Blind Oversampling Data Recovery with Low Hardware Complexity

Michal KUBÍČEK, Zdeněk KOLKA

Dept. of Radio Electronics, Brno University of Technology, Purkyňova 118, 612 00 Brno, Czech Republic

kubicek@feec.vutbr.cz, kolka@feec.vutbr.cz

Abstract. *The paper is focused on the optimization and implementation of fully digital feed-forward blind oversampling CDR (BO-CDR). Two new phase-decision algorithms are proposed. Their complexity is very low, enabling a very simple and fast implementation even in FPGA, which was used as a development platform as well as a target device for the BO-CDR block. The FPGA-based optimization gave the opportunity to perform on-the-fly optimization under real conditions of target link. This greatly shortened the development time as there were no errors caused by inaccurate simulation models.*

Measurement results obtained on real links are included showing the jitter tolerance of the proposed algorithms to be comparable to the performance of modern PLL-based CDRs.

Keywords

Blind Oversampling CDR, FPGA, jitter.

1. Introduction

Serial asynchronous data transmission is used in the majority of today's interconnect and data transmission systems for its simple use and low hardware requirements. The list of standards where asynchronous transmission is used includes USB, SATA, PCI Express, HDMI, Ethernet and many other standards ranging from few Mb/s to 100 Gb/s.

The asynchronous transmission system requires a clock and data recovery block (CDR) to regenerate the clock signal used to sample the received data signal. The CDR has a significant impact on overall link performance. There is a variety of solutions based on different approaches (summarized in [1]) but most of today's CDRs are based on phase-locked loop (PLL). However, in many applications the PLL-based CDR can be a suboptimal solution not only because of its higher power consumption but also because of its analog nature, relatively long lock time, and higher overall cost of the final implementation.

The blind oversampling CDR is an all-digital architecture where several samples per bit of the input NRZ signal are taken and the decision logic uses the sample closest to the eye center to recover data [1], [2], [3]. Many implementations of BO-CDRs were already published, showing a reasonable performance in modern receivers [5], [6]. Several FPGA-based implementations have been mentioned in the literature but without a discussion of jitter sensitivity [10] or with an incorrect interpretation of the results measured [7]. However, until now no direct comparison with other CDRs has been made. Thus many designers are still not complying with its simple concept and presume its low performance compared to traditional PLL-based solutions. This misconception is to be overcome in this article by direct comparison.

The aim of this paper is to introduce two new simple decision algorithms suitable for both FPGA and ASIC implementation. FPGA is a suitable platform for CDR optimization and testing in a real system as it enables in-system reconfiguration. The FPGA can also be used as the target device for the whole receiver implementation, replacing application-specific standard parts. Both algorithms show a low jitter sensitivity and low hardware requirements. A direct comparison with a PLL-based solution is given to prove the good performance of the blind oversampling architecture.

Section 2 of the paper describes the basic principle of BO-CDR, Section 3 deals with the new algorithms, and Section 4 introduces the experimental results.

2. Blind Oversampling CDR

To be precise, blind oversampling performs only data recovery as no clock is actually recovered. In the literature, however, it is often classified as a CDR because its usage and system level properties are very similar to traditional clock and data recovery circuits.

The basic block diagram of the BO-CDR is shown in Fig. 1. The equalized and conditioned received NRZ data signal is asynchronously sampled by a local oversampling clock. The bit period T_{BIT} of the data signal must be approximately an integer multiple of the period T_{OS} of the oversampling clock in order to take a fixed number M of samples per every bit. This multiple, called the oversampling ratio, is the main parameter of the BO-CDR. It is

often unsuitable or even impossible to use an oversampling clock that has a frequency M-times higher than the link data rate. Instead, multiple mutually phase-shifted clock signals with a lower frequency are used, Fig. 1. The phase shift of adjacent clock signals must correspond to the required sampling period T_{BIT}/M. The sampling effectively divides the bit period into M equally spaced sampling domains, Fig. 2.

