

PTP version 1 implementation on FPGA with NIOS processor and Gigabit MAC IP

R&D for T2K experiment

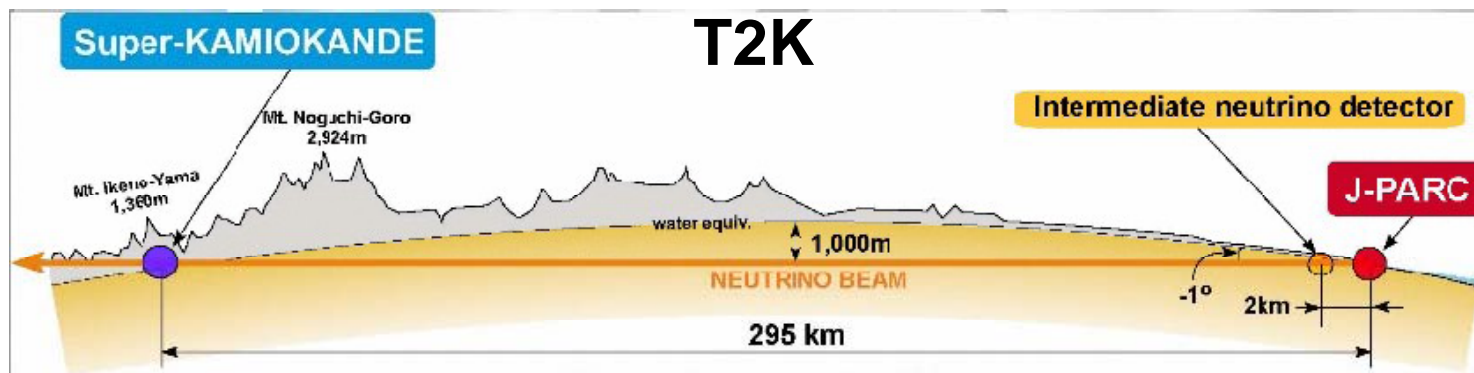
*Dario Autiero - Bruno Carlus – Jacques Marteau – Serge Gardien
Claude Girerd*

**INSTITUT de PHYSIQUE NUCLEAIRE DE LYON
CNRS / IN2P3 / UCBL
Villeurbanne - France**

Development context

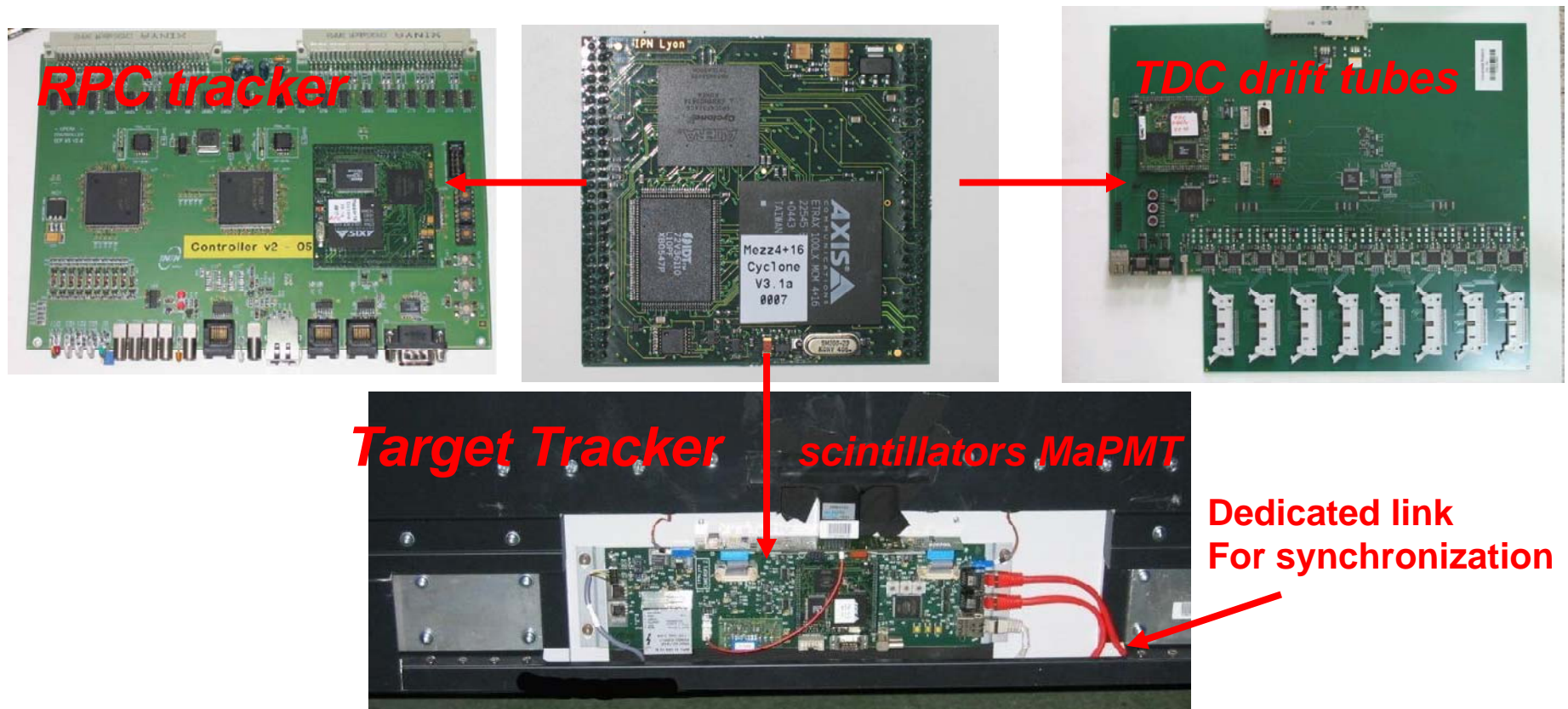
Data acquisition system for Neutrino experiments

- Concept of distributed « smart sensors » over Ethernet already applied successfully for the OPERA experiment
- Continuous and auto-triggerable readout
- Synchronization and event time stamp on each sensor



Network Distributed OPERA DAQ components

- A common mezzanine board with embedded Ethernet controller.
- A Specific interface board for each type of detector.
- A dedicated bus for synchronization



The OPERA experiment is now running on this concept with about 1200 sensors connected on an Ethernet Network

Development objectives

➤ The OPERA DAQ features

- Based on the concept of distributed « smart sensors » over Ethernet
- Embedded microprocesseur under Linux 10 / 100 Mbps Ethernet
- Independant and proprietary clock distribution system for synchronization

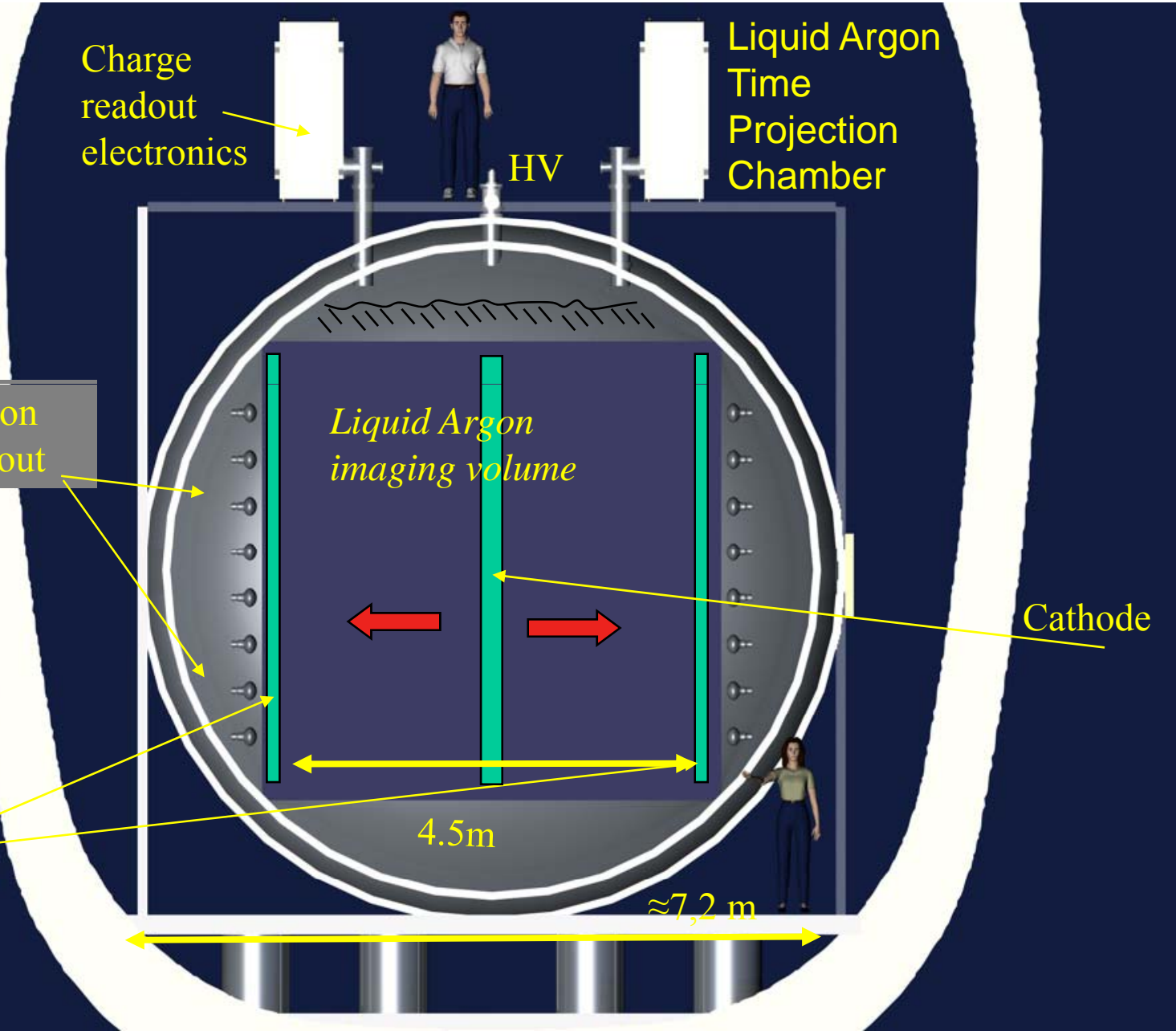
➤ Design improvements for next generation

- Go to the gigabit
- Hardware implementation of the network layers to free the local CPU
- Synchronized over the network (PTP) within an accuracy better than 10ns
- Industrial standard embedding network features (such as ATCA)

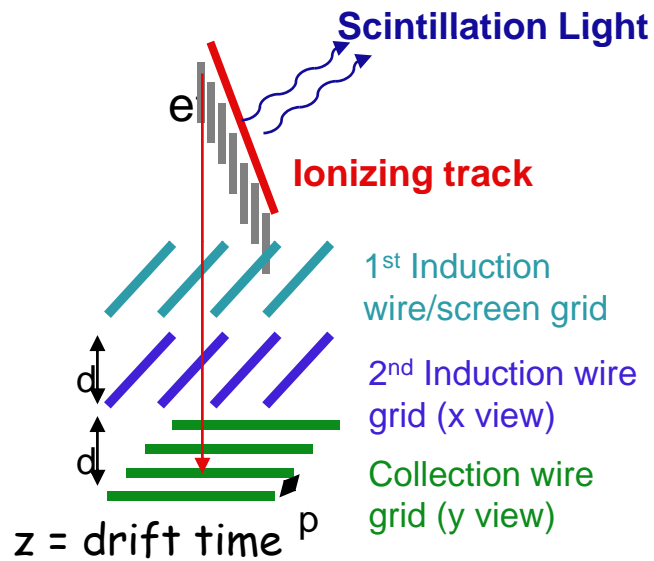
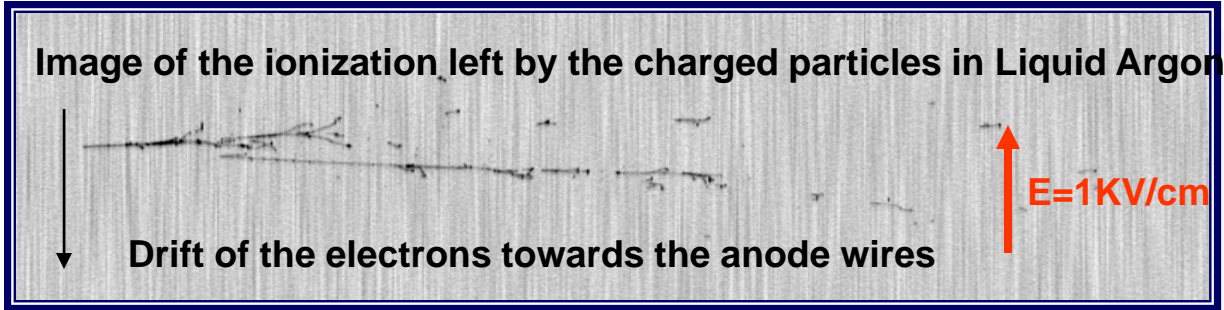
➤ Applied to the LAr TPC in T2K

PTP not mandatory for T2K but interesting for futur
“M-ton” large scale prototypes

