



**TECHNISCHE
UNIVERSITÄT
DRESDEN**

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

GENERATORBASIERTE BEREITSTELLUNG VON HW-AGENTENARCHITEKTUREN

Diplomverteidigung

Michael Rudolph

s1890020@inf.tu-dresden.de

Dresden, 10.6.2009

Inhalt

Einführung

- Motivation

- Agentensysteme

- Dynamisch partielle Rekonfiguration

- Arbeiten

Generatorwerkzeug

- Entwurfsfluss

- Agentengenerator „agentgen“

Ergebnisse

- Architekturerstellung

- Resourceverbrauch

- Vorteile eines Generatorwerkzeuges

Zusammenfassung und Ausblick

01 Motivation

Tendenzen in der Mikroelektronik:

- verstärkter Einsatz von Mikroelektronik in technischen Systemen
- steigende Komplexität
- hohes Maß an Zuverlässigkeit gefordert

Folgerungen:

- Organic Computing
- Implementierung dienstbringender Systeme als Hardwareagentensysteme

Ziel der Arbeit:

- nutzerfreundliches Werkzeug zur Erzeugung prozessorbasierter Hardwareagentenarchitekturen

01 Organic Computing

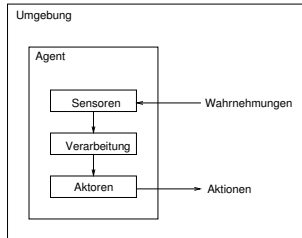
Organic Computing ist ein Wissenschaftszweig, der sich damit beschäftigt, Eigenschaften lebender Systeme auf technische Systeme zu übertragen.

Self-X Eigenschaften:

- selbstoptimierend
- selbstkonfigurierend
- selbstheilend
- selbstschützend

01 Agentensysteme

Ein **Agent** ist ein System, das die Umgebung durch Sensoren beobachtet und in der Umgebung Aktionen durch Aktoren ausführt.



Ein **Multiagentensystem** ist ein System, das aus mehreren Agenten besteht, die versuchen, entweder eine gemeinsame Aufgabe oder verschiedene Aufgaben zu lösen.

01 Eigenschaften und Beispiele

Eigenschaften von Agentensystemen:

- **Autonomie:** Agent arbeitet selbstständig und ist von Benutzereingriffen weitgehend unabhängig
- **Reaktivität:** Agent beobachtet seine Umgebung und trifft Entscheidungen anhand dieser
- **Interaktivität:** Agent kommuniziert mit anderen dienstbringenden Systemen
- **Proaktivität:** Aktionen auf Grund eigener Initiativen ausgelöst

Beispiele für Hardwareagentensysteme:

- Mars-Sonde Pathfinder
- Autopiloten in Flugzeugen
- Fahrassistenzsysteme in Autos

01 Dynamisch partielle Rekonfiguration

- Funktion eines FPGAs durch Inhalt von SRAM-Zellen bestimmt
- SRAM-Zellen veränderbar (JTAG, SelectMAP, ICAP)
- Veränderung kann auch zur Laufzeit erfolgen

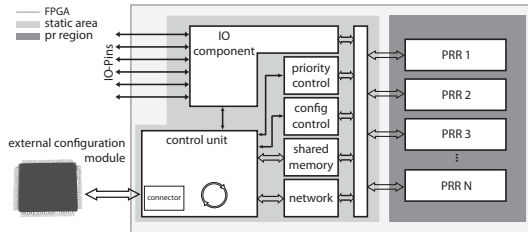
Begriffe:

- partiell rekonfigurierbare Region
- Modul
- statische Logik
- Busmakro
- partieller Bitstream
- vollständiger Bitstream

Vorteile:

- dynamische Anpassung an die Aufgabenstellung
- Fehlerkorrektur

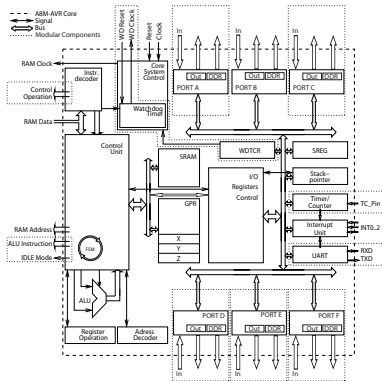
01 Architektur eines Hardwareagenten



- Module zur Laufzeit austauschbar
- Verbindungen von Modulen zu externe E/A-Pins zur Laufzeit austauschbar

