

FPGA Implementation of a Nakagami- m fading channel Simulator using Random Number Generator

Santhosh Kumar B, K. Ayyappa Swamy

Dept. of E&I Engineering, Sree Vidyanikethan Engineering College, Tirupati Andhra Pradesh, India
Corresponding author, e-mail: san448@gmail.com, ayyappa.kondru@gmail.com

Abstract

By realizing the concept of implementation of the radio propagation channels on simulators is the eminent intention of this work. Even though prototyping software simulators are effortless to design we prefer hardware simulators as they endeavors distinguishable quickness, compared to earlier. Prior to this work fading channels like Gaussian, Rayleigh and Rician fading channel simulators were executed on hardware (i.e. FPGA) The significant target of this attempt is to implement Nakagami- m fading channel on hardware as they preserved for unsubstantial consideration. While differentiating with other fading channels Nakagami- m fading channel acquires wide spread applications, as its capability of providing appropriate fit for practical fading data, as this is flexible through its size parameter 'm', over which representation of signal fading conditions that vary from serious to modest, to normal fading or no fading is accomplished. The proposed hardware simulator can hatches complex Nakagami variates after it was implemented on XilinxSpartan-xc3s500e-4fg320 FPGA. This is operated at 50MHz.

Keywords: Nakagami- m , Fading, Quickness, Modest, Representation, FPGA

Copyright © 2016 Institute of Advanced Engineering and Science. All rights reserved.

1. Introduction

Wireless communication systems should be constructed to operate over different radio channels for a wide variety of environmental and weather conditions. Such a system essentially does well in divergent surroundings where the features of radio propagation channel changes significantly [1-3]. But due to its time varying property, multiple path propagation and existence of interference affected by other users, makes wireless communications more fundamental and deceptive. It is extremely necessary to perceive the performance of radio channels in different environments to attain the prototype of adequate, consistent and efficient wireless communications [4]. Such a prototype designing of wireless communications statistical channel modeling plays a critical role; this is a primary step in wireless system design prototyping. The primary intention of channel modeling is to estimate the first and second order parameters of a fading channel. Prototype design of a channel simulator is one of the numerous confrontation challenges in the elaboration of wireless communications [4].

The wireless communication systems are required to be field tested to confirm that they fulfil the requirements. But this is expensive, time costing and more complicated to repeat. By employing a channel simulator above mentioned limitations of the wireless communication systems can be overwhelmed. Henceforth in the process of designing and developing an efficient and for exploring the performance of wireless systems, a channel simulator is a significant component [4]. The wireless fading channel simulation offers more cost-effective, time-efficient, and can yields repeatable and reproducible results efficiently compared to field tests. Though software simulators are there the major interest is to simulate on hardware as they endeavours distinguishable rapidity, compared to earlier [5].

2. Importance of FPGA Implementation

In order to prototyping the fading channel simulators a simple Field-programmable gate arrays (FPGAs) are an excellent platform. These are precise, acceptable and adequate to emulate fading channel simulators, as FPGAs have earlier been used as coprocessors to execute the signal processing algorithms of a channel simulator in a comprehensive manner [5]. For the effective evaluation of a digital system Field-programmable gate arrays (FPGAs) are

used as an appropriate prototyping platform [5]. The complete time for designing the prototype is less due to the user-programmability of FPGAs which simplifies the debugging and design characterization. Adding to this their low cost compared to custom integrated circuits, make the FPGA's more significant. FPGA supports fast prototyping of the wireless communication design, as they have formidable hardware platforms. FPGA provides many attractive merits in performance, power consumption and re-configurability [5].

2.1. Ease of using Nakagami- m Fading Channel

As discussed earlier statistical channels are extensively used in the design of wireless communication channels. As the multipath fading signals statistics are depicted in an efficient manner. Over them Nakagami- m fading channel is an effective distribution which can prototype different fading environments [6]. Compared to Rayleigh, Rician, or Log Normal distributions, flexibility and accuracy in matching of some experimental data was offered by this statistical model, henceforth Nakagami- m fading channel fit some urban radio multipath channel data effectively compared to other statistical fading channels mentioned above [7]. Rayleigh and the one-sided Gaussian distribution as special cases make this statistical fading channel more significant in the design of wireless communication systems. With the parameter ' m ' The Nakagami fading channel model can be reduced to one-sided Gaussian fading ($m = 0.5$) to non fading ($m = \infty$). If $m=1$ Nakagami- m fading channel depicts Rayleigh fading channel. Rician fading channel model also be illustrated by this fading channel model under certain constraints. As a result of its ability of prototyping different fading environments made the Nakagami- m fading model is one of the most versatile [6].

