



TECHNISCHE  
UNIVERSITÄT  
DRESDEN

# Implementierung eines TPIU- Streamdekoders in VHDL

Albert Schulz

Dresden, 07.04.2016



DRESDEN  
concept  
Exzellenz aus  
Wissenschaft  
und Kultur

## Gliederung

1. Motivation
2. Zielarchitektur
3. CoreSight
4. Trace Port Interface Unit (TPIU)
5. TPIU-Stream (Continuous Mode)
6. Implementierung
7. Ergebnisse/Ausblick
8. Quellen

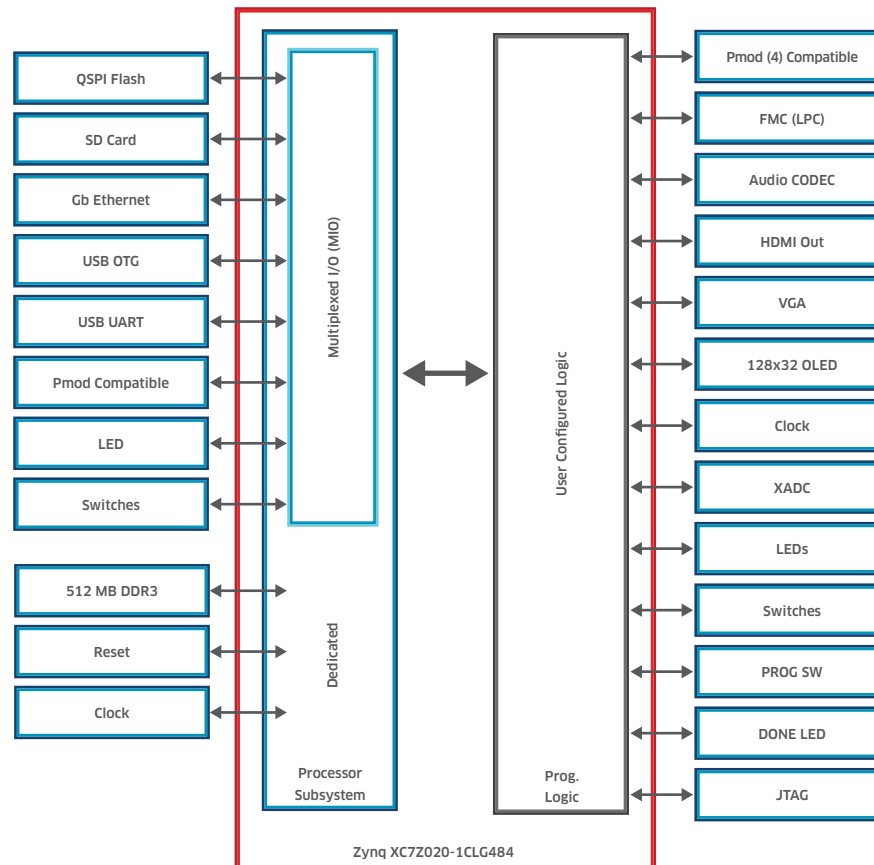
## 1. Motivation

- Tracing erlaubt Mitschneiden des Programmablaufs
  - führt zu großen Datenmengen, insbesondere bei Tracing von Speicherzugriffen und deren Inhalten
- Zynq-Plattform ermöglicht Verarbeitung on-chip mit hoher Bandbreite, um Datenmenge frühzeitig zu filtern/reduzieren

## 2. Zielarchitektur

- ZedBoard mit Zynq-7000 All Programmable SoC
- Zynq-7000 beinhaltet:
  - ARM Cortex-A9 MPCore Zweikernprozessor mit CoreSight
  - programmierbare Logik

