Konzeption und Realisierung von Test- und Debugtechniken zur Prototypevaluation der grobgranular-rekonfigurierbaren ARRIVE-Architektur

Diplomverteidigung

Martin Zimmerling
mz793134@inf.tu-dresden.de

Institut für Technische Informatik

Gliederung

- Aufgabenstellung
- Debugging und Prototyping
- ARRIVE-Architektur
- Konzept und Realisierung
- Ergebnisse

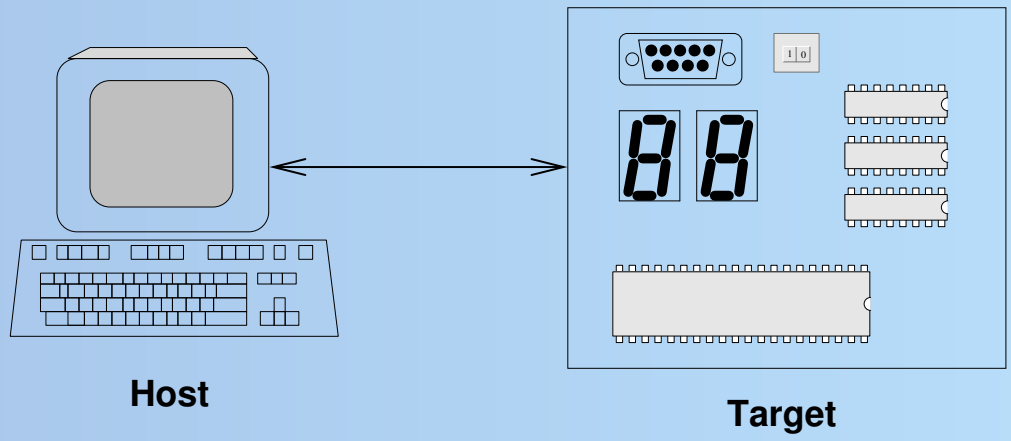
Martin Zimmerling
Institut für Technische Informatik
mz793134@inf.tu-dresden.de

Aufgabe

- Implementationsmöglichkeiten aufzeigen zur Inbetriebnahme, Test und Debugging anhand des VHDL-Modells der ARRIVE-Architektur
- Prototyp-Realisierung auf Basis eines FPGAs
- Visualisierung mittels einer Test- und Debug-Umgebung
- Evaluation bzgl. benötigter HW-Ressourcen und Taktfrequenz
- Vergleich zu Standardzellen-ASIC-Synthese

Martin Zimmerling
Institut für Technische Informatik
mz793134@inf.tu-dresden.de

Debugging I (Ziele)



- Steuerung
- Beobachten

$\Rightarrow$ Finden von Fehlern

Martin Zimmerling
Institut für Technische Informatik
mz793134@inf.tu-dresden.de

Debugging II (Anforderungen)

- Kontrolle des Steuerflusses
  - Starten, Stoppen
  - Singlestepping
  - Befehlszähler setzen
- Kontrolle des internen Zustands
  - Schreiben/Lesen der Register
- Rekonfigurierbarer Prozessor $\Rightarrow$ Konfigurationstabellen
  - Schreiben der Tabellen
  - Lesen der Tabellen (teuer und unnötig!)
- Prototyp: System-on-Chip $\Rightarrow$ Speicher
  - Schreiben/Lesen des Speichers

Anforderungen an das Prototyping

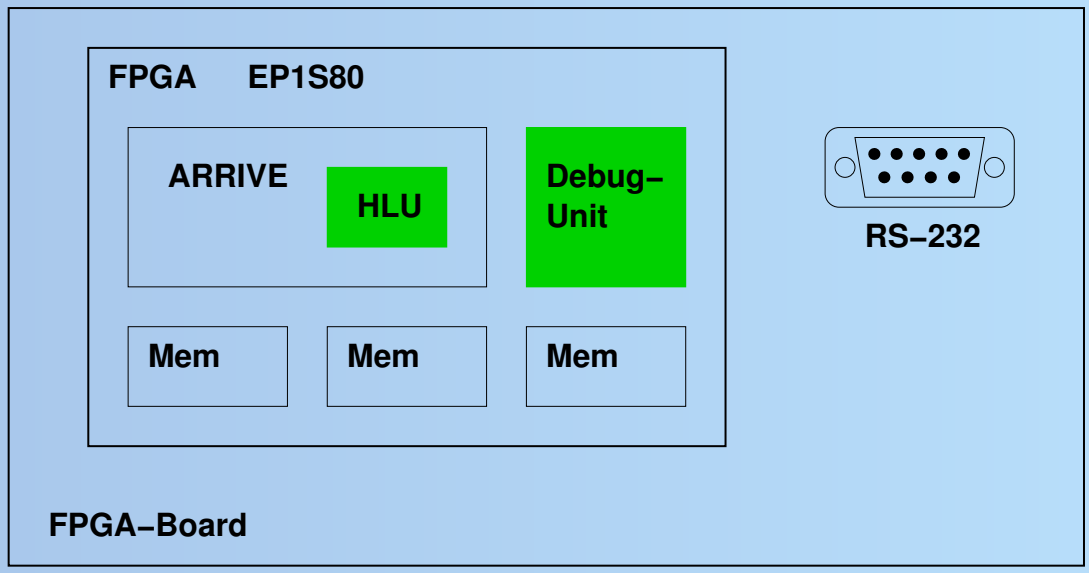
- Flexibilität
- Skalierbarkeit
- Beobachtbarkeit

