Automating Hardware Design with Large Language Models

Jason Blocklove

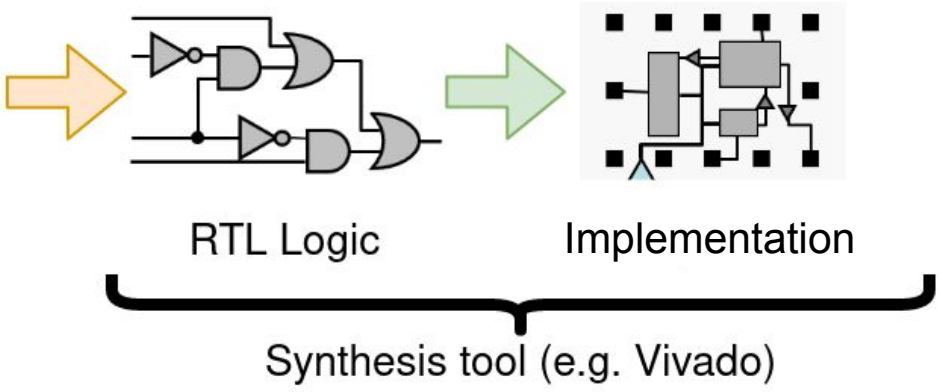
June 2024

The Verilog HDL

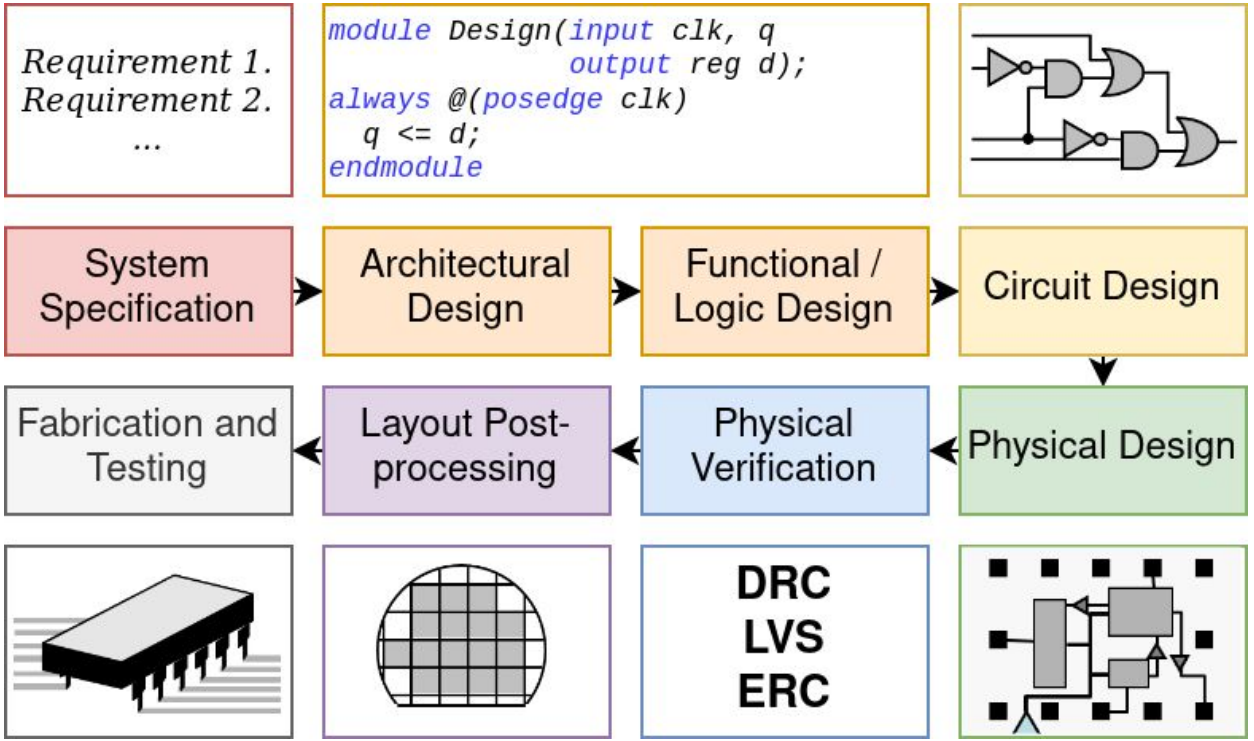
- Verilog used to design digital systems

```
module Design(input a, b, c, d
              output q);
 assign t1 = ~b & c;
 assign t2 = ~c & d;
 assign t3 = a | t1;
 assign q = t2 | t3;
endmodule
```

