WR at MLE - Updates on Dormouse FMC Card, Light Rabbit and Further Contributions

Ulrich Langenbach, Oskar Szakinnis

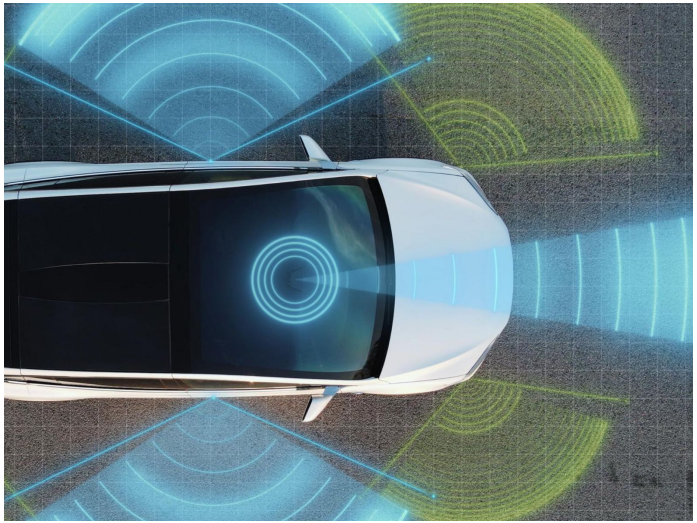
Updates on White Rabbit at MLE

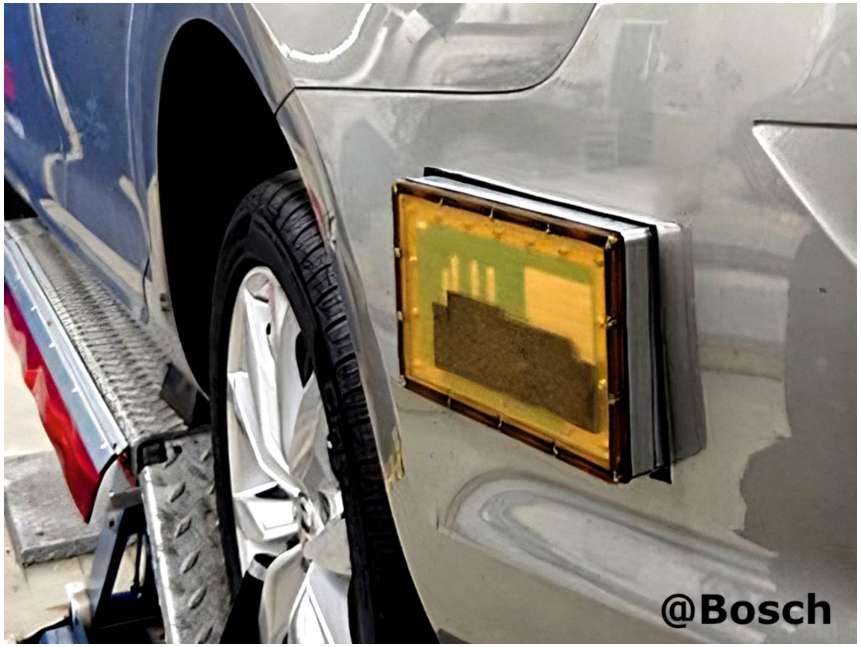
- Dormouse FMC Card
- GNSS Grand Master Enablement
- Light Rabbit
- White Rabbit Switch v4 on ZCU102
- Continuous Integration and Testing

MLE aims at supporting industry to adopt the White Rabbit technology into embedded systems in automotive, sensing, industrial and more markets.

VERANO: Dormouse Motivation/Scenario

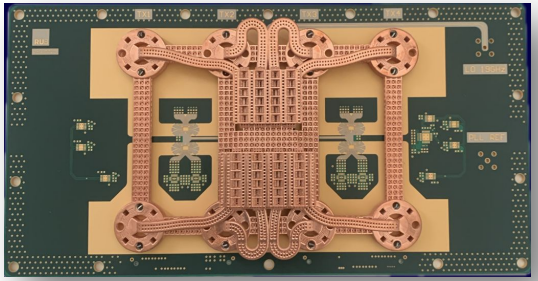
- Publicly funded project (BMFTR 16ME0785K) for automotive radar networks
- Bi- or Multi-static Radar Systems
- 6G Massive MIMO Systems

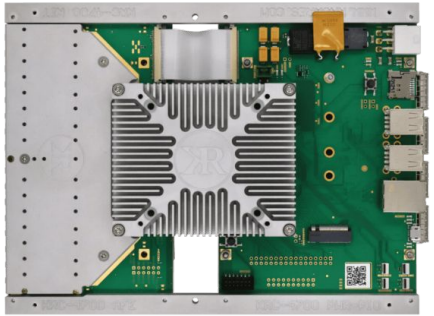




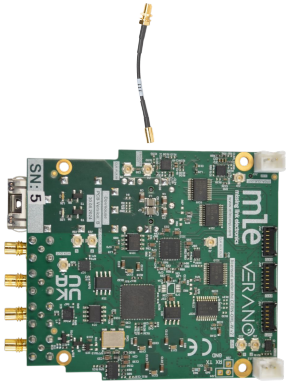
VERANO: Dormouse Motivation/Scenario

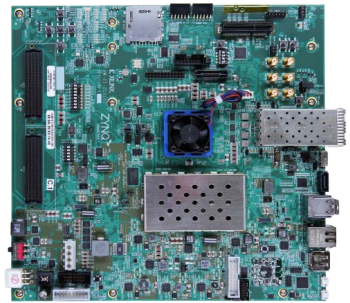
Status: Demonstrators for multi-static RADAR Systems





