Einsatz von eingebetteten FPGA-basierten Mikroprozessorkernen in HW-Agentennetzwerken

Diplomverteidigung

Torsten Schmutzler
s6757418@inf.tu-dresden.de

Technische Universität Dresden
Institut für Technische Informatik

Gliederung

- Aufgabenstellung
- Grundlagen
- Ausgangssituation
- Konzeption
- Implementierung
- Ausblick

Torsten Schmutzler
Technische Universität Dresden
Institut für Technische Informatik
s6757418@inf.tu-dresden.de

Aufgabenstellung

- Literaturstudium
- Untersuchung von Betriebsystemen zum Einsatz in Agentensystemen
- Auswahl eines Microprozessors zur Implementierung auf Virtex2P/5
- Implementierung eines Rekonfigurationsmechanismus.
- Entwicklung der Agentensteuerung und Taskverwaltung
- Integration mit dem Agenten-Management-System „amsys“
- Validierung des Systems mithilfe von Beispielen und Testszenarien
- Auswertung der Ergebnisse

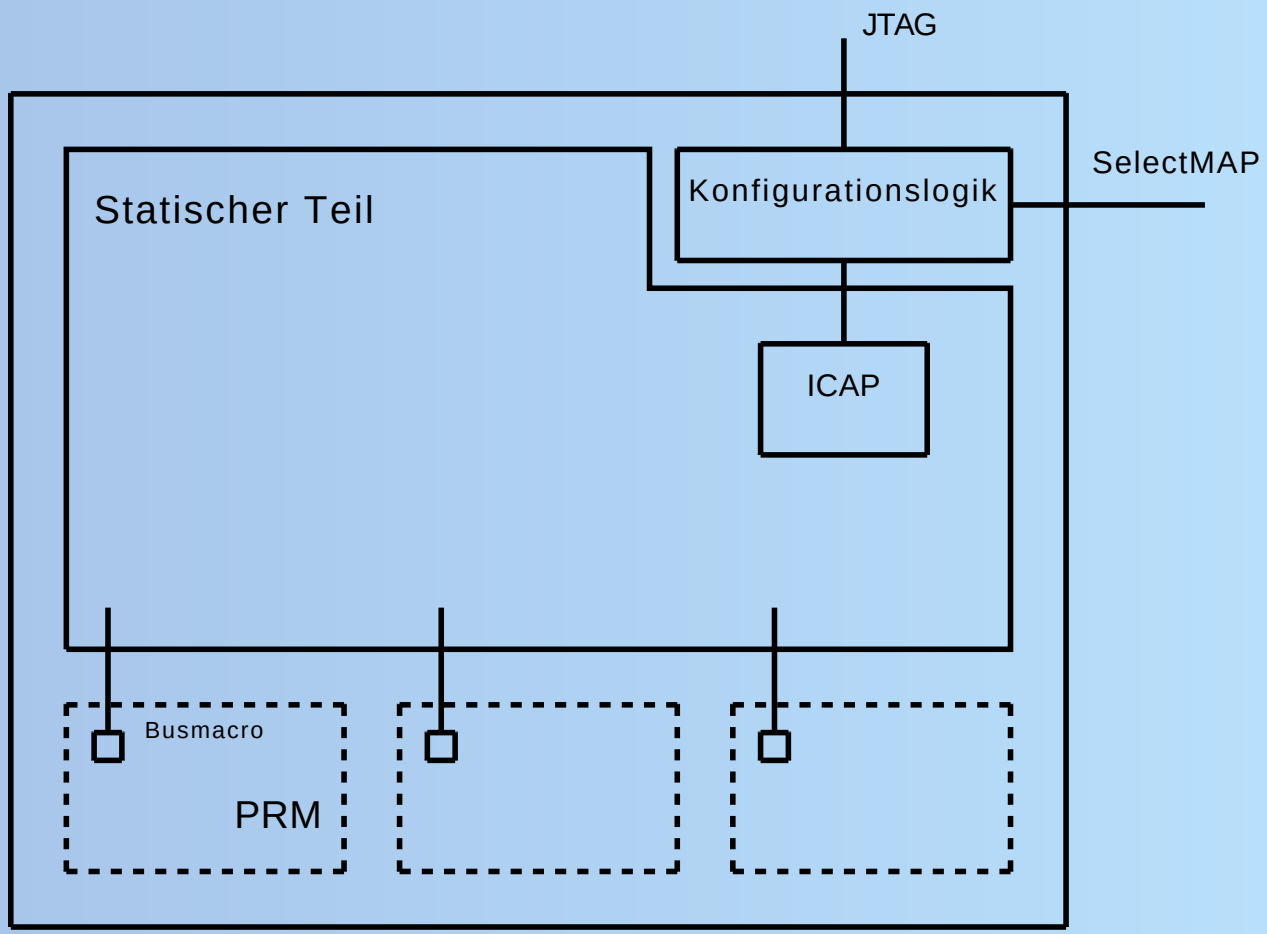
Torsten Schmutzler
Technische Universität Dresden
Institut für Technische Informatik
s6757418@inf.tu-dresden.de

Dynamisch Partielle Rekonfiguration

- Logikstrukturen und Verdrahtung von FPGAs wird durch Speicherzellen definiert
- Veränderung dieser Speicher ist im Betrieb möglich
- Hardwaredesignprozess muss angepasst werden
- Partielle Regionen werden fest auf dem FPGA positioniert
- Alle Kommunikation zwischen partiellen Regionen und statischer Hardware läuft über Busmacros

