



**TECHNISCHE
UNIVERSITÄT
DRESDEN**

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

Konzeption und Realisierung einer Testplattform zur Analyse von seriellen Hochgeschwindigkeitsschnittstellen für SoC Debug-Aufgaben

Diplomverteidigung

Robert Ramm

Robert.Ramm2@mailbox.tu-dresden.de

Dresden, 18.7.2012

Gliederung

1 Einführung

Trace-Datenerfassung

2 Problemstellung

3 Testumgebung

Plattformauswahl

Aufbau

4 Testszenarien

5 Ergebnisse / Erkenntnisse

6 Zusammenfassung

Einführung

- Sinkende Beobachtbarkeit durch steigende Integration der Systemkomponenten → SoC
- Steigende Nutzung der Trace-Datenerfassung
→ Aufzeichnung des Programm- und Datenflusses
- Anwendungsabhängig 0.8 Bit bis 1 Byte an Trace-Daten pro Instruktion
→ Bei 200 – 300 MHz entsprechendes Datenaufkommen von mehrerem 100 MByte/s

Einführung(2)

Problem : Behandlung der entstehenden Datenvolumen
→ 2 verschiedene Ansätze

On-Chip Trace :

- Speicherung der Daten in einem Trace-Puffer
- Übertragung der Daten nach Beendigung der Anwendung über langsame Debug-Schnittstelle

Off-Chip Trace :

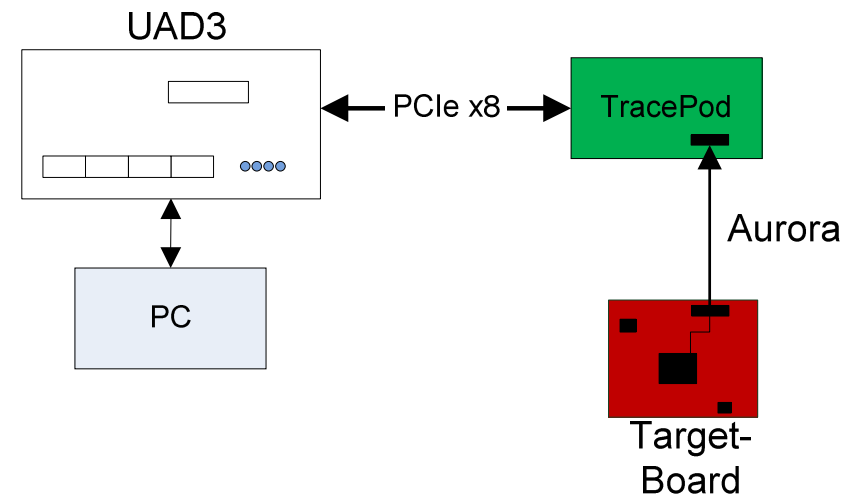
- Übertragung der entstehenden Daten über breitbandige Schnittstelle
- zunehmende Verwendung serieller Gigabit-Schnittstellen inkl. entsprechender Protokolle

Problemstellung

- Dedizierte Zugangshardware für Empfang der Trace-Daten notwendig
- Kein Test an realem Target möglich, da parallel zu diesem entwickelt → Entwicklung eines Emulationssystems notwendig

Anforderungen

- Ermittlung der Parameter-Grenzen für eine sichere Übertragung
- Nachbildung des Trace-relevanten Verhaltens
- Gezielte Stimulierbarkeit der Trace-Schnittstelle
- Steuerung durch Host-PC

