

An FPGA-emulation-based platform for characterization of digital baseband communication systems

*Original*

An FPGA-emulation-based platform for characterization of digital baseband communication systems / LAGOS BENITES, JORGE LUIS; Grosso, Michelangelo; SONZA REORDA, Matteo; Audisio, G.; Pipponzi, M.; Sabatini, M.; Avantaggiati, V. A.. - (2011), pp. 391-398. ( International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems Vancouver, B.C. (CA) Oct. 3-5, 2011) [10.1109/DFT.2011.1].

*Availability:*

This version is available at: 11583/2443375 since:

*Publisher:*

IEEE

*Published*

DOI:10.1109/DFT.2011.1

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

(Article begins on next page)

# An FPGA-Emulation-based Platform for Characterization of Digital Baseband Communication Systems

J. Lagos-Benites, M. Grosso, M. Sonza Reorda

Dipartimento di Automatica e Informatica

Politecnico di Torino

Torino, Italy

{jorge.lagos, michelangelo.grosso, matteo.sonzareorda}@polito.it

G. Audisio, M. Pipponzi, M. Sabatini

Pirelli Tyres S.p.A.

Milano, Italy

{giorgio.audisio, mauro.pipponzi.ex, marco.sabatini}@pirelli.com

V.A. Avantaggiati

Studio Ingegneria dei Sistemi

Milano, Italy

vito.avantaggiati@engsys-vaa.com

**Abstract**— Integrated transceivers play a leading role in an ever-increasing range of systems and applications where dependability is a major concern. In order to reduce time-to-market, it is crucial to verify the correctness of the hardware and software implementations and the overall system performance at early stages of the design flow with respect to specifications or a higher-level model. To this purpose, we identify some specific requirements of these systems, and propose a new methodology for early validation and BER characterization taking into consideration channel noise and sampling frequency offset of the digital baseband communication subsystems. The proposed methodology is based on a low-cost FPGA platform and enables precise control of noise injection and clocking parameters during emulation, so as to reproduce channel and receiver non-ideality effects, while accelerating the validation process by orders of magnitude with respect to simulation-based alternatives. Experimental results on an industrial case consisting of the digital baseband portion of an ultra-wideband wireless link confirm the effectiveness of the approach.

**Keywords**—digital baseband, validation, characterization, FPGA, noise, clock frequency offset.

## I. INTRODUCTION

Integrated digital communication systems are present in a wide range of applications, from entertainment to industrial fields, from telecommunications to transportation, to medical applications. Such systems may include microprocessors, peripherals and application-specific hardware modules, as well as real-time software components. As a direct consequence of the growth in complexity and required performances, the validation, verification and characterization operations become increasingly problematic. Simulation-based techniques are usually employed to ascertain the compliance to specifications and to locate the possible causes of undesired behavior, but simulation time grows more than linearly with the addressed circuit size and component number.

Essential measures of reliability for digital communication systems are the bit error rate (BER), block error rate (BLER) [1] or packet error rate (PER), which constitute fundamental requirements for the design and development of a product. Several stand-alone BER testers [2] are available, which are usually employed on the manufactured communication device at the RF level. Additionally, they require peculiar care for performing repeatable measurements, and generally have limited flexibility and relevant costs.

Hardware-based prototyping of communication systems at the baseband level has several advantages over RF testing using commercial BER testers, including the following ones [3]:

- It enables functionality and performance verification since early design phases. Early discovery and correction of possible design flaws reduces complexity and costs of the required design modifications, and may help avoiding useless silicon re-spins.
- It provides a way to designers to evaluate different design solutions and parameters, and chose the best suited ones for the specific addressed purpose. System trade-offs can be evaluated in a faster way with respect to simulation, and then the selected parameters can be hardwired when getting to silicon.

- It facilitates debug operations of software running within the system, helping, thanks to an acceleration by orders of magnitude with respect to simulation, to evaluate a large number of use cases in mission-like conditions.

