

# Οργάνωση Υπολογιστών

## 1<sup>η</sup> Εργαστηριακή Αναφορά

Σαββίνα Αυγουστή 2018030200

### Σκοπός Άσκησης:

Σκοπός της άσκησης είναι η σχεδίαση ενός single cycle επεξεργαστή και υλοποιώντας σταδιακά τα κυκλώματα του για την ολοκλήρωση του PROC\_SC.

### 1<sup>η</sup> Φάση

Στην 1<sup>η</sup> φάση υλοποιήθηκε η μονάδα αριθμητικών και λογικών πράξεων ALU, όπου ανάλογα με τον κωδικό Op που δέχεται εκτελεί και την ανάλογη εντολή και στην συνέχεια με την δημιουργία ενός αρχείου καταχωρητών.

#### ALU:

- ✓ Συνδυαστικό κύκλωμα, χωρίς ρολόι και δουλεύει με συμπλήρωμα ως προς 2.
- ✓ Για την υλοποίηση των αριθμητικών και λογικών πράξεων εφαρμόστηκε ένα case-when και δέχεται τους ανάλογους κωδικούς **Op**.
- ✓ Το σήμα εξόδου **Ovf** ενεργοποιείται αν στο αποτέλεσμα υπάρχει υπερχείλιση.
- ✓ Το σήμα εξόδου **Zero** ενεργοποιείται όταν το αποτέλεσμα είναι μηδέν.
- ✓ Το σήμα εξόδου **Cout** ενεργοποιείται αν υπάρχει κρατούμενο εξόδου (carry out) και τοποθετήθηκε στην πράξεις της πρόσθεσης και αφαίρεσης όπου εξάγεται το **coutTemp(32)**.

#### RF:

- ✓ Το αρχείο καταχωρητών αποτελείται από 1 αποκωδικοποιητή (5 σε 32), 2 πολυπλέκτες (32 σε 1) και από 32 καταχωρητές τα οποία έγιναν με component μέσα στο RF.

- ✓ Στο WriteEnable υλοποιήθηκε ένα for-generate και ενώθηκε η έξοδος του με την έξοδο του decoder με μια πύλη AND.
- ✓ Στον MUX δημιουργήθηκε ένα package όπου περιέχει ένα **muxIn** για την υλοποίηση των 32 καταχωρητών στο RF.

```
package mux_pkg is
    type muxIn is array(31 downto 0) of std_logic_vector(31 downto 0);
end package;

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use work.mux_pkg.all;
```

## 2η Φάση

Σε αυτό το μέρος της άσκησης υλοποιήθηκαν τα 4 υποκυκλώματα του **DATAPATH** (**IFSTAGE**, **DECSTAGE**, **EXSTAGE**, **MEMSTAGE**) αλλά και η μνήμη **RAM**.

### RAM:

Η μνήμη χρησιμοποιήθηκε στις βαθμίδες **IFSTAGE** και **MEMSTAGE** με βάση των κώδικα που δόθηκε και αποτελείται από 2048 θέσεις, των 32 bits η κάθε θέση. Επιπλέον τα δεδομένα τα διαβάζει από το αρχείο rom.data.

### IFSTAGE:

- ✓ Αποτελείται από έναν καταχωρητή (PC) πλάτους 32 bits, έναν αθροιστή που αυξάνει κατά 4 (incrementor), για να υπολογίζει την τιμή PC+4, έναν αθροιστή (immediate) που υπολογίζει την τιμή (PC+4) + SignExt(Immed)\*4 για τις εντολές διακλάδωσης, ένα πολυπλέκτη (MUX) 2 σε 1 για να περαστεί η ανάλογη τιμή προς τον PC.
- ✓ Ο επιλογέας **PC\_sel** όταν είναι 0 πηγαίνει στον **Adder\_Inc**, ενώ όταν είναι 1 πηγαίνει στον **Adder\_PC\_immed**.
- ✓ Η τιμή του **PC\_LdEn** είναι πάντα 1.

- ✓ Δημιουργήθηκε ένα **IFSTAGE\_TopModule**, στο οποίο ενώθηκε η IFSTAGE με την RAM και εφαρμόστηκε και το αντίστοιχο testbench.