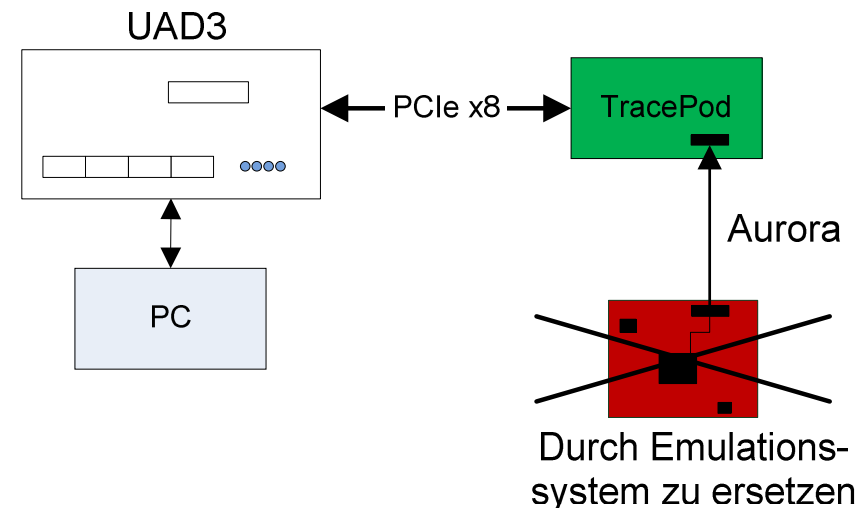


Problemstellung

- Dedizierte Zugangshardware für Empfang der Trace-Daten notwendig
- Kein Test an realem Target möglich, da parallel zu diesem entwickelt → Entwicklung eines Emulationssystems notwendig

Anforderungen

- Ermittlung der Parameter-Grenzen für eine sichere Übertragung
- Nachbildung des Trace-relevanten Verhaltens
- Gezielte Stimulierbarkeit der Trace-Schnittstelle
- Steuerung durch Host-PC



Aurora Simplex Protokoll

- Link-Training aus 3 Phasen : Alignment, Channel-Bonding, Channel-Verifikation
→ Statusübertragung in Gegenrichtung über Sideband-Signale
- Jeweils ein Start- und Stop-Symbol pro Frame
→ sehr geringer Protokolloverhead
- Idle-Sequenz bestehend aus Byte-Alignment- und Channel-Bonding-Symbolen
- Taktkompensationssequenzen mind. alle 10000 Symbole

Anforderungen Testplattform

- Verfügbarkeit serieller Gigabit-Transceiver
 - mindestens 2 Stück, besser 4
 - maximale Bitrate bei mind. 2.5 GBit/s, besser 5 GBit/s
 - Variation der Ausgangspegel
 - Parametrisierung der Emphasis (Signalvorverzerrung)
- Transceiver dynamisch einstellbar in Bitrate und verwendeten Lanes
- Frei konfigurierbar in übertragenem Protokoll
- Schnittstelle zur Kommunikation mit Host-PC
- Normale IO/s für Sideband-Signale

Plattformauswahl

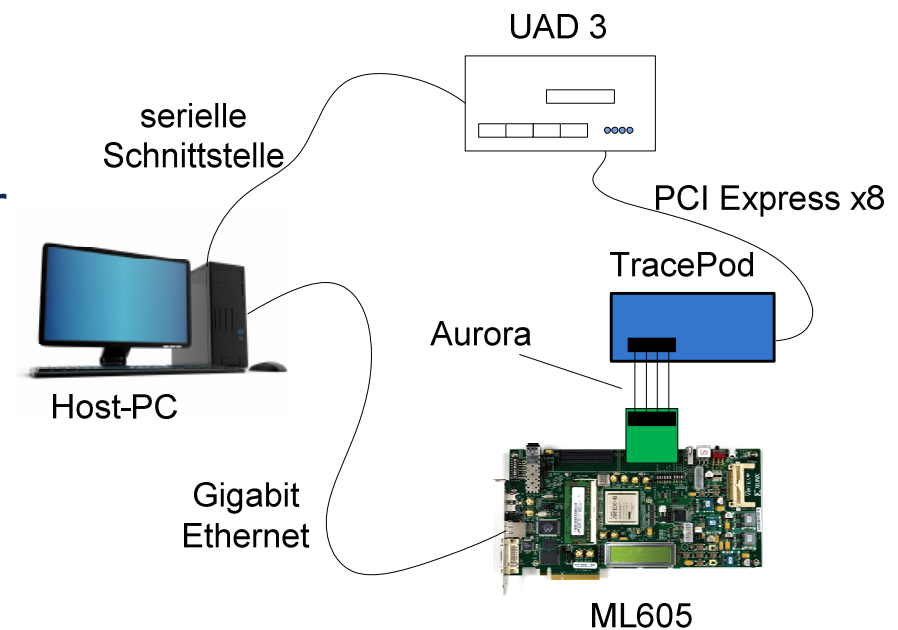
- Mikrocontrollerbasierte Plattform nicht geeignet
 - zu geringe Flexibilität
 - Trace-Protokolle für Implementierung in Hardware entworfen
 - kein Board auffindbar, dass alle Bedingungen erfüllt
- FPGA-Plattform durch höheren Flexibilität und Geschwindigkeit besser geeignet
- Xilinx aufgrund der IP-Core Unterstützung zu bevorzugen
 - Altera Stratix prinzipiell auch möglich, aber Portierung notwendig → Aufwand und Fehlerrisiko

