TECHNISCHE UNIVERSITÄT DRESDEN

Fakultät Informatik Institut für Technische Informatik, Professur für VLSI-Entwurfssysteme, Diagnostik und Architektur

Statusvortrag zur Dissertation

Modellierung von On-Chip-Trace-Architekturen für eingebettete Systeme

Kai-Uwe Irrgang
Kai-Uwe.Irrgang@mailbox.tu-dresden.de

DRESDEN concept

Gliederung

Einleitung und Motivation

Anwendungen und Notwendigkeit von Trace

Anwendungsgebiete

- Fehlersuche in Echtzeit unter realen Einsatzbedingungen
  (Beispiel: Debugging von Steuerungen mechanischer Systeme)
- Leistungsanalyse / Profiling
  (Messung von Schleifenzeiten, Funktionslaufzeiten, Interrupt-Latenzen, ...)
- Analyse und Nachweis der Code-Abdeckung (beispielsweise für Zertifizierung)
- Analyse des Cache-Verhaltens (Messung des Hit/Miss-Verhältnisses)

Rückwirkung auf das zu untersuchende System

- invasiv: Die Rückwirkung verletzt die Echtzeitbedingungen massiv
  (beispielsweise durch das Anhalten des Systems bei Breakpoints).
- *minimal-invasiv*: Die Rückwirkung auf das System ist tolerierbar.
- **nicht-invasiv**: Es gibt keine Rückwirkung auf das System.

Einleitung und Motivation

Leistungsfähige Off-Chip-Analysesysteme

Etablierte nicht-/minimal- invasive Off-Chip-Tools

- In-Circuit-Emulator (Bond-out-chip oder FPGA-Implementierung)
- Logikanalysator

Probleme und Grenzen

- höhere Signalfrequenzen (Grenzen liegen bei circa 100 MHz)
- mögliche Abweichungen im Verhalten zwischen Emulator und realem System (Bond-Drähte der zusätzlichen Pins belasten interne Signale, ...)
- anderer Footprint mit mehr Pins und Signalen (extra PCB für Entwicklung, Einsetzbarkeit in Zielumgebung eingeschränkt,...)
- Trend zu Multi-Core-SoC
- ...

Einleitung und Motivation

On-Chip-Trace für eingebettete Prozessoren und SoC

Herausforderungen beim On-Chip-Trace

- Die Bandbreite des Systemzugangs (z. B. JTAG-Schnittstelle, Nexus-AUX-Port) ist meist zu gering, um alle anfallenden Trace-Daten zu übertragen.
- Es ist eine Pufferung mittels On-Chip-Speicher erforderlich.
- Zusätzliches Problem bei Multi-Core-SoC: Alle Cores müssen sich den Systemzugang (und Trace-Speicher) und damit die Bandbreite teilen.

Paradigmenwechsel bei Off- zu On-Chip-Trace

- Off-Chip: Es wird zunächst (fast) alles aufgezeichnet. Die Filterung und Auswertung der Trace-Daten erfolgt nach dem Trace-Lauf.
- On-Chip: Es werden (möglichst) nur die für die jeweilige Analyseaufgabe relevanten Trace-Daten aufgezeichnet. Zusätzlich werden die Trace-Daten komprimiert. Die Filterung und die Kompression erfolgen On-Chip.

Einleitung und Motivation

Anforderungen und Ziele an den On-Chip-Trace

- Das Tracing muss **nicht-invasiv** erfolgen.
- Die Kompression muss **verlustlos** arbeiten können.
- Der Kompressor muss **echtzeitfähig** sein, d. h. mit der Geschwindigkeit des Prozessors Schritt halten.
- Eine möglichst große Menge NICHT relevanter Trace-Daten soll automatisch ausgefiltert werden.
- Die verbleibenden Trace-Daten sollen mit einem möglichst **hohen Kompressionsfaktor** komprimiert werden.
- Filterung und Kompression sollen mit möglichst **geringem Aufwand**, insbesondere bezüglich der Chipfläche, erfolgen.

