



## Zwischenvortrag zur Studienarbeit

# Entwurf und Implementierung eines statischen Backbones für die Kommunikation mit dynamischen Nutzerpartitionen auf einem Multi-FPGA-Board

Albert Schulz

Dresden, 12.05.2016



## Gliederung

1. Motivation
2. Zielarchitektur
3. Stand der Technik
4. Systemstruktur
5. Protokollentwurf
6. Aktueller Stand
7. Ausblick

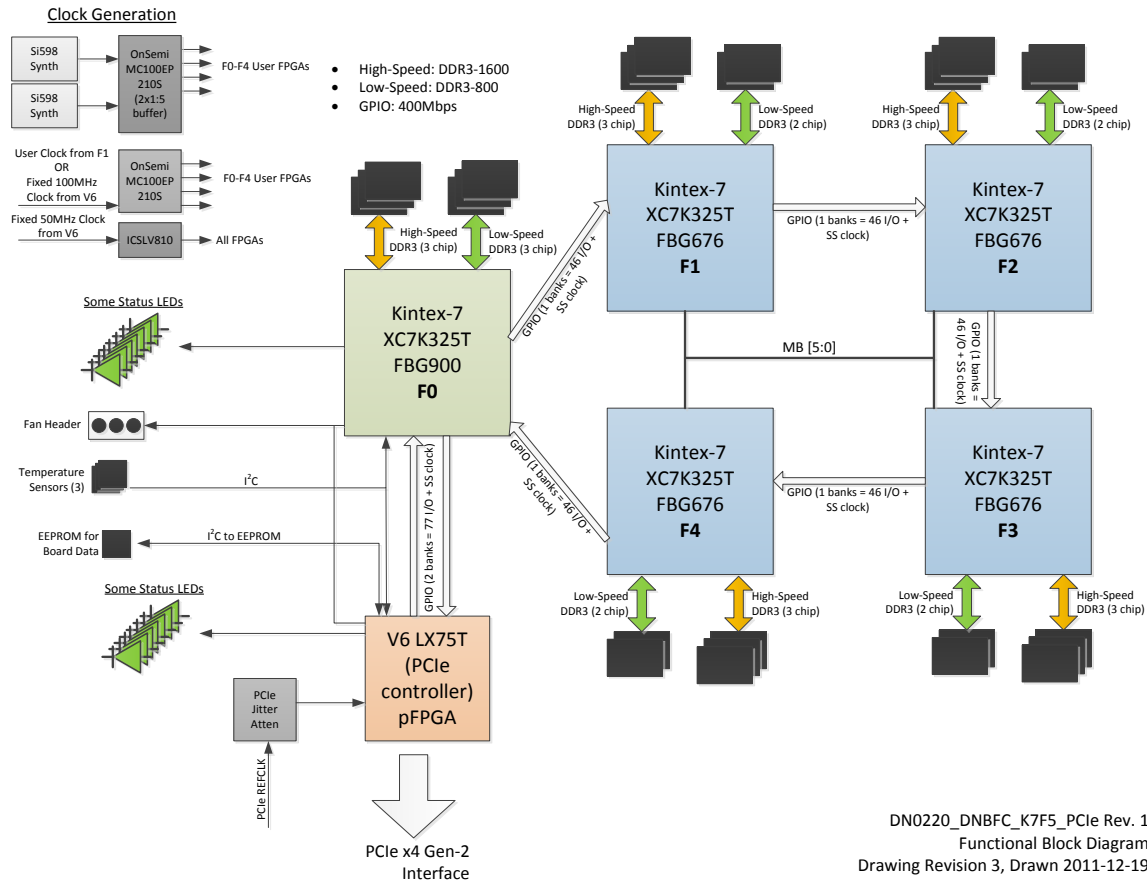
## 1. Motivation

- Multi-FPGA Board mit großen FPGAs
- Optimale Auslastung → mehrere User Designs (Nutzerpartitionen) pro FPGA
- Kommunikation zwischen Host und User Designs über einen gemeinsamen Bus
  - Multiplexen mehrerer Datenströme erforderlich

## 2. Zielarchitektur

- DNK7 F5 PCIe-Board mit 5 Kintex-7 FPGAs
  - 4x gFPGA: „User FPGAs“
  - 1x dFPGA: „Dataflow Manager“
- Verbindung der FPGAs über Ringbus mit 46 GPIO Pins
- PCIe-Interface zu Host-Rechner (mittels pFPGA)

