Pitfalls and Triumphs in FPGA Design: A Switch Fabric Perspective

Oct. 26, 2004

Dr. Itamar Elhanany

College of Engineering
Dept. of Electrical & Computer Engineering
The University of Tennessee

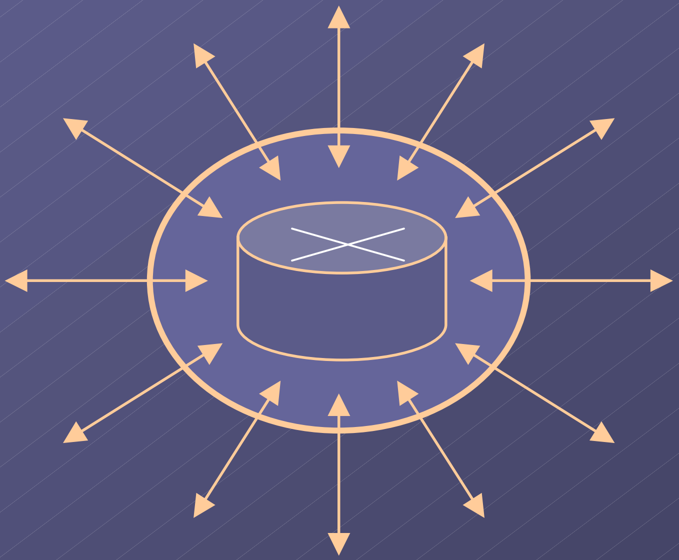
Fall 2004

Outline

- What is a switch fabric?
  - Motivation
  - What makes a router tick?
- Topics in designing with FPGAs
- Test benching
- Summary

What is a Router ?

- System which exchanges packets flowing from input ports (sources) to output ports
- Analogous to a traffic junction
  - Packets ←→ cars
  - Links ←→ lanes
  - Scheduler ←→ traffic light sys.

