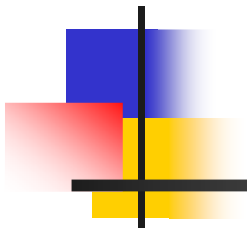
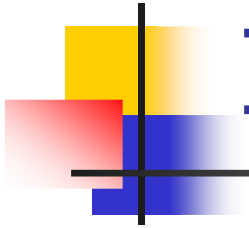


Approaching Ideal NoC Latency with Pre-Configured Routes

George Michelogiannakis, Dionisios Pnevmatikatos and Manolis Katevenis
Institute of Computer Science (ICS)
Foundation for Research & Technology - Hellas (FORTH)
P.O.Box 1385, Heraklion, Crete, GR-711-10 GREECE
Email: {mihelog,pnevmati,kateveni}@ics.forth.gr



Introduction

- Problem: Latency NoCs impose.
- Motivation: Latency introduced to every communication pair.
- Past work: Achieves 1 cycle/hop at 500 MHz.
- We extend speculation to routing decisions.
- Goal: Approach buffered wire latency.
  - Fraction of cycle/hop.

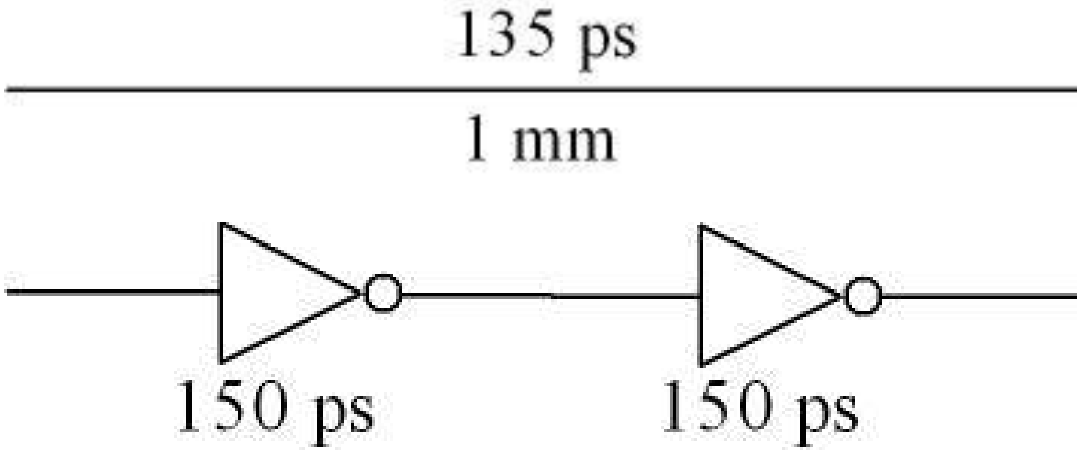
Our Approach

130 nm library

