2008 International Symposium on Nonlinear Theory and its Applications
NOLTA'08, Budapest, Hungary, September 7-10, 2008

FPAA-Based Programmable Implementation of a Chaotic System Characterized with Different Nonlinear Functions

Recai KILIÇ[1] Fatma YILDIRIM DALKIRAN[2]

[1] Dept. of Electrical&Electronics Engineering, Erciyes University, 38039, Kayseri, TURKEY
[2] School of Civil Aviation, Erciyes University, 38039, Kayseri, TURKEY
E-mail: kilic@erciyes@edu.tr

Abstract–In addition to exhibiting a rich variety of bifurcation and chaos via tuning its parameters, a chaotic system introduced by Sprott can be modeled and realized with a fixed main system block and many different changeable nonlinear function blocks such as piecewise-linear function, cubic function and other trigonometric functions. This system is very suitable for implementing a programmable chaos generator according to its changeable nonlinearity. This paper presents FPAA-based programmable implementation of this system. Nonlinear function blocks used in this chaotic system are modeled with FPAA programming and a model is rapidly changed for realizing another nonlinear functions.

1. Introduction

FPAA [Field Programmable Analog Array] is a programmable IC for implementing a rich variety of systems including analog functions. Because this IC has programming feature that can be used to change component values and interconnections, it can be dynamically reconfigured. This meant that a design modification or a completely new design can be downloaded to an FPAA while it is operating in a system without the need to power down or to reset the system. In addition to these features, FPAA provides more efficient and economical solutions for analog dynamical system designs. Using FPAAs, various dynamical systems can be implemented at less cost, in a much smaller size and with increased reliability and component stability [1-3].

On the other hand, among the nonlinear dynamical systems, a chaotic system introduced by Sprott [4,5] has very high potential from point of view programmability and reconfigurability. In addition to exhibiting a rich variety of bifurcation and chaos via tuning its parameters, this system can be modeled and realized with a fixed main system block and many different changeable nonlinear function blocks such as piecewise-linear function, cubic function and other trigonometric functions. The implementation and experimentally investigation of this chaotic system which can be modeled with many different nonlinear functions require a significant variety amount of circuit hardware. FPAA can be effectively used instead of these discrete implementations. This programmable device is more efficient, simpler and economical than using individual op-amps, comparators, analog multipliers and other discrete components used for implementing nonlinear functions. Nonlinear function blocks used in this chaotic system can be modeled with FPAA programming and a model can be rapidly changed for realizing another nonlinear function. Using this design approach, it can easily be investigated the roles and effects of different nonlinear functions in chaotic system structure under a programmable realistic model. The organization of the paper is prepared as follows: In Section 2 the system definitions with different nonlinear functions are summarized. FPAA-based circuit implementations using different nonlinear functions for this model will be introduced in Section 3. Finally, some concluding remarks will be discussed in the last Section.

2. System Definitions

Chaotic system studied here is based on “Jerk” equations [4] and it is defined by the following ordinary differential equations:

$$\begin{aligned} \frac{du}{d\tau} &= v \\ \frac{dv}{d\tau} &= w \\ \frac{dw}{d\tau} &= f(u) - pw - v \end{aligned} \tag{1}$$

Here, while *u, v* and *w* denote the state variables of the system, *p* is the system parameter. *f(u)* represents a nonlinear function which plays important role in the system’s chaos mechanism. It has been verified that this chaotic system can be realized with many different nonlinear functions including different type piecewise-linear functions [4,5]. The most preferred functions in modeling of this system are summarized in Table-1 with the typical parametric definitions. The implementation and experimentally investigation of the generalized chaotic system defined by Eqn.(1) with different nonlinear functions require a significant variety amount of circuit hardware. In these implementations, several circuit topologies using Op-Amps, diodes and other passive components have been used for realizing the referred

Table 1. The most used nonlinear function definitions for modeling chaotic system defined by Eqn.(1).

