

Οργάνωση Υπολογιστών – ΗΡΥ312

Αναφορά 2ής Εργασίας

Δημιουργία επεξεργαστή πολλαπλών κύκλων και
μετατροπή του σε pipeline



Ιάσωνας Γεωργακάς
2017030021

Παναγιώτης Βασιλείου
2017030067

LAB31243750
Μάιος, 2020
Πολυτεχνείο Κρήτης

Contents

1	Εισαγωγή	3
2	Τεχνολογία	3
3	Υλοποίηση	3
3.1	Υλοποίηση Multicycle	3
3.2	Υλοποίηση Pipeline	4
4	Αποτελέσματα	4
4.1	Αποτελέσματα Multicycle	4
4.2	Αποτελέσματα Pipeline	4
5	Συμπεράσματα	5
6	Παράρτημα (Block and State Diagrams)	6

1 Εισαγωγή

Στην παρούσα εργασία σκοπός ήταν η υλοποίηση ενός επεξεργαστή πολλαπλών κύκλων και ενός pipeline επεξεργαστή χρησιμοποιώντας τη γλώσσα περιγραφής υλικού VHDL, εφαρμόζοντας τεχνική σχεδίασης bottom-up. Η υλοποίηση των δύο επεξεργαστών βασιστήκε στα stages του επεξεργαστή ενός κύκλου διαφοροποιώντας το if stage και το execution stage για τις ανάγκες του εκάστοτε επεξεργαστή. Η επαλήθευση της λειτουργίας του επεξεργαστή πολλαπλών κύκλων πραγματοποιήθηκε για όλες τις εντολές του ISA ενώ για τον pipeline επεξεργαστή έγινε για 4 εντολές του ISA, συγκεκριμένα τις li, lw, sw και add καθώς δεν υποστηρίζει branch hazards.

2 Τεχνολογία

Για την υλοποίηση της εργασίας χρησιμοποιήθηκε η σουίτα λογισμικού Xilinx ISE 14.7 και πιο συγκεκριμένα τα εργαλεία XST για σύνθεση και ISim για προσομοίωση. Η γλώσσα περιγραφής υλικού (HDL) πάνω στην οποία αναπτύχθηκε ο επεξεργαστής ήταν η VHDL-200X.

3 Υλοποίηση

3.1 Υλοποίηση Multicycle

Η αρχή λειτουργίας του επεξεργαστή πολλαπλών κύκλων είναι η εκτέλεση των εντολών σε παραπάνω από 1 κύκλο. Συγκεκριμένα, οι εντολές διακλάδωσης εκτελούνται σε 3 κύκλους, οι R-type, I-type και store εντολές χρειάζονται 4 κύκλους για να ολοκληρωθούν ενώ οι load εντολές χρειάζονται 5 κύκλους. Το πλεονέκτημα του Multi-Cycle επεξεργαστή έναντι του Single-Cycle εντοπίζεται στο μικρότερο κύκλο ρολογιού. Επίσης, καθώς εκτελούνται οι εντολές σε πολλούς κύκλους επιτυγχάνεται ο διαμοιρασμός του hardware και επομένως η μείωση του. Ειδικότερα, οι δύο adders που υπήρχαν στο instruction fetch του Single-Cycle επεξεργαστή αφαιρέθηκαν αφού η λειτουργία τους μπορεί να πραγματοποιηθεί με τη χρήση της ALU η οποία διαφορετικά θα έμενε ανενεργή για αρκετούς κύκλους. Ακόμη, προστέθηκαν καταχωρητές ανάμεσα στα stages ώστε να αποθηκεύονται τα δεδομένα που χρειάζεται η εκάστοτε εντολή σε κάθε stage και να εκτελεστεί σωστά. Με την αφαίρεση των δύο adders χρειάστηκε η αύξηση των εισόδων των πολυπλέκτων στο execution stage ώστε να εκτελούνται οι πράξεις που αφορούν τον PC.