This paper proposes a methodology for verification and early BER characterization of digital baseband communication subsystems based on a self-contained FPGA platform. In our previous works [4][5], we developed a co-verification and debug environment for System-on-Chip (SoC) designs, implemented on a low-cost FPGA board. Thanks to an embedded microprocessor mapped on the FPGA and used as interface to the design under test (DUT), complete controllability on the DUT is achieved, including I/O signals, clocks and memories, so as to perform hardware validation/verification and supporting software debug. The proposed BER characterization methodology is built on such platform and provides additional features specifically designed for the digital baseband sections in communication systems. The main additions consist in the development of a generic digital channel model, with the possibility of injecting additive noise samples, and in the realization of a flexible clock domain management system allowing the emulation of sampling clock offset (SCO) effects. The causes of SCO may include different crystal parameters for TX and RX systems or imperfect clock recovery.

The proposed methodology is currently being employed for the verification and characterization of the digital baseband subsystem of an ultra-wideband (UWB) impulse radio (IR) that is an essential part of the Pirelli CyberTyre System. In such system [6], currently in advanced development phases, an accelerometric sensor node is to be attached to the tire inner liner and transmit real-time information to an on-car receiver; the received information can then be used to improve stability and brake control systems. The verification of correct implementation of the circuitry composing the radio link is therefore of crucial importance for the application.

The paper is organized as follows. Section II reviews the main concepts and the state-of-the-art of BER characterization of digital baseband systems, and the general features of our previously developed co-verification and debug environment. Section III describes the proposed methodology in detail, while Section IV presents its implementation. Experimental results are presented in Section V, while Section VI concludes the paper.

## II. BACKGROUND

### A. BER testing

A general digital communication system is composed of a transmitter, a communication channel and a receiver. The most common physical channels for electromagnetic-based communications are electrical connections (e.g., wires), optical media (e.g., optical fibers) or open air (in case of wireless links). The additive white Gaussian noise (AWGN) model [7] applies to a wide range of physical communication channels (including non-electromagnetic channels). The transmitter and the receiver include digital processing and encoding/decoding modules, and may involve RF signal modulation/demodulation sections.

A typical performance indicator for digital communication systems is BER, which corresponds to the ratio between the number of incorrect received bits and the total number of transmitted bits. The expected BER performance of a communication system depends on the modulation/demodulation and coding/decoding schemes, the implemented transmission channel and its physical characteristics which result in signal noise, and non-idealities of the receiver, including receiver noise figure, amplifier non-linearity, phase noise, DC offset, carrier offset and clock frequency offset between transmitter and receiver, etc.: therefore, when characterizing a true communication system, we need to measure the BER and to compare it to the expected BER performance, in order to measure the performance degradation with respect to theoretical expectations, caused by these impairments.

The signal-to-noise ratio (SNR) is a measure used to quantify how much the received signal has been corrupted by noise. It is defined as the ratio of signal power to the noise power corrupting the signal. When designing a communication system, the signal bandwidth and SNR have to be balanced to maximize the channel capacity according to the fundamental Shannon law [7], in order to achieve the BER performance required by the application.

To evaluate the overall performance of a communication system and verify the correctness of its circuital realization, the theoretical and experimental BER versus SNR curves are fundamental. Obtaining such result early in the design flow is key to locate and correct possible design and implementation flaws before they get to the final product, and to optimize the software routines running in the system.

The reference models for a specific architecture are usually obtained using modeling/simulation tools such as MATLAB and Simulink. Performance evaluation of a circuit implementation through simulation soon becomes prohibitive, especially when dealing with high SNR and low BER values, where the transmission of a very large number of packets has to be performed. Emulation techniques have been proposed by [3] and [8] and several commercial platforms. Usually, such platforms do not allow to implement and control a complex digital system, possibly including microprocessor cores and different I/O interfaces. Noise injection features are quite common, while other non-ideal effects are not usually emulated. In [9] a channel emulation system for Orthogonal Frequency-Division Multiplexing (OFDM) communications is presented which includes the emulation of several channel and receiver non-ideality effects; however, the employed models and the general system complexity make it less apt to be used in early digital baseband verification flows, where the channel parameters may still not be fully known.

