Data Collection for Fine-Tuning LLMs for Hardware Verification

Researcher: Alma Babbitt, Computer System Engineering
Mentor: Nakul Gopalan (Assistant Professor)
School of Computing and Augmented Intelligence

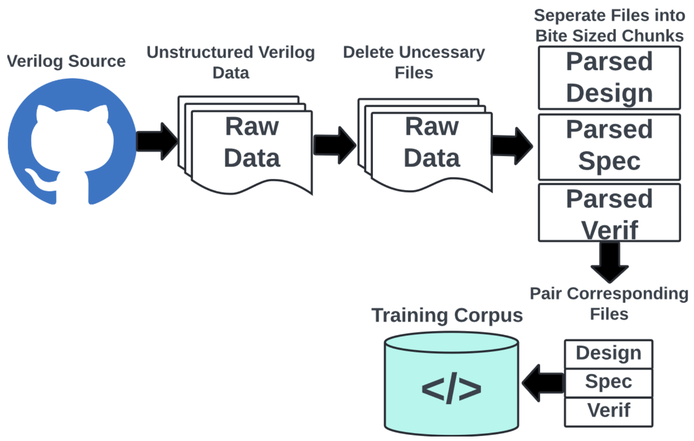
Objective

The chip design industry uses hardware description languages such as Verilog for designing and testing integrated circuits. Verifying a circuit design is complex and resource-intensive. This study aims to collect a dataset that can be used for fine-tuning LLMs (Large Language Models) for generating verification code for circuit designs written in Verilog.

Method

Fine-tuning LLMs necessitates labeled data. GitHub's public repositories were scraped, and a combination of Python scripts and manual collection methods were employed to gather and filter the data. This collected data can then be used to fine-tune ChatGPT.

Fig. 1 Data Collection Pipeline



Challenges

1. Non-uniformity of structure of specification documents
2. Limited Verilog repositories to scrape
3. Parsing code and matching with specification description is non-trivial

Progress

- GitHub identified as the best source for scraping open-source repositories.
- GitHub API calls used to download hundreds of repositories if Verilog files are found in the repository.
- Scripting was used to keep repositories with design, specification, and corresponding verification files for modules and submodules. (License information kept as well). These files are identified by file location, name, and extension.
- Manual collection used in place of scripting to further parse and collect self-contained data points. This was done to better understand how scripts can automate this process.
- 50 hand curated data points have been collected so far.
- We have created a dashboard for the dataset and are working on curating it with more metadata, as well as adding more data points.
- They were fed into ChatGPT to fine-tune the model to verify the dataset collected so far. Improvement in code generated was observed.

Fig. 2 Data Hierarchy for Each Entry in the Dataset

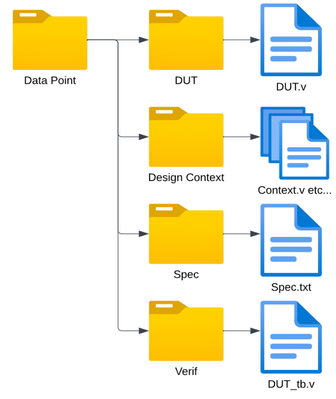


Fig. 3 Snapshot of Dashboard of Collected Data

	A	B	C	D
1	ID	design	spec	verif
2	0	trng0\design	trng0\spec	trng0\verif
3	1	trng1\design	trng1\spec	trng1\verif
4	2	trng2\design	trng2\spec	trng3\verif
5	3	trng3\design	trng3\spec	trng3\verif
6	4	verilog-divider\design	verilog-divider\spec	verilog-divider\verif
7	5	ethmac0\design	ethmac0\spec	ethmac0\verif
8	6	ethmac1\design	ethmac1\spec	ethmac1\verif
9	7	i2c\design	i2c\spec	i2c\verif
10	8	chacha0\design	chacha0\spec	chacha0\verif
11	9	chacha1\design	chacha1\spec	chacha1\verif
12	10	chacha2\design	chacha2\spec	chacha2\verif
13	11	cryptech_uart\design	cryptech_uart\spec	cryptech_uart\verif
14	12	microprocessor\design	microprocessor\spec	microprocessor\verif
15	13	mkmif0\design	mkmif0\spec	mkmif0\verif
16	14	mkmif1\design	mkmif1\spec	mkmif1\verif
17	15	mkmif2\design	mkmif2\spec	mkmif2\verif
18	16	sha10\design	sha10\spec	sha10\verif
19	17	sha11\design	sha11\spec	sha11\verif
20	18	sha12\design	sha12\spec	sha12\verif
21	19	vndecorrelator\design	vndecorrelator\spec	vndecorrelator\verif
22	20	cryptech_uart\design	cryptech_uart\spec	cryptech_uart\verif
23	21	fifo\design	fifo\spec	fifo\verif

Conclusion

Dataset creation is key to fine-tune or train LLMs to generate verification code. We have taken a step in that direction. Future work involves filtering and scraping more open-source repositories for thousands of data points. Further scripting will be used to precisely separate the repositories in digestible chunks for ChatGPT. An LLM will be fine-tuned with this dataset using 100% code coverage as the goal.

Acknowledgements

Special thanks to **Prof. Aman Arora** for collaborating on this research!

FURI

Ira A. Fulton Schools of Engineering
Arizona State University