### DECSTAGE:

- ✓ Αποτελείται από το αρχείο καταχωρητών, έναν πολυπλέκτη 2 σε 1 (32 bits), ένα πολυπλέκτη 2 σε 1 (5 bits) και μια μονάδα που δέχεται σαν είσοδο 16 bits του immediate και την μετατρέπει σε 32 bits.
- ✓ Το συννεφάκι (CONV) επιλέγει αν θα γίνει zero-filling (**ImmExt = 00**), zero-filling with shift (**ImmExt = 01**), sign-extend (**ImmExt == 10**) ή sign-extend with shift (**ImmExt = 11**).
- ✓ Το RF\_WrData\_sel είναι ενεργό όταν η προέλευση των δεδομένων προς εγγραφή προέρχεται από την MEM, ενώ όταν είναι απενεργοποιημένο προέρχεται από την ALU.

### EXSTAGE:

- ✓ Αποτελείται από την ALU και ένα πολυπλέκτη 2 σε 1.
- ✓ Το ALU\_Bin\_sel επιλέγει αν ο πολυπλέκτης θα πάρει είσοδο από το RF\_B ή από το Immediate.
- ✓ Η ALU\_func επιλέγει ποιά πράξη θα υλοποιηθεί από την ALU.

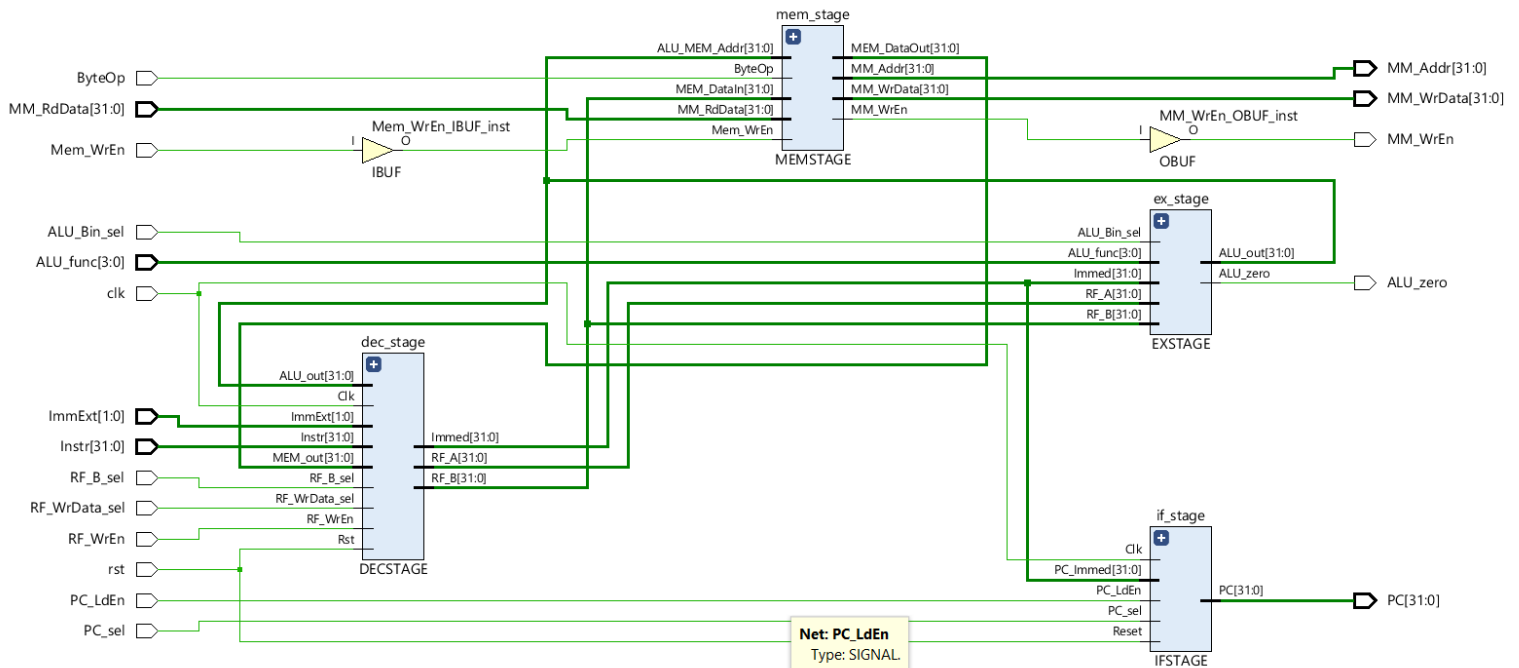
### MEMSTAGE:

- ✓ Ο επεξεργαστής για να διαβάσει και να γράφει τα δεδομένα από το σωστό τμήμα, προστίθεται κατά 0x400 (offset) η διεύθυνση που στέλνει η ALU.
- ✓ Το σήμα ελέγχου ByteOp ενεργοποιείται για lb/sb ή έχει τιμή μηδέν για lw/sw.
- ✓ Υλοποιήθηκε ένα MEMSTAGE\_TopModule που έχει ως component το MEMSTAGE και την RAM.

## 3η Φάση

### DATAPATH:

- ✓ Δέχεται ως component τα υποκυκλώματα (IFSTAGE, DECSTAGE, EXSTAGE, MEMSTAGE) και με εσωτερικά σήματα γίνεται η ένωση τους.

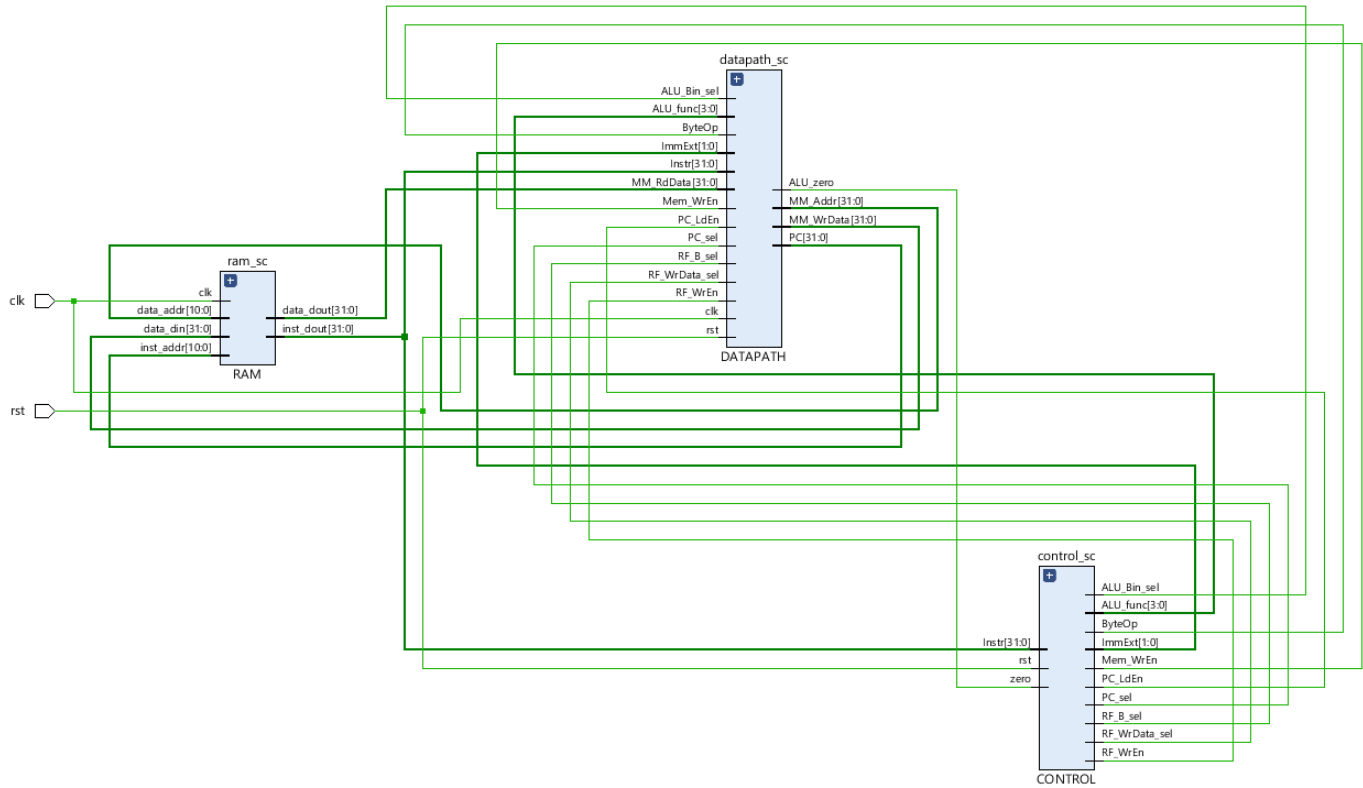


### CONTROL:

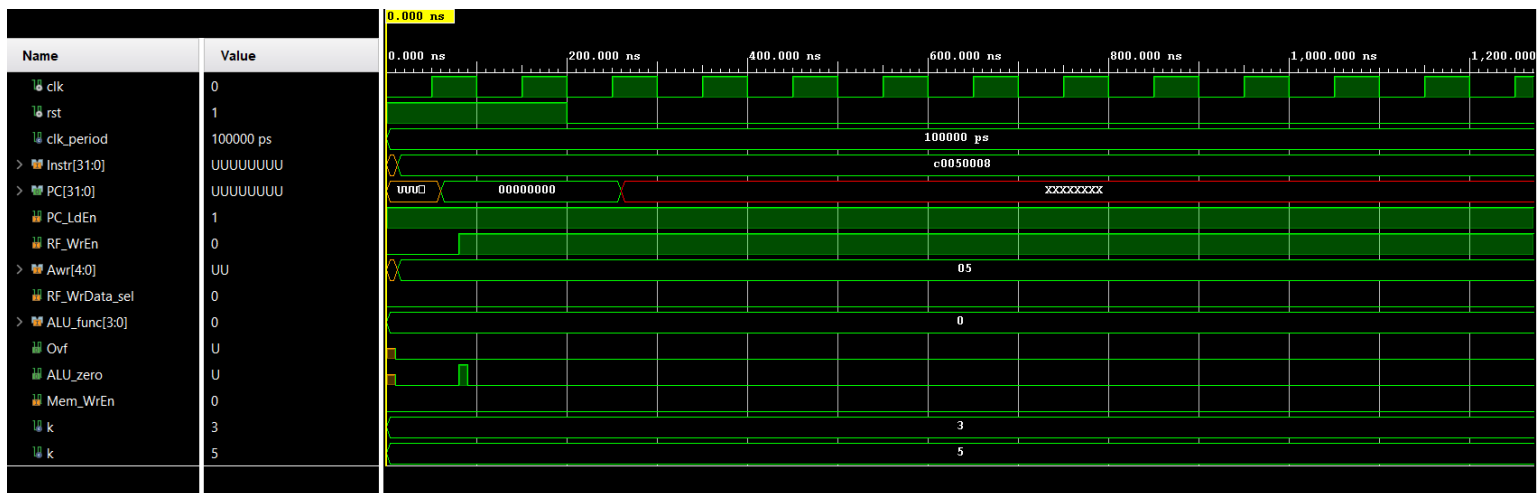
- ✓ Μέσα στο control ορίστηκε ένα σήμα **Op** όπου παίρνει από 31 μέχρι 26 bits του **Instr** ανάλογα με τον κώδικα που δίνει το πινακάκι έτσι ώστε να εκτελεστούν η ανάλογες εντολές.
- ✓ Το **PC\_sel** είναι ενεργό μόνο στις εντολές **b, bne, beq**.
- ✓ Το **ByteOp** είναι 1 στις εντολές **lb/sb**, είναι 0 στις εντολές **lw/sw** και στα υπόλοιπα δεν μας νοιάζει τι τιμές παίρνει (don't care).
- ✓ Το **ALU\_Bin\_sel** είναι 0 μόνο στις εντολές r-type και στις εντολές branch.
- ✓ Το σήμα **MEM\_WrEn** είναι 1 μόνο στις εντολές **sb/sw**.
- ✓ Το σήμα **RF\_B\_sel** είναι ενεργό στις εντολές **b, bne, beq, sb, sw**.

## PROC SC:

Σε αυτό το μέρος συνδέεται το Control, Datapath και την RAM.



## Πρόγραμμα Αναφοράς #1



### Προβλήματα:

Στο PROC\_SC δεν βγάζει σωστά τα σήματα για rom.data και rom1.data αλλά το κύκλωμα συνδεσμολογίας λειτουργεί.