Hybrid Clock and Data Recovery for a High Speed Transceiver implemented on a FPGA

Daniel Cárdenas

***Abstract* – This article describes the clock and data recovery (CDR) subsystem for a 1.25 Gb/s transceiver prototype and 100 Mb/s transceiver and its implementation on FPGA. The CDR block is based on a hybrid approach for computing the optimum sampling instant, i.e. it uses digital signal processing techniques implemented on a logical core (FPGA) in order to extract the instantaneous phase error information and an external VCO for generating the exact sampling clock that drives the ADC at the receiver.**

***Index Terms* – Clock and Data Recovery, CDR, FPGA, DSP, PLL, Synchronization.**

I. Introduction

Clock and Data Recovery is a key element of a communication's receiver. Depending on the characteristics of the transceiver and the whole communication system, different approaches can be taken in order to recover the right clock and data information from the incoming data. For digital systems, one approach uses an analog Voltage Controlled Oscillator (VCO) who drives the receiver's sampler. The error phase information that drives the VCO can be computed in both, analog or digital domain. This article will briefly describe this approach and will show, as an example, a hardware implementation on a Field Programmable Gate Array. Another all-digital approach uses digital signal processing in order to recover the right data information, thus no VCO is needed. Even though the basic initial theory is similar for both approaches (with and without VCO), the all-digital structure is beyond the scope of the present document, please refer to the article "All Digital Timing Recovery and FPGA Implementation" for a better description of such a system and for a complement of the basic theory behind these CDR approaches.

II. Timing Recovery Description

Figure 1 shows a typical baseband PAM communication system where information bits b_k are applied to a line encoder which converts them into a sequence of symbols a_k. This sequence enters the transmit filter $G_T(\omega)$ and then is sent through the channel $C(\omega)$ which distorts the transmitted signal and adds noise. At the receiver, the signal is filtered by $G_R(\omega)$ in order to reject the noise components outside the signal bandwidth and reduced the effect of the ISI. The signal at the output of the receiver filter is

$$y(t;\varepsilon) = \sum_m a_m \, g(t - mT - \varepsilon T) + n(t)$$

Equation 1

where *g(t)* is the baseband pulse given by the overall transfer function *G(ω)* (Equation 2), *n(t)* is the additive noise, *T* is the symbol period (transmitter) and εT is the ***fractional time delay*** (unknown) between the transmitter and the receiver, $|\varepsilon| < ½$. The symbols $\hat{a}_k$ are estimated based upon these samples. They are finally decoded to give the sequence of bits b_k.

$$G(\omega) = G_T(\omega) C(\omega) G_R(\omega)$$

Equation 2. Overall transfer function

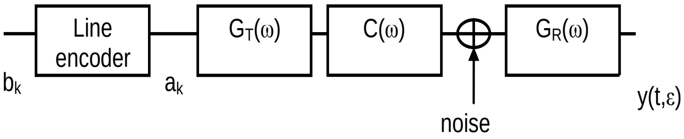

Fig. 1. Basic Communication System for baseband PAM

The receiver does not know a priori the optimum sampling instants {kT+ εT}. Therefore, the receiver must incorporate a timing recovery circuit or clock or symbol synchronizer which estimates the fractional delay ε from the received signal.

Two main categories of clock synchronizers are then distinguished depending on their operating principle: error tracking (feedback) and feedforward synchronizers [1]. In the following the feedback synchronizer will be described.

A. Feedback Synchronizer

The main component of the feedback synchronizer is the timing error detector, which compares the incoming PAM data with the reference signal, as shown in Figure 2. Its output gives

D. Cárdenas is with Electronics Department, Col. Politécnico, Universidad San Francisco de Quito, M103A, Cumbayá, Ecuador (e-mail: dcardenas@usfq.edu.ec).

D. Cárdenas thanks the colleagues at PhotonLab, Istituto Superiore Mario Boella, Torino, Italy, for their economical and technical support. He also thanks Vincent Falchi for the fruitful technical discussions with him.

the sign and magnitude of the timing error $e = \varepsilon - \hat{\varepsilon}$. The filtered timing error is used to control the data sampler. Hence, feedback synchronizers use the same principle than a classical PLL [1, 2].

