



# **NETWORK ON CHIP**

**Architekturen, Herausforderungen, Lösungen**

**Thomas Frank**

Dresden, 24.05.2011

# Inhalt

1. Einleitung
2. Architektur
3. NoC Beispiele
4. Zusammenfassung

# 1. Einleitung

# Bedeutung

## Network On Chip:

- Unterscheidung zw. globaler und lokaler Kommunikation in *System On Chip* (SoC)
- globale Kommunikation auf SoC
- Vereinheitlichung aktueller Trends statt Alternative
- Untermenge von SoC

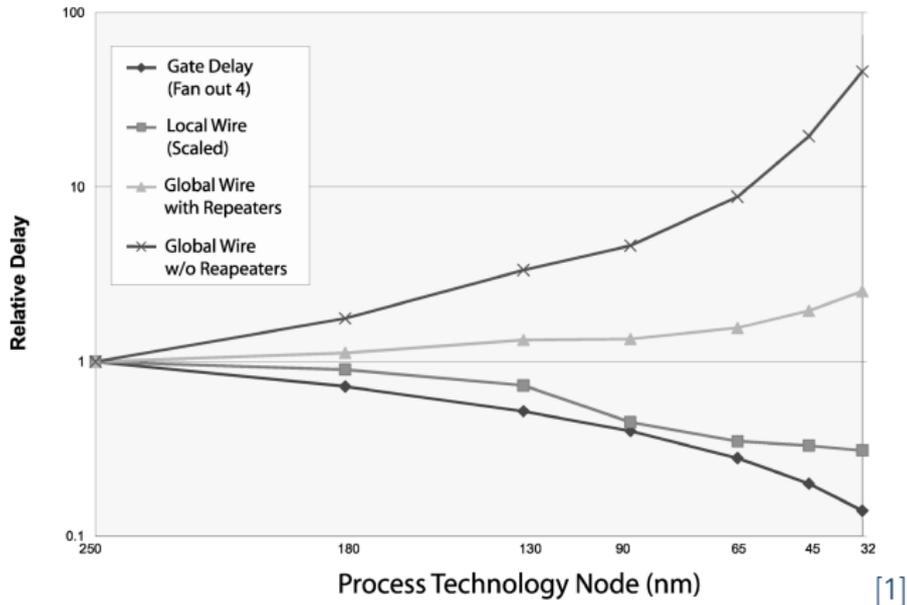
# Herausforderungen

## Paradigmenwechsel

- früher:**
- Rechnen teuer/langsam
  - Kommunikation günstig(Verdrahtung)
- heute:**
- Rechnen wird günstiger (Moore's Law)
  - Kommunikation stößt an physikalische Grenzen(Latenz, Energieverbrauch)

⇒ Trend zum kommunikationsorientierten Design

# Relative Leitungsverzögerungszeit



# Probleme

## **Leitungen**

- Drahtwiderstand(pro Fläche) steigt mit sinkendem Durchmesser
- Kapazität ändert sich wenig
- Länge globaler Verbindungen sinkt nur langsam

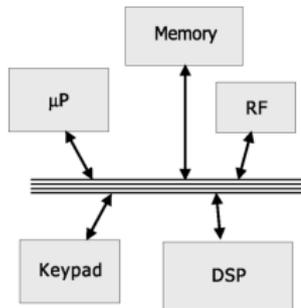
## **Synchronisation**

- Problem, je kleiner Struktur und je größer Chipfläche
- Realisierung GALS-Systeme(global asynchron lokal synchron)

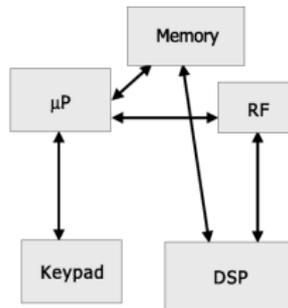
## **Produktivität**

- Design-Zyklus möglichst kurz
- Problemaufteilung: separat entwickelte Kerne mit standardisierter Kommunikationsschnittstelle leicht verbinden

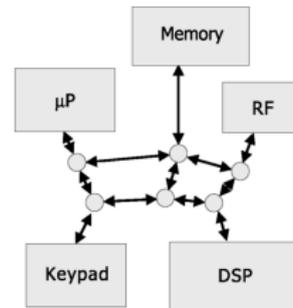
## Beispiele für Kommunikationsstrukturen



bus



point-to-point



network

[1]