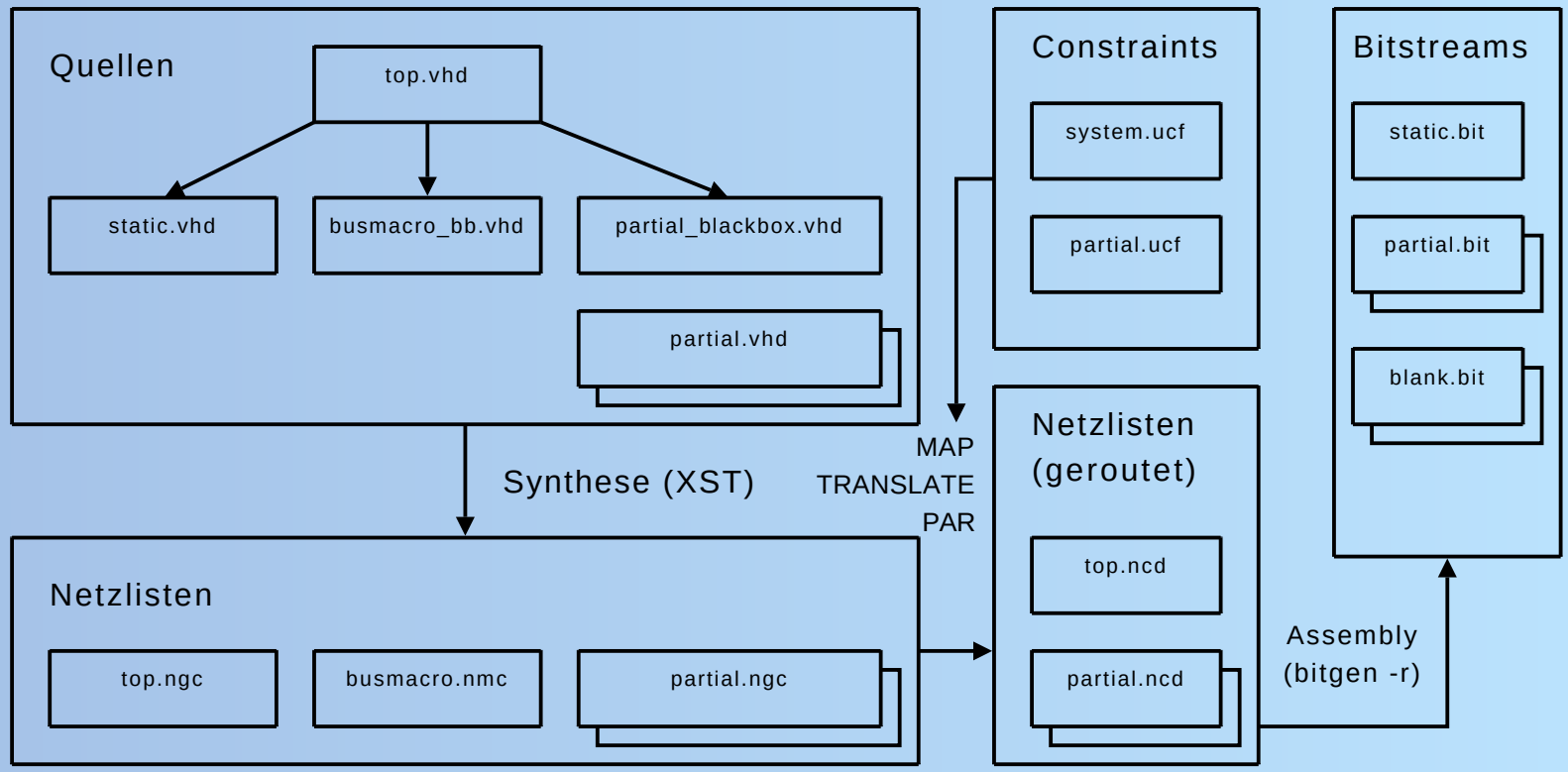
Torsten Schmutzler
Technische Universität Dresden
Institut für Technische Informatik
s6757418@inf.tu-dresden.de

Modulbasierte Partielle Rekonfiguration



Torsten Schmutzler
Technische Universität Dresden
Institut für Technische Informatik
s6757418@inf.tu-dresden.de

