
Entwicklung von Partitionierungsstrategien im Entwurf dynamisch rekonfigurierbarer Systeme

R. Merker,

Technische Universität Dresden, Fakultät ET und IT

J. Kelber,

Fachhochschule Schmalkalden, ET

Gliederung

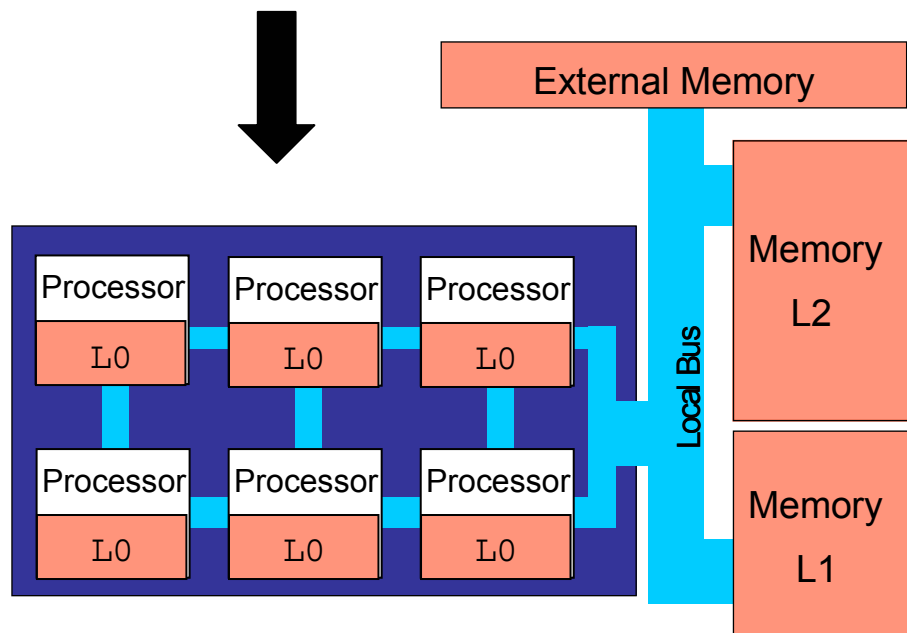
- Vorarbeiten
- Motivation für SPP 1148
- Aufgabenstellung für SPP 1148

Vorarbeiten

WERKZEUGE FÜR HARDWARE-ENTWURF

Entwurfsmethoden und -werkzeuge für applikationsspezifische digitale parallele Hardware (ASIC, FPGA)- eingebettet im System

Algorithmus



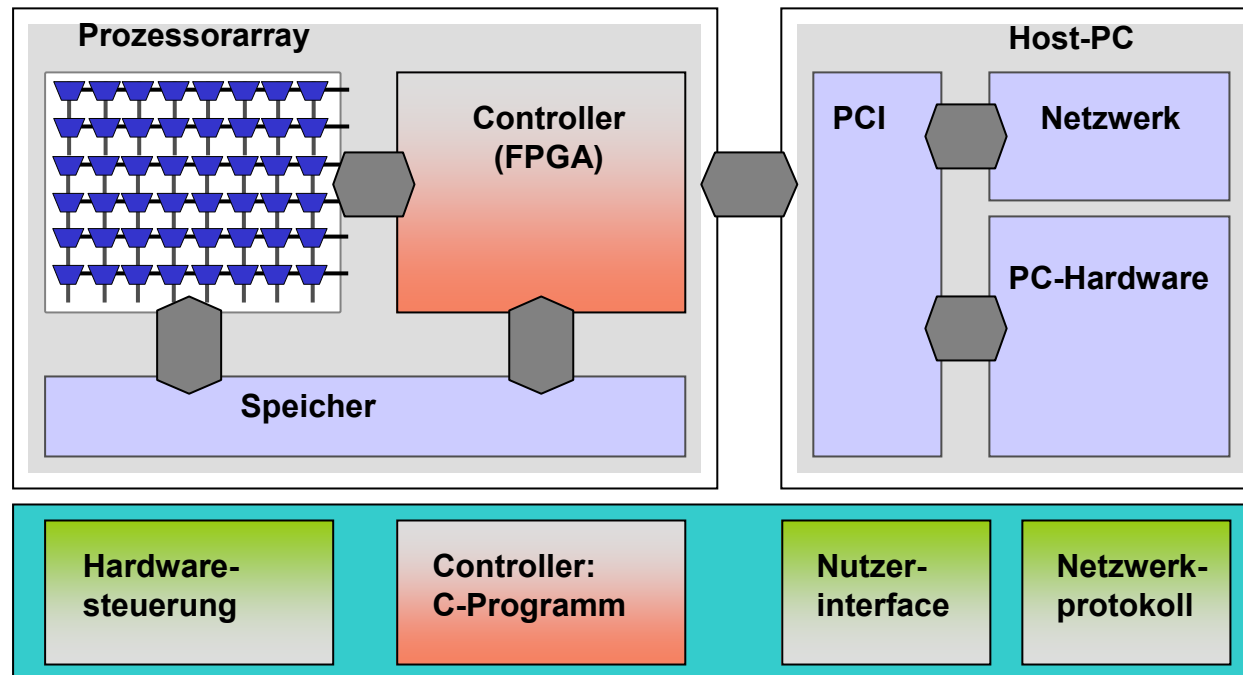
- + Implementierungsbeschränkungen
 - Chipfläche
 - Latenz
 - Modultypen für EXU
 - Zahl der Instanzen der Modultypen
 - Bandbreite/ Latenz der Kanäle
- + Interfacebeschränkungen
 - transportierbare Datenmenge/ Zeit
 - Tiefe Speicherhierarchie
 - Art (FIFO, LRU Cache), Kapazität, Verlustleistung der Speicher