Nonlinear Functions	Function Parameters
$f(u)=\|u\|-r$ (2)	r=2
$f(u)=-B\max(u,0)+C$ (3)	B=6, C=-0.5
$f(u)=-Bu+C\,\mathrm{sgn}(u)$ (4)	B=1.2, C=2
$f(u)=B(u^2/C-C)$ (5)	B=0.58, C=1
$f(u)=Bu(u^2/C-1)$ (6)	B=1.6, C=5
$f(u)=-Bu(u^2/C-1)$ (7)	B=0.9, C=0.47
$f(u)=-B[u-2\tanh(Cu)/C]$ (8)	B=2.15, C=1

piecewise-linear nonlinear functions. To electronically implement other trigonometric nonlinear functions such as $f(u)=-B[u-2\tanh(Cu)/C]$, it is required to use an IC device that operates as universal trigonometric function generator. FPAA can be effectively used instead of nearly all of discrete analog implementations. This programmable device is more efficient, simpler and economical than using individual op-amps, comparators, analog multipliers, trigonometric function generator and other discrete components. Nonlinear function blocks used in this generalized chaotic system can be modeled with FPAA programming and a model can be rapidly changed for realizing another nonlinear functions. In the next section, we will introduce some FPAA-based system implementations consisting different nonlinear functions described in Table 1.

3. FPAA-Based System Implementations

By using FPAA programmable device, a common part of the chaotic system and all of the nonlinear functions listed in Table 1 can be easily realized in the reconfigurable form. Here, we will present three FPAA-based system implementation of generalized chaotic system defined by Eqn.(1) and using three different nonlinear functions listed in Table 1. The first implementation has been constructed in FPAA environment according to Eqn.(1) which consists of nonlinear function defined by Eqn.(4) in Table 1. It is noted that this realization uses piecewise-linear nonlinearity. The dynamic ranges of the state variables u, v, w of this system are (-4.5, 4.5), (-4.4, 4.4) and (-4.4, 4.4). Because FPAA device has ±2 V saturation level, rescaling process is required for this system. After rescaling process, the system equations become as:

$$\begin{aligned}\dot{u}&=0.4v\\ \dot{v}&=0.4w \qquad (9)\\ \dot{w}&=-0.4v-0.24w+0.4[-2\,\mathrm{sgn}(u)+1.2u]\end{aligned}$$

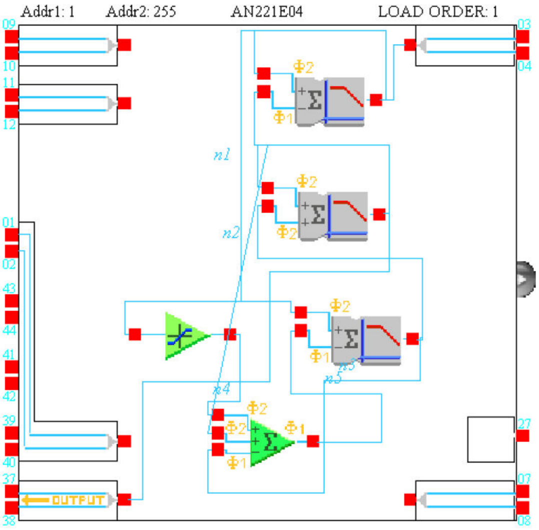
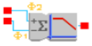
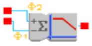
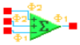


Fig. 1. FPAA scheme of chaotic system defined by Eqn.(9) and using piecewise-linear nonlinearity.

FPAA-based implementation scheme for chaotic system defined by Eqn.(9) is shown in Fig.1. FPAA schemes are prepared in the software development tool [3] that includes predefined configurable analog modules. The blocks used in this implementation have been described in Table 2 with parameter settings.

Table 2. FPAA modules and parameter settings for the chaotic system defined by Eqn.(9).

