TiDe : Timing diagram to Design verification model

Ramya B T (ramya.bt@samsung.com)
Sairam Jujjarapu (s.jujjarapu@samsung.com)
Harshit Sharma (h4.sharma@samsung.com)
Sachin Suresh Upadhya (sachin1.s@samsung.com)
Keerthi Kiran J (keerthi.k@samsung.com)
Samsung Semiconductor India R&D Center, Bangalore, India

accellera SYSTEMS INITIATIVE™

2025 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION INDIA

Introduction

Design verification is a highly time consuming process for design as well as verification engineers. It involves huge manual effort even with the best of the tools available in the market. In today's world where NLP and LLMs are widely being used to solve problems in digital design verification, we propose TiDe model which can streamline the verification process at various levels. The model takes timing diagram as input directly from spec and generates:

1. Unit testcases for basic testing and
2. Assertions for Formal verification

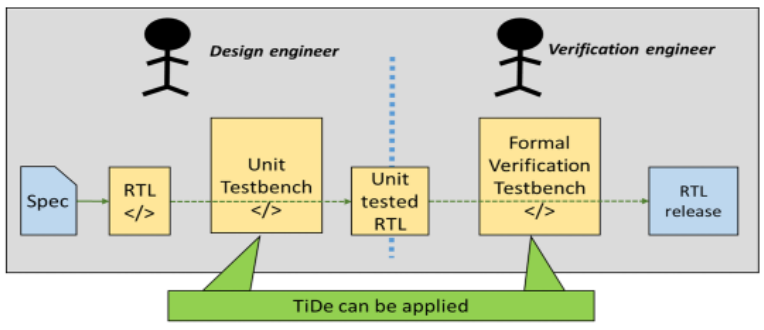
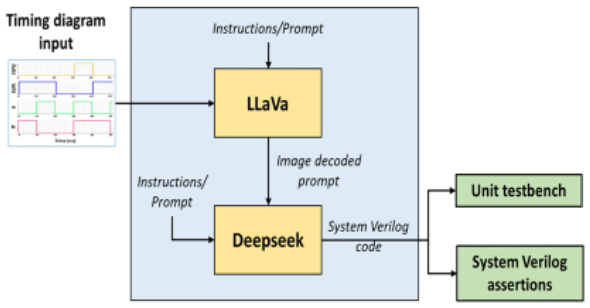


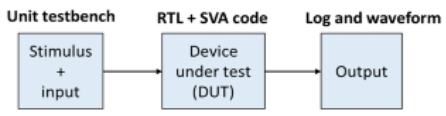
Figure : RTL development lifecycle

Proposed Methodology/Advantages

TiDe : Architecture



- ✓ Combination of two LLMs
- ❑ LLaVa -
  - ❑ Pre-trained model
  - ❑ Timing diagram information extraction
- ❑ Deepseek -
  - ❑ Pre-trained model
  - ❑ Code generation from extracted information
- ✓ Custom script for prompt generation