$$\frac{135\ \text{ps}}{1\ \text{mm}}$$

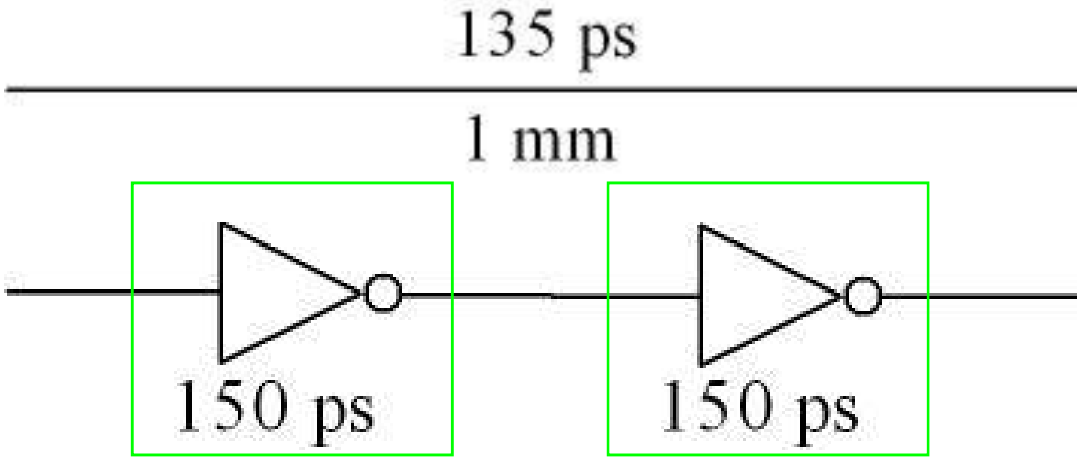
Our Approach

130 nm library



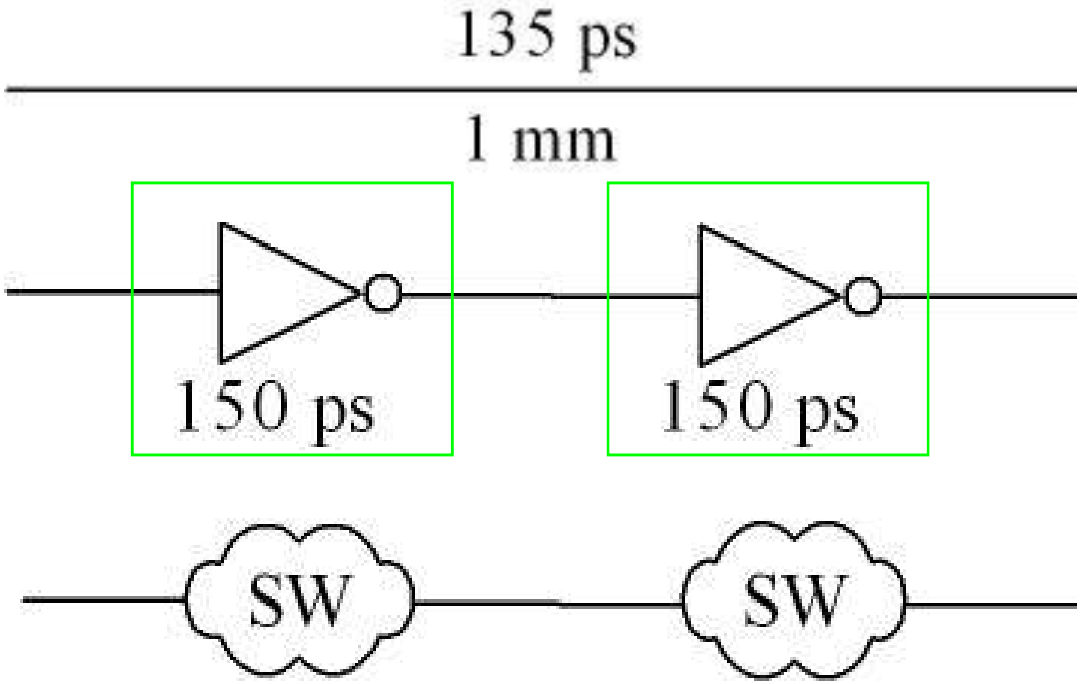
Our Approach

130 nm library

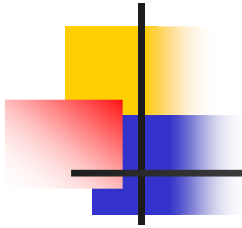


Our Approach

130 nm library



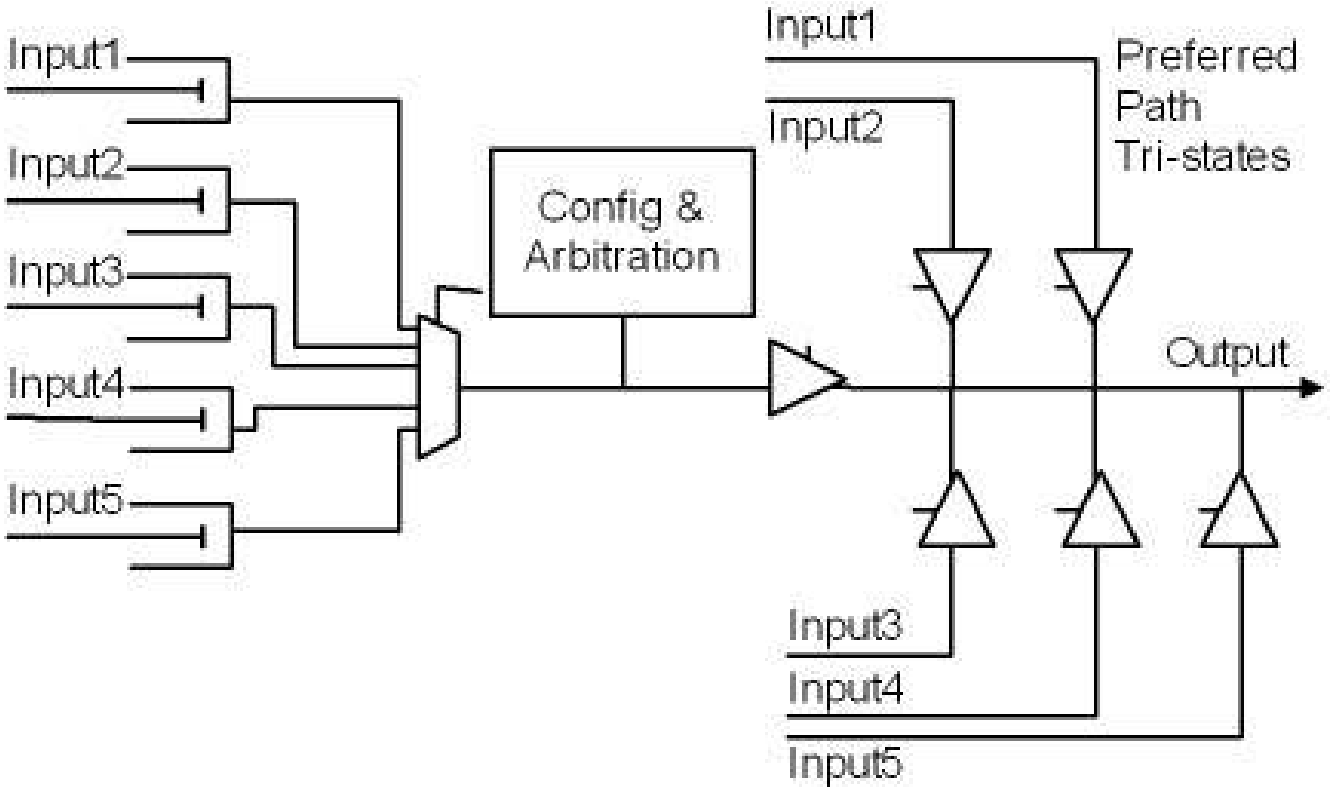
- 400 ps good scenario; 1 cycle otherwise.



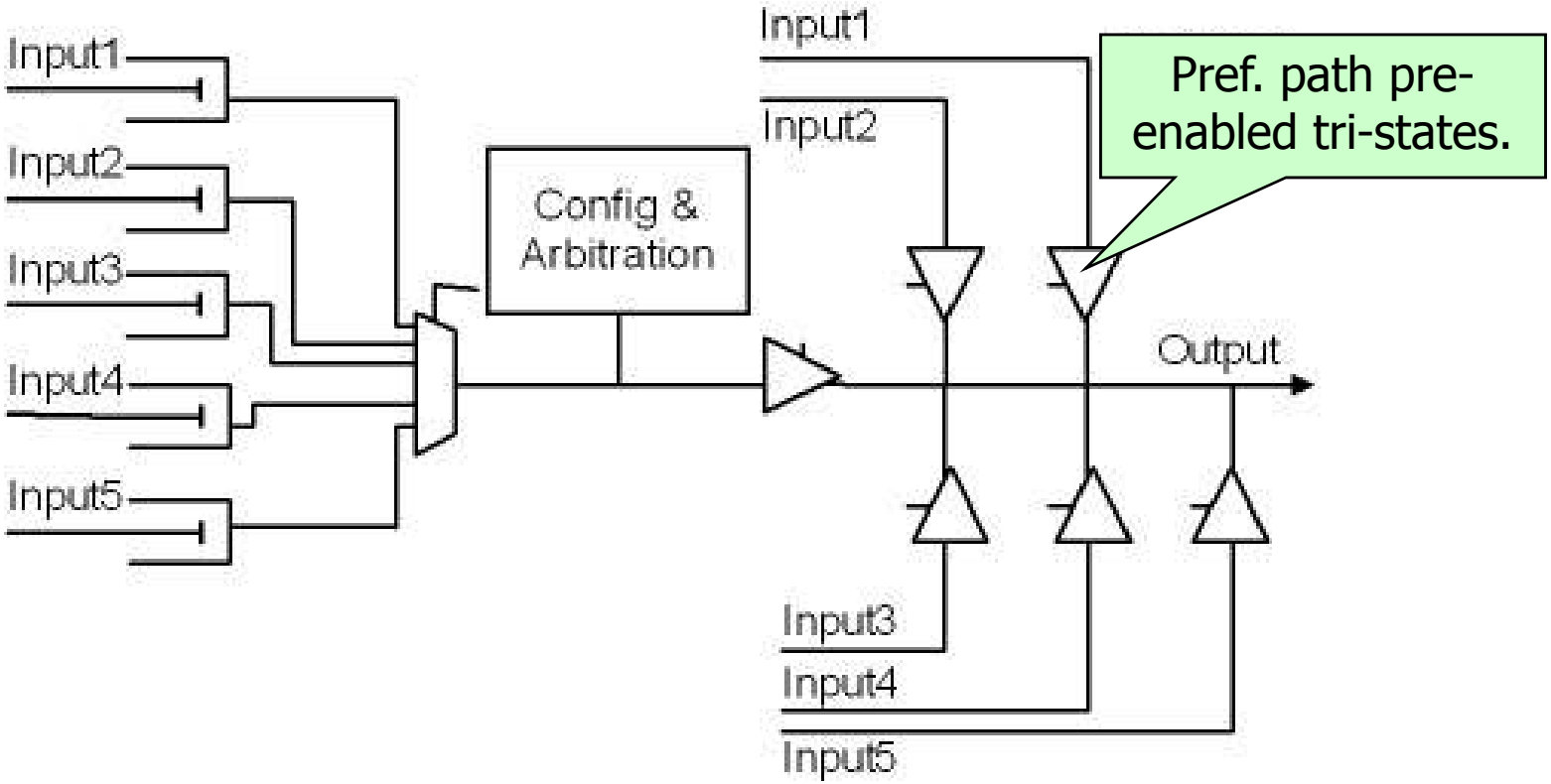
Preferred Paths

- Each output has one preferred input.
- This pref. I/O pair is connected by a single *pre-enabled* tri-state driver.
- Pre-enabling is crucial:
  - 200 ps pre-enabled mux; 500 ps otherwise.
- Later check if flits correctly forwarded.
- Thus, preferred paths are formed.
  - Reconfigurable at run-time.
  - Custom routes (shapes) allowed.

