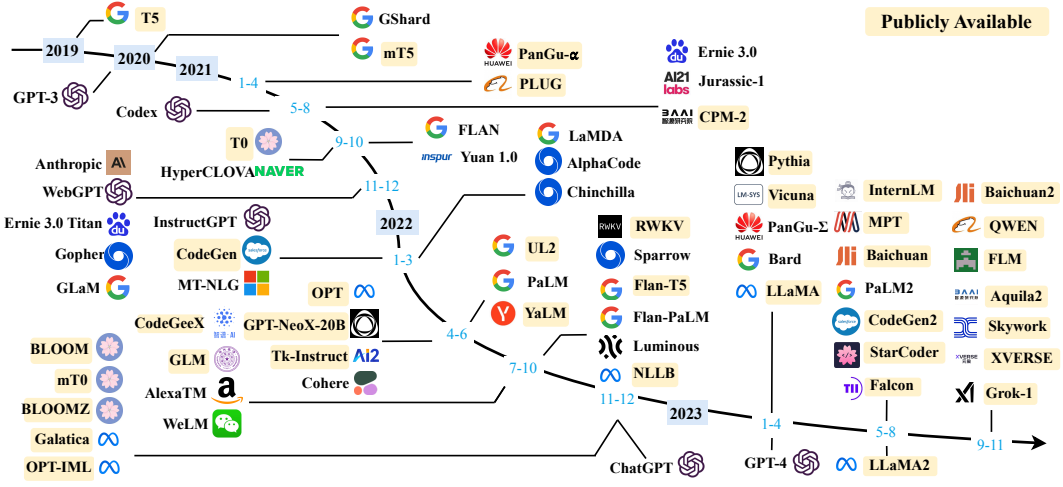
香港中文大學
The Chinese University of Hong Kong

Large Language Model in EDA

Bei Yu
Department of Computer Science & Engineering
Chinese University of Hong Kong
byu@cse.cuhk.edu.hk

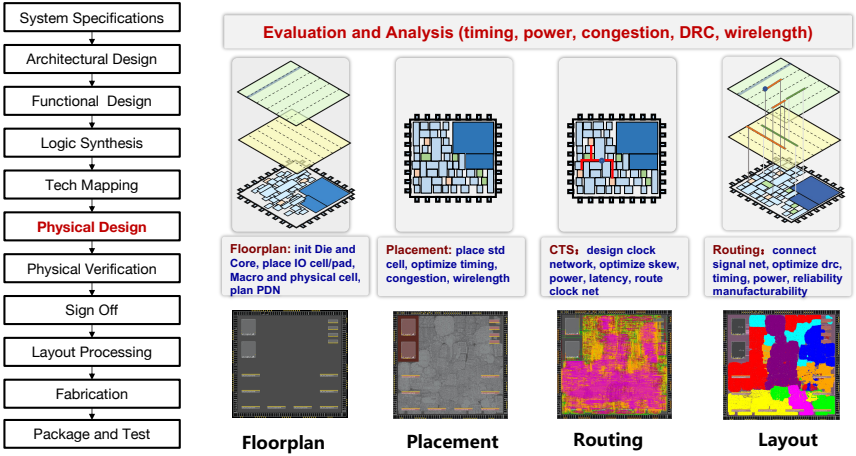
LLM Development Timeline[1]



Wayne Xin Zhao et al. (2023). "A survey of large language models". In: *arXiv preprint arXiv:2303.18223*.

Complicated Circuit Design and EDA Flow

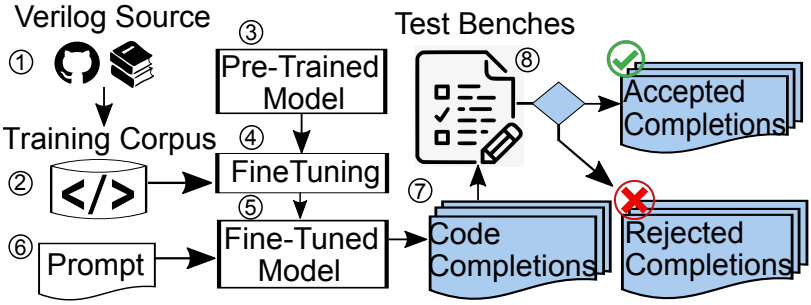
A typical EDA flow...[2]



[2]Xingquan Li et al (2024). "iEDA: An Open-Source Intelligent Physical Implementation Toolkit and Library". In: *Proc. ASPDAC*.

RTL Automatic Generation

RTL Code Generation [DATE 2023][3]

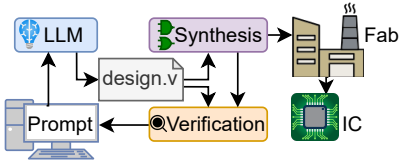


Experimental Evaluation of LLM Verilog Completions.

