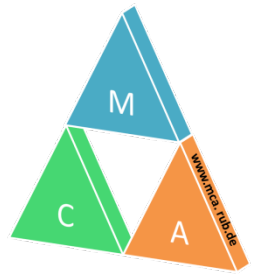
RUB

RUHR-UNIVERSITÄT BOCHUM

MPSoCSim: An extended OVP Simulator for Modeling and Evaluation of Network-on-Chip based heterogeneous MPSoCs

Philipp Wehner, Jens Rettkowski, Tobias Kleinschmidt, Diana Göhringer
Ruhr-University Bochum, Germany
{philipp.wehner, jens.rettkowski, tobias.kleinschmidt, diana.goehringer}@rub.de

FACULTY OF ELECTRICAL ENGINEERING AND INFORMATION TECHNOLOGY
Application-specific Multi-Core Architectures Group



Outline

- Motivation and Main Contributions
- MPSoCSim: An extension of OVP for modeling and evaluation of NoC based heterogeneous MPSoCs
- Comparison between Simulation and Hardware Implementation
- Evaluation in Example of Matrix Multiplication
- Conclusion and Future Prospects

Motivation

- Design space of NoC based systems is large
  - → Great diversity (topology, buffer, routing algorithm)
  - → Simulation useful
- NoC Simulator can be adjusted to meet requirements, e.g. performance or power for an application
- Processor models and traffic generators required
  - → Open Virtual Platform (OVP)

Motivation

- OVPSim
  - → Provides infrastructure for describing single/multi core platforms
  - → Platform creation based on simple C API
  - → C++, SystemC and TLM 2.0 wrappers available
  - → ARM, MIPS, openCores, OR1K, PowerPC, Xilinx MicroBlaze and Altera NIOS II processor models

- To meet the requirements of NoC based MPSoCs
  - → Extension necessary
  - → NoC functionality not provided by OVP

- OVPSim suitable as it provides various processor models that help to create heterogeneous MPSoCs

Motivation

- Main contribution of the paper is MPSoCSim

  → Extension of OVPSim with a Network-on-Chip

  → Transaction Level Modeling (TLM)-based interface

  → Support for traffic generators

  → Scalable mesh topology supporting wormhole routing

  → Routing algorithms: XY, minimal West-First and adaptive West-First

Motivation

- Main contribution of the paper is MPSoCSim

  → Modular structure of the router

  → API for bare-metal and Linux-based programs

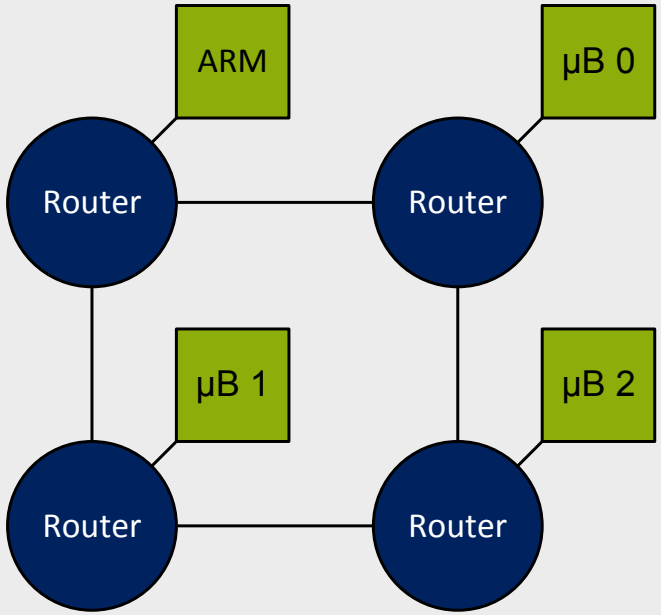
  → Access of simulation statistics of the network interfaces

