

Kolloquium zur AW1-Ausarbeitung von Felix Kolbe

System-on-a-Chip Konzepte für
die Telemetrie- und Steuerungskomponenten
in einem Formula Student Rennwagen

Hochschule für Angewandte Wissenschaften Hamburg, 13.01.09

Vortragstruktur

- Ziel und Anwendung der Ausarbeitung
- Systemumgebung
 - ♦ Formula Student Rennwagen
 - ♦ Telemetriesysteme
 - ♦ Steuerungskomponenten
- Anforderungen der Systeme
 - ♦ Entlastung durch Hardware-Komponenten
 - ♦ HW/SW-Codesign und System-on-a-Chip
- Amdahl's Law
- Grundlagen der Partitionierung
- MicroBlaze als aktuelles SoC-Design
- Ausblick

Ziel und Anwendung der Ausarbeitung

- Überblick über Aspekte des SoC-Designs
- Aufdeckung von vertiefungsgerechten Themen

- Anwendungsbezug:
Telemetrie- und Fahrassistenzsystem des Formula Student Rennwagens
 - ♦ Verbesserung des Mikrocontroller-Systems durch HW-Komponenten bzw. SoC-Konzepte
 - ♦ Partitionierung anhand bestehender Software
 - ✓ *bisherige und zusätzliche gewünschte Funktionalität in SW und HW teilen*

Formula Student Rennwagen

- Studentische Einsitzer
- Entwicklungsziele:
 - Optimales Fahrverhalten
 - Hohe Beschleunigung
 - Hohe Zuverlässigkeit
 - Hohe Wartbarkeit
 - Geringes Gewicht
 - Geringer Preis



H04 in
Hockenheim
© FSG



Telemetriesysteme

- Messen, Protokollieren und Fernüberwachen des Fahrzeugzustandes
 - ♦ Fehlerdiagnose
 - ♦ Optimierungshilfe
 - ♦ Abwendung von Defekten
- 10 16-MHz- μ C mit Task-Scheduler
- 30 Sensoren aktuell, 90 im Zielkonzept



Fernüberwachung neben
der Rennstrecke

Steuerungskomponenten

- Fahrsicherheitssysteme
 - ◊ Antriebs-Schlupf-Regelung
 - ◊ Anti-Blockier-System
 - ◊ Elektronisches-Stabilitäts-Programm
- Fahrassistenzsysteme
 - ◊ Automatikschaltung
 - ◊ Startautomatik



H04 ohne
ASR in
Silverstone

Vortragstruktur

- *Ziel und Anwendung der Ausarbeitung*
- *Systemumgebung*
 - ♦ *Formula Student Rennwagen*
 - ♦ *Telemetriesysteme*
 - ♦ *Steuerungskomponenten*
- Anforderungen der Systeme
 - ♦ Entlastung durch Hardware-Komponenten
 - ♦ HW/SW-Codesign und System-on-a-Chip
- Amdahl's Law
- Grundlagen der Partitionierung
- MicroBlaze als aktuelles SoC-Design
- Ausblick

Anforderungen der Systeme

- Telemetrie- und Fahrassistenzsysteme
 - ♦ Eingebettete Systeme
 - ♦ Umfangreiche Anforderungen
 - ✓ *Geringer Stückpreis, geringer Energieverbrauch, Hohe Rechenleistung, Echtzeitanforderungen (Mess- & Regelzyklen)*

