VHDL Execution for a Virtual Random Integer Producer Created on Camp Chart

Sherzad, Stefan
PG Students, Department of VLSI
Babylon University, Iraq

Abstract

Virtual random amounts are used for numerous suggests everywhere consistency data is favorite. For producing them, there are some valuable approaches, response shift records and dynamical meanings are most shared approaches. Among dynamical meanings, one dimensional disordered map is the humblest ones. The tent map is a piecewise-linear one dimensional map that has been used to produce virtual random amounts. Its accurate explanation formerly was surrounded from zero to one, so a modification was desirable for protective to digital application. After regulating has remained complete, some dynamical testes were practical to confirm if regulating did not disturb its operative. The following phase was to contrivance the map using VHDL; such execution was replicated and approved using a Mat lab cursive. Brief, it was attained a technique for informal generation of virtual random amounts.

I. INTRODUCTION

The virtual random amounts are used for dissimilar procedure. Cryptography asset trusts on amount arbitrariness. In totaling, there are other procedures where haphazardness is extremely preferred; stimulating and essential algorithm is decent example, as any procedure where information necessity is unchanging either. Virtual random amounts can be produced by numerous approaches, but two greatest extensively used are response shift catalogs and dynamical system applications. The FSRs are applied by one or a set of shifting registers. If they retain a linear situation comment, then they are named linear FSRs. Then, if their Feedback is non-linear; they will be non-linear FSRs; though they are not restricted to have only one kind of response, and multiple constructions appear. The dynamical system applications are extra complex than FSRs for organization, because it must comprise each dynamical system existed. One part of dynamical systems is n-dimensional maps. The maps describe a dynamical behavior, that can be chaotic or not. Maps have been extensively studied, owing to there are maps with just one variable, also known as one-dimensional map; thus, they are an easy form to produce chaos, and as consequence, randomness.

Even though they are called one-dimensional map, they do not exactingly depend on only one variable; they have a feedback control parameter generally designated as μ. This parameter controls map performance, transforming a comparative constant map into an extremely dynamical one. The maps can be categorized permitting to their mathematical explanation in polynomial and piecewise-linear. Commonly, endless function one-dimensional maps are categorized as polynomial; on the other hand, when its function is separated in two or more linear functions, the map is called piece-wise linear. Possibly the most famous polynomial one dimensional map is the Logistic one, essentially because it is one of the oldest maps that verified to produce chaos in spite of its minimalism; however, there are numerous extra maps. The Tent map is a good sample for piecewise linear map and one of the first established maps. The tent map will be this work focus.

II. MATHEMATICAL DESCRIPTION AND ITS DIGITAL ADAPTION FOR VHDL USAGE

Digital Adaption for VHDL Usage Permitting to Hauptmann, Tent map calculated depiction is bordered from 1 to 0, and it has a feedback factor (μ), what is in charge of control its dynamical performance. The thorough mathematical depiction performs on next in Eq. 1. In figure 1 performs how digital Tent map performs, numerous μ values were taken and it was executed by a 32 bit system. In assessment with regular Tent map, improved one is quite similar except for its limits.

