Untersuchung von HW-Architekturkonzepten für Agentensysteme

Großer Beleg

Marcel Naggatz
marcelnaggatz@hotmail.de

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

Inhalt

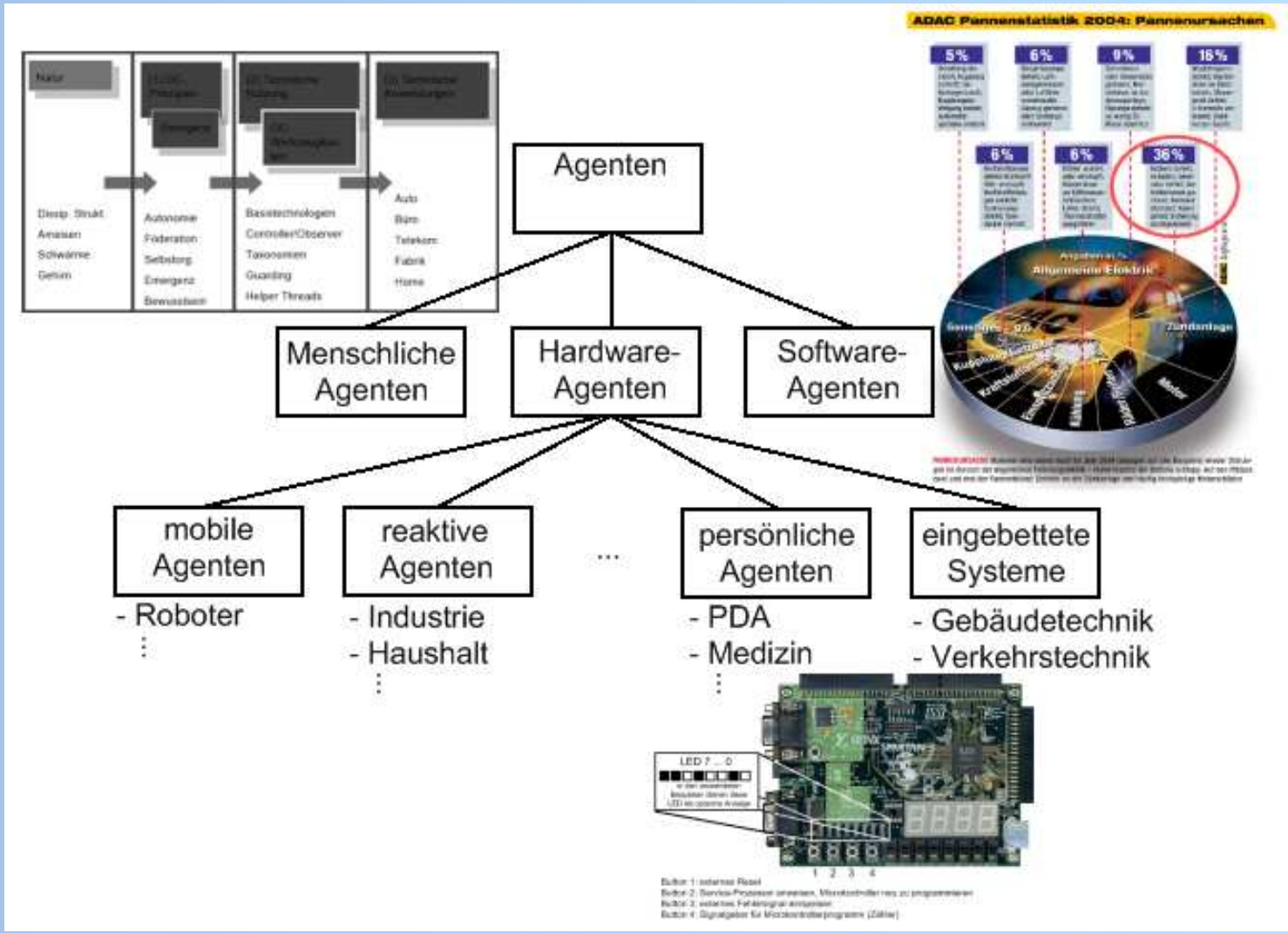
Marcel Naggatz
Technische Universität Dresden
Institut für Technische Informatik
Professur für VLSI-Entwurfssysteme, Diagnostik und Architektur
marcelnaggatz@hotmail.de

Inhalt

1 Einleitung

Marcel Naggatz
Technische Universität Dresden
Institut für Technische Informatik
Professur für VLSI-Entwurfssysteme, Diagnostik und Architektur
marcelnaggatz@hotmail.de

Einleitung



Marcel Naggatz
Technische Universität Dresden
Institut für Technische Informatik
Professur für VLSI-Entwurfssysteme, Diagnostik und Architektur
marcelnaggatz@hotmail.de