Martin Zimmerling
Institut für Technische Informatik
mz793134@inf.tu-dresden.de

ARRIVE: Rekonfigurierbare Architekturen

- grobgranular-rekonfigurierbare Architekturen vs. FPGA, ASIC, Standardprozessor
- Rekonfiguration in jedem Takt durch Befehlszähler
- vereint Vorteile der anderen Ansätze:
  - schnelle Implementierungszeit
  - gute Ausnutzung der HW-Ressourcen
  - schnelle und flexible Anpassungsfähigkeit
  - hohe Geschwindigkeit bei geringem Leistungsverbrauch

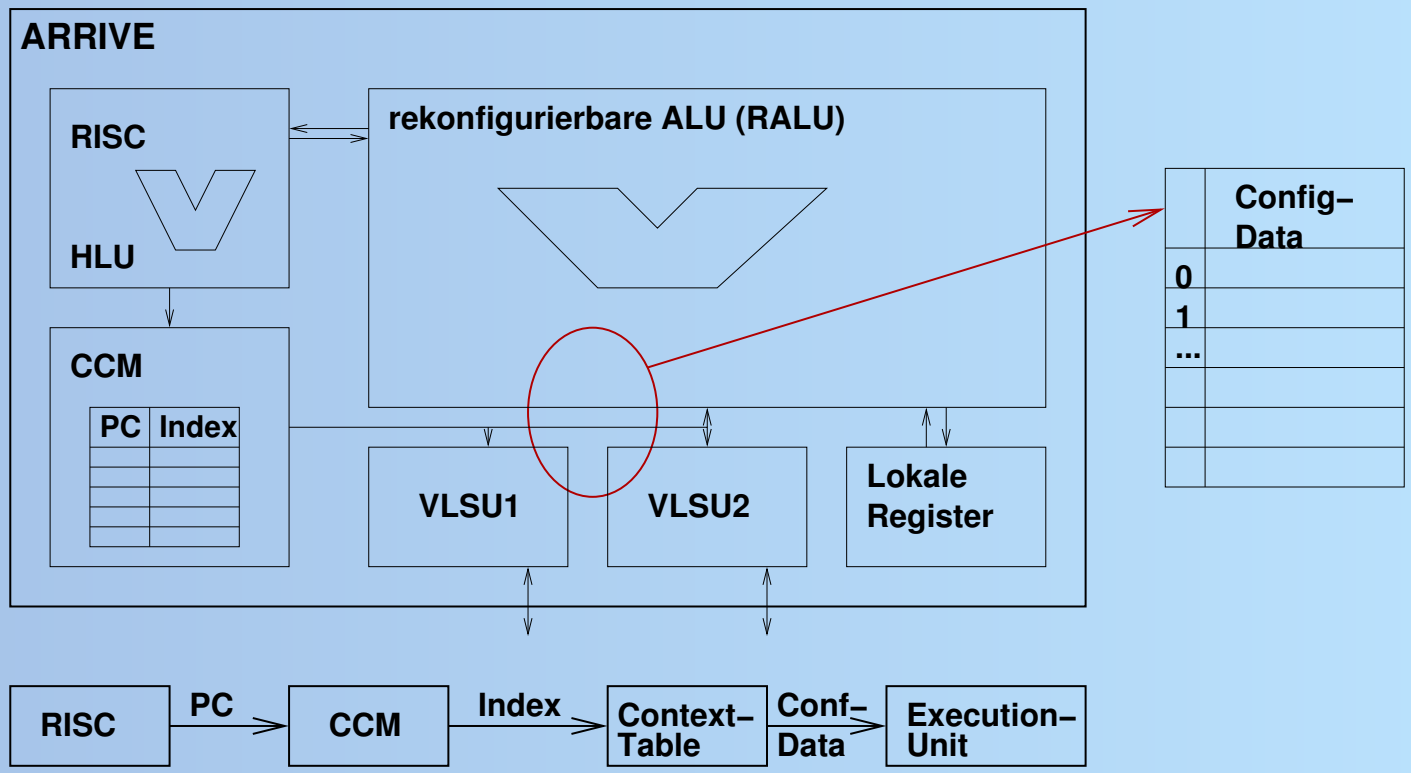
ARRIVE: FPGA-Board



- RS-232 serielle Schnittstelle
