Programmable Jitter Generator

Goran Jovanovic[1], Mile Stojcev[1] and Tatjana Nikolic[1]

Abstract – **As CMOS technology has scaled, supply voltage have dropped, chip power consumption has increased, and clock frequency/data rates increase effects of jitter become critical and jitter budget get tighter. Knowing how to inject/isolate jitter components with time-convolution/correlation will enhance designer ability to determine and locate the root causes so that he/she can then proceed to 'beat down' individual error components one at a time in order to improve reliable system performance. Jitter can be decomposed into several subcomponents, each having specific sets of characteristics and root causes. This paper begins with a short review of jitter fundamentals including a discussion of the various random and deterministic jitter components, and injection method of jitter subcomponents into computer clock signal and/or communication data stream. The jitter injection technique gives test engineers an insight into how jitter components interact. In the rest of the paper a global hardware structure of a jitter generator, which uses digital techniques, based on a voltage controlled delay line is described. A Xilinx xc3s500e-5fg320 FPGA chip is used to validate this design. The programmable jitter generator can be used in jitter tolerance test for computer system and jitter transfer function measurement in communication systems.**

Keywords – **Jitter, Jitter generator, Jitter classification.**

I. INTRODUCTION

With continuous scaling of CMOS IC technology and increase of clock frequency, it is becoming increasingly difficult to guarantee the availability of correct clock signal throughout the chip and system due to: i) the increasing likelihood of manufacturing defects in VLSI ICs (process variation, power supply noise, etc.); and ii) influence of external environment (injected noise, crosstalk, etc.). Therefore, understanding what timing jitter is, and how to characterize it, is the first step during the process of designing high-performance, high-speed, and reliable digital systems [1, 2].

The variations of the timing signal's rising and falling edges as compared to the perfect reference are defined as jitter, and the corresponding time-variation measurements are specified in time units such as picoseconds [2, 3].

In order to evaluate system performance with respect to jitter tolerance, a jitter generator is indispensable. The jitter generator should generate and injects different kinds of jitter into the input data stream in a controllable fashion. Many works have been reported on jitter measurement and analysis [4, 5, 6, 7]. It is relatively simple to measure each component but is challenging to generate, measure, and analyze them if multiple jitter components are simultaneously injected into a data stream or in distributed clock signals.

In this paper, we propose a programmable jitter generator, which targets all the aforementioned challenges. Our intent is to determine how jitter's components can be modeled and combined, and how the total jitter can be changed according to different injection sequences.

This paper is structured as follows: Section II deals with jitter classification. Different types of jitter are discussed and their root causes are identified. Section III explains the meaning of terms unit period and eye diagram. Section IV concentrates on decomposition and jitter modeling. In Section V a global structure of jitter generator is involved and the role of its main building blocks is described. Section VI concentrates on combining method for jitter generation. In Section VII implementation details of jitter generator are given. Section VIII summarizes the conclusions.

II. JITTER CLASSIFICATION

Jitter can be categorized in many different ways [8]. We will use categorization based on the phenomenological properties of the jitter itself. This classification is between jitter that is random and jitter that is not. Jitter that is not random is bounded, that is, its magnitude is finite. In contrast, random jitter is unbounded and, within physical limits, can theoretically reach any magnitude. Fig. 1 shows diagram of jitter classifications, moving from total jitter at the left to the detailed jitter mechanisms at the right of the diagram. Total jitter, J^T, results from combination of random jitter, J^R, and deterministic jitter, J^D (see Fig. 1).

Random jitter is the jitter generation from the accumulation of random processes including thermal noise and shot noise. J^R cannot be predicted, because it has no discernable pattern. In regards to J^R, the following are examples of sources that can cause random jitter: a) thermal noise - due to electron flow in conductors; and b) shot noise - due to electron and hole flow in semiconductors. By its nature J^R is theoretically unbounded and Gaussian in distribution which is characterized by a mean, μ, and a width, σ, as shown in Fig. 2.

Deterministic jitter is the jitter generation from a variety of systematic effects. It arises from the interaction of different system components. The major causes of J^D include electromagnetic interference, crosstalk, signal reflections, driver slew rate, skin effects, and dielectric loss. J^D can be divided further into the following two subclasses: j) jitter that is correlated to data sequence or pattern; and jj) jitter occurring independent of the data.

