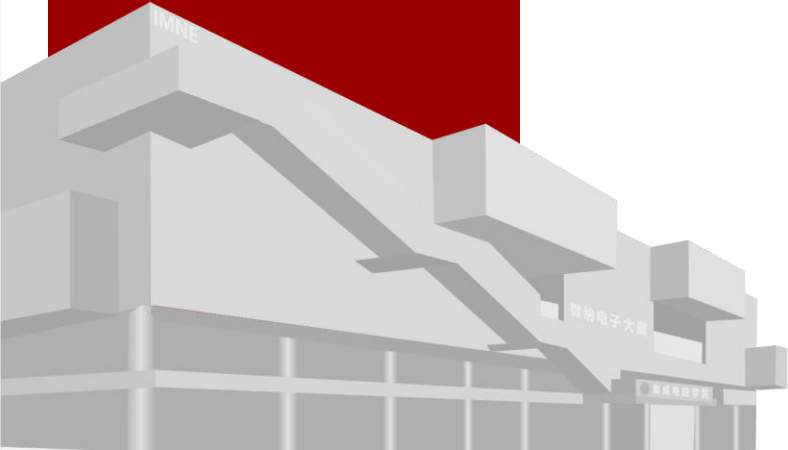


Agentic AI for Agile End-to-End SoC Design

Zhuolin Li, Zhantong Zhu, Han Liu, Tianyu Jia

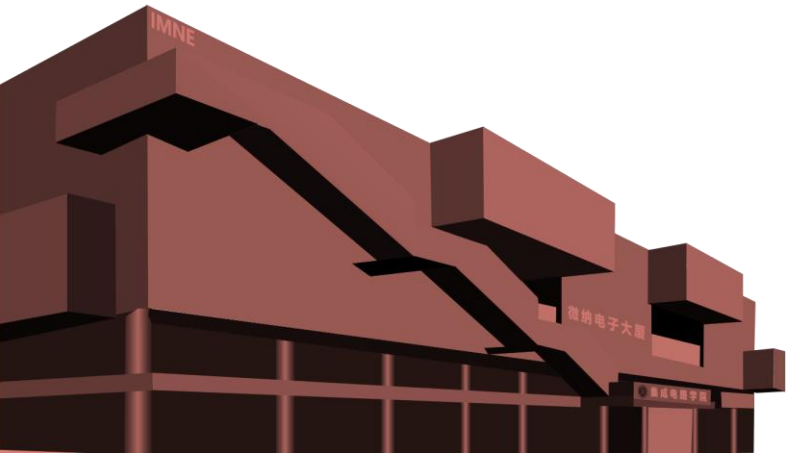
Peking University

March 23, 2026



Outline

- **Background**
- **AI for Architecture DSE**
- **AI for SoC Design**
- **Conclusion**



IC Introduction: SoC Design

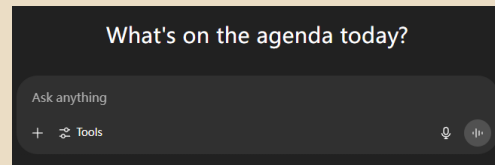


Application Domains

Image Recognition



Text Generation



Speech Detection

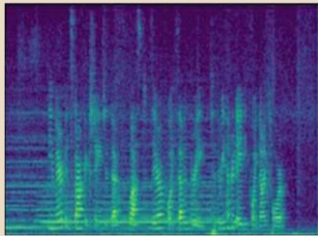
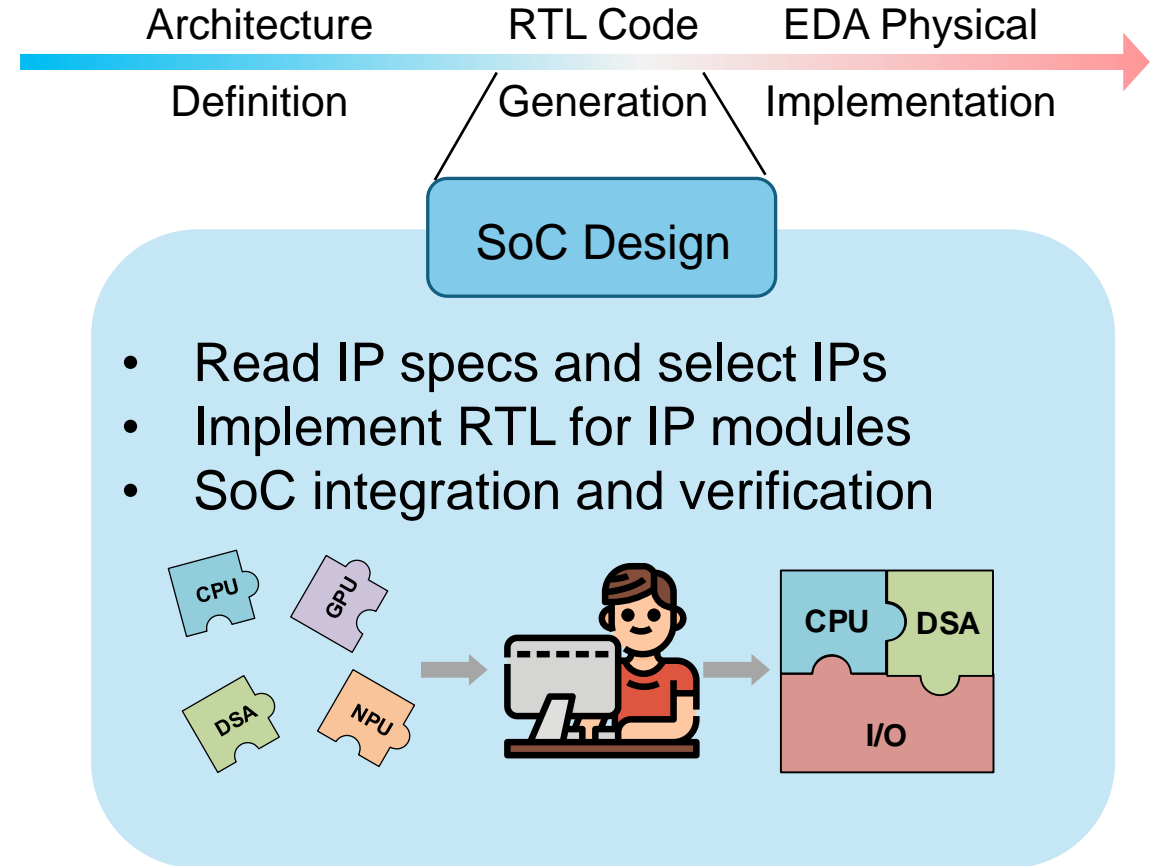


Image Generation



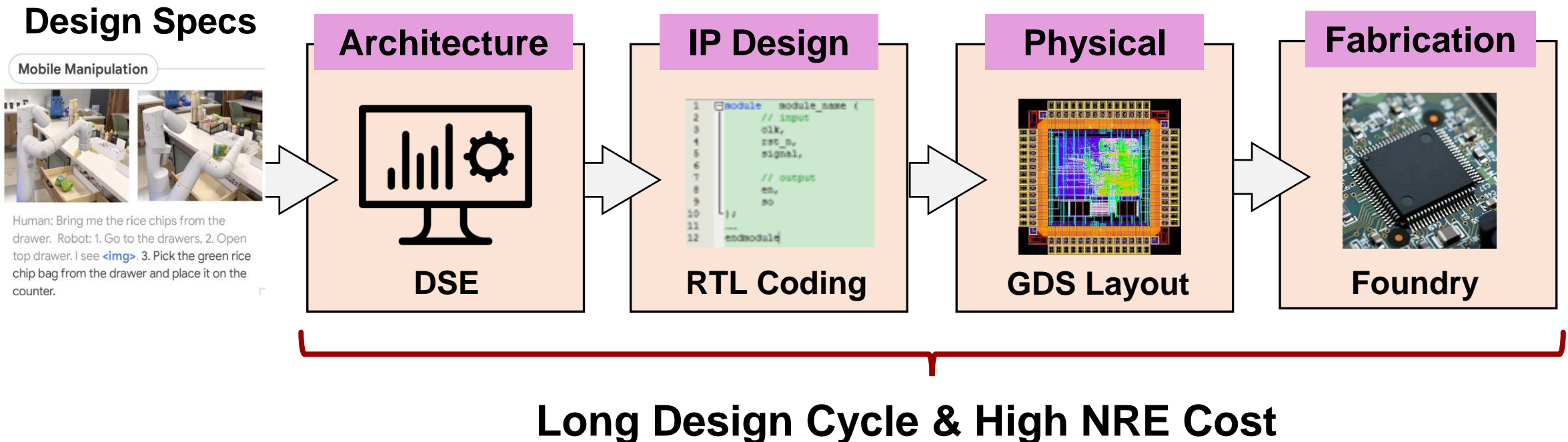
SoC Design Flow



IC Introduction: Chip Design Flow

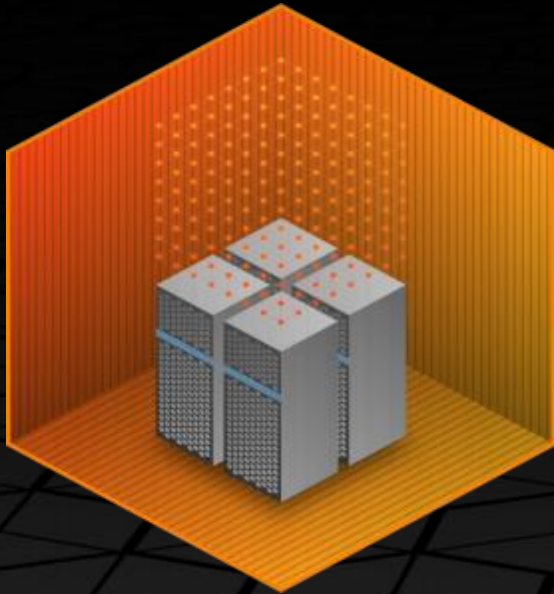


- End to end chip design contains multiple design stages
- Challenges for design chip: years design cycle + high cost