01 A8M

Architektureigenschaften



- 8 Bit RISC-Core
- Befehlssatz kompatibel zum AVR
- modular
- 6 Ein- und Ausgabeports
- UART
- 8 Bit Timer/Counter
- 8 Bit ALU
- 32 Arbeitsregister
- Watchdog-Timer
- Signale zur Fehlerüberwachung

01 A8M

Komponenten zur Selbstüberwachung

- Interrupt-Check
 - in regelmäßigen Abstand Interrupt zum A8M
 - Abarbeitung eines Testprogramms
 - Vergleich mit erwarteten Werten
- Module-Observer
 - Speicherung von Steuersignale in einem Schieberegister
 - Überprüfung der letzten vier Befehlszählerstände
 - Fehlersignal bei gleichen Befehlszählerständen
- Watchdog-Observer
 - Watchdog-Timer des A8M regelmäßig zurücksetzen sonst Watchdog-Reset
 - zählt Anzahl der Watchdog-Resets
 - Fehlersignal bei mehrmaligen Auftreten von Wachdog-Resets

01 Eingebettete Mikroprozessorkerne

Aufgaben:

- Kommunikation mit anderen Systemen (Ethernet)
- Ansteuerung des Konfigurationsport ICAP

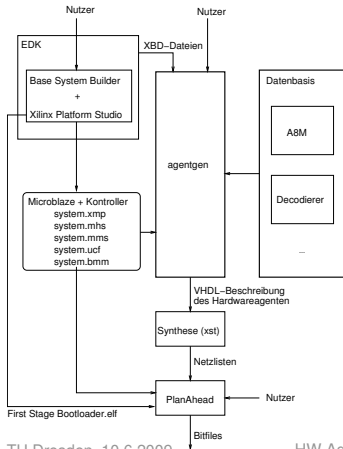
Umsetzung:

- Microblaze
- Petalinux

Einsatz des „Embedded Development Kits“ und Petalinux:

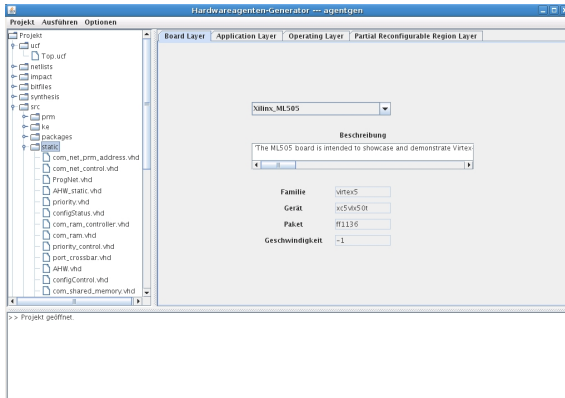
- vorhanden IP-Cores und Treiber
- geringerer Entwicklungsaufwand

02 Entwurfsfluss



- ISE 10.1
- EDK 10.1
- PlanAhead 10.1
- ISE 9.2 mit Patch für partielle Rekonfiguration

02 Werkzeug agentgen



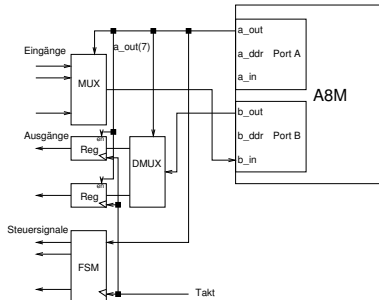
- Windows und Linux
- Setzen der korrekten Umgebungsvariablen
- Syntheseskripte

- Editor
- Textfeld
- Ordnerstruktur für partiell rekonfigurierbare Projekte

02 Entwurfsunterstützung von agentgen

- einfache Auswahl des Boards
 - Auslesen der „Xilinx Board Description“-Datei
 - Erzeugung der Top-Level-VHDL-Beschreibung
 - automatische Erzeugung der „User Constraint“-Datei
- einfache Auswahl von Pins zur Ansteuerung externer Peripherie
- einfache Integration eines Microblaze und IP-Cores
 - erzeugt mit Hilfe des EDK
 - Auslesen der „Microprocessor Hardware Description“-Datei
- einfache Einstellung von Konstanten
 - Anzahl partiell rekonfigurierbarer Region
 - Speichergröße
- Einfügen von Busmakros
 - automatische Auswahl je nach FPGA
 - automatisches Einfügen zwischen statischer und dynamisch rekonfigurierbarer Logik

