Routing on the QPACE parallel computer

Guest Student Programme 2010

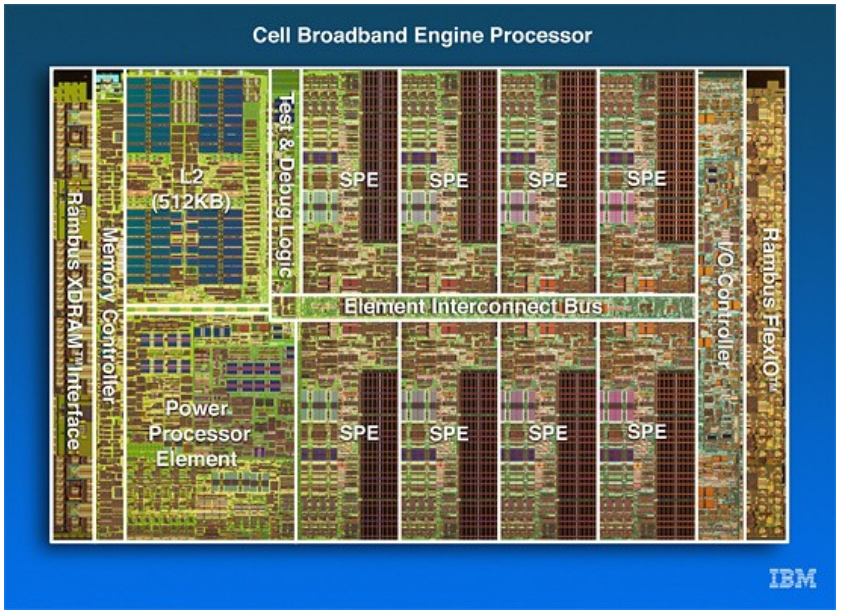
15. November 2010 | Konstantin Boyanov

Overview

- Architecture of QPACE
- The Network Processor and the Torus Network
- Types of routing and applicable routing algorithms
- Event-based simulation environment OMNeT++
- Implementation of a Simulation Model
- Conclusion and Outlook

The QPACE Parallel Computer

- QPACE: „QCD Parallel Computer based on Cell processors“
- Specially developed for QCD numerical simulations
- PowerXCell8i Processor
  - 102 GFlops double precision

