4.	Fachgebiet Architektur eingebetteter Systeme **Advanced Computer Architectures**	SoSe2016

Assignment start: 11.07.2016 Submission deadline: 26.07.2016

Assignment 4 - Simultaneous Multithreading 25 Points

In the final exercise you will investigate the effect of executing two threads simultaneously on the same physical core capable of Simultaneous Multithreading (SMT). SMT allows better utilization of core resources by selecting independent instruction from multiple threads simultaneously. The effectiveness, however, varies depending on the workload the threads are performing. With SMT most core resources are shared (e.g. functional units, reservation stations, reorder buffers, branch predictor entries, etc), only the architectural register files are duplicated. Because the core resources are shared, when threads content for the same resources performance of both threads reduce.

In this assignment one thread is calculating the 40th fibonacci number. Your task is to produce code for the other thread that has *as little as possible influence* on the performance of the fibonacci thread. The code sequences can use all the tricks you can think of except they must be active and running. This means no `sleep`, `printf`, or other system calls that can lead to a blocked thread. Furthermore, no inline assembly is allowed, and the fibonacci thread must not be modified.

The SMT code is available at the ISIS page. The experiments should be performed on a machine with an Intel Xeon E5-2680v3 with 2×12 cores. This processor has the Haswell microarchitecture and the main specification can be found at `http://ark.intel.com/products/81908/Intel-Xeon-Processor-E5-2680-v3-30M-Cache-2_50-GHz`. The Turbo Boost feature has been turned off for this assignment and the processor is clocked at 2.5GHz.

To get an account on this machine you have to give the student assistant your tubit username during the lab. The student assistant will then create you an account on this machine on which you can login with your *tubit* password. The address of this machine is *cell1.aes.tu-berlin.de* and is accessible using SSH from within the *tu-berlin* net. Also the student assistant will assign you a core number to minimize the sharing of cores during the lab. This core number should be filled as the `CPUID_SMT0` in the source file.

The deliverable is the source file of the assignment. In the file the `tf_smt1` should be documented with a brief explanation of the choices you made. The grade for this exercise will be determined by the performance of the fibonacci thread (15 points) and your explanation (10 points).