Vorarbeiten

ENTWURF EINES HARDWARE-SOFTWARE SYSTEMS

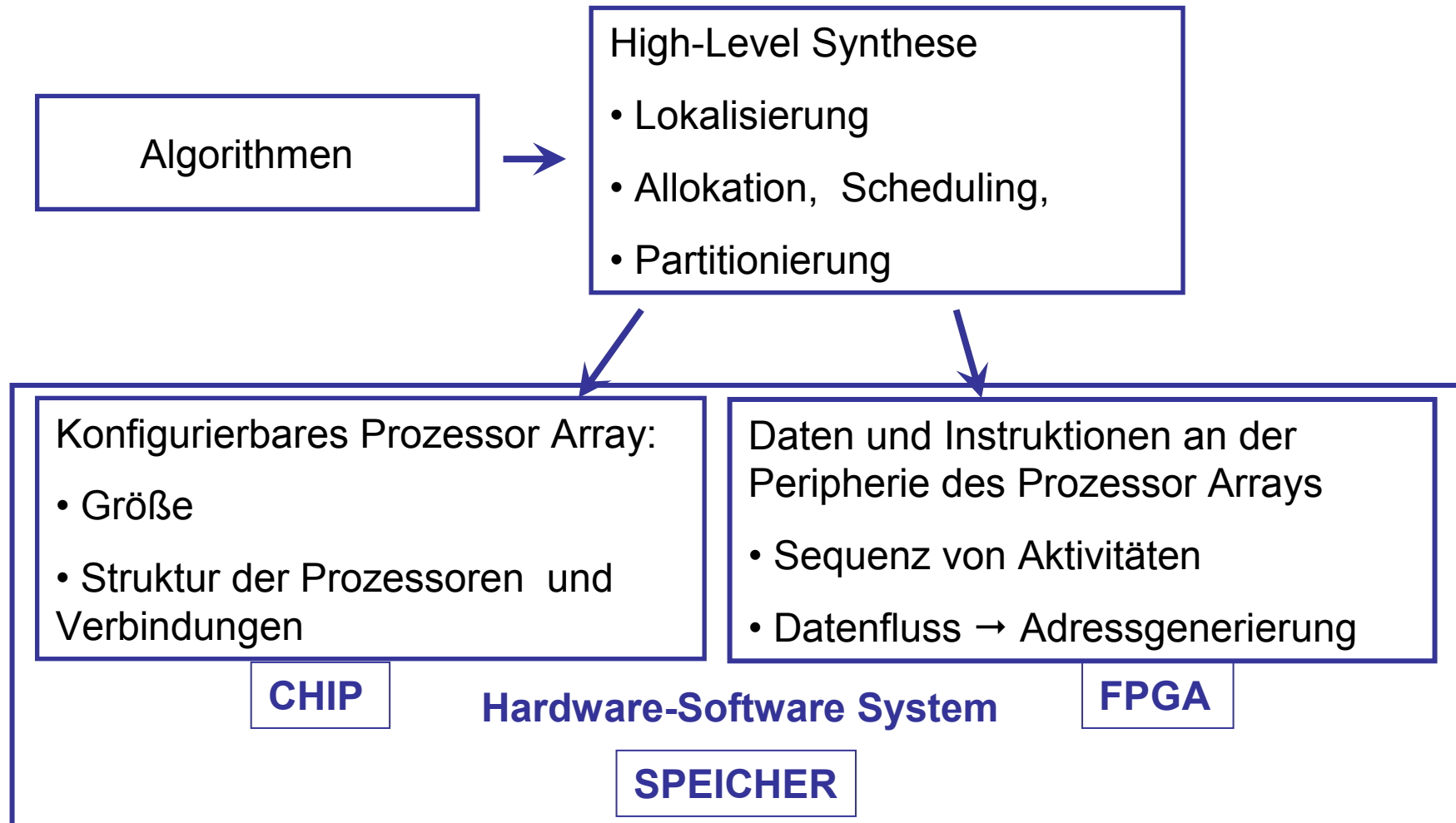
für rechenintensive Probleme (Bildrekonstruktion, Filterung, u.a.):

