

Zeit-Analyse auf Mikrocontroller-internen Verbindungsnetzwerken mithilfe von Mikrobenchmarks

Georg Seifert Peter Hartmüller

17. November 2017

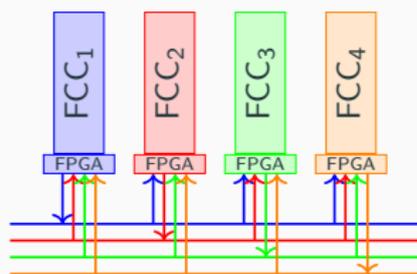
Technische Hochschule
Ingolstadt 

- Motivation
- Problemstellung
- Methodik
 - Analyse des EA-Systems
 - Ergebnisse der Analyse
 - Weitere Arbeiten
- Zusammenfassung

Motivation

Sicherheitskritische Echtzeitsysteme in der Avionik

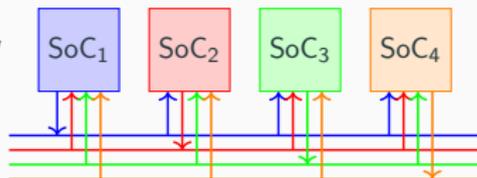
Stand der Praxis



Klassischer vierkanaliger
Flugsteuerungsrechner



Zukünftig



SoC mit externem Redundanznetzwerk



- Prozess-EA
- SoC-interner Kommunikation

Problemstellung

- Hohe Datenraten mit deterministischen Transferzeiten
- Vielzahl an Schnittstellen
- Unbekannte SoC-interne Kommunikationsprotokolle
- ▶ Eingeschränkte Hardwarenutzung

**Erweiterung der statischen WCET-Analysen
um asynchrone DMA**

- WCET-Tools vernachlässigen meist den Einfluss von asynchroner Ressourcennutzung
- Hohe Datenraten verursachen **signifikanten** Einfluss auf die WCET
- ▶ Einflüsse müssen zukünftig analysiert und berücksichtigt werden

Methodik

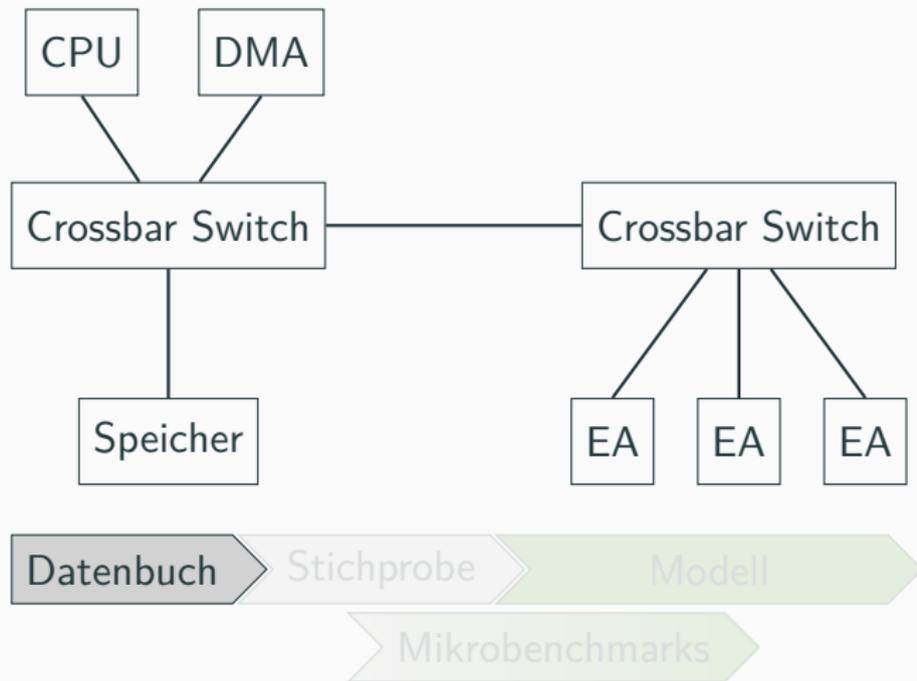
Allgemeines Vorgehen

- Analyse des EA-Systems
- Abstraktes Modellieren der interferenzbehafteten Pfade
- Simulation, Referenzmessung und iteratives Verbessern des Modells

