

RF Distribution over White Rabbit

John Gill

Special thanks to Tomasz Włostowski, Eva Calvo Giraldo, Mattia Rizzi, Miłosz Malczak and ESRF.



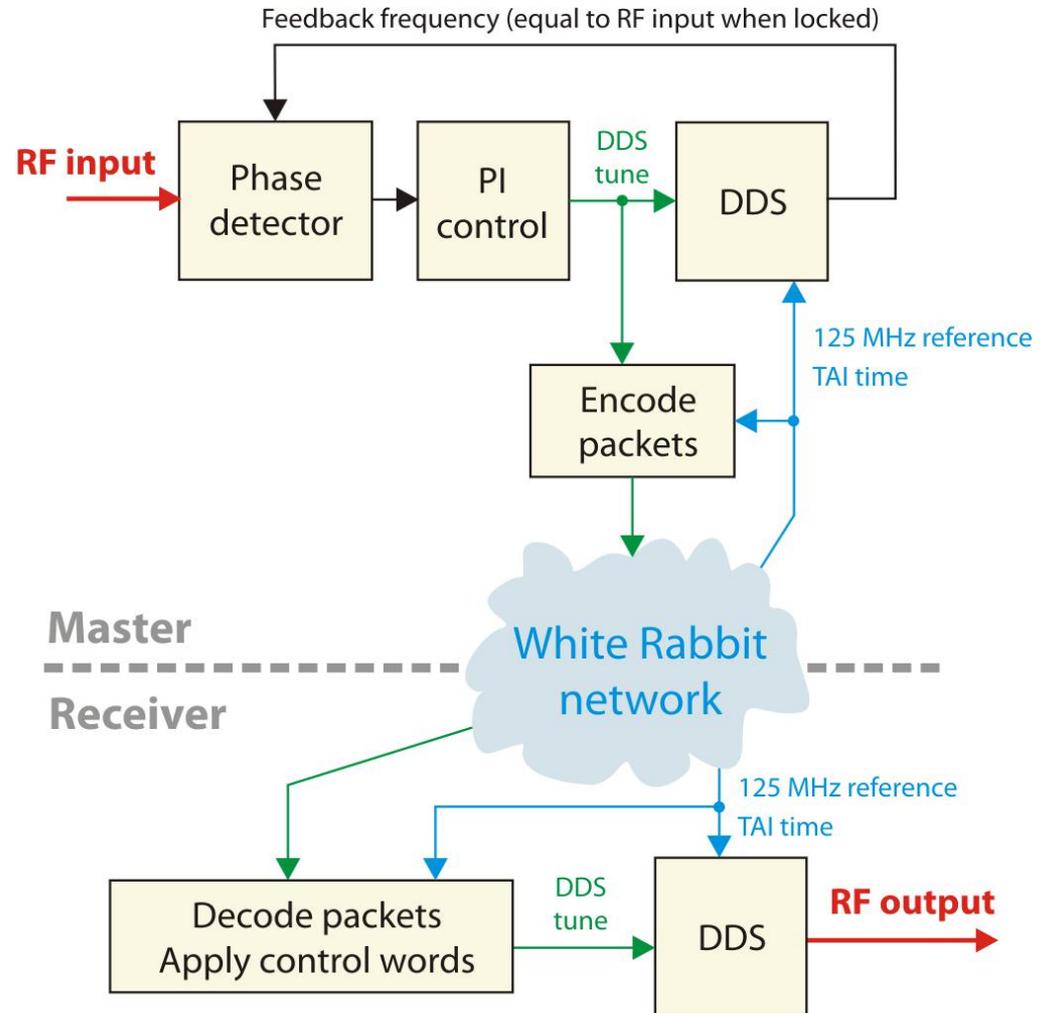
- **Introduction**
- **Status**
- **RFtime**
- **Next Steps**