Τα control σήματα του Multi-Cycle επεξεργαστή παράγονται από μία μηχανή πεπερασμένων καταστάσεων, η οποία παράγει τα σήματα εξόδου της με βάση την κατάστασή της και το instruction (Mealy FSM). Ο πίνακας των σημάτων εξόδου του Control καθώς και το state diagram παρατίθενται παρακάτω.

3.2 Υλοποίηση Pipeline

Η αρχή λειτουργίας του pipeline επεξεργαστή βασίζεται στην παράλληλη εκτέλεση των εντολών με σκοπό την αύξηση του throughput. Συγκεκριμένα, υλοποιήθηκαν οι pipeline registers IF/ID, ID/EX, EX/MEM και MEM/WB (5-stage pipeline) οι οποίοι συνδέουν τα stage και μεταβιβάζουν τα δεδομένα και τα σήματα ελέγχου που χρειάζεται κάθε stage, με αποτέλεσμα να εκτελείται διαφορετικό μέρος διαφορετικής εντολής σε κάθε stage. Στη συνέχεια, μεταφέρθηκε ο ένας εκ των δύο adders του Instruction fetch στο Execution stage, ώστε να έχει εγγραφεί στον PC, η τιμή $PC+4$.

Τα σήματα ελέγχου παράγονται από το module Control, το οποίο είναι το ίδιο module που χρησιμοποιήθηκε στον Single-Cycle επεξεργαστή καθώς η κάθε εντολή στον pipeline επεξεργαστή χρειάζεται να έχει όλα τα κοντρόλ σήματα των επόμενων βαθμίδων στον ID/EX έτσι ώστε να τα χρησιμοποιεί στην ανάλογη βαθμίδα.

Ένα πρόβλημα που συναντάται κατά την υλοποίηση ενός pipeline επεξεργαστή είναι τα data, branch και structural hazards. Στην παρούσα υλοποίηση ζητήθηκε η αντιμετώπιση μόνο των data hazards (structural hazards δεν υπάρχουν εκ των προτέρων) με τη χρήση των τεχνικών forwarding και stall. Η τεχνική forwarding χρησιμοποιείται για να προωθηθούν τα δεδομένα ενός καταχωρητή που ακόμα δεν έχουν εγγραφεί στο αρχείο καταχωρητών ή στη μνήμη αλλά χρειάζονται αυτά τα δεδομένα σε μια επόμενη εντολή. Με την τεχνική των stalls επιτυγχάνεται η καθυστέρηση των εντολών που χρησιμοποιούν ένα καταχωρητή για τον οποίο δεν έχουν υπολογιστεί τα δεδομένα. Για την εφαρμογή των δύο προαναφερθέντων τεχνικών υλοποιήθηκαν δύο modules το Hazard Detection Unit και το Forwarding Unit.

4 Αποτελέσματα

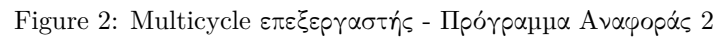
4.1 Αποτελέσματα Multicycle

Για τις κυματομορφές του Multicycle επεξεργαστή παρατηρείται ότι η διάρκεια της εκτέλεσης των εντολών διαρκεί περισσότερο από την αντίστοιχη διάρκεια εκτέλεσης στον Single-Cycle επεξεργαστή, το οποίο αναμένεται αφού χρησιμοποιώντας το ίδιο ρολόι (περίοδος ρολογιού 100ns) ο Multicycle εκτελεί μία εντολή απο 3 μέχρι 5 κύκλους.

Στο Προγραμμα Αναφοράς 2 (Figure 2) παρατηρήθηκε ότι ο Program Counter πραγματοποιεί τις ακόλουθες μεταβάσεις: $0 \rightarrow 4 \rightarrow 8 \rightarrow 0$, το οποίο αναμένεται καθώς ο Program Counter αυξάνεται στο πρώτο κύκλο και η επαναφορά του στο 0 από την εκτέλεση της branch γίνεται στον 3ο κύκλο.