# Kommunikationsstrukturen

**Punkt-zu-Punkt:** - schnellste, einfachste Lösung  
- Anzahl Verbindungen steigt exponentiell mit Anzahl der Kerne

**Bus:** - einfache Modellierung, gut erforscht  
- bei vielen Kernen: wird schnell zum Flaschenhals  
- Crossbar ist eher eine Zwischenlösung

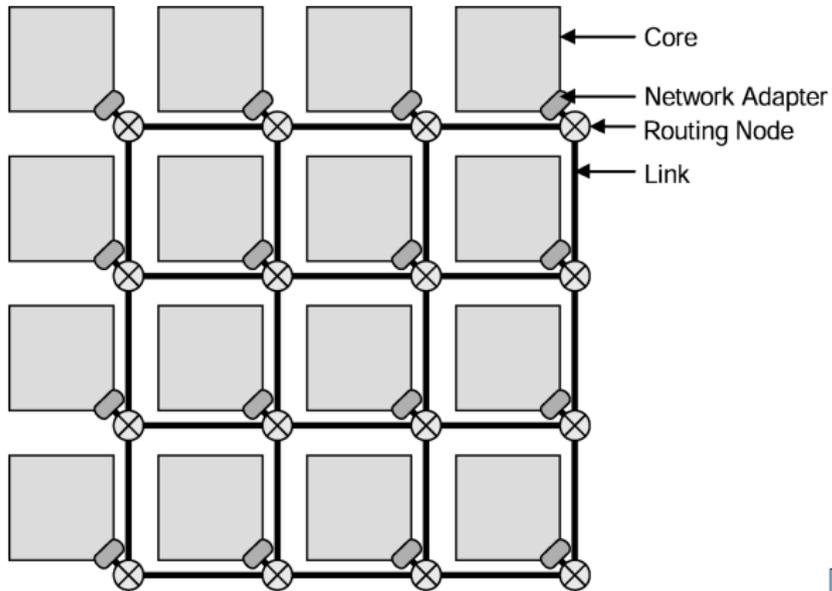
Für maximale Flexibilität ist ein globale, gegliederte und skalierbare Kommunikationsstruktur nötig.

→ **Netzwerk**



## 2. Architektur

## NoC Beispiel



[1]

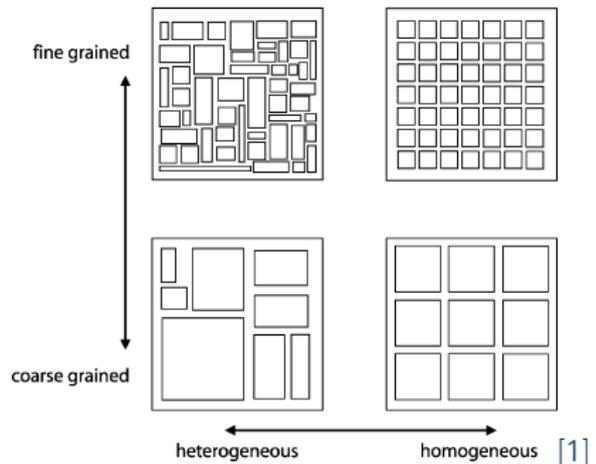
## NoC Komponenten

- Netzwerk Adapter:**
- implementieren Schnittstelle zu den Kernen
  - entkoppeln Berechnung von Kommunikation
- Router**
- implementieren die Verbindungsstrategie
  - leiten Daten entsprechend weiter
- Verbindungen:**
- Verbinden die Knoten
  - können mehrere physikalische oder logische Verbindungen enthalten

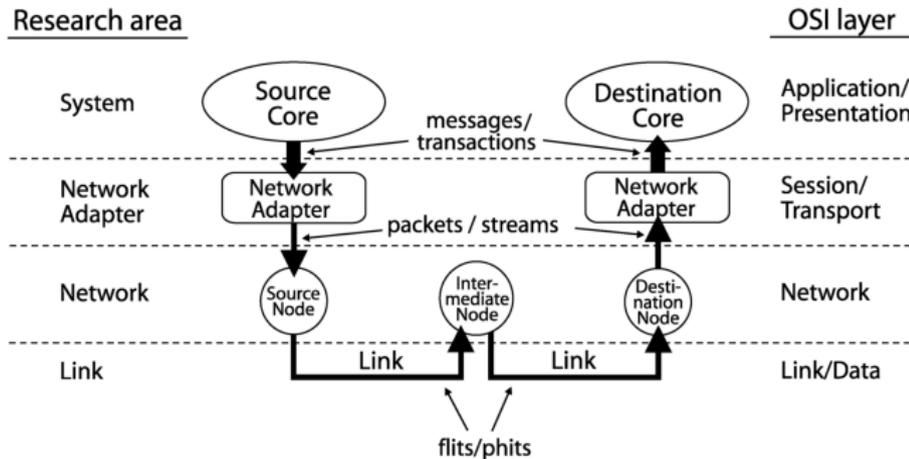
# Systemkomposition

## Eigenschaften

- Granularität
- Clustering
- Rekonfigurierbarkeit



# Netzwerkabstraktion



[1]

