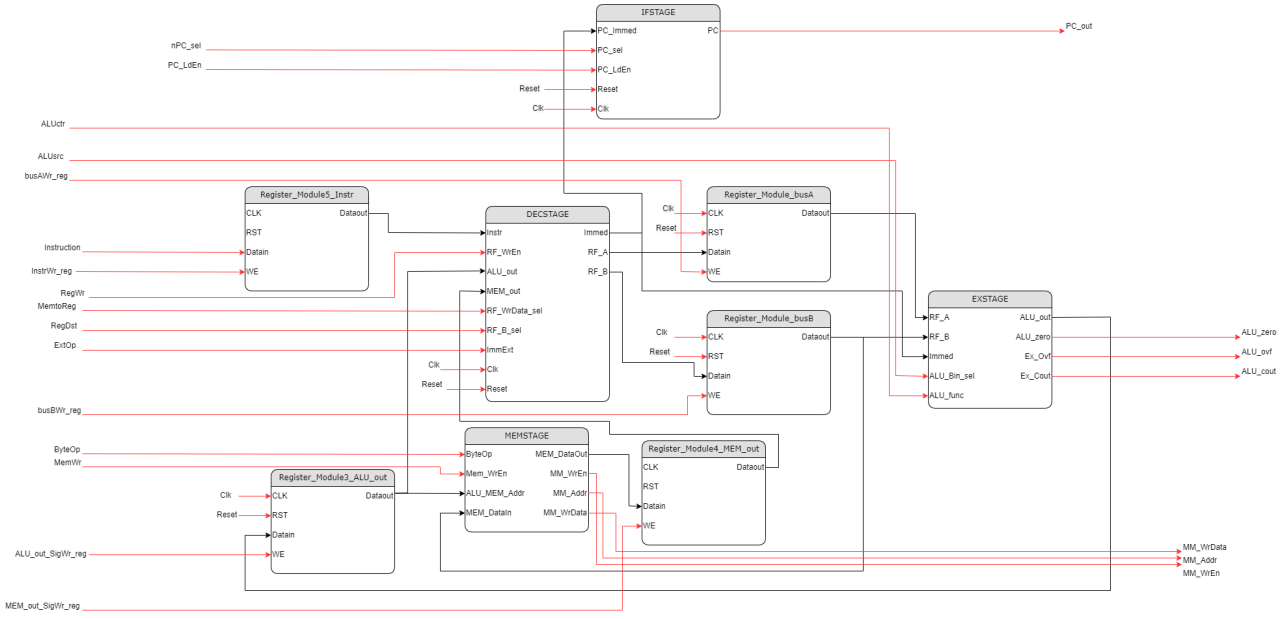
ΑΝΑΦΟΡΑ ΕΡΓΑΣΙΑΣ 2: ΣΧΕΔΙΑΣΗ ΕΠΕΡΓΑΣΤΗ ΠΟΛΛΑΠΛΩΝ ΚΥΚΛΩΝ ΚΑΙ ΜΕΤΑΤΡΟΠΗ ΤΟΥ ΣΕ PIPELINE

ΦΑΣΕΙΣ 1-2-3

Κατά την υλοποίηση των φάσεων 1,2,3 δούλεψα σε 2 υπολογιστές κατά την αντιγραφή των modules στο decstage πάτησα αντί για add Copy την επιλογή add Copy of Source η οποία σώζει τις διαφορές στο ίδιο το ise έτσι το .vhd αρχείο του module δέν ενημερώνεται και παραμένει η αρχική έκδοση. Στην αποστολή του project κατά λάθος μου έστειλα αυτό το παλιο DEC STAGE η οποία σωστή έκδοση του μαζί με screenshots όπου επισημαίνονται οι αλλαγές που έχουν οι 2 εκδόσεις του module παραθέτονται σε αυτήν την εργασία στον φάκελο extra. Σημειώνεται επίσης οτι το module της ram δέν υπάρχει στον φάκελο proc_sc αν και υπάρχει σε άλλους φακέλους (mem).

ΦΑΣΗ 4

Στην Τέταρτη φάση ζητήθηκε να μετατρέψουμε τον επεξεργαστή ενός κύκλου σε επεξεργαστή πολλών κύκλων. Αυτό έγινε προσθέτοντας 5 Registers που λειτουργούσαν ως flip-flops δηλαδή αργούσαν το σήμα για ένα κύκλο ρολογιού έκαστος τα σήματα που έμπαιναν σε αυτούς τους registers ήταν τα: Instruction, MEM_DataOut, ALU_out, RF_A, RF_B. Αυτοι οι registers προστέθηκαν στο datapath του επεξεργαστή μονού κύκλου όπως φαίνεται στο παρακάτω σχήμα.



