# **ChipAgentsBench**: Benchmarking AI Agents for Realistic Chip Design and Verification

kexun@chipagents.ai

ChipAgents AI

ChipAgents AI

- Started in June 2024, Series A Company.
- Backed by Bessemer Venture Partners, Micron, MediaTek, Samsung, and Ericsson.

- Locations:
  - Santa Barbara, California;
  - Santa Clara, California.

  Current Burn: 5+ years.

- **Hiring Plans:**
  - **2025:** ~30 team members
  - **2026:** ~50 team members
  - **2027:** ~100 team members
  - **2028:** ~200 team members

# Team

William Wang
Founder, CEO, and Chairman

Mellichamp Chair Professor of AI, UCSB +
Amazon AWS Bedrock / Amazon Q (2022-2024)

IEEE Laplace Award, BCS Karen Spärck Jones Award,
NSF CAREER Award, IEEE AI's 10 to Watch,
DARPA Young Faculty Award. PhD @ CMU

$25M previous gifts and contracts from Meta, Google, Amazon, Intel, JP Morgan, Adobe, NVIDIA, IBM, CISCO, Apple, etc.
Helped Apple shipped MGIE, and Google's Gemini.

Press: Venture Beat, Business Insider, Wired, GeekWire, Guardian, Fast Company, Fortune, Scientific American etc.

# ChipAgents: Board of Advisors

Wally Rhines
ex-CEO, Mentor Graphics
(Siemens EDA)