- Simulation results of MPSoCSim are compared with a real HW implementation

→ MPSoCSim can be used for current and future NoC-based multi/many core systems

MPSoCSim

Overview



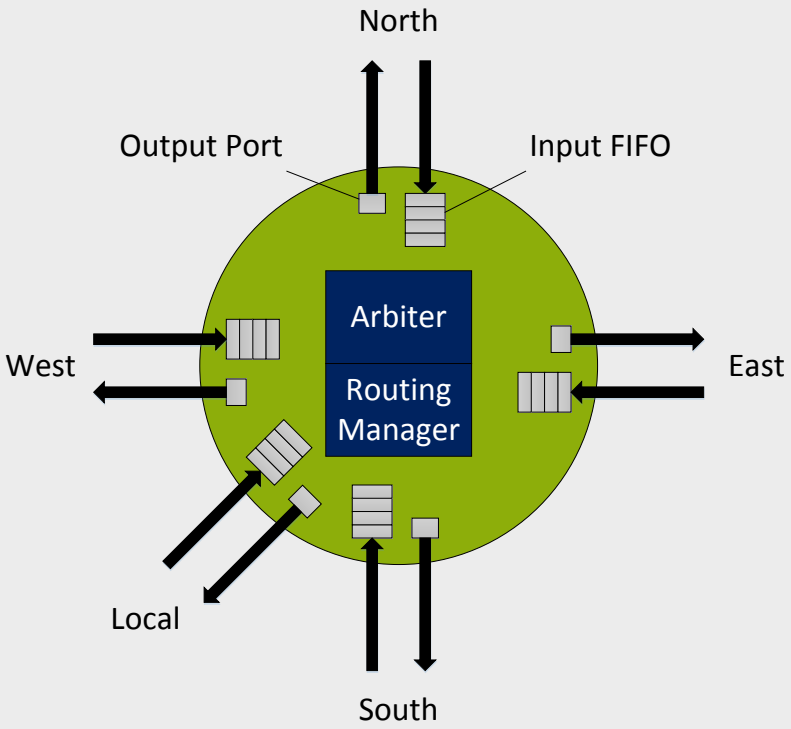
Cartesian mesh network

- Connection of the simulator to already existing SystemC platforms by OVP
- Loosely timed “LT” model
- Processor models are executed in SystemC threads
- Adjustable time delay between the runs of a processor model instance

MPSoCSim

Router Module

- Provides five connectors
- One target and one initiator socket per connector
- FIFO located in the target socket
- Router gets informed about available buffer memory by a signal
- The network interface is connected to the router using the local port

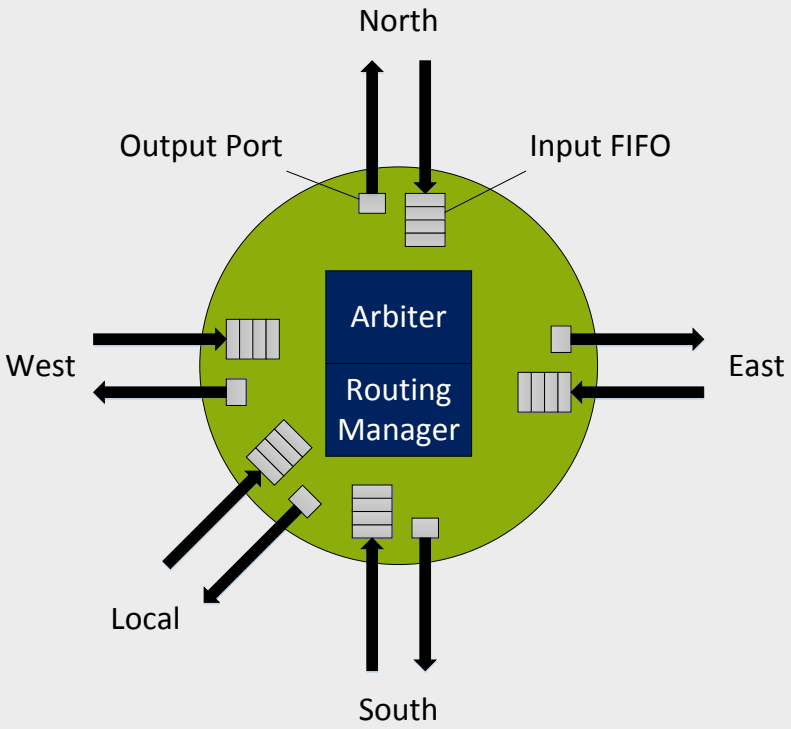


Router with internal components

MPSoCSim

Router Module

- Receiving a flit is handled by callback functions in the target sockets
  - → Request forwarded to routing manager
  - → Calculation of simulation time offset
  - → Request forwarded to arbiter
- Each flit has to await the simulation time offset until it is sent to the target
- In case of a full target FIFO: Flits have to wait until buffer memory is available



Router with internal components

MPSoCSim

Router Module

- Round robin arbitration used to assign output sockets
  - → Waiting list includes input sockets that request the output
  - → Possible routing directions forwarded to arbiter