Verilog Code



The Digital Design pipeline



- Specification → Implementation is hard

2021-2023: ML-written code by Large Language Models

iblnews.org

Home › Top News › StarCoder, a New Free (

StarCoder, a New Free Code-Generating Model Alternative to GitHub's Copilot

By **IBL News** - May 8, 2023

reddit PROGRAMMING comments other discussions (18)

266(

GitHub Copilot · Your AI pair programmer (copilot.github.com)

submitted 2 months ago by violinclipper 4 & 14 more

581 comments share save hide give award report crosspost

The Verge

GitHub and OpenAI launch an AI Copilot tool that generates its own code

GitHub and OpenAI have launched a technical preview of a new AI tool called Copilot, which lives inside the Visual Studio Code editor and ...

Jun 29, 2021

Y **Hacker News** new | threads | past | comments | ask | show | jobs | submit

▲ GitHub Copilot (copilot.github.com)

2905 points by todsacerdoti 75 days ago | hide | past | favorite | 1272 comments

TechTalks HOME BLOG TIPS & TRICKS WHAT IS INT

Home › Blog › What OpenAI and GitHub's "AI pair programmer" means for the softwa

Blog

What OpenAI and GitHub's "AI pair programmer" means for the software industry

By **Ben Dickson** - July 5, 2021

Introducing GitHub Copilot: your AI pair programmer

Nat Friedman

INTERESTING ENGINEERING JOBS Subscribe | Log In

Open AI's Codex tool claims to help developers write code faster, better

It can't fix the code when it does not work though.

Ameya Paleja

Created: Mar 07, 2023 08:02 AM EST

2022-2023 “Conversational” LLMs

OpenAI

Menu

Introducing ChatGPT

We’ve trained a model called ChatGPT which interacts in a conversational way. The dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests.

Illustration: Ruby Chen

HuggingChat v0.2

Making the community’s best AI chat models available to everyone.

NEW Chat UI is now open sourced on GitHub ↗ GitHub repo

Current Model

OpenAssistant/oasst-sft-6-llama-30b

↗ Model page ↗ Dataset page Website

40 YEARS PC

Find products, advice, tech news

Home › News › Artificial Intelligence

ChatGPT Passes Google Coding Interview for Level 3 Engineer With $183K Salary

I'm Bard, your creative and helpful collaborator. I have limitations and won't always get it right, but your feedback will help me to improve.

Not sure where to start? You can try:

Give me some ideas to surprise my movie-loving friend on their birthday

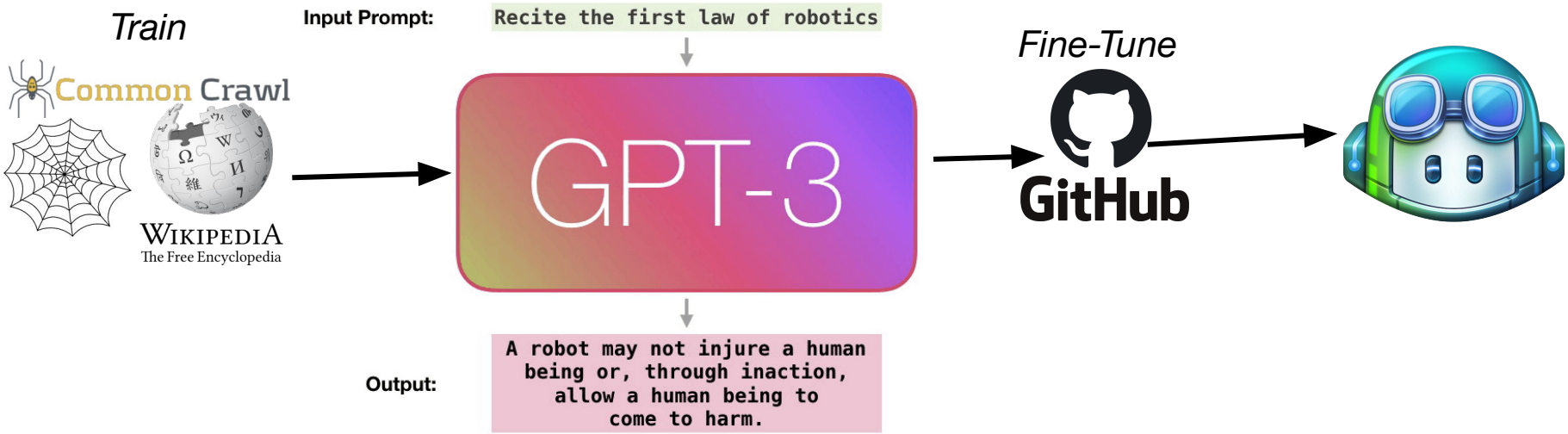
Draft a design brief for my community bookstore’s new logo

