

# Thread basierte partielle Rekonfiguration von SoC Systemen

Frank Opitz

INF-M1 – Anwendung 1 - Wintersemester 2009/2010

24. November 2009

# Inhalt

---

- ▶ Motivation
- ▶ Zielsetzung
- ▶ dynamische Re-/Konfiguration von SoC FPGAs
- ▶ Auswahl eines RTOS
- ▶ Aufbau des Systems zur dynamischen Rekonfiguration
- ▶ Zusammenfassung
- ▶ Ausblick

---

# Motivation

# Motivation

---

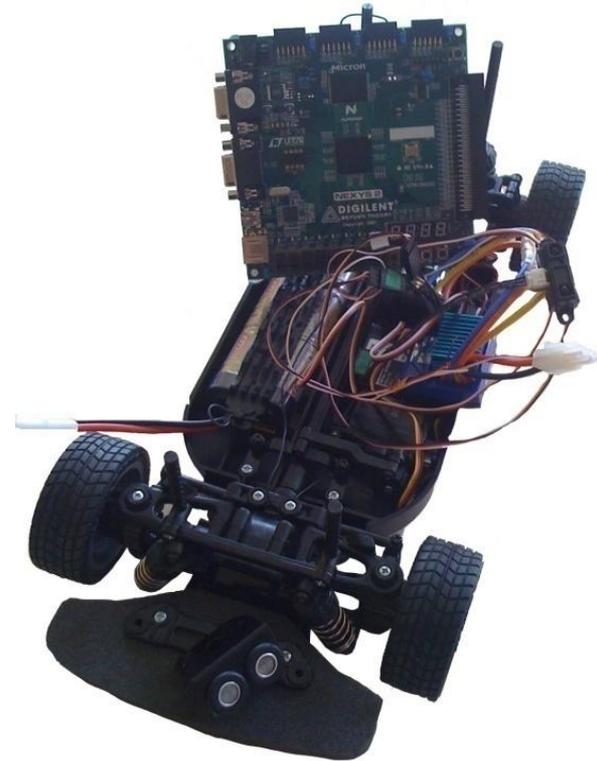
- ▶ Mehrfachnutzung der FPGA Komponenten
- ▶ Anpassen der IP-Funktion an Umgebungsbedingungen
- ▶ Beschleunigung unterschiedlicher komplexer Berechnungen in Threads

# CaroloCup Fahrzeuge

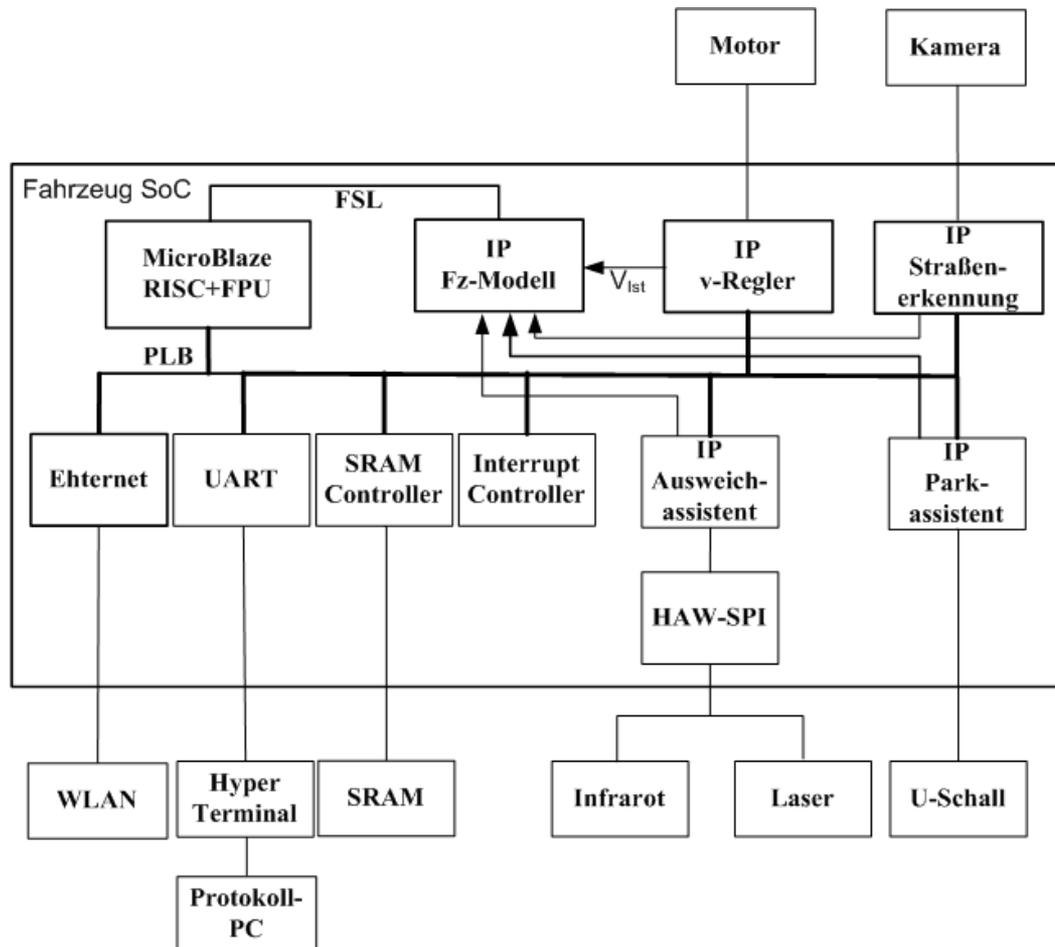
Software  
(Onyx)