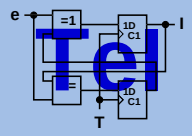
Einleitung

- Anforderungen an zukünftige diensterbringende Systeme sind Flexibilität, Anpassungsfähigkeit und Ausfallsicherheit
- eine universelle Hardwareplattform für dir Umsetzung zukünftiger Informationssysteme
- bekannte Technik aus Softwaretechnologie ist die Agententechnologie (Autonomie, Proaktivität, Reaktivität, Interaktivität, Intelligenz)
- diese Eigenschaften sollen als Architekturkonzepte in Hardware umgesetzt werden
- Aspekte der Agententechnologie werden mit Visionen des Organic Computing ergänzt - Self-X Eigenschaften (selbst-heilend, selbst-konfigurierend, selbst-optimierend, selbst-erklärend, selbst-schützend)

Marcel Naggatz
Technische Universität Dresden
Institut für Technische Informatik
Professur für VLSI-Entwurfssysteme, Diagnostik und Architektur
marcelnaggatz@hotmail.de

Inhalt

2 Agentenarchitekturen



Marcel Naggatz
Technische Universität Dresden
Institut für Technische Informatik
Professur für VLSI-Entwurfssysteme, Diagnostik und Architektur
marcelnaggatz@hotmail.de

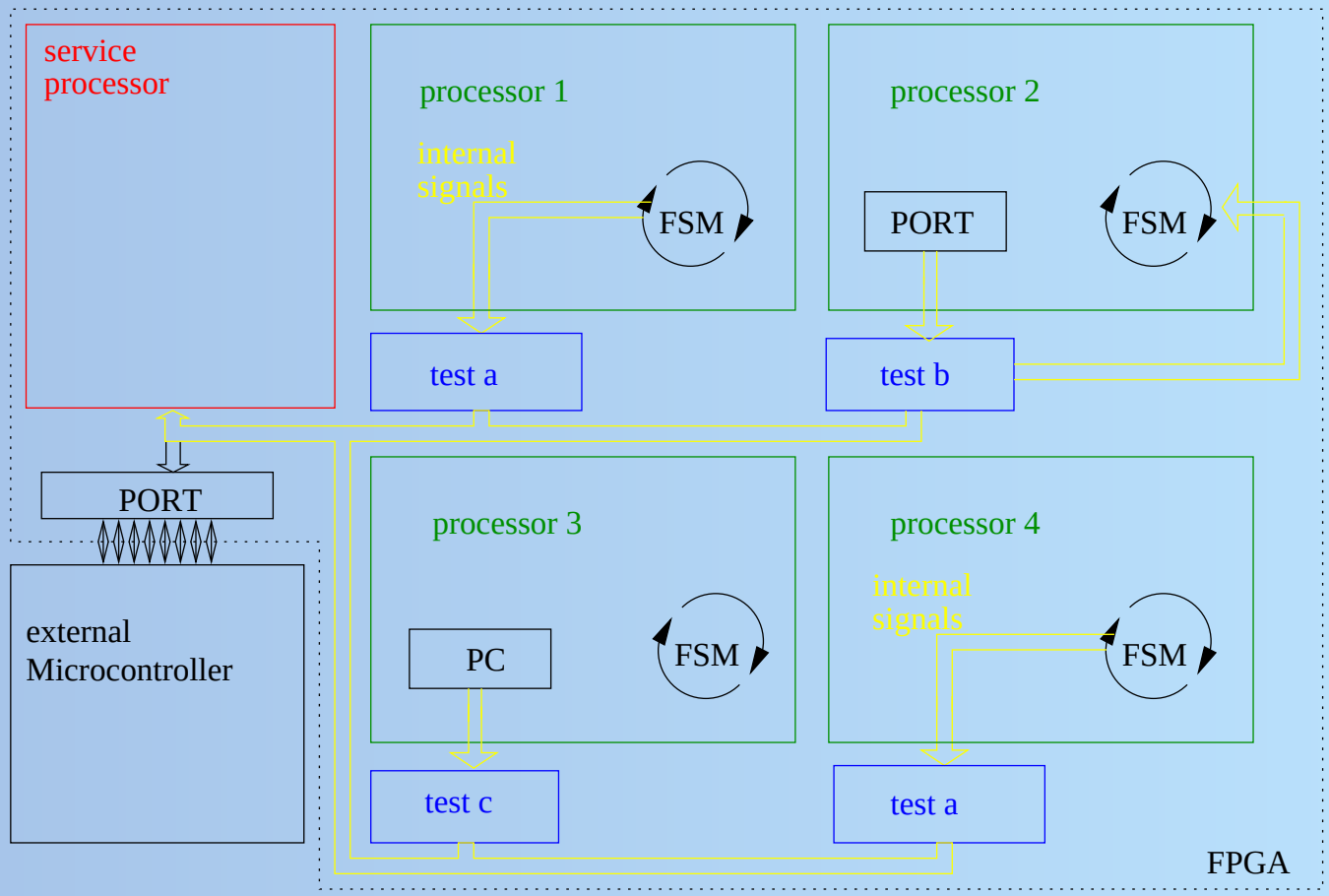
Konzepte von Agentenarchitekturen

Eigenschaften der Konzeption:

- modulare Architektur basierend auf rekonfigurierbarer Logik
- im Vordergrund stehen Flexibilität, Anpassungsfähigkeit und Ausfallsicherheit
- Parametrisierbarkeit und universelle Einsetzbarkeit der Module
  - modularer Microprozessorkern
  - variierbare I/O-Komponenten und Steuersignale
  - Test- und Überwachungskomponenten
  - Serviceprozessoren

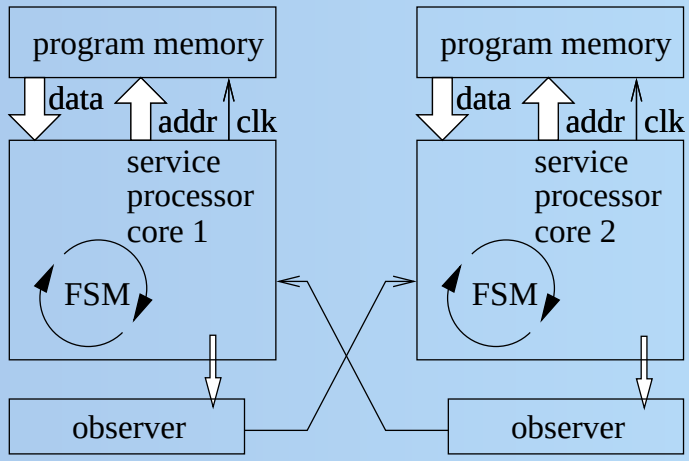
Marcel Naggatz
Technische Universität Dresden
Institut für Technische Informatik
Professur für VLSI-Entwurfssysteme, Diagnostik und Architektur
marcelnaggatz@hotmail.de

Konzepte von Agentenarchitekturen



Marcel Naggatz
Technische Universität Dresden
Institut für Technische Informatik
Professur für VLSI-Entwurfssysteme, Diagnostik und Architektur
marcelnaggatz@hotmail.de

Serviceprozessoren



- eigenständige, vom Hauptsystem getrennte Hardwarearchitektur
- enge Zusammenarbeit mit den Test- und Überwachungskomponenten
- Erhöhung der Ausfallsicherheit und Steuerung der Hardware
- System kann um weitere Serviceprozessoren ergänzt werden

Marcel Naggatz
Technische Universität Dresden
Institut für Technische Informatik
Professur für VLSI-Entwurfssysteme, Diagnostik und Architektur
marcelnaggatz@hotmail.de

Inhalt

3 Entwurf und Implementation

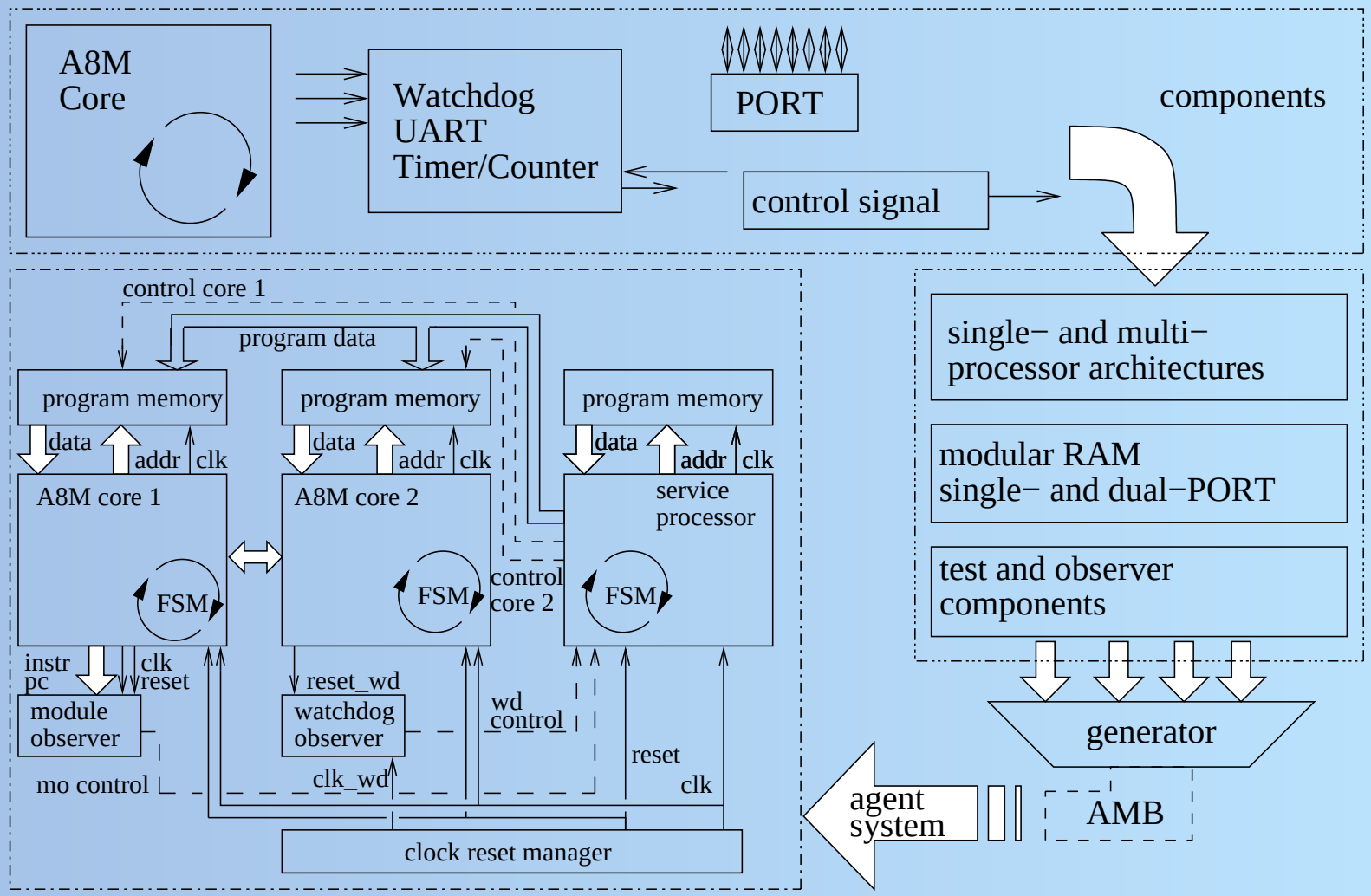
Marcel Naggatz
Technische Universität Dresden
Institut für Technische Informatik
Professur für VLSI-Entwurfssysteme, Diagnostik und Architektur
marcelnaggatz@hotmail.de

