

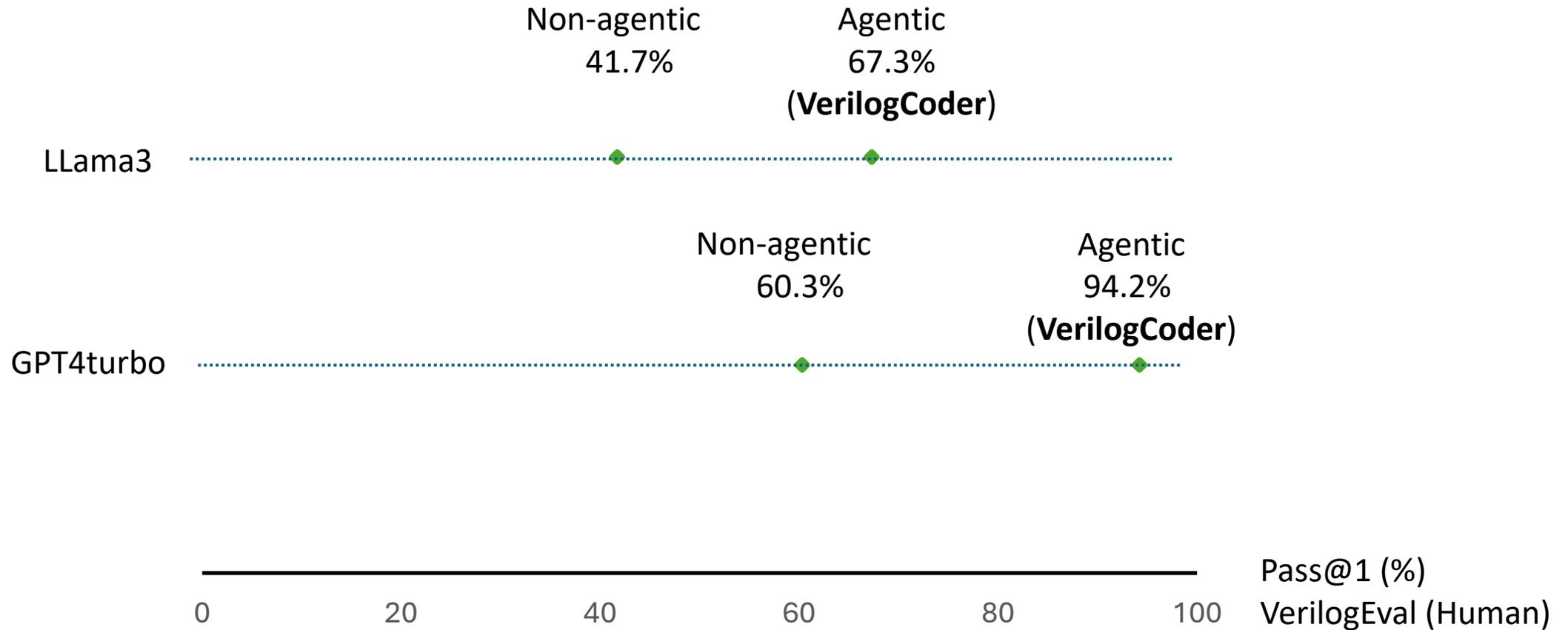


LLM Agents for Chip Design

Haoxing (Mark) Ren, Director of Design Automation Research, NVIDIA

08/25/2024

Agent Advantage



Agentic Concepts

Non-Agentic

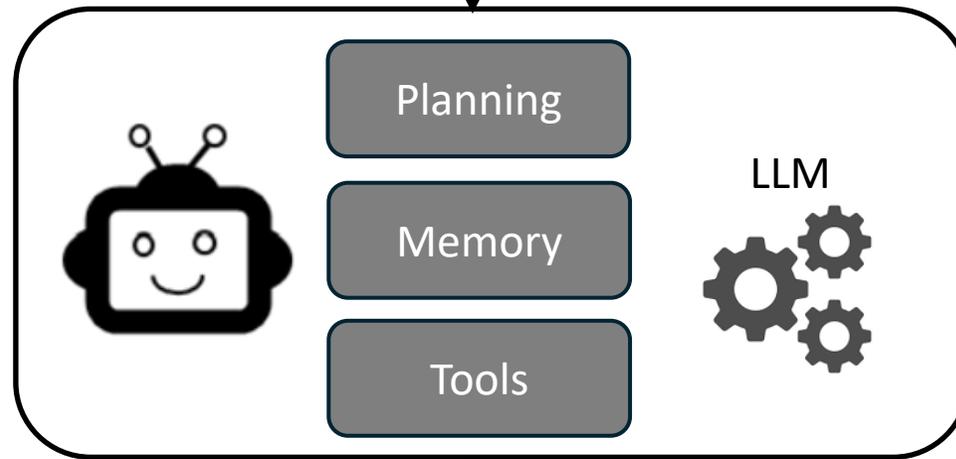
Simple Question



Answer

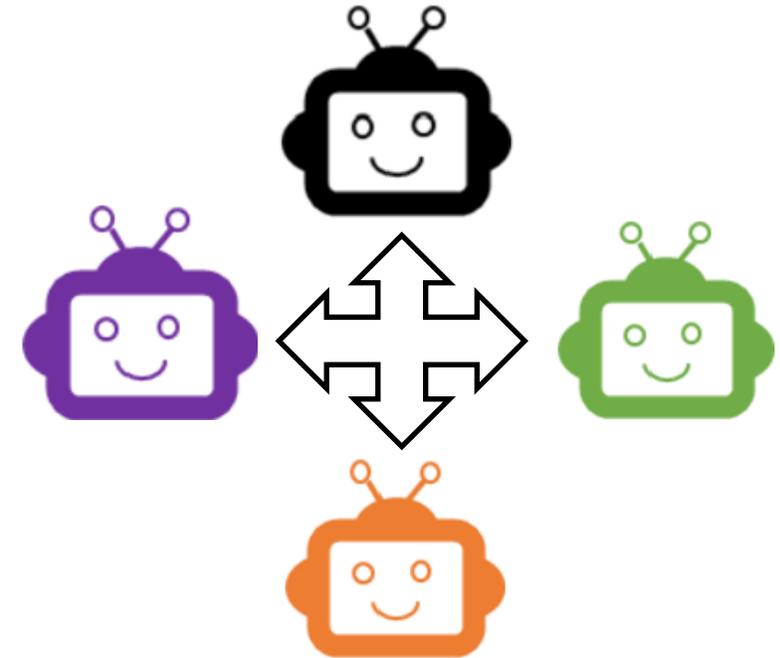
Agentic

Complex Question



Answer

Single Agent



Multi-Agent

Planning

- Without feedback
 - Chain-of-Thought
 - Tree-of-Thought
- With feedback
 - ReACT
 - Self-Reflection

Planning – Without Feedback

Chain-of-Thought (CoT): Step-by-step planning with one-shot task decomposition

Explicit Instruction

Design a RISC-V processor core, follow the following steps:

1. define the microarchitecture of the processor.
2. give the function specification of each microarchitecture component.
3. Implement each function in RTL

Implicit Instruction

Design a RISC-V processor core. Let's think step by step as a design engineer.