System on Chip



# Aufbau des Zielsystems



---

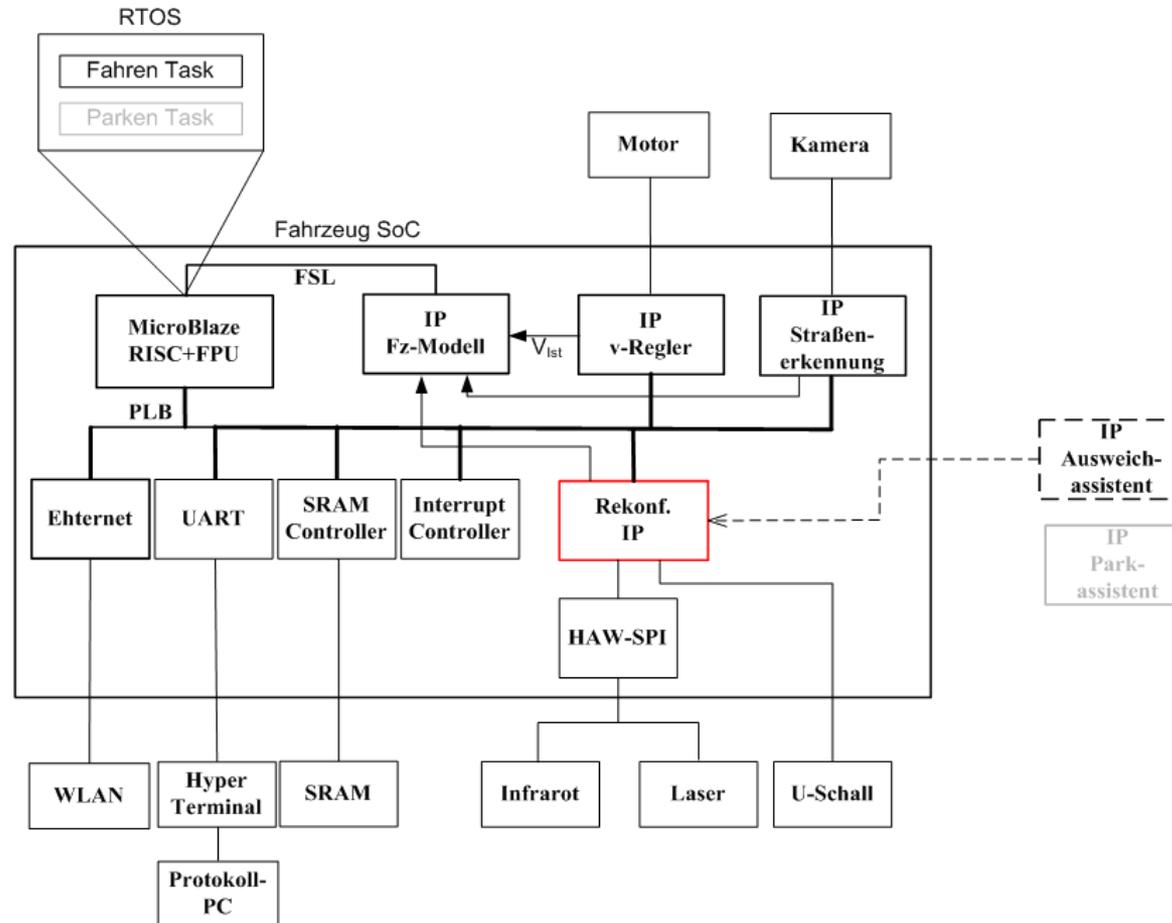
# Zielsetzung

# Zielsetzung

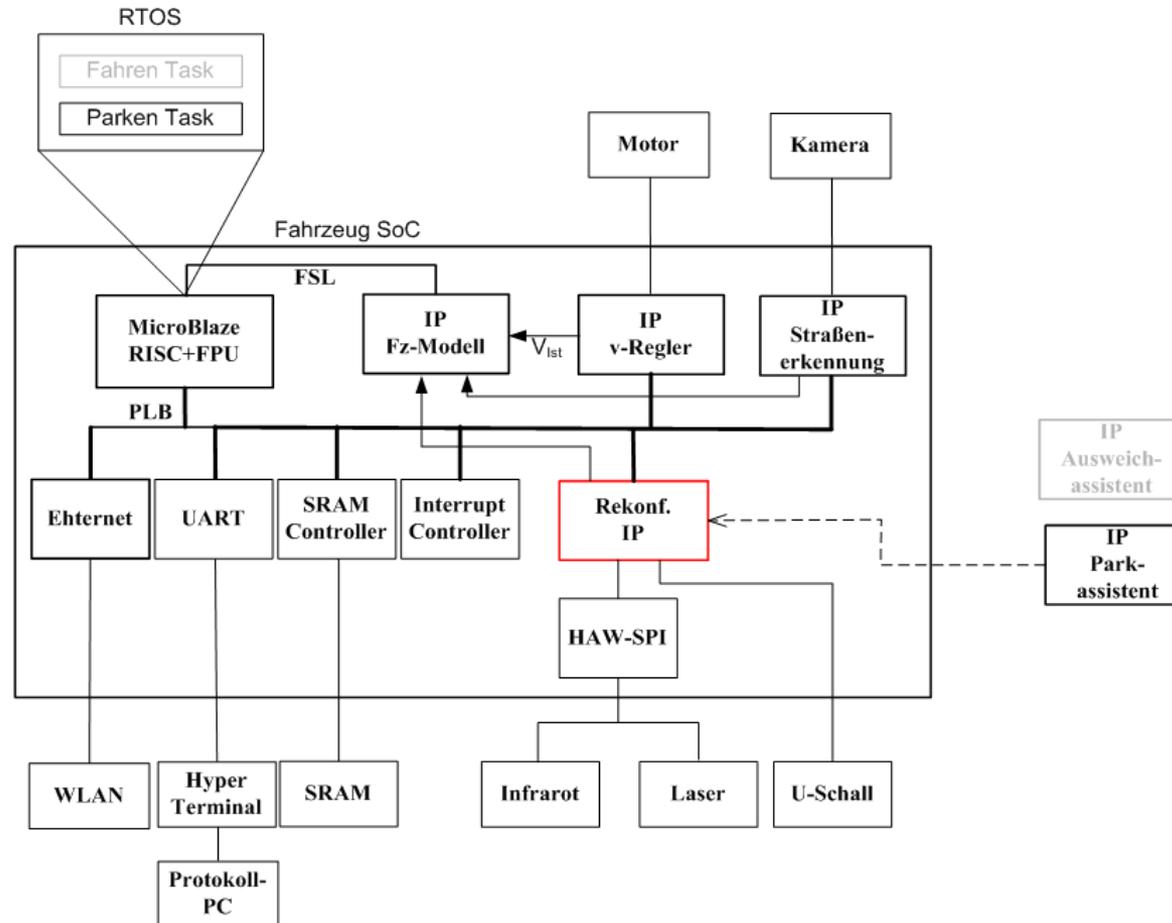
---

- ▶ Implementierung eines Systems zur dynamischen Selbstrekonfiguration
- ▶ Evaluierung eines RTOS für den MicroBlaze Softcore Prozessor
- ▶ Einsatz des PR Systems auf dem SoC Carolo Cup Fahrzeug

# Fahrzeug mit Rekonfiguration



# Fahrzeug mit Rekonfiguration

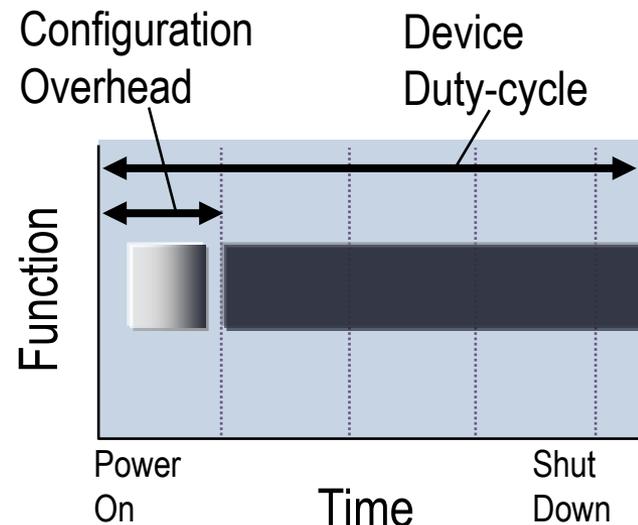