Liquid Argon is at the same time the target and the sensitive medium for secondary particles produced in neutrino interactions

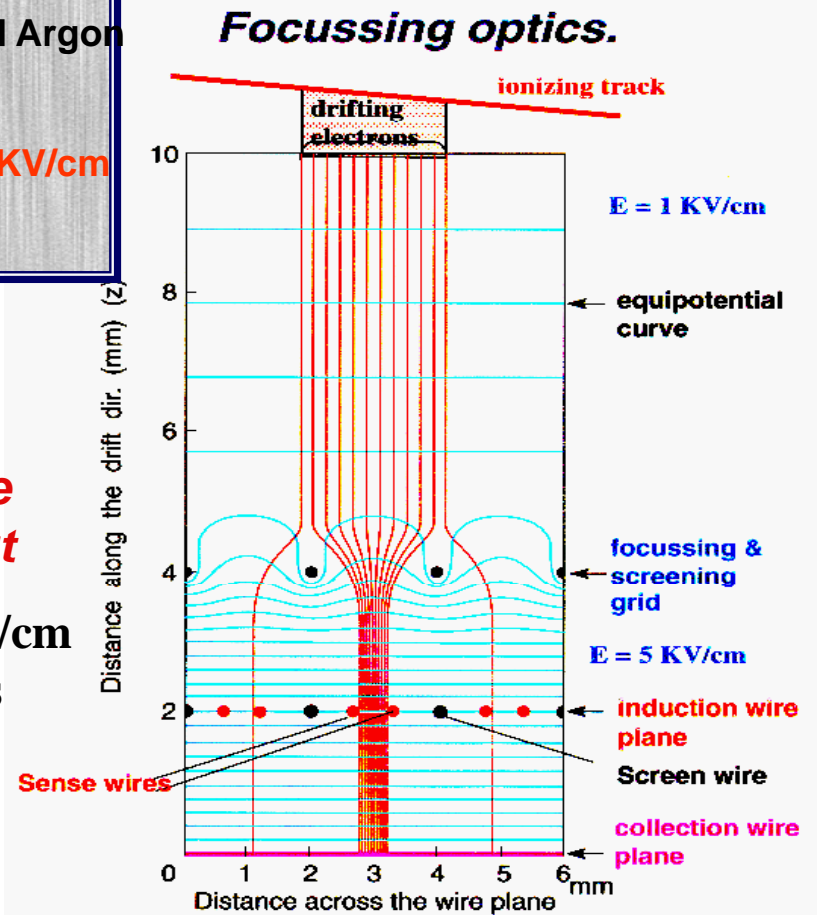


Front view



Non-destructive multiple readout

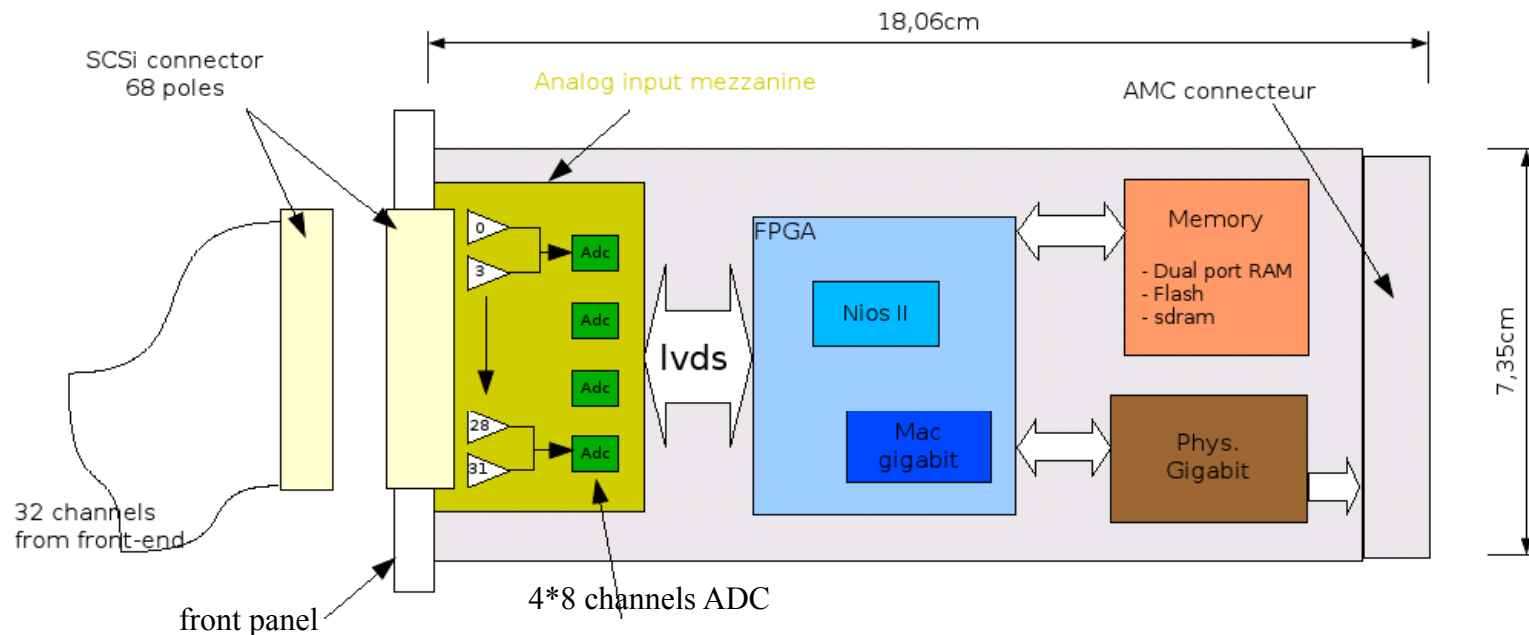
Drift Field: 1 kV/cm
Drift time ~ 3 ms



- Detection of primary ionization in LArgon: 1 m.i.p ~ 20000 electrons on 3 mm
- High resolution calorimetric measurement of e.m. and hadronic showers
- PMs detecting UV scintillation light in Argon used to provide the $t=0$ signal of the event to measure the drift space
- Each wire is sampled with a 2.5 MHz flash ADC to measure the charge distribution along the drift (time)
 - 3D event reconstruction with ~1 mm space resolution

Development objectives

- Designing a 32 channels ADC AMC board
- Using PTP for synchronization and time stamping
- Using gigabit Ethernet for data transmission

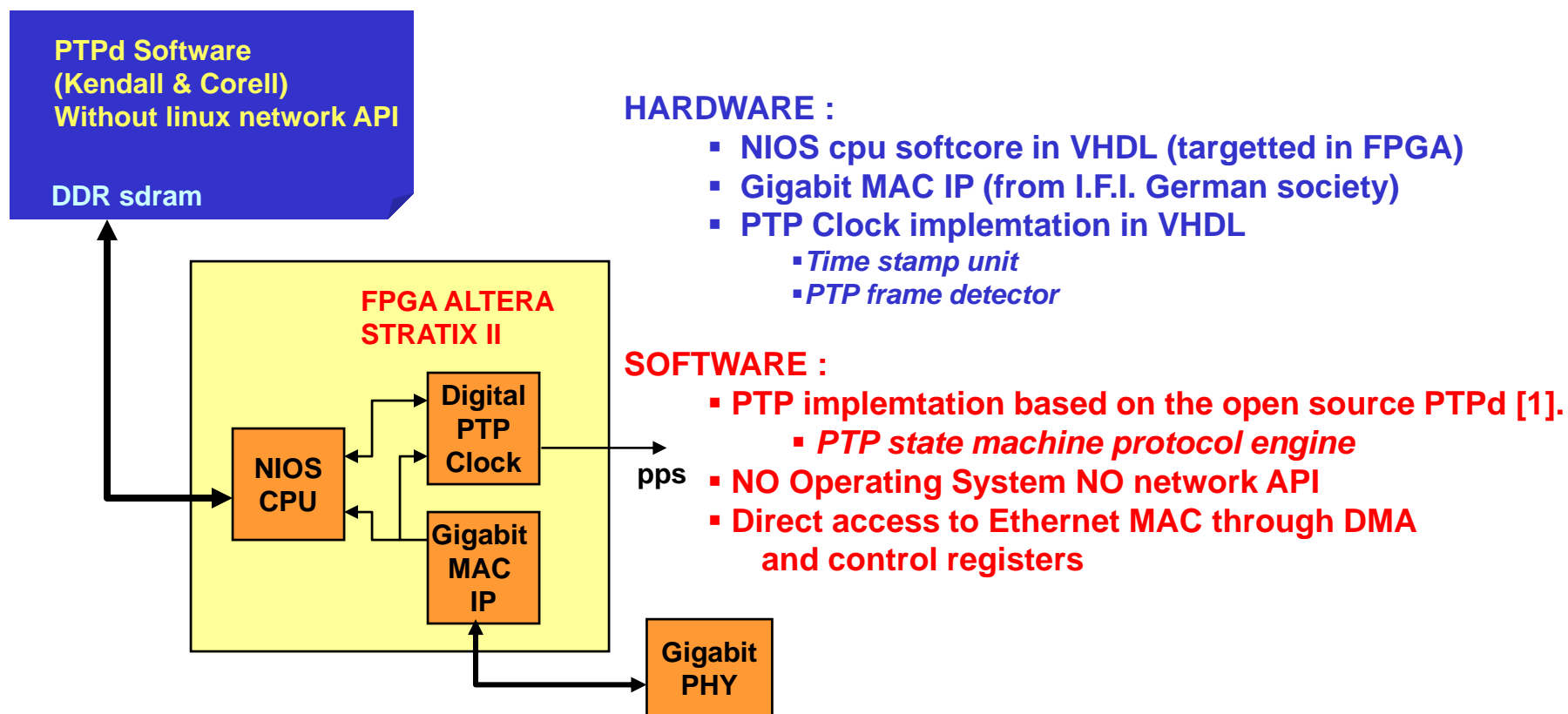


- AMC boards are integrated in a micro-TCA « shelf » :



PTP development overview

- Mixt software / hardware PTP implementation

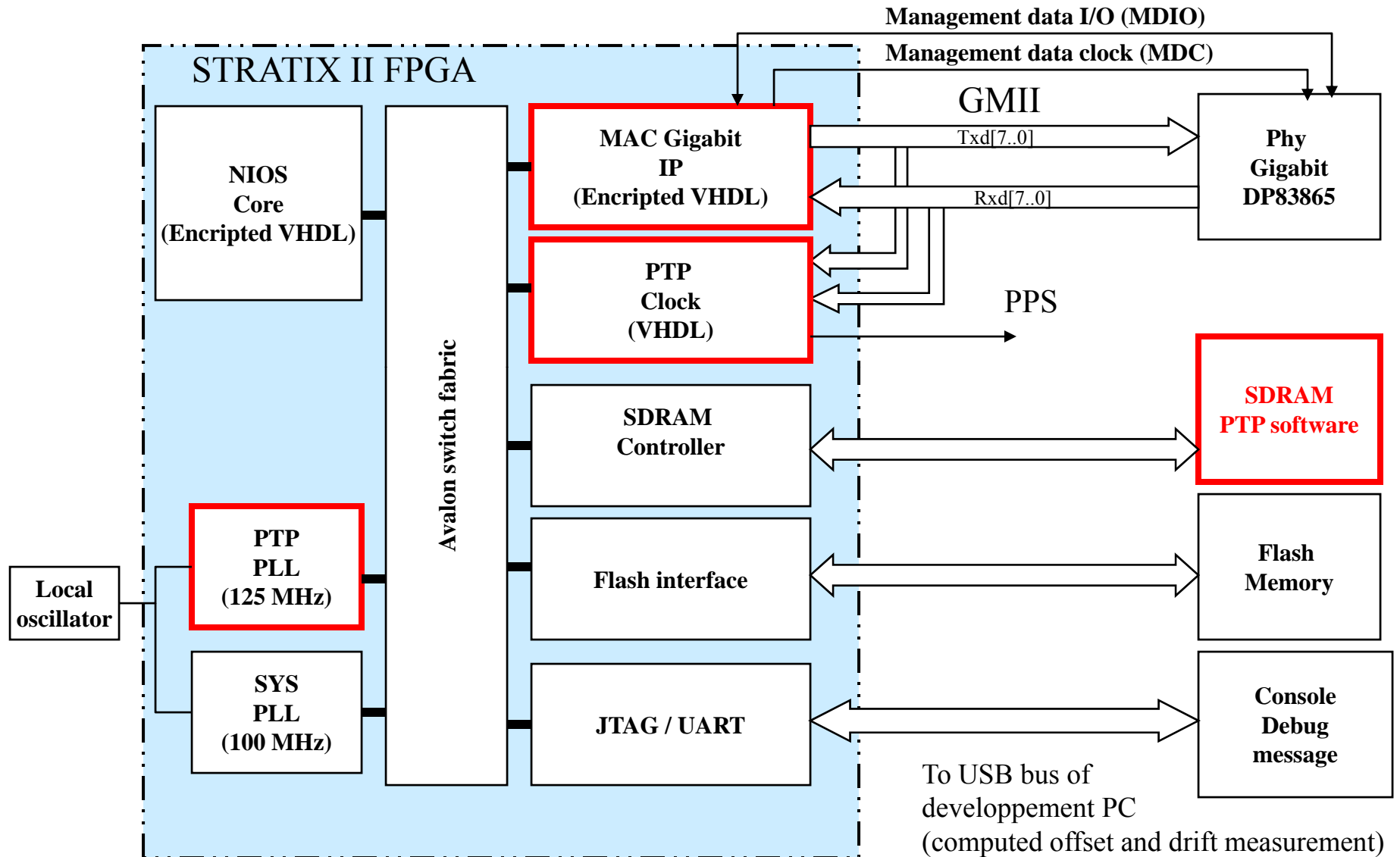


-Based on this configuration we developped the PTP SLAVE and MASTER.

