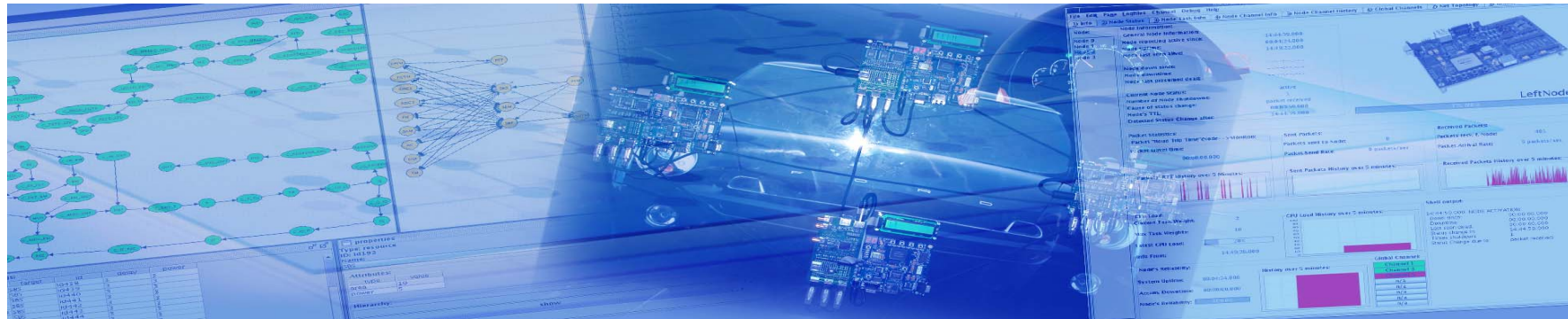


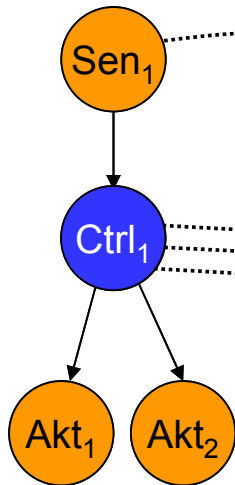
# ReCoNets – Rekonfigurierbare Netzwerke



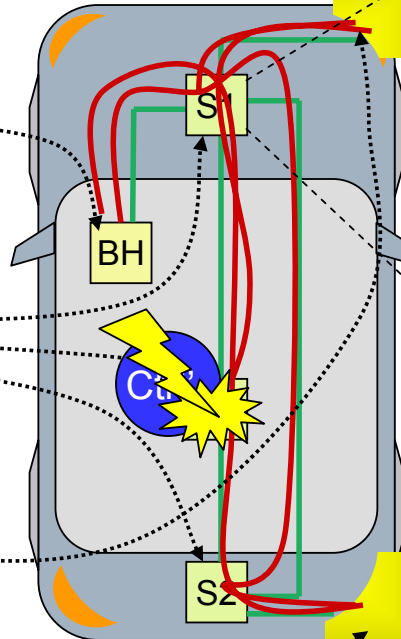
D. Koch, F. Reimann, T. Streichert, J. Teich  
 Lehrstuhl für Informatik 12  
 (Hardware-Software-Co-Design)  
 Universität Erlangen-Nürnberg

# Motivation

Verhalten



Struktur



- = Software-Task
- = Hardware-Task

**Taskplatzierung? Reparaturstrategien? Echtzeit?**  
**Systemintegration? Betriebssystemdienste?**

# Gliederung

---

- Motivation
- Replikplatzierung
- Reparaturstrategien
  - Task-Migration in TDMA-Netzwerken
- Systemintegration / Betriebssystemdienste
  - Rekonfigurierbare Kommunikation
  - ReCoBus-Builder
  - Kopplung ReCoNodes – ReCoNets
- Bearbeitungsstand

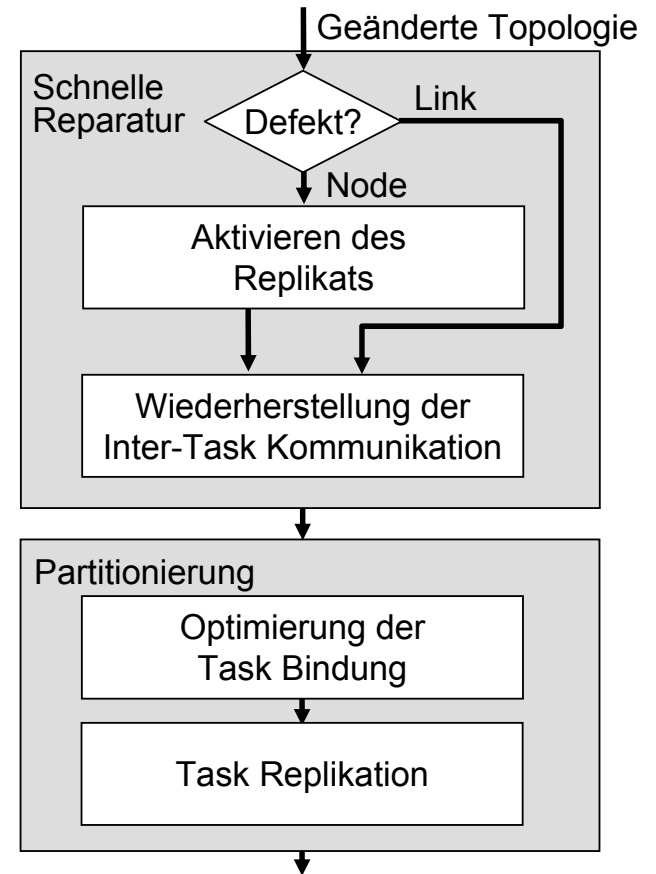
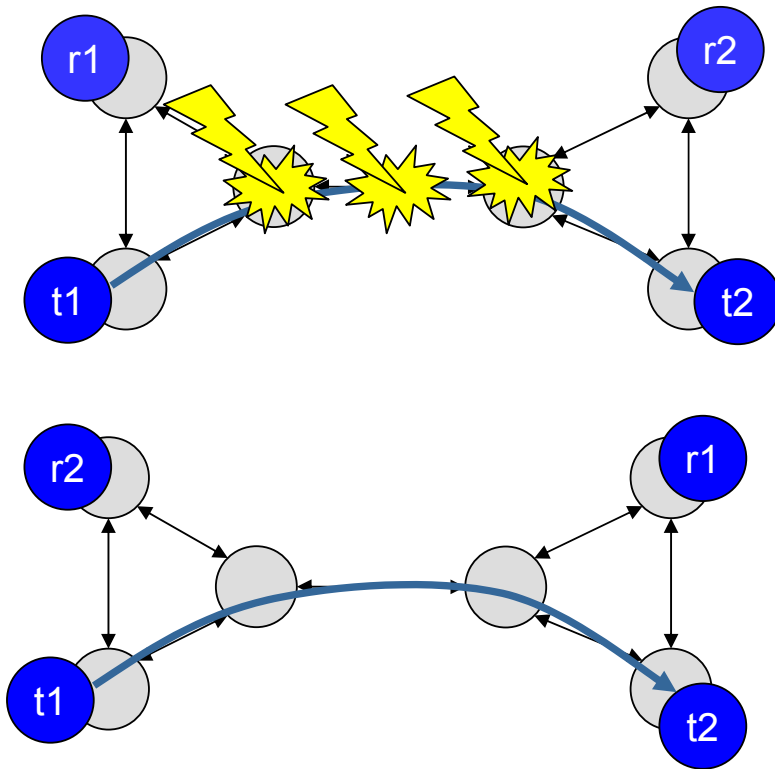
# Gliederung

---

- Motivation
- Replikplatzierung
- Reparaturstrategien
  - Task-Migration in TDMA-Netzwerken
- Systemintegration / Betriebssystemdienste
  - Rekonfigurierbare Kommunikation
  - ReCoBus-Builder
  - Kopplung ReCoNodes – ReCoNets
- Bearbeitungsstand

# Task Replikation

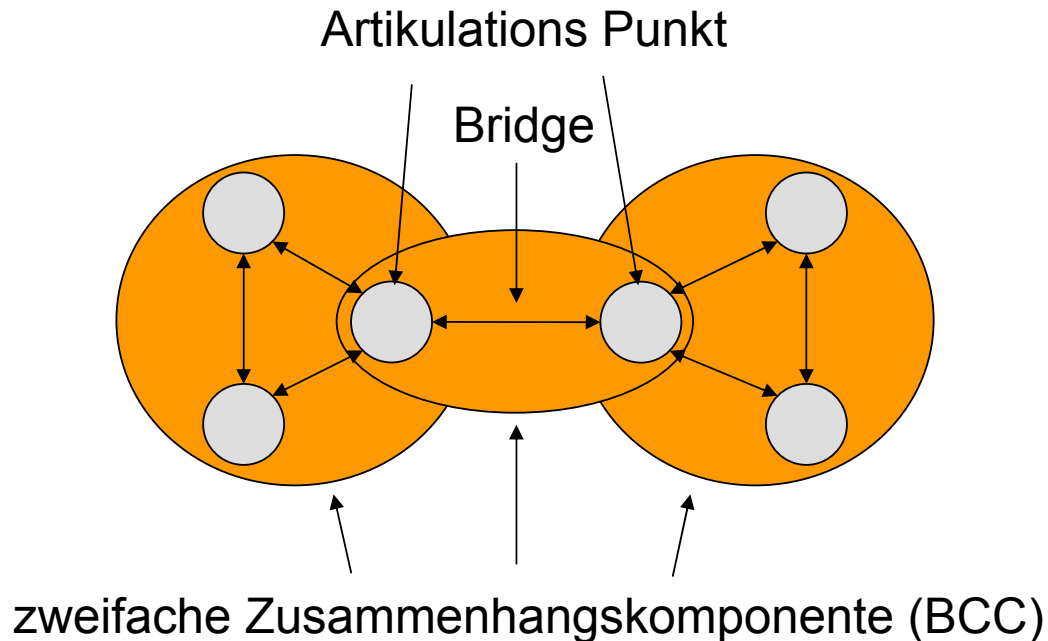
- Wo müssen Replikate platziert werden, damit eine Funktionalität möglichst lange erhalten bleibt?



