

Memristör Tabanlı Kaotik Devre Modelinin FPGA Ortamında Tasarımı ve Gerçeklenmesi

Barış KARAKAYA*, Muhammet Emin ŞAHİN, Hasan GÜLER, Arif GÜLTEN
Fırat Üniversitesi Mühendislik Fakültesi Elektrik-Elektronik Mühendisliği Bölümü, 23119 ELAZIĞ
bkarakaya@firat.edu.tr

Geliş/Received: 12.07.2017; Kabul/Accepted: 31.08.2017

Özet

Bu çalışmada, memristör tabanlı kaotik devre modeli Alanda Programlanabilir Kapı Dizileri (Field Programmable Gate Array, FPGA) ortamında tasarlanıp gerçekleştirilmiştir. Bu kaotik devrede indüktör, kapasitör ve memristör olmak üzere üç devre elemanı bulunmaktadır. Devre modeli, dinamik durum denklemleri içermektedir. Kaotik davranış elde edilebilmesi için modele ait kaos parametreleri verilmektedir. Öncelikle, kaotik devreye ait durum denklemleri İleri Euler yöntemiyle ayrıklaştırılmaktadır. Sonra, ayrık zamanlı kaotik devre denklemleri, kaotik davranışın benzetimini elde etmek için Xilinx Sistem Üretici (Xilinx System Generator, XSG) yazılımında tasarlanmaktadır. Benzetim işleminden sonra, tasarım XSG aracılığıyla sentezlenip FPGA ortamında gerçekleştirilmektedir.

Anahtar Kelimeler: Kaos, Kaotik Devreler, FPGA, Memristör.

Design and Implementation of Memristor-based Chaotic Circuit Model on FPGA Platform

Abstract

In this study, memristor-based chaotic circuit model is designed and implemented on Field Programmable Gate Array (FPGA). In this chaotic circuit, there are three circuit elements; inductor, capacitor and memristor. The circuit model contains dynamic state equations. In order to obtain chaotic behavior, chaos parameters are given. At first, state equations of chaotic circuit are discretized with Forward Euler method. Then discrete time chaotic circuit equations are designed on Xilinx System Generator (XSG) in order to simulate chaotic behavior. After simulation process, the design is synthesized and implemented on FPGA through XSG.

Keywords: Chaos, Chaotic Circuits, FPGA, Memristor.

1. Giriş

Memristör, 1971 yılında, L. O. Chua tarafından dördüncü temel devre elemanı olarak literatüre sunuldu [1]. Memristör diğer temel devre elemanları olan direnç, indüktör ve kapasitörden farklı olarak üzerindeki bilgiyi depolayabilmesinden dolayı hafızalı direnç anlamına gelen *memory-resistor* kelimesinin kısaltılmasıyla memristör olarak isimlendirilmektedir. Memristörün, uzun yıllar boyunca teorik ve matematiksel ifadeden ibaret bir eleman olduğu ve gerçekte mevcut olmadığı düşünüldü. 2008 yılında, Stanley Williams yönetimindeki HP araştırma ve geliştirme ekibi tarafından yapılan çalışmalar sonucunda memristörün pratik olarak bulunduğu ilan edildi. Williams ve ekibi, *memristans* olarak adlandırılan

bu mekanizmayı tanımlayıp denklemini literatüre sundu [2,3].

Memristör konusunda gün geçtikçe daha fazla çalışma yapılmakta ve memristör farklı özelliklerinden dolayı araştırmacıların ilgisini çekmektedir. Böylece memristör içeren yapılar ile ilgili alınan patent ve yapılan proje çalışmalarının sayısı da hızla artmaktadır [4-6]. Memristör teorisi, Chua tarafından bir algı kayması olarak adlandırılmaktadır. Giderek daha fazla bilim adamının memristörü dördüncü temel devre elemanı olarak kabul ettiği görülmektedir.

