

A LOGISTIC MAP RUNGE KUTTA-4 SOLUTION FOR FPGA USING FIXED POINT REPRESENTATION

EMRE GUNGOR, ENVER CAVUS, AND IHSAN PEHLIVAN

ABSTRACT. Logistic map can show simple chaotic behavior pattern. Chaotic patterns with their irregularity and unpredictability properties are often used for random number generation and encryption. Hence, simple implementation of logistic map with chaotic behaviour is studied on how to solve differential equations with Runge Kutta Order-4 on FPGA and compared with floating point and fixed point representations. In the study, the logistic map equation is modelled with Verilog hardware description language and system is simulated using ModelSim and MATLAB environment. This paper reveals fundamental chaotic pattern implementation using logistic map to obtain unpredictable results on FPGA using fixed point number representation. Experimental results confirm both Verilog and MATLAB implementation produce same results. The results obtained from system in different generations can be converted to fixed point numeric values with an error margin. Error margin changes based on bit-length of fixed point number representation design.

Keywords: Logistic Map, Chaos, FPGA, Verilog

1. INTRODUCTION

Chaotic systems are a sub-type of nonlinear dynamical systems. Logistic map is a widely known tool for analyzing chaotic situations since it can create chaotic behavior that is seen at bifurcation diagram. Using these chaotic pattern, generalized logistic map system solution and chaotic field implementation becomes important for applications such as image encryption [1, 2].

Field Programmable Gate Arrays (FPGAs) provide a flexible and fast implementation technique for complex logical circuits. Nowadays, chaotic applications have been implemented using FPGAs; the chaotic pseudo random number generator on FPGA can be given as an example [3, 4]. Synthesized Lorenz chaotic system using FPGA can be given as another example [5]. Also chaotic equations are used in telecommunication for applications such as data stream encryption [6].

2. FIXED & FLOATING POINT LOGISTIC MAP REPRESENTATIONS

Logistic map term describes logistic growth of population under limited resources and environmental conditions [7]. Logistic Map variable definitions under these conditions are defined as follows;

- Using growth rate term: $r(1-x_n)$
- For small x_n growth rate: r
- For large x_n growth rate : 0

So for population estimation in logistic map from generation n to $n+1$, the equation becomes as:

$$(2.1) \quad X_{n+1} = r \cdot X_n \cdot (1 - X_n)$$

Logistic map, in chaos is an example of how complex behavior can arise from simple polynomial equations. With this example, modelling of more complex chaotic systems can be achieved with small changes.

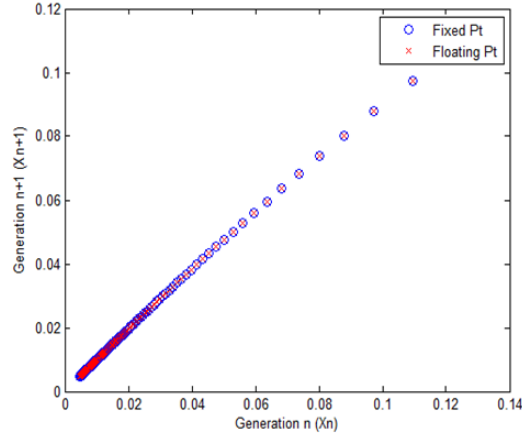


FIGURE 1. $r = 1$, logistic map solution for fixed and floating point.

The logistic map equation produces fixed results within prediction domain of growth as shown in Figure 1. However, if parameter r changes the equation behavior becomes unpredictable (chaotic state) which can be seen in bifurcation diagram. Initial condition of x value, x_0 taken as 0.125 in all logistic map equations for this study. Double bit-length representation of initial input values can easily be converted in binary format using floating and fixed point binary number representations.

