

Programmable jitter generator based on voltage controlled delay line

G. Jovanovic, M. Stojcev, T. Nikolic, Z. Stamenković

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 the time convolution/correlation method will enhance designer ability to determine and locate the root causes so that one can then proceed to ‘beat down’ individual error components one at a time in order to improve system performance. Jitter can be decomposed into several components, 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, the 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 the jitter tolerance test for computer systems and jitter transfer function measurements in communication systems.

Keywords: Jitter, Jitter generator, Jitter classification.

1 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, particles, etc.); and ii) influence of the external environment (injected noise, crosstalk, etc.). Therefore, understanding the clock timing jitter and its characterization is the first step in the process of designing reliable high-performance high-speed digital systems [1, 2].

Manuscript received September 28, 2011 ; accepted December 22, 2011.

Goran Jovanovic, Mile Stojcev, Tatjana Nikolic are with the faculty of Electronic Engineering, Niš, Serbia; Zoran Stamenković is with the IHP GmbH, Innovations for High Performance Microelectronics Leibniz-Institut fuer innovative Mikroelektronik, Im Technologiepark 25, 15236 Frankfurt (Oder) Germany

More generally speaking, the variations of any timing signal 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 inject 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 jitter component separately but is challenging to measure and analyze multiple jitter components having them simultaneously injected into a data stream or in a distributed clock signal.

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 2 deals with jitter classification. Different types of jitter are discussed and their root causes are identified. Section 3 explains the meaning of terms like unit period and eye diagram. Section 4 concentrates on jitter decomposition and modeling. In Section 5 a global structure of the jitter generator is involved and the roles of its main building blocks are described. Section 6 concentrates on a combining method for jitter generation. In Section 7 implementation details of the jitter generator are given. Section 8 summarizes the conclusions.

2 Jitter classification

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

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

Deterministic jitter is caused by 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 a data sequence or pattern; and jj) jitter occurring independently of data.

Jitter that is correlated to the data pattern can be split into two classes: data dependent

jitter J_{DD}^D and periodic jitter J_P^D . Duty cycle distortion jitter J_{DCD}^D and inter-symbol interference jitter J_{ISI}^D are pattern dependent, and they are part of the data dependent jitter class. There are two common causes of J_{DCD}^D : i) The slew rate of the signal rising edges differs from that of the signal falling edges; ii) The decision threshold for a waveform is higher or lower than it should be.

