# 大模型辅助的 RISC-V SoC敏捷设计方法探索

贾天宇

北京大学 集成电路学院

2025年7月

# Outline

集成电路学院

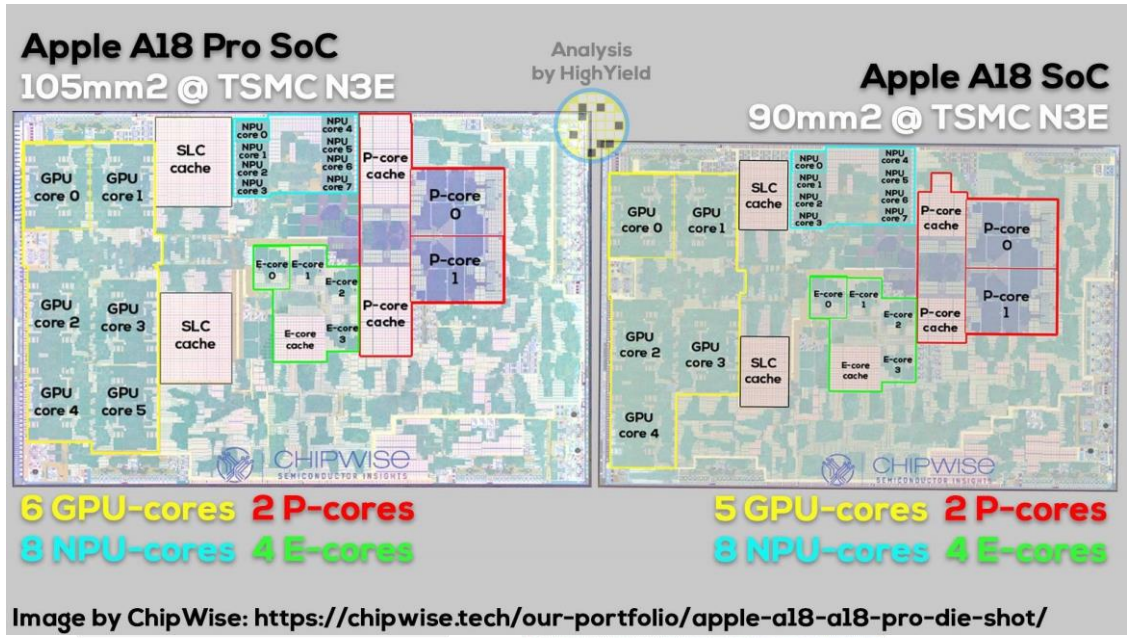
- 研究背景
- 大模型辅助RISC-V SoC设计
  - 设计空间探索
  - 代码生成
  - SoC集成
- 小结