Memristörün fiziksel gerçekleştirimi sonrası, bilim çevresinde ve mühendislik alanında bu elemanın kullanıldığı uygulamaları genişletmek için çalışmalar yapılmaktadır. Memristör; çok küçük boyutta olması, düşük enerji tüketimi, anahtarlama özelliği, sinaps benzeri davranışı ve

hafıza özelliğinin bulunmasının sonucu olarak literatürde birçok çalışma alanında kullanılmaktadır. Memristör tabanlı nöromorfik sistemler üzerine yapılan çalışmalardan bazıları, uyarılma ve öğrenme devreleri ile görüntü işleme konuları üzerinedir. Nöromorfik sistemlerin yanı sıra, geleneksel analog devreler de memristör ile yeniden tasarlanmaktadır. Kazanç kuvvetlendiriciler, analog süzgeçler, osilatörler ve Schmitt-Trigger devresi bu çalışmalara örnek olarak verilebilir. Analog elektronikte memristör tabanlı diğer uygulamalar ise kaotik devreler ve sinir ağlarıdır. Memristörün kendi yapısında barındırdığı doğrusal olmama özelliği kullanılarak kaos temelli bir çok uygulama gerçekleştirilmektedir. Bu tür kaotik devreler, haberleşme ve şifreleme sistemlerinde sıkça çalışılmaktadır [7-9].

Kaotik devrelerin gerçekleşmesi aşamasında birçok donanımsal yapı kullanılmaktadır. Bu yapılardan en çok tercih edileni FPGA çipi içeren donanımsal uygulama ve geliştirme kartlarıdır. FPGA donanım kartlarının en büyük avantajı, sistemin tasarımı yapıldıktan sonra kullanım alanında tekrar programlanabilir olması ve yüksek frekanslarda çalışabilmesidir. FPGA donanım kartları kullanılarak birçok kaotik devre tasarımı yapılmakta ve bu devrelerin çoğu şifreleme uygulamalarında kullanılmaktadır [10-15].

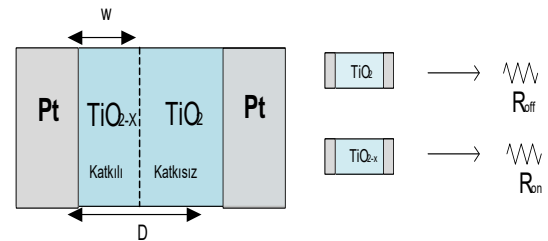
Bu çalışmada, memristör tabanlı kaotik devre modeli [16], Xilinx Sistem Üretici (XSG) yazılımı aracılığıyla FPGA ortamında tasarlanıp gerçekleştirilmektedir. Gerçekleme, Digilent firmasına ait NEXYS 4 Artix-7 FPGA donanım kartında yapılmaktadır. Çalışmanın amacı ise güvenli haberleşme ve şifreleme sistemlerinde kullanılabilir, programlanabilir ve yüksek frekanslarda çalışabilen kaotik bir entropi kaynağı elde etmektir. Bu giriş ile birlikte, ikinci bölümde materyal ve metot yer almaktadır. Bu bölümde memristöre ait durum denklemleri, memristör tabanlı kaotik devre modeli, modele ait durum denklemlerinin ayrıklaştırılması, ayrık zamanlı kaotik devrenin tasarımı ve gerçekleşmesi anlatılmaktadır. Üçüncü bölümde ise elde edilen sonuçlar değerlendirilmekte ve kaynaklar bölümü ile çalışma sonlandırılmaktadır.

2. Materyal ve Metot

2.1. Memristör

Titanyum (Ti) ve silisyum (Si) yarı iletkenleri ile oksijen (O) atomlarından faydalanılarak silisyum dioksit (SiO_2) ya da titanyum dioksit (TiO_2) elde edilir ve memristörün içyapısı bu bileşiklerden herhangi birisi kullanılarak oluşturulur. Genellikle TiO_2 analog devrelere daha uyumlu olduğu için içyapıyı oluşturmada tercih edilir. Memristörün içinde bu bileşiklerin yapılarındaki oksijenler ise küçük boşluklar halinde yerini alır. Memristörden akım geçtikçe katılanmış olan TiO_2 tabakadaki oksijen boşlukları akımın geçtiği yöne yani saf titanyum dioksitine doğru hareket ederler ve böylece saf TiO_2 aralığı gittikçe daralmaya başlar. Katkılı titanyum dioksitlerin arasındaki mesafe artar ve TiO_2 film tabakalarının sınırı değişerek direncin de değişmesine neden olur. Akım akmadığında ise katkılı ve saf TiO_2 tabakaları aralık düzeyini korur. Eğer akım ters yönden verilirse, oksijen boşlukları katkılı tabakaya doğru hareket eder. Ardından saf TiO_2 katmanını genişliğini artırarak direncin de artmasına sebep olur [3].

