

Σχεδίαση επεξεργαστή πολλαπλών κύκλων

1. Μετατροπή του datapath του επεξεργαστή ενός κύκλου σε datapath πολλαπλών κύκλων

- Αρχικά για την υλοποίηση του datapath για χρήση σε επεξεργαστή πολλών κύκλων, αφαιρείτε το IF stage καθώς θα χρειαστούμε μόνο τον καταχωρητή PC. Ποιο συγκεκριμένα ως είσοδός του είναι η έξοδος του πολυπλέκτη που επιλέγει την έξοδο της ALU ή του καταχωρητή που αποθηκεύει την έξοδο της ALU. Ο πολυπλέκτης επιλέγει την επόμενη εντολή ($PC + 4$) ή εντολή από branch ($PC + 4 + \text{Immed value}$) ανάλογα με την περίπτωση.
- Στην συνέχεια τοποθετείτε ο Instruction register ο οποίος κρατά την εντολή που έχει διαβαστεί από την μνήμη και η έξοδος θα πηγαίνει στο Decode stage αλλά και στο Control.
- Κατά το Decode Stage, το οποίο παραμένει ίδιο σε σχέση με το επεξεργαστή ενός κύκλου, γίνεται το decode της εντολής, σύμφωνα με τιμές ελέγχου του Control και οι έξοδοι αυτού αποθηκεύονται σε 2 καταχωρητές, τους Reg A και Reg B οι οποίοι κρατούν τους καταχωρητές, από το Register file, που χρειάζονται για κάθε εντολή. Τέλος το Immediate, που έχει υπολογιστεί πηγαίνει στο τροποποιημένο Dec Stage για να χρησιμοποιηθεί ανάλογα.
- Το Execute Stage έχει τροποποιηθεί. Ως είσοδοι της ALU είναι οι έξοδοι 2 πολυπλεκτών εκ των οποίων ο πρώτος επιλέγει αν η είσοδος της ALU είναι η διεύθυνση που κρατά ο Program Counter, ώστε να γίνει increment, είτε την τιμή του καταχωρητή A. Ο δεύτερος πολυπλέκτης έχει ως εισόδους την τιμή του καταχωρητή B είτε την τιμή +4, ώστε να πάρουμε την διεύθυνση της επόμενης εντολής, είτε το Immediate value που υπολογίστηκε κατά το Execute Stage για να αποθηκευθεί η διεύθυνση της εντολής που χρειάζεται. Τα αποτελέσματα της ALU αποθηκεύονται στον καταχωρητή Reg ALU αλλά και στην πρώτη θέση του πολυπλέκτη του PC.
- Για τον Reg ALU, η έξοδος του οποίου καταλήγει ως δεύτερη είσοδος του πολυπλέκτη του Decode Stage, ως δεύτερη είσοδος του πολυπλέκτη του PC αλλά και στην μνήμη.
- Τέλος κατά το Memory Stage η εντολή που διαβάστηκε αποθηκεύεται στο Instruction register αλλά και τα δεδομένα τα οποία διαβάστηκαν αποθηκεύονται στον Reg MEM. Από το Reg MEM τα δεδομένα αποθηκεύονται στην πρώτη θέση του πολυπλέκτη του Decode Stage.