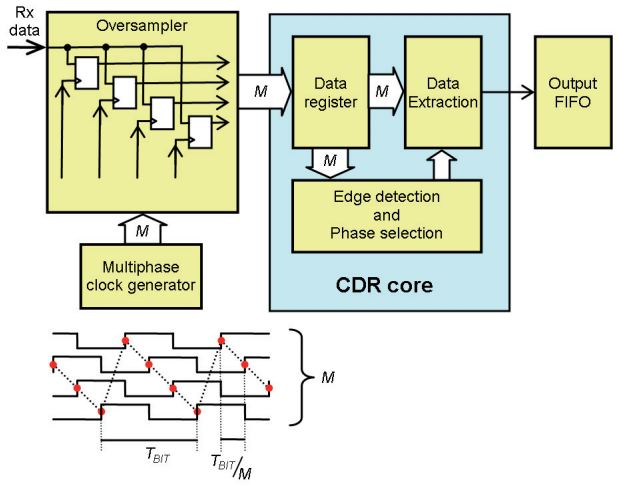

Fig.1. Blind oversampling CDR architecture with multi-phase clock oversampler (drawn for $M = 4$).

The phase relationship of received data signal and local clock signal is not known in advance and varies with time. The CDR core logic must continuously determine this phase relationship and use a phase-adjusted local clock signal to sample the received data as close to the center of the data eye as possible, Fig. 2.

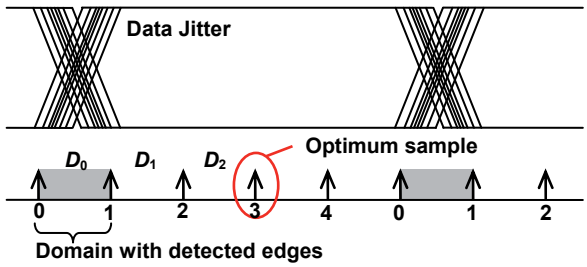

Fig.2. Choosing the optimum sampling phase ($M = 5$).

There are two basic concepts of sample processing: *Majority Voting* and *Phase Picking*. The majority voting scheme assumes that the majority of samples should have the current value of the bit being sampled. As it has a worse performance compared with the phase picking method, it will not be considered in further discussion [8]. In the case of phase picking, the optimum sampling phase is selected in the middle of the domains where edges were detected, i.e. approximately in the middle of the data eye. This approach requires a more complex decision logic compared to the majority voting method but is less susceptible to jitter bound to the received data signal.

Reducing the complexity of the BO-CDR, while keeping a high jitter tolerance, is probably the greatest challenge as it is the limiting factor for its efficiency and maximum achievable data rate. In the discussed case of phase picking BO-CDR, the decision algorithm for choosing a proper sample takes the major part of digital circuitry, [3]. In [3] only a theoretical discussion of choosing a particular algorithm is given but no further research towards practical implementation was made.

There are two commonly used decision algorithms for phase picking: *Direct Phase Picking* (DPP) and *Averaged Phase Picking* (APP) [3]. DPP makes a new phase decision immediately after edge detection. It is the simplest possible implementation of the algorithm, but it shows poor performance on signals containing jitter. The APP algorithm makes a decision upon receiving a fixed-length interval of W bits and this decision is applied to the delayed oversampled data signal it was calculated from. The occurrence of edges within sampling domains is counted and the sampling domain containing the majority of edges is considered to be the farthest one from the center of the data eye. Although this scheme is based on simple mathematical operations, it requires the implementation of a relatively complex digital circuitry. It results in increased power consumption and chip area, and it may also negatively affect the maximum achievable data rate [4]. On the other hand, this algorithm shows good BER performance even for signals containing a significant amount of jitter.

By increasing the oversampling ratio it is possible to improve the phase resolution of the sampler, i.e. to decrease the sampling phase error. On the other hand, a higher oversampling ratio rapidly increases the complexity of the receiver, mainly the domain selection block. Obviously, there is always a tradeoff between performance in terms of jitter tolerance on one hand and power consumption, data rate and hardware requirements on the other [3].

An ideal decision algorithm should have the BER performance of the APP algorithm while its complexity should not be much higher than that of the DPP algorithm.

3. New Decision Algorithms

Two low-complexity algorithms having good BER performance are proposed (referred to as *S2par* and *Ccnt*). Both algorithms have a parameter W that can be used to adjust CDR properties.

The *S2par* decision algorithm works as follows: there is a floating interval of W **consecutive bits** where the edge position is evaluated. Whenever all edges detected within this interval belong to the same sampling domain, a new decision on the optimum sampling phase is made.

