A Hardware/Software Co-Design Approach for Ethernet Controllers to Support Time-triggered Traffic in the Upcoming IEEE TSN Standards

Friedrich Groß Till Steinbach
Franz Korf Thomas C. Schmidt Bernd Schwarz

Hamburg University of Applied Sciences
{friedrich.gross, till.steinbach, korf, schmidt, schwarz}@informatik.haw-hamburg.de

4th IEEE International Conference on Consumer Electronics - Berlin
September 8th, 2014

SPONSORED BY THE
Federal Ministry of Education and Research

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Agenda

1 Introduction & Motivation

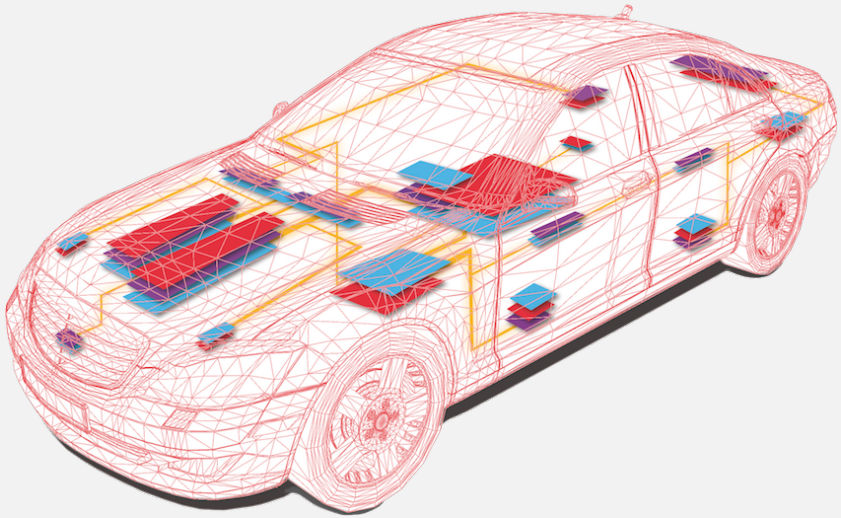
2 Concept & Results

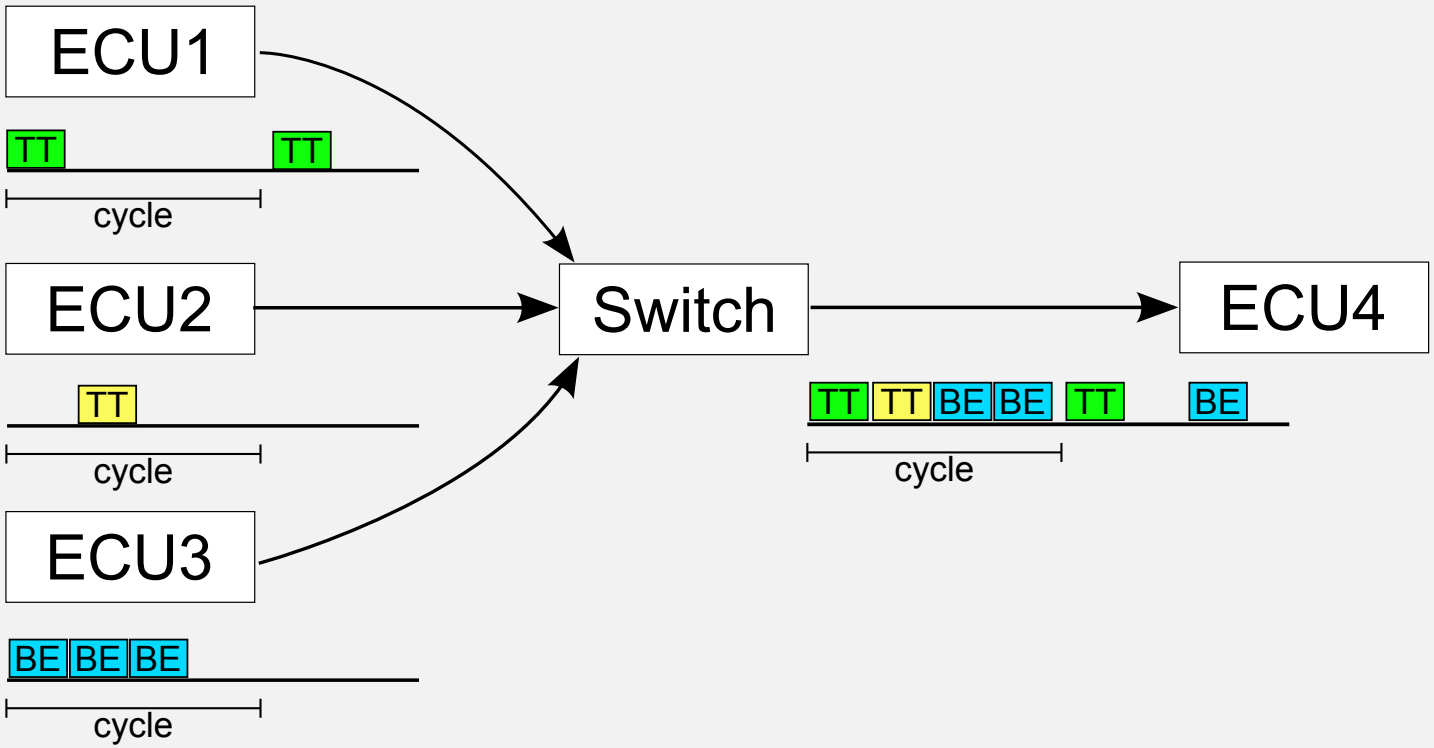
3 Conclusion & Outlook

Motivation

Why is Time-Triggered Ethernet needed?