4.2 Αποτελέσματα Pipeline

Στο πρόγραμμα αναφοράς prog11.data παρατηρείται ότι ο συνολικός χρόνος εκτέλεσής του είναι 1us. Ο χρόνος αυτός είναι αρκετά μικρότερος σε σχέση με τα



The timing diagram illustrates the behavior of the RISC-V processor over a 13,100 ns period. The signals shown include:

- NOT**: A signal that transitions from 0 to 1 at approximately 1,000 ns and remains high.
- CLK**: A periodic clock signal with a period of approximately 1,000 ns.
- PC**: The Program Counter, which starts at 0 and increments by 4 at each clock cycle, reaching 44 at 13,100 ns.
- PC_L0EN**: A signal that transitions from 0 to 1 at approximately 1,000 ns and remains high.
- RF_WRITE_ENABLE**: A signal that transitions from 0 to 1 at approximately 1,000 ns and remains high.
- RF_ADDRESS_WRITE**: A signal that transitions from 0 to 1 at approximately 1,000 ns and remains high.
- RF_WRITE_DATA**: A signal that transitions from 0 to 1 at approximately 1,000 ns and remains high.
- MEM_READ**: A signal that transitions from 0 to 1 at approximately 1,000 ns and remains high.
- MEM_WRITE_ENABLE**: A signal that transitions from 0 to 1 at approximately 1,000 ns and remains high.
- R3**: Register 3, which starts at 0 and increments by 4 at each clock cycle, reaching 44 at 13,100 ns.
- R5**: Register 5, which starts at 0 and increments by 4 at each clock cycle, reaching 44 at 13,100 ns.
- R10**: Register 10, which starts at 0 and increments by 4 at each clock cycle, reaching 44 at 13,100 ns.
- R15**: Register 15, which starts at 0 and increments by 4 at each clock cycle, reaching 44 at 13,100 ns.
- RAM[256]**: Memory location 256, which starts at 0 and increments by 4 at each clock cycle, reaching 44 at 13,100 ns.
- RAM[257]**: Memory location 257, which starts at 0 and increments by 4 at each clock cycle, reaching 44 at 13,100 ns.
- RAM[258]**: Memory location 258, which starts at 0 and increments by 4 at each clock cycle, reaching 44 at 13,100 ns.
- RAM[259]**: Memory location 259, which starts at 0 and increments by 4 at each clock cycle, reaching 44 at 13,100 ns.
- RAM[255]**: Memory location 255, which starts at 0 and increments by 4 at each clock cycle, reaching 44 at 13,100 ns.

Figure 3: Pipeline επεξεργαστής - Πρόγραμμα Αναφοράς prog11.data

5 Συμπεράσματα

Ο Multi-Cycle επεξεργαστής θα εκτελέσει πιο πολλές εντολές ανά μονάδα χρόνου σε σχέση με τον Single-Cycle, παράλληλα μειώνοντας τις απαιτήσεις σε hardware. Το μειονέκτημα αυτής της αρχιτεκτονικής είναι το πιο πολύπλοκο Control.

Ο Pipeline επεξεργαστής μειώνει σημαντικά το συνολικό χρόνο εκτέλεσης

των εντολών σε σχέση με τους Single-Cycle και Multi-Cycle επεξεργαστές και διατηρεί μικρή πολυπλοκότητα όσον αφορά την υλοποίηση του Control σε σχέση με τον Multi-Cycle. Όμως το μειονέκτημα του είναι ότι απαιτεί περισσότερο hardware διότι τα hazards που μπορεί να προκύψουν απαιτούν για την αντιμετώπιση τους επιπρόσθετη λογική. Για την αύξηση της απόδοσης και στους δύο επεξεργαστές μπορεί να επιτευχθεί με την αύξηση των stages. Στον Multi-Cycle επεξεργαστή κάτι τέτοιο αυξάνει τη δυσκολία του Control ενώ στον Pipeline αυξάνει τη δυσκολία αντιμετώπισης των hazards. Η αύξηση των stages ωστόσο συνίσταται μόνο στον Pipeline επεξεργαστή μιας και το speedup σε αυτήν την περίπτωση είναι αρκετά μεγάλο.

