



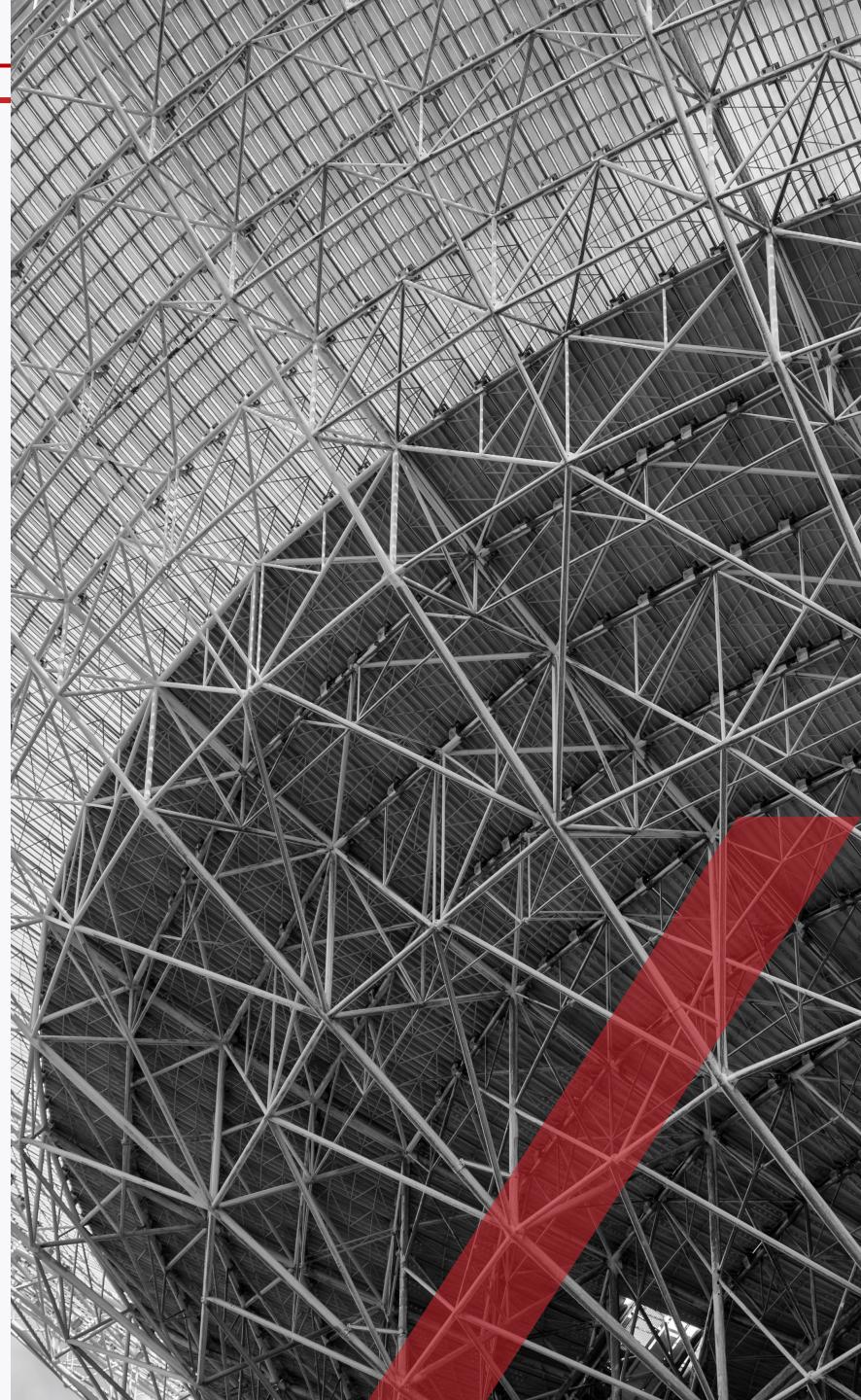
OPEN
Compute Project ®

The TAP Grandmaster

ELPROMA comments

“Better Safe Than Sorry”

Janusz Uzycki
Tomasz Widomski
© 2020



Speakers:

Agenda

- 1) Cyber-Security remarks & issues – Tomasz Widomski
- 2) Technical review & comments – Janusz Użycki

Comments to:

Linked IN (YouTube), including ITFS webinar 23/09

https://www.linkedin.com/posts/leoleovich_time-appliances-project-ocp-tap-project-activity-6704863005760348160-0nOF/

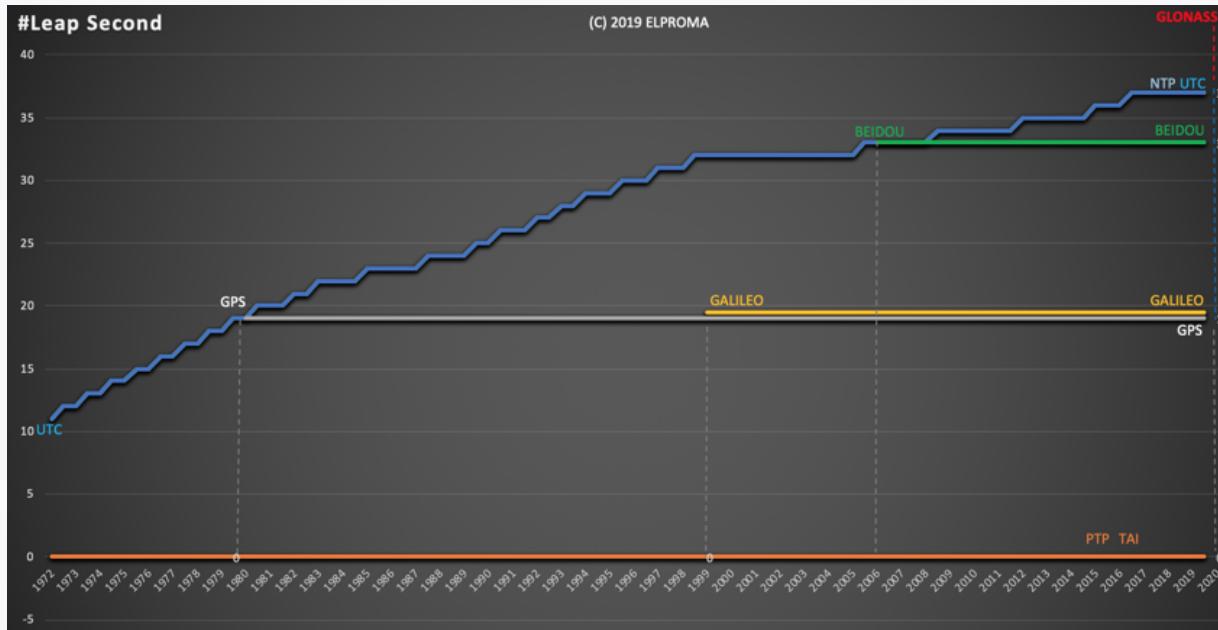
and GitHub document

<https://github.com/opencomputeproject/Time-Appliance-Project/blob/master/Open-Grand-Master-Spec-Draft.pdf> on 1/10/2020
as the reference.

