

# Αναφορά δεύτερου μέρους εργασίας Οργάνωσης υπολογιστών

Ιωαννίδης Χρήστος 2018030006

Το παρόν project υλοποιήθηκε με την χρήση του Xilinx ISE 13.4

## Γενικές πληροφορίες:

- Όλα τα Modules που ζητούνται από το Multi Cycle έχουν υλοποιηθεί και φαίνεται να παρουσιάζουν την επιθυμητή λειτουργικότητα.
- Δεν έχει υλοποιηθεί το κομμάτι με το pipelining.
- Στα Modules δεν έχει χρησιμοποιηθεί η εντολή after που προσομοιώνει την καθυστέρηση ενός σήματος στο υλικό του επεξεργαστή.
- Εφόσον μας ζητήθηκε να έχουμε την μνήμη εκτός του Module PROCESSOR\_MC δημιουργήθηκε το Module PROCESSOR\_MC\_MEM έτσι ώστε να μπορέσουν να γίνουν δοκιμές πάνω στο πρόγραμμα του πρώτου μέρους και του δικού μου προγράμματος. **Πάνω σε αυτό το module έχει δημιουργηθεί και το testbench.**
- Κατά την προσομοίωση η μνήμη διαβάζει το αρχείο rom.data το οποίο περιέχει το πρόγραμμα του πρώτου μέρους, για να διαβάζει το άλλο πρόγραμμα πρέπει να γίνει αντικατάσταση στον κώδικα της μνήμης με το αρχείο rom1.data.

## Σημεία που χρήζουν προσοχής:

- Έχει γίνει μια μικρή διόρθωση μέσα στο IFSTAGE στο σήμα PC\_Immed το οποίο είχε παραληφθεί να πολλαπλασιαστεί \*4 πριν μπει στον αθροιστή +4 ο οποίος εν τέλει καταλήγει στο PC με αποτέλεσμα οι εντολές Branch να μην έχουν την επιθυμητή λειτουργία.

Παρακάτω παρουσιάζεται ο κώδικας πριν και μετά.

### ΠΡΙΝ

```
immed:PC_Immed_Adder
    PORT MAP(PC_Immed=>PC_Immed, PC_plus_four=>plus4_out
    ,output=>immed_out);
```

### ΜΕΤΑ

Προστέθηκε το σήμα: `signal PC_Immed_tmp:STD_LOGIC_VECTOR (31 downto 0);`

Με:

```
begin
```

```
PC_Immed_tmp<=PC_Immed(29 downto 0)&"00";
```

```
immed:PC_Immed_Adder
```

