2025 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION CHINA

accellera SYSTEMS INITIATIVE

Automated SVA Generation with LLMs

Lik Tung Fu, School of Integrate Circuits, Southeast University, Nanjing, China, (*liktungfu@seu.edu.cn*)

Shaokai Ren, National Center of Technology Innovation for Electronic Design Automation, Nanjing, China, (*renshaokai@nctieda.com*)

Mengli Zhang, National Center of Technology Innovation for Electronic Design Automation, Nanjing, China, (*zhangmengli@nctieda.com*)

Sichao Yang, X-EPIC Corporation Limited, Shanghai, China, (*sichaoy@x-epic.com*)

Ruiming Zeng, X-EPIC Corporation Limited, Shanghai, China, (*raymondz@x-epic.com*)

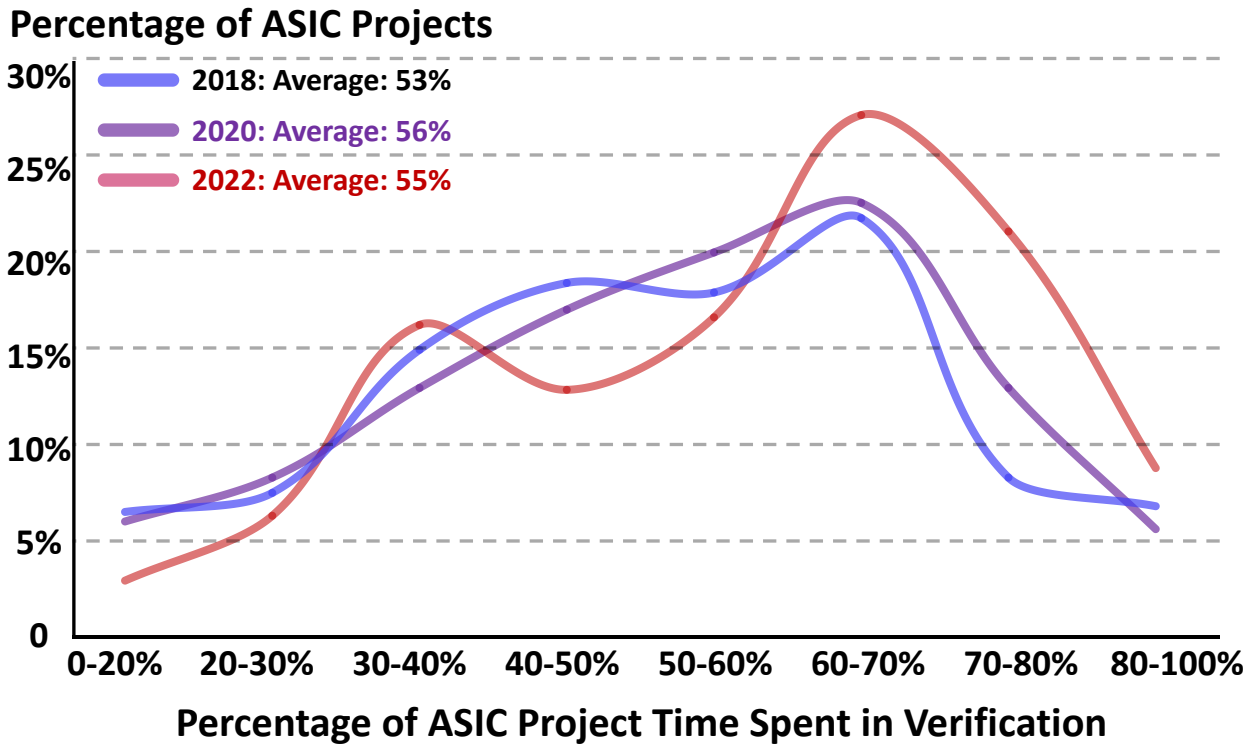
Zhe Jiang, School of Integrate Circuits, Southeast University, Nanjing, China, (*101013615@seu.edu.cn*)

Xi Wang, School of Integrate Circuits, Southeast University, Nanjing, China, (*xi.wang@seu.edu.cn*)

Jun Yang, School of Integrate Circuits, Southeast University, Nanjing, China, (*dragon@seu.edu.cn*)

