imat

Rapid Control Prototyping mechatronischer Systeme auf Basis rekonfigurierbarer FPGAs

Steffen Toscher, Roland Kasper,
Thomas Reinemann

Institut für Mechatronik und Antriebstechnik
Otto-von-Guericke-Universität Magdeburg

Rapid Control Prototyping mechatronischer Systeme auf Basis rekonfigurierbarer FPGAs

Inhalt

- Einleitung
- Rekonfiguration von FPGAs
- Rapid Prototyping in Software und Hardware
- Spezifikation und Implementierung eines dynamisch rekonfigurierbaren Controllers
- Steuerung der dynamischen Rekonfiguration
- Schnittstellen
- Anwendung
- Zusammenfassung

Einleitung

- feldprogrammierbare Gate Arrays (FPGAs) bieten hohe Rechenleistung, Echtzeitfähigkeit und generelle Rekonfigurierbarkeit
- Einsatz in Rapid Prototyping Systemen bisher vor allem für I/O-Funktionen, hardwarenahe und zeitkritische Aufgaben ohne echte Ausnutzung der Rekonfigurierbarkeit / Reprogrammierbarkeit
- dynamische Rekonfiguration von FPGAs ermöglicht Portierung von aus der Software-Entwicklung bekannten Features in den Bereich der Hardware-Entwicklung
- online Parameteränderungen, Visualisierung interner Größen, Änderung der Reglerstruktur, Austausch von Funktionen und Anpassung von Schnittstellen sind so für die Hardware-Entwicklung realisierbar

➔ zielsystemnahe Umsetzung von flexiblen und strukturvariablen Hardware-Systemen innerhalb eines dynamisch rekonfigurierbaren Rapid Prototyping Systems

Rekonfiguration von FPGAs

Rekonfiguration: Anpassung der Hardware an veränderte Anforderungen

statische Rekonfiguration

- Grundeigenschaft SRAM-basierter FPGAs
- FPGA ist inaktiv und wird komplett oder partiell rekonfiguriert
- Schnittstellen, flexible Hardware-Plattform, Weiterentwicklung

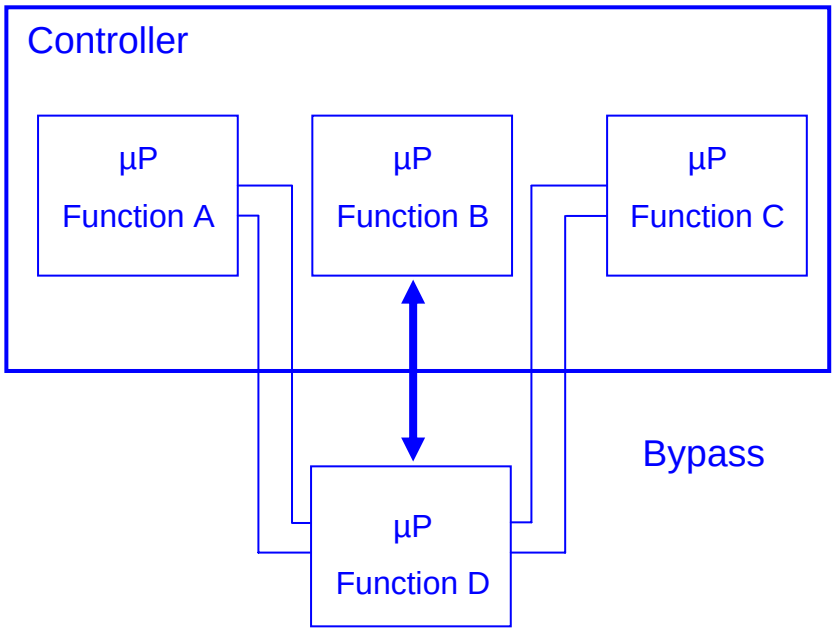
dynamische Rekonfiguration

- Anpassung an Arbeitspunkte, Kennlinien, Betriebszustände
- FPGA ist aktiv, Austausch von Teilen der Hardware im laufenden Betrieb
- komplexe Werkzeuge und Methoden

➔ dynamisch rekonfigurierbarer Controller für Rapid Control Prototyping

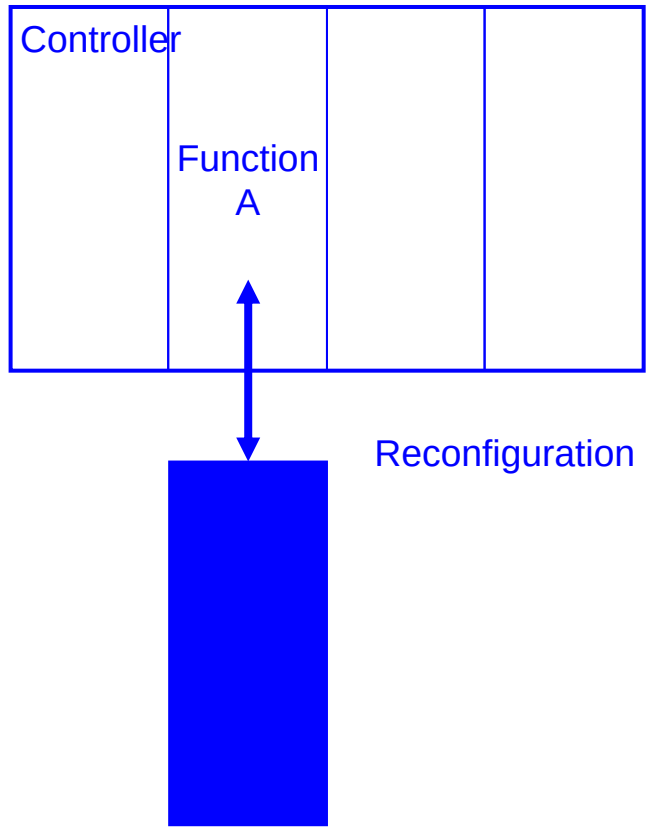
Rapid Prototyping in Software und Hardware