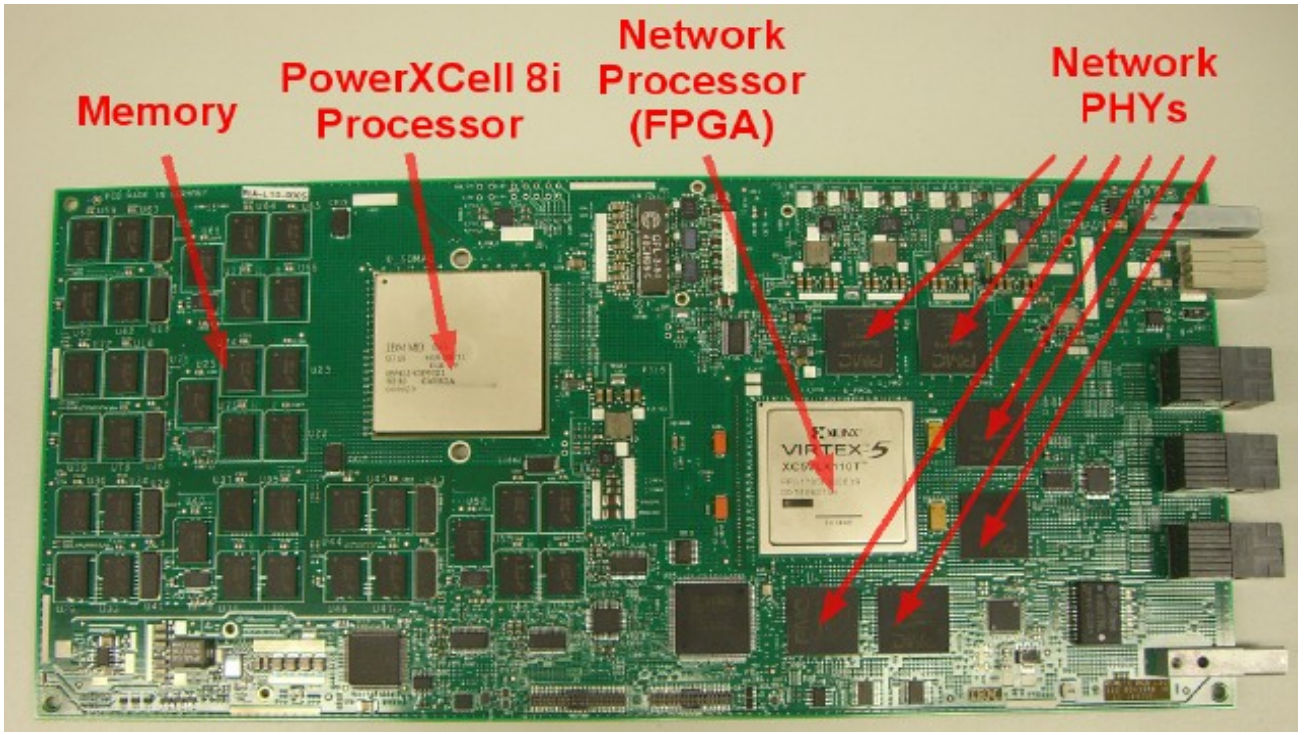


- Cell CPUs are interconnected through custom torus network

From [1]

- Number #1 in Green500 (June 2010)
  - High Performance LINPACK - 44.50 TFlops sustained on 512 nodes
  - Energy efficiency - 773 MFflops / Watt
    - #2 in Green 500 - Nebulae 492,64 MFlops/Watt