Demonstrative Examples

Implement a SPARC processor core :

The SPARC processor core is divided into 7 main blocks: Instruction fetch unit (IFU), Execution unit (EXU), Load/Store unit (LSU), ...

IFU: The instruction fetch unit (IFU) is responsible for maintaining the program counters (PC) of different threads and fetching the corresponding instructions.

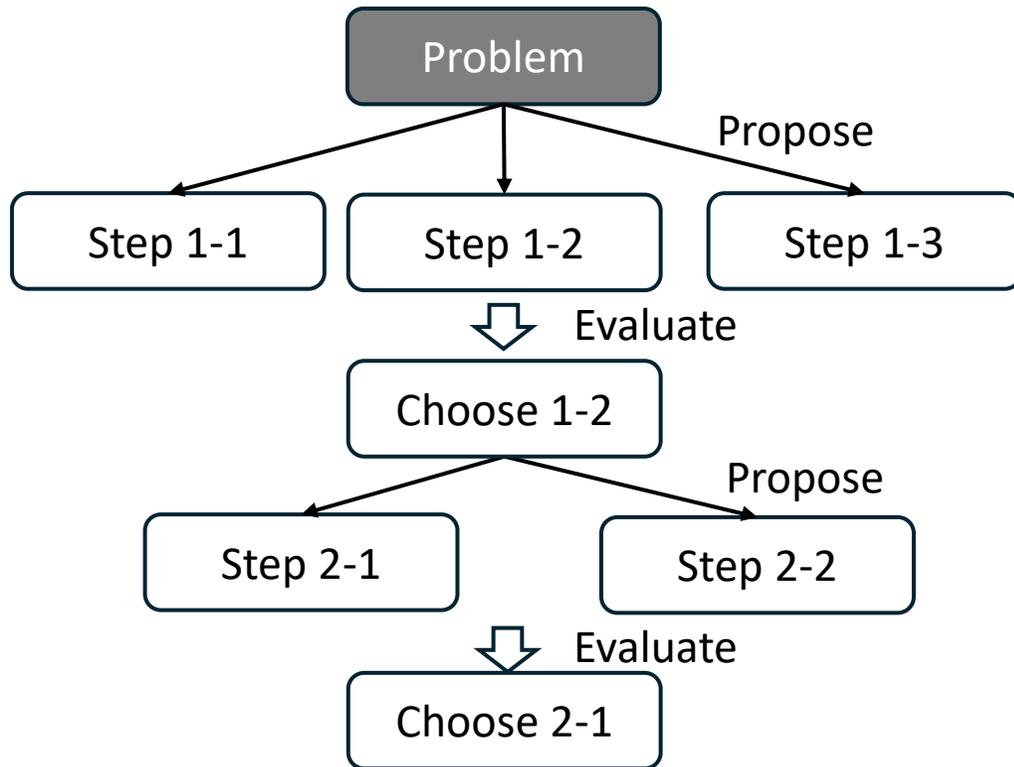
...

RTL code for IFU: ...

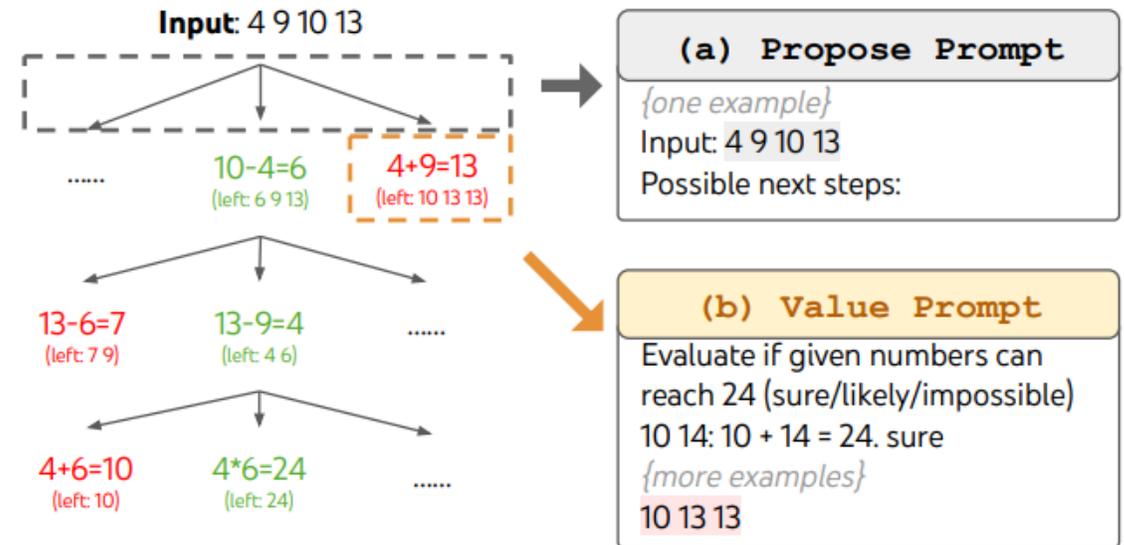
Now design a RISC-V processor core:

Planning – Without Feedback

Tree-of-Thought (ToT): plan one step at a time, choose best from multiple choices at each step, extend with BFS or DFS, or Graph



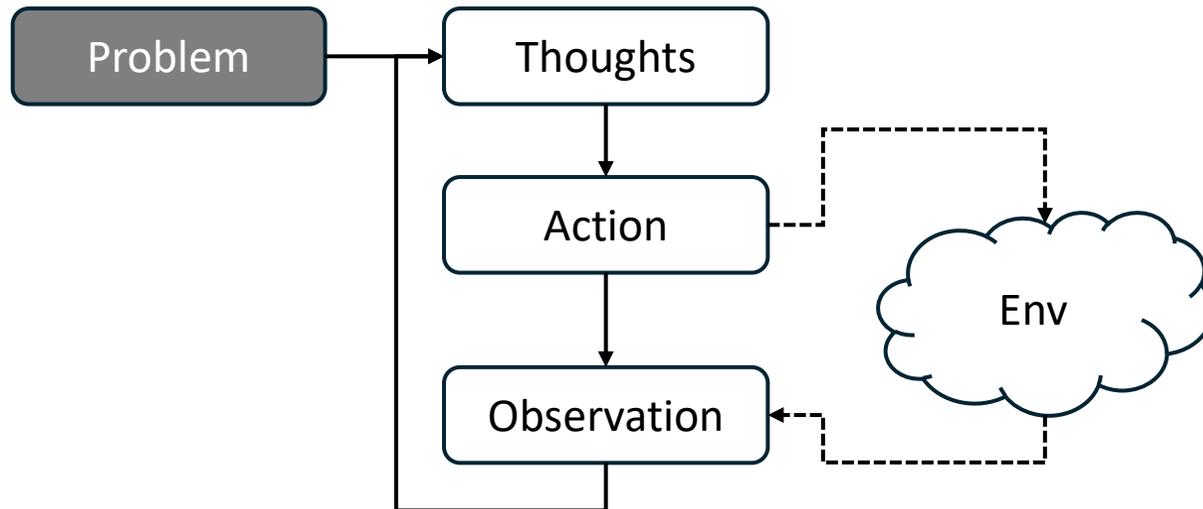
use 4 numbers and basic arithmetic operations (+-*/) to obtain 24



Planning – With Feedback

Ground planning with environment feedback

ReACT: Thought, Action and Observation



(1d) ReAct (Reason + Act)

Thought 1: I need to search Apple Remote and find the program it was originally designed to interact with.