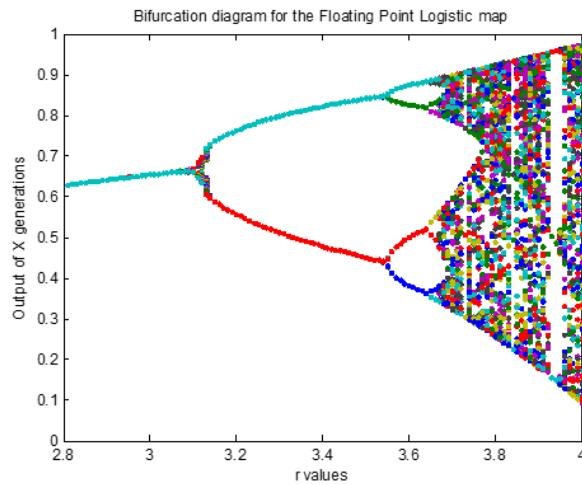


FIGURE 2. A logistic map bifurcation diagram.

When r value reaches to near 3.8 in logistic map, the outputs become unpredictable as it can be seen in Figure 2, so in that area output values becomes chaotic as shown in Figure 3.

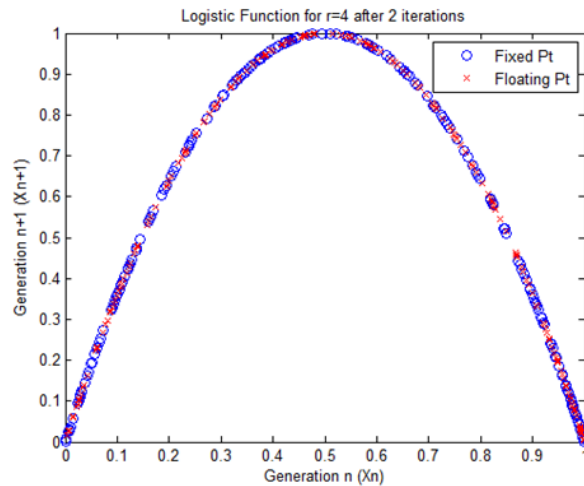


FIGURE 3. $r = 4$, logistic map solution for fixed and floating point.

A chaotic structure exists in Figure 2 and Figure 3, however, fixed and floating point format outputs becomes different because of fractional storage capability which creates errors margin; hence, choosing number representation and error magnitude is important when constructing a digital circuit for chaotic attractors. In Figure 4, generation changes can be seen and this type of change as $r = 3.4$ is the same in both floating and fixed point number representations.

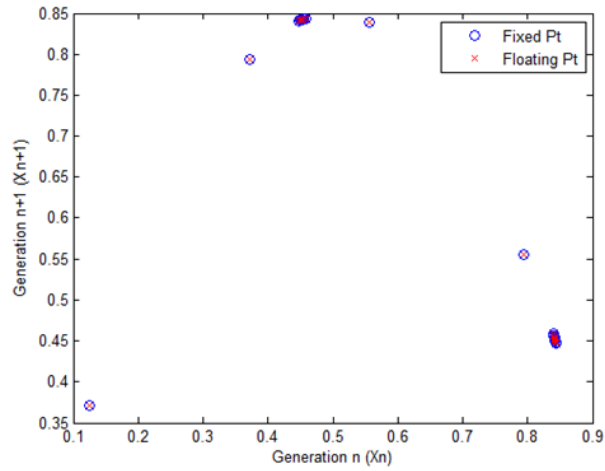


FIGURE 4. $r = 3.4$, logistic map X_n vs X_{n+1} .

In Figure 5, 2-cycle outputs can be seen as in bifurcation diagram. This yields predictable outcome and this r values cannot be used in chaotic applications so the design only should include chaotic fields where $r > 3.75$ that chaotic behavior can be seen. In this study a model is coded using Verilog, and simulated using ModelSim-XE using parameter $r = 4$. In modelling using Verilog without any predefined structures, different sources are studied to understand concepts of FPGA design with their applications [8,9].