Entwurf und Implementierung

- Entwicklung und Konzeption eines Basisprozessors und Test- und Überwachungskomponenten
- Entwurf eines 8-Bit Basisprozessors (A8M) als Referenzentwurf
  - in mehreren Merkmalen modularisierbar und parametrisierbar
  - Modifikation zum Service Prozessor
- Entwurf von drei Test- und Überwachungskomponenten
- Entwurf eines Generatorwerkzeuges (AMB)
  - A8M parametrieren und modularisieren
  - Single- und Multi-Kern Entwurf
  - Auswahl und Anbindung der Testkomponenten
- Test des Agentensystems mit unterschiedlichen Konfigurationen und Testprogrammen

Marcel Naggatz
Technische Universität Dresden
Institut für Technische Informatik
Professur für VLSI-Entwurfssysteme, Diagnostik und Architektur
marcelnaggatz@hotmail.de

Agent Model Builder - AMB



Marcel Naggatz
Technische Universität Dresden
Institut für Technische Informatik
Professur für VLSI-Entwurfssysteme, Diagnostik und Architektur
marcelnaggatz@hotmail.de

Agent Model Builder - AMB

VHDL-Erweiterung:

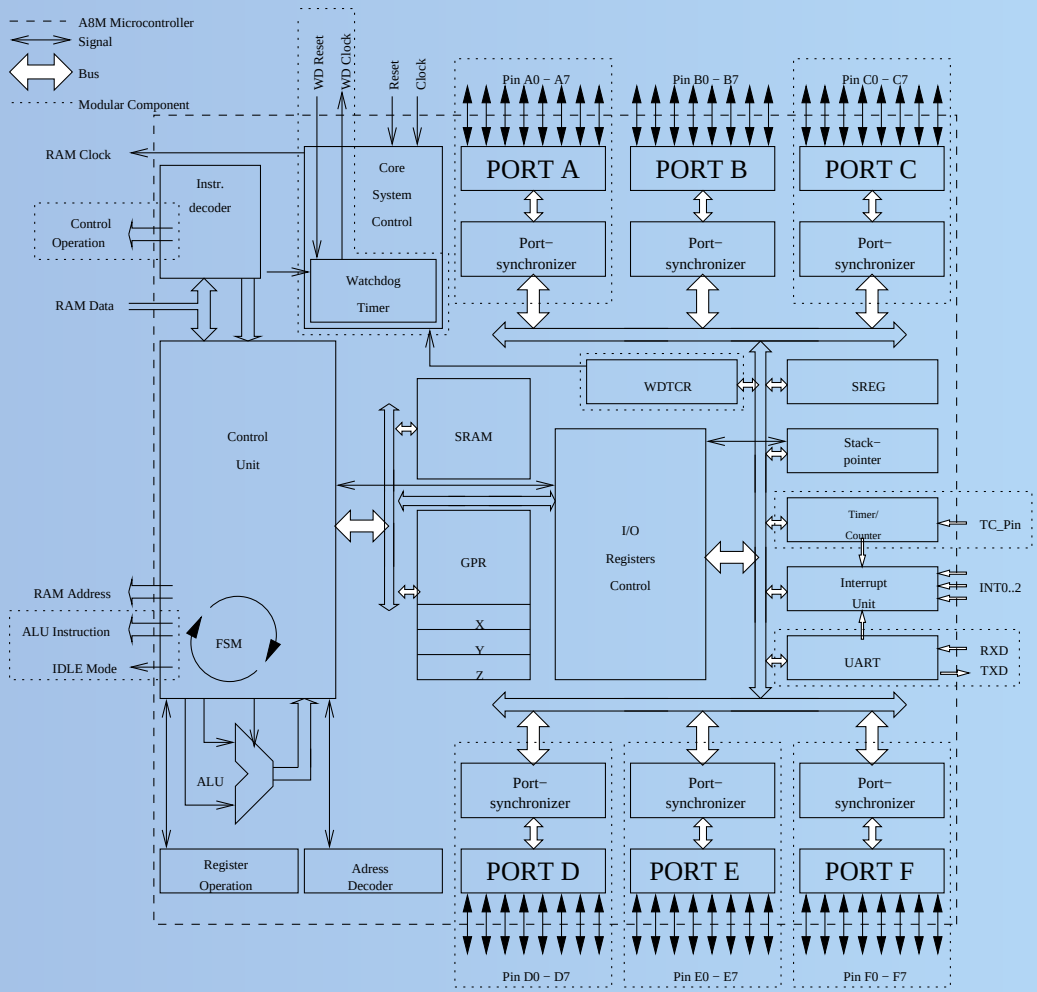
```
i_memory : RAMB16_S18 --%% [LOAD_RAM] i_memory : RAMB16_S9_S18
generic map (
--%% [RAM]
)
port map (
--%% [LOAD_RAM] wea => load_wr,
--%% [LOAD_RAM] ena => '1',
...
we => '0' --%% [LOAD_RAM] web => load_wr,
en => '1' --%% [LOAD_RAM] enb => '1',
...
)
```

