Diplomverteidigung

Untersuchungen zur funktionalen Effizienz von Kontrollflussbasierter Rekonfiguration bei RISC-Prozessoren

Martin Zabel
zabel@ite.inf.tu-dresden.de

Institut für Technische Informatik

Inhalt

- Einführung
- Grobgranular rekonfigurierbare Systeme
- Funktionale Flexibilität grobgranularer Strukturen
- Flexibilität kontra Komplexität anhand der ARRIVE-Architektur
- Effizienz der ARRIVE-Architektur
- Zusammenfassung und Ausblick

Martin Zabel
Institut für Technische Informatik
zabel@ite.inf.tu-dresden.de

Aufgabenstellung

- Anzahl und Komplexität der Verarbeitungseinheiten von RISC-Prozessoren und DSPs
- Auslastungsgrad der Verarbeitungseinheiten anhand von Referenz-Algorithmen
- Zweidimensionale rekonfigurierbare Arithmetikstrukturen sowie rekonfigurierbare Datenpfade
- Abhängigkeit zwischen der funktionalen Flexibilität einer grobgranular rekonfigurierbaren Verarbeitungseinheit und deren Chipflächenbedarf
- Experimentelle Untersuchung dieser Abhängigkeit am Modell der kontrollflussbasierten Rekonfiguration der ARRIVE-Architektur
- Nachweis der erzielten Effizienz von ARRIVE anhand von Simulationen
- Dokumentation der Ergebnisse

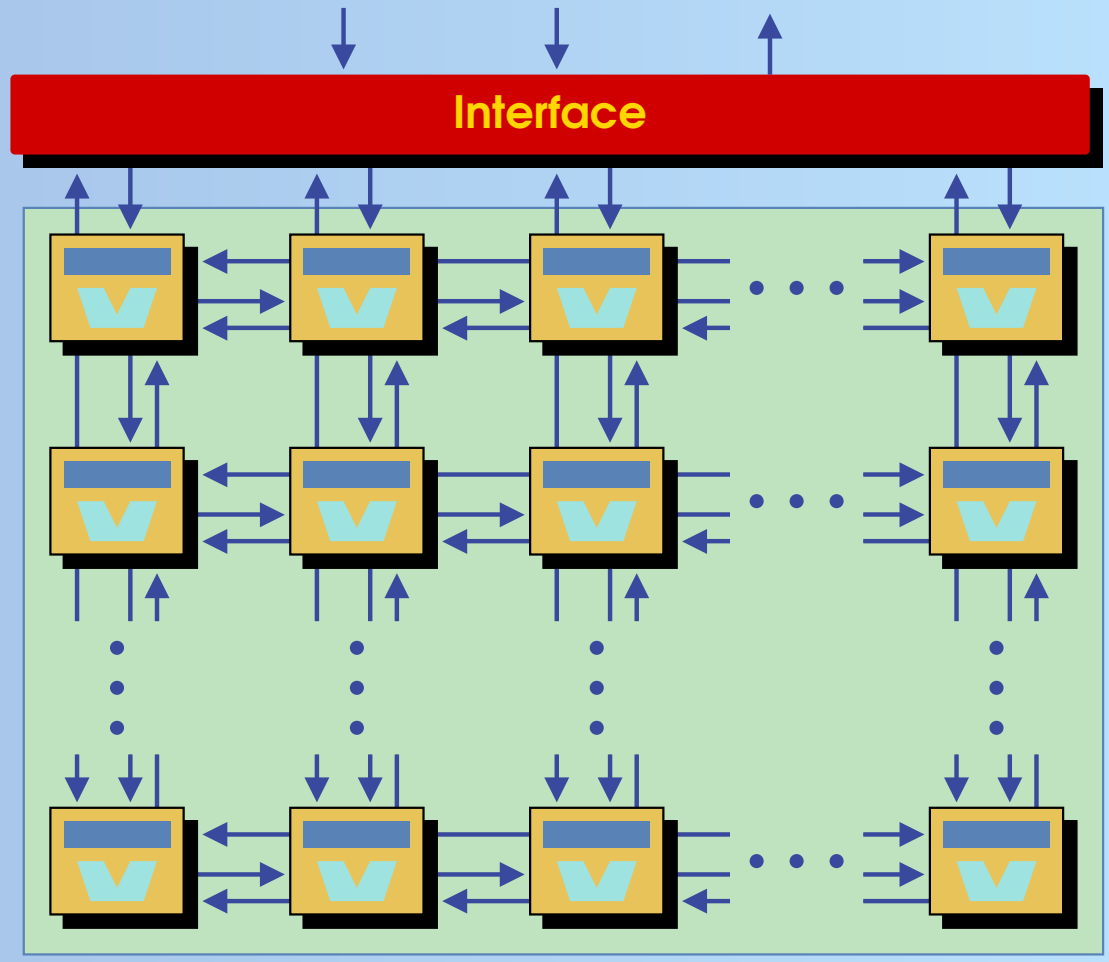
Martin Zabel
Institut für Technische Informatik
zabel@ite.inf.tu-dresden.de

