Friedrich-Alexander-Universität
Technische Fakultät

# Flexible and Efficient QoS Provisioning in AXI4-Based Network-on-Chip Architecture - A brief comprehension

An In-Depth Analysis of the Paper and its Key Findings

# Agenda

# Objective

## Goal

- Understanding the concepts of Wang & Lu (2022): „Flexible and Efficient **QoS** Provisioning in **AXI4**-Based **Network-on-Chip** Architecture“
- Critical reflection on their architecture and results

## Questions

- What makes their architecture special?
- What alternative approaches exist in the literature? (only discussed in the paper)
- Is their architecture still state of the art?

# Paper Concepts

# Paper Concepts

Network-on-Chip (NoC) - 1

## Definition

- Communication backbone in modern SoCs (connects CPUs, memory, accelerators)

## Motivation

- Shared buses became bottleneck with increasing number of cores
- Hierarchical buses & bus matrices improved things, but lacked scalability and flexibility

## NoC idea

- Replace buses with a **packet-based** on-chip network, inspired by computer networks
- Provides scalability, parallelism, IP reuse and energy efficiency

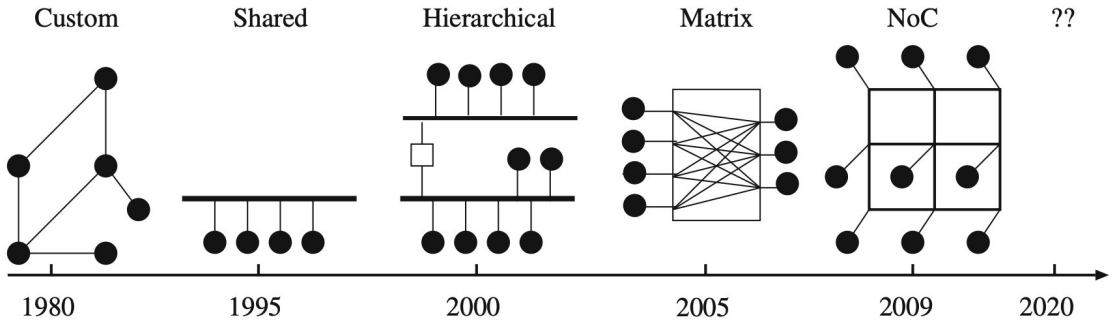


Fig. 1 - Evolution of Interconnects [1]

# Paper Concepts

Network-on-Chip (NoC) - 2

## Key Components

- **Routers**: Switch and forward packages
- **Links**: Connecting routers
- **Network Interfaces** (NIs): Translate between component (Protocols) and NoC packets (~Network)

## Advantages

- scalable bandwidth, modular design, better performance

## Challanges

- more design effort, area/ power overhead, complex routing

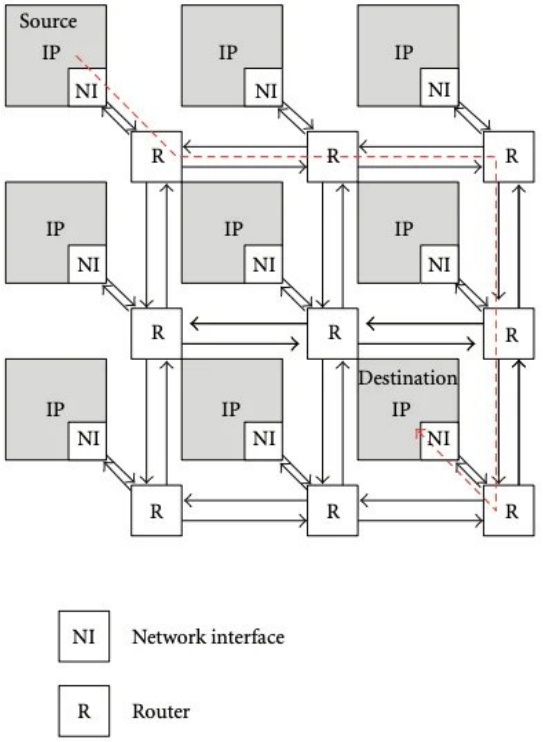


Fig. 2 - NoC Components [2]

