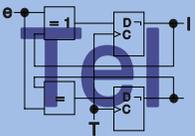


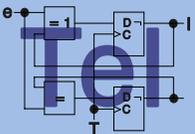
# Untersuchungen zu Hochgeschwindigkeits- transfertechniken für den Debug-Zugang in eingebetteten ARM/XScale-Prozessorsystemen über den Test-Access-Port (JTAG)

Stephan Hartmann

`stephan.hartmann@inf.tu-dresden.de`



- Motivation
- Grundlagen zu JTAG
  - Signalverlauf
  - TAP-Protokoll
- Ist-Zustand des Debug-Zuganges für die XScale-Architektur
- Kontinuierlicher Datentransfer
- Ausblick

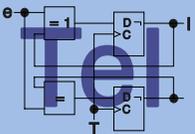


## ➤ Beweggründe:

- Hochgeschwindigkeitszugriff spielt beim Software-Debugging in eingebetteten Systemen besondere Rolle → Download-Rate, Echtzeit-Tracing
- optimale Ausnutzung des Maximalpotenzials unter Berücksichtigung verschiedener Chip-Eigenschaften

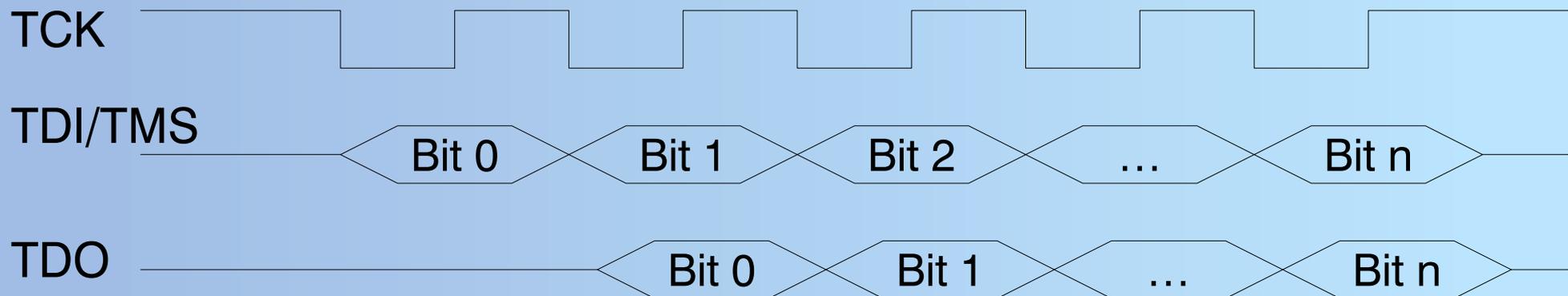
## ➤ Schlussfolgerungen:

- Analyse des TAP-Protokolls bzgl. Signalfluss, Signalintegrität und Signalregenerierung bei hohen Taktraten
- Garantie eines kontinuierlichen Datenflusses
- Bereitstellungen flexibler Konfigurationsmöglichkeiten der Transferparameter des Test-Access-Port

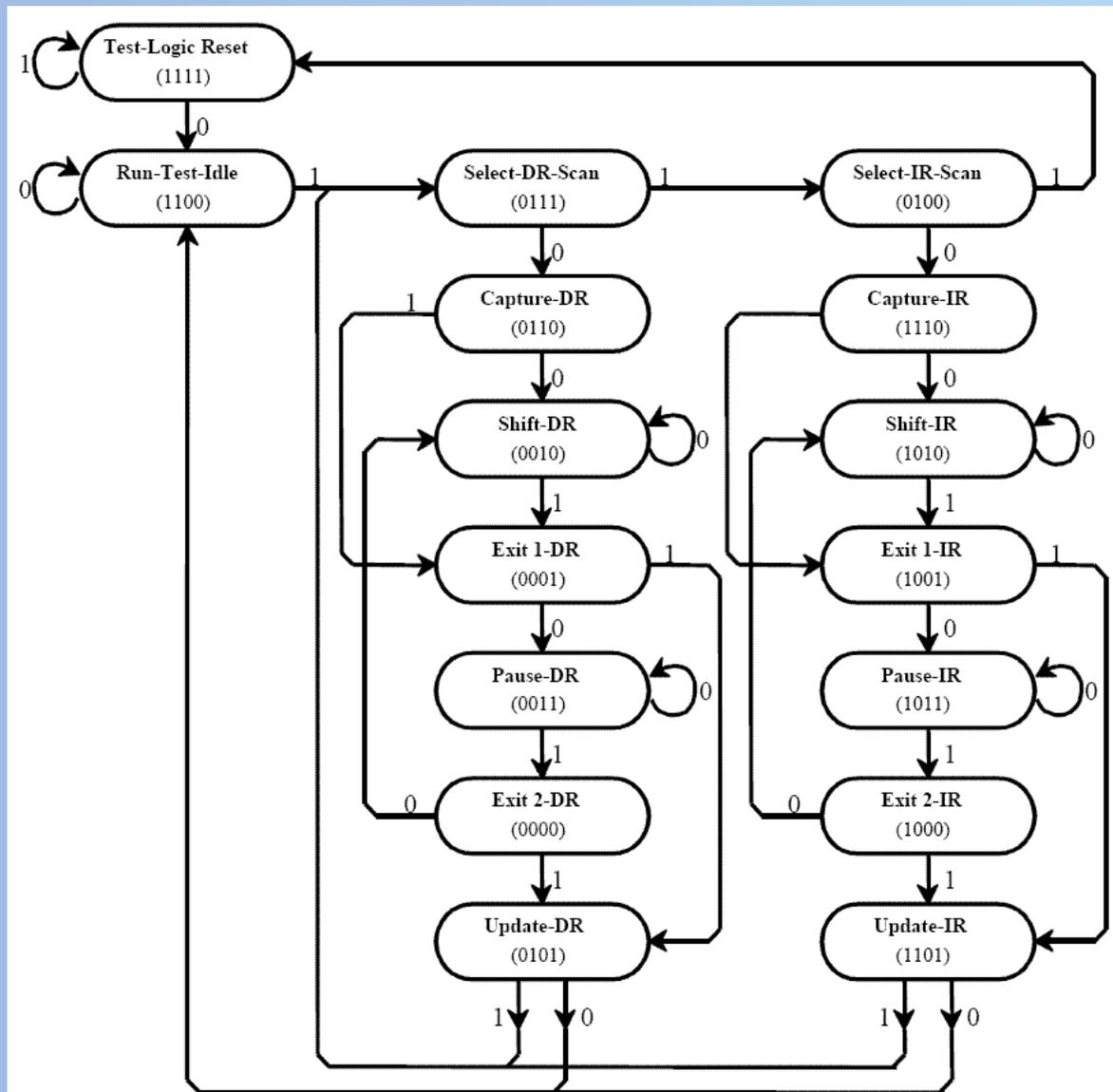


➤ Signale:

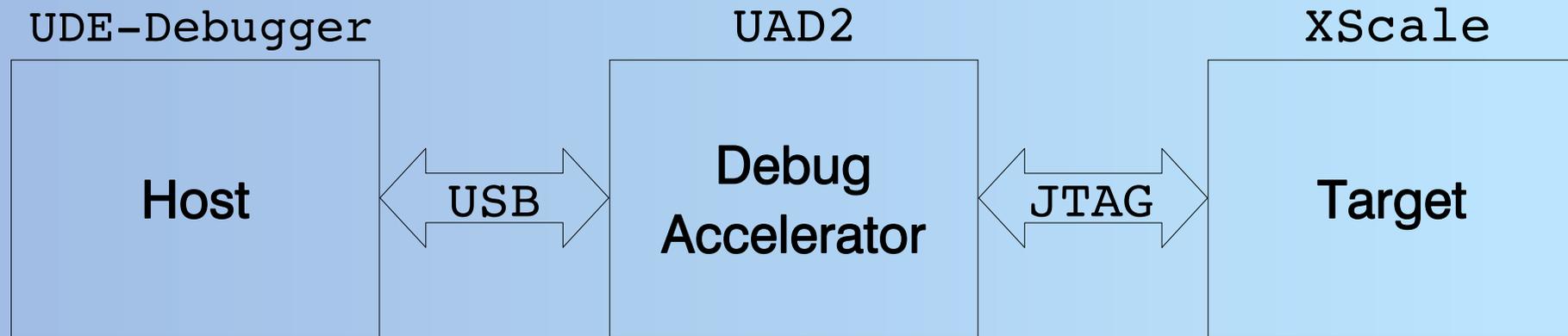
- TCK - Test Clock → Takt für den TAP-Controller
- TMS - Test Mode Select → Steuerung des TAP-Controllers
- TDI - Test Data Input → serieller Dateneingang
- TDO - Test Data Output → serieller Datenausgang
- RTCK - Return Test Clock (optional)
- nTRST - Test Reset → Rücksetzen des TAP-Controller (optional)



