



Schwerpunktprogramm 1148
„Rekonfigurierbare Rechensysteme“

imat

Rekonfigurierbare Controller

Roland Kasper

Thomas Reinemann

Institut für Mechatronik und Antriebstechnik

Otto-von-Guericke-Universität Magdeburg

Rekonfigurierbare Controller

Inhalt

- Controller Anwendungsfelder
- Implementierungsformen: Mikrocontroller - FPGA
- Anforderungen der Mechatronik
- Algorithmen in Hardware und ihre Spezifikation
- Rekonfiguration
- Einbindung in SPP

Controller Anwendungsfelder

KFZ

- Motormanagement
- Fahrdynamik
ABS, ASR, ESP, ...
- Getriebe
- Komfort
- ...



Consumer

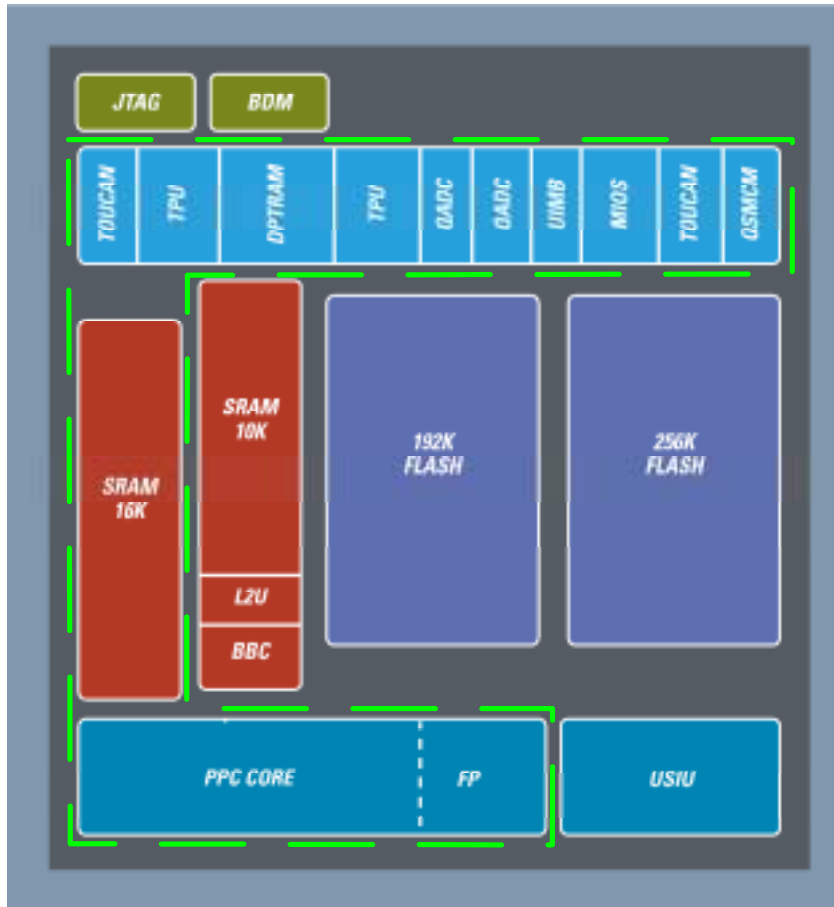
- Drucker, Kopierer
- Videorecorder, Camcorder,
Fotoapparat



Optimierung bezüglich

- Kosten: CPU, Speicher, Peripherie
- Baugröße, Betriebsumgebung

Implementierungsform Mikrocontroller



Fest

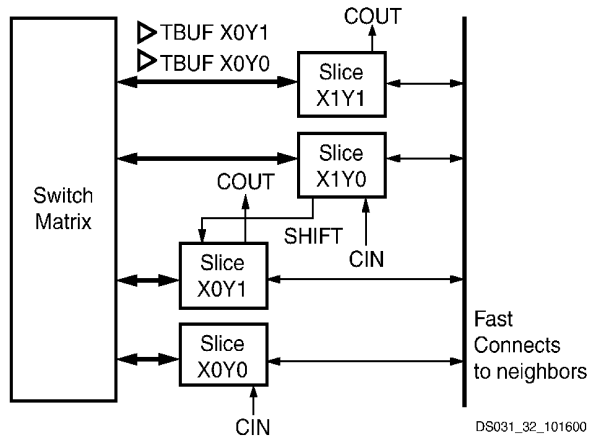
- Hardwarestruktur
- Speichergröße
- Wortbreite 8, 16, 32 Bit
- Interne Busse
- Co-Prozessoren
- Sequentielle Arbeitsweise

