

**KAOS TABANLI ÇİFT ENTROPİ ÇEKİRDEKLİ
GERÇEK RASGELE SAYI ÜRETECİ TASARIMI VE
FPGA ÜZERİNDE GERÇEKLEŞTİRİLMESİ**

**2017
DOKTORA TEZİ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ**

Murat TUNA

**KAOS TABANLI ÇİFT ENTROPİ ÇEKİRDEKLİ GERÇEK RASGELE SAYI
ÜRETECİ TASARIMI VE FPGA ÜZERİNDE GERÇEKLEŞTİRİLMESİ**

Murat TUNA

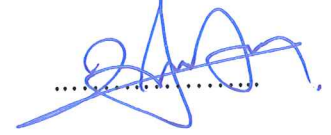
**Karabük Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik- Elektronik Mühendisliği Anabilim Dalında
Doktora Tezi
Olarak Hazırlanmıştır**

**KARABÜK
Temmuz 2017**

Murat TUNA tarafından hazırlanan “KAOS TABANLI ÇİFT ENTROPİ ÇEKİRDEKLİ GERÇEK RASGELE SAYI ÜRETECİ TASARIMI VE FPGA ÜZERİNDE GERÇEKLEŞTİRİLMESİ” başlıklı bu tezin Doktora Tezi olarak uygun olduğunu onaylarım.

Yrd. Doç. Dr. Can Bülent FİDAN

Tez Danışmanı, Mekatronik Mühendisliği Anabilim Dalı



Bu çalışma, jürimiz tarafından oy birliği ile Elektrik Elektronik Mühendisliği Anabilim Dalında Doktora tezi olarak kabul edilmiştir. 24/07/2017

Ünvanı, Adı SOYADI (Kurumu)

İmzası

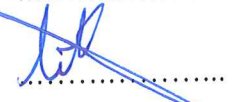
Başkan : Doç. Dr. Mustafa AKTAŞ (KBÜ)

Üye : Doç. Dr. İhsan PEHLİVAN (SAÜ)

Üye : Yrd. Doç. Dr. Can Bülent FİDAN (KBÜ)

Üye : Yrd. Doç. Dr. Ahmet Reşit KAVSAOĞLU (KBÜ)

Üye : Yrd. Doç. Dr. İsmail KOYUNCU (AKÜ)



...../...../2017

KBÜ Fen Bilimleri Enstitüsü Yönetim Kurulu, bu tez ile, Doktora derecesini onamıştır.

Prof. Dr. Filiz ERSÖZ

Fen Bilimleri Enstitüsü Müdürü V.



“Bu tezdeki tüm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiğini ve sunulduğunu; ayrıca bu kuralların ve ilkelerin gerektirdiği şekilde, bu çalışmadan kaynaklanmayan bütün atıfları yaptığımı beyan ederim.”


Murat TUNA

ÖZET

Doktora Tezi

KAOS TABANLI ÇİFT ENTROPİ ÇEKİRDEKLİ GERÇEK RASGELE SAYI ÜRETECİ TASARIMI VE FPGA ÜZERİNDE GERÇEKLEŞTİRİLMESİ

Murat TUNA

Karabük Üniversitesi

Fen Bilimleri Enstitüsü

Elektrik Elektronik Mühendisliği Anabilim Dalı

Tez Danışmanı:

Yrd. Doç. Dr. Can Bülent FİDAN

Temmuz 2017, 171 sayfa

Bu tez çalışmasında, FPGA üzerinde 4 farklı nümerik algoritma kullanılarak tasarımı gerçekleştirilen 2 adet yeni kaotik sistem ile 1 adet ring osilatör yapıları kullanılarak, yüksek çalışma ve bit üretim hızına sahip kaos tabanlı çift entropi çekirdekli Gerçek Rasgele Sayı Üreteçleri (GRSÜ) gerçekleştirilmiştir.

Bu çalışma doğrultusunda, ilk olarak son yıllarda literatüre sunulan ve üzerinde FPGA ile çalışma yapılmamış farklı matematiksel özelliklere sahip 8 adet kaotik sistem seçilmiştir. Bu sistemlerin matematiksel denklem yapıları incelenerek bilgisayar ortamında nümerik çözümleri elde edilmiştir. Ayrıca kaotik sistemlerin zaman serileri, faz portreleri ve Lyapunov üstelleri gibi kaotik dinamik analizleri gerçekleştirilmiştir.

İkinci aşamada; referans olarak seçilen Pehlivan C Kaotik Sistemi (PCKS) ve Pehlivan 3 Boyutlu Kaotik Sistemi (P3BKS) dört farklı nümerik metot ile modellenmiş ve sistemlerin dinamik davranışları incelenerek kaos analizleri yapılmıştır. Yine bu aşamada PCKS ve P3BKS tasarımları FPGA üzerinde donanım tanımlama dili VHDL ile 32 bit IQ-Math sabit noktalı sayı standardına uygun olarak modellenmiştir. Modelleme aşamasında Euler, Heun, RK4 ve RK5-Butcher olmak üzere dört farklı algoritma kullanılmıştır. Xilinx Virtex-6 ailesi XC6VLX75T-3FF784 nolu çipi kullanılan tasarımlar Xilinx ISE Design Tools 14.2 benzetim programı kullanılarak sentezlenmiştir. Ayrıca bu aşamada FPGA üzerinde VHDL dili kullanılarak ring osilatör tasarımı gerçekleştirilmiştir. Kaotik ve ring osilatör tasarımlarının FPGA çip kaynak kullanımına ve ünitelerin saat hızlarına ait parametrelerinin istatistikleri incelenmiştir. Elde edilen sonuçlara göre ring osilatörün çalışma frekansı 1,5 GHz, kaotik osilatörlerin çalışma frekansı 390-464 MHz dolaylarında değişmektedir. Bu aşamada tasarımlardan elde edilen sonuçlar literatürdeki benzer çalışmalarla karşılaştırılmıştır.

Üçüncü aşamada; Euler, Heun, RK4 ve RK5-Butcher algoritmaları kullanılarak FPGA-tabanlı PCKS ve P3BKS tasarımları ile ring osilatör yapısı birleştirilerek çift entropi çekirdekli yeni GRSÜ tasarımı gerçekleştirilmiştir. Genel olarak iki farklı kaotik sistem, dört farklı nümerik algoritma ve bir kuantalama işlemi ile toplamda 8 farklı GRSÜ ünitesi tasarlanmıştır. FPGA tabanlı GRSÜ modellemelerinde 32 bit IQ-Math sabit noktalı sayı standardı kullanılmıştır. Geliştirilen modeller VHDL donanım tanımlama dili kullanılarak kodlanmıştır. Tasarımı yapılan 8 farklı GRSÜ ünitesi Xilinx firmasının ürettiği Virtex-6 ailesinin XC6VLX75T-3FF784 çipi için sentezlenerek, FPGA çip kaynak kullanımına ve ünitelerin saat hızlarına ait parametrelerin istatistikleri incelenmiştir. GRSÜ ünitelerinin verileri işleme süreleri Xilinx ISE Design Tools 14.2 simülasyon programı kullanılarak elde edilmiştir. Tasarlanan yeni GRSÜ üniteleri sistemin belirtilen çalışma frekansında yaklaşık olarak 390-464 Mbit/s arasında değişen yüksek bit üretim hızları elde edilmiştir.

Son aşamada; FPGA üzerinde kaos tabanlı çift entropi çekirdekli tasarlanan 8 adet GRSÜ'lerinden elde edilen sayı dizilerinin rasgelelik testleri gerçekleştirilmiştir. Test aşamasında literatürde yoğun bir şekilde kullanılan uluslararası FIPS 140-1 ve NIST

800-22 test paketleri kullanılmıştır. Tasarımı gerçekleştirilen 8 adet GRSÜ ünitesinden alınan 20 Kbit sayı dizileri FIPS testlerine, 1 Mbit sayı dizileri NIST testlerine tabi tutulmuştur. Tüm tasarımlar rasgelelik testlerinden son işlem uygulamasına gerek kalmadan geçerek, sistemin çalışma frekansı ile aynı değerlerde yüksek bit üretim hızları (390-464 Mbit/s) elde edilmiş ve buda önerilen yeni tasarımın özgün bir çalışma olduğunu göstermiştir.

Kaotik osilatörlerin gürültü benzeri özellikler taşımaları ve bilgi işaretini gizleyebilme gibi özelliklerinden dolayı kaos tabanlı GRSÜ yapıların geliştirilmesi üzerine son yıllarda büyük çabalar sarf edilmektedir. Rasgele sayı üreticileri kriptografi, Monte-Carlo metodunun kullanıldığı uygulamalar, bilgisayar benzetimleri ve modellemeleri ile sayısal analiz uygulamaları gibi birçok alanda kullanılmaktadır. Bu tez çalışmasında FPGA üzerinde kaos tabanlı çift entropi çekirdekli tasarımı gerçekleştirilen yeni GRSÜ ünitelerinden elde edilen yüksek çalışma ve bit üretim hızına sahip rasgele sayı dizileri; hızlı, güvenli ve yoğun işlem gerektiren kriptografi ve güvenli haberleşme alanlarında kullanılabileceklerdir.

Anahtar Sözcükler : Gerçek rasgele sayı üretici, kaos, kaotik osilatör, nümerik algoritma, FPGA, NIST 800-22, FIPS 140-1.

Bilim Kodu : 905.1.179

ABSTRACT

Ph. D. Thesis

CHAOS-BASED DUAL ENTROPY CORE TRUE RANDOM NUMBER GENERATOR DESIGN AND ITS REALIZATION ON FPGA

Murat TUNA

Karabük University

Graduate School of Natural and Applied Sciences

Department of Electrical - Electronics Engineering

Thesis Advisor:

Assist. Prof. Dr. Can Bülent FİDAN

July 2017, 171 pages

In this thesis study, chaos-based dual entropy core True Random Number Generators (TRNG) with high operation capacities and bit generation speeds were realized by using 2 new chaotic systems and 1 ring oscillator structures that are designed on FPGA by using 4 different numerical algorithms.

In the direction of this study, firstly, 8 chaotic systems having different mathematical characteristics, which have not been studied on FPGA and have been presented to the literature in recent years, were selected. Numerical solutions were obtained in computer environment by examining the mathematical equation structures of these systems. In addition, chaotic dynamic analyzes such as time series, phase portraits and Lyapunov exponents of chaotic systems, were realization.

In the second phase; The Pehlivan C Chaotic System (PCCS) and the Pehlivan 3D Chaotic System (P3DCS), which were selected as reference, were modeled by four different numerical methods and chaos analyzes were carried out by analyzing the dynamic behaviors of the systems. Again in this phase, the PCCS and P3DCS designs were modeled on the FPGA in accordance the 32 bit IQ-Math fixed point number standard with hardware description language VHDL. Four different algorithms as Euler, Heun, RK4 and RK5-Butcher were used in the modeling stage. The designs, in which the XC6VLX75T-3FF784 numbered chip of Xilinx Virtex-6 family were utilized, were synthesized using the Xilinx ISE Design Tools 14.2 simulation program. In addition, a ring oscillator design was actualized on the FPGA by using the VHDL language. The statistics of parameters of chaotic and ring oscillator designs belonging to FPGA chip source usage and clock rates of the units have been examined. According to the obtained results, the operating frequency of the ring oscillator is changed around 1,5 GHz, and the operating frequency of the chaotic oscillator is changed around 390-464 MHz. At this phase, the results obtained from the designs are compared with the results of the similar studies in the literature.

In the third phase; new dual entropy core TRNG design was actualized by combining the FPGA-based PCCS and P3DCS designs with the ring oscillator structure, and by using Euler, Heun, RK4 and RK5-Butcher algorithms. In generally, 8 different TRNG units were designed with two different chaotic systems, four different numerical algorithms and a quantization operation. While modeling FPGA-based TRNGs, 32 bit IQ-Math fixed-point number standard was used. The developed models were coded using the VHDL hardware description language. 8 different TRNG units, which were designed, were synthesized for the XC6VLX75T-3FF784 chip of the Virtex-6 family produced by Xilinx Company, and the statistics of parameters belonging to FPGA chip source usage and clock rates of the units were analyzed. Data processing durations of TRNG units were obtained by using the Xilinx ISE Design Tools 14.2 simulation program. High bit production rates of about 390-464 Mbit/s were obtained at the specified operating frequency of the systems of new designed TRNGs.

In the last stage; randomness tests of number sequences obtained from the 8 chaos-based dual entropy core TRNGs designed on FPGA were carried out. In the test phase, international FIPS 140-1 and NIST 800-22 test packages, which are intensively used in the literature, were utilized. The 20 Kbit number sequences obtained from the designed 8 TRNG units were subjected to FIPS tests, and the 1 Mbit number sequences were subjected to NIST tests. All designs passed through randomness tests without any need for the application of finishing, high bit production rates (390-464 Mbit/s) having the same values with the operating frequency of the system was obtained, which shows that the proposed new design is a unique study.

Due to the certain characteristics of chaotic oscillators such as having noise and their ability to hide the informatory signs, a massive amount of effort for the development of chaos-based TRNG structures have been spent in the recent years. Random number generators have been utilized in numerous fields such as cryptography, applications using Monte-Carlo method, computer simulations and modeling and numerical analysis applications. In this thesis study, random number arrays with high operation and bit production rate obtained from new chaos-based dual entropy core TRNG units, which are designed on FPGA; will possibly be used in cryptography and secure communication areas that require fast, secure and intensive operations.

Key Word : True random number generator, chaos, chaotic oscillator, numerical method, FPGA, NIST 800-22, FIPS 140-1.

Science Code : 905.1.179

TEŞEKKÜR

Tüm hayatım boyunca her daim yanımda olan, İslamiyet ve iman gibi sonsuz nimetlerle nimetlendirerek kainatın sultanına ümmet olma şerefini bizlere sunan yüce Rabbime en içten samimiyetimle hamd ü senalar olsun.

Bu tez çalışmasının planlanmasında, araştırılmasında, yürütülmesinde ve oluşumunda ilgi ve desteğini esirgemeyen, engin tecrübelerinden yararlandığım, yönlendirme ve bilgilendirmeleriyle çalışmamı bilimsel temeller ışığında şekillendiren sayın danışman hocam Yrd. Doç. Dr. Can Bülent FİDAN'a sonsuz teşekkürlerimi sunarım.

Doktora çalışması boyunca sürekli yardımcı olan, karşılaştığım zorluklarda destek vererek tüm bilgi ve birikimini benimle paylaşarak bu çalışmada sona gelmemi sağlayan çok değerli arkadaşım Afyon Kocatepe Üniversitesi öğretim üyesi Yrd. Doç. Dr. İsmail KOYUNCU'ya en içten teşekkürlerimi sunarım.

Konu içerisindeki katkılarıyla ve yönlendirmeleriyle her zaman yardımlarını esirgemeyen Sakarya Üniversitesi öğretim üyesi Doç. Dr. İhsan PEHLİVAN'a çok teşekkür ederim. Ayrıca ders ve tez dönemlerinde maddi ve manevi katkılarıyla bana destek olan Karabük Üniversitesi öğretim üyesi Doç. Dr. Mustafa AKTAŞ'a da minnet duyuyorum.

Sevgili eşim Öznur TUNA'ya ve oğullarım Ömer Said ile Yusuf Enes'e maddi ve manevi hiçbir yardımı esirgmeden her zaman yanımda oldukları için tüm ruhu kalbimle teşekkür ederim. Ayrıca hayatım boyunca her türlü desteklerini esirgemeyen anneme, babama, kardeşlerime ve üzerimde emeği olan herkese teşekkürlerimi sunarım.

İÇİNDEKİLER

	<u>Sayfa</u>
KABUL	ii
ÖZET	iv
ABSTRACT	vii
TEŞEKKÜR	x
İÇİNDEKİLER	xi
ŞEKİLLER DİZİNİ	xv
ÇİZELGELER DİZİNİ	xx
SİMGELER VE KISALTMALAR DİZİNİ	xxii
BÖLÜM 1	1
GİRİŞ	1
1.1. TEZİN AMACI	5
1.2. TEZİN ÖZGÜN DEĞERİ	6
1.3. TEZDE İZLENECEK YOL	8
BÖLÜM 2	10
KONU VE KAPSAM	10
2.1. KAOS VE KAOTİK SİSTEMLER	10
2.2. RING (HALKA) OSİLATÖRLER	12
2.3. FPGA TEKNOLOJİSİ	15
2.4. GERÇEK RASGELE SAYI ÜRETEÇLERİ (GRSÜ)	20
2.4.1. Gürültü İşaretinin Doğrudan Kuvvetlendirilmesi	21
2.4.2. Çift Osilatör Örneklemeye Yöntemi	22
2.4.3. Kaotik Devre Tabanlı	23
BÖLÜM 3	26
KAOS ANALİZ YÖNTEMLERİ VE LİTERATÜRE SON YILLARDA YENİ SUNULAN ÜZERİNDE FPGA TABANLI ÇALIŞMA YAPILMAMIŞ KAOTİK SİSTEMLER	26

3.1. KAOTİK ANALİZ YÖNTEMLERİ.....	26
3.2. SON YILLARDAKİ LİTERATÜRDEKİ YENİ KAOTİK SİSTEMLER....	29
3.2.1. Yeni Pehlivan C Kaotik Sistemi (PCKS) (2007).....	30
3.2.2. Yeni Pehlivan 3 Boyutlu Kaotik Sistemi (P3BKS) (2012)	33
3.2.3. Yeni Abooe 3 Boyutlu Kaotik Sistemi (2013).....	36
3.2.4. Yeni Deng 3 Boyutlu Kaotik Sistemi (2014)	39
3.2.5. Yeni Zhoua Kaotik Sistemi (2008).....	42
3.2.6. Yeni Pehlivan 3 Boyutlu Kelebek Şeklinde Kaotik Çekicisi (2014).....	45
3.2.7. Yeni Wei Hiperkaotik Sistemi (2013)	47
3.2.8. Yeni Pehlivan 3 Boyutlu 4 Sarmallı Kaotik Sistemi (2011)	50
 BÖLÜM 4	54
REFERANS KAOTİK SİSTEMLERİN NUMERİK MODELLERİ İLE FPGA ÜZERİNDE KAOTİK OSİLATÖR TASARIMI	54
4.1. REFERANS KAOTİK SİSTEMLERİN NÜMERİK MODELLERİ	54
4.1.1. PCKS Tasarımı	54
4.1.1.1. PCKS Tasarımının Nümerik Yöntemlerle Modellenmesi	55
4.1.2. P3BKS Tasarımı	58
4.1.2.1. P3BKS Tasarımının Nümerik Yöntemlerle Modellenmesi	58
4.2. REFERANS KAOTİK SİSTEMLERİN AYRIKLAŞTIRILMIŞ MODELLERİ.....	61
4.3. REFERANS KAOTİK SİSTEMLERİN FPGA ÜZERİNDE GERÇEKLEŞTİRİLMESİ	65
4.4. FPGA TABANLI KAOTİK OSİLATÖRLERİN TEST SONUÇLARI.....	70
4.4.1. FPGA Tabanlı PCKS Tasarımının Gerçekleştirilmesi	70
4.4.2. FPGA Tabanlı P3BKS Tasarımının Gerçekleştirilmesi	74
4.5. FPGA TABANLI RİNG OSİLATÖR TASARIMI VE TEST SONUÇLARI	81
4.6. LİTERATÜRDE FPGA TABANLI TASARIMI GERÇEKLEŞTİRİLEN KAOTİK OSİLATÖR TASARIMLARI	82
 BÖLÜM 5	85
KAOS TABANLI ÇİFT ENTROPİ ÇEKİRDEKLİ GERÇEK RASGELE SAYI ÜRETECİ TASARIMI VE FPGA ÜZERİNDE GERÇEKLEŞTİRİLMESİ	85

5.1. LİTERATÜRDEKİ GRSÜ TASARIMLARI	87
5.2. KAOS TABANLI ÇİFT ENTROPİ ÇEKİRDEKLİ GERÇEK RASGELE SAYI ÜRETECİ TASARIMI VE FPGA ÜZERİNDE GERÇEKLEŞTİRİLMESİ	96
5.2.1. FPGA-Tabanlı Gerçek Rasgele Sayı Üreteci Tasarımı	97
5.2.1.1. Sabit Noktalı Sayı (SNS) Tabanlı GRSÜ	97
5.2.1.2. Gerçek Rasgele Sayı Üretecinin FPGA Üzerinde Gerçekleştirilmesi	98
 BÖLÜM 6	102
FPGA ÜZERİNDE GERÇEKLEŞTİRİLEN KAOS TABANLI ÇİFT ENTROPİ ÇEKİRDEKLİ GERÇEK RASGELE SAYI ÜRETECLERİNİN RASGELELİK TEST SONUÇLARI	102
6.1. RASTGELELİK TESTLERİ	102
6.1.1. FIPS 140-1 Testi	103
6.1.1.1. Monobit Testi (Monobit Test)	103
6.1.1.2. Poker Testi (Poker Test)	103
6.1.1.3. Koşu Testi (Run Test)	103
6.1.1.4. Uzun Koşu Testi (Long Runs Test)	104
6.1.2. NIST 800-22 Testi	104
6.1.2.1. Frekans (Monobit) Testi	107
6.1.2.2. Blok Frekans Testi	109
6.1.2.3. Akış Testi	110
6.1.2.4. Blok içindeki Birlerin En Uzun Akış Testi	111
6.1.2.5. İkili Matris Rankı Testi	114
6.1.2.6. Ayrık Fourier Dönüşüm Testi	116
6.1.2.7. Örtüşmeyen Şablon Eşleştirme Testi	117
6.1.2.8. Örtüşen Şablon Eşleştirme Testi	120
6.1.2.9. Maurer “Evrensel İstatistik” Testi	121
6.1.2.10. Lempel-Ziv Sıkıştırma Testi	124
6.1.2.11. Doğrusal Karmaşıklık Testi	125
6.1.2.12. Seri Testi	127
6.1.2.13. Yaklaşık Entropi Testi	129
6.1.2.14. Kümülatif Toplamlar Testi	131

	<u>Sayfa</u>
6.1.2.15. Rasgele Gezinimler Testi	133
6.1.2.16. Rasgele Gezinimler Değişken Testi	136
6.2. TASARLANAN GRSÜ FIPS 140-1 TEST SONUÇLARI	138
6.3. TASARLANAN GRSÜ NIST 800-22 TEST SONUÇLARI.....	139
 BÖLÜM 7	 144
SONUÇLAR VE ÖNERİLER	144
KAYNAKLAR	148
 EK AÇIKLAMALAR A. XILINX VİRTEX-6 FPGA AİLESİ ÖZELLİKLER.....	 163
 ÖZGEÇMİŞ	 171

ŞEKİLLER DİZİNİ

Sayfa

Şekil 1.1. Ayrık zamanlı önerilen kaos tabanlı çift entropi çekirdekli yeni GRSÜ...	6
Şekil 2.1. Tek sayılı evirici içeren ring osilatör yapısı	13
Şekil 2.2. Farklı yapıda ring osilatör yapıları	14
Şekil 2.3. Xilinx firmasının Virtex-4 FPGA çipinin görünümü.	16
Şekil 2.4. Bir FPGA çipinin genel yapısı.....	17
Şekil 2.5. Geleneksel GRSÜ blok şeması.....	20
Şekil 2.6. Çiçek ve arkadaşları tarafından GRSÜ'lerin detaylı sınıflandırılması. ...	21
Şekil 2.7. Gürültünün doğrudan kuvvetlendirilmesi yöntemi.....	21
Şekil 2.8. GRSÜ kullanılan çift osilatör örnekleme yapısı.....	23
Şekil 2.9. Kaotik rastgele sayı üreten sistemin blok diyagramı.....	24
Şekil 2.10. Örnek bir kaotik devrenin x-y faz portresinin Matlab ve osilaskop çıkışı.....	24
Şekil 3.1. Yeni PCKS'nin x, y, z kaotik durum değişkenlerinin zamana göre değişimi.....	32
Şekil 3.2. Yeni PCKS tasarımının x-y, x-z, y-z kaotik çekicileri ve üç boyutlu x-y-z yörüngesi	32
Şekil 3.3. Yeni tasarlanan PCKS'nin Lyapunov üstelleri.....	33
Şekil 3.4. Pehlivan vd. tarafından tasarlanan yeni P3BKS'nin x, y, z ve x-y-z zaman serileri.....	34
Şekil 3.5. Pehlivan vd. tarafından tasarlanan yeni P3BKS'nin x-y, x-z, y-z faz portleri ve 3 boyutlu x-y-z faz yörüngesi	35
Şekil 3.6. Pehlivan vd. tarafından tasarlanan yeni P3BKS tasarımının Lyapunov üstelleri.	35
Şekil 3.7. Abooe vd. tarafından tasarlanan yeni kaotik sistemin x, y, z ve x-y-z zaman serileri.....	37
Şekil 3.8. Önerilen yeni kaotik sistemin oldukça benzer başlangıç koşullarındaki $x(t)$, $y(t)$ ve $z(t)$ zaman serileri cevabı.	37
Şekil 3.9. Abooe vd. tasarladıkları yeni kaotik sistemin x-y, x-z ve y-z kaotik çekicileri ve üç boyutlu x-y-z yörüngesi.	38
Şekil 3.10. Abooe vd. tasarladıkları yeni kaotik sisteminin Lyapunov üstelleri.....	38

Şekil 3.11. Deng yeni kaotik sisteminin x, y, z kaotik durum değişkenlerinin zamana göre değişimi.	40
Şekil 3.12. Deng vd. tarafından önerilen yeni kaotik sistemin benzer başlangıç koşullarındaki $x(t), y(t)$ ve $z(t)$ zaman serileri cevabı.....	40
Şekil 3.13. Deng vd. tasarladıkları yeni kaotik sisteminin $x-y, x-z, y-z$ kaotik çekicileri ve üç boyutlu $x-y-z$ yörüngesi.	41
Şekil 3.14. Deng vd. tarafından tasarlanan yeni kaotik sisteminin Lyapunov Üstelleri.....	41
Şekil 3.15. Zhoua yeni kaotik sisteminin x, y, z kaotik durum değişkenlerinin zamana göre değişimi.	43
Şekil 3.16. Zhoua yeni kaotik sisteminin $x-y, x-z, y-z$ kaotik çekicileri ve üç boyutlu $x-y-z$ yörüngesi.	43
Şekil 3.17. Zhoua yeni kaotik sisteminin “ d ” kontrol parametresinin değişimine ($d=1,6, d=-1,6, d=1,8$) göre kaos durumu.....	44
Şekil 3.18. Zhoua yeni kaotik sisteminin Lyapunov üstelleri.....	45
Şekil 3.19. Pehlivan yeni kaotik sisteminin x, y, z kaotik durum değişkenlerinin zamana göre değişimi.	46
Şekil 3.20. Pehlivan vd. tarafından tasarlanan kelebek şeklinde 3 boyutlu yeni kaotik sisteminin $x-y, x-z, y-z$ kaotik çekicileri ve üç boyutlu $x-y-z$ yörüngesi.....	46
Şekil 3.21. Pehlivan vd. tarafından tasarlanan kaotik sisteminin Lyapunov üstelleri.	47
Şekil 3.22. Wei kaotik sisteminin x, y, z kaotik durum değişkenlerinin zamana göre değişimi.	48
Şekil 3.23. Wei yeni kaotik sisteminin $x-y, x-z, y-z, x-w, y-w$ ve $z-w$ 2 boyutlu kaotik faz portreleri.	49
Şekil 3.24. Wei yeni kaotik sisteminin 3 boyutlu $x-y-z, x-y-w$ ve $y-z-w$ faz yörüngeleri.	49
Şekil 3.25. Wei yeni hiper-kaotik sisteminin Lyapunov üstelleri.....	50
Şekil 3.26. Pehlivan 4 sarmallı yeni kaotik sisteminin x, y, z kaotik durum değişkenlerinin zamana göre değişimi.....	51
Şekil 3.27. Pehlivan 4 sarmallı yıldız şeklinde yeni kaotik sisteminin 2 boyutlu $x-y, x-z, y-z$ faz portreleri.	52
Şekil 3.28. Pehlivan 4 sarmallı yeni kaotik sisteminin 3 boyutlu faz portresi.....	52
Şekil 3.29. Pehlivan 4 sarmallı yeni kaotik sistemin $x_0=0,6, y_0=0, z_0=0$ başlangıç koşulları ve $a=4, b=0,50, c=0,60$ parametre değerlerinde Lyapunov üstelleri.	53
Şekil 4.1. PCKS'nin Euler nümerik modeli x, y, z zaman serileri.	55
Şekil 4.2. PCKS'nin Heun nümerik modeli x, y, z zaman serileri.	56

Şekil 4.3. PCKS'nin RK4 nümerik modeli x, y, z zaman serileri.....	56
Şekil 4.4. PCKS'nin RK5-Butcher nümerik modeli x, y, z zaman serileri.....	56
Şekil 4.5. PCKS'nin Euler nümerik modeli x-y, x-z, y-z faz portreleri.	57
Şekil 4.6. PCKS'nin Heun nümerik modeli x-y, x-z, y-z faz portreleri.	57
Şekil 4.7. PCKS'nin RK4 nümerik modeli x-y, x-z, y-z faz portreleri.....	57
Şekil 4.8. PCKS'nin RK5-Butcher nümerik modeli x-y, x-z, y-z faz portreleri.....	58
Şekil 4.9. P3BKS tasarımının Euler modeli x, y, z zaman serileri.	59
Şekil 4.10. P3BKS tasarımının Heun modeli x, y, z zaman serileri.	59
Şekil 4.11. P3BKS tasarımının RK4 modeli x, y, z zaman serileri.	59
Şekil 4.12. P3BKS tasarımının RK5-Butcher modeli x, y, z zaman serileri.	60
Şekil 4.13. P3BKS tasarımının Euler nümerik modeli x-y, x-z, y-z faz portreleri....	60
Şekil 4.14. P3BKS tasarımının Heun nümerik modeli x-y, x-z, y-z faz portreleri....	60
Şekil 4.15. P3BKS tasarımının RK4 nümerik modeli x-y, x-z, y-z faz portreleri	61
Şekil 4.16. P3BKS tasarımının RK5-Butcher nümerik modeli x-y, x-z, y-z faz portreleri	61
Şekil 4.17. FPGA-tabanlı yeni kaotik osilatör ünitesi en üst seviye blok diyagramı	65
Şekil 4.18. FPGA-tabanlı yeni kaotik osilatör ünitesi ikinci seviye blok diyagramı	66
Şekil 4.19. Euler-tabanlı yeni kaotik osilatör ünitesi üçüncü seviye blok diyagramı	67
Şekil 4.20. Heun-tabanlı yeni kaotik osilatör ünitesi üçüncü seviye blok diyagramı	68
Şekil 4.21. RK4-tabanlı yeni kaotik osilatör ünitesi üçüncü seviye blok diyagramı	69
Şekil 4.22. RK5-Butcher tabanlı yeni kaotik osilatör ünitesi üçüncü seviye blok diyagramı.	70
Şekil 4.23. Euler-tabanlı PCKS osilatör ünitesi Xilinx ISE simülatör sonuçları.....	71
Şekil 4.24. Heun-tabanlı PCKS osilatör ünitesi Xilinx ISE simülatör sonuçları.....	71
Şekil 4.25. RK4-tabanlı PCKS osilatör ünitesi Xilinx ISE simülatör sonuçları.....	71
Şekil 4.26. Euler-tabanlı PCKS osilatör ünitesi FPGA çip kullanım istatistikleri.....	72
Şekil 4.27. Heun-tabanlı PCKS osilatör ünitesi FPGA çip kullanım istatistikleri.....	72
Şekil 4.28. RK4-tabanlı PCKS osilatör ünitesi FPGA çip kullanım istatistikleri.....	72
Şekil 4.29. FPGA üzerinde gerçekleştirilen Euler tabanlı PCKS'nin x(t), y(t), ve z(t) zaman serileri	73

Şekil 4.30. FPGA üzerinde Euler-tabanlı PCKS osilatör ünitesi faz portreleri	73
Şekil 4.31. FPGA üzerinde RK4-tabanlı PCKS osilatör ünitesi faz portreleri	74
Şekil 4.32. Euler-tabanlı P3BKS osilatör ünitesi Xilinx ISE simülatör sonuçları.....	75
Şekil 4.33. Heun-tabanlı P3BKS osilatör ünitesi Xilinx ISE simülatör sonuçları.....	75
Şekil 4.34. RK4-tabanlı P3BKS osilatör ünitesi Xilinx ISE simülatör sonuçları	75
Şekil 4.35. RK5-Butcher tabanlı P3BKS osilatör ünitesi Xilinx ISE simülatör sonuçları.....	76
Şekil 4.36. Euler-tabanlı P3BKS osilatör ünitesi FPGA çip kullanım istatistikleri ..	76
Şekil 4.37. Heun-tabanlı P3BKS osilatör ünitesi FPGA çip kullanım istatistikleri ..	76
Şekil 4.38. RK4-tabanlı P3BKS osilatör ünitesi FPGA çip kullanım istatistikleri....	77
Şekil 4.39. RK5-Butcher tabanlı P3BKS osilatör ünitesi FPGA çip kullanım istatistikleri	77
Şekil 4.40. RK5-Butcher tabanlı P3BKS tasarımını $x(t)$, $y(t)$, ve $z(t)$ zaman serilerinin karşılaştırılması	78
Şekil 4.41. Matlab üzerinde Euler-tabanlı P3BKS osilatör ünitesi faz portreleri.....	79
Şekil 4.42. FPGA üzerinde Euler-tabanlı P3BKS osilatör ünitesi faz portreleri.....	79
Şekil 4.43. FPGA üzerinde RK4-tabanlı P3BKS osilatör ünitesi faz portreleri	80
Şekil 4.44. FPGA-tabanlı Ring osilatör ünitesi en üst seviye blok şeması	81
Şekil 4.45. FPGA üzerinde tasarlanan ring osilatör Xilinx ISE Simülatör sonuçları	82
Şekil 4.46. FPGA üzerinde tasarlanan ring osilatör ünitesi çip kullanım oranları.....	82
Şekil 5.1. Rasgele sayı üreticilerinin sınıflandırılması.....	87
Şekil 5.2. Geleneksel GRSÜ blok şeması.....	87
Şekil 5.3. Ayırık zamanlı önerilen çift entropi çekirdekli yeni GRSÜ tasarımı.	96
Şekil 5.4. IQ-Math sabit noktalı sayı formatı gösterimi.	97
Şekil 5.5. FPGA üzerinde SNS formatında tasarlanan kaos tabanlı çift entropi çekirdekli yeni GRSÜ blok şeması.....	98
Şekil 5.6. Euler algoritması ile tasarlanan PCKS tabanlı çift entropi çekirdekli GRSÜ ünitesi Xilinx ISE Simülatörü sonuçları.	99
Şekil 5.7. Heun algoritması ile tasarlanan PCKS tabanlı çift entropi çekirdekli GRSÜ ünitesi Xilinx ISE Simülatörü sonuçları.	99
Şekil 5.8. RK4 algoritması ile tasarlanan PCKS tabanlı çift entropi çekirdekli GRSÜ ünitesi Xilinx ISE Simülatörü sonuçları.	99
Şekil 5.9. RK5-Butcher algoritması ile tasarlanan PCKS tabanlı çift entropi çekirdekli GRSÜ ünitesi Xilinx ISE Simülatörü sonuçları.	99

Şekil 5.10. Euler algoritması ile tasarlanan P3BKS tabanlı çift entropi çekirdekli GRSÜ ünitesi Xilinx ISE Simülatörü sonuçları.	100
Şekil 5.11. Heun algoritması ile tasarlanan P3BKS tabanlı çift entropi çekirdekli GRSÜ ünitesi Xilinx ISE Simülatörü sonuçları.	100
Şekil 5.12. RK4 algoritması ile tasarlanan P3BKS tabanlı çift entropi çekirdekli GRSÜ ünitesi Xilinx ISE Simülatörü sonuçları.	100
Şekil 5.13. RK5-Butcher algoritması ile tasarlanan P3BKS tabanlı çift entropi çekirdekli GRSÜ ünitesi Xilinx ISE Simülatörü sonuçları.	101
Şekil 6.1. Örnekteki S' kümesi için rasgele yürüyüş sonucu.	133

ÇİZELGLER DİZİNİ

	<u>Sayfa</u>
Çizelge 3.1. Son yıllarda literatürdeki yeni kaotik osilatör tasarımları.....	30
Çizelge 3.2. Kaotik sistemlerinin dinamik özelliklerinin karşılaştırılması	53
Çizelge 4.1. Yeni Kaotik osilatörlerin Xilinx Virtex-6 FPGA çip istatistikleri	74
Çizelge 4.2. Yeni Kaotik osilatörlerin Xilinx Virtex-6 FPGA çip istatistikleri	80
Çizelge 4.3. Literatürde FPGA tabanlı gerçekleştirilen kaotik osilatörler ve teknik özellikleri	84
Çizelge 5.1. Literatürde CMOS tabanlı gerçekleştirilen GRSÜ tasarımları ve teknik özellikleri.	89
Çizelge 5.2. Literatürde FPGA tabanlı Klasik osilatörlerle gerçekleştirilen GRSÜ tasarımları ve teknik özellikleri	93
Çizelge 5.3. Literatürde FPGA üzerinde tasarlanan tamamen sayısal diferansiyel kaotik devre tabanlı GRSÜ.	95
Çizelge 5.4. SNS-Tabanlı PCKS ve P3BKS ile tasarlanan çift entropi çekirdekli GRSÜ ünitelerinin FPGA çip istatistikleri	101
Çizelge 6.1. Koşu testi için blok uzunluklarına göre blok sayıları.....	104
Çizelge 6.2. Önerilen minimum dizi ve blok uzunlukları.	112
Çizelge 6.3. Her bir M-bit blok için en uzun birlerin beklenen akış frekans değerleri	112
Çizelge 6.4. M değerine göre belirlenen K ve N parametre değerleri	112
Çizelge 6.5. Alt bloklara ayrılan en uzun birlerin akış frekanslarının incelenmesi.....	113
Çizelge 6.6. m=3 bitlik B=001 şablonunun M1 ve M2 bloklarının içinde incelenmesi	118
Çizelge 6.7. Önerilen B=11 şablonunun M1 bloğu içinde bulunma durumu	120
Çizelge 6.8. L-bit uzunluğundaki blokların Maurer testine göre gösterimi	122
Çizelge 6.9. Dört başlangıç değeri olası L-bit değerleri	122
Çizelge 6.10. Tekrar bloğu ve test bölümü için L-bit değerleri.	123
Çizelge 6.11. L değerleri için beklenen ve varyans değerleri.	124
Çizelge 6.12. Üretilen bit dizisine göre Lempel-Ziv sözlük oluşumu.	124
Çizelge 6.13. Seçilen örneğin Lempel-Ziv kod çıkış örneği.....	125
Çizelge 6.14. İleri (mode=0) ve geri (mode=1) yönlü metot uygulaması.....	131

Çizelge 6.15. Üretilen ε dizinde oluşan rasgele yürüyüş döngü frekansları	134
Çizelge 6.16. Üretilen ε dizi içerisinde x durum değişkeninin oluşma olasılığı.....	135
Çizelge 6.17. n=106 adet örnekte verilen bit dizisinin test istatistik sonuçları.....	136
Çizelge 6.18. n=106 adet örnekte verilen bit dizisinin test istatistik sonuçları.....	138
Çizelge 6.19. FPGA üzerinde PCKS osilatörün Euler, Heun, RK4 ve RK5-Butcher tabanlı çift entropi çekirdekli GRSÜ ünite tasarımının FIPS 140-1 test sonuçları.	139
Çizelge 6.20. FPGA üzerinde P3BKS osilatörün Euler, Heun, RK4 ve RK5-Butcher tabanlı çift entropi çekirdekli GRSÜ ünite tasarımının FIPS 140-1 test sonuçları.	139
Çizelge 6.21. FPGA üzerinde PCKS osilatörün Euler ve Heun algoritmaları kullanılarak gerçekleştirilen çift entropi çekirdekli GRSÜ NIST 800-22 istatistiksel test sonuçları.	140
Çizelge 6.22. FPGA üzerinde PCKS osilatörün RK4 ve RK5-Butcher algoritmaları kullanılarak gerçekleştirilen çift entropi çekirdekli GRSÜ NIST 800-22 istatistiksel test sonuçları.	141
Çizelge 6.23. FPGA üzerinde P3BKS osilatörün Euler ve Heun algoritmaları kullanılarak gerçekleştirilen çift entropi çekirdekli GRSÜ NIST 800-22 istatistiksel test sonuçları.....	142
Çizelge 6.24. FPGA üzerinde P3BKS osilatörün RK4 ve RK5-Butcher algoritmaları kullanılarak gerçekleştirilen çift entropi çekirdekli GRSÜ NIST 800-22 istatistiksel test sonuçları.	143

SİMGELER VE KISALTMALAR DİZİNİ

SİMGELER

d	: gecikme süresi
$\text{erf}(z)$	: hata fonksiyonu
$\text{erfc}(z)$	: tamamlayıcı hata fonksiyonu
J	: sıfır geçişlerinin toplam sayısı
k	: ring evirici sayısı
Li	: doğrusal karmaşıklık değeri
M	: blok uzunluğu
n	: bit uzunluğu
N	: örtüşmeyen blok sayısı
N_i	: gözlemlenen frekans
Ne_i	: beklenen frekans
P-değeri	: NIST-800-22 testinde rasgelelik ölçütü
$S(\text{obs})$	: gözlemlenen değer
S_n	: rasgele değişkenlerin toplamı
t	: zaman
T	: toplam saat periyodu
T_i	: rasgele değişken değeri
λ	: lyapunov üstelleri
V_i	: akış frekans değerleri
$v_{i_1 \dots i_m}$	: m bitlik $i_1 \dots i_m$ örneklerin frekansı
$v_{i_1 \dots i_{m-1}}$	: m-1 bitlik $i_1 \dots i_{m-1}$ örneklerin frekansı
$v_{i_1 \dots i_{m-2}}$	: m-2 bitlik $i_1 \dots i_{m-2}$ örneklerin frekansı
$V_n(\text{obs})$	: bit osilasyon sayısı
x	: kaotik sistem durum değişkenleri
y	: kaotik sistem durum değişkenleri

z	: kaotik sistem durum deęişkenleri
ρ	: kaotik sistem parametreleri
σ	: kaotik sistem parametreleri
ε	: üretilen bit dizisi
μ	: beklenen deęer
σ^2	: varyans
$\Gamma(x)$	: gama işlevi
χ^2	: ki-kare
Q	: matris sütun sayısı
φ	: kümülatif olasılık dağılım fonksiyonu
ξ	: ziyaret edilen durumların toplam sayısı
$\varphi(m)$	: blokların ampirik dağılım frekansı
$\Phi(z)$	: olasılık yoğunluk fonksiyonu
$\nabla \psi^2(\text{obs})$	: bir dizideki m-bit örneğinin gözlemlenen frekansı
$\nabla^2 \psi^2(\text{obs})$	: bir dizideki m-bit örneğin beklenen frekans deęeri

KISALTMALAR

AFD	: Ayrık Fourier Dönüşümü
ASIC	: Application Specific Integrated Circuit (Uygulamaya Özel Tümdevre)
CAD	: Computer Aided Design (Bilgisayar Destekli Tasarım)
CLB	: Configurable Logic Blocks (Yapılandırılabilir Mantıksal Bloklar)
CMOS	: Complementary Metal Oxide Semiconductor
DGBKK	: Doğrusal Geri Beslemeli Kayan Kaydedici
DSPs	: Digital Signal Processors (Sayısal İşaret İşlemciler)
ECAD	: Electronic Computer Aided Design
ERFC	: The Complementary Error Function
FPAA	: Field Programmable Analog Array
FPGA	: Field Programmable Gate Array (Alan Programlanabilir Kapı Dizileri)
GRSÜ	: Gerçek Rasgele Sayı Üreteçleri
HRSÜ	: Hibrit Rasgele Sayı Üreteçleri
IEEE	: The Institute of Electrical and Electronical Engineers

IEEE 754	: IEEE Floating Point Number (Kayan Noktalı Sayı Formatı)
IOB	: Input/Output Blocks (Giriş Çıkış Blokları)
IQ-Math	: Fixed-Point Number (Sabit-Noktalı Sayı Formatı)
IP-core	: Intellectual Properties Core
ISE	: Integrated Software Environment
IGAMC	: Incomplete Complementary Gamma Function
LET	: Lyapunov Exponent Toolbox
LFSR	: Linear Feedback Shift Register
LUT	: Look-up Table (Değer Tablosu)
P3BKS	: Pehlivan 3 Boyutlu Kaotik Sistemi
PAL	: Programmable Array Logic (Programlanabilir Dizi Mantıkları)
PCKS	: Pehlivan C Kaotik Sistemi
PLA	: Programmable Logic Array (Programlanabilir Mantık Dizisi)
PLL	: Phase Locked Loop (Faz Kilitlemeli Döngü)
RK4	: Dördüncü Dereceden Runge-Kutta Algoritması
RK5	: Beşinci Dereceden Runge-Kutta Algoritması
RO	: Ring Osilatör (Halka Osilatör)
RSÜ	: Rasgele Sayı Üreteci
SNS	: Sabit Noktalı Sayı
SRSÜ	: Sözde Rasgele Sayı Üreteçleri
VHDL	: Very High Speed Integrated Circuit Hardware Description Language (Çok Yüksek Hızlı Tümlleştirilmiş Devre Donanım Tanımlama Dili)
VLSI	: Very Large Scale Integrated Circuit (Çok Geniş Ölçekli Tümlleştirme)
XOR	: Exclusive Or (Özel Veya)

BÖLÜM 1

GİRİŞ

Son yıllarda temel bilimlerde üzerinde araştırma ve inceleme yapılan bilim alanlarından birisi de doğrusal olmayan kaotik sistemlerin varlığı ve bu sistemlerin bulundukları devreler üzerinde oluşturdıkları olumlu/olumsuz etkileridir. Kaotik sistemler başlangıç koşullarına hassas bağlı, karmaşık ve düzensiz görünümündedir ve deterministik doğrusal olmayan zamanla değişen sistemlerde ortaya çıkarlar. Kaotik sistemlerin çok karmaşık olmayan devre yapılarına sahip olmalarına rağmen ilginç dinamik özelliklere sahip olmaları, son yıllarda kaotik sistemlere olan ilgiyi oldukça arttırmıştır. Kaos tabanlı mühendislik uygulamalarında kullanılması gereken temel yapı, gerekli kaotik işareti üreten bir kaos üreticidir. Mühendisliğin pek çok alanında kaotik sistemlerin varlığının ortaya çıkarılması, bu konuda yapılan yoğun çalışmalar ve yaşanan gelişmeler kaotik sistemlerin birçok uygulama alanında kullanılabileceğini göstermiştir. Bu uygulama alanlarına kriptoloji [1], biyomedikal [2,3], güvenli haberleşme [4,5], kuantum elektronigi [6], görüntü işleme [7], optik elektronik [8], bulanık mantık [9,10], güç elektronigi [11], kaos kontrol [12,13], optimizasyon [14], mekatronik [15], yapay sinir ağları [16–19] gibi alanlar örnek olarak verilebilir. Kaotik sinyaller elektrik-elektronik mühendisliğinde güvenli haberleşme, şifreleme, gürültü üreticileri, rasgele sayı üreticileri gibi uygulama alanları bulunmaktadır. Bu nedenle yeni ve farklı özelliklere sahip kaotik işaret üreticilerinin literatüre kazandırılması gerekliliği de her zaman önem arz etmektedir [20].