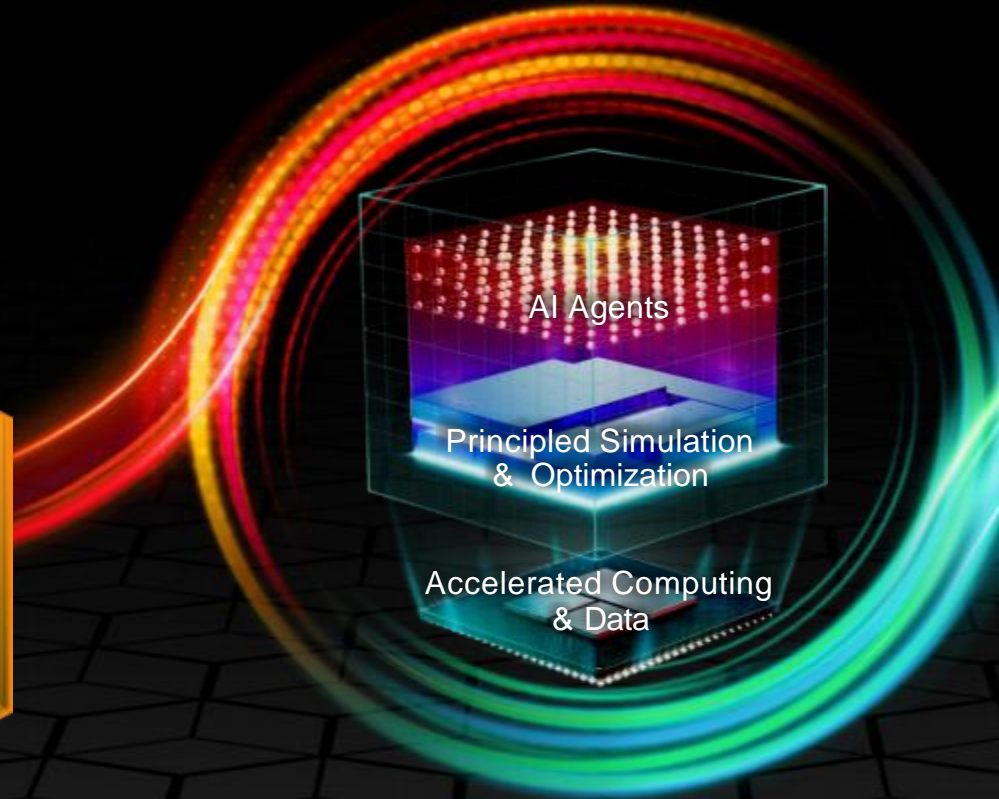
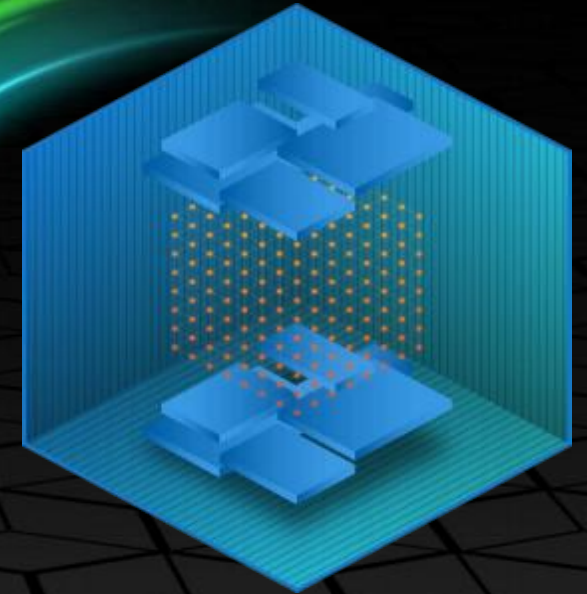
Design for AI