6 Παράρτημα (Block and State Diagrams)

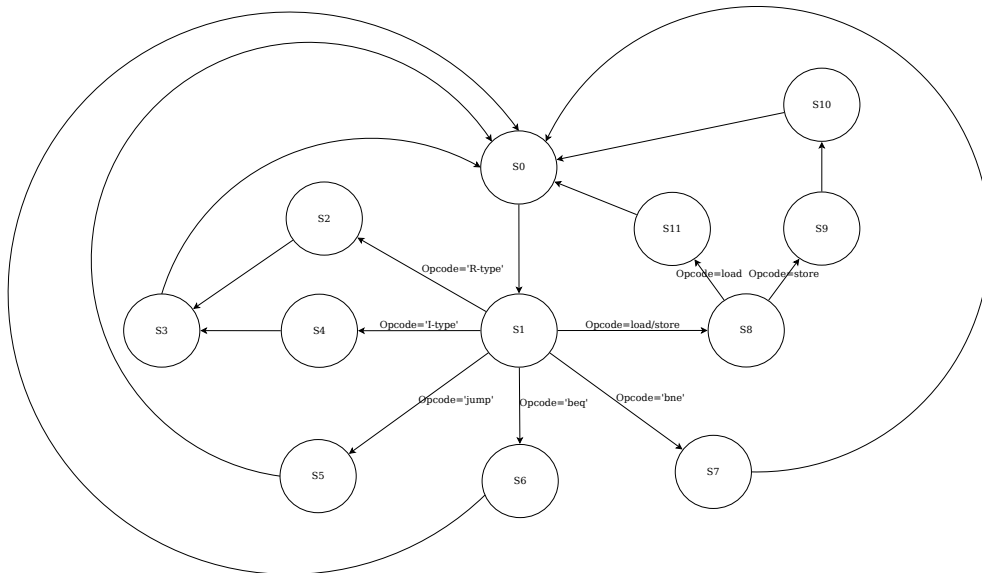


Figure 4: Διάγραμμα καταστάσεων (state diagram) του Multicycle επεξεργαστή

Output	State and/or Input
ALU_Op = "000"	S2
ALU_Op = "001"	S0, S1, S8, (S4 AND Opcode='addi')
ALU_Op = "010"	S6, S7
ALU_Op = "011"	S4 AND (Opcode='li' OR 'lui' OR 'ori')
ALU_Op = "100"	(S4 AND Opcode='nandi')
ALU_Ain_sel = 1	S0, S1
ALU_Bin_sel = "01"	S1, S4, S8
ALU_Bin_sel = "10"	S0
RF_A_sel = '1'	S1 AND (Opcode='li' OR 'lui')
RF_B_sel = '0'	S1 AND (Opcode='R-type')
MEM_WrEn = '1'	S11
RF_WrEn = '1'	S3, S10
RF_WrData_sel = '1'	S10
ByteOp = '1'	(S10 AND Opcode='lb'), (S11 AND Opcode='sb')
Immed_ctrl = "00"	S1 AND Opcode = 'lui'
Immed_ctrl = "01"	S1 AND (Opcode = 'nandi' OR 'ori')
Immed_ctrl = "10"	S0, (S1 AND (Opcode = 'b' OR 'beq' OR 'bne'))
Immed_ctrl = "11"	S1 AND (Opcode = 'addi' OR 'li' OR 'lb' OR 'sb' OR 'lw' OR 'sw')
PC_sel = '1'	S5, S6, S7
PC_LdEn = '1'	S0, S5, (S6 AND ALU_Zero = 1), (S7 AND NOT ALU_Zero)
IR_WrEn = '1'	S0

Table 1: Διάγραμμα καταστάσεων (state diagram) της FSM του Multicycle επεξεργαστή

