Timing for Accelerators

Javier Serrano

European Organisation for Nuclear Research
(CERN)

CERN Accelerator School

30 June 2021

Goals of this lecture

In scope

- A quick introduction to event-based systems
- How to distribute a common notion of time to many nodes
- Usual timing performance specification methods
- Existing technologies for different performance goals

Out of scope

- A detailed survey of all deployed solutions
- How to use event systems to sequence accelerator operation

Outline

1. Introduction
2. Event systems
3. Timing concepts
   - Background on phase noise
   - Background on phase-locked loops
4. Timing technologies
   - Millisecond timing
   - Microsecond timing
   - Nanosecond and picosecond timing
   - Femtosecond timing

General background

Why timing systems

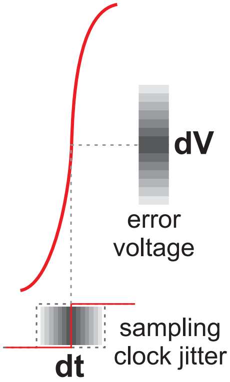
- Having many systems act in sync.
- Providing a common notion of time to make sense of distributed diagnostics data.

Challenges

- Generating a very good (periodic) clock signal at the source.
- Evaluating transmission delay from that source to each destination so we can account for it.

An example application

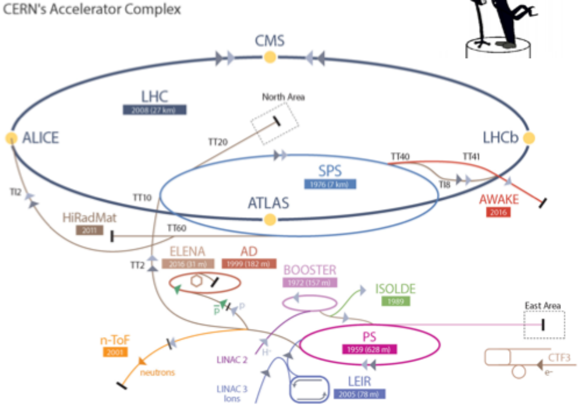
Clocking an ADC from a recovered clock signal in a timing receiver



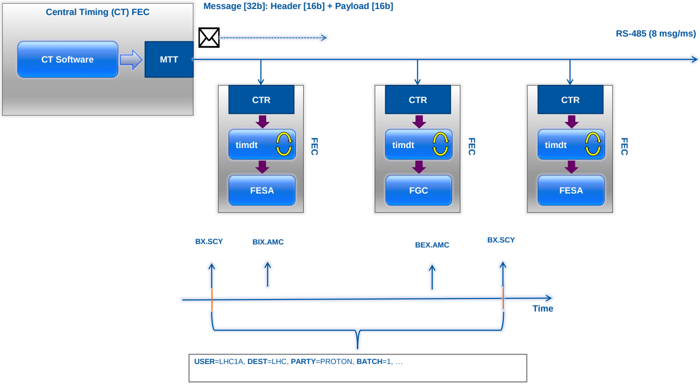
Clock jitter becomes amplitude noise in the sampled signal, with a conversion factor depending on signal slope.

Outline

Sequencing and synchronisation

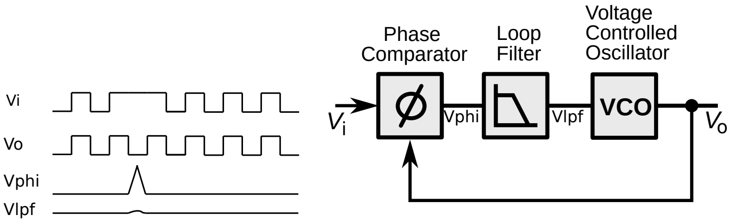


General Machine Timing lower layers



Courtesy Greg Kruk

Phase Locked Loops (PLL) for Clock & Data Recovery (CDR)



Outline

Table of Contents

The imperfect sine wave

With both amplitude and phase noise

$$a(t) = A(1 + \alpha(t))\sin(\omega t + \varphi(t))$$

If we use hard-limiters, AGCs, etc.

