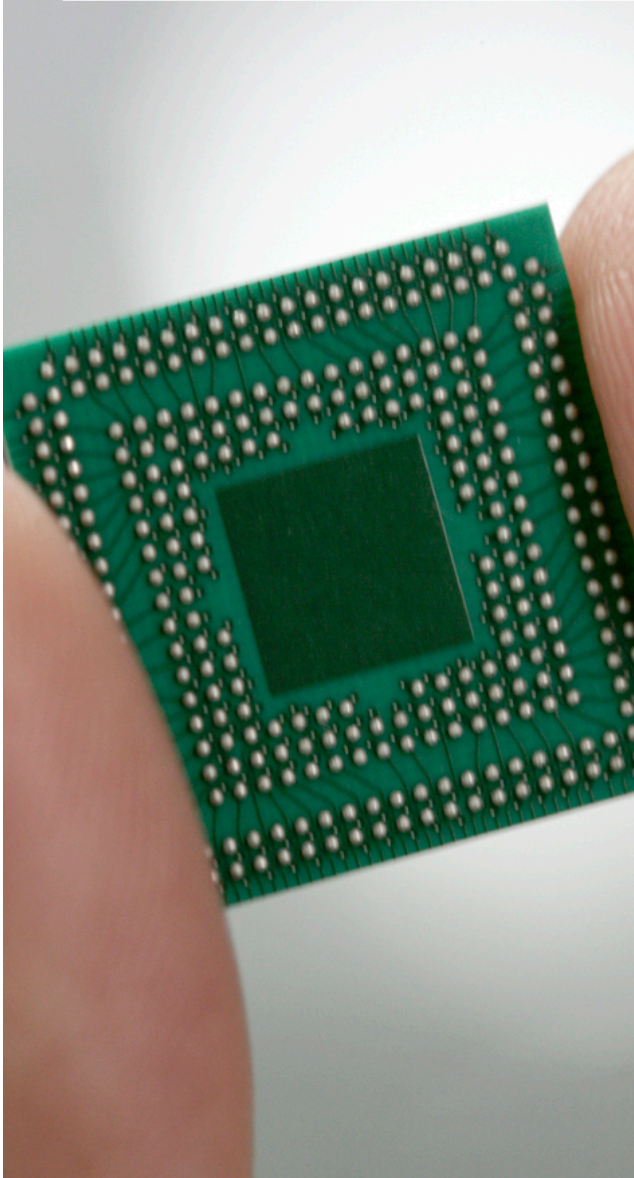


# MultiCore - SingleBus

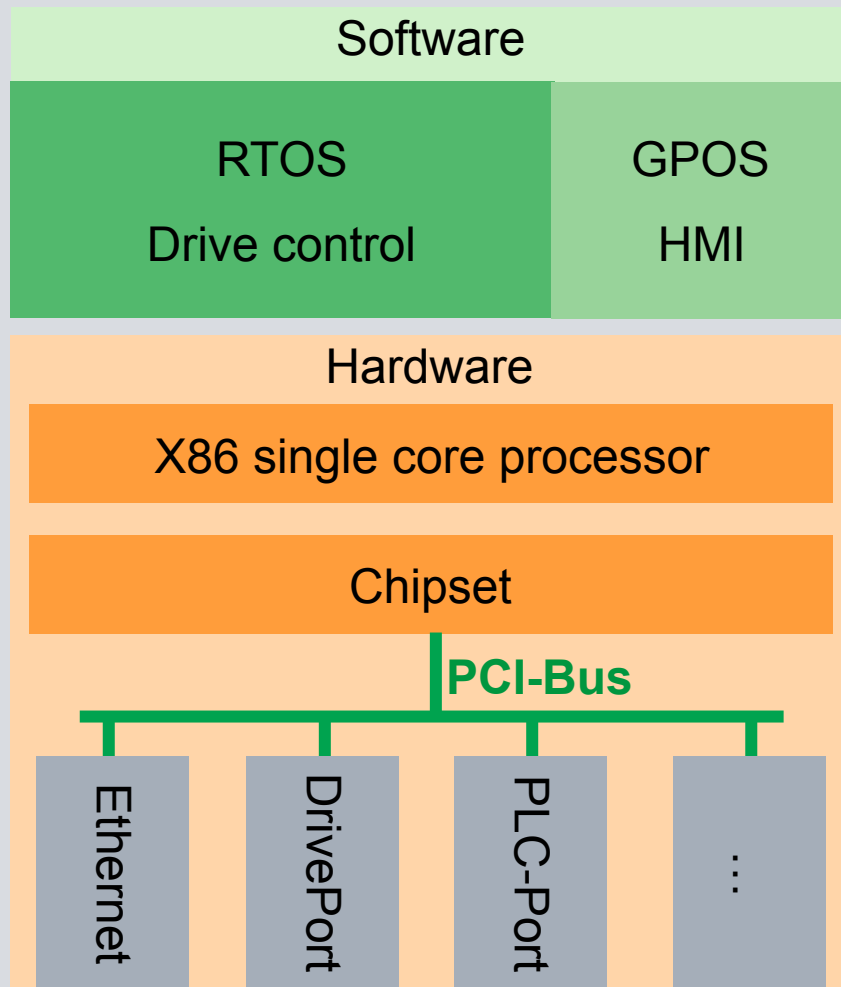
Mutual Influence between Processors and  
Peripherals