2.2. Nakagami- m Fading Channel Simulator Design Prototyping

An efficient hardware emulation of a Nakagami- m fading channel simulator is presented. The Nakagami variates are to be derived from correlated Rayleigh variates. In this process of transforming the correlated Rayleigh variates into Nakagami- m a chain of logarithmic and linear domain analysis along with piece-wise linear approximations were needed to implement the prerequisite nonlinear numerical functions efficiently [4]. And the correlated Rayleigh variates are derived from uniform random variates. The LFSR (Linear Feedback Shift Register) plays a significant role in the generation of uniform random variates. A uniform random generator plays a significant role in designing all the types of random number generators [8]. Box-Muller method and inverse transformation methods are the best and excellent methods for calibration between the generators. We used the combination of LFSR and Box-Muller method to generate Rayleigh variates.

3. Designing an Efficient Nakagami- m Fading Channel Simulator

An efficient hardware emulation of a Nakagami- m fading channel simulator is presented. The Nakagami variates are to be derived from correlated Rayleigh. In this process of transforming the correlated Rayleigh variate into Nakagami- m a chain of logarithmic and linear domain analysis along with piece-wise linear approximations were needed to implement the prerequisite nonlinear numerical functions efficiently [4]. And the correlated Rayleigh variates are derived from uniform random variate. The LFSR (Linear Feedback Shift Register) plays a significant role in the generation of uniform random variate. A uniform random generator plays a significant role in designing all the types of random number generators. Box-Muller method and inverse transformation methods are the best and excellent methods for calibration between the generators. We used the combination of LFSR and Box-Muller method to generate Rayleigh variate [8].

3.1. LFSR as a Uniform Random Number Generator

The linear feedback shift register (LFSR) is a shift register by using feedback; itself changes the state for each rising edge of the clock. The present input bit of LFSR is always a linear function of its previous state. The initiatory value of the LFSR is known as seed [8]. The LFSR hatches the random sequences of 1's and 0's, when the shift register is clocked before it was initiated with a seed value (any value except all zeros). LFSR can yield a series of random bits by picking up a well chosen feedback function which has a very long cycle. The uniform random variable produced by the LFSR can have the maximum length equal to $2^n - 1$ where n is

the length of shift register in LFSR. In this work we use (n=12bit) then the number of random variable = $4k = 4096$ numbers.

In order to opt the feedback polynomials with maximum period the table presented by Xilinx is considered. The feedback polynomial for 12-bit is $x^{12} + x^6 + x^4 + x + 1$. The Figure 1 represents the constructional design for a 12-bit LFSR. Here the LFSR design is established by the cascading simple D-flip-flops [8].

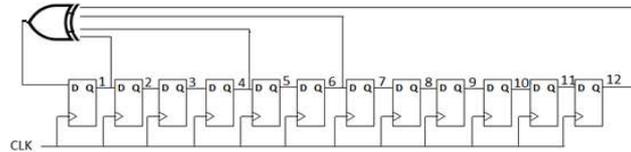


Figure 1. Architecture design for 12-bit LFSR as a Uniform Random number generator [8]

3.2. Mapping form Uniform RNG to Different RNG

In order to mapping between uniform RNG to any other random number generators in efficient manner Box-Muller method and inverse transformation methods are illustrated. In this work we used Inverse Transformation method to convert from the random number with uniform distribution to another type of distribution [9].

3.2.1. The Inverse Transformation Method (ITM)

The generation of non-uniform random variables using this inversion method is depend on the Simple observation that a random variable X with F as cumulative distribution function (CDF) can be generated by:

$$X = F^{-1}(U) \tag{1}$$

Where U denotes a uniform $U(0, 1)$ random variable and the X is a random variable with desired distribution [9]. The proposed architectures for Rayleigh distribution are based on this method.