---

# Dynamische Re-/Konfiguration von SoC FPGAs

# Typische Konfiguration

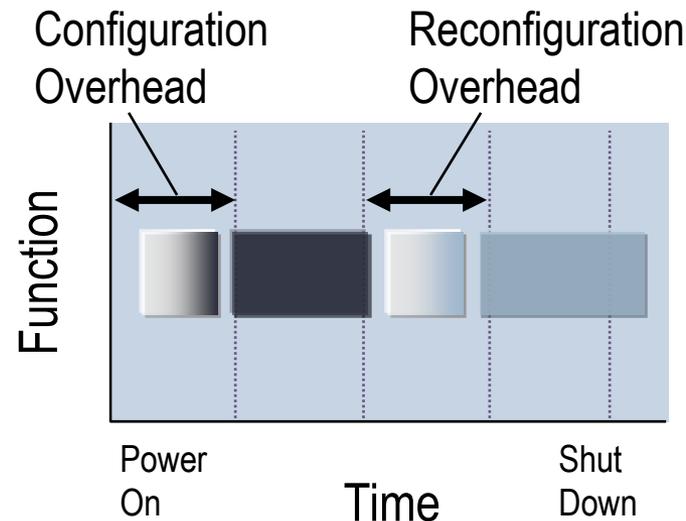
- ▶ Laden der Konfiguration bei Power-On
- ▶ Eine Konfiguration bis zum Reset des Systems



[2]

# Rekonfiguration

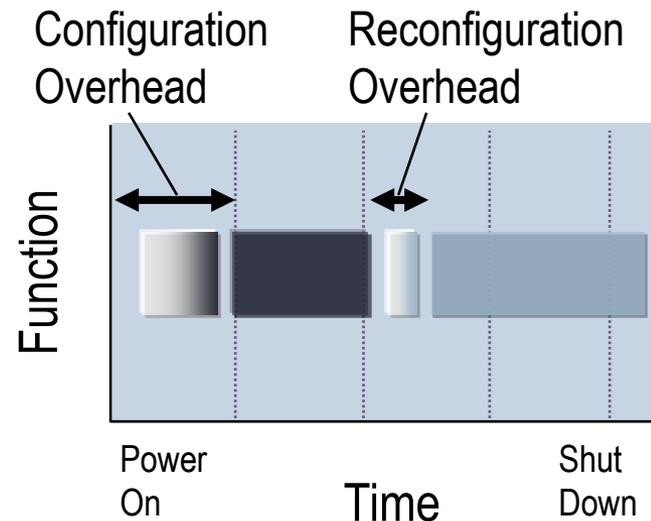
- ▶ Laden einer Konfiguration bei Power-On
- ▶ Laden von 1-n Konfigurationen im Betrieb
- ▶ Reset des Systems bei der Rekonfiguration



