The White Rabbit project

Sub-nanosecond synchronization and determinism in accelerator controls

+ some thoughts on RF distribution over Ethernet

T. Włostowski

BE-CO Hardware and Timing section
CERN

September 30, 2011

Outline

Origin of the name

Oh dear! Oh dear! I shall be too late!

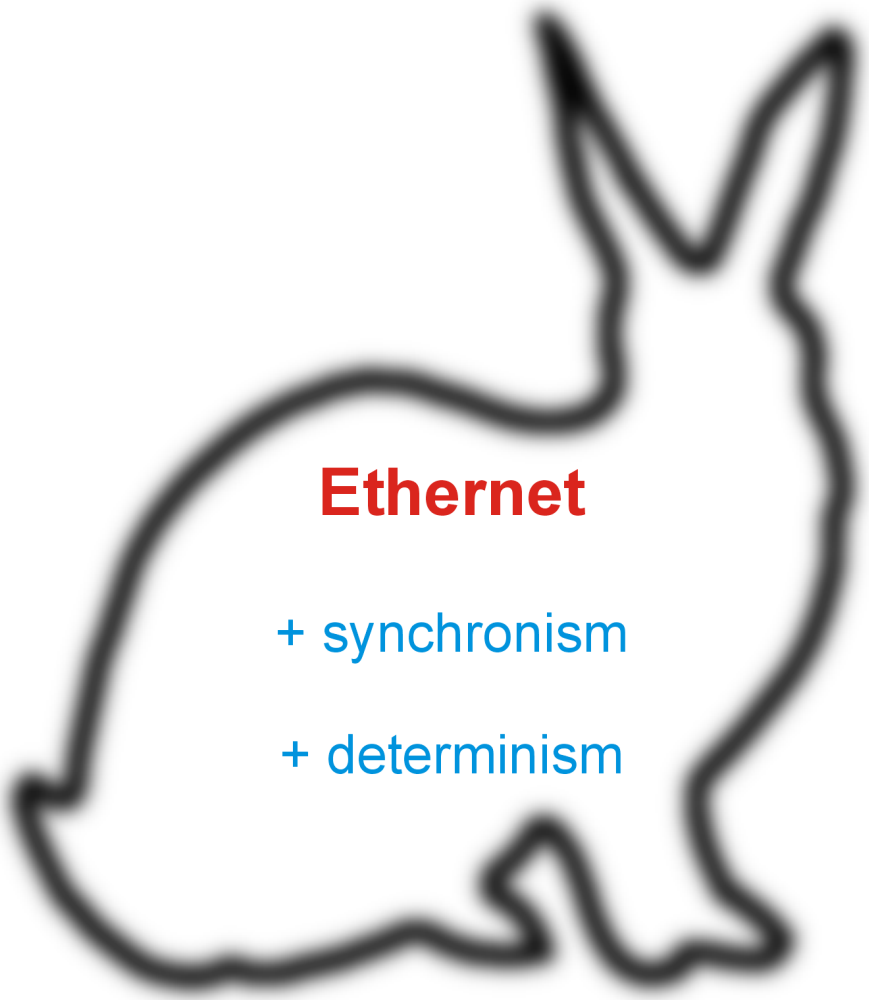
The White Rabbit in charge of real time

Why White Rabbit?

General Machine Timing system works fine, but has some limitations:

- Low speed (500 kbps)
- Lack of bidirectional communication
- Manual delay compensation
- Very CERN-specific solution
- Complicated maintenance

What is White Rabbit?



What is White Rabbit?

An **extension** to **Ethernet** which provides:

- **Synchronous mode** (Sync-E) - common clock for physical layer in entire network, allowing for precise time and frequency transfer.
- **Deterministic routing** latency - a guarantee that packet transmission delay between two stations will never exceed a certain boundary.

Design goals

Scalability

Up to 4000 nodes.

Range

10 km fiber links.

Accuracy and precision

1 ns end-to-end sync accuracy, 20 ps jitter.

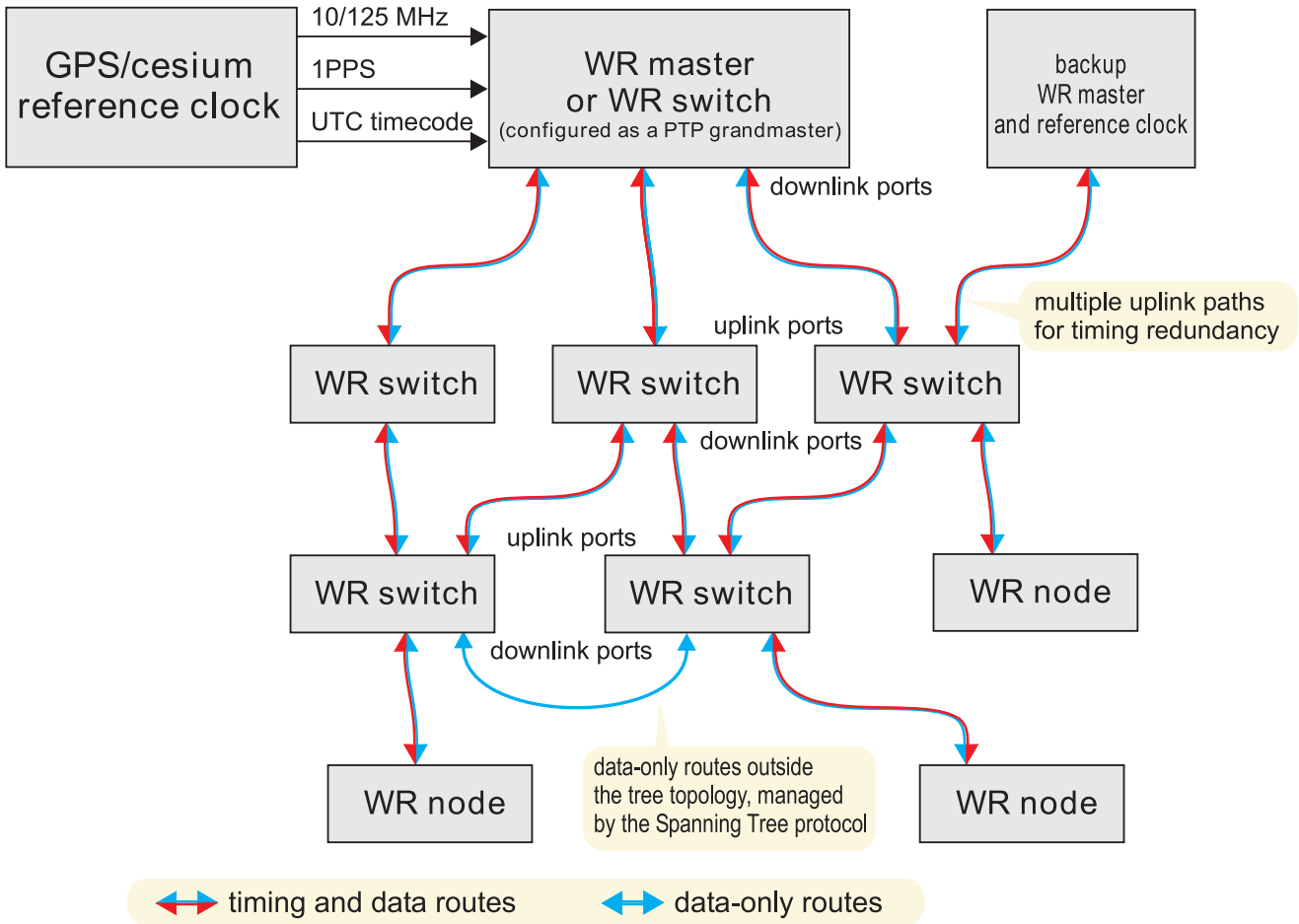
Outline

Technologies used in White Rabbit

Sub-nanosecond synchonization in WR is achieved by using the following three technologies together:

- Precision Time Protocol (IEEE1588)
- Synchronous Ethernet
- DMTD phase tracking

Network topology



PTP Protocol (IEEE1588)