## 2. Zielarchitektur

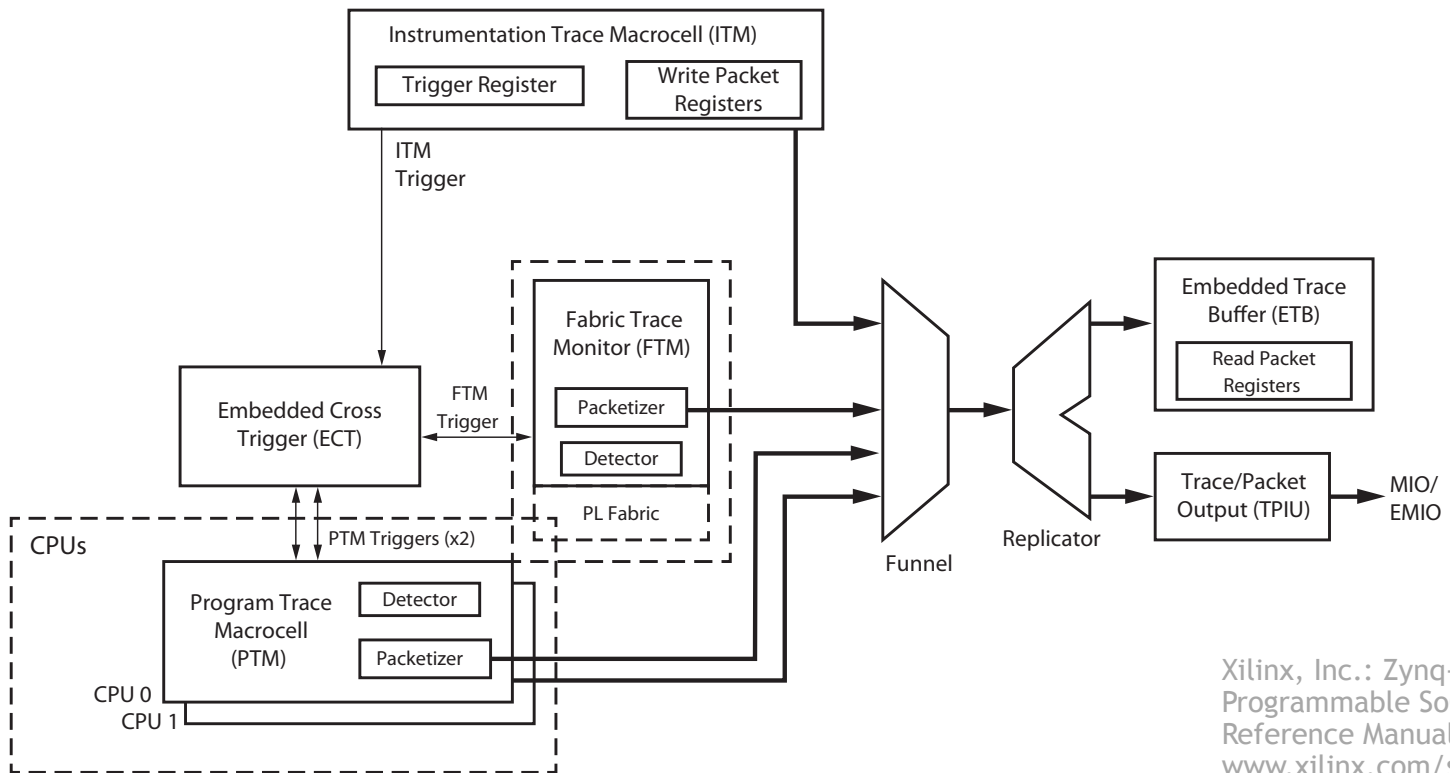


Avnet, Inc.: Avnet Product Brief ZedBoard, [http://zedboard.org/sites/default/files/product\\_briefs/PB-AES-Z7EV-7Z020\\_G-v12.pdf](http://zedboard.org/sites/default/files/product_briefs/PB-AES-Z7EV-7Z020_G-v12.pdf), Abruf: 26.01.2016

### 3. CoreSight

- Technologie von ARM für Trace- und Debuggingzwecke
- Analyse Prozessor und programmierbarer Logik
- Komponenten werden in vier Klassen eingeteilt:
  - Trace-Quellen
  - Trace-Senken
  - Trace-Verbindungen
  - Zugang und Kontrolle
- Quellen kann eindeutige ID zugeordnet werden

