The Design and Implementation of a Low-Latency On-Chip Network

Robert Mullins

11th Asia and South Pacific Design Automation Conference (ASP-DAC), Jan 24-27th, 2006, Yokohama, Japan.

Introduction

- Current economic and technology scaling trends will force a step change in computing architectures and approaches to VLSI design
- Design methodologies will shift from computation-centric to communication-centric ones.
- This talk will examine a major component of such approaches: the *on-chip network*

Economic Trends

- Falling chip design budgets
  - Hardware budgets squeezed as software complexity grows
  - Rising Non-Recoverable Engineering (NRE) costs as fabrication technologies scale
- Continued time to market pressures
- Need to reduce complexity and risk

Technology Scaling Trends

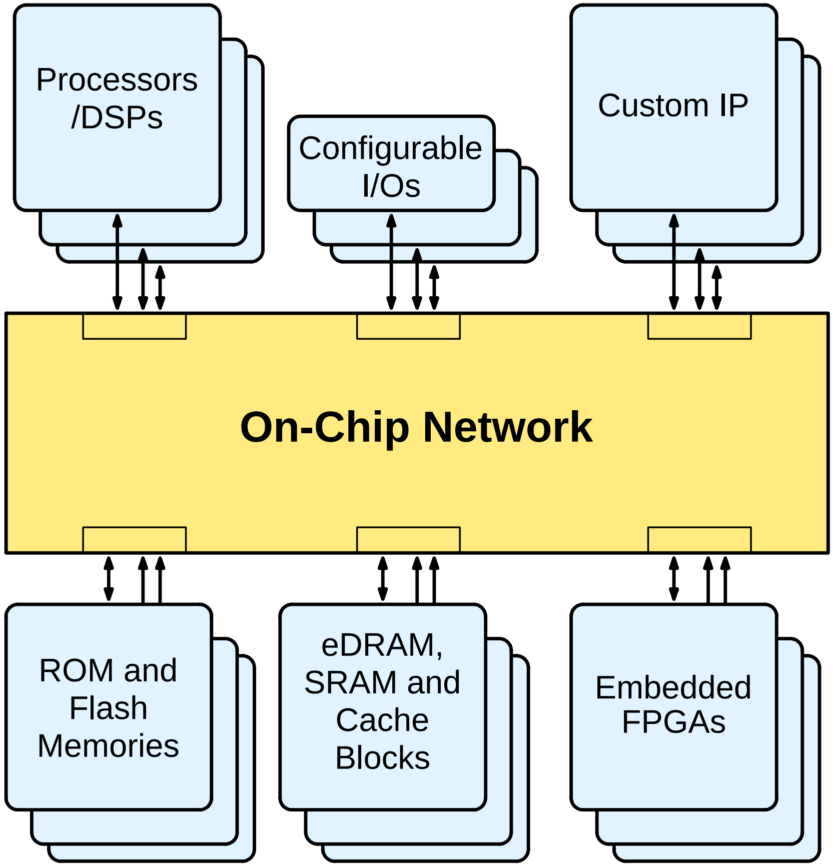
- Interconnect scales poorly
  - Begins to dominate delay, power budgets and area
- Benefits of regular interconnects increase
  - Ability to better optimise power and delay
  - Reduced verification effort
  - Simple to analyse, low risk
- Yield and reliability issues
  - Fault tolerant design, remapping and reconfiguration
- Power limited designs
  - Optimizing power boosts performance

Design Trends

- Systems will be continue to be composed from larger numbers of IP blocks
- Increasing use of coarse-grain parallelism
  - The last remaining tool to maintain historical performance gains in a power constrained environment
- Economic and risk pressures are forcing designs to become increasingly programmable and general purpose
  - Ability to map many applications to a single chip

Communication-Centric SoC Design

- Scalable communication infrastructure
  - Regular and optimised
- Network eases application mapping, reuse and integration issues
  - General purpose interconnect
- **Network schedules compute resources:**
  - Optimises/manages power
    - Has global view and influence
  - Manages local thermal budgets
  - Central to fault tolerant abilities
- Much more than simply a move from buses to networks

