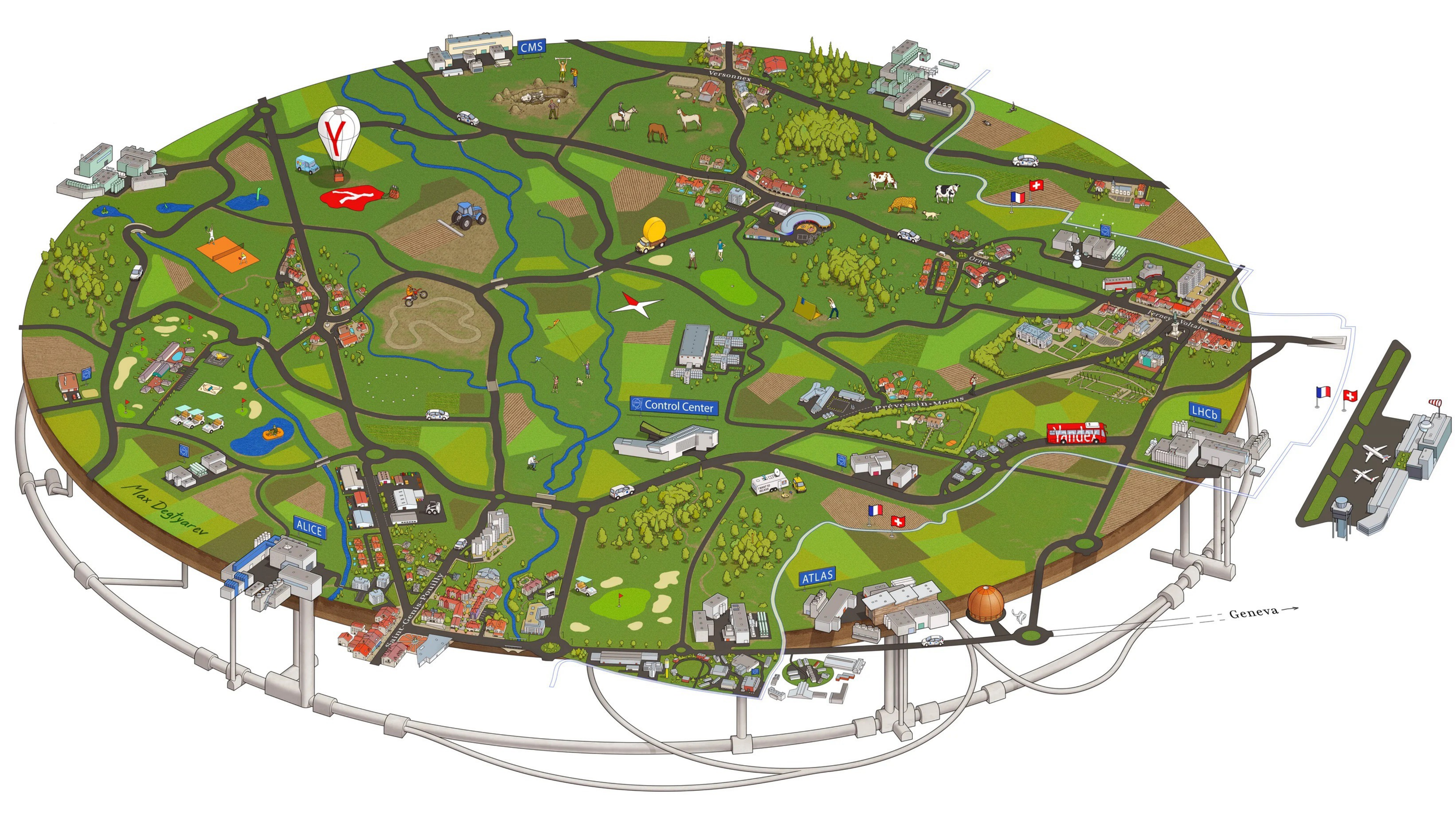
Phase Monitoring and Alignment on AMD Transmitters for Timing Distribution in HEP

Edoardo Orzes

CERN - EP ESE BE - 12 June 2024

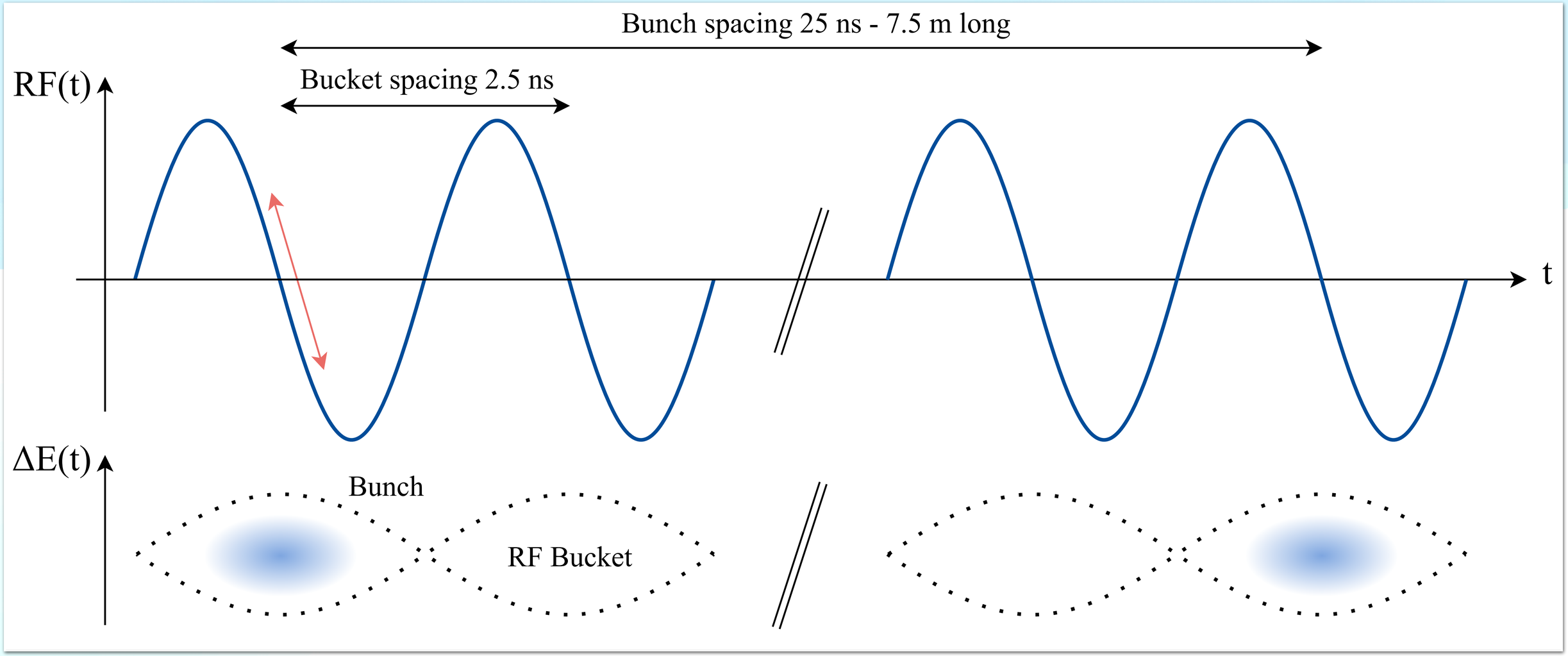
Contents

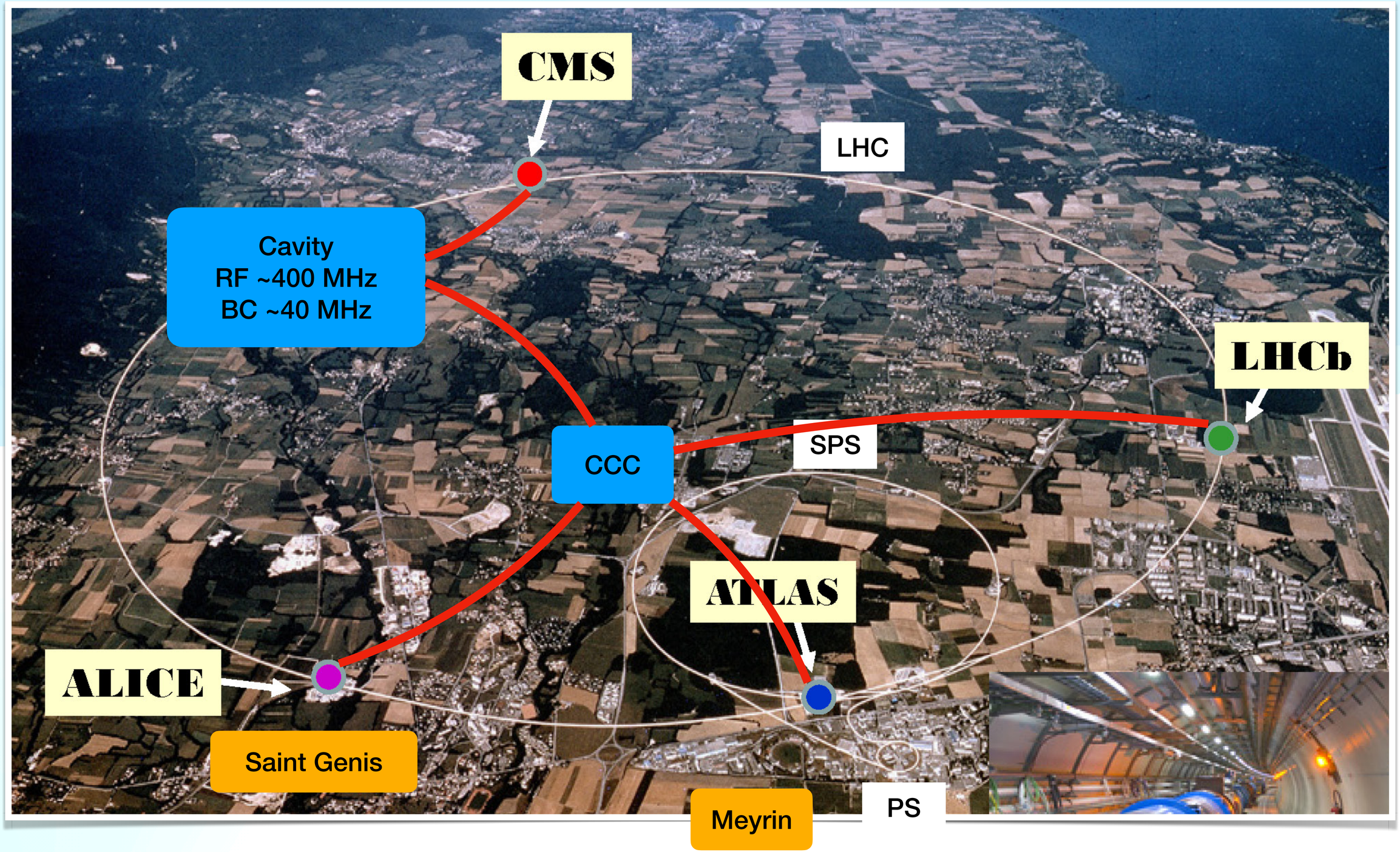
- What is phase determinism and alignment
- Why is it needed
- Proposed transceiver configuration
- Phase monitoring and alignment measurements
- DDMTD theory
- Comparison with other solutions