The replaceable GNSS modules (Resolution-T std. I/O)

Advantages:

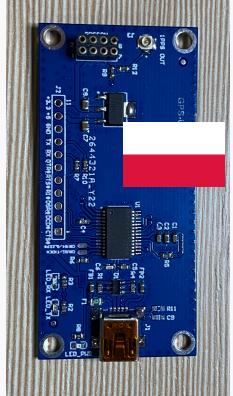
- the fast way to solve
 - * UTC leap second (*instability & incompatibility*)
 - * firmware bugs (*overflows, timescale, GPS WNRO ...*)
 - * time gaps (*13.5 µs GPS SVN23 26/01/2016*)
- easy update => IRNSS ...
- easy migration => L1+L2, L1+L5, L1+L2+L5
- jamming/spoofing => switch to OSC holdover mode
- Improves UTC –to-TAI timescale recalculation for PTP IEEE1588



Risks:

- UTC/TAI offset differences** between GNSS subsystems
 - e.g. 30-40ns (UTC) between GPS & GLONASS
 - e.g. Glonass(T) basis on UTC (via local Moscow time)
 - Beidou(T) +4s ahead to UTC, GPS +18s ahead to UTC
- Geopolitical dependency** - not ensuring ref. availability
 - GPS, GLONASS, BEIDOU, IRNSS – military nature
 - GALILEO – civil nature, but tightly related to GPS
- TAI is artificial timescale** existing at NMI LAB-level
 - PTP mostly retrieves TAI from input ref. UTC (highly risk)

Elroma contribution => releasing open hardware GNSS modules



USB DEV-KIT



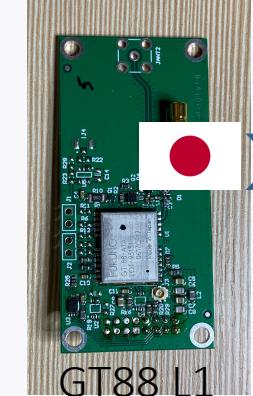
M8T L1



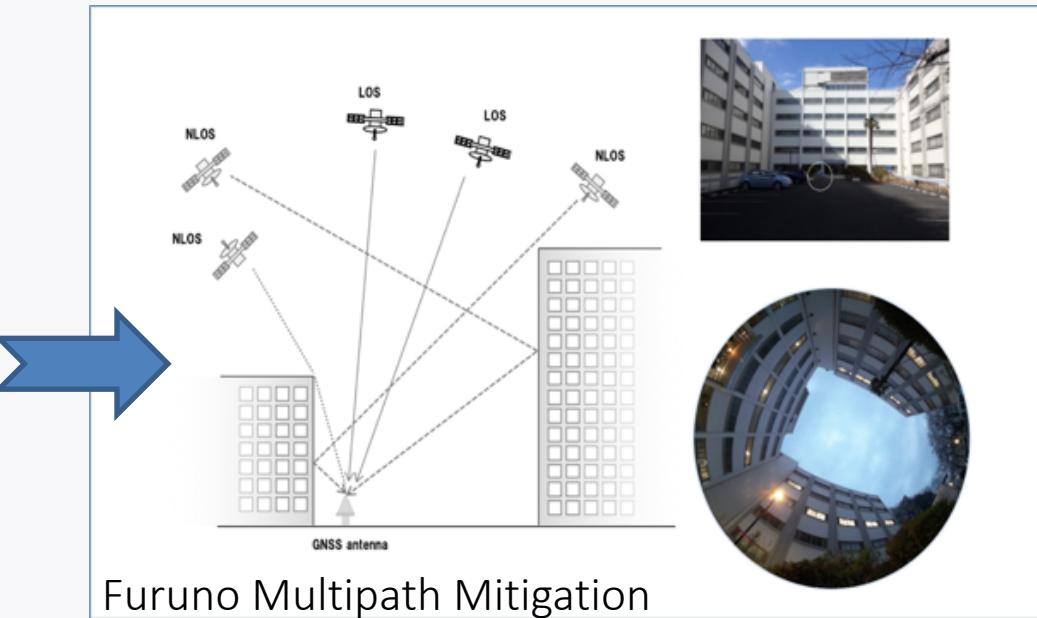
M8T L1



F9T L1+L2



GT88 L1



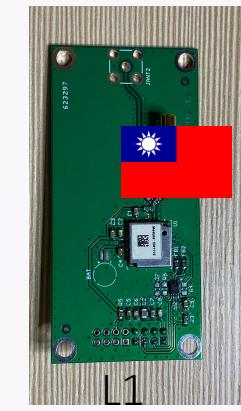
NVS



TRIMBLE L1



SKYTRAQ L1



CD-TOP L1

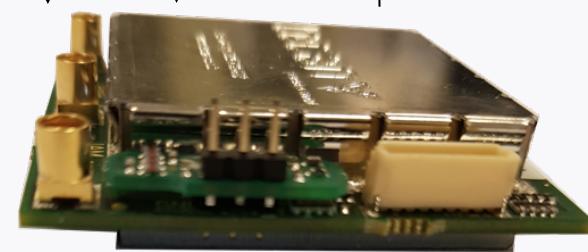


Furuno

GPS
GALILEO
GLONASS
BEIDOU
IRNSS
L1+L5

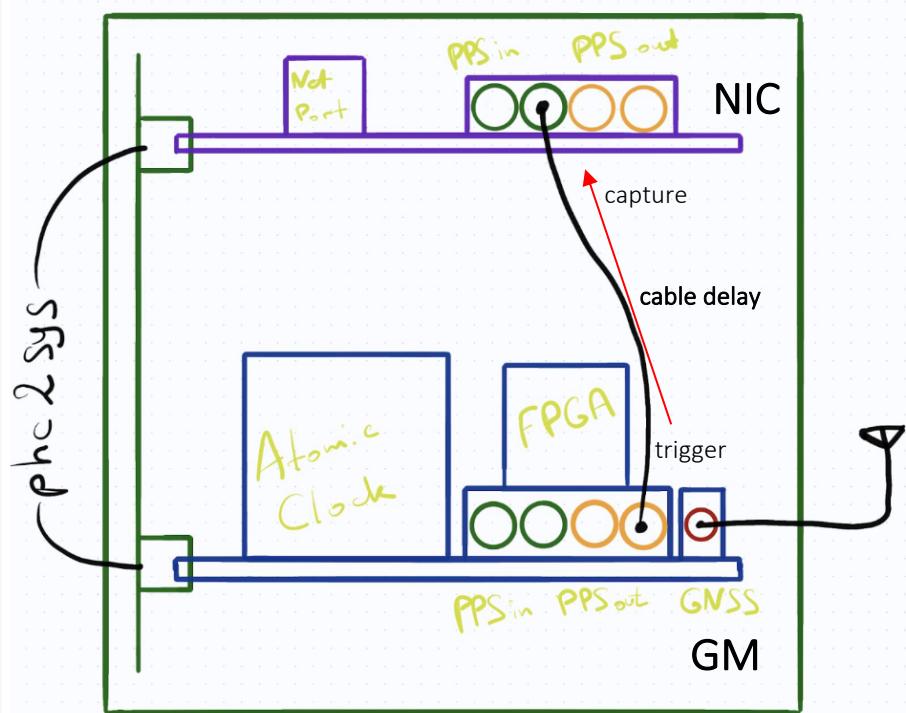
Considering Anti-Jamming/Spoofing filtering

ANT1 ANT2



RCV RF
1.5GHz (L1)



5.3. Time Card

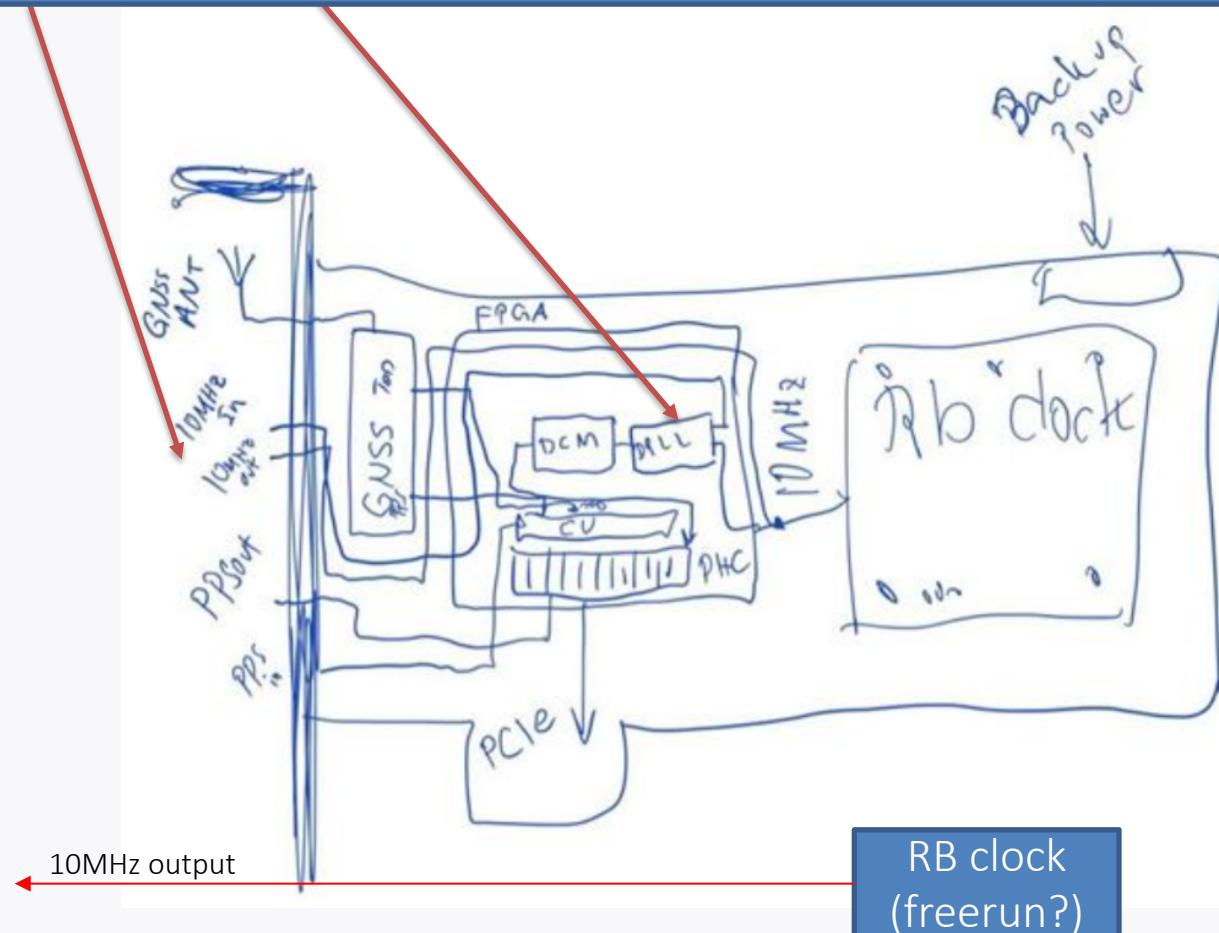
Synchronization between GM and NIC card issue.