Act 1: `Search[Apple Remote]`

Obs 1: The Apple Remote is a remote control introduced in October 2005 by Apple ... originally designed to control the `Front Row media center` program ...

Thought 2: Apple Remote was originally designed to control the `Front Row media center` program. I need to search `Front Row` next and find what other device can control it.

Act 2: `Search[Front Row]`

Obs 2: Could not find [Front Row]. Similar: ['Front Row Seat to Earth', 'Front Row Motorsports', '`Front Row (software)`', ...]

Thought 3: `Front Row` is not found. I need to search `Front Row (software)` .

Act 3: `Search[Front Row (software)]`

Obs 3: `Front Row` is a discontinued media center software ...

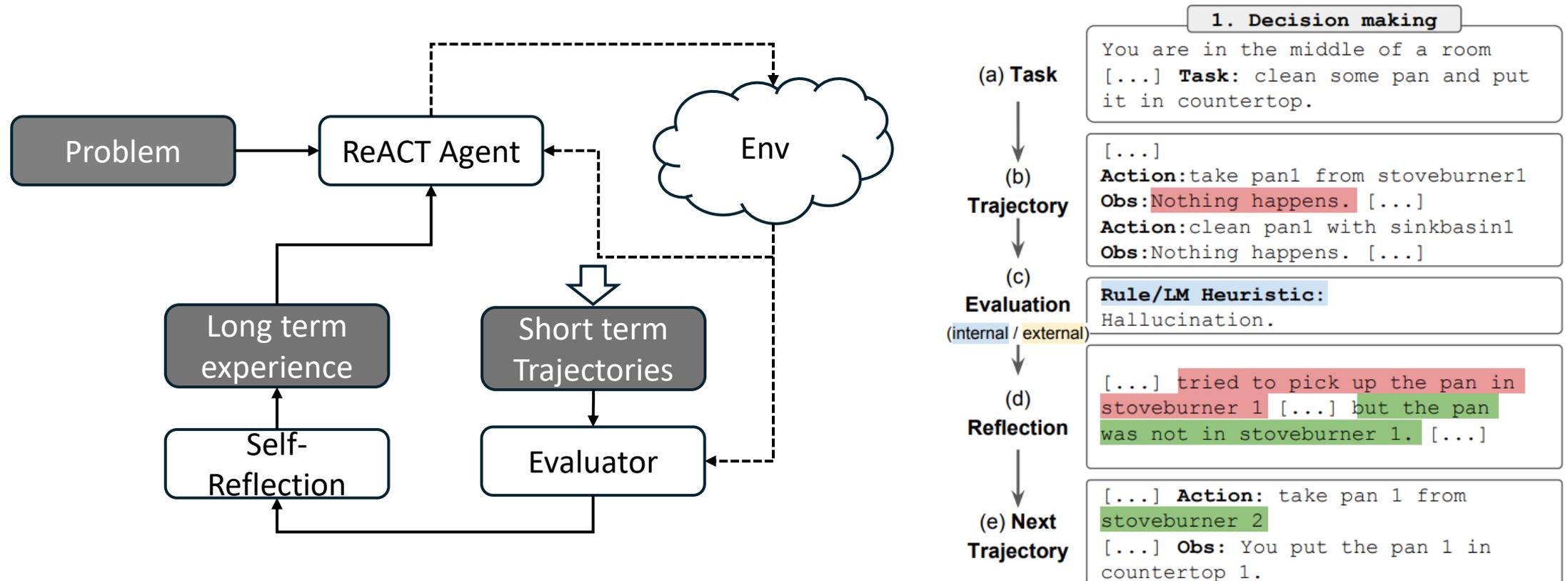
Thought 4: `Front Row (software)` is controlled by an Apple Remote or the keyboard function keys. So the answer is keyboard function keys.