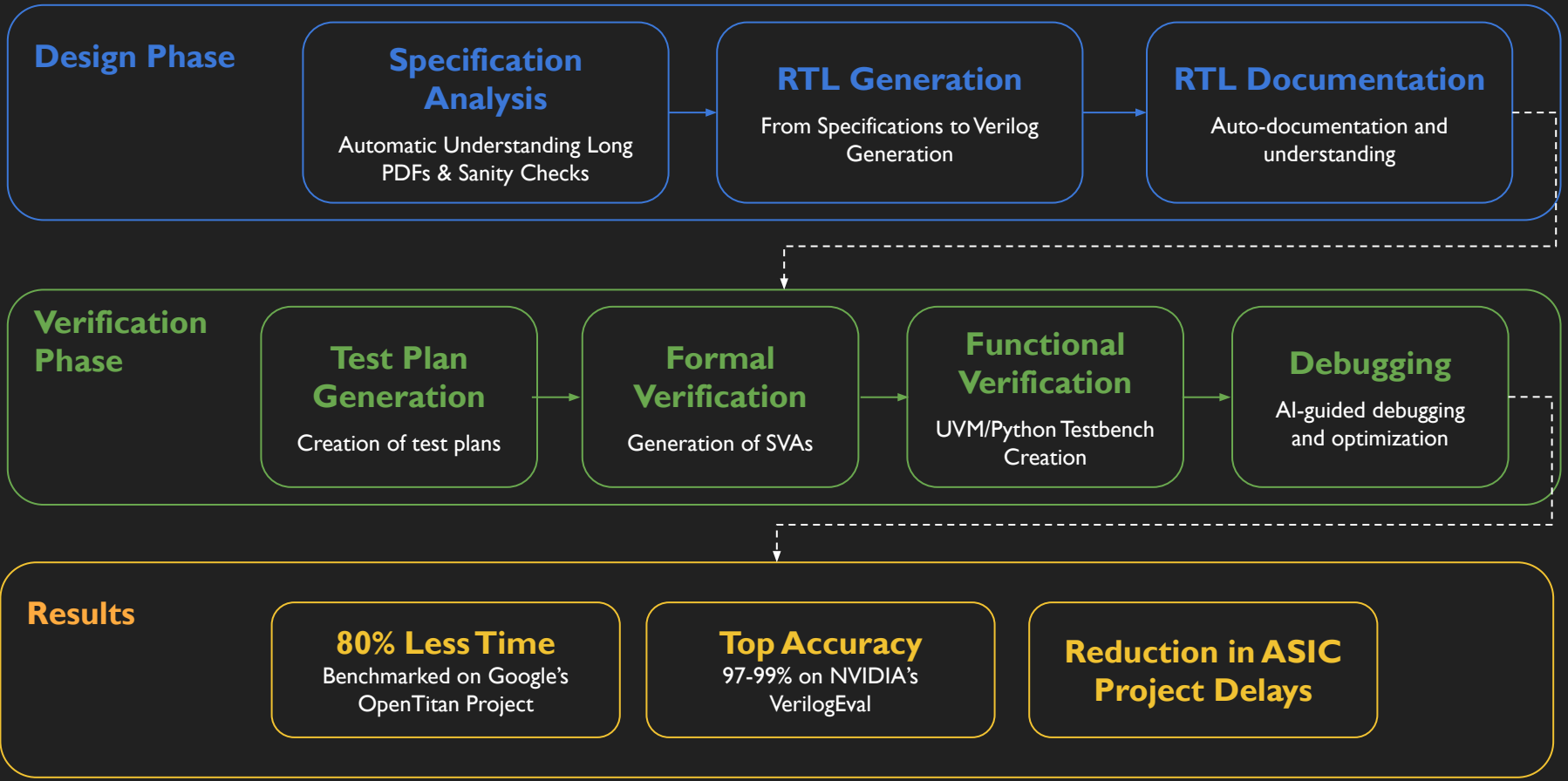
Raul Camposano
ex-CTO, Synopsys

Jack Harding
ex-CEO, Cadence

*"ChipAgents demonstrates the major impact that AI can have on the full range of integrated circuit design tasks". "I've met with three major semiconductor companies that have done competitive assessments of AI-based design solutions and ChipAgents is the number one choice at all three".* - Wally Rhines

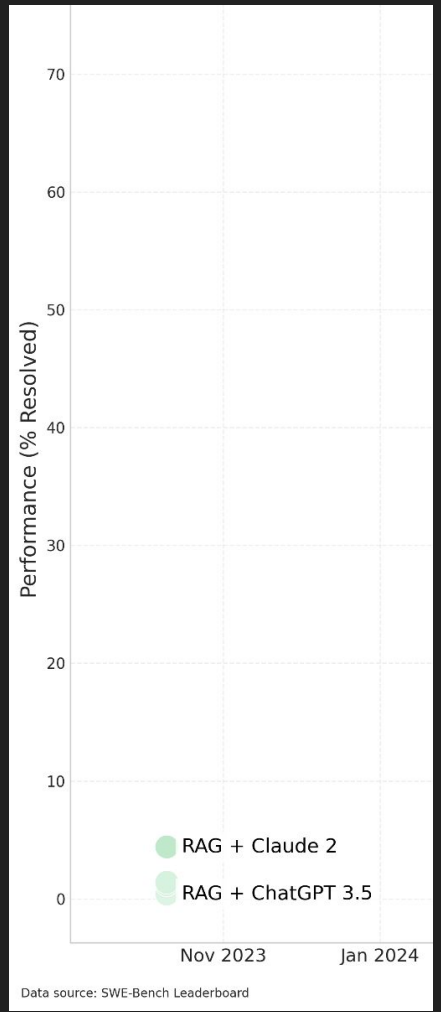
20 key AI, data, frontend, and verification engineers.

