A Low-Latency Asynchronous Interconnection Network with Early Arbitration Resolution

Georgios Faldamis

Cavium Inc.

Weiwei Jiang

Dept. of Computer Science
Columbia University

Gennette Gill

D.E. Shaw Research

Steven M. Nowick

Dept. of Computer Science
Columbia University

ACM/IEEE Asia and South Pacific Design Automation Conf. (ASP-DAC 14)

Motivation for Networks-on-Chip

- Future of computing is multi-core
  - 2 to 4 cores are common, 8 to 16 widely available

    e.g. Niagara 16-core, Intel 10-core Xeon, AMD 16-core Opteron
  - Expected progression: hundreds or thousands of cores
  - Trend towards complex systems-on-chip (SoC)
- Communication complexity: new limiting factor
- NoC design enables orthogonalization of concerns:
  - Improves scalability
    - buses and crossbars unable to deliver desired bandwidth
    - global ad-hoc wiring does not scale to large systems
  - Provides flexibility
    - handle pre-scheduled and dynamic traffic
    - route around faulty network nodes
  - Facilitates design reuse
    - standard interfaces increase modularity, decrease design time

Key Active Research Challenges for NoCs

- **Power consumption**
  - Will exceed future power budgets by a factor of 10x
    - [Owens IEEE Micro-07]
  - Global clocks: consume large fraction of overall power
  - Complex clock-gating techniques
    - [Benini et al., TVLSI-02]
- **Chips partitioned into multiple timing domains**
  - Difficult to integrate heterogeneous modules
  - Dynamic voltage/frequency scaling (DVFS) for lower power
    - [Ogras/Marculescu DAC-08]
- **A key performance bottleneck = latency**
  - Latency critical for on-chip memory access
  - Important for chip multiprocessors (CMP's)

Potential Advantages of Asynchronous Design

➢ Lower power

- No clock power consumed
- Idle components consume no dynamic power
  - IBM/Columbia FIR filter [Tierno, Singh, Nowick, et al., ISSCC-02]

➢ Greater flexibility/modularity

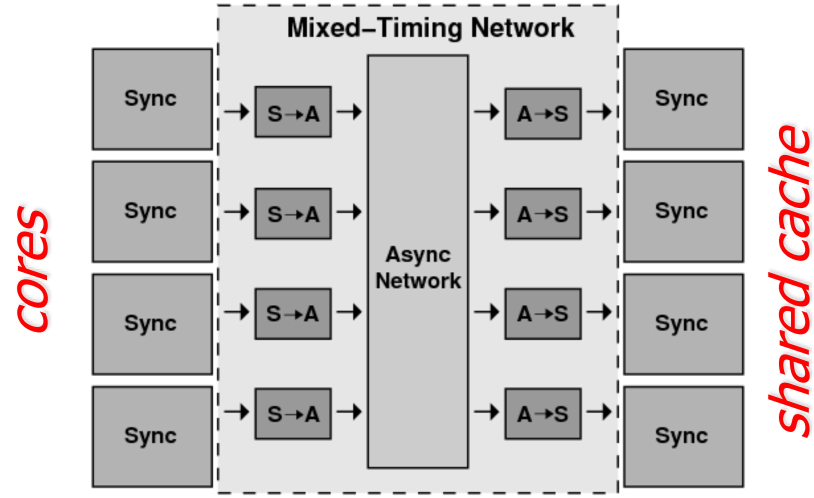
- Easier integration between multiple timing domains
- Supports reusable components
  - [Bainbridge/Furber, IEEE Micro-02 Magazine]
  - [Dobkin/Ginosar, Async-04]

➢ Lower system latency

- No per-router clock synchronization ➡ no waiting for clock
  - [Sheibanyrad/Greiner et al., IEEE Design & Test '08]
  - [Horak, Nowick, et al., NOCS-10]

Motivation for Our Research

- Target = interconnection network for CMP's
  - Network between processors and cache memory
  - GALS NoC: sync/async interfaces + async network
- Requires high performance
  - Low system-level latency
    - Lightweight routers for low-latency
  - High sustained throughput
    - Maximize steady-state throughput
- Target topology = variant MoT ("Mesh-of-Trees")



  - Tree topologies becoming widely used for CMP's:
    - XMT [Balkan/Vishkin et al., Hot Interconnects-07]
    - Single-cycle network [Rahimi, Benini, et al., DATE-11]
    - NOC-OUT [Grot, Falsafi, et al., IEEE Micro-12]
- Our two main contributions:
  - High-performance async network with advance arbitration
  - Detailed comparative evaluation on 8 benchmarks

Contributions (1)

- Mesh-of-Trees (MoT) network with "early arbitration"
  - Target system-latency bottleneck
  - Observe newly-entering traffic
  - Perform early arbitration + channel pre-allocation
  - ❖ Net benefit: bypass arbitration logic + pre-opened channel
- "Early arbitration" capability in fan-in router nodes
  - Simple and fast ➡ operate as FIFO in many traffic scenarios
- Monitoring network:
  - Rapid advance notification of incoming data
  - Fast and lightweight
  - Key component for early arbitration

Contributions (2)

➢ Detailed experimentation and analysis

- "Early arbitration" network vs. "baseline" and "predictive"
  - "baseline": [Horak/Nowick, NOCS-10]
  - "predictive": [Gill/Nowick, NOCS-11]
- 8 diverse synthetic benchmarks
  - represent different network conditions
- Significant latency improvement and comparable throughput
  - New vs. baseline: 23-30% latency improvement
  - New vs. predictive: 13-38% latency improvement
- Low end-to-end system latency
  - ~1.7ns (at 25% load, 90nm): through 6 router nodes + 5 hops

Related Work: NoC Acceleration Techniques

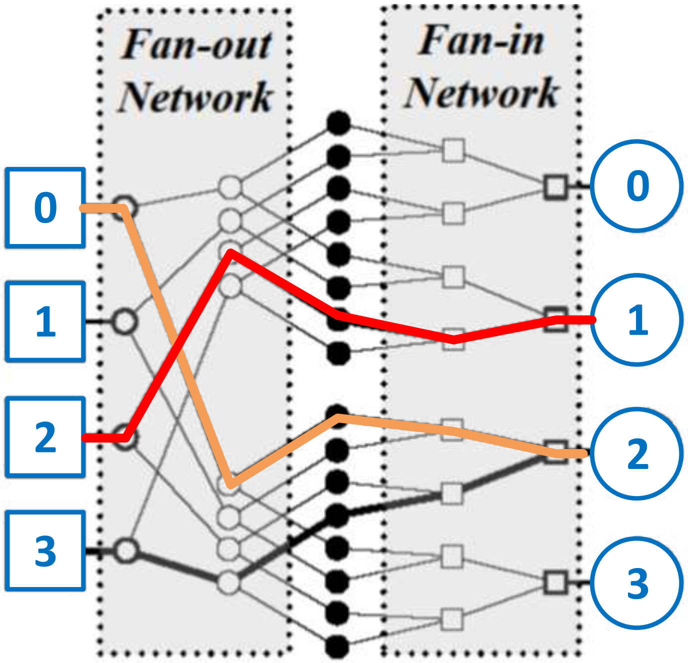
- **Express virtual channels** [Kumar/Peh, ISCA-07]
  - Selective packets use dedicated fast channels
  - Virtually bypass intermediate nodes
    - ⇨ improvements only against slow coarse-grained baseline: 3-cycle operation
- **SMART NoC** [Chen/Peh, DATE-13]
  - Selective packets traverse multiple hops in one cycle
    - ⇨ requires advanced circuit-level techniques + aggressive timing assumptions
- **Hybrid network** [Modarressi/Arjomand, DATE-09]
  - A normal packet-switched network + fast circuit-switched network
  - Flits can switch between two sub-networks
    - ⇨ requires partitioned network (statically-allocated) + large circuit-switched setup time
- **NoC using "advanced bundles"** [Kumar et al., ICCD-07]
  - Provides advanced information of flit arrival
  - Closer to our approach
    - ⇨ "advance bundles" advance only one cycle per hop (unlike our approach)

Outline

- Introduction
- Background
- New Asynchronous MoT Network
  - Overview of the “Early Arbitration” Approach
  - Monitoring Network
  - Design of the New Arbitration Node
- Experimental Results
  - Simulation Setup
  - Network-Level Results
- Conclusion and Future Work

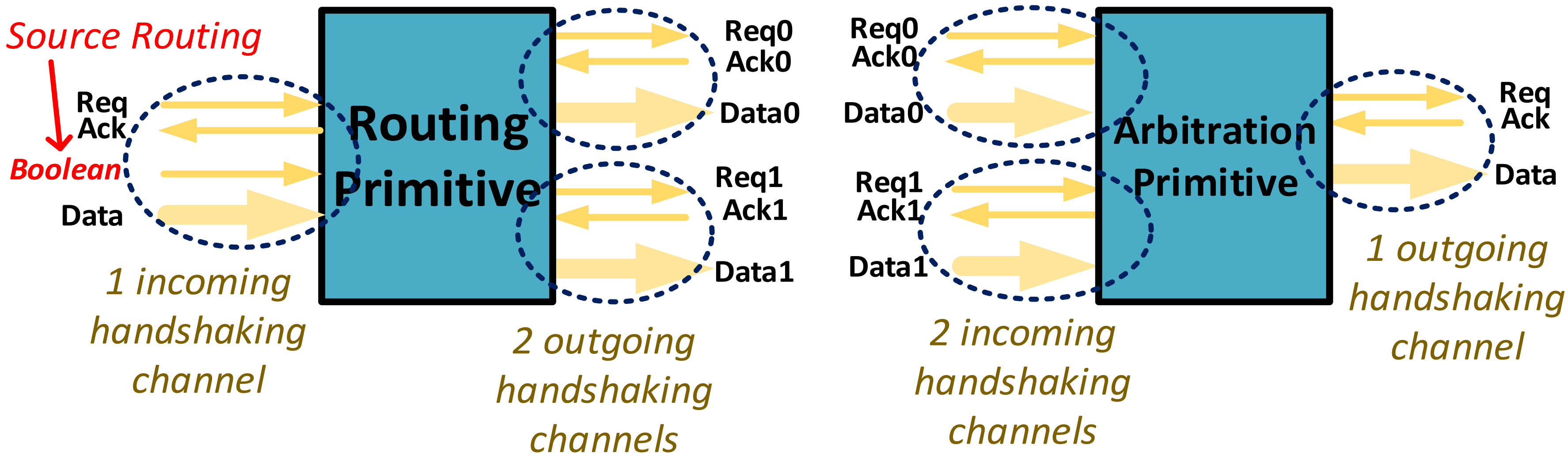
Background: Mesh-of-Trees (MoT) Variant

- **Topology basics**
  - Fan-out and fan-in network
    ➡ "inverse" of classical MoT (Leighton)
  - Two node types
    Routing: 1 input and 2 output channels
    Arbitration: 2 input and 1 output channels



- **Routing features**
  - Deterministic wormhole routing
    Path examples shown in the figure
  - No contention between distinct source/sink pairs
- **Potential performance benefits**
  - Lower latency and higher throughput over 2D-mesh
  - Shown to perform well for CMP's
    [Balkan/Vishkin, Trans. VLSI, Oct. 09], [Balkan/Vishkin, Hot Interconnects-07]

Background: Two Node Types

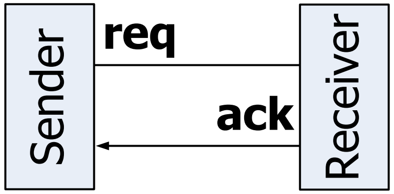


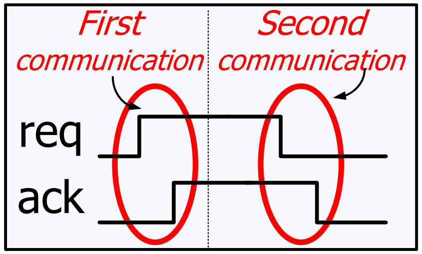
- **Routing primitive**
  - 1 input channel and 2 output handshaking channels
  - Route the input to one of the outputs
- **Arbitration primitive**
  - 2 input and 1 output handshaking channels
  - Merge two input streams into one output stream

Background: Asynchronous Protocols

➢ Handshaking: transition signaling (two-phase)

- Two events per transaction
  - Req/Ack toggle
- Merits over level signaling (four-phase):
  - 1 roundtrip communication per data item
  - High throughput and low power
- Challenge of two-phase signaling:
  - designing lightweight implementations





➢ Data encoding: single-rail bundled data

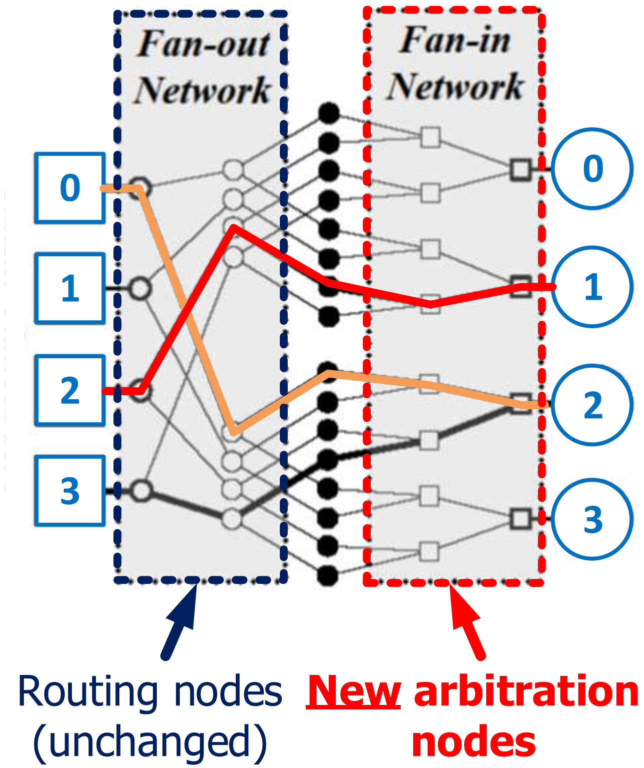
- Standard synchronous single-rail data + extra "bundling" req
- Merits of single-rail bundled data:
  - *low power* and *very good coding efficiency*
  - allow to re-use synchronous components
- Challenge: requires matched delay for "bundling req"
  - one-sided timing constraint: "request" must arrive after data is stable

Outline

- Introduction
- Background
- New Asynchronous MoT Network
  - Overview of the "Early Arbitration" Approach
  - Monitoring Network
  - Design of the New Arbitration Node
- Experimental Results
  - Simulation Setup
  - Network-Level Results
- Conclusion and Future Work

Overview: Early Arbitration Strategy

- **Key network bottleneck**
  - System-latency
    - bottleneck of arbitration logic in fan-in nodes
- **Basic strategy = anticipation**
  - Observe newly-entering traffic
  - Do early arbitration + channel pre-allocation
  - ❖ Net benefit: bypass arbitration logic



- **Proposed network**
  - As soon as flit enters network:
    - all downstream nodes quickly notified (by a monitoring network)
    - fan-in nodes: initiate early arbitration + channel pre-allocation
  - When flit arrives at each fan-in node:
    - quickly sent out through pre-allocated channel

Outline

- Introduction
- Background
- New Asynchronous MoT Network
  - Overview of the "Early Arbitration" Approach
  - Monitoring Network
  - Design of the New Arbitration Node
- Experimental Results
  - Simulation Setup
  - Network-Level Results
- Conclusion and Future Work