Act 4: `Finish[keyboard function keys]`



Planning – With Feedback

Self-reflection: Plan with feedback from accumulated experience



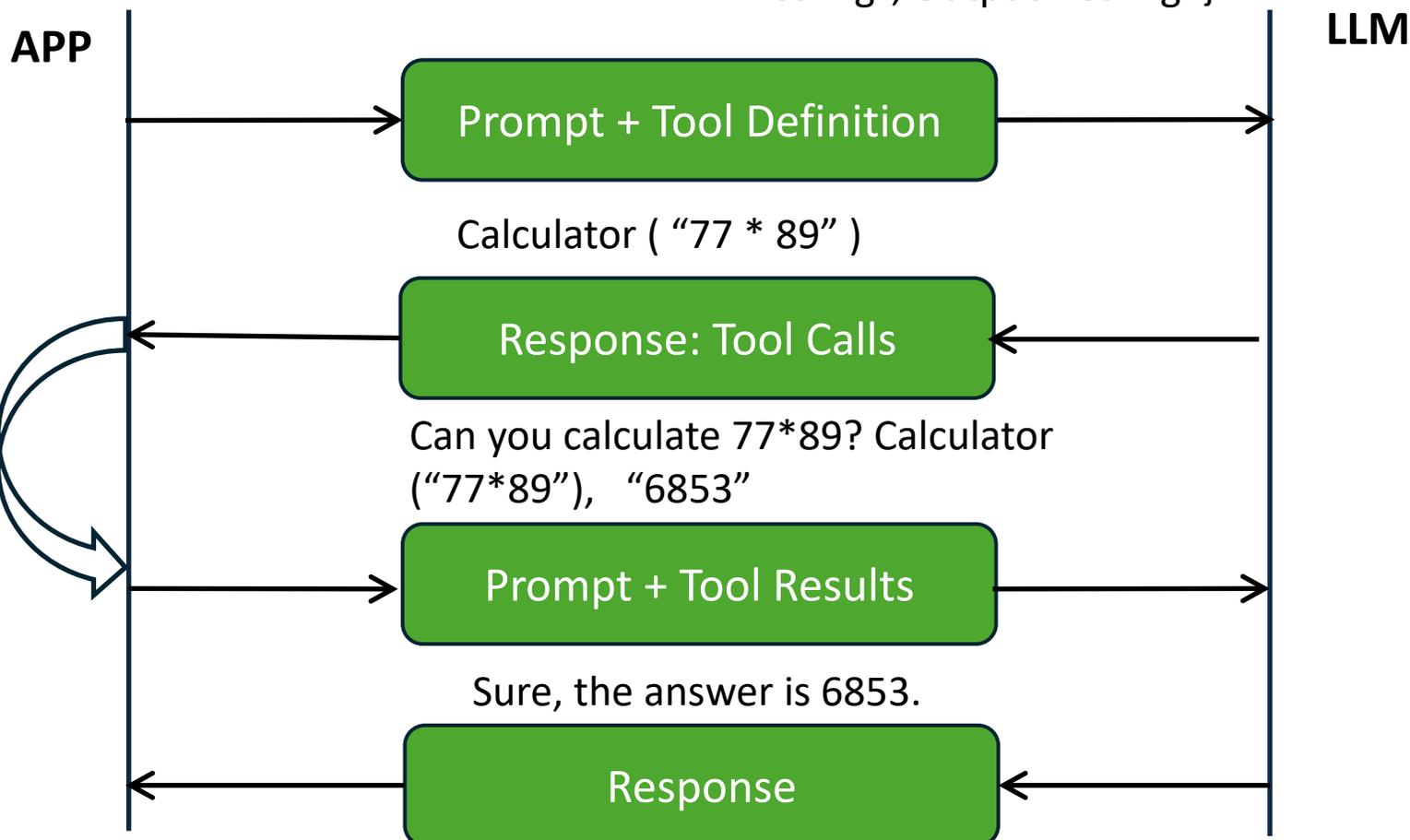
Memory

- Mechanics
 - In context → context limit
 - External files → archive
 - External vector DB → RAG
 - Structure memory (e.g. Knowledge Graph) → search
- Scope
 - Local → for single agent
 - Global → for all agents in the system
 - Short term/Long term
- Operation
 - Read/Write/Reflection

Tools

{API: "Calculator", Description: "calculate a math equation", Input: "string", Output: "string"}

Can you calculate $77 * 89$?



Execute Tools

$77 * 89 = 6853$

Two Agent Conversation

Connecting Agents to form conversations

Implicit task decomposition by roles

Your name is Joe and you are a part of a duo of comedians.



Cathy, tell me a joke.

Sure, here's one for you:
Why don't scientists trust atoms?
Because they make up everything!

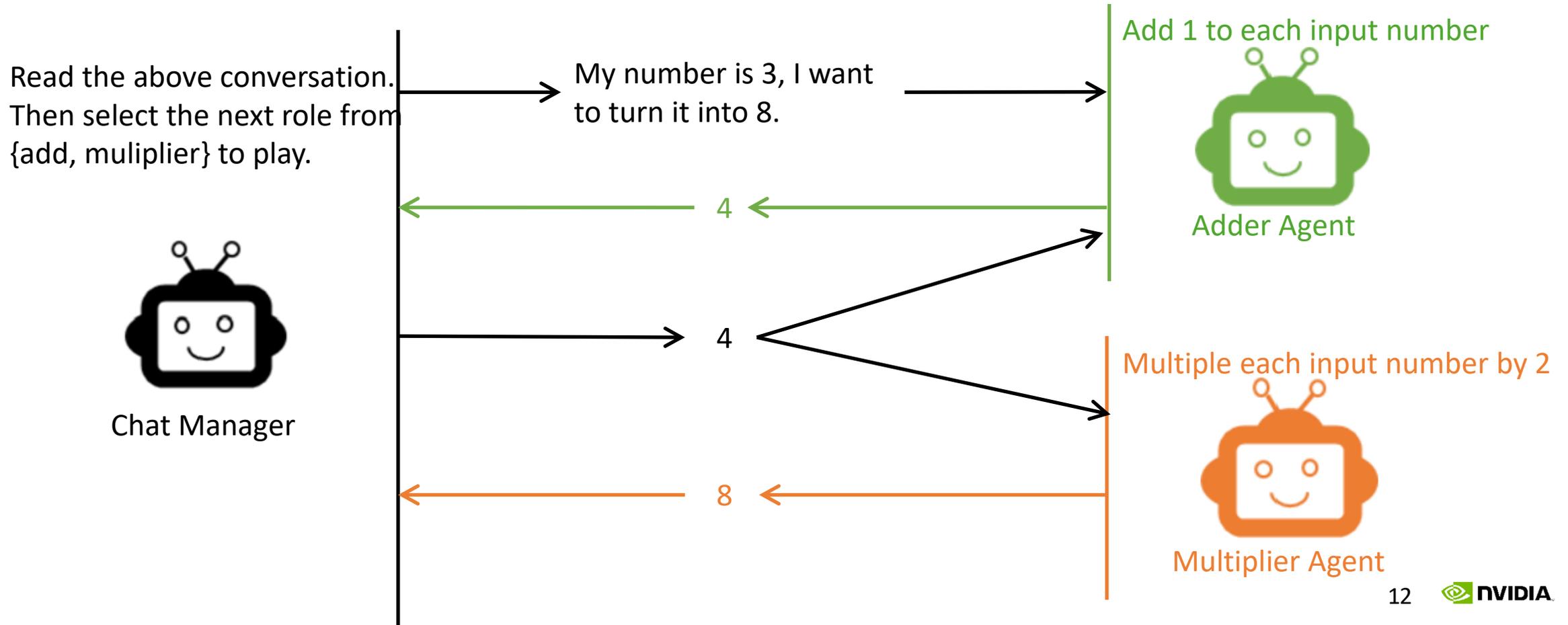
Good one, Cathy! Here's one for you:
Why was the math book sad?
Because it had too many problems!

Your name is Cathy and you are a part of a duo of comedians.



