Optimization of Oversampling Data Recovery

Zdenek Kolka, Michal Kubicek, Dalibor Biolek, Viera Biolkova
Faculty of Electrical Engineering and Communications
Brno University of Technology
Brno, Czech Republic

***Abstract*—The paper deals with the design and optimization of blind oversampling Clock and Data Recovery (CDR) based on FPGA prototyping. The main advantage of the oversampling CDR is the fully digital architecture, which enables the FPGA-based testing and its subsequent integration into any ASIC technology. The oversampling CDR is a promising block for Free Space Optical (FSO) applications because of its extremely short reacquisition time, which is the key feature for efficient communication over the frequently fading channel.**

An efficient statistical simulation model for the CDR optimization is presented. Our effort in optimization was focused mainly on the simplification of the decision algorithm while maintaining acceptable jitter tolerance. The suggested method was verified on the Xilinx FPGA platform.

I. Introduction

The technology of Free-Space Optical links consists in transmitting information by means of light beams in space or in the atmosphere. At present, terrestrial systems with a range of up to 2 km using a simple On-Off Keying are commercially available. FSO is a very promising technology for Unmanned Aerial Vehicles and for High-Altitude Platforms [1].

Long-range atmospheric links are influenced by atmospheric turbulence, which cause power fluctuations at the receiver with total fades on the millisecond time scale for static terminals and on the microsecond time scale for aircraft. Communication terminals for FSO applications should implement data protection methods as frequent outages would badly influence the behavior of the TCP protocol, which is most likely to be used [2].

Atmospheric turbulence also influence the pulse arrival time, resulting in jitter seen by the receiver. The RMS of turbulence-induced jitter can be estimated to the order of picoseconds during clear weather [3]. The total atmospheric jitter at the receiver is the sum of direct atmospheric jitter and jitter induced through parasitic AM/PM conversion in the receiver. As the main part of turbulence energy is contained in an approximately 200Hz band for static terminals the atmosphere contributes to random low-frequency jitter components. The main source of high-frequency jitter will be the receiver noise, clock generators and the CDR circuitry.

This paper describes the implementation of an all-digital blind oversampling CDR block, which is a part of FSO terminal. In comparison with other types of CDR that employ analog blocks (oscillators, filters, etc.) the all-digital system is easy to port to another technology or can be entirely tested in FPGA. The oversampling CDR for high-speed links first appeared in [4], followed by many other papers. Statistical behavioral simulations are desirable as they are in several orders of magnitude faster in comparison with Monte-Carlo methods [5].

The paper is focused on the optimization and design of all-digital CDR for FSO applications with adaptive decision window. Section 2 deals with a statistical model of CDR. Sections 3 and 4 present a test implementation in Xilinx FPGA including the results measured.

II. Blind Oversampling CDR with Probabilistic Decision Window

Figure 1 shows the basic principle of blind oversampling CDR, where the input digital stream is usually sampled by an odd number of clock phases uniformly distributed over one period T (also called the *Unit Interval – UI*). The number of phases is called the *oversampling rate M*. The decision logic searches for transitions in the stream and recovers data using such a phase that is delayed by $T/2$ from transitions, i.e. the sampling point is being continuously adjusted to the approximate center of the eye diagram. The decision on the optimum sampling phase is made upon receiving W bits in order to average the random variations of edge positions. The clock source need not be synchronized with the transmitter clock.

The digital signal phase is quantized into M levels. The theoretical maximum difference between the receiver clock and the data phase within an interval of consecutive identical symbols, where no phase-update decision can be made, is

$$\frac{\lfloor M/2 \rfloor}{M} UI \,. \tag{1}$$

This work was supported by the Czech Science Foundation under projects No. 102/08/0851, No. 102/08/0784, No. 102/08/H027 and No. 102/09/0550, the Czech Ministry of Education under research projects No. MSM0021630513 and No. 2C06012 and by the European Community's Seventh Framework Programme under grant agreement no. 230126.

