Int. J. Communications, Network and System Sciences, 2009, 4, 249-324
doi:10.4236/ijcns.2009.24028 Published Online July 2009 (http://www.SciRP.org/journal/ijcns/).

True Random Bit Generator Using ZCDPLL Based on TMS320C6416

Qassim NASIR
Department of Electrical and Computer Engineering College of Engineering, University of Sharjah, Sharjah, UAE
Email: *nasir@sharjah.ac.ae*
Received September 4, 2008; *revised December* 8, 2008; *accepted February* 6, 2009

ABSTRACT

A True Random Binary Generator (TRBG) based on a zero crossing digital phase-locked loop (ZCDPLL) is proposed. In order to face the challenges of using the proposed TRBG in cryptography, the proposed TRBG is subjected to the AIS 31 test suite. The ZCDPLL operate as chaotic generator for certain loop filter gains and this has been used to generate TRBs. The generated binary sequences have a good autocorrelation and cross-correlation properties as seen from the simulation results. A prototype of TRBG using ZCDPLL has been developed through Texas Instruments TMS320C6416 DSP development kit. The proposed TRBG successfully passed the AIS 31 test suit.

Keywords: True Random Binary Sequence, Zero Crossing Digital Phase Locked Loop, Spreading Sequences, Cryptography, TMS320C64x

1. Introduction

Random and pseudo-random numbers are used in many areas including test data generation, Monte-Carlo simulation techniques, generation of spreading sequences for spread spectrum communications, and cryptography [1]. Pseudo-random spreading sequences used in spread spectrum communications must be repeatable, while for most simulations using random numbers this is not necessary. In cryptographic applications, security depends on the randomness of the source and the unpredictability of the used random bits [1].

Chaotic circuits represent an efficient alternative to classical TRBG [2]. Studies in nonlinear dynamics show that many of the seemingly complex systems in nature are described by relatively mathematical equations [3]. Although chaotic systems appear to be highly irregular, they are also deterministic in the sense that it is possible to reproduce them with certainty. These promising features of chaotic systems attracted many researchers to try chaos as a possible medium for secure communication.

The nonlinear phenomenon of chaos poses a promising alternative for pseudo-random number generation due to its unpredictable behaviour.

The chaotic system generates "unpredictable" pseudo random orbits which can be used to generate RNGs (Random Number Generators). Many different chaotic systems have been used to generate RNGs such as Logistic map [4], and its generalized version [5], Chebyshev map, [1] piecewise linear chaotic maps [6] and piecewise nonlinear chaotic maps [7]. Chaotic systems are characterized by a "sensitivity dependence on initial conditions", and with such initial uncertainties, the system behaviour leads to large uncertainty after some time.

The aim of this paper is to show how to use a second order zero crossing digital phase locked loop (ZCDPLL) operating in a chaotic mode as True Random Binary Generator (TRBG). Digital Phase locked Loops (DPLLs) were introduced to minimize some of the problems associated with the analogue loops such as sensitivity to DC drift and the need for periodic adjustments [8,9]. The most commonly used DPLL is the Zero Crossing Digital Phase Locked Loop (ZCDPLL). The ZCDPLL operation is based on non uniform sampling techniques. The loop is simple to implement and easy to model. The ZCDPLL consists of a sampler that acts as phase detector, a digital filter, and a Digital Controlled Oscillator (DCO) [8].

The global dynamics of the second order ZCDPLL shows chaos operation for certain values of filter gains [10]. The non-linear behaviour of ZCDPLL shows period doubling or bifurcation instabilities to its route to chaos. The chaotic behaviour has been confirmed through the use of phase error spectrum, bifurcation diagram, and Lyapunov exponent [3,10]. The proposed TRBG pass the statistical tests described by AIS 31 documents [11] However, one should notice that the statistical tests prove that the generator is an ideal random bit generator (the tests are in fact necessary but not sufficient) [12]. The proposed TRBG is implemented on a Texas Instruments TMS320C6416 DSP development platform [13].

The remainder of the paper is organized as follows. In Section 2, the ZCDPLL model is presented. Random bit sequence generation using ZCDPLL is presented in Section 3. Simulation results are discussed in Section 4. Section 5 draws the conclusion.

2. Second Order ZCDPLL

The structure of second order ZCDPLL is shown in Figure 1. As soon as the filter finishes its operation, the stored data are transferred to the register II. The input signal to the loop is taken as $x(t) = s(t) + n(t)$, where $s(t) = A\sin(\omega_0 t + \theta(t))$, and $n(t)$ is additive white Gaussian Noise (AWGN); $\theta(t) = \theta_0 + \Omega_0 t$ by which the signal dynamics are modelled; θ_0 is the initial phase which we will assume to be zero; Ω_0 is the frequency offset from the nominal value ω_0. The input signal is sampled at time instances t_k determined by the Digital Controlled Oscillator (DCO). The DCO period control algorithm is given by [14] is

$$T_k = T_0 - c_{k-1} = t_k - t_{k-1} \quad \textbf{(1)}$$

where $T_0=(2\pi/\omega_0)$ is the nominal period, c_{k-1} is the loop digital filter. The sample value of the incoming signal x(t) at t_k is

$$x(t_k)=s(t_k)+n(t_k) \quad \textbf{(2)}$$

Or simply

$$x_k=s_k+n_k \quad \textbf{(3)}$$

where $s_k = A\sin(\omega_0 t_k + \theta(t_k))$. The sequence x_k is passed through a digital filter whose transfer function is $D(z)$s whose output ck is used to control the period of the DCO. For noise free analysis, then

