# In-system Jitter Measurement Based on Blind Oversampling Data Recovery

*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 describes a novel method for simple estimation of jitter contained in a received digital signal. The main objective of our research was to enable a non-invasive measurement of data link properties during a regular data transmission. To evaluate the signal quality we estimate amount of jitter contained in the received signal by utilizing internal signals of a data recovery circuit.*

*The method is a pure digital algorithm suitable for implementation in any digital integrated circuit (ASIC or FPGA). It is based on a blind-oversampling data recovery circuit which is used in some receivers instead of a traditional PLL-based clock and data recovery (CDR) circuit. Combination of the described jitter measurement block and the data recovery block forms a very efficient input part of the digital receiver. In such a configuration it is able to simultaneously perform both data communication (data recovery) and signal quality estimation (jitter measurement). The jitter measurement portion of the receiver requires no special connection of the received data signal. Thus the measured signal is not influenced by the measurement circuitry at all.*

*To verify the method we performed a measurement on a laboratory free-space optics link. Results of the measurement are satisfactory and can be used for on-line channel analysis.*

## Keywords

Jitter, Blind Oversampling Data Recovery, FPGA, ASIC, Time-to-Digital, Signal Quality, Bit Error Rate

## 1. Introduction

One of the basic measurements in the area of high-speed baseband communication is the jitter measurement. The amount of jitter contained in the received signal has a direct impact on the link performance and as such can be used, for example, as an indicator of the received signal quality [1]. Also, it can be used to estimate the bit error rate (BER) during regular data transmission, which is not achievable using the traditional BER measurement devices.

The traditional BER analyzers need to transmit a known data pattern in order to be able to detect errors. Another problem of the direct method is the relatively long measurement time required to achieve sufficient confidence level at low bit error rates. On the other hand, the indirect jitter-based BER measurement is usually somewhat less accurate. However, it can give us reliable results in a much shorter time while user data is being transmitted over the link. Thus it is possible to perform the adjustment of link parameters in real time during regular transmission (error correction strategy, link speed, etc.).

When measuring jitter in a traditional way (using an oscilloscope for example), it is necessary to split the received signal among the measurement instrument and the receiver itself. This is an invasive method of measurement that may significantly influence the measured signal or even corrupt the measurement completely. To be able to measure the received signal this way it is necessary to either use very sophisticated measurement tools (probes) or to design the receiver with having the possibility of future measurements in mind.

Up to now several digital methods of jitter measurement were published. They utilize internal signals of a receiver thus eliminating any influence of the measured signal. The jitter measurement method based on phase tracking (so called "follow me"; [2]) can capture only a low frequency jitter, which is of relatively low impact on BER performance. Another method, based on jitter histogram scanning [3], can give us more precise results but requires much longer measurement time. These methods were considered to be unsuitable for our application.

The newly developed method benefits from its target platform. It gives us both simple implementation and high frequency jitter measurement. It is based on a blind oversampling data recovery circuit (BO-CDR; [4], [5], [6], [7]). For testing purpose it was fitted into an FPGA utilizing its programmable logic. A variety of other platforms can be used to run the algorithm as well (ASIC, processor).

A significant benefit of combining the BO-CDR based data receiver with the proposed jitter measurement circuitry lies in the overall system simplification. This results in a lower part number and a smaller PCB area. The

received signal need not be split between the receiver and a jitter measurement device, thus the received (measured) signal is in no way affected. Moreover, the receiver is completely asynchronous, without any loopback, and thus inherently stable (unlike some fast-responding PLL-based systems [8]). On the other hand, the simplicity of the proposed method leads to somewhat lower accuracy.

Section 2 of the paper describes the basic principle of the measurement method; Section 3 deals with its implementation and introduces the experimental results.

## 2. The Method of Jitter Measurement

The jitter measurement algorithm utilizes internal signals of a blind oversampling data recovery. In principle it estimates edge density distribution function over one unit interval (UI). The unit interval is a time interval equal to the duration of one data symbol on a link. The whole unit interval is subdivided into several smaller equidistant subintervals by BO-CDR input oversampler. The number of subintervals depends on the oversampling ratio of the BO-CDR used and ranges from 3 to 8 for most implementations.

In Fig. 1 an example of received data signal sampling with oversampling ratio $M = 5$ is shown. The data signal is sampled by five mutually phase-shifted sampling clocks having the same frequency equal to the link data rate. The five subintervals in each unit interval defined by neighboring clocks are called sampling domains. In the example shown, edges are detected in the B-C domain thus clock E is chosen to sample the data as it is the sample nearest to the center of the symbol.