CMS
Versonnex
Ornex
Ferney-Voltaire
Prévessin-Moëns
Control Center
LHCb
Yandex
ALICE
Saint-Genis-Pouilly
ATLAS
Geneva
Max Degtyarev

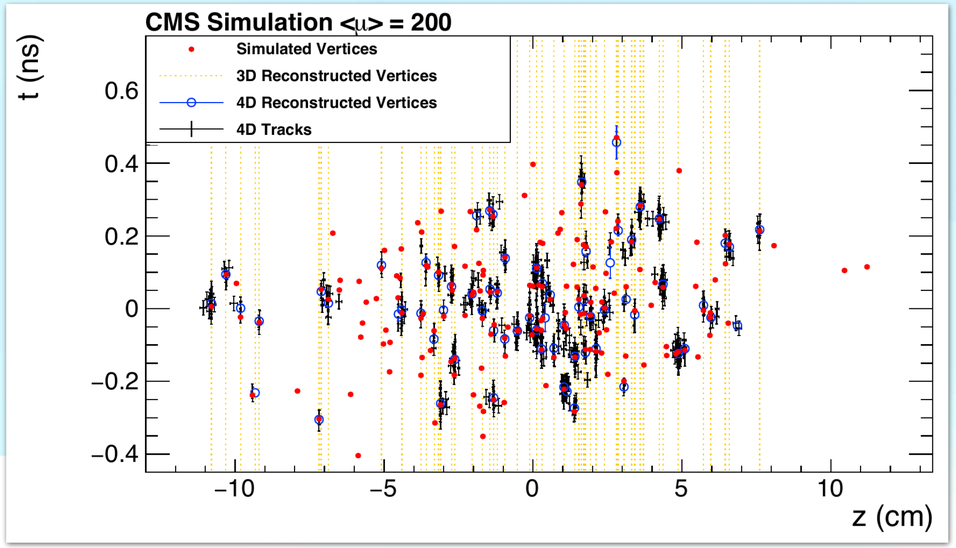
Particles are maintained in their configuration thanks to the RF Cavities in point 4.

Bunch Structure in LHC



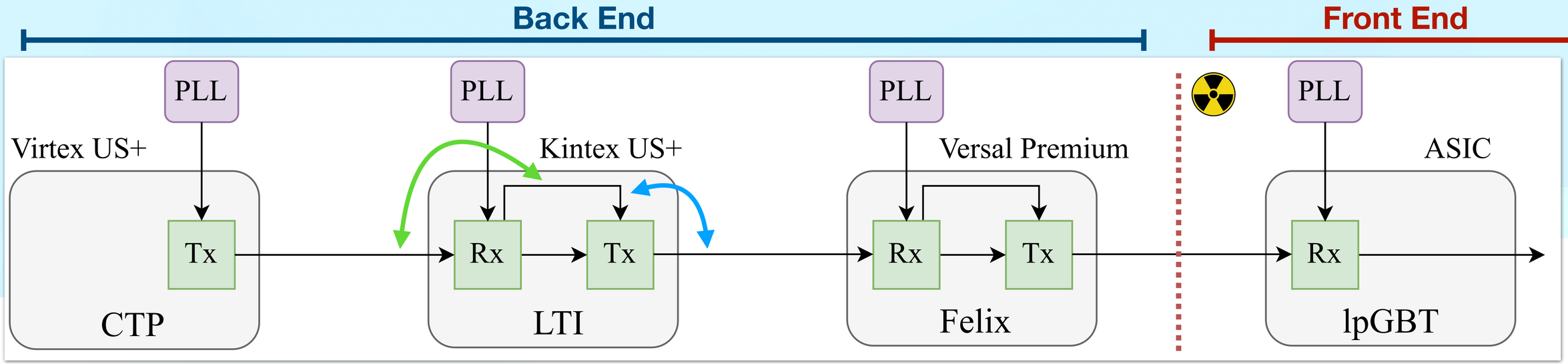

CMS
LHC
Cavity
RF ~400 MHz
BC ~40 MHz
LHCb
CCC
SPS
ATLAS
ALICE
Saint Genis
Meyrin
PS

Phase Stability in HL-LHC



- With the High-Lumi upgrade there will be 200 collisions per bunch crossing, which is a factor of 5 from the current scenario. **High Pileup** because of superposed collisions.
- Adding a 4th dimension allows to distinguish superposed collisions in space.
- Picosecond-level phase stability is required.

Timing Distribution System - ATLAS Network



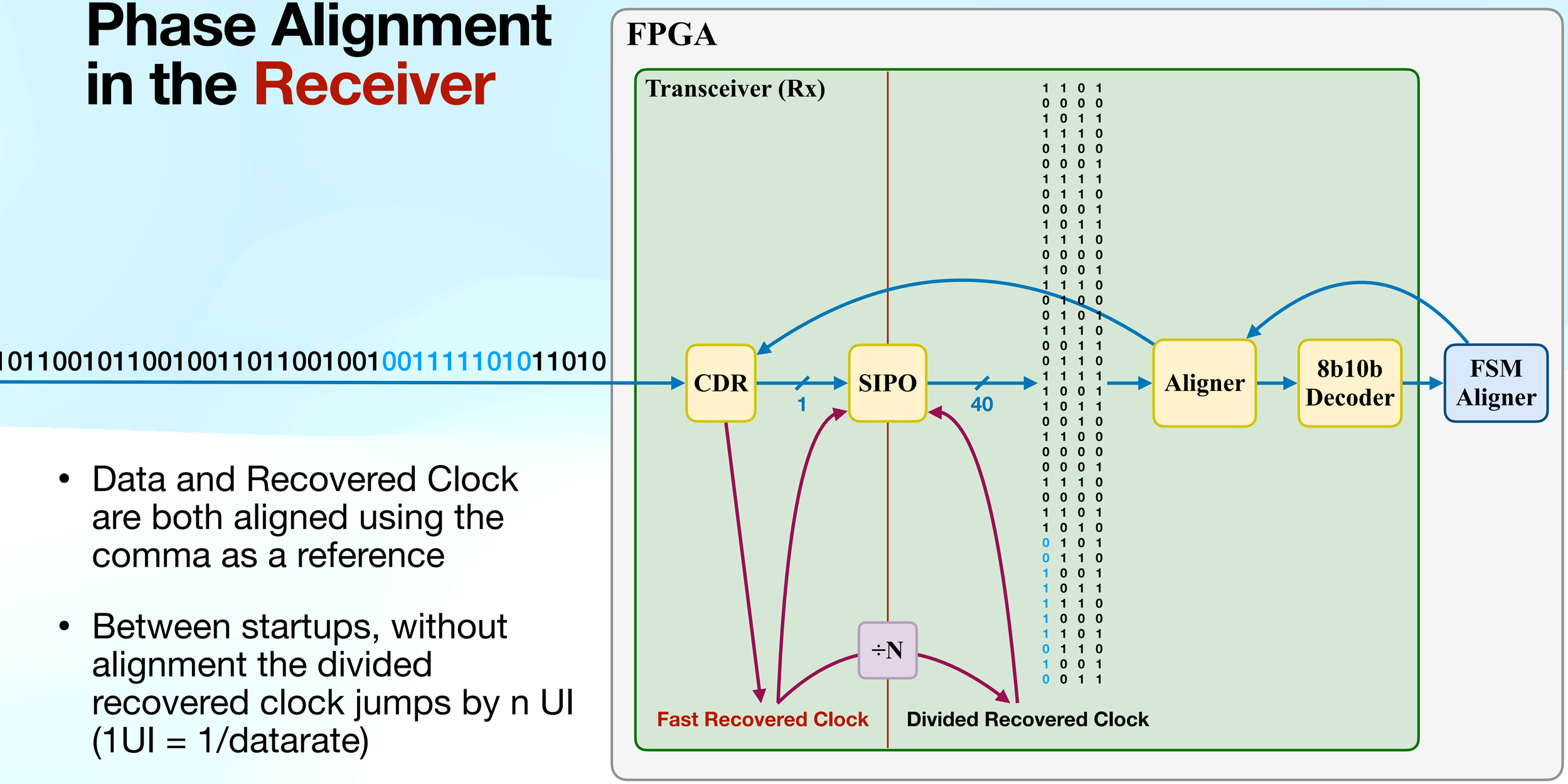
- The Reference Clock is embedded in the data stream.
- To maintain synchronization the clock is recovered from the incoming data and used as a reference for the next transmission of the cascade.
- The transmission is synchronous (phase aligned) to its reference clock.

