15/3/2018

# 2ή Εργαστηριακή Άσκηση

**ΣΧΕΔΙΑΣΗ ΒΑΣΙΚΩΝ ΒΑΘΜΙΔΩΝ ΤΟΥ DATAPATH ΕΝΟΣ ΑΠΛΟΥ ΕΠΕΞΕΡΓΑΣΤΗ**

**11/03/2018**

Ομάδα LAB31235515

|  |
| --- |
| *ΧΡΗΣΤΟΣ ΖΗΣΚΑΣ 2014030191* |
| *ΑΝΤΩΝΗΣ ΑΝΤΩΝΑΚΑΚΗΣ 2014030160* |

## Σκοπός εργαστηριακής άσκησης

Είναι η εντριβή με τη γλώσσα για τον ορισμό της αρχιτεκτονικής εντολών ως αντικείμενο μελέτης. Παράλληλα , ακολουθεί η σχεδίαση των βαθμίδων που συντελούν κομμάτια ενός απλού επεξεργαστή και αφορούν βαθμίδες για τις παρακάτω λειτουργίες:

* Ανάκληση εντολών
* Αποκωδικοποίηση εντολών
* Εκτέλεση εντολών
* Πρόσβαση μνήμης

Η υλοποίηση ολόκληρης της σχεδίασης διαθέτει το αρχείο καταχωρητών και την υπολογιστική μονάδα (ALU) σε συνδυασμό με επιπρόσθετα στοιχεία μνήμης καθώς και συνδυαστική λογική

## Προεργασία

Παρουσιάζονται οι συσχετίσεις που αφορά την αρχιτεκτονική συνόλου εντολών ενός non-pipelined επεξεργαστή βασισμένου σε υποσύνολο αρχιτεκτονικής συνόλου εντολών CHARIS-4 .

Η αρχιτεκτονική συνόλου εντολών αποτελείται:

* 32 καταχωρητές των 32 bits. Ο καταχωρητής R0 είναι πάντα μηδέν.
* 32 bit πλάτος εντολών µε µμέγεθος και θέση πεδίων που περιγράφονται  
  παρακάτω.
* Εντολές αριθµητικών και λογικών πράξεων: add, sub, and, not, or, shr, shl,  
  sla, rol, ror, li, addi, andi, ori.
* Εντολές διακλάδωσης: b, beq, bneq.
* Εντολές µνήµης: lb, sb, lw, sw.

Οι παραπάνω εντολές έχουν δύο τύπους format:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 6-bits | 5-bits | 5-bits | 5-bits | 5-bits | 6-bits |
| Opcode | rs | rd | rt | not-used | func |

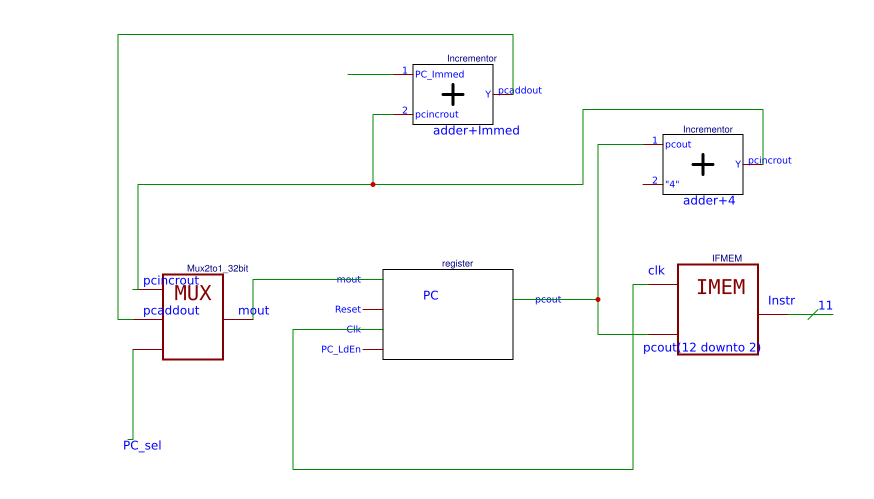
|  |  |  |  |
| --- | --- | --- | --- |
| 6-bits | 5-bits | 5-bits | 16-bits |
| Opcode | rs | rd | Immediate |

Παράλληλα , παρουσιάζονται σχηματικά διαγράμματα των βαθμίδων ξεχωριστά, για την απεικόνιση της λειτουργικότητας τους η οποία συνεισφέρει στη μερική υλοποίηση του datapath ενώ ολοκληρώνεται σε συνδυασμό με αντικείμενα συνδυαστικής λογικής.

