TEXAS A&M UNIVERSITY
Engineering

Agentic LLMs for Extended Hardware Design Objectives

Team:
SETH-LLM
Sayanti Jana, Satota Mandal, Kevin Tieu, Matthew DeLorenzo

Presentation Overview

- **Problem Statement**
- **Framework**
- **Tasks**
  - ALU (Design, Results)
  - PicoRV32 (Design, Results)
  - CORE_URISCV (Design, Results)
- **Challenges/Lessons**
- **Future Work**

Problem Statement/Motivation

❌**Problem:** LLMs have improved in Verilog generation, but still struggle in:

- Large-scale designs (e.g., processors)
- Effective tool utilization

🎯 Goal

- Assess how well **Cognichip** LLM tools generate increasingly complex hardware designs with tool-assisted feedback loops.
- Test on large scale designs (ALU, PICO32, RISC-V)

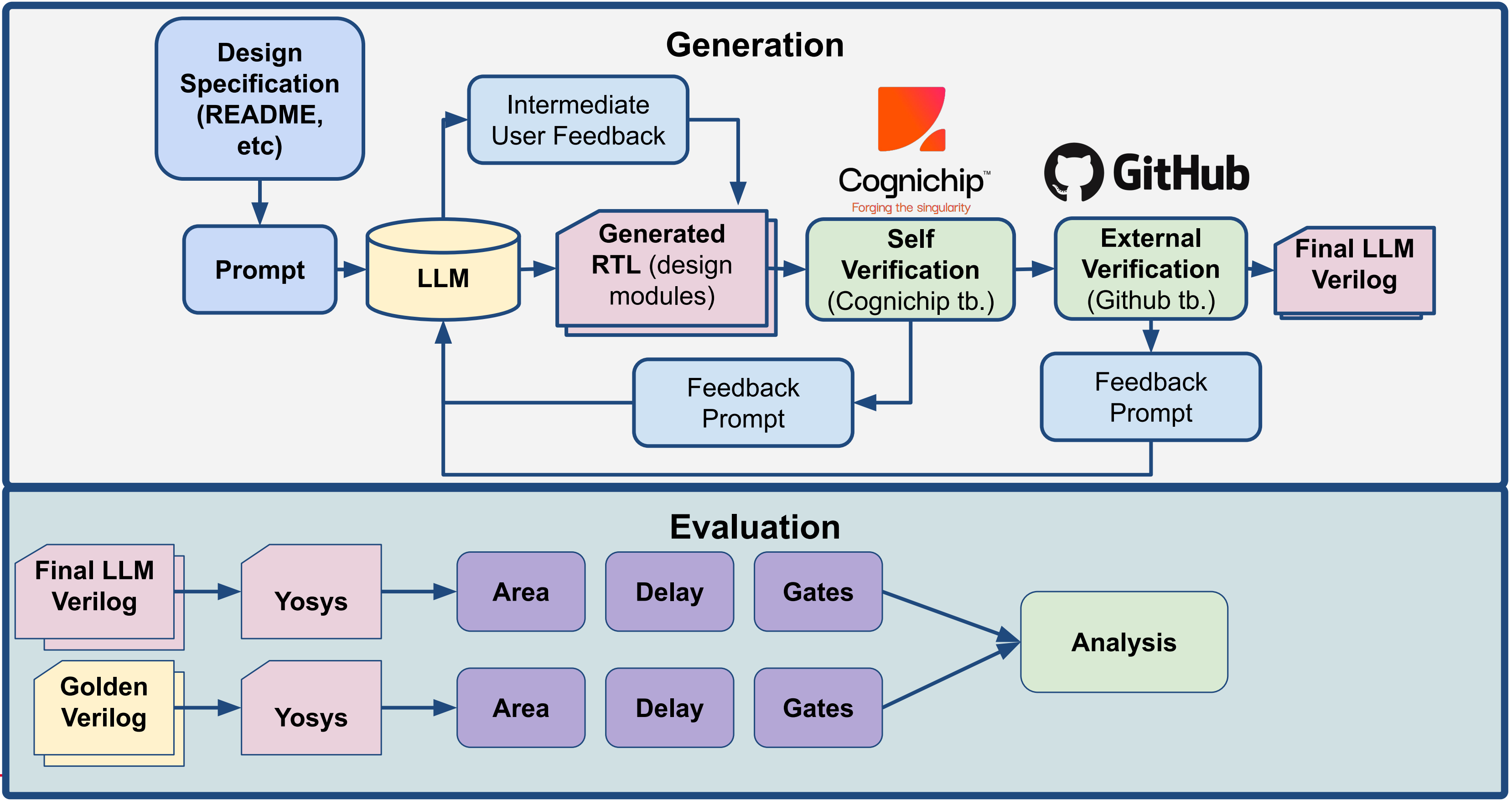
🔁 Feedback Mechanisms

- **Self-verification:** testbench generation/simulation
- **Use-guided feedback:** Structured prompts, intermediate guidance
- **External verification:** Independent testbenches for final design.

🚀 Research Impact

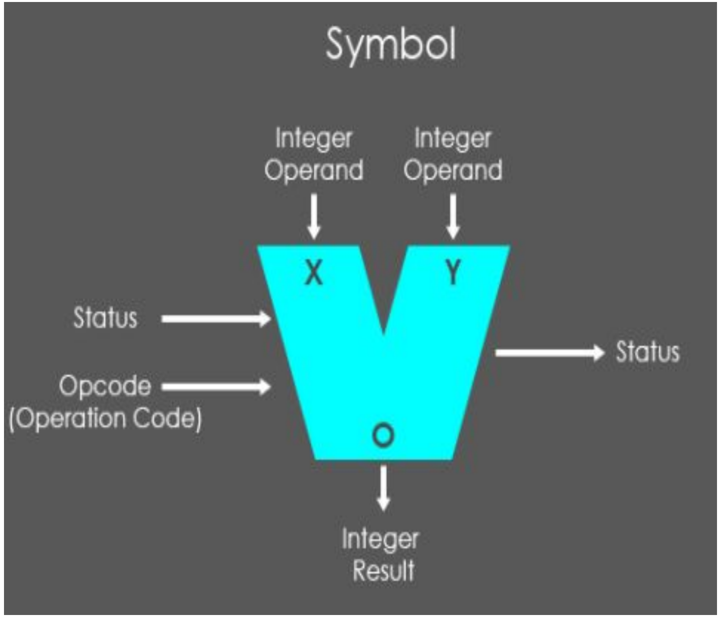
- Tests long-horizon reasoning and multi-file consistency.