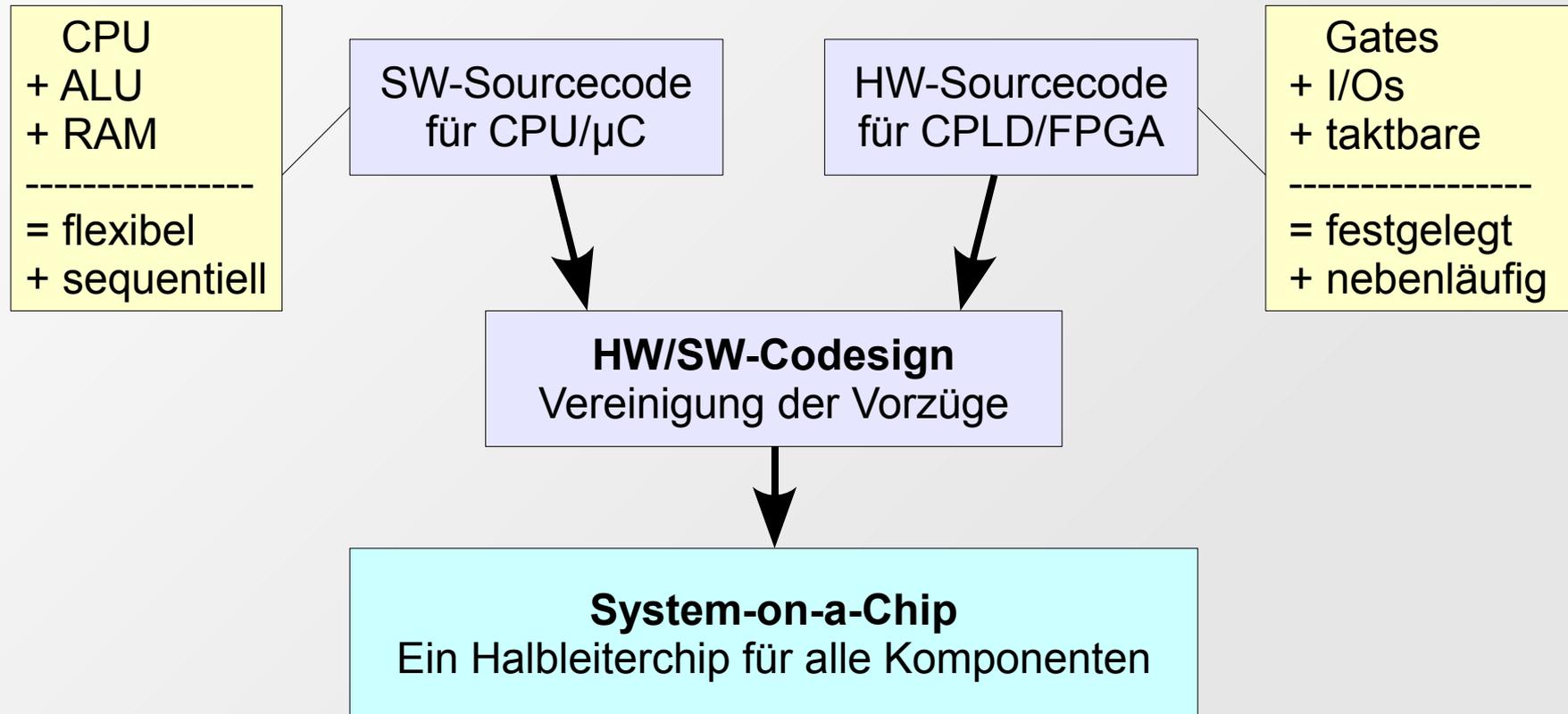
- Erhöhung der System-Rechenleistung durch:
 - ♦ Erhöhung durch Taktfrequenz
 - ✓ *Begrenzt möglich*
 - ✓ *Steigende Störanfälligkeit*
 - ♦ Hinzufügen weiterer CPUs/ μ C
 - ✓ *Erhöhter Kommunikationsaufwand*

- Andersweitige Entlastung von CPUs erforderlich

Entlastung durch HW-Komponenten

- SW/CPU kontrolliert den Programmablauf
 - ◊ Kommunikation
 - ◊ Fehlerbehandlung
- HW/FPGA ist Dienstleister für CPU
 - ◊ Beschleunigung von komplexen Berechnungen
 - ✓ *Prüfsummen*
 - ✓ *Filter*
 - ◊ Abwicklung von zyklischen Nebenaufgaben
 - ✓ *Sensorwerterfassung*
 - ✓ *Aktorbedienung (z.B. PWM)*

Hardware/Software-Codesign



Amdahl's Law

- Berechnung der maximalen Beschleunigung
 - Bisherige Ausführungszeit: t
 - Parallelisierbare Fraktion: $f \in [0..1]$
 - Nicht parallelisierbare Fraktion: $1-f$

$$t = f \cdot t + (1 - f) \cdot t$$

- Faktor der Parallelisierbarkeit von f : s

$$t' = \frac{f \cdot t}{s} + (1 - f) \cdot t$$

- Faktor der Gesamtbeschleunigung: s'