100 GbE



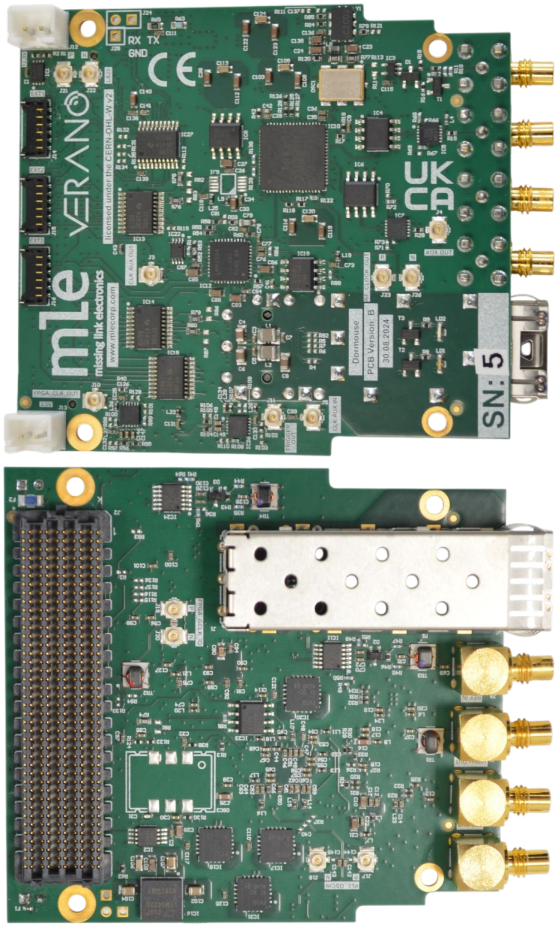


1 GbE

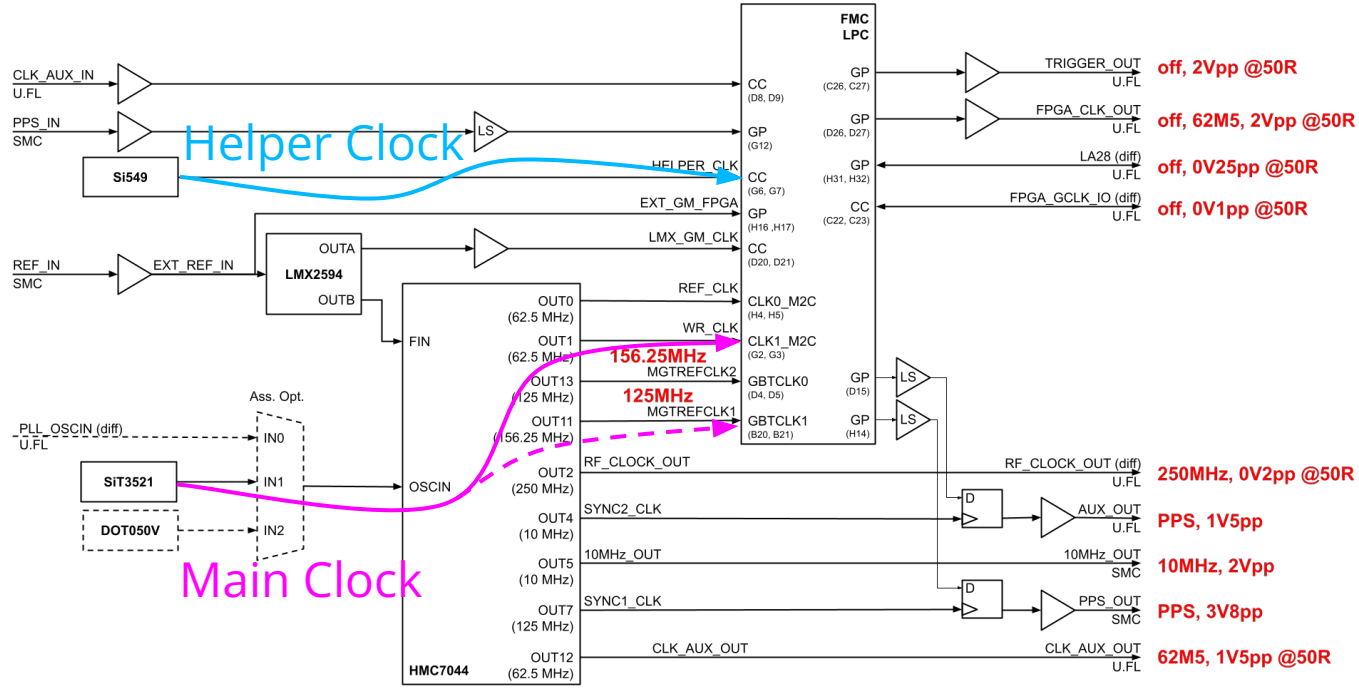
WR

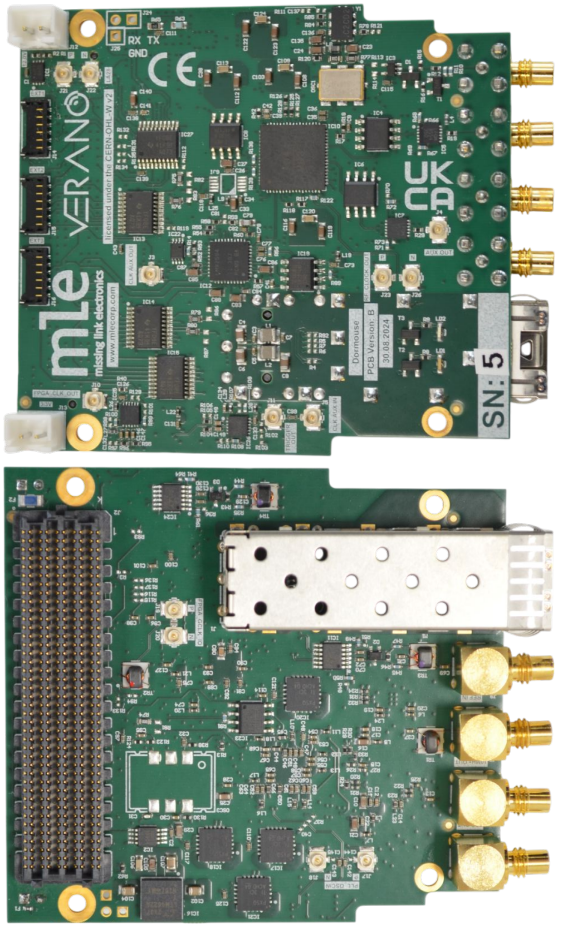
Dormouse FMC Card

- Based on FCWR
- Swapped one Si549 to SiT3521 DPLLs
- Individual contributions to wr-cores/wrpc-sw required for dormouse operation
  - SiT3521 operation
  - HMC7044 driver extension
  - HMC7044 CLI interface for individual output channel control
- Commercially available