[2]

# Partielle Rekonfiguration

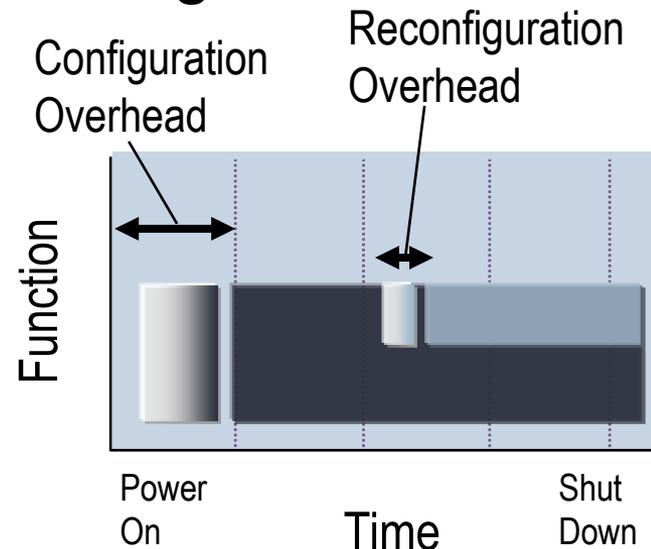
- ▶ Gesamt-Konfiguration bei Power-On
- ▶ 1-n Konfigurationen während des Betriebes
- ▶ Übertragung eines Teil-Byte-Streams
- ▶ Berechnungen werden unterbrochen



[2]

# Dynamische partielle Rekonfiguration

- ▶ Gesamt-Konfiguration bei Power-On
- ▶ 1-n Konfigurationen während des Betriebes
- ▶ Übertragung eines Teil-Byte-Streams
- ▶ HW/SW Berechnungen werden weiter durchgeführt



[2]

# Varianten der partiellen Rekonfiguration

---

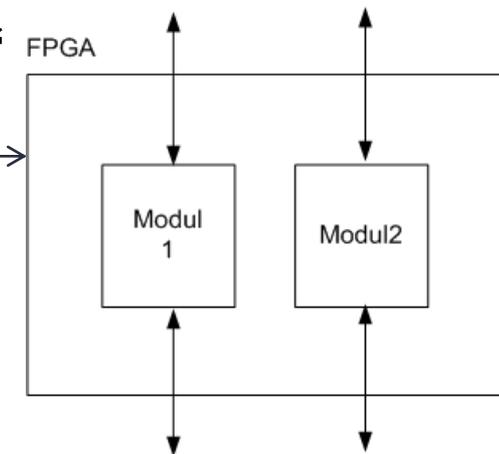
- ▶ „Small-Bit Manipulations“
  - ▶ Übertragung von Änderungen an LUTs
  
- ▶ „Module-Based Partial Reconfiguration“
  - ▶ Unabhängige Module
  - ▶ Abhängige Module über Bus Makros

[1]

---

# Varianten der partiellen Rekonfiguration

- ▶ „Small-Bit Manipulations“
  - ▶ Übertragung von Änderungen an LUTs
- ▶ „Module-Based Partial Reconfiguration“
  - ▶ Unabhängige Module
  - ▶ Abhängige Module über Bus Makros

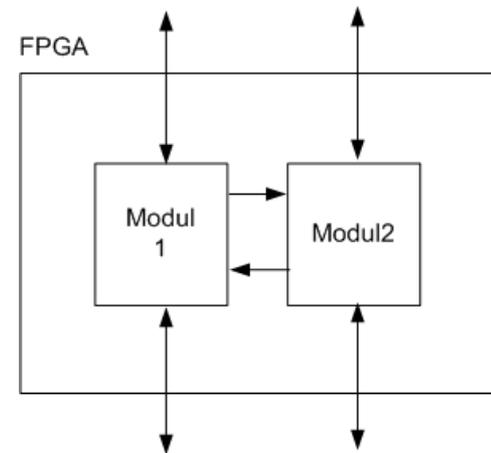


[1]