DATAPATH επεξεργαστή πολλών κύκλων

Στο παραπάνω σχήμα φαίνεται το Datapath με τα εσωτερικά signals να σημειώνονται με μαύρο ενώ τα inputs/outputs με κόκκινο επίσης τα clk και reset συνδέονται όλα σε 2 κοινά σήματα άλλα δεν σημειώθηκαν για χάρη ευκολότερης αναγνωσιμότητας.

Για τον σχεδιασμό του control τα states σχεδιάστηκαν ανάλογα με τους registers που υπήρχαν στο datapath με κάθε state να χειρίζεται σήματα που χρησιμοποιούνται σε component του ίδιου η του επομένου state ενώ χειρίζεται τα write enable των register στα επόμενα state και προηγούμενα state. Σχετικά με τις τιμές παίρνουν τα σήματα για κάθε εντολή πάρθηκαν από το control του επεξεργαστή ενός κύκλου. Ο διαμοιρασμός στα states έχει ως εξής:

1. Reset: Ολα τα σήματα αρχικοποιούνται κατά το reset state

2. Fetch: Γίνεται fetched μια εντολή από την ram αφού βγαίνει τιμή από τον pc register και τον alu register έτσι λαμβάνεται η διεύθυνση εντολής και από την ram βγαίνει η ίδια η εντολή, ενεργοποιείται ακόμη το wren του instruction register.

3. Decode: Η εντολή μπαίνει στον instruction register και καθορίζεται το επόμενο state ανάλογα με το opcode της εντολής

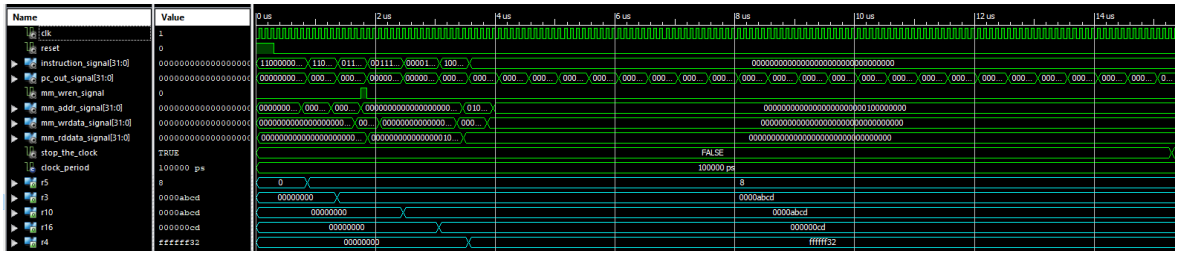
4. Ex: Αρχίζει η εκτέλεση της εντολής από το rf λαμβάνονται οι registers που θα χρειαστούν για την εκτέλεση της απο το rf ενώ ενεργοποιείται ακόμη το wren του RF_A και RF_B register.

5. Ex2 : Γίνεται η πράξη στην alu ανάλογα με την εντολή, ενεργοποιείται το wren του alu register.

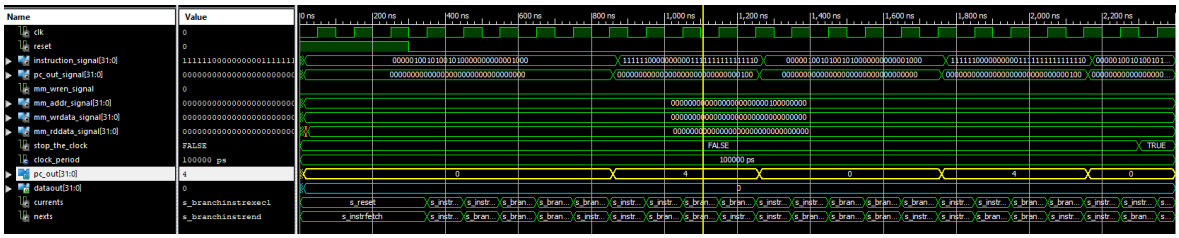
6. End : Υπολογίζεται το επόμενο PC. Το αποτέλεσμα της alu μπαίνει στον alu register και στο επόμενο state γράφεται στο RF για όσες εντολές ζητείται αυτό για εντολές store εφόσον η ram έχει σύγχρονο store γίνονται stored στην αρχή του fetch.