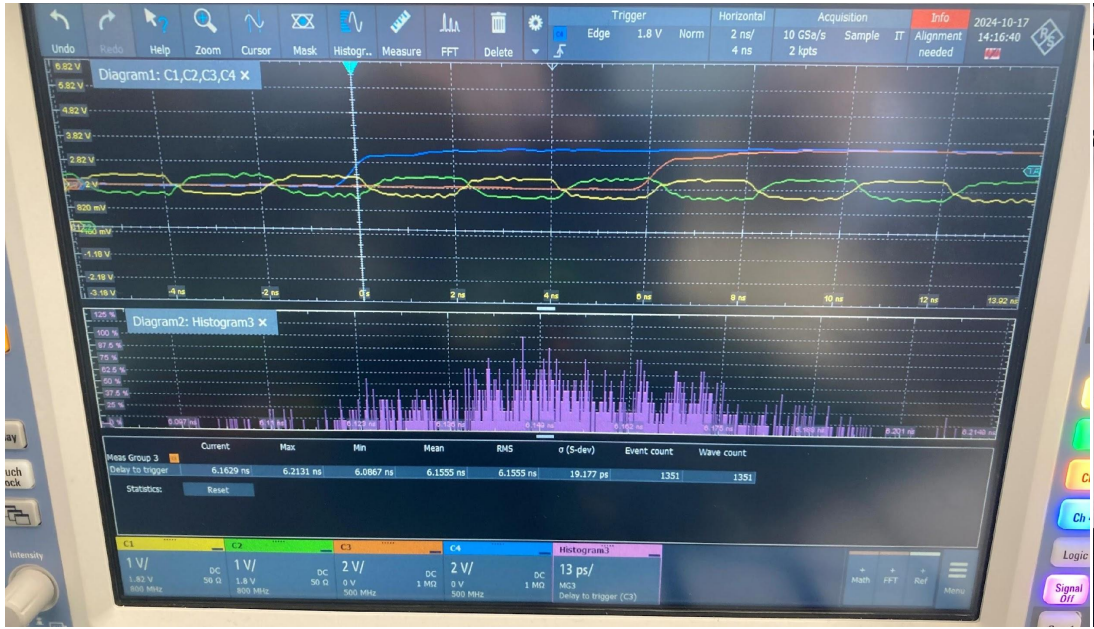


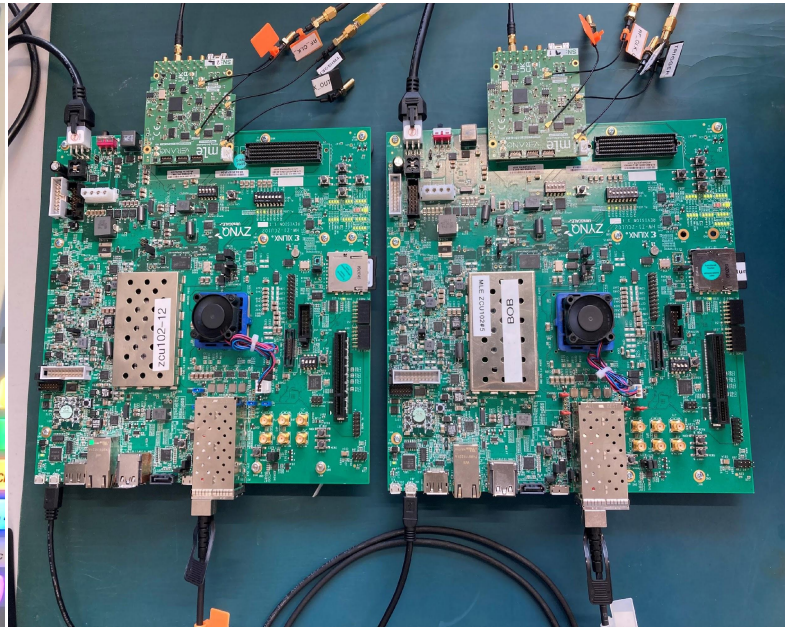
Dormouse FMC Card





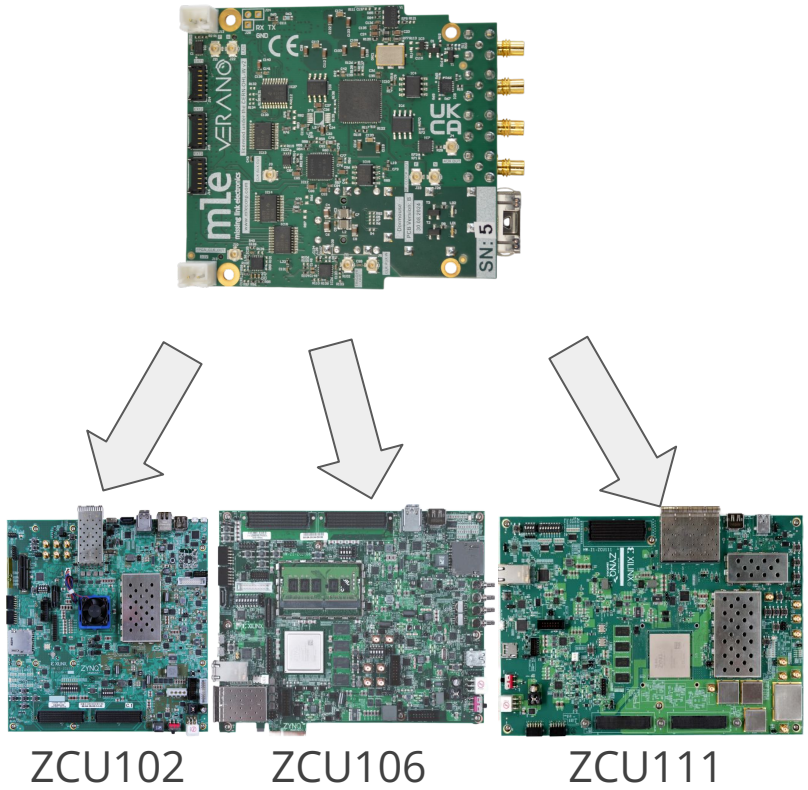
Dormouse FMC Card





	Current	Max	Min	Mean	RMS	σ (S-dev)	Event count	Wave count
Meas Group 3								
Delay to trigger	6.1629 ns	6.2131 ns	6.0867 ns	6.1555 ns	6.1555 ns	19.177 ps	1351	1351