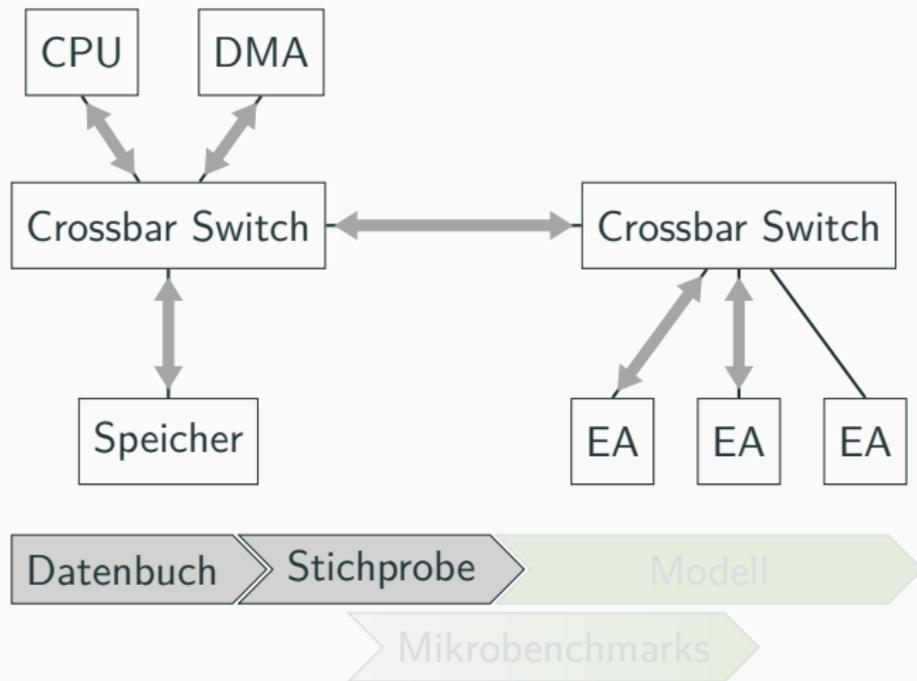
Methodik

Analyse des EA-Systems

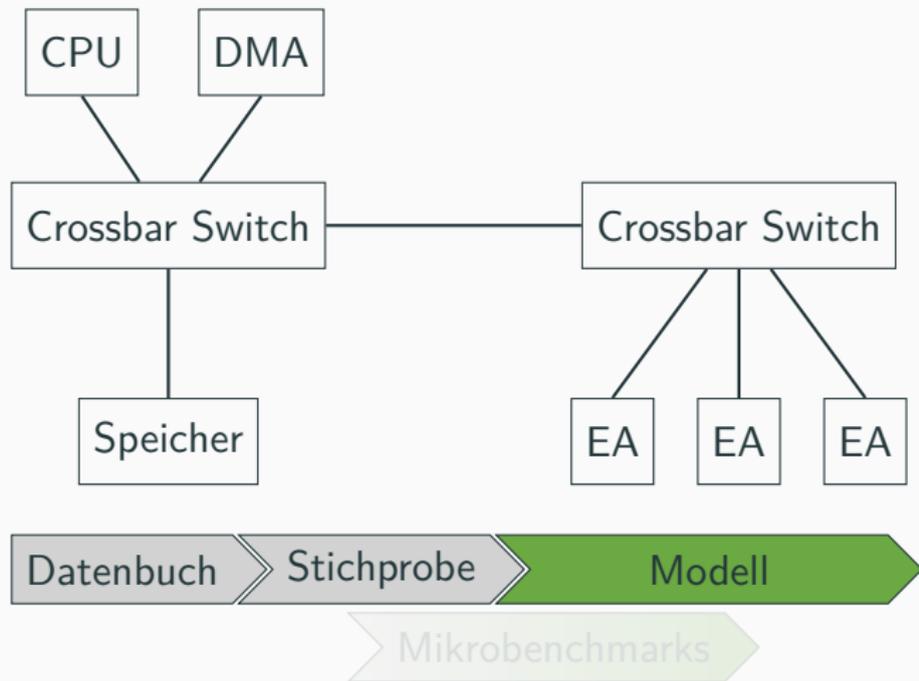
Möglichkeiten der Analyse



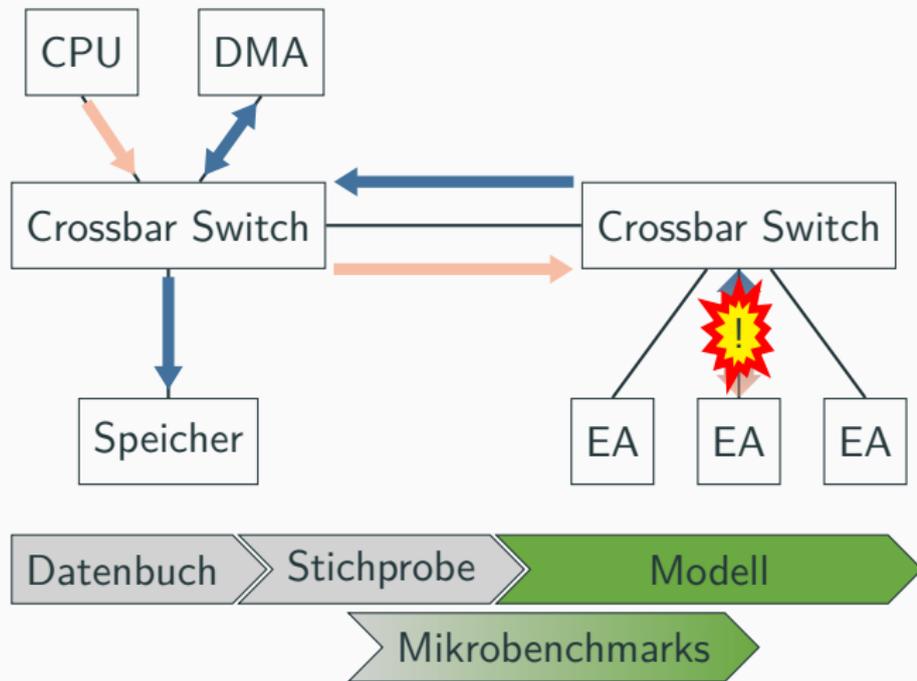
Möglichkeiten der Analyse



Möglichkeiten der Analyse



Möglichkeiten der Analyse



Definition Mikrobenchmarks

Ein Mikrobenchmark ist ein minimalistisches Programm, welches einzelne Pfade isoliert nutzt.