$$a(t) = A\sin\left(\omega\left(t + \frac{\varphi(t)}{\omega}\right)\right)$$

Phase noise Power Spectral Density (PSD)

Parseval's theorem

$$\int_{-\infty}^{+\infty} |\varphi(t)|^2\, dt = \int_{-\infty}^{+\infty} |\Phi(f)|^2\, df$$

Truncated signal

$$\Phi_T(f) = \int_{-T/2}^{+T/2} \varphi_T(t) e^{-j2\pi ft} dt$$

Truncated Parseval

$$\frac{1}{T} \int_{-T/2}^{+T/2} |\varphi_T(t)|^2\, dt = \int_{-\infty}^{+\infty} \frac{|\Phi_T(f)|^2}{T} df$$

Phase noise Power Spectral Density (PSD)

Wiener-Khintchine theorem

$$S^{II}_{\varphi}(f) = \lim_{T\to\infty} \frac{1}{T} \left|\Phi_T(f)\right|^2$$

In practice

$$S_{\varphi}(f) \approx \frac{2}{T} \left\langle \left|\Phi_T(f)\right|^2 \right\rangle_m$$

Integrating PSD: jitter

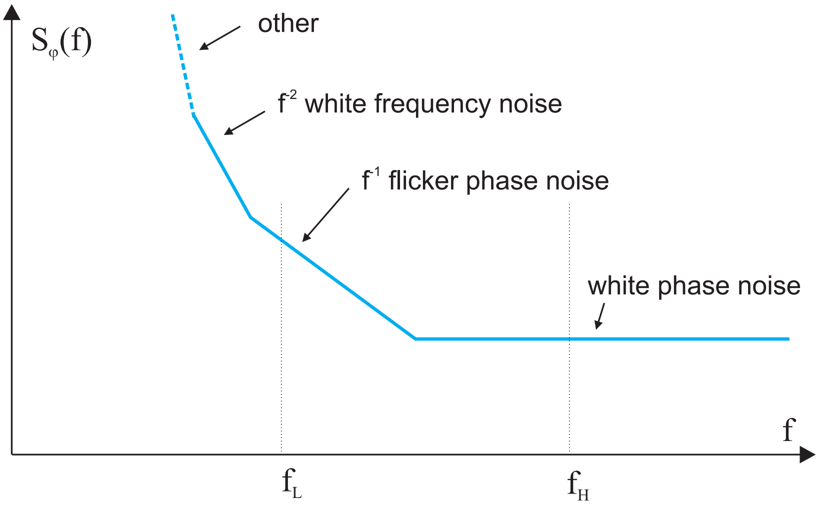
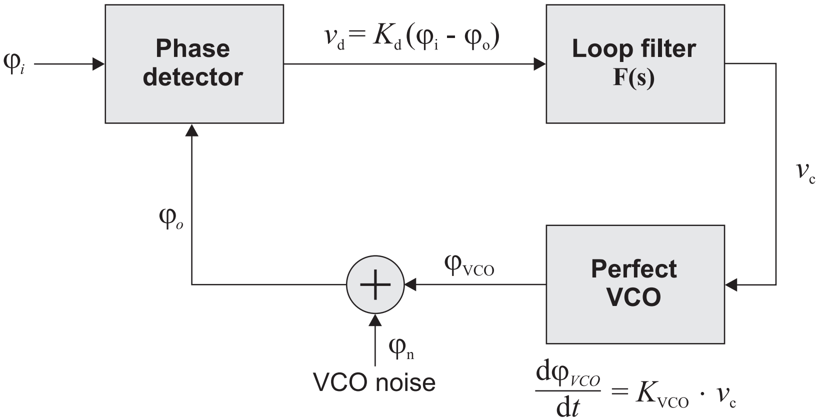


