Distributed Elastic Switch Architecture for efficient Networks-on-FPGAs

Antoni Roca, Jose Flich
Parallel Architectures Group
Universitat Politechnica de Valencia (UPV)
Valencia, Spain

Giorgos Dimitrakopoulos
Electrical and Computer Engineering Dept.
Democritus University of Thrace (DUTH)
Xanthi, Greece
dimitrak@ee.duth.gr

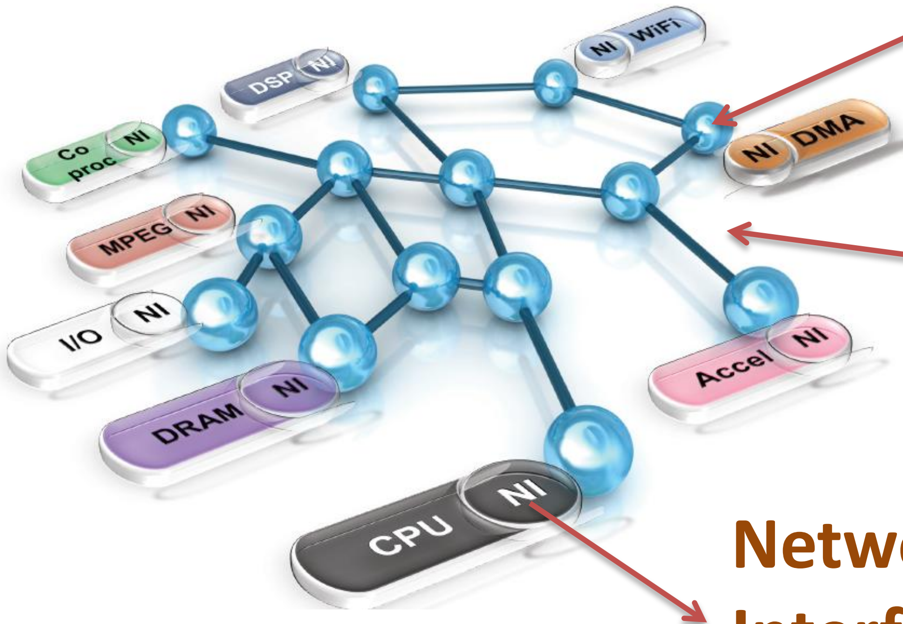
Outline

- Interconnecting System-on-Chips implemented on top of FPGAs
  - Network-on-FPGAs
- Constraints set by the FPGA architecture
- Elastic switch architecture
  - Step-by-step construction
  - Layout freedom due to distributed nature
- Experimental results
- Conclusions

The need for scalable soft interconnect

- FPGAs can host a complete system-on-chip – Platform FPGAs
- System integration and cores communication is an increasingly complex problem
  - Already well-defined abstractions (AXI) are used
  - Need scalability both at the physical level and at the logical-abstraction-level
- Apply networking principles to the on-FPGA environment
  - Idea already adopted in the ASIC domain
    - Can the basic architectures migrate to the FPGA environment?

