

Αναφορά Εργαστηρίου 4

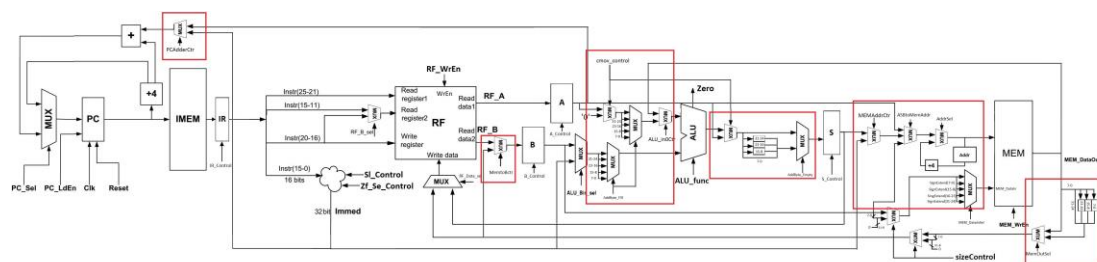
LAB31231454

Μανώλης Πετράκος

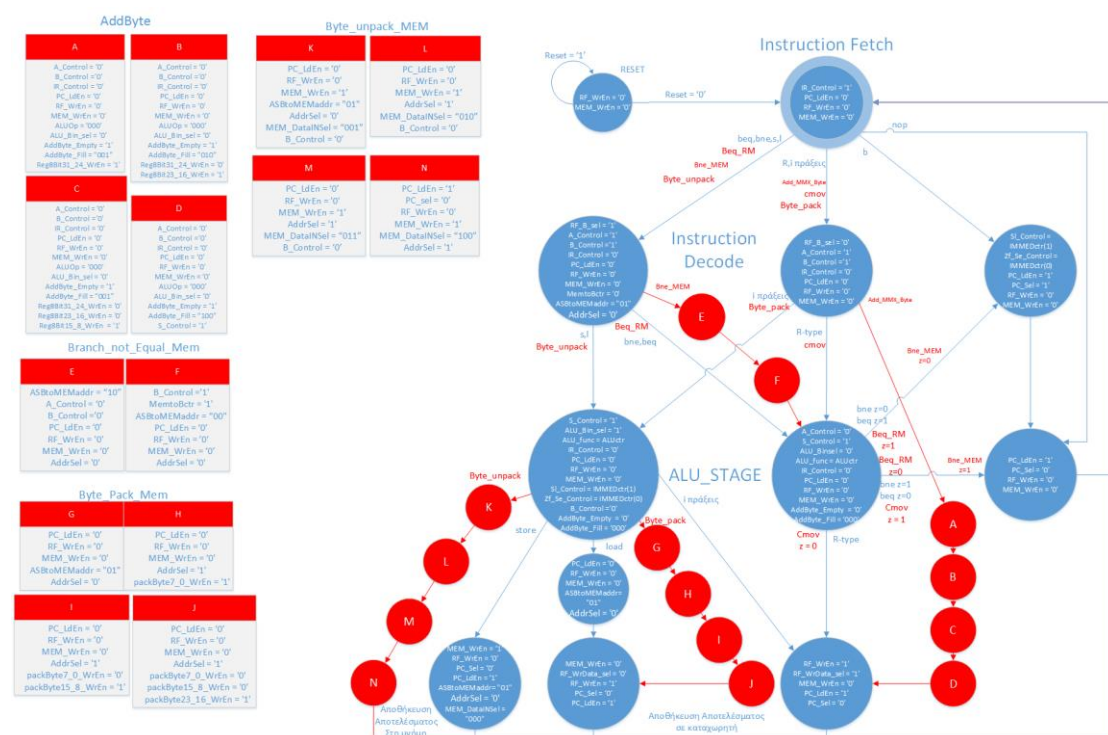
Μιχαήλ Δακανάλης

Προεργασία

Datapath:



FSM:



Ότι καινούργιο είναι με κόκκινο χρώμα. Επίσης στην fsm οι καινούργιες καταστάσεις έχουν σε υπόμνημα τα σήματα τους.

Περιγραφή Άσκησης

Προσθέσαμε ό,τι λογική χρειαζόταν (πολυπλέκτες, καταχωρητές κ.λπ.) στο datapath από το προηγούμενο εργαστήριο και προσθέσαμε ότι παραπάνω καταστάσεις και ελέγχους χρειαζόνταν στην fsm.

