Join our CPU Performance Analysis, Modelling and Verification teams in Arm Sophia-Antipolis. At the crossroad between software and hardware, you will join an international team of talented and passionate engineers that is shaping the future of technology.

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

1. Use of UNIX and shell programming
2. Keen interest in Computer Science
3. Software (Python/C/ C++) or Hardware (VHDL/Verilog) knowledge

Interested in the latest CPU technologies? Join us at: [https://www.arm.com/company/careers](https://www.arm.com/company/careers)
[2021-V6] – CPU RTL versus model: correlation beyond testing

Many crucial algorithms for CPU performance are first implemented in a C model for investigations. Then, significant effort is required for ensuring the RTL implementation matches the model performance. You will work on a state-of-the-art formal approach to reduce RTL and model correlation work and help finding performance bugs. You will learn about tools used to prove equivalence between C and RTL code and study how they can be applied to selected performance features.

Knowledge of C/C++, Verilog and interest in formal methods would be a plus.

[2021-V7] – Verify CPU Memory system with Litmus test in random stimuli

Barrier instructions are key instructions in ARM instruction set of ARM’s superscalar CPUs with out-of-order execution. However, it leads to tricky implementation and validation concerns.

The goal of this internship is to implement a strong validation strategy from existing ARM internal tools. You will first need to get familiar with ARM memory model & ordering rules around barriers a.k.a Litmus sequences. Then you will enhance the ARM verification environment by enabling Litmus sequences and the related checks. You will analyze the results of it. Knowledge of Python is a plus.

[2021-P3] – Adding Linux System Call emulation to a CPU Performance model

Modern application processors rely on advanced prefetching techniques to counter the "memory wall". The goal of this internship is to provide another option of running workloads on an existing CPU micro-architectural model. The model, used for performance projections and estimations, is currently capable of running either a bare metal binary or a Linux image. For some workloads, it can be challenging to build them as bare-metal binaries. While using a full Linux image might create a big and unnecessary overhead. Extending the model with functionality to emulate a subset of the Linux syscall ABI should allow to run binaries, compiled to run on Linux systems, without booting a full kernel. This internship requires strong C programming skills.