- HW/SW-Partitionierung,
- Entwurf von auf Kernrechnungen zugeschnittener paralleler Hardware (ASIC),
- Controller-Entwurf (FPGA)



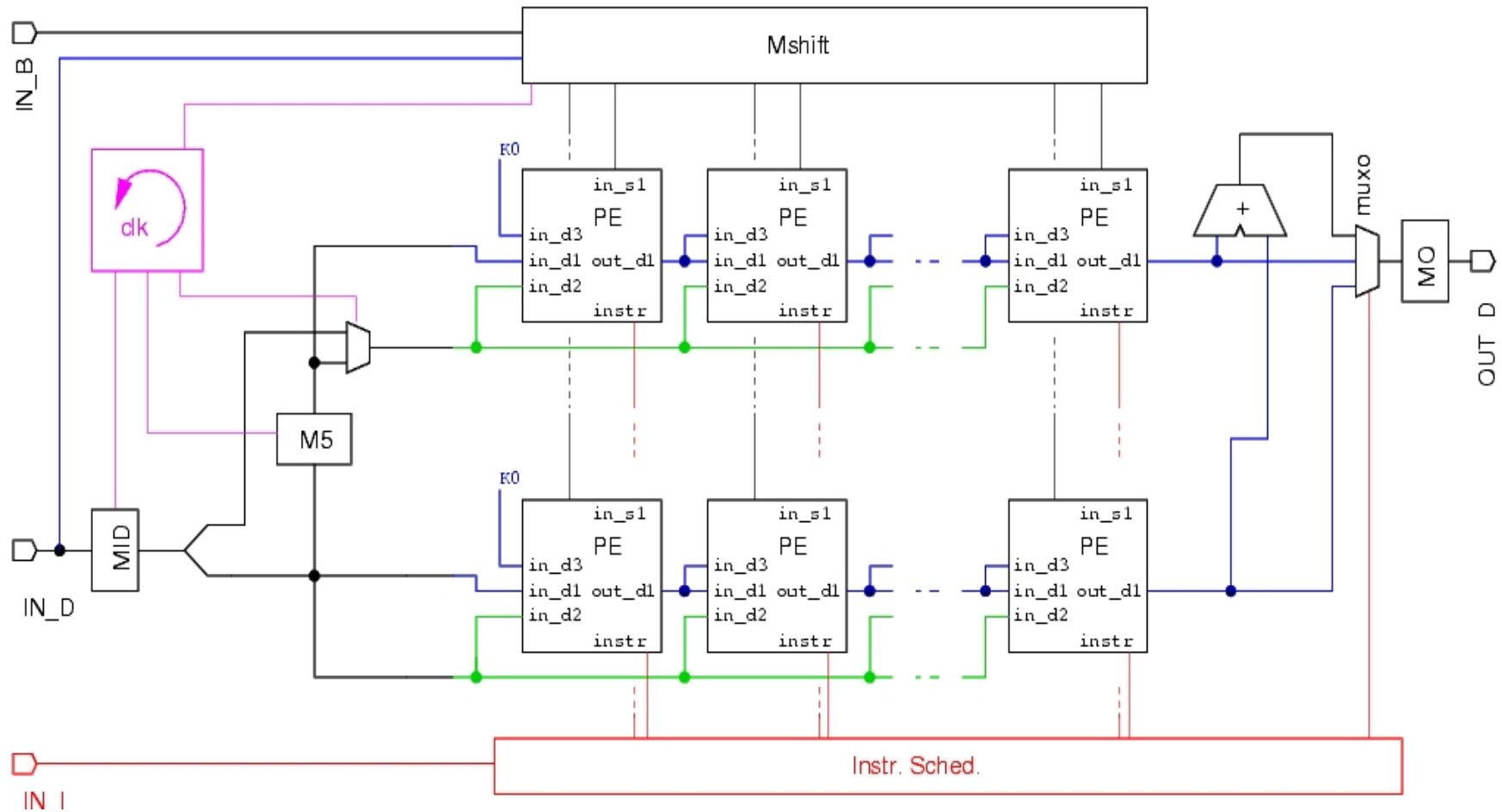
Vorarbeiten

ENTWURFSFLUSS FÜR HARDWARE-SOFTWARE SYSTEM



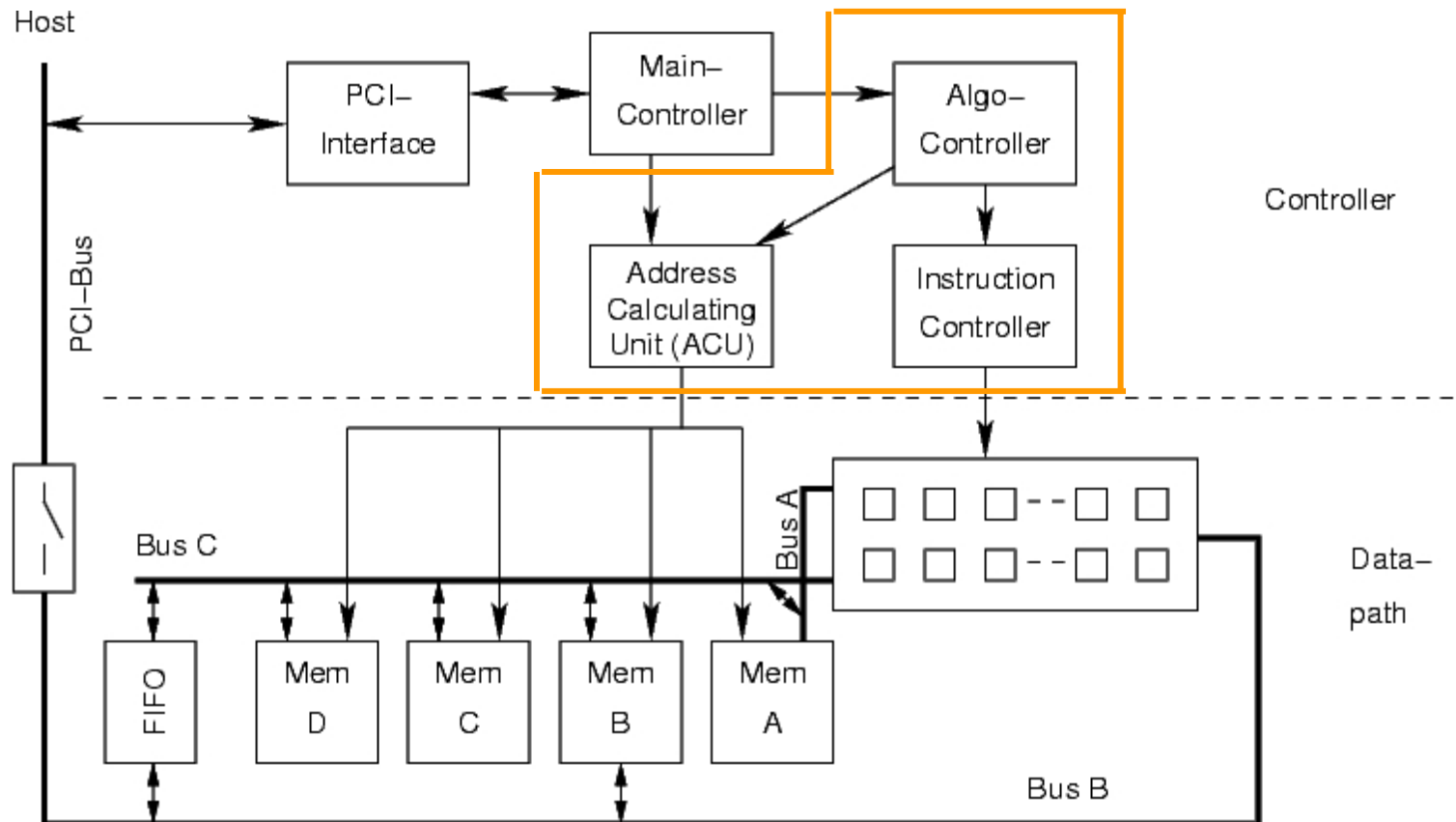
Vorarbeiten

Konfigurierbares Prozessor Array