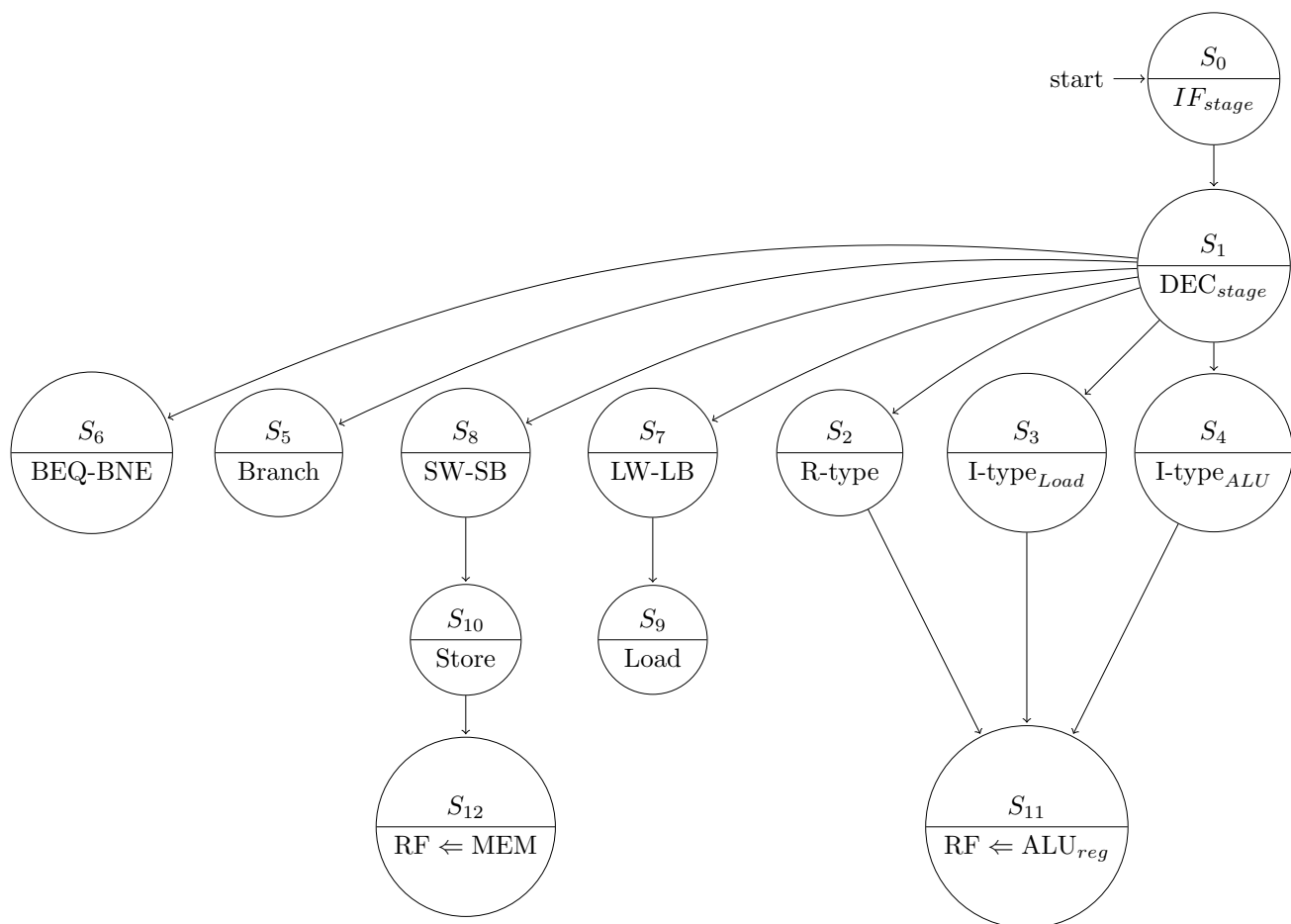
2. Σχεδιασμός και υλοποίηση της μονάδας ελέγχου (control)

Για την υλοποίηση του Control σχεδιάστηκε η FSM που ελέγχει τις καταστάσεις που πρέπει να ακολουθούν ανάλογα με την κάθε εντολή. Ως είσοδοι του Control είναι το OpCode και το Func από την εντολή που κρατά ο Instruction Register αλλά και το σήμα, flag, ALU zero από το Execute Stage.

Τα στάδια είναι τα εξής:

- Καταγραφή της επόμενης εντολής στον Instruction Register και αύξηση του PC.
- Decode Stage όπου γίνεται το Decode της εντολής και αποθηκεύονται στους registers A και B οι τιμές της εκάστοτε εντολής.
- Κατά το στάδιο του Execute υπολογίζεται η έξοδος της ALU, που θα αποθηκευθεί στον Reg ALU. Η τιμή που μπορεί να αποθηκευθεί είναι αποτελέσματα πράξης, είτε αριθμητικής είτε λογικής, αλλά και την τιμή της διεύθυνσης που θα έχει ο PC.
- Σε επόμενο στάδιο, αποθηκεύονται ή διαβάζονται δεδομένα από την μνήμη για την περίπτωση των εντολών SW/SB και LW/LB αντίστοιχα.
- Τέλος σε περίπτωση που πρέπει να αποθηκευθεί τιμή σε καταχωρητή γράφουμε στο Register File την τιμή που κρατά ο Reg ALU είτε ο Mem Reg από την μνήμη.