7. Ex3: Υπάρχει μόνο για τις εντολές τύπου load κατά το οποίο ενεργοποιείται ο memory register και απο το επόμενο state (end) λαμβάνεται η τιμή του στην είσοδο του RF (χωρίς να γράφεται) όπου μετά στο fetch θα γραφεί.

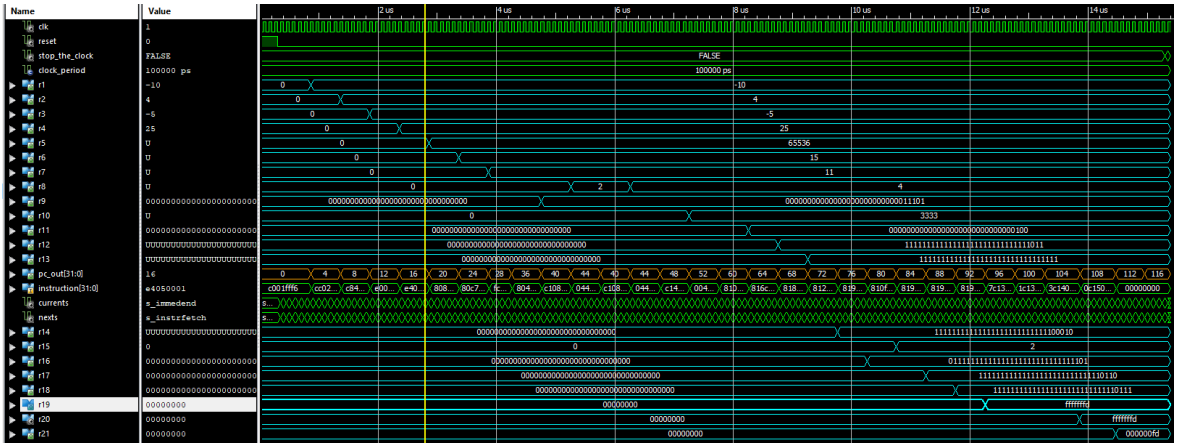
Testbenches



Testbench Προγραμματος αναφοράς 1(romprog1)

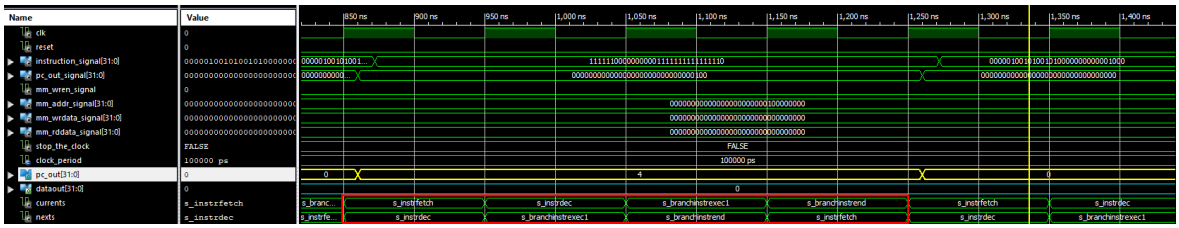


Testbench Προγραμματος αναφοράς 2(romprog2



Testbench Προγραμματος αναφοράς 3 (rom10)

