# WR technology

23rd March 2023

SAFRAN

ELECTRONICS & DEFENSE

# Fermilab Workshop

23rd March 2023

SAFRAN

# Agenda

01

General considerations

02

Phase measurement

03

Protocol explanation

01

# General considerations

SAFRAN

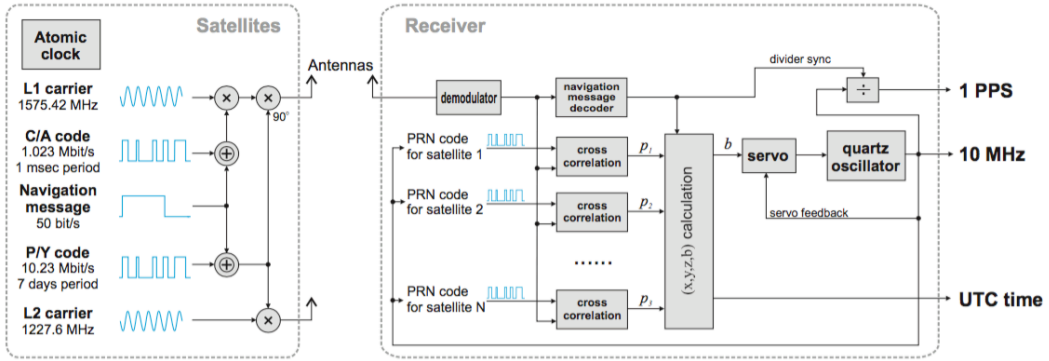
# Time reference

## Atomic clocks count time relying on the resonance frequency of atoms excited by microwave, optical or ultraviolet radiation

- They are one of the most stable time references available
- The most widespread atomic clocks are based on cesium or rubidium

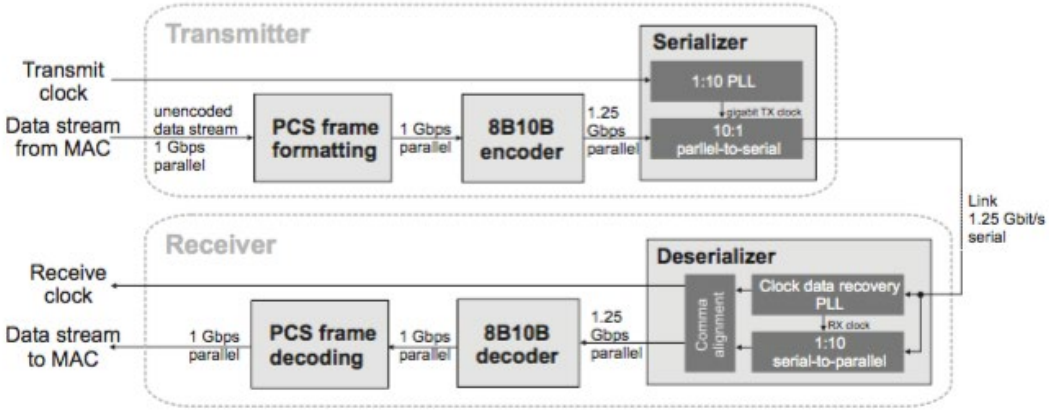
## Any GPS receiver gets an accurate time, traceable to the atomic clocks present inside GPS satellites, as it is necessary to calculate the receiver position

- GPS-synchronized time receivers just ignore the position information and adjust a local tunable oscillator according to the data obtained from the satellites



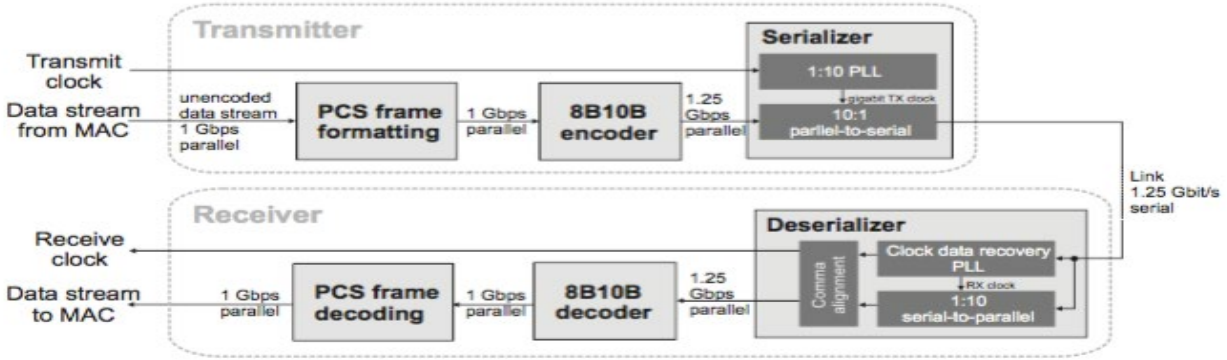
# Ethernet consideration

- **The use of fiber makes encoding easier, as the optical medium does not suffer from crosstalk or electric interference**
- **1.25 Gbps serial data stream comes out of the serializer**
- **Fully deterministic physical layer**
  - New FPGA family require complex configuration to support the new serializer and remove any source of uncertainty



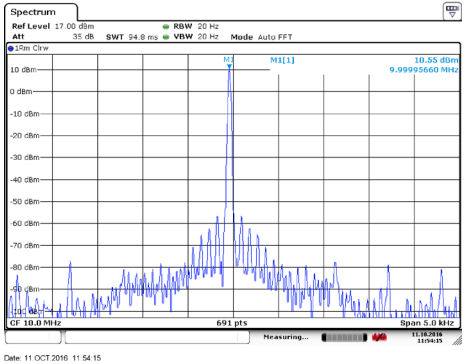
# Optical Gigabit Ethernet low level encoding

