

FPGA tabanlı farklı nümerik algoritmalar ile kaotik osilatör tasarımları

İsmail Koyuncu¹, Murat Tuna² and Murat Alçın^{3*}

¹Afyon Kocatepe Üniversitesi, Teknoloji Fakültesi, Elektrik-Elektronik Mühendisliği, Afyon, Türkiye

²Kırklareli Üniversitesi, Teknik Bilimler Meslek Yüksekokulu, Elektrik Bölümü, Kırklareli, Türkiye

³Afyon Kocatepe Üniversitesi, Teknoloji Fakültesi, Mekatronik Mühendisliği, Afyon, Türkiye

*Sorumlu yazar e-mail: muratalcin@aku.edu.tr

Özet

Bu çalışmada, literatüre sunulan sürekli zamanlı otonom 3-Boyutlu kaotik sistem, FPGA üzerinde IQ-Math sabit noktalı sayı formatında Euler, Heun, RK4 ve RK5-Butcher nümerik algoritmaları kullanılarak VHDL dilinde ayrık zamanlı olarak gerçekleştirilmiştir. Tasarlanan bu sistemler Xilinx Virtex-6 ailesi FPGA çipinde sentezlenmiş ve test edilmiştir. FPGA üzerinde tasarlanan kaotik osilatörlerin performans analizleri ve çip donanım kullanım istatistikleri karşılaştırılmıştır. Ayrıca FPGA üzerinden alınan ayrık zamanlı sonuçlar, Matlab programından alınan sürekli zamanlı sonuçlarla karşılaştırılarak başarılı sonuçlar elde edildiği görülmüştür. Kaotik osilatörlerin bilgisayar ortamında kurgulanan benzetimi ile FPGA üzerinde gerçekleştirilen tasarımın aynı sonuçlar vermesi ile bu sistemlerin donanımsal olarak sentezlenmesi sağlanmıştır. Bu sayede farklı nümerik algoritmalarla ayrık zamanlı olarak tasarlanan FPGA-tabanlı kaotik osilatörler ile gömülü kaos-tabanlı çeşitli mühendislik uygulamaları gerçekleştirilebilir.

Anahtar Kelimeler: Kaotik sistemler, Nümerik Algoritmalar, Sayısal gerçekleştirme, FPGA, VHDL.

1. GİRİŞ

Günümüzde elektrik-elektronik mühendisliği üzerinde araştırma yapılan bilim alanlarından birisi de doğrusal olmayan kaotik sistemlerin devreler üzerinde oluşturdukları olumlu/olumsuz etkileri üzerine olmuştur (Çavuşoğlu et al., 2014). Kaos tanımı incelendiğinde başlangıç koşullarına üstel duyarlı, nonlinear, deterministik karakterli, uzun vadede periyodik olmayan dinamik sistemler olduğu görülmektedir (Tuna et al., 2018; Tuna and Fidan, 2016). Kaotik sistemlerin çok karmaşık olmayan devre yapılarına sahip olmalarına rağmen ilginç dinamik özelliklere sahip olmaları, son yıllarda kaotik sistemlere olan ilgiyi oldukça arttırmıştır (Koyuncu et al., 2014; Pehlivan and Wei, 2012). Son yıllarda kaotik sistemlerin araştırılması ve uygulanmasına yönelik biyomedikal, haberleşme, optik elektronik ve elektromanyetik, görüntü işleme (Rajagopal et al., 2017), bulanık mantık, güç elektroniği, optimizasyon, robot kontrolü ve mekatronik kriptografi, güvenli haberleşme, gürültü üreteçleri, şifreleme (Çiçek et al., 2013) ve rasgele sayı üreteçleri (Tuna and Fidan, 2018) gibi alanlar örnek olarak verilebilir (I. Koyuncu et al., 2013; İsmail Koyuncu et al., 2013; Lai et al., 2018; Tuna et al., 2016). Kaosun elektronik yapılarda başlıca kullanım alanlarına bakıldığında öncelikle şifreleme, analog işaret işleme, güç elektroniği, sayısal haberleşme ve entropi kaynağı olarak gürültü işareti yerine kullanıldığı söylenebilir (Akgül et al., 2015). Ayrık ve sürekli zamanda kaotik sistemler kullanılarak bu belirtecek ihtiyaçları karşılayacak devreler tasarlanabilmektedir (Kaçar, 2016).

Kaotik sistemlerin bilgi güvenliği alanlarında geniş bir uygulama alanına sahip olmasından dolayı bu sistemler analog veya sayısal tabanlı olmak üzere iki farklı şekilde gerçekleştirilebilmektedir (Alçın et al., 2018; Tavas et al., 2010). Sayısal devre tabanlı FPGA (Field Programmable Gate Array) çipleri yüksek hız ve kapasiteleri nedeniyle özellikle yüksek performans ve işlemci gücü gerektiren kriptoloji ve güvenli haberleşme gibi uygulamalarda önemli bir potansiyele sahiptir (Ergun and Ozoguz, 2007; Sadoudi et al., 2009; Tuna et al., 2015). FPGA çipleri tasarımcının yeniden ve sahada programlayabileceği tüm devrelerdir (Çavuşoğlu et al., 2016). Kullanıcının tasarladığı mantıksal devreye göre, mantıksal bloklar, aralarındaki bağlantılar ve giriş/çıkış blokları ile programlanabilmektedir. Paralel işlem yapabilme kabiliyeti başta olmak üzere, sahada programlanabilme, tasarımın test edilip doğrulanabilmesi ve istenirse içine işlemci sistemi dahi gömülebilmesi gibi birçok özelliği FPGA günümüzde popüler kılan taraflarıdır. FPGA tabanlı kaotik sistemler üzerindeki donanımsal çalışmalar genel olarak kayan noktalı sayı sistemi kullanılarak gerçekleştirilmiştir (Akgül et al., 2015; Koyuncu, 2018). Fakat

donanımsal olarak daha etkin gerçekleştirme özelliğine sahip sabit noktalı sayı gösterimi kullanılarak tasarlanan kaotik sistemlere gerektiği kadar değinilmemiştir.

