ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΣΧΟΛΗ Η.Μ.Μ.Υ

ΜΑΘΗΜΑ: ΗΡΥ 302 – ΟΡΓΑΝΩΣΗ ΥΠΟΛΟΓΙΣΤΩΝ

ΕΡΓΑΣΤΗΡΙΟ ΟΡΓΑΝΩΣΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

Αναφορά 1ου Εργαστηρίου :

Όνομα: Νικόλαος Παπουτσάκης

ΑΜ: 2019030206

Εισαγωγή:

Στο συγκεκριμένο project στόχος μας ήταν η δημιουργία ενός single cycle processor βασισμένος πάνω στην αρχιτεκτονική εντολών CHARIS (CHAnia Risc Instruction Set). Σε 1η φάση, κληθήκαμε να δημιουργήσουμε την αριθμητική και λογική μονάδα ALU, αλλά και το αρχείο καταχωρητών. Έπειτα σε δεύτερη φάση σχεδιάσαμε τέσσερις βαθμίδες οι οποίες ήταν χρήσιμες για την υλοποίηση του Datapath. Αρχικά ξεκινήσαμε με τη βαθμίδα ανάκλησης εντολών (instruction fetch), στη συνέχεια δημιουργήσαμε τη βαθμίδα αποκωδικοποίησης εντολών (decode stage), τη βαθμίδα εκτέλεσης εντολών (execute stage) και τέλος τη βαθμίδα πρόσβασης μνήμης (memory stage). Τέλος, με βάση τη δεύτερη φάση δημιουργήσαμε το Datapath και το Control του επεξεργαστή και κάναμε τις απαραίτητες συνδέσεις προκειμένου να φτάσουμε στο τελικό στάδιο.

1η Φάση:

ALU:

Σε πρώτη φάση όπως προαναφέραμε, δημιουργήσαμε τη μονάδα ALU. Η μονάδα αυτή είναι υπεύθυνη για την εκτέλεση λογικών και αριθμητικών πράξεων και η χρήση της είναι ιδιαίτερα σημαντική καθώς για την εκτέλεση οποιαδήποτε εντολής είναι χρήσιμο το αποτέλεσμα της. Μερικές από τις πράξεις που μπορεί να υλοποιήσει είναι οι add, sub, and, or, not, nand, nor, sra, srl, sll, rol και ror. Η συγκεκριμένη μονάδα υλοποιήθηκε ως ένα module .vhd.

Register File:

Το αρχείο καταχωρητών περιέχει 32 καταχωρητές (Register32Bit.vhd), έναν αποκωδικοποιητή (Decoder5to32.vhd) και δύο πολυπλέκτες (MUX32to1.vhd). Είναι ιδιαίτερα χρήσιμο module καθώς εκεί αποθηκεύονται όλες οι τιμές που μας ενδιαφέρουν και συμβάλλουν στη σωστή λειτουργία του επεξεργαστή. Το συγκεκριμένο module έχει τη δυνατότητα εγγραφής και ανάγνωσης προς και από καταχωρητές και διαθέτει σήμα WrEnable. Για την υλοποίηση του πολυπλέκτη χρειάστηκε να δηλώσουμε μία μεταβλητή τύπου array 32 θέσεων προκειμένου να μπορούμε να έχουμε στην είσοδο του κάθε πολυπλέκτη έναν 32bit αριθμό. Έτσι με τη χρήση ενός package μπορέσαμε να υλοποιήσουμε σωστά το συγκεκριμένο κύκλωμα.