## Overview

- System description
- Modeling of the system
- Analysis
- Results
- Summary and outlook

## System description



### Two main parts

- Real time drive application
- GPOS based HMI

### Separated by hardware timer

- 70% RTOS
- 30% GPOS

### Peripherals on PCI

- Ethernet ports
- Drive connections
- PLC connection
- ...

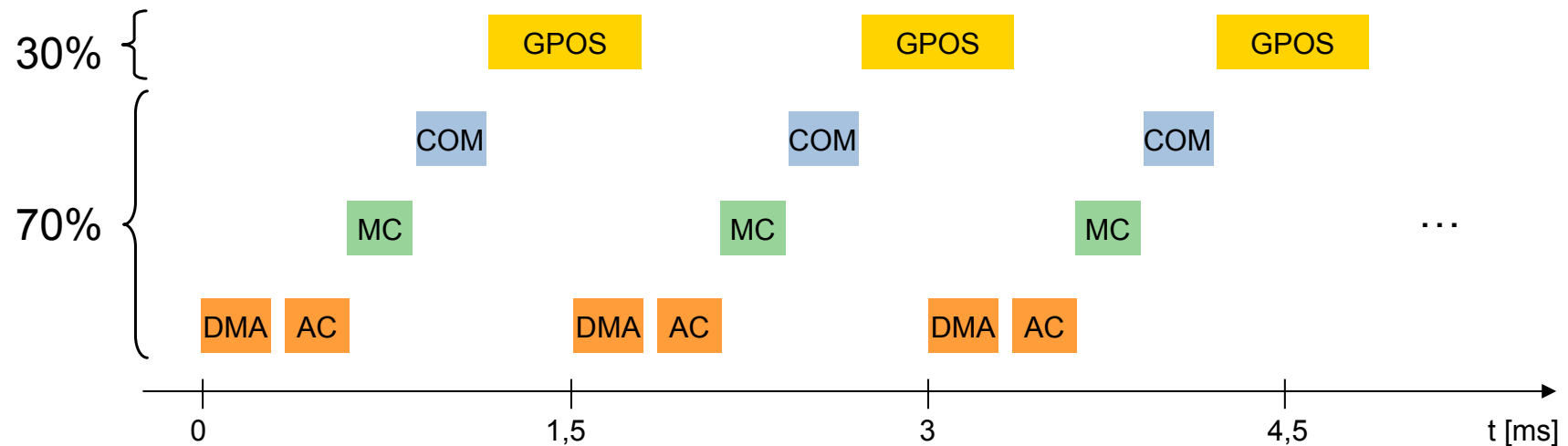
## Software structure

### RTOS tasks

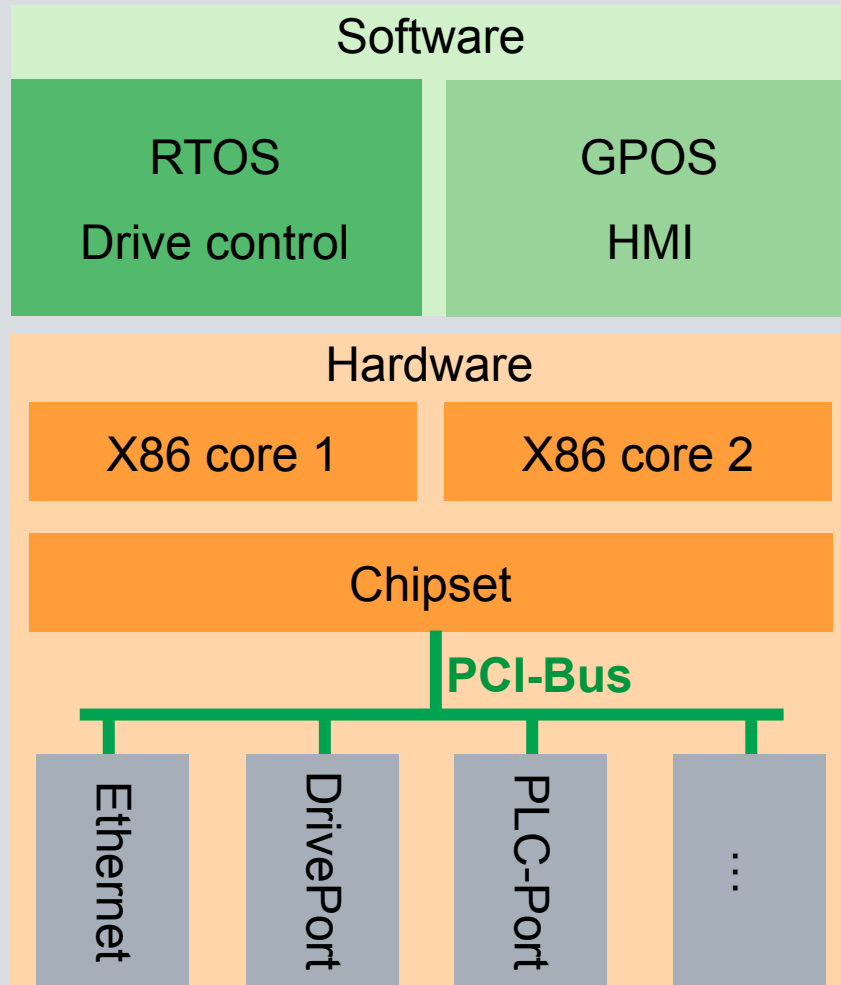
- Axis controller (AC)
- Motion controller (MC)
- Communication (COM)

### GPOS/HMI tasks

- Communication (HMI data)
- Visualization



## New system design



### Two main parts

- Real time drive application
- GPOS based HMI

### Separated by core affinity

- 100% RTOS
- 100% GPOS

### Peripherals on single PCI bus

- Ethernet ports
- Drive connections
- PLC connection
- ...

## Expectations vs. reality

### Expectations

- GPOS-HMI runs up to three times faster
  - More throughput on the network
  - Better reaction of the GUI
- Real time worst case runtime nearly unchanged

### Reality

- GPOS HMI runs significantly faster
- BUT: Real time side runs up to three times slower than before!!!!

→ System analysis by replacing HMI by well-defined load tests

## Reality figures

### Scenario

- First analysis leads to collisions on PCI bus → further investigation
- RTOS software runs as it is, MC can run 1, 2 or 4 times slower than AC
- GPOS generates load on PCI bus by copying 4k blocks