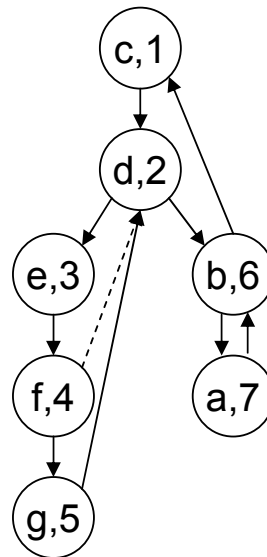
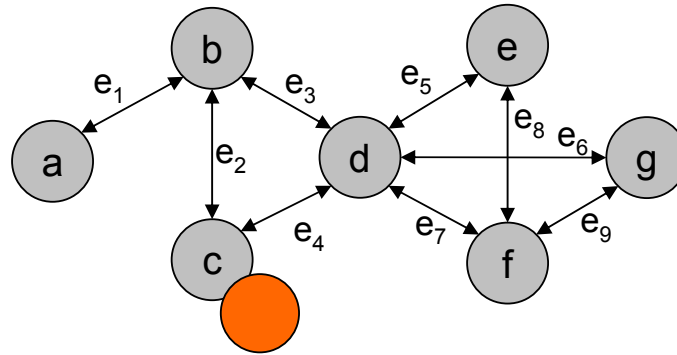
# Task Replikation – Grundidee

---

- Partitionierung des Netzwerks in zweifache Zusammenhangskomponenten (Biconnected Components BCC)



# Bestimmung der BCCs

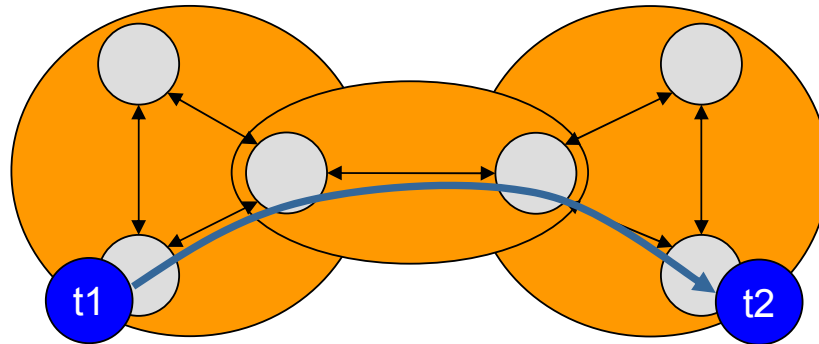


1. BCC {d,e,f,g}
2. BCC {a,b}
3. BCC {c,d,b}

# Task Replikation – Grundidee

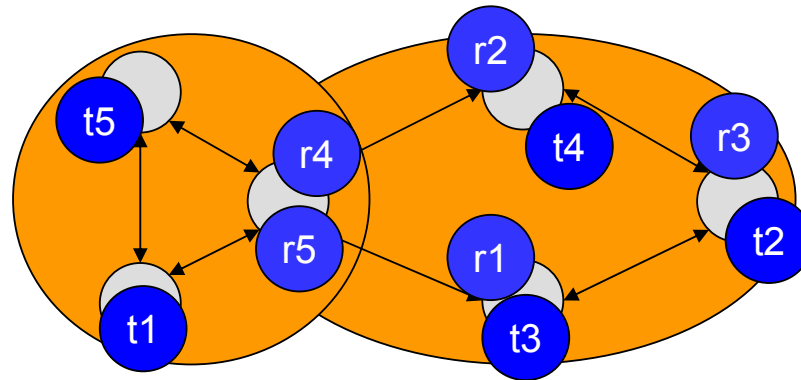
---

- Ausgehend von einer festen Task-Bindung werden zusätzliche Knoten für Replikate gesucht.
- Verschiedene Heuristiken möglich:
  - Replikabindung an Knoten in einer anderen Zusammenhangskomponente mit möglichem Kommunikationspartner
  - Replikabindung an Knoten in einer Zusammenhangskomponente mit der maximalen Anzahl an Tasks
  - ...



# Aktive Replikation

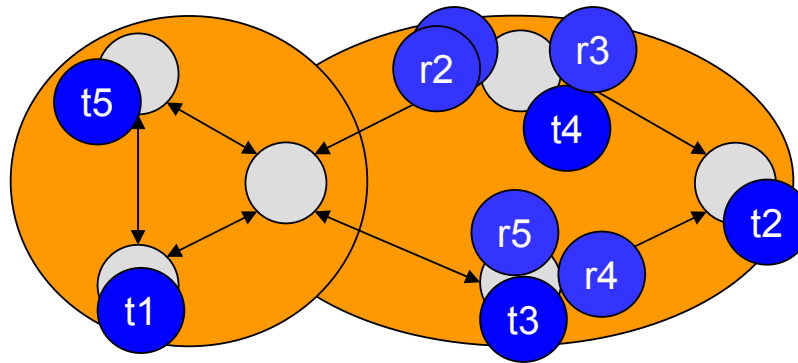
- Annahme:  
Jeder Task belegt 50% der auf einem Knoten verfügbaren Last.
- Tasks kommunizieren untereinander
- Ein aktives Replikat belegt die gleiche Last wie der zugehörige Task.



- Ein einfacher Ressourcendefekt kann zu einem Totalausfall führen!

# Passive Replikation

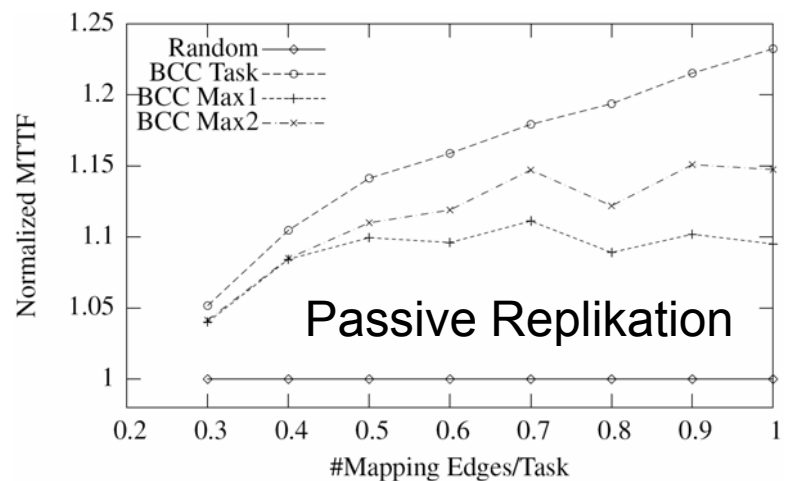
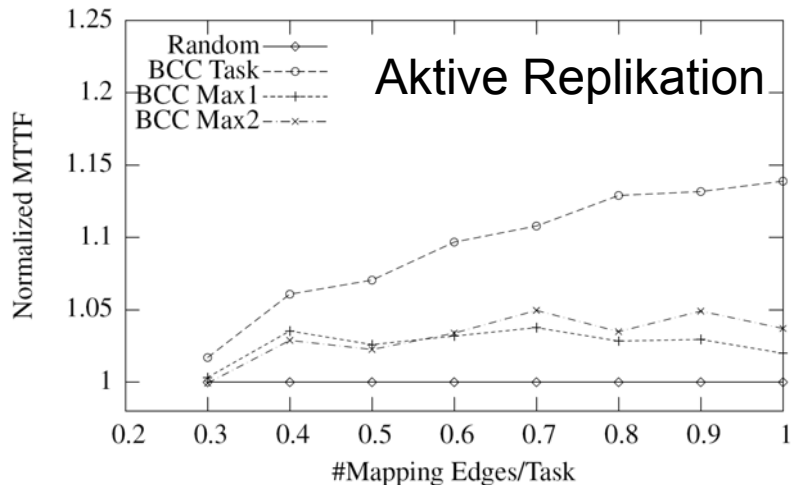
- Annahme:  
Jeder Task belegt 50% der auf einem Knoten verfügbaren Last.
- Tasks kommunizieren untereinander
- Ein Replikat belegt erst nach seiner Aktivierung Last.



- Ein einfacher Ressourcendefekt ist tolerierbar!

# Experimental Evaluation

- Simulative Bestimmung der Mean-Time-To-Failure (MTTF)
- Vergleich mit randomisierter Replikation:
  - Aktive Replikation verbessert die MTTF bis zu 14%
  - Passive Replikation verbessert die MTTF bis zu 23%



- T. Streichert, M. Glaß, R. Wanka, C. Haubelt and J. Teich: *Topology-Aware Replica Placement in Fault-Tolerant Embedded Networks*, ARCS08, Dresden, Deutschland, 2008
- T. Streichert, C. Strengert, D. Koch, C. Haubelt and J. Teich: *Communication Aware Optimization of the Task Binding in Hardware/Software Reconfigurable Networks*, Journal on Integrated Circuits and Systems, Vol. 2, Nr 1, pp. 29-36, März 2007
- T. Streichert, C. Haubelt, D. Koch and J. Teich: *Concepts for Self-Adaptive and Self-Healing Networked Embedded Systems*, Organic Computing, pp. 241-260, Springer, 2008

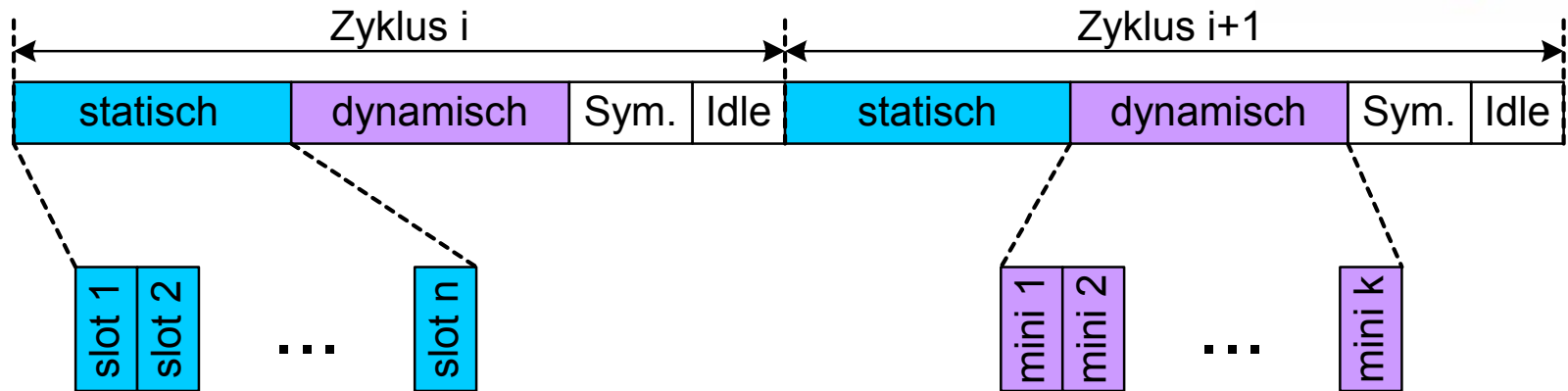
# Gliederung

---

- Motivation
- Replikplatzierung
- Reparaturstrategien
  - Task-Migration in TDMA-Netzwerken
- Systemintegration / Betriebssystemdienste
  - Rekonfigurierbare Kommunikation
  - ReCoBus-Builder
  - Kopplung ReCoNodes – ReCoNets
- Bearbeitungsstand