- Vorteil: Standard, Betriebssystemunterstützung

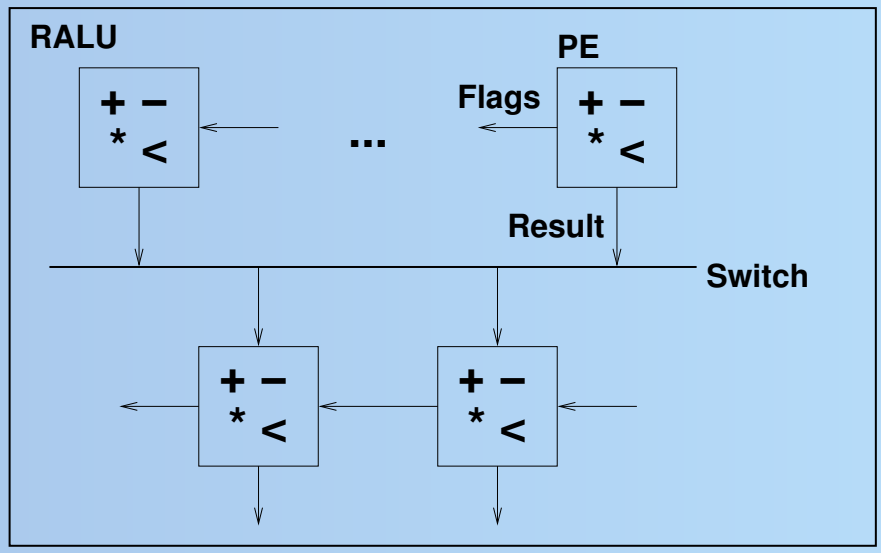
Martin Zimmerling
Institut für Technische Informatik
mz793134@inf.tu-dresden.de

ARRIVE: Architektur



Martin Zimmerling
Institut für Technische Informatik
mz793134@inf.tu-dresden.de

ARRIVE: RALU



- Paramterierbar; zwei typische Varianten
  - 16-Bit PE, 1 Cluster à 4x4 PE $\Rightarrow$ 16 PE
  - 8-Bit PE, 2x2 Cluster à 4x4 PE $\Rightarrow$ 64 PE

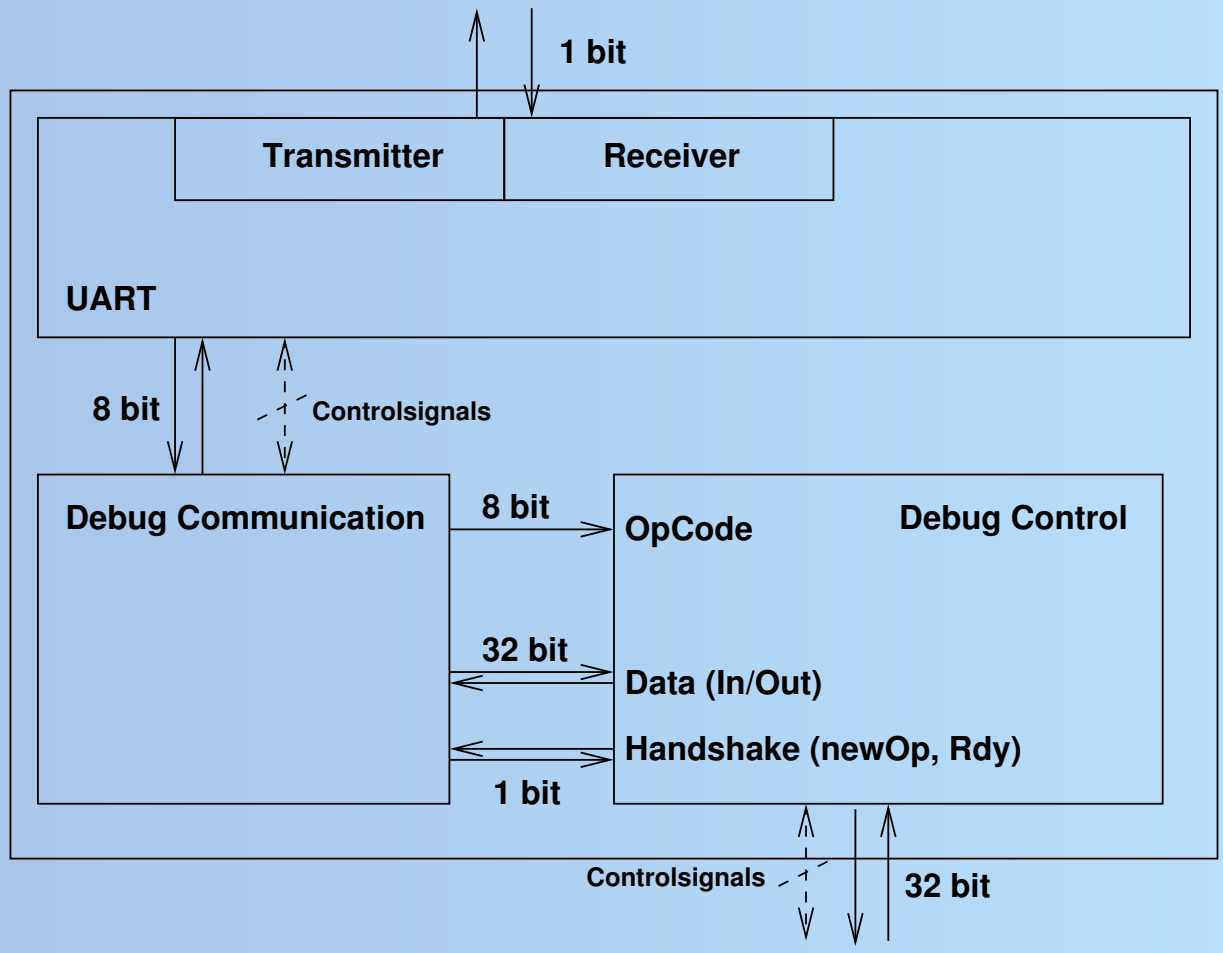
Martin Zimmerling
Institut für Technische Informatik
mz793134@inf.tu-dresden.de