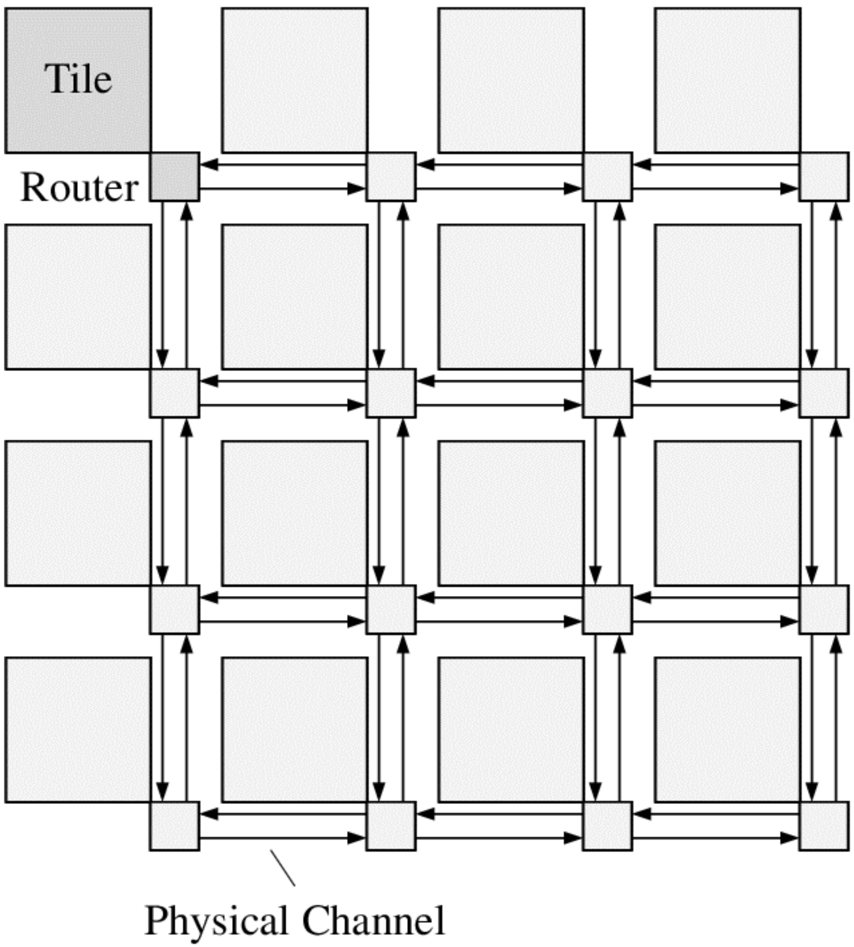


Many Challenges

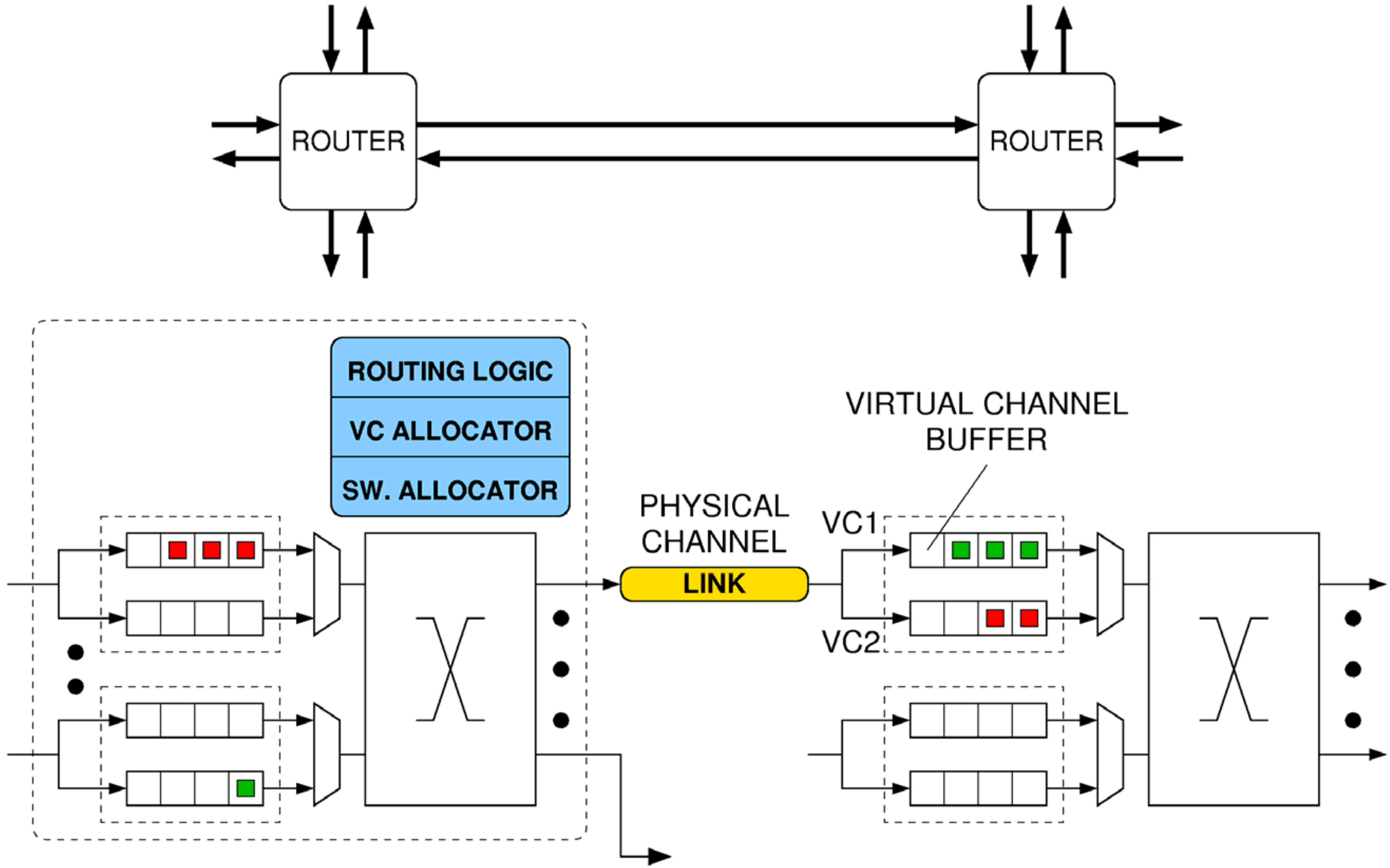
- Application mapping
- Network topologies
- Fault-tolerant techniques
- System-level communication-centric power management
- Guaranteeing correctness in these increasingly distributed systems
- Low-power techniques for on-chip networks
- …..
- This talk will look at:
  - **Building low-latency on-chip routers**
  - **How to clock on-chip networks**

