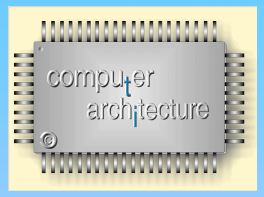
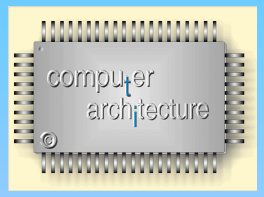


ESCUG 2004

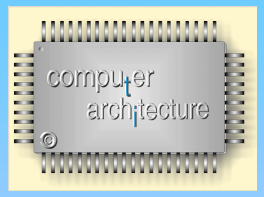
System Level Modeling of a High Performance System Area Network Chip

David Slogsnat, Ingo Feldner, Ulrich Brüning
slogsnat@uni-mannheim.de
Computer Architecture Group
University of Mannheim, Germany



Presentation Outline

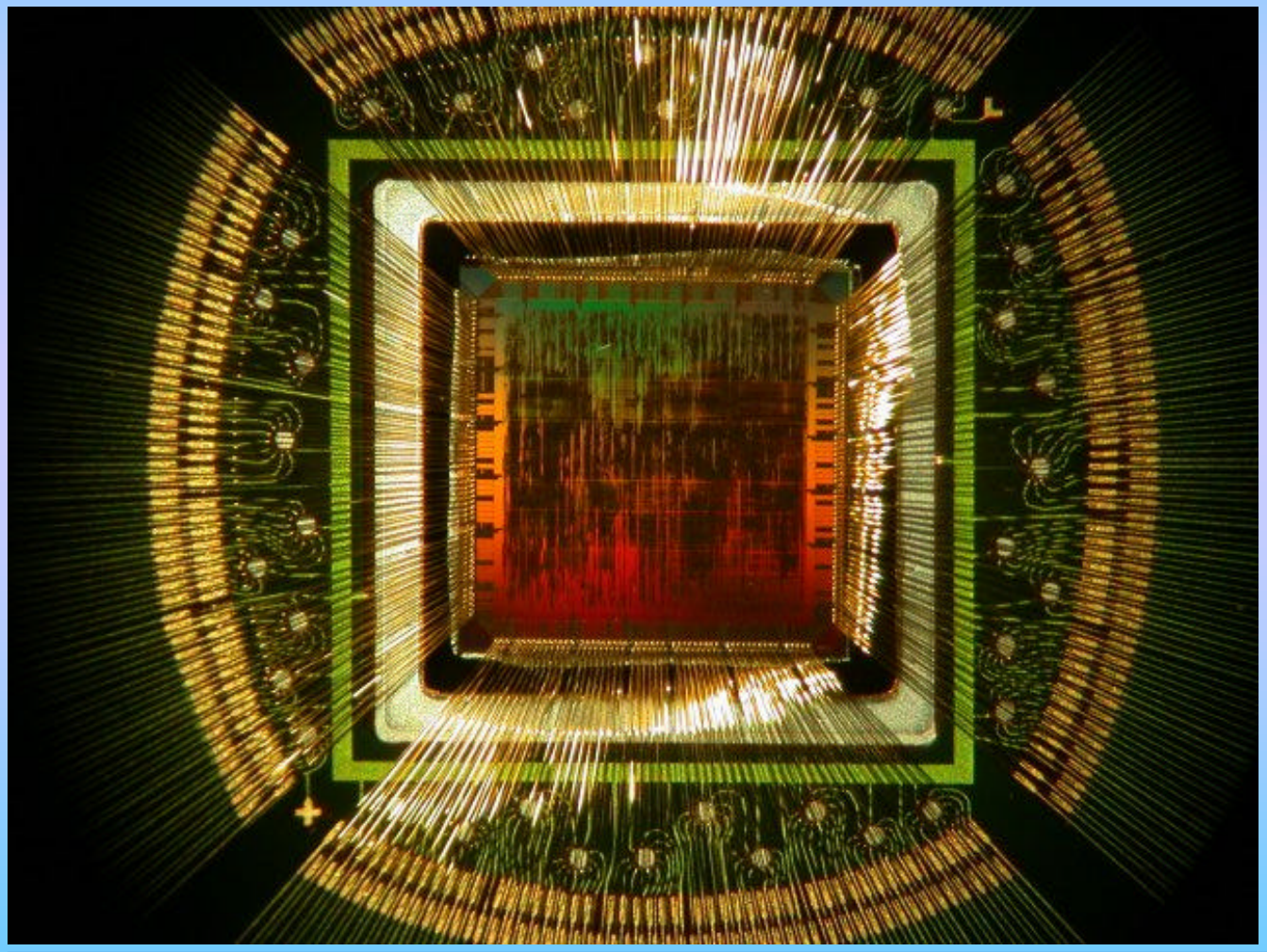
- Motivation
- Overview of the ATOLL hard- and software
- Simulation implementation
- Results
- Outlook & Ongoing Work