Cmovn: Μετά το Instr_Fetch κάνει decode όπως οι R εντολές. Στο ALU_Stage κάνει μία αφαίρεση του rt με το 0 χρησιμοποιώντας ένα πολυπλέκτη για να αντικαταστήσει το A και αν το αποτέλεσμα είναι διαφορετικό του μηδέν βάζει τον rs στον S από τον A και μετά τον αποθηκεύει στο αρχείο καταχωρητών. Αλλιώς απλά πάει στην επόμενη εντολή.

Add_MMX_byte: Μετά το Instr_Fetch κάνει decode όπως οι R εντολές. Έπειτα σπάει τα περιεχόμενα κάθε καταχωρητή (A,B) σε 4 bytes χρησιμοποιώντας δύο πολυπλέκτες. Στους επόμενους τρεις κύκλους κάνει πράξεις με τα αντίστοιχα bytes και τα αποθηκεύει σε καταχωρητές ενός byte. Στον τέταρτο κάνει τη τελευταία πράξη και κατευθείαν ενώνει τα αποτελέσματα και τα αποθηκεύει στον S. Τέλος αποθηκεύει το αποτέλεσμα στον RF.

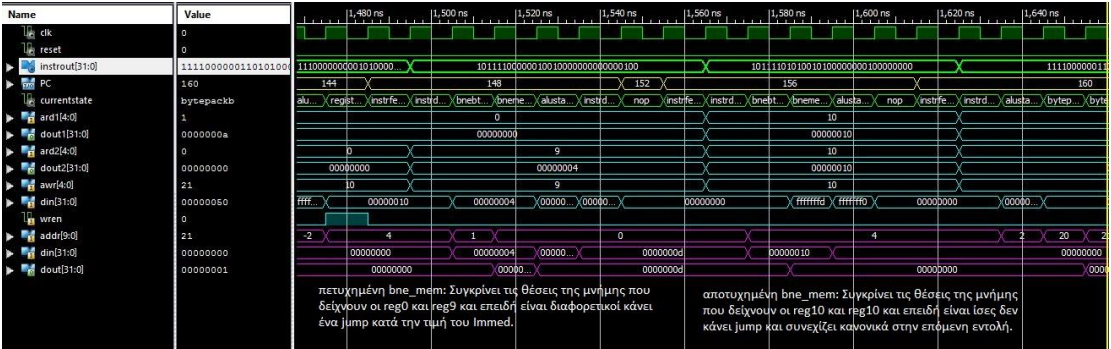
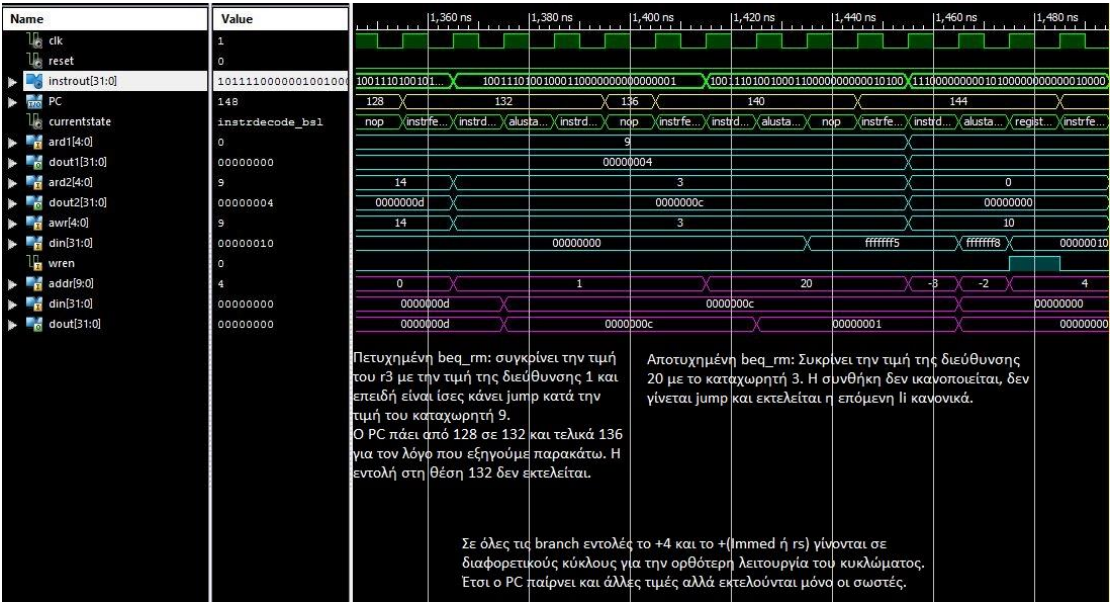
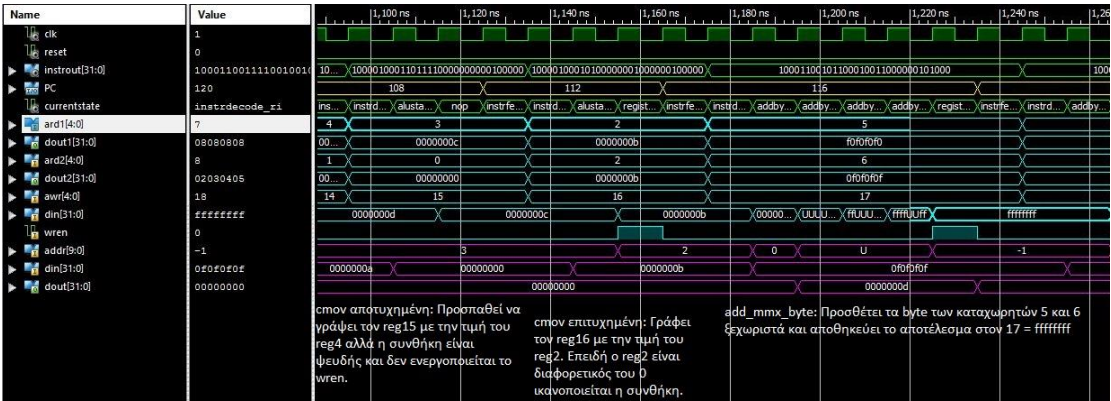
branch_equal_reg_mem: Μετά το Instr_Fetch κάνει decode και το ALU_Stage όπως οι branch εντολές εκτός του ότι στην θέση του rs χρησιμοποιεί μια τιμή από την μνήμη, επιλεγμένη από τον Immediate με τη χρήση ενός πολυπλέκτη στην διεύθυνση της μνήμης και ενός δεύτερου στην πρώτη είσοδο της ALU. Τέλος αν το αποτέλεσμα της πράξης είναι 0 θα προσθέσει στον PC την τιμή του rs, που είναι στον A.