ARRIVE: Ressourcen

- RISC-Prozessor
  - 17 Register (R0 - R15, Flags)
  - Hardware-Loop-Unit
- ARRIVE-Komponenten
  - 8 Lokale Register
  - CCM (32-64 Einträge)
  - VLSU Contexttables (2)
  - Switch-Matrizen Contexttables (4)
  - RALU Contexttables (16 - 64)
  - PE-Register (16 - 64)

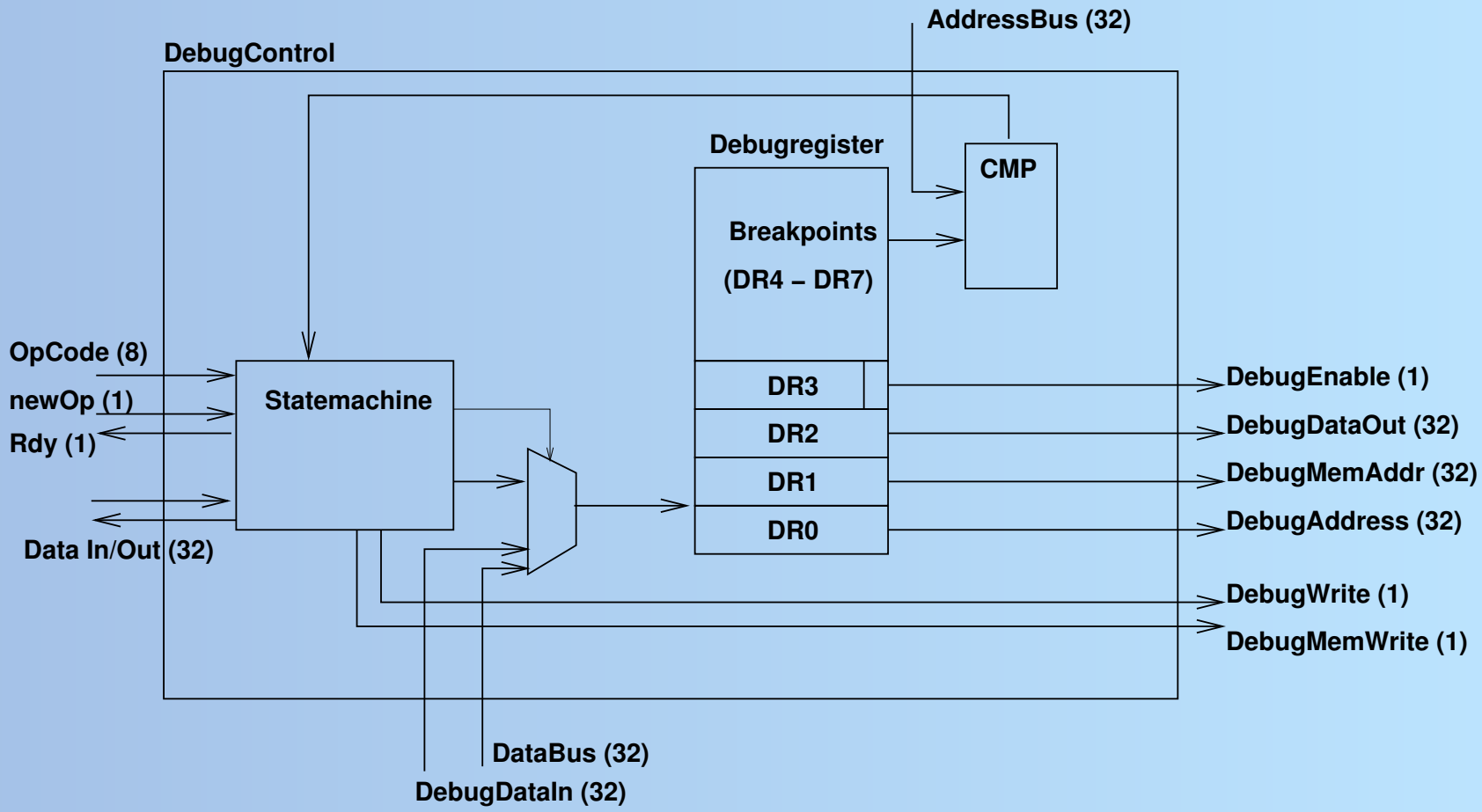
Martin Zimmerling
Institut für Technische Informatik
mz793134@inf.tu-dresden.de

Debugeinheit: Übersicht



Martin Zimmerling
Institut für Technische Informatik
mz793134@inf.tu-dresden.de

Einheit DebugContol



Martin Zimmerling
Institut für Technische Informatik
mz793134@inf.tu-dresden.de

Debugcontrol: Funktionen

- getrennte Adressräume (Speicher, Zustand)
- jeder Adressraum 4G x 32 Bit
- 1 Bit zur Steuerung des Kontrollflusses
- HW-Breakpoints
- Auto-Inkrement der Debugregister