Jitter that is correlated to the data pattern can be broken into two categories: data dependent jitter J^D_{DD} and periodic jitter J^D_P. Duty cycle distortion jitter, J^D_{DCD}, and inter-symbol interference jitter, J^D_{ISI}, are pattern dependent, and they are part of a jitter class called data dependent jitter, J^D_{DD}. There are two common causes of J^D_{DCD}: i) The slew rate of the rising

[1]Goran Jovanovic, Mile Stojcev and Tatjana Nikolic are with the Faculty of Electronic Engineering, Aleksandra Medvedeva 14, 18000 Nis, Serbia, E-mail: goran.jovanovic{mile.stojcev, tatjana.nikolic}@elfak.ni.ac.rs.

edges differs from that of the falling edges; ii) The decision threshold for a waveform is higher or lower than it should be.

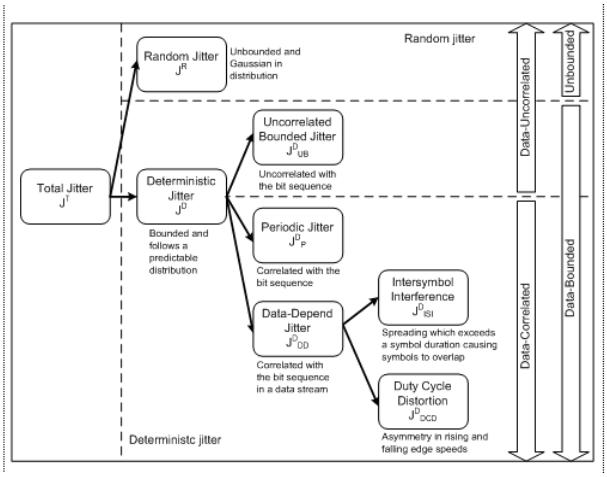
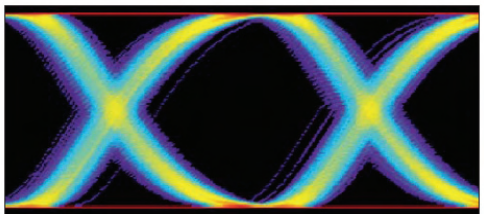
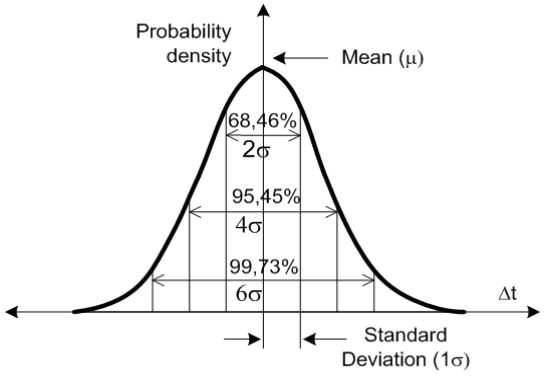


Fig. 1. Total jitter subcomponents



a)



b)

Fig. 2. a) An eye diagram; and b) Histogram for Gaussian distribution
Notice: A Gaussian power density function (PDF) with corresponding probability area of 2σ, 4σ i 6σ widths covering 68,46%, 95,45% and 99,77% of the underneath area

Fig. 3 shows waveform typical for duty-cycle distortion caused by non-symmetrical rise/fall times and incorrect detection threshold.

Fig. 3 (upper part) demonstrates the first case. Here, the decision voltage is at 50% amplitude point but the slow rise time of the waveform causes the rising edges to cross the threshold latter than the falling edges. Fig. 3 (bottom part) presents the second case, in which the waveform has balanced rise and fall times but the decision threshold is not set at the 50% amplitude point.