Günümüzde sayısal haberleşmede kullanılan teknolojilerde hızlı gelişmeler olmuştur. Bunların başında özellikle internet ve cep telefonlarındaki gelişim gelmektedir. Bu gelişim, insanların hayatlarında önemli değişiklikler yapmasının yanı sıra gizli ve özel olan bilgiye erişmeye çalışan saldırganları da mümkün hale getirmiştir. Bu bilgiyi korumanın çözümlerinden biride kriptografiyi kullanmaktır [21]. Kriptografi

özellikle son yüzyılda askeri alanlarda gizli bir bilginin bir yerden diğer bir yere güvenli olarak taşınması amacıyla kullanılmaktadır. Kriptografi, veriyi şifreleme ve şifre çözme işlemleri ile ilgilenir. Veri şifreleme, gizli bir anahtar kullanımı ile açık metnin anlaşılabilir bir biçime (şifreli-metne) dönüştürme süreci olarak tanımlanır; şifreyi çözme ise tam tersidir [21,22]. Son gelişmeler kriptografi biliminde, anahtar ile rasgele sayılar arasında kuvvetli bir bağ oluşmasına neden olmuştur. Çünkü iyi kriptografi iyi rasgele sayılara gereksinim duyar. Kaotik işaretlerin gürültü benzeri işaretler üretmesi, periyodik olmayan davranış sergilemeleri, başlangıç koşullarına hassas bağlı olmaları vb. özellikleri sebebiyle kaotik sistemler son yıllarda kriptolojide rasgele sayı üretiminde ve güvenli haberleşme alanında sıklıkla kullanılmaya başlanmıştır [24–26].

Rasgele sayıların üretilmesi Rasgele Sayı Üreteçleri (RSÜ) ile sağlanmaktadır [27]. RSÜ'leri tarafından üretilen sayılar bilgi güvenliği sistemlerinde anahtar olarak kullanılabilir. Ancak anahtarların sistem dışında kontrolsüz ortamlarda üretimi sistemin güvenilirliğini azaltmaktadır [28–30]. Bu dezavantajı ortadan kaldırmak için günümüzde donanımsal kriptografinin ve güvenli haberleşmenin gelişimi, programlanmış kript aritmetiğinin DSPs (Sayısal İşaret İşlemciler (Digital Signal Processors) [31–33], ASIC (Application Specific Integrated Circuit- Uygulamaya Özel Tüm Devre) [34–36], FPAAs [36,37], PSoC [39], ADC [40], Microcontroller [41] ve FPGA (Field Programmable Gate Array - Alan Programlanabilir Kapı Dizileri) [42–45] gibi sayısal devre tabanlı entegrelerde üretilmeye başlanmıştır. Kaos tabanlı mühendislik uygulamalarının artırılması ve yaygınlaştırılması açısından bir taraftan ortaya konan kaotik devre modellerinin çeşitlendirilmesi ve diğer taraftan da bu modellerin daha fonksiyonel ve esnek bir biçime kavuşturulmaları gerekmektedir. Esnek elektronik devre tasarımında kullanılan FPGA'ler ile yeni devre tekniklerinin kaotik işaret üreteç tasarımına uygulanması, kaotik devre modellerini ve bu modellerin kullanıldığı uygulamaları çok daha esnek ve kullanışlı kılacaktır. FPGA-tabanlı kaotik devre modelleri programlanabilirlik ve yeniden yapılandırılabilirlik açısından son derece uygun sistemlerdir. Çünkü kaotik sistemlerde parametre değişimlerine göre farklı modda işaret üretilmesi ve ilgili kaotik sistemin farklı doğrusal olmayan fonksiyonlarla alternatifli olarak gerçekleştirilmesi söz konusu olup bu özelliklere sahip kaotik

devre modelleri programlanabilir ve yeniden yapılandırılabilir bir şekilde esnek ve donanım karmaşası olmadan kolay bir şekilde tasarlanabilirler.

Rasgele sayı üreteçleri Sözde Rasgele Sayı Üreteçleri (SRSÜ) ve Gerçek Rasgele Sayı Üreteçleri (GRSÜ) iki sınıfa ayrılmaktadır [45,46]. GRSÜ'ler entropi kaynağı olarak deterministik olmayan fiziksel olaylar kullanılmaktadır [48]. Bu yapı daha güvenli bir kaynak oluşturduğu için kriptografide anahtar olarak öngörülemeyen bit dizisi oluşturmada sıklıkla kullanılmaktadır. GRSÜ'ler yavaş, maliyetli ve donanıma bağımlı olması gibi dezavantajlara sahip olmasına rağmen yüksek güvenlik uygulamaları gerektiren güvenli haberleşme ve kriptoloji gibi uygulamalarda daha çok kullanılmaktadır [48].

Kriptografik uygulamalar RSÜ'lerinin özelliklerine aşırı derecede duyarlıdır. Bir rasgele sayının bilinmemesi kavramı kriptografik çalışmalarda temeldir. Kriptografide kullanılacak bir RSÜ'nin ilave olarak tahmin edilemez olması da gerekir. Yani, diziyi oluşturan algoritma ya da donanıma ilişkin tüm bilgi ve akıştaki tüm önceki bitler verilse dahi bir sonraki rasgele bitin ne olduğunu tahmin etmek hesapsal açıdan olanaksız olmalıdır. Literatürdeki çalışmalarda şifreleme sistemlerinde gerçek sayı üreteçlerinin anahtar olarak kullanımının önemi gün geçtikçe artmaktadır. GRSÜ gerçek entropi kaynaklı tamamen rasgele sayılar üreten bir donanımdır. Literatürde GRSÜ uygulamaları için çeşitli yöntemler sunulmuştur [49–51]. Bu yöntemleri;

1. Gürültü İşaretinin Doğrudan Kuvvetlendirilmesine Dayalı RSÜ,
2. Osilatör Örnekleme Yöntemine Dayalı RSÜ,
3. Kaos Tabanlı RSÜ'leri olarak üç ana başlık altında toplamak mümkündür.

Gürültü İşaretinin Doğrudan Kuvvetlendirilmesine Dayalı Gerçek RSÜ, devre gerçeklemelerinde ısı veya saçılma gürültüsü gibi fiziksel gürültü üreten kaynaklar kullanılmaktadır. Bu devrelerin gerçeklemelerinde, bu tür kaynakların devre üzerinde gerçekleşmesinin bazı sakıncaları ortaya çıkmıştır. Bu sakıncalardan en temel olanı devre içinde var olan gürültü kaynaklarının ürettikleri işaretlerin çok düşük güçlü olmaları, dolayısıyla devredeki istenmeyen işaretlerden etkilenmeleridir. Analog ve

sayısal blokların birlikte gerekçelendiği karışık devre yapılarında, besleme ve toprak uçlarındaki muhtemelen periyodik olan istenmeyen işaret seviyelerinin, fiziksel gürültü işaretlerinin ürettiği gürültünün tipik seviyelerin çok üstünde olduğu bilinmektedir. Öte yandan, fiziksel gürültüyü kuvvetlendirmek için kullanılacak kuvvetlendiricinin kazancında oluşacak bant sınırlaması nedeniyle, bu kuvvetlendiricinin çıkışındaki gürültünün bandı da sınırlı olacaktır. Bunun yanında ayrıca ısı gürültü tabanlı GRSÜ yapıları bulunduğu sistemin içinde oluşan elektromanyetik alandan, sıcaklıktan, besleme gürültüsünden ve kuvvetlendiricinin gürültüsünden etkilenmektedir [47,51–53].

Osilatör Örneklemeye Yöntemine Dayalı Gerçek RSÜ, yapısında yavaş osilasyonun yükselen kenarlarında hızlı osilatörün çıkışları örneklenmektedir. Yapıda seğirmeli yavaş osilasyonu üretmek için kullanılan alt devreler değişiklik göstermektedir. Bu şekilde gerçekleştirilen RSÜ'nin bit üretme hızı, seğirme gürültülü yavaş osilatörün ortalama frekansına eşit olmaktadır. Osilatör örneklemeye yöntemine dayalı RSÜ'lerin besleme gerilimlerinden gelen istenmeyen işaretlere ve gürültünün sınırlı bant genişliğinden gelen olumsuzluklara, gürültünün doğrudan kuvvetlendirilmesine göre daha az duyarlıdır. Osilatör örneklemeye yöntemine göre gerçekleştirilen GRSÜ yapılarının ortalama bit üretim hızının belirli bir hızın üzerine çıkamaması [54,55] ve gerçekleştirilen RSÜ'nin etkili bir rasgelelik sağlayamamasıdır [56,57]. Analog elemanlar kullanılarak sürekli zamanda gerçekleştirilen osilatörlerin çalışma frekanslarının düşük olması ve ayrıca bu yapıların sistem parametrelerinin değişimine dirençli olması gibi problemleri ortaya çıkmaktadır [58,59].

Gerçek rastgele sayı üretiminde kullanılan ilk iki yöntemin olumsuz yönlerini iyileştirmek, rasgele sayı üretiminde kullanılan ana kaynağın entropisini daha da arttırmak ve yüksek çalışma frekansı ile yüksek bit üretim hızında daha gerçekçi rasgele sayı dizileri elde etmek amacıyla kaotik tabanlı sistemlerin kullanımı en uygun alternatiflerdir [60,61].

Bilgi güvenliği alanındaki kriptolojik sistemler, belirli devre ve aritmetik işlemlerin birleşiminden oluşmaktadır. Günümüzde donanımsal kriptolojinin gelişimi, programlanmış kriptolama aritmetiğinin bir çip içerisine gömülerek

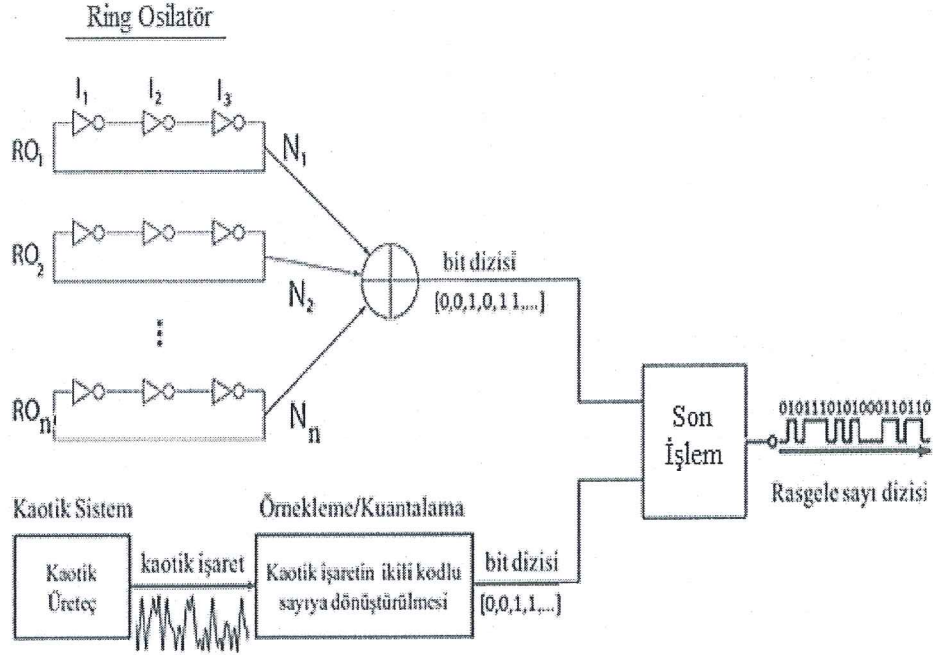
gerçekleştirilmesi yönündedir. Kriptoloji ve güvenli haberleşme sistemlerinde büyük önemi olan RSÜ'lerinin geleneksel üretimindeki yukarıda belirtilen olumsuz etkileri yok etmek ve daha rasgele diziler elde etmek için Kaotik devre tabanlı GRSÜ alternatif olarak görülmüş ve son bir kaç yılda bu alanda önemli çalışmalar yapılmıştır. Bu sistemlerde kaotik çekerlerin gürültü yerine kullanılabileceği görülmüştür. Kaotik sistemler dinamik yapılardır yani zamana bağlı olarak değişim göstermektedir. İkinci olarak bu sistemler değişken ve aperiodyik yapılardır yani kendilerini tekrarlamazlar. Kaotik sistemler kararsız ve giriş koşullarına üstel duyarlıdırlar. Böylece kaotik sistemler deterministik sistemler olmalarına rağmen bu özelliklerinden dolayı yapılacak perturbasyon sonuçları incelendiğinde sistemlerin rasgele davrandığı gözlenmektedir. Düzensiz davranışları ve başlangıç koşullarına aşırı hassas oluşları nedeniyle, kaotik işaretler de rasgelelik kaynağı olarak değerlendirilmekte ve GRSÜ yapımında kullanılmaktadırlar.

1.1. TEZİN AMACI

Bu tezin amacı; FPGA üzerinde gerçek zamanlı, farklı nümerik diferansiyel denklem çözüm yöntemleriyle tasarlanacak yüksek bit üretim hızına sahip yeni kaotik sistemler ile yüksek çalışma frekanslı ring osilatörler kullanılarak Kaos Tabanlı Çift Entropi Çekirdekli GRSÜ'lerin tasarlanması ve gerçekleştirilmesi hedeflenmektedir. Bunun yanında literatürdeki GRSÜ'de kullanılan yöntemlerin sorunlarına mutlak çözüm bulmaktır. Bu tasarım ile yeni bir çift entropi çekirdekli GRSÜ ilk defa FPGA üzerinde tasarlanarak literatüre sunulmuştur.

Sunulan tezin en önemli amaçları bir önceki bölümde zikredilen GRSÜ üretiminde kullanılan ilk iki yöntemin temel sorunlara çözüm bulmaktır. Birinci sorunun çözülmesi amacıyla tez sürecinde klasik osilatörler yerine kaotik tabanlı osilatörler geliştirilerek ortalama bit üretim hızı artırılacaktır. İkinci sorunun çözümüne yönelik olarak da yüksek esnekliğe sahip FPGA tabanlı sistem tasarımlarından yararlanılarak osilatörlerin çalışma frekansları artırılacak ve ısı duyarlılıkları azaltılacaktır. Ayrıca literatürdeki birçok GRSÜ yapısına bakıldığında daha güçlü bir entropi kaynağı sağlamak için fiziksel gürültüden yararlanıldığı görülmektedir. Kaotik işaretlerin gürültü davranışına benzer düzensiz bir yapıya

sahip olması ve zamanla çabuk kaybolan otokorelasyon fonksiyonlarından dolayı kaotik işaretlerin RSÜ yapılarında gürültü kaynaklarının yerine entropi kaynağı olarak kullanılması düşünülmüştür. Bu bilgiler doğrultusunda Şekil 1.1’de gösterilen tasarımda rastgele sayı üretimi için ayırık zamanlı çift entropi çekirdekli yeni bir yapı önerilmiştir.



Şekil 1.1. Ayrık zamanlı önerilen kaos tabanlı çift entropi çekirdekli yeni GRSÜ.

Önerilen bu yapıda; birinci entropi kaynağı olarak çalışma frekansı yüksek olan ring osilatörler kullanılmıştır. İkinci entropi kaynağı olarakta güçlü belirsizlik özelliğine sahip yeni kaotik sistemler kullanılmıştır. Daha sonra her iki entropi kaynağından alınan bitler kendi arasında karıştırılarak daha güçlü, daha hızlı ve istatistiksel değerleri çok daha kuvvetli rasgele sayı dizileri üreten yeni bir yapının tasarımı gerçekleştirilmiştir.

1.2. TEZİN ÖZGÜN DEĞERİ

FPGA tabanlı sayısal kaotik sistemler rasgele sayı üretimi, kriptoloji ve güvenli haberleşme alanlarında bilgi güvenliği kapasitesini iyileştirmede önemli bir potansiyele sahiptir. FPGA ve genel amaçlı mikroişlemciler ile donanımsal olarak

RSÜ yapıların geliştirilmesi alanında oldukça büyük çabalar sarf edilmektedir. Literatür kısmında sunulan çalışmalardan da görüleceği üzere yoğun bir biçimde GRSÜ konusunda çalışmalar yapılmaktadır. Bu çalışmaların bir kısmı, kaos tabanlı GRSÜ'lerinin CMOS (Analog Bütünleyici Yarı İletken Metal Oksit (Complementary Metal Oxide Semiconductor)) üretim süreci ile tasarlanmaktadır. Fakat belirtildiği gibi bu yapılar ısı duyarlılık gibi fiziksel özelliklerden etkilenmektedirler. Yapılan çalışmaların diğer bir kısmı da, klasik osilatörler ile GRSÜ'nin FPGA tabanlı olarak gerçekleştirilmesi üzerinedir. Klasik osilatörler kullanan FPGA tabanlı GRSÜ'leri yüksek çalışma frekansı sağlamakta ancak kullanılan osilatörlerden dolayı ortalama bit üretim hızı sınırlı olmaktadır.

Bu tez ile gerçekleştirilen çalışmanın literatürdeki FPGA tabanlı kaotik GRSÜ çalışmalarından farklı olarak çift entropi çekirdekli olması ve bu yeni yapının VHDL (Very High Speed Integrated Circuit Hardware Description Language (Çok Yüksek Hızlı Tümlşik Devre Donanım Tanımlama Dili)) dilinde IQ-Math Fixed-Point Number (Sabit-Noktalı Sayı) formatında tasarımının gerçekleştirilmesi olacaktır.

Tezin birinci özgün yönü, FPGA tabanlı yeni kaotik osilatörler tasarlayarak GRSÜ için çift entropi çekirdekli yeni yapının gerçekleştirilmesidir. Bu yapı sayesinde kaotik sistem ile ring osilatörün avantajları birleştirilerek, yüksek çalışma frekansına ve yüksek bit üretim hızına sahip yeni bir GRSÜ sistemi tasarlanmıştır.

Tezin ikinci özgün yönü, önerilen sistemin FPGA üzerinde VHDL dilinde tasarlanmasında IQ-Math sabit noktalı sayı formatının kullanılmasıdır. Literatürdeki IEEE 754-1985 kayan nokta sayı formatındaki tasarımlarına nazaran sabit noktalı sayı formatının daha az FPGA çip istatistikleri kullandığı ve bu sayede önerilen yapının çalışma frekansının ve bit üretim hızının arttığı gözlemlenmiştir. VHDL tamamen sayısal çalışmaktadır. Bu yüzden reel sayılar üzerinde işlem yapmak için sayıların sabit noktalı (IQ-Math Fixed-Point) veya kayan noktalı (IEEE-754 Floating-Point) biçimde ifade edilmesi gerekir. Sabit noktalı sayı biçimi hızlı ve kolay uygulanabilmesine rağmen, kayan noktalı sayı biçimi daha duyarlı işlemler için kullanılır.

1.3. TEZDE İZLENECEK YOL

Bu tez çalışması, yüksek çalışma frekansı ve bit üretim hızına sahip kaos tabanlı çift entropi çekirdekli yeni Gerçek Rasgele Sayı Üretici'nin tasarlanması ve bu tasarımın FPGA üzerinde gerçekleştirilmesi için yedi bölümden oluşmaktadır. Bu doğrultuda, ikinci bölümde çalışmanın içinde kullanılan alakalı temel kavramlar açıklanacaktır.

Üçüncü bölümde; kaos analiz yöntemlerinden bahsedilerek literatüre son yıllarda sunulan ve üzerinde FPGA tabanlı çalışma yapılmamış kaotik sistemler tanıtılacaktır. Tanıtılan her kaotik sistem MATLAB programında sayısal analiz yöntemlerinden Heun algoritması kullanılarak nümerik analiz çözümleri elde edilecektir. Ayrıca her bir kaotik sistemin kaotik dinamik analizlerinden zaman serileri, faz portreleri ve Lyapunov üstelleri hesapları yapılarak sistemlerin kaotik oldukları gösterilecektir.

Dördüncü bölümde; referans alınan PCKS ve P3BKS kaotik sistemleri MATLAB programında farklı numerik diferansiyel denklem çözüm algoritmalarıyla modellenecektir. Daha sonra seçilen kaotik sistemler FPGA üzerinde Euler, Heun, RK4 ve RK5-Butcher algoritmaları ile tasarım ve gerçekleştirme çalışmaları yapılacaktır. Ayrıca tezde önerilen çift entropi çekirdekli yapı oluşturulabilmek için ring osilatörün FPGA üzerinde tasarımı da bu bölümde gerçekleştirilecektir. Yapılan tüm çalışmalar sonucunda kullanılan algoritmaların hassasiyet analizi yapılarak FPGA tabanlı yeni kaotik sistem yapılarının performans ve çip istatistikleri incelenecektir. Son kısımda ise literatürde FPGA tabanlı kaotik osilatör çalışmaları hakkında bilgi verilecektir.

Beşinci bölümde; daha önceki bölümde tasarımları sunulan FPGA-tabanlı PCKS ve P3DKS kaotik sistemleri ile ring osilatör yapısı kullanılarak VHDL dilinde 32 bit IQ-Math sabit noktalı sayı standardında çift entropi çekirdekli gerçek rasgele sayı üretici tasarımı gerçekleştirilecektir. Xilinx firmasının ürettiği Virtex-6 ailesinin XC6VLX75T-3FF784 çipi için sentezlenerek, FPGA çip kaynak kullanımına ve ünitelerin saat hızlarına ait parametrelerin istatistikleri incelenecektir. Tasarımı yapılan 8 farklı yapıdaki GRSÜ ünitelerinin verileri işleme süresi, Xilinx ISE Design

Tools 14.2 simülasyon programı kullanılacaktır. Literatürdeki GRSÜ tasarımları hakkında detaylı bilgi verilecektir.

Altıncı bölümde; FPGA üzerinde kaos tabanlı çift entropi gerçekleştirilen 8 adet yeni GRSÜ'nden elde edilen sayı dizileri test edilecektir. Bu amaçla literatürde geliştirilmiş çeşitli istatistiksel testler kullanılmaktadır. GRSÜ'nin kriptolojik uygulamalarda güvenli bir şekilde kullanılabilmesi amacıyla uluslararası FIPS 140-1 ve NIST 800-22 istatistiksel test paketleri kullanılacaktır.

Son bölümde; FPGA üzerinde yüksek çalışma ve bit üretim hızına sahip yeni kaos tabanlı çift entropi çekirdekli GRSÜ'nden elde edilen sonuçlar yorumlanacaktır. Ayrıca tez sonrası yapılabilecek çalışmalar hakkında öneriler paylaşılacaktır.

BÖLÜM 2

KONU VE KAPSAM

Günümüzde kaos ve kaotik sistemler, popüleritesini iyice arttırarak güncel konular arasında yerini almış ve rasgele sayı üretimi, kriptografi, güvenli haberleşme, sayısal sinyal işleme, ekonomi, güç trafoları, hastalık teşhisi, yapay sinir ağları, hava tahminleri, optimizasyon, sürücü sistemleri, kontrol sistemleri gibi pek çok bilimsel alandaki araştırmalara ve projelere, yüksek lisans ve doktora seviyesindeki tezlere konu olmuştur. Kaotik sistemlere ait işaretlerin istemli bir biçimde meydana getirilmesi ister mühendislik bilimlerinde olsun ister diğer bilim dallarında olsun oldukça önemli bir aşamadır. Kaos sistem dinamiklerinin istemli bir biçimde elektronik devreler ile oluşturulması için de günümüzde oldukça fazla çalışmalar yapılmakta ve bu çalışmalar gün geçtikçe artış göstermektedir [63–67].

2.1. KAOS VE KAOTİK SİSTEMLER

Kaos tanımı incelendiğinde başlangıç koşullarına üstel duyarlı, nonlinear, deterministik karakterli, uzun vadede periyodik olmayan dinamik sistemler olduğu görülmektedir. En kısa tarifiyle ise, düzensizliğin düzeni şeklinde tanımlanan ve doğrusal olmayan olayları açıklamaya yarayan bir bilim dalıdır. Karmaşık, ama kendi iç düzenine sahip bir süreçtir. Özellikle dikkat edilmesi gereken bir nokta, kaosun rasgelelik olmadığıdır. Dinamik sistemlerde bilinen en karmaşık kararlı hal davranışı kaos'tur. Kaos ile ilgili çalışmalar, doğrusal olmayan dinamik sistemler teorisinin bir kısmıdır. Bu durum daha çok “deterministik kaos” olarak adlandırılır [67,68].

Kaos kavramı matematiksel olarak 1900'lerde incelenmeye başlanmış ve ilk ciddi bilimsel çalışma Edward Lorentz'in 1961'de hava tahmini yapmak için oluşturduğu matematiksel meteorolojik modelin sonuçları sayesinde elde edilmiştir. Edward

Lorentz bilgisayarla yaptığı modelde sayısal analizlerden elde ettiği sonuçları hızlandırmak için aldığı verileri yuvarlayarak kullanmıştır. Ancak sonuçlar çok hızlı bir şekilde değişerek tahmin edilemez bir hal almıştır. Böylece Lorentz, farkında olmadan kaos teorisinin temellerini atmıştır. Bu gelişmenin doğrultusunda kaos teorisi o zamana kadar çözülemeyen bir çok fiziksel ve matematiksel problemlerin çözülmesini sağlamıştır. Günümüzde kaos kavramı astronomi, biyofizik, biyoloji, fizik, jeoloji, kimya, matematik, meteoroloji, mühendislik, plazma, tıp hatta sosyal bilimlerde açıklanması zor olan olguların modellenmesi ve çözülmesinde kullanılmaktadır. Literatürde çok sayıda otonom kaotik devre geliştirilmiş olsa da üzerinde en çok çalışma yapılan ve kaotik dinamikleri en iyi bilinen otonom sistemler Chua, Rössler, Lü, Liu, Duffing, Chen ve Lorenz sistemleridir [70]. 1984'te geliştirilen otonom Chua devresi [71], elektronik devre olarak gerçekleştirilen ilk kaos devresidir. Chua kaotik sistemi basit bir devre yapısına sahip olmasına rağmen kompleks dallanma ve kaos sergilemesi dolayısıyla elektronikteki kaos olayının açıklanmasında model devre olmuştur. Kaos ve kaotik işaretlerle ilgili yapılan uygulamalarda kaos üretici olarak genelde Chua devresi kullanılmıştır [72–76]. Literatürde özellikle kaotik işaret üreten doğrusal olmayan osilatör devreleri otonom ve otonom olmayan kaotik osilatör devreleri olarak sınıflandırılmaktadır. Kaotik sistemlerin tanımına bakıldığında [76,77];

1. İlk olarak dinamik yapılardır, yani zamana bağlı olarak değişim göstermektedir.
2. Başlangıç şartlarına hassas bağımlılığı,
3. Bu sistemler değişken ve aperiodyk yapılardır, yani kendilerini tekrarlamazlar. Sınırsız sayıda değişik periyodik salınımlar içerir.
4. Kaotik yapılar karmaşık yapılar gibi gözüксе de aslında basit yapılardan meydana gelmektedir.
5. Bu sistemler nonlineer yapılardan oluşmaktadır.
6. Gürültü benzeri geniş güç spektrumuna sahiptirler.
7. Limit kümesinin parçalı (fraktal) boyutlu olması,
8. Genliği ve frekansı tespit edilemeyen, ancak sınırlı alanda değişen işaretler içermesi,
9. Son olarakta kaotik sistemler deterministik yapılardır.

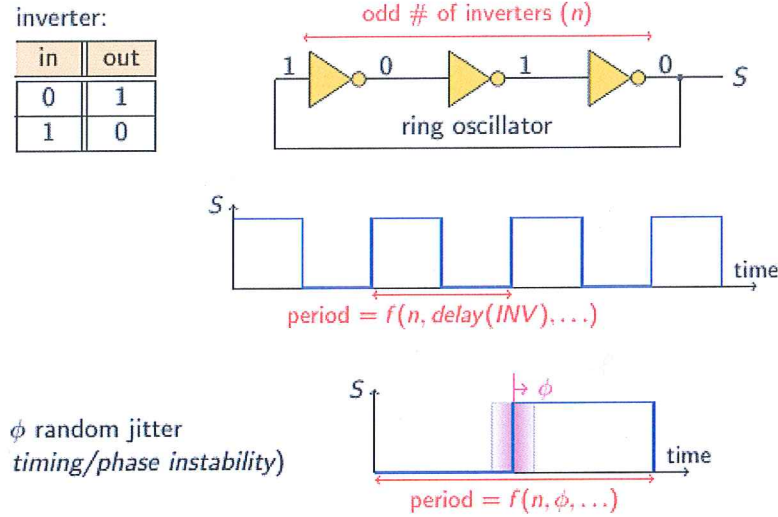
Ama bu determinizm kavramı beraberinde bazı anlam kargaşalarını da beraberinde getirmektedir. Kaotik sistemler aperiyojik ve tahmin edilemez gibi gözükseler de deterministik kavramından dolayı bu sistemlerin rasgelelik gösterdiği söylenememektedir. Yani kaotik sistemlerin bir andaki durumu tam olarak bilindiğinde, sonraki herhangi bir andaki durumu da tam olarak belirlenebilmelidir. Fakat kaotik sistemler kararsız ve giriş koşullarına üstel duyarlıdır. Böylece kaotik sistemler deterministik sistemler olmalarına rağmen yukarıda sayılan özelliklerinden dolayı bu sistemlerin rastgele davrandığı gözlenmektedir [79]. Kaotik sistemler ayrık zamanda kaotik sistemler ve sürekli zamanda kaotik sistemler olmak üzere ikiye ayrılır [79,80].

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. Ayrık ve sürekli zamanda kaotik sistemler kullanılarak bu belirtecek ihtiyaçları karşılayacak devreler tasarlanabilmektedir. Sürekli zamanlı kaos tabanlı GRSÜ uygulamalarının genellikle büyük bir alanı işgal eden diferansiyel dinamiklerini tanımlamak için OPAMP, osilatör ve endüktörler gibi büyük analog yapı bloklarına ihtiyaç duyması sonucunda yüksek güç tüketirler [82]. Analog kaotik üreteç yapıları kullanan sistemlerde bilgi sinyalinin tekrar elde edilmesi, verici ile alıcının çok iyi bir şekilde senkronize edilebilmesine bağlıdır. Bu ise verici ve alıcı parametrelerinin yüksek doğruluk derecesinde eşleşmesiyle sağlanabilir. Sıcaklık ve kullanım ömrü ile değerleri değişen analog devre elemanları ile bu parametreleri eşleştirebilmek analog sistemlerde oldukça güçtür [82,83]. En iyi çözüm sürekli zamanlı kaotik üreteçlerin sayısal donanım (FPGA, DSPs, ASIC) kullanarak ayrık zamanlı olarak gerçekleştirilmesidir.

2.2. RING (HALKA) OSİLATÖRLER

Arka arkaya bağlanmış tek sayıdaki değil kapısından oluşan osilatörlerdir. Her kapının çıkışı bir sonraki kapının girişine, son kapının çıkışı ise ilk kapının girişine bağlanır. Halka osilatörler, halka içerisindeki gecikmelere bağlı olarak bir frekansta kare dalga üretirler. Dolayısıyla elde edilen kare dalganın frekansı, halkayı oluşturan

elemanlardaki statik ve dinamik etkenlere göre farklılık gösterir. Yani, eşit şekilde dizilmiş iki osilatörün ürettiği işaretlerin frekansları aynı olmayacaktır. Bu, halka osilatörlerin ürettikleri işaretlerin frekanslarındaki farklılık rasgele bitler üretmek için kullanılabileceklerini göstermektedir [85–87]. Şekil 2.1’de ring osilatörün genel yapısı gösterilmektedir.



Şekil 2.1. Tek sayılı evirici içeren ring osilatör yapısı.

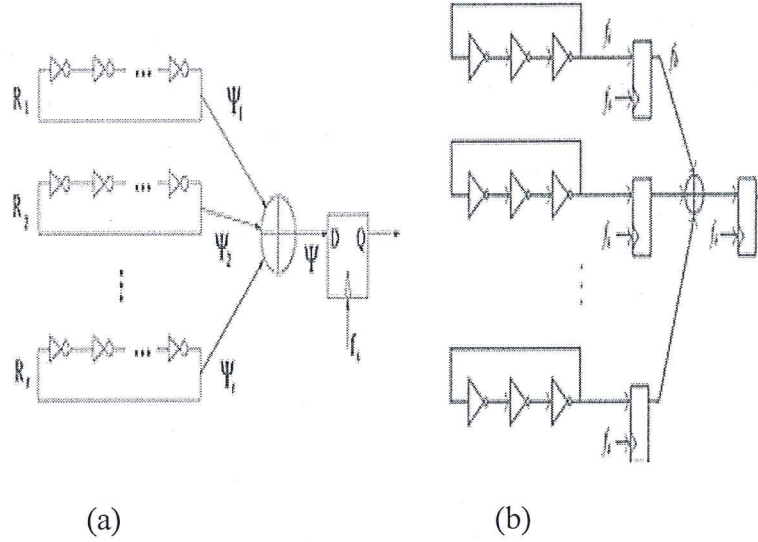
Tümdevre uygulamaları ve FPGA tabanlı gerçek rasgele sayı üreticilerinin çoğu rasgelelik kaynağı olarak ring osilatör yapısını kullanırlar. Halkanın içine bağlanan eviricilerin gecikme kararsızlığı üretilen saat frekansı/faz kararsızlığı (jitter) olarak gözükmektedir. Jitter örnekleme biriminde flip flop veya latch kullanılarak ayıklanır. Çünkü üretilen sinyalin içindeki jitter doğrudan rasgelelik kaynağı olarak kullanılmaktadır. Kaynağın çalışmasından önce geçici ve istatistiksel parametrelerinin değerlendirilmesi gerekmektedir. Genellikle donanımsal olarak mantık kapıları ile üretilen jitter, hızlı osilaskop kullanılarak ölçülmektedir. Ring osilatör içinde üretilen jitter, rasgele Gauss salınımı ve deterministik bir jitter olarak bilinmektedir. Üretilen bu jitterin rasgele sayı üretimine etkisini değerlendirmek önemlidir. Ayrıca deterministik jitterin durumu da oldukça önem arz etmektedir. Çünkü bu cihaz dışından manipüle edilebilir ve üreticinin üzerine bazı ataklar gerçekleştirilebilir. Deterministik ve rasgele jitterin, üretilen rasgele sayılar üzerindeki etkisine yeterince bakılmamıştır. Dışarıdan gerçekleştirilen etkiler rasgele sayıların kalitesini olumsuz etkileyebilir. Valtchanov vd. yaptıkları çalışmada hem bu ataklardan korumak ve

hem de jitterin rasgele sayı üretimine etkisini incelemek için Altera ve Actel firmalarının ürettikleri FPGA çipleri üzerinde VHDL dilinde donanım tabanlı ring osilatör tasarımlarını gerçekleştirmişlerdir [87].

Ring osilatörler tek sayılı invertör içeren bağımsız çalışan osilatörlerdir. Halka içindeki her bir evirici sırasıyla iki yarım periyotta üretilen saat sinyalinin yükselen ve düşen kenarını üretmektedir. Böylece üretilen toplam saat periyodu Eşitlik 2.1’de ki gibi;

$$T = 2 \cdot k \cdot d \quad (2.1)$$

hesaplanmaktadır. Burada $k \in (3, 5, 7, \dots, 2n-1)$ bir halkadaki evirici sayısı, d ise bir eviricinin gecikmesidir. Yapılan deneysel çalışmalarda k ’nın en küçük değerinde her bir gecikme elemanının sinyal fazını maksimum 60° ertelediği gözlemlenmiştir [88]. Ring osilatör saat sinyalinin frekansı doğrudan döngüde kullanılan kapıların sayısına bağlıdır. Diğer taraftan ring osilatörün saat jitterin entropisi de döngüde kullanılan gecikme elemanlarının sayısı ile ilişkilidir.



Şekil 2.2. Farklı yapıda ring osilatör yapıları. a) Berk vd. oluşturduğu ring osilatör yapısı, b) Wold vd. oluşturduğu ring osilatör yapısı.

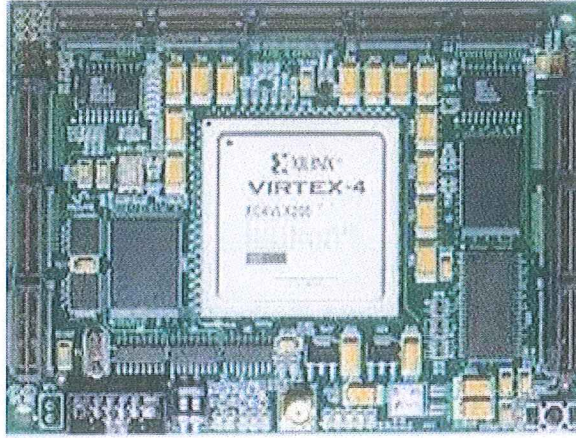
Yapılan birkaç deneysel çalışmalar ve literatür taraması incelendiğinde ring osilatör tasarımlarında oldukça farklı çalışmalar mevcut olduğu görülmüştür. Şekil 2.2a’da

görülen Berk vd.'nin yaptığı çalışmada 114 tane ring osilatör ve her bir ring içinde 13 tane invertör yapısını XOR yaptıktan sonra D flip/flop ile örneklemiştir [89]. Şekil 2.2b'de gösterilen diğer bir çalışmada Wold vd. 25 tane ring osilatör ve her bir ring içinde 3 tane invertörden oluşan yapının her bir halkasını D flip/flop ile örnekleme yaptıktan sonra sonuçları karıştırıp en sonunda tekrar örnekleme yaparak yeni bir yapı oluşturmuşlardır [90].

2.3. FPGA TEKNOLOJİSİ

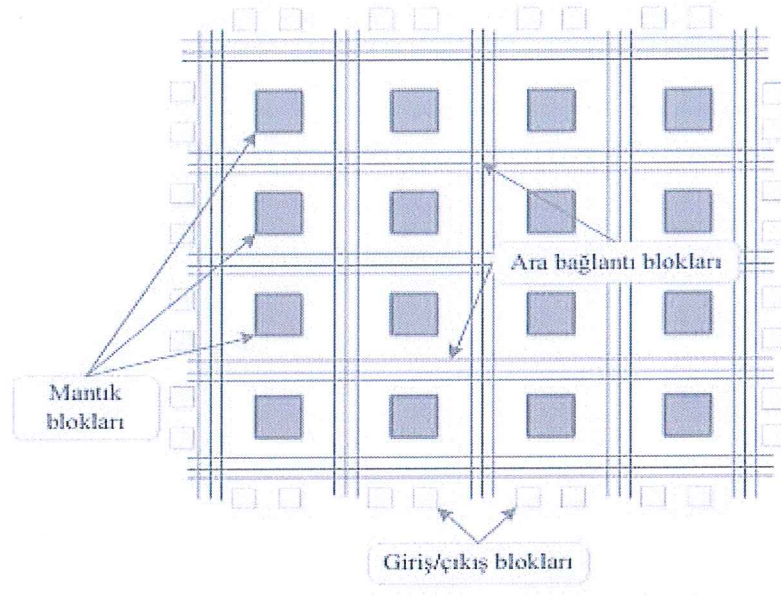
Sayısal elektronikte zaman içerisinde yaşanan gelişim çok büyük ölçekli sistemlerin tasarımını da beraberinde getirmiştir. Yarıiletken ve üretim teknolojilerinde geline nokta sayesinde bugün VLSI (Very Large Scale Integrated Circuit) yongaları içerisinde milyonlarca transistor bulunmaktadır. Malzeme ve üretim teknolojilerinin daha iyi noktalara taşınmış olması bunun son ürünlere de yansıtılması gerekliliğini beraberinde getirmektedir. Nitekim gelişen teknoloji yanında tüketici istekleri de artan oranda büyümektedir. Bilimdeki gelişmeleri mühendislik anlayışı içerisinde en kısa sürede ve verimlilikle tüketiciye ulaştırabilmek amacıyla yeni araç ve gereçler mühendislerin hizmetine sunulmaktadır. FPGA da sayısal tasarım mühendislerinin, bilimin oluşturduğu imkânları teknolojiye yansıtabilmesi için geliştirilmiş bir yongadır. Bugün geline nokta itibariyle markette, FPGA üretimi alanında, çok fazla sayıda olmamakla birlikte çeşitli üretici firmalar mevcuttur. Bu firmaların geliştirdiği FPGA'lar da farklı mimarilere ve üretim teknolojilerine dayanmaktadır. Farklı mimarilerin ve teknolojilerin birbirlerine göre üstünlükleri, güvenlik, güç tüketimi, maliyet gibi alanlarda değişmektedir.

FPGA çipleri tasarımcının yeniden ve sahada programlayabileceği tüm devrelerdir. 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. Şekil 2.3'te FPGA çiplerinin genel görünümü verilmiştir. Bu çipler yapılandırılabilir mantıksal bloklar, giriş/çıkış blokları ve ara bağlantılar olmak üzere üç temel bileşenden oluşurlar.



Şekil 2.3. Xilinx firmasının Virtex-4 FPGA çipinin görünümü.

1. Yapılandırılabilir Mantıksal Bloklar (Configurable Logic Blocks (CLB)), mantıksal fonksiyonların oluşturulabildiği Look-up table (LUT), tek bitlik bilgilerin saklanabildiği Flip-Flop'lar bilgi akışını yönlendiren çoklayıcılar ve toplayıcıların tanımlanmasında kullanılan “elde-zinciri” gibi çeşitli elemanlardan oluşurlar. Oluşturulmak istenen mantıksal devreler çeşitli yazılım araçları sayesinde bir CLB'ye veya parçalara bölünerek birden fazla CLB'ye otomatik olarak uygulanır.
2. Giriş Çıkış Blokları (Input/Output Blocks (IOB)), FPGA çiplerinin programlanabilir giriş/çıkış terminalleridir. Bu bloklar içinde yer alan pinler isteğe göre giriş, çıkış ya da çift yönlü olarak programlanabilir. Bu blokların temel görevi dış dünya ile çip içindeki donanım arasında köprü vazifesi görmektir. FPGA çipinin paket türüne göre bir çipteki IOB sayısı (yani pin sayısı) 1000'li sayılara ulaşabilmektedir.
3. Ara Bağlantılar (Interconnections), hem CLB'ler arasında hem de CLB'ler ile IOB'ler arasında bağlantıları yapılandırmada kullanılırlar. Programlanabilir olduklarından çok esnek bir yapıya sahiptirler. Bu temel bileşenlerin yanında bazı FPGA çipleri, sayısal sinyal işaret işleme için tasarlanmış özel bloklar, RAM hafıza blokları, hatta işlemci çekirdek üniteleri içerecek şekilde üretilmektedirler. Şekil 2.4'te FPGA çiplerinin genel yapısı görülmektedir.



Şekil 2.4. Bir FPGA çipinin genel yapısı.

FPGA'lar PAL (Programmable Array Logic-Programlanabilir Dizi Mantıkları), PLA (Programmable Logic Array-Programlanabilir Mantık Dizisi) gibi mantıksal kapı dizilerine benzer mantıkta üretilen ancak onlardan çok daha üstün teknolojiye sahip yongalardır. İçlerinde bugün gelinen son teknolojiye göre milyonlarca mantıksal kapı barındıran FPGA'lar üzerinde gerçekleştirilecek olan sayısal devreler FPGA üreticisi firmanın bilgisayar destekli tasarım (CAD) araçları sayesinde tasarlanmakta ve yongalara yüklenmektedir. FPGA'lar, üzerlerinde mantıksal kapı dizilerinden başka, hazır çarpıcı blokları, hazır RAM blokları, saat frekansı üretimi için DLL/PLL blokları ve hatta daha gelişmiş mimarilerde gömülü işlemci çekirdekleri bulundurmaktadır. Yarıiletken teknolojisinin gelişimi ve mikronaltı teknolojilerin pratikte uygulanmaya başlamasından itibaren bir zamanlar kapı gecikmelerinin yanında ihmal edilen yol gecikmeleri de hesaba katılmak durumunda kalmıştır. İşte bu nedenle FPGA'lar da kullanılan yarıiletken teknolojisi ile bağlantılı olarak, tasarımlar da farklı FPGA'lar da farklı saat frekanslarında çalışabilmektedir. Yine aynı şekilde üzerinde daha fazla sayıda mantıksal birim ve daha büyük boyutlarda RAM blokları bulunduran FPGA'lar da daha fazla işleme imkân tanınmasından dolayı daha kabiliyetlidir. Özellikle işaret işleme uygulamalarında kullanılan FPGA'lar da, yonga üzerinde bulunan RAM bloklarının boyutları büyük öneme sahip olmaktadır. Şimdiye kadar mikroişlemcilerle gerçekleştirilen pek çok uygulamanın bugün FPGA'lar ile gerçekleştiriliyor olması elbette ki yalnızca yonga üzerinde bulunan birkaç megabayt

blok RAM'dan dolayı değildir. FPGA'lar üzerinde gerçekleştirilen tasarımlar, mikroişlemcilerin aksine tamamen bir devreye karşılık gelmektedir. Bu durumda bir devrenin bir işlemi gerçekleştirme süresi, tasarıma bağlı olmakla birlikte sadece kapı gecikmeleri kadar uzun sürecektir. Hâlbuki mikroişlemcili sistemlerde bir tek komutun gerçekleştirilmesi, işlemcinin mimarisine bağlı olarak çok sayıda saat periyodu sürmektedir. Kısacası mikroişlemcilerde işler adım adım, sayısal işaret işlemcilerde (DSP) bir adımda üç, dört işlem şeklinde gerçekleştirilirken, FPGA'lar da işlemler blok blok gerçekleştirilmektedir. Bu avantajından dolayı FPGA tabanlı gerçekleştirmeler özellikle gerçek zamanlı uygulamalarda yani gecikmeye tahammülün olmadığı, zamanlamanın çok önemli olduğu noktalarda mikroişlemcili bir sistem gerçekleştirilmesine tercih edilmektedir.