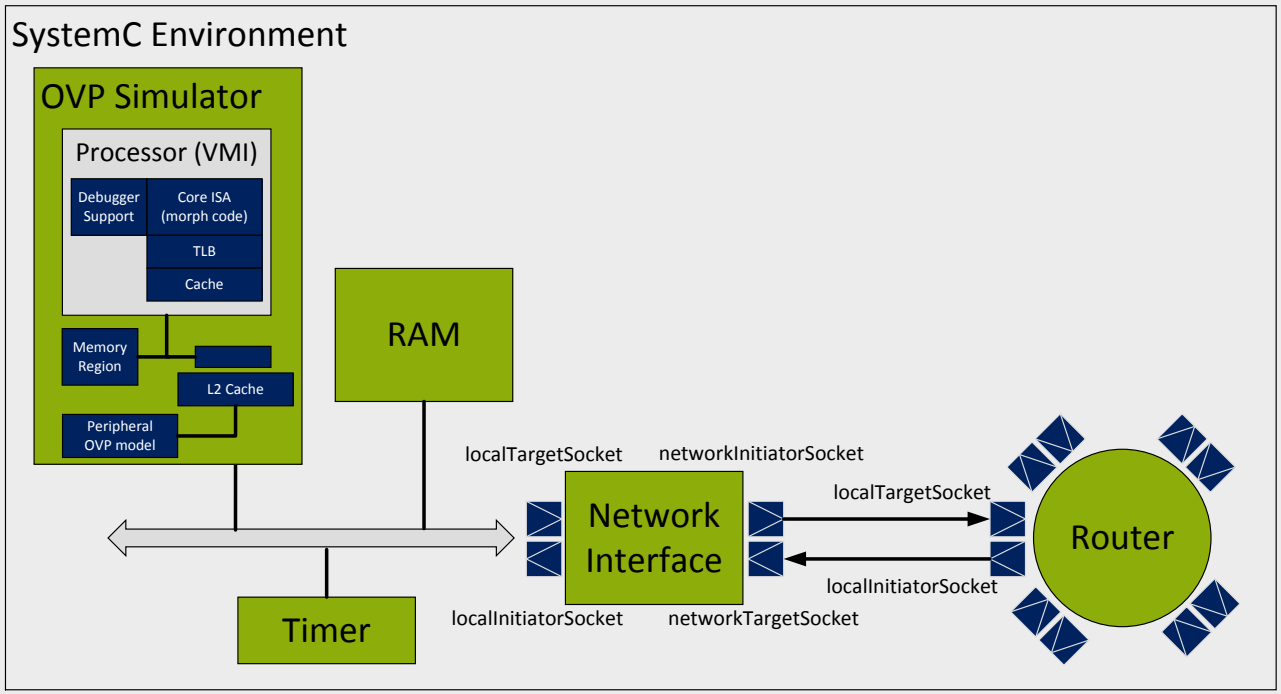
- Not every router needs to have a connection to processing elements or traffic generators
  - → During simulation, SystemC does not allow unbound sockets
  - → MPSoCSim automatically connects them to dummy elements

MPSoCSim

Network Interface

- Two initiator sockets, two target sockets and one port for the FIFO signal
- Acts equivalent to ordinary peripheral components