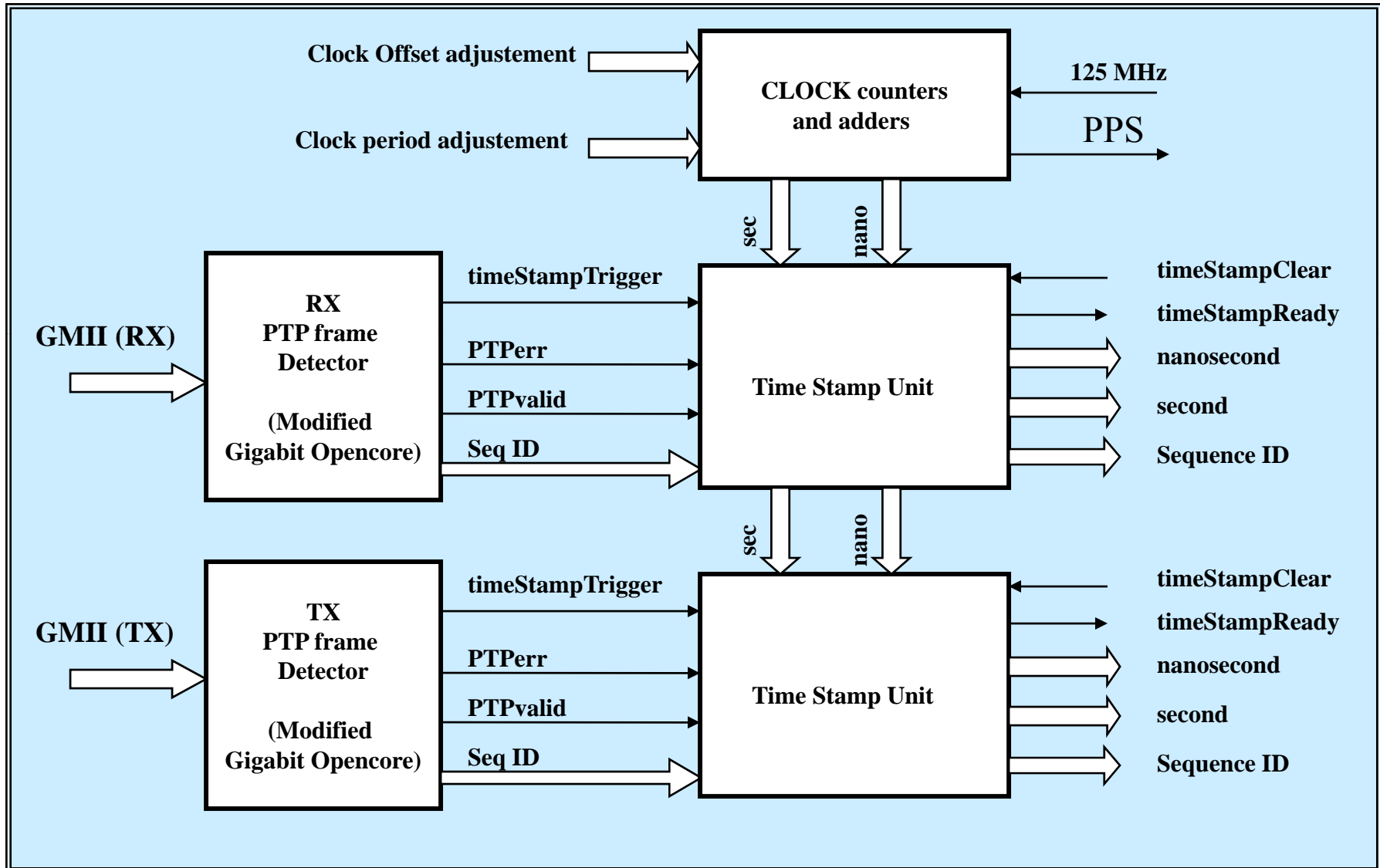
Brief description of the NIOS concept

- The NIOS processor is a soft-core 32 bit RISC Microprocessor
 - It is described in VHDL
 - Can be targetted in ALTERA FPGAs
- It is associated with:
 - an avalon switch fabric allowing data exchange
(a kind of configurable system bus for communication between peripheral and microprocessor)
 - A set of standard and custom peripherals
(You can create your own peripheral)
- SOPC Builder software provided by ALTERA:
 - This is a user friendly interface allowing the system description
 - Used to define the whole system:
NIOS cpu + peripheral + interconnection

