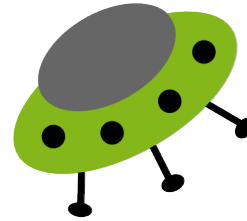


# UPHOS



Utilizing Programmable Hardware in Operating Systems

**Arbeitsgruppe**  
**Eingebettete Systemsoftware**  
**TU Dortmund**

Lehrstuhl  
Systemarchitektur  
Universität Karlsruhe

**Michael Engel**  
**GI-Fachgruppentreffen**  
**Betriebssysteme**  
**Wiesbaden, 11. März 2008**



# Überblick

---

- FPGAs und Rekonfigurierung
  - Aktuelle Trends
  - Programmierung von FPGAs
  - Dynamische Rekonfigurierung
- FPGAs in eingebetteten Systemen
  - Systemstrukturen
  - Einsatzmöglichkeiten
- UPHOS
  - Bisherige Arbeiten
  - Projektziele
- Ausblick



# Überblick

- **FPGAs und Rekonfigurierung**
  - Aktuelle Trends
  - Programmierung von FPGAs
  - Dynamische Rekonfigurierung
- **FPGAs in eingebetteten Systemen**
  - Systemstrukturen
  - Einsatzmöglichkeiten
- **UPHOS**
  - Bisherige Arbeiten
  - Projektziele
- **Ausblick**



# Rekonfigurierbare Hardware: FPGAs

- FPGA: „Field Programmable Gate Array“
  - Halbleiter-Baustein, der programmierbare Logik-Komponenten („Logikblöcke“) und programmierbare Verbindungen beinhaltet
  - Logikblöcke können einfache Funktionen (AND, XOR) oder komplexere Funktionen (Decoder, einfache mathematische Funktionen) implementieren
  - Die meisten FPGAs beinhalten zudem eine begrenzte Menge an stat. RAM (Flip-Flops)
  - Konfiguration (Festlegung der Funktionalität) des FPGA ist änderbar über spezielle Programmierwerkzeuge
- Kapazitäten aktueller FPGAs:
  - Xilinx Spartan XC3S400: 3584 „slices“, ~250 kbit RAM
  - Xilinx Virtex IIp XC2VP30: 12032 slices, ~2.5 Mbit RAM



# Motivation

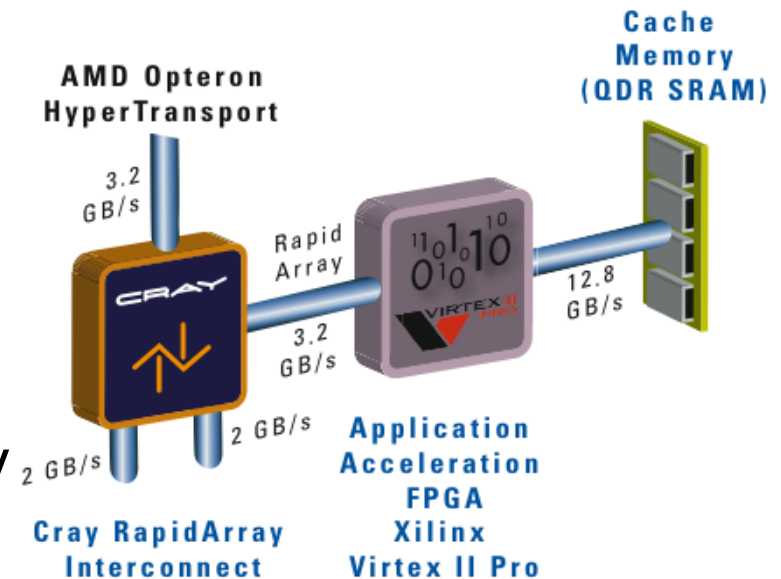
---

- Rekonfigurierbare Hardware in FPGAs...
  - ist günstig geworden
  - hat hohe Kapazitäten und Taktraten
  - ermöglicht echt parallele Ausführung
- aber...
  - hat ein deutlich abweichendes „Programmier“modell (eigentlich: Konfigurationsmodell)
  - es existieren viele verschiedene Standards bei Spezifikationssprachen, Hardware-Realisierungen usw.
  - momentan werden FPGAs als Applikationsbeschleuniger nur von einzelnen Spezialanwendungen genutzt