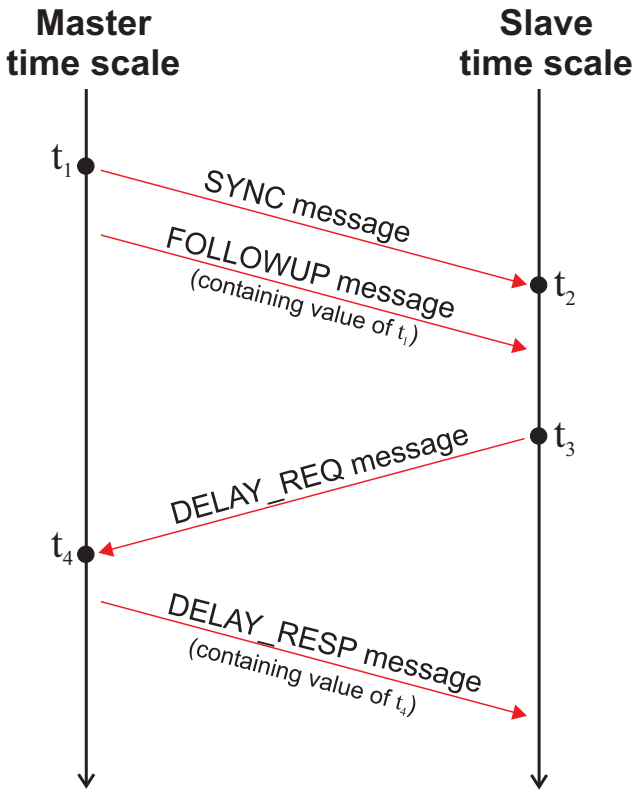
PTP

Synchronizes local clock with the master clock by measuring and compensating the delay introduced by the link.

Packet timestamping

Link delay is measured by exchanging packets with precise hardware transmit/receipt timestamps

PTP Protocol (IEEE1588)



Having values of $t_1 \ldots t_4$, slave can:

- calculate one-way link delay: $\delta_{ms} = \frac{(t_4 - t_1) - (t_3 - t_2)}{2}$
- syntonize its clock rate with the master by tracking the value of $t_2 - t_1$
- compute clock offset: $offset = t_2 - t_1 + \delta_{ms}$

Disadvantages of traditional PTP

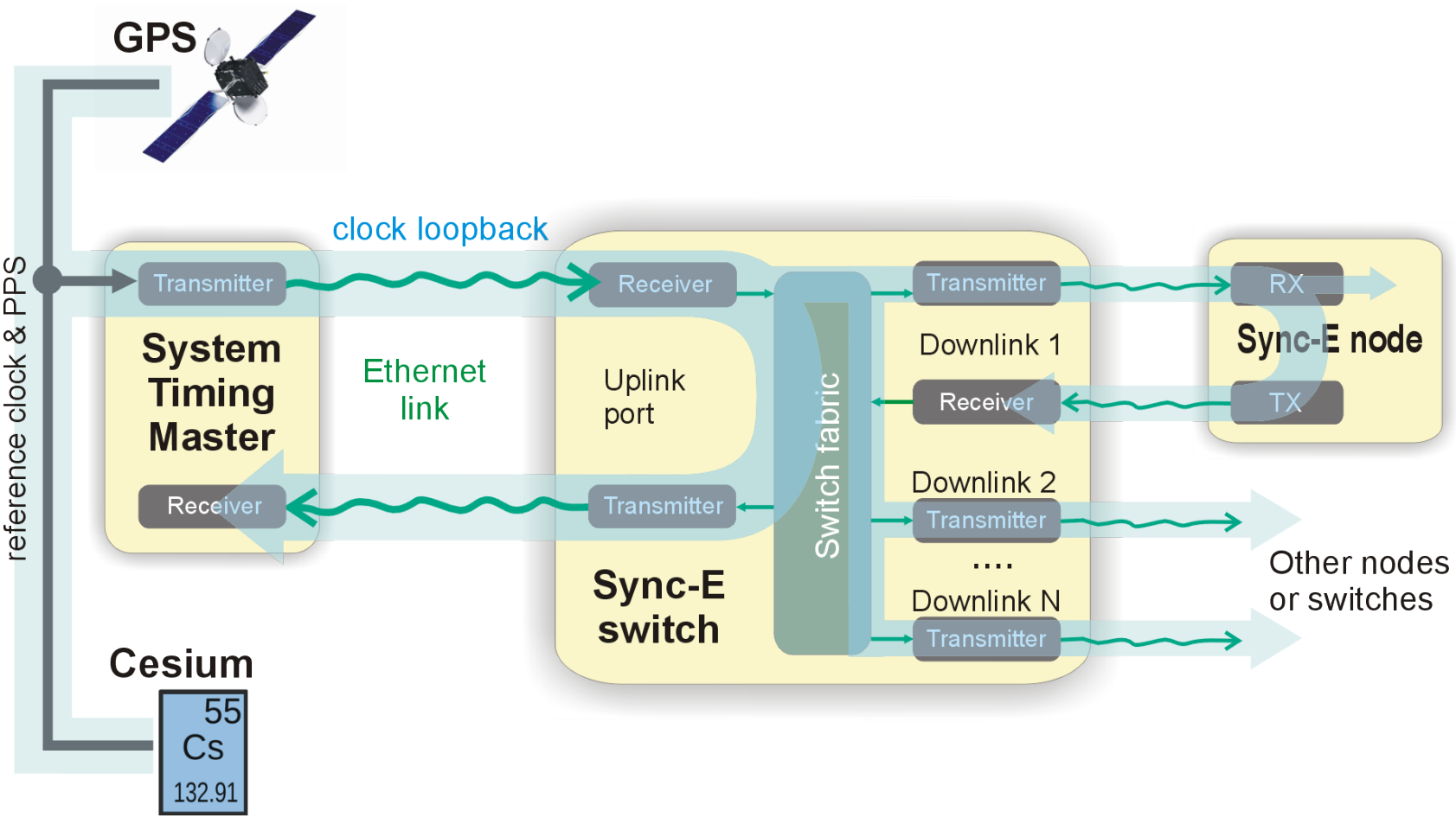
- Syntonization performance depends on network traffic. Better clock = more packets.
- ... which doesn't go well with the determinism.
- Rate of slave clock is adjusted with timestamped packets causing high low frequency jitter.
- Timestamping precision is limited to single PHY clock cycle (8 ns for gigabit Ethernet)
- No method for compensating link asymmetry

Synchronous Ethernet

Common clock for the entire network

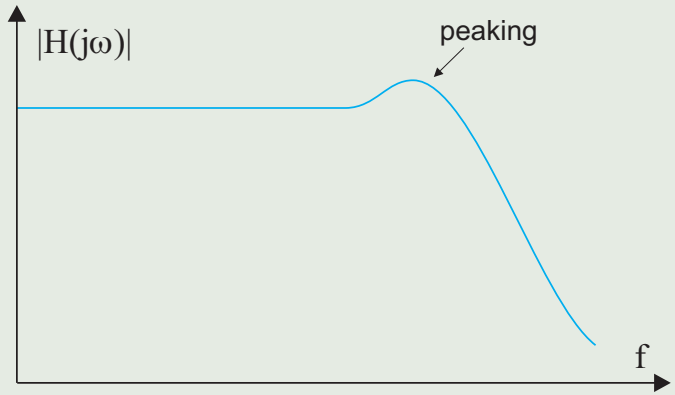
- All network nodes use the same physical layer clock, generated by the System Timing Master
- Clock is encoded in the Ethernet carrier and recovered by the PLL in the PHY.
- PTP is used only for compensating clock offset
- With synchronous mode, we can use phase measurements instead of direct timestamping.

