Introduction
This master project was focused on validating the behavior of atomic instructions in the memory subsystem of a CPU, in a multiple cores environment. It involved the understanding of part of the ARM architecture, the verification methodologies used, the memory subsystem testbench (named D-side testbench) and part of the RTL design of a recent ARM CPU.

Theory/Method/Hypothesis
It was important to study the micro-architecture of ARM high-end processors. They are pipelined and multi-scalar processors with out-of-order execution, using a RISC instruction set and a load-store architecture.

The atomic instructions were added in a recent update of ARM architecture to make easier the co-operation of different CPUs working on shared data. These atomic instructions go beyond the classical load-store architecture. Though they are strictly speaking a tighten bundle of loads and stores, they allow the program to directly modify a value in memory without necessary loading it in the register bank.

It was also useful to know the theory of RTL-design validation with stimulus-oriented testbench. Indeed, the purpose of the master project was to validate the new atomic instructions in the memory subsystem; in practise, this means that the testbench testing this part of the design had to be updated to generate new stimulus and checking them properly.

All the entities of a classic testbench had to be modified: stimulus generator, driver, monitor and checkers.

The main part of the project was to update the driver of the testbench, which is a model of the real core. It fetches instructions, issue them out-of-order in different issue queues, handle precise aborts, branch speculation, exception levels, etc.

We used an iterative method to validate both the behavior of the core model and the DUT. Once a first implementation of atomics was done, RTL simulations were realized and bugs found. It was then determined whether it was a testbench or RTL bug, and the issue was fixed; afterwards new simulations could be ran, until no bugs are found!

Results
The core model was incrementally improved to support the new atomic instructions. Such development, in parallel of the RTL implementation, allowed to further define the specifications and expectations of the real core and the memory subsystem regarding atomics.

Thanks to the update of the D-side testbench checkers, and with a lot of simulations, partial validation of the atomics implementation of the DUT could be made. There are remaining bugs that have been clearly identified and are being fixed. A decent pass rate of the simulations (more than 98%) has been reached, and new scenarios will be tested soon to further stress the design.

To sum up, the main result consisted in enabling the RTL design to be improved thanks to the testbench simulations results, making the atomics implementation in the DUT possible!

Conclusion/Perspectives
Partial validation of the new atomic instructions introduced in ARM architecture v8.2
Further development of the testbench for a full validation.

Acknowledgments
EPFL : Dr. Yusuf Leblebici
ARM : Florent Begon, Frederic Bœuf

References