Bu çalışma da Pehlivan vd. tarafından literatüre sunulan 3 boyutlu kaotik sistem sabit noktalı sayı formatında Euler, Heun, RK4 ve RK5-Butcher nümerik algoritmaları kullanılarak FPGA üzerinde tasarımı gerçekleştirilmiştir. Bu sayede sabit noktalı sayı tabanlı 3 boyutlu kaotik sistemin donanımsal olarak basit, hızlı ve daha etkin bir biçimde tasarlanması ve gerçekleştirilmesi amaçlanmıştır. İkinci bölümde matematiksel modeli verilen kaotik sisteminin ayrılaştırılmış matematiksel modelleri verilerek üçüncü bölümde FPGA üzerinde 32 bit IQ-Math (16I-16Q) sabit noktalı sayı formatında dört farklı nümerik algoritma kullanılarak VHDL dilinde ayrık zamanlı olarak tasarımları yapılmıştır. Tasarlanan bu sistemler Xilinx Virtex-6 ailesi XC6VLX75T-3FF784 FPGA çipinde sentezlenmiş ve test edilmiştir. Son bölümde FPGA üzerinde farklı nümerik algoritmalarla tasarlanan 3 boyutlu kaotik sistemin hem sabit noktalı sayı gösteriminden hemde kayan nokta sayı gösteriminden elde edilen sonuçlar karşılaştırılıp değerlendirilmiştir.

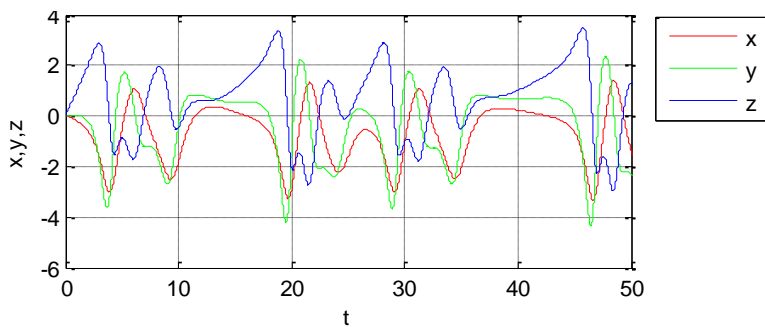
2. MATERYAL VE METOD

2.1. Kaotik sistemin matematiksel modeli

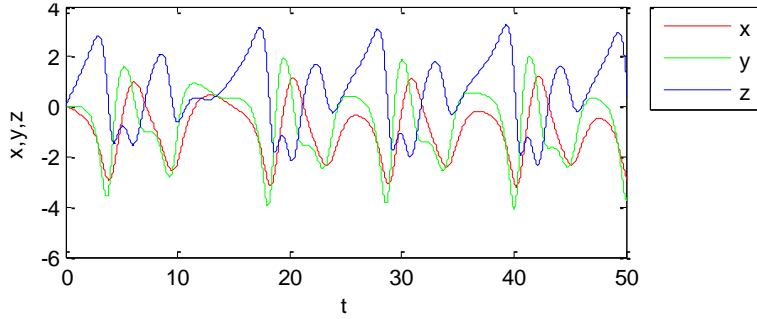
Pehlivan vd. yaptıkları çalışmalarında yeni tasarladıkları üç boyutlu sürekli zamanlı iki kuadratik doğrusalsızlık içeren otonom bir kaotik sistem tanıtmışlardır (Pehlivan and Uyaroğlu, 2012). Kaos uygulamaları için önerilen yeni sistemin matematiksel modeli Denklem (1) de verilmiştir. Önerilen bu sistem, 2 adet ikinci dereceden doğrusal olmayan ifade (xz ve xy) olmak üzere 8 terim ve 2 parametre içermektedir. Matematiksel modelde verilen $a=0,50$ ve $b=1,0$ sistem parametrelerinin değerleri ve kaotik sistemin başlangıç şartları ($x_0=0$, $y_0=0$, $z_0=0$) da olarak verilmiştir.

$$\begin{cases} \dot{x} = y - x - a.z \\ \dot{y} = x.z - x \\ \dot{z} = -x.y - y + b \end{cases} \quad (1)$$

Önerilen kaotik sistem Matlab programında Euler, Heun, RK4 ve RK5-Butcher nümerik diferansiyel denklem çözüm yöntemleri ile modellenmiştir. Matlab üzerinden alınan sonuçlar, FPGA üzerinde nümerik algoritmalarla tasarlanan 3-boyutlu kaotik sistemden alınan sonuçların doğrulanması amacıyla kullanılmıştır. Makalenin şekil yönüyle çok kalabalık olmaması için burada kaotik sistemin Şekil 1’de Euler ve Şekil 2’de RK4 tabanlı nümerik analiz sonucunda elde edilen x, y, z kaotik durum değişkenlerinin zaman göre değişimi verilmiştir.