What’s a fast, balanced vegetarian meal for me to make? It should be high in protein and fiber

How do LLMs work?

How are LLMs made?

- LLMs are trained over large text datasets to “learn” to predict the next tokens
- Can be fine-tuned for specific tasks,
  - e.g. code-writing:



How do LLMs "generate"? (simplified)

Prompt: (code, comments → tokens)

int	main	(	int

Suggestions:

Token	Probability
argc	92%
)	6%
argv	1%
*	0.1%
{	0.1%
\n	0.04%

Generating Hardware with LLMs

Verilog-specific LLMs

- DAVE - Deriving Verilog Automatically from English (2020)
  - Finetuned GPT-2 over textbook problems
  - Was good with syntax, but had no “creativity” and wouldn’t create any new variables
- VeriGen (2023)
  - Finetuned Codex model
  - Trained over a large corpus of Verilog from both GitHub and from textbooks
  - Outperformed state of the art general models at the time

Chip-Chat

- Human feedback (a “conversation”) leads to design success
- How complex a design can we go?
- Let’s try make an (8-bit) processor

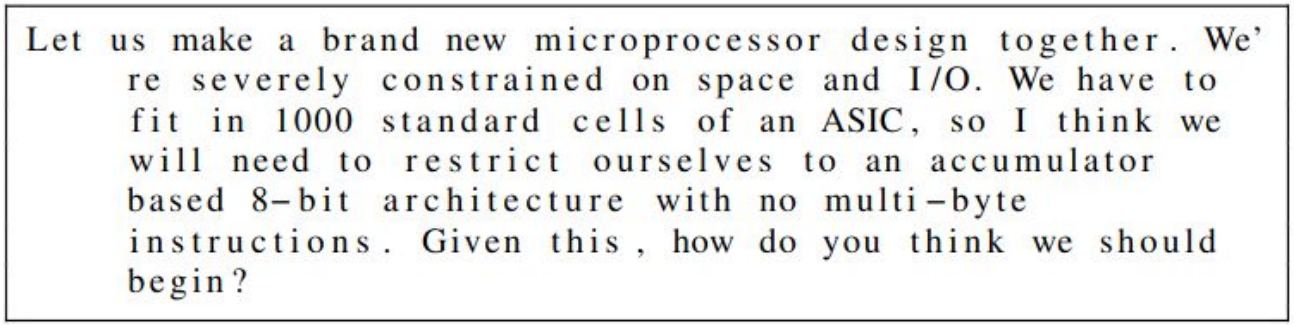
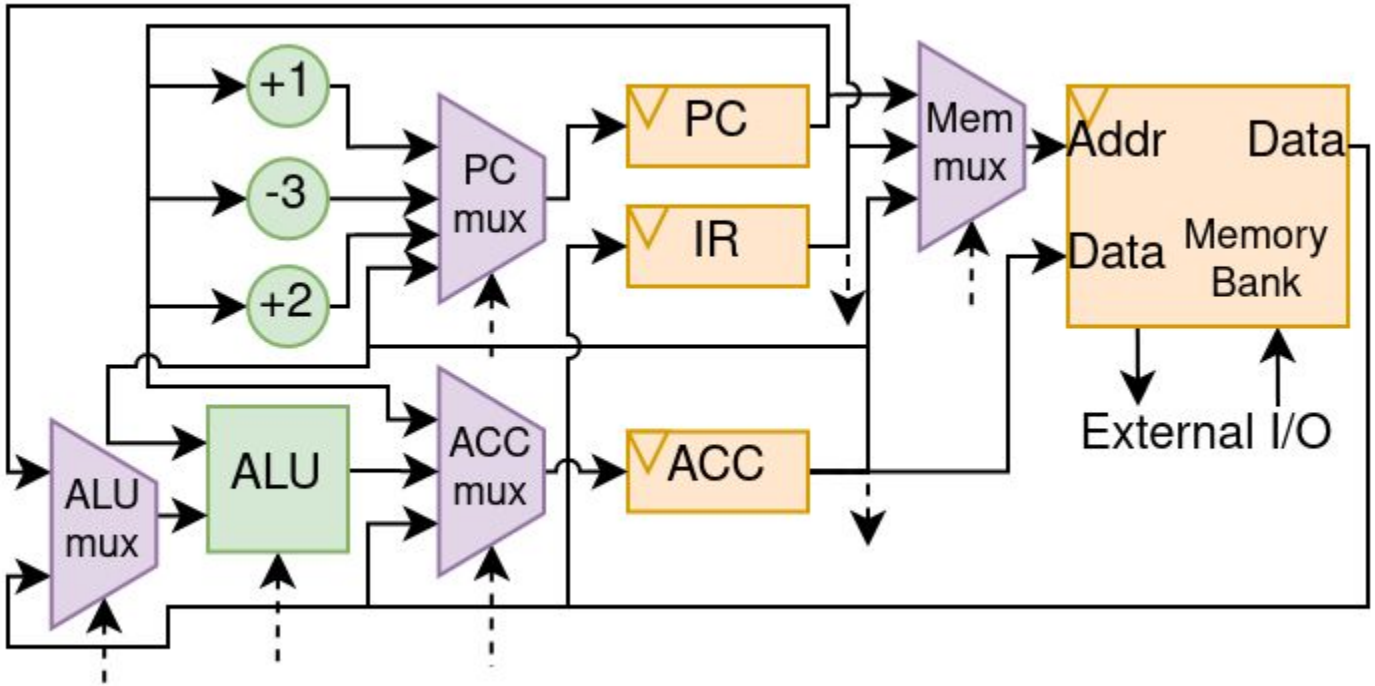
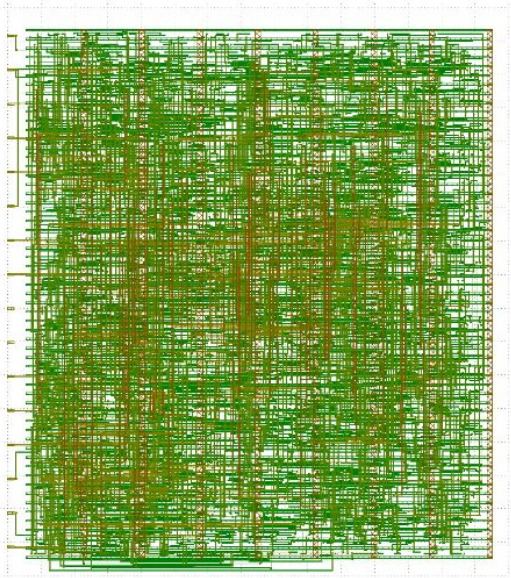


Fig. 10. 8-bit accumulator-based processor: Starting co-design prompt

Cont. T. ID	T. ID	Topic	# User Msgs	# Restart	# User Lines	# User Chars	# LLM Lines	# LLM Chars