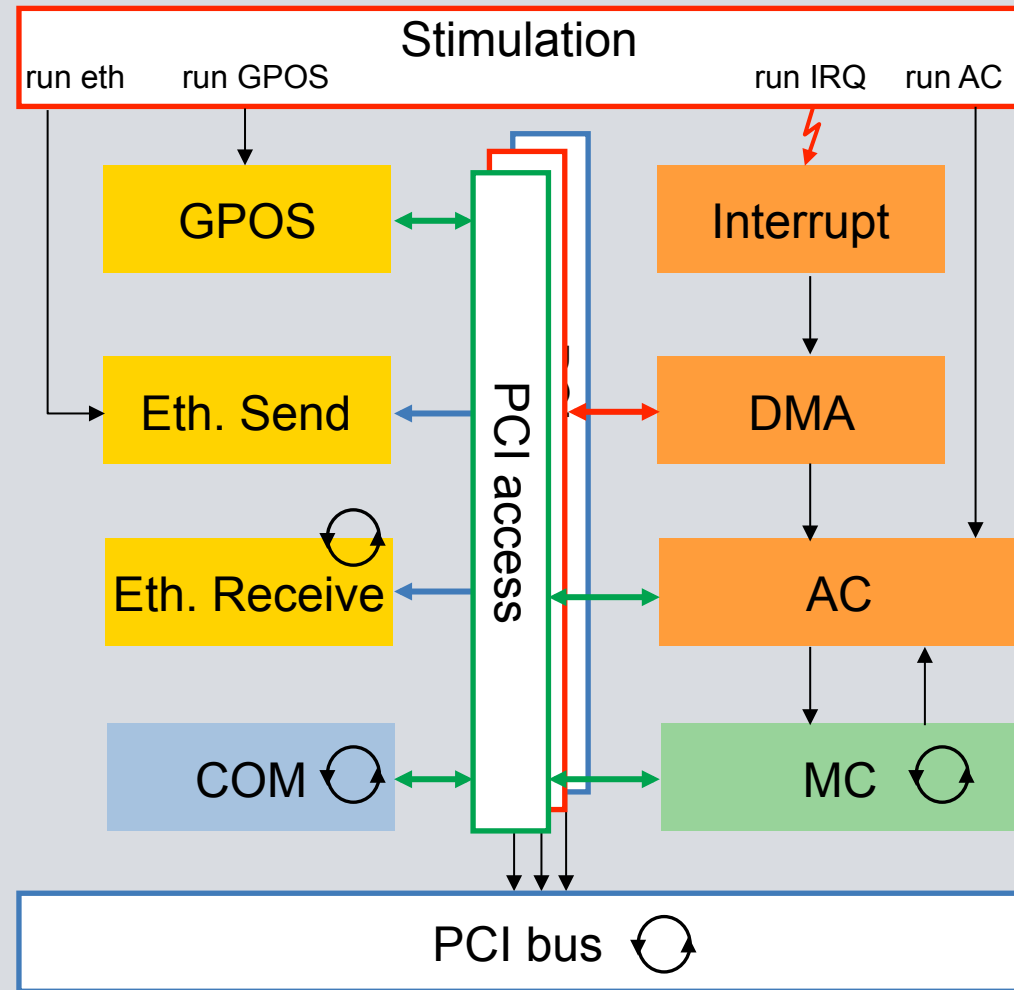
AC 1.5 ms		MC 1.5 ms		MC 3.0 ms		MC 6.0 ms	
		max.AC	max.MC	max.AC	max.MC	max.AC	max.MC
1 Core	Idle state	110 $\mu$ s	365 $\mu$ s	110 $\mu$ s	365 $\mu$ s	110 $\mu$ s	365 $\mu$ s
1 Core	PCI load	<b>110 <math>\mu</math>s</b>	<b>365 <math>\mu</math>s</b>	<b>110 <math>\mu</math>s</b>	<b>365 <math>\mu</math>s</b>	<b>110 <math>\mu</math>s</b>	<b>365 <math>\mu</math>s</b>
2 Core	Idle state	110 $\mu$ s	365 $\mu$ s	110 $\mu$ s	365 $\mu$ s	106 $\mu$ s	365 $\mu$ s
2 Core	PCI load	<b>235 <math>\mu</math>s</b>	<b>925 <math>\mu</math>s</b>	<b>233 <math>\mu</math>s</b>	<b>930 <math>\mu</math>s</b>	<b>232 <math>\mu</math>s</b>	<b>920 <math>\mu</math>s</b>