# Task-Migration in TDMA-Netzwerken

- Time Division Multiple Access am Beispiel FlexRay

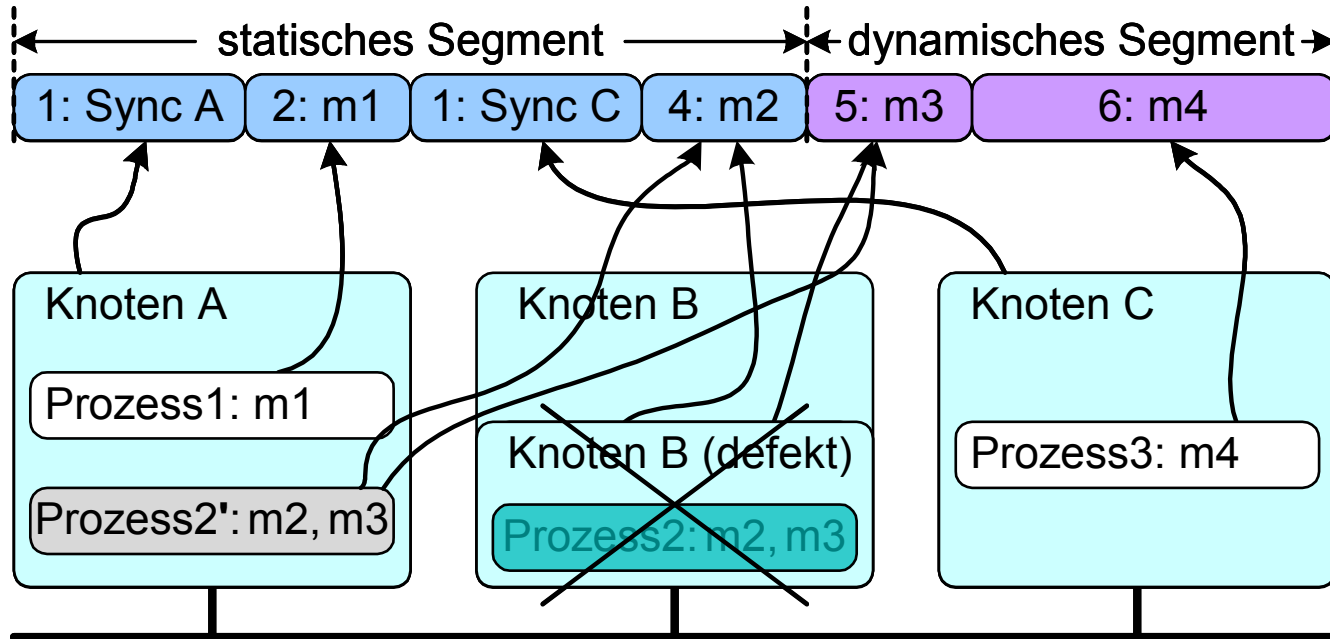


- **Statisches Segment** für TDMA Datenübertragung
- **Dynamisches Segment** für Ereignisgesteuerte Kommunikation
- **Symbolfenster** für Administration (z.B.: WakeUp & Sleep)
- **Network Idle Time** zur Uhrensynchronisation

➤ Slotvergabe? ➤ **Rekonfigurationstechniken?** ➤ Overhead?

# Task-Migration in TDMA-Netzwerken

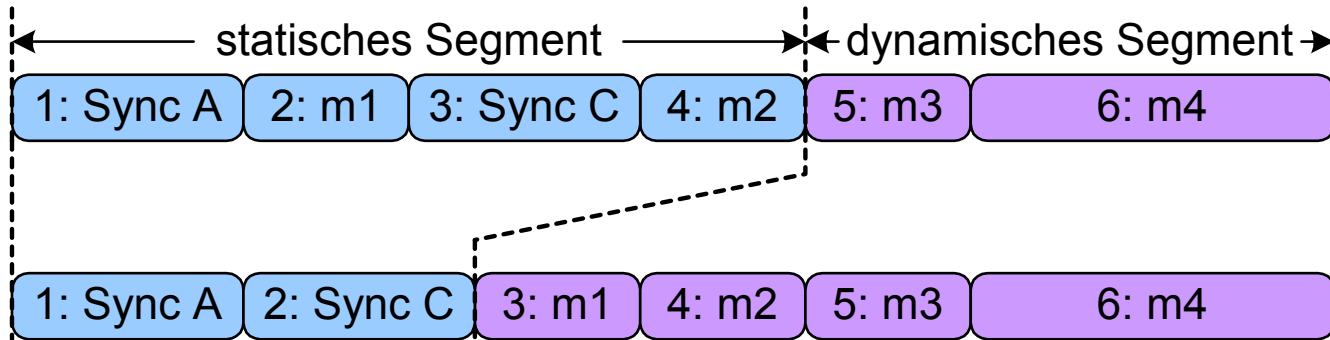
- Variante 1: Vollständige Rekonfiguration im Fehlerfall



- Sehr langsam:  $t_{\text{reset}} + t_{\text{config}} + t_{\text{integrate}} = 0.1 \text{ ms} + 4,4 \text{ ms} + \{25 \dots 30\} \text{ ms}$
- Ausfall der Kommunikation während der Rekonfiguration
- Mit jedem Communication Controller implementierbar

# Task-Migration in TDMA-Netzwerken

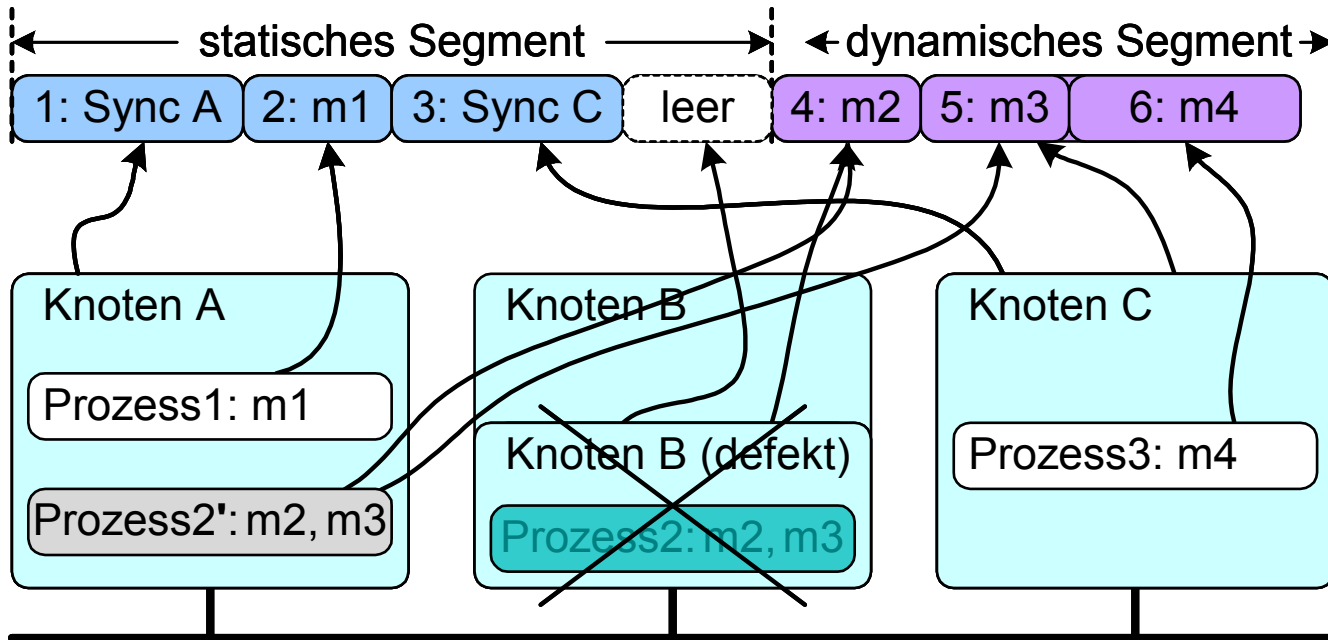
- Variante 2: Partielle Rekonfiguration im dynamischen Segment (Alternative Reallokation von Sendepuffern im dynamischen Segment zur Laufzeit)



- ➔ Mit vielen Communication Controllern implementierbar
- ➔ Unterbrechungsfreie Kommunikation während der Rekonfiguration
- ➔ Bandbreitengarantie
  - Priorisierung
  - Parametrisierung (Länge des dynamischen Segments)
- ➔ Latenz von einem Buszyklus

# Task-Migration in TDMA-Netzwerken

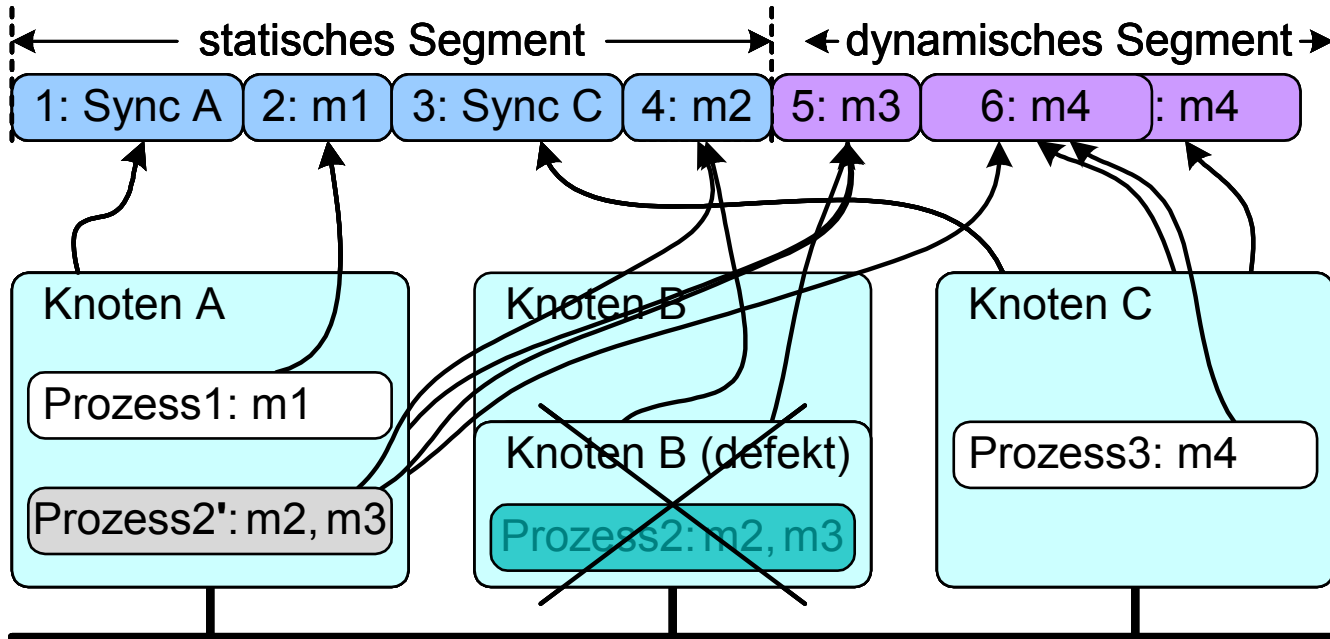
- Variante 3: Sekundäre Empfangspuffer im dynamischen Segment



- Umkonfigurieren der Kommunikation innerhalb eines Zyklus
- Verschwendung von Bandbreite