Debugeinheit: Steuerbefehle

Nur 9 Befehle:

Befehlskodierung	Befehl
10001-Reg	SetDebugRegister (+32 bit)
11001-Reg	GetDebugRegister (-32 bit)
001-I-0xxx	WrMem[Inc]
011-I-0xxx	RdMem[Inc]
000-I-0xxx	WrReg[Inc]
010-I-0xxx	RdReg[Inc]
00001-000	STOP
00001-001	STEP
00001-111	RUN

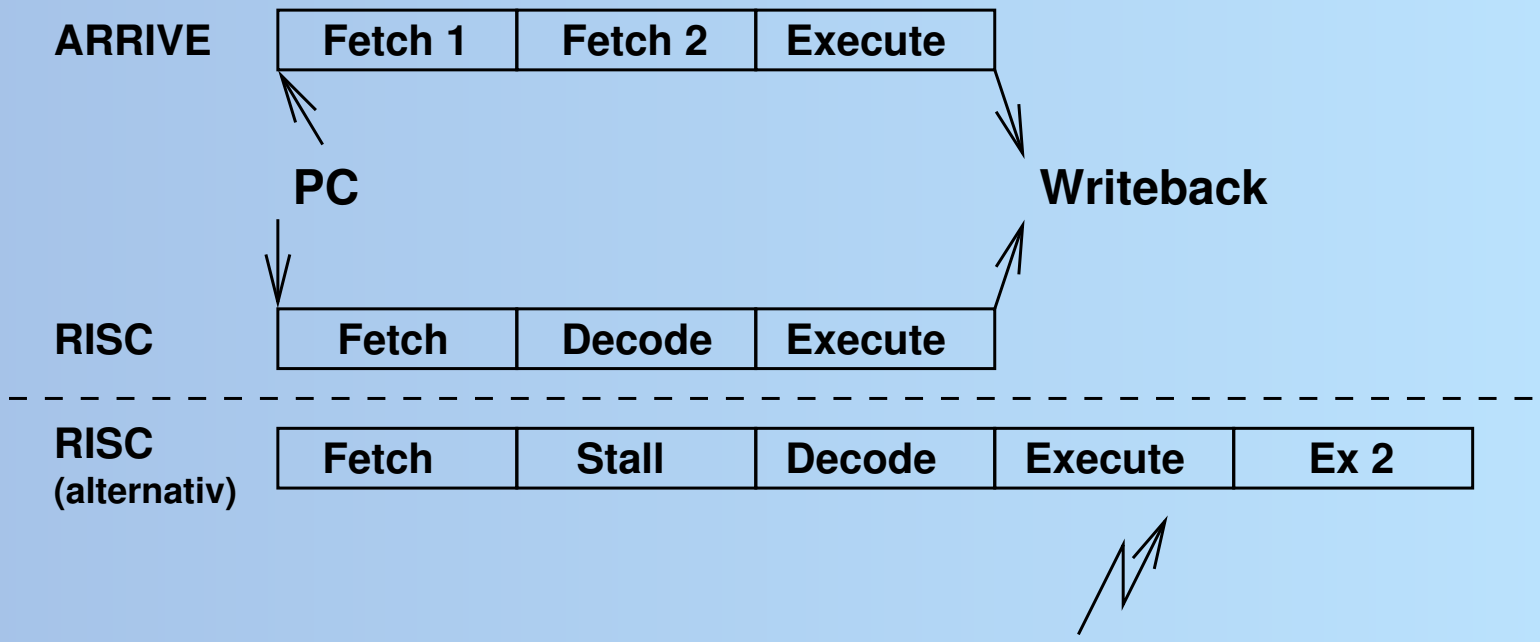
$\Rightarrow$ vollständige Kontrolle über Target

Martin Zimmerling
Institut für Technische Informatik
mz793134@inf.tu-dresden.de

ARRIVE: Anhalten

- Anhalten an Befehlsgrenzen
- Jeder geholte Befehl wird ausgeführt!

Pipelining:

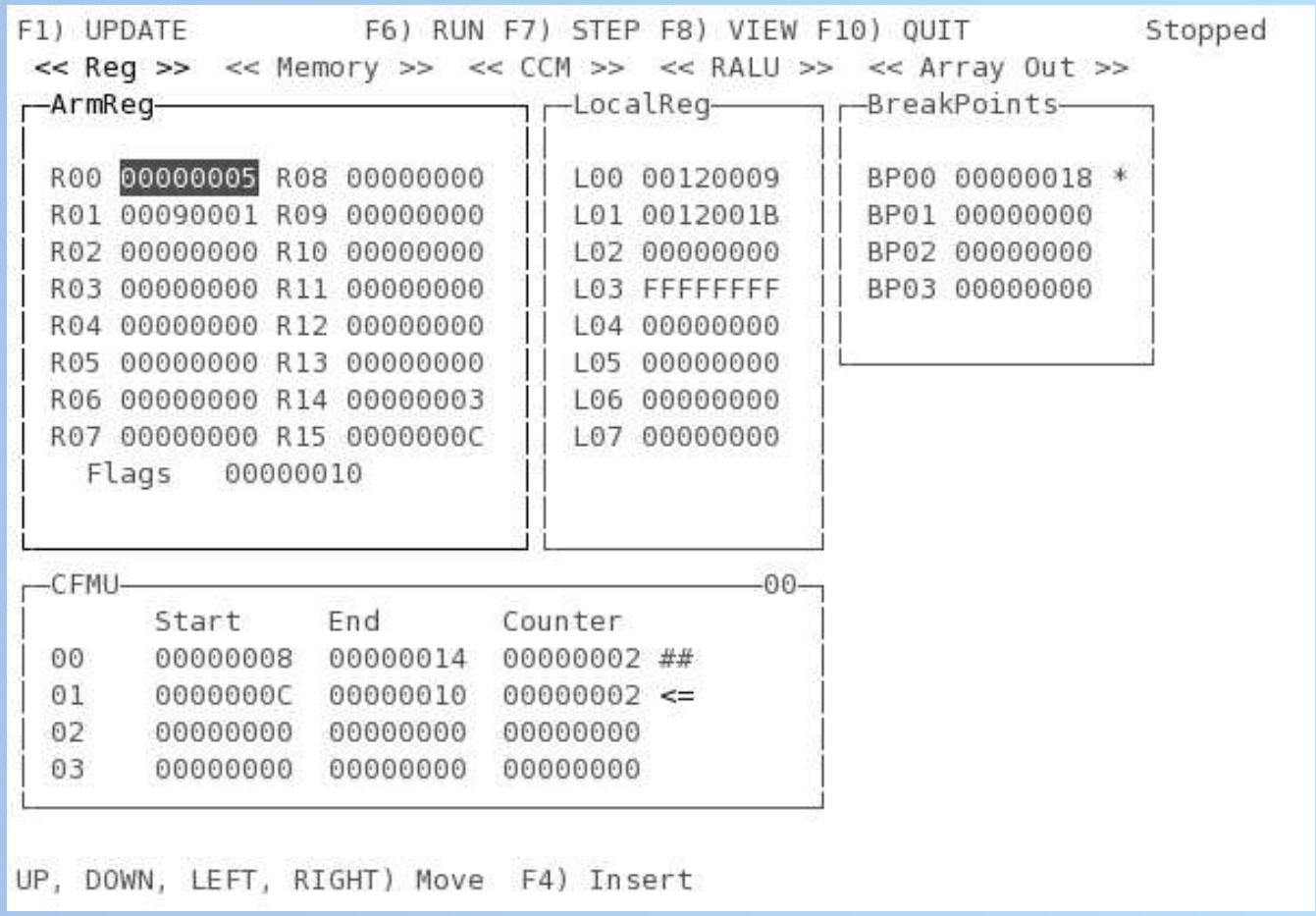


Martin Zimmerling
Institut für Technische Informatik
mz793134@inf.tu-dresden.de

Debug-Werkzeuge

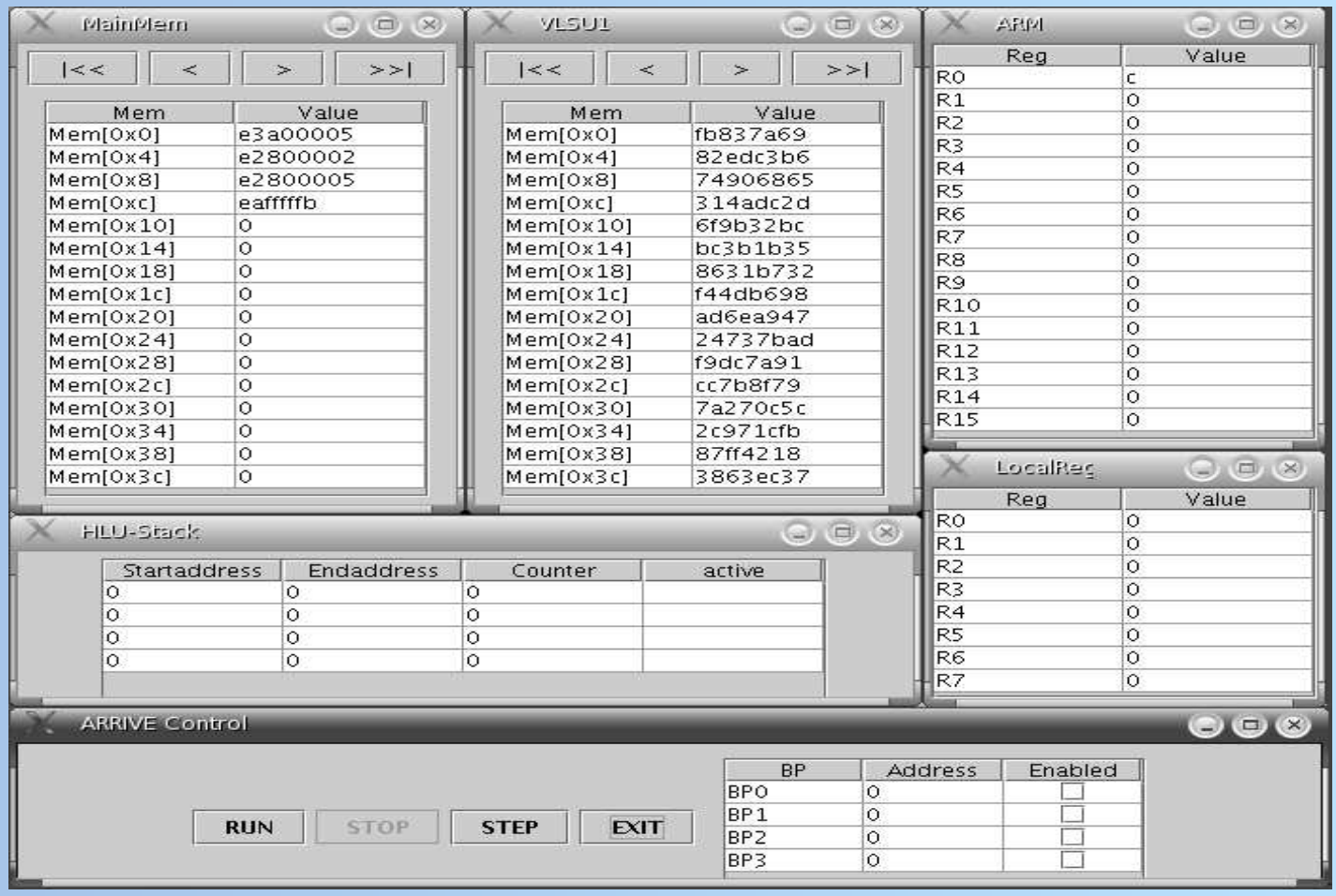
- Sammlung an Werkzeugen
  - Oberflächen (Textmodus, Java)
  - Programmlader für TADL-Programme
  - diverse andere Programme (Speicher-, Registerdump, ...)
- Debugsoftware einfach zu erstellen $\Rightarrow$ Flexibilität
- Programme bisher nur bedingt parametrierbar (nur zur Compile-Zeit)