Shailja Thakur et al. (2023). "Benchmarking Large Language Models for Automated Verilog RTL Code Generation". In: *Proc. DATE*, pp. 1–6.

Conversational Hardware Design [MLCAD 2023][4]

Interactive design approach:

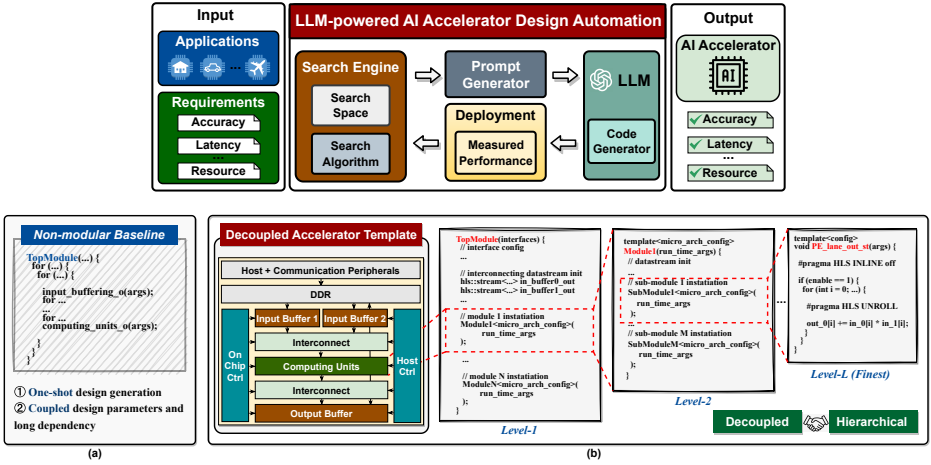


Cont. T. ID	T. ID	Topic	# User Msgs	# Restart	# User Lines	# User Chars	# LLM Lines	# LLM Chars
-	00	Specification	22	10	45	5025	498	44818
-	01	Register specification	6	2	59	4927	91	9961
-	02	Shift registers and memory	5	5	65	5444	269	9468
-	03	Multi-cycle planning and ALU	7	2	103	7284	243	10148
-	04	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

Why multiple conversation threads:
LLMs have a fixed-size context window
$\Longrightarrow$ breakup the larger design into subtasks

Jason Blocklove et al. (2023). "Chip-Chat: Challenges and Opportunities in Conversational Hardware Design". In: *Proc. MLCAD*.

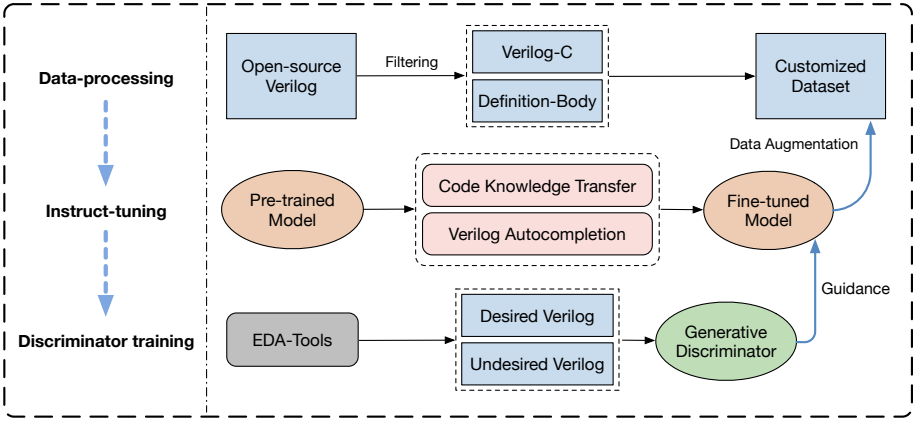
AI Accelerator Design [ICCAD 2023][5]



- Reduce code size
- Utilize hierarchy, easier fine-tuning

Yonggan Fu et al. (2023). "GPT4AIGChip: Towards next-generation AI accelerator design automation via large language models". In: *Proc. ICCAD*.

