**Project Summary: RISC-V 3-Stage Pipelined Processor**

**Project Name**: RISC\_3\_Stage  
**Toolchain**: Xilinx Vivado 2024.2  
**Top-Level Module**: RISCV\_32I\_TOP.v

**Overview**

This project implements a **32-bit RISC-V (RV32I) processor** with a **3-stage pipeline** architecture. The design integrates separate instruction and data caches, control/status register (CSR) support, and simulation-driven trace logging for debugging and verification.

The pipeline stages are organized as follows:

* **Stage 1: Instruction Fetch (IF)**
  + Program Counter (pc.v)
  + Instruction Memory (I\_cache)
  + Pipeline Register Block 1 (reg\_block\_1)
* **Stage 2: Instruction Decode (ID)**
  + Instruction Decoder (msrv32\_decoder.v)
  + Immediate Generator (imm\_generator.v)
  + Immediate Adder (imm\_adder.v)
  + Branch Unit (branch\_unit.v)
  + Integer Register File (msrv32\_integer\_file.v)
  + Control Unit (machine\_control.v)
  + CSR File (msrv32\_csr\_file.v)
  + Write Enable Generator (write\_enable\_gen.v)
  + Pipeline Register Block 2 (reg\_block\_2)
* **Stage 3: Execute / Memory / Writeback (EX/WB)**
  + Arithmetic Logic Unit (ALU.v)
  + Load Unit (load\_unit.v)
  + Store Unit (store\_unit.v)
  + Writeback Mux Unit (wb\_mux\_sel\_unit.v)
  + Data Memory (D\_cache)

**Key Features**

* **RV32I ISA Compliance**: Supports base integer instruction set.
* **3-Stage Pipeline**: Efficient instruction throughput with IF, ID, and EX/WB stages.
* **CSR Integration**: Provides system-level control and status functionality.
* **Branch and Immediate Handling**: Dedicated branch unit and immediate adder.
* **Modular Memory Design**: Separate instruction (I\_cache) and data (D\_cache) memories.
* **Trace and Debug Support**:
  + fwrite\_file1.v module logs PC values, register writes, and memory operations to result\_status.txt during simulation.
  + Facilitates step-by-step analysis of processor state.

**Applications**

This design serves as a platform for:

* Studying pipeline behavior in RISC-V architectures.
* Developing and testing compiler-generated RISC-V programs.
* Extending with custom instructions for application-specific accelerators (e.g., AI/ML, DSP).
* Educational and research purposes in computer architecture and VLSI design.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

In my zip file I have kept Assembly instruction file ,memory file and result status file.



this file is loaded in Icache module and Dcase module