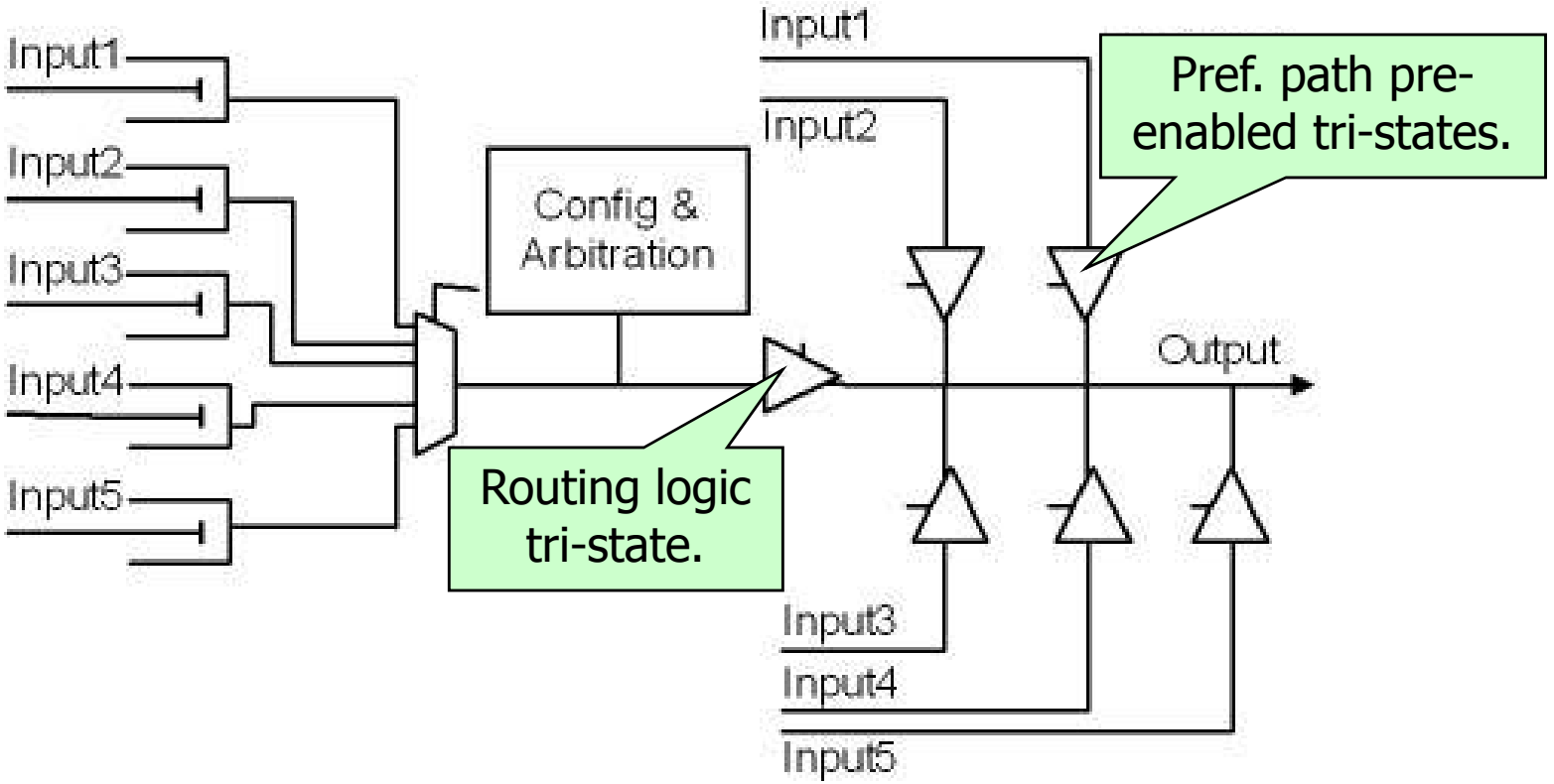
Switch Architecture - Output



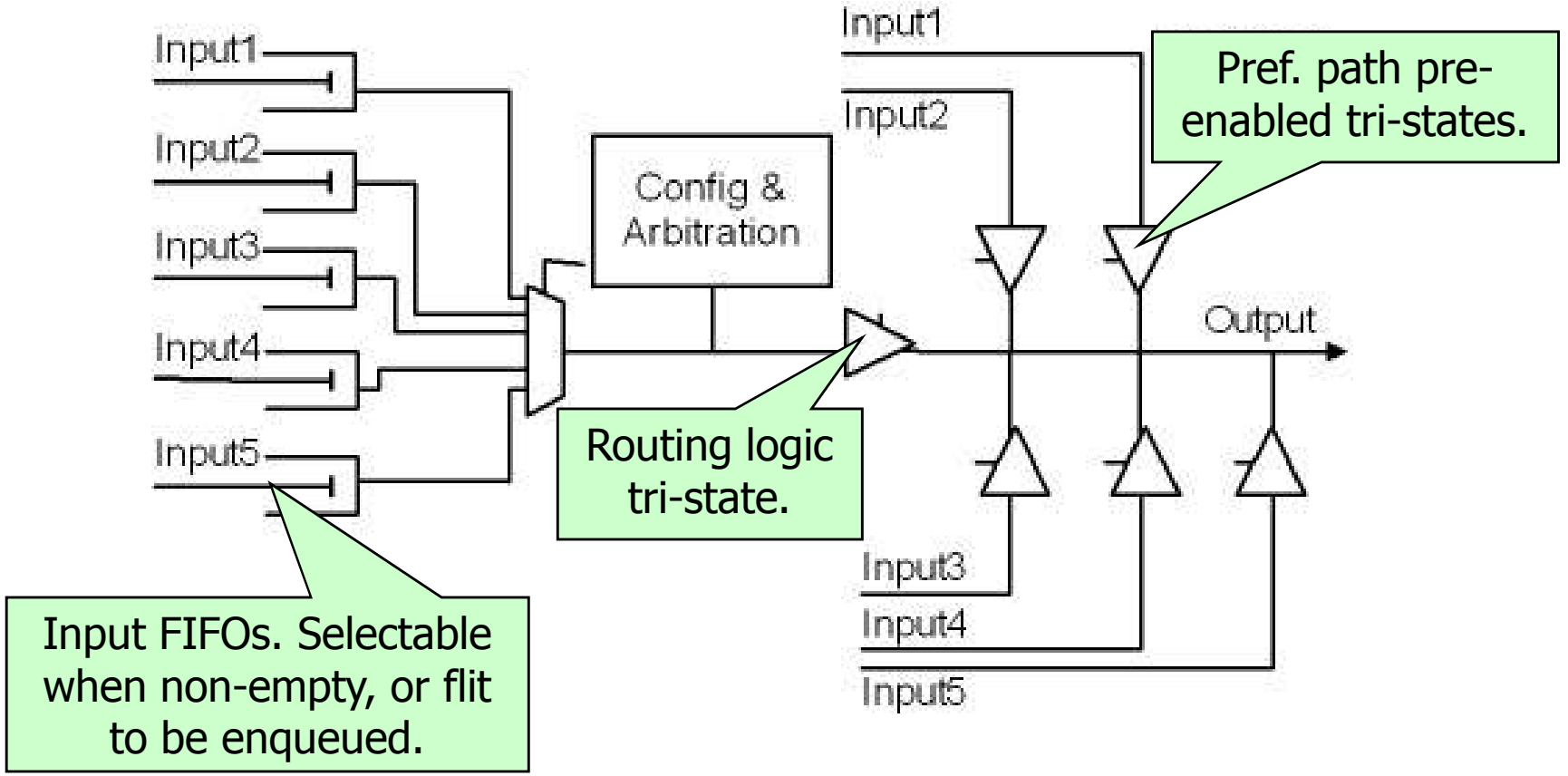
Switch Architecture - Output



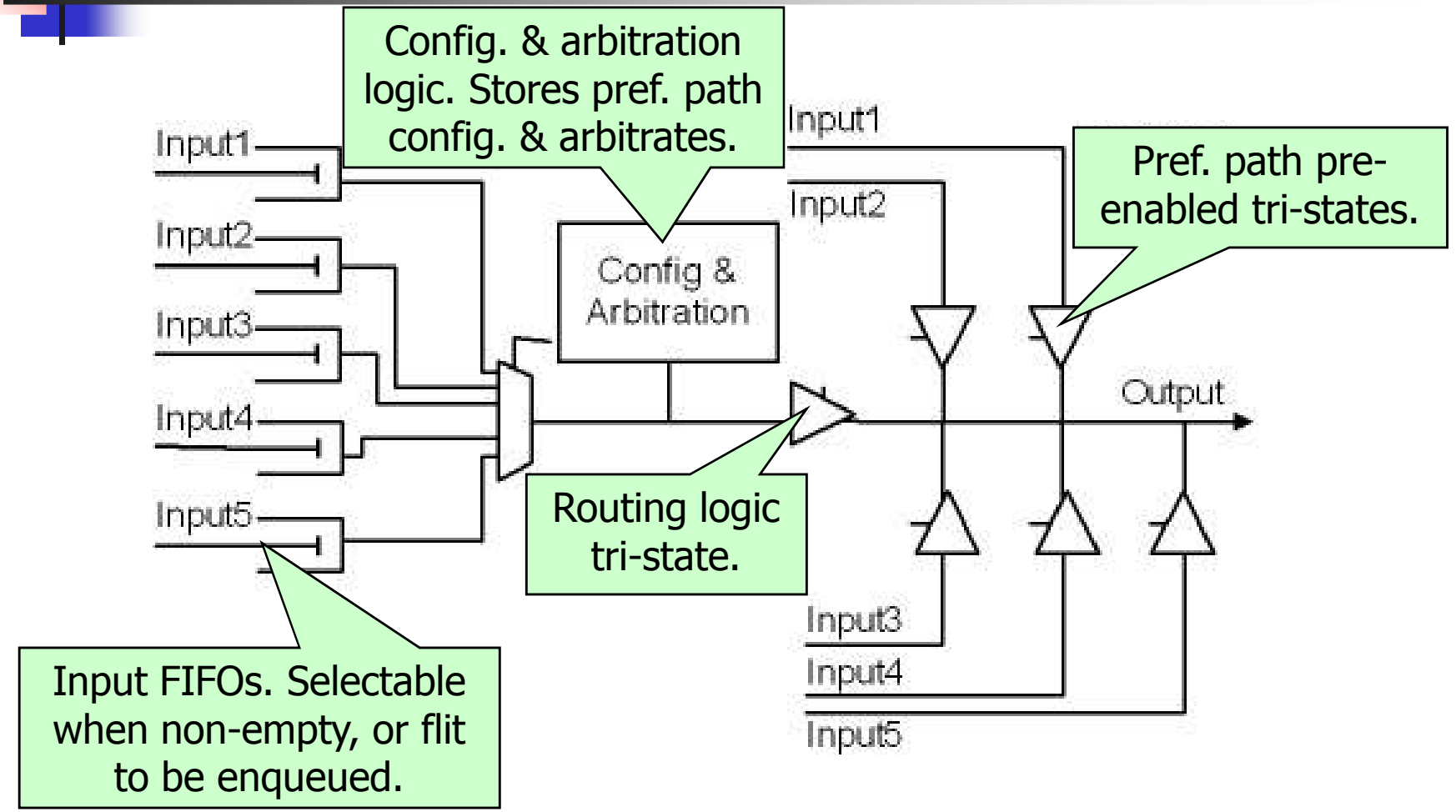
Switch Architecture - Output



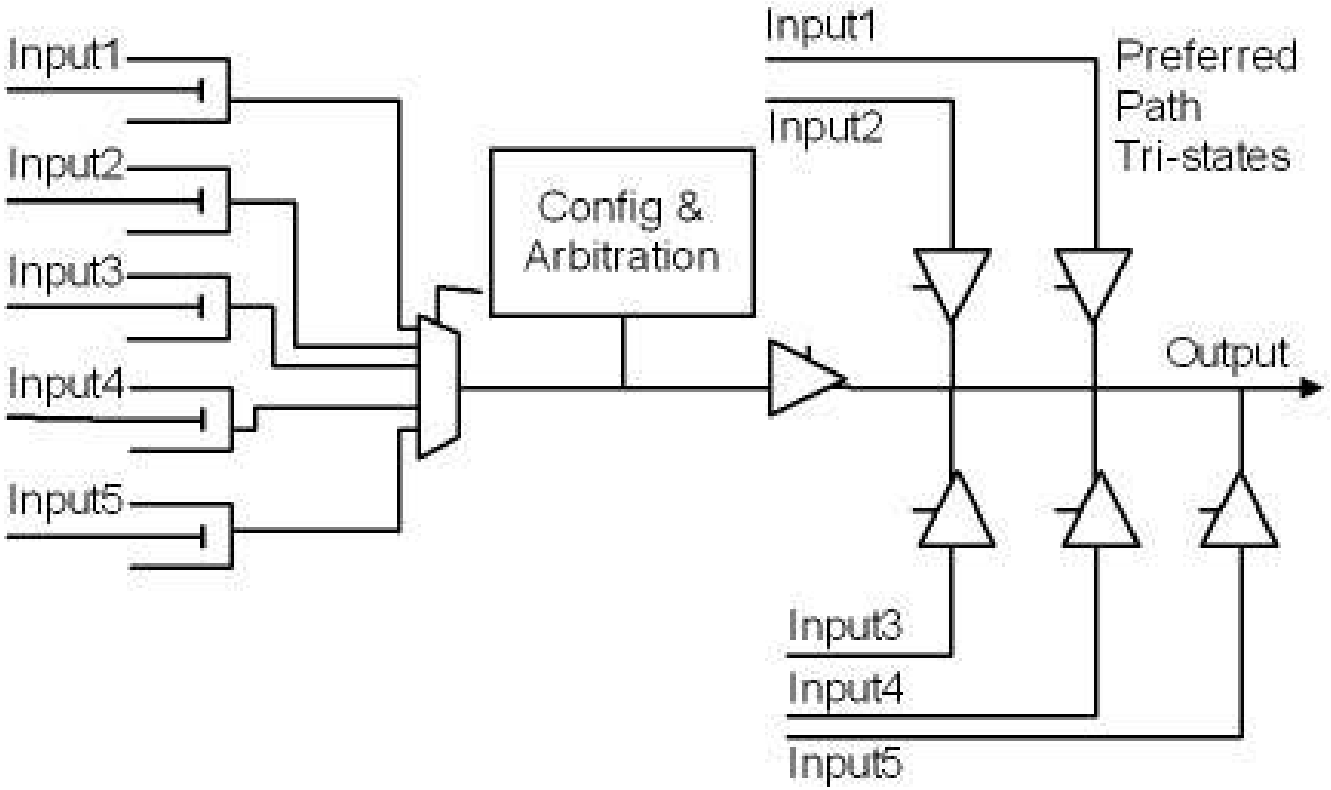
Switch Architecture - Output



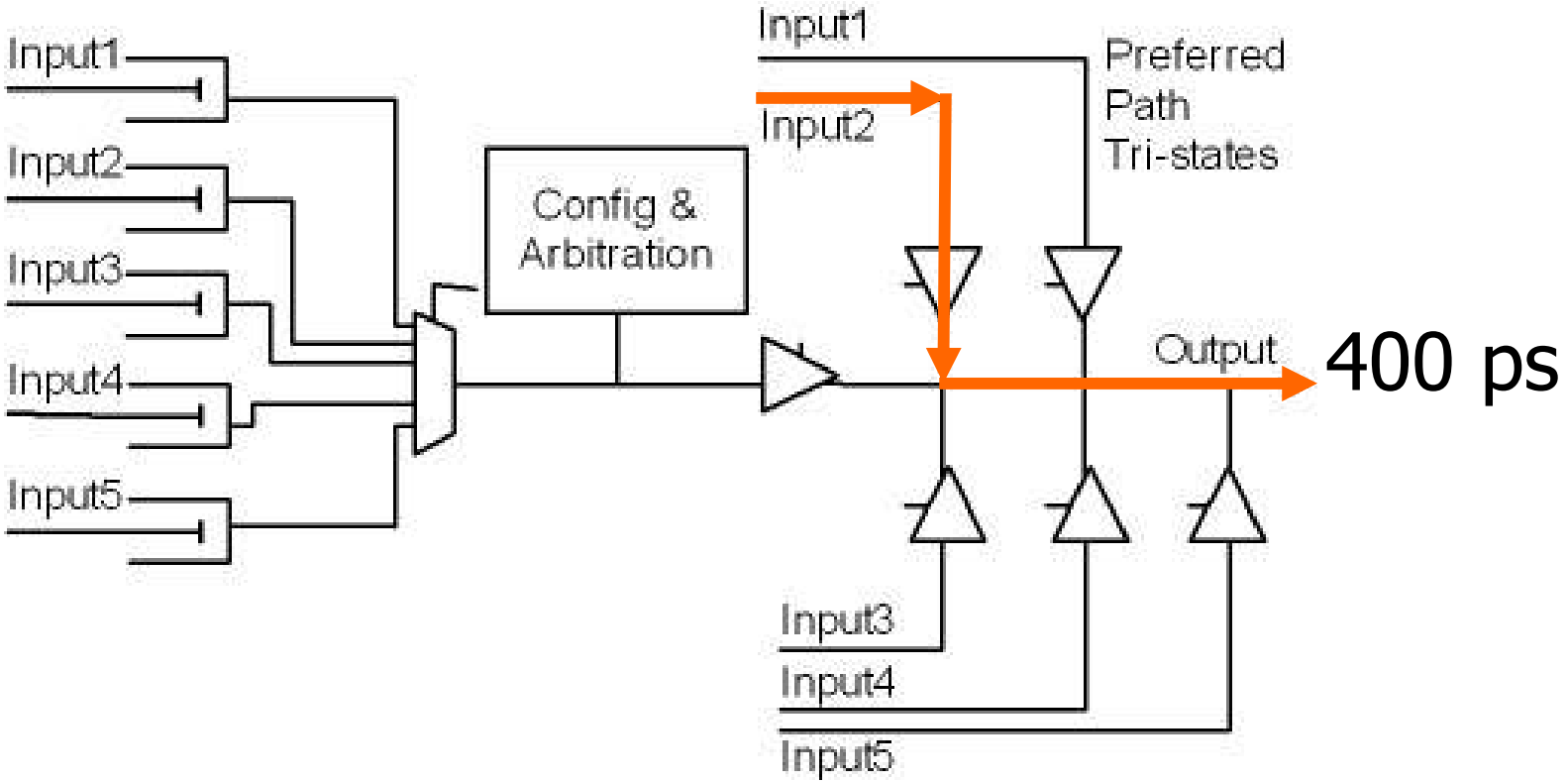