FPGA'ların tercih edilmesinde tek unsur hız değildir. Bilindiği gibi mikroişlemci tabanlı sistemlerde, sistem üzerinde koşan işlem yazılımı ile gerçekleştirilir ve mikroişlemci tarafından adım adım gerçekleştirilir. Dolayısıyla sistemin program belleğine erişildiği anda sistem üzerinde koşan fonksiyonun ne olduğu da kolaylıkla anlaşılabilir. Bu durum mikroişlemcileri, kriptosistemler gibi güvenliğin büyük öneme sahip olduğu sistemlerin gerçekleştirilmesinde devre dışı bırakmaktadır. FPGA'lar ise gerçekleştirdikleri sistemlere, üretimlerinde kullanılan teknolojiye bağlı olarak ama kesinlikle mikroişlemciler kadar düşük olmayacak seviyede yüksek güvenlik sağlarlar. Pek çok FPGA da aynı mikroişlemci gibi güç kesildiğinde üzerindeki bilgiyi kaybeder, dolayısıyla bir program belleğine ihtiyaç duyar ancak program belleğinden FPGA'ya yüklenen, mikroişlemciye olduğu gibi komutlara karşılık düşen makine kodları değil, devre bağlantılarını belirten programlama dosyasıdır. Dolayısıyla istisnai durumlar bir kenara bırakılacak olursa bu dosyadan devrenin nasıl çalıştığı bilgisine erişmek pek mümkün değildir. Ayrıca çok daha yüksek güvenlik vadeden bazı FPGA'lar da bu dosyanın aktarımı da ortadan kaldırılmış, yonga, harici bir program belleğinden beslenmek yerine tamamen farklı bir mimari ile tasarlanarak SRAM tabanlı değil Flash bellek tabanlı üretilmiştir. Bu sayede yonga elektrik kesilse dahi üzerindeki bilgiyi korumaktadır ve dışarıdan tekrar tekrar programlanma gereği duymamaktadır.

FPGA teknolojisi günümüzde hala çok büyük miktarlarda üretimler söz konusu olduğunda ASIC teknolojisinden daha maliyetli bir çözümdür. Ayrıca ASIC çözümler FPGA'dan daha hızlıdır. Ancak FPGA tasarımları, pazara ulaşma süresinin çok kısa olması nedeniyle çok büyük miktarlarda üretimin söz konusu olmadığı durumlarda ASIC çözümlere tercih edilmektedir. FPGA tasarımları bugün ASIC tasarımın ilk adımı olarak görülmektedir ve model üretimler ilk olarak FPGA'lar da gerçekleştirilmekte, sistemin olası hata ve eksiklikleri tespit edildikten sonra ASIC tasarım ve üretime geçilmektedir. Bunun nedeni ASIC üretimin ilk maliyetinin çok yüksek olmasıdır. Hâlbuki bu maliyet FPGA'lar da yaklaşık olarak hiç yoktur ve sonuca ulaşma süresi de yok denecek kadar kısadır. Bu nedenlerden ötürü FPGA üretimi sürekli artmakta, teknolojisi gelişmekte ve fiyatları da her geçen gün düşmektedir.

FPGA çipleri, ilk üretim aşamasında, üretimin piyasaya hızlı bir şekilde sunulması söz konusu olduğunda çok büyük avantaj sağlarlar. ASIC çipleri, uzun süre gerektiren çeşitli tasarım ve üretim aşamalarından geçtikten sonra ancak kullanıma hazır hale gelebilmektedir. Fakat FPGA çipleri kullanılarak bu süre sadece tasarlanan ürünün FPGA çiplerine yüklenmesi kadar kısa bir sürede gerçekleştirilebilmektedir. Ayrıca tasarımda yapılan bir hatadan dolayı ASIC çipleri, kullanılamaz hale gelmesine rağmen bu dezavantaj FPGA çiplerinde yeniden programlanabilme özelliğinden dolayı, çok kısa bir sürede ve ekonomik bir zarara uğranılmadan telafi edilebilmektedir. Çiplerin programlanma süresi, boyutlarına ve tasarıma bağlı olarak yaklaşık 200 ms'den daha az sürmektedir. Çipleri yapılandırmak için tasarlanan donanım modülleri, sadece bir özel problemin çözümü için tasarlandıklarından yazılıma göre çok hızlı çalışmaktadırlar. FPGA çipleri sınırsız olarak tekrar programlanabilir ve 1 GHz saat hızlarına çıkabilirler. Ayrıca birden fazla FPGA çipi birbiri ile paralel olarak çalışabildiği hatta tek bir çip içerisinde birden fazla modül yerleştirilebildiği için yazılıma göre çok yüksek hız kazançları elde edilebilmektedir. FPGA çiplerinin gelişerek yüksek hız ve kapasiteye ulaşması bu çipleri pek çok alanda kullanılabilir hale getirmiştir. Özellikle yüksek performans ve işlemci gücü gerektiren uygulamalar için ucuz ve esnek bir alternatif olarak ön plana çıkmışlardır. FPGA tabanlı sayısal sistemler, kriptoloji ve güvenli haberleşme alanlarında bilgi güvenliği kapasitesini iyileştirmekte de önemli bir kullanım potansiyeline sahiptir.

2.4. GERÇEK RASGELE SAYI ÜRETEÇLERİ (GRSÜ)

GRSÜ'ler entropi kaynağı olarak deterministik karaktere sahip olmayan doğal fiziksel olayları kullanmaktadır. Bu sayede GRSÜ tekniği ile üretilen sayılar periyodik ve tahmin edilebilir olmayan bir yapıya kavuşmakta ve oluşan sayının rasgelelik kalitesi olabildiğince artmaktadır. Bu öngörülemeyen rastgele sayı üretiminin temelinde kararsız dinamik sistemler bulunmaktadır. Bu sistemlerin davranışı kaos teorisi ile açıklanmaktadır. Bu teorinin temeline bakıldığında işlemler deterministik gözüktse dahi gerçek hayatta saptanamayacak kadar küçük seviyelerdeki giriş koşulları farklılıkları ve durum değişkenleri sapmaları sistemin öngörülemez bir hale gelmesini sağlamaktadır. GRSÜ'ler donanım ve yazılım tabanlı olmak üzere iki farklı teknikle gerçekleştirilebilirler [91].

Gerçek rasgele sayılar kriptografik uygulamalar için çok önemlidir. Rasgele sayılar kriptografide başlangıç vektörü, özel ve gizli anahtarların oluşumunda ve Sözde RSÜ'de tohum üretici olarak kullanılır. Bu kullanılan rasgele sayıların tahmin edilememesi, iyi istatistiksel özellikler göstermesi ve düzgün dağılımlı olması kriptografik açıdan çok önemlidir. Şekil 2.5'te gösterildiği gibi GRSÜ'ler üç bileşenden oluşmaktadır. İlk olarak gerekli rasgeleliği üreten entropi kaynağı, ikinci olarak o kaynağın rastgeleliğini örnekleyici ve son olarakta istatistiksel kusurları düzeltmek için bir son işlemci bloklarından oluşmaktadır [92]. Şekil 2.6'da gösterilen ve Çiçek vd. tarafından yapılan çalışmada ise GRSÜ'lerinin daha detaylı bir şekilde sınıflandırılması yapılmıştır [93]. GRSÜ'leri Gürültü İşaretinin Doğrudan Kuvvetlendirilmesi Dayalı RSÜ, Osilatör Örneklemeye Yöntemine Dayalı RSÜ ve Örnek Kaos Tabanlı RSÜ'leri olarak üç ana başlık altında toplamak mümkündür.

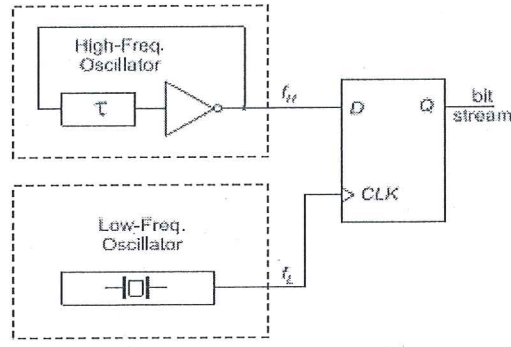


Şekil 2.5. Geleneksel GRSÜ blok şeması.

düşük 1/f gürültüsü üretmesi gereklidir. Bu sayede kuvvetlendirilecek gürültü beyaz gürültü özelliğini koruyabilecektir. Ancak kuvvetlendirici tasarımında bu dengeyi kurabilmek çok zordur. Bu sorunların üstesinden gelmek için giriş kapasitesi ve kazanç-bant genişliği çarpımı istenilen büyüklükte tasarlanmakta ve oluşacak olan yüksek seviyedeki 1/f gürültüsü bir filtre yardımı ile süzülmemekte ya da birim güç spektrumu seviyesine kadar bastırılmaktadır. Bu gereksinim de devrenin tasarım yükünü arttırmaktadır. Ayrıca bu tür kaynakların devre üzerinde gerçekleşmesinin bazı sakıncaları ortaya çıkmıştır. Bu sakıncalardan en temel olanı devre içinde var olan gürültü kaynaklarının ürettikleri işaretlerin çok düşük güçlü olmaları, dolayısıyla devredeki istenmeyen işaretlerden etkilenmeleridir. Analog ve sayısal blokların birlikte gerekçelendiği karışık devre yapılarında, besleme ve toprak uçlarındaki muhtemelen periyodik olan istenmeyen işaret seviyelerinin, fiziksel gürültü işaretlerinin ürettiği gürültünün tipik seviyelerin çok üstünde olduğu bilinmektedir. Bu nedenle fiziksel gürültünün kuvvetlendirilerek kullanıldığı tümleşik RSÜ'ler de bu tür rastlantısal olmayan istenmeyen işaretlerin etkilerini en aza indirmek için, gürültünün ürettiği alt devreye çok dikkatli bir şekilde elektromanyetik ekranlamanın uygulanması gerekir. Ancak üretilen fiziksel gürültü bu etkilerden mutlaka biraz etkileneceğinden, tüm devre ortamında üretilen fiziksel gürültünün aslında bozulmuş gürültü işareti olduğu düşünülmelidir.

2.4.2. Çift Osilatör Örnekleme Yöntemi

Çift osilatörlü yapıda rasgelelik serbest çalışan yavaş osilatörün faz gürültüsü (seğirme (jittered)) sayesinde elde edilmektedir. Bu yöntemde hızlı osilatör D tipi flip flopun işaret girişine uygulanırken faz gürültüsüne sahip yavaş osilatör D tipi flip flopun saat girişine uygulanmakta ve faz gürültüsünün rasgeleliği sayesinde hızlı osilatör rastgele örneklenebilmektedir. Çift osilatörlü yapıdaki rasgelelik için hızlı ve yavaş osilatörün oranları uygun seçilmelidir. Yapıda seçirmeli yavaş osilasyonu üretmek için kullanılan alt devreler (ring osilatör, PLL, ısı gürültü, termal veya johnson gürültüsü, saçma gürültüsü, kaotik işaret, Metastability vs.) değişmektedir. Şekil 2.8'de basit bir çift osilatör yapısı verilmiştir.



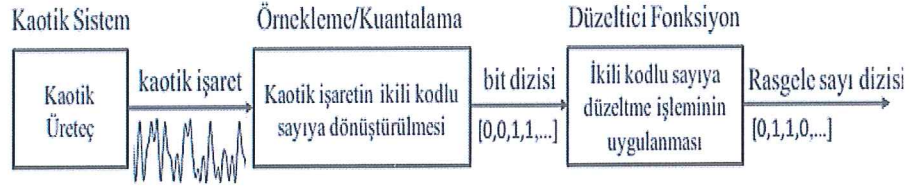
Şekil 2.8. GRSÜ kullanılan çift osilatör örnekleme yapısı [54].

Bazı durumlarda yavaş osilatörün çıkışının seçirmesi yeterli rastgele dağılımı gösterememekte, gürültü kaynaklı ya da gerilim kontrollü osilatörler kullanılarak seçirme seviyesi arttırılmaktadır. Bu tür tümdevre yapılarında entropi kaynağı genelde ring osilatör kullanılarak gerçekleştirilmektedir. Bu yapının çalışma kriterlerine bakıldığında yavaş osilatörün frekansı çıkış hızını belirlemektedir ve bu frekansı düşük tutmak çıkış hızını düşürmektedir. Çıkış hızını arttırmak için yavaş osilatörün frekansı arttırılmalıdır. Ancak buna bağlı olarak hızlı osilatörün frekansını da arttırmak gerekir ki bu işlem tümleşik yapıların tasarımını zorlaştırmaktadır. Sadece osilatör örnekleme yöntemi kullanılarak gerçekleştirilen RSÜ etkili bir rasgelelik sağlayamamaktadır. Bundan dolayı sisteme başka bir belirsizlik entropi kaynağı daha eklenmelidir. Entropi kaynağı olarak direnç ısı gürültüsü veya kaotik işaret üreten devreler kullanılabilir. Literatürde çok değişik şekillerde çift osilatör yöntemine dayalı çalışmalar mevcuttur [47,51,54,55,61,81,82,85–89].

2.4.3. Kaotik Devre Tabanlı

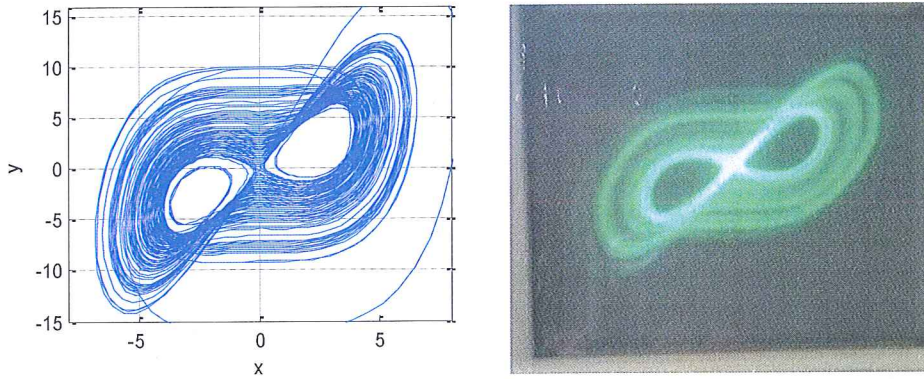
Son yıllarda yapılan çalışmalarda, kaotik sistemlerin GRSÜ devrelerinin gerçekleştirilmesinde gürültü yerine kullanılabileceği görülmüştür. Bu yapılarda kullanılan kaotik sistemler başlangıç değerlerine üstel olarak duyarlı sistemler olup, bunların çözümlerinin uzun zaman aralıkları için öngörülmesi mümkün olmamaktadır. Bunun yanında kaotik işaretlerin periyodik olmamaları nedeniyle, frekans spektrumları sürekli ve geniştir. Tüm bu özellikler kaotik sistemlerin yüksek başarılı RSÜ'lerin de kullanılabileceğini göstermektedir. Şekil 2.9'da kaos üreticisine dayalı RSÜ'lerin genel yapısı verilmiştir. Bu sistemde öncelikle osilatör

çıkışı belirli bir frekansta örneklenir, çıkış değerleri kuantalanır, daha sonra da elde edilen bit dizisine düzeltici algoritmalar uygulanarak rastgele sayı üretimi sağlanır.



Şekil 2.9. Kaotik rastgele sayı üreten sistemin blok diyagramı [133].

Kaos tanımı incelendiğinde başlangıç koşullarına üstel duyarlı, doğrusal olmayan, deterministik karakterli, uzun vadede periyodik olmayan dinamik sistemler olduğu görülmektedir. Düzensiz davranışları ve başlangıç koşullarına aşırı hassas oluşları nedeniyle, kaotik işaretler de rasgelelik kaynağı olarak değerlendirilmekte ve RSÜ yapımında kullanılmaktadırlar. Şekil 2.10'da örnek bir kaotik devrenin faz eğrisi verilmektedir.



Şekil 2.10. Örnek bir kaotik devrenin x-y faz portresinin Matlab ve osilaskop çıkışı.

Kaotik sistemleri otonom ve otonom olmayan kaotik sistemler olarak iki grupta sınıflandırabiliriz. Otonom olmayan kaotik osilatörler zamana bağlı periyodik bir kaynak fonksiyonu $f(t)$ içeren lineer olmayan bir diferansiyel denklem sistemi ile modellenebilmektedir. Otonom bir sistemin kaos üretebilmesi için Poincare Bendixson Teoremi sonucu olarak sistemin en azından üçüncü dereceden olması gerekmektedir. Otonom olmayan sistemlerde ise diferansiyel denklemin sağ tarafı " t " zaman değişkenine bağlı özel bir fonksiyon içermektedir. Bu zaman değişkeni

fazladan bir durum deęiřkenine karřı geldięinden, otonom olmayan sistemlerin kaos üretebilmesi için bunların en azından ikinci dereceden olmasının gerektięi sonucuna kolaylıkla varılabilir.

Bu tez çalışmasında; FPGA üzerinde gerçek zamanlı, farklı nümerik diferansiyel denklem çözüm yöntemleriyle yüksek bit üretim hızına sahip yeni kaotik sistemler ile yüksek çalışma frekanslı halka osilatör kullanılarak çift entropi çekirdekli Gerçek Rasgele Sayı Üreteçlerinin tasarlanması ve gerçekleştirilmesi hedeflenmektedir. Bunun yanında yukarıda zikredilen rastgele sayı üretiminde kullanılan yöntemlerin sorunlarına çözüm bulmaktır. Tezin sunulduęu tarih itibarı ile literatürdeki FPGA tabanlı kaotik GRSÜ çalışmalarından farklı olarak ayrık zamanlı çift entropi çekirdekli olması ve VHDL dilinde IQ-Math Fixed Point Number (Sabit Noktalı Sayı) formatında tasarımın gerçekleştirilmesi olacaktır. Literatürdeki çalışmalardan farklı olarak;

1. Tezin birinci özgün yönü FPGA tabanlı yeni kaotik osilatörler tasarlayarak oluşturulacak GRSÜ'nin çift entropi çekirdekli yeni bir yapı olmasıdır. Bu yapı sayesinde kaotik sistem ile halka osilatörün avantajlarının birleştirilmesi hedeflenmektedir. Tezde önerilen yeni yapının daha güvenli, testlerde daha başarılı, yüksek çalışma frekansına ve bit üretim hızına sahip olması beklenmektedir.
2. Tezin ikinci özgün yönü önerilen sistemin FPGA üzerinde VHDL dilinde tasarlanmasında IQ-Math sabit noktalı sayı formatının kullanılmasıdır. Literatürdeki IEEE 754-1985 kayan nokta sayı formatındaki tasarımlarına nazaran sabit noktalı sayı formatının daha az FPGA çip istatistikleri kullanacağı ve daha yüksek çalışma frekansına ve bit üretim hızına sahip olacağı beklenmektedir.

BÖLÜM 3

KAOS ANALİZ YÖNTEMLERİ VE LİTERATÜRE SON YILLARDA YENİ SUNULAN ÜZERİNDE FPGA TABANLI ÇALIŞMA YAPILMAMIŞ KAOTİK SİSTEMLER

Kaotik sistem denklemlerle ifade ediliyorsa deterministik kaos olarak adlandırılır. Kaotik sistemlerin matematiksel modelleri doğrusal olmayan bir yapıya sahiptir. Hem sürekli zamanlı diferansiyel denklemlerle hem de ayrık zamanlı fark denklemleri ile ifade edilebilmektedir. Sistemlerin matematiksel modellerinin olması determinizm kavramını ortaya koyarken uzun dönemdeki davranışın kestirilemez olması; kaotik sistemleri doğrusal olmayan sistemler altında incelenen birçok modelden farklı kılmaktadır. Kaotik sistem modelleri basit olmasına karşın davranışları çok karmaşık gözükmemektedir. Kaotik sistemlerden elde edilen verilere geleneksel birçok istatistiki test uygulandığında üretilen zaman dizilerinin analizini geleneksel yöntemlerle yapmak yetersiz kalmaktadır. Bu nedenle doğrusal olmayan zaman dizilerinin analizlerinde kaotik analiz algoritmaları ön plana çıkmaktadır [94].

Bu bölümde ilk aşamada kaotik sistemlerinin dinamik davranışlarını belirlemek için kullanılan kaos analiz yöntemlerinin özelliklerinden bahsedilecektir. Daha sonra üzerinde FPGA tabanlı bilimsel çalışmalar yapılmamış son yıllarda literatüre yeni sunulan kaotik çekerler araştırılacak ve bu sistemlerin dinamik davranışlarını belirlemek amacıyla kaos analizleri yapılacaktır. Seçilen örnek kaotik sistemlerin dinamik davranışlarını belirlemek için farklı nümerik diferansiyel denklem çözümleri kullanılarak benzetim ve analiz programı Matlab'ta modellenecektirler.

3.1. KAOTİK ANALİZ YÖNTEMLERİ

Kaos ve kaotik işaretlerin başlıca özellikleri; başlangıç şartlarına aşırı duyarlılık göstermesi, zaman boyutunda düzensizliği, rasgele değil deterministik tipte olması,

sınırsız sayıda deęişik periyodik salınımlar içermesi, gürültü benzeri güç spektrumuna sahip olması, genlięi ve frekansı tespit edilemeyen ancak sınırlı bir alan içerisinde deęişen karmaşık davranışlar sergilemesidir [95]. Nonlinear dinamik bir sistemin kaotik olduęunun belirlenebilmesi için çeşitli şartlar bulunmaktadır. Bu şartlardan ilki sistemde mutlaka bir veya daha fazla doğrusal olmayan bir elemanın bulunmasıdır. Dięer bir şart ise sürekli zamanlı sistemlerde sistemin en az üçüncü dereceden bir sistem olmasıdır. Eęer sistem ayrık zamanlı ise böyle bir şart bulunmamaktadır. Eęer bir sistem bu iki şartı sağlıyorsa bu sistemde kaos analizi yapılabilir demektir. Literatürde bir sistemin kaotik analizinin yapılabilmesi için geliştirilmiş çeşitli yöntemler bulunmaktadır [96]. Bu yöntemlerden bazıları şunlardır;

1. Yörüngeyi izlenmesi (zaman serileri – Time series),
2. Sistemin faz uzayının incelenmesi (Phase portrait),
3. Lyapunov üstelleri,
4. Lyapunov üstelleri frekans spektrumu,
5. Poincare haritalama,
6. Fraktal Boyut analizi,
7. Güç spektrumu,
8. Çatallaşma diyagramı (Bifurcation diagram) örnek olarak verilebilir [95,96].

İlk ve önemli bir özellik başlangıç şartlarına hassas baęlılıktır ki iki yörünge, başlangıçta birbirine çok yakın da olsa, zamanla birbirinden uzaklaşır ve yörüngeler arasındaki uzaklık genellikle zamana göre üssel olarak artar. Kaotik davranışın dięer bir özellięi de sistemin limit kümesinin fraktal (parçalı) olmasıdır. Denge noktası, limit çevrim ve torusda boyut bir tamsayı iken kaotik bir sistemin boyutu parçalıdır. Kaotik sistemler temelde deterministik olsalar bile, kaotik sinyallerin rastgele süreçlerle benzer karakteristiklere sahip olduęu görülmüştür. Kaotik bir davranış dięer davranışlardan ayıran dięer göstergeler, faz resmi görünümü ve frekans spektrumudur. Kaotik yapıya sahip sistemlerde faz resminin zaman gelişimi, dinamik sistemin yapısının belirledięi faz uzayı bölgesinde, sayılamayacak kadar yörüngeyle dolması şeklinde olur. Zaman ilerledikçe, yörüngeler faz uzayını doldurmaya başlar

ve hiçbir zaman üzerine kapanmaz, tekrar eder. Faz uzayının bu şekilde dolması kaotik işaretlerden biridir.

Lyapunov üstelleri yöntemi, Aleksandra Mikhailovich Lyapunov tarafından geliştirilmiş olup, sistemin zaman serisinin kaotik bileşenler içerip içermediğini gösteren matematiksel bir analiz yöntemidir [96,97]. Lyapunov üsteli, başlangıçta birbirine yakın olan yörüngelerin ıraksama ya da yakınsama oranını ölçer. Bazı kaotik sistemler üzerinde aynı denemeler iki kez tekrarlanırsa, bir sonrakinde tamamen farklı sistem değişimleri gözlenebilir. Lyapunov üsteli yöntemi, kaotik sistemlerin en önemli özelliklerinden birisi olan “başlangıç şartlarına hassas bağlılık” özelliğinin sayısal bir ifadesidir. Eğer üç boyutlu bir faz uzayında Lyapunov yöntemi ile kaotik analizi yapılan sistemin üstellerinin ($\lambda_1, \lambda_2, \lambda_3$) işaretleri sırasıyla (+, 0, -) ise sistem kaos durumundadır. Dört boyutlu bir sistemde (+, 0, -, -) ve (+, +, 0, -) olmak üzere iki mümkün durum vardır. (+, +, 0, -) durumu hiper Kaos olarak adlandırılmaktadır [96].

Faz uzayı içindeki takriben bütün yörüngelere karşılık gelen eğriler planına Poincare kesit denir. Dinamik bir sistemin faz uzayındaki davranışı, faz uzayı belirli bir düzlemle kesilerek (Poincare kesit), yörüngelerin o düzlemi kestiği noktaların oluşturduğu bir geometrik harita şeklinde de gözlemlenebilir. Poincare kesitteki noktaların dağılımı tek ve küçük bir bölgede sonlu sayı da ise hareket periyodik, kapalı bir eğri ise hareket yarı periyodik, belirli alanlarda yoğunlaşmış kümeler şeklinde ise hareket kaotiktir [99].

Fraktal geometri, bilimsel literatürde Mandelbrot tarafından ortaya konan bir terimdir. Mandelbrot, doğal objelerin yapısal bütünlüğünü incelemiştir. Bir nokta, bir çizgi, bir kare veya bir küp gibi matematiksel objeler sırasıyla sıfır, bir, iki veya üç boyutlu objeler olarak bilinirler. Toplam boyut sayısı objenin geometrik yapısı hakkında bize bilgi verir. Mandelbrot, tamsayı ile ifade edilen boyutların kaotik yapılara uygun olmadığını göstermiştir. Doğrusal sistem çekerleri tamsayı boyutlarla ifade edilebilirken, kaotik sistem çekerleri fraktal (kesirli) boyutlara sahiptir [99].

3.2. SON YILLARDA LİTERATÜRDEKİ YENİ KAOTİK SİSTEMLER

Günümüzde son birkaç yılda kaosun uygulama alanlarının gelişmesi ile literatürde yeni kaotik osilatör tasarımlarına yönelik araştırmalarda önemli artışlar görülmektedir [89–95,99]. Pehlivan 2007 yılında yaptığı doktora tez çalışmasında yeni birçok kaotik sistemini tanıtmış ve devrelerin simülasyon, senkronizasyon ve elektronik devre gerçeklemelerini yaparak literatüre yeni garip çekerler kazandırmıştır [95,99].

Abooe vd. 2013 yılında yeni üç boyutlu otonom kaotik sistemini tanıtmışlardır [101]. Yaptıkları çalışmada yeni kaotik sistemin dinamik denklemlerini çıkartarak denge kararsızlığı, garip çeker, başlangıç koşullarına hassasiyeti, Lyapunov üstelleri ve fraktal boyut analizi gibi sistemin bazı temel dinamik özelliklerini incelemişlerdir. Ayrıca sistemin OrCAD-PSpice yazılımında simülasyonu ve analog devre elemanları ile devre gerçekleşmesini yaparak sonuçları karşılaştırmışlardır.

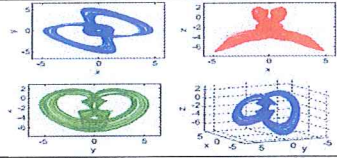
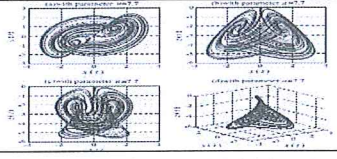
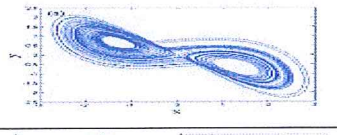
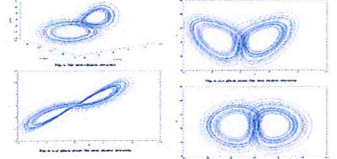
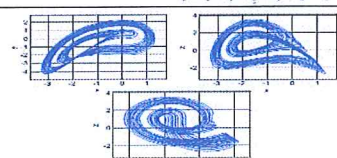
Deng vd. 2014 yılında yeni üç boyutlu 7 terimli, 3 parametrelili ve sadece bir tane pozitif değer içeren otonom kaotik sistemini sunmuşlardır [102]. Önerdikleri yeni çekerin basit dinamik özellikleri olan faz portresini, denge noktasını, Lyapunov üstellerini, Poincare haritalama ve çatallaşma diyagramını incelemişlerdir.

Zhoua vd. 2008 yılında üç boyutlu 6 terimli, 3 parametrelili yeni otonom bir kaotik çeker önemişlerdir [103]. Önerdikleri sistemin Lyapunov üstelleri, Poincare haritalama, fraktal boyut, çatallaşma diyagramı, sürekli spektrumu gibi dinamik davranışlarını hem nümerik hem de analitik olarak incelemişlerdir.

Pehlivan vd. 2012 yılında üç boyutlu sürekli zamanlı 8 terim, 2 parametre ve 2 kuadratik doğrusalsızlık içeren yeni bir otonom kaotik sistemi tanıtmışlardır [104]. Bu kaotik sistemin dinamik yapısını ve dinamik davranışlarını analitik ve nümerik olarak incelemişlerdir. Ayrıca kaotik devrenin Matlab ve OrCAD-PSpice simülasyon sonuçlarını doğrulamak ve devrenin uygulanabilirliğini göstermek için elektronik elemanlarla devreyi gerçekleştirerek sonuçları karşılaştırmışlardır. Çizelge 3.1’de

literatüre son yıllarda yeni sunulan kaotik sistemlerin teknik özellikleri, diferansiyel denklemleri ve faz portreleri verilmiştir.

Çizelge 3.1. Son yıllarda literatürdeki yeni kaotik osilatör tasarımları.

Çalışmayı Yapan	Kaotik Sistem Özellikleri	Diferansiyel Denklemleri	Faz Portreleri
Pehlivan, [95,99]	3 boyutlu 8 terimli 1 parametrelili Otonom	$\dot{x} = y \cdot (1 - z)$ $\dot{y} = -a \cdot x + y \cdot (z + 1)$ $\dot{z} = a - x \cdot y - y^2$	
Abooe vd., [101]	3 boyutlu 7 terimli 6 parametrelili Otonom	$\dot{x} = a(y - x) + byz^2$ $\dot{y} = cx + dxz^2$ $\dot{z} = hz + kx^2$	
Deng vd., [102]	3 boyutlu, 7 terimli, 3 parametrelili Otonom	$\dot{x} = -x - 2 \cdot y$ $\dot{y} = -x \cdot z - b \cdot y - a \cdot x$ $\dot{z} = x \cdot y - c \cdot z$	
Zhoua vd., [103]	3 boyutlu 6 terimli 3 parametrelili Otonom	$\dot{x} = a \cdot (y - x)$ $\dot{y} = b \cdot x - x \cdot z$ $\dot{z} = x \cdot y + c \cdot z$	
Pehlivan vd., [104]	3 boyutlu 8 terimli 2 parametrelili Otonom	$\dot{x} = y - x - a \cdot z$ $\dot{y} = x \cdot z - x$ $\dot{z} = -x \cdot y - y + b$	

Literatüre son yıllarda sunulan üzerinde FPGA tabanlı donanımsal çalışma yapılmamış Çizelge 3.1’de kısaca bahsedilen farklı özelliklere sahip kaotik sistemler detaylı bir şekilde incelenerek sırasıyla kaos analizleri gerçekleştirilmiştir.

3.2.1. Yeni Pehlivan C Kaotik Sistemi (PCKS) (2007)

Günümüzde üzerinde pek çok araştırma ve çalışmalar yapılan doğrusal olmayan sistemler alanından birisi de kaos bilimi veya kaotik sistemlerdir. Kaotik çekerler başlangıç koşullarına hassas bağlı, karmaşık ve düzensiz görünümlü olmakla birlikte bu çekerler deterministik doğrusal olmayan zamanla değişen sürekli ve ayrık sistemlerde ortaya çıkarlar. Kaotik sistemlerin araştırılması ve uygulanmasına yönelik bilimsel ve endüstriyel alanlarda önemli çalışmalar gerçekleştirilmektedir.

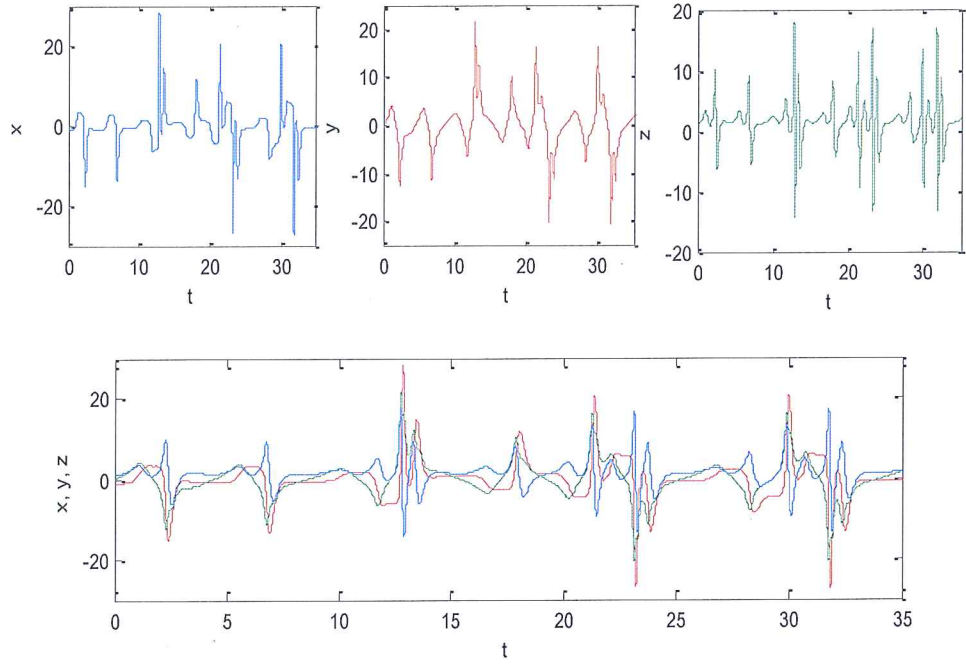
Bu amaçla Pehlivan vd. tarafından yeni tasarlanan PCKS'nin matematiksel ifadesi Eşitlik 3.1'de verilmiştir [95,104]. Yeni tasarlanan bu kaotik sistemde sistem parametreleri ρ ve σ Eşitlik 3.2'de ve kaotik sistemin başlangıç şartları (x_0, y_0, z_0) Eşitlik 3.3'te verilmiştir.

$$\begin{aligned}\frac{dx}{dt} &= y \cdot (z - \rho) \\ \frac{dy}{dt} &= y \cdot (z - \rho) - x \cdot (z + \rho) \\ \frac{dz}{dt} &= -y \cdot (\rho \cdot x - y) - \sigma \cdot (z - \rho)\end{aligned}\tag{3.1}$$

$$\rho = 1,30, \sigma = 4,0\tag{3.2}$$

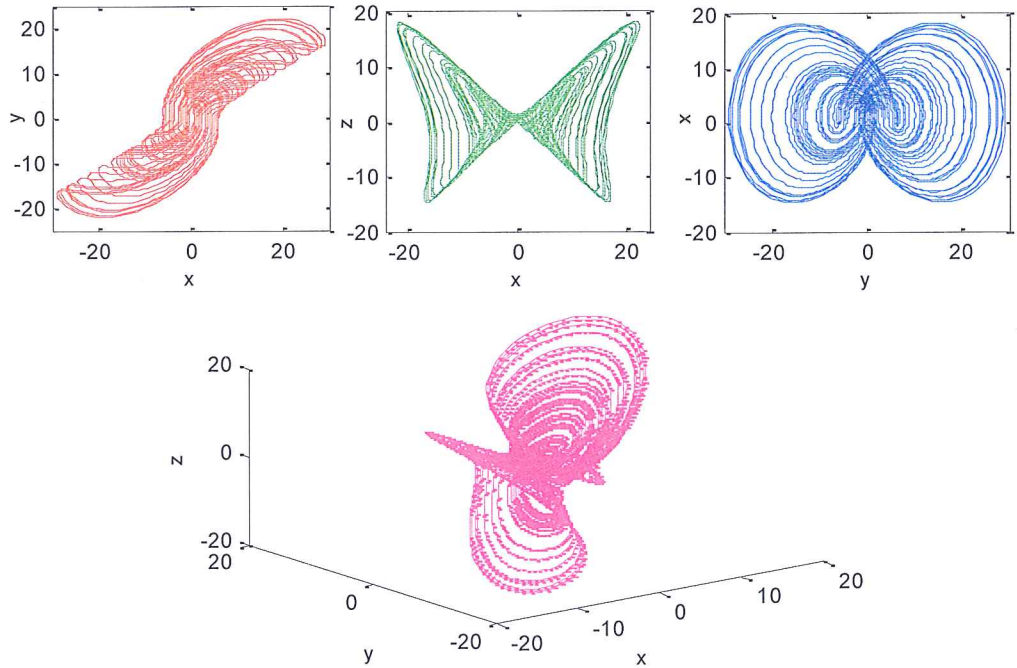
$$x_0 = -1, y_0 = 0, z_0 = 1\tag{3.3}$$

Tasarlanan yeni kaotik sistem 5 adet ikinci dereceden doğrusal olmayan ifade (yz, xz, yx, y^2) olmak üzere 10 terim ve 2 parametre içermektedir. Doğrusal olmayan diferansiyel denklem sisteminin çözümü özellikle başlangıç şartlarına ve sistem parametrelerine bağlıdır. Bu sistemin nümerik olarak çözümünde Heun (iyileştirilmiş Euler) algoritması kullanılmıştır. Matlab programında nümerik analiz sonucunda Şekil 3.1'de yeni kaotik sistemin x, y, z kaotik durum değişkenlerinin zaman göre değişimi görülmektedir.



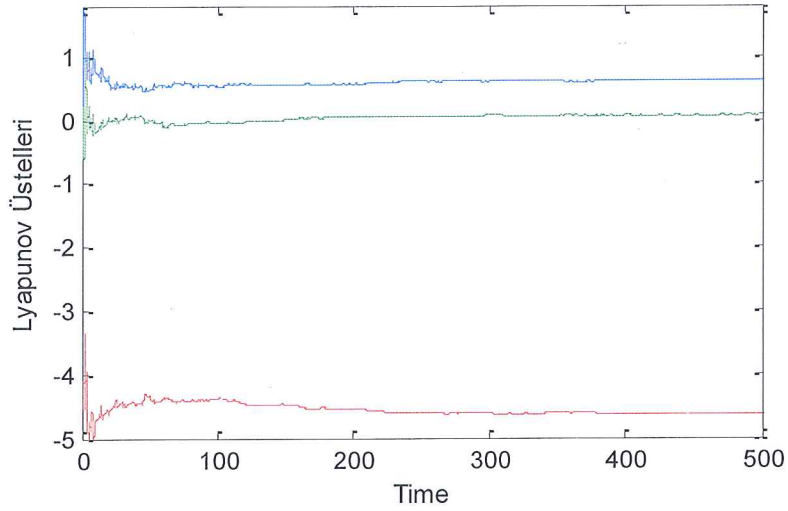
Şekil 3.1. Yeni PCKS'nin x , y , z kaotik durum değişkenlerinin zamana göre değişimi.

Aynı parametre ve başlangıç şartlarındaki x - y , x - z ve y - z kaotik faz portreleri ile üç boyutlu x - y - z yörüngesi Şekil 3.2'de görülmektedir.



Şekil 3.2. Yeni PCKS tasarımının x - y , x - z , y - z kaotik çekicileri ve üç boyutlu x - y - z yörüngesi.

Tasarlanan PCKS'nin kaos analizi için gerekli Lyapunov üstelleri Matlab programında geliştirilen Lyapunov Exponent Toolbox (LET) kullanılarak elde edilmiştir [106]. Lyapunov Exponent Toolbox (LET) kullanılarak elde edilen Lyapunov üstelleri Şekil 2.3'te verilmiştir. Görüldüğü gibi Lyapunov üstelleri $\lambda_1=+0,53860$, $\lambda_2=0$, $\lambda_3=-4,4030$ olarak bulunmuştur. Buradan kaotik davranışın oluşması için gereken (+, 0, -) durumunun sağlandığı ve sistemin kaotik olduğu görülmektedir.



Şekil 3.3. Yeni tasarlanan PCKS'nin Lyapunov üstelleri.

3.2.2. Yeni Pehlivan 3 Boyutlu Kaotik Sistemi (P3BKS) (2012)

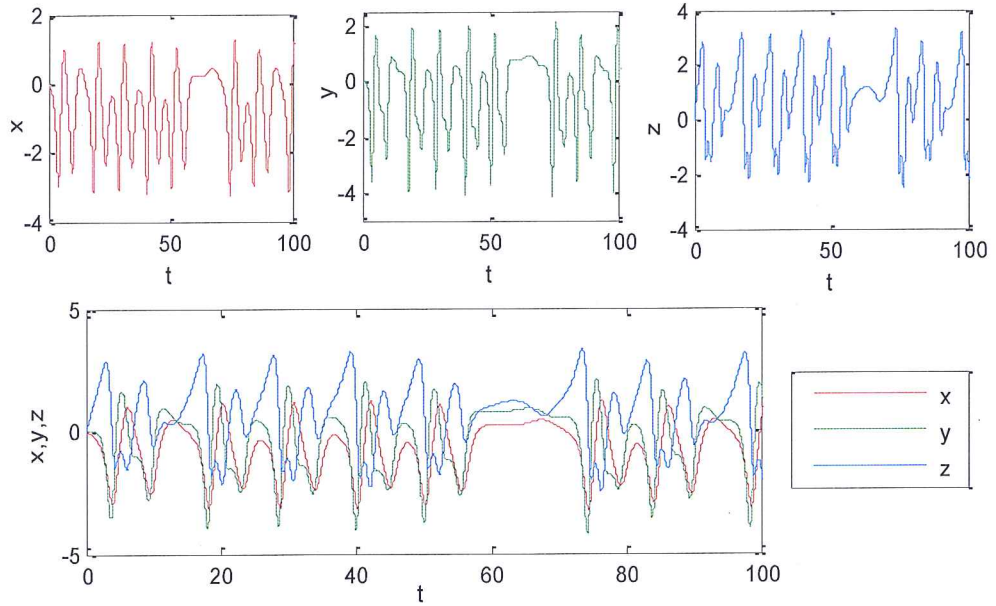
Pehlivan vd. yaptıkları çalışmalarında yeni tasarladıkları üç boyutlu sürekli zamanlı iki kuadratik doğrusalsızlık içeren otonom bir P3BKS'ni tanıtmışlardır [100]. Kaos uygulamaları için önerilen yeni sistemin matematiksel modeli Eşitlik 3.4'te verilmiştir. Matematiksel modelde verilen a ve b sistem parametrelerinin değerleri Eşitlik 3.5'te ve kaotik sistemin başlangıç şartları (x_0, y_0, z_0) da Eşitlik 3.6'da tanımlanmıştır.

$$\begin{aligned}\dot{x} &= y - x - az \\ \dot{y} &= xz - x \\ \dot{z} &= -xy - y + b\end{aligned}\tag{3.4}$$

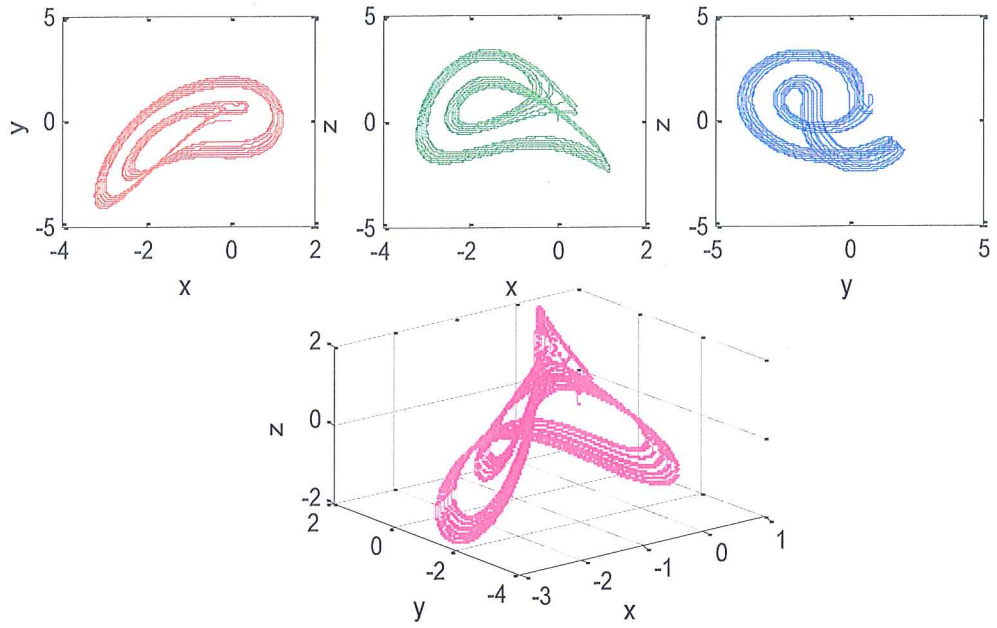
$$a = 0,50, \quad b = 1,0 \quad (3.5)$$

$$x_0 = 0, \quad y_0 = 0, \quad z_0 = 0 \quad (3.6)$$

Pehlivan ve arkadaşlarının önerdiği bu sistem, 2 adet ikinci dereceden doğrusal olmayan ifade (xz ve xy) olmak üzere 8 terim ve 2 parametre içermektedir. Bu sistem Matlab programında RK4 nümerik algoritması kullanılarak modellenmiştir. Nümerik modelden alınan sonuçlardan Şekil 3.4'te sistemin kaotik zaman serileri ve Şekil 3.5'te faz portleri verilmiştir.