AML-Script:

```
--%%[PATH] /project/a8m/

--%%[PATH] /project/program_memory/

--%%[NAME] core_one

--%%[SHORTNAME] c1

--%%[COMPONENT] LOAD_RAM
```

Marcel Naggatz
Technische Universität Dresden
Institut für Technische Informatik
Professur für VLSI-Entwurfssysteme, Diagnostik und Architektur
marcelnaggatz@hotmail.de

A8M Microprozessor

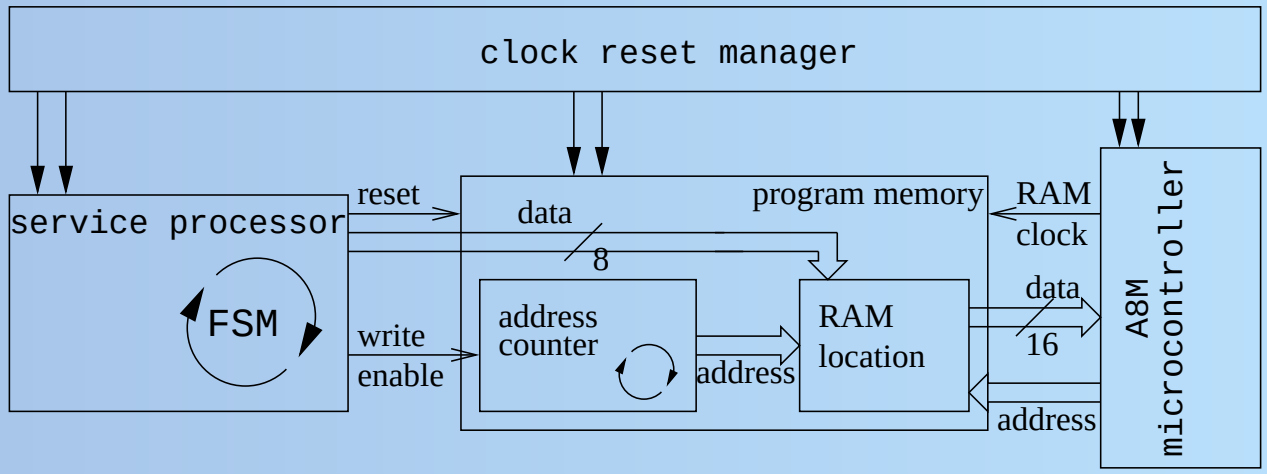


Eigenschaften:

- angelehnt an ATMEL AVR ATmega32
- 8-Bit Microprozessor
- variable Taktraten bis zu 64 MHz
- bis zu 6 I/O-Ports
- 8-Bit Timer/Counter
- UART
- Watchdog Timer
- optionale Steuer- und Kontrollsignale

Marcel Naggatz
Technische Universität Dresden
Institut für Technische Informatik
Professur für VLSI-Entwurfssysteme, Diagnostik und Architektur
marcelnaggatz@hotmail.de