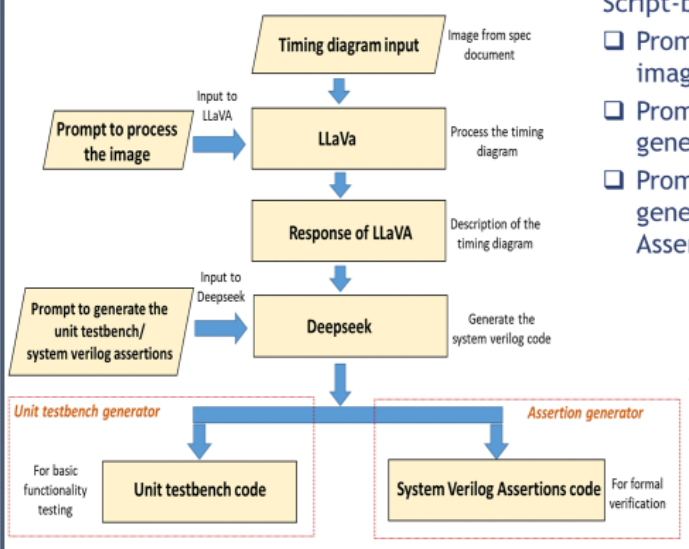


Verification : Architecture

Experimental setup	
Software	LLaVa LLM
	DeepSeek LLM
Verification	EDA Playground

Implementation Details

Implementation



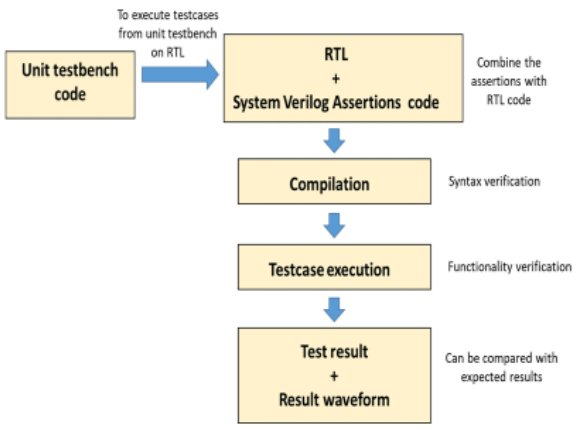
Script-based prompt generation

- ❑ Prompt for LLaVa to extract image information
- ❑ Prompt for Deepseek to generate unit testbench
- ❑ Prompt for Deepseek to generate System Verilog Assertions

TiDe flow has 2 modules:

- ❑ Unit testbench generator : To generate unit testcases
- ❑ Assertion generator : To generate System Verilog Assertions

Verification



Benchmarking

- ❑ Syntax - checking the compilation of the unit testbench and SVA code
- ❑ Functionality - intently injecting error into the RTL

Expectation

- ❑ Generated code is compiled for syntax checking and functionally verified
- ❑ Test is expected to pass for perfectly generated code and fail with appropriate assertions when error is injected

Results

- ❑ TiDe successfully generated System Verilog Assertions without fine-tuning in 60% of timing diagrams
- ❑ With minimal manual guidance/fine-tuning, the rate improved to approximately 70%

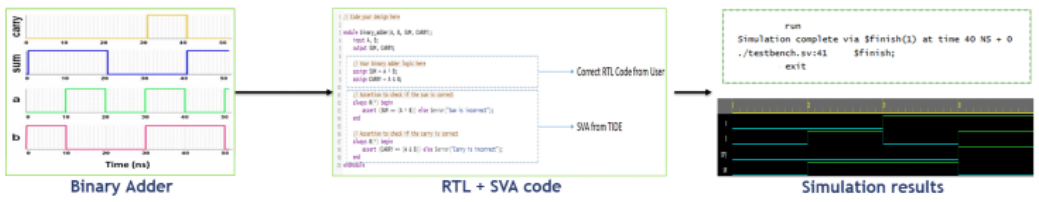


Figure 1 : TiDe executed on non-error injected RTL

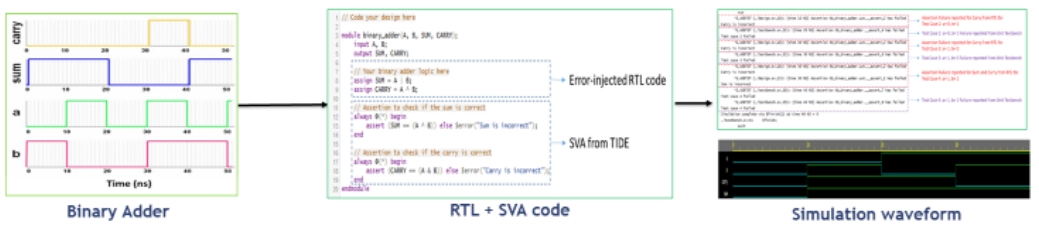


Figure 2 : TiDe executed on error-injected RTL

Conclusion

- As TiDe can be used to generate testbench from timing diagram, it can be highly useful for Design engineers to do unit testing and Formal verification engineers to do functionality verification.
- On consultation with design experts and verification experts and assortment of the data collected during an RTL project development life cycle, this has been found to reduce unit testing effort for design engineers by upto 80% and atleast 60% for verification engineers
- As unit testbench development and verification testbench development processes can be parallelized, TiDe can help save the time on overall project development life cycle as well
- The accuracy of the testcases depends upon the diagram. With the present level of training and fine-tuning, TiDe is able to convert timing diagrams with relatively less complexity. Hence, it can be considered a pilot deployment

REFERENCES

[1] Yao Lu et al, "RTLLM: An Open-Source Benchmark for Design RTL Generation with Large Language Model" 29th Asia and South Pacific Design Automation Conference (ASP-DAC) 2024

[2] M. Abrahams, J. Barkley , "RTL verification strategies", Wescon/98 IEEE Conference

SAMSUNG
Samsung Semiconductor India Research