Memristör; üzerinden bir yönde akım geçerken direnci artan, zıt yönde akım geçtiğinde ise direnci azalan bir yapıya sahiptir. Memristörün üzerinden geçen akım kesildiğinde memristör, tekrar bir akım verilene kadar aynı direnç değerinde kalır. Bu durum memristörün bir hafızalı direnç (*memory-resistor*) olduğu anlamına gelir ve kısaltması olan memristör ismini de buradan almaktadır. Memristörün içyapısı ve akı – yük ilişkisi Şekil 1'de verilmektedir.



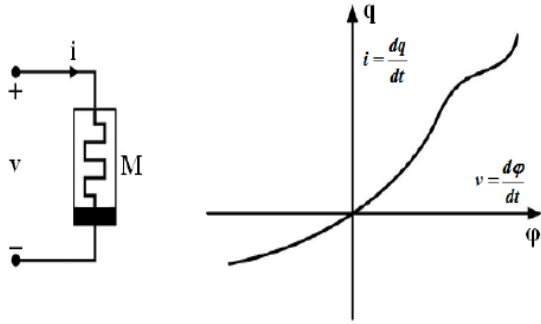
Şekil 1. Memristörün içyapısı.

$$d\phi = M.dq \quad (1)$$

Memristör için Denklem 1’de verilen akı-yük ilişkisi incelendiğinde, elektriksel karakteristik olarak memristans (M) elde edilirken, yük-akı ilişkisi memdüksans (W) karakteristiğini vermektedir. Memristans, direnç ile aynı birime sahip iken, memdüksans direncin biriminin tersi olarak gösterilmektedir. Aşağıda verilen denklemlerde de görüldüğü gibi memristör akı-yük ve yük-akı ilişkisi bakımından iki şekilde modellenebilmektedir. Şekil 2’de ise memristörün devre bağlantısı ile akı – yük karakteristiği verilmektedir.

$$M(q) = \frac{d\varphi(q)}{dq} \quad (2)$$

$$W(\varphi) = \frac{dq(\varphi)}{d\varphi} \quad (3)$$

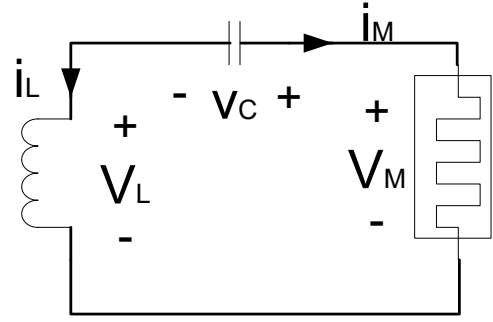


Şekil 2. Memristörün devre bağlantısı ve akı (φ) - yük (q) karakteristiği.

2.2. Memristör tabanlı kaotik devre modeli

Bu bölümde, memristör tabanlı kaotik bir devrenin dinamik davranışı incelenmektedir. Bu devre Muthuswamy ve Chua tarafından 2010 yılında tasarlanmış olup kaotik davranış sergileyen devre modeli Şekil 3’te verilmektedir. Üç elemanlı bu devrede, M memristörü temsil etmekte ve devreye ait dinamik durum denklemleri Denklem 4’te verilmektedir.

$$\begin{aligned} \dot{x} &= \frac{y}{C} \\ \dot{y} &= \frac{-1}{L} [x + \beta(z^2 - 1)y] \\ \dot{z} &= -y - \alpha z + yz \end{aligned} \quad (4)$$

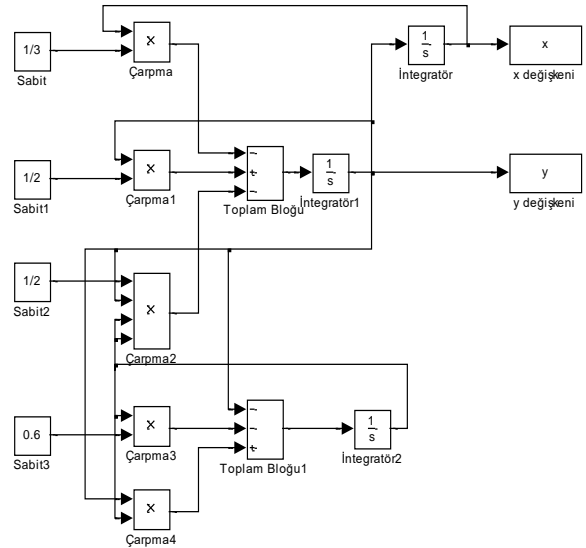


Şekil 3. Kaotik davranış sergileyen memristör tabanlı devre modeli.