# Varianten der partiellen Rekonfiguration

- ▶ „Small-Bit Manipulations“
  - ▶ Übertragung von Änderungen an LUTs

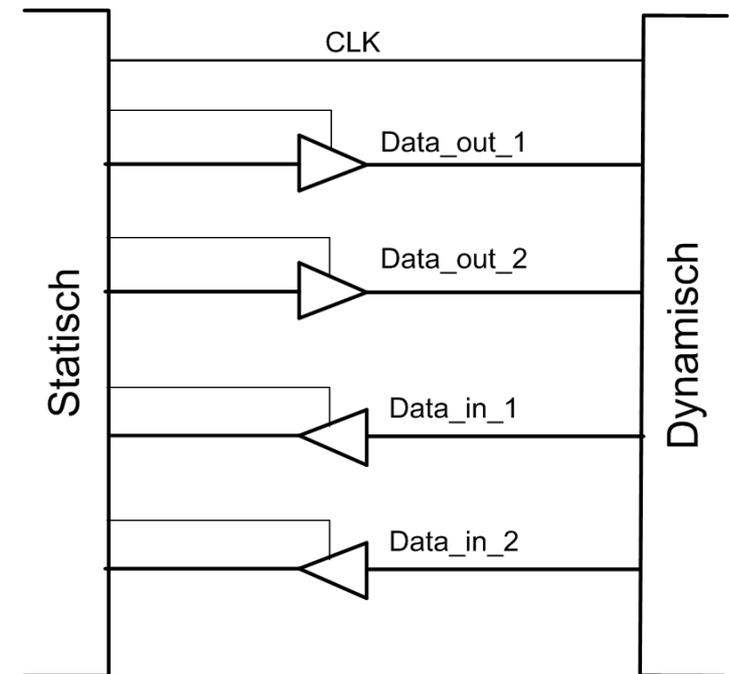
- ▶ „Module-Based Partial Reconfiguration“
  - ▶ Unabhängige Module
  - ▶ Abhängige Module über Bus Makros



[1]

# Bus Markos

- ▶ Bestimmen die Route der Signale
- ▶ Alle I/O Signale (außer CLK) der PR Entity sind durch Bus Makros fest zu setzen



[1]

---

# Auswahl eines „Real-Time Operating Systems“ (RTOS)

# Anforderungen an das RTOS

---

- ▶ Aufgaben eines OS :
  - ▶ Taskverwaltung
  - ▶ Betriebsmittelverwaltung
  - ▶ Interprozesskommunikation
  
- ▶ Geringer Speicherbedarf
  
- ▶ Vorhersagbares Verhalten

[4]

---

# Auswahl des RTOS

- ▶ Kriterien von der Canadian Space Agency aufgestellt

Kategorie Name	Kat.	Kriterium	Krit.	Ges. <sup>1</sup>
Kernel	13 %	Architektur	35 %	5 %
		Multi-Prozessor	25 %	3 %
Interrupt und Exception Handling	8 %	Preemtable ISR	30 %	2,4 %
		Modifizierbarkeit der Interupt Vektor Tabelle	20 %	1,6 %

1) Ges. = Kat. \* Krit.

[3]

## μC/OS-II

---

- ▶ Platz 9 (von 48) in der Studie der Canadian Space Agency
- ▶ Speicherbedarf < 24 Kbytes
- ▶ OS kann komplett im ROM gespeichert werden
- ▶ Portierung für den MicroBlaze
- ▶ TCP/IP Implementierung

[5] [6]