Flexibilität

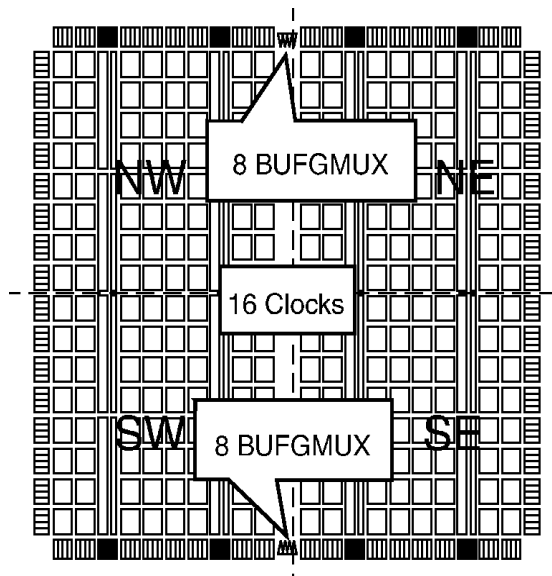
- Software
- ➔ Echtzeitprobleme

Implementierungsform FPGA

Verlagerung der Algorithmen vom Speicher auf eine Logikebene



- Frei programmierbare Logikzellen
- Freie Verdrahtung
- Massive Parallelität
- Deterministisches Zeitverhalten
- Geringe Echtzeitprobleme
- Geringe Taktraten
- Weniger vertraute Technologie



Anforderungen der Mechatronik

Strukturen/Elemente

- Regler
- Steuerungen
- Digitalfilter
- Adaption
- Komplexe Zustandsmaschinen
- Protokolle
- ...

Betriebsbedingungen

- Samplingraten 0,1-100kHz
- Hohe Präzision
- Sicherheit
- Energieverbrauch
- Kosten
- ...

Mikrocontroller + FPGA
Software + konfigurierbare Hardware

Algorithmen in Hardware

Bekannte Lösungen

- Basiskomponenten: Hardwareaddierer, -multiplizierer
- Spezialstrukturen: FIR, IIR Filter
- Bitparallele Verarbeitung
- Spezifikation mittels HDL

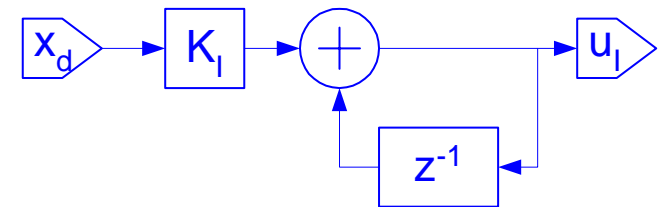
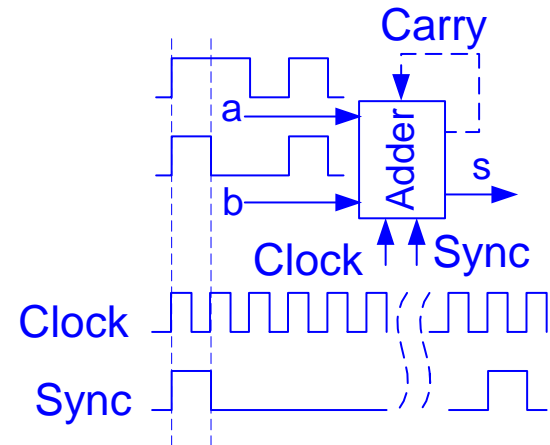
Nachteile