branch_not_equal_mem: Μετά το Instr_Fetch συνεχίζει στο decode όπως οι branch εντολές. Έπειτα μέσω ενός πολυπλέκτη παίρνει τις τιμές των rs και rt από τους καταχωρητές A και B και με αυτές παίρνουμε από την μνήμη δύο τιμές και τις συγκρίνουμε. Επειδή αυτή η διαδικασία χρειάζεται δύο κύκλους, κατά τον πρώτο αποθηκεύουμε στον καταχωρητή B το αποτέλεσμα του MEM[rt] και στον δεύτερο που φτάνει το MEM[rs] γίνεται η πράξη κατευθείαν. Αν το αποτέλεσμα είναι διάφορο του μηδέν προσθέτουμε στον PC τον Immed της εντολής. Αλλιώς συνεχίζεται η εκτέλεση του προγράμματος σειριακά.

Byte_pack_mem: Μετά το Instr_Fetch κάνει decode και το ALU_Stage όπως οι load εντολές για να δημιουργήσει το base_addr. Έπειτα βάζει το αποτέλεσμα σε έναν καταχωρητή, κατευθείαν βγάζει από την μνήμη τη πρώτη τιμή και αποθηκεύει το πρώτο byte της σε έναν καταχωρητή. Μετά επαναλαμβάνει την διαδικασία αυτή αυξάνοντας την διεύθυνση κατά 4 μέχρι να βγάλει δύο ακόμα αποτελέσματα και να αποθηκεύσει τα πρώτα τους byte σε έναν καταχωρητή το καθένα. Τέλος βγάζει την τελευταία τιμή και ενώνει το τελευταίο της byte με τις τιμές που έχουμε αποθηκεύσει και βάζουμε το αποτέλεσμα στην μονάδα καταχωρητών.

Byte_unpack_mem: Μετά το Instr_Fetch κάνει decode και το ALU_Stage όπως οι i-type εντολές για να δημιουργήσει το base_addr. Χρησιμοποιεί την ίδια διαδικασία με τη προηγούμενη εντολή για να ελέγξει την διεύθυνση της μνήμης. Σε κάθε κύκλο αποθηκεύει το κάθε byte του rd sign extended στην αντίστοιχη θέση μνήμης.

Κυματομορφές





Συμπεράσματα

Μάθαμε πως να επεκτείνουμε της εντολές ενός επεξεργαστή πολλαπλών κύκλων προσθέτοντας επιπλέον λογική στην σχεδιάσή μας.