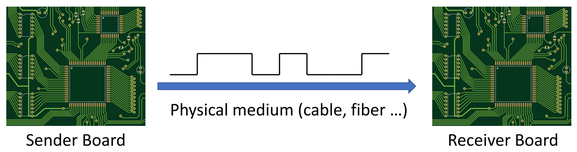
Design and Implementation of a digital CDR in FPGA

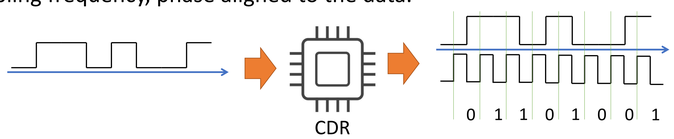
F. Marini[1,2]
[1]University of Padova, Dept. of Physics and Astronomy G. Galilei, Italy
[2]INFN Padova, Italy

Introduction:

The capability to extract timing informations out of a serial data stream to decode the incoming message has become a very common requirement.

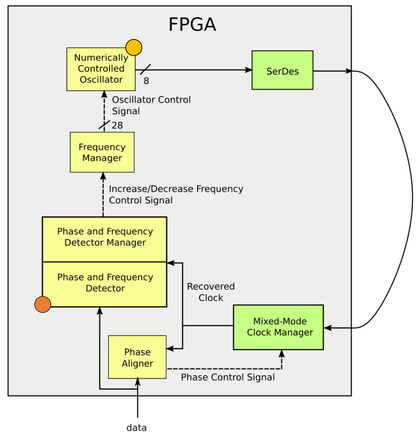


To sample the incoming data, the receiver usually relies on a Clock and Data Recovery (CDR) chip, which generates a clock signal at the corresponding sampling frequency, phase aligned to the data.



The typical CDR architecture is similar to al PLL model, where components like Voltage Control Oscillators (VCO) and Charge Pumps are used. The proposed project is an FPGA (Xilinx Kintex-7) implementation of a CDR, where these usually-used components are not natively available

CDR Architecture:



The main custom components of the design are:

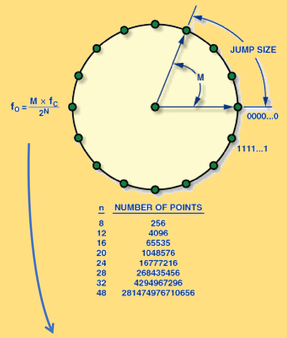
- **Numerically Controlled Oscillator** (NCO). Creates a frequency-controlled clock signal
- **Phase and Frequency Detector** (PFD). Monitors the the NCO's clock frequency to match it with the data-rate.
- **Phase Aligner**. Together with the Mixed-Mode Clock Manager (MMCM) tile featured in Xilinx 7 Series devices, dynamically adjust residual clock drifting and provide a deterministic clock-data phase relationship.

The core uses high-range (HR) GPIO pins of the FPGA over dedicated transceivers, for reduced power consumption and straightforward design. Moreover the core is easily portable to different FPGA architectures.

Numerically Controlled Oscillator:

The design consists of two parts:

- A Phase Accumulator, which acts as a counter incremented by the reference clock.
- A Phase-to-Amplitude converter, which takes the counter output as an index to a pseudo Look-Up Table (LUT).



To understand how the NCO works, we can think of a vector rotating around a phase-circle where to each point correspond a waveform value. One revolution of the vector results in one complete cycle of the output wave.

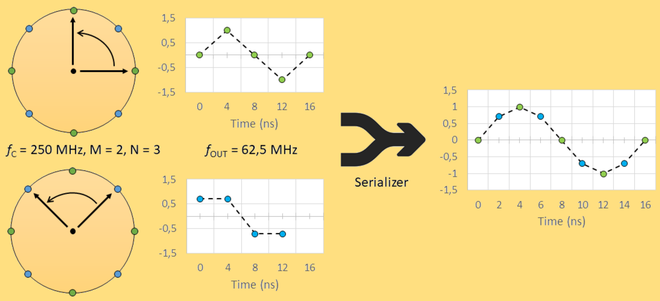
The Phase Accumulator output correspond to the phase-point related to the vector position, while the pseudo LUT assign the waveform value to that point. The vector changes position every system clock cycle, with jumps of "*M*" points.