## 2. Zielarchitektur

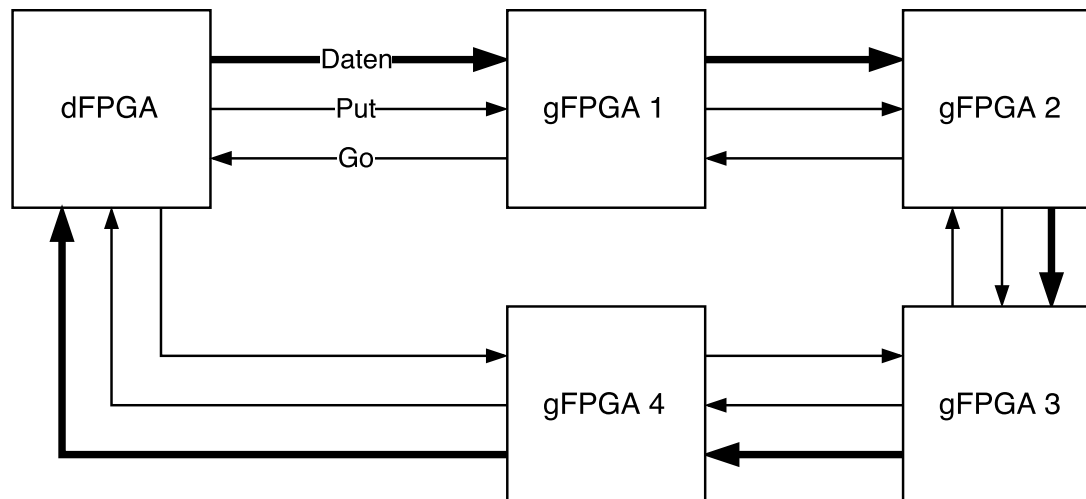


Quelle:

[http://www.dinigroup.com/product/data/DNK7\\_F5PCIe/files/Hardware\\_Manual\\_DNK7\\_F5\\_PClE\\_REV4.pdf](http://www.dinigroup.com/product/data/DNK7_F5PCIe/files/Hardware_Manual_DNK7_F5_PClE_REV4.pdf)

### 3. Stand der Technik

- systolische Datenübertragung über Ringbus
- Eingangs- und Ausgangspuffer an Sender/Empfänger
- Flusskontrolle zwischen benachbarten Knoten über Put/Go-Signale

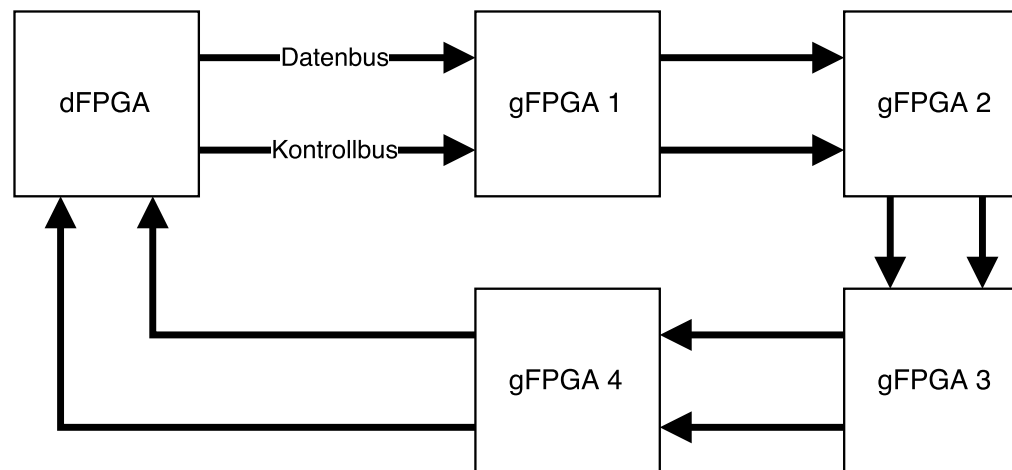


### 3. Stand der Technik

- Source-Synchronous-Schaltung
- differenzieller Takt (durch gFPGA 1 erzeugt)