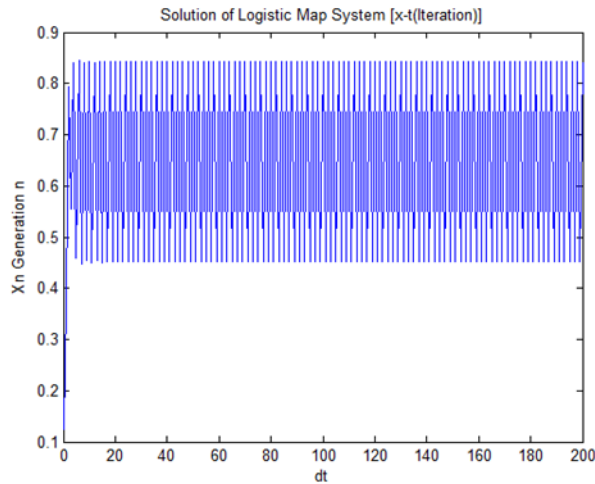


FIGURE 5. $r = 3.4$, logistic map X_n vs dt graph.

3. DIFFERENTIAL SOLUTION

3.1. Runge Kutta Order-4. Runge-Kutta methods are important in numerical analysis as typical differential equations. In this study, Runge-Kutta Order-4 differentiation method is used for more accurate approximation. This approach required a lot of space than a simpler method to implement. However, if more accurate solutions are needed, then this method produces better outputs for chaotic system implementations. The logistic map is a test case for system usage on chaotic applications. More simple differential methods can also be used for comparison purposes, however in this case error rate increases. Differential equations are important solution methods for solving different chaotic attractors to obtain accurate results. As a result, Runge Kutta order 4 (RK-4) is widely used for different chaotic attractors. If less accuracy is needed simpler method can be used as differential solution. RK-4 method can be implemented using the following equations:

$$(3.1) \quad y' = f(t, y) \quad y(t_0) = y_0$$

$$(3.2) \quad y_{n+1} = y_n + \frac{1}{6} \cdot (k_1 + 2k_2 + 2k_3 + k_4)$$

$$(3.3) \quad t_{(n+1)} = t_n + h$$

In logistic map, time variable t is shown in Equation (3.3) indicates that each generation is going to be re-calculated. The symbol h shows the time change normally, and generation in logistic map case. The k values are calculated using the Equation (3.4), and then placed into Equation (3.2).

$$(3.4) \quad k_1 = hf(t_n, y_n)$$

$$(3.5) \quad \begin{aligned} k_1 &= hf(t_n, y_n) \\ k_2 &= hf(t_n + 1/2h, y_n + 1/2k_1) \\ k_3 &= hf(t_n + 1/2h, y_n + 1/2k_2) \\ k_4 &= hf(t_n + h, y_n + k_3) \end{aligned}$$

3.2. RK-4 Implementation in Matlab and ModelSim. In MATLAB, RK-4 method is derived from numerical integration equations manifesto [10, 11]. The MATLAB solutions are taken in the form of floating point number representations since no quantization is needed. However, on fixed point implementation RK-4 method quantization is necessary at all calculation steps. RK-4 algorithm is used for differential solution since chaos is sensitive to each iteration on generations. FPGA synthesis stage complexity and simplicity should be primarily considered. If behavioral model is used during implementation, data range should be explicitly specified.

4. FIXED POINT IMPLEMENTATION

Fixed point implementation approach is selected at logical design. In fixed point approach, arithmetic operations are handled separately at different modules. During the design phase, basic logistic map is implemented in modular fashion for hiding the details at each individual modules. Rounding in fixed point format is implemented to obtain more accurate solution.

```

module lgstemp3(xwout,xin,r);
input[36:0] r;
input[36:0] xin;

wire[72:0] xtempx;
wire[72:0] xtempx2;

output[36:0] xwout;

//EQUATION:: x(i+1)=r*x(i)*(1-x(i));

assign xtempx=xin*(r);

assign xtempx2=(xtempx[68:32]+xtempx[31]) * ((1<<32)-xin);
assign xwout=xtempx2[68:32]+xtempx2[31];

endmodule

```

FIGURE 6. A part of Logistic map Verilog implementation.

Fixed point number representation is used at Verilog implementation, and IEEE-754 format is used on MATLAB simulations for floating point numbers to model logistic map function. A quantization function is also defined to match Verilog and MATLAB models.

Verilog implementation results of floating point logistic map model is shown in Figure 7. Table 1 depicts floating point and fixed point number outputs for each generation in Verilog model.