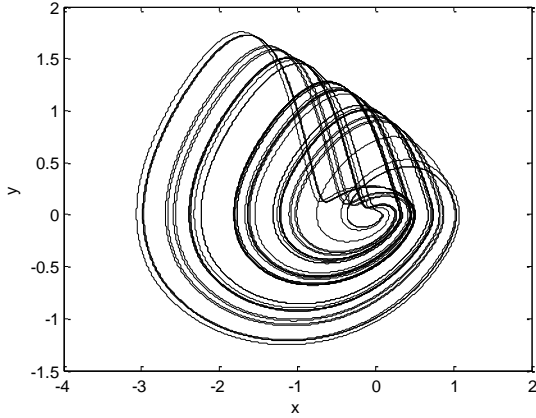
Dinamik durum denklemlerinde verilen kaos parametrelerinin değerleri sırasıyla $C=1$, $L=3$, $\alpha = 0.6$ ve $\beta=1.5$ verilmektedir. x ve y durum değişkenleri, sırasıyla kondansatör üzerine düşen gerilimi (V_C) ve indüktanstan geçen akımı (i_L) temsil etmektedir. z durum değişkeni ise memristif sistemin dahili durumunu aşağıdaki denklemlerle birlikte tanımlamaktadır.

$$\begin{aligned} V_m &= \beta(z^2 - 1)i_m \\ \dot{z} &= -i_m - \alpha z + i_m z \end{aligned} \quad (5)$$

Dinamik durum denklemleri ve tanımlamalarının zamana bağlı olarak MATLAB Simulink ortamında benzetimi ve $x - y$ durum uzay diyagramı sırasıyla Şekil 4 ve Şekil 5’te gösterilmektedir.



Şekil 4. Dinamik durum denklemlerinin MATLAB Simulink ortamında benzetim modeli.



Şekil 5. x - y durum uzay diyagramı.

2.3. Durum denklemlerinin ayrıklaştırılması

Diferansiyel denklemlerin sayısal tasarımlarının yapılabilmesi için denklemlerin zamandan bağımsız hale getirilerek ayrıklaştırılması gerekmektedir. Literatürde birçok ayrıklaştırma yöntemi yer almaktadır. Bu yöntemlerin birbirlerine göre avantaj ve dezavantajları vardır. Örneğin, bir yöntemde diğerine göre donanımsal olarak kaynak kullanımı az iken aynı yöntem diğerine kıyasla daha fazla hata payı ile ayrıklaştırma yapabilmektedir. Bu çalışmada kullanılan devre yapısı gereği hata payının uygun sınırlarda yüksek olması büyük önem arz etmemektedir. Çalışmanın sonucunda kaotik bir davranışın sayısal bir platformda tasarımı elde edileceğinden kaynak kullanımının az olması yöntem seçiminde etkin rol almaktadır. Bu nedenle ayrıklaştırma yöntemi olarak İleri Euler seçilmiştir. Memristör tabanlı kaotik devre modeline ait durum denklemleri $T= 1e-3$ adım aralığıyla aşağıdaki ayrık zamanlı denklemlere dönüştürülmektedir. Durum değişkenlerinin başlangıç değerleri ise $x(0) = (0.1)$, $y(0) = 0$ ve $z(0) = 0$ olarak uygulanmaktadır.