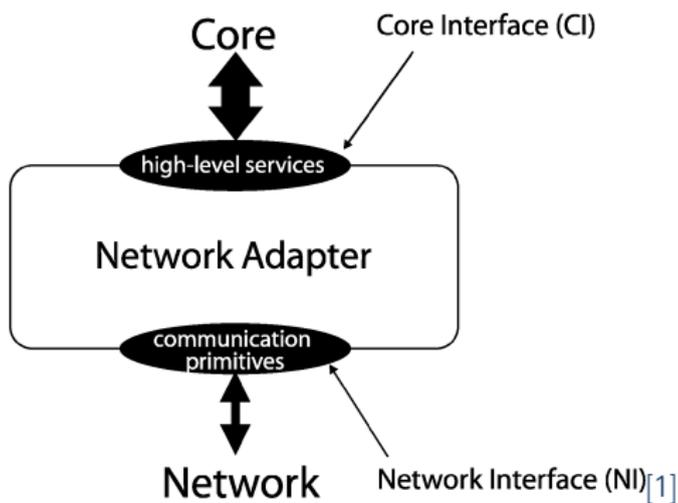
**Flits(flow control units):** atomare Einheiten eines Datenpakets/-flusses

**Phits(physical units):** kleinstes übertragbares Datenpaket

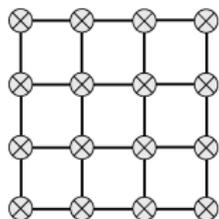
▷ haben meistens die gleiche Bedeutung

## Netzwerk Adapter

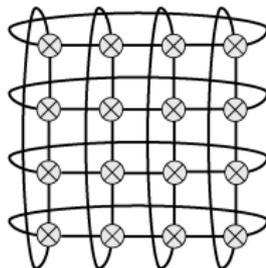
- Sockets: OCP / VCI
- NA Dienste:  
Entkoppelungsgrad: hoch /  
niedrig



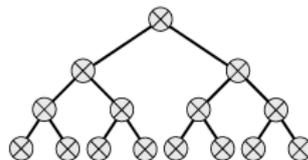
## Netzwerk - Reguläre Topologie



Mesh



Torus



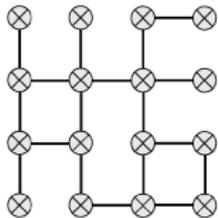
Binary Tree

[1]

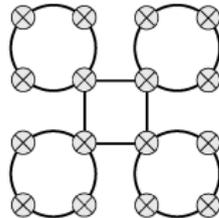
Netz: gute Auslastung

Baum: gute Ausnutzung der Verkehrslokalität

## Netzwerk - Irreguläre Topologie



Irregular Connectivity



Mixed Topology [1]

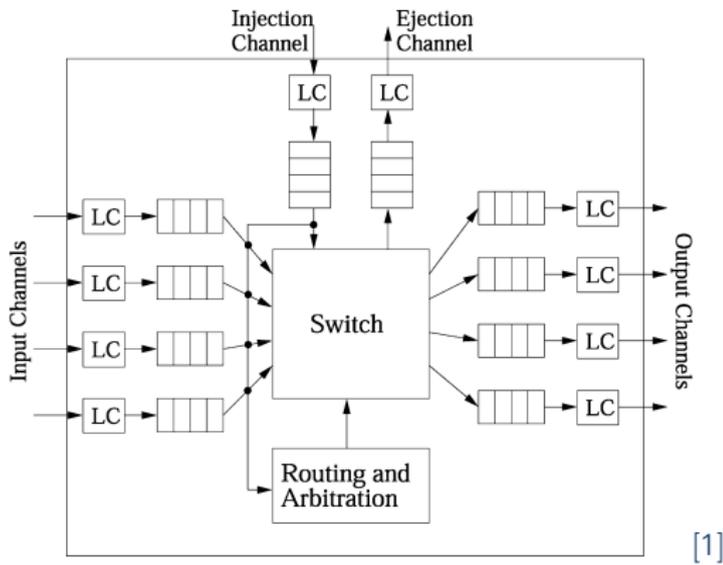
- Mischungen verschieden Topologien
- Fläche und Energieverbrauch skalieren nicht linear
- basieren generell auf dem Konzept *Clustering*

