

Entwicklung einer Optimierungs-Strategie für einen Auto-Tuner für Hardware-Koprozessoren

Vivado High-Level Synthese (VHLS) [1] ist ein Tool zur automatischen Erstellung von Hardware-Komponenten aus C/C++-Funktionen. Diese Komponenten können als Hardware-Koprozessoren auf einem FPGA zur Beschleunigung besonders rechenintensiver Teile einer Software verwendet werden.

Um effiziente HW-Komponenten zu erhalten, müssen in die Funktionen jedoch manuell PRAGMAS (d. h. Hinweise für den Compiler) zur Programm-Optimierung (z. B. zum *function inlining*, *loop unrolling* und *pipelining*) sowie zur Schnittstellen-Synthese eingefügt werden.

Da dies systematisch geschehen kann, wurde im Sommersemester 2018 ein **Auto-Tuner** entworfen und in JAVA implementiert, der PRAGMAS zur Schleifen-Optimierung automatisch an den richtigen Stellen eines C-Programms einfügt. Die Implementierung basiert auf dem Source-to-Source-Compiler-Framework CETUS [2], das die Funktionen einliest, analysiert und PRAGMAS einfügt. (Dieses Vorgehen ermöglicht auch, den Source-Code weiter von Hand zu optimieren.) Der Auto-Tuner ruft das VHLS-Tool auf und wertet den Synthese-Report aus, um die beste PRAGMA-Kombination zu finden. Diese wird dann zur Erzeugung der HW-Komponente verwendet.

Die vorhandene Implementierung verwendet zur Analyse der PRAGMA-Kombinationen einen simplen Backtracking-Algorithmus mit exponentieller Laufzeit. Dies ist angesichts der langen Laufzeit des VHLS-Tools ungünstig. Deshalb soll in diesem Projekt eine Optimierungs-Strategie entwickelt werden, die die Wechselwirkungen zwischen Schleifen und PRAGMAS berücksichtigt und damit den Suchraum stark verkleinert. Mögliche Ansätze finden sich beispielsweise im *OpenTuner*-Projekt [3]. Desweiteren sollen auch PRAGMAS für Funktionen, Schnittstellen und Speicher berücksichtigt werden und eine Kostenfunktion, die verschiedene Qualitäts-Metriken kombiniert, verwendet werden.

Referenzen:

1. *Vivado Design Suite User Guide – High-Level Synthesis*
https://www.xilinx.com/support/documentation/sw_manuals/xilinx2017_4/ug902-vivado-high-level-synthesis.pdf
2. C. Dave et al.: *CETUS: A Source-to-Source Compiler Infrastructure for Multicores*, IEEE Computer, Dec. 2009
3. J. Ansel et al.: *OpenTuner: An Extensible Framework for Program Autotuning*, International Conference on Parallel Architectures and Compilation Techniques, Edmonton, Canada, 2014.

Hinweis:

Diese Aufgabe kann auch in einer **Masterarbeit** bearbeitet werden. In diesem Fall soll eine umfangreichere Tuning-Strategie mit weiteren Optimierungen für mehrere HW-Komponenten entworfen werden.

Für Fragen wenden Sie sich bitte an m.weinhardt@hs-osnabrueck.de.