    - JUGENE 363,9 MFlops / Watt

QPACE Architecture

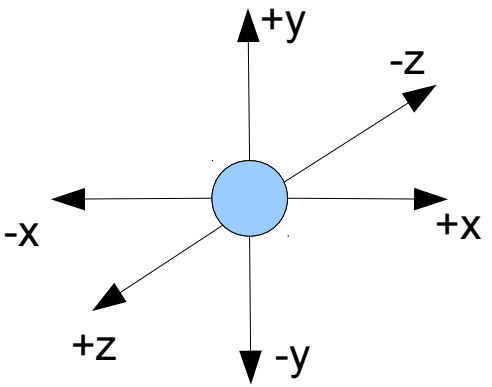


From [2]

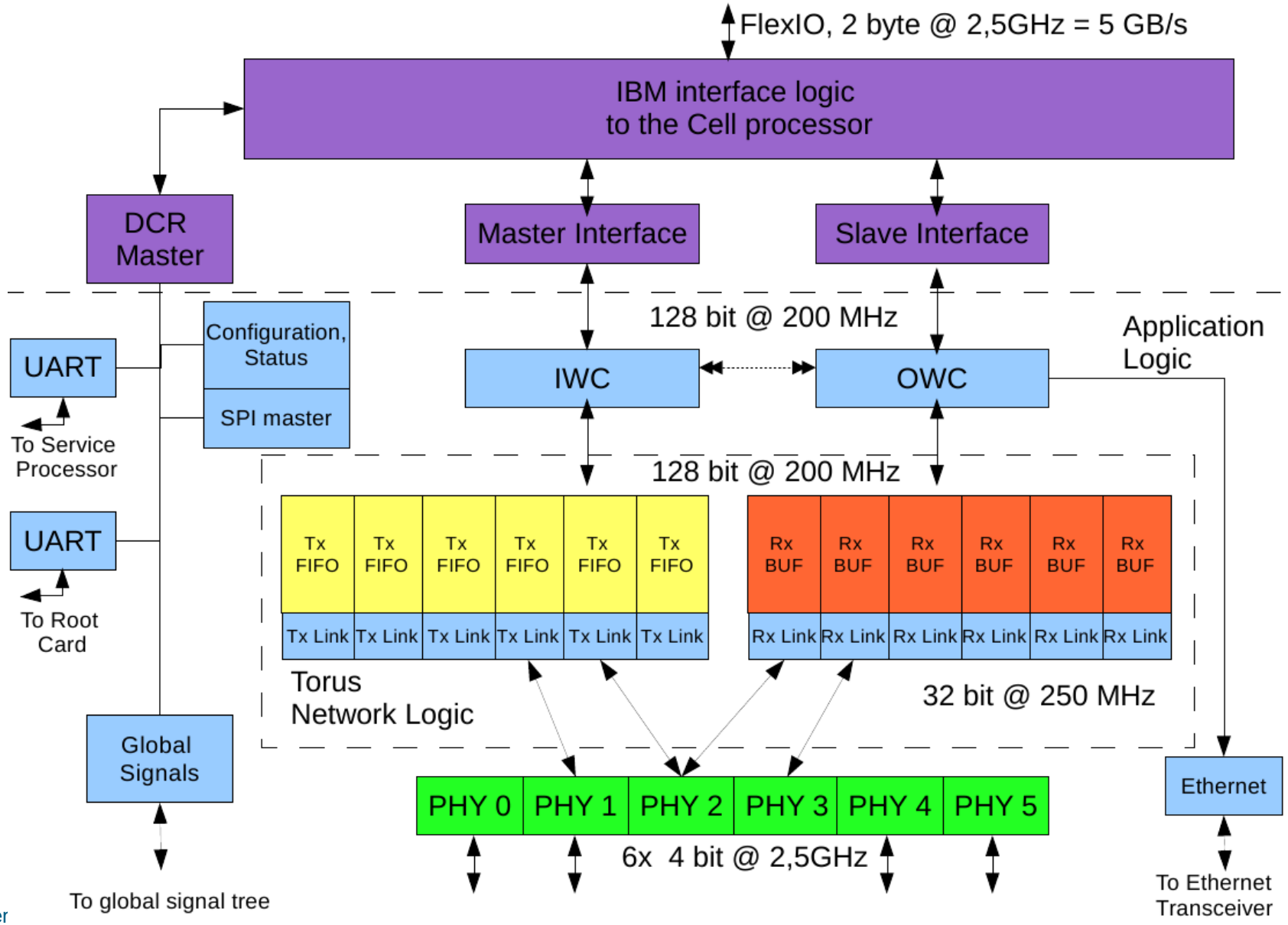
- QPACE rack = 8 backplanes x 32 node cards
- Liquid cooling makes high performance density possible
- 2 installations with 4 racks each