White Rabbit Dormouse Support



WR EP GNSS Grandmaster Enablement

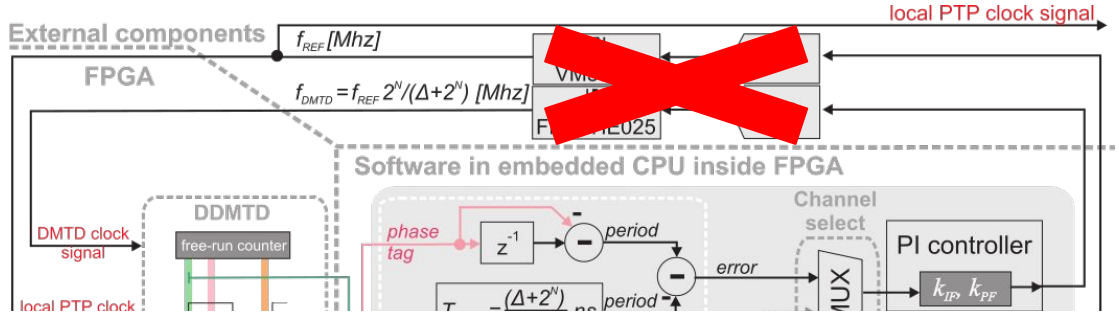
WR EP		
10MHz IN	PLL Control	
UART	Risc-V NMEA parsing for ToD	
PPS IN		

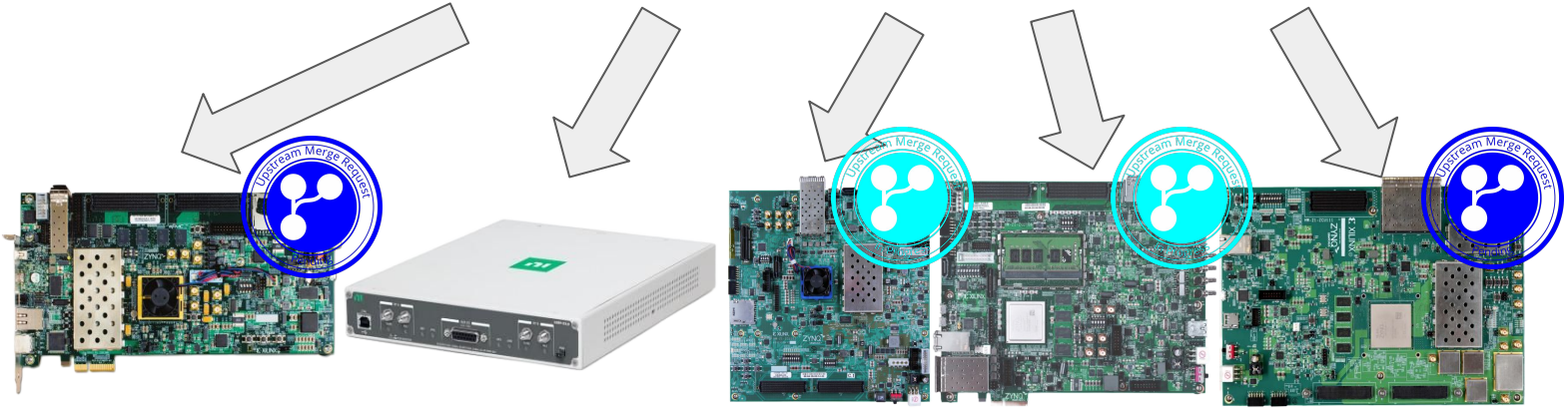
ZC706 X310 ZCU102

WRPC as GNSS-enabled Grandmaster

- Motivation: WRPC as master, used in a distributed sensing scenario
- Use GNSS as a fallback if the main synchronisation mechanism fails
- wr-cores:
  - Second UART for NMEA input (ZCU102/ZCU106)
  - MMCM with appropriate configuration for 10MHz reference input
  - https://gitlab.com/ohwr/project/wr-cores/-/merge_requests/27
- wrpc-sw:
  - NMEA sentence parsing
  - wrpc shell commands for synchronization to GNSS time
  - https://gitlab.com/ohwr/project/wrpc-sw/-/merge_requests/25