Abgrenzung des Arbeitsgebietes

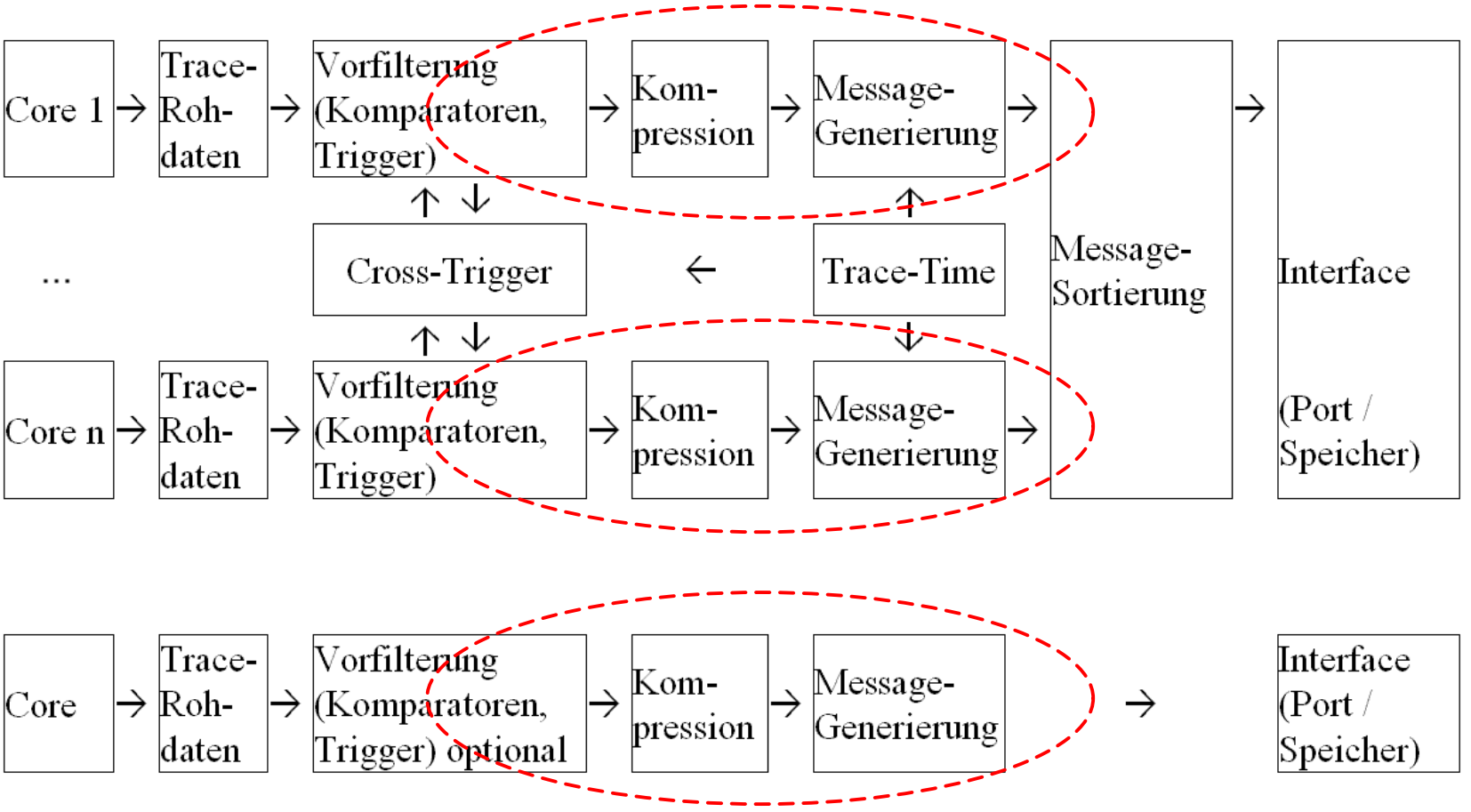
Klassifikation von On-Chip-Trace

Unterscheidung nach zu untersuchenden Komponenten

- **Core-Trace** (Single-Core mit Einbindbarkeit in Multi-Core-Trace)
- Logik-Trace (komplexe Peripherie)
- Bus-Trace (insbesondere Multi-Master-Busse)
- System-Trace (insbesondere Interaktion von Cores / Logik / Bussen)

Trace-Daten innerhalb des Core-Trace

- **Instruktions-Trace**
- Kontext-Trace / Ownership-Trace
- *Daten-Trace* (Daten-Adressen, Daten-Werte, lesend und/oder schreibend)
- Watchpoint-Trace
- sonstige Trace-Messages (Zeitstempel, ...)


Core 1
Trace-Roh-daten
Vorfilterung (Komparatoren, Trigger)
Kom-pression
Message-Generierung
...
Cross-Trigger
Trace-Time
Message-Sortierung
Interface
Core n
Trace-Roh-daten
Vorfilterung (Komparatoren, Trigger)
Kom-pression
Message-Generierung
(Port / Speicher)
Core
Trace-Roh-daten
Vorfilterung (Komparatoren, Trigger) optional
Kom-pression
Message-Generierung
Interface (Port / Speicher)

Abgrenzung des Arbeitsgebietes

Kompression und Message-Generierung

Reduktion des Datenvolumens

- explizite Relevanzfilterung (Trace-Qualification)
  - abhängig von der konkreten Analyseaufgabe
  - manuelle oder halbautomatische Konfiguration der Filter-Logik
- **implizite Relevanzfilterung** (bekannte Informationen, Redundanzen)
  - abhängig vom Trace-Typ (**Instruktions-Trace**, Daten-Trace, ...)
  - Ausnutzung spezifischer impliziter Eigenschaften
  - vollautomatische Filterung
- **Kompression** im engeren Sinne
  - abhängig vom Trace-Typ
  - vollautomatischer Ablauf
- **Message-Generierung**
  - erfolgt zumeist mit impliziter Relevanzfilterung oder Kompression
  - vollautomatischer Ablauf

Stand der Forschung

Gesamtüberblick

Standard

- IEEE-ISTO 5001™-2003 (Nexus)

akademische Lösungen

- Autorengruppe um Andrew B. T. Hopkins und Klaus D. McDonald-Maier
- Autorengruppe um Xiao Hu und Shuming Chen
- Autorengruppe um Ing-Jer Huang und Chung-Fu Kao
- Autorengruppe um Aleksander Milenkovic und Vladimir Uzelac

Industrielle Lösungen

- ARM Embedded Trace Macrocell (ETM) und CoreSight
- Infineon Multi-Core-Debug-Solution (MCDS)
- MIPS PDtrace

Stand der Forschung

Gesamtüberblick - Zeitachse

	bis 2006	2007	2008	2009	2010	2011
IEEE-ISTO 5001™ (Nexus)	[IEE99] **[IEE03]**					
Andrew B. T. Hopkins; Klaus D. McDonald-Maier	**[HM06a]** [HM06b] **[HM06c]** [SHM06]	[HM07a] [HM07b] [MSM07] [SHM+07]	[HMP+08]			
Ing-Jer Huang; Chung-Fu Kao		**[KHH07]** [KHL07]	[KCH08] [LSH08] [YHZ08]	[LYK+09]	[CJH10] [CJW+10] [YCH10]	[YLK+11]
Xiao Hu; Shuming Chen	**[HMC+06]**	**[HC07]** [HMC07]				
Aleksander Milenkovic; Vladimir Uzelac	[MM03] [MMK03]	[MM07] [MMB07]		**[UM09]** **[UMM+09]**	[Uce10] [UM10] **[UMB+10]**	**[MUM+11]** [UM11]
ARM CoreSight & Embedded Trace Macrocell	[ARM05]	[ARM07]	[ARM08a] [ARM08b]			

Stand der Forschung

Implizite Relevanzfilterung beim Instruktions-Trace

Program Flow Change Model (Basis: Objekt-/Binärcode)

- Der Programmfluss kann sich nur an Verzweigungsbefehlen (jmp, call, ret) oder beim Auftreten eines Interrupts oder einer Exception ändern.
- Bei sequentiellen Anweisungen zwischen den Verzweigungen genügt deren Zählung (bei „Predicated Instructions" optional mit Ausführungs-Bitmap).
- Unbedingte direkte Verzweigungen können wie sequentielle Anweisungen behandelt werden.
- Bei bedingten direkten Verzweigungen genügt das Tracen, ob die Bedingung wahr oder falsch war.
- Bedingte Verzweigungen mit nicht erfüllter Bedingung können als nicht ausgeführte sequentielle Anweisungen behandelt werden.
- Nur für indirekte ausgeführte Verzweigungen ist das Tracen der Zieladresse erforderlich.

Voraussetzung: Der Code ist mindestens während des Trace-Laufes statisch. (Selbstmodifizierender Code ist nur eingeschränkt tracebar.)

Stand der Forschung

Einfache Mechanismen der Trace-Kompression

Zwei Arten von Lokalität

- zeitliche Lokalität: Kürzlich zugegriffene Adressen haben eine größere Wahrscheinlichkeit, in der nahen Zukunft wieder zugegriffen zu werden.
- räumliche Lokalität: Nahe beieinander liegende Adressen haben eine größere Wahrscheinlichkeit, nacheinander zugegriffen zu werden.