$$x[n+1] = x[n] + T * y[n]$$

$$y[n+1] = \frac{x[n] * \frac{(-T)}{3} + y[n]}{1 + 0.5 * T * (z[n]^2 - 1)} \quad (6)$$

$$z[n+1] = \frac{z[n] - T * y[n]}{1 + T * (0.6 - y[n])}$$

2.4. Xilinx sistem üretici platformu

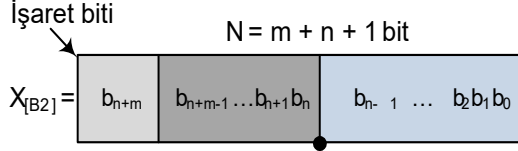
Xilinx Sistem Üretici (XSG), Xilinx FPGA kartlarında hızlı ve kolay bir şekilde tasarım oluşturmak, tasarımın donanım tabanlı ön-benzetimini (Hardware Co-Simulation) yapmak ve kart üzerinde gerçek zamanlı uygulamaları gerçekleştirmek için kullanılan MATLAB-Simulink tabanlı yüksek seviyeli bir yazılım platformudur [17]. XSG, aritmetik ve lojik uygulamalar, hafıza uygulamaları ve sayısal işaret işleme (Digital Signal Processing, DSP) uygulamaları için MATLAB-Simulink ortamında bit veya çevrim tabanlı bloklar içeren kütüphaneye sahiptir. XSG bloklarının genel Simulink bloklarından tek farkı XSG bloklarının ayrık zamanlı ve sabit noktalı sayı formatında kullanılır olmasıdır.

2.5. Ayrık zamanlı kaotik durum denklemlerinin XSG yazılımı ile tasarımı ve gerçekleştirilmesi

Bu çalışmada toplama ve çarpma gibi aritmetik işlemler yapılırken 32 bitlik işaretli ondalıklı sayılar kullanılmaktadır. Ondalıklı sayıların sayısal tasarımı yapılabilmesi için literatürde iki tip sayı formatı yer almaktadır. Bunlar; kayan noktalı (floating point) ve sabit noktalı (fixed point) sayı formatlarıdır. Ayrık zamanlı memristör tabanlı kaotik devre denklemlerinin tasarımında 2'ye tümleyen işaretli 32 bit sabit noktalı sayı formatı tercih edilmektedir. Sabit noktalı sayı formatını sembolize etmek için $Qm.n$ kullanılır. Bu formata göre 32 bitin en ağırlıksız bitinden başlayarak son 16 biti ondalık kısım için, en ağırlıklı biti işaret biti ve kalan 15 bit ise tam sayı kısmı için kullanılmaktadır. Böylece tasarımda kullanılan sayı formatı, $Q16.16$ olarak sembolize edilip Şekil 6'daki gibi gösterilmektedir. $Q16.16$ işaretli sabit noktalı sayı formatı gereği, ondalıklı sayının en ağırlıksız bitinin değişimine karşı düşen adım aralığı $= 2^{-n} = 2^{-16} = 1.5259 e-5$ olarak elde edilir.

XSG blokları kullanılarak oluşturulan tasarım, donanım tabanlı ön-benzetim, sentezleme ve donanım tanımlama dili (Hardware Description Language, HDL) kodlaması yapılarak gerçek zamanlı olarak çalıştırılabilir. Bu çalışmada, memristör tabanlı kaotik devre

modeli, XSG yazılımında tasarlanarak donanım tabanlı ön-benzetimi NEXYS 4 Artix-7 FPGA kartı ile gerçekleştirilmektedir.



Şekil 6. $Q_m.n$ sayı işaretli sabit noktalı sayı formatı gösterimi.