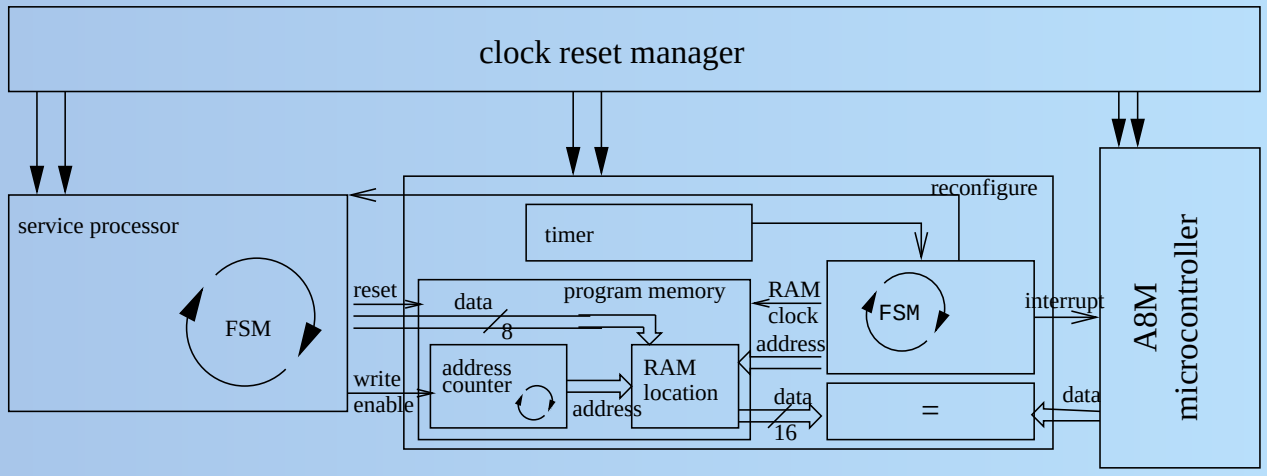
Befüllen des Befehlsspeichers



- universelle Programmspeicher-Komponente
- Dual- oder Single-Port RAM
- interner Adresszähler
- Steuerung erfolgt über das `write enable` und `reset` Signal
- Befüllen kann durch Serviceprozessor oder extern erfolgen

Marcel Naggatz
Technische Universität Dresden
Institut für Technische Informatik
Professur für VLSI-Entwurfssysteme, Diagnostik und Architektur
marcelnaggatz@hotmail.de

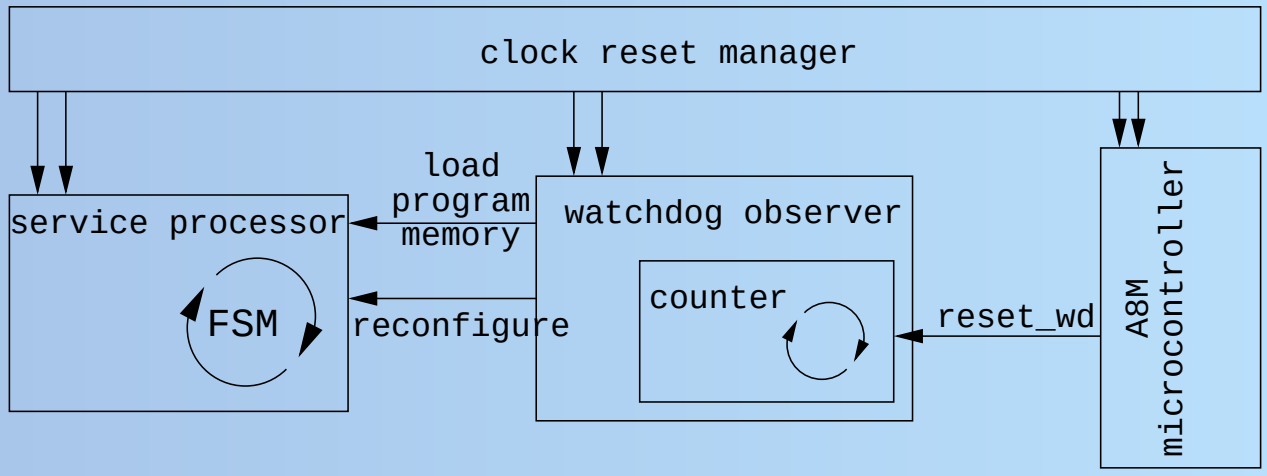
Interrupt Check



- komplexestes und aufwendigstes Testverfahren
- prüft die interne Architektur durch ein Testprogramm
- nutzt die Interruptstruktur des A8M
- besitzt einen internen Timer um die Testintervalle zu steuern
- beinhaltet die Programmspeicher-Komponente als Vergleichsspeicher für die Erwartungswerte

Marcel Naggatz
Technische Universität Dresden
Institut für Technische Informatik
Professur für VLSI-Entwurfssysteme, Diagnostik und Architektur
marcelnaggatz@hotmail.de