# 传统处理器设计

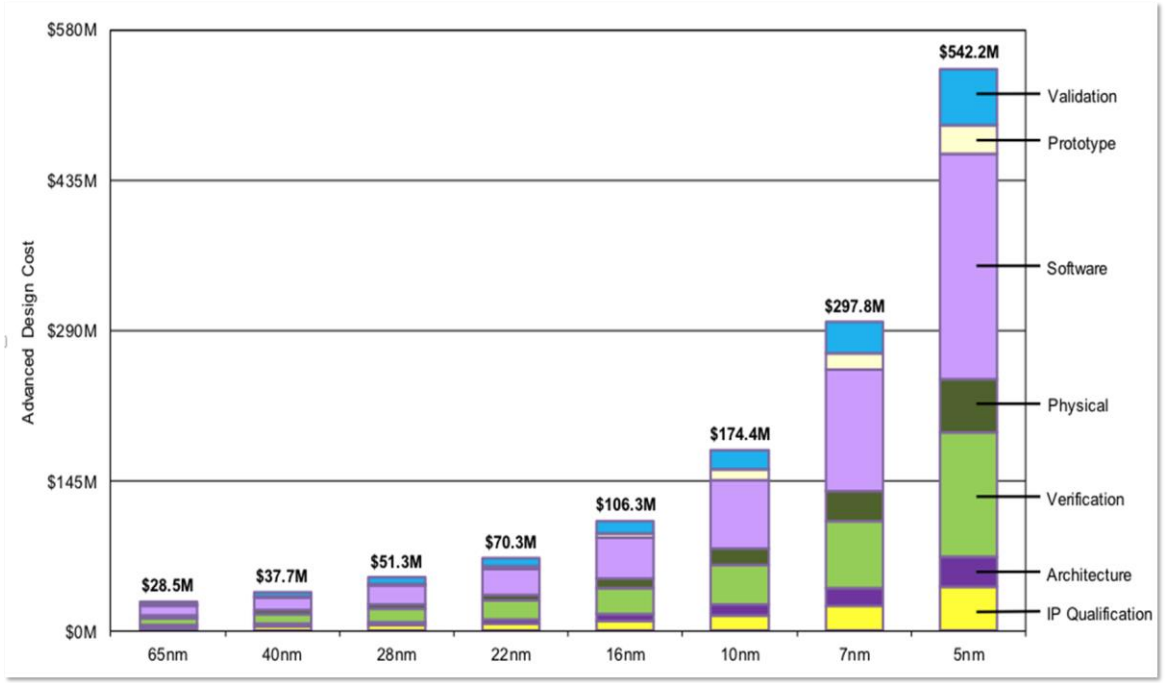


- 现代处理器经过50年的发展，已大幅提升性能、能效、集成度等
- 由于复杂的设计流程，传统处理器设计**迭代周期长、研发成本高**

## 芯片设计案例：Apple A18




## 研发成本高



# 开源和敏捷芯片设计

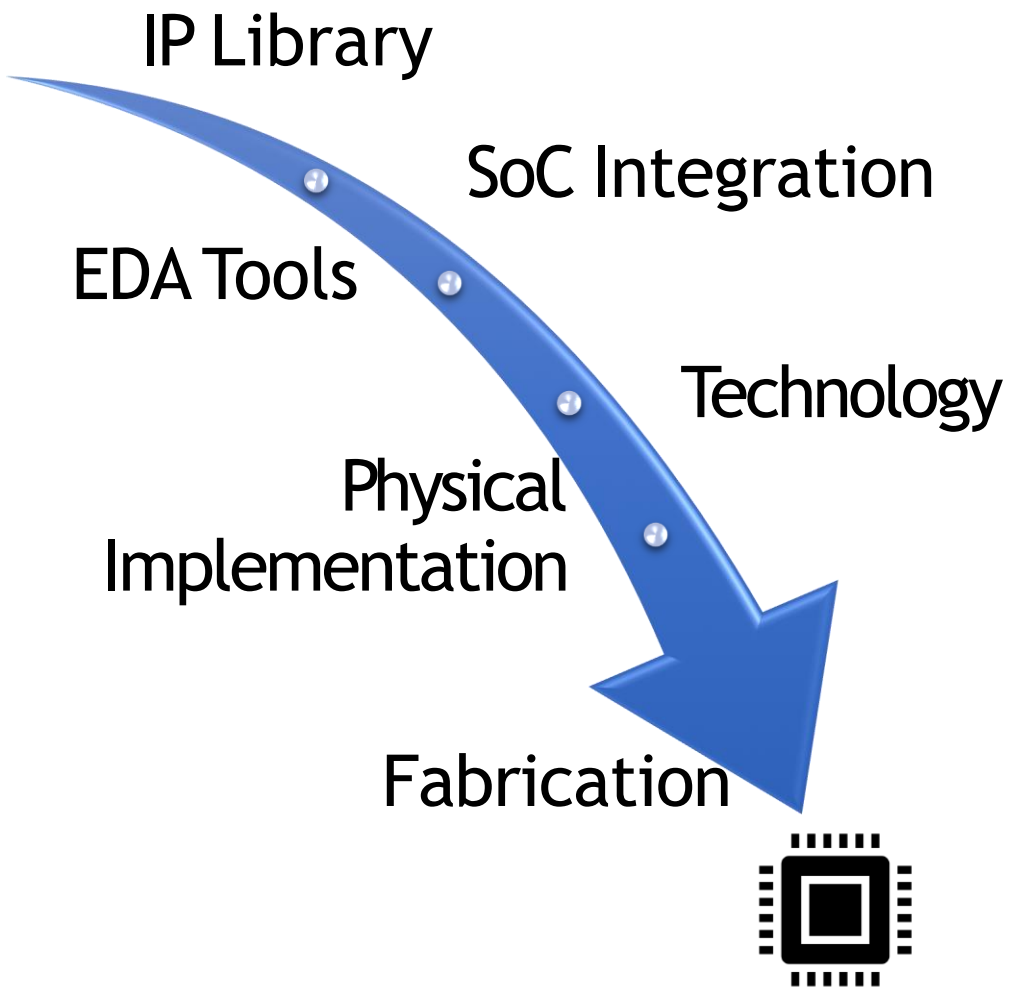
集成电路学院

RISC-V

CHIPYARD

opentitan

efabless.com




ESP

24h OpenROAD

OpenPiton

skywater

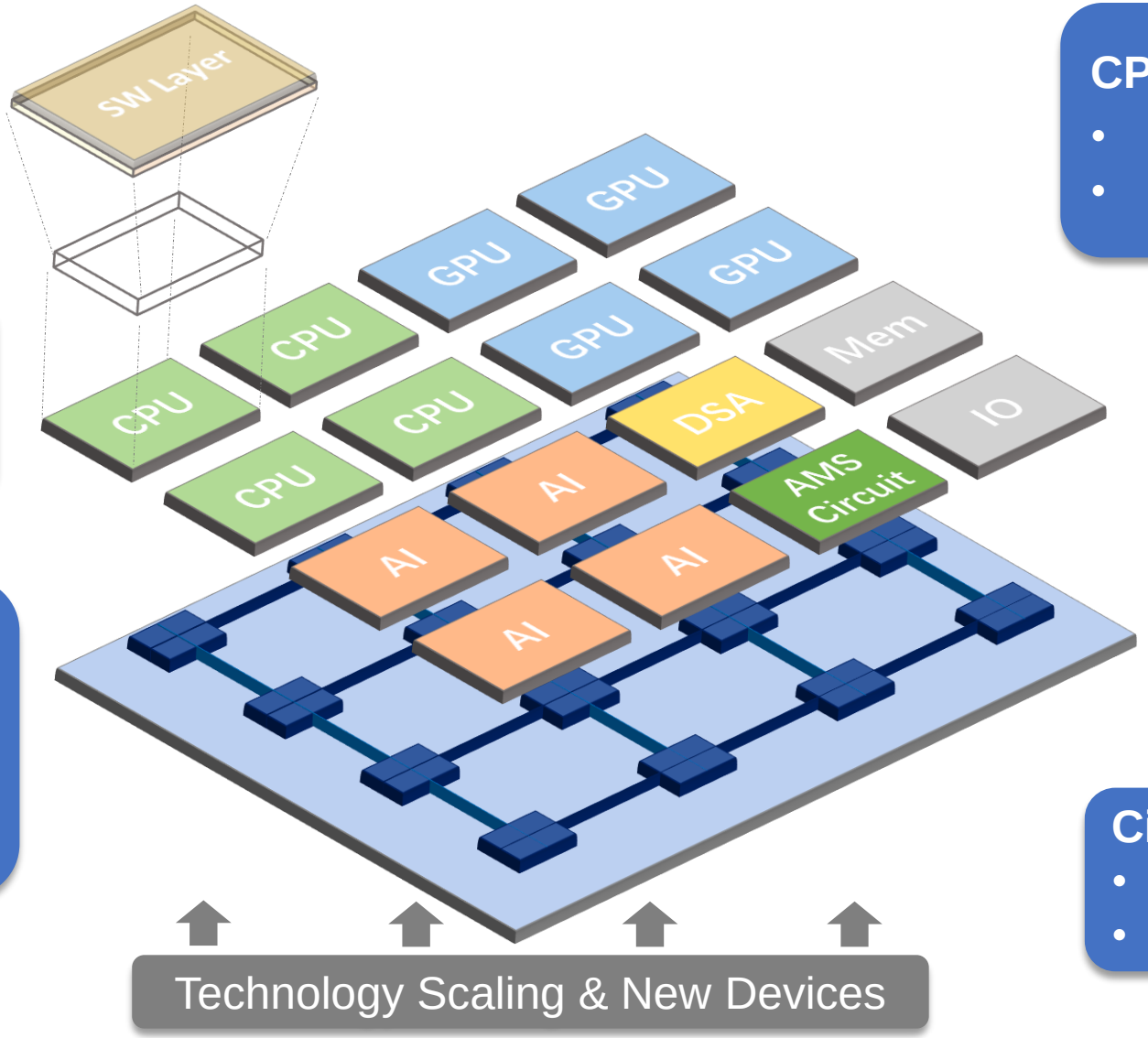
# 异构的SoC计算芯片



**SW/HW mapping:**
- Compilation
- Scheduling

**Design Method:**
- Open-source HW
- Agile Flow

**Interconnect for Heterogeneous:**
- Network on Chip
- Cache coherence
- Memory system



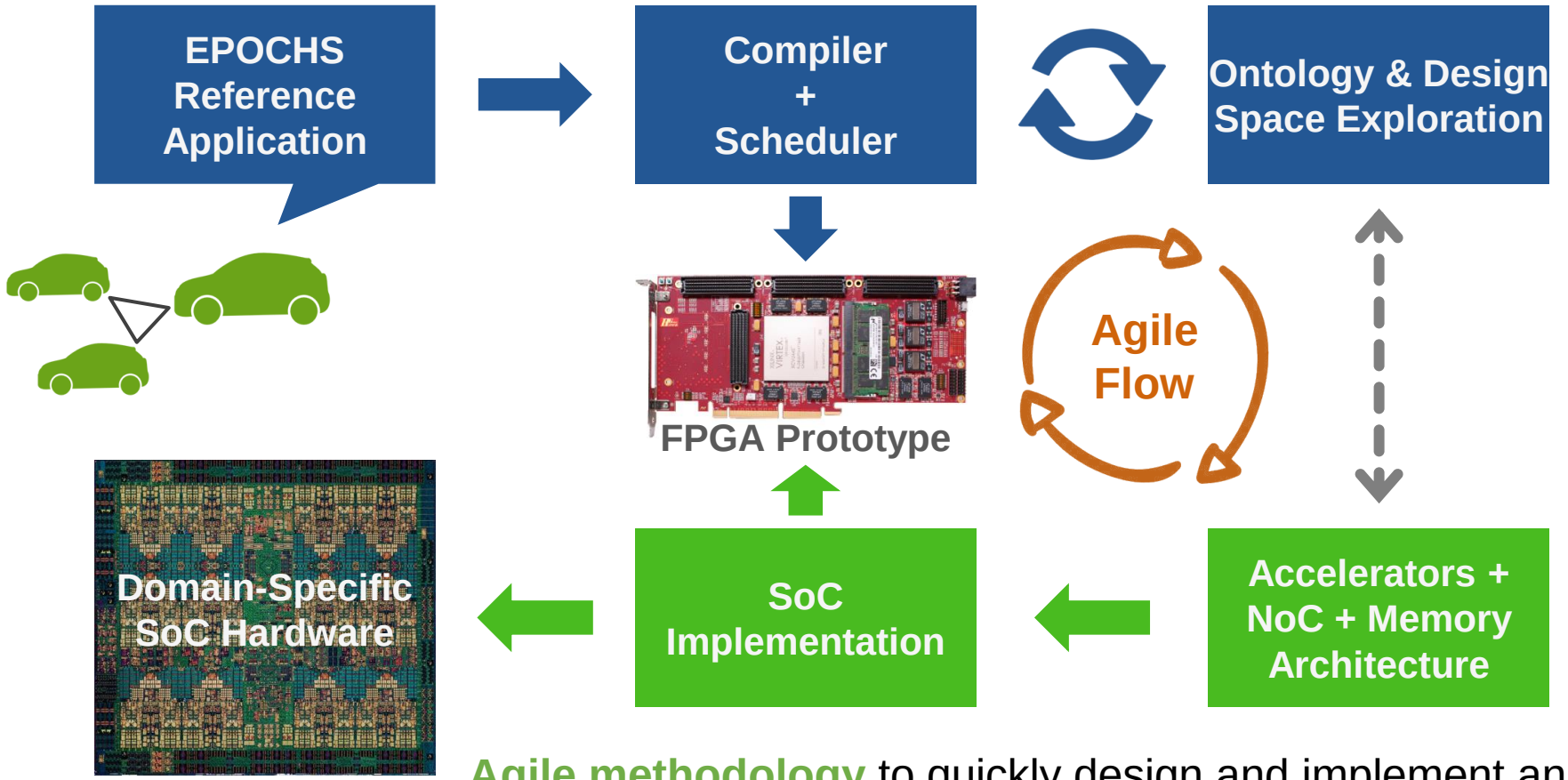

**CPU/GPU:**
- Multi-core/parallelism
- Microarchitecture

**DS Accelerator:**
- DNN-based
- Specialized AI
- Voice, ISP, etc.

**Circuits:**
- Power, Clock
- RF, IO, etc.

# 国外研究：DARPA DSSoC






**10X – 100X** reduction in person-years

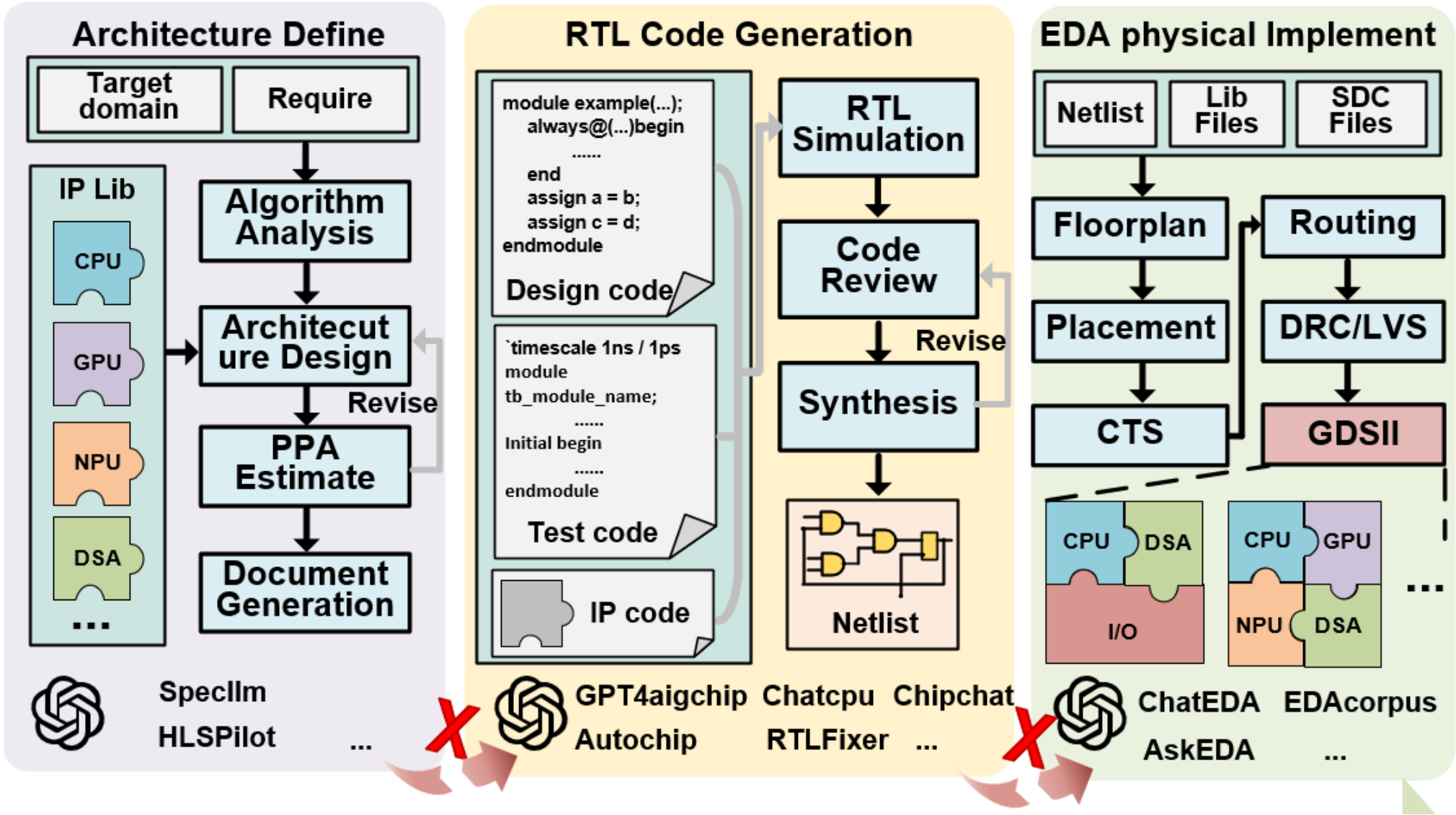
FPGA prototype, emulation, optimization, software bring-up

**Agile methodology** to quickly design and implement an easily programmed domain-specific SoC for real-time cognitive decision engines in connected vehicles
**"Super"-Domain:** Software-Defined Radio + Computer Vision

Source: DARPA ERI Summit

# 大模型辅助的设计方法



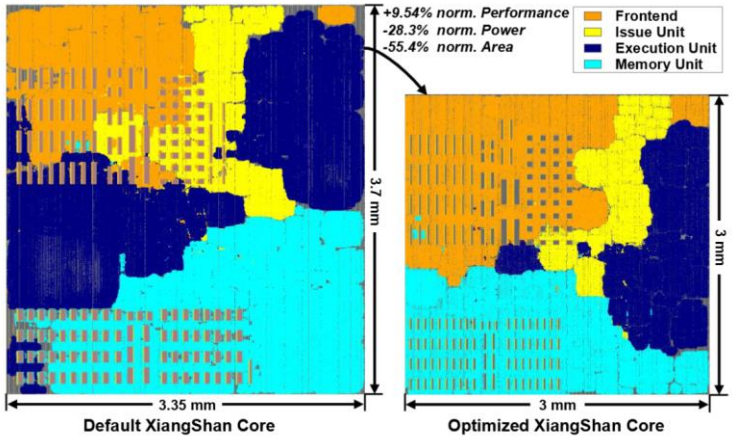

Architecture Define
Target domain
Require
IP Lib
CPU
GPU
NPU
DSA
...
Algorithm Analysis
Architecut ure Design
Revise
PPA Estimate
Document Generation
SpecIlm
HLSPilot
...
RTL Code Generation
module example(...);
always@(...)begin
.....
end
assign a = b;
assign c = d;
endmodule
Design code
`timescale 1ns / 1ps
module
tb_module_name;
.....
Initial begin
.....
endmodule
Test code
IP code
RTL Simulation
Code Review
Revise
Synthesis
Netlist
GPT4aigchip
Autochip
Chatcpu
RTLFixer
Chipchat
...
EDA physical Implement
Netlist
Lib Files
SDC Files
Floorplan
Placement
CTS
Routing
DRC/LVS
GDSII
CPU
DSA
I/O
CPU
GPU
NPU
DSA
...
ChatEDA
AskEDA
EDAcorpus
...

