



# Timing and Synchronisation in HEP Experiments



Jeroen Hegeman



EURIZON Detector School – July 2023, Wuppertal

# Why me?

My name is **Jeroen Hegeman**, and I have been working on DAQ, timing, data-flow, etc. projects for HEP (on and off) for several years

- ▶ My first: basic DAQ and GPS software for HiSPARC detectors for cosmic particles built with and for high schools in NL (<https://www.hisparc.nl/en/>)
- ▶ My most recent (since 2009, in different roles): several generations of the timing and trigger control system at the CMS experiment (<https://cms.cern/>)

**For comments and/or questions I can be reached at: [jeroen.hegeman@cern.ch](mailto:jeroen.hegeman@cern.ch)**



# My sources deserve as much credit as my presentation

These slides contain information from many sources.

**Credit goes to the creators, and any mistakes you may find were introduced by me.**

Images used from external sources are accompanied by links to their sources. None of these images were changed, apart from possibly cropping for relevance and/or size/aspect ratio constraints.

Images, examples, and references were chosen for their usefulness for this presentation, and do not imply support/endorsement.

# Heads up! We will have to take a few shortcuts



PNGEGG

- ▶ We will look at **what timing and synchronisation is all about** (in HEP experiments).
- ▶ We will focus on **LHC collider experiments** (and be biased towards CMS).
- ▶ The first part is mostly example-based, the second part is a bit more detailed/technical
- ▶ We will go over some topics quite quickly, and from quite a distance. Various references have been added as starting point for further reading.

# Overview

- ▶ Timing – What is it and why does it matter?
- ▶ Synchronisation at HEP experiments
- ▶ Caffeine break ☕
- ▶ Clock generation, -distribution, -recovery, -quality

Timing – What is it and why does it matter?

# Base layer terminology

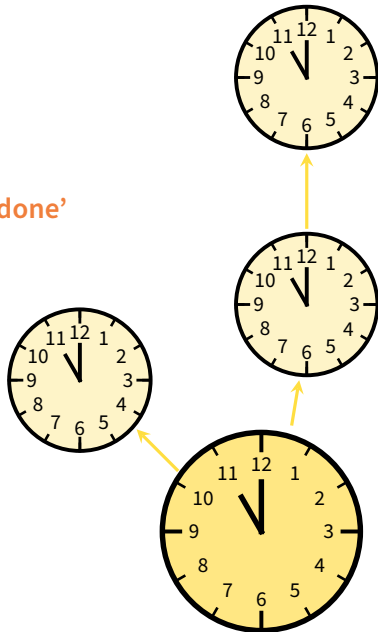
For our purposes:

- ▶ Time: **'The thing that is measured with clocks'**
- ▶ Timing: **'The control of when something should be done'**

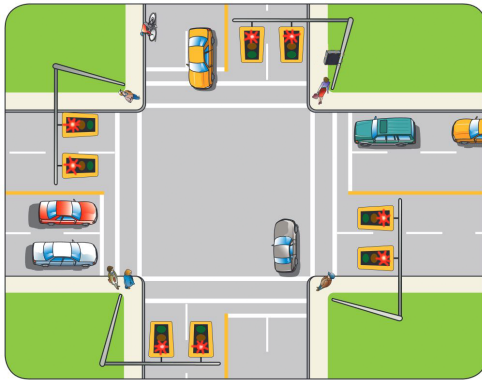
The above boils down to:

- ▶ Syntonisation: **Tuning to the same frequency**
- ▶ Synchronisation: **Aligning to indicate the same time/counter values**

**The first task is to identify the reference**



# Relative timing is part of everyday life



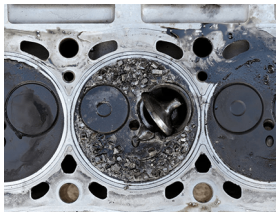
TranBC

In many cases one only appreciates the synchronisation when it goes missing

# Proper relative timing is required for many things to work

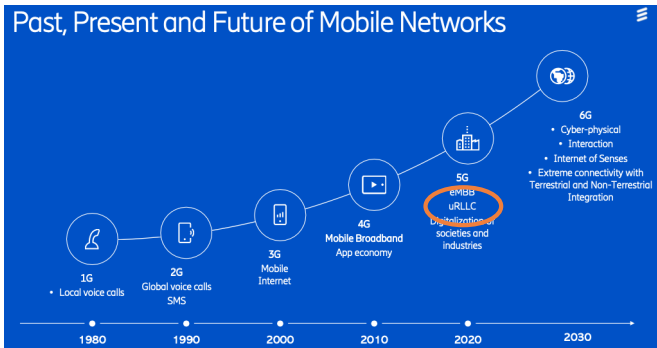
When properly adjusted, and with correct timing, your car engine will run smoothly and efficiently.

As soon as the timing (belt) breaks down, your engine quickly becomes an expensive mess.



Capitol Toyota

# Slightly off-topic, but what a hot topic: latency



M. Tatipamula, Ericsson, from the Cadence Breakfast Bytes blog

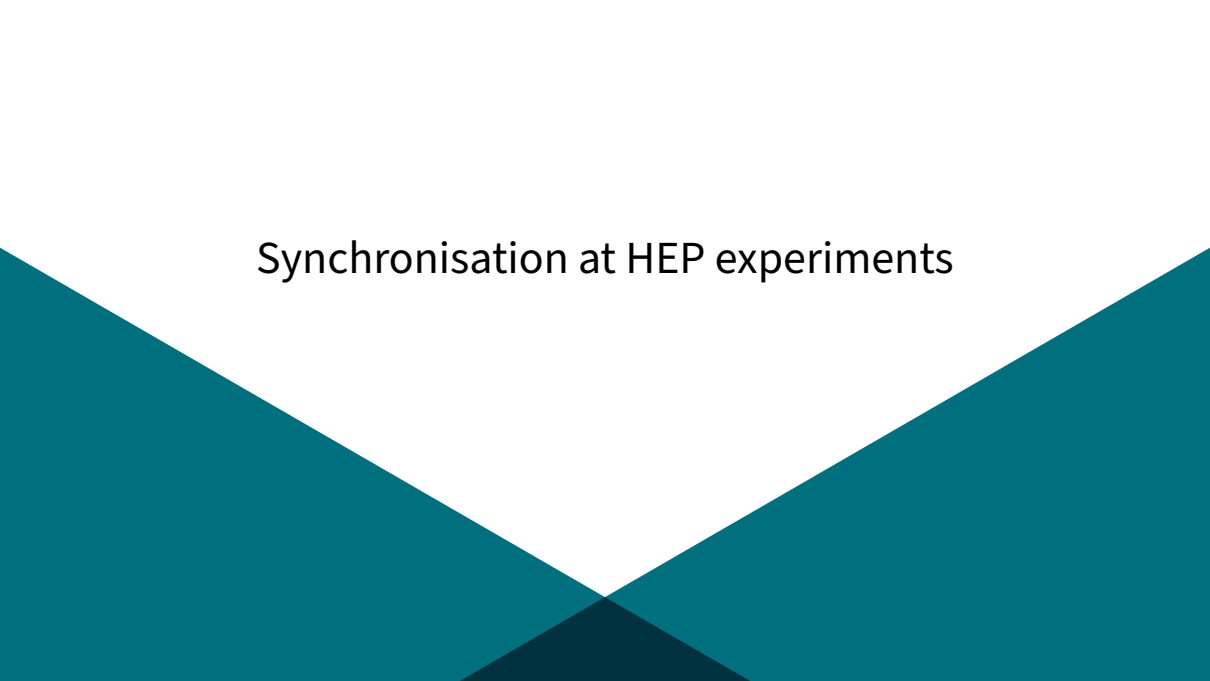
URLLC:  
Ultra Reliable  
and Low Latency  
Communications

Speed/bandwidth is about getting enough data

**Latency** is about getting the **right data at the right time**



# Synchronisation at HEP experiments

The background of the slide features a white upper section and a teal lower section. The teal section is composed of two large triangular shapes that meet at a point at the bottom center, creating a V-shape. The teal color is a dark, muted blue-green.

‘Timing’ in HEP experiments covers many aspects

Timing

# ‘Timing’ in HEP experiments covers many aspects

**Syntonisation and synchronisation**

# 'Timing' in HEP experiments covers many aspects

## Syntonisation and synchronisation

### Syntonisation/Clocking

- ▶ Clock generation
- ▶ Clock distribution

### Dance to the same beat

# 'Timing' in HEP experiments covers many aspects

## Syntonisation and synchronisation

### Syntonisation/Clocking

- ▶ Clock generation
- ▶ Clock distribution

**Dance to the same beat**

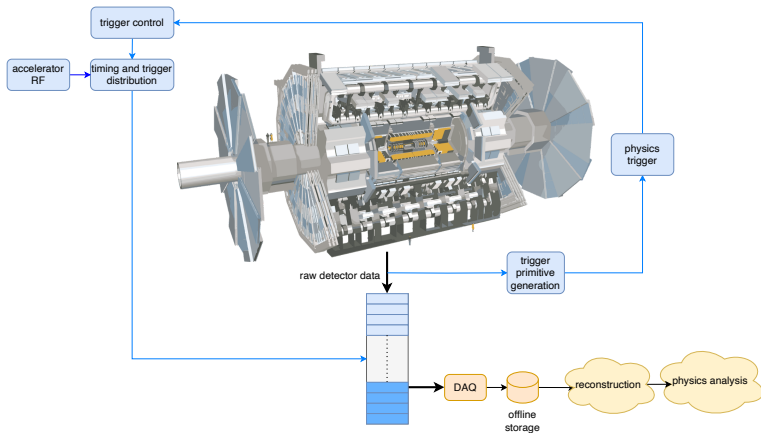
### Synchronisation

- ▶ Deskewing/aligning
- ▶ 'Timing-in'

**Avoid stepping on each other's feet**

# Our context: the HEP experiment trigger-DAQ loop

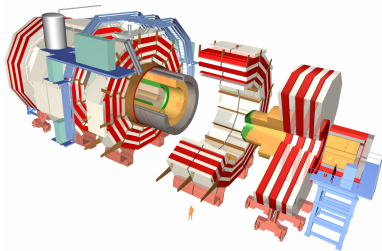
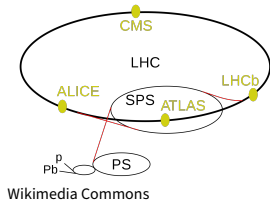
- ▶ **Beam-synchronous**
- ▶ Loop has to close before data falls from the buffers, i.e., within the **trigger latency**
- ▶ Largely dominated by delays in cables and interconnects
- ▶ **Ideally dominated by physics algorithm processing time**



# HEP timing ties together accelerator(s) and experiment(s)

## Accelerator complex

- ▶ Injectors to accelerators
- ▶ Injection and dump kicker magnets
- ▶ Accelerator to experiments