Phase Alignment

- Data streams are parallelized when entering in an FPGA, to be handled with a slower clock. (Ex: 9.6Gbps with 240MHz clock means division by 40)
- To handle the data always with the same offset, frame headers are used to perform the alignment.
- With 8b10b encoding, the headers are called commas: in the receiver a state machine performs the alignment by shifting bit by bit the parallelized data until the comma is in the predefined position.

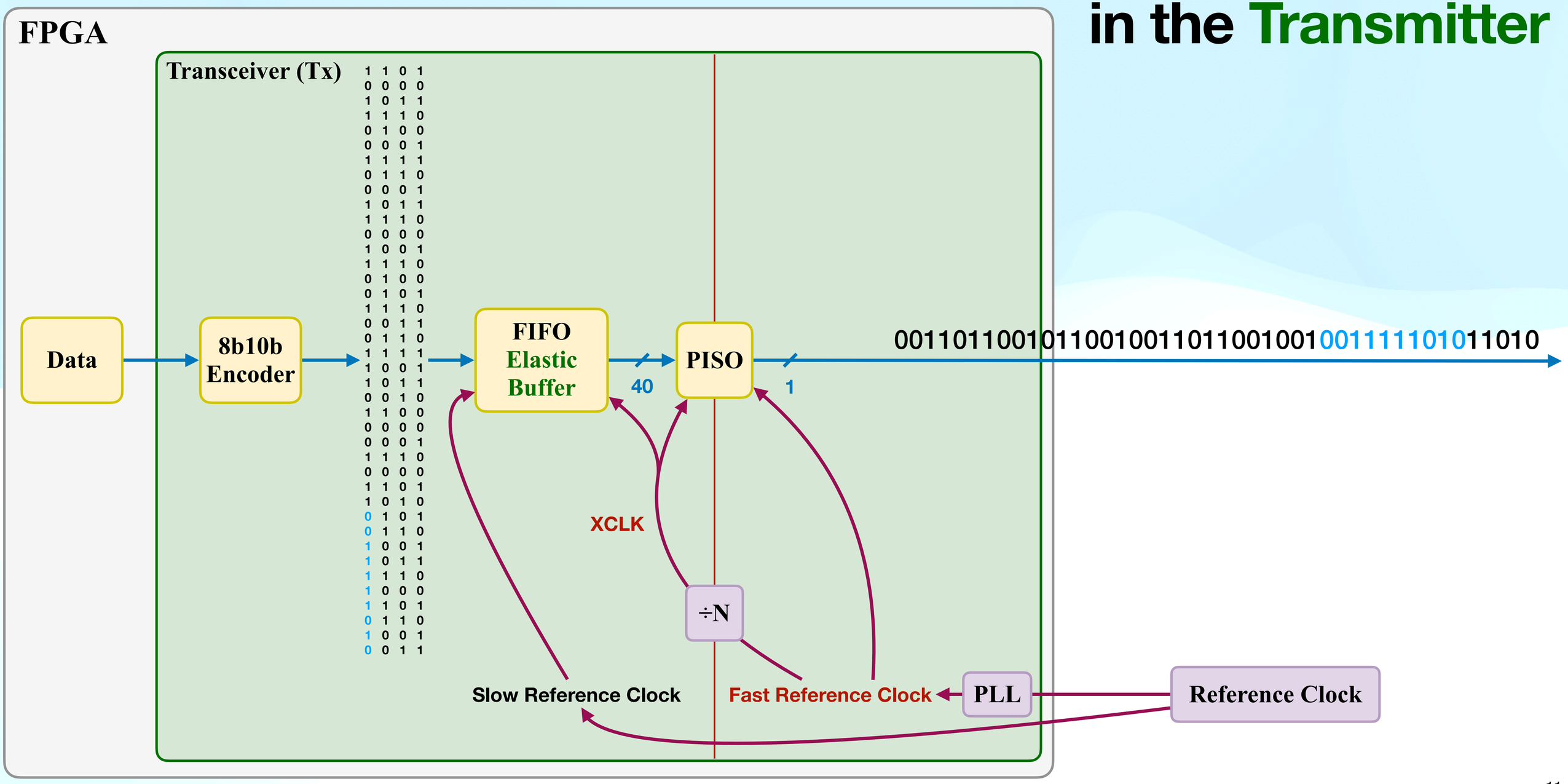
0100101001	1011001001	1101001011	Comma (10 bit)
0100101001	1011001001	1101001011	0100110110
0100101001	1011001001	1101001011	0100110110

Phase Alignment in the Receiver



- Data and Recovered Clock are both aligned using the comma as a reference
- Between startups, without alignment the divided recovered clock jumps by n UI (1UI = 1/datarate)

Phase Alignment in the Transmitter



Default clocking architecture (non deterministic)

