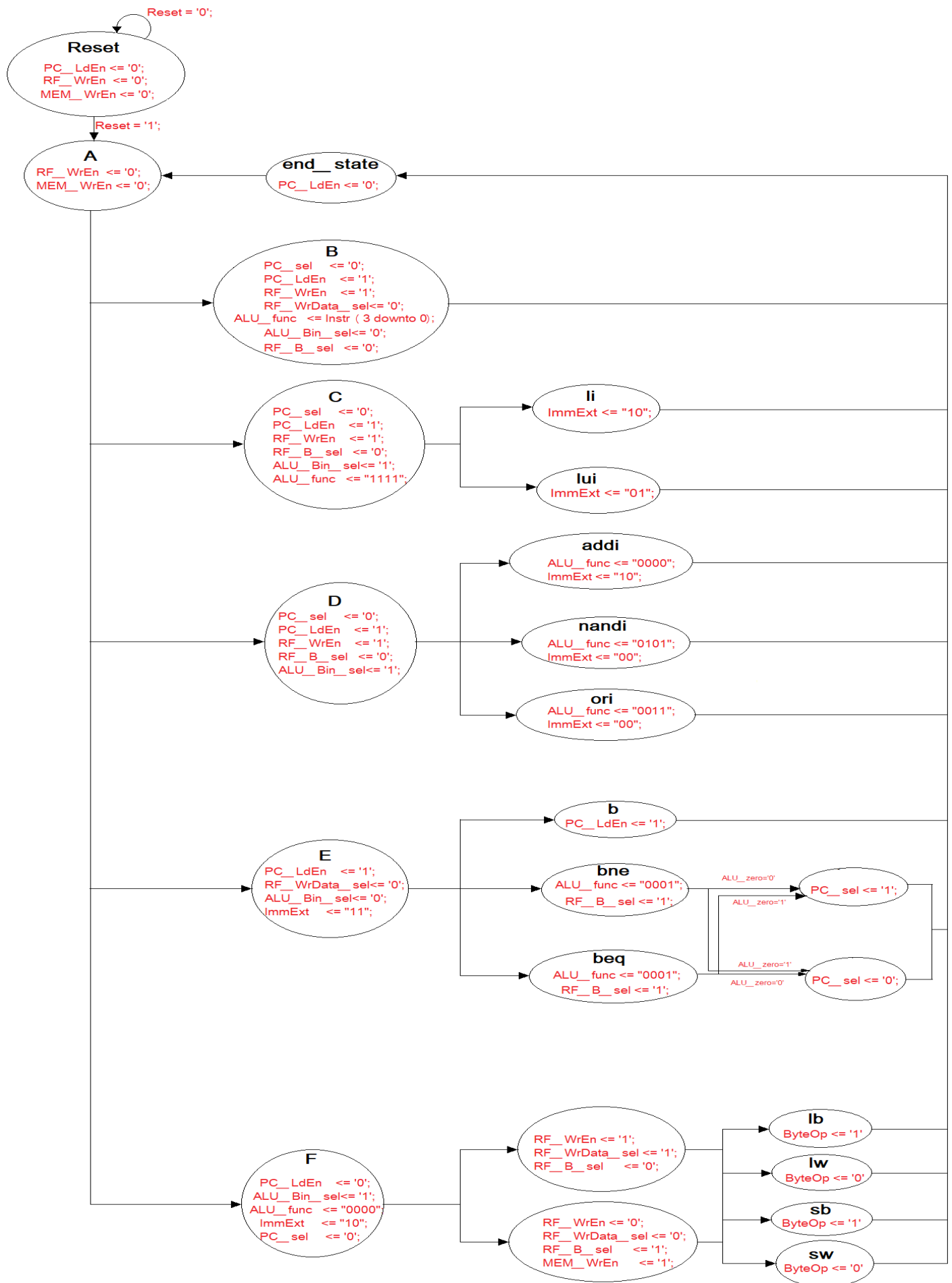
##### A) Μετατροπή του datapath του επεξεργαστή ενός κύκλου σε datapath πολλαπλών κύκλων

Στο datapath που κατασκευάσαμε στην Εργασία#1 και προσθέσαμε καταχωρητές μεταξύ των βαθμίδων όπου χρειάζεται ώστε να κρατάμε τις τιμές για κάποια σήματα που παράγονται από μια βαθμίδα και πρέπει να χρησιμοποιηθούν σε επόμενη βαθμίδα. Το τελικό datapath φαίνεται στο παρακάτω block diagram, όπου με κόκκινο φαίνονται οι επιπλέον καταχωρητές.