Software



- Bypass als Beispiel
- Auslagerung von Software-Funktionen auf externe Hardware

Hardware (FPGA)



- Funktionen werden direkt in Hardware abgebildet
- Austausch von Teilen der Hardware durch dynamische Rekonfiguration

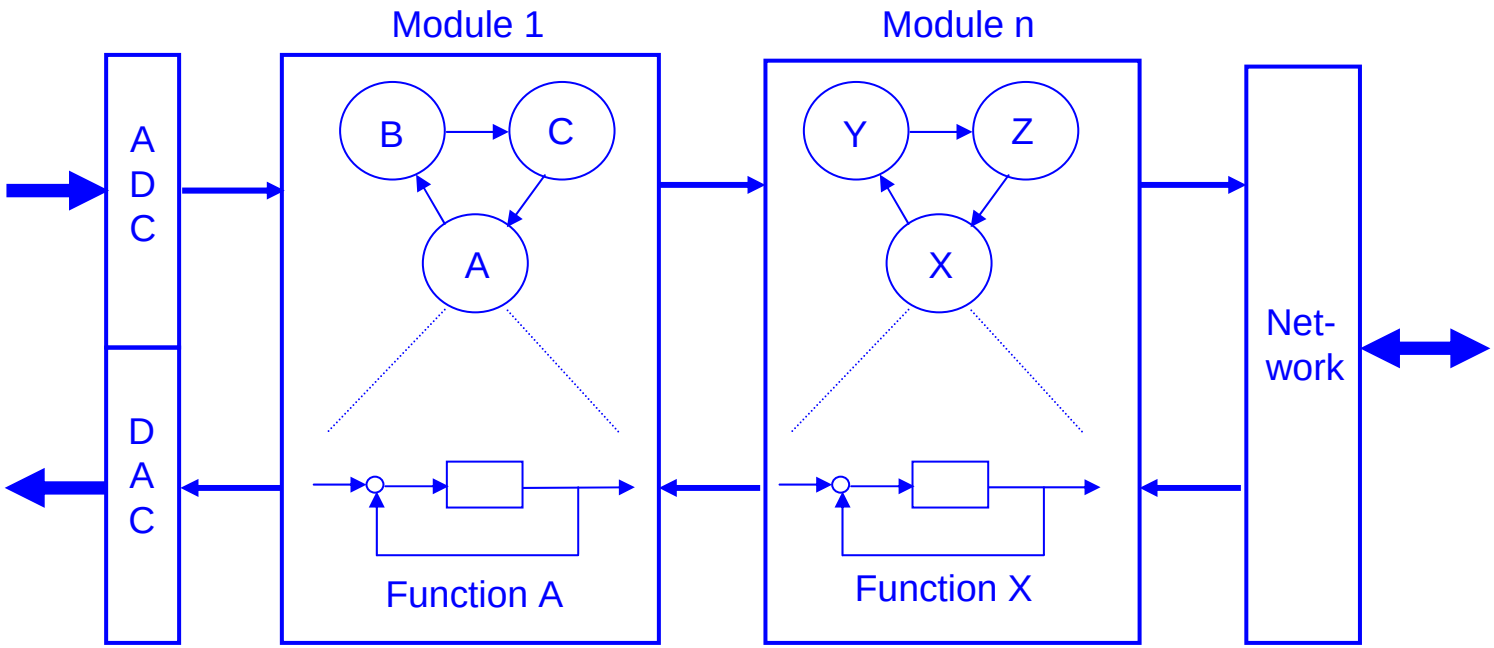
Spezifikation eines dynamisch rekonfigurierbaren Controllers

Datenfluss

- Blockdiagramm basierend
- Beschreibung der Signalverarbeitung

Steuerfluss

- Aktivierung der Algorithmen, Strukturvariabilität
- Zustandsautomaten (FSM) basierend