Name	Options		Parameters	
SumFilter1 **(SumFilter v1.1.2)**  *Anadigm* *(Approved)*	**Output Changes On** **Input 1** **Input 2** **Input 3**	*Phase 1* *Non-inverting* *Inverting* *Off*	**Corner Frequency [kHz]** **Gain 1 (UpperInput)** **Gain 2 (LowerInput)**	*0.399* *1.00* *0.889*
SumFilter2 **(SumFilter v1.1.2)**  *Anadigm* *(Approved)*	**Output Changes On** **Input 1** **Input 2** **Input 3**	*Phase 1* *Non-inverting* *Non-inverting* *Off*	**Corner Frequency [kHz]** **Gain 1 (UpperInput)** **Gain 2 (LowerInput)**	*0.800* *1.00* *1.00*
SumFilter3 **(SumFilter v1.1.2)**  *Anadigm* *(Approved)*	**Output Changes On** **Input 1** **Input 2** **Input 3**	*Phase 1* *Non-inverting* *Inverting* *Off*	**Corner Frequency [kHz]** **Gain 1 (UpperInput)** **Gain 2 (LowerInput)**	*0.799* *1.35* *1.00*
GainLimiter1 **(GainLimiter v1.0.3)**  *Anadigm* *(Approved*)*			**Gain** **Output Voltage Limit**	*20.0* *1.00*
SumDiff1 **(SumDiff v1.2.4)**  *Anadigm* *(Approved)*	**Output Phase** **Input 1** **Input 2** **Input 3** **Input 4**	*Phase 1* *Non-inverting* *Non-inverting* *Inverting* *Off*	**Gain 1 (UpperInput)** **Gain 2 (MiddleInput)** **Gain 3 (LowerInput)**	*2.25* *1.00* *0.400*

As shown in Fig.1, system state-variables *u, v* and *w* are represented at the output of SUMFILTER blocks. Circuit gains are implemented by SUMFILTER block gains. GAIN LIMITER and SUM/DIFFERENCE blocks were used for implementing piecewise-linear nonlinear function including sgn(.) term.

Fig.2. Experimental setup for FPAA-based implementation. AN221E04 type FPAA produced by Anadigm [3] has been used in this setup.

After modeling circuit implementation in FPAA software tool, this model is downloaded to the FPAA development board via serial interface as illustrated in Fig.2. Experimental measurements are obtained from I/O connections of the FPAA board. For monitoring chaotic dynamics in time and frequency domain, and X-Y mode, we used a virtual measurement system shown in Fig. 2. This system is a PC-compatible virtual measurement system using a PC oscilloscope module [6]. PC oscilloscope module incorporates a software program that turn into an oscilloscope and spectrum analyzer. This system is flexible, easy to use and has many advantages over conventional instruments, including multiple views of the same signal, and on-screen display of voltage and time. The chaotic dynamics and double scroll chaotic attractor obtained from the first FPAA-based chaotic system using piecewise-linear nonlinearity are shown in Fig.3, respectively.

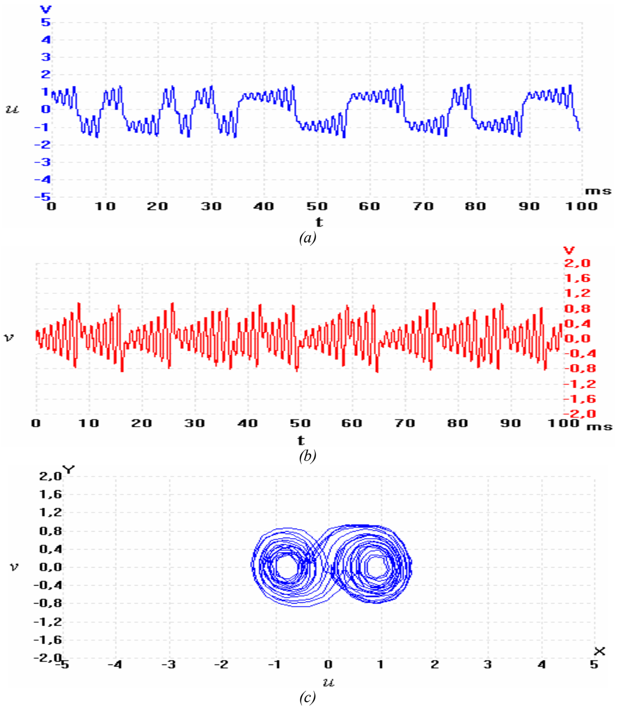


Fig.3. Experimental results for the first FPAA implementation, (a)-(b) Time responses of *u(t)* and *v(t)* chaotic dynamics, (c) Chaotic attractor projection in *u*-*v* plane.