`gpioJumpLabel:`

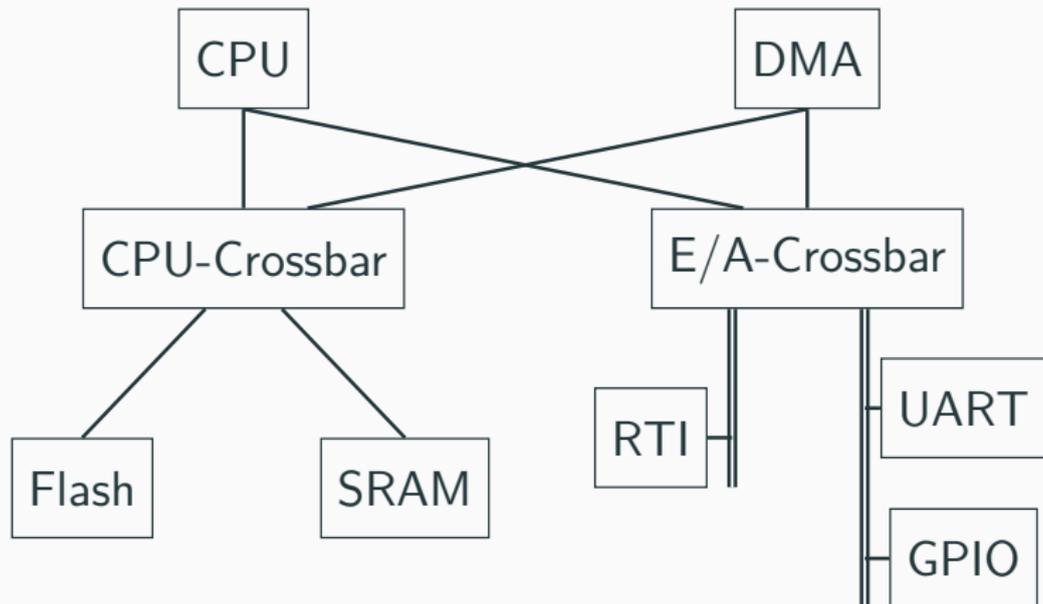
```
    str r1, [Data Clear Register]
    str r1, [Data Set Register]
    subs r2, r2, #1
    bne gpioJumpLabel
```

Messmethoden von Mikrobenchmarks

- Messung der Laufzeit **aller** Iterationen
- Messung der Laufzeit **einzelner** Iterationen
 - mit externen Messinstrument
 - mit dedizierter Tracing-Hardware
 - mit internen Zeitgebern

