Hardware Assisted Clock Synchronization with the IEEE 1588-2008 Precision Time Protocol

Eleftherios Kyriakakis Jens Sparsø Martin Schoeberl

Department of Applied Mathematics and Computer Science, Technical University of Denmark (DTU)





- 1. Introduction
- 2. Background
- 3. Design
- 4. Evaluation
- 5. Conclusion

Introduction

Introduction



Automotive and industrial automation networks require:

- Time-predictable and bounded execution
- $\cdot\,$ Time-sensitive and deterministic communication
- Mixed criticality network traffic

Time-Sensitive Networking (TSN) is identified as the standard for network communication in Fog Computing and Industrial Internet of Things.

• Relies on a **global time reference** provided by the IEEE 1588-2008 Precise Time Protocol

Commonly, PTP is **implemented** in:

- Software
- Compatible PHY transceivers

Motivation



We investigate and propose a PTP hardware-assist unit:

- MAC-layer based
- Nanosecond synchronization
- WCET analyzable software
- Implemented in FPGA

Why?

- + Not investigated extensively.
- + Industrial platforms will **include** FPGAs (i.e. Intel's Fog reference design)
- + Comparable accuracy to PHY-based
- + Increased price of PTP-capable PHY transceivers **compared** to low-range FPGAs
- + FPGA resources can be **modified** as well as **shared** with other hardware units (i.e. hardware accelerators)

Background

Network Clock Synchronization

- Scheduling operations and collecting measurements across the network
- Calculate the **time difference** between network devices
- Maintain a minimal **offset** from the global time reference
- Network Time Protocol:
 - Client-server based **polling** protocol
 - Application layer protocol over UDP
 - Propagation delays are not accounted
 - Best-case millisecond accuracy



Control system



source (i.e. GPS

Data-acquisition system

Network time example



C

Application

IEEE 1588-2008 PTP (1/3)

- Operates on local area networks
- Systems that require **nanosecond** accuracy
- Based on a **master-slave** hierarchy
 - \cdot A grand-master is equipped with a high-precision clock (i.e. GPS)
- Messages are exchanged over UDP or raw Ethernet frames
- Accounts for propagation delay through devices
- Allows for **sub-microsecond** clock synchronization
- Each Ethernet port of a compatible device implements the following fundamental **blocks**:
 - IEEE 1588-2008 clock
 - Frame/Packet recognizer
 - Timestamp capturing
 - Clock adjustment

6

IEEE 1588-2008 PTP (2/2)

- Exchange of four messages:
 - 1. SYNC
 - 2. FOLLOW_UP
 - 3. DELAY_REQUEST
 - 4. DELAY_REPLY
- **Collecting** four timestamps: *t*₁, *t*₂, *t*₃, *t*₄.
- Offset is **calculated** as:

 $offset = t_2 - t_1 - delay$

where:

$$delay = \frac{(t_2 - t_1 + t_4 - t_3)}{2}$$





Related Work (1/2)



\cdot Timestamping

- Software-based
 - No requirement for hardware support
 - Software induced delays cause **jitter**
 - Achieves microsecond precision
- MAC-layer
 - Monitors the received frame nibbles from the MAC controller
 - Can achieve sub-microsecond precision
 - Implemented in modern commercial MCUs (i.e. STM32F107xx)
 - Has **not** been explored extensively
- PHY-layer
 - $\cdot\,$ As close to the wire as possible timestamping
 - Implemented in commercial Texas Instruments PHYTER
 - Has been characterized in various projects
 - Ensures nanosecond precision

Related Work (2/2)



· Clock adjustment

- Not specified by the standard
- Can be implemented in different layers
- Three common methods:
 - 1. Directly **setting** the time
 - 2. Clock rate adjustment by pulse addition and swallowing
 - 3. No active correction but keep an error register instead

State-of-art

- White-Rabbit application (CERN)
- \cdot PTP on a custom network
- Fiber-optic links
- Synchronous Ethernet
- Achived sub-nanosecond precision