Plattformauswahl (2)

Plattform	Ausschlusskriterium
Warp FPGA Board (Virtex 4 FX100)	- Keine Aurora IP-Core Unterstützung für Virtex 4
Mars MX2 (Spartan 6 LX45T)	- Zusatzplatine für Spannungsversorgung, Konnektoren notwendig
SX95 (Virtex 5 SX95T)	- Keine Schnittstelle für Kommunikation mit Host-PC
SP605 Evaluation Kit (Spartan 6 LX45T)	- Nur eine Lane pro Konnektor ➔ Aufwendige Adapterplatine
SP623 Characterization Kit (Spartan 6 LX150T)	- Nur serielle Schnittstelle für Host-PC Kommunikation vorhanden - FPGA überdimensioniert - Mit 4000\$ zu teuer
ML505 Evaluation Platform (Virtex 5 LX50T)	- Nur eine Lane pro Konnektor ➔ Aufwendige Adapterplatine
ML605 Evaluation Kit (Virtex 6 LX240T)	- nicht vorhanden

Aufbau Testumgebung

- Initiierung und Steuerung der Testfälle durch Host-PC
- ML605 zur Generierung der Trace-Pakete und Implementierung der Aurora-Schnittstelle
- Auswertung empfangener Daten direkt in Firmware der UAD3



