



**TECHNISCHE  
UNIVERSITÄT  
DRESDEN**

**Fakultät Informatik** Institut für Technische Informatik, Professur für VLSI-Entwurfssysteme, Diagnostik und Architektur

# **Untersuchung zur hardwareunterstützten Entwurfsverifikation von Stream-basierten Kommunikations- und Verarbeitungsalgorithmen**

**Diplomverteidigung**

**Jan Schirok**

Dresden, 04.03.2009

# Gliederung

Motivation

Simulation im Entwurf

Simulationsplattform

Ergebnisse

Zusammenfassung und Ausblick

## Motivation

- Simulation ist essentieller Bestandteil im Entwurf integrierter Schaltkreise
  - Fokus: Schaltungssimulation auf Register-Transfer-Ebene (RTL)
- Steigende Komplexität integrierter Schaltkreise
  - Geschwindigkeit der softwarebasierten, sequentiellen Simulatoren sinkt
  - Zahl der Simulationsläufe steigt
- Notwendigkeit einer Simulationsbeschleunigung
  - Unterstützung durch an Host-PC angebundene FPGA-basierte Hardware
  - Problemstellung: Kommunikation zwischen Hard- und Software
- Einschränkung des Anwendungsbereichs
  - Stream-basierte Kommunikations- und Verarbeitungsalgorithmen
  - Latenzverdeckung der Kommunikation zwischen Hard- und Software möglich

# Simulation im Entwurf

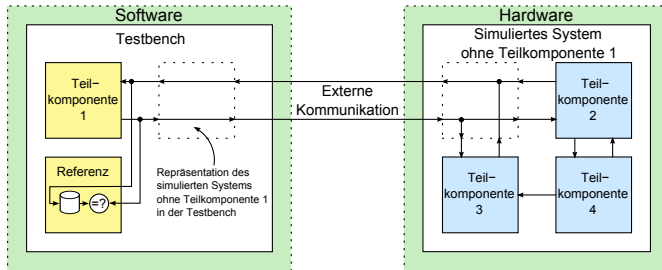
## RTL Schaltungssimulation

- Systembeschreibung auf Register-Transfer-Ebene
- Komponenten in Hardwarebeschreibungssprache
- Verschiedene Anwendungsfälle
  - Verifikation erfordert sehr hohe Simulationsgeschwindigkeit
  - Fehlersuche benötigt zusätzlich Sichtbarkeit in das System und geringe Vorbereitungsdauer
- Nutzung von FPGAs zur Beschleunigung der Simulation
  - Synthese der RTL Beschreibung in FPGA Strukturen
  - Stimulation und Auslesen der Module: Ein- und Ausgänge im FPGA auf Signal-/Portebene
  - Kopplung an Host-PC z.B. in Form einer Einsteckkarte

# Simulation im Entwurf

## Hardwarebeschleunigung

- Partitionierung in Hardware- und Softwareteil
  - Systemszenario (Hardware: RTL-basiertes System, Software: Stimuli und Vergleich der Ausgangsdaten)
  - Komponentenszenario:



# Simulation im Entwurf

## Kommunikation

- Externe Kommunikation zwischen Hard- und Software
- Interne Kommunikation zwischen Partitionen (Teil der Modellierung)
- Verschiedene Muster der internen Kommunikation
  - **Einfaches Request-Response Schema**  
Signaländerung in jedem Takt möglich (intern)  
Stimulation — Takt — Auslesen (extern)
  - **Aktionsbasierte Kommunikation**  
Nur Stimulation und Auslesen falls notwendig
  - **Transaktionsbasierte Kommunikation**  
Transaktionen verarbeiten, Anfragen senden, Antworten empfangen
- Externe Kommunikation kritisch für Simulationsgeschwindigkeit
- Nutzung der Muster der internen Kommunikation zur Verringerung des Datenaufkommens der externen Kommunikation und zur Simulationsbeschleunigung

# Simulationsplattform

## Eigenschaften

- PCI Express Einsteckkarte mit Altera Stratix II GX FPGA
- Einbindung der Hardwareplattform in Linux als Kernelmodul
- Kopplung an ModelSim SE per *Foreign Language Interface*
- Umsetzung der Kommunikationsmuster in zwei Prototypen
  - Einfaches Request–Response Schema und aktionsbasierte Kommunikation übertragen Signaldaten  
⇒ Bitebenenprototyp
  - Transaktionsbasierte Kommunikation überträgt Anfragen und Antworten  
⇒ Transaktionsbasierter Prototyp
- Richtung der Aktionen bzw. Anfragen festgelegt (Hardware zu Software)