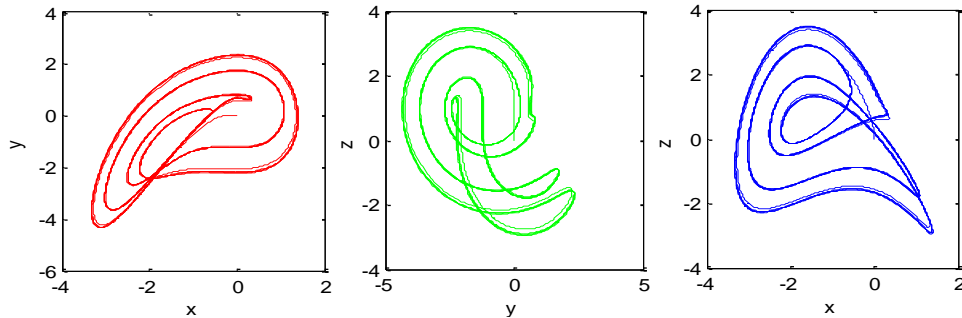


Şekil 1. Kaotik sistemin Matlab’ta tasarlanan nümerik Euler modeli x, y, z zaman serileri

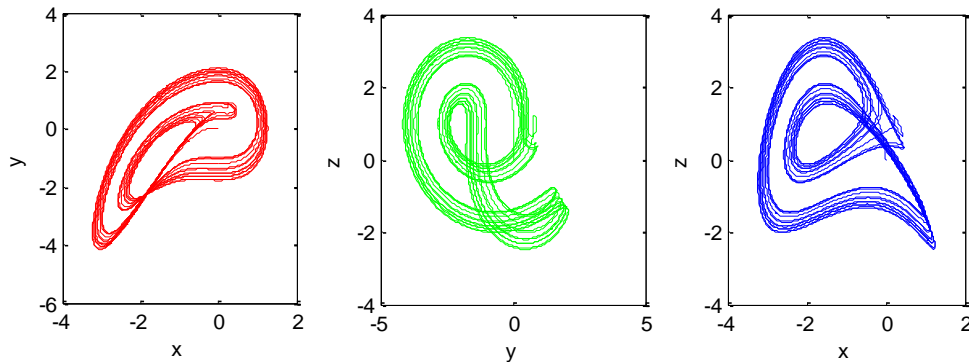


Şekil 2. Kaotik sistemin Matlab'ta tasarlanan nümerik RK4 modeli x, y, z zaman serileri

Aynı parametre ve başlangıç şartlarındaki x-y, x-z ve y-z kaotik faz portreleri ile üç boyutlu x-y-z yörüngesi Euler ve RK4 tabanlı olarak sırasıyla Şekil 3 ve Şekil 4'te görülmektedir.



Şekil 3. Kaotik sistemin Matlab tasarımının Euler nümerik modelinin x-y, x-z, y-z faz portreleri



Şekil 4. Kaotik sistemin Matlab tasarımının RK4 nümerik modelinin x-y, x-z, y-z faz portreleri

2.2. Kaotik Sistemin Ayırıklaştırılması

Bu kısımda üzerinde önerilen kaotik sistem Euler, Heun, RK4 ve RK5-Butcher nümerik algoritmaları ile 32 bit IQ-Math sabit sayı standardında VHDL dili ile modellenmiştir. FPGA üzerinde yapılan gerçekleştirilmeden alınan sonuçların performans ve çip istatistikleri incelenerek kullanılan algoritmaların hassasiyet analizleri yapılmıştır. Bu bölümde Euler ve RK4 algoritmaları kullanılarak çıkartılan ayırıklaştırılmış matematiksel denklemler verilecektir. Algoritmaların ayırıklaştırılmış modellerindeki $x(k)$, $y(k)$ ve $z(k)$ 'nin başlangıç değerleri kullanılan kaotik sistemi için $x(k)=0,0$, $y(k)=0,0$ ve $z(k)=0,0$ olarak alınmıştır. Kaotik sisteminin Euler algoritması kullanılarak ayırıklaştırılmış matematiksel modeli Denklem (2) de sunulmaktadır.

$$\begin{aligned}
 x(k+1) &= x(k) + \Delta h(y(k).z(k) - \rho.y(k)) \\
 y(k+1) &= y(k) + \Delta h(y(k).z(k) - \rho.y(k) - x(k).z(k) - \rho.x(k)) \\
 z(k+1) &= z(k) + \Delta h(-y(k).\rho.x(k) + y^2(k) - \sigma.z(k) + \sigma.\rho)
 \end{aligned}
 \tag{2}$$