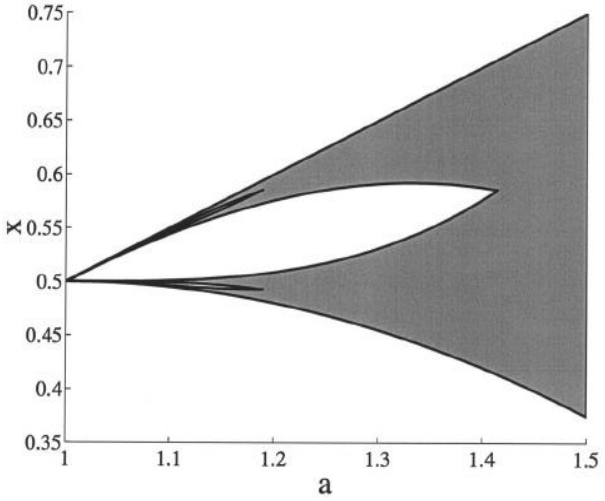


Fig 1 Bifurcation diagram for Tent map

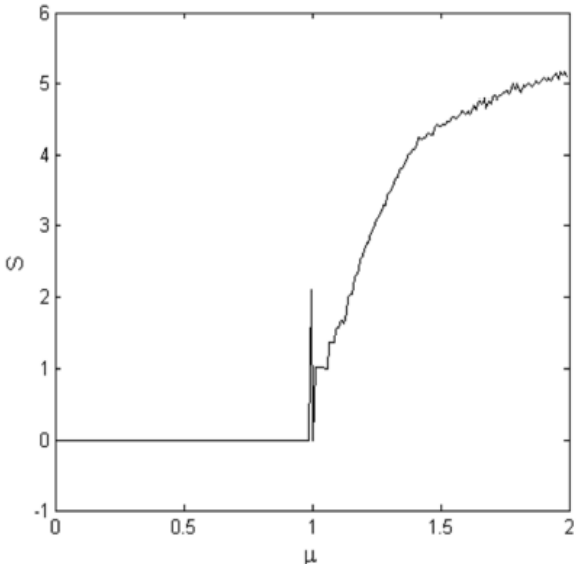


Fig 2 Entropy diagram for Tent map

A. *Dynamical Behavior of Digitalized Tent Map*

In order to confirm its dynamical performance, two traditional tools were used, Bifurcation and Entropy plans. Bifurcation plan procedures range where conceivable outputs can be located. And Entropy plan determinates actual unpredictability into an assured system for a specific μ. using both diagrams shown at figure; it is possible to attain a good landscape of how random a system is. There are concordances amongst both diagrams. First one, both figures show that Tent map surprises its chaotic performance from μ=1 and tops on μ=2. Other concordance, the advanced μ is the spreader output performs to be.

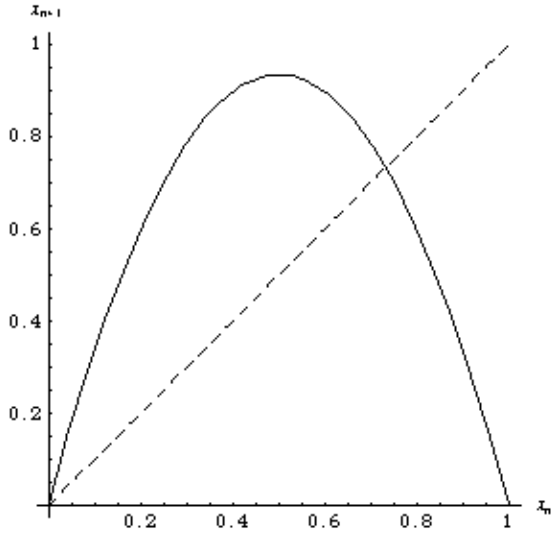


Fig 2 Behavior of Tent map for different feedback factors using

B. *VHDL Implementation*

After dynamical confirmation, the map was executed using a hardware explanation language as VHDL. For its execution, the system was divided in order to achieve a stronger view of its function. The figure 3 illustrates blocks that take portion of proposed system Origin chooser. It selects what data is familiarized to the system, it is significant in order to use the primary value for actual first time; and thereon, last intended value. - 2bits-xn and Selector MSB. Tent map has two conceivable outputs conferring to conditions presented at Eq. 2. One of possible output is μ (2bits-xn), and extra is μxn. Both opportunities have as common aspect μ; in significance, it is conceivable to abstract it and create multiplication later, to decrease difficulty.