Motivation

- *Ziel*: hoher Durchsatz an Operationen
- Applikationsspezifische Schaltungen
- VLIW- und SIMD-Architekturen
- Universalprozessoren
- Grobgranulare Rekonfiguration
  - Hoher Grad an Verarbeitungsparallelität
  - Niedriger Stromverbrauch
  - Hohe Flexibilität
  - Sehr gute Auslastung für viele Algorithmen

Martin Zabel
Institut für Technische Informatik
zabel@ite.inf.tu-dresden.de

Schema



Martin Zabel
Institut für Technische Informatik
zabel@ite.inf.tu-dresden.de

Überblick

Name	Granularität	Arithmetik	Datenpfade	Integration
ARRIVE	multigranular	ALU, S, M	Crossbar	CPU
CHESS	4 Bit, multigranular	ALU	NN, Bus	I/O
COLT	16 Bit	ALU, S, M	NN, Bus, Crossbar	I/O
DRA		ALU, S, M	Bus	Coprozessor
DReAM	8 und 16 bit	ALU, S, M, speziell	NN, Bus	I/O
MATRIX	8 Bit, multigranular	ALU, S, M, PLA	NN, Bus	
MONTIUM	16 und 32 Bit	ALU, M	Crossbar	
PADDI-2	16 Bit	ALU, S, Booth	NN, Crossbar	I/O
PipeRench	multigranular	ALU, S, LUT	Crossbar	
RAW	multigranular	ALU, S, M, LUT	Bus, dynamisch	
REMARC	16 Bit	ALU, S	NN, Bus	Coprozessor
Smart Memories	64 Bit	Gleitkomma, ALU, S, M	Crossbar, dynamisch	
...				

Martin Zabel
Institut für Technische Informatik
zabel@ite.inf.tu-dresden.de

Flexibilität Rechenwerk

- Funktionale Flexibilität
  - *Ziel:* konkreter Zahlenwert
  - Anzahl echt verschiedener Funktionen
- Rechenwerk: Basisfunktionen
  - Logische Funktion: NAND
  - Verschiebung nach links
  - Verschiebung nach rechts

  $\rightarrow$ alle Funktionen zusammensetzbar

Martin Zabel
Institut für Technische Informatik
zabel@ite.inf.tu-dresden.de

Flexibilität des Datenflusses I

- Ein Prozessorelement
  - Gleichberechtigte Eingänge: $a + b = b + a$
  - Kombinationen ohne Wiederholung

$$F_{inp}(N_v, N_o) = \begin{cases} \binom{N_v}{N_o} & : \quad N_v \geq N_o \\ 0 & : \quad N_v < N_o \end{cases}$$

Martin Zabel
Institut für Technische Informatik
zabel@ite.inf.tu-dresden.de

Flexibilität des Datenflusses I

- Ein Prozessorelement
  - Gleichberechtigte Eingänge: $a + b = b + a$
  - Kombinationen ohne Wiederholung

$$F_{inp}(N_v, N_o) = \begin{cases} \binom{N_v}{N_o} & : \quad N_v \geq N_o \\ 0 & : \quad N_v < N_o \end{cases}$$