Monitoring Network: Overview

- **Purpose:** **rapid advance notification** of incoming data

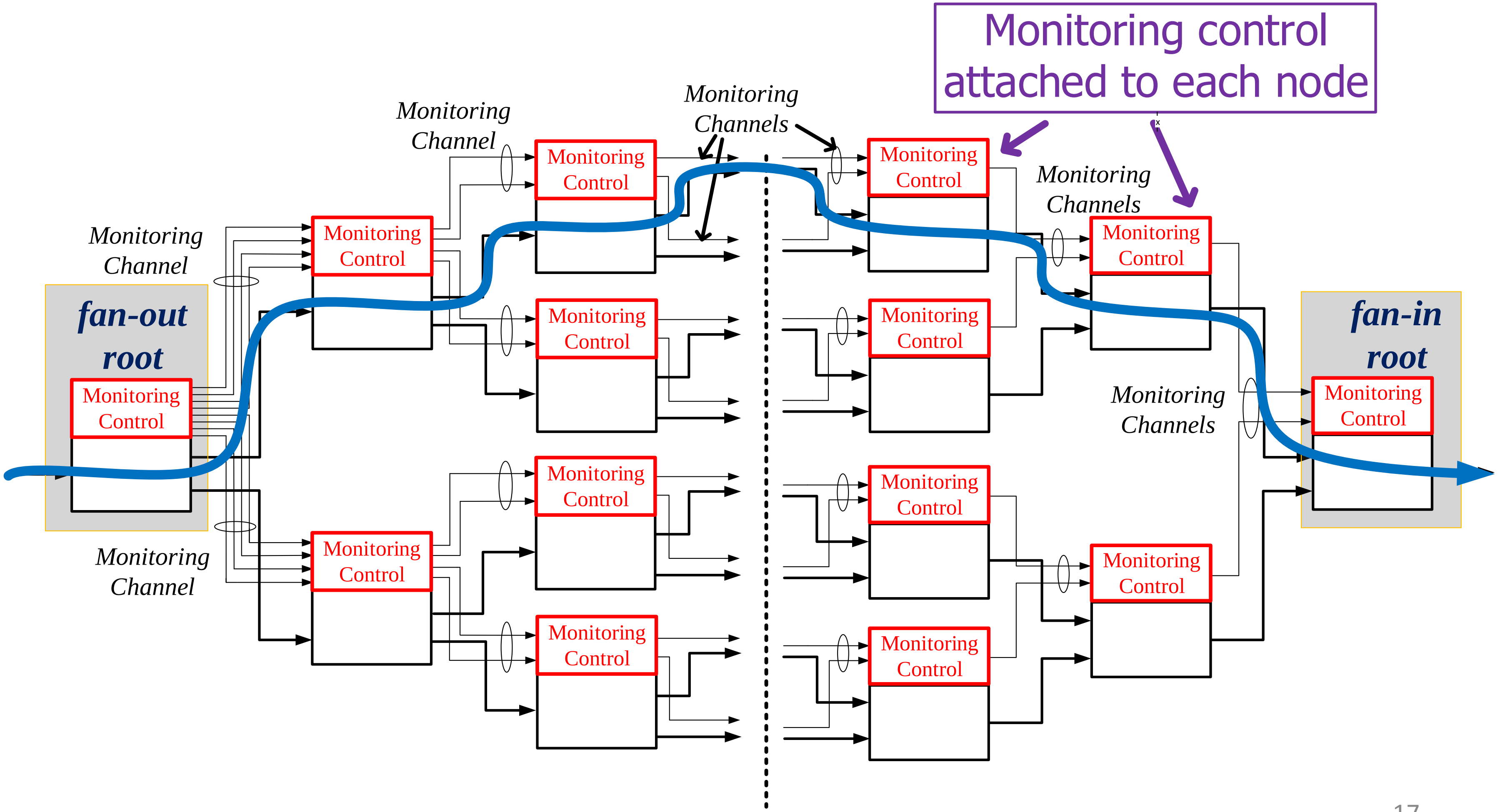
- **Structure:** **lightweight shadow replica** of MoT network
  - Small *monitoring control unit* attached to *each node*
    - i.e. both routing and arbitration

- **Fast and lightweight**
  - Implemented by **several gates** for each control unit

- **Different role for fan-out and fan-in monitoring**
  - Fan-out: fast forward early notification without using it
  - Fan-in: fast forward and use it for early arbitration

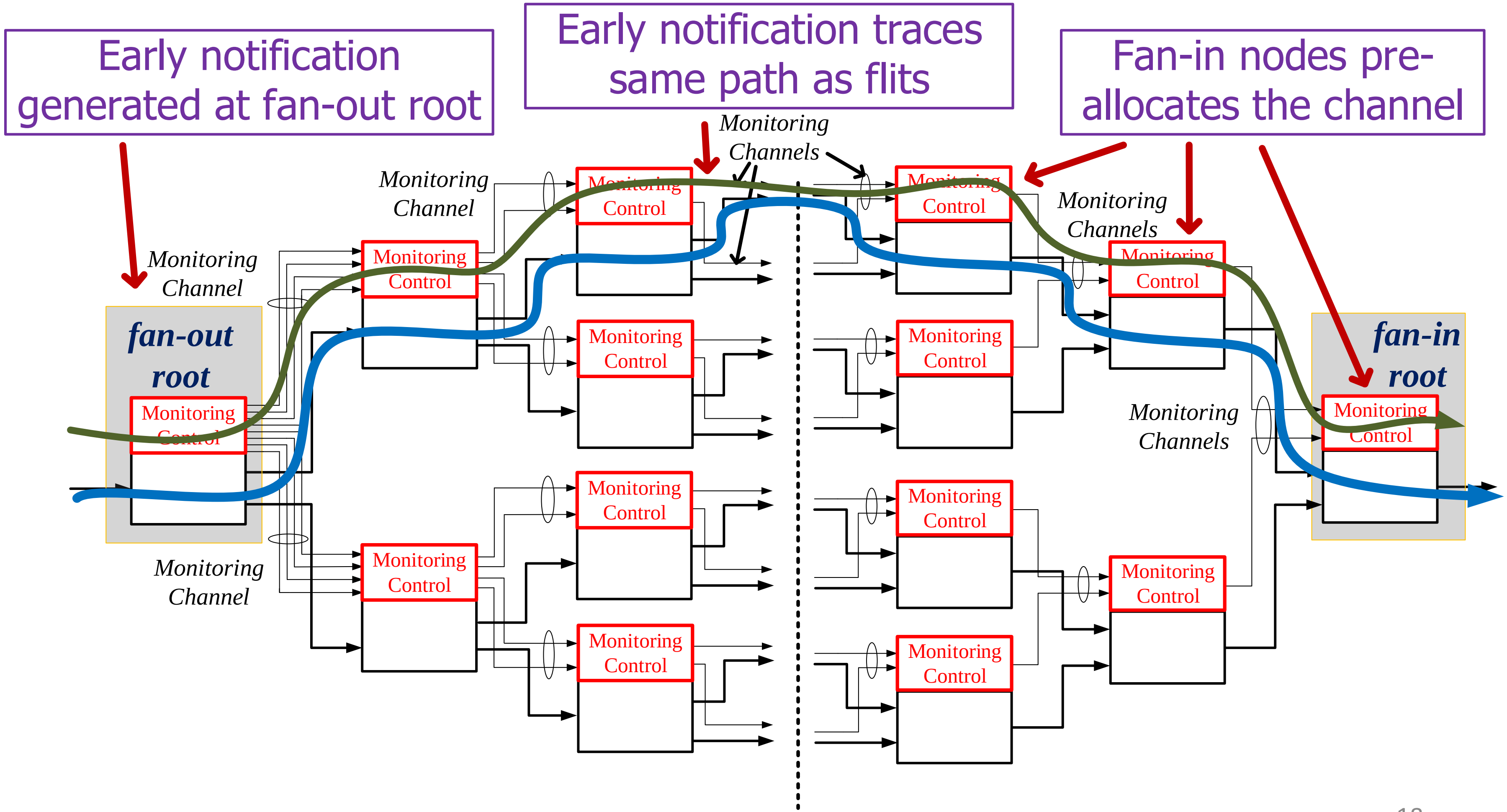
Monitoring Network: Structure

- **Structure:** a shadow replica of MoT network
  - Small and fast monitoring control unit attached for each node



Monitoring Network: Operation

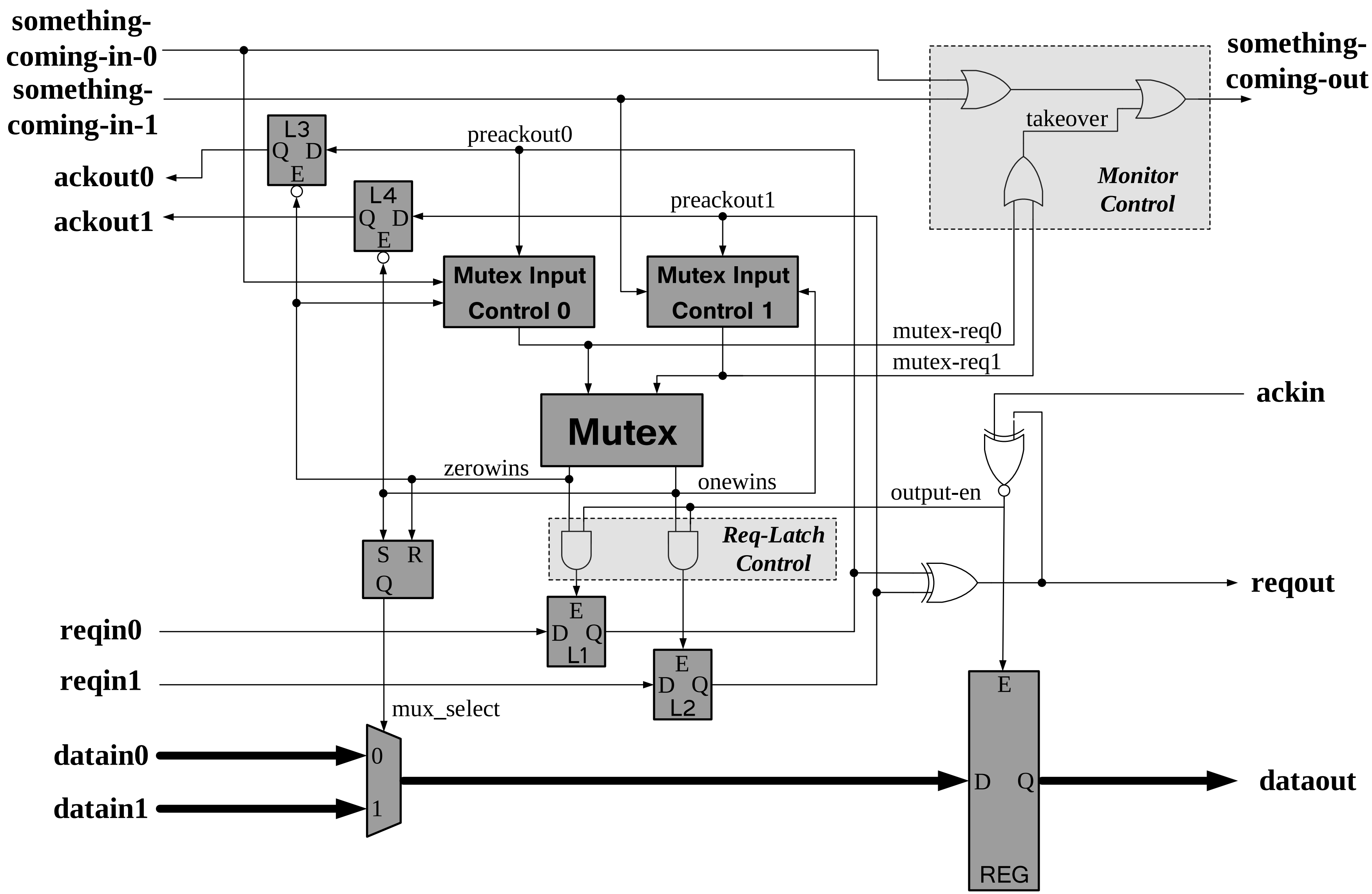
- When a flit enters the network
  - Early notification generated and fast forwarded



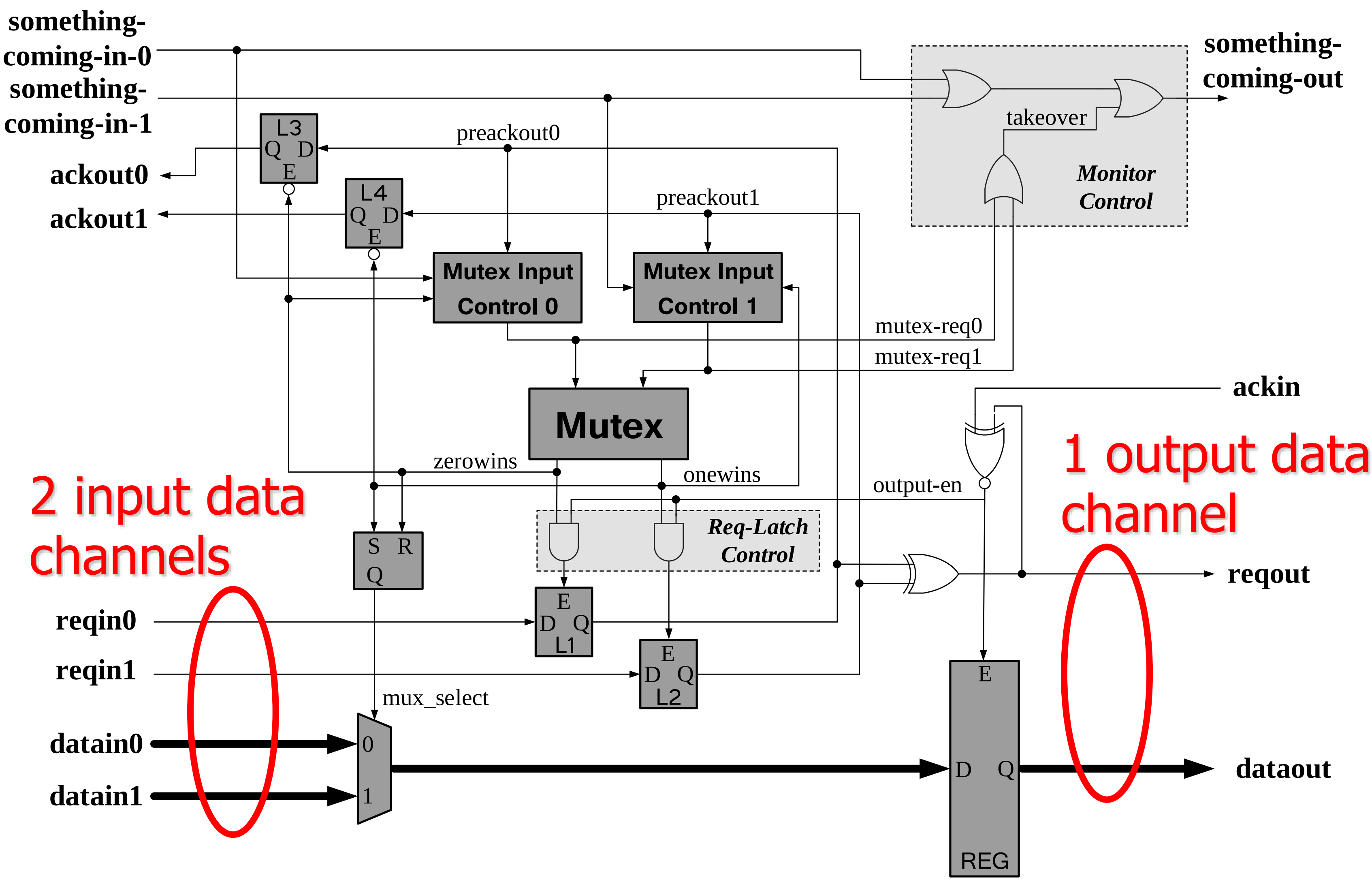
Outline

- Introduction
- Background
- New Asynchronous MoT Network
  - Overview of the "Early Arbitration" Approach
  - Monitoring Network
  - Design of the New Arbitration Node
- Experimental Results
  - Simulation Setup
  - Network-Level Results
- Conclusion and Future Work