# Beispiel: Cray XD-1

- AMD Opteron CPUs (bis zu 12 x 12), HyperTransport
- Xilinx Virtex II pro / Virtex 4 FPGAs (optional)
- 16 GB QDDR Cache für FPGA
  - SRAM (!), 12.8 GB/s Bandbreite
  - mapped in OS virtual memory
- 64-bit-Register in FPGA von BS aus zugreifbar
- Any-to-any – jeder FPGA von allen CPUs aus nutzbar
- In-execution reprogrammability
- Linux als BS
- Celoxica-C, Mitrion-C+Xilinx Tools





# Programmiermethoden für FPGAs

- Schaltungsentwurf auf Registerebene
    - ähnlich traditioneller Entwicklung mit 74LSxx
  - Hardware-Beschreibungssprachen
    - VHDL, Verilog
  - Compiler für „traditionelle“ Programmiersprachen
    - Handel-C (Celoxica)
    - DIME-C (Nallatech)
    - Impulse-C (Impulse Accelerated Technologies)
    - Mitrion-C
  - C++-Klassenbibliothek für Hardware-Entwurf
    - SystemC
- Die Grenze zwischen Hard- und Software verschwimmt!



# Rekonfigurierbare Hardware

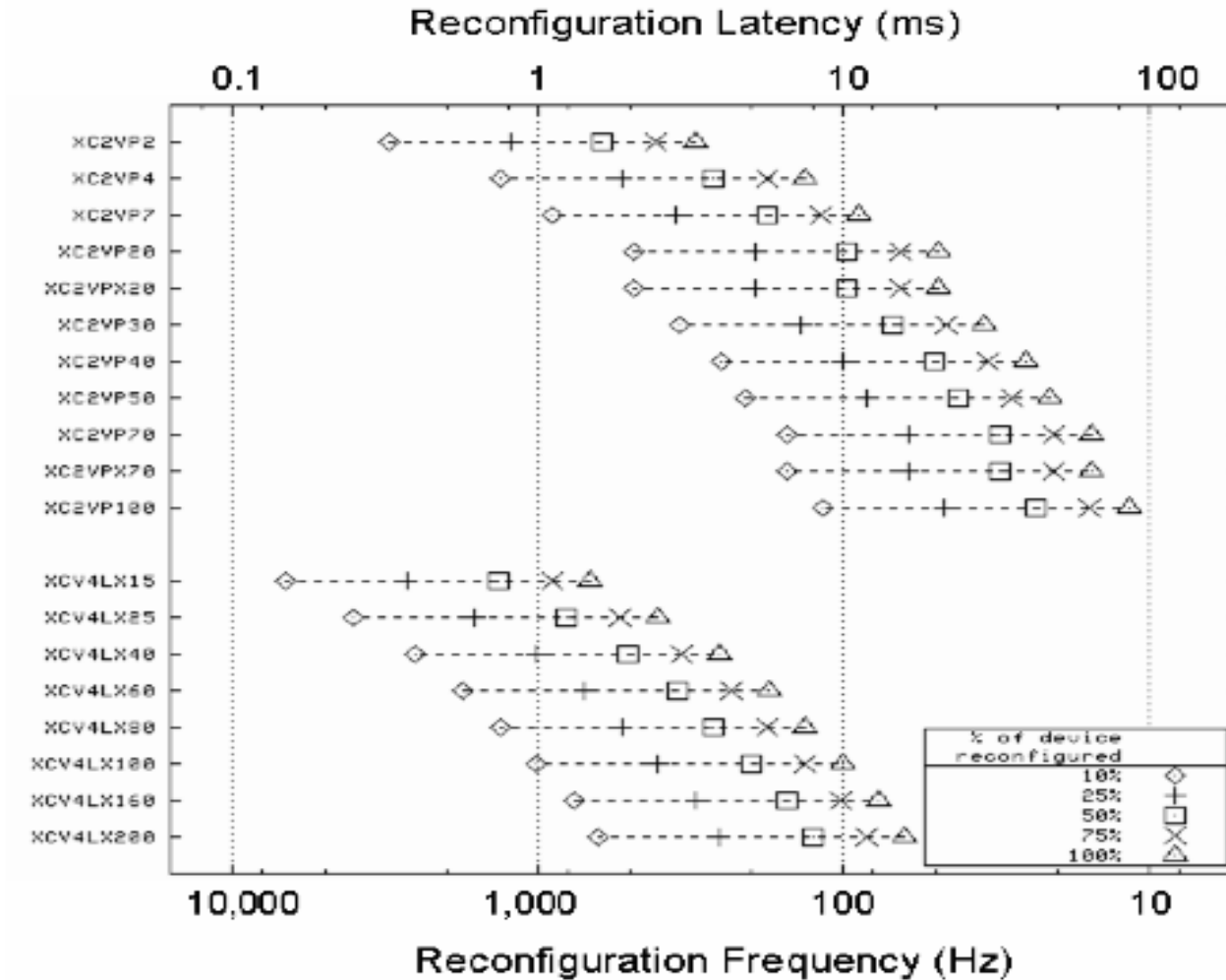
---

- FPGAs sind programmierbare Hardware
  - normalerweise aber nur beim Systemstart!
- Problem: Rekonfigurierung im Betrieb
  - ist partielle Rekonfigurierung machbar?
  - erzielbare Rekonfigurierungsrate?
  - Layout und Interconnect Management des FPGA
  - Schutzkonzepte?
- Rekonfigurierung ist nicht offiziell unterstützt
  - Aber durchaus realisierbar [1]
  - Xilinx Tools (impact) können partielle Bitstreams per JTAG in FPGA laden (early access)