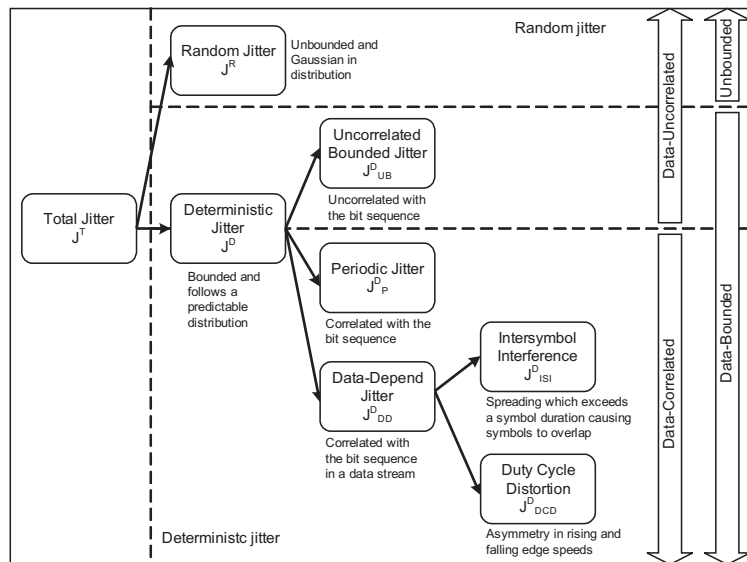


Fig. 1: Total jitter components

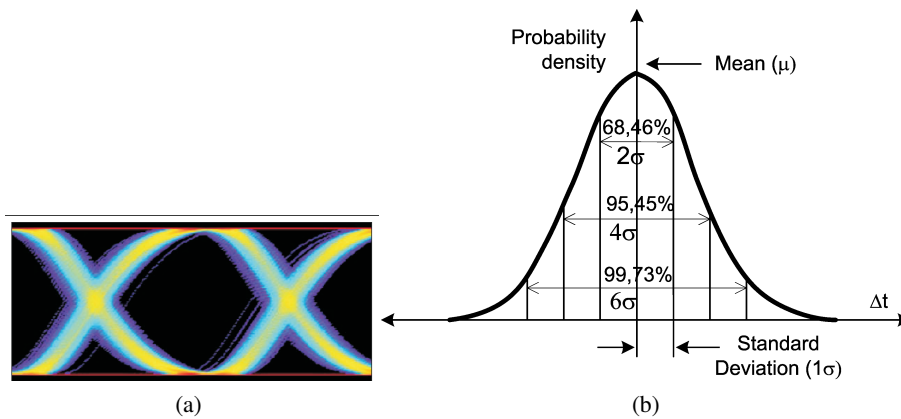


Fig. 2: a) An eye diagram; and b) Histogram for Gaussian distribution Notice: A Gaussian probability density function (PDF) with the corresponding probability area for 2 , 4 and 6 widths

Fig. 3 shows waveforms typical for duty-cycle distortion caused by the non-symmetrical rise/fall times and incorrect detection threshold. Fig. 3a demonstrates the first case. Here,

the decision voltage is at 50% amplitude point but the long rise time of the waveform causes the rising edges to cross the threshold latter than the falling edges. Fig. 3b presents the second case, in which the waveform has the balanced rise and fall times but the decision threshold is not set at the 50% amplitude point.

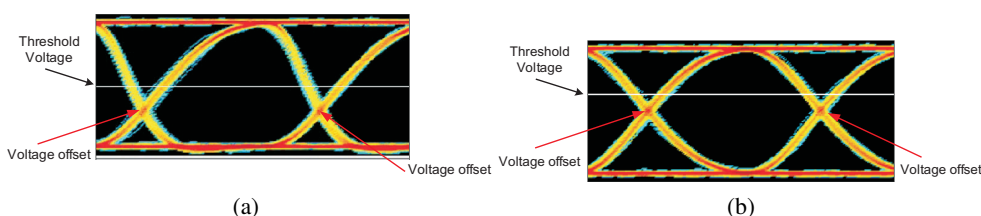


Fig. 3: Waveforms showing duty cycle distortion

The J_{ISI}^D jitter is usually a result of the bandwidth limitation of either transmission or physical media. Fig. 4 presents how bandwidth limitations produce the inter-symbol interference.

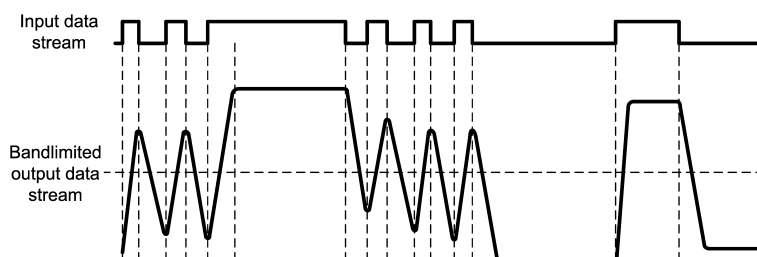


Fig. 4: Inter-symbol interference due to a limited bandwidth

J_{ISI}^D has three main causes: a) Bandwidth limitation produces limited edge speeds and, in turn, the limited edge speeds will result in varying pulse amplitudes at high data rates. The 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. They usually produce distortions in the amplitude of data signal; c) Non-linear phase response of the transmission media causes the 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 non-linear response causes edge shifts that depend on the transition density within the data stream.

Periodic jitter is correlated to the data and its frequency is an integer sub-rate of the data rate. The source of J_p^D is usually interference coming from data signals, ground bounce or power supply variations.

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

3 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 is possible to express jitter in the absolute time normalized to a unit interval (UI). Unit interval 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 called 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 $1 \text{ UI} = 5000 \text{ ps}$. Jitter as ratio of UI is $100\text{ps}/5000\text{ps} = 0.02 \text{ UI}$, 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). The inner area of the diagram is called 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 no open area remains in the center of the diagram.