Motivation

- Executable Specification for the ATOLL device in SystemC
  - “Traditional” simulation flow insufficient
  - Explore the possibilities/abilities of hardware/software co-design with SystemC for complex systems
  - Template and methodology for future projects
- Requirements:
  - High simulation performance (simulation model orders of magnitudes faster than RTL simulations)
  - User applications must run without recompilation/relinking
  - Switch from hardware to simulation transparent to software
  - Smooth transition from executable specification to RTL model

ATOLL Chip Photo



ATOLL-Chip with bonding

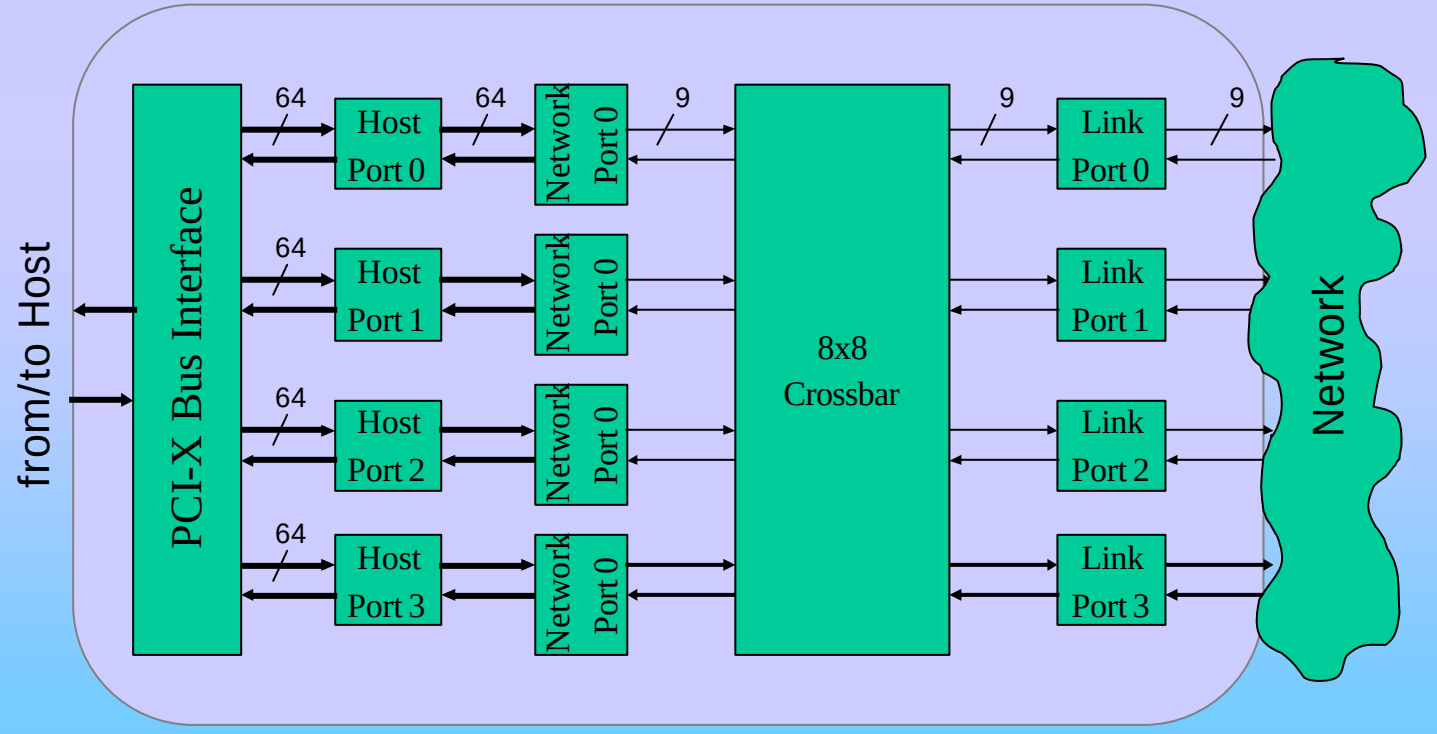
4,5 Mio. transistors

0.18µm CMOS process

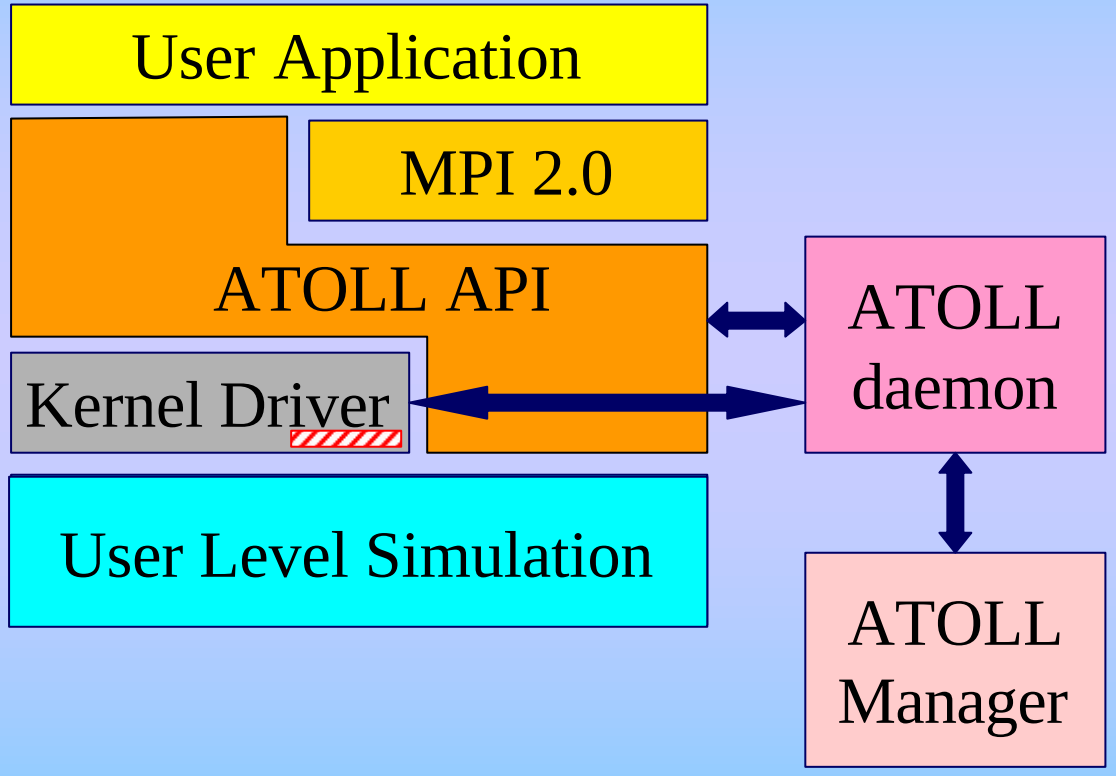
5,7 x 5,7 mm Chip

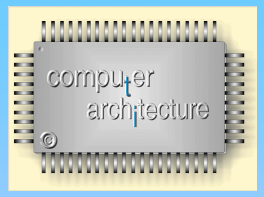
Basic Architecture of ATOLL

- User-level lowest-latency communication through 4 independent host ports (ingress)
- Crossbar integrated in the NIC, no central switch required