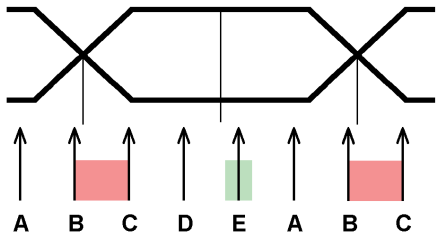


**Fig. 1.** Principle of edge detection and oversampling data recovery.

The proposed algorithm is counting number of detected edges in particular domains. This number is proportional to the probability density function (PDF) of edge occurrence over one UI. Thus it is possible to estimate the PDF of edge distribution (i.e. received data jitter) based on this simple statistic.

We can obtain a very authentic image of the actual PDF by increasing the oversampling ratio. This is very similar to traditional jitter measurement technique called time to digital [9], [10]. However, the hardware complexity of such a measurement device would not be suitable for implementation due to large number integrators. For practical reasons the oversampling ratio of the jitter measurement device will be always equal to the oversampling ratio of the used BO-CDR.

We have focused on the estimation of jitter RMS value $\sigma$ to obtain measurement results comparable with other measurement methods. The $\sigma$ is the main parameter of the random jitter PDF. To estimate $\sigma$, we have used the following equation that is based on a basic equation for standard deviation of discrete random variable

$$\sigma_{est} = \sqrt{\sum_{i=-\frac{M-1}{2}}^{\frac{M-1}{2}} \left( \left( \frac{i}{M} - \mu \right)^2 \cdot p_i \right)} \quad \text{[UI]} \tag{1}$$

where $\mu$ is the mean value of the distribution, and $p_i$ is the probability of edge occurrence in the $i$-th domain. The term $i/M$ represents center of the sampling domain. It is assumed that the maximum of PDF is located within the center sampling domain ($\mu$ is close to zero).

The method of $\sigma$ calculation can be considered to be reliable only for links showing random jitter distribution or very narrow double Dirac distribution [9]. A more complex jitter distribution results in a less accurate estimation of the RMS jitter value. However, even in such cases the method can be used to give a relative measure of signal quality.

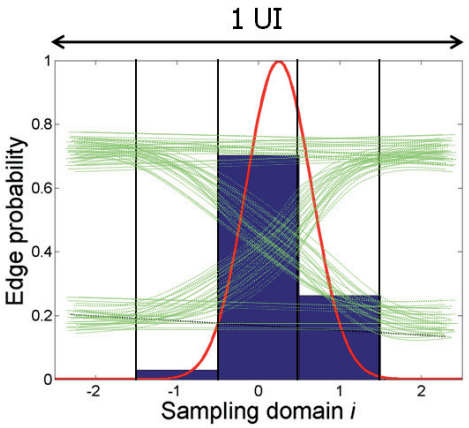


**Fig. 2.** Example of rough edge density estimation over one unit interval using the oversampling ratio $M = 5$.

To be able to estimate the jitter RMS we have to consider a frequency offset between transmitter and receiver. To explain the principle of the method we will first consider zero frequency offset. That means that the position of jitter histogram within the sampling domains is fixed (like the one in Fig. 2). Now let us denote $n_i$ the number of edges detected within a fixed measurement time in the $i$-th sampling domain, and $N$ the number of all edges detected during the measurement (including $n_i$). Those values can be measured easily in a system utilizing BO-CDR and can be used to calculate the measured edge density as

$$\tilde{P}_i = \frac{n_i}{N}. \tag{2}$$

This value is the required estimate of $p_i$ needed for RMS jitter calculation, which approximates the theoretical value

$$P_i(\mu) = \int_{i-0.5\cdot M^{-1}}^{i+0.5\cdot M^{-1}} PDF(\tau,\mu)\cdot d\tau \qquad (3)$$

where $\tau$ is the normalized time in UI, $PDF(\tau,\mu)$ is the probability density function of the measured jitter, and $\mu$ is the mean value relative to the center of the sampling domain $i = 0$.

In a real system there is always a frequency offset between the transmitter and the receiver reference clock. As a result, the mean value of the PDF is drifting slowly through the whole UI. Without any synchronization the measured long-time edge density would be same over all domains. That's why the algorithm must track on the sampling domain with currently highest edge density. The tracing can be easily accomplished by utilizing internal signal for selecting an optimum sampling domain of the BO-CDR [5].