Lokalitätsbasierte Kompression

- Ausnutzung der zeitlichen Lokalität durch Bezug auf den Vorgängerwert
- Ausnutzung der räumlichen Lokalität durch einfache mathematische oder logische Operationen:
  - XOR: nur unikate LSB werden übertragen,
    Unterdrückung führender Nullen
  - SUB: arithmetische Differenz zum Vorgänger inklusive Vorzeichenbit,
    Unterdrückung führender Nullen oder Einsen (=Vorzeichen)

Stand der Forschung

IEEE 5001 Nexus - Überblick

Definiert sind

- physikalisches Interface (insbesondere AUX-OUT als Trace-Port)
- Message-basiertes Protokoll
- vier Klassen von Leistungsmerkmalen für Debug und Trace
- Register-Infrastruktur (Nexus Recommended Registers - NRRs)

Nicht definiert sind

- sämtliche konkreten Implementierungen (Komparatoren, Trigger, logische und sequentielle Verknüpfungen der Trigger, ...)
- Interfaces der Debug/Trace-Einheit zum Prozessor
- Cross-Trigger
- Trace-Speicher

Stand der Forschung

Akademische Lösungen - Hopkins, McDonald-Maier (1)

[MSM05] MAYER, Albrecht ; SIEBERT, Harry ; MCDONALD-MAIER, Klaus D.: Debug Support, Calibration and Emulation for Multiple Processor and Powertrain Control SoCs. In: *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE 2005)*, 2005, ISBN: 978-0-7695-2288-3, ISSN: 1530-1591, S. 148-152

[HM06a] HOPKINS, Andrew B. T. ; MCDONALD-MAIER, Klaus D.: Debug Support Strategy for Systems-on-Chips with multiple Processor Cores. In: *IEEE Transactions on Computers*, Bd. 55, Nr. 2, 2006, ISSN: 0018-9340, S. 174-184

[HM06b] HOPKINS, Andrew B. T. ; MCDONALD-MAIER, Klaus D.: A Generic On-Chip Debugger for Wireless Sensor Networks. In: *Proceedings of the First NASA/ESA Conference on Adaptive Hardware and Systems (AHS 2006)*, 2006, ISBN: 978-0-7695-2614-0, S. 338-342

[HM06c] HOPKINS, Andrew B. T. ; MCDONALD-MAIER, Klaus D.: Debug support for complex systems on-chip: a review. In: *IEE Proceedings: Computers and Digital Techniques*, Bd. 153, Nr. 4, 2006, ISSN: 1350-2387, S. 197-207

[SHM06] SCOTTOW, Richard G. ; HOPKINS, Andrew B. T. ; MCDONALD-MAIER, Klaus D.: Instrumentation of Real-Time Embedded Systems for Performance Analysis. In: *Proceedings of the IEEE Instrumentation and Measurement Technology Conference (IMTC 2006)*, 2006, ISBN: 978-0-7803-9359-2, ISSN: 1091-5281, S. 1307-1310

[HM07a] HOPKINS, Andrew B. T. ; MCDONALD-MAIER, Klaus D.: Debug Support for Hybrid SoCs. In: *Proceedings of the Second NASA/ESA Conference on Adaptive Hardware and Systems (AHS 2007)*, 2007, ISBN: 978-0-7695-2866-3, S. 195-202

[HM07b] HOPKINS, Andrew B. T. ; MCDONALD-MAIER, Klaus D.: Trace algorithms for deeply integrated complex and hybrid SoCs. In: *Proceedings of the Second NASA/ESA Conference on Adaptive Hardware and Systems (AHS 2007)*, 2007, ISBN: 978-0-7695-2866-3, S. 641-646

[MSM07] MAYER, Albrecht ; SIEBERT, Harry ; MCDONALD-MAIER, Klaus D.: Boosting Debugging Support for Complex Systems on Chip. In: *IEEE Computer*, Bd. 40, Nr. 4, 2007, ISSN: 0018-9162, S. 76-81

[SHM+07] SARTAIN, Pieter. ; HOPKINS, Andrew B. T. ; MCDONALD-MAIER, Klaus D. ; HOWELLS, W. Gareth J.: A System Level Framework for Monitoring and Self Diagnosis in ESPACENET. In: *Proceedings of the Second NASA/ESA Conference on Adaptive Hardware and Systems (AHS 2007)*, 2007, ISBN: 978-0-7695-2866-3, S. 677-681

[HMP+08] HOPKINS, Andrew B. T. ; MCDONALD-MAIER, Klaus D. ; PAPOUTSIS, Evangelos ; HOWELLS, Gareth: Adaptive Online Profiling Hardware for ICmetrics Based Security. In: *Proceedings of the Third NASA/ESA Conference on Adaptive Hardware and Systems (AHS 2008)*, 2008, ISBN: 978-0-7695-3166-3, S. 498-504

Stand der Forschung

Akademische Lösungen - Hopkins, McDonald-Maier (2)

Gesamtkonzept und Komponenten

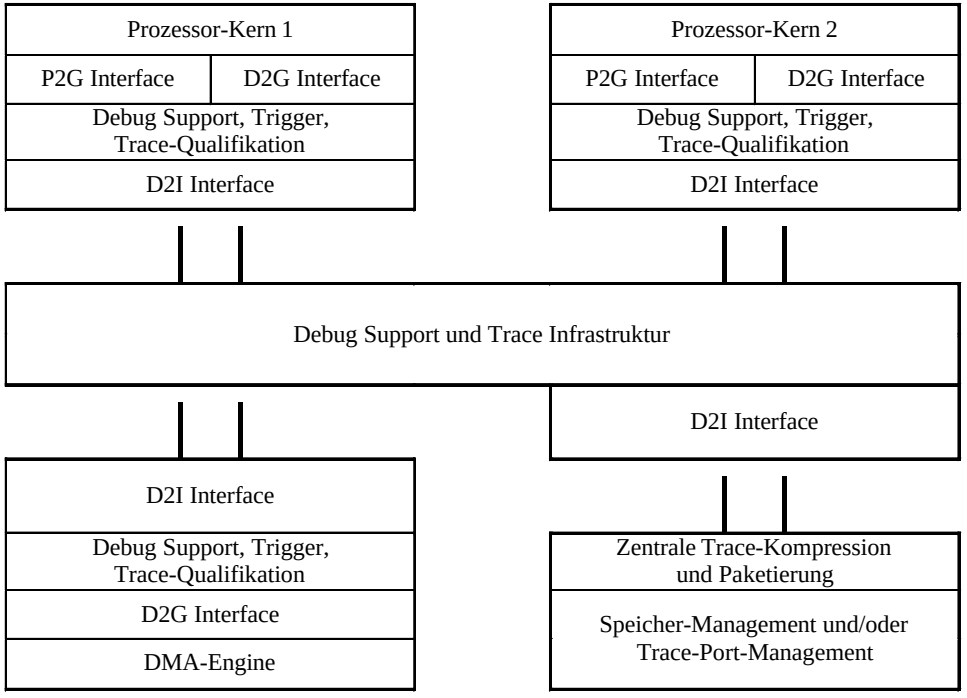
- Zielstellung: Debug/Trace-Framework für ein Multi-Core-SoC
- zentrale Trace-Infrastruktur mit Cross-Trigger und zentraler Kompression
- Wrapper und Interfaces für Prozessoren und Peripherie
- Trigger-Generierung und Trace-Qualification

Aussagen bezüglich zentraler Kompression

- Die Kompressionslogik ist nur ein Mal pro SoC vorhanden und muss nicht für jeden Core repliziert werden. Dies spart Chipfläche.
- Es ist eine bessere Skalierung Leistungsfähigkeit versus Aufwand möglich.
- Lokale Daten-Bursts können besser ausgeglichen werden.