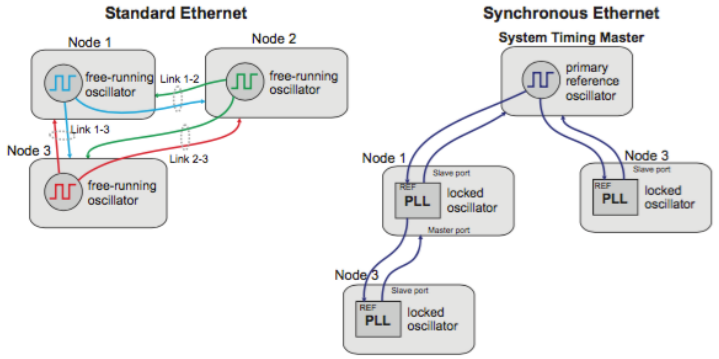
- Encapsulation of Ethernet frames in the PCS: preambles, start/end, idle pattern, other control sequences
- There must always be some pattern in the medium
- After PCS, data is fed to an 8B/10B encoder: no more than 5 subsequent equal bits, no DC component, comma symbols
- 1.25 Gbps serial datastream comes out of the serializer
- To maximize measurement accuracy, packet timestamping is done by the PCS module



# Sync-E

- In a regular Ethernet network, every node uses its own free running oscillator. Small differences of frequency between tx and rx circuits are compensated by asynchronous packet buffers (not deterministic/supported by WR)
- Sync-E defines a hierarchical structure where the master at the top is connected to a primary clock
- STM syntonizes its oscillator to the primary clock and uses this frequency to encode the data
- At the receiver end, the same frequency is recovered using PLLs and it is used with lower nodes in the hierarchy and back to the master



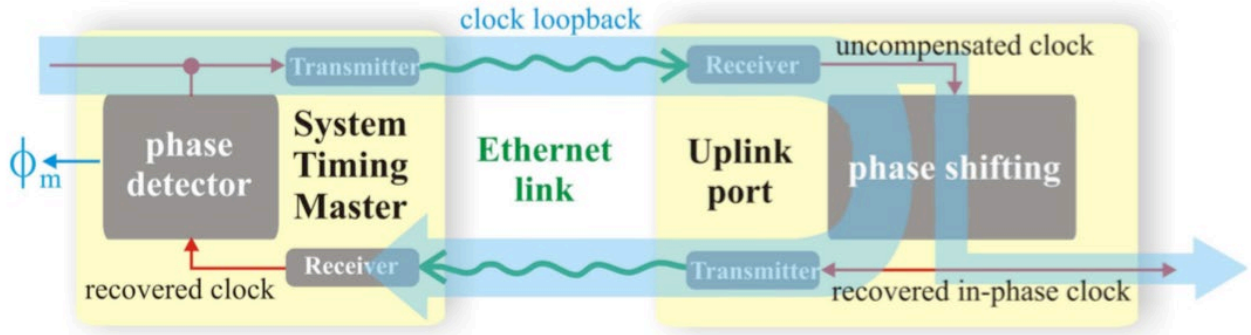



02

# Phase measurement

# Phase measurement

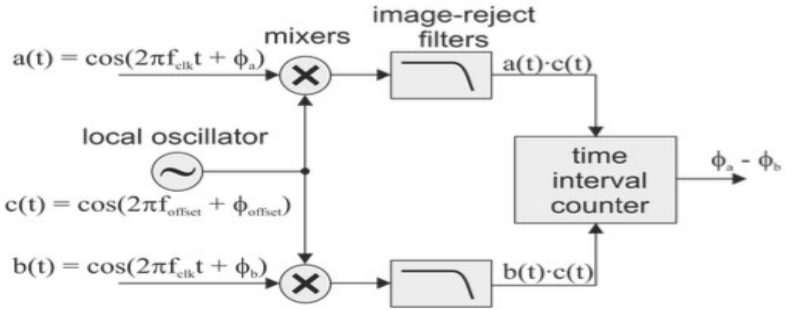
- Ethernet packets can be received at a device at any moment
- The 125 MHz transmission clock only provides 8 ns granularity
- Phase offset measurements improve the accuracy of PTP timestamp exchanges
- How can we add this sub-clock cycle timestamp capability?



# Phase measurement: theoretical background

- **First approach when measuring phase difference:**
  - A digital counter? Frequency needed would be too high (+10 GHz for subns accuracy!)
- **The Dual Mixer Time Difference is a more convenient way of measuring phase differences between two signals if they are synthonous.**
- **For the sake of simplicity, we will have a look at the analog version of the system:**

- a(t) can be our local oscillator clock
- b(t) is the reference we get from the master through the Rx Ethernet path
- An intermediate frequency offset is used to shift the frequency spectrum



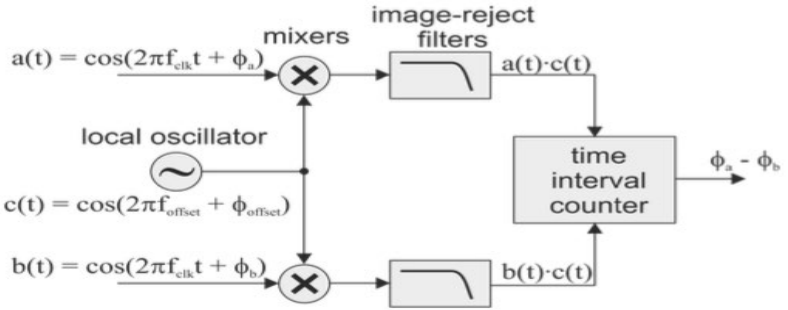

**DMTD phase detector**

# Phase measurement: theoretical background

- Comprised of a local oscillator, two identical mixers and low-pass filters and a time interval counter
- Two clocks ( a(t), b(t) ) of the same frequency (fclk) and arbitrary phases (Φa, Φb) as inputs
- The local oscillator must have a frequency offset very close to fclk
- Multiplication of input signals with the local oscillator is:

$$
\begin{aligned}
a(t)\cdot c(t) &= cos(2\pi t f_{clk}+\phi_a)\cdot cos(2\pi t f_{offset}+\phi_{offset}) \\
&= \frac{1}{2}cos(2\pi t(f_{clk}+f_{offset})+\phi_a+\phi_{offset}) \\
&+ \frac{1}{2}cos(2\pi t(f_{clk}-f_{offset})+\phi_a-\phi_{offset})
\end{aligned}
$$

