

Entwicklung einer FPGA-basierten asymmetrischen MPSoC Architektur

INF-M1 Seminar Vortrag
Heiko Wilken
25. November 2010

Betreuer: Prof. Dr.-Ing. Bernd Schwarz

Übersicht

1. Motivation
2. Zielsetzung & Vorarbeiten
3. Arbeitsschwerpunkte
4. Risiken
5. Zusammenfassung & Ausblick

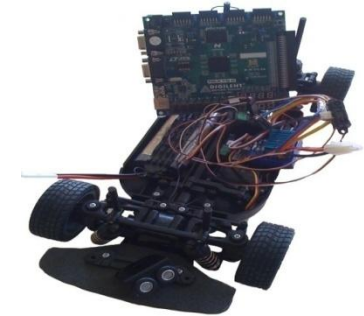
Motivation – Warum MPSoC ?

- „High Performance Embedded Computing“
 - Erhöhung des Datendurchsatzes bei niedrigeren Taktfrequenzen
- hoher Energiebedarf und steigende Wärmeabgabe bei „Singlecores“
 - Leistungssteigerung bei Reduzierung des Energiebedarfs (Spezialisierung)
- Echtzeitsysteme mit zeitkritischen Anforderungen
 - Video- und Audioanwendungen oder Fahrerassistenzsysteme

Einsatzgebiete

An der HAW Hamburg:

- FAUST Projekt „High Performance Embedded Computing“
- FAUST Projekt „Sensor Controlled Vehicle (SCV)“
- HW/SW CoDesign Projekt „Distributed Computing System“



Weltweit:

- Automobiltechnik / Fahrerassistenzsysteme / Car-Entertainment
- Smartphones, PDA, Playstation 3
- Echtzeitsysteme zur Bild- und Audioverarbeitung



Charakteristiken einer Multiprozessor-Architektur

Asymmetrische Multiprozessoren (AMP):

- Heterogene Prozessoren
- Jeder Prozessor hat eine fest zugewiesene Aufgabe
- Unabhängige und verschiedene RTOS Implementierungen