# 大模型辅助的设计方法



## ■ 基于大模型辅助的RISC-V的敏捷探索设计工作尝试

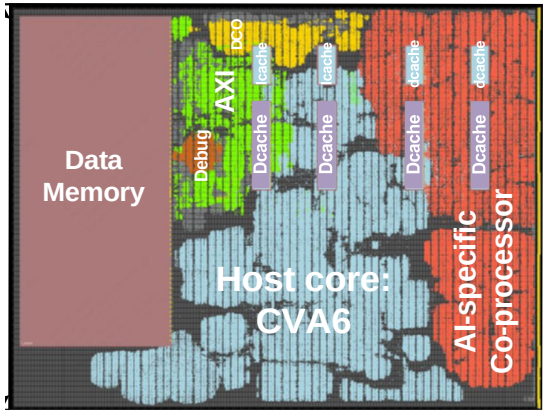
**阶段1：芯片架构**
**设计空间探索**




LLM

基金委区域联合
集成项目支持

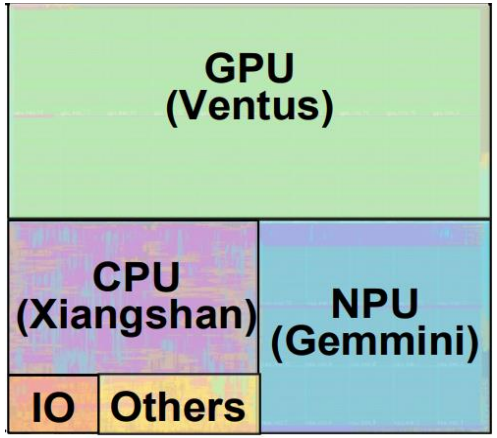
**阶段2：芯片设计**
**代码自动生成**




LLM

ISEDA 2025
最佳论文提名

**阶段3：芯片实现**
**自动集成实现**




LLM

ISLPED 2025
最佳论文提名

# Outline



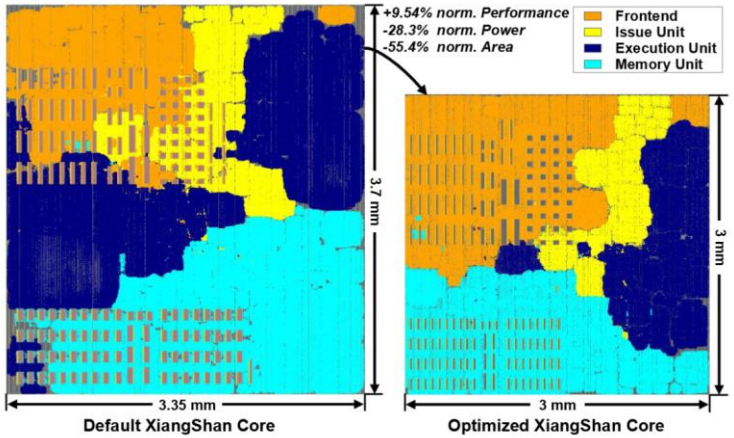
- 研究背景
- **大模型辅助RISC-V SoC设计**
  - **设计空间探索**
  - 代码生成
  - SoC集成
- 小结

