An FPGA-based Prototyping Platform for Research in High-Speed Interprocessor Communication

V. Papaefstathiou, G.Kalokairinos, A.Ioannou, M.Papamichael, G.Mihelogiannakis, S.Kavadias, E.Vlahos,

D.Pnevmatikatos and M.Katevenis

Inst. of Computer Sci. (ICS) – FORTH – Crete, Greece

Presented by:

M. Katevenis

FPGA-based Prototyping

Purpose:

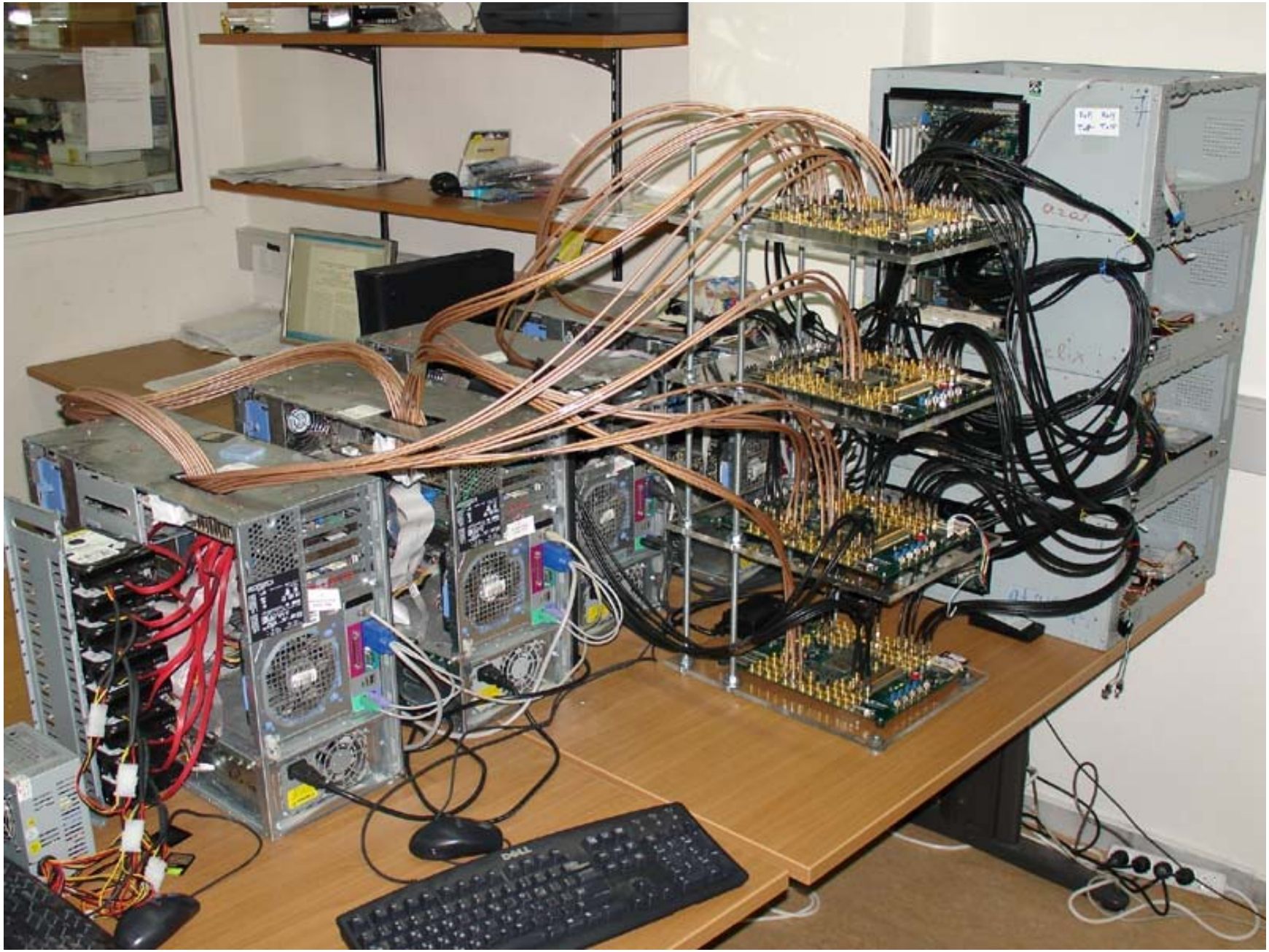
- be realistic when designing new interconnects architectures
- evaluate implementation cost
- enable systems S/W development and experimentation

First System (8 nodes):

- simple – quickly brought up (PCI, RDMA, single-queue)
- later added: Read RDMA, 8 VOQ's, 4-way multipath & resequ.

Next System (20++ nodes):

- processor & lightweight NI in the same FPGA
- queue organization scalable to *O*(64 K) nodes



Photograph of First System (8 nodes)