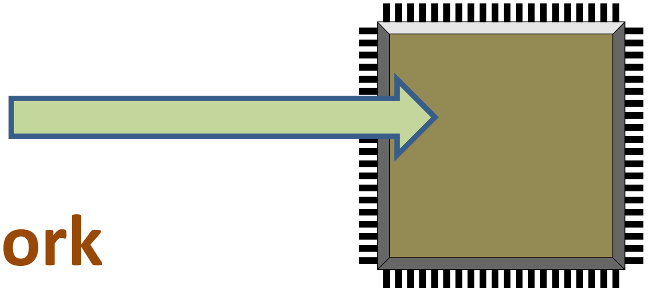
Network-on-FPGAs



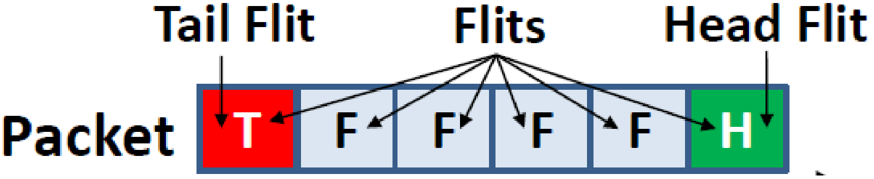
Switches

Links

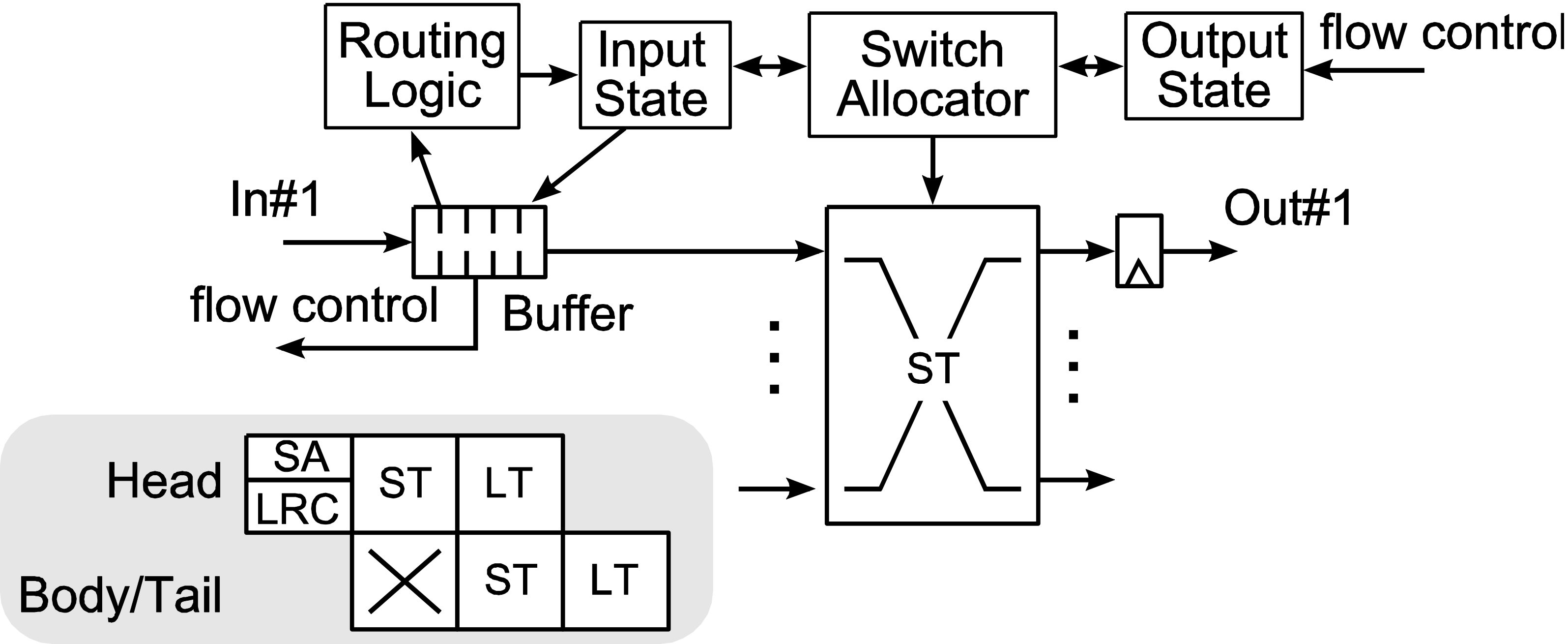
Network Interfaces



- The soft interconnection network is the glue that binds together the components of the system
- The NoC parallelizes communication using switches connected with point-to-point links
- Data are packetized and transmitted flit-by-flit on the links



Baseline switch architecture

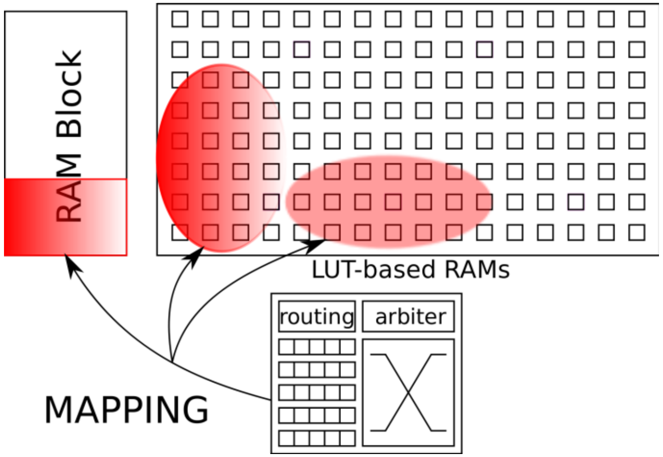


Head	SA LRC	ST	LT	
Body/Tail		✗	ST	LT

- Tasks can be implemented in a single cycle
- Merged arbiter multiplexer (FPL 2011) helps in this direction by unifying SA and ST

Mapping NoC Switches on FPGAs

- SA, ST and LRC are mapped on LUTs and FFs of FPGA
  - Delay affected heavily on high-radix switches
  - Word width does not increase delay much
- For buffers we have many options
  - Use RAM macros -> Poor Utilization of a scarce resource
  - Use Distributed RAM -> Better utilization-overhead tradeoff
  - Use FFs -> High overhead
- Abundant wiring for wide links