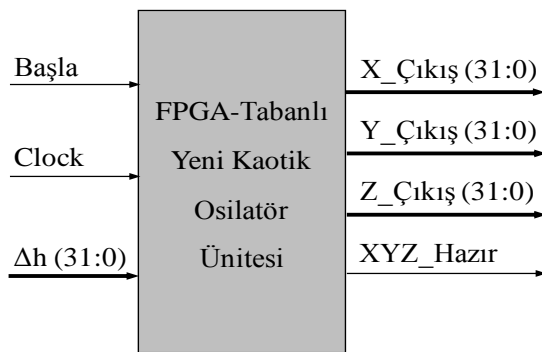
RK4 algoritması kullanılarak kaotik sistemin ayrıklaştırılmış matematiksel modeli Denklem (3) teki f , g ve δ fonksiyonlarına göre Denklem (4) te verilmektedir. Bu denklemlerde bulunan $\kappa_1, \kappa_2, \kappa_3$ ve κ_4 parametreleri, kaotik sistemin birinci denkleme ait katsayıları, $\lambda_1, \lambda_2, \lambda_3$ ve λ_4 parametreleri ikinci denkleme ait katsayıları ve ξ_1, ξ_2, ξ_3 ve ξ_4 parametreleri ise üçüncü denkleme ait katsayıları göstermektedir. Bu katsayılar RK4 algoritmasında yerine konularak, kaotik sistemin Δh kadar adım sonraki değeri olan $x(k+1)$, $y(k+1)$ ve $z(k+1)$ değerleri hesaplanmaktadır. Her iterasyon sonunda sistemin çıkışları olan $x(k+1)$, $y(k+1)$ ve $z(k+1)$ değerleri hem çıkış olarak hem de bir sonraki iterasyon da algoritmanın başlangıç şartları olarak kullanılmaktadır.

$$\begin{aligned}\dot{x} &= f(t, x, y, z) = y \cdot (z - \rho) \\ \dot{y} &= g(t, x, y, z) = y \cdot (z - \rho) - x \cdot (z + \rho) \\ \dot{z} &= \delta(t, x, y, z) = -y \cdot (\rho \cdot x - y) - \sigma \cdot (z - \rho)\end{aligned}\quad (3)$$

$$\begin{aligned}x(k+1) &= x(k) + \frac{1}{6} \Delta h [\kappa_1(k) + 2\kappa_2(k) + 2\kappa_3(k) + \kappa_4(k)] \\ y(k+1) &= y(k) + \frac{1}{6} \Delta h [\lambda_1(k) + 2\lambda_2(k) + 2\lambda_3(k) + \lambda_4(k)] \\ z(k+1) &= z(k) + \frac{1}{6} \Delta h [\xi_1(k) + 2\xi_2(k) + 2\xi_3(k) + \xi_4(k)]\end{aligned}\quad (4)$$

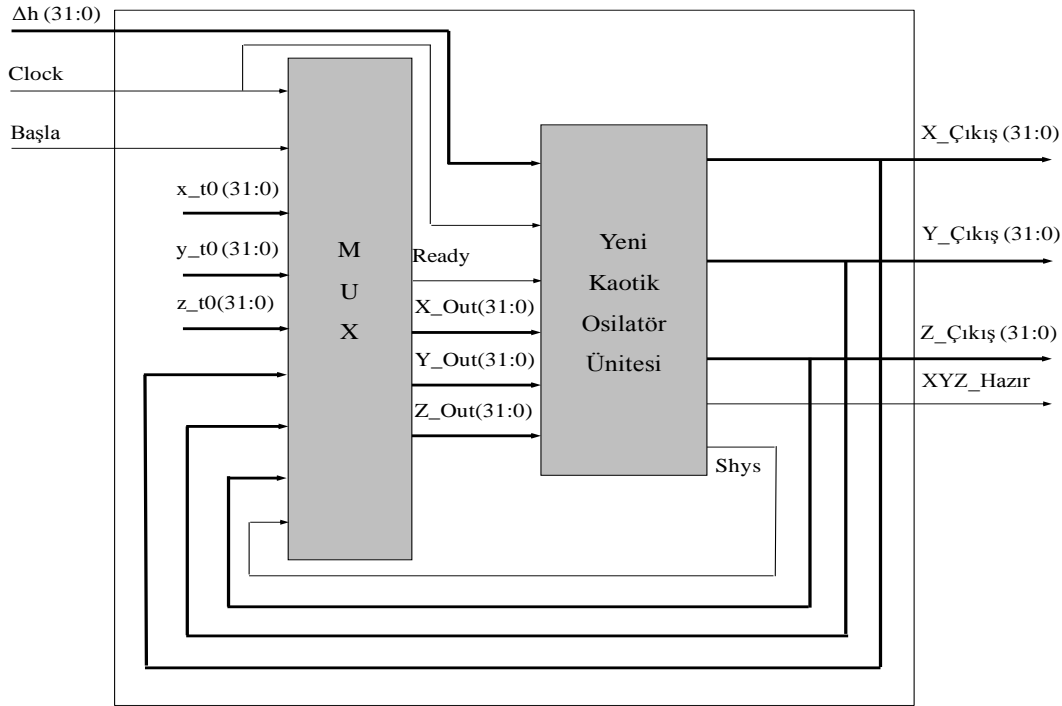
2.3. Kaotik Sistemin FPGA Üzerinde Gerçekleştirilmesi