- f_0 = Output Frequency
- f_C = Reference Clock Frequency
- M = Jump Size
- N = Number of bit for the Phase Wheel

NCO Limitations?

- Maximum output frequency (Nyquist)
- Phase resolution (Reference clock period)

To overcome these limitations, the parallel computation feature of FPGAs is exploited, generating multiple Phase-wheels

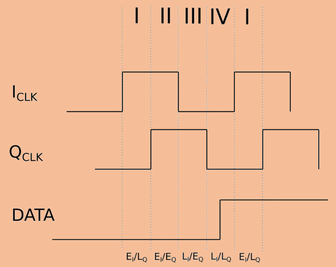


The serialization of the different phase-wheels outputs is carried out by a SerDes (Xilinx OSERDESE2) tile which must be connected to an I/O Block (IOB). Therefore a loopback in and out of the FPGA is a precondition to be foreseen when designing the PCB.

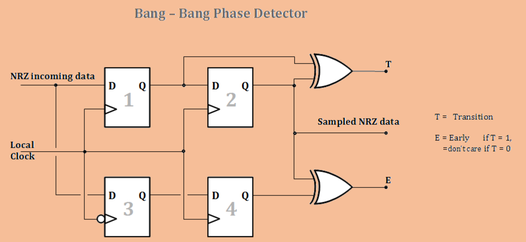
The entire project, as well as documentation files can be found at:

https://github.com/FilMarini/FPGA_CDR_core

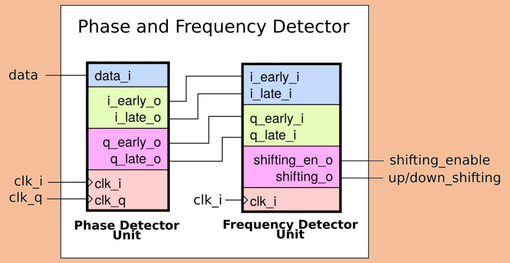
The Frequency mismatch detection capability is provided by dividing the clock period into four quadrants. If the data edge shifts between quadrants, the frequency must be corrected accordingly



Phase and Frequency Detector:

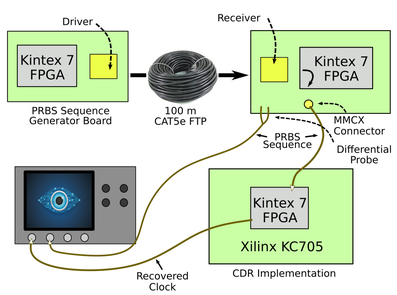


First stage for the data edges quadrant identification is carried out by the Phase Detector Unit which features a couple of Bang-Bang Phase Detectors connected to orthogonal clock signals

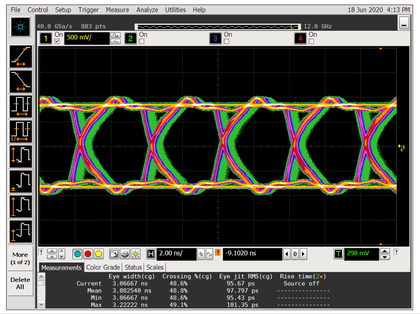


First stage for the data edges quadrant identification is carried out by the Phase Detector Unit which features a couple of Bang-Bang Phase Detectors connected to orthogonal clock signals

CDR Testing:



- 100 m CAT5e cable trip
- PRBS-7 250 Mbps data
- Recovered data eye obtained from CDR recovered clock



- Vertical eye width minimally impacted
- Horizontal width is about 3 ns (out of 4 ns)
- BER > 10^{-12} rejected with CL = 95%