Για τη δημιουργία του array αλλά και για τη δήλωση του package ήταν ιδιαίτερα χρήσιμες οι παρακάτω διευθύνσεις:

- https://www.nandland.com/vhdl/examples/example-array-type-vhdl.html
- https://www.nandland.com/vhdl/examples/example-package.html

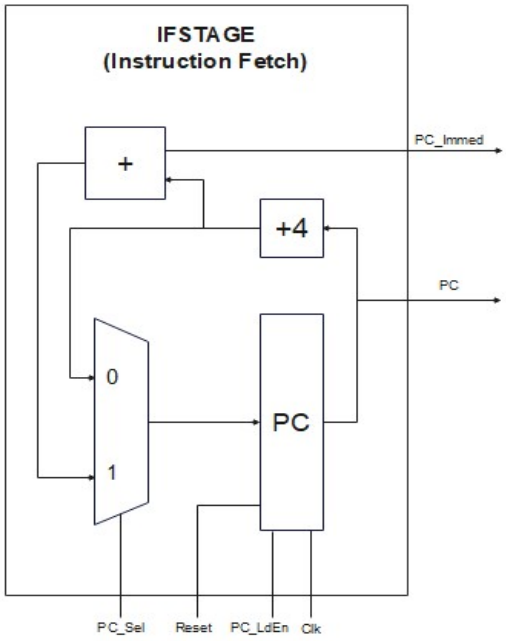
Τέλος, γνωρίζουμε ότι στον mips ο καταχωρητής R0 έχει πάντα την τιμή 0. Επομένως, για να θέσουμε την τιμή 0 στον καταχωρητή R0, βάλαμε στο σήμα Reset την τιμή '1' και στο σήμα WE την τιμή '0'. Έτσι, τίποτα δεν θα μπορέσει να γραφτεί στον καταχωρητή ενώ παράλληλα η έξοδος να παραμένει πάντα μηδέν.

$2^{η}$ Φάση:

Στη $2^{η}$ φάση δημιουργήσαμε τις 4 βαθμίδες του Datapath, ξεκινώντας με τη βαθμίδα ανάκλησης εντολών.

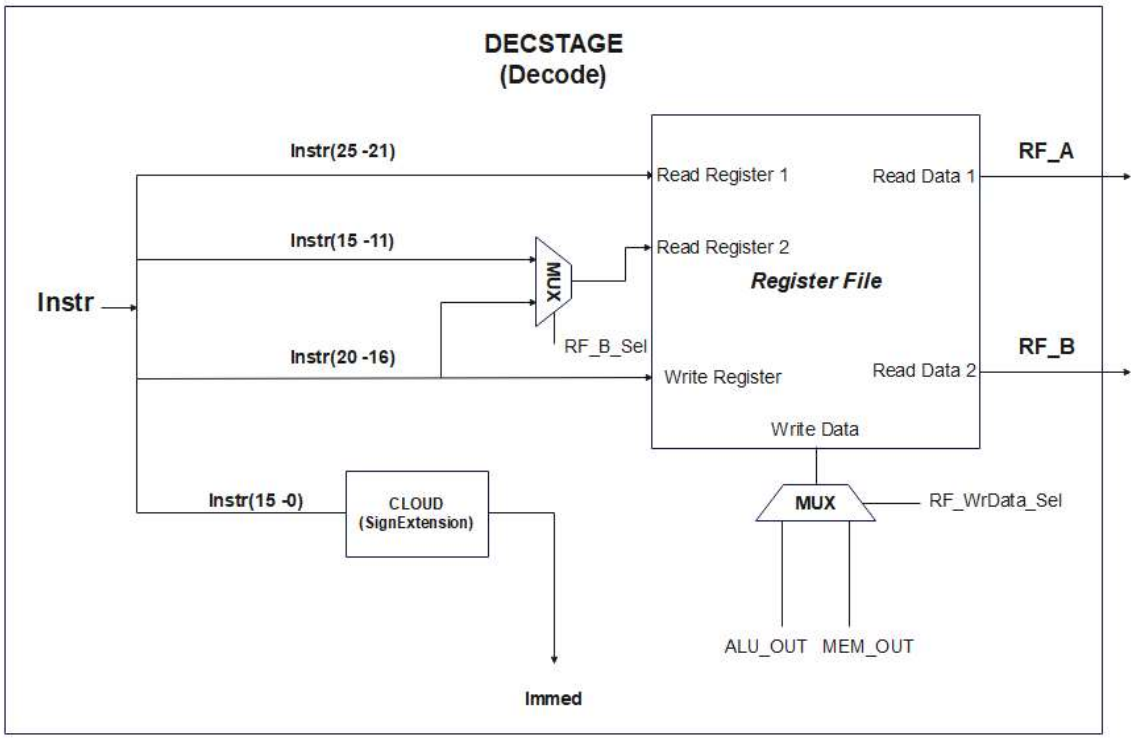
IFSTAGE:

Το συγκεκριμένο stage αφορά την ανάκληση των εντολών. Διαθέτει ένα καταχωρητή, ο οποίος αποθηκεύει την τιμή του PC. Το PC δείχνει στην διεύθυνση μνήμης που βρίσκεται η εντολή. Δεδομένου ότι κάθε εντολή έχει μέγεθος 32bits, ο PC μετά από κάθε εκτέλεση προχωρά κατά 4 για να διαβάσει την επόμενη εντολή. Η είσοδος PC_Immed χρησιμοποιείται κυρίως για εντολές branch και η εκτέλεση του προγράμματος να μεταβεί στην διεύθυνση PC + 4 + SignExt(Pc_Immed) << 2.



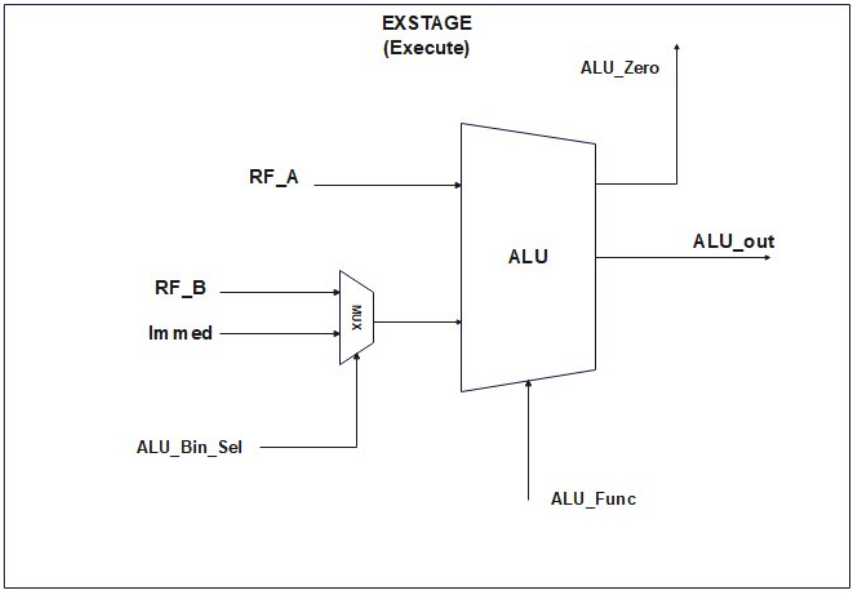
DECSTAGE:

Το συγκεκριμένο stage αφορά την αποκωδικοποίηση των εντολών. Έχει ως είσοδο την εντολή που έρχεται από τη μνήμη και τη συνδέει με το αρχείο καταχωρητών. Πιο συγκεκριμένα με έναν πολυπλέκτη επιλέγουμε κάθε φορά τη διεύθυνση του καταχωρητή που θέλουμε να γράψουμε μέσα στο αρχείο καταχωρητών και με άλλον έναν επιλέγουμε τα δεδομένα άλλοτε από την ALU και άλλοτε από τη μνήμη. Για τις εντολές I-Type, γίνεται χρήση ενός νέου module δημιουργήσαμε το οποίο επεξεργάζεται τον Immediate ανάλογα με το opcode της εντολής (cloud.vhd). Στη συνέχεια ο Immediate θα συνδεθεί με την προηγούμενη βαθμίδα, αλλά θα το δούμε πιο αναλυτικά στην 3^{η} φάση.