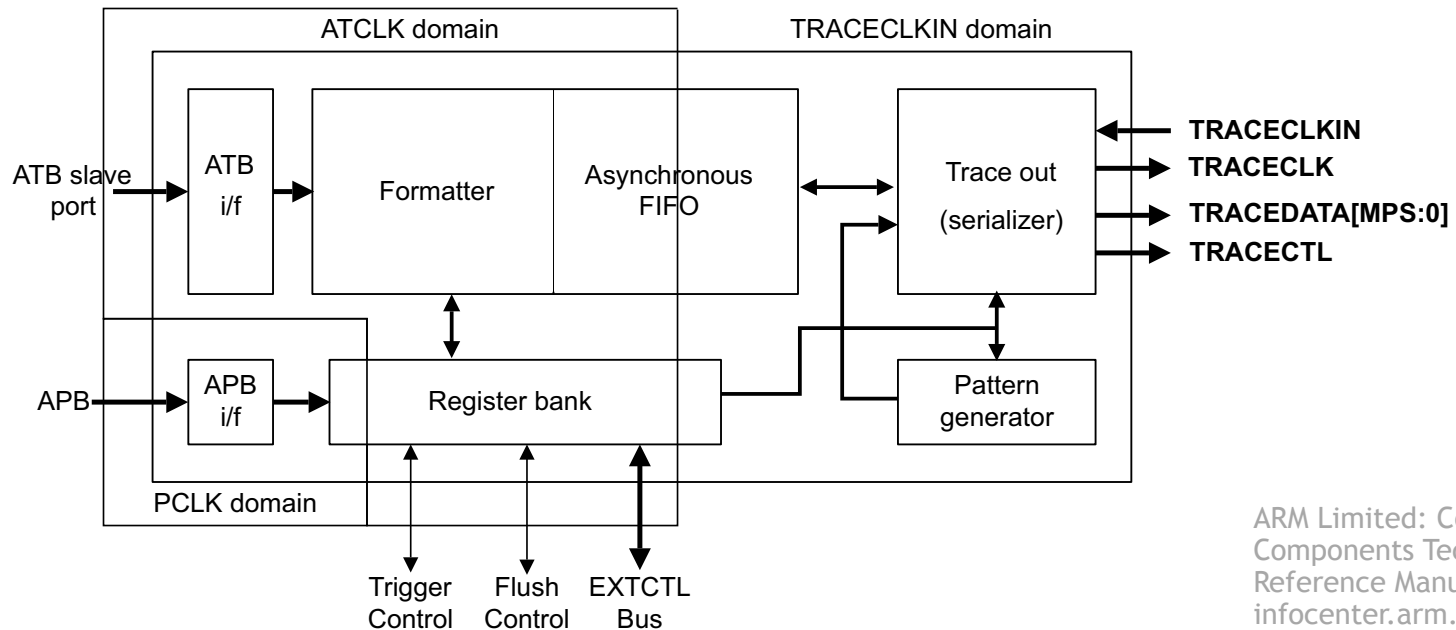
### 3. CoreSight



Xilinx, Inc.: Zynq-7000 All Programmable SoC Technical Reference Manual, [http://www.xilinx.com/support/documentation/user\\_guides/ug585-Zynq-7000-TRM.pdf](http://www.xilinx.com/support/documentation/user_guides/ug585-Zynq-7000-TRM.pdf), Ab-ruf: 26.01.2016

## 4. Trace Port Interface Unit (TPIU)

- Schnittstelle für Trace-Daten verschiedener Quellen



ARM Limited: CoreSight™  
 Components Technical  
 Reference Manual [http://infocenter.arm.com/help/topic/com.arm.doc.ddi0314h/DDI0314H\\_coresight\\_components\\_trm.pdf](http://infocenter.arm.com/help/topic/com.arm.doc.ddi0314h/DDI0314H_coresight_components_trm.pdf), Abruf:  
 27.01.2016



## 4. Trace Port Interface Unit (TPIU)

- Trigger signalisieren Eintreten eines bestimmten Ereignisses, ausgelöst z.B.:
  - durch Cross Trigger Interface (CTI) ausgelöst
  - Ende eines Flushs
- unbrauchbar bei mehreren Trace-Quellen aufgrund Zeitverzögerungen zwischen Eintreten des Ereignisses und Ausgabe der zugehörigen Daten
  - Trigger meist innerhalb der Daten kodiert