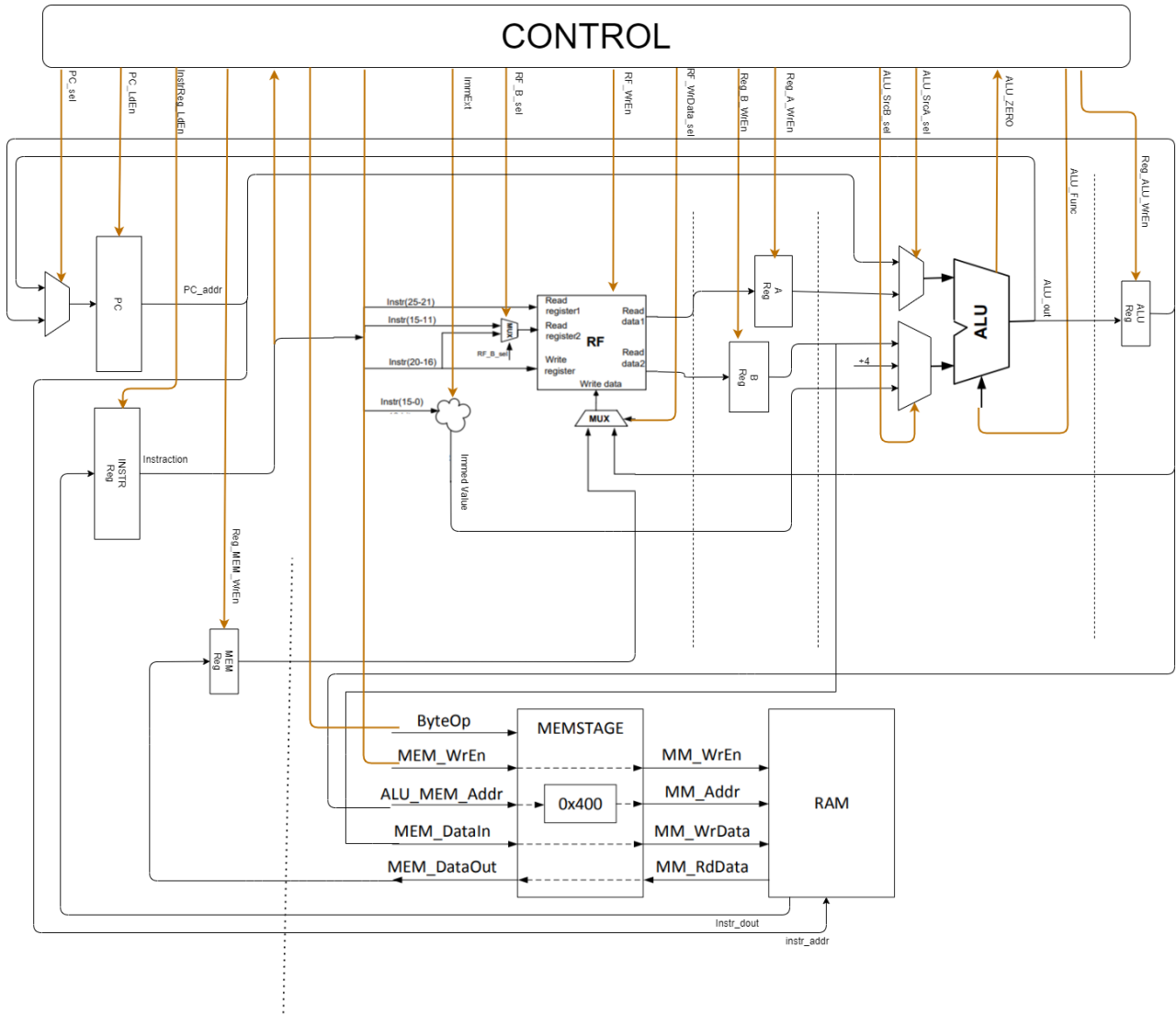
Το State machine που υλοποιήθηκε με τις αντίστοιχες καταστάσεις και στάδια:



3. Ολοκλήρωση και έλεγχος ορθότητας

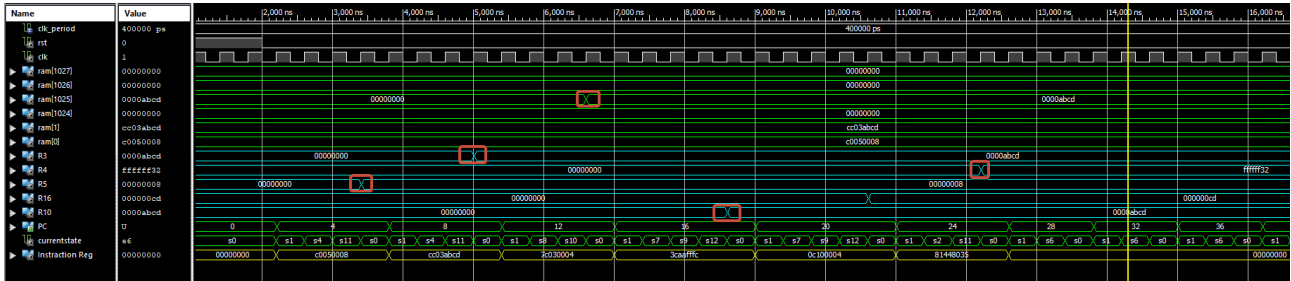
Σε τελικό στάδιο ολοκληρώθηκε ο επεξεργαστής πολλών κύκλων χωρίς την μνήμη, PROCESSOR-MC αλλά και με την μνήμη, PROCESSOR-MC-RAM μέσω του οποίου ελέγχθηκε η ορθότητα της σχεδίασης για τα προγράμματα αναφοράς της πρώτης φάσης αλλά και για πρόγραμμα που περιέχει όλες τις εντολές.