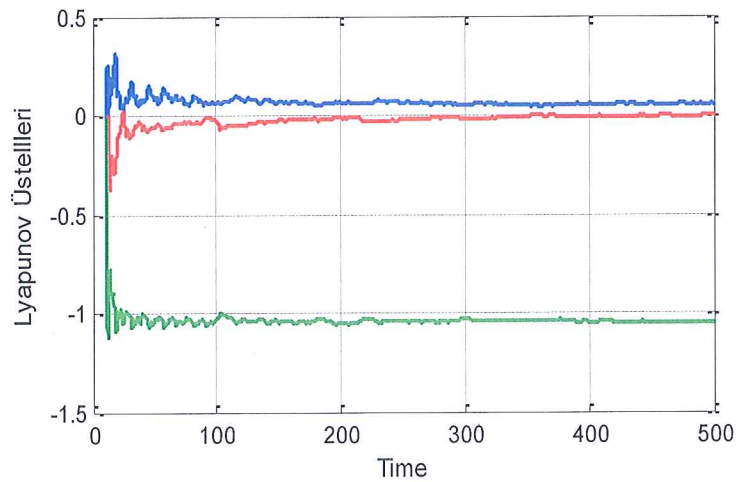


Şekil 3.4. Pehlivan vd. tarafından tasarlanan yeni P3BKS'nin x, y, z ve x-y-z zaman serileri.



Şekil 3.5. Pehlivan vd. tarafından tasarlanan yeni P3BKS'nin x-y, x-z, y-z faz portleri ve 3 boyutlu x-y-z faz yörüngesi.

Pehlivan yeni kaotik sisteminin kaos analizi için gerekli Lyapunov üstelleri Matlab programında geliştirilen Lyapunov Exponent Toolbox (LET) kullanılarak elde edilmiştir. Yapılan Lyapunov üstelleri çalışmasından alınan sonuçlar Şekil 3.6'da verilmiştir. Sonuçlardan da görüleceği üzere Lyapunov üstellerinin işaretlerinin birisi pozitif, diğeri sıfır ve sonuncusu negatiftir. Sonuçlara göre sistemin işaretleri (λ_1 , λ_2 , λ_3) sırasıyla (+, 0, -) olduğundan bu sistemin kaotik olduğu kanıtlanmıştır.



Şekil 3.6. Pehlivan vd. tarafından tasarlanan yeni P3BKS tasarımının Lyapunov üstelleri.

3.2.3. Yeni Abooe 3 Boyutlu Kaotik Sistemi (2013)

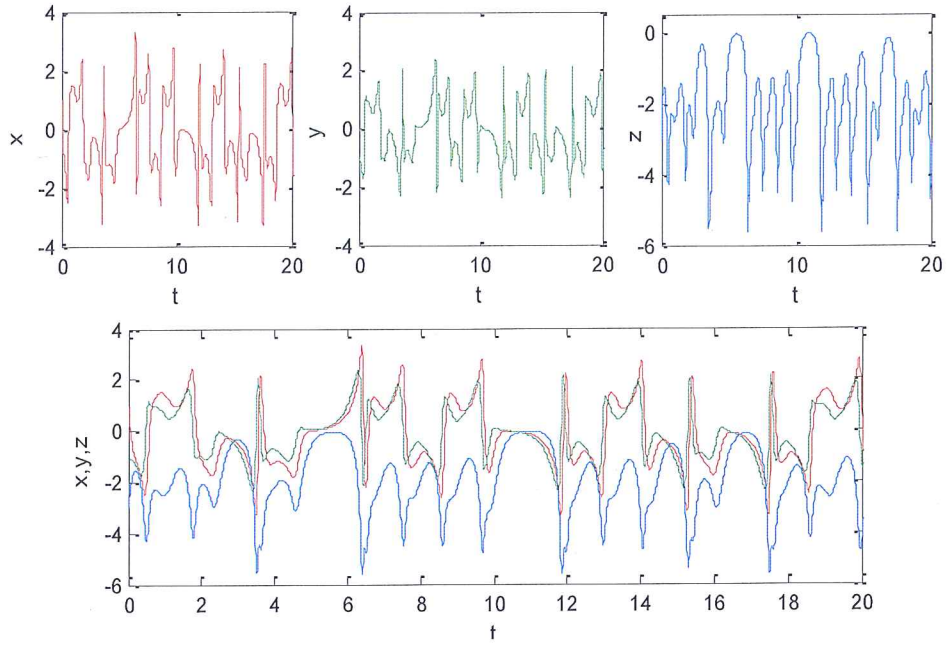
Abooe vd. üç boyutlu otonom yeni kaotik sistemini tanıtmışlardır [101]. Yaptıkları çalışmada yeni kaotik sistemin dinamik denklemlerini çıkartarak denge kararsızlığı, garip çeker, başlangıç koşullarına hassasiyeti, Lyapunov üstelleri ve fraktal boyut analizi gibi sistemin bazı temel dinamik özelliklerini incelemişlerdir. Önerilen sistemin diferansiyel matematiksel modeli Eşitlik 3.7'de verilmiştir. Eşitlik 3.8'de sistemin parametre değerleri (a, b, c, d, h, k) ve Eşitlik 3.9'da ise başlangıç koşulları (x_0, y_0, z_0) verilmiştir.

$$\begin{aligned}\dot{x} &= a \cdot (y - x) + b \cdot y \cdot z^2 \\ \dot{y} &= c \cdot x + d \cdot x \cdot z^2 \\ \dot{z} &= h \cdot z + k \cdot x^2\end{aligned}\tag{3.7}$$

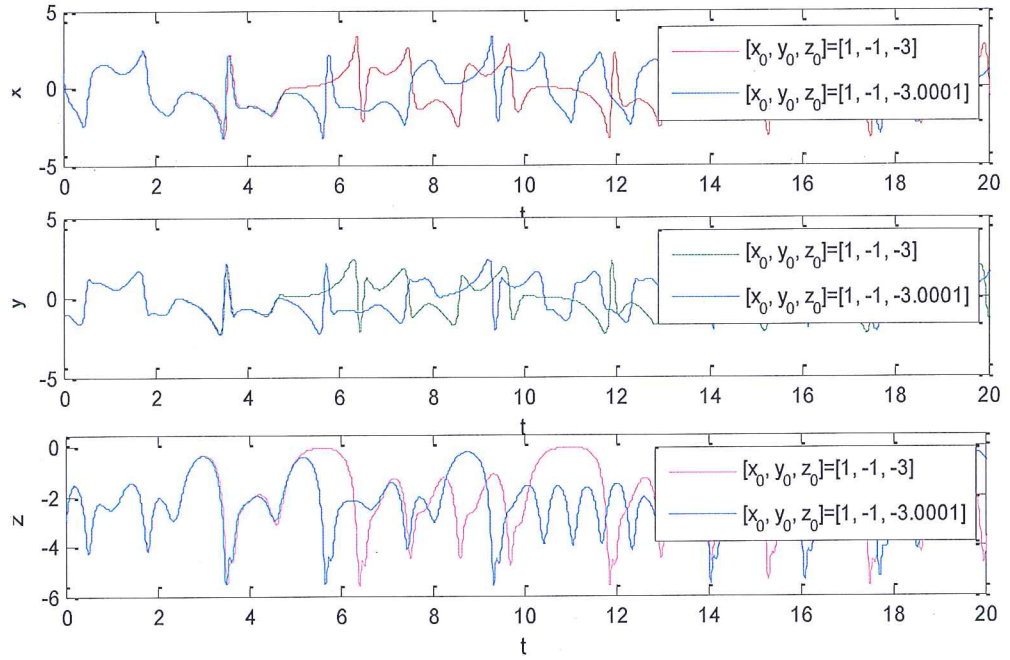
$$a = 7,70, b = 1, c = 5, d = -1, h = -5, k = -6\tag{3.8}$$

$$x_0 = 1, y_0 = -1, z_0 = -3\tag{3.9}$$

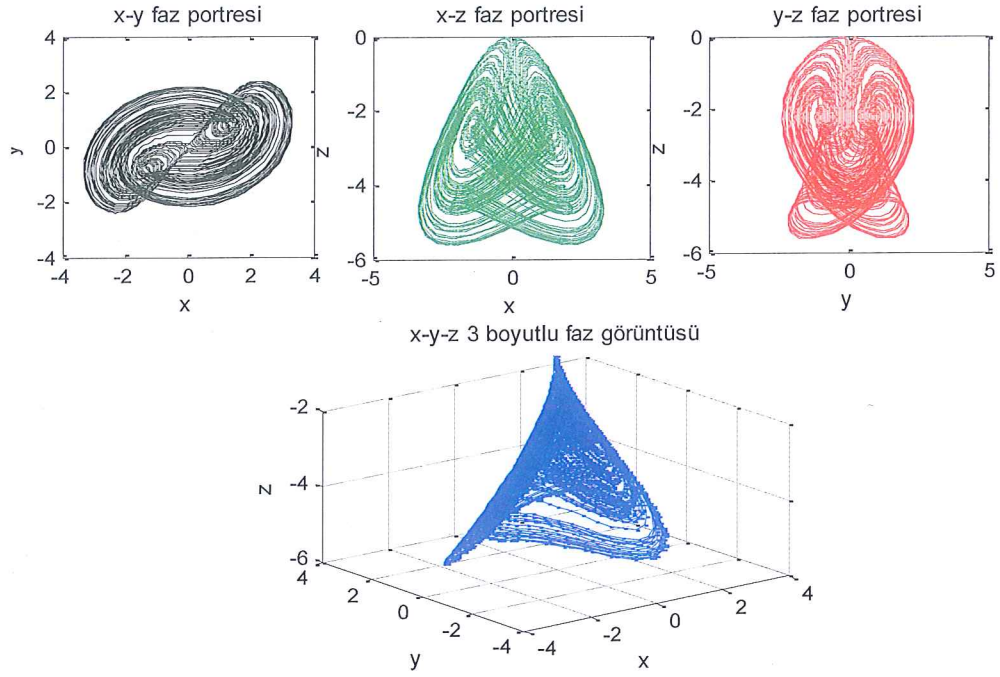
Abooe vd. tarafından tasarlanan bu sistem, 2 adet ikinci dereceden doğrusal olmayan ifade olmak üzere 7 terim ve 6 parametre (a, b, c, d, h, k) içermektedir. Bu sistem Matlab programında nümerik olarak modellenerek kaos analizleri gerçekleştirilmiştir. Şekil 3.7'de tasarlanan sistemin x, y ve z çıkışlarından elde edilen nümerik sonuçlar grafiksel olarak verilmiştir. Ayrıca Şekil 3.8'te başlangıç koşullarındaki ($x_0=1, y_0=-1, z_0=-3$) çok küçük bir değişikliğin ($x_0=1, y_0=-1, z_0=-3,0001$) sistem çıkışları üzerindeki etkisi incelenmiştir. Birbirlerine çok benzer başlangıç koşullarında bile sistemin verdiği cevaplar oldukça farklılık göstermektedir. Şekil 3.9'da tasarlanan kaotik sistemin x-y, x-z, y-z faz portleri ve üç boyutlu x-y-z yörüngesi görülmektedir.



Şekil 3.7. Abooe vd. tarafından tasarlanan yeni kaotik sistemin x , y , z ve x - y - z zaman serileri.

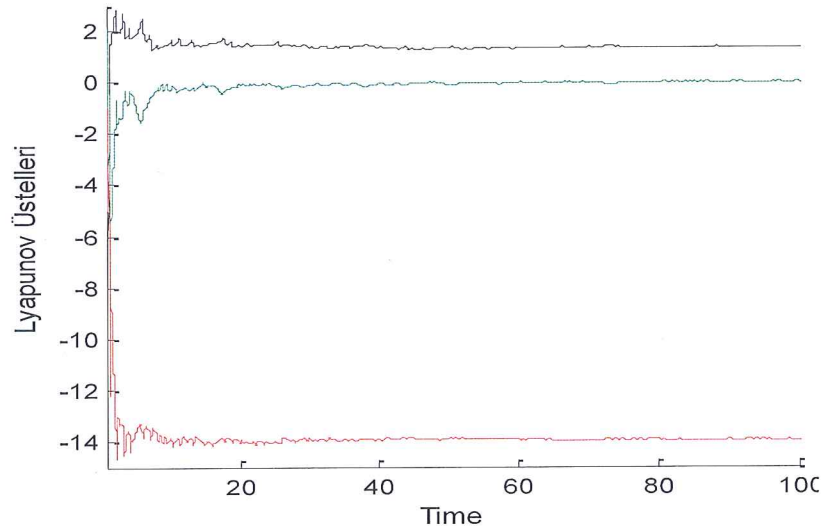


Şekil 3.8. Önerilen yeni kaotik sistemin oldukça benzer başlangıç koşullarındaki $x(t)$, $y(t)$ ve $z(t)$ zaman serileri cevabı.



Şekil 3.9. Abooe vd. tasarladıkları yeni kaotik sistemin x-y, x-z ve y-z kaotik çekicileri ve üç boyutlu x-y-z yörüngesi.

Abooe vd. tarafından tasarlanan yeni kaotik sistemin Lyapunov Exponent Toolbox (LET) kullanılarak elde edilen Lyapunov üstelleri Şekil 3.10'da verilmiştir. Görüldüğü gibi Lyapunov üstelleri $\lambda_1=+1,3380$, $\lambda_2=0$, $\lambda_3=-13,98$ olarak bulunmuştur. Buradan kaotik davranışın oluşması için gereken (+, 0, -) durumunun sağlandığı ve sistemin kaotik olduğu görülmektedir.



Şekil 3.10. Abooe vd. tasarladıkları yeni kaotik sisteminin Lyapunov üstelleri.

3.2.4. Yeni Deng 3 Boyutlu Kaotik Sistemi (2014)

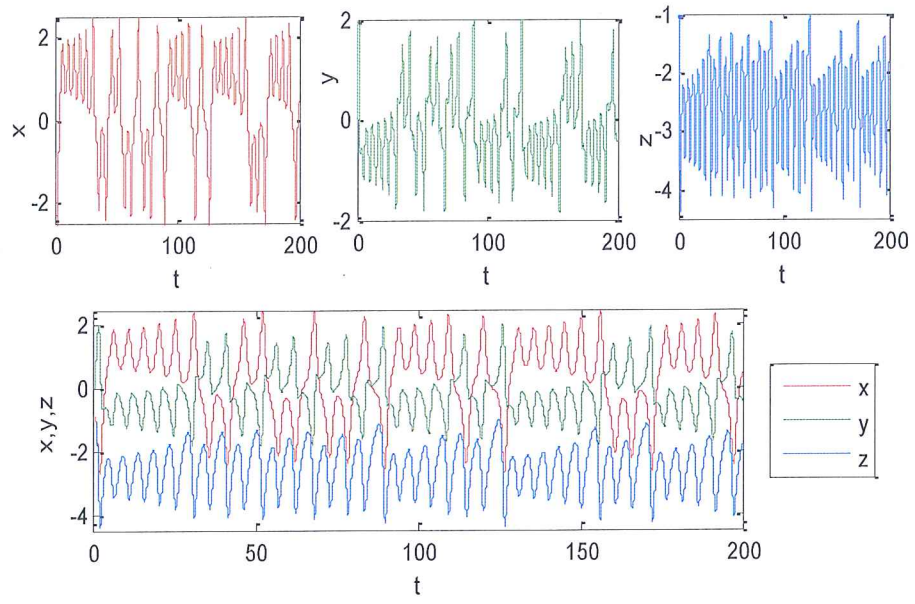
Deng vd. üç boyutlu sadece bir tane pozitif değer içeren otonom yeni kaotik sistemini sunmuşlardır [102]. Yaptıkları çalışmada önerdikleri yeni kaotik çekerin basit dinamik özellikleri olan faz portesini, denge noktasını, Lyapunov üstellerini, Poincare haritalama ve çatallanma diyagramını incelemişlerdir. Önerilen sistemin diferansiyel matematiksel modeli Eşitlik 3.10'da verilmiştir. Eşitlik 3.11'de sistemin parametre (a, b, c) değerleri ve Eşitlik 3.12'de ise kaotik sistemin başlangıç koşulları (x_0, y_0, z_0) verilmiştir.

$$\begin{aligned}\dot{x} &= -x - 2y \\ \dot{y} &= -xz - by - ax \\ \dot{z} &= xy - cz\end{aligned}\tag{3.10}$$

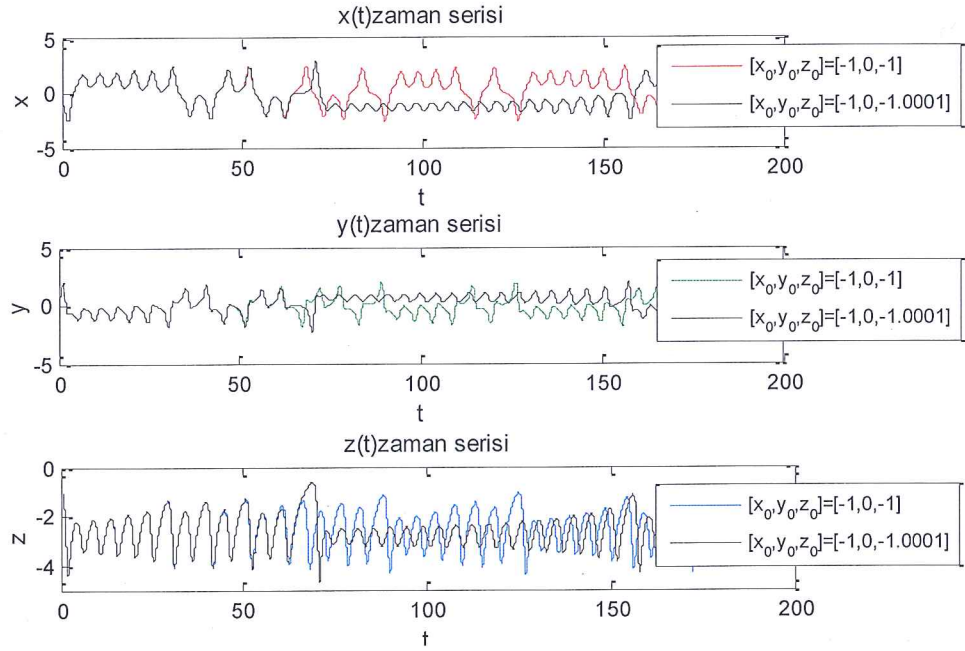
$$a = 3, \quad b = 0,30, \quad c = 0,30\tag{3.11}$$

$$x_0 = -1, \quad y_0 = 0, \quad z_0 = -1\tag{3.12}$$

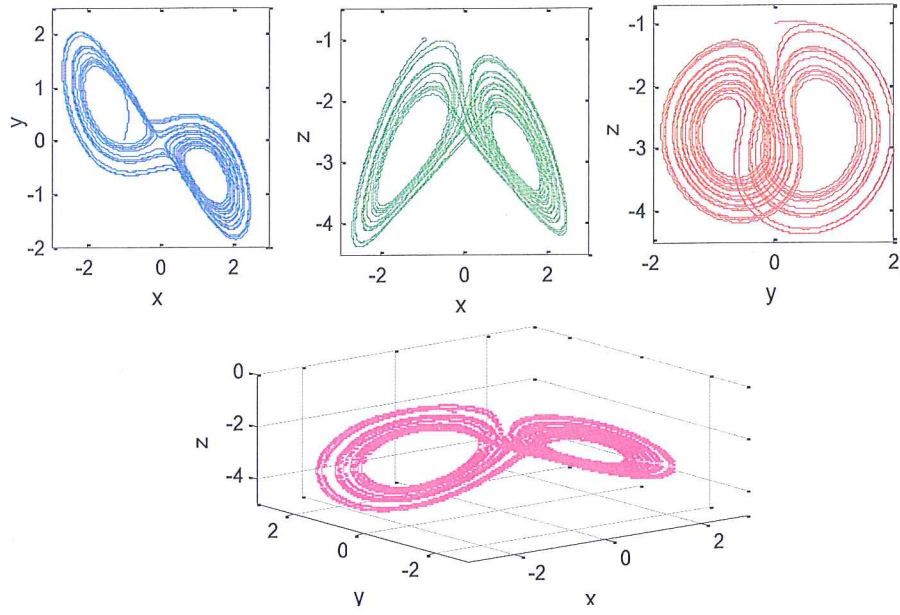
Deng vd. tarafından tasarlanan bu sistem, 2 adet ikinci dereceden doğrusal olmayan ifade (xy, xz) olmak üzere 7 terim ve 3 parametre (a, b, c) içermektedir. Bu sistem Matlab programında nümerik olarak modellenerek kaos analizleri gerçekleştirilmiştir. Şekil 3.11'de tasarlanan sistemin x, y ve z çıkışlarından elde edilen nümerik sonuçlar grafiksel olarak verilmiştir. Ayrıca Şekil 3.12'de başlangıç koşullarındaki $([x_0, y_0, z_0] = [-1, 0, -1])$ çok küçük bir değişikliğin $([x_0, y_0, z_0] = [-1, 0, -1,0001])$ sistem çıkışları üzerindeki etkisi incelenmiştir. Birbirlerine çok benzer başlangıç koşullarında bile sistemin verdiği cevaplar oldukça farklılık göstermektedir. Şekil 3.13'te tasarlanan kaotik sistemin x-y, x-z, y-z faz portleri ve üç boyutlu x-y-z yörüngesi görülmektedir.



Şekil 3.11. Deng yeni kaotik sisteminin x , y , z kaotik durum değişkenlerinin zamana göre değişimi.

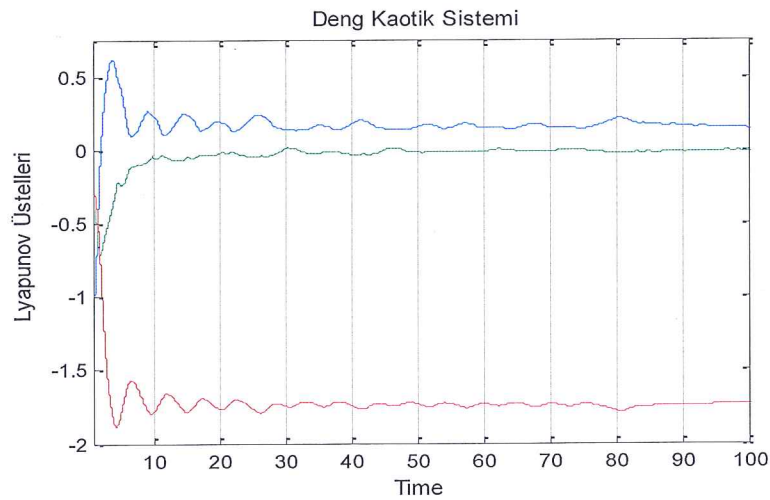


Şekil 3.12. Deng vd. tarafından önerilen yeni kaotik sistemin benzer başlangıç koşullarındaki $x(t)$, $y(t)$ ve $z(t)$ zaman serileri cevabı.



Şekil 3.13. Deng vd. tasarladıkları yeni kaotik sisteminin x-y, x-z, y-z kaotik çekicileri ve üç boyutlu x-y-z yörüngesi.

Bu kaotik sistemin kaos analizi için gerekli Lyapunov üstelleri Matlab programında geliştirilen Lyapunov Exponent Toolbox (LET) kullanılarak elde edilmiştir. Yapılan Lyapunov üstelleri çalışmasından alınan sonuçlar Şekil 3.14'te verilmiştir. Sonuçlardan da görüleceği üzere Lyapunov üstellerinin işaretlerinin birisi pozitif, diğeri sıfır ve sonuncusu negatiftir. Sonuçlara göre sistemin işaretleri ($\lambda_1=0,15030$, $\lambda_2=0$, $\lambda_3=-1,7410$) sırasıyla (+, 0, -) olduğundan bu sistemin kaotik olduğu kanıtlanmıştır.



Şekil 3.14. Deng vd. tarafından tasarlanan yeni kaotik sisteminin Lyapunov üstelleri.

3.2.5. Yeni Zhoua Kaotik Sistemi (2008)

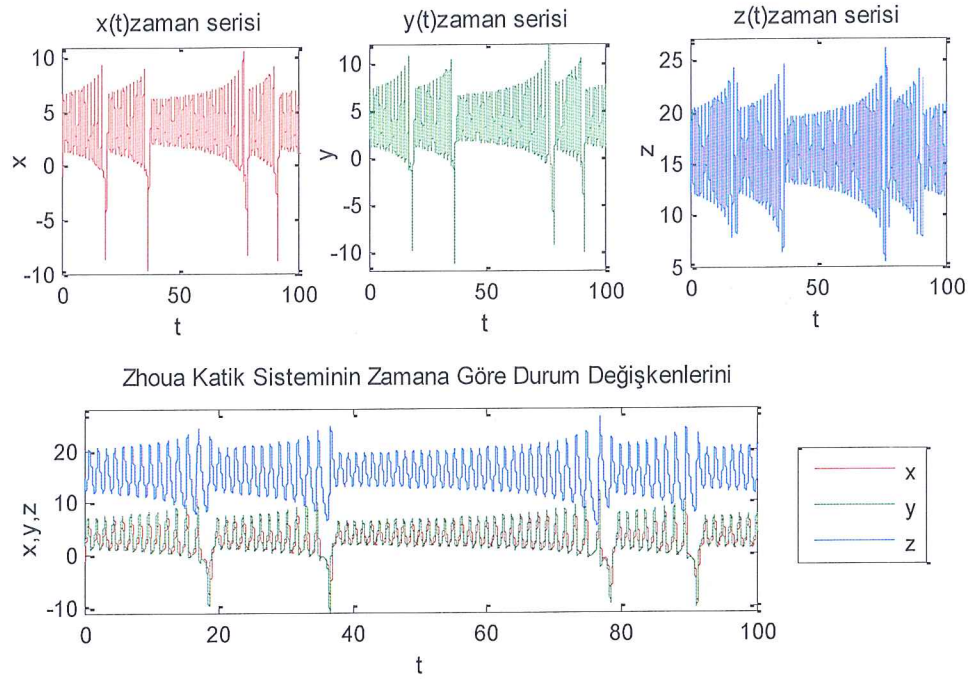
Zhoua vd. üç boyutlu otonom ve kontrollü yeni bir kaotik sistem önermişlerdir [103]. Yaptıkları bu çalışmada önerdikleri yeni kaotik sistemin Lyapunov üstelleri, Poincare haritalama, fraktal boyut, çatallaşma diyagramı, süreklilik spektrumu gibi dinamik davranışlarını hem nümerik hem de analitik olarak incelemişlerdir. Tasarlanan sistemin dinamik denklemleri Eşitlik 3.13'te verilmiştir. Eşitlik 3.14'te sistemin parametre değerleri (a, b, c, d) ve Eşitlik 3.15'te ise başlangıç koşulları (x_0 , y_0 , z_0) verilmiştir.

$$\begin{aligned}\dot{x} &= a \cdot (y - x) \\ \dot{y} &= b \cdot x - x \cdot z + d \\ \dot{z} &= x \cdot y + c \cdot z\end{aligned}\tag{3.13}$$

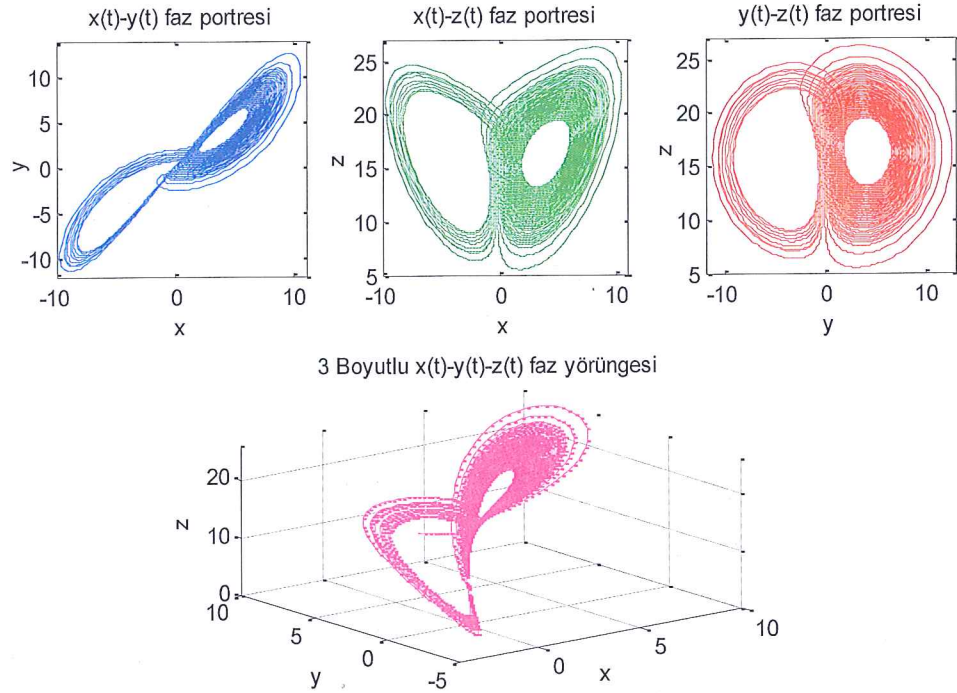
$$a = 10, b = 16, c = -1, -2 < d < 2\tag{3.14}$$

$$[x_0, y_0, z_0] = [-1, 2, 15]\tag{3.15}$$

Zhoua vd. tarafından tasarlanan bu yeni kaotik sistem, 2 adet ikinci dereceden doğrusal olmayan ifade (xy, xz) olmak üzere 7 terim ve 4 parametre (a, b, c, d) içermektedir. Tasarladıkları bu yeni kaotik sistemin kaotikliği d parametresi ile kontrol edilmektedir. $-2 < d < 2$ aralığında farklı kaotik değerler elde edilerek d parametresinin bu sistem için önemli bir parametre olduğunu vurgulamışlardır. Bu sistem Matlab programında nümerik olarak modellenerek kaos analizleri gerçekleştirilmiştir. Şekil 3.15'te tasarlanan sistemin zamanla değişen x, y ve z çıkışlarından alınan sonuçlar grafiksel olarak verilmiştir. Şekil 3.16'da tasarlanan kaotik sistemin x-y, x-z, y-z faz portleri ve üç boyutlu x-y-z yörüngesi görülmektedir.



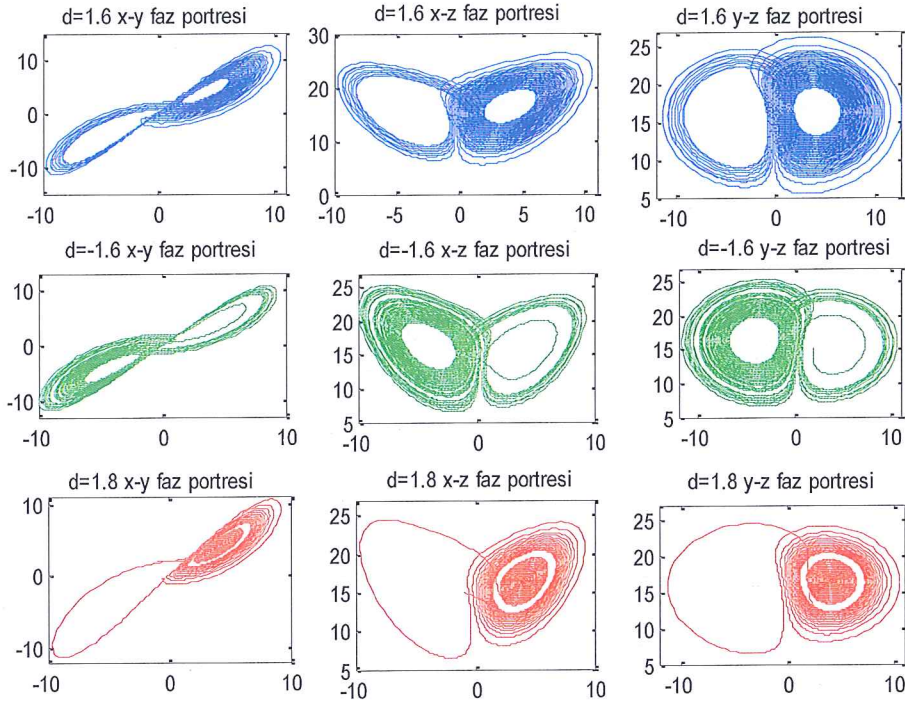
Şekil 3.15. Zhoua yeni kaotik sisteminin x , y , z kaotik durum değişkenlerinin zamana göre değişimi.



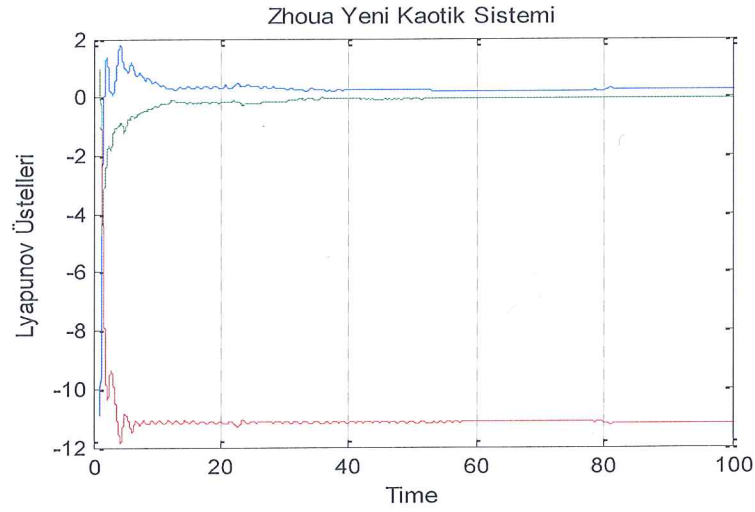
Şekil 3.16. Zhoua yeni kaotik sisteminin x - y , x - z , y - z kaotik çekicileri ve üç boyutlu x - y - z yörüngesi.

Bu kaotik sisteminde “ d ” kontrol parametresinin değişimine göre bu sistemin kaos davranışının değişimi Şekil 3.17’de verilmiştir. Şekillerde görülebileceği gibi “ d ”

büyüdükçe kaos durumunun kaybolduğu, “d” küçüldükçe kaosun ortaya çıktığı anlaşılmaktadır. Onun için bu sistemde “d” önemli bir parametredir. Zhoua yeni kaotik sistemin kaos analizi için gerekli Lyapunov üstelleri Matlab programında geliştirilen Lyapunov Exponent Toolbox (LET) kullanılarak elde edilmiştir. Yapılan Lyapunov üstelleri çalışmasından alınan sonuçlar Şekil 3.18’de verilmiştir. Sonuçlardan da görüleceği üzere Lyapunov üstellerinin işaretlerinin birisi pozitif, diğeri sıfır ve sonuncusu negatiftir. Sonuçlara göre sistemin işaretleri ($\lambda_1=0,30290$, $\lambda_2=0$, $\lambda_3=-11,150$) sırasıyla (+, 0, -) olduğundan bu sistemin kaotik olduğu kanıtlanmıştır.



Şekil 3.17. Zhoua yeni kaotik sisteminin “d” kontrol parametresinin değişimine ($d=1,6$, $d=-1,6$, $d=1,8$) göre kaos durumu.



Şekil 3.18. Zhoua yeni kaotik sisteminin Lyapunov üstelleri.

3.2.6. Yeni Pehlivan 3 Boyutlu Kelebek Şeklinde Kaotik Çekicisi (2014)

Pehlivan vd. tarafından yapılan çalışmada üç boyutlu sürekli zamanlı 4 adet ikinci dereceden doğrusal olmayan ifade içeren otonom yeni bir kaotik sistemi tanıtmışlardır [104]. Tasarladıkları sistem belirli parametreler içerisinde periyodik ve kaotik özellik göstermektedir. Kaos uygulamaları için önerilen yeni sistemin matematiksel modeli Eşitlik 3.16’da verilmiştir. Matematiksel modelde ifade edilen kaotik sistemin sistem parametreleri (a, b, c, d) Eşitlik 3.17’de ve Eşitlik 3.18’de ise kaotik sistemin başlangıç şartları (x_0, y_0, z_0) ifade edilmektedir.

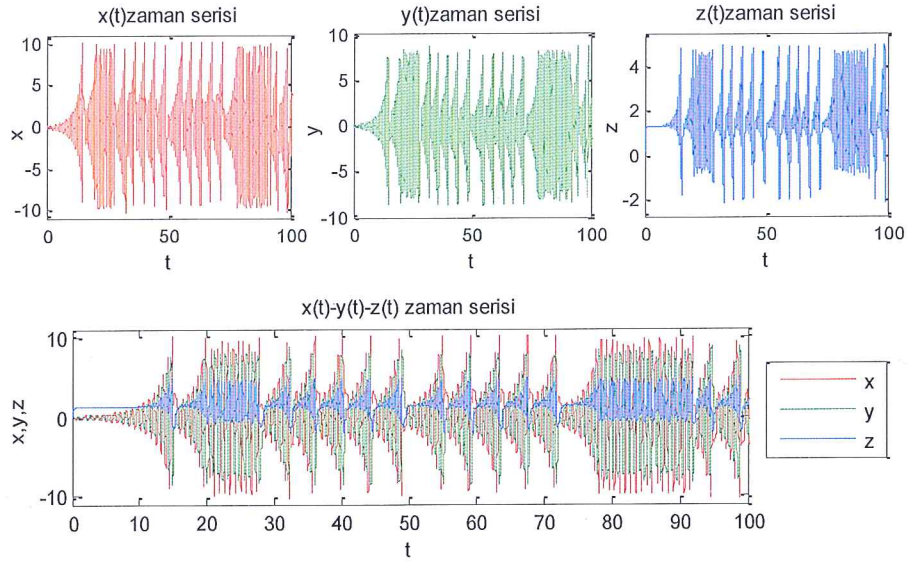
$$\begin{aligned}\dot{x} &= a \cdot y(1+z) - b \cdot x \\ \dot{y} &= -a \cdot x \cdot z + b \cdot y \cdot z - c(x-1) \\ \dot{z} &= -a \cdot z + b \cdot y \cdot x + d\end{aligned}\tag{3.16}$$

$$a = 4,40, \quad b = 1,50, \quad c = -0,20, \quad d = 5,70\tag{3.17}$$

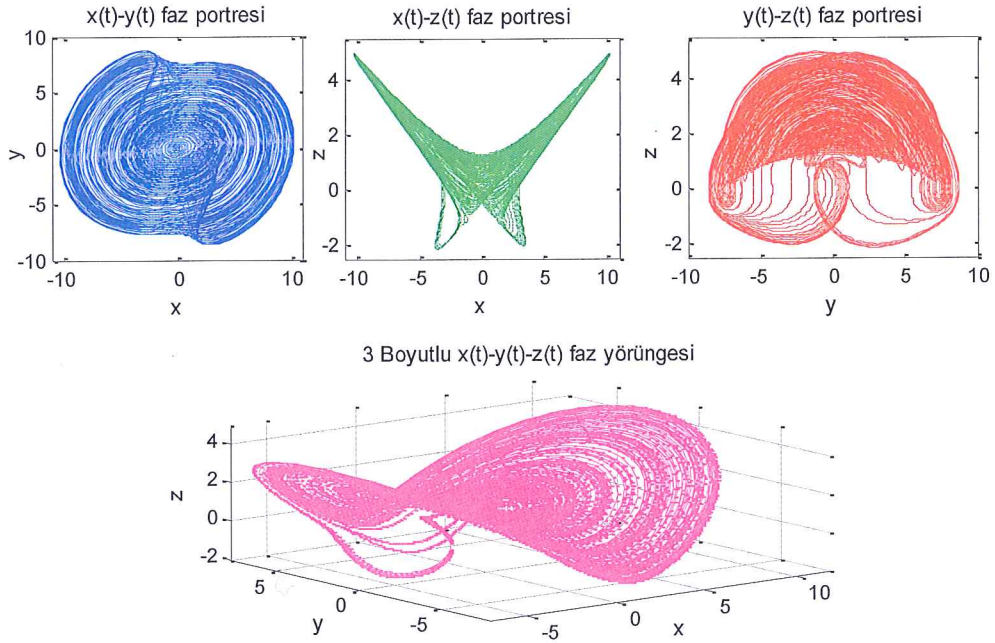
$$[x_0, y_0, z_0] = [0, 0, 0]\tag{3.18}$$

Pehlivan ve arkadaşlarının önerdiği bu sistem, 4 adet ikinci dereceden doğrusal olmayan ifade (yz, xz ve yx) olmak üzere 10 terim ve 4 parametre içermektedir. Bu sistem Matlab programında Runge-Kutta 4 nümerik algoritması kullanılarak

matematiksel olarak modellenmiştir. Nümerik modelleme sonucunda Şekil 3.19'da tasarlanan sistemin zamanla değişen x , y ve z çıkışlarından alınan sonuçlar grafiksel olarak verilmiştir. Şekil 3.20'de ise tasarlanan kaotik sistemin x - y , x - z , y - z faz portleri ve üç boyutlu x - y - z yörüngesi görülmektedir.

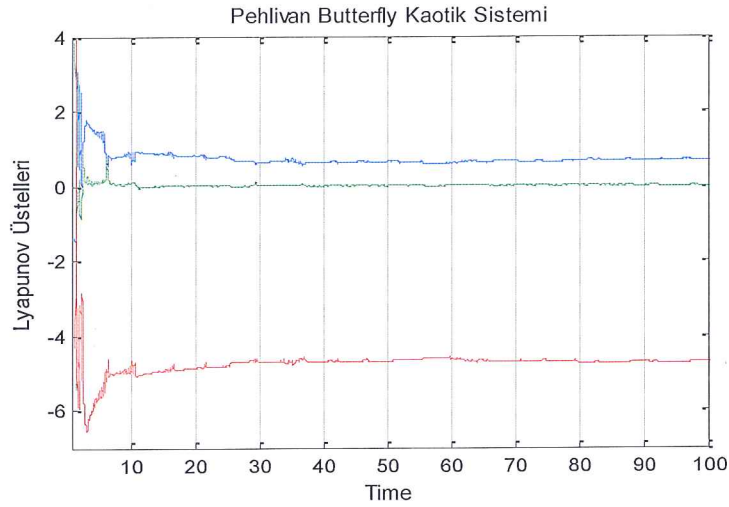


Şekil 3.19. Pehlivan yeni kaotik sisteminin x , y , z kaotik durum değişkenlerinin zamana göre değişimi.



Şekil 3.20. Pehlivan vd. tarafından tasarlanan kelebek şeklinde 3 boyutlu yeni kaotik sisteminin x - y , x - z , y - z kaotik çekicileri ve üç boyutlu x - y - z yörüngesi.

Bu kaotik sistemin kaos analizi için gerekli Lyapunov üstelleri Matlab programında geliştirilen Lyapunov Exponent Toolbox (LET) kullanılarak elde edilmiştir. Yapılan Lyapunov üstelleri çalışmasından alınan sonuçlar Şekil 3.21’de verilmiştir. Sonuçlardan da görüleceği üzere Lyapunov üstellerinin işaretlerinin birisi pozitif, diğeri sıfır ve sonuncusu negatiftir. Sonuçlara göre sistemin işaretleri ($\lambda_1=0,64150$, $\lambda_2=0$, $\lambda_3=-4,6390$) sırasıyla (+, 0, -) olduğundan bu sistemin kaotik olduğu kanıtlanmıştır.



Şekil 3.21. Pehlivan vd. tarafından tasarlanan kaotik sisteminin Lyapunov üstelleri.