16 Apr 2025 上海 SHANGHAI

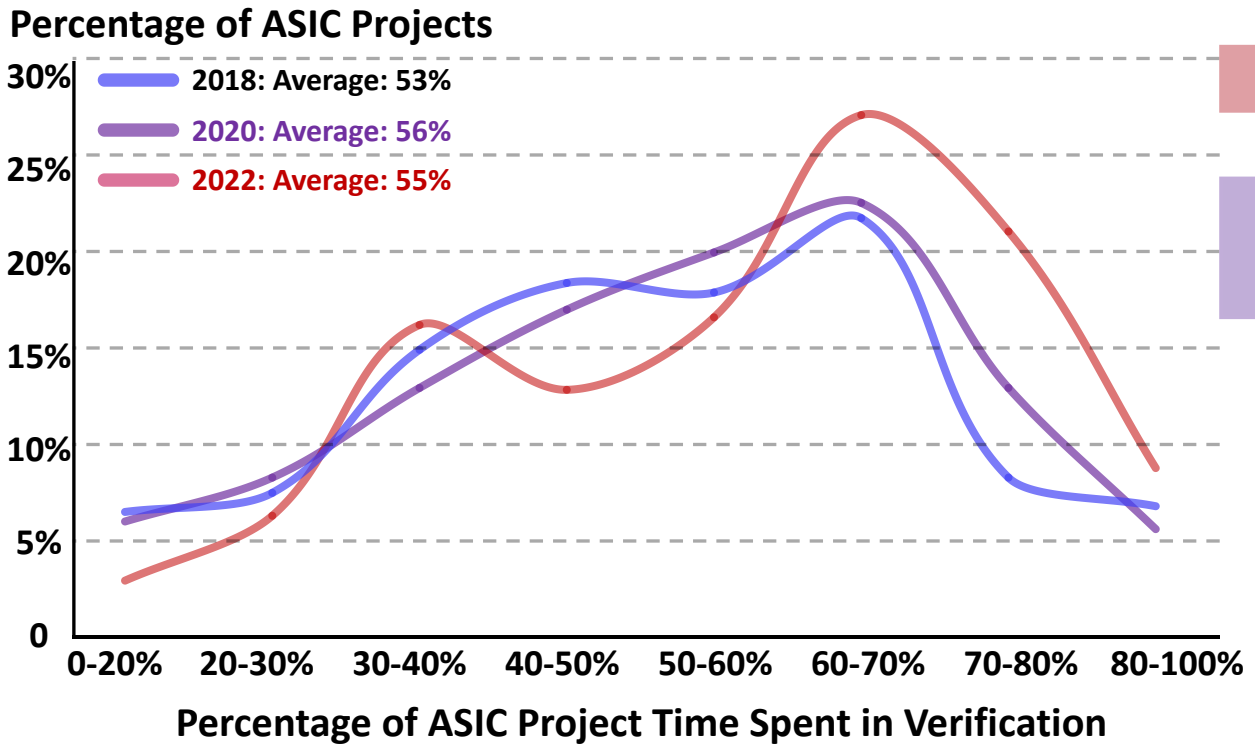
Background

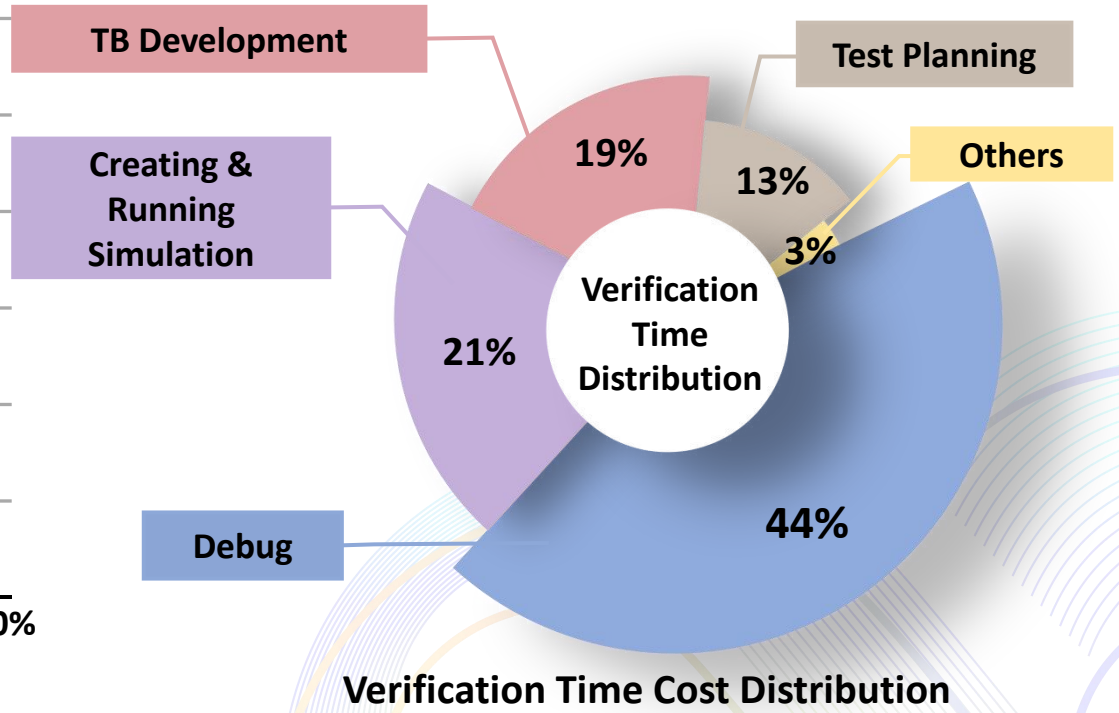


- **Verification accounts for over 50% development time.**
- **The engineer ratio of design to verification ranges from 1:2 to 1:3.**

Source: Harry Foster et. al.. The 2022 Wilson Research Group Functional Verification Study.

Background





- Verification accounts for over 50% development time.
- The engineer ratio of design to verification ranges from 1:2 to 1:3.
- Debugging dominates the verification process.

Source: Harry Foster et. al.. The 2022 Wilson Research Group Functional Verification Study.

What is Assertion?

When the posedge of signal <clk> is detected and reset signal <rstn> is active, if <rstn> rises, then implies <vld_out> equals to 1-bit binary number 0.

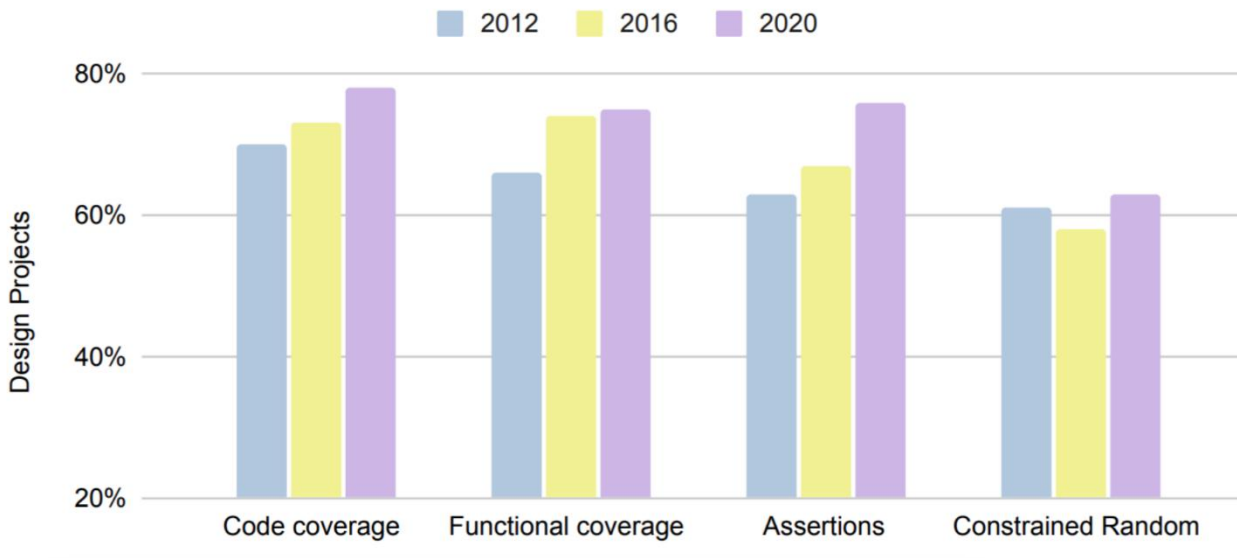
```
property vld_out_reset;
@(posedge clk) disable iff(!rstn) $rose(rstn)|-> vld_out == 1'b0;
endproperty
Assert_vld_out_reset:assert property (vld_out_reset) else $error(" unexpected <vld_out> reset
behavior ")
```

SVA (SystemVerilog Assertions) specifies the expected behavior under predefined conditions and captures potential errors through real-time verification of their correctness.