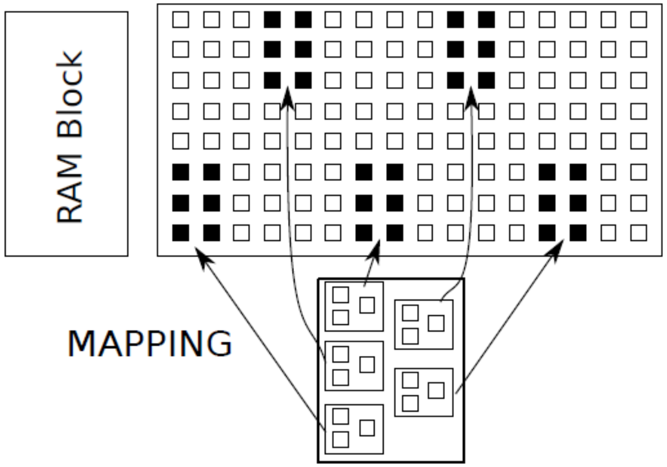


Initial analysis done by M. Papamichael, J. Hoe, ACM FPGA 2012

NoC Switch Design Guidelines for FPGAs

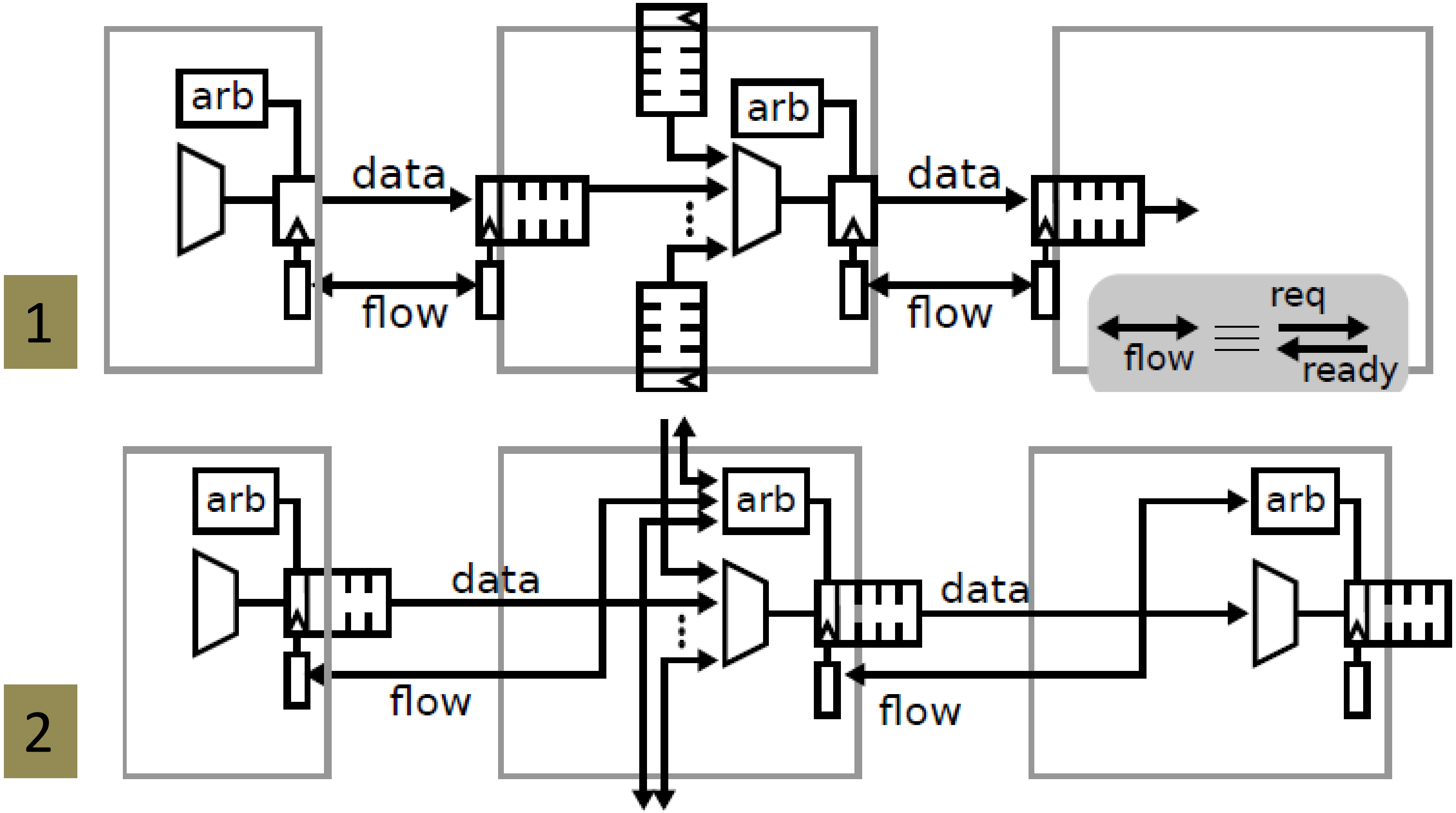
- **Utilize wide datapaths** taking advantage of the abundant wiring without violating the area constraints.
  - This approach benefits also packet latency by decreasing its serialization part
- **Use the appropriate radix** for the switches that does not limit the maximum clock frequency of the network.
  - As long as the clock frequency constraint is not violated high-radix switches can be built further reducing network latency
  - Balanced pipelines is a good option (state-of-the-art targets single cycle implementations)
- **Use cautiously the storage resources** for buffering without negatively affecting network throughput
  - State-of-the-art Uses LUT RAM for flit buffers
- **Allow any form of flexibility** in the implementation that would not stress too much the corresponding CAD tools

Main idea



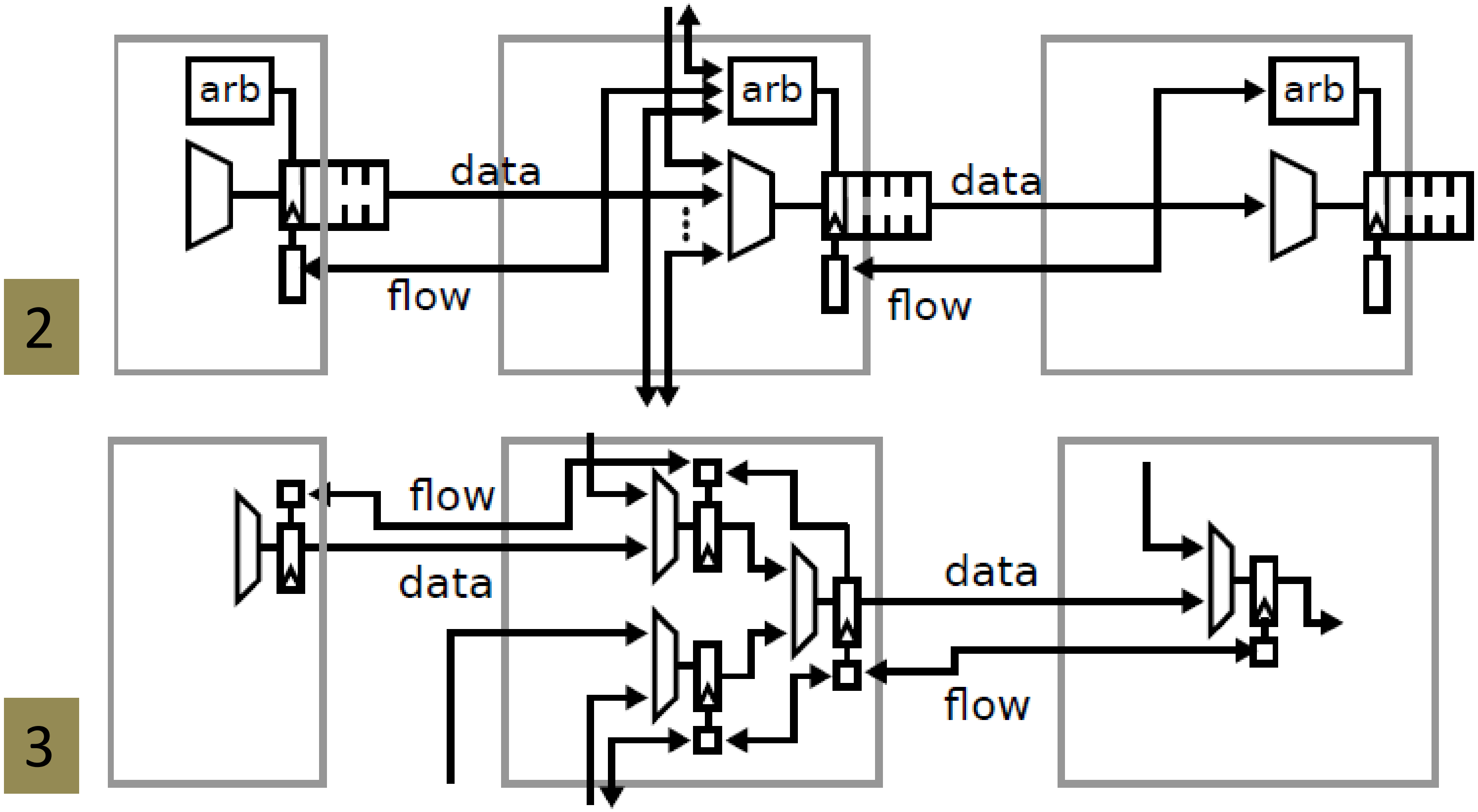
- Allow for fine-grained switch decomposition
  - Build switch from identical modules that operate independently
- Each module (called AC) can store data, arbitrate and switch data independently
  - Distributed switch
- Each AC uses is mapped only on the logic array (LUTs and registers) leaving any of the scarce RAM resource free to other modules of the system