Table of Contents

PLL block diagram



PLL transfer functions

Total output phase spectrum

$$\Phi_o(s) = H(s) \cdot \Phi_i(s) + E(s) \cdot \Phi_n(s)$$

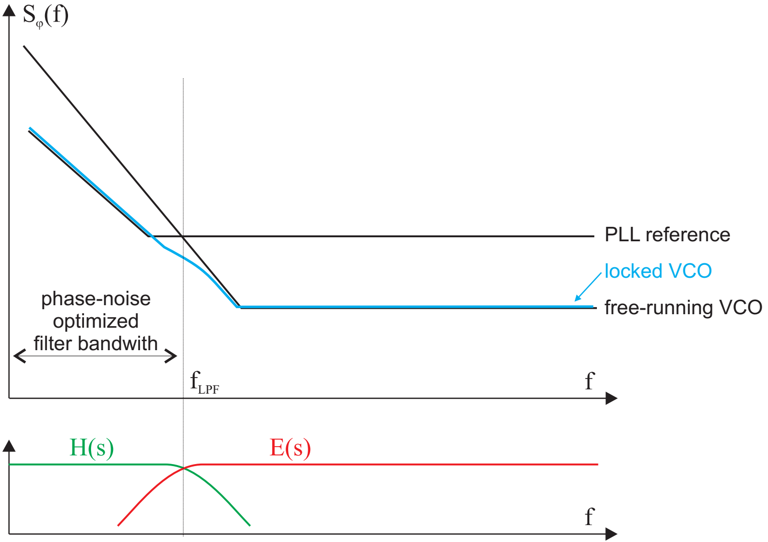
System transfer function (low pass)

$$H(s) = \frac{K_{VCO}K_dF(s)}{s+K_{VCO}K_dF(s)}$$

Error transfer function (high pass)

$$E(s) = 1 - H(s) = \frac{s}{s+K_{VCO}K_dF(s)}$$

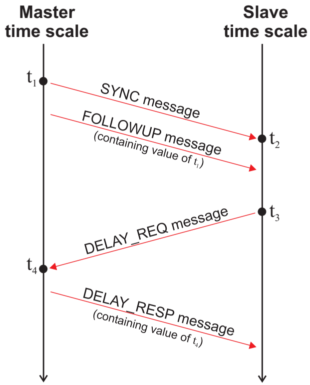
Jitter optimisation



Outline

1. Introduction
2. Event systems
3. Timing concepts
   - Background on phase noise
   - Background on phase-locked loops
4. Timing technologies
   - Millisecond timing
   - Microsecond timing
   - Nanosecond and picosecond timing
   - Femtosecond timing

Two-way delay compensation schemes



Having the values of t_1, t_2, t_3 and t_4, the slave can calculate the one-way link delay:

$$\delta_{ms} = \frac{(t_4 - t_1) - (t_3 - t_2)}{2}$$

Table of Contents

Millisecond timing

Example: Network Time Protocol (NTP)

Used in general-purpose computers

- Works across the Internet.
- Each client (slave) gets synchronised to one or more servers.

Cannot do better than 1 ms

- Asymmetries in network, switches and routers.
- Non-determinism due to OS scheduler (time tags done in SW).
- Requires strong statistics artillery to average over many measurements.

Table of Contents

Microsecond timing

Example: Precision Time Protocol (PTP, IEEE1588)

Acts on both of NTP's shortcomings

- Time-tagging can be done in HW.
- Special PTP switches ensure no loss in precision.

Has a hard time doing better than $1\mu s$

- Typical nodes use a free-running oscillator.
- Frequency offset (and drift) compensation generates extra traffic.

Table of Contents

White Rabbit technology - sub-ns synchronisation

Based on

- IEEE 1588 Precision Time Protocol on Gigabit Ethernet over fibre

White Rabbit technology - sub-ns synchronisation

Based on

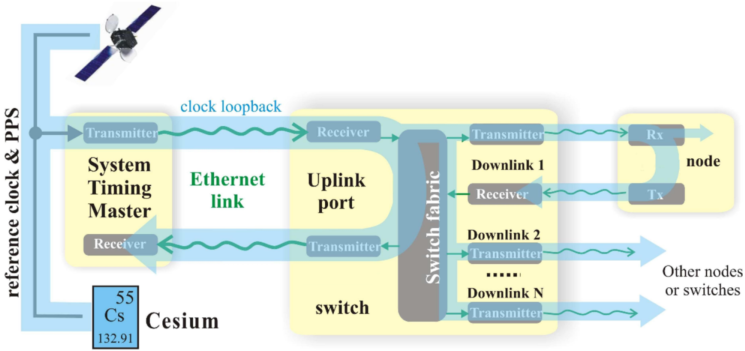
- IEEE 1588 Precision Time Protocol
  on Gigabit Ethernet over fibre