  → Addressed via a local bus



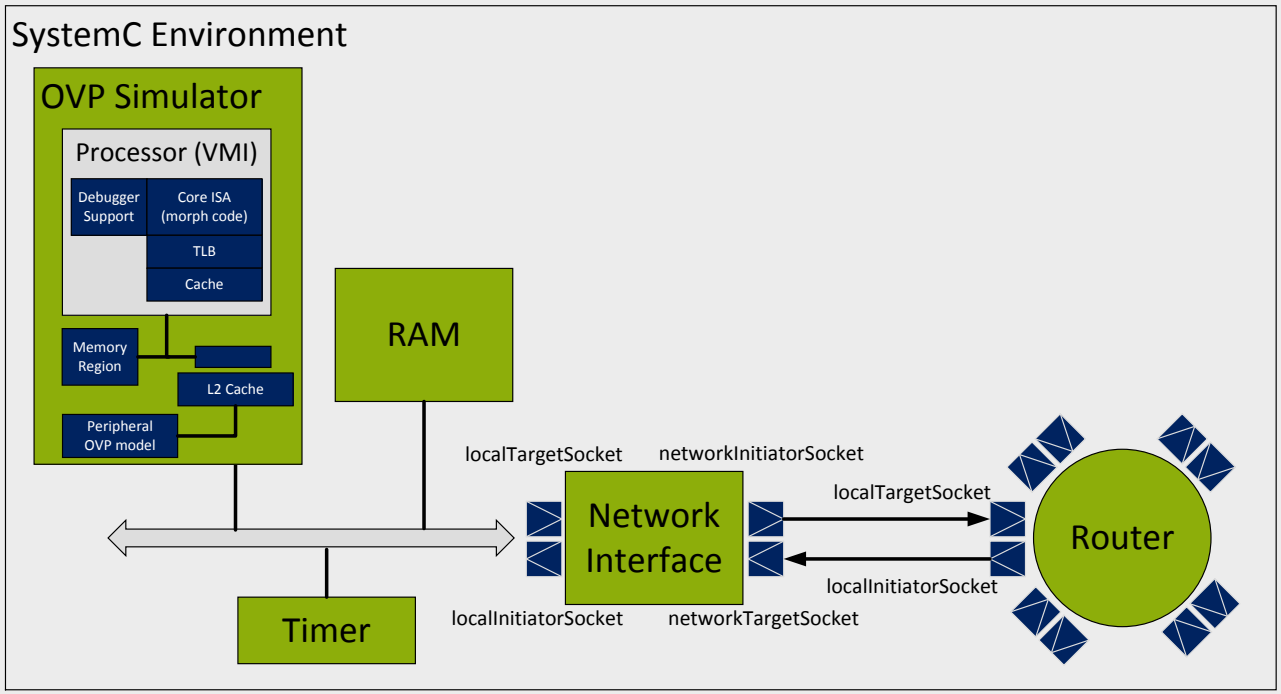
Extension of the OVP simulator

MPSoCSim

Network Interface

- Received data is stored in a memory, accessible by the local elements
- Stored at a known base address plus an offset defined by a header flit

  → Sender decides where data is placed in the memory

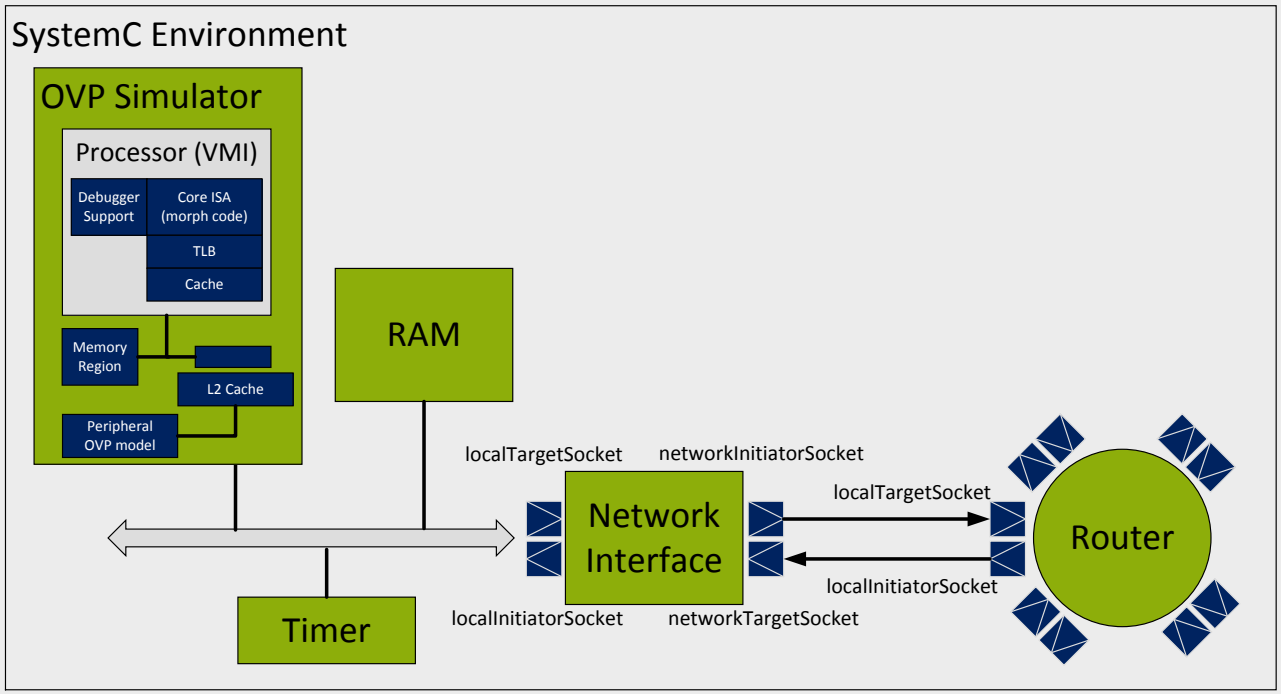


Extension of the OVP simulator

MPSoCSim

Network Interface

- A timer, counting simulation time, can be attached to a processing element
  → Accessible from within the application running on the processors
- Functions to start, stop, read and reset the timer are available