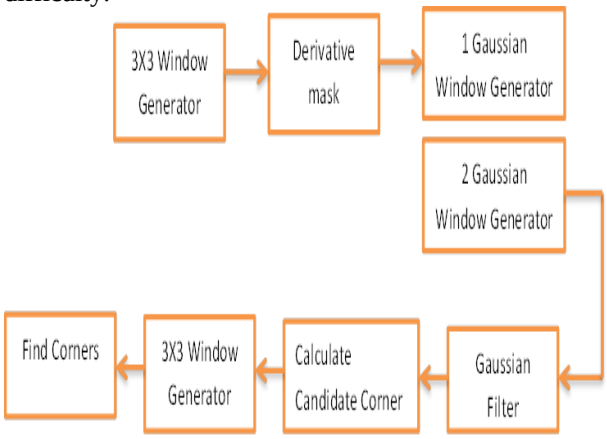


Fig 3 Block diagram for VHDL implementation

The condition is evaluated using most significant bit (MSB) from xn, so Selector only let pass correct output. - Multi xμ and Output register. Here the Selector MSB output is multiplied to complete mathematical description. When two numbers of m and n bits are multiplied, their result is an m*n bits number; thus, eight least significant bits are thrown away and 32 most significant are collected and stored by output register. Output register does data available to be used for another system. Summarizing, proposed implementation has two inputs, one of 32 bits as initial value, and another one of 8 bits, which works as μ value. Therefore, system output is a 32 bit word.

1) *Power consumption*

Table shows the dynamic power ingesting for the chaotic producers executed on both the Xilinx Virtex 6 and Altera Cyclone III FPGAs, with and deprived of the use of DSP masses.

As predictable, on both FPGAs, the dynamic power ingesting is reduced when DSP blocks are essentially used. Captivatingly, in terms of dynamic power ingesting, the Chebychev and Tent chaotic producers do not constantly rank in the similar order on the Altera and Xilinx FPGAs. While on Xilinx they always rank as second and third, on Altera they regularly rank as third and second individually.

2) *Resource usage*

In terms of FPGA resource usage, the amount of resources required by the implemented generators show different ranking on both FPGAs for different word lengths. Whether DSP blocks can be used or not also influences the ranking with the exception of the Cubic generator that is always last. As shown in Table III, when DSP blocks are allowed on the Xilinx FPGA, the Bernoulli and Chebychev generators rank first by requiring the same resources at all word lengths while Tent ranks third. However, when DSP blocks are not used, the Bernoulli generator clearly requires fewer resources followed by Chebychev and Tent.

Also shown in Table III, on the Altera FPGA, the Bernoulli producer still positions first. Otherwise, results are comparable to those for the Xilinx FPGA with the exemption of the Tent producer using fewer possessions than the Chebychev producer when DSP blocks are restricted. Captivatingly, the Altera Quartus II synthesizer was able to conclude a sum of removed courses from the continuous development in the Bernoulli generator, eliminating an expensive multiplier. It was also the case for the 16 bit Tent generator.

3) *Clock frequency*

Lastly, as exposed in Board IV, for the Altera Cyclone III, the Bernoulli producer is also the fastest with a maximum clock frequency reaching from 8% to 155% developed than its neighboring opponent, the Chebychev generator. On the Xilinx Virtex 6, results are comparable with the Bernoulli map performance having a higher supreme clock amount than the Chebychev generator extending from 0% to 124%.

Notice that the Cubic chaotic generator enterprise was not continually able to meet the smallest directed clock frequency of 100 MHz for the Virtex 6 and 48 MHz for the Cyclone III. Though not shown here, the targeted frequency could be met by changing the design into a pipelined design with a primary potential of 3 clock sequences at the overhead of greater resource usage.

All in the entire Bernoulli chaotic producer is obviously the most energy efficient between the compared generators with its low resource usage, high extreme execution frequency and low dynamic power consumption. However, by visual inspection of equation, the generator based on the Tent map was expected to come second on all aspects and it is not the case.

4) *Gold number generator*

As a position, on the Xilinx Virtex 6, a Gold number producer with R = 6 necessitates 6 lookup tables, can be implemented at the theoretical extreme clock incidence of 800 MHz and consumes an assessed 2.22 mW of dynamic power. On the Altera Cyclone III, the same Gold amount generator inhabits 12 logic features, can be implemented with a greatest clock frequency of 621.5 MHz and consumes an estimated 40μW.