New Arbitration Node: Circuit-Level

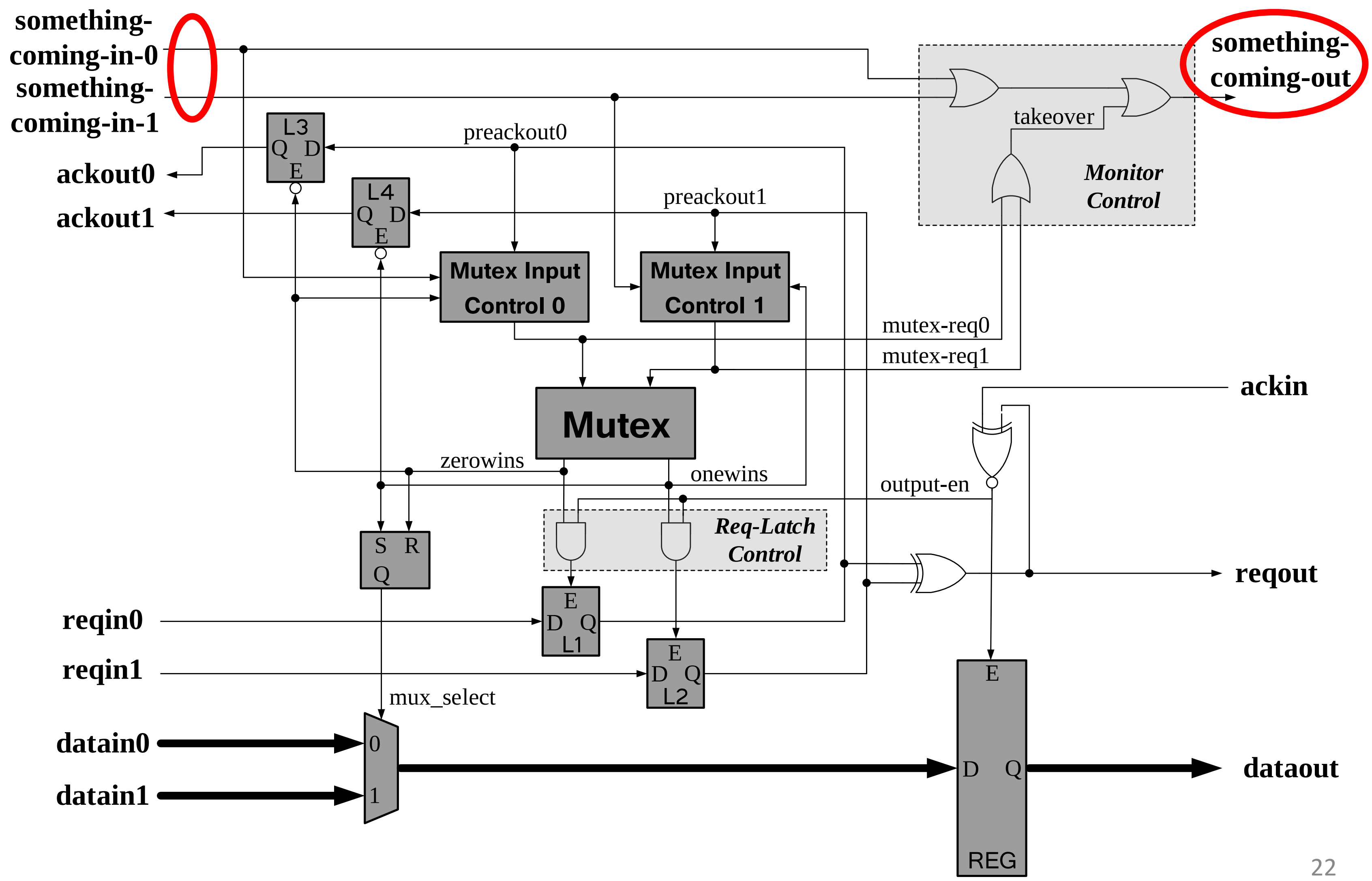


New Arbitration Node: Interfaces



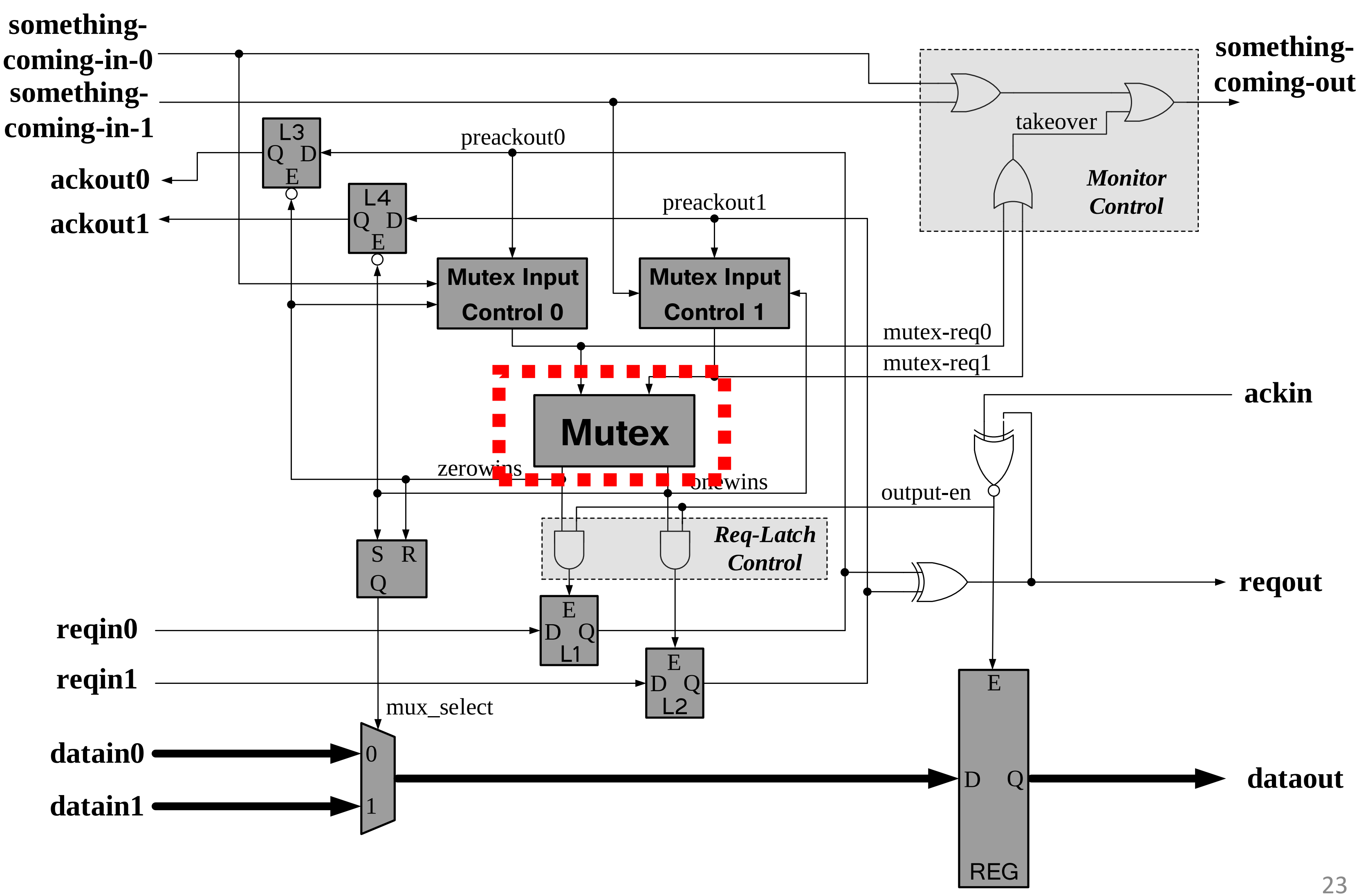
New Arbitration Node: Interfaces (cont.)

Monitoring channels: provide advance info. on incoming traffic



New Arbitration Node: Structure

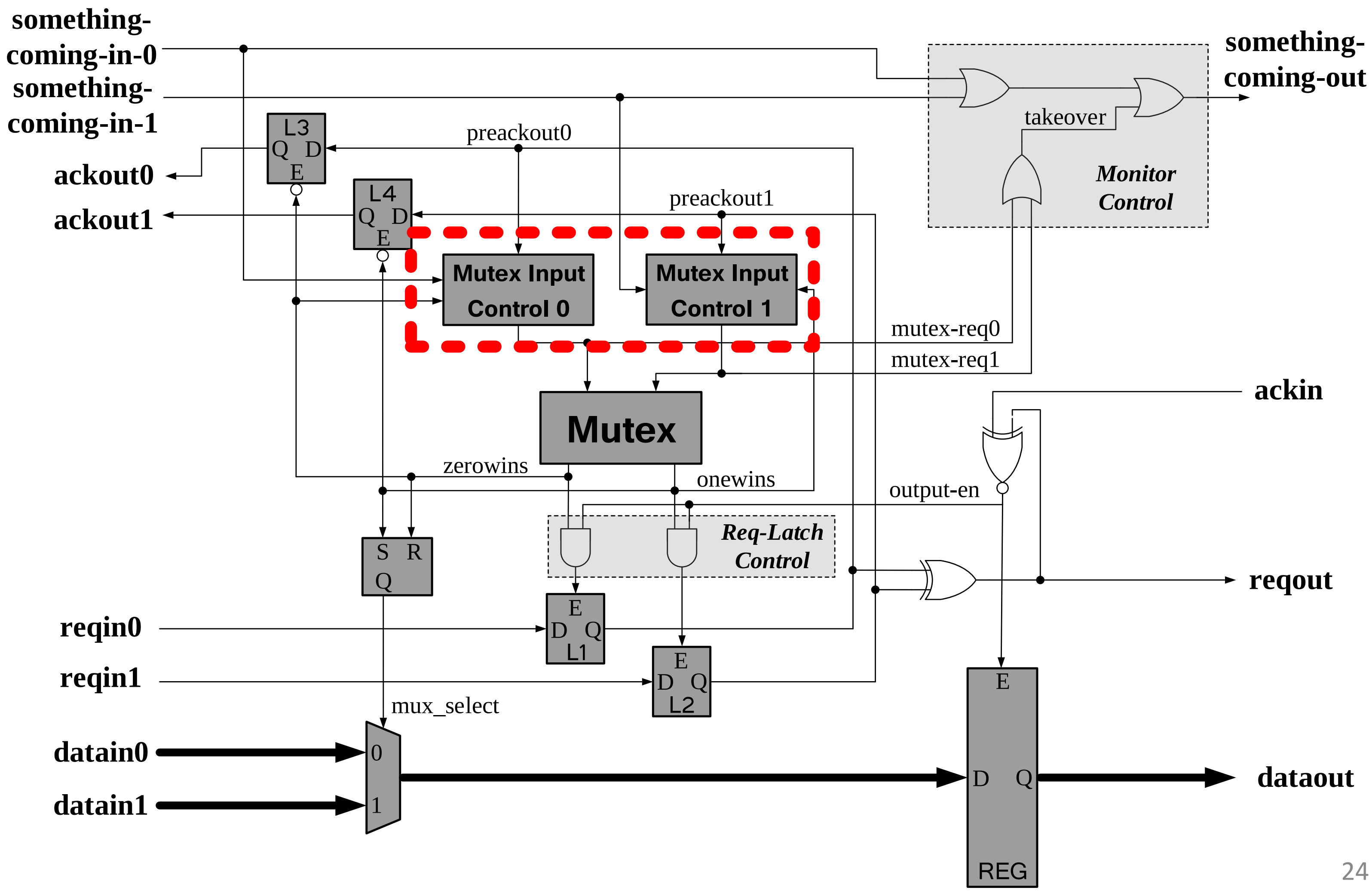
Mutex: resolves arbitration between 2 input channels



New Arbitration Node: Structure (cont.)

Mutex Input Control: requests/releases Mutex

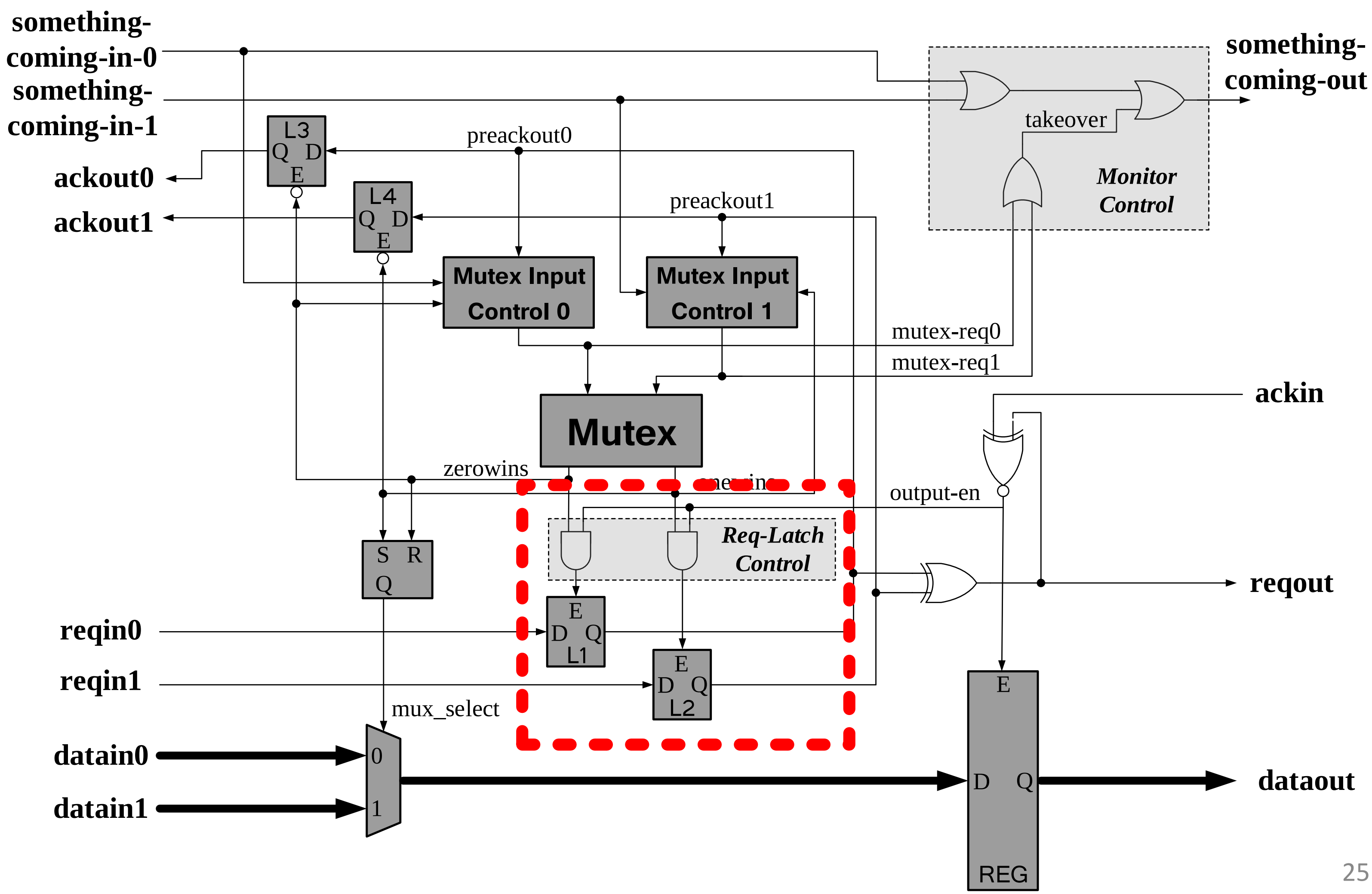
Key component to enable early arbitration



New Arbitration Node: Structure (cont.)

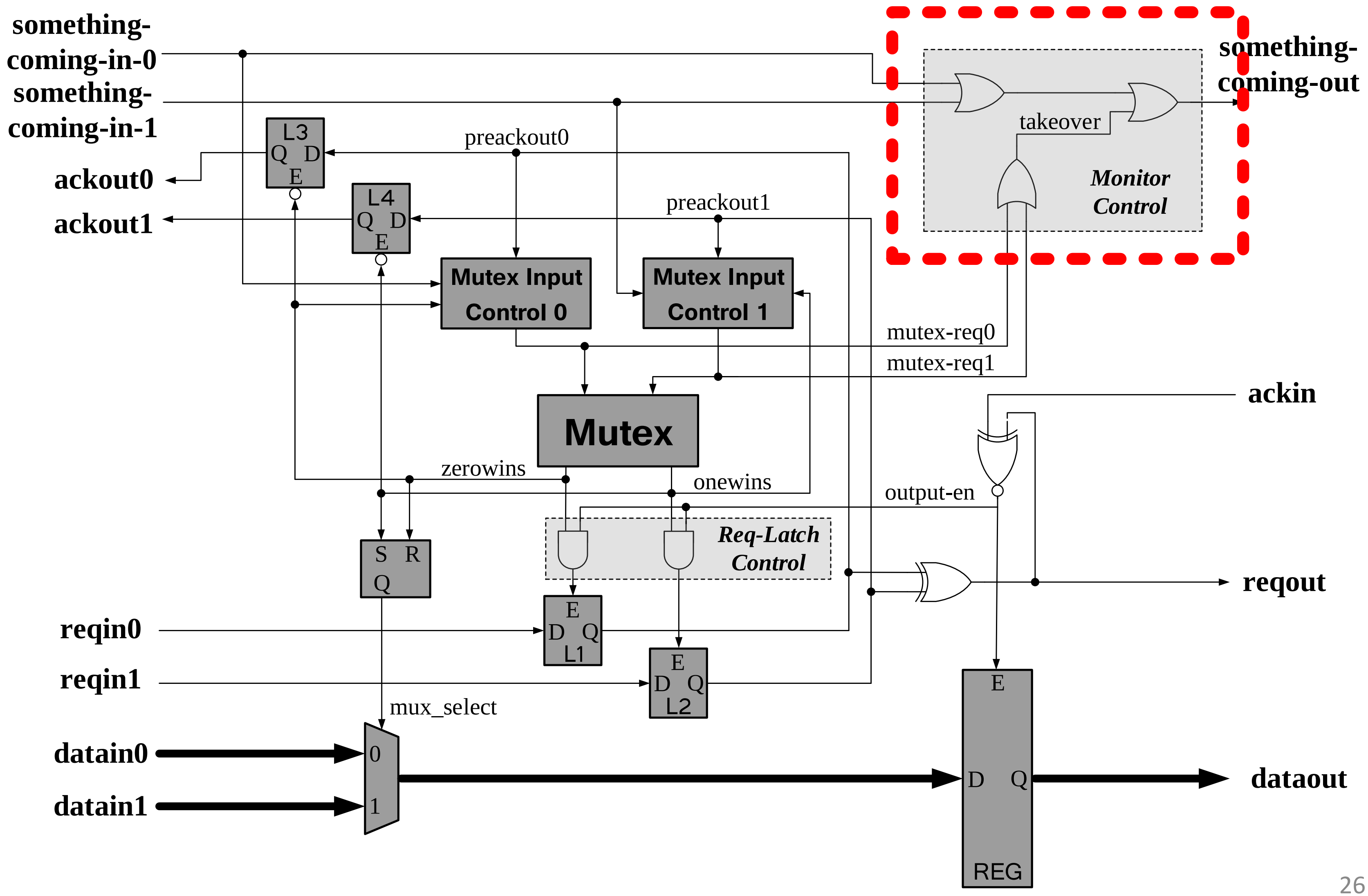
Input channel latch + control:

Two functions: (i) enables channel pre-allocation, (ii) flow control



New Arbitration Node: Structure (cont.)