# ChipAgents™: Agentic AI Powered Design & Verification Flow



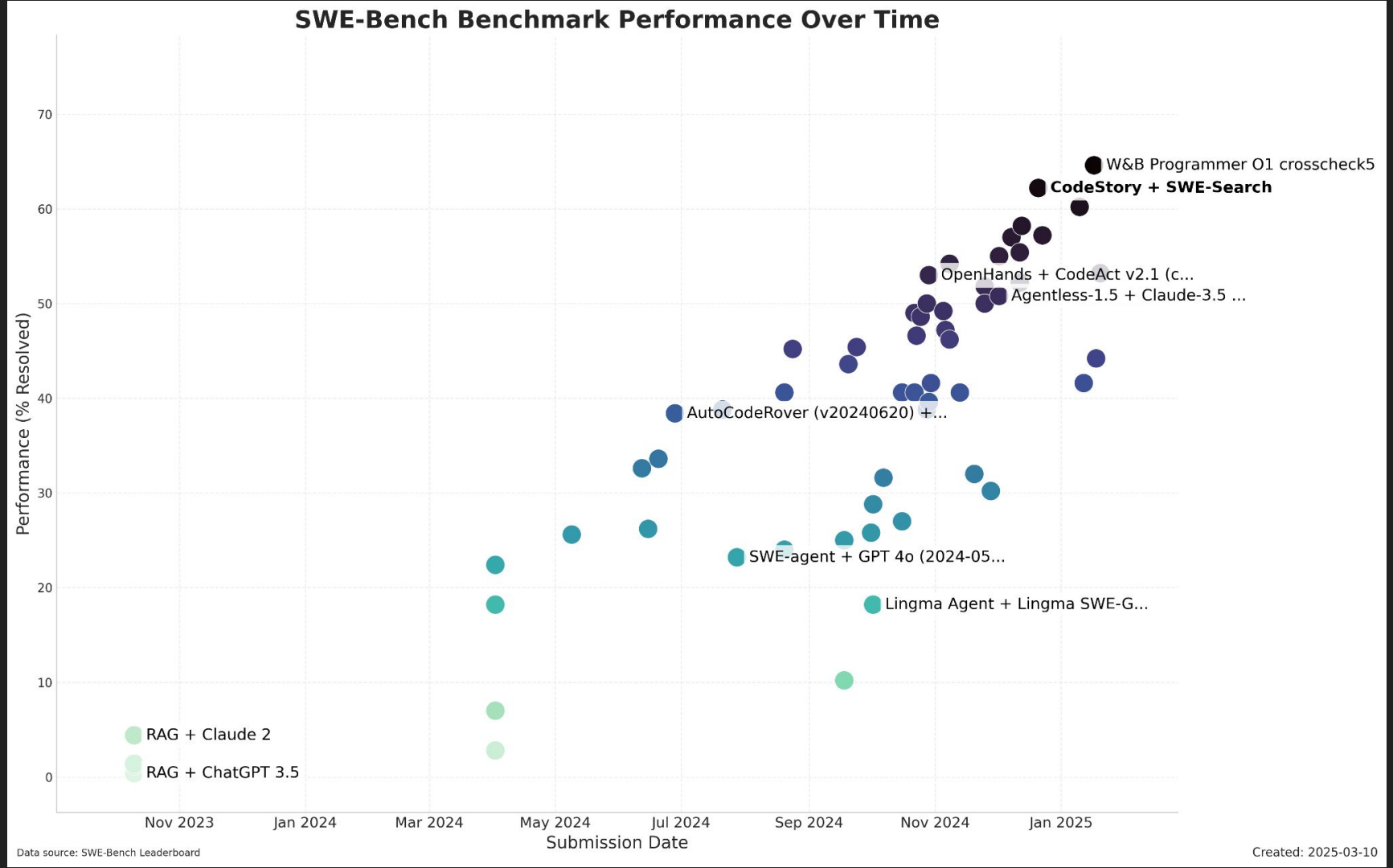
# Good benchmarks drive progress in AI.

What's happened in software …




## Expert Reviewer for SWE-Bench:

*"Starting with such low baseline model performance might ultimately prove to be an error - **the tests may simply not be realistic to solve**, but is worth the experiment."*


SWE-Bench Benchmark Performance Over Time
Performance (% Resolved)
Submission Date
0
10
20
30
40
50
60
70
Nov 2023
Jan 2024
Mar 2024
May 2024
Jul 2024
Sep 2024
Nov 2024
Jan 2025
W&B Programmer O1 crosscheck5
CodeStory + SWE-Search
OpenHands + CodeAct v2.1 (c...
Agentless-1.5 + Claude-3.5 ...
AutoCodeRover (v20240620) +...
SWE-agent + GPT 4o (2024-05...
Lingma Agent + Lingma SWE-G...
RAG + Claude 2
RAG + ChatGPT 3.5
Data source: SWE-Bench Leaderboard
Created: 2025-03-10