Methodik

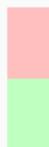
Ergebnisse der Analyse



Auszug Blockschaltbild Hercules TMS570LC4375

Verifikation konkurrierender Zugriffe

CPU DMAC	RTI	ECAP	GIO	N2HET	UART
RTI	Kollisionsfreie Pfade	Kollisionsfreie Pfade	Kollisionsfreie Pfade	Kollisionsfreie Pfade	Kollisionsfreie Pfade
ECAP	Kollisionsfreie Pfade	Kollisionsbehaftete Pfade	Kollisionsfreie Pfade	Kollisionsfreie Pfade	Kollisionsfreie Pfade
ePWM	Kollisionsfreie Pfade	Kollisionsbehaftete Pfade	Kollisionsfreie Pfade	Kollisionsfreie Pfade	Kollisionsfreie Pfade
UART	Kollisionsfreie Pfade	Kollisionsfreie Pfade	Kollisionsbehaftete Pfade	Kollisionsbehaftete Pfade	Kollisionsbehaftete Pfade
SPI	Kollisionsfreie Pfade	Kollisionsfreie Pfade	Kollisionsbehaftete Pfade	Kollisionsbehaftete Pfade	Kollisionsbehaftete Pfade



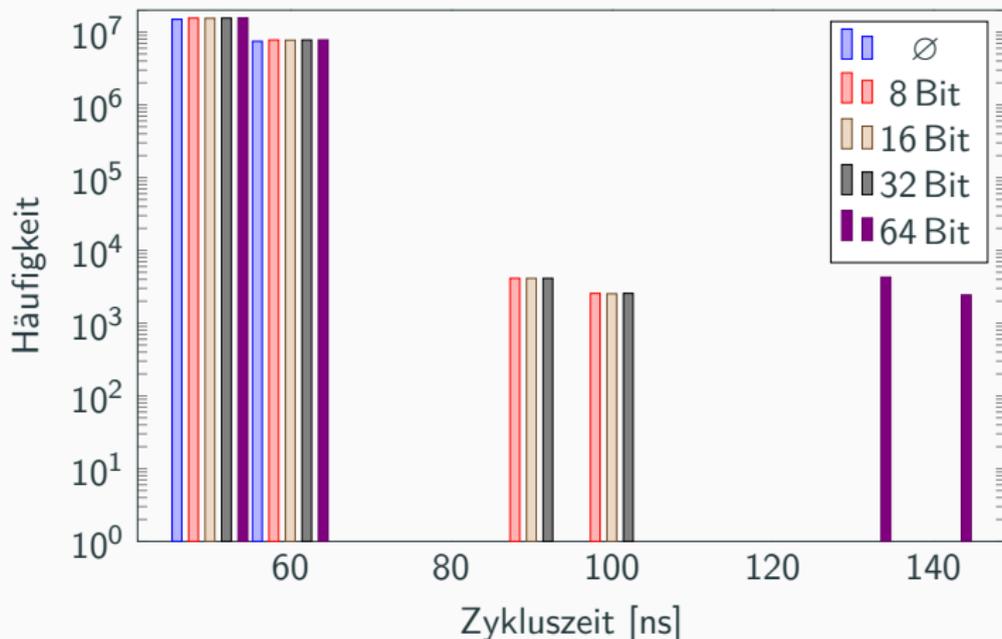
Kollisionsbehaftete Pfade

Kollisionsfreie Pfade

Kollisionen konkurrierender Zugriffe (Auszug)

Konkurrierende Transfers:

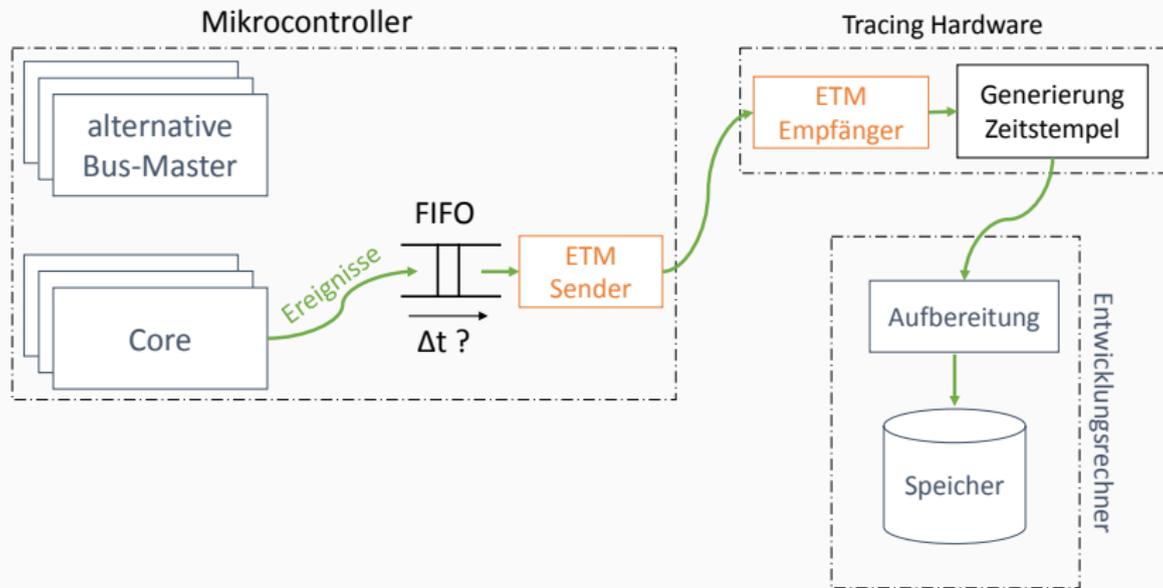
Toggeln GPIO durch CPU, gestört durch UART-DMA-Transfers