$$x_k=A\sin[\omega_0(kT_0-\sum_{i=0}^{k-1}c_i+\theta_k] \quad \textbf{(4)}$$

The phase error is defined to be

$$\phi_k=\theta_k-\omega_0\sum_{i=0}^{k-1}c_i \quad \textbf{(5)}$$

Then

$$\phi_k-\phi_{k-1}=\theta_k-\theta_{k-1}-\omega_0 c_{k-1} \quad \textbf{(6)}$$

Using z-operator, equation (6) can be written as

$$(1-z^{-1})\phi_k=(1-z^{-1})\theta_k-\omega_0 z^{-1}c_k \quad \textbf{(7)}$$

The control signal c_k is the output of the digital filter and is formed by

$$c_k = D(z)x_k = D(z)\sin(\phi_k) \quad \textbf{(8)}$$

Substituting (8) into (7) yields

$$\phi_k=\theta_k-\frac{\omega_0 z^{-1}D(z)}{1-z^{-1}}A\sin(\phi_k) \quad \textbf{(9)}$$

In the second order ZCDPLL, the digital filter $D(z)=G_1+\frac{G_2}{1-z^{-1}}$, then (6) becomes

$$\phi_k=2\phi_{k-1}-\phi_{k-2}+K_1\sin(\phi_{k-2})-rK_1\sin(\phi_{k-1})=f(\phi_k) \quad \textbf{(10)}$$

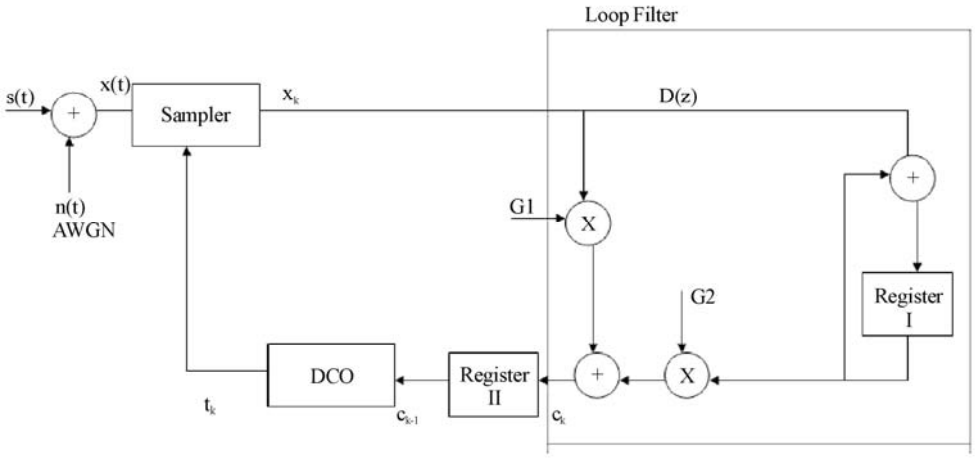

Figure 1. Block diagram of the second order ZCDPLL.

where $K_1 = AG_1\omega_0$ and $r = 1 + \frac{G_2}{G_1}$ Let $x_x=\phi_k$, $y_x=\phi_{k-1}$, $\mathbf{X} = (x, y)^T$ then (1) can be rewritten as

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} 2x_k - y_k + K_1 \sin(y_k) - rK_1 \sin(x_k) \\ x_k \end{pmatrix} \\ \vdots \quad = \begin{pmatrix} g(\mathbf{X}_k) \\ g(\mathbf{X}_k \end{pmatrix} = G(x_k) \tag{11}$$

If the above system equation is linearized around the equilibrium points $\mathbf{X}^*_k=0$, so that $\sin(x^*_k) \approx x^*_k$ and $\sin(y^*_k) \approx y^*_k$. Then (11) becomes

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} (2 - rK_1)x_k + (K_1 - 1)y_k \\ x_k \end{pmatrix} \\ = \begin{Bmatrix} (2 - rK_1) & (K_1 - 1) \\ 0 & 1 \end{Bmatrix} \begin{pmatrix} x_k \\ y_k \end{pmatrix} \tag{12}$$

Consequently the Jacobian $\mathbf{G'}(x)=\partial g_i/\partial x$ is given by

$$G'(x^*) = \begin{Bmatrix} (2 - rK_1) & (K_1 - 1) \\ 0 & 1 \end{Bmatrix} \tag{13}$$

The loop will converge if the eigen values $\mathbf{G'}(x^*)$ are be less than one. Following [14] the operational regions of the second order ZCDPLL are given by (as shown in Figure 2): region (I), the loop converges locally to $\mathbf{x}^*=0$, $2n\pi$, region (II), the loop phase error ϕ oscillates between two values, region (III), the loop phase error ϕ oscillates between n values or diverges, while in region (IV), the loop phase error ϕ diverges.

Fundamental to most definitions of chaos is the concept that two trajectories of the system, no matter how closely they start to one another, will eventually diverge. This divergence is of exponential order. The Lyapunov exponent is used to measure the average rate of divergence of nearby trajectories. It is defined as [3,10]:

$$\lambda = \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N} \ln \left| f'(\phi_n) \right| \tag{14}$$

A positive Lyapunov exponent indicates chaos. The largest Lyapunov exponent for the two dimensional a dynamical ZCDPLL system is defined as [3]

$$\lambda = \lim_{N\to\infty} \frac{1}{2N} \sum_{n=0}^{N} \ln \left| \frac{(a + bY_n')^2 + (c + dY_n')^2}{1 + Y_n'^2} \right| \tag{15}$$

where Y' is the tangent of the direction of maximum growth which evolves according to

$$Y'_{n+1} = \frac{c + dY'_n}{a + bY'} \tag{16}$$

where $a = \partial g_1/\partial x_k, b = \partial g_1/\partial y_k, c = \partial g_2/\partial x_k, a = \partial g2/\partial y_k,$ are members of Jacobian matrix of (13).