Stand der Forschung

Akademische Lösungen - Hopkins, McDonald-Maier (3)



Stand der Forschung

Akademische Lösungen - Xiao Hu, Shuming Chen (1)

[HMC+06] Hu, Xiao ; Ma, Pengyong ; Chen, Shuming ; Guo, Yang ; Fang, Xing: TraceDo: An On-Chip Trace System for Real-Time Debug and Optimization in Multiprocessor SoC. In: Proceedings of the 4th International Symposium on Parallel and Distributed Processing and Applications (ISPA 2006), 2006, ISBN: 3-540-68067-5, S. 806-817

[HC07] Hu, Xiao ; Chen, Shuming: Applications of On-chip Trace on Debugging Embedded Processor. In: Proceedings of the 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2007), 2007, ISBN: 978-0-7695-2909-7, S. 140-145

[HMC07] Hu, Xiao ; Ma, Pengyong ; Chen, Shuming: Scheduling for Combining Traffic of On-Chip Trace Data in Embedded Multi-core Processor. In: Proceedings of the 3rd International Conference on Embedded Software and Systems (ICESS 2007), 2007, ISBN: 978-3-540-72684-5, S. 67-79

Stand der Forschung

Akademische Lösungen - Xiao Hu, Shuming Chen (2)

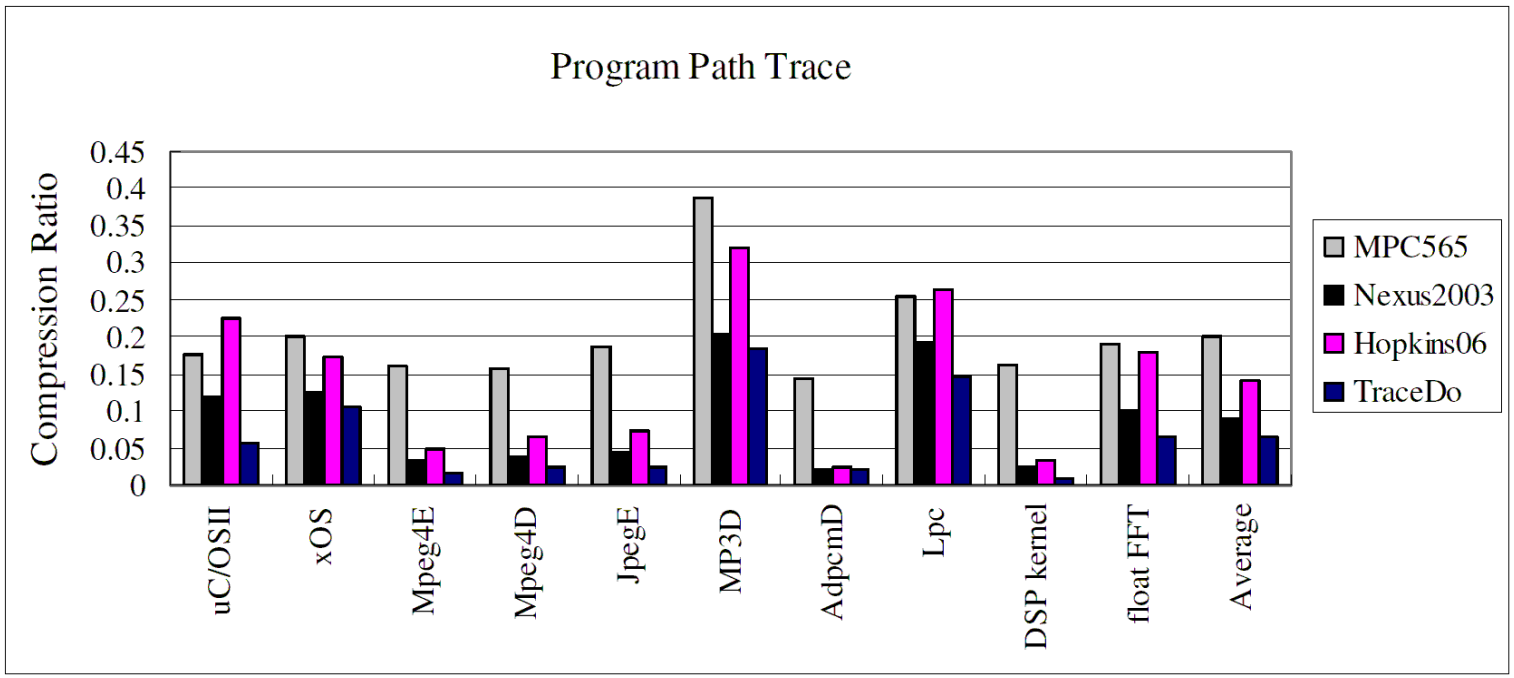
Gesamtkonzept und Komponenten

- Basisarchitektur: heterogenes Multi-Core-System mit einem RISC-Prozessor und vier gleichartigen VLIW-DSPs
- nur die DSPs sind in den Trace eingebunden
- aufgezeichnet werden: Programmpfad, Datenzugriffe, periphere Events und Pipeline-Stall-Events
- je ein separates globales Enable-Bit für die vier Sprungtypen (direkt | indirekt sowie ausgeführt | nicht ausgeführt)
- keine Trigger und keine Trace-Qualification
- Anbindung über breiten Trace-Port an externen Emulator

Stand der Forschung

Akademische Lösungen - Xiao Hu, Shuming Chen (3)

Besonderheit: basiert nur auf Sprungbefehlen (statt auf allen Befehlen)



Stand der Forschung

Akademische Lösungen - Chung-Fu Kao, Ing-Jer Huang (1)

[KHH07] Kao, Chung-Fu ; Huang, Shyh-Ming ; Huang, Ing-Jer: A Hardware Approach to Real-Time Program Trace Compression for Embedded Processors. In: IEEE Transactions on Circuits and Systems I: Regular Papers, Bd. 54, Nr. 3, 2007, ISSN: 1549-8328, S. 530-543

[KHL07] Kao, Chung-Fu ; Huang, Ing-Jer ; Lin, Yi-Ting: An Embedded Multi-resolution AMBA Trace Analyzer for Microprocessor-based SoC Integration. In: Proceedings of the 44th ACM/IEEE Design Automation Conference (DAC 2007), 2007, ISBN: 978-1-59593-627-1, ISSN: 0738-100X, S. 477-482

[KCH08] Kao, Chung-Fu ; Chen, Hsin-Ming ; Huang, Ing-Jer: Hardware-Software Approaches to In-Circuit Emulation for Embedded Processors. In: IEEE Design and Test of Computers, Bd. 25, Nr. 5, 2008, ISSN: 0740-7475, S. 462-477

[LSH08] Lin, Yi-Ting ; Shiue, Wen-Chi ; Huang, Ing-Jer: A Multi-resolution AHB Bus Tracer for Real-time Compression of Forward/Backward Traces in a Circular Buffer. In: Proceedings of the 45th ACM/IEEE Design Automation Conference (DAC 2008), 2008, ISBN: 978-1-60558-115-6, ISSN: 0738-100X, S. 862-865

[YHZ08] Yang, Fu-Ching ; Huang, Wen-Kai ; Zhong, Jing-Kun ; Huang, Ing-Jer: Automatic Verification of External Interrupt Behaviors for Microprocessor Design. In: IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Bd. 27, Nr. 9, 2008, ISSN: 0278-0070, S. 1670-1683