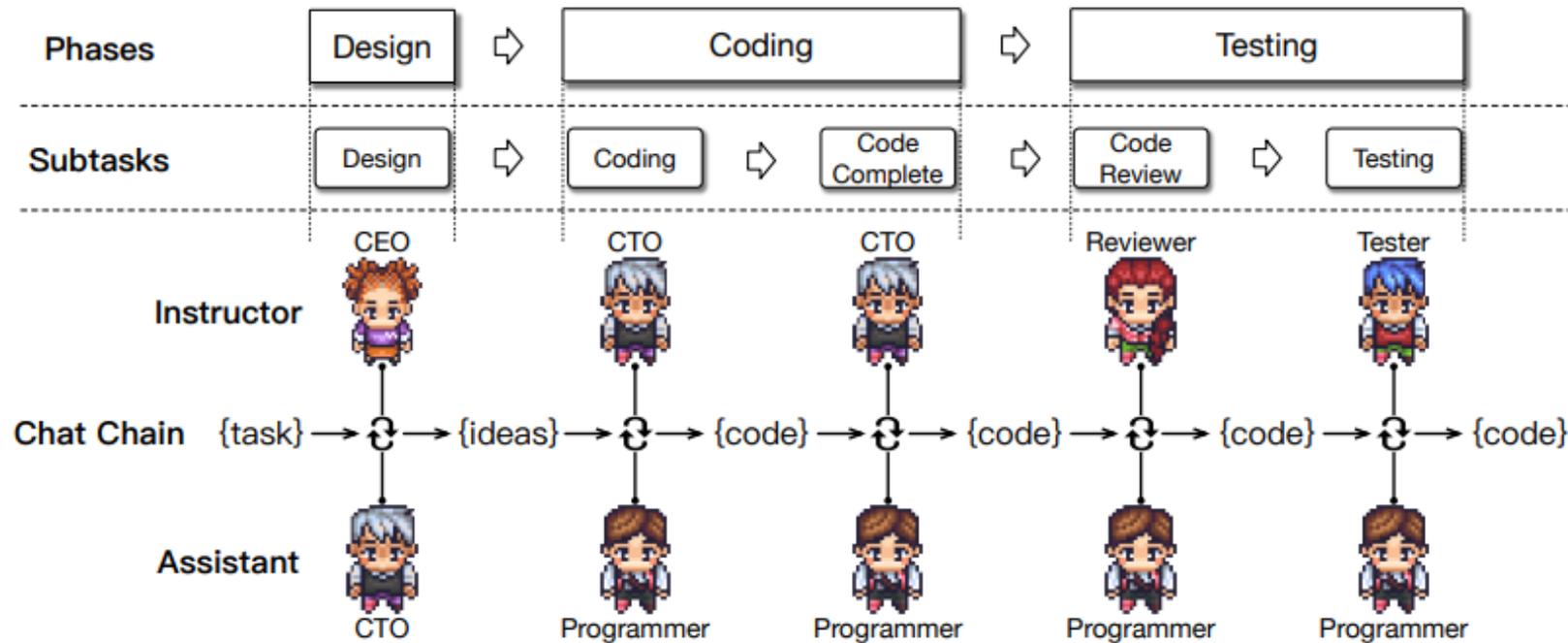
Group Chat

Chat manager orchestrates chat



Task flow

Sequence of multiple tasks implemented by agent chats
Explicit task decomposition



Timing Report Analysis

Task: Compare timing reports of two runs [datecode] to provide key takeaways and design QoR summary

Timing reports:

```

=====
|                               |                               |                               |
| Corner  datecode |WNS  TNS  FEP  | 0  | -0.05  -0.015  -0.02  -0.025
|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|c1      2024Mar13  -0.46 -8.155 63  | 9   1     4     2     2
|c1      2024Mar21  -0.432 -7.861 49  | 5   1     3     0     2
|c2      2024Mar13  -0.28  -1.453 22  | 0   3     0     2     0
|c2      2024Mar21  -0.254 -1.97  22  | 1   0     0     1     0

```

Analysis:

- The datecode <> generally shows improvement in TNS and FEP in all corners, ...
- However, the WNS degrades in most corners, ...
- The slack distribution shows improvement in all corners, ...
- The corner <> still suffers many timing violations,...

Tools and Prompts

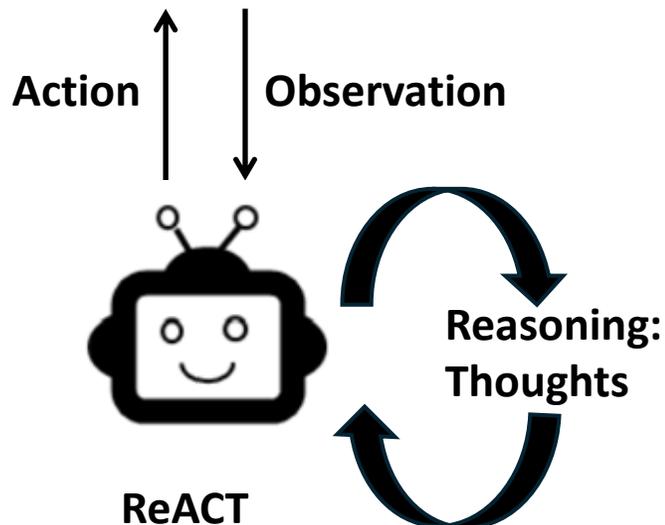
Tools

timing_metric_calculation_tool :

Calculate changes in WNS, TNS, or FEP

slack_distribution_calculation_tool :

Calculate changes in the slack distribution



CoT prompting

Let's think step by step.

Firstly, analyze the change of WNS, and TNS for all designs in all PVT corners tables of these two "datecode" settings and explain with numbers using tools.

Secondly, comparing "FEP" of two datecode settings in all PVT corners tables.

Thirdly, analyze the slack distribution to identify the distribution of "slack less than 0" paths.

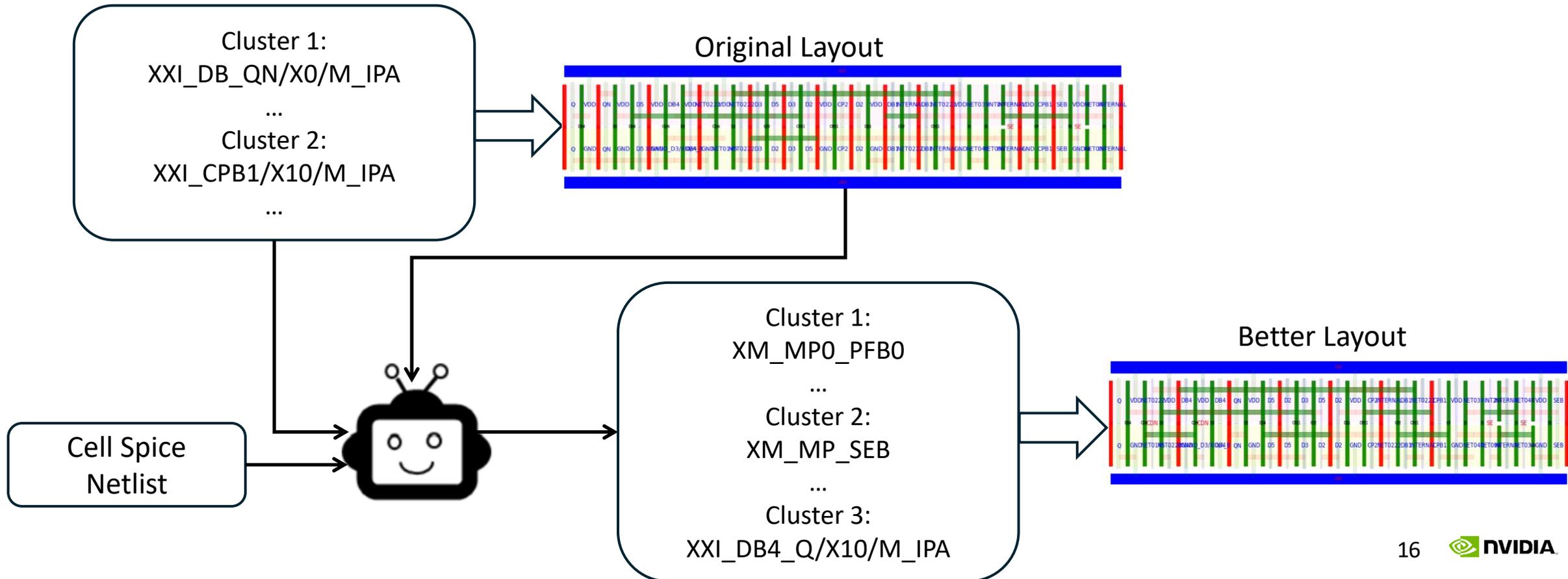
Finally, summarize the analysis in the following aspects:

- 1. Provide key takeaways, comparison, and suggestions with bullet points of the two "datecode" settings based on previous steps.*
- 2. Identify the corner which still suffers many timing violations if any.*

You need to use the **provided tools** to analyze the timing metrics!
You are not good at math!