Synchronous Ethernet



Sync-E limitations

Cascaded PLLs are prone to **gain peaking** effect, which may cause instability of whole system:



Solution

- avoid daisy-chaining nodes, use tree-like topology
- careful design of loop filters

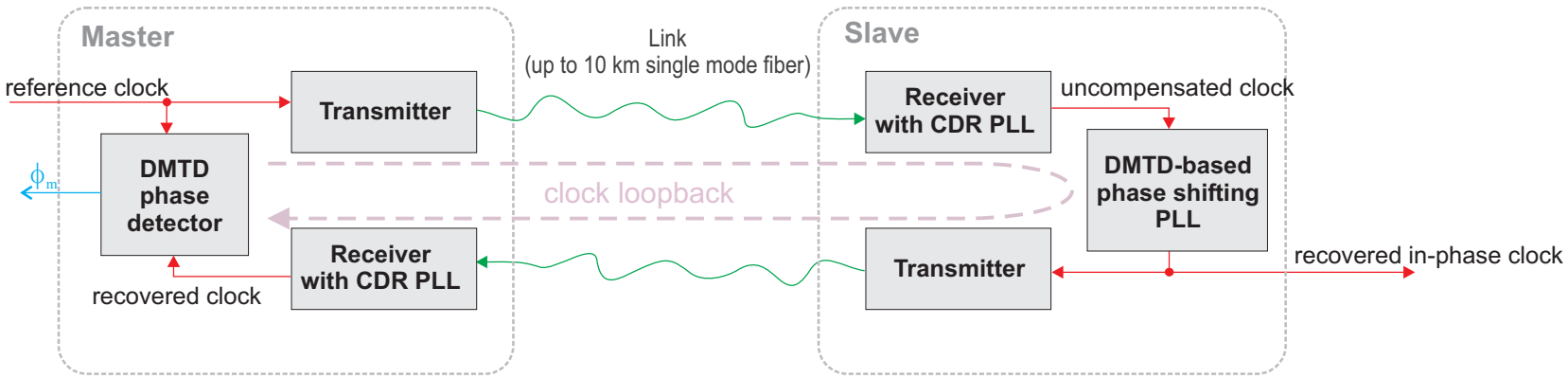
Plain PTP

PTP alone is not enough if we want very good accuracy because of the granularity of the timestamps.

Solution

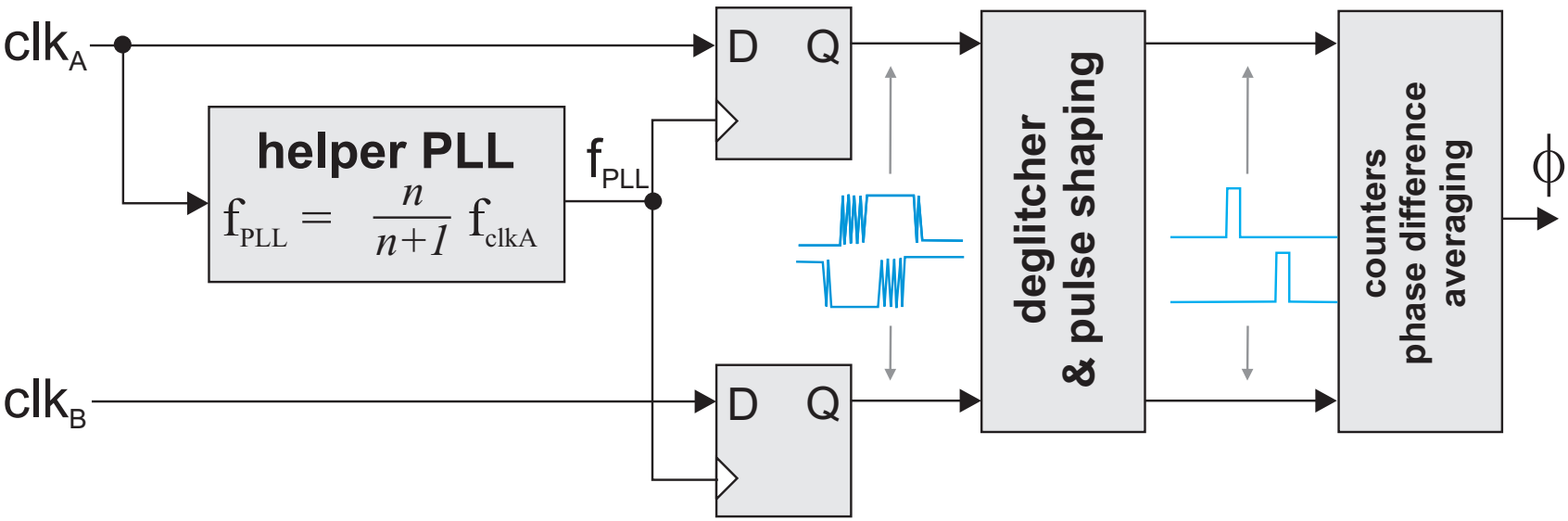
Measure the phase shift between transmit and receive clock on the master side, taking the advantage of Synchronous Ethernet.

Phase tracking



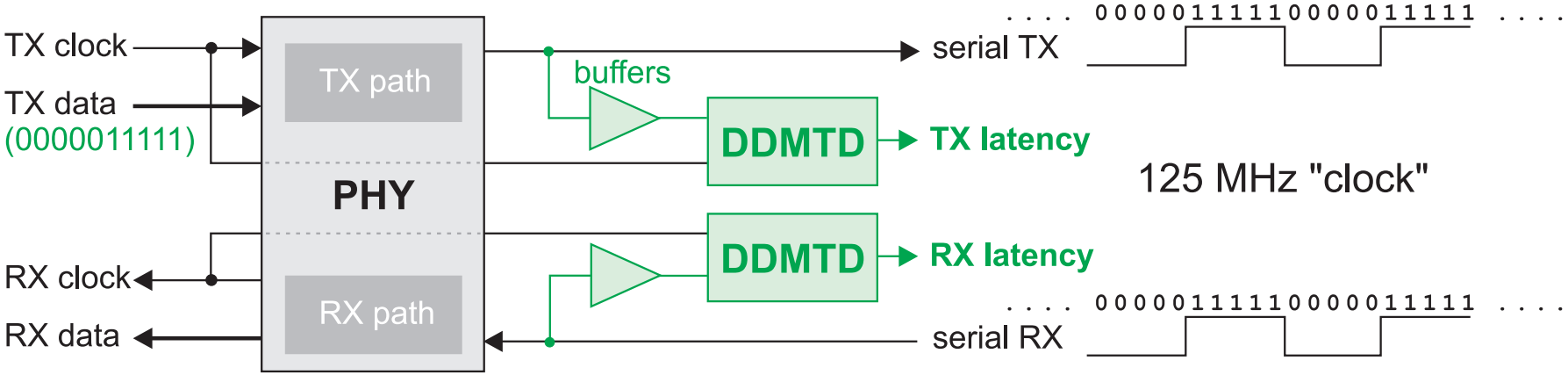
- Monitor phase of bounced-back clock continuously
- PLL in the slave follows the phase changes measured by the master
- Every clock tick is put to good use: performance is equivalent to PTP with messages exchanged every 8 ns.

Digital Dual Mixer Time Domain (DMTD) phase detector



- Fully digital, so fully linear
- In a loop, it becomes a linear phase shifter
- Can handle multiple channels without need for extra hardware