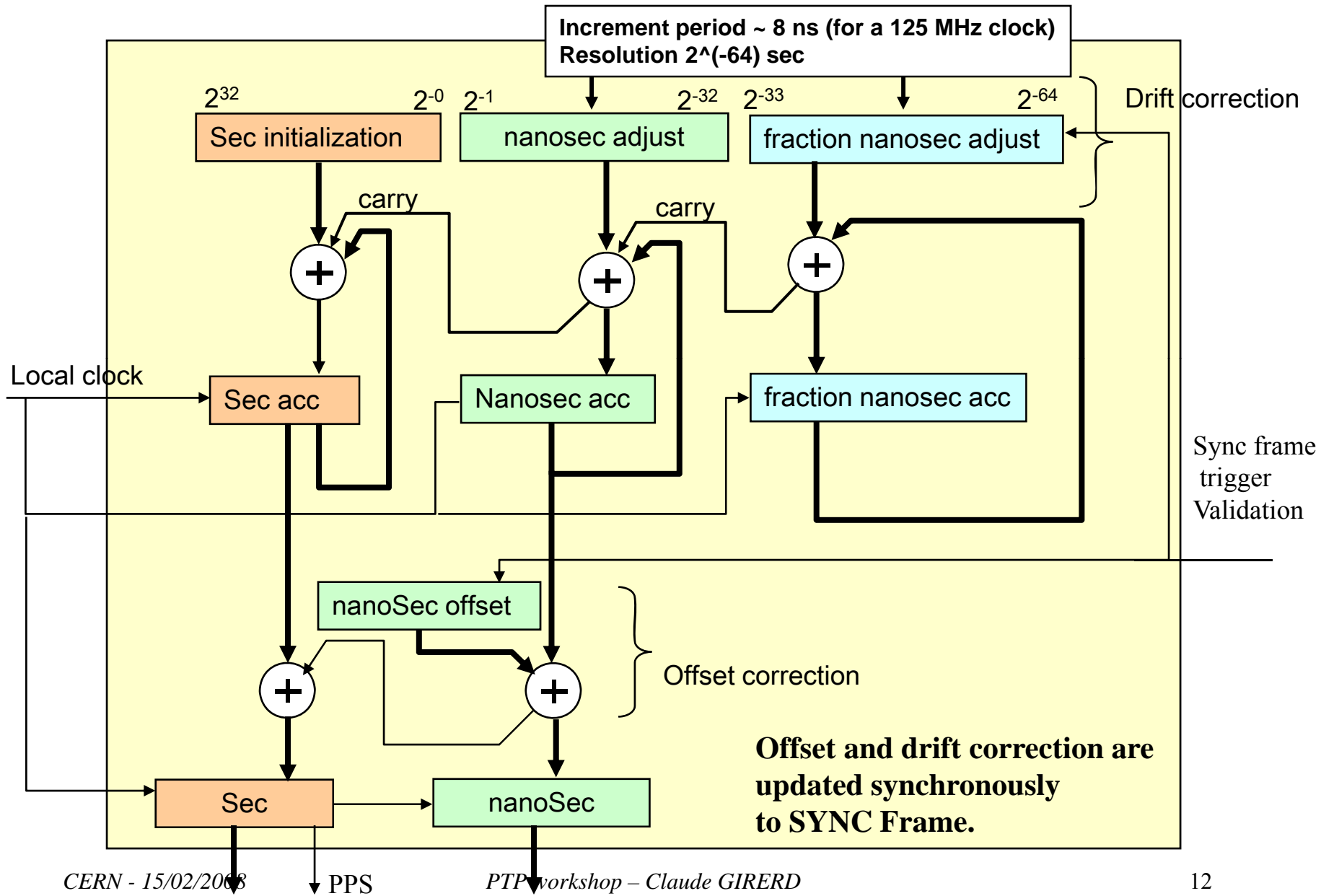
PTP hardware implementation with NIOS



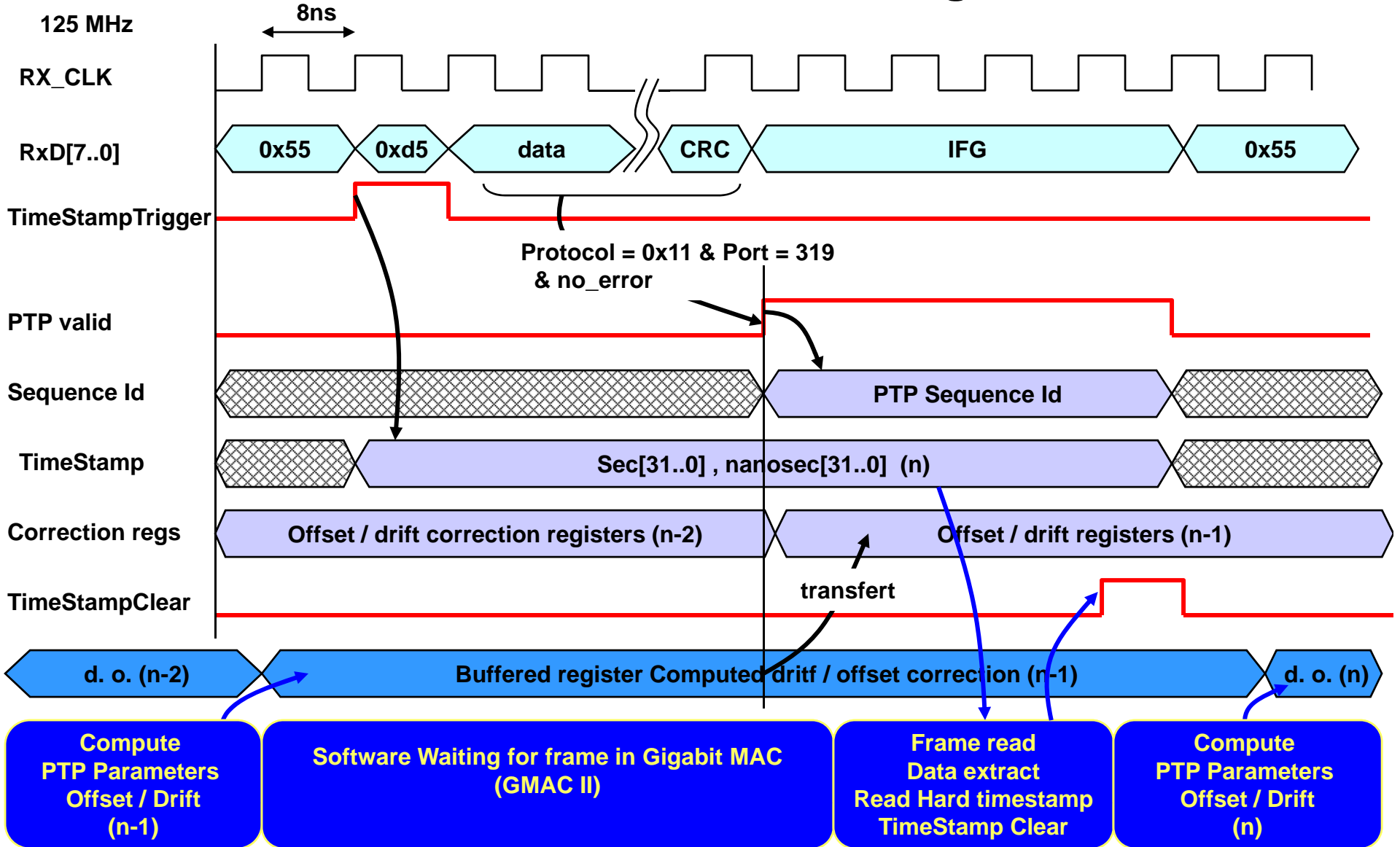
Hardware PTP clock details



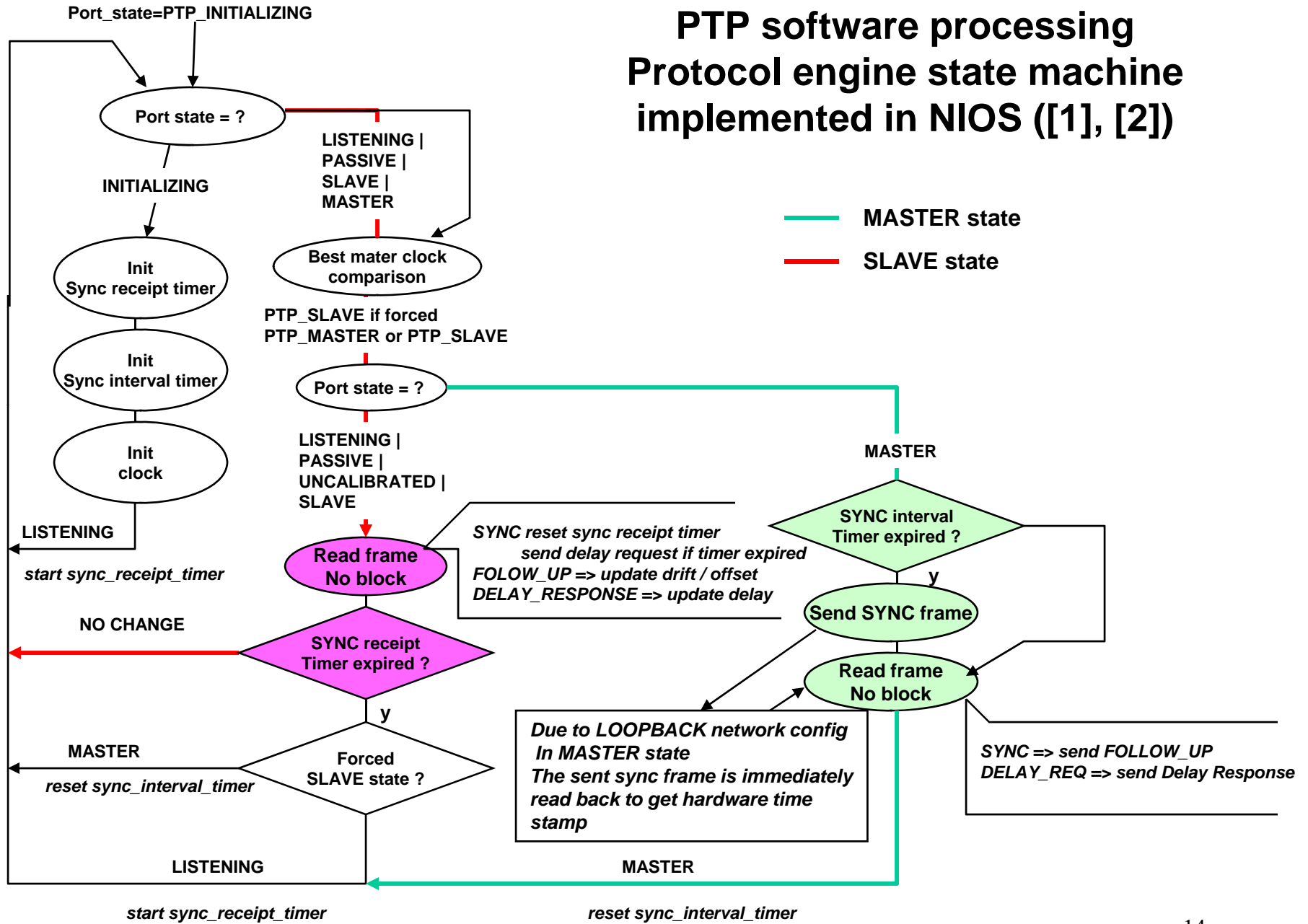
PTP clock counter module



PTP clock module timing



PTP software processing Protocol engine state machine implemented in NIOS ([1], [2])



MASTER

SLAVE

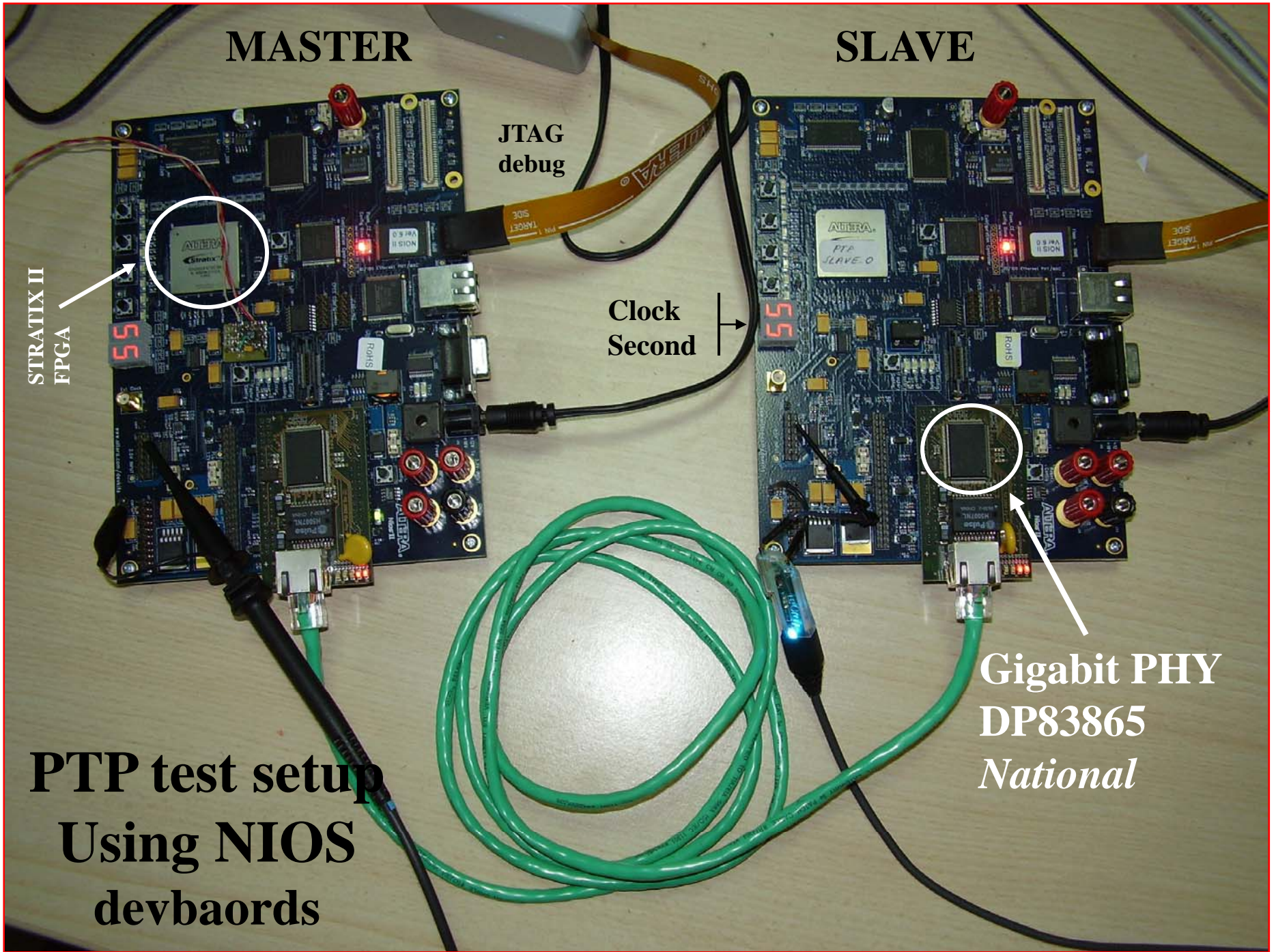
JTAG
debug

STRATIX II
FPGA

Clock
Second

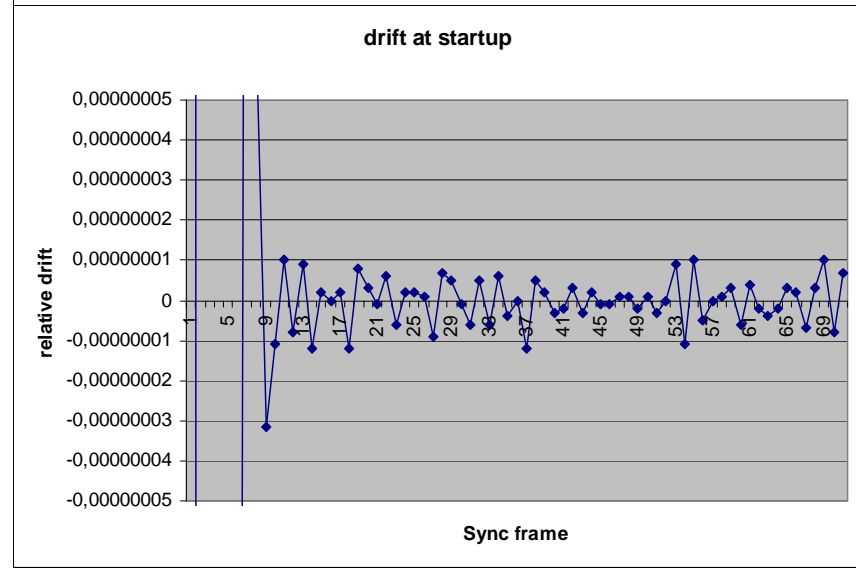
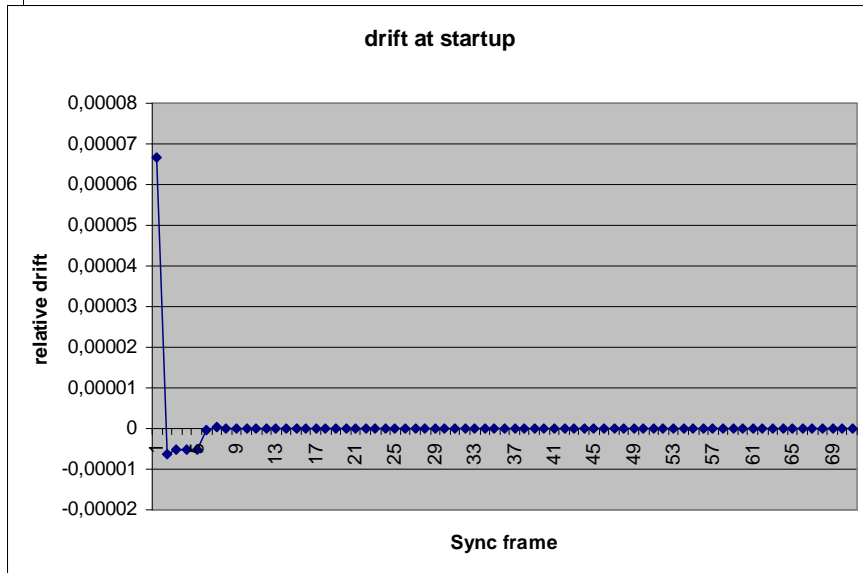
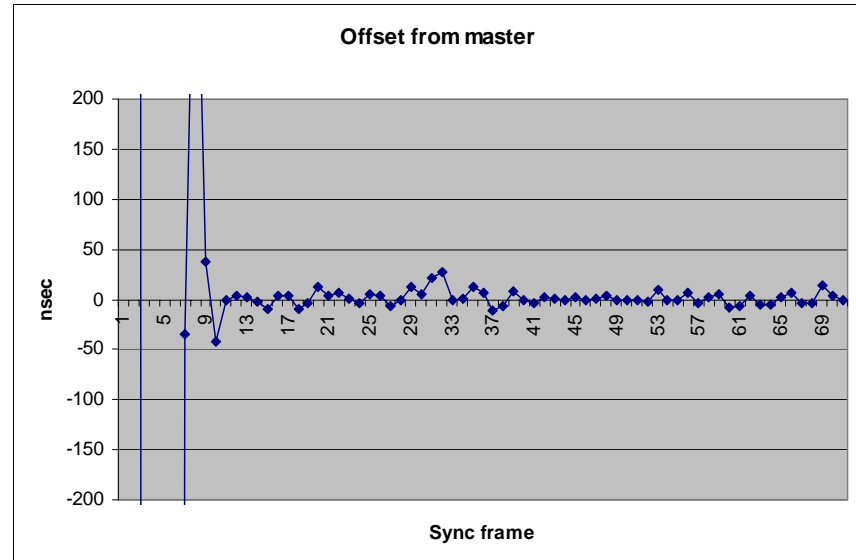
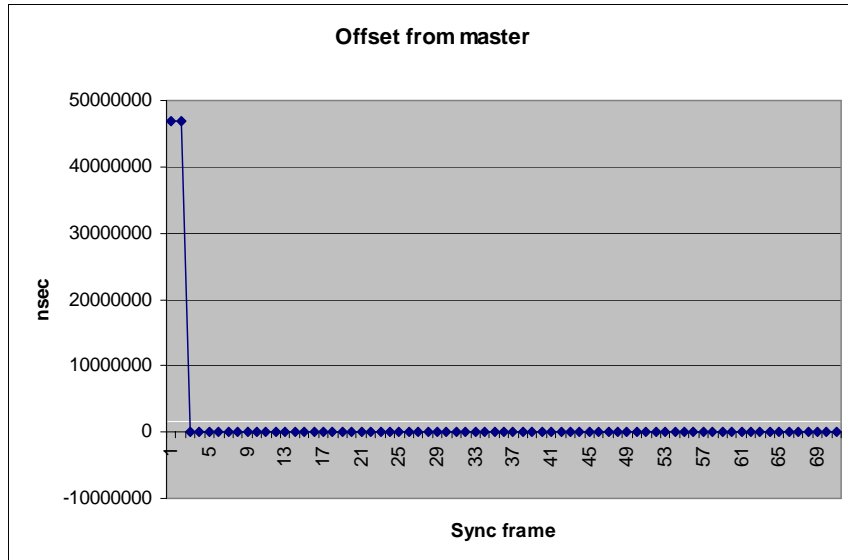
Gigabit PHY
DP83865
National