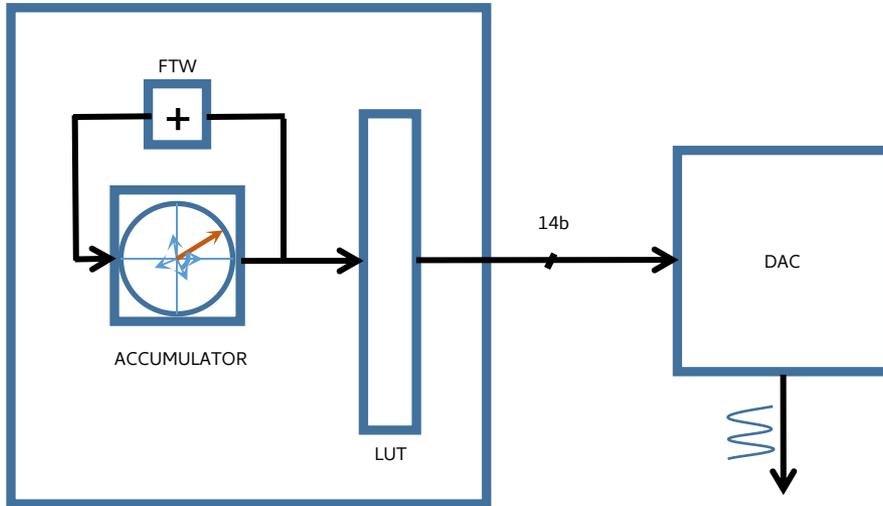
RF Distribution

General idea is to take an input RF signal and regenerate an in-phase copy of that signal anywhere in the White Rabbit network.

Packets containing Frequency Tuning Word (FTW) updates and snapshots of the accumulator values are regularly transmitted from the master to the receivers via the White Rabbit network.

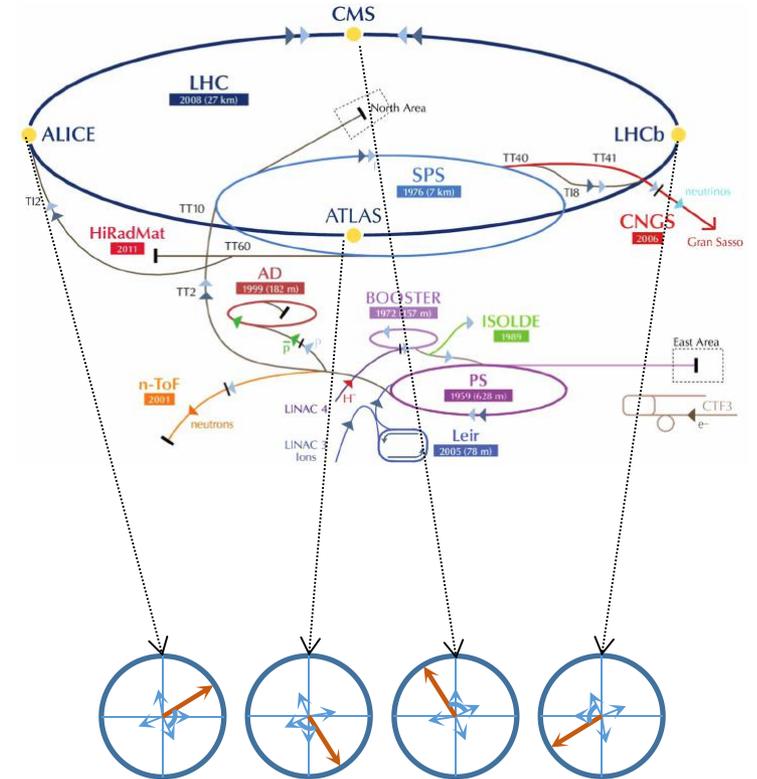
These packets facilitate the control of remote Numerically Controlled Oscillators (NCO) in the the White Rabbit network.



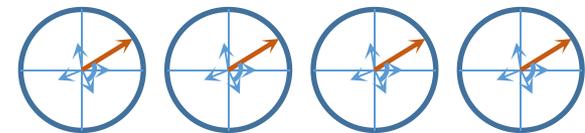


FPGA

WR clock cycle	Accumulator <small>FTW = 3, AccumulatorSize=8</small>
0	0
1	3
2	6
3	1
4	4
5	7
6	2
7	5



