Design of basic receiving functions for an SDR based QPSK base band demodulator for a Reconfigurable Data-Link

Presented by

Ivan Iudice *Embedded Systems and Communication*

i.iudice@cira.it

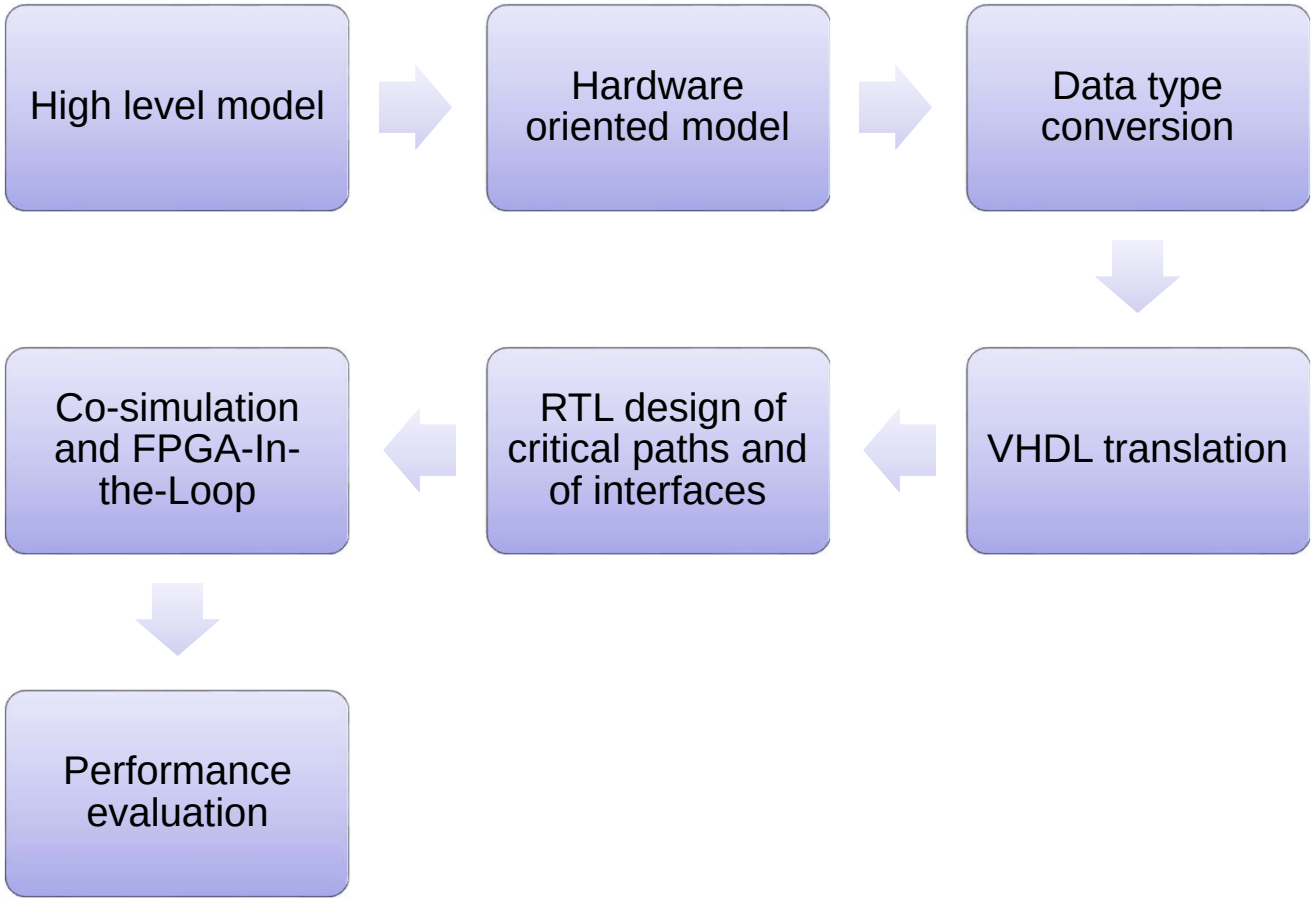
Introduction

- Reconfigurable Data Link (RDL): multiple radio personalities without hardware changing
- Aeronautical communications (UAVs)
- Flexibility and scalability
- Why?
  - Environment link adaption
  - Multi standard operations
- How?
  - Field-Programmable Gate Arrays, General Purpose Processors and Digital Signal Processors
- Good balance among:
  - Computational power
  - Re-configurability
  - Costs
- FPGA
  - Parallelism
  - Embedded hardware
  - Area/power efficiency
- Design effort?