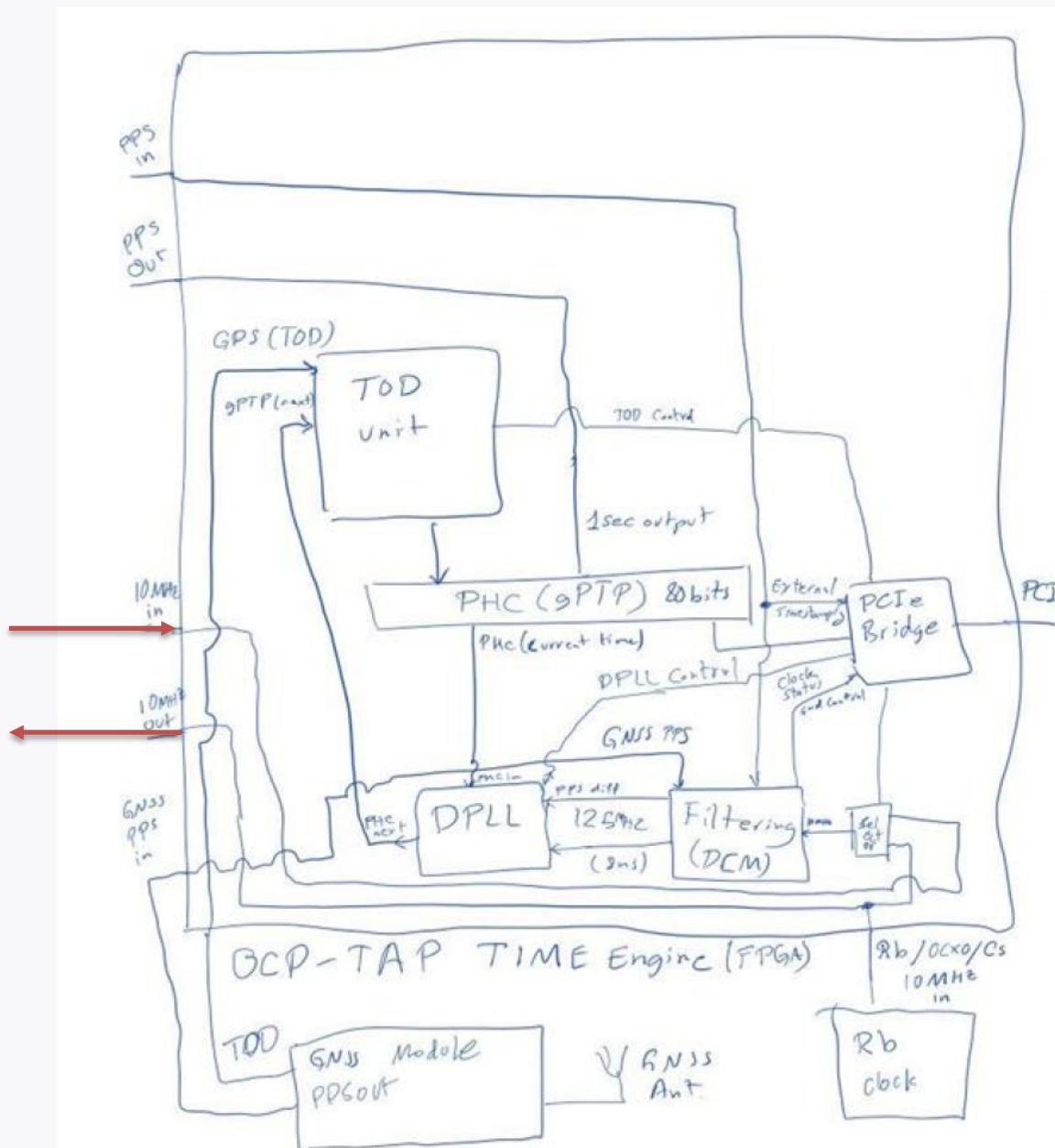
Difference between 1PPS and phc2sys with PCIe PTM conjunction:

- 1) Your NIC is likely based on **TCXO** which varies at nanosecond level significantly. So better to **adjust** the oscillator **more frequently** to follow more stable oscillator of the GM card.
- 2) **1PPS signal** allows to observe **every 1 second** with e.g. 8ns trigger resolution (GM's 1PPS output granularity). The NIC has own capture resolution what will effect in higher jitter.
- 3) Hardware assisted **PCIe PTM** and **phc2sys** allows for hardware “**cross-timestamping**” **more often** than once per second - more data to **average** (precision improved) and shorter NIC's PLL time constant possible – more smooth tracking of the GM. Single shot trigger/capture timestamp resolution are still issue but averaged. PTM compensates a delay (PCIe latency). For 1PPS signal total delay **calibration** is needed.

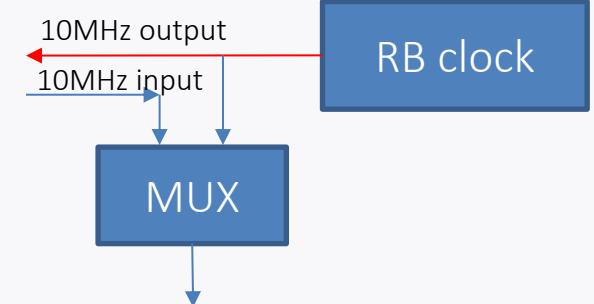
DPLL inside FPGA is more noisy than good one **external**, important when you distribute its output outside of the card.