Partieller Designflow



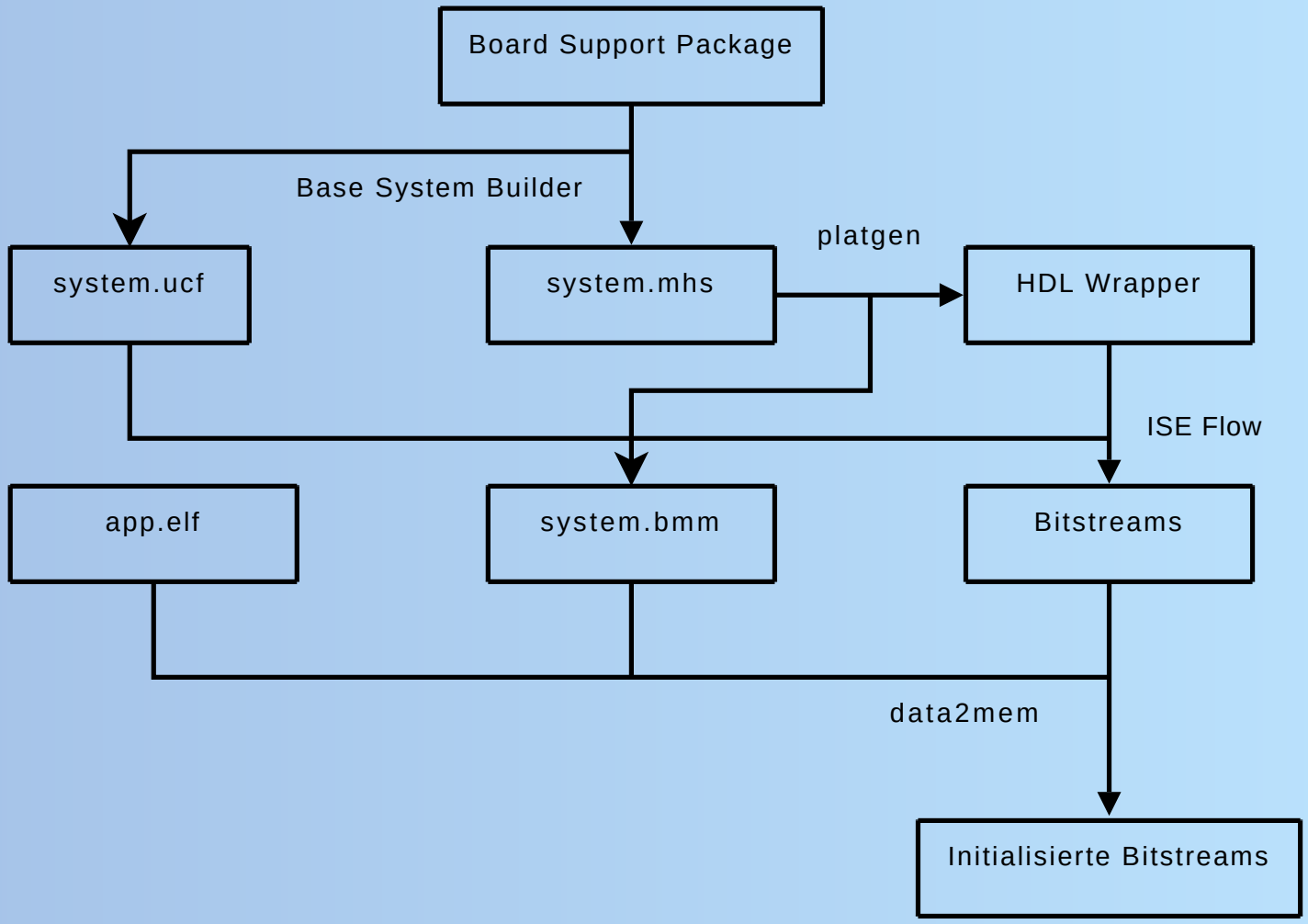
Torsten Schmutzler
Technische Universität Dresden
Institut für Technische Informatik
s6757418@inf.tu-dresden.de

Embedded Development Kit

- Xilinx Tool zur Erstellung von System On Chip Systemen zur Implementierung auf FPGAs
- Microprozessoren, Peripherie und Bussystem stehen als IP Cores zur verfügung
- Softwaretreiber zur Steuerung der Peripherie ist enthalten
- Ein Minimalkernel „xilkernel“ kann optional eingesetzt werden
- Compiler Toolchain zur Entwicklung von stand-alone Applikationen
- Die Addressen der Peripherie kann exportiert werden um Third Party Betriebsysteme einzusetzen

Torsten Schmutzler
Technische Universität Dresden
Institut für Technische Informatik
s6757418@inf.tu-dresden.de

EDK Design Flow



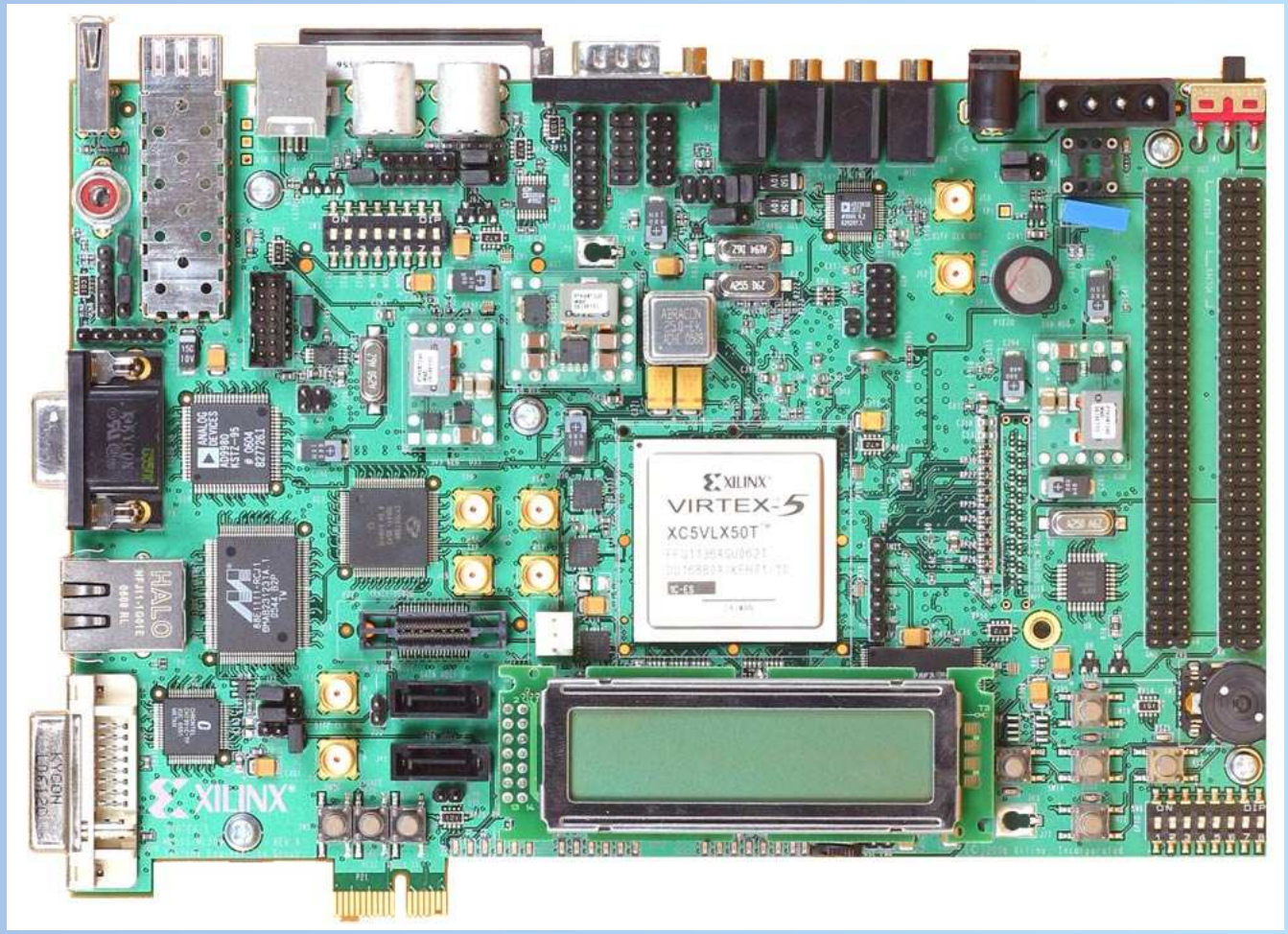
Torsten Schmutzler
Technische Universität Dresden
Institut für Technische Informatik
s6757418@inf.tu-dresden.de