Spezifikation eines dynamisch rekonfigurierbaren Controllers

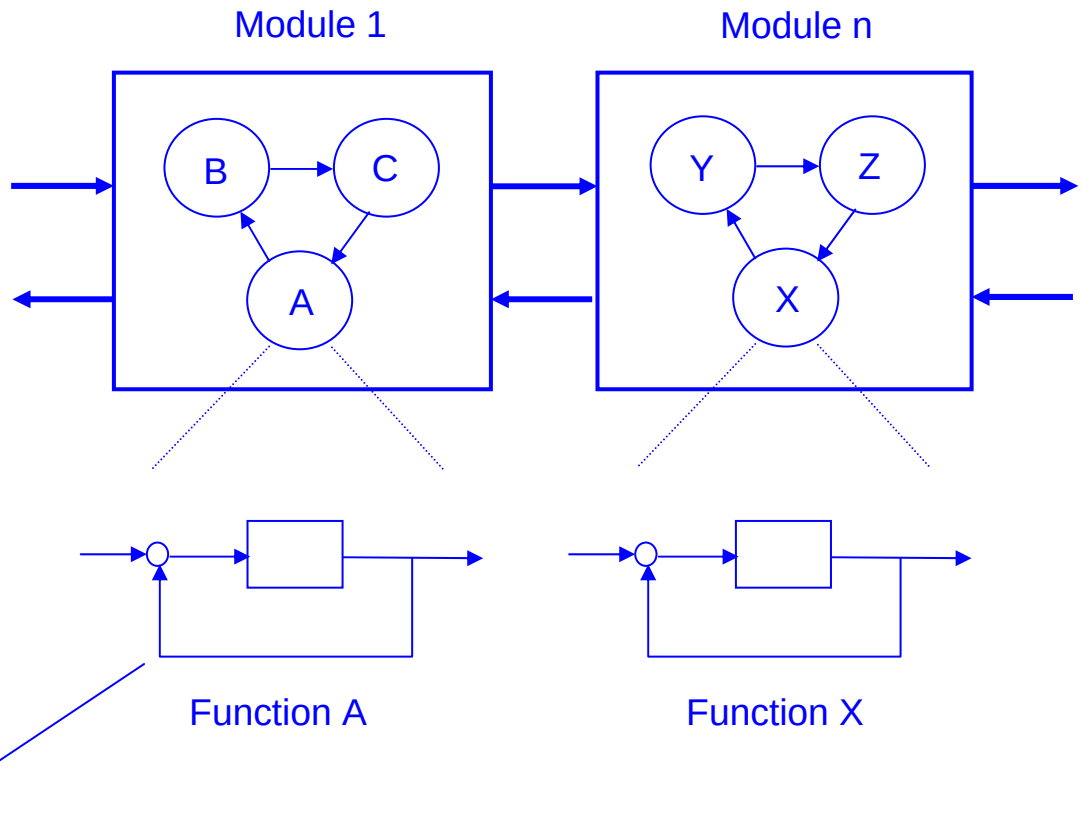
Toplevel ⟺ **Module**, Slots

Funktionen ⟺ **Typen** als Zustände von FSMs

Blöcke ⟺ Gatter, Register

Signale ⟺ Leitungen

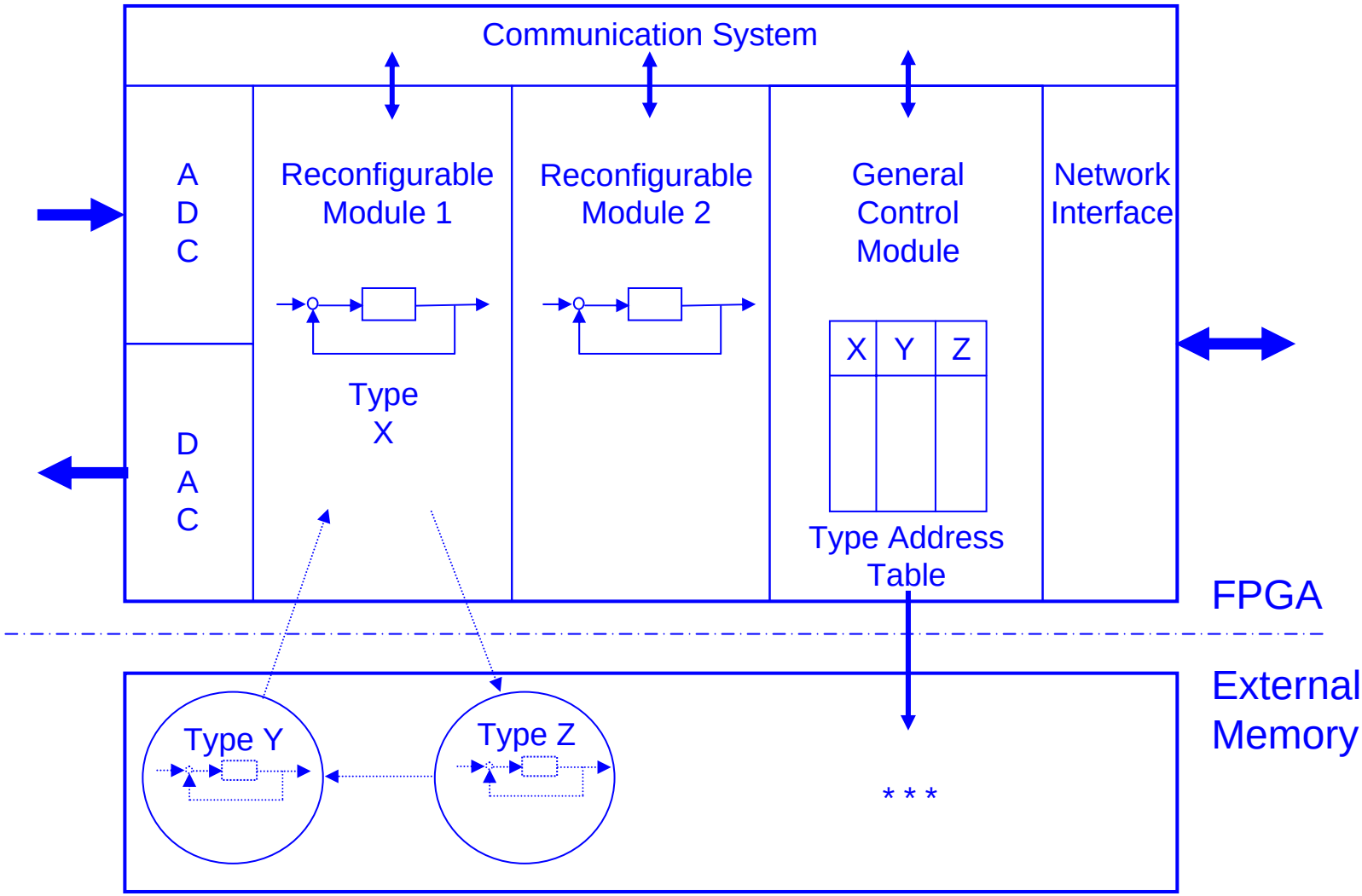
bitseriell



➔ Funktionsbibliotheken für Blockdiagrammeditor

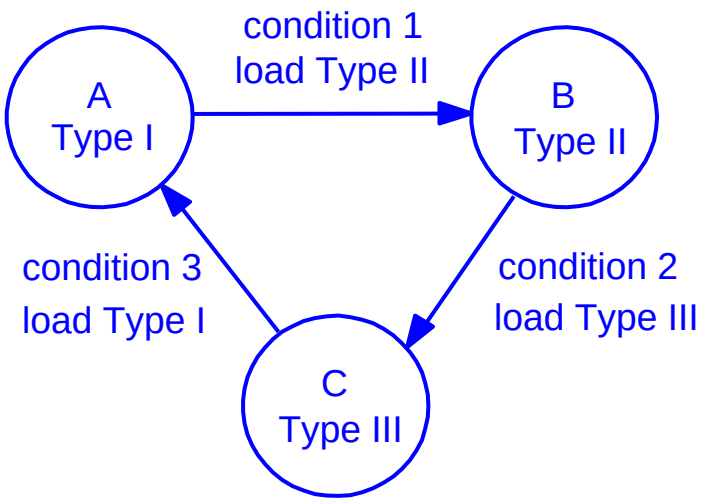
➔ Hardwarebeschreibungssprachen (HDLs) werden umgangen

Implementierung eines dynamisch rekonfigurierbaren Controllers



Steuerung der dynamischen Rekonfiguration
Modul FSM

- definiert die Reihenfolge für das Laden und Aktivieren der Typen (Funktionen) für ein Modul