QPACE Network Processor

- Implemented on Field Programmable Gate Array (FPGA)
  - Reprogrammable logic blocks connected by reconfigurable interconnects
  - Ready-to-use circuit modules (Ethernet MAC, PCIe cores, memory)
  - Flexible development, Non-Recurring-Engineering costs low
- QPACE Network Processor is a southbridge with various tasks
- Physical links
  - 2x FlexIO links to Cell
  - 6x 10 GbE torus network links
  - 1 Gigabit Ethernet link for I/O
- Torus Network
  - Nearest neighbour communication, Local Store to Local Store data transfer

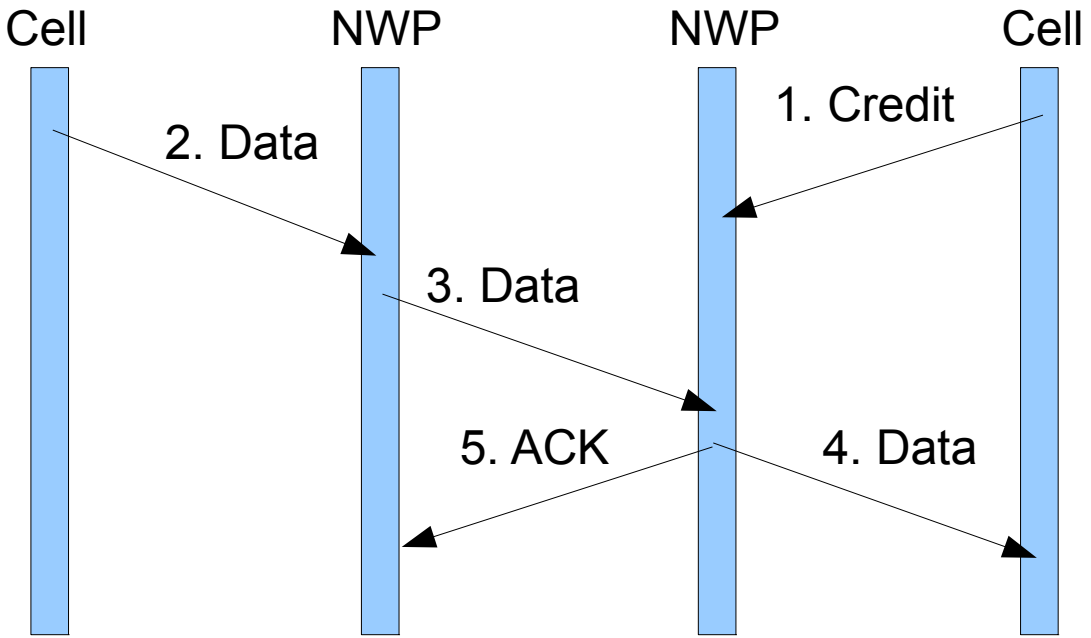


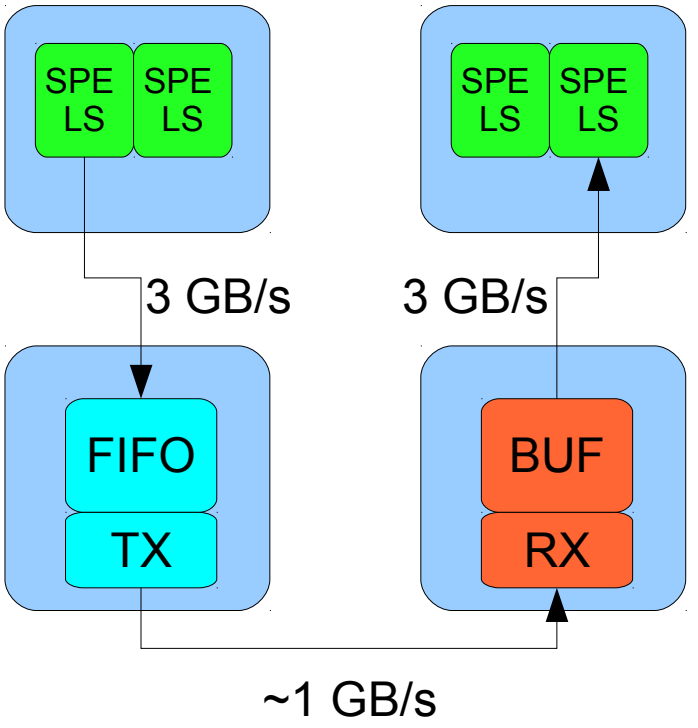
QPACE Network Processor



Torus Network: Communication Protocol

- Custom 2-sided communication protocol
  - Receive operation: `tnw_credit()`
  - Send operation: `tnw_put()`



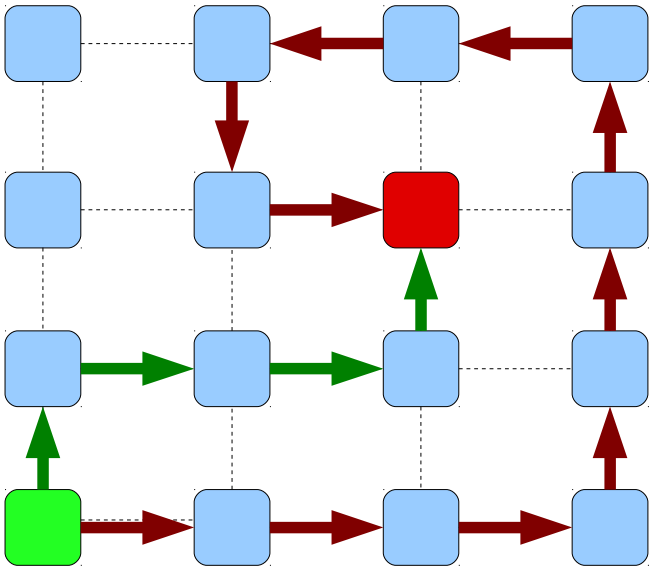


Why Routing in QPACE?

- QPACE allows now only for nearest-neighbour communication
- Routing makes any-to-any communication possible
- Processor and parallel architecture promising for other applications
- For example FFT-based applications
  - Row-column FFT algorithms on large data sets
    - *2-dimensional data (matrix)*
    - *Perform one-dimensional FFT on one dimension first, then perform FFT on the other*
    - *Before starting the FFT on the second dimension, transpose the data set to allow transforms to operate on local/continuous data*
    - *Communication pattern in parallel environment similar to transposing very large matrices*

Some Routing Terminology and Techniques

- Static vs. Dynamic Routing
  - Routing table created at system start time
  - Routes change dynamically during runtime
- Common Routing Techniques [4]
  - Store-and-forward routing
  - Virtual cut-through routing
  - Wormhole routing
  - Hot-potato routing
- Deadlock and Livelock
  - Packet(s) cannot be forwarded due to resource contention or lack of forward routes
  - Packet(s) travel in circular fashion around their destination forever



Proposed Routing Algorithm for QPACE (1)

- Messages in QPACE are composed of equally sized packets (128 byte)
  - Lean protocol demands certain packet ordering rules
- Extended packet header
  - Address of each node is represented by its coordinates in the 3D torus
  - Contains the relative offset between source and destination along the 3 dimensions
  - For example communication between node A (source)

    and node B(destination):