### B. FPGA-based co-verification and debug platform

In our previous work [4][5], we presented a co-verification and debug system based on full FPGA emulation of a complex device under test (DUT, possibly including a microprocessor) and on a dedicated infrastructure-microprocessor-based interface, in parallel with a suitable software environment. This enables the early validation and debug of the hardware design and also of any software running into it. Moreover, the emulated system is as close as possible to the final implementation in hardware and software. No verification- or debug-related software routines are needed to be run into the system under verification to increase observability, and no hardware modifications are applied, thus avoiding that the system itself operates differently from mission time.

The functionalities offered by the proposed platform to design and verification engineers and to software developers are the following:

- Synchronized pattern application on the DUT input ports and results readback on output ports
- Embedded memory read/write operations
- Observation of critical internal DUT signals
- Breakpoint setting applying complex logical conditions based on signal observation
- Interactive step-by-step execution.

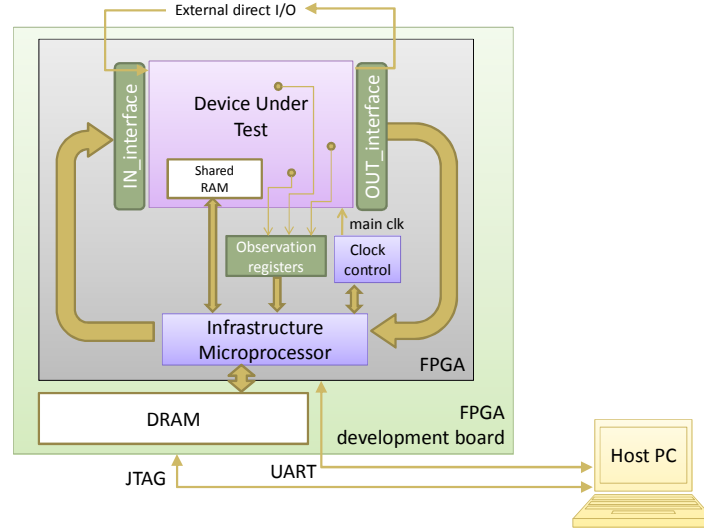


Figure 1. The co-verification and debug platform presented in [5].

Such operations are managed by an *infrastructure microprocessor* (IM), which is also mapped on the FPGA and provides a powerful and flexible interface between the DUT and the host computer (Fig. 1). The use of a microprocessor provides an easy way to develop customized verification and debug flows, requiring only the writing of high-level code. A library of general, automatically generated supporting functions to be run by the IM is used to further simplify procedure development: the basic functionalities deal with feeding patterns and reading output ports or other signals, applying a series of clock cycles, and reading and writing any DUT memory core. More complex functions can be developed to apply specific protocols on a subset of the DUT I/O signals. As an example, if a subset of the DUT input ports is a PCI interface, a specific function implementing the PCI protocol can be developed and run on the IM to perform the required data transfers by activating the proper control signals.

Breakpoints, which are highly desirable when performing the debug of extended system emulation runs, can be programmed exploiting the complete flexibility of a high-level language, correlating different observed signal conditions through logical functions and software counters. DUT signal observation is granted by the *out\_interface* and the observation registers.

Input/output, clock management and breakpointing tasks have to work in parallel on the IM to correctly stimulate the DUT and apply the desired flows. The coordinating tasks are delegated to a suitably developed software scheduler which introduces a minimum time overhead on the application of the flow, by keeping track of the elapsed clock cycles and determining clock-by-clock the low-level operations that need to be performed. The infrastructure microprocessor controls the clocks feeding the DUT in order to apply the input stimuli and sample the output results at accurate relative time intervals in spite of the unsteady flow determined by the concurrent execution of the coordination procedures. Therefore, the DUT emulation is clock-accurate and orders of magnitude faster than simulation.

### III. THE PROPOSED METHODOLOGY

In order to perform the verification and early BER characterization of the implementation of a digital baseband communication system, we propose to implement in the previously described FPGA system, as DUT, both the transmitter and receiver modules of the considered system (Fig. 2). Such modules are interfaced through a digital *channel model*, which operates the required format conversions and emulates the behavior of other system components (possibly analog circuitry) so as to enable data transmission. The digital channel model is equipped with a *noise injection system*, able to add to noise samples the transmitted signal. The noise samples are previously stored in one of the available platform memories (*noise sample memory*): no specific noise generator module was employed to provide additional flexibility in the characterization experiments.