The required steps to compute the desired random variable using inverse transformation method are:

1. Calculate the CDF of the desired random variable X .
2. Fix $F(X) = U$.
3. Determine the equation $F(X) = U$ for X in terms of U .
4. Generate uniform random numbers U_1, U_2, U_3, \dots and compute

The desired random variable by $X_i = F^{-1}(U_i)$

Applying the above steps for Rayleigh distribution results:

The PDF for Rayleigh distribution:

$$f(x) = \frac{x}{\sigma} e^{-\frac{x^2}{2\sigma^2}} \tag{2}$$

Then its CDF will be:

$$F(x) = 1 - e^{-\frac{x^2}{2\sigma^2}} \tag{3}$$

$$F(x) = U \tag{4}$$

$$x = \sigma\sqrt{-2\ln U} \tag{5}$$

3.3. Generation of Rayleigh Variates

It is mentioned in Figure 2 shown below the process of yielding of distinct models of random number by using LFSR's. Let the zero distributed and unit variance and normally

distributed duo of random variable as (X1, X2). the constructional design prototype is having a pair of 12-bit LFSR's to hatch the random variates with uniform distribution, now by using look up table (LUT) calculation and storing the values of X1 and X2 can be done. Now the LFSR yield 4096 random numbers .there are 4 RAMs in LUT such a way that each RAM can hold 1024 states (i.e. 1K X 16-bit) for each equation [8].

The value of sigma is the variance (σ^2) of the Rayleigh distribution. The non-restoring divider is used to divide the Value of X2 with the value of lambda (λ) to yield the exponential distribution.

The produced values of The Rayleigh distributed output is having magnitude (r) and is transformed to a complex number ($c_i + jc_q$) with two variates c_i and c_q along with zero mean and equal variance. The value of r_R^2 is nothing but the magnitude of a complex number.

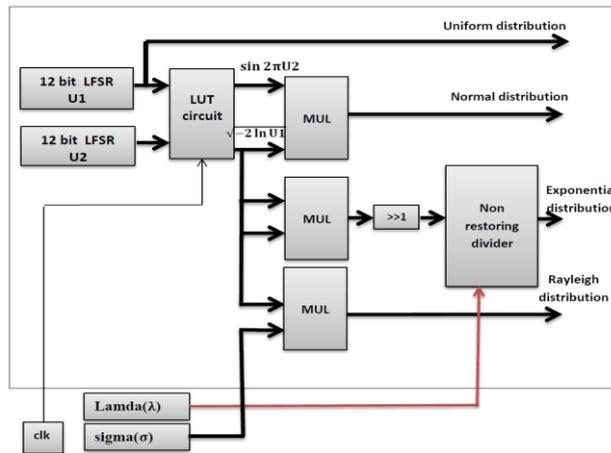


Figure 2. Architecture of distinct random number generators [8]

4. Generation of Nakagami Random Variates from the Reference of Rayleigh Variates

With the reference of an inverse cumulative distribution function and transferring a uniformly-distributed random variates $u \in (0,1)$, produces a RV having the CDF given by the function $F^{-1}(u)$ [4]. Similarly a uniform random variable u can be transformed into a Nakagami- m random variable n_N [4].

$$n_N = F_N^{-1}(u) \tag{6}$$

The sequence hatched from the Figure 2 of zero-mean unit-variance with the complex variates $c=c_i + c_j$ modulus $r_R=\sqrt{c_i^2 + c_j^2}$, here the values of c_i and c_j are independent and Rayleigh-distributed. By using the inverse CDF transformation and, a series of Rayleigh random variates can be converted to samples with a uniform distribution.

$$u=F_R(r_R)= 1 - e^{-\frac{r_R^2}{\sigma^2}} \tag{7}$$

Here the value of $F_R(.)$ is nothing but the CDF of Rayleigh RVs and σ^2 is the variance of the complex number c . Now the uniform RVs can be converted into correlated Nakagami- m RVs using Equation (7).

To develop the complex Nakagami- m fading variates x_X , the real-valued Nakagami-distributed random variables x_i and x_q were:

$$now \quad x_X = x_i + j x_q \tag{8}$$

The necessary thing to yield Nakagami- m RVs is determining the inverse Nakagami- m CDF($F_N^{-1}(u)$).