# Task-Migration in TDMA-Netzwerken

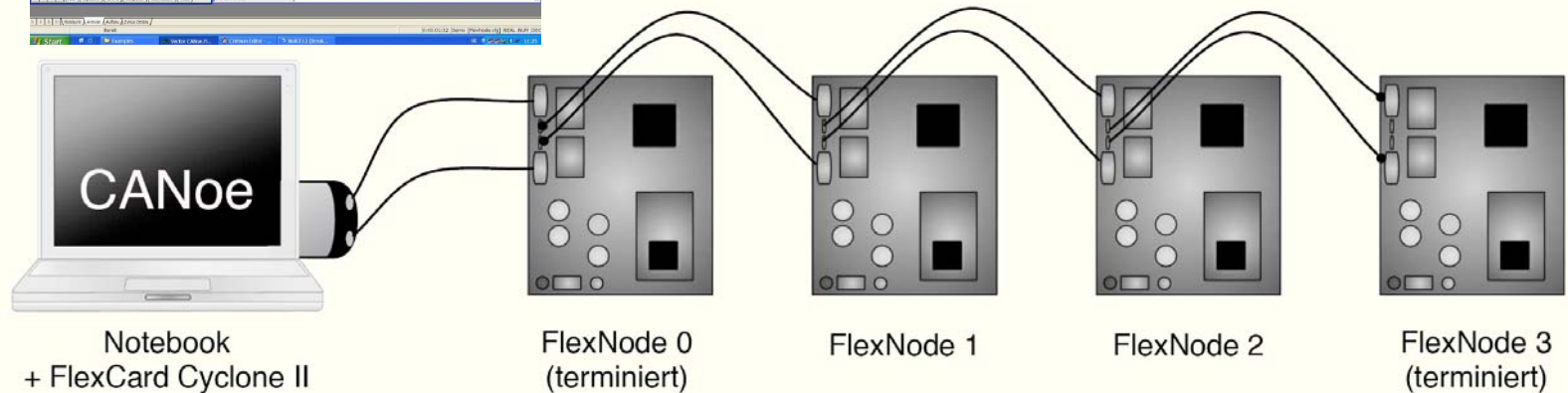
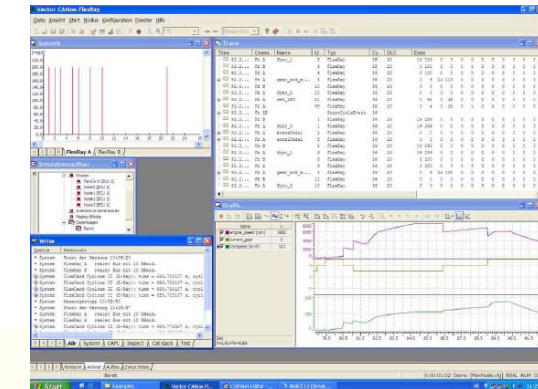
## ➤ Variante 4: Partielle Rekonfiguration



- Umkonfigurieren der Kommunikation innerhalb eines Zyklus
- Nicht auf jeder Hardware umsetzbar

# Task-Migration in TDMA-Netzwerken

## ➤ Versuchsaufbau



R. Brendle, T. Streichert, D. Koch, C. Haubelt und J. Teich:  
*Dynamic Reconfiguration of FlexRay Schedules for Response Time Reduction in Asynchronous Fault-Tolerant Networks*, ARCS08, Dresden, Deutschland, 2008

# Gliederung

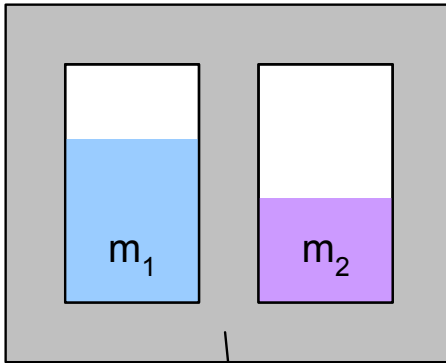
---

- Motivation
- Replikplatzierung
- Reparaturstrategien
  - Task-Migration in TDMA-Netzwerken
- **Systemintegration / Betriebssystemdienste**
  - Rekonfigurierbare Kommunikation
  - ReCoBus-Builder
  - Kopplung ReCoNodes – ReCoNets
- Bearbeitungsstand

# Rekonfigurierbare Kommunikation

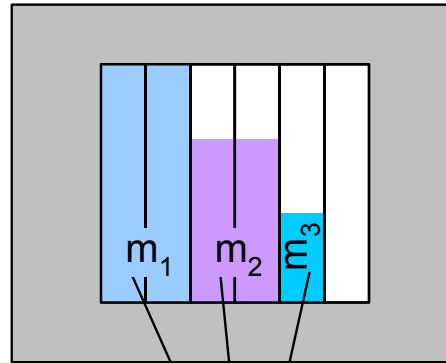
## ➤ Rekonfigurationstechniken

a) Insel-Rekonfiguration



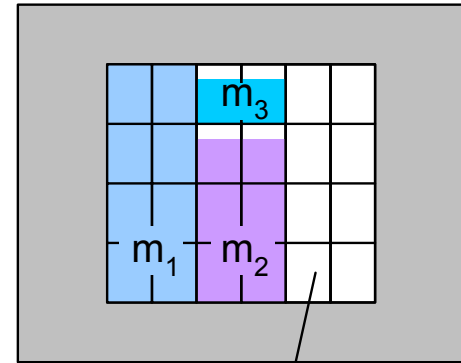
statischer Systemteil

b) Slot-Rekonfiguration



verschiedene Module

c) Gitter-Rekonfiguration

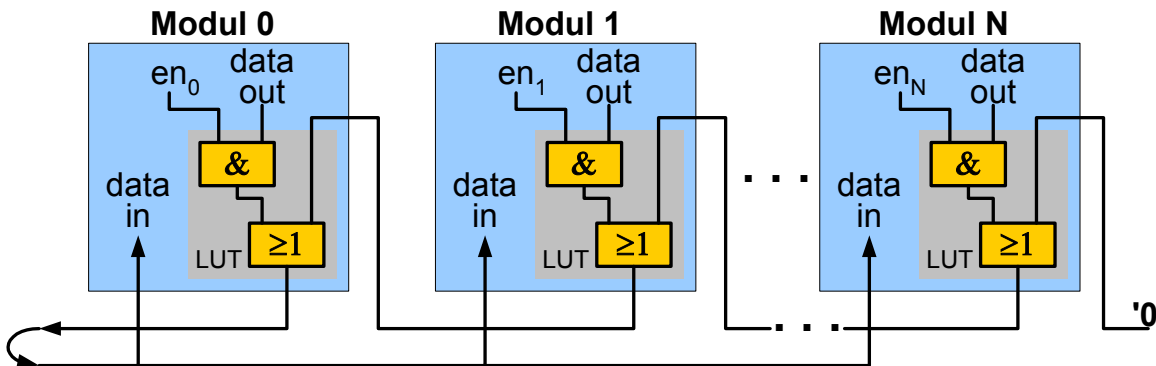
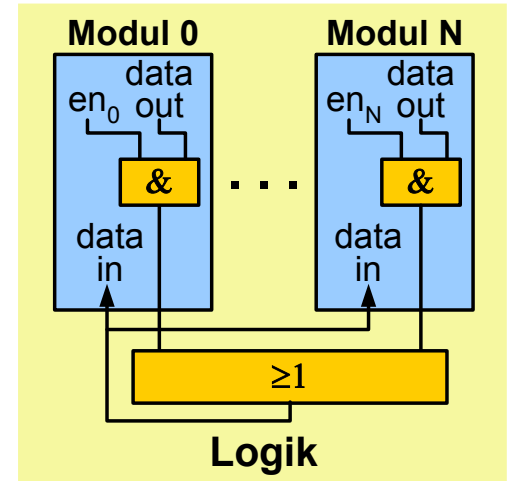
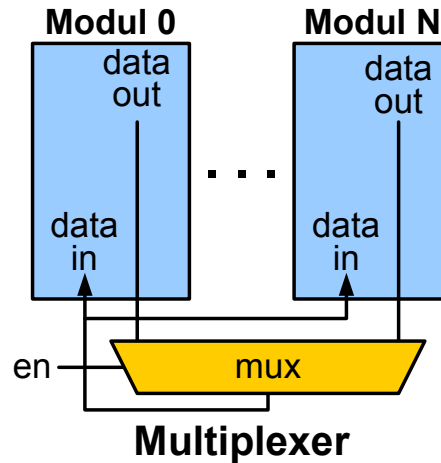
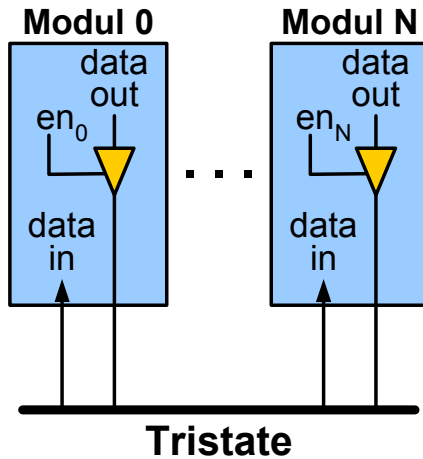


rekonfigurierbare Fläche

- Feine Granularität vermindert interne Fragmentierung (Verschnitt)
- Bedarf einer effizienten Kommunikationsinfrastruktur
  - Busse (Shared-Memory-Kommunikation)
  - Punkt-zu-Punkt-Verbindungen (I/O, Fifo-Kanäle)