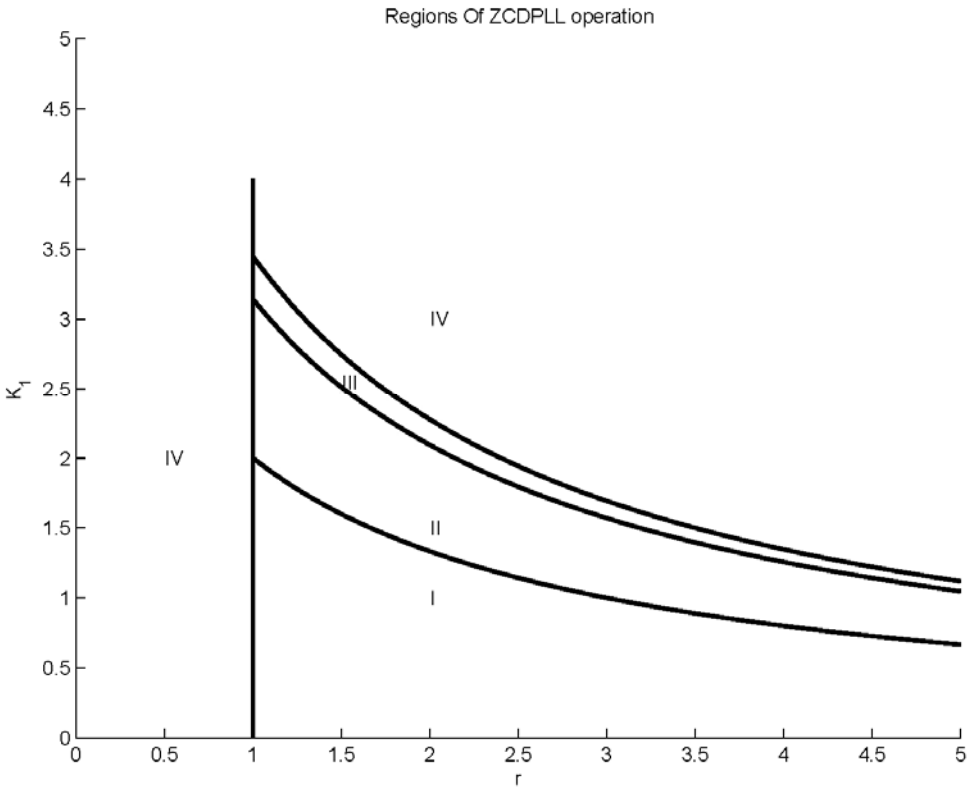

Figure 2. Loop behaviour as a function of K_1 and r for second-order ZCDPLL.

3. True Random Binary Generation-TRBG

Security algorithms and protocols rely on the unpredictability of the random bits they use, True Random Binary Generators (TRBGs) play an important role in cryptographic applications [15] Classical TRBG uses some random physical phenomenon. The most frequently used phenomenon in embedded TRBGs is a jitter noise of digital clock signals [16].

Chaotic circuits represent an efficient alternative to the classical TRBGs [16,17]. Contrary to traditionally used sources of randomness they use a well-defined analog deterministic circuit that exhibits chaos. Many random number generators based on analog and deterministic chaotic phenomena have been proposed, see [16,17].

To obtain random bits from ZCDPLL, the output of the sampler is monitored and assigned bit values of zero and one based on whether the output of the sampler is positive or negative as it can be seen in Figure 3. Hence, $\varphi \in [0,\pi)$ corresponds to a "1" and $\varphi \in [\pi, 2\pi)$ corresponds to a "0".

A natural source of random bits may not give unbiased bits as direct output. Many applications, especially in cryptography, rely on sequences of unbiased bits. There are various techniques to extract unbiased bits from a defective generator with unknown bias. These are called de-skewing techniques [18]. These techniques also eliminate the correlation in the output of the natural sources of random bits. Von Neumann [19] proposed a digital post-processing technique that balances the distribution of bits. Post-processing converts non-overlapping pairs of bits into output bits by converting the bit pair [0,1] into an output 0, input pair [1,0] into an output 1, input pairs [1,1] and [0,0] are discarded [18].

A large number of statistical test suits have been proposed to assess the statistical properties of random number generators. A finite set of statistical tests may only detect defects in statistical properties. Furthermore, the tests cannot verify the unpredictability of the random sequence which is demanded in cryptographic applications. In this section, the proposed random bit generator is evaluated with statistical tests proposed by BSI as Applications Notes and Interpretation of the Scheme (AIS) [11]. The statistical tests are applied to a bit sequence obtained after the de-skewing. A better result of the statistical test can be obtained, if a strong de-skewing technique is used such as a cryptographic hash function, but these create a long delay [18].

Although random numbers play an important role in numerous cryptographic applications, Information Technology Security Evaluation Criteria (ITSEC) [20] nor Common Criteria [21] (CC) specify any uniform evaluation criteria for random numbers. For this purpose the AIS 31 Document Standard describes some evaluation criteria for true (physical) random number generators. AIS has been effective in Germany since 2001 [11]. The monobit, poker, runs, long runs, and entropy tests have been used in this paper as defined by AIS 31 document in order to test the randomness of the chaotic bits sequence generated by the ZCDPLL.