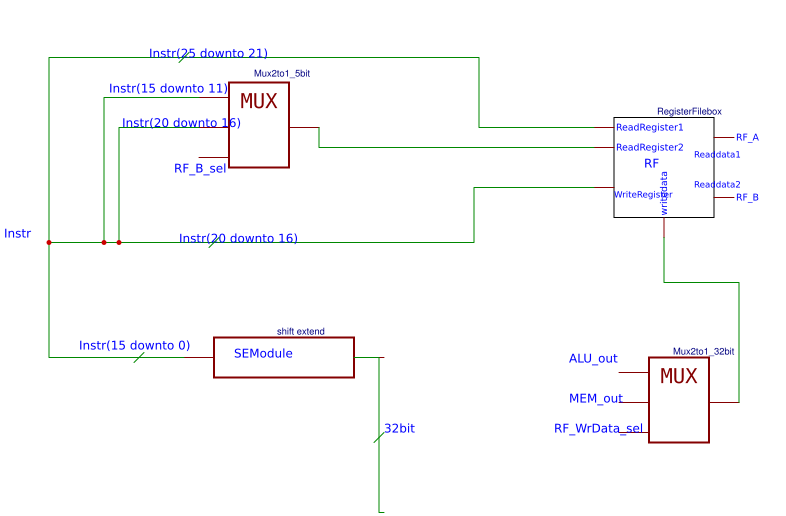
Κωδικοποίηση των εντολών

|  |  |  |  |
| --- | --- | --- | --- |
| Opcode | FUNC | ΕΝΤΟΛΗ | ΠΡΑΞΗ |
| 100000 | 110000 | add | RF[rd] ← RF[rs] + RF[rt] |
| 100000 | 110001 | sub | RF[rd] ← RF[rs] - RF[rt] |
| 100000 | 110010 | nand | RF[rd] ← RF[rs] NAND RF[rt] |
| 100000 | 110100 | not | RF[rd] ← ! RF[rs] |
| 100000 | 110011 | or | RF[rd] ← RF[rs] | RF[rt] |
| 100000 | 111000 | sra | RF[rd] ← RF[rs] >>1 |
| 100000 | 111001 | sll | RF[rd] ← RF[rs] <<1 (Logical, zero fill LSB) |
| 100000 | 111010 | srl | RF[rd] ← RF[rs] >>1 (Logical, zero fill MSB) |
| 100000 | 111100 | rol | RF[rd] ← Rotate left(RF[rs]) |
| 100000 | 111101 | ror | RF[rd] ← Rotate right(RF[rs]) |
| 111000 | - | li | RF[rd] ← SignExtend(Imm) |
| 111001 | - | lui | RF[rd] ← Imm << 16 (zero-fill) |
| 110000 | - | addi | RF[rd] ← RF[rs] + SignExtend(Imm) |
| 110010 | - | nandi | RF[rd] ← RF[rs] NAND ZeroFill(Imm) |
| 110011 | - | ori | RF[rd] ← RF[rs] | ZeroFill(Imm) |
| 111111 | - | b | PC ← PC + 4 + (SignExtend(Imm) << 2) |
| 000000 | - | beq | if (RF[rs] == RF[rd]) PC ← PC + 4 + (SignExtend(Imm) << 2) else PC ← PC + 4 |
| 000001 | - | bne | if (RF[rs] != RF[rd]) PC ← PC + 4 + (SignExtend(Imm) << 2) else PC ← PC + 4 |
| 000011 | - | lb | RF[rd] ← ZeroFill(31 downto 8) & MEM[RF[rs] + SignExtend(Imm)](7 downto 0) |
| 000111 | - | sb | MEM[RF[rs] + SignExtend(Imm)] ← ZeroFill(31 downto 8) & RF[rd] (7 downto 0) |
| 001111 | - | lw | RF[rd] ← MEM[RF[rs] + SignExtend(Imm)] |
| 011111 | - | sw | MEM[RF[rs] + SignExtend(Imm)] ← RF[rd] |