Switch Architecture - Output



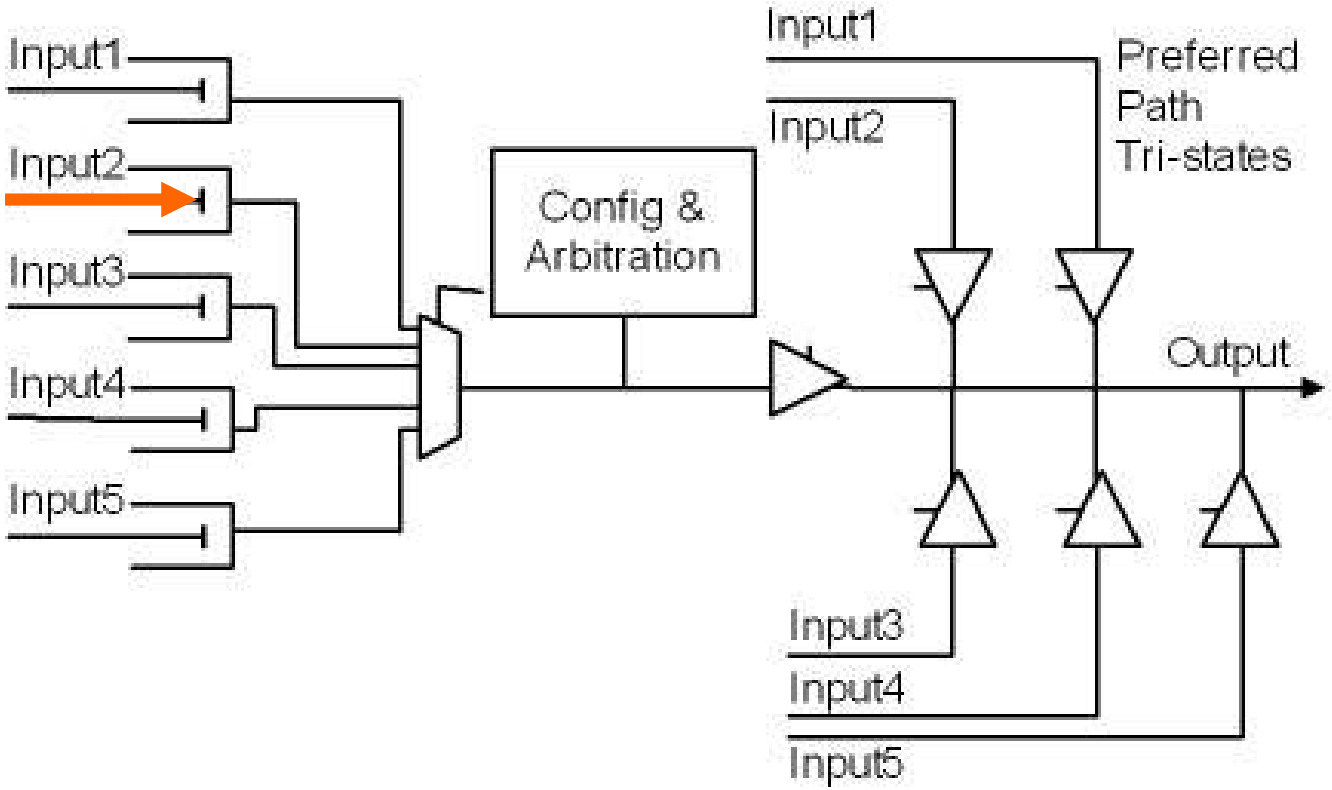
Switch Architecture - Output



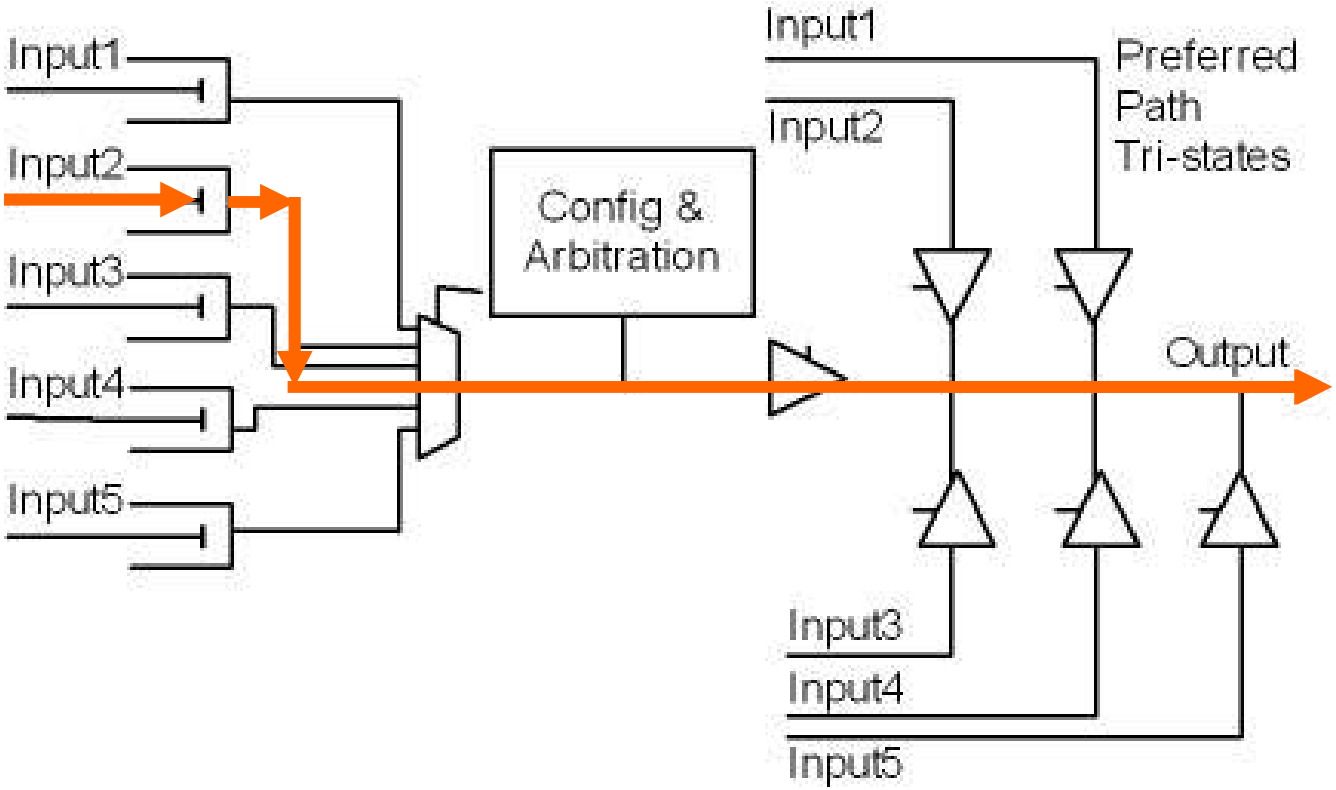
Switch Architecture - Output



Switch Architecture - Output



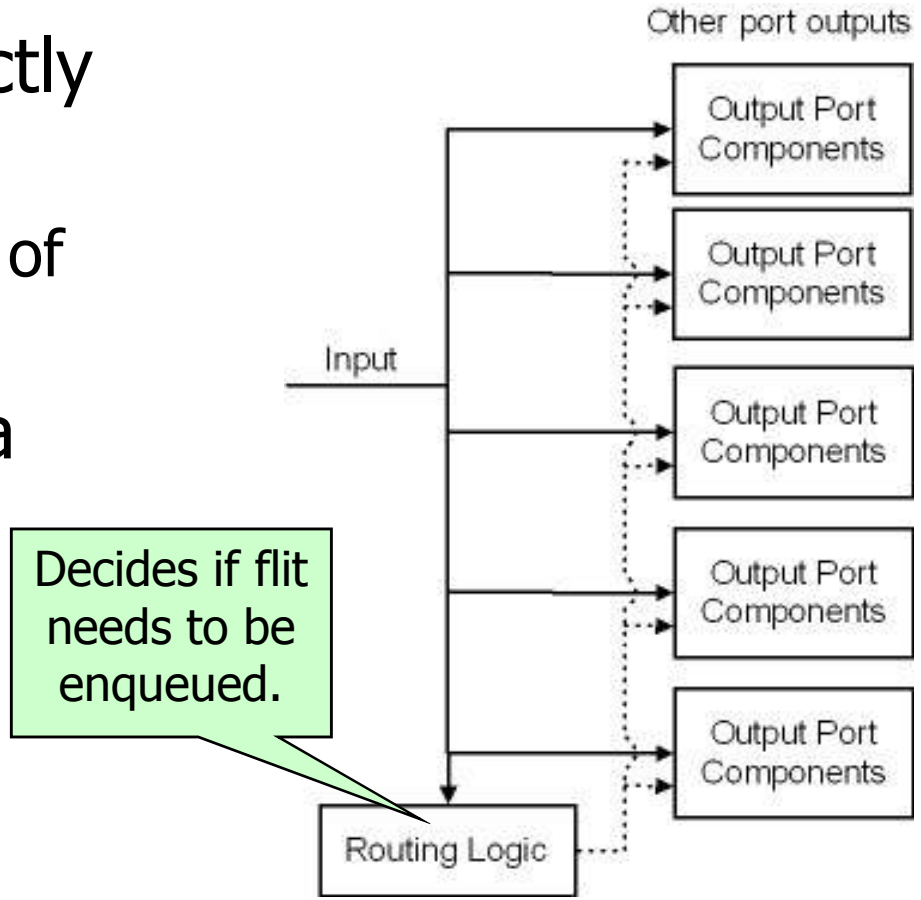
Switch Architecture - Output

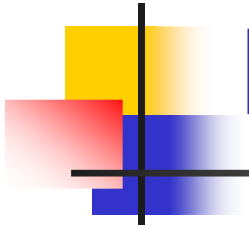


1 cycle