ATOLL-Software

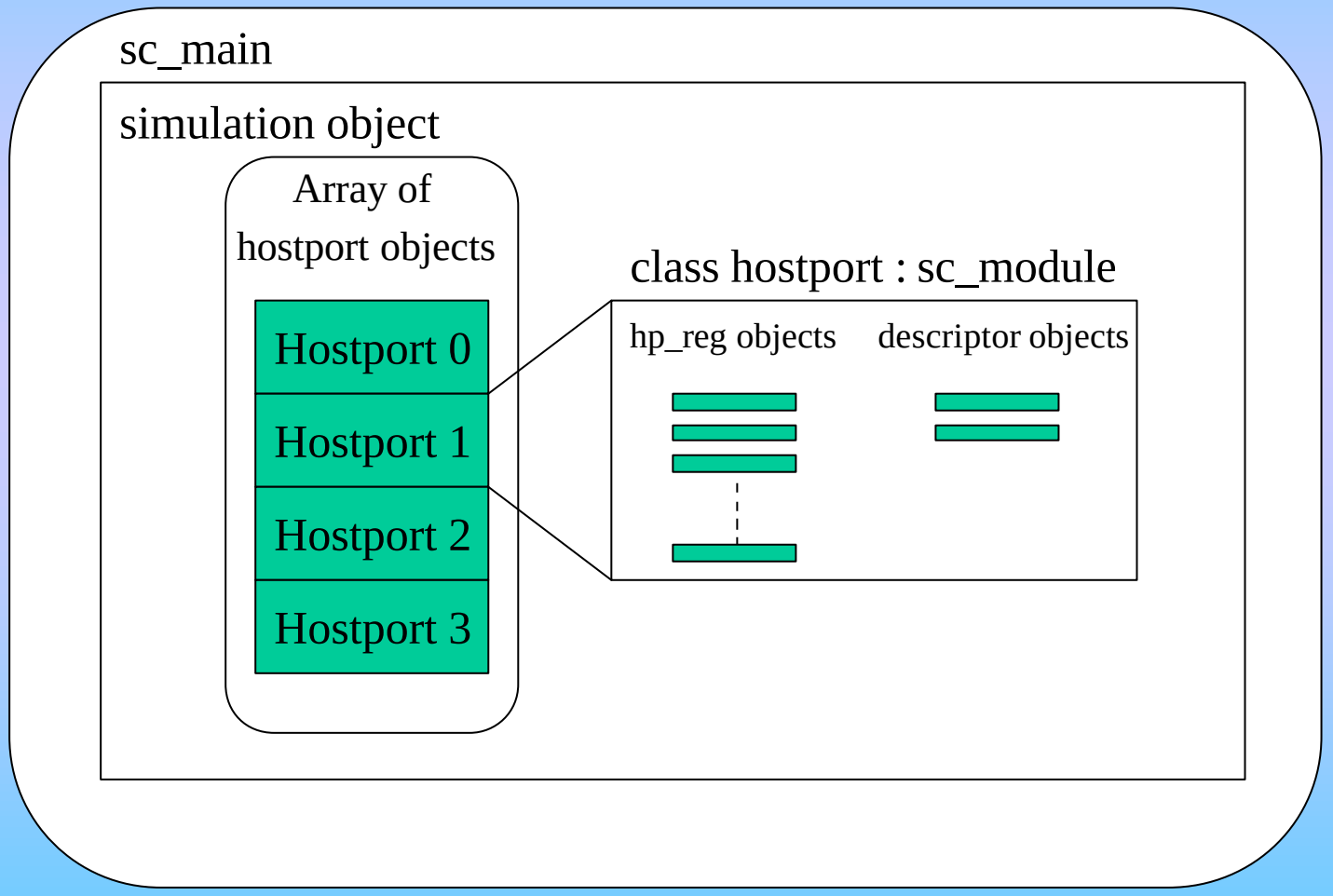


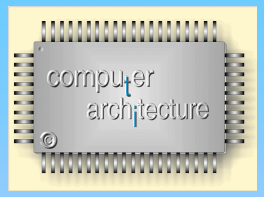


Driver Modifications

- Switch between simulation mode and normal operation by compile flag
- Emulation of device memory (ATOLL PCI address space)
  1. Allocate the virtual memory area reserved for the device
  2. Initialize all pointers to the device memory to the allocated memory
  3. Make the allocated memory available to the simulation process (mmap)
  4. Map the allocated memory to a user application during open
- ~ 30 lines of code changed

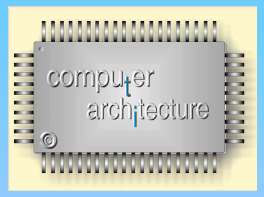
Simulation Overview





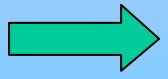
Simulation Class

- The simulation object represents the ATOLL NIC during simulation
- Tasks of the simulation object:
  1. Communicates with the ATOLL device driver and accesses the driver´s “device memory”
  2. Creates and initializes all structural elements (Host ports) of the ATOLL simulation.
  3. Starts the SystemC simulation
  4. During simulation: data transfer between different host port objects

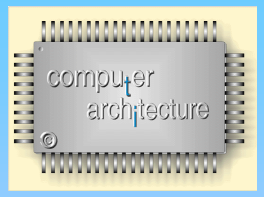


Hostport Class

- The Hostport (HP) objects are the interface between user-level application and the ATOLL network
  - Hostports are the Hardware communication endpoints of the ATOLL network
  - Applications send and receive messages by accessing HP registers
  - In DMA mode, the HP is responsible for the data transfer between the ATOLL network und the user application
- The SystemC model for the Host Ports:
  - Every Host Port contains a seperate SystemC process
  - HPs poll for the availibility of new send jobs
  - Calls the simulation object to transfer message data to the target HP
  - Provides all functionality for receiving messages and notifying the application that is currently attached to the HP