# ReCoBus-Kommunikation

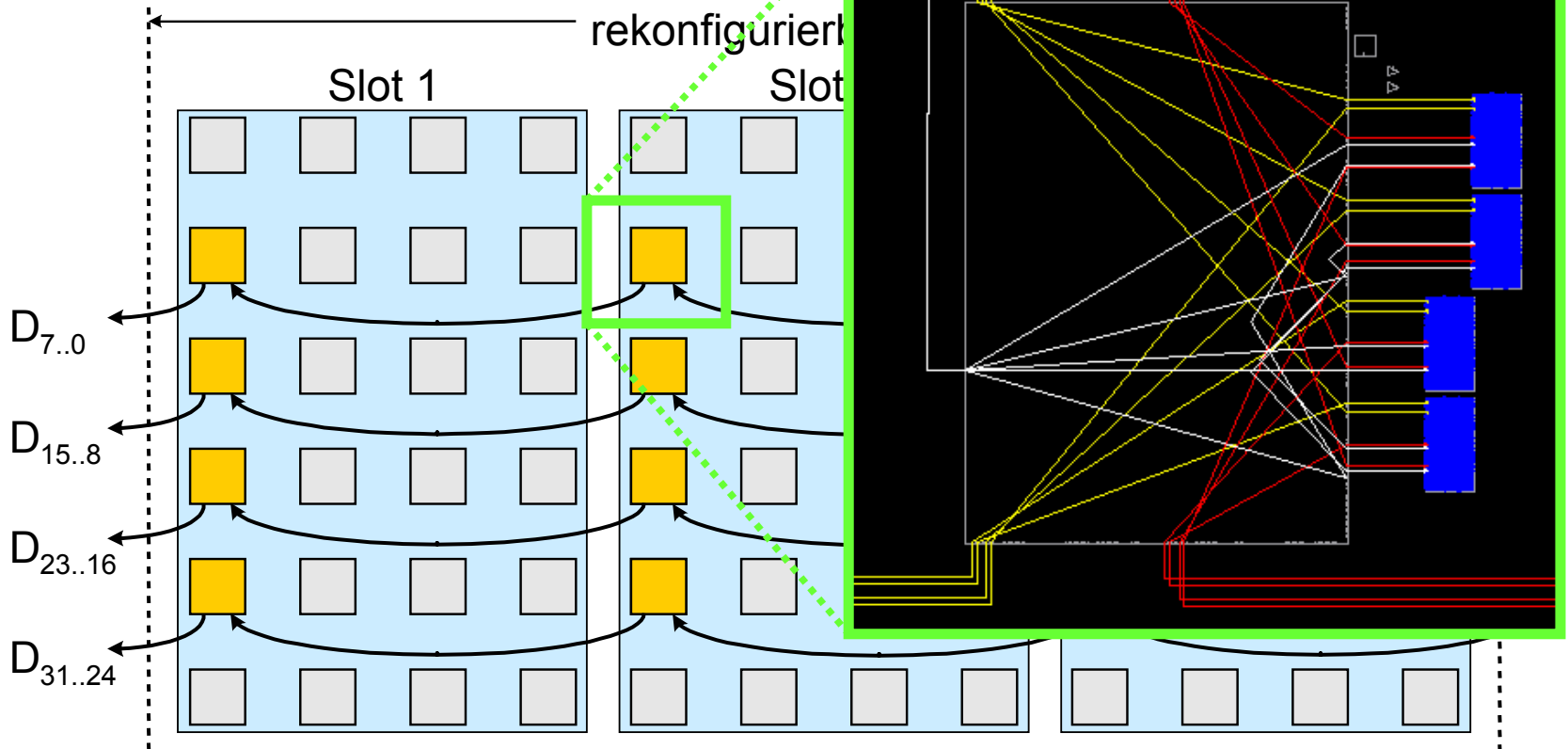
- Grundidee eines Busses für FPGA-basierte rekonfigurierbare SoCs am Beispiel von gemeinsam genutzten Lesedaten:



- Gleichförmige Verdrahtung
- Identische Logik innerhalb des Busses

# ReCoBus-Kommunikation

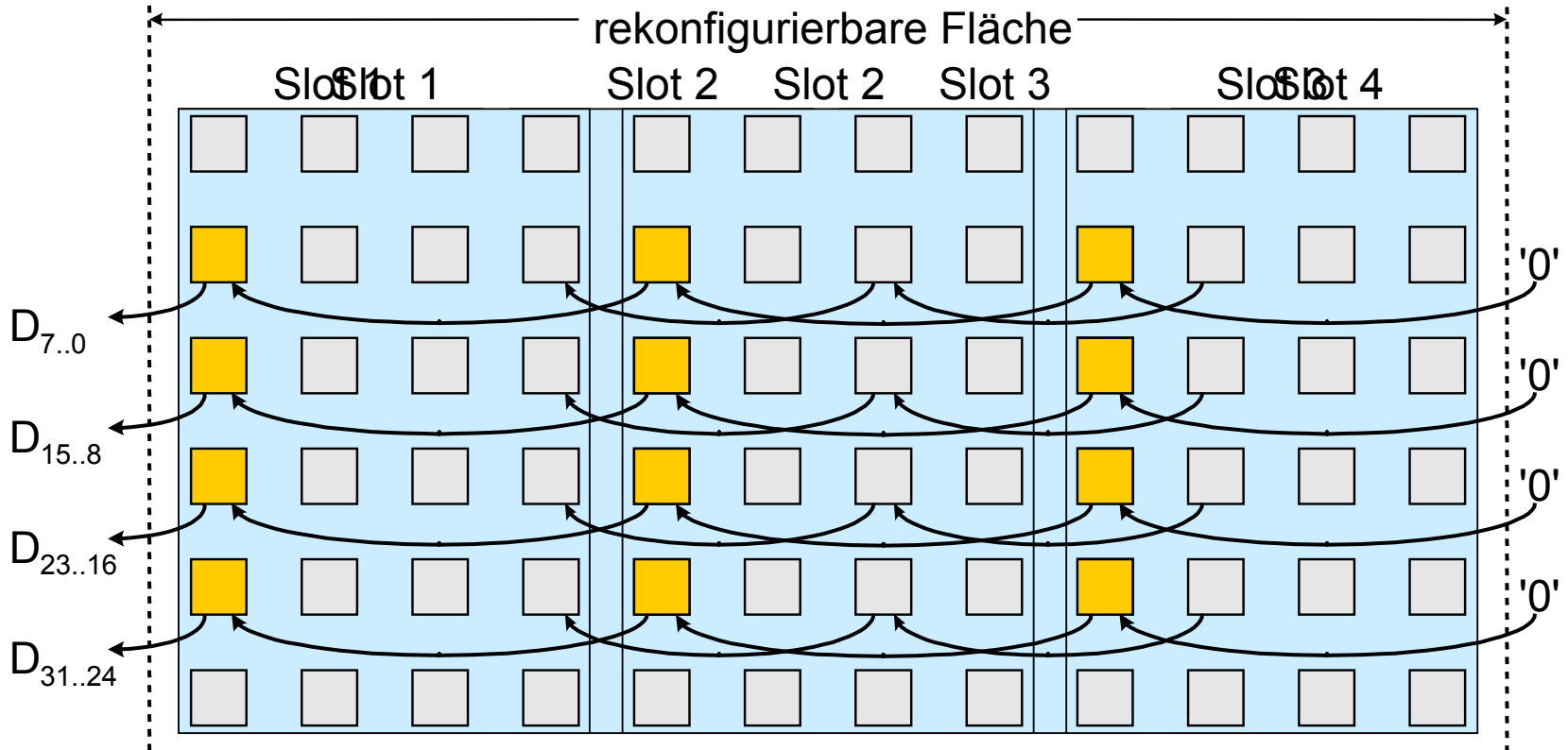
- Problem: Struktur einer Multiplexer-Kette ist unäunstia für feingranulare Implementierungen



- Logik-Overhead:  $4/24 = 17\%$

# ReCoBus-Kommunikation

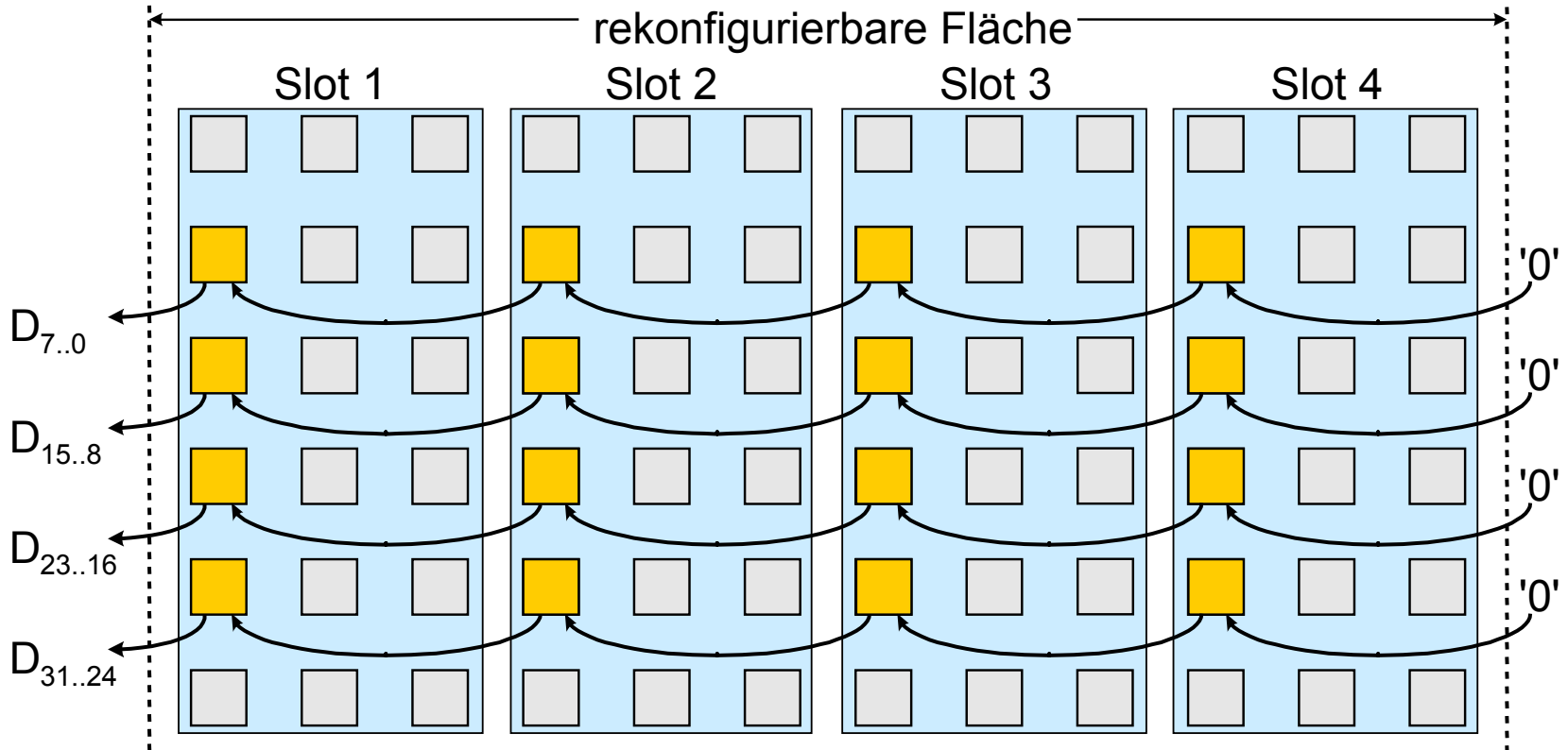
- Problem: Struktur einer Multiplexer-Kette ist ungünstig für feingranulare Implementierungen:



- Logik-Overhead:  $4/18 = 22\%$

# ReCoBus-Kommunikation

- Problem: Struktur einer Multiplexer-Kette ist ungünstig für feingranulare Implementierungen:

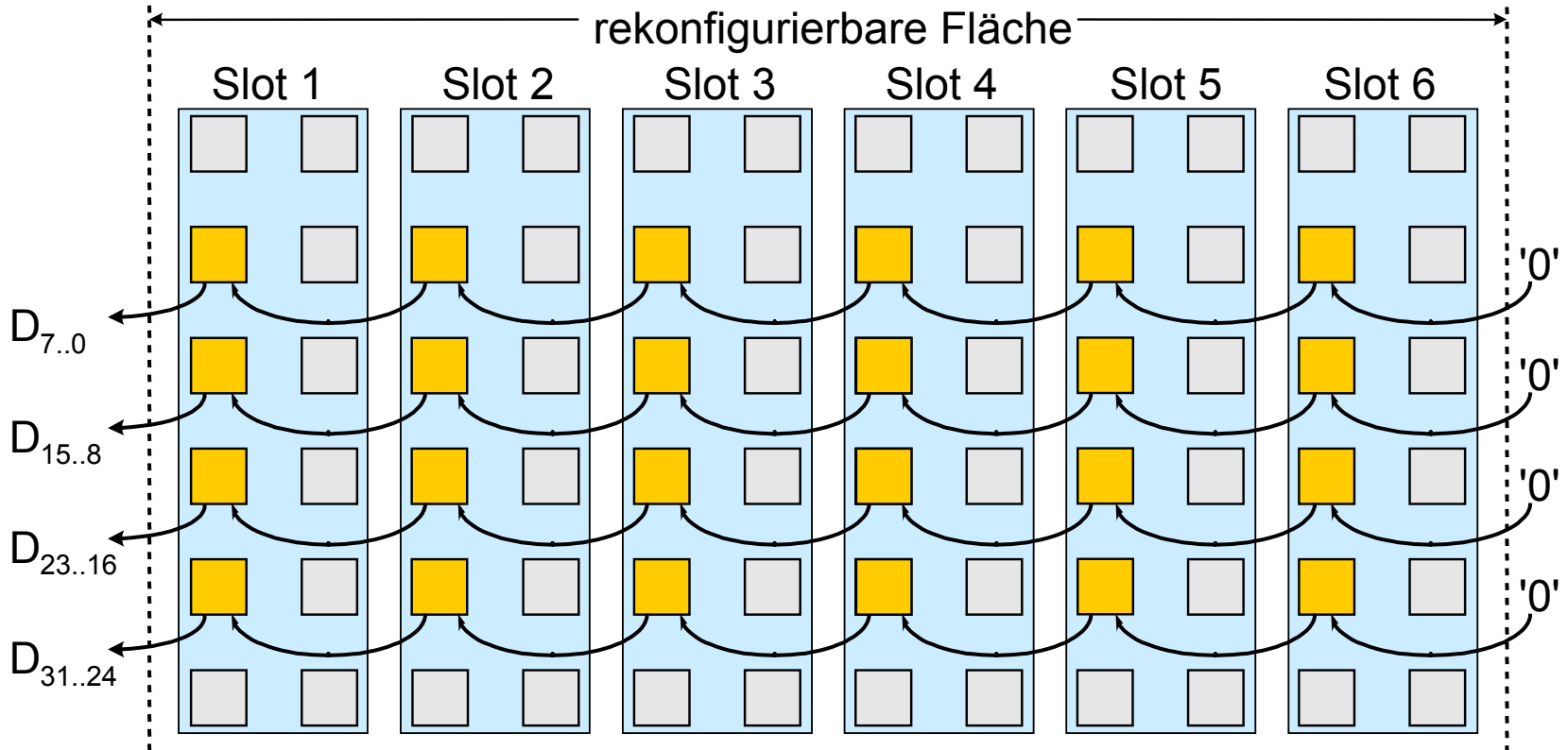


- Logik-Overhead:  $4/24 = 17\%$



# ReCoBus-Kommunikation

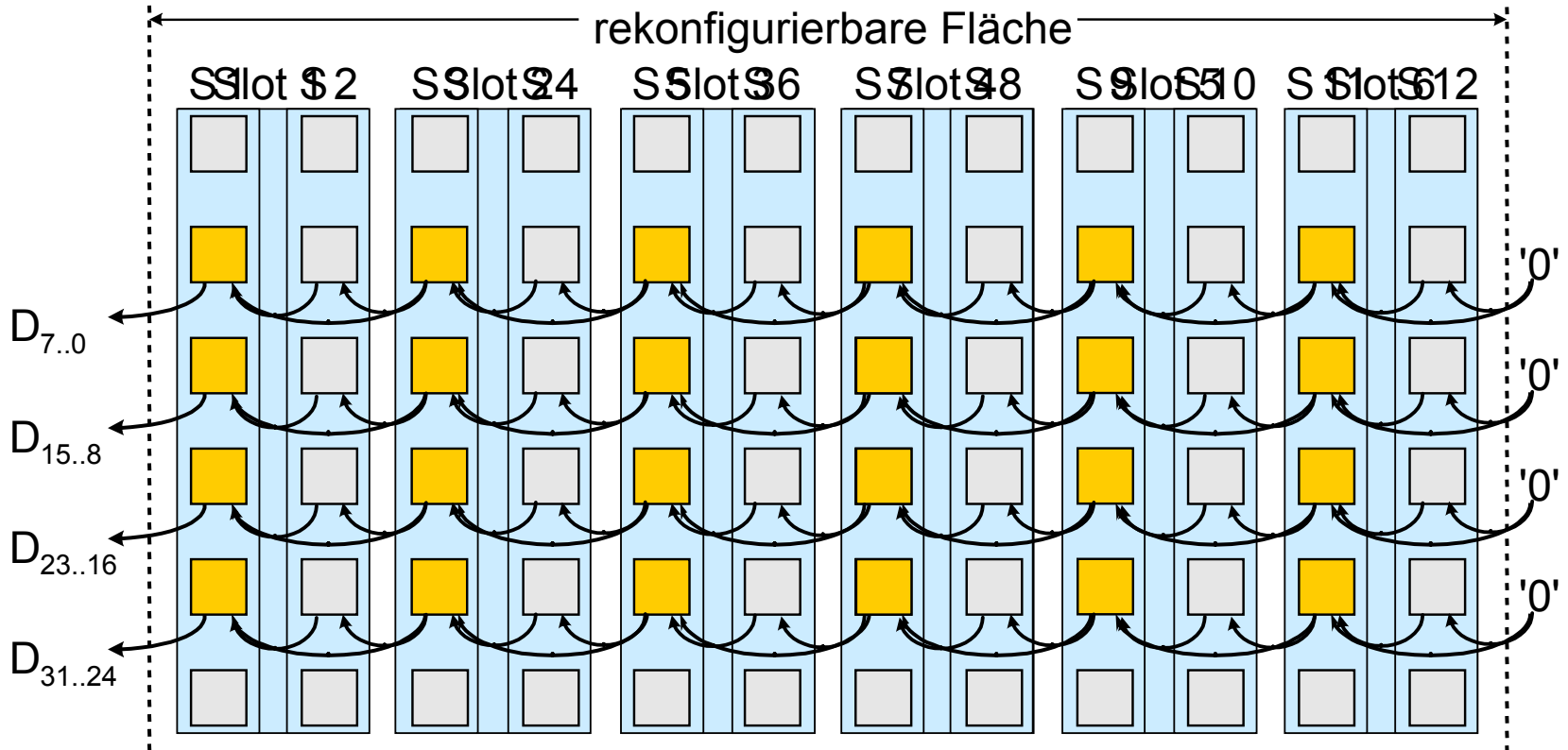
- Problem: Struktur einer Multiplexer-Kette ist ungünstig für feingranulare Implementierungen:



- **Logik-Overhead:  $4/12 = 33\%$**

# ReCoBus-Kommunikation

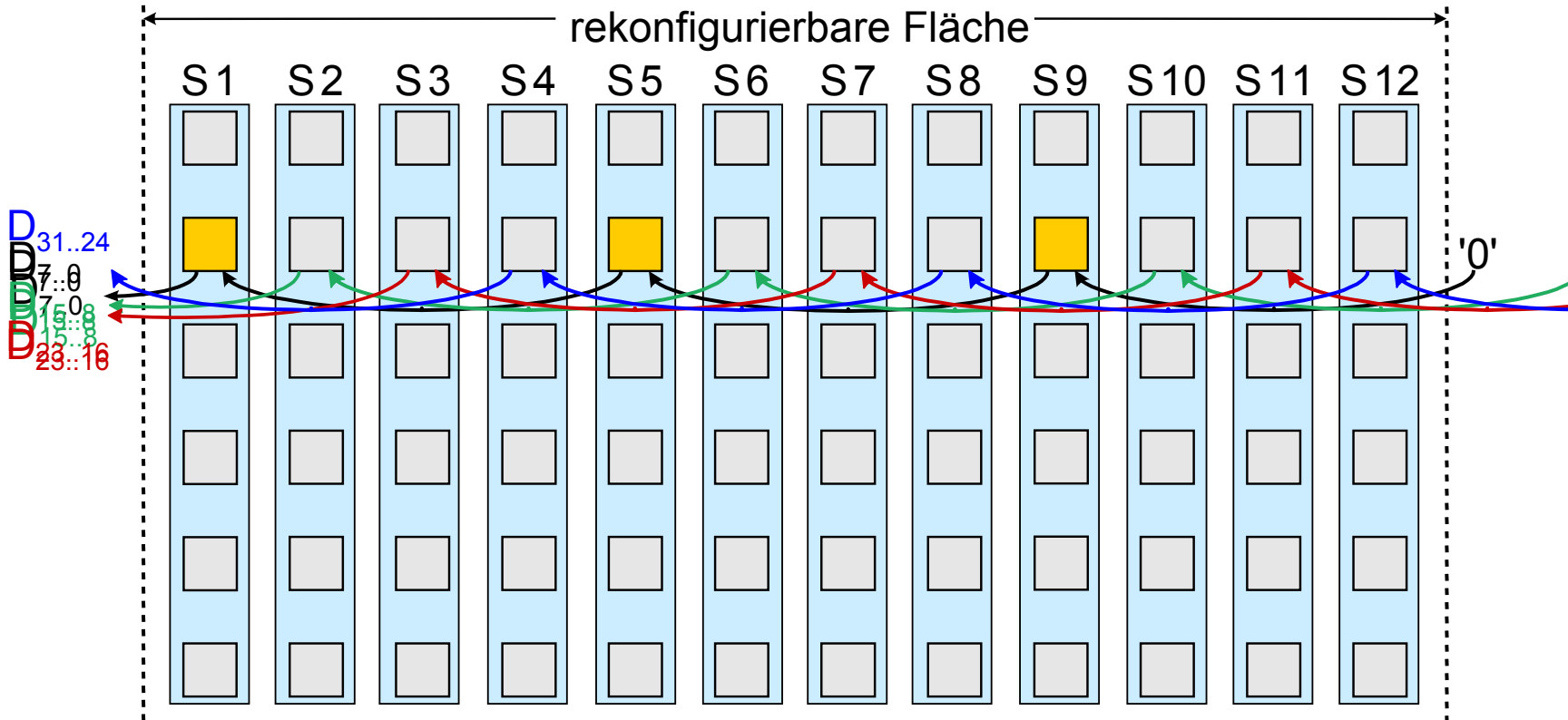
- Problem: Struktur einer Multiplexer-Kette ist ungünstig für feingranulare Implementierungen:



- Logik-Overhead:  $4/6 = 66\%$ ; **extrem große Latenz!**

# ReCoBus-Kommunikation

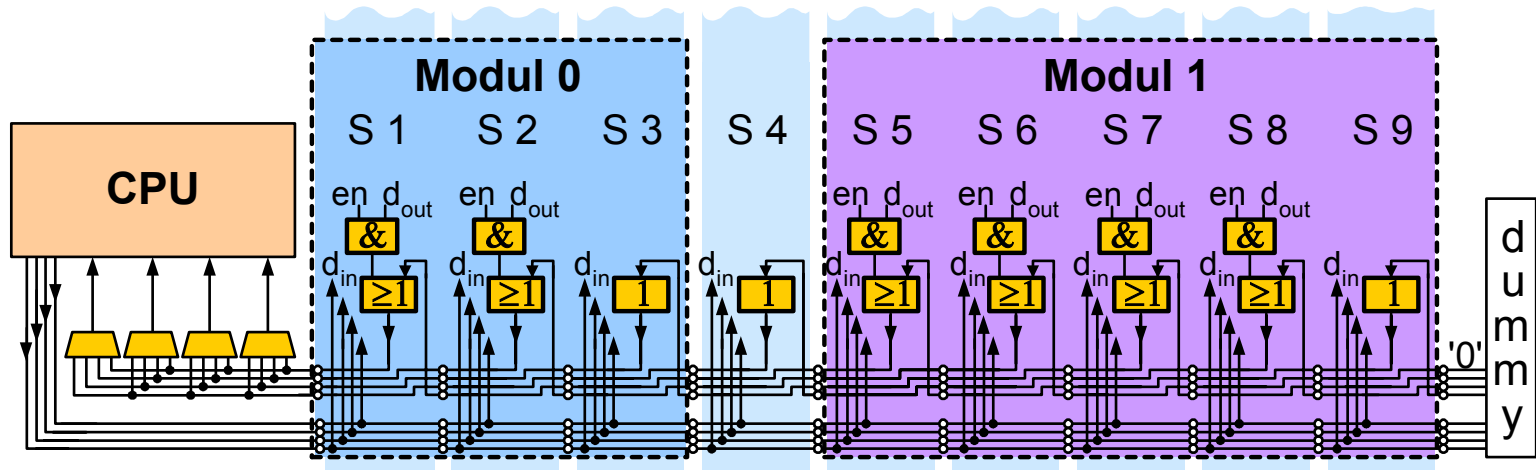
- Lösung: Verschränkung mehrere unabhängiger einzelner Multiplexer-Ketten:



- Logik-Overhead:  $1/6 = 17\%$ ; **geringe Latenz trotz feiner Granularität!**

# ReCoBus-Kommunikation

- Beispielsystem:



- Alignment-Multiplexer erlaubt freie Modulplatzierung
- Interface wächst automatisch mit der Modulkomplexität (Größe)

- D. Koch, T. Streichert, C. Haubelt und J. Teich: *Efficient Reconfigurable On-Chip Buses*, Europäisches Patent EP07017975, Hinterlegungstag 13.09.2007
- D. Koch, C. Haubelt und J. Teich: *Efficient Reconfigurable On-Chip Buses for FPGAs*, FCCM 2008, Stanford, Kalifornien, 2008

# ReCoBus – Besonderheiten

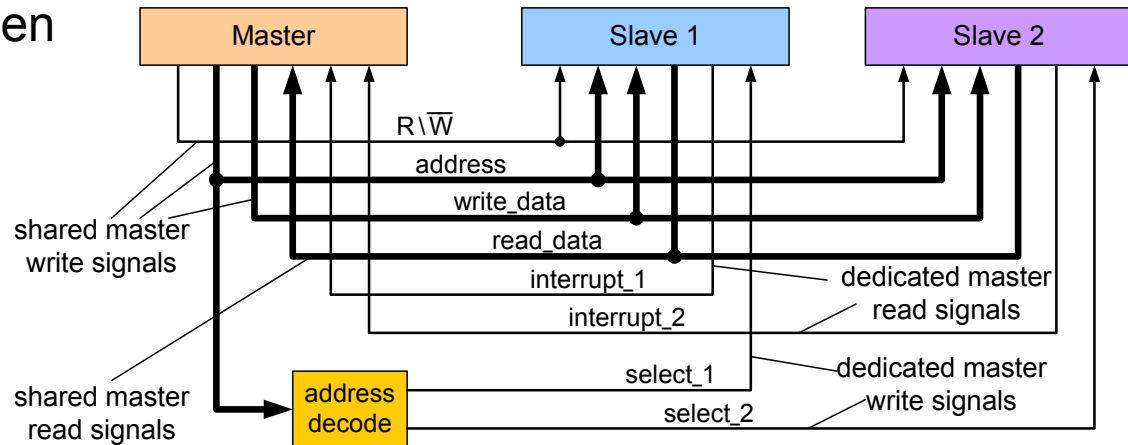
---

- Direkte Busanbindung der Module
- Kompatibel zu allen etablierten Standards (AMBA, Wishbone, ...)
- Modul-Relokation
- Flexible Platzierung & variable Modulgrößen
- Modul-Mehrfachinstantiierungen
- Sehr geringer Logik-Overhead
- Für hohen Durchsatz geeignet
- Hot-Swap-Fähig: keine Störung des Busses durch Rekonfiguration

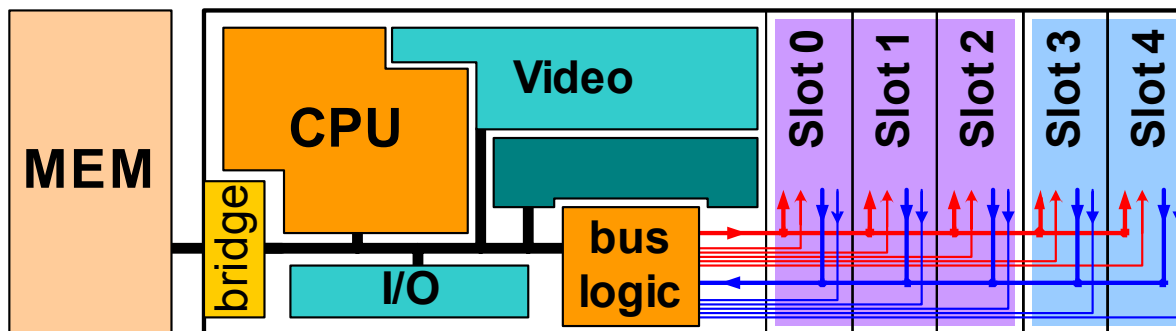
# ReCoBus

➤ Jedes Busprotokoll kann mit vier Klassen von Busleitungen implementiert werden

1. Shared Write
2. Dedicated Write
3. Shared Read
4. Dedicated Read



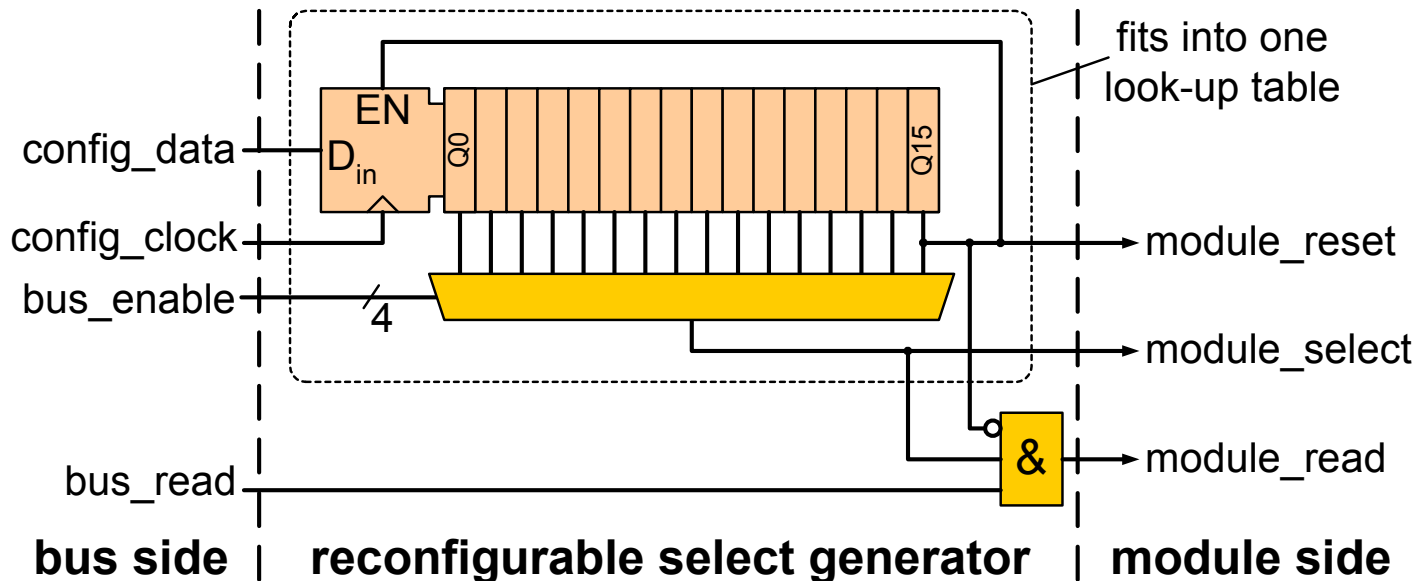
➤ Beispiel eines rekonfigurierbaren SoCs:



➤ Problem:  
gleichförmige  
Verdrahtung  
dedizierter  
Busleitungen

# ReCoBus – Dedizierte Schreibleitungen

- Lösung: gleichförmig verteilte Adresskomparatoren im Bus (Realisierung durch SRL16 Schieberegister-Primitive)