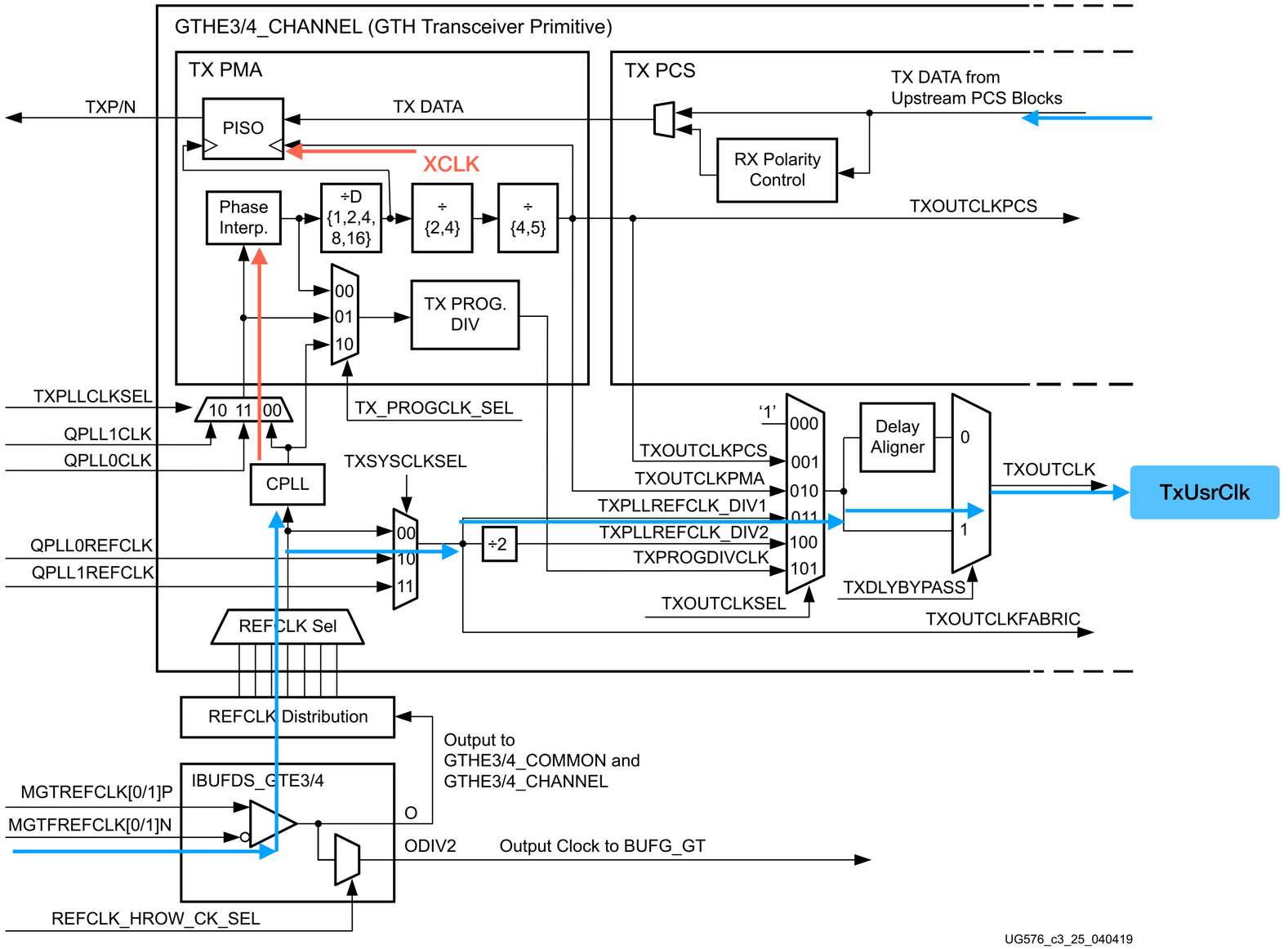


Figure 3-29: **TX Serial and Parallel Clock Divider**

Proposed clocking architecture for Phase Monitoring and Alignment

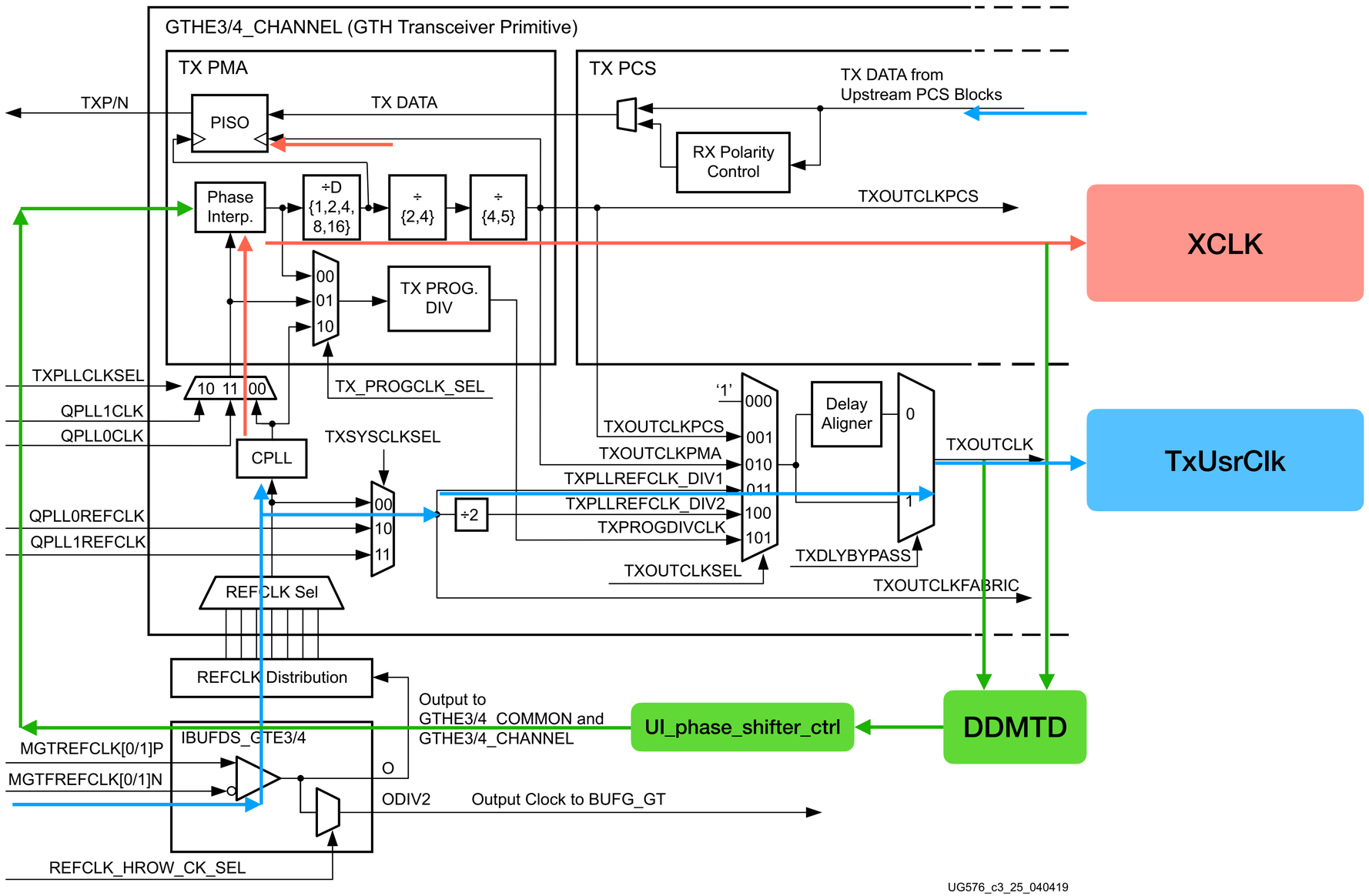
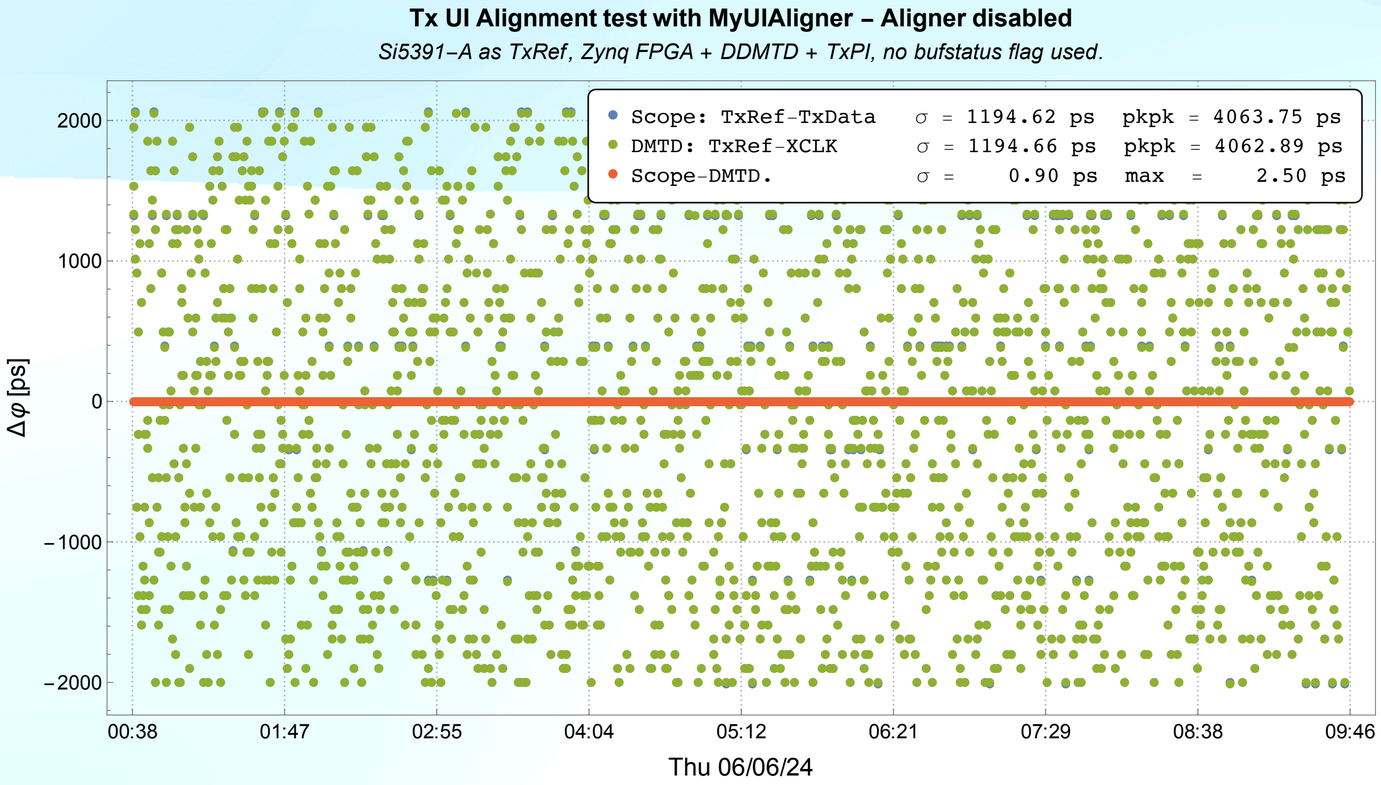


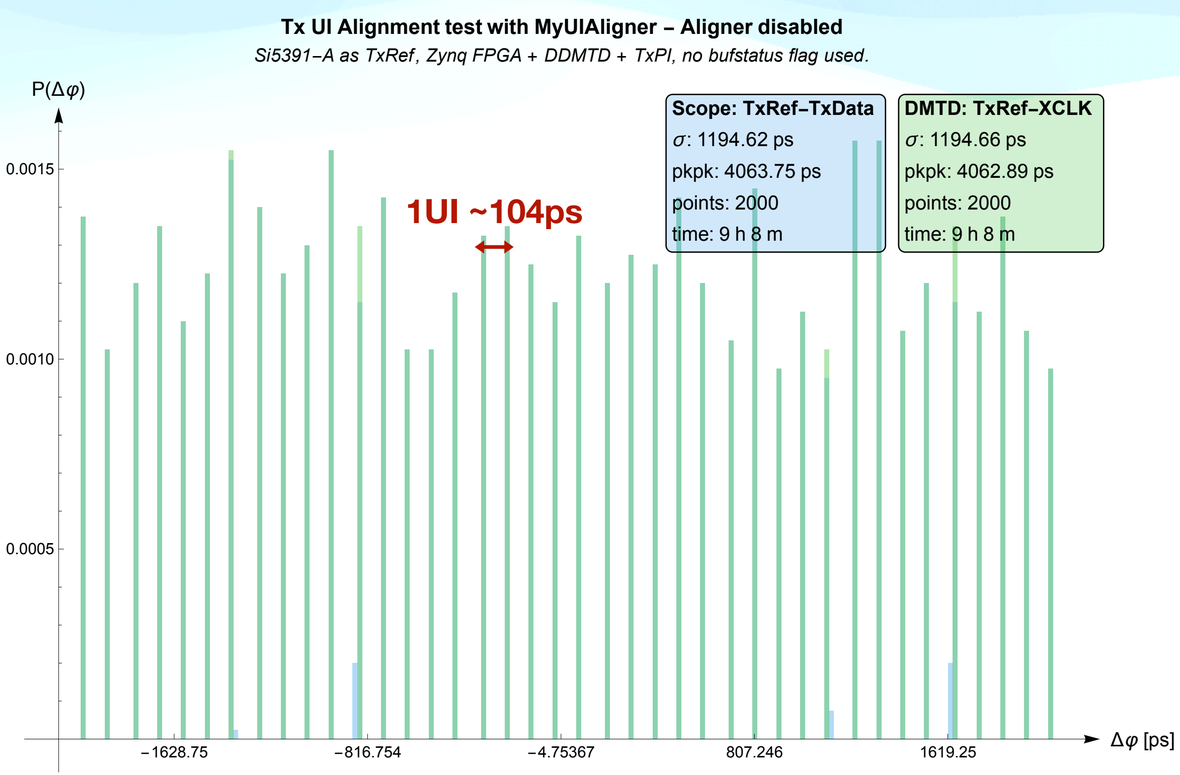
Figure 3-29: **TX Serial and Parallel Clock Divider**

- It is possible to extract the serializer clock, XCLK, which is fixed phase with the data stream
- A DDMTD measures the phase between XCLK and TxRef
- The transceiver Phase Interpolator (PI) is controlled via DRP to perform the phase shift
- The step is UI/64 (1UI = 1/datarate)
- The required shifts are by n x 64