Design flow - 1

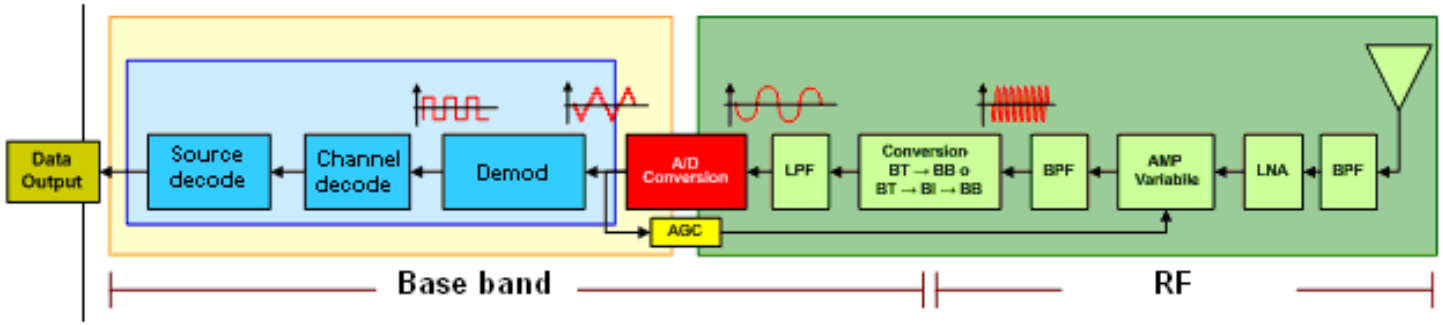
- FPGA design effort:
  - Hardware description with VHDL @ RTL level
  - High level design
- Model-based design
  - Description of the algorithm through models
  - Effectiveness of the simulation
  - Parameters fine tuning
  - Hardware oriented modeling
  - VHDL code integration and optimization

Design flow - 2



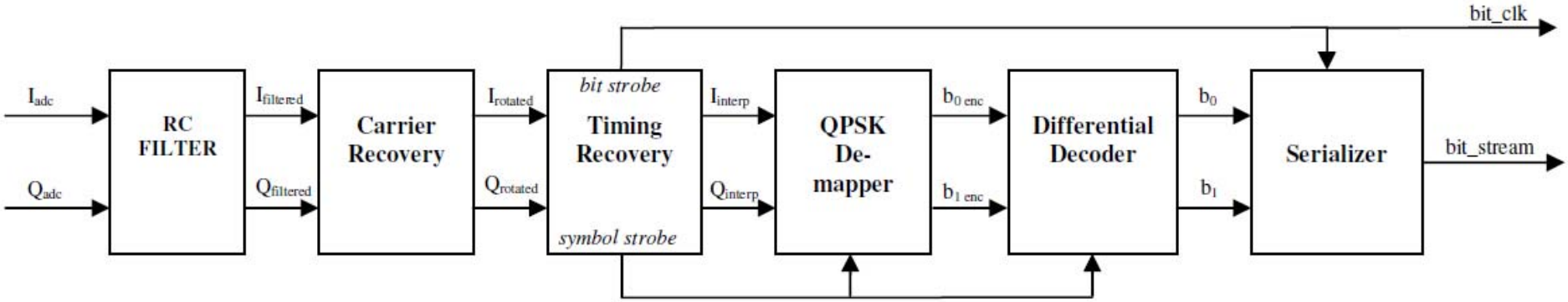
Design of the basic functions

- Library of basic functions, modular systems
- Baseband processing



- Case study:
  - QPSK Demodulator
  - Bit timing recovery
  - Carrier recovery

Case study: QPSK demodulator



$$s(t) = \sqrt{\frac{2E_s}{T}}\, g(t) \cos\left(2\pi f_c t + \frac{\pi}{4}(2m(t)+1)\right)$$