Building the AI infrastructure

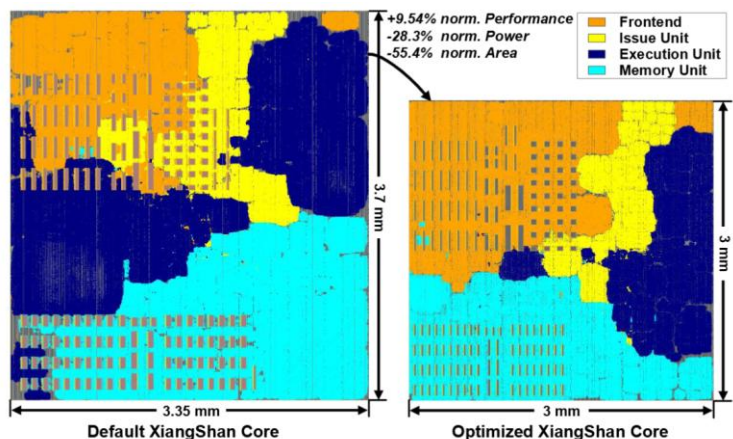


AI for Design

Using AI in our design solutions



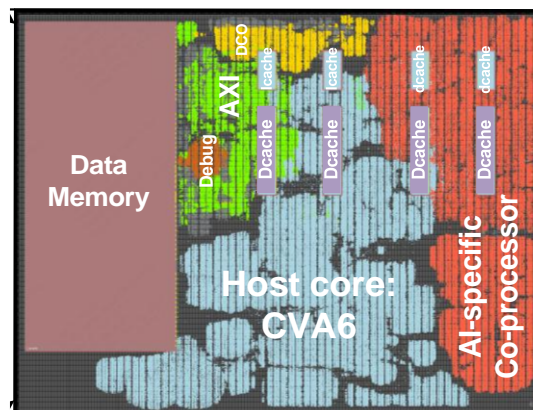
Stage 1: Architecture



Design Space
Exploration

ASP-DAC 2026
DAC 2026

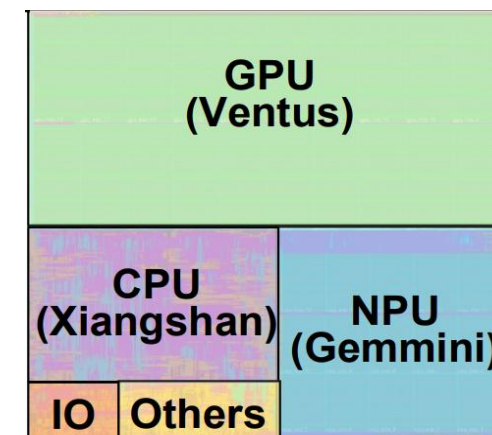
Stage 2: IP Design



RTL Coding

ISEDA 2025
Honorable Mention

Stage 3: SoC Integration

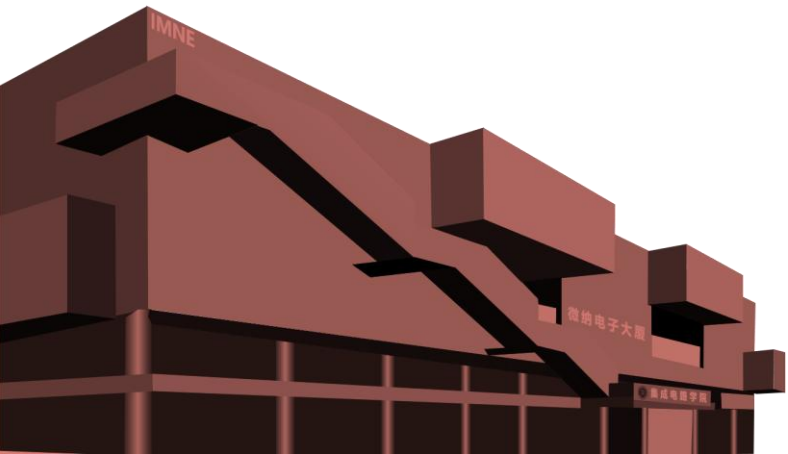


Design &
Verification

ISLPED 2025
Best Paper Award

Outline

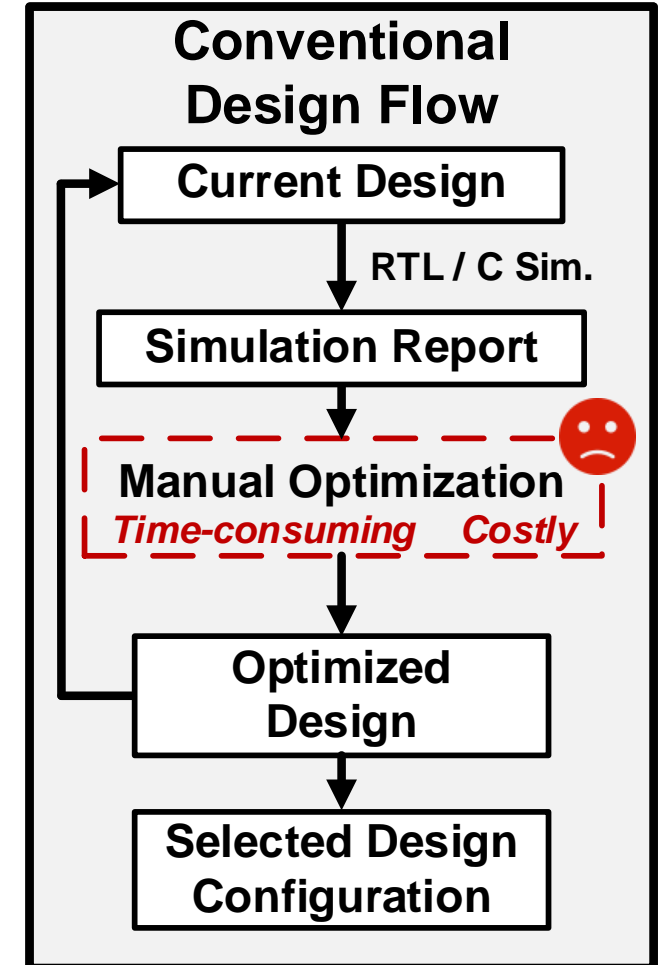
- Background
- **AI for Architecture DSE**
- AI for SoC Design
- Conclusion



IC Introduction: Modern CPU Design



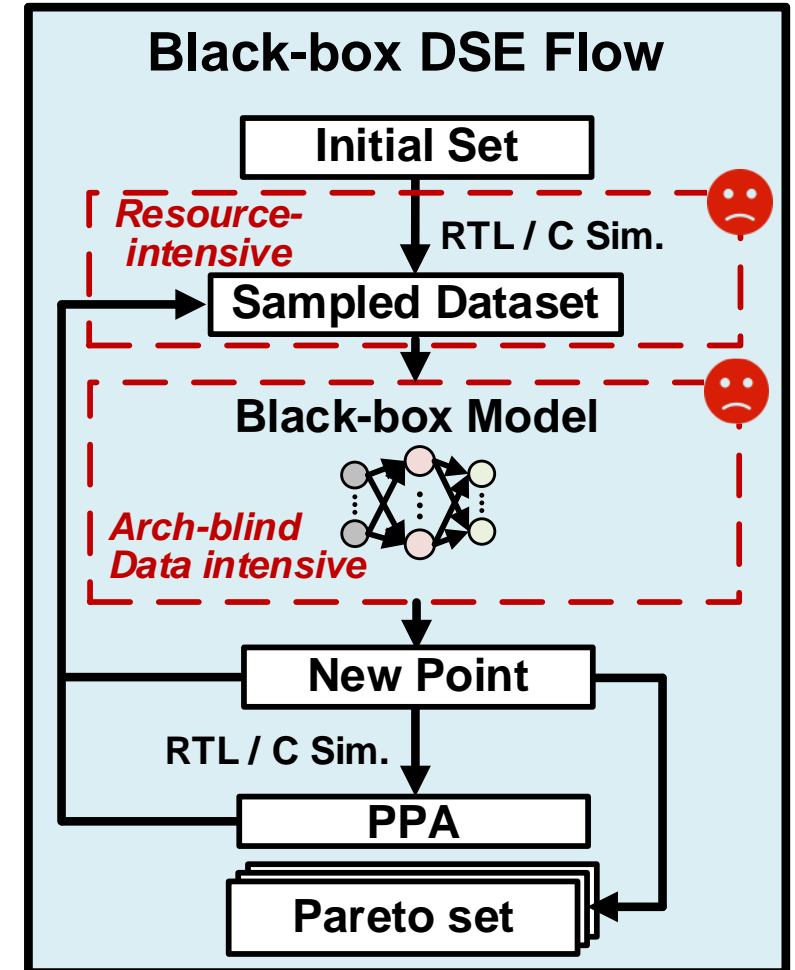
- *Increasing design complexity* of modern high-performance CPUs
- Design and optimization of CPU microarchitecture
 - Evaluate design to obtain simulation results
 - *Manual selection* of optimal parameters
 - Dependent on *engineer expertise*



IC Introduction: Black-box DSE Methods



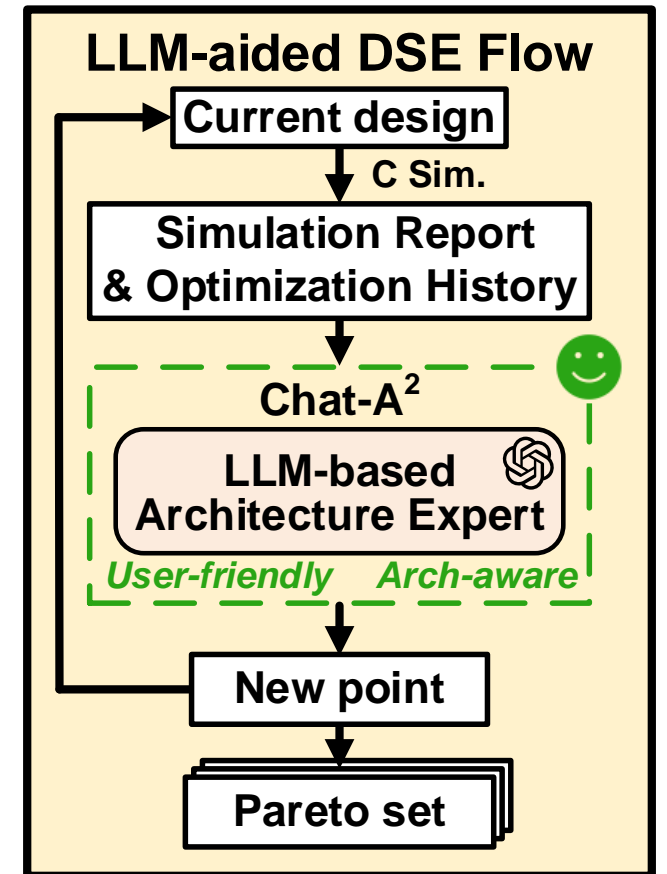
- Multi-objective *Bayesian optimization*
- Various techniques for *surrogate modeling* and *acquisition functions*
- *Algorithm-driven* optimization
- Overlook *architectural information*
- *Sampled dataset* for model training



IC Our Work: LLMs for CPU DSE



- Chat-A², an LLM-aided DSE framework
- CPU microarchitecture DSE
 - LLM-based architecture expert
 - Customized gem5 simulator and power/area evaluator