# Simulationsplattform

## Bitebenenprototyp

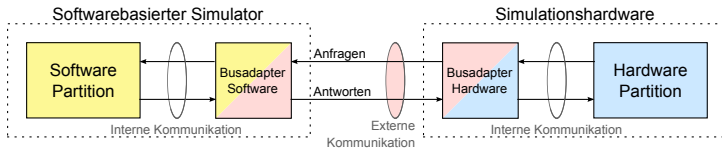
- Übertragung von Eingangs-/Ausgangssignalen der Partitionen pro simuliertem Takt
- Aktionsbasierte Kommunikation: taktweise Aktionen und erwartete Reaktionen
- Phasen ohne Aktionen oder erwartete Reaktionen erfordern keine taktweise Kommunikation (höhere Granularität)
- Unterstützung dieser Phasen durch zwei Parameter
  - NREAD: Maximale Anzahl gleichzeitig übertragener Takte
  - NSKIP: Maximale Anzahl übersprungener Takte
  - Anzahl der übertragenen bzw. übersprungenen Takte wird zusätzlich durch eintretende Aktionen und erwartete Reaktionen begrenzt
- Standardfall: NREAD=1, NSKIP=0, Übertragung in jedem Takt



# Simulationsplattform

## Transaktionsbasierter Prototyp

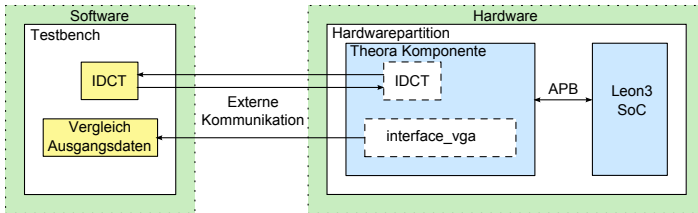
- Übertragung von Transaktionen
  - Anfragen von Hardwarepartition
  - Antworten von Softwarepartition
- Angepasste, zyklengenaue Busadapter notwendig
- Kommunikation mit Busadapters innerhalb der Hardware FIFO-basiert
- Nebenläufigkeit der Hardwarepartition zum softwarebasierten Simulator möglich



# Ergebnisse

## Gewähltes Beispiel

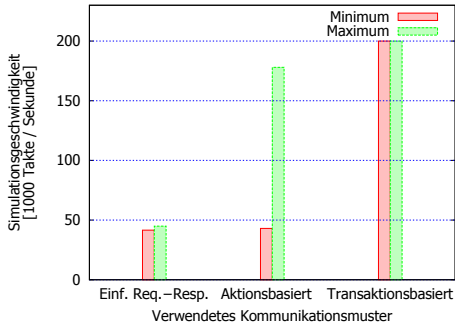
- Dekodierung von *Ogg Theora* Videodaten
- Quelloffenes, eingebettetes System auf Basis des Leon3 Prozessors
- Beschleunigung der Videodekodierung durch Theora Komponente
- IDCT (*Inverse Discrete Cosine Transform*): Teiloperation der Dekodierung als Softwarepartition



# Ergebnisse

## Simulationsgeschwindigkeit / Kommunikationsmuster

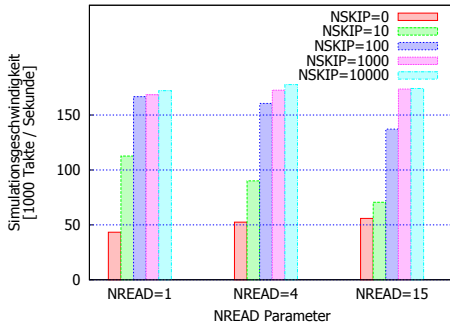
- Geschwindigkeit rein softwarebasierter Simulation: 77,4 Takte pro Sekunde
- Erreichte Geschwindigkeiten der untersuchten Kommunikationsmuster:



# Ergebnisse

## Einfluss der Parameter beim Bitebenenprototyp

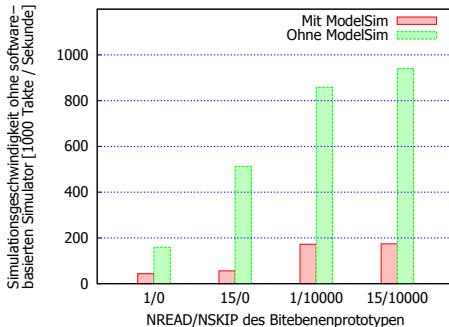
- Weite Geschwindigkeitsspanne durch Variation von NSKIP/NREAD erreichbar
- NREAD=1, NSKIP=0 entspricht einfachem Request-Response Schema