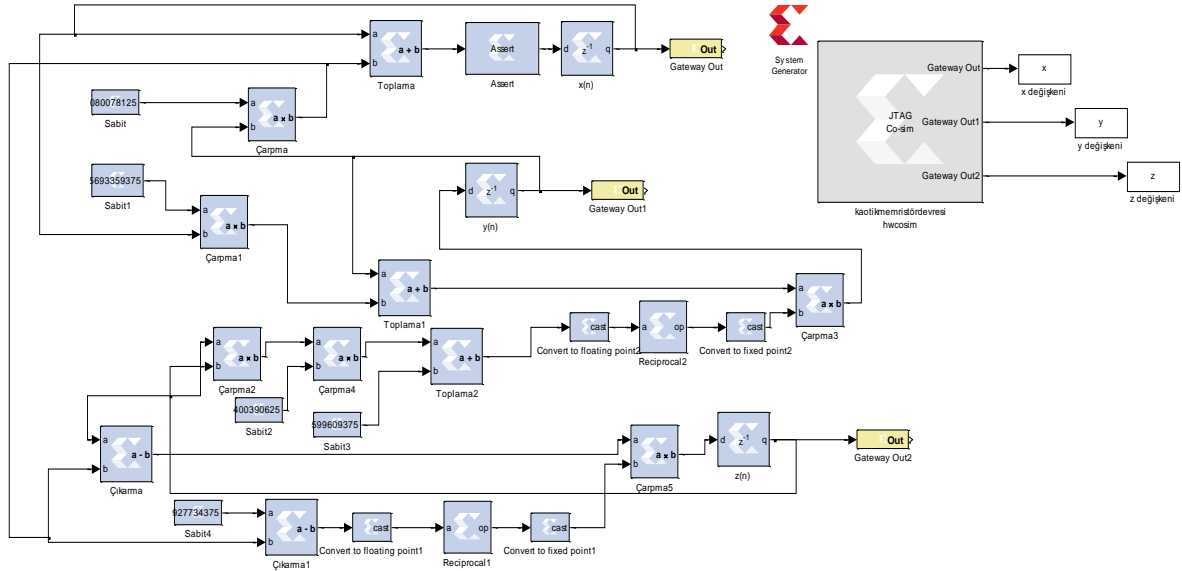
Kaotik devre modeli için oluşturulan tasarımda kullanılan aritmetik işlem blokları sayısı ve donanım üzerindeki kaynak kullanımı Tablo 1 ve Tablo 2'de sırasıyla verilmektedir. XSG yazılımında bir sayıyı diğer bir sayıya bölebilmek için bölünen tersini alıp bölünen ile çarpma işlemi yapılması gerekmektedir. Bu nedenle tasarımda, tersini alma işlemi bloğu (*Reciprocal*) kullanılmaktadır. Ayrıca bu blok, sadece kayan noktalı sayı formatında işlem yapabildiğinden, tasarımda bu bloğa gelen sinyal öncelikle sabit noktalı sayı formatından kayan noktalı sayı formatına dönüştürülüp tersini alma işlemi bittikten sonra aynı dönüşümün tersi işlemini yapmak gerekir. Bu nedenle tasarımda sayı formatı dönüştürücü bloğu kullanılmaktadır.

Tablo1. Kullanılan aritmetik işlem ve XSG blokları.

Aritmetik İşlem ve XSG Blokları	Kullanım Sayısı
Toplama İşlemi Bloğu	3
Çıkarma İşlemi Bloğu	2
Çarpma İşlemi Bloğu	6
Kayıtçı Bloğu	3
Sabit Değer Bloğu	5
Sayı Formatı Dönüştürücü Bloğu	4
Tersini Alma İşlemi Bloğu	2

Tablo 2. Donanım üzerindeki kaynak tüketimi.

Donanımsal kaynaklar	Kullanım Sayısı
Lojik Dilim	686
Flip-Flop	1044
Doğruluk Tablosu (LUT)	1136
Giriş çıkış (IOB)	96
Çoğullayıcı (Multiplexer)	24

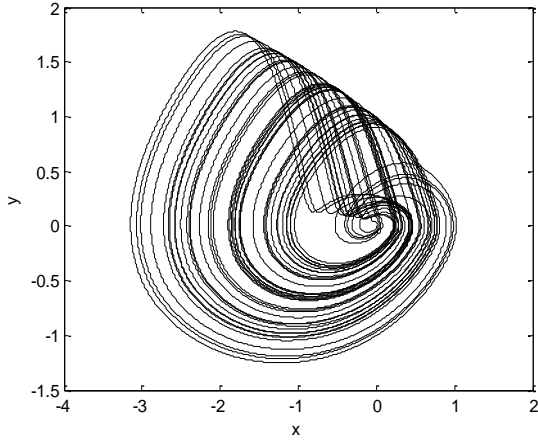


Şekil 7. Memristör tabanlı kaotik devre modelinin XSG ortamında tasarımı.

Memristör tabanlı kaotik devre modeli XSG yazılım ortamında tasarlanarak donanım tabanlı ön-benzetim işlemi JTAG Co-Sim bloğu

üzerinden NEXYS 4 Artix – 7 FPGA donanım kartında gerçekleştirilmektedir. Tasarıma ait ekran görüntüsü ve gerçekleştirme sonucu elde edilen $x - y$

durum uzay diyagramı sırasıyla Şekil 7 ve Şekil 8'de verilmektedir.



Şekil 8. FPGA gerçekleştirimi sonucu elde edilen $x - y$ durum uzay diyagramı.