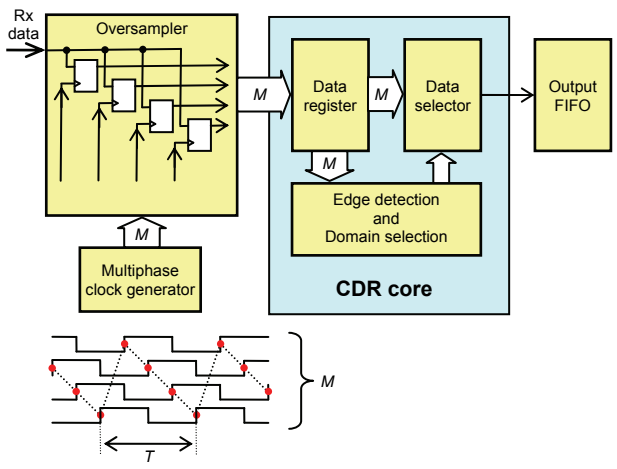

Figure 1. Blind oversampling CDR.

As the input signal passes through a number of limiters before being latched, the effect of noise on the bit error rate will be modeled on the horizontal eye opening. We will assume an ideal local clock and jitter generated by CDR will be included in the data signal. As the received optical power decreases, the ambiguity of zero crossing grows, mainly due to the noise of photodiode preamplifier.

Generally, the total jitter can be divided into random and deterministic components [6]. The random jitter (RJ) is modeled as unbounded having Gaussian distribution characterized by zero mean and standard deviation σ_{RJ}. The deterministic component (DJ) is represented by high-frequency inter-symbol interference and duty-cycle distortion with peek-to-peek amplitude d. Its probability density function (pdf) is approximated by a dual Dirac function. Convolving both pdfs we obtain the total pdf of zero-crossing [6]

$$PDF(\tau) = \frac{1}{2\sqrt{2\pi}\sigma_{RJ}}\left(\exp\left(-\frac{(\tau - d/2)^2}{2\sigma_{RJ}}\right) + \exp\left(-\frac{(\tau + d/2)^2}{2\sigma_{RJ}}\right)\right). \quad (2)$$

Although the parameters for the positive and the negative crossing can differ, they will be considered the same.

If the left crossing in Fig. 2 occurs after the sampling point T_s or the right crossing occurs before it, an erroneous symbol is sampled. For a pseudorandom data stream with a transition occurrence probability of approximately 0.5, the probability of bit error can be estimated as

$$P_{be}(T_s) \approx \frac{1}{2}\left(\int_{T_S}^{\infty} PDF(\tau)d\tau + \int_{-\infty}^{T-T_s} PDF(\tau)d\tau\right). \quad (3)$$

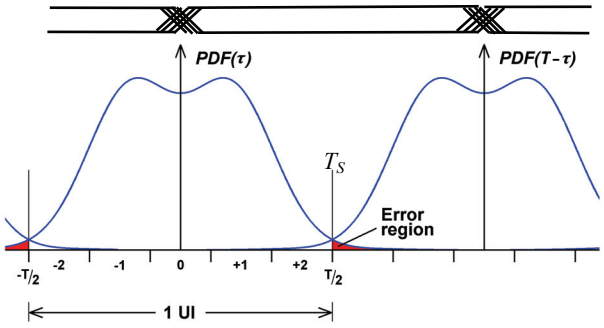

Figure 2. False symbol detection (drawn for $M = 5$).

The method for estimating the Bit Error Rate (BER) is based on a modified method [5] originally developed for sinusoidal jitter tolerance analysis.