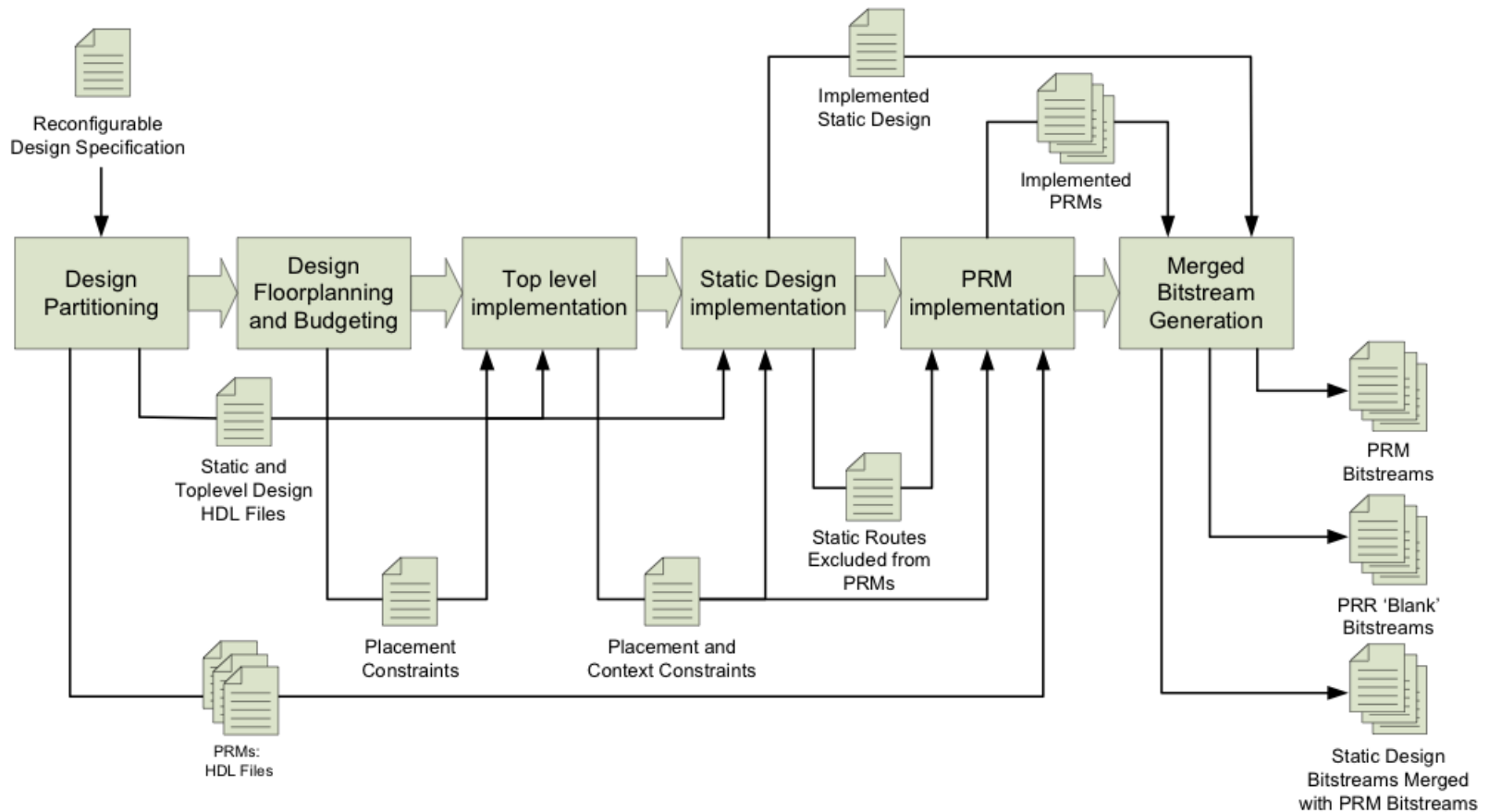
# Rekonfigurierbare Hardware





# Rekonfigurierbare Hardware

- Design Flow:





# Überblick

- FPGAs und Rekonfigurierung
  - Aktuelle Trends
  - Programmierung von FPGAs
  - Dynamische Rekonfigurierung
- FPGAs in eingebetteten Systemen
  - Systemstrukturen
  - Einsatzmöglichkeiten
- UPHOS
  - Bisherige Arbeiten
  - Projektziele
- Ausblick



# Konfigurationen

---

- Konventionelle CPU + FPGA
  - PCI(e)-Gerät oder Bus-Anbindung (z.B. HyperTransport)
  - Oft mit lokalem Speicher
- CPU als „hard“ core auf FPGA integriert
  - Xilinx Virtex IIpro, Virtex 4, Virtex 5
  - 1 bis 4 PowerPC 405 cores on chip (300-450 MHz)
  - Peripherie der Prozessoren als soft core im FPGA
- CPU als „soft“ core auf FPGA integriert
  - CPU als VHDL- oder Verilog-Beschreibung
  - CPUs von 8 bis 64 Bit, CISC/RISC/VLIW realisierbar
  - vergleichsweise geringe Taktrate (10-100 MHz)



# Anwendungsbeispiele

---

- Auslagerung rechenintensiver Prozesse in FPGA
  - Geringere Rechenleistung in System-CPU erforderlich
  - Signal- und Bildverarbeitung
    - En-/Decodierung von Audio- und Videoformaten
  - Kryptographie
    - Sichere Kommunikation und Datenhaltung