AC-based switch – Step 1



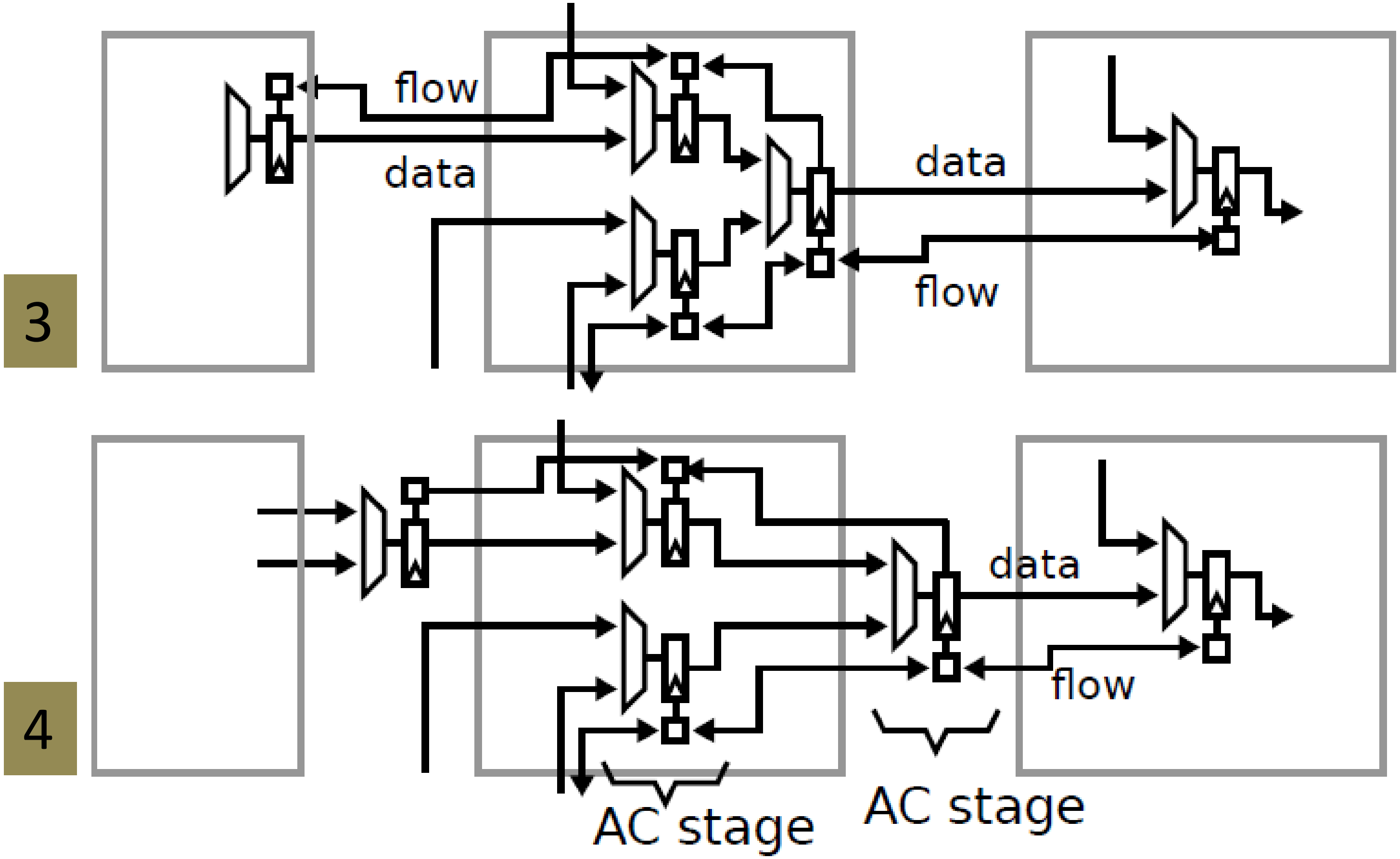
- Move input buffers to the upstream output port