Bu bölümde 3-boyutlu kaotik sistem Euler, Heun, RK4 ve RK5-Butcher nümerik algoritmaları kullanılarak 32 bit IQ-Math sabit noktalı sayı standardı ile FPGA tabanlı olarak modellenmiş ve bir donanım tanımlama dili VHDL ile kodlanmıştır. Yapılan tasarımlarda kullanılan sabit noktalı sayı standardına uygun çarpıcı, toplayıcı ve çıkarıcı gibi üniteler, Xilinx ISE Design Tools ile geliştirilen IP CORE Generator kullanılarak oluşturulmuştur. Euler, Heun, RK4 ve RK5-Butcher tabanlı Ünitelerin en üst seviye blok diyagramı tüm algoritmalarda aynı olup Şekil 5'te görülmektedir. Ünitelerin girişlerinde bulunan 1-bitlik "Başla" ve "Clock" sinyalleri, ünitelerin içerisindeki alt ünitelerin zamanlaması ve ünitelerin bağlı bulunduğu sistem ile arasındaki senkronizasyonu sağlamak amacıyla kullanılmaktadır. Algoritmanın hassasiyetini belirleyen " Δh " adım sayısı parametresini belirtmektedir. Bu sinyal tasarımın daha esnek olmasını sağlamak amacıyla dışarıdan uygulanmaktadır. Sistemin ilk çalışması anında ihtiyaç duyduğu başlangıç şartları, tasarımda kullanılan FPGA çipinin kaynaklarını azaltmak amacıyla tasarımın içerisine gömülmüştür. Ancak ihtiyaç duyulduğunda bu sinyaller, 32 bitlik 3 farklı sinyal tanımlaması yapılarak tasarımda küçük değişiklikler ile değerleri kullanıcı tarafından ayarlanacak şekilde de tasarlanabilir. Tasarlanan Euler, Heun, RK4 ve RK5-Butcher tabanlı kaotik osilatörler de 3 adet sabit noktalı sayı standardında 32 bit çıkış sinyalleri "X_Çıkış", "Y_Çıkış", "Z_Çıkış" ve bu çıkış sinyallerinin hazır olduğunu göstermek amacıyla kullanılan 1-bit "XYZ_Hazır" sinyali bulunmaktadır.



Şekil 5. FPGA-tabanlı yeni kaotik osilatör ünitesi en üst seviye blok diyagramı

Şekil 6’da yeni kaotik osilatör ünitesinin ikinci seviye blok diyagramı görülmektedir. Sistemin ikinci seviye blok diyagramını bir *Multiplexer Ünitesi* ve bir tane de *FPGA-Tabanlı Yeni Kaotik Osilatör Ünitesi*’nden oluşmaktadır. Tasarımda *Multiplexer Ünitesi* kullanılmasının amacı, başlangıç koşulu değerlerini ilk çalışma anında, kullanıcı tarafından atanan başlangıç sinyalleri olan 32 bit sabit noktalı sayı formatında x_{t0} , y_{t0} ve z_{t0} sinyallerinden almasını ve bundan sonraki tüm aşamalar için bu değerlerin *Yeni Kaotik Osilatör Ünitesi* çıkışından almasını sağlamaktır. 1-bitlik *Shys* sinyali kaotik sistem sonuç ürettiği durumlarda ‘1’, bunun dışındaki tüm durumlarda ‘0’ değeri vermektedir. Bu şekilde *Yeni Kaotik Osilatör Ünitesi* ilk değerlerini ürettiğinde, *Shys* sinyali ‘1’ olmakta ve bu sinyali *Multiplexer Ünitesi*’ne göndererek kullanıcı tarafından atanan başlangıç değerleri yerine kaotik sistemin ürettiği değerleri kullanmasını sağlamaktadır.



Şekil 6. FPGA-tabanlı yeni kaotik osilatör ünitesi ikinci seviye blok diyagramı

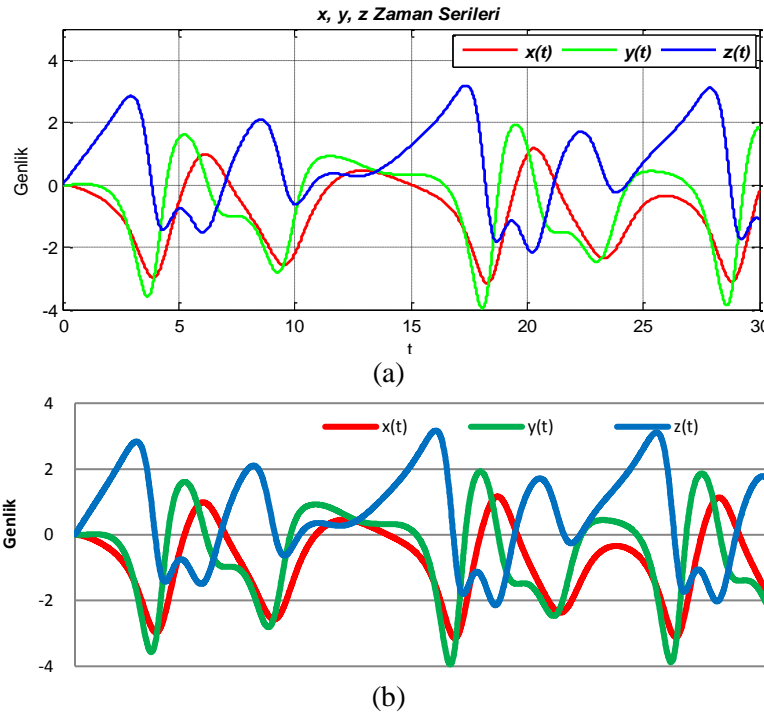
2.4. FPGA Tabanlı kaotik sistemin Test sonuçları