Block Diagram του επεξεργαστή πολλών κύκλων

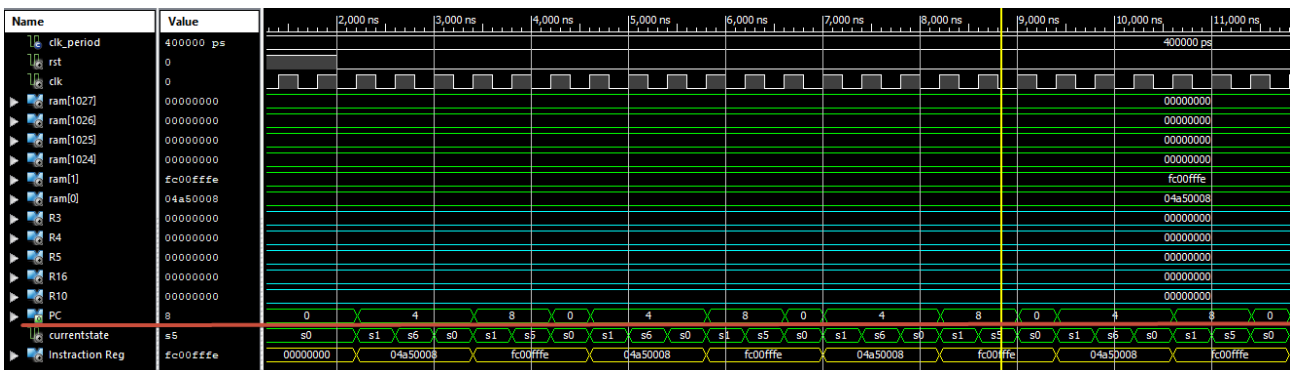


Προγράμματα αναφοράς

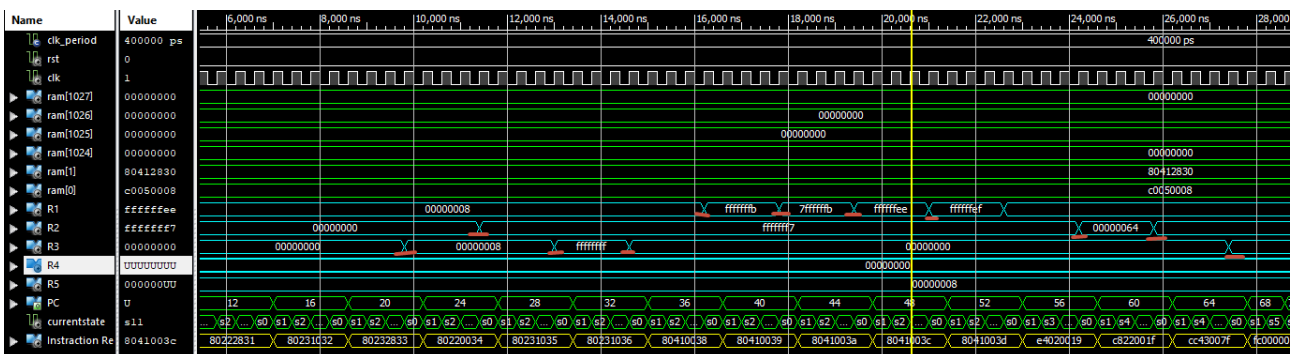
1. (a) `addi r5,r0,8;`
 (b) `ori r3,r0,ABCD;`
 (c) `sw r3,4(r0);`
 (d) `lw r10,-4(r5);`
 (e) `lb r16,4(r0);`
 (f) `nand r4,r10,r16;`



2. (a) `bne r5,r5,8`
 (b) `b -2`
 (c) `addi r1,r0,1`

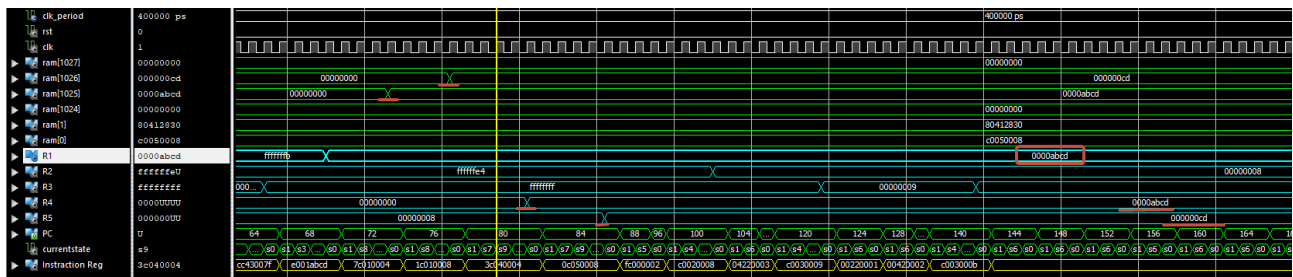


3. Το πρόγραμμα αναφοράς που ελέγχοντε όλες οι εντολές. οι R-type και I-type εντολές: Πραγματοποιούντε



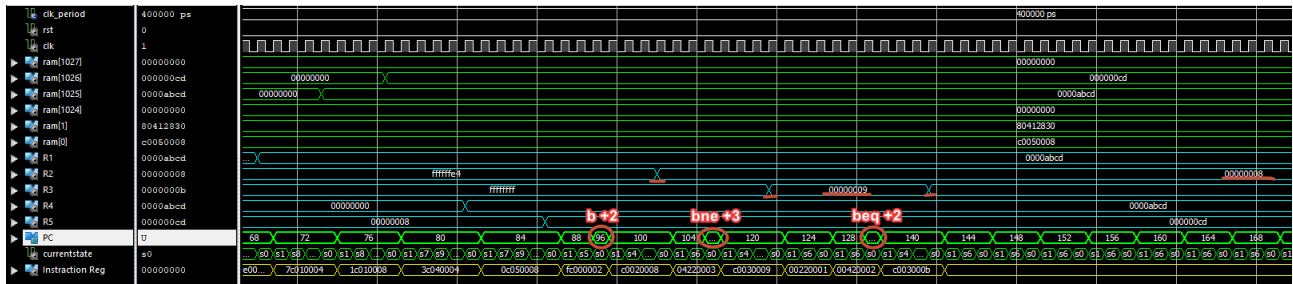
όλες οι πράξεις σύμφωνα με το .data αρχείο.

4. Οι load-store εντολές:



Αποθηκεύονται και διαβάζονται τα δεδομένα από την μνήμη.

5. Οι branch εντολές:



Γίνονται τα σωστά branch στο πρόγραμμα μιας και γίνονται οι σωστές πράξεις μετά από κάθε μεταπήδηση.

Σχεδίαση επεξεργαστή pipeline

Δεν ολοκληρώθηκε λόγω έλλειψης χρόνου.