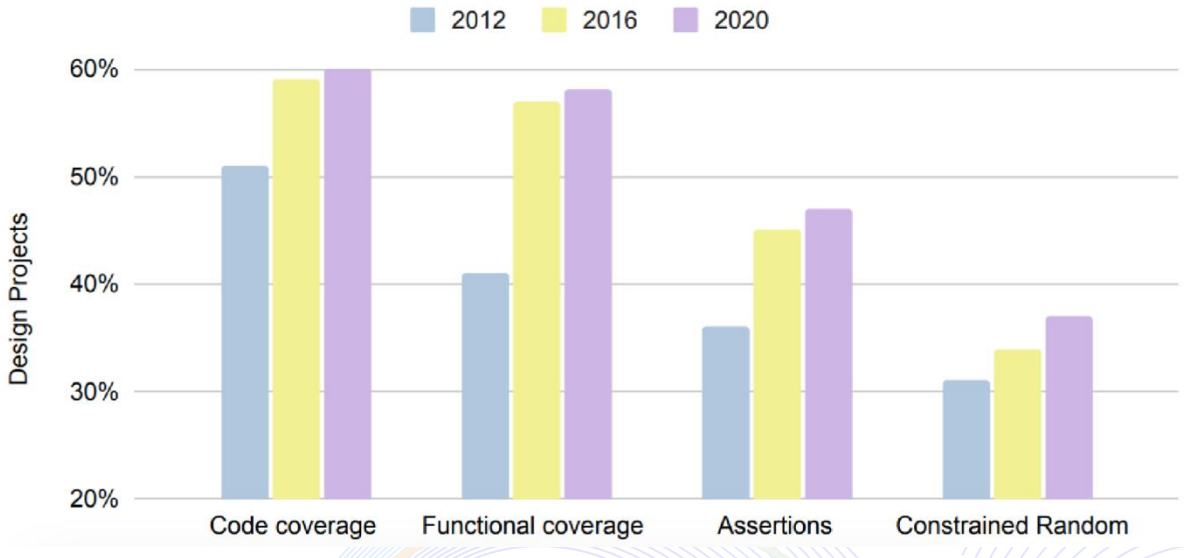
Advantage of Assertion:

- Early-Stage Error Detection
- Observability
- maintainability

Assertion Based Verification



ASIC functional verification trend

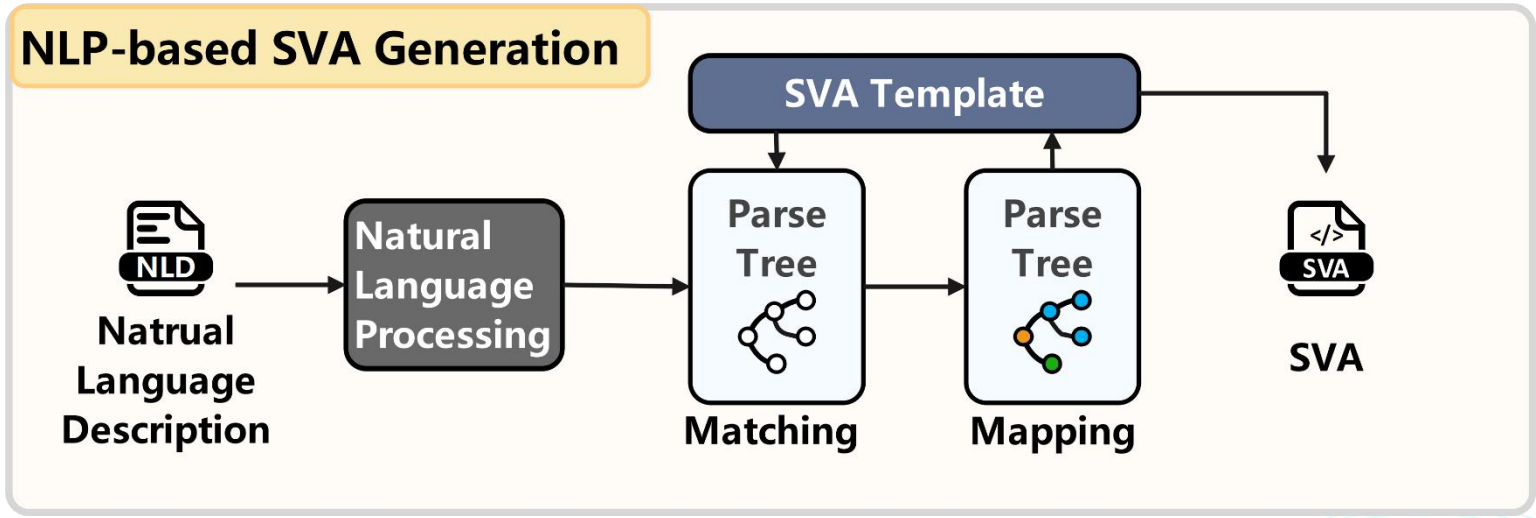


FPGA functional verification trend

However, SVA writing is labor-intensive and error-prone, agile SVA Generation is highly needed.

Source: Harry Foster et. al.. The 2020 Wilson Research Group Functional Verification Study.

NLP-based SVA Generation

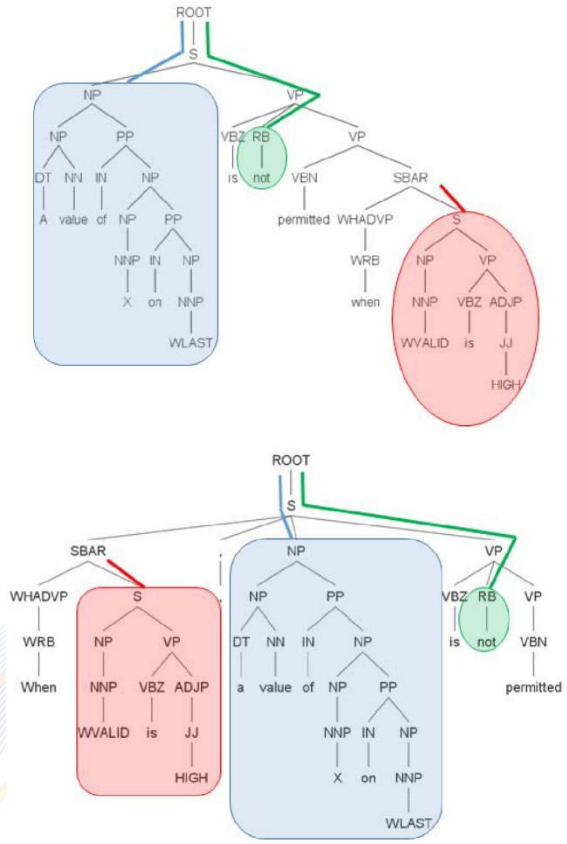


NLP-based approaches aim to develop translators that convert natural language verification requirements into SVA code. Early approach employed NLP techniques to parse SVA natural language descriptions into parse trees. These trees are then structurally matched to SVA templates, with key components like **signals**, **conditions**, and **actions** being automatically segmented. Finally, predefined mapping rules populate these modular components into template slots to generate SVA assertions.