Enhanced with

- Layer 1 syntonisation
- Digital Dual Mixer Time Difference (DDMTD)
- Link delay model

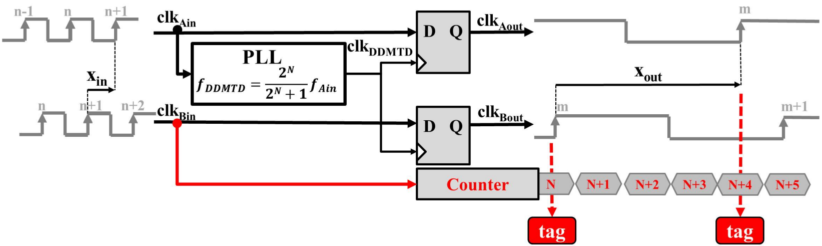
Layer 1 Syntonisation

- Clock is encoded in the Ethernet carrier and recovered by the receiver chip
- All network devices use the same physical layer clock
- Clock loopback allows phase detection to enhance precision of timestamps



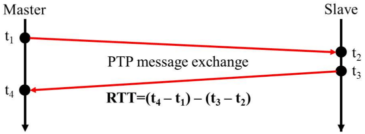
Digital Dual Mixer Time Difference (DDMTD)

- Precise phase measurements in FPGA
- WR parameters:
  - $clk_{in} = 62.5$ MHz
  - $clk_{DDMTD} = 62.496185$ MHz (N=14)
  - $clk_{out} = 3.814$ kHz
- Theoretical resolution of 0.977 ps



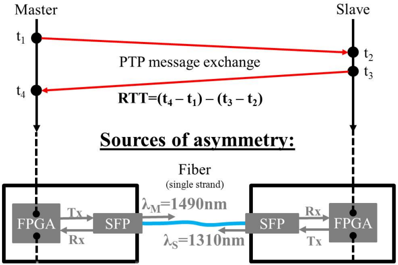
Link delay model

- Correction of Round Trip Time (RTT) for asymmetries



Link delay model

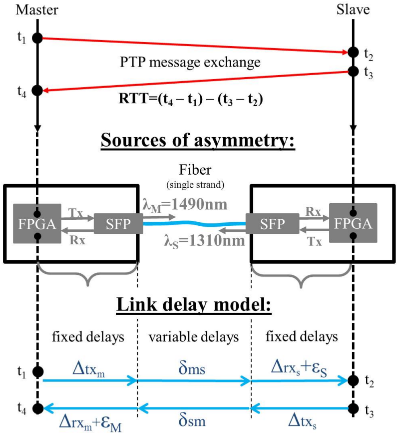
- Correction of Round Trip Time (RTT) for asymmetries
- Asymmetry sources: FPGA, PCB, SFP electrics/optics, chromatic dispersion



Link delay model

- Correction of Round Trip Time (RTT) for asymmetries
- Asymmetry sources: FPGA, PCB, SFP electrics/optics, chromatic dispersion
- Link delay model:
  - **Fixed delays** – calibrated/measured
  - **Variable delays** – evaluated online with:

$$\alpha = \frac{\nu_g(\lambda_s)}{\nu_g(\lambda_m)} - 1 = \frac{\delta_{ms}-\delta_{sm}}{\delta_{sm}}$$