Let us center the sampling domains on the ideal crossing position and let us denote them D_0, $D_{\pm1}$, $D_{\pm2}$, etc. Upon receiving W bits the decision logic selects the i-th sampling domain among other domains with the probability P_{Di}, i.e. $\sum P_{Di} = 1$. Then, the bit error rate can be estimated as

$$BER \approx \frac{1}{W}\sum_{i=-(M-1)/2}^{(M-1)/2} P_{Di}\left(\sum_{j=1}^{W} P_{be}\left(T\left(1/2 + i/M\right) + \Delta t(j)\right)\right). \quad (4)$$

The formula includes an empirical term Δt, which accounts for deterministic narrow-band jitter and wander, which are not negligible for large values of W. A simple approximation

$$\Delta t(j) = j(\Delta f / f)\,[UI] \quad (5)$$

takes into account the difference of oscillator frequencies of the transmitter and the receiver.

The proposed decision algorithm chooses the sampling clock domain when N consecutive edges are detected in the same domain. The conditional probability that an edge will be detected in the i-th domain (if the edge occurs in data stream) is

$$P_i = \int_{T/M(i-1/2)}^{T/M(i+1/2)} PDF(\tau)d\tau. \quad (6)$$

Assuming statistical independence, the probability that all N consecutive edges belong to the i-th domain is

$$P_{Ni} = (P_i)^N. \quad (7)$$

Then the conditional probability of selecting the i-th domain (if the decision is taken) is

$$P_{Di} = P_{Ni} / \sum P_{Ni} \quad . \tag{8}$$

As the process of taking decisions can be described by the Poission distribution, the average number of bits between decisions, i.e. the average length of decision window W_{avg}, is

$$W_{avg} \approx 2 / \sum P_{Ni} \tag{9}$$

for a transition probability of 0.5.

The average length of the decision window depends on the amount of jitter. For low jitter conditions, W converges to N (domain 0 is "always" selected). As jitter grows, the value of W increases, as shown in Fig. 3. There is an optimum value of the parameter N. A low value results in insufficient filtering while a large value decreases the tracking ability of the CDR as the decision window grows quickly.

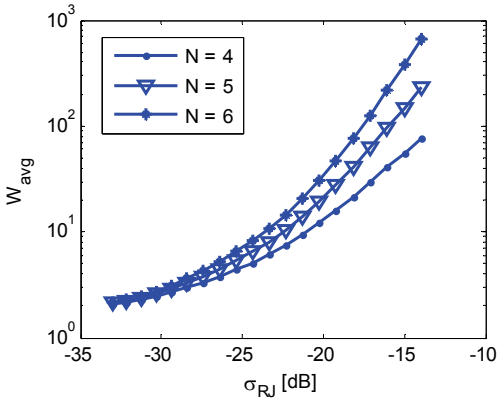

Figure 3. Average lenght of decision window (d = 0.1 UI, M = 5).

III. FPGA-BASED IMPLEMENTATION

To be able to implement the oversampling CDR in an FPGA a multiphase clock source is needed, as shown in Fig. 1. Considering oversampling rate M, it is necessary to generate M clock signals with a frequency equal to the data rate, shifted by 1/M of UI. Each clock signal is used to take one sample per data bit. It is relatively simple to generate such clock signals with different phases in an ASIC. In FPGA, however, this task is nontrivial for higher frequencies and puts limits on the final implementation in terms of maximum data rate and oversampling rate.

In modern FPGAs there are embedded hard-IP blocks for clock processing. These blocks are PLL- or DLL-based and are used to derive clock signals with various frequency and phase offset from a reference clock signal. Their ability to generate a suitable clock signal for the oversampler is limited. For example, a Digital Clock Manager (DCM) block in Xilinx Spartan-3 FPGA is able to generate suitable clock signals with a phase resolution not smaller than 1.5 ns. The same block in Virtex-5 FPGA is able to generate sampling signals with a phase resolution of 0.45 ns. These values can be used for the direct calculation of maximum achievable data rate at a certain oversampling rate as shown in Table 1.

TABLE I. MAXIMUM DATA RATE OF OVERSAMPLING CDR IN XILINX SPARTAN-3 AND VIRTEX-5 FPGAS.