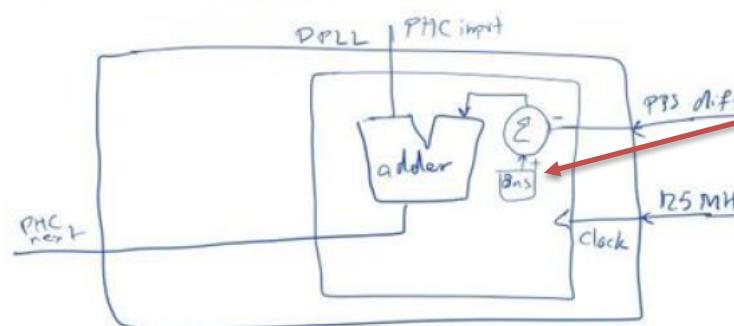
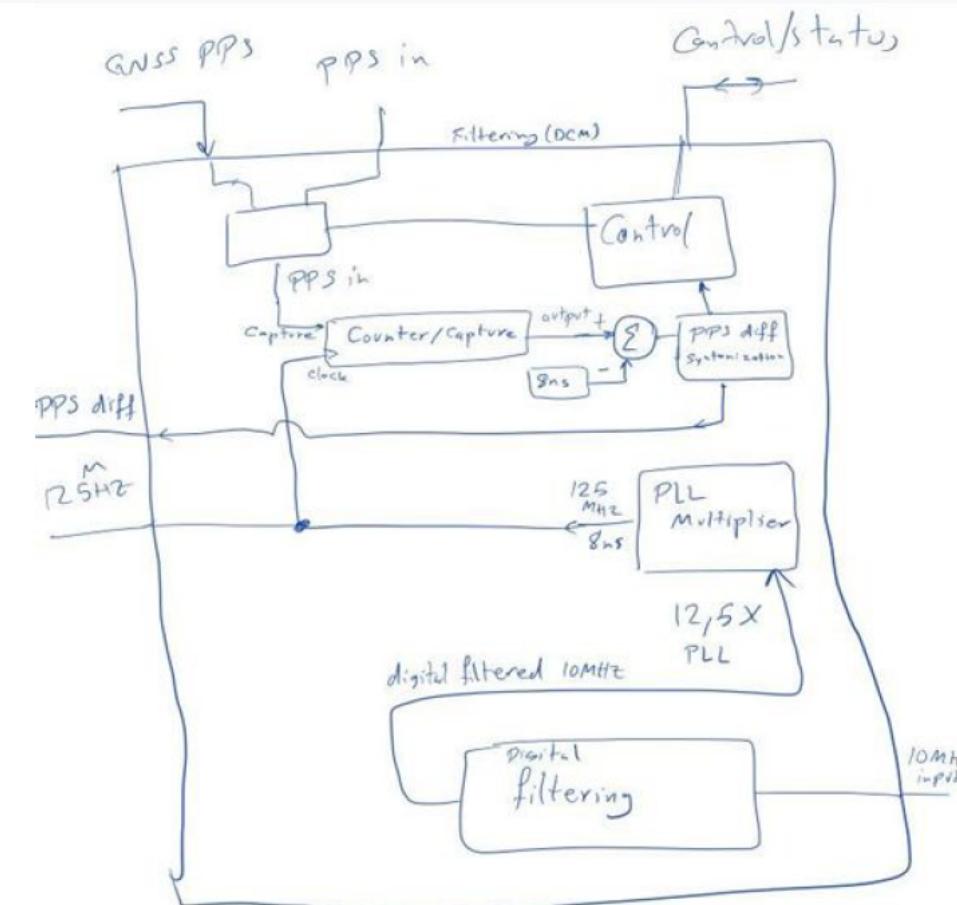
Here the **GM** outputs 10MHz directly from RB clock which is **freerun** if not controlled.
According your diagram - the output is raw local oscillator frequency (undisciplined).





So you cannot **output** 10MHz adjusted from external 10MHz in. You cannot monitor clock of PTP master in physical way to validate instant PTP clock (faster and more accurate than evaluate from PTP packets timestamps).





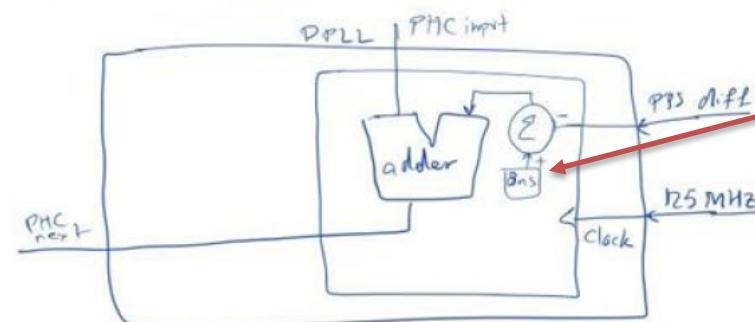
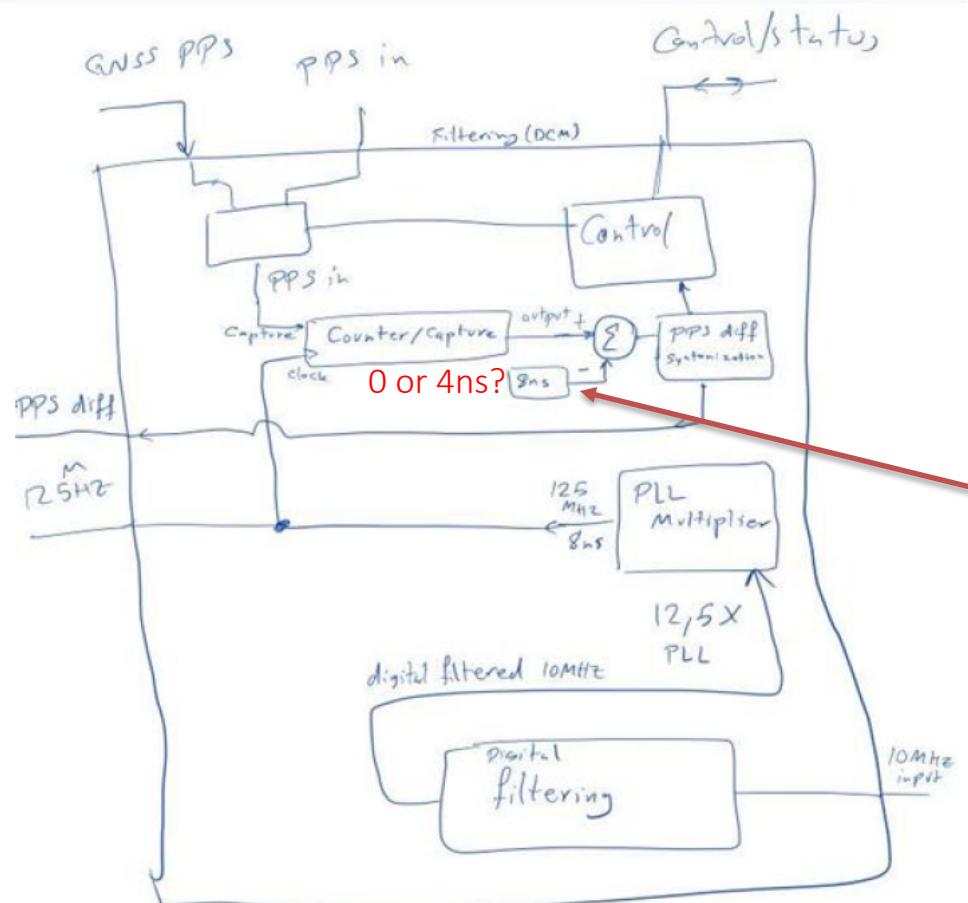
Here you do adjust freq. offset using **PHC counter**. Even with fine grade (fractional) **step** it is virtual scale only which you can output as 1PPS/timestamp with +/-8ns **granularity**. This can also effect in a **sawtooth** - so keep in mind. It depends on freq. control algorithm.

1PPS input counter capture pattern example-jitter [ns]:
0 8 8 0 0 0 0 -8 0 -8 0...

You can average but it depends on local oscillator.

1PPS output: **sawtooth**
examples – time error [ns]:
-8 0 8 0 -8 0...
0 8 16 0 8 16 0...