Incorporate LLM as an *architecture expert* in CPU DSE

IC Observations

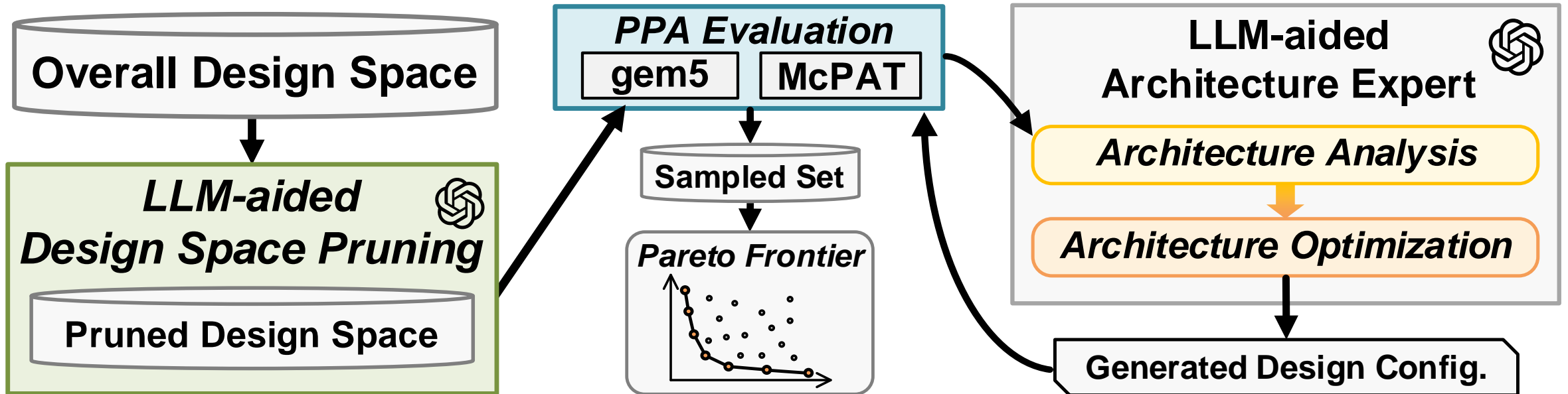


- **Observation 1:** LLMs struggle to complete *multi-step tasks* like microarchitecture DSE autonomously.
- **Observation 2:** LLMs can determine *whether parameter values should be adjusted* based on statistical reports.
- **Observation 3:** LLMs can *refer to exploration history* through in-context learning to improve current optimization.
- **Observation 4:** LLMs can *correlate different hardware parameters* with global information and intrinsic knowledge.

IC Framework Overview



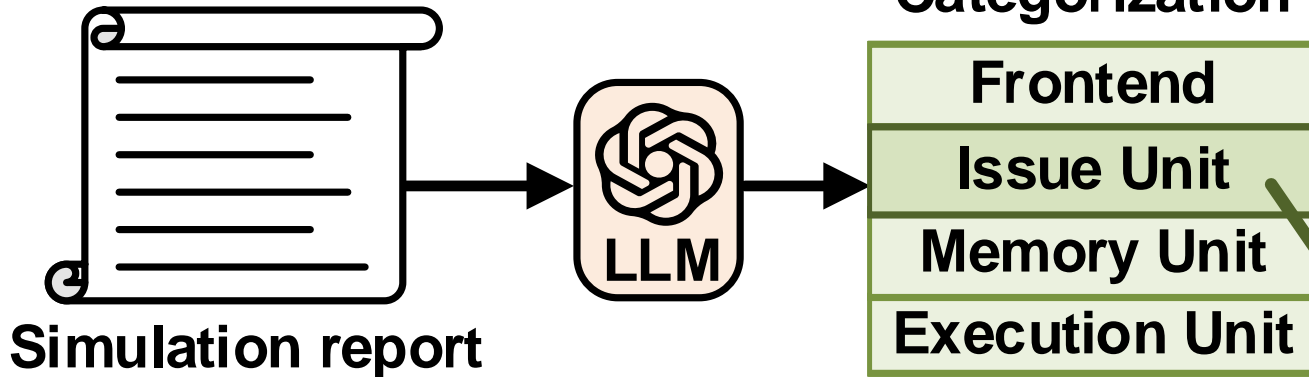
- LLM-aided *architecture analysis* and *optimization*
- PPA evaluation w/ customized gem5 and McPAT



IC Step 1: LLM-aided Architecture Analysis



- Top-down *parameter categorization*
- Identify *critical modules*
 - Simulation report
 - PPA metrics



```
{System Prompt}
Analyze the performance of Frontend, Issue, Memory, and Execution Unit based on the following simulation stats.

---- Frontend ----
- fetch.nisnDist::i
- decode.idleCycles
- decode.blockedCycles
.....

---- Issue Unit ----
..... Depend on operation mode

Report the critical modules that display
{bottlenecks, redundancies}.
```

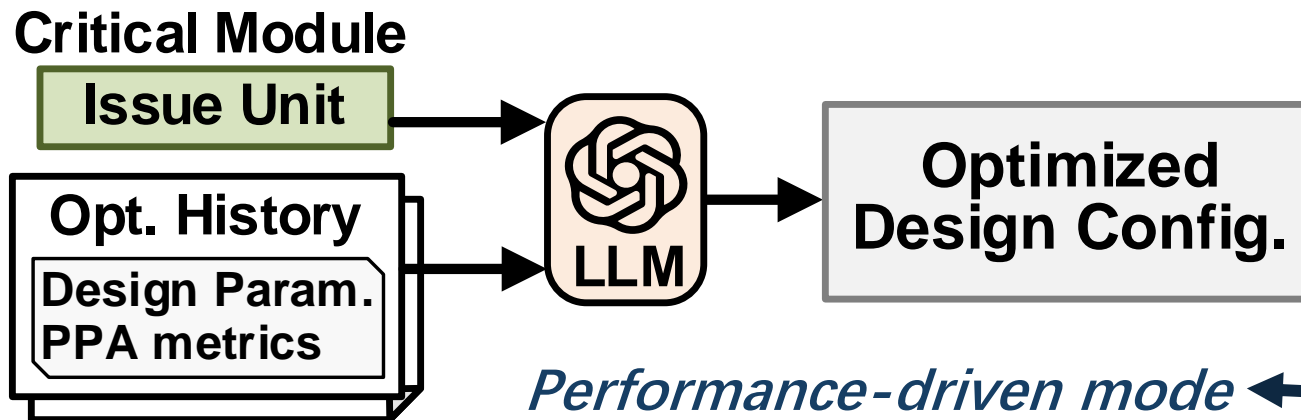
Example prompt for architecture analysis

Critical module: Identified bottleneck/redundancy

IC Step 2: LLM-aided Architecture Optimization



- **Optimize hardware parameters within the critical modules**
 - **Based on simulation reports and PPAs**
 - **Incorporate global optimization history**



```
{System Prompt}
[Issue Unit] displays significant
bottlenecks. Here are PPA metrics,
hardware parameters, and sim. stats.

CPI: 0.5169; Power: 15.15; Area: 50.47
I$ Size: 16kB , candidates: 8/32/64kB
.....
Issue Width: 4 , candidates: 5/6/7/8
ROB Entry: 96 , candidates: 128/192/256

---- Issue Unit ----
- issue.nisnDist::0          228661
- rename.ROBFullevents    4227
- rename.IQFullevents      0

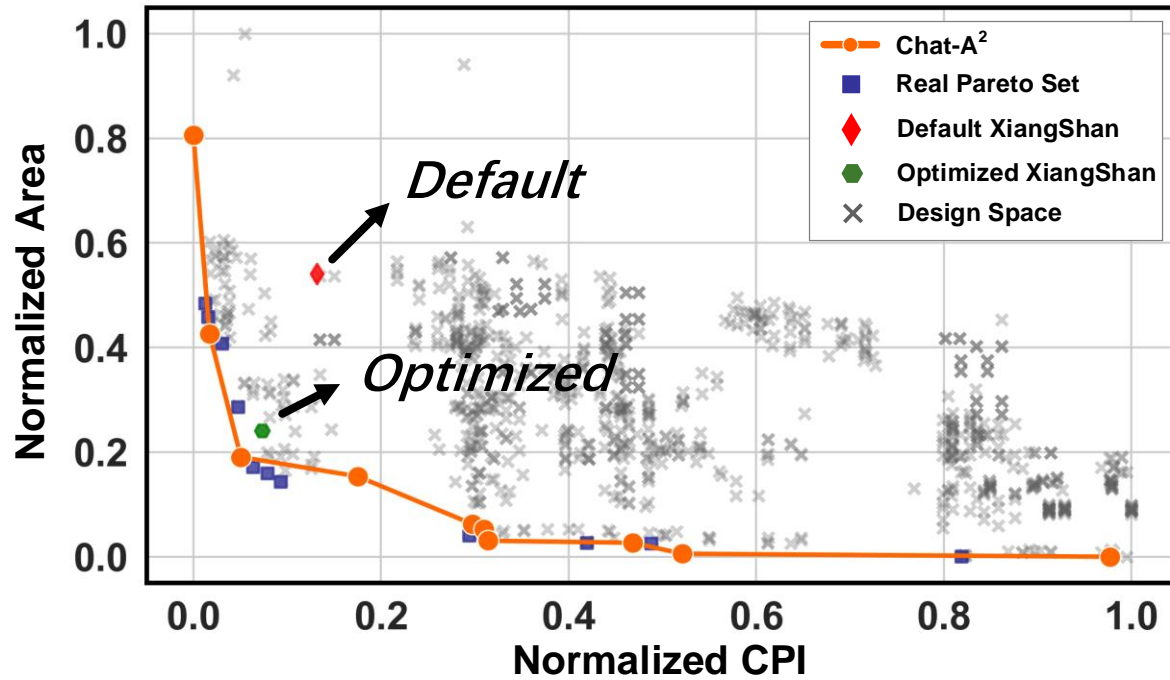
What are suggested parameter values
for improving system performance?
```