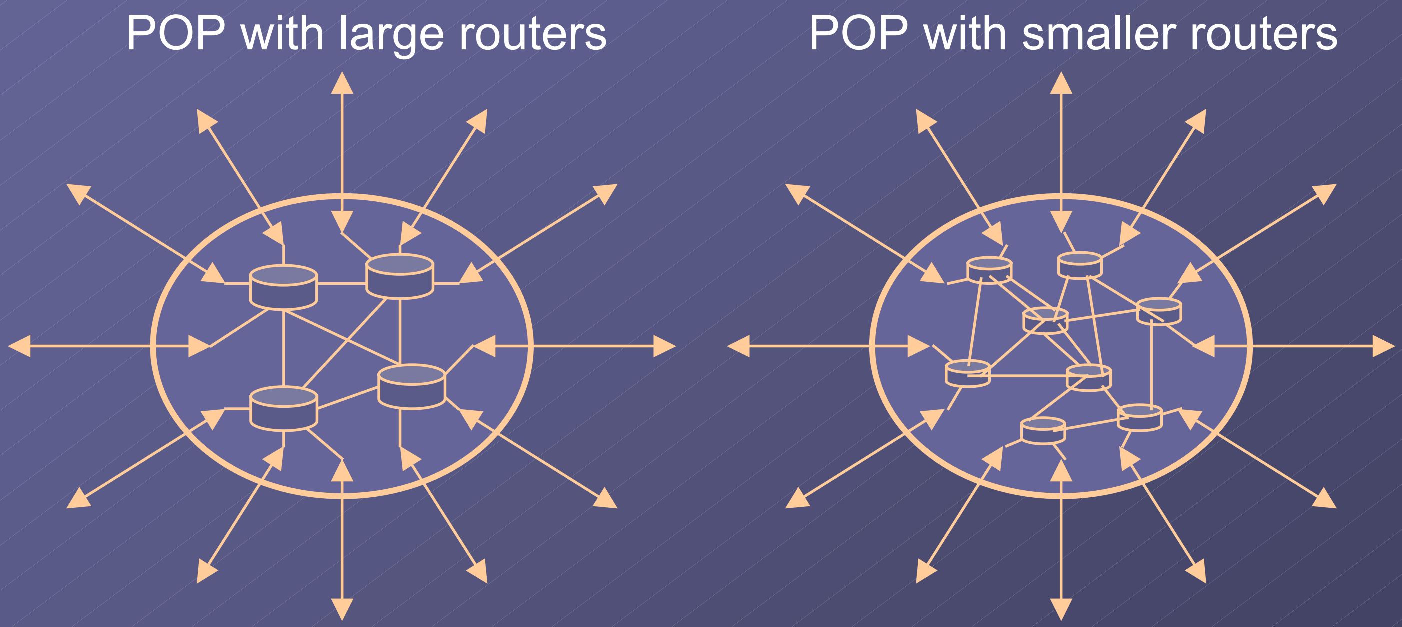


Challenges in Switch Fabric Design

- High port densities – 100's and even 1000's of ports
- Increasing data rates – 2.5Gbps, 10Gbps, 40Gbps,…
  - Packet durations ~50 nsec
- Introduction of optical building blocks
- Demand for low latency
- Demand for QoS provisioning – differentiated services