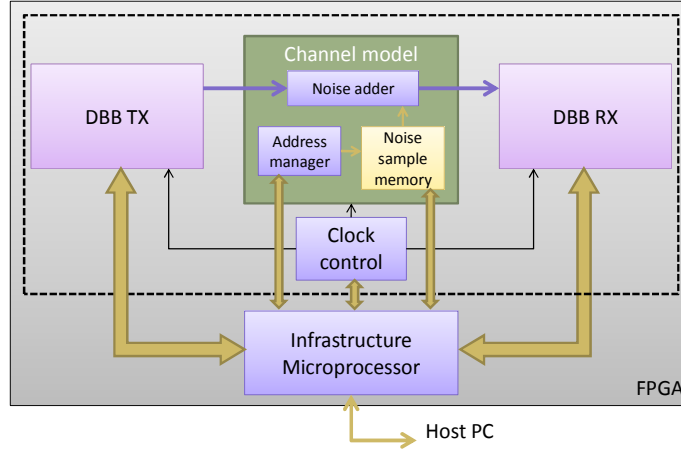


Figure 2. The proposed FPGA-based BER/PER characterization platform.

A *clock control module* is also implemented on the platform, which produces the required timing signals for the emulated systems, enabling the operation of multiple clock domains and the application of clock parameter perturbations.

To configure and interact with the selected DUT, the previously presented technique based on an Infrastructure Microprocessor and memory-mapped registers is employed. The IM provides the necessary input patterns to activate the emulated components, performs data readout and computes the required statistics, and manages the BER characterization operations by programming clock generators and the channel model.

Among the possible useful measurements that can be performed, we can cite the following:

- Verification of transmission correctness under different system configurations and with different data to be transmitted (hardware/software system validation and debug);
- Bit/symbol error rate performance verification in presence of additional channel noise (noise samples are generated off-line and thus any noise model can be employed);
- Bit/symbol error rate performance verification in presence of noise and frequency offsets or drifts between the transmitter and the receiver.

#### A. Noise injection system

The noise injection system is based on an arithmetic adder and on a memory storing previously elaborated digital noise samples. Overflow conditions are handled by suitably forcing the channel model output to maximum and minimum values, as needed. The platform allows the computation of signal and noise statistics (e.g., average value and power) before and after data clipping at the receiver input.

The noise sample memory is implemented as a dual-port RAM core, which can be accessed by the IM for writing and by an address generator module for reading. The noise sample parallelism and the sample memory depth can be customized in order to avoid artificial degradation of the noise statistics due to clipping or to noise sample correlation. As a matter of fact, for a reliable low BER measurement, it is important to accurately reproduce uncorrelated noise samples and special care must be taken to the ones at the tails of the probability density function (PDF) [3].

Noise injection activation can be selectively activated so as to enable characterization of specific functions in the system. For instance, if each transmission is composed of an initial synchronization sequence followed by the actual packet transmission, the noise injection can be selectively activated in time to determine the effect on the communication.

### B. Clock control module

The proposed FPGA-based platform enables the implementation of complex systems owning multiple clock domains, and the emulation of mutual clock parameter perturbation effects. This is made possible through the use of a dedicated clock control module and an accurate IM data transfer operation scheduling. The resulting emulation is performed at a different frequency with respect to the nominal DUT one, but the event sequence is preserved. It must be noted, however, that emulation at the nominal frequency would not be useful for validating a design since the FPGA and the target technology propagation and transition delays are different.

Let us first concentrate on the emulation of a multiple clock domain system in ideal conditions. For each clock, the clock control module includes the following structures:

- a programmable *frequency counter*, which is preliminary programmed by the IM so as to define each clock period (the actual value to be programmed needs to take into account the relative frequencies of the different clocks, e.g., if two clocks have the nominal frequencies of 2 MHz and 3.33 MHz, the relative counter values may be 5 and 3)
- a programmable *request counter* that is used to activate the application of a specific number of clock cycles on that clock (the other clocks will be coherently activated to maintain constant frequency relationships).