[LYK+09] Lai, Chun-Hung ; Yang, Fu-Ching ; Kao, Chung-Fu ; Huang, Ing-Jer: A Trace-Capable Instruction Cache for Cost Efficient Real-Time Program Trace Compression in SoC. In: Proceedings of the 46th ACM/IEEE Design Automation Conference (DAC 2009), 2009, ISBN: 978-1-60558-497-3, ISSN: 0738-100X, S. 136-141

[CJW+10] Chen, Liang-Bi ; Ju, Jiun-Cheng ; Wang, Chien-Chou ; Huang, Ing-Jer: HPChecker: An AMBA AHB On-Chip Bus Protocol Checker with Efficient Verification Mechanisms. In: IEICE Transactions on Information and Systems, Bd. E93-D, Nr. 8, 2010, Online ISSN: 1745-1361, Print ISSN: 0916-8532, S. 2100-2108

[CJH10] Chen, Chien-Hung ; Ju, Jiun-Cheng ; Huang, Ing-Jer: A synthesizable AXI protocol checker for SoC integration. In: Proceedings of the IEEE International SoC Design Conference (ISOCC2010), 2010, ISBN: 987-1-4244-8633-5, S. 103-106

[YCH10] Yang, Fu-Ching ; Chiang, Cheng-Lung, Huang, Ing-Jer: A Reverse-Encoding-Based On-Chip Bus Tracer for Efficient Circular-Buffer Utilization. In: IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Bd. 18, Nr. 5, 2010, ISSN: 1063-8210, S. 732-741

[YLK+11] Yang, Fu-Ching ; Lin, Yi-Ting ; Kao, Chung-Fu ; Huang, Ing-Jer: An On-Chip AHB Bus Tracer With Real-Time Compression and Dynamic Multiresolution Supports for SoC. In: IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Bd. 19, Nr. 4, 2011, ISSN: 1063-8210, S. 571-583

Stand der Forschung

Akademische Lösungen - Chung-Fu Kao, Ing-Jer Huang (2)

Gesamtkonzept und Komponenten

- dreiphasiger, verlustloser und in Echtzeit arbeitender Hardware-Kompressor auf Basis des Lempel-Ziv-Algorithmus
- keine Trigger, keine Trace-Qualification, keine Multi-Core-Interaktion
- Die drei Komponenten (Phasen) sind parametrierbar und modular einsetzbar.
- Instruktions-Trace-Daten bestehen zu einem großen Teil aus sich häufig wiederholende "Zeichenketten", welche hauptsächlich durch häufig ausgeführte (innerste) Schleifen im Programmfluss generiert werden.
- Diese Zeichenketten sind perfekte Kandidaten für eine LZ-Kompression.
- LZ-Algorithmen erreichen bei alleiniger Implementierung Kompressionsraten im Bereich von nur drei bis zehn.
- Durch die Kombination mit anderen Methoden wird eine erheblich höhere Kompressionsrate erzielt.

Stand der Forschung

Akademische Lösungen - Chung-Fu Kao, Ing-Jer Huang (3)

Kompressionsfaktoren

Programm	P1	P2	P3 (27)	P3 (25)	P1*P2	P1*P2*P3 (27)	P1*P2*P3 (25)	(27)/(25)
FIB	1,70	2,67	254,00	1,59	4,54	1152,46	7,20	159,96
Hanoi	4,21	2,31	308,23	0,80	9,74	3001,23	7,83	383,26
DCT	3,89	2,27	58,84	1,09	8,82	518,86	9,63	53,86
FFT	5,15	2,01	19,88	0,79	10,36	206,02	8,22	25,07
DTMF	3,89	1,78	9,17	0,99	6,93	63,57	6,86	9,26
JPEG Encoder	4,25	2,56	25,77	19,16	10,86	279,85	208,00	1,35
G.711	1,96	2,19	214,06	2,02	4,29	917,54	8,67	105,88
Dhrystone	3,51	1,63	56,66	0,90	5,72	324,34	5,15	63,04
geometr. Mittel	3,36	2,15	63,44	1,58	7,23	458,62	11,39	40,25
harmon. Mittel	3,12	2,12	32,69	1,19	6,80	254,29	8,40	30,27

Stand der Forschung

Akademische Lösungen - Milenkovic, Uzelac (1)

[MM03] Milenkovic, Aleksandar ; Milenkovic, Milena.: Stream-Based Trace Compression. In: Computer Architecture Letters, Bd. 2, 2003, ISSN: 1556-6056, S. 14-17

[MMK03] Milenkovic, Aleksandar ; Milenkovic, Milena ; Kulick, Jeffrey: N-Tuple Compression: A Novel Method for Compression of Branch Instruction Traces. In: Proceedings of the 16th International Conference on Parallel and Distributed Computing Systems (PDCS-2003), 2003, ISSN: 1049-3301, S. 49-55

[MM07] Milenkovic, Aleksandar ; Milenkovic, Milena: An Efficient Single-Pass Trace Compression Technique Utilizing Instruction Streams. In: ACM Transactions on Modeling and Computer Simulation, Bd. 17, Nr. 1, 2007, ISSN: 1049-3301, S. 1-27

[MMB07] Milenkovic, Milena ; Milenkovic, Aleksandar ; Burtscher, Martin: Algorithms and Hardware Structures for Unobtrusive Real-Time Compression of Instruction and Data Address Traces. In: Proceedings of the 2007 Data Compression Conference (DCC'07), 2007, ISBN: 978-0-7695-2791-8, ISSN: 1068-0314, S. 55-65

[UM09] Uzelac, Vladimir ; Milenkovic, Aleksandar: A Real-Time Program Trace Compressor Utilizing Double Move-to-Front Method. In: Proceedings of the 46th ACM/IEEE Design Automation Conference (DAC 2009), 2009, ISBN: 978-1-60558-497-3, ISSN: 0738-100X, S. 738-743

[UMM+09] Uzelac, Vladimir ; Milenkovic, Aleksandar ; Milenkovic, Milena ; Burtscher, Martin: Real-time, Unobtrusive, and Efficient Program Execution Tracing with Stream Caches and Last Stream Predictors. In: Proceedings of the 2009 International Conference of Computer Design (ICCD 2009), 2009, ISBN: 978-1-4244-5029-9, ISSN: 1063-6404, S. 173-178

[Uce10] Uzelac, Vladimir: Algorithms and Hardware Structures for Real-Time Compression of Program Traces. The University of Alabama in Huntsville, Dissertation, 2010, ISBN: 978-1-124-04721-8

[UM10] Uzelac, Vladimir ; Milenkovic, Aleksandar: Hardware-Based Data Value and Address Trace Filtering Techniques. In: Proceedings of the International Conference on Compilers, Architectures and Synthesis of Embedded Systems (CASES'10), 2010, ISBN: 978-1-60558-903-9, S. 117-126

[UMB+10] Uzelac, Vladimir ; Milenkovic, Aleksandar ; Burtscher, Martin ; Milenkovic, Milena: Real-time Unobtrusive Program Execution Trace Compression Using Branch Predictor Events. In: Proceedings of the International Conference on Compilers, Architectures and Synthesis of Embedded Systems (CASES'10), 2010, ISBN: 978-1-60558-903-9, S. 97-106