Switch Architecture - Input

- Dead flits: Incorrectly eagerly forwarded.
  - Terminated at end of preferred path.
- Switch resembles a buffered crossbar.



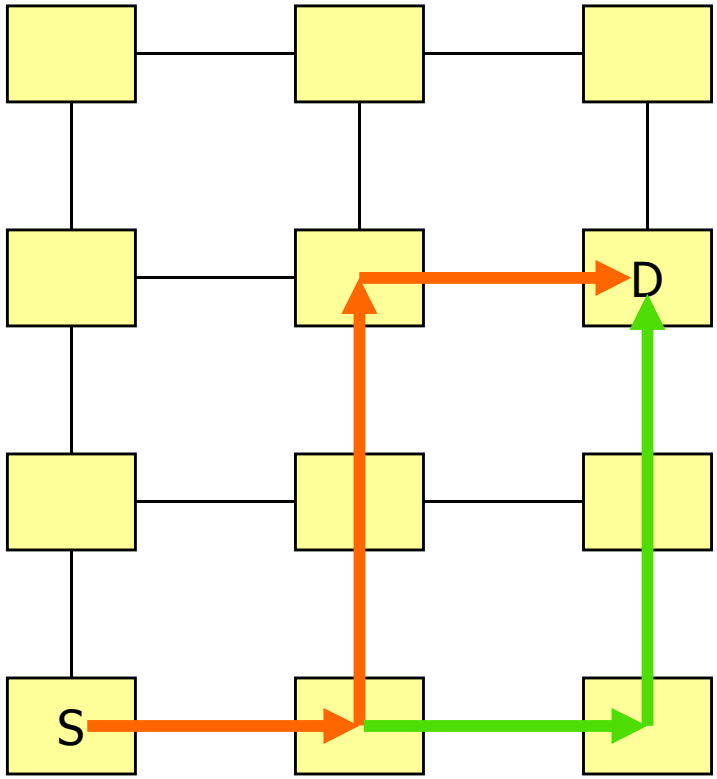


Routing Algorithm

- Deterministic routing employed.
- Non-preferred paths follow XY routing.
- We slightly modify XY routing to handle preferred paths:
  - Flit correctly eagerly forwarded if it approaches the destination in any axis.
  - Flit considered dead otherwise.

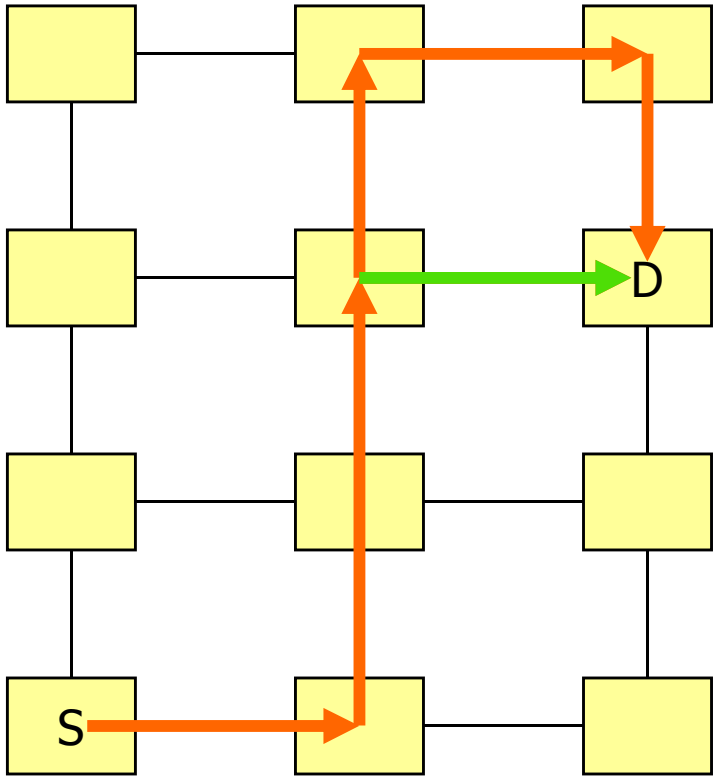
Routing Characteristics

- **Flits in preferred paths may not follow XY routing.**
- Duplicate copies of a flit may be delivered.
- XY routing.
- Pref. paths.