Euler, Heun, RK4 ve RK5-Butcher Tabanlı Osilatör Üniteleri, Xilinx Virtex-6 ailesi XC6VLX75T çipi için sentezlenerek, FPGA çip kaynak kullanımına ve ünitelerin saat hızlarına ait parametrelerin istatistikleri incelenmiştir. Tasarımı yapılan 4 farklı yapıdaki ünitelerin verileri işleme süresi, Xilinx ISE Design Tools 14.2 benzetim programı kullanılarak elde edilmiştir. Burada kaotik osilatörün ISE Design Tools kullanılarak FPGA’de gerçekleştirilmesinden elde edilen x , y ve z sinyallerinin kaotik osilatör ünitesinde karşılıkları olan $X_{\text{Çıkış}}$, $Y_{\text{Çıkış}}$ ve $Z_{\text{Çıkış}}$ sinyallerinin zaman serilerine ait değerler 32 bit tam sayı formatında gösterilmiştir. Euler, Heun, RK4 ve RK5-Butcher tabanlı kaotik osilatör ünitelerinin Xilinx ISE Simülatöründen elde edilen sonuçlar sırasıyla Şekil 7-10’da verilmiştir.

osilatör Xilinx Virtex-6 ailesi FPGA çipinde sentezlenerek test edilmiştir. Tablo 1’de Euler, Heun, RK4 ve RK5-Butcher tabanlı kaotik osilatörlerin sentezlenmesi işleminin ardından yapılan Place&Route sonrasında elde edilen Xilinx Virtex-6 ailesi xc6vlx75T-3ff784 FPGA çip istatistikleri verilmiştir. Çip istatistiklerinden de görüldüğü üzere kaotik osilatörlerin maksimum çalışma frekansı 464,688 MHz ve minimum çalışma periyodu 2,152 ns’dir. Ayrıca kaotik osilatörlerin FPGA üzerinde gerçekleştirilmesinden elde edilen $X_Çıkış$, $Y_Çıkış$ ve $Z_Çıkış$ sinyallerinin zaman serilerine ait 32 bit IQ-Math formatındaki ikilik değerler benzetim test aşamasında bir dosyaya kaydedilmiştir. Kaydedilen değerler gerçel sayı sistemine dönüştürüldükten sonra kaotik osilatörün ürettiği ilk 3x3500 veri seti yardımıyla $X_Çıkış$, $Y_Çıkış$ ve $Z_Çıkış$ sinyallerinin zaman serileri ve faz portreleri elde edilmiştir. Örnek olarak Şekil 11a’da Matlab ve Şekil 11b’de FPGA üzerinde gerçekleştirilen RK5-Butcher tabanlı kaotik sistemin $x(t)$, $y(t)$, ve $z(t)$ zaman serileri birbirleriyle karşılaştırılmıştır. Karşılaştırma sonucunda iyi bir eşleşme olduğu gözlemlenmiştir.

Tablo 1. Yeni Kaotik osilatörlerin Xilinx Virtex-6 FPGA çip istatistikleri.

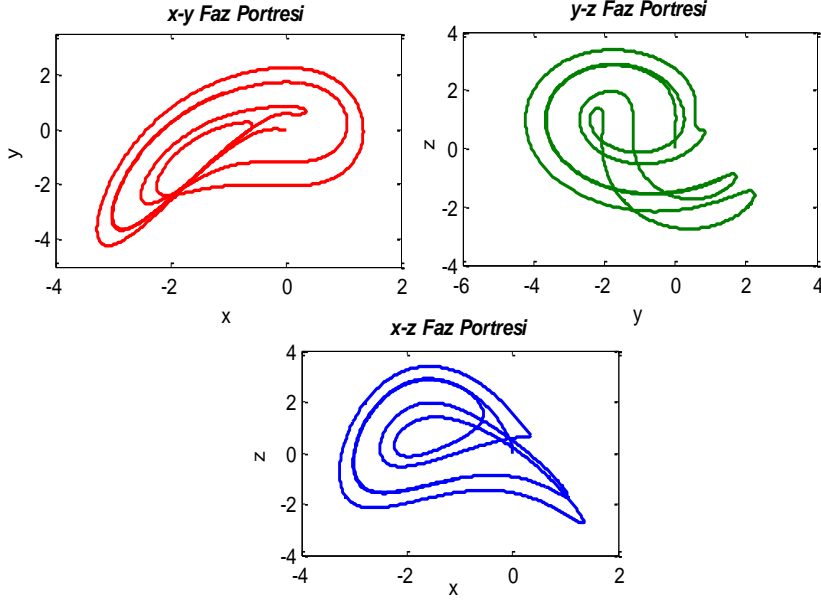
Kaotik Osilatör	Slice Regs. Sayısı / %	LUTs Sayısı / %	Bonded IOBs Sayısı / %	Maks. Saat Frekansı (MHz)	Minimum Darbe Periyodu (ns)
	Kullanılan / Kullanım Oranı %				
Euler-Tabanlı Kaotik Sistem	1196 / 0	1070 / 0	99 / 13	464,688	2,152
Heun-Tabanlı Kaotik Sistem	2447 / 2	2328 / 5	99 / 13	464,688	2,152
RK4-Tabanlı Kaotik Sistem	5338 / 5	5039 / 10	99 / 13	464,688	2,152
RK5-Butcher Kaotik Sistem	10794 / 11	10070 / 21	99 / 13	436,143	2,292