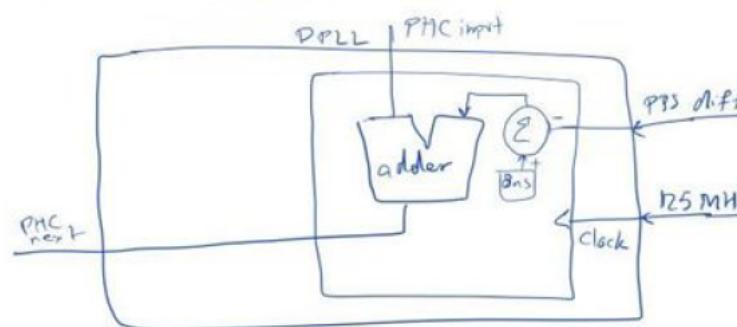
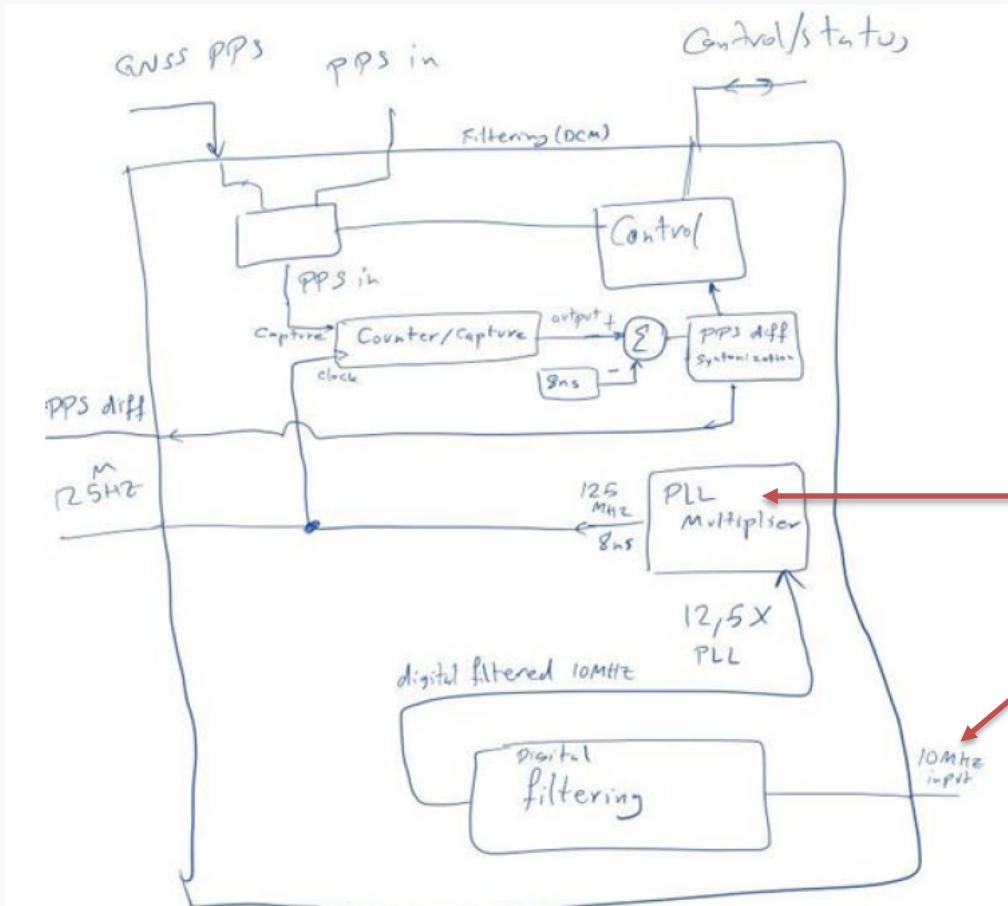
External device MUST average...



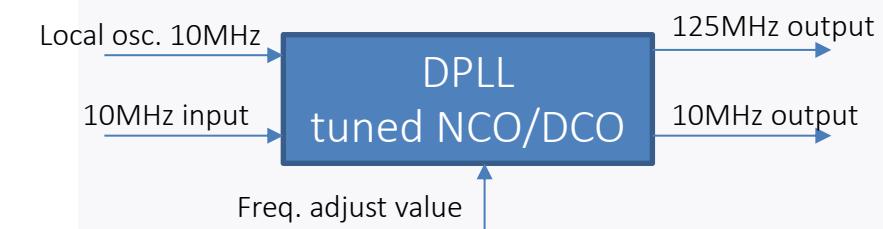
Here you do adjust freq. offset using **PHC counter**. Even with fine grade (fractional) **step** it is virtual scale only which you can output as 1PPS/trigger with +/-8ns **granularity**. This can also effect in a **sawtooth** - so keep in mind. It depends on freq. control algorithm.

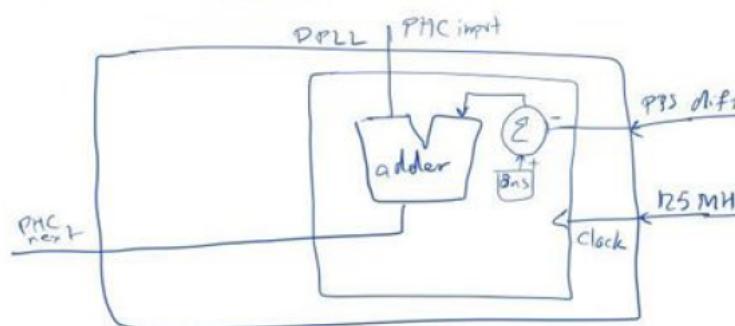
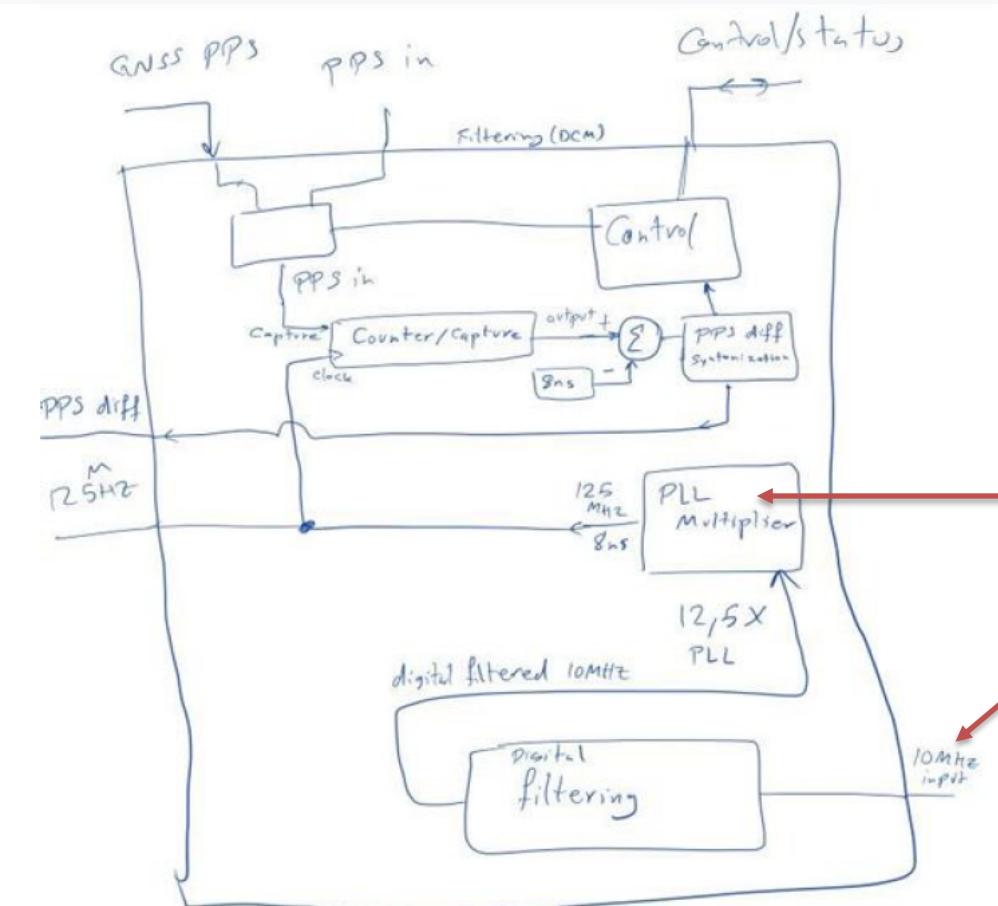
Freq. adjust (fractional) pattern examples [ns]:
8 8 8 7 8 8 8 7.... (to achieve -0.25ppb)
8 8 8 8 8 9 8 8 8 9.... (to achieve +0.2ppb)

On the picture there is no filter (Integral/Proportional: PI) to average phase error and control 2nd order PLL's frequency?