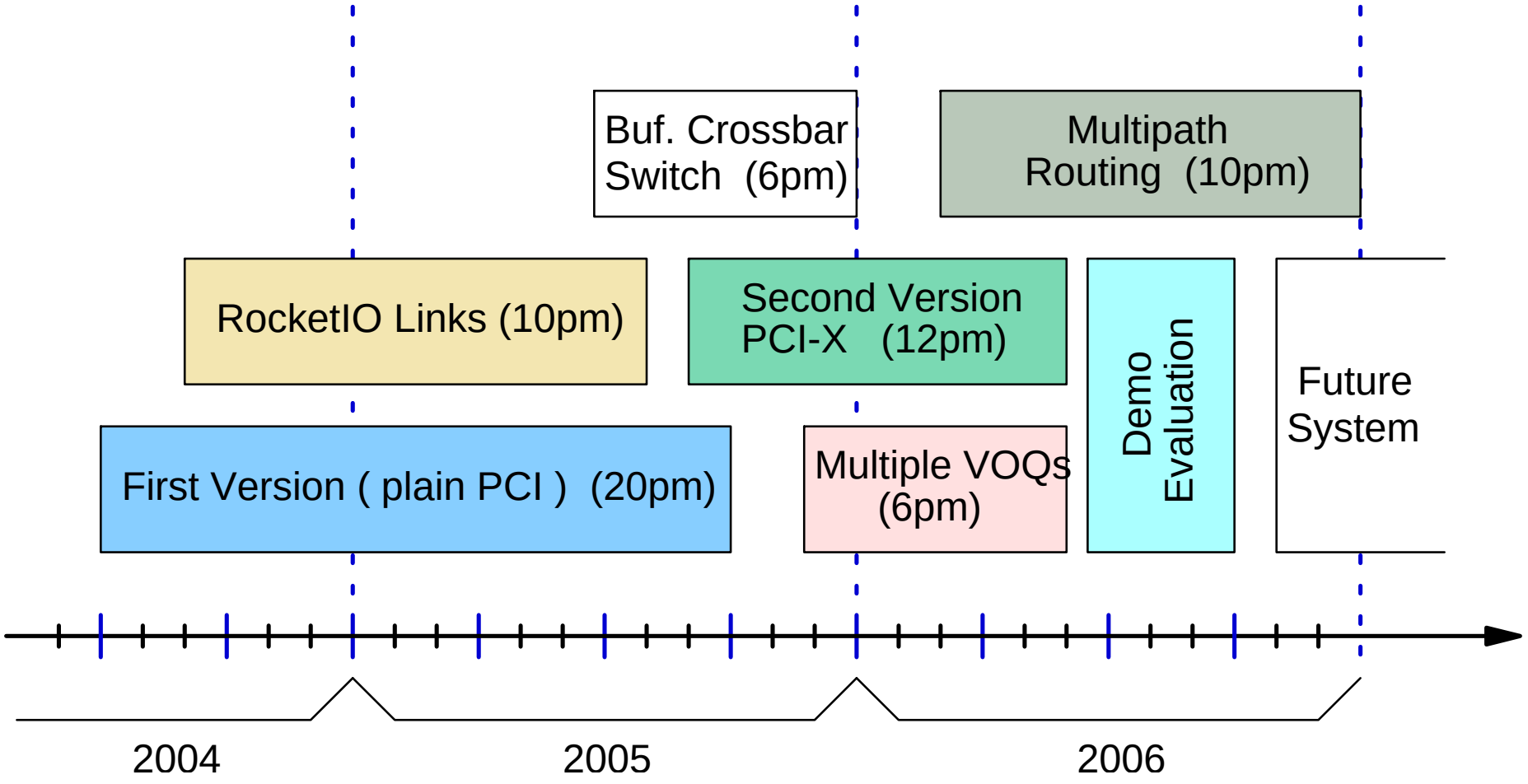
Hardware Components (First System)

- Nodes: 8 commercial PCs
  - 4 Intel Xeon, 4 AMD Opteron
- Network Interfaces (NI): 8 FPGA dev. boards
  - DiniGroup DN6000K10SC boards ($ 4.5K each), containing Xilinx Virtex II Pro XC2VP40 FPGA
  - host interface: PCI-X, 64 bits, 100 MHz
  - network interface: 4 RocketIO links × 2.5 Gbits/s each
  - link bundling (10 Gb/s) modes: Byte-level, or Packet-level
- Interconnection Network (Switches): 4 FPGA dev. boards
  - Xilinx ML325 boards ($ 5K each), containing Xilinx Virtex II Pro XC2VP70 FPGA, 20 link interfaces each

Key Features (First System)

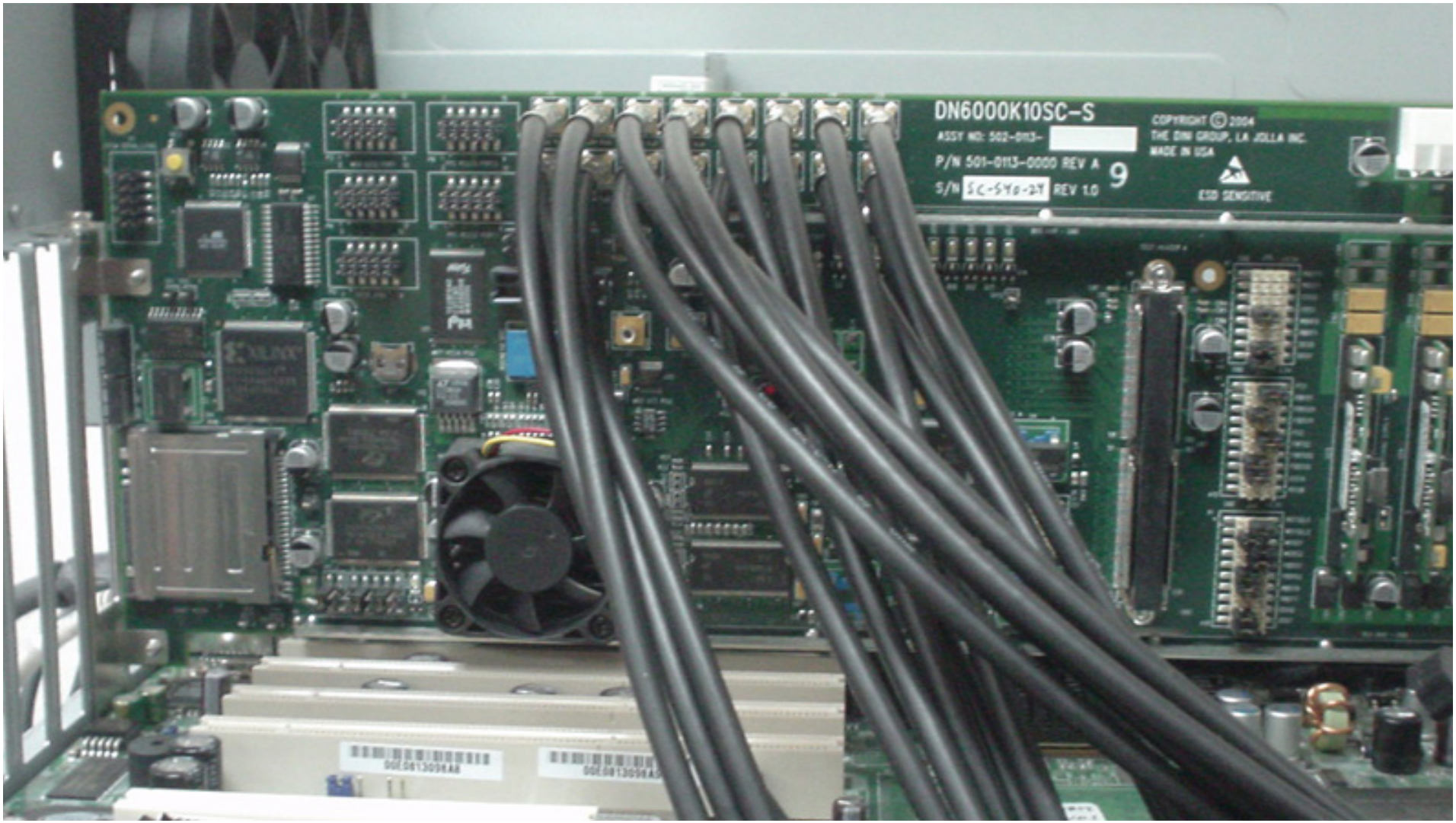
- Remote DMA (RDMA) based operation
- Notifications: departure and/or arrival, interrupt or enqueue
- Remote Enqueue for short messages, multiple senders
- Credit-based flow control: lossless communication
- Per-destination Virtual Output Queues (VOQ's): flow isolation
- Extensive event logging, debugging & performance counters
- Switch:
  - 8x8 implemention – 32-bit datapath @78.125 MHz
  - achieved up to 16x16 – 16-bit dpth @156.25MHz to fit in FPGA
- Linux already adapted for this platform (kernel-mode comm.)
- *MPI port for this platform under way*