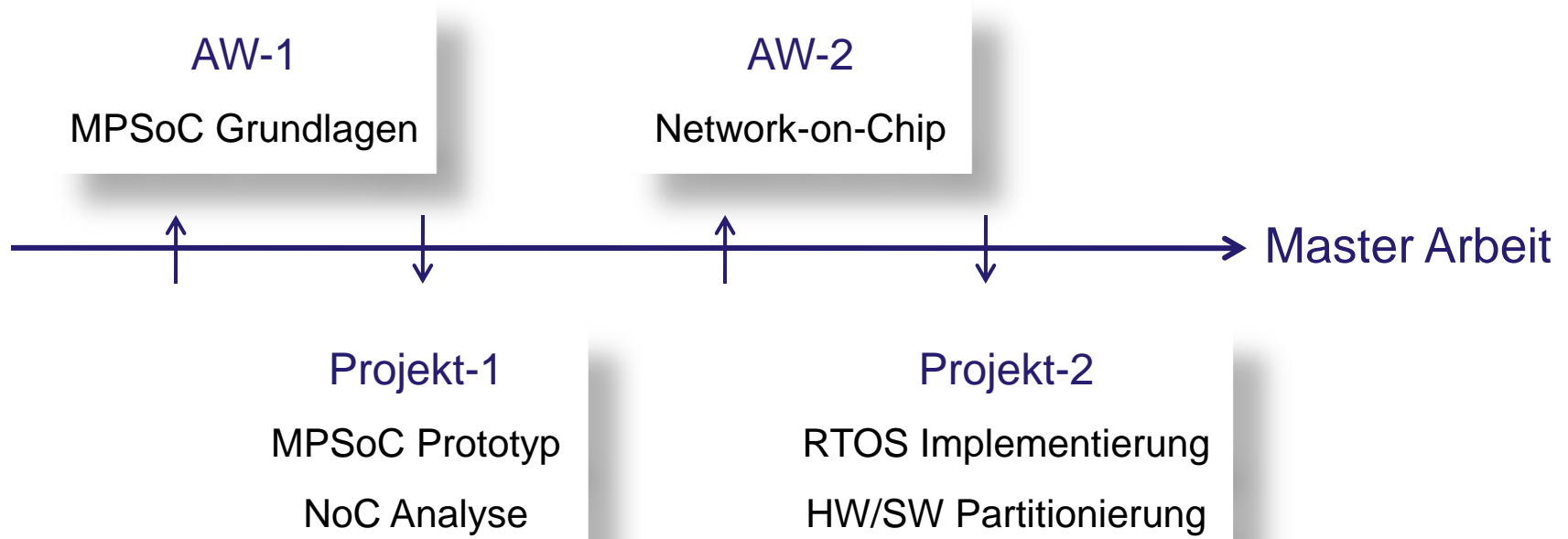


Symmetrische Multiprozessoren (SMP):

- Homogene Prozessoren mit gemeinsamen Adressraum
- Dynamische Verteilung von Aufgaben
- Zentrales RTOS mit „Uniform Memory Access“



Vorarbeiten & Aktueller Stand



Zielsetzung Master Arbeit