Betriebsysteme

- Montavista Linux RTOS
- Bluecat Linux
- Petalinux
  - uCLinux-Distribution für Microblaze (PowerPC ohne MMU zur Implementierung auf Xilinx FPGAs)
  - Microblaze gcc - Toolchain
  - Linux 2.4 und 2.6
  - Umfangreiche Sammlung von Applikationen (busybox, uClib)
  - KBuild System, Support Skripte

Torsten Schmutzler
Technische Universität Dresden
Institut für Technische Informatik
s6757418@inf.tu-dresden.de

Hardware - Xilinx ML505 Entwicklungsboard



Torsten Schmutzler
Technische Universität Dresden
Institut für Technische Informatik
s6757418@inf.tu-dresden.de

Ausgangssituation

Xilinx ML505

- Virtex5 FPGA (ICAP_VIRTEX5)
- SPI Flash
- Ethernet PHY
- SystemACE
- RS232

Torsten Schmutzler
Technische Universität Dresden
Institut für Technische Informatik
s6757418@inf.tu-dresden.de

Ausgangssituation

Bestehende Projekte

- Java Server „amsys“ zur Steuerung von Atmega32 basierten HW-Agenten
- GUI und Datenbank Anbindung
- Agenten Hardware (ahw) Modulverbindungssystem (Crossbar, gemeinsammer Speicher, IO) von Marcel Naggaz

Ausgangssituation

Entwicklungsumgebung

- ISE + EDK 10.1 (Systemsynthese)
- ISE 9.1 + Partial Reconfiguration Early Access + PlanAhead (Partieller DesignFlow)
- Petalinux v0.30-rc1 / subversion
- Eclipse

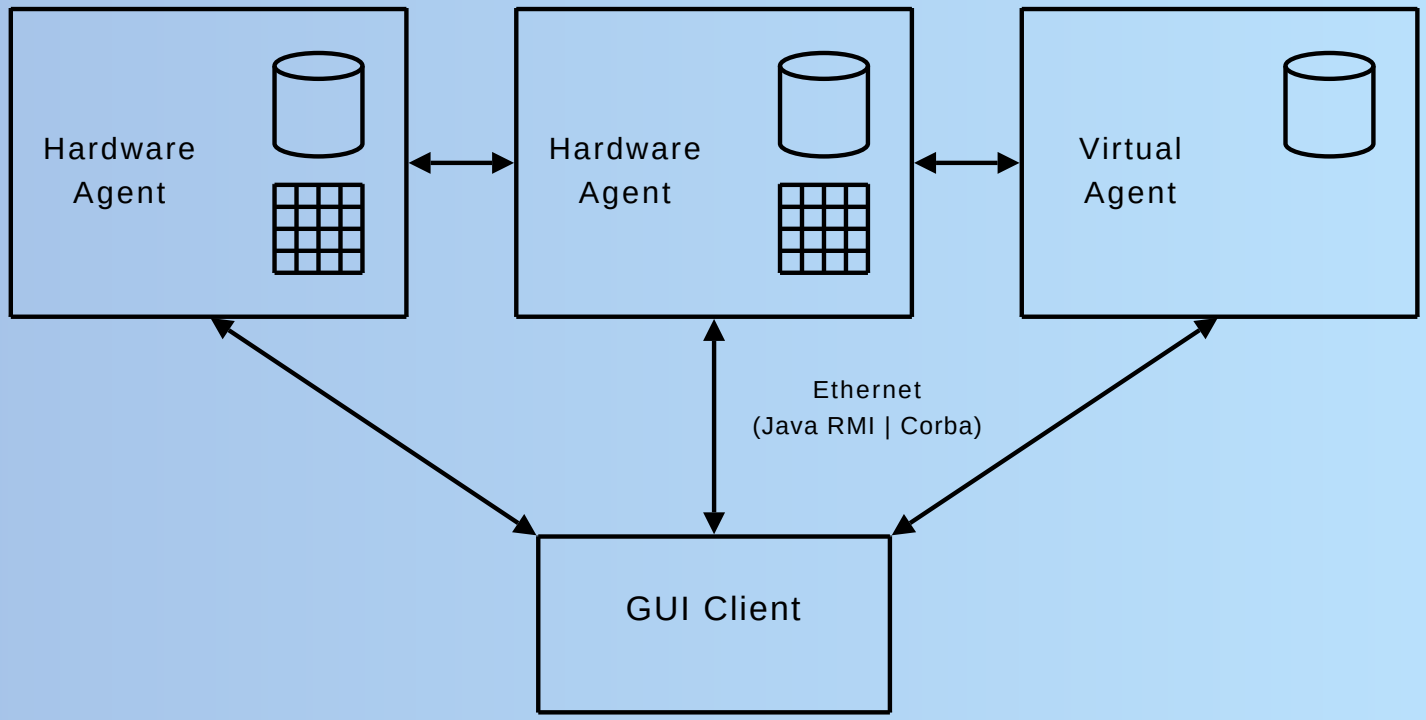
Torsten Schmutzler
Technische Universität Dresden
Institut für Technische Informatik
s6757418@inf.tu-dresden.de