- resultiert aus der Spezifikation der Zustandsautomaten

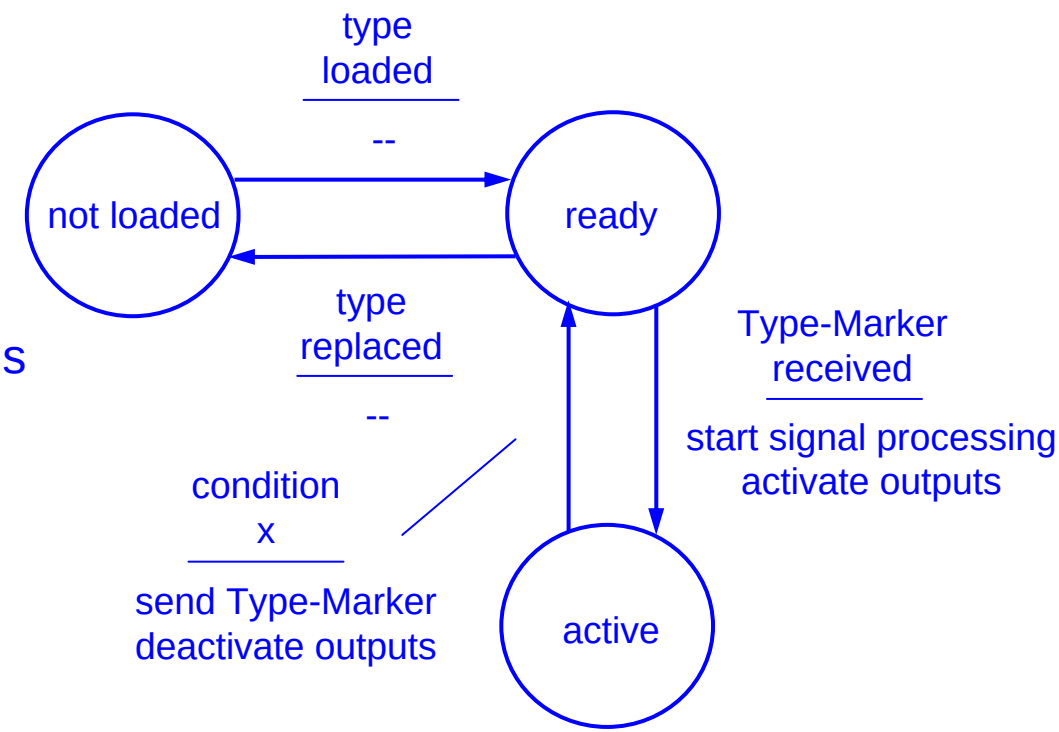


Beispiel-FSM

- eine Aktivierungsinformation ist auszutauschen, da immer nur ein Typ aktiv sein darf
- verteilte Modul-FSM Implementierung, mehrere Typen sind gleichzeitig geladen
- um Echtzeitbedingungen einzuhalten können alle erreichbaren Zustände vorgeladen werden

Steuerung der dynamischen Rekonfiguration
Typ FSM

- steuert die Aktivierung und Deaktivierung der Typen eines Moduls
- Struktur ist für alle Typ-FSMs identisch



Typ-Marker

- jeder Typ hat seinen eigenen Typ-Marker zur Aktivierung
- Zuordnung Typ ⇔ Typ-Marker erfolgt während der Kompilierung