Self-Align Verilog with C [arXiv 2024][6]

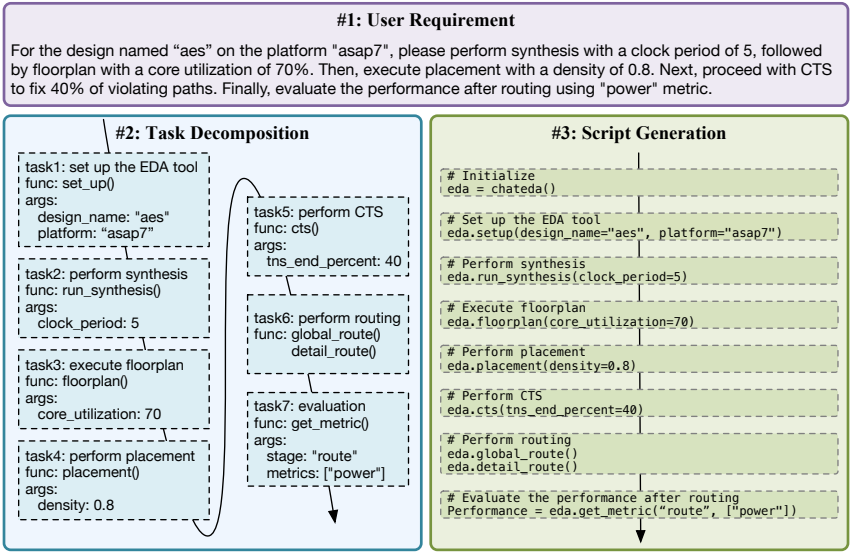


Align Verilog with C to generate self-supervised dataset.

Zehua Pei et al. (2024). "BetterV: Controlled Verilog Generation with Discriminative Guidance". In: *arXiv preprint arXiv:2402.03375*.

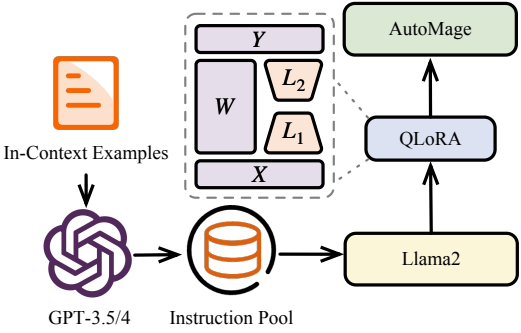
EDA Script Generation

Flow Script Generation [MLCAD 2023][7]

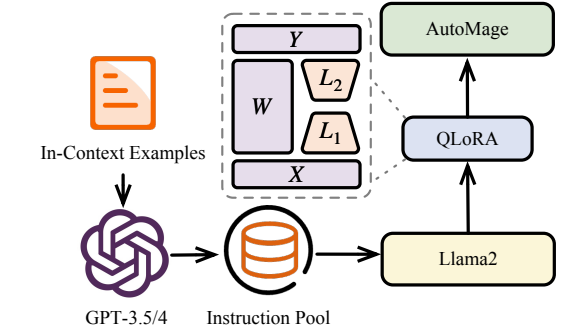


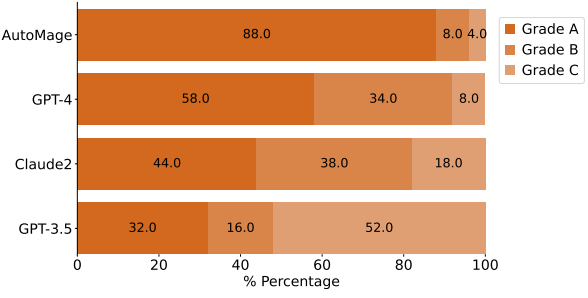
Zhuolun He et al. (2023). “ChatEDA: A Large Language Model Powered Autonomous Agent for EDA”. In: *Proc. MLCAD*.