-	00	Specification	22	10	45	5025	498	44818
-	01	Register specification	6	2	59	4927	91	9961
-	02	Shift registers and memory	5	5	65	5444	269	9468
-	03	Multi-cycle planning and ALU	7	2	103	7284	243	10148
-	04	Control signal planning	13	21	216	9205	414	20364
-	05	Control Unit state logic	12	11	216	9898	742	21663
-	06	ISA to ALU opcode	4	0	72	4576	149	5624
-	07	Control unit output logic	11	6	266	8632	518	19180
-	08	Datapath components	12	0	144	5385	516	15646
-	09	Python assembler	3	4	127	4231	218	6270
00	10	Spec. branch update	1	1	14	1275	15	1635
07	11	Control Unit branch update	2	2	98	3743	101	3969
08	12	Datapath branch update	2	0	25	888	20	726
11	13	Control Unit bug fixing	6	1	190	5413	241	8001
-	14	Memory mapped components	7	0	79	3079	516	16237
-	15	Shift Register bug fix	2	0	38	985	85	2593
12	16	Datapath bug fixing & updates	6	0	116	2979	128	4613
14	17	Memory mapped constants	4	0	21	849	101	4655
03	18	ALU optimization	1	0	2	98	32	1368
TOTALS			125	65	1896	83916	4897	206939


+1
-3
+2
PC mux
PC
IR
Mem mux
Addr
Data
Data
Memory Bank
External I/O
ALU mux
ALU
ACC mux
ACC

- What we have done up to this point is “chip-chat”, talking with the LLMs to generate 100% of our Verilog
- Is this practical?
  - Scalability concerns
  - Third party models
  - Human component
  - Correctness?