Fig. 2 illustrates the ideal selection process for $M = 5$ and Fig. 3 shows the hardware implementation of the proposed CDR block. The edge detector output in one-of-M coding is led to M decision blocks. Each block tests the exclusive occurrence of edges in one (i-th) domain. After

receiving W bits the input "1" is clocked-out of the shift register if and only if all edges occurred in the i-th domain. If an edge occurred in a different domain, the shift register is reset. The occurrence of "1" at the output of the i-th block signalizes meeting the condition for selecting a new sampling phase. In the case of odd M the optimum sampling phase is

$$p_{opt} = \left(i + \frac{M+1}{2} \right) \bmod M \,. \tag{1}$$

The *Ccnt* algorithm works slightly differently even though its implementation is very similar. In this case, a new decision is made whenever W **consecutive edges** on the received data signal belong to the same sampling domain.

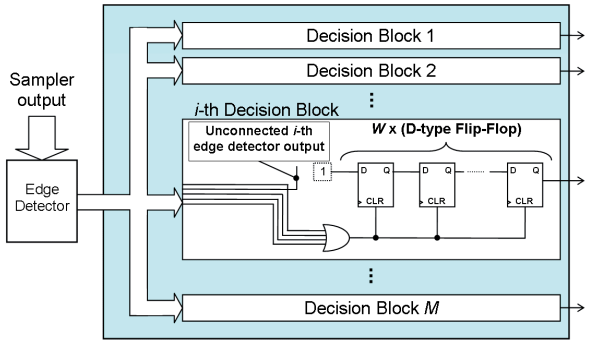

Fig. 3. Implementation of *S2par* algorithm.

The implementation in Fig. 4 is similar to the *S2par* algorithm. In this case the shift register clock is enabled only after receiving an edge in the i-th domain.

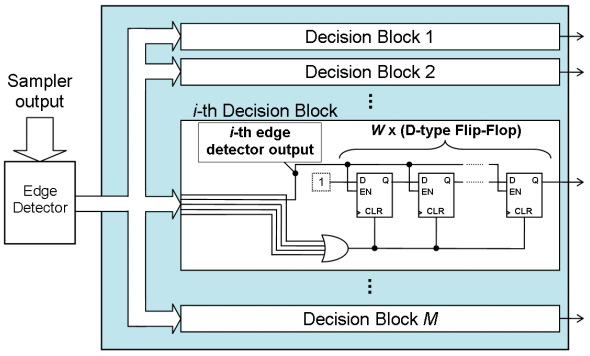

Fig. 4. Implementation of *Ccnt* algorithm.

Both algorithms are based on shift registers, which can operate in very high frequencies. The only combinatorial block in the domain selection block is an (M-1)-input OR gate. It has a negligible propagation delay compared to adders and multipliers used in other published solutions, which results in a high maximum operating frequency (i.e. maximum achievable data rate).

The clock signal used to trigger the flip-flops of all decision blocks should be one of clock signals used for asynchronous data sampling (usually the one with zero phase offset). Its nominal frequency is equal to the data rate.

4. Experimental Results

To evaluate the performance of different BO-CDR algorithms, a test platform in Fig. 5 was used. The input signal was degraded, using an attenuator in the optical domain, which effectively adds high-frequency jitter to the NRZ data signal. Transmitting and receiving data rates were derived from different crystal oscillators.

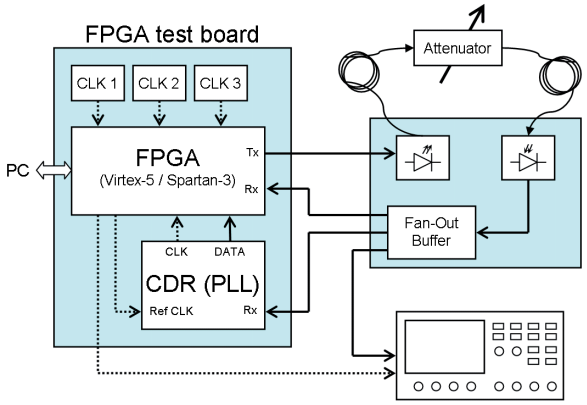

Fig.5. Simplified block diagram of FPGA-based CDR test platform.

All CDRs with tested algorithms were implemented in one FPGA sharing a single common data sampler. Thus it was possible to compare all algorithms under exactly the same conditions. Comparison was made for several signal quality levels (different amounts of jitter contained in the signal) and for different patterns (standard PRBS sequences 2^{23}-1 and 2^{7}-1). Results of the comparison for the *S2par* and *Ccnt* algorithms are shown in Fig. 6 and Fig. 7, respectively. As derived in [9], the higher the value of W the longer the interval between decisions, i.e. for a high W the CDR is not able to follow the data signal phase due to unavoidable frequency offset between the transmitter and the receiver clocks.