Entwicklung einer asymmetrischen Multiprozessorplattform

- Shared-Memory Architektur mit Synchronisation
- Interprozesskommunikation
- (Einsatz von Network-on-Chip als Verbindungsnetzwerk)

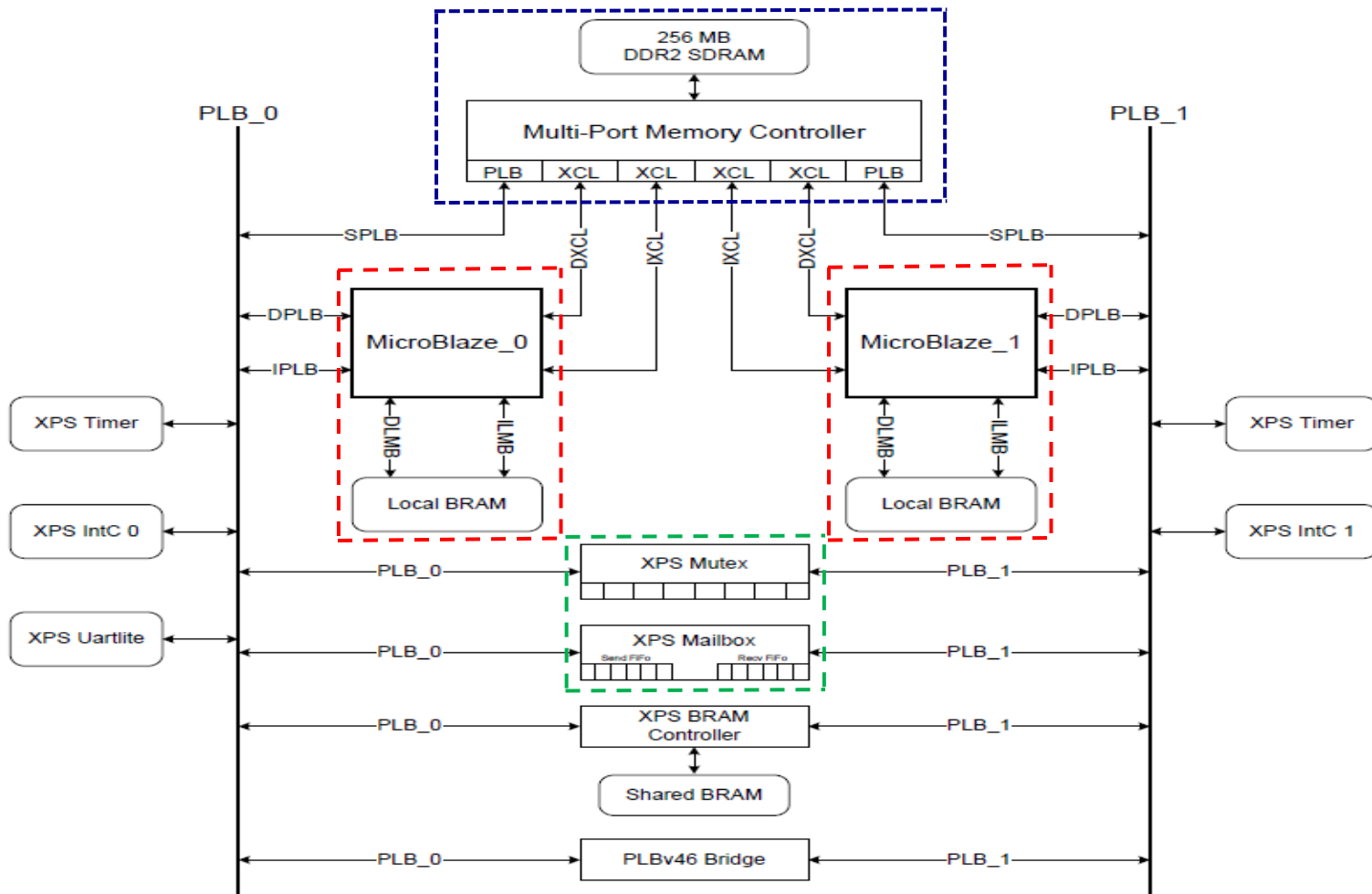
Portierung der FAUST SCV Software auf ein Xilinx Virtex-6 FPGA