	Spartan-3 Phase resolution 1.5 ns		Virtex-5 Phase resolution 0.45 ns	
Oversampling rate (M)	***Bit period [ns]***	***Data rate [Mb/s]***	***Bit period [ns]***	***Data rate [Mb/s]***
3	4.5	222	1.4	733
4	6	166	1.8	550
5	7.5	133	2.3	440
6	9	111	2.7	366
7	10.5	95	3.2	314

The decision algorithm should be as simple as possible to ensure high maximum data rate, low hardware requirements and low power consumption of the CDR. In common oversampling CDRs an algorithm that averages the edge position over a fixed length interval is usually used. This algorithm requires the implementation of some basic arithmetic blocks, like adders and multipliers, which require a lot of hardware resources. Using the complex logic may negatively affect the maximum data rate when a combinatorial path delay approaches 1 UI time (delay of logic gates and interconnection). Power consumption is also affected as it is proportional to the amount of hardware resources used. On the other hand, a CDR algorithm with too simple decision algorithm cannot process received signal containing large amount of jitter with sufficiently low BER and is suitable only for high quality links (like an onboard chip interconnection).

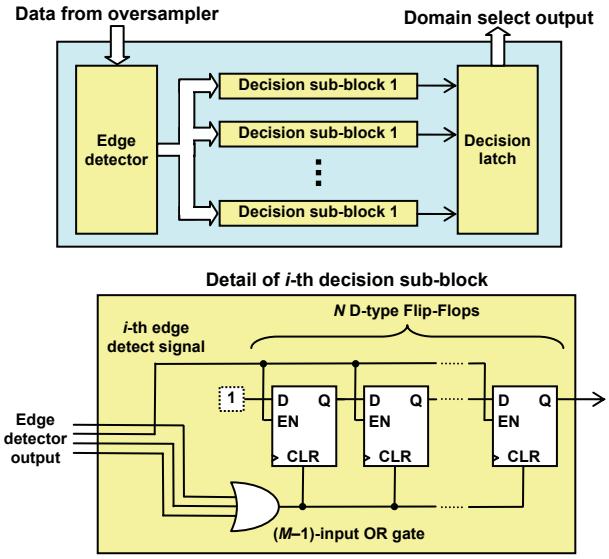

Figure 4. Detailed view of the edge detection and domain selection block.

The architecture of the CDR algorithm proposed in Section 2 can be seen in detail in Fig. 4. Each output of the edge detection block is used as the enable signal of one shift

register. When an edge is detected in domain *i*, logical 1 value is shifted into the *i*-th register and all the other registers are reset to zero. When the logical 1 value propagates to the end of the shift register, the actual sampling domain is updated. To implement the algorithm, *M* shift registers of length *N* are needed.

In contrast to other solutions incorporating arithmetic operations, in this case there are only a few logic gates between registers and so high clock frequencies are achievable. The most critical path is the reset logic of each shift register composed of (*M*-1)-input OR gate.

IV. Measured Results

The proposed method with oversampling rate $M = 5$ was implemented in the Spartan-3 FPGA to prove its performance. To be able to compare it with a standard PLL-based CDR, a dedicated CDR IC was used as a reference (Micrel SY-87700). One more oversampling CDR was chosen to be compared with the proposed solution. Xilinx provides an oversampling CDR reference design that uses a direct decision algorithm where the actual domain is selected directly according to actual edge position (without averaging) [7]. The resulting CDR has lower hardware requirements but it suffers from high jitter sensitivity. Unfortunately, in the CDR documentation there is no note about jitter sensitivity, which may lead to misunderstanding by designers who may try to use it in applications where received signal contains relatively large amount of jitter. To be able to compare the performance of the direct decision CDR and the proposed one, both were implemented into the FPGA simultaneously. Furthermore, the proposed CDR was tested with different values of the parameter *N*.

The test board setup is shown in Fig. 5. In the FPGA, both oversampling CDR blocks share a common oversampler. The resulting error rate was measured using bit error rate testers (BERT) implemented directly in the FPGA, one for each CDR.