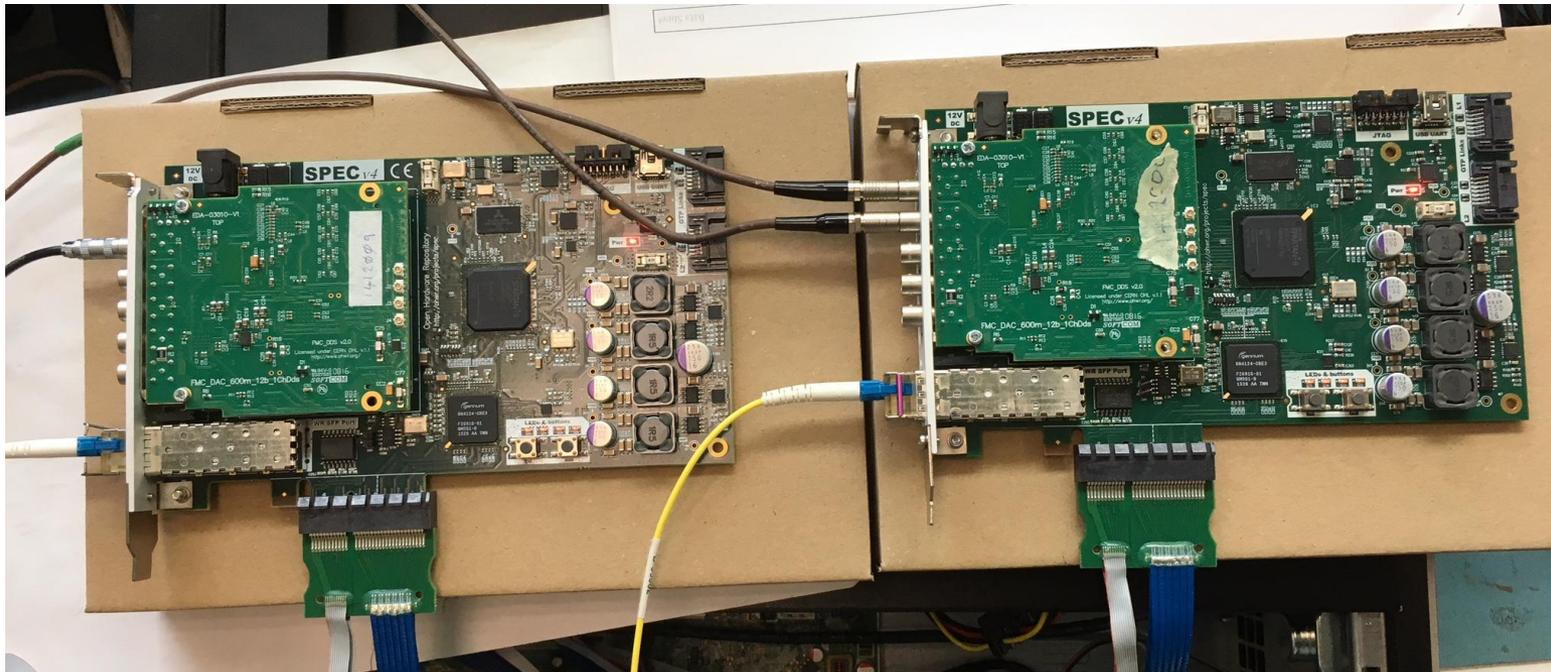
WR Timing ensures and allows us to maintain all the Numerically Controlled Oscillators (NCO) with the same values everywhere.





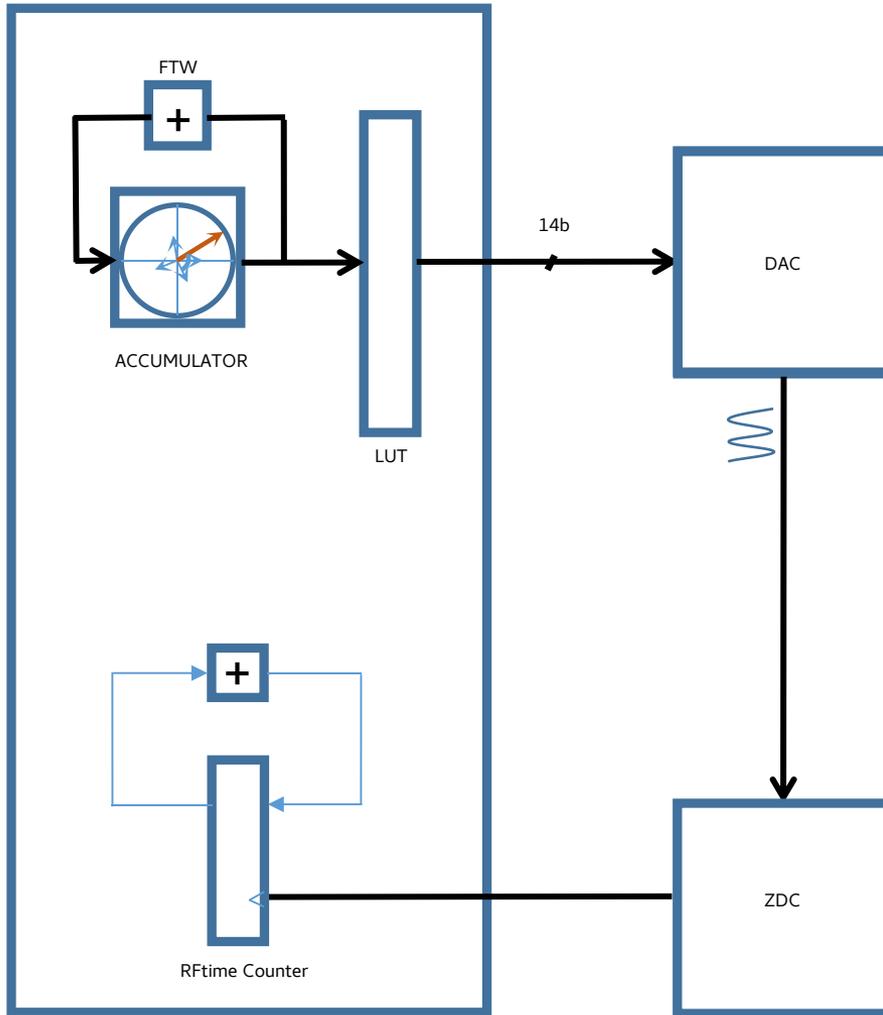
- We are currently developing a RFoWR demonstration platform using the existing [fmc-dac-600m-12b-1cha-dds](#)
- These boards have some known [problems](#). In particular, the ADF4002 anti-aliasing filter and the noise contribution by ADC due to the reference voltage.
- Usage requires gateway (RTL), firmware (embedded C code) and host applications (scripts) plus drivers (around half a dozen).