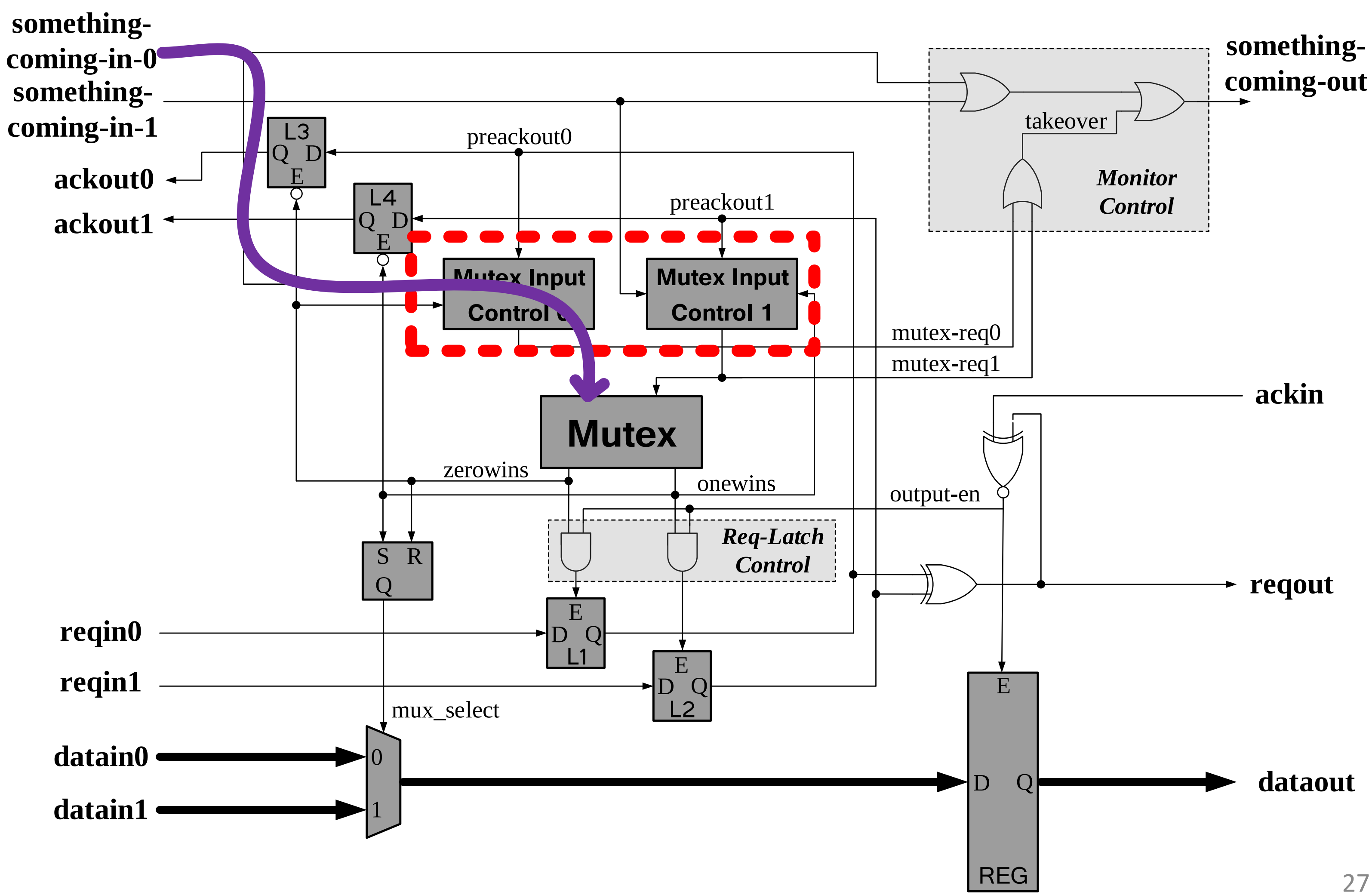
Monitoring control: fast forwards early notification



New Arbitration Node: Key Feature (1)

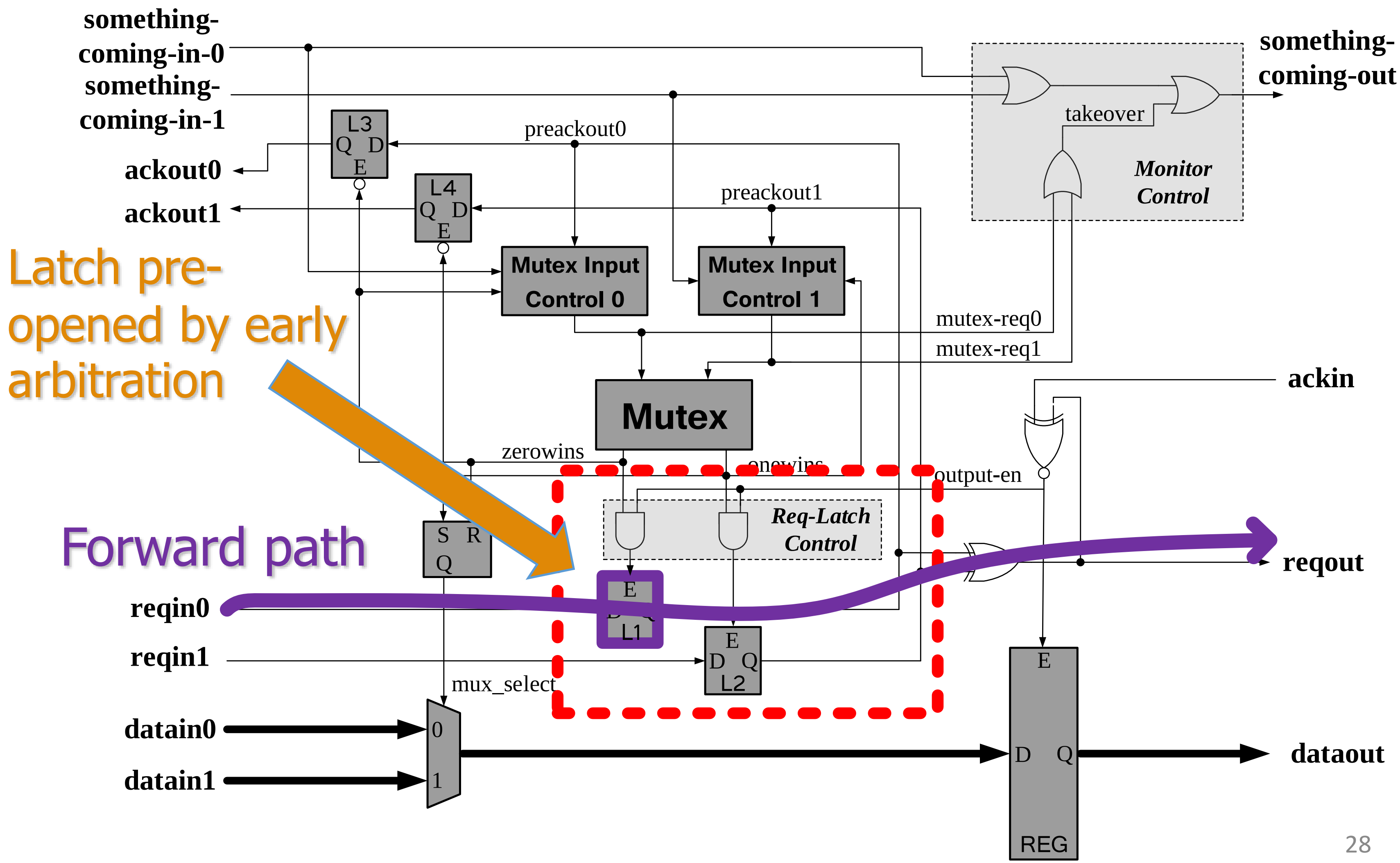
Early arbitration capability:

Monitoring signals initiate arbitration, before actual flit arrival



New Arbitration Node: Key Feature (2)

Highly optimized forward path:
contains only 1 pre-opened latch = FIFO stage



Simulation: Overview

- Two simulations

  #1. Single-flit scenario

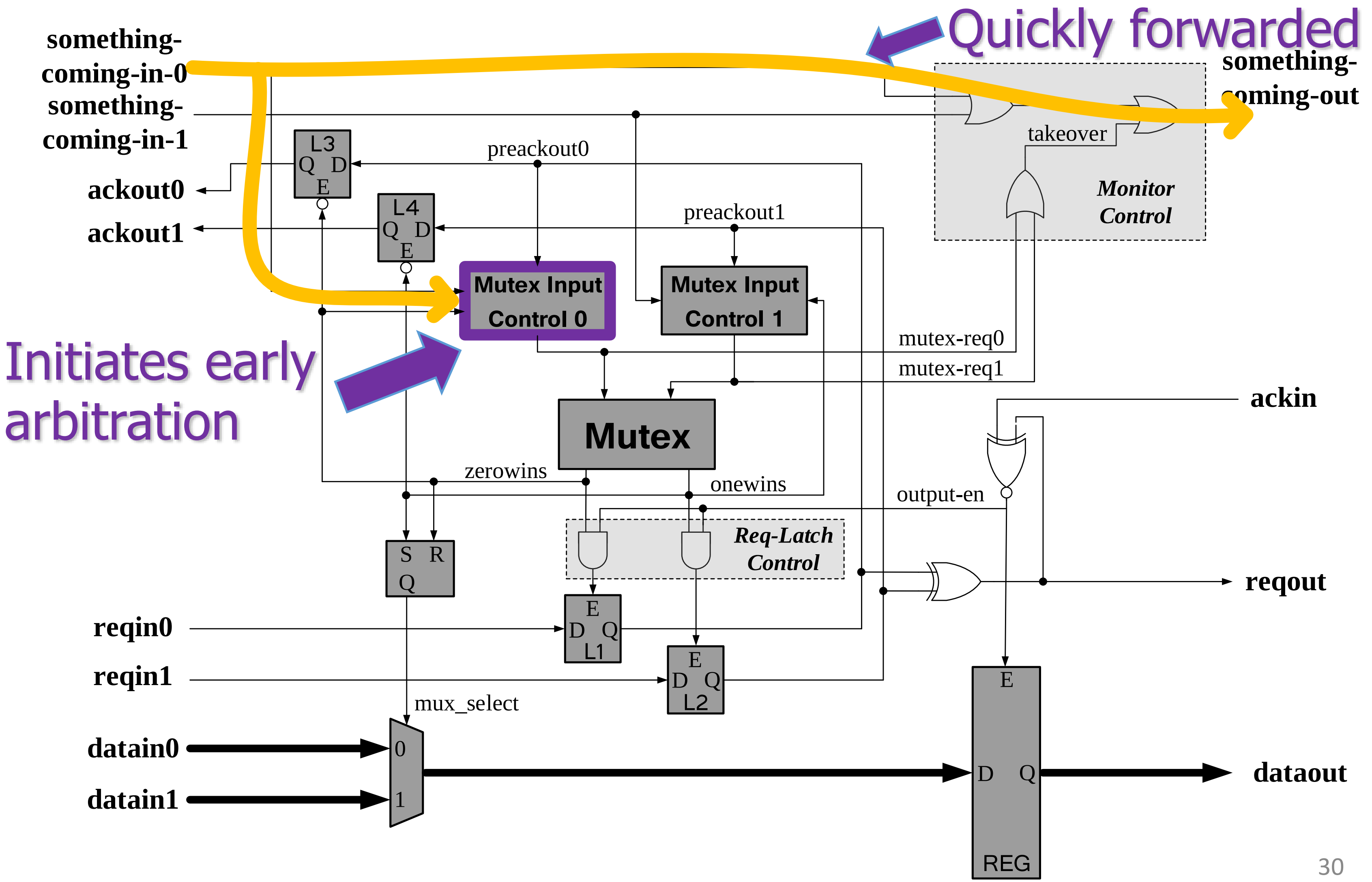
  - friendly case
  - illustrate how early arbitration works

  #2. Contention between two input channels

  - more advanced and adversarial case
  - illustrate how to resolve contention

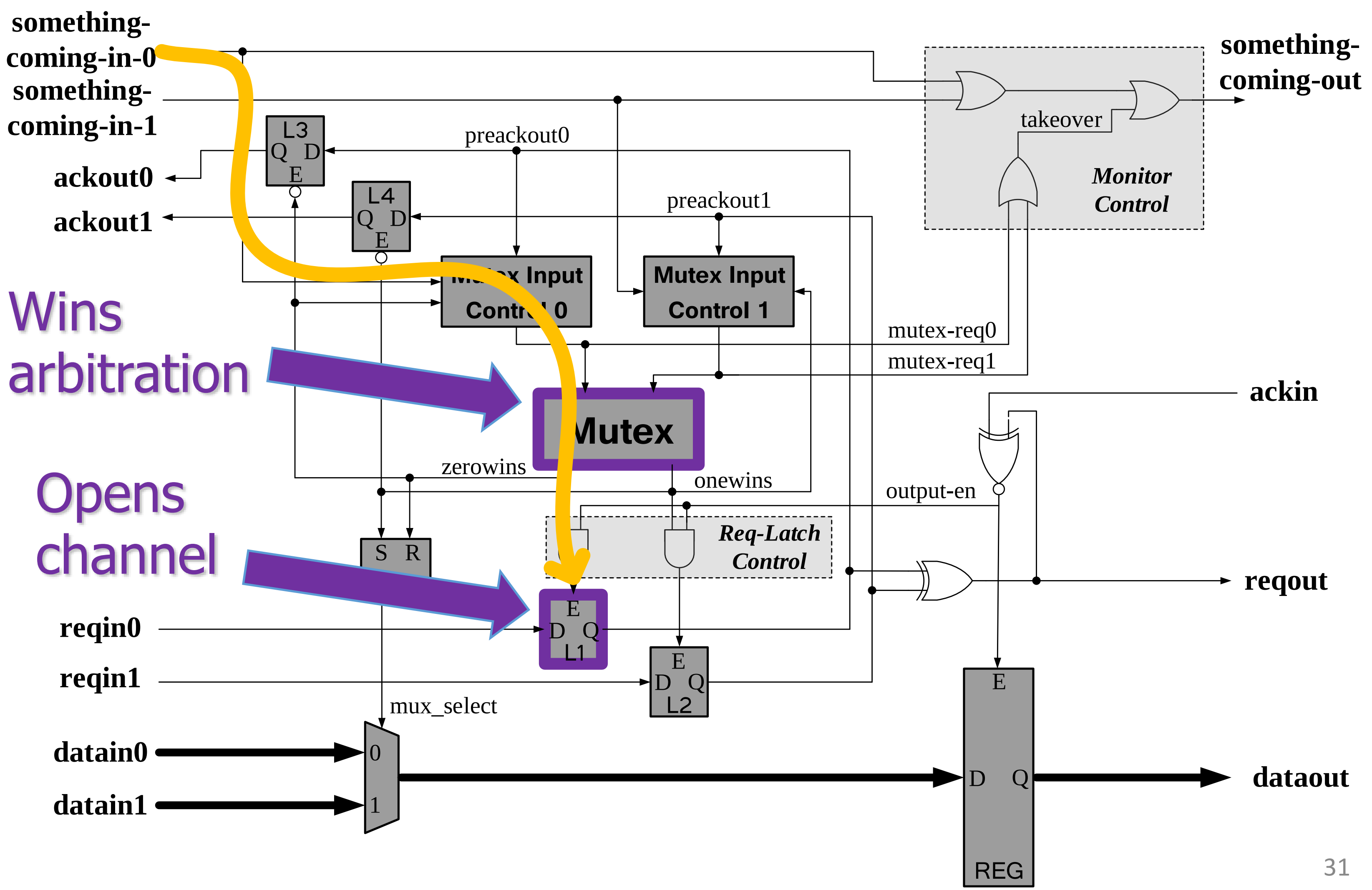
Simulation #1: Single-Flit

Step #1: Monitoring signal arrives (*well before* actual flit)