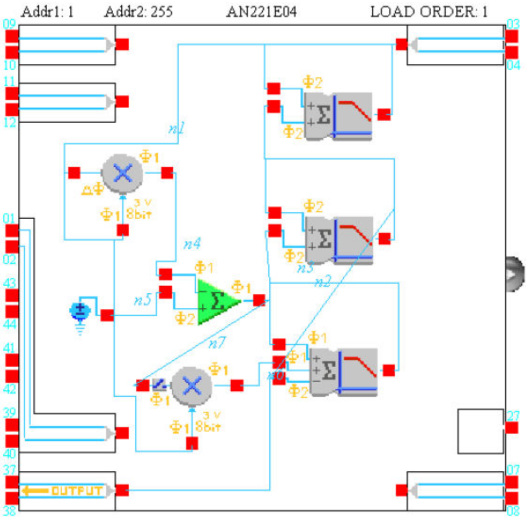


Fig.4. FPAA scheme of chaotic system defined by Eqn.(10) and using cubic-like nonlinearity.

After completing an experiment on a FPAA-based model and getting experimental results, the same FPAA device can be reconfigured for another modeling on the fly. So, in our modeling and experimental study for the generalized chaotic system, by keeping the common part of the circuit, we changed the nonlinear function used in the first implementation. The second FPAA-based implementation has been constructed in FPAA environment according to below mathematical model:

$$\begin{aligned}\dot u &= v\\ \dot v &= w \qquad\qquad (10)\\ \dot w &= -v - 0.6w + [1.6u(u^2/5 - 1)]\end{aligned}$$

Since the dynamic ranges of the state variables u, v, w in this model don't exceed the saturation level, it is not required a rescaling process. FPAA modeling according to the system definitions in Eqn.(10) is given in Fig.4. While the first FPAA-based implementation is realized with piecewise-linear nonlinearity, this modeling is realized with cubic-like nonlinearity. In addition to common blocks [SUMFILTER] used in both models, MULTIPLIER, SUM/DIFFERENCE and VOLTAGE REFERENCE blocks were used to implement cubic-like nonlinear function. After reconfiguration and downloading process, the experimental measurements can be done as in the first one. The chaotic dynamics and the double scroll chaotic attractor illustrations belonging to this modeling have been shown in Fig.5.

We will implement this chaotic system using another nonlinear function defined in Table 1. The third FPAA-based implementation has been constructed in FPAA environment according to below mathematical model:

$$\begin{aligned}\dot u &= v\\ \dot v &= w \qquad\qquad (11)\\ \dot w &= -v - 0.6w - 2.15u(u - 2\tanh(u))\end{aligned}$$

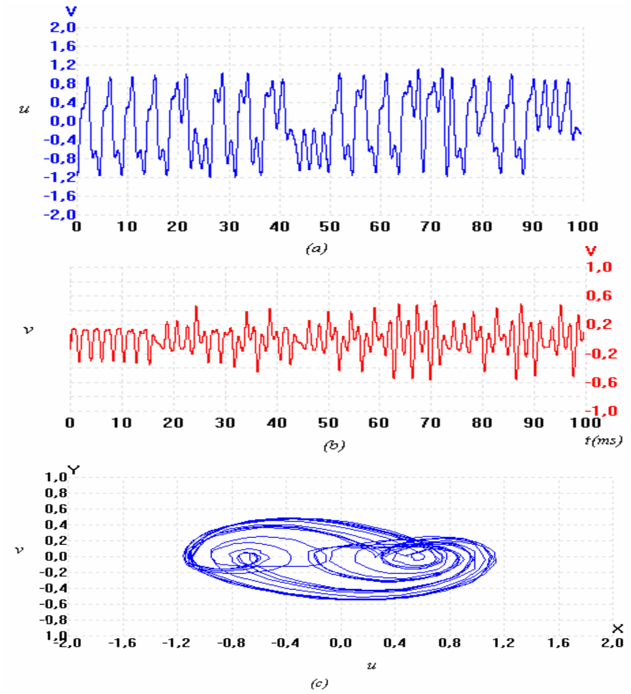


Fig.5. Experimental results for the second FPAA-based implementation, (a)-(b) Time responses of *u(t)* and *v(t)* chaotic dynamics, (c) Chaotic attractor projection in *u*-*v* plane.

Rescaling process is not required for this modeling as in the former mathematical modeling. FPAA modeling according to system definitions in Eqn.(11) is given in Fig.6. In addition to common SUMFILTER blocks used in other models, a user-defined TRANSFER FUNCTION block was used for implementing trigonometric function including tanh(.) term. This transfer function module produces an output voltage with 256 quantization steps according to a Lookup Table constituted by user. After reconfiguration and downloading process, the experimental measurements can be done as in the others. The chaotic dynamics and the double scroll chaotic attractor illustrations are shown in Fig.7.