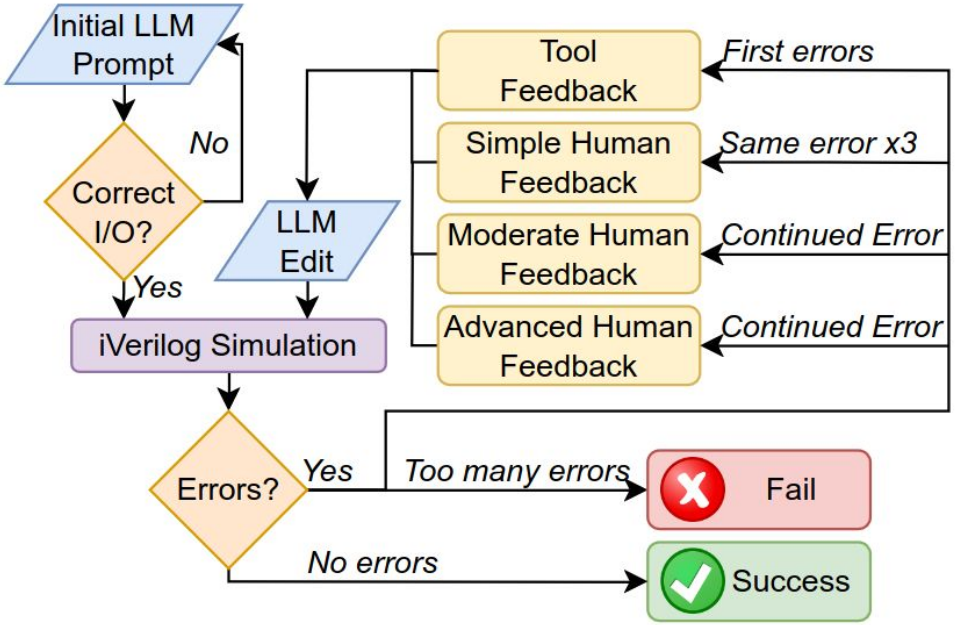
Component	Count
Comb. Logic	999
Diode	4
Flip Flops	168
Buffer	126
Tap	300

Above: (a) Components.

Left: (b) Final processor GDS render by ‘klayout’, I/O ports on left side, grid lines = 0.001 um.

Evaluating LLMs via “Script”-ed design flows

Structured conversation flow

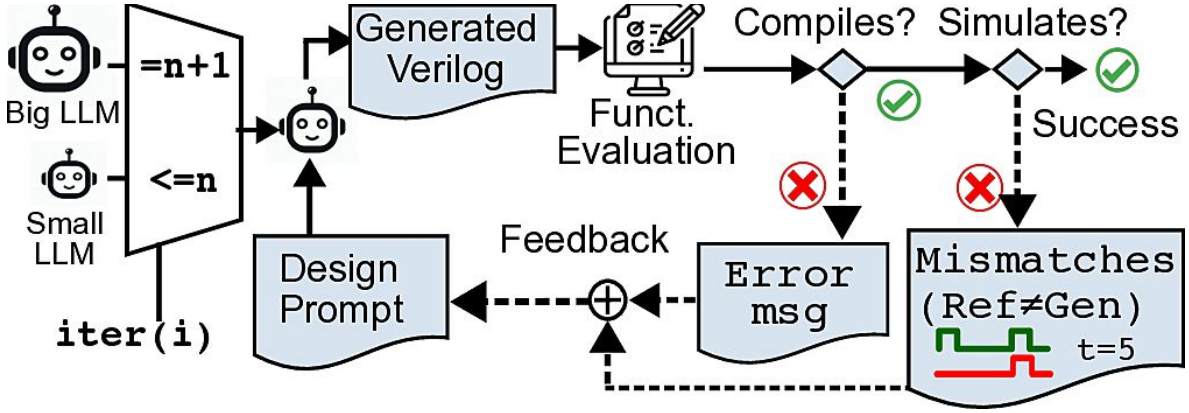


J. Blocklove, S. Garg, R. Karri, and H. Pearce, “Chip-Chat: Challenges and Opportunities in Conversational Hardware Design,” Sep. 2023, doi: https://doi.org/10.1109/mlcad58807.2023.10299874.