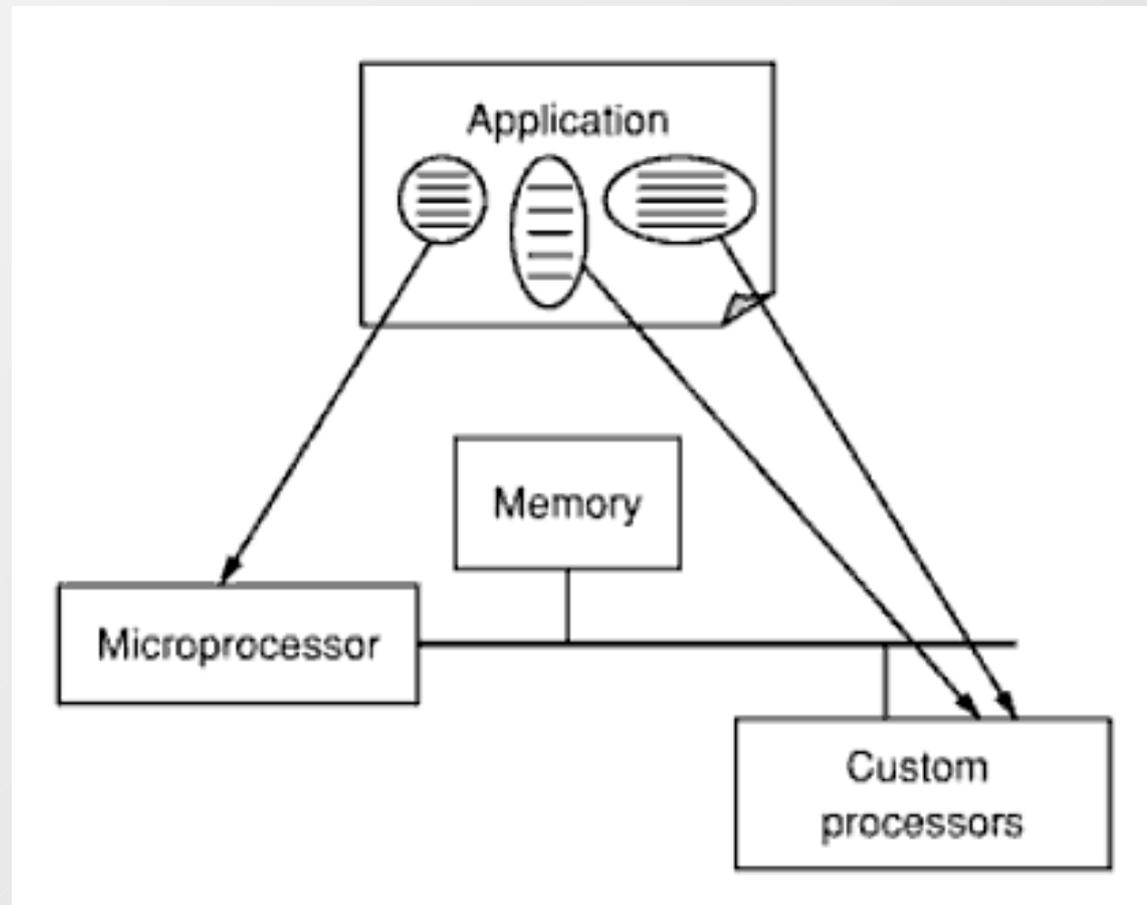
$$s' = \frac{t}{t'} = \frac{1}{\frac{f}{s} + (1 - f)}$$

Vortragstruktur

- *Ziel und Anwendung der Ausarbeitung*
- *Systemumgebung*
 - ♦ *Formula Student Rennwagen*
 - ♦ *Telemetriesysteme*
 - ♦ *Steuerungskomponenten*
- *Anforderungen der Systeme*
 - ♦ *Entlastung durch Hardware-Komponenten*
 - ♦ *HW/SW-Codesign und System-on-a-Chip*
- *Amdahl's Law*
- *Grundlagen der Partitionierung*
- *MicroBlaze als aktuelles SoC-Design*
- *Ausblick*

Grundlagen der Partitionierung

- Teilung der Funktionalität in Software- und Hardware-Anteile



[Vahid]

Regionen und Granularität

- Partitionierung: Funktionalität in Regionen aufteilen
- Region in SW xor HW implementiert
 - ◊ 2 Möglichkeiten pro Region
- n Regionen führen zu 2^n theoretischen Partitionierungen
 - ◊ Problem mit exponentieller Komplexität
- Regionengranularität:
 - ◊ Funktionen, Code-Blöcke, Schleifen
 - ◊ Komplexität vs. Optimierung
 - ◊ Unterschiedliche Granularität je nach Ausführungshäufigkeit