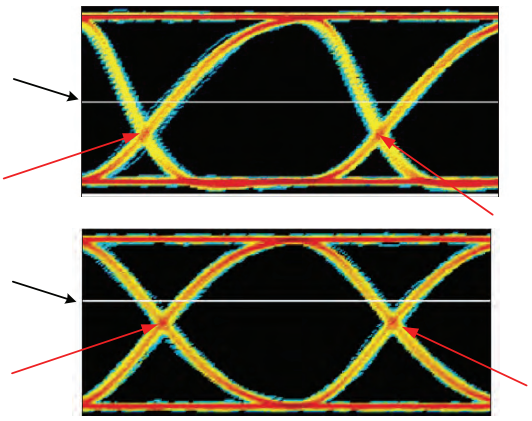


Fig. 3. Waveforms showing duty cycle distortion

J^D_{ISI}: jitter is usually the result of a bandwidth limitation problem in either the transmitter or physical media. Fig. 4 presents how bandwidth limitations produce inter-symbol interference timing.

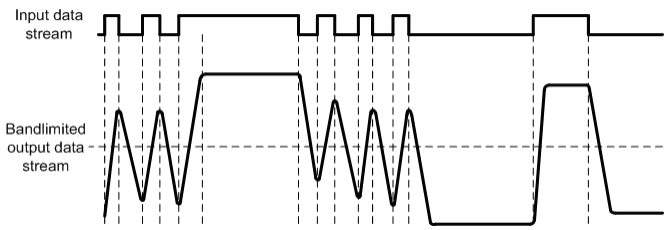


Fig. 4. Inter-symbol interference due to bandwidth limited problem

J^D_{ISI} has three main causes. a) Bandwidth limitation problem: limited bandwidth produces limited edge speeds, and limited edges speeds will result in varying pulse amplitudes at high-speed data rates. Varying pulse amplitudes will then result in transition timing errors; b) Signal reflections: arise due to improper terminations or impedance anomalies within the physical media. Signal reflections usually produce distortions in the amplitude of the data signal; and c).Non-linear phase response of the transmission media: these limitations cause frequency-dependent group delay. For example, the skin effect is proportional to the square root of the frequency, while the dielectric loss is proportional to the frequency [9]. Therefore, the skin effect dominates data loss at a lower frequency, whereas the dielectric loss dominates at a higher frequency. The nonlinear response causes edge shifts that depend on the transition density within the data stream.

Periodic jitter deals with jitter that is periodic and correlated to the data, and whose frequency is an integer sub-rate of the data rate. The source of J^D_P is usually interference from signals related to data pattern, ground bounce or power supply variations.

Bounded uncorrelated jitter, J^D_{UB}, can be due to bounded but non periodic sources. This kind of jitter appears as a consequence of coupling from adjacent data-carrying links or on-chip random logic switching [9].

III. Referent Period and Eye Diagram

Jitter is typically measured and specified in picoseconds from some reference edge and over some number of specified cycles. It's possible to express jitter in absolute time normalized to a unit interval, UI. A UI is the ideal or average time duration of a single bit or the reciprocal of the average data rate. Another method is to express the jitter as a percentage of the reference period. In many applications the reference period is call a unit interval, UI, and the jitter is specified as a ratio of percentage of the UI. For example, for clock frequency of 200 MHz with ± 50 ps of jitter or a total jitter of 100 ps, we have 1 UI= 5000 ps. Jitter as ratio of UI is 100ps/5000ps=0.02 UI, or 2% UI.

Traditionally, an eye diagram, like that shown in Fig. 5a), has served to specify signal integrity limits, including jitter. An eye diagram is a composite of all bit periods of the captured bits superimposed on each other relative to a bit clock (recovered or available from the source). We call the area within the eye the eye opening. In Fig. 5b) arrows are used to show the vertical and horizontal events in the eye opening. As the noise in a signal increases the eye becomes less open, either horizontally or vertically or both. The eye is said to be closed when non open area remains in the center of the diagram.