The compact generator of Nakagami-*m* variate generator is presented in the following Figure 3 [4].

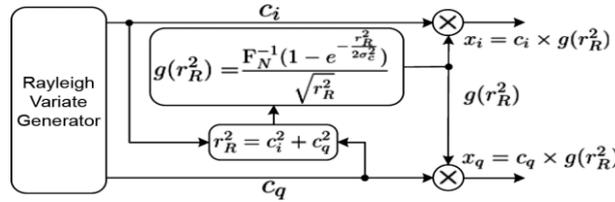


Figure 3. Process of deriving the Nakagami-*m* variate from a correlated Rayleigh variate generator [4]

While the proximate value of rational polynomial for $F_N^{-1}(u)$ is given as:

$$F_N^{-1}(U) \approx \eta(U) + \frac{a_1\eta(U)+a_2\eta(U)^2+a_3\eta(U)^3}{1+b_1\eta(U)+b_2\eta(U)^2} \tag{9}$$

Here,

$$\eta(U) = \sqrt[m]{\ln \frac{1}{1-U}} \tag{10}$$

In order to miniaturize the approximation error For a given value of *m*, the five coefficients *a*₁, *a*₂, *a*₃, *b*₁ and *b*₂ are calculated [10]. Form the compact and accurate Nakagami generator, the complex Nakagami variates yielded from Equation 8.

$$x_X = g(r_R^2) * (c_i + j c_q) \tag{11}$$

The value of the transfer function is:

$$g(r_R^2) = \frac{F_N^{-1}(1 - e^{-\frac{r_R^2}{2\sigma^2}})}{\sqrt{r_R^2}} \tag{12}$$

5. Simulated Results

In this section all the results are presented. First the result of 12-bit LFSR is presented i.e. the generation of uniform random variates. Next the results of Nakagami-*m* variates along with Rayleigh variates for different values of ‘*m*’ were presented and then the results of Hardware implementation of Nakagami-*m* variate generator along with design summary also were presented in this section.

5.1. Simulation Output of 12-bit LFSR

Figure 4 shows the output of 12-bit LFSR. It also consists the values of seed (initial value of LFSR) along with the LFSR output i.e. uniform random number generators.

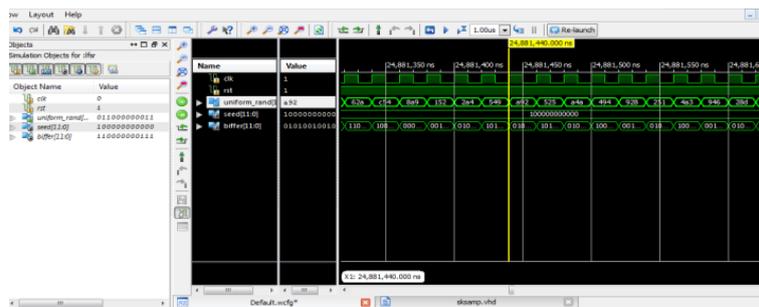


Figure 4. Output of 12-bit LFSR (i.e. uniform random number)

5.2. Simulation Outputs of Nakagami- m Variate Generator along with Rayleigh Variate Values

The following Figure 5 shows the generated Nakagami- m variates along with Rayleigh variates for the value of $m=0.5$.

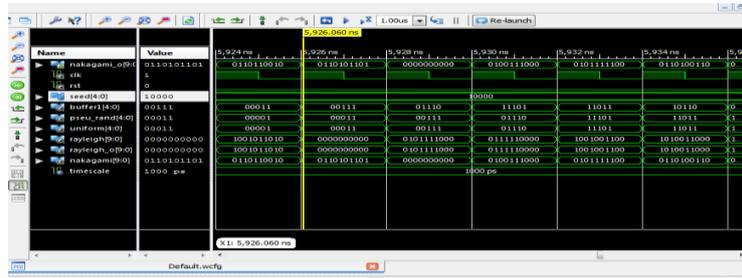


Figure 5. Generated Nakagami variates along with Rayleigh variates for $m=0.5$

The following Figure 6 shows Nakagami variates along with Rayleigh variates for $m=0.75$.