Konzeption

- Microblaze basiertes Embedded System auf dem ML505 Virtex5 Board
- Petalinux Betriebsystem
- Partielle Rekonfiguration über HWICAP
- Anbindung an das AHW Kommunikationssystem zur Steuerung der partiellen Module (Linux Treiber)
- Managementsystem „amsys“ als Javaimplementation
- Getestete JVMs: jamvm, kaffe, mika, javaee

Torsten Schmutzler
Technische Universität Dresden
Institut für Technische Informatik
s6757418@inf.tu-dresden.de

Verteilte System Struktur



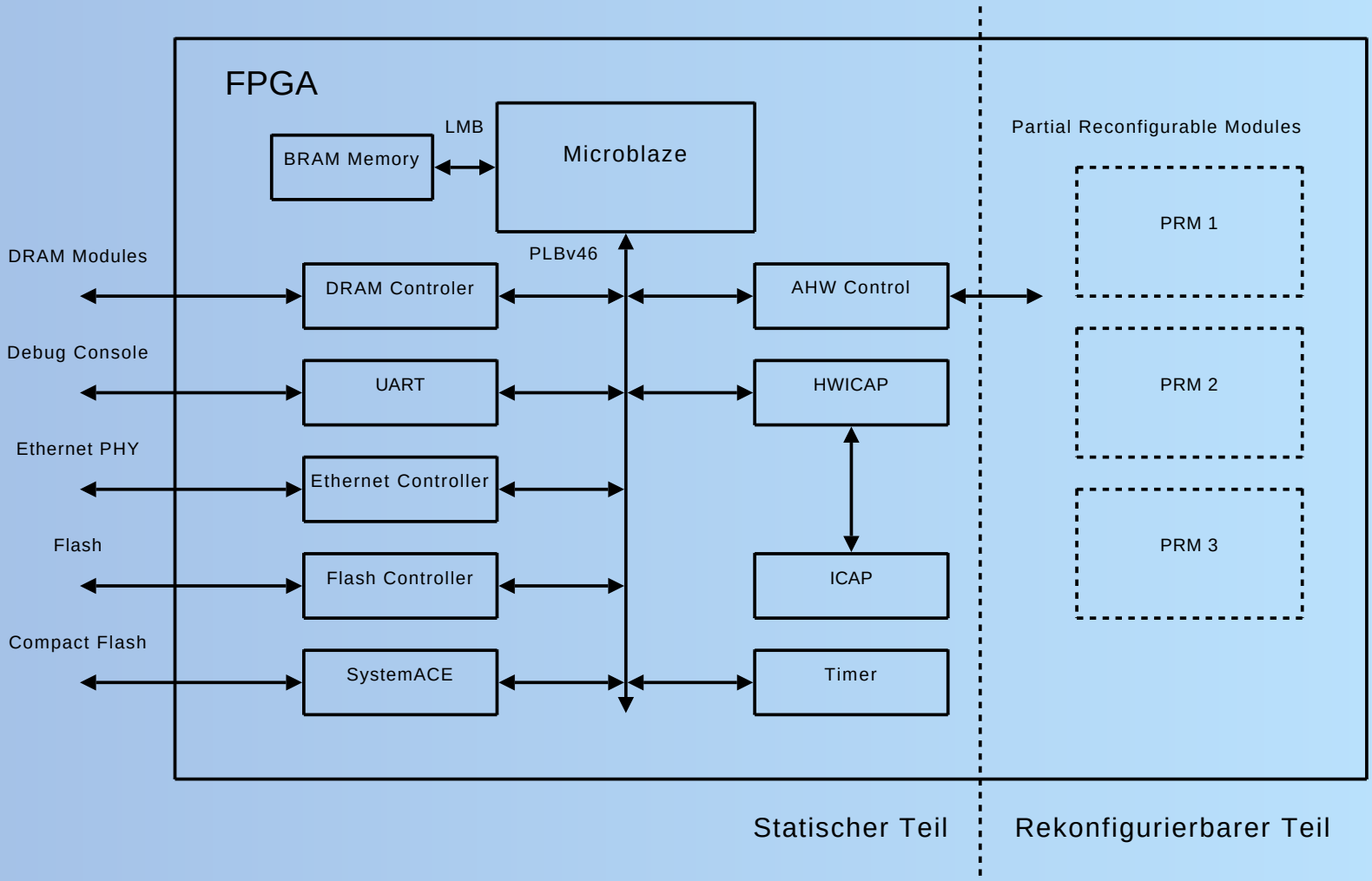
Torsten Schmutzler
Technische Universität Dresden
Institut für Technische Informatik
s6757418@inf.tu-dresden.de