Partitionierungsmethoden

- Algorithmen
 - ♦ Knapsack problem
 - ♦ Iteratives Vorgehen
 - ♦ Kosten-Nutzen-Analyse
- Verfahrensoptimierungen
 - ♦ Kürzere Heuristiken statt Synthese
 - ♦ Erweiterte Charakterisierung der Regionen
- Ausgangskonstellation
 - ♦ All-in-SW, all-in-HW, all-blank
- Optimierungsziele (Metriken)
 - ♦ Performanz, Chipfläche
 - ♦ Energieverbrauch
 - ♦ Implementierungskosten, Entwicklungskosten
 - ♦ Zuverlässigkeit, Wartbarkeit

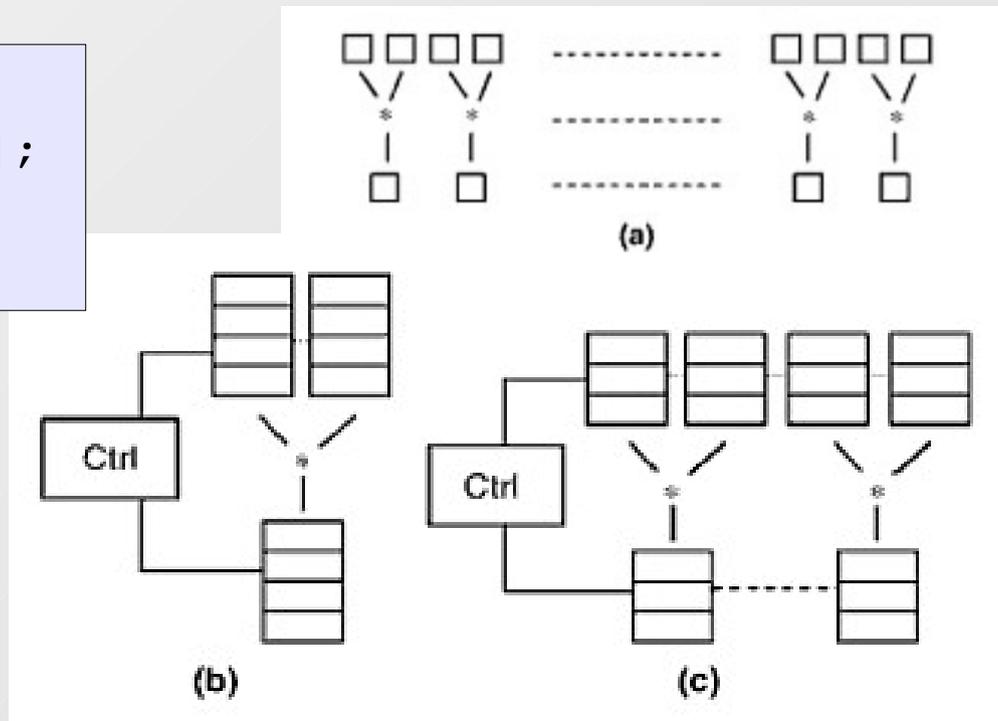
Alternative Implementations

- Verschiedene Implementierungen von Algorithmen
 - Unterschiedlich gut in SW/HW umsetzbar
 - Parallel -> schnell, viele Gatter
 - Sequentiell -> langsam, wenig Gatter
 - Gatterzahl vs. Berechnungszeit

```
// 100 Multiplikationen  
int a[100], b[100], c[100];  
for( i=0; i<100; i++ )  
    c[i] = a[i]* b[i];
```