# Paper Concepts

## AMBA & AXI Protocol

**AMBA Evolution:**

- 1996 - 1999: AMBA 1/2: Basic buses
- 2003: AMBA 3: AXI3 – out-of-order transactions
- 2010: AMBA 4: AXI4 family (**AXI4**, AXI4-Lite, AXI4-Stream)
- *2013+: AMBA 5: Cache-coherent extensions*

**Advantages**

- High bandwidth & scalability
- Flexible timing (decoupled address/ data)
- Fits diverse SoC needs

**AXI4 Key Features:**

- 5 independent channels (AW, W, B, AR, R)
- Separate READ & WRITE: parallelism & high troughput
- Burst transfers (fixed, incr, wrap)
- Out-of-Order transactions
- Handshake protocol (VALID/ READY)

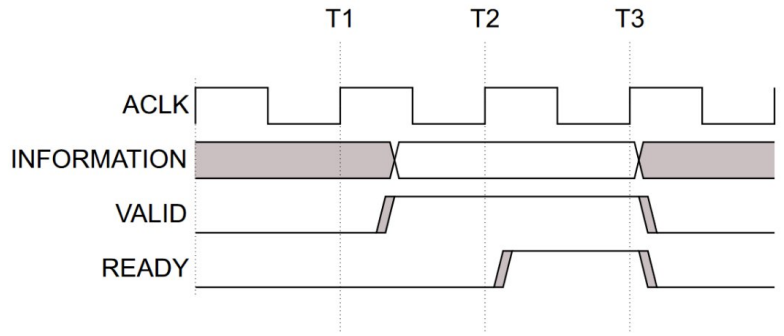


Fig. 3 - AXI Handshake (VALID before READY) [3]

# Paper Concepts

## Quality of Service (QoS)

### Definition

- Technique to manage & prioritize network traffic

### How QoS works

- classify & mark packets (by priority)
- queuing & scheduling (priority queues)
- bandwidth management

### Key Parameters

- **Bandwidth**/ Throughput: capacity vs. actual rate
- **Latency**: transmission delay
- **Jitter**: variation in packet arrival
- **Packet loss**: dropped data

### QoS Models

- Best Effort – no guarantees
- IntServ – strict guarantees (resource reservation)
- DiffServ – scalable, widely used (class-based)

# Paper Analysis

# Paper Analysis

## Overall Architecture

**Goal**

- Provide flexible QoS for AXI4-based SoCs

**Three QoS Classes**

- LCS – low latency, bursty traffic (CPU)
- GRS – guranteed bandwidth, streaming traffic (GPU)
- URS – best effort, fair resource sharing (I/O)

**Main Components**

- AXI Masters/ Slaves
- Network Interfaces (NIs) – protocol conversion, QoS
- Dual Subnetworks: VC-based for LCS or URS and TDM-based for GRS