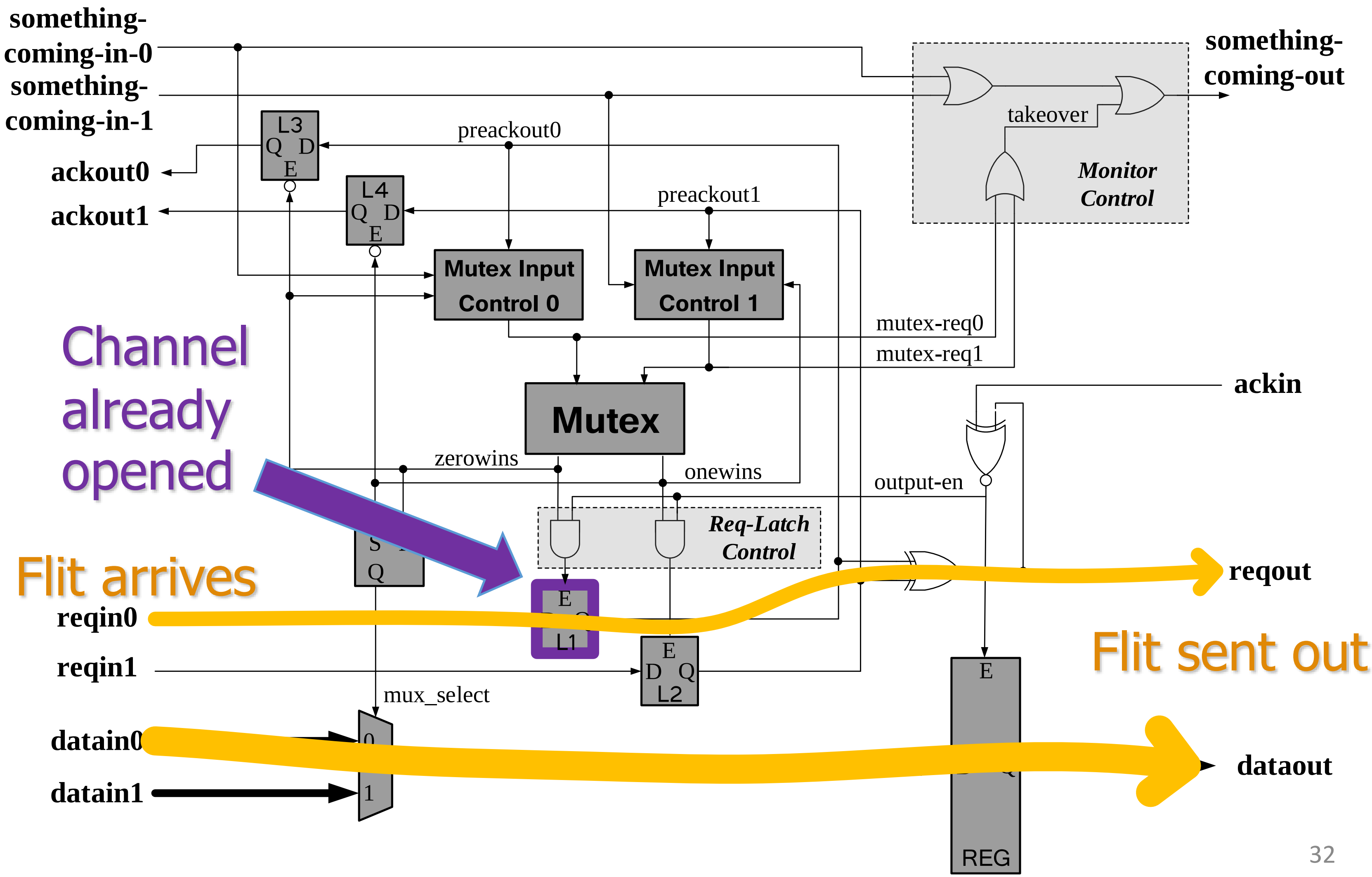
Simulation #1: Single-Flit (cont.)

Step #2: Completes early arbitration

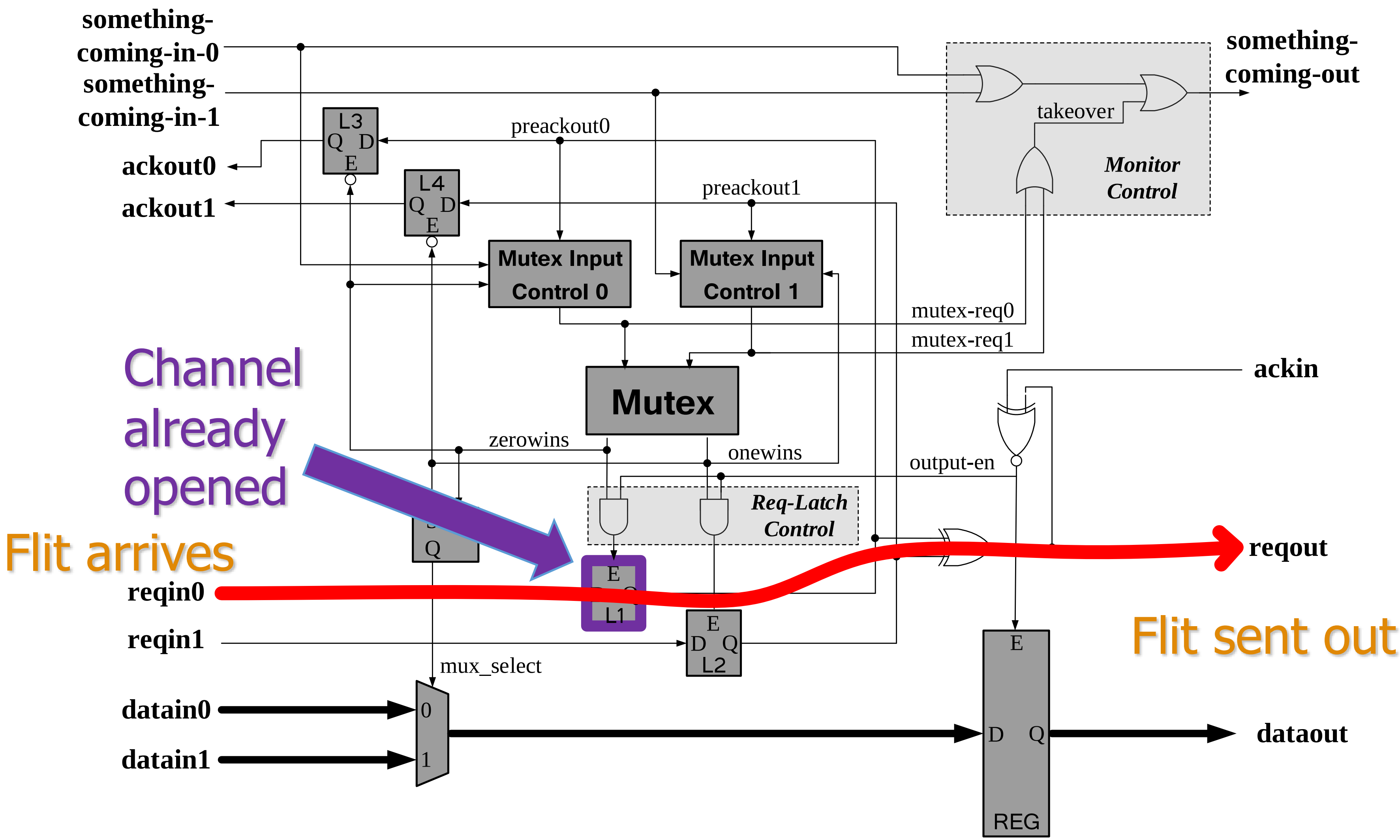


Simulation #1: Single-Flit (cont.)

Step #3: Flit arrives and gets through pre-allocated channel



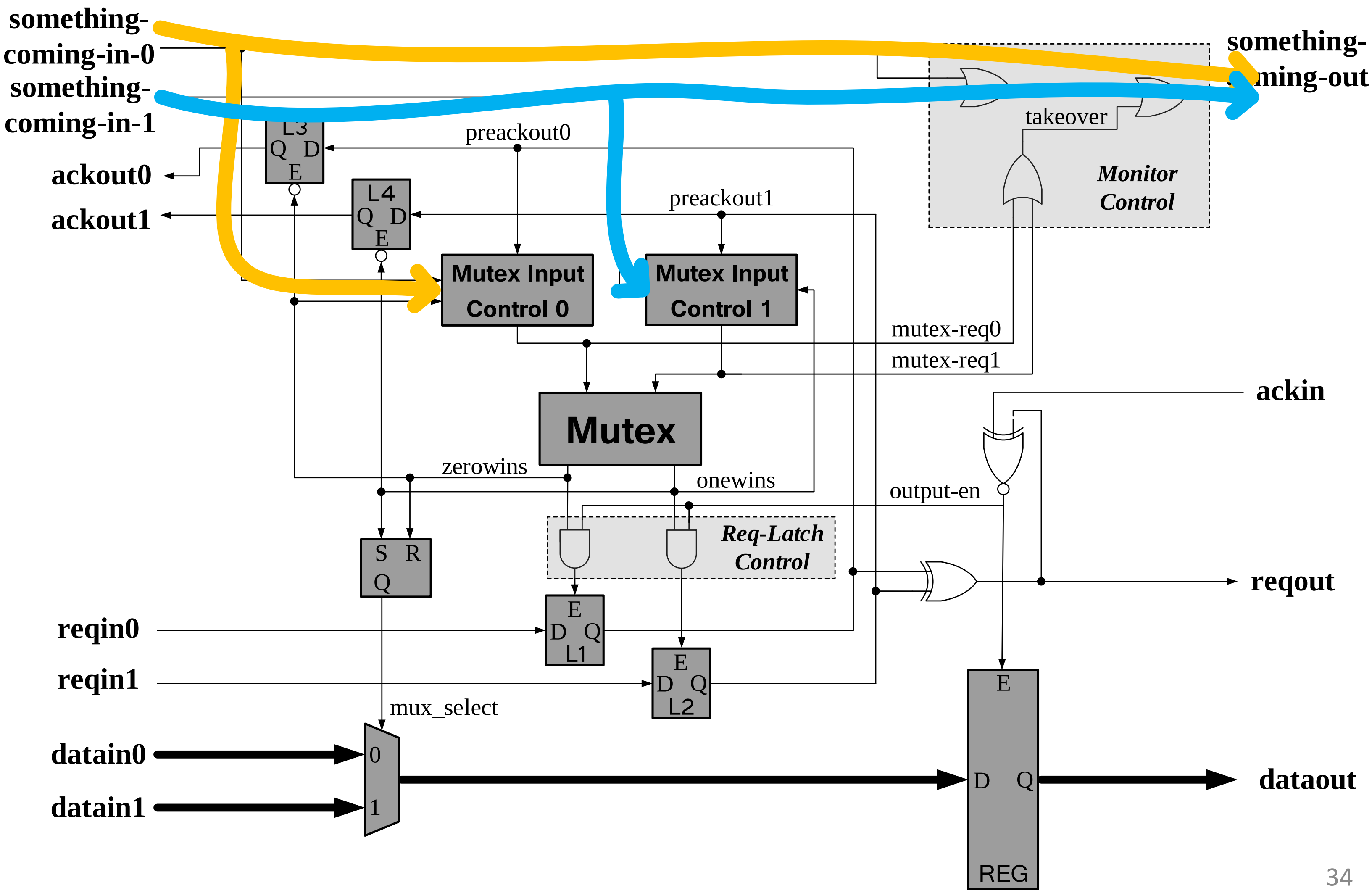
Forward Latency: Single-Flit



Forward latency = D-latch + XOR2 gate

Simulation #2: Contention

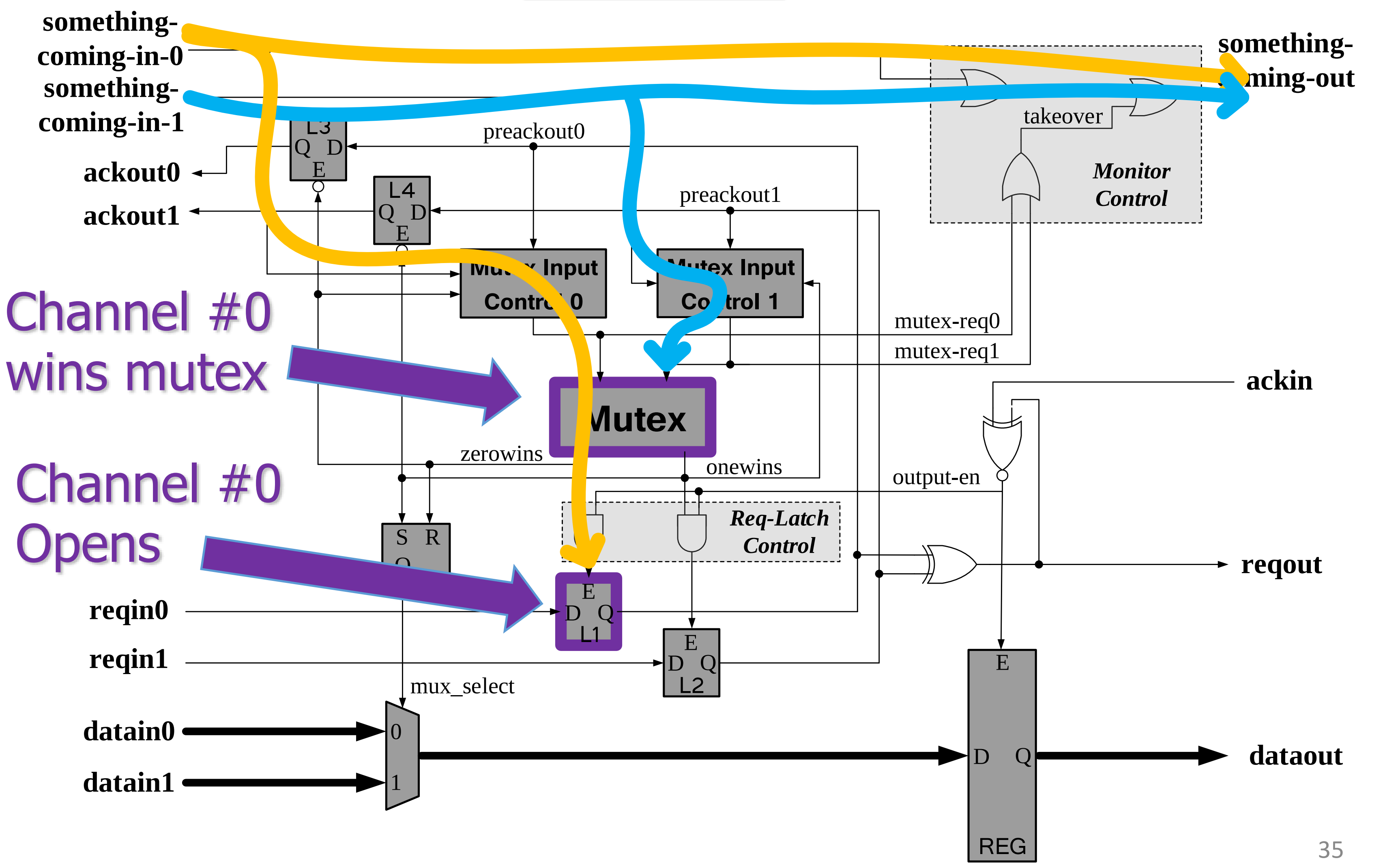
Both monitoring signals arrive **almost simultaneously**



Simulation #2: Contention (cont.)