4. System Performance

The TRBG using ZCDPLL has been verified by simulation. A modulation free input signal $x(t)= \sin(\omega_1 t)$ is considered, where ω_1 is the input frequency, and the center frequency of the DCO is assumed to be $\omega_0 = 1$ rad/sec.

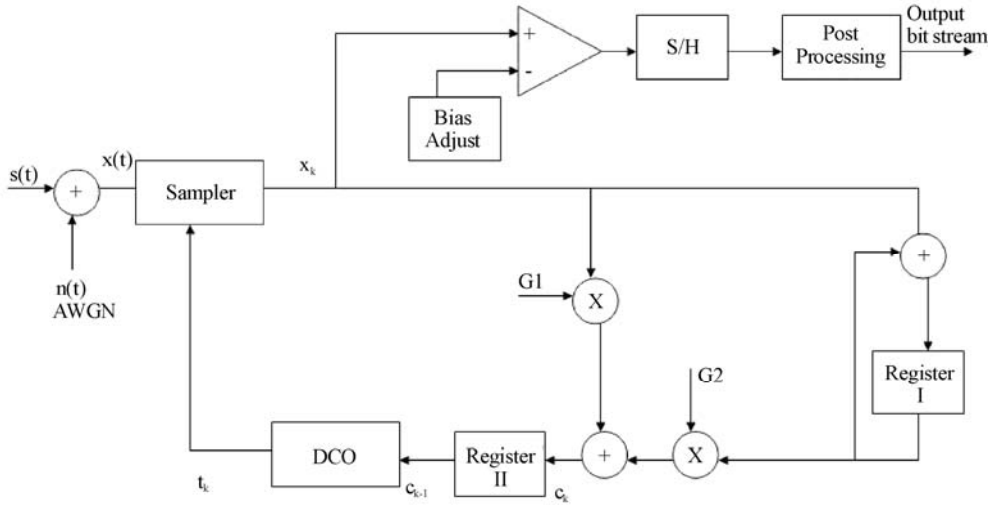

Figure 3. Block diagram of the TRBG using second order ZCDPLL.

In order to see where the loop has chaotic behavior, Lyapunov exponent has been calculated for all possible values of the loop parameters (K_1 and r) and has been given black dot when they have values greater than 1 (positive Lyapunov represents chaotic operation). The distribution of these black dots is shown in Figure 4. In order to produce chaotic sequence of phase error, one should work in black regions.

The chaotic phase φ_k distribution generated by the second order ZCDPLL distribution is shown in Figure 5 where K_1 =2.5, and r =2.5. Sample 2 is shown in Figure 6 for K_1 =3.0, and r =2.5. The two phase sequences are random. The one million bits extracted from the phase sequence of ZCDPLL have been collected ($\varphi \in [0,\pi)$ corresponds to a "1" and $\varphi \in [\pi, 2\pi)$ corresponds to a "0". The autocorrelation properties of the outputs are shown in Figure 7. Chaotic sequence have very low cross correlation as shown in Figure 8 for the two binary streams generated using two different loop parameters. This is an important issue with regard to the security, because the receiver can not be determined from a few points in the sequence.

The chaotic bit stream generated by the second order ZCDPLL of length 20000 bits is subject to each of the tests. The Monobit test is projected as evidence if the number of 1's and 0's in the sequence are nearly equal. The AIS standard specified the value of the number of 1's in the bit stream to be somewhere between 9654 and 10346. The Pocker Test requires the division of the initial sequence into 4 bit contiguous segments, the counting and the storing of each of the 16 possible 4 bit values. Denoting as f (k) the number of each value, $0 \le k \le 15$, the X statistic is computed by means of (The test will pass if $1.03 \le X \le 57.4$):

$$X = \frac{16}{5000}\left(\sum (f(k))^2 - 5000\right) \quad \textbf{(16)}$$

The third test used is runs test. If we describe a run as the maximal sequence of consecutive bits of the same kind then the incidence of runs for both consecutive zeros and consecutive ones of all lengths of what between 1 and 6 in the sample stream should be in the corresponding interval as specified in Table 1.

It is worth noticing the fact that the sequences longer than 6 are considered of length 6 when counting them. The test is passed if for the generated sequence the number of consecutive bits of each length is between the limits given in the Table 2. The long run test is passed if there are no runs of length 34 or more.