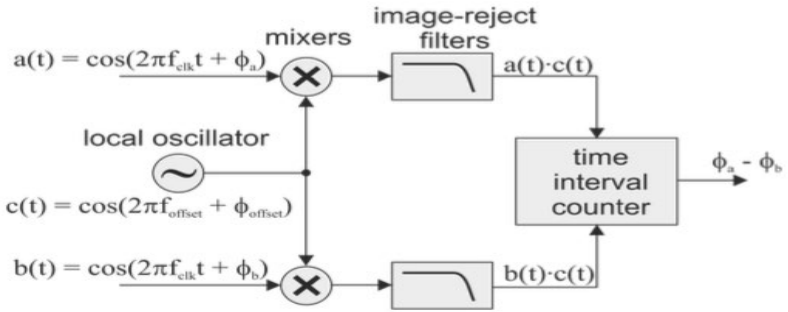
LPF-filtered




**DMTD phase detector**

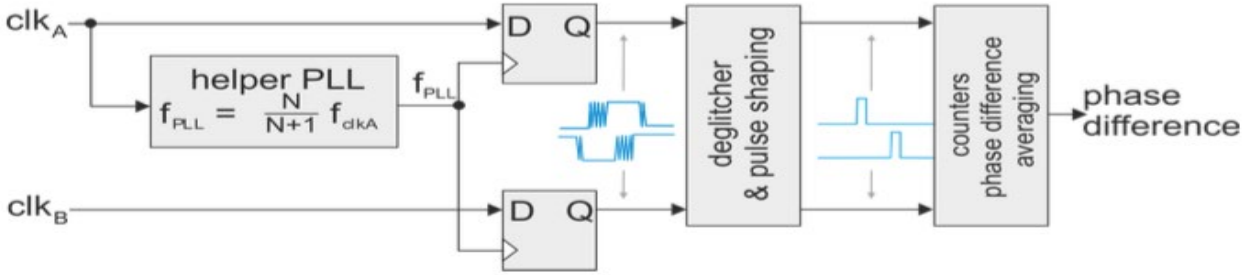
# Phase measurement: theoretical background

- **At the end of the filters, we have:**
  - cos(2πt(fclk – foffset) + Φa - Φoffset) / 2
  - cos(2πt(fclk – foffset) + Φb - Φoffset)
- **The phase difference between the clocks is unaltered**
- **Suppose fclk=125 MHz and offset=124.99 MHz**
  - The phase shift can be easily measured from a 10 kHz output signal (100 us period) can be measured using an affordable  counter with a 10 ns clock period



# Phase measurement: theoretical background

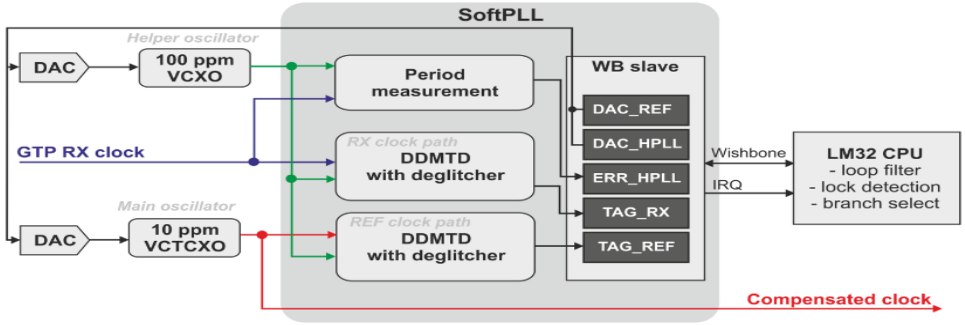
- $f_{PLL}$ generated from input clocks with a frequency few kHz away from received frequency
- The replacement of the mixed with the D-type flip-flop generate glitches due to setup/hold violation: deglitching filter is required prior counting cycles to determine the phase differences




White-Rabbit has developed a digital version of this system (Digital DMTD or DDMTD)
Low pass filtering is implemented using D-type flip-flop

# Phase measurement: theoretical background

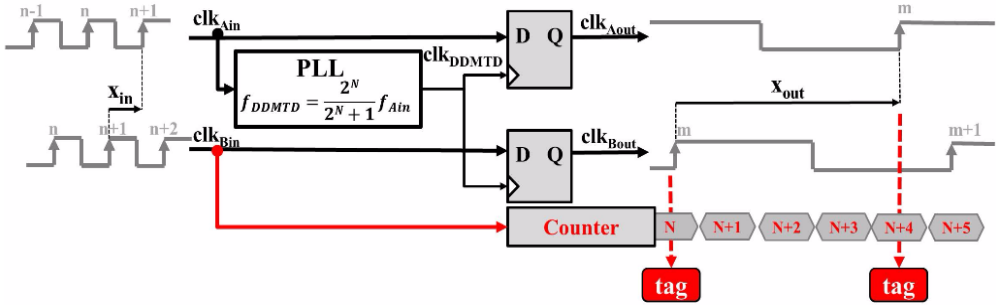
- The Soft PLL performs the syntonization of the local reference clock to the RX clock received from the data stream
- The firmware module contains the minimum functionalities and most of the control loop implementation is in the SW
- Here are defined the Digital DMTDs to obtain the phase differences between clocks
- The Soft PLL module uses two external VCXO oscillators tuned by DACs




**DMTD phase detector: SoftPLL implementation**

# Phase measurement: theoretical background

- **Input:**
  - frequency $f_{in}$ [Hz]
  - phase $\varphi_{in}$ [rad]
- **Output:**
  - proportional lower frequency $f_{out}$ [Hz]
  - equal phase: $\varphi_{in}$ [rad]= $\varphi_{out}$ [rad]
- **Zooming effect:**
  - $x_{in}[ns] = 1/(1+2\text{^}N) \cdot x_{out}[ns]$