Both monitoring signals request mutex

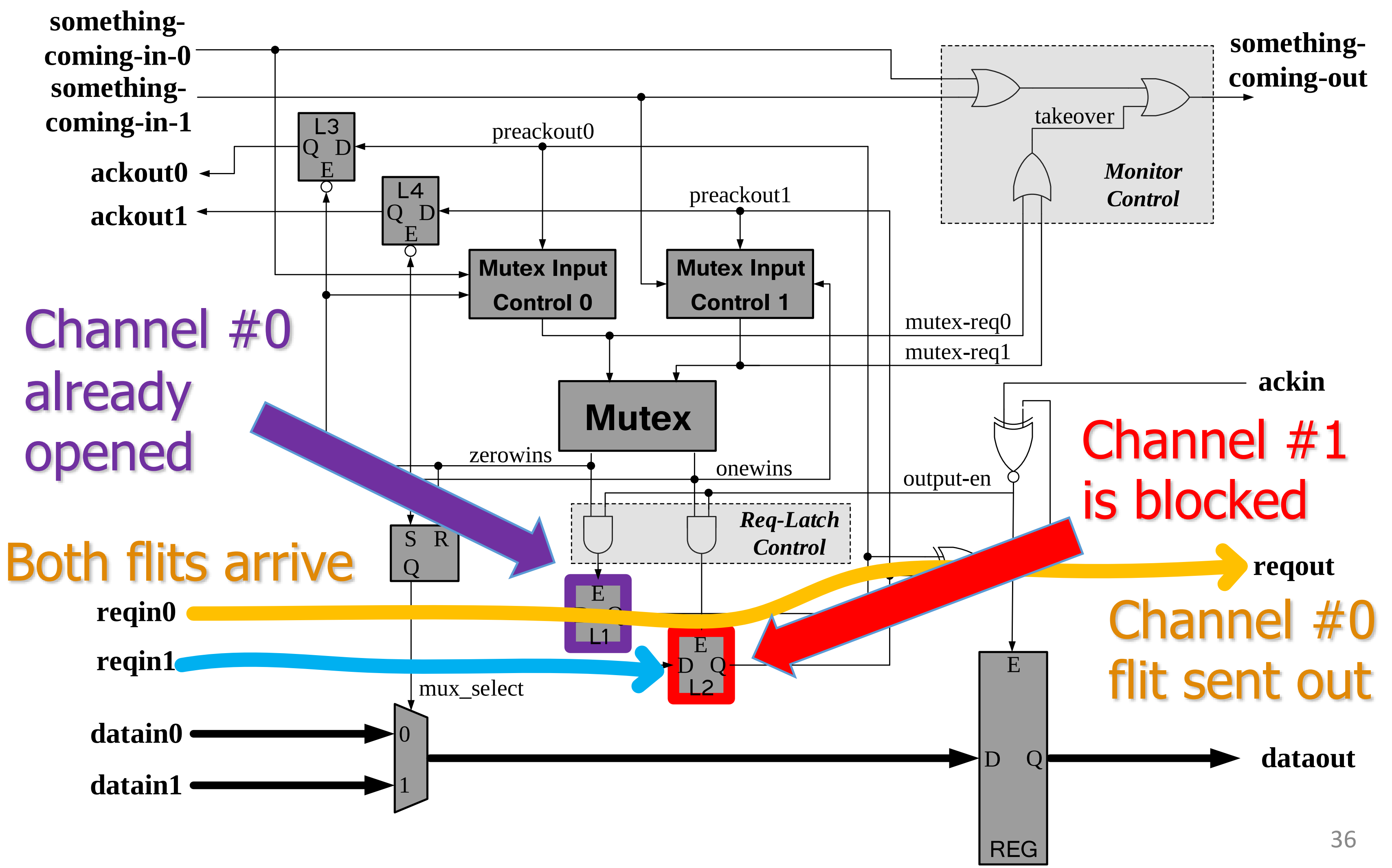
➡ Assume channel #0 wins arbitration



Simulation #2: Contention (cont.)

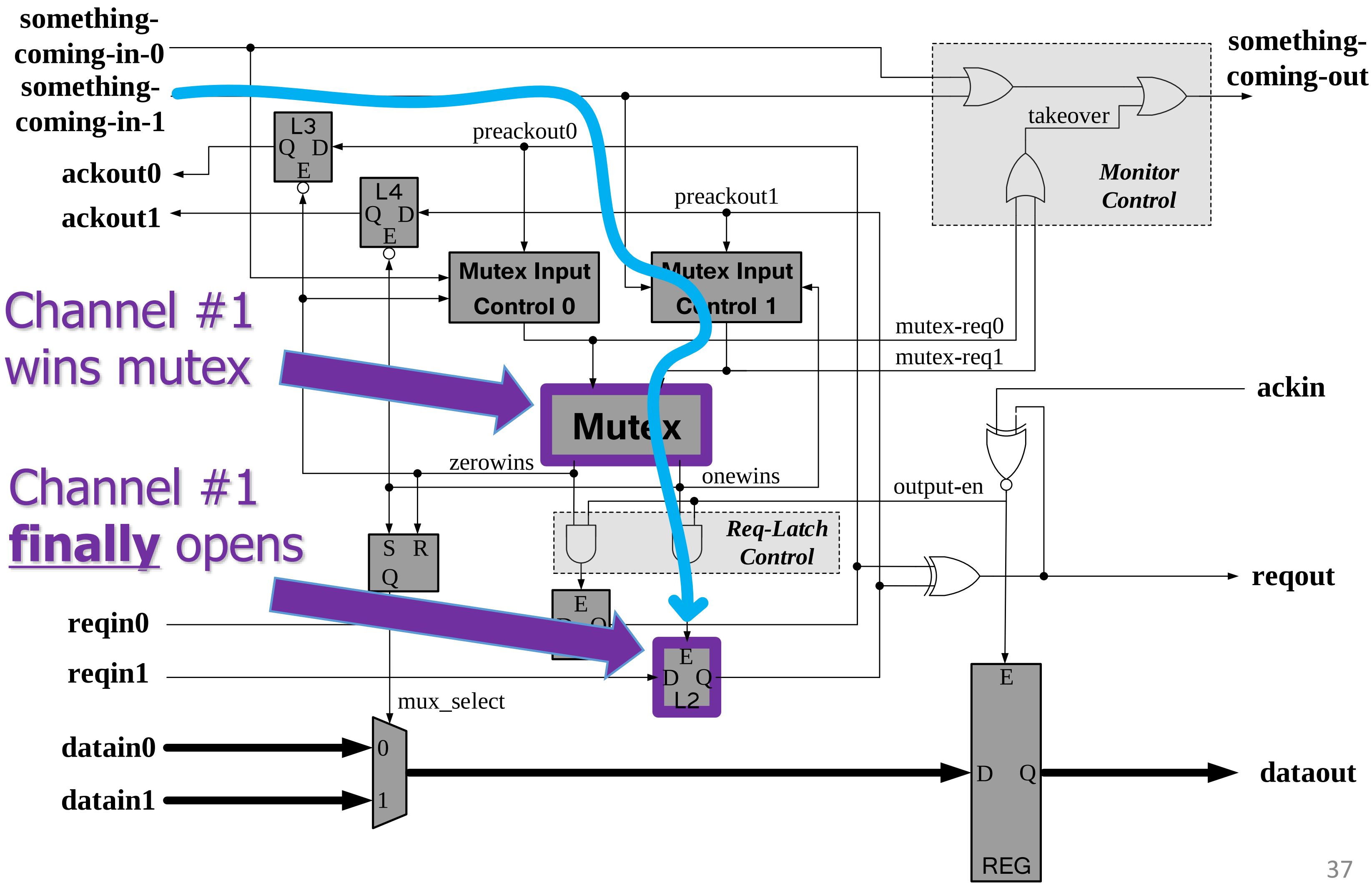
Flit on channel **#0** arrives and goes through pre-allocated channel

Flit on channel **#1** arrives but is blocked



Simulation #2: Contention (cont.)

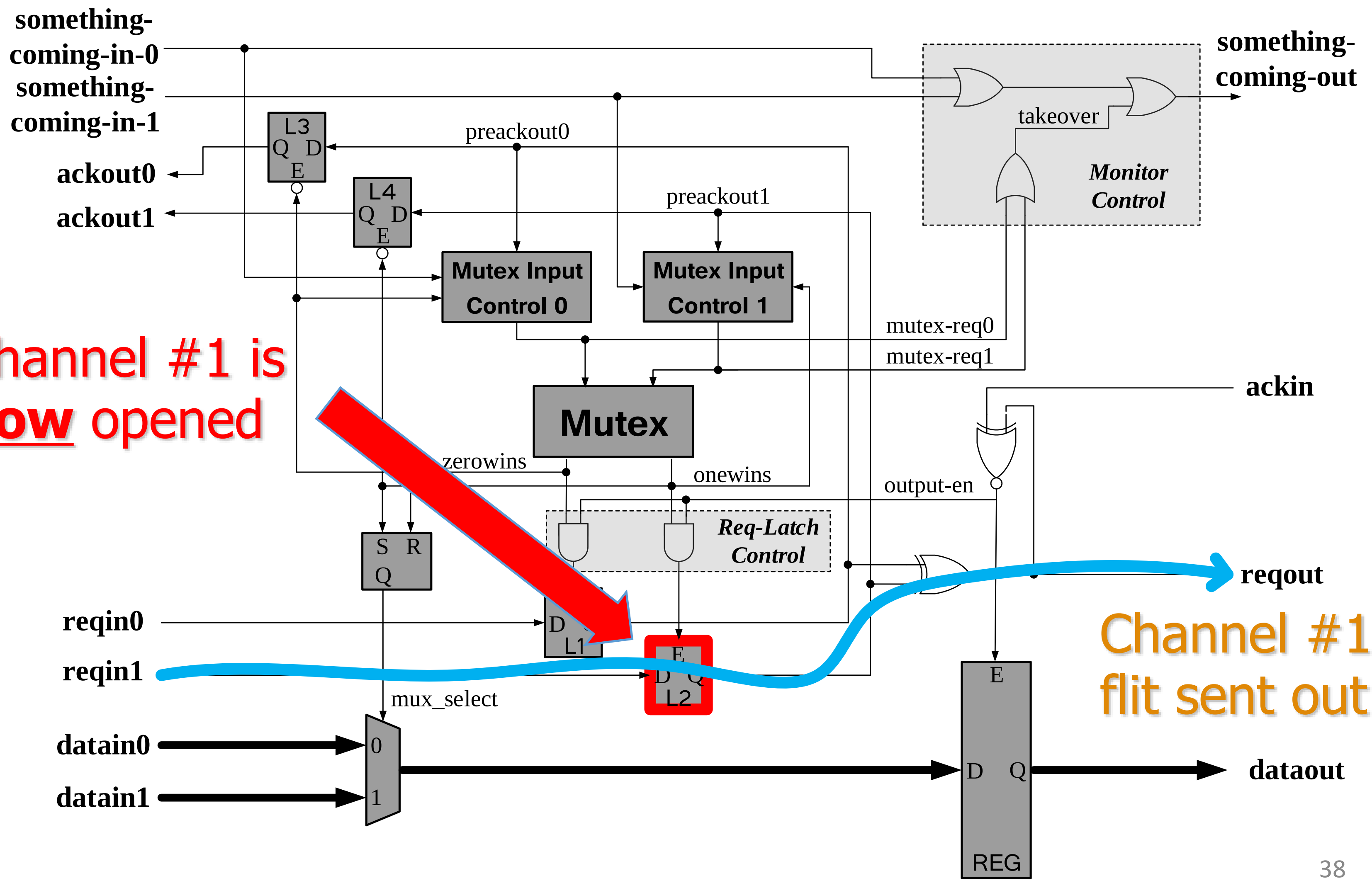
Channel #0 finally releases mutex ➡ channel #1 wins



Simulation #2: Contention (cont.)

Flit on channel **#1** gets through

Channel #1 is **now** opened

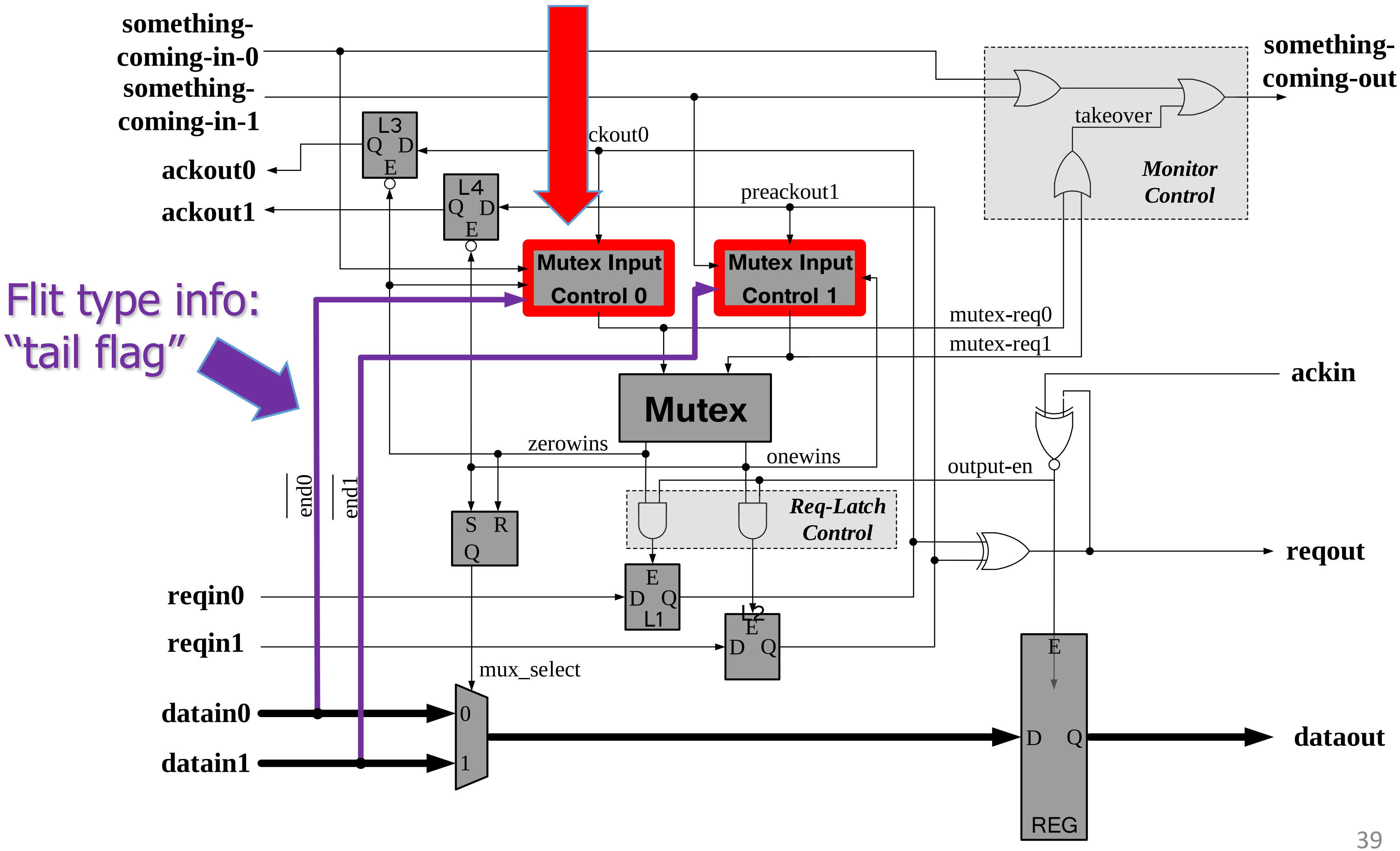


Channel #1 flit sent out

New Arbitration Node: Multi-Flit Design

Structure largely the same as single-flit design

Different **Mutex Input Control**: receives "tail flag"



Outline

- Introduction
- Background
- New Asynchronous MoT Network
  - Overview of the "Early Arbitration" Approach
  - Monitoring Network
  - Design of the New Arbitration Node
- Experimental Results
  - Simulation Setup
  - Network-Level Results
- Conclusion and Future Work