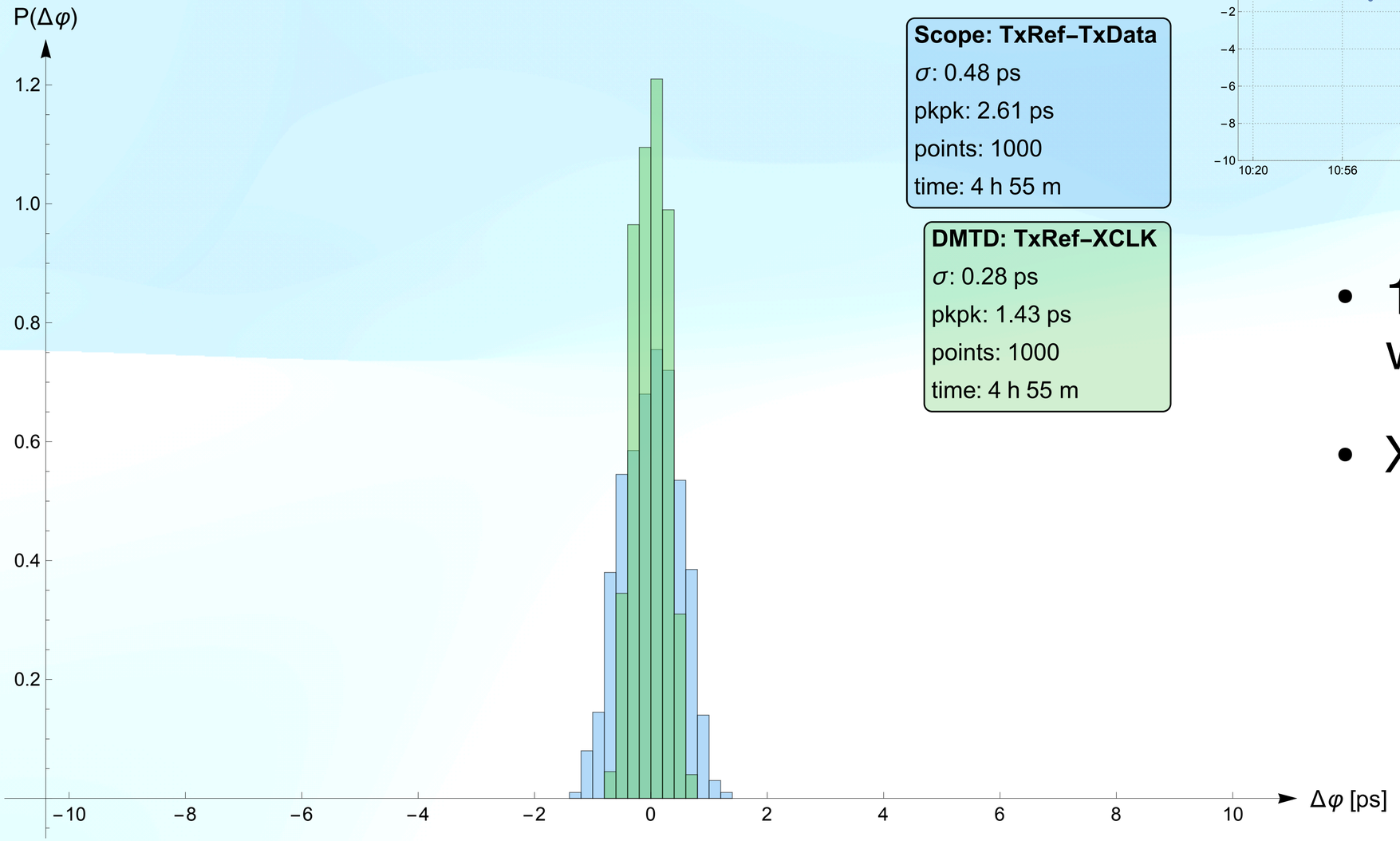
Default clocking architecture (non deterministic)

- At each tx reset the phase relation between XCLK and RefClk changes by n UI
- The phase relation between XCLK and the Data Stream is always fixed





Deterministic solution



Scope: TxRef–TxData

σ: 0.48 ps

pkpk: 2.61 ps

points: 1000

time: 4 h 55 m

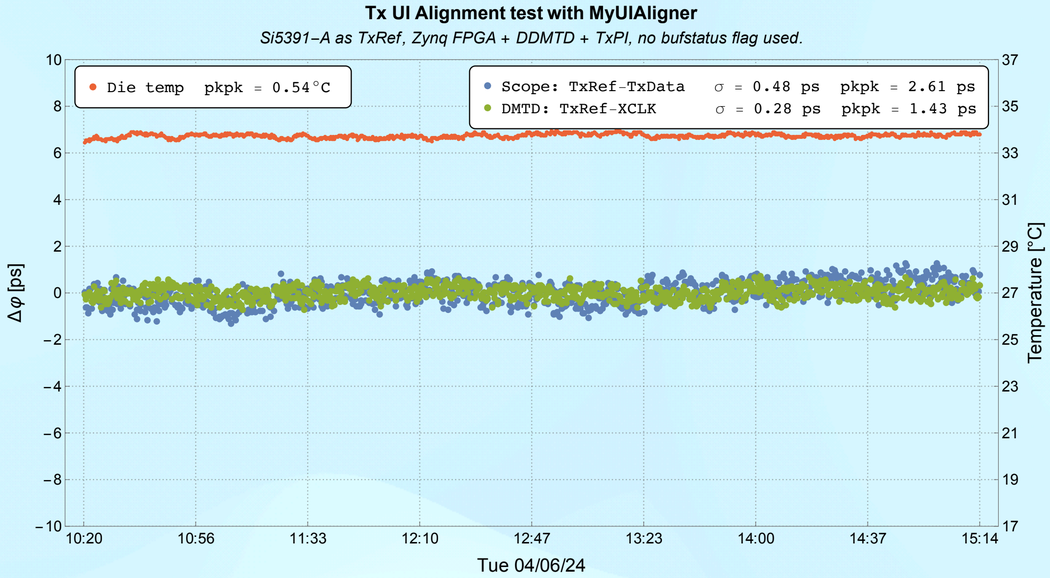
DMTD: TxRef–XCLK

σ: 0.28 ps

pkpk: 1.43 ps

points: 1000

time: 4 h 55 m



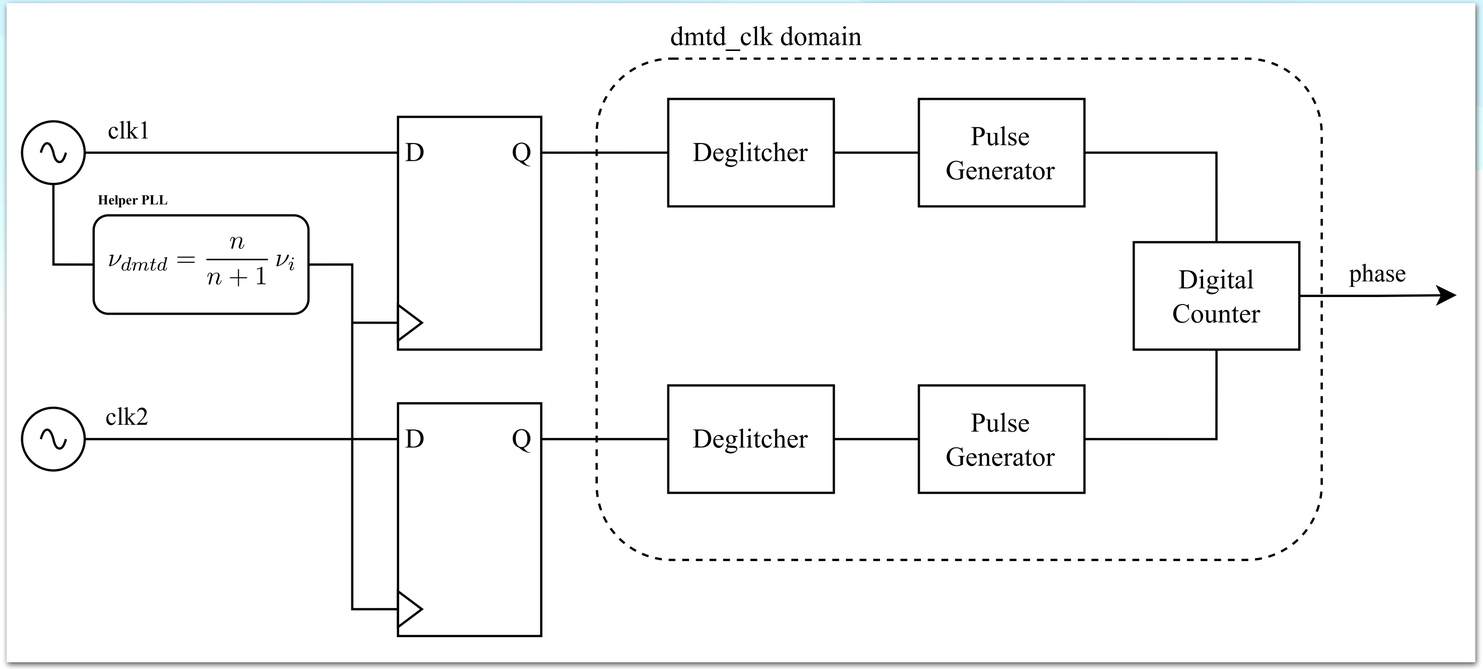
- 1.4ps phase determinism within 1000 tx resets
- XCLK-TxRef & Data-TxRef

My phase alignment project: https://gitlab.cern.ch/eorzes/tx-ui-aligner/-/tree/master