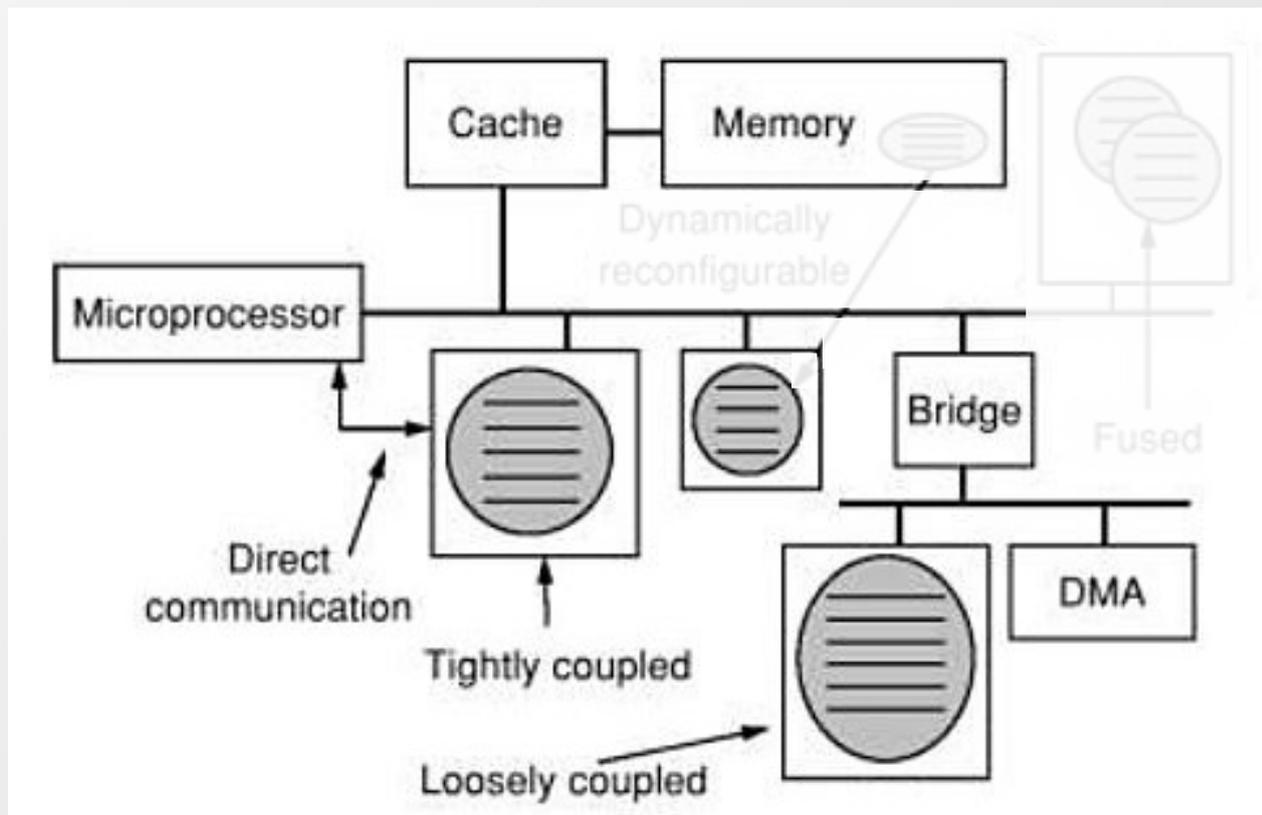
- Aufwand je nach Umsetzung:
 - a: 100 Multiplizierer, 1 Schritt
 - b: 1 Multiplizierer, 100 Schritte
 - c: 2 Multiplizierer, 50 Schritte

[Vahid]