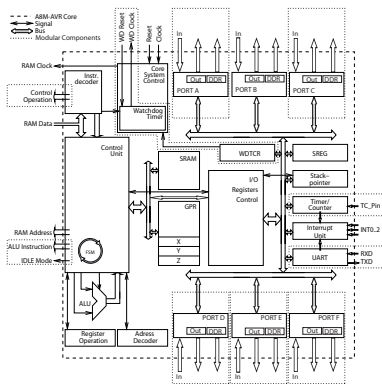
02 Integration eines A8M



- Ansteuerung der Prioritätenverwaltung
- Ansteuerung des Verbindungsnetzwerk
- Ansteuerung des gemeinsamer Speicher
- Ansteuerung der E/A-Komponente
- Programmierung des Programmspeichers eines anderen A8M
- Programmierung des Vergleichsspeichers der IC-Komponente eines anderen A8M
- C-Bibliothek

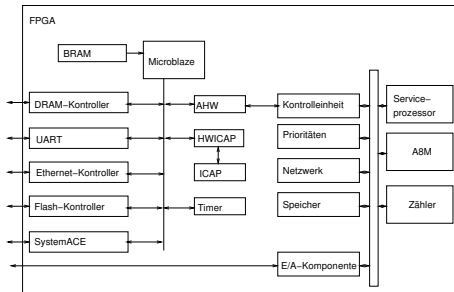
02 Entwurfsunterstützung von agentgen

A8M



- Programm im Intel-Hex-Format
- Auswahl der Taktfrequenzen
- einfache Zusammenstellung der Komponenten
 - Watchdog-Observer
 - Module-Observer
 - Interrupt-Check
 - Ports, UART, Timer
- automatisches Verbinden

03 Architektur



- Serviceprozessor
 - Watchdog-Observer
- A8M
 - Watchdog-Observer
 - Port C
- Zähler
- Microblaze
 - DRAM
 - UART
 - Ethernet
 - Flash
 - SystemACE
 - HWICAP
 - Timer

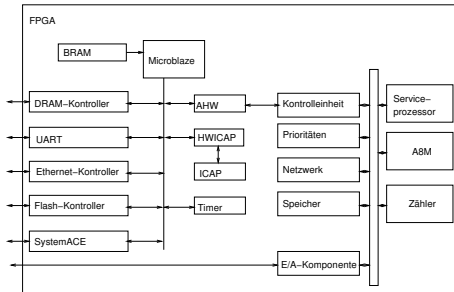
03 Ressourcenverbrauch der statischen Logik

Resource	Bedarf
LUT	14720
FF	13382
SLICEL	3080
SLICEM	1120
BSCAN	1
BUFGCTRL	16
BUFIO	8
DSP48E	6
ICAP	1
IDELAYCTRL	5
ILOGIC	72
IODELAY	91
OLOGIC	147
PLL_ADV	2
RAMBFIFO36	37
TEMAC	1
max. Taktfrequenz	133,050 MHz

03 Ressourcenverbrauch eines A8M

Resource	Bedarf - A8M minimal	Bedarf - A8M vollständig
LUT	806	1289
FF	219	614
SLICEL	185	295
SLICEM	62	99
DSP48E	1	1
RAMBFIFO36	2	1
max. Taktfrequenz	75,460 MHz	73,697 MHz

03 Vorteile eines Generatorwerkzeuges



- 191 VHDL-Dateien
- ca. 26000 Zeilen VHDL-Code
- 31 VHDL-Dateien zu bearbeiten
- ca. 10000 Zeilen VHDL-Code

Vorteile eines Werkzeuges:

- schneller als der Nutzer
- zuverlässiger

04 Zusammenfassung und Ausblick

- Literaturstudium
 - Agentensysteme
 - partielle Rekonfiguration
- Entwurf und Implementierung eines Generatorwerkzeuges
 - Microblaze
 - A8M
- Validierung
- Implementierung auf der „ML505 Evaluation Platform“
- Auswertung der Ergebnisse

- Hardware-Bibliothek

Literatur



Stuart Russel und Peter Norvig.
Artificial Intelligence. A modern Approach.
Prentice Hall, 1991.



Jacques Ferber.
Multiagentensysteme. Eine Einführung in die verteilte künstliche Intelligenz.
Addison-Wesley, 2001.



Marcel Naggatz.
Untersuchung von HW-Architekturkonzepten für Agentensysteme
Studienarbeit, TU Dresden, 2007



Marcel Naggatz.
Dynamische Architekturadaption von Hardware-Agentensystemen.
Diplomarbeit, TU Dresden, 2008



Torsten Schmutzler.
Einsatz von eingebetteten FPGA-basierten Mikroprozessorkernen in HW-Agentennetzwerken.
Diplomarbeit, TU Dresden, 2008



Xilinx, Inc.
<http://www.xilinx.com>.