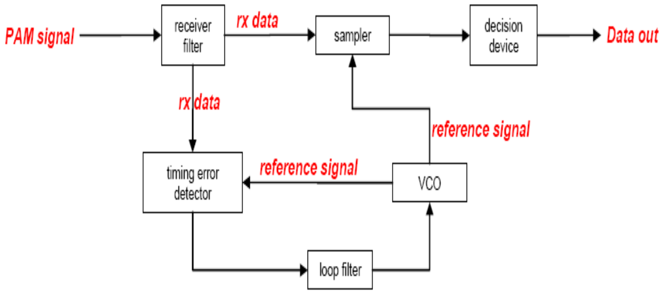

Fig. 2. Feedback (closed loop) Synchronizer

The feedback synchronizer minimizes the timing error signal, the reference signal is used to correct itself thanks to the closed loop.

The synchronizer can also work in continuous or discrete time.

III. Hybrid Synchronizer Architecture

This section describes the general architecture of a hybrid synchronizer which was used for the CDR block of two prototypes, one aimed to work at 100Mb/s and a second one at 1.25Gb/s. Thus, the theory applies for both units, nonetheless, specific details on each one will be given as the theory is analyzed.

We describe an approach named “hybrid synchronizer”, which is partially implemented on the digital domain and partially on the analogue domain; a fully digital synchronizer, which fully operates in discrete time would have an equivalent architecture of a feedback synchronizer; therefore, the theory for computing the loop parameters is exactly the same.

The basic architecture of the hybrid design is depicted in Figure 3. The incoming data is sampled by the A/D and then it is sent to the timing error detector. The error signal goes through a loop filter; its output is converted to the analog domain in order to control a VCO; hence, sampling is synchronized with the incoming signal. The clock generated by the VCO controls not only the A/D converter but also the entire digital logic of the receiver in the FPGA. The initial frequency of the VCO is close to the nominal sample rate.

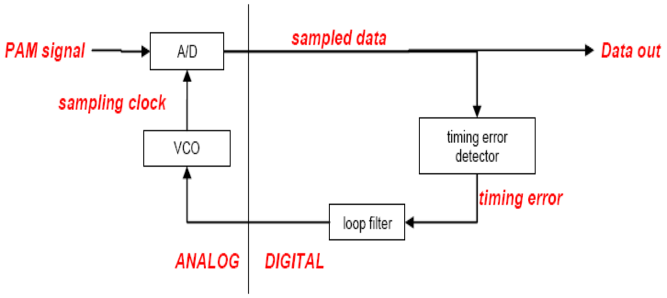

Fig. 3 Hybrid-CDR Architecture

A. Timing Error Detector (TED)

The timing error detector TED resembles the operation of a Phase Detector in an analogue PLL, i.e. it gives the error information based on the phase difference between the incoming signal and the reference clock at its input.

There are several algorithms to implement digitally a timing error detector depending on the oversampling factor or modulation format [1, 3, 4]. The available hardware for the prototypes, specifically the ADC, allows to have at most two samples/symbol (100Mb/s prototype) and one sample/symbol (1.25 Gb/s prototype); moreover, it would be better if its implementation has a good trade-off between complexity and performance, hence, we selected the error detector from Gardner [5, 6] and also the Müller & Mueller algorithm [1], respectively.

Gardner error detector

This algorithm for timing error detection was developed for BPSK/QPSK signals but it works also, with a low penalty, over 8-PAM signals.

The timing error is computed from the input samples accordingly to Equation 3. It uses two samples per symbol interval T. The samples are denoted as x, the indexes (n) and (n-1) denote the samples spaced by one symbol interval (or symbol numbers); the index (n-1/2) denotes the sample lying midway between them.

$$e(n) = x(n - 1/2) \bullet [x(n) - x(n-1)]$$

Equation 3 Gardner TED algorithm

In order to see how this algorithm works, let us examine a received signal with binary antipodal modulation with the two levels defined by -1 and +1; Figure 4 shows an example of such a modulation for two consecutive symbols +1 and -1 sampled at discrete times n-1 and n. Let examine the ideal situation, i.e. when the receiver clock and the transmitter clock are equal and in phase so that there is no timing error. In this case, Figure 4 shows that the symbol samples are taken at the right sampling instants n-1 and n, and the intersymbol sample is taken midway between them. If the symbols at n-1 and n are antipodal, the intersymbol sample x(n-1/2) is null; if the two symbols are equal, their difference x(n)-x(n-1) is zero. In both cases, according to Equation 3, the timing error signal is null as expected since we are analyzing the ideal case.