Hardware Development Cost (1st System, 2 versions)



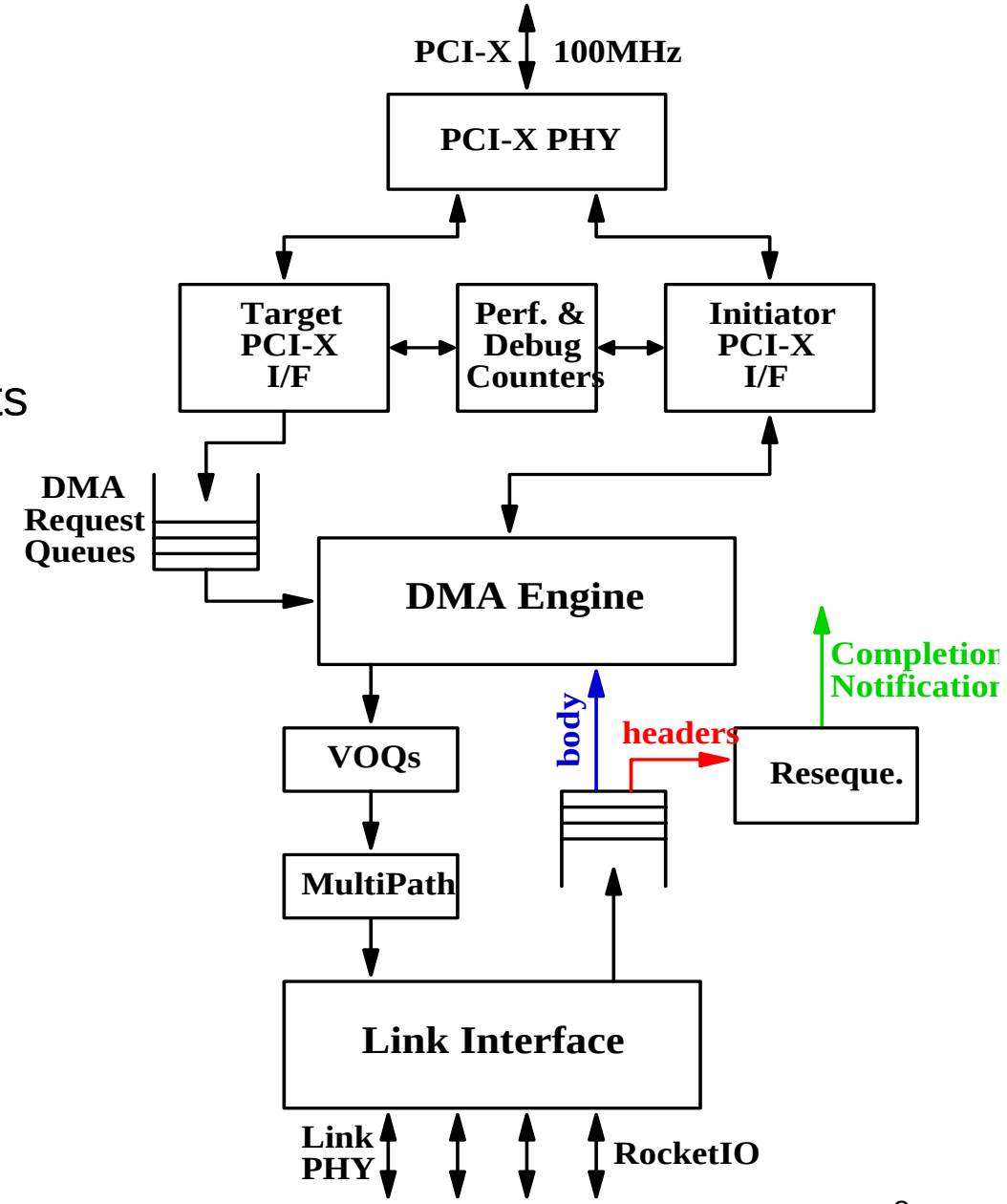
pm = person-months

NI Photo, with 4 RocketIO links

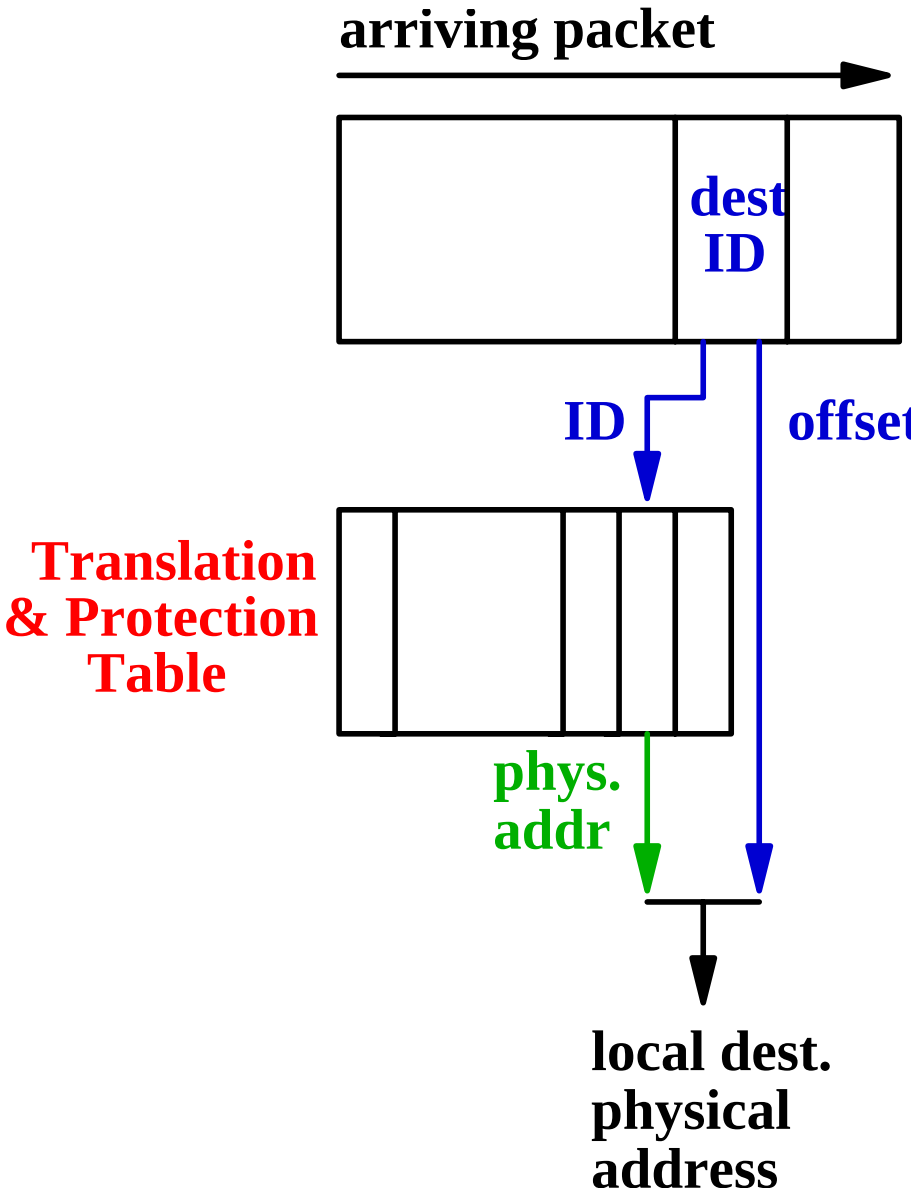


NI Architecture

- 64-bit PCI-X @100MHz to Host
- Per-dest. DMA request Q's
- Per-destination VOQ's
- DMA segementation into packets
- Link bundling: 4×2.5 = 10 Gb/s
  - inverse multiplexing
    multipath routing
- Out-of-order packet arrivals:
  - DMA body immediately written into memory
  - headers wait in resequ. Q's
  - notify completion after resequencing
  - resequ. tolerates 1-pck loss