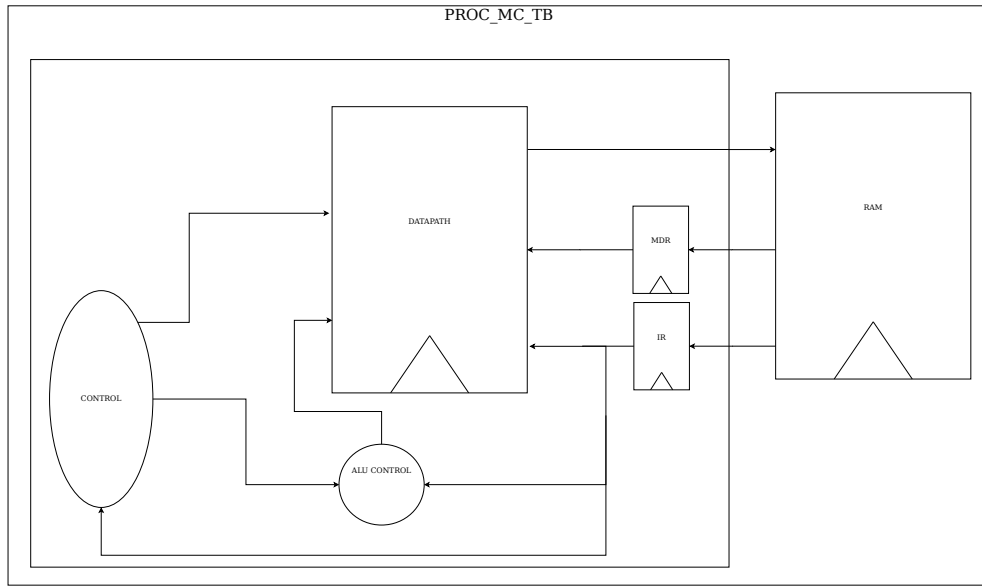


Figure 5: Top-Level διάγραμμα του Multicycle επεξεργαστή

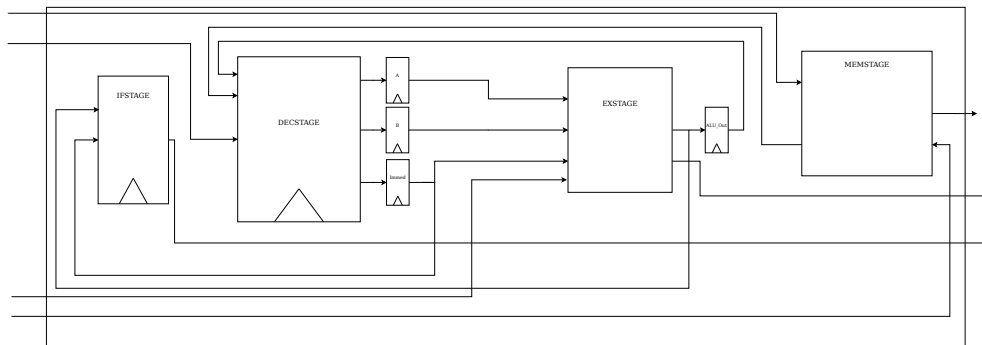


Figure 6: Διάγραμμα του Datapath του Multicycle επεξεργαστή

