**Project Plan: RISC-V 5-Stage Pipelined CPU and Verification**

**Assumptions:**

* 9 hours/week.
* Total project time: ~30 weeks (9 hrs × 30 weeks = 270 hrs).
* Work divided into sequential one-week sprints. Each sprint has a primary objective and deliverables.

**Phase 1: Foundations (Weeks 1–3)**

**Sprint 1 (9 hrs): Project Setup**

* Install and configure toolchain: Questa Sim, Vivado, UVM library.
* Set up Git repository and directory structure (rtl, tb, docs, scripts, fpga).
* Write initial project specification draft.
* Deliverable: Baseline spec document and working simulation environment.

**Sprint 2 (9 hrs): RV32I ISA & Pipeline Architecture Definition**

* Define RV32I subset scope (RV32I base ISA only, no extensions).
* Draft 5-stage pipeline block diagram (IF, ID, EX, MEM, WB).
* Define instruction formats, control signals, and hazards to be handled.
* Deliverable: Architecture diagram and design notes.

**Sprint 3 (9 hrs): Verification Strategy Planning**

* Plan UVM environment architecture: testbench hierarchy, drivers, monitors, scoreboard.
* Define test categories: directed unit tests, random instruction streams, compliance suite.
* Create test list mapping to ISA requirements.
* Deliverable: Verification plan document.

**Phase 2: RTL Development (Weeks 4–11)**

**Sprint 4 (9 hrs): Instruction Fetch (IF) Stage**

* Implement PC, instruction memory interface, sequential fetch.
* Create simple SystemVerilog testbench (not UVM yet) to validate IF.
* Deliverable: IF stage RTL validated in isolation.

**Sprint 5 (9 hrs): Instruction Decode (ID) Stage**

* Implement instruction decoder, register file read.
* Integrate IF + ID.
* Write directed tests for decoding correctness.
* Deliverable: IF/ID RTL integrated and tested.

**Sprint 6 (9 hrs): Execute (EX) Stage**

* Implement ALU supporting all RV32I operations.
* Define forwarding and hazard detection stubs.
* Integrate EX with IF/ID.
* Deliverable: IF/ID/EX functional in basic tests.

**Sprint 7 (9 hrs): Memory (MEM) Stage**

* Implement data memory interface for loads/stores.
* Handle alignment, byte/halfword/word accesses.
* Integrate MEM into pipeline.
* Deliverable: IF/ID/EX/MEM path operational.

**Sprint 8 (9 hrs): Writeback (WB) Stage**

* Implement register file writeback logic.
* Validate correct propagation of results through pipeline.
* Deliverable: Full 5-stage path assembled.

**Sprint 9 (9 hrs): Pipeline Control (Stalls & Forwarding)**

* Implement hazard detection unit.
* Implement forwarding paths for EX/MEM/WB.
* Validate with directed hazard tests.
* Deliverable: Hazard-free execution of sequential programs.

**Sprint 10 (9 hrs): Branch & Jump Control**

* Implement branch comparator and PC update logic.
* Add branch/jump flush mechanism.
* Validate with branch-heavy programs.
* Deliverable: Control flow correctness established.

**Sprint 11 (9 hrs): RTL Integration Testing**

* Write small hand-coded programs to test end-to-end pipeline.
* Validate correctness under typical instruction mixes.
* Deliverable: Initial functional CPU RTL.

**Phase 3: UVM Verification (Weeks 12–20)**

**Sprint 12 (9 hrs): UVM Environment Skeleton**

* Implement UVM env, agent, sequencer, driver.
* Connect DUT interfaces to driver/monitor.
* Deliverable: Compilable UVM testbench skeleton.

**Sprint 13 (9 hrs): UVM Monitor & Scoreboard**

* Implement instruction monitor capturing executed instructions.
* Implement scoreboard for architectural state checking.
* Deliverable: End-to-end UVM checking infrastructure.

**Sprint 14 (9 hrs): Directed Test Sequences**

* Implement UVM sequences for directed testing (ALU ops, load/store, branches).
* Deliverable: Regression suite of directed tests.

**Sprint 15 (9 hrs): Randomized Test Sequences**

* Add constrained-random instruction generators.
* Begin coverage model definition.
* Deliverable: Randomized instruction stream tests with coverage bins.

**Sprint 16 (9 hrs): Compliance Suite Integration**

* Download and integrate official RISC-V compliance tests.
* Write run scripts to automate test execution.
* Deliverable: Compliance regression harness.

**Sprint 17 (9 hrs): Debug & Fix Non-Compliance Issues**

* Run compliance tests, analyze failing cases.
* Debug pipeline control, forwarding, or corner cases.
* Deliverable: Passing rate improved, bugs eliminated.

**Sprint 18 (9 hrs): Coverage Closure Effort**

* Use Questa Sim coverage reports.
* Add missing directed tests for uncovered instructions/paths.
* Deliverable: Coverage > 95%.

**Sprint 19 (9 hrs): Regression Stabilization**

* Ensure 100% compliance suite pass.
* Freeze RTL baseline.
* Deliverable: Verified CPU RTL and stable UVM environment.

**Sprint 20 (9 hrs): Automation Scripts**

* Write shell or Python scripts to run all tests or subsets.
* Document usage.
* Deliverable: Turnkey test execution scripts.

**Phase 4: FPGA Prototyping (Weeks 21–24)**

**Sprint 21 (9 hrs): FPGA Build Setup**

* Create Vivado project.
* Write memory initialization procedure.
* Deliverable: Baseline synthesis run.

**Sprint 22 (9 hrs): FPGA Constraints & Integration**

* Add clock/reset constraints.
* Map instruction/data memories to BRAMs.
* Deliverable: Post-synthesis simulation functional.

**Sprint 23 (9 hrs): FPGA Bitstream Generation**

* Synthesize and implement pipeline on FPGA board.
* Load simple test program (add, branch).
* Deliverable: Running CPU on hardware.

**Sprint 24 (9 hrs): FPGA Demo Programs**

* Run small assembly programs: loops, memory tests.
* Verify outputs through on-chip debugging or UART.
* Deliverable: Demonstration of functional FPGA prototype.

**Phase 5: Documentation and Presentation (Weeks 25–30)**

**Sprint 25 (9 hrs): Design Documentation Drafting**

* Write RTL design description.
* Include block diagrams for pipeline stages.
* Deliverable: Initial draft of capstone report.

**Sprint 26 (9 hrs): Verification Documentation**

* Document UVM environment architecture.
* Add diagrams of agents, monitors, scoreboard.
* Deliverable: Verification methodology section.

**Sprint 27 (9 hrs): FPGA Documentation**

* Describe synthesis flow and FPGA test programs.
* Deliverable: Prototyping chapter in report.

**Sprint 28 (9 hrs): Report Integration**

* Merge drafts into cohesive final document.
* Add references, appendices, compliance coverage results.
* Deliverable: Complete technical report.

**Sprint 29 (9 hrs): Poster Creation**

* Create academic poster with diagrams, pipeline summary, verification results.
* Deliverable: Capstone poster draft.

**Sprint 30 (9 hrs): Final Polishing**

* Revise report and poster.
* Ensure reproducibility of simulation and FPGA build.
* Deliverable: Final report and poster ready for submission.

**Total Duration:** 30 weeks  
**Total Effort:** 270 hrs  
**Final Deliverables:**

* Verified RV32I 5-stage pipelined CPU RTL in SystemVerilog.
* UVM-based verification environment with 100% compliance.
* FPGA prototype executing simple programs.
* Report with diagrams and methodology.
* Presentation poster.