In a real situation, i.e. when there is a timing error due to a phase shift of the transmitter clock and the receiver clock, the samples are shifted from the ideal ones. Therefore, if the symbols are antipodal, the intersymbol sample is not null and thus the error signal is not null; or, if the symbols are equal, the difference between the respective samples is not null since these samples are shifted with respect to the ideal case and thus, the error signal is not null either.

As usual in clock recovery techniques, a signal with good transition density is desired.

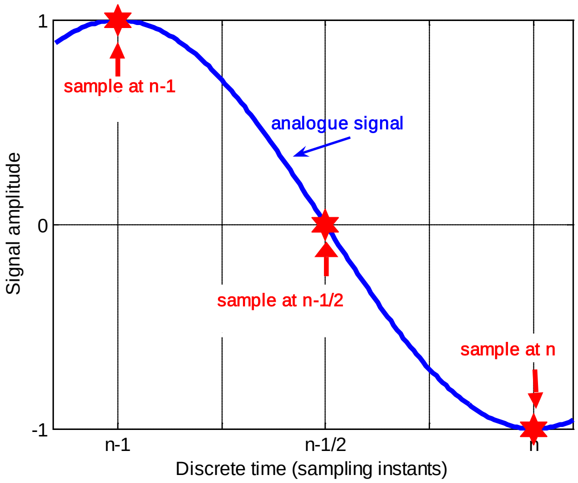

Fig. 4. Binary modulated signal showing the optimum sampling instants (two samples /symbol)

The error information should be always low-pass filtered and thus, in average, the error information can be retrieved. As in analogue Phase Detectors, the sensitivity of the TED can be derived from the S-curve, which is the plot of the phase error (output) versus the phase difference (input), obtained in open loop. Figure 5 shows the S-curve of the Gardner TED, in this case it gives the average TED's output when varying the timing error. The plot was obtained for 64 samples per symbol at the transmitter; hence, the delay was varied from 0 up to 64 samples. The delayed signal was down-sampled to 2 and then filtered by a FIR filter equivalent to the channel before being passed to the TED. All of these parameters contribute in the determination of the gain Kd of the TED, which is the slope of the S-curve at the zero crossing point and thus measured in V/rad. This gain is one of the basic parameters to design the loop filter described later. The TED's gain Kd in this case was 0.9 V/rad.

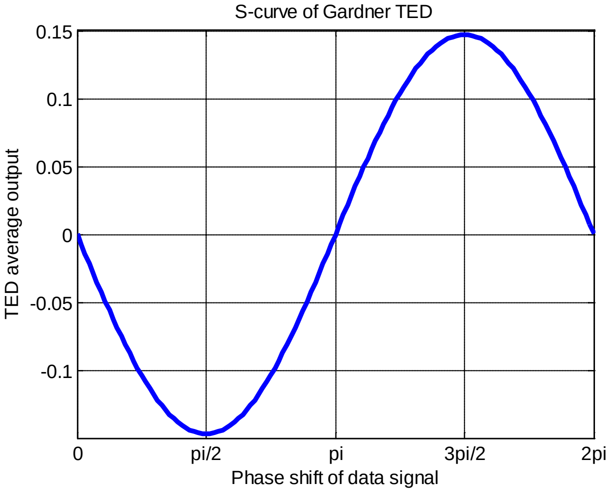

Fig. 5 S- curve of the Gardner TED

Müller & Mueller TED

The description of this TED is similar to the Gardner's algorithm, except that this one uses only one sample/symbol. This algorithm extracts the timing information from the received symbols (x_i) and the corresponding decided values (a_i) as in the following equation:

$$e(k) = x_k a_{k-1} - x_{k-1} a_k$$

Equation 4 M&M algorithm

The TED's gain Kd in this case was 0.38 V/rad.

B. Loop filter and closed loop analysis

Consider the basic PLL in Figure 6, where θ_i is the phase of the incoming signal and θ_o is the phase of the VCO signal. The PLL includes a phase detector with gain *Kd* in [V/rad], a loop filter *F(s)* and a VCO with gain *Ko* [rad/s/V]. The system transfer function is given by:

$$H(s) = \frac{\theta o(s)}{\theta i(s)} = \frac{KdKoF(s)}{s + KdKoF(s)}$$

Equation 5

Where $\theta_o(s)$ is the Laplace transform of θ_o, and $\theta_i(s)$ is the Laplace transform of θi.