Schnittstellen

Netzwerk-Schnittstelle

- Anbindung an übergeordnete Strukturen
- Universal Serial Bus 2.0 (USB 2.0) ist implementiert
- statisch rekonfigurierbar ausgelegt, um Austausch der Schnittstelle zu ermöglichen
- Schnittstellen der Informationstechnik (Ethernet…), der IC-Kommunikation (I^2C, SPI…) und Feldbusschnittstellen (CAN, Profibus…) sind alternativ implementierbar

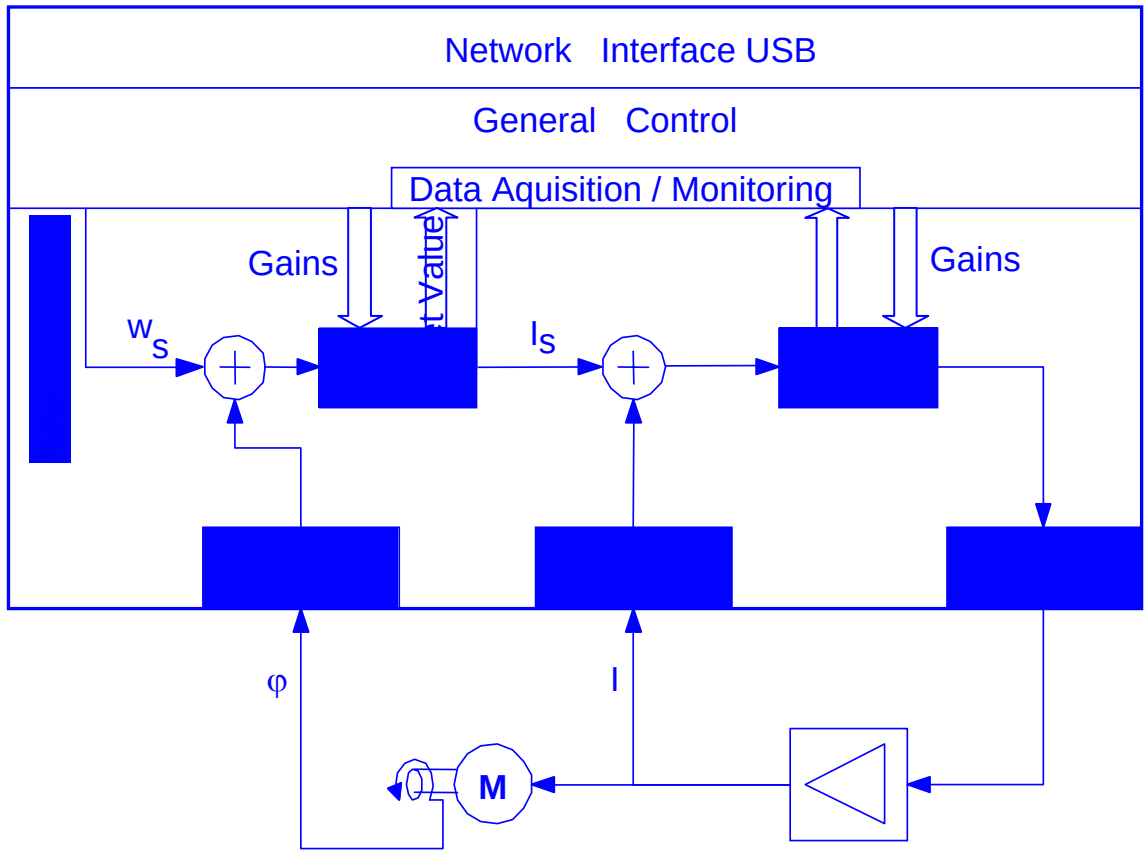
Prozess-Schnittstelle

- Anbindung der Prozesssignale
- digital: PWM
- Delta-Sigma-Analog/Digital-Wandler: Wortbreite 8-16 Bit, variable Kanalanzahl, Samplingfrequenzen bis 100kHz
- Implementierung anpassbarer digitaler Filter direkt im FPGA unterstützt Rekonfigurierbarkeit