## 4. Trace Port Interface Unit (TPIU)

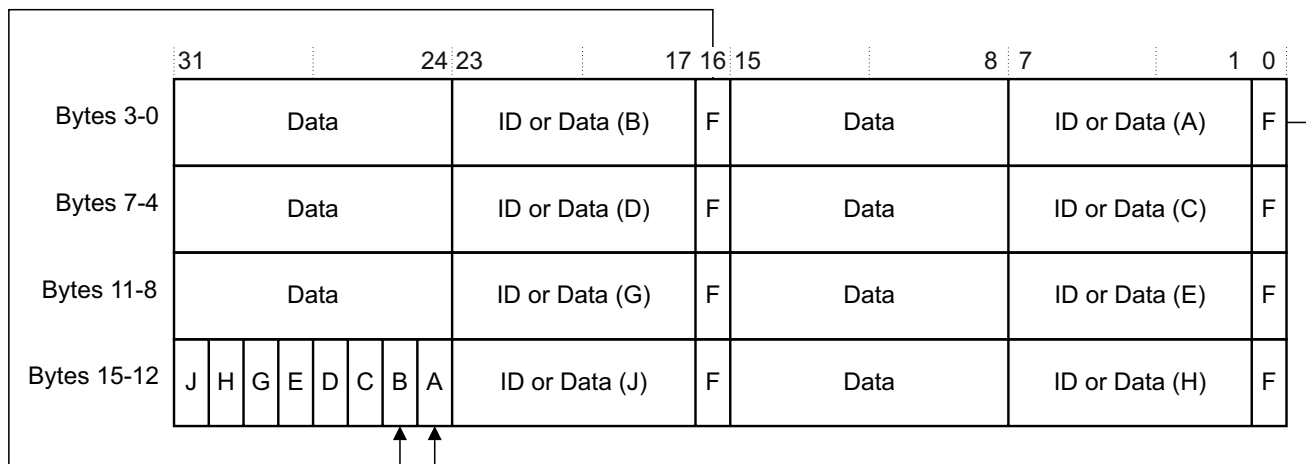
Modus	Beschreibung
Bypass	<ul style="list-style-type: none"><li>• bei einer Trace-Quelle nutzbar</li><li>• Keine Formatierung der Eingangsdaten</li><li>• Signalisierung von Triggern und validen Daten über TRACECTL-Pin</li></ul>
Normal	<ul style="list-style-type: none"><li>• Kodierung von IDs und Daten in Trace-Stream</li><li>• Signalisierung von Triggern und validen Daten über TRACECTL-Pin</li></ul>
Continuous	<ul style="list-style-type: none"><li>• Kodierung von IDs und Daten in Trace-Stream</li><li>• Kodierung von Triggern und validen Daten ebenfalls im Trace-Stream</li></ul>

## 5. TPIU-Stream (Continuous Mode)

- TPIU Protokoll spezifiziert 16-Byte-Frames
- Kodierung von IDs der Trace-Quellen und zugehörige Daten

## 5. TPIU-Stream (Continuous Mode)

- Frame besteht aus:
  - 7 Byte Daten
  - 8 Byte Daten oder IDs
  - 1 Byte mit 8 Informationsbits



ARM  
Limited:  
CoreSight™  
Components  
Technical  
Reference  
Manual  
[http://  
infocenter.a  
rm.com/  
help/topic/  
com.arm.do  
c.ddi0314h/  
DDI0314H\\_c  
oresight\\_  
components  
\\_trm.pdf](http://infocenter.arm.com/help/topic/com.arm.doc/ddi0314h/DDI0314H_coresight_components_trm.pdf),  
Abruf:  
27.01.2016

## 5. TPIU-Stream (Continuous Mode)

- Synchronisationspakete ( $0x7FFFFFFF$ )
  - periodisch zwischen Frames ausgegeben
  - an 4-Byte Grenze ausgerichtet
- Idle-Pakete ( $0x7FFF$ )
  - signalisieren, dass keine Trace-Daten zur Verfügung stehen
  - treten auch innerhalb von Frames auf
  - an 2-Byte Grenze ausgerichtet