  - Netzwerk
    - Protokoll-Verwaltung in Hardware
- Integration von Legacy-Systemen
  - Nicht mehr erhältliche Komponenten (RoHS) als soft cores auf FPGA realisierbar
    - Sicherung der Investition in Software und Werkzeuge
    - Weiterverwendung zertifizierter SW-Komponenten



# Anwendungsbeispiele

---

- Neuartige Peripheriegeräte
  - Implementierung von flexibleren Ein-/Ausgabemethoden
    - Software Defined Radio
- Flexibilität des Systems
  - Implementierung von Geräten in programmierbarer Hardware
    - Aktualisierung von Ein/Ausgabe-Komponenten im laufenden System
    - Fehlerbehebung
    - Anpassung an neue Protokoll-Anforderungen



# FPGAs in HW/SW-Systemen

Probleme der Verwendung von FPGAs:

- Definition von Schnittstellen
  - Zugriff auf FPGA von Anwendungen und Betriebssystem
- Definition von Verwaltungsinfrastrukturen
  - Isolation und Integration verschiedener Komponenten auf einem FPGA
- Effizientes „Scheduling“ der verfügbaren FPGA-Fläche
  - Reloizierung von Komponenten möglich?
- On-Chip Kommunikationsinfrastrukturen für FPGAs
  - Network-on-Chip (NoC)
  - Schnittstelle zu sonstigen Netzwerkinfrastrukturen?