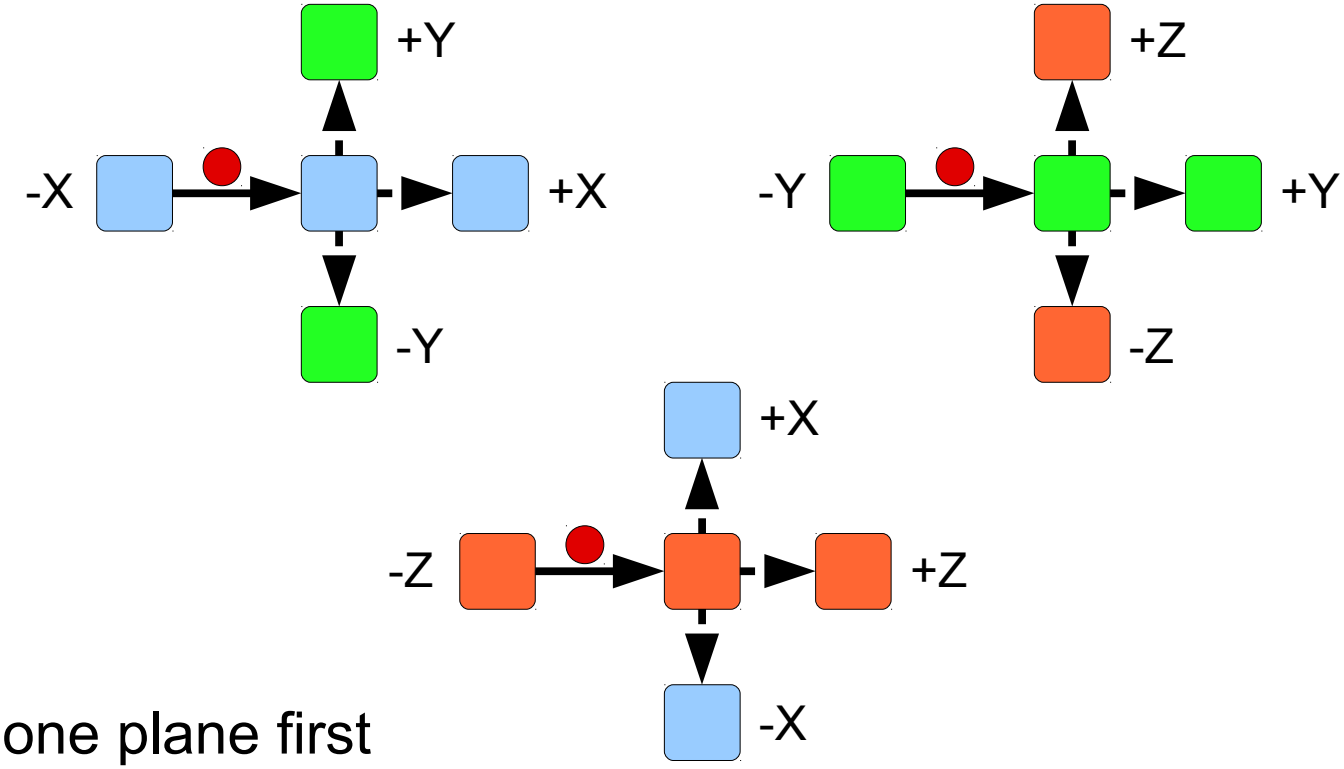
    $(A_x, A_y, A_z) = (1,2,1)$, $(B_x, B_y, B_z) = (0,2,2)$

    Header contains $(\Delta_x, \Delta_y, \Delta_z) = (B_x, B_y, B_z) - (A_x, A_y, A_z) = (-1, 0, 1)$
  - At each hop the corresponding value in the header is decremented or incremented

Proposed Routing Algorithm for QPACE (2)

- Forward directions

From	Towards
+X	-X, +Y, -Y
-X	+X, -Y, +Y
+Y	-Y, +Z, -Z
-Y	+Y, +Z, -Z
+Z	-Z, +X, -X
-Z	+Z, +X, -X