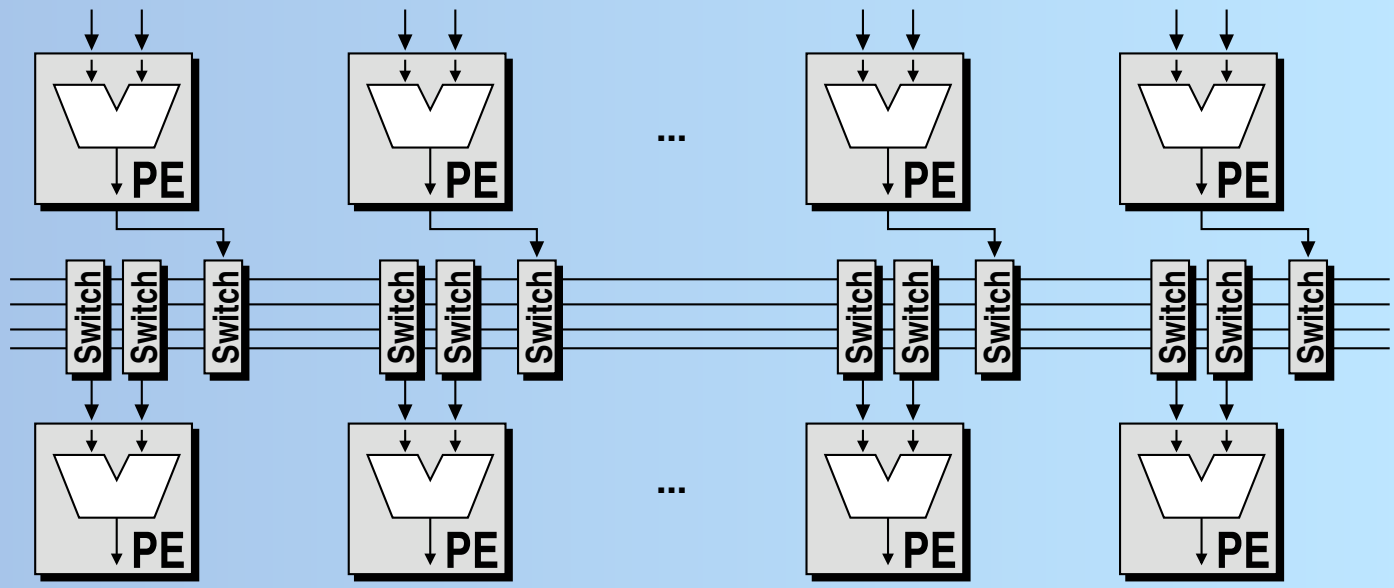
- Crossbar und direkte Verbindungen

$$F_{crossbar} = (F_{inp}(N_v, N_o))^{N_p}$$

Martin Zabel
Institut für Technische Informatik
zabel@ite.inf.tu-dresden.de

Flexibilität des Datenflusses II

- Bussysteme



  - N_v Variablen ($\widehat{=}$ schreibenden PEs)
  - N_c Kanäle
  - N_p Prozessorelemente ($\widehat{=}$ lesenden PEs)
- Abbildungen ohne Beschränkung: $Abb(N_v, N_p) = (F_{inp}(N_v, 2))^{N_p}$

Martin Zabel
Institut für Technische Informatik
zabel@ite.inf.tu-dresden.de

Flexibilität des Datenflusses III

- Bussysteme
  - Abbildungen ohne Kanalbeschränkung: $Abb(N_v, x_1, \ldots, x_n)$
  - Bestehend aus surjektiven Abbildungen:

$$Abb(N_v, x_1, \ldots, x_n) = \sum_{k=0}^{N_v} \binom{N_v}{k} Sur(k, x_1, \ldots, x_n)$$

$$Sur(N_v, x_1, \ldots, x_n) = \sum_{k=0}^{N_v} (-1)^k \binom{N_v}{k} Abb(N_v - k, x_1, \ldots, x_n)$$

  - N_c Kanäle $\widehat{=}$ höchstens N_c verschiedene Variablen

$$F_{bus}(N_v, N_c, x_1, \ldots, x_n) = \sum_{k=0}^{N_c} \binom{N_v}{k} Sur(k, x_1, \ldots, x_n)$$