NLP-based SVA Generation

Disadvantages :

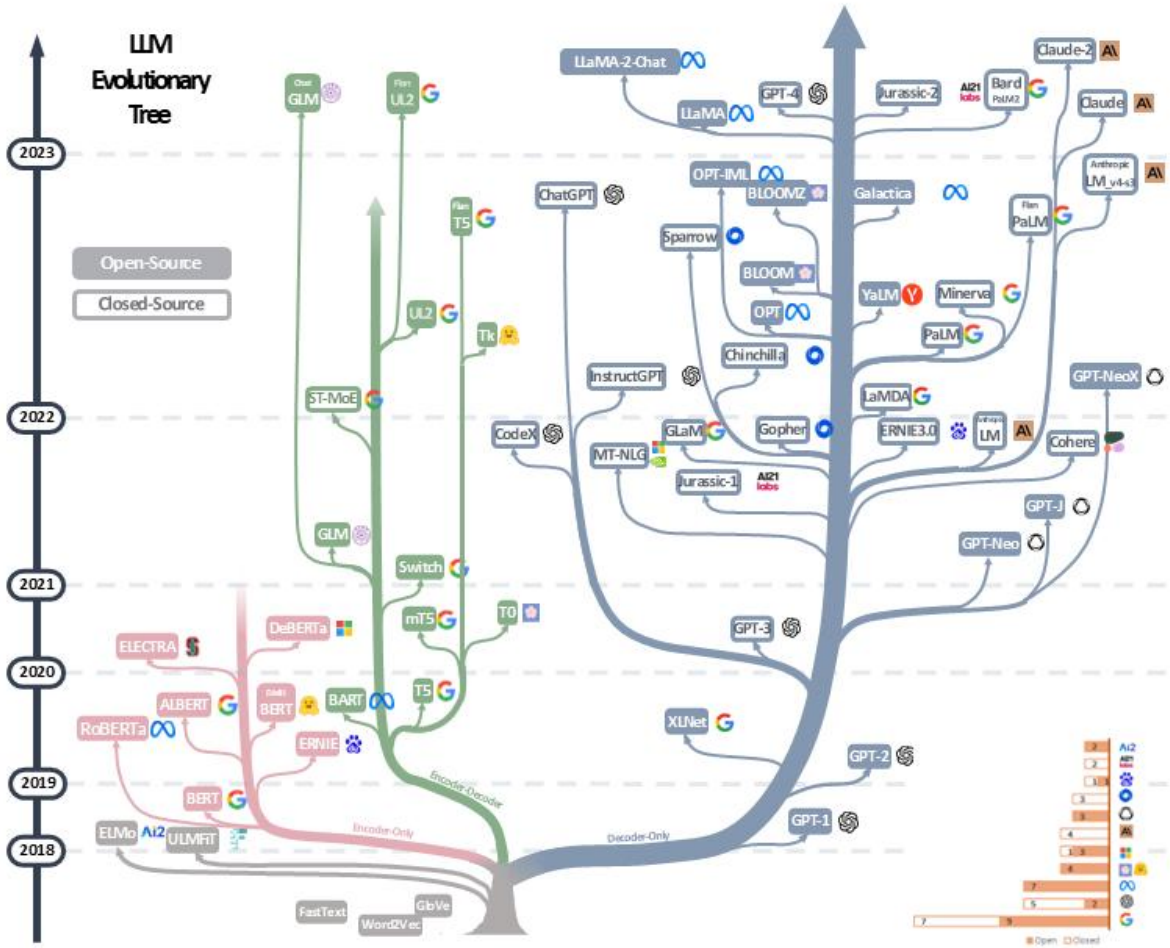
- Low generalization ability
- Limited abstraction-level of input
- Poor generative ability when faced with complex descriptions



The parse trees of two sentences with the same meaning but different expressions.[1]

[1] J. Zhao and I. G. Harris, "Automatic assertion generation from natural language specifications using subtree analysis," in 2019 Design, Automation Test in Europe Conference Exhibition (DATE), 2019, pp. 598–601.

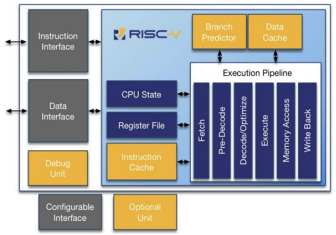
LLM Development



LLMs demonstrate comprehensive intelligent and automated capabilities.

LLM in IC

LLM+Architecture



Correlative Paper:

1. ChatCPU: An Agile CPU Design & Verification Platform with LLM
2. LLM-Based Processor Verification: A Case Study for Neuronnorphic Processor
3. Research on LLM Acceleration Using the High-Performance RISC-V Processor "Xiangshan" (Nanhu Version) Based on the Open-Source Matrix Instruction Set Extension (Vector Dot Product)
4. Using LLM such as ChatGPT for Designing and Implementing a RISC Processor: Execution, Challenges and Limitations
5. Llmcompass: Enabling efficient hardware design for large language model inference

……

LLM+FPGA

Correlative Paper:

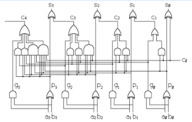
1. Understanding the potential of fpga-based spatial acceleration for large language model inference
2. The power of large language models for wireless communication system development: A case study on fpga platforms
3. Flightllm: Efficient large language model inference with a complete mapping flow on fpgas
4. Edgellm: A highly efficient cpu-fpga heterogeneous edge accelerator for large language models

……

LLM+RTL

```
module alu_8bit(
input[7:0] a,b,
input[2:0]Oper,
output reg c_out,
output reg[7:0] sum
   );

always@(a or b or Oper)
begin
        case(Oper)
        3'b000:  {c_out,sum} = a + b;
        3'b001:  {c_out,sum} = a - b;
        3'b010:  {c_out,sum} = b - a;
        3'b011:  {c_out,sum} = a | b;
        3'b100:  {c_out,sum} = a & b;
        3'b101:  {c_out,sum} = a ^ b;
        3'b110:  {c_out,sum} = a ~^ b;
        default: {c_out,sum} = 9'bx;
        endcase
end
endmodule
```