3.2.7. Yeni Wei Hiperkaotik Sistemi (2013)

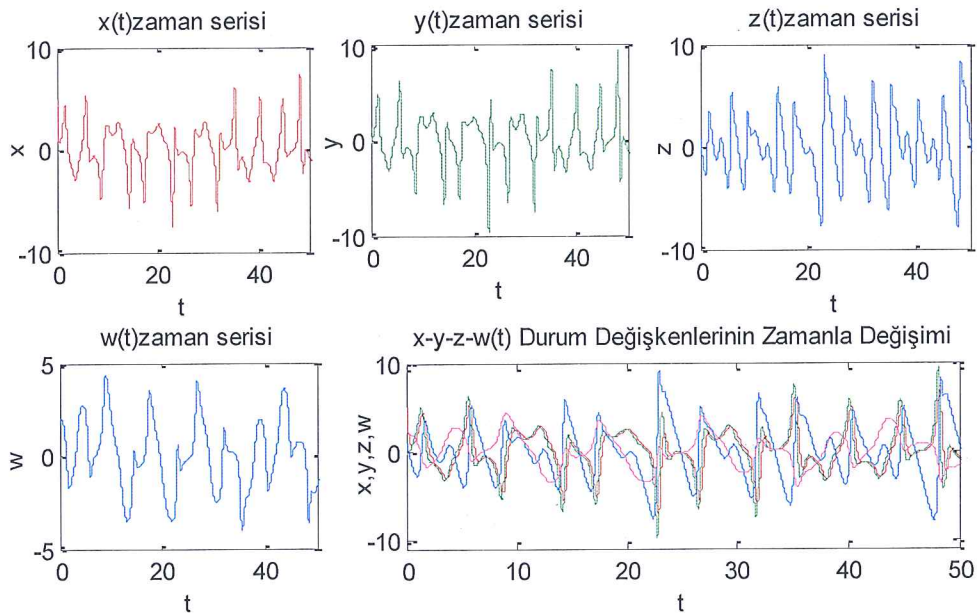
Wei vd. tarafından gerçekleştirilen çalışmada denge noktasız yeni bir hiperkaotik sistem tasarlamışlardır [107]. Yeni hiperkaotik sistemi, genelleştirilmiş yayınımsız Lorenz kaotik sistem denklemleri üzerinde gerçekleştirdikleri çalışma sonucunda elde etmişlerdir. Bu çalışmada kaotik sistemin değişik parametre değerleri için faz portrelerini incelemişlerdir. Ayrıca hem analog devre üzerinde hem de bilgisayar tabanlı elektronik devre analiz programı OrCAD-PSpice üzerinde devreyi tasarlayarak nümerik analiz sonuçları ile karşılaştırmışlardır. Kaos uygulamaları için önerilen yeni hiperkaotik sistemin matematiksel modeli Eşitlik 3.19’da verilmiştir. Matematiksel modelde ifade edilen kaotik sistemin sistem parametreleri (a , b , c , k , m) Eşitlik 3.20’de ve Eşitlik 3.21’de ise kaotik sistemin başlangıç şartları (x_0 , y_0 , z_0) ifade edilmektedir.

$$\begin{aligned}
\dot{x}_1 &= a \cdot (y_1 - x_1) \\
\dot{y}_1 &= -x_1 \cdot z_1 - c \cdot y_1 + k \cdot w_1 \\
\dot{z}_1 &= -b + x_1 \cdot y_1 \\
\dot{w}_1 &= -m \cdot y_1
\end{aligned} \tag{3.19}$$

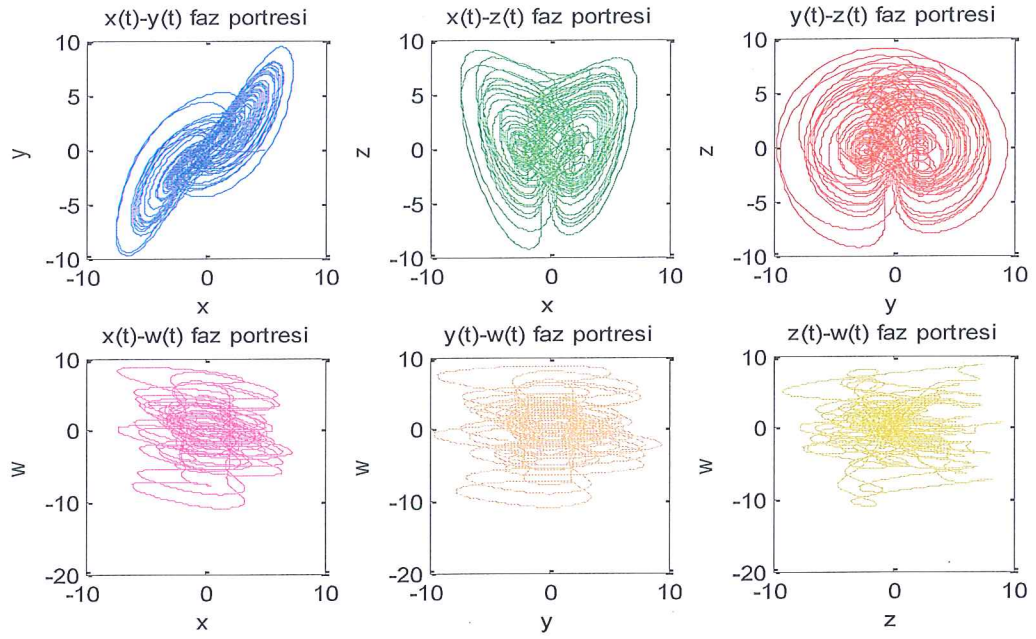
$$a = 5, b = 5, c = 0, k = 1, m = 1 \tag{3.20}$$

$$[x_0, y_0, z_0, w_0] = [5, -0.1, 10, 2] \tag{3.21}$$

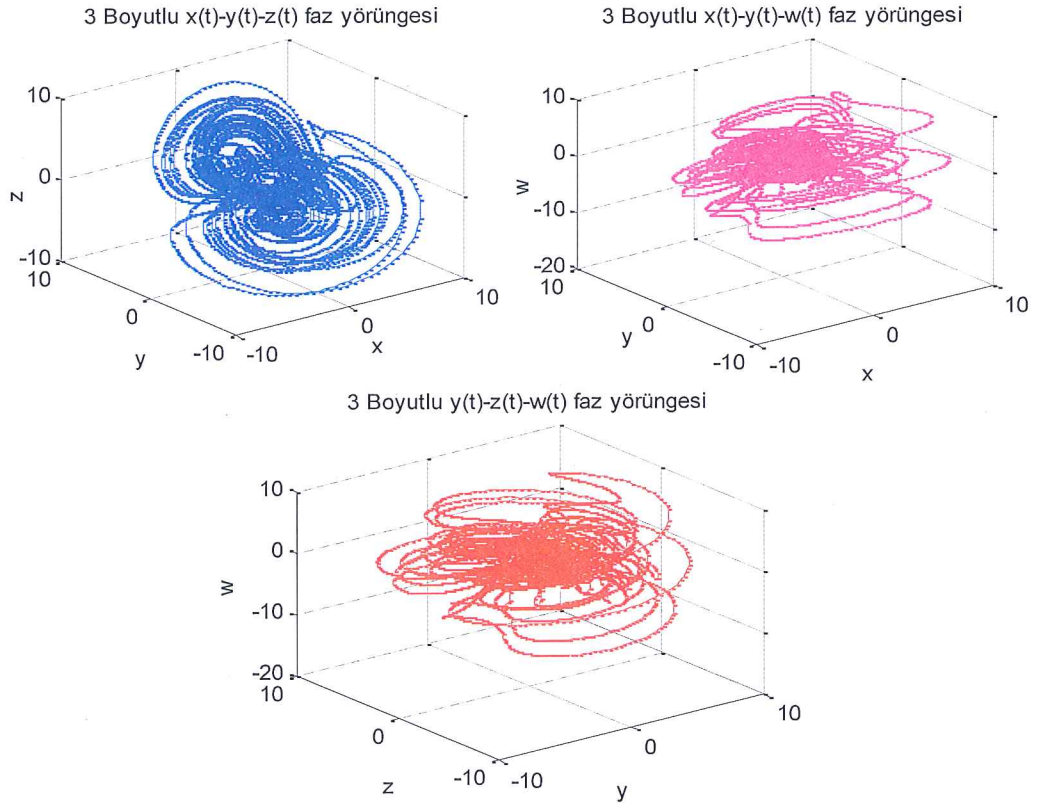
Wei vd. tarafından önerilen bu sistem, 4 adet durum değişkeni (x, y, z, w), 2 adet ikinci dereceden doğrusal olmayan ifade (xz ve xy) olmak üzere 8 terim ve 5 parametre içermektedir. Bu sistem Matlab programında Runge-Kutta 4 nümerik algoritması kullanılarak matematiksel olarak modellenmiştir. Şekil 3.22’de tasarlanan sistemin zamanla değişen x, y, z ve w çıkışlarından alınan sonuçlar grafiksel olarak verilmiştir. Şekil 3.23’te tasarlanan kaotik sistemin x - y , x - z , y - z , x - w , y - w , z - w faz portleri ve Şekil 3.24’te üç boyutlu x - y - z , x - z - w , y - z - w yörüngeleri görülmektedir.



Şekil 3.22. Wei kaotik sisteminin x, y, z kaotik durum değişkenlerinin zamana göre değişimi.

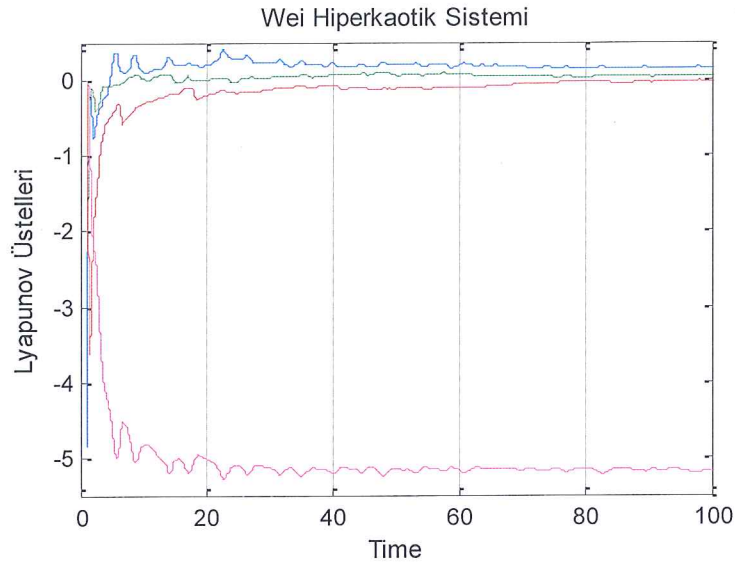


Şekil 3.23. Wei yeni kaotik sisteminin x - y , x - z , y - z , x - w , y - w ve z - w 2 boyutlu kaotik faz portreleri.



Şekil 3.24. Wei yeni kaotik sisteminin 3 boyutlu x - y - z , x - y - w ve y - z - w faz yörüngeleri.

Wei vd. tarafından tasarlanan yeni hiperkaotik sistemin kaos analizi için gerekli Lyapunov üstelleri Matlab programında geliştirilen Lyapunov Exponent Toolbox (LET) kullanılarak elde edilmiştir. Yapılan Lyapunov üstelleri çalışmasından alınan sonuçlar Şekil 3.25'te verilmiştir. Sonuçlardan da görüleceği üzere Lyapunov üstellerinin işaretlerinin ikisi pozitif, biri sıfır ve sonuncusu negatiftir. Sonuçlara göre sistemin işaretleri ($\lambda_1=0,18610$, $\lambda_2=0,0490$, $\lambda_3=0$, $\lambda_4=-5,1760$) sırasıyla (+, +, 0, -) olduğundan bu sistemin hiperkaotik olduğu kanıtlanmıştır.



Şekil 3.25. Wei yeni hiper-kaotik sisteminin Lyapunov üstelleri.

3.2.8. Yeni Pehlivan 3 Boyutlu 4 Sarmallı Kaotik Sistemi (2011)

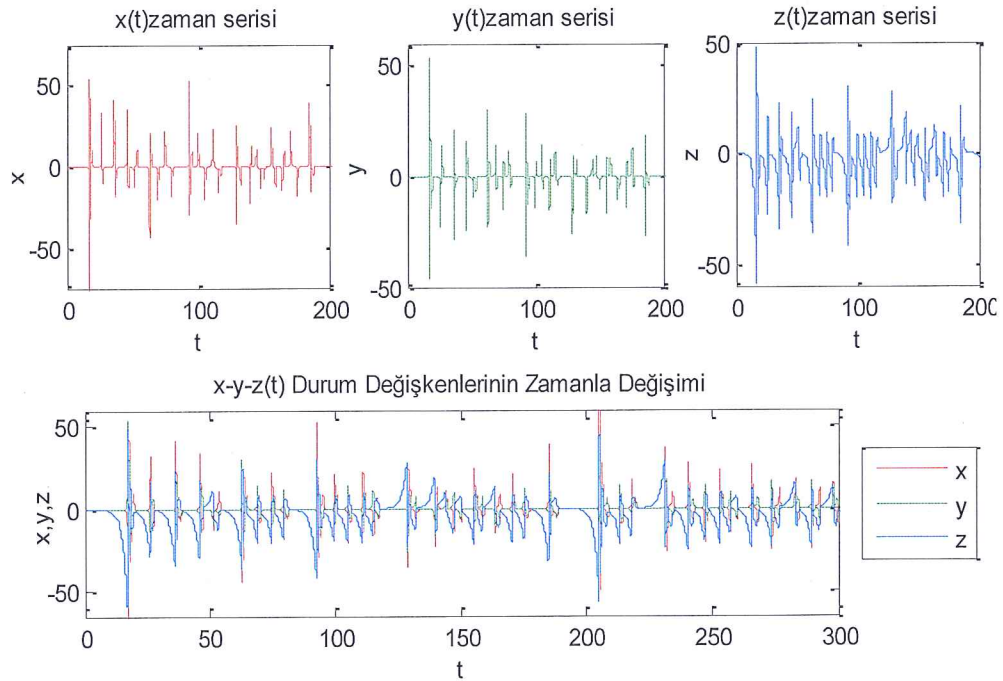
Pehlivan yaptığı çalışmada literatüre dört sarmallı yıldız şeklinde 3 boyutlu otonom yeni kaotik sistemini tanıtmıştır [108]. Yaptığı bu çalışmada yeni kaotik sistemin dinamik denklemlerini çıkartarak denge noktaları, Lyapunov üstelleri ve 2-3 boyut faz portreleri gibi sistemin bazı temel dinamik özellikleri üzerinde çalışmıştır. Ayrıca tasarladığı yeni kaotik sistemin elektronik devre gerçeklemesini yaparak nümerik analiz sonuçları ile analog devre sonuçlarını karşılaştırmıştır. Önerilen sistemin diferansiyel matematiksel modeli Eşitlik 3.22'de verilmiştir. Eşitlik 3.23'te sistemin parametre değerleri (a, b, c) ve Eşitlik 3.24'te ise başlangıç koşulları (x_0 , y_0 , z_0) verilmiştir.

$$\begin{aligned}
\dot{x} &= -a \cdot x + y + y \cdot z \\
\dot{y} &= x - a \cdot y + b \cdot x \cdot z \\
\dot{z} &= c \cdot z - b \cdot x \cdot y
\end{aligned}
\tag{3.22}$$

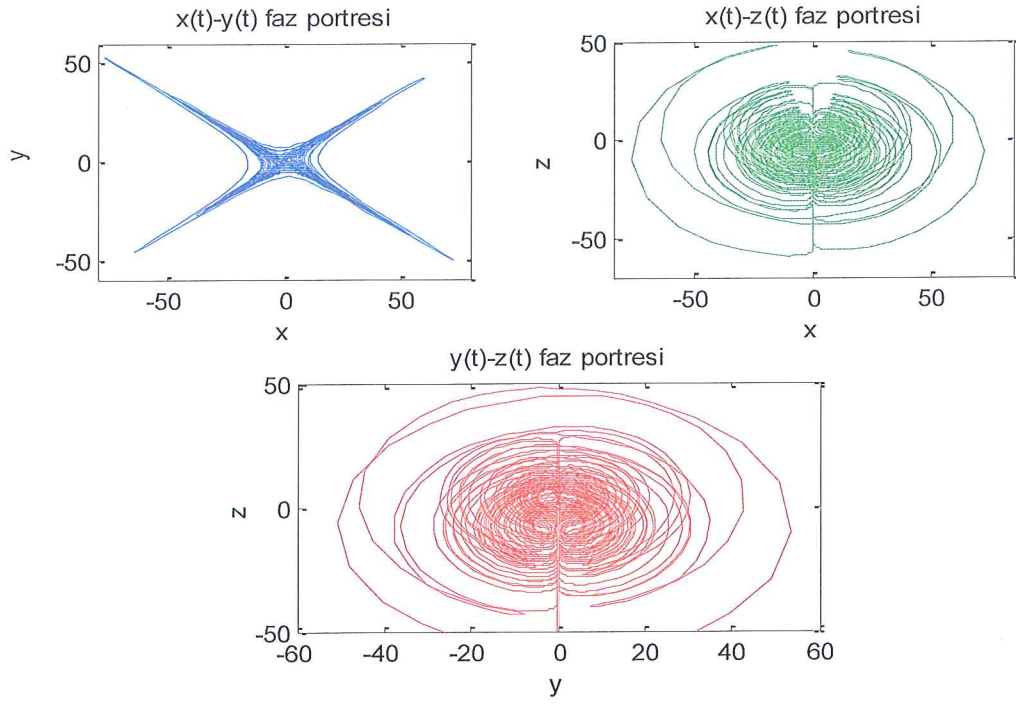
$$a = 4, \quad b = 0,50, \quad c = 0,60 \tag{3.23}$$

$$[x_0, y_0, z_0] = [0,60, 0, 0] \tag{3.24}$$

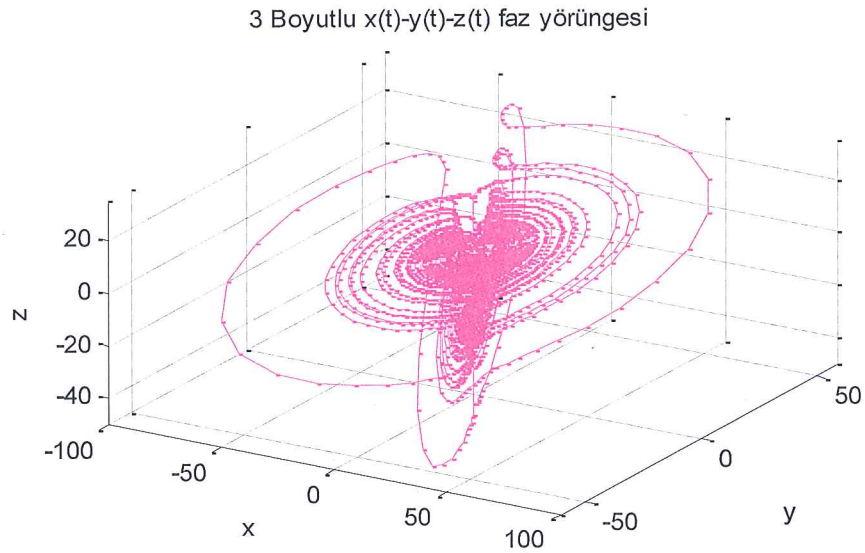
Pehlivan tarafından tasarlanan yeni kaotik sistem, 3 adet ikinci dereceden doğrusal olmayan ifade (yz, xz ve xy) olmak üzere 8 terim ve 3 parametre içermektedir. Bu sistem Matlab programında nümerik algoritmalar kullanılarak matematiksel olarak modellenmiştir. Şekil 3.26'da tasarlanan sistemin zamanla değişen durum değişkenleri x, y ve z çıkışlarından alınan sonuçlar grafiksel olarak verilmiştir. Şekil 3.27'de tasarlanan kaotik sistemin x-y, x-z, y-z faz portleri ve üç boyutlu x-y-z yörüngesi görülmektedir.



Şekil 3.26. Pehlivan 4 sarmallı yeni kaotik sisteminin x, y, z kaotik durum değişkenlerinin zamana göre değişimi.



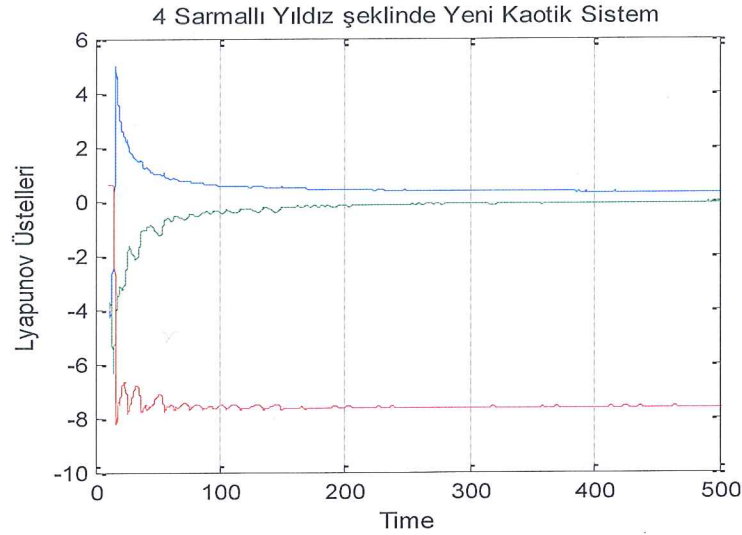
Şekil 3.27. Pehlivan 4 sarmallı yıldız şeklinde yeni kaotik sisteminin 2 boyutlu x-y, x-z, y-z faz portreleri.



Şekil 3.28. Pehlivan 4 sarmallı yeni kaotik sisteminin 3 boyutlu faz portresi.

Pehlivan tarafından tasarlanan yıldız şeklinde 4 sarmallı yeni kaotik sistemin kaos analizi için gerekli Lyapunov üstelleri Matlab programında geliştirilen Lyapunov Exponent Toolbox (LET) kullanılarak elde edilmiştir. Yapılan Lyapunov üstelleri çalışmasından alınan sonuçlar Şekil 3.29’da verilmiştir. Sonuçlardan da görüleceği

üzere Lyapunov üstellerinin işaretlerinin birisi pozitif, diğeri sıfır ve sonuncusu negatiftir. Sonuçlara göre sistemin işaretleri ($\lambda_1=0,26780$, $\lambda_2=0$, $\lambda_3=-7,47560$) sırasıyla (+, 0, -) olduğundan bu sistemin kaotik olduğu kanıtlanmıştır.



Şekil 3.29. Pehlivan 4 sarmallı yeni kaotik sistemin $x_0=0,60$, $y_0=0$, $z_0=0$ başlangıç koşulları ve $a=4$, $b=0,50$, $c=0,60$ parametre değerlerinde Lyapunov üstelleri.

Çizelge 3.2’de bu tez kapsamında araştırılan literatüre yeni sunulan kaotik sistemlerin dinamik denklemlerinin teknik özellikleri karşılaştırılmıştır.

Çizelge 3.2. Kaotik sistemlerinin dinamik özelliklerinin karşılaştırılması.

	PCKS Tasarımı 2007	P3BKS Tasarımı 2012	Aboee Kaotik Sistemi 2013	Deng Kaotik Sistemi 2014	Zhoua Kaotik Sistemi 2008	Kelebek Kaotik Sistemi. 2014	Wei Hiperkaotik Sistemi 2013	4Sarmallı Kaotik Sistem 2011
Doğrusal Olmayan İfade	5	2	2	2	2	4	2	3
Terim	10	8	7	7	7	10	8	8
Parametre	2	2	6	3	4	4	5	3
Durum Değişkeni	3	3	3	3	3	3	4	3

BÖLÜM 4

REFERANS KAOTİK SİSTEMLERİN NUMERİK MODELLERİ İLE FPGA ÜZERİNDE KAOTİK OSİLATÖR TASARIMI

FPGA yüksek hız ve kapasiteleri nedeniyle özellikle yüksek performans ve işlemci gücü gerektiren kriptoloji ve güvenli haberleşme gibi uygulamalarda bilgi güvenliği kapasitesini iyileştirmede önemli bir potansiyele sahiptir. Bu bölümde, üçüncü bölümde bahsedilen ve üzerinde FPGA ile çalışma yapılmamış sürekli zamanlı otonom PCKS ve P3BKS tasarımları seçilerek farklı numerik diferansiyel denklem çözüm algoritmalarıyla modellenmiştir. Daha sonra seçilen kaotik sistemler FPGA üzerinde Euler, Heun, RK4 ve RK5-Butcher algoritmaları ile tasarım ve gerçekleştirme çalışmaları yapılmıştır [109-112]. Ayrıca tezde önerilen çift entropi çekirdekli yapı oluşturulabilmek için Ring osilatörün FPGA üzerinde tasarımı da bu bölümde gerçekleştirilmiştir. Yapılan tüm çalışmalar sonucunda kullanılan algoritmaların hassasiyet analizi yapılarak FPGA tabanlı yeni kaotik sistem yapılarının performans ve çip istatistikleri incelenmiştir. Matlab'tan alınan nümerik sonuçlarla FPGA üzerinden alınan donanımsal sonuçlar karşılaştırılarak hata analizleri gerçekleştirilmiştir. Son kısımda ise literatürde FPGA tabanlı kaotik osilatör çalışmaları hakkında bilgi verilmiştir.

4.1. REFERANS KAOTİK SİSTEMLERİN NÜMERİK MODELLERİ

4.1.1. PCKS Tasarımı

PCKS'nin matematiksel ifadesi Eşitlik 4.1'de verilmiştir. Yeni tasarlanan bu kaotik sistemde sistem parametreleri ρ ve σ Eşitlik 4.2'de ve kaotik sistemin başlangıç şartları (x_0, y_0, z_0) Eşitlik 4.3'te verilmiştir.

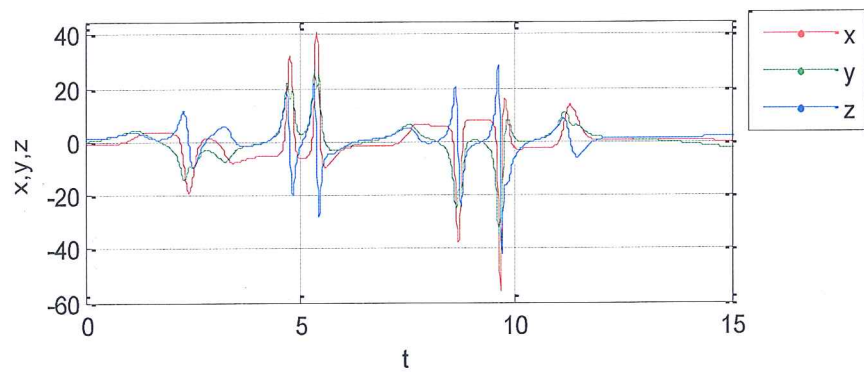
$$\begin{aligned}
\frac{dx}{dt} &= y \cdot (z - \rho) \\
\frac{dy}{dt} &= y \cdot (z - \rho) - x \cdot (z + \rho) \\
\frac{dz}{dt} &= -y \cdot (\rho \cdot x - y) - \sigma \cdot (z - \rho)
\end{aligned} \tag{4.1}$$

$$\rho = 1,30, \sigma = 4 \tag{4.2}$$

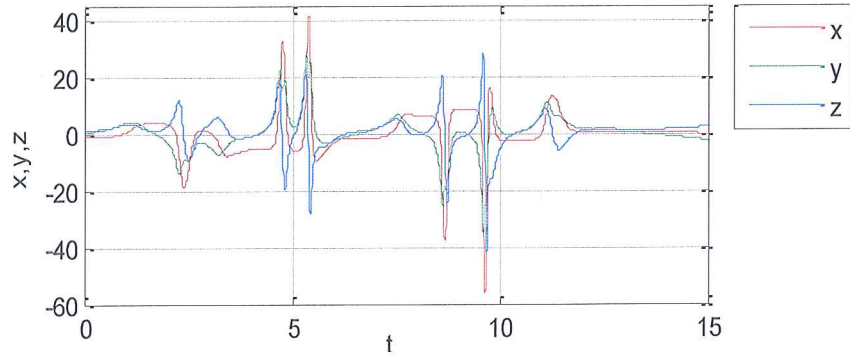
$$x_0 = -1, y_0 = 0, z_0 = 1 \tag{4.3}$$

4.1.1.1. PCKS Tasarımının Nümerik Yöntemlerle Modellenmesi

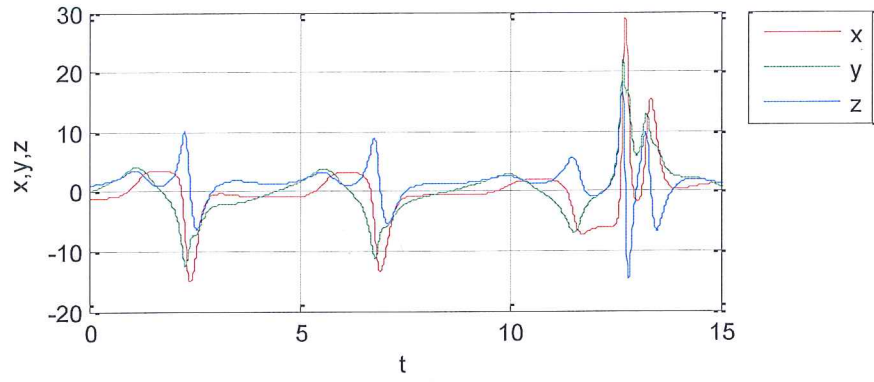
PCKS Matlab programında Euler, Heun, RK4 ve RK5-Butcher nümerik diferansiyel denklem çözüm yöntemleri ile modellenmiştir. Matlab üzerinden alınan sonuçlar, FPGA üzerinde nümerik algoritmalarla tasarlanan PCKS den alınan sonuçların doğrulanması amacıyla ile kullanılmıştır. Kaotik sistemin sırasıyla Euler, Heun, RK4 ve RK5-Butcher nümerik analiz sonucunda elde edilen x, y, z kaotik durum değişkenlerinin zaman göre değişimi Şekil 4.1, Şekil 4.2, Şekil 4.3 ve Şekil 4.4'te grafiksel olarak verilmiştir.



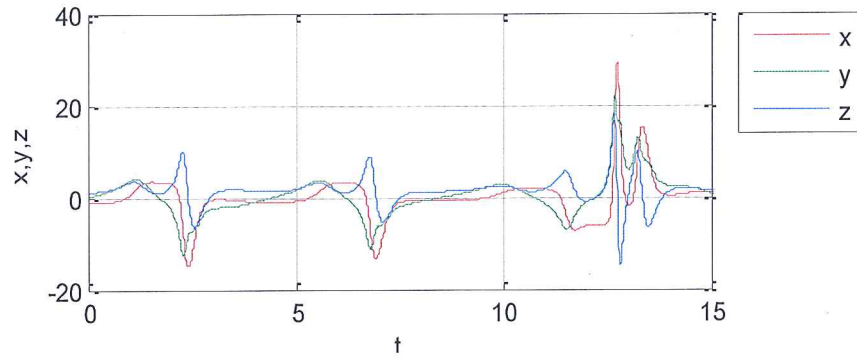
Şekil 4.1. PCKS'nin Euler nümerik modeli x, y, z zaman serileri.



Şekil 4.2. PCKS'nin Heun nümerik modeli x , y , z zaman serileri.

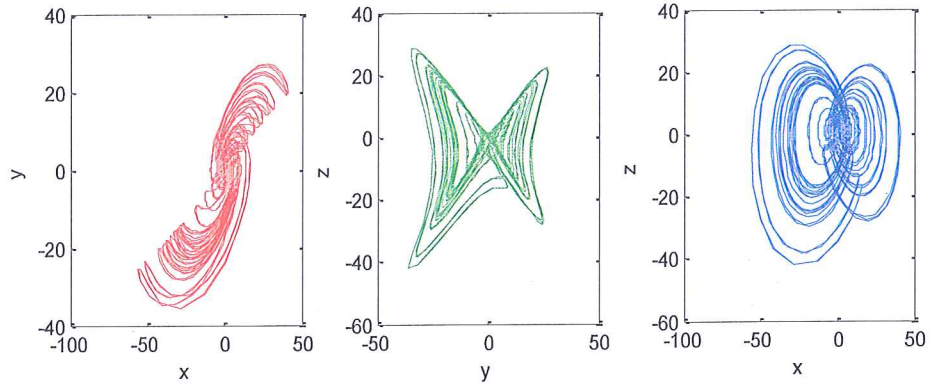


Şekil 4.3. PCKS'nin RK4 nümerik modeli x , y , z zaman serileri.

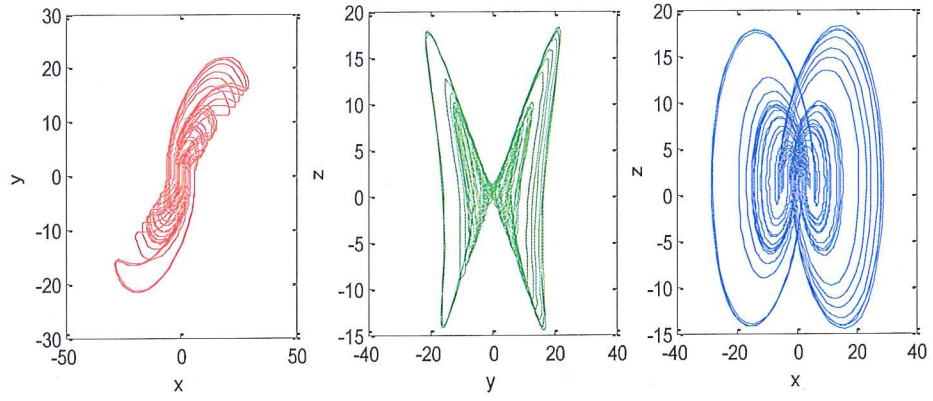


Şekil 4.4. PCKS'nin RK5-Butcher nümerik modeli x , y , z zaman serileri.

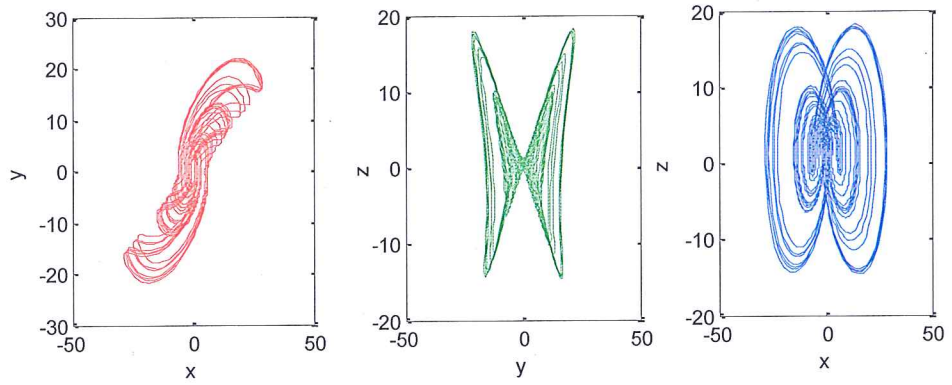
Aynı parametre ve başlangıç şartlarındaki x - y , x - z ve y - z kaotik faz portreleri ile üç boyutlu x - y - z yörüngesi sırasıyla Şekil 4.5, Şekil 4.6, Şekil 4.7 ve Şekil 4.8'de görülmektedir.



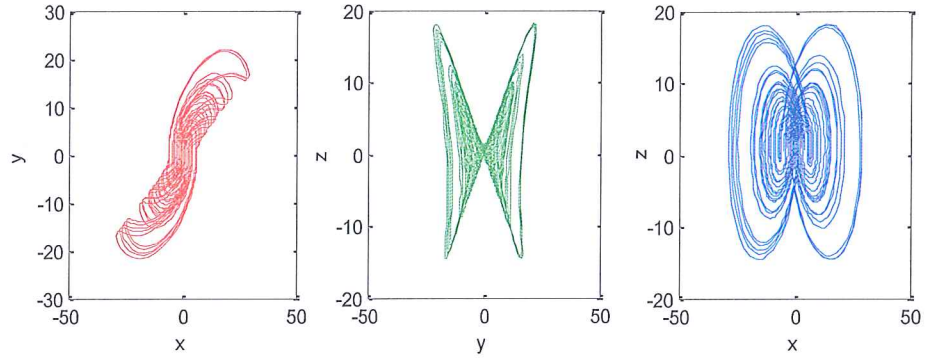
Şekil 4.5. PCKS'nin Euler nümerik modeli x-y, x-z, y-z faz portreleri.



Şekil 4.6. PCKS'nin Heun nümerik modeli x-y, x-z, y-z faz portreleri.



Şekil 4.7. PCKS'nin RK4 nümerik modeli x-y, x-z, y-z faz portreleri.



Şekil 4.8. PCKS'nin RK5-Butcher nümerik modeli x-y, x-z, y-z faz portreleri.

4.1.2. P3BKS Tasarımı

P3BKS sürekli zamanlı iki kuadratik doğrusalsızlık içeren otonom tasarımının matematiksel modeli Eşitlik 4.4'te verilmiştir. Matematiksel modelde verilen a ve b sistem parametrelerinin değerleri Eşitlik 4.5'te ve kaotik sistemin başlangıç şartları (x_0, y_0, z_0) da Eşitlik 4.6'da tanımlanmıştır. Önerilen bu sistem, 2 adet ikinci dereceden doğrusal olmayan ifade, 8 terim ve 2 parametre içermektedir.

$$\begin{aligned}\dot{x} &= y - x - az \\ \dot{y} &= xz - x \\ \dot{z} &= -xy - y + b\end{aligned}\tag{4.4}$$

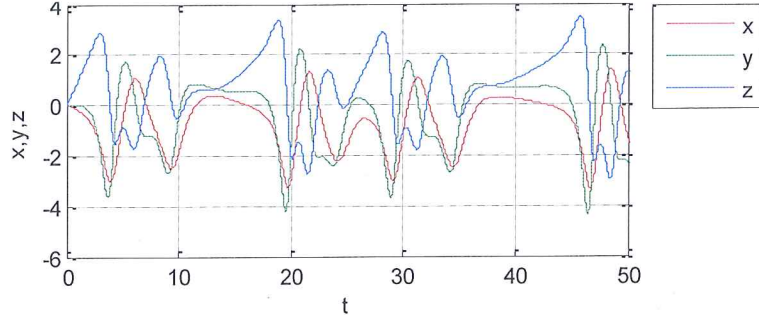
$$a = 0,50, \quad b = 1\tag{4.5}$$

$$x_0 = 0, \quad y_0 = 0, \quad z_0 = 0\tag{4.6}$$

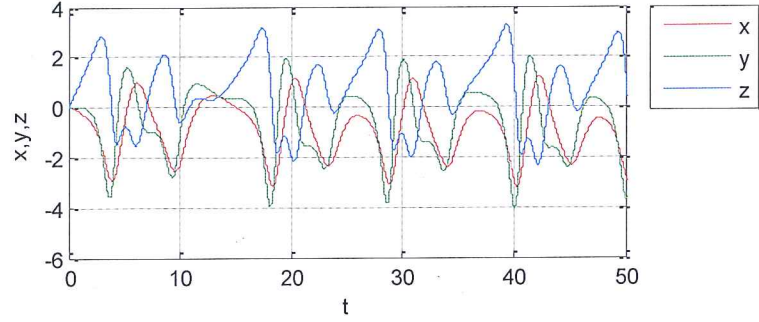
4.1.2.1. P3BKS Tasarımının Nümerik Yöntemlerle Modellenmesi

P3BKS Matlab programında Euler, Heun, RK4 ve RK5-Butcher nümerik diferansiyel denklem çözüm yöntemleri ile modellenmiştir. Matlab üzerinden alınan sonuçlar, FPGA üzerinde nümerik algoritmalarla tasarlanan P3BKS' den alınan sonuçların doğrulanması amacıyla ile kullanılmıştır. Yeni kaotik sistemin sırasıyla Euler, Heun, RK4 ve RK5-Butcher nümerik analiz sonucunda elde edilen x, y, z

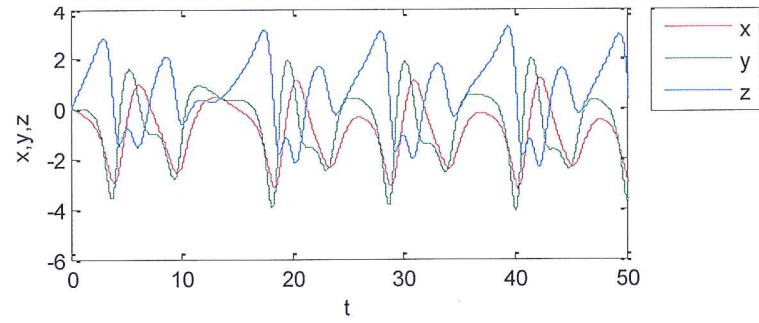
kaotik durum deęiřkenlerinin zaman gre deęiřimi řekil 4.9, řekil 4.10, řekil 4.11, ve řekil 4.12’de grafiksel olarak verilmiřtir.



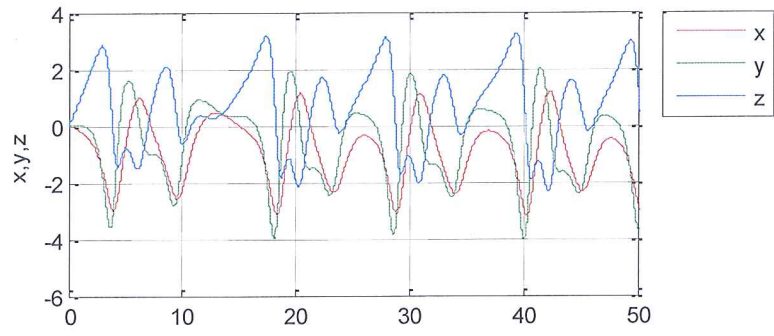
řekil 4.9. P3BKS tasarımınnın Euler modeli x, y, z zaman serileri.



řekil 4.10. P3BKS tasarımınnın Heun modeli x, y, z zaman serileri.

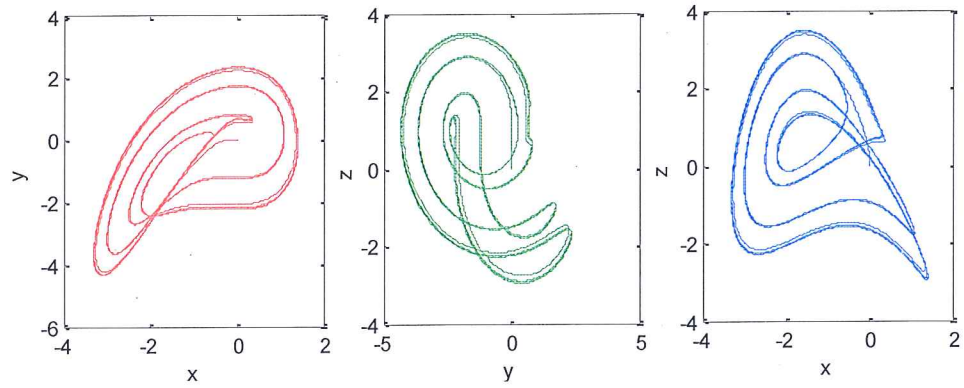


řekil 4.11. P3BKS tasarımınnın RK4 modeli x, y, z zaman serileri.

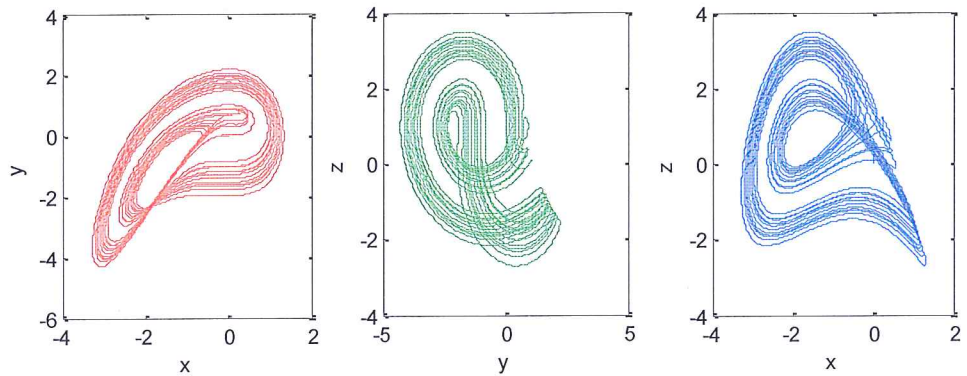


Şekil 4.12. P3BKS tasarımının RK5-Butcher modeli x , y , z zaman serileri.

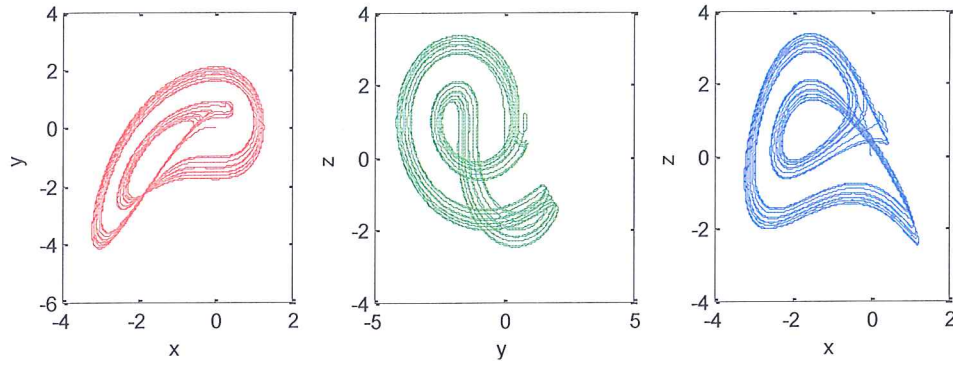
Aynı parametre ve başlangıç şartlarındaki x - y , x - z ve y - z kaotik faz portreleri ile üç boyutlu x - y - z yörüngesi sırasıyla Şekil 4.14, Şekil 4.14, Şekil 4.15 ve Şekil 4.16'da görülmektedir.