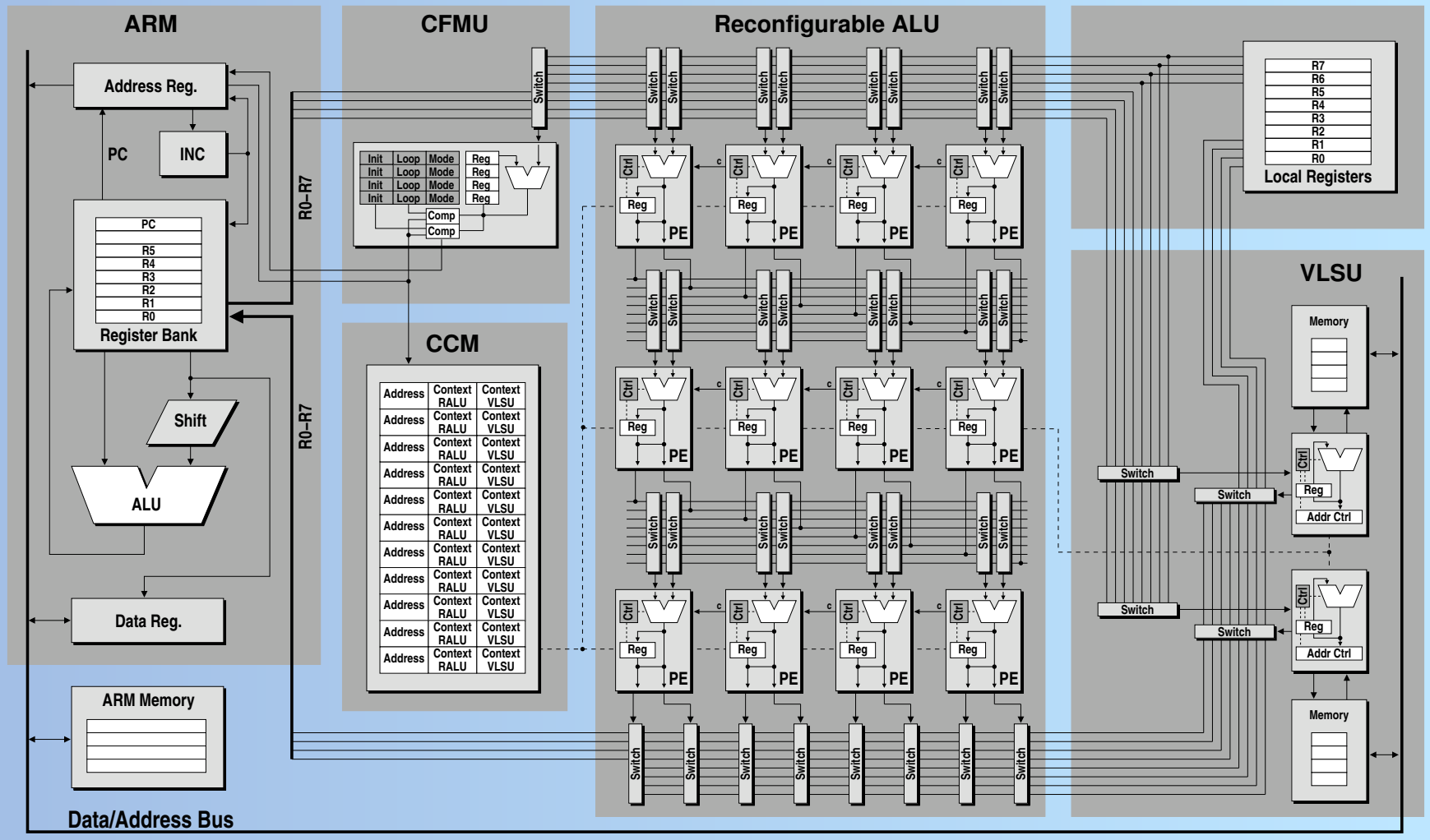
Martin Zabel
Institut für Technische Informatik
zabel@ite.inf.tu-dresden.de

Flexibilität des Steuerwerks

- Anzahl der Kontrollflüsse
  - Konkatenation der Befehlszähler
  - kein Einfluss auf die Flexibilität
- Anzahl der Befehlsströme
  - SIMD: kein Flexibilitätsgewinn gegenüber SISD
  - VLIW: jedes Element unabhängig von anderen

Martin Zabel
Institut für Technische Informatik
zabel@ite.inf.tu-dresden.de

ARRIVE-Modell



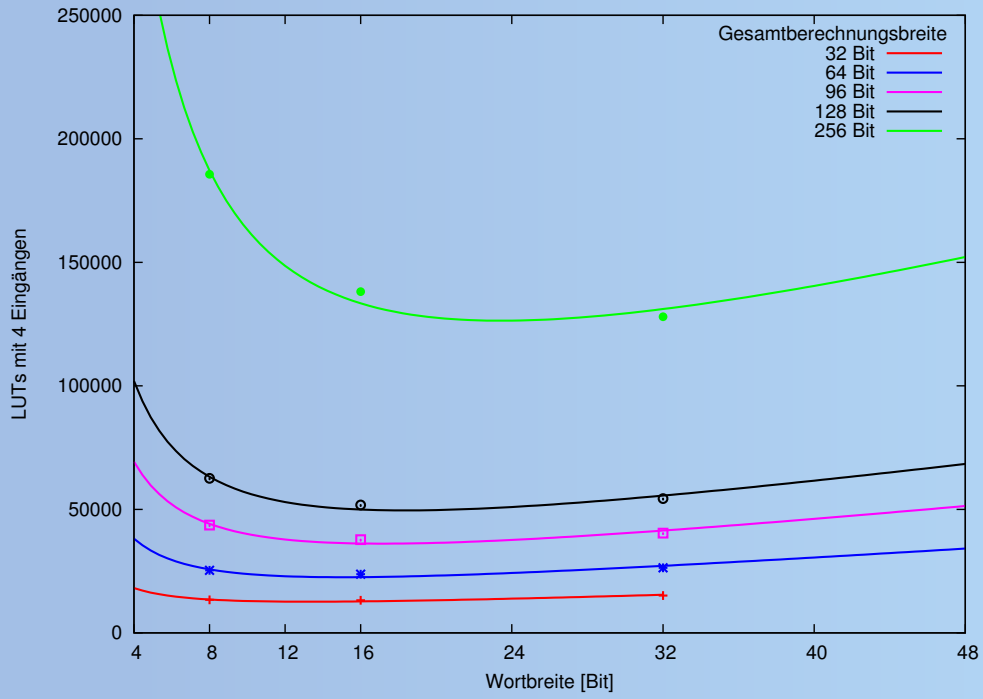
Martin Zabel
Institut für Technische Informatik
zabel@ite.inf.tu-dresden.de