Bootprozess

- Firststage Bootloader im BRAM des Microblaze
  - Speicherinitialisierung
  - Laden des Secondstage Bootloaders vom FLASH
- uboot
  - Kernel und RomFS Updates im FLASH über DHCP/TFTP
  - Laden des Linux Kernels
- Linux
- Root Device: FLASH (RomFS) oder SystemACE/CF (ext2)
- Persistent Storage: FLASH (JFFS2)

Torsten Schmutzler
Technische Universität Dresden
Institut für Technische Informatik
s6757418@inf.tu-dresden.de

HW-Agenten Struktur



Torsten Schmutzler
Technische Universität Dresden
Institut für Technische Informatik
s6757418@inf.tu-dresden.de

Umsetzung der Agenteneigenschaften

- autonomes Arbeiten: stärkere Entkopplung vom Supervisor (dem Managementserver)
- soziales Verhalten: Agentenkommunikation, Taskverteilung
- Weitere Eigenschaften: implementiert in den Partiellen Modulen

Torsten Schmutzler
Technische Universität Dresden
Institut für Technische Informatik
s6757418@inf.tu-dresden.de

Implementierung der Taskverwaltung

- C Implementation von amsys-core
- Tasks sind Prozesse und werden mit libamsys gelinkt
- Zu jedem Task gehört ein Verzeichnisbaum, welcher das binary, die Hardwaremodule und Nutzerdaten enthält
- Tasks konfigurieren die Hardware selbst müssen die Ressourcen aber beim Managementsystem anfordern
- Jeder Task oder „amsys“ kann andere Agenten kontaktieren um z.B. dessen Ressourcen zu nutzen
- Neustart von Task nicht kritischen Fehlersituationen (z.B. belegte Ressourcen)
- Ereignisse (Task beendet, Fehler) können mithilfe der Nutzerschnittstelle abgefragt werden

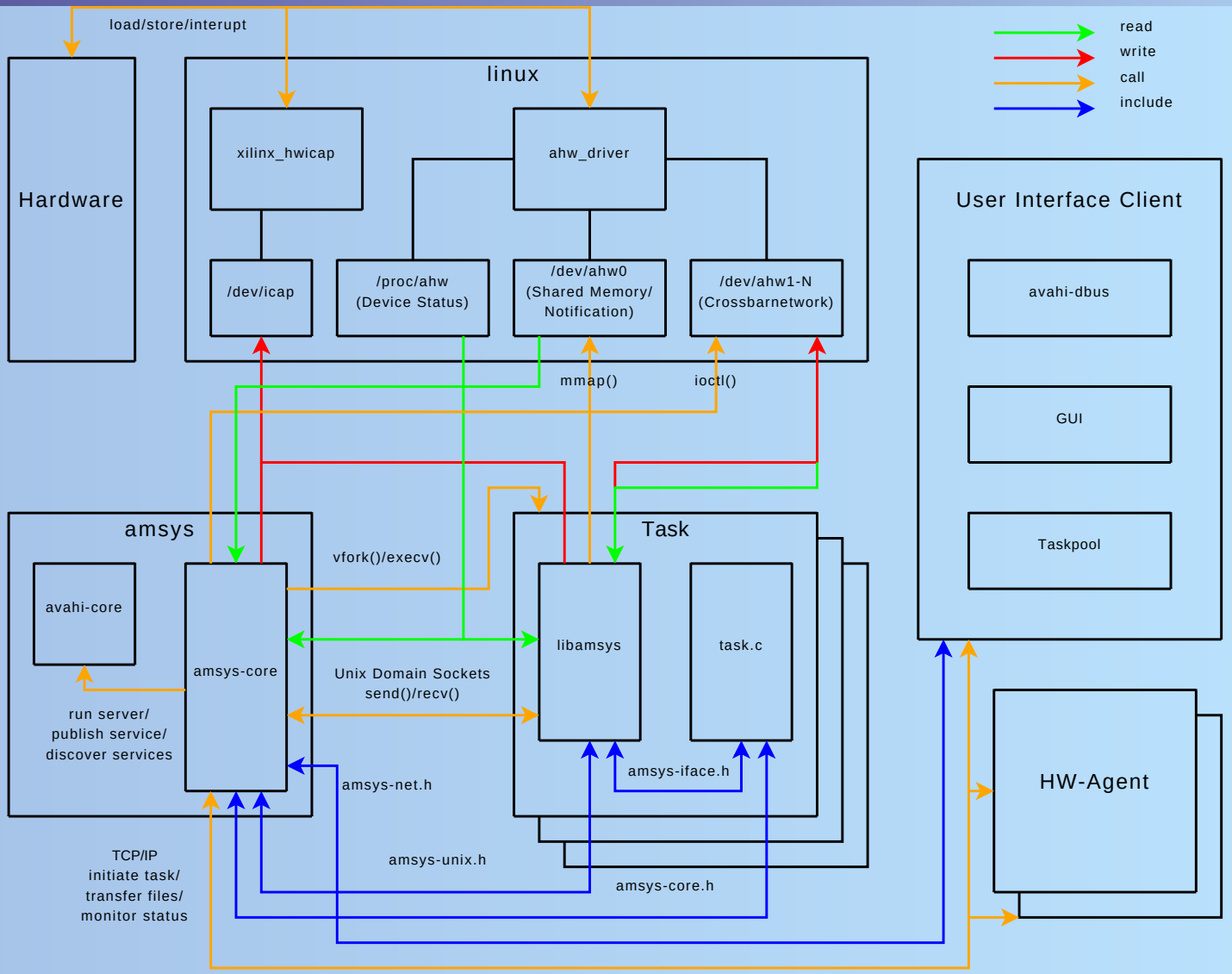
Torsten Schmutzler
Technische Universität Dresden
Institut für Technische Informatik
s6757418@inf.tu-dresden.de