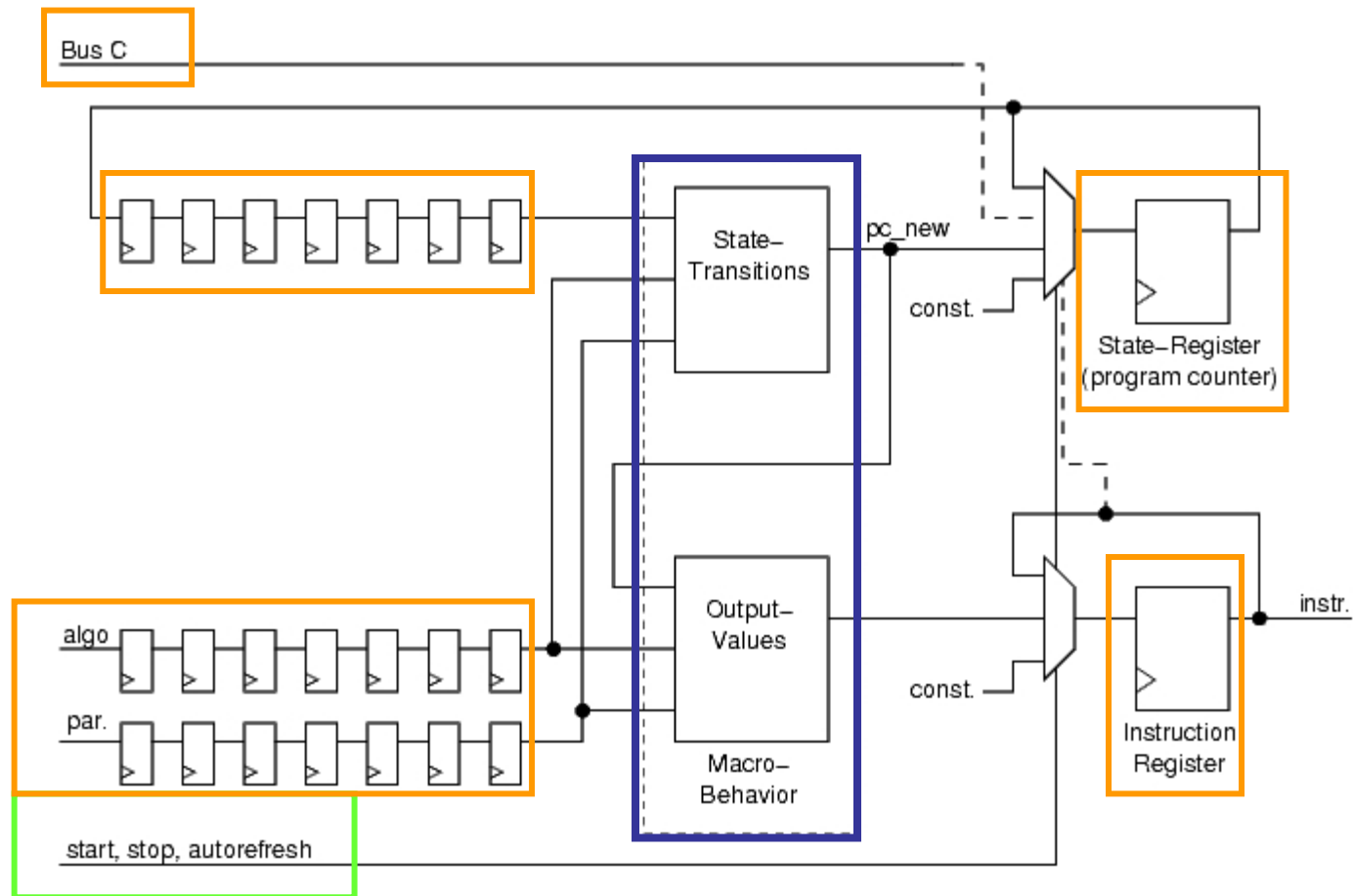
Vorarbeiten

Gesamtssystem



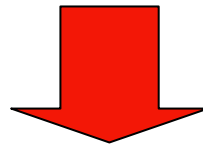
Vorarbeiten

Implementation Steuerfluss: Algo- Controller

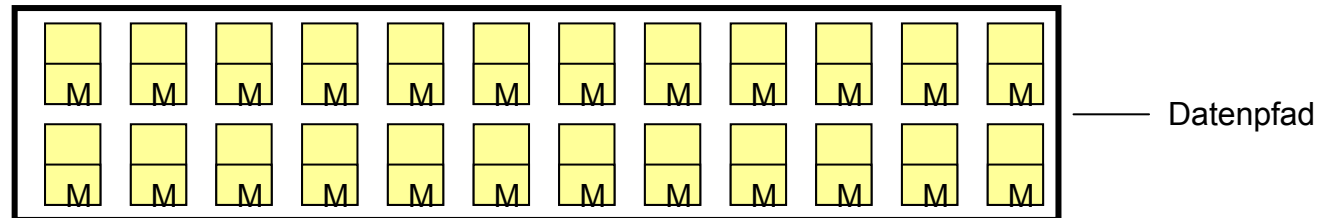


Motivation für SPP 1148

- Entwurfskonzept vorhanden
- bisherige Plattform jedoch zu applikationsspezifisch