➔ Bibliotheken für Blockdiagrammeditor

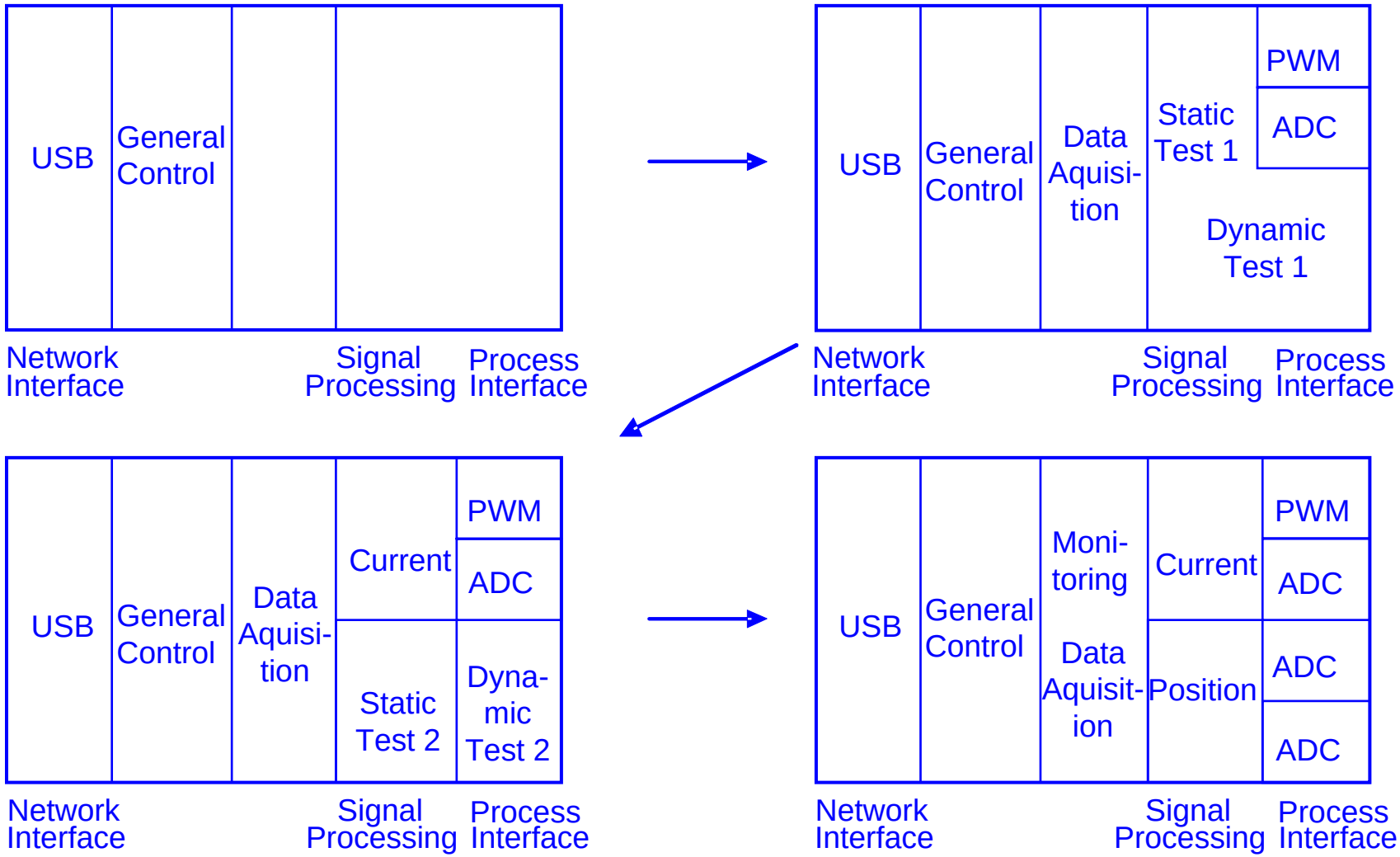
Anwendung

Universeller rekonfigurierbarer Antriebscontroller auf Xilinx Virtex-II FPGA



- online Änderung der Regelstruktur
- online Parameteränderungen
- online Visualisierung controllerinterner Größen
- online Funktionsänderung durch Austausch von Modulen
- online Sollwertvorgabe
- Rekonfiguration der Schnittstellen

Anwendung



Inbetriebnahme des universellen rekonfigurierbaren Antriebscontrollers

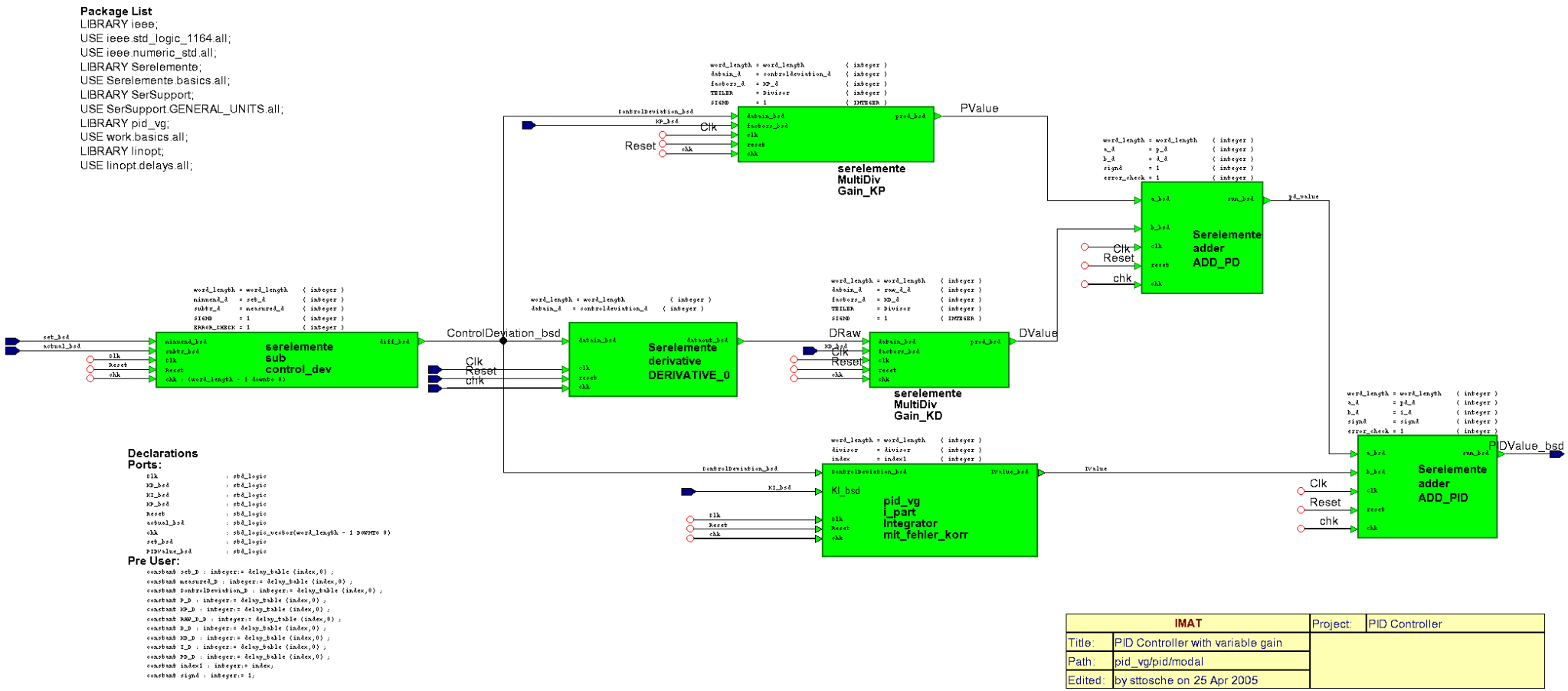
➔ online Funktionsänderung durch Austausch von Modulen