- Modern cars: $>$ 70 ECUs; about 2500 message types
- Bandwidth and timing requirements increase
- Now used communication systems came to their limits due they are not scalable
- Next generation backbones will most likely base on real-time Ethernet




ECU1
TT
TT
cycle
ECU2
Switch
ECU4
TT
cycle
TT
TT
BE
BE
TT
BE
cycle
ECU3
BE
BE
BE
cycle

- Real-time extension for standard Ethernet
- Deterministic behavior, low latency and jitter
- Clock synchronisation and special switches are needed
- TTEthernet, Profinet IRT, upcoming IEEE 802.1Qbv, ...
- This work focuses TTEthernet AS6802

Motivation
Why Hardware/Software Co-Design?

- Reduce computational power
  - On a high network load up to 90% of a CPU is used (ARM9 @ 200Mhz) [1]
  - One reception buffer for all traffic-classes
  - Every received frame must be handled immediately for garbage collection
- Reduce timing requirements for OS
  - CAN-Bus and FlexRay achieved good results with HW/SW Co-Design

[1] K. Müller "A Real-time Ethernet Prototype Platform for Automotive Applications," in 2011 ICCE-Berlin

Contribution

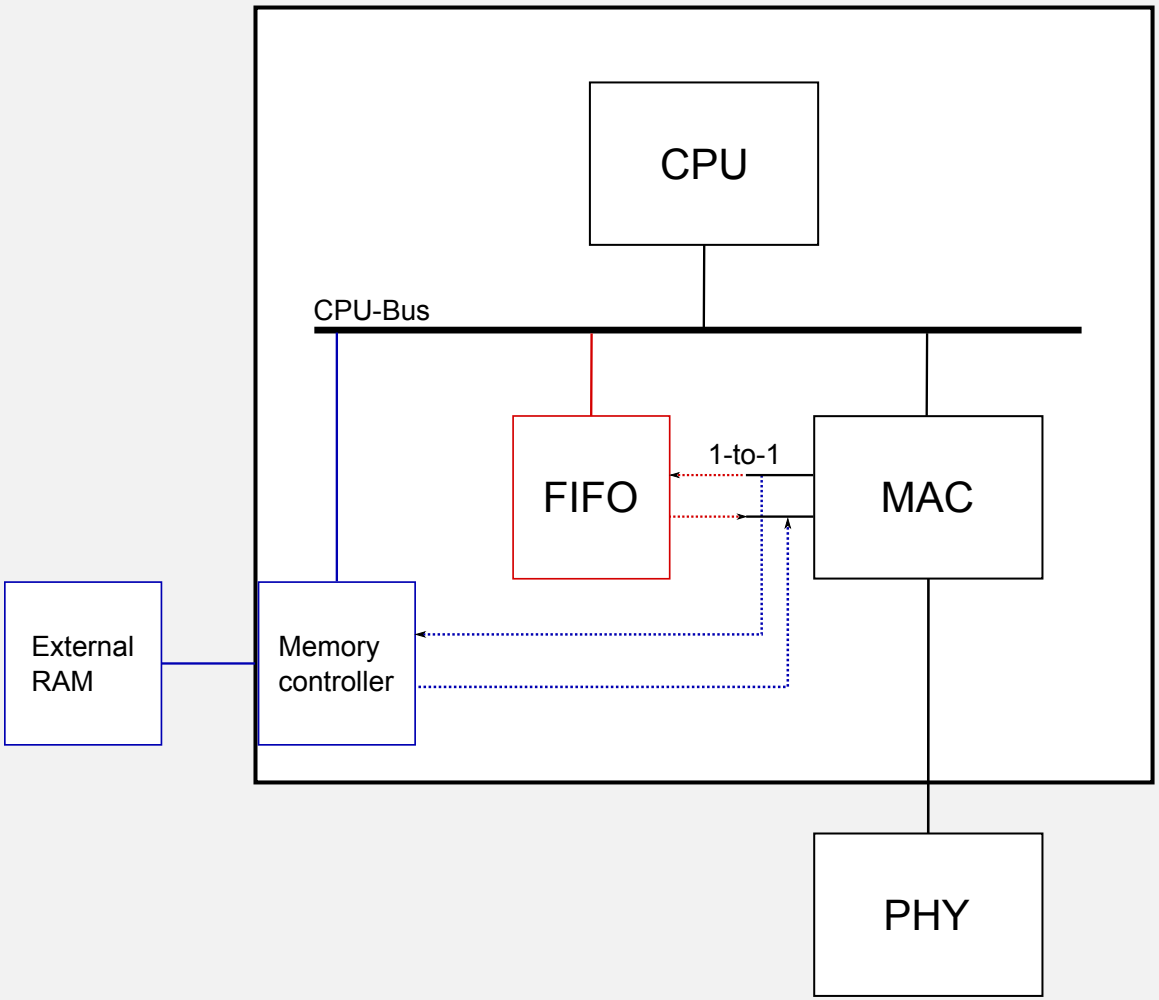
- Scalable HW/SW Co-Design TTEthernet Controller
- Include clock synchronisation
- Results of a prototype implementation on a FPGA