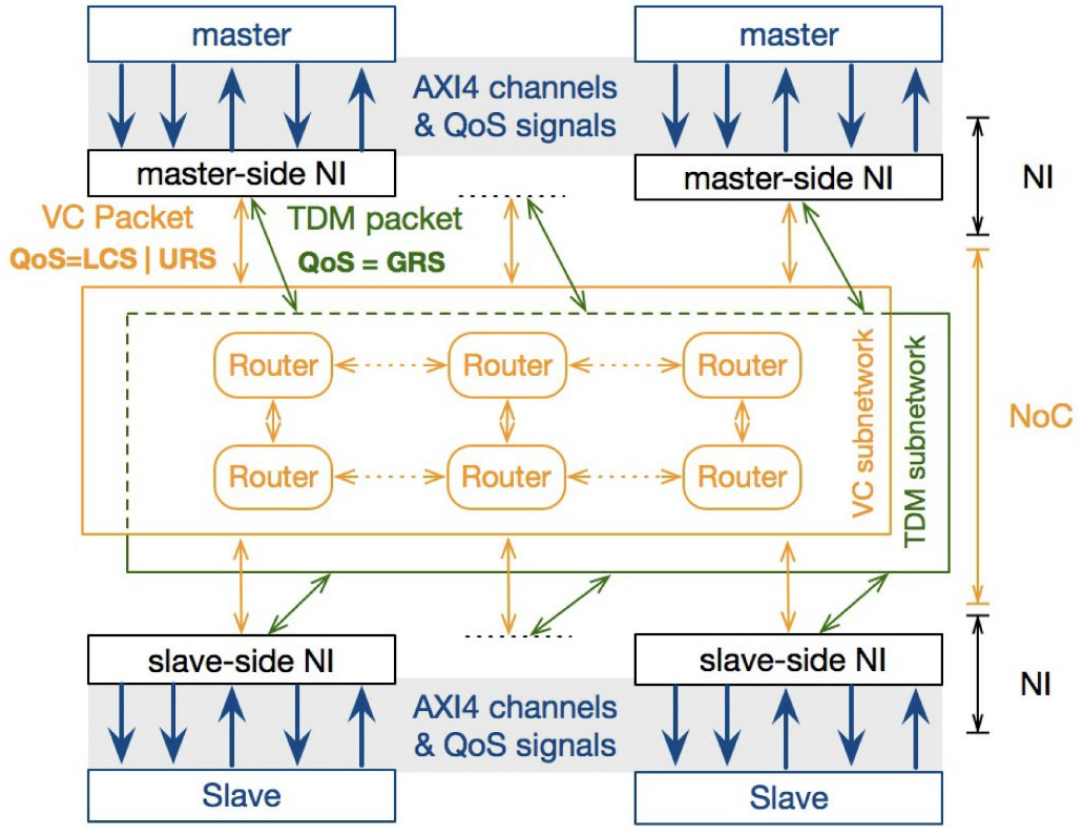


Fig. 4 - System Architecture [4]

# Paper Analysis

## Message Conversion & Network Interface

### Why Message Conversion?

- Direct mapping (5 AXI channels → 5 packet formats) = inefficient
- Adopted approach: 4 unified packet types
  - Read request / Read response
  - Write request / Write response
- Packets carry QoS label (LCS/GRS/URS)

### NI Roles

- direct packets to correct AXI channel / subnetwork
- Conversion: AXI4 ↔ NoC packets
- QoS inheritance: responses keep QoS class of request

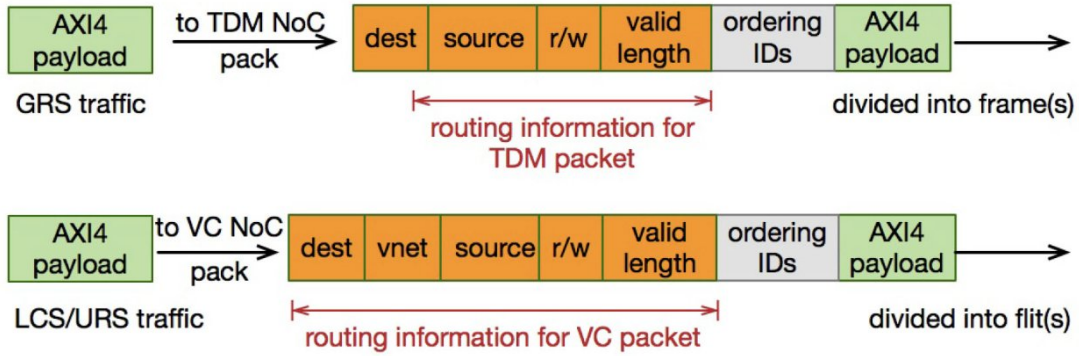


Fig. 5 - Message Format Conversion [4]

# Paper Analysis

## Dual Subnetworks & Traffic Converter

### VC Subnetwork (GRS, LCS)

- Supports different flow control schemes
- Goal: prioritize latency-sensitive LCS while maintaining fairness for URS

