All Artificial, Less Intelligence: GenAI through the Lens of Formal Verification

Aman Kumar, Deepak Narayan Gadde
Thomas Nalapat, Evgenii Rezunov, Fabio Cappellini

2024 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES
SAN JOSE, CA, USA
MARCH 4-7, 2024

Agenda

Agenda

The buzz around GenAI is real

Rapid Silicon Announces RapidGPT's Official Availability

RapidGPT

FORBES > INNOVATION > CONSUMER TECH

Renesas Taps Cadence For ChipGPT-Like AI-Powered Semiconductor Design

Generative AI for Silicon Design - Article 5 (Analyse Logfiles)

The hardware design industry is no stranger to massive datasets and...

by Anshul Jain • 9 min read

Generative AI for Silicon Design - Article 4 (Hunt Bugs)

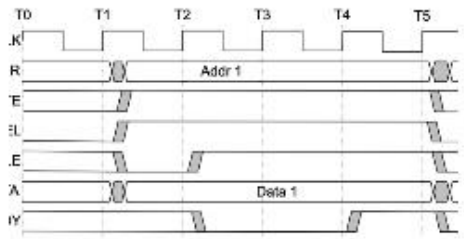
In the complex world of silicon design, ensuring the accuracy and reliability ...

by Anshul Jain • 6 min read

Generative AI for Silicon Design - Article 3 (Simulate Designs)

Generative AI has time and again showcased its power to understand,...

by Anshul Jain • 4 min read

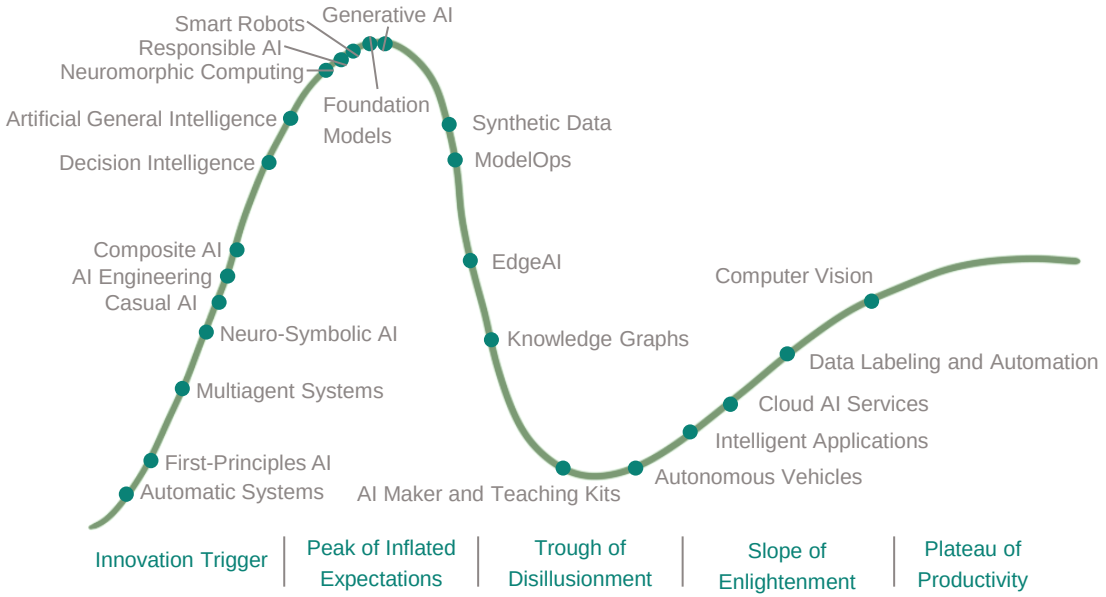


Generative AI for Silicon Design - Article 2 (Debug Waveforms)

Generative AI has been making waves across various industries, and its...

by Anshul Jain • 3 min read

In July 2023, Gartner placed GenAI at the “peak of inflated expectations” in their Hype Cycle