Extension of the OVP simulator

MPSoCSim

Traffic Generator

- Traffic generator as optional processing element
  - → Periodically sends messages to the network
  - → Enables performance and functional analyses
  - → Number of flits and data rate adjustable
  - → Start and stop functions available

- Messages are sent randomly to network nodes
  - → But a specific target address can be defined

MPSoCSim

Flits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Target				Size of Payload								Memory Address																			

Header flit

- Upper four bits identify the target network address
  → Adjustable according to the size of the NoC

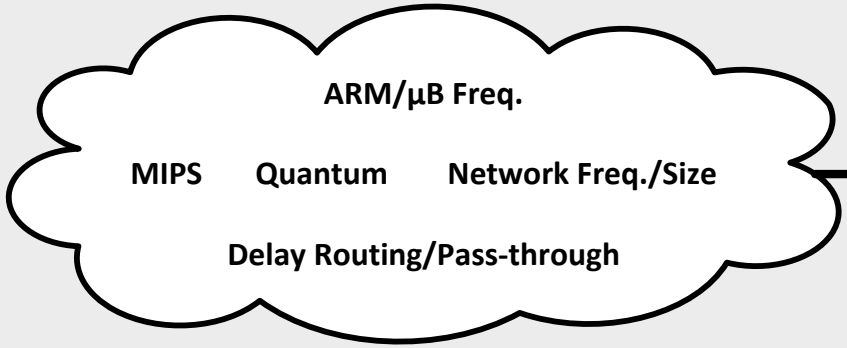
- Size of the payload is specified by the following eight bit
  → Additional tail flit not required

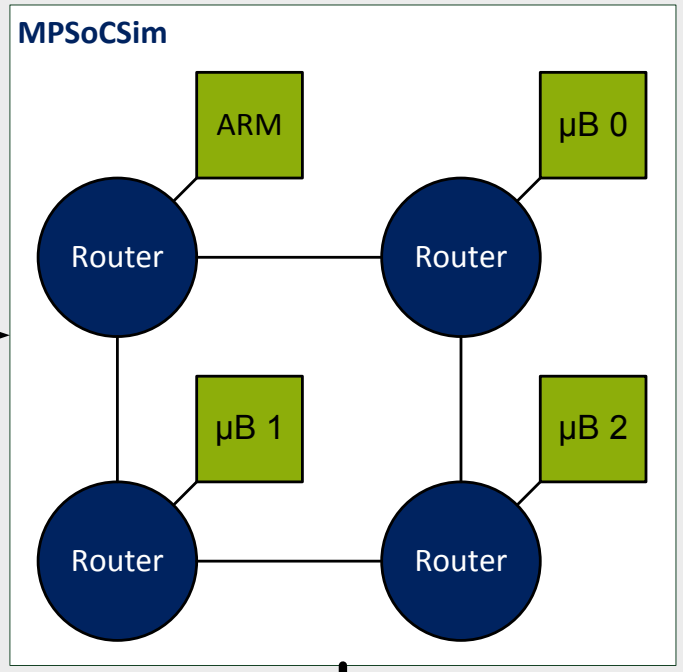
- Memory address (20 bits) needed by network interface to store data

- Network interface counts packets and included messages, sent and received data as well as the current and maximal data rate

MPSoCSim

Input/Output

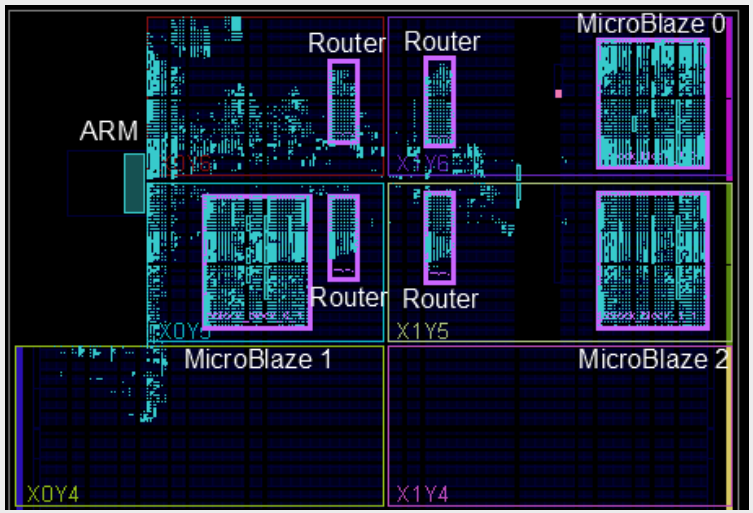




Statistics of NetworkInterface(0, 0):