**DMTD phase detector: SoftPLL implementation**

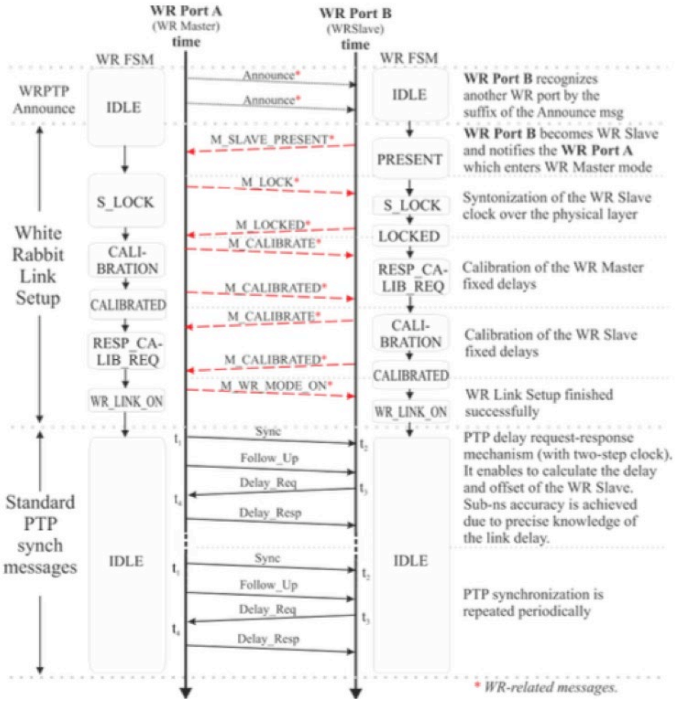
03

# Protocol explanation

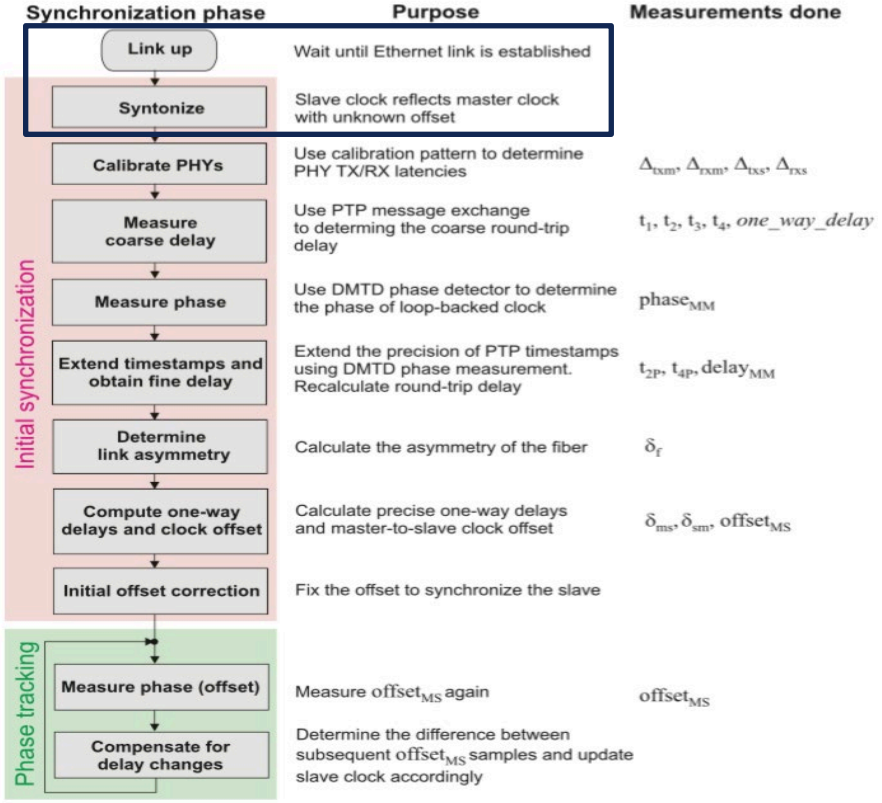
SAFRAN

# WR: Timing protocol

- Link up
- Syntonize
- Calibrate PHYs
- Measure coarse delay
- Measure phase
- Extend timestamps and obtain fine delay
- Determine link asymmetry
- Compute one-way delays and clock offset
- Initial offset correction
- Measure phase
- Compensate for delay changes

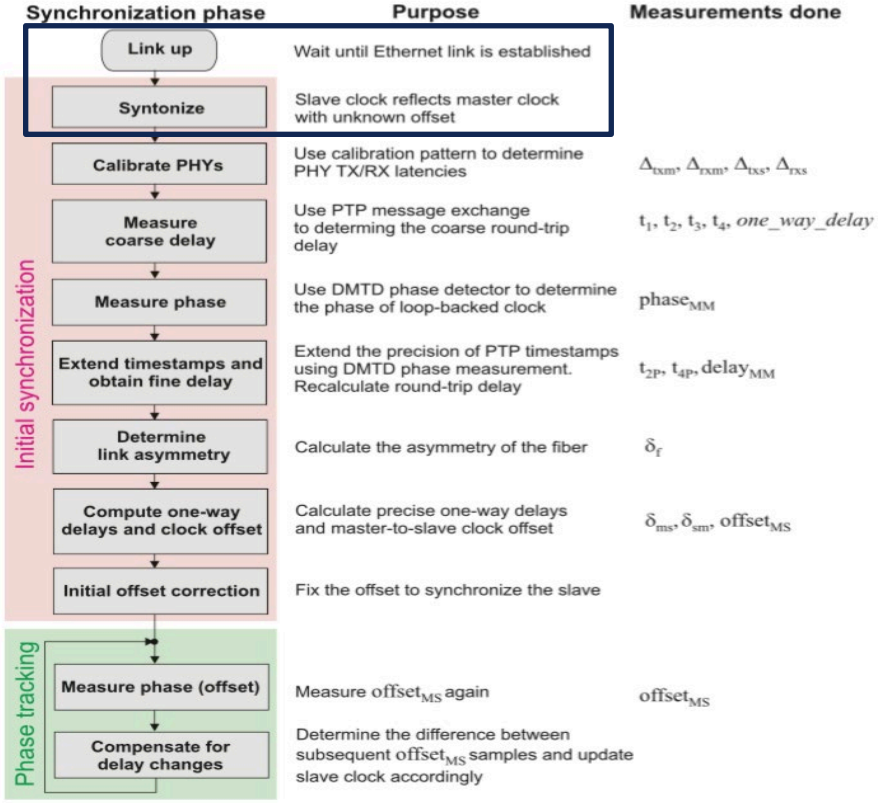


# WR protocol: Link detection and syntonization




- Initially, the link between master and slave is not established
- Their PHYs send the idle pattern but they do not receive any meaningful data
- When they align the symbols and receive the idle pattern, the Ethernet link will become active
- Following an active link, master sends ANNOUNCE messages and eventually the slave will answer with a SLAVE_PRESENT
- When the master sends a LOCK message, the slave will start to recover the clock from the data stream
- Once the PLL is locked, the slave issues a LOCKED message and both nodes are syntonized (same frequency, different phase)