## TAP-Controller: Zustandsautomat

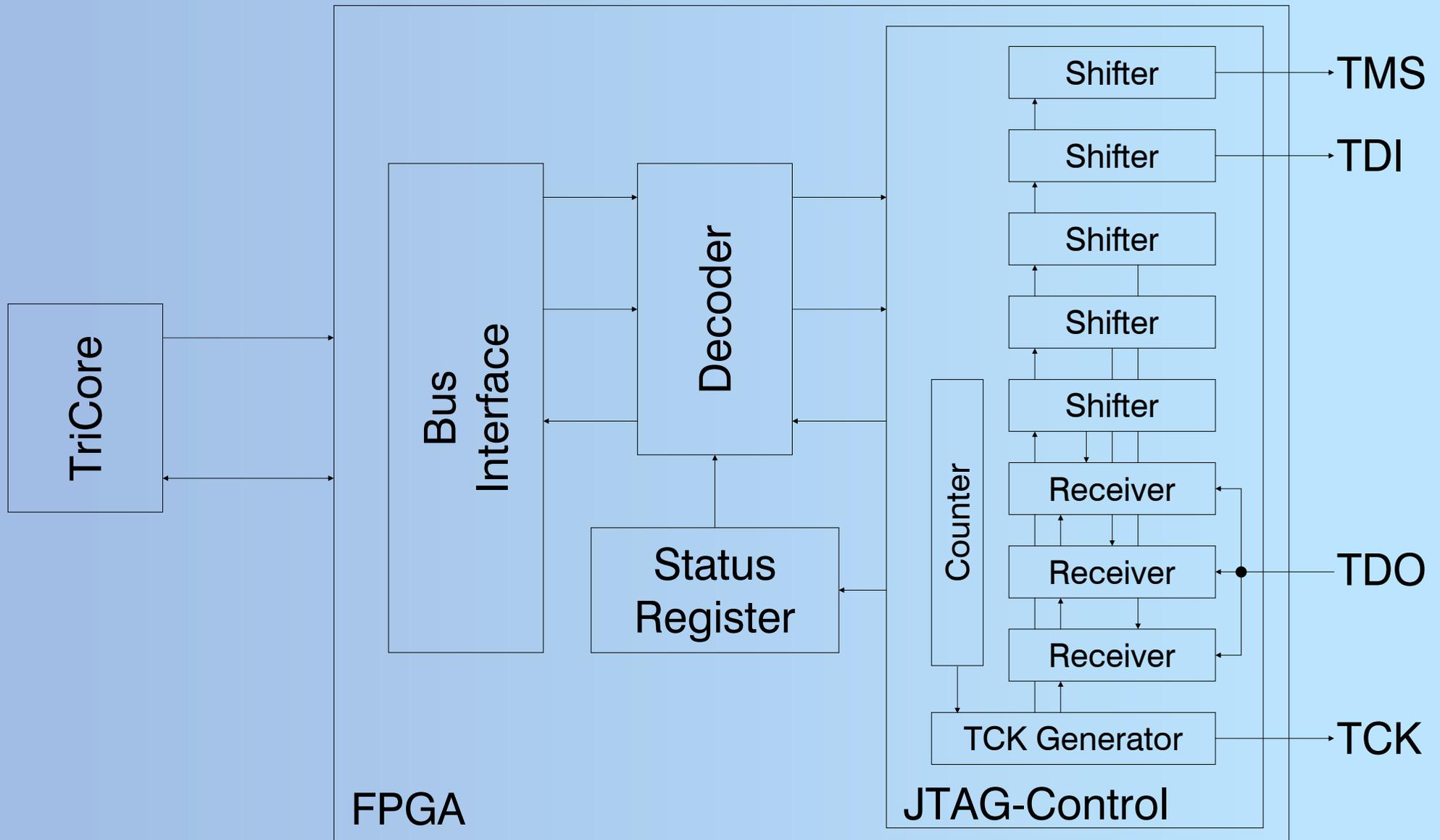


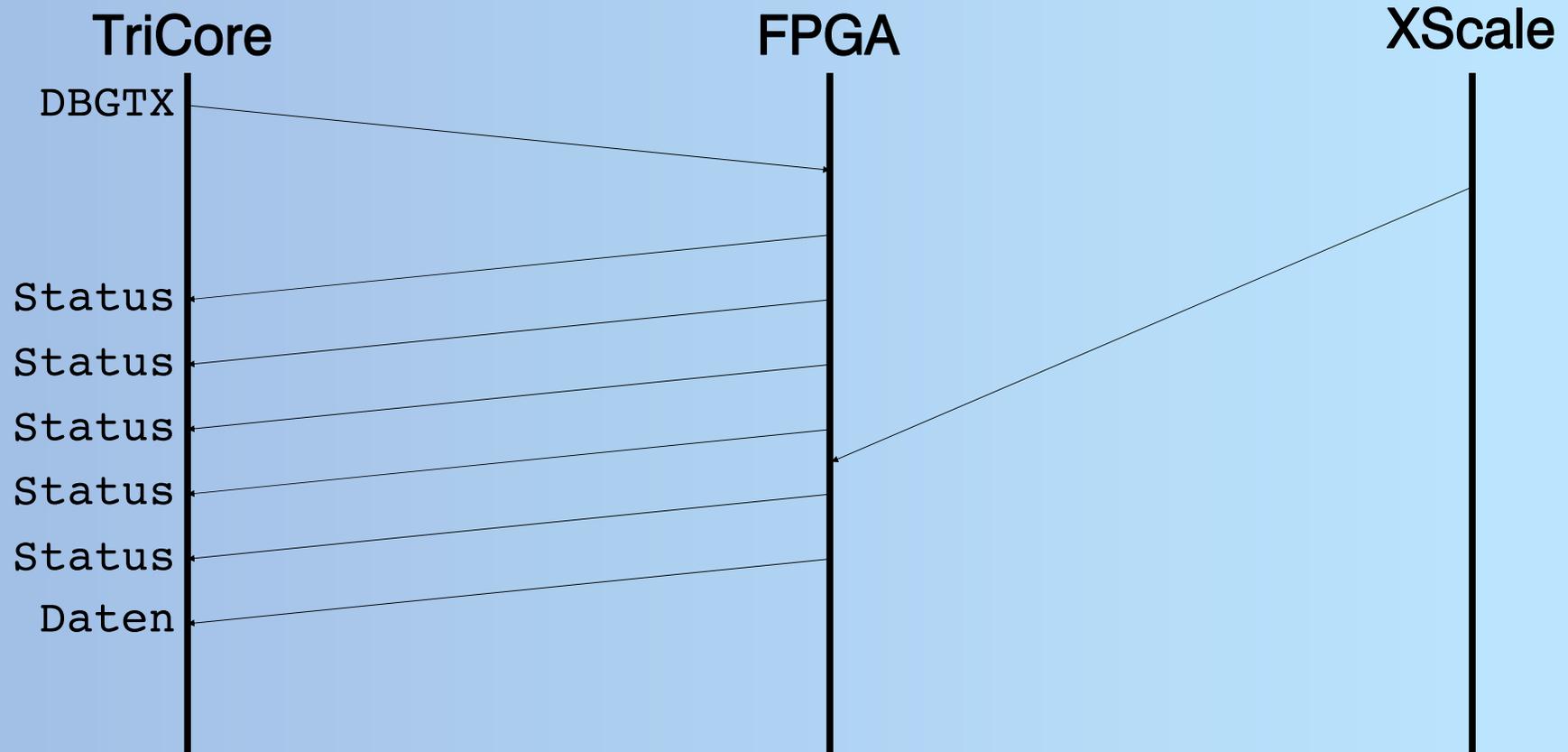
- 16 Zustände
- Zustandsübergang erfolgt an steigender TCK-Flanke
- 5 steigende TCK-Flanken mit TMS „high“, führen immer in „Test-Logic Reset“
- nTRST wird nicht benötigt