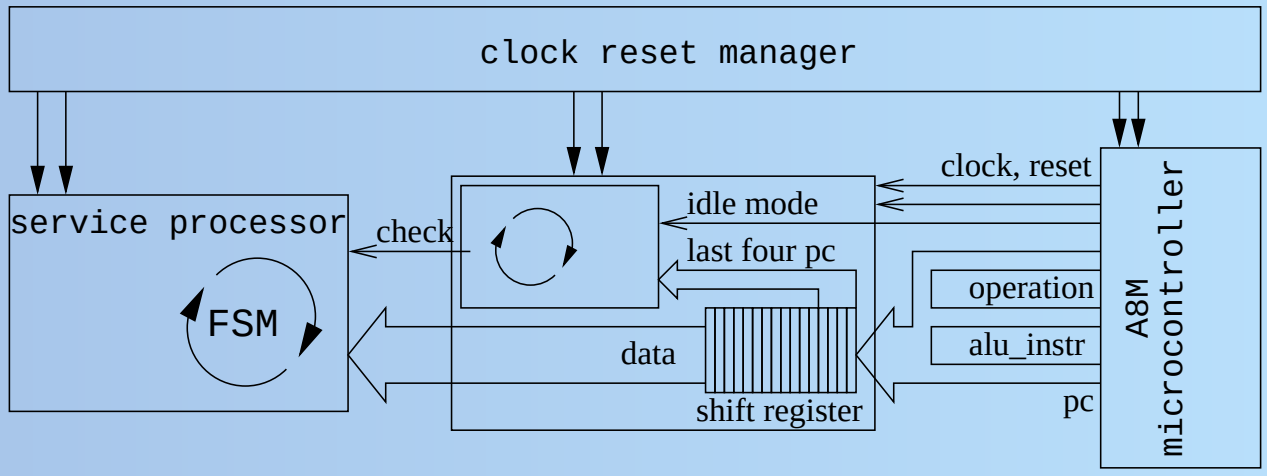
Watchdog-Observer



- registriert einen internen Watchdog Reset
- ein 2-Bit Counter zählt die Resetereignisse
- besitzt zwei Steuerleitungen zum Serviceprozessor
- frei wählbare Fehlerbehandlungsstrategie, z.B.:
  - 2x Reset: veranlasst Neuladen des Befehlsspeichers
  - 4x Reset: Neukonfigurieren des FPGA

Marcel Naggatz
Technische Universität Dresden
Institut für Technische Informatik
Professur für VLSI-Entwurfssysteme, Diagnostik und Architektur
marcelnaggatz@hotmail.de

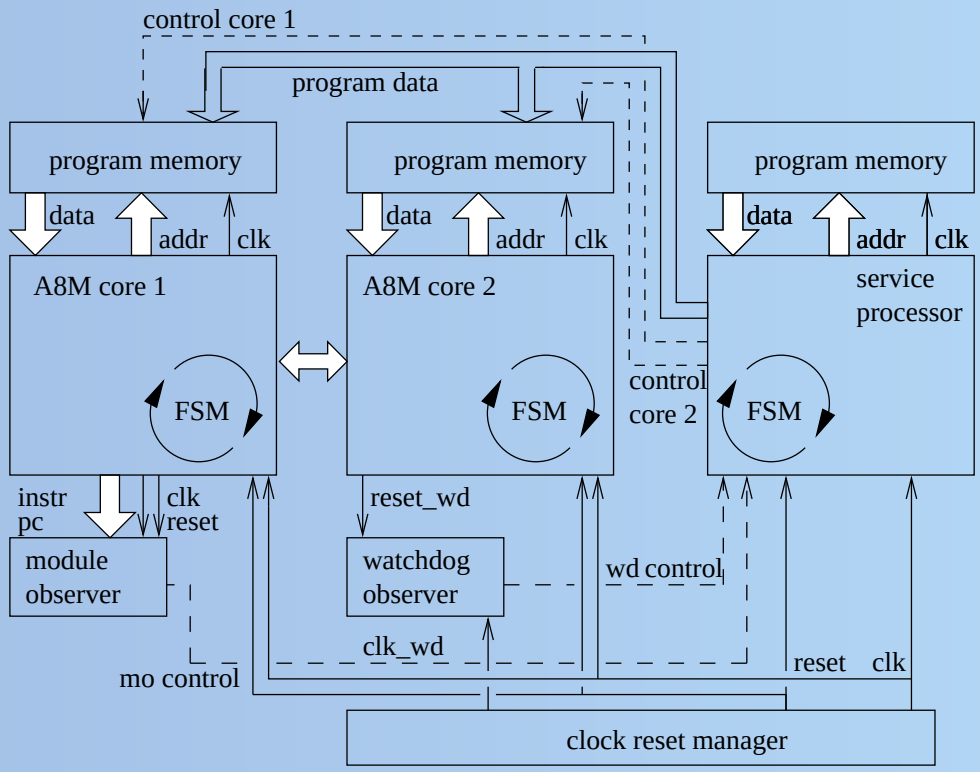
Module-Observer



- überwacht vorhandene Steuersignale des A8M
- Werte werden in einem 16 stufigen Trace File aufgezeichnet
- gleichzeitig werden die letzten vier PC auf Gleichheit überprüft
- Serviceprozessor steuert Fehlebehandlungsstrategie
- Trace File kann vom Serviceprozessor ausgelesen werden