Red points in the characteristics indicate a bit extraction error which occurs when an improper number of bits is extracted from the data (one bit is omitted or an extra bit is inserted). This is a fatal error, which totally corrupts the link until a new frame synchronization procedure is performed.

To be able to directly compare the performance of oversampling CDR to traditional PLL-based CDRs, the test platform in Fig. 5 was also equipped with a Micrel SY87700 CDR circuit, and one Virtex-5 GTP transceiver was used. The received data signal is fed to both the PLL-based CDR and to the FPGA, where BO-CDR is implemented. This concept ensures exactly the same test conditions for CDRs under comparison over the whole experiment. The results are shown in Fig. 8.

Probably the most obvious benefit of using the oversampling CDR architecture is its exceptionally low power consumption. The BO-CDR block utilizing the *Ccnt* algorithm (N=5) and implemented in a Spartan-3 FPGA requires only 51 mW at 125 Mb/s. The major part of the

power is required by the reference clock generator, which is a DCM block in this case (about 46 mW), Fig. 9. It should be noted that a digital circuit implemented in an FPGA requires much more power than the same circuitry migrated into the ASIC technology. Thus further reduction of power consumption can be expected when implementing a BO-CDR block as an ASSP or ASIC.

Comparable PLL-based CDRs for a desired data rate show a much higher power consumption. A "low power" PLL-based CDR AD807 needs 170 mW [12] to operate at 125 Mb/s, the GTP transceiver (Virtex-5 FPGA) requires 222 mW and the SY87700 takes about 560 mW [11], which is about 10 times more than in the case of BO-CDR.

The benefit of using BO-CDR is even greater when multiple receivers are to be implemented in a single device. In such a case all BO-CDRs operating at the same data rate may share a common oversampling clock generator. The power consumption of 8 BO-CDR blocks is then only about 83 mW, which is less than twice of what is needed by a single BO-CDR block! This contrasts with the same hypothetical implementation using 8 PLL-based CDRs, where every block needs its own recovered clock source (PLL). Even the lowest-power CDR implementation requires about 16 times more power than the BO-CDR solution, Fig. 10. For BO-CDR and GTP the power consumption was calculated using the Xilinx power estimator tool, while typical datasheet values were taken into account in the case of AD807 and SY87700.

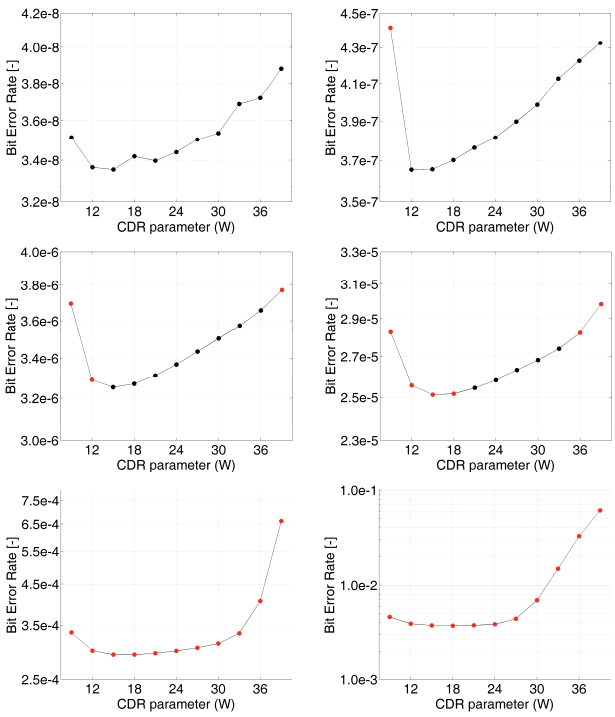

Fig. 6. Impact of parameter W on BER for different amounts of jitter contained in received signal (algorithm S2par; RMS jitter: first row 0.048 UI, 0.052 UI; second row 0.057 UI, 0.064 UI; third row 0.080 UI; 0.124 UI, test pattern PRBS 2^{23}-1).