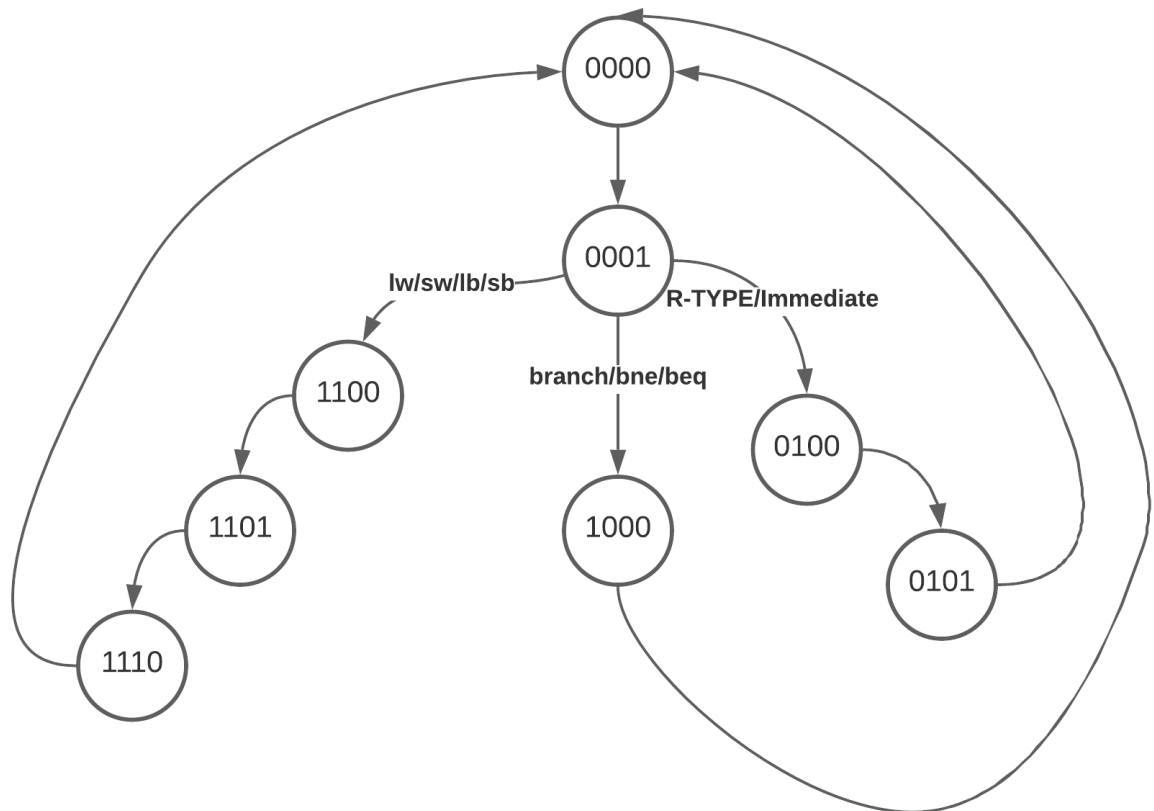
```
PORT MAP(PC_Immed=>PC_Immed_tmp,PC_plus_four=>plus4_out  
,output=>immed_out);
```

## Multi-Cycle

- Για την υλοποίηση του επεξεργαστή πολλαπλών κύκλων προστέθηκαν 5 καταχωρητές στο datapath(RF\_A,RF\_B,ALU\_OUT,INSTR,MEM\_DATA) ενώ χρειάστηκε άλλος ένας για την αποθήκευση του State μέσα στο Control\_MC\_FINAL.
- Το Control του πρώτου μέρους τροποποιήθηκε σε FSM προκειμένου να μπορέσει να υλοποιήσει την επιθυμητή λειτουργικότητα. Η κωδικοποίηση των καταστάσεων παρουσιάζεται στο διάγραμμα παρακάτω. Επιπλέον προστέθηκε ένα ακόμα σήμα εξόδου το οποίο λειτουργεί ως Enable για τον καταχωρητή Instruction του Datapath.