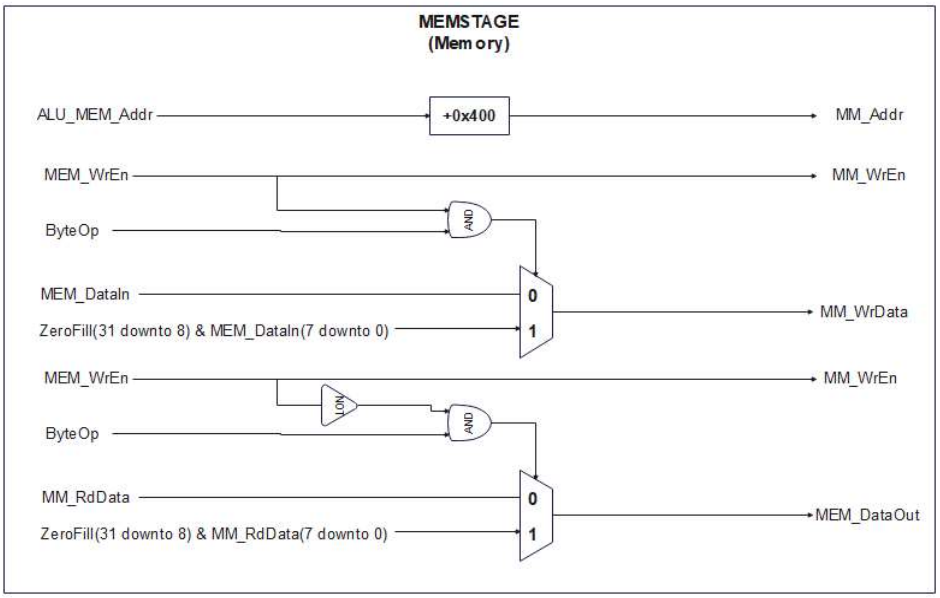
EXSTAGE:

Το συγκεκριμένο stage αφορά την εκτέλεση των εντολών και έχει απλή δομή. Αποτελείται από έναν πολυπλέκτη ο οποίος είναι υπεύθυνος για την επιλογή του δεύτερου τελεστή της ALU και την ALU. Πιο αναλυτικά, μετά την αποκωδικοποίηση της εντολής, τα δεδομένα των καταχωρητών της προηγούμενης βαθμίδας εισέρχονται ως είσοδο σε αυτό το στάδιο. Έτσι, με κατάλληλα σήματα ελέγχου μπορούμε να καθορίσουμε τις εσωτερικές συνδέσεις της βαθμίδας για να έχουμε το επιθυμητό αποτέλεσμα στην έξοδο της ALU.



MEMSTAGE:

Το συγκεκριμένο stage αφορά την πρόσβαση μνήμης. Είναι υπεύθυνο για τη μεταφορά των σημάτων ελέγχου, αλλά και των δεδομένων από και προς τη μνήμη. Το συγκεκριμένο κύκλωμα τροποποιεί τις εισόδους του κατάλληλα προκειμένου η μνήμη να έχει έγκυρες τιμές εισόδου και να δίνει σωστά αποτελέσματα στις εξόδους. Το σήμα ByteOp είναι υπεύθυνο για την επιλογή byte/word.