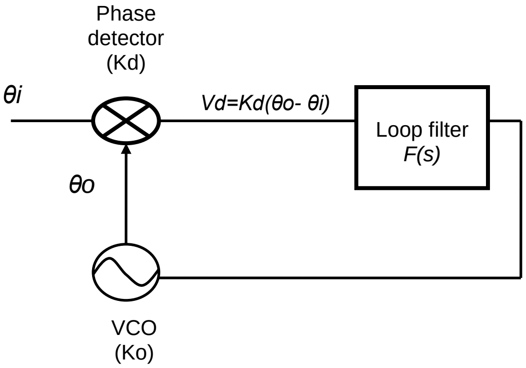

Fig. 6. Basic PLL

The order of the denominator of H(s) gives the order of the loop; the design of the loop filter affects the behavior of the PLL [7].

A second order loop is able to track phase and frequency steps of the incoming signal referred to the VCO signal, a desired feature in the prototype design. A second order loop can be obtained placing a loop filter with a proportional plus an integral path. This loop filter is given by:

$$F(s) = -\frac{s\tau_2 + 1}{s\tau_1} = -\left(K_1 + \frac{K_2}{s}\right)$$

Equation 6

where: $K_1 = \tau_2 / \tau_1$, and $K_2 = 1 / \tau_1$, and τ_2, τ_1 are the time constants for this filter.

The system transfer function becomes now:

$$H(s) = \frac{KdKo(K_1 s + K_2)}{s^2 + sKdKoK_1 + KdKoK_2}$$

Equation 7

The PLL is then a *second order type 2 PLL*.

For second order type 2 PLL exclusively, is better to define the *loop parameters*: *natural frequency* ω_n and *damping factor* ζ; and then determine the gains and the time constants.

The system transfer function now is expressed as:

$$H(s) = \frac{2\,\zeta\omega_n\, s + \omega_n^2}{s^2 + 2\,\zeta\omega_n\, s + \omega_n^2}$$

Equation 8

where

$$\omega_n = \sqrt{\frac{KdKo}{\tau_1}} = \sqrt{KdKoK_2}$$

$$\zeta = \frac{\tau_2\omega_n}{2}$$

Equation 9

The loop can be now designed based on these loop parameters. Usually a damping factor of 0.707 is chosen and the other is obtained accordingly; actually, an analysis of the performance of the loop is carried out in order to set an optimal value of natural frequency. The damping factor is linked to the stability and time-response of the closed loop. Decreasing the damping factor means that the time-response of the system is smaller but the system could become unstable. In any case, such an analysis has to take the natural frequency also into account.

Note that the gain of the phase detector *Kd*, and the one of the VCO *Ko*, must be known. In this case, the counterpart of the phase detector is the TED, whose gain was already evaluated.

The analogue loop filter *F(s)* must be transformed to the digital domain *F(z)* in order to be implemented in the FPGA.

IV. FPGA Implementation and Results

The implemented architecture of the hybrid CDR is depicted in Figure 7. This was implemented on both prototypes.

The hybrid CDR has been designed and tested in simulations; it has also been the first practical solution available in FPGA.

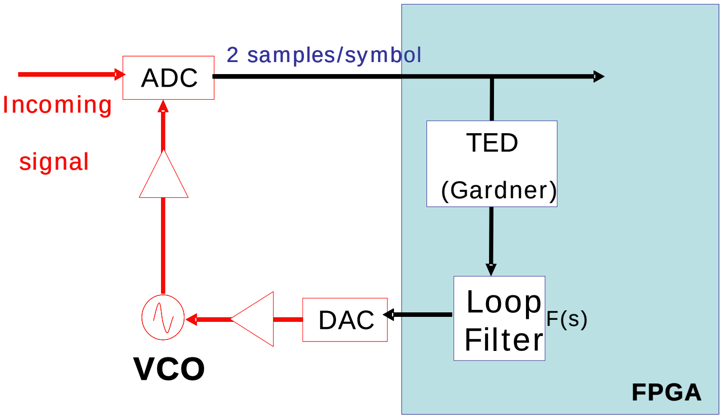

Fig. 7. Hybrid CDR implementation

The transmitter consists on a pseudo-random binary sequence (PRBS) generator followed by the multilevel-PAM modulator. Due to some signal processing, the signal is sent at 41.66 Mbaud (symbol period of 24 ns) for the first prototype and 1.1Gbaud for the second prototype.