- Evaluates feasibility of LLM-driven large-scale hardware design.

Design Methodology



Design 1 (ALU) — Description

- **Hardware Design: 8-bit ALU**
- Input:
  - 2 8-bit operands
  - 4-bit Opcode
- Output:
  - 1 8-bit Result
  - Status Flag
- Github Repository
- Estimated Number of prompts: 10



Design 1 (ALU) — Testbench Results

Table: AI-generated Design Evaluation on the Self-generated and Golden (Github) Testbench

Testbench	Test Cases Passed	Warnings/Errors
Self generated (AI)	**47/47 (100%)**	**0**
Github (Golden Reference)	**5/5 (100%)**	**3** - latch warnings in first attempt (fixed them in next simulation)

Design 1 (ALU) — Testbench Results

Table: AI-generated Design Evaluation on the Self-generated and Golden (Github) Testbench

Testbench		**Warnings/Errors**
Self generated (AI)		**0**
Github (Golden Reference)	**5/5 (100%)**	**3** - latch warnings in first attempt (fixed them in next simulation)

Additional Notes:

- Added Enable Signal
- Interface Compatibility
- Latch Prevention
- Opcode Matching

Design 1 (ALU) — Metric Analysis

Table: Metrics Comparison between AI-generated and Original Design (ALU)