Einzel-Iterationen gemessen mit Logic Analysator

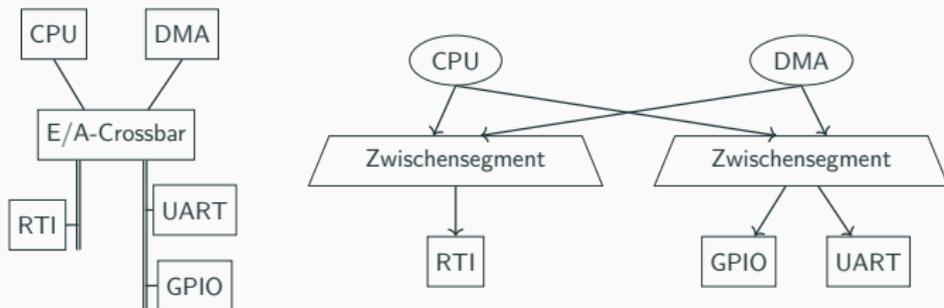
- Ergebnisse basieren auf Messungen
- Ergebnisse sind nur für eine spezielle Konfiguration gültig
- Messungenaugigkeit durch verbaute Hardware

Genauigkeit der Zeitstempel der ETM



Methodik

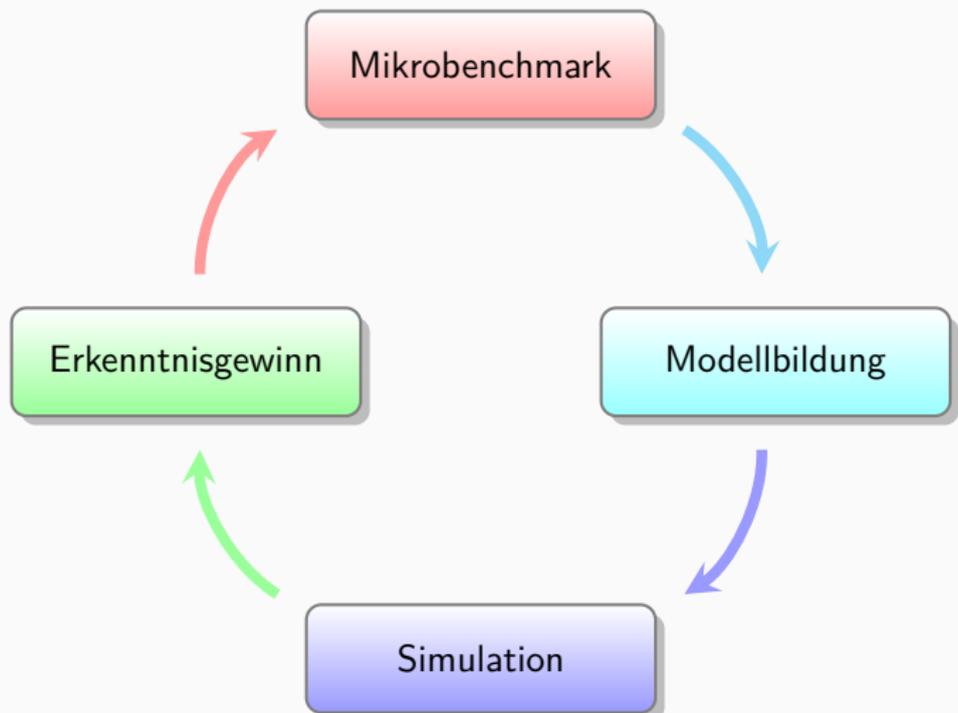
Weitere Arbeiten



- Abstraktion der Netzwerkhierarchien
- Abstraktion der CPU und DMA-Controller
- Abstraktion der Peripheriegeräte

- Validieren der Ergebnisse
- Iteratives Verbessern des EA-Modells
- Ableiten von Kollisionswahrscheinlichkeiten

Zusammenfassung



Fragen?

Kommentare!

