RV College of Engineering®
Mysore Road, RV Vidyaniketan Post,
Bengaluru - 560059, Karnataka, India

NBA Accredited (UG - 6 Years)

Department of Electronics & Communication Engineering

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

COMPUTER ARCHITECTURE

"RISC V Instruction Front End Buffering Unit"

Submitted by

R Chithra
1RV23EC112

Poornachandra B R
1RV23EC095

Academic Year 2025 - 26

1.Objectives

The primary objective of this project is to design and verify a RISC-V Instruction Front-End Buffering Unit that effectively manages the flow of instructions between the decode stage and the execution stage.
The design aims to improve instruction throughput, prevent execution stalls, preserve FIFO ordering, and ensure correct handling of short and long instructions. Additionally, the project focuses on validating the design through simulation, output analysis, and waveform inspection.

---

2. Methodology

The project follows a modular and verification-driven methodology. First, the algorithm for a Level-2 Instruction Front-End Buffering Unit was defined, incorporating execution look-ahead and bypass optimization. Based on this algorithm, the RTL was implemented using SystemVerilog. A structured testbench was developed to apply controlled stimulus, while a monitor module was used to check design invariants and ensure correctness. Simulation outputs were analyzed using both textual output tables and waveform visualization to validate functional behavior under multiple operating scenarios.

---

3. Software Tools Used

- SystemVerilog – Used for RTL design, testbench, and monitor implementation
- Icarus Verilog (iverilog) – Used for compiling and simulating the design
- GTKWave – Used for viewing and analyzing simulation waveforms
- VS Code – Used for code editing and result documentation
- GitHub – Used for version control, documentation, and public project hosting

---

4. Results

The simulation results confirm that the Instruction Front-End Buffering Unit operates correctly across all tested scenarios.
The buffer successfully handles bypass, push, and pop operations, maintains correct buffer count updates at the positive edge of the clock, enforces stall conditions when full, and avoids execution bubbles through look-ahead logic.
The output table and GTKWave waveform clearly demonstrate correct timing behavior, FIFO ordering, and stable state transitions, validating the functional correctness of the design.

---

5. Future Scope

This work can be extended by integrating the Instruction Front-End Buffering Unit with a renaming unit developed by another team to support more advanced front-end pipeline functionality.
Further, the combined design can be implemented and tested on hardware platforms such as FPGA, enabling real-time validation and performance evaluation.
Additional enhancements may include multi-issue buffering, speculative execution support, and tighter integration with a complete RISC-V processor pipeline.