Each host port can only access its own memory regions like the real hardware

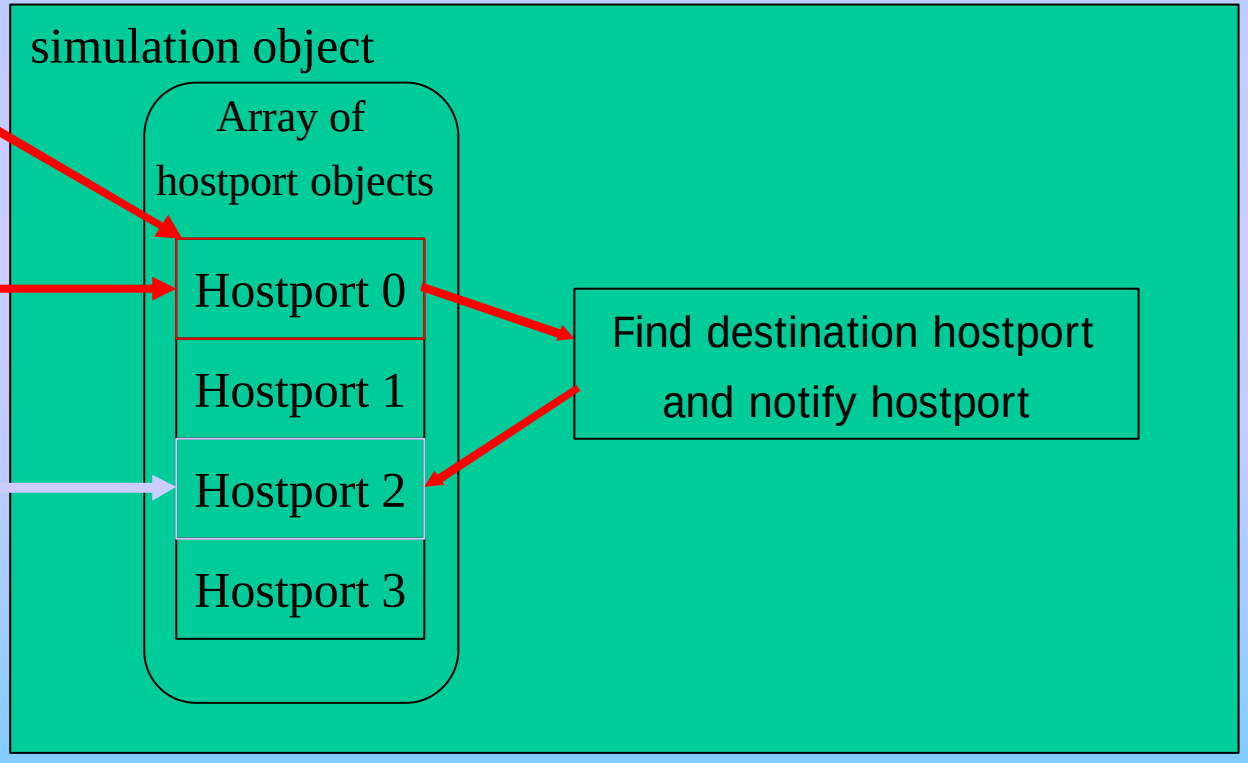


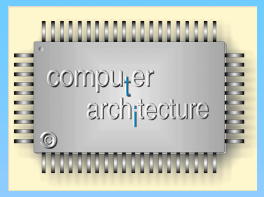
Message Transfer

Application 1:
send request
to application 2

Application 1
reserves host
port 0

Application 2
reserves host
port 2

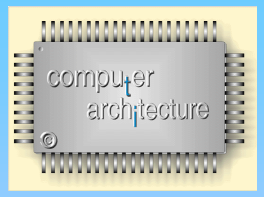




Results

- Easy-to-use simulation:
  1. Load the simulation driver
  2. Start the simulation process
  3. Start user application(s)
- Simulation of existing applications for the ATOLL device without re-linking
- Excellent methodology for Design Space Exploration of HW/SW Systems
- Modeling of the functionality with SystemC was straightforward (~1 month) compared to adapting the driver functionality (~2 month)

- The only weaknesses:
  - SystemC Library can not be ported on 64bit Opteron
  - Register access from software can only be detected by polling in a SystemC process



Outlook & Ongoing Development

- Future extension for Simulation of multiple nodes:
  - As with the real hardware, one SystemC simulation per node
  - Simulation Objects communicate with each other using a Message Passing API (e.g. MPI)
- Ongoing Development: HW/SW codesign for the next-generation System Area Network ExToll
  - Integration of a processor core developed with the LISATek Tool suite by CoWare
  - On-chip caches for accessing data and instructions and access to main memory in case of a cache miss
  - Parts of the design are clock-cycle accurate => mixed level design

Simulation Startup