- Development of a demonstration platform using the [SPEC v4.0 carrier card](#).
- Firmware algorithms heavily based on previous work from TW and ESRF.
- Firmware updated to use the new [Mock-turtle](#) framework, using uRISCv cpu cores.
- Plan to make it available shortly on the OHWR portal. Goal is to give everyone a common starting point for developing their own RF distributed systems.
- Highlighting features for configuration: RF input divide, initial FTW and maximum delay.
- And an initial implementation of Rftime.



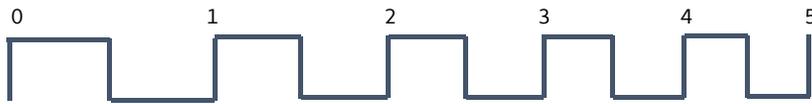
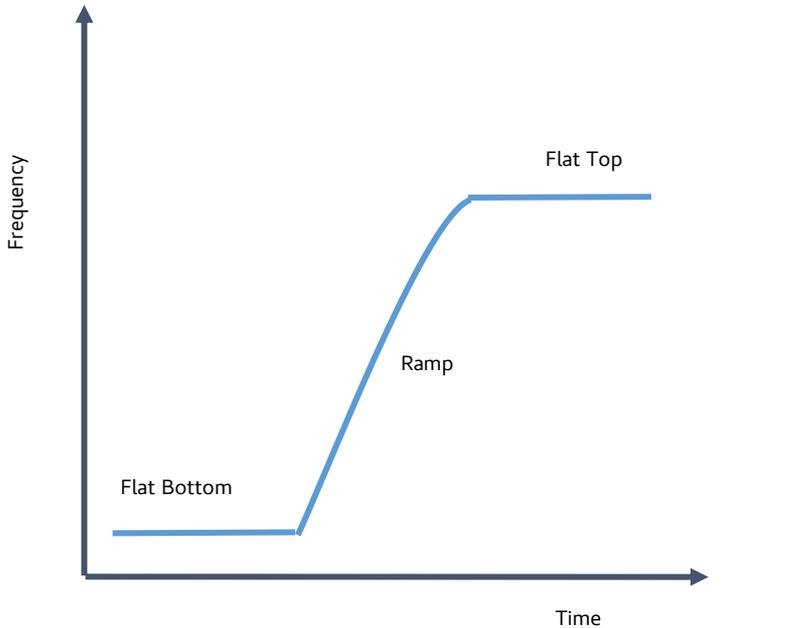
Status - RFoWR demonstration platform

FPGA

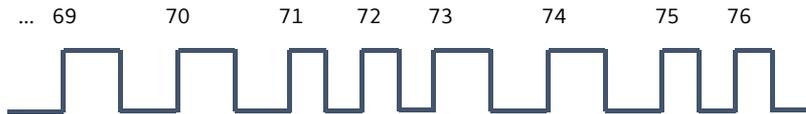


WR clock cycle	Accumulator	Rftime
0	0	0
1	3	0
2	6	0
3	1	1
4	4	1
5	7	1
6	2	2
7	5	2

Rftime is simply a counter and it can be synchronised to be the same in all receivers.