**PTP test setup
Using NIOS
devbaords**



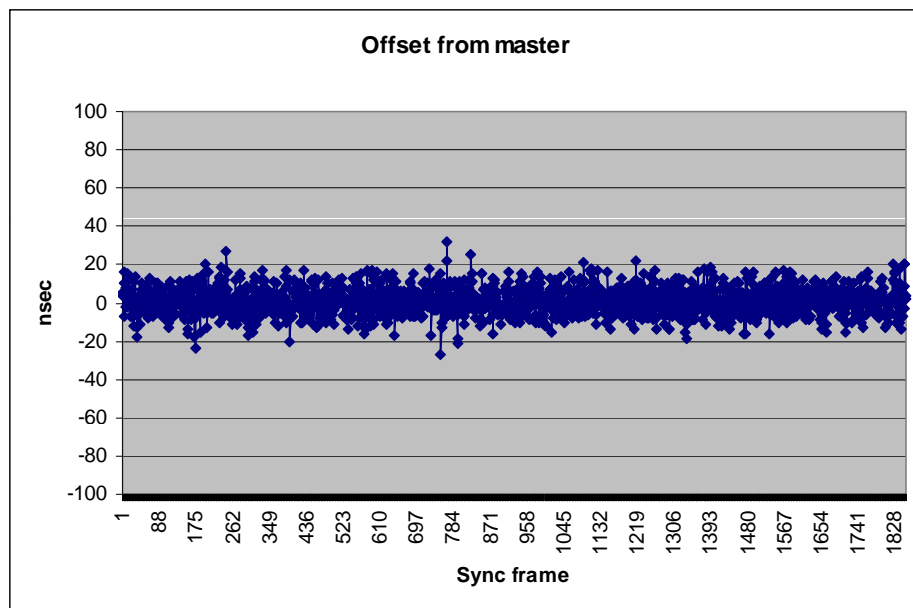
Clock recovery and local oscillator on slave (1)

Computed PTP parameters with PTP timestamps

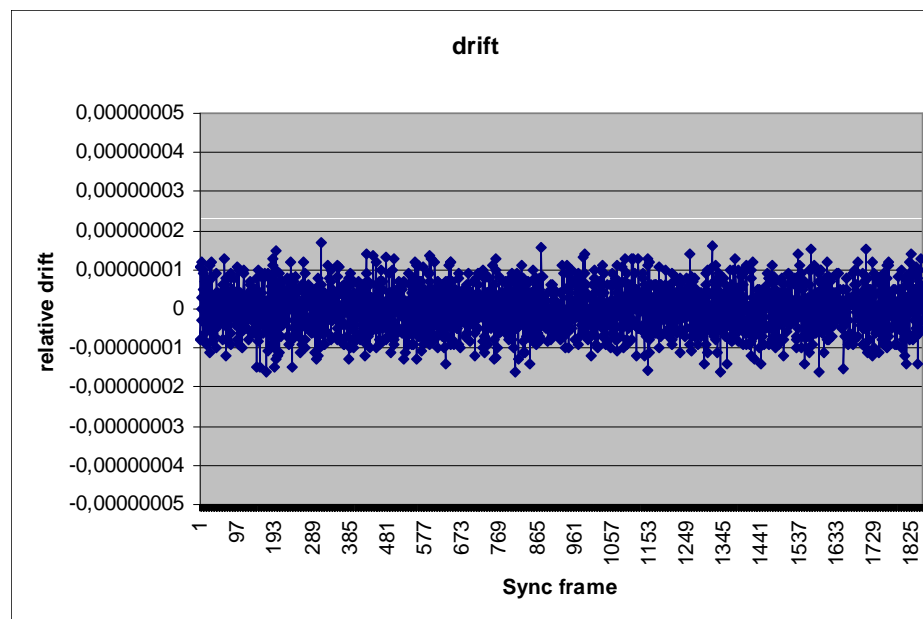


Clock recovery and local oscillator on slave (2)

Computed PTP parameters with timestamps

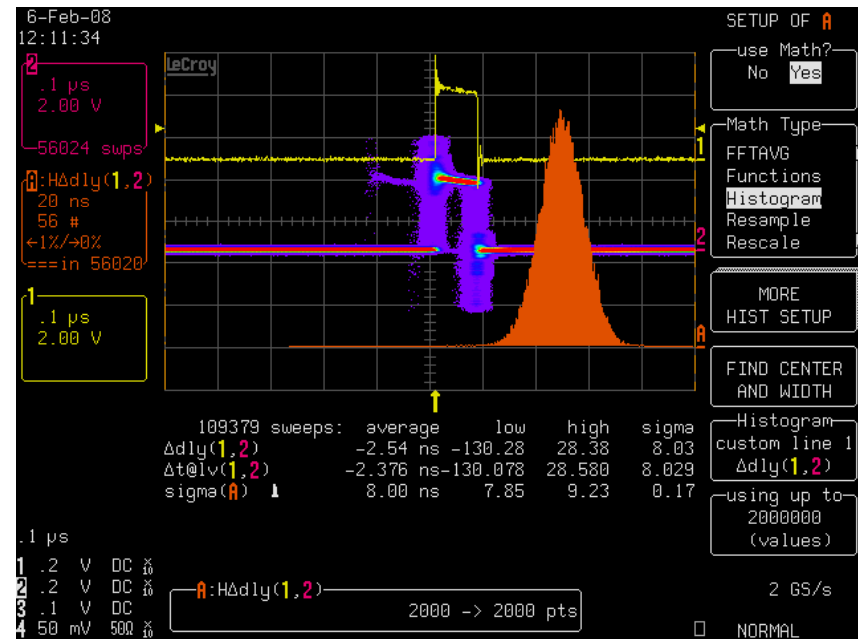
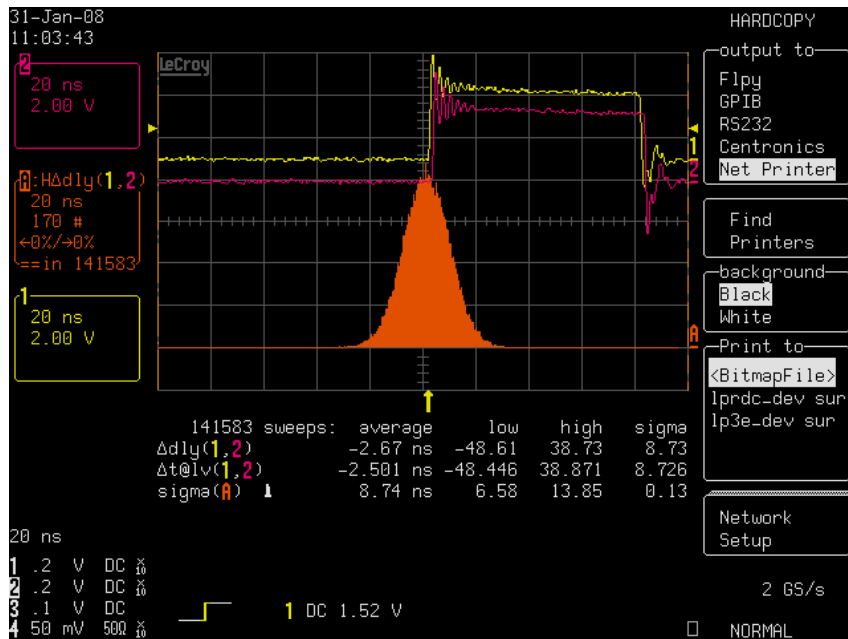


Offset pp \sim +/- 20 ns
(point to point no switch)



Drift pp \sim +/- 10 ppb
(point to point no switch)

Clock recovery and local oscillator on slave (3) hardware measurement



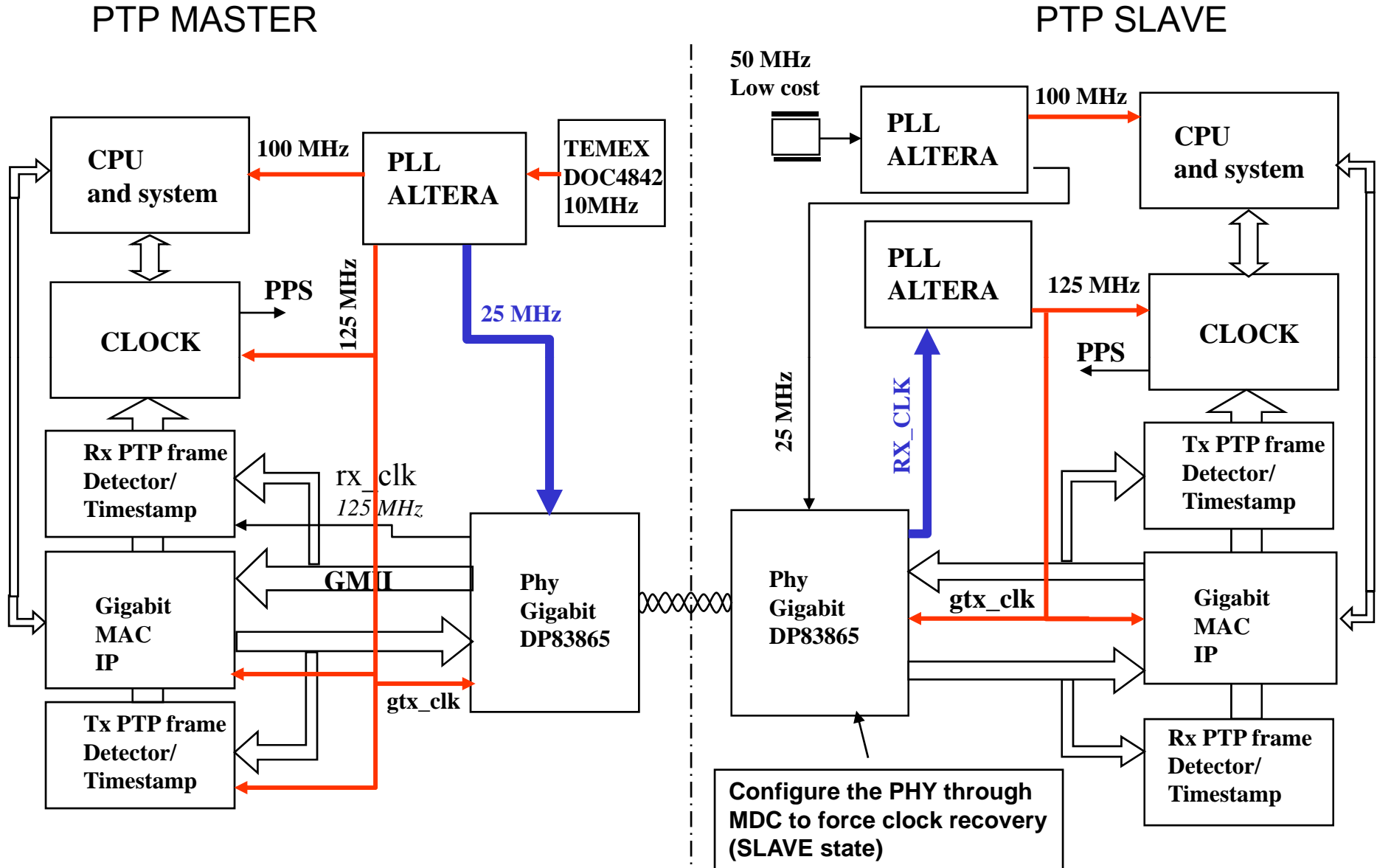
**PPS synchronization
8.74 ns rms
(point to point without network traffic)**