Anbindung an die Agenten HW Kommunikation

- Implementierung einer Hardwarekontrolleinheit zur Verbindung des AHW Systems an den IPIC Bus
- IPIC - PLB Brücke
- Linux Treiber zur Steuerung der Hardware

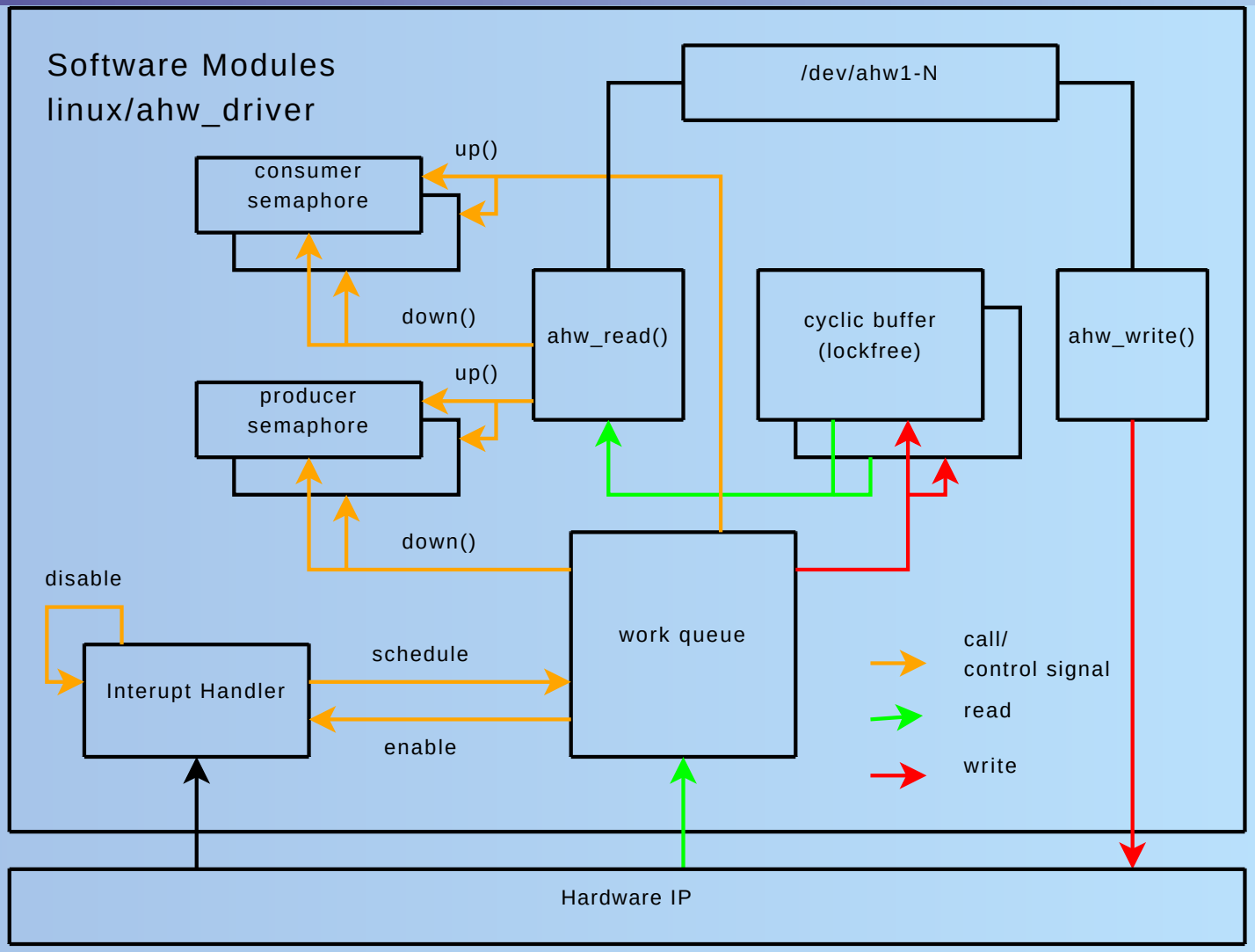
Torsten Schmutzler
Technische Universität Dresden
Institut für Technische Informatik
s6757418@inf.tu-dresden.de