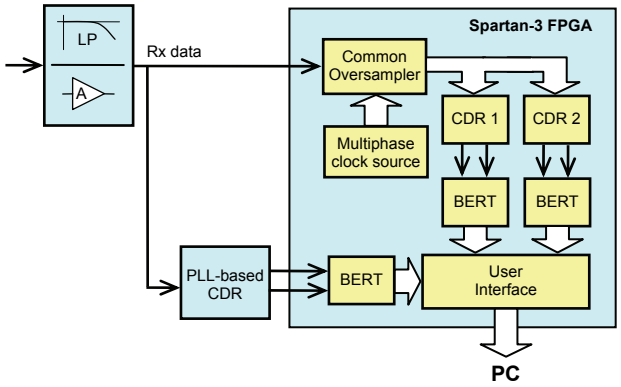

Figure 5. Setup of receiver for CDR comparison.

As can be seen in Fig. 6, the proposed oversampling CDR algorithm approaches the BER performance of a conventional PLL-based CDR. The direct-decision oversampling CDR shows very poor performance.

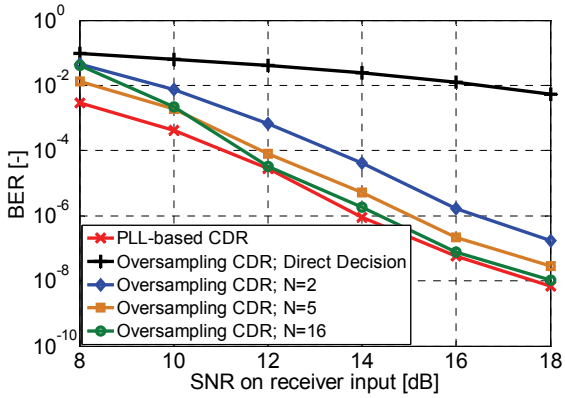

Figure 6. CDR mesurement results (125Mb/s).

The implementation of the oversampling CDR with parameter $N = 5$ and oversampling rate $M = 5$ in the Spartan-3 FPGA requires only 2 DCM blocks, 100 registers and 152 Look-up Tables (98 slices). Power consumption of the logic is only about 8 mW (including oversampler and deserializer). The power consumption of the multiphase clock generator (two DCM blocks) is about 84 mW. The FPGA quiescent power is about 24 mW. ASIC implementation of the proposed CDR can be expected to be even more power efficient. Power consumption of the PLL-based CDR is 560 mW at the same data rate.

V. Conclusion

Using the statistical modeling of CDR for optimization and FPGA-based platform for verification it is possible to design both highly efficient and fast oversampling CDR algorithm that is comparable with a PLL-based CDR.

References

[1] H. Henniger, B. Epple, and D. Giggenbach, "Mobile FSO Activities in Europe and Fading Mitigation Approaches," In Proc. 17th Int'l. Conf. Radioelektronika, Brno(Czech Republic), 2007, pp. 263-268.

[2] V. W. S. Chan, "Free-Space Optical Communications," J. of Lightwave Technology, vol. 24, no. 12, 2006, pp. 4750-4762.

[3] N. Perlot, "Characterization of Signal Fluctuations in Optical Communications with Intensity Modulation and Direct Detection Through the Turbulent Atmospheric Channel", PhD. dissertation, Shaker Verlag, Germany, 2006.

[4] K. Lee et al., "A CMOS serial link for fully duplex data communications," IEEE J. Solid-State Circuits, vol. 30, Apr. 1995, pp. 353–364.

[5] J. Kim and D. K. Jeong, "Multi-gigabit-rate clock and data recovery based on blind oversampling," IEEE Communications Magazine, vol. 41, no. 12, Dec. 2003, pp. 68- 74.

[6] M. A. Kossel and M. L. Schmatz, "Jitter measurements of high-speed serial links," IEEE Design & Test of Computers, vol. 21, no. 6, Nov.-Dec. 2004, pp. 536 – 543.

[7] SAWYER, N. Data Recovery. Xilinx application note XAPP224. Available at www.xilinx.com.