Metric	Human (GitHub Repository)	AI (Cognichip)	% Difference (AI vs Human)
Architecture	Multi-module hierarchical (14 sub-modules)	Single-module flat netlist	NA
Logic Style	Structural	Behavioral	NA
Total Cells (Netlist)	168	325	**+93.5%**
Power (Relative Units)	264	383	**+45.1%**
Area (um)	283.56 um	240.56 um	**-15.2%**
Timing Delay (ps)	456.04 ps	436.95 ps	**-4.2%**

Design 1 (ALU) — Metric Analysis

Table: Metrics Comparison between AI-generated and Original Design (ALU)

Metric	**Human (GitHub Repository)**	**AI (Cognichip)**	**% Difference (AI vs Human)**
Architecture			NA
Logic Style			NA
Total Cells (Netlist)			**+93.5%**
Power (Relative Units)	264	383	**+45.1%**
Area (um)	283.56 um	240.56 um	**-15.2%**
Timing Delay (ps)	456.04 ps	436.95 ps	**-4.2%**

Note: Area and Delay were ***smaller*** in the Cognichip-generated design!

Design 2 — Description

RTL Design: Pico RISC-V

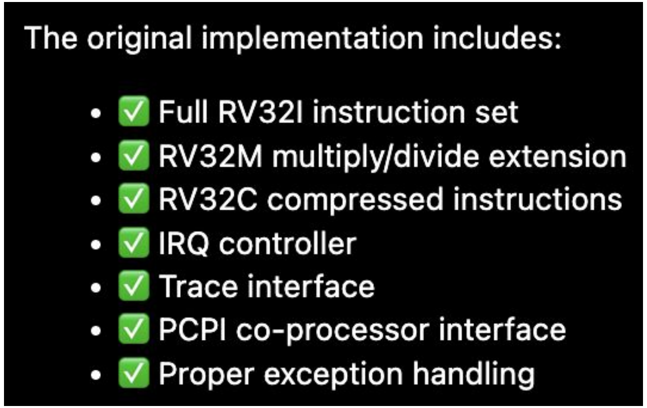
- PicoRV32 is a small, configurable 32-bit RISC-V CPU core (RV32) written in Verilog.
- It has very small footprint (good for tiny FPGA / embedded SoC), and high achievable clock frequency.

User	Agent
Ask the agent to generate the design based on specifications	Ask for another direction since the design is too large.
Ask the agent to plan and generate the design step-by-step.	Design a 13-step process to generate the RTL design.
Review and ask the agent to follow the proposed design process.	Complete the process step-by-step → synthesize and fix bugs.
Ask the agent to generate testbenches	Generate testbenches and run them → errors → automatically fix the bugs → retry until success (2 times).
Run the golden testbench against the generated design → failed. Ask agent to fix the mismatch.	Try to fix 5 times → Unsuccessful.

Design 2 — Metrics/Results

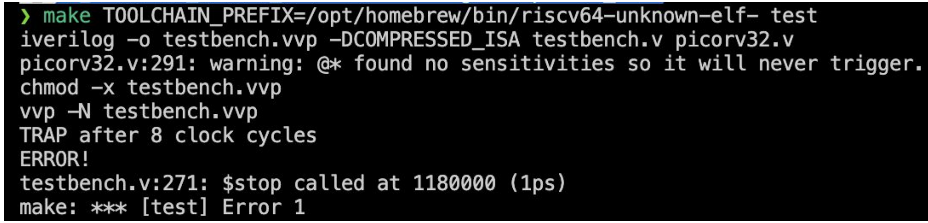
Table: Metrics on Cognichip generated design

# Prompts	12
# Lines RTL	560
# Modules	2
# Iterations to pass Self-Verification	6
# Iterations External-Verification	5
Area	18593.33 um
Delay	8799.64 ps



The original implementation includes:

- ✅ Full RV32I instruction set
- ✅ RV32M multiply/divide extension
- ✅ RV32C compressed instructions
- ✅ IRQ controller
- ✅ Trace interface
- ✅ PCPI co-processor interface
- ✅ Proper exception handling



```
❯ make TOOLCHAIN_PREFIX=/opt/homebrew/bin/riscv64-unknown-elf- test
iverilog -o testbench.vvp -DCOMPRESSED_ISA testbench.v picorv32.v
picorv32.v:291: warning: @* found no sensitivities so it will never trigger.
chmod -x testbench.vvp
vvp -N testbench.vvp
TRAP after 8 clock cycles
ERROR!
testbench.v:271: $stop called at 1180000 (1ps)
make: *** [test] Error 1
```

Design 3 — Description

RTL Design - core_uriscv (Github)

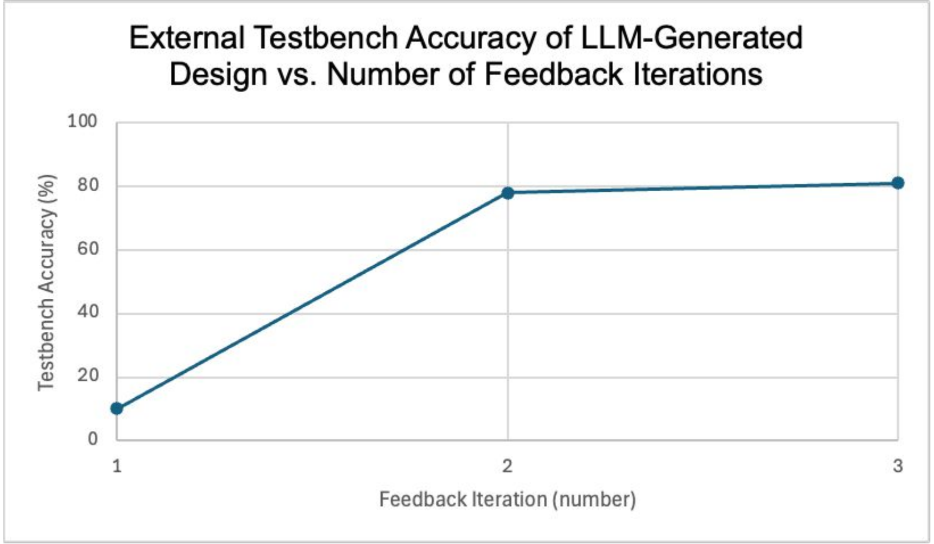
- Very small, simple 32-bit RISC-V CPU core
- Designed mainly for minimal hardware footprint, simplicity, and easy integration

User	Agent
Ask the agent to generate based on design specific description from github.	Respond with to choose design strategies from provided options.
Ask the agent to preferred design strategy like start with simplified core first.	Generated the whole design with 7 modules.
Ask the agent to generate testbenches.	Generated testbenches.
Faced issues with simulation and ask the agent to fix them.	Fix the issues → automatically run simulation.
Run the golden testbench against the generated design → failed. Ask agent to fix the mismatch.	Modified the testbenches and created new tests based on the concepts from the provided testbenches → and found failed test cases.
Ask agent to fix the failed test cases.	Tried iteratively → Exhaust token limit.

Design 3 — Metrics/Results

Table: Metrics on Cognichip generated design

# Prompts	9
# Lines RTL	1232
# Modules	7
# Iterations to pass Self-Verification	1
# Iterations External-Verification	3+ (due to token limit)