##### B) Σχεδιασμός και υλοποίηση της μονάδας ελέγχου (control)

Σχεδιάσαμε τη μονάδα ελέγχου, η οποία είναι μια Μηχανή Πεπερασμένων Καταστάσεων (FSM) που ελέγχει τη ροή εκτέλεσης της κάθε εντολής γεννώντας τα απαιτούμενα σήματα ελέγχου σε κάθε κύκλο ρολογιού. Η FSM έχει σαν είσοδο την εντολή (opcode), το Zero, κ.λπ., και σαν εξόδους όλα τα σήματα ελέγχου του Datapath. Παρακάτω φαίνεται η λογική που ακολουθήσαμε.

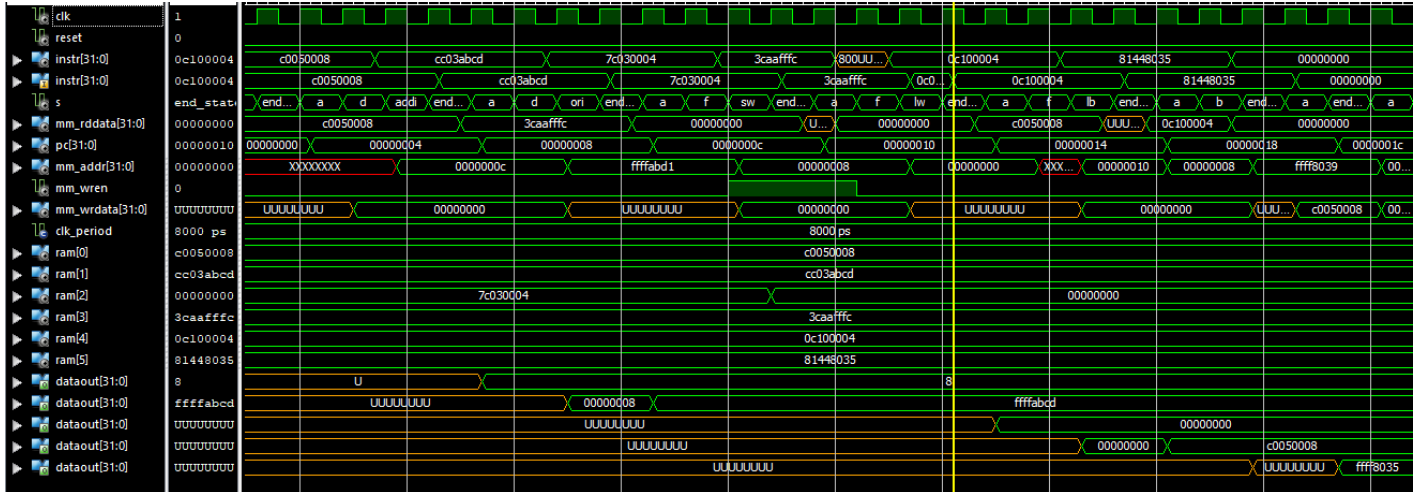


### Γ) Ολοκλήρωση και έλεγχος ορθότητας

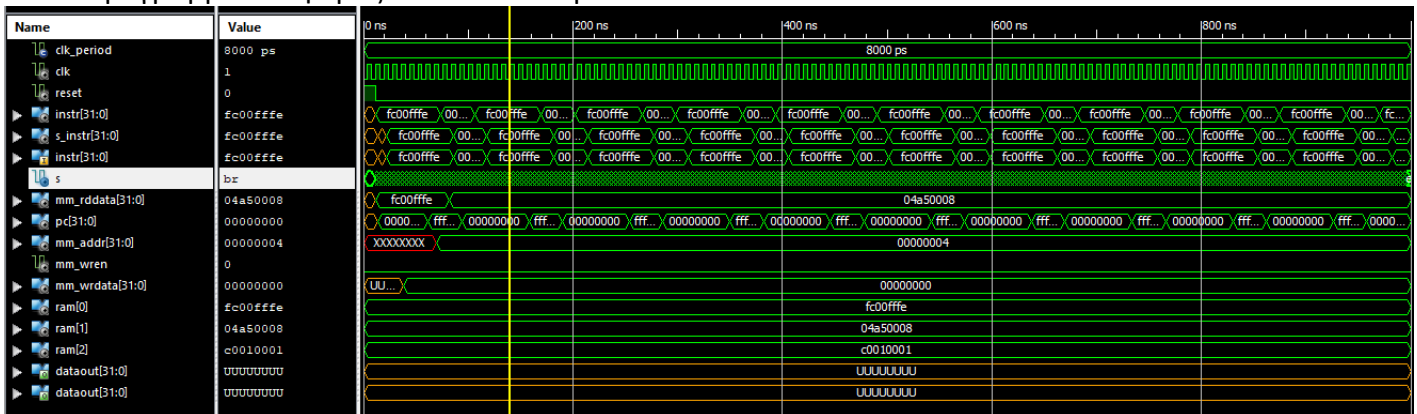
Συνδέσαμε το datapath με το control ώστε να υλοποιήσουμε την πλήρη λειτουργία ενός επεξεργαστή πολλαπλών κύκλων. Η κύρια μνήμη βρίσκεται εκτός αυτού του module.