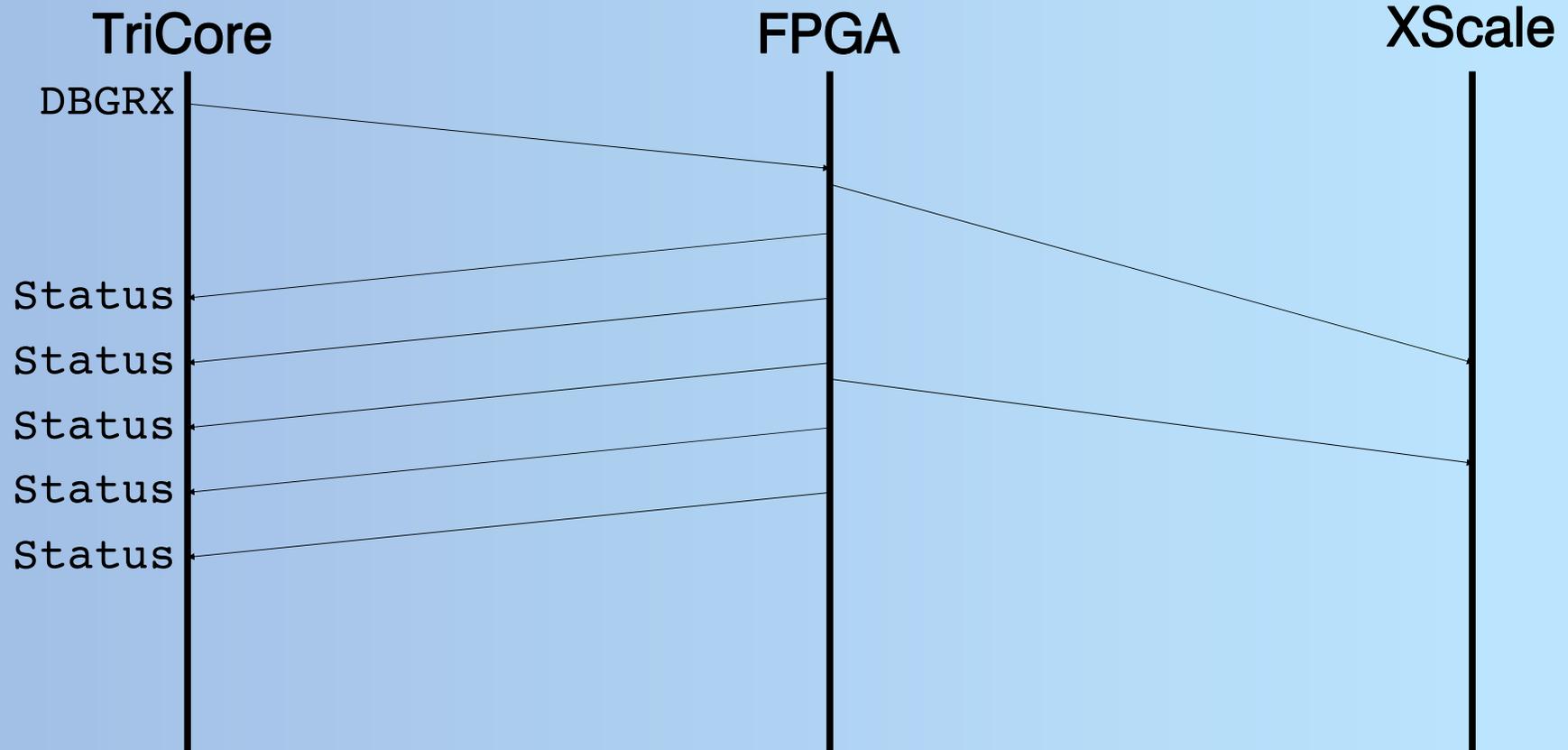
- UAD2 ist selbst eingebettetes System
- Debugging der Firmware mithilfe JTAG möglich
- weiteres UAD2 erforderlich

## FPGA-Design (vereinfacht)





➤ Timeout Behandlung durch Zähler in Firmware



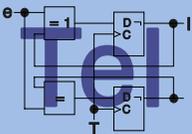
➤ Timeout Behandlung durch Zähler in Firmware

## ➤ Problem

- Länge des zweiten Teils von DBGRX ist abhängig von JTAG-Frequenz (für optimale Übertragungsbedingungen)

