# FINAL PROJECT

# EXTRA CREDIT

# Aarya Soni

# CMPEN 331- Section 2

# Dr. Almekkawy

## Abstract:

In my Verilog implementation of the pipelined MIPS CPU, I modularized each component corresponding to the five pipeline stages (IF, ID, EXE, MEM, WB). I used a top-level DataPath module to connect these components and manage signal flow across stages.

The code begins by decoding the instruction fields from the fetched instruction (dinstOut) and assigning them to corresponding wires such as op, rs, rt, rd, funct, imm, and address. These fields are then passed into the ControlUnit to generate control signals like wreg, m2reg, wmem, and ALU-related controls.

In the Instruction Fetch stage, I included a ProgramCounter, a PCAdder to increment the PC, and a npc\_MUX to handle PC source selection (for branches, jumps, and sequential flow). The InstructionMemory fetches the instruction, and the IFID\_PipelineRegister stores it along with PC+4 for the next stage.

In the Decode stage, the Register\_File reads operands, and based on the regrt signal, a multiplexer (Regrt\_MUX) selects the destination register. The immediate field is extended using the Immediate\_Extender, and fwd\_a\_mux and fwd\_b\_mux handle operand forwarding to minimize data hazards.

The EXE stage contains ALU operand selection (alu\_a\_mux and alu\_b\_mux), execution via the ALU, and special handling for the jal instruction using PC\_EXE\_Adder, jal\_mux, and jal\_f. Outputs are passed through the EXE\_MEM\_Pipeline.

In the Memory stage, I used a DataMemory module to perform memory reads/writes controlled by mwmem. This data, along with control and result values, is stored in the MEM\_WB\_Pipeline.

Finally, the Write Back stage uses WB\_Mux to decide whether to write ALU results or memory data back into the register file.

All modules were integrated and instantiated in a logically ordered and well-commented manner, with additional wiring for hazard control, jump/branch logic, and pipeline stall management.

**Device: Zyboboard (XC7Z010- -1CLG400C)**

## Testbench

A screenshot of a computer

AI-generated content may be incorrect.

## Schematic Design

A computer screen shot of a computer

AI-generated content may be incorrect.

## Waveform

A screenshot of a computer

AI-generated content may be incorrect.

## Floorplanning

A screenshot of a computer

AI-generated content may be incorrect.

## IO Planning A screenshot of a computer game AI-generated content may be incorrect.