$$m(t) \in \{0,1,2,3\}\ \forall t \in R$$

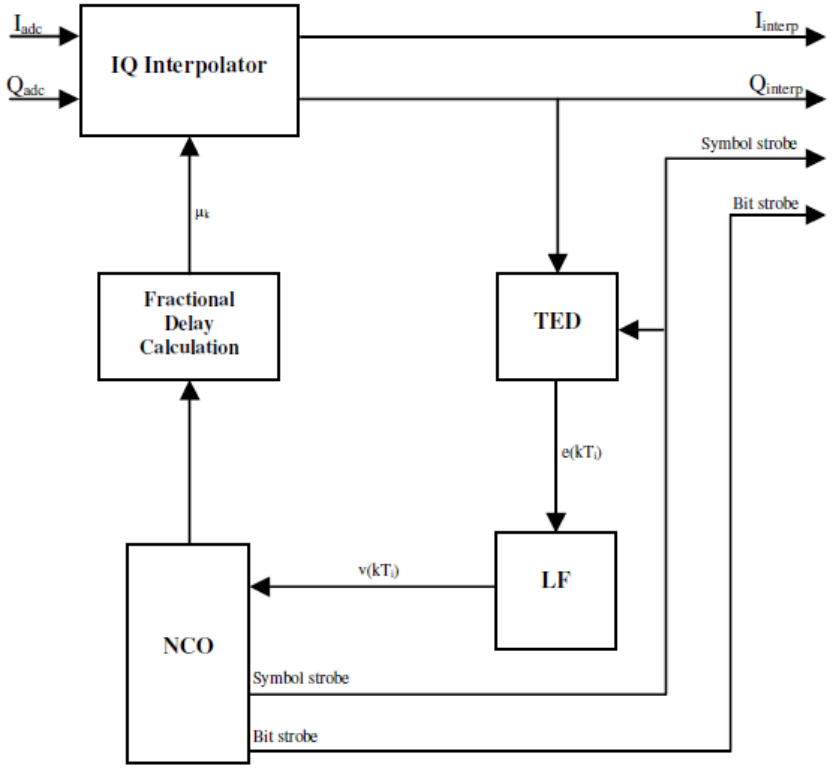
$$I(t) = \sqrt{\frac{2E_s}{T}}\, g(t) \cos\left(\frac{\pi}{4}(2m(t)+1)\right) = \sqrt{\frac{E_s}{T}}\, g(t) m_I(t)$$

$$Q(t) = \sqrt{\frac{2E_s}{T}}\, g(t) \sin\left(\frac{\pi}{4}(2m(t)+1)\right) = \sqrt{\frac{E_s}{T}}\, g(t) m_Q(t)$$

$$m_I(t), m_I(t) \in \{-1,1\}\ \forall t \in R$$

Bit timing recovery – 1

- The sampling period of analog-to-digital converter in the down-conversion stage is not aligned to the symbol period, i.e. it is asynchronous with symbols, and so timing errors affect demodulation performance in presence of noise