The tracing mechanism BO-CDR adjusts phase in discrete steps of $1/M$. Considering the uniform distribution of μ over interval $<-0.5/M, 0.5/M>$ (see Fig. 3), the theoretical value of the edge density is equal to the long time average

$$R_i = M \int_{-0.5\cdot M^{-1}}^{0.5\cdot M^{-1}} P_i(\mu)d\mu\,. \qquad (4)$$

$R_i$ is an edge density actually measured in a real system using the described method. By substituting $p_i = R_i$ in (1) we can calculate a so called pseudo RMS jitter value

$$\sigma_D = \sqrt{\sum_{i=-\frac{M-1}{2}}^{\frac{M-1}{2}} \left[\left(\frac{i}{M}\right)^2 \cdot R_i\right]}\,. \qquad (5)$$

This value is not a real jitter RMS as it is affected by the drift as illustrated in Fig. 3. This phase drift distorts shape of jitter PDF seen by the sampler by convoluting it with a pulse represented by a sampling domain. The distorted PDF can be seen in Fig. 4 as the blue curve. The red curve in Fig. 3 represents measured (integrated) edge density for the distorted distribution, while the green curve is actual jitter PDF that we want to characterize.

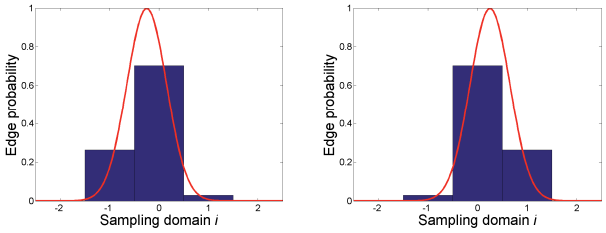


**Fig. 3.** Drift of a maximum of the edge density within a single sampling domain due to frequency offset between transmitter and receiver.

Due to the phase drift effect given by (4) it is necessary to make a correction of the calculated pseudo value $\sigma_D$ to get the real value of jitter RMS. We have assumed the normal distribution of the random jitter. Then $\sigma_D$ can be easily expressed as a function of standard deviation $\sigma$ of the normal distribution using (3), (4), and (5)

$$\sigma_D = f(\sigma)\,. \qquad (6)$$

This calculation is relatively complex for an embedded system and thus it is more efficient first to evaluate (6) by a computer and then to use the calculated values as a look-up table during the measurement itself. The correction function can be seen in Fig. 5.

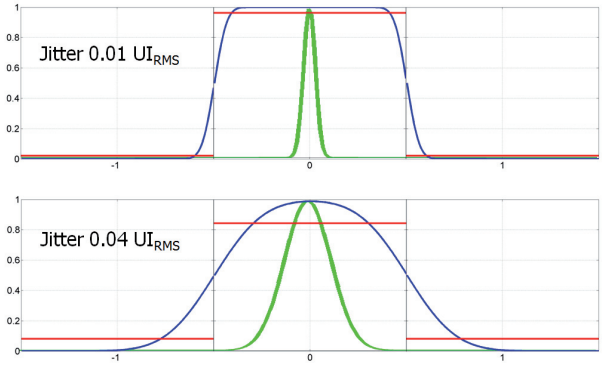


**Fig. 4.** Impact of the phase drift on the measured signal; the green curve is actual jitter PDF, the blue curve represents convolution of the actual jitter PDF with sampling domain pulse and the red one represents actually measured edge density in particular domains (interval integral of the blue curve).

The phase drift complicates the implementation of the algorithm because of the correction function. However, it ensures that the jitter can be measured even for a very low RMS value, i.e. when the major portion of the PDF fits within a single sampling domain (like the one shown in the upper example in Fig. 4). In case of very low jitter RMS value it would not be possible to measure the jitter with acceptable precision by using low oversampling ratios (like $M = 5$). For our method the low oversampling ratio is sufficient and the measurement resolution (precision) is even higher for low jitter levels, thanks to the small initial slope of the correction function.

There are applications where an exact RMS jitter value is irrelevant and a relative signal quality indication is sufficient. In such a case the correction calculation can be omitted, thus simplifying the system significantly.

Please note that wander (slow variations in frequency) and low frequency jitter cannot be evaluated by the proposed device as these are filtered-out by the data recovery circuit via phase tracking the received data signal. On the other hand, these are usually irrelevant for most systems as they have only marginal impact on BER [11]. As the jitter measurement device is driven by the data recovery circuit, it is ensured that each and every phase variation causing an error is detected.