# 大模型辅助的设计方法



## ■ 基于大模型辅助的RISC-V的敏捷探索设计工作尝试

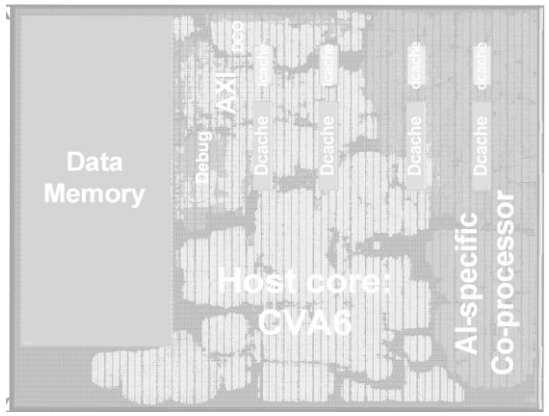
**阶段1：芯片架构**
**设计空间探索**




LLM

**基金委区域联合**
**集成项目支持**

阶段2：芯片设计
代码自动生成




LLM

ISEDA 2025
最佳论文提名

阶段3：芯片实现
自动集成实现

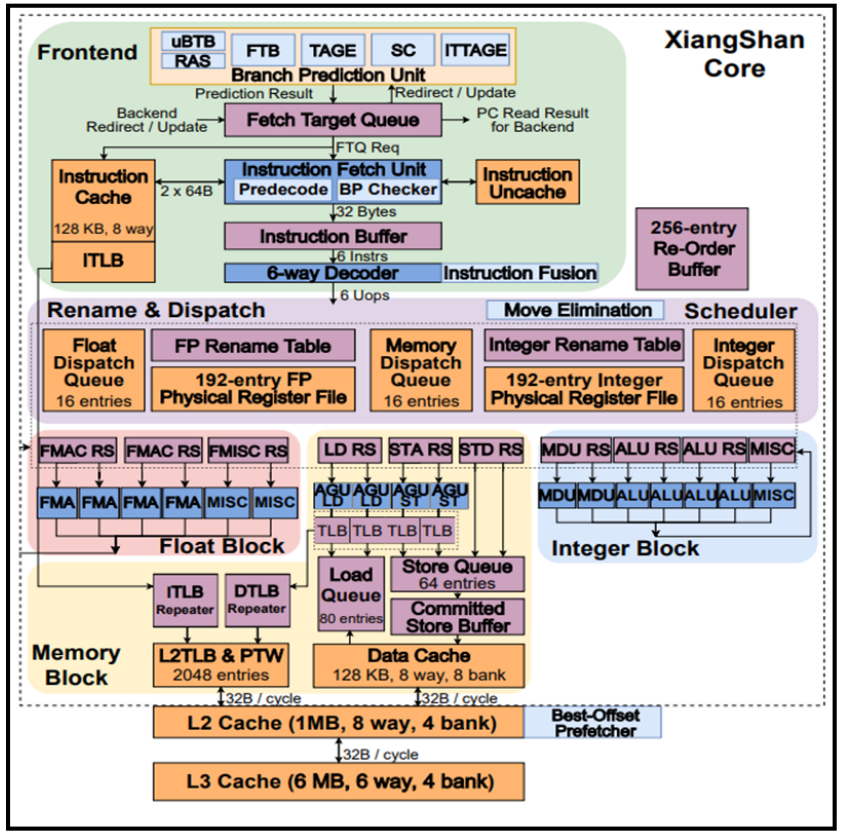

LLM

ISLPED 2025
最佳论文提名

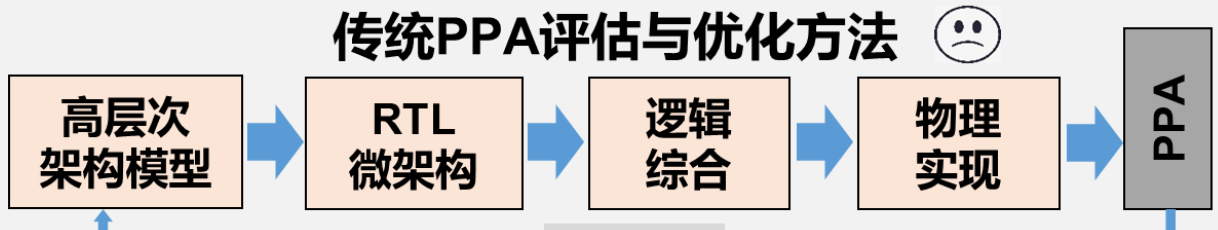
# Case 1: DSE for XiangShan CPU



## 高性能通用处理器

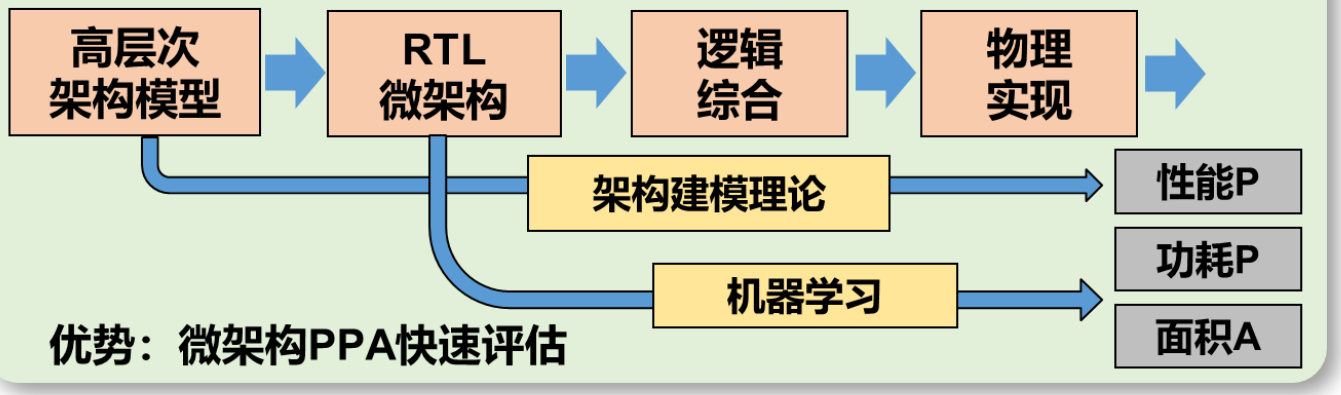



### 传统PPA评估与优化方法




问题：迭代周期冗长

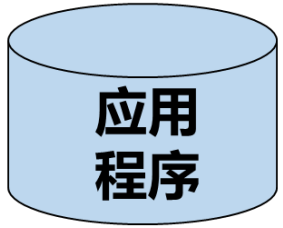
### 层次化快速PPA评估与优化



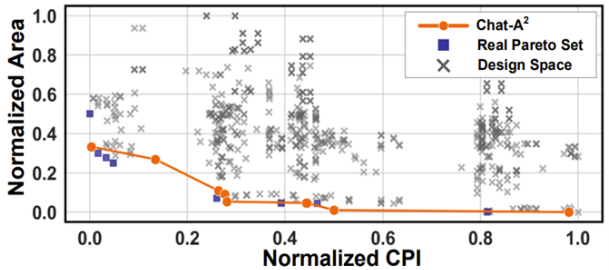

优势：微架构PPA快速评估

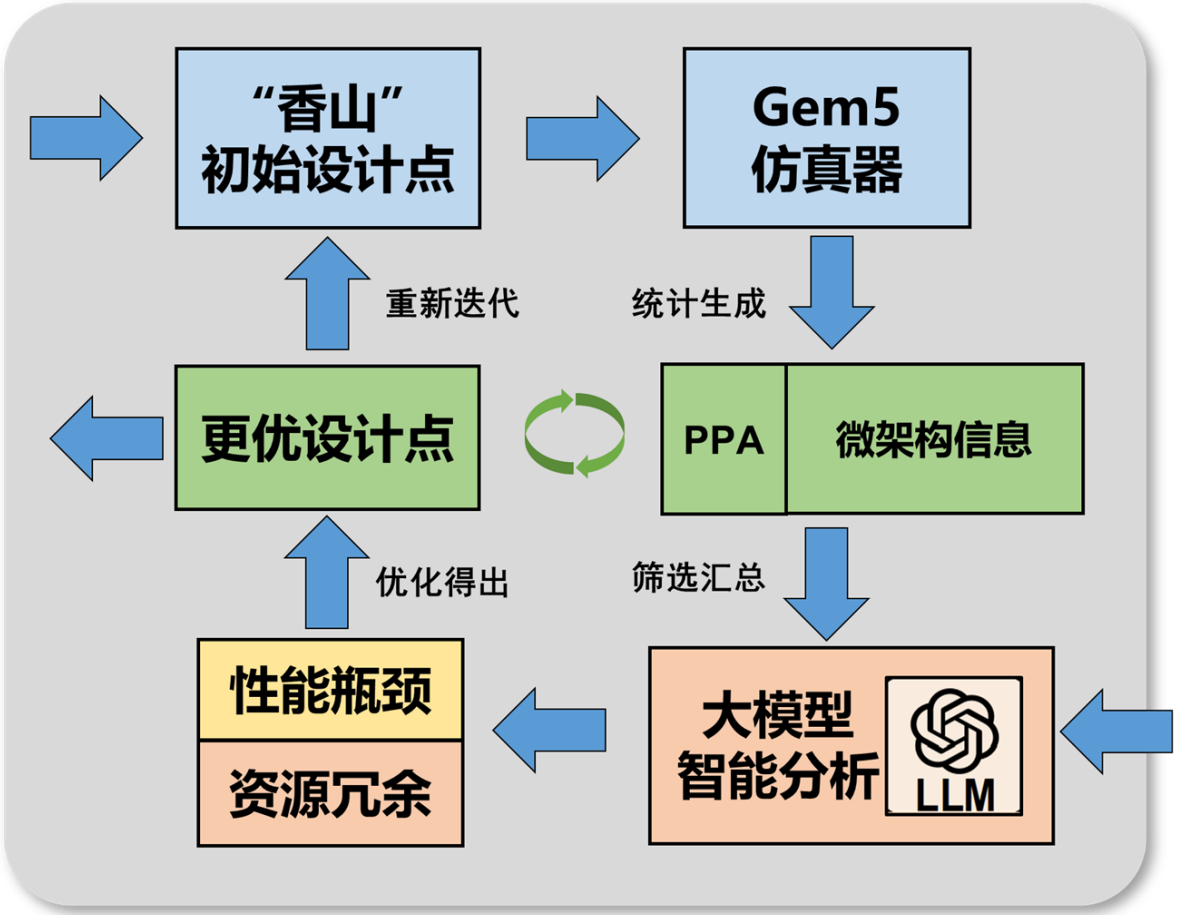
# Overall Framework






## 帕累托边界




## 提示词工程

**Assuming you're a professional CPU architect. Your task is to optimize the CPU parameters.** *Prompt*

For *numROBEntry*, the report is shown as below:

system.cpu.rename.ROBFullEvents *p*

***p*** **means the number of times rename has blocked due to ROB full.**

**Here is the report in *design(i)* :**

system.cpu.rename.ROBFullEvents 75515

**To obtain better PPA, *numROBEntries* should be just enough.**

**What are *ROBFullEvents* and *CPI* equal to in previous designs?**

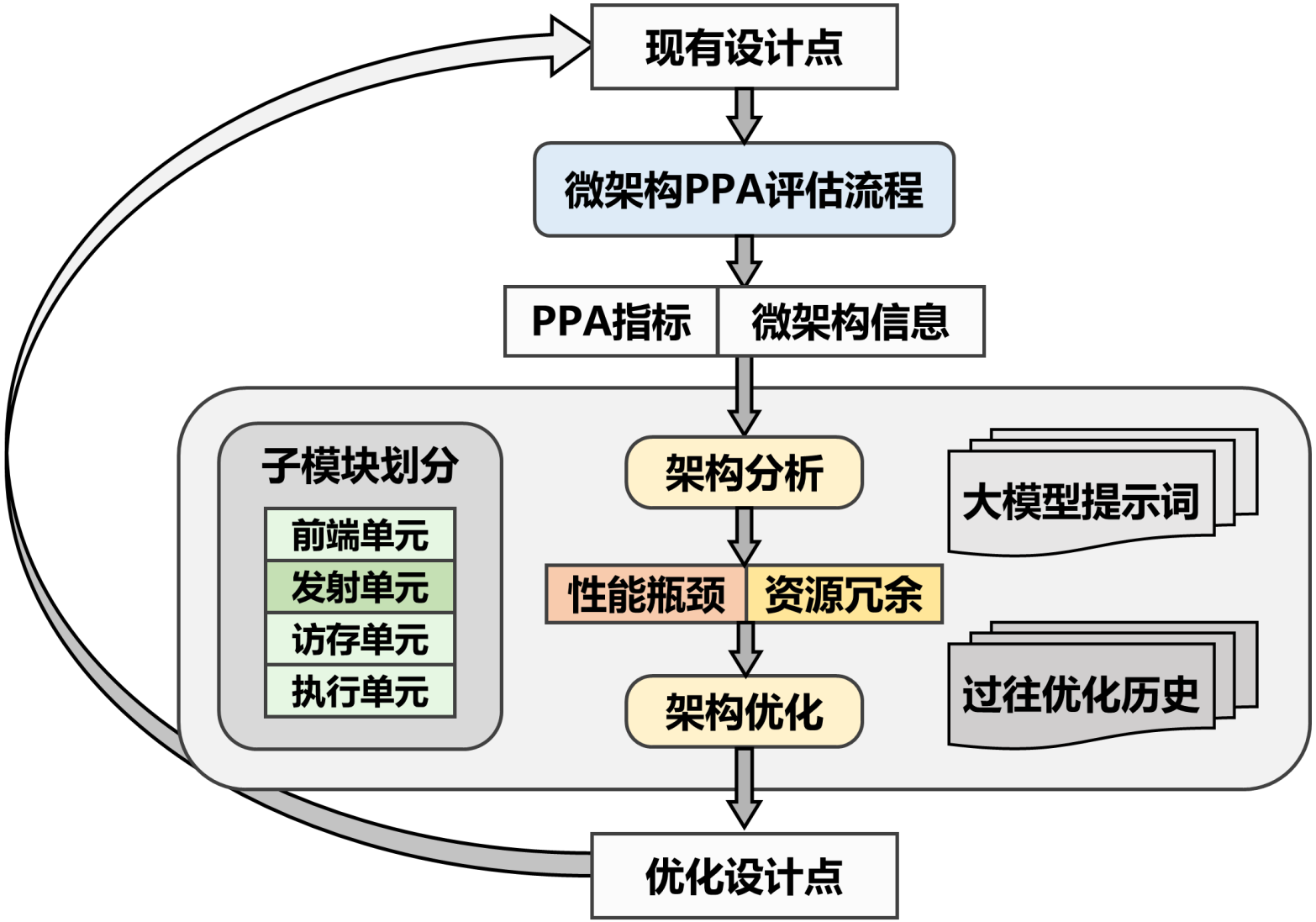
**Compare previous {ROBFullEvents,CPI} with the current *design(i)* .**

**And refer to other parameters in the *design(i)* .**

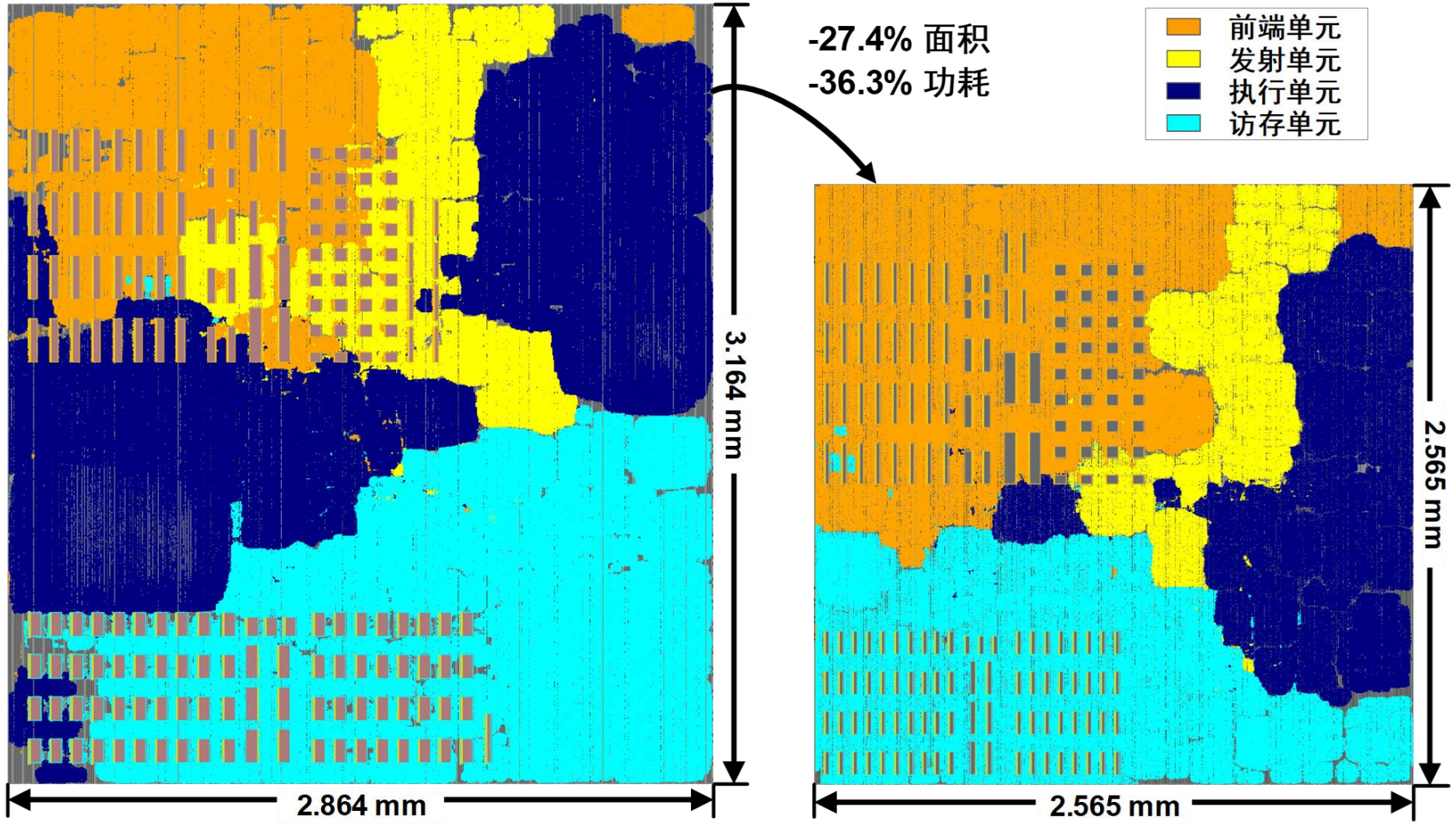
**Should We increase or decrease or maintain *ROB_Entry* ?**