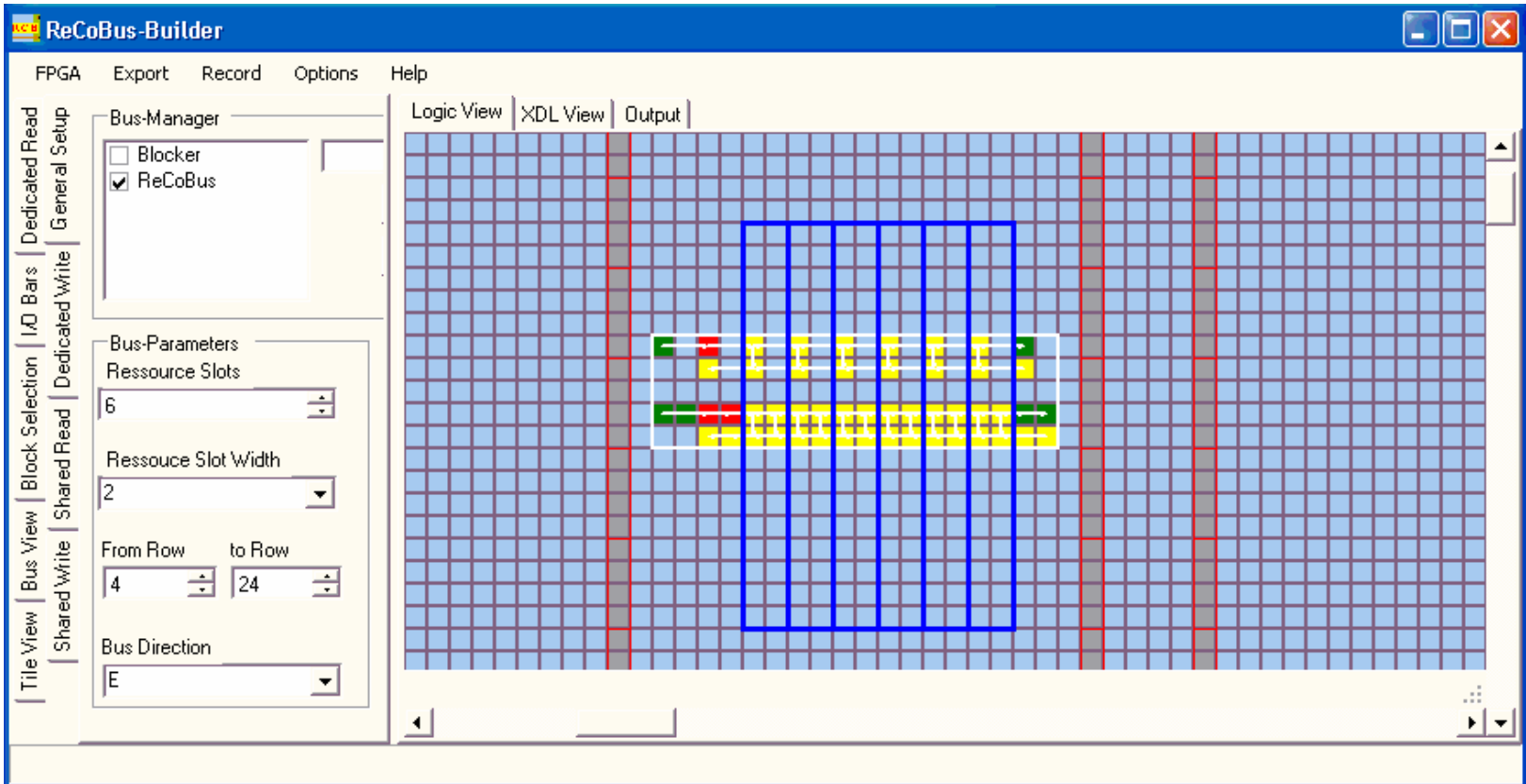


- Zweistufige Konfiguration:
  1. FPGA: Initialisiere Schieberegister mit 0xFFFF
  2. Logik: Konfiguriere internen Adresskomparator & aktiviere Modul



# ReCoBus-Builder

- Floorplanning- und Bussynthesewerkzeug zur RSoC-Generierung
- Unterstützt sämtliche Spartan-3, Virtex-II und Virtex-IIPro FPGAs



# ReCoBus-Builder

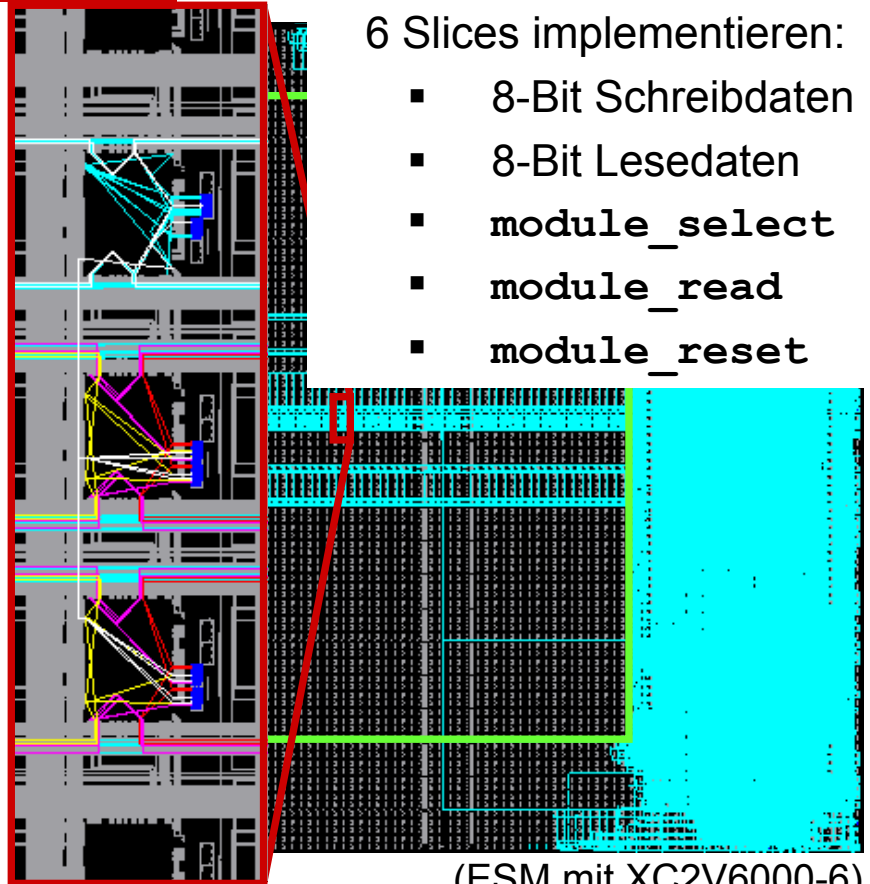
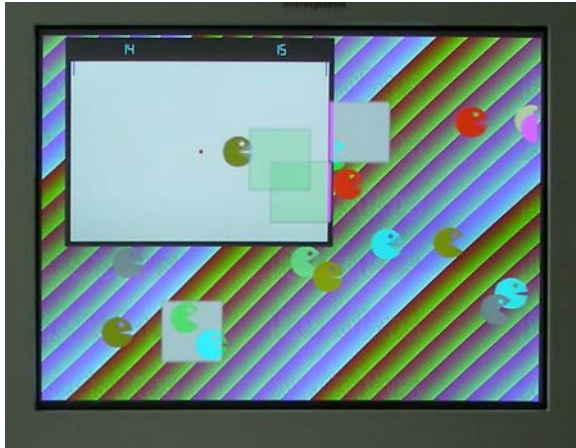
- Einfache Integration in die Xilinx-ISE Umgebung
- Blocker Macros als Workaround für mangelhafte Werkzeuge

**Kooperationen  
willkommen!**

D. Koch, C. Beckhoff and J. Teich:  
*ReCoBus-Builder – a Novel Tool and  
Technique to Build Statically and  
Dynamically Reconfigurable  
Systems for FPGAs,*  
Akzeptiert für die FPL 2008

# ReCoBus – Beispielsystem

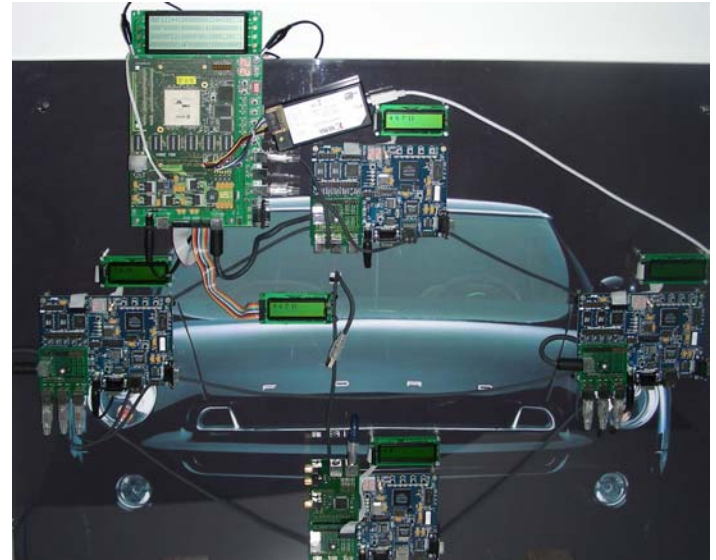
- RSoC mit 60 Slots
- 8-Bit Daten/Slot,  
(6x Verschränkt)
- 27 partielle Module  
(individuell adressierbar)
- Buslatenz < 10 ns



Vorgestellt auf der FCCM Configurable Computing Demo Night, Stanford, Kalifornien, 2008

# Kopplung: ReCoNodes – ReCoNets

- Rekonfigurations-Management:
  - Remote-Rekonfiguration
  - HW Bitstream-Dekomprimierer  
z.B.: 120 LUTs, 400 MB/s,  
garantierter Mindestdurchsatz,  
Kompressionsrate  $\sim \text{gzip --best}$
- Modul-Kompaktierung
  - Dynamik im Laufzeitsystem  
bewirkt Fragmentierung der  
rekonfigurierbaren FPGA-Fläche
  - Problem: die Anzahl der aktuell zur Verfügung stehenden Slots  
sagt allein nicht aus, ob ein Modul platziert werden kann.
  - Lösung: unterbrechungsfreie Kompaktierung



- D. Koch, C. Beckhoff und J. Teich: *Bitstream Decompression for High Speed FPGA Configuration from Slow Memories*, ICFPT07, Kokurakita, Kitakyushu, JAPAN, 2007
- S. Fekete, T. Kamphans, N. Schweer, J. van der Veen, J. Angermaier, D. Koch, und J. Teich: *No-break Dynamic Defragmentation of Reconfigurable Devices*, Akzeptiert für die FPL 2008

# Gliederung

---

- Motivation
- Replikplatzierung
- Reparaturstrategien
  - Task-Migration in TDMA-Netzwerken
- Systemintegration / Betriebssystemdienste
  - Rekonfigurierbare Kommunikation
  - ReCoBus-Builder
  - Kopplung ReCoNodes – ReCoNets
- **Bearbeitungsstand**

# Bearbeitungsstand (Dritte Förderperiode)

---

- Replikaplatzierung
  - AP1 Modellierung verschiedener Replizierungsstrategien
  - AP2 Online-Strategien zur Replika-Platzierung
  - AP3 Analytische Zuverlässigkeitsanalyse
  - AP4 Simulative Zuverlässigkeitsanalyse
- Echtzeitverhalten der Reparaturstrategien
  - AP5 Task-Migration in TDMA-Netzwerken (Busse)
  - AP6 TDMA in Punkt-zu-Punkt-Netzwerken
  - AP7 Task-Migration in echtzeitfähigen Punkt-zu-Punkt Netzen
  - AP8 Analyse der Zuverlässigkeit unter Echtzeitaspekten
- Kopplung der Mikro- und Makroebene
  - AP9 Kommunikationsstrukturen für das HW/SW-Morphing
  - AP10 Integration von Online-Platzierungsverfahren
  - AP11 Unterstützung der ESM in der ReCoNets-Entwurfsmethodik
- Implementierung
  - AP12 Prototypische Umsetzung und Demonstrator