[MUM+11] Milenkovic, Aleksandar ; Uzelac, Vladimir ; Milenkovic, Milena ; Burtscher, Martin: Caches and Predictors for Real-Time, Unobtrusive, and Cost-Effective Program Tracing in Embedded Systems. In: IEEE Transactions on Computers, Bd. 60, Nr. 11, 2011, ISSN: 0018-9340, S. 992-1005

[UM11] Uzelac, Vladimir ; Milenkovic, Aleksandar: Hardware-Based Load Value Trace Filtering for On-the-Fly Debugging. In: ACM Transactions on Embedded Computing Systems, Bd., Nr., 2011,

Stand der Forschung

Akademische Lösungen - Milenkovic, Uzelac (2)

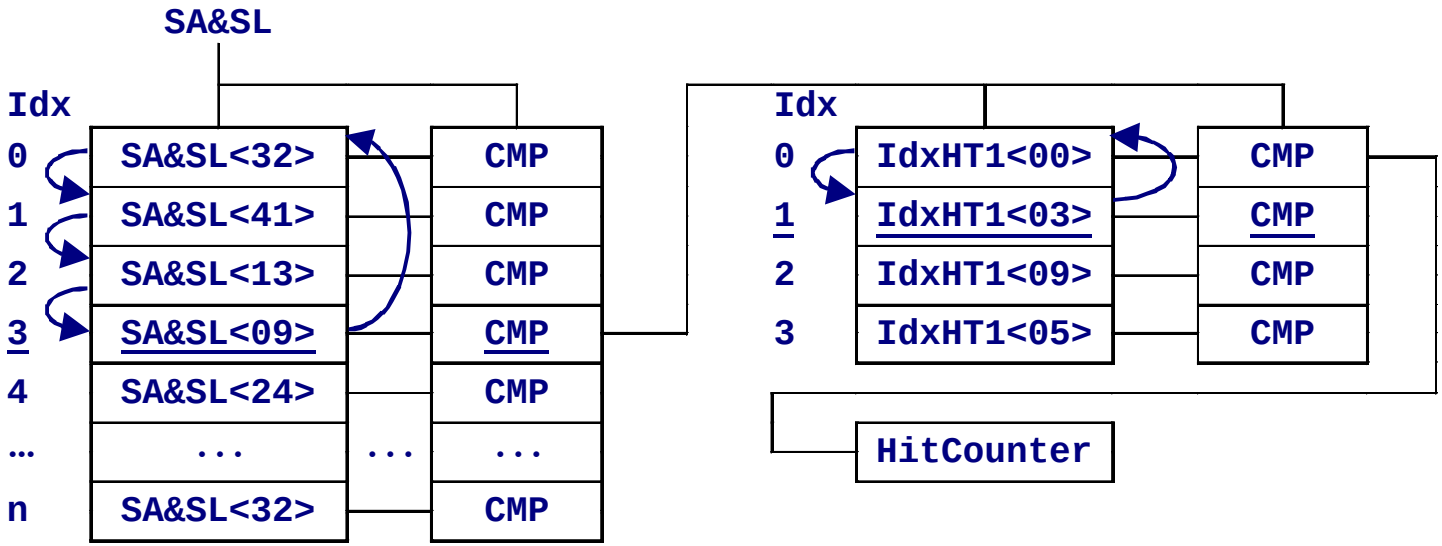
Gesamtkonzept und Komponenten

- zwei verschiedene Architekturen eines Hardware-Kompressors zur "Stream-Based Compression":
  - auf der Basis von History-Tables ("Double Move-to-Front")
  - auf der Basis von Cache-Speichern
- Ziel: Die "Stream-Based Compression" soll die Lücke schließen zwischen einfachen Ansätzen mit geringer Kompressionsrate und solchen mit hoher Kompressionsrate und hohem Aufwand.
- Ein Instruktionsstrom ist definiert als eine Menge sequentieller Operationen, die an der Zieladresse eines ausgeführten Sprungbefehls beginnt und mit dem ersten ausgeführten Sprungbefehl in dieser Sequenz endet.
- Ein Instruktionsstrom ist eindeutig gekennzeichnet durch seine Anfangsadresse (Stream Address SA) und seine Länge (Stream Length SL).
- keine Trigger, keine Trace-Qualification, keine Multi-Core-Interaktion

Stand der Forschung

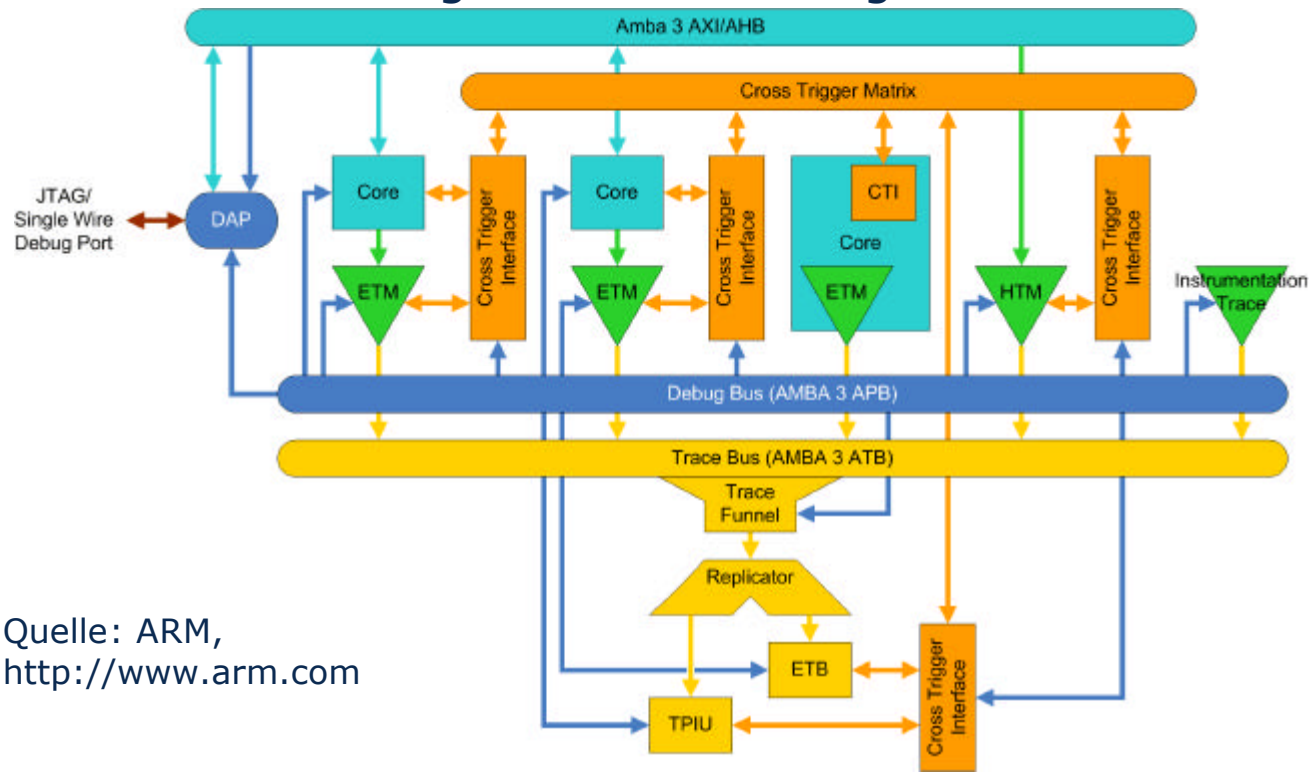
Akademische Lösungen - Milenkovic, Uzelac (3)