Other possibility

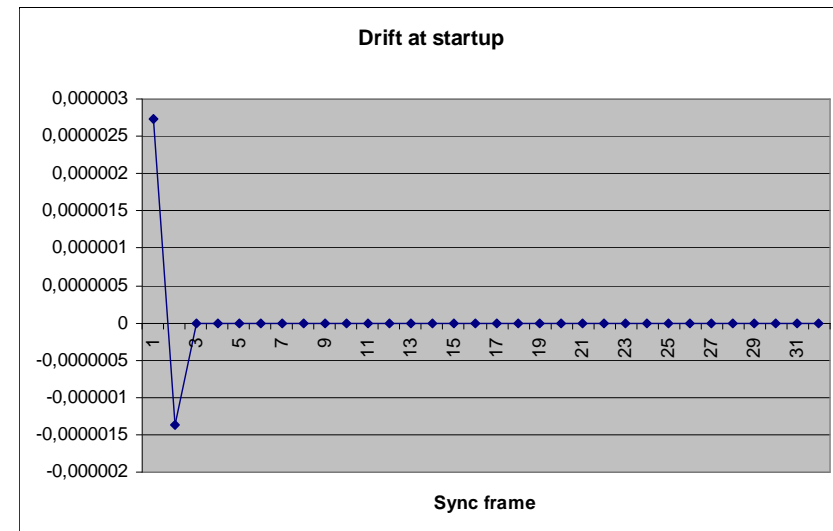
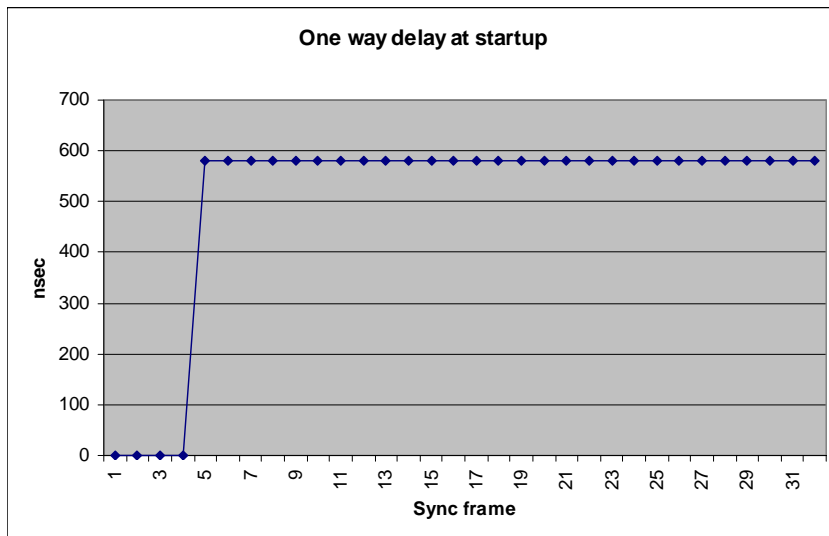
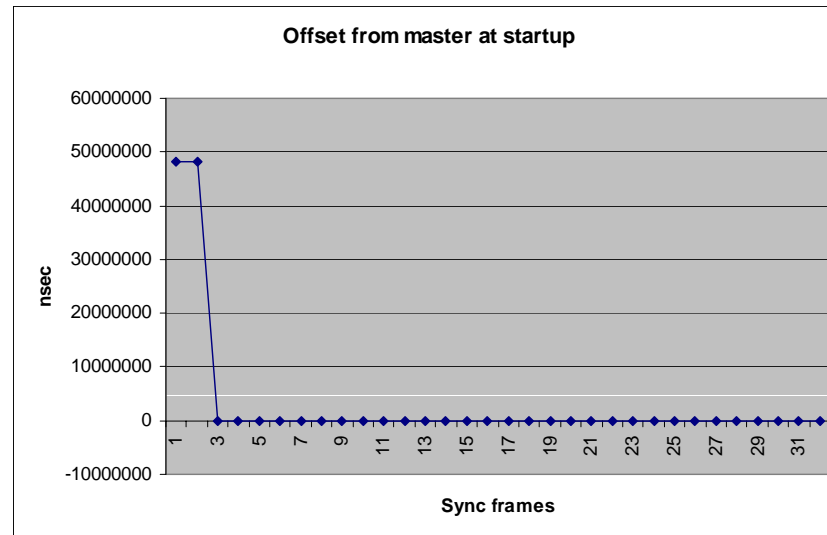
Using RX_CLK as local clock on slave

- It is possible to force a gigabit PHY to be in SLAVE state (*not at the PTP but for the ethernet link point of vue*)
- In the SLAVE state a gigabit PHY recovers the RX_CLK from the frame and remains synchronous over time.
- The idea is to use the RX_CLK clock in place of the local clock
- In this condition the drift is eliminated and the clock are synchronous between the two nodes in a point to point link

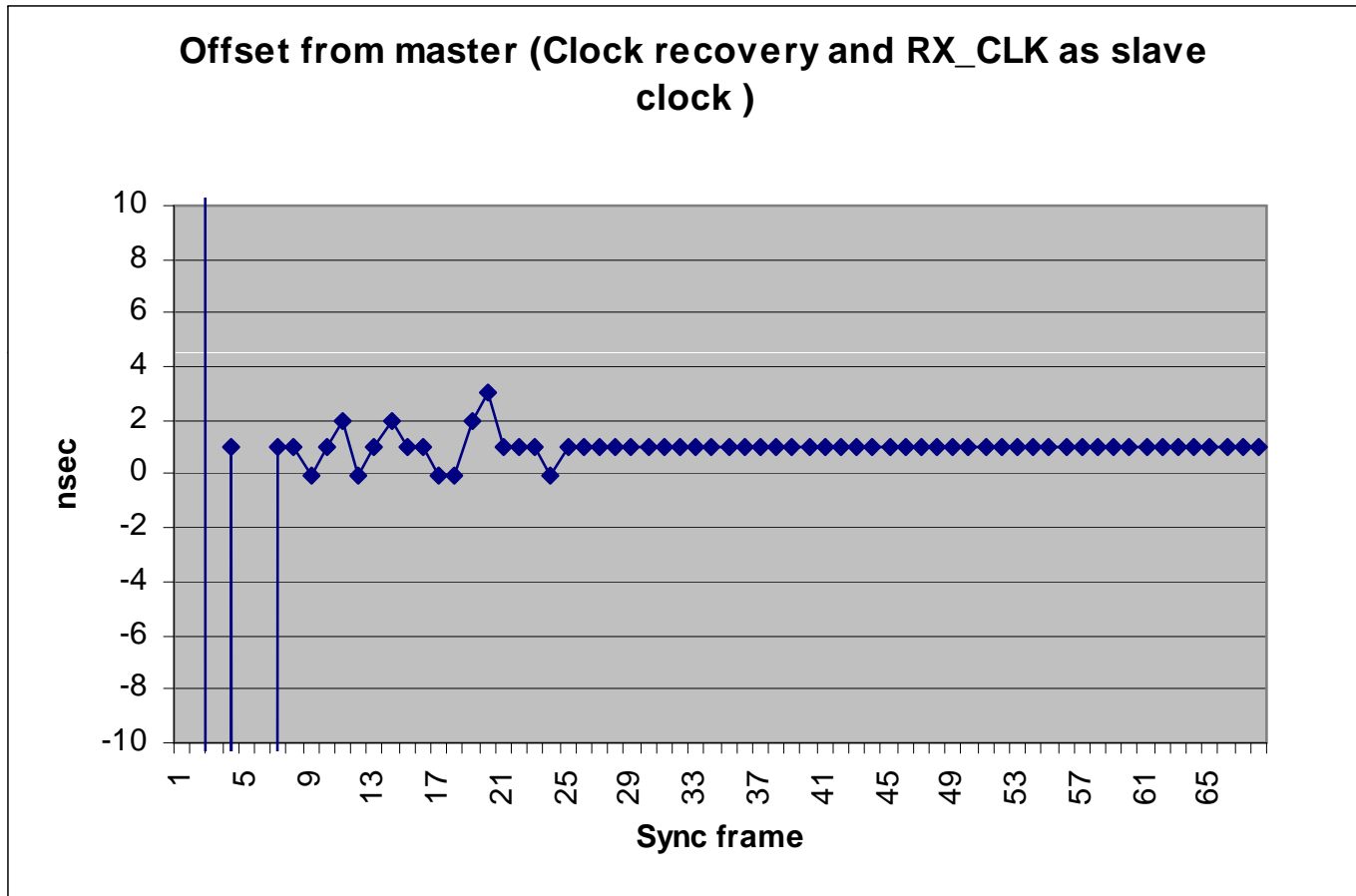
USING RX_CLK as local clock on slave (1)



USING RX_CLK as local clock on slave (2) computed values with timestamp on NIOS



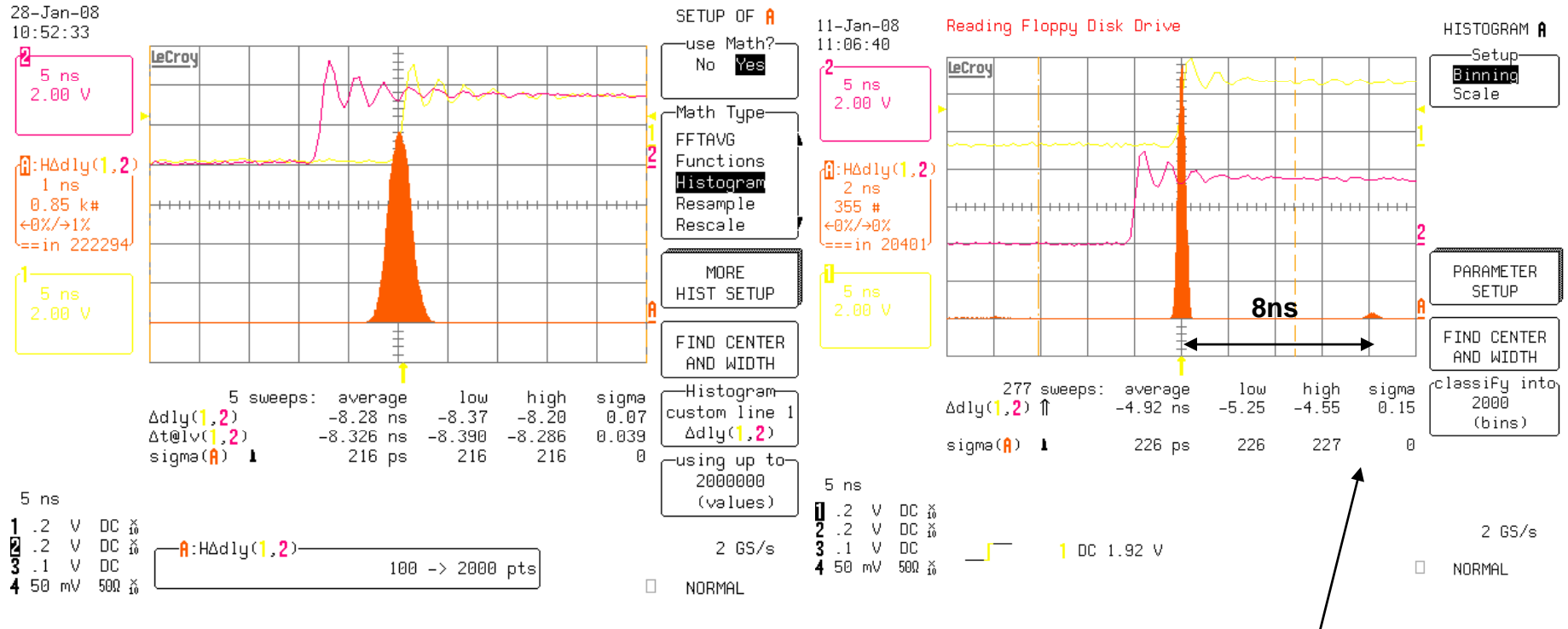
USING RX_CLK as local clock on slave (3) computed value with timestamps on NIOS



The offset error is smaller than the clock resolution

USING RX_CLK (125 MHz) as local clock on slave (4)

Hardware measurement



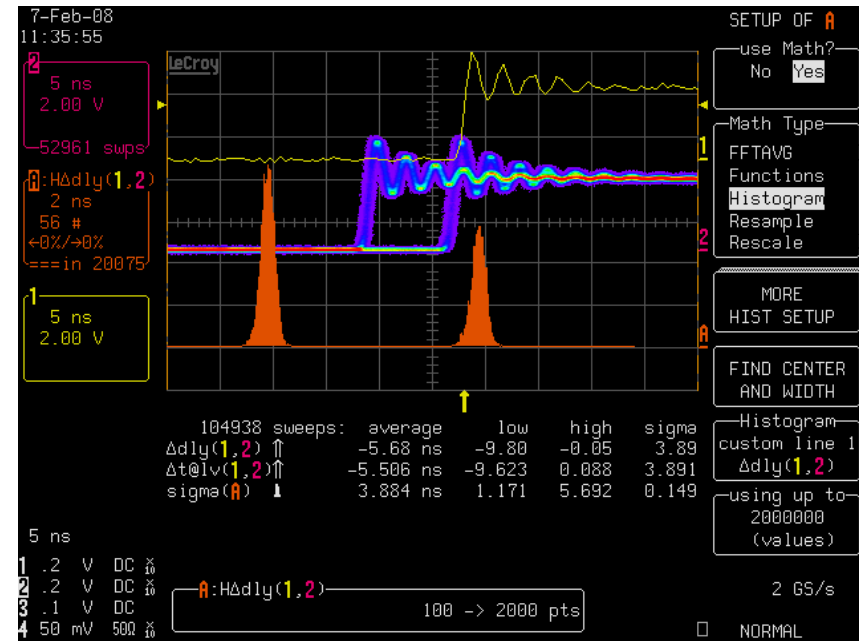
**PPS synchronization
216 ps rms**

**PPS generation base on clock adders
is not optimized in this case.**

**It would be better to use simple tick counts
for timestamp and pps**

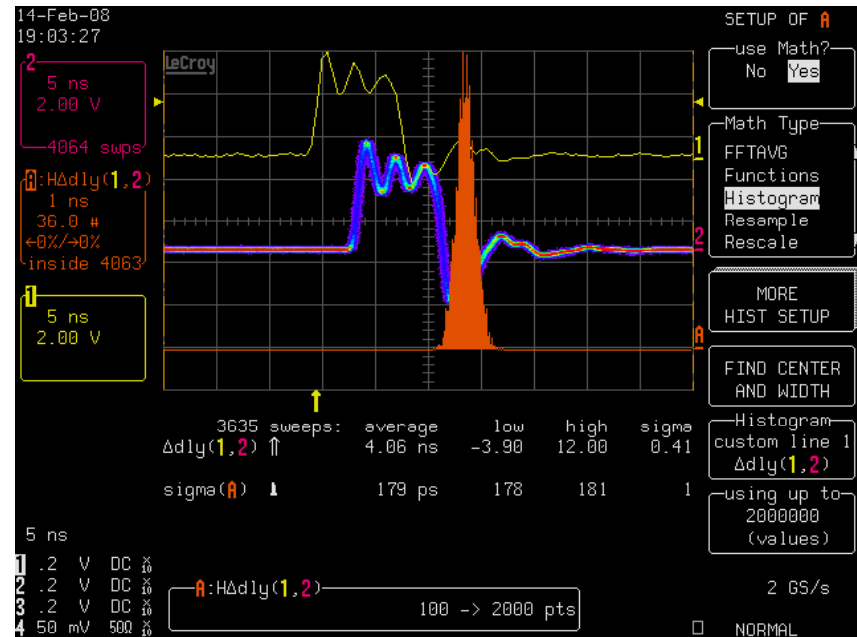
USING RX_CLK (125 MHz) as local clock on slave (5)