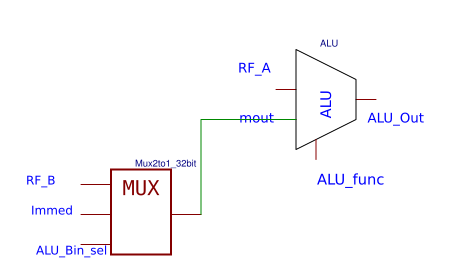
Βαθμίδα ανάκλησης εντολών (IF)



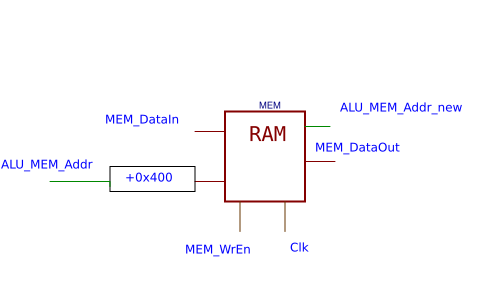
Βαθμίδα αποκωδικοποίησης εντολών (DECODE)



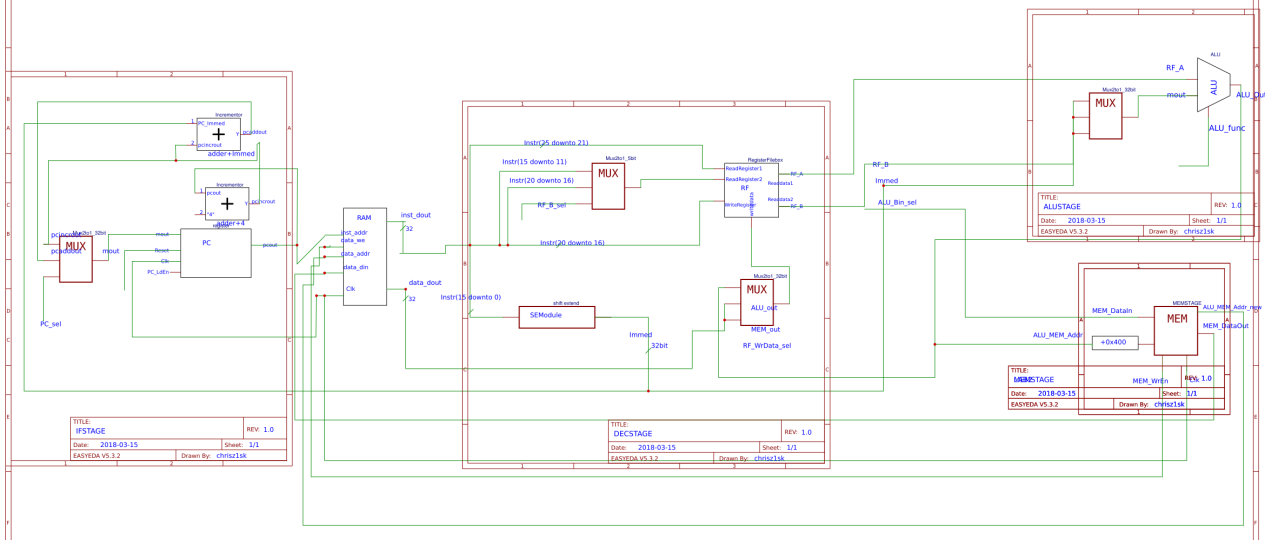
Βαθμίδα Εκτέλεσης Εντολών (ALU)



Βαθμίδα Πρόσβασης Μνήμη (MEM)



Συνολική Βαθμίδα (Datapath)



## Περιγραφή

Το ζήτημα της αρχιτεκτονικής εντολών αφορά την κωδικοποίηση του CHARIS , παρατηρώντας την ομαδοποίηση τους για την όσο το δυνατόν ευκολότερη αποκωδικοποίηση τους για την παραγωγή των σημάτων ελέγχου του ολοκληρωμένου συστήματος.