The IM performs read and write operations on the DUT interface registers (to apply input patterns or read sampled outputs) while keeping the clocks off. Then, it programs the required clock application on the clock control module, and waits until they have been applied. Then, the operation is repeated. Fig. 3.a graphically shows the adopted mechanism, where *IM\_clk* is the main free-running platform synchronization signal feeding the IM; *in\_data* is any of the signals controlled by the IM, which changes when the DUT clocks are off; *clk\_1* and *clk\_2* are two DUT clocks running at different frequencies when requested.

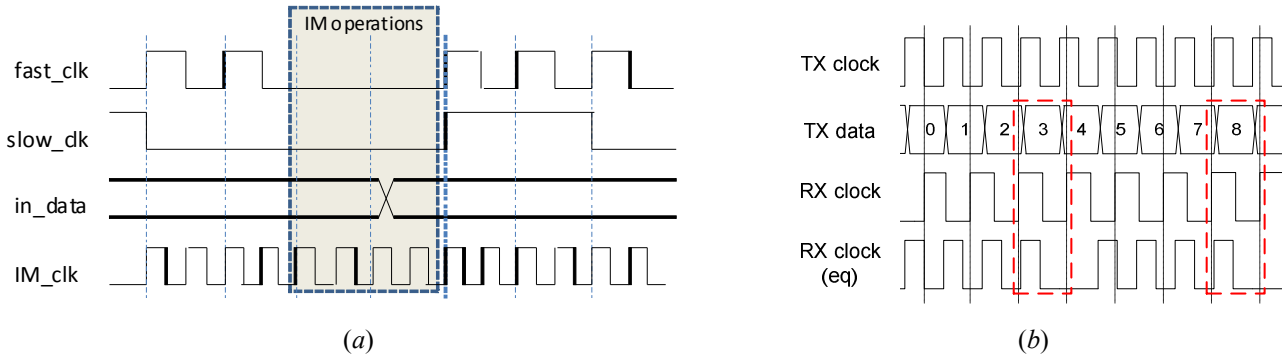


Figure 3. Clocking mechanism applied by the proposed platform (a) and frequency offset effect on multiple clock domains (b).

In order to perform BER testing, the clock control module allows emulating frequency drift effects between different clock domains. To do so, the first available option would be to feed the two clock domains with different (possibly external) clock sources. However, this strategy implies a relevant cost, since at least one of the clock generator units needs to be finely controlled, and the integrity of the clocking signal feeding the emulated circuit has to be preserved. An alternative solution may consist in the quantization of the clock period. A high-frequency clock is fed to two programmable counters: each time the programmable counters get to 0, the relative clock signal is switched. In this case, the emulation speed is decreased proportionally as the parameter setting sensitivity is increased.

In either case, the obtained effect is that data at the interface between the two clock domains is sampled with a different phase. This implies that setup/hold time constraints may not be satisfied and that, each time the phase difference between the two clocks exceeds  $2\pi$ , a data sample crossing the clock domain boundaries is lost. The first effect cannot be reproduced in FPGA since the technologic parameters are inherently different from the target technology ones, and may be taken into account during the injection of additive noise. The second effect can be obtained by periodically disabling one of the clocks for one cycle. Let us refer to the clock of a transmitter (TX) and a receiver (RX), as in Fig. 3.b; the receiver samples the transmitted data (TX data) at each rising RX clock edge. We suppose that the receiver has a lower frequency with respect to the transmitter (this phenomenon has been purposely exaggerated in figure). The equivalent RX clock (last row) has the same frequency as the TX clock, but a clock cycle is periodically removed to enable the emulation of the frequency offset effect. This can be easily implemented with a programmable counter periodically disabling the RX clock generator (or TX in case a higher RX clock frequency has to be emulated).

## IV. CASE STUDY

The proposed co-verification and debug platform has been employed in the development phases of a new integrated sensor system by Pirelli Tyres S.p.A. The system is intended to be placed within the tire and communicates with the on-board car computing system. An architectural description of the system can be found in [6]. Its general behavior is the following:

tri-axial accelerometric data are sampled, filtered and conditioned by means of suitable analog and digital circuitry, while a bidirectional radio channel is used to transmit pre-processed data to and receive status information and commands from a remote host device (within the car body). A microprocessor subsystem is in charge of managing the flows of data and synchronizing the operations so as to optimize data computation and power performance.