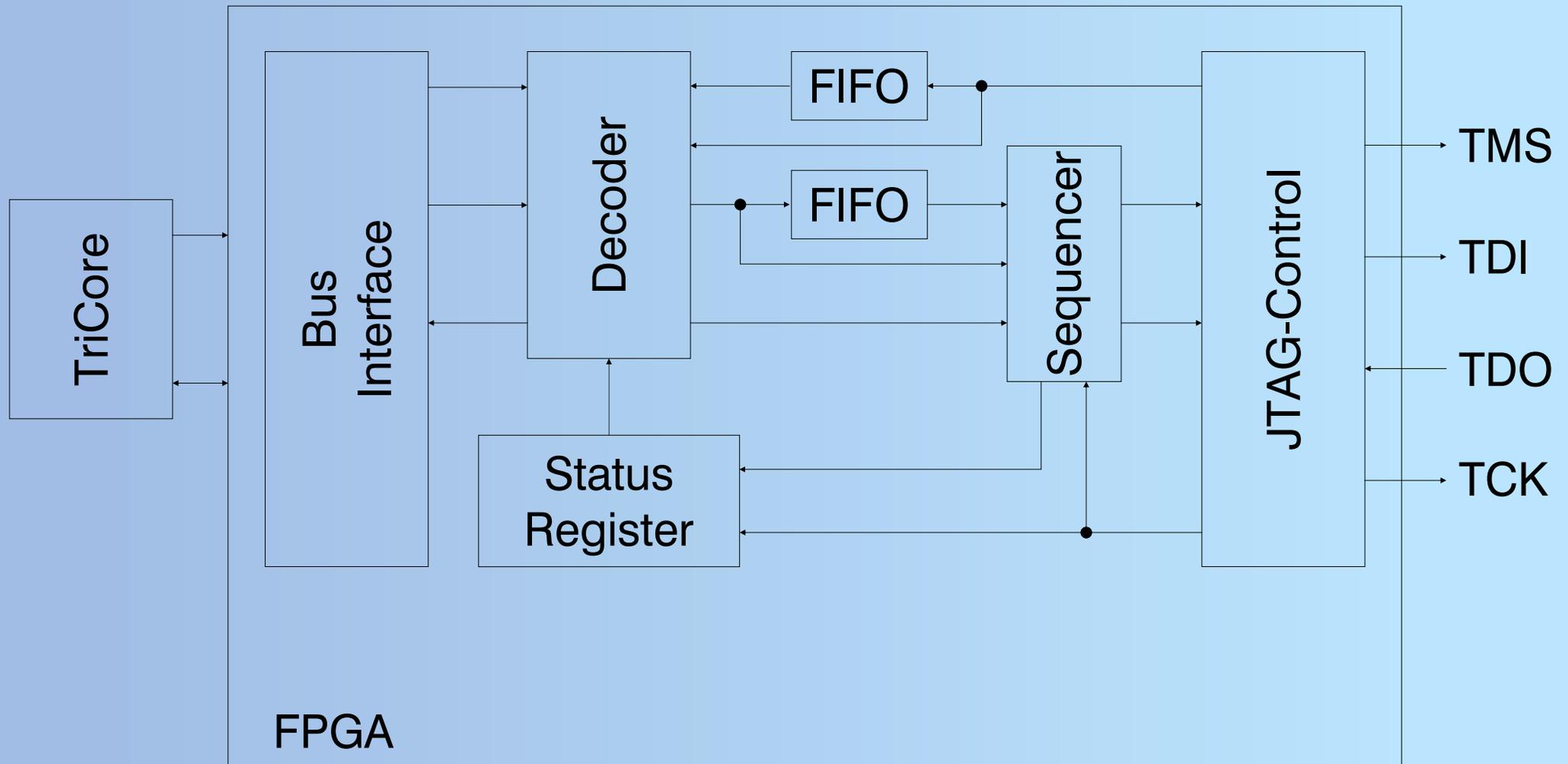
## ➤ Lösung

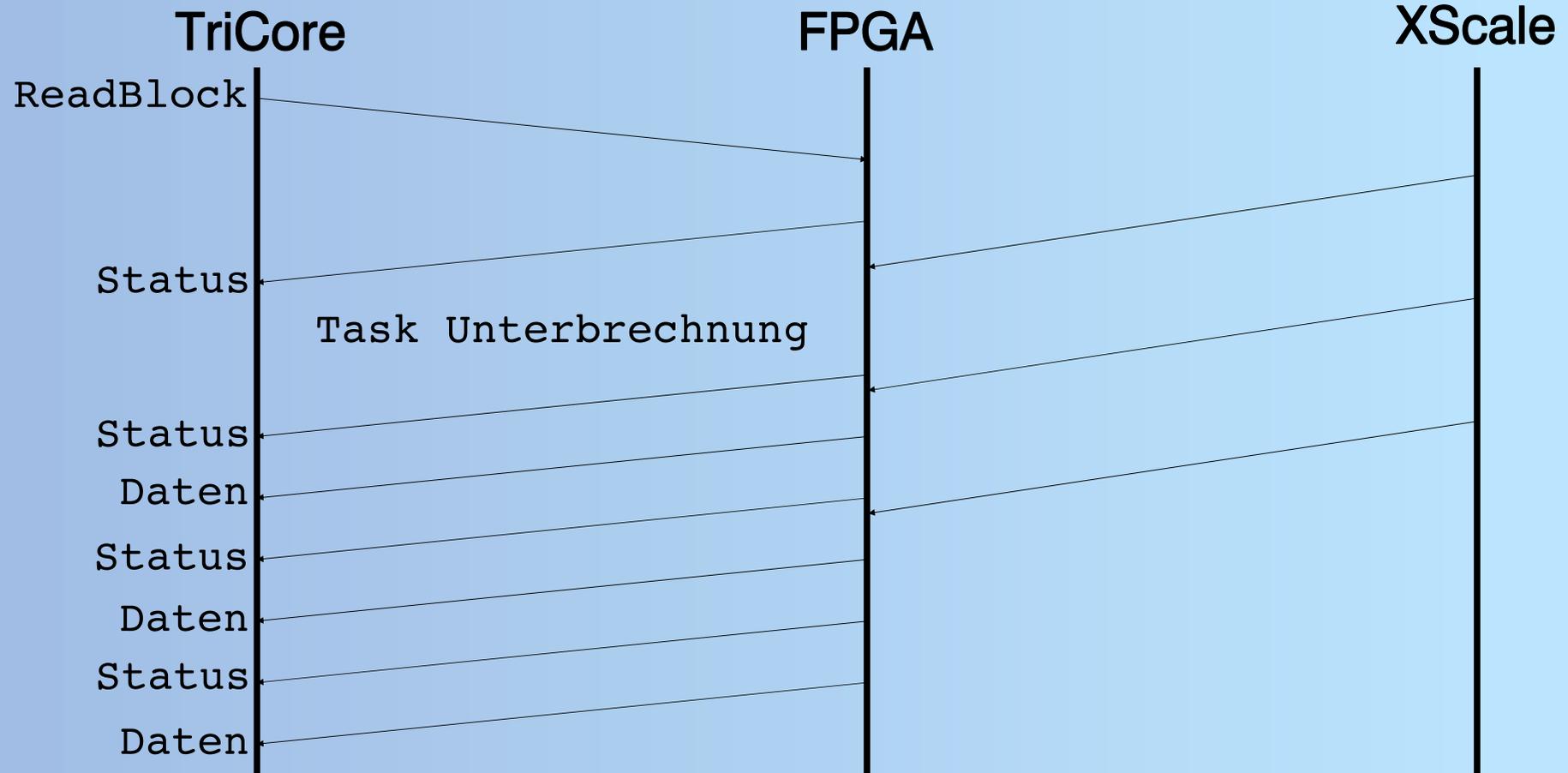
- Verwendung von zwei Zählern zur Unterbrechung des Shifts von TMS
- erster Zähler bestimmt die Anzahl der einleitenden Shifts (Wechsel von „SHIFT-DR“ in „RUN-TEST IDLE“)
- zweiter Zähler legt die Anzahl der TCK-Zyklen fest, für die „RUN-TEST IDLE“ gehalten werden soll
- letzterer kann durch Firmware beim „Connect“ konfiguriert werden