Όσον αφορά τη βαθμίδα IF συνεισφέρει στην οργάνωση της μνήμης σύμφωνα με τις εντολές CHARIS ενώ παράλληλα διευθετεί τις εντολές branch. Αποτελείται από τον καταχωρητή program counter , αθροιστές και ένα πολυπλέκτη που στις εντολές branch καθοδηγεί την μνήμη σύμφωνα με την τιμή που αποδίδεται στο Immidiate. Η τιμή του Immidiate υφίσταται ολίσθηση σε αυτές τις εντολές για σωστή διαχείριση της μνήμης . Διαφορετικά η διεύθυνση της μνήμης αυξάνεται κατά μια θέση και η οποία ενημερώνει τον καταχωρητή(+4 bits = 1 θέση στη μνήμη).

Ακολούθως, η βαθμίδα DECODE διαχειρίζεται από ένα αρχείο καταχωρητών, πολυπλέκτες και μια μονάδα που μετατρέπει το immidiate στη κατάλληλη κωδικοποίηση του για την περάτωση των εντολών της αρχιτεκτονικής. Ο πολυπλέκτης εισόδου δεδομένων αφορά εντολές είτε store και παίρνει δεδομένα από τη μνήμη και τα ενημερώνει για εγγραφή, είτε οποιεσδήποτε άλλες εντολές τις οποίες εισέρχονται από την έξοδο της ALU. Οι διευθύνσεις εγγραφείς εξαρτώνται από τις εκάστοτε εντολές που εκτελούνται και διαμορφώνονται από τα format των εντολών(ενεργοποιούνται/απενεργοποιούνται τα control των πολυπλεκτών) . Για R-type εντολές το control του πολυπλέκτη στην είσοδο του register file απενεργοποιείτε ώστε οι τιμές των καταχωρητών rs,rt να αλληλεπιδρούν και να αποθηκεύονται στον rd. H κωδικοποίηση του immidiate πραγματοποιείται σύμφωνα με την αρχιτεκτονική των εντολών.

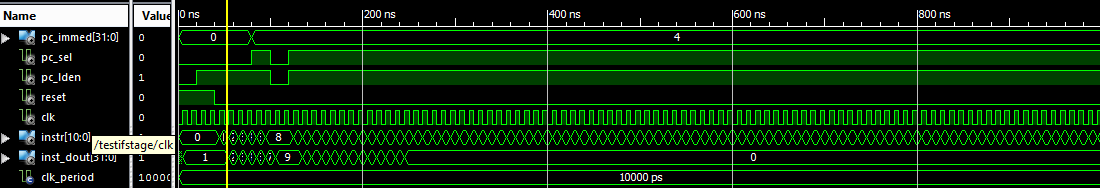
Επιπλέον , σύμφωνα με τη βαθμίδα ALU που διαθέτει την μονάδα υπολογισμών ALU καθώς και ένα πολυπλέκτη εξέρχονται τα αποτελέσματα που διευθετεί η ALU. Το control της ALU επηρεάζει τους υπολογισμούς (shift,logic,arithmetic) ενώ το control του πολυπλέκτη επηρεάζει την είσοδο και αφορά την εισροή του immidiate στο αποτέλεσμα λόγω της εκάστοτε εντολή (προέρχεται από την μονάδα κωδικοποίηση του immidiate)

Η σχεδίαση ολοκληρώνεται με την βαθμίδα ΜΕΜ .Η πραγματική διεύθυνση που φτάνει στη μνήμη είναι το εξερχόμενο από την ALU προστιθέμενη 0x400- η βαθμίδα ΜΕΜ το δέχεται ως διεύθυνση. Τα δεδομένα της μνήμης εισέρχονται από τη βαθμίδα MEM και κατευθύνονται στη μνήμη RAM . Εξυπηρετεί ουσιαστικά σαν ενδιάμεσο βήμα ώστε τα δεδομένα να οδηγούνται στη RAM