Array-Struktur auf Tomographiealgorithmen zugeschnitten
hoher Steueraufwand für Implementierung weiterer Algorithmen



Verallgemeinerung des Entwurfskonzeptes für dynamisch rekonfigurierbare Plattformen



Datenpfad: nutzbar für eine Vielzahl von Algorithmen
Zuschnitt des Datenpfades auf Algorithmus mit Controller

Aufgabenstellung für SPP 1148

Beitrag zur Synthese dynamisch rekonfigurierbarer Systeme auf FPGA-Plattformen

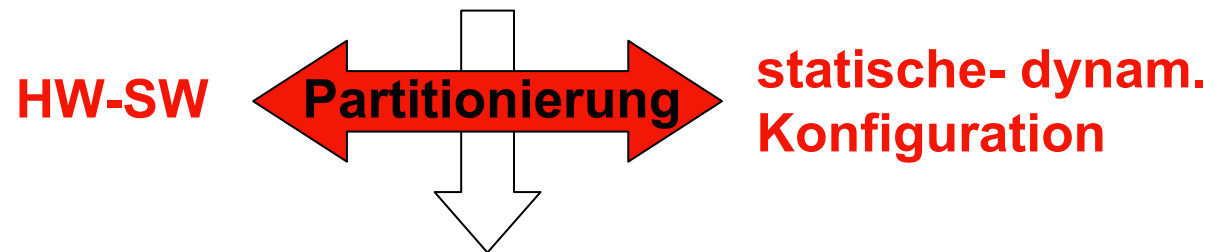
1. Entwicklung von Partitionierungs-Strategien (HW-SW; statisch-dynamisch)
2. Integration der Partitionierungs-Strategien in Entwurfsmethoden