III PROPOSED SYSTEMS

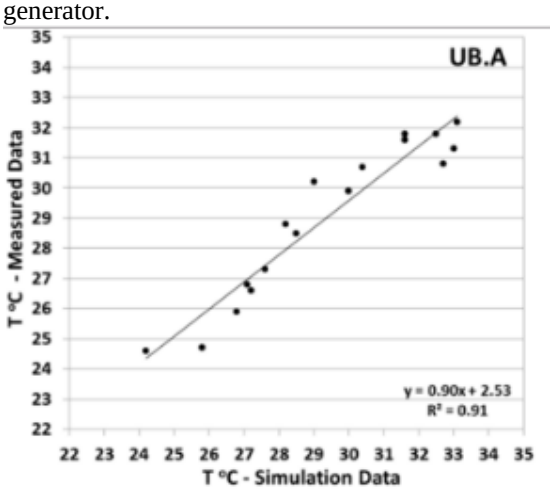
The VHDL code was replicated using University program VWF from Quartus II Suite. The first simulation was using a 4 bits representation, basically for quick correcting. 4 bit system had an initial value of 6 and a μ=1.375.

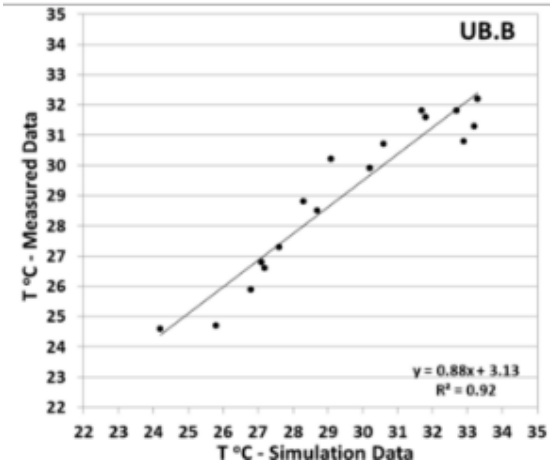
The achieved results were fulfilled, owing to concurrency among simulation results and intentions; so the next step was occupied, pretend using 32 bits for representation. The primary was set on AAAABBBB on hexadecimal what resemble to 2863315899 in decimal;

and μ=1.5, whose hexadecimal demonstration is C0. Figure shown simulations for 4 and 32 bit systems.

Basically because it had simple operations, 4 bit system intentions were handmade; nevertheless, for 32 bits things alteration and operations were more composite. In importance, a Mat lab script was intricate. In Table 1 appears a comparison between results obtained from Mat lab and simulation. The Altera synthesizer was set to use a balanced optimization technique and to put extra effort on power optimization. All characteristics were obtained from the implemented and routed designs with and without the use of digital signal processing blocks.

At both simulations of Figure 4, initial values are signal "a", signal "u" corresponds to μ value and signal "c" is system output. First both system output is 0; in other words, they do not have a valid output. Such behavior prevents malfunctions or possible misunderstandings in other systems that could be connected after the designed pseudo random number generator.





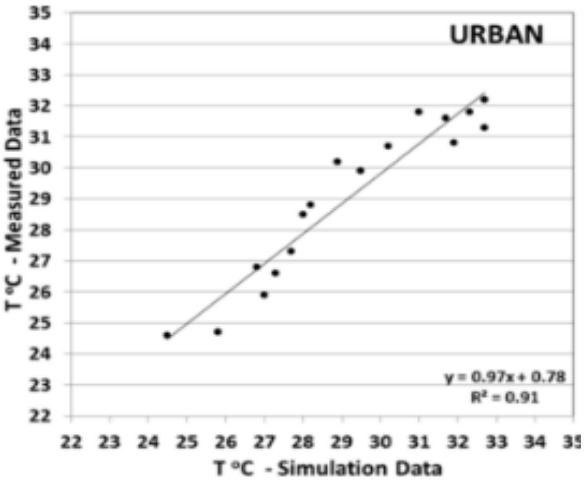


Fig 4 Simulation of Tent map implemented by a 4 (up) and 32 (down) bit representation system