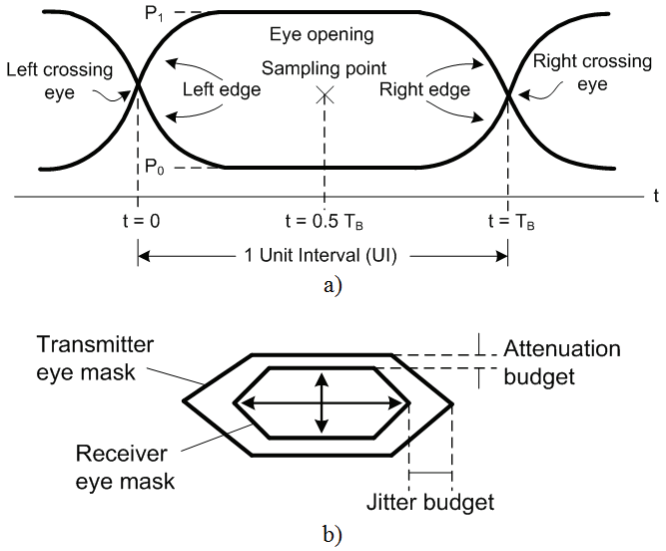


Fig. 5. Eye diagram; a) Eye diagram with definition of terms; b) Overlapping the masks gives an indication of attenuation and jitter budget

IV. Jitter Decomposition and Modelling

To understand how real systems behave, it is often useful to use a mathematical model of the system. The behavior of such a model can be tuned by adjusting the parameters of its individual components. If the parameters of the model are chosen based on observations of the real system, then the model can be used to predict the behavior of the system in other situations. Thus one of the motivations for jitter decomposition (also called jitter separation) is to extrapolate system performance to cases that would be difficult or time-consuming to measure directly.

Another motivation for modeling the system this way has to do with analysis. If each of the model components is associated with one or more underlying physical effects, an understanding of the model can provide insight into the precise cause or causes of excessive jitter.

Researchers and engineers commonly model J^R by the Gaussian distribution function [2, 10]

$$J^R(\Delta t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\Delta t-\mu)^2}{2\sigma^2}} \quad (1)$$

where $J^R(\Delta t)$ denotes J^R probability density function, PDF, σ is a standard deviation of the Gaussian distribution, and (Δt–μ) is time displacement relative to the ideal time position. This function is characterized as un-bounced because its PDF is not zero unless the jitter Δt approaches infinity. No matter, how large value Δt we peak, the probability is never zero. From mathematical point of view the mean of Gaussian form equals μ, and its standard deviation equals σ.

J^D_{DD} can be modeled through linear time invariant, LTI, system in which an ideal data pattern is the input of the LTI. J^D_{DD} is calculated from the output waveform via its deviation of the edges transition times from the corresponding ideal edges transition times [10].

The J^D_{DCD} can be best modeled by the dual-Dirac delta function [2, 10]. The sum of two δ functions that represent J^D_{DCD} PDF is:

$$J^D{}_{DCD}(\Delta t) = \frac{\delta\left(\Delta t - \frac{w}{2}\right) + \delta\left(\Delta t + \frac{w}{2}\right)}{2} \quad (2)$$

where w is peak-to-peak duty-cycle-distortion magnitude, and Δt is the time displacement relative to the ideal time position.

J^D_{ISI} is caused by timing spread of various pulses with different run lengths within the transmitted pattern. To calculate total J^D_{ISI} it is necessary to know probability of occurrence of each edge pattern and the corresponding jitter magnitude [10]. Let pi denotes the probability that given bit pattern i will occur, and Δt_i corresponds to the magnitude of the bit pattern. If the jitter magnitude of each distinct edge pattern remains constant over time, than weighted some of δ functions can be used to represent the PDF of each edge, with the weights corresponding to the edge pattern probability. Accordingly J^D_{ISI} PDF can be expressed as [10].

$$J^D_{ISI}(\Delta t) = \sum_{i=1}^{N} p_i * \sigma(\Delta t - \Delta t_i) \quad (3)$$

where: N is the number of distinct edge patterns, pi is the probability of occurrence of edge pattern i, Δt_i is the jitter magnitude of the i-th edge pattern, Δt is the time displacement relative to the ideal time position.

J^D_P is a repeating jitter signal at a certain period or frequency, and is viewed as bounded and uncorrelated

narrow-band jitter. If we assume that the J^D_P is sinusoidal it can be described mathematically by the following

$$\Delta t = A\cos\left(\omega t + \phi_0\right) \tag{4}$$

where ω is angular frequency and ϕ_0 is the initial phase.

The conclusions established can apply well to other periodic jitters with different profiles, such as rectangular, triangular, saw-tooth, or trapezoid.