Dealing with link asymmetry



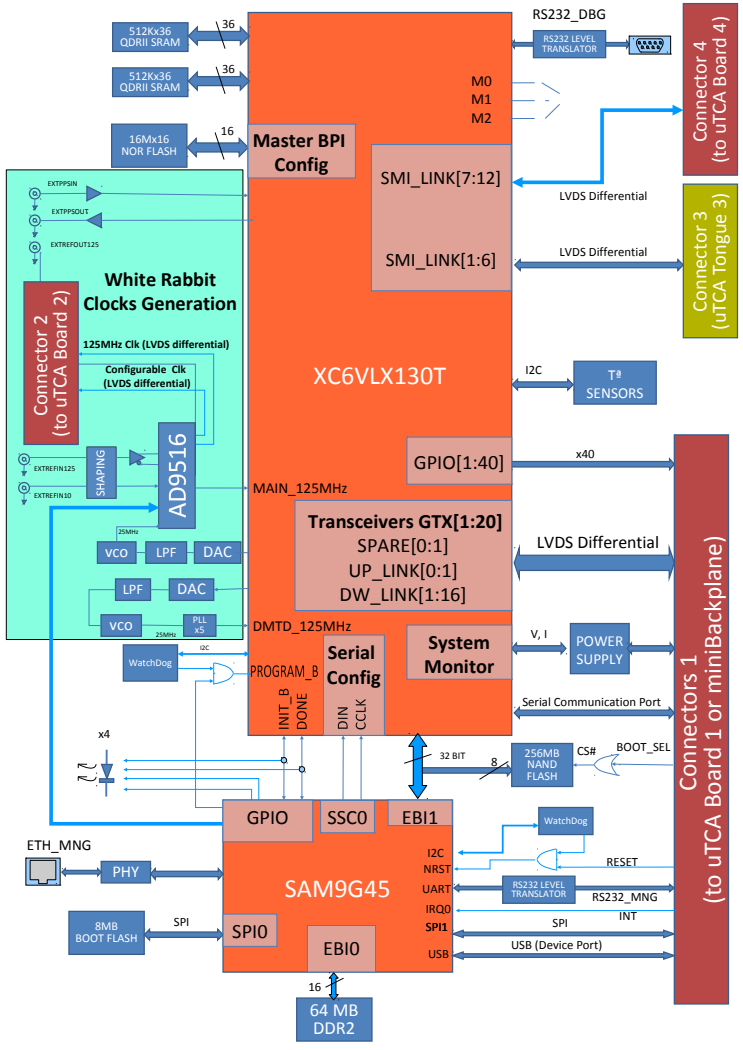
- PHY TX/RX latencies differ between each power-up cycle, but they can be measured using a DMTD and a crosspoint switch.
- Fiber delay asymmetry is caused by difference in wavelengths for upstream and downstream traffic and it proportional to the round-trip link delay.

Outline

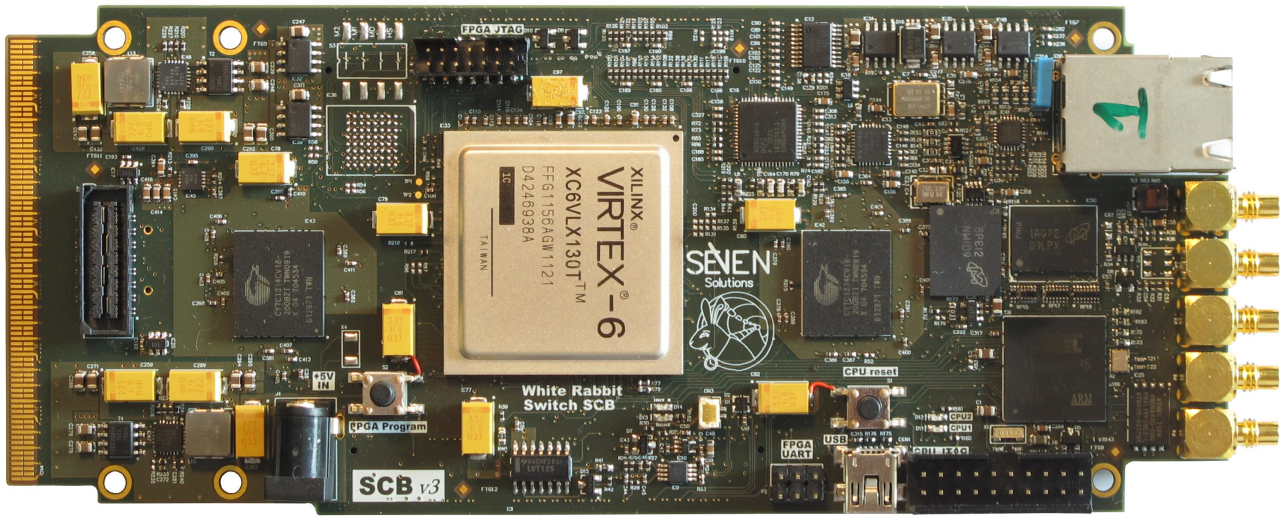
Specifications

- Core component of a WR network
- Custom design, built from scratch
- uTCA MCH format
- 18 Gigabit Ethernet ports (fiber)
- Fully compatible with 802.1 and 802.3
- 300 ps synchronization accuracy
- Can work as a master clock, referenced to external 10 MHz / PPS
- Copper support (1000Base-T) in development
- Stand-alone operation possible via mini-backplane

Block diagram



Version 3 Hardware



- FPGA implements the actual Ethernet switch and low level timing functions (PLL, DMTDs and fine timestamping)
- CPU runs the PTP stack, Spanning Tree and management protocols

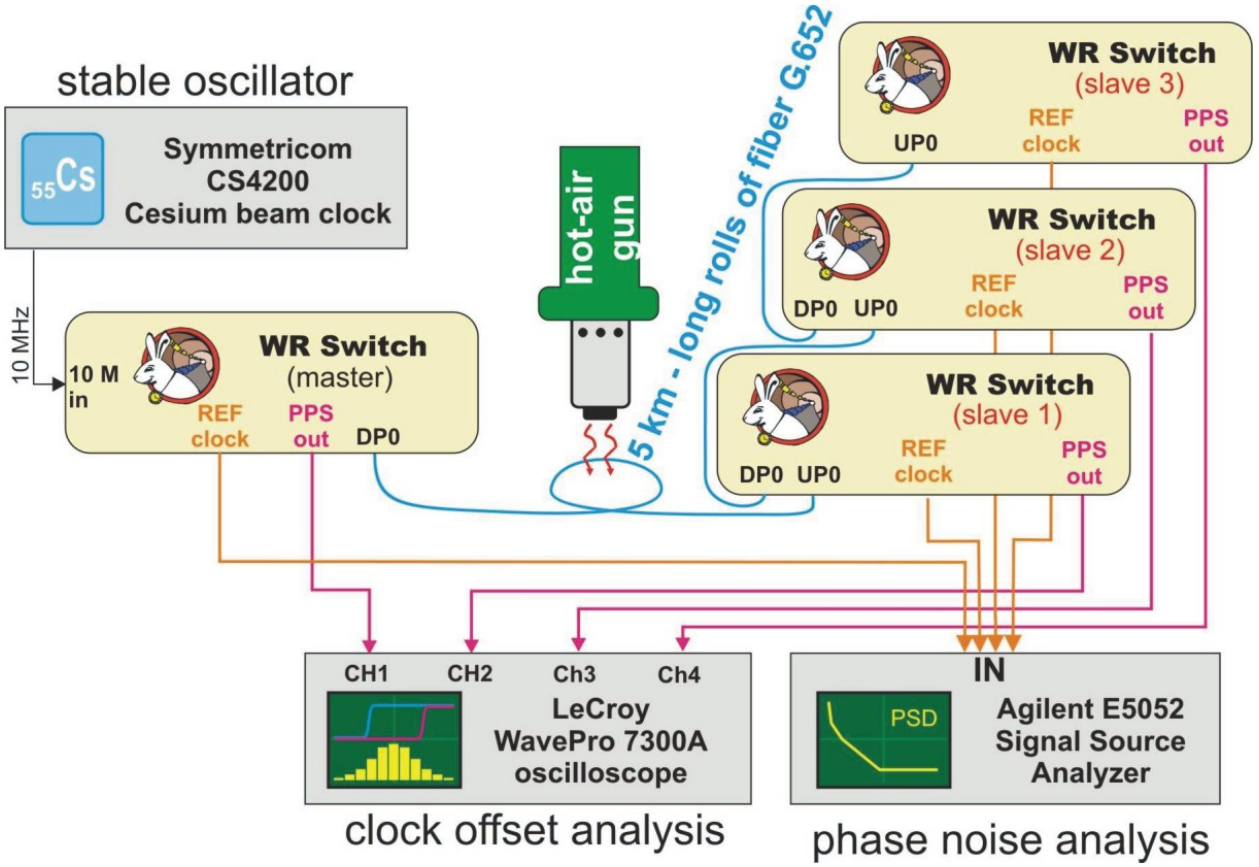
WR Switch FPGA

- Low-cost Virtex-6 (XC6VLX130T)
- Uses GTX transceivers with "bit-slip" calibration
  - No external components required
- Does all low-latency packet processing:
  - Hybrid switching core: store-and-forward for regular traffic and cut-through bypass for critical control messages
  - Routing Table Unit with hardware-accelerated Spanning Tree protocol to speed up switching over broken links
- Implements all timing components:
  - Embedded LM32 running realtime timing tasks (PLL filters, reference switch/holdover)
  - Sub-nanosecond timestampers and DMTDs
  - The only external components are the oscillators