Introduction to On-Chip Networks

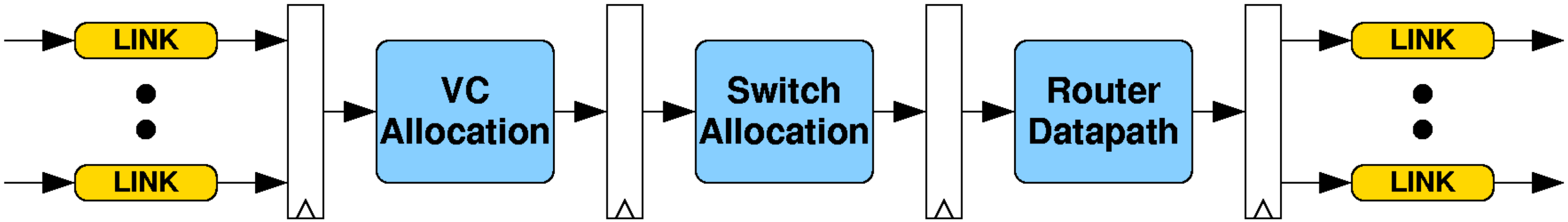
- All chip-wide communications are handled by an on-chip network
- Packet-switched network
- Each router contains
  - Input buffers
  - Routing logic
  - Scheduling hardware
    - Arbitration
  - Crossbar