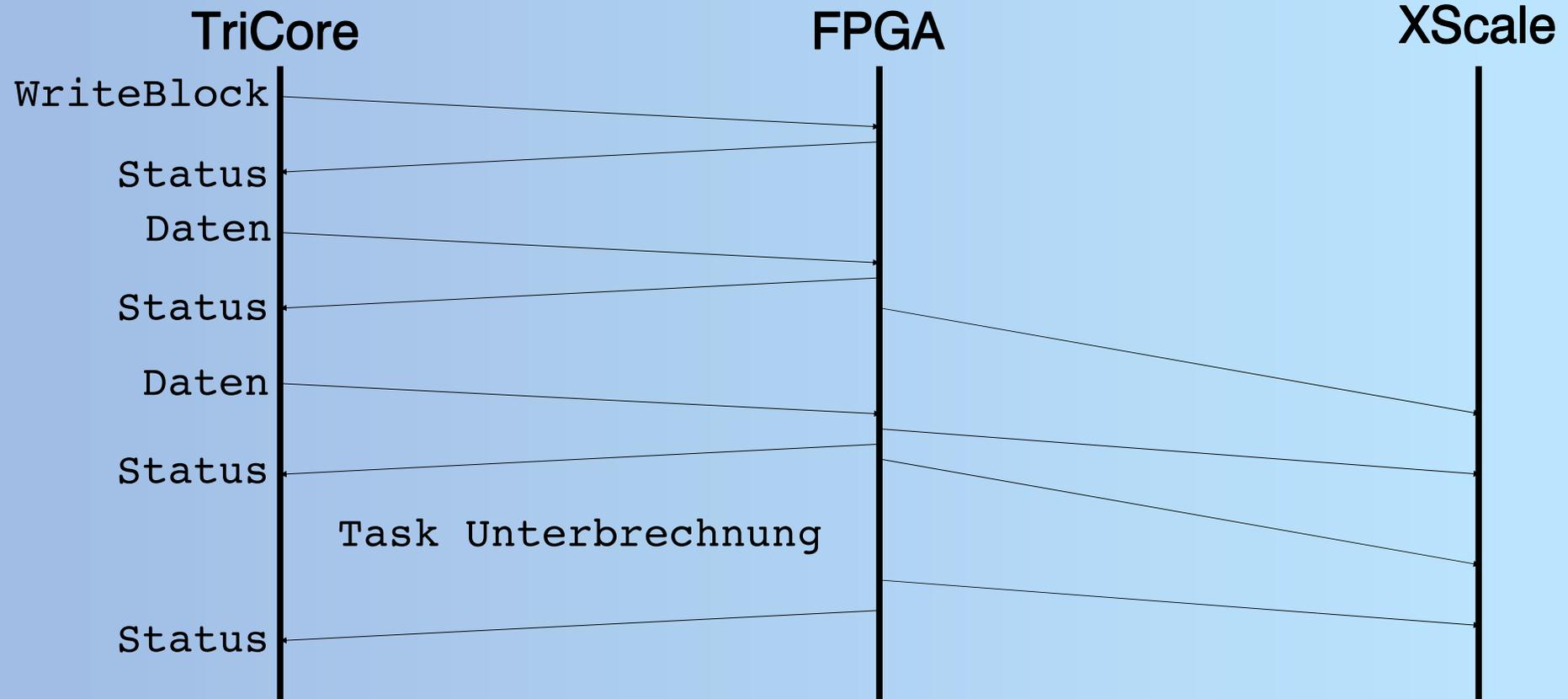


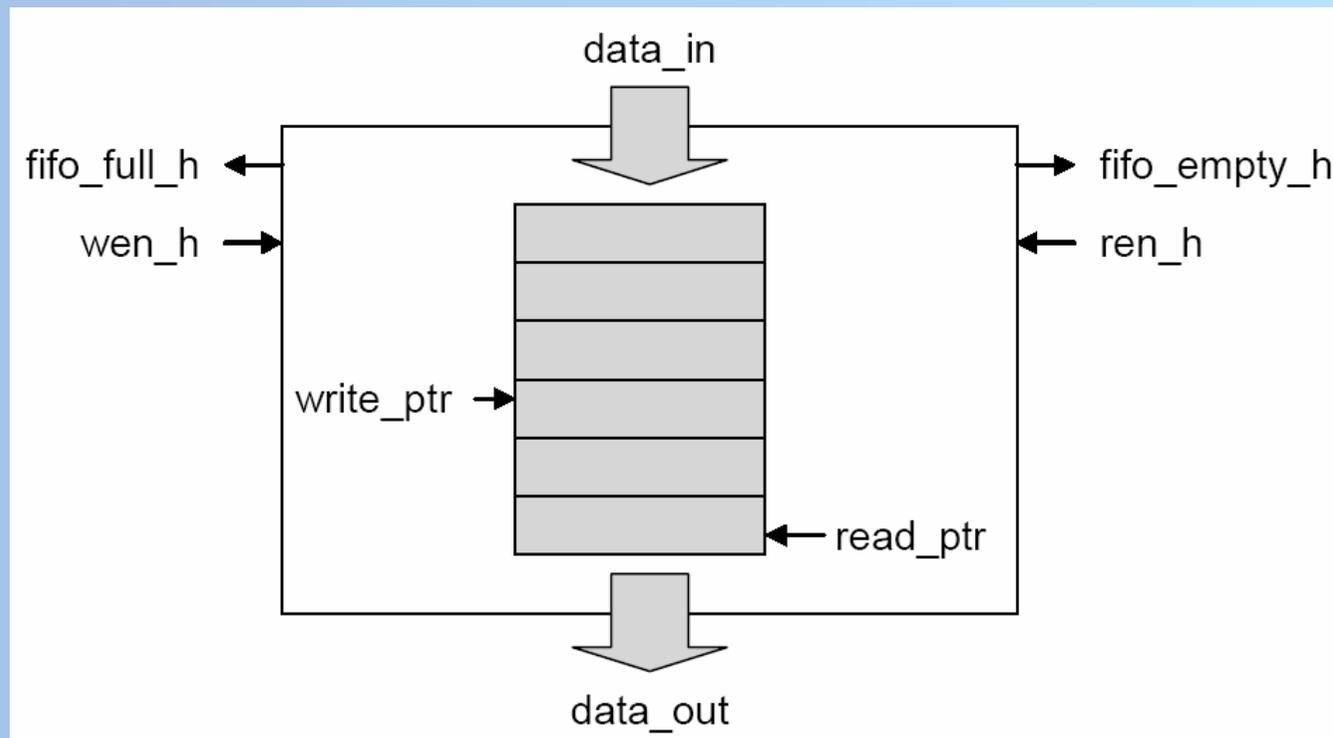
# Kontinuierlicher Datentransfer

## Erweiterung FPGA-Design (vereinfacht)



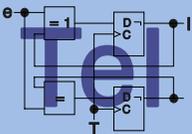




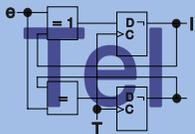
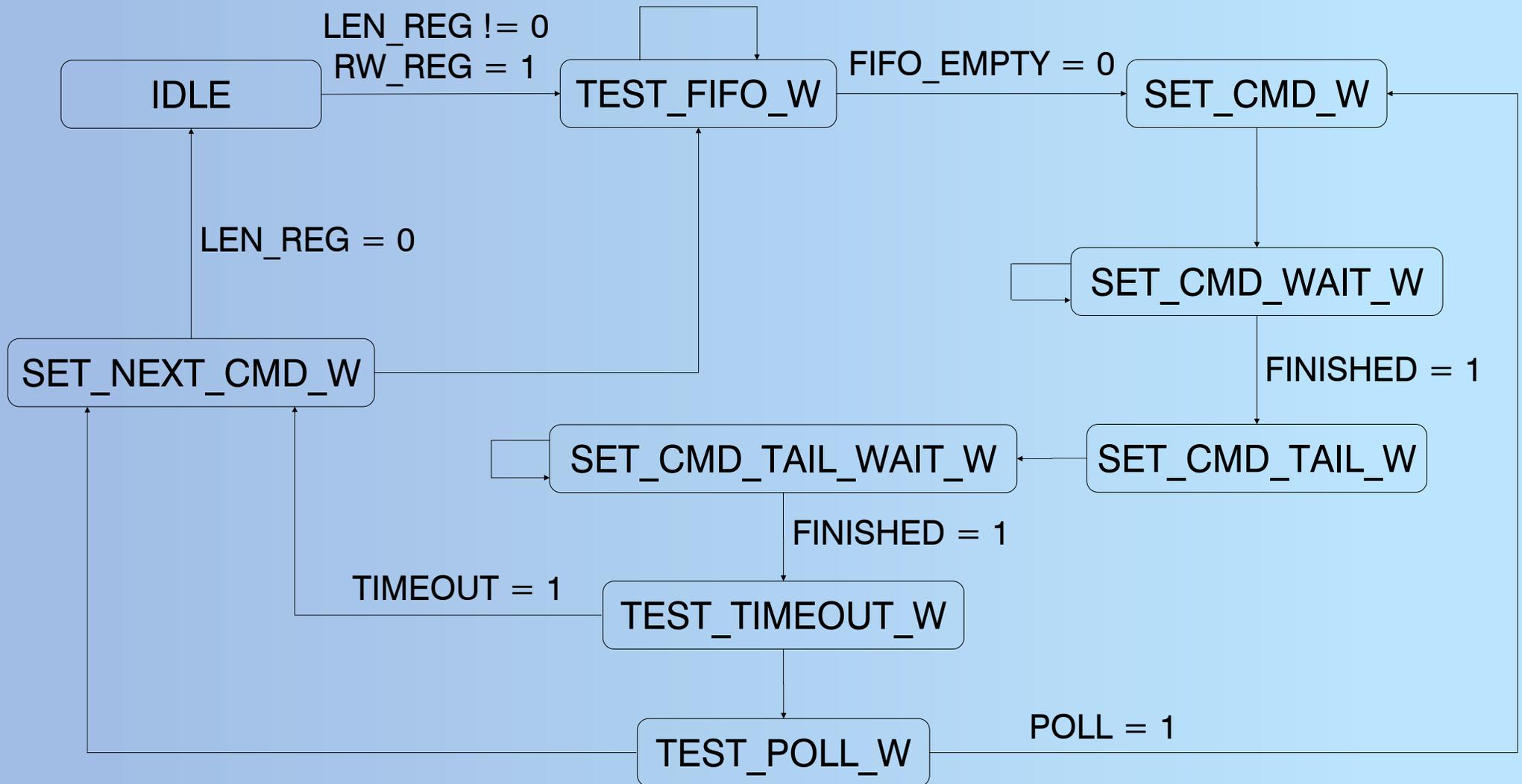


- zur bisherigen Implementierung wurde Alternative zum Vergleich erstellt
- praktisch kein Unterschied in Timing und Hardwareaufwand

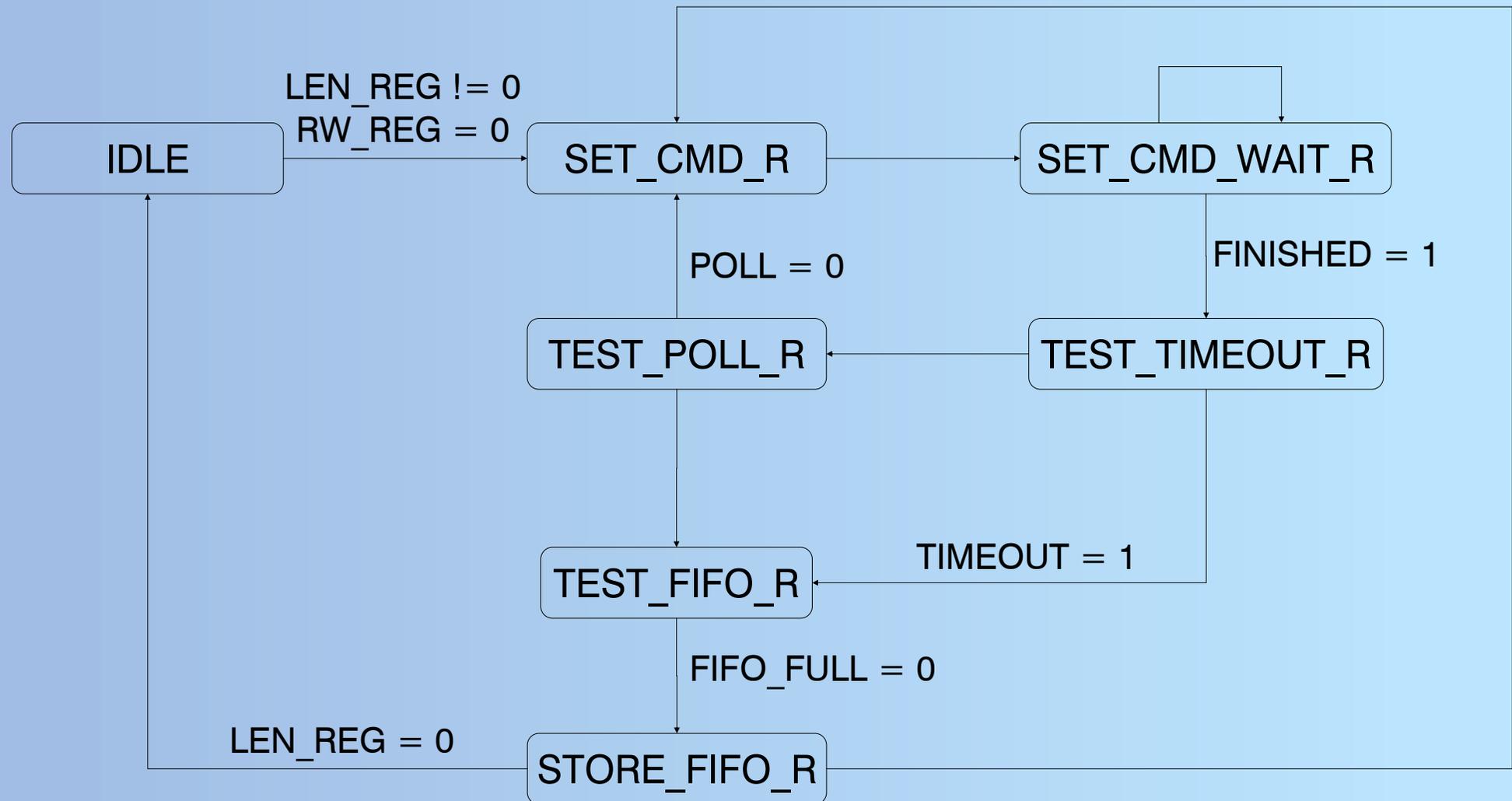
- Zustandsautomat mit 15 Zuständen
- 24-Bit Zähler für Timeout-Steuerung
- 14-Bit Register für Anzahl der Datenpakete
- Bypass aller Kommandos und Eingangsdaten, wenn kein Blocktransfer ausgeführt wird
- Little- und Big-Endian Unterstützung am Ein- bzw. Ausgang der FIFO's
- prinzipielles Übertragen bzw. Empfangen aller Datenpakete, auch wenn Timeout bereits eingetreten ist → garantiertes Leeren der FIFO's ohne zusätzliche Steuerung
- Ressourcen-Verbrauch: 88 Slices (11%)