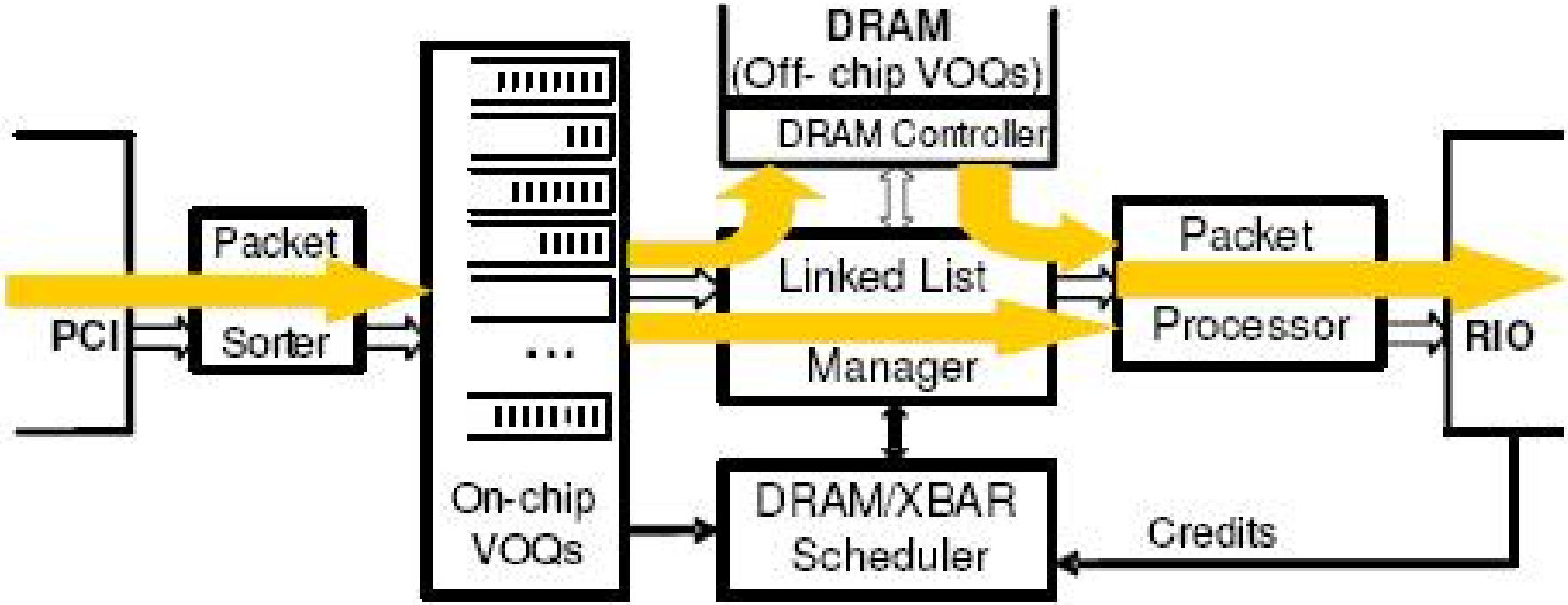


Address Translation and Protection



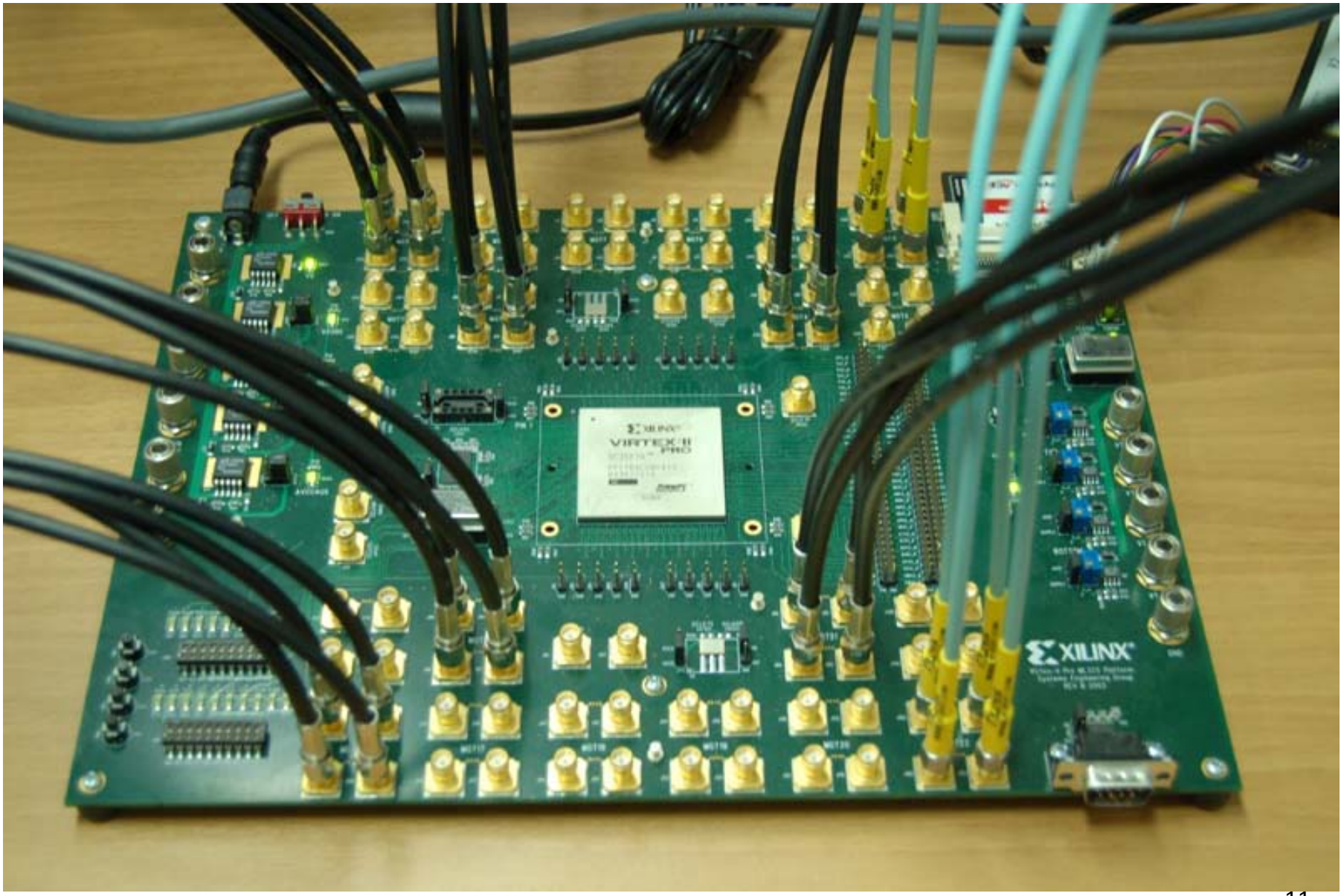
- RDMA packets contain destination IDs & offsets rather than physical addresses
- Translation and Protection Table in the Receiver

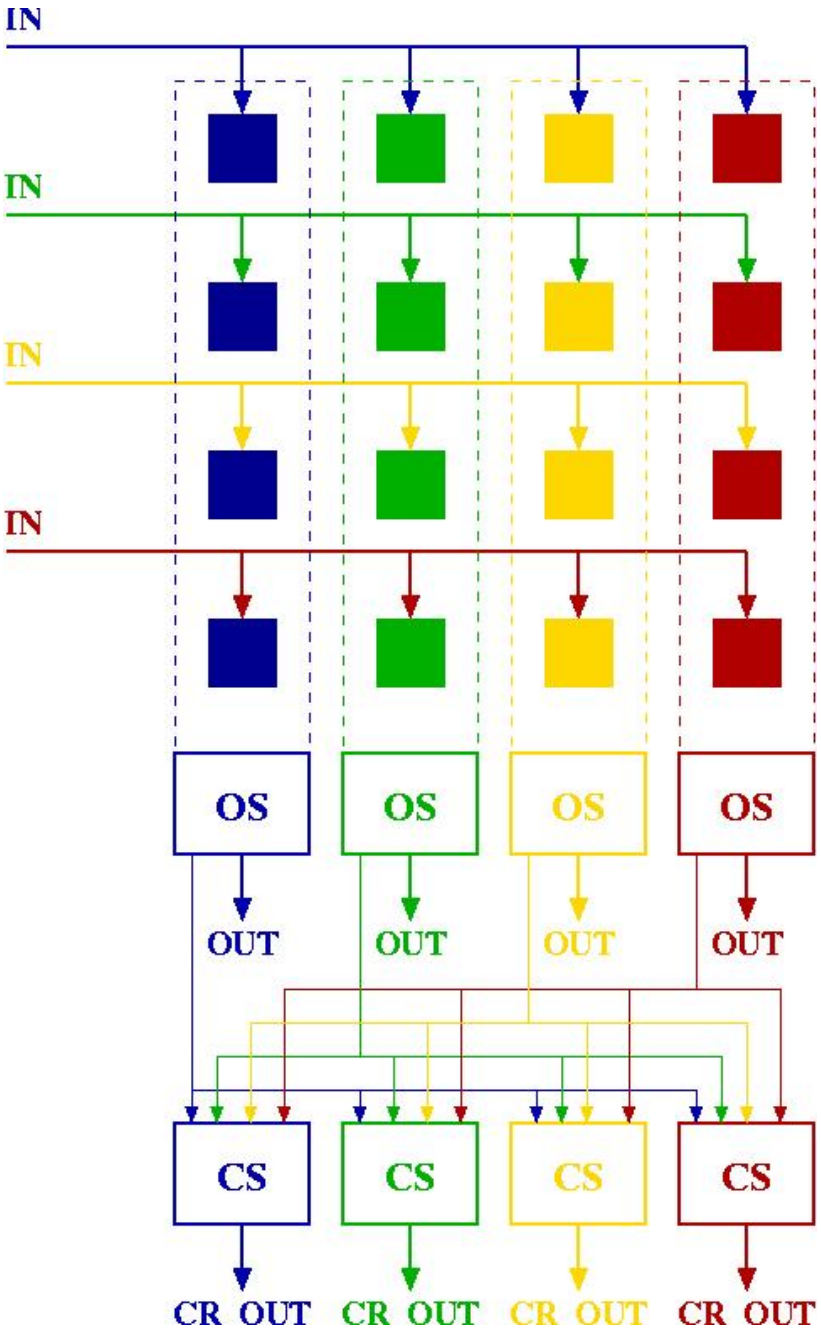
Multiple VOQ support



- Multiple VOQs per destination, to avoid HOL blocking
- Option to format traffic into variable-size multi-packet segments
  - initially, segments reside in on-chip memory
  - when VOQ's grow, they migrate to off-chip DRAM
  - pointer-based linked-list-queue management

