ΠΟΛΥΤΕΧΝΕΙΟ
ΚΡΗΤΗΣ

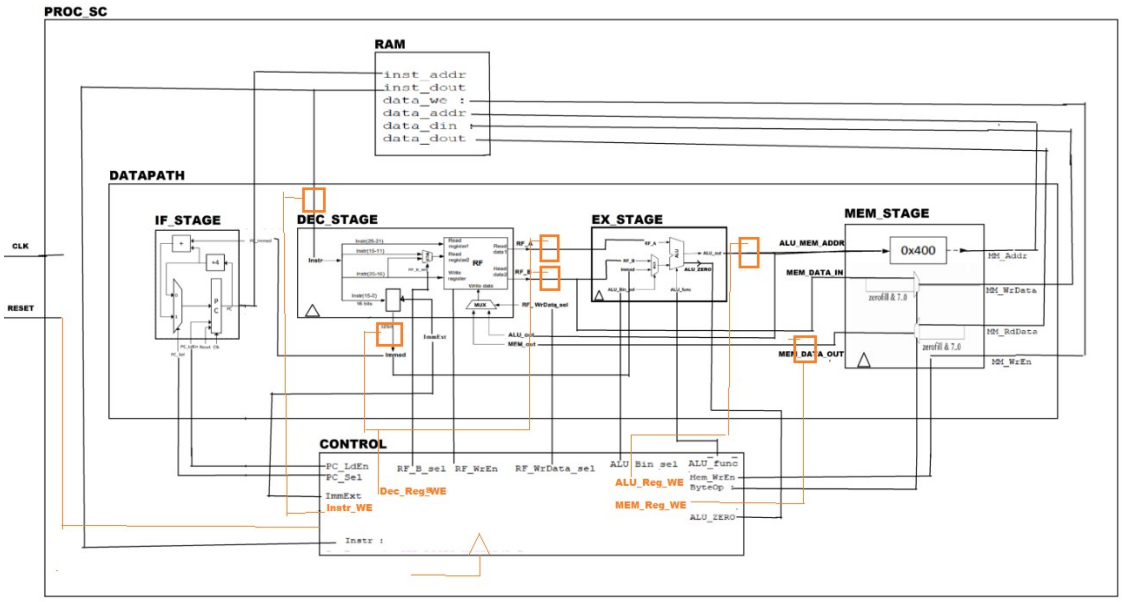
ΟΡΓΑΝΩΣΗ ΥΠΟΛΟΓΙΣΤΩΝ 2022
ΑΝΑΦΟΡΑ ΠΡΩΤΗΣ ΕΡΓΑΣΙΑΣ

Αλέξανδρος Τερζής ΑΜ: **2013030184**

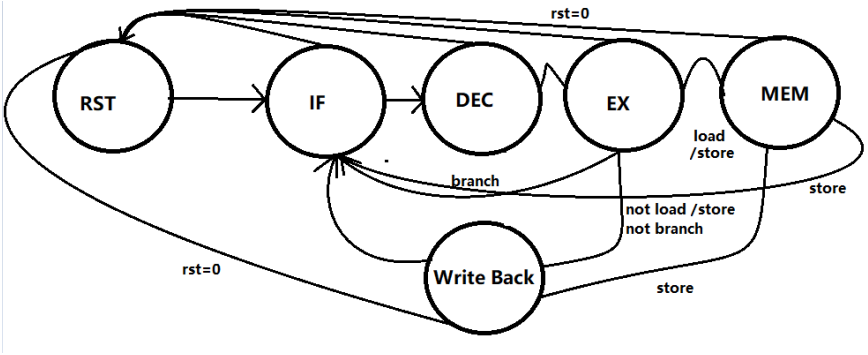
ΤΕΤΑΡΤΗ ΦΑΣΗ

Σε αυτό το κομμάτι της εργασίας κληθήκαμε να μετατρέψουμε τον single cycle επεξεργαστή σε multi cycle. Με αυτόν τον τρόπο ο κύκλος ρολογιού δεν θα πρέπει να είναι τουλάχιστον όσος είναι ο χρόνος που χρειάζεται για να εκτελεστεί η μεγαλύτερη σε χρονική καθυστέρηση εντολή (load). Το ρολόι μπορεί να πάρει μικρότερη τιμή κάνοντας τον επεξεργαστή μας πιο γρήγορο. Εκτελεί δηλαδή την κάθε εντολή για τον χρόνο που χρειάζεται και μετά πάει στην επόμενη (non-pipelined).Αυτό συμβαίνει στην περίπτωση που δεν έχουμε Load, οι εντολές θέλουν έναν κύκλο ρολογιού λιγότερο για να εκτελεστούν, και αυτό σε βάθος χρόνου εκτέλεσης εντολών, κάνει συνολικά τον επεξεργαστή πιο γρήγορο από τον single cycle.