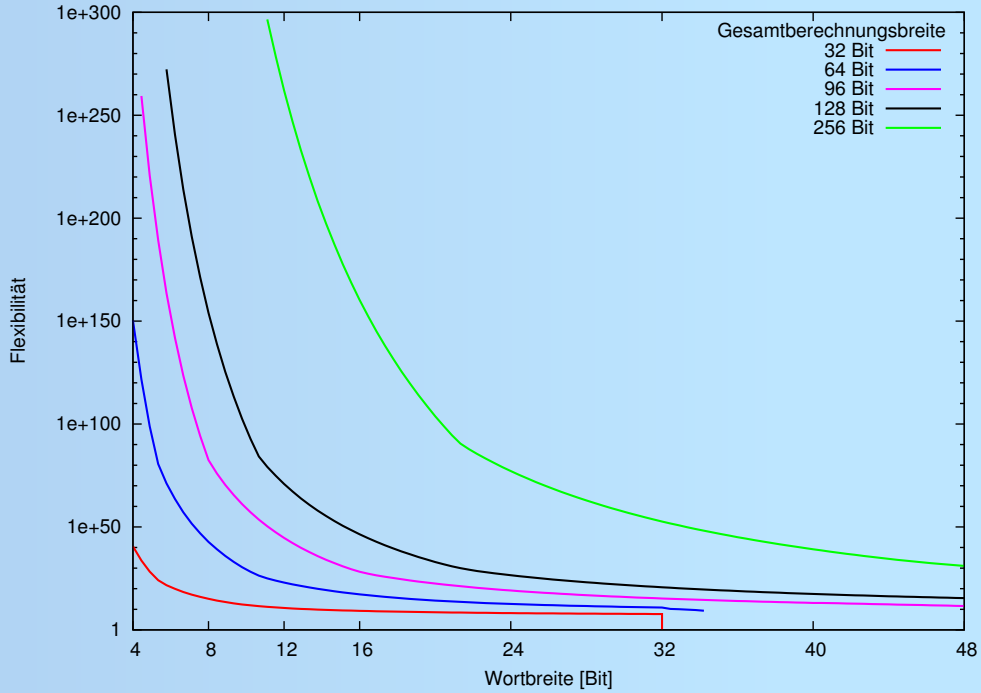
Voraussetzungen

- ARRIVE-Modell aus meiner Belegarbeit
- inklusive Erweiterungen/Modifikationen
- Variationen
  - Wortbreite
  - Spaltenanzahl
  - Zeilenanzahl
  - Registeranzahl
  - Datenbusbreite
- Komplexität: LUT
- Flexibilität: ändert sich bzgl. Datenfluss

