Large Language Model in EDA

Bei Yu
Department of Computer Science & Engineering
Chinese University of Hong Kong
byu@cse.cuhk.edu.hk
A typical EDA flow...²

RTL Automatic Generation
Experimental Evaluation of LLM Verilog Completions.

Interactive design approach:

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

Jason Blocklove et al. (2023). “Chip-Chat: Challenges and Opportunities in Conversational Hardware Design”. In: Proc. MLCAD.
• 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.
Align Verilog with C to generate self-supervised dataset.

EDA Script Generation
For the design named “aes” on the platform “asap7”, please perform synthesis with a clock period of 5, followed by floorplan with a core utilization of 70%. Then, execute placement with a density of 0.8. Next, proceed with CTS to fix 40% of violating paths. Finally, evaluate the performance after routing using "power" metric.

```
# Initialize
eda = chateda()

# Set up the EDA tool
eda.setup(design_name="aes", platform="asap7")

# Perform synthesis
eda.run_synthesis(clock_period=5)

# Execute floorplan
eda.floorplan(core_utilization=70)

# Perform placement
eda.placement(density=0.8)

# Perform CTS
eda.cts(tns_end_percent=40)

# Perform routing
eda.global_route()
eda.detail_route()

# Evaluate the performance after routing
Performance = eda.get_metric("route", ["power"])
```
Fine-tuning with Self-instruct

In-Context Examples

GPT-3.5/4  Instruction Pool

AutoMage

QLoRA

Llama2

Y

W

L2

L1

X
Fine-tuning with Self-instruct

- In-Context Examples
- Instruction Pool
- AutoMage
- QLoRA
- Llama2

% Percentage
GPT-3.5
Claude2
GPT-4
AutoMage
32.0
44.0
58.0
88.0
16.0
38.0
34.0
8.0
52.0
18.0
8.0
4.0

Grade A
Grade B
Grade C
Design and EDA Flow Debug
1. Design description
2. Erroneous module code
3. Compiler feedback

Feedback Loop

Compiler

Simple Syntax Fixer

Revised Code

Compilers Logs

Passed

LLM

Promt Template

GTP-3.5
GTP-4

ReAct

Planning

Thought
Action
Observation

Intermediate Steps

Tools

Retriever
Retrieve
Database
Store
Human Guidance

RAG

Retriever

Hardware-related Domain Adaptive Pretraining (DAPT):

- **Pretraining**: Trillions tokens of internet data
  - 10^5 – 10^6 GPU hrs

- **Foundation Models**
  - LLaMA2 (7B, 13B)

- **Domain-Adaptive Pretraining**
  - 24B tokens of chip design docs/code
  - Thousands GPU hrs

- **Supervised Fine-Tuning**
  - 128K chat insts + 1.1K task insts
  - 100+ GPU hrs

- **ChipNeMo Chat Models** (7B, 13B)

- **ChipNeMo Foundation Models** (7B, 13B)

---

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, · · ·

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!