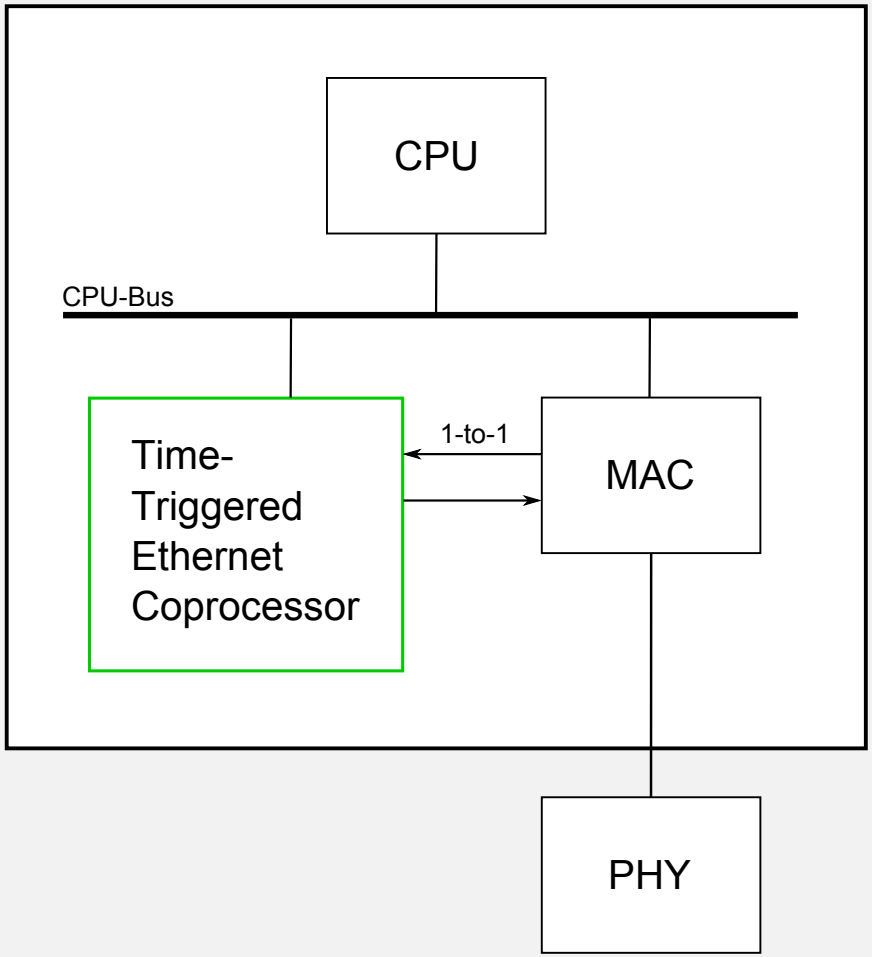
Agenda

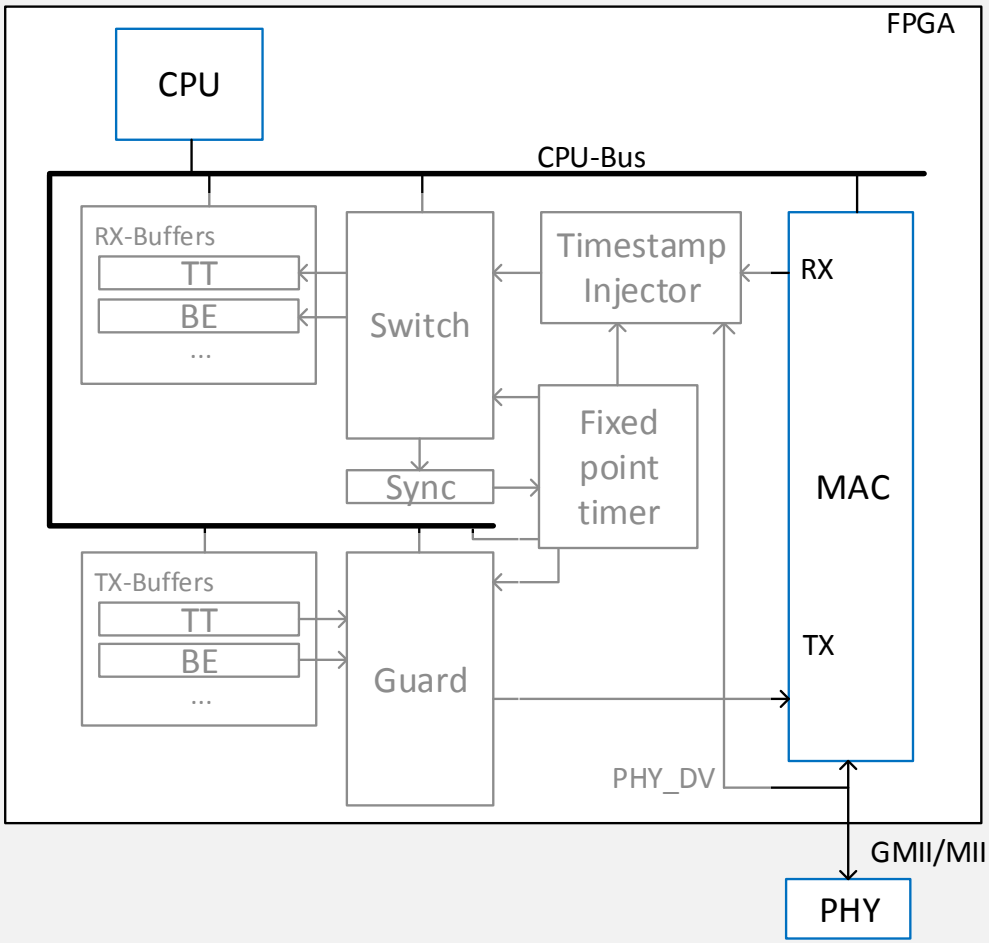
1 Introduction & Motivation

2 Concept & Results

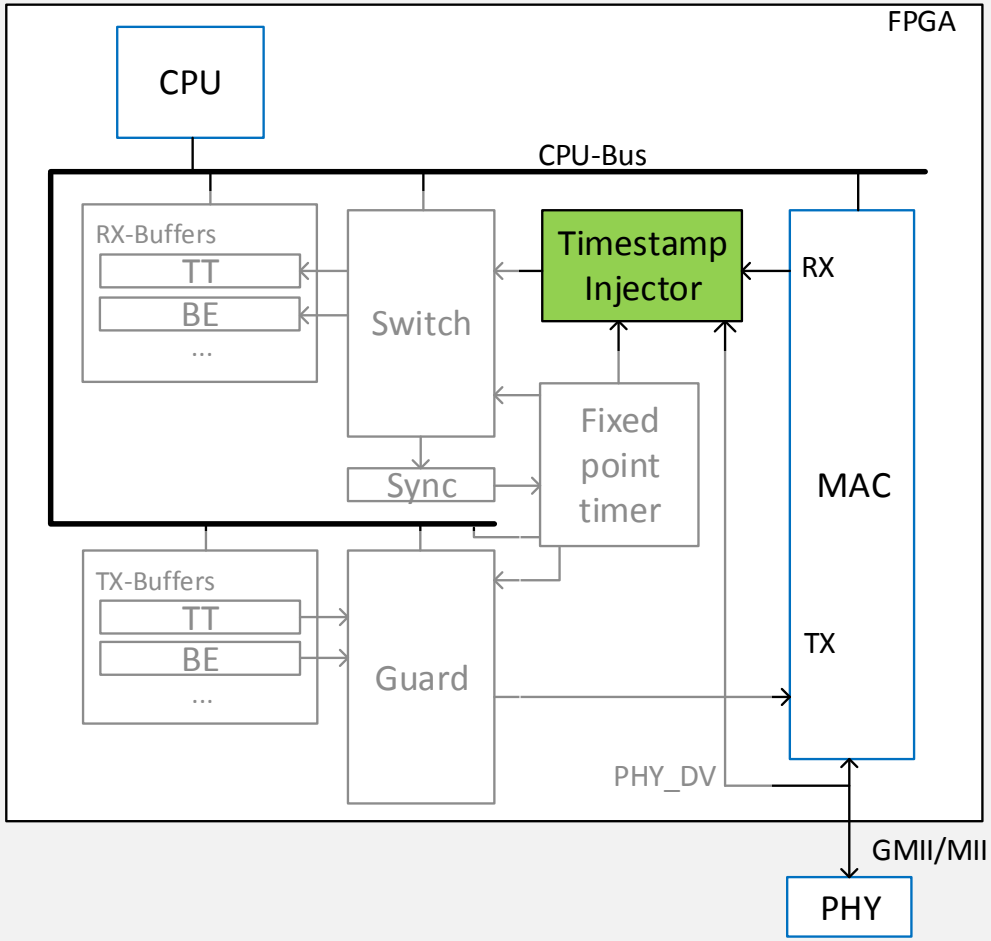
3 Conclusion & Outlook


CPU
CPU-Bus
FIFO
1-to-1
MAC
External RAM
Memory controller
PHY


CPU
CPU-Bus
Time-
Triggered
Ethernet
Coprocessor
1-to-1
MAC
PHY

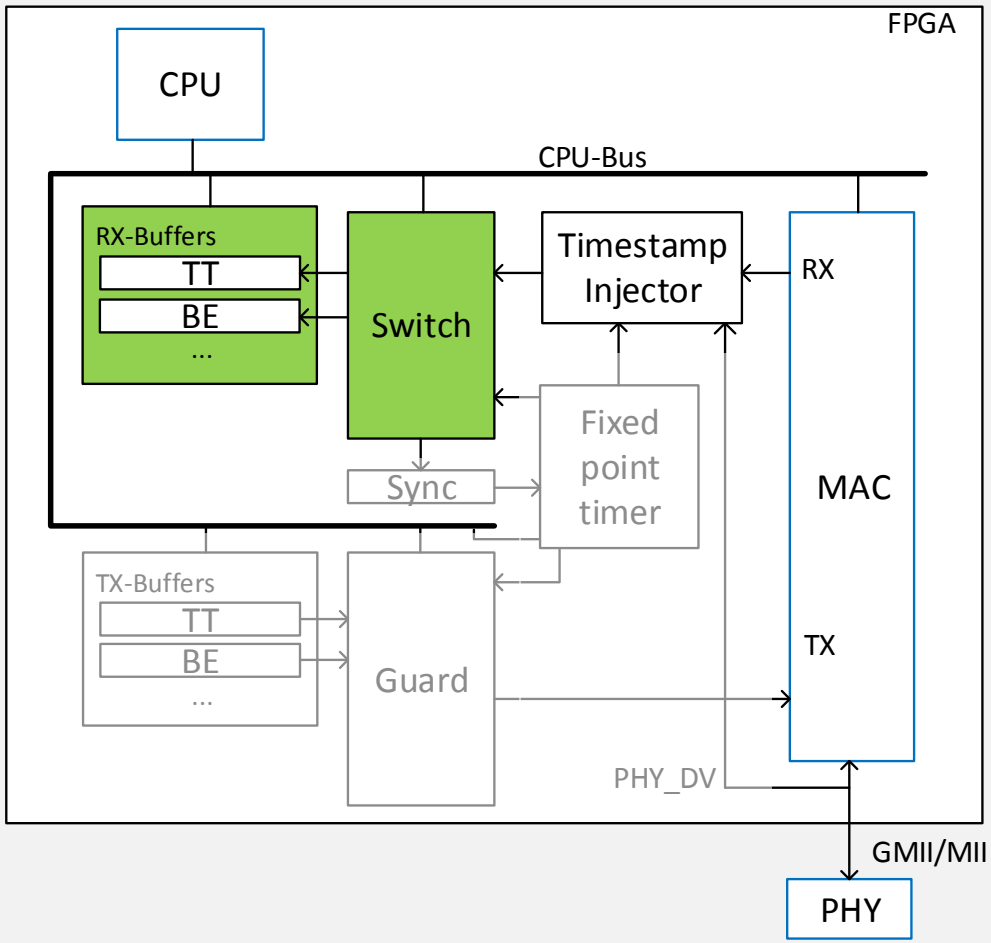

FPGA
CPU