# Ergebnisse

## Einfluss des softwarebasierten Simulators

- Vergleich zwischen ModelSim und Testprogramm mit geringer Rechenzeit
- Kommunikation ist in beiden Fällen Teil der Messung



## Zusammenfassung

- Beschleunigung der Simulation auf RT Ebene erreicht  
⇒ Günstig für Verifikation und Fehlersuche
- Sicht in Teilkomponenten durch Komponentenszenario  
⇒ Fehlersuche in diesen Teilkomponenten  
⇒ Geringe Vorbereitungsdauer bei Änderungen in der Softwarepartition
- Untersuchung verschiedener Kommunikationsmuster
  - Request-Response Schema: universell, geringste Beschleunigung
  - Aktionsbasierte Kommunikation: parametrierbar, mittlere Beschleunigung
  - Transaktionsbasierte Kommunikation: hohe Anforderungen an simuliertes System, höchste Beschleunigung
  - Latenzverdeckung gegenüber Request-Response Schema
- Aktionsbasierte Kommunikation erscheint für Simulationen auf RT Ebene als günstigste Variante

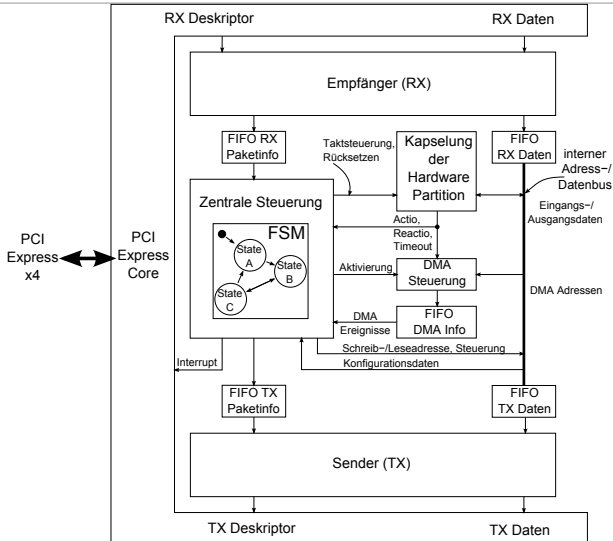
## Ausblick

- Potential der Hardwarelösung bei aktions- und transaktionsbasierter Kommunikation nicht ausgeschöpft
- Verallgemeinerung der Kommunikationsmuster auf weitere Algorithmenklassen
- Nutzung der Simulationsbeschleunigung zur Messung von Systemeigenschaften (Latenzen, Durchsätze)
- Weiterführende Untersuchungen der Simulation mit nebenläufiger Hardwarepartition
- Kombination des transaktionsbasierten Prototypen mit datenflussbasierten Simulatoren

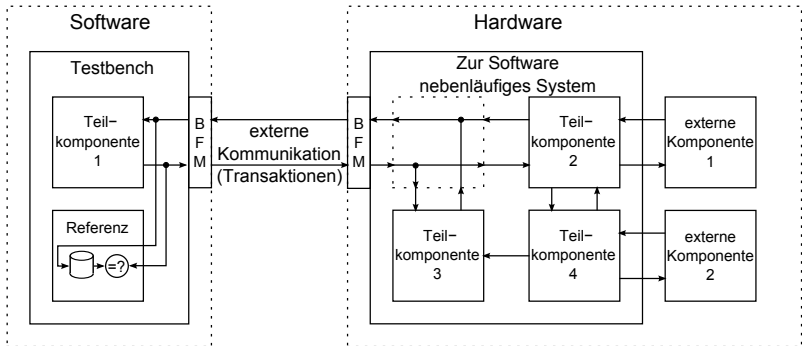
Vielen Dank für Ihre Aufmerksamkeit.







# Nebenläufige Simulation



# Aktionsbasierte Kommunikation

- Voraussetzungen
  - Definition von Aktion (bzw. Anfrage) — z.B. Reset-Eingang eines Zählers, Chip Enable Leitung am Speicherbaustein.
  - Definition von Reaktion (bzw. Antwort) — z.B. Increment eines Zählers, Setzen der Datenleitungen durch den Speicherbaustein
  - Richtung von Aktion und Reaktion festgelegt
  - Keine selbstinitiierte Reaktion (nur erwartete)
- Phasen ohne Aktion und erwartete Reaktion: überspringen — z.B. kein Buszugriff
- Phasen mit Aktion ohne erwartete Reaktion: zusammenhängend übertragen — z.B. nur Schreibzugriffe
- Phasen mit erwarteter Reaktion: Einzelübertragung, Reaktion erwarten — z.B. Lesezugriff