Στο DATAPATH προσθέσαμε ενδιάμεσους καταχωρητές ώστε να μπορούμε να αποθηκεύουμε τις τιμές αυτών για όσους κύκλου ρολογιού χρειάζεται.



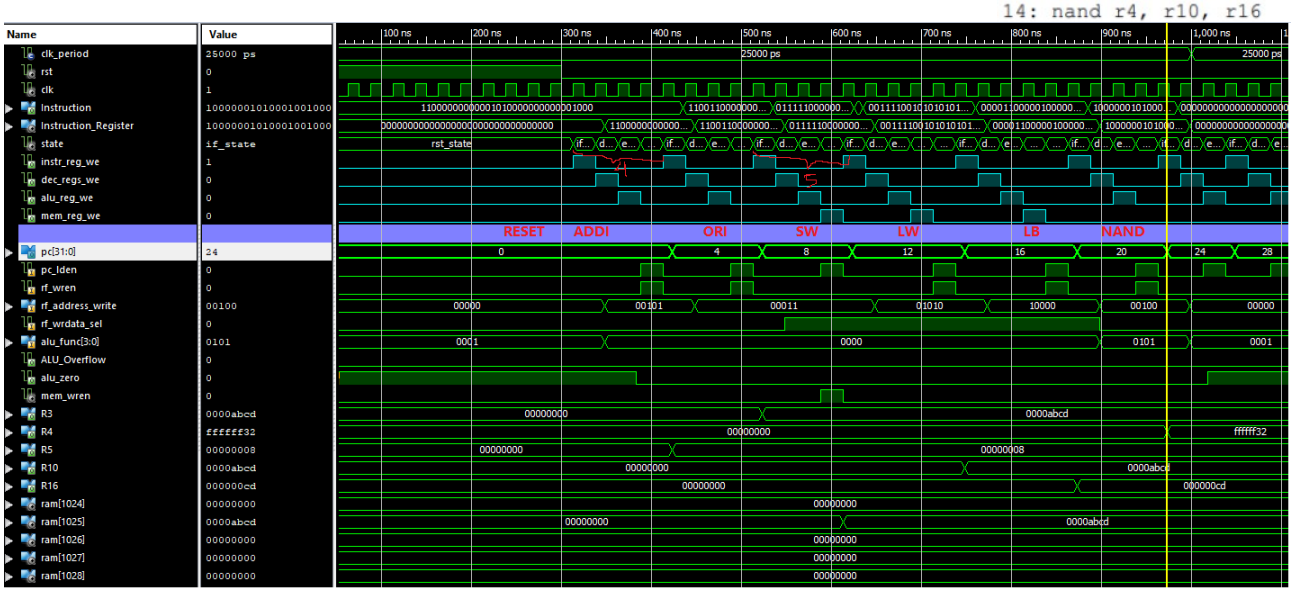
Στο CONTROL, το οποίο από ένας πίνακα αληθείας που ήταν στην προηγούμενη φάση, μετατράπηκε σε μια σύγχρονη μηχανή πεπερασμένων καταστάσεων (FSM) μικτού τύπου Mealy/Moore, καθώς κάποια σήματα εξαρτώνται και από την κατάσταση και από την είσοδο(MEM_Wr_En) ενώ κάποια άλλα μόνο από την κατάσταση(RF_Wr_En). Με αυτόν τον τρόπο καταφέρνουμε να παίρνουμε την κάθε εντολή από το αρχείο .rom και να βγαίνουν τα σωστά σήματα ελέγχου προς το Datapath, να εκτελείται η κάθε εντολή και να πηγαίνουμε στην επόμενη, σε λιγότερο κατά μέσο όρο χρόνο.