Switch Photo





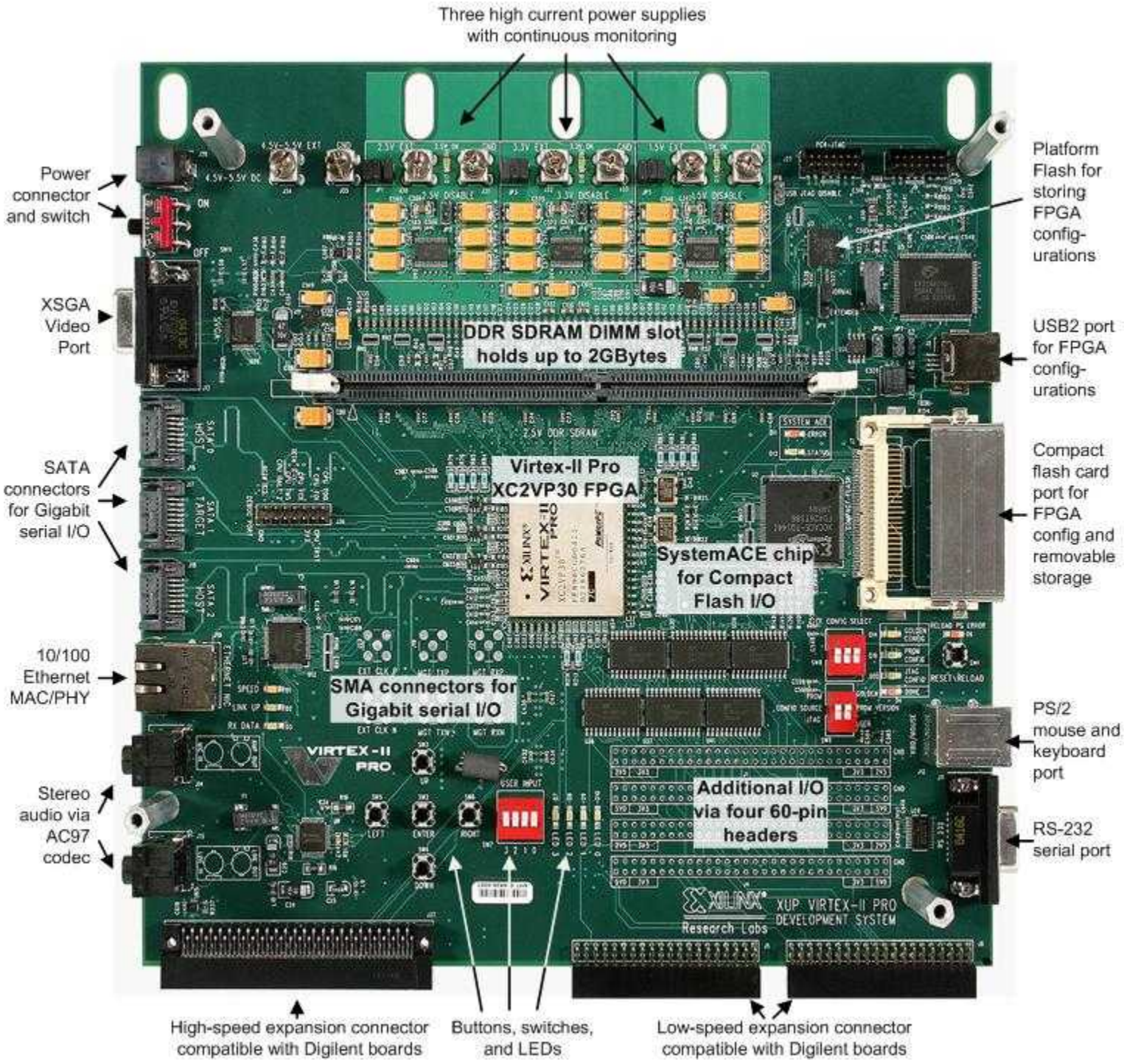
Switch Architecture

- 8x8 buffered crossbar (CICQ) switch
  - Inherent switching of variable-size packets.
  - 64 crosspoints × 2KB each
  - Single priority.
- Round Robin Scheduling
  - Per-output schedulers (OS)
  - Cut-through operation
- Credit-based flow control
  - CS = Credit schedulers
  - Credits and Data share the same links