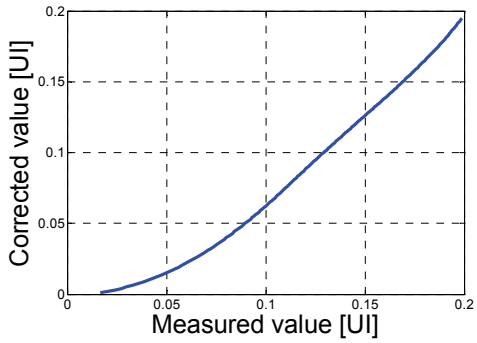


**Fig. 5.** The relationship of *measured value* ($\sigma_D$) and *corrected value* ($\sigma$) of measured jitter RMS due to the phase drift effect (6).

## 3. The Measurement

To verify the quality of the proposed method several real measurements were performed. The simplified measurement setup is shown in Fig. 6. Please note that no external reference clock signal is necessary for the measurement, which is in contrast to other recently published methods [10], [12], [13].

The whole receiver frontend including the jitter measurement device was implemented in Virtex-5 FPGA populated on the Xilinx ML-505 development board. The FPGA was connected directly to a conditioned received signal of an experimental 125 Mb/s free-space optic link receiver (System under test).

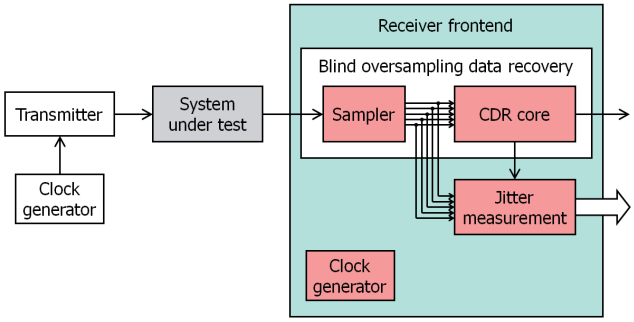


**Fig. 6.** Comparison of RMS jitter measured by oscilloscope and by the proposed device implemented on an FPGA.

Hardware requirements for algorithm implementation are very low. We decided to use 20 bit binary counters for integrator implementation. It is worth mentioning, that it is only necessary to implement one full-length counter for the central sampling domain and another ones for the remaining domains, as the edge density is always the largest in the central domain (for reasonable signal quality). Overall hardware cost is only 32 slices of the Virtex-5 FPGA for the integrators while the maximum design frequency is unaffected by adding the jitter measurement functionality. The rest of the algorithm is implemented in a soft-core microprocessor, which have been already present in the design.

The processor performs several tasks regarding the jitter measurement. First it takes care of reset of the integrators. When choosing the integration time we have to trade between time resolution and precision of the measurement. For our implementation we are using relatively low integration time as we want to monitor fast changes of the measured channel properties. Because the integration time is defined by software, it is very easy to change it and it can be adjusted even during the measurement. The second task of the processor is calculation of measurement correction (as shown in Fig. 5) by performing a look-up to a correction table saved in a processor RAM. This look-up can be alternatively implemented in hardware by utilizing few more slices and one BRAM block of the FPGA. The last task of the processor regarding the jitter measurement is data presentation. In our case we are using a terminal application to log the statistics so the processor is configured to continuously send the measured data over the serial link to a PC.

The measurement results captured using the FPGA-based jitter measurement device were compared with reference values acquired using an oscilloscope (Tektronix DPO7254). As can be seen in Fig. 7, there is only little difference between the two measurements, caused mainly by mid-frequency jitter components (frequencies close to the BO-CDR tracking bandwidth).

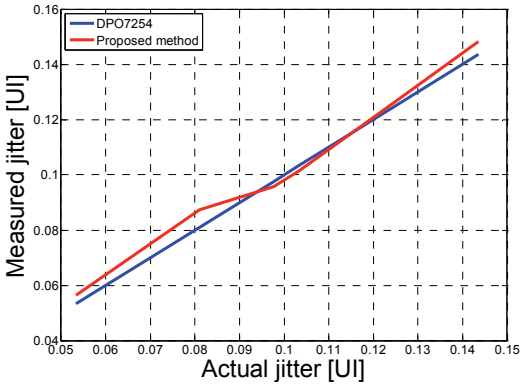


**Fig. 7.** Comparison of RMS jitter measured by oscilloscope and by the proposed device implemented on an FPGA.

Our further research in this area will be focused on improvement of the measurement precision by more detailed analysis of the measured edge density over all sampling domains.