The strive for larger and faster routers

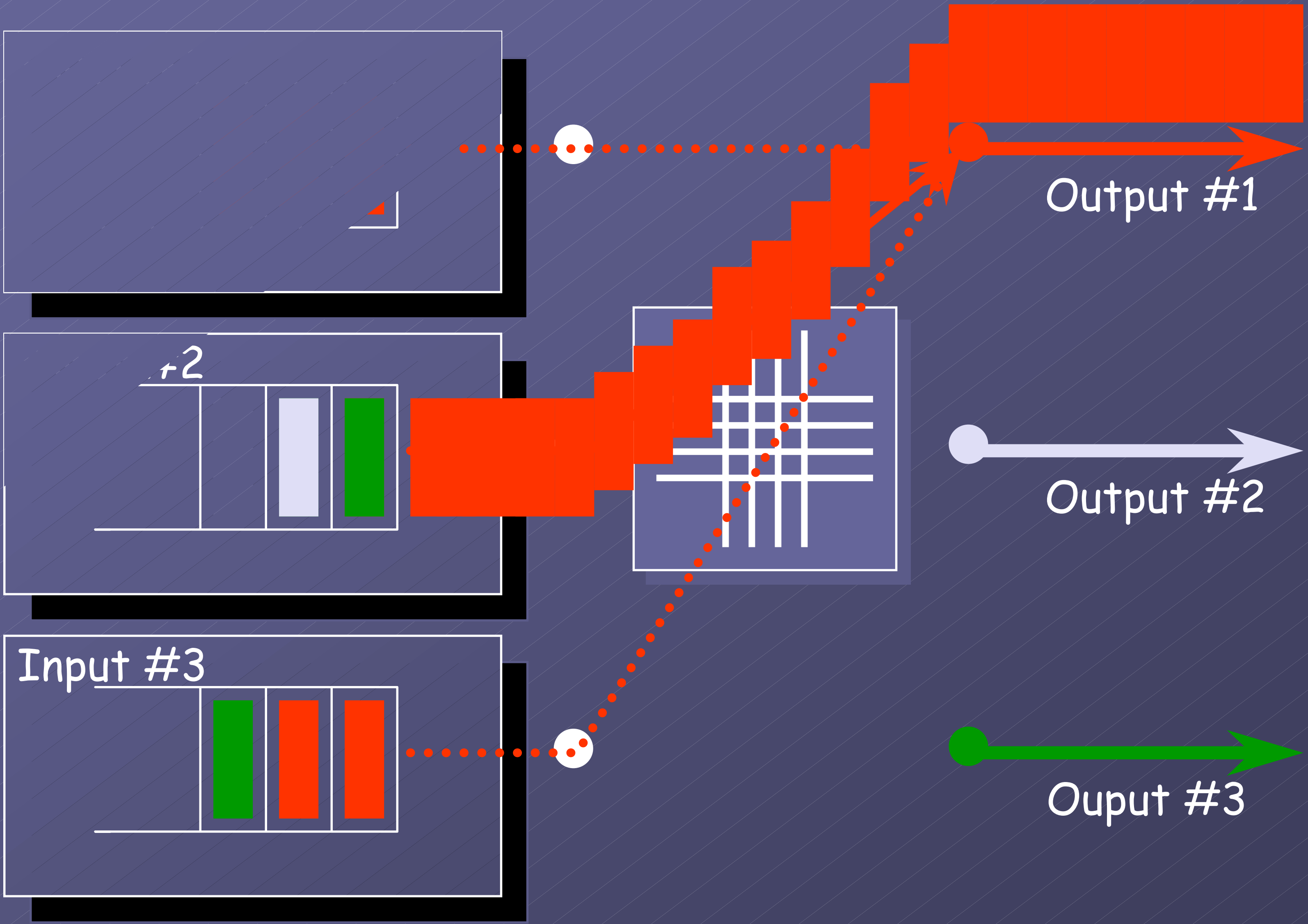
- Big POPs need big routers



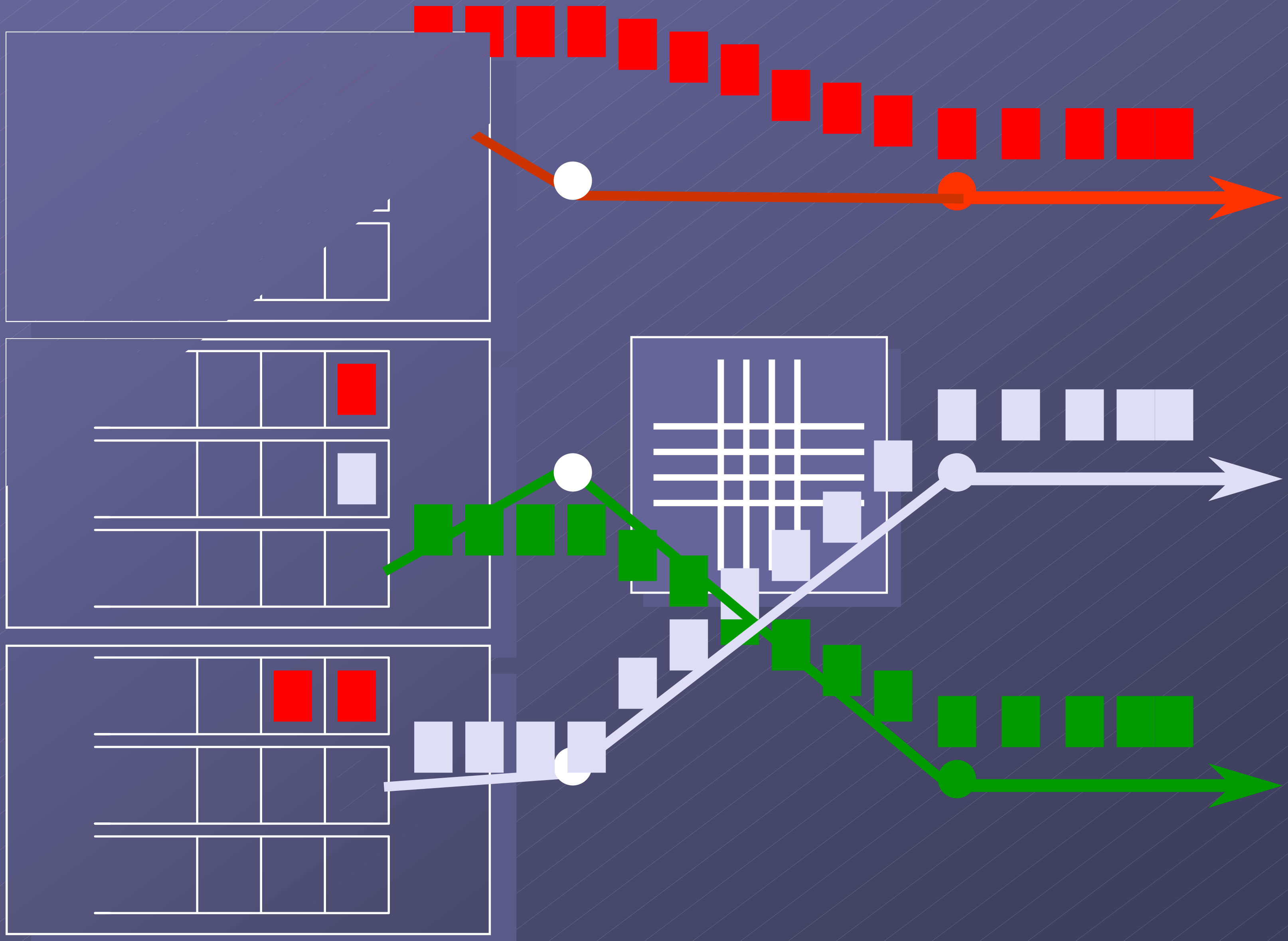
- Power and price considerations
- About 50-60% of interfaces are used for interconnection within the POP
- Industry trend is towards large, single router per POP