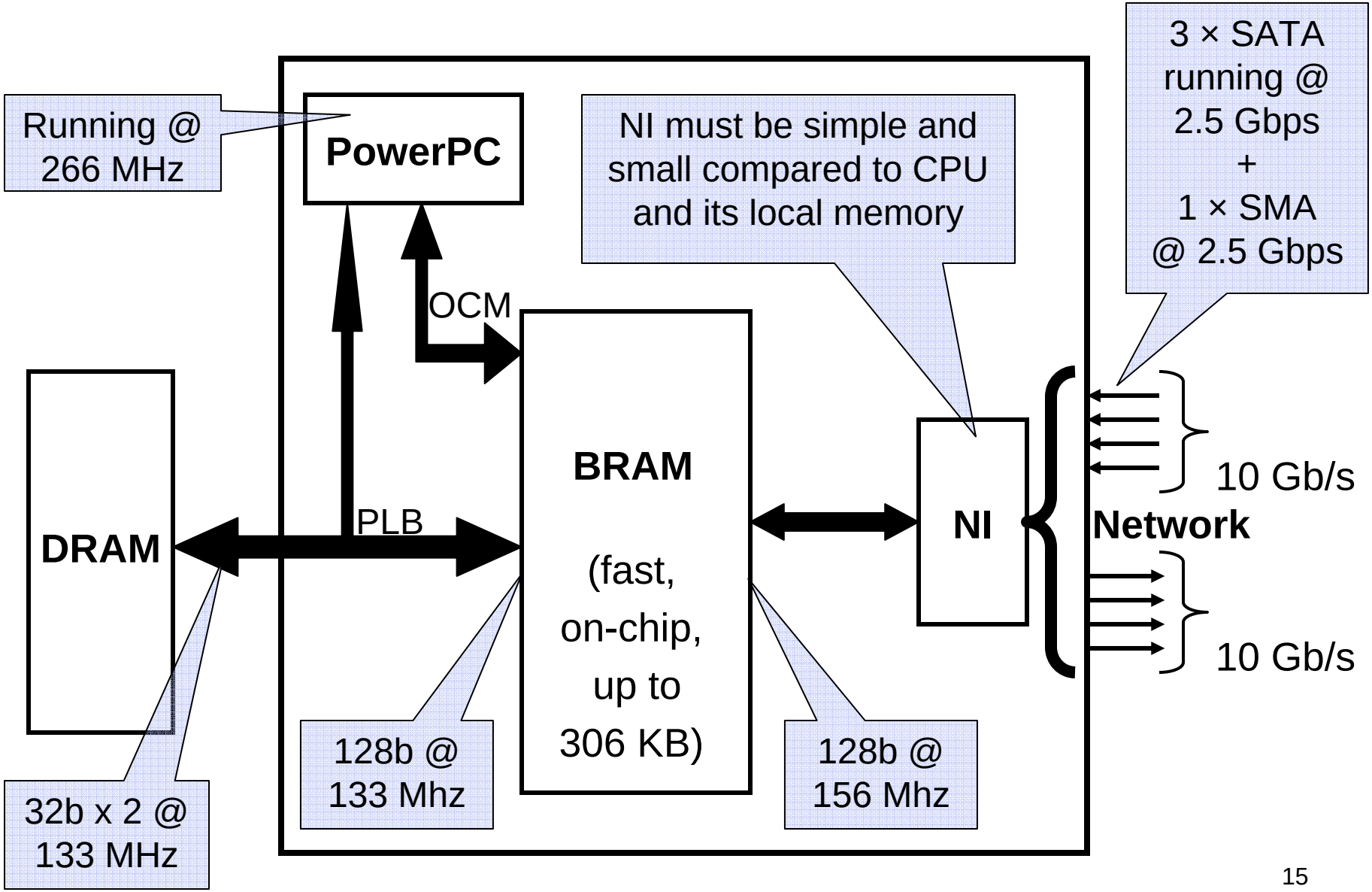
Next Generation System

- Reduce size, fan noise, cost, and…
- Tightly couple the NI to the host processor:
  - replace PC's with the processors inside the FPGA's
  - abandon PCI-X
  - use inexpensive (~ 400 $) Xilinx University-Program boards
- Lightweight Network Interface
- Architecture for supporting *O(* 64K *)* nodes
  - Q's & resources allocated only to active connections
  - accordingly adapt flow control & congestion management
- Timeframe: 2007…