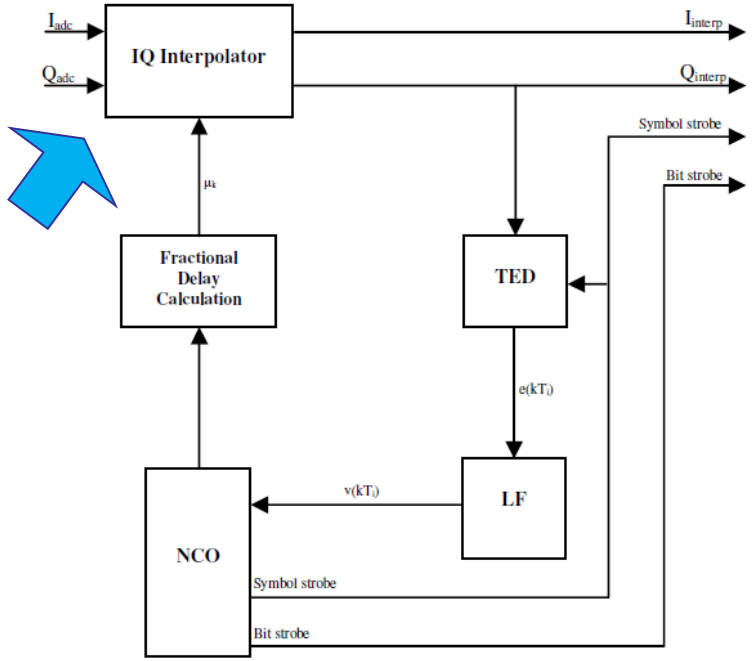
Non-Data-Aided algorithm

Feedback structure

4 samples per symbol

Bit timing recovery – 2

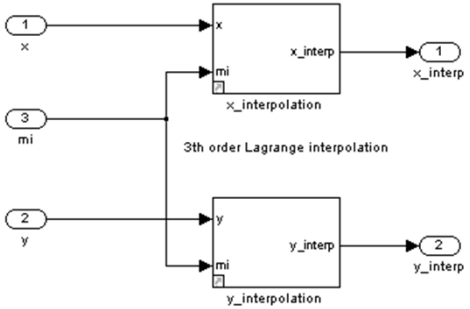
- Re-sampling is possible without changing the sampling frequency of the ADC (All digital timing recovery)
- I/Q signals are interpolated in order to have the “optimum instant” samples



- The interpolant is calculated according to: $q(kT_i) = q((m_k + \mu_k)T_s)$

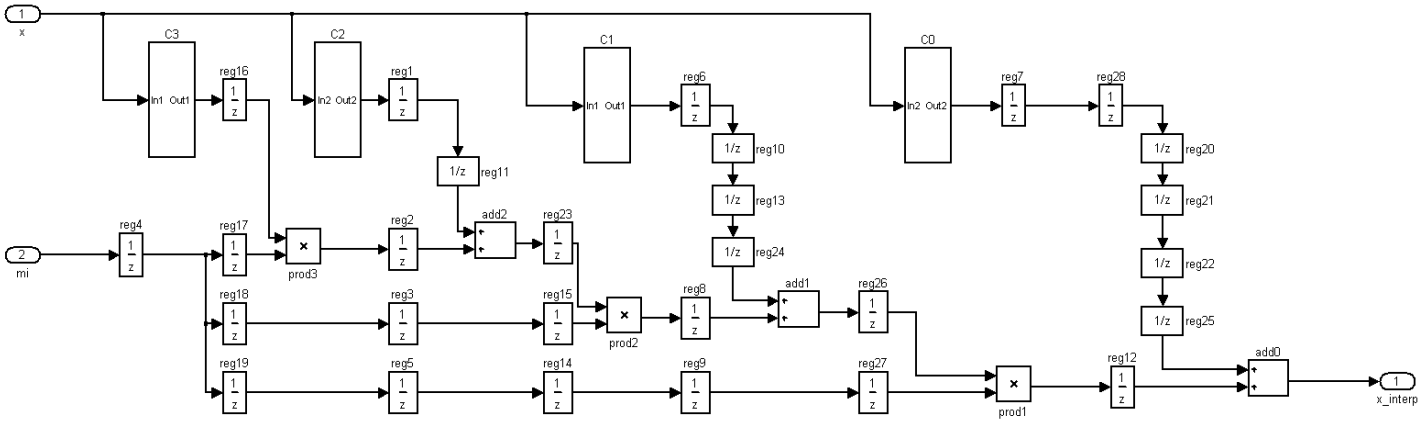
Bit timing recovery – 3

- Lagrange polynomial interpolation arranged in a Farrow structure
- High accuracy but high area occupancy
- Resource sharing, control logics optimization