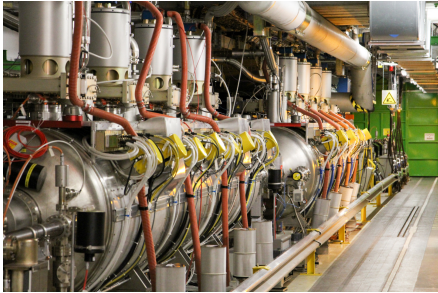


CMS experiment

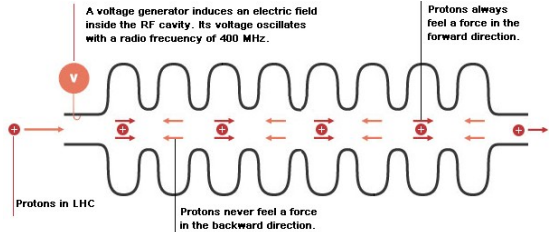
## Experiments/detectors

- ▶ Timing for trigger and DAQ
- ▶ Front-end sampling clocks and physics timestamping: 'precision timing'

# At the LHC, the RF drives the show



CERN CDS

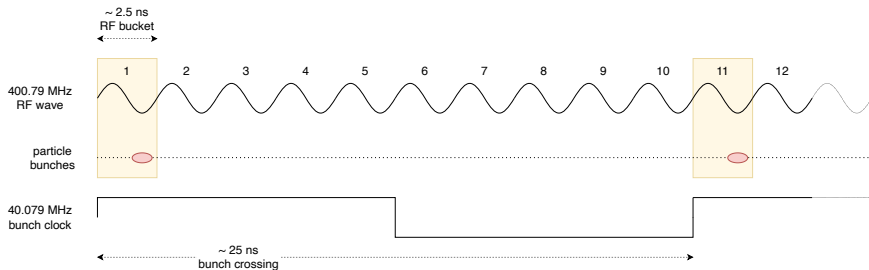


[https://www.lhc-closer.es/taking\\_a\\_closer\\_look\\_at\\_lhc/0.rf\\_cavities](https://www.lhc-closer.es/taking_a_closer_look_at_lhc/0.rf_cavities)

- ▶ Dipole magnets keep the particle bunches on their orbit
- ▶ Radio-frequency cavities inject electromagnetic energy to accelerate the particles
- ▶ **The RF defines the bunch structure**

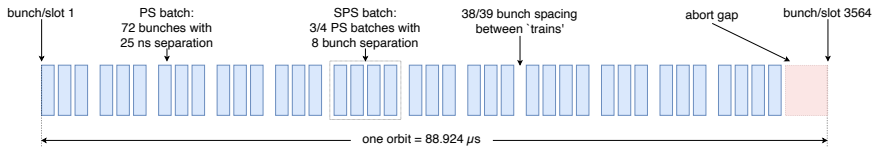


# Particles, bunched into buckets, form the LHC beams



- ▶ The RF constrains the particle bunches into RF 'buckets'. Minimal bunch spacing is (agreed to be) ten buckets.
  - ▶ The LHC RF operates at 400.79 MHz
- ▶ Experiments synchronise to the passage of particle bunches in their detectors
  - ▶ The LHC bunch clock is 40.079 MHz

# The filling pattern then determines who sees collisions



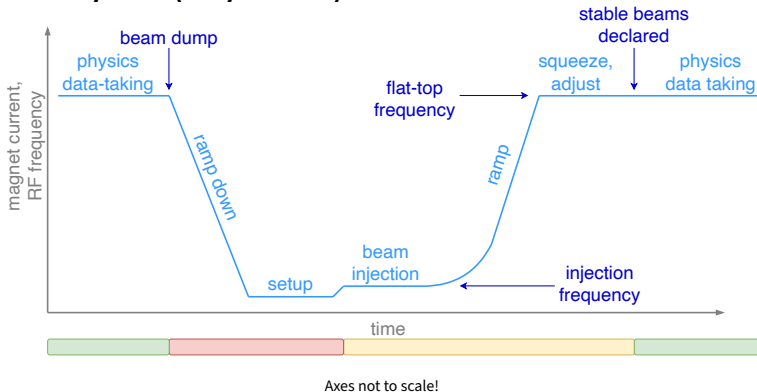
- ▶ 'Trains' of bunches are carefully arranged to fill one 'orbit' (or 'turn') of the LHC
  - ▶ Per-beam patterns used to fine-tune number of colliding bunches in each experiment
  - ▶ Each pattern has to respect the timing requirements of all injection/extraction, and dump kicker magnets
- ▶ Experiments need to know:
  - ▶ When two bunches collide (bunch clock)
  - ▶ Which two bunches collide (orbit signal)
- ▶ Beam instrumentation: similar, but down to the RF bucket

# Intermezzo: bunch-crossing spacing over the years

- ▶ ISR (1971 – 1984): unbunched, coasting beams
- ▶ LEP (1989 – 2000):  $4 \times 4 = 16$  bunches,  $22 \mu\text{s}$  bunch spacing
- ▶ Tevatron in Run-1 (1992 – 1996): 6 bunches, spaced by  $3.5 \mu\text{s}$
- ▶ Tevatron in Run-2 (2001 – 2011):  $3 \times 12 = 36$  bunches, spaced by  $396 \text{ ns}$
- ▶ LHC (2009 – now): 3564 bunch slots,  $25 \text{ ns}$ ,  $40 \text{ MHz}$

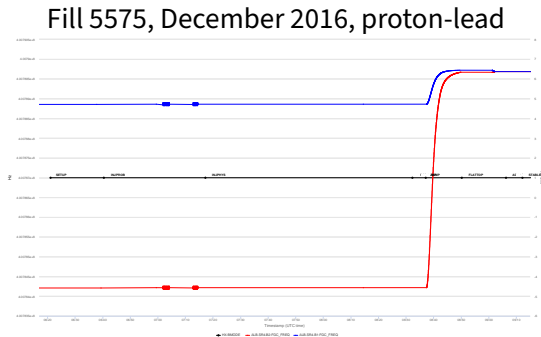
Note: bunch clock frequency  $\neq$  bunch crossing rate  $\neq$  collision rate, although the differences can be subtle

# The LHC fill cycle (stylised)



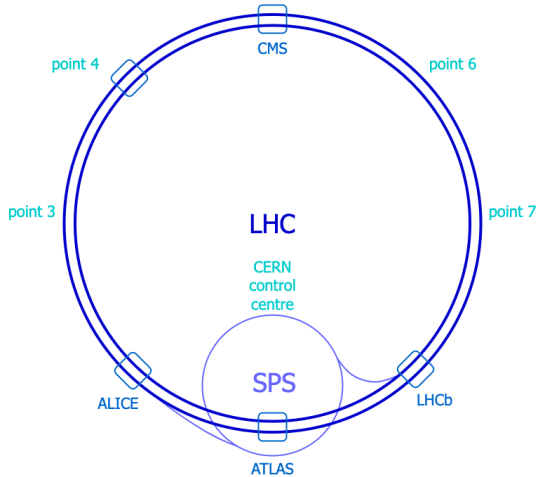
- **During physics data taking the experiments must be locked to the beams**
- Following injection, ramp, etc. is interesting for beam background and luminosity measurements
- In ramp down and setup the RF is actively manipulated. Don't try to follow the RF during these periods.

# The RF frequency change during acceleration is minute

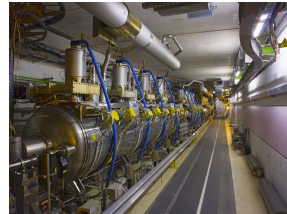


- ▶ Protons:  $f_{\text{injection}} \approx 400.788\,860\text{ MHz}$ ,  $f_{\text{flat-top}} \approx 400.789\,685\text{ MHz} \rightarrow \Delta_f \approx 825\text{ Hz}$
- ▶ Lead:  $f_{\text{injection}} \approx 400.784\,218\text{ MHz}$ ,  $f_{\text{flat-top}} \approx 400.789\,685\text{ MHz} \rightarrow \Delta_f \approx 5.45\text{ kHz}$

# The LHC bunch clock originates at the LHC RF at P4

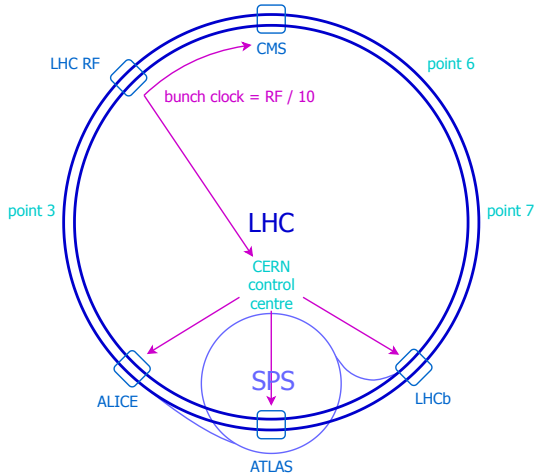


CERN CDS

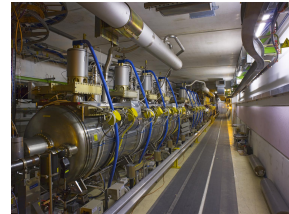


CERN CDS

# The LHC bunch clock originates at the LHC RF at P4



CERN CDS



CERN CDS

# Timing information available to LHC experiments

## Bunch clock and orbit signals

- ▶ One 40.079 MHz bunch clock for each of the two beams
- ▶ One orbit (or turn) signal for each of the two beams (11.246 kHz)

## Beam-Synchronous Timing (BST)

Beam-synchronous acquisition triggers, injection/dump triggers, etc.