Benchmark	Test Set	ChatGPT-4			ChatGPT-3.5		
		Outcome	Compliant	# Messages	Outcome	Compliant	# Messages
Shift Register	T1	TF	✓	3	SHF	✓	13
	T2	TF	✓	9	FAIL	-	25
	T3	AHF	✓	15	FAIL	-	11
Sequence Gen.	T1	AHF	✓	14	FAIL	-	25
	T2	TF	✓	4	FAIL	-	7
	T3	AHF	✓	20	FAIL	-	25
Sequence Det.	T1	FAIL	-	24	FAIL	-	21
	T2	SHF	✓	9	SHF	✗	8
	T3	TF	✓	13	SHF	✗	8
ABRO	T1	FAIL	-	16	FAIL	-	25
	T2	AHF	✓	20	MHF	✓	15
	T3	TF	✓	12	NFN	✗	3
LFSR	T1	TF	✓	12	FAIL	-	25
	T2	SHF	✓	7	TF	✓	4
	T3	SHF	✓	9	FAIL	-	11
Binary to BCD	T1	TF	✓	4	SHF	✗	8
	T2	NFN	✓	2	FAIL	-	12
	T3	SHF	✓	9	TF	✗	4
Traffic Light	T1	TF	✓	4	FAIL	-	25
	T2	SHF	✓	12	FAIL	-	13
	T3	TF	✓	5	FAIL	-	18
Dice Roller	T1	SHF	✗	8	MHF	✗	9
	T2	SHF	✓	9	FAIL	-	25
	T3	SHF	✗	18	NFN	✗	3

- ChatGPT4 Passes most tests with only tool feedback

Automating the process

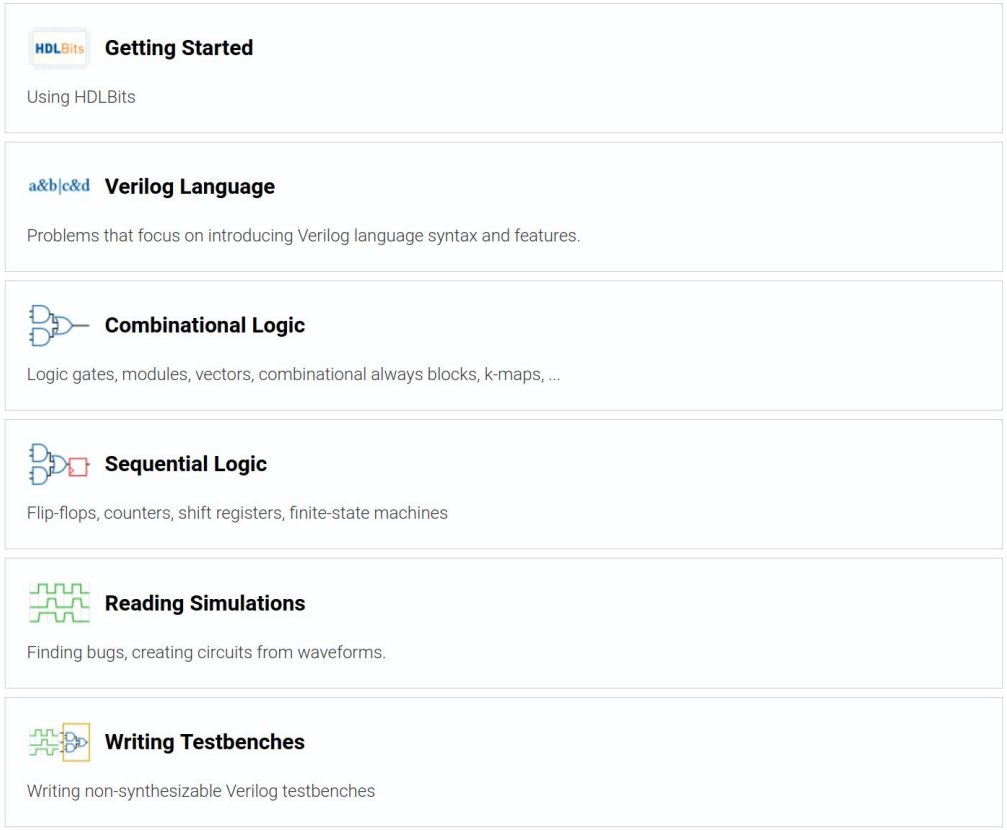


- Human feedback is nebulous and costly, can we reduce it?
- Can LLMs understand tool feedback if the testbenches are explicit enough?

