



**ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ**  
**ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΩΝ & ΥΛΙΚΟΥ**  
**ΕΡΓΑΣΤΗΡΙΑΚΕΣ ΑΣΚΗΣΕΙΣ ΓΙΑ ΤΟ ΜΑΘΗΜΑ:**  
**Οργάνωση Υπολογιστών**  
**ΗΡΥ 302**  
**<http://www.mhl.tuc.gr>**  
**ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2024**

## Αναφορά Εργαστηριακής Άσκησης 2

Αριθμός Ομάδας:66

Λαμπράκης Μιχάλης 2020030077  
Δήμας Χρήστος 2021030183

### 1.Σκοπός της Άσκησης

Σκοπός αυτής της εργαστηριακής άσκησης είναι η μετατροπή του επεξεργαστή μονού κύκλου που σχεδιάστηκε στην 1<sup>η</sup> φάση, σε επεξεργαστή πολλαπλών κύκλων. Η μετατροπή αυτή θα γίνει με την προσθήκη καταχωρητών μεταξύ των βαθμίδων, όπου χρειάζεται, ώστε να διατηρούνται οι τιμές των σημάτων που παράγονται από μια βαθμίδα και πρέπει να χρησιμοποιηθούν. Ακόμη με τη σχεδίαση μιας μηχανής πεπερασμένων καταστάσεων (FSM) ,η οποία αναλαμβάνει τον ρόλο του control του επεξεργαστή, ελέγχετε η ροή εκτέλεσης της κάθε εντολής δίνοντας τις κατάλληλες τιμές στα σήματα ελέγχου.

### 2.Περιγραφή της Σχεδίασης

Στον επεξεργαστή πολλαπλών κύκλων που πρόκειται να κατασκευάσουμε, κάθε εντολή χρειάζεται διαφορετικό αριθμό κύκλων του ρολογίου για να ολοκληρωθεί. Για αυτό το λόγο σχεδιάστηκαν οι εξής καταστάσεις λειτουργίας:

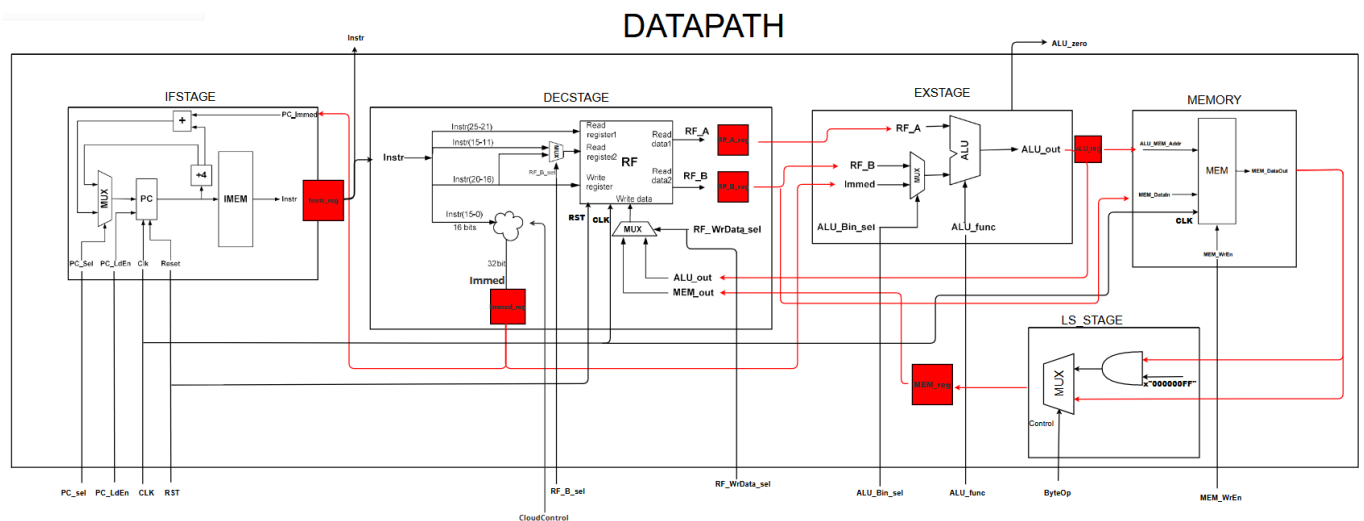
1. **if\_state**:Φορτώνεται η εντολή από την μνήμη IMEM και παρνίεται στον καταχωρητή PC.
2. **dec\_state**:Αποκωδικοποιείται η εντολή που βρίσκεται στον καταχωρητή PC.
3. **exec\_state**:Πραγματοποιείται ο διαχωρισμός των εντολών σε:
  - (α') **R-type/li/lui/addi/andi/ori**: Υπολογισμός πράξης της ALU.
  - (β') **B/beq/bne**: Υπολογισμός της διεύθυνσης της επόμενης εντολής.
  - (γ') **Lw/Lw/Sw**: Υπολογισμός της διεύθυνσης πρόσβασης της μνήμης.
4. **mem\_state**:Έχοντας την διεύθυνση πρόσβασης της μνήμης, γίνεται το load/store αντίστοιχα.
5. **wb\_state**:Αναλόγως την εντολή που εκτελείται γράφεται το κατάλληλο αποτέλεσμα στο αρχείο καταχωρητών.
6. **init\_state**: Πρακτικά κενή κατάσταση για τις εντολές διακλάδωσης ώστε να μην χαλάει η ροή του προγράμματος.