### TDM Subnetwork (GRS)

- Static routing via precomputed time slots
- Guarantees bandwidth and predictable time slots

### Traffic Converter Subnetwork (GRS)

- VC → TDM: offload LCS if VC congested
- VC → TDM: offload GRS if TDM congested
- Improves utilization, latency, throughput

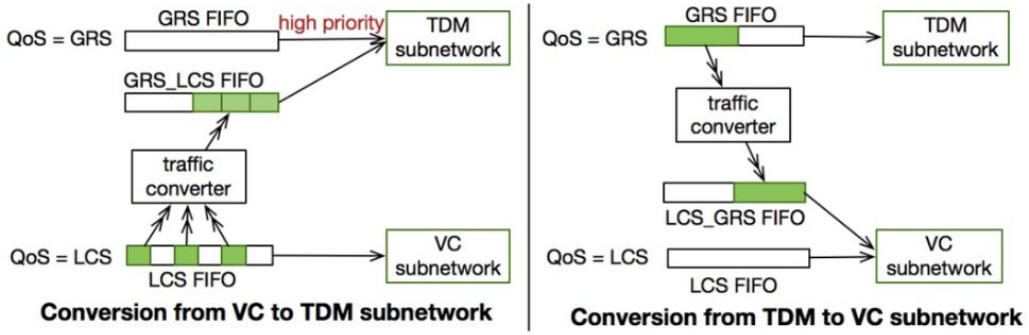


Fig. 6 - Traffic Conversion Unit [4]

# Results

# Results

## Key Findings & Conclusion

### Key Findings

- High Throughput
  - VC: 12.3 Gb/s, TDM: 7.4 Gb/s → ~19.7 Gb/s total
- Latency
  - LCS stays low with Individual_Shared flow control
  - URS higher delay, but fairness maintained
- Resource Utilization
  - VC ports: 4.4% → 16.5% utilization before saturation
  - TDM slots: ~17% utilization
- Traffic Converter
  - VC→TDM: cuts LCS latency
  - TDM→VC: reduces GRS queuing delay
  - Up to ~94% performance improvement

### Simulator Setup

- Custom Simulator (C++, BookSim2- & Gem5-based)
- 168 nodes, dual-subnetwork NoC (VC + TDM)
- Realistic traffic via two-level MMP generator

### Conclusion

- Dual-subnetwork
- Adaptive conversion achieves scalable throughput, low LCS latency, and balanced QoS across traffic types

# Critical Review

## Contributions

- AXI4-compatible NoC with flexible QoS provisioning
- Dual-subnetwork design
- NI: AXI4 → packet conversion + QoS inheritance
- Adaptive load balancin: VC ↔ TDM Conversion

## Strengths

- Integrates three QoS services in one unified framework
- Decouples AXI4 protocol details from NoC fabric
- Demonstrates tangible performance improvements

## Limitations

- Hardware complexity: Dual subnetworks + traffic converter
- Synthetic traffic only: Limited validation with real workloads
- Adaptability: Rule-based switching; no runtime intelligence

# Bibliography


[1] B. A. Abderazek, Multicore Systems On-Chip: Practical Software/Hardware Design, 2nd ed. 2013, vol. 7.

[2] J. Hertz, Why SoCs Need NoCs: Network on Chip and the Future of Computing, in ALL ABOUT CIRCUITS, 25.07.2025, (https://www.allaboutcircuits.com/news/why-socs-need-nocs-network-on-chip-and-future-computing/) – accessed 28.09.2025.

[3] P. Holzinger, On-Chip Bussysteme und Perpherie (Schwerpunkt CPU Entwicklung) in CPU-Entwicklung mit VHDL, SoSe 2025, Slide 12.

[4] B. Wang and Z. Lu, “Flexible and Efficient QoS Provisioning in AXI4-Based Network-on-Chip Archtecture,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 41, no. 5, pp. 1523–1536, May 2022.