Master Thesis:

**Nexus# Co-Processor; An Integrated Hardware-Software Approach to Task Graph Management**

It is expected to have hundreds of cores on a single chip in the next few years. Utilizing the huge computational power of such a chip is an active research field, that uncovered the need of a non-classical parallel programming models. An example is OmpSs [2] programming model that enables the user to add pragmas to his code. The OmpSs runtime uses these pragmas afterwards to create tasks, detect dependencies between them, and schedule them to run on one core. However, doing all these task management processes in software became a bottleneck, thus, hardware acceleration is required.

Nexus++ [1] is a hardware task management system for OmpSs, its main goal is to provide a fast, scalable dependency-resolution and task-scheduling mechanisms.

Nexus#, A distributed version of Nexus++, was developed and evaluated using trace-based simulations. It has the same interface as Nexus++, but since it has multiple instances of Nexus++’s task manager, did not fit on the Virtex5 FPGA used when integrating Nexus++ with VSs.

The goal of this Master work is to port Nexus# to a larger FPGA board, namely the Xilinx ZC706 board. This work is a hardware-software co-design. It includes exploring ZC706’s system design alternatives and defining the proper interface between Nexus#, the ARM core on board (if needed), and the PCIe bus. Based on that, a compatible device driver should be developed to enable communication with Nexus# from the host machine. Finally, the target RTS needs to be modified to make use of the driver, in order to enabling running real applications with Nexus#.

This work requires knowledge in C/C++, Linux device drivers, VHDL, as well as FPGA system design.

**Contact Persons**

Tamer Dallou (tamer.dallou@aes.tu-berlin.de)
Ahmed Elhossini (ahmed.elhossini@aes.tu-berlin.de)
Ben Juurlink (ben.juurlink@aes.tu-berlin.de)

**Literatur**