# Netzwerk - Protokoll

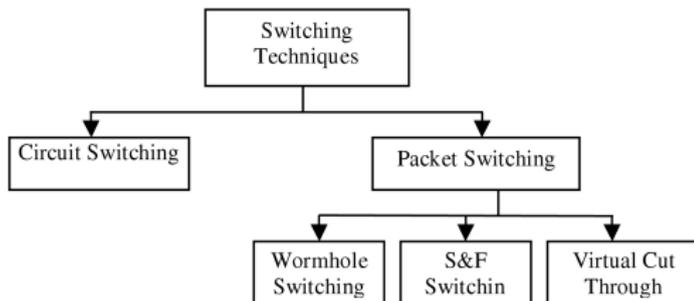
## Übertragungseigenschaften

- paketorientiert / Datenstrom
- verbindungsorientiert / verbindungslos
- deterministisch / adaptiv
- Wegfindung: minimal / nicht minimal
- Datenverlust / Datenverzögerung
- Entscheidungsfindung: zentral / dezentral

## Netzwerk - Protokoll



# Routingstrategien



<b>Strategie</b>	<b>Ressourcenbelegung bei Verstopfung</b>
Store-and-Forward	Verbindung mit beiden dazugehörigen Knoten
Wormhole	alle Knoten und Verbindungen, auf die das Paket verteilt ist
Virtual-Cut-Through	lokalen Knoten

# Netzwerk - Flusskontrolle

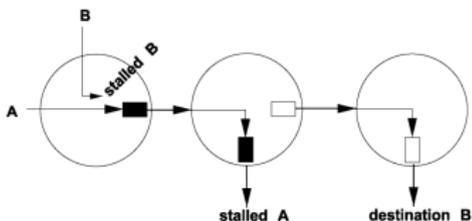
## Virtual Channels

- physikalische Verbindung in logische Verbindungen aufteilen
- + Deadlocks verhindern
- + Leitungsausnutzung
- + Performance
- + ermöglicht *Quality of Service*
- zusätzlicher Flächen- und Energieverbrauch, sowieso erhöhte Latenz

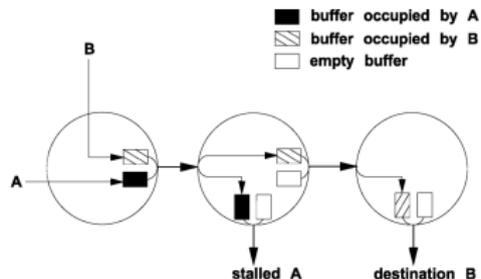
## Puffer

- speichern Input am Router
- nimmt Hauptteil der Fläche des Routers ein
- + können ruckartigen Datenverkehr abfangen
- verzögern Verstopfung
- Anzahl minimierbar mit statistischen Wissen über Datenverkehr (ohne Performanceverlust)
- pufferlose Architekturen auch möglich (meist bei *circuit switched*)

# Netzwerk - Flusskontrolle



Incomming data stream B is stalled by stream A



Virtual channels allow stream B to pass stalled stream A<sup>[1]</sup>

## Netzwerk - Quality of Service

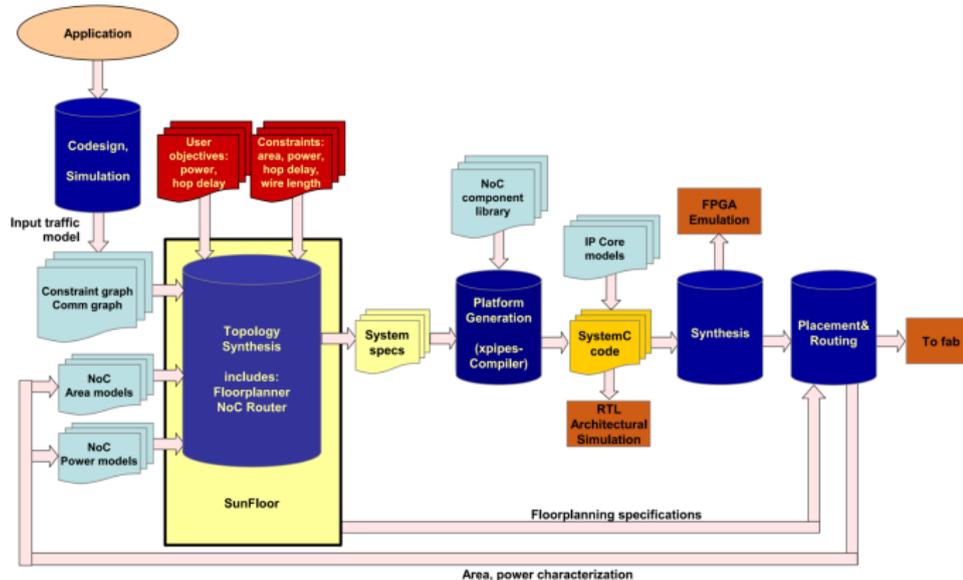
- bestimmter Dienst für Kern bereitgestellt, z.B.:
  - niedriger Latenz
  - hoher Durchsatz
  - geringer Energieverbrauch
  - hohe Signalqualität (wenig *Jitter*)
- Dienst wird ausgehandelt
- 2 Klassen:
  - best-effort (BE): keine Garantien
  - garantierter Service (GS)
  - GS/BE-Hybride auch Teil von Forschungen