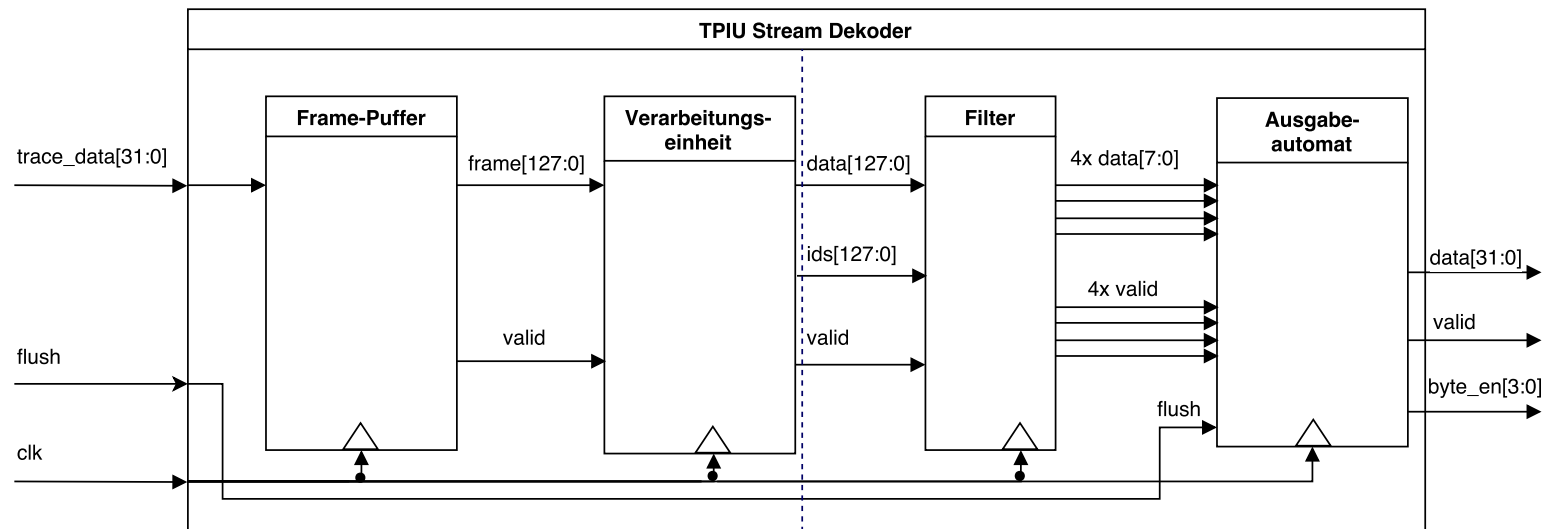
## 5. TPIU-Stream (Continuous Mode)

- Wahl der Trace-Quellen-ID frei, bis auf folgende Ausnahmen

ID	Beschreibung
0x00	kennzeichnet ungültige Quelle (zur Vervollständigung von Frames)
0x7F	Steht in Konflikt mit Sync- und Idle-Paketen
0x7D	Kennzeichnet Trigger im Trace-Stream
0x7E, 0x70–0x7C	Reserviert

## 6. Implementierung

- IP-Core aus vier synchron gekoppelten Automaten (mit gleichem Takt)



## 6. Implementierung

- Frame-Puffer:
  - speichert 16-Byte-Frames aus 4-Byte Stream
  - Synchronisierung über Sync-Pakete
  - Verwerfen von Sync- und Idle-Paketen
  - Ausgabe der (nicht dekodierten) 16-Byte-Frames
  - Gültige Daten werden über Valid-Bit signalisiert



## 6. Implementierung

- Verarbeitungseinheit:
  - Dekodiert Frames und stellt Zusammenhang von Daten und IDs her
  - letzte ID des vorhergehenden Frames wird gespeichert
  - Ausgabe von Daten und dazu zugehörigen IDs (jeweils über 16-Byte Interface)
  - Ungültigen Daten wird ID 0x00 zugeordnet
  - Gültige Ausgabe über Valid-Bit signalisiert

## 6. Implementierung

- Filter:
  - Filterung der Trace-Daten auf eine festgelegte ID
  - Gültige Daten und IDs am Eingang werden gespeichert
  - gefilterte Ausgabe des 16-Byte Frames in 4-Byte Teilen
  - Ausgabe über 4x Byte-Interfaces
  - Gültigkeit jedes Bytes über Valid-Bit angegeben