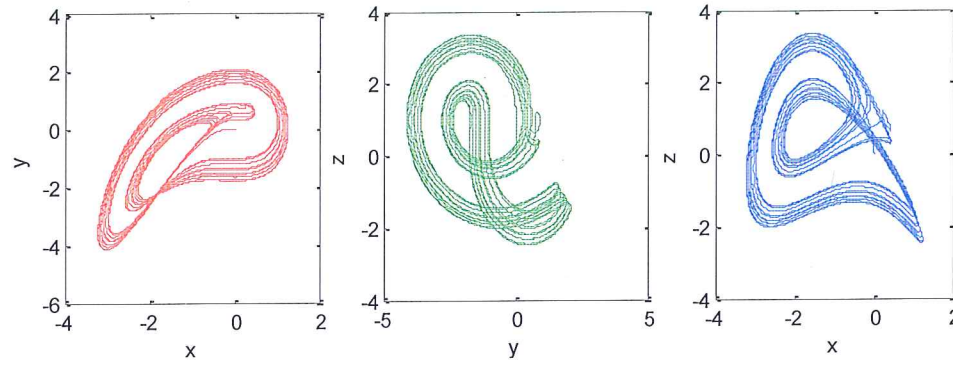
Şekil 4.13. P3BKS tasarımının Euler nümerik modeli x - y , x - z , y - z faz portreleri.



Şekil 4.14. P3BKS tasarımının Heun nümerik modeli x - y , x - z , y - z faz portreleri.



Şekil 4.15. P3BKS tasarımının RK4 nümerik modeli x-y, x-z, y-z faz portreleri.



Şekil 4.16. P3BKS tasarımının RK5-Butcher nümerik modeli x-y, x-z, y-z faz portreleri.

4.2. REFERANS KAOTİK SİSTEMLERİN AYRIKLAŞTIRILMIŞ MODELLERİ

Bu kısımda üzerinde FPGA tabanlı çalışma yapılmayan PCKS ve P3BKS osilatörlerin Euler, Heun, RK4 ve RK5-Butcher nümerik algoritmaları ile 32 bit IQ-Math sabit sayı standardında VHDL dili ile modellenmiştir. FPGA üzerinde yapılan gerçekelemelerden alınan sonuçların performans ve çip istatistikleri incelenerek kullanılan algoritmaların hassasiyet analizleri yapılmıştır. Her iki kaotik sistem için kullanılacak nümerik algoritmaların ayırıklaştırılmış modelleri çıkartılarak tüm işlemler gerçekleştirilmiştir.

Algoritmaların ayırıklaştırılmış modellerindeki $x(k)$, $y(k)$ ve $z(k)$ 'nin başlangıç değerleri PCKS için $x(k)=-1.0$, $y(k)=0.0$ ve $z(k)=1.0$ ve P3BKS için $x(k)=0.0$, $y(k)=0.0$ ve $z(k)=0.0$ olarak alınmıştır. Çalışmada sadece bir kaotik sistemin

ayrıklaştırılmış algoritmaları verilecektir. PCKS'nin kaotik sisteminin Euler algoritması kullanılarak ayrıklaştırılmış matematiksel modeli Eşitlik 4.7'de sunulmaktadır.

$$\begin{aligned}x(k+1) &= x(k) + \Delta h(y(k).(z(k) - \rho)) \\y(k+1) &= y(k) + \Delta h(y(k).(z(k) - \rho) - x(k).(z(k) + \rho)) \\z(k+1) &= z(k) + \Delta h(-y(k).(\rho.x(k) - y(k)) - \sigma.(z(k) - \rho))\end{aligned}\tag{4.7}$$

Heun tabanlı PCKS'nin ayrıklaştırılmış matematiksel modeli Eşitlik 4.8'de verilmektedir. Matematiksel modelden de görüldüğü gibi Heun algoritması iki aşamalı bir algoritmadır. Birinci aşamada $x(k)$ değeri kullanılarak $x(k_0+1)$ değeri hesaplanmaktadır. İkinci aşamada ise $x(k_0+1)$ değeri ve $x(k)$ değerleri kullanılarak sistemin Δh kadar adım sonraki değeri olan $x(k+1)$ değeri hesaplanmaktadır.

$$\begin{aligned}x(k^0+1) &= x(k) + \Delta h(y(k).(z(k) - \rho)) \\x(k+1) &= x(k) + \Delta h((y(k).(z(k) - \rho)) + x(k^0+1))/2 \\y(k^0+1) &= y(k) + \Delta h(y(k).(z(k) - \rho) - x(k).(z(k) + \rho)) \\y(k+1) &= y(k) + \Delta h((y(k).(z(k) - \rho) - x(k).(z(k) + \rho)) + y(k^0+1))/2 \\z(k^0+1) &= z(k) + \Delta h(-y(k).(\rho.x(k) - y(k)) - \sigma.(z(k) - \rho)) \\z(k+1) &= z(k) + \Delta h((-y(k).(\rho.x(k) - y(k)) - \sigma.(z(k) - \rho)) + z(k^0+1))/2\end{aligned}\tag{4.8}$$

RK4 algoritması kullanılarak PCKS ayrıklaştırılmış matematiksel modeli Eşitlik 4.9'da ki f , g ve δ fonksiyonlarına göre Eşitlik 4.10 ve Eşitlik 4.11'de verilmektedir. Bu denklemlerde bulunan κ_1 , κ_2 , κ_3 ve κ_4 parametreleri, kaotik sistemin birinci denkleme ait katsayıları, λ_1 , λ_2 , λ_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.

$$\begin{aligned}
\dot{x} &= f(t, x, y, z) = y.(z - \rho) \\
\dot{y} &= g(t, x, y, z) = y.(z - \rho) - x.(z + \rho) \\
\dot{z} &= \delta(t, x, y, z) = -y.(\rho.x - y) - \sigma.(z - \rho)
\end{aligned} \tag{4.9}$$

$$\begin{aligned}
x(k+1) &= x(k) + \frac{1}{6} \Delta h [\kappa_1(k) + 2\kappa_2(k) + 2\kappa_3(k) + \kappa_4(k)] \\
y(k+1) &= y(k) + \frac{1}{6} \Delta h [\lambda_1(k) + 2\lambda_2(k) + 2\lambda_3(k) + \lambda_4(k)] \\
z(k+1) &= z(k) + \frac{1}{6} \Delta h [\xi_1(k) + 2\xi_2(k) + 2\xi_3(k) + \xi_4(k)]
\end{aligned} \tag{4.10}$$

$$\begin{aligned}
\kappa_1 &= f(x(k), y(k), z(k)) \\
\lambda_1 &= g(x(k), y(k), z(k)) \\
\xi_1 &= \delta(x(k), y(k), z(k)) \\
\kappa_2 &= f(x(k) + \frac{1}{2} \Delta h \kappa_1, y(k) + \frac{1}{2} \Delta h \lambda_1, z(k) + \frac{1}{2} \Delta h \xi_1) \\
\lambda_2 &= g(x(k) + \frac{1}{2} \Delta h \kappa_1, y(k) + \frac{1}{2} \Delta h \lambda_1, z(k) + \frac{1}{2} \Delta h \xi_1) \\
\xi_2 &= \delta(x(k) + \frac{1}{2} \Delta h \kappa_1, y(k) + \frac{1}{2} \Delta h \lambda_1, z(k) + \frac{1}{2} \Delta h \xi_1) \\
\kappa_3 &= f(x(k) + \frac{1}{2} \Delta h \kappa_2, y(k) + \frac{1}{2} \Delta h \lambda_2, z(k) + \frac{1}{2} \Delta h \xi_2) \\
\lambda_3 &= g(x(k) + \frac{1}{2} \Delta h \kappa_2, y(k) + \frac{1}{2} \Delta h \lambda_2, z(k) + \frac{1}{2} \Delta h \xi_2) \\
\xi_3 &= \delta(x(k) + \frac{1}{2} \Delta h \kappa_2, y(k) + \frac{1}{2} \Delta h \lambda_2, z(k) + \frac{1}{2} \Delta h \xi_2) \\
\kappa_4 &= f(x(k) + \Delta h \kappa_3, y(k) + \Delta h \lambda_3, z(k) + \Delta h \xi_3) \\
\lambda_4 &= g(x(k) + \Delta h \kappa_3, y(k) + \Delta h \lambda_3, z(k) + \Delta h \xi_3) \\
\xi_4 &= \delta(x(k) + \Delta h \kappa_3, y(k) + \Delta h \lambda_3, z(k) + \Delta h \xi_3)
\end{aligned} \tag{4.11}$$

Aşağıda RK5-Butcher algoritması kullanılarak PCKS'nin ayrıklaştırılmış matematiksel modeli Eşitlik 4.12'de verilmiştir [111]. Buradaki $\kappa_1 \dots \kappa_6$, $\lambda_1 \dots \lambda_6$ ve $\xi_1 \dots \xi_6$ değişkenlerinin açılımı Eşitlik 4.13'te verilmektedir. RK5-Butcher algoritması RK4 algoritmasıyla benzer yapıya sahip olmakla birlikte bu algoritma da beşinci ve altıncı dereceden terimler bulunduğundan RK4 algoritmasına göre daha hassas çözümler üretmektedir.

$$\begin{aligned}
x(k+1) &= x(k) + \frac{1}{90} \Delta h [7\kappa_1(k) + 32\kappa_3(k) + 12\kappa_4(k) + 32\kappa_5(k) + 7\kappa_6(k)] \\
y(k+1) &= y(k) + \frac{1}{90} \Delta h [7\lambda_1(k) + 32\lambda_3(k) + 12\lambda_4(k) + 32\lambda_5(k) + 7\lambda_6(k)] \\
z(k+1) &= z(k) + \frac{1}{90} \Delta h [7\xi_1(k) + 32\xi_3(k) + 12\xi_4(k) + 32\xi_5(k) + 7\xi_6(k)]
\end{aligned} \tag{4.12}$$

$$\begin{aligned}
\kappa_1 &= f(x(k), y(k), z(k)) \\
\lambda_1 &= g(x(k), y(k), z(k)) \\
\xi_1 &= \delta(x(k), y(k), z(k)) \\
\kappa_2 &= f(x(k) + \frac{1}{4} \Delta h \kappa_1, y(k) + \frac{1}{4} \Delta h \lambda_1, z(k) + \frac{1}{4} \Delta h \xi_1) \\
\lambda_2 &= g(x(k) + \frac{1}{4} \Delta h \kappa_1, y(k) + \frac{1}{4} \Delta h \lambda_1, z(k) + \frac{1}{4} \Delta h \xi_1) \\
\xi_2 &= \delta(x(k) + \frac{1}{4} \Delta h \kappa_1, y(k) + \frac{1}{4} \Delta h \lambda_1, z(k) + \frac{1}{4} \Delta h \xi_1) \\
\kappa_3 &= f(x(k) + \frac{1}{8} (\Delta h (\kappa_1 + \kappa_2), y(k) + \frac{1}{8} (\Delta h (\lambda_1 + \lambda_2), z(k) + \frac{1}{8} (\Delta h (\xi_1 + \xi_2))) \\
\lambda_3 &= g(x(k) + \frac{1}{8} (\Delta h (\kappa_1 + \kappa_2), y(k) + \frac{1}{8} (\Delta h (\lambda_1 + \lambda_2), z(k) + \frac{1}{8} (\Delta h (\xi_1 + \xi_2))) \\
\xi_3 &= \delta(x(k) + \frac{1}{8} (\Delta h (\kappa_1 + \kappa_2), y(k) + \frac{1}{8} (\Delta h (\lambda_1 + \lambda_2), z(k) + \frac{1}{8} (\Delta h (\xi_1 + \xi_2))) \\
\kappa_4 &= f(x(k) - \frac{1}{2} \Delta h \kappa_2 + \Delta h \kappa_3, y(k) - \frac{1}{2} \Delta h \lambda_2 + \Delta h \lambda_3, z(k) - \frac{1}{2} \Delta h \xi_2 + \Delta h \xi_3) \\
\lambda_4 &= g(x(k) - \frac{1}{2} \Delta h \kappa_2 + \Delta h \kappa_3, y(k) - \frac{1}{2} \Delta h \lambda_2 + \Delta h \lambda_3, z(k) - \frac{1}{2} \Delta h \xi_2 + \Delta h \xi_3) \\
\xi_4 &= \delta(x(k) - \frac{1}{2} \Delta h \kappa_2 + \Delta h \kappa_3, y(k) - \frac{1}{2} \Delta h \lambda_2 + \Delta h \lambda_3, z(k) - \frac{1}{2} \Delta h \xi_2 + \Delta h \xi_3) \\
\kappa_5 &= f(x(k) + \frac{3}{16} \Delta h \kappa_1 + \frac{9}{16} \Delta h \kappa_4, y(k) + \frac{3}{16} \Delta h \lambda_1 + \frac{9}{16} \Delta h \lambda_4, z(k) + \frac{3}{16} \Delta h \xi_1 + \frac{9}{16} \Delta h \xi_4) \\
\lambda_5 &= g(x(k) + \frac{3}{16} \Delta h \kappa_1 + \frac{9}{16} \Delta h \kappa_4, y(k) + \frac{3}{16} \Delta h \lambda_1 + \frac{9}{16} \Delta h \lambda_4, z(k) + \frac{3}{16} \Delta h \xi_1 + \frac{9}{16} \Delta h \xi_4) \\
\xi_5 &= \delta(x(k) + \frac{3}{16} \Delta h \kappa_1 + \frac{9}{16} \Delta h \kappa_4, y(k) + \frac{3}{16} \Delta h \lambda_1 + \frac{9}{16} \Delta h \lambda_4, z(k) + \frac{3}{16} \Delta h \xi_1 + \frac{9}{16} \Delta h \xi_4) \\
\kappa_6 &= f(x(k) - \frac{3}{7} \Delta h \kappa_1 + \frac{2}{7} \Delta h \kappa_2 + \frac{12}{7} \Delta h \kappa_3 - \frac{12}{7} \Delta h \kappa_4 + \frac{8}{7} \Delta h \kappa_5, y(k) - \frac{3}{7} \Delta h \lambda_1 + \frac{2}{7} \Delta h \lambda_2 + \\
&\quad \frac{12}{7} \Delta h \lambda_3 - \frac{12}{7} \Delta h \lambda_4 + \frac{8}{7} \Delta h \lambda_5, z(k) - \frac{3}{7} \Delta h \xi_1 + \frac{2}{7} \Delta h \xi_2 + \frac{12}{7} \Delta h \xi_3 - \frac{12}{7} \Delta h \xi_4 + \frac{8}{7} \Delta h \xi_5) \\
\lambda_6 &= g(x(k) - \frac{3}{7} \Delta h \kappa_1 + \frac{2}{7} \Delta h \kappa_2 + \frac{12}{7} \Delta h \kappa_3 - \frac{12}{7} \Delta h \kappa_4 + \frac{8}{7} \Delta h \kappa_5, y(k) - \frac{3}{7} \Delta h \lambda_1 + \frac{2}{7} \Delta h \lambda_2 + \\
&\quad \frac{12}{7} \Delta h \lambda_3 - \frac{12}{7} \Delta h \lambda_4 + \frac{8}{7} \Delta h \lambda_5, z(k) - \frac{3}{7} \Delta h \xi_1 + \frac{2}{7} \Delta h \xi_2 + \frac{12}{7} \Delta h \xi_3 - \frac{12}{7} \Delta h \xi_4 + \frac{8}{7} \Delta h \xi_5) \\
\xi_6 &= \delta(x(k) - \frac{3}{7} \Delta h \kappa_1 + \frac{2}{7} \Delta h \kappa_2 + \frac{12}{7} \Delta h \kappa_3 - \frac{12}{7} \Delta h \kappa_4 + \frac{8}{7} \Delta h \kappa_5, y(k) - \frac{3}{7} \Delta h \lambda_1 + \frac{2}{7} \Delta h \lambda_2 + \\
&\quad \frac{12}{7} \Delta h \lambda_3 - \frac{12}{7} \Delta h \lambda_4 + \frac{8}{7} \Delta h \lambda_5, z(k) - \frac{3}{7} \Delta h \xi_1 + \frac{2}{7} \Delta h \xi_2 + \frac{12}{7} \Delta h \xi_3 - \frac{12}{7} \Delta h \xi_4 + \frac{8}{7} \Delta h \xi_5)
\end{aligned} \tag{4.13}$$

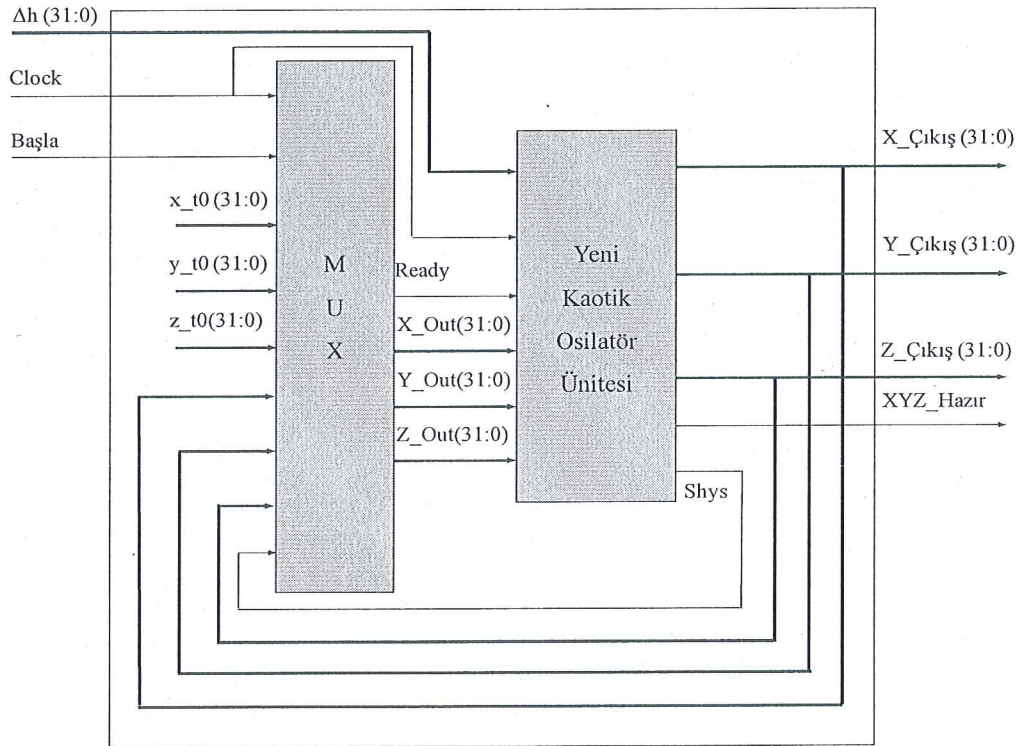
4.3. REFERANS KAOTİK SİSTEMLERİN FPGA ÜZERİNDE GERÇEKLEŞTİRİLMESİ

Bu çalışmada referans alınan PCKS ve P3BKS tasarımları Euler, Heun, RK4 ve RK5-Butcher algoritmaları kullanılarak, 32 bit IQ-Math sabit noktalı sayı standardı ile FPGA tabanlı olarak modellenmiş ve bir donanım tanımlama dili VHDL'de 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 diyagramları her iki yeni kaotik sistem için aynı olup Şekil 4.17'de görülmektedir. Ünitelerin girişlerinde bulunan 1-bitlik *Start* 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* ve *Z_Çıkış*) ve bu çıkış sinyallerinin hazır olduğunu göstermek amacıyla kullanılan 1-bit *XYZ_Hazır* sinyali bulunmaktadır.



Şekil 4.17. FPGA-tabanlı yeni kaotik osilatör ünitesi en üst seviye blok diyagramı [63].

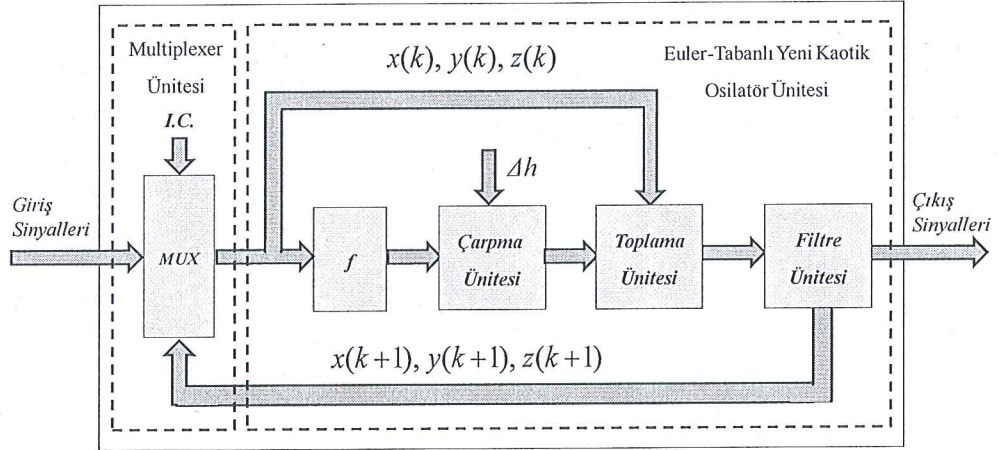
Şekil 4.18’de yeni kaotik osilatör ünitesinin ikinci seviye blok diyagramı görülmektedir. Sistemin ikinci seviye blok diyagramı bir *Multiplexer Ünitesi* ve bir tane de *FPGA-Tabanlı Yeni Kaotik Osilatör Ünitesi*’nden oluşmaktadır. Tasarımda *Multiplexer Ünitesi* kullanılmasının amacı, başlangıç koşulu 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 4.18. FPGA-tabanlı yeni kaotik osilatör ünitesi ikinci seviye blok diyagramı [63].

Şekil 4.19’da *Euler-Tabanlı Yeni Kaotik Osilatör Ünitesi* üçüncü seviye blok diyagramı verilmektedir. Osilatör yapısında *MUX*, *Çarpma*, *Toplama*, *Filtre* ve *f* üniteleri olmak üzere 5 ünite kullanılmıştır. *MUX* ünitesinin işlevi *Başla* sinyali

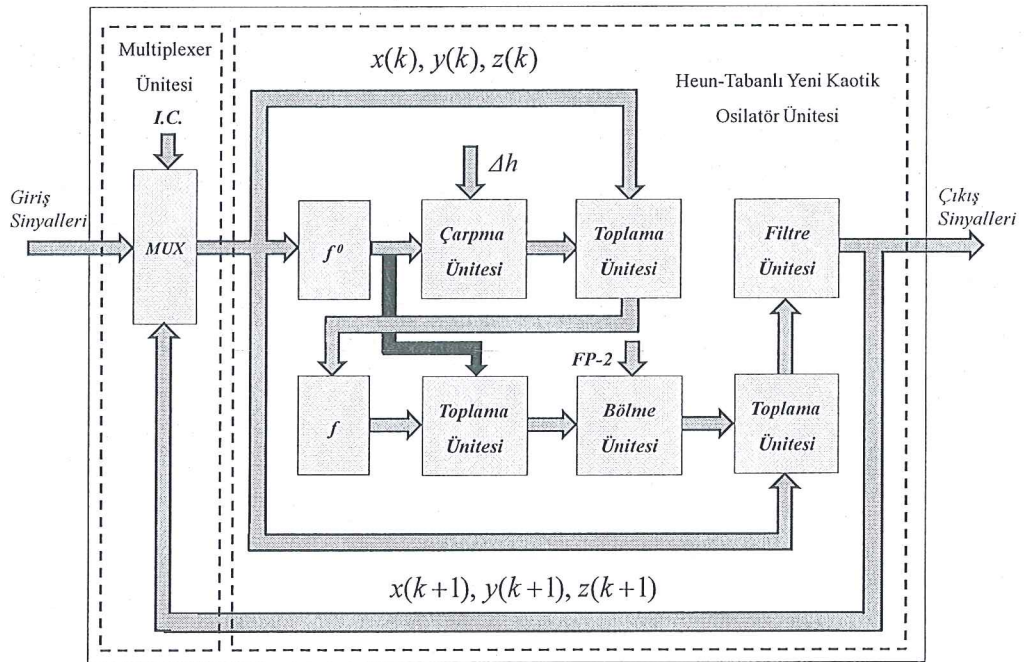
geldiğinde sistemin ihtiyaç duyduğu başlangıç şartlarının atanmasını sağlamaktır. Diğer bir ifade ile sisteme kullanıcı tarafından atanan başlangıç şartları ile sistemin çıkışından elde edilen ve bir sonraki algoritmanın hesaplanmasında başlangıç şartları olarak kullanılan $x(k+1)$, $y(k+1)$ ve $z(k+1)$ sinyalleri arasında seçim yaparak bu sinyalleri sisteme göndermektir. MUX ünitesinin girişlerinden birisi olan *I.C.* kaotik sistemin başlangıç şartlarını ifade etmektedir. Sistemdeki f ünitesi MUX ünitesinden gelen kontrol sinyalleri ile PCKS ve P3BKS denklemlerinin hesaplanmasını sağlamaktadır. Bu ünitelerden çıkan sinyaller ile algoritma adım sayısı olan Δh değeri *Çarpma Ünitesi*'si tarafından çarpılmaktadır. Ardından *Toplama Ünitesi* başlangıç şartı değerlerini toplayarak sonuçları *Filtre Ünitesi*'sine göndermektedir. Sistemde *Filtre Ünitesi*, kaotik osilatörün istenmeyen sinyaller üretmesini engellemek diğer bir ifade ile filtrelemek amacıyla kullanılmaktadır. Ünite pipe-line olarak çalışmakta ve Euler-Tabanlı kaotik osilatör ünitesi 32 saat darbesi sonunda ilk değerini üretmektedir. Kaotik osilatör ünitesinin ürettiği sinyaller ($x(k+1)$, $y(k+1)$ ve $z(k+1)$) sistemin hem çıkışlarını hem de bir sonraki algoritmanın hesaplanması için MUX ünitesine gönderilerek başlangıç şartlarını oluşturmaktadır.



Şekil 4.19. Euler-tabanlı yeni kaotik osilatör ünitesi üçüncü seviye blok diyagramı [63].

Şekil 4.20'de *Heun-Tabanlı Yeni Kaotik Osilatör Ünitesi* üçüncü seviye blok diyagramı görülmektedir. Osilatör yapısında MUX, f^0 , *Çarpma*, *Bölme*, *Toplama*, f ve *Filtre* üniteleri olmak üzere 9 ünite bulunmaktadır. Buradaki f^0 ünitesi MUX ünitesinden gelen kontrol sinyalleri ile PCKS ve P3BKS denklemlerindeki $x(k^0 + 1)$,

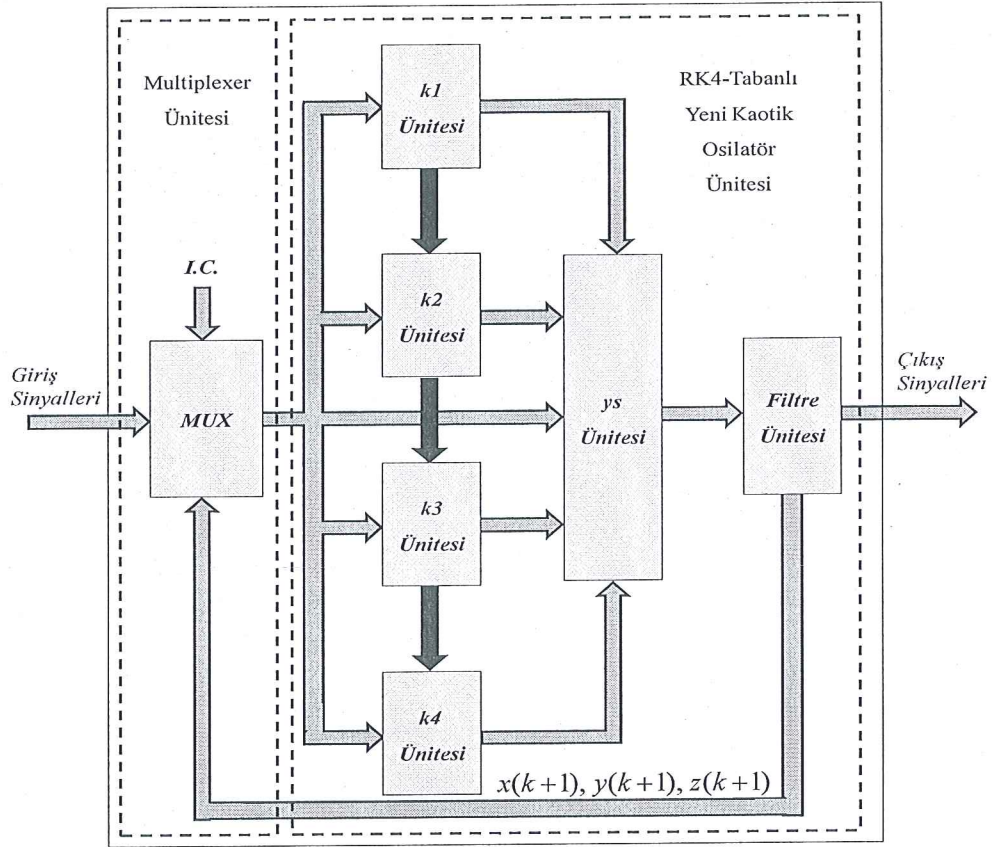
$y(k^0 + 1)$ ve $z(k^0 + 1)$ değerleri hesaplanmaktadır. Bu üniteden elde edilen sinyaller *Çarpma Ünitesi* yardımıyla Δh ile çarpılmakta ve kaotik osilatörün başlangıç şartı değerleri ile $(x(k), y(k)$ ve $z(k))$ toplanmaktadır. Bu şekilde algoritmanın ilk aşaması tamamlanmaktadır. İkinci aşamada *Toplama Ünitesi*'nden çıkan sinyaller f_0 ünitesinden elde edilen sinyaller ile toplanmakta ve sabit noktalı sayı formatında 2.0 değerine (FP-2) *Bölme Ünitesi* ile bölünmektedir. Bölme ünitesinden çıkan sinyaller ile kaotik osilatörün ürettiği bir önceki sinyaller $(x(k), y(k)$ ve $z(k))$ *Toplama Ünitesi*'nde toplanarak sonuçlar *Filtre Ünitesi*'ne gönderilmektedir. *Filtre Ünitesi*'nden çıkan $x(k+1)$, $y(k+1)$ ve $z(k+1)$ sinyalleri sistemin hem çıkışlarını hem de bir sonraki algoritmanın hesaplanması için *MUX* ünitesine gönderilerek başlangıç şartlarını oluşturmaktadır. *Heun-Tabanlı Kaotik Osilatör Ünitesi* ilk değerini 66 saat darbesi sonunda üretmektedir. *Kaotik Osilatör Ünitesi* pipe-line olarak çalıştığından her 66 saat darbesi sonunda sistemin 32 bitlik çıkış sinyalleri olan *X_Çıkış*, *Y_Çıkış* ve *Z_Çıkış* değerlerini çıkışa aktarmaktadır.



Şekil 4.20. Heun-tabanlı yeni kaotik osilatör ünitesi üçüncü seviye blok diyagramı [133].

Şekil 4.21'de *RK4-Tabanlı Yeni Kaotik Osilatör Ünitesi* üçüncü seviye blok diyagramı verilmektedir. Kaotik osilatörde *MUX*, k_1 , k_2 , k_3 , k_4 , y_s ve *Filtre Üniteleri* olmak üzere toplam 7 ünite bulunmaktadır. k_1 , k_2 , k_3 , k_4 Üniteleri kaotik sistemin

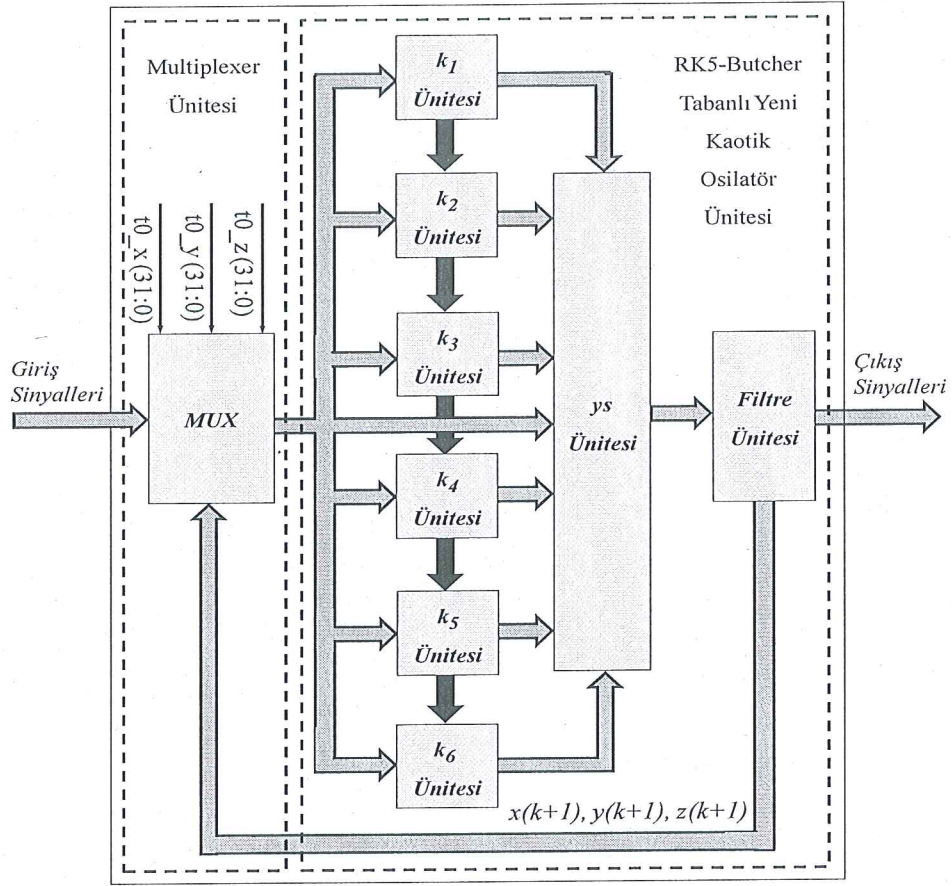
ayrıklaştırılmış modelindeki $\sigma=1...4$ 'e kadar κ_σ , ξ_σ ve λ_σ değerlerini hesaplamaktadır. RK4 algoritmasında Eşitlik 4.10'da verilen $x(k+1)$, $y(k+1)$ ve $z(k+1)$ değerleri ise ys ünitesinde hesaplanmaktadır. Ünite pipe-line olarak çalışmakta ve kaotik osilatör 136 saat darbesi sonunda ilk değerini üretmektedir. Ancak ünitenin bir sonraki $x(k+1)$, $y(k+1)$ ve $z(k+1)$ değerlerini üretebilmesi için geri besleme değerlerine ihtiyaç göstermekte ve bu değerleri 136 saat darbesi sonra üretmiş olduğu çıkış sinyallerini girişteki *MUX Ünitesine* göndererek sağlamaktadır. Kaotik osilatörün sonuç üretmediği durumlarda, çıkışa istenmeyen sinyallerin ulaşamaması için bir *Filtre Ünitesi* kullanılmıştır. Bu şekilde bütün istenmeyen sinyaller filtrelenmekte ve çıkışa sadece istenen çıkış sinyalleri gönderilmektedir.



Şekil 4.21. RK4-tabanlı yeni kaotik osilatör ünitesi üçüncü seviye blok diyagramı [63].

Şekil 4.22'de RK5-Butcher-Tabanlı Yeni Kaotik Osilatör Ünitesi üçüncü seviye blok diyagramı görülmektedir. Kaotik Osilatör Ünitesinde, k_1 , k_2 , k_3 , k_4 , k_5 ve k_6 , ys ve *Filtre Üniteleri* olmak üzere toplam 9 ünite bulunmaktadır. RK4-Tabanlı Yeni Kaotik

Osilatör Ünitesi'nden farklı olarak bu kısımda sadece beşinci ve altıncı dereceden terimlerin hesaplanması için k_5 ve k_6 *Üniteleri* eklenmiştir. Bunun dışında *Ünite*, *RK4-Tabanlı Yeni Kaotik Osilatör Ünitesi* ile benzer biçimde çalışmaktadır.



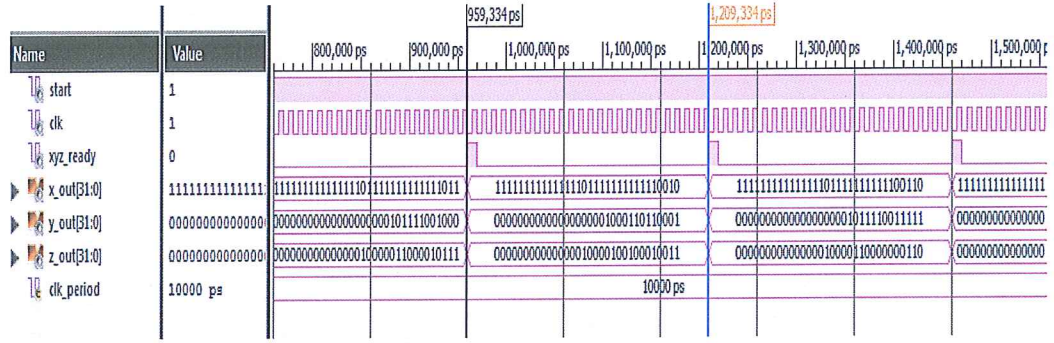
Şekil 4.22. RK5-Butcher tabanlı yeni kaotik osilatör ünitesi üçüncü seviye blok diyagramı [112].

4.4. FPGA TABANLI KAOTİK OSİLATÖRLERİN TEST SONUÇLARI

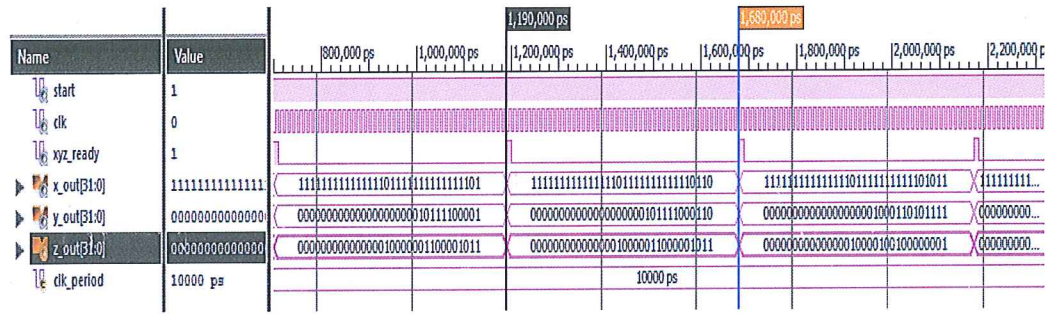
4.4.1. FPGA Tabanlı PCKS Tasarımının Gerçekleştirilmesi

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

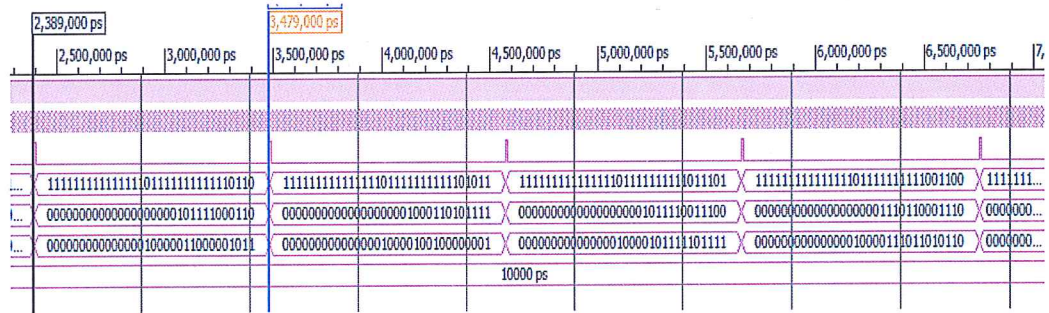
osilatör ünitesinde karşılıkları olan X_cikis , Y_cikis ve Z_cikis sinyallerinin zaman serilerine ait değerler 32 bit tam sayı formatında gösterilmiştir. Euler, Heun, RK4 ve RK5-Butcher tabanlı PCKS ünitelerinin Xilinx ISE Simülatöründen elde edilen sonuçlar Şekil 4.23-25 ve FPGA çip istatistikleri ise Şekil 4.26-28’da verilmiştir.



Şekil 4.23. Euler-tabanlı PCKS osilatör ünitesi Xilinx ISE simülatör sonuçları.



Şekil 4.24. Heun-tabanlı PCKS osilatör ünitesi Xilinx ISE simülatör sonuçları.



Şekil 4.25. RK4-tabanlı PCKS osilatör ünitesi Xilinx ISE simülatör sonuçları.

Device Utilization Summary (estimated values)				[1]
Logic Utilization	Used	Available	Utilization	
Number of Slice Registers	1471	93120	1%	
Number of Slice LUTs	1249	46560	2%	
Number of fully used LUT-FF pairs	999	1721	58%	
Number of bonded IOBs	99	240	41%	
Number of BUFG/BUFGCTRLs	1	32	3%	
Number of DSP48E1s	36	288	12%	

Şekil 4.26. Euler-tabanlı PCKS osilatör ünitesi FPGA çip kullanım istatistikleri.

Device Utilization Summary (estimated values)				[1]
Logic Utilization	Used	Available	Utilization	
Number of Slice Registers	2998	93120	3%	
Number of Slice LUTs	2687	46560	5%	
Number of fully used LUT-FF pairs	2117	3568	59%	
Number of bonded IOBs	99	360	27%	
Number of BUFG/BUFGCTRLs	1	32	3%	
Number of DSP48E1s	72	288	25%	

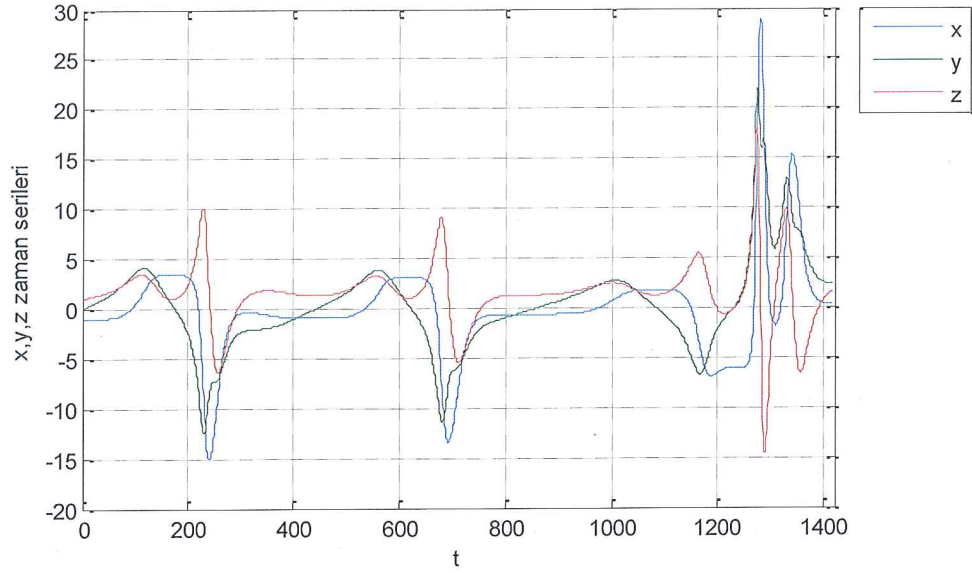
Şekil 4.27. Heun-tabanlı PCKS osilatör ünitesi FPGA çip kullanım istatistikleri.

Device Utilization Summary (estimated values)				[1]
Logic Utilization	Used	Available	Utilization	
Number of Slice Registers	6439	93120	6%	
Number of Slice LUTs	5756	46560	12%	
Number of fully used LUT-FF pairs	4548	7647	59%	
Number of bonded IOBs	99	360	27%	
Number of BUFG/BUFGCTRLs	1	32	3%	
Number of DSP48E1s	168	288	58%	

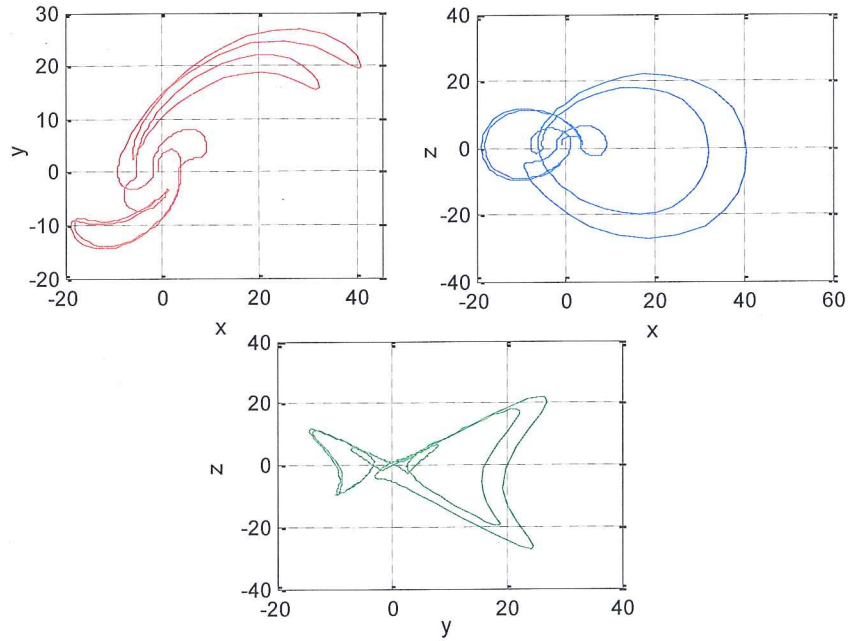
Şekil 4.28. RK4-tabanlı PCKS osilatör ünitesi FPGA çip kullanım istatistikleri.