Double Move-to-Front: Architektur



Stand der Forschung

Industrielle Lösungen - ARM CoreSight



Quelle: ARM,
http://www.arm.com

Stand der Forschung

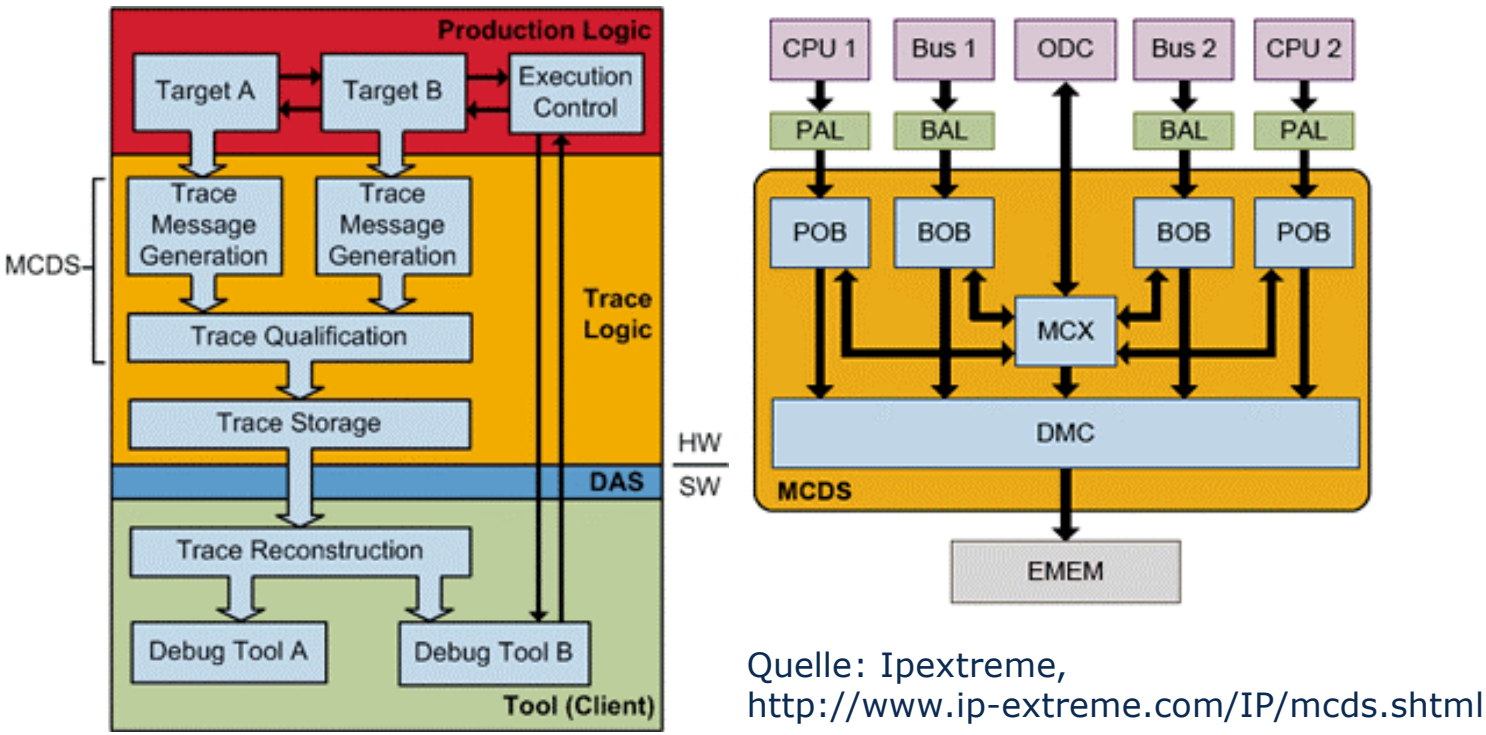
Industrielle Lösungen - Embedded Trace Macrocell

Aussagen aus ARM Whitepaper [Orm08]

- Ein reiner Instruktions-Trace ist mit relativ wenig Aufwand machbar.
- Es wird eine gute Kompression von etwa 1 Bit pro Instruktion bei einem Hardwareaufwand von 7 k Gates erreicht.
- Ein 4 KByte großer RAM kann die Trace-Daten von über 30'000 Zeilen Assembler-Code aufnehmen
- Der zyklen-akkurate Trace erhöht die Bandbreite signifikant auf etwa 4 Bit pro Instruktion.
- Daten-Trace verursacht deutlich höhere Kosten als Instruktions-Trace.
- Die mittlere Bandbreite beim Daten-Trace liegt in der Größenordnung von 1...2 Byte pro Instruktion.

Stand der Forschung

Industrielle Lösungen - Infineon MCDS



Quelle: Ipextreme, http://www.ip-extreme.com/IP/mcds.shtml

Stand der Forschung

Industrielle Lösungen - MIPS PDtrace

Übersicht

- basiert auf MIPS EJTAG, PDtrace ist zusätzlicher paralleler Port
- ist eng an den MIPS Instruktionssatz gekoppelt
- Schwerpunkt liegt mehr auf Debugging mittels Breakpoints als auf Trace
- periodisches Sampling des Programmzählers möglich
- Trace-Speicher kann on-chip oder off-chip sein
- Trace-Control-Block enthält Trigger und Verknüpfungen dieser

Kompression

- Kompression des Programmzählers mittels Differenzbildung zum Vorgänger
- keine Angaben über Kompressionsfaktoren

Stand der Forschung

Zusammenfassende Bewertung

		Kompressions-faktor (KF)	Angaben der Autoren zum Aufwand	Aufw. in kBit	KF / kGates	KF / kBits
Nexus2003	**[IEE03]**	**12**	**600'000 µm^2 (180 nm)** *(60 k Gates)*	**20**	**0,2**	**0,6**
ARM	**[Orm08]**	**32**	**7 k Gates**	**2,3**	**4,6**	**13,9**
Hopkins, McDonald-Maier	**[HM06a]**	**7**	**284'000 µm^2 (180 nm)** *(28 k Gates)*	**9,5**	**0,3**	**0,8**
Xiao Hu, Shuming Chen	**[HC07]**	**14**	**1'166'000 µm^2** *(117 k Gates)*	**39**	**0,1**	**0,4**
Chung-Fu Kao, Ing-Jer Huang	**[KHH07]**	**64** (454)	**51,7 k Gates**	**3,1**	**1,2**	**20,6**
Milenkovic, Uzelac eDMTF_192	**[UM09]**	**268**	**24,6 k Gates**	**7,0**	**10,9**	**38,3**
Milenkovic, Uzelac eDMTF_128	**[UM09]**	**206**	**16,5 k Gates**	**4,6**	**12,5**	**44,8**
Milenkovic, Uzelac Cache_128	**[MUM$^+$10]**	**184**	**63'300 µm^2 (180 nm)** *(6,3 k Gates)*	**3,8**	**29,2**	**48,4**

Stand der Forschung

Kritik und Verbesserungspotential (1)

Bewertung des Kompressionsfaktors

- Die Ermittlung der Kompressionsfaktoren erfolgt immer auf einem kleinen Satz von Benchmarkprogrammen.
- Die Auswahl der Benchmarks ist teilweise wenig repräsentativ in Bezug auf typische Anwendungen im Bereich der eingebetteten Systeme.
- Die Schnittmenge der Benchmarkprogramme aller Autoren ist klein.
- Es werden keine Angaben über Compilereinstellungen gemacht.
- Eine unmittelbare Vergleichbarkeit ist kaum gegeben.
- Es wird zu stark auf hohe Spitzenwerte orientiert, anstatt auf möglichst ausgeglichene Eigenschaften der Trace-Architekturen.