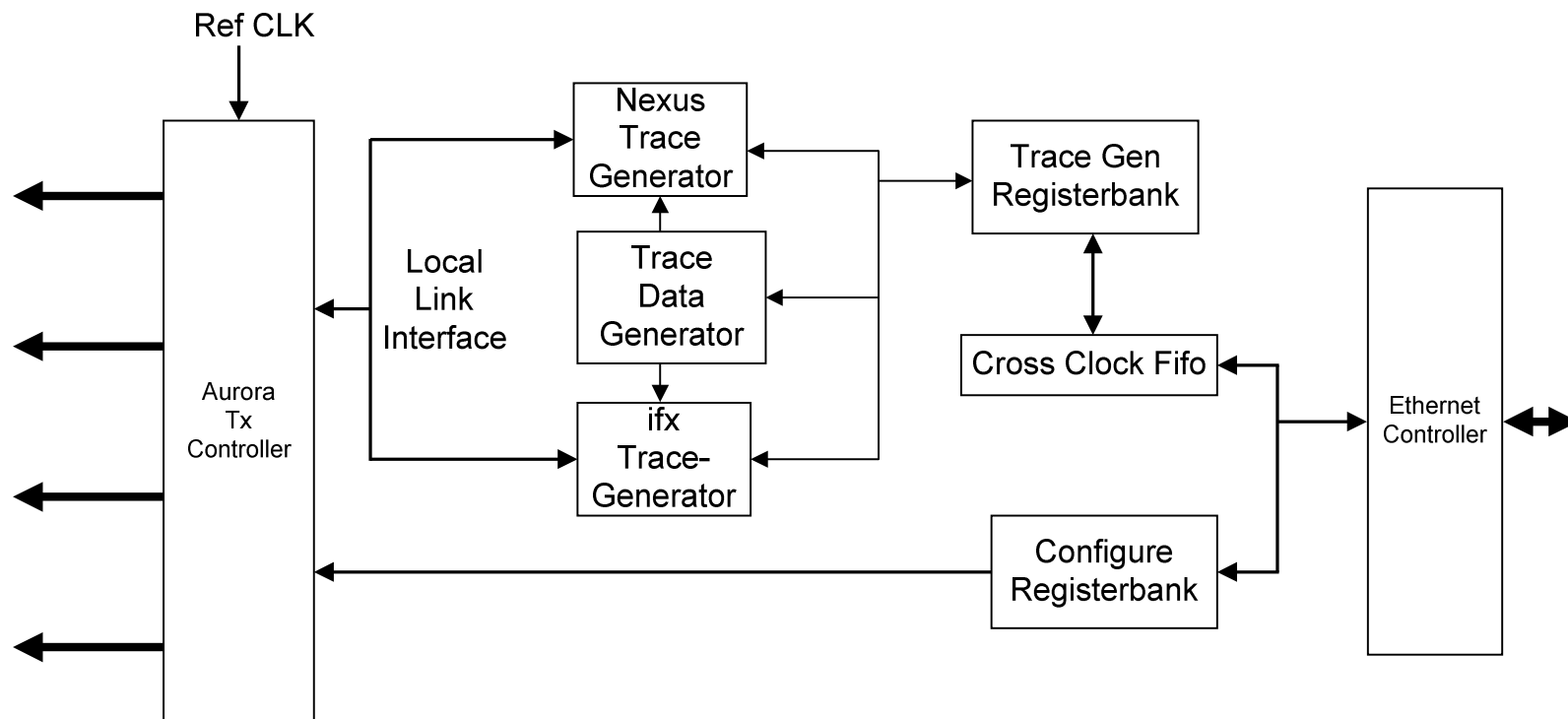
Testszzenarien – PRBS Tests

- Ermittlung der Qualität des Übertragungskanals
- Erzeugung von pseudozufälligen Bitsequenzen im Transmitter
 - wortweise Auswertung eingehender Daten im Receiver
 - keine Ermittlung der BER möglich, aber Aussage über Qualität der Übertragung
- Übertragungsdatenrate für Tests bis zu 5 GBit/s
 - doppelte Zieldatenrate
- Variation der Ausgangstreiberstärke nach unten zur Simulation störender Einflüsse

Testszzenarien – Synthetische Trace-Tests

- Emulation des eigentlichen Targets
- Prüfung der logischen Funktionalität des TracePods und der weiteren Verarbeitung der Daten bis zum Trace-Speicher
- Nachbildung unterschiedlicher Datenlasten
- Auswertung der Daten im Trace-Speicher

FPGA Design



Test Aurora TracePod

Connect disconnect

Update Memory Window

Select Testcase

- Bit Level Test
- Ifx Trace Test
- Nexus Trace Test

Aurora Status ML605

- Aurora Reset
- Aurora Aligned
- Aurora Bonded
- Aurora Verified

Aurora Status TracePod

- Reset Off
- Aligned
- Not Bonded
- Verified
- Channel Up

Ifx Trace Options

- CRC Error Injection
- 115 Packets
- Random Data Enable
- CRC Enabled
- ifx complex test

Nexus Trace Options

- 0 Packets
- Symbols per Packet : 0
- variable Packet Length
- Random Data Enable
- CRC Enabled
- CRC Error Enable
- nexus complex test

Bit Level Options

- User Data
- errors
- Reset
- GO
- Error Duration : 0
- User Data
- errors
- Reset
- GO
- Error Duration : 0
- User Data
- errors
- Reset
- GO
- Error Duration : 0