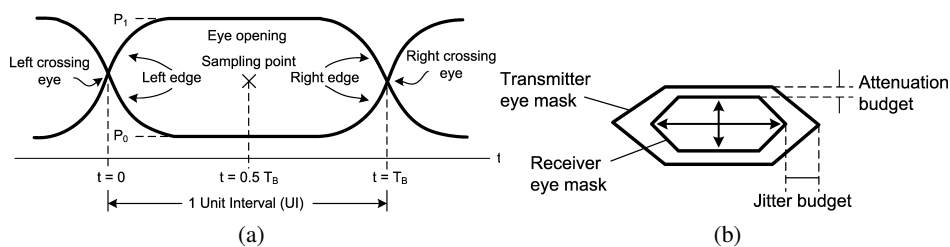


Fig. 5: a) Eye diagram with definition of terms; b) Overlapping of the transmitter and receiver masks gives an indication of attenuation and jitter budget

4 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 the J^R probability density function, σ is the standard deviation of the Gaussian distribution, and $(\Delta t - \mu)$ is the 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.

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

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

$$J_{DCD}^D(\Delta t) = \frac{\delta(\Delta t - \frac{w}{2}) + \delta(\Delta t + \frac{w}{2})}{2} \quad (2)$$

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

J_{ISI}^D is caused by timing spread of various pulses with different run lengths within the transmitted pattern. To calculate total J_{ISI}^D , it is necessary to know the probability of occurrence of each edge pattern and the corresponding jitter magnitude [10]. Let p_i 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_{ISI}^D PDF can be expressed as [10]

$$J_{ISI}^D(\Delta t) = \sum_{i=1}^N p_i * \delta(\Delta t - \Delta t_i) \quad (3)$$

where N is the number of distinct edge patterns, p_i 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_p^D 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 J_p^D is sinusoidal, it can be described mathematically by the following expression

$$\Delta t = A \cos(\omega t + \phi_0) \quad (4)$$

where ω is the 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_P^D is the summation of cosine functions with phase deviation, modulation frequency, and peak amplitude [2].

$$J_P^D(\Delta t) = \sum_{i=1}^N A_i \cos(\omega_i t + \varphi_i) \quad (5)$$

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

PDF for sine J_P^D (single tone)

$$J_{P_SIN}^D(\Delta t) = \begin{cases} \frac{1}{\pi \sqrt{\frac{m}{2} - \left(\sqrt{\frac{2}{m}} \Delta t\right)^2}} & \text{for } |\Delta t| < \frac{m}{2} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Uncorrelated bounded jitter J_{UB}^D can be modeled as combination of independent periodic jitters if it is caused by independent root sources. Its time-domain PDF is truncated Gaussian defined by [10]

$$J_{UB}^D(\Delta t) = \begin{cases} \frac{P_{UB}^J}{\sigma_{UBJ} \sqrt{2\pi}} e^{-\frac{\Delta t^2}{2\sigma_{UBJ}^2}} & \text{for } |\Delta t| \leq A_{UB} \\ 0 & \text{for } |\Delta t| > A_{UB} \end{cases} \quad (7)$$

where A_{UB} is the peak value, σ_{UBJ} is the sigma value, and P_{UB}^J is the normalized probability for the J_{UB}^D probability density function.

Assuming that each jitter component is already known, the total jitter probability density function is given by the convolution of the PDFs of all components [8, 10]

$$J^T(t) = J^R(t) * J_{DCD}^D(t) * J_{ISI}^D(t) * J_P^D(t) * J_{UB}^D(t) \quad (8)$$

where “*” is a convolution operator.

5 Structure of jitter generator

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

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

Histogram logic (HL) with two constituents. 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.