Martin Zabel
Institut für Technische Informatik
zabel@ite.inf.tu-dresden.de

Variation der Wortbreite

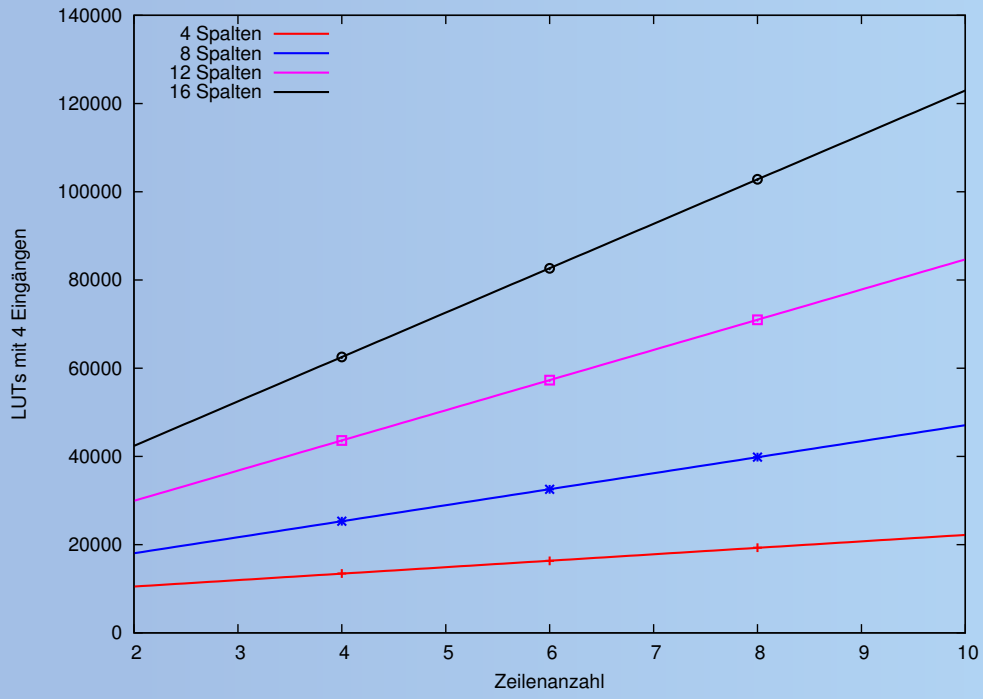


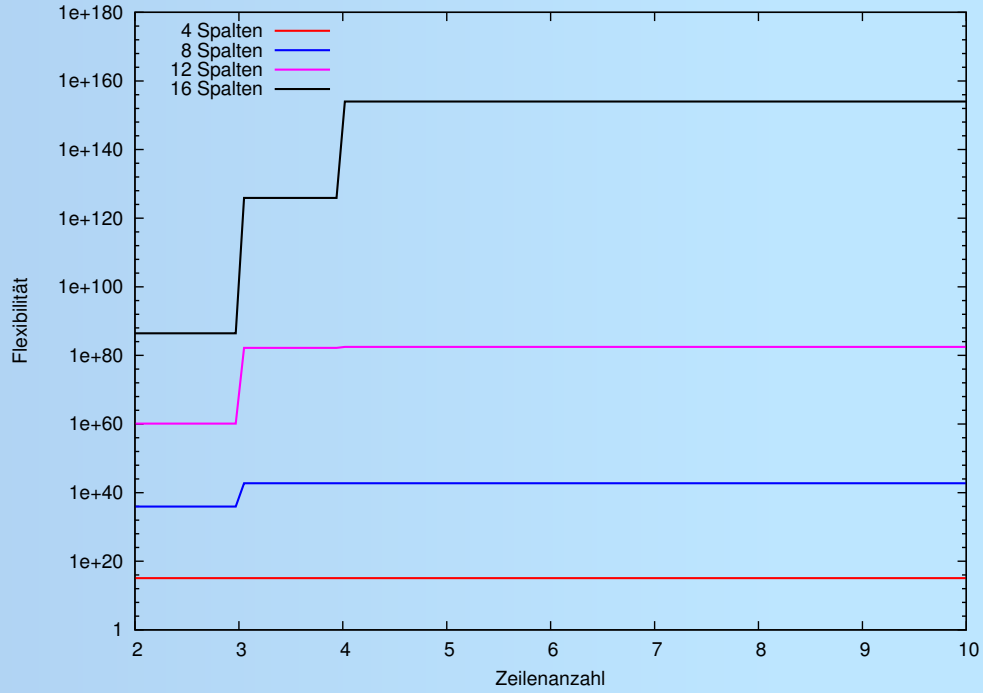


- Minimum der Komplexität zwischen 12 und 24 Bit

Martin Zabel
Institut für Technische Informatik
zabel@ite.inf.tu-dresden.de

Variation der Zeilenanzahl





- Flexibilitätsgewinn nur bis zur Zeile $r_{max} = 1 + \lceil \log_2 N_{col} \rceil$