Interpolation on both channels

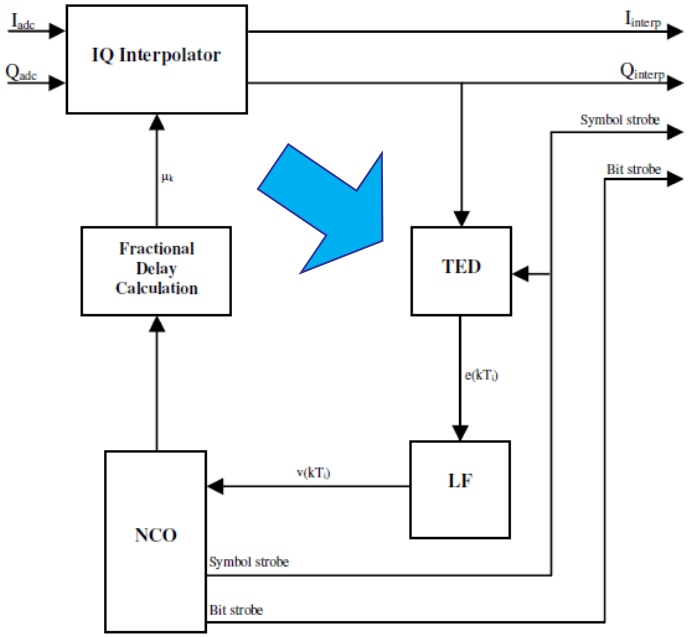
Pipelined data paths



Bit timing recovery – 4

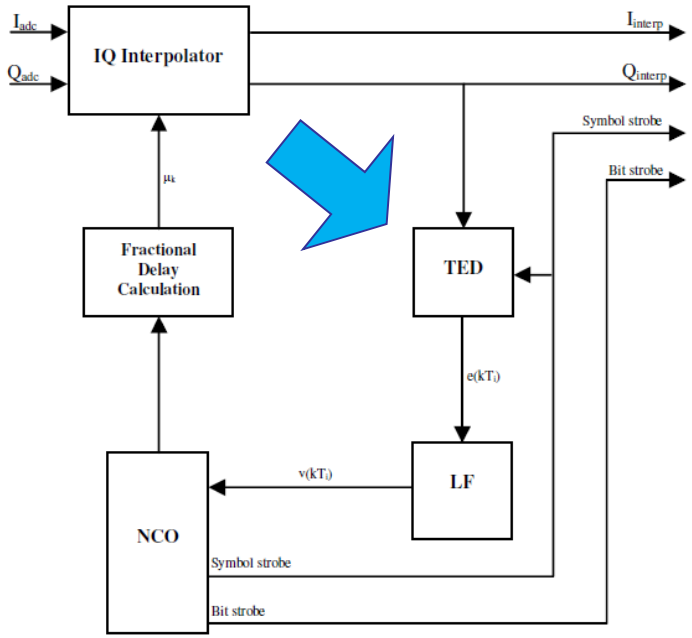
- Timing Error Detector is used to estimate the error between the right sampling instant and the current one
- Non-Data-Aided approach: Gardner algorithm
- It is based on finding zero crossing between two consecutive symbols
- It works with two samples per symbol, the estimated error is:

$$\hat{e}(kT_i) = \left(q(kT_i) - q\left((k-1)T_i\right)\right) q\left(\left(k - \frac{1}{2}\right)T_i\right)$$



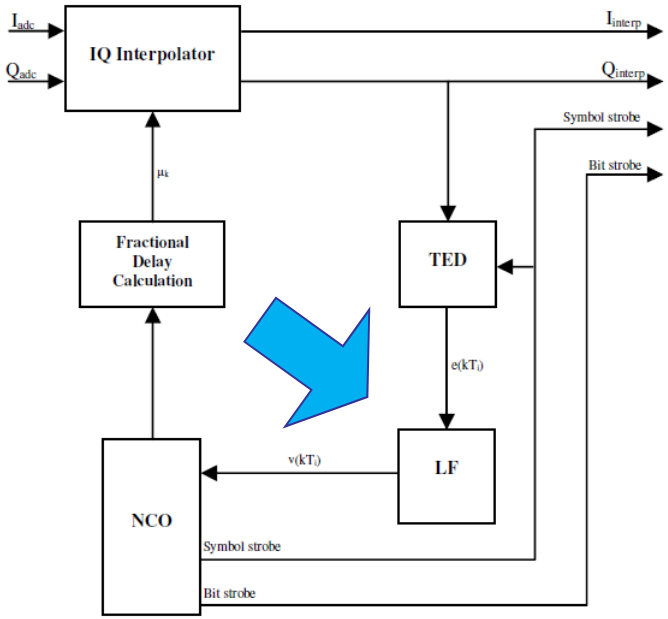
Bit timing recovery – 5

- The TED output is filtered in order to have the control signal for the timing adjustment. A first order Proportional-plus-Integrator (PI) loop filter is used for this purpose.
- The choice of filter coefficients and the data representation are critical
- It is able to track out both phase and frequency error (according to loop bandwidth)



Bit timing recovery – 6

- Interpolation control triggers other blocks in order to select the sample (one out of four) that is at the peak of the symbol, on the basis of the filtered error from TED
- It calculates the distance in time between the desired optimum sample and the closest ADC preceding one. This amount takes the name of fractional delay. On this basis, the interpolator is able to compute the interpolant closest to the optimum sample.