Iteration	Matlab (hexadecimal)	Simulation (hexadecimal)
1	7FFFE667	7FFFE667
2	BFFFD99A	BFFFD99A
3	60003999	60003999
4	90005665	90005665
5	A7FF7E68	A7FF7E68
6	8400C264	8400C264
7	B9FEDC6A	B9FEDC6A
8	6901B561	6901B561
9	9D829011	9D829011

Table 1. Comparison between Mat lab and University program VWF

V. CONCLUSION

Tent map ascertained to be valuable to produce pseudo random numbers. Its mathematical description was adapted and familiar in order to simplify digital execution. Once it was attained, its description was verified through traditional tools for dynamical systems, as Bifurcation and Entropy tables. The system was intended, analyzed and separated, creating blocks that were used for its function depiction. Three blocks are portion of system: "Origin selector", "2bits-xn and Chooser MSB" and "Output register". Every block was executed using a hardware depiction language and simulated. Simulation results were associated with calculations, for 4 bit system calculations were handmade, and for 32 bit system calculations were achieved from a Mat lab script, completely made for such determination. Both system simulations were

consistent with their particular calculations; such authentication gave trust in a well done execution. Lastly, it is probable to accept that Tent map using a fixed-point image not only is conceivable but simple, next manufacture correct adjustments.

REFERENCE

[1] Susan Hohenberger, Venkata Koppula and Brent Waters, Adaptively Secure Puncturable Pseudorandom Functions in the Standard Model, (2014) IACR Cryptology ePrint Archive, Volume 2014, p 521.

[2] Dennis Hofheinz, Akshay Kamath Venkata Koppula and Brent Waters, Adaptively Secure Constrained Pseudorandom Functions , (2014) IACR Cryptology ePrint Archive, Volume 2014, p720.

[3] Suvajit Dutta, Tanumay Das, Sharad Jash, Debasish Patra and Pranam Paul, A Cryptography Algorithm Using the Operations of Genetic Algorithm & Pseudo Random Sequence Generating Functions, (2014) International Journal of Advances in Computer Science and Technology, Volume 3, No 5, pp 325-330.

[4] Ming Li and Dongdai Lin, A Class of FSRs and Their Adjacency Graphs, (2014) IACR Cryptology ePrint Archive, Volume 2014, p 658.

[5] Ricardo Francisco Martinez-Gonzalez and Jose Alejandro Diaz-Mendez, Implementation of a Stream Cipher Based on Bernoulli's Map, (2014) International Journal of Computer Science & Information Technology, Vol 6 No 6, pp. 113-121.

[6] Guang Zeng, Wenbao Han and Kaicheng He, High Efficiency Feedback Shift Register: sigma-LFSR, (2007) IACR Cryptology ePrint Archive, Volume 2007, p. 114.

[7] Navin Rajpal, Anil Kumar, Sureka Dudhani and Pravesh Raja Jindal, Copyright Protection Using Non-Linear Forward Feedback Shift Register and Error-correction Technique, (2004) 7th Annual International Conference Map India, New Delhi, India.

[8] Yongbin Zhao, Yupu Hu and Shunbo Li, A New Analysis Method for Nonlinear Component of Stream Cipher, (2013) Journal of Information & Computational Science, Vol 10 No 16, pp. 5313-5321.

[9] S. Ramahrishnan, B. Elakkiya, R. Geetha and P Vasuki, Image Encryption Using Chaotic Maps in Hybrid Domain, (2014) International Journal of Communication and Computer Technologies, Volume 2 No 13 Issue 5, pp 44-48.

[10] Chai Wah Wu and Nikolai F. Rulkov, Studying Chaos via 1-D Maps – A Tutorial, (1993) IEEE Transactions on Circuits and Systems: Fundamental, Theory and Applications, Vol 40 No 10, pp 707-721.