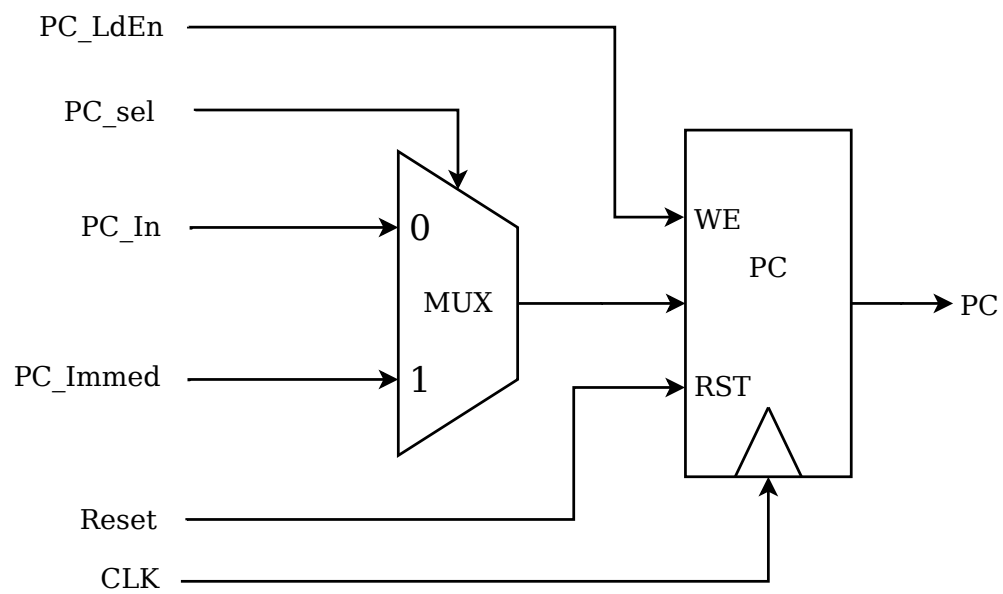


Figure 7: Διάγραμμα IFSTAGE του Multicycle επεξεργαστή

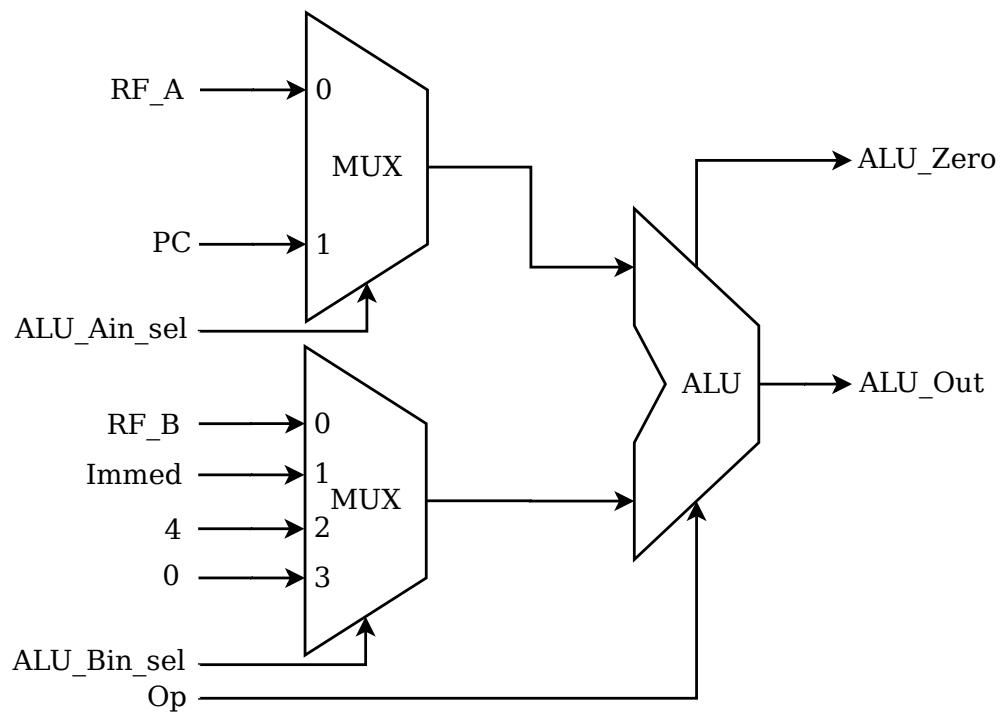


Figure 8: Διάγραμμα EXSTAGE του Multicycle επεξεργαστή