Martin Zabel
Institut für Technische Informatik
zabel@ite.inf.tu-dresden.de

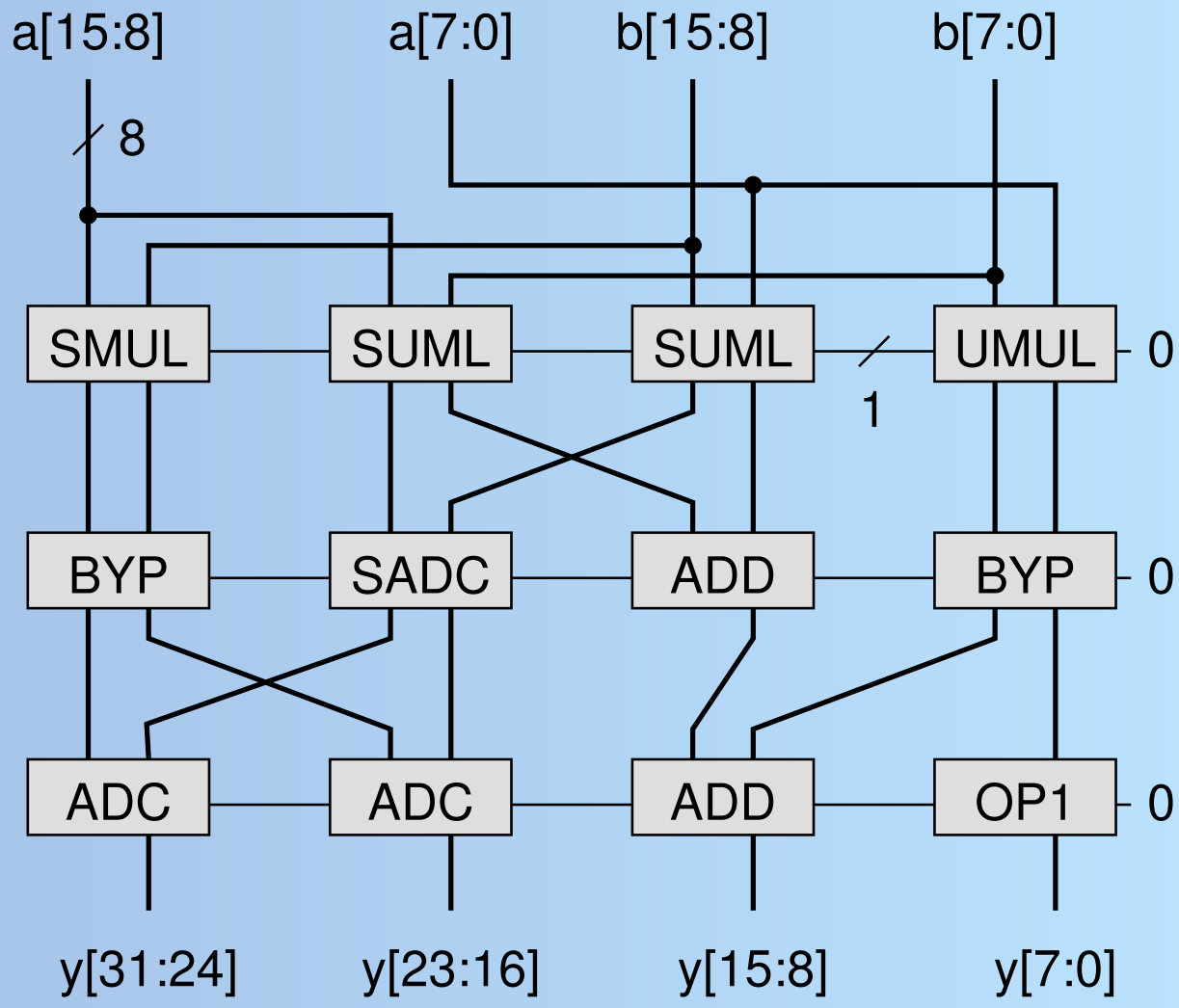
Algorithmen

- ARRIVE mit 64 8-Bit-PEs in Clustern zu 4×4 PEs
- Schnelle Fourier-Transformation
- Viterbi-Dekoder
- Diskrete Cosinus-Transformation
- FIR-Filter
- IIR-Filter

Addr.	ARM7	RALU	CFMU	VLSU1	VLSU2
0x1100			Init L0		
0x1104				LD y0(n-1,n-2)	LD b1(1) b2(1)
0x1108		MUL/MUL/ADD	Init L1	LD x(n)	LD b0(1)
0x110C		MUL		LD yi(n-1,n-2)	LD a1(i) a2(i)
0x1110	ADD	MUL/MUL/ADD		ST yi1(n-1,n-2)	LD b1(i+1) b2(i+1)
0x1114	SUB	MUL/MUL/ADD	Ende L1		LD b0(i+1)
0x1118		MUL/ADD	Ende L0	LD ys(n-1,n-2)	LD a1(s) a2(s)