DDMTD

Digital Dual Mixer Time Difference

- HDL module which allows measure phase inside the FPGA with ps-level accuracy
- Requires an offset clock to be slightly slower than the input clock

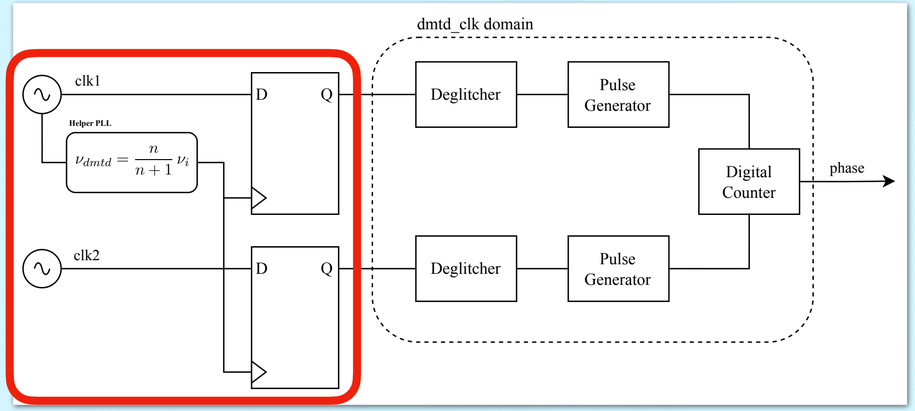


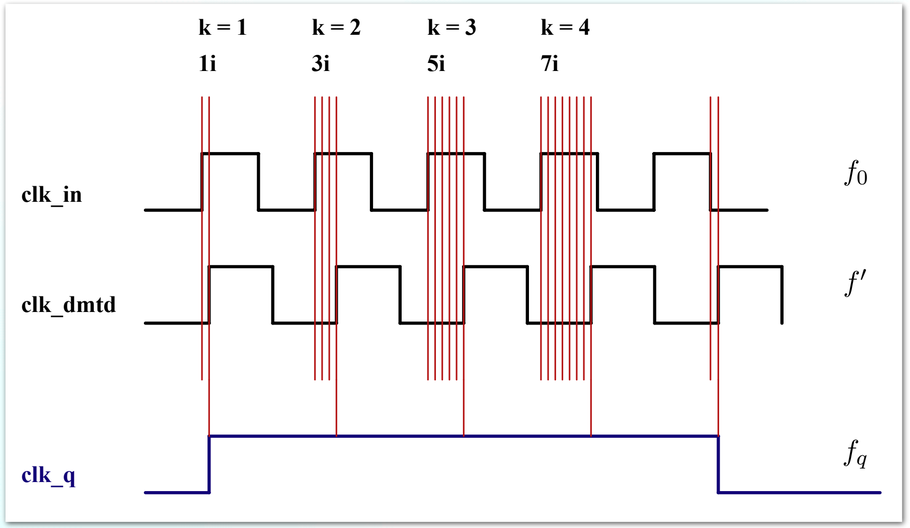
My DDMTD: https://gitlab.cern.ch/eorzes/tx-ui-aligner/-/tree/master/src/hdl/myddmtd
Original DDMTD: https://white-rabbit.web.cern.ch/documents/DDMTD_for_Sub-ns_Synchronization.pdf

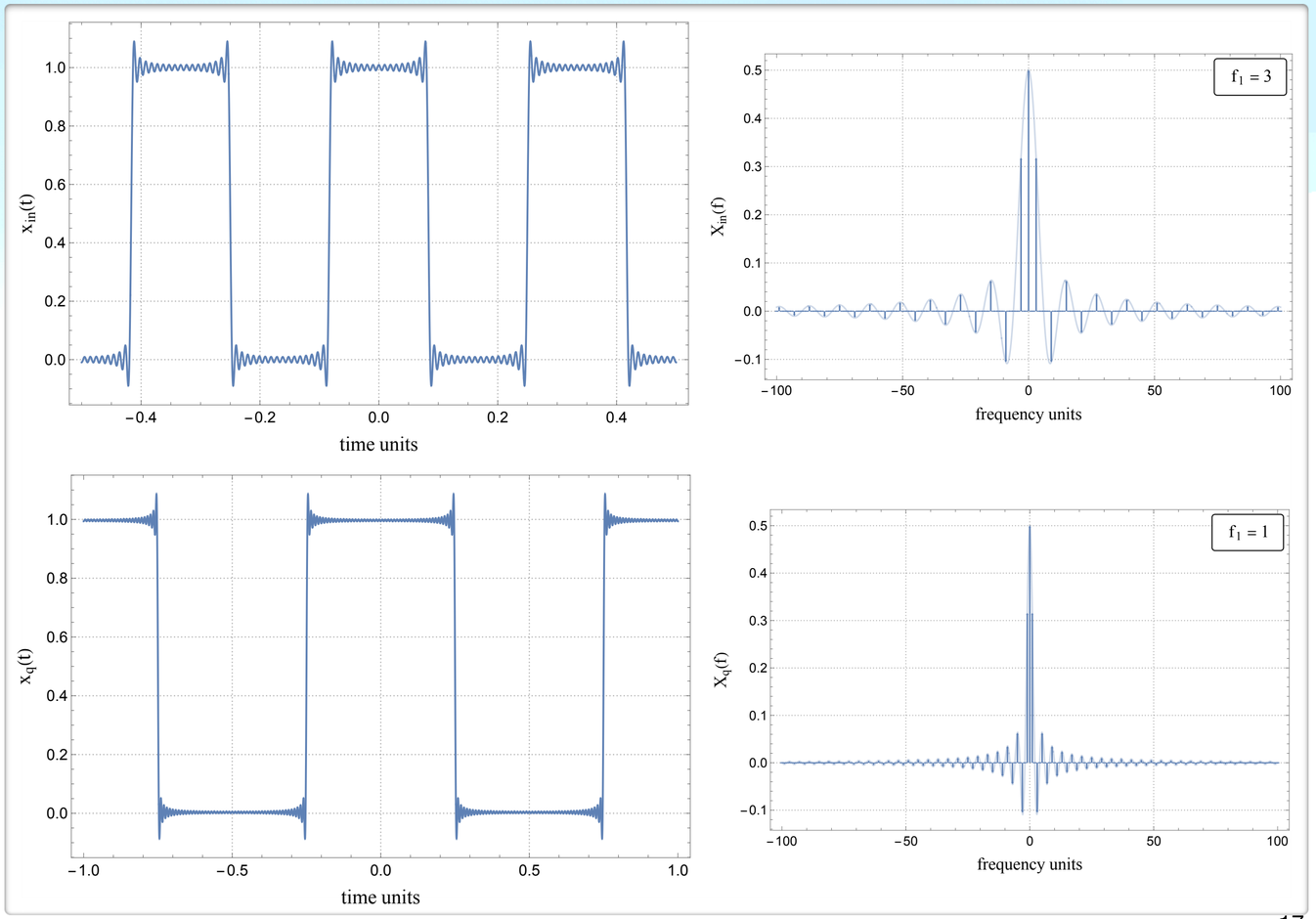
DDMTD - Sampler

- The output period is the amplification of the input period by a factor of n+1.

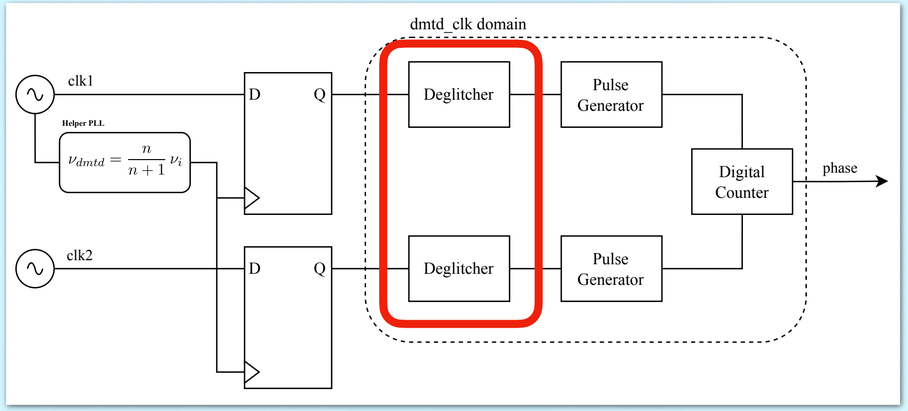
$$\nu_{dmtd} = \frac{n}{n+1}\nu_i \qquad \nu_q = \frac{1}{n+1}\nu_i$$