3. Sonuçlar ve Tartışma

Bu çalışmada, memristör tabanlı kaotik devre modeli, XSG yazılımı aracılığıyla FPGA ortamında tasarlanıp gerçekleştirilmiştir. Gerçekleme, Digilent firmasına ait NEXYS 4 Artix-7 FPGA donanım kartında yapılmakta, tasarıma ait aritmetik işlem blokları ve donanımsal kaynak kullanım sayıları Tablo 1 ve Tablo 2'de verilmektedir.

Yapılan çalışma ile kaotik devrelerin sayısal gerçeklemelerine yeni bir boyut kazandırılmakta ve özellikle memristör kullanılarak oluşturulan devre modeli ilk kez XSG yazılımı aracılığıyla FPGA ortamında gerçekleştirilmektedir. İlerleyen çalışmalarda, memristör tabanlı kaotik devre modeli kullanılarak haberleşme sistemlerinde güvenliği artıracak şifreleme sistemi tasarımının gerçekleştirilmesi hedeflenmektedir.

4. Kaynaklar

1. Chua L.O. (1971), Memristor-the missing circuit element, IEEE Transactions on Circuit Theory, 18(5): 507 – 519.
2. Strukov D.B., Snider G.S., Stewart D.R. Williams R.S. (2008), The missing memristor found, Nature, 453: 80–83.
3. Yang J.J., Pickett M.D., Li X, Ohlberg D.A.A., Stewart D.R. Williams R.S. (2008), Memristive switching mechanism for metal/oxide/metal nanodevices, Nature Nanotechnology, 3: 429-433.

4. Sahin M.E., Guler H, Kaya T (2016), LabVIEW Model Of Memristor With Nonlinear Dopant Drift, European Journal of Technic, 6 (2) : 124-130.
5. Pershin Y.V., Ventra M.D. (2008), Spin memristive systems: Spin memory effects in semiconductor spintronics, Phys. Rev. B, Condens.Matter, 78: 1-4.
6. Driscoll T, Kim H.T., Chae B.G., Ventra M.D., Basov D.N. (2009), Phase-transition driven memristive system, E-print arXiv:0806.2151.
7. Arik S, Kilic R (2014), Reconfigurable hardware platform for experimental testing and verifying of memristor-based chaotic systems, Int. Journal of Circuits, Systems, and Computers, 23(10) : 1450145 18 pages.
8. Güler H, Kaya T (2016), Parça Parça Lineer Memristor Tabanlı Chua Osilatörünün LabVIEW'de Gerçekleştirilmesi, Fırat Üniv. Müh. Bil. Dergisi (Science and Eng. J of Fırat Univ.), 28 (2) : 29-33.
9. Itoh M, Chua L.O. (2008), Memristor oscillators, International Journal of Bifurcation and Chaos, 18(11) : 3183-3206.
10. Fischer V, Bernard F, Bochar N, Varchola, M (2008), Enhancing security of ring oscillator-based trng implemented in FPGA, International Conference on Field Programmable Logic and Applications, 245-250.
11. Wold K, Tan C.H. (2009), Analysis and Enhancement of Random Number Generator in FPGA Based on Oscillator Rings, International Journal of Reconfigurable Computing, Article ID 501672, 8 pages.
12. Wold K, Petrović S (2011), Behavioral Model of TRNG Based on Oscillator Rings Implemented in FPGA, 14th IEEE International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS), 163-166.
13. Tuncer T, Avaroğlu E, Türk M, Özer A.B. (2014), Implementation of non-periodic sampling true random number generator on FPGA, J. Microelectron. Electron. Compon. Mater, 44(4) : 296–302.
14. Karakaya B, Çelik V, Gülten A (2017), Chaotic cellular neural network-based true random number generator, Int. J. Circ. Theor. Appl., doi: 10.1002/cta.2374.
15. Karakaya B, Çelik V, Gülten A (2017), Implementation Of Two Cell Non-Autonomous Cnn Model On FPGA, International Conference on Advances and Innovations in Engineering (ICAIE) 2017.
16. Muthuswamy B, Chua L.O. (2010), Simplest Chaotic Circuit, International Journal of Bifurcation & Chaos, 20(5) : 1567-1580.
17. Xilinx Inc., System Generator for Digital Signal Processing, <http://www.xilinx.com / tools / dsp.htm>.