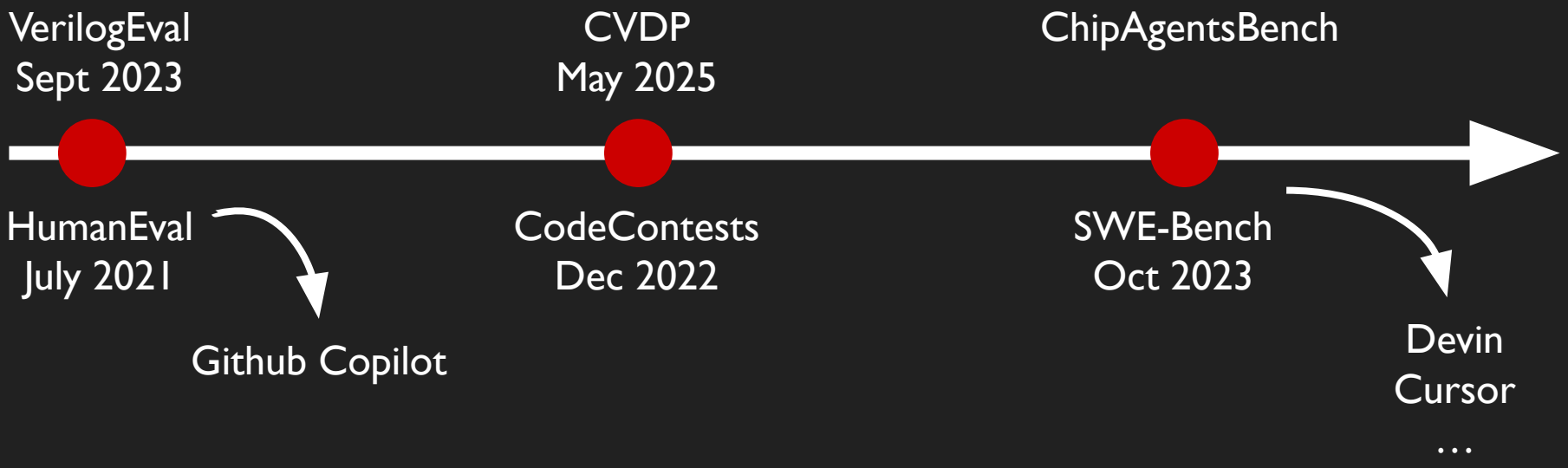
# Good benchmarks drive progress in AI.

What's happened in software …

# Good benchmarks drive progress in AI.

What's happened in software … is happening in chip design.

Huge shoutout to Mark Ren and the team!



# What's not enough about current benchmarks

Simple contexts, saturated tasks.

Lack of certain task types that represent industry practices.

# What's not enough about current benchmarks

**Simple contexts, saturated tasks:**

Example from VerilogEval →

State-of-the-art solution
already at **99.4%** accuracy.

**Question Prompt:**
Implement the Verilog module based on the following description. Assume that signals are positive clock/clk edge triggered unless otherwise stated.

**Problem Description:**
Given an 8-bit input vector [7:0], reverse its bit ordering.

```
module top_module (
        input [7:0] in,
        output [7:0] out
);
```

**Canonical Solution:**

```
  assign {out[0],out[1],out[2],out[3],out[4],
  ↪  out[5],out[6],out[7]} = in;
endmodule
```

# What's not enough about current benchmarks

**Simple contexts, saturated tasks:**

Example from CVDP →

Frontier models already at
**55%+** accuracy.

**Input Prompt**

Complete the existing `sorting_engine` module given below to implement the **brick sort** algorithm using finite state machine (FSM).

*[Brick sort description, algorithm example, and port list are omitted due to space constraints]*

**Parameters**
- `N` (Default is 8, Greater than 0): Number of elements to sort. Assume `N` is an even integer
- `WIDTH`(Default is 8, Greater than 0): Bit-width of each input element

**Latency Considerations**
Total latency = **(N * (N - 1)) / 2 + 4**
Perform a single compare-and-swap operation per clock cycle (sequential approach):
- **1 clock cycle** for moving from `IDLE` state to `LOAD`.
- **1 clock cycle** to load the data.

# What's not enough about current benchmarks

**Lack of certain task types:**

There is test generation, but no UVM generation.

There is debugging, but no waveform debugging.