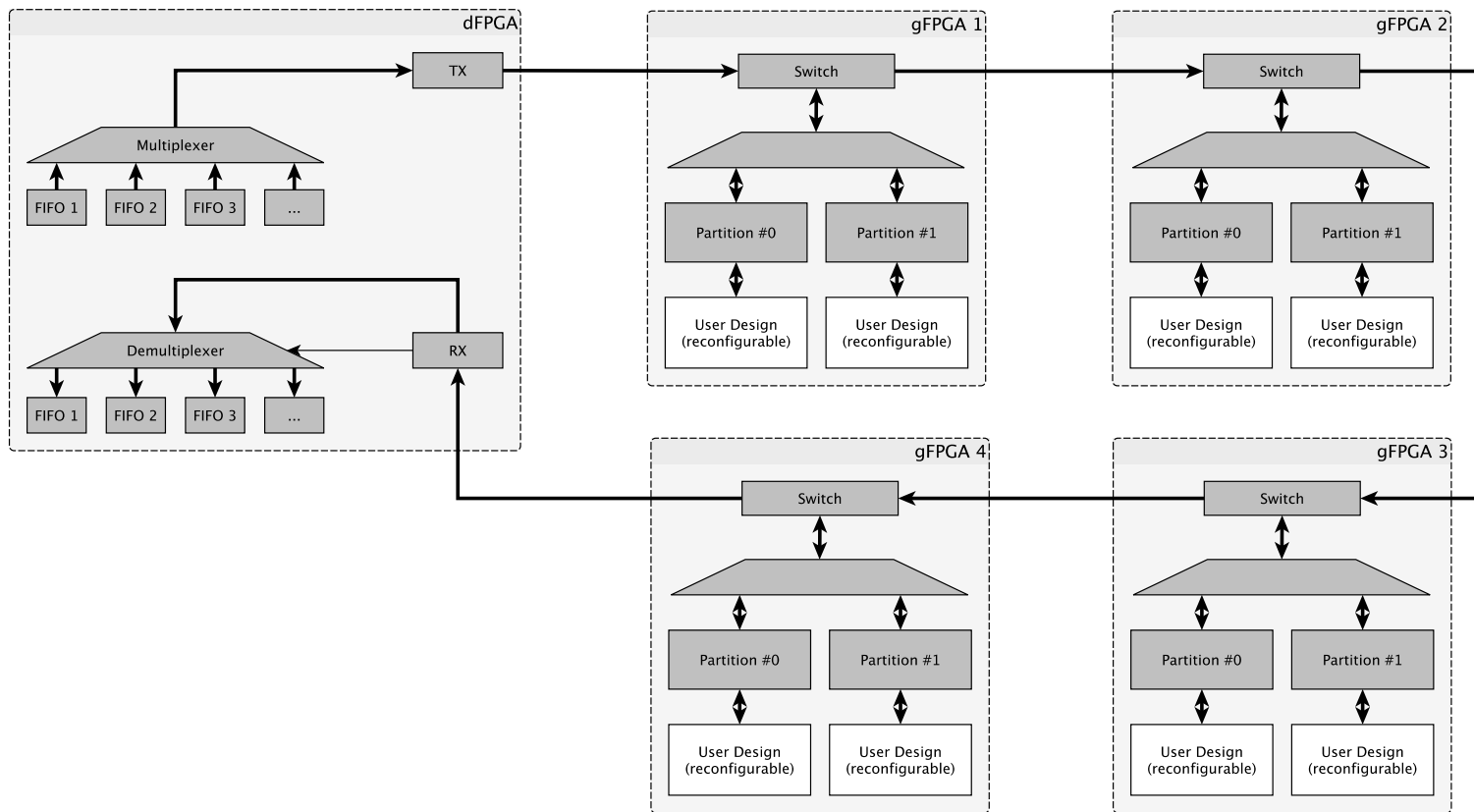
## 4. Systemstruktur - Überblick

- 2 getrennte Busse:
  - 32-Bit Datenbus
  - 8-Bit Kontrollbus für latenzarme Übermittlung von Status- und Kontrollinformationen