CPU-Bus
RX-Buffers
TT
BE
...
Switch
Timestamp Injector
RX
Fixed point timer
Sync
MAC
TX-Buffers
TT
BE
...
Guard
TX
PHY_DV
GMII/MII
PHY

Concept
Architecture - Timestamp Injector



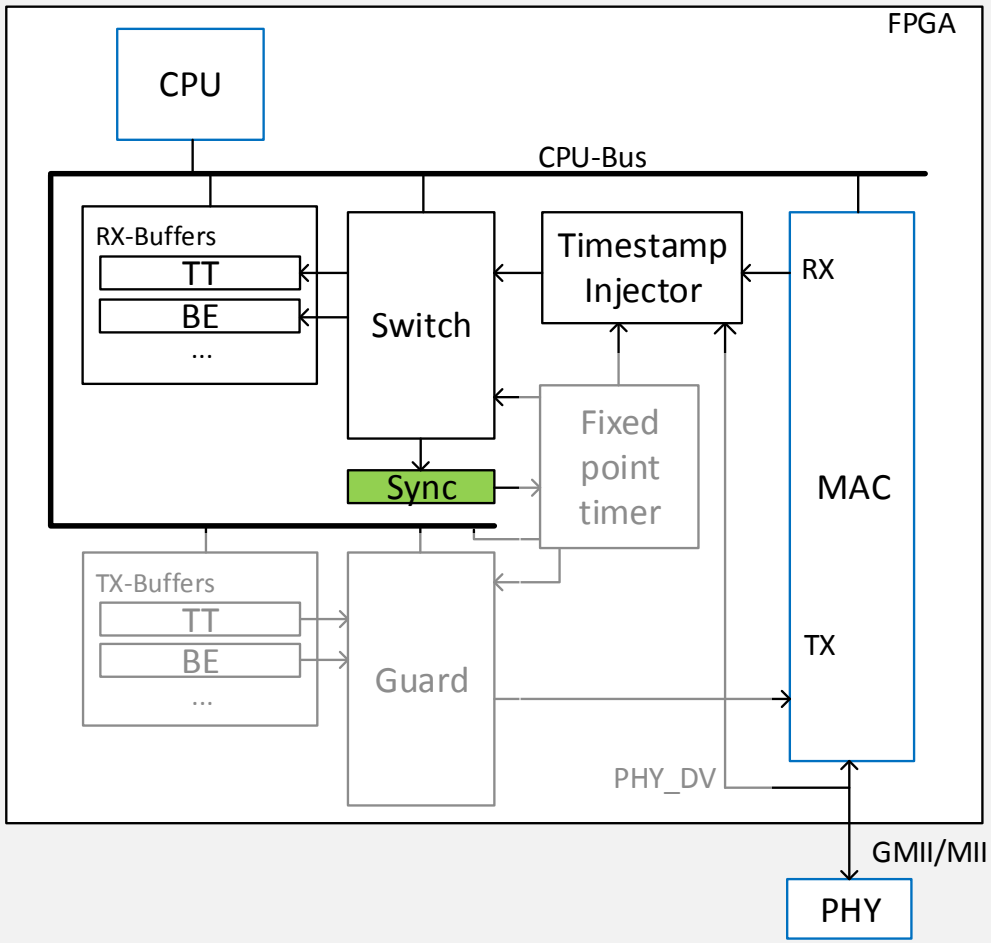
- Timestamps needed for Synchronisation and validation of TT Frames
- SW-Implementation has low accuracy
- Record Timestamps with PHY_DV (10 ns jitter)
- Validate with rx interrupt
- FIFO for TS
- Works on the fly
- Delay of 2 clock cycles

109 LUTs (≙ 5%); 72 Flip Flops (≙ 3.8%) of all (+ HW FIFO)



- Need to separate Traffic to different Buffers
- SW-Implementation needs the most CPU-Resources
- Switches based on Dest-MAC, Ether-Type, Timestamp
- Independent interrupts
- Application specific buffer size
- Works on the fly
- Delay of 6 clock cycles

