



# Entwurf und Implementierung einer parametrierbaren Trace- Hardware am Beispiel der SHAP-Mikroarchitektur

Vortrag zum Diplom – 01. Juli 2009

Stefan Alex  
s2174321@inf.tu-dresden.de

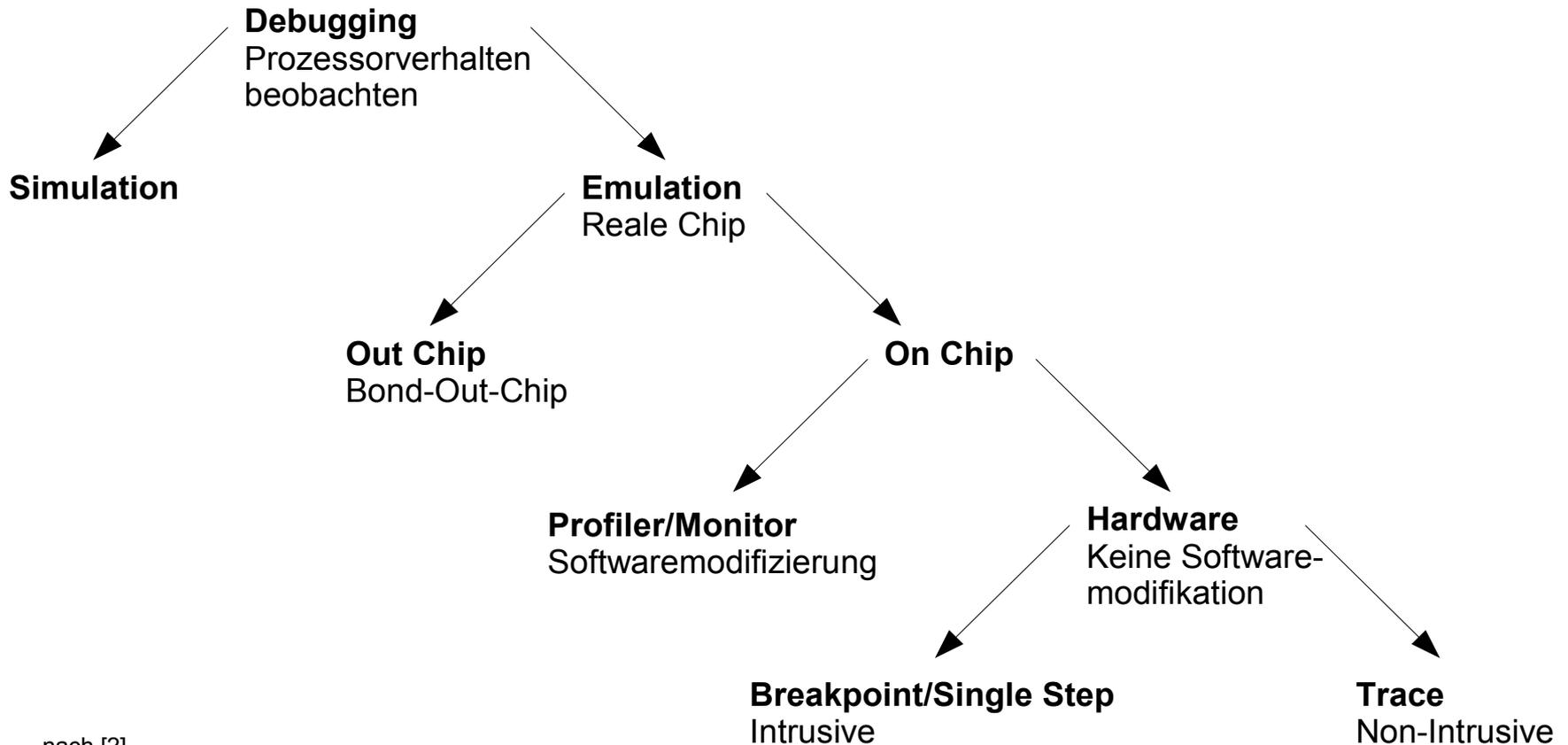
# Gliederung

1. Aufgabenstellung
2. Begriffsbestimmung
3. Tracing
4. Anforderungsanalyse
5. Weitere Vorgehensweise
6. Quellen

# 1. Aufgabenstellung

- Literaturstudium zu Trace-Hardware in eingebetteten Systemen
- Analyse der Referenzszenarien im Hinblick auf notwendige Trace-Infrastruktur und deren Parametrierbarkeit.
- Evaluierung der Ansätze aus der Literatur bezüglich Eignung für die Referenzszenarien.
- Darauf aufbauend, Entwurf einer geeigneten allgemeinen Trace-Infrastruktur nebst Schnittstellen.
- Implementierung von Prototypen der Trace-Hardware-Module inklusive Module für die Vorverarbeitung.
- Test der Trace-Infrastruktur am Beispiel der SHAP-Mikroarchitektur mittels der Referenzszenarien
- Zusammenfassung und Dokumentation der Ergebnisse.