## 4. Conclusion

We have presented a novel, very efficient, fully digital circuitry for jitter (signal quality) estimation. It utilizes internal signals of a blind oversampling data recovery circuit. It offers an opportunity to measure the received signal quality during a regular transmission without affecting the signal path (i.e. without any impact on the measured received signal). Moreover, due to very low hardware requirements of the proposed method, it is possible to eas-

ily implement the whole receiver (signal conditioning, CDR, signal quality measurement, data processing) into a single chip (FPGA or ASIC) and thus simplify the final device.

Accuracy of the proposed system was verified using a reference measurement. Considering the simplicity of the measurement we can state that the error of measured jitter is relatively small. The method can provide a simple measure of signal quality for adjusting parameters of a transceiver (like forward error correction scheme) but it in current setup is not suitable for reference measurements.

Both the full VHDL source code for the algorithm and the Matlab code for correction function calculation are available upon request at authors.

## Acknowledgements

This work has been supported by the Czech Science Foundation under grant No P102/11/1376, by the Czech Ministry of Industry and Trade under grant agreement No. FR-TI4/148 and by the project CZ.1.07.2.3.00.20.0007 WICOMT, financed from the operational program Education for Competitiveness. 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 described research was performed in laboratories supported by the SIX project; the registration number CZ.1.05/2.1.00/03.0072, the operational program Research and Development for Innovation.

## References

[1] *Total Jitter Measurement at Low Probability Levels, Using Optimized BERT Scan Method*. [online] Agilent White Paper, 2005. Available at WWW:<http://www.home.agilent.com/ agilent/ redirector.jspx?action=ref&cname=AGILENT_EDITORIAL&ckey=682410&lc=eng&cc=US>

[2] BORGOSZ, J. "Follow Me" - Digital jitter measurement method. In *Measurement Science Review*, 2006, vol. 6, sect. 3. ISSN 1335 – 8871

[3] KUBÍČEK, M. In-system jitter measurement using FPGA. In *Proceedings of 20th International Conference Radioelektronika 2010*. Brno (Czech Republic), 2010. p. 187 - 190. ISBN 978-1-4244-6319-0.

[4] JOU, S.-J., LIN, C.-H., CHEN, Y.-H., LI, Z.-H. Design and analysis of digital data recovery circuits using oversampling. *IET Circuits Devices Systems*, 2007, vol. 1, no. 1. ISSN: 1751-858X

[5] KUBÍČEK, M., KOLKA, Z. Blind oversampling data recovery with low hardware complexity. *Radioengineering;* 2010, vol. 19, no. 1, p. 74 - 78.

[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, ISSN 1549-7747.

[7] BUEREN, G., RODONI, L., JAECKEL, H., HUBER, A., BRUN, R., HOLZER, D., SCHMATZ, M. 5.75 to 44Gb/s quarter rate CDR with data rate selection in 90nm bulk CMOS. In *34th European Solid-State Circuits Conference ESSCIRC 2008*. Edinburg (UK), Sept. 2008, p. 166-169. ISBN 978-1-4244-2361-3

[8] 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, 2008.

[9] MILLER, M., SCHNECKER, M. *A Comparison of Methods for Estimating Total Jitter Concerning Precision, Accuracy and Robustness*. [Online], 2007. Cited 2009-08-05. Available at: <http://www.lecroy.com/tm/Library/WhitePapers/PDF/LeCroy_Jitter_Methods_DesignCon2007.pdf>

[10] MARINS, C. N. M., et al. *New Jitter Measurement Technique Using TDC Principle in a FPGA Component*. 4 pages. [Online] Cited 2009-08-05. Available at: http://www.inatel.br/cict/ downloads/doc_download/2737-antonio-alves-ferreira-junior.

[11] YIN, J., ZENG, L.-G. A statistical jitter tolerance estimation applied for clock and data recovery using oversampling. In *Proceedings of IEEE Region 10 Conference TENCON*, 2006.Hong Kong (China), 2006, p. 1-4. ISBN 1-4244-0548-3

[12] JAE WOOK LEE, JI HWAN CHUN, ABRAHAM, J.A. A random jitter RMS estimation technique for BIST applications. In *Proceedings of the 2009 Asian Test Symposium*. ATS '09. Taichung (Taiwan), 2009, p. 9-14, ISBN 978-0-7695-3864-8.

[13] LI, M. P., JINHUA CHEN New methods for receiver internal jitter measurement. In *IEEE International Test Conference. ITC 2007*.Santa Clara (USA), Oct. 2007, p. 1-10, ISBN 978-1-4244-1127-6.

## About Authors ...

**Michal KUBÍČEK** was born in Svitavy, Czech Republic, in 1983. He received his M.Sc. from Brno University of Technology in 2006 and Ph.D. degree at the Department of Radio Electronics in 2010 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 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.