Για να επιτευχθεί αυτός ο διαχωρισμός της εκτέλεσης σε διαφορετικές καταστάσεις, θα πρέπει να εισάγουμε καταχωρητές μετωξύ των σταδίων έτσι ώστε να αποθηκεύονται οι τιμές που χρειάζεται να χρησιμοποιηθούν σε κάποιο επόμενο κύκλο ρολογιού.

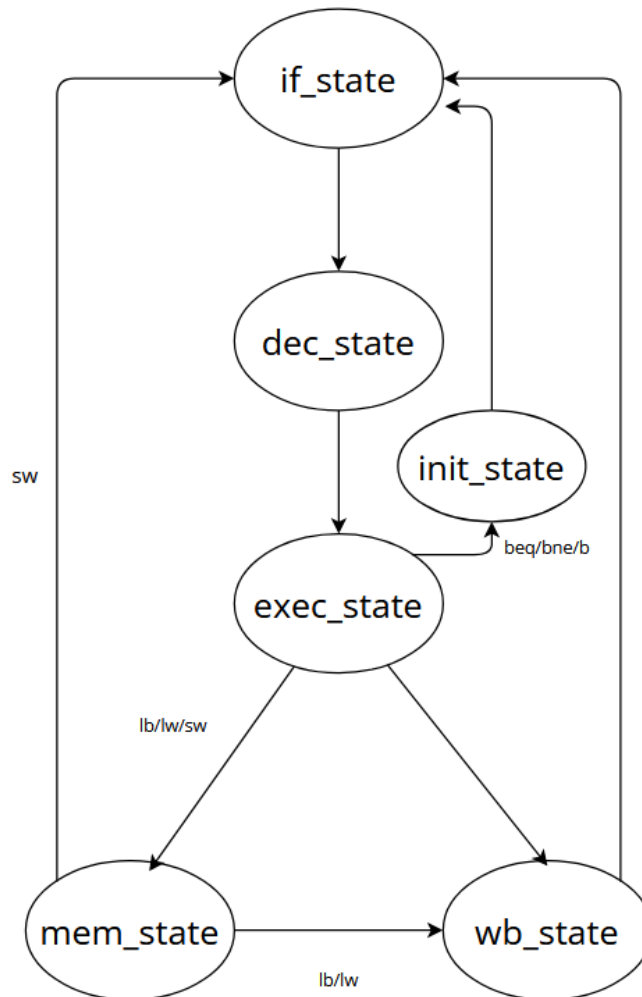
Προστέθηκαν οι εξής καταχωρητές:

- **Instr\_register:** Αποθηκεύει την εντολή.
- **RF\_A/RF\_B\_register:** Αποθηκεύουν τα δεδομένα εξόδου του Register File.
- **Immed\_register:** Αποθηκεύει την τιμή του textitmediate.
- **ALU\_register:** Αποθηκεύει το αποτέλεσμα της ALU.
- **MEM\_register:** Αποθηκεύει τα δεδομένα της μνήμης.

Παρακάτω φαίνεται το DATAPATH του επεξεργαστή πολλαπλών κύκλων (με κόκκινο χρώμα οι καταχωρητές που προστέθηκαν):



Τέλος, η μηχανή πεπερασμένων καταστάσεων είναι υπεύθυνη να κάνει τις μετβάσεις μεταξύ των σταδίων ανάλογα με την εντολή. Η μηχανή αλλάζει σύμφωνα με την θετική ακμή του ρολογιού και παράγει τα απαιτούμενα σήματα ελέγχου. Παρακάτω φαίνεται το State Diagram της FSM:



### 3. Συμπεράσματα

Σε αυτήν την εργαστηριακή άσκηση, κατασκευάσαμε έναν επεξεργαστή πολλαπλών κύκλων. Στον επεξεργαστή μονού κύκλου, η περίοδος του ρολογιού πρέπει να είναι αρκετά μεγάλη ώστε να προλαβαίνουν να ολοκληρωθούν ακόμα και οι πιο χρονοβόρες εντολές. Επομένως περιορίζεται και δεν μπορεί η περίοδος του να μειωθεί κάτω από ένα συγκεκριμένο όριο. Στον επεξεργαστή πολλαπλών κύκλων, με το να χωρίζονται οι εντολές σε στάδια εκτέλεσης, παρατηρούμε ότι μπορεί να μειωθεί σημαντικά η περίοδος του ρολογιού.