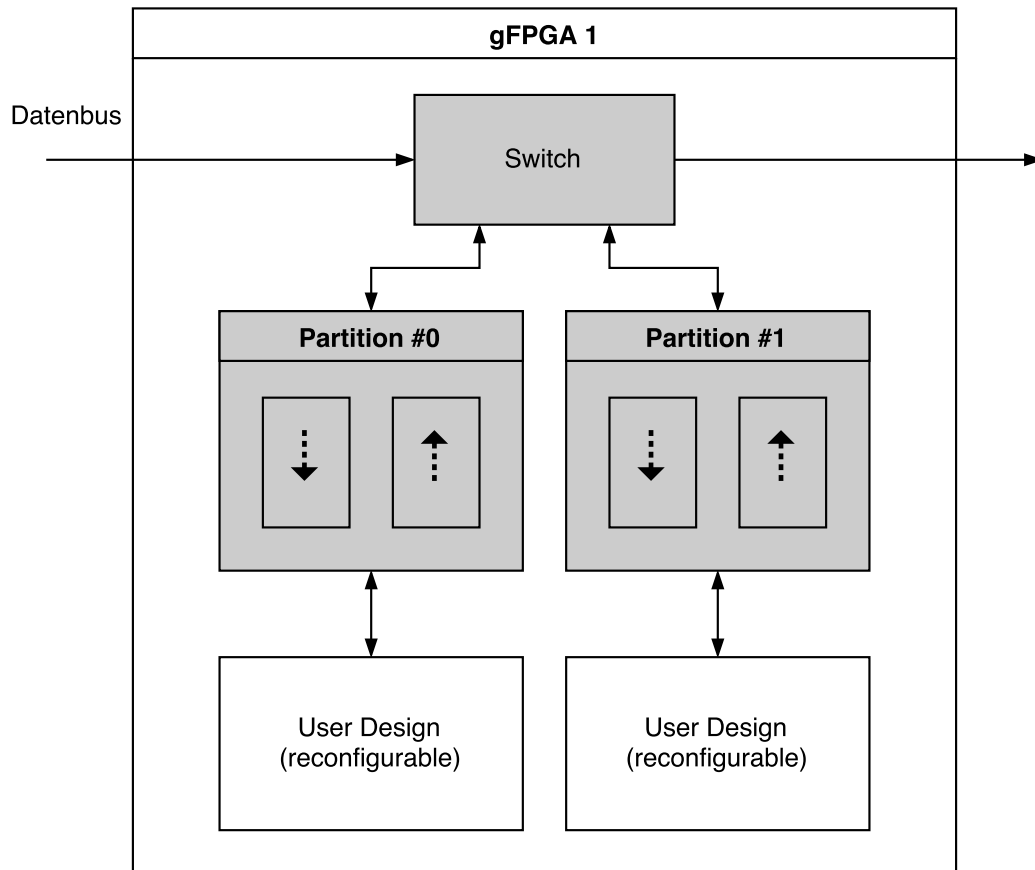




# 4. Systemstruktur - Datenfluss



## 4. Systemstruktur - Datenfluss



## 5. Protokollentwurf

- paketorientiertes Protokoll
- Nutzung der 2 Busse:
  - Datenpakete über breiten Datenbus
  - Pakete für Flusskontrolle, Interrupts und Statusinformationen über Kontrollbus
- Paketaustausch nur zwischen dFPGA <> gFPGAs (User Design)

## 5. Protokollentwurf - Datenpaketformat

Frame-Format für Datenpakete:

32 Bit	8 Bit	8 Bit	12 Bit	4 Bit	0-16 KiByte
Sentinel	Ziel- adresse	Quell- adresse	Datenlänge	Pakettyp	Daten

- Sentinel zur Erkennung eines gültigen Pakets
  - notwendig, da ungültiger Zustand nach Bus-Start beobachtet

## 5. Protokollentwurf - Datenpaketformat

Frame-Format für Datenpakete:

32 Bit	8 Bit	8 Bit	12 Bit	4 Bit	0-16 KiByte
Sentinel	Ziel- adresse	Quell- adresse	Datenlänge	Pakettyp	Daten

- Adressformat = 3 Bit ChipID + 5 Bit User Design ID
  - 3 Bit ChipID notwendig für 5 FPGAs (dFPGA + 4 gFPGAs)
  - max. 32 User Designs pro gFPGA