Kopplung von CPU und FPGA

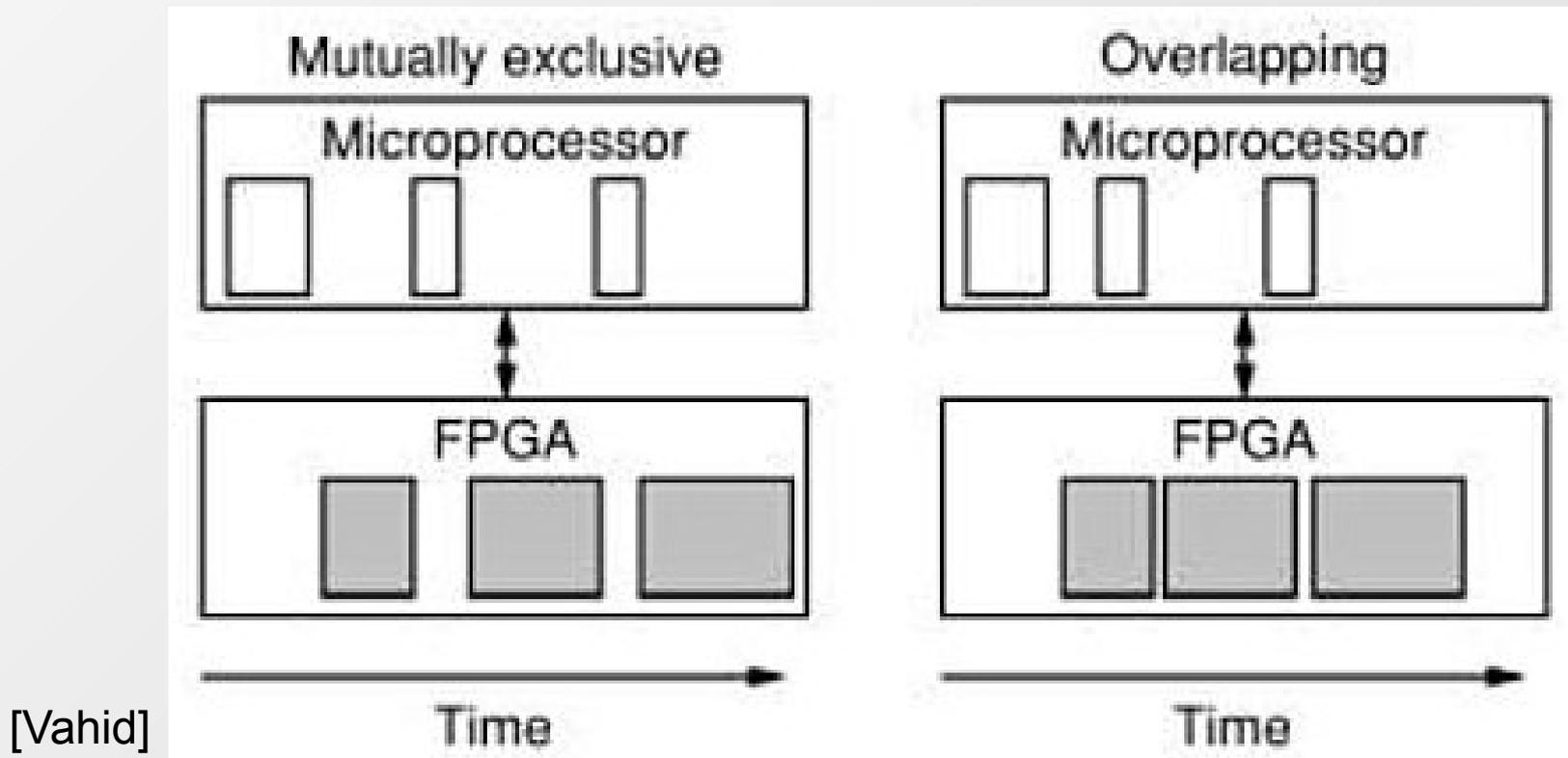
- Anbindung der Beschleuniger an CPU
 - Register: schnell, begrenzte Anzahl
 - Adress-Daten-Bus
 - Peripheriebus



[Vahid]