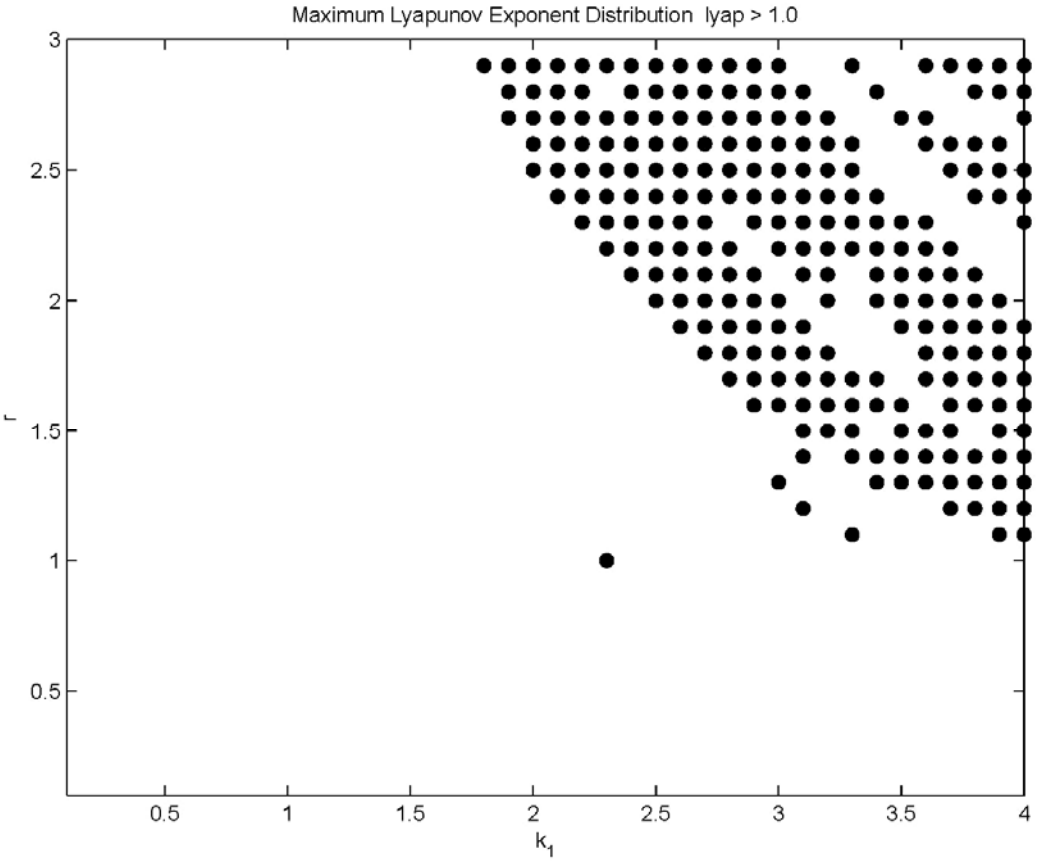

Figure 4. Largest Lyapunov Exponent of ZCDPLL - dot when it is greater than 1.0.

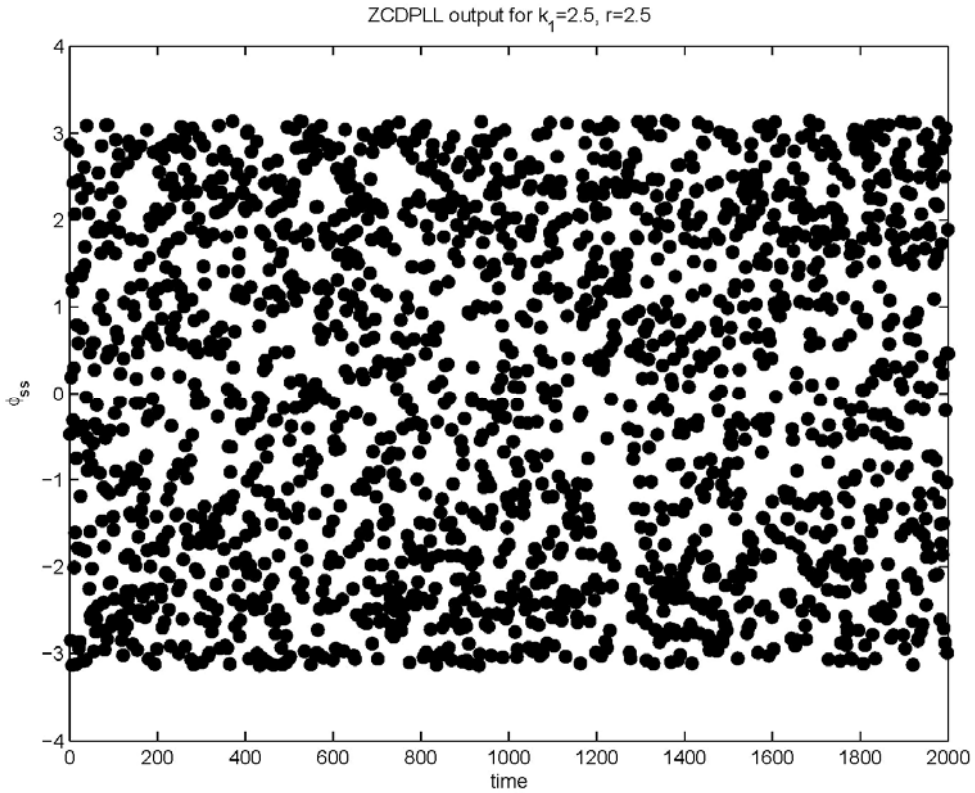

Figure 5. Distribution of ZCDPLL output (Sample 1) for a set of filter gain.