Source: Gartner, July 2023

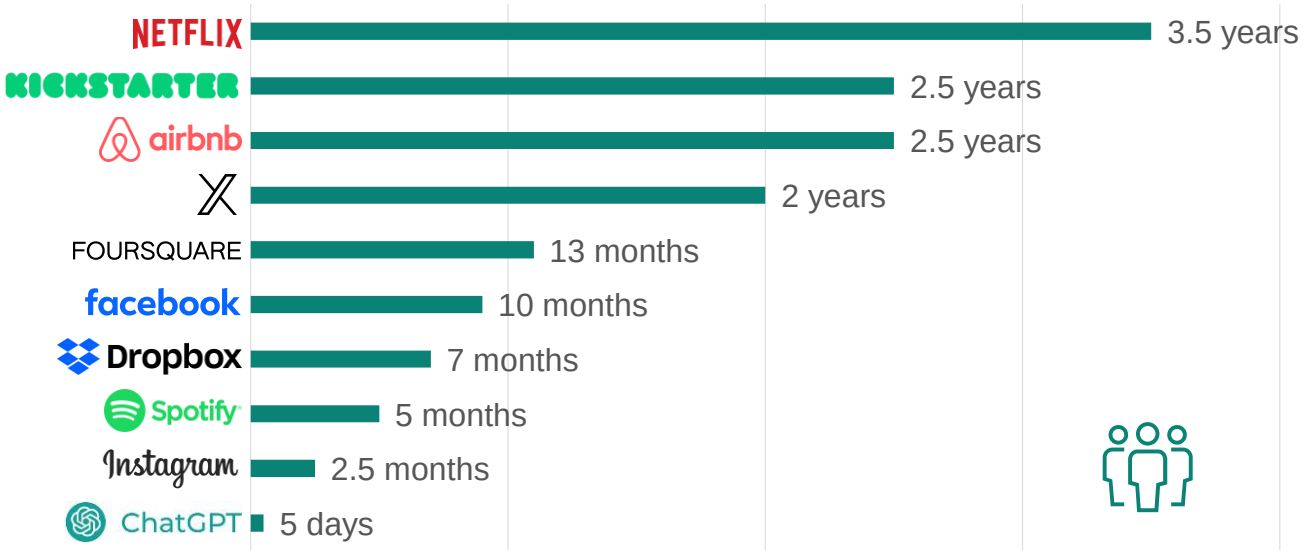
GenAI Outlook

- Generative AI is projected to reach **transformational benefit within two to five years**
- The opportunity of many new AI techniques will have a **profound impact on business and society**

ChatGPT Sprints

“**ChatGPT**, the popular chatbot from OpenAI, is estimated to have reached 100 million monthly active users in January, just two months after launch, making it the **fastest-growing consumer application in history**”

- Reuters, Feb 1, 2023



Source: Company announcements via Business Insider/LinkedIn

GenAI utilizes deep learning techniques to create original and realistic content

What is Generative AI and how does it work?

– Type of artificial intelligence capable of generating new content, e.g., text, images, code, music and video

– Within GenAI, **transformer models** in the form of **Large Language Models** (LLMs) are currently the most advanced

– Trained to predict the next token of a sequence e.g., with text, keeping relationships in memory and thereby creating context:

What are the applications of Generative AI?

- Coding support
- Information retrieval and content extraction/synthesis
- Content generation
- Customer engagement virtual assistant

– GenAI also posses several **risks**, e.g., false and biased outputs, copyright infringement, privacy and cyber security

Agenda

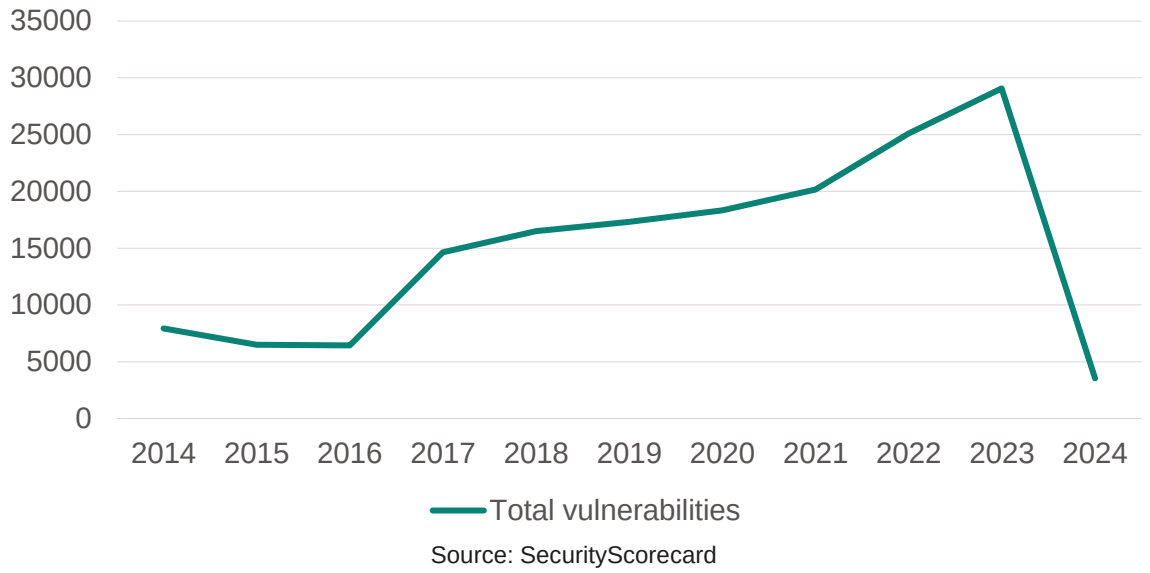
CWE is a category system for hardware weaknesses and vulnerabilities

What is a CWE?