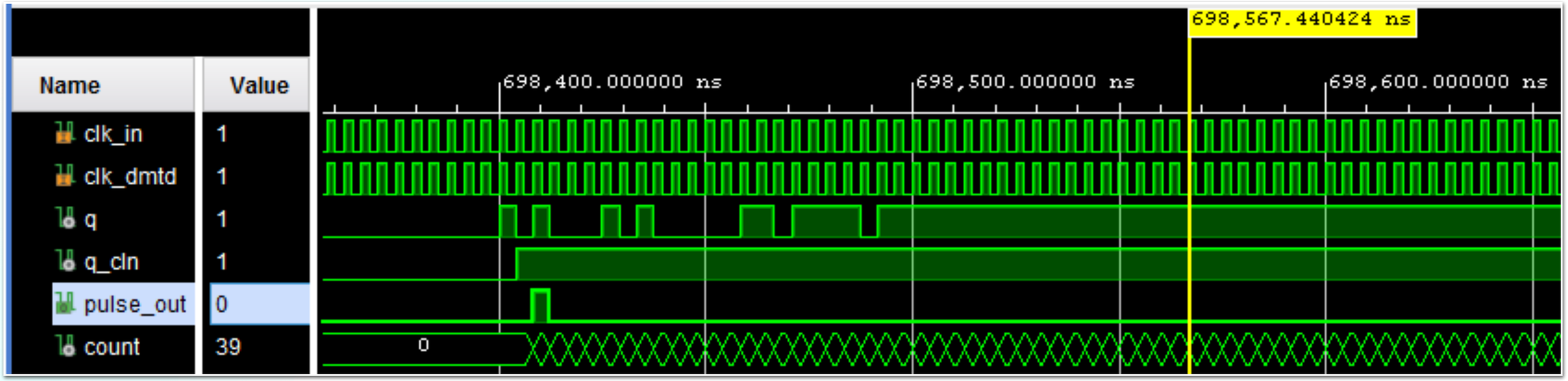




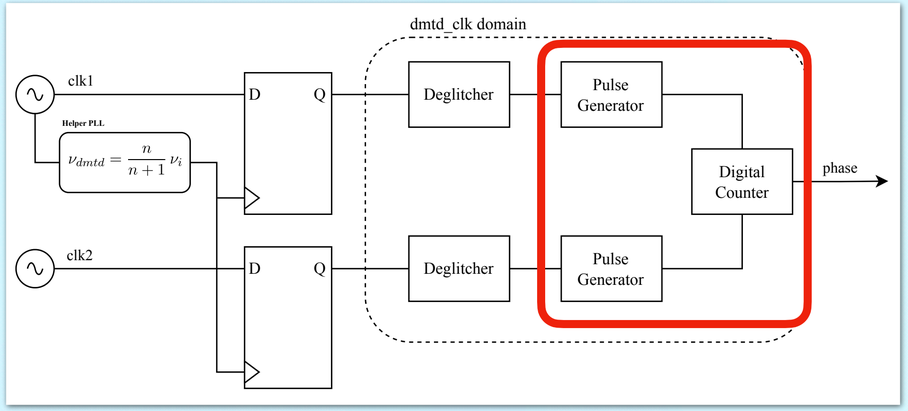
DDMTD - Deglitcher



- At high resolutions jitter is sampled, causing glitches on the sampler's output.
- The deglitcher is a state machine that removes the glitches by choosing as a valid transition the last one.



DDMTD - Digital Counter

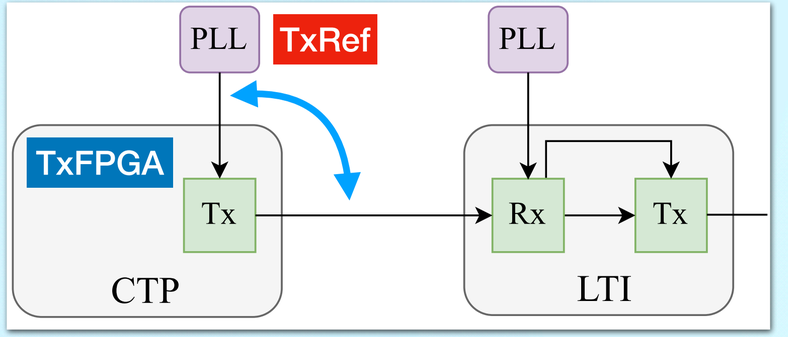


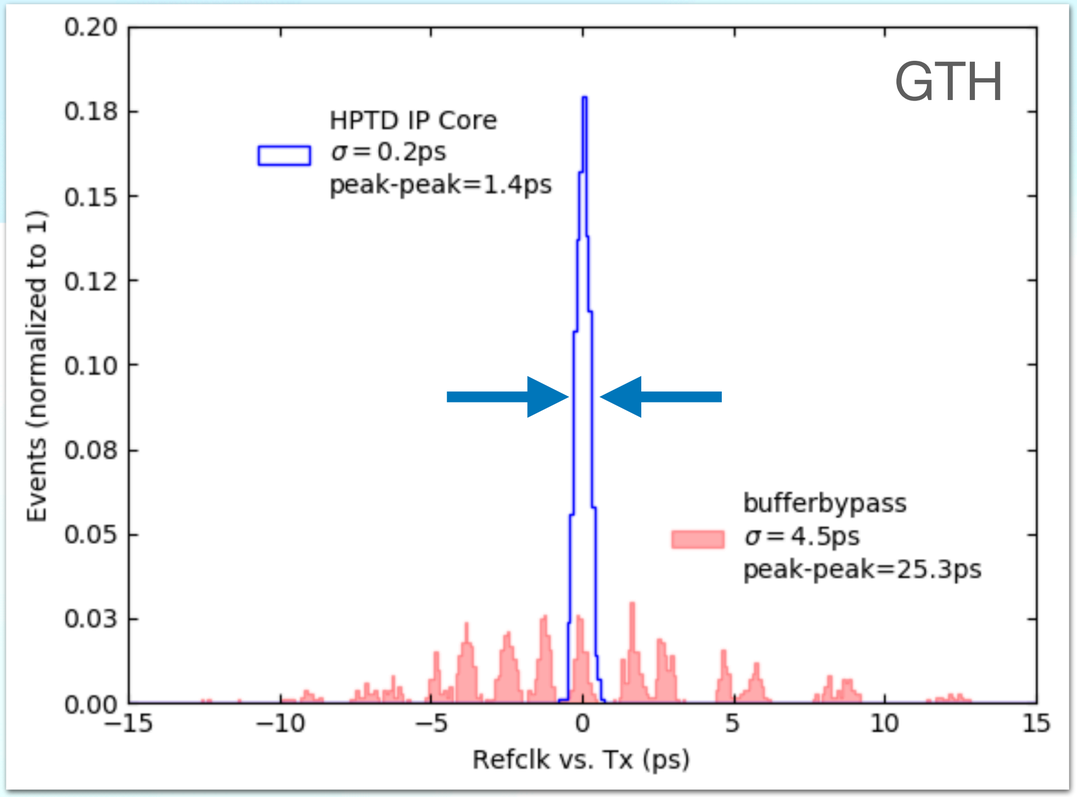
- A digital counter is used to count between the pulses corresponding to the edges of the two deglitchers
- **Averages** are necessary to mitigate **metastability** and jitter
- The conversion in time is given by the following function

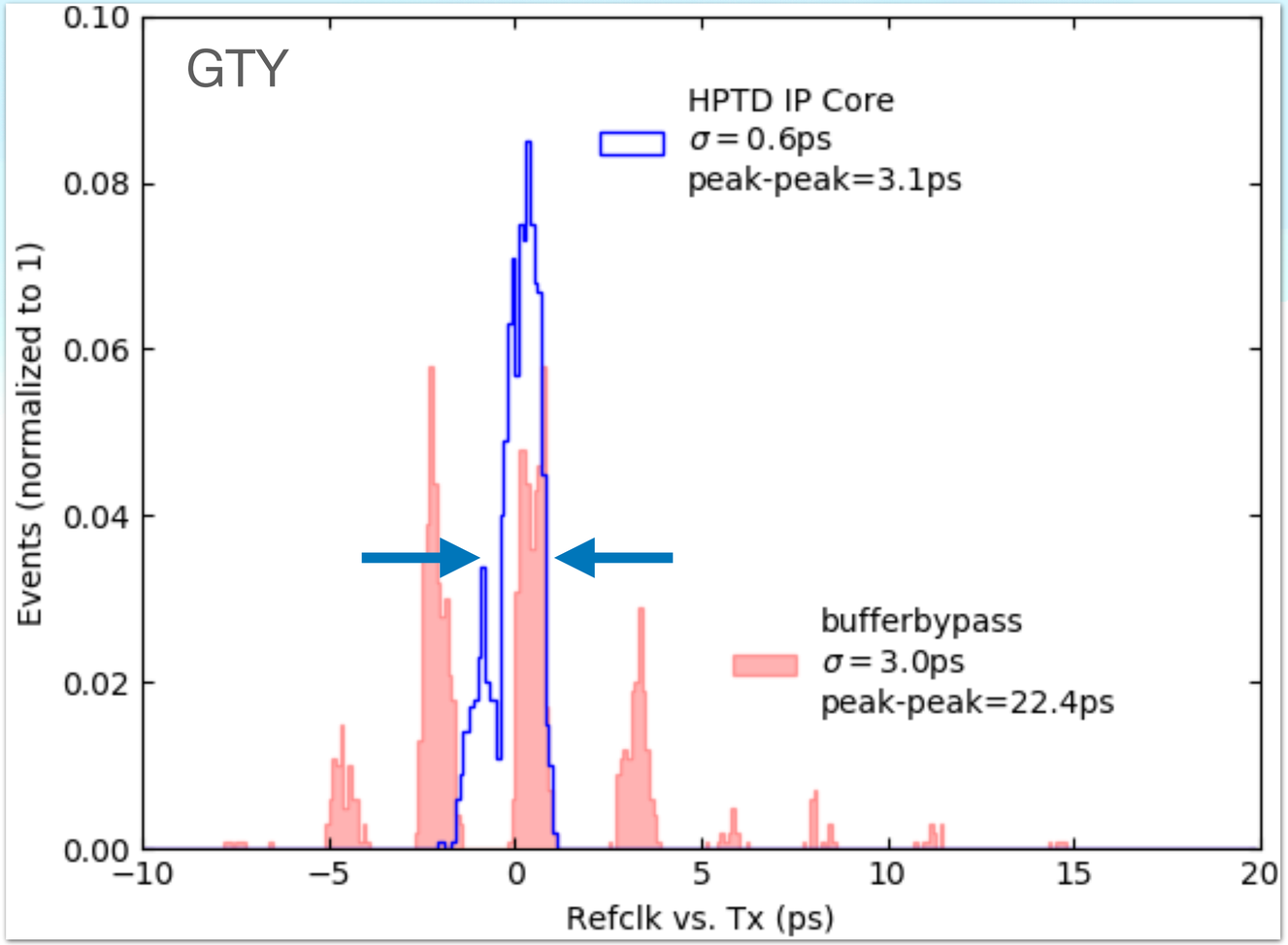
$$phase\_ps = counter\_n * 10^{12} * \frac{f_{in} - f_{dmtd}}{f_{in} f_{dmtd}}$$