Light Rabbit



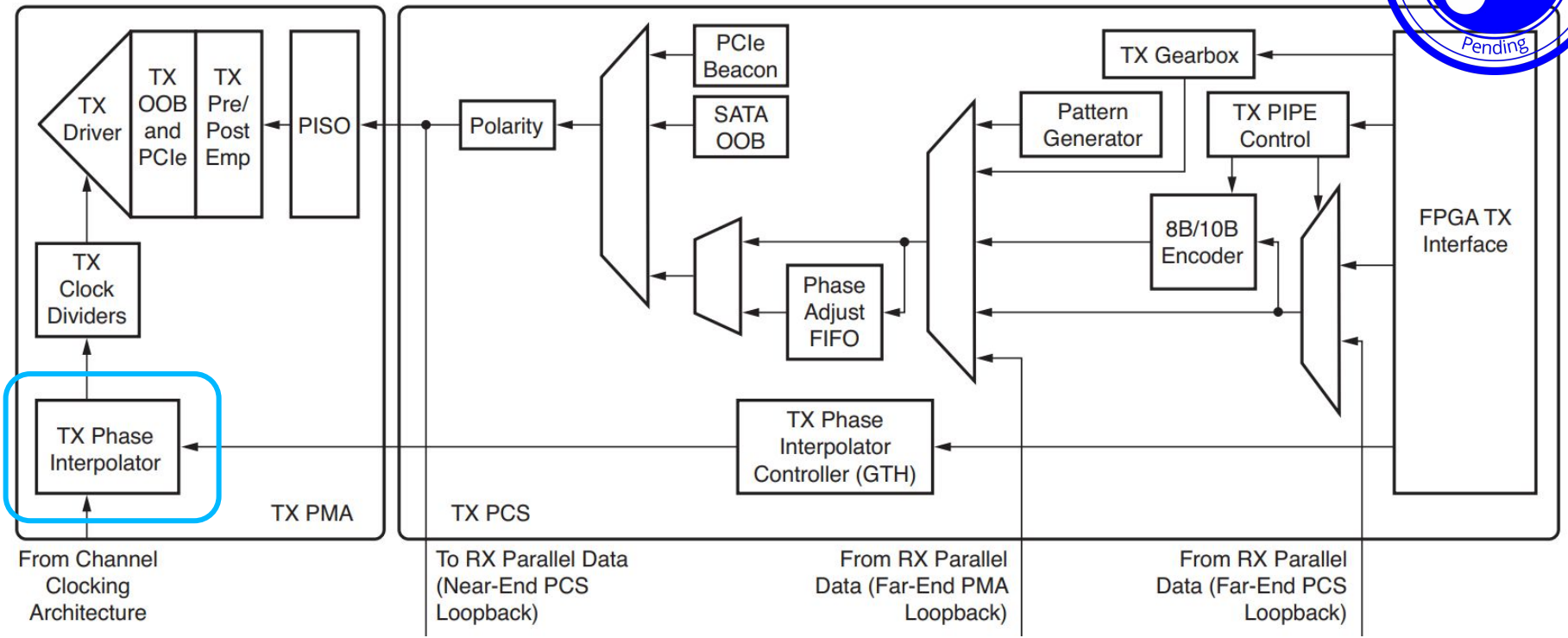


ZC706 X310 ZCU102 ZCU106 ZCU111

Light Rabbit revisited

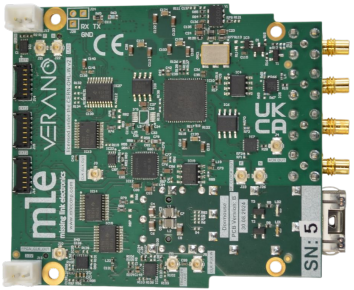
- Idea: Implementing White Rabbit on COTS AMD development boards without suitable VCXOs
- Merge Requests available
- QPLL-based on AMD MPSoC series, GTH/ GTY Transceivers:
  - On ZCU102/ZCU106
  - wr-cores MR: https://gitlab.com/ohwr/project/wr-cores/-/merge_requests/18
- MMCM-based on AMD 7 series, GTH Transceivers
  - On ZC706
  - wr-cores MR (1/2): https://gitlab.com/ohwr/project/wr-cores/-/merge_requests/28
  - wr-cores MR (2/2) : https://gitlab.com/ohwr/project/wr-cores/-/merge_requests/29
- Related wrpc-sw (already merged): https://gitlab.com/ohwr/project/wrpc-sw/-/merge_requests/24

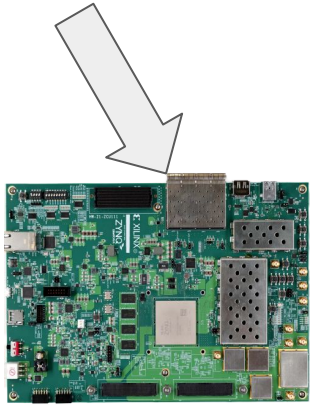
Light Rabbit: PICXO - using TX PI



https://docs.amd.com/v/u/en-US/ug476_7Series_Transceivers

White Rabbit Dormouse GTY support





White Rabbit Dormouse GTY support

- Quick adaption of the GTH transceiver configuration to GTY
- Implemented on ZCU111
  - Light Rabbit
  - White Rabbit (with Dormouse support)
- Experimental LDPC implementation, not yet stable / final

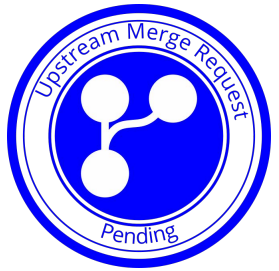
Continuous Integration

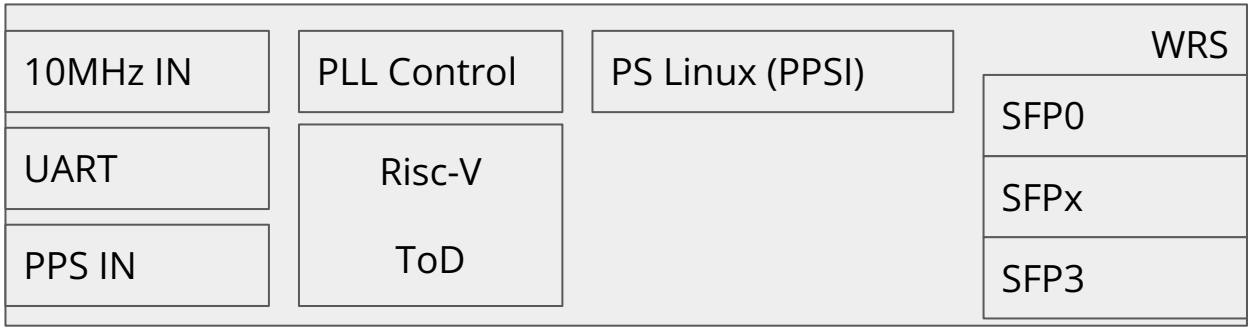
- Run the build flows (MLE supported platforms)
  - Gateware
  - Firmware
- Artifacts
  - Logs
  - Bitstreams
  - SD-Card images
- Archive Results

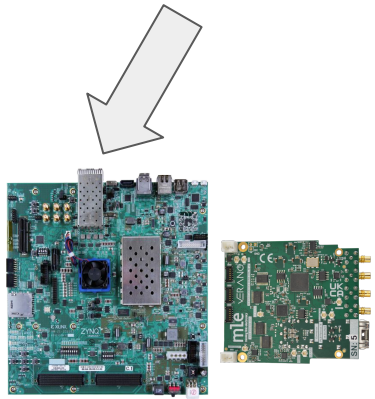
Continuous Testing

- Full remote hardware access
- Orchestrate hardware (boards) via labgrid “places”
- Operate hardware via labgrid “strategies”
  - Stateful interaction with the boards
  - Allows for automated and manual interaction
- Pytest based tests
  - Check connectivity with known good bitstreams on a node
    - Boot
    - Gain WR lock
  - Reboot the node with the Release under test
  - Gain WR lock
  - Optional: GNSS lock, loss of lock, GNSS re-lock (power cycling the GNSS module)
  - Dormouse based tests
  - Various long and short term automated PPS measurements

WIP: (Small) White Rabbit Switch







ZCU102

WIP: (Small) White Rabbit Switch

- Retargets the WRSv4 implementation to ZCU102 + Dormouse to keep it as similar as possible
- Re-uses developments for the Dormouse Endpoint Implementation on ZCU102
- Optional integration with Light Rabbit
- Current state
  - Gateware available
  - Firmware available
  - Software available
  - System boots, but no functional WRS yet => debugging
- Current challenges
  - Software tool chain and build process

Conclusion

- On-going support and enhancement of MLE enabled platforms
- Providing MRs to make the developments available to all of us

⇒ What's next? Work-in-progress…

- Absolute Calibration with the special SFP Loopback Module
- Further look into AUX clock use cases combined with Light Rabbit
- Investigate reset and power-on randomness sources

Help others to understand and integrate this great technology…

Thank You!