# WR protocol: Link delay measurement



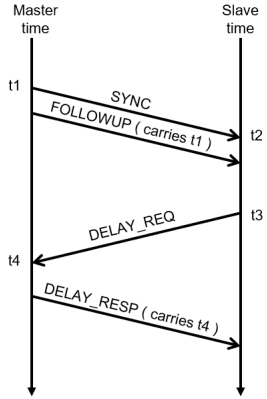

- **Once syntonized, we still have to synchronize both nodes. This can be divided into two tasks:**
  - **Coarse** delay measurement, based on a PTP exchange
  - **Precise** delay measurement that combines the coarse delay with a DDMTD phase measurement

# WR protocol: Coarse delay measurement

| Synchronization phase | Purpose | Measurements done |
|---|---|---|
| Link up | Wait until Ethernet link is established | |
| Syntonize | Slave clock reflects master clock with unknown offset | |
| Calibrate PHYs | Use calibration pattern to determine PHY TX/RX latencies | $\Delta_{txm}$, $\Delta_{rxm}$, $\Delta_{txs}$, $\Delta_{rxs}$ |
| Measure coarse delay | Use PTP message exchange to determing the coarse round-trip delay | $t_1$, $t_2$, $t_3$, $t_4$, *one_way_delay* |
| Measure phase | Use DMTD phase detector to determine the phase of loop-backed clock | $phase_{MM}$ |
| Extend timestamps and obtain fine delay | Extend the precision of PTP timestamps using DMTD phase measurement. Recalculate round-trip delay | $t_{2P}$, $t_{4P}$, $delay_{MM}$ |
| Determine link asymmetry | Calculate the asymmetry of the fiber | $\delta_f$ |
| Compute one-way delays and clock offset | Calculate precise one-way delays and master-to-slave clock offset | $\delta_{ms}$, $\delta_{sm}$, $offset_{MS}$ |
| Initial offset correction | Fix the offset to synchronize the slave | |
| Measure phase (offset) | Measure $offset_{MS}$ again | $offset_{MS}$ |
| Compensate for delay changes | Determine the difference between subsequent $offset_{MS}$ samples and update slave clock accordingly | |