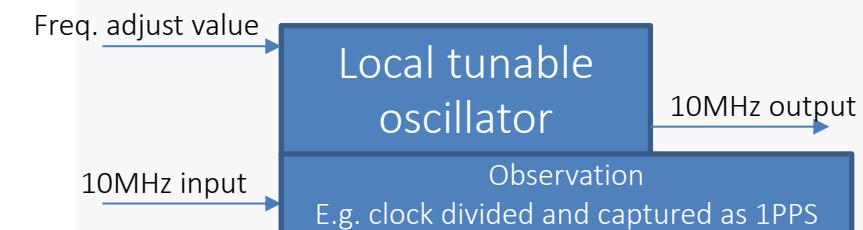


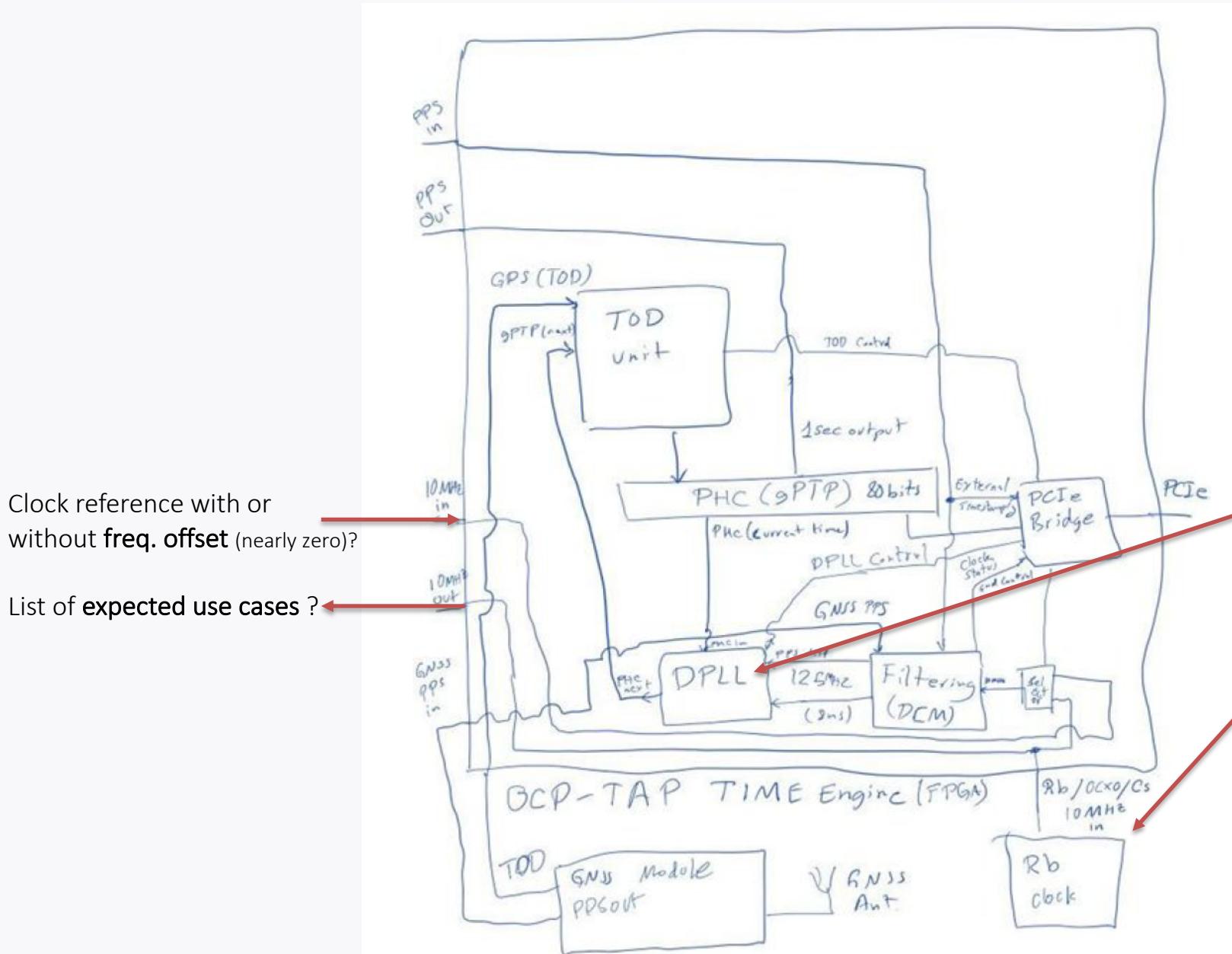
It would be better to tune frequency using DPLL (Numerically/Digitally Controlled Oscillator) or tune local oscillator (digitally or over DAC). This can reduce **1PPS jitter** and **sawtooth effect** thanks to much more fluently tuned frequency. Adv. easy **validation** of 10MHz output. Useful for **1PPS calibrations** too.





It would be better to **tune frequency** using **DPLL** (Numerically/Digitally Controlled Oscillator) or **tune local oscillator** (digitally or over DAC). This can reduce **1PPS jitter** and **sawtooth effect** thanks to much more fluently tuned frequency. Adv. easy **validation** of 10MHz output. Useful for **1PPS calibrations** too.





You need to **clarify goals** for 10MHz output. What is purpose of the output for you and the GM? If you want to tune and output the **adjusted** 10MHz and your input is external 10MHz - better to:

- use **DPLL** (e.g. Network Synchronizer chip dedicated for IEEE1588 with 1ppt or even better resolution)
- or **tune local oscillator** to the external frequency reference. It will also be useful for SyncE implementation in future.

About oscillator choice – read its **specification** carefully:

Rubidium, especially **miniature**, will vary with **temperature** noise (expected 5-10ns variation on rapid temperature change, e.g. 20ppt = 20ps/s error your phase detector will see at **8ns resolution** after 400s...).

Let's consider latest **super (D)OCXO**.

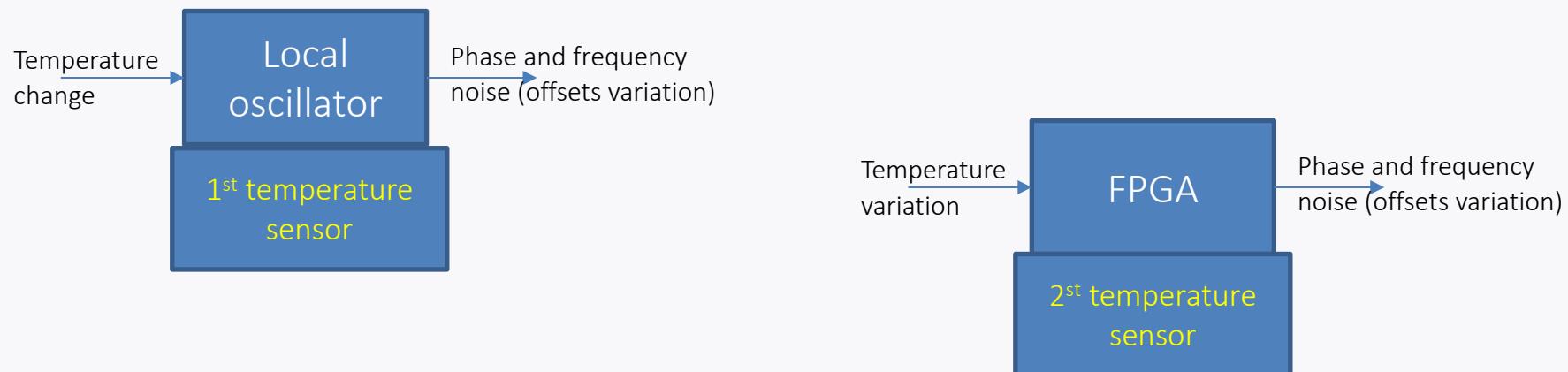
On the other hand **high quality Rubidium** oscillator in long-term give better holdover (smaller time error), especially in constant temperature. Unfortunately the oscillator is quite **big** and usually power hungry for the GM PCIe-form-factor card and you rather use external 10MHz input for it.