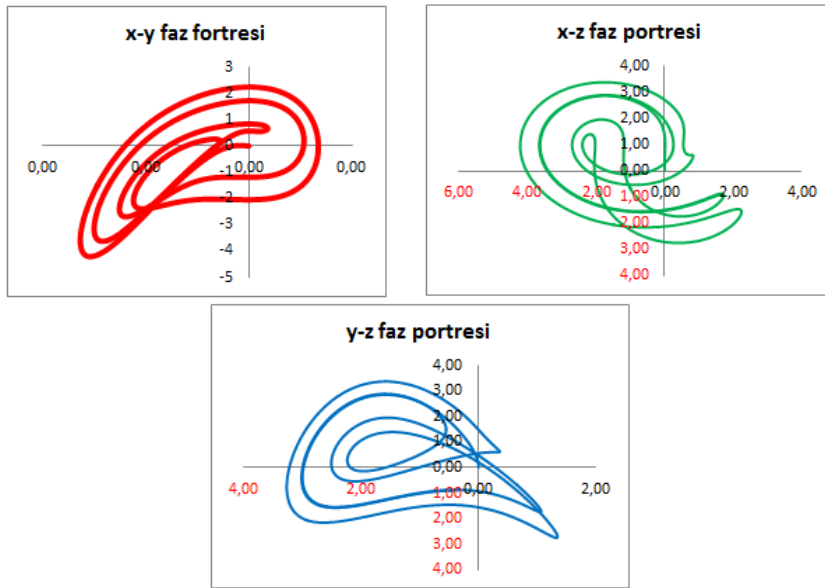
Şekil 11. RK5-Butcher tabanlı kaotik tasarımının $x(t)$, $y(t)$, ve $z(t)$ zaman serilerinin karşılaştırılması

a) Matlab üzerinde tasarım, b) FPGA üzerinde 32 bit gerçekleştirme

Ayrıca Şekil 12’de Matlab üzerinden alınan Euler tabanlı kaotik osilatörün faz portreleri ile Şekil 13’te Euler tabanlı kaotik sistemin FPGA üzerinde 32-bit gerçekleştirilmesinden elde edilen faz portreleri verilerek karşılaştırılmıştır. Şekillerden de görüleceği üzere FPGA üzerinde donanım tabanlı gerçekleştirilen tasarımın oldukça başarılı olduğu gözükmemektedir. Diğer tüm tasarımlardan aynı başarılı modelleme sonuçları elde edilmiştir.



Şekil 12. Matlab üzerinde Euler-tabanlı kaotik osilatör ünitesi faz portreleri



Şekil 13. FPGA üzerinde Euler-tabanlı kaotik osilatör ünitesi faz portreleri

4. TARTIŞMA

FPGA üzerinde sabit noktalı sayı formatında tasarlanan kaotik sistemin, literatürdeki kayan noktalı sayı formatındaki benzer tasarımlara göre daha başarılı olduğu görülmektedir. Ayrıca FPGA-tabanlı kaotik üzerinden alınan ayırık zamanlı sonuçlar Matlab programından alınan sürekli zamanlı sonuçlarla karşılaştırıldığında başarılı

bir eşleşme olduğu görülmüştür. Literatüre sunulan FPGA üzerinde sabit noktalı sayı tabanlı tasarlanan kaotik osilatör kullanılarak daha az maliyetli ve daha hızlı işlem gücü gerektiren kriptografi, güvenli haberleşme ve rasgele sayı üreticileri gibi gömülü kaos tabanlı mühendislik uygulamaları gerçekleştirilebilir. İleride FPGA üzerinde sabit nokta sayı formatında ayırık zamanlı olarak gerçekleştirilen kaotik osilatör kullanılarak kaos tabanlı senkronizasyon, bilgi gizleme ve rasgele sayı üretimi gibi çeşitli uygulamalar gerçekleştirilebilir.