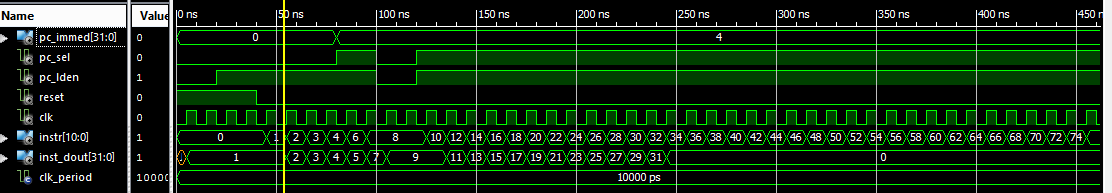
## Κυματομορφές-Προσομοίωση

Παρουσιάζονται οι κυματομορφές των ποικίλων βαθμίδων

Βαθμίδα ανάκλησης εντολών (Μονάδα IF)

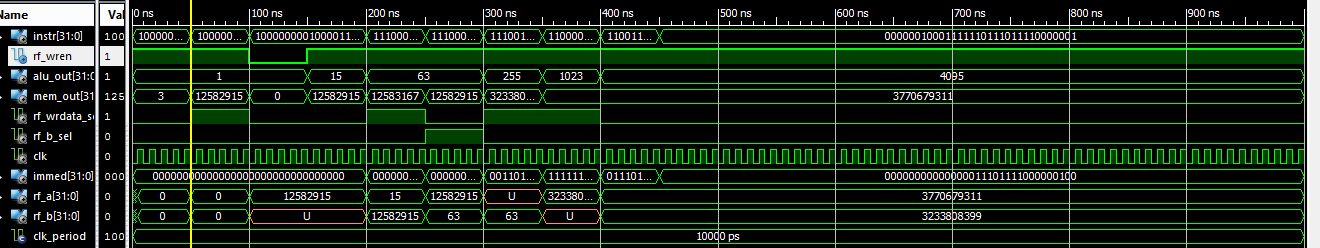


Η είσοδος που δέχεται η βαθμίδα IF επιδέχεται την λειτουργικότητα της σχεδίασης .Η μνήμη ξεκινάει από την μηδενική θέση και το σύστημα περιμένει να απενεργοποιηθεί το σήμα reset . Με reset & enable το σύστημα μένει αδρανές. Το reset “πέφτει” και η μνήμη διευθύνεται . Κάθε φορά που εκτελείται μια εντολή εκτός branch , η μνήμη παρατείνεται κατά μια θέση (+4 bits). Κατά την εκτέλεση εντολών branch η μνήμη αυξάνεται επιπρόσθετα σύμφωνα με την διεύθυνση του immidiate .Απενεργοποιείται το enable και η βαθμίδα αδρανοποιείται.