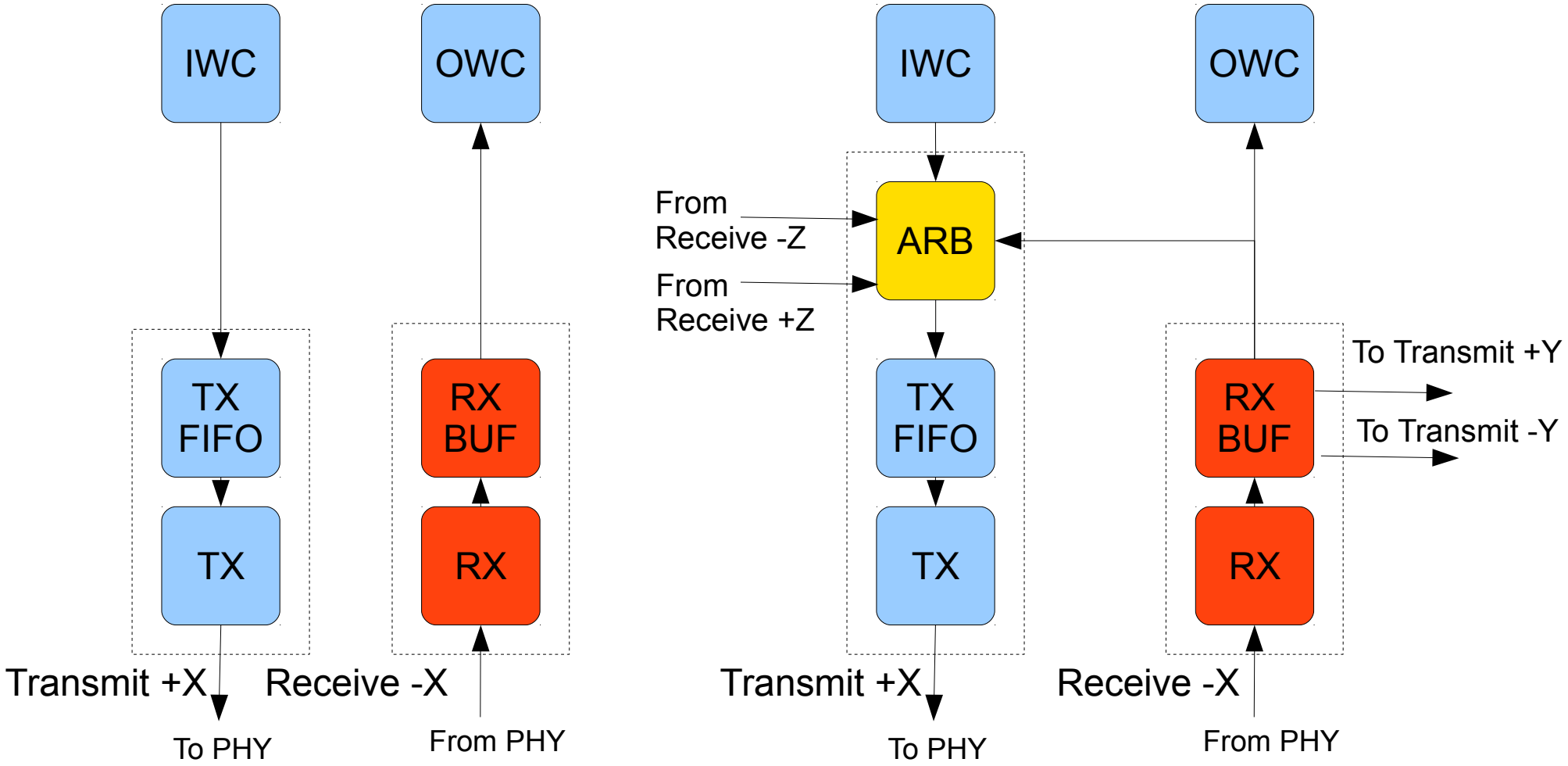
- A packet is routed in one plane first
  - Continues its travel forward or makes a turn according to $(\Delta x, \Delta y, \Delta z)$
  - Packets must start into particular direction at the sender node to avoid being stuck with no further forward route

Proposed Routing Algorithm for QPACE (3)

- Advantages
  - Minimal number of connections between transmit (TX) and receive (RX) links
  - No additional buffers
  - Simple routing logic
  - No complex routing tables
  - No significant changes to the hardware architecture of the network processor

Proposed Routing Algorithm for QPACE (4)

- Required changes to the network processor



Proposed Routing Algorithm for QPACE (5)

- Proposed routing algorithm is not deadlock-free
  - Receive buffer of node A is filled with packets waiting to be forwarded → no further packets can travel along this link
  - Also packets with node A as final destination might be blocked
  - If this condition occurs on all nodes simultaneously, the communication system deadlocks
- Strategies to prevent deadlocks
  - Give higher priority to packets within the network than packets injected into the network
  - Limit message size so no receive buffer gets full
  - Software control of messages