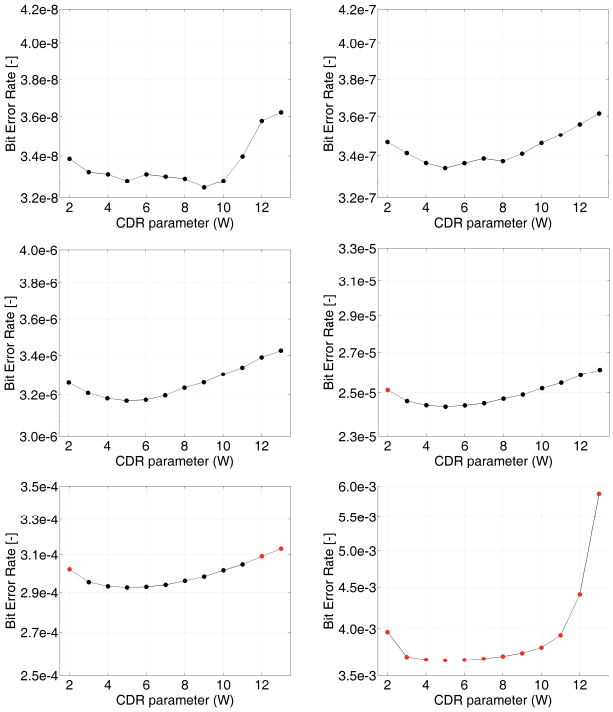

Fig. 7. Impact of parameter W on BER for different amounts of jitter contained in received signal (algorithm Ccnt, jitter is the same as in Fig. 6).

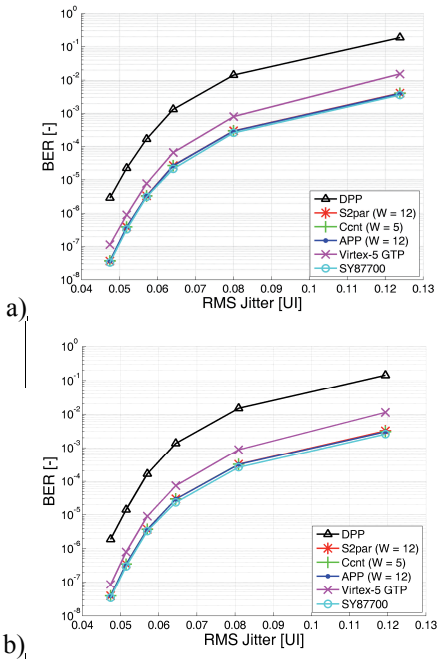

Fig. 8. BER performance of selected BO-CDRs and PLL-based CDRs; reference clock deviation for PRBS 2^{23} - 1, a) 30 ppm; b) 500 ppm.

It is worth mentioning that the implementation of the reference clock source for BO-CDR is significantly less complex than the implementation of the PLL of a traditional CDR. This is mainly due to the simple fact that the BO-CDR clock source needs not be synchronized to the jitter-degraded received data signal and as such can be simplified.

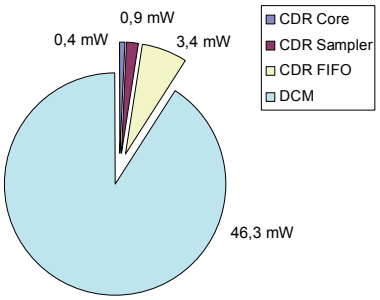

Fig. 9. Power consumption of a BO-CDR block implemented in a Spartan-3 FPGA; 51 mW total power (f_{BIT} = 125 MHz, M=5, N=5, f_{REC} = 156,25 MHz); results were obtained using the Xilinx Power Estimator tool.

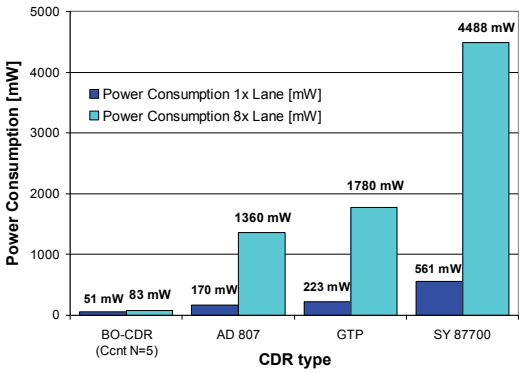

Fig. 10. Power consumption of CDR circuits for single receiver (1x Lane) and 8 receivers (8x Lane) configuration.