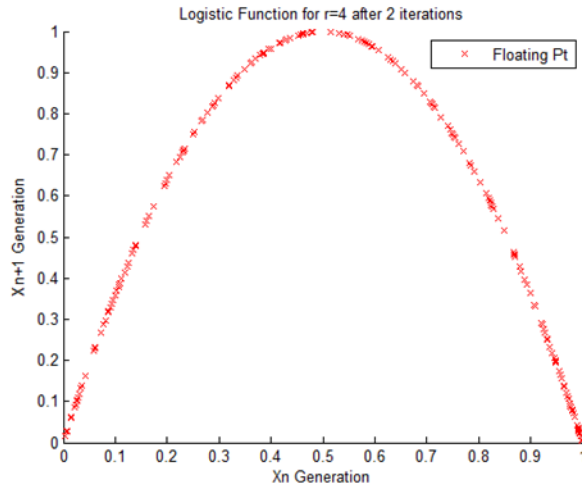


FIGURE 7. Logistic map Verilog floating point implementation results at different generations.

The solution does not need to be reconsidered for calculations in floating point format. However, in Verilog dynamically changing the length affects the calculations, especially in multiplication. For that reason, the data length needed to be reduced even some data loss occurs.

TABLE 1. Floating point to fixed point conversion

Floating Point Generation and Fixed Point Equivalents for 32 bit Scale Integers		
Generation	Floating Point	Fixed Point
1	0.1250	536870912
2	0.4375	1879048192
3	0.9844	4227858432
4	0.0615	264241152
5	0.2310	991936512

Fixed point logistic map model in Figure 6 creates some error on different formats. Integer and fraction parts on fixed point determine change on difference between floating and fixed model output value changes. Thus, scaling is important factor on error estimation and it needs to be modelled on this error constraints.

In lower r values in logistic map, usage of floating and fixed point representations yields same outputs, however chaotic states creates different outputs because of the accuracy changes. So system response become different for fixed point number representation model.

4.1. **Simulation on Matlab and Verilog codes.** Fixed point Verilog code behavior and floating point representation differs on chaotic levels of logistic map during simulation. The difference is shown in Figure 8 of these two representations.

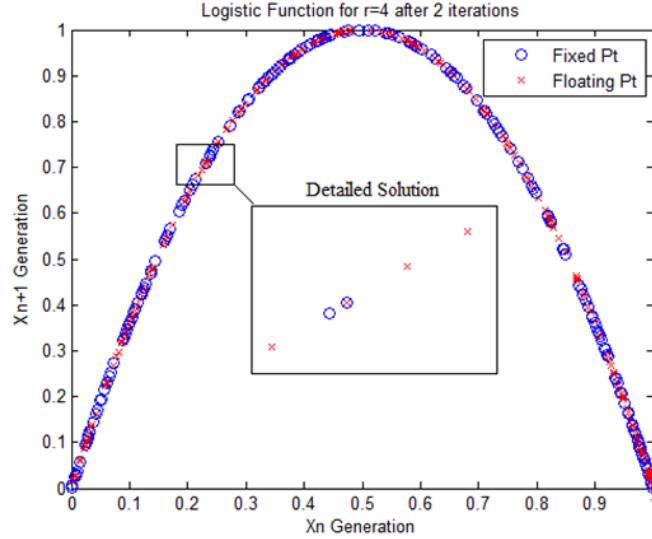


FIGURE 8. Logistic map floating & fixed number difference in chaotic state.

4.2. **Simulation on Matlab and Verilog codes.** Verilog modules for Logistic Map calculation consists calculation modules for sum, difference, multiplication and module for differential calculations. Finite state machine (FSM) is used to cover all possible input and register changes to calculate results.

In Verilog implementation, bit length taken as specific default value to calculate. In fixed numbering, fraction and integer parts are taken as total 36-bit and 1-bit sign extension to represent negative values. Taking different bit lengths to represent formats other than 16 and 32 bits is to study the system outputs and accuracy on different bit scale and effects on the system. Representation of number formats change system output. Hence error margin in design should also be taken into account when realizing the system on FPGA. In Figure 9, Verilog and Matlab implementation results can be seen in fixed point format. To compare system outputs, Kdiff3 software is used [12].