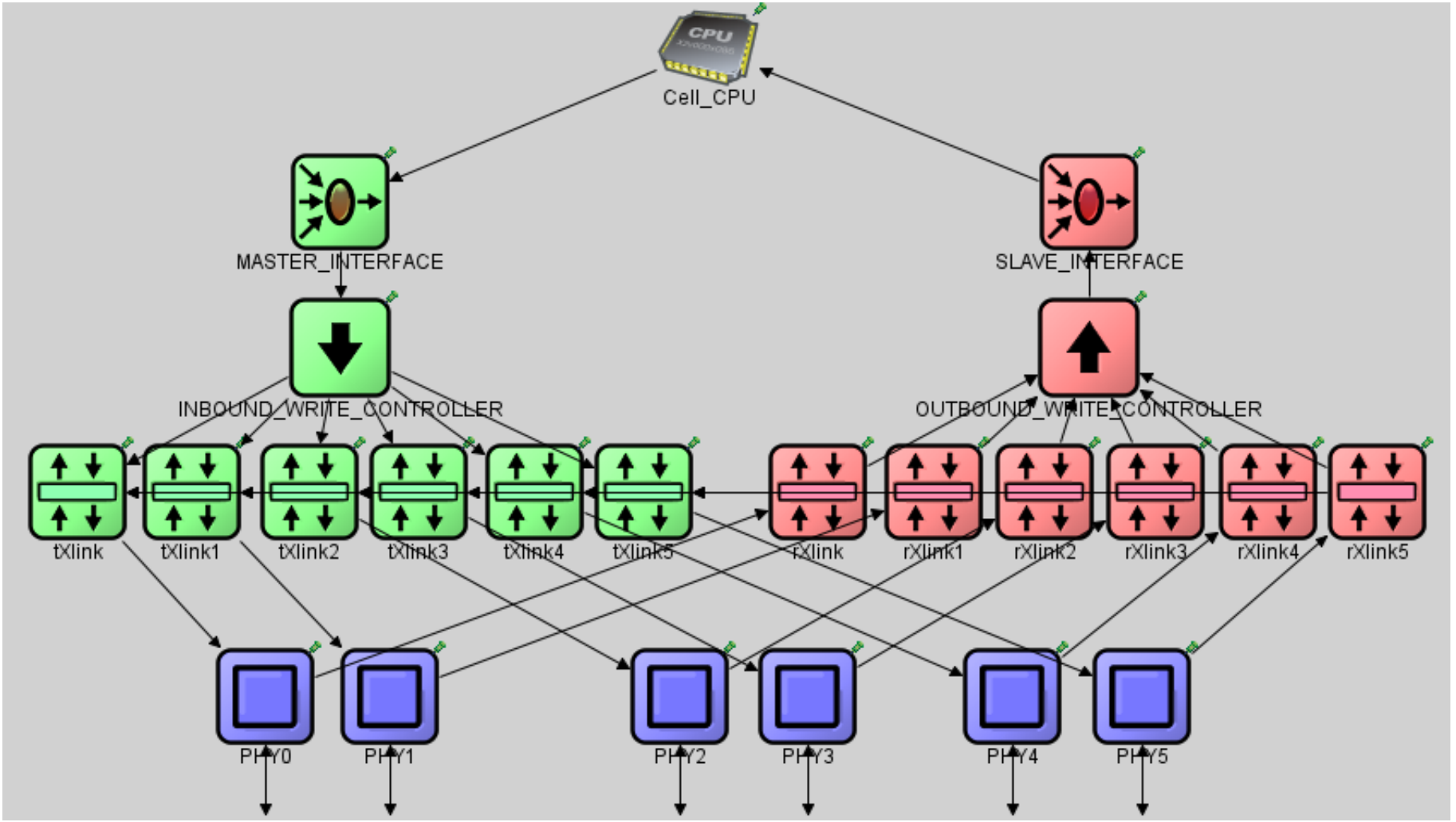
The OMNeT++ Simulation Environment

- OMNeT++ → Objective Modular Network Testbed in C++ [3]
- Event-based Simulation
  - System is represented as chronological sequence of events
  - Every event marks a change in the state of the system
  - Test new functionality of complex system without physically altering the system
  - Diagnose issues in existing system, test performance
- OMNeT++ Simple and Compound Modules
  - Simple custom language for module description (NED)
  - Functionality encapsulated in C++ classes and methods
- OMNeT++ Connections
  - Allow to define data rates, delays

Differences with Respect to the Real Hardware

- Simplification: no Virtual Channels
  - In real hardware 8 virtual channels per link/direction
- Messages and packets
  - No Credits between Receive Buffers and CPU
  - No feedback on the links
- Backpressure
  - Between processor and Slave Interface (sender side)
- Contention information and flow control among neighbouring nodes
  - OMNeT++ signalling