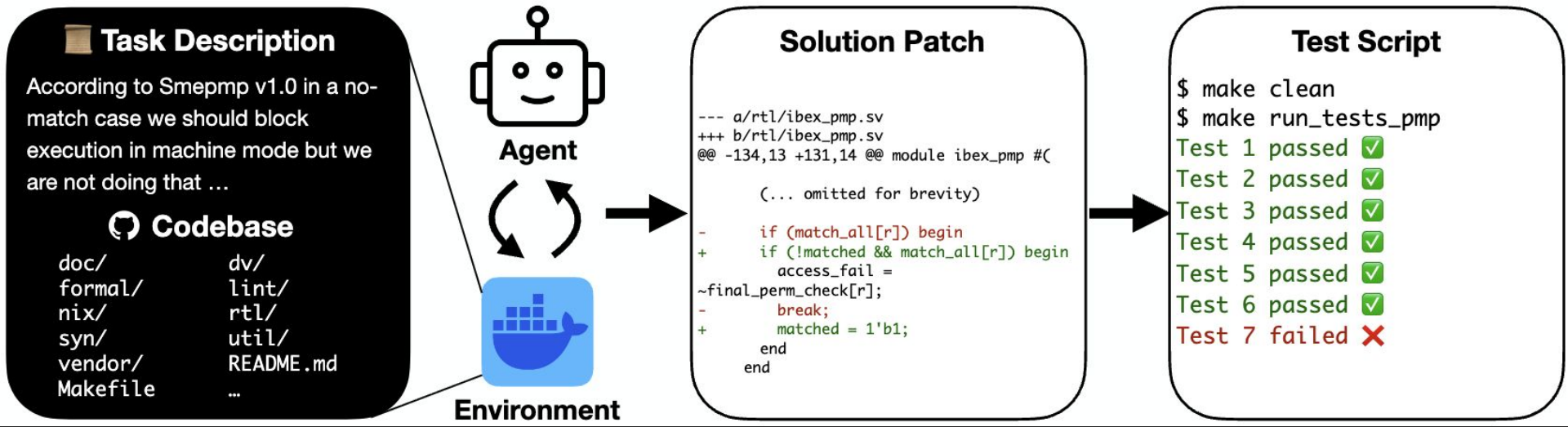
# ChipAgentsBench

- Agentic tasks with complex file structures
- Significantly larger context, **80x** more lines of code and **30x** more files
- New task types like **waveform debugging** and **UVM generation**
- Best open-source agents perform <25%.

| | RTL gen. | debug | optimization | test gen. | UVM gen. | waveform debug | lines | files |
|---|---|---|---|---|---|---|---|---|
| VerilogEval v2 | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | 22.9 | 1 |
| RTLLM v2 | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | 26.8 | 1 |
| RTLRewriter | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | 91.7 | 1.6 |
| CVDP (non-agentic) | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | 128.3 | 1 |
| CVDP (agentic) | ✓ | ✓ | ✗ | ✓ | ✗ | ✗ | 363.6 | 3.7 |
| CHIPAGENTSBENCH | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | **30373.2** | **112.2** |

# ChipAgentsBench

Each task is based on a complex, open-source project,
with an interactive Docker environment,
and corresponding human-written tests.

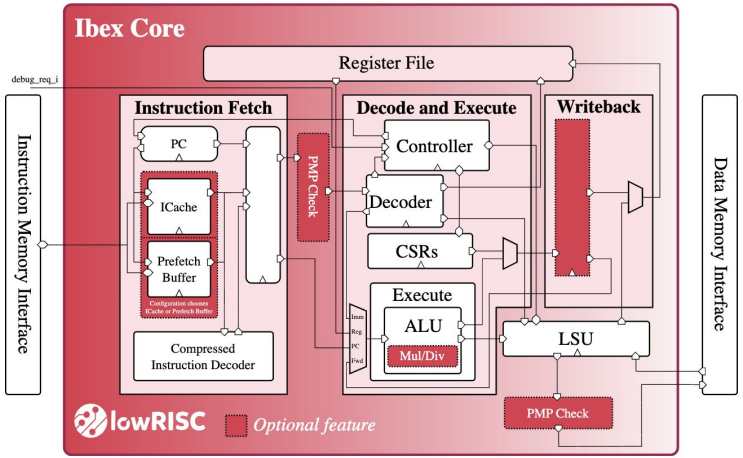


# ChipAgentsBench: Example Task

**Description:** *According to SMEPMP v1.1, in a no-match, in a no match case we should block execution in machine code but we are not doing that.*

## Ibex RISC-V Core

Ibex is a production-quality open source 32-bit RISC-V CPU core written in SystemVerilog. The CPU core is heavil
parametrizable and well suited for embedded control applications. Ibex is being extensively verified and has seen
multiple tape-outs. Ibex supports the Integer (I) or Embedded (E), Integer Multiplication and Division (M),
Compressed (C), and B (Bit Manipulation) extensions.




Ibex was initially developed as part of the PULP platform under the name "Zero-riscy", and has been contributed to lowRISC who maintains it and develops it further. It is under active development.