Example prompt for architecture optimization

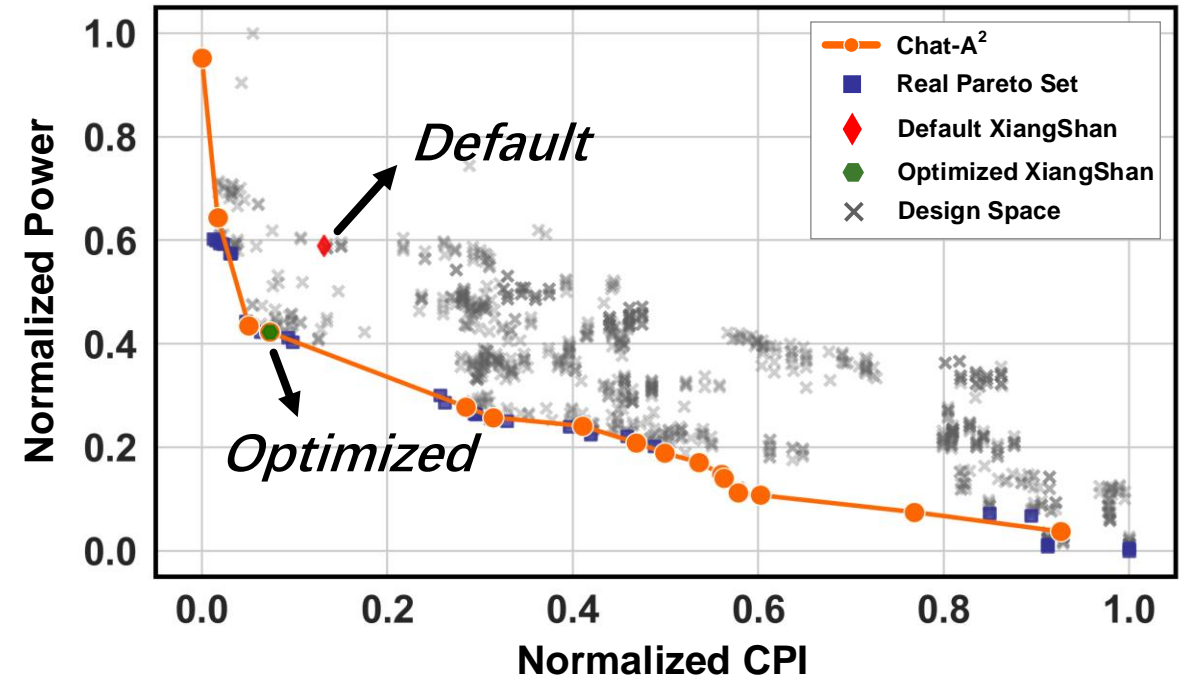
IC Exploring Pareto Optimal in Design Space



- **98.93%** and **97.88%** hypervolume in the performance-area and performance-power view of the Pareto set, respectively



Perf.-area view of learned PPA pareto front

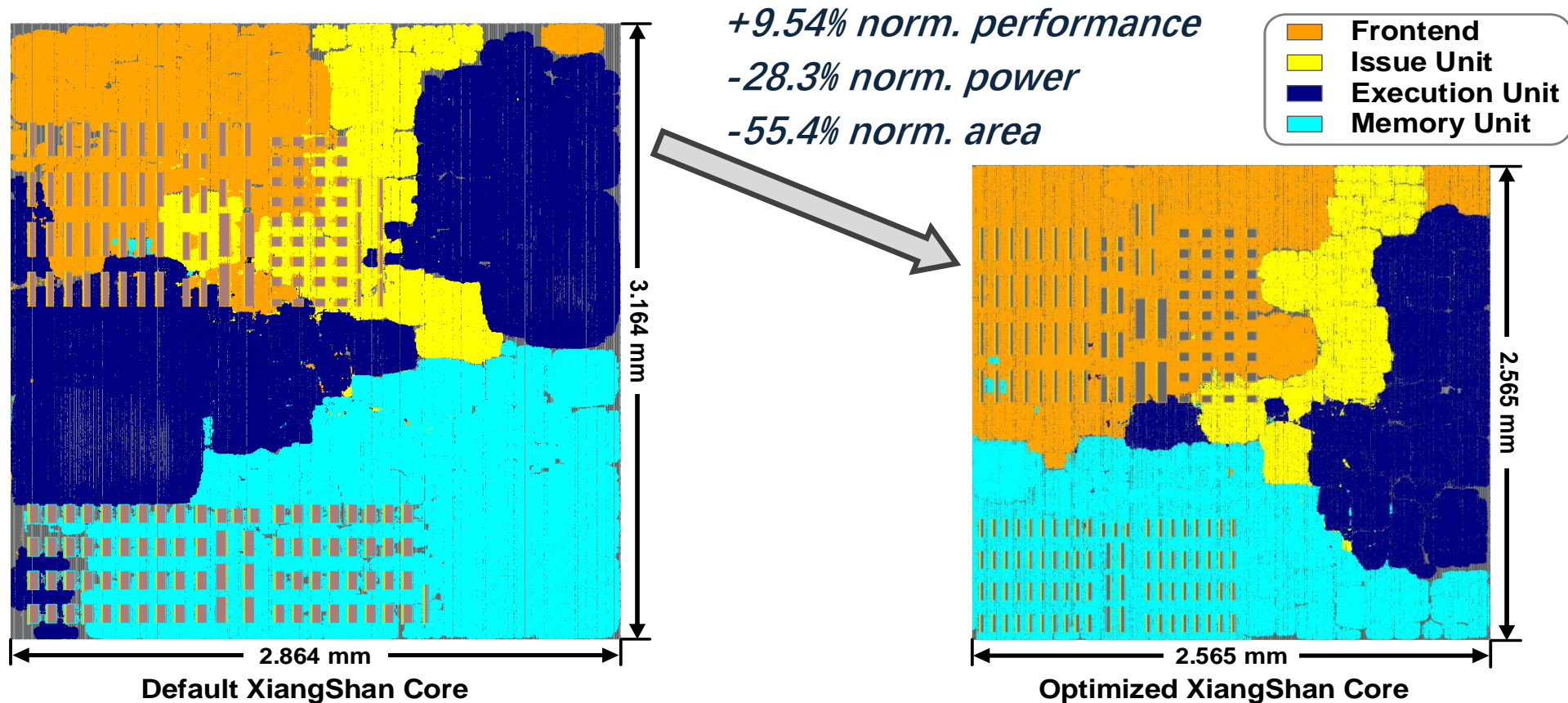


Perf.-power view of learned PPA pareto front

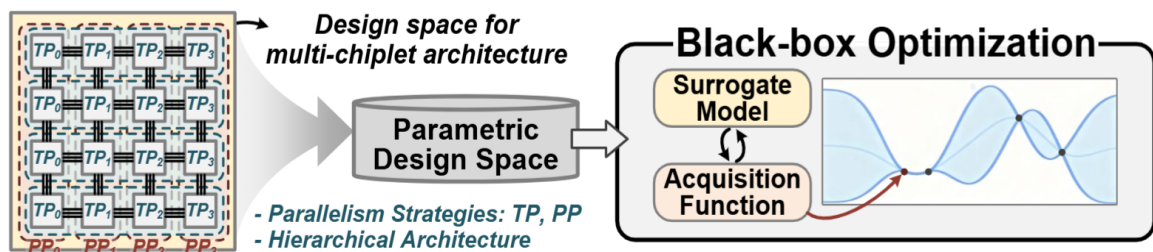
IC Optimization Benefits



- Physical implementation of optimized and default configuration



IC AgenticDSE: Agentic-Assisted DSE for Chiplets



Fixed design space with complex architecture-parallelism parameters

Black-box optimization w/o domain knowledge

(a)