5. Conclusion

Two simple data extraction algorithms were proposed and successfully tested. They allow a very simple, all-digital hardware implementation. At the same time, properties comparable to PLL-based CDRs are achieved. Both algorithms can be adjusted using a parameter that can balance the performance, latency and complexity of the design.

Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 230126. The research was also financially supported by the Czech Science Foundation under projects No. 102/08/0851 and No. 102/09/0550, and by the Ministry of Education of the Czech Republic under research projects No. MSM0021630513.

References

[1] HSIEH, M., SOBELMAN, G. E. Architectures for multi-gigabit wire-linked clock and data recovery. *IEEE Circuits and Systems Magazine*, 2008, vol. 8, no. 4, p. 45-57.

[2] NEDOVIC, N. et al. A 40–44 Gb/s 3× oversampling CMOS CDR/1:16 DEMUX. *IEEE Journal of Solid-State Circuits,* 2007, vol. 42, no. 12, p.2726-2735.

[3] KIM, J., JEONG, D. K. Multi-gigabit-rate clock and data recovery based on blind oversampling. *IEEE Communication Magazine*, 2003, vol. 41, no. 12. p. 68–74.

[4] VAN IERSSEL, M., SHEIKHOLESLAMI, A., TAMURA, H., WALKER, W. W. A 3.2 Gb/s CDR using semi-blind oversampling to achieve high jitter tolerance. *IEEE Journal of Solid-State Circuits*, Volume 42, Issue 10, 2007. P. 2224-2234.

[5] YANG, C.-K., FARJAD-RAD, R., HORWITZ, M. A 0.5-µm CMOS 4.0-Gbit/s serial link transceiver with data recovery using oversampling. *IEEE Journal of Solid-State Circuits*, 1998. vol. 33, no. 5. p. 713- 722.

[6] PARK, S.-H., CHOI, K.-H., SHIN, J.-B., SIM, J.-Y., PARK, H.-J. A single-data-bit blind oversampling data-recovery circuit with an add-drop FIFO for USB2.0 high-speed interface. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2008, vol. 55, no. 2, p.156-160.

[7] BUSHNAG, S., NAKURA, T., IKEDA M., ASADA, K. All digital baseband 50 Mbps data recovery using 5× oversampling with 0.9 data unit interval clock jitter tolerance. In *Proc. of 12th Int. Symp. on Design and Diagnostics of Electronic Circuits & Systems 2009*. p. 206-209.

[8] YANG, C-K. K. Design of high-speed serial links in CMOS. *Ph.D. Dissertation*, CSL-TR-98-775, 1998, Stanford Univ., California.

[9] KUBÍČEK, M., KOLKA, Z., BIOLEK, D., BIOLKOVÁ, V. Optimization of oversampling Data Recovery. In *52nd IEEE International Midwest Symposium on Circuits and Systems*, Mexico: IEEE, 2009, p. 467-470.

[10] Xilinx, Inc., 2100 Logic Drive San Jose, CA 95124-3400, USA. *Data Recovery (xapp224).* [online], 2000, cit. [2007-10-15]. Available at http://www.xilinx.com/support/documentation/application_notes/xapp224.pdf.

[11] Micrel Inc. 2180 Fortune Drive San Jose, CA 95131, USA. *SY87701V Clock and Data Recovery*. [online], 2003 [cit. 2007-09-10]. Available at http://www.micrel.com/_PDF/HBW/sy87700v.pdf.

[12] Analog Devices, Inc., 3 Technology Way Norwood, MA 02062, USA. *Fiber Optic Receiver with Quantizer and Clock Recovery and Data Retiming* [online], 2000 [cit. 2009-06-12]. Available at http://www.analog.com/static/imported-files/data_sheets/AD807.pdf>

About Authors ...

Michal KUBÍČEK was born in Svitavy, Czech Republic, in 1983. He received his M.Sc. from the Brno University of Technology in 2001. Since then, he has been pursuing his Ph.D. degree at the Department of Radio Electronics at the same university. He is concerned with FPGA-based digital systems. His work is focused on high speed data transmissions, data processing and embedded systems.

Zdeněk KOLKA was born in Brno, Czech Republic, in 1969. He received the M.Sc. degree in 1992 and the Ph.D. degree in 1997, both in Electrical Engineering, from the Brno University of Technology. In 1995 he joined the Department of Radio Electronics, Brno University of Technology. His research and teaching interests are in computer-aided design of electronic circuits, modeling, and numerical algorithms.