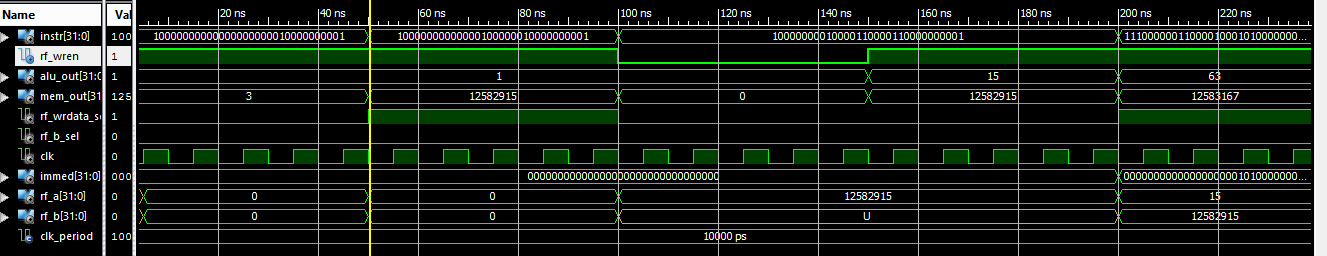


Ενεργοποιείται το enable και συνεχίζεται η λειτουργία του κυκλώματος θεωρώντας ότι δεχόμαστε εντολές branch και η τιμή της μνήμη μεγαλώνει κατά την τιμή του Immidiate- ολισθαίνοντας το κατά 2 θέσεις ώστε να αναφερόμαστε σε θέσεις μνήμης- αναφερόμενοι πάντα για word addresable μνήμη. Η κατάληξη λοιπόν της σχεδίασης φτάνει στο να διαβάζει ολόκληρο το αρχείο εισόδου χωρίς και με διακλάδωση στη μνήμη.

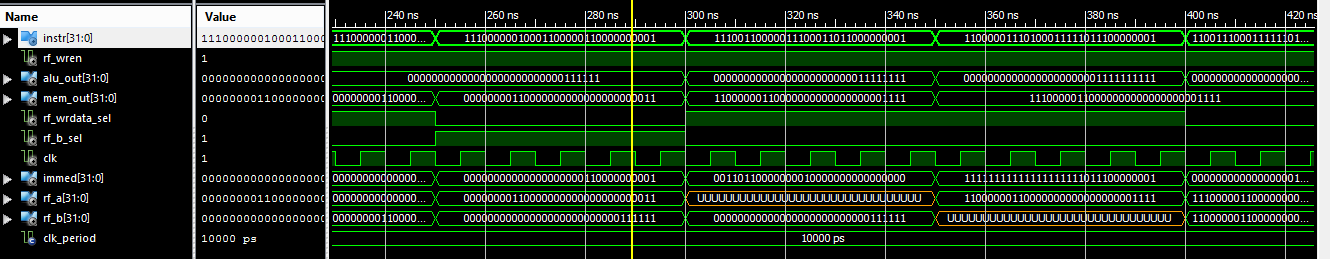
Βαθμίδα αποκωδικοποίησης εντολών (Μοναδα DEC)



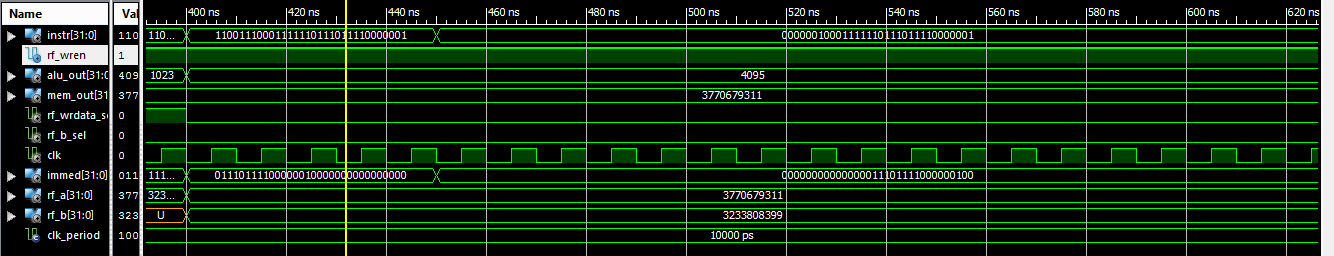
Αυτό το κομμάτι της σχεδίασης αποκωδικοποιεί την εντολή που εξάγεται από τη μνήμη. Χωρίζεται η εντολή σε κομμάτι ανάγνωσης & εγγραφής ,Opcode , Immidiate και τα δεδομένα εισόδου τίθενται προς εγγραφή στο αρχείο καταχωρητών.