# FPGAs in eingebetteten Systemen

---

Probleme durch spezielle Eigenschaften eingebetteter Systeme:

- Begrenzter Speicher und Rechenleistung
  - Einfache, kleine Lösungen zur Rekonfigurierung
- Kostenfaktor
  - FPGA-Ressourcen effizient nutzen
- Energiefaktor
  - Software vs. Hardware-Implementierung von Funktionalität unter Energiegesichtspunkten
- Generierung rekonfigurierbarer Komponenten
  - Werkzeuge der FPGA-Hersteller sind proprietär
  - Nur als x86-Binärcode (Windows, Linux) verfügbar
- Durchgängige Konfigurierung von Hard- und Software





# FPGAs in eingebetteten Systemen

Probleme durch spezielle Eigenschaften eingebetteter Systeme:

- Begrenzter Speicher und Rechenleistung
- Diese Probleme sollten eigentlich durch ***Betriebssystem-Unterstützung*** gelöst werden
- Generierung rekonfigurierbarer Komponenten
  - Werkzeuge der FPGA-Hersteller sind proprietär
  - Nur als x86-Binärcode (Windows, Linux) verfügbar
- Durchgängige Konfigurierung von Hard- und Software



# State of the Art

---

- G. Wigley, D. Kearney:  
„*The First Real Operating System for Reconfigurable Computers*“
  - Traditionelle BS-Aufgaben und deren Adaption für FPGAs
- P. Merino, J.C. Lopez, M. Jacome:  
„*A Hardware OS for Dynamic Reconfiguration of FPGAs*“
  - Ress.-Verwaltung: Partitionierung durch Menge v. VHDL-Prozessen
  - Keine Integration in ein Host-OS
- P. Knezević, B. Radunović, A. Elhouni:  
„*Issues in FPGA-based Configurable Computer Design for Operating Systems and Multimedia Applications*“
  - Diskussion der Implementierung von Algorithmen aus dem Bereich Multimedia und BS auf FPGAs, kein BS-Kontext
- H. Kwok-Hay So, R. Brodersen:  
„*Improving Usability of FPGA-based Reconfigurable Computers Through Operating System Support*“
  - BORPH (Berkeley) – rekonfig. HW-Plattform + Linux für HPC



# State of the Art

- G.Wigley, D. Kearney:  
„*The First Real Operating System for Reconfigurable Computers*“
  - Traditionelle BS-Aufgaben und deren Adaption für FPGAs
- P. Merino, J.C. Lopez, M. Jacome:

Wenige Arbeiten

Keine im Bereich der eingebetteten Systeme!

- „*Improving Usability of FPGA-based Reconfigurable Computers Through Operating System Support*“
  - BORPH (Berkeley) – rekonfig. HW-Plattform + Linux für HPC



# Überblick

- FPGAs und Rekonfigurierung
  - Aktuelle Trends
  - Programmierung von FPGAs
  - Dynamische Rekonfigurierung
- FPGAs in eingebetteten Systemen
  - Systemstrukturen
  - Einsatzmöglichkeiten
- UPHOS
  - Projektziele
  - Bisherige Arbeiten
- Ausblick



# UPHOS-Projekt

Kooperation TU Dortmund und Universität Karlsruhe

- Arbeitsgruppe Eingebettete Systemsoftware (Prof. Spinczyk)
- Lehrstuhl Systemarchitektur (Prof. Bellosa)

Betriebssystem-Unterstützung und  
Entwicklungsmethodik für die  
transparente und effiziente Nutzung,  
Konfigurierung und Rekonfigurierung von  
programmierbarer Hardware in  
eingebetteten Systemen



# Projektziele

---

- Abstraktion der Ressource „FPGA“
  - Hardware-Software-Schnittstellen und Kommunikation
  - Hersteller- und Plattform-Unabhängigkeit
- Multiplexing der Ressource „FPGA“
  - Scheduling und Zugriffskontrolle
- Neue Mechanismen zur Konfigurierung von FPGAs
  - Merkmalmodellierung und aspektorientierte Programmierung
- Auslagerung von Betriebssystem-Funktionen in FPGA
  - Implementierung „klassischer“ BS-Probleme im FPGA
  - Ermitteln von Funktionalität, die von FPGA profitiert
  - Konfigurierung der FPGAs durch das eingebettete BS
- Virtualisierung des FPGAs
  - Dynamisch Funktionalität in HW oder SW realisieren
  - Integrierte, HW/SW-übergreifende Scheduling-Komponente



