Report

This report contains the description of working and implementation of 3-staged pipelined RISC-V processor. The 3 stages are as follow:

1. Instruction Fetch (IF)
2. Instruction Decode (ID) and Instruction Execute (EX)
3. Memory (MEM) and Write Back (WB)

The execution flow is explained as there are different modules, each implements a specific task. The modules include:

- ***ALU.sv*** Implements the Arithmetic and Logic Unit (ALU) responsible for executing arithmetic and logical operations.
- ***Add4.sv*** A module for adding a constant value of 4 to its input.
- ***BranchCondition.sv*** Determines the branch condition based on the comparison of two inputs.
- ***CSR_Registerfile.sv*** Implements the Control and Status Register (CSR) file for storing and managing control and status information.
- ***CSR_buffer.sv*** A buffer module for handling Control and Status Register (CSR) operations.
- ***Controller.sv*** Controls the overall execution flow of the processor by generating control signals.
- ***DataMemory.sv*** Represents the data memory module responsible for storing and retrieving data.
- ***ForwardingUnit.sv*** Manages data forwarding between pipeline stages to resolve data hazards.
- ***ImmediateGenerator.sv*** Generates immediate values for instructions.
- ***InstructionMemory.sv*** Manages the instruction memory and fetches instructions for the processor.
- ***Mux2.sv*** A simple 2-to-1 multiplexer module.
- ***PC.sv*** Controls the flow of instructions in the pipeline.
- ***PipelineControl.sv*** Implements instruction control during pipeline execution.
- ***PipelineRegister.sv*** Implements pipeline registers for storing intermediate values during pipeline execution.
- ***Pipeline_Branch.sv*** Handles branch instructions in the pipeline.
- ***Pipeline_IR.sv*** Manages the Instruction Register (IR) in the pipeline.
- ***Processor.sv*** Integrates all the modules to create the complete processor.
- ***RegisterFile.sv*** Implements the general-purpose register file for storing data.
- ***WriteBack.sv*** Manages the write-back stage of the pipeline.
- ***config.gtkw*** Configuration file for viewing simulation results using GTKWave.
- ***dmem_out.mem*** Memory file for data memory output.
- ***inst.mem*** Memory file containing instructions.
- ***inst_out.mem*** Memory file for instruction outputs.
- ***processor.vcd*** Value Change Dump (VCD) file for waveform viewing.
- ***rf.mem*** Memory file for register file data.
- ***rf_out.mem*** Memory file for register file output.
- ***tb_processor.sv*** Testbench for the processor.
- ***Transcript*** A log file containing the simulation transcript.

For the implementation of pipeline, I've created pipeline files for each case and then based on the requirements I've created their instance in the stages in ***Processor.sv*** file. The control is managed in separate file where as the values for registers, branch and instruction register as the instructions move in the pipeline are managed separately.