- Komplexität
- Hoher Ressourcenverbrauch
- Herstellerabhängigkeit der Bibliotheken

➡ Alternative: Bitserielle Verarbeitung

Bitserielle Verarbeitung

- Grundidee:
Bitserielle Verarbeitung und Übertragung der Operanden
- Basiskomponenten:
Umfangreiche Bibliothek bitserieller Operatoren und Funktionsblöcke
- Komplexe Systeme:
Automatisches Synchronisationsverfahren



Bibliothek bitserieller Komponenten

- Addition, Subtraktion, Verstärkung, ...
- Multiplikation, Kennlinienapproximation, ...
- PID-Regler, FIR/IIR-Filter, ...
- Parallel-seriell und seriell-parallel Wandler, A/D und D/A-Wandler

Automatische Synchronisation

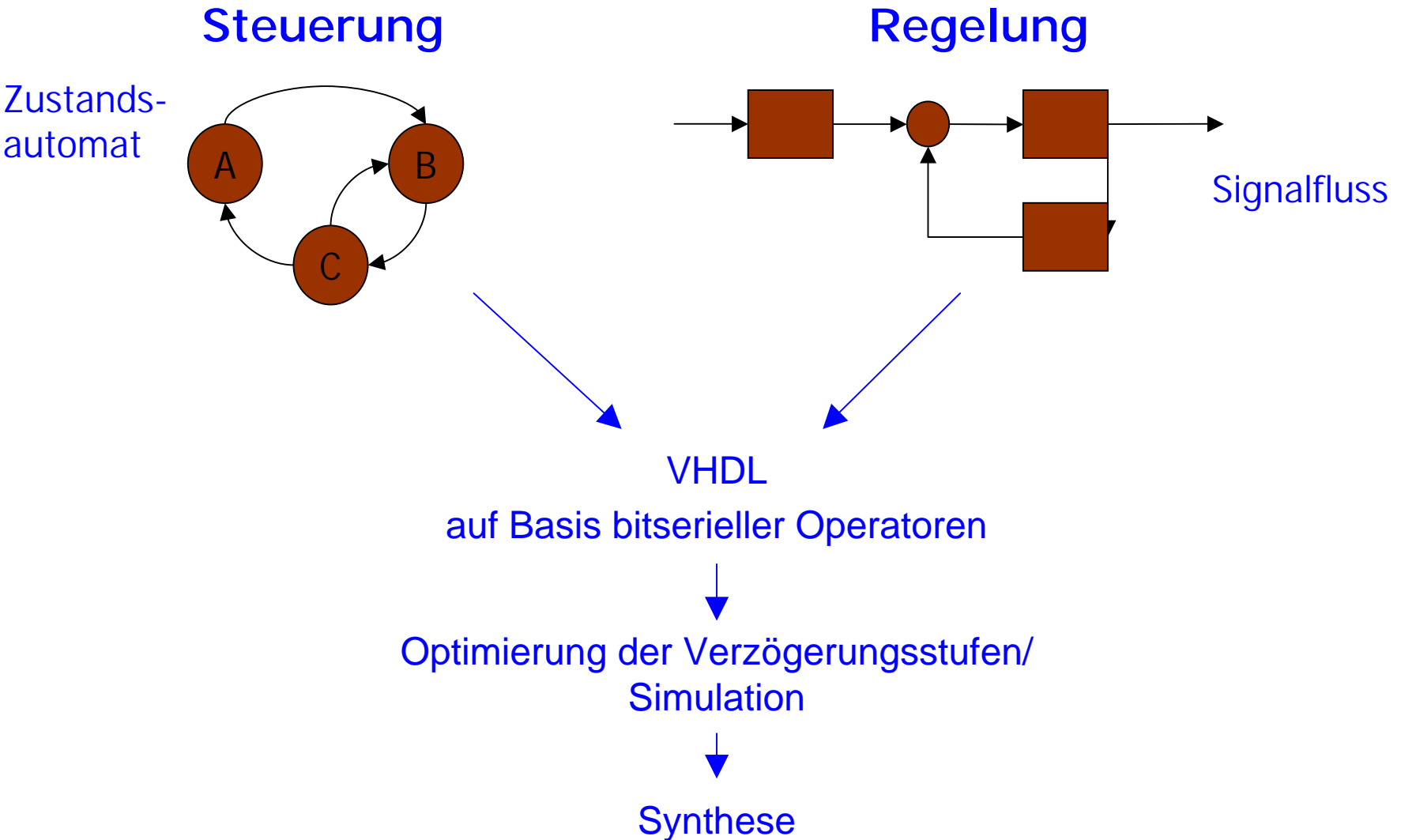
- Analyse des VHDL-Codes
- Formulierung linearen Gleichungssystems
- Lösen des linearen Optimierungsproblems
- Einspeisen der erhaltenen
Signalverzögerungen in die
Simulation/Synthese