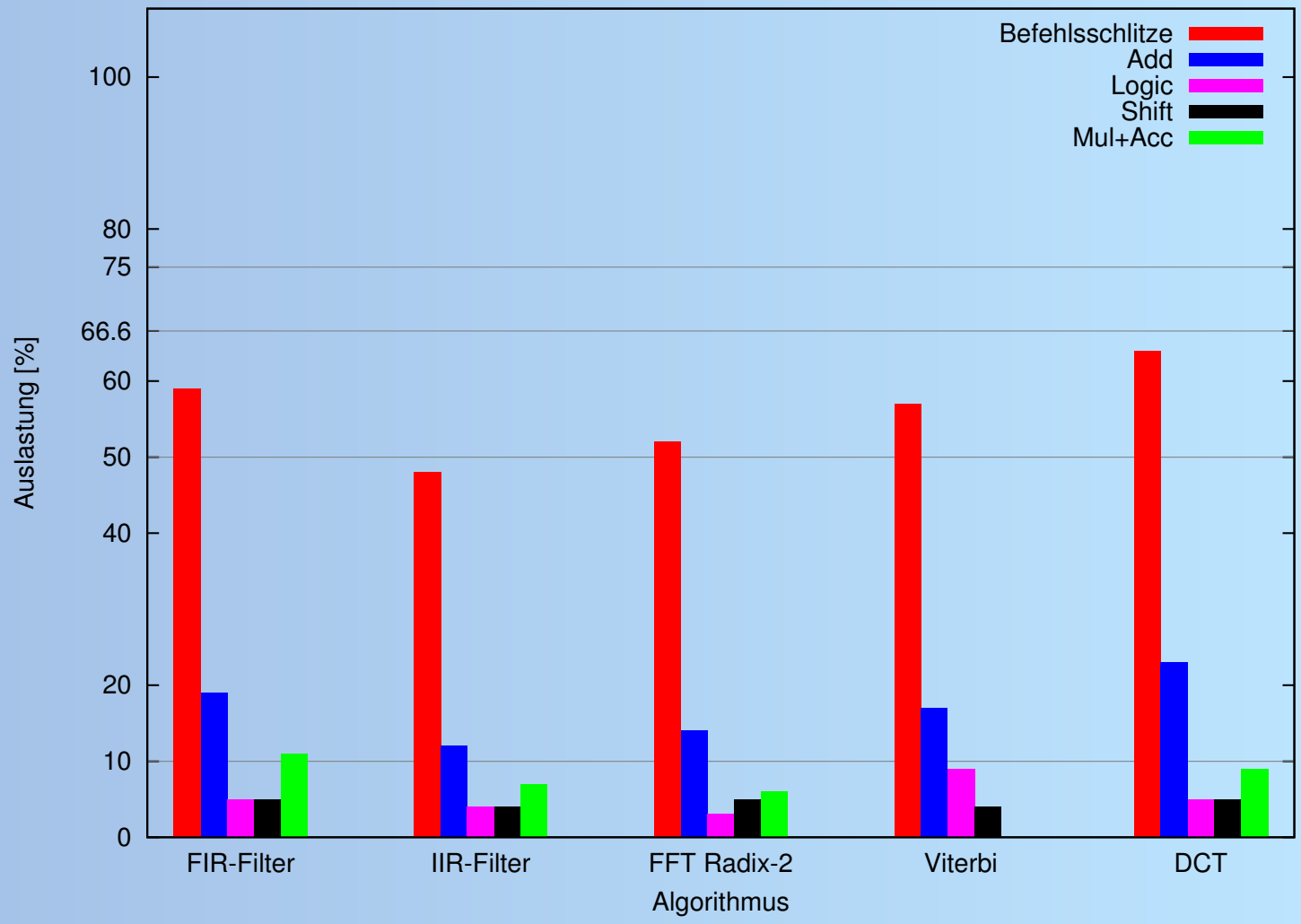
Martin Zabel
Institut für Technische Informatik
zabel@ite.inf.tu-dresden.de

Kontext



Martin Zabel
Institut für Technische Informatik
zabel@ite.inf.tu-dresden.de

Auslastung



Martin Zabel
Institut für Technische Informatik
zabel@ite.inf.tu-dresden.de

Zusammenfassung und Ausblick

- Zusammenfassung
  - Komplexität RISC- und DSP-Architekturen
  - Analyse der Benchmark-Algorithmen hinsichtlich Auslastung
  - Aufbau grobgranular rekonfigurierbarer Architekturen
  - Flexibilität kontra Komplexität der grobgranular rekonfigurierbaren Strukturen
  - Flexibilität/Komplexität anhand der ARRIVE-Architektur
  - Auslastung von ARRIVE bei der Ausführung von Benchmark-Algorithmen
- Ausblick
  - Flexibilität des Rechenwerks als Zahlenwert
  - Optimierung der ARRIVE-Architektur

Martin Zabel
Institut für Technische Informatik
zabel@ite.inf.tu-dresden.de