Report — AutoChip

Student: Naveen Kumar Senthil Kumar

Course/Lab: AutoChip Tutorial

Date: September 18, 2025

1) Introduction

This detailed report presents the configuration, execution, and analysis of the AutoChip Tutorial ETS 2025. The assignment explored the integration of GenAI tools into hardware design workflows, focusing on the automatic generation of Verilog RTL modules, simulation with Icarus Verilog, and iterative refinement based on testbench feedback. The provided configuration file was used to set up and control the experiment pipeline.

2) Configuration File Details

The following key parameters were extracted from the provided configuration file:

- Prompt File: ./tokenbucket.v
- Module Name: top_module
- Testbench File: ./tokenbucket_tb.v
- Model Family: ChatGPT
- Model ID: gpt-4o-mini
- Number of Candidates per Iteration: 5
- Maximum Iterations: 5
- Output Directory: outputs_tokenbucket
- Log File: log.txt
- Ensemble Mode: False

The configuration highlights a single-model setup using ChatGPT (`gpt-4o-mini`) with a limit of five iterations. Each iteration generates five candidate Verilog designs, which are compiled and simulated against the provided testbench.

3) Workflow & Execution

The AutoChip loop executed the following sequence:

1. Initialization: Imported dependencies (`openai`, `tiktoken`, `iverilog`) and set up the environment in Colab.
2. Conversation Setup: System prompt defined AutoChip as a Verilog autocomplete engine.
3. Input Prompt: The design specification was read from the prompt file (`tokenbucket.v`).
4. Code Generation: At each iteration, multiple candidate designs were produced by the LLM.
5. Compilation: Each design was compiled using Icarus Verilog. Errors/warnings were captured in logs.
6. Simulation: Executed with the provided testbench (`tokenbucket_tb.v`) to measure correctness.
7. Ranking: Each response was ranked based on mismatches, compilation results, and code length.
8. Iteration Loop: Errors triggered refinement until a correct (rank 1.0) solution was found or iterations ended.
9. Output Logging: All iteration logs were stored in `outputs_tokenbucket/log.txt` with feedback and costs.

4) Results

The token bucket regulator design converged successfully. Iterations began with a set of candidate modules, where some produced simulation mismatches or compilation warnings. Through refinement, the best-ranked design achieved **zero mismatches** when compared to the testbench, confirming correctness.

Key results include:

- Consistent improvement in ranks across iterations.
- Multiple candidates achieving rank 1.0 in final iterations.
- Negligible cost per iteration (fractions of a cent per run).
- Logs capturing token usage, costs, and Verilog corrections.

5) Analysis

The experiment demonstrated how GenAI can be integrated with traditional simulation-based verification. For simple designs like the token bucket, convergence was achieved within a few iterations. The structured configuration ensured reproducibility, while the testbench-driven validation guaranteed functional correctness.

Compared to manual RTL design, the AutoChip loop accelerated prototype generation and correction. This approach is scalable to more complex circuits, though sequential FSM-based modules may require more iterations.

6) Learnings

- Structured Prompts: A well-defined prompt greatly improves LLM output quality.
- Testbench Role: Verification is critical; errors were only detectable through simulation.
- Iteration Strategy: Allowing multiple candidates increases the likelihood of convergence.
- Cost Efficiency: Even across multiple iterations, total computational cost was minimal.
- Reproducibility: The JSON config ensures that experiments can be rerun consistently.

7) Conclusion

The AutoChip assignment with the provided configuration file successfully demonstrated AI-driven RTL design. By leveraging `gpt-4o-mini`, iterative generation, and simulation feedback, the system converged to a correct Verilog implementation of the token bucket regulator. This workflow highlights the future potential of combining GenAI with EDA tools for accelerated chip design and verification.