Source: Stanford University

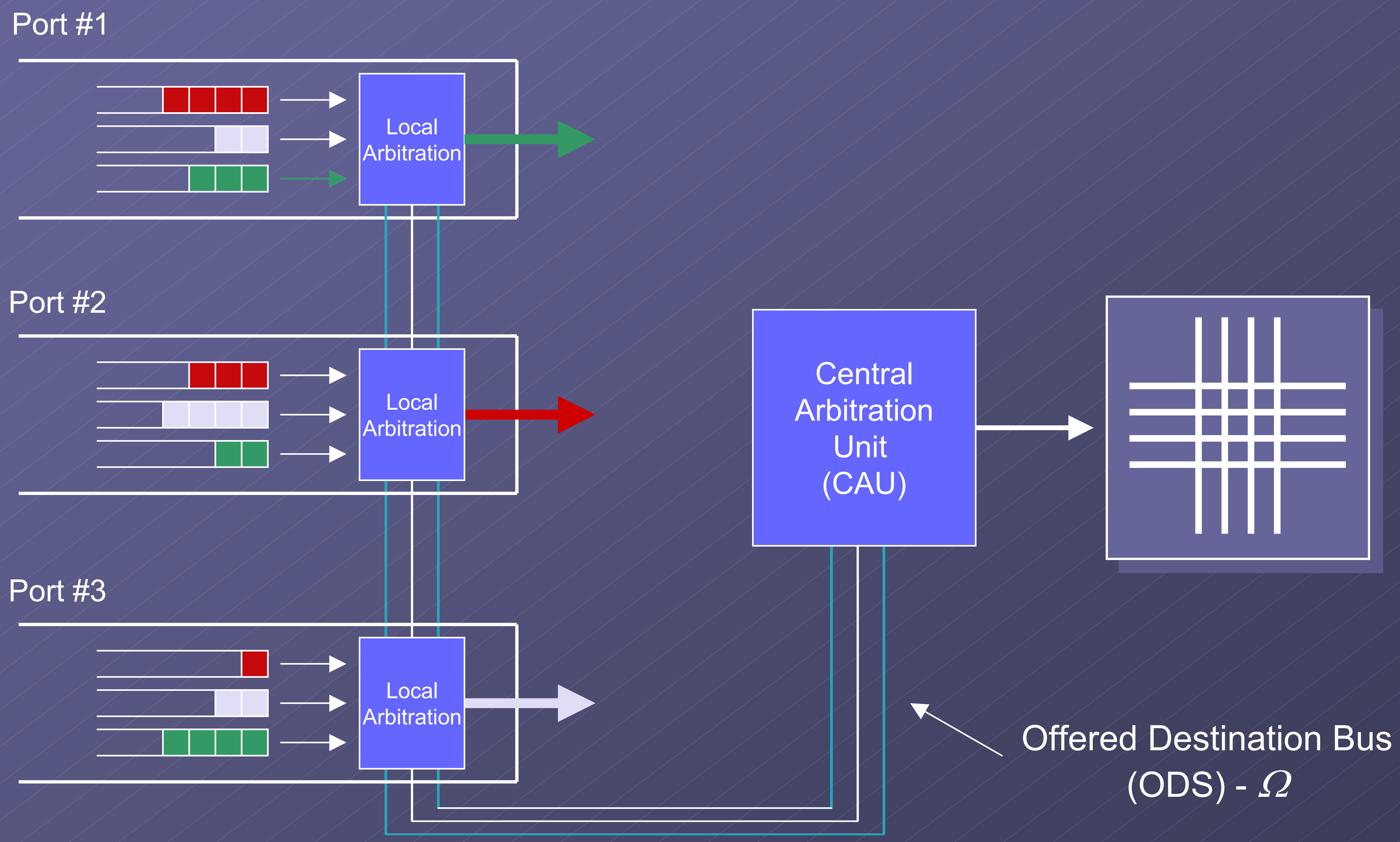
Head of line (HOL) blocking



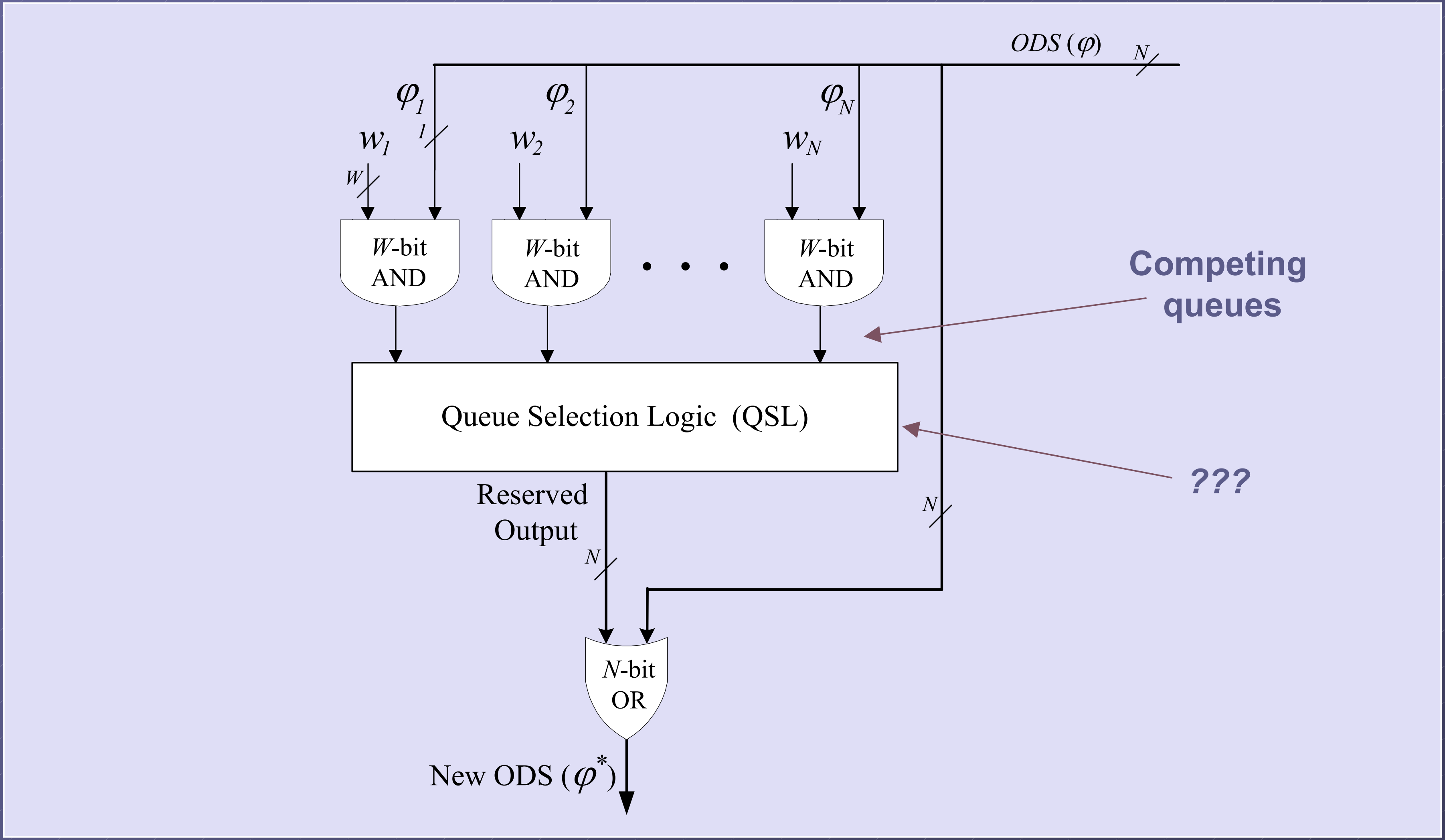
Overcoming HOL blocking (VOQ cont.)