S. Thakur, J. Blocklove, H. Pearce, B. Tan, S. Garg, and R. Karri, "AutoChip: Automating HDL Generation Using LLM Feedback," *arXiv.org*, Nov. 08, 2023. https://arxiv.org/abs/2311.04887 (accessed Mar. 26, 2024).

Benchmarking

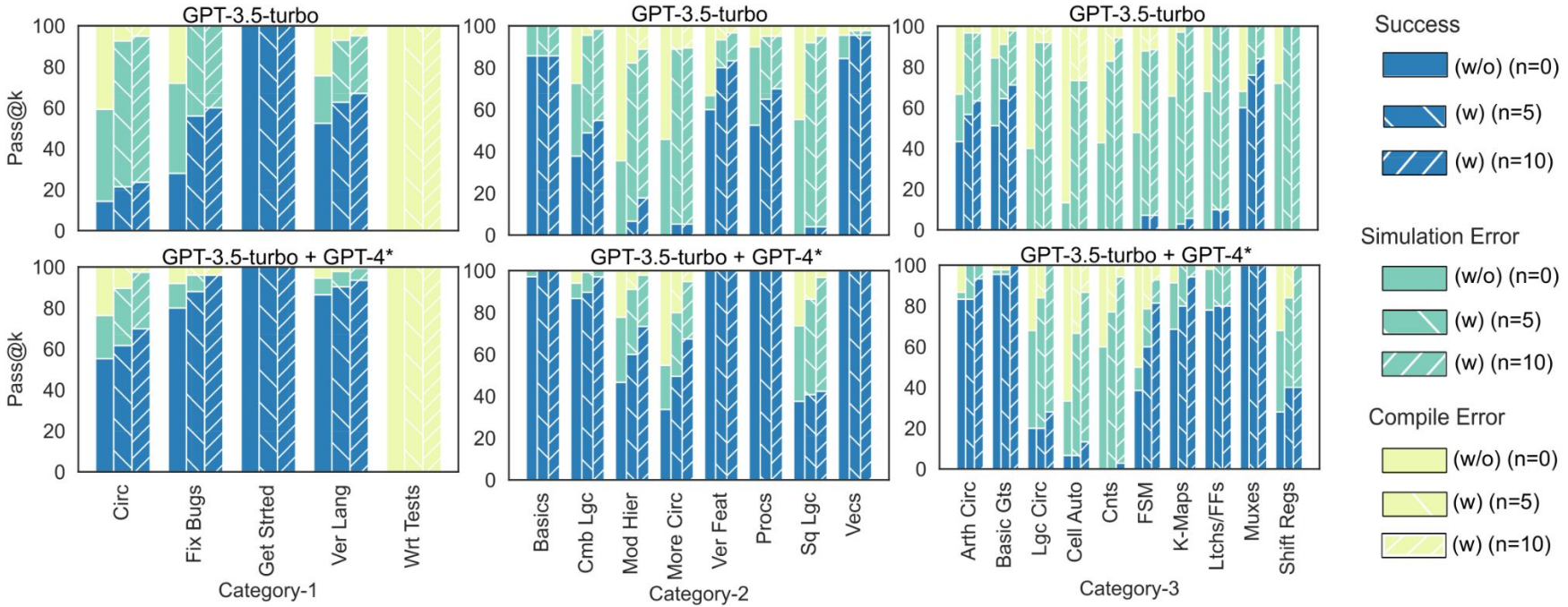
- HDLBits
  - A website for teaching Verilog
  - Selected 120 prompts from different categories