Implementierung



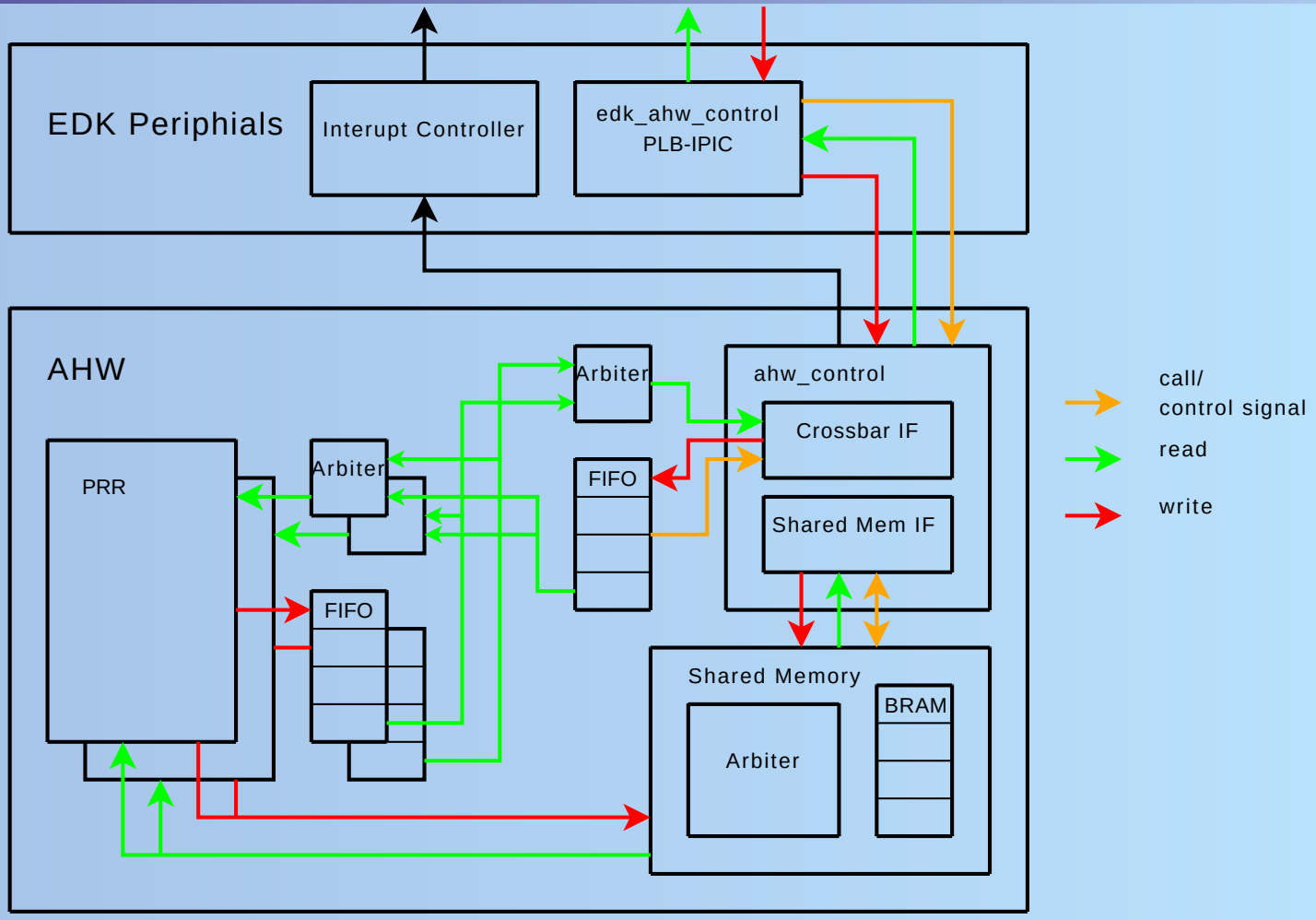
Torsten Schmutzler
Technische Universität Dresden
Institut für Technische Informatik
s6757418@inf.tu-dresden.de

Implementierung



Torsten Schmutzler
Technische Universität Dresden
Institut für Technische Informatik
s6757418@inf.tu-dresden.de

Implementierung



Torsten Schmutzler
Technische Universität Dresden
Institut für Technische Informatik
s6757418@inf.tu-dresden.de

Kommunikationsprotokoll

- Geplant: Java RMI, Corba oder ICE
- Binäres Paketorientiertes Protokoll (Magictoken, Pakettyp, Daten)
- Synchrone Abarbeitung auf der Hardware über Unix Domain Sockets
- Asynchrone Kommunikation zwischen den Agenten und der Nutzerschnittstelle über Internet Sockets
- Operationen: Regionen reservieren, Daten übertragen, Tasks starten

Torsten Schmutzler
Technische Universität Dresden
Institut für Technische Informatik
s6757418@inf.tu-dresden.de

Service Discovery

- Agenten müssen sich gegenseitig im Netz finden
- Dies soll ohne spezielle Konfiguration möglich sein: Zeroconf
- Als Zeroconf Implementation wurde „avahi“ eingesetzt und auf Microblaze portiert
- Die Nutzerschnittstelle greift auf „avahi“ über DBus zu mithilfe von „java-dbus“
- Jeder Agent veröffentlicht seinen Diest als „_amsysagent._tcp.local“
- Über Multicast DNS und DNS Service Discovery werden diese Informationen abgerufen

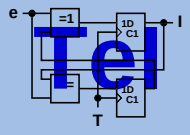
Torsten Schmutzler
Technische Universität Dresden
Institut für Technische Informatik
s6757418@inf.tu-dresden.de

Ausblick

- Verzicht auf die Softcore Peripherie um mehr Fehler behandeln zu können
  - FPGAs mit integriertem SoC
  - SoC mit FPGA Ressourcen
  - SoC und FPGA komplett trennen
- Einsatzes eines Microkontrollers mit MMU (für z.B. JVM)
- Entwicklung von Entwurfsprozessen, welche partielle Module effizienter erzeugen kann
- Entwicklung einer Hardwarebibliothek von partiellen Modulen

Torsten Schmutzler
Technische Universität Dresden
Institut für Technische Informatik
s6757418@inf.tu-dresden.de

Vielen Dank für Ihre Aufmerksamkeit!



Torsten Schmutzler
Technische Universität Dresden
Institut für Technische Informatik
s6757418@inf.tu-dresden.de