Παρακάτω βλέπουμε την προσομοίωση του πρώτου προγράμματος αναφοράς, η διαφορά που παρατηρούμε στον multicycle σε σχέση με τον singlecycle είναι ότι η κάθε εντολή παίρνει ακριβός όσο χρόνο χρειάζεται κάνοντας τον επεξεργαστή πιο γρήγορο.

Πρόγραμμα αναφοράς #1

```
00: addi r5, r0, 8
04: ori  r3, r0, 0xABCD
08: sw   r3, 4(r0)
0C: lw   r10, -4(r5)
10: lb   r16, 4(r0)
14: nand r4, r10, r16
```



ΠΕΜΤΗ ΦΑΣΗ

Σε αυτή την φάση μετατρέψαμε τον επεξεργαστή ώστε να λειτουργεί με pipeline στην εκτέλεση των εντολών. Με αυτόν τον τρόπο δηλαδή, σε κάθε κύκλο ρολογιού φορτώνεται μια καινούρια εντολή στον PC, πλήν εξερέσεων (stalling), με αποτέλεσμα να εκτελούνται εως 5 εντολές παράλληλα, η κάθε μια σε διαφορετική μονάδα του επεξεργαστή (IF,EX,DEC,MEM,WriteBack) με αποτέλεσμα την δραματική μείωση του χρόνου που χρειάζεται για να εκτελεστεί το πρόγραμμα αναφοράς.