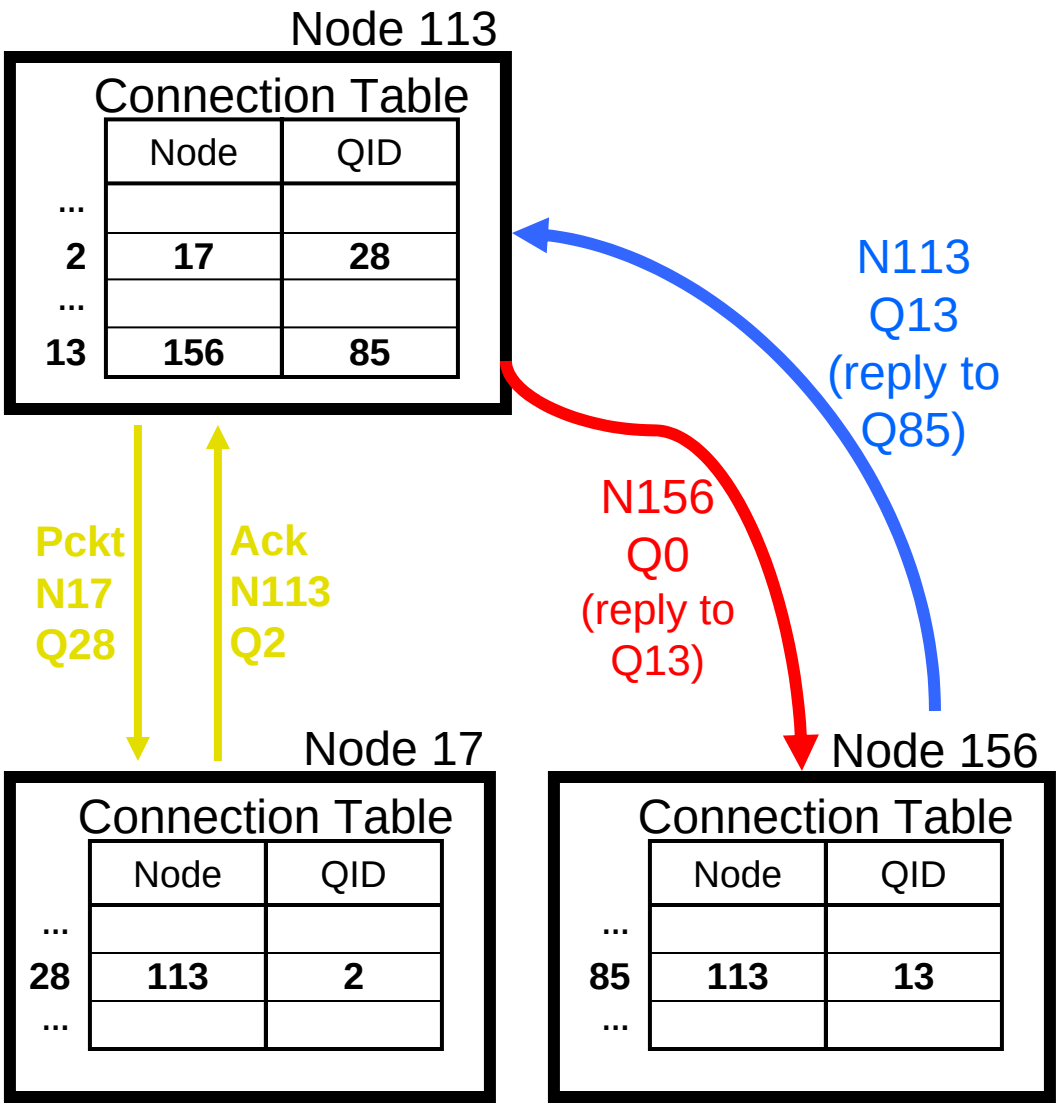
Next-Generation System Node



Next Generation Node: Block Diagram

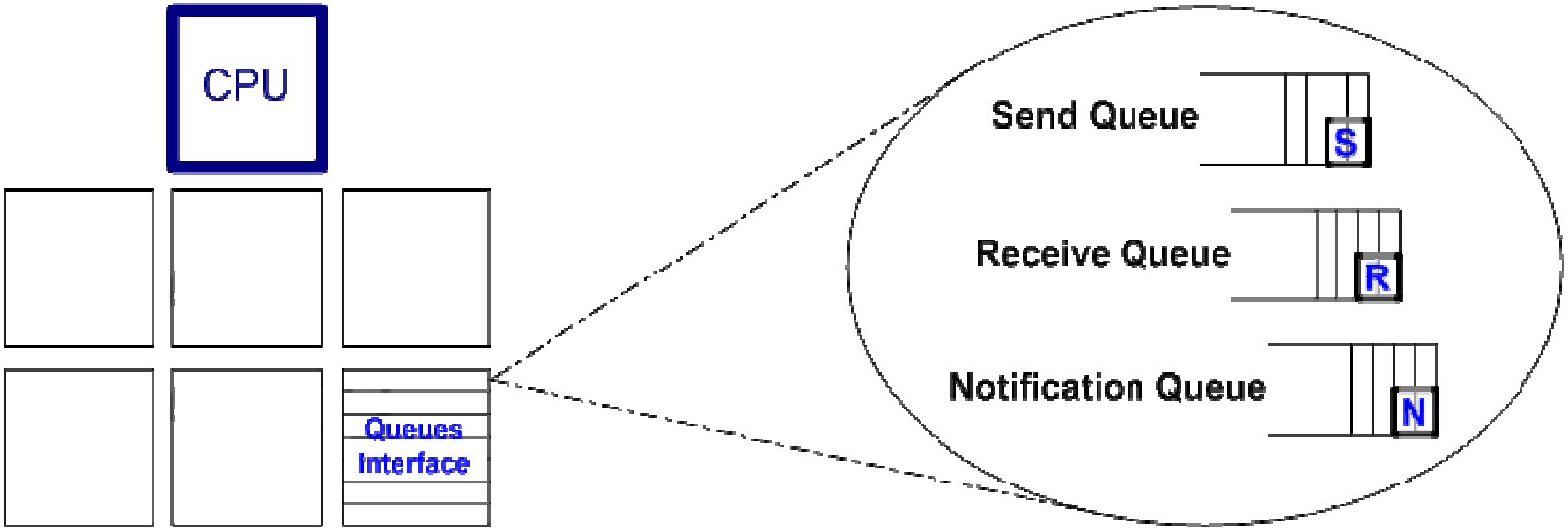


Next Generation Queues: Connection example



- Nodes 113 and 17 already connected and communicating
- Node 113 requests connection with node 156
- Node 156 handles request
- Node 156 sends response to node 113
- Node 113 handles response

Envisioned Future CMP Network Interfaces



- NI Queues in the Cache
- NI in the Cache Controller
- construct packets via store instructions
- receive packets via load instructions

RDMA for large transfers;

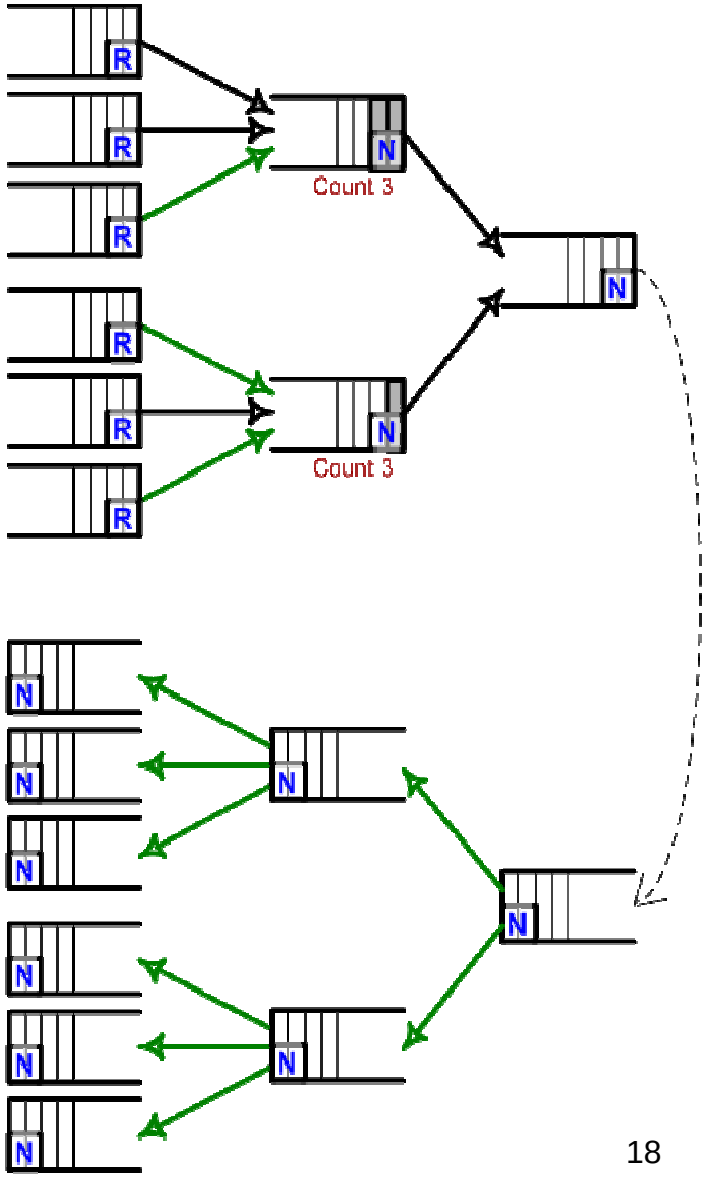
Remote Queues for:

- small messages
  - requests, commands, pointer-passing
- multiple writers and/or multiple readers
- arrivals may trigger actions, including *Notification* generation

Envisioned Future Synchronization Support

- Notification Queues (NQs)
  - *Notification* =
    Address of posting queue
  - Multiple Writers – Single Reader
  - Trigger *Notifications*
    (incl. from *notifications* collected)
  - Interrupt Coalescing/Reduction

- Hierarchical Barrier Example



Conclusions

- High-Speed Interprocessor Communication Research
- Prototyping, in order to keep as close to reality as possible
- Network Research:
  - Switch Architecture
  - Flow Control, Congestion Management, Multipath Routing
- Network Interface Research:
  - Tight Coupling to the Host
  - Low cost – resource sharing with host memory
  - RDMA and Remote Queue support
  - Multipath, Multiqueue, Virtualization support
  - Synchronization and Notification support