The system addressed in this work is the UWB radio uplink channel, which is used to transmit sensor data from the tire to the car. Due to the extremely low-power requirements of the transmitter and to the harsh conditions of the channel environment, the case study system is characterized by high receiver complexity. The packet structure and processing are based on, but not necessarily compliant with, IEEE 802.15.4a standard [6].

The BER verification and characterization platform has been implemented on Xilinx Virtex 5-LX platform, and the Xilinx MicroBlaze soft core has been selected as infrastructure microprocessor. The aforementioned DBB transmitter and receiver modules have been implemented as DUT on FPGA (Fig. 1 and 2). The two modules are accessed through an Extended Special Function Register (ESFR) interface and a Serial Peripheral Interface (SPI), respectively, which are implemented by means of support IP modules connected to the IM memory-mapped registers. The modules have been interfaced through a channel model including noise injection features, and the programmable clock control unit has been connected to the TX and RX clocks. Such unit allows emulating frequency offsets between the TX and RX clocks as detailed in Section III.

Fig. 4.a presents the architecture of the channel model with noise injector, while figure 4.b details the specially tailored architecture used to store the required 9-bit noise samples in a 32/64-bit per word memory so as to keep the circuitry simple and waste the lowest possible memory area.

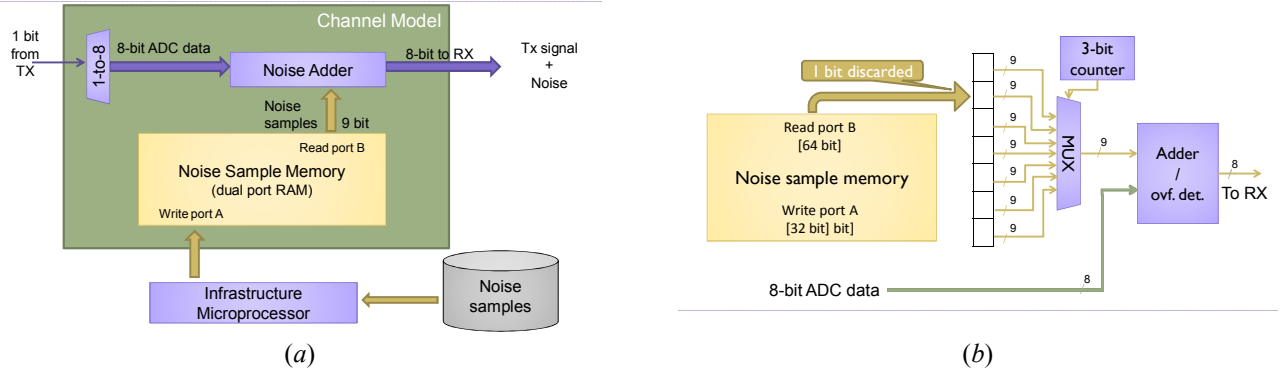


Figure 4. The noise injector module (a) and a detail for the current implementation (b).

Table I reports the statistical properties of a set of white Gaussian noise samples to be applied in the characterization experiment, where the values of the Q-function (the tail probability of the standard normal distribution) are considered as the figure-of-merit [8]. These results were calculated considering the generation of noise with a SNR of 0dB with respect to a TX signal with amplitude equal to half the available dynamic range, and correspond to the noise signal after re-normalization for zero mean and unit variance. For the 1-million and 57344-sample cases, the actual values of mean and normalized variance are (0.038125, 1.000436) and (0.282471, 1.001281), respectively. These results clearly show that the noise properties are not influenced by the relatively reduced set of samples employed.