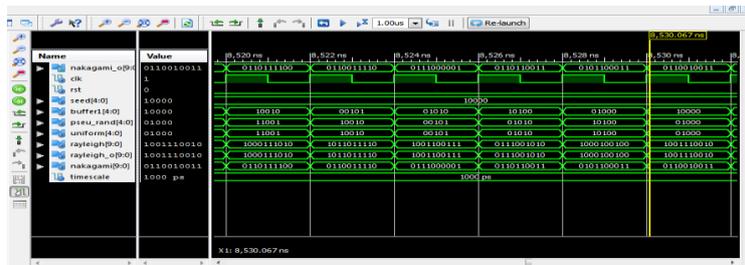


Figure 6. Generated Nakagami variates along with Rayleigh variates and uniform random variates for $m=0.75$

The following Figure 7 shows Nakagami- m variates along with Rayleigh variates for the value of $m=2$.

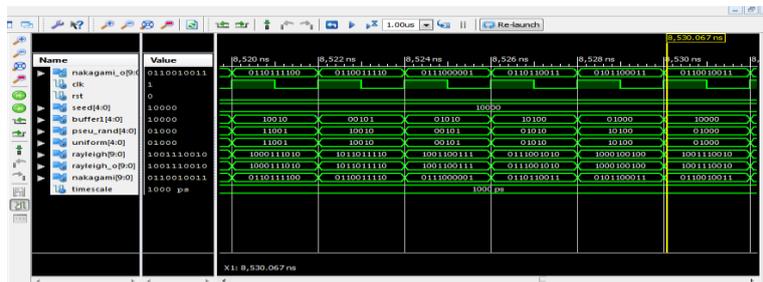


Figure 7. Generated Nakagami variates along with Rayleigh variates and uniform random variates for $m=2$

The following Figure 8 shows the generated Rayleigh variates along with Rayleigh variates for the value of $m=3$.

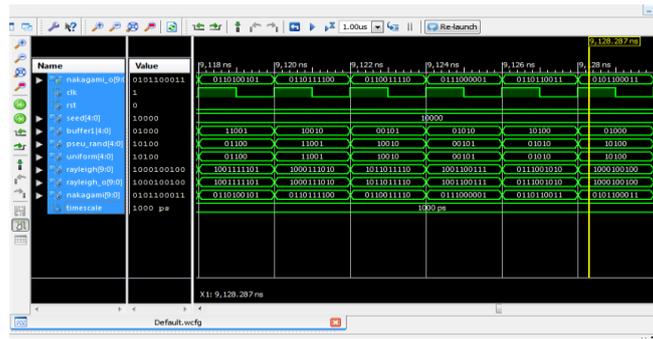


Figure 8. Generated Nakagami variates along with Rayleigh variates for $m = 3$

The following Figure 9 shows the Nakagami variates along with Rayleigh variates for $m=4$.

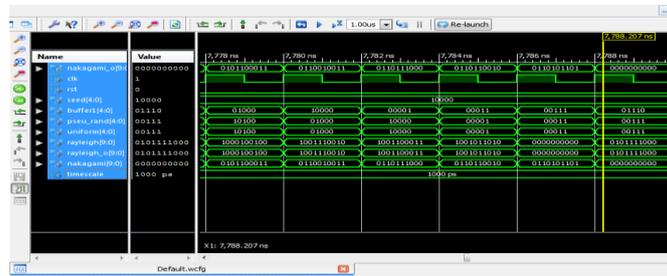


Figure 9. Generated Nakagami variates along with Rayleigh variates and uniform random variates for $m=4$

5.3. Comparison of Results between Software Simulation and Hardware Simulation

In this section the values yielded from the designed Nakagami- m variate generator are compared. The red spot in the graph indicates software simulated values and the blue spot indicates the hardware simulated values. For the value of $m=0.5$.