# ChipAgentsBench: Example Task

**Description:** *According to SMEPMP v1.1, in a no-match, in a no match case we should block execution in machine code but we are not doing that.*

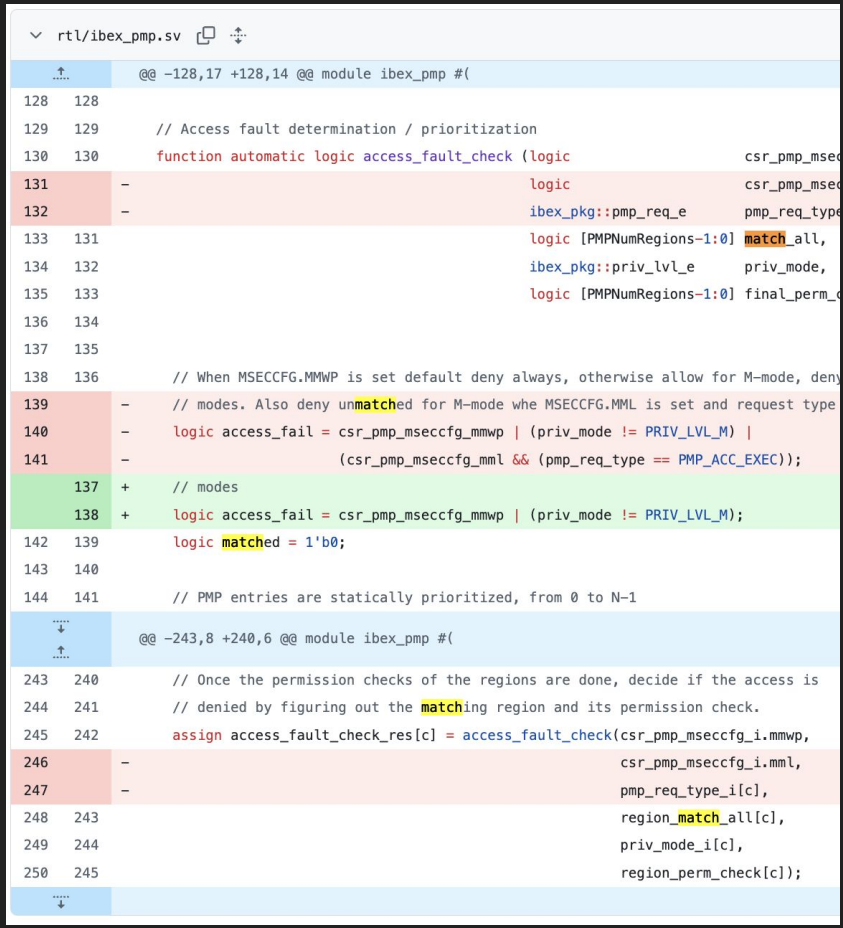
The agent needs to:

- Navigate the project with 603 SV files and complex file hierarchy
- Figure out how to get error logs by running `make TEST=riscv_pmp_full_random_test`
- Analyze logs and waveforms
- Locate the fixes inside rtl/ibex_pmp.sv

# ChipAgentsBench: Example Task

After project understanding, fault localization, iterative debugging and testing, the agent proposes a *patch*.

The patch will then be evaluated with human-written tests.



rtl/ibex_pmp.sv

```
@@ -128,17 +128,14 @@ module ibex_pmp #(
128 128
129 129     // Access fault determination / prioritization
130 130     function automatic logic access_fault_check (logic                    csr_pmp_msec
131     -                                                logic                    csr_pmp_msec
132     -                                                ibex_pkg::pmp_req_e      pmp_req_type
133 131                                                  logic [PMPNumRegions-1:0] match_all,
134 132                                                  ibex_pkg::priv_lvl_e     priv_mode,
135 133                                                  logic [PMPNumRegions-1:0] final_perm_c
136 134
137 135
138 136       // When MSECCFG.MMWP is set default deny always, otherwise allow for M-mode, deny
139     -     // modes. Also deny unmatched for M-mode whe MSECCFG.MML is set and request type
140     -     logic access_fail = csr_pmp_mseccfg_mmwp | (priv_mode != PRIV_LVL_M) |
141     -                         (csr_pmp_mseccfg_mml && (pmp_req_type == PMP_ACC_EXEC));
    137 +     // modes
    138 +     logic access_fail = csr_pmp_mseccfg_mmwp | (priv_mode != PRIV_LVL_M);
142 139       logic matched = 1'b0;
143 140
144 141       // PMP entries are statically prioritized, from 0 to N-1
@@ -243,8 +240,6 @@ module ibex_pmp #(
243 240       // Once the permission checks of the regions are done, decide if the access is
244 241       // denied by figuring out the matching region and its permission check.
245 242       assign access_fault_check_res[c] = access_fault_check(csr_pmp_mseccfg_i.mmwp,
246     -                                                         csr_pmp_mseccfg_i.mml,
247     -                                                         pmp_req_type_i[c],
248 243                                                           region_match_all[c],
249 244                                                           priv_mode_i[c],
250 245                                                           region_perm_check[c]);
```