Two independent signals, one for each of the beams, updated once per orbit

## General Machine Timing (GMT)

One message network to synchronise to the CERN accelerator complex

---

*An FPGA Based Multiprocessing CPU for Beam Synchronous Timing in CERN's SPS and LHC, CERN-AB-2003-112-CO*

*Nanosecond Level UTC Timing Generation and Stamping in CERN's LHC, CERN-AB-2003-111-CO*



# LHC bunch clock and orbit signals

- ▶ Clocks and orbits are distributed as 'pure, analogue' signals
  - ▶ Receiver applies threshold to 're-edge' the signal
- ▶ Extremely stable
- ▶ Very low phase noise at high frequencies (jitter RMS  $\ll 1$  ns)
- ▶ Large seasonal drift/wander ( $O(10$  ns))
  - ▶ Experiments monitor and adjust the phase of the received clock

# Beam-Synchronous Timing

- Synchronises actions and measurements between the SPS, the LHC, and their experiments
- Contains a UTC timestamp plus telegrams from the GMT
- Synchronous to the SPS and LHC beams
- Low-jitter
- Information updated turn-by-turn

Byte	Description	Beam	Data format	Update Rate	Info				
0	GPS Absolute Time	1 OR 2	32 bits : Number of microseconds since last second	Turn	Updated by BST Master Every Turn				
1									
2									
3									
4									
5									
6									
7									
8 to 16	BI Specific Bytes								
17	BST Master Status Register	1 OR 2	Bit Enumeration	Turn	Sent by BST Config				
18	Turn Count Number	1 OR 2	0 ... 4294967294 (106 hours - reset on first injection)	Turn	Updated by BST Master Every Turn				
19									
20									
21	LHC Fill Number	1 = 2	32 bits : Integer	1 Hz	Sent by BST Config Data from LHC telegram.				
22									
23									
24									
25	Beam Mode	1 = 2	Enumerated type	On Change (latency 1s)	Sent by BST Config Data from LHC telegram.				
26									
27	Particle Type	1	Enumerated type	On Change	Sent by BST Config				
28	Particle Type	2	Enumerated type	On Change	Sent by BST Config				
29	Beam Momentum	1 = 2	2 bytes in GEV/c	1 Hz	Sent by BST Config Data from LHC telegram.				
30									
31									
32	Total Intensity	1	Integer $\times 10E10$ charges from telegram.	1 Hz	Sent by BST Config Data from LHC telegram.				
33									
34									
35	Total Intensity	2	Integer $\times 10E10$ charges from telegram.	1 Hz	Sent by BST Config Data from LHC telegram.				
36									
37									
38	BI Specific Bytes								
39									
40 to 63	BI Specific Bytes								

From the BST specification

# General Machine Timing

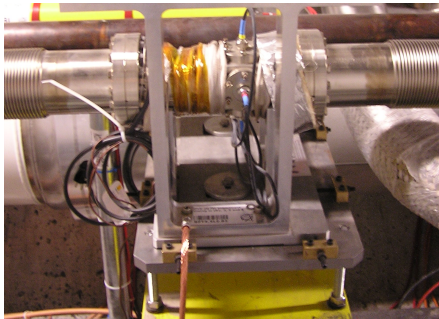
The LHC GMT design dates from around 2003, and was made backwards compatible with the existing GMT, to integrate the LHC into the accelerator complex.

- ▶ RS-485 multi-drop network @ 500 kbit/s
- ▶ The cabled network limits the timing message precision to a jitter of  $\approx 14$  ns
- ▶ A special hybrid PLL recovers a 40 MHz clock with a jitter of 1 ns
- ▶ The whole network is GPS-disciplined, and therefore locked to UTC  
→ timestamps formulated as 'UTC down to the second + # 25-ns ticks'
- ▶ A dedicated interpolation ASIC allows timestamping down to 25 ps

# The beams are always right

**Timing-in the experiments uses dedicated beam pick-ups, 175 m upstream in both beams**

- ▶ A pilot bunch injected into each beam allows experiments and accelerator to agree on which bunch crossing is the one called '1'
- ▶ 'Cogging' moves bunch X in both beams into the same bunch clock period
- ▶ Fine adjustment places the collisions in the middle of the detectors
- ▶ Experiments confirm with tracker-based 'beamspot' measurements

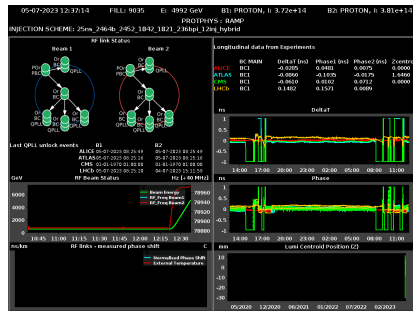


**The above procedures were all done manually at the start.** Nowadays, verification and monitoring is done continuously and automatically.

# The beams are always right

Timing-in the experiments uses dedicated beam pick-ups, 175 m upstream in both beams

- ▶ A pilot bunch injected into each beam allows experiments and accelerator to agree on which bunch crossing is the one called '1'
- ▶ 'Cogging' moves bunch X in both beams into the same bunch clock period
- ▶ Fine adjustment places the collisions in the middle of the detectors
- ▶ Experiments confirm with tracker-based 'beamspot' measurements

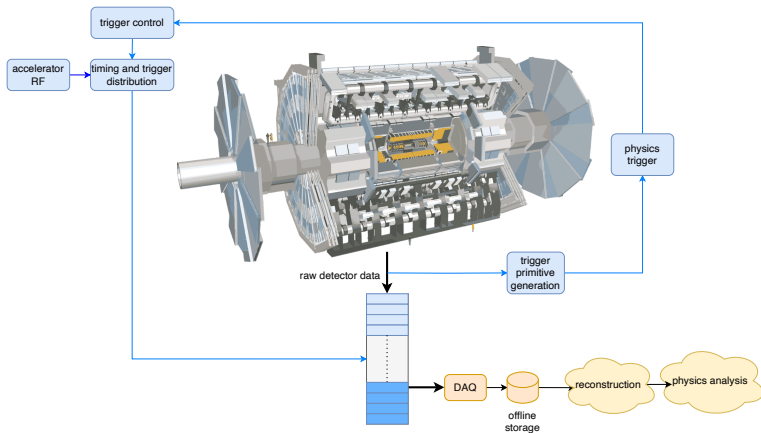


LHC Vistas

The above procedures were all done manually at the start. Nowadays, verification and monitoring is done continuously and automatically.

# Our context: the HEP experiment trigger-DAQ loop

- ▶ **Beam-synchronous**
- ▶ Loop has to close before data falls from the buffers, i.e., within the **trigger latency**
- ▶ Largely dominated by delays in cables and interconnects
- ▶ **Ideally dominated by physics algorithm processing time**



# The trigger-DAQ loop hinges on proper synchronisation

## Specific clock signals are used everywhere

- ▶ To sample physics data: Requires low jitter/noise, synchronicity with the physics (i.e., the beams)
  - ▶ Phase-adjusted in all subdetectors to optimise the detector signal
- ▶ To drive digital electronics: Requires permanent presence, stability
- ▶ To drive (high-speed) serial links: Requires low jitter/noise

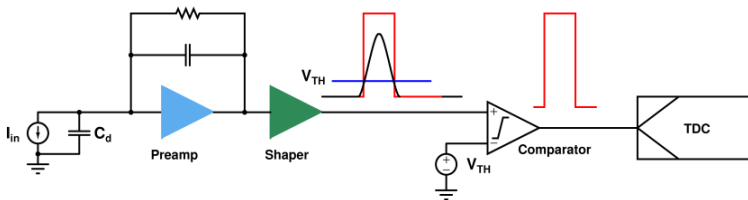
# The trigger-DAQ loop hinges on proper synchronisation

## Correct synchronisation is required throughout

- ▶ To timestamp/mark data: Requires the system is synchronised to the beams, to a reference time, or both
- ▶ Timestamps use orbit and BX numbers
  - ▶ To combine event fragments from all read-out units into a coherent event
  - ▶ Used to time-align the experiment to the bunch pattern
- ▶ Sub-bunch timestamping used for pileup suppression

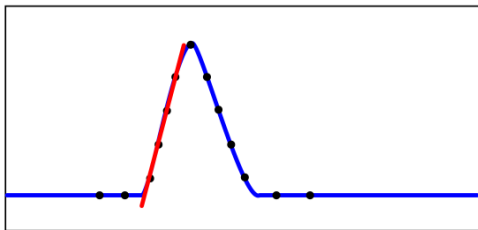


# The basics of single-sample detector front-ends



- ▶ The sensor signal is usually amplified and shaped
- ▶ A comparator generates a square pulse
- ▶ The threshold crossing time is captured and digitised by a time-to-digital converter
- ▶ The TDC measures the passing time of the pulse/particle
  - ▶ Using the **bunch clock** as trigger
  - ▶ Using a high-speed clock to quantify the elapsed time. Typically a **multiple of the bunch clock**.

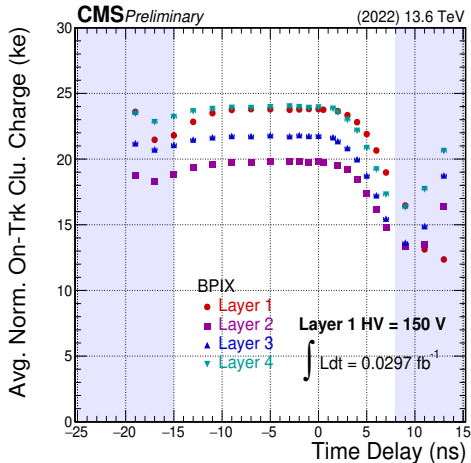
# The basics of multi-sample detector front-ends



- ▶ The sensor signal is usually amplified and shaped
- ▶ The full waveform is sampled and digitised at high speed (a multiple of the bunch clock) by an ADC
- ▶ Information on shape, amplitude, trigger time, etc., is extracted from the digitised waveform/samples using DSP algorithms
- ▶ **Critical to have sampling points correctly placed and spaced**

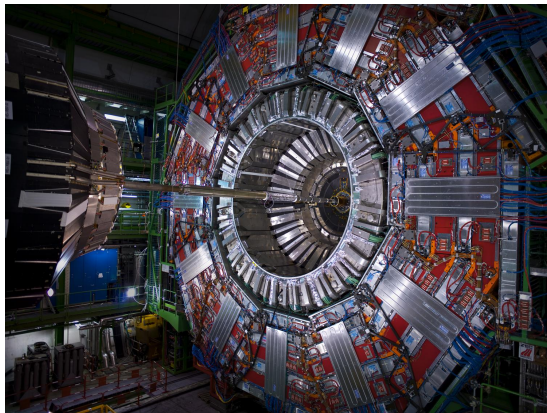
# Timing-in the CMS pixel detector

- ▶ The front-ends measure the charge generated by the passing of collision particles in 25 ns time slices
- ▶ A phase adjustment/clock delay aligns the sampling/read-out clock to the LHC bunch clock
- ▶ Adjustable in steps of 500 ps
  - ▶ Scan a full bunch clock period, while monitoring hit efficiency, cluster charge, cluster size, ...
  - ▶ Decide on 'the optimal value'



Status of the CMS pixel detector, DOI: 10.22323/1.420.0008

# The CMS clock and timing system, just as an example

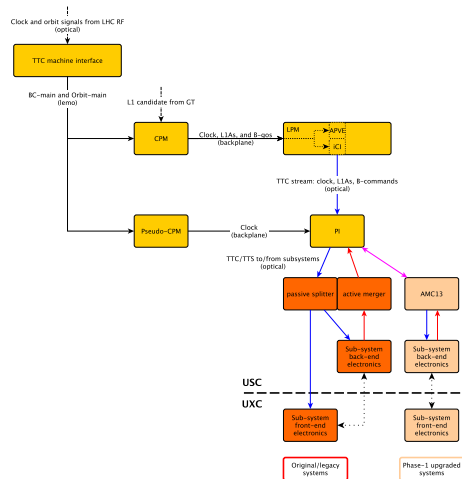


CMS image gallery

- ▶ Based on LHC-wide TTC technology and protocol  
**RD12: 'Timing, Trigger and Control systems for LHC detectors'**
- ▶ Upgraded in LS1 to accommodate the Phase-1 upgrades
- ▶ Serves both on-detector (front-end) and off-detector (back-end) end points
- ▶ Distributes clock, trigger, and sync commands