number of messages received:	1461151
mean number of hops:	3.299563
mean delay:	3.566966 µs
bytes received:	89.181580 MiByte
bytes transmitted:	85.283936 MiByte
max data rate received:	114.168440 MiByte/s
max data rate transmitted:	101.634434 MiByte/s
average data rate received:	84.934838 MiByte/s
average data rate transmitted:	81.222796 MiByte/s

Hardware Implementation



Implemented MPSoC on a ZC706 Board (xc7z045ffg900-2)

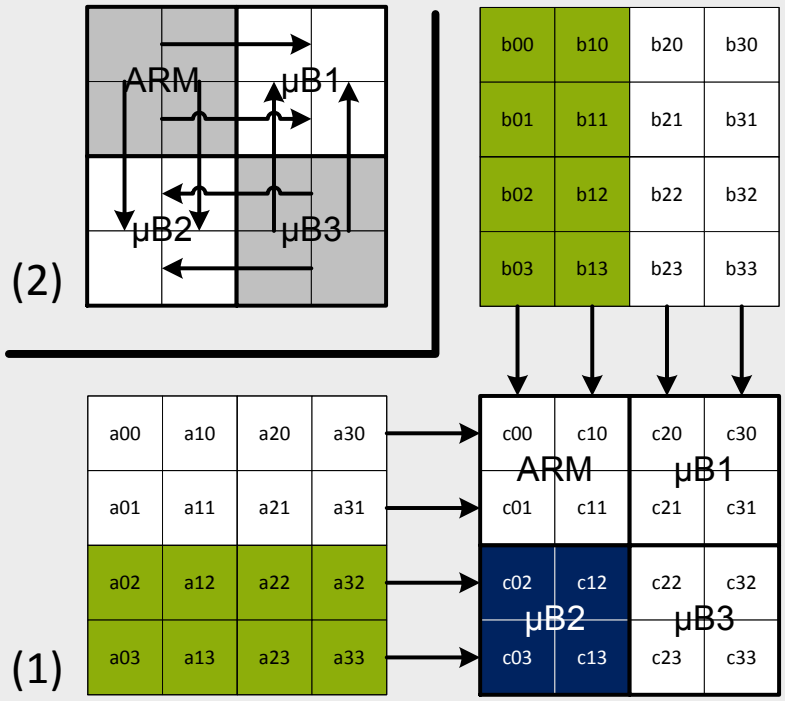
- MPSoC implemented on Xilinx Zynq
  - → ARM processor besides FPGA
  - → FPGA contains three MicroBlazes and the Network on Chip
  - → 32 bits flit size, equal to simulation
  - → 667 MHz (ARM) and 100 MHz (FPGA)
- ARM communicates via high performance (HP) port with the network
- MicroBlazes are linked via FSL interface to the NoC