Device Clustering for Standard Cell Layout

Task: Improve device clustering constraints for cell layout generator



Device Clustering Optimization Agent

Domain knowledge + ReACT

Domain knowledge extraction

Identify important and problematic nets with previous layout

Tools

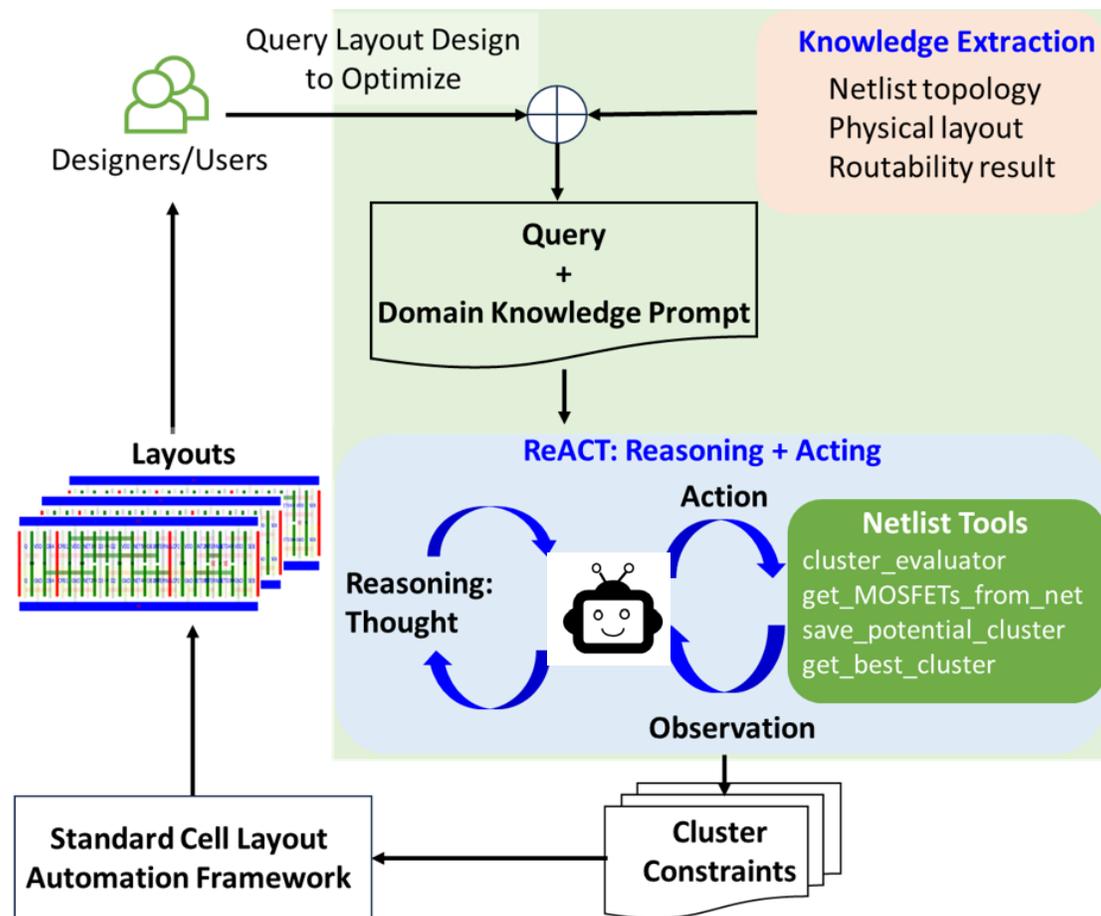
Cluster evaluator

Get group devices from nets

Save cluster (Memory)

Get the best clusters (Memory)

Achieved 5% area reduction



Device Clustering Optimization Agent

Domain knowledge:

Important Net: NET0222 (5 source and drain terminals, 3 PMOS, 2NMOS)

Action:

```
get_MOSFETs_from_net(NET0222)
# returns XM_MP1_PFB, ...
cluster_evaluation(XM_MP1_PFB,...)
```

Action:

```
save_cluster(XM_MP1_PFB,...)
```

Thought:

By identifying the nets with the most connections and resolving the issues with problematic nets, I can improve the score.

Observation:

*The MOSFETs connected to XM_MP1_PFB ...
Cluster score : 0.4875*

Thought:

I should save this potential cluster and continue searching for more potential clusters.

VerilogCoder Agent

Improve SOTA on VerilogEval(Human) from 60.3% to 94.2%

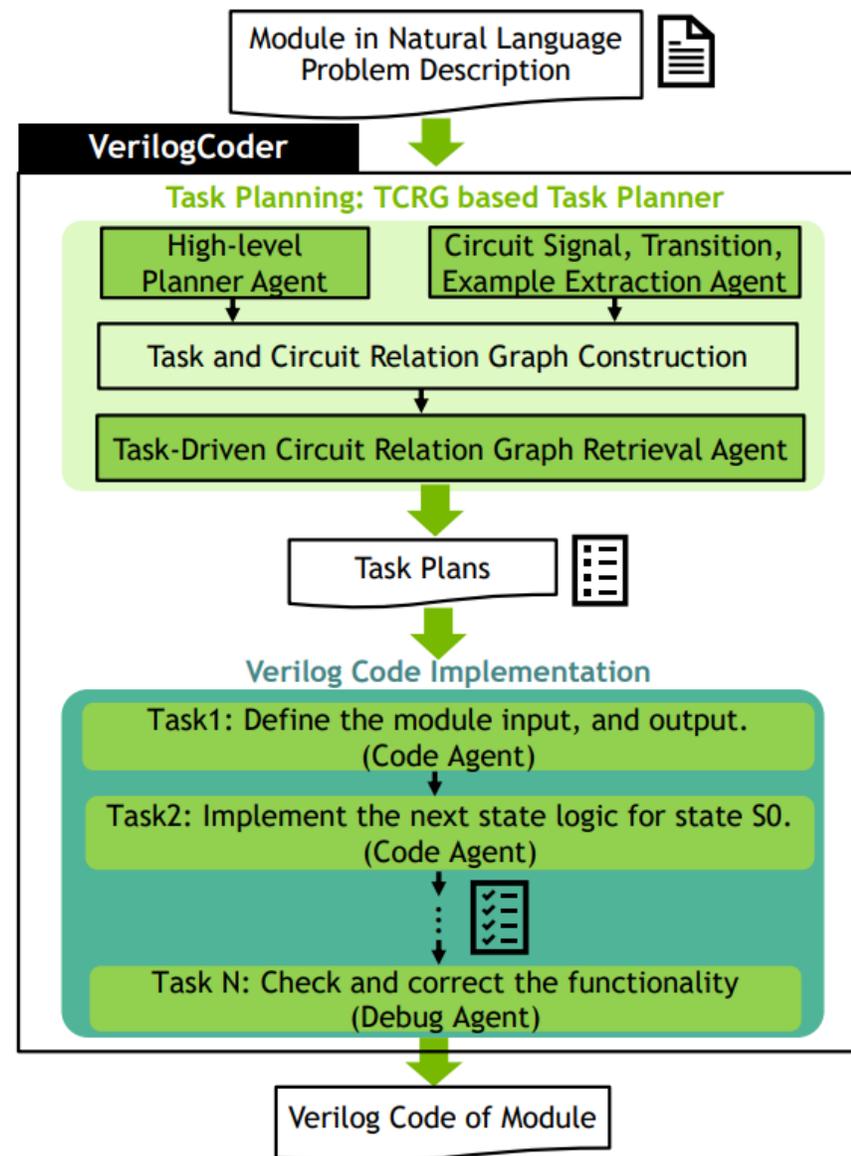
Task-Flow Driven Multi-Agents with specialized tools