Aufgabenstellung

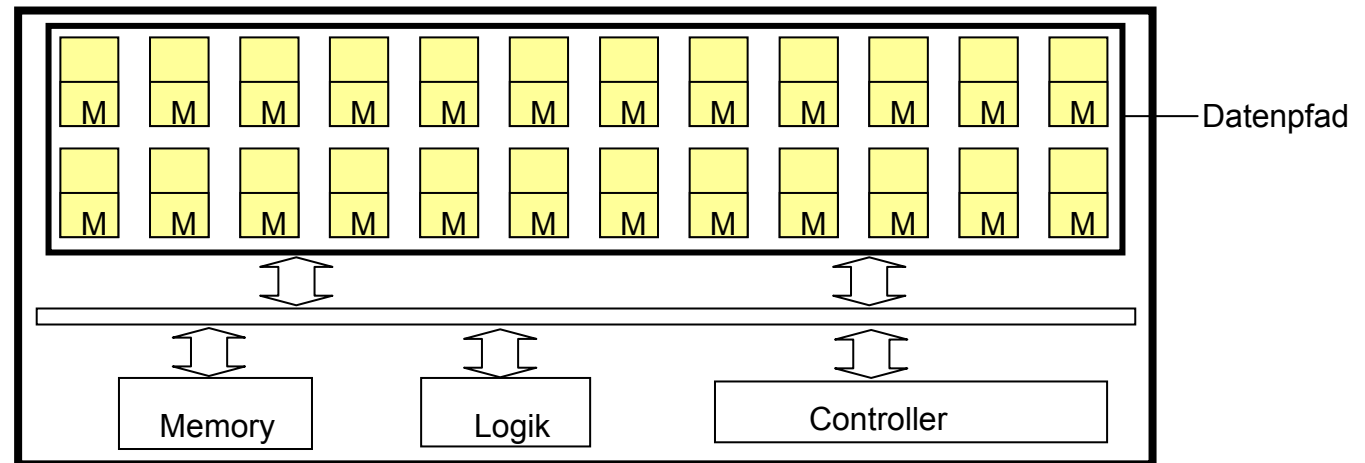
1. Partitionierungsstrategien

Anwendungen: Video, Audio, drahtlose Kommunikation u.a.

Ziel:
Effiziente
Realisierung
einer
Algorithmen-
Vielfalt

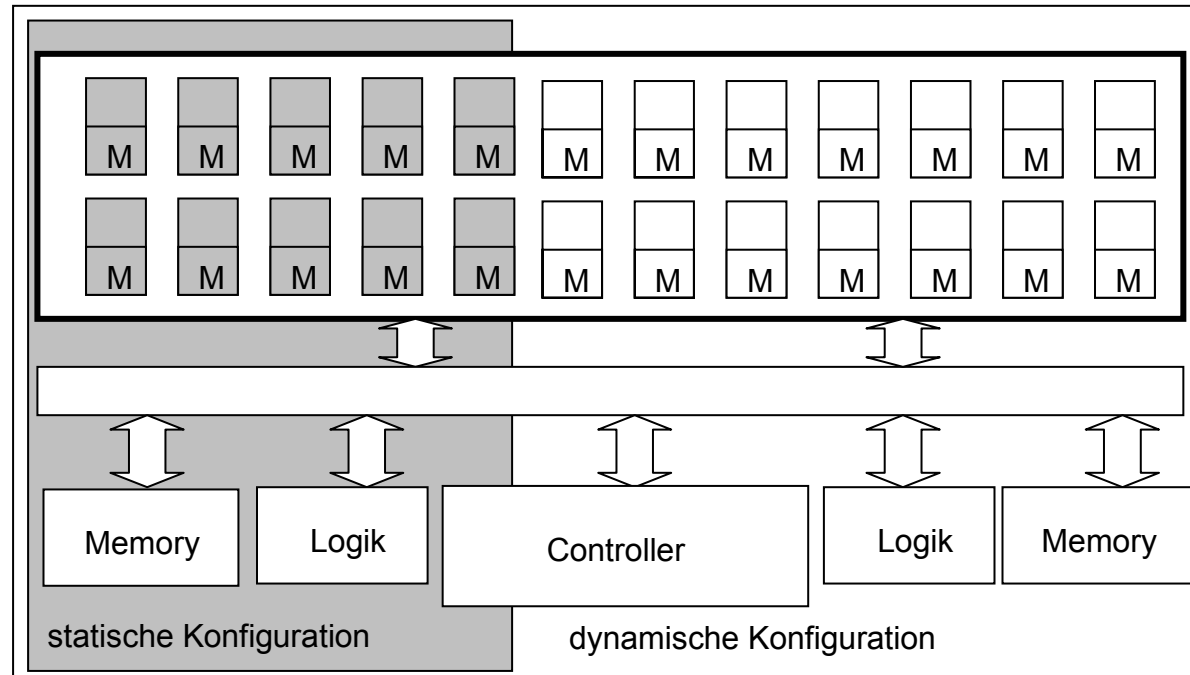


Architektur: FPGA-basierte Plattform



Aufgabenstellung

Variante I:



- Datenpfad (Pipeline/Array-Architektur): Ausdehnung dynamisch veränderbar
- Controller für Datenpfad und Adressrechnung (Zuschnitt auf Algorithmus): statisch und dynamisch rekonfigurierbar
- Algorithmus-spezifische Logik: dynamisch rekonfigurierbar