Οι αρχικές διευθύνσεις ανάγνωσης αφορούν το μηδενικό καταχωρητή στον οποίο είναι αδύνατο να υπάρξει εγγραφή . Ως παραγόμενο εμφανίζεται μηδενική έξοδος ενώ το Immidiate εξαρτάται από τον τύπο των εντολών που υλοποιούνται (Rtype, Itype , Store, branch,Load). Για R-type το Immidiate είναι μηδέν. Διαφοροποιείται μόνο η διεύθυνση εγγραφής οπότε στον καταχωρητή 2 εγγράφεται η τιμή που προέρχεται από την MEM. Εξαρτάται από τον έλεγχο του πολυπλέκτη της εισόδου δεδομένων. Για 0 το αποτέλεσμα προέρχεται από την ALU αλλιώς από την μνήμη. Ως συνέχεια γίνεται ανάγνωση στο καταχωρητή 2 και 1 ενώ γράφεται ο καταχωρητής 3. Απενεργοποιείται το enable οπότε δεν υπάρχει εγγραφή. Ενεργοποιείται και συνεχίζεται . Ως είσοδο ανάγνωσης μπαίνει η προηγούμενη διεύθυνση και επαληθεύεται η εγγραφή.

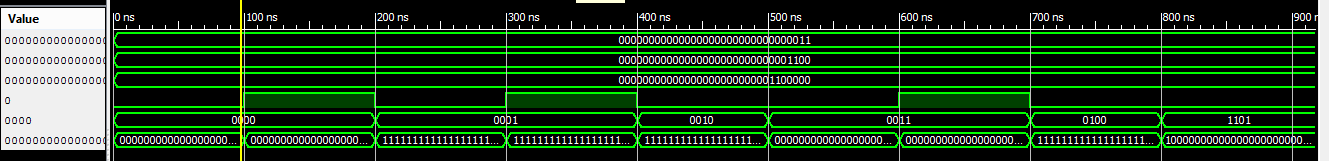


Συνεχίζεται το τεστ για εντολές τύπου Ι. Το immidiate δέχεται sign extend ενω η εγγραφή προέρχεται από την μνήμην. Οι εγγραφές συνεχίζουν και επαληθεύονται προοδευτικά ενώ παρουσιάζονται και οι μετατροπές του immidiate σύμφωνα με το opcode που εμφανίζει η εντολή. Επαληθεύεται ο καταχωρητής 3 και γράφεται ο καταχωρητής 6 από την ALU. Το control του πολυπλέκτη εισόδου όταν είναι ενεργό βοηθάει στο να περνάει η διεύθυνση εγγραφής και ανάγνωσης#2. Αποθηκεύονται τιμές στους διάφορους καταχωρητές ενώ παράλληλα ελέγχονται και οι τιμές που λαμβάνει το immidiate



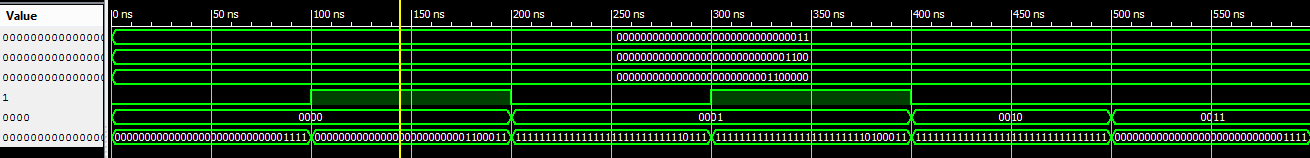
|  |  |  |
| --- | --- | --- |
| Opcode | Instr-16bit | Immidiate-32 bit |
| 111000 (SignExtend) | 0000110000000001 | 00000000000000000000110000000001 |
| 111001 (Imm<<16(zerofill)) | 0011011000000001 | 00110110000000010000000000000000 |
| 110000 (SignExtend(Imm)) | 1111011100000001 | 11111111111111111111011100000001 |
| 110011 (ZeroFill(Imm)) | 0111011110000001 | 00000000000000000111011110000001 |
| 000000SignExtend(Imm)<<2 | 0111011110000001 | 00000000000000011101111000000100 |
| 100000 | X | 00000000000000000000000000000000 |

Βαθμίδα εκτέλεσης εντολών (Μονάδα ALU)