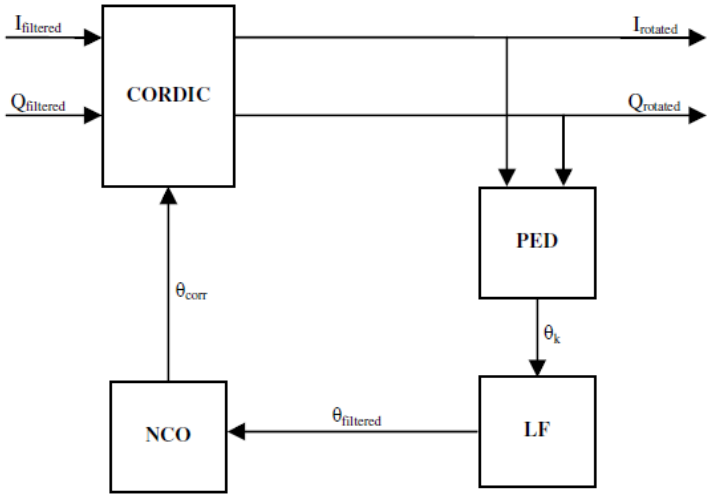


Carrier recovery

- A second-order frequency control loop is used. The carrier error is calculated by a decision directed phase error detector.

$$\theta_k = sign(I_k)Q_k - I_k sign(Q_k)$$

- The filtered phase error is accumulated by NCO and used to rotate the constellation, through the CORDIC, in order to adjust the input signal vector.

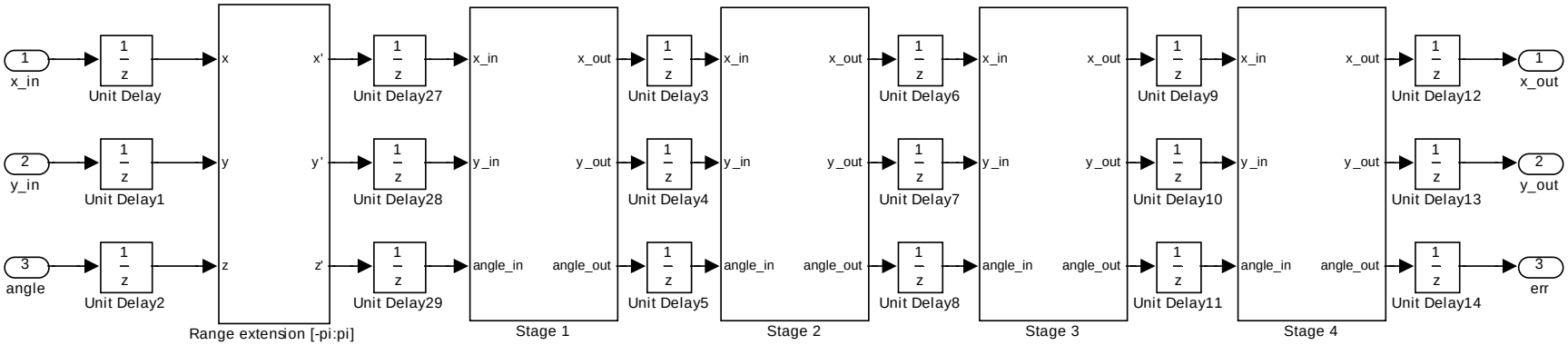


- There is a trade-off between fast acquisition and noise mitigation

$$B_n \approx \frac{\Delta f_{\max}}{2\pi\sqrt{2}\zeta}$$

Phase rotator

- Rotation of signal input vector is made through a CORDIC processor

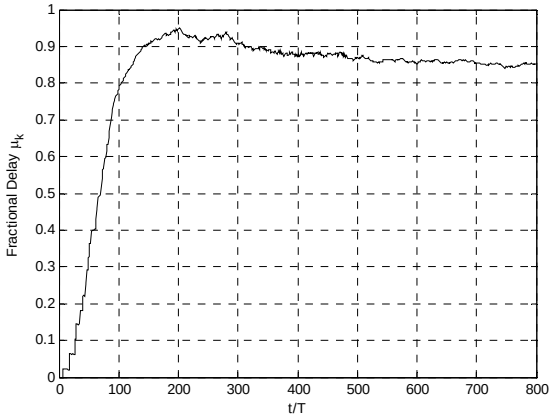


- Hardware oriented model
  - CORDIC: adders, barrel shifters and LUTs
  - Pipelined processor results in reduce critical paths
  - Fixed point arithmetic