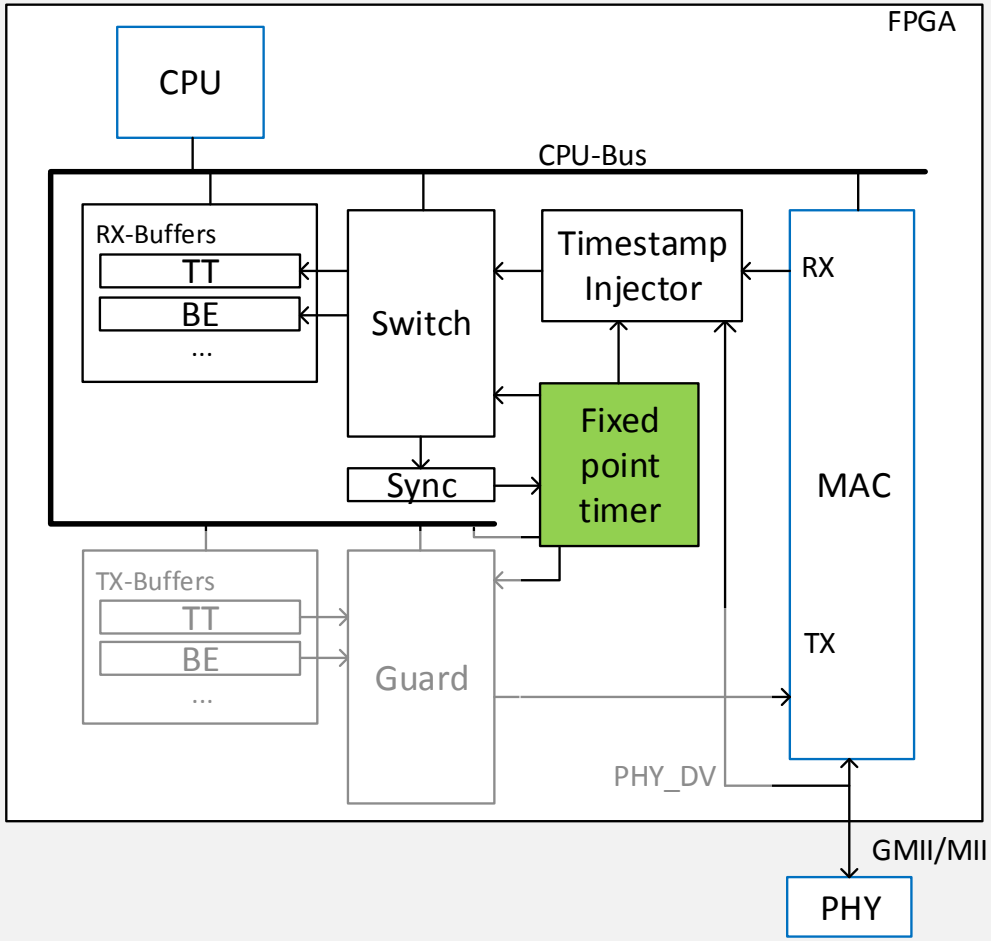
307 LUTs (≙ 14.1%); 529 Flip Flops (≙ 11.7%) of all



- Synchronize internal clock to network clock
- SW-Implementation need more energy
- On full HW-Implementation OS modification is very low
- Full AS6802 client implementation
- Rate-correction

1100 LUTs (≙ 50.3%); 736 Flip Flops (≙ 21%) of all

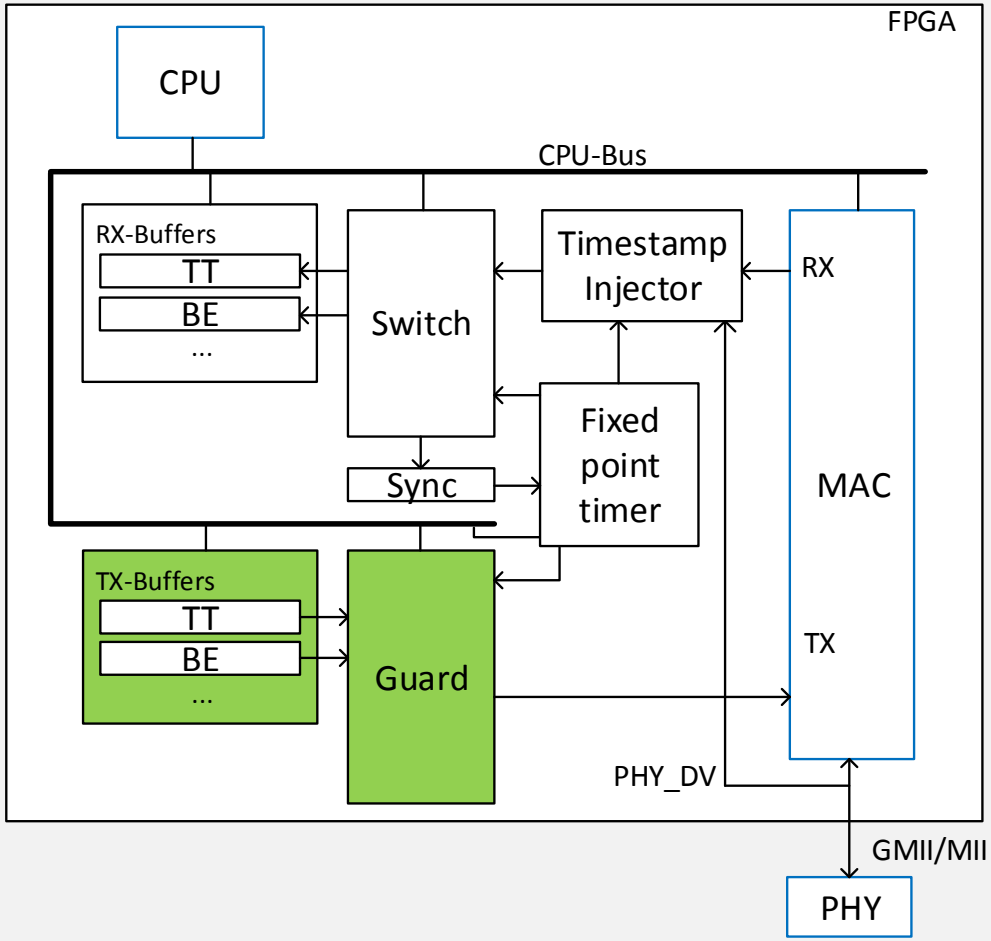
Concept
Architecture - Fixed point timer