The CPUs

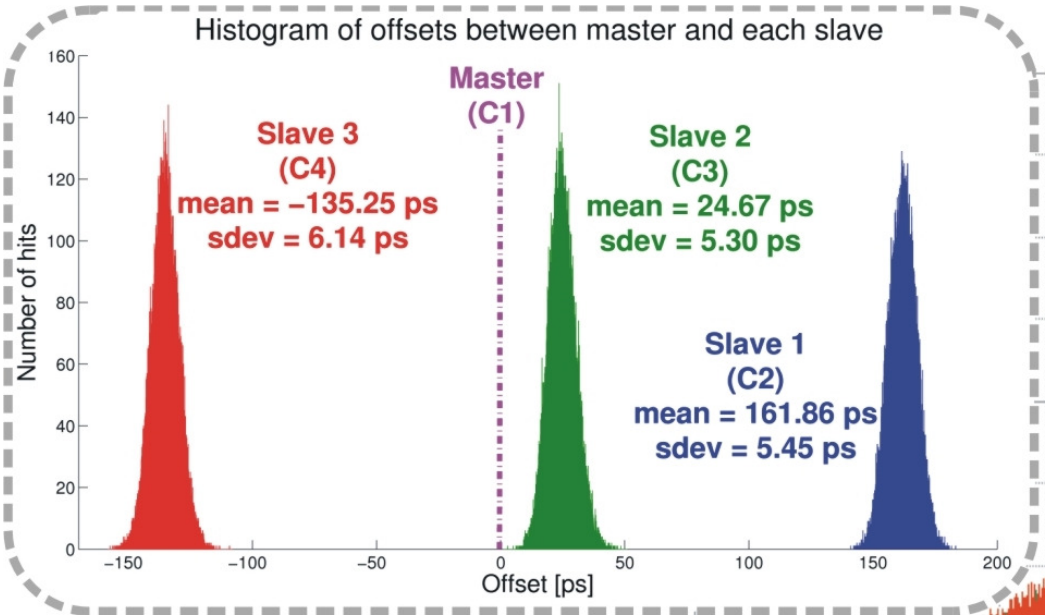
- Atmel's ARM9 (AT91SAM9G45) running Embedded Linux
  - WR-PTP daemon based on `ptpv2d`, completely new modular PTP stack in development.
  - Learning, VLAN and Spanning Tree high level software.
  - SNMP management agent based on `net-snmp`.
- Small Cortex-M3 microcontroller
  - uTCA management
  - watchdog and safe firmware update