# CMS clock and timing distribution svstem

- ▶ Clocks and signals follow long paths to where they're needed
- ▶ Use optical transmission where possible
- ▶ Take care to keep clock and data aligned
- ▶ Careful whenever regenerating clocks (and avoid if possible)
- ▶ Off-detector end points can use FPGAs, on-detector end points require radiation-tolerant ASICs



# CMS timing signals: bunch clock and sync commands

- ▶ The LHC bunch clock
- ▶ An LHC turn signal, disguised as 'bunch counter reset', and a 'start of gap' marker
- ▶ Dedicated command sequences at start and stop of data-taking runs
- ▶ Various recovery command sequences to maintain data taking despite SEUs etc.
  - ▶ Resync, to recover from minor glitches (SEU, buffer overflow, pipeline misalignment, ...)
  - ▶ HardReset, to recover from more serious problems (that require front-end reprogramming, ...)
  - ▶ Issued periodically, or watchdog-driven
- ▶ Preemptive (semi-)periodic resets of various system components to mitigate SEU and other effects

# CMS timing signals: bunch clock and sync commands

- ▶ The LHC bunch clock
- ▶ An LHC turn signal, disguised as ‘bunch counter reset’, and a ‘start of gap’ marker
- ▶ Dedicated command sequences at start and stop of data-taking runs
- ▶ Various recovery command sequences to maintain data taking despite SEUs etc.
  - ▶ Resync, to recover from minor glitches (SEU, buffer overflow, pipeline misalignment, ...)
  - ▶ HardReset, to recover from more serious problems (that require front-end reprogramming, ...)
  - ▶ Issued periodically, or watchdog-driven
- ▶ Preemptive (semi-)periodic resets of various system components to mitigate SEU and other effects

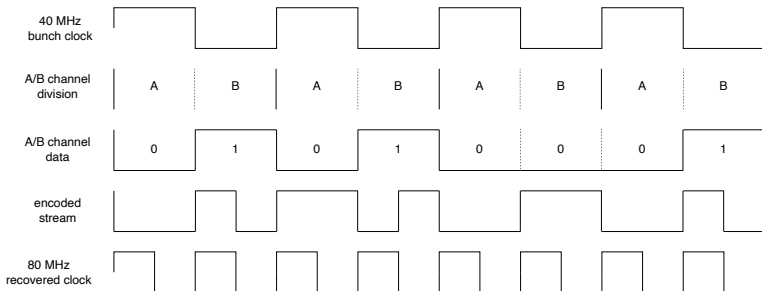
**The central timing system is exposed to the details of all detector systems. Even (mis)behaviour that can be handled/hidden in the back-ends may benefit from central support.**

# TTC signals

- ▶ Self-synchronous and beam-synchronous, at  $4 \times$  the LHC bunch clock frequency
- ▶ Time division multiplexed. For every clock tick information is sent on two channels:
  - ▶ A: single-bit trigger information (fixed, low latency)
  - ▶ B: frame-based timing information (idle high, start bit low)
- ▶ Use of 'bi-phase mark encoding' guarantees enough level transitions to recover a clock signal
  - ▶ logical 0: encoded as 'no transition'
  - ▶ logical 1: encoded as 'transition'



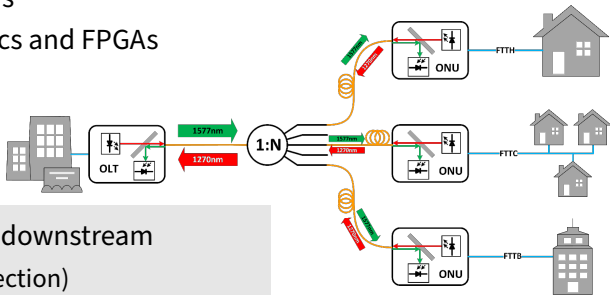
# Bi-phase mark encoding explained



- ▶ Encode '0' as 'no transition' and '1' as 'transition'
  - ▶ Encoding provides DC balance and enough edges for clock reconstruction
- ▶ Phase relationship between input and recovered clocks lost
- ▶ Heuristic (i.e., a trick) needed to distinguish A and B channels, and recover bunch clock phase → Rely on channel A being mostly zeroes.

# TTC-PON – Fibre-to-the-home for your experiment

- ▶ PON: Passive Optical Network, inspired by fibre-to-the-home networks
- ▶ Point-to-multipoint, self-synchronous
- ▶ Implemented using commercial optics and FPGAs
- ▶ Operates using custom protocol(s)

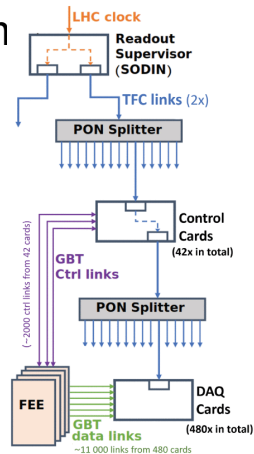


- ▶ Strong point: single fibre for up- and downstream
  - ▶ Two wavelengths (one for each direction)
- ▶ Disadvantage: up != down
  - ▶ Downstream: high bandwidth: 9.6 Gbit/s
  - ▶ Upstream: shared bandwidth (round robin) at 2.4 Gbit/s

Adopted for the ALICE and LHCb Phase-1 upgrades

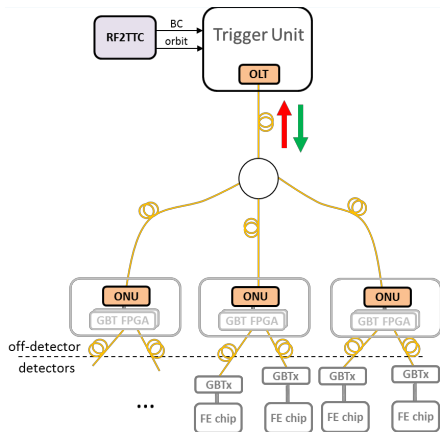
# PON at work – The new LHCb timing system

- ▶ Built on the TTC-PON system
  - ▶ DAQ and control cards implemented on a custom 'PCIe40' board
  - ▶ Clock recovery (in TTC-PON and in GBT receiver) uses a frame header to phase-align to the bunch clock
  - ▶ Measured maximum skew between distributed bunch clock and recovered clock on DAQ cards  $\ll 500$  ps
  - ▶ Random jitter  $< 80$  ps
- ▶ Measured maximum skew between distributed bunch clock and recovered clock on DAQ cards  $\ll 500$  ps
  - ▶ Random jitter  $< 80$  ps



From the below paper

# TTC-PON in ALICE

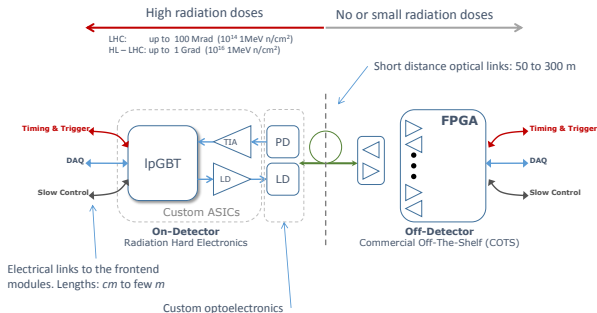


OLT: Optical Line Terminal / ONU: Optical Network Unit  
TTC-PON in ALICE, ACES 2020

- ▶ Benefits from the flexible topology to reduce the number of intermediate boards
- ▶ Heavy downstream traffic:
  - ▶ Trigger type: 32 bits per tick
  - ▶ Event ID: 44 bits (12 BCID + 32 ORBID) per tick
  - ▶ Heartbeat: 1 bit per orbit
- ▶ Relatively light upstream traffic:
  - ▶ Heartbeat acknowledge
  - ▶ Detector readiness flags and buffer status

# The next link generations combine timing and data

- ▶ The Versatile Link(+): radiation-tolerant optical link systems for the LHC and HL-LHC upgrades
  - ▶ On-detector: custom ASICs and optoelectronics
  - ▶ Off-detector: qualified commercial optoelectronics

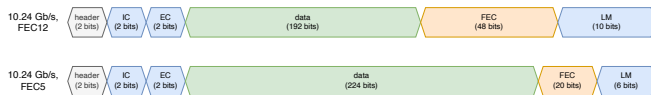


*The IpGBT: a radiation tolerant ASIC for Data, Timing, Trigger and Control Applications in HL-LHC, TWEPP 2019*

*The VTRx+, an Optical Link Module for Data Transmission at HL-LHC, DOI: 10.22323/1.313.0048*

# The next link generations combine timing and data

- ▶ Downstream combines clock, timing, trigger, slow control
  - ▶ Reduces the number of fibres and on-detector ASICs
- ▶ High-bandwidth upstream transfers physics data plus slow-control replies



Example lpGBT frame formats at 10.24 Gbit/s

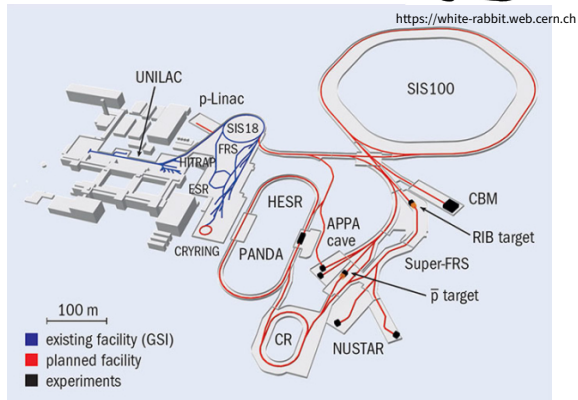
- ▶ Targeted at HL-LHC HEP experiments
  - ▶ Radiation-tolerant ASICs
  - ▶ Fixed latency
  - ▶ Fixed-phase clock recovery
  - ▶ Strong (Reed-Solomon) FEC to correct bursts of bit errors
  - ▶ Scrambling to provide DC balancing and to aid in low-jitter clock recovery

# Synchronising large networks: White Rabbit

Initiated by GSI and CERN, and grown into a large and diverse collaboration

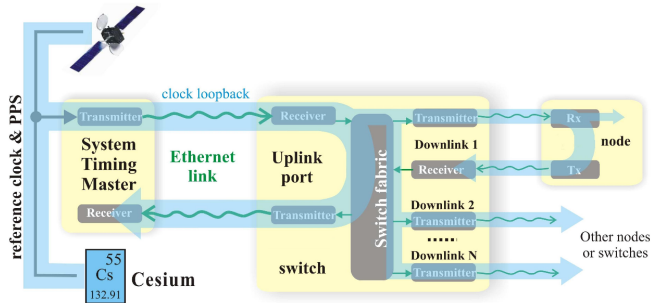


- ▶ Based on Gigabit Ethernet and PTP
- ▶ Extends those to provide deterministic data-transfer and sub-nanosecond synchronisation
- ▶ The whole system is disciplined by a GPS receiver
- ▶ Ratified as standard: **IEEE 1588-2019**
- ▶ Open source (HW, SW, FW)
- ▶ Commercially available (HW)