Fine-tuning with Self-instruct



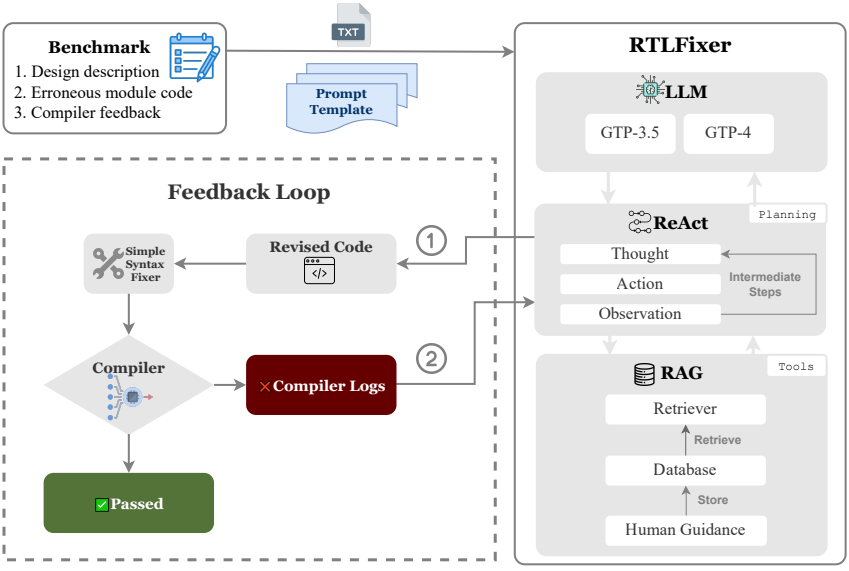
Fine-tuning with Self-instruct





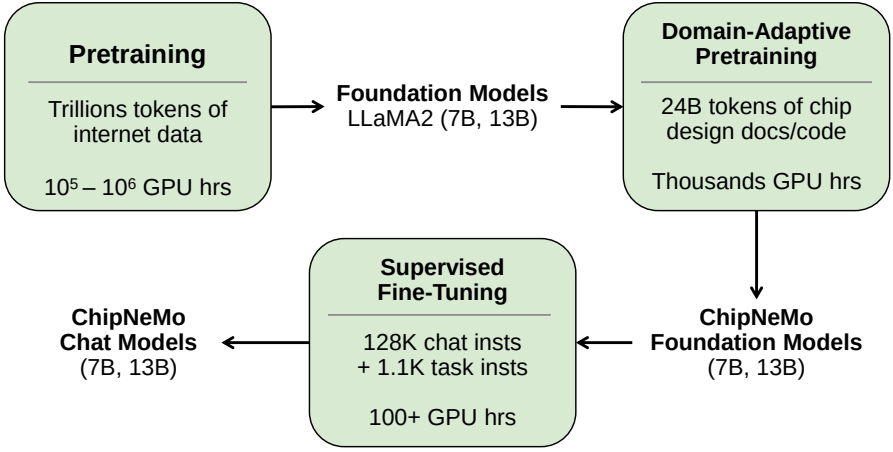
Design and EDA Flow Debug

Verilog Debugging [arXiv 2023][8]



YunDa Tsai, Mingjie Liu, and Haoxing Ren (2023). "RTLFixer: Automatically Fixing RTL Syntax Errors with Large Language Models". In: *arXiv preprint arXiv:2311.16543*.

Hardware-related Domain Adaptive Pretraining (DAPT):



Mingjie Liu et al. (2023). "ChipNeMo: Domain-Adapted LLMs for Chip Design". In: *arXiv preprint arXiv:2311.00176*.

ChipNemo Applications

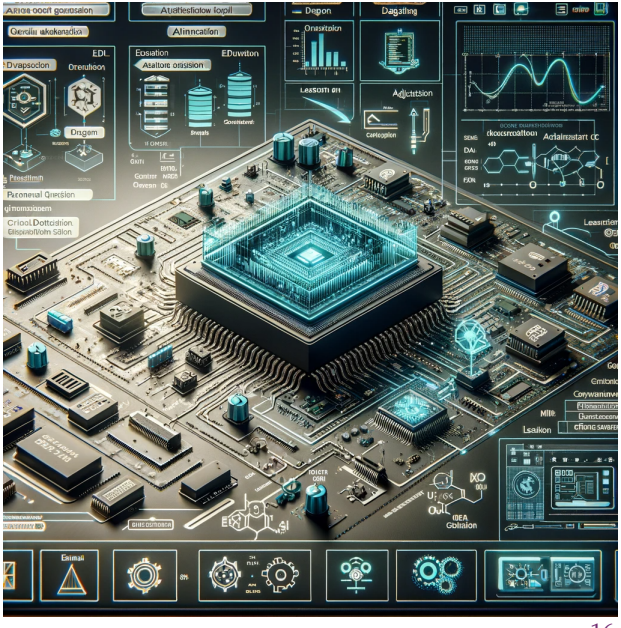
- Q & A: Engineering Assistant Chatbot
  - Motivation: Internal studies have shown that up to 60% of a typical chip designer's time is spent in debug or checklist related tasks: spec, testbench, architecture, tools, infrastructure, $\cdots$
- Code generation: EDA Script Generation
  - Motivation: Learning libraries, navigating tool documentation, writing and debugging scripts, can take up a significant amount of engineering time.
- Analysis & report: Bug Summarization and Analysis
  - Motivation: Engineering managers spend a lot of time reviewing internal issue tracking databases to build understanding of the state of the project and help speed their execution.
- Triage (future work)

Summary: Applications of LLM

- RTL Automatic Generation
- EDA Tool Script Generation
- RTL&EDA Flow Debug

Other applications:

- Design and Tool Parameter Optimization
- Layout-level Optimization
- Education: toward Customers or Universities



Challenges: LLM + EDA

- Understand and customize RTL
- Understand and customize complicated EDA flow
- Security: How not to hallucinate
- Privacy: How not to leak IP
- Ethical use of AI

THANK YOU!