JoSDC'24

المسابقة الوطنية لتصميم الشرائح الإلكترونية

Jordan National Semiconductors Design Competition

Innovation Phase

Guidelines and Rubric

Dr. Dheya Mustafa

Abdalrahman Alshannaq
Abdalrahman Ebdah

1. Overview

This document outlines the requirements and guidelines for the third phase of the JoSDC'24 Competition. The three phases are: (i) Qualifying (ii) Development and (iii) Innovation

1. **Qualifying Phase:** Teams focus on verification and debugging. The goal is to demonstrate a basic working design that can execute a minimal set of MIPS instructions.
2. **Development Phase**: The second phase involves expanding and refining the design. Teams should focus on supporting an extended instruction set, handling hazards, and introducing a pipelined architecture to improve performance.
3. **Innovate Phase**: The final stage emphasizes creative improvements and optimizations. Teams can propose and implement advanced features to enhance performance, efficiency, or functionality.

2. Innovation Phase Details

Welcome to the Innovation Phase, the final and most exciting stage of the competition. This phase is your opportunity to break boundaries, push limits, and explore innovative ideas that redefine performance, verification, and functionality. Here, creativity meets technical expertise, and your goal is to showcase the best of both.

Unlike earlier phases with structured tasks, the Innovation Phase invites you to go beyond basic requirements, think outside the box, and implement solutions that truly set your design apart. Whether it's enhancing speed, improving testing efficiency, or introducing new features, this is your chance to demonstrate originality and technical brilliance.

This document provides an overview of the key focus areas, suggests potential enhancements, and outlines the guidelines for submitting and presenting your ideas effectively. Let your innovation lead the way!

2.1 Performance

"Performance speaks – only the fastest will rise to the top!"

The primary goal of this phase is to achieve the highest performance results. This is your opportunity to push your design beyond its limits, focusing on speed, efficiency, and innovation. Your starting point must be a **functional pipelined processor**. From there, your challenge is to implement advanced techniques and optimizations that showcase your creativity and technical skills. Performance is the key metric here, and the fastest, most efficient designs will be rewarded.

Suggested Enhancements:

- Branch Prediction
  - Static or dynamic methods to reduce branch stalls.
- VLIW (Very Long Instruction Word)
  - Bundle multiple instructions to execute simultaneously.
- Superscalar (In-Order or Out-of-Order)
  - Implement parallel execution of multiple instructions in the same cycle.
- Out-of-Order Execution
  - Dynamically reorder instructions to minimize stalls and enhance ILP (Instruction Level Parallelism).
- Multicore
  - Shared memory multiprocessor, SIMD architecture

Important Notes:

- ➢ Software Enhancements are **allowed** for tools like schedulers or performance simulators or compilers.
- ➢ Cache implementation is **not allowed** for all teams. This decision ensures uniformity across teams, given that the current instruction and data memory on the FPGA can be accessed in a single cycle (on chip memory), acting as a cache. Additionally, the limited memory size on the FPGA makes cache implementation less practical.
- ➢ Refer to Organizing Team **Exploration Topics** Uploaded on MS Teams for more guidance and Concept Highlights.

- If you have a unique performance improvement idea, submit a **proposal** to the committee using the following Google Form. Once accepted, you may begin your implementation.

  https://forms.gle/54qcJPqr54xXT7qB7

2.2 Verification

This section focuses on improving the verification process, reducing debugging time, and ensuring design correctness, noting that cycle accurate simulator from previous phase is required to start working in this section.

Suggested Enhancements:

- Automated Testing Environment
  - Create a framework to automate test cases and compare results.
- SystemVerilog Integration
  - Utilize advanced verification tools and techniques with SV.
- Design for Testability (DFT)
  - Implement techniques to simplify testing and debugging.

Important Notes:

- Refer to Organizing Team **Exploration Topics** Uploaded on MS Teams for more guidance and Concept Highlights.
- Refer to **Innovation Phase Benchmarks** folder uploaded on MS Teams, where you can find a large number of benchmarks, make sure to **test all provided benchmarks OR as much as possible** from the folder, to get higher accuracy and to help committee evaluate your design.
- Ensure you have an efficient and effective testing strategy in place, allowing you to test a large set of benchmarks and test cases with minimal time and effort. This will help you achieve better results in your testing and improve your committee evaluation.
- If you would like to contribute by adding your benchmark(s) to the set of posted benchmarks, your participation is welcomed. Follow this link to submit and upload your benchmark: https://forms.gle/NXPVMauoHSQsmmch7

2.3 Standalone Enhancements

Standalone enhancements are additional features that improve usability or presentation without drastically affecting the core design.

Examples:

- VGA Integration
  - Display execution flow, results, or performance metrics visually for verification or presentation purposes.
- Exception Handler
  - Handle and manage exceptions efficiently for robustness and flexibility.

Important Notes:

- Floating Point is **not allowed**, and there is no benchmark will measure floating point execution.
- If you have a standalone enhancement idea, submit it via the Google Form for committee review and approval.

  Google form link : https://forms.gle/pwyYdGUjsUuGMzdv5

2.4 Guidelines for All Enhancements

- Clearly document all results, including performance metrics and testing outcomes.
- Provide detailed steps for reproducing your work.
- If your design encounters bugs during implementation, document them thoroughly and explain the resolution process.

- Collaborate with your team and refer to mentors via Meetings Form for any explanation and assistant.

Good luck! We are excited to see your innovative designs and enhancements in this final phase.

4. Conclusion and Phase Submission

At the end of the Innovation phase, teams are required to submit the following:

1. **Full Quartus Project Folder for Enhanced Processor**

Submit your complete project folder, including assembly code files and machine code files for any benchmarks used, organized in a subdirectory named “bench” with a proper "readme" file. The project should be free of errors and fully functional, showcasing your innovations and enhancements.

2. **Full Report Following the Provided Template**

The report should include detailed explanations of your design, improvements, and testing. This should cover:

- Design Explanations: Clarify your design choices for any advanced enhancements (e.g., branch prediction, VLIW, superscalar, etc.).
- Innovations and Improvements: Explain the modifications you made during the phase to improve performance, efficiency, and functionality.
- Testing Plan and Results: Outline your testing strategy, flow, and the outcomes from testing your enhancements, including performance analysis.
- Performance Comparison: Compare the results of your innovations against the baseline pipelined design and single cycle implemented previously.
- Signal Analysis and Relevant Documentation: Include any necessary technical details.
- Executive Summary (max 5 pages, font size 11pts): Summarize the core findings, conclusions, and key recommendations for a quick evaluation by the committee.

3. **Additional Materials**

You may submit supplementary materials such as:

- Supplementary diagrams to better explain your Datapath or control unit.
- Additional benchmarks or test cases that you applied to validate your design.
- Software Tools and implementations.

Submission deadline: February 2, 2025