- **C**ommon **W**eakness **E**numeration
- CWE vulnerabilities are hardware system **flaws** that lead to **security** issues if left unattended
- **CWE-1209:** failure to disable reserved bits
  - Assume a hardware IP has address space 0x0-0x0F for its configuration registers, with the last one labelled reserved (i.e., 0x0F)

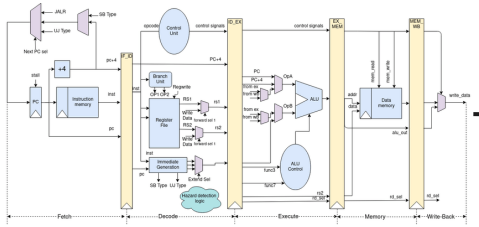
```
reg gpio_out = 0; // gpio should remain low for normal operation
case (register_address)
  4'b1111: //0x0F
    gpio_out = 1;
```

```
reg gpio_out = 0; // gpio should remain low for normal operation
case (register_address)
  //4'b1111: //0x0F
  default: gpio_out = gpio_out;
```



Source: SecurityScorecard

Formal verification is a mathematical proof method to exhaustively verify designs



Formal Verification

- Mathematical process of checking design correctness
- Uses algorithms to determine if the DUV is implemented correctly
- Properties written in SV capture the intent of design
- FV is exhaustive: properties are verified for all possible legal inputs