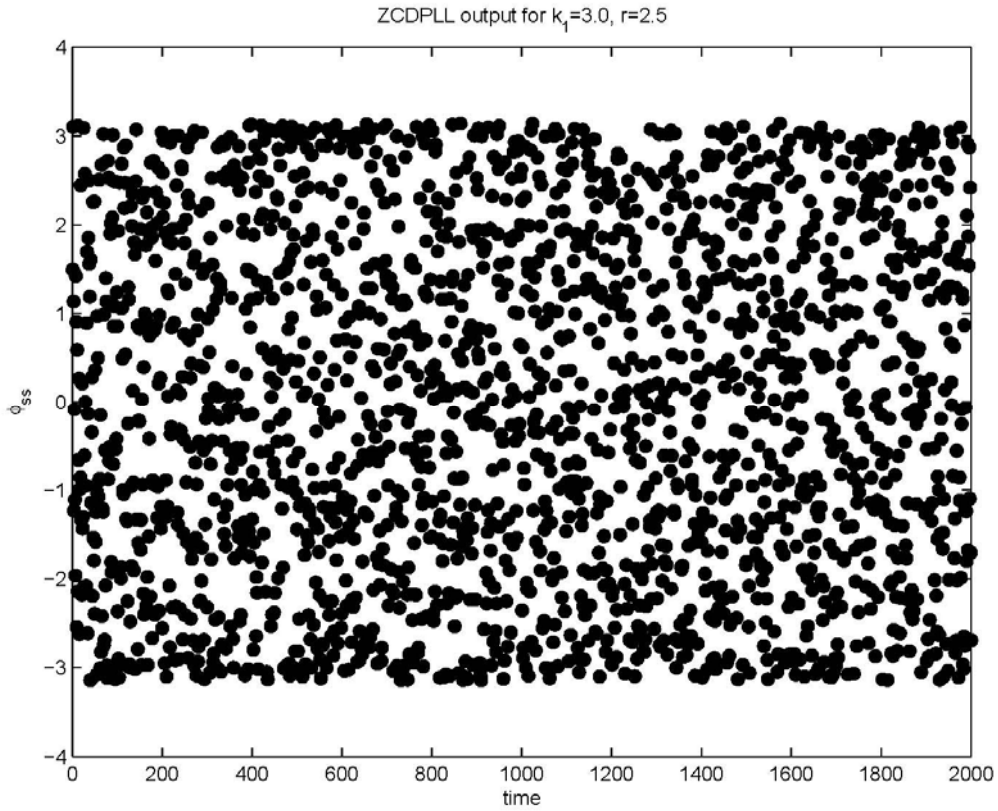

Figure 6. Distribution of ZCDPLL output (Sample 2) for other set of filter gain.

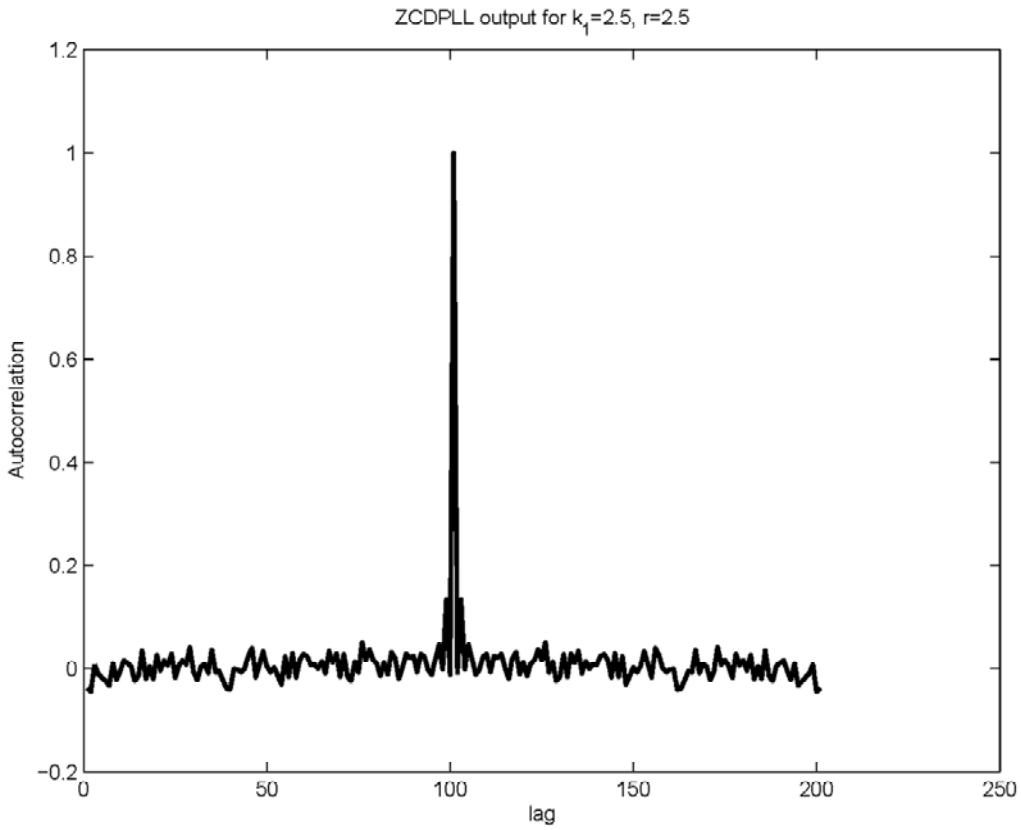

Figure 7. Autocorrelation of ZCDPLL mm output (Sample 1).