Virtual-Channel Flow Control

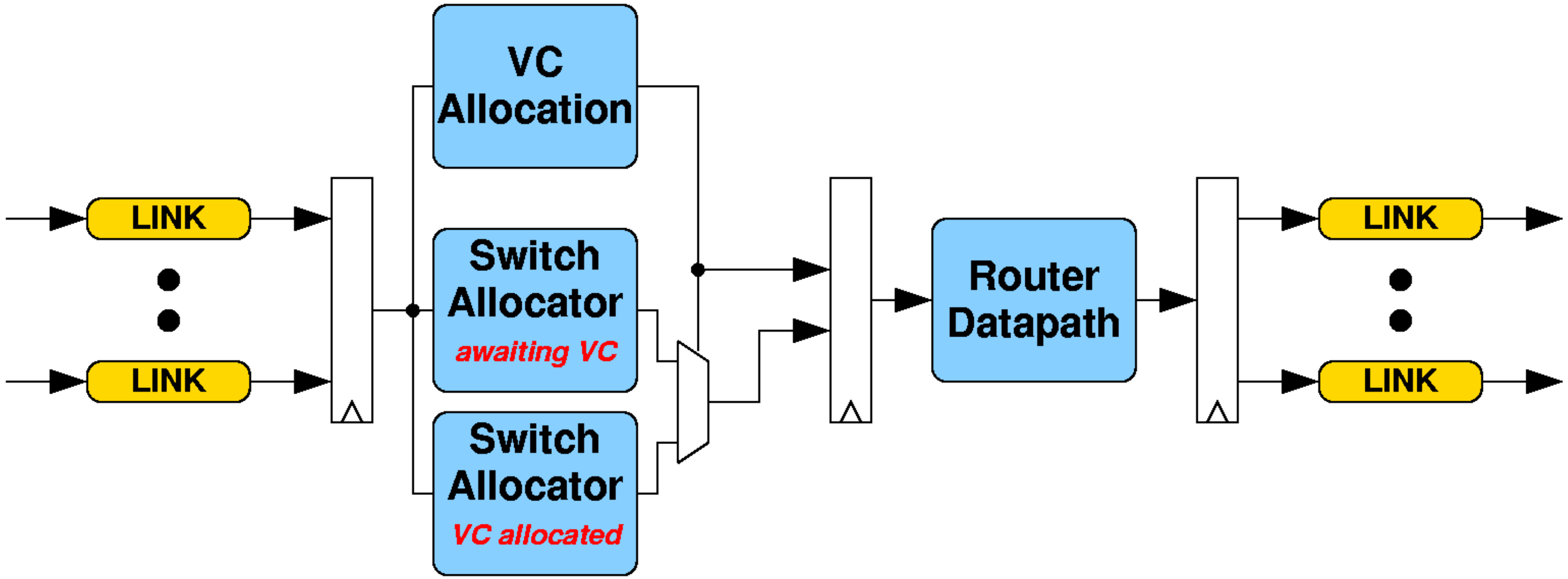


Synchronous Router Pipeline



- Router Pipeline may be many stages
  - Increases communication latency
  - Can make packet buffers less effective
  - Incurs pipelining overheads