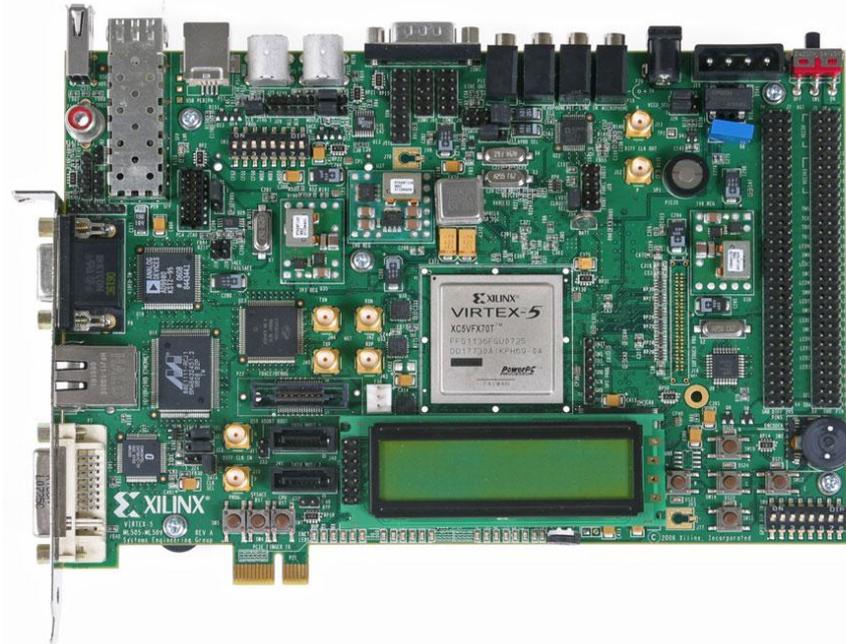
---

---

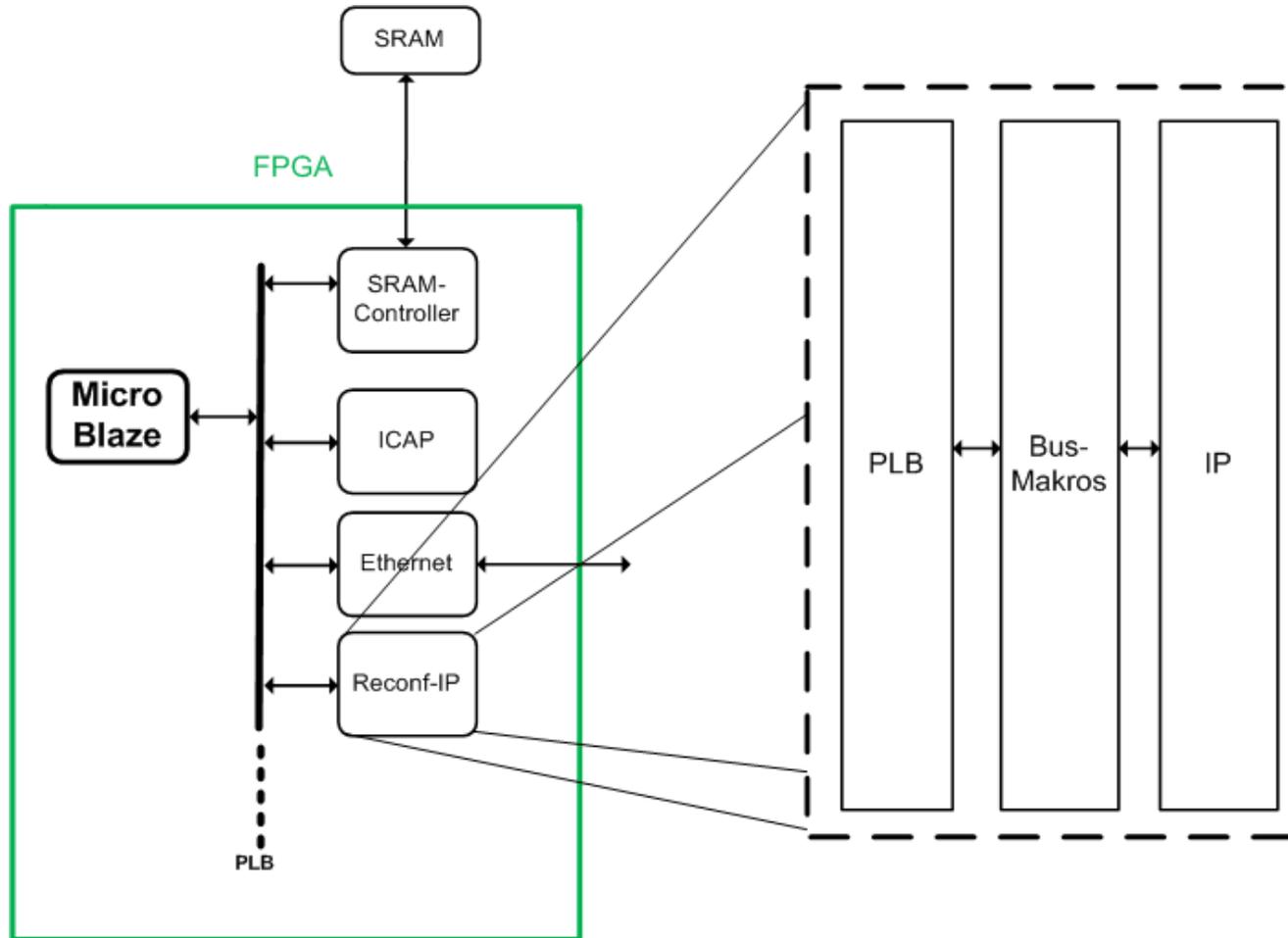
# Aufbau des Systems zur dynamischen Rekonfiguration

## ML 507

- ▶ Virtex 5
- ▶ 256 MB DDR2
- ▶ DVI/VGA
- ▶ PCI-Express
- ▶ Ethernet 10/100/1000 MB/s



# Testsystem



---

# Zusammenfassung

# Zusammenfassung

---

- ▶ Partielle dynamische Rekonfiguration:
  - ▶ Berechnungen laufen weiter
  - ▶ Anpassen der Hardware an die Umgebung
  
- ▶ Kriterien die für  $\mu\text{C}/\text{OS-II}$  sprechen:
  - ▶ Ressourcenbedarf
  - ▶ Existierende Implementierung
  