Link delay model

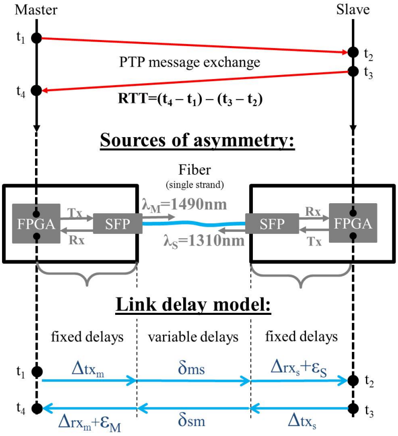
- Correction of Round Trip Time (RTT) for asymmetries
- Asymmetry sources: FPGA, PCB, SFP electrics/optics, chromatic dispersion
- Link delay model:
  - **Fixed delays** – calibrated/measured
  - **Variable delays** – evaluated online with:

$$\alpha = \frac{\nu_g(\lambda_s)}{\nu_g(\lambda_m)} - 1 = \frac{\delta_{ms} - \delta_{sm}}{\delta_{sm}}$$

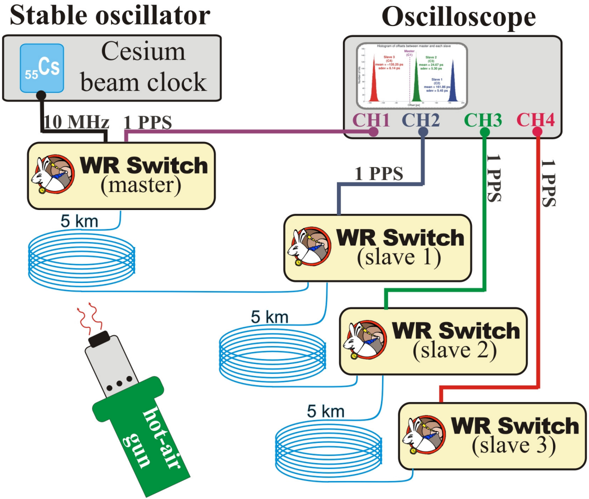
- Accurate offset from master (OFM):

$$\delta_{ms} = \frac{1+\alpha}{2+\alpha}\left(RTT - \sum\Delta - \sum\epsilon\right)$$

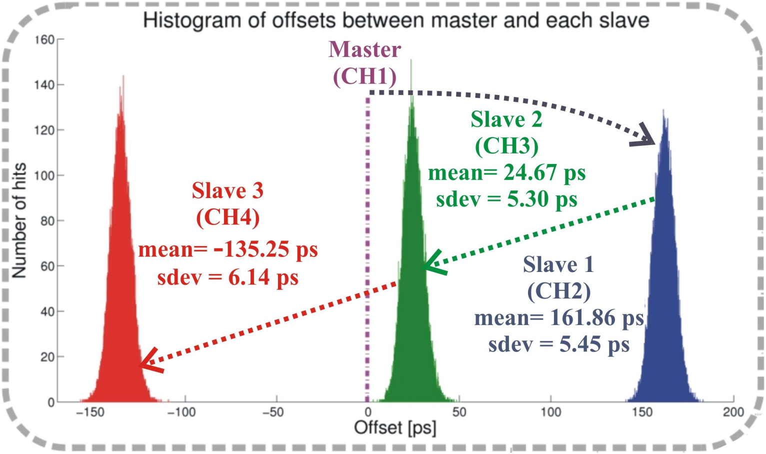
$$OFM = t_2 - (t_1 + \delta_{ms} + \Delta_{txm} + \Delta_{rxs} + \epsilon_S)$$