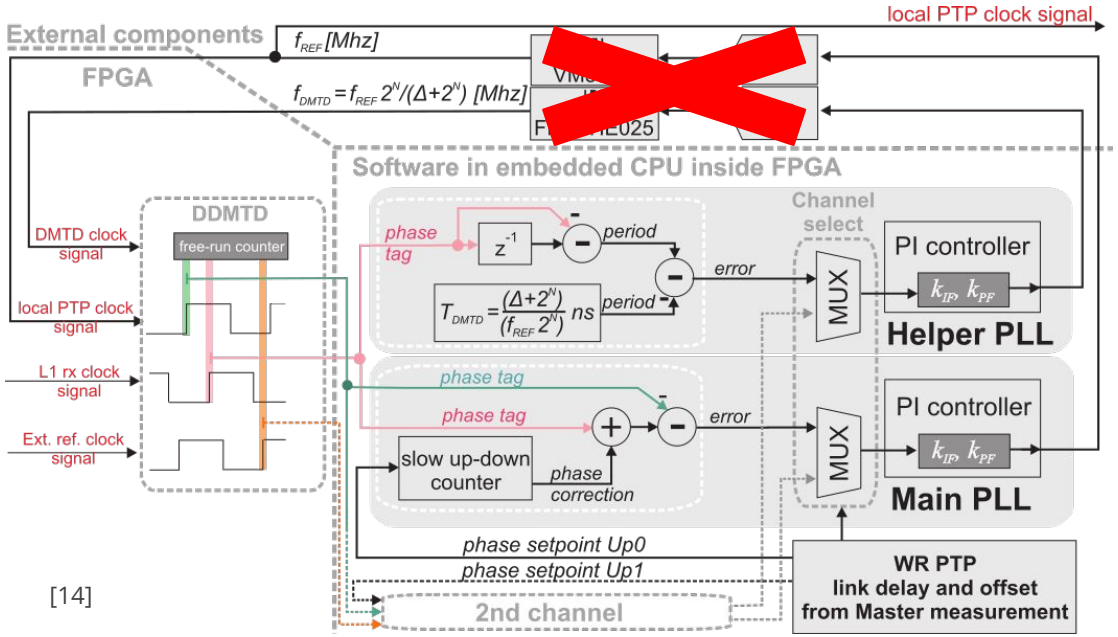
- DESY for providing loaner of a White Rabbit Switch
- U Ulm for supporting the measurements, e.g. with their R&S RTO, etc.
- BMFTR for funding the 6G-ICAS4Mobility and VERANO projects

Let’s continue to jointly work on a wide adoption of the WR Technology!

Backup

Light Rabbit

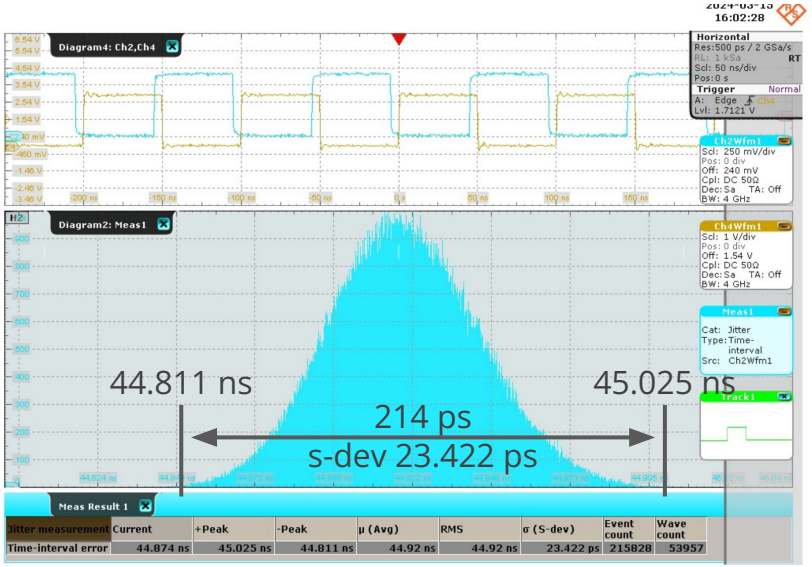
- Lower barrier to entry
- Reduces board complexity
- Enables ‘legacy’ HW w/o VCXOs



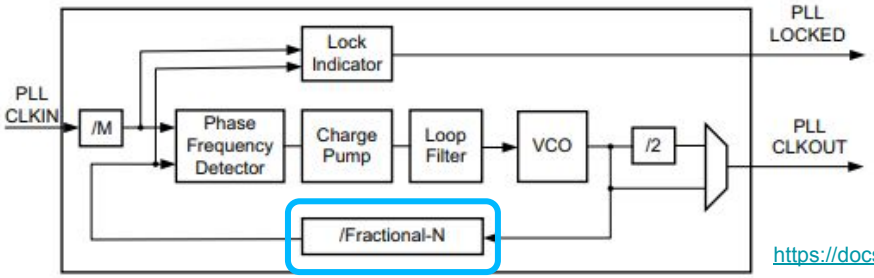
[14]

Refresher: QPLL-based Implementation

- VCXOs replaced by QLLs of UltraScale/UltraScale+ GT
- Frequency control by SDM interface of QLLs fractional-N feature
- Internally the QPLL performs a sigma-delta toggle between N and N+1 based on user provided fraction