Evaluation

Matrix Multiplication

- Enables the analysis of the simulator compared to real hardware

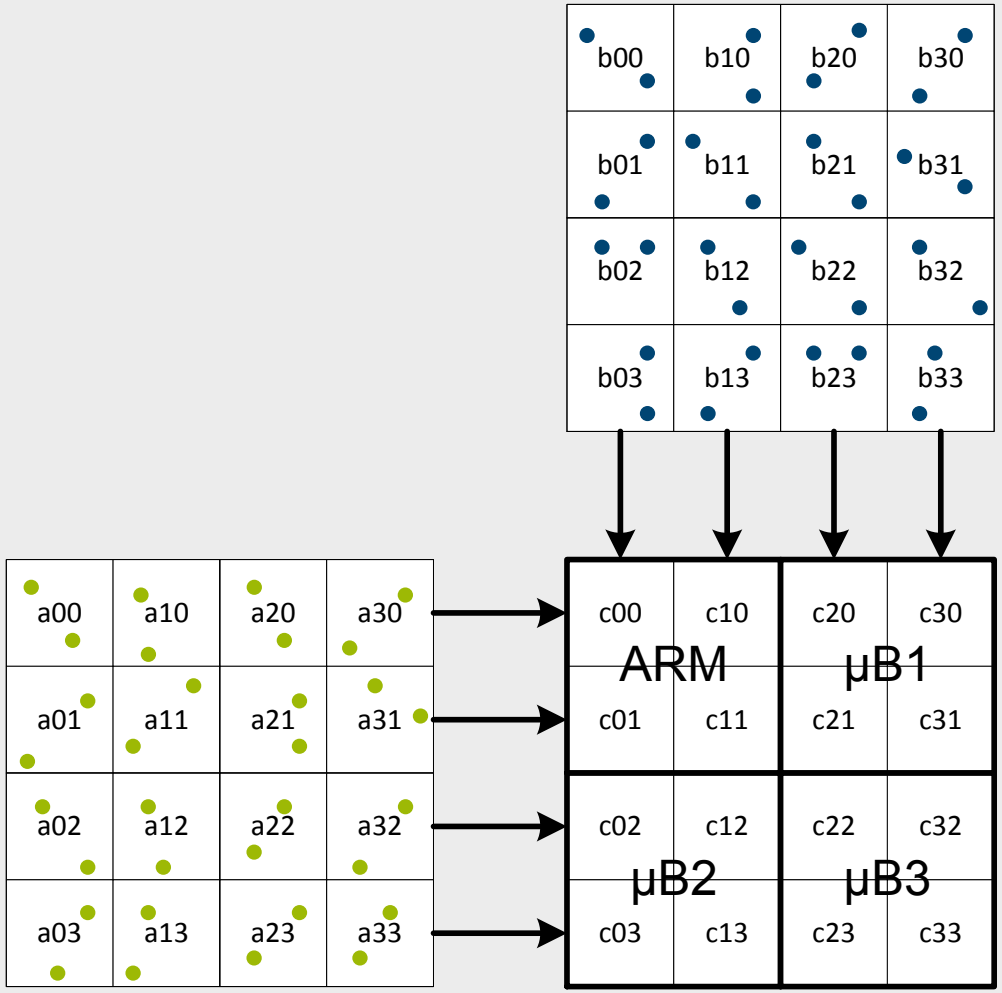
- Two versions:
  1. Involved matrices divided equally and sent to the MicroBlazes by the ARM processor
  2. Rows and columns only sent to the MicroBlazes that are responsible for the diagonal elements of the output matrix



Tiled matrix multiplication in (1) normal and (2) diagonal mode

Evaluation

Matrix Multiplication



Tiled matrix multiplication

Evaluation

Matrix Multiplication

Size	64x64		32x32		16x16		8x8		4x4	
	Std.	Diag.	Std.	Diag.	Std.	Diag.	Std.	Diag.	Std.	Diag.
Sim. time (in msec)										
Simulated time	190	180	30	30	10	0	0	0	0	0
User time	390	380	120	120	60	60	40	50	40	40
System time	10	0	0	0	0	10	0	0	0	0
Elapsed time	390	380	120	120	60	60	40	50	40	40
Sim. exec. time	112,85	105,94	15,86	14,13	2,56	2,15	0,50	0,38	0,11	0,09