Accuracy measurements

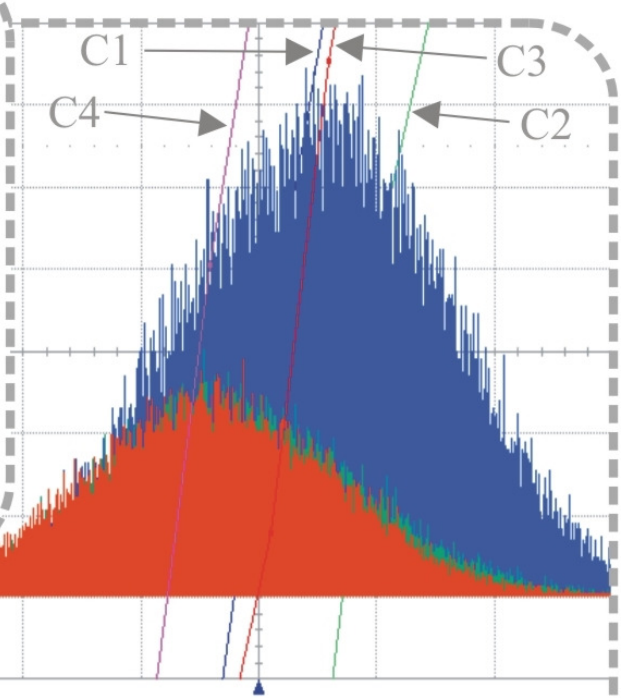


4 V2 switches daisy-chained with 5 km-long fibers

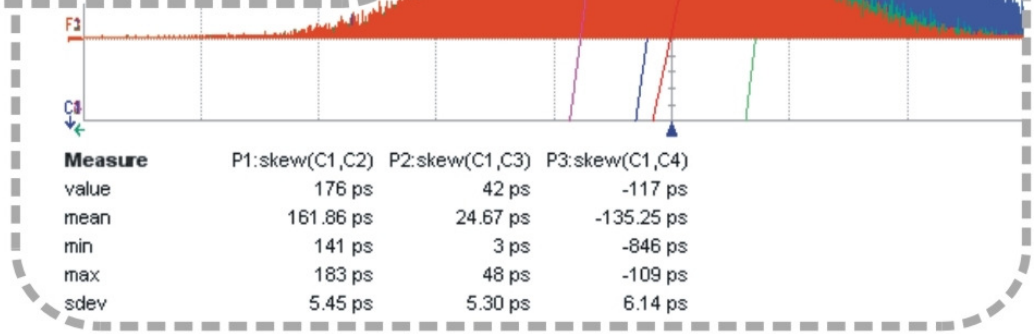
Results



Matlab plot of collected data

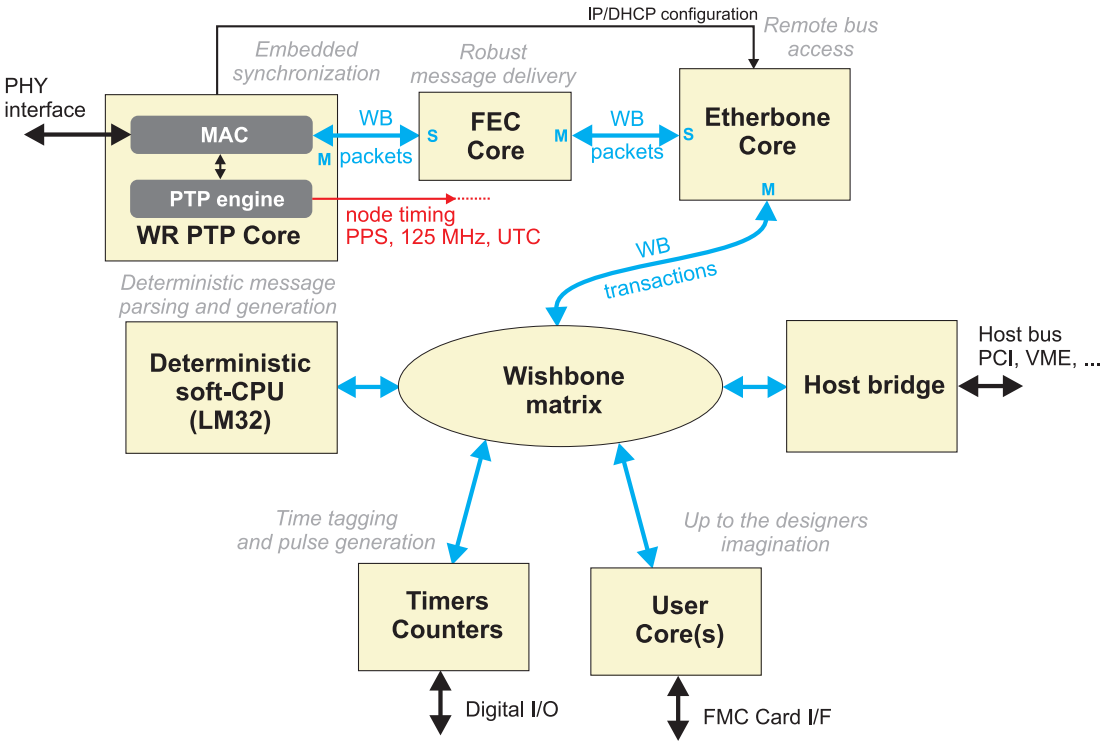


Oscilloscope screenshot



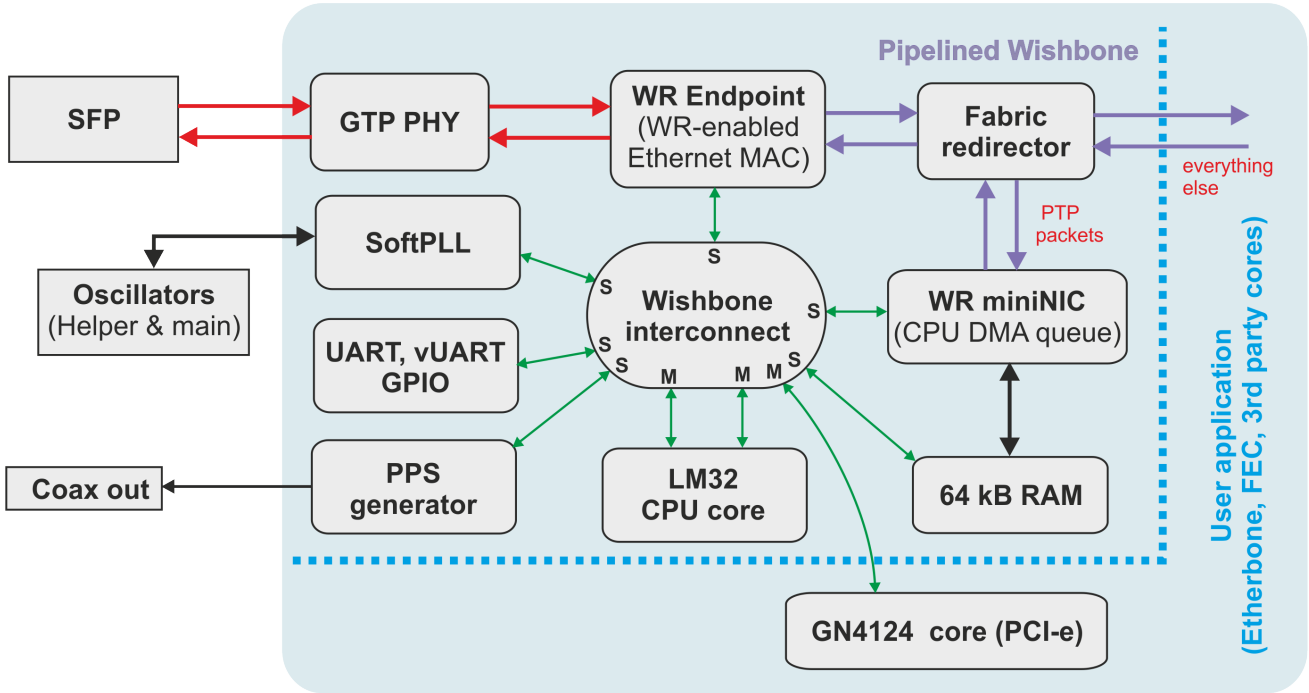
Measure	P1:skew(C1,C2)	P2:skew(C1,C3)	P3:skew(C1,C4)
value	176 ps	42 ps	-117 ps
mean	161.86 ps	24.67 ps	-135.25 ps
min	141 ps	3 ps	-846 ps
max	183 ps	48 ps	-109 ps
sdev	5.45 ps	5.30 ps	6.14 ps

WR Node



The fundamental building block of WR-based control systems.

Node building blocks



WR PTP Core

Black-box FPGA core implementing WR master/slave.

WR PTP Core

- Outputs in-phase 125 MHz, PPS, UTC and filtered packets
- Integrated WR Gigabit Ethernet MAC:
  - TX and RX with sub-ns timestamping
  - Packet classification (PTP/Etherbone/for the host)
  - Directly interfaces with Xilinx GTPs and other PHYs
- LM32 softcore CPU:
  - 32-bit RISC, FPGA-optimized
  - Runs a size-optimized version of PTPd
  - Can be used to configure other IP cores (i.e. Etherbone, FEC)
  - Provides simple sync status monitor and management interface
- SoftPLL:
  - Implements the HW part of the PLLs (just the phase/frequency detectors)
  - Drives the VCXO/VCTCXO DACs
  - Very simple – all calculations done by the CPU

Etherbone Core

A distributed system bus via Ethernet

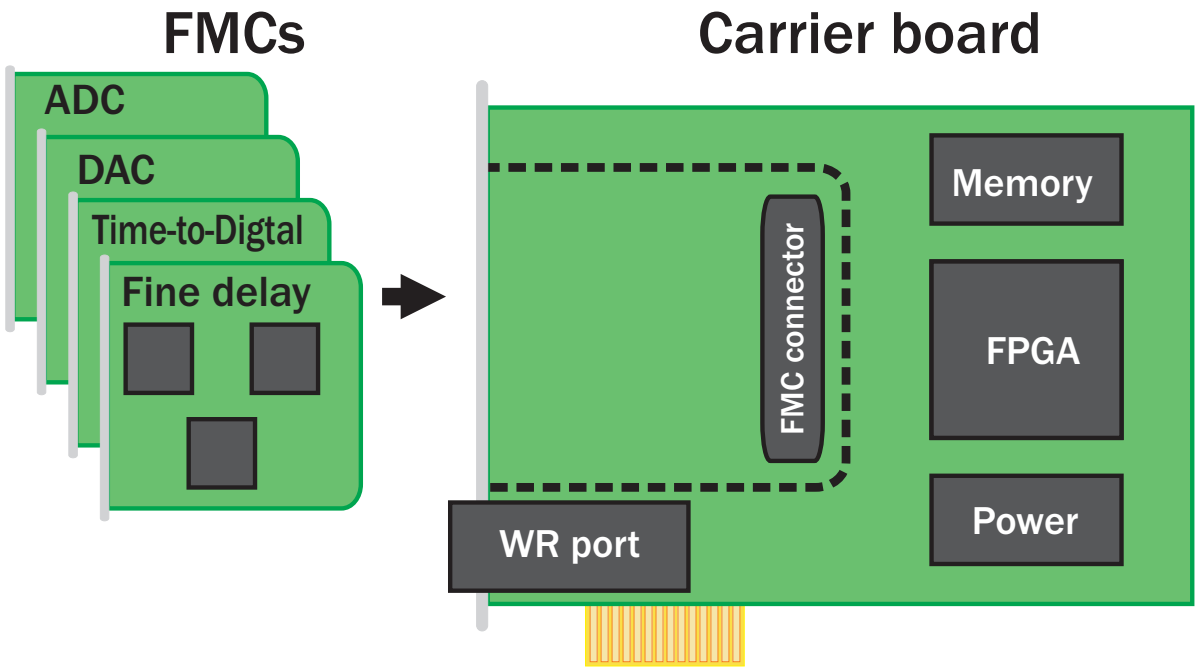
- Takes Ethernet or UDP packets and produces Wishbone transactions.
  - ... and vice versa
  - ... so two Etherbone cores connected together create a transparent Wishbone bridge, which can connect remote FPGAs
- Can be used to control the node by directly accessing its registers from a PC.
- Based on pipelined Wishbone (b4) specification.
- Protocol optimized for low latency.

Softcore CPU

Control Message parser and generator

- LM32 pipelined RISC CPU (independent from the WRPC CPU)
- Fully deterministic: very simple timing model.
- Open core: supports application-specific instructions.
- Flexible: can build WR control system on top of existing protocols.

