

# Abstract

---

Modern computing systems apply heterogeneity to increase efficiency and capability. Different chip architectures in CPUs and GPUs are used together to create a tailored platform for any application. An increasingly important aspect of heterogeneous systems is programmable hardware, such as FPGA-like programmable logic. Such programmable hardware solutions, however, suffer traditionally from inadequate development tools and frameworks, resulting in expert knowledge being required for their implementation.

In this thesis, we are approaching this issue from an aspect that, in our opinion, has not received the attention it deserves: memory bandwidth. We first analyze the memory architecture of different generations of FPGA-SoCs to evaluate whether such memory bandwidth bottlenecks exist in these devices. Based on those results, we propose a framework to identify at an early development stage, whether a design will run into a memory bandwidth bottleneck. The framework takes a software implementation, a partitioning scheme and configuration parameters, derives the memory access behavior of a functionally-equivalent hardware/software-codesign and performs those memory accesses. Thereby, designers can make reasonable decisions on whether their design choices have to be adapted to reach the required performance. We also address the issue of memory virtualization in FPGA-SoCs. As many such devices lack a memory management unit for the programmable logic, we present such a unit.

# Zusammenfassung

---

Heutige Computersysteme verwenden Heterogenität, um ihre Effizienz und Fähigkeiten zu erhöhen. Verschiedenartige Chiparchitekturen in CPUs und GPUs werden dabei zusammen eingesetzt, um eine optimale Plattform für konkrete Anwendungen zu ermöglichen. Eine zunehmend bedeutende Rolle spielen dabei FPGA-ähnliche programmierbare Logikkomponenten. Derartige Lösungen für programmierbare Hardware leiden allerdings traditionell unter unzureichenden Entwicklungswerkzeugen, was Expertenwissen für ihre Implementierung erforderlich macht.

In dieser Dissertation betrachten wir dieses Problem von einem Blickwinkel aus, der unserer Meinung nach bisher zu wenig Aufmerksamkeit erfahren hat: Speicherbandbreite. Wir analysieren zuerst die Speicherarchitekturen verschiedener Generationen derartiger Chips um herauszufinden, ob derartige Engpässe existieren. Aufbauend auf diesen Ergebnissen schlagen wir ein Framework vor, um festzustellen, ob Speicherbandbreitenengpässe ein Problem für ein Projekt darstellen könnten. Das Framework benötigt eine Softwareimplementierung als Referenz, eine geplante Aufteilung des Projektes in Hardware- und Softwarekomponenten, sowie Konfigurationsparameter. Anschließend leitet es aus diesen Informationen das Speicherzugriffsverhalten eines geplanten Codesigns her und führt diese Speicherzugriffe aus. Durch die ermittelte Bandbreite können Designer\*innen gewährleisten, dass ihr Design die erforderliche Leistungsfähigkeit erreicht. Wir beschäftigen uns zudem mit der Problem der Speichervirtualisierung in derartigen Chips. Häufig verfügen diese über keine ausreichende Speicherverwaltungseinheit, was zu unzureichenden Ergebnissen führt. Wir präsentieren eine spezielle Einheit, die diese Problem löst.