**A clear number is preferred**

# uArch Analysis and Optimzations



# Optimizations for XiangShan CPU




默认配置 香山 CPU 核

优化配置 香山 CPU 核

# Outline

集成电路学院

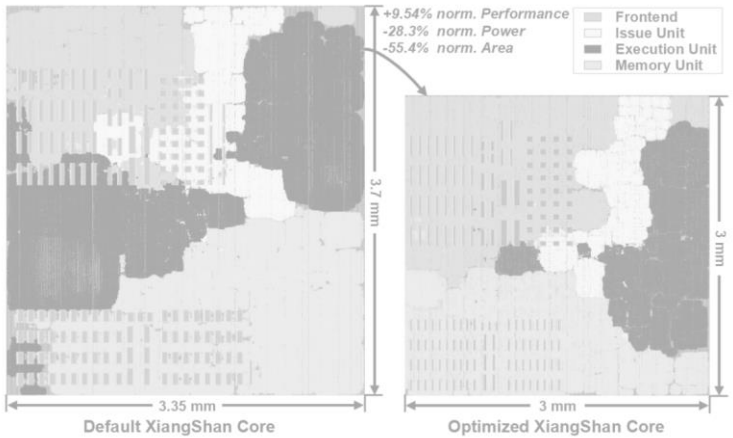
- 研究背景
- **大模型辅助RISC-V SoC设计**
  - 设计空间探索
  - **代码生成**
  - SoC集成
- 小结

# 大模型辅助的设计方法



## ■ 基于大模型辅助的RISC-V的敏捷探索设计工作尝试

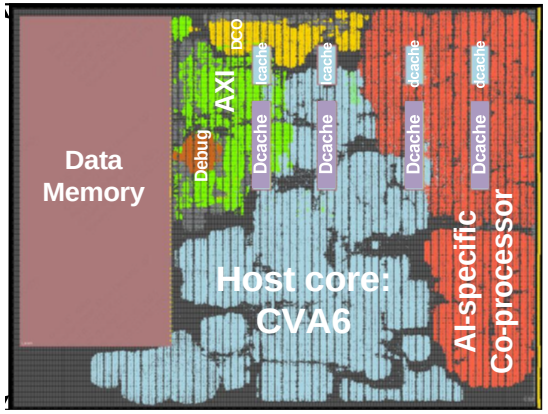
阶段1：芯片架构
设计空间探索




LLM

基金委区域联合
集成项目支持

阶段2：芯片设计
代码自动生成




LLM

ISEDA 2025
最佳论文提名

阶段3：芯片实现
自动集成实现

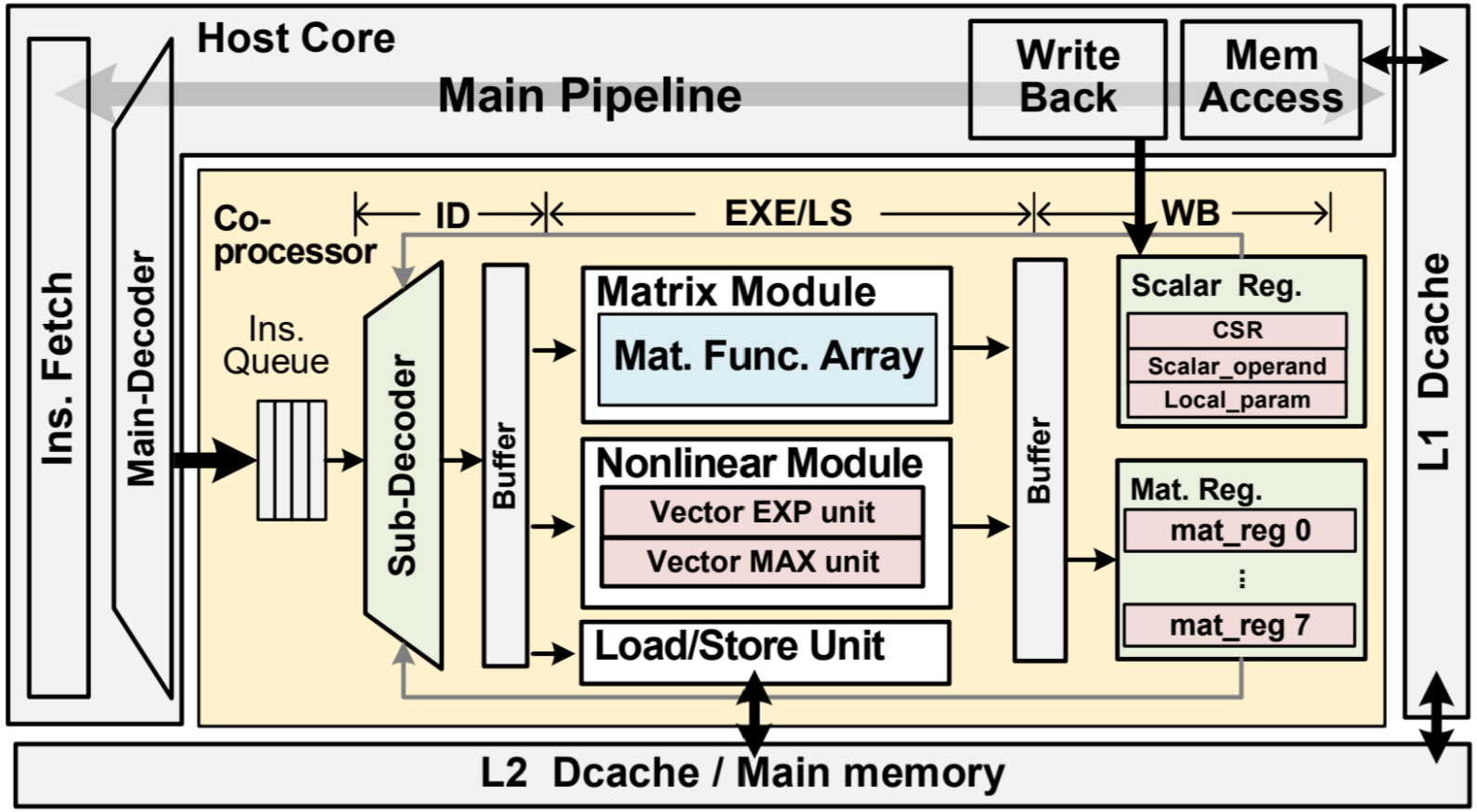

LLM

ISLPED 2025
最佳论文提名

# Co-processor Architecture

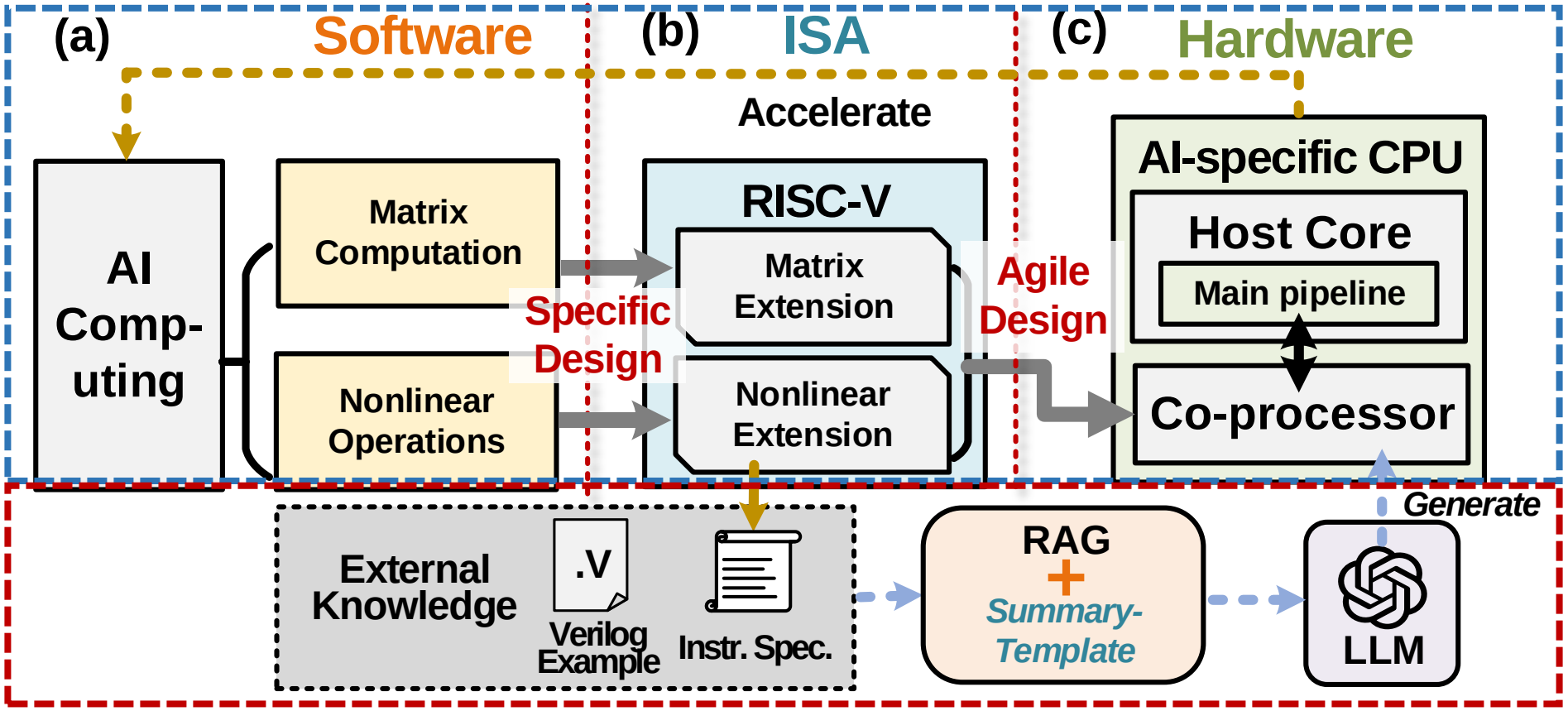
- Directly connected interface
- 3-stage pipeline
- Shared physical reg for mat. and nonlinear instructions
- Scalar CSR
- Dedicated L/S port to L2 D-cache for high bandwidth

## Co-processor Architecture



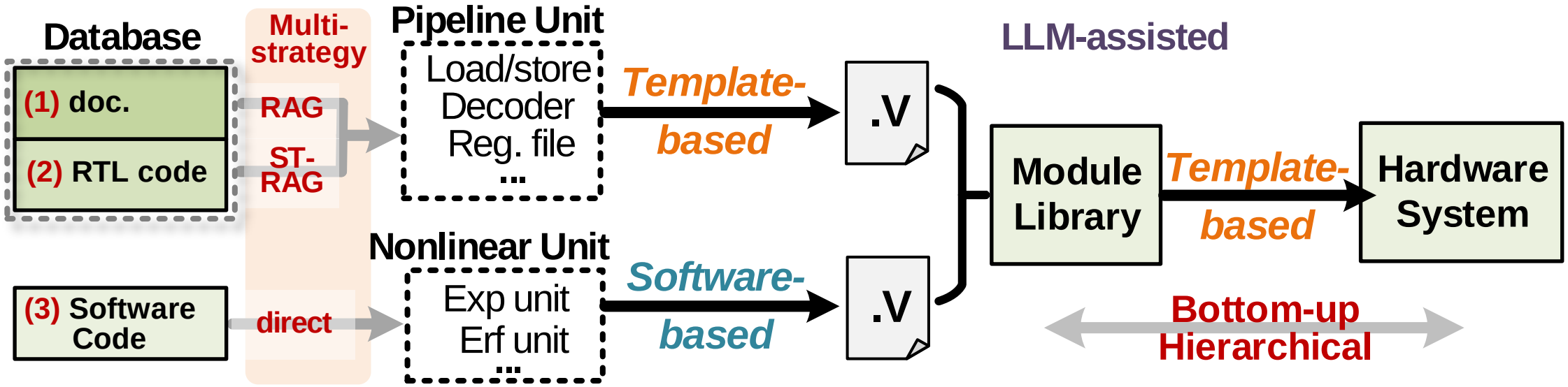
# Overview of SW/HW

- ISA Extension and coprocessor template
- VeriRAG for agile hardware code generation