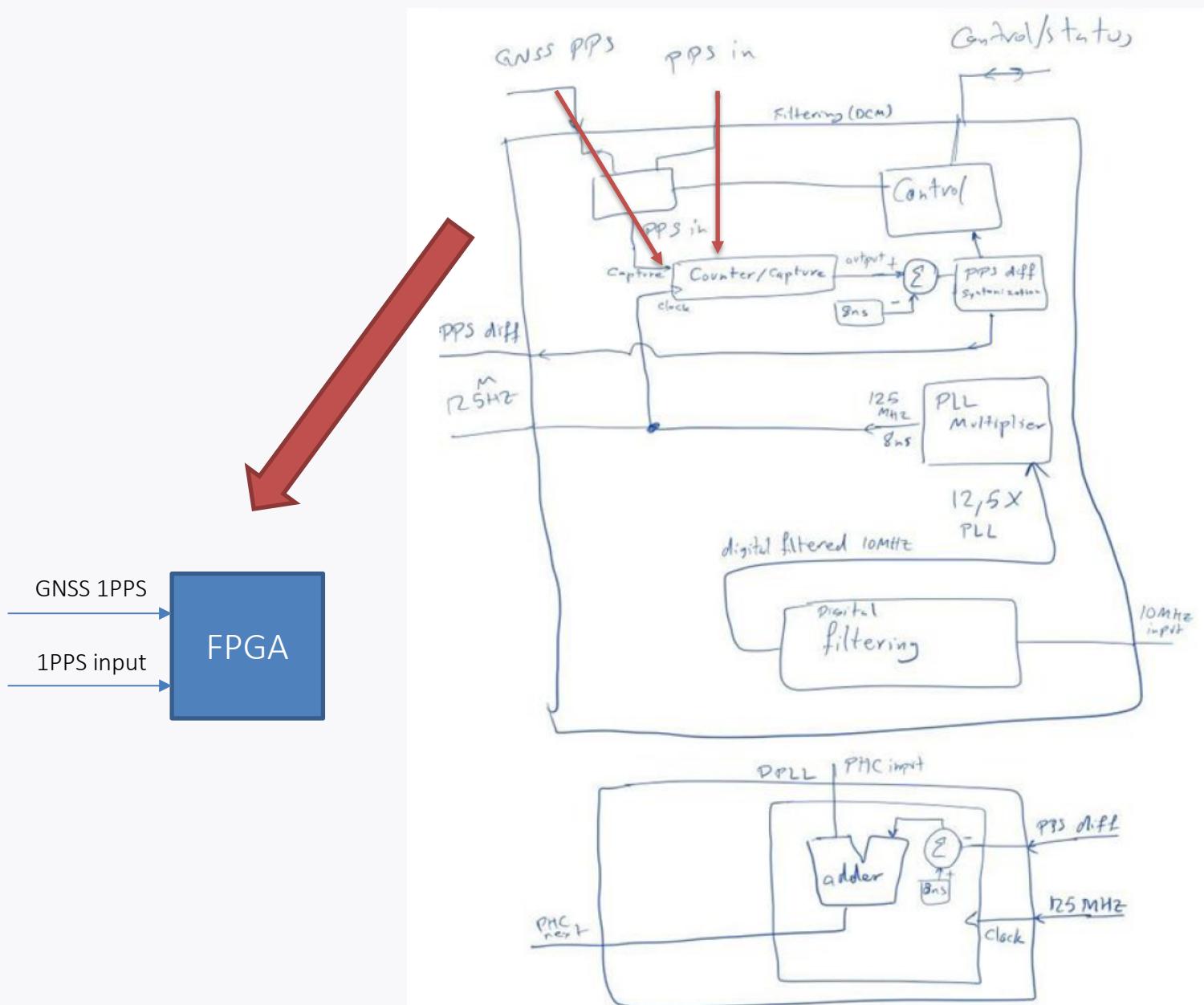
BTW. In reference to SiTime's presentation:
MEMS TCXO is right for PTP slave, not for GM – at least **not for good holdover**.

At least one **temperature sensor** located close to oscillator to monitor and **distinguish environmental temperature influence from other factors.**

A second sensor in/near FPGA if you would like to improve 8ns resolution to below 1ns accuracy.

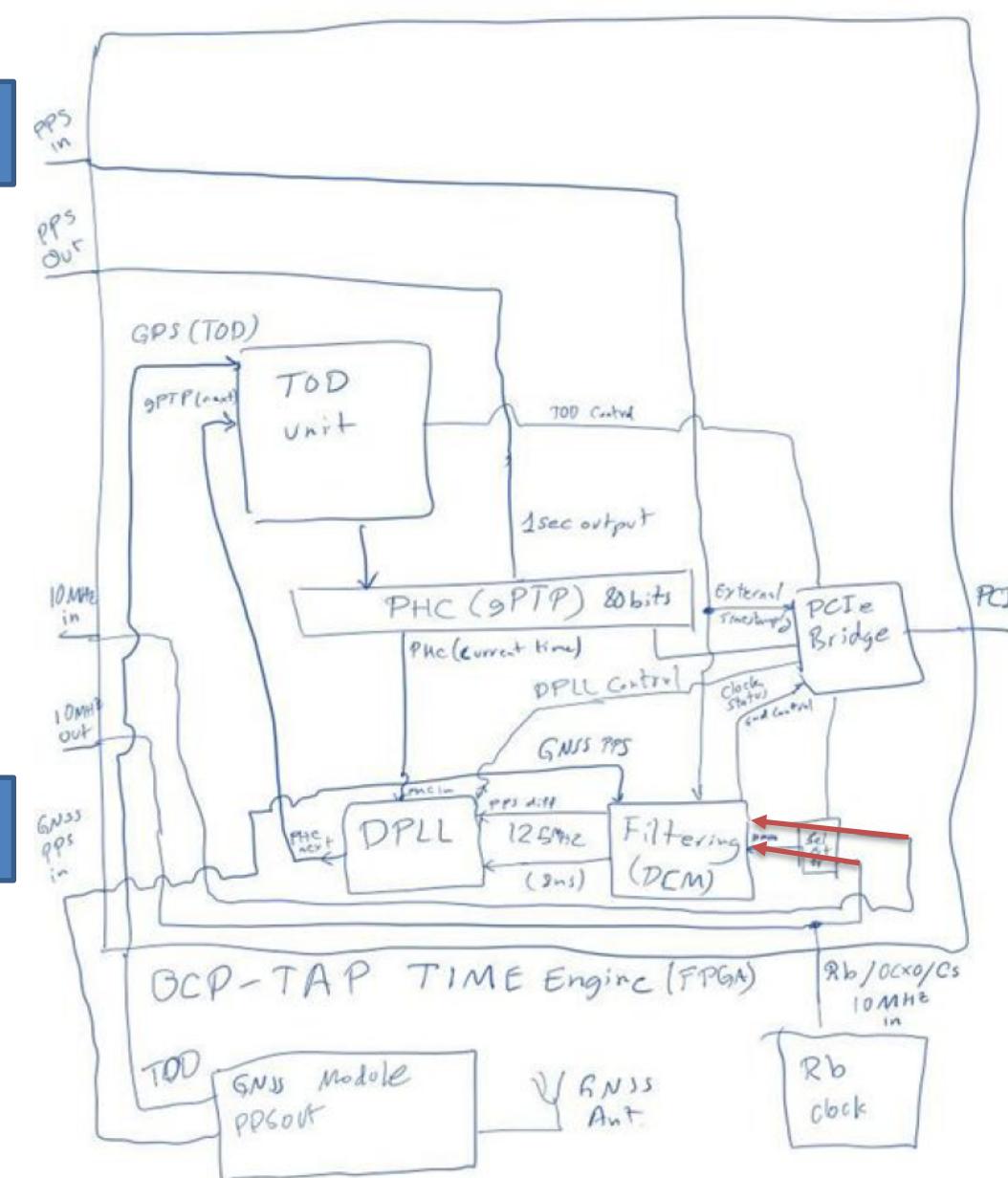
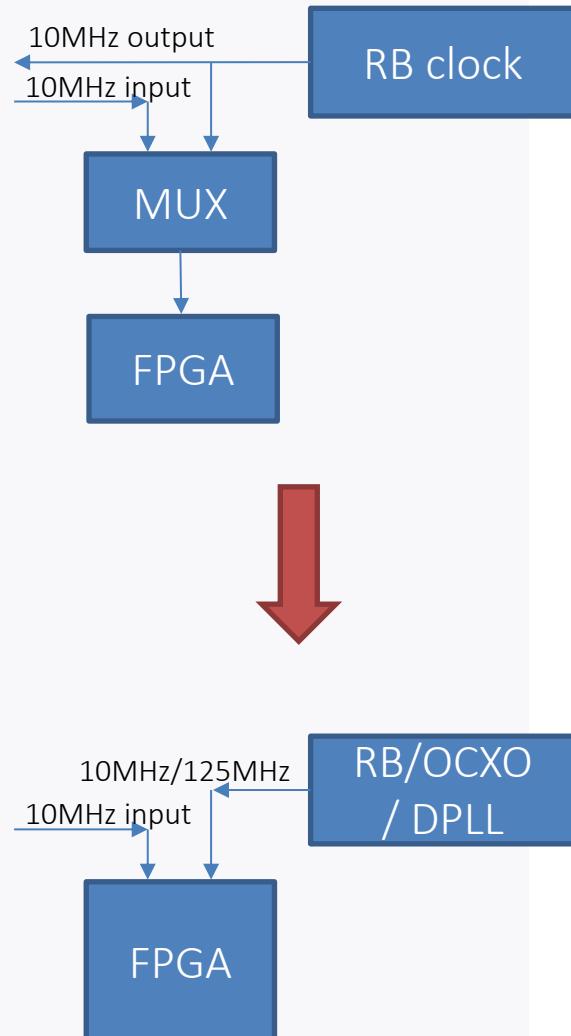


Your solution will cover current needs (+/-8ns resolution, <10us accuracy for PTP slave).
However let's keep door open for future higher accuracy.