TABLE I.  $Q(x)$  RELATIVE ERROR OF THE NOISE SAMPLES

x	Theoretical Q(x)	Relative Q(x) error of this work (%)	
		1 million samples	57344 samples
0.0	0.500	-0.5692	0.2965
0.2	0.4207	0.5019	0.2200
0.4	0.3446	0.1813	0.2809
0.6	0.2743	-0.1200	-0.1766
0.8	0.2119	-0.4831	-0.7214
1.0	0.1587	-0.9866	0.2316
1.2	0.1151	1.0249	-0.7510
1.4	0.0808	0.4400	-1.1857
1.6	0.0548	-0.5498	0.0506

Table II summarizes the FPGA resource occupation figures for the developed platform.

The platform has been employed to verify and characterize the DBB modules in a wide range of configurations, encompassing different encoding algorithms, error correction policies and packet sizes. Fig. 5 shows the result of a BER/PER test experiment obtained by applying different noise sample sets, for a specific configuration of the TX and RX DBB modules (SNR is expressed in an arbitrary scale for confidentiality reasons). For each SNR value, the platform is programmed to transmit random payload packets and stop only after a minimum of 20 lost packets have been detected or at least 5,000 packets have been transmitted. The comprehensive experiment has required the transmission of about 200,000 packets, and has taken about 20 hours to complete. For the sake of comparison, the RTL simulation of a single packet transmission on an Intel E6750 with 2 GB RAM takes about 20 seconds, resulting in more than 1,000 hours of simulation for 200,000 packets, i.e., almost 6 weeks of simulation. Therefore, the obtained time saving can be estimated to about 97%.

Fig. 6 finally reports the results of a second experiment where, for a selected configuration, the effect on PER of relative frequency offsets between the transmitter and the receiver reference clocks has been evaluated. In this case, the system emulates a slower receiver clock reference frequency with respect to the nominal value. As the graph clearly shows, the addition of this non-ideal condition affects the receiver noise immunity.

The analysis of the obtained figures and the comparison to specifications have confirmed the correct realization of the DBB modules.

TABLE II. FPGA RESOURCE OCCUPATION

Module	LUT [#]	Registers [#]
DBB TX module	1,565	1,199
DBB RX module	2,780	2,191
Channel model, clock control and support IPs	5,930	3,254
Microblaze and memory controllers	3,781	4,412
<i>Available FPGA resources</i>	<i>28,800</i>	<i>28,800</i>

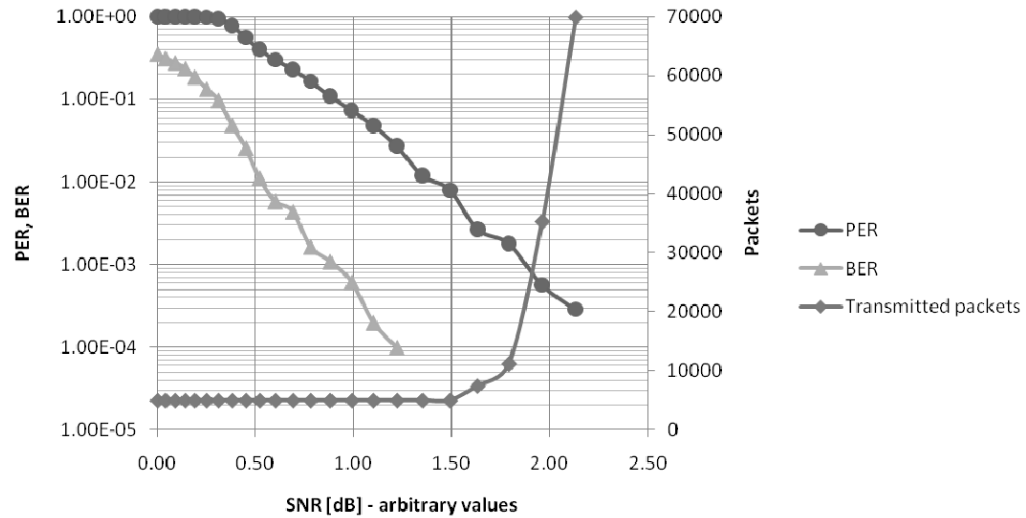


Figure 5. BER and PER traces with respect to SNR (expressed in arbitrary dB values).