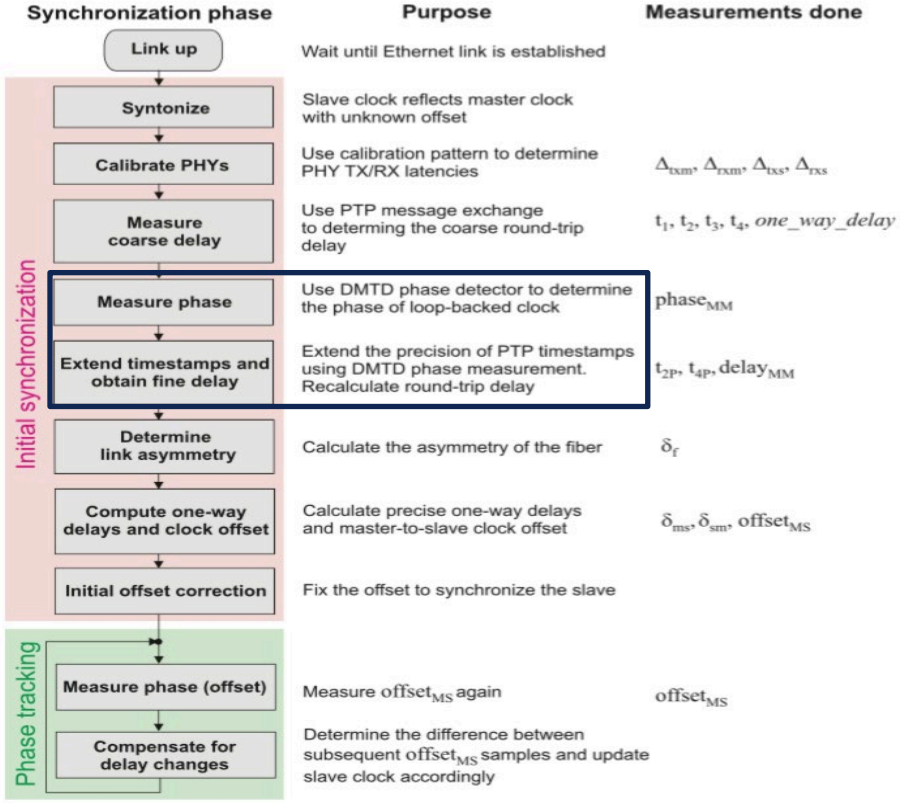
Initial synchronization

Phase tracking



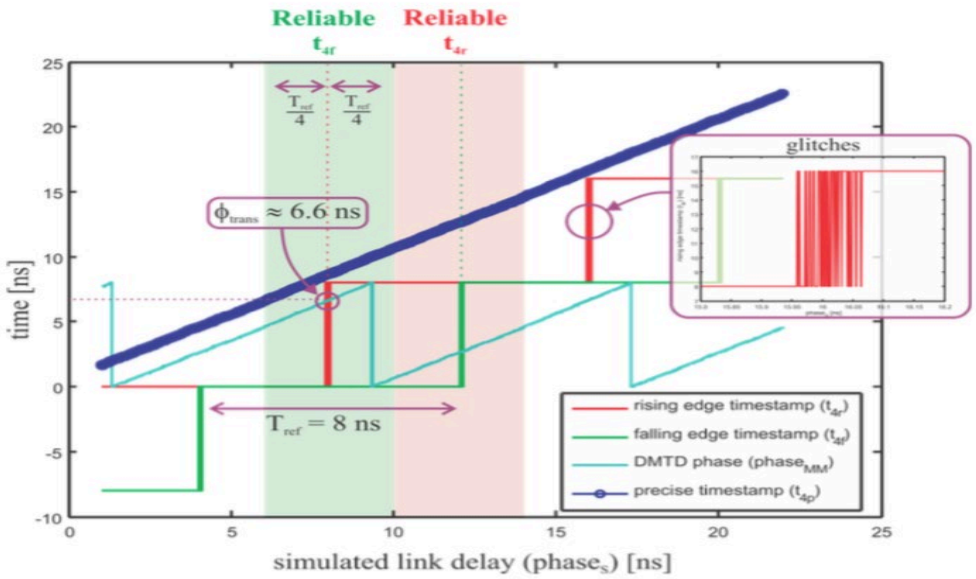

- The coarse delay is measured using a PTPv2 two-step packet exchange
- The packets timestamps are hardware-generated
- This measurement produces timestamp values: $t_1$, $t_2$, $t_3$, $t_4$
- Due to the possibility of jitter-related problems, $t_2$ and $t_4$ timestamps are generated for both rising and falling edge

# WR protocol: Fine delay measurement



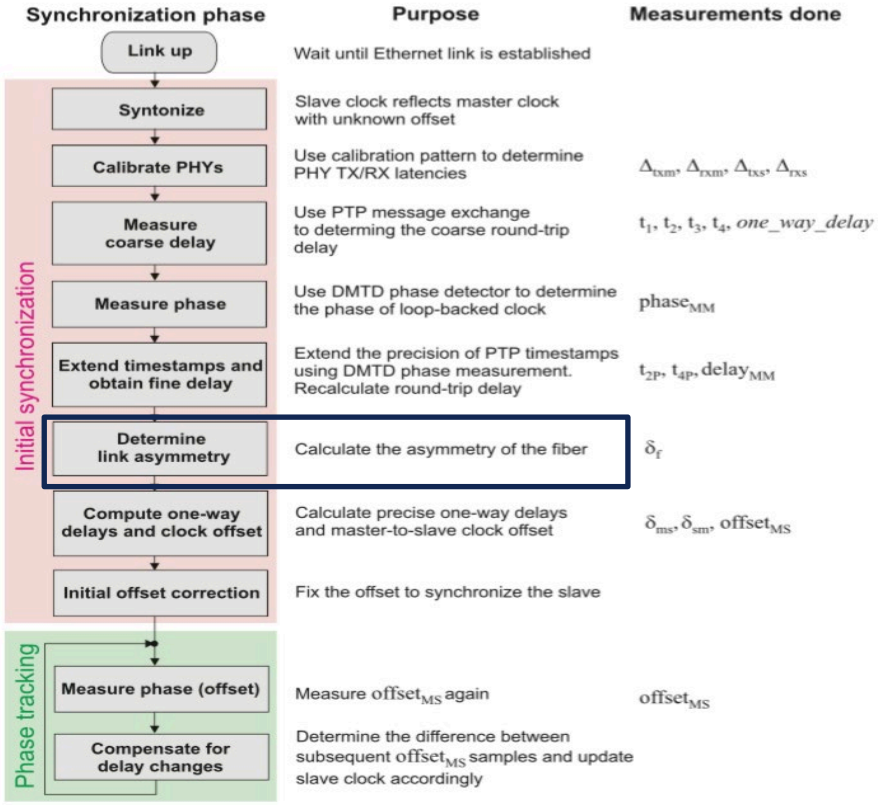

- **At this point we have:**
  - $t_1$, $t_2$, $t_3$, $t_4$ PTP timestamps
  - $phase_{MM}$ round-trip phase-shift
  - $phase_S$ slave setpoint
- **The precision of the timestamps is extended to include the DDMTD measurements: $t_{2p}$, $t_{4p}$ (not needed for $t_1$ & $t_2$ because transmission always use the device reference clock – phase = 0)**
- **An algorithm decides whether the rising edge or falling edge timestamps are the reliable ones**
- **Some magic: because clocks are syntonized, phase relationship are constant and therefore they can be integrated many times (hundreds) to improve accuracy in the phase values**
  - Note that Start-of-Frame (SoF) signals share a sub-cycle timestamp offset equal to the inter-frequency phase offset (and therefore timestamps can be improved using the corresponding DDMTD phase measurements) for correction

# WR protocol: Fine delay measurement




- **At this point we have:**
  - $t_1$, $t_2$, $t_3$, $t_4$ PTP timestamps
  - phase$_{MM}$ round-trip phase-shift
  - phase$_S$ slave setpoint
- **The precision of the timestamps is extended to include the DDMTD measurements: $t_{2p}$, $t_{4p}$**
- **An algorithm decides whether the rising edge or falling edge timestamps are the reliable ones**
- **And precise round-trip delay can be calculated as**
  - $delay_{MM} = ( t_{4p} - t_1 ) - ( t_3 - t_{2p} )$
- **More advanced: $\Phi_{trans}$ is a constant parameter for a given device**
- **$\Phi$ = phase$_{MM}$ - $\Phi_{trans}$**

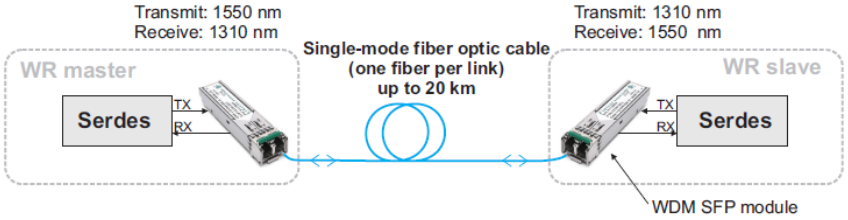
# WR protocol: Link asymmetry




- **The asymmetry cannot be directly measured. It can only be estimated from $delay_{MM}$ and knowledge of the medium and the transmission circuits**
- **All these asymmetry sources are taken into account:**
  - Propagation delays of electronic components and PCB traces
  - Optical transceivers delay asymmetry
  - Fiber Rx/Tx different diffraction index
  - Internals of the chips structure
- **Fiber asymmetry can be variable depending on operating conditions and link length. All the rest are considered constant per device**