<https://cerncourier.com/a/fair-forges-its-future/>

# Synchronising large networks: White Rabbit



<https://white-rabbit.web.cern.ch>

## Synchronous Ethernet

- ▶ Deterministic packet delivery
- ▶ Synchronisation at the physical network layer by encoding the clock in the Ethernet carrier

## PTP without assumptions

- ▶ PTP improved by removing the approximation that  $\Delta_{\text{there}} = \Delta_{\text{back}}$  again
- ▶ Timestamps are corrected for phase drifts based on clock loop-back measurements



# Synchronising large networks: White Rabbit

- ▶ More than 'just' a timing distribution system
- ▶ Clock/frequency distribution using remote DDS
- ▶ Trigger distribution
- ▶ Time-based control
- ▶ Precision timestamping
- ▶ Ethernet network with fixed latency data transfer

Deployed at various scales in many different environments

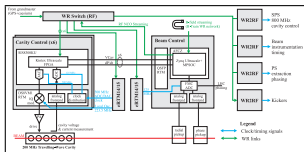
Used for the SPS RF upgrade recently, and planned for the HL-LHC RF upgrade



HERA construction photo log



Stanford University/Deutsche Börse Group



10.18429/JACoW-ICALEPCS2021-THBR02

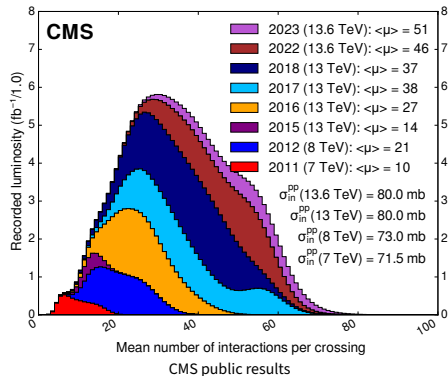
# Pileup – A luxury problem

## Luminosity

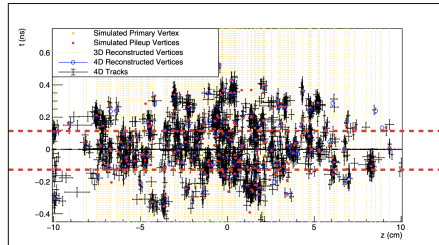
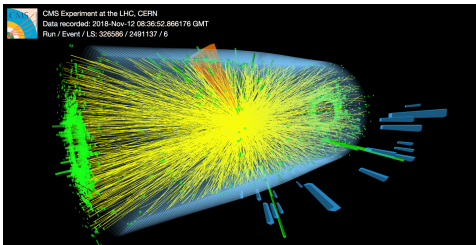
- ▶ The ‘brightness’ of the particle interaction region (instantaneous)
- ▶ The ratio between the interaction probability and the number of expected events (integrated)

## Pileup

- ▶ The number of additional proton-proton interactions within a single bunch crossing
- ▶ A challenge for triggering, event reconstruction, etc.



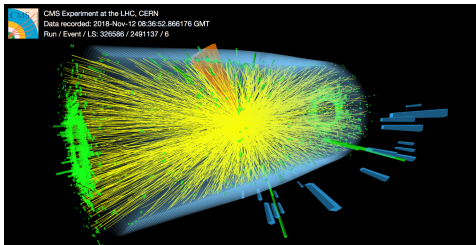
# Pileup – A luxury problem



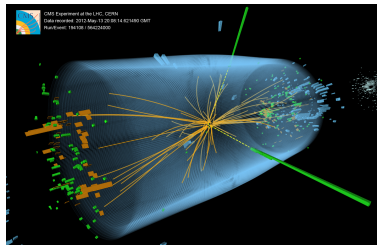
## The High-Luminosity LHC targets a pileup of 200

- ▶ The spatial density will increase beyond the achievable resolution
- ▶ Collisions will still be spaced ‘relatively well’ in time
- ▶ A time resolution  $30 \text{ ps}$  will allow slicing the collision window such that the  $O(200)$  pileup is reduced to an effective pileup of  $O(30-40)$

# Pileup – A luxury problem



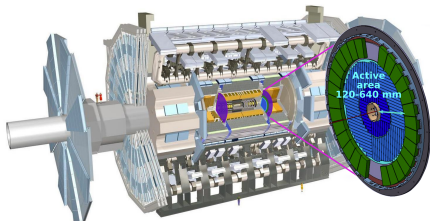
vs.



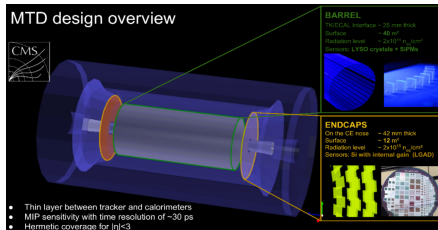
## The High-Luminosity LHC targets a pileup of 200

- ▶ The spatial density will increase beyond the achievable resolution
- ▶ Collisions will still be spaced ‘relatively well’ in time
- ▶ A time resolution 30 ps will allow slicing the collision window such that the  $O(200)$  pileup is reduced to an effective pileup of  $O(30-40)$

# ATLAS and CMS are retooling to include precision timing



DOI: 10.13140/RG.2.2.28529.81768



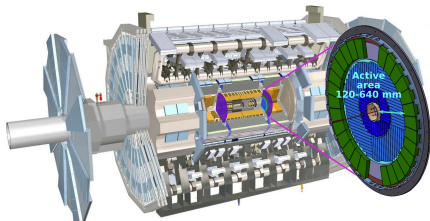
DOI: 10.1088/1742-6596/1162/1/012035

- ▶ Adds precise hit timestamping to the (spatial) tracking information
  - ▶ Reconstruction now takes place in 4D
- ▶ Clock distribution should stay in the shadow of the intrinsic detector resolution

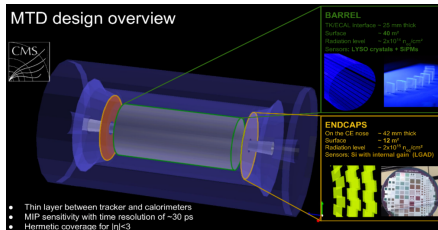
*A High-Granularity Timing Detector (HGTD) for the Phase-II upgrade of the ATLAS detector*, DOI: 10.1088/1748-0221/14/10/C10028

*Development of the CMS MIP timing detector*, DOI: 10.1016/j.nima.2019.04.044

# ATLAS and CMS are retooling to include precision timing



DOI: 10.13140/RG.2.2.28529.81768



DOI: 10.1088/1742-6596/1162/1/012035

## For example, the CMS MIP Timing Detector barrel

$$\sigma_t^{\text{BTL}} = \underbrace{\sigma_t^{\text{clock}}}_{\text{O}(15 \text{ ps})} \oplus \underbrace{\sigma_t^{\text{digi}}}_{\text{O}(7 \text{ ps})} \oplus \underbrace{\sigma_t^{\text{ele}}}_{\text{O}(8 \text{ ps})} \oplus \underbrace{\sigma_t^{\text{pho}}}_{\text{O}(25 \text{ ps})} \oplus \underbrace{\sigma_t^{\text{DCR}}}_{\text{O}(50 \text{ ps})}$$

A High-Granularity Timing Detector (HGTD) for the Phase-II upgrade of the ATLAS detector, DOI: 10.1088/1748-0221/14/10/C10028

Development of the CMS MIP timing detector, DOI: 10.1016/j.nima.2019.04.044

# Many factors affect detector timing accuracy

Many factors may affect detector timing accuracy, including:

- ▶ Pulse amplitude variations
- ▶ Pulse shape variations
- ▶ Random (front-end) electronics noise
- ▶ Signal integrity
- ▶ **Random and/or deterministic clock noise**
  - ▶ Irregular sampling clock distorts the signal or gives incorrect timing information
  - ▶ Multiplying a 'dirty clock' makes things even worse

→ **Need to generate and distribute a 'clean clock'**

Caffeine break ☕



# Many factors affect detector timing accuracy

Many factors may affect detector timing accuracy, including:

- ▶ Pulse amplitude variations
- ▶ Pulse shape variations
- ▶ Random (front-end) electronics noise
- ▶ Signal integrity
- ▶ **Random and/or deterministic clock noise**
  - ▶ Irregular sampling clock distorts the signal or gives incorrect timing information
  - ▶ Multiplying a 'dirty clock' makes things even worse

→ **Need to generate and distribute a 'clean clock'**

Clock generation, -distribution, -recovery, -quality

# Different views on clock quality: jitter and phase noise

$$S(t) = B(t) + A(t) \sin(\omega_c t + \phi(t))$$

## There are two (main) ways to look at clock quality

- ▶ In the time domain: **jitter**
- ▶ In the frequency domain: **phase noise**

## Jitter and phase noise

- ▶ Give complementary information
- ▶ Typically require different measurement instruments

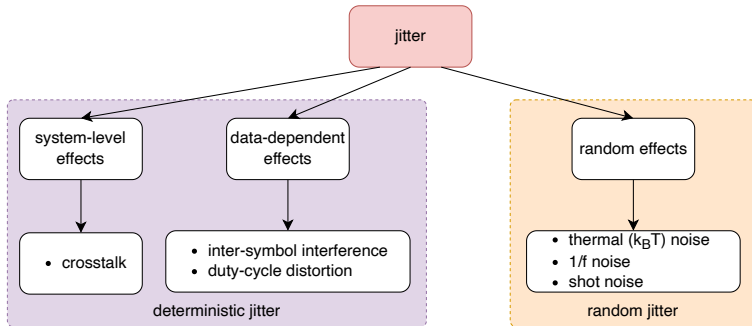
# Jitter as quality measure for digital signals

**Jitter** applies to all digital signals, and is often encountered in the context of high-speed serial signals, and clocks

**Jitter** means ever-so-slightly different things to different people

**Jitter** is about the **presence of signal transitions at times that they were not expected**

# Jitter has many sources, and two main types



- ▶ Deterministic, bounded jitter,  $J_{pp}^D$ , characterised by its peak-to-peak value
- ▶ Random, unbounded jitter,  $J_{RMS}^R$ , characterised by its RMS (and assumed to be Gaussian)

# Several 'jitter definitions' are common

There are different ways to (statistically) characterise clock quality

- ▶ Cycle-to-cycle jitter

The spread in differences between periods

- ▶ Period jitter

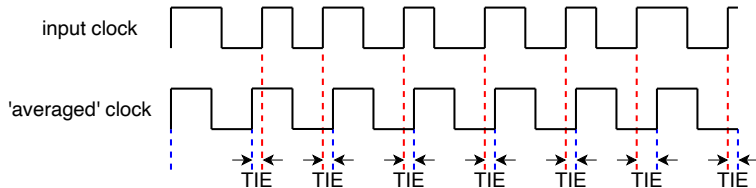
The RMS of the differences between successive periods