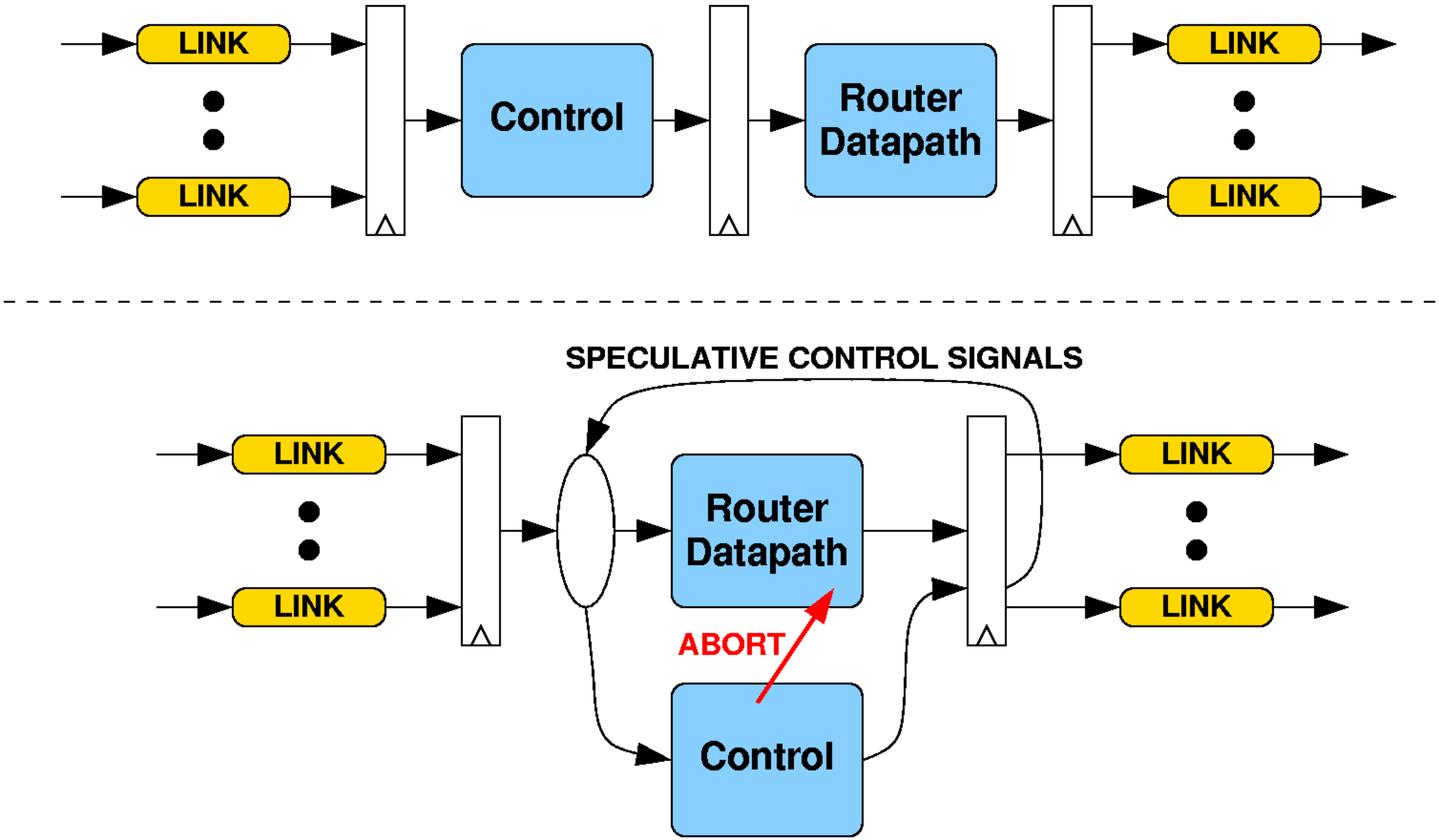
Speculative Router Architecture



- VC and switch allocation may be performed concurrently:
  - Speculate that waiting packets will be successful in acquiring a VC
  - Prioritize non-speculative requests over speculative ones

Li-Shiuan Peh and William J. Dally, "*A Delay Model and Speculative Architecture for Pipelined Routers*", In Proceedings HPCA'01, 2001.

Single Cycle Speculative Router



R. D. Mullins, A. West and S. W. Moore, "*Low-Latency Virtual-Channel Routers for On-Chip Networks*", In Proceedings ISCA'04.