## 5. Protokollentwurf - Datenpaketformat

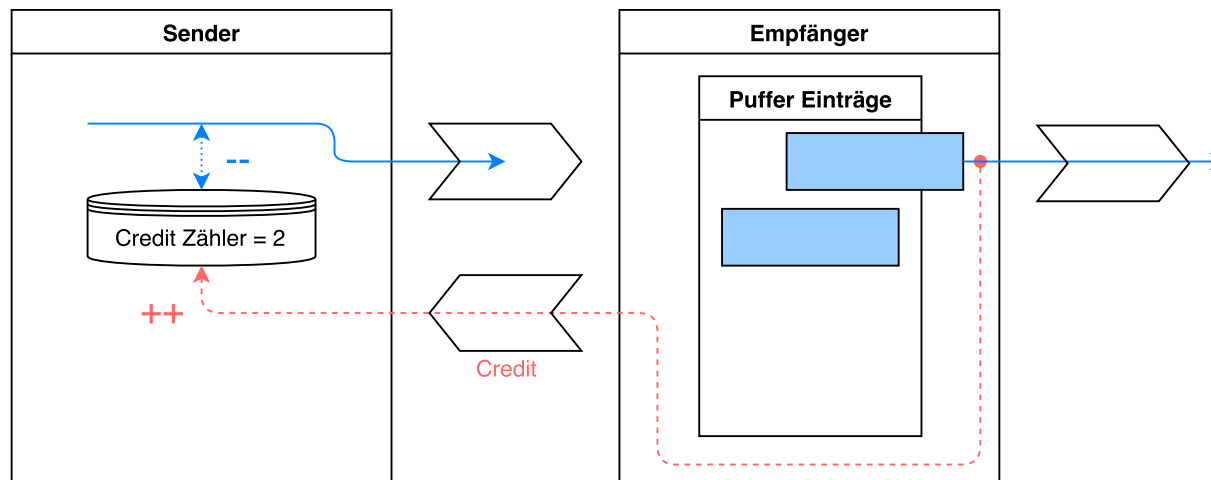
Frame-Format für Datenpakete:

32 Bit	8 Bit	8 Bit	12 Bit	4 Bit	0-16 KiByte
Sentinel	Ziel- adresse	Quell- adresse	Datenlänge	Pakettyp	Daten

- max. Paketgröße:  $2^{12}$  Worte + Header = ~16 KiByte
- 16 Pakettypen definierbar
  - z.B. Daten für Anwendung, Konfigurationsdaten, Speicherinhalt
  - Weiterleitung der Daten durch Switch zu FIFOs, ICAP Interface oder RAM

## 5. Protokollentwurf - Flusskontrolle

- Flusskontrolle zwischen dFPGA und User Designs über „Credit-Based Flow Control“
  - verlustfrei, kein wiederholtes Senden von Paketen aufgrund von Pufferüberlauf nötig