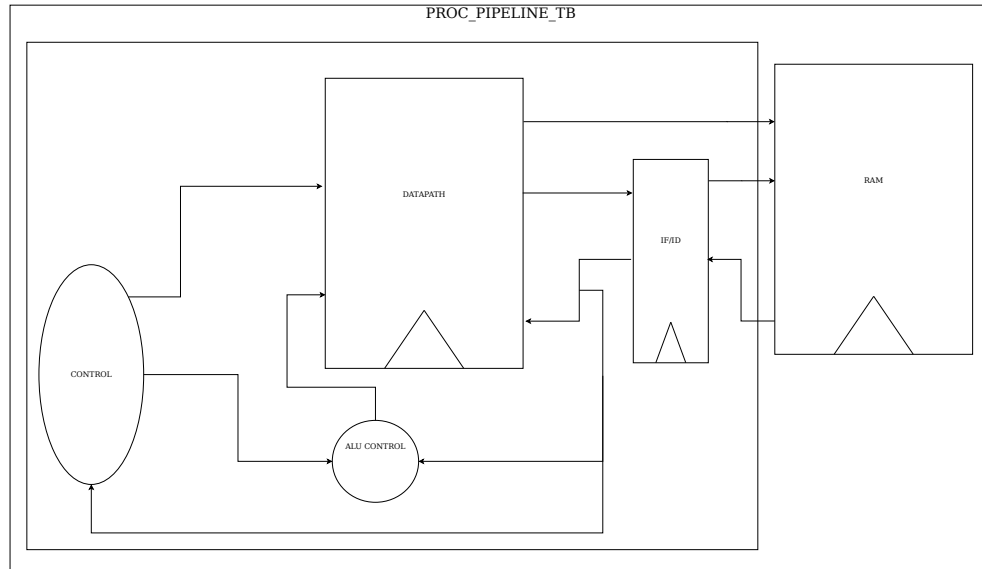


Figure 9: Top-Level διάγραμμα του Pipeline επεξεργαστή

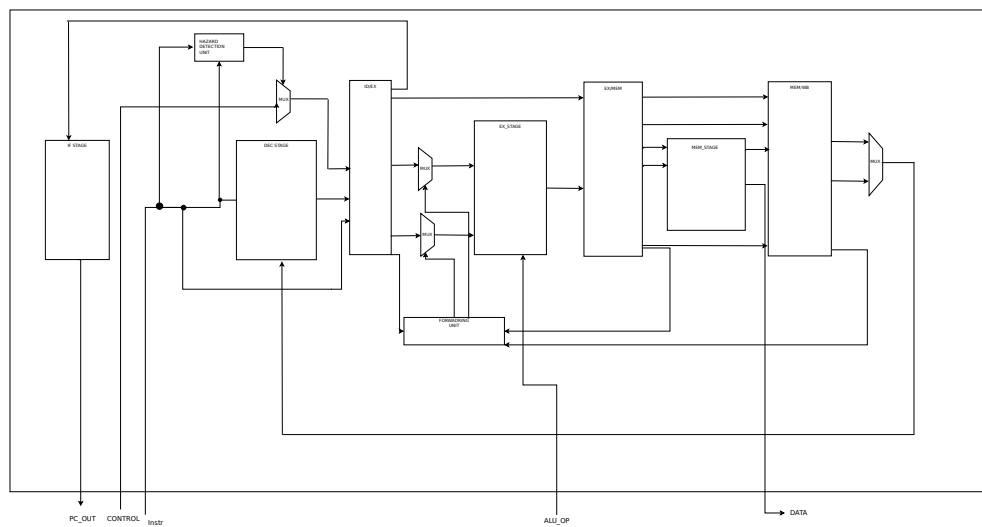


Figure 10: Διάγραμμα του Datapath του Pipeline επεξεργαστή