To keep your architecture more flexible and open for the future, please consider not only multiplexed **GNSS 1PPS and external 1PPS input** connected to FPGA but to connect them both (separately) **to FPGA**. It will allow in future:

- Monitor (statistics) alternative time source before switching on.
- ePRTC implementation: 1PPS from Cesium clock



Similarly current diagram multiplexes **local oscillator** and **10MHz input**. Our advice is to connect the external 10MHz also to **FPGA directly** plus the current multiplexer.

This will occupy little more FPGA resources (e.g. two Time-Interval-Counters/Time-Stamp-Counters/Time-To-Digital-Converters for the 1PPSes).

It will allow to connect 10MHz clock from Cesium clock to build ePRTC functionality in future.

In reference to Facebook's presentation at **ITSF** on 23/09/2020 and Q&A:
2 sites, when one or both GNSS failed – local holdover only.

SyncE advantage: lower phase difference (noise) than PTP only, more robust

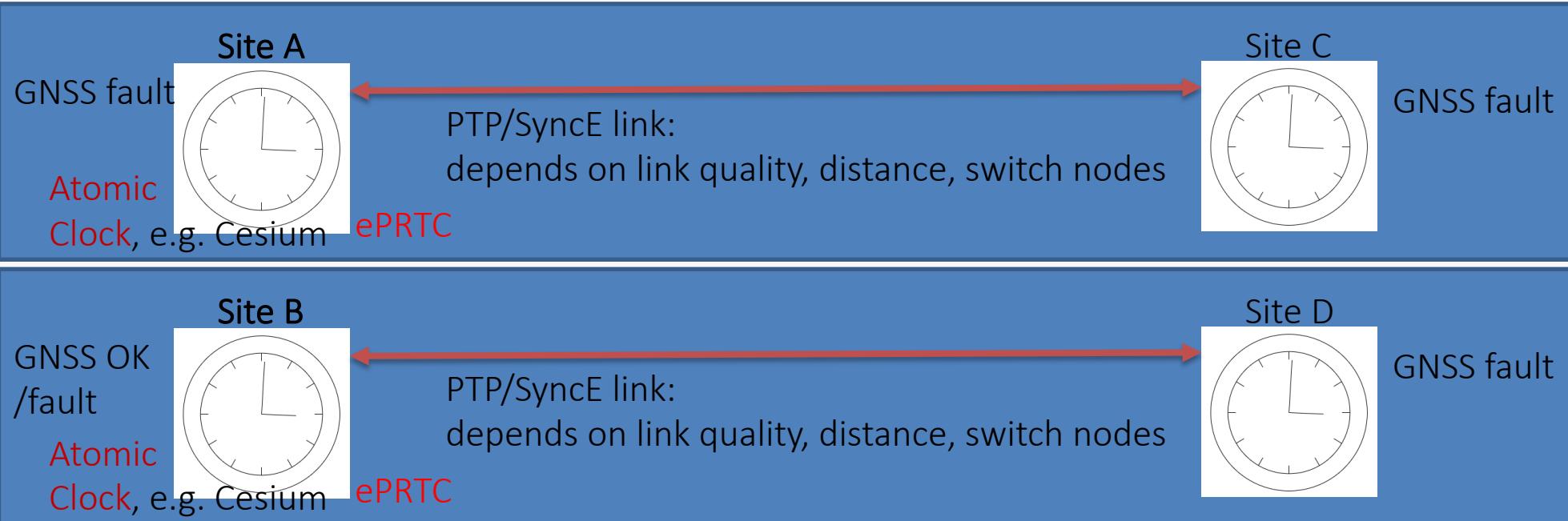


2-4 sites, when one or all GNSS failed (e.g. due to high area jamming/spoofing):

SyncE: to transfer high quality in long-term remote clock (from site A/B to C/D)

ePRTC holdover: lower **absolute** (to UTC) phase difference, more robust

Very useful when 2 sites (A-B) are “time islands”, i.e. no PTP/SyncE link/fibre between them.



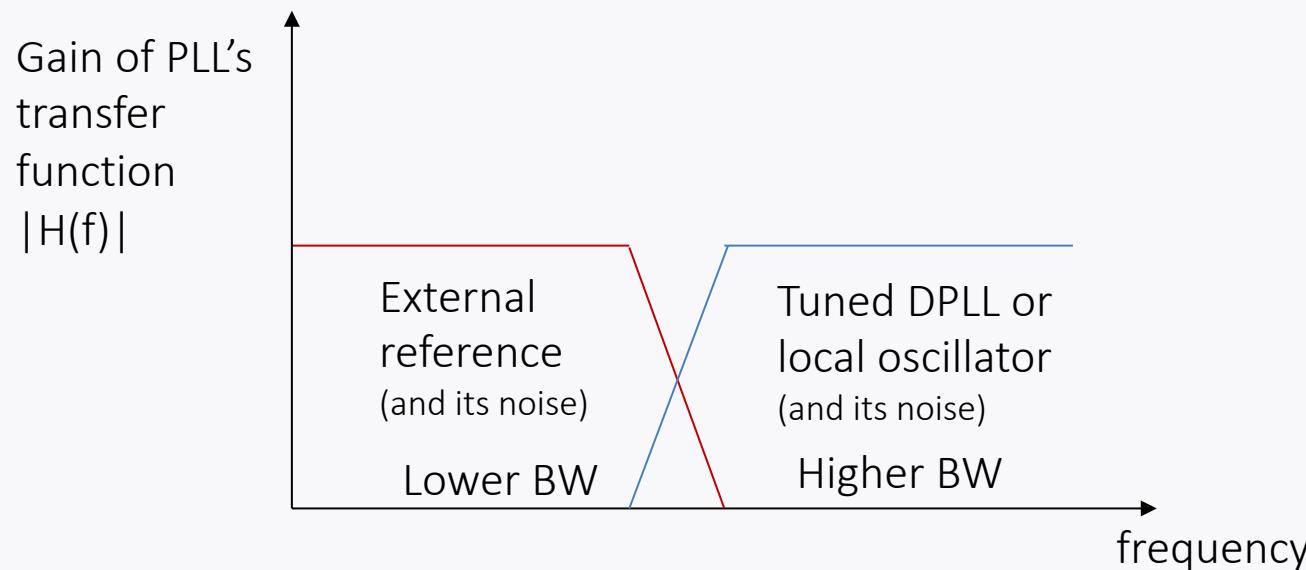
Thank you.

Q & A

Backup slides

DPLLs usually have higher **short-term noise**

(depends on PLL's bandwidth, *BTW. it affects ADEV below related integration time*)
in comparison to tuned local HQ oscillators (Double/Super OCXO, RB).



DPLLs usually have higher **short-term noise**

(depends on PLL's bandwidth, BTW. it affects ADEV below related integration time)
in comparison to tuned local HQ oscillators (Double/Super OCXO, RB).

But DPLLs do not require **calibration** (have known VCO gain / freq. step for the PLL),
i.e. you can set frequency offset without tuning range calibration (for voltage adjusted oscillators).
The DPLL's advantage is not valid when you use **digitally controlled** e.g. Rubidium/atomic oscillator.