AC-based switch – Step 2



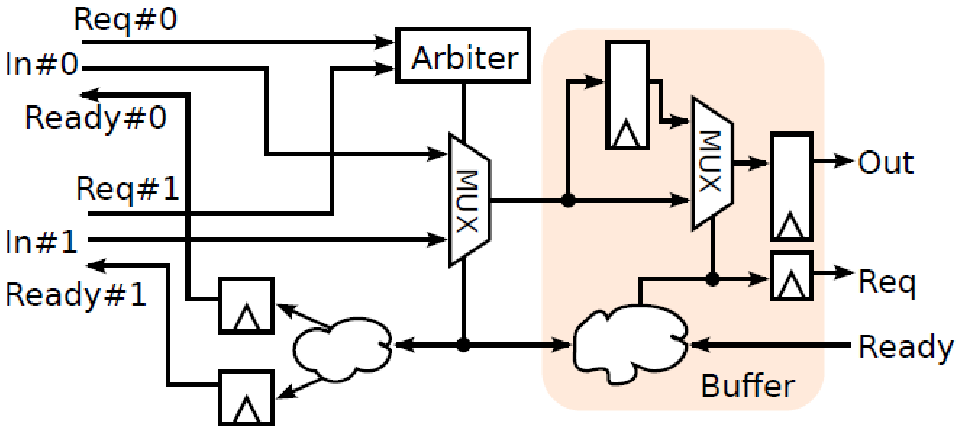
- Retime input buffers inside the crossbar’s multiplexers
- Arbitration done per stage

AC-based switch – Step 3



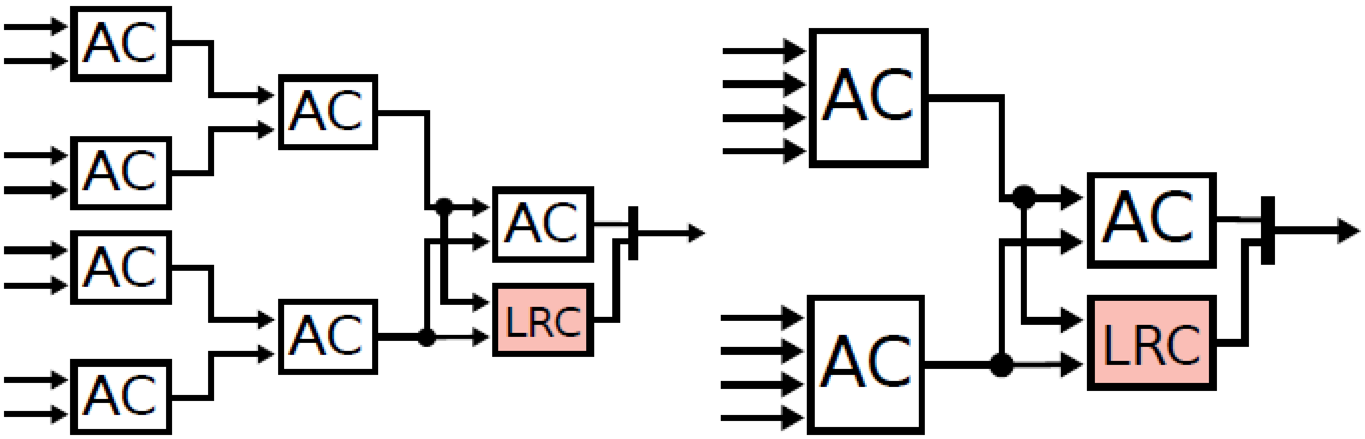
- Balance wire length
- Data transferred using an elastic request/ready pipeline

Elastic AC module



- Data transferred when request and ready are both true
- The two incoming branches are round-robin arbitrated
- Backpressure propagates only one cycle backwards
  - Two registers per output are needed to catch any in-flight data
    - Effectively the 2 registers form a 2-slot FIFO
    - More registers can be added to act as a larger FIFO if needed for performance
  - One register per elastic stage is possible but reduces the throughput to half

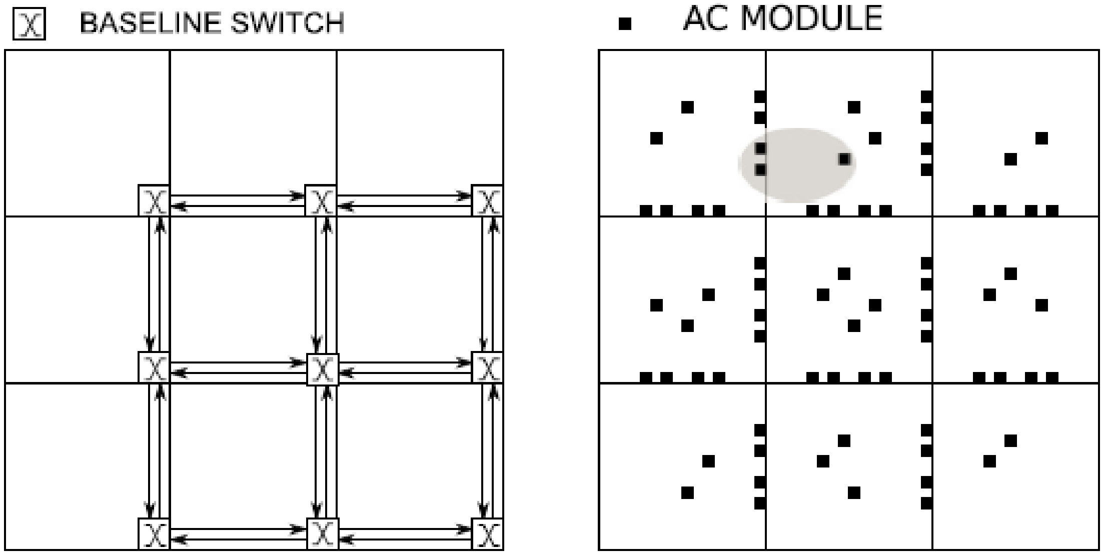
AC radix: Latency vs Buffering



(a) With radix 2 AC modules (b) With radix 4 AC modules

- Latency affected by clock freq. and number of cycles to cross a switch
  - High-radix AC modules reduces the number of cycles per switch but increase clock frequency
    - Connecting wires become longer
- High-radix AC modules require less the buffering