Experimental Results: Overview

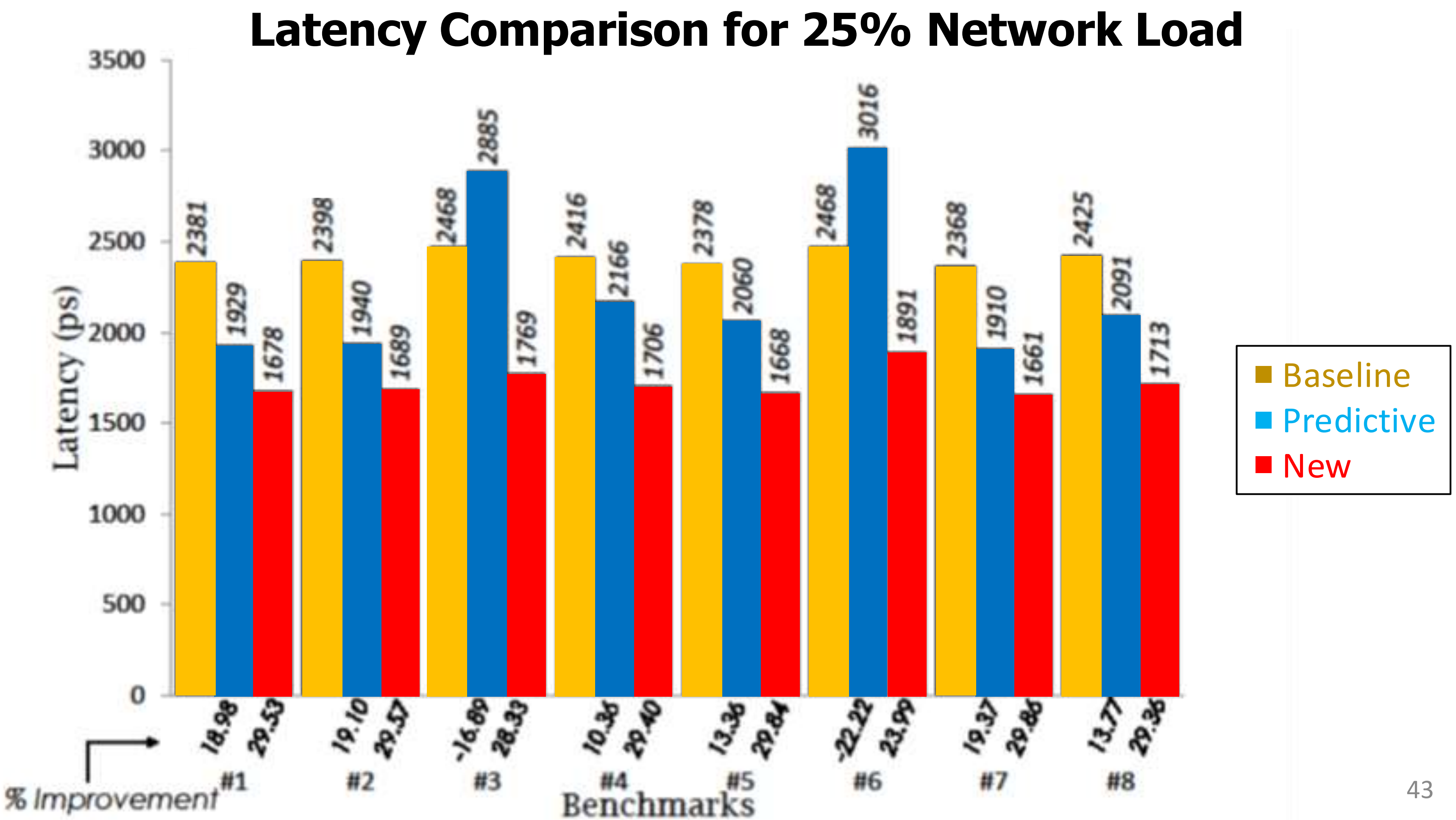
- Two levels of evaluation:
  - Node-level: new arbitration node <u>in isolation</u>
  - Network-level: 8×8 network with new node
- Node-level evaluation: see paper for details
  - New arbitration node vs. two previous designs:
    - Baseline [Horak/Nowick NOCS-10]
    - Predictive [Gill/Nowick NOCS-11]
  - 90nm ARM standard cells, gate-level SPICE simulation
- Network-level evaluation: our focus
  - Three 8×8 MoT networks: each has 112 router nodes
    - Baseline, Predictive, New
  - Modeled in structural technology-mapped Verilog
    - <u>more accurate model</u> than in [Gill/Nowick NOCS-11]
  - 8 synthetic benchmarks: a wide range of traffic patterns

Benchmarks

- 8 diverse benchmarks
  - The same as those in NOCS-11
  - Represent different network conditions
- Classification
  - Three friendly benchmarks:
    - **(1)** Shuffle, **(2)** Tornado and **(7)** Single Source broadcast [Dally`03]
    - No contention
  - Three moderately adversarial benchmarks:
    - **(4)** Simple alternation with overlap
    - **(5)** Random restricted broadcast with partial overlap
    - **(8)** Partial streaming with random interruption
    - No contention for some nodes, light or moderate contention for others
  - Two most adversarial benchmarks:
    - **(3)** All-to-all random and **(6)** Hotspot8
    - Heavy contention at some nodes

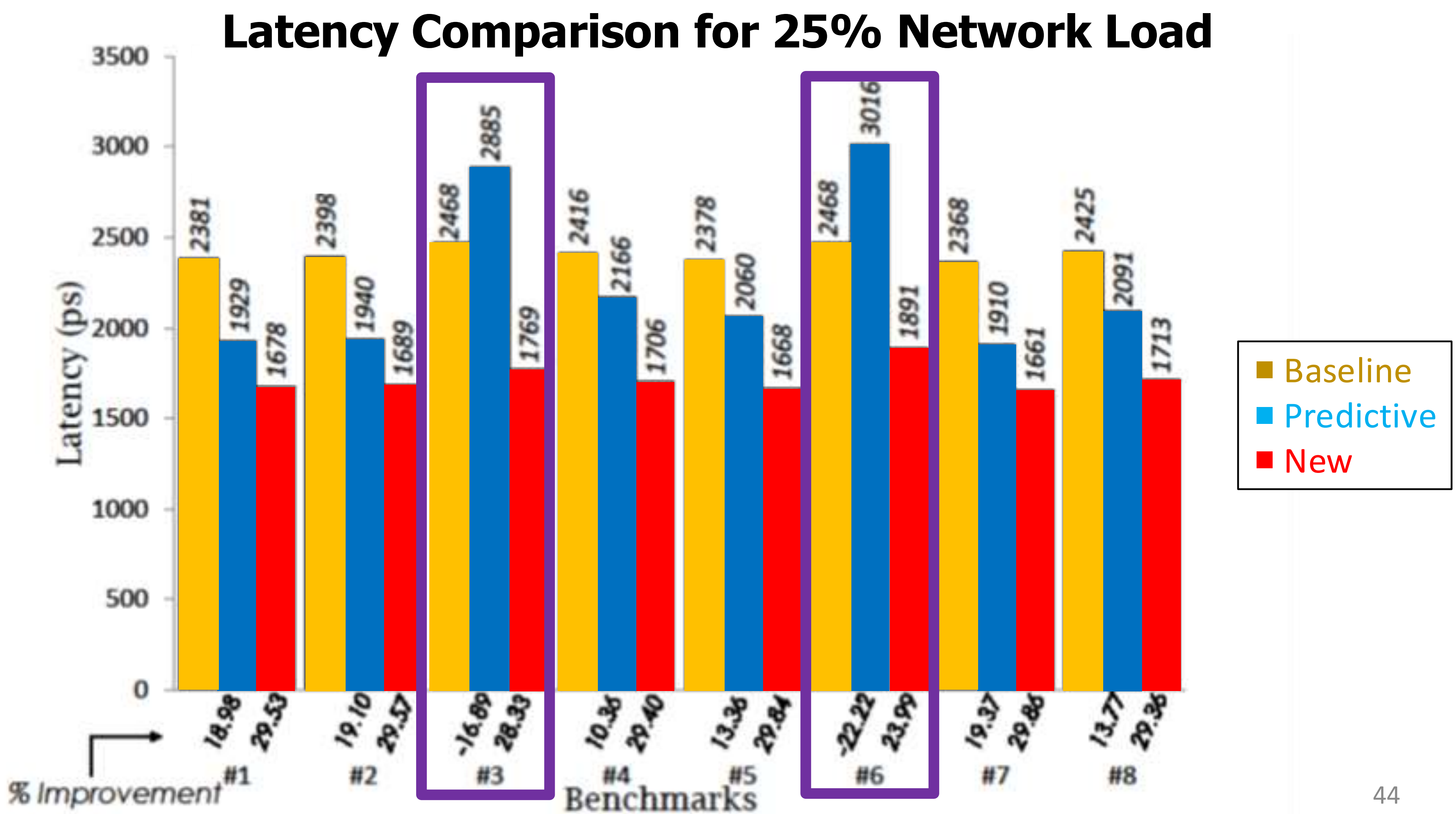
Network-Level Latency: Single-Flit Design

➢ Moderate to significant improvement over <u>all benchmarks</u>
- New vs. baseline: 23-30% improvement
- New vs. predictive: 13-38% improvement



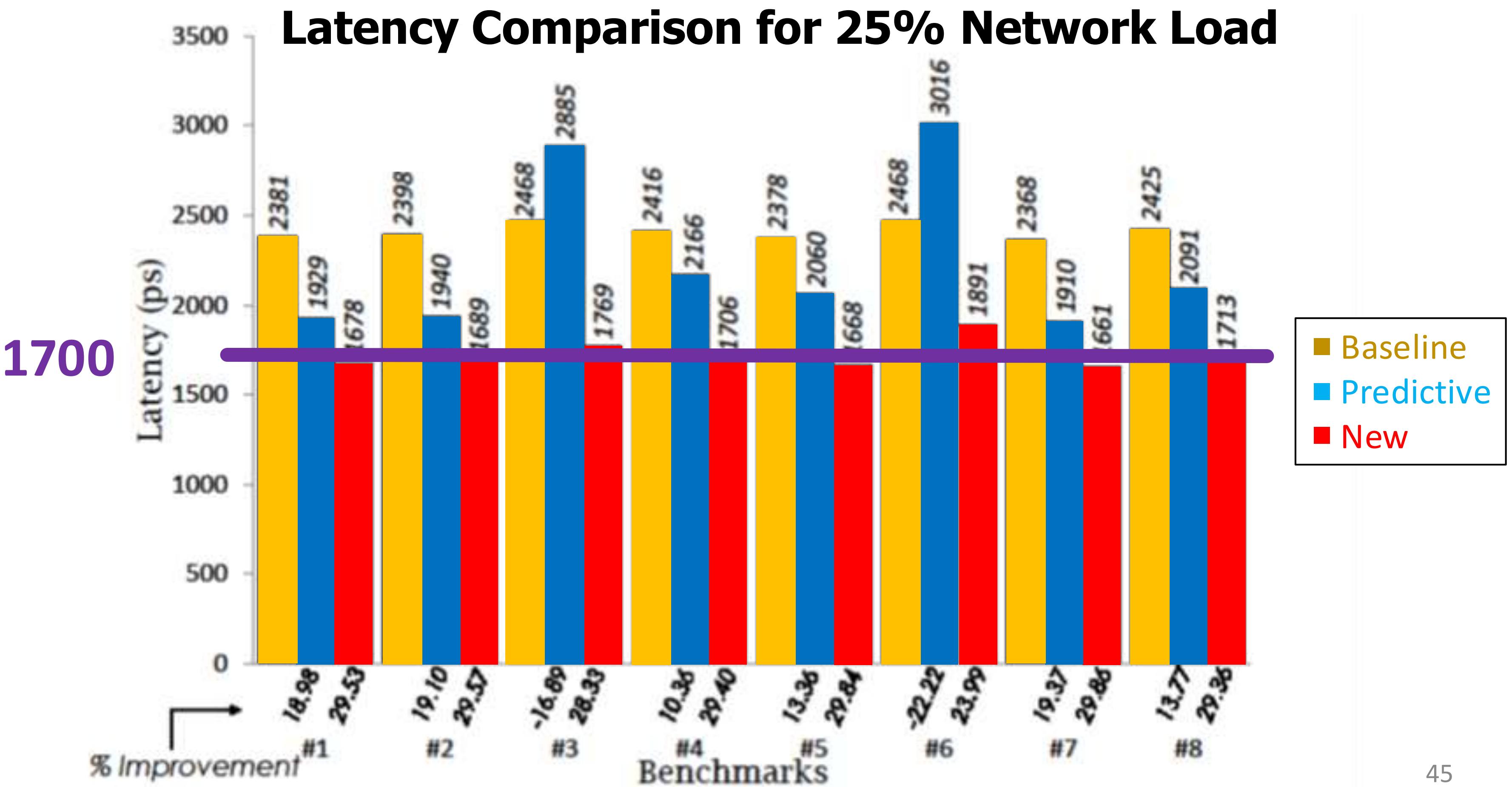
Network-Level Latency: Single-Flit Design

- Perform well for benchmark #3 and #6 (adversarial cases)
  - Predictive: even worse than baseline (~20% higher latency)
  - New: better than baseline (~25% lower latency)



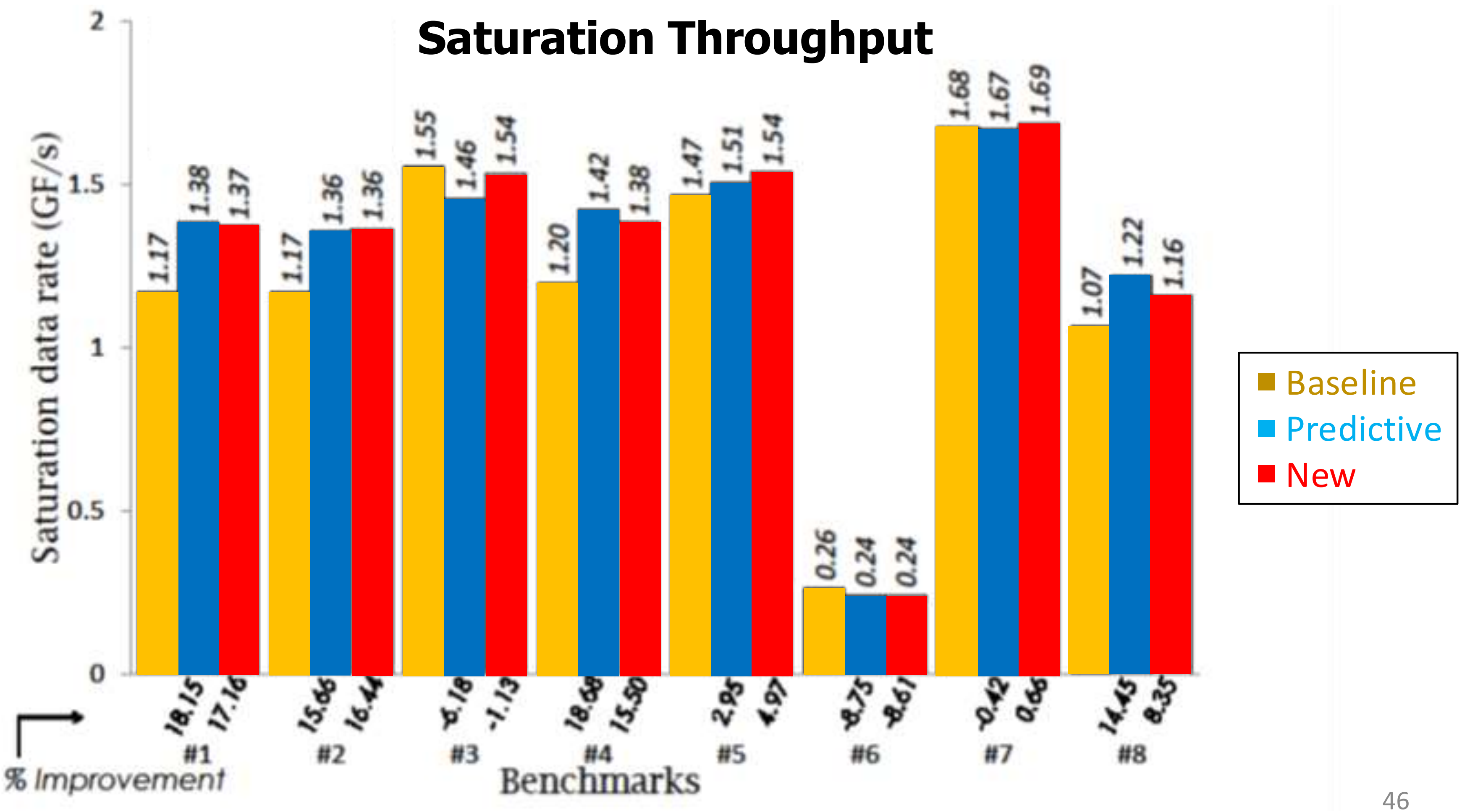
Network-Level Latency: Single-Flit Design

- Excellent latency stability: provides predictable behavior
  - Network latency = ~1700ps, across all benchmarks
    through 6 router nodes + 5 hops
  - Important for memory access in CMP's