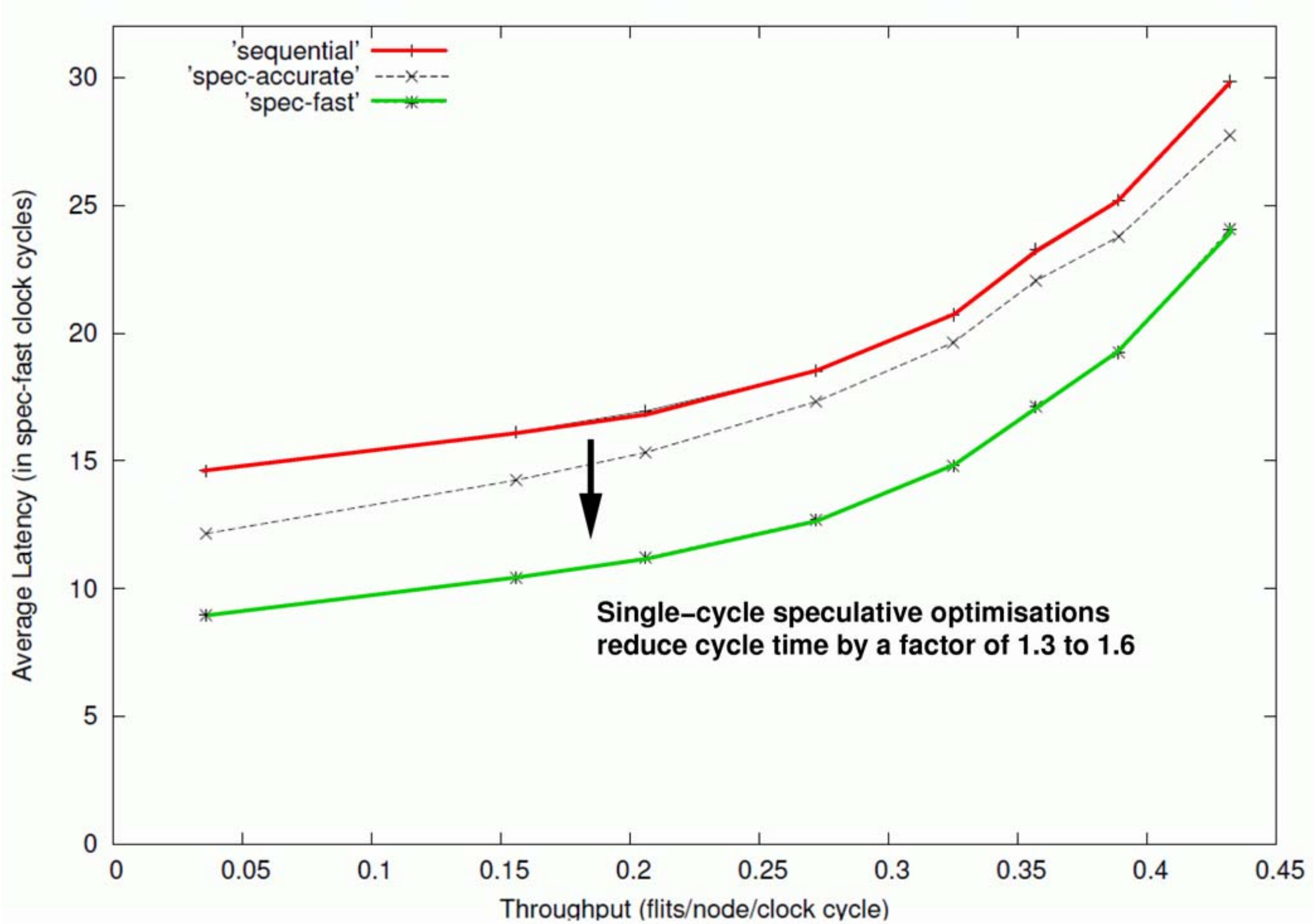
Basic Concept

- Consider two extremes of operation:
- Multiple flits are queued waiting for access to the same output port
  - We have all the information we need to schedule the output port accurately ahead of time
- No requests are outstanding for a particular output port
  - In this case we speculate that arbitration will be unnecessary and permit any new flit to be routed to its required output immediately
  - Easy to abort if things go wrong. Just look at newly arriving flits and the output ports they require

Optimisations

- To produce control signals for the next clock cycle we compute the requests (VC or switch allocation) that we know will remain
- In the case of the VC allocator it is important for performance that this is accurate
- For the switch allocator logic a better trade-off is to minimise this logic and obtain gains through reduction in cycle-time

Results

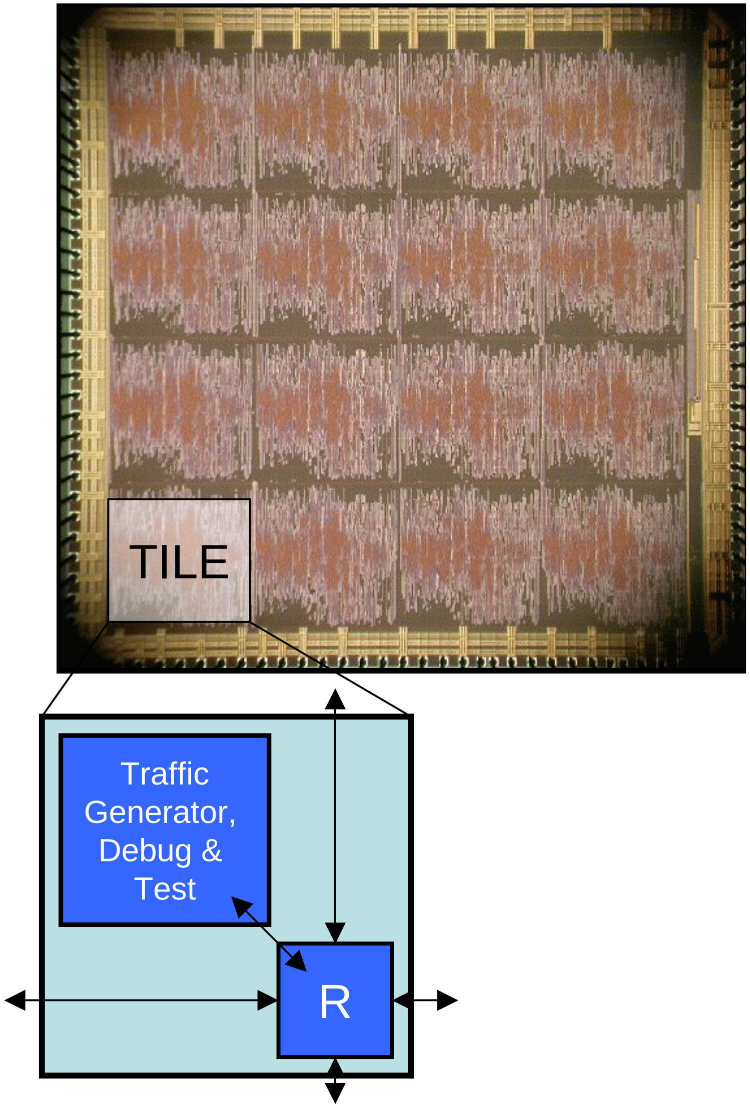


Comparison to single-cycle router without speculative optimisations

4x4 mesh network, random traffic, 4 flit (256-bit) packets

The LOCHSIDE Testchip

- UMC 0.18um Process
- 4x4 mesh network, 25mm^2
- Single Cycle Routers (router + link = 1 clock)
- May be clocked by both traditional H-tree and DCG
- 4 virtual-channels/input
- 80-bit links
  - 64-bit data + 16-bit control
- 250MHz (worst-case PVT) 16Gb/s/channel (~35 FO4)
- Approx 5M transistors