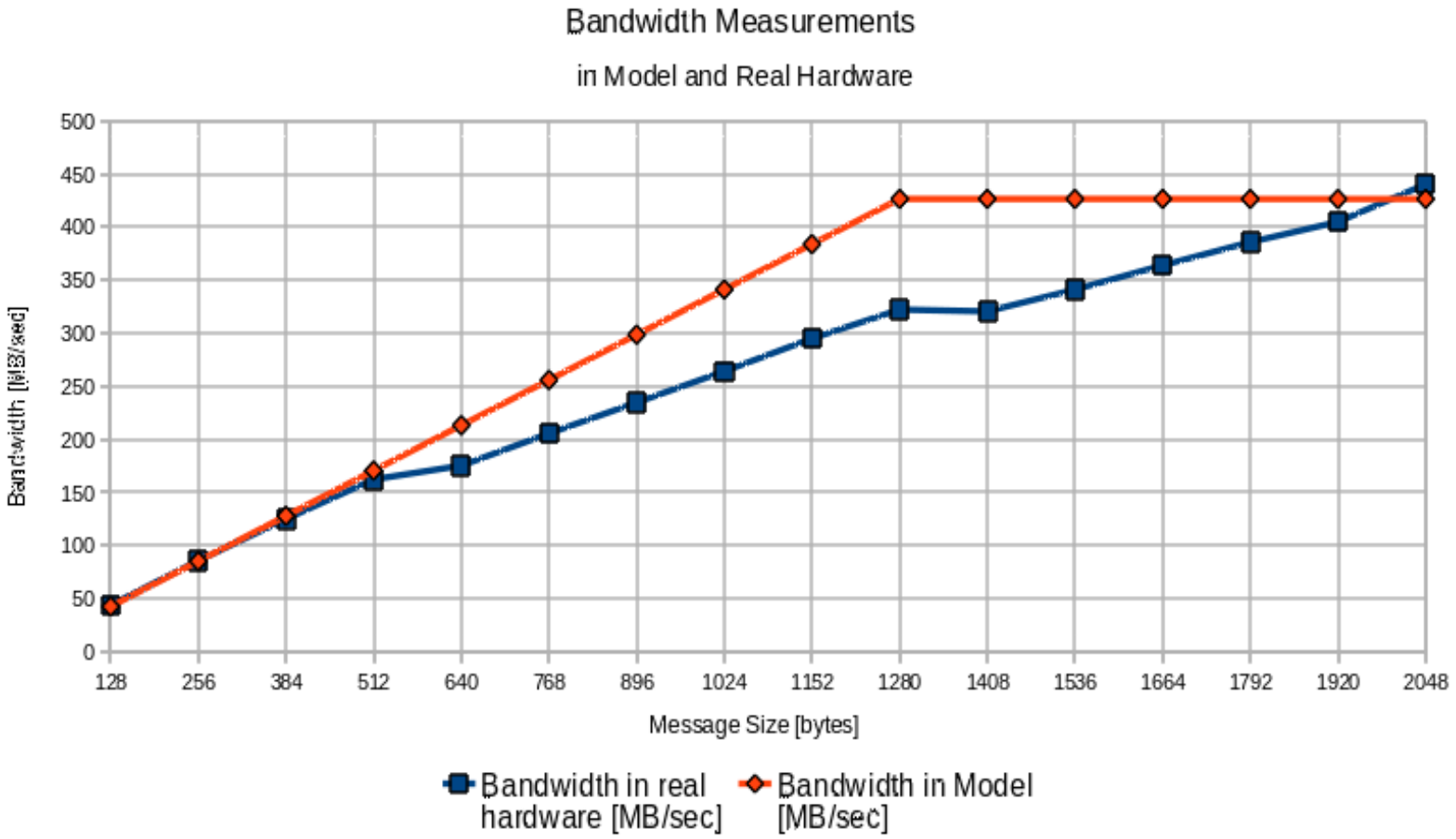
The QPACE Model in OMNeT++



Model Verification: Bandwidth Comparison for Point-to-Point Communication

- Using a latency limited communication pattern

  one message every 3 µs



Testing the Routing Algorithm

- Communication pattern from an example application
- Transposing large quadratic matrices
  - Every node ***i*** is assigned a set of rows in the matrix
  - Every node ***i*** sends parts of its row to every other node `j`
  - For example for a 2x2x2 torus (nodes 0-7) and 128x128 matrices every node gets 16 rows.
    - *Node 0 sends entries 16-31 of all its rows to node 1, entries 32-47 to node 2, etc.*
    - *Node 0 keeps entries 0-15 and receives all entries 0-15 from all other nodes*
- At the end every node sends messages to any other node, and receives from any other node
- Functional test of my model successfully passed using up to 1024 nodes

Conclusion and Outlook

- Good overlapping of bandwidth measurements in simulation model and on real hardware
- Routing algorithm implemented in simulation environment
- Routing algorithm verified by test application
  - Transpose Matrix communication pattern does not fully utilize link bandwidth
- Precise and comprehensive measurements to be made
  - End-to-end packet transmission delay
  - Bandwidth at a reference receiver node as function of torus size and packet count for matrix transposition
- Tests with other communication patterns

The End

Thank you for your attention!

Questions?

References

- [1] http://www.research.ibm.com/cell/
- [2] http://hpc.desy.de/qpace/
- [3] http://www.omnetpp.org/
- [4] Terry Tao Ye, Luca Benini, Giovanni De Micheli - “Packetization and Routing Analysis of On-Chip Multiprocessor Networks”
- [5] M. Blumrich, D. Chen, P. Coteus - “Design and Analysis of the BlueGene/L Torus Interconnection Network”
- [6] Erik Demaine, Sampalli Srinivas - “Routing Algorithms on static interconnection networks: A classification scheme”

General Information and Disclamer

Konstantin Boyanov

Deutsches Elektronen Synchrotron - DESY (Zeuthen),

DESY Zeuthen, Platanenallee 6, 15738 Zeuthen

Tel.:+49(33762)77178

konstantin.boyanov@desy.de