# Project 1: Report

## Single Cycle Implementation:

### File Name: Alu.v

- ➢ Encodes ALU
  - Has three inputs, A, B, and Control
  - Outputs result or zero
  - Has add and subtract function

### File Name: Alu_Control_Unit.v

- ➢ Immediate
  - Inputs an immediate value (integer)
  - Outputs Sign Immediate

### File Name: Control_Unit.v

- ➢ Sets opcode values
  - Outputs ALUOp, MemWrite, RegWrite, RegDst, MemtoReg, ALUSrc, Branch, and Jump
  - Inputs opcode

### File Name: Data_Memory.v

- ➢ Reads and writes to data memory
  - Inputs A, WD, WE, clk, rst_n
  - Outputs RD
  - Writes or reads to an address in memory

### File Name: Imm_Sign_Extend.v

- ➢ Assigns new sign extension
  - Inputs an immediate value (integer)
  - Outputs Sign Immediate

### File Name: Instruction_Memory.v

- ➢ Instruction of memory
  - Inputs A
  - Outputs RD
  - Read instruction of memory.

### File Name: MIP_Single_Cycle.v

- ➢ Single Cycle
  - Inputs A
  - Outputs RD
  - Contains all the wires for instructions

### File Name: PC.v

- PC counter
    - Inputs clk, rst_n, immediate, jump low 26 bits, PCSrc,
    - Outputs the zero from the ALU as input and PC for the output.

## File Name: Register_Files.v

- encloses all the registers that are independent of the processor which is to perform read and write operations which is done by getting the input and assigning whether its for reading or writing
    - Input A1,A2,A3,RegWrite,WD3
    - Output RD1,RD2
    - Sets RegWrite to 1 when lw Instr is executed

## File Name: tb_SingleCycle_.v

- Connects all the files and assigns dump files where gtkwave will use to form waves
    - Also controls the clock cycle

## File Name: Mips_wave.vcd

- Dump file for tb_SingleCycle

## File Name: memfile.dat

- Data input

# Pipeline:

## File Name: Alu.v

- ALU with pipeline support
  - Has inputs rst_n (system clock), SignImm, ALUControl, ALUSrc, ForwardAE, ForwardBE, ResultW, ALUOutM, RD1E, RD2E
  - Outputs SRCB_Forward, ALUOut, and Zero
  - Can forward data or stall

## File Name: ALU_Control_Unit.v

- ALU control unit
  - Has inputs Funct and ALUOp
  - Outputs ALUControl
  - Receives ALU opcode from control unit and converts it to ALU

## File Name: Branch_Unit.v

- Branch Unit
  - Has inputs BranchD, SignImm, PCPlus4D, RD1, RD2, ALUOutM, ForwardAD, and ForwardBD
  - Outputs PCBranchD and PCSrcD
  - Allows hardware to branch and jump to other lines of code

## File Name: Control_Unit.v

- Sets opcode values
  - Inputs opcode and rst_n (system clock)
  - Outputs ALUOp, MemWrite, RegWrite, RegDst, MemtoReg, ALUSrc, Branch, and Jump

## File Name: Data_Memory.v

- Reads and writes to data memory
  - Inputs A, WD, WE
  - Outputs RD
  - Writes or reads to an address in memory

## File Name: EX_MEM_Register.v

- Gives memory address to read and write to
  - Has inputs clk, rst_n, ALUOut, WriteDataE, WriteRegE, Zero, RegWriteE, MemtoRegE, MemWriteE
  - Outputs ALUOutM, WriteDataM, WriteRegM, ZeroM, RegWriteM, MemtoRegM, MemWriteM
  - Third stage of clock cycle to execute

## File Name: Forward_Unit.v

- Solve the data hazards
  - Input rst_n, RsE, RtE, RsD,RtD, WriteRegM, WriteRegW,RegWriteM,RegWriteW

- First, if hazard in EX/MEM.RegisterRd = ID/EX.RegisterRs then the first ALU Operand come from last ALU calculate result.
- Second, the hazard is happened in MEM/WB.RegisterRd = ID/EX.RegisterRt then he first ALU Operand come from last ALU calculate result or Memory data.
- Last, addi Hazard: EX.MEM.RegisterRd
- Output ForwardAE, ForwardBE, ForwardAD, ForwardBD

## File Name: ID_EX_Resgister

➢ Instruction decode and executive
- Input: clk, rst_n, RD1, RD2, Rt, Rs, Rd, SignImm, PCPlus4D, ALUControlD, FlushE, RegWriteD, MemtoRegD, MemWriteD, RegDstD, ALUSrcD,
- Output: RD1E, RD2E, RtE ,RsE, RdE ,SignImmE, PCPlus4E,RegWriteE, MemtoRegE, MemWriteE, RegDstE, ALUSrcE, ALUControlE
- the register numbers to read the two registers.

## File Name: IF_ID_Register.v

➢ Instruction Fetch and Instruction decode
- Input clk, rst_n, RD1, RD2, Rt, Rs, Rd, SignImm, PCPlus4D
- Output PCPlus4D,InstrD

## File Name: Imm_Sign_Extend.v

➢ Assigns new sign extension
- Input Immediate
- Output SignImm
- Sign Extend immediate

## File Name: Instruction_Memory.v

➢ Instruction of memory
- Inputs A
- Outputs RD
- Read instruction of memory.

## File Name: PC.v

➢ PC counter
- Inputs clk, rst_n, PCSrcD, StallF, PCBranchD, jumpD, InstrD_Low25Bit
- Outputs the zero from the ALU as input and PC for the output

## File Name: Register_Files.v

➢ encloses all the registers that are independent of the processor which is to perform read and write operations which is done by getting the input and assigning whether its for reading or writing
- Input A1, A2, A3, RegWrite, WD3
- Output RD1,RD2
- Sets RegWrite to 1 when lw Instr is executed

**File Name: MIPS_Pipeline.v**

- ➢ Contains all the wire and is what set everything for every instruction
  - Input: clock, rest_n

**File Name: tb_MIPS_Pipeline.v**

- ➢ Connects all the files and assigns dump files where gtkwave will use to form waves
  - Also controls the clock cycle

**File Name: Mips_wave.vcd**

- ➢ Dump file for tb_MIPS_Pipeline

**File Name: memfile.dat**

- ➢ Data input

**File Name: Stall_Unit.v**

- ➢ Hazard fix for Load Use Data and it initiates a stall
  - Input: clock, rest_n, RsD, RtD, RtE, WriteRegE, WriteRegM, RegWriteE, MemtoRegE, MemtoRegM, BranchD
  - Output: FlushE, StallD, StallF