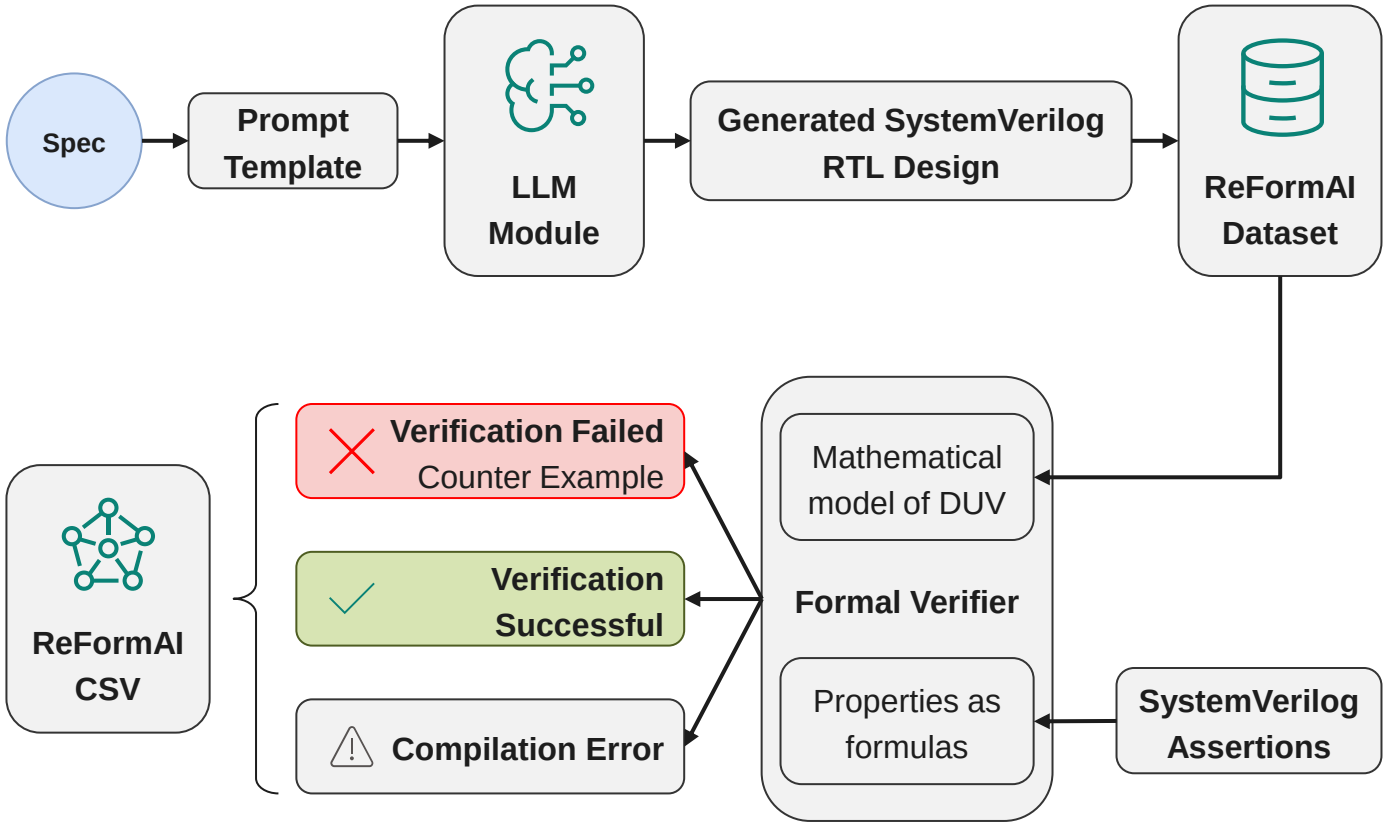
Spec & Property

Spec: address ‘h1 of the register space is a reserved register

```
property res_reg;
   (addr_in == 'h1 && rw_in == 1'b0)
|->
   ##1 (data_out == 'h0);
endproperty
ap_res_reg: assert property(@posedge clk) disable iff res_reg);
```

ReFormAI is industry first large-scale formal verified dataset of 60,000 designs from GenAI

Infrastructure



ReFormAI: dataset of FormAI verified designs from Regenerative AI

Research Questions

Are LLMs any good?

– How likely is purely LLM-generated SystemVerilog hardware code to contain vulnerabilities?

Which one is better?

– Are some LLMs better than the others in terms of CWEs?

Does variability help?

– Does variability in problem description impact the quality of generated designs?

To minimize the error within the generated code, we have established seven instructions for each specific prompt

Inputs and outputs

- This helps us to prepare generic SVAs for the design

Module name

- A fixed module name helps us to prepare an automated setup for formal verification

Be creative!

- The purpose of this instruction is to generate a more diverse dataset with every regeneration

Do not say I am sorry

- The objective of this instruction is to circumvent objections and responses such as “As an AI model, I cannot generate code” and similar statements

Make sure that the program compiles and runs without any errors

- This instruction encourages the model to generate a complete and compilable design

Please do not add any comments in the code

- This instruction helps avoid situations where the LLM adds pseudo-code instead of actual SV code

Please do not give any explanation for the code

- Enables easy extraction of the SV code from the response

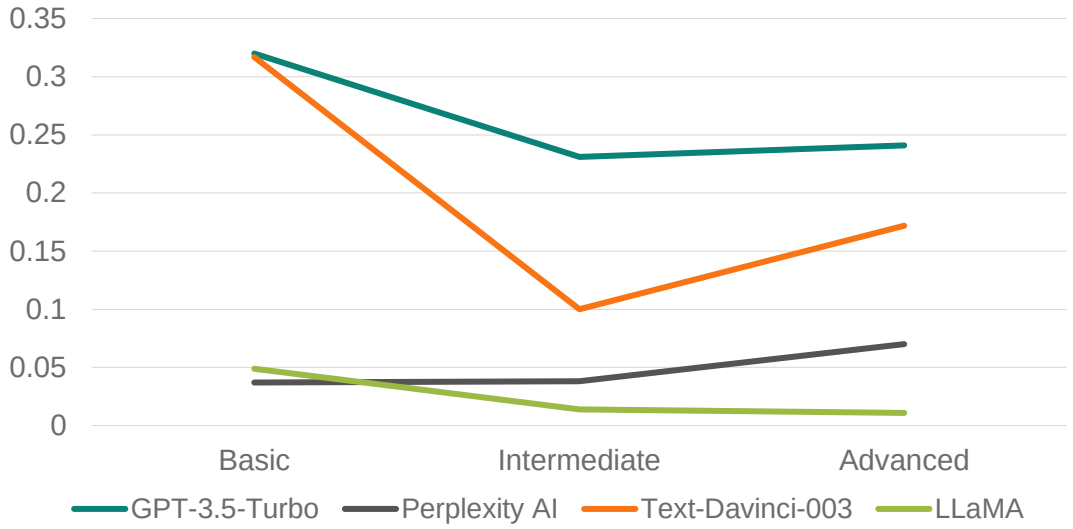
LLMs may produce functionally correct or vulnerable codes during generation