Zusammenfassung

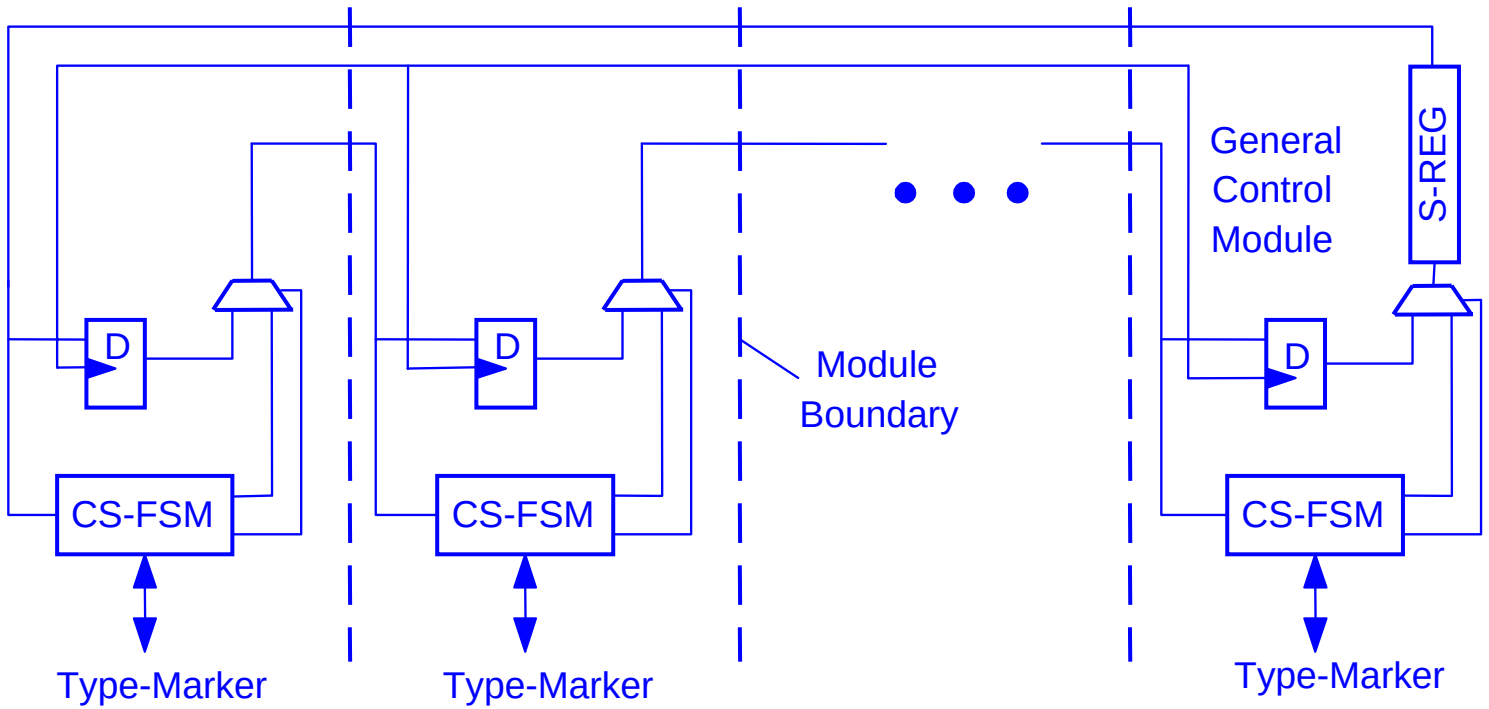
- neuartiger rekonfigurierbarer Controller als Kern eines Rapid Prototyping Systems bzw. zur Erweiterung bestehender Systeme wurde entwickelt
- Methoden für den Einsatz der dynamischen Rekonfiguration auf FPGAs in Rapid Prototyping Umgebungen wurden implementiert
- echtzeitfähige Rekonfigurationsmechanismen wurden entwickelt und implementiert
- Entwurfsmethoden erlauben Einsatz der dynamischen Rekonfiguration auch ohne detaillierte Kenntnisse der Hardware-Plattform FPGA (Blockdiagrammeditor)
- Portierung von aus der Software-Entwicklung bekannten Prototyping Features in den Bereich der Hardware-Entwicklung
- rekonfigurierbare Schnittstellen unterstützen die Flexibilität des Systems

➔ Validiertes Konzept zur Nutzung der dynamischen Rekonfiguration im Bereich Rapid Control Prototyping mechatronischer Systeme ist vorhanden

Vielen Dank!