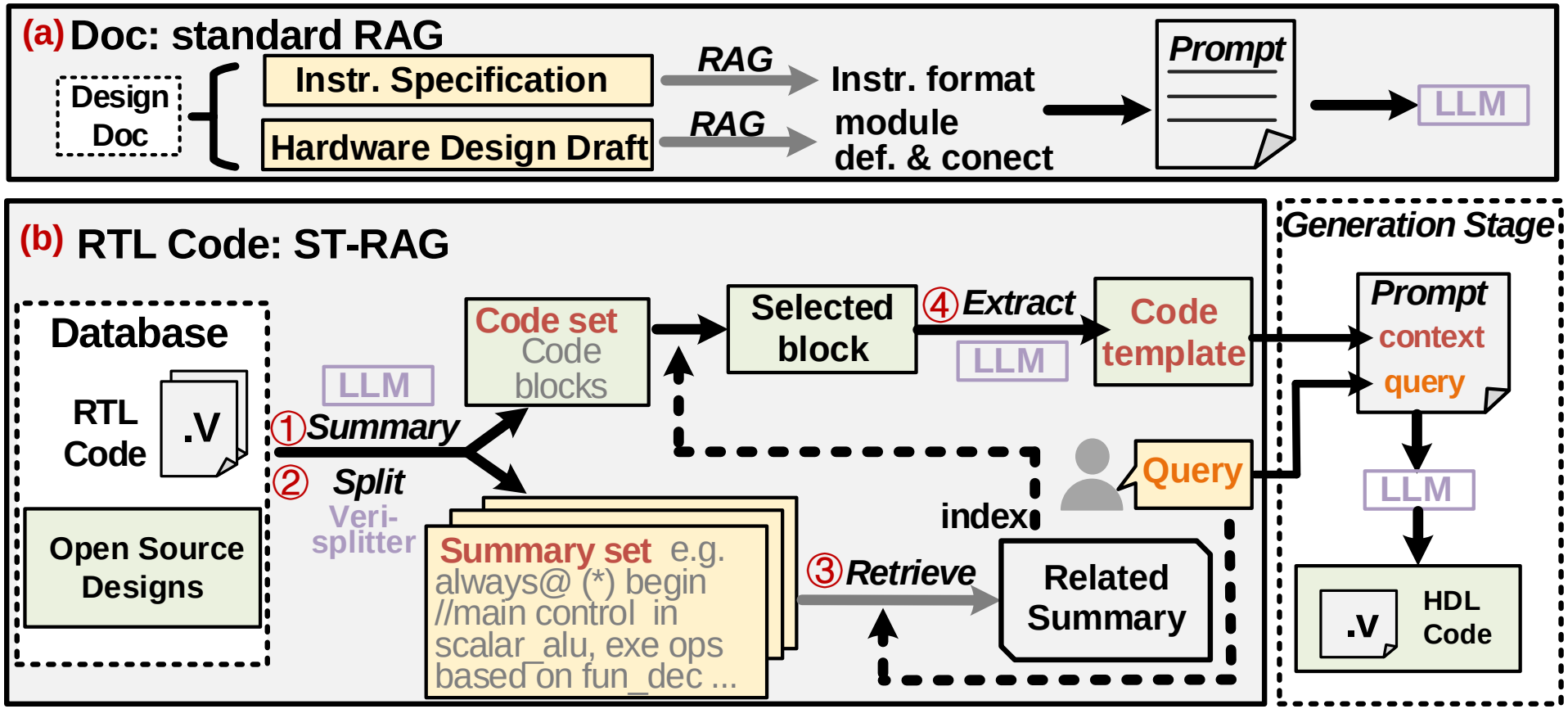
# Code Generation Flow Overview

- Three text types in database: natural language doc, RTL code, software code
- Special Summary-Template RAG for RTL code
- Two RTL code generation methods



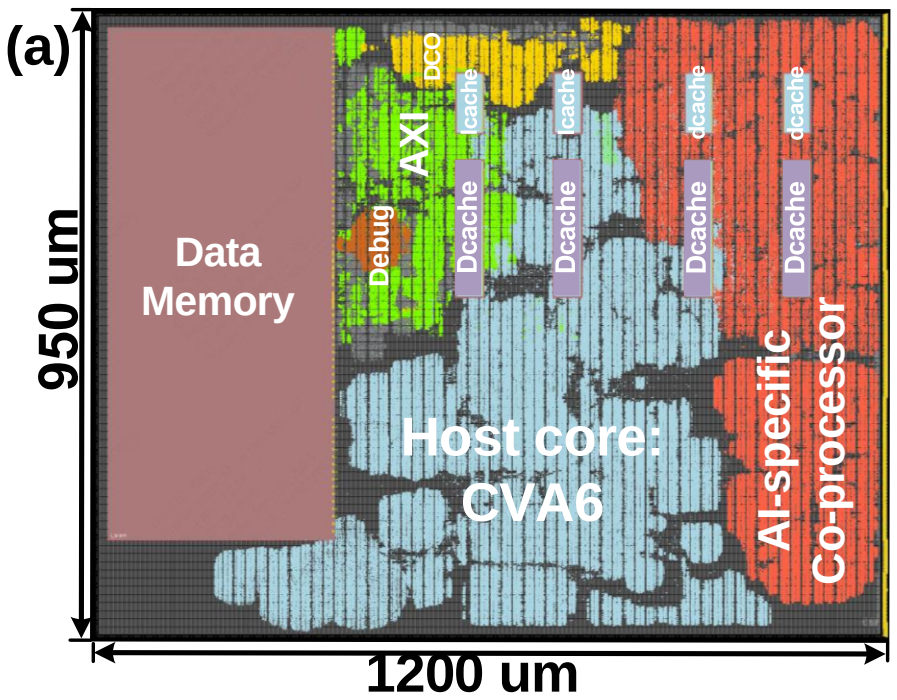
# RAG for RTL code Generation

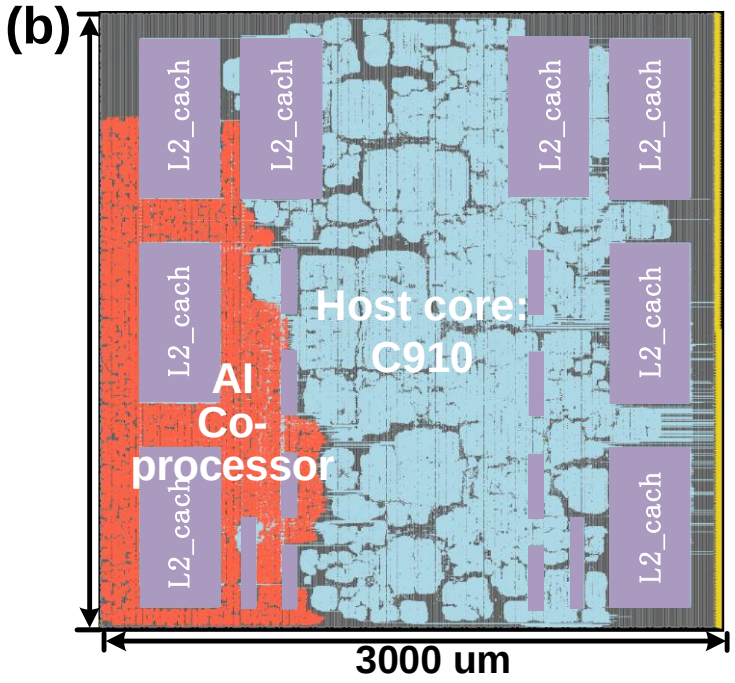
- New stage in RAG: pre-**summary** and **template extract** after RAG



# Co-processor Implementation

- Different configuration for edge and high-end CPU



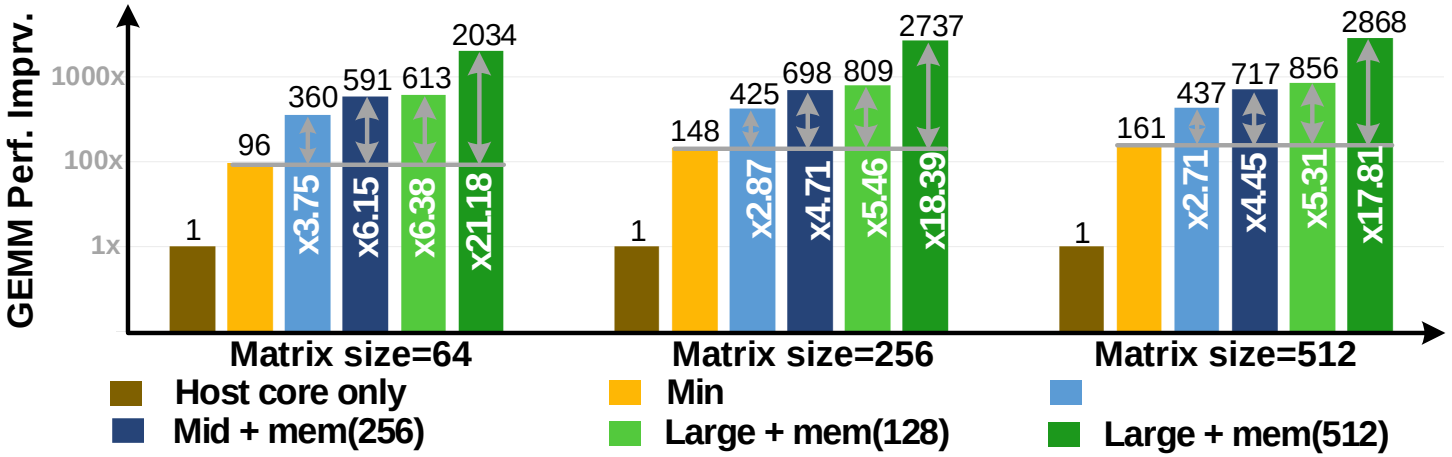




| Edge | High-end |
|---|---|
| CVA6 | Xuantie C910 |

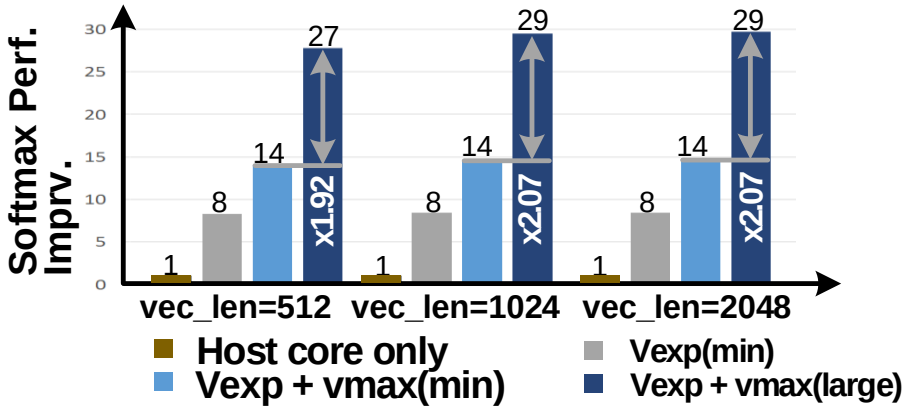
| | nR | RLEN |
|---|---|---|
| minimum | 4 | 128 |
| medium | 8 | 256 |
| large | 16 | 512 |