In general, the model of the total periodic jitter J^D_P is summation of cosine functions with phase deviation, modulation frequency, and peak amplitude [2].

$$J_P^D\left(\Delta t\right) = \sum_{i=1}^{N} A_i \cos\left(\omega_i t + \phi_i\right) \tag{5}$$

where N is the number of cosine components (tones).

PDF for sine J^D_P (single tone)

$$J_{P\_SIN}^D\left(\Delta t\right) = \begin{cases} \dfrac{1}{\pi\sqrt{\dfrac{m}{2} - \left(\sqrt{\dfrac{2}{m}}\Delta t\right)^2}} & for\ \left|\Delta t\right| < \dfrac{m}{2} \\ 0 & otherwise \end{cases} \tag{6}$$

Uncorrelated bounded jitter, J^D_{UB}, can be modeled by many independent periodic jitters if it is composed by many independent root sources. Its time-domain PDF is truncated Gaussian, and is defined as [10]:

$$J_{UB}^D\left(\Delta t\right) = \begin{cases} \dfrac{p_{UB}^J}{\sigma_{UBJ}\sqrt{2\pi}} e^{-\frac{\Delta t^2}{2\sigma_{UBJ}^2}} & for\ \left|\Delta t\right| \le A_{UB} \\ 0 & for\ \left|\Delta t\right| > A_{UB} \end{cases} \tag{7}$$

where A_{UB} is the peak value, σ_{UBJ} is the sigma value, and P^J_{UB} is the normalized probability for the J^D_{UB} PDF.

Assuming that each jitter component is already known, the total jitter probability density function is given by the convolution of the PDF's of each component [8, 10].

$$J^T\left(t\right) = J^R\left(t\right) * J_{DCD}^D\left(t\right) * J_{ISI}^D\left(t\right) * J_P^D\left(t\right) * J_{UB}^D\left(t\right) \tag{11}$$

where "*" is a convolution operation.

V. Structure of Jitter Generator

The structure of a jitter generator, JG, is given in Fig. 6. Main building blocks of JG are:

Voltage controlled delay line, VCDL: composed of 32 controlled buffer-cells (D_0, D_1, ..., D_{31}) of the same structure and delay time.

Histogram logic, HL: consists of two constituent. The first is called pseudorandom generator and is implemented as a linear feedback shift register, LFSR, with five D flip-flops and five feedback taps. Having in mind the speed of operation, Galois method was used for realization of LFSR. The second constituent is realized as memory block in which different PDF patterns are stored.

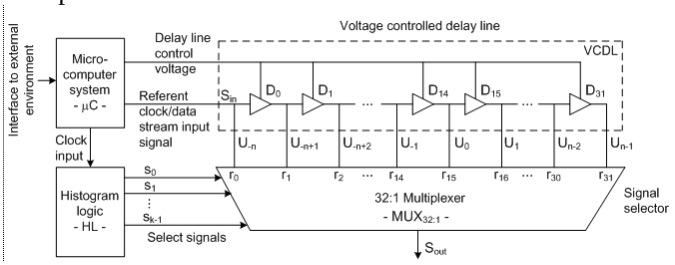


Fig. 6. Structure of jitter generator

Signal selector, MUX32:1: digital multiplexer which has one output pin S_{out} and 32 input pins named as r_0, r_1, ..., r_{31}, respectively. The signal selection lines s_0, s_1, ..., s_{31}, select a delay buffer-cell and connect it to the output pin S_{out}.

Microcomputer system, μC: used as a control block for JG. From one side it is coupled with the external environmental (keyboard, display, etc.) and from the other side it generates control signals for driving VCDL (delay line control voltage, and referent clock/data stream input signal), and clock signal for driving the HL.