Simulated execution time: Time measured by the timer module during execution

User time: Time spent for the execution on the host machine

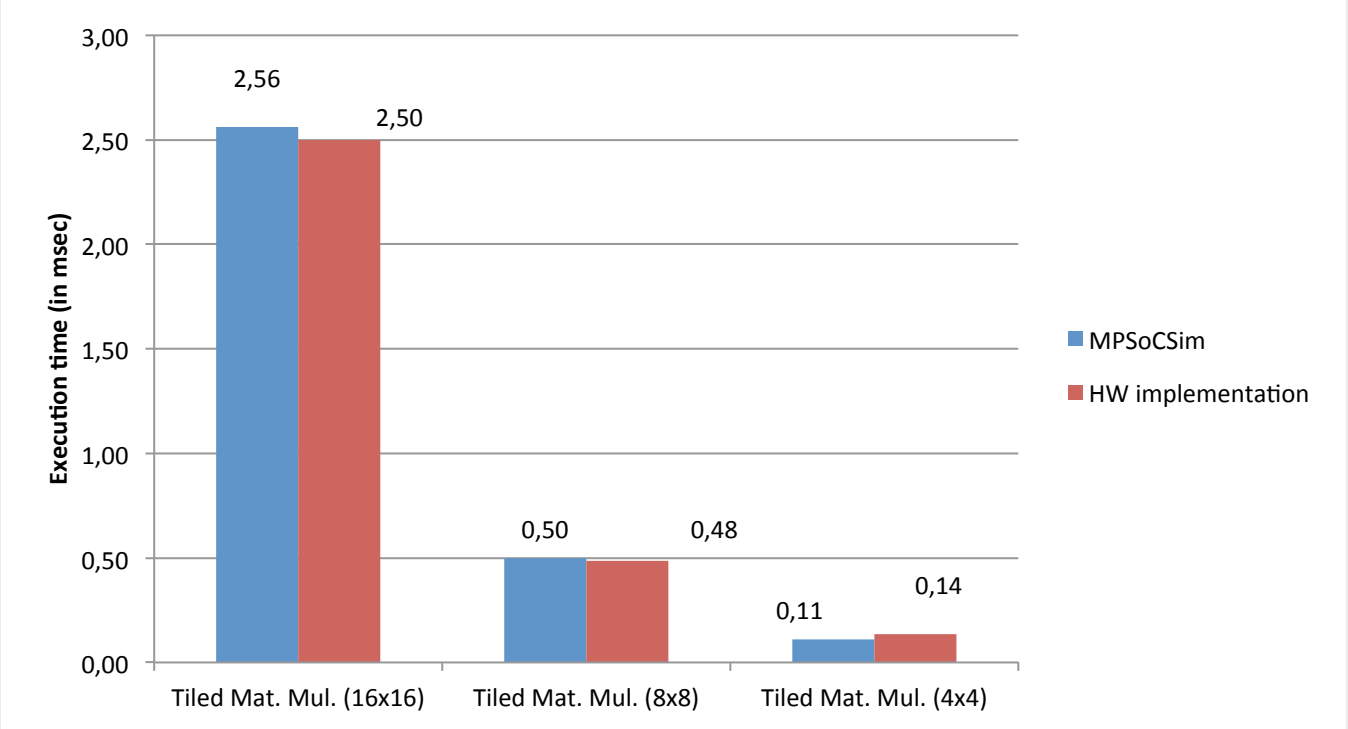
System time: Spent by the host machine to execute instr. of the simulation process

Elapsed time: Simulation time from beginning to end

Simulated time: Duration of the simulation process in simulated time

Evaluation

Matrix Multiplication



Comparison between MPSoCSim and the real HW implementation

- Maximum relative deviation (sim. execution time in relation to the execution time on hardware) is 17,4% for the 4x4 matrix and 2,5% for the multiplication of 16x16 matrices

Conclusion
And Future Prospects

- MPSoCSim: Simulator for modeling and evaluation of NoC-based heterogeneous MPSoCs
  - → Supports several processor models
  - → Flexible TLM2.0 communication infrastructure
  - → Access to simulation statistics of the network interfaces
  - → Support for operating systems running on the processors
- Relative deviation of 2,5% for the multiplication of 16x16 matrices compared to real hardware implementation

- In future work, MPSoCSim will be extended with simulation for power and further NoC topologies
  - → Performance metrics and further traffic patterns
- MPSoCSim will be tested with further applications/benchmarks

Conclusion

Related Work

Overview of Simulation Platforms for NoC-based MPSoC

Simulator	Modelling Language	Communication Infrastructure	Topology	Parameterizable	Processing Elements	Simulation Results
Nirgam [4]	SystemC	Network-on-Chip	Mesh, Torus, Butterfly etc.	Yes	Traffic Generators, Application Core	Performance, Power
Noxim [5]	SystemC	Network-on-Chip	Mesh	Yes	Traffic Generators	Performance, Power
Booksim 2.0 [6]	C++	Network-on-Chip	Mesh, Torus, Butterfly etc.	Yes	Traffic Generators	Performance
HNOCS [7]	C++	Network-on-Chip	Arbitrary	Yes	Traffic Generators	Performance, Power
Rosa et al. [8]	SystemC	Bus	-	-	OVP processor models	Performance, Power
MPSoCBench [9]	SystemC, ArchC	Network-on-Chip	Mesh	No	PowerPC, MIPS, SPARC, ARM	Performance, Power
MC-Sim [10]	C	Network-on-Chip	Mesh	Yes	MIPS	Performance
MPARM [12]	SystemC	Bus	-	-	ARM	Performance, Power
MPSoCSim	SystemC	Network-on-Chip	Mesh	Yes	Traffic Generators, OVP processor models	Performance

RUB

Thank you for your attention!