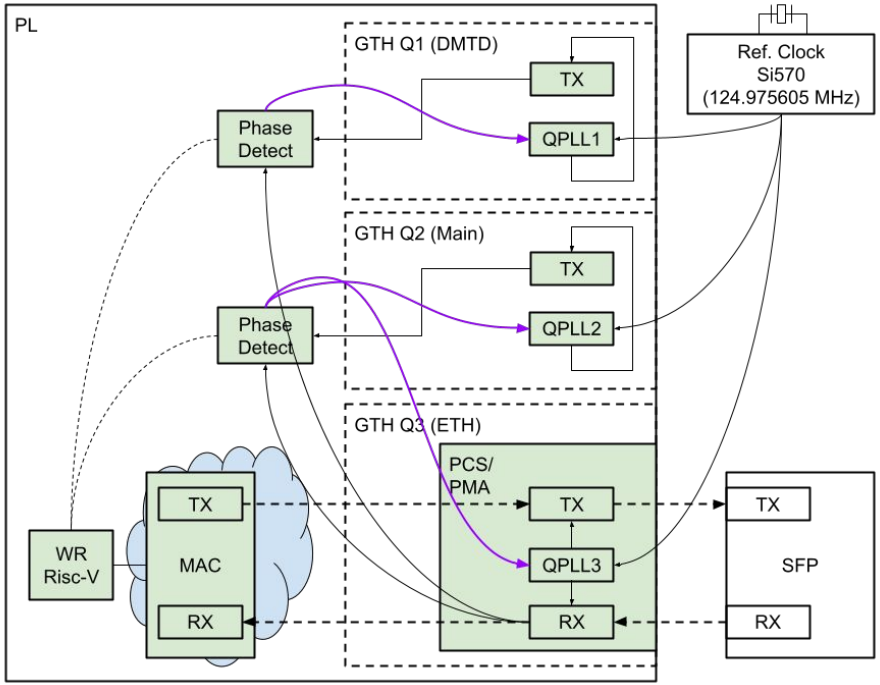
R&S Jitter Wizard
R&S RTO1044



https://docs.amd.com/v/u/en-US/ug578-ultrascale-gty-transceivers

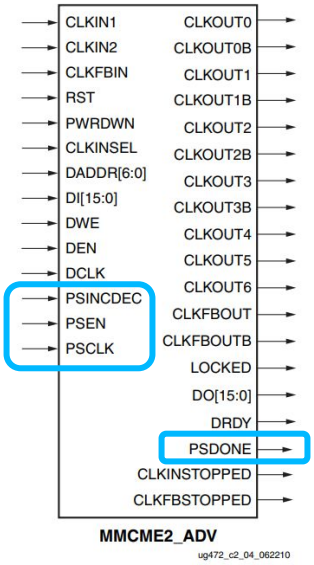
QPLL-based Implementation

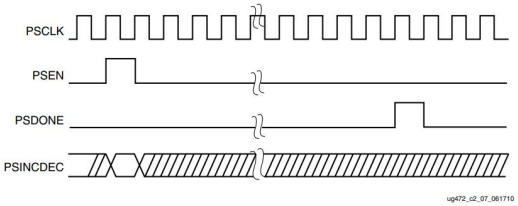
- ZCU102 using GTH QPLLs
- Requires multiple GTH Quads
- Requires external fixed GTH reference clock slightly below (or above) 125 MHz
- Main and ETH QPLL are tuned equally and simultaneously
- QPLL output clock needs to pass through a channel TX to be available in fabric
- Frequency is adjusted using the QPLL “SDM” feature



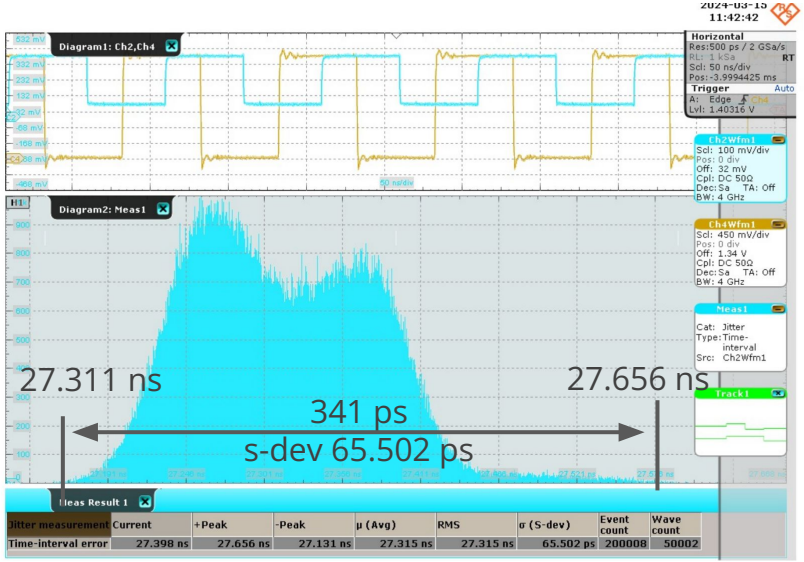
Refresher: MMCM-based Implementation

- VCXOs replaced by MMCMs
- Frequency control using dynamic phase shift interface of MMCM