# Results and Analyses with Open-Source Agents

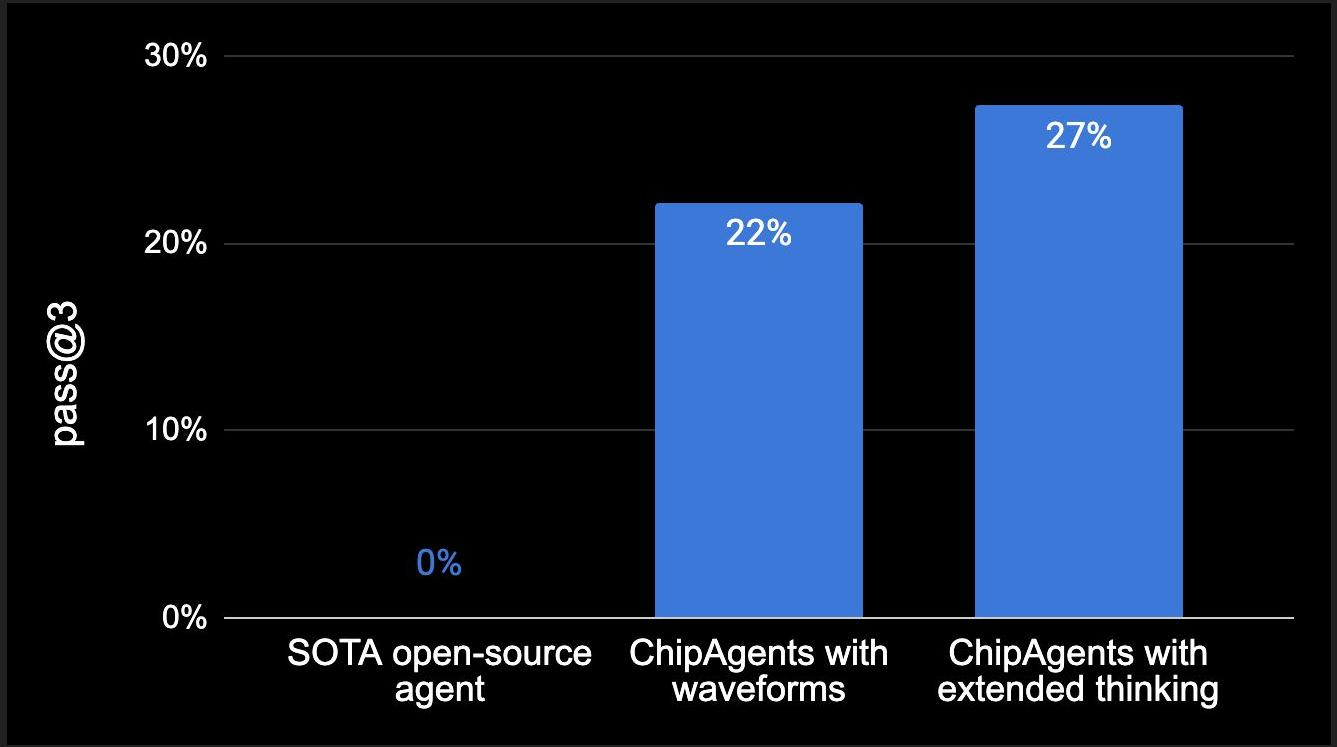
State-of-the-art open-source agents perform poorly on ChipAgentsBench, with **<25% pass rate**.

Generating boilerplate UVM such as scoreboard is easy, with **50% pass rate**.

Generating UVM sequencer/stimuli for good coverage is hard.

Complex debugging with waveforms is extremely hard.

# Reading Waveforms is Important for Complex Debugging



# Caution: Reward Hacking

When the testbench is accessible to an agent, it might modify the testbench to make its incorrect RTL pass.

# Coming soon…

We will:

- Open-source a subset of the tasks while retaining others to avoid contamination
- Open-source the test harness so people can run evaluation
- Maintain a leaderboard for open-source agents

We welcome community contributions and evaluation requests.