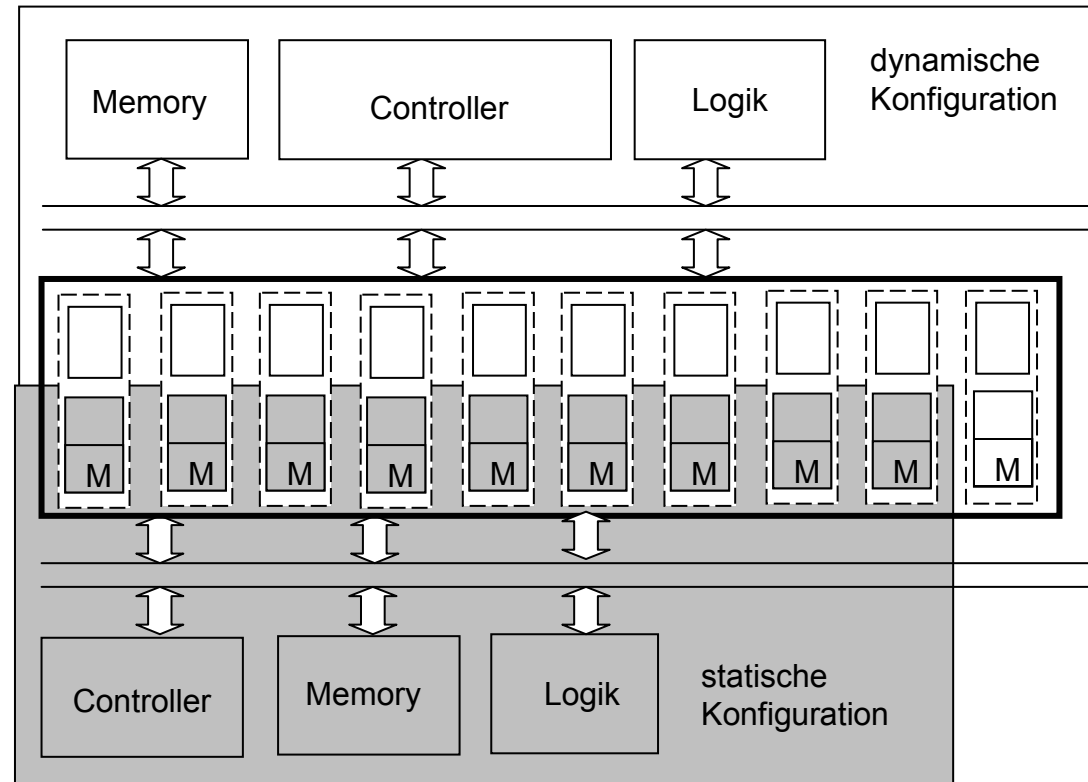
Aufgabenstellung

Aufgabenstellungen für Variante I:

- Algorithmenanalyse, Extraktion einer allg. Pipeline-/ Array-Struktur mit „universellen“ Prozessoren
- Systematischer Entwurf eines Controllers:
Extraktion eines Controller-Kerns (statischer Teil- RAM)
HW/SW-Teilung des dyn. rekonf. Teils des Controllers
- Speicherorganisation unter dem Aspekt der Optimierung der Zugriffe (Leistungsverbrauch, Timing)

Aufgabenstellung

Variante II:



- Datenpfad (Pipeline-/Array-Architektur):
Array- Ausdehnung & ergänzende Prozessor- Funktionalität dynamisch rekonfigurierbar

Aufgabenstellung

Aufgabenstellung zu Variante II:

- Algorithmenanalyse, Extraktion einer Pipeline-/Array-Architektur mit „minimaler“ Prozessor-Funktionalität
- Strategien zur HW/SW-Partitionierung der dynamischen rekonfigurierbaren Anteile in den Prozessoren
- Strategien zur Controller-Partitionierung in statischen und dynamischen Teil
- Speicherorganisation unter dem Aspekt der Optimierung der Zugriffe (Leistungsverbrauch, Timing)

Aufgabenstellung

Bewertung der Varianten bezüglich

Kriterien:

- Fläche, Durchsatz, Latenz
- Rekonfigurationskosten
- Leistungsverbrauch durch Speicherzugriffe

Entwicklung einer Kosten-Nutzenanalyse

Vergleich mit anderen Realisierungsformen (z.B. DSPs)

Aufgabenstellung

2. Entwurfsmethoden

Entwicklung von Werkzeugen zur Partitionierung in statische und dynamisch rekonfig. Anteile mit Ziel:

- Maximale Auslastung des statischen Teils
- Minimale Rekonfigurationskosten für dynamischen Teil
- Minimierung der Zugriffe auf Speicherhierarchie (Leistungsverbrauch)

Ziel: Entwurfsfluss