Ausführungsfenster von CPU und FPGA

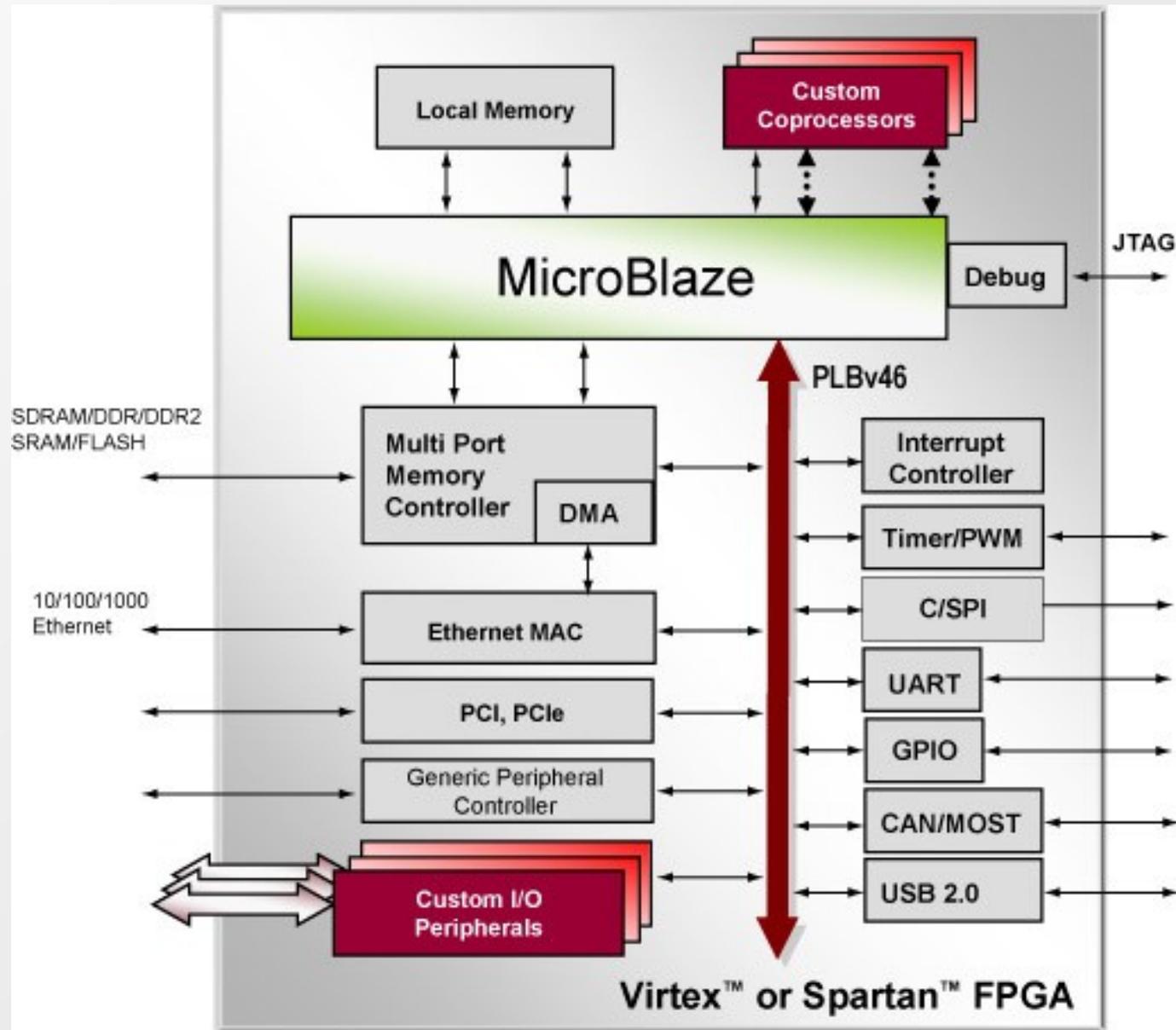
- alternierend oder überlappend
- Überlappung kürzt Leerlauf und Zyklusdauer, erfordert ggf. Mehraufwand durch Datensynchronisation



MicroBlaze als aktuelles SoC-Design

- Konfigurierbarer Softcore-Mikrocontroller für Xilinx-FPGAs als Element eines SoC
- Peripherie/Beschleuniger durch Konfiguration des restlichen FPGA-Anteils -> μC
 - ♦ Ankopplung an μBlaze über
 - ✓ *Processor Local Bus (IBM Core Connect Technologie)*
 - ✓ *FSL (Fast Simplex Link)*
 - ✓ *DCR (Device control Register)*
- Programmierbar wie üblicher μC