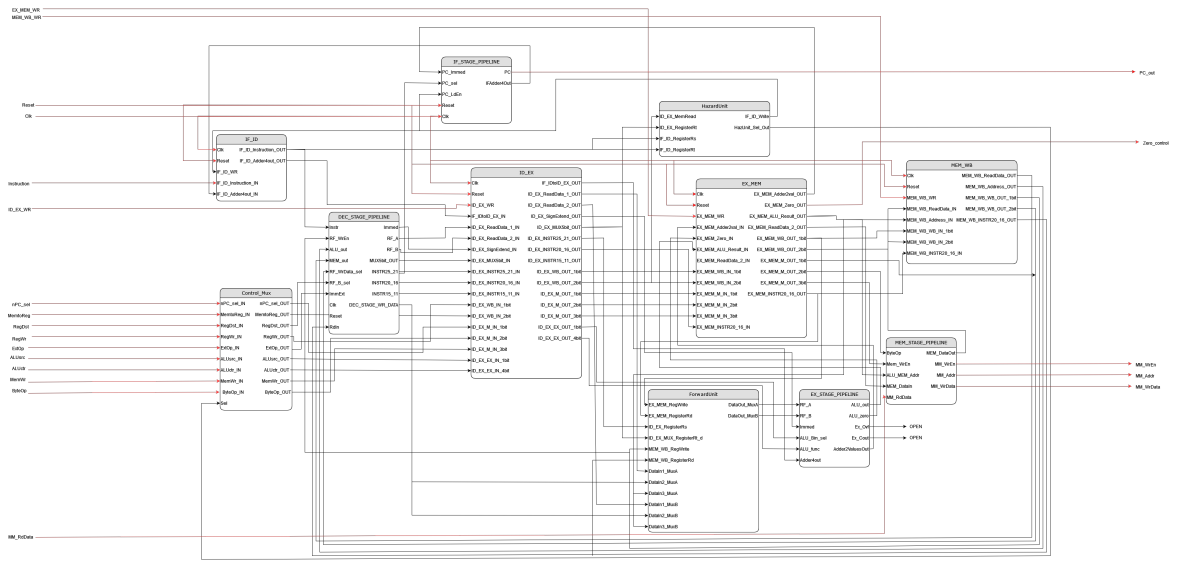
Γίνετε reset για 3 κύκλους του clock και οι τιμές είναι οι αναμενόμενες στα προγράμματα 1 και 3 και στο 2 το πρόγραμμα μπαίνει σε ατέρμονο loop όπως φαίνεται από την τιμή του PC. Σημειώνεται ότι στο σχήμα προγράμματος αναφοράς 1 και 3 και δεν φαίνονται τα states του multicycle λόγω μεγέθους ενώ στο σχήμα προγράμματος 2 δεν είναι ευδιάκριτα τα ex states. Παρακάτω παραθέτεται ένα μεγεθυσμένο σχήμα προγράμματος 2 κατά το οποίο βλέπουμε τα states που ακολουθεί η εντολή b



Εντολή b Προγραμματος αναφοράς 2

ΦΑΣΗ 5

Στην πέμπτη φάση ζητήθηκε να μετατραπεί ο επεξεργαστής σε επεξεργαστή PIPELINE κατά τον οποίο θα εκτελούταν διαφορετικές εντολές σε κάθε state για οικονομία χρόνου. Για την μετατροπή εισάχθηκαν στο datapath 4 νέα modules IF_ID, ID_EX, ED_MEM, MEM_WB τα οποία περιέχουν μέσα τους διάφορους registers που λειτουργούν ως flip flops καθυστερώντας το σήμα που δέχονται κατά ένα κύκλο ρολογιού. Επίσης έγινε αλλαγή στο IF stage καθώς αφαιρέθηκε ο adder 2 τιμών και προστέθηκε στο Ex stage επίσης σε ορισμένα stages εσωτερικά σήματα τέθηκαν ως και εξωτερικά. Τέλος έγινε η σχεδίαση 2 νέων modules του HazardUnit και του ForwardUnit και συνδέθηκαν και αυτά στο datapath το νέο datapath παρουσιάζεται πλήρες στο παρακάτω σχήμα. Το control παρέμεινε γενικά ίδιο με την μόνη διαφορά να είναι ότι αφαιρέθηκαν οι τιμές των σημάτων που επηρεαζόταν από τα branch instructionst και προστέθηκαν τα σήματα wren των register module του datapath, το proc επίσης παρέμεινε ίδιο.



Datapath pipeline επεξεργαστή

Τα νέα modules σχεδιάστηκαν για να αντιμετωπιστούν προβλήματα data Hazard κατά τα οποία μια εντολή αλλάζει την τιμή ενός register ο οποίος χρησιμοποιείται ως Rs στις

αμέσως επόμενες εντολές με αποτέλεσμα οι παρακάτω εντολές να χρησιμοποιούν παλία τιμή του register Rs. Για την αντιμετώπιση αυτών των προβλημάτων γίνεται χρήση Forwarding και Stalling.

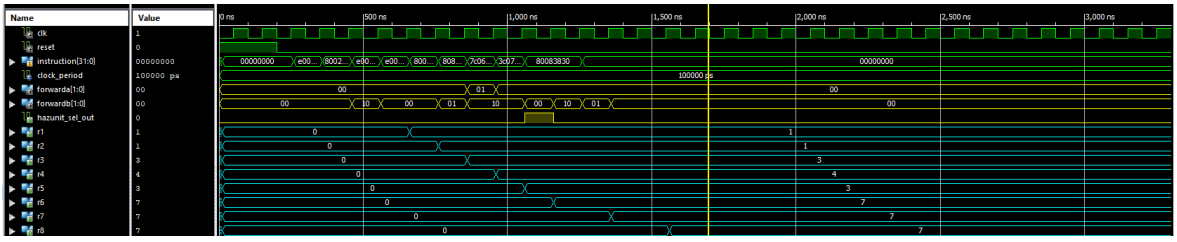
Το Forwarding εκτελείται από το ForwardUnit το οποίο είναι ικανό να κάνει forward δεδομένα καταχωρητή έως και 2 εντολές παρακάτω. Μέσα του γίνεται έλεγχος αν ο καταχωρητής Rd που βγαίνει χρησιμοποιείται σε εντολή που μόλις πέρασε το EX_MEM ισούται με τον καταχωρητή RS, RD, RT που χρησιμοποιείται σε εντολή που μόλις πέρασε το ID_EX δηλαδή σε αμέσως επόμενη εντολή. Εάν αυτή η συνθήκη ισχύει τότε ως είσοδος της επόμενης εντολής στο EXSTAGE (RFA H RFB) θέτετε το ALU OUT από την παλιά εντολή μέσω των Mux ανάλογα με το ποιοι καταχωρητές είναι ίσοι.