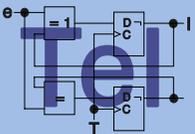
# Sequencer: Zustandsautomat (Teil 1)



## Sequencer: Zustandsautomat (Teil 2)



- Bereitstellung neuer Low-Level-Funktionen für Blocktransfer (Little- und Big-Endian)
- Verbesserung der Blocktransfer in Zwischenschicht durch Nutzung der zuvor definierten Low-Level-Funktionen
- Änderung des Einzelworttransfers (DBGRX, DBGTX) mit FIFO-Nutzung (ohne Task-Unterbrechung), aber Verschlechterung der Übertragungszeiten



➤ ohne FIFO:

Blockgröße	25 MHz	50 MHz
1k	3.1/2.9	2.3/2.1
2k	6.2/5.7	4.7/4.2

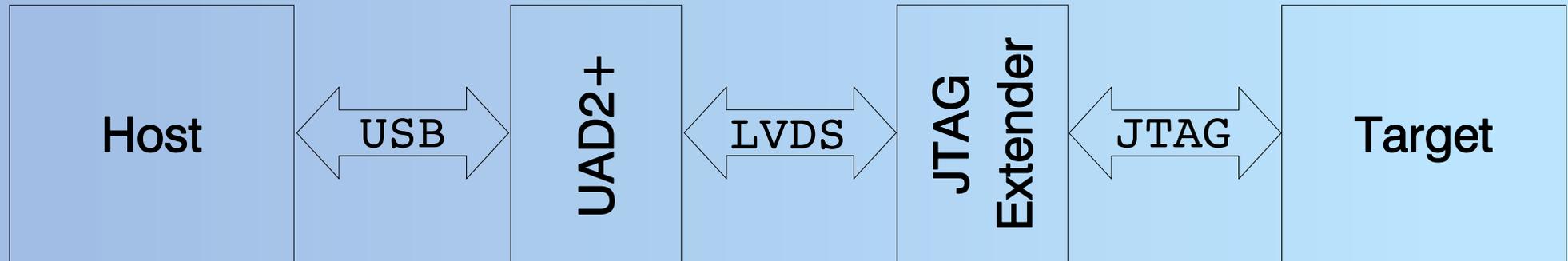
➤ mit FIFO:

Blockgröße	25 MHz	50 MHz
1k	2.2/1.8	1.3/1.2
2k	4.3/3.5	2.7/2.3

- alle Zeitangaben in ms (schreiben/lesen)
- 1k Block entsprechen 4096 Bytes

## Erweiterung zu UAD2+

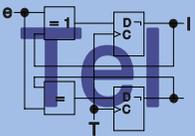
UDE-Debugger



## ➤ LVDS (low voltage differential signal):

- relativ immun gegenüber Störungen
- längere Verbindungskabel zwischen UAD2+ und Target möglich

- variable Verzögerung durch JTAG-Extender, d.h. TCK und TDO am FPGA-Eingang nicht mehr synchron
- weitere Verzögerung zwischen JTAG-Extender und Target (deutlich geringer als vorherige)
- Einengung des Abtastbereiches durch hohe JTAG-Frequenz in Bezug auf FPGA-Taktfrequenz
- bisher zwei Lösungsansätze möglich



## ➤ 1. Ansatz

- konfigurierbare Abtastverzögerung im FPGA
- ungünstige zeitliche Auflösung erfordert Abtastung auf steigender und fallender Taktflanke
- Gesamtverzögerung muss vorher bekannt sein

## ➤ 2. Ansatz

- Nutzung des RTCK durch Rückführung von TCK im JTAG-Extender
- RTCK und TDO sind quasi synchron → Übernahme auf fallender Flanke von RTCK
- Verzögerung zwischen JTAG-Extender und Target bleibt unberücksichtigt
- für UAD2 ist Rückführung des (kombinatorischen) TCK auf FPGA erforderlich → schlechter Designstil durch hohen Net-Skew