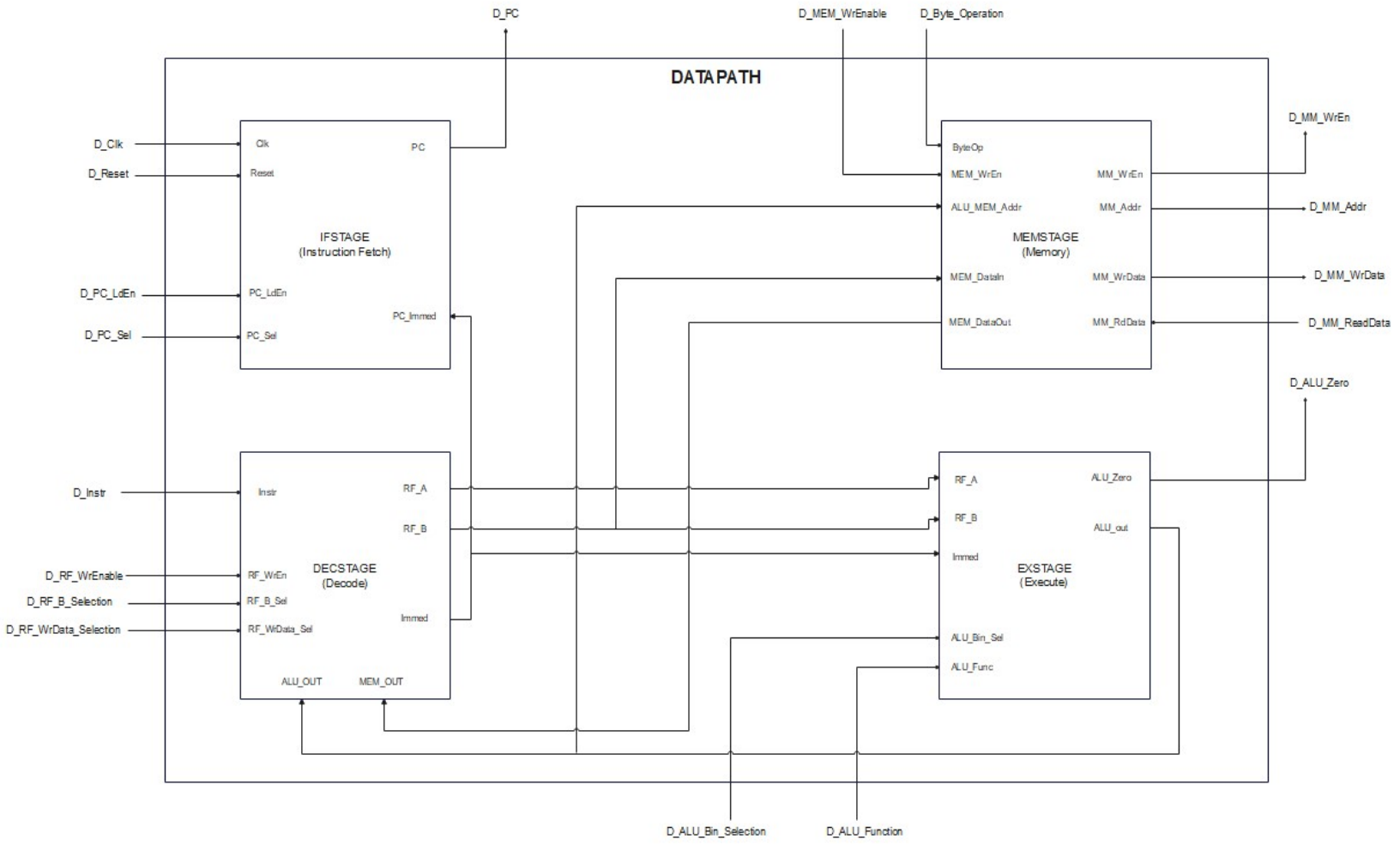


3η Φάση:

Στην 3η φάση του project, σκοπός μας είναι η υλοποίηση του Datapath, του Control αλλά και η μεταξύ τους σύνδεση/επικοινωνία για την ολοκλήρωση του single cycle processor.

DATAPATH:

Το συγκεκριμένο module υλοποιεί το Datapath του επεξεργαστή και αποτελείται από τις τέσσερις προηγούμενες βαθμίδες που υλοποιήσαμε στη δεύτερη φάση συνδεδεμένες μεταξύ τους. Η συνδεσμολογία των βαθμίδων φαίνεται ξεκάθαρα στο παρακάτω block diagram. Οι κυρίες έξοδοι είναι αυτές που θα συνδεθούν με τη μνήμη ενώ από την άλλη όλες οι υπόλοιπες είσοδοι θα ελέγχονται από το control.