- Addon for Synchronisation
- Keeps clock synchronized during the whole cycle
- Smaller reservation window
  -> more bandwidth
- SW-Implementation impossible
- Rate-correctable timer implemented as Fixed-Point timer

55 LUTs (≙ 2.5%); 55 Flip Flops (≙ 1.2%) of all

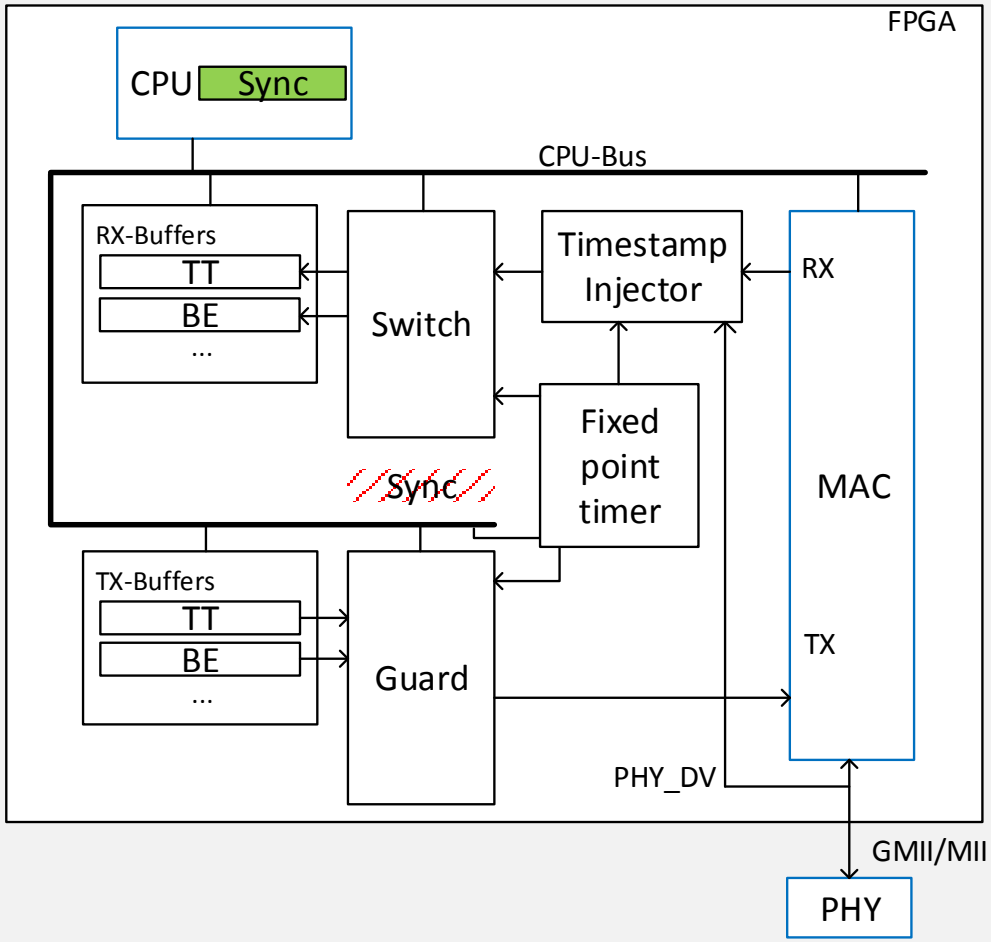
Concept
Architecture - TX-Buffers and Guard



- Sends TT on schedule and BE messages between TT messages
- CPU can put async messages to Buffers
- OS don't need Time-Triggered schedule functions
- SW-Implementation has high Jitter 1µs to 10µs
- HW-Implementation has 80ns Jitter
- Application specific buffer size

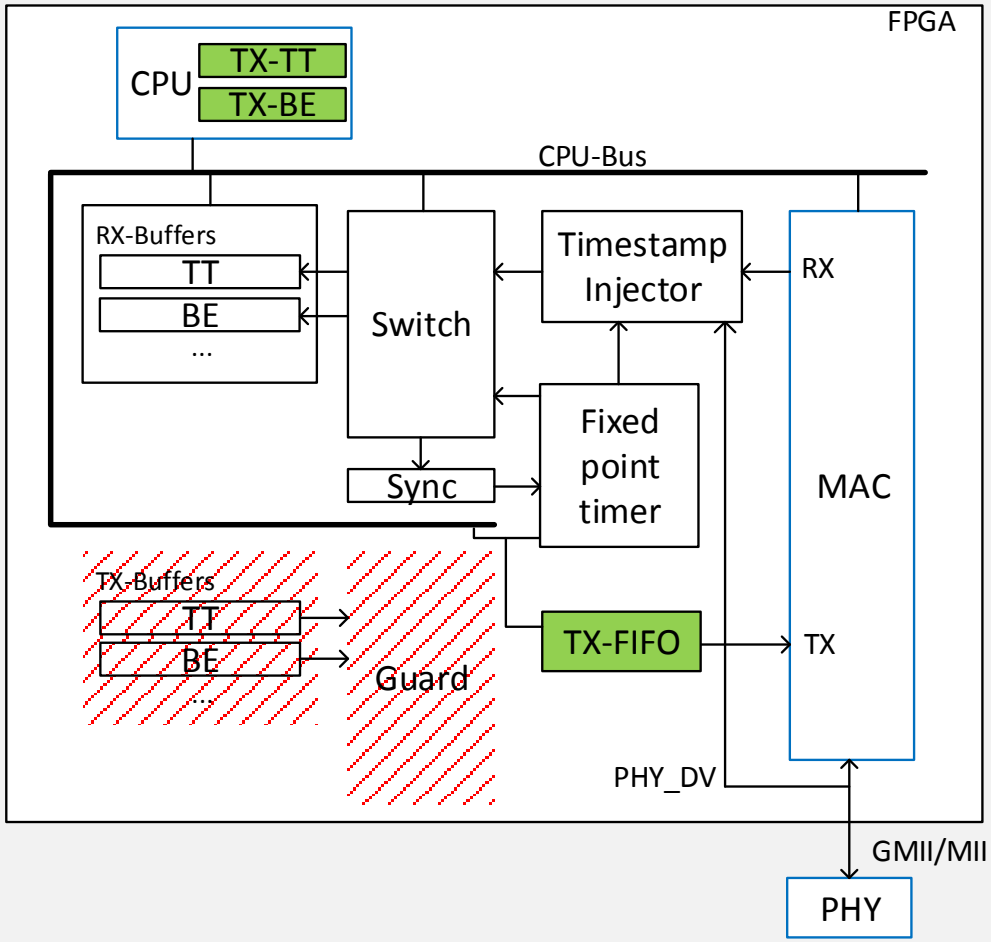
614 LUTs (≙ 28.1%); 475 Flip Flops (≙ 5.4%) of all