$$\mathbf{u} = \mathbf{P} \cdot \mathbf{y} + \mathbf{P}_0 \cdot \mathbf{u}_0$$

$$\mathbf{y} = \mathbf{C}_v \cdot \mathbf{v} + \mathbf{C}_u \cdot \mathbf{u} + \mathbf{v}_k$$

$$\mathbf{y}_0 = \mathbf{C}_0 \cdot \mathbf{y}$$

Spezifikation mittels Blockdiagrammeditor



Restriktionen programmierbarer Hardware

- Temporär nicht benötigte Logikkomponenten verbrauchen Ressourcen und Energie
- Chipfläche / Integrationsdichte
- Energieverbrauch
- Variantenvielfalt

Abhilfe durch Rekonfiguration

statisch

dynamisch

Austausch von Komponenten
außerhalb des regulären Betriebs

Austausch von Komponenten
während des regulären Betriebs

(Entwurfszeit, Updates)

(Laufzeit)

Statische Rekonfiguration

Auswahl und Anpassen von

- Komponenten und Funktionen,
- Parametern
- und Schnittstellen

außerhalb des regulären Betriebs

Vorteile

- Reduzierung der Variantenvielfalt
- Ermöglichen von Änderungen im späten Entwicklungsstadium
- Einspielen von Updates

Aufgabenstellungen zur Realisierung der statischen Rekonfiguration

- Modellierung des Ressourcenbedarfs
- Erweitern der Spezifikationsumgebung um Ressourcenmodelle
- Erweitern der Syntheseumgebung um Ressourcenmodelle
- Multirate, Multiwidth
- Rekonfigurierbare Anlogschnittstellen

Randbedingungen

- Anzahl der Eingänge und Ausgänge
- Verfügbare Logikblöcke
- Kompatibilität z.B. bei Updates

Dynamische Rekonfiguration

Dynamischer Austausch von

- Komponenten und Funktionen
- Parametern
- und Schnittstellen

während des regulären Betriebs mit konsistenten
Übergängen

Vorteile

- Reduzierung von Chipfläche und Energieverbrauch

Aufgabenstellungen zur Realisierung der dynamischen Rekonfiguration

- Entwicklung von Rekonfigurationsmechanismen auf Komponenten- und Systemebene
- Dynamisch rekonfigurierbare Anlogschnittstellen
- Erweitern der Spezifikations- und Syntheseumgebung

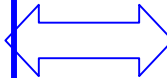
Randbedingungen

- Kombination vorgegebener Konfigurationen
- Rekonfigurierbarkeit der Hardware
- Momentan genutzte Eingänge und Ausgänge, verfügbare Logikblöcke
- Daten- und Signalkonsistenz bei Umschaltung
- Echtzeitbetrieb

Einbindung in das Schwerpunktprogramm

Recontrol

- Effiziente serielle Implementierungsstrategien
- Bereitstellung von Leistungs- und Kostenmodellen
- Anwendungsbeispiele für rekonfigurierbare Controller



SPP

- Diverse Implementierungsstrategien
- Nutzung von Leistungs- und Kostenmodellen
- Konfiguration von Prozessorsystemen

Auf gute Zusammenarbeit