# WR protocol: Link asymmetry

- **Circuit asymmetries**
  - Different length in traces
  - Different structure of internal clock distribution
  - Different paths in FPGA logic
  - Solutions:
    - Measure them in lab and consider them to be constant
    - Develop a model that characterizes asymmetries depending on conditions
    - Reduce asymmetries as much as possible in design stage




- **Fiber asymmetries**
  - Chromatic dispersion due to different wavelengths used
  - Characterized by α parameter
- **Transceivers asymmetry**
  - Serializers/De-serializers introduce a random latency between the recovered clock and the data stream
  - Solution: disabling the automatic comma alignment and manual bit-shifting

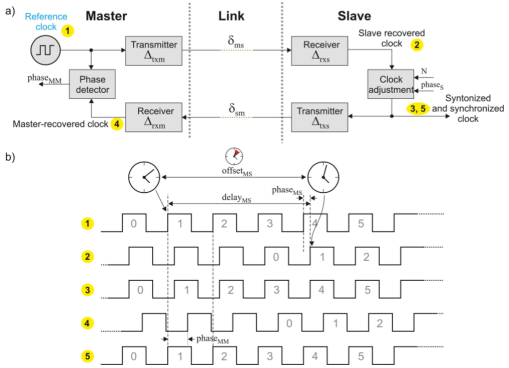
# WR protocol: Computing one-way delay and offset

| Synchronization phase | Purpose | Measurements done |
|---|---|---|
| Link up | Wait until Ethernet link is established | |
| Syntonize | Slave clock reflects master clock with unknown offset | |
| Calibrate PHYs | Use calibration pattern to determine PHY TX/RX latencies | $\Delta_{txm}$, $\Delta_{rxm}$, $\Delta_{txs}$, $\Delta_{rxs}$ |
| Measure coarse delay | Use PTP message exchange to determing the coarse round-trip delay | $t_1$, $t_2$, $t_3$, $t_4$, *one_way_delay* |
| Measure phase | Use DMTD phase detector to determine the phase of loop-backed clock | $phase_{MM}$ |
| Extend timestamps and obtain fine delay | Extend the precision of PTP timestamps using DMTD phase measurement. Recalculate round-trip delay | $t_{2P}$, $t_{4P}$, $delay_{MM}$ |
| Determine link asymmetry | Calculate the asymmetry of the fiber | $\delta_f$ |
| Compute one-way delays and clock offset | Calculate precise one-way delays and master-to-slave clock offset | $\delta_{ms}$, $\delta_{sm}$, $offset_{MS}$ |
| Initial offset correction | Fix the offset to synchronize the slave | |
| Measure phase (offset) | Measure $offset_{MS}$ again | $offset_{MS}$ |
| Compensate for delay changes | Determine the difference between subsequent $offset_{MS}$ samples and update slave clock accordingly | |

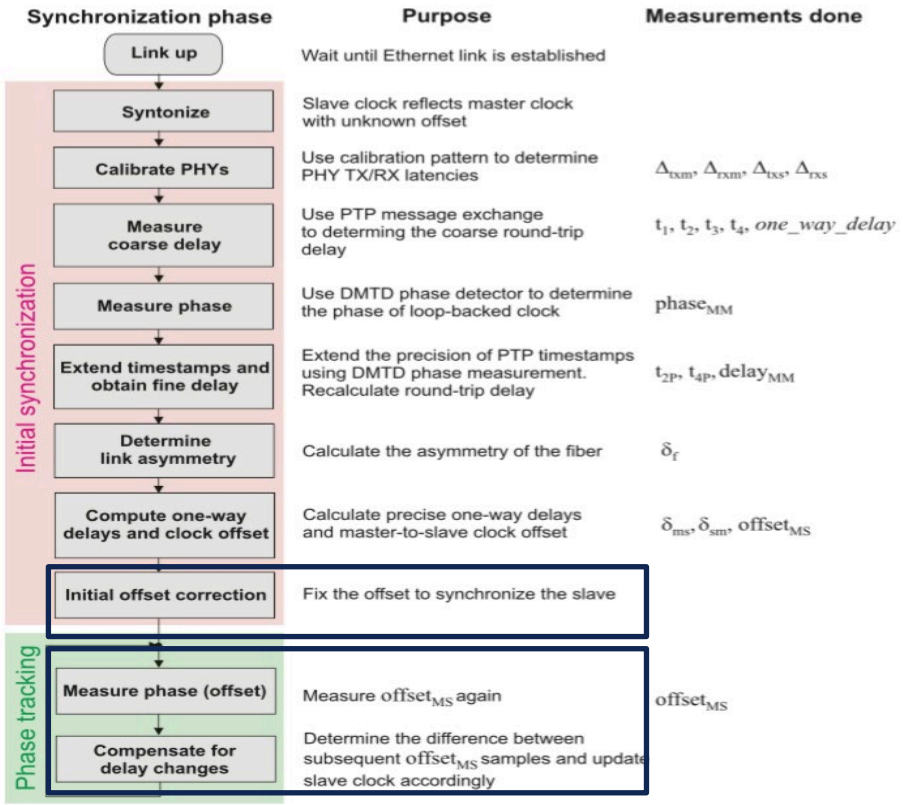
Initial synchronization

Phase tracking

- $\Delta = \Delta_{txm} + \Delta_{rxm} + \Delta_{txs} + \Delta_{rxs}$
- $delay_{MM} = \Delta + \delta_{ms} + \delta_{sm}$
- $\alpha = (\delta_{ms} / \delta_{sm}) - 1$
- **We obtain one-way fiber delay $\delta_{ms}$ from these equations as:**
  - $\delta_{ms} = (1+\alpha)(delay_{MM}-\Delta) / (2+\alpha)$
- **and adding the asymmetric delays gives the final products:**
  - $delay_{ms} = (1+\alpha)(delay_{MM}-\Delta) / (2+\alpha) + \Delta_{txm} + \Delta_{rxs}$
  - $offset_{ms} = t_1 - t_{2p} - delay_{ms}$