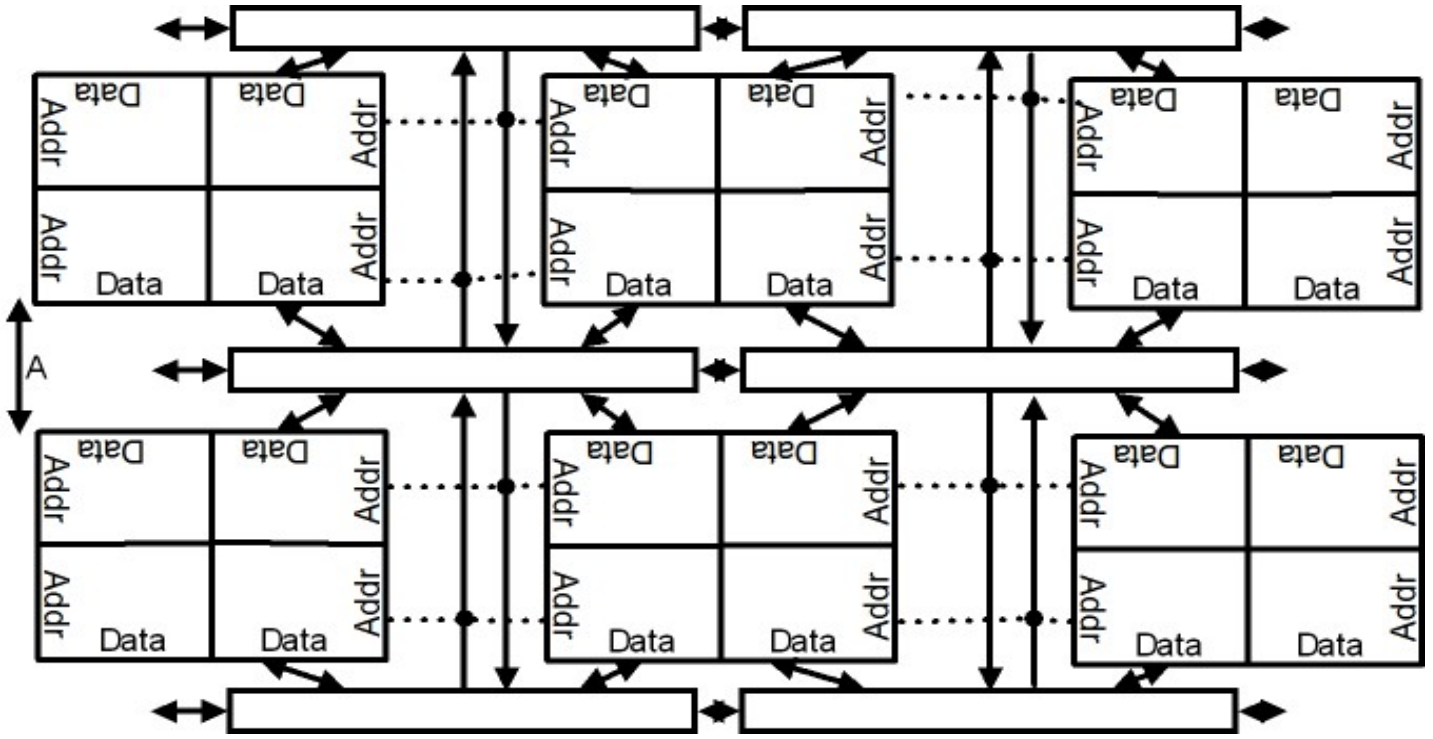
Routing Characteristics

- Flits in preferred paths may not follow XY routing.
- **Duplicate copies of a flit may be delivered.**
- XY routing.
- Pref. paths.

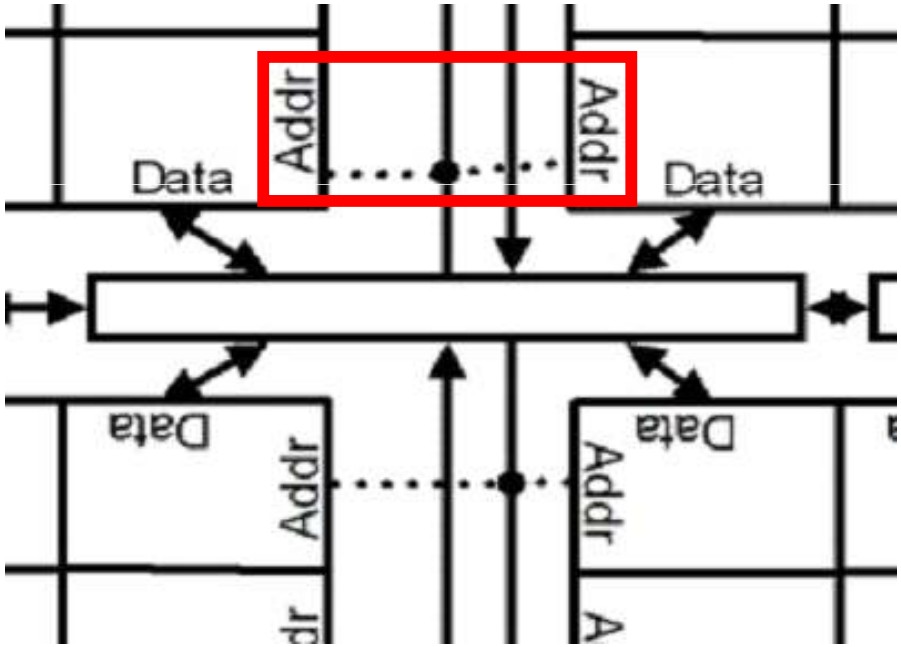


NoC Topology – Bar Floorplan

- Application: Tiled CPU and RAM blocks.
- Each switch is 6x6 and serves 4 PEs.