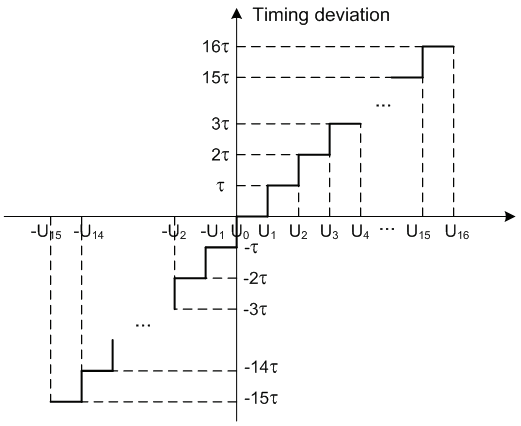


Fig. 7. Time deviations at output of JG

In computer system the signal Sin is input jitter-free clock signal, while in communication systems Sin corresponds to a data stream. In our case, for both applications, Sin is used as a signal into which jitter is injected. All buffer-cells involve identical delay τ, so that the total timing deviation $\Delta T = 32*\tau < T/2$, where T corresponds to the period of a clock signal in computer systems, or data–bit–interval in communication systems. If we take that the central buffer-cell's output represents a referent signal, its rising edge is referred to as phase zero (see Fig. 7). This means that the delay chain is able to generate 17 positive and 15 negative phase deviations.

VI. Combining Methods for Jitter Generation

In real signals multiple jitter components are simultaneously present. Bearing this in mind, our objective now was to determine how different jitter components, in a

controlled way, can be combined, and how the composition of a total jitter can be changed according to different injection sequences. Fig. 8 illustrates the combining method. The sequence of each jitter model can be interchanged to study the impact of different injection sequences. The jitter combining scheme was simulated using Matlab. In order to predict the overall system jitter the separate components were first analyzed individually, after that developed and characterized, and then combined using Matlab. The Matlab simulation shows us how jitter components combine and how the total jitter depends on the jitter injection sequence.

During the combining process some of the jitter components can be switched on or off. Our goal during jitter analysis was to understand how different jitter components combine to form the jitter distribution. In that sense, like illustration, some examples of individually injected jitter components are shown in Fig. 9. The left upper part of Fig. 9 shows random jitter with Gaussian distribution, while its right upper part corresponds to mixture of triangular periodic jitter and uncorrelated bounded jitter. The left bottom part of Fig. 9 presents convolution of sinusoidal periodic jitter and data correlated intersymbol interference jitter. Finally, the right bottom part of Fig. 9 gives combination of Gaussian, sinusoidal periodic and uncorrelated bounded jitter.

From jitter-measurement point of view, there are several benefits to separating jitter and injecting components in a clock/data stream [11]:

Once the jitter has been broken down in this way it is possible to extrapolate the performance of a system under test at very low probability levels without measuring trillions of events, offering the promise of greatly reduced testing times.

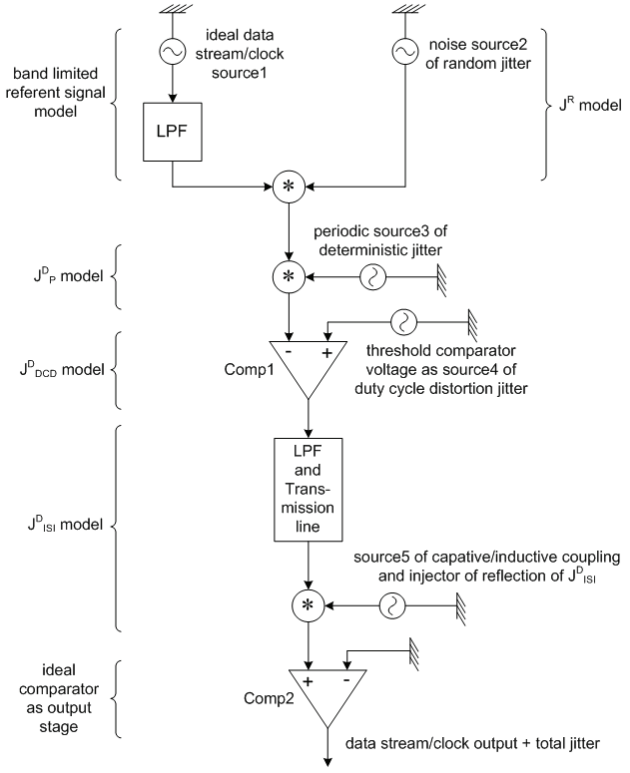


Fig. 8. Combining method for jitter

Each jitter component has one or more known causes and well-understood effect on the probability of bit errors. Thus, an efficient design strategy for reducing some jitter components can be developed.