# Bisherige Arbeiten

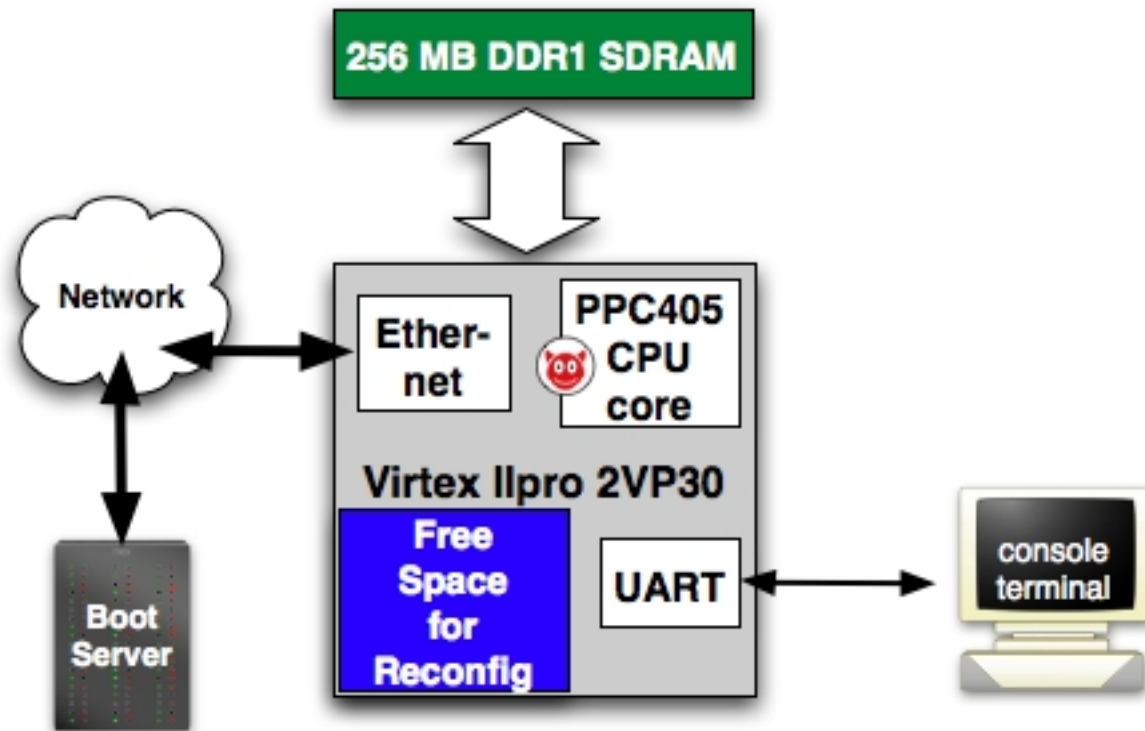
---

- System-on-Chip Integration von Embedded Controllern
  - Integration von „legacy“-Komponenten auf FPGAs
  - Network-on-Chip-Integration von Systemen
  - Verwendung von CAN als NoC-Bussystem
  - Anbindung an physikalischen Bus
  - Qualität von Open Source Hardware-Komponenten
  - IIES-Workshop 2008, Glasgow
- Aspekte in Hardware-Beschreibungssprachen
  - Analyse von querschneidenden Belangen in Systemen, die in HW-Beschreibungssprachen implementiert sind
  - Anwendung aspektorientierter Technologien auf HW-Beschreibungssprachen
  - ACP4IS-Workshop 2008, Brüssel



# Aktuelle Arbeiten

- Hardware-Bootloader
  - Laden von HW-Komponenten via Netzwerk
    - Analog zu Netzwerk-Boot von Betriebssystem
  - Initiale FPGA-Konfiguration
  - Flexible Testumgebung
  - Laden von HW und SW-Komponenten
  - Konfiguration des Kerns anhand der geladenen HW-Komponenten