## 2. Begriffsbestimmung



nach [2]

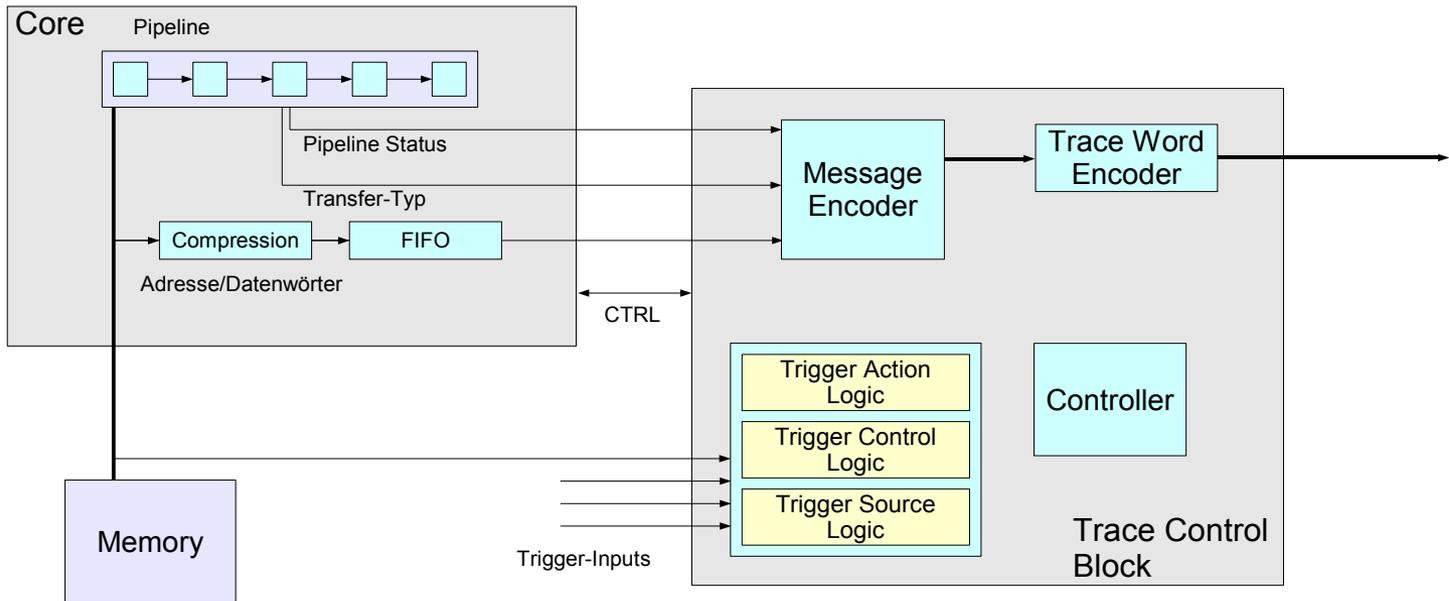
## 3. Tracing

### Trace-Arten

- Program-Trace
  - Program-Counter des aktuellen Befehls
- Ownership-Trace
  - Context-ID des laufenden Prozesses
- Data-Trace
  - Adresse und Datenwert bei Load/Store-Operationen
- Event-Trace
  - Pipeline-Stalls
  - Cache-Misses
  - Performance-Counter
  - ...