Clock based on clock period increment and accumulators



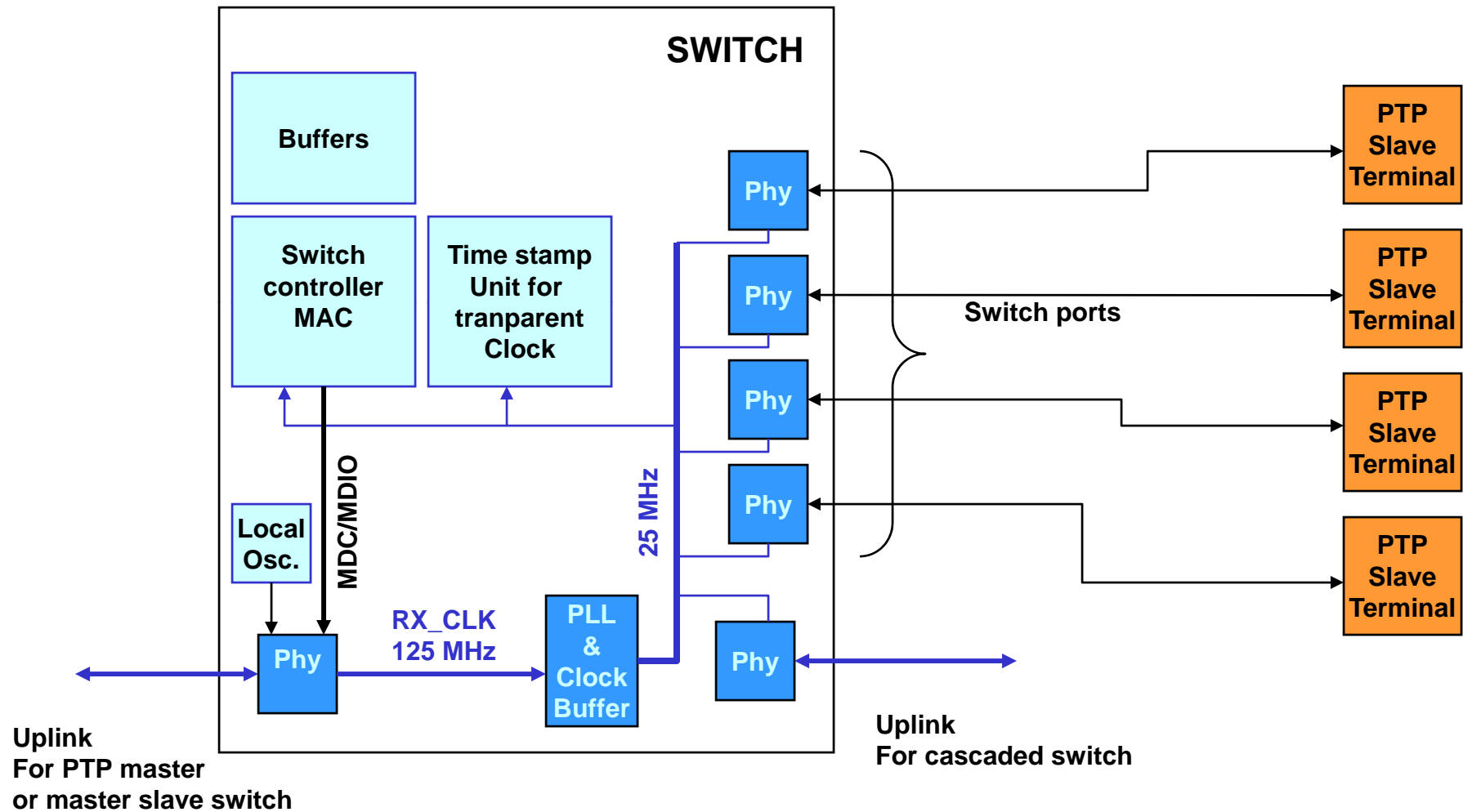
This problem is probably due the finite precision of the clock period increment and the timestamps

Clock based on clock tick counters



The timestamp are a number of clock period.
The PPS period is fixed by a number of clock ticks.
The same technic is implemented on the MASTER

Idea of synchronous gigabit switch



references

- **[1]-Design considerations for software only implementations of the IEEE 1588 Precision time protocol**, Kendall Correll, Nick Barendt (VXI technology, Inc. Cleveland, Ohio USA), Michael Branicky (EECS Dept, Case Western Reserve University Cleveland, USA).
- **[2]- IEEE 1588 standard- IEC 61588.**
- **[3]- IEEE-1588 Standard For a Precision Clock Synchronization Protocol for Networked [1]- [4]- Measurement and Control Systems**, John Eidson, Mike Fisher, Joe White. 34th Annual Precise Time and Time Interval (PTTI) meeting.
- **[5]- Tutorial on IEEE 15888**, John Eidson October 10, 2005- Agilent technologies.
- **[6]- Tutorial on IEEE 1588**, Prof. Hans Weibel, Zurich University of Applied Sciences Winterthur (ZIW).
- **[7]- DP83640 IEEE 1588 PTP Synchronized Clock Output** – Application Note AN-1729 – National Semiconductor Inc.
- **[8]- DP83640 Synchronous Ethernet Mode: Achieving Sub Nanosecond Accuracy in PTP applications.** Application Note AN-1730 – National Semiconductor Inc.
- <http://ieee1588.nist.gov>

Altera SOPC Builder - std_2s60

File Module System View Tools Help

System Contents | Board Settings | **Nios II** More "cpu" Settings | System Generation

Altera SOPC Builder

Create New Component...

Avalon Components

- [-] Nios II Processor - Altera
- [-] Bridges
- [-] Communication
- [-] Display
- [-] EP1C20 Nios Developer
- [-] EP1S10 Nios Developmen
- [-] EP1S40 Nios Developmen
- [-] EP2C35 Nios Developer
- [-] EP2S60 DSP Board Stratix
- [-] EP2S60 Nios Developmen
- [-] Ethernet
- [-] Interfaces and Periphera
 - [-] IFI_GMACII - IFI Tech
- [-] Legacy Components
- [-] Memory
- [-] Other
- [-] User Logic
 - [-] avalon_ptp_clock
 - [-] avalon_ptp_mac_rx
 - [-] avalon_ptp_timestar

Installed Components

Add... Check

Target

Board: Nios Development Board, Stratix II (EP2560) RoHS

Device Family: Stratix II HardCopy Compatible

Clock	Source	MHz	Pipeline
ssram_clk	c2 from pll_sys	100.0	<input type="checkbox"/>
rx_clk	External	125.0	<input type="checkbox"/>
clk_100	c0 from pll_sys	100.0	<input type="checkbox"/>

Use	Module Name	Description	Input Clock	Base	End	IRQ
<input checked="" type="checkbox"/>	cpu	Nios II Processor - Altera Corporation	clk_100			
	instruction_master	Master port				
	data_master	Master port				
	jtag_debug_module	Slave port				
<input checked="" type="checkbox"/>	ifi_gmacii_0	IFI_GMACII - IFI Technologie	clk_100			
	gmacii_master	Master port				
	gmacii_slave	Slave port		0x01200000	0x0121FFFF	7
	dpram_slave	Slave port		0x00000000	0x000FFFFFFF	
<input checked="" type="checkbox"/>	onchip_memory_DPRAM_4K	On-Chip Memory (RAM or ROM)	ssram_clk			
<input checked="" type="checkbox"/>	ext_ssram_bus	Avalon Tristate Bridge	clk_100			
<input checked="" type="checkbox"/>	ext_ssram	Cypress CY7C1380C SSRAM		0x01000000	0x011FFFFFFF	
<input checked="" type="checkbox"/>	ddr_sdram	DDR SDRAM Controller MegaCore Function - Altera Corp...	clk_100	0x02000000	0x03FFFFFFF	
<input checked="" type="checkbox"/>	uart1	UART (RS-232 serial port)	clk_100	0x01232000	0x0123201F	2
<input checked="" type="checkbox"/>	sysid	System ID Peripheral	clk_100	0x01232020	0x012320D7	
<input checked="" type="checkbox"/>	sys_clk_timer	Interval timer	clk_100	0x01232030	0x0123203F	0
<input checked="" type="checkbox"/>	seven_seg_pio	PIO (Parallel I/O)	clk_100	0x01232080	0x0123208F	
<input checked="" type="checkbox"/>	pll_ptp	PLL (Phase-Locked Loop)	clk_in_ptp	0x01232040	0x0123205F	
<input checked="" type="checkbox"/>	reconfig_request_pio	PIO (Parallel I/O)	clk_100	0x01232090	0x0123209F	
<input checked="" type="checkbox"/>	led_pio	PIO (Parallel I/O)	clk_100	0x012320A0	0x012320AF	
<input checked="" type="checkbox"/>	jtag_uart	JTAG UART	clk_100	0x012320D8	0x012320DF	4
<input checked="" type="checkbox"/>	ext_flash_enet_bus	Avalon Tristate Bridge	clk_100			
<input checked="" type="checkbox"/>	ext_flash	Flash Memory (Common Flash Interface)		0x00000000	0x00FFFFFFF	
<input checked="" type="checkbox"/>	epcs_controller	EPCS Serial Flash Controller	ssram_clk	0x01231000	0x012317FF	6
<input checked="" type="checkbox"/>	button_pio	PIO (Parallel I/O)	clk_100	0x012320C0	0x012320CF	3
<input checked="" type="checkbox"/>	high_res_timer	Interval timer	clk_100	0x01232060	0x0123207F	1
<input checked="" type="checkbox"/>	avalon_ptp_mac_rx_0	avalon_ptp_mac_rx	clk_100	0x01232100	0x0123213F	
<input checked="" type="checkbox"/>	pll_sys	PLL (Phase-Locked Loop)	clk_in	0x01220040	0x0122005F	
<input checked="" type="checkbox"/>	avalon_ptp_clock_0	avalon_ptp_clock	clk_125	0x01220000	0x0122003F	

▲ Move Up ▼ Move Down

⚠ Pin assignments for **ifi_gmacii_0** must be made in your Quartus II project.

ℹ **cpu** was generated with full capabilities and must be compiled in Quartus II with the same license.

ℹ Done checking for updates.