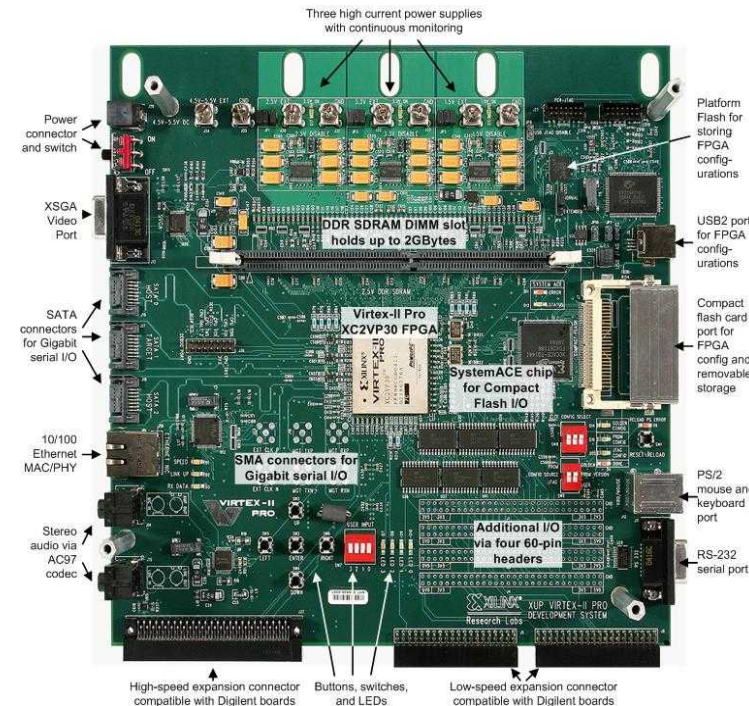






# Zukünftige Arbeiten

- Hardware-/Software-System mit FPGA
  - Xilinx XUP-Board: Virtex II pro FPGA mit 2 PPC 405-Cores
  - Betriebssysteme: CiAO, NetBSD
  - Zugriff auf statisch konfigurierte Hardware-Komponenten
  - Realisierung von Multiplexing und Zugriffskontrolle
- Neue Konfigurierungsansätze
  - Merkmalmmodellierung von BS- und FPGA-Komponenten (z.B. Gerätetreiber und Geräte-implementation)
  - Verwendung aspektorientierter Programmierung über die HW/SW-Grenze
- Dynamische Konfigurierung
  - Dyn. Laden von HW-Komponenten
  - Scheduling der FPGA-Nutzung





# Überblick

---

- FPGAs und Rekonfigurierung
  - Aktuelle Trends
  - Programmierung von FPGAs
  - Dynamische Rekonfigurierung
- FPGAs in eingebetteten Systemen
  - Systemstrukturen
  - Einsatzmöglichkeiten
- UPHOS
  - Bisherige Arbeiten
  - Projektziele
- **Ausblick**



# Ausblick

- Evaluierung von HW-/SW-Systemen mit FPGAs in Hinblick auf Effizienz und Energieverbrauch
- Multicores vs. Rekonfigurierbare Komponenten
  - Programmiermethoden
  - Effizienz für eingebettete Anwendungen
- Vision: Open Source Hardware-/Software-System
  - Soft cores implementieren CPU, Peripherie, Interconnects
  - Open Source-Software für Firmware, Betriebssystem, Anwendungen
  - Hardware eines eingebetteten Open Source-Systems:
    - FPGA großer Kapazität
    - RAM



# Literatur

- [1] P. Lysaght, B. Blodget, J. Mason, J. Young, B. Bridgford:  
*„Enhanced Architectures, Design Methodologies and CAD Tools for Dynamic Reconfiguration of Xilinx FPGAs“*  
16th International Conference on Field Programmable Logic and Applications (FPL'06), Madrid, 2006
- [2] R. Kress, R. Hartenstein, U. Nageldinger:  
*„An Operating System for Custom Computing Machines based on the Xputer Paradigm“*
- [3] P. Knezević, B. Radunović, A. Elhouni:  
*„Issues in FPGA-based Configurable Computer Design for Operating Systems and Multimedia Applications“*
- [4] G. Wigley, D. Kearney:  
*„The First Real Operating System for Reconfigurable Computers“*
- [5] P. Merino, J.C. Lopez, M. Jacome:  
*„A Hardware Operating System for Dynamic Reconfiguration of FPGAs“*
- [6] H. Kwok-Hay So, R. Brodersen:  
*„Improving Usability of FPGA-based Reconfigurable Computers Through Operating System Support“*