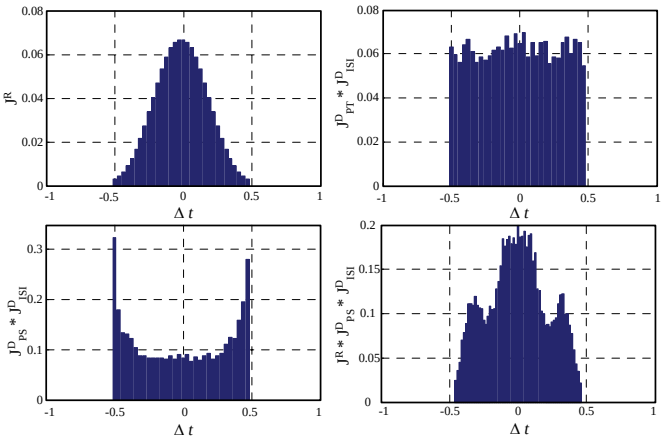


Fig. 9. PDF of jitter using the convolution method

There are two major approaches to separating individual jitter components at the receiver side. One method is based on the jitter PDF or cumulative distribution function, CDF, measurement. PDF is the normalized histogram of the signal edge times (i.e., how often the signal transitions at a particular time point). CDF, on the other hand, sorts the sampled edge time data in an ascending order to show the distribution profile. Another method is based on jitter time record. Jitter PDF can be measured by instrument such as sampling oscilloscope or time interval analyzer. Jitter CDF (also called BER CDF) can be measured by bit-error-rate tester. The jitter time record can be measured by time interval analyzer or real time oscilloscope [11, 12].

VII. Implementation

In order to estimate the performance, the jitter generator structure was described at RTL level using VHDL and has been implemented using Xilinx FPGA technology. Target device xc3s500e-5fg320 from Spartan3E series is selected. For synthesis, routing and mapping a Xilinx development CAD tool Xilinx ISE Design Suite 12.3_1 was used.

In the current design, the voltage controlled delay line was realized with 32 fixed buffer delay cells since in the FPGA chip there is no voltage-controlled buffer. In the histogram logic the pseudorandom generator was implemented as five taps LFSR, while the RAM block as an array of size 1024*5 bits. Fig. 10 depicts the simulation results that correspond to phase deviation generation. The waveforms correspond to the outputs of a delay line. At seventeen VCDL's outputs we have positive, while at fifteen outputs negative phase deviation.

Details which relate to implementation of a jitter generator on a FPGA chip xc3s500e-5fg320 from Spartan 3E series are given in Table I.

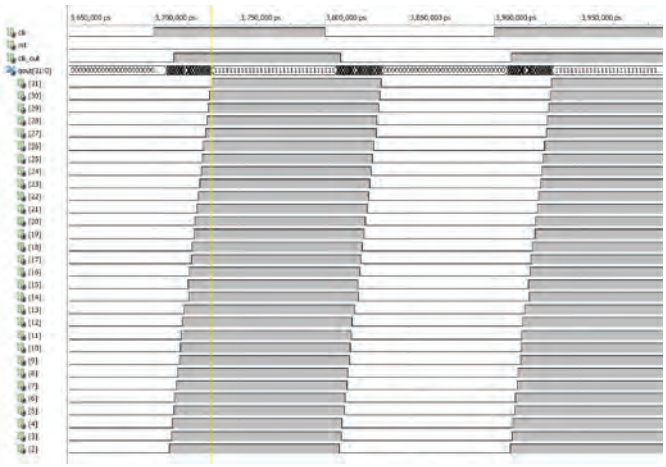


Fig. 10 Simulation results

TABLE I
DEVICE UTILIZATION SUMMARY

Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	10	9,312	1%
Number of 4 input LUTs	49	9,312	1%
Number of occupied Slices	29	4,656	1%
Number of Slices containing only related logic	29	29	100%
Number of Slices containing unrelated logic	0	29	0%
Total Number of 4 input LUTs	49	9,312	1%
Number of bonded IOBs	35	232	15%
Number of RAMB16s	1	20	5%
Number of BUFGMUXs	1	24	4%
Average Fanout of Non-Clock Nets	3.24		

According to the results given in Table I we can conclude that the hardware overhead due to implementation of a jitter generator on FPGA chip is really minor. Utilization of flip/flops, four-input LUTs, and number of occupied slice is always 1%, and number of RAM bits is about 5%.