- Add up 16 Bit DAC value every 12 cycles
- Shift by 1/56 th of a VCO period

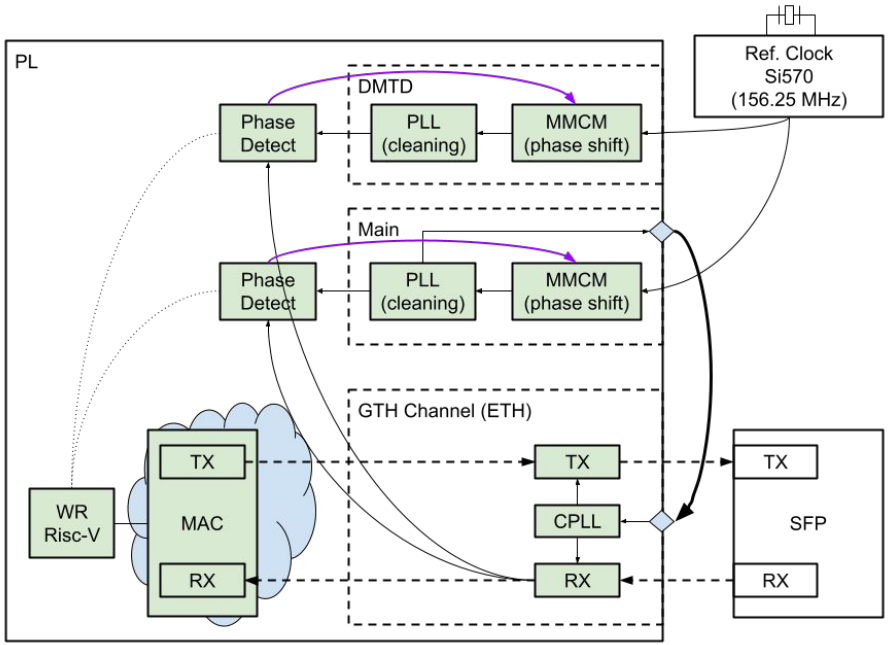


R&S Jitter Wizard
R&S RTO1044

https://docs.amd.com/v/u/en-US/ug472_7Series_Clocking

MMCM-based Implementation

- ZC706 using fabric MMCMs
- Clock oscillator can have any reasonable frequency
- Generated reference clock needs FPGA-external path to GTH reference clock pin (directly or through external PLL pass-through)
- Frequency is adjusted by repeated phase shifts



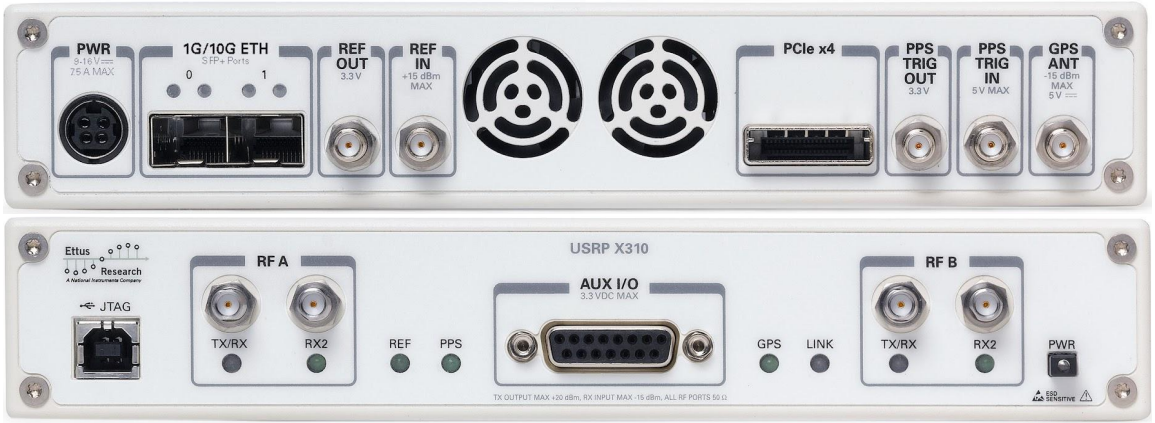
White Rabbit … on an Ettus X310 USRP

- AMD Kintex-7 XC7K410T FPGA
- PCIe
- 2x SFP+ (10G or 1G)
- Optional GPSDO
- DA-15 GPIO port
- 2x RF Front-End Daughter Board slots

⇒ **No VCXOs!**

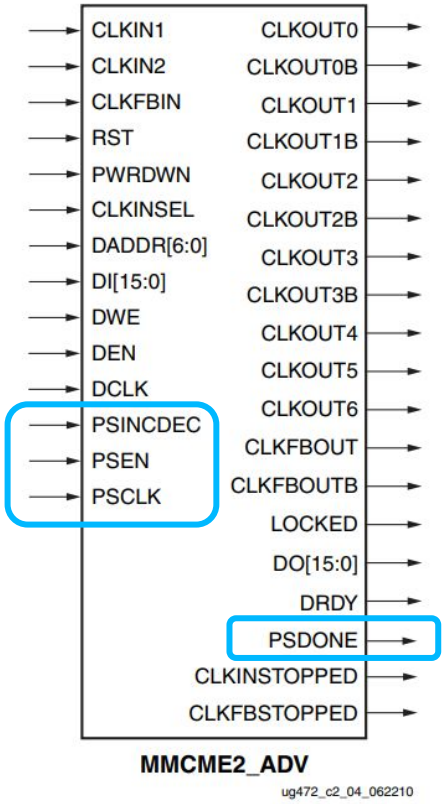
⇒ **No QPLLs with FRAC-N support!**

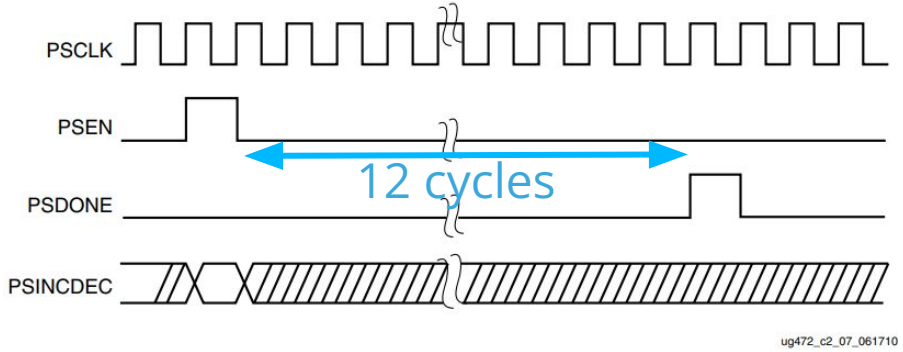
⇒ **7Series fabric ⇒ use MMCMs!**



[25]

MMCM: Dynamic Phase Shift Interface





- Add up (unsigned part of) 16 Bit DAC value every 12 cycles
- Sign bit → PSINCDEC
- On wraparound → PSEN = ‘1’
- Shift by 1/56 th of a VCO period

https://docs.amd.com/v/u/en-US/ug472_7Series_Clocking

Light Rabbit III: PICXO-based Implementation

- clock is tuned using GTX/GTH TX phase interpolator (PI) circuit
- Based on AMD XAPP589 (PICXO) to control ppm offset
- Reduced internal priority, so no MR planned for now