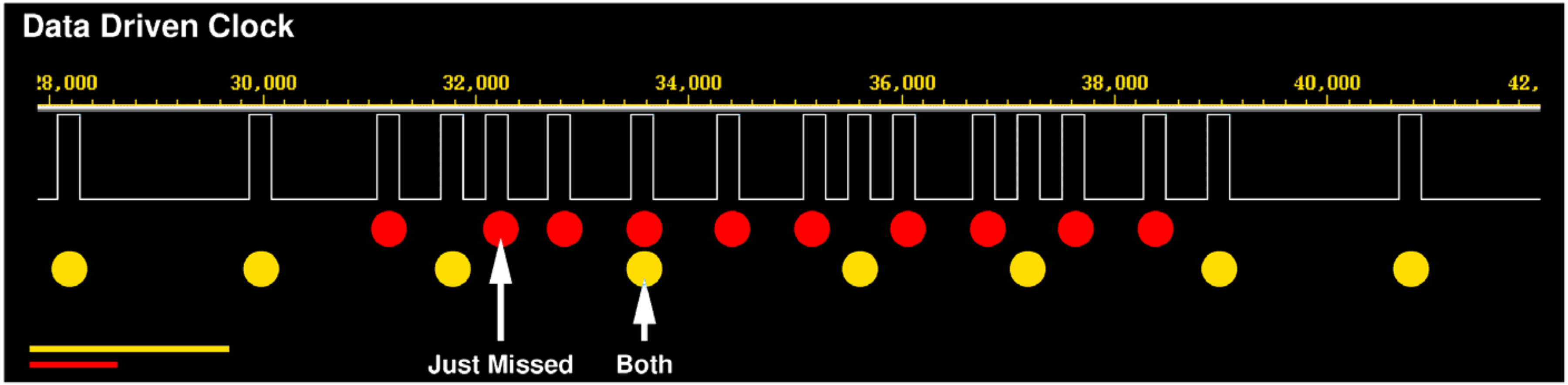


Clocking On-Chip Networks

- Challenges:
  - Clock Distribution Issues
    - Challenging due to networks physically distributed implementation
    - Potentially a high-frequency clock
      - Power and skew concerns
  - Synchronization
    - IP Blocks will run at many different or even adaptive clock frequencies
  - What frequency does network run at?
    - Interesting problem!
    - Would like to avoid running at max. freq all the time - may not want to increase latency?

Data-Driven Clocking

- Idea:
  - Generate the clock locally at each router
  - Generate clock pulses only when required!
    - Existence of data on router's input triggers new clock pulses
    - Local calibrated delay line ensures clock frequency never exceeds router's maximum
    - Clock is aperiodic

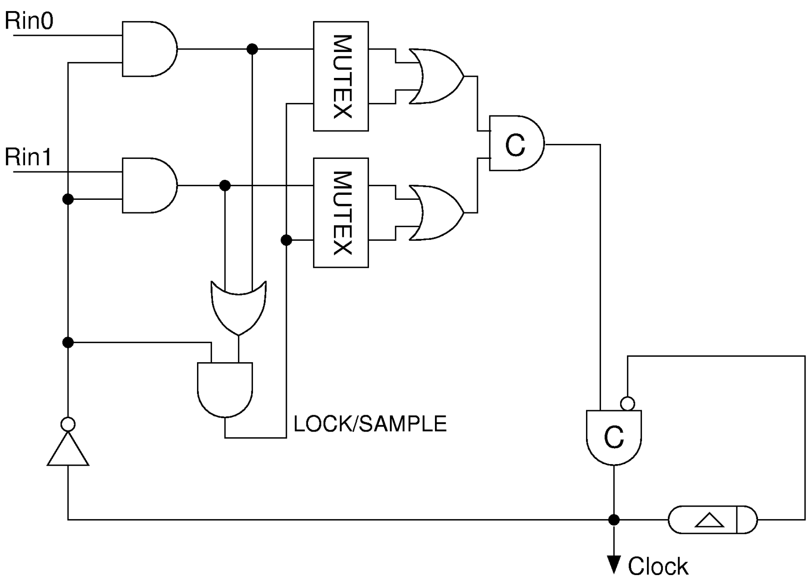


Benefits of Data-Driven Clocking

- Robust value safe synchronization
  - No synchronization delay if router is quiescent
- Event-driven synchronous system!
- Benefits of asynchronous implementation but router remains fast and simple
  - Can still exploit synchronous single-cycle router design
  - No one single network operating frequency
  - No global clock!
  - Network links can be fully-asynchronous if beneficial