Oberfläche (Text-Modus)



Martin Zimmerling
Institut für Technische Informatik
mz793134@inf.tu-dresden.de

Oberfläche (graphisch)

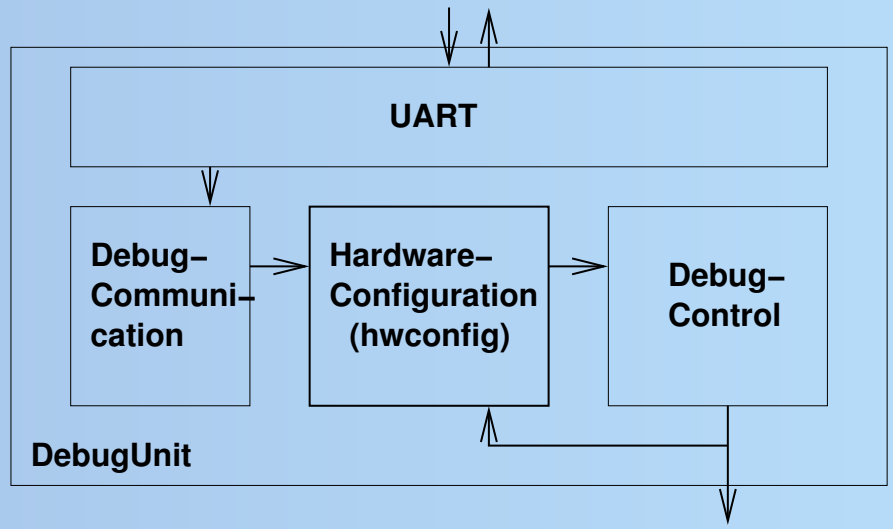


Martin Zimmerling
Institut für Technische Informatik
mz793134@inf.tu-dresden.de

Debugeinheit: Erweiterung

Ziel: erweiterte Unterstützung des Debuggings

- Memory-Tracing
- komplexe Debug-Befehle bzw. Ausnutzung der Debugeinheit durch RISC-Kern



- Umsetzung des Konzepts funktioniert

Martin Zimmerling
Institut für Technische Informatik
mz793134@inf.tu-dresden.de

Ergebnisse (Algorithmen)

Testalgorithmen:

Algorithmus	Arrive-Variante	Ladezeit	Resultat
IIR-Filter	8-Bit	1895 ms	läuft korrekt
FIR-Filter	8-Bit	1045 ms	läuft korrekt
Viterbi-Decoder	8-Bit	1953 ms	läuft korrekt
DCT8	8-Bit	1080 ms	läuft korrekt
DCT8x8	8-Bit	1129 ms	läuft korrekt
Turbo-Decoder	16-Bit	7270 ms	läuft korrekt

- Einschränkung: 16-Bit Variante nicht kompatibel zu Prosim
- FFT nicht lauffähig (eingeschränkter Befehlssatz des RISC)