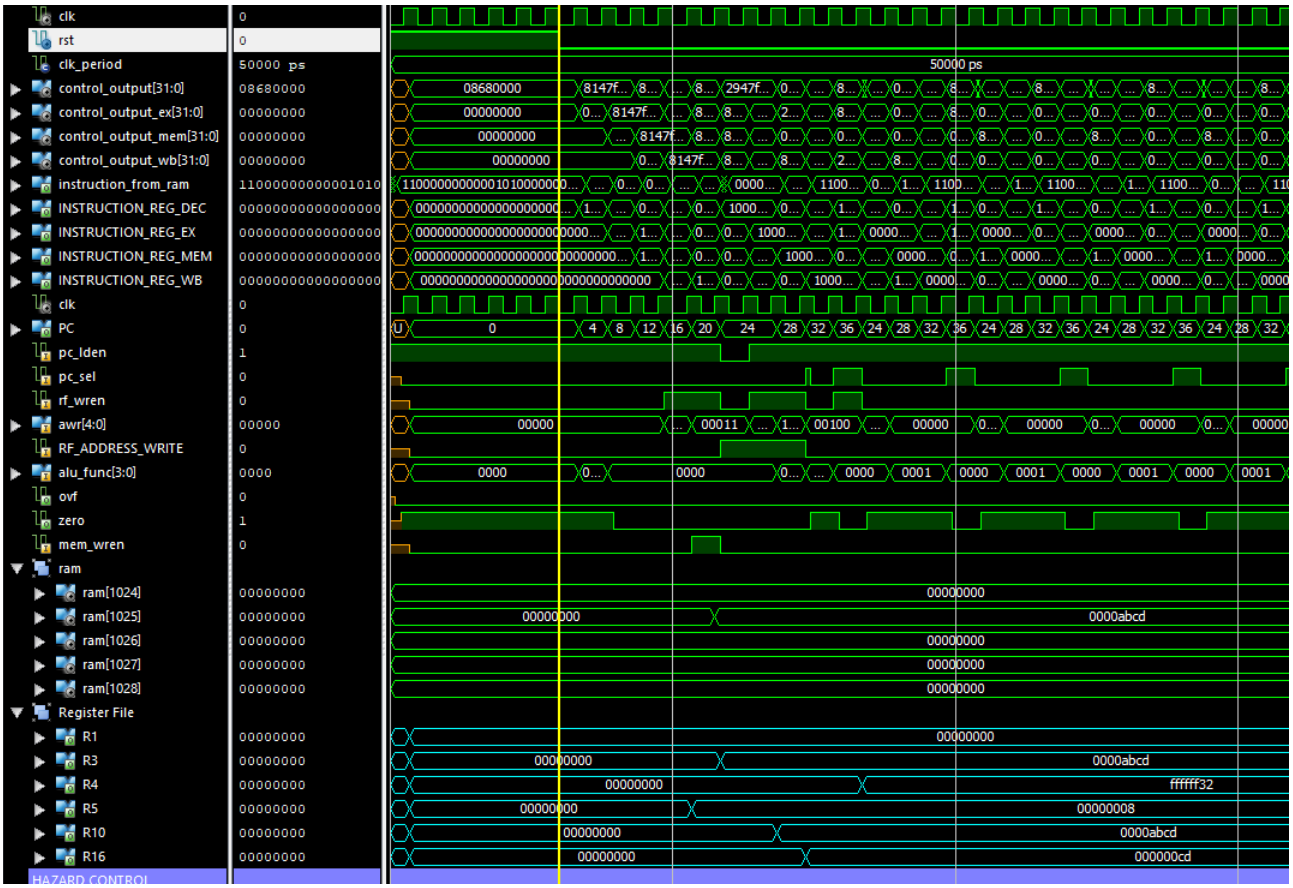
Το στάδιο MEM το χρησιμοποιούν όλες οι εντολές ακόμα και αυτές που δεν χρειάζονται πρόσβαση στην μνήμη έτσι ώστε να έχουν όλες οι εντολές το ίδιο πλήθος κύκλων ρολογιού ώστε να αποφύγουμε εξ αρχής τα Structural Hazards.

Στο **DATAPATH** του Pipeline, προστέθηκαν ενδιάμεσοι καταχωρητές ανάμεσα από την κάθε βαθμίδα εκτέλεσης εντολών, ώστε να γνωρίζουμε σε κάθε βαθμίδα ποιά είναι η εντολή ο PC τα control σήματα και όλα τα data της εντολής(rf_a,rf_b,immed,alu_out,mem_out). Μία ακόμα αλλαγή που έγινε στο Datapath αφορά το σήμα PC_Select το οποίο παίρνει τιμή από λογική στο Datapath κατά το στάδιο εκτέλεσης EX και όχι κατευθείαν από το control όπως συνέβαινε στις προηγούμενες φάσεις. Το Control πλέον ενημερώνει το Datapath κατά το στάδιο DEC μέσω ενός νέου 2-μπιτου σήματος (Branch_Control) για το αν έχουμε, και το τι τύπου Branch έχουμε έτσι ώστε στον επόμενο κύκλο(ex_stage) να παρθεί η απόφαση για το αν θα πετύχει το branch. Το ίδιο το σήμα PC_Select καθορίζει το αν θα γίνει Flush.

Εντός του Datapath ενσωματώσαμε ενα καινούριο module **HAZARD_CONTROL** το οποίο ανιχνεύει τα data hazards. Συγκεκριμένα ελέγχει αν είναι ίδιοι οι καταχωρητές προέλευσης που βρίσκονται στην βαθμίδα εκτέλεσης Ex, με τον καταχωρητή προορισμού σταδίου MEM και τον καταχωρητή προορισμού σταδίου WrBack. Με αυτόν τον τρόπο βλέπουμε αν υπάρχει εξάρτηση και καθορίζει τα select των πολυπλεκτών του forwarding. Επιπλέον ελέγχει αν είναι ίδιοι οι καταχωρητές προέλευσης που βρίσκονται στην βαθμίδα εκτέλεσης DEC, με τον καταχωρητή προορισμού σταδίου EX, στην περίπτωση που η βαθμίδα EX έχει load, δίνει εντολή να γίνει stall. Σημειώνω οτι το PC_Ld_En εξαρτάται πλέον μόνο από το stall και δεν υπολογίζεται από το control.