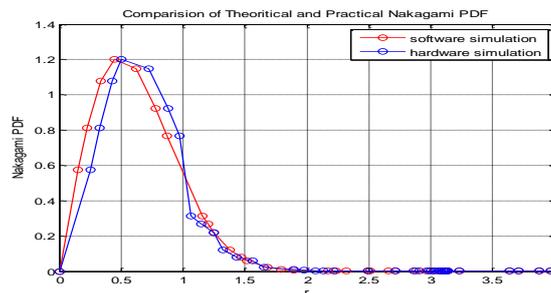


Figure 10. comparative results of hardware and software simulation for $m=0.5$

6. Conclusion

After the simulation process, the entire code is synthesized and program file is generated. The generated bit file is programmed into the FPGA SPARTAN 3E board. After programming, the results can be observed on board. As the clock frequency is in order of MHz, we cannot observe the variation on output LEDs of board. For observing hardware results in a GUI, we used Xilinx chip scope tool which captures the hardware results and displays on the

computer screen. For this purpose, ILA and ICON IP cores are added to the code [11-13]. The hardware results showing the Nakagami- m variates are shown in Figure 11.

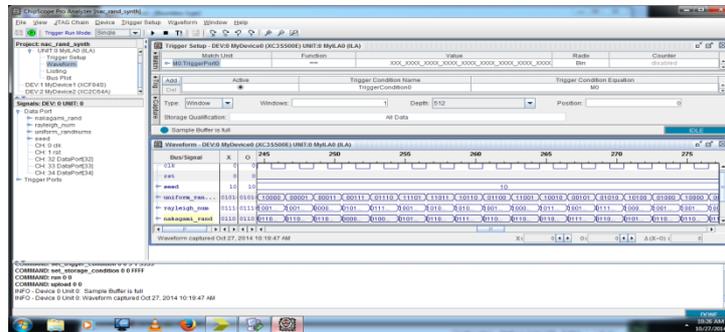


Figure11. Hardware (FPGA) Implementation of Nakagami- m variate generator

Acknowledgements

I would like to thank my guide Dr.Vinay Kumar, Assistant Professor, Dept. of ECE, JNT University college of Engineering (A), Kakinada, who gave the idea of this work and for his valuable suggestions and motivation. Also I would like to thank Mr. B. Pullaiah, Sr. Software Engineer, at Cognizant consultant services, Mumbai for his cooperation.

References

- [1] Zhou Shengkui. *Analysis and Simulation for Temporal Correlated Composite Fading Channels*. 3rd Asia-Pacific Conference on Antennas and Propagation. 2014.
- [2] Lei Shi, Zhen Guo, Yanming Liu, Xiaoping Li. A Novel Simulation Method for Nakagami- m Fading Channel, *IEEE*. 2014.
- [3] William S Sward, Dr Taylor Swanson, McKay Williams. *Scintillation Simulator Test Results Hardware-In-The-Loop Emulation of Ionospheric Scintillation*. 2014 IEEE Military Communications Conference. 2014.
- [4] Amirhossein Alimohammad, Saeed Fouladi Fard, Bruce F Cockburn. Hardware Implementation of Nakagami and Weibull Variate Generators. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*. 2012; 20(7).
- [5] Mark A Wickert, Jeff Papenfuss. Implementation of a Real-Time Frequency-Selective Channel Simulator Using a Hybrid DSP-FPGA Architecture. *IEEE Transactions on Microwave Theory and Techniques*. 2001; 49(8).
- [6] Andrea Goldsmith. *Wireless Communications*. Stanford University.
- [7] Marvin K Simon, Mohamed-Slim Alouini. *Digital Communication over Fading Channels*.
- [8] Mahmood. Architectural Design of Random Number Generators and Their Hardware. *Al-Rafidain Engineering*. 2014; 22(2).
- [9] G Box, M Muller. A Note on the Generation of Random Normal Deviates. *Annals Math. Statistics*. 1958; 29: 610-611.
- [10] Elina Pajala, Tero Isotalo, Abdelmonaem Lakhzouri, Elena Simona Lohan. *An improved simulation model for Nakagami-m fading channels for satellite positioning applications*. Proceedings of the 3rd Workshop on Positioning, Navigation and Communication (WPNC'06). 2006.
- [11] Efficient Shift Registers, LFSR Counters, and Long Pseudo-Random Sequence Generators. http://www.xilinx.com/support/documentation/application_notes/xapp_052.pdf.
- [12] David Bishop. Fixed point package user's guide. <http://www.vhdl.org/vhdl-200x/vhdl-200x-ft/packages/files.html>, 2006.
- [13] Xilinx. Spartan-3E FPGA Starter Kit Board User Guide UG230 (v1.2). 2011.