KAYNAKLAR

- Akgul, A., Calgan, H., Koyuncu, I., Pehlivan, I., Istanbulu, A., 2015. Chaos-based engineering applications with a 3D chaotic system without equilibrium points. *Nonlinear Dyn.* 84, 481–495. <https://doi.org/10.1007/s11071-015-2501-7>
- Alcin, M., Koyuncu, I., Tuna, M., Varan, M., Pehlivan, I., 2018. A novel high speed Artificial Neural Network-based chaotic True Random Number Generator on Field Programmable Gate Array. *Int. J. Circuit Theory Appl.* <https://doi.org/10.1002/cta.2581>
- Çavuşoğlu, Ü., Uyaroglu, Y., Pehlivan, İ., 2014. Design of A Continuous-Time Autonomous Chaotic Circuit and Application of Signal Masking. *J. Fac. Eng. Archit. Gazi Univ.* 29, 79–87. <https://doi.org/10.17341/gummfd.73592>
- Çavuşoğlu, Ü., Akgül, A., Kaçar, S., Pehlivan, İ., Zengin, A., 2016. A novel chaos-based encryption algorithm over TCP data packet for secure communication. *Secur. Commun. Networks* 9, 1285–1296. <https://doi.org/10.1002/sec.1414>
- Çiçek, S., Uyaroğlu, Y., Pehlivan, İ., 2013. Simulation and Circuit Implementation of Sprott Case H Chaotic System and Its Synchronization Application for Secure Communication Systems. *J. Circuits, Syst. Comput.* 22, 1350022. <https://doi.org/10.1142/S0218126613500229>
- Ergun, S., Ozoguz, S., 2007. A Chaos-Modulated Dual Oscillator-Based Truly Random Number Generator, in: 2007 IEEE International Symposium on Circuits and Systems. IEEE, pp. 2482–2485. <https://doi.org/10.1109/ISCAS.2007.378742>
- Kaçar, S., 2016. Analog circuit and microcontroller based RNG application of a new easy realizable 4D chaotic system. *Opt. - Int. J. Light Electron Opt.* 127, 9551–9561. <https://doi.org/10.1016/j.ijleo.2016.07.044>
- Koyuncu, I., 2018. Implementation of High Speed Tangent Sigmoid Transfer Function Approximations for Artificial Neural Network Applications on FPGA. *Adv. Electr. Comput. Eng.* 18, 79–86. <https://doi.org/10.4316/AECE.2018.03011>
- Koyuncu, I., Alcin, M., Pehlivan, I., 2013. Electronic circuit realization and synchronization application of Sprott 94 S chaotic system for secure communication systems, in: 2013 21st Signal Processing and Communications Applications Conference (SIU). IEEE, pp. 1–4. <https://doi.org/10.1109/SIU.2013.6531315>
- Koyuncu, I., Ozcerit, A.T., Pehlivan, I., 2014. Implementation of FPGA-based real time novel chaotic oscillator. *Nonlinear Dyn.* 77, 49–59. <https://doi.org/10.1007/s11071-014-1272-x>
- Koyuncu, I., Ozcerit, A.T., Pehlivan, I., 2013. An analog circuit design and FPGA-based implementation of the Burke-Shaw chaotic system. *Optoelectron. Adv. Materials-Rapid Communications* 7, 635–638.
- Lai, Q., Zhao, X.-W., Rajagopal, K., Xu, G., Akgul, A., Guleryuz, E., 2018. Dynamic analyses, FPGA implementation and engineering applications of multi-butterfly chaotic attractors generated from generalised Sprott C system. *Pramana* 90, 6. <https://doi.org/10.1007/s12043-017-1493-x>
- Pehlivan, İ., Uyaroğlu, Y., 2012. A new 3D chaotic system with golden proportion equilibria: Analysis and electronic circuit realization. *Comput. Electr. Eng.* 38, 1777–1784. <https://doi.org/10.1016/j.compeleceng.2012.08.007>
- Pehlivan, İ., Wei, Z., 2012. Analysis, nonlinear control, and chaos generator circuit of another strange chaotic system. *Turkish J. Electrical Engineering Comput. Sciences* 20, 1229–1239.
- Rajagopal, K., Akgul, A., Jafari, S., Karthikeyan, A., Koyuncu, I., 2017. Chaotic chameleon: Dynamic analyses, circuit implementation, FPGA design and fractional-order form with basic analyses. *Chaos, Solitons & Fractals* 103, 476–487. <https://doi.org/10.1016/J.CHAOS.2017.07.007>
- Sadoudi, S., Azzaz, M.S., Djeddou, M., Benssalah, M., 2009. An FPGA Real-time Implementation of the Chen's Chaotic System for Securing Chaotic Communications. *Int. J. Nonlinear Sci.* 7, 1749–3889.
- Tavas, V., Demirkol, A.S.S., Ozoguz, S., Kilinc, S., Toker, A., Zeki, A., Kılınç, S., Toker, A., Zeki, A., 2010. An IC random number generator based on Chaos, in: International Conference on Applied Electronics (AE). Pilsen,

pp. 1–4.

- Tuna, M., Fidan, C.B., 2018. A Study on the importance of chaotic oscillators based on FPGA for true random number generating (TRNG) and chaotic systems. *J. Fac. Eng. Archit. Gazi Univ.* 2018, 1–1.
- Tuna, M., Fidan, C.B., 2016. Electronic circuit design, implementation and FPGA-based realization of a new 3D chaotic system with single equilibrium point. *Opt. - Int. J. Light Electron Opt.* 127, 11786–11799. <https://doi.org/10.1016/j.ijleo.2016.09.087>
- Tuna, M., Fidan, C.B., Koyuncu, I., Pehlivan, I., 2016. Real time hardware implementation of the 3D chaotic oscillator which having golden-section equilibra, in: 2016 24th Signal Processing and Communication Application Conference (SIU). IEEE, pp. 1309–1312. <https://doi.org/10.1109/SIU.2016.7495988>
- Tuna, M., Koyuncu, I., Fidan, C.B., Pehlivan, I., 2015. Real time implementation of a novel chaotic generator on FPGA, in: 2015 23rd Signal Processing and Communications Applications Conference (SIU). IEEE, pp. 698–701. <https://doi.org/10.1109/SIU.2015.7129921>
- Tuna, M., Koyuncu, I., Alçın, M., 2018. Fixed and Floating point-Based High-Speed Chaotic Oscillator Design with Different Numerical Algorithms on FPGA. *Int. J. Adv. Res. Electr. Electron. Instrum. Eng.* 7, 3179–3187. <https://doi.org/10.15662/IJAREEIE.2018.0707014>