# 3. Tracing

## Architektur am Beispiel PDTrace

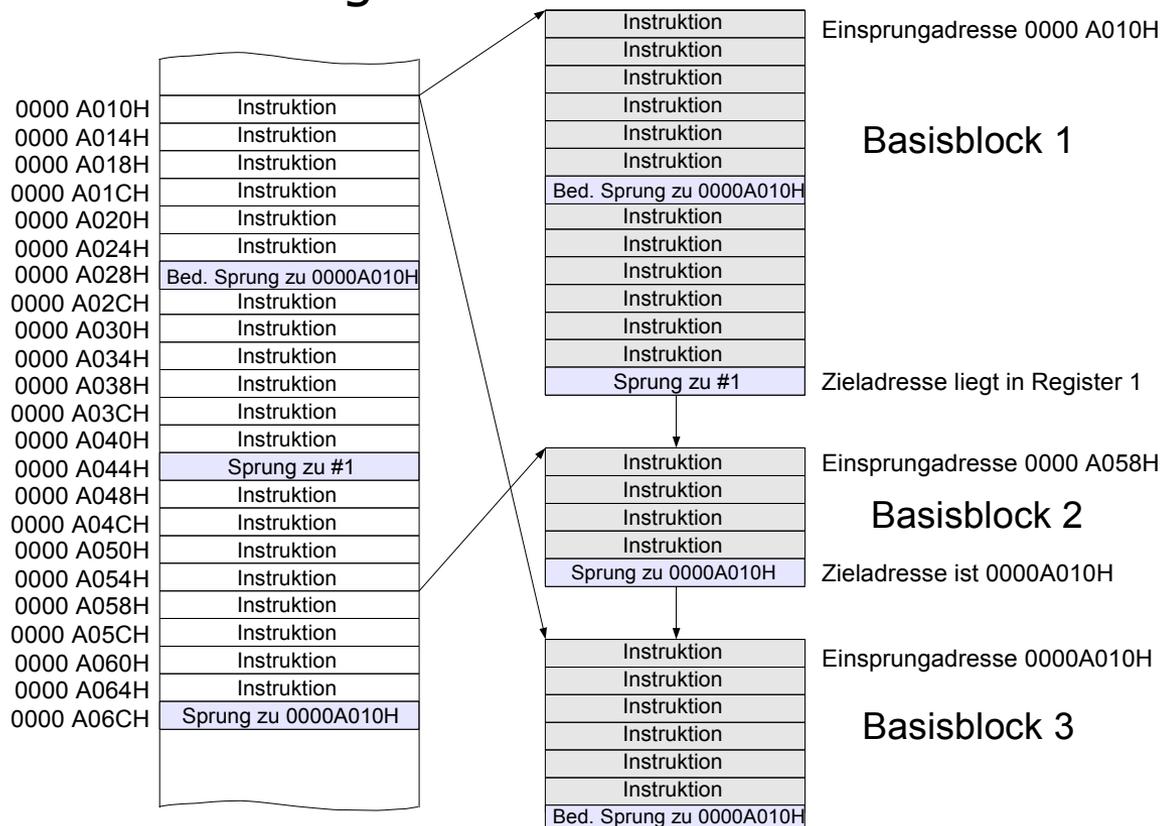


## 3. Tracing Komprimierung

- Problem: Datenvolumen
- 100 Mhz, 32-Bit-Architektur
  - 400 MB/s Programmflussinformationen
  - 80-120 MB/s Datentrace
  - ca. 60 MB/s Framing-Informationen
- Lösung: Trace-Algorithmen
  - Program-Flow-Change-Model
  - Ausnutzen der Lokalität von Adresszugriffen
    - Übertragen der Adressdifferenz (PDATS, Mache, VPC)
    - Zweite Stufe: generische Algorithmen (LZ)