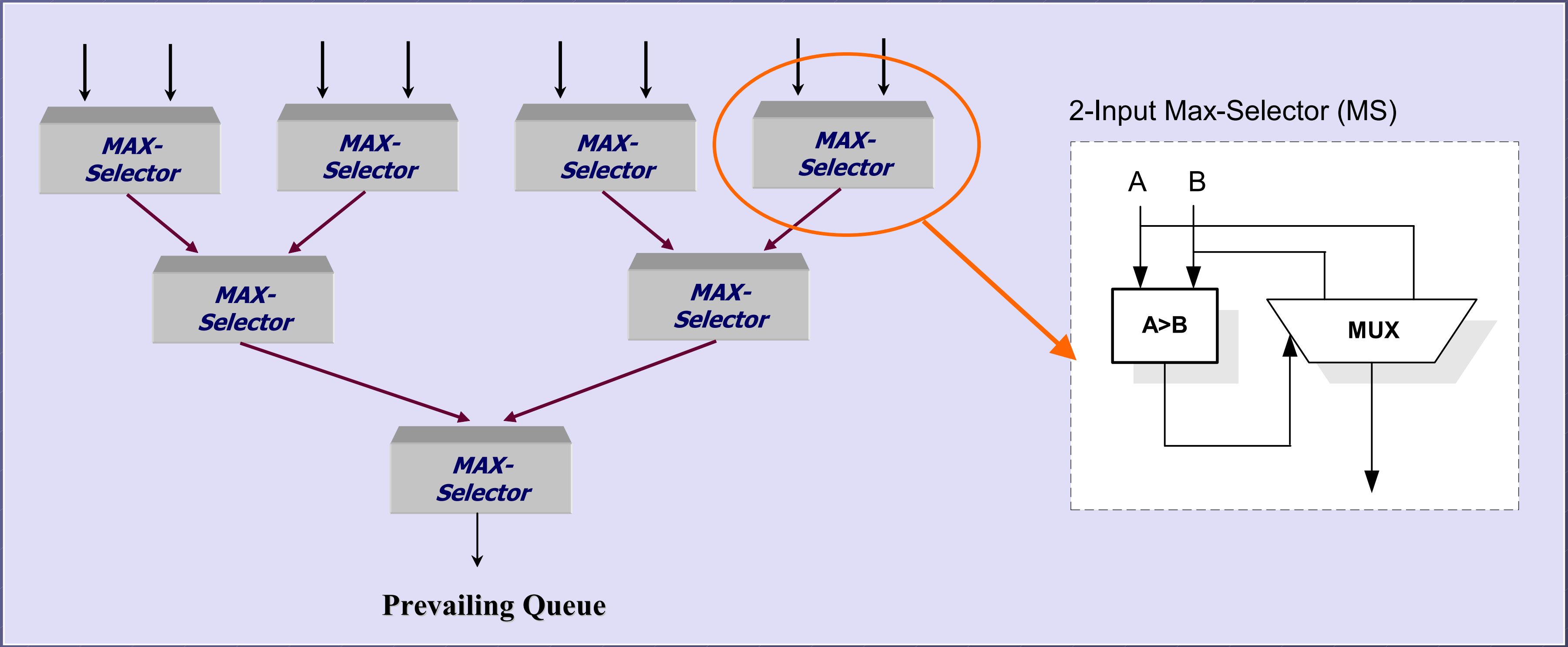
DISA Switch Architecture & Scheduling Algorithm



Implementing DISA: output reservation/allocation logic



Queue selection logic / longest-queue-first (LQF)



- Latency = $\log_2 N \cdot t_c$ where t_c is the MS propagation delay
- Can we make this better for larger number of ports?