→ System modeling necessary to identify problems

# Modeling of tasks

## Modeling with ChronSim

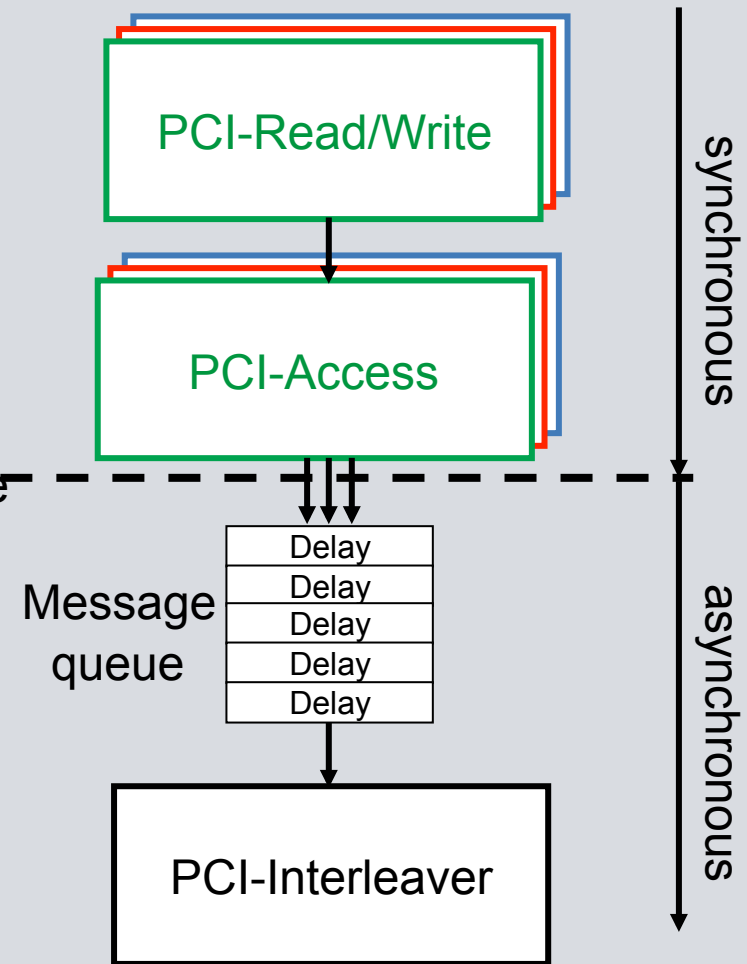
- Cyclic activation of tasks
- Synchronization with message queues
- Three channels for PCI to simulate arbitration logic
- Resources
  - CPU (GPOS, AC, MC, COM)
  - Ethernet (send / receive)
  - PCI bus
  - DMA (read)



## Modeling of PCI bus

### PCI bus is an independent resource

- Message queue to serialize accesses
- Queuing from three different channels
  - Mixing accesses from different sources
  - Leads to significant delays on collision
  - Various arbitration logic strategies possible
- Interleaver idles corresponding to data size
  - Normal read: 2.3 MB/s
  - Normal write: 16 MB/s
  - DMA (burst): 40 MB/s (max 128 byte)



## Model vs. reality

### Green numbers show deviation from reality figures

- Model shows significant raise of runtime for real time tasks

### Can be used for further investigation

- Different arbitration logic
- Software controlled access of non-real time tasks to PCI bus

AC 1.5 ms		MC 1.5 ms		MC 3.0 ms		MC 6.0 ms	
		max.AC	max.MC	max.AC	max.MC	max.AC	max.MC
1 Core	Idle state	108 $\mu$ s	369 $\mu$ s	109 $\mu$ s	370 us	110 us	363 $\mu$ s
	Deviation	-2 $\mu$ s	+4 $\mu$ s	-1 $\mu$ s	+5 $\mu$ s	0 $\mu$ s	-2 $\mu$ s
1 Core	PCI load	112 $\mu$ s	384 $\mu$ s	116 $\mu$ s	380 $\mu$ s	122 $\mu$ s	378 $\mu$ s
	Deviation	+2 $\mu$ s	+19 $\mu$ s	+6 $\mu$ s	+15 $\mu$ s	+12 $\mu$ s	+13 $\mu$ s
2 Core	Idle state	108 $\mu$ s	367 $\mu$ s	109 $\mu$ s	366 $\mu$ s	114 $\mu$ s	367 $\mu$ s
	Deviation	-2 $\mu$ s	+2 $\mu$ s	-1 $\mu$ s	+1 $\mu$ s	+8 $\mu$ s	+2 $\mu$ s
2 Core	PCI load	187 $\mu$ s	904 $\mu$ s	190 $\mu$ s	885 $\mu$ s	166 $\mu$ s	860 $\mu$ s
	Deviation	-48 $\mu$ s	-21 $\mu$ s	- 43 $\mu$ s	-45 $\mu$ s	- 66 $\mu$ s	- 60 $\mu$ s