## 6. Implementierung

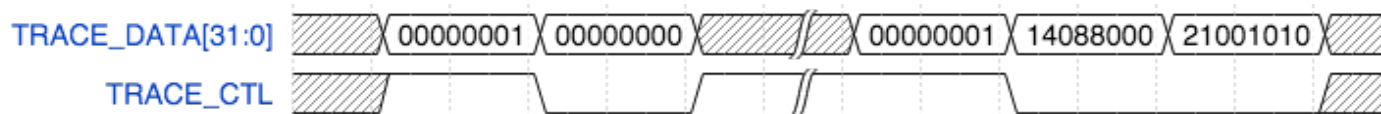
- Ausgabeautomat:
  - puffert vier eingehende, gültige Daten-Bytes
  - Ausgabe von 4-Byte Datenpaketen pro Takt
  - Gültigkeit über Valid-Bit und 4-Bit `byte_en` Leitung signalisiert
  - Zusätzlich Flush-Signal um Ausgabe der gepufferten Bytes zu erzwingen

## 7. Ergebnisse/Ausblick

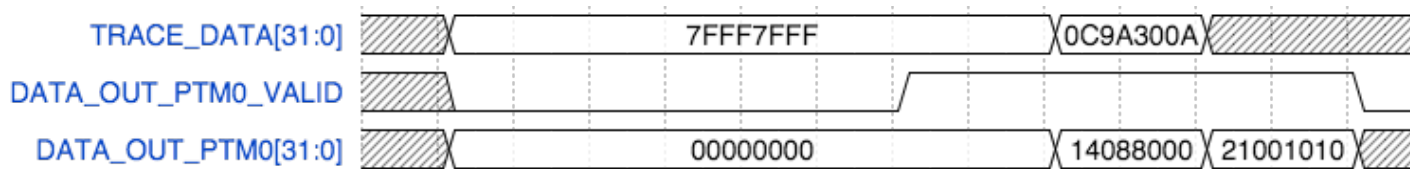
- Software um CoreSight zu aktivieren/konfigurieren
- Erzeugung von Trace-Daten mittels While-Schleife
- Analyse der Trace-Daten an TPIU bzw. eigenem IP-Core über ChipScope

## 7. Ergebnisse/Ausblick

- Vergleich der Trace-Daten Ausgabe in Bypass-Modus und dekodierte Daten im Continuous-Modus liefert gleiche (und ähnliche) Daten



Daten (Bypass-Modus)



Daten (dekodiert, Continuous-Modus)

## 7. Ergebnisse / Ausblick

- Funktionsweise des Dekoders nachgewiesen
- Filterung nach ID als Beispiel für einfache Verringerung der Datenmenge
- Grundlage für weitere Verarbeitung/Filterung der eigentlichen Trace-Daten gelegt

## 8. Quellen

- [1] ARM LIMITED: ARM CoreSight™ Architecture Specification v2.0, [http://infocenter.arm.com/help/topic/com.arm.doc.ih0029d/IH0029D\\_coresight\\_architecture\\_spec\\_v2\\_0.pdf](http://infocenter.arm.com/help/topic/com.arm.doc.ih0029d/IH0029D_coresight_architecture_spec_v2_0.pdf), Abruf: 26.01.2016
- [2] Xilinx, Inc.: Zynq-7000 All Programmable SoC Technical Reference Manual, [http://www.xilinx.com/support/documentation/user\\_guides/ug585-Zynq-7000-TRM.pdf](http://www.xilinx.com/support/documentation/user_guides/ug585-Zynq-7000-TRM.pdf), Abruf: 26.01.2016
- [3] Alex Bereza: Coresight-Zugang auf dem ZedBoard, 26.08.2015
- [4] Uwe Hermann, Jpa: arm\_tpiu [http://sigrok.org/wiki/Protocol\\_decoder:Arm\\_tpiu](http://sigrok.org/wiki/Protocol_decoder:Arm_tpiu), Abruf: 26.01.2016
- [5] Avnet, Inc.: Avnet Product Brief ZedBoard™ [http://zedboard.org/sites/default/files/product\\_briefs/PB-AES-Z7EV-7Z020\\_G-v12.pdf](http://zedboard.org/sites/default/files/product_briefs/PB-AES-Z7EV-7Z020_G-v12.pdf), Abruf: 26.01.2016
- [6] ARM Limited: CoreSight™ Components Technical Reference Manual [http://infocenter.arm.com/help/topic/com.arm.doc.ddi0314h/DDI0314H\\_coresight\\_components\\_trm.pdf](http://infocenter.arm.com/help/topic/com.arm.doc.ddi0314h/DDI0314H_coresight_components_trm.pdf), Abruf: 27.01.2016