- Hardware / Software Partitionierung
- Implementierung von zwei unabhängigen RTOS

Entwicklungsschwerpunkte

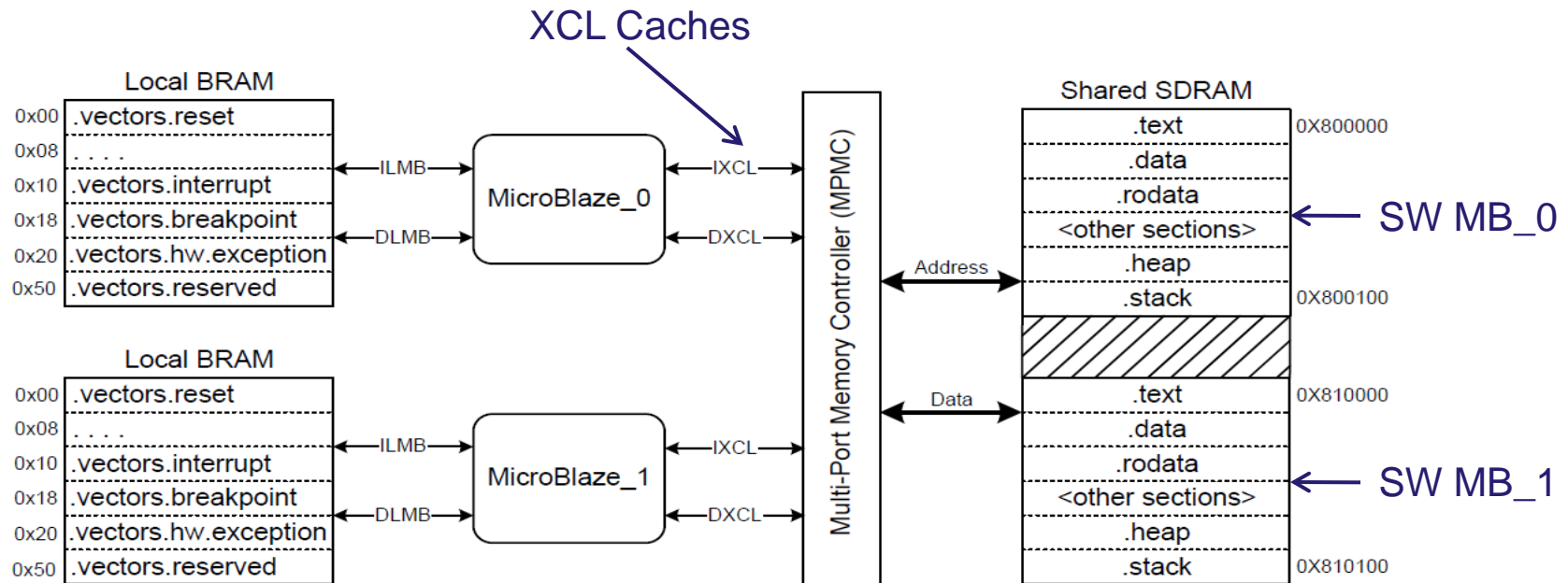
MPSoC Architektur

MPSoC mit Shared-Memory Architektur



1. Shared-Memory Speicherarchitektur

- Geteilte Adressräume (AMP) → Gemeinsamer Zugriff über MPMC
- Software Partitionierung → Lokaler BRAM Speicher für Bootsektor



2. Gewährleistung des Wechselseitigen Ausschlusses

Hardware: Instanziierung eines XPS Mutex IP

- 32 Memory-Mapped Mutexregister für „Lock“ und „Unlock“
- Zugriff und Kopplung über zwei getrennte PLB-Bussysteme

Software: Implementierung eines RTOS (MicroC/OS-II)

- Preemptives MicroC/OS-II mit bis zu 64 nebenläufigen Tasks
- Prioritäten basiertes Task-Scheduling (64 Prioritätsstufen)
- „Mutual exclusion semaphores“ / „Deterministic Execution Time“