Aurora & GTX Config

- Lane Select: Lane 0, Lane 1, Lane 2, Lane 3
- Lane Mapping: Lane 0, Lane 1, Lane 2, Lane 3
- Bit Transfer Rate: 2.50000 Gbit/s
- Output Driver Strength: 810 mV
- Post Emphasis: 0
- Pre Emphasis: 0
- Equalization Control: 7
- ML605 Reset
- TracePod Reset
- Init PLL
- Connect Automatically

Address	Value
0x00	0x00000001
0x01	0x00000000
0x02	0x00000000
0x03	0x00000000
0x04	0x00000000
0x05	0x00000000
0x06	0x00000414
0x07	0x00000000
0x08	0x00000000
0x09	0x00000000
0x0A	0x00000000
0x0B	0x08000263
0x0C	0x000001E4
0x0D	0x00000F27
0x0E	0x00000000
0x0F	0x00007A73
0x10	0x000096ED
0x11	0x00000000
0x12	0x00000000
0x13	0x00000000
0x14	0x00000000
0x15	0x00000000
0x16	0x00000000
0x17	0x00000000
0x18	0x00000000
0x19	0x00000000
0x1A	0x00000000
0x1B	0x00000000
0x1C	0x00000000
0x1D	0x00000000
0x1E	0x00000000
0x1F	0x00000000
0x20	0x00000000
0x21	0x00000000
0x22	0x00000000
0x23	0x00000000
0x24	0x00000000
0x25	0x00000000
0x26	0x00000000
0x27	0x00000000
0x28	0x00000000
0x29	0x00000000
0x2A	0x00000000
0x2B	0x00000000
0x2C	0x00000000
0x2D	0x00000000
0x2E	0x00000000
0x2F	0x00000000

```

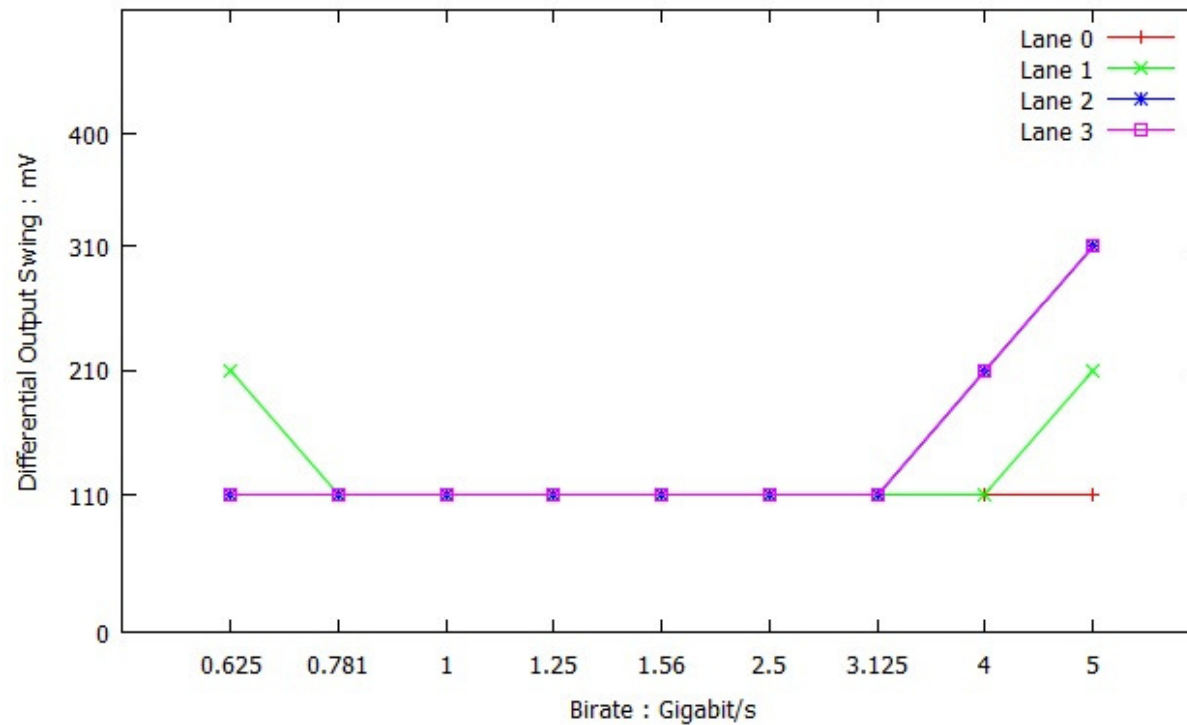
Connection to UR03 established!
Connection to ML605 established!!!
-----
Starting Complex Test of IFX Decoder
Sending different number of IFX Packets without CRC.....passed
Sending 100 MByte of Data per IFX Packets without CRC.....passed
Sending 250 MByte of Data per IFX Packets without CRC.....passed
Sending 500 MByte of Data per IFX Packets without CRC.....passed
Sending 1 GByte of Data per IFX Packets without CRC.....passed
Sending 2 GByte of Data per IFX Packets without CRC.....passed
Sending 4 GByte of Data per IFX Packets without CRC.....passed
-----