- ▶ **Time Interval Error (TIE)**

The deviation of each (clock) edge from its 'expected' value

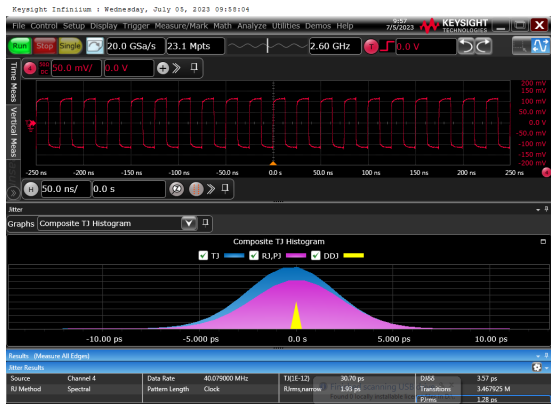
Expectation can be based on an ideal clock, on an average of the signal, or on the clock recovered from the signal

# TIE is the go-to time-domain clock quality measure



- ▶ Measures jitter as the difference between idealised and actual clock edges
- ▶ Statistical measurement
- ▶ Typically:
  - ▶ Performed using an oscilloscope
  - ▶ Based on an approximated/averaged 'ideal' clock
  - ▶ Sensitive to high-frequency jitter components

# A Keysight oscilloscope example of TIE

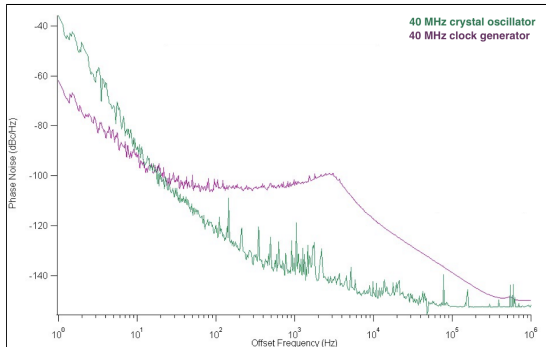


Beware: manufacturers all have their own jargon, models, etc.



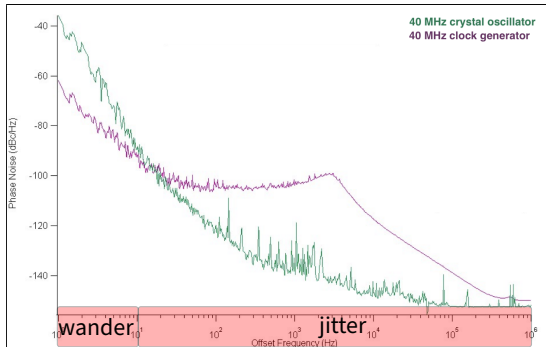
# Clock quality in the frequency domain: phase noise

- Provides insight into the spectral content of the jitter
- Specified w.r.t. a specific carrier (i.e., clock) frequency
- Sensitive to low-frequency jitter components
- Phase noise integrated over frequency window approximates total jitter
- Requires a dedicated (and expensive) phase noise analyser



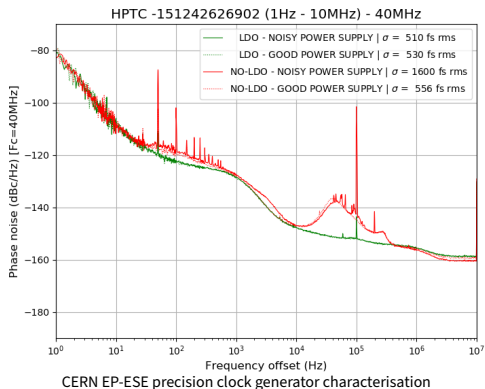
# Clock quality in the frequency domain: phase noise

- ▶ Low(-ish) frequencies
  - ▶ Often called '**wander**' ( $\leq 10$  Hz)
  - ▶ Hard/impossible to filter. Address by calibration and/or monitoring.
- ▶ High(er) frequencies
  - ▶ Can be filtered, at least to some extent
  - ▶ Reduce by careful system design, incl. power, cooling, etc.)



# Many system-level effects can creep into the clock quality

Good timing performance requires careful system design



- ▶ High-precision lab clock generator
- ▶ Used here to demonstrate the effect of power supply noise on clock quality

Extreme care to be taken in all areas of the design

**Performance is easier lost than maintained**

# Instrument manufacturers provide excellent literature

Agilent published a good three-part series, *Jitter – Understanding it, Measuring It, Eliminating It* (by J. Hancock), as well as several application notes

Other measurement instrumentation manufacturers all have their own, often excellent, documentation on jitter measurement and analysis

# Clock generation

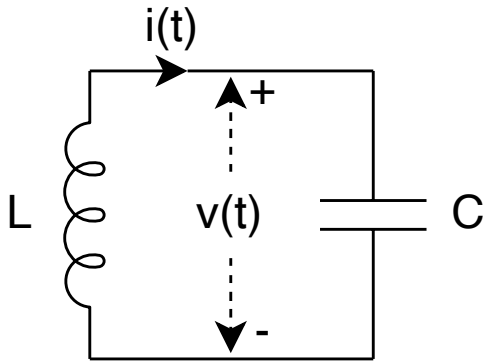
Or, in posher words: **frequency synthesis**

- ▶ Simple oscillator (RC, or LC)
- ▶ Crystal oscillator
- ▶ Phase-Locked Loop (PLL)
- ▶ Direct Digital Synthesis (DDS)

# Clock generation

Or, in posher words: **frequency synthesis**

- ▶ Simple oscillator (RC, or LC)
- ▶ Crystal oscillator
- ▶ Phase-Locked Loop (PLL)
- ▶ Direct Digital Synthesis (DDS)



# Quartz crystal oscillators: simple and robust

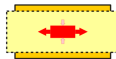
- ▶ Crystals cut and shaped to a characteristic frequency
- ▶ Crystals placed in oscillation circuit with positive feedback
- ▶ Cheap
- ▶ Very high Q-factor ( $10^6$  possible)
- ▶ Very low phase noise, especially at higher frequencies
- ▶ Crystal cut (not shape/size) determines frequency: reasonably temperature-stable



Wikimedia commons

# Quartz crystal oscillators: simple and robust

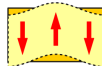
- ▶ Crystals cut and shaped to a characteristic frequency
- ▶ Crystals placed in oscillation circuit with positive feedback
- ▶ Cheap
- ▶ Very high Q-factor ( $10^6$  possible)
- ▶ Very low phase noise, especially at higher frequencies
- ▶ Crystal cut (not shape/size) determines frequency: reasonably temperature-stable



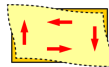
Longitudinal mode



Thickness shear mode



Flexural mode



Face shear mode



Tuning fork

Wikimedia commons



# Crystal oscillators can be further temperature-stabilised

## ► Temperature-compensated crystal oscillator (TCXO)

- Uses a varicap diode to pull the crystal frequency to oppose its drift

## ► Oven-controlled crystal oscillator (OCXO)

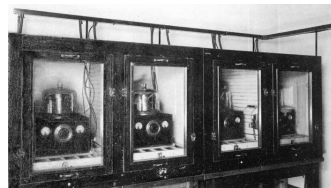
- Stabilises the crystal temperature, instead of compensating frequency drifts
- Expensive, and power-hungry, and 'large'

## ► Microcomputer-compensated crystal oscillator (MCXO)

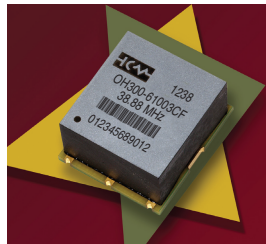
- The oscillator is allowed to drift, and the output is corrected by a processor algorithm deleting pulses to match the required frequency

## ► Rubidium crystal oscillators (RbXO)

- Periodically syntonises the XO to a Rubidium reference
- Too large, too heavy, too expensive, ...



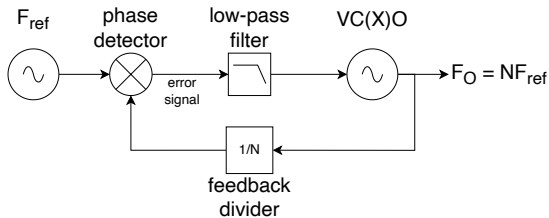
Wikimedia commons



↑22 × 26 × 13 mm  
Connor Winfield

# Phase-Locked Loops, for easy frequency multiplication

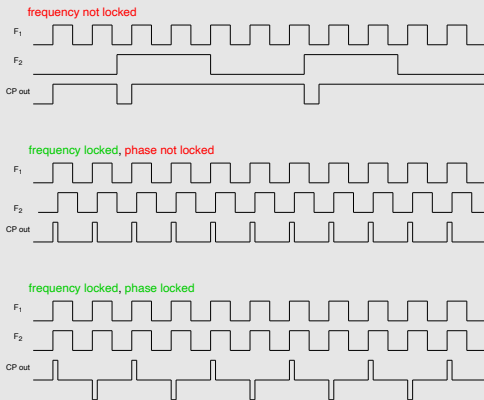
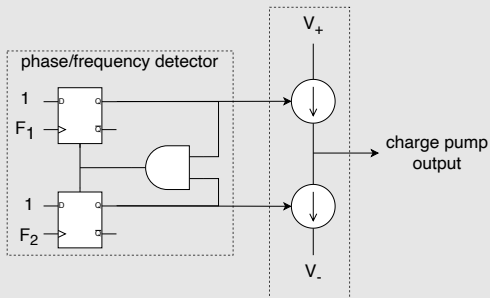
- ▶ Traditionally analogue, digital more and more common
- ▶ For frequency generation, clock recovery, -tracking, -cleaning
- ▶ Typical negative-feedback control loop



**Careful: noise in  $F_{\text{ref}}$  gets multiplied together with the clock signal, by a factor  $20 \log N$**

# Driving phase and frequency require the same action

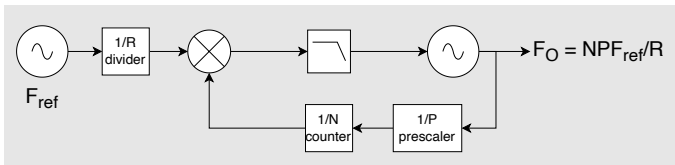
## Popular example of a phase detector and VFO driver



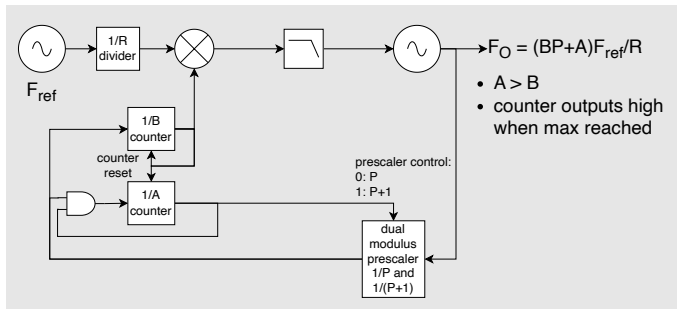
Continuously advancing the phase is the same as increasing the frequency