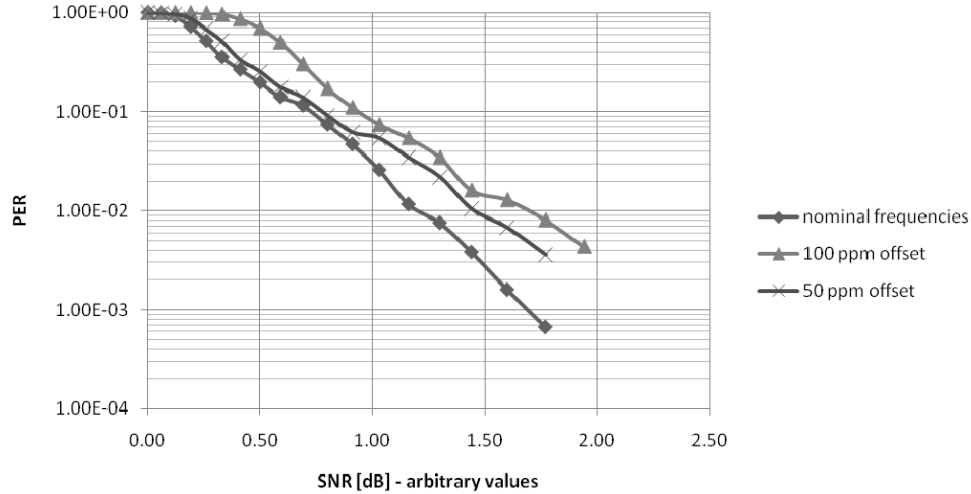


Figure 6. PER traces with respect to SNR (expressed in arbitrary dB values) with the application of a relative clock frequency offset between the transmitter and receiver systems.

## V. CONCLUSIONS

An FPGA-based platform for the verification and characterization of digital baseband communication systems has been proposed, based on a previously developed microprocessor-based architecture. The main innovations consist in the design and realization of a parametrizable noise injection mechanism and of a clock control module able to feed different clock domains and emulate frequency offsets between the TX and RX circuitries. Experimental results on an industrial case study show the effectiveness of the approach.

## REFERENCES

- [1] E.A. Newcombe, S. Pasupathy, "Error rate monitoring for digital communications", *Proceedings of the IEEE*, vol. 70, n. 8, Aug. 1982, pp. 806-825
- [2] T.A. Lindsay, "Innovations in BER testers enable fast and accurate eye diagram, eye mask, Q-factor, and jitter measurements", *IEEE Optical Fiber Communication Conference*, 2004
- [3] A. Alimohammad, S.F. Fard, B.F. Cockburn, "A flexible layered architecture for accurate digital baseband algorithm development and verification", *IEEE/ACM Design Automation & Test in Europe Conference & Exhibition*, 2009, pp. 45-50
- [4] M. Di Marzio, M. Grosso, M. Sonza Reorda, L. Sterpone, G. Audisio, M. Sabatini, "A novel scalable and reconfigurable emulation platform for embedded systems verification", *IEEE International Symposium on Circuits and Systems*, 2010, pp. 865-868
- [5] J. Lagos-Benites, M. Grosso, L. Sterpone, M. Sonza Reorda, G. Audisio, M. Pipponzi, M. Sabatini, "A low-cost emulation system for fast co-verification and debug", *IEEE European Test Symposium*, 2011 (in press)
- [6] S.C. Ergen, A. Sangiovanni-Vincentelli, X. Sun, R. Tebano, S. Alalusi, G. Audisio, M. Sabatini, "The tire as an intelligent sensor", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, n. 7, July 2009, pp. 941-955
- [7] J.G. Proakis, M. Salehi, "Digital Communications", Fifth Edition, McGraw-Hill, New York, 2008
- [8] Y. Fan, Z. Zilic, "BER Testing of Communication Interfaces", *IEEE Transactions on Instrumentations and Measurement*, vol. 57, n. 5, May 2008, pp. 897-906
- [9] C.-C. Fu, T.-P. Wang, K.C. Chang, C.H. Liao, T.-D. Chiueh, "A real-time digital baseband channel emulation system for OFDM communications", *IEEE Asia Pacific Conference on Circuits and Systems*, 2006, pp. 984-987