$$phase = counter\_n * T_{dmtd} \frac{T_{in}}{T_q}$$

Other techniques
HPTD IP Core & BufferBypass

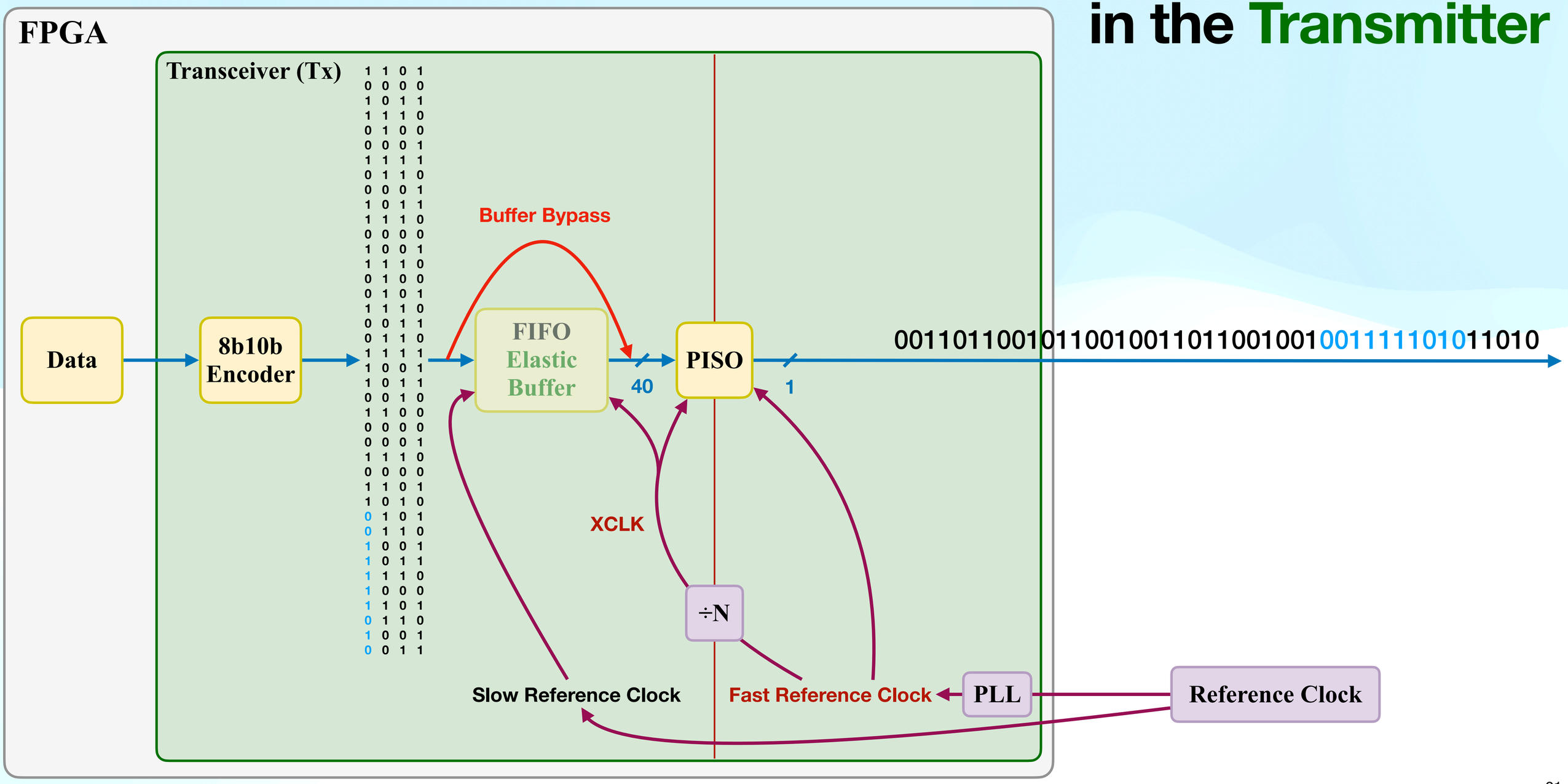




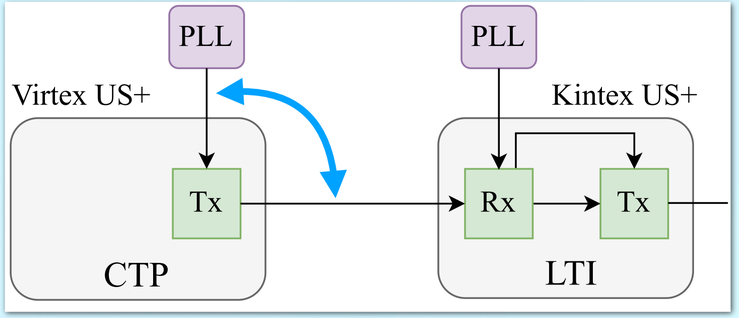


Measurements from Eduardo Mendes.

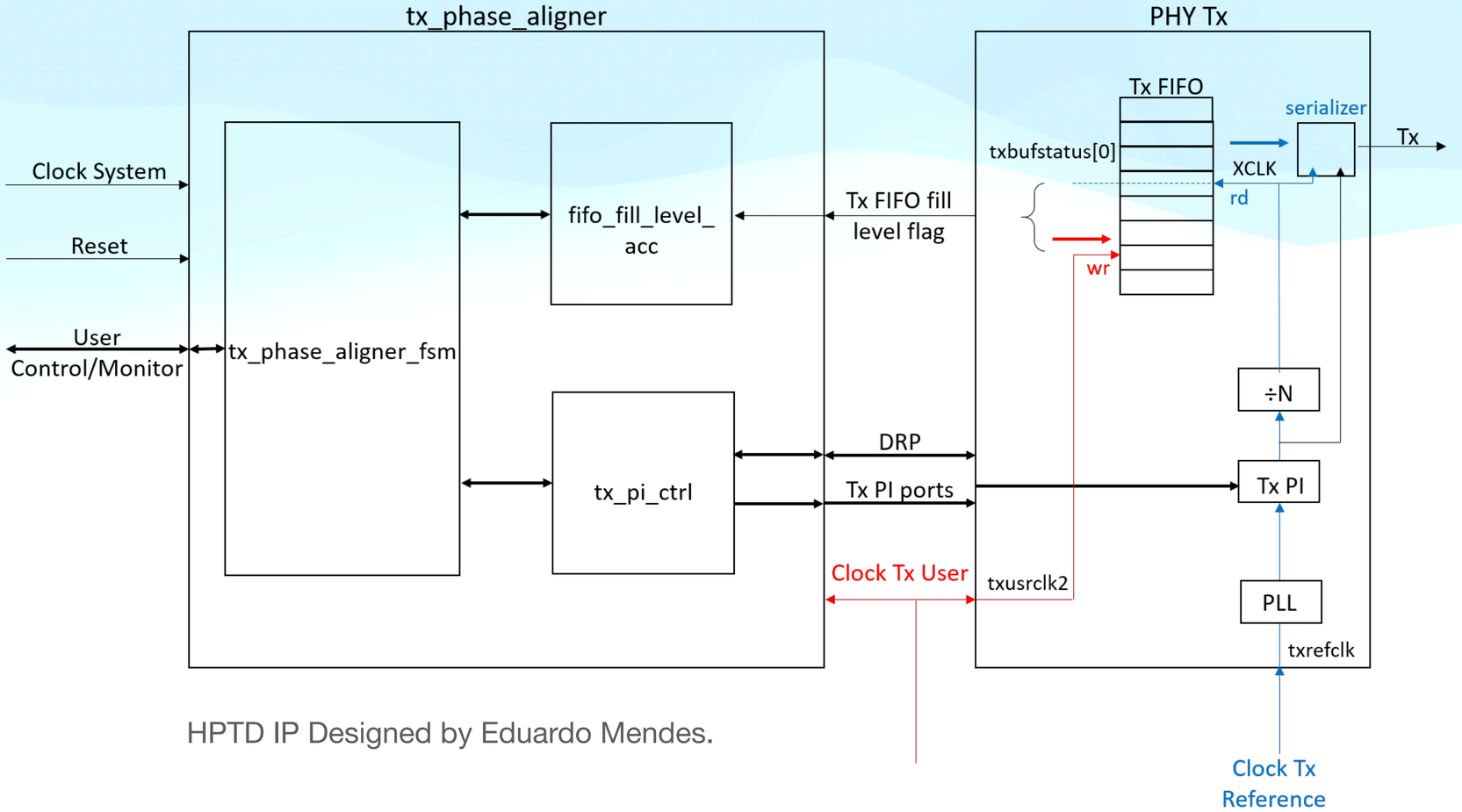
Phase Alignment in the Transmitter



HPTD IP Core



- Tx resets causes the transmitted data to change the alignment with the reference clock
- No aligner is present in the transmitter IP core
- *tx_phase_aligner* is a soft VHDL core to perform the alignment exploiting *txbufstatus* flag and Elastic Buffer



HPTD IP Designed by Eduardo Mendes.

HPTD IP Core: https://gitlab.cern.ch/HPTD/tx_phase_aligner

Deterministic solutions:	Proposed UI Aligner	HPTD IP Core	Buffer Bypass
Determinism	**2ps**	**2ps**	**30ps**
Resources	1 DDMTD DRP Controller GT's TxPI	FSM for alignment DRP Controller GT's TxPI	/
Requirements	Access to XCLK	Access to txbufstatus flag	Buffer Bypass setting

My phase alignment project: https://gitlab.cern.ch/eorzes/tx-ui-aligner/-/tree/master