# Strategic dividers and prescalers help improve resolution

- ▶ Resolution/step size is set by the input frequency
- ▶ But, stable low-frequency sources are hard to find
- ▶ Solution: **use a stable, high-frequency source, divide that down, then account for that division in the feedback loop**
  - ▶ May also help reduce phase noise

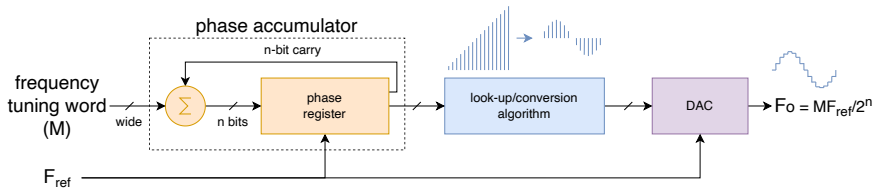


# The more the merrier: multi-scalers for fractional division

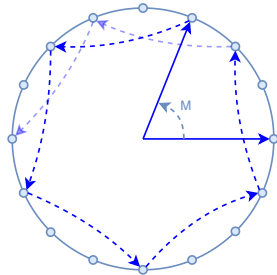


- ▶ Resolution is no longer determined by the reference frequency
- ▶ Possible to generate **any** output frequency
- ▶ Possible to reduce the in-loop multiplier, reducing phase noise in  $F_O$
- ▶  **$F_O$  is only correct on average**
  - ▶ Intrinsically more phase noise spurs due to constant pulling on the VCO

# Direct Digital Synthesis, for synthesis at a distance

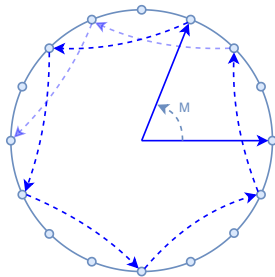


- ▶ Purely digital  $\rightarrow$  integrates well in digital systems
  - ▶ Allows 'remote control' frequency synthesis
- ▶ Hyper fine resolution possible (determined by  $n$ )
- ▶ Extremely (!) agile in frequency changes
- ▶ Well suited for generation of correlated signals



## Direct Digital Synthesis, for synthesis at a distance

- ▶ Points of attention to reduce jitter:
  - ▶ The DAC needs enough precision to get the output amplitude right
  - ▶ The reference clock needs to be sufficiently stable to get the output timing right
- ▶ In any case: low-pass filtering is required to remove spurs due to the sampled nature of the output signal



# Clock distribution – Separate or together with the data?

- ▶ Direct distribution of the clock signal itself: **source-synchronous**
  - ▶ Very clean clock signal possible
  - ▶ Requires dedicated clock path (i.e., more cables/fibres)
  - ▶ Requires clock-to-data phase adjustment on receiver
- ▶ Clock embedded in serial data stream: **self-synchronous**
  - ▶ No phase adjustments necessary on receiver side
  - ▶ Clock recovery in receiver not trivial
  - ▶ Requires care in encoding to help clock recovery

**Neither approach is intrinsically better. Each has its own characteristics and (dis-)advantages.**

In our detector designs, we tend to opt for embedded clock distribution, because it saves on fibres, electronics, etc.



# Synchronous = same clock, and known relative phase

**Synchronous** Systems A and B operate at the **same clock frequency**, and with a **fixed and known phase difference**

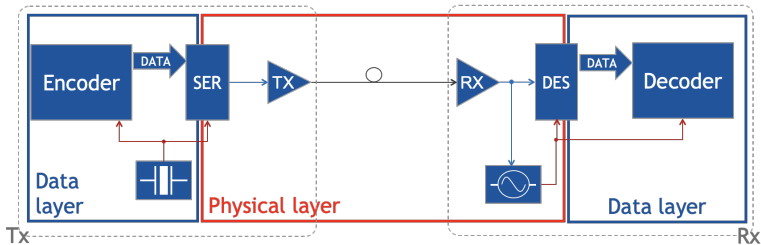
**Mesochronous** Systems A and B operate at the same clock frequency, and with a fixed but unknown phase difference

**Plesiochronous** Systems A and B operate at the same (nominal) clock frequency, but with a possible 'small' frequency mismatch, leading to a drifting phase difference

**Asynchronous** Systems A and B operate at different clock frequencies

**For our purposes, 'synchronous' is what you want**, and 'mesochronous' is relevant when writing firmware

# Self-synchronous: derive the clock from the incoming data



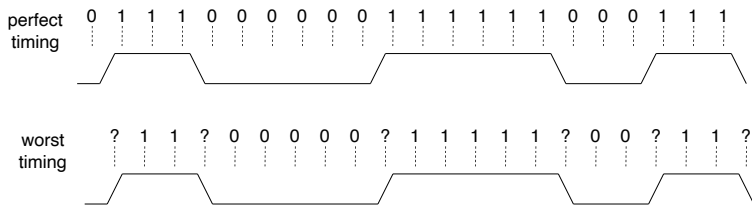
## Physical layer

- ▶ Physical interface (LVDS, CML, ...)
- ▶ Modulation Schemes (NRZ, PAM4, ...)
- ▶ Clock and Data Recovery (CDR)
- ▶ Signal integrity considerations
- ▶ Pre-emphasis, equalisation

## Data layer

- ▶ Encoding/scrambling
- ▶ Frame alignment
- ▶ Comma detection
- ▶ Error correction schemes
- ▶ Clock domain crossing

# Clock recovery using oversampling of serial data



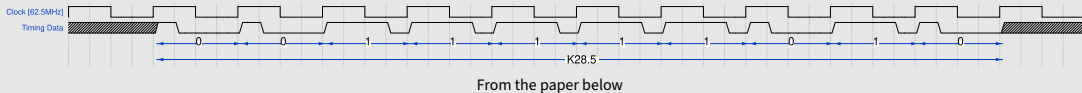
A clock can be reconstructed based on edges detected in the data stream. Oversampling by a factor  $N$  results in a clock with a quantisation jitter of  $1/N$ .

The data stream can now be sampled with the reconstructed clock, recovering the data.

Voilà, a basic Clock and Data Recovery (CDR) unit

# DUNE distributes its clock using only rising edges

## Duty Cycle Shift Keying (DCSK)

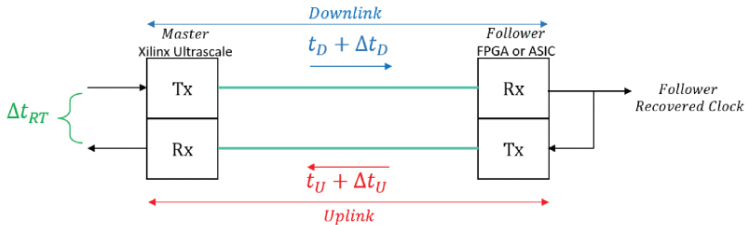


Partly born out of necessity, during the recent 'silicon crisis', when their chosen CDR ASIC was not available. All tested PLLs turn out to use the rising edges only, leaving one to encode the data in the falling edges → no CDR required.

Even possible to sample the DCSK signal with a delayed version of itself → no PLL required.

Additional benefit of GPIO-speed transmission: upstream data can be re-clocked with the recovered clock, after leaving any end-point logic.

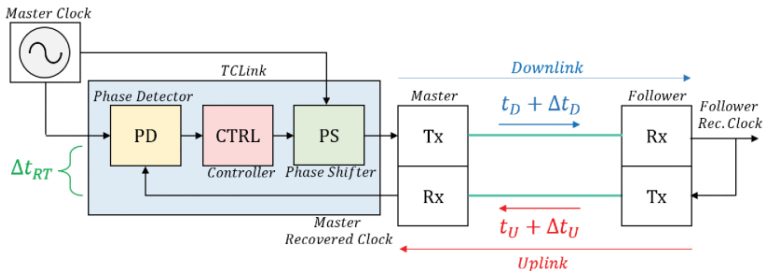
# Stabilising clock phase across serial links: the TCLink



The source-synchronous endpoint uses the recovered clock to drive the uplink. At the source, the round-trip time is measured (modulo the clock period) and taken as ‘desired phase’.

An assumption is required on the distribution of the round-trip delay between the downlink and the uplink. The default is  $\Delta t_D = \Delta t_U$ .

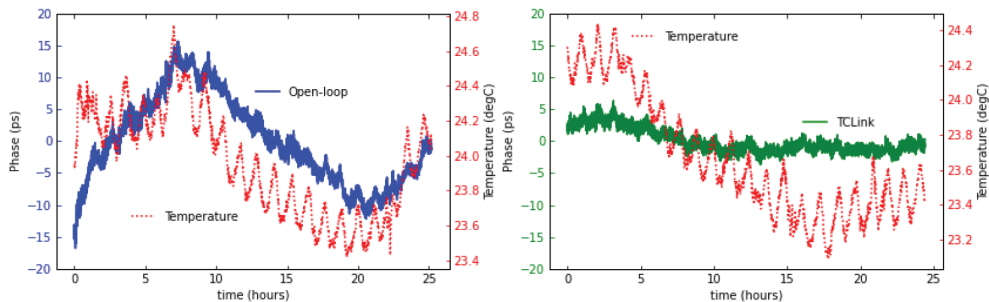
# Stabilising clock phase across serial links: the TCLink



A source-side control loop monitors the phase between the TX and RX clocks, and adjusts the phase of the TX clock to stabilise that phase.

The current implementation is bound to the GTH/GTY transceivers of the Xilinx UltraScale(+) generation of FPGAs

# Stabilising clock phase across serial links: the TCLink

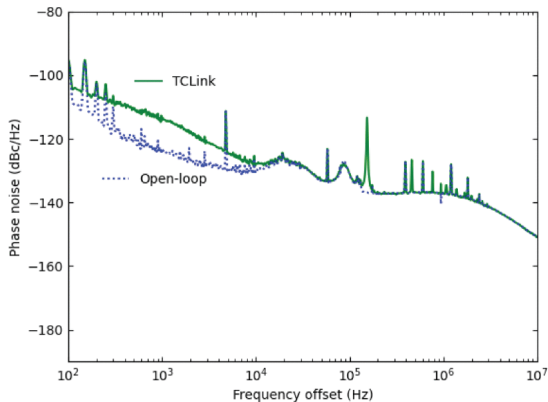


Tests with fibre lengths and temperature variations (semi-)representative for a detector setup show significant improvement in phase stability

---

*TCLink: A Fully Integrated Open Core for Timing Compensation in FPGA-Based High-Speed Links*, DOI: 10.1109/TNS.2023.3240539

# Stabilising clock phase across serial links: the TCLink

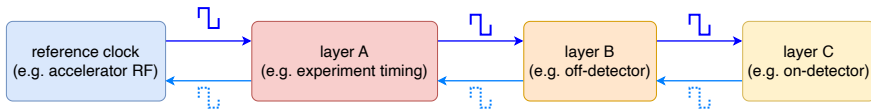


Just as with PLLs, nothing comes for free. Manipulating the TX clock introduces some additional phase noise, including some small spurs related to the sigma-delta converter frequency.