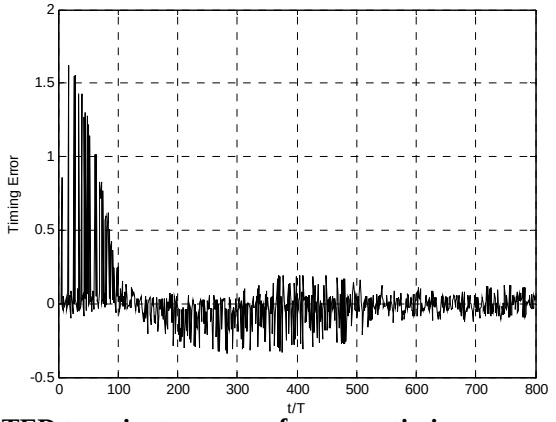
Simulation results – 1

- Simulations verify that the synchronization and demodulation functions work properly.
- A PN11 bit sequence is used as source of the signal waveform.
- The coefficients of the timing loop filter are designed considering a damping factor of $\frac{1}{\sqrt{2}}$ and an equivalent noise bandwidth of 1 % of the symbol rate.
- Timing recovery
  - A step timing error
  - A ramp timing error
- The coefficients of the carrier loop filter are designed considering an unity damping factor and an equivalent noise bandwidth of 1 % of the symbol rate

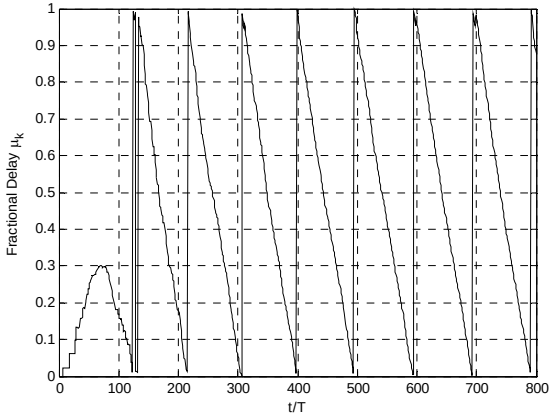
Simulation results – 2



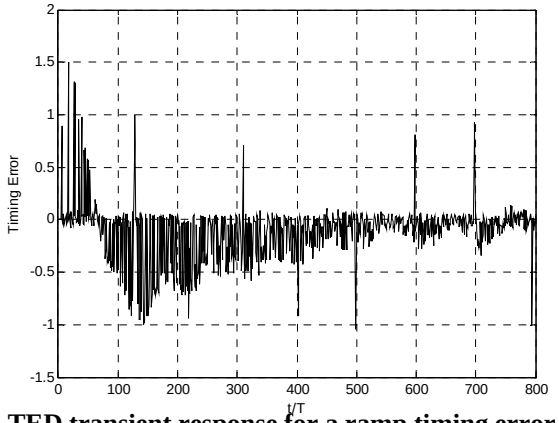
Fractional delay transient response for a step timing error