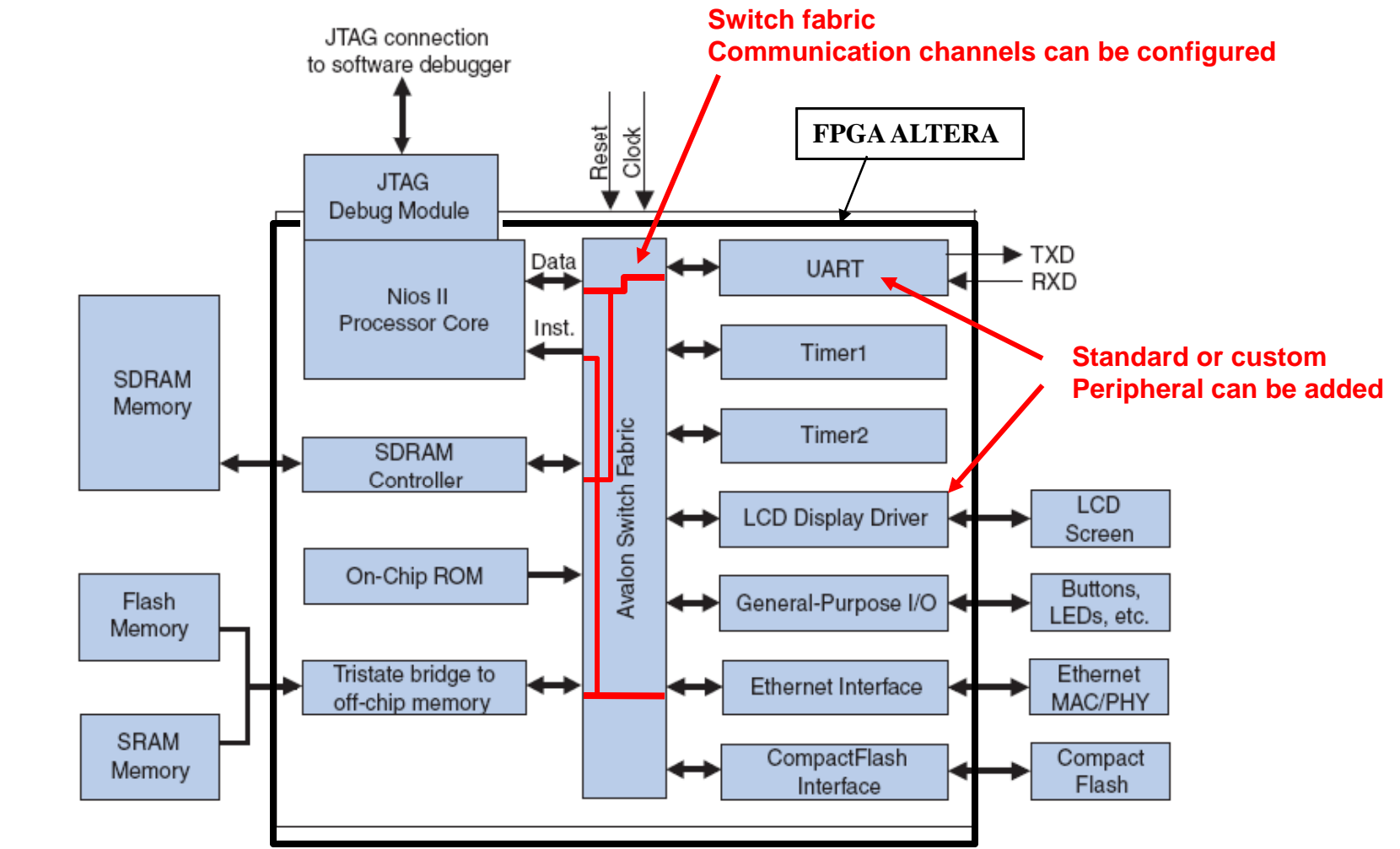
Exit
< Prev
Next >
Generate

Gigabit
MAC IP

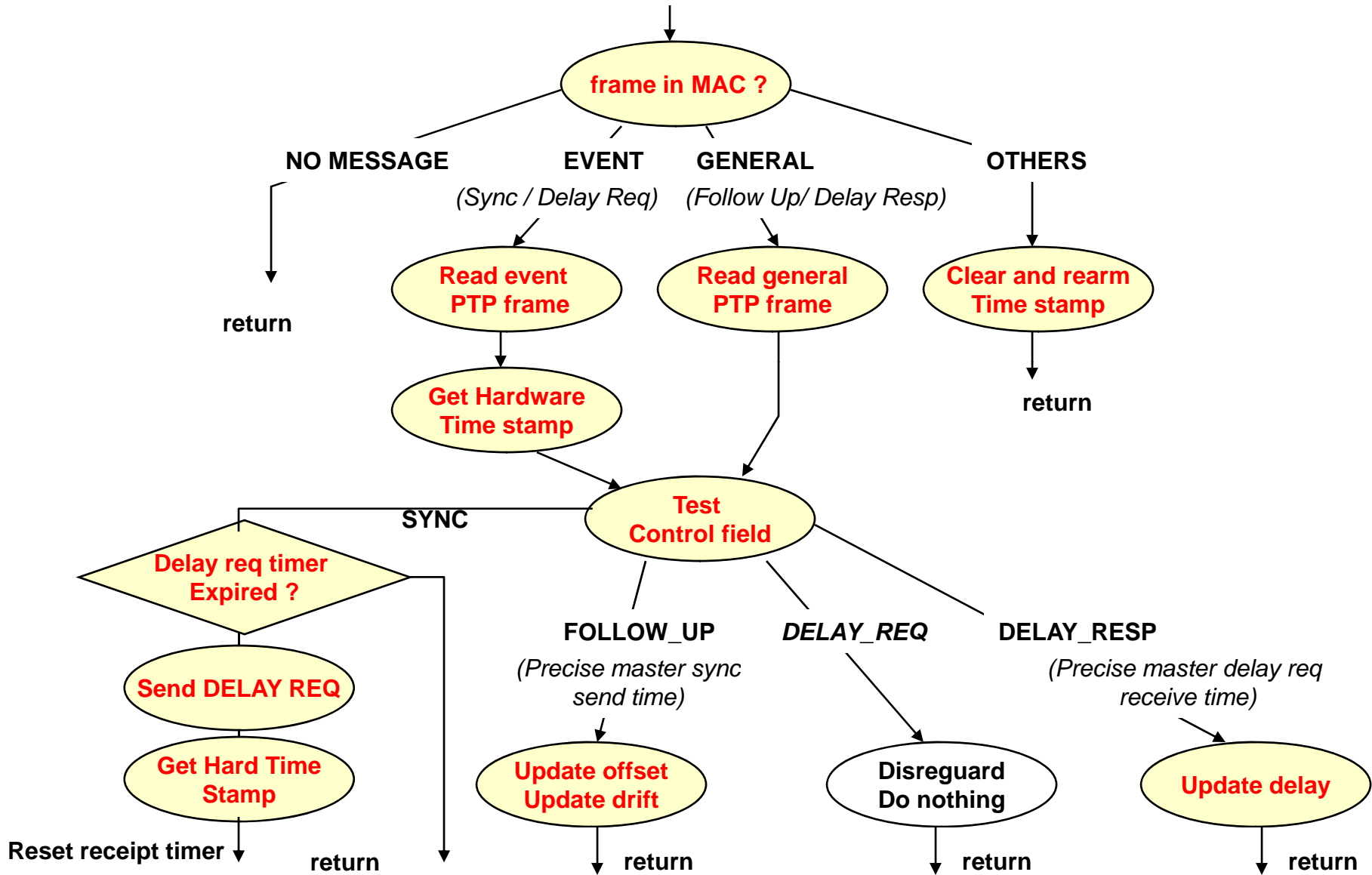
PTP
Clock

Default development board configuration

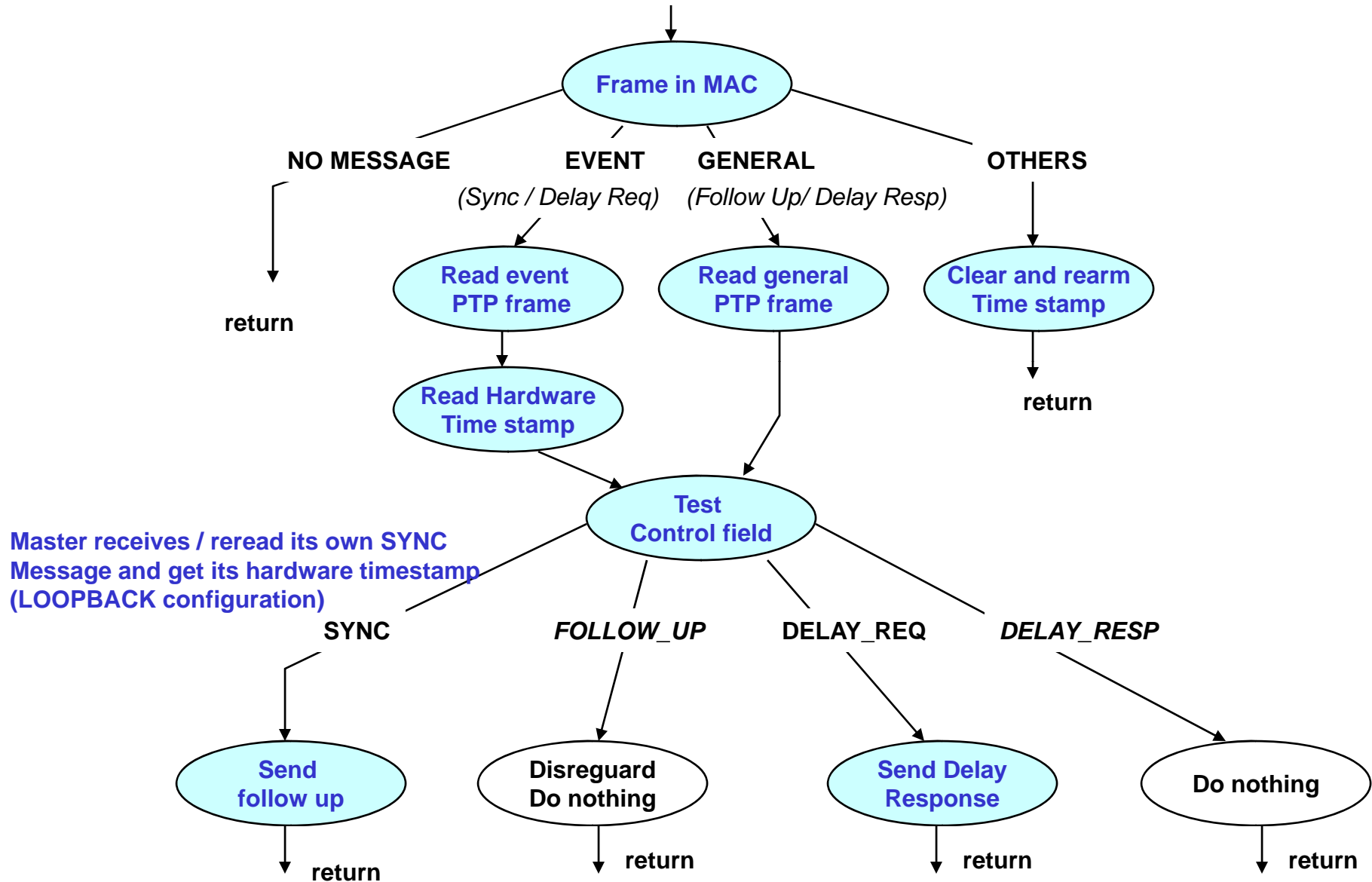
Figure 1-1. Example of a Nios II Processor System



Simplified ptp message processing (SLAVE)



Simplified ptp MASTER processing



Drift correction

On slave t_s^k Is measured with an offset correction of $offset_s^{k-1}$

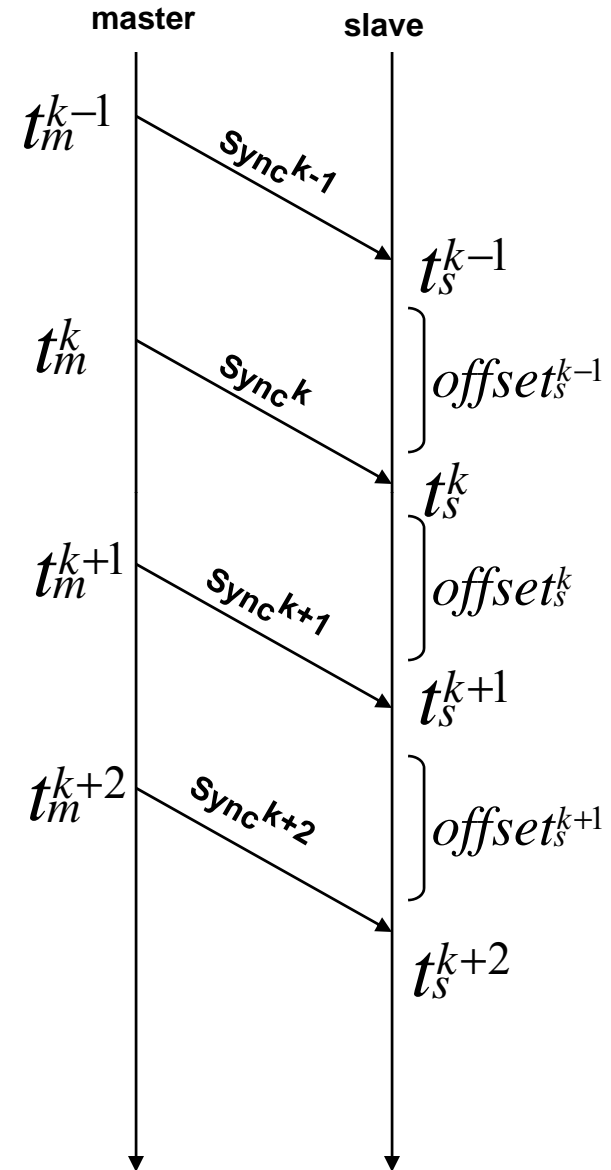
On slave t_s^{k+1} Is measured with an offset correction of $offset_s^k$

To avoid a drift measurement error due to offset correction between two consecutive SYNC we subtract the offset difference

$$drift = \frac{(t_m^{k+1} - t_m^k) - (t_s^{k+1} - t_s^k) - (offset^k - offset^{k-1})}{(t_s^{k+1} - t_s^k) - (offset^k - offset^{k-1})}$$

The period increment T_{incr}^n Is corrected in the following way

$$T_{incr}^n = T_{incr}^{n-1} + T_{incr}^{n-1} \times (-drift)$$



Offset correction

The well known formula for offset are :

$$oneWayDelay = \frac{(t_s^k - t_m^k) + (t_m^j - t_s^j)}{2}$$

$$offset^n = (t_s^n - t_m^n) - oneWayDelay$$

The new offset register value $offset^{n+1}$

Is corrected in the following way

$$offset_reg^n = -offset^n + offset^{n-1}$$