Experimental Platform T-CREST



- Multi-core **research** platform
- Time-predictable VLIW Patmos processor
- Argo TDM network-on-chip
- \cdot WCET optimized toolchain
 - Custom LLVM-based compiler
 - WCET analysis tool platin
- Research use cases:
 - Time-predictable computing
 - Network-on-chip
 - Real-time systems



T-CREST architecture overview

Design



- Integrated within the T-CREST platform as a single IP core
- The unit is composed of **three** functional entities:
 - 1. The two RX/TX timestamp units
 - 2. The IEEE 1588-2008 Clock
 - 3. The PTP software stack



Implementation of PTP Hardware-Assist unit inside a T-CREST node



- Provides MAC-layer hardware timestamping
- Ensures **as early as possible** timestamping (with std. PHY)
- Offloads PTP frame recognition parsing to hardware



Implementation of the proposed timestamp unit

Clock Adjustment

- Composed of **four** parts:
 - 1. The clock counter
 - 2. The abrupt update register
 - 3. The offset correction register
 - 4. The Rate LUT
- A LUT selects a clock time-step increment
- Offset is reduced gradually
- Configurable rate adjustment through LUT



Implementation of the proposed clock adjustment unit



Software stack



- The PTP software stack is responsible for the following **tasks**:
 - Initializing Patmos in master or slave PTP port mode.
 - Executing the clock synchronization protocol.
 - Controlling the PTP hardware assist unit.
 - **Reporting** the clock offset at each synchronization interval.
- Both the PTP_MASTER and the PTP_SLAVE **share** the same codebase.
- The PTP_MASTER and the PTP_SLAVE roles are **explicitly** defined.
- WCET analyzable code:
 - Static allocation
 - Zero-copy
 - Non-blocking
 - Bounded loops

Evaluation

Experimental Setup





Picture of the evaluation setup. Seven segment displays the current time in seconds in hexadecimal



FPGA resource utilization (Altera Cyclone IV FPGA - 114480 Total Logic Elements)

Entity	Combinational LUTs	Registers
PTP Hardware-Assist	1485	1182
MIITimestampUnit	454	402
DeserializePHYbyte	13	11
DeserializePHYBuffer	65	64
RTC	431	234



WCET Analysis of PTP Software Stack

Function	WCET	
	Clock Cycles	Time (at 80 MHz)
ptpv2_issue_msg()	2560141	32 ms
<pre>check_ptpv2_frame()</pre>	684	8.55 us
ptpv2_handle_msg()	3893	48.6 us

Results - Timestamping method comparison





PTP-Slave clock offset comparison between software-based (top) and hardware-based (bottom) timestamping using only abrupt updates.

Results - Clock adjustment method comparison





PTP-Slave clock offset comparison between abrupt updates (top) and rate-control (bottom) adjustment.

Conclusion



- **Extend** the experimental setup to a larger TSN network of T-CREST nodes.
- Increase the resolution of the clock and evaluate the results using different SYNC message rates.
- Investigate a complete **in-hardware** PTP synchronization solution aiming to:
 - Automated periodic PTP frame transmission
 - Automated clock offset calculation and correction
 - Eliminate CPU processing time for clock synchronization

Conclusion



- PTP hardware-assist implementation
 - PTP frame recognition timestamp unit
 - Clock rate adjustment
 - WCET analyzable software
 - Peer evaluated in-hardware
- Achieves **nanosecond** clock synchronization
 - Comparable jitter of 50 ns as the commercial TI PHYTER
 - Improved worst-case offset of 134 ns compared to the STM32F107xx micro-controller implementation of 260 ns
- Uses minimal FPGA resources
 - Only 1.7% of the total available resources of a medium-range FPGA device
 - And 11 % of the total size of the medium-sized Patmos processor
- The IP core is implemented but-**not-limited**-to FPGA