Concept
Possible Partitioning



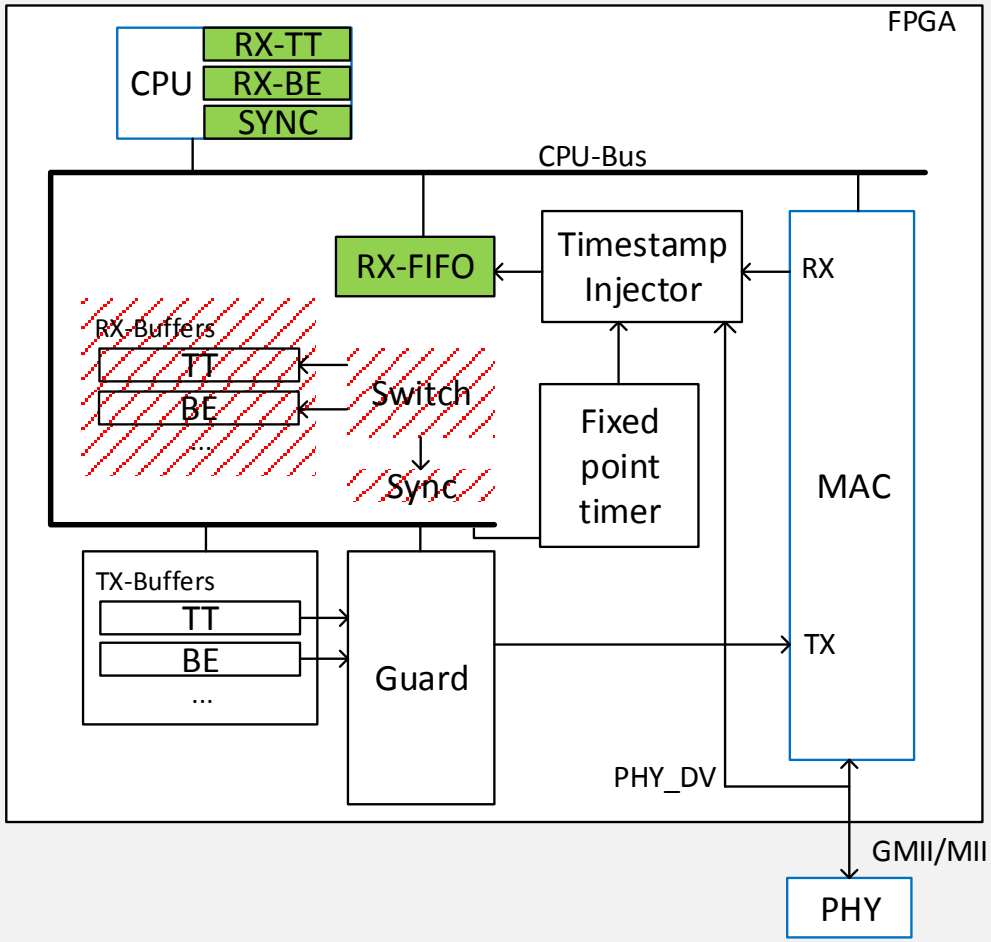
- CPU-Res.: few percent
- HW-Res.: safe 21% FF's and 50.3% LUT's
- Accuracy: no effect
- Energy: much higher

Concept
Possible Partitioning



- CPU-Res.: few percent
- HW-Res.: safes 1% FIFO's and 18.6% LUT's
- Accuracy: Jitter rises up to 10µs
- Energy: still is no statement possible

Concept
HW/SW Co-Design



- CPU-Res.: up to 90%
- HW-Res.: safes 31% FIFO's and 55% LUT's
- Accuracy: no effect
- Energy: much higher

Agenda

1 Introduction & Motivation

2 Concept & Results

3 Conclusion & Outlook

Conclusion

- Full Hardware implementation of a TTEthernet Controller
- Approach how to scale it
- Result: HW/SW Co-Design is a good way to deduce CPU consumption of a Time-Triggered Ethernet protocol stack.

Outlook

- AUTOSAR is a automotive system architecture without time-triggered scheduling mechanisms
- Develop driver for AUTOSAR
- Run different partitions of HW/SW Co-Desing

Thank you!

Thank you for your attention!

- Website of CoRE research group:
  http://www.haw-hamburg.de/core

SPONSORED BY THE

Federal Ministry of Education and Research