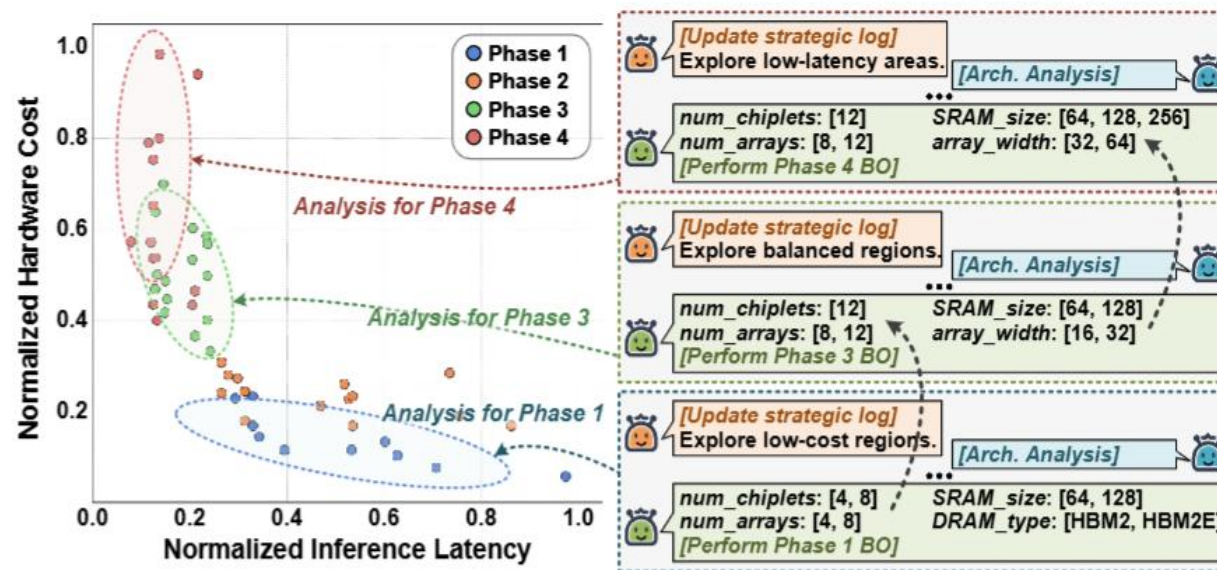
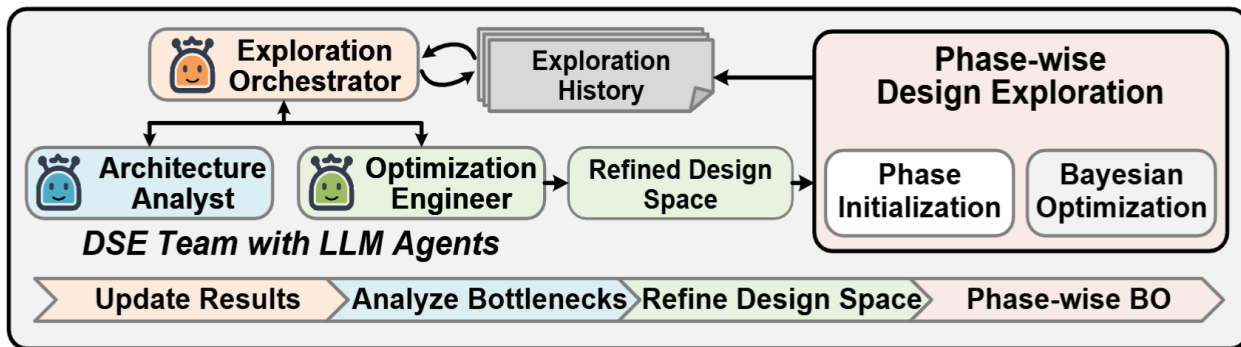
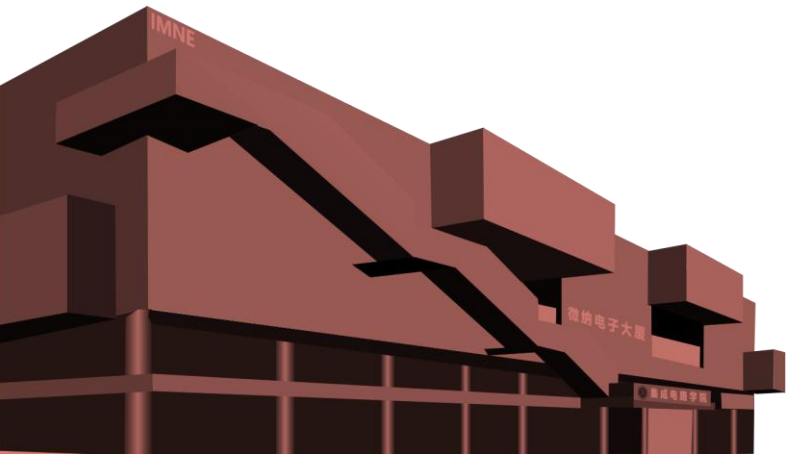


Figure 7: Optimization trajectory over exploration phases.

Outline

- Background
- AI for Architecture DSE
- **AI for SoC Design**
- Conclusion



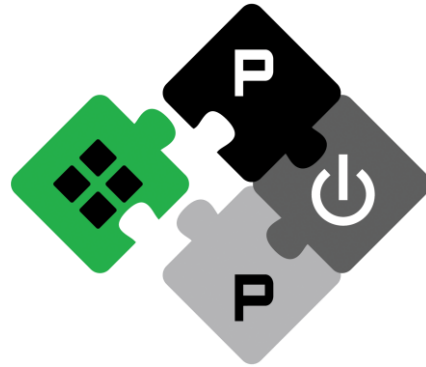
IC Introduction: OSH Platforms



- OSH SoC development platforms provide automated SoC generation based on open-source IP



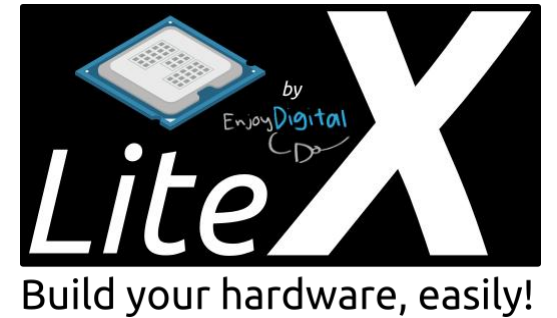
Chipyard



PULP



ESP



LiteX

IC Introduction: SoC Generation

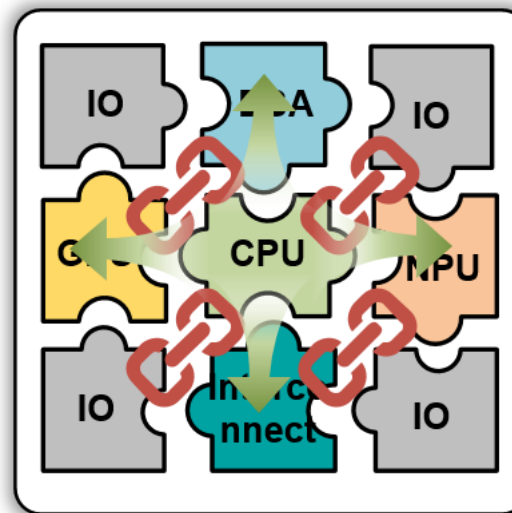


- Question: what if we want to build a more custom SoC?



```
module example(...);  
  //need comments  
  always@(...)begin  
    .....  
  end  
endmodule  
-----  
//need documents  
This module describe a ...  
This module consist of ...
```