Distributed Layout



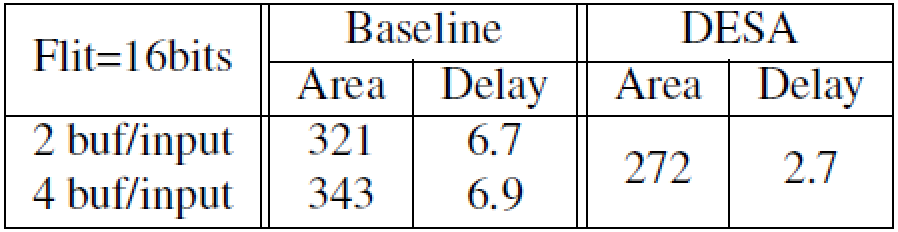
(a) Floorplan with standard switches.

(b) Floorplan with AC-based switches.

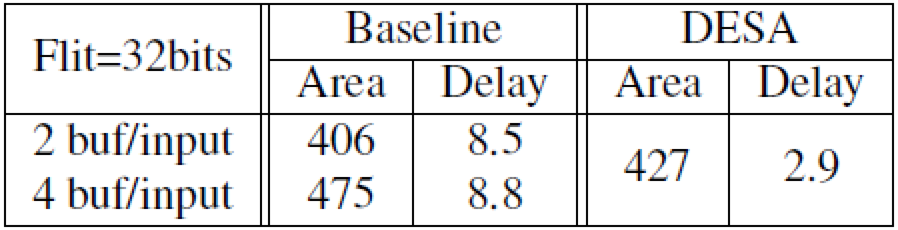
- Small pressure to the place & route CAD tools
- Even if starting from a regular topology the final placement can be arbitrary without any impact on clock frequency

Results

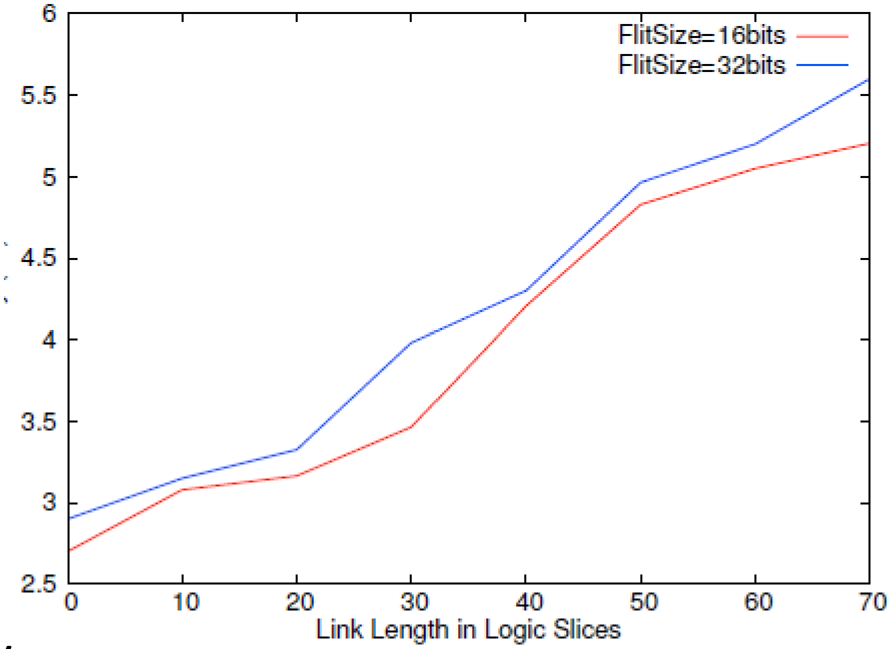
5x5 switches with DOR for a 2D mesh on a Virtex 5 FPGA



Flit=16bits	Baseline		DESA	
	Area	Delay	Area	Delay
2 buf/input	321	6.7	272	2.7
4 buf/input	343	6.9		



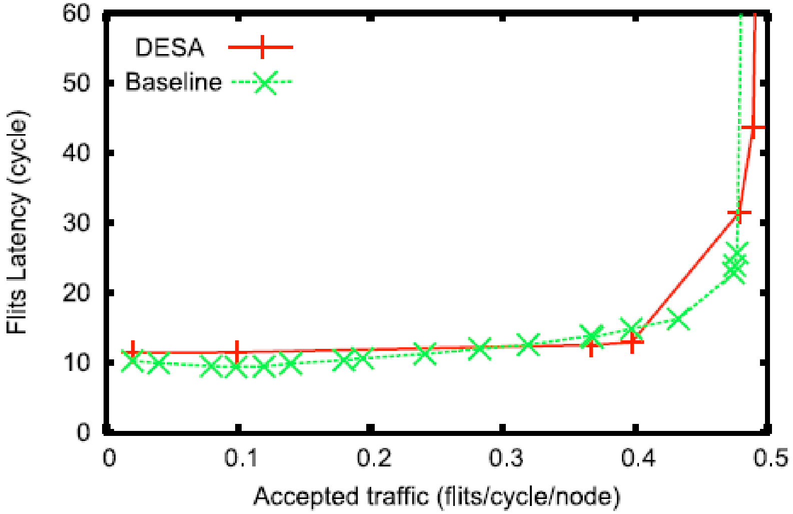
Flit=32bits	Baseline		DESA	
	Area	Delay	Area	Delay
2 buf/input	406	8.5	427	2.9
4 buf/input	475	8.8		