Ayrıca kaotik osilatörlerin FPGA’de gerçekleşmesinden elde edilen X_{cikis} , Y_{cikis} ve Z_{cikis} 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 3x1450 veri seti yardımıyla X_{cikis} , Y_{cikis} ve Z_{cikis} sinyallerinin zaman serileri ve faz portreleri elde edilmiştir. Örnek olarak Şekil 4.29’de Euler tabanlı PCKS’nin $x(t)$,

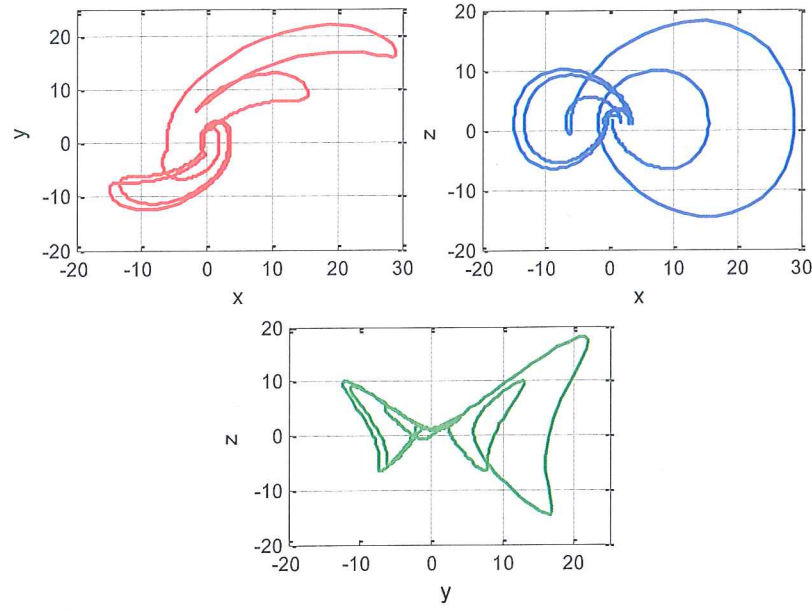
$y(t)$, ve $z(t)$ zaman serileri verilmiştir. Şekil 4.30'da Euler tabanlı ve Şekil 4.31'de RK4 tabanlı PCKS'nin FPGA üzerinde gerçekleştirilmesinden elde edilen faz portreleri verilmiştir.



Şekil 4.29. FPGA üzerinde gerçekleştirilen Euler tabanlı PCKS'nin $x(t)$, $y(t)$, ve $z(t)$ zaman serileri.



Şekil 4.30. FPGA üzerinde Euler-tabanlı PCKS osilatör ünitesi faz portreleri.



Şekil 4.31. FPGA üzerinde RK4-tabanlı PCKS osilatör ünitesi faz portreleri.

Çizelge 4.1’de Euler, Heun ve RK4 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.

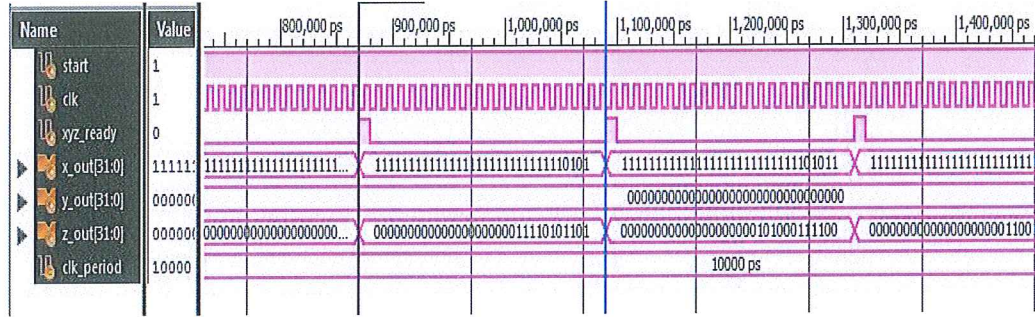
Çizelge 4.1. Yeni Kaotik osilatörlerin Xilinx Virtex-6 FPGA çip istatistikleri.

Kaotik Osilatör	Slice Regs. Sayısı / %	LUTs Sayısı / %	Bonded IOBs Sayısı / %	Maks. Saat Frekansı (MHz)	Minimum Darbe Periyodu (ns)
Kullanılan / Kullanım Oranı%					
Euler-Tabanlı PCKS	1471 / 1	1249 / 2	99 / 27	464,688	2,152
Heun-Tabanlı PCKS	2998 / 3	2687 / 5	99 / 27	464,688	2,152
RK4-Tabanlı PCKS	6439 / 6	5756 / 12	99 / 27	464,688	2,152

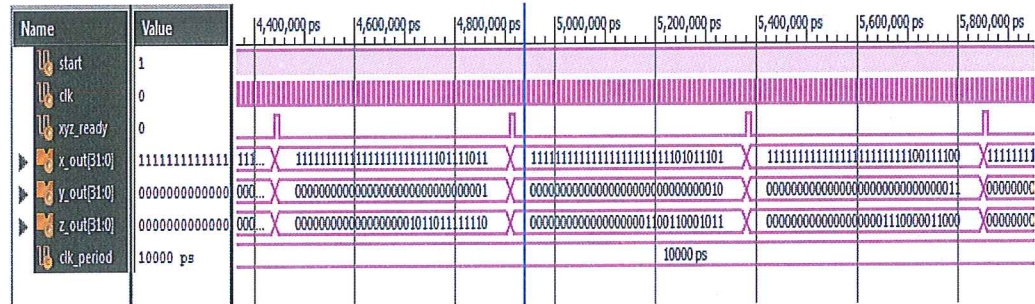
4.4.2. FPGA Tabanlı P3BKS Tasarımının Gerçekleştirilmesi

Euler, Heun, RK4 ve RK5-Butcher Tabanlı Osilatör Üniteleri, Xilinx Virtex-6 ailesi XC6VLX75T çipi için sentezlenerek, FPGA çip kaynak kullanımına ve ünitelerin

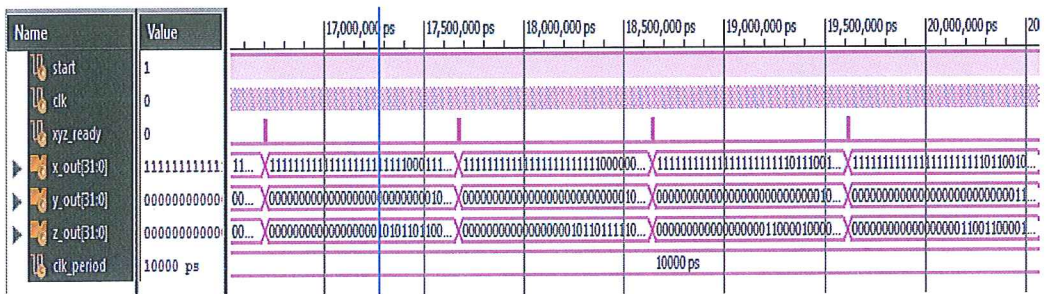
saat hızlarına ait parametrelerin istatistikleri incelenmiştir. Tasarımı yapılan 4 farklı yapıdaki ünitelerin verileri işleme süresi, Xilinx ISE Design Tools 14.2 benzetim programı kullanılarak elde edilmiştir. Burada kaotik osilatörün ISE Design Tools kullanılarak FPGA’de gerçekleştirilmesinden elde edilen x , y ve z sinyallerinin kaotik osilatör ünitesinde karşılıkları olan X_cikis , Y_cikis ve Z_cikis sinyallerinin zaman serilerine ait değerler 32 bit tam sayı formatında gösterilmiştir. Euler, Heun, RK4 ve RK5-Butcher tabanlı P3BKS osilatör ünitelerinin Xilinx ISE Simülatöründen elde edilen sonuçlar Şekil 4.32-4.35 ve FPGA çip istatistikleri ise Şekil 4.36-4.39’da verilmiştir.



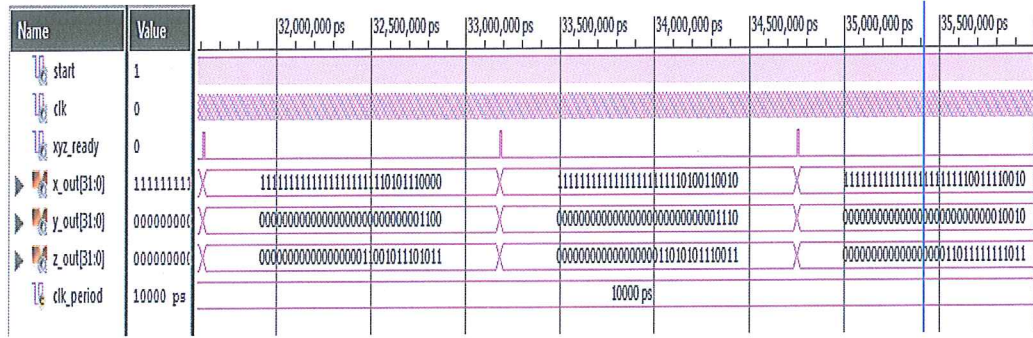
Şekil 4.32. Euler-tabanlı P3BKS osilatör ünitesi Xilinx ISE simülatör sonuçları.



Şekil 4.33. Heun-tabanlı P3BKS osilatör ünitesi Xilinx ISE simülatör sonuçları.



Şekil 4.34. RK4-tabanlı P3BKS osilatör ünitesi Xilinx ISE simülatör sonuçları.



Şekil 4.35. RK5-Butcher tabanlı P3BKS osilatör ünitesi Xilinx ISE simülator sonuçları.

Device Utilization Summary (estimated values)				[...]
Logic Utilization	Used	Available	Utilization	
Number of Slice Registers	1196	455040	0%	
Number of Slice LUTs	1070	227520	0%	
Number of fully used LUT-FF pairs	820	1446	56%	
Number of bonded IOBs	99	720	13%	
Number of BUFG/BUFGCTRLs	1	32	3%	
Number of DSP48E1s	24	576	4%	

Şekil 4.36. Euler-tabanlı P3BKS osilatör ünitesi FPGA çip kullanım istatistikleri.

Device Utilization Summary (estimated values)				[...]
Logic Utilization	Used	Available	Utilization	
Number of Slice Registers	2447	93120	2%	
Number of Slice LUTs	2328	46560	5%	
Number of fully used LUT-FF pairs	1758	3017	58%	
Number of bonded IOBs	99	240	41%	
Number of BUFG/BUFGCTRLs	1	32	3%	
Number of DSP48E1s	48	288	16%	

Şekil 4.37. Heun-tabanlı P3BKS osilatör ünitesi FPGA çip kullanım istatistikleri.

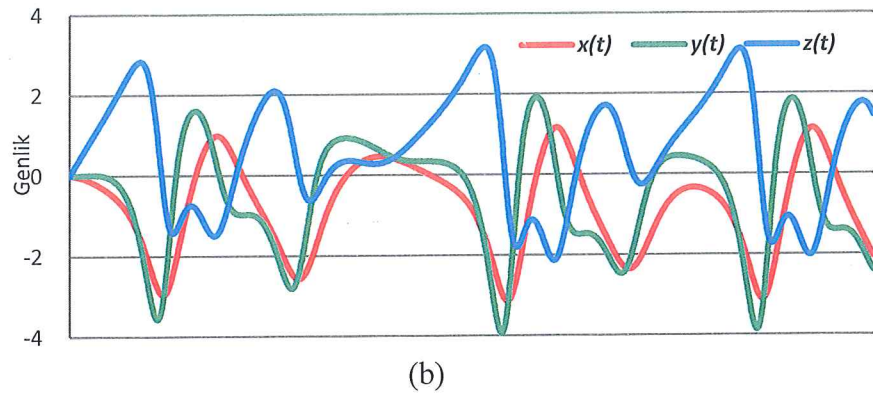
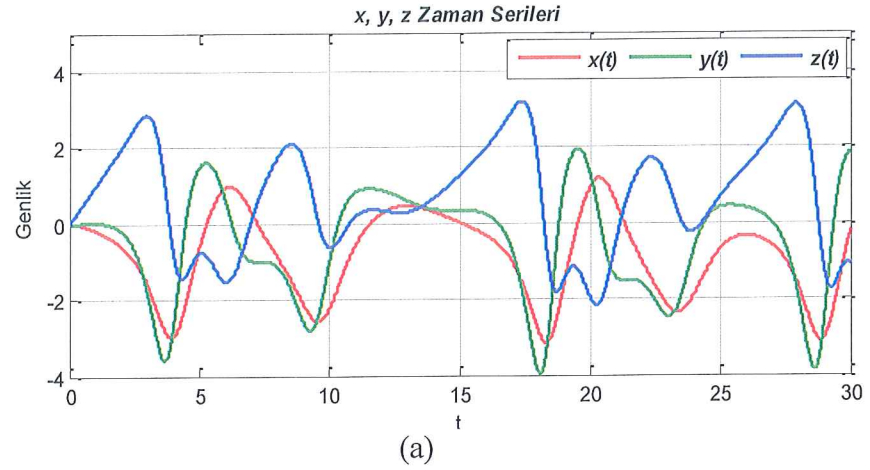
Device Utilization Summary (estimated values)			[...]
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	5338	93120	5%
Number of Slice LUTs	5039	46560	10%
Number of fully used LUT-FF pairs	3831	6546	58%
Number of bonded IOBs	99	360	27%
Number of BUFG/BUFGCTRLs	1	32	3%
Number of DSP48E1s	120	288	41%

Şekil 4.38. RK4-tabanlı P3BKS osilatör ünitesi FPGA çip kullanım istatistikleri.

Device Utilization Summary (estimated values)			[...]
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	10794	93120	11%
Number of Slice LUTs	10070	46560	21%
Number of fully used LUT-FF pairs	7695	13169	58%
Number of bonded IOBs	99	360	27%
Number of BUFG/BUFGCTRLs	1	32	3%
Number of DSP48E1s	276	288	95%

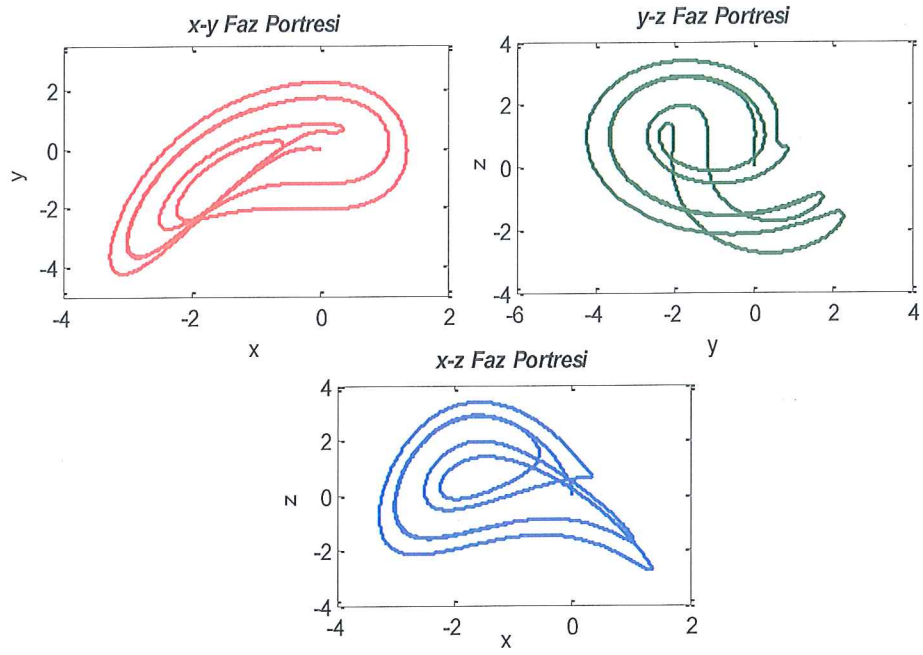
Şekil 4.39. RK5-Butcher tabanlı P3BKS osilatör ünitesi FPGA çip kullanım istatistikleri.

Ayrıca kaotik osilatörlerin FPGA’de gerçekleşmesinden elde edilen X_cikis , Y_cikis ve Z_cikis 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 3x3000 veri seti yardımıyla X_cikis , Y_cikis ve Z_cikis sinyallerinin zaman serileri ve faz portreleri elde edilmiştir. Örnek olarak Şekil 4.40a’da Matlab ve Şekil 4.40b’de FPGA üzerinde gerçekleştirilen RK5-Butcher tabanlı P3BKS’nin $x(t)$, $y(t)$, ve $z(t)$ zaman serileri birbirleriyle karşılaştırılmıştır. Karşılaştırma sonucunda iyi bir eşleşme olduğu gözlemlenmiştir.

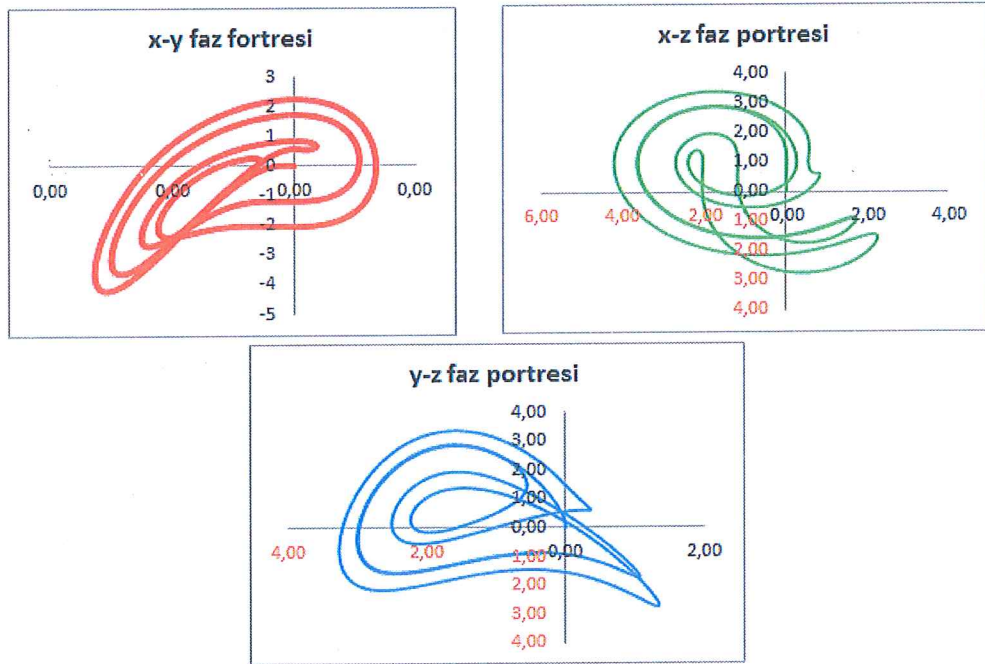


Şekil 4.40. RK5-Butcher tabanlı P3BKS tasarımının $x(t)$, $y(t)$, ve $z(t)$ zaman serilerinin karşılaştırılması. a) Matlab üzerinde tasarım, b) FPGA üzerinde gerçekleştirme.

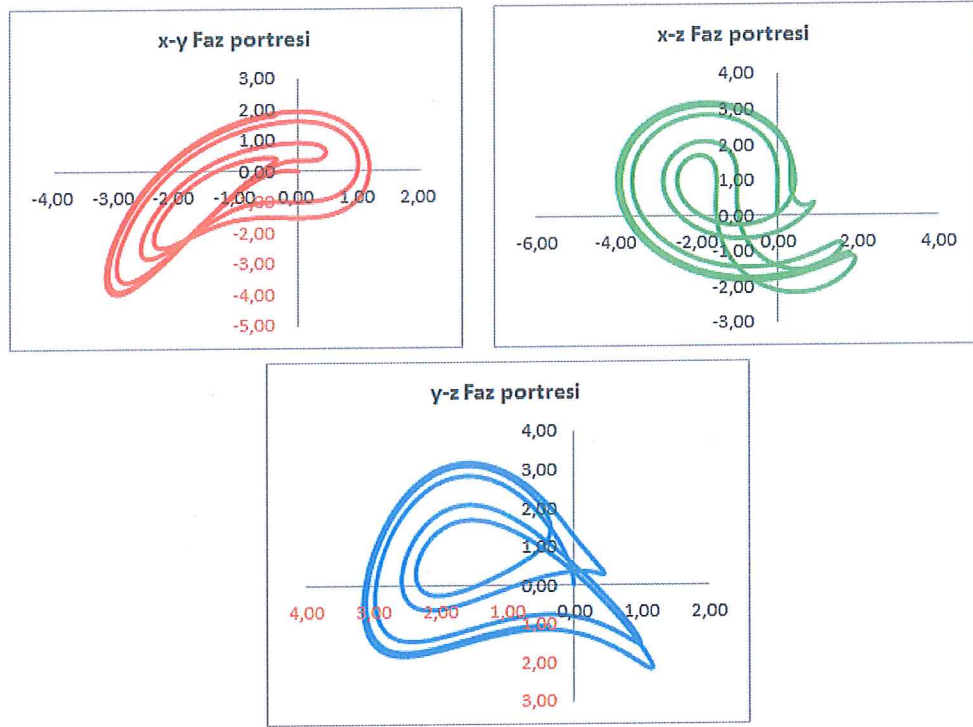
Ayrıca Şekil 4.41’de Matlab üzerinden alınan Euler tabanlı P3BKS osilatörün faz portreleri ile hem Şekil 4.42’de Euler tabanlı hem de Şekil 4.43’te RK5-Butcher tabanlı P3BKS’nin FPGA üzerinde gerçekleştirilmesinden elde edilen faz portreleri verilerek karşılaştırılmıştır. Şekillerden de görüleceği üzere FPGA üzerinde donanım tabanlı gerçekleştirilen tasarımın oldukça başarılı olduğu gözükmemektedir. Diğer tüm tasarımlardan aynı başarılı modelleme sonuçları elde edilmiştir.



Şekil 4.41. Matlab üzerinde Euler-tabanlı P3BKS osilatör ünitesi faz portreleri.



Şekil 4.42. FPGA üzerinde Euler-tabanlı P3BKS osilatör ünitesi faz portreleri.



Şekil 4.43. FPGA üzerinde RK4-tabanlı P3BKS osilatör ünitesi faz portreleri.

Çizelge 4.2’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.

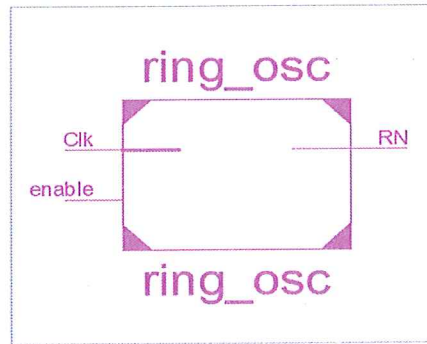
Çizelge 4.2. Yeni Kaotik osilatörlerin Xilinx Virtex-6 FPGA çip istatistikleri.

Kaotik Osilatör	Slice Regs. Sayısı / %	LUTs Sayısı / %	Bonded IOBs Sayısı / %	Maks. Saat Frekansı (MHz)	Minimum Darbe Periyodu (ns)
Kullanılan / Kullanım Oranı%					
Euler-Tabanlı P3BKS	1196 / 0	1070 / 0	99 / 13	464,688	2,152
Heun-Tabanlı P3BKS	2447 / 2	2328 / 5	99 / 13	464,688	2,152
RK4-Tabanlı P3BKS	5338 / 5	5039 / 10	99 / 13	464,688	2,152
RK5-Butcher Tabanlı P3BKS	10794 / 11	10070 / 21	99 / 13	436,143	2,292

4.5. FPGA TABANLI RİNG OSİLATÖR TASARIMI VE TEST SONUÇLARI

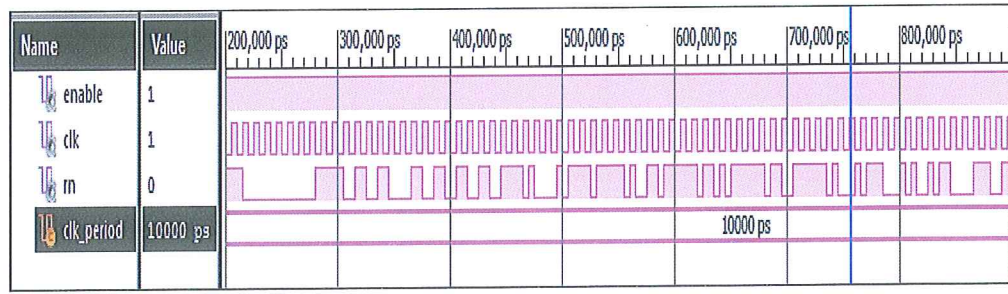
Tümdevre uygulamaları ve FPGA tabanlı gerçek rasgele sayı üreticilerinin çoğu rasgelelik kaynağı olarak ring osilatör yapısını kullanırlar [85,87,89,90,146-148]. Ring osilatörler tek sayılı invertör içeren bağımsız çalışan osilatörlerdir. Halka içindeki her bir evirici sırasıyla iki yarım periyotta üretilen saat sinyalinin yükselen ve düşen kenarını üretmektedir. Halkanın içine bağlanan eviricilerin gecikme kararsızlığı üretilen saat frekansı/faz kararsızlığı (jitter) olarak gözükmemektedir. Jitter örnekleme biriminde flip flop veya latch kullanılarak ayıklanır. Çünkü üretilen sinyalin içindeki jitter doğrudan rasgelelik kaynağı olarak kullanılmaktadır.

Bu bölümde FPGA üzerinde VHDL dili kullanılarak ring osilatör tasarımı gerçekleştirilmiştir. Şekil 4.44'te FPGA üzerinde tasarımı gerçekleştirilen Ring osilatör ünitesinin en üst seviye blok şeması gösterilmektedir. Tüm halka osilatörlerin çıkışlarından yüksek frekanslı rasgele sinyal almak için XOR işlemi uygulanmıştır.



Şekil 4.44. FPGA-tabanlı Ring.osilatör ünitesi en üst seviye blok şeması.

FPGA-Tabanlı Ring Osilatör Ünitesi Xilinx Virtex-6 ailesi xc6vcx75t-3ff784 çipi için sentezlenmiştir. Sistemin gecikmesi 0.635 ns ve çalışma frekansı 1,5 GHz' dir. Tasarımı yapılan ünitelerin verileri işleme süreleri, Xilinx ISE Design Tools 14.2 benzetim programı kullanılarak elde edilmiştir. FPGA üzerinde tasarlanan ring osilatör ünitesinin Xilinx ISE Simülatöründen alınan sonuçlar Şekil 4.45'te ve FPGA çip istatistikleri Şekil 4.46'da verilmiştir.



Şekil 4.45. FPGA üzerinde tasarlanan ring osilatör Xilinx ISE Simülâtör sonuçları.

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	154	46560	0%
Number of fully used LUT-FF pairs	0	154	0%
Number of bonded IOBs	2	360	0%

Şekil 4.46. FPGA üzerinde tasarlanan ring osilatör ünitesi çip kullanım oranları.

4.6. LİTERATÜRDE FPGA TABANLI TASARIMI GERÇEKLEŞTİRİLEN KAOTİK OSİLATÖR TASARIMLARI

Kaotik osilatörlerin FPGA tabanlı modellenmesine yönelik çalışmalara literatürde oldukça fazla önem verilmektedir [26,29,41,50,62,104,108,111–119]. Azzaz vd. yaptıkları çalışmada 3 boyutlu kaotik sistemi Xilinx Virtex-II XC2VP30FFG896-7 FPGA bordu üzerinde VHDL dilinde 32 bit (16Q16) sabit noktalı sayı formatında tasarımını gerçekleştirmişlerdir [42]. Sürekli zamanlı kaotik sistemini sayısallaştırmada Euler nümerik algoritmasını kullanmışlardır. Tasarımın çalışma frekansı 38,86 MHz olarak belirtmişlerdir.

Diğer bir çalışmada Koyuncu vd. 2014 yılında yaptıkları çalışmada literatüre yeni sundukları Pehlivan-Wei kaotik sistemini Xilinx firmasının üretmiş olduğu Virtex-6 FPGA kiti üzerinde gerçekleştirmişlerdir [63]. Yeni kaotik sistemi VHDL dilinde üç farklı nümerik algoritma (Euler, Heun ve RK4) kullanarak tasarlayıp FPGA çip istatistiklerini karşılaştırmışlardır. Gerçekleştirdikleri tasarımın çalışma frekansını 463,688 MHz olarak belirtmişlerdir.

Tuna vd. yaptıkları çalışmada literatüre yeni sundukları kaotik sistemini FPGA üzerinde sayısal olarak gerçekleştirmişlerdir [105]. Sunulan yeni kaotik sistem, FPGA üzerinde IEEE 754-1985 kayan noktalı sayı formatında, Heun algoritması kullanılarak VHSIC HDL (Very High Speed Integrated Circuit Hardware Description Language (Çok Yüksek Hızlı Tümlşik Devre Donanım Tanımlama Dili)) dili ile gerçekleştirilmiştir. Tasarlanan sistem Xilinx Virtex-6 FPGA çipinde sentezlenmiş ve test edilmiştir. Test sonuçlarına göre FPGA-tabanlı yeni kaotik işaret üreticinin çalışma frekansı 390 MHz olarak belirlenmiş ve çip istatistikleri ile performans sonuçları verilmiştir.

Sadoudi vd. güvenli kaotik haberleşme sistemleri için Chen kaotik sistemini FPGA de modellemişlerdir [43]. Bu çalışmada, 32 bit kesirli sayı standardı ile RK4 algoritması kullanılmış ve tasarım bir adet Xilinx Virtex-II çip içeren XCV1000FG456-4 bordu ile test edilmiştir. Yapılan tasarımdan elde edilen sinyaller ile Matlab sonuçları karşılaştırılmıştır. Sistemin çalışma frekansı 22,85 MHz olarak belirtilmiştir.

Eroğlu yaptığı tez çalışmasında Lorenz, Chua, Rössler, Linz ve Sprott kaotik sistemlerini Xilinx Virtex-4 FPGA kiti üzerinde tasarlamıştır [108]. Koyuncu vd. Burke-Shaw kaotik sistemini Xilinx Virtex-6 XC6VCX75T FPGA üzerinde IEEE-754 32 bit kayan nokta sayı standardını kullanarak VHDL dilinde tasarlamışlardır [113]. Kaotik sistemi sayısallaştırmada RK5-Butcher nümerik algoritmasını kullanmışlardır. Sistemin çalışma frekansını 373,094 MHz olarak belirtmişlerdir.

Merah vd. yaptıkları diğer bir çalışmada bilgi güvenliği uygulamaları için Lorenz kaotik sistemini FPGA üzerinde modellemişlerdir [117]. Lorenz kaotik sisteminin tasarım aşamasında donanım tanımlama dillerinden birisi olan VHDL kullanılmıştır. Tasarlanan kaotik sistem Xilinx Spartan-3 FPGA ailesi çipinde gerçekleşmiştir. Tasarımın çalışma frekansı yaklaşık olarak 18 MHz olarak ifade edilmiştir.

Diğer bir çalışmada De micco vd. RK4 algoritmasını kullanarak Lorenz kaotik sistemini FPGA'de gerçekleştirmişlerdir [120]. Kaotik sistem tasarımında IEEE-1985-754 kayan noktalı sayı standardı kullanılmıştır. Test aşamasında kaotik sistem Altera

EP3C120F7 bordu ile test edilmiş ve sistemin çalışma frekansı yaklaşık 1 MHz olarak belirtilmiştir. Literatürde son yıllarda FPGA tabanlı gerçekleştirilen kaotik çekerlerin özellikleri, kullanılan algoritma ile çalışma frekansı gibi teknik özellikleri Çizelge 4.3'te verilmiştir.

Çizelge 4.3. Literatürde FPGA tabanlı gerçekleştirilen kaotik osilatörler ve teknik özellikleri.

Çalışmayı Yapan	Kullanılan Kaotik Çeker	Kullanılan Algoritma	Kullanılan Sayı Stand.	FPGA Özellikleri	Çalışma Frekans MHz
Azzaz vd. [42], 2013.	3D Hibrit kaotik sistem (Lorenz, Lü, Chen)	Euler	32 bits (16Q16) Fixed-point	Xilinx Virtex-II XC2VP30FFG896-7	38,86
Sadoudi vd. [43], 2009.	Chen	RK4	IEEE-754 Floating-point- 32 bit	Xilinx Virtex-II XCV1000FG456-4	22,85
Koyuncu vd. [63], 2014.	Pehlivan-Wei Kaotik sistem	Euler, Heun, RK4	IEEE-754 Floating-point- 32 bit	Xilinx Virtex-6 XC6VCX75T	463,688
Tuna vd., [105] 2015.	Yeni bir kaotik sistem	Heun	IEEE-754 Floating-point- 32 bit	Xilinx Virtex-6 XC6VCX75T	390
Eroğlu [108], 2007.	Lorenz, Chua Röessler, Linz and Sprott,	Simulink and Xilinx Systems Generator	IEEE-754 Floating-point- 32 bit	Xilinx Virtex-4 XC4VSX35-10ff668	-----
Koyuncu vd. [113], 2013.	Burke-Shaw	RK5-Butcher	32 Bit IEEE-754	Xilinx Virtex-6 XC6VCX75T	373,094
Merah vd. [117], 2013.	Lorenz	RK4	32 bits (12Q20) Fixed-point	Xilinx Spartan-3	18
Tlelo-Cuautle vd. [119], 2016.	50-scroll Kaotik çekici	-----	-----	Altera Cyclone IV GX FPGA DE2i- 150	66
De Micco vd. [120], 2011.	Lorenz	RK4	IEEE-754	Altera Ciclone III EP3C120F7	1

BÖLÜM 5

KAOS TABANLI ÇİFT ENTROPİ ÇEKİRDEKLİ GERÇEK RASGELE SAYI ÜRETECİ TASARIMI VE FPGA ÜZERİNDE GERÇEKLEŞTİRİLMESİ

Bu bölümde, daha önceki bölümde tasarımları sunulan FPGA-tabanlı PCKS ve P3BKS tasarımları ile ring osilatör yapısı kullanılarak çift entropi çekirdekli gerçek rasgele sayı üretici tasarımı gerçekleştirilmiştir. Fiziksel yapıda veya sayısal alanda metotlar kullanarak çıkışında otokorelasyon bulunmayan ve istatistiksel olarak birbirinden bağımsız sayılar üreten sistemlere Rastgele Sayı Üretici (RSÜ) denir. Bu üreteçler, önceki veriler yardımıyla daha sonraki verilerin tahmin ve öngörülelemeyeceği rastgelelik seviyesinde çıkış üretebilen yapılardır. Rastgele sayı üreteçleri bu özelliklerinden dolayı birçok değişik alanda kullanılmaktadır. Monte-Carlo metodunun kullanıldığı uygulamalar [121], bilgisayar benzetimleri ve modellemeleri, sayısal analiz uygulamaları, istatistiksel analizler ve özellikler, kriptografide rastgele sayı gereksinimleri bu üreteçler tarafından karşılanmaktadır [122–124].

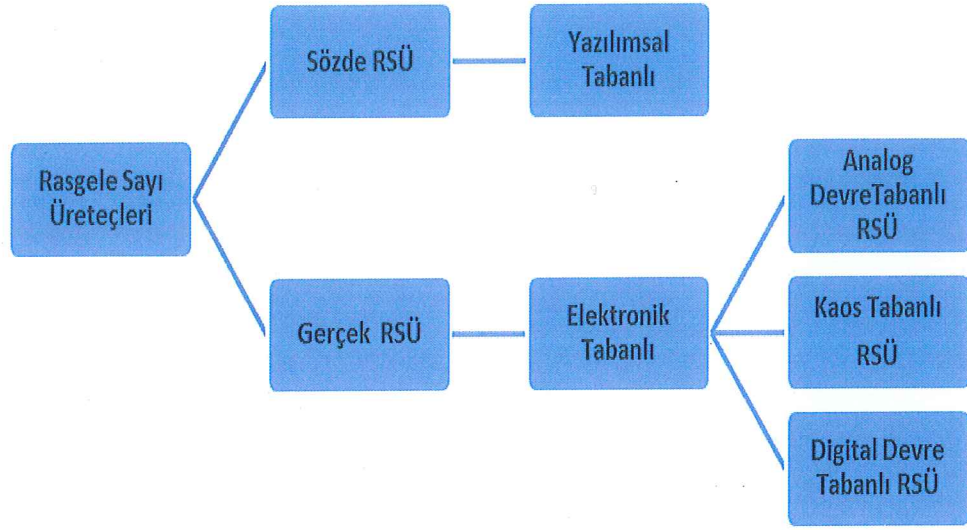
Rasgele sayılar çeşitli kriptografik uygulamalar için zorunluluktur [125]. Çünkü kriptografi; anahtar üretimi ve dağıtımında, başlangıç vektörü oluşturulmasında, kimlik doğrulama protokollerinde, asal sayı ve şifre üretiminde rasgele sayılara ihtiyaç duyar. Bir kriptografik sistemin güvenliği elde edilen sayıların gerçek rastgeleliğine dayanmaktadır. Bu sebeple, rasgele sayı üreteçleri birçok güvenlik alanlarının en önemli bölümünü oluşturur [126]. Uygun olmayan üreteçler ile üretilen rasgele sayılar güvenlik sistemlerini bütünüyle zayıflatabilir. Pek çok programlama dili veya uygulamada bulunan sözde geleneksel RSÜ'leri kriptografik uygulamalar için uygun değildir. Optimal durumda rasgele sayılar, kestirilemez rastgeleliğin gerçek fiziksel kaynaklarını temel almaktadırlar. Bu tip kaynaklar yarı-iletken cihazlardaki gürültüyü, bir ses girişinin en küçük önem bitlerini, kullanıcının klavye tuşlarına vuruşlarını ve farenin

hareketlerini, cihazların kesme noktaları arasındaki aralıkları ve kaotik sistemleri içerebilir. Rastgelelik içermek amacıyla genelde binlerce bitten oluşan geniş bir havuz kullanılır ve havuzdaki her bir bit giriş gürültüsünün her bir bitine ve havuzdaki diğer her bite kriptografik olarak güçlü bir yolla bağımlı yapılır.

Bir saldırgan RSÜ'leri ile üretilmiş rasgele sayıların büyük bir kısmını biliyor olsa bile, rasgele sayıların tahmin edilemez olması gerekliliği açıktır. Kriptografik sistemlerde kullanılan rasgele sayıların bazı sıkı gereksinimleri sağlamaları gerekmektedir. Bu gereksinimler şu şekilde açıklanabilir [124,125]:

1. İyi istatistiksel özellikler ile çıkış bitleri üretebilmelidir,
2. Tahmin edilememe,
3. Düzgün dağılıma sahip olmalıdır,
4. Yüksek entropi ile çıkış bitleri üretebilmelidir,
5. Saldırırlara karşı sağlam olmalıdır,
6. Çevresel değişimlere karşı sağlam olmalıdır,
7. Ürünün kullanım ömrü boyunca sabit özelliklere sahip olmalıdır,
8. Oldukça yüksek veri ve bit hızına sahip olmalıdır,
9. Basit bir uygulama olmalıdır.

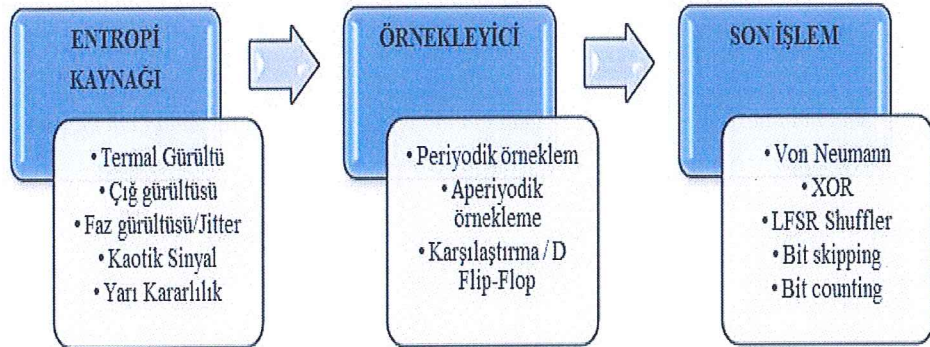
Bilgi güvenliği alanındaki kriptolojik sistemler, belirli devre ve aritmetik işlemlerin birleşiminden oluşmaktadır. Günümüzde donanımsal kriptolojinin gelişimi, kriptolanmış aritmetik yazılımın bir çip içerisine gömülerek gerçekleştirilmesi yönündedir. FPGA ve genel amaçlı mikroişlemciler ile donanımsal olarak rasgele sayı üreten yapıların geliştirilmesi alanında oldukça büyük çabalar sarf edilmektedir. RSÜ'ler Sözde Rasgele Sayı Üreteçleri (SRSÜ) ve Gerçek Rasgele Sayı Üreteçleri (GRSÜ) olmak üzere iki ana sınıfa ayrılmaktadır [43,57,90–92,124–132]. RSÜ'lerin genel olarak sınıflandırılmasına ilişkin diyagram Şekil 5.1'de verilmiştir.



Şekil 5.1. Rasgele sayı üreteçlerinin sınıflandırılması.

5.1. LİTERATÜRDEKİ GRSÜ TASARIMLARI

GRSÜ'lerinin tipik uygulanması Şekil 5.2'de görüldüğü gibi üç ana bölümden oluşmaktadır. İlk olarak, bir FPGA'nın tipik olarak seçirme veya yarı kararlılık durumunda fiziksel rasgele gürültü kaynağıdır. İkinci olarak, örnekleyici, entropi kaynağından rasgelelik toplar ve ikili bit dizisi üretir. Üçüncü olarak son işlem, korelasyonu azaltarak ve veri sıkıştırması yaparak, rasgele bit dizisinin istatistiksel özelliklerini geliştirir.



Şekil 5.2. Geleneksel GRSÜ blok şeması.

Literatürde birçok GRSÜ uygulamaları önerilmiştir. Bunlardan literatürde analog CMOS (Complementary Metal Oxide Semiconductor (Analog Bütünleyici Yarı

İletken Metal Oksit)) tabanlı gürültü devreleri, klasik osilatörler ve kaotik sistemler kullanılarak gerçekleştirilen GRSÜ tasarımları karşılaştırılacaktır. Bu karşılaştırmalar sonucunda son yıllarda FPGA'lar üzerinde kaos tabanlı tasarlanan GRSÜ'lerinin teknik üstünlükleri ortaya konulacaktır.

Literatürde CMOS tabanlı gerçekleştirilen GRSÜ tasarımları üzerine yapılan çalışmalar oldukça yoğundur [43,54,58,81,123,134–142]. Çiçek vd.'nin sunduğu çalışmada CMOS teknolojisi ile ayırık zamanlı kaos tabanlı yeni bir tasarım metodu kullanılarak GRSÜ yapısı geliştirilmiştir [82]. GRSÜ yapısında bulunan kaotik sistem için tek boyutlu harita kullanılmıştır. Yapılan tasarımdan elde edilen rasgele bitler NIST-800-22 testlerine tabi tutulmuş ve 11 testten geçtiği belirtilmiştir.

Ergün vd. yaptıkları çalışmada, otonom olmayan kaotik sistem kullanılarak CMOS teknolojisi ile GRSÜ yapısını önermişlerdir [124]. Önerilen yapının çalışma frekansı 1,24 MHz ve bit üretim hızı 10 Mbit/s olarak belirtilmiştir. Sunulan çalışmadan elde edilen rasgele dizinin NIST-800-22 (National Institute of Standards and Technology) testlerinden başarılı bir şekilde geçtiği vurgulanmıştır.

Tavas vd.'nin 2008 ve 2010 yıllarında yaptıkları çalışmalarda, RSÜ devresinde kullanılabilecek ve tümleştirilmeye uygun yeni kaos üretici önerilmiştir [54,140]. Ayrıca tasarım ortamında yapılan benzetimlerden sonra bu devrelerin 0,35 µm AMS CMOS prosesi ile tümleşik devre üretimleri yapılmış ve GRSÜ oluşturulmuştur. Üretilen kaotik devrelerden 16 MHz ile 25 MHz arasında değişen bant genişlikli kaotik işaretler elde edilmiştir. GRSÜ yapısının doğrulanması için FIPS 140-2, NIST 800-22 ve DIEHARD rastgele sayı testleri uygulanmıştır. Üretimi yapılan GRSÜ devresinin ortalama bit üretim hızı 2 Mbit/s olarak verilmiştir.

Farklı bir çalışma Pareschi vd. tarafından yapılmıştır [141]. Yapılan çalışmada kriptografik uygulamalar için yeni bir kaos tabanlı GRSÜ sunulmuştur. Tasarımın prototipi 0,35 µm CMOS teknolojisi ile gerçekleştirilmiş ve çalışmanın doğrulanması için sisteme NIST testleri uygulanmıştır. Yapılan testlerde sistemin bit üretim hızı 40 Mbit/s'ye kadar çıkmaktadır. Son yıllarda CMOS tabanlı gerçekleştirilen GRSÜ

tasarımlarda kullanılan yöntemler, uygulanan testler, çalışma frekansları ve bit üretim hızları ile ilgili bilgiler Çizelge 5.1’de verilmiştir.

Çizelge 5.1. Literatürde CMOS tabanlı gerçekleştirilen GRSÜ tasarımları ve teknik özellikleri.