RF Clock with varying frequency



RF Clock with FSK

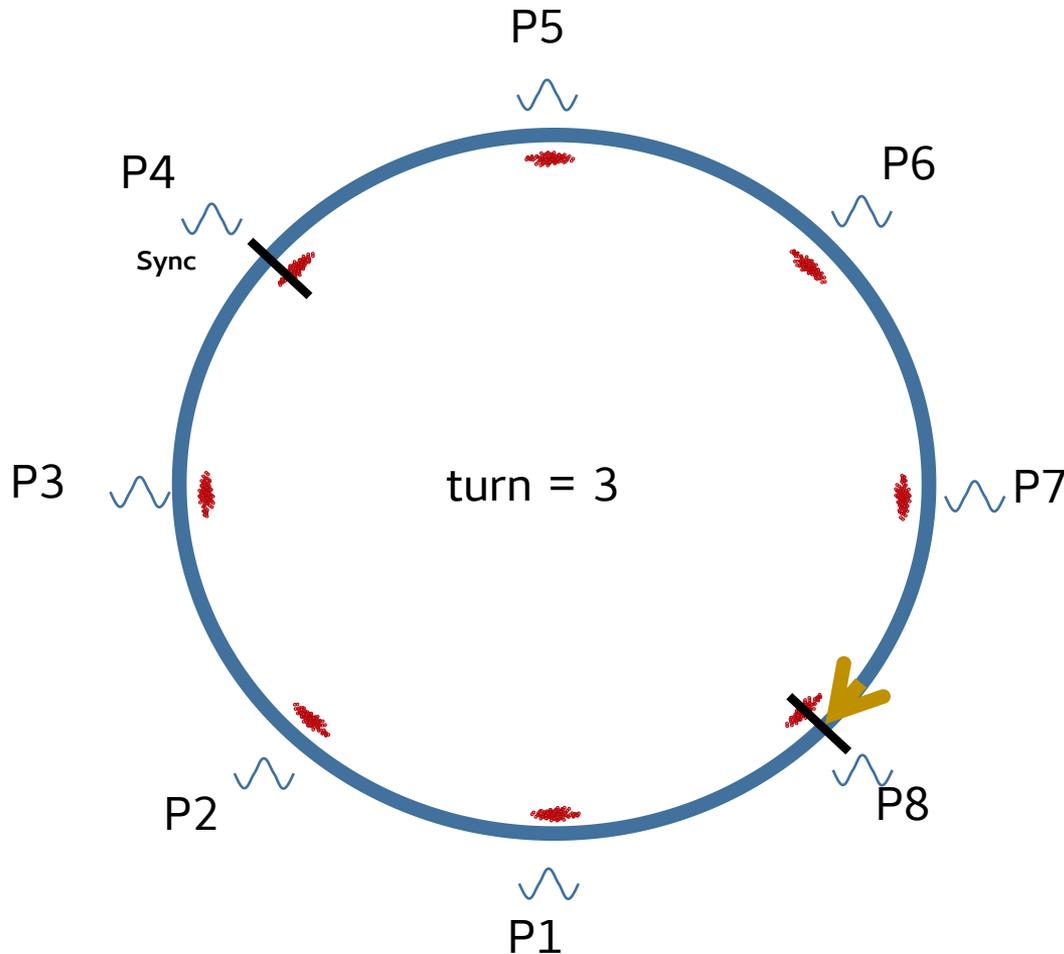
Here at CERN, the input RF is not necessarily constant. The frequency has to change as particles are accelerated.

The RF at injection (flat bottom) into the LHC is ~ 87 Hz or ~ 550 Hz less the frequency reached at the flat top.

For synchrotrons in the LHC injector chain, notably the SPS, this change in RF frequency is even more pronounced.

By expressing events and triggers in Rftime, we remove the need to express events in TAI time and avoid complicated calculations for each receiver.

Furthermore, the changes in FTW or Phase Accumulator can be issued as commands from the master and queued by the receivers to make precisely controlled changes to their NCOs.



Synchrotrons have a harmonic number. This is the number of RF oscillations required to complete a revolution of the synchrotron.

Scheduling events, pulses and triggers becomes straightforward once we adopt Rftime and express it as a function of Turn# and Bunch#.

For instance: generate a pulse at Point8, on turn #3 with a bunch offset of 4.

$$\text{Rftime} = 3 \times 8 + 4$$

$$\text{Rftime} = \text{turn} \times \text{RF harmonic} + \text{bunch}$$

- Release v1.0 of the RFoWR demonstration platform.
- Improvements planned for the PI control loop and accumulator snapshot processes.
- New release of the FMC-DAC v3.0
- Verifying the Rftime counter and concept.
- Packet formats (FTW updates etc) need to be standardised. See later talk about WRXI.
- Engage with WR community to understand your requirements and build a platform we can share.