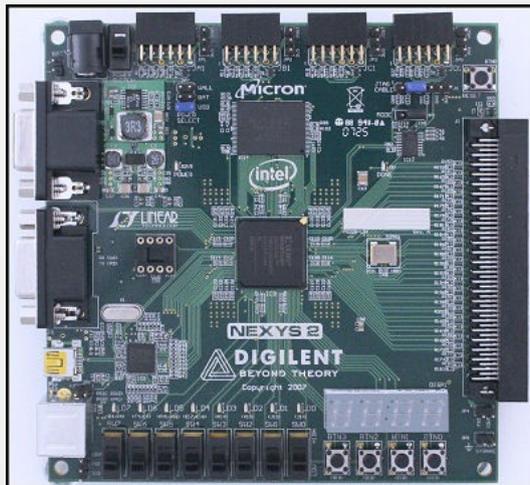
Architektur des MicroBlaze-SoCs



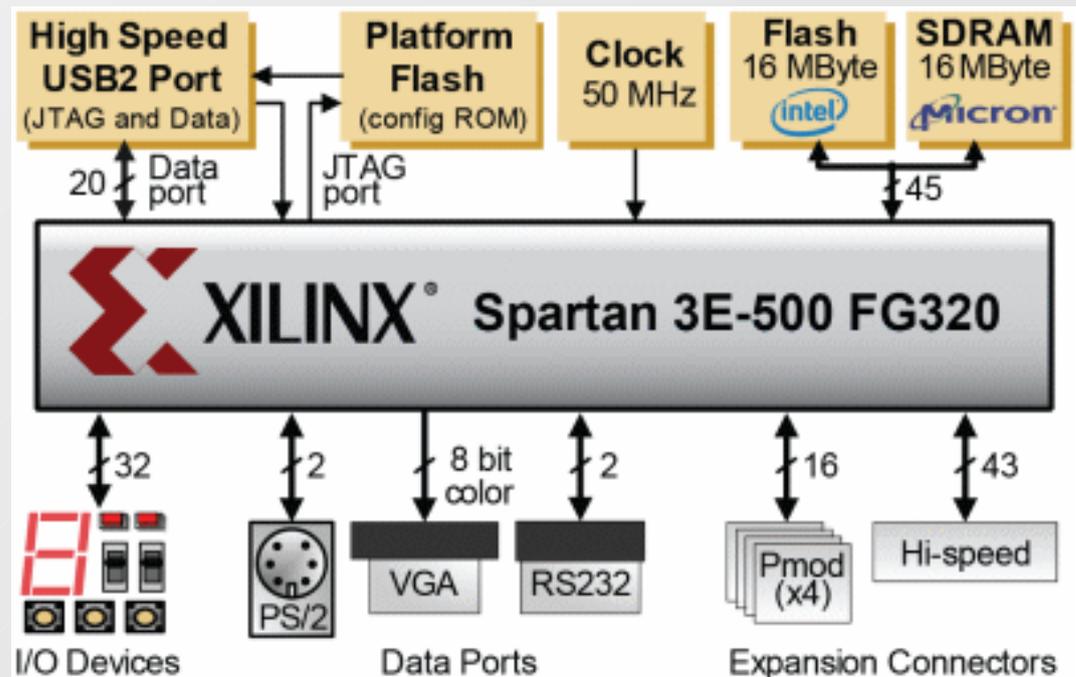
[Xilinx]

SoC mit Nexys-2 Entwicklungsboard

- HW-Beschleuniger FPGA XC3S1200E
 - 1,2 Mio Gatter
 - Zum Vergleich: μ P68000 mit ca. 17100 Gattern
- Softcore Mikrocontroller
 - 3 oder 5-stufige Pipeline-ALU
 - 32 Bit RISC
 - 50 MHz
- 112\$



[Xilinx]
[Digilent]



Ausblick

- Methodisch:
 - ♦ Algorithmen und Heuristiken
 - ♦ Automatisierte Partitionierung
 - ✓ *Modellierung von HW auf der Ebene von SW*
 - × *z.B. mit SystemC, SystemVerilog*
 - ✓ *XPRES Compiler*
 - ✓ *Alternative Implementations*
 - ✓ *Erschwert durch Pointer, Memalloc und Rekursion*
- Operativ:
 - ♦ Alternativlösung zu bestehendem System des Rennwagens entwickeln
 - ✓ *auf Basis eines SoC-Konzeptes*
 - ✓ *z.B. mit dem Nexys-2-Entwicklungsboard*

Vortragstruktur

- Ziel und Anwendung der Ausarbeitung
- Systemumgebung
 - ♦ Formula Student Rennwagen
 - ♦ Telemetriesysteme
 - ♦ Steuerungskomponenten
- Anforderungen der Systeme
 - ♦ Entlastung durch Hardware-Komponenten
 - ♦ HW/SW-Codesign und System-on-a-Chip
- Amdahl's Law
- Grundlagen der Partitionierung
- MicroBlaze als aktuelles SoC-Design
- Ausblick

Quellenverweise

- [Vahid] — VAHID, F. ; STITT, G.: Hardware/Software Partitioning. Kap. 26, S. 539–560. In: HAUCK, S. (Hrsg.) ; DEHON, A. (Hrsg.): Reconfigurable Computing : The Theory and Practice of FPGA-Based Computation, Morgan Kaufmann/Elsevier, 2008. – ISBN 0-12-370522-3
- [Xilinx] — XILINX, Inc.: MicroBlaze Processor. 2008. – URL http://www.xilinx.com/products/design_resources/proc_central/microblaze.htm – Abruf: 2009-01-12
- [Digilent] — Digilent, Inc.: Nexys-2. 2008. – URL <http://www.digilentinc.com/Products/Detail.cfm?Prod=NEXYS2> – Abruf: 2009-01-12