Correlative Paper:

1. Rtllm: An open-source benchmark for design rtl generation with large language model
2. RTLCoder: Fully Open-Source and Efficient LLM-Assisted RTL Code Generation Technique
3. Zero-Shot RTL Code Generation with Attention Sink Augmented Large Language Models
4. ITERTL: An Iterative Framework for Fine-tuning LLMs for RTL Code Generation
5. Make every move count: Llm-based high-quality rtl code generation using mcts
6. RTLRewriter: Methodologies for Large Models aided RTL Code Optimization
7. AutoVCoder: A Systematic Framework for Automated Verilog Code Generation using LLMs
8. OriGen: Enhancing RTL Code Generation with Code-to-Code Augmentation and Self-Reflection
9. Benchmarking large language models for automated verilog rtl code generation

……

Large Language Model

LLM+EDA

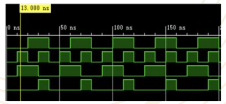
Correlative Paper:

1. Chateda: A large language model powered autonomous agent for eda
2. Large Language Models for EDA: Future or Mirage?
3. HLSPilot: LLM-based High-Level Synthesis
4. Llm4eda: Emerging progress in large language models for electronic design automation
5. EDA Corpus: A Large Language Model Dataset for Enhanced Interaction with OpenROAD
6. Customized Retrieval Augmented Generation and Benchmarking for EDA Tool Documentation QA
7. Chipnemo: Domain-adapted llms for chip design

……

LLM+Testbench

```
<module_name> # (
  .<parameter_name> (<parameter_value>)
)
<instance_name> (
  .<port_name> (<signal_name>),
  .<port_name> (signal_name>)
);
…… ……
initial begin
  and_in = 2b'00;
  #10 and_in = 2b'01;
  #10 and_in = 2b'10;
  #10 and_in = 2b'11;
end
…… ……
```

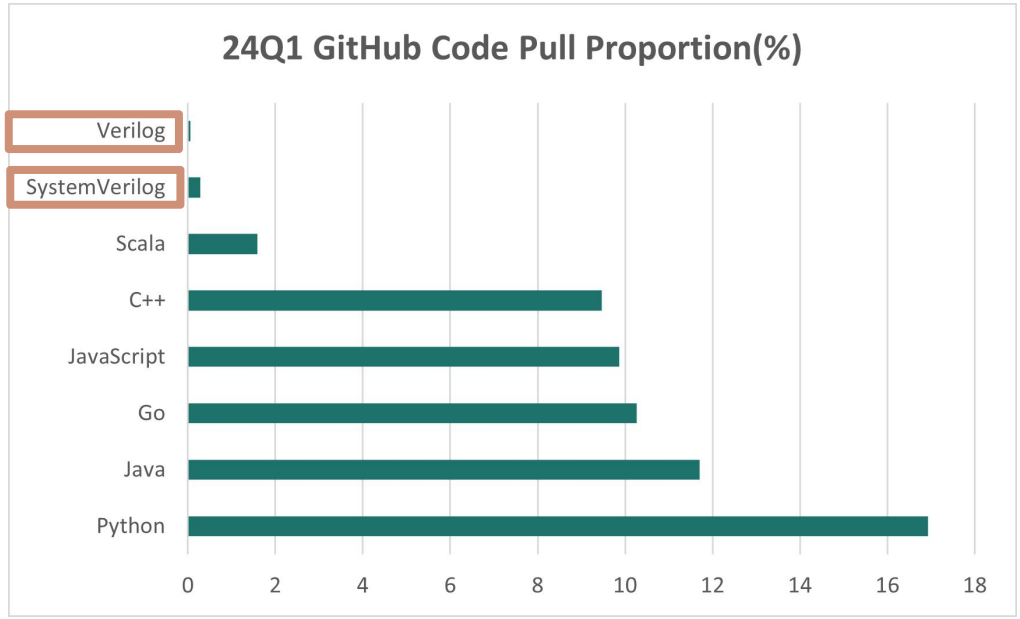


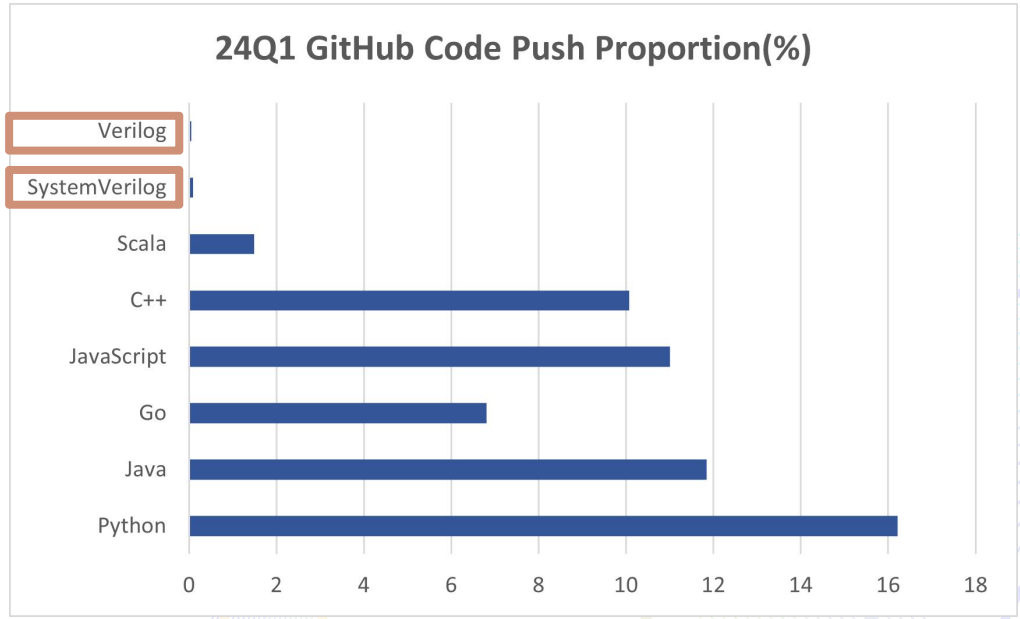
Correlative Paper:

1. From rtl to sva: Llm-assisted generation of formal verification testbenches
2. OpenLLM-RTL: Open Dataset and Benchmark for LLM-Aided Design RTL Generation
3. LAAG-RV: LLM Assisted Assertion Generation for RTL Design Verification
4. Revisiting VerilogEval: Newer LLMs, In-Context Learning, and Specification-to-RTL Tasks
5. MEIC: Re-thinking RTL Debug Automation using LLMs
6. AutoBench: Automatic Testbench Generation and Evaluation Using LLMs for HDL Design