## Netzwerk - Verbindungsebene

- Synchronisation: - GALS (global asynchron lokal synchron)
- Implementation: - Verbindungssegmentation mit Repeater-Puffer  
- Pipelining (mehr Durchsatz)
- Zuverlässigkeit: - übersprechen von Leitungen → Fehlererkennung/-korrektur  
- Fabrikationsfehler → Selbsttestmechanismen
- Kodierung: - Reduzierung Energieverbrauch (Negationsbit)  
- Fehleranfälligkeit (Paritätsbit)

## 3. NoC Beispiele

# xpipes

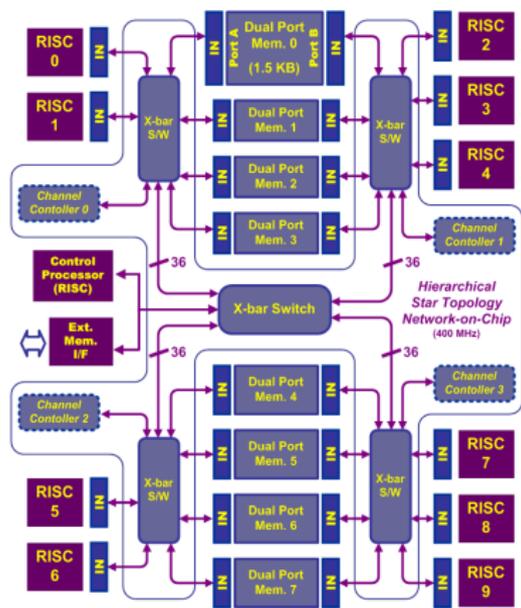


[3]

- xpipes Compiler: Generiert NoC Elemente von parametrisierbaren Blöcken (SystemC)
- Wormhole-Routing mit eigenem Protokoll "street sign"
- Pipeling zur Leitungssegmentierung
- CRC-Dekodierer arbeiten parallel mit den Routern

# Memory Centric NoC

- 10 RISC Prozessoren (200Mhz)
- Netzwerk getaktet auf 400Mhz
- 8 dual port Speicher
- 4 Channel Controller für Crossbar-Switches
- hierarchische Sternarchitektur
- mesochrones Netzwerk (Phase nicht synchron)
- Einsatz: optische Mustererkennung
- Zitat der Entwickler: "Fancy Network Concepts are not necessary"



## Weitere NoC - Architekturen

- Æthereal NoC
- QNoC
- SPIN NoC

## 4. Zusammenfassung

- Versuch der Separation von Berechnung und Kommunikation
- viel Forschung bzgl. Hardwarearchitekturprobleme vorhanden
- effiziente integrierte Lösung für Modellierung, Design und Entwicklung fehlt
- viel Forschung noch offen im Bereich Fläche-,Leistungs und -Energieeffizienz

## Quellen

- [1] T. Bjerregaard, S. Mahadevan, "A Survey of Research and Practices of Network-on-Chip", ACM Comput. Surv., vol. 38, Mar. 2006
- [2] Ankur Agarwal, Cyril Iskander, Ravi Shankar, "Survey of Network on Chip (NoC) Architectures & Contributions", Journ. of Enig., Comp. and Archit., Volume 3, Issue 1, 2009
- [3] Antonio Pullini et al., "NoC Design and Implementation in 65nm Technology", Network-On-Chip Symposium 2007
- [4] Donghyun Kim et al., "Solutions for Real Chip Implementation Issues of NoC and Their Application to Memory-Centric Networks-on-Chip", Network-On-Chip Symposium 2007