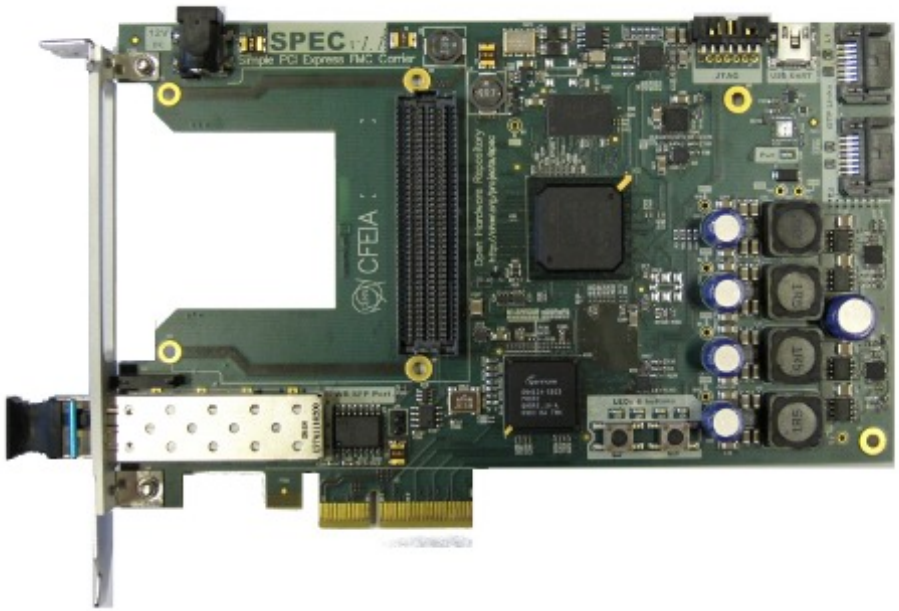
WR Node Hardware Kit



Our approach

- General purpose **carriers** in PCIe, VME, uTCA formats
- **FMC mezzanines** implementing users' functions (ADC, DAC, TDC)

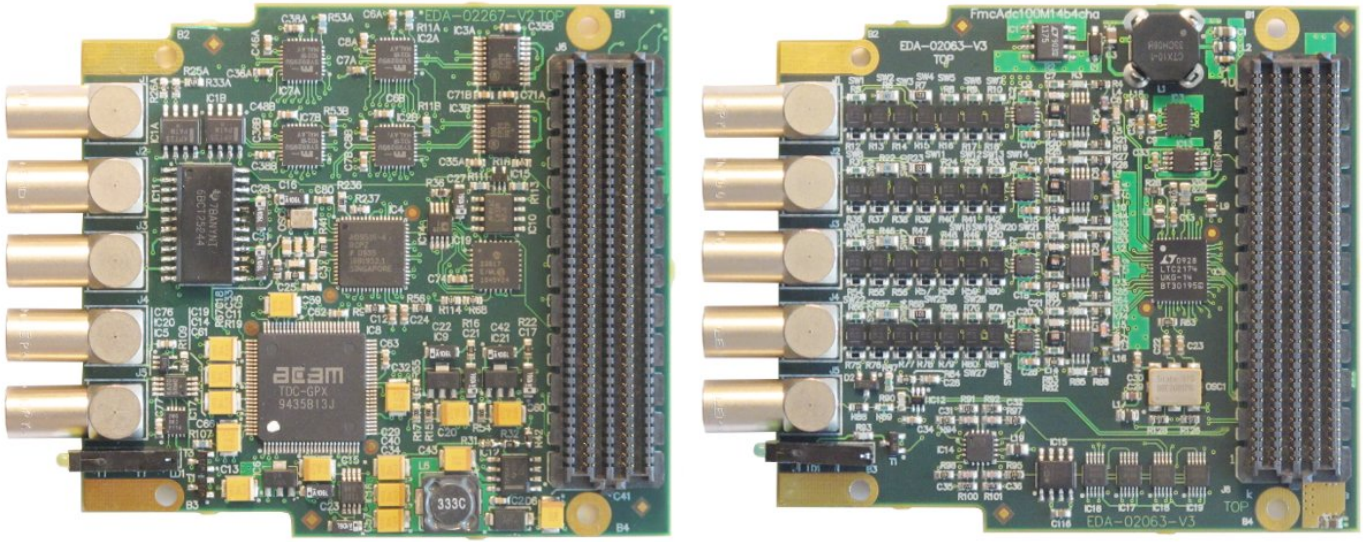
Example: SPEC card



Simple PCI Express Carrier

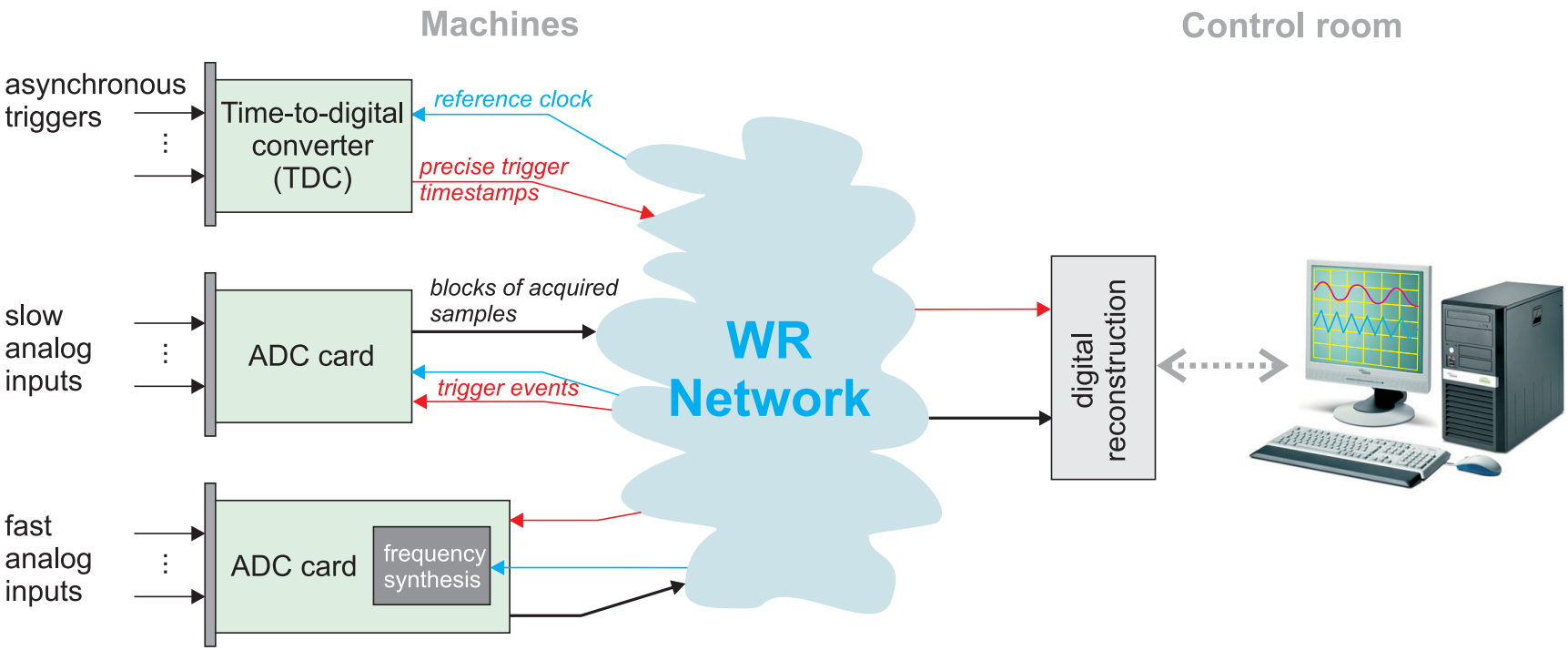
- Reference platform for WR node development.
- Spartan-6 FPGA (XC6SLX45T) + 256 MB DDR3 RAM.
- FMC low pin count connector.