Task Planning

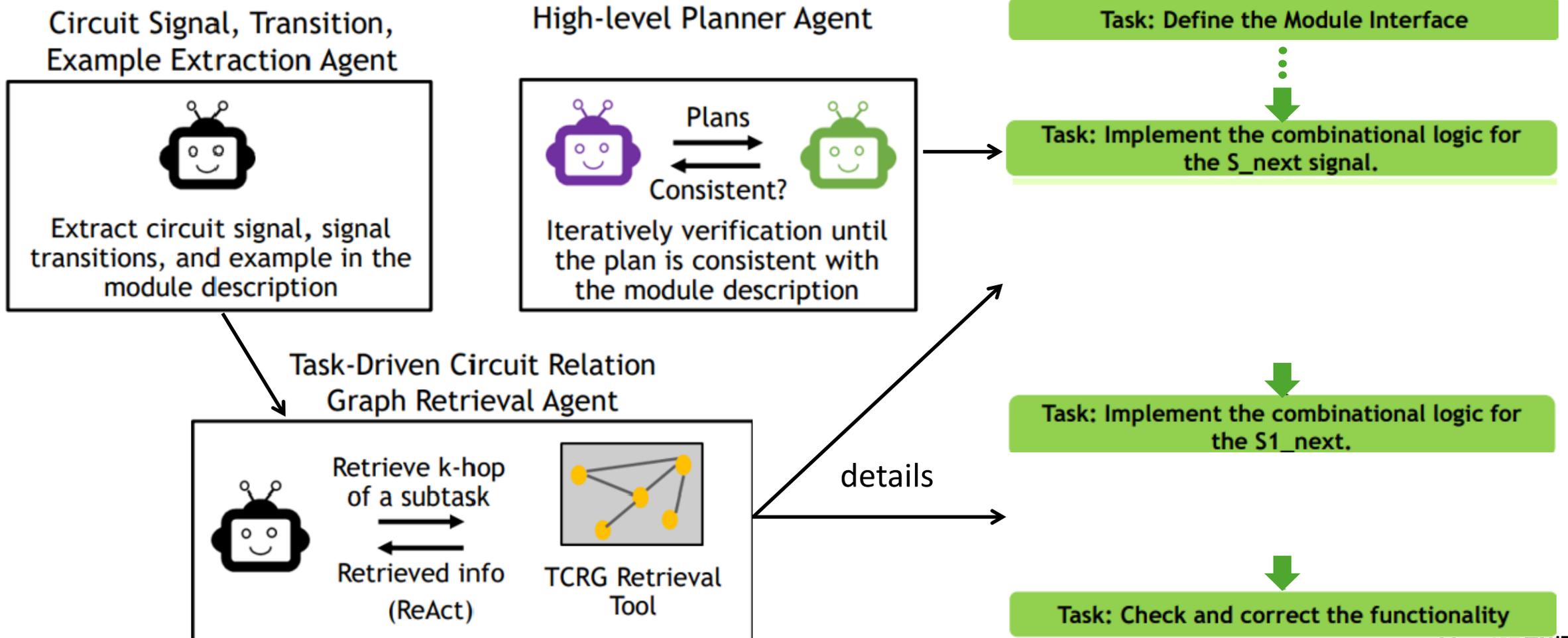
Task-Driven Circuit Relation Graph (TCRG)