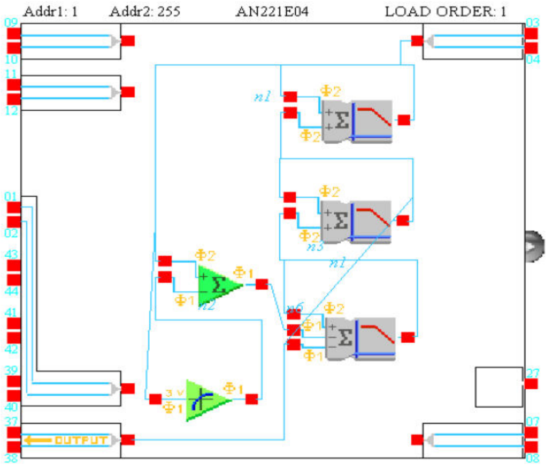


Fig.6. FPAA scheme of chaotic system defined by Eqn.(11) and using trigonometric nonlinearity.

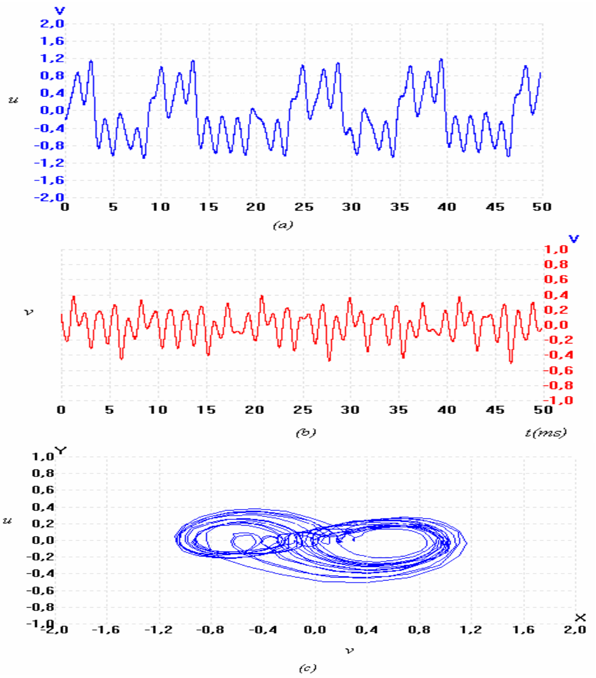


Fig.7. Experimental results for the third FPAA-based implementation, (a)-(b) Time responses of *u(t)* and *v(t)* chaotic dynamics, (c) Chaotic attractor projection in *u*-*v* plane.

4. Conclusion

FPAA-based programmable implementation of a chaotic system which is suitable for reconfigurable design according to its nonlinear structure has been presented. Experimental results agree with the results obtained from experiments established by discrete components. This system can be effectively used as a programmable chaos generator in many chaos-based applications.

Acknowledgments

This work has been supported by “The Scientific and Technological Research Council of Turkey” (TUBITAK) under project number 107E069. The authors would like to thank Serdar Ertekin (B.Sc.) and Alper Mutlu (B.Sc.) for their contribution to experimental studies.

References

[1] R. Caponetto, A.D. Mauro, L.Fortuna, and M.Frasca, “Field Programmable Analog Array to Implement a Programmable Chua’s Circuit,” *Int.Journal of Bifurcation and Chaos*., vol.15, pp.1829–1836, 2005.

[2] S. Callegari, R. Rovatti and G. Setti, “First Direct İmplementation of a True Random Source On Programmable Hardware”, *Int. J. Circ. Theor. Appl.*, vol.33,pp.1–16, 2005.

[3] Anadigm:www.anadigm.com

[4] J.C. Sprott, “A New Class of Chaotic Circuit”, *Physics Letters A*, vol.266, pp. 19-23, 2000.

[5] J.C. Sprott, “Simple chaotic systems and circuits”, *Am. J. Phys*, vol.68, pp. 758-763, 2000.

[6] Pico Technology Limited:www.picotech.com