- ▶ Testsystem auf dem ML507 Board

---

# Ausblick

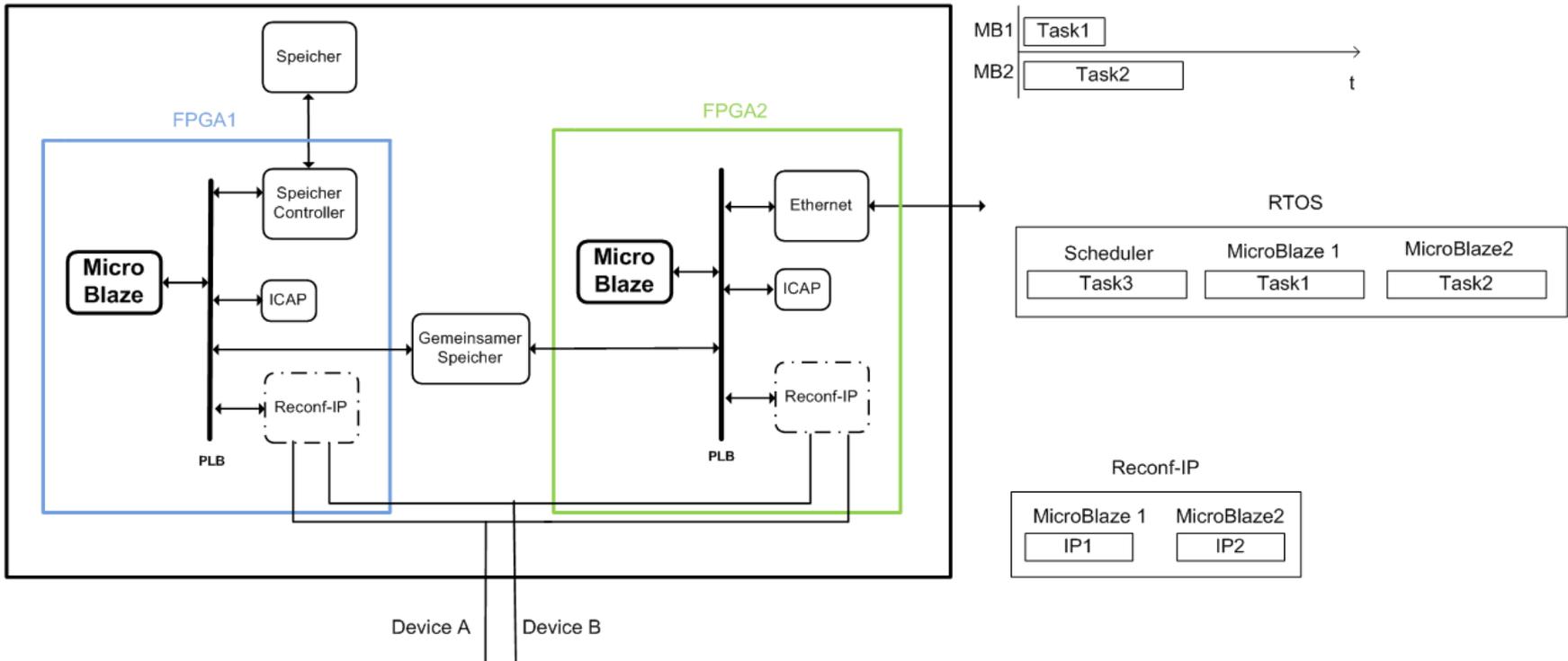
# Ausblick

---

- ▶ Verteilung der Rekonfiguration auf Multi-Core SoC Systemen
  - ▶ Verteilung der IPs
  - ▶ Verteilung der Threads je nach aktueller IP Belegung
- ▶ Aufbau eines Multi-Core SoC Systems mit selbst regulierender Core-Anzahl
  - ▶ Instanziierung der Cores je nach Auslastung