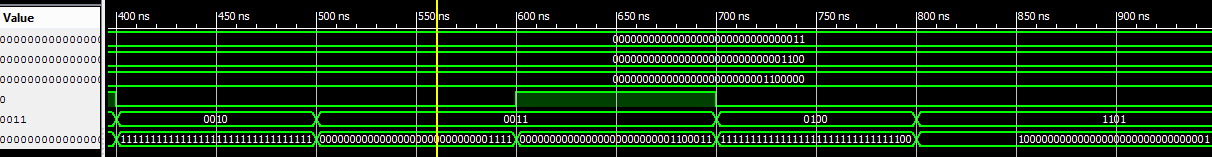


Η βαθμίδα ALU ειδικεύεται στην αναγνώριση των πράξεων και επιπλέον διαμορφώνει την επιλογή των τελεστέων εξετάζοντας τα ανάλογα σήματα (Alu\_Bin\_Sel). Για Alu\_bin\_sel = 0,οι τελεστέοι είναι οι έξοδοι από τα rf ενώ για τιμή 1 ο δεύτερος τελεστέος διαμορφώνεται από το immidiate. Η ALU εκτελεί την λειτουργικότητα της σύμφωνα με τα control που διαθέτει και επιπλέον πραγματοποιεί τις πράξεις σύμφωνα με τα func.

Παρατηρείται ότι αρχικά προστίθενται οι έξοδοι από το rf και στη συνέχεια γίνεται η πρόσθεση του rfa με το Immidiate. Ακολούθως πραγματοποιείται η πράξη της αφαίρεσης με όμοια ακολουθία .



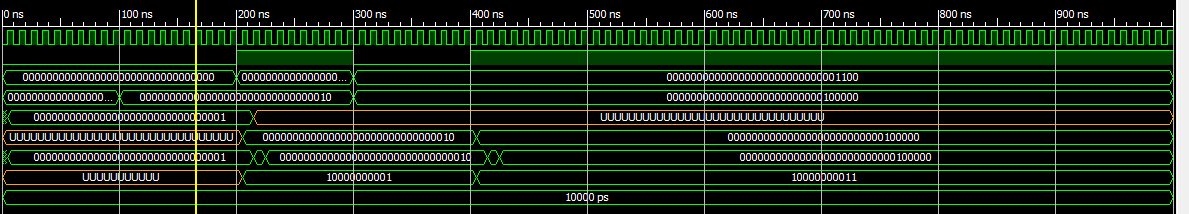
Επαληθεύεται η συνολική σχεδίαση εφαρμόζωντας τη λογικές και ολισθητικές πράξεις.



|  |  |  |
| --- | --- | --- |
| 0001 | ...000011- ..001100 | 11111111111111111111111111110111 |
| 0010 | ...000011 NAND ..001100 | 11111111111111111111111111111111 |
| 0011 | ...000011 OR ..001100 | ..001111 |
| 0100 | 00000000000000000000000000000011 | 11111111111111111111111111111100 |
| 1101 | 00000000000000000000000000000011 | 10000000000000000000000000000001 |
| 0000 | ...000011+..001100 | ..001111 |

Βαθμίδα πρόσβασης μνήμης (Μονάδα ΜΕΜ)

Η βαθμίδα MEM συμπεριφέρεται ως ενδιάμεσο μέσο αποθήκευσης. Για enable η είσοδος των δεδομένων διαπερνά τη βαθμίδα ώστε να καταχωρηθούν στη ram τα δεδομένα



Και η διεύθυνση εξόδου να τεθεί ως εξαγόμενο. Για απενεργοποίηση-ενεργοποίηση του enable η μνήμη επαναλαμβάνει εγγραφή δεδομένων.

## Συμπεράσματα/Προβλήματα

Η εργαστηριακή άσκηση καταλήγει σε σημαντικά αποτελέσματα καθώς συνεχίζεται η πρόοδος της υλοποίηση του επεξεργαστή με την ανάλυση σε βαθμίδες επεξεργασίας εντολών ενώ εξετάζεται η κατάτμησή και η διαχείριση της μνήμης για datapath ενός κύκλου