# 3. Tracing

## Program-Flow-Change-Model



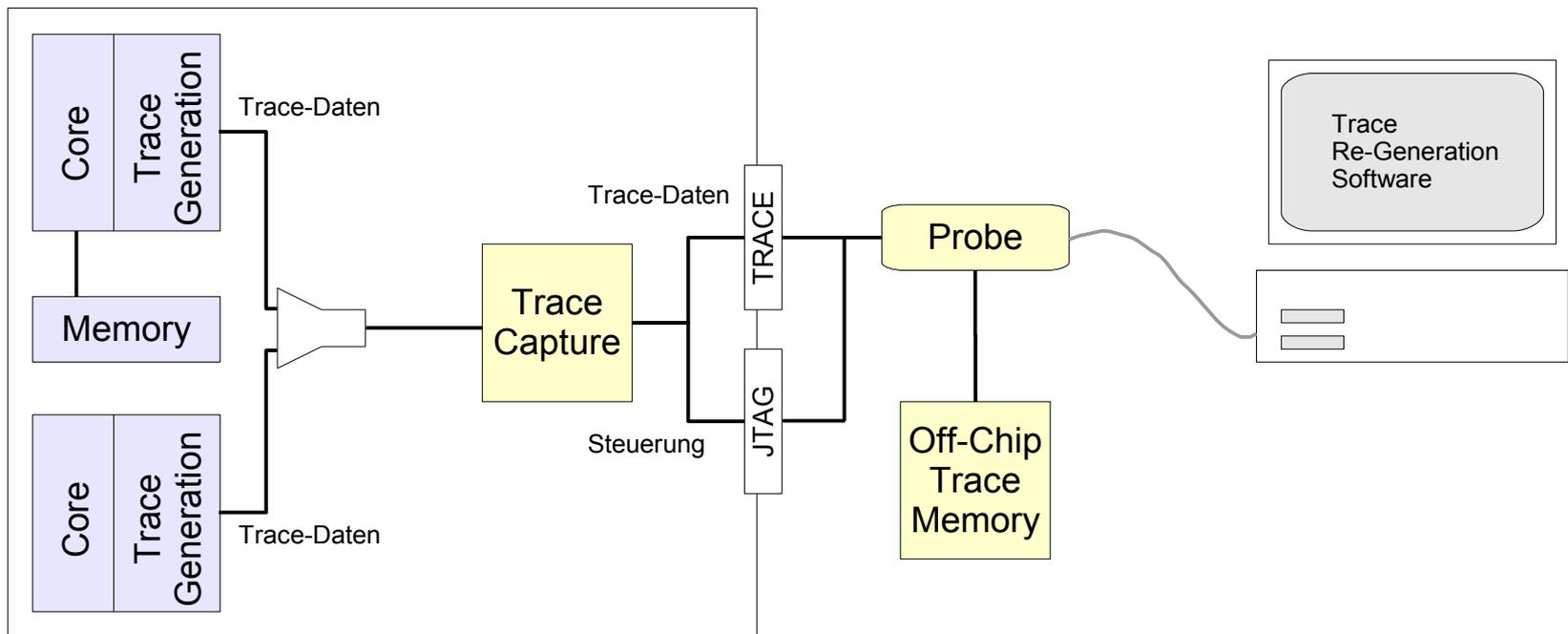
# 3. Tracing

## Weitere Problemstellungen

- Synchronisation
  - Übertragung der vollen PC- und Datenadresse
  - zusätzliche Informationen (Prozessor-Modus, Konfiguration, ...)
- Puffer-Überlauf
  - Priorisierung bei mehreren Puffern
  - Pipeline-Stall
  - Filtern
  - Neusynchronisation
- Trace-Speicher