In the figure above:  $\sigma_{\text{open loop}} = 1.5 \text{ ps}$  and  $\sigma_{\text{TCLink}} = 1.6 \text{ ps} \rightarrow$  of academic interest only



# A (technical) note on configuration order



- ▶ Without clock, none of the digital electronics or links will function
- ▶ Even missing a single tick is likely to upset some things

## Configuration of system components has to follow the clock distribution path(s)

- ▶ Step-by-step bring-up of downstream distribution
- ▶ Ideally, the upstream/feedback thereafter self-organises
- ▶ Only when full chain is up: centrally distribute resets and synchronisation commands

# A (technical) note on configuration order

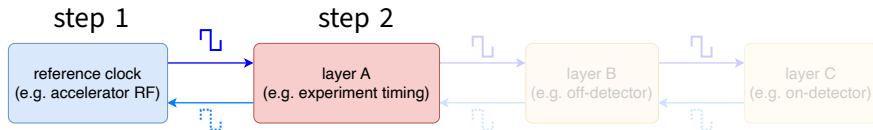


- ▶ Without clock, none of the digital electronics or links will function
- ▶ Even missing a single tick is likely to upset some things

## Configuration of system components has to follow the clock distribution path(s)

- ▶ Step-by-step bring-up of downstream distribution
- ▶ Ideally, the upstream/feedback thereafter self-organises
- ▶ Only when full chain is up: centrally distribute resets and synchronisation commands

# A (technical) note on configuration order

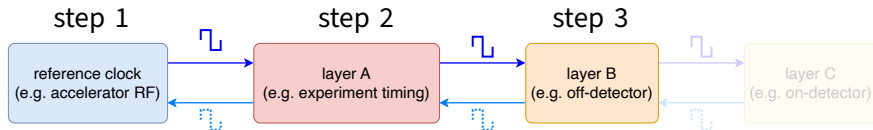


- ▶ Without clock, none of the digital electronics or links will function
- ▶ Even missing a single tick is likely to upset some things

## Configuration of system components has to follow the clock distribution path(s)

- ▶ Step-by-step bring-up of downstream distribution
- ▶ Ideally, the upstream/feedback thereafter self-organises
- ▶ Only when full chain is up: centrally distribute resets and synchronisation commands

# A (technical) note on configuration order

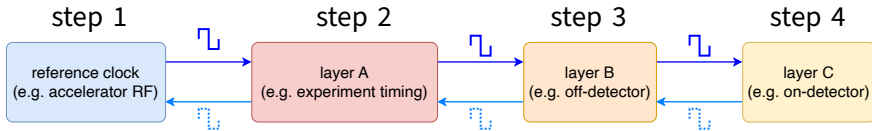


- ▶ Without clock, none of the digital electronics or links will function
- ▶ Even missing a single tick is likely to upset some things

## Configuration of system components has to follow the clock distribution path(s)

- ▶ Step-by-step bring-up of downstream distribution
- ▶ Ideally, the upstream/feedback thereafter self-organises
- ▶ Only when full chain is up: centrally distribute resets and synchronisation commands

# A (technical) note on configuration order

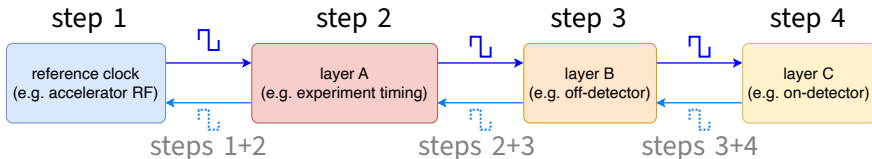


- ▶ Without clock, none of the digital electronics or links will function
- ▶ Even missing a single tick is likely to upset some things

## Configuration of system components has to follow the clock distribution path(s)

- ▶ Step-by-step bring-up of downstream distribution
- ▶ Ideally, the upstream/feedback thereafter self-organises
- ▶ Only when full chain is up: centrally distribute resets and synchronisation commands

# A (technical) note on configuration order

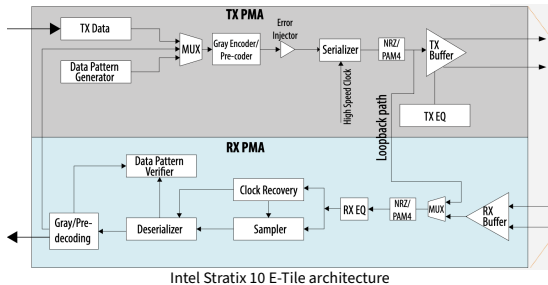


- ▶ Without clock, none of the digital electronics or links will function
- ▶ Even missing a single tick is likely to upset some things

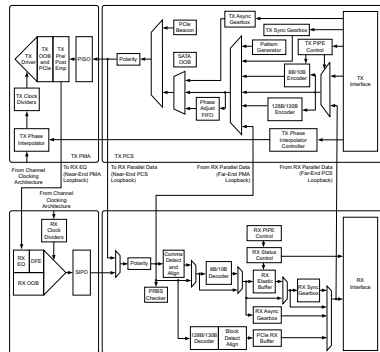
## Configuration of system components has to follow the clock distribution path(s)

- ▶ Step-by-step bring-up of downstream distribution
- ▶ Ideally, the upstream/feedback thereafter self-organises
- ▶ Only when full chain is up: centrally distribute resets and synchronisation commands

# Serial links are designed to transmit data, not clocks



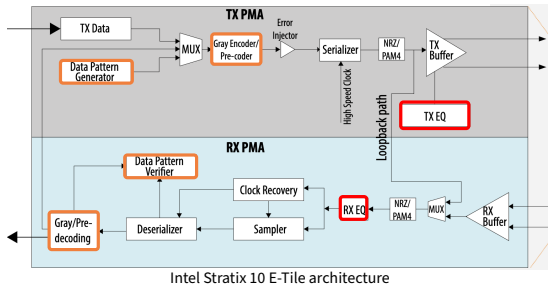
Intel Stratix 10 E-Tile architecture



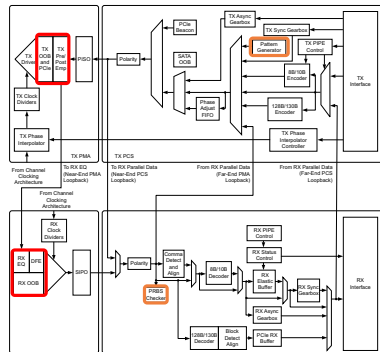
Xilinx UltraScale GTY User Guide

- Modern (FPGA) transceivers are amazingly robust against voltage and temperature changes, fibre attenuation, etc.
- Transceiver dynamics are designed for reliable data transport, not for predictable/fixed latency

# Serial links are designed to transmit data, not clocks



Intel Stratix 10 E-Tile architecture

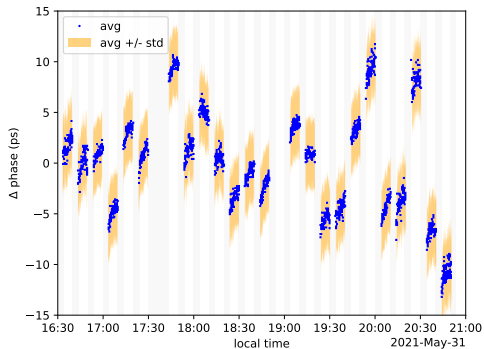
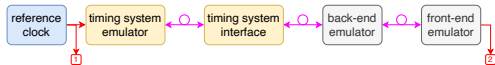


Xilinx UltraScale GTY User Guide

- Modern (FPGA) transceivers are amazingly robust against voltage and temperature changes, fibre attenuation, etc.
- Transceiver dynamics are designed for reliable data transport, not for predictable/fixed latency



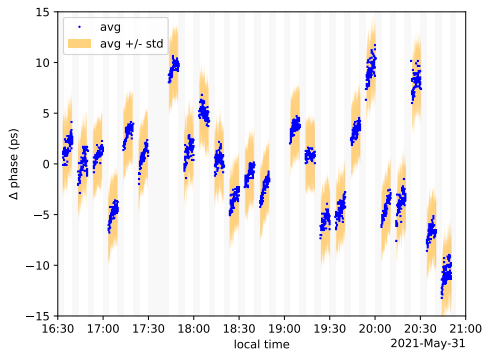
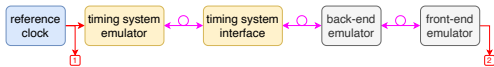
# Non-fixed latency means non-fixed sampling clock phase



Phase-2 CMS DAQ TDR, CERN-LHCC-2021-007

- ▶ Representative clock distribution demo
- ▶ Measure phase between points 1 and 2
- ▶ Periodically reconfigure the full chain
  - ▶ Under the hood, some of the transceiver 'choices' affect the signal latency, i.e., the recovered clock phase
  - ▶ Latency/phase jumps at each stage are random and bounded
  - ▶ Cascading stages accumulates phase jumps

# Non-fixed latency means non-fixed sampling clock phase



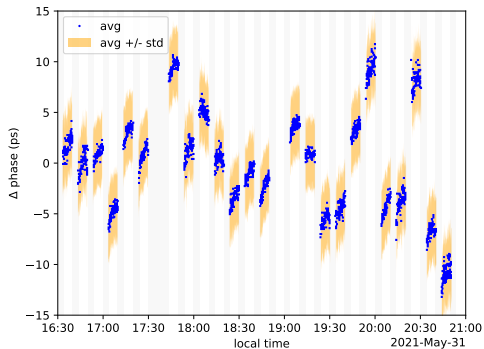
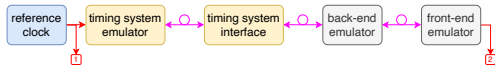
Phase-2 CMS DAQ TDR, CERN-LHCC-2021-007

## Phase calibration will require collision data

Exploit that:

- ▶ all channels fed by a single link share the same clock phase
- ▶ time differences within a single subsystem are immune to global phase jumps
- ▶ Firmware design (see also TCLink) as much as possible constrains the transceivers, to mitigate this effect
- ▶ Impossible to fully pin down clock phase without affecting link functionality

# Non-fixed latency means non-fixed sampling clock phase



Phase-2 CMS DAQ TDR, CERN-LHCC-2021-007

Precise timing requires extremely-precise timing-in, and continuous recalibration, monitoring, and adjustment

Wrap up

# Wrapping up...

- ▶ In the context of HEP experiments,  
**'timing and synchronisation' covers a broad range of topics**
- ▶ **Each of these topics is relevant** for detector design, data-taking, object reconstruction, and data analysis
- ▶ **Understanding the basics is important for all of us**
- ▶ For those of you who'd like to dig deeper, **there are many opportunities**
  - ▶ in the design of new detectors and experiments,
  - ▶ in understanding our current detectors and systems,
  - ▶ in reconstruction and analysis

The background of the slide features a white central area with two large teal triangles pointing towards each other from the left and right sides, meeting at a point at the bottom. The text is centered in the white area.

Time for (more) questions

Now, live, or later via [jeroen.hegeman@cern.ch](mailto:jeroen.hegeman@cern.ch)