Example: CERN's FMC projects



- 16-bit, 100 MSPS, 4-channel ADC
- Fine delay generator (1 ns accuracy, 300 ps jitter)

Example: Distributed oscilloscope



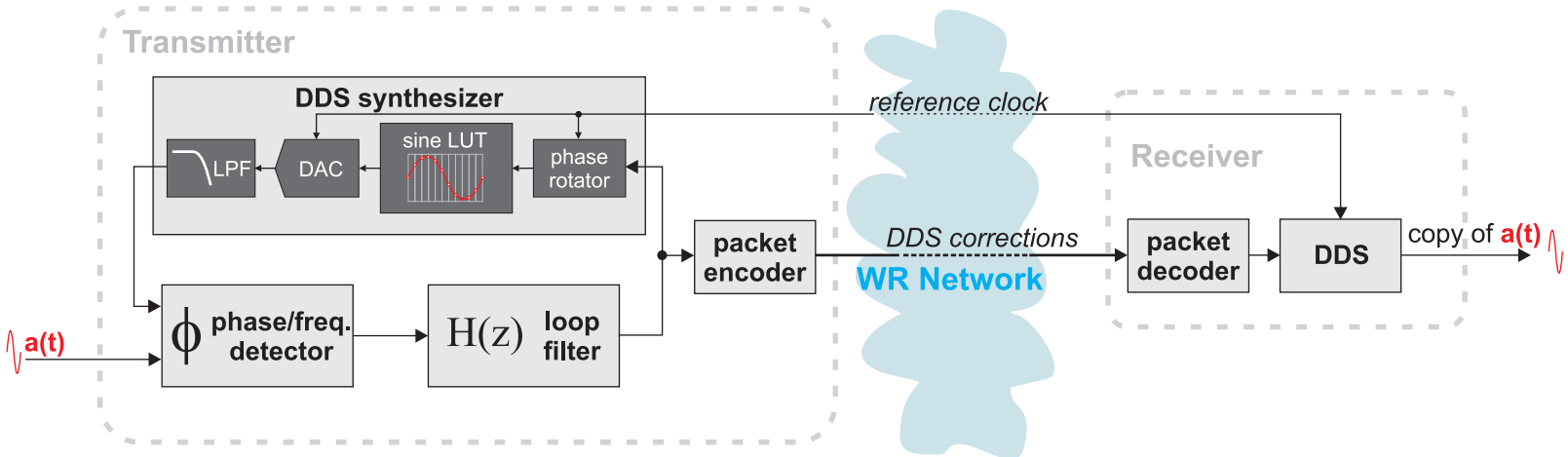
Outline

RF over Ethernet

Why?

- Dozens of analog cables replaced with a single fiber
- Distance has no impact on the quality of the distributed signals
- @CERN: all timing signals (bunch clocks, GMT timing for all machines) delivered through a single, standardized system.

Distributed DDS



Concept

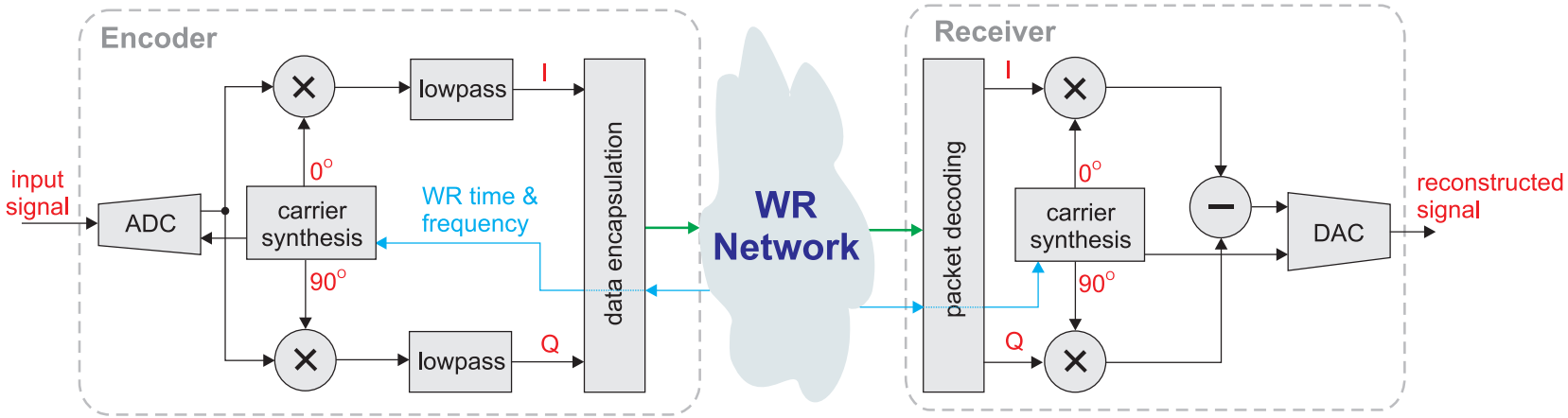
- DDS synthesizer in a PLL loop
- DDS setpoints broadcast to all receivers
- Receivers feed them to identical DDS chips

Distributed DDS

Pros and cons

- Little complexity (it's a simple PLL)
- Very small data bandwidth for stable signals
- Can't handle phase jumps
- No amplitude/phase modulation

Remote IQ modulator



The idea

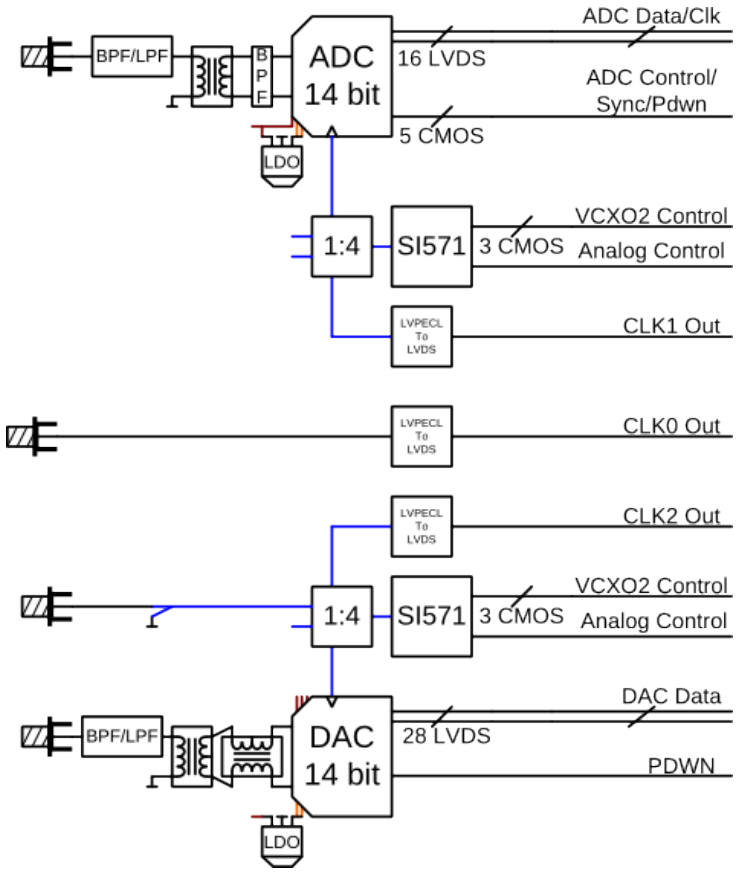
- Digitization and downconversion at the encoder
- Transmission of low-bandwidth IQ components
- Upconversion in the receiver

Remote IQ modulator

Pros and cons

- Works with any narrowband modulation
- Bandwidth limited only by network throughput
- "Mixed" solution possible: IQ demodulator at the encoder, VC(X)O at the receiver
  - No noisy DAC – may offer better phase noise for clock signals
- May need more network bandwidth
  - ... but we can do real-time lossless compression of the I/Q signals
- More complex hardware
  - ... but can use already existing ADC/DAC cards as long as they can be synchronized to WR

Test platform



A dedicated FMC:

- 125 MSPS 14-bit ADC and 600 MSPS 14-bit DAC with analog front-ends
- two independent Si571 programmable clocks (DDS + PLL combos)
- Extra clock outputs

Outline

Future plans

WR Switch

- Fully functional V3 prototype: end of 2011
- Commercial product: end of Q1 2012

WR Node

- HW platform already available
- Final specification ready by end of October

RF over Ethernet

- Test system availability: Jan 2012

Open Hardware

Publish everything needed to review

Specifications, discussions, schematics and layouts in some human-readable format, HDL, etc. Publish universally, no NDAs.

Publish everything needed to modify

Schematics and PCB layout files for your favorite EDA tool.

Publish everything needed to produce

Manufacturing files, bill of materials, etc.

Open Hardware Repository: http://www.ohwr.org

A very useful tool

A web-based collaborative tool for electronics designers.

Made itself of open software

- Redmine for wiki and task/issue management.
- Sympa mailing list manager.
- SVN/GIT for version management (integrated in Redmine).

Other possible uses

- Traceability for Technology Transfer departments.
- Prove prior art with UTC time stamps in SVN, GIT, wiki...

Summary

- A data link fulfilling all our needs in **synchronization** and **determinism**.
- Fully based on **standards** like Synchronous Ethernet and PTP.
- Fully **open** design.
- A successful **collaboration** including institutes and companies.