Using trace-based performance models to accelerate customer evaluations

Knute Lingaard[1], Nasr Ullah[1], Sam Grove[1] and Jon Taylor[2]

[1]MIPS
[2]Synopsys

Abstract

Trace based models are higher levels of abstraction than true design data but incorporate descriptions of micro architectural features such as pipeline, bus or memory system behavior. They typically cannot execute code directly but take an execution trace from a model such as an ISS (which is architecturally accurate but does not have timing information). Together they can be used to estimate application performance before the design is complete. These complimentary technologies are extremely important in their own right, however, when used together create a powerful system that facilitates hardware-software co-design bringing software engineers and micro-architects together using the same technology that each discipline is already familiar with. Both the ISS and trace-based models are commonly developed ahead of RTL. In the context of IP licensing, customers can provide an execution trace to the IP company; the IP company does not need to share or productize the trace-based model, while the customer doesn't need to share the software or source code for workloads. RISC-V international has a working group developing the STF trace format designed for exactly this use case.

This paper describes how MIPS and Synopsys have developed a flow that can easily be deployed and scaled to support customer evaluation of MIPS processors.

Introduction

When bringing a new CPU to market, there can be a catch-22 problem of customers wanting performance data, but the design not being finished so RTL is not available to simulate the design. There can also be questions of confidentiality – a designer may be unwilling to share their RTL (since it's the entire design database), while similarly a customer may not want to share their application code with the IP provider.

Developing cycle accurate models of CPUs is challenging – it can be as complex as developing the RTL design itself to achieve a model that is fully accurate. Even if they are developed, it may be that they are suitable for use by the designers, but not easily made into a product that a customer can use.

MIPS Atlas Explorer combines an ISS, performance models and performance visualization tools that enable customers to examine software workloads with micro-architectural execution data on MIPS RISC-V processors.

Synopsys offers ImperasFPMs – fast, instruction accurate models of RISC-V processors. They can be customized to match individual designs, including custom instructions and status registers. They are well proven in use, and Synopsys worked with MIPS to add the ability to generate STF trace from these models.

STF trace format

The Simulation Trace Format (STF) trace offers the ability to trace the architectural state of a workload in a low overhead and efficient format. The instruction set architecture (ISA) agnostic format captures information relating to instructions, associated registers, memory data and data addresses associated with, page table walks, interrupts and bus or fabric transactions. The native binary format in conjunction with compression makes capturing and storing this data disk (and compute) friendly.

Flow

The expected development flow is as follows. The customer can port, migrate, and debug software workloads using ImperasFPM to get their code running on a model of the MIPS RISC-V processors. Once the code is functional, developers can collect execution-based performance data to determine how well the workloads runs on a specific processor. Developers can then use this data to compare the impact of compiler flags, code refactoring, data type translation and other optimization techniques to compare the trade-offs for each. Alternatively, customers could compare the workloads execution on one processor versus another processor. Ultimately, we have presented a robust and

scalable method for using software to drive and improve hardware and a solution for software developers to enhance software for existing hardware in an efficient way using technology that is familiar to each.

Future work

Additional use cases and advanced solution information and disclosures are TBD right now but might be able to be shared at the event in May'25.

Conclusions

Combining a fast architectural simulator with a trace-based pipeline model provides an efficient way to measure approximate performance without needing to develop a cycle accurate model. This way customers can preserve the confidentiality of their code while also being able to extract useful performance data. Similarly, MIPS is able to engage earlier and more easily with customers to evaluate performance of their applications, make late design decisions based on actual software workloads and without having to share proprietary design details with potential customers.

References

STF Specification (https://github.com/sparcians/stf_spec/blob/master/generated/stf-spec-github.adoc)