In the case of the 100Mb/s prototype, at the receiver, the samples obtained by the ADC go to the hybrid CDR. The hybrid CDR closes the loop by sending the control signal to the VCO by means of a DAC. The loop shall recover a frequency equal to 83.33 MHz, or a period of 12 ns, since the ADC gets two samples per symbol.

The clocks from the transmitter, as well as the recovered clock at the receiver, goes to a time counter. The time counter is able to measure, apart from absolute values of frequencies (or periods), the frequency (period) ratio and the time interval between the two channels, which is the parameter we are interested in. These data are sent to a PC for further statistics.

Some initial experiments were carried out with an external electrical loop with short length cables, instead of the real channel; therefore, no pre-equalization nor non-linearity compensation were considered. The experiment had the purpose to resemble the good simulations results but in a longer period. The loops tested were designed for four natural frequencies (f_n): f_n =60 kHz, f_n =70 kHz, f_n =80 kHz and f_n =90 kHz.

Table I shows the standard deviation of the time interval between the transmitted and the recovered clock obtained for these configurations; 1200 measurements were considered on each configuration. It should be noted that the mean value of the measurements is meaningless since it represents the absolute phase shift between the transmitter and the receiver clocks. On the contrary, the standard deviation indicates how the recovered clock varies.

TABLE I
TIME DIFFERENCE BETWEEEN THE TRANSMITTED AND THE RECEIVED CLOCK, NOMINAL PERIOD: 12 ns

Natural frequency	**Standard deviation**
[kHz]	**[ns]**
60	0.48
70	0.70
80	0.44
90	0.56

A quick look to these values would lead to a first conclusion that the recovered clock jitters half a nanosecond, which represents a 4% of the transmitted clock period of 12 ns; in any case, a longer measurement could draw better conclusions. Actually, as was previously mentioned, the absolute value of the time difference between the recovered and the transmitted clock is indeed meaningless *only* if it is not greater than the bit period; otherwise it means that a "jump" of an entire cycle has occurred; this problem is known as a cycle slip in PLL theory.

Later on some changes were also included in its hardware implementation. A reduction of the natural frequency of the loop was the outcome of this first debug. A smaller natural frequency involves a smaller "initial" frequency difference between the VCO and the transmission clock (acquisition window). This involved slight changes in the board containing the external VCO, so that its frequency in open loop was closer to the nominal one. The system was tested with simulations and also with the real transmission channel, initially with a f_n =35kHz.

Table II shows the results obtained with the new loop with *fn*=35kHz in a system with a real transmission channel.

TABLE II
TIME DIFFERENCE BETWEEN THE TRANSMITTED AND RECOVERED CLOCK AND MEASURED BER. CDR LOOP WITH FN=35KHZ,NOMINAL PERIOD:12 ns

Time difference (standard deviation)	0.30 ns
BER	~ 10^{-8}

The obtained BER involves a penalty of 3 dB with respect to the system where the receiver clock is the same as the transmitter one.

These are the best results we have obtained until now for the 100 Mb/s prototype and in any case, the CDR analysis is still being carried out.

In the case of the high speed prototype (1.25 Gb/s), simulation results show that the hybrid CDR works well. It should be noted that the hardware implementation (on the FPGA and on analog electronics) in this case is by far much more complex than the low speed prototype. In this case the receiver has a very sensible VCO nominally at 1.1 Gb/s and a high speed ADC. Moreover, when combining the CDR block with other subsystems and taking into account the effect of the channel in the eye closure, the implementation is harder and the performance could be degraded.

The simulations, in any case, show a steady eye diagram (Figure 8) of the incoming sampled signal even in the presence of noise and data jitter, illustrating that the correct frequency of the incoming stream is recovered. Note that the results also take into account the effect of the real hardware implementation, i.e. real hardware behavior with finite arithmetic logic and real channel limitations.

