Hardware / Software Engineers – CPU development Verification, Modelling & Performance Analysis

Arm Sophia Antipolis
Le Paros – 25 allée Pierre Ziller
Sophia-Antipolis
06560 VALBONNE
Tel: 04.97.23.51.00

[2018-V-1] Detect data corruption on partial memory dump
[2018-V-2] Enable ARM CPU verification technique on FPGA
[2018-V-3] FPGA Checksum hardware accelerator
[2018-V-4] How good are my formal verification abstractions?
[2018-V-5] Identification of a "good" set of assertions using machine learning or other approaches
[2018-V-6] Machine learning for testbench stress optimization

[2018-M-1] Loop instruction buffer microarchitectural investigations
[2018-M-2] Development of a fast simulation mode in a micro-architectural CPU model
[2018-M-3] CPU model integration in a modular platform simulator
[2018-M-4] A Top-Down Method for Performance Analysis and Counters Architecture

Essential skills:
Different set of skills and knowledge are required depending on the selected internship topic. These should desirably include:

1- Microprocessor, ASIC systems
2- HDL and Synthesis design knowledge
3- Programming language (C/C++, Python, …)
4- Use of UNIX and shell programming
5- Strong analytic skills
6- Ability to express ideas and communicate effectively

Interested in the latest processor technologies? Join us at: **http://www.arm.com/about/careers/students/student-vacancies.php**

Functional & Formal Verification
goo.gl/SYKddH

[2018-V-1] Detect data corruption on partial memory dump
Checking data consistency is one of the biggest challenge in multi-processors verification environments as a large amount of transactions is generated and needs to be processed as fast as possible. The goal of this internship is to be able to check the correctness of partial memory dumps. This will lead to study the Arm Memory Model, challenge algorithms and database system to seek for the best performance in different types of environments (Arm execution traces, Emulator, FPGA, ...).

[2018-V-2] Enable Arm CPU verification technique on FPGA
Running tests on FPGA is a good way of improving CPU verification speed and efficiency, at the cost of debug and checking abilities. A capture technique exists but doesn't fulfill complex system needs. The goal of this internship is to study, define and implement a smart data capture that enables high value checkers to operate in an FPGA environment.

[2018-V-3] FPGA Checksum hardware accelerator
When developing a new CPU, lots of computation time is used in FPGA farms to check its correct behaviour. The goal of this internship is to study how to implement a hardware accelerator to speed up this part. This will require to study and compare different algorithms, and to implement one as an IP in the FPGA environment, and to check its benefits.

[2018-V-4] How good are my formal verification abstractions?
Formal verification is one of the best techniques to address complex bug hunting. But the complexity of the design makes it hard to cover all its states. Abstractions ease this process. The goal of this internship is to find a way to evaluate existing abstractions, and implement, or generate, better ones.

[2018-V-5] Identification of a "good" set of assertions using machine learning or other approaches
Assertions are a very powerful way of checking the correctness of a design. However, they have an impact on performances on various verification environments, and nobody is going to accept to remove any of them, unless... The goal of this internship is to define a metric to measure the pertinence of an assertion set, and implement a software tool to select the best subset. To achieve this, any approach, including machine learning and formal tools, can be explored.

[2018-V-6] Machine learning for testbench stress optimization
The goal of this internship is to implement machine learning capabilities to an existing CPU verification framework. The end goal is to set up a self-improving system aiming at optimizing the generation of complex verification tests to increase their stress level.

CPU Modelling & Performance Analysis
goo.gl/6tU47g

[2018-M-1] Loop instruction buffer microarchitectural investigations
In a modern superscalar CPU, loop instruction buffer is a key element of performance and power efficiency. Short repetitive sequences can be cached in the buffer, limiting the use of the front-end stages of the pipeline. The student will study the state-of-the-art techniques, identify and implement a possible micro-architecture in an accurate Arm CPU model, and analyze its performance and power efficiency.

[2018-M-2] Development of a fast simulation mode in a micro-architectural CPU model
Various techniques exist to work around the speed limitation of accurate, detailed micro-architectural CPU models, and reach the regions of interests in large workloads and benchmarks. One possibility is to restore the state of a simulation previously run (a checkpoint) and re-start from that point. One, complementary solution consists in adding a purely functional fast simulation mode. The aim of the internship is to study the existing literature and implement support for these two techniques in an existing Arm CPU model.

[2018-M-3] CPU model integration in a modular platform simulator
The goal of this internship is to integrate an existing Arm CPU model into a generic, modular platform simulator, and assess the CPU performance in constrained environments. After the usual ramp-up to get familiar with the various components, the student will instantiate the CPU in the platform simulator and optimize its integration. In a final stage, the student will run various benchmarks in this environment and assess the system impact on the CPU performance.

[2018-M-4] A Top-Down Method for Performance Analysis and Counters Architecture
Benchmarking and performance analysis are part of the most significant activities that drive the development of high-end processors.
The objective of this internship is to develop a new top-down and hierarchical method that would faster point performance engineers to the right areas to investigate.
The intern will first get familiar with an existing CPU design, with the current performance analysis infrastructure, and with the state-of-the-art performance analysis and counter architecture.
Then, the intern will develop a structured, top-down, performance counter architecture to faster point engineers to the right areas to investigate.
Finally, he/she will validate the improvements in the anomaly detection process by collecting performance results on industry standard benchmarks and exporting the solution to other Arm processors.