



## Vector assembly instruction support for Graal

Bachelor / Master thesis for ...

Matr.-Nr.: ...

Email: ...

Graal, which is part of the Maxine project [1], is an effort to create a new just-in-time compiler for Java that is itself written in Java. It is based on a port of the HotSpot client compiler from C++ to Java.

In order to prepare the infrastructure for more advanced optimizations like vectorization, the assembler backend, which currently supports the basic x64 and SSE2 instruction sets, needs to be extended to support more advanced instruction sets like SSE3, SSSE3, SSE4 and AVX [2].

The first part of this project is to create an overview of the instructions that are available in these instruction sets. This will then be used to, together with the supervisor, determine which instructions are required and how the API for using them should look like.

The second part of this project consists of the actual implementation of the code that emits the instructions.

Goals for this project are:

- Familiarize yourself with, and write an overview of, the register files and the types of instructions available in the SSE and AVX instruction sets. The “Intel Intrinsics Guide”, available from [2], is a good starting point.
- Design an API that fits into the current assembler backend.
- Implement the instruction emitting.
- Create functions to detect if the SSE and AVX instruction sets are available.
- Create test code to verify the created assembly code. This test code will be added to the suite of frequently executed tests.

The work's progress should be discussed with the supervisor at least every 2 weeks. Please note the guidelines of the Institute for System Software when preparing the written thesis.

Supervisor: Dipl.-Ing. Lukas Stadler

[1] <http://labs.oracle.com/projects/maxine/>

[2] <http://software.intel.com/en-us/avx/>