3. Interprozesskommunikation zwischen Prozessoren

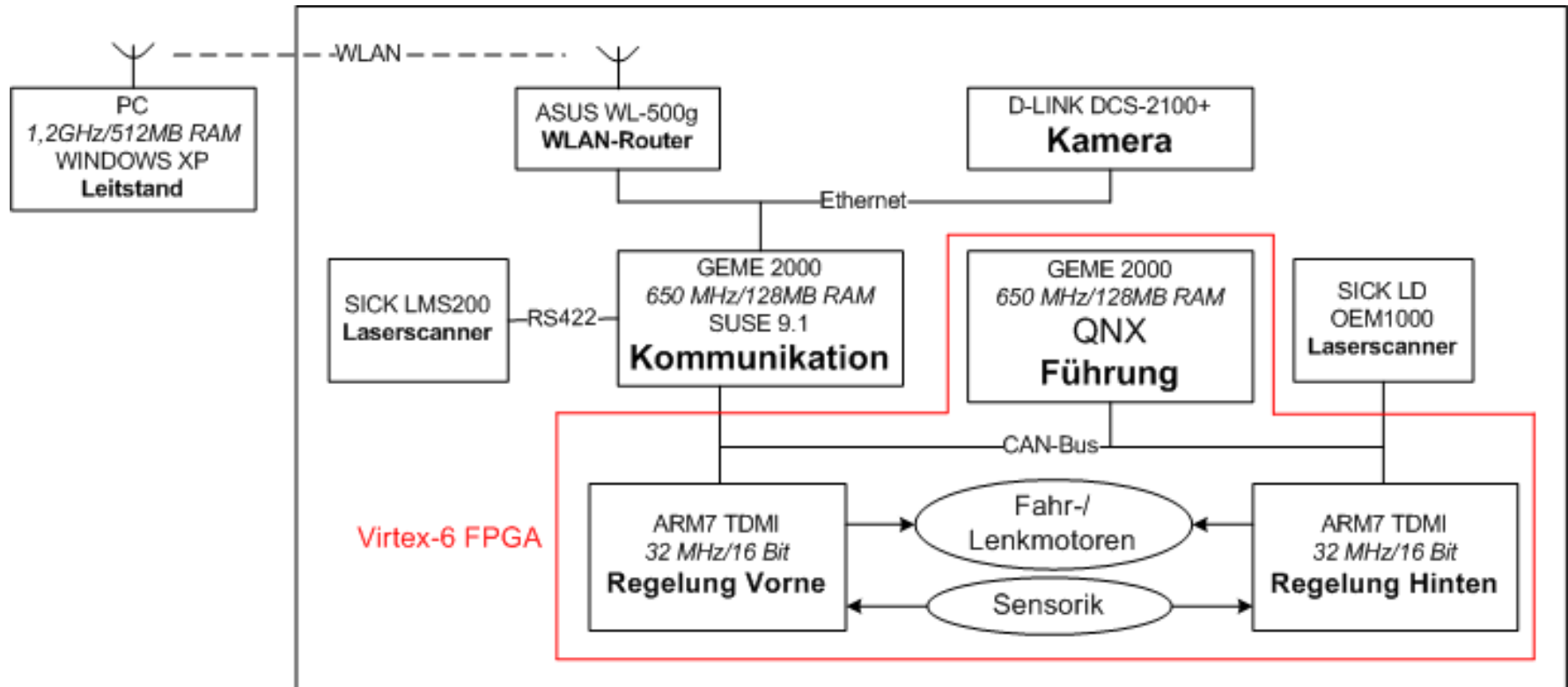
XPS Mailbox IP

- Zwei getrennt konfigurierbare FIFOs für Sender und Empfänger
- Kopplung von zwei getrennten PLB-Bussystemen
- Nachteil: Hoher Ressourcenbedarf und langsamer als FSL

FSL-Bus: Direkte Kommunikation der Prozessoren

- Unidirektionale Punkt-zu-Punkt Verbindung mit interner FIFO
- Vorteil: Schnell durch Kopplung der Registerfiles der Prozessoren

FAUST – Sensor Controlled Vehicle (SCV)



4. Hardware / Software Partitionierung

Portierung der SCV Fahrzeugführungs-Software auf ein Virtex-6 FPGA

- Auslagerung von unabhängigen Aufgaben auf verschiedene Prozessoren
- SW Parallelisierung von nebenläufigen Aufgaben durch Threads
- SW-Module der ARM μ C durch Hardware-Beschleuniger (IPs) ersetzen
- Komplexe Algorithmen auf HW-Portierung untersuchen

5. RTOS Implementierung auf Xilinx FPGA

RTOS	MicroBlaze	Bemerkung
μC/OS-II	+	kein SMP
Nucleus RTOS	+	Lizenzpflichtig
FreeRTOS	+	OpenSource
QNX Neutrino 2	-	nur PowerPC
Xilinx XilKernel	+	OpenSource

Risiken

- MPSoC:**
- Ressourcenverbrauch
 - Timing-Eigenschaften für Echtzeitsystem
 - Delay durch Interprozesskommunikation
- Shared-Memory:**
- Kein HW-Support für Cache Kohärenz
 - Speicherflaschenhals beim MPMC Zugriff
- FAUST SCV:**
- Keine Parallelisierbarkeit
 - Kein CAN IP Core für FPGA Implementierung
- Network-on-Chip:**
- Komplexität und Zeitaufwand
 - Ressourcenverbrauch für NI und Router