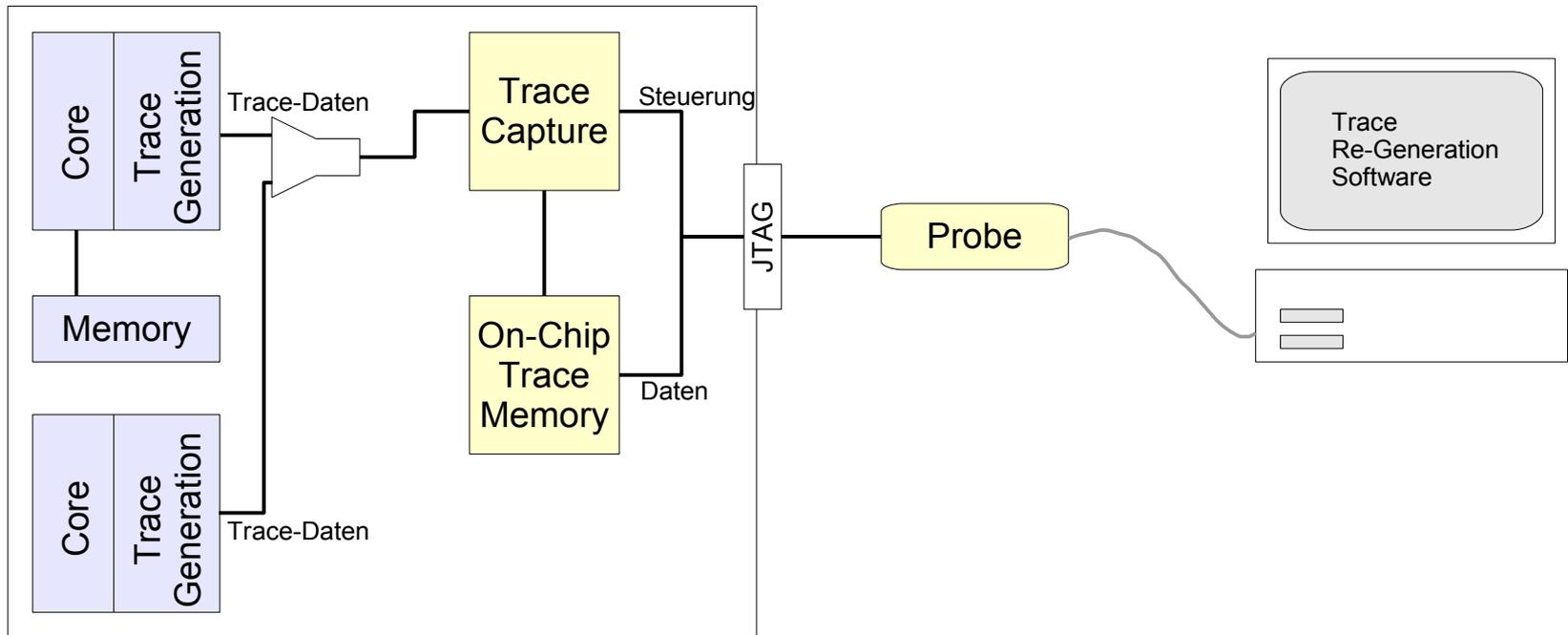
# 3. Tracing

## Off-Chip Trace-Speicher

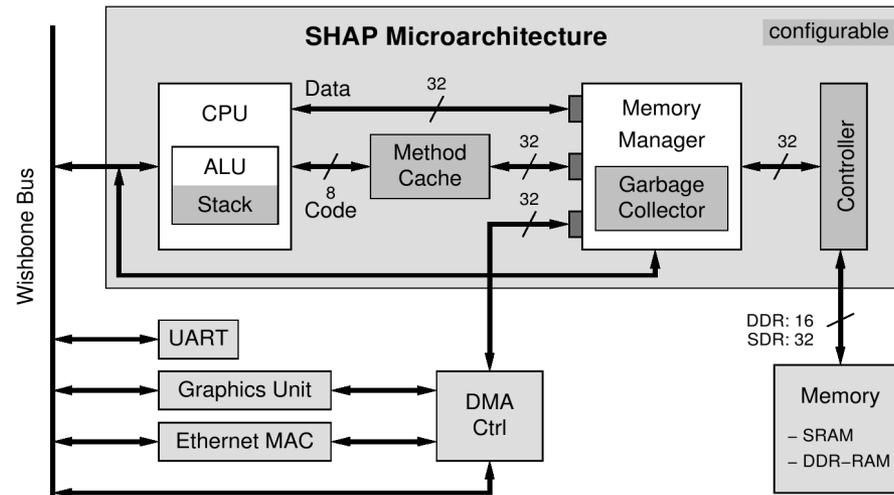


# 3. Tracing

## On-Chip Trace-Speicher

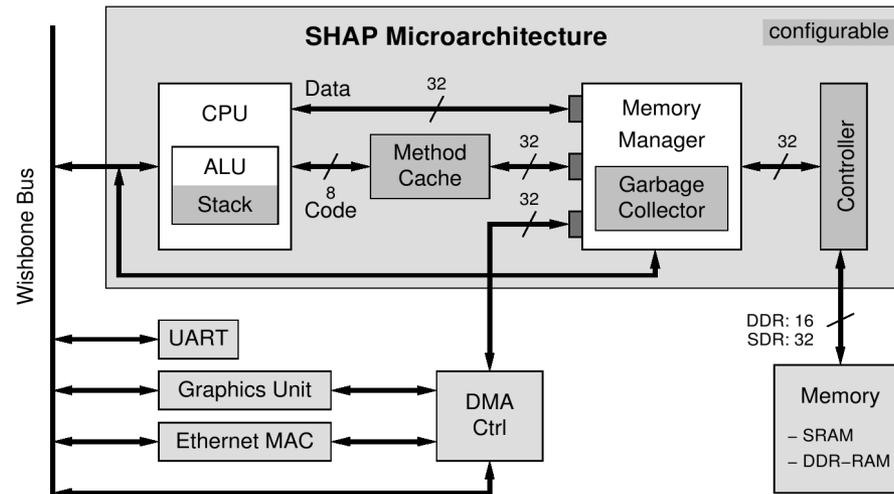


## 4. Anforderungsanalyse



- Program-Trace
  - Methodenadresse + Program-Counter bei Bytecode-Fetch
  - Adresse des Microcodes
  - Markierung von Interrupts
- Ownership-Trace
  - Taskwechsel durch Tracen des Switch-Kommandos mit Thread-ID