536870912	x
1879048192	1879048192
4227858432	4227858432
264241152	264241152
991936512	991936512
3051382336	3051382336
3534046170	3534046170
2504447839	2504447839
4176294976	4176294976
461573353	461573353
1647875175	1647875175
4062501148	4062501148
879535445	879535445
2797686842	2797686842
3901237459	3901237459
1430542757	1430542757
3816263543	3816263543
1701395661	1701395661
4109639233	4109639233
709324590	709324590
2368711440	2368711440

FIGURE 9. Numeric results of both Verilog (left) and MATLAB (right) simulations on Kdiff.

5. CONCLUSION

Verilog implementation of the Logistic Map simulation using fixed point number representation as 16-bit mantissa and 16-bit fraction part gives the same results as in computer simulation. Figure 9 shows MATLAB and Verilog simulation results of the Logistic Map. The first x value is not printed in simulation because it represents initial condition of x. Output values only displayed in Verilog simulation.

Sometimes different chaotic systems need to be implemented using variable bit lengths. In that case, parametric design becomes important. A chaotic application can easily be synthesized using an FPGA with fitting consideration. It is observed that changing the differential equation affects FPGA space and timings factors. Hence, usage factors of the chip and the error rate have larger impact on design.

This study can be extended using constraints and defining a parametric structure. After determining the design constraints, implementation of different chaotic applications are possible. Logistic map study is important for studying simple and chaotic structure. In this way more complex chaotic systems can be analyzed easily. This study shows initial steps of implementing chaotic applications on FPGA.

REFERENCES

- [1] N.K. Pareek, Vinod Patidar, K.K. Sud, Image encryption using chaotic logistic map Image and Vision Computing Vol. 24, Issue 9, pp. 926-934, 1 Sept. 2006.
- [2] Kocarev, Ljupo, and Goce Jakimoski. "Logistic map as a block encryption algorithm." Physics Letters A 289.4, pp. 199-206, 2001.
- [3] K.H. Tsoi, K.H. Leung , and P.H.W. Leong, Compact FPGA-based True and Pseudo Random Number Generators Field-Programmable Custom Computing Machines, , FCCM 2003, 11th Annual IEEE Symposium, pp. 51-61 ,9-11 April 2003.
- [4] Phatak, S. C., and S. Suresh Rao. "Logistic map: A possible random-number generator." Physical review E 51.4, 3670, 1995.
- [5] Aseeri M.A, Shobhy M.I., Lee P., Lorenz chaotic model using Filed Programmable Gate Array(FPGA), 45th Midwest Symposium on Circuits and Systems, MWSCAS-2002, vol. 1, pp. I-527-30, 4-7 Aug. 2002.
- [6] Azzaz M.S, Tanougast C., Sadoudi S., Dandache A., Real-time FPGA implementation of Lorenzs chaotic generator for ciphering telecommunications, Circuits and Systems and TAISA Conference, NEWCAS-TAUSA 09, pp. 1-4, 2009.
- [7] May, Robert M. "Simple mathematical models with very complicated dynamics." Nature 261.5560,pp 459-467, 1976.
- [8] Modern Automation Systems, Laxmi Publications, Ltd., 01-May-2009.
- [9] Granberg, T. , Handbook of digital techniques for high-speed design: design examples, signaling and memory technologies, fiber optics, modeling and simulation to ensure signal integrity., Prentice-Hal, 2004.
- [10] Hall C.D., Manifesto on Numerical Integration of Equations of Motion Using Matlab, 2002.
- [11] MATLAB, The MathWorks, Inc., Natick, Massachusetts, United States.
- [12] Eibl, J. "http://kdiff3.sourceforge.net/" . 2003.

COMPUTER ENGINEERING DEPARTMENT, ALANYA HAMDULLAH EMIN PASA UNIVERSITY
Email address: emre.gungor@ahep.edu.tr

ELECTRICAL AND ELECTRONICS ENGINEERING DEPARTMENT, ANKARA YLDRM BEYAZT UNIVERSITY

ELECTRICAL AND ELECTRONICS ENGINEERING DEPARTMENT, SAKARYA UNIVERSITY OF APPLIED SCIENCES