Package List
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.numeric_std.all;
LIBRARY Serelemente;
USE Serelemente.basics.all;
LIBRARY SerSupport;
USE SerSupport.GENERAL_UNITS.all;
LIBRARY pid_vg;
USE work.basics.all;
LIBRARY linopt;
USE linopt.delays.all;
serelemente sub control_dev
ControlDeviation_bsd
Clk
Reset
chk
Serelemente derivative DERIVATIVE_0
DRaw
DValue
serelemente MultiDiv Gain_KD
serelemente MultiDiv Gain_KP
PValue
Serelemente adder ADD_PD
pid_value
Serelemente adder ADD_PID
PIDValue_bsd
pid_vg i_part integrator mit_fehler_korr
Declarations
Ports:
Pre User:
IMAT
Project:
PID Controller
Title:
PID Controller with variable gain
Path:
pid_vg/pid/modal
Edited:
by sttosche on 25 Apr 2005

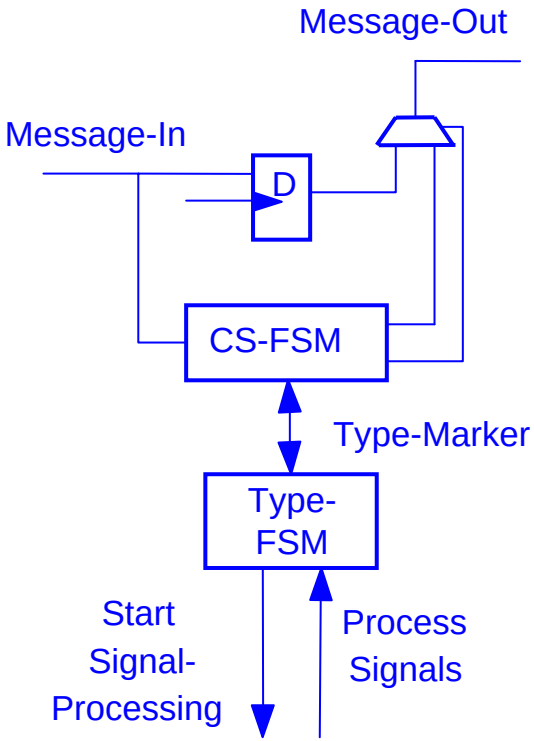
Steuerung der dynamischen Rekonfiguration
Communication System



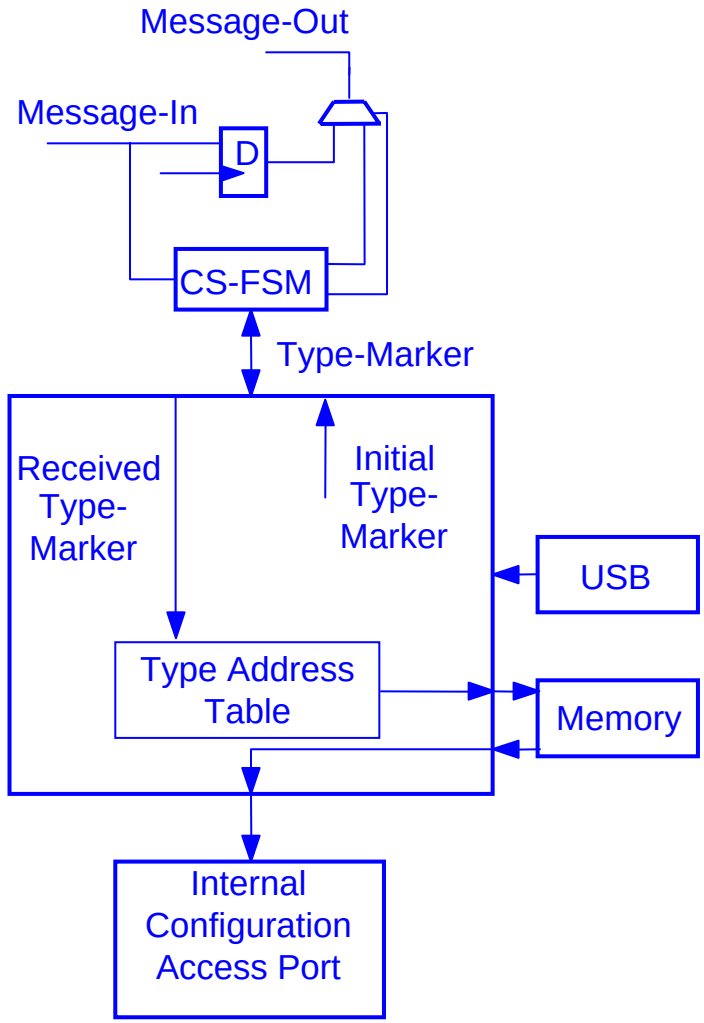
- Austausch von Informationen zur Rekonfiguration (Typ-Marker, Status, …)
- statisch, nicht rekonfigurierbar, Überbrücken von Slots während der Rekonfiguration
- General Control Module (GCM) nimmt an der Kommunikation teil
- deterministisches Zeitverhalten

Steuerung der dynamischen Rekonfiguration Control System

Rekonfigurierbarer Slot



General Control Modul



Dezentrale Lösung

- weiterschalten der Typ-FSM erfolgt im Slot
- GCM muss keine Prozesssignale verarbeiten