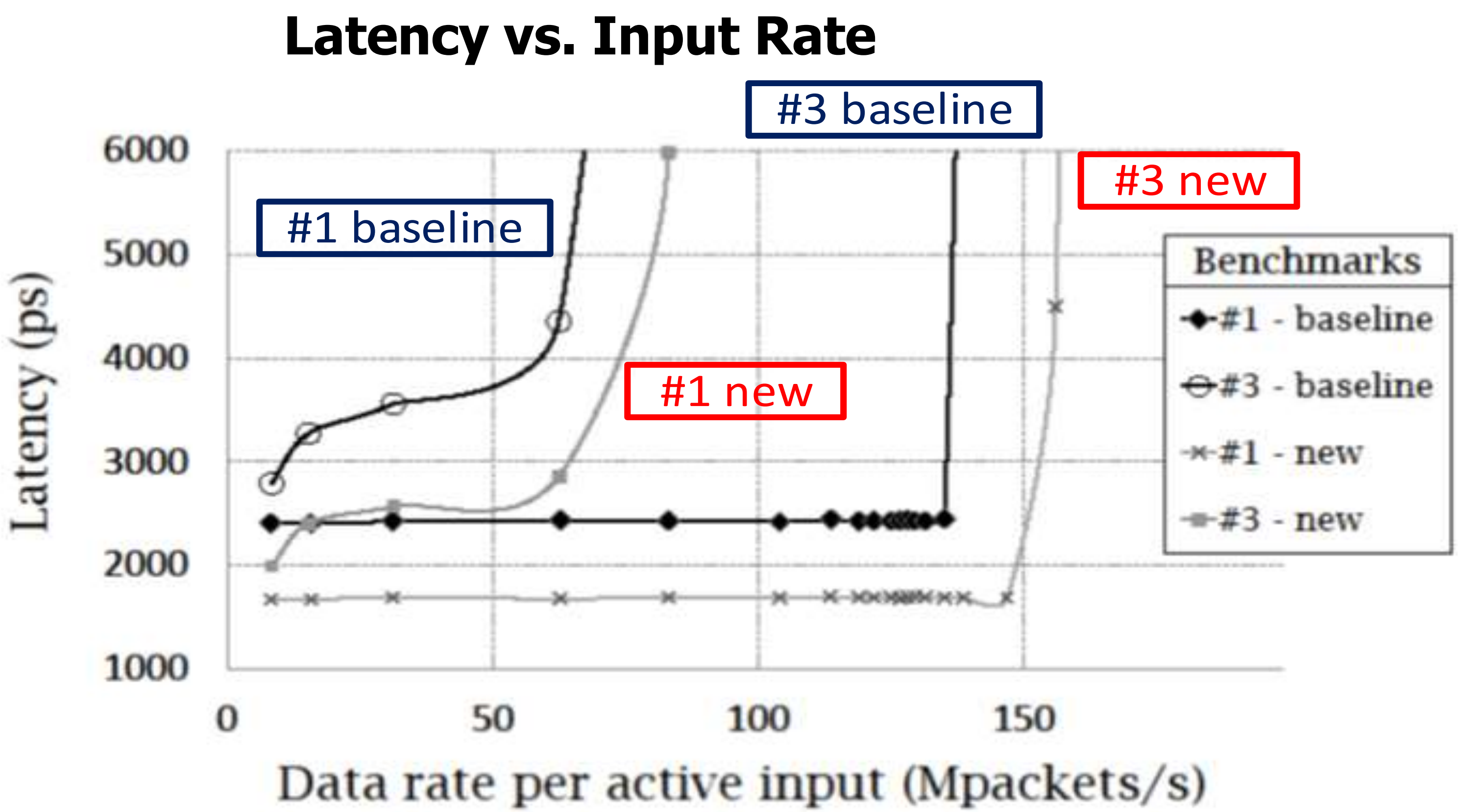
Network-Level Throughput: Single-Flit Design

- New vs. baseline: improvement up to 17% on 6 benchmarks
- New vs. predictive: comparable throughput over all benchmarks



Network-Level Results: Multi-Flit Design

- Fixed packet length = 3 flits/packet
- Results only for benchmark #1 and #3
- For both benchmarks:
  - ~30% latency and ~14% throughput improvement



Conclusion

- Introduced a MoT network using "early arbitration"
  - Address system-latency bottleneck
  - Observe newly entering traffic
    - via lightweight shadow monitoring network
  - Perform early arbitration + channel pre-allocation
- Detailed experimentation and analysis
  - Significant improvements in system-latency
    - New vs. baseline: 23-30% across all benchmarks
    - New vs. predictive: up to 38%

Future Work

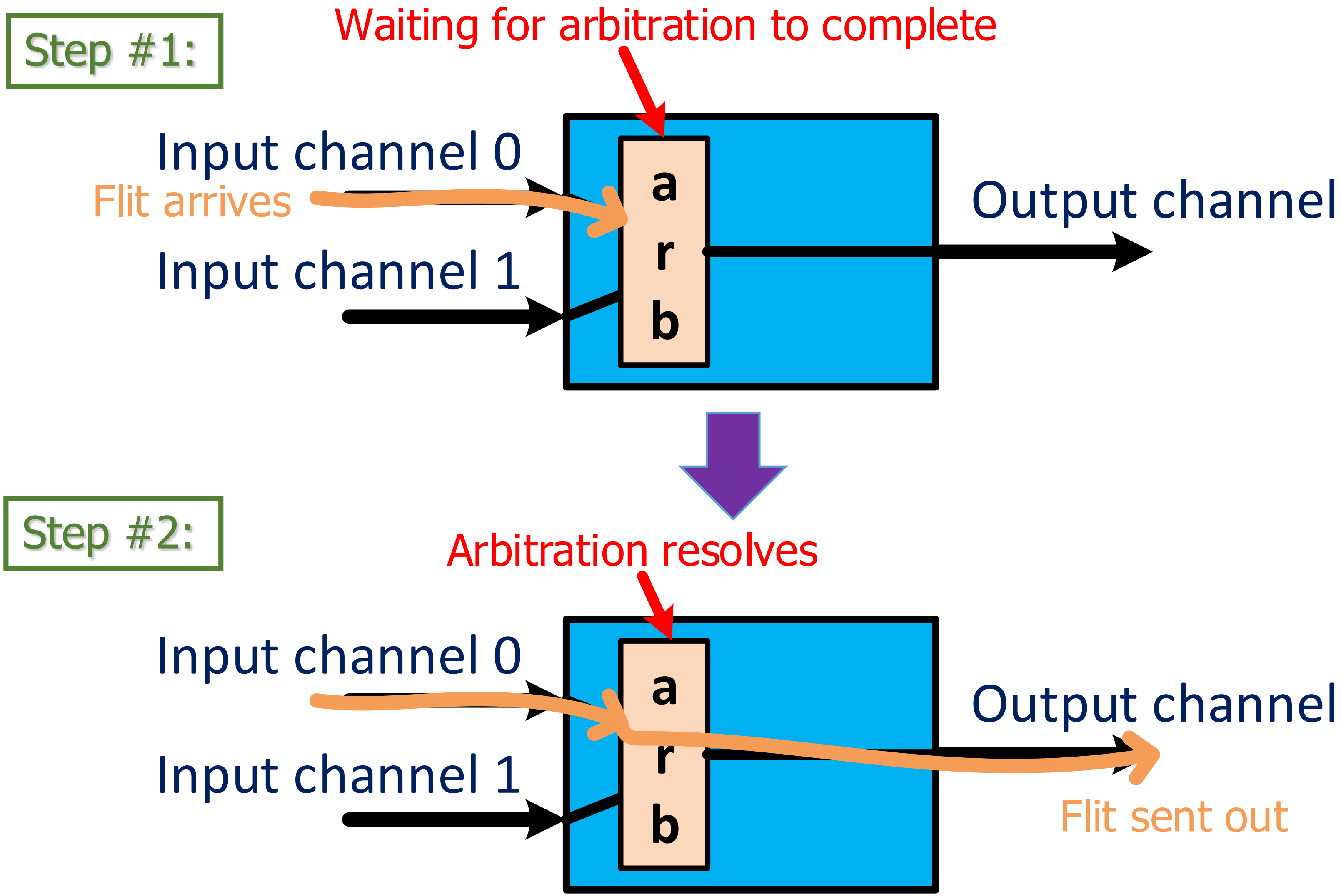
- Narrow channel reservation window
  - Decrease time between "channel reservation" and "flit arrival"
  - Increase network utility
- Target different topology
  - Extend "early arbitration" to 2D-mesh, Clos network, etc.
- Build a complete GALS system
  - Add mixed-timing interface ⇨ connect cores by the network
- More experiments
  - Real traffic benchmarks

Back-up Slides

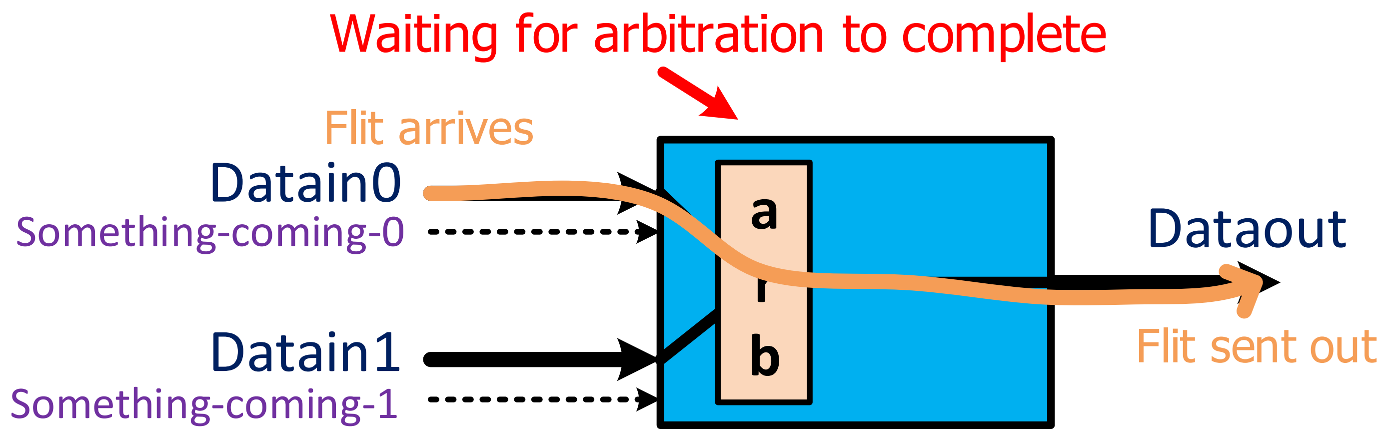
Strategy Comparison: Overview

- Three network designs
  - Baseline
    - [Horak/Nowick, NOCS-10]
    - foundation of the research
  - Predictive
    - [Gill/Nowick, NOCS-11]
    - a more recent design
  - New
    - the proposed design

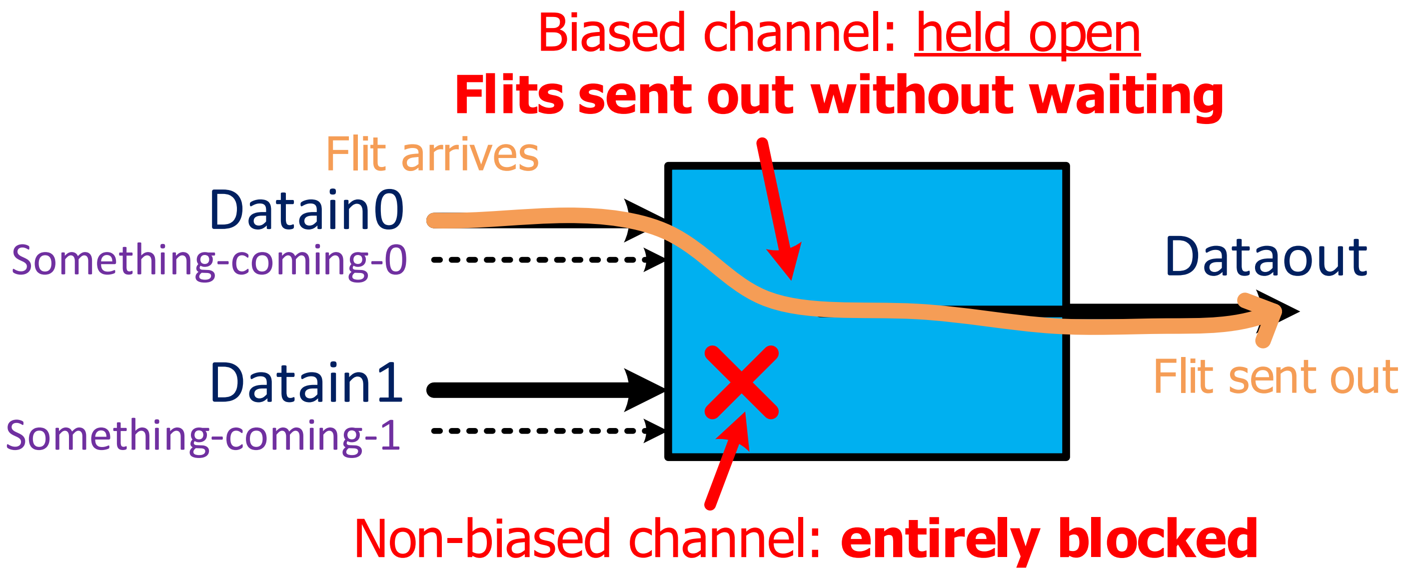
Baseline Arbitration Node: Operation



Predictive Arbitration Node: Operation

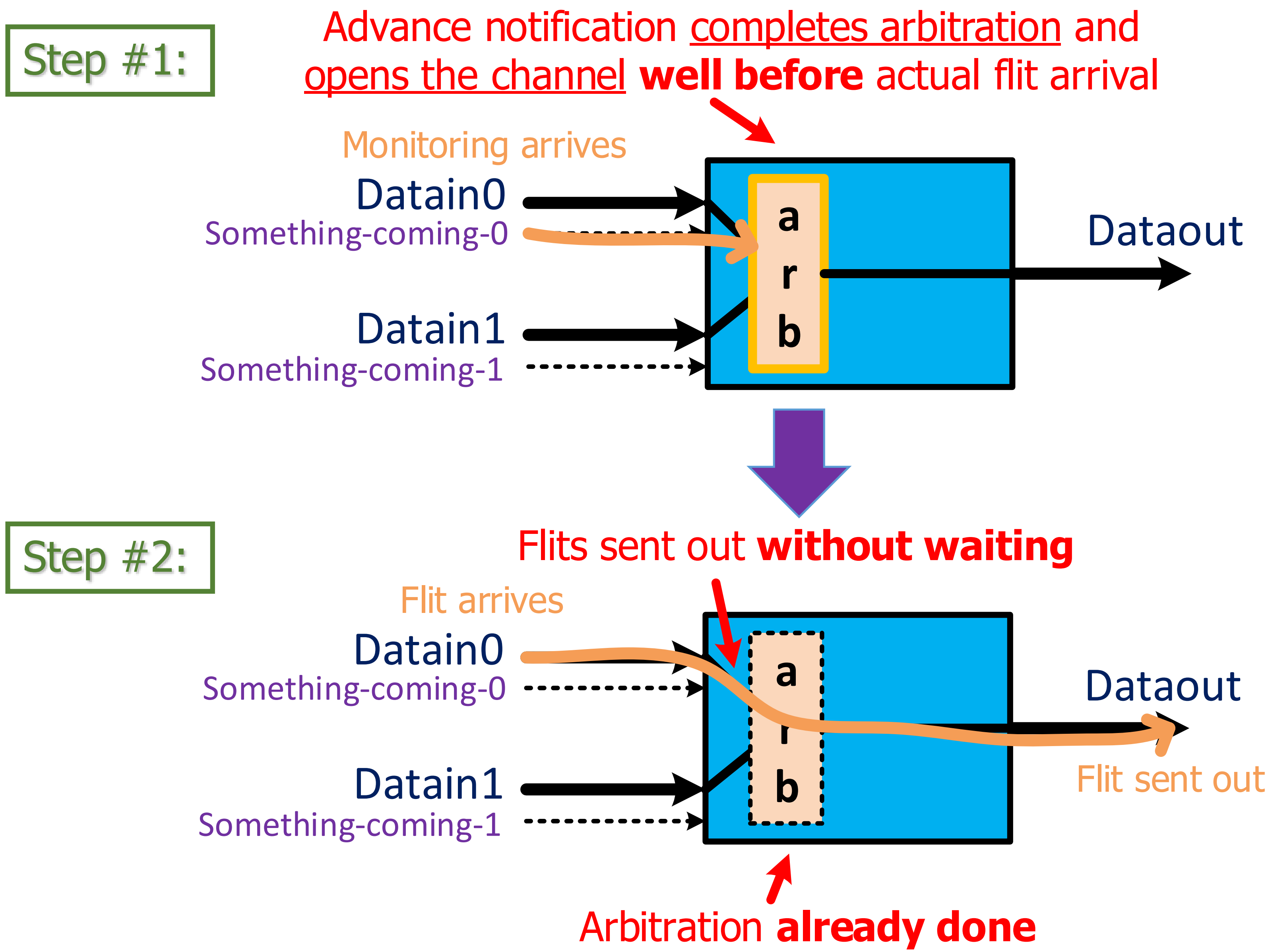


Default Mode: similar operation as "baseline" design



Biased Mode: optimized for one input channel (by prediction)

New Arbitration Node: Operation



The Role of Monitoring Network

- Predictive design [Gill/Nowick, NOCS-11]
  - Facilitates mode change
    - from optimized (biased) to unoptimized (default) only
  - For safety purpose only
    - plays secondary role
- New design
  - Key component of early arbitration strategy
    - directly initiates early arbitration
  - For higher performance
    - especially **system-latency**