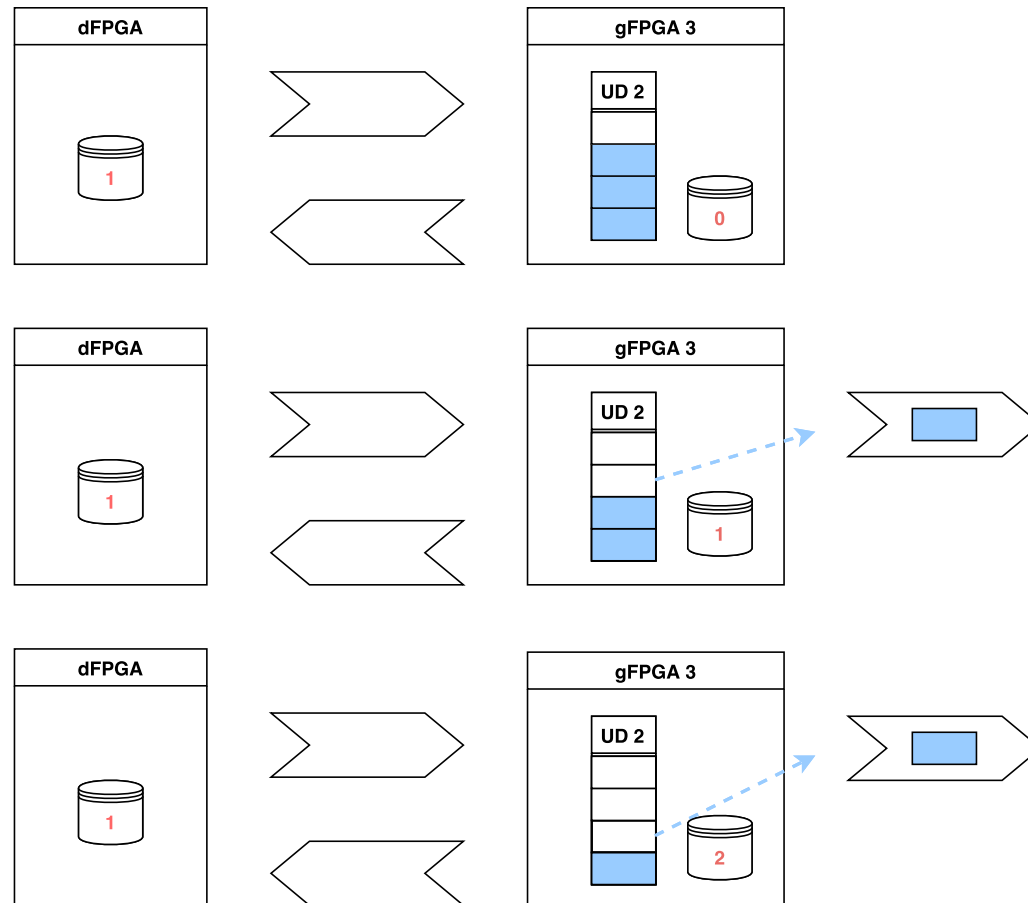


## 5. Protokollentwurf - Flusskontrolle

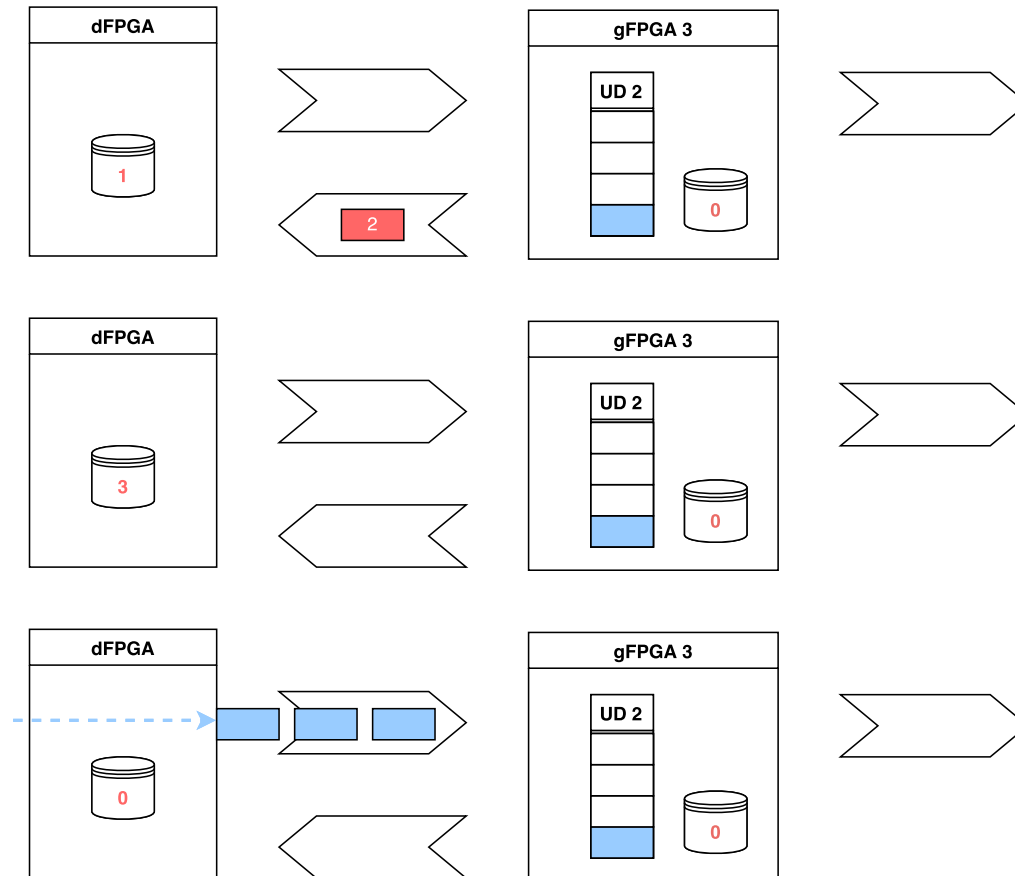
- Empfänger vergibt Credits, wenn  $N$  Einträge aus Puffer gelesen wurden ( $N = \text{Halbe Puffergröße}$ )
  - geringere Belastung des Kontrollbusses