Marcel Naggatz
Technische Universität Dresden
Institut für Technische Informatik
Professur für VLSI-Entwurfssysteme, Diagnostik und Architektur
marcelnaggatz@hotmail.de

Beispiel für eine Agentenarchitektur



bestehend aus:

- 2 A8M Microprozessoren
- 1 Serviceprozessor
- 2 Testkomponenten
  - Watchdog Observer
  - Module Observer

Marcel Naggatz
Technische Universität Dresden
Institut für Technische Informatik
Professur für VLSI-Entwurfssysteme, Diagnostik und Architektur
marcelnaggatz@hotmail.de

Inhalt

4 Test und Auswertung

Marcel Naggatz
Technische Universität Dresden
Institut für Technische Informatik
Professur für VLSI-Entwurfssysteme, Diagnostik und Architektur
marcelnaggatz@hotmail.de

Test und Auswertung

- der Test der entwickelten Hardwarearchitekturen erfolgte mittels Testbenches und einem Prototypen Board
- für die Testausführung wurden Assemblerprogramme geschrieben, welche von den jeweils verwendeten A8M Kernen ausgeführt wurden
- Beispielprogramm ist eine einfache Benutzereingabe und ein Zähler, dessen Wert über einen I/O-Port ausgegeben wird
- erweitert wurde das Testprogramm um eine künstliche Fehlereinspeisung
- für jede Test- und Überwachungskomponente wurden die Self-X-Eigenschaften geprüft und nachgewiesen

Marcel Naggatz
Technische Universität Dresden
Institut für Technische Informatik
Professur für VLSI-Entwurfssysteme, Diagnostik und Architektur
marcelnaggatz@hotmail.de

Beispiel für Assemblerprogramme

auf externen Tastendruck reagieren	laden des Befehlsspeichers

```
.def MEM = r16
LDI     r20, $FF
WAIT:
IN      r21, $10
MOV     r23, r22
AND     r23, r21
SBRC    r23, 0
JMP     LOAD_RAM
RETURN:
MOV     r22, r21
COM     r22
JMP     WAIT
  ⋮
```

```
LOAD_RAM:
  ⋮
LDI     MEM, $0F
OUT     $18, MEM
SBI     $1B, 2
CBI     $1B,2
LDI     MEM, $EF
OUT     $18, MEM
SBI     $1B, 2
CBI     $1B,2
LDI     MEM, $01
  ⋮
```

anlegen eines neuen Befehlsteils

auf den internen Verbindungsbus legen

Steuerung des Schreibbefehles

einlesen eines Tastendrucks

Tastenbetaetigung testen

steigende Flanke erkennen

Marcel Naggatz
Technische Universität Dresden
Institut für Technische Informatik
Professur für VLSI-Entwurfssysteme, Diagnostik und Architektur
marcelnaggatz@hotmail.de

Inhalt

5 Zusammenfassung und Ausblick

Marcel Naggatz
Technische Universität Dresden
Institut für Technische Informatik
Professur für VLSI-Entwurfssysteme, Diagnostik und Architektur
marcelnaggatz@hotmail.de

Zusammenfassung

- Studien zum Entwurf von Hardwareagenten
- Konzeption einer Hardwarearchitektur und deren Erweiterungen (Test- und Überwachungskompnenten und Serviceprozessoren)
- Entwurf eines modularen Microprozessors A8M
- Entwurf von drei kombinierbaren Test- und Überwachungskomponenten
- Entwurf einer Generatorsoftware zum Anpassen des A8M an die jeweiligen Erfordernisse
- Entwurf einer Scriptsprache zur Steuerung der Generatorsoftware
- Entwurf eines Hardwareagenten-Prototypen und Test der Architektur

Marcel Naggatz
Technische Universität Dresden
Institut für Technische Informatik
Professur für VLSI-Entwurfssysteme, Diagnostik und Architektur
marcelnaggatz@hotmail.de

Ausblick

- Erweiterung der verwendeten Komponenten, um einen Pool von modularen Blöcken zu erhalten
- Erweiterung der Architektur um eine geeignete Kommunikationsstruktur
- Entwurf einer komplexeren Beispielarchitektur zur Anwendung einer solchen Agentenarchitektur
  - Auswertung verschiedener Sensoren
  - Steuerung von Aktoren
- Erweiterung der Generatorsoftware, um Programmcode zu übertragen
- Erweiterung der Generatorsoftware, um die automatische Erstellung einer Topkomponente und dem automatischen Aufruf des Synthesetools

Marcel Naggatz
Technische Universität Dresden
Institut für Technische Informatik
Professur für VLSI-Entwurfssysteme, Diagnostik und Architektur
marcelnaggatz@hotmail.de

Inhalt

Vielen Dank für
Ihr Interesse

Fragen ?

Marcel Naggatz
Technische Universität Dresden
Institut für Technische Informatik
Professur für VLSI-Entwurfssysteme, Diagnostik und Architektur
marcelnaggatz@hotmail.de