Stand der Forschung

Kritik und Verbesserungspotential (2)

Bewertung des Hardwareaufwandes

- Der Hardwareaufwand wird sehr unterschiedlich bewertet.
- Eine unmittelbare Vergleichbarkeit ist nicht gegeben.
- Bei der Bewertung des Aufwandes einer Speicherzelle gehen die Angaben weit auseinander:
  - Milenkovic und Uzelac [MUM+10] betrachten eine Zelle als einfachen Bus-Keeper, der nur 1,66 mal größer als ein NAND2 ist.
  - Kao [KHH07] setzt fast 16 Gates pro Bit (Speicherzelle + Komparator) im Wörterbuch an. Basis sind vermutlich aufwendige D-Flip-Flop.
  - Caches werden meist auf der Basis von CAM-Zellen betrachtet, wobei der Aufwand mit 2,5 bis 3 Gattern pro CAM-Zelle angesetzt wird.

Stand der Forschung

Kritik und Verbesserungspotential (3)

Skalierbarkeit und Verhältnis von Nutzen zu Aufwand

Der aktuelle Stand der Technik lässt derzeit zwei "Extreme" erkennen:

1. Einfache Lösungen mit geringem Aufwand, aber auch nur moderaten Kompressionsraten (PFCM, XOR, ...)
2. Sehr aufwendige Lösungen mit hohen Kompressionsfaktoren (große History-Tabellen, Caches oder Wörterbücher)

Forschungsfragen

Zielstellung und Metrik (1)

Ist der Kompressionsfaktor ein geeignetes Nutzen-Maß?

- Wichtig ist eine einheitliche Bezugsgröße für die Rohdatenmenge.
- Fast alle Autoren verwenden die Bit-Menge aller zugegriffenen Instruktionen.
- Aus der Sicht des Nutzers (der meist auf dem Niveau des Quellcodes arbeitet) wäre ein Maß geeigneter, welches beispielsweise die unterbrechungsfrei aufzeichenbare "Quellcodemenge" angibt (Tracedatenmenge pro Zeiteinheit).
- Probleme:
  - Die Art und Menge der Trace-Ereignisse hängt von der Architektur des Prozessors ab. (Beispiel: mit oder ohne "Predicated Instructions")
  - Die Art und Menge der Trace-Ereignisse hängt vom Compiler ab. (Beispiel: Loop-unrolling reduziert die Sprunganzahl).

Forschungsfragen

Zielstellung und Metrik (2)

Welches Maß für den Aufwand ist am geeignetsten?

- Grundsätzlich sind die selben Maße geeignet, welche für die Bewertung des Hardwareaufwandes einer Schaltung allgemein verwendet werden:
  - Chipfläche bei einer konkreten Synthese-Technologie
  - Anzahl der Gatteräquivalente
  - Anzahl der Slices bei einer FPGA-Implementierung
  - Verlustleistung
  - ...
- Die Hardware lässt sich unterteilen in Logik für Komparatoren, Trigger-Verknüpfungen, etc. und (sofern vorhanden) in Speicher in Form von History-Tabellen, Caches, Puffer-FIFOs und Trace-Speicher.
- Generell nicht angegeben und bewertet wird die Verlustleistung. Dies erscheint legitim unter der Annahme, dass die Trace-Einheit für den normalen Wirkbetrieb des Systems abgeschaltet werden kann.

Forschungsfragen

Ausnutzung der Eigenschaften von Trace-Daten

Existieren (über bereits genannte und ausgenutzte hinaus gehende) **Eigenschaften von Trace-Daten, die für eine effektivere Kompression ausgenutzt werden können?**

- Lassen sich Gemeinsamkeiten finden, wie beispielsweise typische Sprungdistanzen?
- Lassen sich typische Verteilungsmodelle daraus ableiten?
- Welche Auswahl von Benchmark-Programmen realisiert eine einheitliche und auf typische Anwendungsfälle bezogene Bewertung von Trace-Architekturen?
- Lassen sich aus der Kenntnis der für die Kompression ausnutzbaren Eigenschaften der Trace-Daten und den Ansätzen für Hardware-Kompressoren aus der Literatur effektivere Architekturen ableiten?

Forschungsfragen

Architektonische Merkmale

Wie sollte eine effektive Kodierung der Trace-Messages aussehen und schaltungstechnisch realisiert sein?

- Die Argumentation XOR versus SUB ist in der Literatur nicht einheitlich.
  - Ist eines dieser Verfahren besser?
  - Gibt es noch bessere Varianten?
- Die Effektivität wird von der Clustergröße beeinflusst.
  - Mit kleinen Clustern wird eine bessere Filterung führender Nullen erreicht.
  - Große Cluster erzeugen weniger "Follow-Bits".
  - Gibt es ein Optimum?
- History-Bitmaps werden derzeit gar nicht komprimiert.
  - Gelingt es, einen Kompressions-Ansatz für History-Bitmaps zu finden?

Zusammenfassung

Abgrenzung des Arbeitsgebietes / Schwerpunkte

- On-Chip-Trace für eingebettete Prozessoren und SoC
- Core-Trace
- Instruktions-Trace
- Multi-Core-Fähigkeit (Core-Interaktion ist jedoch kein Schwerpunkt)
- nicht-invasiv (minimal-invasiv wird mit geringer Wichtung behandelt)
- verlustlos (Verlust von Trace-Daten nur durch explizite Konfiguration erlaubt)
- echtzeitfähig (in Bezug auf den Kompressor, nicht auf Trace-Speicher und Interface)
- **Kompression der Trace-Daten**
- **Message-Generierung**

Zusammenfassung

Stand der Forschung

- Standard
- akademische Lösungen
- industrielle Lösungen

Forschungsfragen

- Welche Metrik zur Bewertung von Aufwand und Nutzen ist geeignet?
- Existieren Eigenschaften von Trace-Daten, die für eine effektivere Kompression ausgenutzt werden können?
- Gelingt es, eine Trace-Architektur bei ausgewogenem Aufwand-Nutzen-Verhältnis skalierbar auszulegen?
- Wie sollte eine effektive Kodierung der Trace-Messages aussehen sein?