## 5. Protokollentwurf - Flusskontrolle: Szenario



## 5. Protokollentwurf - Flusskontrolle: Szenario



## 5. Protokollentwurf - Kontrollpakete

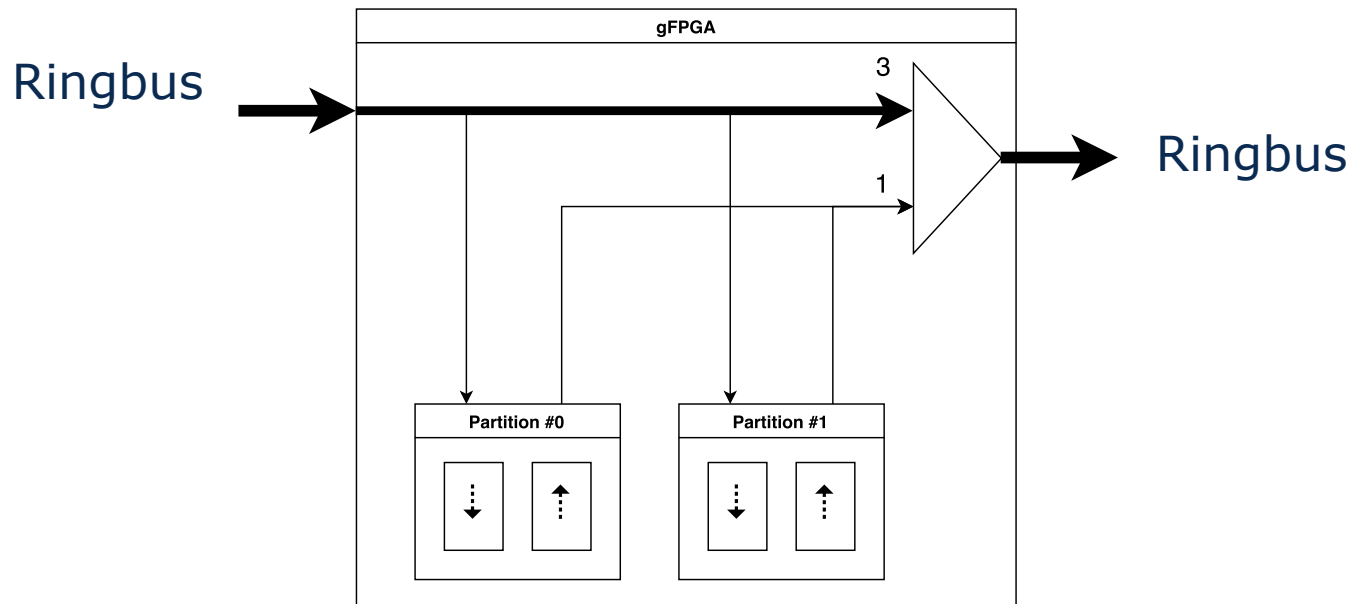
Frame-Format für Kontrollpakete:

1 Byte	1 Byte	1 Byte	1 Byte	1-n Byte
Sentinel	Zieladresse	Quelladresse	Pakettyp	Daten

- 256 mögliche Pakettypen
  - für z.B. Credits für Flusskontrolle, Statusinformationen, Interrupts
- Datenlänge abhängig vom Pakettyp, z.B.:
  - 1 Byte bei Credit-Paketen (Anzahl an Credits)

## 5. Protokollentwurf - Arbitrierung

- faire Buszuteilung mittels gewichtetem Round-Robin-Verfahren (3:1)



## 6. Aktueller Stand

- Protokoll im Wesentlichen spezifiziert
- VHDL-Testbench
- Implementierung des Datenpfades für 32-Bit Bus (TX, RX, Switch, FIFOs, MUX)
  - funktionierender Datentransfer zwischen dFPGA-Design und User Designs

## 7. Ausblick

- Flusskontrolle implementieren
- Implementierung der Arbitrierung
- Kommunikation zwischen Host-Rechner und dFPGA
- Fehlererkennung/-korrektur (falls notwendig)
- funktionaler Test des Designs (simulativ & physisch)
- Performance Test
- Dokumentation (Entwurf, Realisierung, Testergebnisse)

## Quellen

- „Infrastructure Proposal for Sharing the DINI FPGA Cluster“ - Dr.-Ing. Thomas B. Preußner
- „DNK7\_F5\_PCIe Hardware Manual“ ([Link](#)) - N. Harder (The DINI Group)
- „Traffic Management for High-Speed Networks: Fourth Lecture International Science Lecture Series“- H.T. Kung
- <https://de.wikipedia.org/wiki/Weighted-Fair-Queuing>, Abruf 9.5.2016