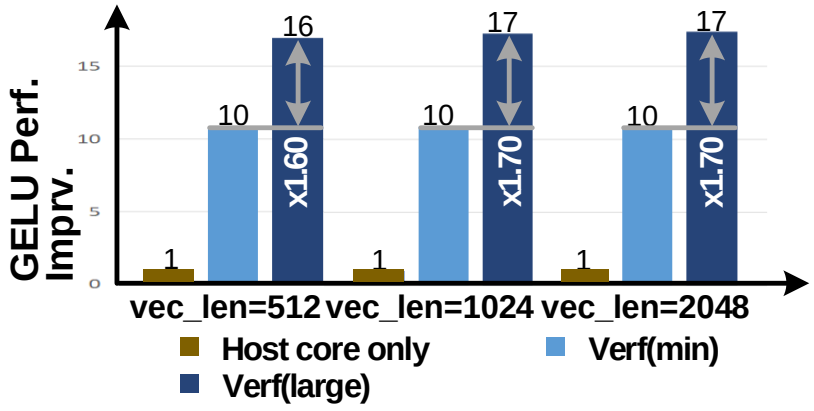
# Performance Improvement

- Matrix Extension Improvement




- Nonlinear Extension Improvement



# Outline



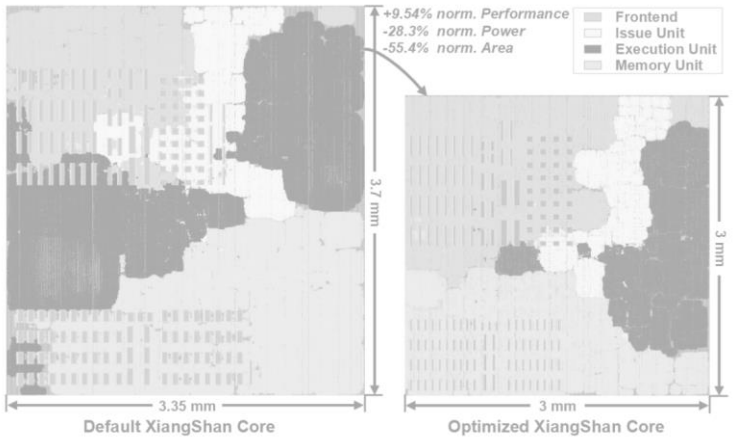
- 研究背景
- **大模型辅助RISC-V SoC设计**
  - 设计空间探索
  - 代码生成
  - **SoC集成**
- 小结