## Measures

- Reducing of PCI read
  - Reading take a lot more time than writing
  - Peripherals should write data into main memory, if possible
  - Usage of DMA
- Use of software locks
  - GPOS has to wait, until RTOS has finished PCI access
  - Drawback: Software needs a lot of changes
- Other aspects
  - Cache was also an issue
  - No modelling possible because of lack of knowledge on cache behavior

## Conclusions

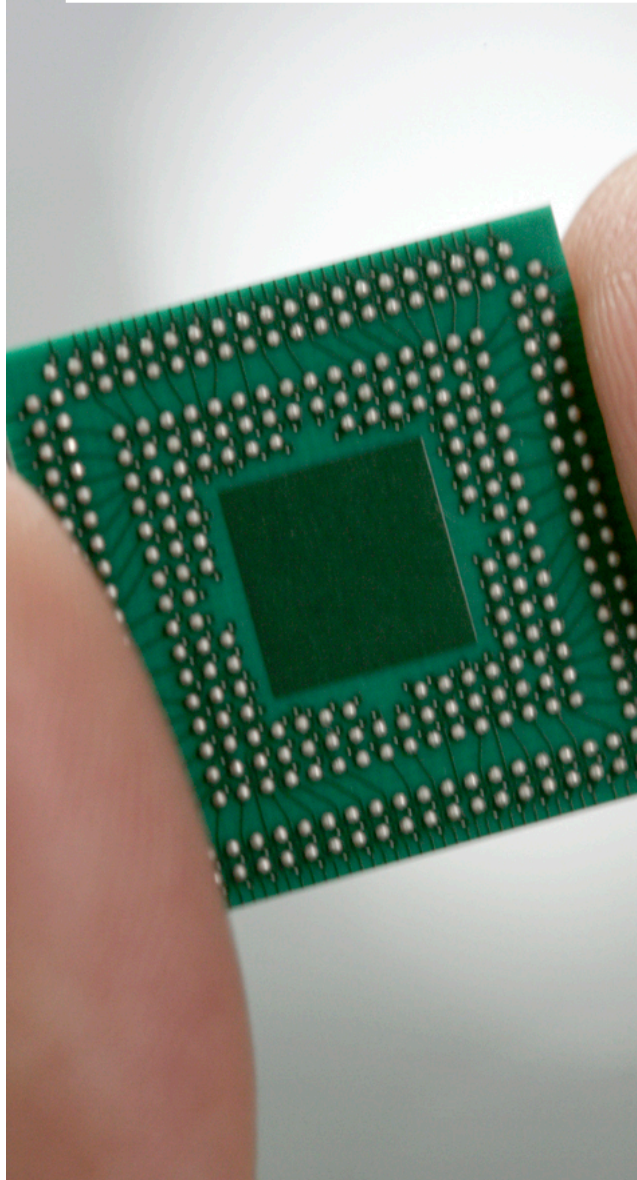
### Pro

- Simulation can help to find bottlenecks
- Easy use of modeling and simulation tools
- Integrated diagrams and reporting
- Variation of system parameters to evaluate different solutions

### Contra

- Model should be used early in the development
- Hardware model not available, but simulated by software
- Long calculation times

**Thank you for your attention!**



**Dr. René Graf**  
Group leader  
I IA IA&DT-ATS 1 1

Gleiwitzer Straße 555  
90475 Nürnberg

Phone: +49 (911) 895-3179  
Fax: +49 (911) 895-3715  
Cellular: +49 (173) 3013668

E-Mail: [Rene.Graf@siemens.com](mailto:Rene.Graf@siemens.com)