## 4. Anforderungsanalyse



- Data-Trace
  - Speicherzugriff durch CPU
    - Ziel: Zugriffsmuster auf Objekte
  - Erweiterter Daten-Trace für Wishbone-CPU-Verbindung
- Event-Trace
  - Cache-Hit/Cache-Miss des Methodencaches
  - Ausführungszeiten von Bytecode-Operationen

## 4. Anforderungsanalyse

- Spezielle Nachrichten
  - Garbage Collection – Übertragung der markierten Referenzen + Status
- Trigger
  - Adressbereiche des Bytecodes
    - Ziel: Trennung der Aufzeichnung der API und des Nutzerprogramms
- Multi-Core-Unterstützung

## 5. Weitere Vorgehensweise

- Ermitteln von Metriken der Tracequellen
  - Häufigkeit, Datenvolumen, Lokalität
- Ableitend: Entwurfsentscheidungen
  - Bestehende Ansätze in Betracht ziehen (Nexus, etc.)
- Konkreter Entwurf
  - Einbeziehung vorhandener Module für Methoden-Cache, Garbage-Collector
- Implementierung und Test

## 6. Quellen

- [1] Preußner, T.B.; Zabel, M.; Reichel, P., "The SHAP Microarchitecture and Java Virtual Machine" Technical Report Fakultät Informatik, TU Dresden, 2007
- [2] Hu, X.; Chen, S.; „Applications of On-chip Trace on Debugging Embedded Processor“, Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007
- [3] Kao, C.-F.; Chen, H.-M.; Huang, I.-J., "Hardware-Software Approaches to In-Circuit Emulation for Embedded Processors", IEEE Design & Test, Volume 25, Nr. 5, 2008, IEEE Computer Society Press,
- [4] Huang, I.-J.; Kao, C.-F.; Chen, H.-M.; Juan, C.-N.; Lu, T.-A., "A Retargetable Embedded In-Circuit Emulation Module for Microprocessors", IEEE Design & Test, Volume 19, Nr. 4, 2002, IEEE Computer Society Press
- [5] Kao, C.-F.; Huang S.-M.; Huang, I.-J., "A Hardware Approach to Real-Time Program Trace Compression for Embedded Processors", Circuits and Systems I: Regular Papers, IEEE Transactions on, 2007
- [6] Hopkins, A. B. T., McDonald-Maier, K. D., "Debug Support Strategy for Systems-on-Chips with Multiple Processor Cores", IEEE Trans. Comput., Volume 55, Nr. 2, 2006, IEEE Computer Society
- [7] "MIPS PDtrace Specification", MIPS Technologies Inc., Rev 6.11, 2008
- [8] "CoreSight Components - Technical Reference Manual", ARM Limited, Edition F, 2008
- [9] "The Nexus 5001 Forum - Standard for a Global Embedded Processor - Debug Interfacen", IEEE-Industry Standards and Technology Organization (IEEE-ISTO), Edition 2.0, 2003

## 6. Quellen

- [10] Johnson, E. E.; Ha, J.; Zaidi, M. B., "Lossless Trace Compression", IEEE Trans. Comput., Volume 50, Nr. 2, 2001, IEEE Computer Society
- [11] Samples, A. D., "Mache: no-loss trace compaction", SIGMETRICS Perform. Eval. Rev., Volume 17, Nr. 1, 1989, ACM
- [12] Hopkins, A. B. T.; McDonald-Maier, K. D., "Trace algorithms for deeply integrated complex and hybrid SoCs", AHS '07: Proceedings of the Second NASA/ESA Conference on Adaptive Hardware and Systems, 2007, IEEE Computer Society
- [13] Jackson, S. J.; Ke, J.; Ratanaworabhan, P.; Burtscher, M.; Ganusov, I.; Sam, N. B., "The VPC Trace-Compression Algorithms", IEEE Trans. Comput., Volume 54, Nr. 11, 2005, IEEE Computer Society
- [14] Ball, S. R., "Debugging Embedded Microprocessor Systems", 1998, Newnes