Παρακάτω έχουμε τα waveforms με τα προγράμματα αναφοράς της Εργασίας#1 και ακόμα ένα το οποίο περιλαμβάνει όλες τις εντολές του ISA από μία φορά.

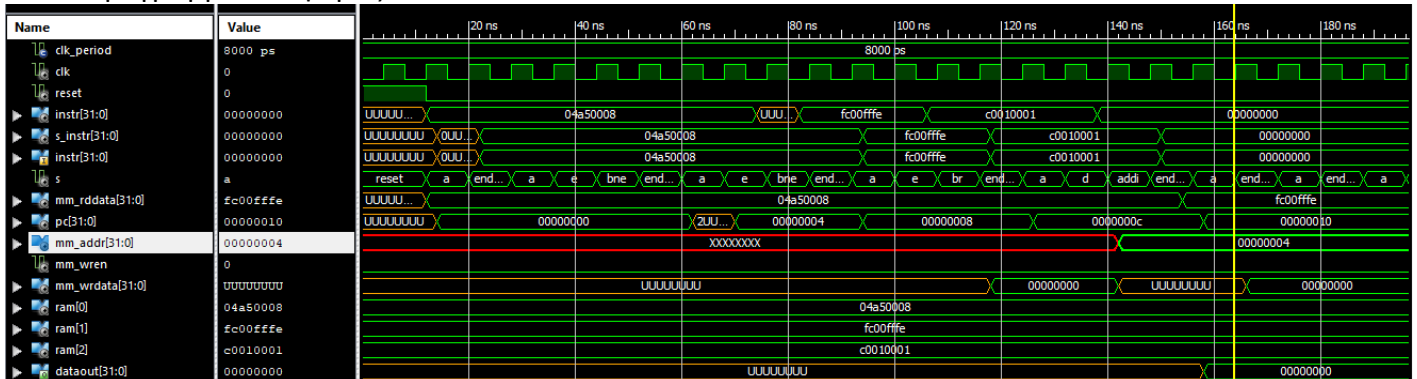
## Πρόγραμμα αναφοράς #1



## Πρόγραμμα αναφοράς #2-infinite loop



## Πρόγραμμα αναφοράς #2



[illegible]

## Β) Σχεδιασμός και υλοποίηση της μονάδας ελέγχου (control)

Σχεδιάσαμε τη μονάδα ελέγχου ώστε να δημιουργεί τα απαιτούμενα σήματα ελέγχου για κάθε εντολή. Χρησιμοποιήσαμε τη μονάδα ελέγχου του επεξεργαστή ενός κύκλου (Εργασία#1) ως σημείο εκκίνησης για τις εντολές `add`, `li`, `sb` και `sw`.

## Γ) Ολοκλήρωση και έλεγχος ορθότητας

Συνδέσαμε το `datapath` με το `control` ώστε να υλοποιήσουμε την πλήρη λειτουργία ενός pipeline επεξεργαστή.

Χρησιμοποιήσαμε ένα δικό μας πρόγραμμα αναφοράς για το `testbench`, το οποίο περιλαμβάνει μόνο τις εντολές `li`, `lw`, `sw` και `add` πολλές φορές ώστε να φανεί η λειτουργία του pipeline και η αντιμετώπιση των κινδύνων δεδομένων. Παρακάτω φαίνεται και το waveform που προκύπτει.