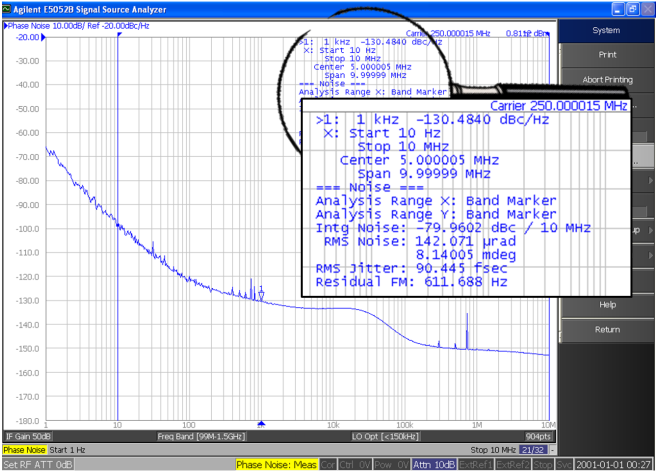
WR time transfer: out-of-the-box



WR time transfer: out-of-the-box



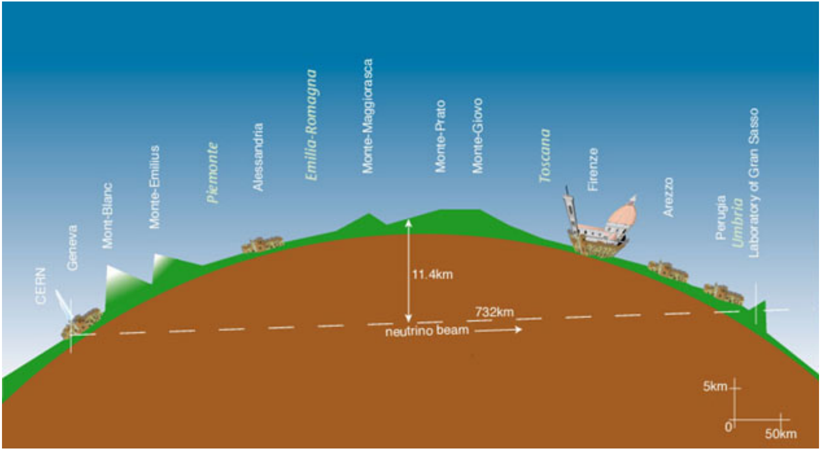
WR time & frequency transfer: state of the art



- **GM-out to end-node-out**: accuracy of <**10 ps**
- **GM-out to end-node-out**: jitter of <**100 fs** RMS 10 Hz–10 MHz

Nanosecond and picosecond timing

Another example: neutrino oscillation experiments



Nanosecond and picosecond timing

Another example: neutrino oscillation experiments

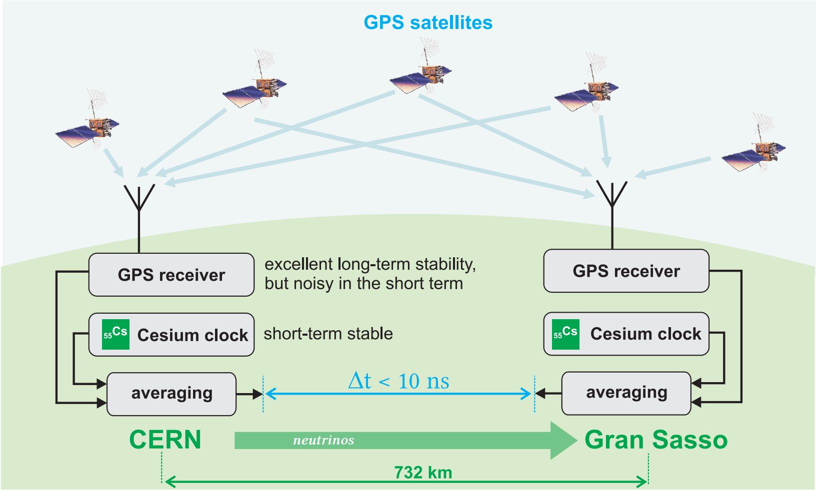
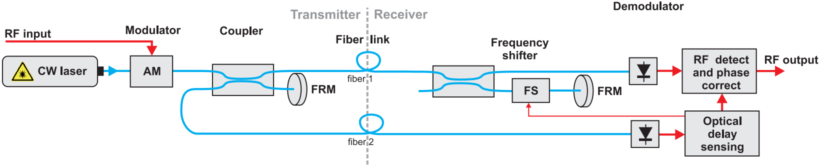


Table of Contents

Femtosecond timing

Example: Continuous Wave (CW) system



Parting words

Specify well

Jitter (with PSD integration limits), UTC vs. beam-synchronous, automatic delay compensation...

Choose well

Going from milliseconds to femtoseconds has costs (money, complexity, reliability. . .). Pick the technology which suits your needs best.