Εάν δεν ισχύει αυτή η περίπτωση τότε γίνεται ο ίδιος έλεγχος άλλα για τους καταχωρητές που χρησιμοποιούνται από εντολή που μόλις πέρασε το MEM_WB και όχι το EX_MEM δηλαδή αυτή η περίπτωση αναφέρεται σε εντολές που είναι 2 states μακριά και εάν οι συνθήκες επαληθεύονται τότε ως είσοδο στο EXSTAGE (RFA H RFB) στην νεότερη εντολή θέτεται τα δεδομένα που είναι προς εγγραφή στο RF του DECSTAGE απο την παλαιότερη εντολή.

Σε περίπτωση που δεν ισχύει καμία από τις παραπάνω περιπτώσεις τότε δεν πραγματοποιείται forwarding

Το stalling εκτελείται από το HazardUnit όταν η στην προηγούμενη εντολή διαβάζονται δεδομένα από την μνήμη γίνεται δηλαδή load σε έναν register και η τωρινή εντολή χρησιμοποιεί αυτόν τον register (είναι καταχωρητής Rt/Rs). Δεν μπορούμε να κάνουμε forwarding καθώς τα δεδομένα χρειάζονται ακόμη έναν κύκλο για να φορτωθούν από την μνήμη έτσι για αυτόν τον κύκλο χρήσιμοποιήται stall απενεργοποιείται δηλαδή ο PC register, ο IF/ID και δεν φορτώνονται τιμές από το control έτσι δεν λαμβάνεται επόμενη εντολή για έναν κύκλο. Στον επόμενο κύκλο που θα έχει φορτωθεί η τιμή του register από την μνήμη καλείται forward ώστε η τωρινή εντολή να λάβει τις σωστές τιμές. Στο forward θέτεται η τιμή εισόδου του exstage της νέας εντολής ως η τιμή απο προς εγγραφή στο decstage.

Testbenches



PIPELINE testbench

Το πρόγραμμα που χρειάστηκε για testbench στο pipeline μπορεί να βρεθεί στον φάκελο

1. Η πρώτη εντολή li r1, 1 φορτώνει το 1 στον καταχωρητή 1

2. Στην δεύτερη εντολή add r2, r0, r1 ο καταχωρητής r1 δεν έχει προλάβει να πάρει την τελική του τιμή άρα για να δουλέψει σωστά υπάρχει forwarding και λαμβάναι απο το alu out η τιμή του r1 και γίνεται είσοδος στο exstage της δεύτερης εντολής

3. Η τρίτη εντολή είναι η li r3, 3

4. Η τέταρτη εντολή είναι η li r4, 4

5. Η πέμπτη εντολή add r5, r0, r3 θα χρειαστεί πάλι forwarding και η τιμή του r3 θα ληφθεί απο το decstage ώστε να πάει στο exstage της τωρινής εντολής.

6. Η έκτη εντολή add r6, r4, r5 χρειάζεται forwarding για 2 registers r4 και r5 έτσι λαμβάνεται ο r4 απο το decstage και ο r5 απο το alu out και θέτονται ως RFA και RFB στο exstage της τωρινής εντολής.

7. Η έβδομη εντολή sw r6, 4(r0) χρειάζεται forwarding για το r6 και το παίρνει από το alu out

8. Η ογδόη εντολή είναι η lw r7, 4(r0)

9. Η ένατη εντολή add r8, r0, r7 παρουσιάζει stall καθώς η προηγούμενη εντολή ήταν load και χρησιμοποιούσε τον r7 ως Rd τον οποίο χρησιμοποιεί και η τωρινή. Μετά τον stall για να πάρει την σωστή τιμή ο r7 χρειάζεται forwarding