TED transient response for a step timing error

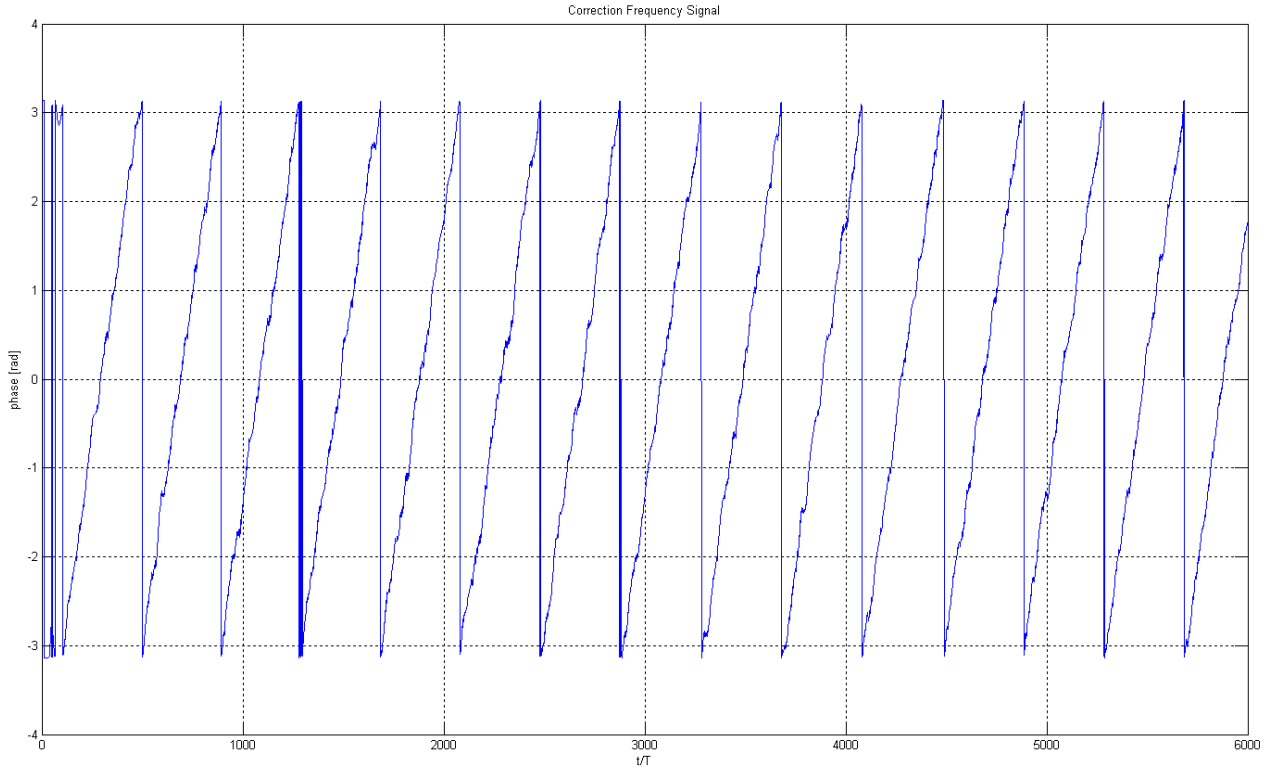


Fractional delay transient response for a ramp timing error (1% of symbol period)



TED transient response for a ramp timing error

Simulation results – 3

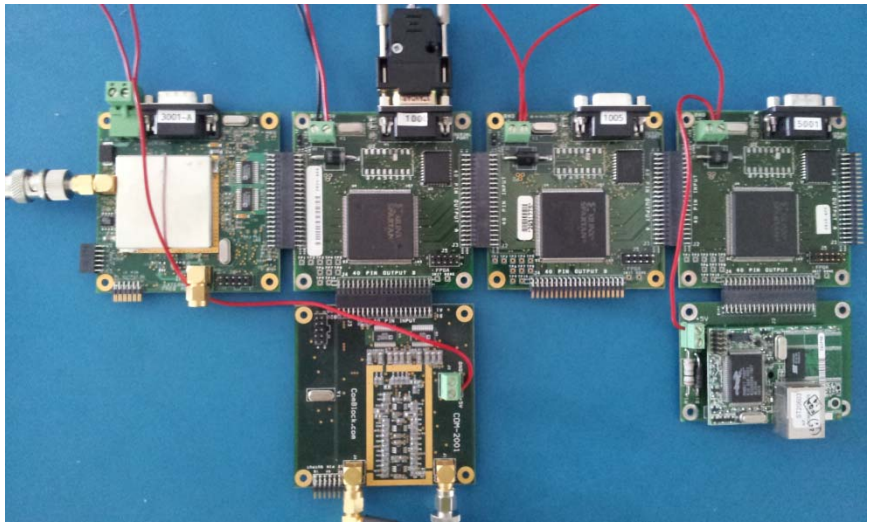


Carrier recovery loop response for a frequency step error

The carrier loop tracks phase error to correctly rotate the constellation.

Future Development

- The system model will be converted in VHDL description
- Co-simulation will validate the design
- After synthesis, VHDL code will be reviewed to meet area and speed requirements
  - At model level
  - At RTL level
- The target hardware is a Xilinx Spartan 6 FPGA
- Comblock® boards are used



Conclusions

Basic receiving functions of a communication system and related design approach were investigated for a QPSK base-band demodulator case with the aim to create a library of modular blocks that can be used to implement a Reconfigurable Data-Link.

A model-based approach was used in the design flow. In such way the transition from a high level block models to a close hardware ones will be simplified and the development effort and time will be considerably reduced.

Q&A