**ΘΕΜΑ 1:**

Για την δημιουργία του single-cycle mips με περισσοτερες εντολές αρχικά πήγα στο control και πρόσθεσα ολες τις εντολές που χρειαζόμαστε.Για να τις προσθέσω πρώτα βρήκα τα opcodes τους και μετα για κάθε εντολή έβρισκα τα (RegDst, ALUSrc, MemtoReg, RegWrite, MemRead, MemWrite, Branch, AluOP) και έτσι πρόσθετα τις εντολές.

Μετα έπρεπε να αλλάξω το alu και να προσθέσω και εκεί τις εξτρα εντολές αλλα επειδή με τα 4 bit που είχαμε ως output στο εργαστήριο δεν μπορούσαν να μπούν όλες οι εντολές έπρεπε να το κάνω 5 bit.Μετά πρόσθεσα όλες τις εντολές τους έβαλα το κατάλληλο ALUop και την κατάλληλη πράξη για κάθε εντολή.

Το τελευταίο πράγμα που έπρεπε να αλλάξω ηταν το alucontrol που έπρεπε να αυξήσω το AluCtrl σε 5 bit για να χωράνε ολες οι εντολές και μετά αντιστοίχισα την κάθε εντολή με το σωστό AluCtrl απο την alu.Στο τέλος έπρεπε να βρώ το AluOp και Func code τις κάθε εντολής και να το αντιστοιχίσω με κάθε εντολή.

**ΘΕΜΑ 2:**

Για την επέκταση του single-cycle mips με pipeline αρχικά έπρεπε να δημιουργήσω του 4 καταχωρητές.Πρώτα δημιούργησα των if\_id\_reg που είναι ο καταχωρητής μεταξύ του IF:instruction fetch και του ID: instruction decode που θα περνά τα αποθηκευμένα αποτελέσματα απο το πρώτο στάδιο στο επόμενο στάδιο με τον επόμενο κύκλο ρολογίου.Μετά δημιούργησα τους υπόλοιπυς 3 καταχωρητές που κάνουν την ίδια δουλεία αλλα ο ID: instruction decode στον EX: execute operation, ο EX: execute operation στον MEM: access memory και ο MEM: access memory στο WB: write result.

Μετά δημιούργησα το hazard detection unit που βρίσκει τους πιθανούς Κίνδυνος δομής (structure hazards), Κίνδυνος δεδομένων (data hazard) η Κίνδυνος ελέγχου (control hazard).