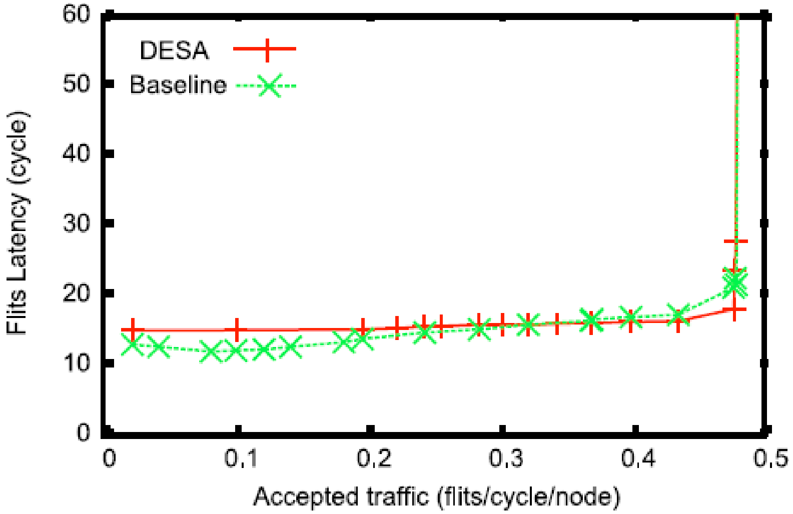
- DESA offers smaller area and delay
  - Delay benefits come from pipelining and from simple logic per pipeline stage
  - The delay in each stage is mostly determined by the delay of the links
  - If we pipeline the baseline for speed this will increase RTT and will add significant area overhead due to the extra buffers needed to cover RTT
- DESA delay linearly increases with the link distance between AC modules

Network-level performance

- Network performance on a 4x4 2D mesh
  - Uniform and Bit-complement traffic
- Both switches have equal numbe of buffers
- Baseline slightly lower latency at low loads
  - DESA needs 2-cycles per switch but a 2x the clock frequency
  - So absolute latency in ns is better or the same
- Same saturation throughput
  - DESA has 2x the clock frequency which means 2x absolute throughput



(a) Uniform-random traffic



(b) Bit-complement traffic

Conclusions

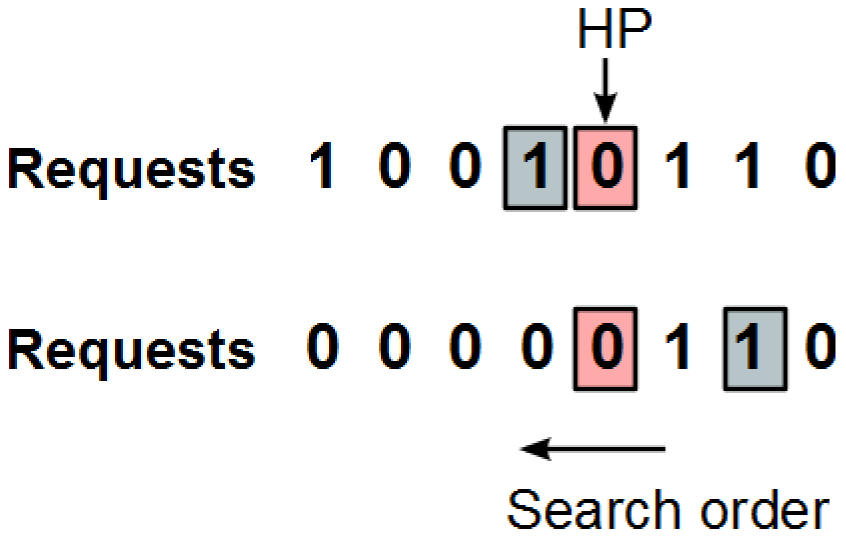
- Optimize the automatic mapping of networks on FPGAs using a distributed elastic switch architecture
  - Avoids using any of the scarce memory blocks
  - Reduces the effects of long links
- The AC module reduces significantly the per-switch latency while keeping network throughput un-affected.
  - Area savings are also noticeable
- AC radix provides a tradeoff between latency, speed and buffering

Backup slides

- Complementary to FPL 2011 paper

Round-robin arbitration

- Most widely used selection policy
- Arbitration begins at the highest-priority position (HP) and searches for the first active request
- The search moves in a cyclic manner and covers all positions