Παρακάτω παρουσιάζεται το διάγραμμα καταστάσεων της FSM

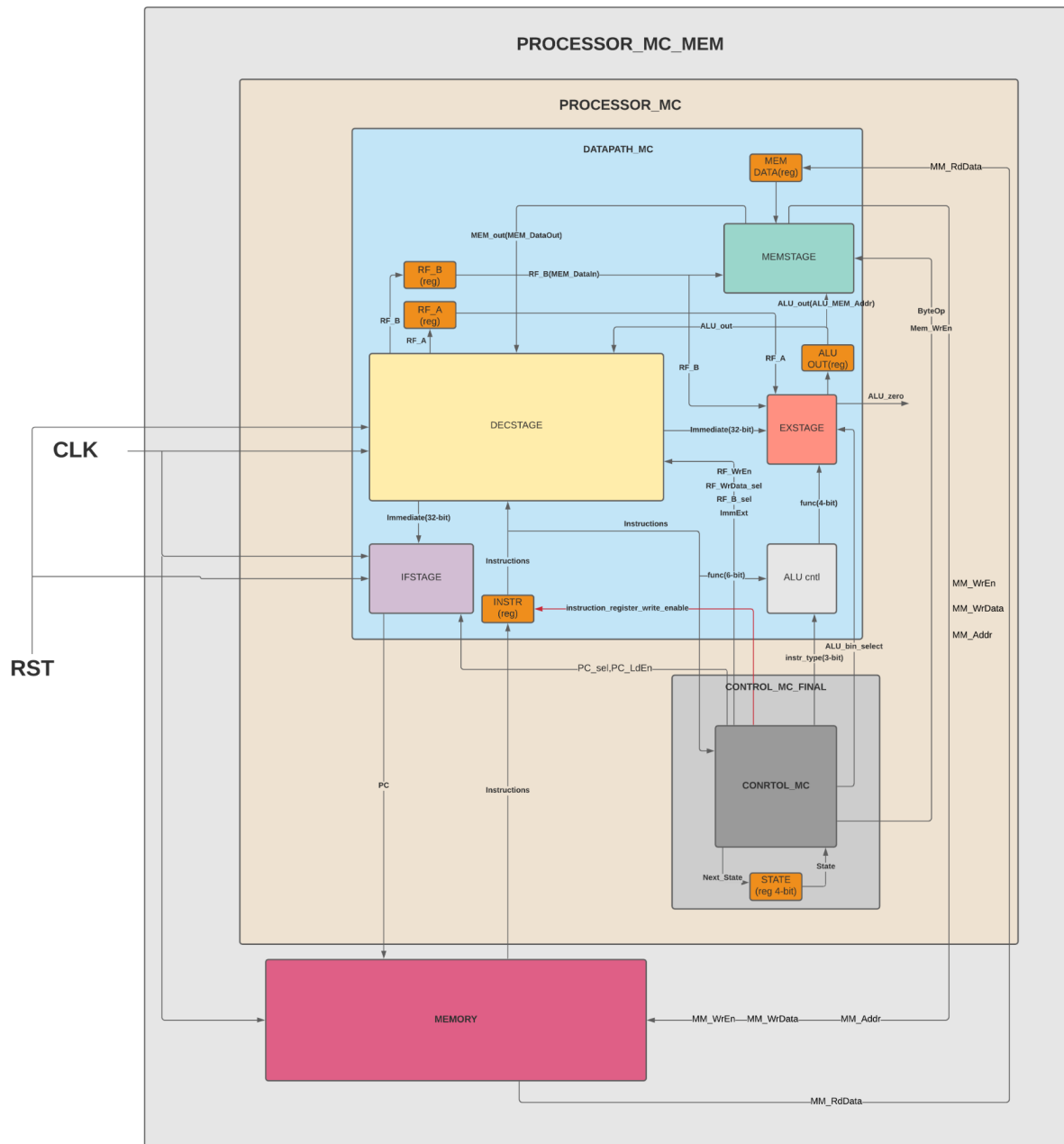
CONTROL\_MC\_STATE\_DIAGRAM



Στο Παρακάτω Block Diagram παρουσιάζεται μια χονδρική απεικόνιση του συνολικού κυκλώματος με τα κυριότερα Modules και σήματα καθώς και οι επιμέρους συνδέσεις τους.

Στην εικόνα φαίνονται ευκρινώς όλα τα σήματα με την χρήση μεγέθυνσης

## Multi-Cycle Diagram



## Multi-Cycle Προγραμμα TESTBENCH

11000000000001010000000000001000	addi r5,r0,8
11000000000001100000000000000111	addi r6,r0,7
10000000101001110011000000110001	sub r5,r7,r6 (8-7=1)
10000000101001110011000000110010	and r5,r7,r6
1000000010100000000000000111010	sll r5
11001100000000111010101111001101	ori r3,r0,ABCD
01111100000000110000000000000100	sw r3,4(r0)
00111100101010101111111111111100	lw r10,-4(r5)
00001100000100000000000000000100	lb r16,4(r0)
10000001010001001000000000110101	nand r4,r10,r16
000001001010010100000000000001000	bne r5,r5,8
11111100000000001111111111111110	b -2
11000000000000010000000000000001	addi r1,r0,1

### Simulation screenshot

Στην εικόνα φαίνονται ευκρινώς όλα τα σήματα με την χρήση μεγέθυνσης