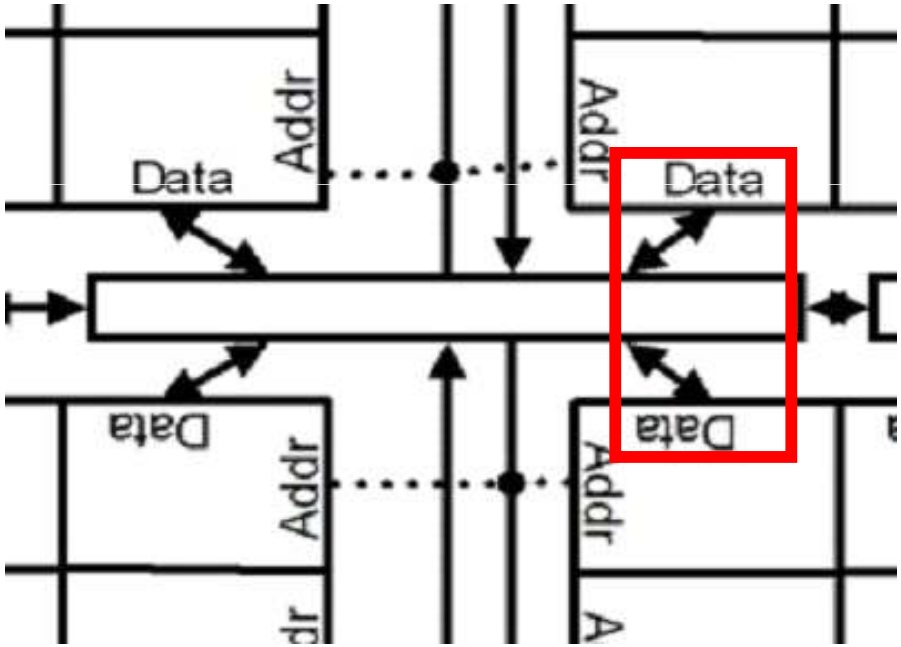


Bar Floorplan

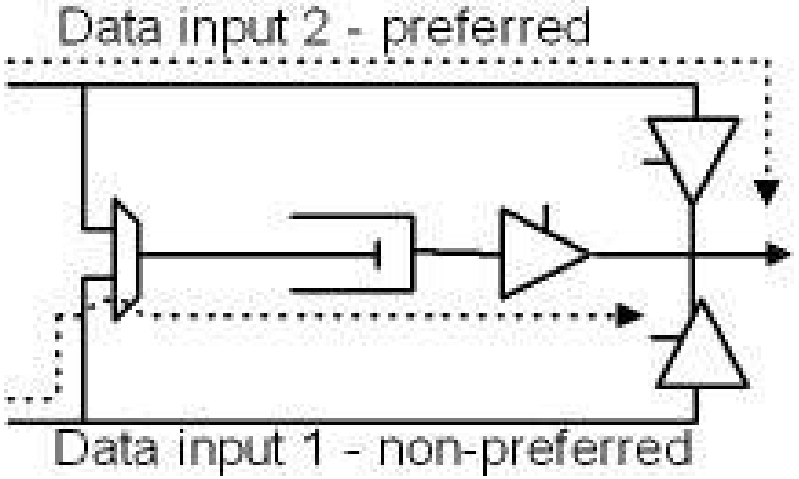


- Would be 8x12:
  - Vertical links drive address inputs.
  - 2 PE data ports served by 1 switch port.

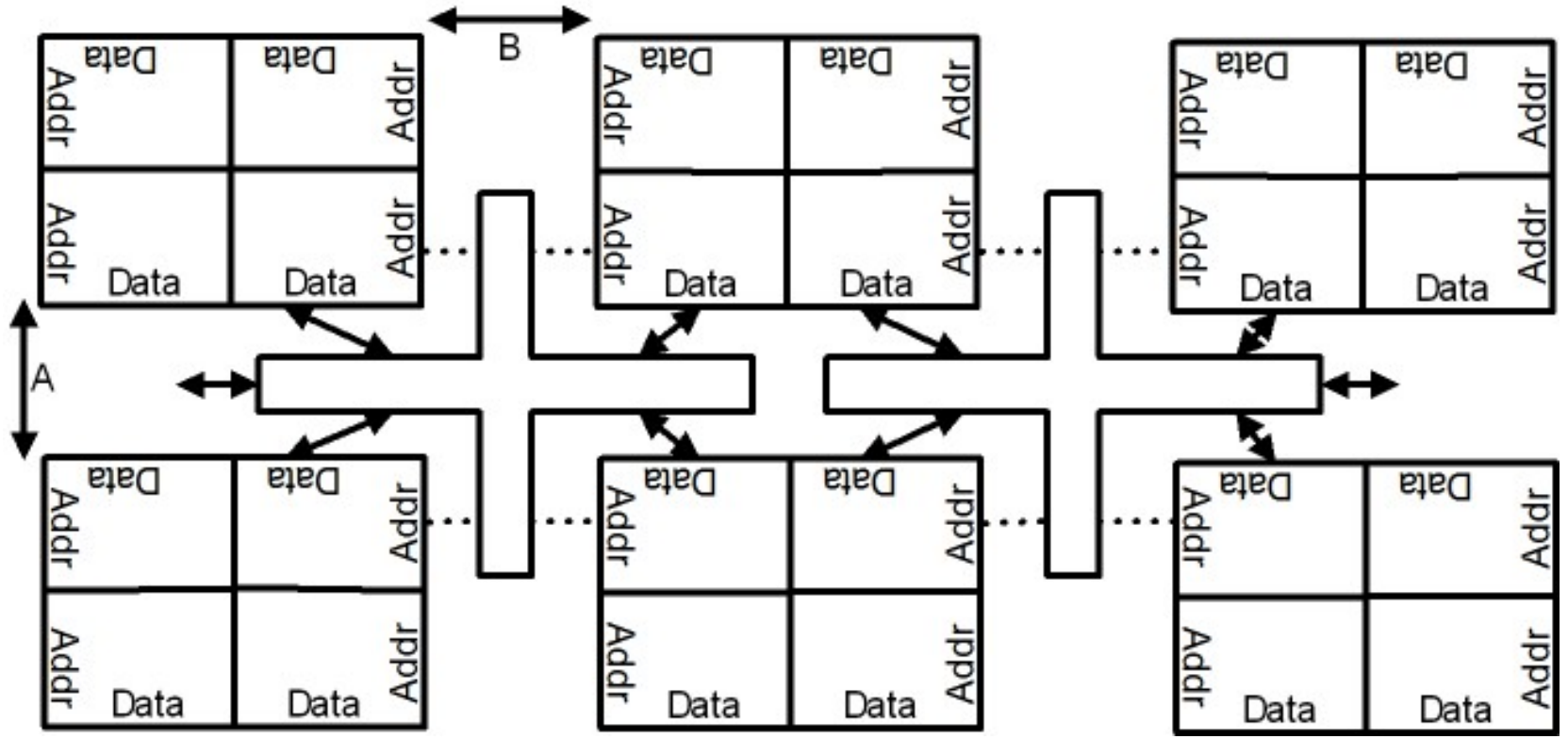
Bar Floorplan



- Would be 8x12:
  - Vertical links drive address inputs.
  - 2 PE data ports served by 1 switch port.



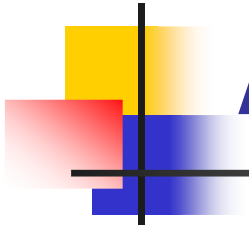
Cross Floorplan



Layout Results

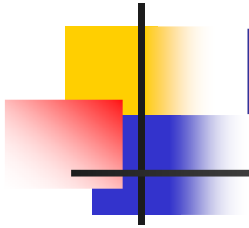
- 130 nm implementation library. Typical case.
- Pref. path latency:
  - 300-420 ps.
  - 450-500 ps (incl. 1mm).
- 1 cycle/node otherwise.
- Past work: 1 cycle/node at 500 MHz.

Clock frequency	667 MHz
Flit width	39
FIFO lines	2
Number of FIFOs	30
Bar area overhead	13%
Cross area overhead	18%
Number of cells	15 K
Number of gates	45 K
Total dynamic power	80 mW



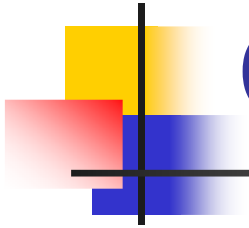
Advanced Issues

- Deadlock & livelock freedom.
  - Constraints to prevent circle.
  - Keep NoC functional in any case.
- Out-of-order delivery of flits in the same packet.
  - Apply reconfiguration at a “safe” time.
- Adaptive routing.



Future Work

- Synchronization issues – A flit may arrive at any time.
  - Impose preferred path constraints.
  - Implement switch asynchronously.
- Evaluation in complete system.
- Implement fault-tolerance.



Conclusion

- We approach ideal latency.
  - By pre-enabled tri-state paths.
- Our NoC is a generalized "mad-postman" [C. R. Jesshope et al, 1989].
- Our NoC is easily generalized – topology may need to be changed.
- Past NoC research can be applied for further optimizations.