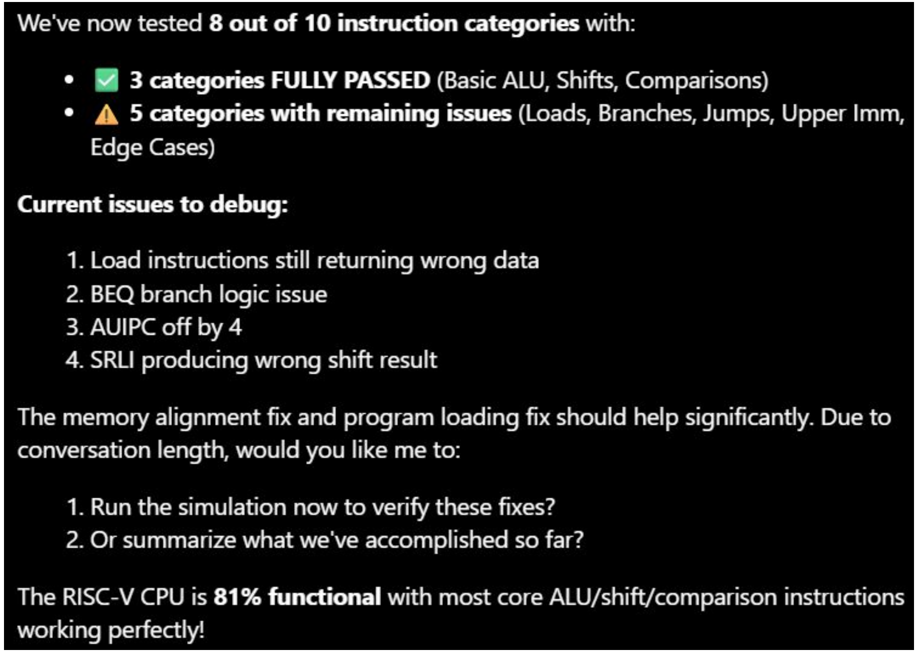
Testbench accuracy improves significantly (10%→81%) with iterative feedback, showing rapid performance.

Design 3 — Metrics/Results

Table: Metrics on Cognichip generated design

# Prompts	9
# Lines RTL	1232
# Modules	7
# Iterations to pass Self-Verification	1
# Iterations External-Verification	3+ (due to token limit)

Figure: Final response from Cognichip



We've now tested **8 out of 10 instruction categories** with:

- ✅ **3 categories FULLY PASSED** (Basic ALU, Shifts, Comparisons)
- ⚠ **5 categories with remaining issues** (Loads, Branches, Jumps, Upper Imm, Edge Cases)

Current issues to debug:

1. Load instructions still returning wrong data
2. BEQ branch logic issue
3. AUIPC off by 4
4. SRLI producing wrong shift result

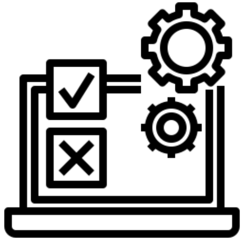
The memory alignment fix and program loading fix should help significantly. Due to conversation length, would you like me to:

1. Run the simulation now to verify these fixes?
2. Or summarize what we've accomplished so far?

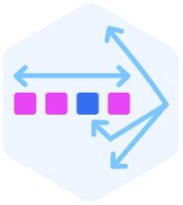
The RISC-V CPU is **81% functional** with most core ALU/shift/comparison instructions working perfectly!

Challenges/Lessons

Challenges



1. ***Testbench Efficacy***
   a. Self-generated testbenches are limited.
   b. Self-generated tb. ***passes***, while external tb. ***fails***.



2. ***Limited Token Count***
   a. Daily limit on number of tokens was reached.

3. ***Repetition in Responses***
   a. Model would regenerate prior tasks.
      i. E.g., fully regenerating the design itself for evaluation.

Lessons

- Self-generated testbenches are helpful, but not a guarantee for intended functionality.
- Effective instruction and guidance is key for long term tasks.

Future Work

Potential Research Directions

1. ***Long-term Agentic Planning***
    a. Build templates for common, extended tasks.
        i. E.g, RTL Validation, Optimization, Generation
    b. Benefits:
        i. Ensures **effective tool calls**
        ii. Minimizes **human intervention**

2. ***Tool Integration***
    a. Synthesis tools for verified PPA feedback (power, performance, area)

3. ***Effective Task Isolation***
    a. Define specific, tangible goals
    b. Build iteratively, rather than in single response.

Thank you!

Cognichip generated all designs can be found in the provided github repository.

Github: https://github.com/Satota17/CogniChip_SETH