Code Implementation

AST-guided waveform debugging tool



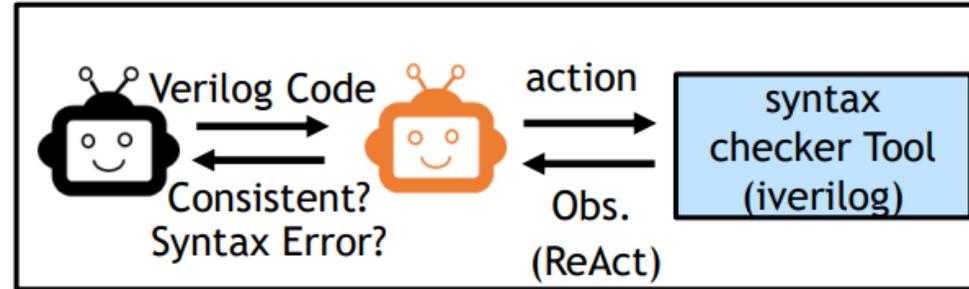
Task Planning



Code Implementation

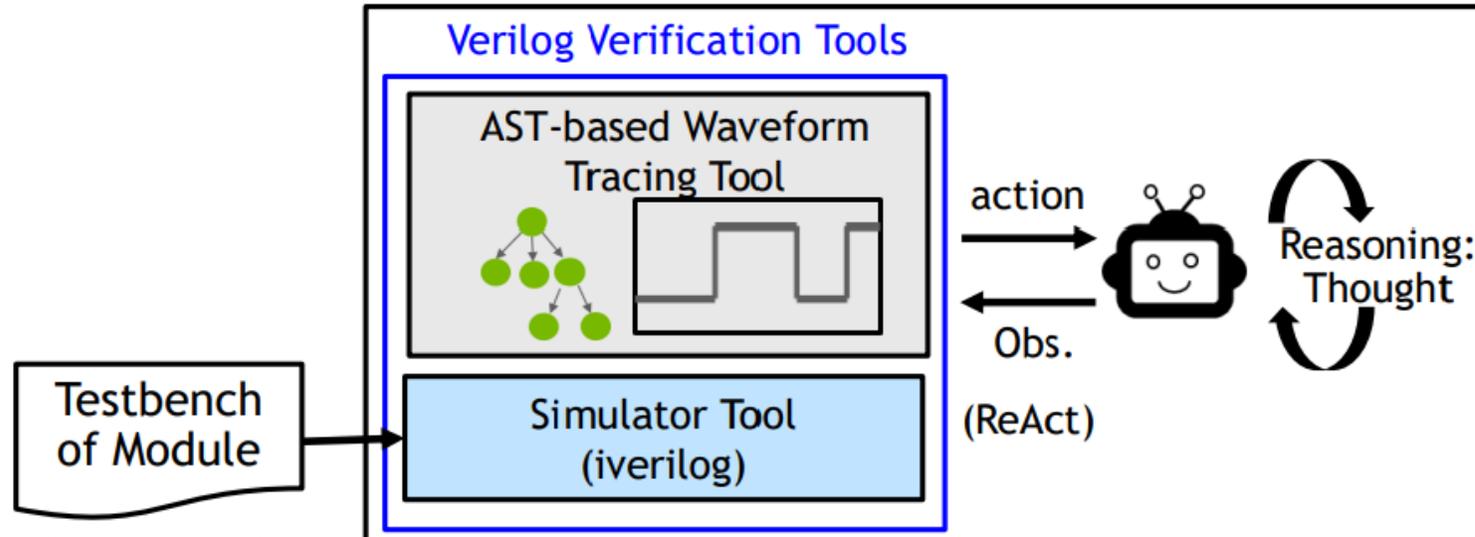
Code Agent: Write partial Verilog code

Writing code for each task



Debug Agent: Check and Correct the functionality

Debug final code



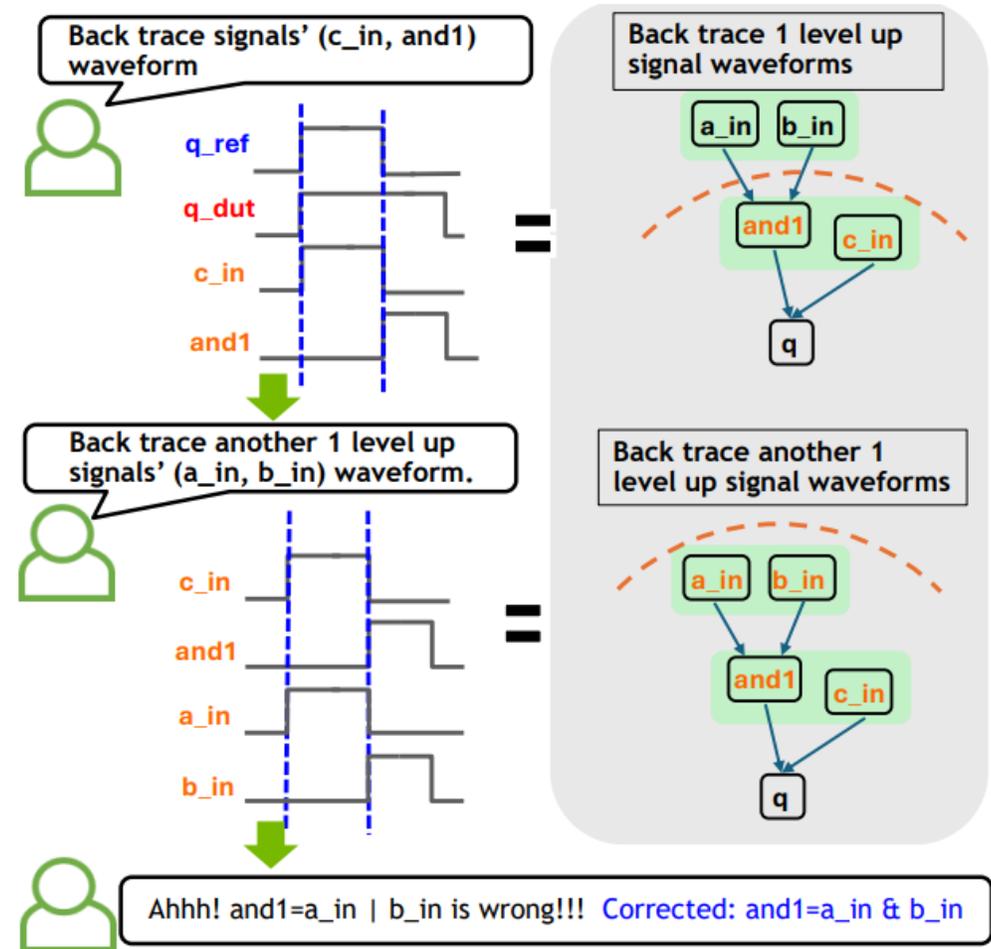
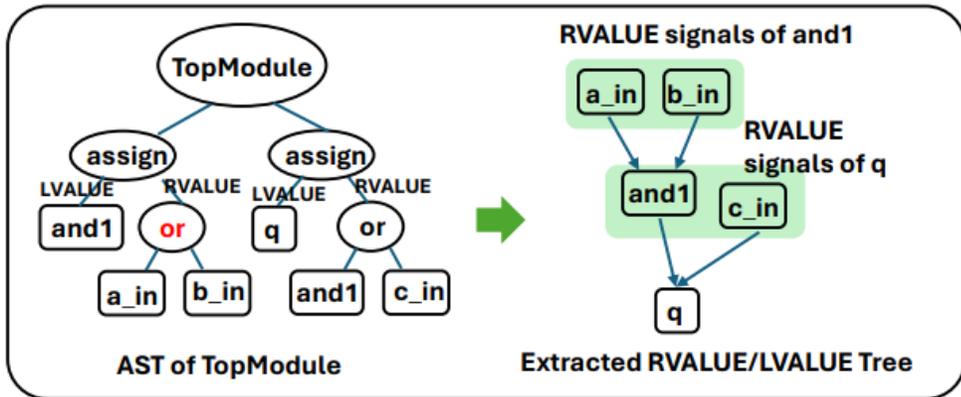
Debug with AST-Guided Waveforms

Use AST to extract relevant signal traces for debug with ReACT

```

module TopModule (
  input wire a_in,
  input wire b_in,
  input wire c_in,
  output wire q
);
  wire and1;
  assign and1 = a_in | b_in;
  assign q = c_in | and1;
endmodule
    
```

*** Simulator Output ***
 VCD info: dumpfile wave.vcd opened for output.
 Hint: Output 'q' has 12 mismatches.
 Simulation finished at 4526 ps
 Mismatches: 12 in 72 samples



Closing Thoughts

- What problems can be solved with agent today?
 - Design generation, debugging, review/analysis, optimization
 - Can be decomposed into tasks with low to medium intellectual effort
 - Increase productivity, make designer's life easier
- Extensive domain capability required to build useful agents for chip design
 - Diverse design knowledge: Logic design, physical design, verification, analog design, etc.
 - Diverse set of tool commands and file formats: PrimeTime, ICC2, Innovus, reports, LEF/DEF, etc.
- Agent capability comes from the designer's insights and specialized tools
- Can agent become intelligent enough to solve hard design problems?