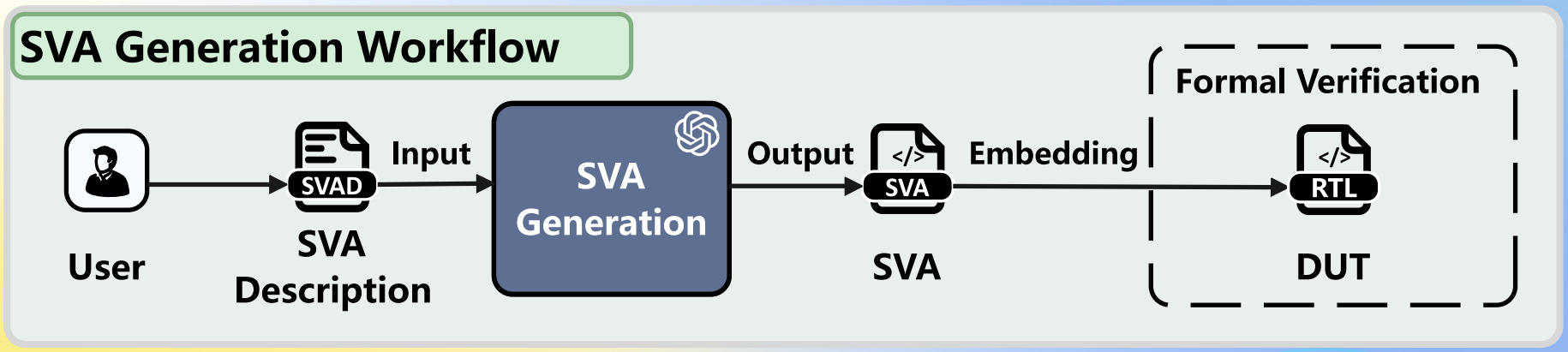
……

LLM in HDL Generation



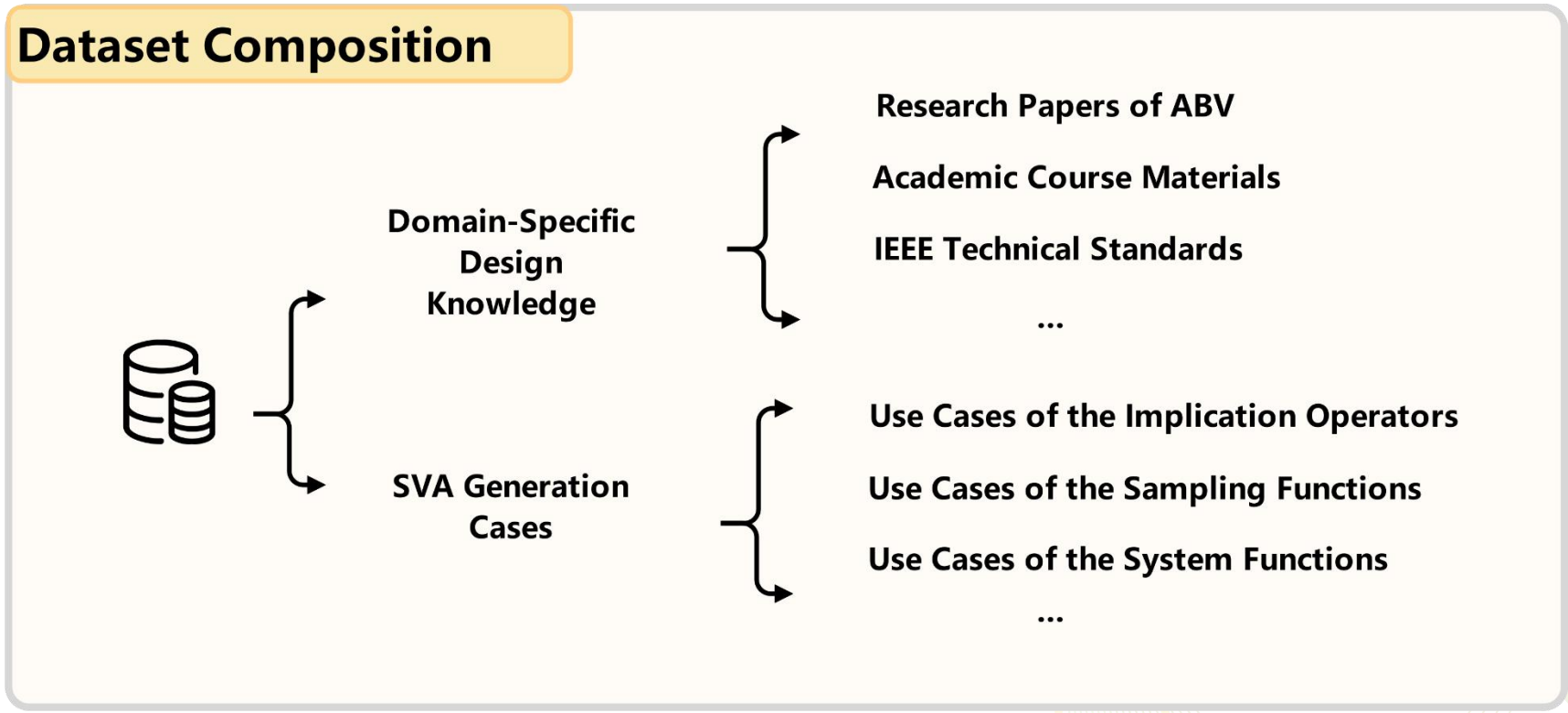


Our Workflow

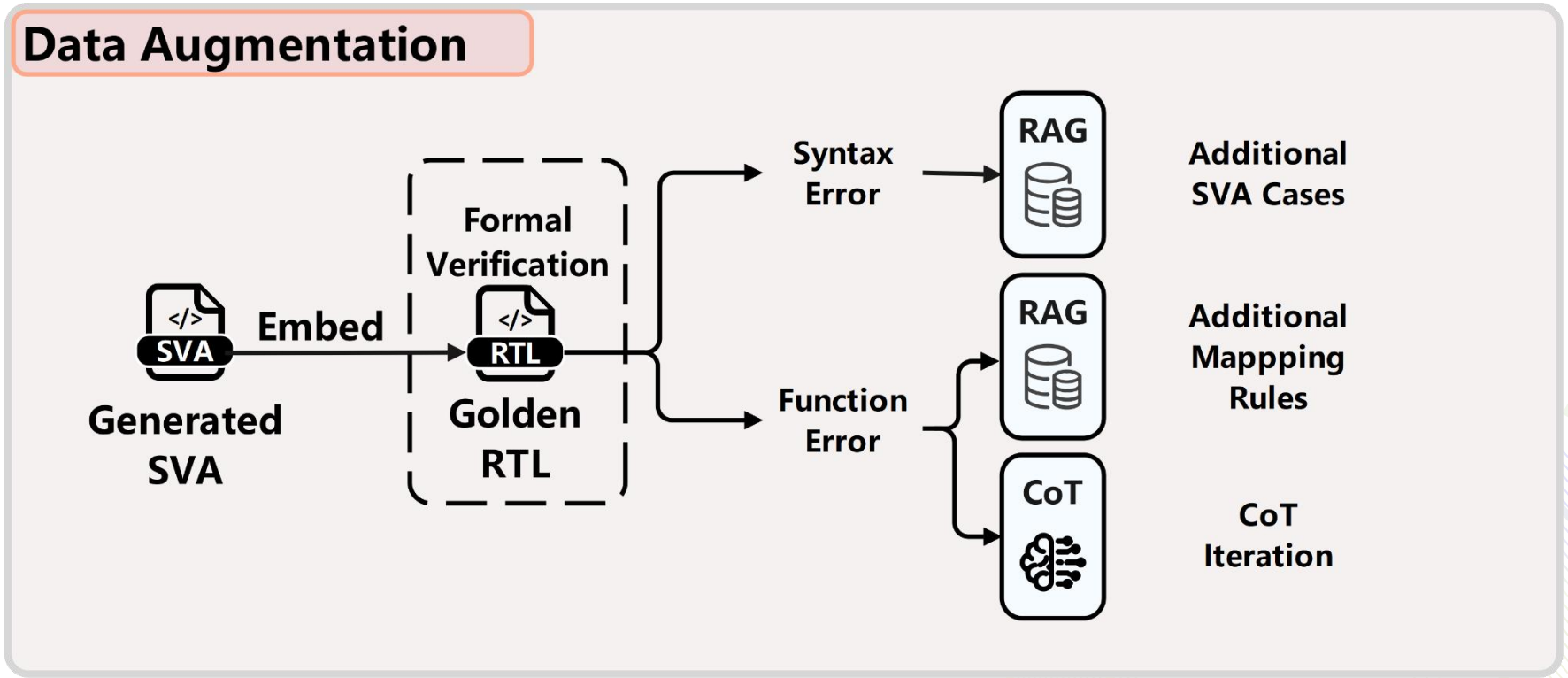


We designed an end-to-end automated generation system from SVA descriptions to SVAs, a LLM combined with customized Chain of Thought (CoT) and Retrieval-Augmented Generation (RAG), achieving high-quality SVA generation.