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

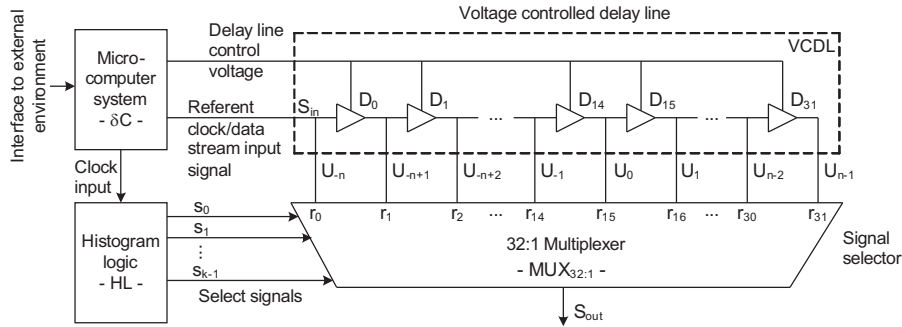


Fig. 6: Structure of jitter generator

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

In a computer system, the S_{in} signal is an incoming jitter-free clock signal, while in communication systems S_{in} corresponds to a data stream. In our case, for both applications, S_{in} is used as a signal into which jitter is injected. All buffer-cells involve identical delay τ , so that the total timing deviation $\Delta T = 32 * \tau g T / 2$, 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.

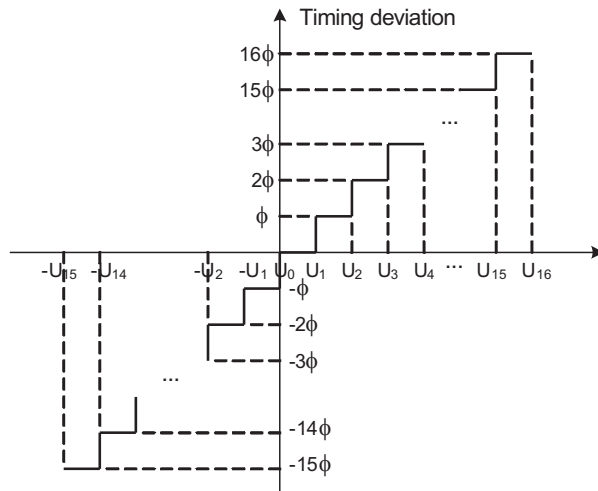


Fig. 7: Time deviations at output of JG

6 Combining methods for jitter generation

In real signals, multiple jitter components are always present. Bearing this in mind, our objective now is to determine how different jitter components, in a controlled way, can be combined, and what are the implications of different jitters combinations on the total jitter. Fig. 8 illustrates a jitter combining method. The jitter models can be interchanged to study the impact of different injection sequences. The jitter combining scheme was simulated using the Matlab modeling and simulation tool. In order to predict the overall system jitter, the separate jitter components were first analyzed individually, after that developed and characterized, and then combined. The Matlab simulations showed us how the 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 the random jitter with Gaussian distribution, while its right upper part corresponds to a mixture of the triangular periodic jitter and uncorrelated bounded jitter. The left bottom part of Fig. 9 presents a convolution of the sinusoidal periodic jitter and data correlated intersymbol interference jitter. Finally, the right bottom part of Fig. 9 gives a combination of the Gaussian, sinusoidal periodic, and uncorrelated bounded jitter.

From jitter-measurement point of view, there are several benefits of separating and injecting jitter components in a clock or 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 with a very high fidelity without measuring trillions of events and offering greatly reduced testing times.

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.

There are two major approaches to separating individual jitter components at the receiver side. One method is based on the jitter's cumulative distribution function (CDF) measurements. 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 the jitter time record. Jitter's PDF can be measured by an instrument such as the sampling oscilloscope or time interval analyzer. Jitter's CDF (also called BER CDF) can be measured by the bit-error-rate tester. The jitter time record can be measured by the time interval analyzer or real time oscilloscope [11, 12].