Open-Source Hardware IP



We need to spend a lot of time and effort integrating and verifying SoC



SoC Integration and Verification



- Lack of structured organization
- Lack of documents and annotations

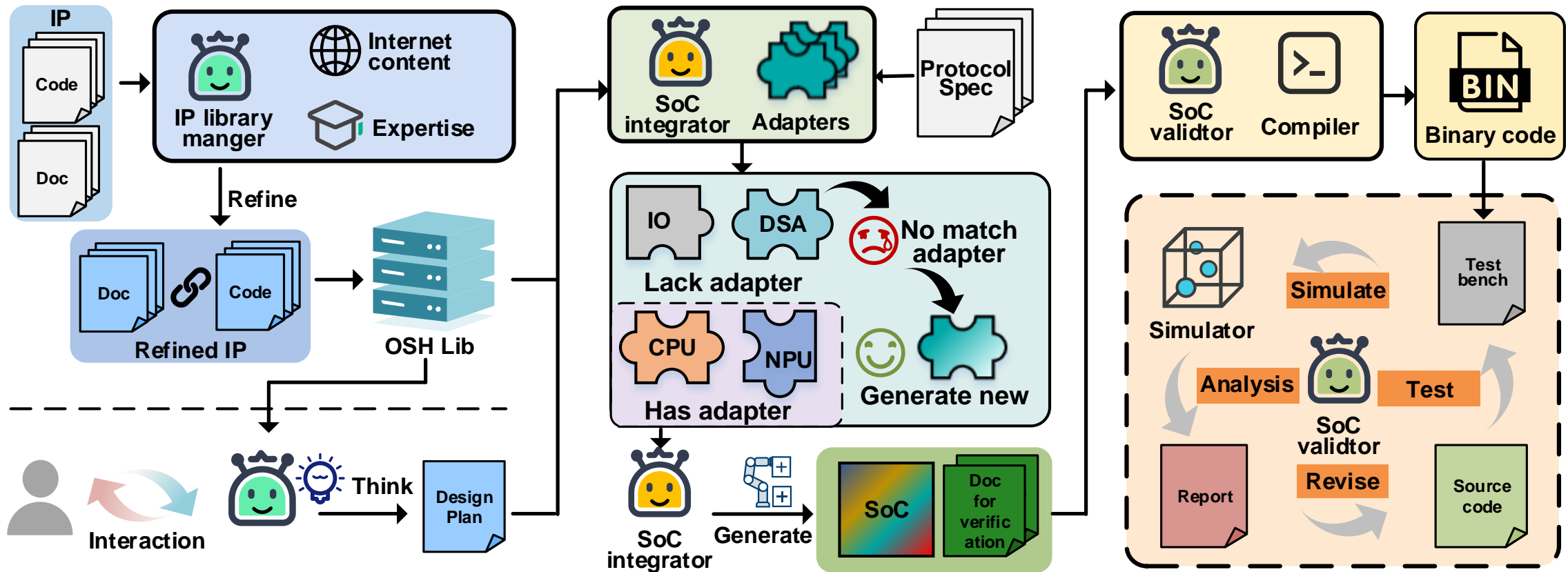


- Labor-intensive and error-prone
- Time-consuming and hard to debug

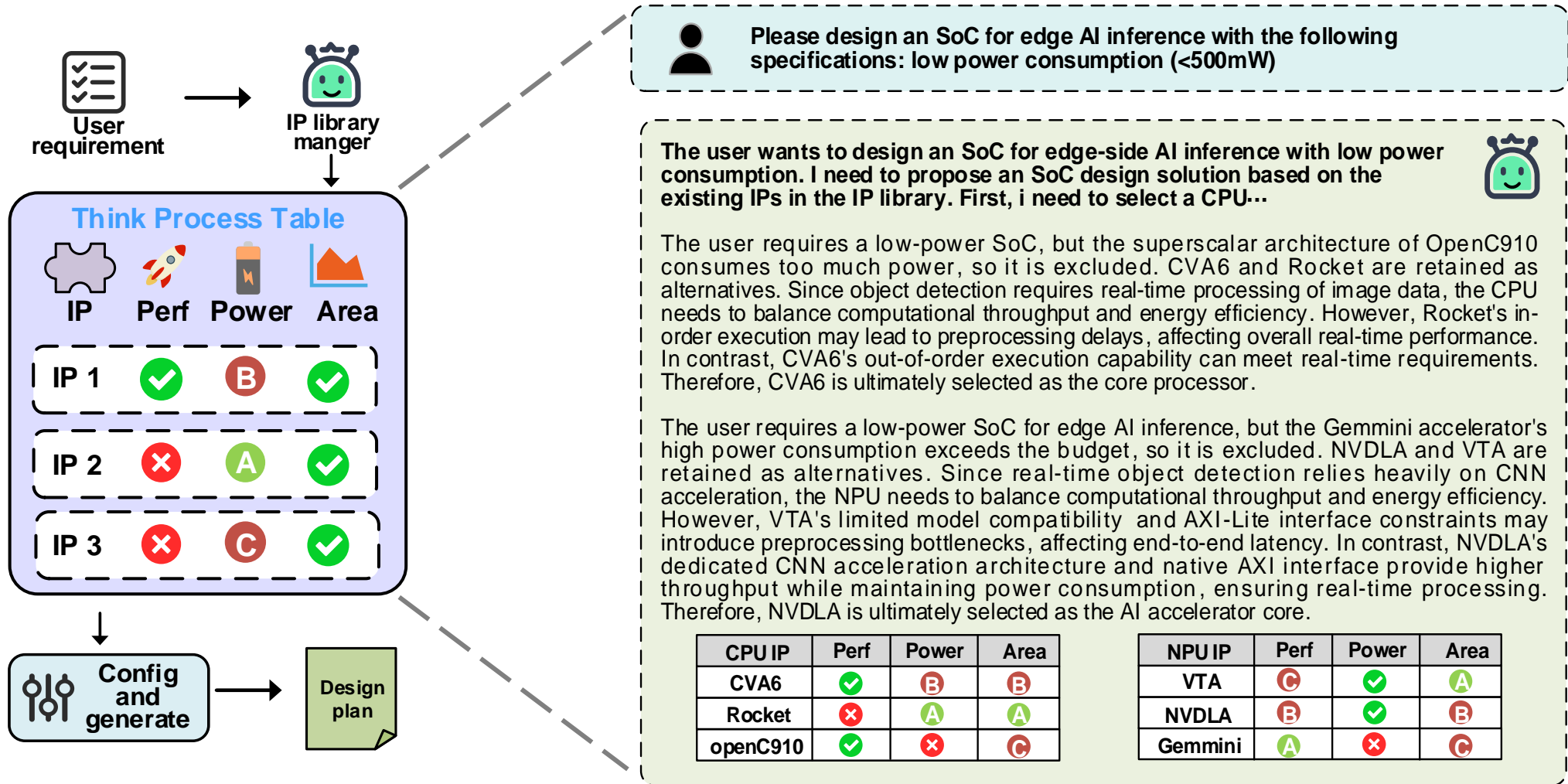
IC Our Work: GenSoC Workflow



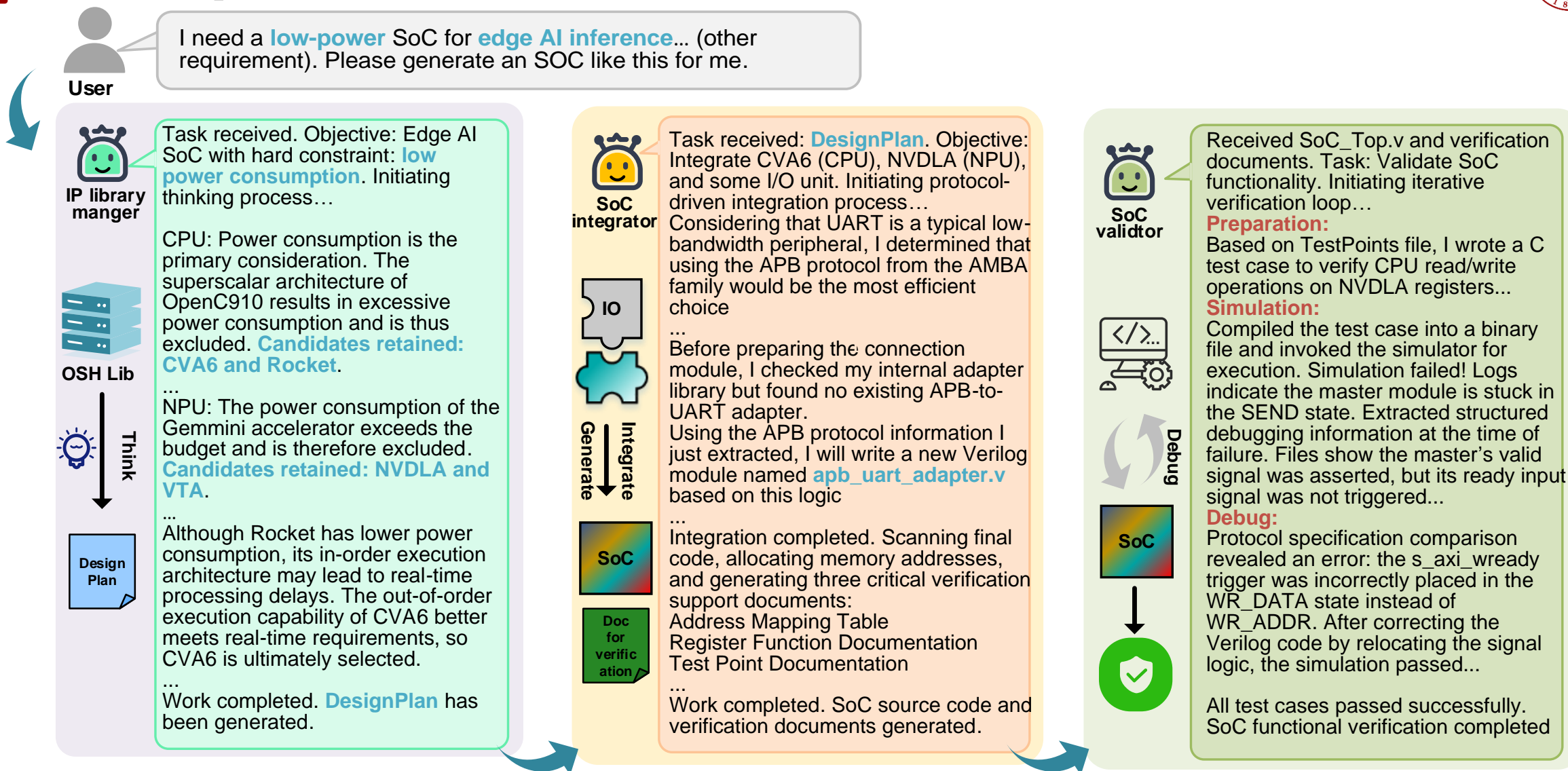
- **GenSoC: OSH and LLM agents are utilized for agile SoC design**