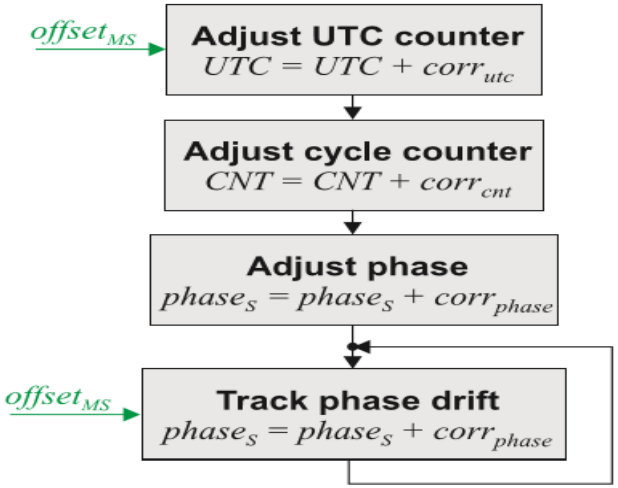
# WR protocol: Initial offset correction and phase tracking

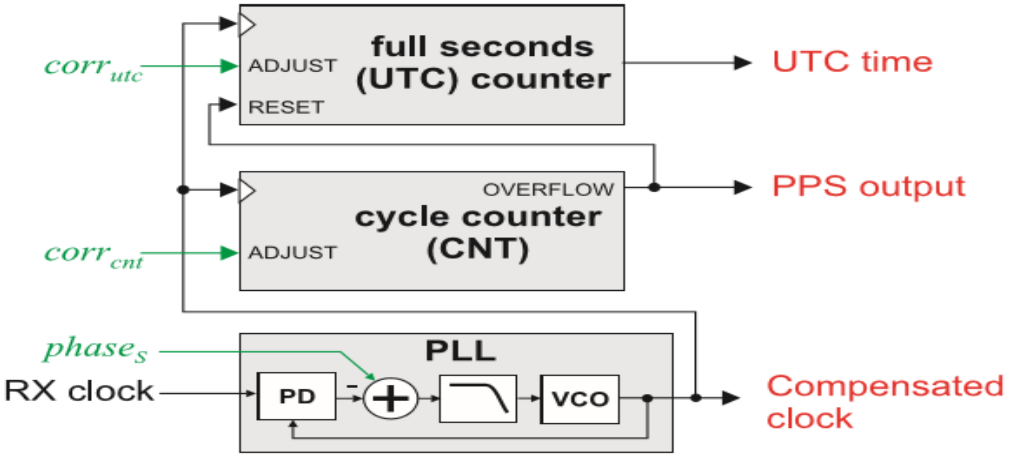



- $delay_{ms} = (1+\alpha)(delay_{MM}-\Delta) / (2+\alpha) + \Delta_{txm} + \Delta_{rxs}$)
- $offset_{ms} = t_1 - t_{2p} - delay_{ms}$
- **The second's counter is increased/decreased with the integer part of $offset_{MS}$**
- **The counter used to generate PPS signal is corrected with the remainder of $offset_{MS}$ (in full 8-ns cycles)**
- **The $phase_S$ value is corrected with the picoseconds remainder**
- **Once synchronized, the packet exchange is repeated periodically. $offset_{ms}$ is updated and the little phase changes are corrected**
- **PROBLEM: experimental errors on determining the contributions of α and Δ may significantly impact links calibration**
  - The error in α is multiplied by the distance of the link. This is not a problem with small distances, but it is for large ones

# WR protocol: Synchronization stages summary

- **The offset corrections are made on the UTC and nanosecond counter by the PPSGen and on the reference VCO by the SoftPLL**

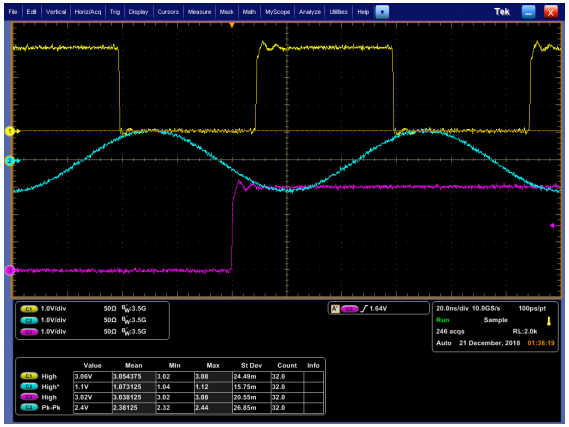


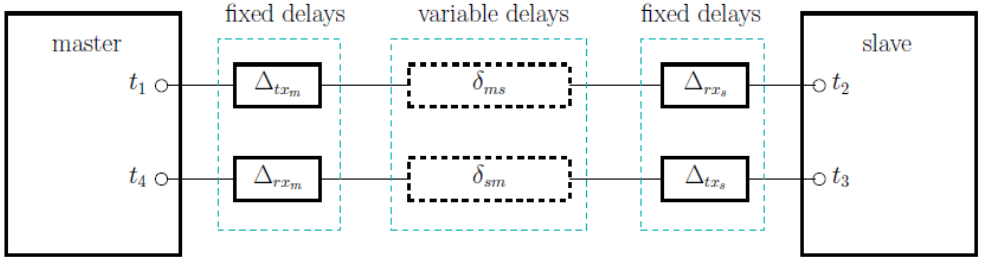



# WR protocol: Equations summary

- $delay_{MM} = ( t_{4p} - t_1 ) - ( t_3 - t_{2p} )$
- $\Delta = \Delta_{txm} + \Delta_{rxm} + \Delta_{txs} + \Delta_{rxs}$
- $delay_{MM} = \Delta + \delta_{ms} + \delta_{sm}$
- $\alpha = (\delta_{ms} / \delta_{sm}) - 1$
- $\delta_{ms} = (1+\alpha)(delay_{MM}-\Delta) / (2+\alpha)$
- $delay_{ms} = (1+\alpha)(delay_{MM}-\Delta) / (2+\alpha)$
- $+ \Delta_{txm} + \Delta_{rxs})$
- $offset_{ms} = t_1 - t_{2p} - delay_{ms}$

*PPS – 10MH phase relationship*





# POWERED BY TRUST