HP

Requests 1 0 0 1 0 1 1 0

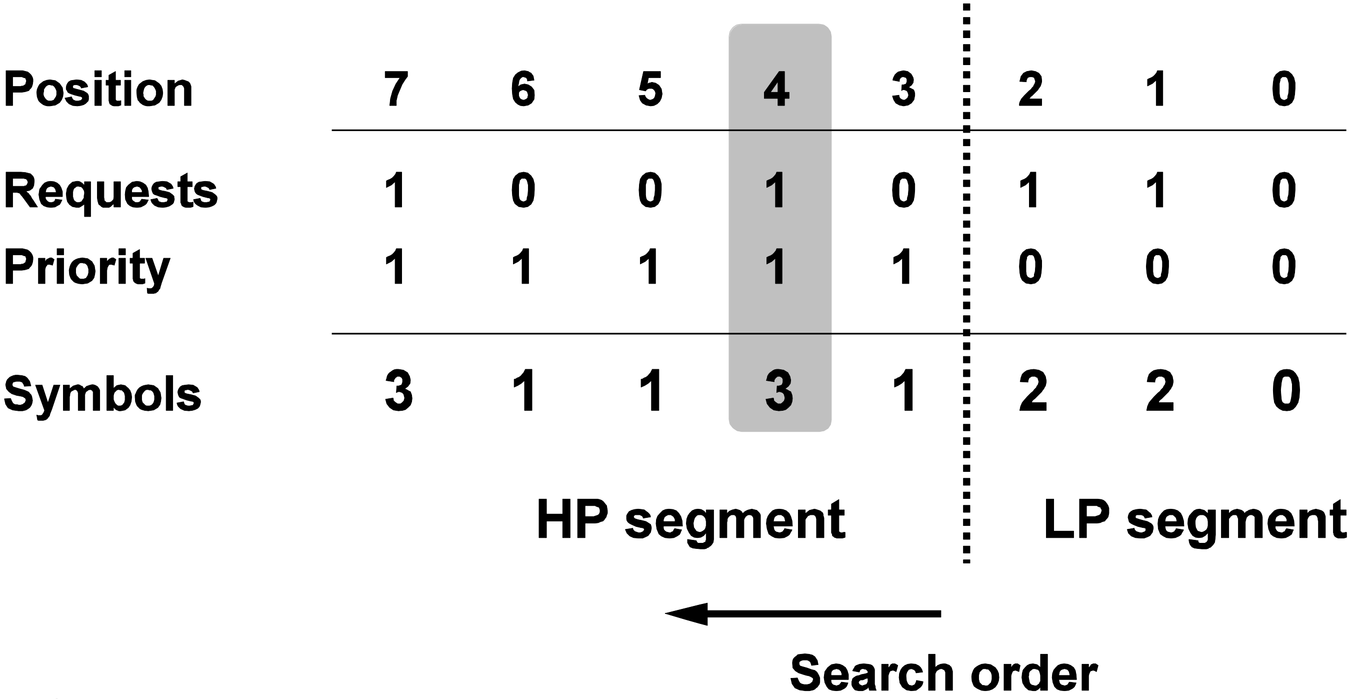
Requests 0 0 0 0 0 1 1 0

Search order

- The request that won receives the lowest priority in the next cycle

Requests-Priority recoding

- Transform each request and priority bit to a 2bit unsigned arithmetic symbol
  - The request the MSBit
- Arbitration equivalent to sorting (max selection)



Position	7	6	5	4	3	2	1	0
Requests	1	0	0	1	0	1	1	0
Priority	1	1	1	1	1	0	0	0
Symbols	3	1	1	3	1	2	2	0

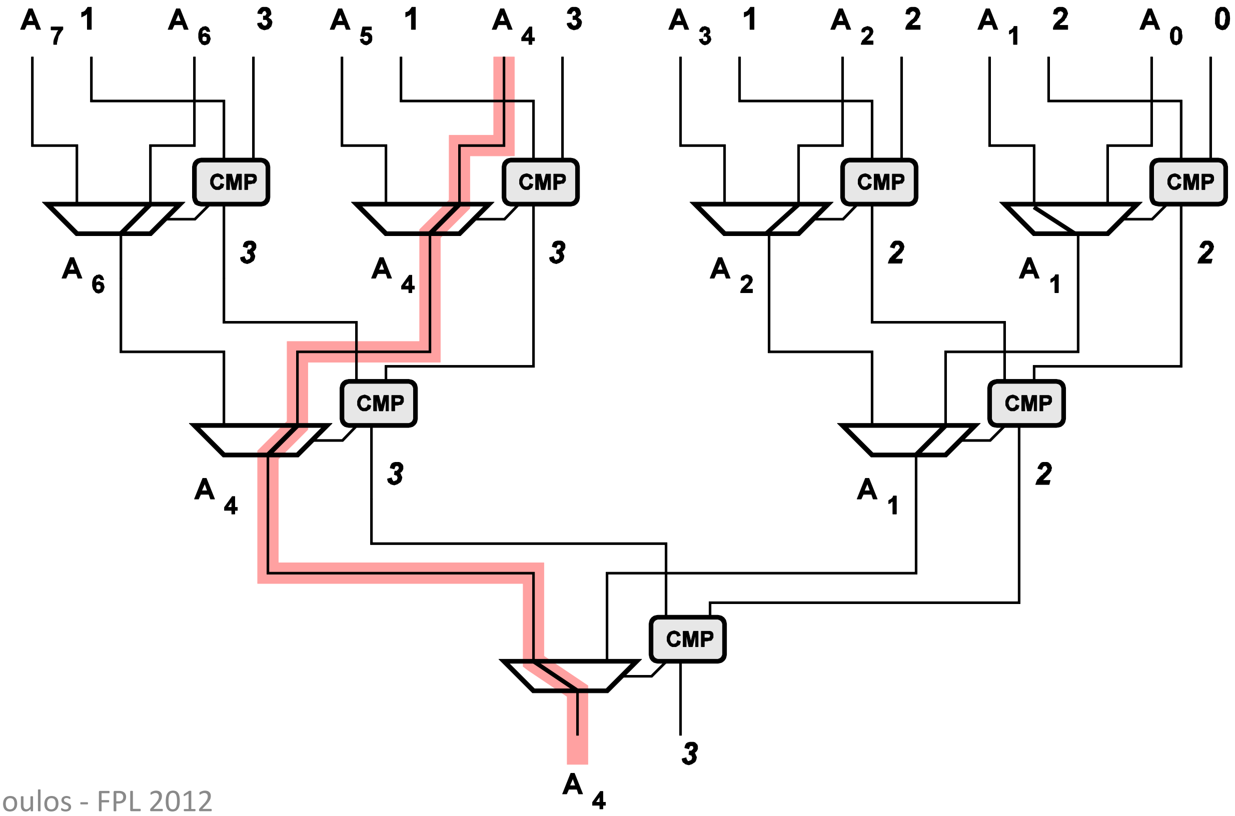
HP segment

LP segment

Search order

Merged arbiter multiplexer

Data	A7	A6	A5	A4	A3	A2	A1	A0
Requests	0	1	0	1	0	1	1	0
Priority	1	1	1	1	1	0	0	0
Symbols	1	3	1	3	1	2	2	0



A_7 1
A_6 3
A_5 1
A_4 3
A_3 1
A_2 2
A_1 2
A_0 0
CMP
CMP
CMP
CMP
A_6
3
A_4
3
A_2
2
A_1
2
CMP
CMP
A_4
3
A_1
2
CMP
3
A_4