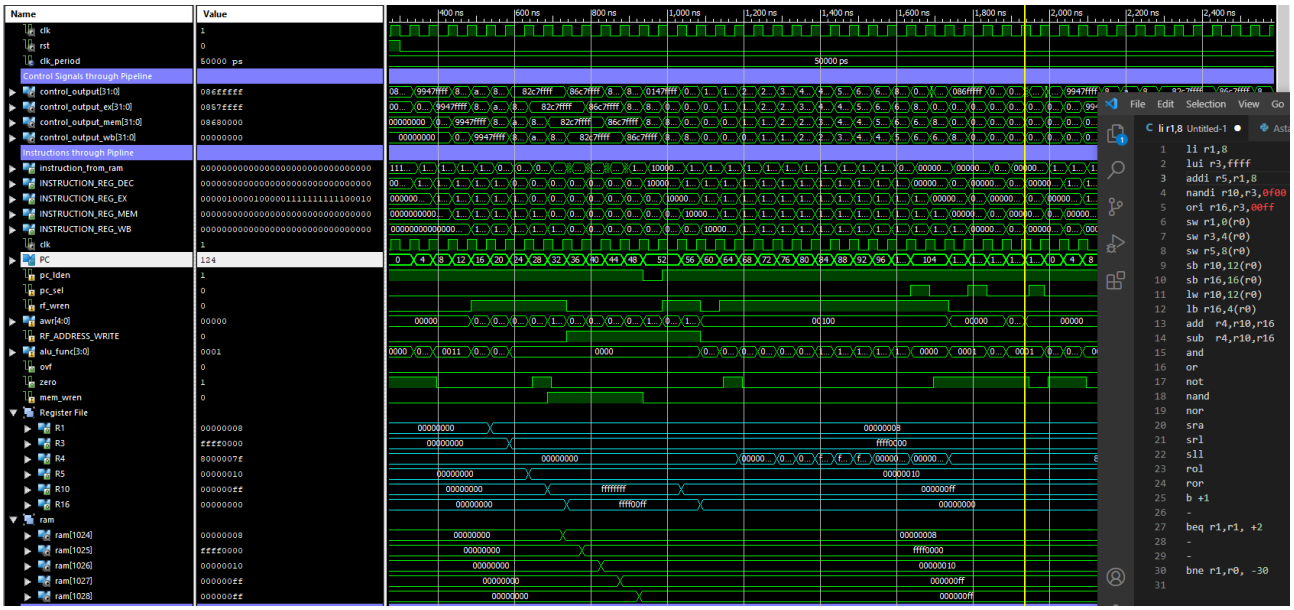
Για το **CONTROL_Pipeline** χρησιμοποιήσαμε αυτό του Single Cycle κάνοντας την μετατροπή για τις διακλαδώσεις που αναφέρεται παραπάνω. Οι εντολές αποκωδικοποιούνται κατά το στάδιο DEC και τα Control σήματα μεταφέρονται μέσω των καταχωρητών pipeline στην εκάστοτε βαθμίδα που τα χρησιμοποιεί.

Παρακάτω βλέπουμε τα προγράμματα αναφοράς 1 και 2



Παρακάτω βλέπουμε τα προγράμματα αναφοράς που ζητήθηκε.

Παρακάτω βλέπουμε τα προγράμματα αναφοράς όλων των εντολών.



ΒΙΒΛΙΟΓΡΑΦΙΑ:

-e-class.gr Υλικό Εργαστηρίου/Διαλέξεις

-xilinx.com

-stackoverflow.com