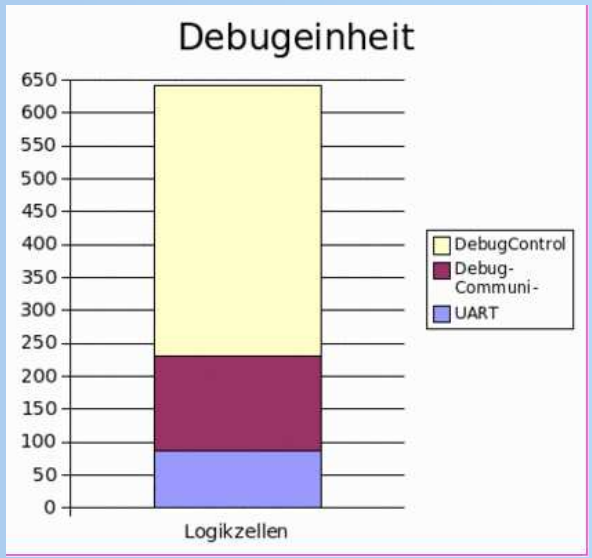
Martin Zimmerling
Institut für Technische Informatik
mz793134@inf.tu-dresden.de

Benchmark-Ergebnisse

	ARM	ARRIVE		
		8-bit PE	16-bit PE	
FIR Filter	9.5	0.5	0.25	cycles/tap
IIR Filter	59.7	3	2	cycles/biquad
FFT Radix-2	99608	8739	5100	cycles/512pt
Viterbi Decoder	212	11	12	cycles/symbol
Turbo Decoder	4648	–	216	cycles/symbol
DCT	5213	170	136	cycles

Ergebnisse (Logikzellen I)

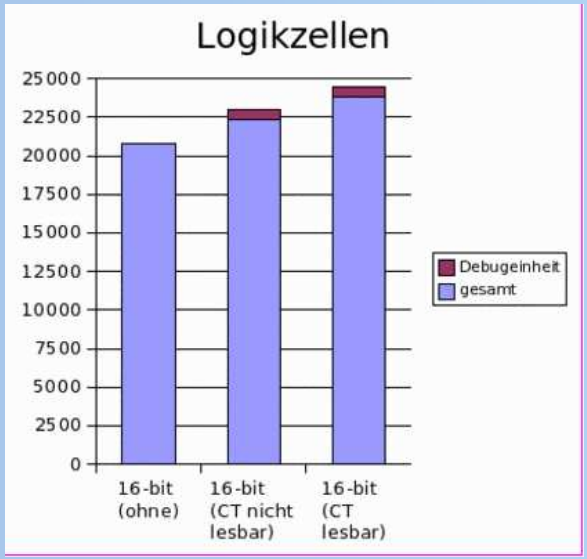
- FPGA EP1S80: ca. 80000 Logikzellen (LC)
- Debugeinheit: 643 Logikzellen
- vgl. RISC-Kern: 3100 Logikzellen
- konstanter Anteil des Gesamtmehraufwands: etwa 1500 LC

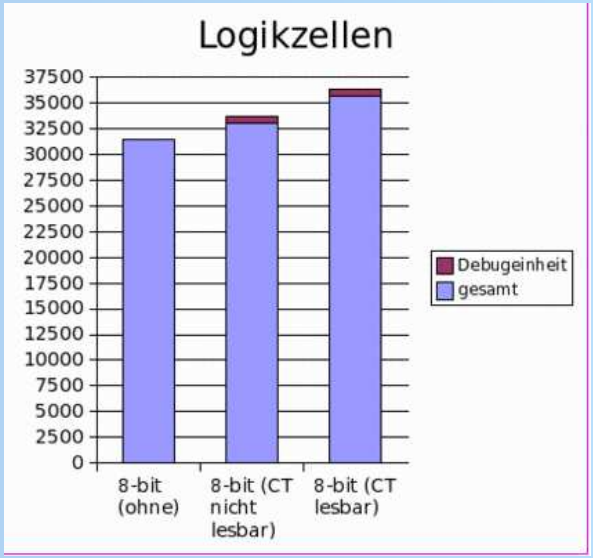


Martin Zimmerling
Institut für Technische Informatik
mz793134@inf.tu-dresden.de

Ergebnisse (Logikzellen II)

- Logikzellen 16-bit: +7.3% +14.4%
- Logikzellen 8-bit: +4.8% +13.4%
- Geschwindigkeit bleibt gleich (11.6 MHz bzw. 6.2 MHz)
- kritischer Pfad durch RALU





Martin Zimmerling
Institut für Technische Informatik
mz793134@inf.tu-dresden.de

Zusammenfassung

- Vorstellung ARRIVE als Beispiel für eine rekonfigurierbare Architektur
- Erfüllung der Debuganforderungen (Flexibilität, Skalierbarkeit, Beobachtbarkeit) in der Realisierung
- Implementationsdetails (stoppen RISC-Kern, Entwurf und Realisierung der HLU) wurden nicht genannt
- ARRIVE-Prototyp konnte durch Debugeinheit erfolgreich auf FPGA in Betrieb genommen und getestet werden
- Standardzellen-ASIC-Synthese leider nicht möglich gewesen

Martin Zimmerling
Institut für Technische Informatik
mz793134@inf.tu-dresden.de

Ausblick

- Verbesserung der Debugwerkzeuge
  - verbessertes Parsen der TADL-Programme
  - Parametrierung der Oberfläche zur Startzeit oder Laufzeit
  - Parametrierung der Werkzeuge zur Nutzung mit anderen Architekturen
- Erweiterung des Mikroprogrammsteuerwerks
- Standardzellen-ASIC-Synthese

Martin Zimmerling
Institut für Technische Informatik
mz793134@inf.tu-dresden.de