# 大模型辅助的设计方法



## ■ 基于大模型辅助的RISC-V的敏捷探索设计工作尝试

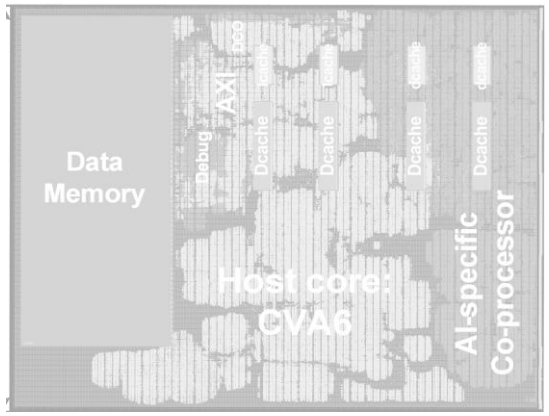
阶段1：芯片架构
设计空间探索




LLM

基金委区域联合
集成项目支持

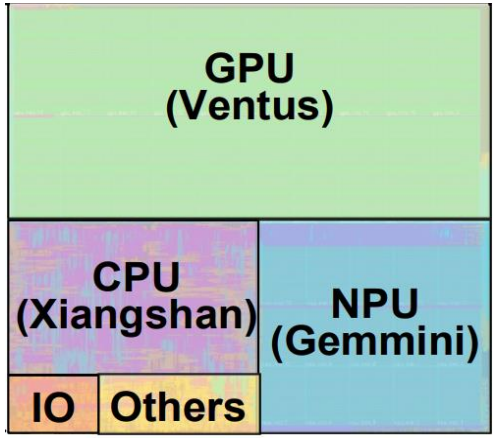
阶段2：芯片设计
代码自动生成




LLM

ISEDA 2025
最佳论文提名

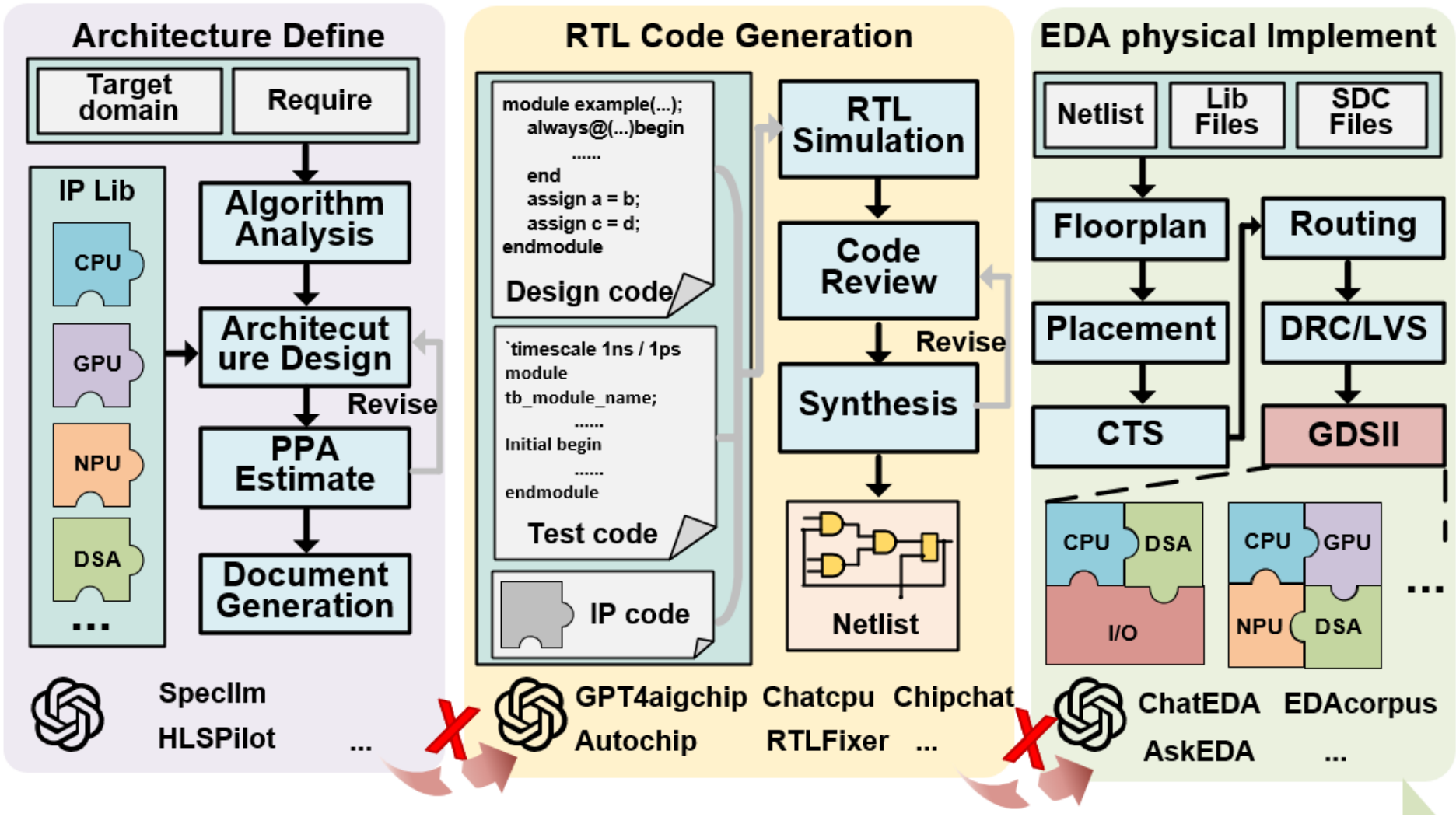
阶段3：芯片实现
自动集成实现



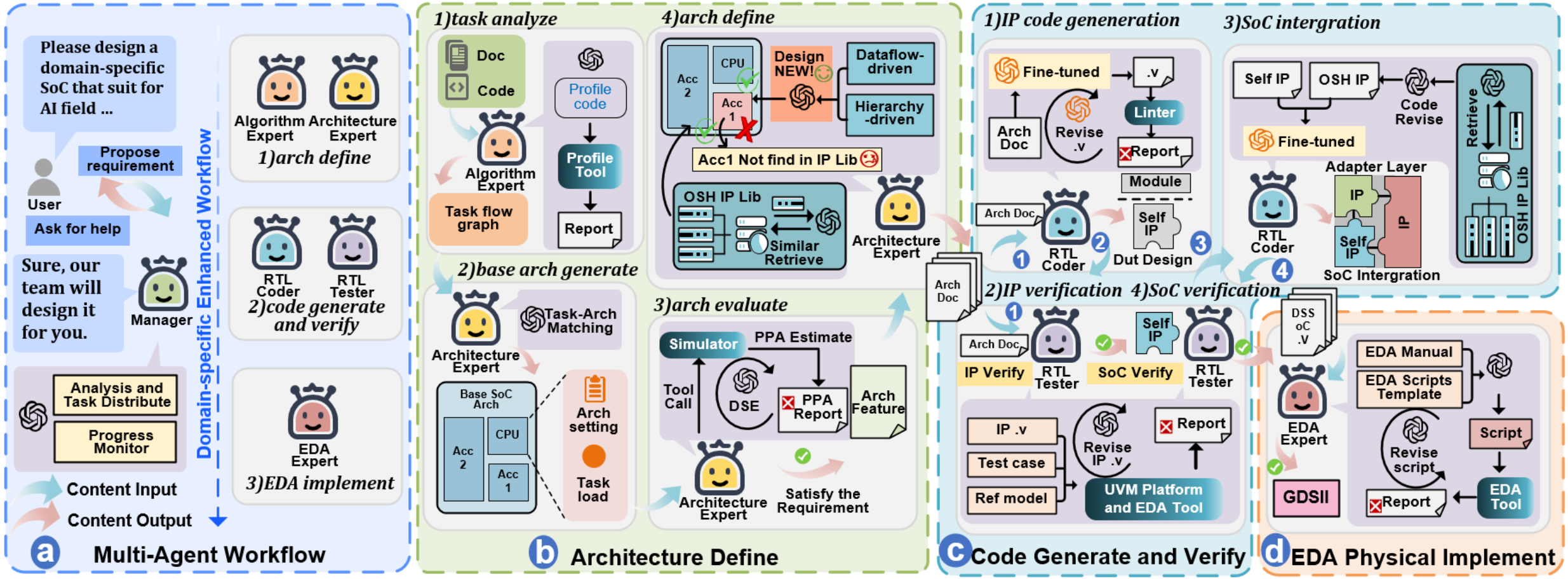

LLM

ISLPED 2025
最佳论文提名

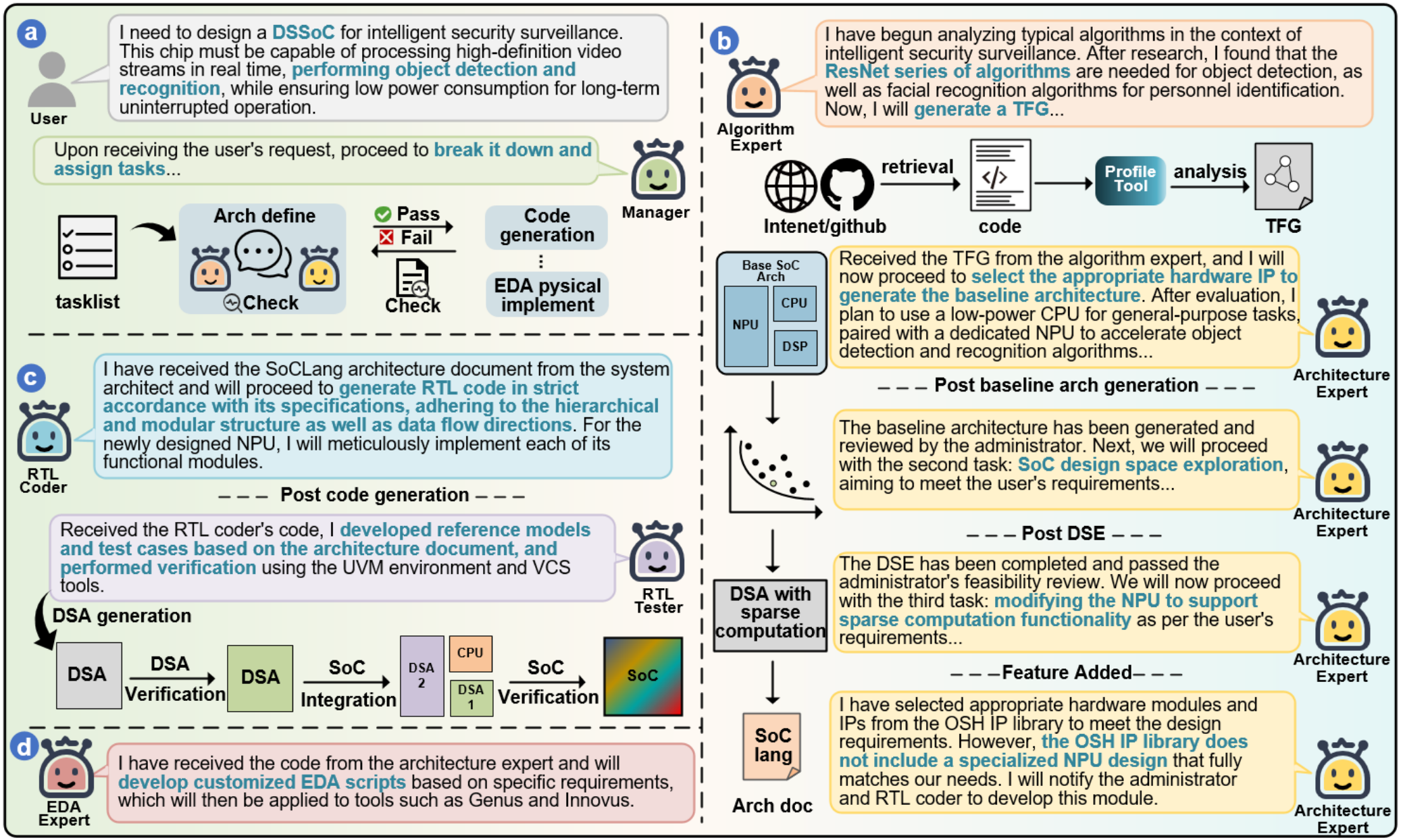
# End-to-End SoC Generation



# Multi-Agents for SoC Design

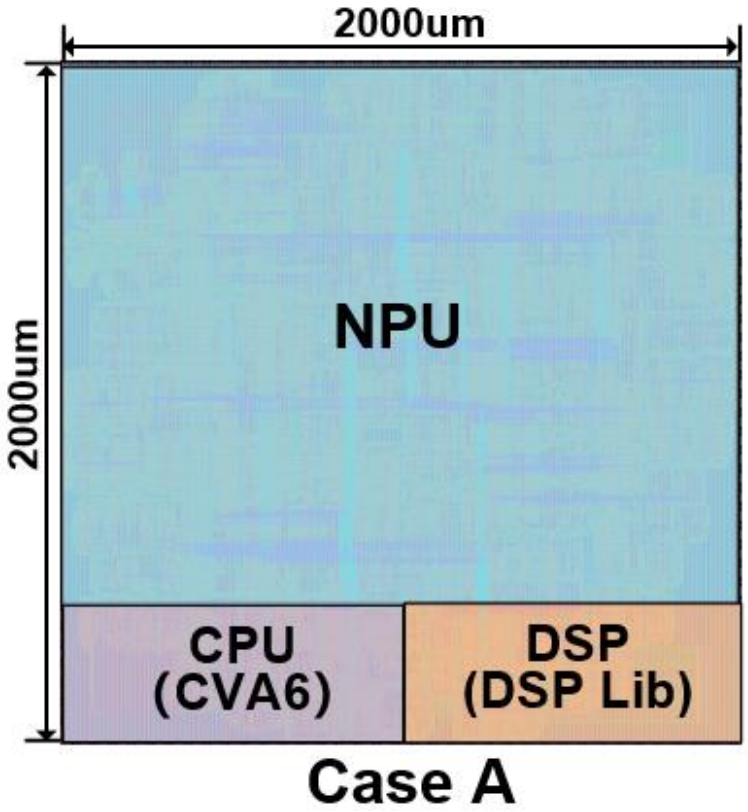


# Multi-Agents for SoC Design

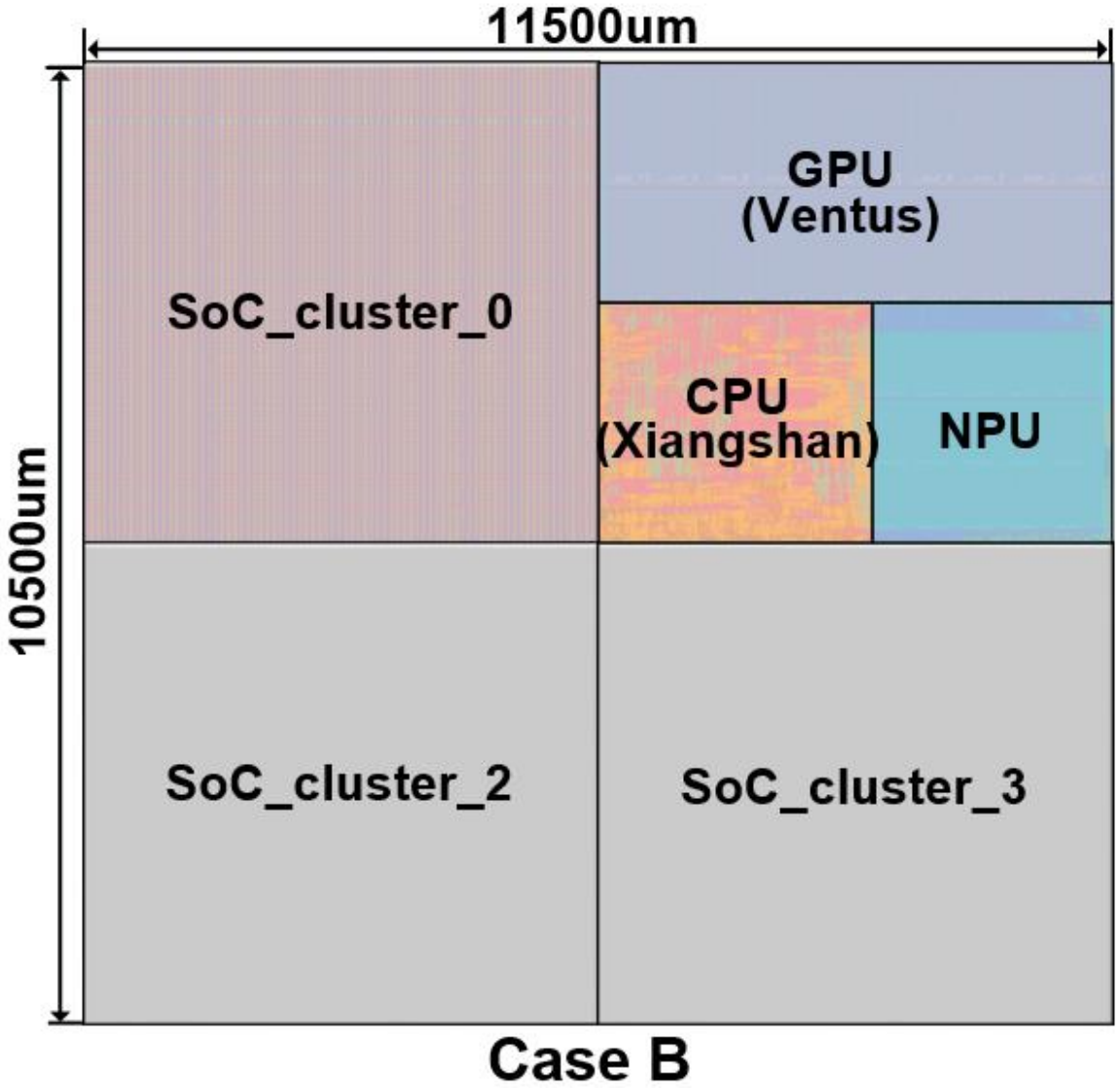


# Design Cases






Case A



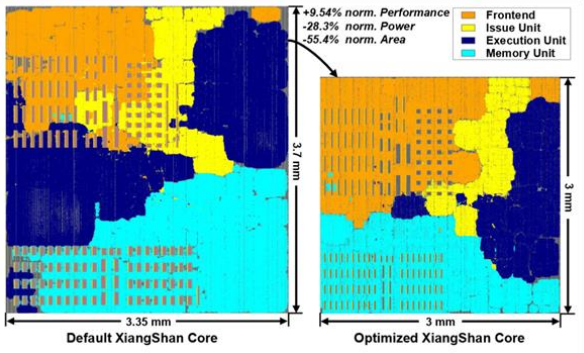

Case B

# 总结：大模型辅助的设计方法

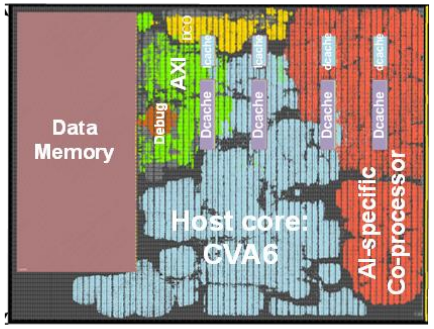


- 开源与敏捷芯片设计可以极大提升芯片的开发效率
- 大模型辅助+开源RISC-V，进一步提升设计效率
- LLM在设计空间探索、代码生成、芯片集成等多方面辅助
- 仅仅是开始，期待更加完整的敏捷开发设计流程

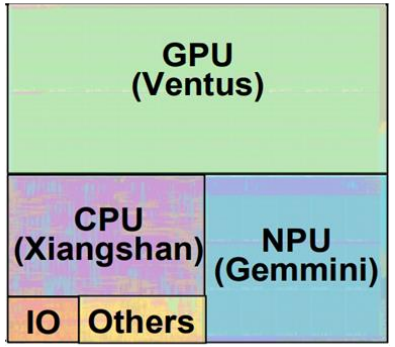
阶段1：芯片架构
设计空间探索




阶段2：芯片设计
代码自动生成




阶段3：芯片实现
自动集成实现



# 谢谢!

贾天宇

北京大学 集成电路学院

tianyuj@pku.edu.cn