

Architectural Trade-offs in Dynamically Reconfigurable Processors

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY
ZURICH

for the degree of
Doctor of Technical Sciences

presented by

ROLF ENZLER

Dipl. El.-Ing. ETH
born 30 September 1971
citizen of Walchwil ZG

accepted on the recommendation of
Prof. Dr. Gerhard Tröster, examiner
PD Dr. Marco Platzner, co-examiner

Abstract

This dissertation deals with the design and evaluation of dynamically reconfigurable processor architectures targeting the embedded computing domain. The main objective is the investigation of the architectural trade-offs involved in terms of computational performance and chip area.

Reconfigurable architectures promise to be a valuable alternative to conventional computing devices such as processors and application-specific integrated circuits. The hardware of reconfigurable architectures is, in contrast to processors or application-specific integrated circuits, not static but adapted to the applications at hand. Through dynamic hardware customization, reconfigurable architectures potentially achieve a higher efficiency than processors while maintaining a higher level of flexibility than application-specific integrated circuits. This work focuses on hybrid, dynamically reconfigurable processors that couple a standard CPU core with a reconfigurable processing unit.

The development of such technology includes a multitude of design decisions and architectural trade-offs. In order to study these issues, we propose a system-level evaluation methodology that allows a designer to measure the computational performance of hybrid reconfigurable processors and to estimate their chip area. The methodology is based on a hybrid architecture model, a system-wide, cycle-accurate simulation environment, and a parameterized area estimation model. Our approach enables a designer to investigate the system-level impact of specific architectural design features.

In order to characterize the targeted embedded computing domain, we analyze a pool of applications that represents a typical embedded workload. We distinguish between the application groups multimedia, cryptography, and communications. The analysis shows that the selected embedded workload differs significantly from a general-purpose workload. Furthermore, the analysis reveals that the three particular application groups compared to each other also feature distinctive characteristics and hence stress different architectural features of a processor. The consequence is that a reconfigurable processor design must account for the peculiarities of the targeted application domain.

For the class of data-streaming applications, which is part of the multimedia group, we present a hybrid, dynamically reconfigurable processor architecture that couples a multi-context, coarse-grained reconfigurable array as a coprocessor to a CPU core. The CPU is responsible for the data transfer, context loading, and control of the reconfigurable array. The array stores several configurations on-chip and thus allows for fast adaptation of its functionality. The envisioned programming model makes use of hardware virtualization, which allows for the abstraction of limited reconfigurable hardware resources.

In a case study, we implemented large finite impulse response filters and quantitatively investigated various design features of the reconfigurable processor. The experiments show that hardware virtualization is a suitable programming model, and that multi-context devices can successfully be employed in this regard. The results further prove that hybrid multi-context architectures have the potential to yield significant speedups. In our experiments, we achieved speedups of up to an order of magnitude over the stand-alone CPU at the expense of moderate area overheads. Overall, the case study emphasizes the importance of our system-level evaluation approach.

Zusammenfassung

Die vorliegende Arbeit befasst sich mit dem Entwurf und der Evaluation dynamisch rekonfigurierbarer Prozessorarchitekturen im Bereich eingebetteter Systeme. Das Hauptziel der Arbeit besteht dabei in der Untersuchung der Design-Tradeoffs hinsichtlich Rechenleistung und Chipfläche.

Rekonfigurierbare Architekturen stellen eine vielversprechende Alternative zu konventionellen Rechenbausteinen dar, zu denen Prozessoren und applikationsspezifische integrierte Schaltungen gehören. Die Hardware rekonfigurierbarer Architekturen ist im Gegensatz zu Prozessoren oder applikationsspezifischen integrierten Schaltungen nicht statisch, sondern kann dynamisch der aktuellen Applikation angepasst werden. Durch ihre parallelen Hardwarestrukturen erzielen rekonfigurierbare Bausteine eine potenziell höhere Rechenleistung als Prozessoren. Gleichzeitig ermöglicht die dynamische Adaption eine grössere Flexibilität im Vergleich zu anwendungsspezifischen integrierten Schaltungen. Die vorliegende Arbeit konzentriert sich auf hybride, dynamisch rekonfigurierbare Prozessoren, welche eine Standard-CPU mit einer rekonfigurierbaren Recheneinheit koppeln.

Bei der Entwicklung solcher Architekturen sind eine Vielzahl von Designentscheidungen und Tradeoffs zu berücksichtigen. Um die damit verbundenen Fragestellungen zu untersuchen, präsentieren wir eine Evaluationsmethodik auf Systemebene, mit der die Rechenleistung rekonfigurierbarer Prozessoren gemessen sowie deren Chipfläche abgeschätzt werden kann. Die Methodik basiert auf einem hybriden Architekturmodell, einer systemweiten, zyklengenauen Simulationsumgebung und einem parametrisierten Modell für die Flächenabschätzung. Mit Hilfe unseres Ansatzes kann der Einfluss der spezifischen Merkmale einer Prozessorarchitektur auf Systemebene untersucht werden.

Um das Anwendungsgebiet der eingebetteten Systeme zu charakterisieren, analysieren wir eine Reihe von Benchmark-Applikationen, welche eine für eingebettete Systeme typische Arbeitslast darstellen. Wir unterscheiden dabei die Applikationsgruppen Multimedia, Kryptographie und Kommunikation. Die Analyse belegt, dass sich der ausgewählte Applikationsmix deutlich von Applikationen aus dem General-Purpose-Bereich unterscheidet. Darüber hinaus zeigt sich, dass auch die