Results

Feedback Type	Metric	LLM	Success (%)				Simulation Error (%)				Compile Error (%)			
			(w/o)	(w)			(w/o)	(w)			(w/o)	(w)		
			n=0	n=1	n=5	n=10	n=0	n=1	n=5	n=10	n=0	n=1	n=5	n=10
Succinct	Pass@1	Claude 2	32.50	37.50	44.17	**47.50**	36.67	46.67	54.17	50.83	30.83	15.83	1.67	**1.67**
		GPT-3.5 (G3)	26.45	30.00	35.00	**37.50**	40.50	50.00	55.83	57.50	33.06	20.00	9.17	**5.00**
		GPT-4	60.83	69.16	**81.16**	-	19.16	18.33	12.5	-	20.0	12.5	7.3	-
		G3+GPT-4*	57.05	85.14	**87.15**	75.18	20.42	8.84	9.24	20.96	22.53	6.02	3.61	**3.86**
		CodeLlama	35.29	**36.21**	36.21	36.21	20.17	20.69	20.69	20.69	44.54	43.10	43.10	**43.10**
		CodeLlama+GPT-4*	58.25	62.53	62.53	62.53	29.21	31.41	31.41	31.41	12.52	6.05	6.05	6.05
		VeriGen	27.35	-	-	-	12.04	-	-	-	60.60	-	-	-
	Pass@5	Claude 2	32.83	38.58	45.35	**47.38**	40.83	48.39	50.42	50.08	26.33	13.03	4.23	**2.54**
		GPT-3.5 (G3)	27.27	31.17	36.00	**39.00**	37.69	49.33	55.50	54.67	35.04	19.50	8.50	**6.33**
		GPT-4	63.16	70.40	**84.45**	-	19.00	21.9	11.53	-	17.83	7.68	**4.00**	-
		G3+GPT-4*	81.06	65.39	72.84	**89.19**	7.49	24.14	22.94	7.77	11.45	10.46	4.23	**3.04**
		CodeLlama	34.29	35.71	**36.59**	36.59	18.82	21.43	22.47	22.47	46.89	42.86	**40.94**	40.94
		CodeLlama+GPT-4*	70.30	74.50	74.75	74.75	20.63	21.37	21.16	21.16	9.03	4.11	4.07	4.07
		VeriGen	27.82	-	-	-	10.02	-	-	-	62.16	-	-	-
Full	Pass@1	Claude 2	31.67	33.33	41.23	**42.11**	36.67	56.14	54.39	54.39	31.67	10.53	4.39	**3.51**
		GPT-3.5	26.67	30.25	34.45	**36.13**	33.33	43.70	53.78	52.94	40.00	26.05	11.76	**10.92**
	Pass@5	Claude 2	32.50	36.71	42.48	**44.23**	38.67	48.95	51.57	50.70	28.83	14.34	5.94	**5.07**
		GPT-3.5	28.00	30.47	34.51	**36.36**	35.67	48.82	56.06	55.39	38.33	20.71	9.43	**8.25**

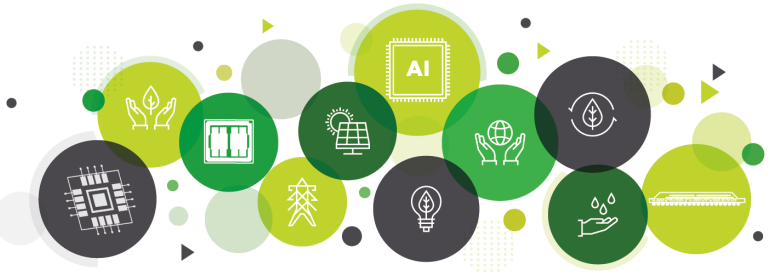
Results


GPT-3.5-turbo
GPT-3.5-turbo + GPT-4*
Pass@k
0
20
40
60
80
100
Circ
Fix Bugs
Get Strted
Ver Lang
Wrt Tests
Category-1
Basics
Cmb Lgc
Mod Hier
More Circ
Ver Feat
Procs
Sq Lgc
Vecs
Category-2
Arth Circ
Basic Gts
Lgc Circ
Cell Auto
Cnts
FSM
K-Maps
Ltchs/FFs
Muxes
Shift Regs
Category-3
Success
(w/o) (n=0)
(w) (n=5)
(w) (n=10)
Simulation Error
(w/o) (n=0)
(w) (n=5)
(w) (n=10)
Compile Error
(w/o) (n=0)
(w) (n=5)
(w) (n=10)

Thank you to our sponsors!

MEPTEC
THE NEXT GENERATION

ASE



ASE

ASE is enabling the heterogeneous integration and chiplets era through VIPack™ while delivering sustainable advanced packaging innovations for...

AI | HPC | Data Center | Automotive | 6G | IoT | and more.

@aseglobal @aseglobal aseglobal.com

COPYRIGHT NOTICE

www.meptec.org

MEPTEC
THE NEXT GENERATION