Zusammenfassung & Ausblick

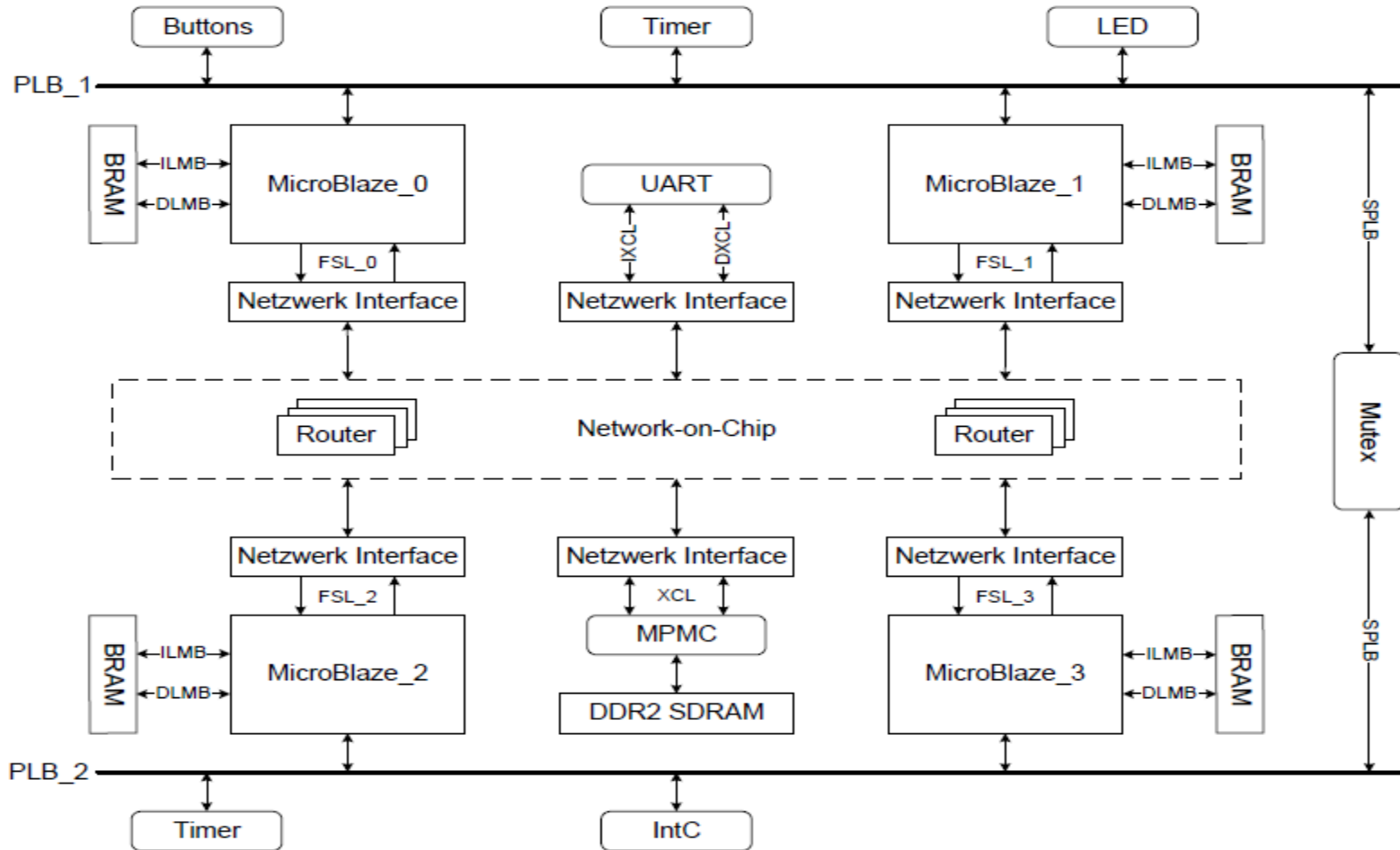
Aufbau einer FPGA-basierten Multiprozessorplattform (MPSoC)

- Portierung der SCV Fahrzeugsteuerungsplattform
- HW / SW Partitionierung der SCV Steuersoftware

Einsatz in einem „Distributed Computing System“

- Verteilung von FPGA Ressourcen
- Einsatz eines NoC's als Verbindungsnetzwerk

Zukunft: Network on Chip



Vielen Dank für Ihre Aufmerksamkeit

Fragen ?

Literaturverzeichnis

- [1] Jerraya, A. A., & Wolf, W. (2005). *Multiprocessor Systems-on-Chips*. Elsevier Verlag.
- [2] Micheli G., & Benini L. (2006). *Networks-on-Chips: Technology and Trends*. Elsevier Verlag.
- [3] Mahr T., & Gessler R. (2007). *Hardware/Software Codesign*. Vieweg+Teubner
- [4] S.W.Keckler, K.Olukotun, & H.P.Hofstee. (2009). *Multicore Processors and Systems*. Springer Verlag.
- [5] T.Rauber, & G.Rünger. (2008). *Multicore: Parallele Programmierung*. Springer Verlag.
- [6] Xilinx. (2007). *Designing Multiprocessor Systems in Platform Studio*. Xilinx White Paper.
- [7] Xilinx, & Asokan, V. (2008). *Dual Processor Reference Design Suite* . Xilinx Application Note.