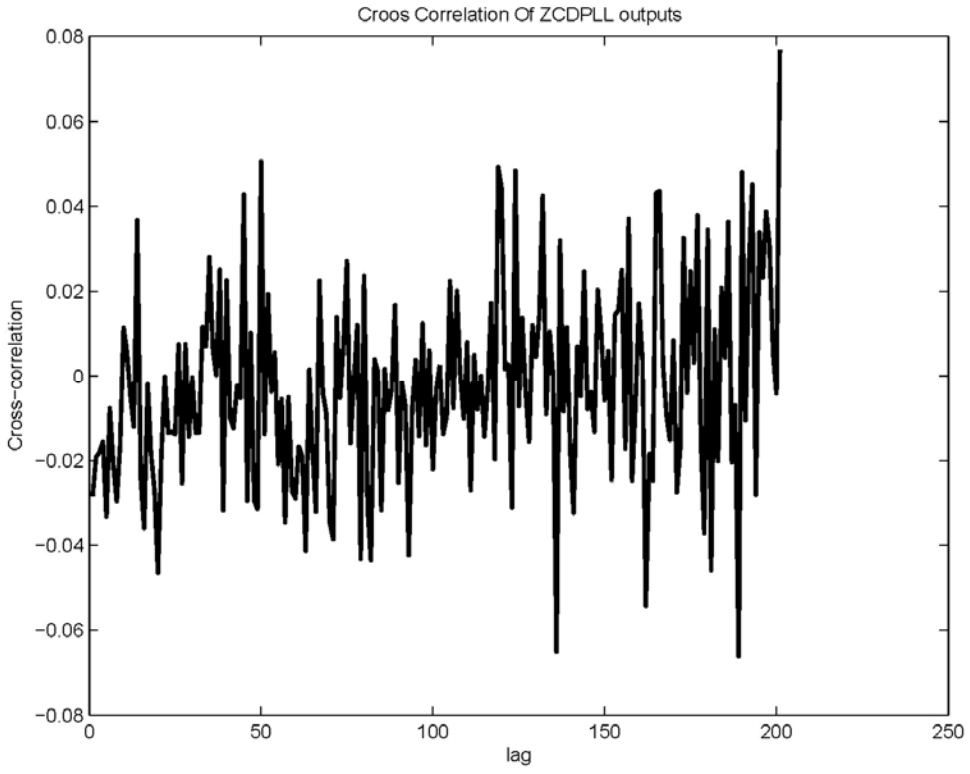

Figure 8. Cross Correlation of two ZCDPLL outputs (Sample 1 and Sample 2).

Table 1. Number of occurrences versus length.

Length	1	2	3	4	5	6+
Number Of occurrences	2267 / 2733	1079 / 1421	502 / 748	223 / 402	90 / 223	90 / 223

Table 2. Simulation test results.

AIS 31 Statistical Test	Result	Limits
Test 1 (Monobit Test)	Monobit = 9986	9645 < Monobit < 10346
Test 2 (Poker Test)	X = 44.498	1.03 < X < 57.4
Test 3 (Runs Test)	All Passed	
Test 4 (Long Run Test)	Long Run = 16	Long Run = 34

The proposed true random bit generator successfully passes all four statistical tests for every run. Table 2 shows the results of AIS 31 Standard Statistical Tests ran over 1 million random bits generated by ZCDPLL with K_1=2.5, r=2.5. As can be seen, all results are within the accepted range of the tests.

5. TMS320C6416 Implementation

The TRBG using ZCDPLL has been implemented in the software code targeted at a Texas Instruments TMS 320C6416 DSP. The generated bits are collected by the host PC. The key issue in the realization of the TRBG using ZCDPLL is the implementation of variable sample rate signal processing. In the realization based of DSP, variable sampling rate can be efficiently implemented using the DSP chip timer. At the beginning of the software program, the DSP timer (e.g Timer1) is set to be equal to maximum value of the sampling period. While the timer counts towards zero, an input (error) sample is read from the Analogue to Digital converter (ADC). On the basis of the error sample, the controller output and the actual value of the sampling period T are computed. Then the computed value of the sampling period is used to set the timer period. The DSP processor will enter an idle state till the timer expired, then the processed will be interrupted and the Interrupt Service Routine (ISR) will be called and the program loop repeats (see Figure 9).

Real-Time Data Exchange (RTDX) is used to provide real time, continuous visibility into the way TRBG software application operates in TMS320C6416. RTDX allows transfer the random bits generated in the DSP to a host PC for testing. On the host platform, an RTDX host library operates in conjunction with Code Composer Studio. In RTDX an output channel should be configured within ZCDPLL software. The generated data from ZCDPLL is written to the output channel. This data is immediately recorded into a C6416 DSP buffer defined in the RTDX C6416 library. The data from this buffer is then sent to the host PC through the JTAG interface. The RTDX host library receives this data from the JTAG interface and records it into either a memory buffer for testing purposes.

One million bits have been collected by the host PC. The same AIS 31 test suit has been used to check the truly randomness of the generated bits by the DSP kit. The results of these tests are shown in Table 3. It can be seen that the bits passed all the three proposed tests.

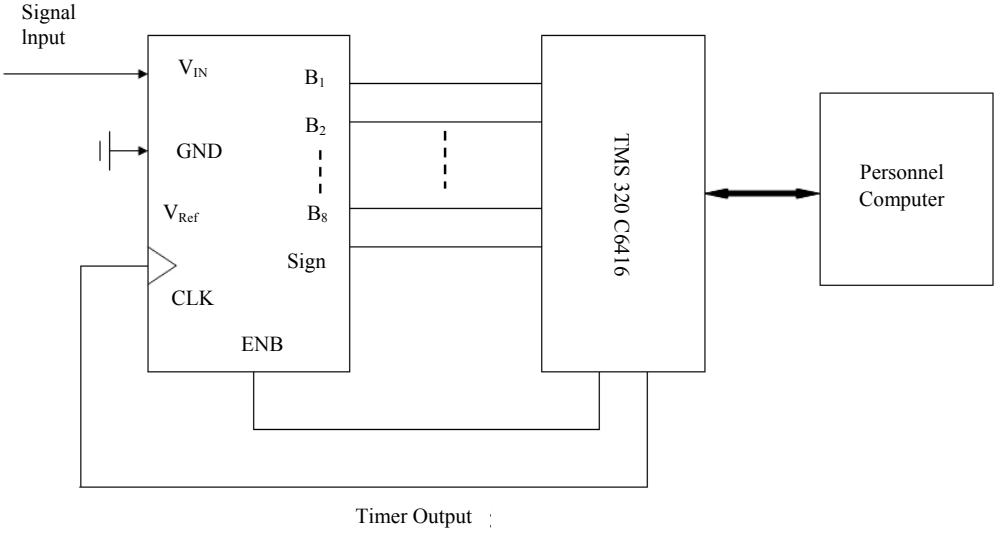

Figure 9. TMS320C6416 based TRBG.