Data-Driven Clocking

- Arbitration is necessary to determine whether input data is admitted on the subsequent clock cycle or not.
- If there are always input requests waiting the clock will be periodic and operating at its maximum frequency



Summary

- Single cycle speculative routers
  - Reduce router pipeline to single stage
  - This provides a significant reduction in network latency
- Data-driven clocking for on-chip networks
  - Removes need for global clock
  - Network router are clocked at rate determined by traffic

Conclusion

- Current trends suggest a major shift to a communication-centric approach will be inevitable
- On-chip networks are one important piece of the puzzle!
- Continued performance gains depend on shift in design practices
  - End of the road for evolutionary advances
  - Cannot rely on technology alone for gains

Thank You

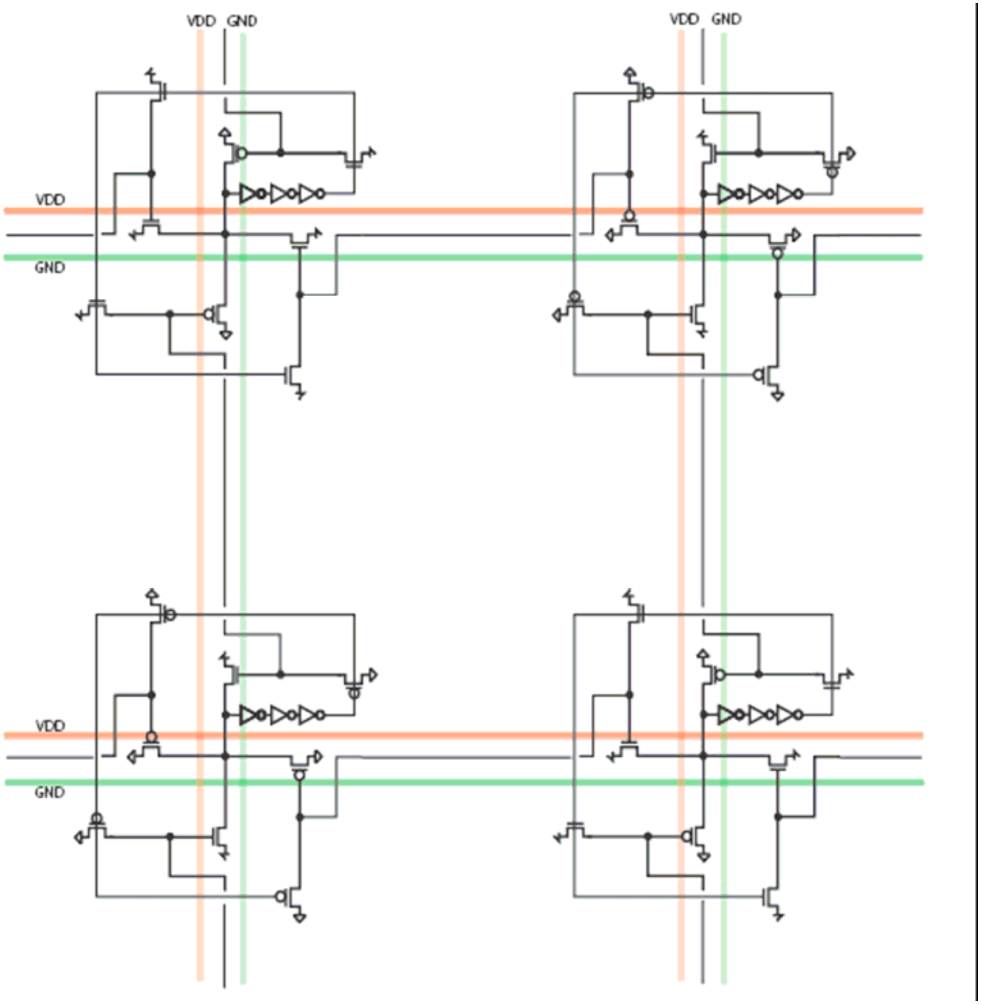
Comments/Questions? Email: Robert.Mullins@cl.cam.ac.uk

Papers, slides and tutorial at http://www.cl.cam.ac.uk/users/rdm34

Other Slides….

Distributed Clock Generator (DCG)

- Exploits self-timed circuitry to generate and distribute a clock in a distributed fashion
- Low-skew and low-power solution to providing global synchrony
- Topology matches that of a mesh network
- Single Frequency
- clock gating?



S. Fairbanks and S. Moore "Self-timed circuitry for global clocking", ASYNC'05