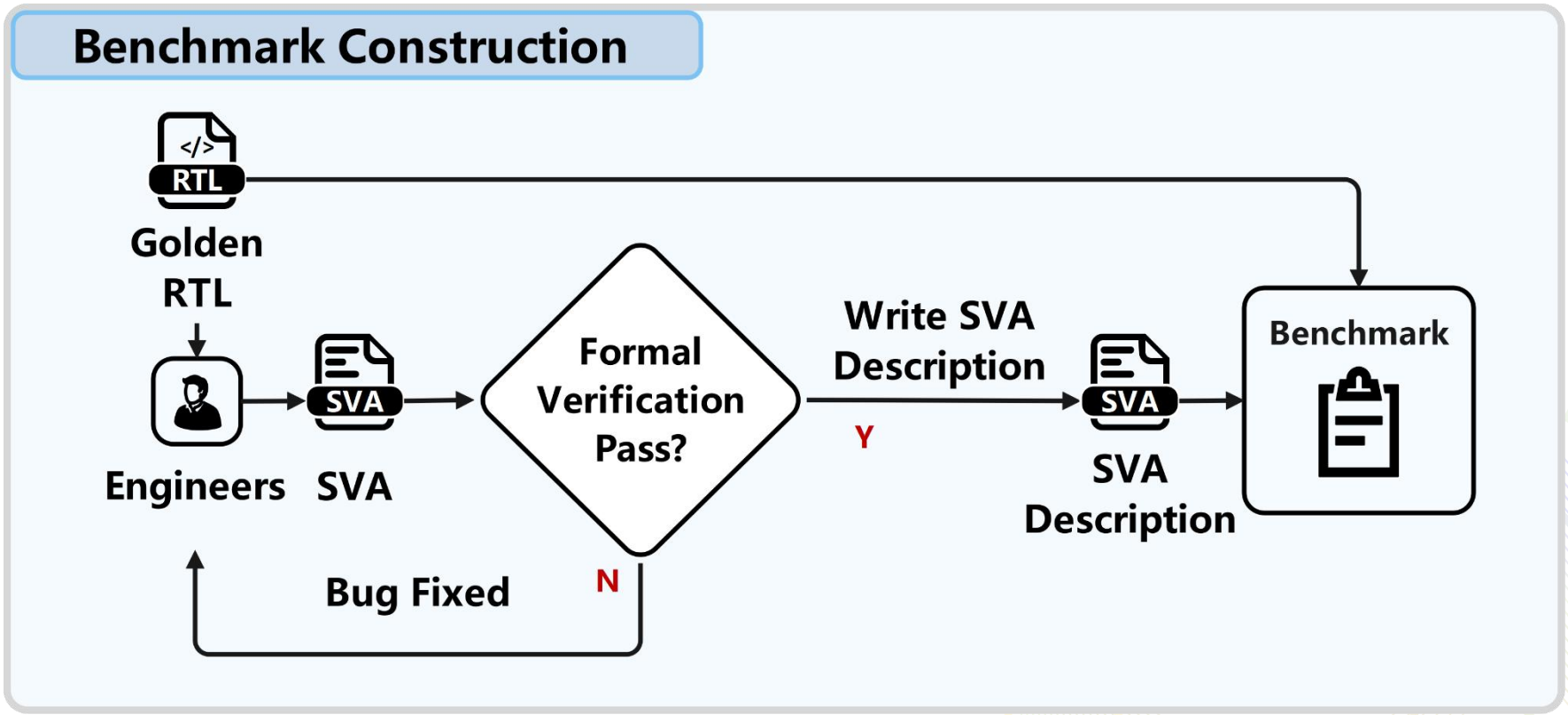
Dataset Composition of RAG



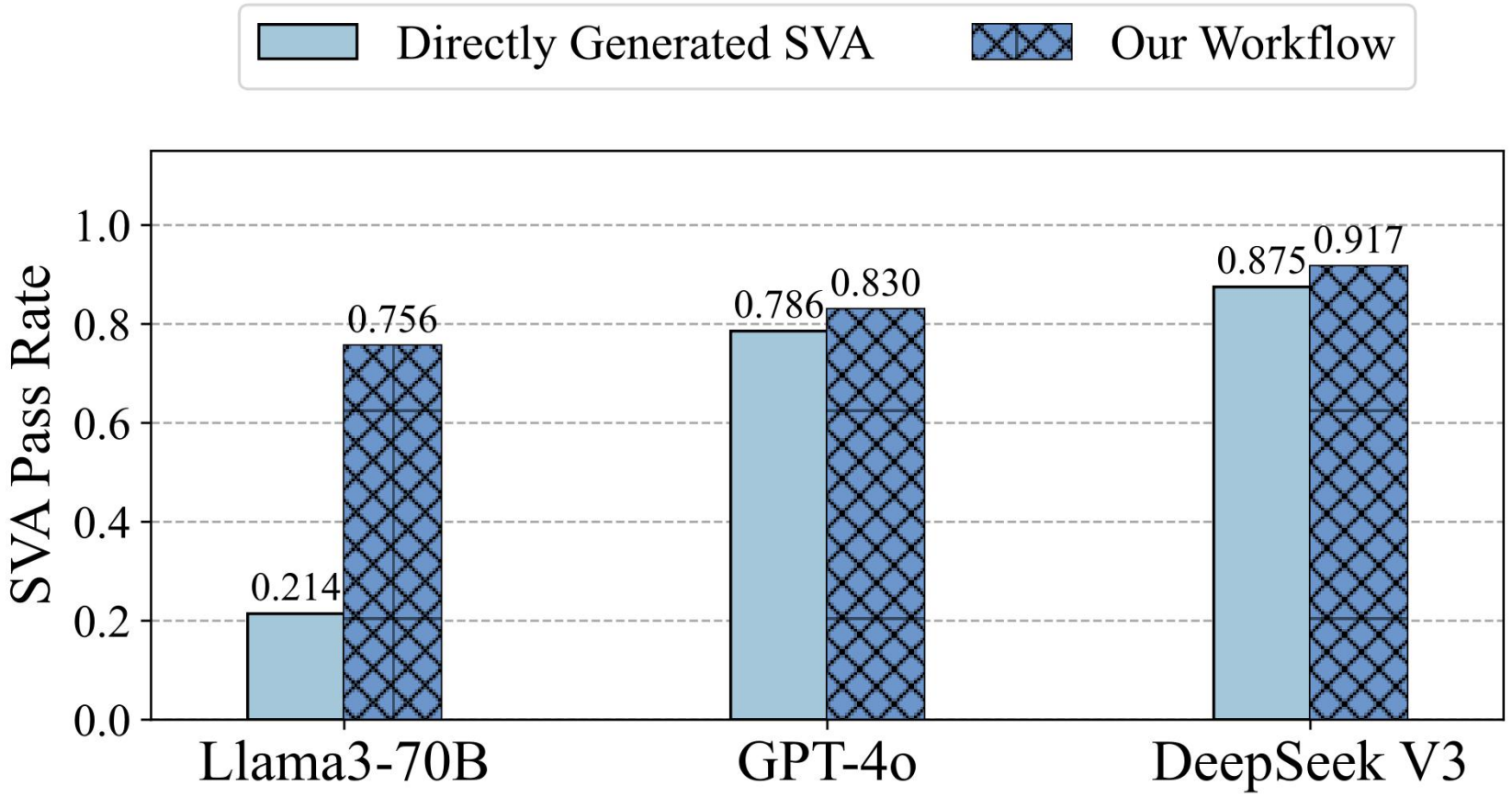
Data Augmentation



Benchmark Construction



Result


Directly Generated SVA
Our Workflow
SVA Pass Rate
1.0
0.8
0.6
0.4
0.2
0.0
0.214
0.756
0.786
0.830
0.875
0.917
Llama3-70B
GPT-4o
DeepSeek V3

Result

Situation	Example Input	Expected Output	Correctness of DeepSeek V3 with RAG+CoT
Complex Logic Nesting	When the posedge of signal <clk> is triggered, the result of the reduce and of not **<init_state>** and **<a>** should be true. where **<a>** is the reduce and of **<old_we_1>** and the result of **<ic_ram[{old_adr, 3'b001}]>** equal to the 23rd to 16th bit of **<old_di>.**	@(posedge clk) **(~init_state** & **(old_we_1 &** **(ic_ram[{old_adr, 3'b001}]** **== old_di[23:16]))))**	**75%**
Complex Temporal Expression	When the posedge of signal <wb_clk_i> is triggered and the signal <wb_rst_i> is not active, if <wb_cyc_i> and <wb_std_i> **both rise**, then it implies nonoverlappingly that the negation of the value of <wb_ack_o> at past 1 cycle is true for **at least 1 cycle** is true, and <wb_ack_o> should be low **1 clock cycle** later.	@(posedge wb_clk_i) disable iff(wb_rst_i) **$rose**(wb_cyc_i && wb_std_i) \|=> ~**$past**(wb_ack_o,1)**[*1:$]** **##1** !wb_ack_o	**60%**

Further Discussion

Template of SVA Description:

We imposed no restrictions on sentence structure or phrasing during dataset construction, though designers naturally exhibit preferred phrasing tendencies. Leveraging LLMs' strong generalization capabilities, simply introducing new cases into the RAG framework enhances their parsing accuracy for novel expressions.

Ensuring Accuracy of LLM-Generated SVA:

Current LLM-based automated SVA generation remains an assistive tool. Engineers must verify the correctness of generated assertions and employ iterative dialog refinement to obtain functionally valid SVAs.

Future Outlook:

This research demonstrates the significant enhancement of LLM assertion generation capabilities through the RAG+CoT framework, particularly when leveraging large-scale datasets. As generation strategies become more refined, verification-specific datasets expand, and LLMs evolve to the next generation, assertion pass rates will progressively increase. This advancement will enable assertion-based verification with substantially reduced engineering effort.

Special Thanks To

芯华章®

国家集成电路设计自动化技术创新中心
National Center of Technology Innovation for EDA

2025 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION CHINA

accellera SYSTEMS INITIATIVE

Thanks for listening!

16 Apr 2025 上海 SHANGHAI