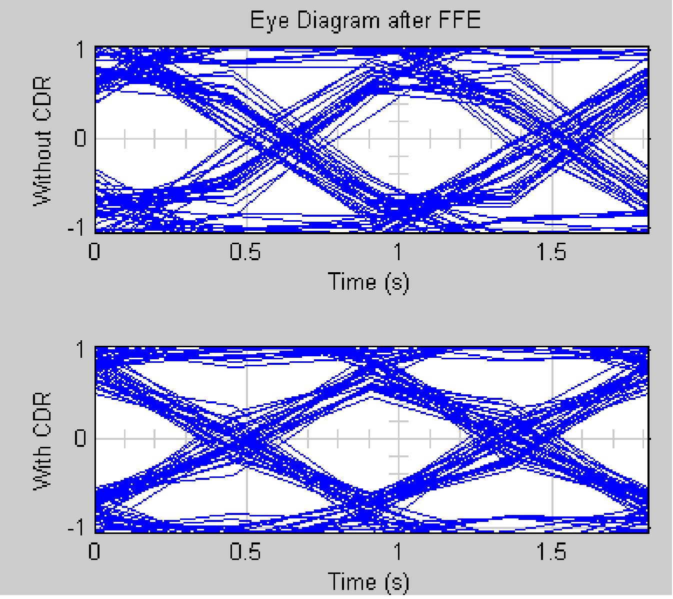

Fig. 8.Eye diagram with CDR is stable while the one without CDR is constantly moving during simulation

Figure 9 shows the VCO's control signal for a step frequency difference between transmitter and receiver. It should be noted that the control signal converges to that step difference.

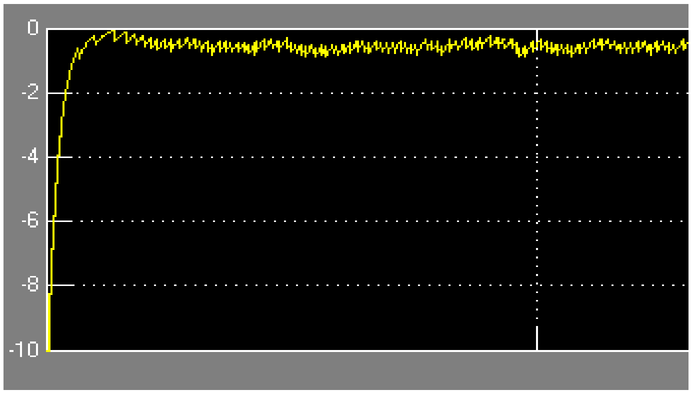

Fig. 9. VCO's control signal

The CDR block is now being implemented and tested with the whole structure of the receiver on a FPGA.

V. Acknowledgements

The author would like to thank the PhotonLab Staff at ISMB, Torino, Italy for the fruitful discussions with them.

VI. References

[1] Meyr Heinrich, Marc Moeneclaey, and Stefan A. Fechtel, *"Digital Communications Receivers: synchronization, channel estimation and signal processing"*. Vol. 2. 1998: John Wiley & Sons.

[2] U.Mengali, A.N.D'Andrea, Synchronization Techniques for Digital Receivers. 1997, New York: Plenum Press.

[3] K. H. Mueller and M. Müller, Timing Recovery in Digital Synchronous Data Receivers. IEEE Trans. Commun., May 1976. COM-24: p. 516-531.

[4] M. Oerder and H. Meyr, Digital Filter and Square Timing Recovery. IEEE Trans. Commun, May 1988. COM-36: p. 605-612.

[5] Gardner, F.M., A BPSK/QPSK timing-error detector for sampled receivers. IEEE Transactions on Communications, 1986. CM-34(5): p. 423.

[6] M. Oerder and H. Meyr, Derivation of Gardner's Timing Error Detector from the Maximum Likelihood Principle. IEEE Trans. Commun., June 1987. COM-35: p. 684-685.

[7] Gardner, F.M., Phaselock Techniques. Third ed. 2005, New York: Wiley.

VII. Biography

Daniel Cárdenas received the Eng. degree in telecommunications engineering from Escuela Politécnica Nacional, Quito, Ecuador, and the M.Sc. (optical communications) and Ph.D. (electronics engineering) from Politecnico di Torino, Torino, Italy in 2004 and 2008, respectively.

He is a coauthor of a patent in the field of telecommunications over plastic optical fibers. He collaborated with Istituto Superiore Mario Boella, Torino, Italy and he was a external consultant for Francecol, France, and Siemens, Germany. He is currently with USFQ, Ecuador. His current research activities include polymer-optical-fiber-based access systems, and digital signal processing techniques and their implementation on FPGA/DSP.