```

Ergebnisse PRBS

- Auftreten von Übertragungsfehlern bis zu 3.125 GBit/s nur bei niedrigstem Ausgangspegel (110mV) und optimaler Equalization
- Für 4 und 5 GBit/s bei ab 310 mV keine Fehler
→ typische Ausgangspegel 800mV
- Lanes ohne Vias zeigen bestes Übertragungsverhalten
- Equalization und Post-/Pre-Emphasis müssen immer genau auf jeweiligen Kanal abgestimmt werden

Ergebnisse PRBS(2)



minimale Bitrate für „fehlerfreie“ Übertragung bei optimaler Equalization

Ergebnisse Synthetische Trace Tests

- Zu Beginn kein Verbindungsaufbau möglich
→ vermutlich Glitch auf der asynchronen Reset-Leitung während der Zustandsumschaltung
- Kein Empfangen von Daten im Trace-Speichers während der ersten Testläufe
→ fehlerhafte Implementierung des Trace-Frame-Dekoders
- Im Folgenden keine weiteren Übertragungsfehler
→ Übertragung bei unterschiedlichen Lasten und Datenraten ohne weitere Fehler
- Bandbreite der TracePod-Anbindung auch bei 4 PCIe Lanes hinreichend groß

Zusammenfassung

- Elektrische Verbindung genügt Ansprüchen bzgl. Robustheit
 - Übertragungsraten von bis 5 GBit/s unter Laborbedingungen möglich
 - Betriebsparameter für sichere Kommunikation auch in realen Targets möglich
- Übertragung von Trace-Daten entsprechend den Anforderungen problemlos möglich
- Für zukünftige Erhöhung der Bitrate zusätzliche Betrachtungen des Übertragungskanals bzgl. Dispersion usw. notwendig

Quellen

- R. Höller et al. "Eine neuartige Lösung für Test und Debugging in vernetzten eingebetteten Systemen". In: Elektrotechnik & Informationstechnik. Vol. 4. Springer- Verlag, 2010, pp. 91–97.
- Aurora Protocol Specification. SP002 (v2.0). Xilinx. Sept. 2007.
- Steffen Köhler, Rainer G. Spallek, and Jens Braunes. "Trace-Buffer Size Requirements for Application Context Reconstruction in Embedded Microprocessors". In: The 2011 System, Software, SoC and Silicon Debug Conference (S4D). Oct. 2011.
- Andrea Martin. "LONG-TERM TRACE ETMv3". LAUTERBACH NEWS 2009. Lauterbach Development Tools. Mar. 2009.
- Embedded Trace Buffer Technical Reference Manual. Rev 0. Feb. 2002.

Vielen Dank für die Aufmerksamkeit