Conclusions so far …

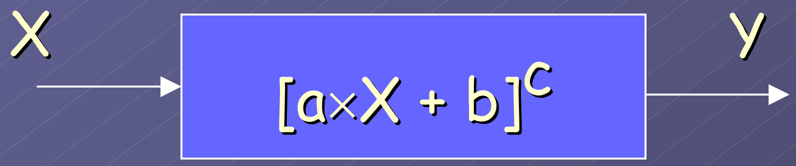
- Parallelization is the key to success
  - Time/Space tradeoff
  - Accelerate the design
- Use LUT (ROM) instead of logic – whenever possible
- Current FPGA technology -> 6 to 1 Boolean function is extremely fast

Outline

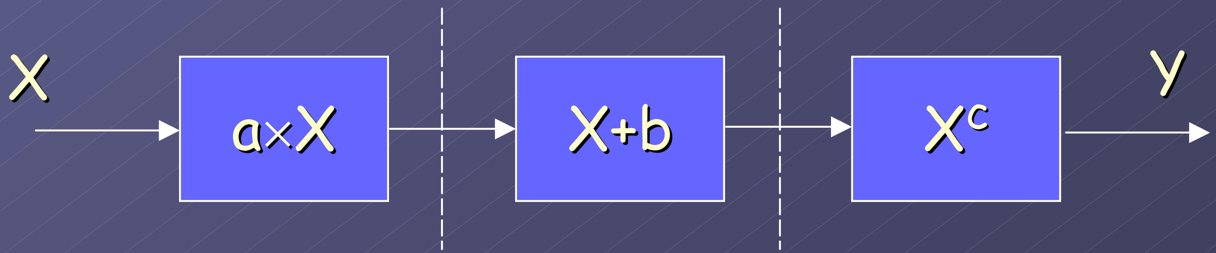
- What is a switch fabric?
- Topics in designing with FPGAs
  - Pipelining
  - Intelligent Testbenching
  - ILA
  - Layout optimization
  - General hurdles and tips
- Test benching
- Summary

Pipelining

- Partitioning of combinational logic to sub-segments that can be performed simultaneously (hence are independent)
- Example: $Y = [a \times X + b]^c$



- If the process takes 3T then our max clock rate is 1/3T
- In a pipelined architecture …



- Here our clock rate can be 1/T at a 2T delay cost !!

Pipelining

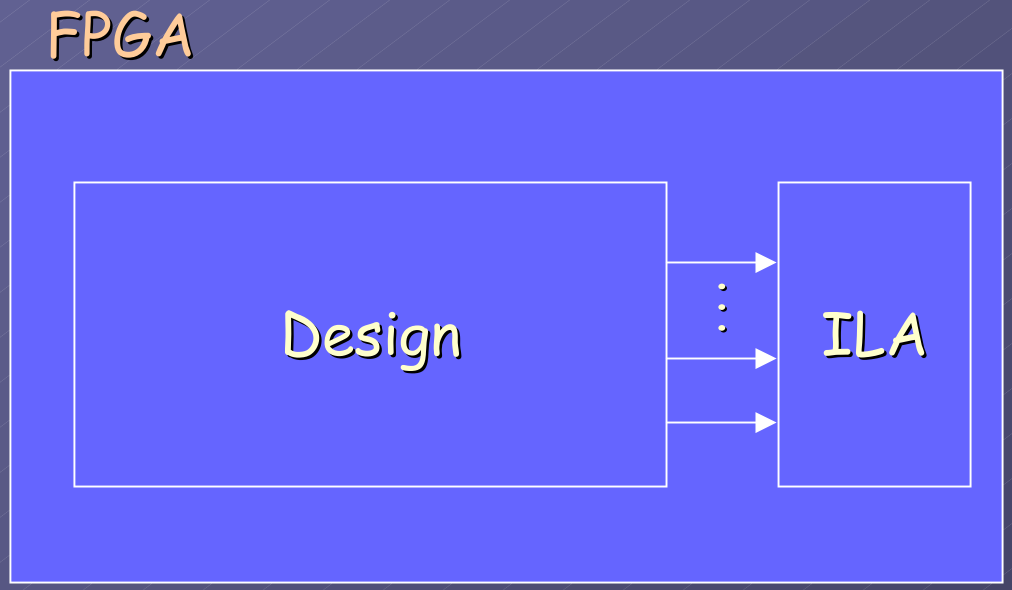
- Pipelining should be exploited wherever possible
  - Not always possible – but should be explored
- Key to a good pipelined design → balancing the prop. delay of each phase
- Requires repeated attempts – especially with FPGA when results are sometimes unpredictable

