

Zusammenfassung

Die Dissertation präsentiert einen grundlegenden Beitrag in dem neuen Gebiet des Echtzeit-Multitasking auf rekonfigurierbaren Hardwarebausteinen (RHDs).

Aufgrund der wachsenden Logikkapazitäten können auf RHDs mehrere Hardwareprozesse (Berechnungen, implementiert als digitale Schaltungen) parallel ausgeführt werden. Weiterhin ermöglicht die Rekonfiguration zur Laufzeit die Wiederverwendung von Ressourcen für nacheinander ausgeführte Prozesse. Um RHDs für die Echtzeitanwendungen eingebetteter Systeme zu verwenden, werden sowohl Scheduling-Verfahren als auch Betriebssysteme benötigt, welche ein vorhersehbares Zeitverhalten der Prozesse erzeugen.

Insbesondere wird in dieser Arbeit die Ablaufplanung periodischer Echtzeitprozesse zur Ausführung auf gemeinsam genutzten RHD-Ressourcen betrachtet, welche sich von der Planung in Einzel- und Multiprozessorsystemen unterscheidet. In einem ersten Modell werden die Prozesse modelliert durch ihren Bedarf an Hardwareressourcen (Fläche), ihre maximale Ausführungszeit und ihrer Periode, welche die Zeitspanne zwischen aufeinanderfolgender Anfragen und auch die Frist der Prozesse definiert. Mehrere Prozesse können gleichzeitig ausgeführt werden, solange ihr gemeinsamer Flächenbedarf die Ressourcen des betrachteten RHD nicht übersteigt. Weiterhin kann die Ausführung eines Prozesses zu beliebigen Zeiten unterbrochen und später fortgesetzt werden (preemptive multitasking). Es werden drei neue, auf *Earliest Deadline First (EDF)* basierende, Scheduling-Verfahren entwickelt und deren Leistungsfähigkeit analysiert.

1. Bei dem *global EDF* Verfahren werden alle Ressourcen global verwaltet und unter Berücksichtigung aller aktiven Prozesse zugewiesen. Für dieses Verfahren wird ein in linearer Zeit berechenbarer *Scheduling-Test* entwickelt, der für eine gegebene Prozessmenge entscheidet, ob alle Fristen eingehalten werden.
2. Das *partitioned EDF* Verfahren partitioniert die gegebene Prozessmenge zur Entwurfzeit. Die Prozesse jeder Teilmenge werden zur Laufzeit durch einen separaten EDF-Scheduler auf exklusiven RHD Ressourcen ausgeführt. Die Einhaltung der bekannten *Einzelprozessor-EDF* Auslastungsschranke je Teilmenge garantiert die Fristen aller Prozesse. Für kleine Prozessmengen wird die optimale Partitionierung mittels ganzzahliger linearer Programmierung errechnet, während für größere Prozessmengen eine *next fit* Heuristik nahezu optimale Lösungen erreicht.
3. *MSDL* ist ein serverbasiertes Scheduling-Verfahren welches Prozesse zur parallelen Ausführung zu periodischen Servern zusammenfasst. Zur Laufzeit werden die Server sequentiell auf dem RHD durch einen EDF-Algorithmus ausgeführt. Dabei läuft immer nur ein Server gleichzeitig, wodurch das Verfahren besonders für RHDs geeignet ist, welche partielle Rekonfiguration nicht unterstützen.

In einem analytischen Leistungsvergleich wird gezeigt, dass sich die drei Verfahren gegenseitig nicht dominieren. Dennoch zeigen Simulationsexperimente, dass im Durchschnitt *global EDF* die beste Leistung erzielt, gefolgt von *partitioned EDF* und *MSDL*. Weiterhin wurde gezeigt, dass der entwickelte *Scheduling-Test* für das *global EDF* Verfahren relativ pessimistisch ist. Die Rekonfigurationszeiten, in der das RHD keine Prozesse ausführen kann, wurden für alle drei Verfahren analysiert und in die Scheduling-Bedingungen integriert. Simulationsexperimente zeigen, dass durch diese Verluste die Leistung von *global EDF* stärker als die der anderen Verfahren gemindert wird.

Weiterhin werden zwei Erweiterungen des Prozessmodells entwickelt, um die spezifischen Charakteristika von rekonfigurierbaren Rechensystemen und deren Anwendungen zu berücksichtigen: Zuerst wird betrachtet, dass für jeden Prozess mehreren alternativen Hardwareimplementierungen existieren können. Für das *partitioned EDF* Verfahren wird ein ganzzahliges lineares Programm entwickelt, welches die optimale Hardwareimplementierung für jeden Prozess auswählt und damit die Leistung des Scheduling-Verfahrens signifikant steigert. In der zweiten Modellerweiterung wird der Speicherzugriff der Prozesse unter Echtzeitbedingungen berücksichtigt. Die entwickelte Methode erlaubt die effiziente gemeinsame Benutzung externer Speicherbänke, wodurch die benötigten Speicherressourcen verringert werden, ohne dabei die Prozessfristen durch blockierende Zugriffe zu gefährden.

Abschließend wird die Implementierung eines FPGA-Betriebssystemkerns vorgestellt, bei welchem im Gegensatz zu anderen Systemen alle Betriebssystemfunktionen vollständig in Hardware realisiert sind. Der Prototyp realisiert das *MSDL* Scheduling-Verfahren und verwendet vollständige Rekonfiguration, wodurch er für den Großteil heutiger RHDs geeignet ist.

Zusammenfassend stellt die Dissertation eine Grundlage für Echtzeit-Multitasking auf RHDs dar, einschließlich Modellbildung, Scheduling-Algorithmen, Analyse und prototypischer Implementierung.