VIII. CONCLUSION

As speed as well as the desire for improved performance increases, the amount of jitter in system or product will become more of concern than it is today. However, by having a fundamental understanding of these effects, better choice will be made in design and characterization of a product, which in turn, will lead to improved system reliability.

A good jitter generator must be capable of controlling the jitter frequency and the magnitude of jitter amplitude. Here, the jitter amplitude represents the amount of the phase shift of a data stream signal or the computer clock signal, while the jitter frequency is the times of generation a phase shift. Currently in the market, there are several jitter generator instruments that may satisfy these demands, however they are expensive and not suitable for being utilized in mass production. In this paper we present relatively simple and low-cost programmable jitter generator which can be used for jitter tolerance test and jitter transfer function measurement. The jitter generator is implemented using digital circuits based on chip from FPGA family, and software. In near future we plan to build this jitter generator into a CMOS VLSI IC, such as dual-processor fault-tolerant system for automotive.

ACKNOWLEDGEMENTS

This work was supported by the Serbian Ministry of Science and Technological Development, Project No. TR-32009 – "Low-Power Reconfigurable Fault-Tolerant Platforms".

REFERENCES

[1] C. Metra, M. Omana, TM. Mak, A. Rabman, S. Tam, Novel On-Chip Clock Jitter Measurement Scheme for High Performance Microprocessors, IEEE International Symposium on Defect on Fault Tolerance of VLSI Systems, 2008, pp. 465-473

[2] N. Ou, T. Farahmand, A. Kuo, S. Tabatabaei, A. Ivanov, Jitter Models for the Design and Test of Gbps-Speed Serial Interconnects, IEEE Design & Test of Computers, Vol. 21, No. 4, July-August 2004, pp. 302-313

[3] J. Solorzano, Jitter basic, part I, Wavecrest Corp., Application note 2004, http://www.micrel.com/hbw_news/Jitter_Article_Part%201.pdf

[4] S. Sunter, A. Roy, On-Chip Digital Jitter Measurement from Megahertz to Gigahertz, IEEE Design & Test, Vol. 21, No. 4, July-August 2004, pp. 314 – 321

[5] "Jitter Analysis Techniques for High Data Rates, Agilent Application Note 1432, http://cp.literature.agilent.com/litweb/pdf/5988-8425EN.pdf

[6] T. Xia, P. Song, K. A. Jenkins, J.C.Lo, "Delay Chain Based Programmable Jitter Generator", Proc. of the 9-th IEEE European Test Symposiums (ETS), May 2004

[7] M. Fleischer-Reumann, "Jitter Tolerance Testing", Chapter 10, pp. 533-561 in Digital Communications Test and Measurement, by Dennis Derickson, Marcus Muller, Editors, Prentice Hall, Upper Saddle River, 2008.

[8] Marcus Muller, Ransom Stephens, Jitter Basics, Chapter 2, pp. 29-60, in Digital Communications Test and Measurement, by Dennis Derickson, Marcus Muller, Editors, Prentice Hall, Upper Saddle River, 2008.

[9] Horward W. Johnson, Martin Graham, High-Speed Digital Design: A Handbook of Black Magic, PTR Prentice Hall, New Jersey, 1993.

[10] M.P. Li, Jitter, Noise, and Signal Integrity at High Speed, Prentice Hall, Upper Saddle River, NJ, 2008

[11] Greg D. Le Cheminant, Characterizing High Speed Digital Communications Signals and Systems with the Equivalent – Time Sampling Oscilloscopes, Chapter 7, pp. 329-419, in Digital Communications Test and Measurement, by Dennis Derickson, Marcus Muller, Editors, Prentice Hall, Upper Saddle River, 2008.

[12] "Understanding and Characterizing Timing Jitter" Tektronix application note 55W-161146-0, Sept. 2007.