# Beispiel der IP Verteilung(1)

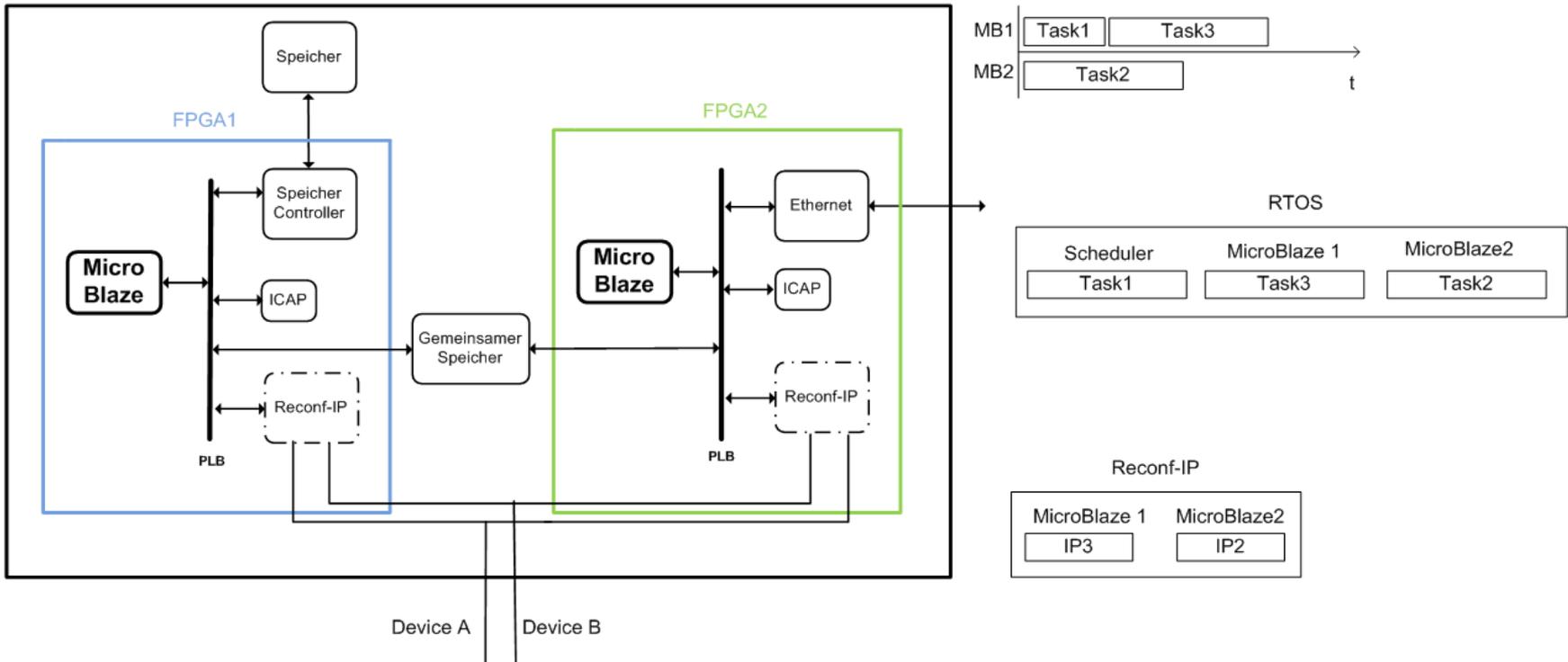
Multi FPGA System



Anmerkung: IP 1, IP3, IP4 auf FPGA1 und IP2 auf FPGA2

# Beispiel der IP Verteilung (2)

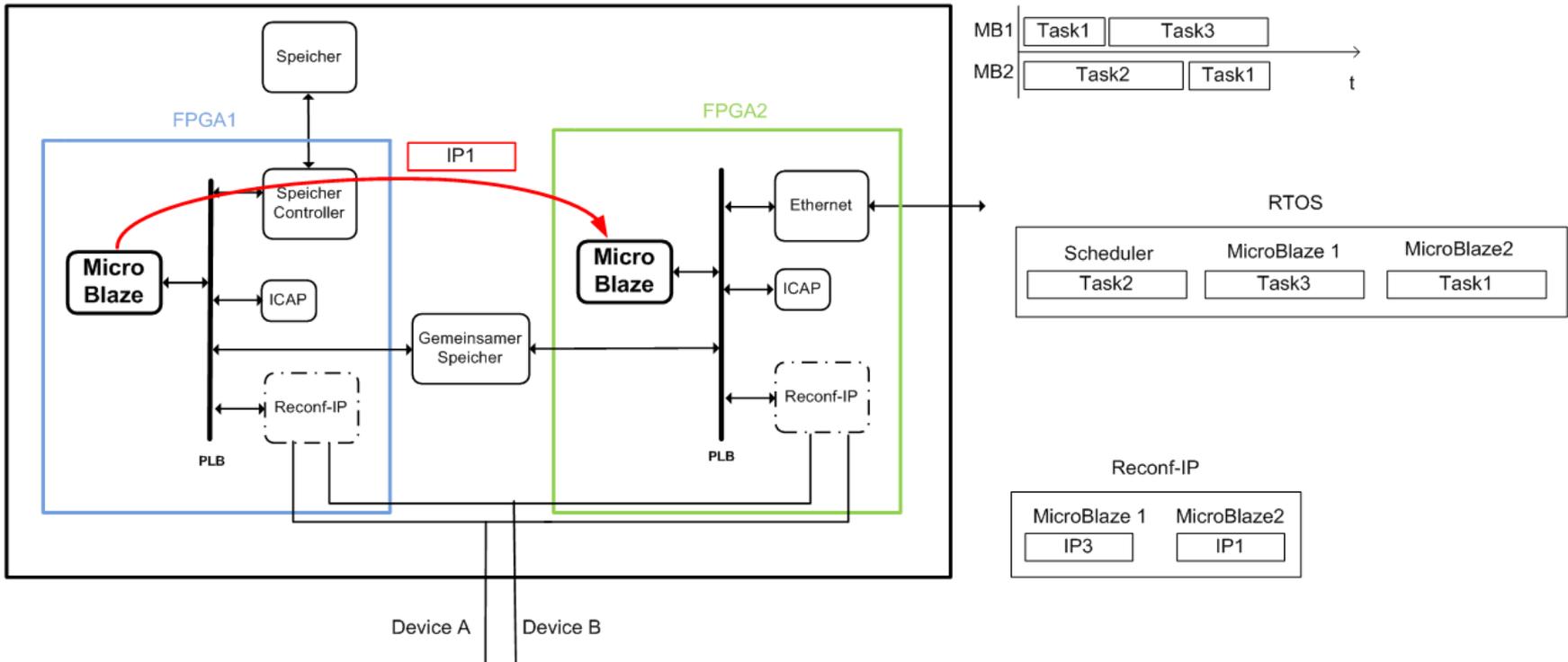
Multi FPGA System



Anmerkung: IP 1, IP3, IP4 auf FPGA1 und IP2 auf FPGA2

# Beispiel der IP Verteilung (3)

Multi FPGA System



Anmerkung: IP 1,IP3,IP4 auf FPGA1 und IP2 auf FPGA2

# Ausblick

---

- ▶ Verteilung der Rekonfiguration auf Multi-Core SoC Systemen
  - ▶ Verteilung der IPs
  - ▶ Verteilung der Threads je nach aktueller IP Belegung
- ▶ Aufbau eines Multi-Core SoC Systems mit selbst regulierender Core-Anzahl
  - ▶ Instanziierung der Cores je nach Auslastung

---

# Fragen?

# Literatur

---

- [1] **Xilinx. 2002.** Two Flows for Partial Reconfiguration: Module Based or Small Bit Manipulations. *Xilinx:Documentation*. [Online] Mai 17, 2002. [Cited: Oktober 16, 2009.] <http://www.xilinx.com/support/documentation/index.htm>.
- [2] **Xilinx. 2008.** 9.2.04 EA Partial Reconfiguration Software Training. Longmont, Colorado : 2008.
- [3] **Melanson, P. and Tafazoli, Siamak. 2006.** *A Selection Methodology for the RTOS Market*. Quebec : Canadian Space Agency, 2006.
- [4] **Wörn, H. and Brinkschulte, U. 2005.** *Echtzeitsysteme*. Heidelberg: Springer, 2005. 3-540-20588-8.
- [5] **Labrosse, Jason J. 2002.** *MicroC/OS-II The Real Time Kernel*. San Francisco : CMP Books, 2002. 1-57820-103-9.
- [6] **Micrium. 2009.**  $\mu$ C/OS-II Kernel.  *$\mu$ C/OS-II Kernel*. [Online] 5. Mai 2009. [Zitat vom: 2. November 2009.] <http://micrium.com/page/products/rtos/os-ii>.