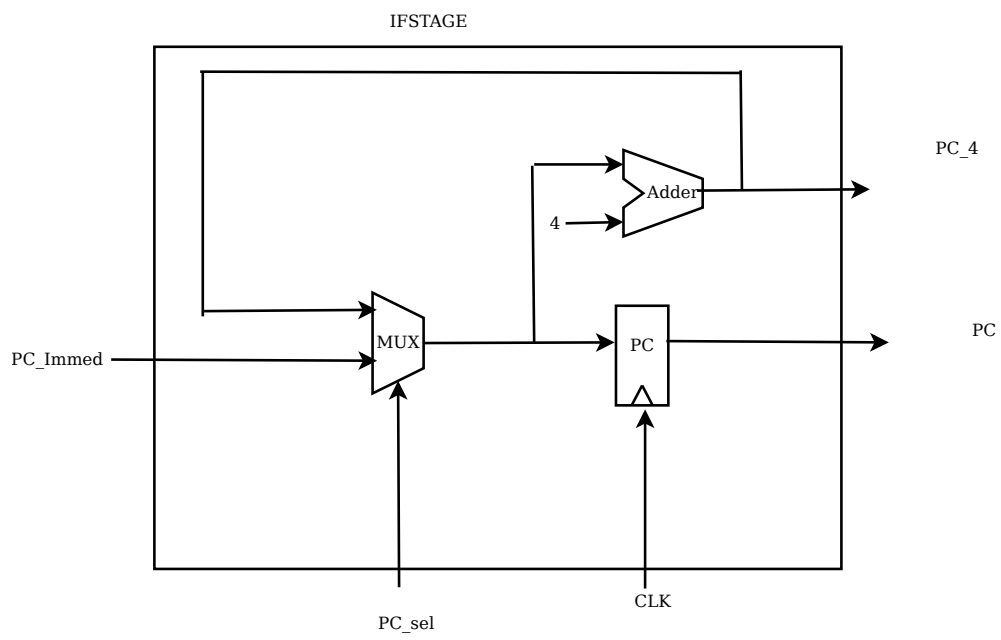


Figure 11: Διάγραμμα IFSTAGE του Pipeline επεξεργαστή

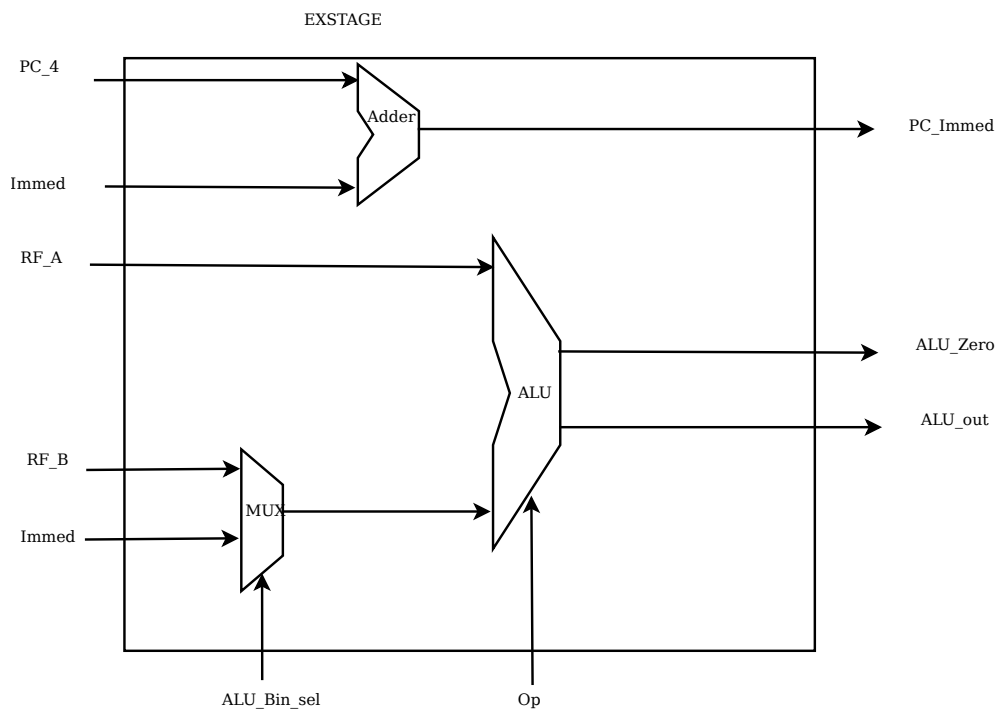


Figure 12: Διάγραμμα EXSTAGE του Pipeline επεξεργαστή