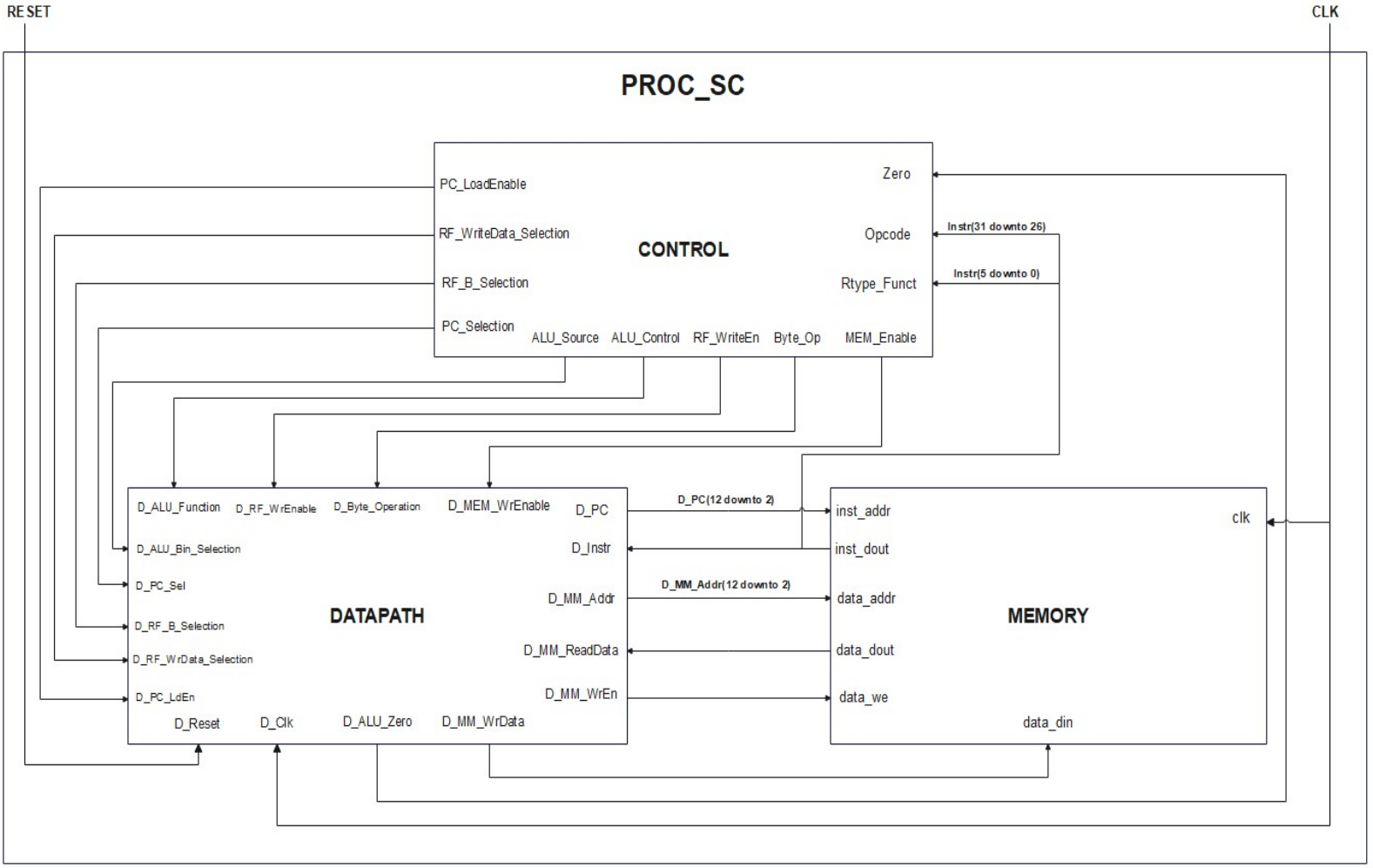


CONTROL:

Το συγκεκριμένο module πρόκειται για ένα συνδυαστικό κύκλωμα το οποίο είναι υπεύθυνο για την δημιουργία των σημάτων ελέγχου του Datapath. Διαθέτει πολλά σήματα εξόδου τα οποία συνδέονται άμεσα με το module του Datapath, αλλά και σήματα εισόδου όπως το Opcode, Rtype_Funct και Zero. Τα σήματα εισόδου του control είναι αυτά που καθορίζουν τα σήματα ελέγχου.

PROC_SC:

Το τελικό στάδιο του project, σύνδεση Control και Datapath. Παρακάτω φαίνεται το block diagram του επεξεργαστή ενός κύκλου.



ΤΕΛΟΣ ΕΡΓΑΣΤΗΡΙΑΚΉΣ ΑΝΑΦΟΡΆΣ