Importance of Design Verification

- Industry fact: 2:1 ratio between design/verification engineers
- Start out as a verification engineer – better understanding
- Complex testbench environments
  - File management
  - Statefull (state machine based) testing
  - Hours/days to run at a time
- Key to the success of a product

Internal Logic Analyzer (ILA)

- Xilinx terminology, also offered by Altera
- Allows us to monitor signals within the chip in real-time



- We need to define "trigger signals"
  - Once activated – stores data in RAM blocks and sends it out to the screen
- Supports complex conditioning

What kind of memory do I want synthesized ?

- Assume a synchronous design (clock rising/falling edge sensitive)
- Two primary types of R/W memory
  - Random access (RAM – RAM Blocks)
  - Sequential (Shift-Register – FFs)

```
if clock'event and clock='1' then
        …
        out <= mem(1);
        …
        mem = x & mem(7 down to 1)
        …
end if;
```

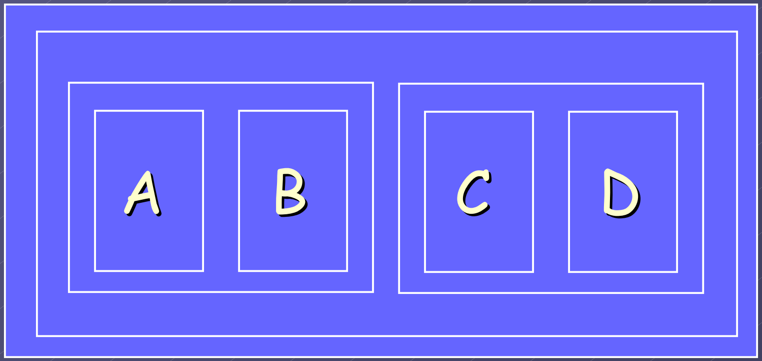
```
if clock'event and clock='1' then
        …
        out <= mem(n);
        …
        mem = x & mem(7 down to 1)
        …
end if;
```

Relative Location Constraints (RLOC)

- RLOC: allow you to define the location of any element relative to other elements in the set, regardless of eventual placement in the overall design

- Floor-planning – placement of the design in target FPGA
- For most FPGAs – hardware is much better than software
  - We need to help the tools
  - P&R critical in terms of latency performance
  - The designer knows the best layout strategy

FPGA



Writing structural vs. behavioral

- Ease of coding ←→ performance gain
- The more structural you write the higher the chances it will be synthesized optimally
  - Use gate-level statements and primitives when possible
- for/while – replicates hardware – use carefully!! (OK for testbenching)
- for generate – powerful replication tool

General hurdles and tips

- Delay
  - Capacitance is the enemy
  - In/out of a chip (IO pads)
  - through gates
  - fan-out is not free
  - So … always look at the synthesis result for improvement (critical path usually the place to start)
- Clock Nets
  - Low skew lines within the FPGA (distance-equal)
  - 16 in large chips
  - Use them well
- Spend a lot of time on your design – it will pay off !!

Summary

- There is an art to VLSI design (and verification)
- Engineering is about making tradeoffs
  - Spend a lot of time thinking about the design!
  - You will need to know what are the tradeoffs you can apply to provide a robust design which offers value to the user/customer
  - Make sure you test your design as much as possible – a design that breaks (as fancy as it may be) is worthless !
- The hardware is great, but the software tools need our help as designers
- If you're interested in hardware design of SF take ECE-692 (Spring 2005)

Questions ?