IC Example: IP Selection for SoC Design



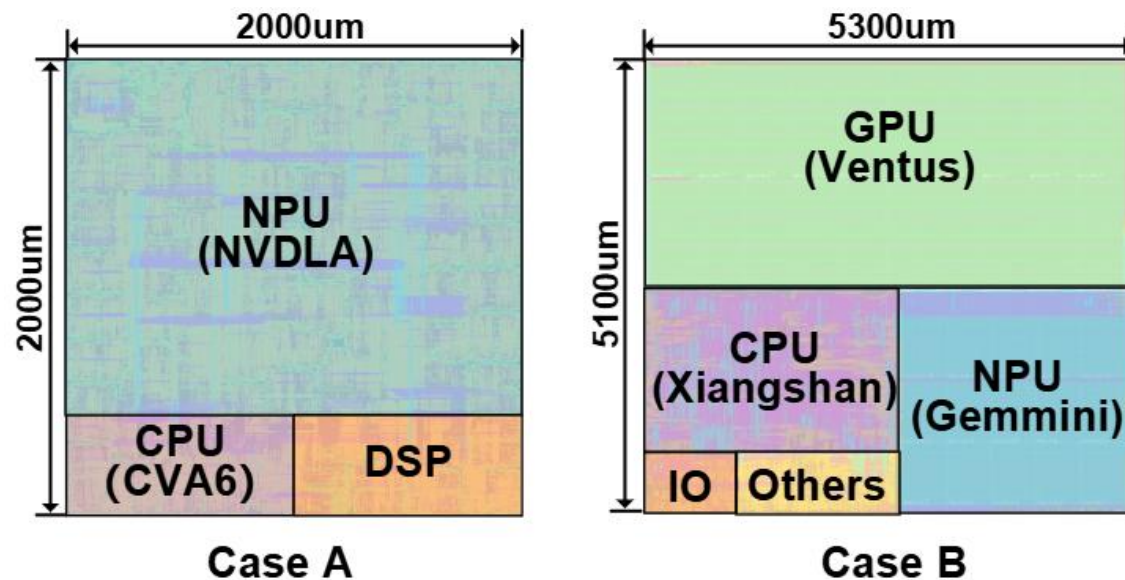
IC Example: Overall Workflow



IC Evaluation: SoC Design Examples



- Two design cases are developed by GenSoC with TSMC 22nm
- Case A: edge AI inference, Case B: general mobile computing



Feature	Case A	Case B
PDK	TSMC 22nm	TSMC 22nm
Target	Edge AI	General Mobile
Frequency	500 MHz	1 GHz
Total Power	332.4 mW	16.23 W
Total Area	4 mm ²	27.03 mm ²

- End-to-end design flow including EDA script generation

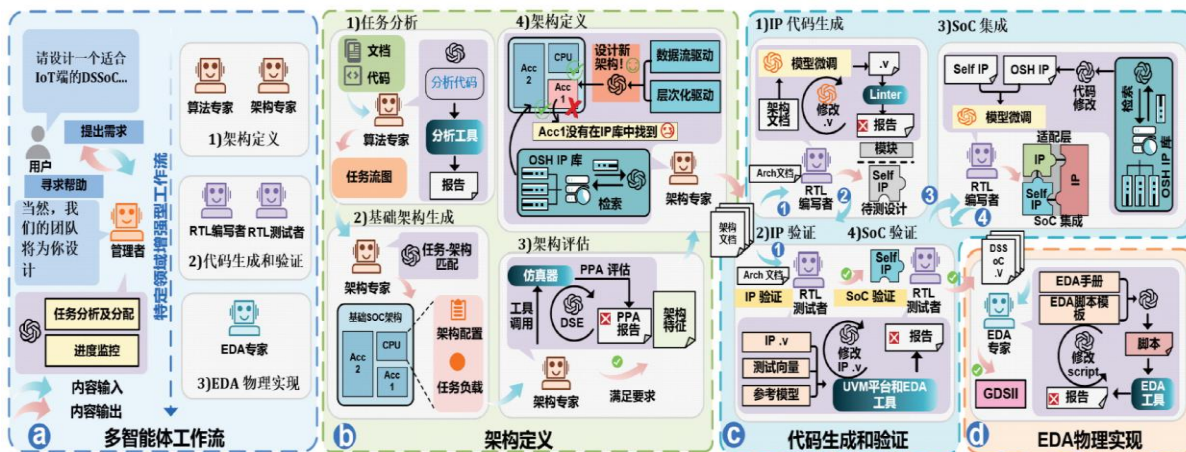
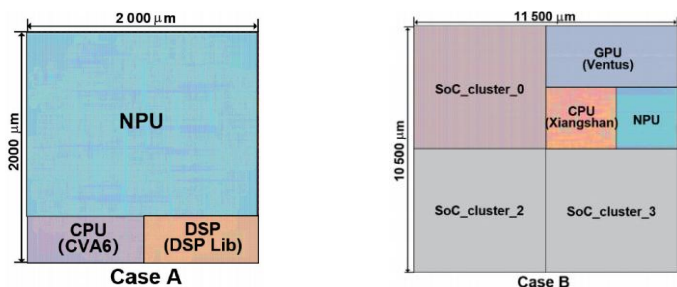
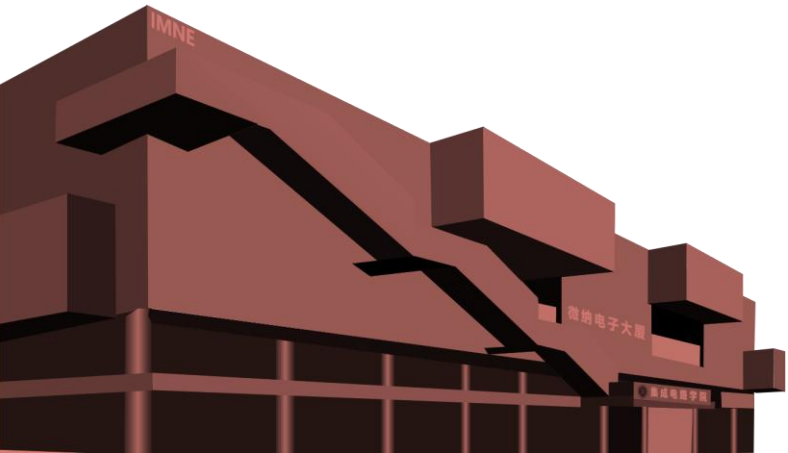


图 3 基于 LLM 多智能体系统辅助的端到端敏捷 SoC 设计流程概述



Outline

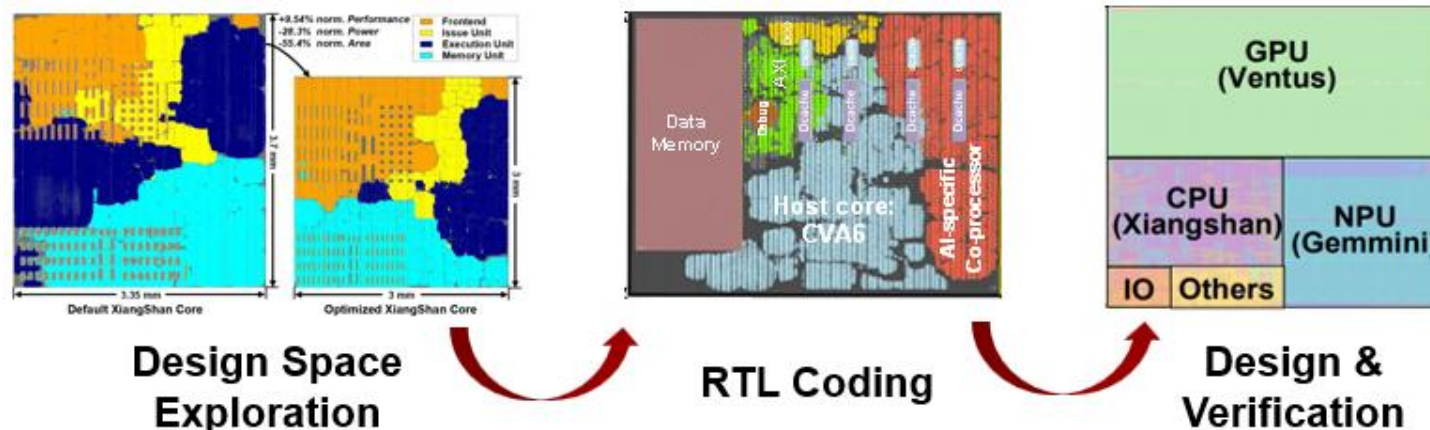
- Background
- AI for Architecture DSE
- AI for SoC Design
- **Conclusion**



IC Summary



- AI for IC design is promising to accelerate IC design cycle
- Agentic AI has great potentials from multiple aspects
 - Chat-A²: LLM-aided CPU DSE framework
 - AgenticDSE: Agentic-assisted DSE for Chiplets
 - GenSoC: End-to-end SoC design and generation
- Much more can be done for Agentic-driven design methodology





集成电路学院

Thank You