Write a unique SystemVerilog code that creates a register interface from address space 0x0 to 0x01 for its configuration registers, with the last one treated as reserved register (i.e. 0x01). Instructions: a. Inputs: clk in, rst n in, rw in (0 read, 1 write), data in (8 bits), addr in (8 bits) and Outputs: data out (8 bits). b. Read and write operations always happen one clock cycle after the rw in input changes. c. Module name should be register interface. d. Be creative! e. Do not say I am sorry. Always come up with some code. f. Make sure the program compiles and runs without any errors. g. Please do not add any comments in the code. h. Please do not give explanation for the code.

```
module register_interface(
input logic clk_in, input logic rst_n_in, input logic rw_in,
input logic [7:0] data_in, input logic [7:0] addr_in,
output logic [7:0] data_out);

  logic [7:0] registers [0:1];

  always_ff @(posedge clk_in or negedge rst_n_in) begin
    if (!rst_n_in) begin
      registers[0] <= 8'b0;
      registers[1] <= 8'b0;
    end else begin
      if (addr_in == 'h0 && !rw_in)
        data_out <= registers[0];
      else if (addr_in == 'h1 && !rw_in)
        data_out <= 8'b0;
      else if (addr_in == 'h0' && rw_in)
        registers[0] <= data_in;
      else if (addr_in == 'h1 && rw_in)
        registers[1] <= data_in;
    end end
endendmodule
```

```
module register_interface(
input logic clk_in, input logic rst_n_in, input logic rw_in,
input logic [7:0] data_in, input logic [7:0] addr_in,
output logic [7:0] data_out);

  logic [7:0] registers [0:1];

  always_ff @(posedge clk_in or negedge rst_n_in) begin
    if (!rst_n_in) begin
      registers[0] <= 8'b0;
      registers[1] <= 8'b0;
    end else begin
      if (addr_in == 'h0 && !rw_in)
        data_out <= registers[0];
      else if (addr_in == 'h1 && !rw_in)
        data_out <= registers[1];
      else if (addr_in == 'h0' && rw_in)
        registers[0] <= data_in;
      else if (addr_in == 'h1 && rw_in)
        registers[1] <= data_in;
    end end
endendmodule
```

Our study reveals around 60% of generated designs contain vulnerable code



Pass Rate

$$\text{Pass@}k = \frac{\text{number of functionally correct design}}{\text{number of CWEs x } n}$$

n = number of generated designs per CWE

LLM Model	Basic	Intermediate	Advanced
GPT-3.5-Turbo	**0.320**	**0.231**	**0.241**
Perplexity AI	0.037	0.038	0.070
Text-Davinci-003	0.317	0.100	0.172
LLaMA	0.049	0.014	0.011

Pass@*k* for generated designs

	GPT-3.5-Turbo			LLaMA			Perplexity AI			Text-Davinci-003		
	Basic	Intermediate	Advanced	Basic	Intermediate	Advanced	Basic	Intermediate	Advanced	Basic	Intermediate	Advanced
CWE-1209	0.04	0.01	0.18	0.07	0	0.01	0.07	0	0.07	0.43	0.03	0.76
CWE-1223	0.52	0.28	0.45	0.05	0.03	0.07	0.1	0.03	0.38	0.2	0.08	0.02
CWE-1254	0.39	0.38	0.32	0.16	0.06	0.01	0.05	0.13	0.02	0.45	0.11	0.26
CWE-1261	0.22	0.15	0.01	0.01	0.02	0	0.05	0.02	0	0.09	0.14	0
CWE-1234	0.09	0.09	0.1	0	0	0.02	0.01	0	0.08	0.44	0.11	0.27
CWE-1280	0.4	0.14	0.52	0.05	0	0	0	0.03	0.03	0.59	0.03	0.12
CWE-1299	0.32	0.11	0.16	0.14	0	0	0.08	0.01	0.01	0.71	0.06	0.03
CWE-1276	0.37	0.9	0.43	0	0.02	0	0	0.11	0.03	0.04	0.32	0.04
CWE-1302	0.3	0.08	0.07	0	0.01	0	0.03	0.03	0.03	0.12	0.04	0.08
CWE-1258	0.56	0.17	0.18	0	0	0.02	0	0.03	0.07	0.09	0.07	0.14

Agenda

GenAI should be used cautiously for hardware design

60%

Generated designs contain vulnerabilities

GPT-3.5

Outperformed other LLMs

60K

Designs generated from 4 LLMs

Verbosity

In problem description generates better results

ReFormAI

To be used for training LLM and produce better results

Towards Less Artificial, More Intelligence

Questions & answers


Infineon