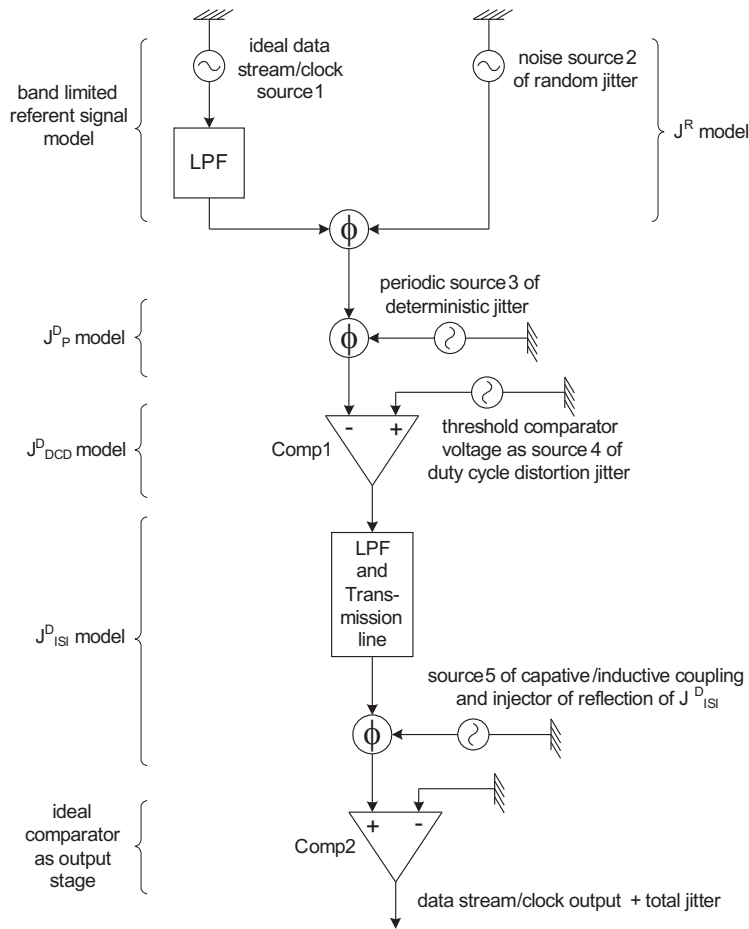


Fig. 8: Combining method for jitter

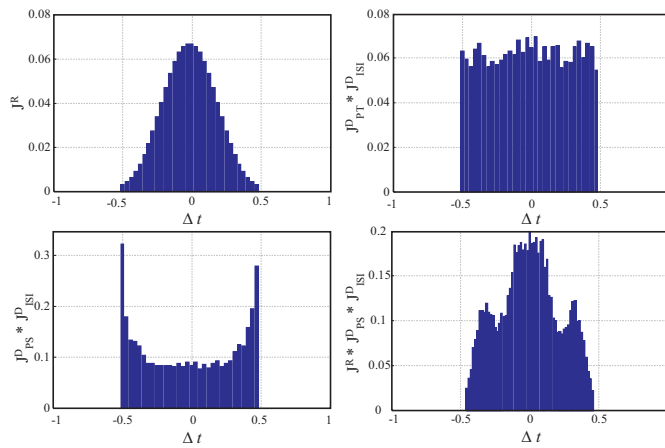


Fig. 9: Probability density functions of different jitters obtained using the convolution method and Matlab simulations

7 Implementation

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

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

Details of a jitter generator implemented on the xc3s500e-5fg320 FPGA chip are given in Table I.

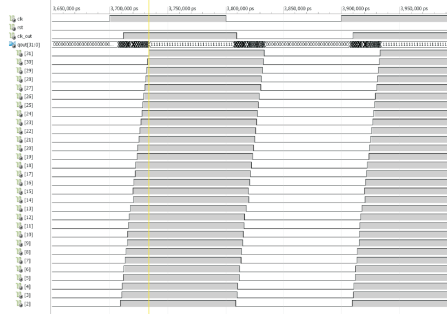


Fig. 10: Simulation results

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%
Fanout of Non-Clock Nets	3.24		

Table 1: Device utilization summary

According to the results given in Table I, we can conclude that the hardware overhead

due to implementation of a jitter generator is really minor. Utilization of flip-flops, four-input LUTs, and logic slices is about 1%. Utilization of RAM blocks is about 5%.

8 Conclusion

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

A good jitter generator must be capable of controlling the jitter frequency and amplitude. Here, the jitter amplitude represents the amount of the phase shift of the data stream or computer clock signal, while the jitter frequency corresponds to the speed of phase shift generation. Currently in the market, there are several jitter generator instruments that may satisfy aforementioned demands. However, they are expensive and not suitable for being utilized in mass production. In this paper, we have presented a relatively simple and low-cost programmable jitter generator which can be used for the jitter tolerance test and jitter transfer function measurements. The jitter generator is implemented using digital FPGA circuits. In near future, we plan to build this jitter generator as a CMOS ASIC and use it in a dual-processor fault-tolerant device for automotive applications.

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.