Table 3. DSP card based test results.

AIS 31 Statistical Test	Result	Limits
Test 1 (Monobit Test)	Monobit = 10122	9645 < Monobit < 10346
Test 2 (Poker Test)	X = 51.32	1.03 < X < 57.4
Test 3 (Runs Test)	All Passed	
Test 4 (Long Run Test)	Long Run = 28	Long Run = 34

6. Conclusions

In this paper, True Random Binary Generator (TRBG) using second order ZCDPLL has been described and evaluated. The chaotic phase error produced by the ZCDPLL has been used to generate TRB. The proposed TRBG is subjected to statistical test suit AIS 31. The proposed TRBG successfully passed the tests described by AIS31 document. Another essential result of this paper is that the proposed TRBG based on ZCDPLL is implemented fully by software based on TMS320C6416 DSP kit. TRBG synchronization is still a challenging task which will be dealt within future work. Post processing is used to improve the statistical properties of the bit sequences generated by the ZCDPLL.

7. References

[1] T. Kohda and A. Tsuneda., "Chaotic bit sequences for stream cipher cryptography and their correlation functions," Proceedings of SPIE's International Symposium on Information, Communications and Computer Technology, Applications and Systems, Vol. 2612, pp. 86−97, 1995.

[2] M. Drutarovsk and P. Galajda, "Chaos based true random number generator embedded in a mixed-signal reconfigurable hardware," Journal of Electrical Engineering, Vol. 57, pp. 218−225, April 2006.

[3] J. C. Sprott, "Chaos and time-series analysis," Oxford University Press, 2003.

[4] N. Sajeeth, K. Philip, and J. Babu, "Chaos for stream cipher," http://uk.arxiv:org/PScache/cs/pdf/0102012v1.pdf.

[5] R. Matthews, "On the derivation of a chaotic encryption," Vol. 8, pp. 29−49, January 1989.

[6] N. Masuda and K. Aihara, "Cryptosystems with discretized chaotic maps,". IEEE Transactions on Circuits and Systems, Vol. 49, pp. 28−40, January 1999.

[7] S. Tao, W. Ruili, and Y. Yixun, "The theoretical design for a class of new chaotic feedback stream ciphers," Acta Eletronica Sinica, Vol. 27, pp. 47−50, July 1999.

[8] W. Lindsay and C. M Chie, "A survey of digital phase locked loops," IEEE Proceedings, Vol. 69, pp. 410−431, April 1981.

[9] G. Hsieh and C. Hung, "Phase locked loops–A survey," IEEE Transactions on Industrial Electronics, Vol. 43, pp. 609−615, December 1996.

[10] Q. Nasir, "Chaotic behavior of first order zero crossing digital phase locked loop," 2004 IEEE Asia-Pacific Conference on Circuits and Systems, pp. 977−980, 2004.

[11] AIS 31, "Functionality classes and evaluation methodology for true (physical) random number generators ver 3.1," Bundesmat fur Sicherheir in der Information technik (BSI), Bon, Germany, September 2001.

[12] M. E. Yalcin, J. K. Suykens, and J. Vandewalle, "True random bit generation from a double scroll attractor," IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, Vol. 51, pp. 1395−1404, July 2004.

[13] Texas Instrument, TMS320C6416 DSP Starter Kit (DSK), http://focus.ti.com/docs/prod/folders/print/tms320c6416.html.

[14] H. C. Osborne, "Stability analysis of an nth power digital phase-locked loop–Part I: First-order DPLL," IEEE Transactions on Communications, Vol. 28, No. 8, pp. 1343−1354, 1980.

[15] D. Croker and J. Schiller, "Randomness recommendations for security," Request for Comments 1750, 1994. http://www.ietf.org/rfc/rfc1750.txt.

[16] M. Bucci, L. Germani, R. Luzzi, A. Trifiletti, and M. aranouovo, "A high-speed oscillator-based truly random number source for cryptographic applications on a smart card IC," IEEE Transactions on Computers, Vol. 52, pp. 403−409, April 2003.

[17] M. D. Restituto and A. R. V'azques, "Chaos-based random number generators–Part II: Practical realization," IEEE Proceedings, Vol. 90, pp. 747−767, May 2002.

[18] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, "Handbook of applied cryptography," CRC Press, 1997. http://www.cacr.math.uwaterloo.ca/hac/.

[19] J. Von Neumann, "Various techniques used in connection with random digits," Applied Math Series, Notes by G. E. Forsythe, in National Bureau of Standards, Vol. 12, pp. 36−38, 1951.

[20] Standards, Information Technology Security Evaluation Criteria (ITSEC), Provisional Harmonized Criteria, Version 1.2, 1991.

[21] CEM-99/045, Common Methodology for Information Technology Security Evaluation (CEM) version 1.0,. 1999.