Çalışmayı Yapan/ Kaynak	Kullanılan Yöntem	Uygulanan Testler	Tasarım	Çalışma Frekansı (MHz)	Bit Üretim Hızı (Mbit/s)
Tavas vd., 2008 , [55]	Osilatör örnekleme	NIST 800-22 FIPS 140-2	CMOS Teknolojisi	16-25	2
Petrie vd., 2000, [59]	Termal gürültü Osilatör örnekleme	FIPS 140-1	CMOS Teknolojisi	1,4	1,4
Çiçek vd., 2014, [82]	Ayrık zamanlı kaos tabanlı tek boyutlu harita (Lojistik, Tent, Bernoulli)	NIST 800-22	CMOS Teknolojisi	-----	-----
Ergün vd., 2007, [124]	Osilatör örnekleme	NIST 800-22	CMOS Teknolojisi	1,24	10
Dhanuskodi vd., 2014,[134]	Osilatör örnekleme yöntemi	NIST 800-22	CMOS Teknolojisi	-----	127
Aguilar vd., 2014, [135]	Ayrık Zamanlı kaos tabanlı tek boyutlu harita	NIST 800-22 FIPS 140-2	CMOS Teknolojisi	-----	0,250
Moqadasi vd., 2015, [136]	Kaotik Chua Devresi	FIPS 140-1	CMOS Teknolojisi	-----	-----
Park vd., 2015, [137]	Boolean Kaotik Devre	NIST 800-22	CMOS Teknolojisi	-----	-----
Bucci vd., 2003, [138]	Jitter Osilatör örnekleme	NIST 800-22 FIPS 140-2	CMOS Teknolojisi	10	10
Bucci vd., 2016, [139]	Ayrık zamanlı kaos tabanlı	AIS31	CMOS Teknolojisi	50	12,5
Tavas vd., 2010, [140]	Osilatör örnekleme	FIPS 140-2	CMOS Teknolojisi	2,2	2
Pareschi vd., 2010, [141]	Osilatör örnekleme	NIST 800-22	CMOS Teknolojisi	-----	40
Ning vd., 2015, [142]	Ring osilatör	DieHard	CMOS Teknolojisi	20	0,1-20

Literatürde FPGA tabanlı klasik osilatörlerle gerçekleştirilen GRSÜ tasarımları da önemli yer almaktadır [26,86,87,89,131,143,144]. Kohlbrenner vd. yaptıkları GRSÜ tasarımında her biri iki açık tutucu, bir tampon ve bir tümleyenden oluşan osilatör halkası kullanmışlardır [27]. Entropi kaynağı olarak osilatör halkadan üretilen saat sinyalindeki seçirme kullanılmıştır. Halka osilatör saatlerindeki seçirme halka osilatör döngüsü içeren mantık kapılarının kararsız yayılma gecikmelerinden ileri gelmektedir. Örnekleyici devre, diğer halka çıkışı saat girişi ile bağlı iken osilatör halkalardan birinin çıkışı veri girişine bağlı olan D flip-flop’tur. Rasgele bitin çıkışına örnekleyici devre karar verir. Bu kurulumun doğru çalışması için; iki osilatör

halkasının sıklığı, tamamen aynı veya doğru bir şekilde eşleşmiş olmalıdır. Bu GRSÜ'nün çıkışı XOR son işlemine tabi tutulmuştur. Çıkış oranı düşüktür ve elde edilen bit 1 Mbit/s den daha azdır. Çıkış dizisi istatistiksel olarak NIST testine tabi tutularak doğrulanmıştır. Bu tasarımın avantajı, bütün FPGA'lerin ortak kaynaklarını kullanması, matematiksel model uygulanabilmesi, çok az lojik kaynaklar kullanması ve düşük güç tüketimidir.

Danger vd. yaptıkları çalışmada klasik PLL ve Ring osilatörlerin sınırlı bant aralığına sahip olduklarını ve bu osilatörler ile oluşturulan GRSÜ'lerinin bit üretim hızlarının birkaç Mbit/s'yi geçmeyeceğini belirtmişlerdir [52]. Bu probleme çözüm olarak daha yüksek bit üretim hızı sağlayan yeni bir yapı sunulmuş ve bu yapı FPGA ile gerçekleştirilmiştir. FPGA'de açık döngü yarı kararlılığına dayanan bir GRSÜ önermişlerdir. Bu tasarımın prensibi gecikme dizisidir ve dizideki birkaç kademe noktasından örneklenmiş sinyaller birlikte XOR edilerek rasgele bir sinyal oluşturmak için örneklenmektedir. Elemanlar arasındaki gecikme önemli ve bu duruma karşı özel özen gösterilmelidir. Bu GRSÜ'nün rapor edilen bit oranı 20 Mbit/s'dir. Sunulan yeni yapının doğrulanması NIST testleri ile yapılmıştır. Sonuç olarak yapının çalışma frekansı 20 MHz olarak verilmiştir.

Bir diğer çalışma István vd. tarafından FPGA tabanlı klasik jitter osilatör yöntemi kullanılarak yapılmıştır [58]. Önerilen yöntemin doğruluğu NIST 800-22 ve TestU01 testleriyle sağlanmıştır. Sistemin çalışma frekansı 50 MHz bulmakla beraber, kullanılan osilatör nedeniyle sistemin bit üretim hızı 1,92 Mbit/s'yi aşmamaktadır.

Sunar vd. tarafından 2007 yılında halka osilatöre dayalı FPGA tabanlı yeni bir GRSÜ tasarımı önerilmiştir [89]. Temel olarak serbest salınımlı halka osilatörlerin çıkışları XOR işlemi ile birlikte toplanır ve sonra örneklenir. Rasgelelik kaynağı faz seçirmesidir. Halka osilatör tek sayıda tümleyen içeren gecikmelerin birleşimsel döngüsüdür. Bu tasarımda her biri 13 tümleyenden oluşan 114 adet halka osilatör kullanılmaktadır. Kullanılan halka osilatörlerin sayısı, ölçülür seçirmelere göre belirlenir. Tüm halka osilatörlerin çıkışları yüksek frekanslı rasgele sinyal almak için XOR işlemine girer. XOR işleminin çıkışı da düşük frekanslı saat frekansı ile örneklenir. Daha sonra resilient fonksiyon son işlemi kullanılarak, rasgele sinyaldeki

birler ve sıfır arasındaki orantısızlıklar düzeltilebilir. Bu GRSÜ'de son işlemi de içererek başarılan bit oranı 2,5 Mbit/s'dir. Çıkış bit dizileri Diehard ve NIST testleri kullanılarak doğrulanmıştır.

Wold vd. 2008 yılında yaptıkları çalışmada, Sunar tasarımının [89] her bir osilatör halkasına D flip-flop ekleyerek ve XOR'lar arasındaki hız sorununu giderecek yeni bir yapı önermişlerdir [90]. Burada önerilen GRSÜ'de sadece 25 halka osilatör kullanarak Diehard ve NIST 800-22 istatistiksel testlerini başarılı bir şekilde geçmişlerdir. Elde edilen çıkış hızı 100 Mbit/s olmuştur.

Diğer bir başka çalışmada Fischer vd. PLL tabanlı osilatörü FPGA çipleriyle gerçekleştirmişlerdir [143]. GRSÜ yapısı VHDL'de tanımlanmış ve Altera firmasının Quartus-II programı kullanılarak gerçekleştirilmiştir. Kriptografik uygulamalar için tasarlanan GRSÜ'nin performans analizi NIST testleri ile yapılmıştır. Testlerden elde edilen sonuçlara göre tasarlanan sistemin bit üretim hızı 70 Kbit/s olarak elde edilmiştir. Kolay tasarım ve uygulamaları, sentezin bütünüyle FPGA araçları içinde yapılabilmesi, sabit çıkış hızına sahip olması ve düşük güç tüketimi bu tasarımın avantajları olarak gösterilebilir. Düşük çıkış hızı ve bu GRSÜ tasarımının, analog PLL içeren FPGA'ler de sınırlandırılmış olması bu yapının dezavantajlı yanıdır.

Schellekens vd. çoklu ring osilatörünü FPGA'de modellemişlerdir. FPGA çipi olarak Xilinx firmasının Virtex-II çipi kullanılmıştır [145]. Tasarımı yapılan sistemin örnekleme zamanı 40MHz olarak verilmiştir. Tasarımın rasgelelik kontrolü standart NIST 800-22 testleri ile yapılmıştır. Test sonuçlarına göre GRSÜ'nin bit üretim hızı 2,5 Mbit/s olduğu ifade edilmiştir.

Dichtl vd. FPGA'de Fibonacci ve Galois ring osilatörünü modellemişlerdir [146]. Tasarımın gerçekleştirilmesi Xilinx firmasının üretmiş olduğu FPGA çiplerinden Spartan-3 çipini içeren Starter kit kullanılarak yapılmıştır. Sistemin ürettiği bit üretim hızı 12,5 Mbit/s olarak verilmiştir. Ayrıca Dichtl ve arkadaşları bu çalışmada Sunar [89] tarafından gerçekleştirilen tasarımı eleştirmişlerdir. Bu eleştiriler arasında örnekleme oranı ve halka osilatörlerin bağımsızlık varsayımı vardır.

Tuncer vd. yaptıkları çalışmalarında GRSÜ'lerinin çıkış hızını düşürmeden, istatistikî zayıflıkları gideren yeni bir son işlem algoritması önermişlerdir [147,148]. Bu uygulamada entropi kaynağı olarak halka osilatörler tarafından üretilen faz seğirmesi kullanılarak FPGA ortamında donanımsal GRSÜ gerçekleştirilmiştir. Önerilen son işlem kaotik davranış sergileyen lojistik haritadır. Lojistik harita 1. dereceden bir denklem olup bir başlangıç koşulu ve bir kontrol parametresine gerektirmektedir. Bu sebepten dolayı donanım üzerinde gerçekleştirilmesi kolaydır. Gerçekleştirilen RO tabanlı bir GRSÜ sisteminde lojistik haritanın üretilen sayılara etkisini gözlemleyebilmek için dört farklı senaryo geliştirilmiştir. GRSÜ sisteminde kullanılan RO sayıları ve RO'ler deki tümleyen sayıları sırasıyla (114,13) (25,3) (10,3) (5,3) seçilmiştir. Tüm senaryolarda son işlem olarak lojistik harita kullanılmış olup her bir sistem Altera'nın EP4CE115F29C7 tabanlı FPGA bordunda gerçekleştirilmiştir. Burada lojistik haritanın son işlem olarak kullanımını gösterebilmek için Sunar tarafından önerilen [89] RO tabanlı GRSÜ sistemi kullanılmıştır. Sunar tarafından üretilen sistemde entropi kaynağından üretilen rasgele bit dizilerinde korelasyon olması nedeniyle son işlemsiz olarak testleri geçememiştir. Bunun sonucunda resilient fonksiyonu olarak adlandırılan son işleme tabi tutulmuş olup çıkış hızı ve oranı 1/16 oranında azalmıştır. Bu amaçla sistemin entropi kaynağında bulunan olası eksikliği gidermek için lojistik harita son işlem olarak önerilmiştir. Lojistik haritanın rasgeleliğinin tahmin edilememe ve rasgelelik özellikleri katması nedeniyle GRSÜ sisteminin güvenliği artacaktır. Sunar sisteminde çıkış hızı 2,5 Mbit/s iken geliştiren sistemde yaklaşık 20 Mbit/s olmuştur. Her bir sistem tarafından elde edilen sayıların istatistiksel testleri NIST 800-22 testine göre elde edilmiştir. Test sonuçlarına göre lojistik haritanın son işlem olarak kullanılabileceği gösterilmiştir.

RO tabanlı GRSÜ'lerin avantajı, bağımsız teknoloji, kolay tasarım ve uygulamaları, sentezin bütünüyle FPGA araçları içinde yapılabiliyor olması, nispeten sabit ve yüksek çıkış hızına sahip olmalarıdır. Ancak RO sayısının çokluğundan dolayı yüksek güç tüketimi, halka osilatörlerin bağımsız olmaması rasgelelik kalitesinin düşmesine ve çıkışın korelasyonlu olması, resilient fonksiyonundan dolayı saldırıların tespit edilemeyecek olması, güç tüketimi dolayısıyla aşırı lokal ısınmalar tasarımın dezavantajlarıdır. Klasik osilatörler kullanılarak son birkaç yılda FPGA

üzerinde gerçekleştirilen GRSÜ tasarımlarında kullanılan osilatör, FPGA kart özellikleri, uygulanan testler, çalışma frekansı ve bit üretim hızı gibi teknik özellikler Çizelge 5.2’de verilmiştir.

Çizelge 5.2. Literatürde FPGA tabanlı Klasik osilatörlerle gerçekleştirilen GRSÜ tasarımları ve teknik özellikleri.

Çalışmayı Yapan	Kullanılan Teknik	FPGA Özellikleri	Uygulanan Testler	Çalışma Frekansı (MHz)	Bit Üret. Hız (Mbit/s)
Kohlbrener vd., 2004,[27]	Ring osilatör	Xilinx Virtex XCV1000	NIST 800-22	-----	< 1
Jessa vd., 2010, [49]	Ring osilatör	Xilinx Virtex-5	NIST 800-22	300	7,14
Danger vd., 2009, [52]	Yeni bir yapı	Xilinx Spartan-3	NIST 800-22	20	20
Wold vd., 2009, [57]	Ring osilatör	Altera Cyclone II	NIST 800-22 DIEHARD	300	-----
Istvan vd., 2009, [58]	Klasik jitter	Xilinx Spartan3E	NIST 800-22 TestU01	50	1,92
Sunar vd., 2007, [89]	Ring osilatör	Xilinx Virtex-2	NIST 800-22 DIEHARD	-----	2,5
Wold vd., 2008, [90]	Ring osilatör	Altera Cyclone II	NIST 800-22 DIEHARD	-----	100
Çiçek vd., 2014, [92]	Ayrık Kaotik sistem (Bernoulli map FPAA)	Xilinx Spartan XC3S1600E	NIST 800-22	2	1,5
Wieczorek vd., 2014, [117]	Çift kararlı Flip/Flop	Xilinx Spartan3E	NIST 800-22	50	5
Lozach vd., 2013, [118]	Open Loop Metastability	Xilinx Virtex-5 XC5VLX50T	AIS.31 NIST 800-22	22	20
Fischer vd., 2004, [143]	PLL osilatör	Altera Quartus II	NIST 800-22	-----	1
Schellekens vd., 2006, [145]	Çoklu ring osilatör	Xilinx Virtex II	NIST 800-22	40	2,5
Ditchev vd., 2008, [146]	Ring osilatör	Xilinx Spartan-3	NIST 800-22	-----	12,5
Tuncer ve Avaroğlu, 2015, [147, 148]	Ring osilatör	Altera EP4CE115F29C7	NIST 800-22 TESTU01	450	25

Literatürde FPGA üzerinde tamamen sayısal diferansiyel kaotik sistemleri kullanarak GRSÜ tasarımları yer almaktadır. Koyuncu [149] doktora tezinde 2 adet yeni kaotik sistemi 4 farklı algoritma ile 3 ayrı kuantalama tekniği kullanarak VHDL dilinde 32 bit IEEE 754-1985 kayan nokta sayı standardında Xilinx Virtex-6 FPGA çipi üzerinde GRSÜ tasarlamıştır. Tasarım sonucunda 24 adet farklı GRSÜ sonuçları elde etmiştir. Elde ettiği sonuçlar üzerine uluslararası rasgelelik testleri olan NIST 800-22 ve FIPS 140-1 testlerini uygulayarak optimum sonucu veren kuantalama ve algoritmayı ortaya çıkarmıştır.

Zidan vd. yaptıkları çalışmada gömülü, tamamen sayısal diferansiyel kaosa dayalı GRSÜ önermişlerdir [150]. Sayısal devrenin çıkışı maksimum Lyapunov üslerinin çıkış zaman serilerini hesaplayarak kaotik olduğunu kanıtlamıştır. Sistemin çözümünde dördüncü dereceden Runge Kutta, orta nokta ve Euler tekniği olmak üzere üç farklı metod kullanılmıştır. Sistem, VHDL’de kodlanarak ve Xilinx Virtex 4 FPGA çipi üzerinde gerçekleştirilmiştir. Devre oldukça küçük bir alan kaplamış ve çıkış hızı 2,1 Gbit/s olarak elde edilmiştir. Yeni önerdikleri son işlem den geçirildikten sonra NIST testine tabi tutulmuş ve başarıyla geçmiştir.

Koyuncu vd. yaptıkları çalışmada Sprott 94G kaotik sistemini Euler nümerik algoritması ile IEEE 754-1985 kayan nokta sayı formatında VHDL programlama dilini kullanarak Xilinx Virtex-6 FPGA çipinde tasarlamışlardır [151]. Yaptıkları tasarımda kuantalama ünitesinden çıkan ikili sayı bitleri üzerine düzeltici fonksiyon olarak XOR yöntemini uygulamışlardır. Tasarım sonucunda elde ettikleri sayı dizilerinin rasgeleliğini kanıtlamak için uluslararası test standartları olan NIST 800-22 ve FIPS 140-1 testlerini uygulayarak testlerden başarılı bir şekilde geçtiklerini yaptıkları çalışmada göstermişlerdir.

Wang vd. yaptıkları çalışmada yeni bir metodoloji geliştirerek sonlu karar mekanizması yöntemi ile yüksek boyutlu sayısal kaotik sistem tasarımı gerçekleştirmişlerdir [152]. Önerdikleri sistemi rasgele sayı üretiminde kullanmışlardır. Bu çalışmada tasarlanan sistemin Devaney’in kaos tanımını karşıladığını göstererek sistemin Lyapunov üstelleri hesaplanmıştır. Ayrıca bu çalışmada tasarladıkları sistem ile iki adet Altera DE2-FPGA geliştirme boardu kullanarak 3 boyutlu bir görüntü şifrelemesini başarılı bir şekilde gerçekleştirerek kaosun sayısal dünyadaki uygulamasına teşvik etmişlerdir.

Bir diğer çalışmada Koyuncu vd. geliştirdikleri kaotik sistemi kullanarak FPGA üzerinde yüksek hızlı kaos tabanlı gerçek rasgele sayı üretici geliştirmişlerdir [153]. Tasarladıkları kaotik sistemin zaman serilerini, faz portrelerini çıkartarak Lyapunov üstellerini hesaplamışlardır. Ayrıca kaotik sistemi hem PSpice bilgisayar yazılımı üzerinde hem de analog devre üzerinde tasarlamışlardır. Her iki tasarımdan aldıkları sonuçlar üzerinde başarılı bir eşleşme elde etmişlerdir. İkinci olarak yeni kaotik

sistemi RK4 metodunu kullanarak donanım tanımlama dili VHDL ile Xilinx Virtex-6 FPGA çipi üzerinde tasarlayarak kaotik osilatörü analiz ve test etmişlerdir. Nümerik olarak bilgisayar üzerinden alıkları kaotik osilatör sonuçlarıyla FPGA'dan aldıkları sonuçları karşılaştırmışlardır. Yine aynı FPGA geliştirme bordu üzerinde tasarladıkları yeni kaotik sistemi kullanarak kaos tabanlı gerçek rasgele sayı üretici gerçekleştirmişlerdir. Tasarımın çalışma frekansını 293 MHz ve bit üretim hızı 58,76 Mbit/s olarak vermişlerdir. FPGA üzerinden aldıkları rasgele sayıları test etmek amacıyla uluslararası FIPS-140-1 ve NIST 800-22 istatistiki testlerini kullanmışlardır. Bu çalışma ile literatüre sundukları yeni sayısal kaos tabanlı gerçek rasgele sayı üreticinin gömülü kriptoloji uygulamalarında kullanılabileceğini göstermişlerdir.

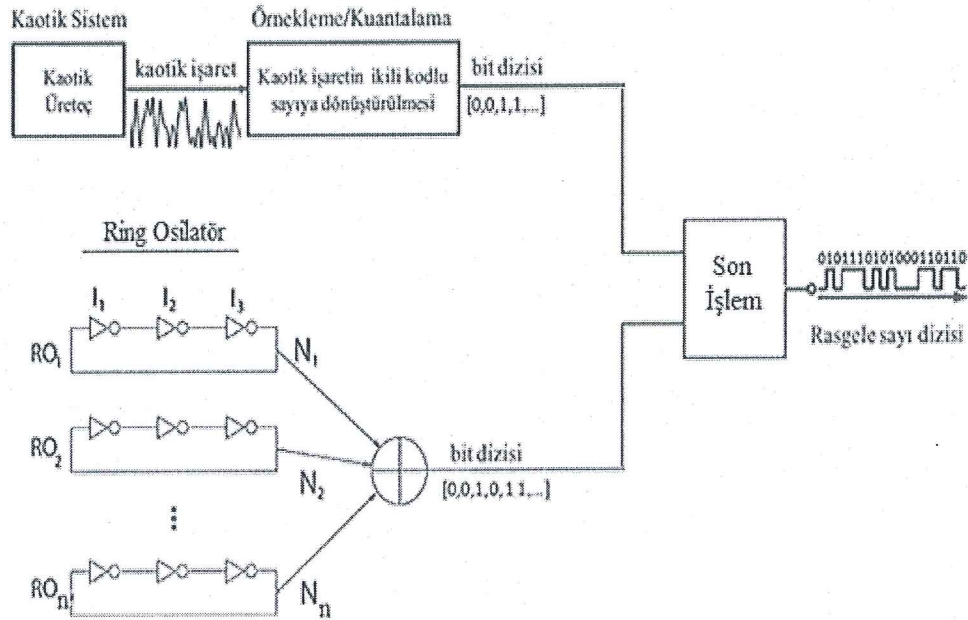
Literatürde son birkaç yılda FPGA üzerinde tamamen sayısal diferansiyel kaotik devre tabanlı gerçekleştirilen GRSÜ tasarımlarında kullanılan osilatör, FPGA kart özellikleri, uygulanan testler, çalışma frekansı ve bit üretim hızı gibi teknik özellikler Çizelge 5.3'te verilmiştir.

Çizelge 5.3. Literatürde FPGA üzerinde tasarlanan tamamen sayısal diferansiyel kaotik devre tabanlı GRSÜ.

Çalışmayı Yapan	Kullanılan Kaotik Sistem Tasarım özellikleri	Son İşlem	FPGA Özellikleri	Uygulanan Testler	Çalış. Frek. (MHz)	Bit Üretim Hızı
Fatemi, vd., 2016, [46]	Kaotik harita (Bernoulli Map)	XOR	FPGA	NIST 800-2	-----	-----
Koyuncu, Doktora Tez, 2014, [149]	2 adet yeni kaotik sistem 4 farklı Algoritma 3 farklı Kuantalama 32 bit IEEE 754-1985 Kayan nokta sayı	XOR	Xilinx Virtex-6 XC6VLX550T-2FF1759	NIST 800-22 FIPS 140-1	339-401	53-132 Mbit/s
Zidan vd., 2011, [150]	Lyapunov üslerinin çıkış zaman serileri 3 Farklı algoritma	Yeni son işlem önerilmiş	Xilinx Virtex 4	NIST 800-22	-----	2 Gbit/s
Koyuncu vd., 2014, [151]	Sprott 94G Euler algoritma 32 bit IEEE 754-1985	XOR	Xilinx Virtex-6 XC6VCX75T	NIST 800-22 FIPS 140-1	400	-----
Wang vd. 2016,[152]	Yüksek boyutlu sayısal kaotik sistem	XOR	Altera DE2 Cyclone IV	NIST 800-2 Diehard	50	-----
Koyuncu vd., 2016, [153]	Sundarapandian-Pehlivan kaotik sistemi (SPCS) Runge-Kutta (RK4) 32 bit IEEE 754-1985	XOR	Xilinx Virtex-6 C6VLX240T-1-FF1156	FIPS 140-1 NIST 800-22	293	58,76 Mbit/s

5.2. KAOS TABANLI ÇİFT ENTROPİ ÇEKİRDEKLİ GERÇEK RASGELE SAYI ÜRETECİ TASARIMI VE FPGA ÜZERİNDE GERÇEKLEŞTİRİLMESİ

Bu bölümde, daha önceki bölümde tasarımları sunulan FPGA-tabanlı PCKS ve P3BKS tasarımları ile ring osilatör yapısı kullanılarak çift entropi çekirdekli gerçek rasgele sayı üretici tasarımı gerçekleştirilmiştir. Tasarımını gerçekleştirmeyi düşündüğümüz GRSÜ blok şeması Şekil 5.3'te verilmiştir. GRSÜ için yapılan FPGA-tabanlı tasarım daha önceki bölümde sunulan Euler, Heun, RK4 ve RK5-Butcher algoritmaları kullanılarak yapılan tasarımların her birine uygulanmıştır.



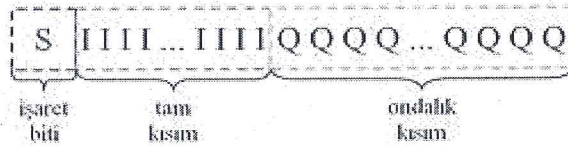
Şekil 5.3. Ayırık zamanlı önerilen çift entropi çekirdekli yeni GRSÜ tasarımı.

Genel olarak iki farklı kaotik sistem, dört ayrı algoritma ve bir Kuantalama yöntemi ile toplamda 8 farklı çift entropi çekirdekli GRSÜ ünitesi tasarlanmıştır. Kuantalama için sabit noktalı sayı yöntemi sunulmuştur. Yapılan tasarımlar sonucunda elde edilen FPGA-tabanlı kaotik GRSÜ ünitelerinin performans ve çip istatistikleri incelenmiştir.

5.2.1. FPGA-Tabanlı Gerçek Rasgele Sayı Üretici Tasarımı

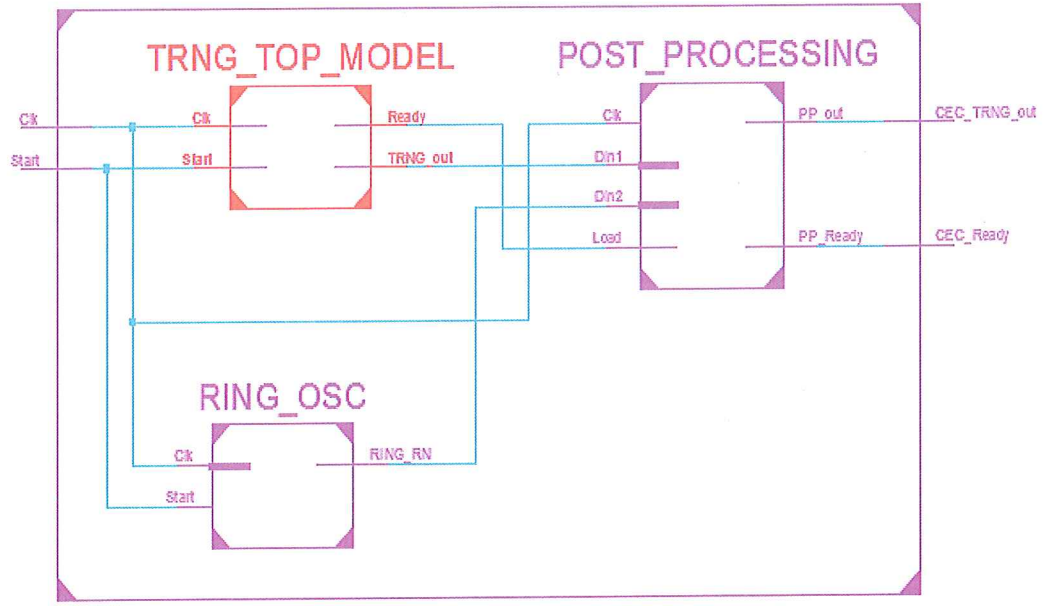
5.2.1.1. Sabit Noktalı Sayı (SNS) Tabanlı GRSÜ

Şekil 5.4'te gösterilen Texas Instruments firması tarafından kullanılan IQ-Math SNS formatı sayının işaret, tam kısmı ve ondalık kısmı olmak üzere üç kısma ayrılır [153,154]. Burada "S" işaret bitinin değerini ifade eden MSB seviyesinde en son bittir. Bu değer "0" olduğunda sayı pozitif, "1" olduğunda sayı negatif olmaktadır. Bu sayı formatında tasarlanacak sisteme göre, istenilen sayının maksimum büyüklüğü ve hassasiyet göz önünde bulundurularak tam ve ondalık kısmının uzunlukları ayrı ayrı belirlenir [155,156].



Şekil 5.4. IQ-Math sabit noktalı sayı formatı gösterimi.

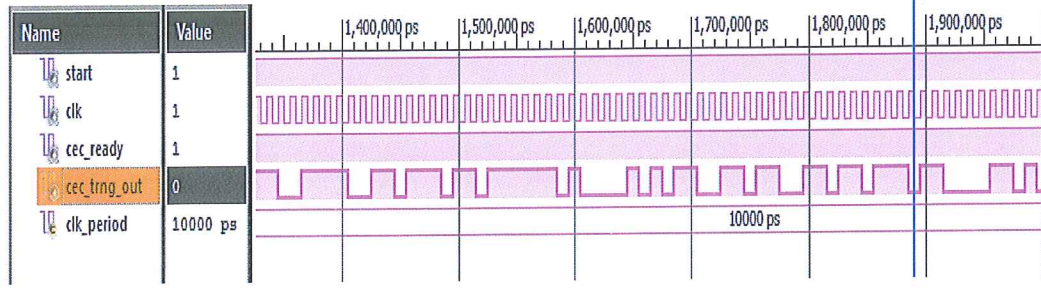
Bu tez çalışması için başlangıçta Şekil 5.3'te önerdiğimiz yeni çift entropi çekirdekli GRSÜ'nin FPGA üzerinde SNS tabanlı tasarlanan blok diyagramı Şekil 5.5'te verilmiştir. *Kuantalama* ünitesi içerisinde bulunan *Sayı Üreteci*nde 31'inci bit (MSB) işaret biti olarak isimlendirilmektedir. İşaret (significant) biti değeri eğer '0' ise sayı pozitif, eğer '1' ise sayı negatif olmaktadır. Kesirli kısımda en düşük değerlikte veya en yüksek hassasiyete sahip olan bit ise 0'ıncı bit (LSB) olmaktadır. Bu bit değeri sayı değerinin çok az değişmesiyle değişebilmektedir. SNS tabanlı GRSÜ yapısı kaotik osilatör ünitesinin her bir ürettiği sabit noktalı sayı standardındaki 32 bitlik sayının kesirli kısmındaki 14 bitinin alınması ile gerçekleştirilmiştir.



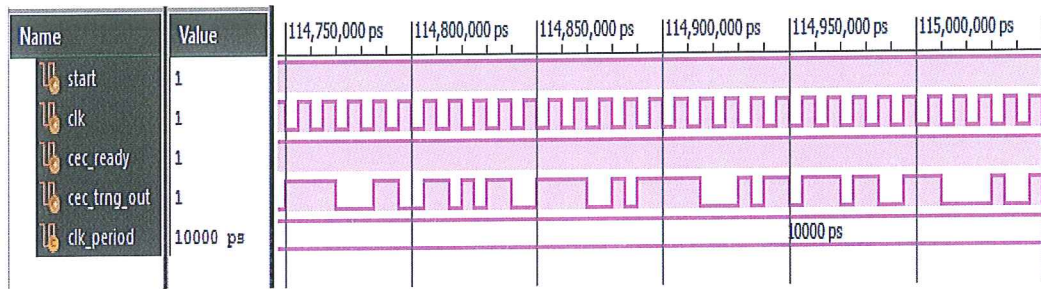
Şekil 5.5. FPGA üzerinde SNS formatında tasarlanan kaos tabanlı çift entropi çekirdekli yeni GRSÜ blok şeması.

5.2.1.2. Gerçek Rasgele Sayı Üreticinin FPGA Üzerinde Gerçekleştirilmesi

Bu bölümde PCKS ve P3BKS tasarımları Euler, Heun, RK4 ve RK5-Butcher algoritmaları kullanılarak SNS tabanlı yapı ile tasarımı yapılan 8 farklı GRSÜ ünitesi Xilinx firmasının ürettiği Virtex-6 ailesinin XC6VLX75T-3FF784 çipi için sentezlenerek, FPGA çip kaynak kullanımına ve ünitelerin saat hızlarına ait parametrelerin istatistikleri incelenmiştir. Tasarımı yapılan 8 farklı yapıdaki GRSÜ ünitelerinin verileri işleme süresi, Xilinx ISE Design Tools 14.2 simülasyon programı kullanılarak elde edilmiştir. PCKS tasarımı kullanarak FPGA üzerinde gerçekleştirilen SNS-Tabanlı çift entropi çekirdekli GRSÜ üniteleri için Xilinx ISE Simülatöründen elde edilen sonuçlar Euler algoritması için Şekil 5.6, Heun algoritması için Şekil 5.7, RK4 algoritması için Şekil 5.8 ve RK5-Butcher algoritması için Şekil 5.9’da verilmiştir.



Şekil 5.6. Euler algoritması ile tasarlanan PCKS tabanlı çift entropi çekirdekli GRSÜ ünitesi Xilinx ISE Simülatörü sonuçları.



Şekil 5.7. Heun algoritması ile tasarlanan PCKS tabanlı çift entropi çekirdekli GRSÜ ünitesi Xilinx ISE Simülatörü sonuçları.

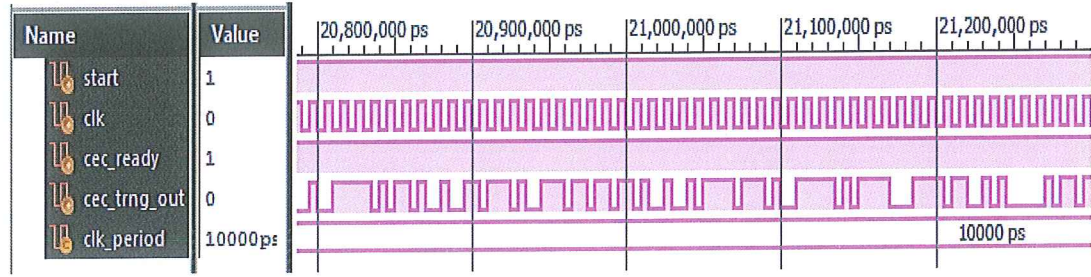


Şekil 5.8. RK4 algoritması ile tasarlanan PCKS tabanlı çift entropi çekirdekli GRSÜ ünitesi Xilinx ISE Simülatörü sonuçları.

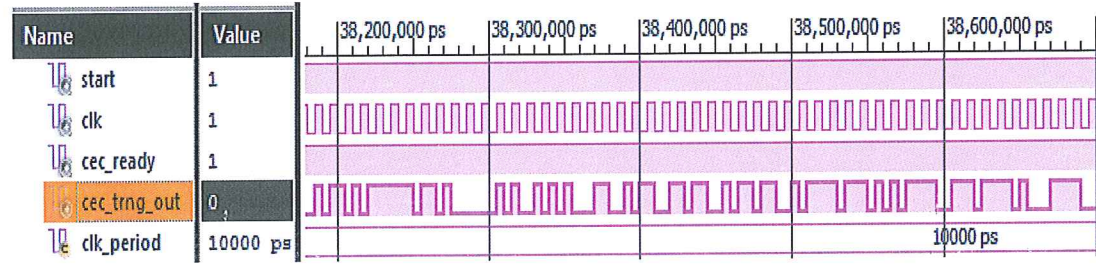


Şekil 5.9. RK5-Butcher algoritması ile tasarlanan PCKS tabanlı çift entropi çekirdekli GRSÜ ünitesi Xilinx ISE Simülatörü sonuçları.

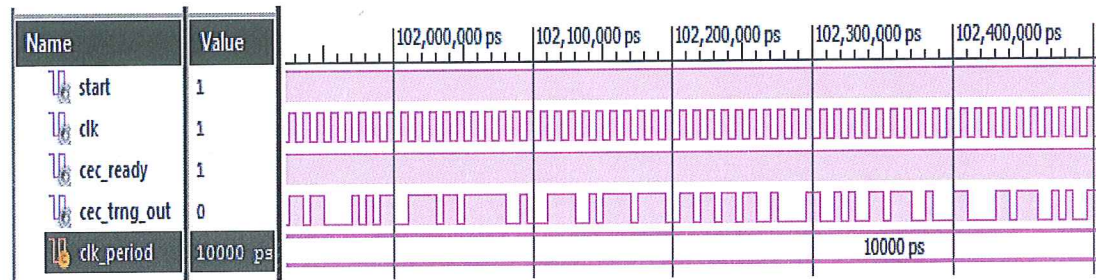
P3BKS kullanılarak FPGA üzerinde gerçekleştirilen SNS-Tabanlı çift entropi çekirdekli GRSÜ üniteleri için Xilinx ISE Simülatöründen elde edilen sonuçlar Euler algoritması için Şekil 5.10, Heun algoritması için Şekil 5.11, RK4 algoritması için Şekil 5.12 ve RK5-Butcher algoritması için Şekil 5.13'te verilmiştir.



Şekil 5.10. Euler algoritması ile tasarlanan P3BKS tabanlı çift entropi çekirdekli GRSÜ ünitesi Xilinx ISE Simülatörü sonuçları.



Şekil 5.11. Heun algoritması ile tasarlanan P3BKS tabanlı çift entropi çekirdekli GRSÜ ünitesi Xilinx ISE Simülatörü sonuçları.



Şekil 5.12. RK4 algoritması ile tasarlanan P3BKS tabanlı çift entropi çekirdekli GRSÜ ünitesi Xilinx ISE Simülatörü sonuçları.



Şekil 5.13. RK5-Butcher algoritması ile tasarlanan P3BKS tabanlı çift entropi çekirdekli GRSÜ ünitesi Xilinx ISE Simülatörü sonuçları.

Çizelge 5.4. SNS-Tabanlı PCKS ve P3BKS ile tasarlanan çift entropi çekirdekli GRSÜ ünitelerinin FPGA çip istatistikleri.

Kaotik Osilatör	Nümerik Algoritma	Slice Register Sayısı	LUTs Sayısı	LUT-FF Çifti Sayısı	Bonded IOBs Sayısı	Maksimum Saat Frekansı (MHz)	Bit Üretim Hızı (Mbit/s)
Kullanılan /Kullanım Oranı%							
PCKS	Euler	1591 / 1	1606 / 3	1159 / 56	4 / 1	462,731	462,731
PCKS	Heun	3118 / 3	3044 / 6	2275 / 58	4 / 1	462,731	462,731
PCKS	RK4	6583 / 7	6144 / 13	4726 / 59	4 / 1	436,172	436,172
PCKS	RK5-Butcher	12564 / 13	11405 / 24	8827 / 62	4 / 1	436,172	436,172
P3BKS	Euler	1318/1	1355/3	1050/54	4 / 1	464,688	464,688
P3BKS	Heun	2575/2	2758/5	1575/55	4 / 1	464,688	464,688
P3BKS	RK4	5785/5	5523/10	3750/58	4 / 1	464,688	464,688
P3BKS	RK5-Butcher	11741/11	10375/21	6487/59	4 / 1	436,143	436,143

BÖLÜM 6

FPGA ÜZERİNDE GERÇEKLEŞTİRİLEN KAOS TABANLI ÇİFT ENTROPİ ÇEKİRDEKLİ GERÇEK RASGELE SAYI ÜRETEÇLERİNİN RASGELELİK TEST SONUÇLARI

Kriptolojik uygulamalarda kullanılmak üzere geliştirilen sözde rasgele ve gerçek rasgele sayı üreteçlerinin rasgelelik ve istatistiksel özelliklerinin incelenmesi ve test edilmesi gerekmektedir. Bu amaçla literatürde geliştirilmiş çeşitli istatistiksel testler kullanılmaktadır. Bu bölümde, FPGA üzerinde geliştirilen çift entropi çekirdekli yeni kaotik GRSÜ üniteleri, kriptolojik uygulamalarda güvenli bir şekilde kullanılabilmesi amacıyla uluslararası FIPS 140-1 [158] ve NIST 800-22 [159] istatistiksel testlere tabi tutulacaktır. Bu iki farklı test uygulamaları rasgele sayıların rasgeleliğini test etmede literatürde oldukça fazla kullanılmaktadır [34,40,45,51,54–56,58,81,86,91,92,123,126,127,149,151,152,159–162].

6.1. RASTGELELİK TESTLERİ

Sayı üreteçlerinin çıkışının rasgele olduğu, matematiksel olarak kanıtlanamamasına rağmen, geçerli istatistiksel testleri uygulayarak, sayı dizilerinin rasgele olmadığını veya bunun tersini söyleyebiliriz. Bu testler, üretcin çıkışının gerçek bir rasgele diziden beklenenleri karşılayıp karşılamadığını söyler. Ayrıca testlerin sonuçlarına bakılarak RSÜ'nin kalitesi hakkında yorum yapılabilir. Bir sayı dizisinin rasgele olduğunu söylemek için, tüm testlerden geçmesi gerekir. Sadece bir tane test başarısız olsa bile dizi rasgele kabul edilemez.

Bilinen testlerden başlıca iki tanesi Ulusal Standartlar ve Teknoloji Enstitüsü tarafından yayınlanan FIPS 140-1 ve NIST 800-22'dir. FIPS 140-1 testi blok uzunluğu küçük olan verilerin testinde kullanılır. NIST 800-22 testi ise görece uzun bloklardan oluşan verileri test etmek için kullanılır ve FIPS 140-1 testine göre

ölçütler daha zorlayıcıdır. Bu nedenle kullanım amacına uygun olarak bu testlerden bir veya bir kaç tercih edilebilir [84,133,163,164].

6.1.1. FIPS 140-1 Testi

FIPS 140-1 testi dört tane ayrı testten oluşur [84,164]. RSÜ'nin çıkışından alınan ve 20 bin tane bit içeren bir bit dizisi dört teste birden tabi tutulur ve dizinin rasgele olabilmesi için tüm testlerden geçmesi gerekir. Bu testler aşağıda kısaca açıklanmıştır.

6.1.1.1. Monobit Testi (Monobit Test)

Bu testin amacı, bit dizisindeki 0 ve 1 sayısının rasgele bir diziden beklendiği gibi olup olmadığını tespit etmektir. Testin başarılı olabilmesi için 20 000 bitlik bir dizideki '1' sayısının $9654 < n < 10346$ aralığında olması gerekir.

6.1.1.2. Poker Testi (Poker Test)

Bu testte, 'k' bit içeren dizi, $k \geq 5 \cdot 2^m$ olacak şekilde, üst üste çakışmayan m-bitlik parçalara ayrılır ve i. parça n_i diye adlandırılır. Rasgele bir diziden beklenen, tüm m-bitlik blokların k uzunluklu bir dizide aynı sayıda birbirini tekrar etmesidir. Testin başarılı olabilmesi için Eşitlik 6.1'deki formülle hesaplanan X değerinin, $k=20\ 000$ ve $m=4$ için, $1,030 < X < 57,40$ aralığında olması gerekir.

$$X = \frac{2^m}{k} \left(\sum_{i=1}^{2^m} n_i^2 \right) - k \quad (6.1)$$

6.1.1.3. Koşu Testi (Run Test)

Elimizdeki bit dizisinin bu testten başarıyla geçmesi için, dizide ardarda gelen '1' ve '0' lar dan oluşan çeşitli uzunluktaki blokların sayısının Çizelge 6.1'de belirtildiği gibi olması beklenir. 6 bitten daha uzun bloklar 6 bitlik olarak kabul edilmektedir.

Çizelge 6.1. Koşu testi için blok uzunluklarına göre blok sayıları.

Blok uzunluğu	Blok sayısı aralığı
1	2267-2733
2	1079-1421
3	502-748
4	223-402
5	90-223
6+	90-223

Örneğin üretilen bit dizisi $\epsilon=100100011010100100001100111001011001101011110$ olarak alındığında;

1. 1 uzunluklu “1” sayısı=7
2. 1 uzunluklu “0” sayısı=5
3. 2 uzunluklu “1” sayısı=4
4. 2 uzunluklu “0” sayısı=5
5. 3 uzunluklu “1” sayısı=1
6. 3 uzunluklu “0” sayısı=1
7. 4 uzunluklu “1” sayısı=1
8. 4 uzunluklu “0” sayısı=1
9. Diğer durumlar “yok” olarak koşu testi uygulanır

6.1.1.4. Uzun Koşu Testi (Long Runs Test)

Uzun koşu testinin başarılı olabilmesi için 20 000 bitlik bir bit dizisindeki blok (runs) uzunluklarının hepsinin 34’ten küçük olması gerekir. Bu şart sağlanmadığı takdirde bit dizisi testi başarılı sayılmamaktadır.

6.1.2. NIST 800-22 Testi

NIST 800-22 testi, FIPS 140-1 testine göre çok daha güçlü bir testtir. FIPS 140-1 testini geçen bir bit dizisi NIST 800-22 testinden başarısız olup sınıfta kalabilir. Bu nedenle ciddi uygulamalarda NIST 800-22 testi tercih edilmektedir. Bu test kendi içinde 16 tane ayrı testten oluşmaktadır. Teste tabi tutulan bit dizisinin başarılı olabilmesi için tüm testleri başarıyla geçmesi gerekmektedir [133,163]. Bu testler aşağıda kısa ve ayrıntılı açıklamalı olarak verilmiştir:

1. Frekans testi (Frequency test)
2. Bir Blok içinde Frekans Testi (Frequency Test with in a Blok)
3. Akış Testi (Runs Test)
4. Bir Blok içinde en Uzun Birler Akış Testi (Tests for the Longest-Run-of-Ones in a Block)
5. İkili Matris Derece Testi (The Binary Matrix Rank Test)
6. Ayırık Fourier Dönüşüm Testi (The Discrete Fourier Transform (FFT) Test)
7. Örtüşmeyen Şablon Eşleştirme Testi (The Non-overlapping Template Matching Test)
8. Örtüşen Şablon Eşleştirme Testi (The Overlapping Template Matching Test)
9. Maurer'in "Evrensel İstatistik" Testi (Maurer's "Universal Statistical" Test)
10. Lempel-Ziv sıkıştırma testi
11. Doğrusal Karmaşıklık Testi (The Linear Complexity Test)
12. Seri Testi (The Serial Test)
13. Yaklaşık Entropi Testi (The Approximate Entropy Test)
14. Kümülatif Topamlar Testi (The Cumulative Sums Test)
15. Rasgele Gezinimler Testi (The Random Excursions Test)
16. Rasgele Gezinimler Değişken Testi (The Random Excursions Variant Test)

NIST 800-22 test suitindeki 16 testin uygulanma sırası tamamen isteğe bağlıdır. Fakat frekans testinin, bir dizide rasgele olmayan bölgelerin varlığı ile ilgili temel ipuçları verdiğinden öncelikle uygulanması önerilmektedir. Eğer bu test başarısız sonuçlanırsa diğer testlerin de başarısız olma olasılığı yüksektir. Zaman kriteri açısından en karmaşık test doğrusal karmaşıklık testidir. Bu testte, test edilecek rasgele bit dizisinin bazı parametreleri dışarıdan belirlenebilmektedir. Bu testlerde en önemli parametrelerden birisi olan *P-değeri* teste tabi tutulan rasgele dizilerin rasgeleliğinin bir ölçütü olarak kabul edilmektedir. *P-değeri* gerçekten rasgele bir dizi için 1'e yakın, bunun tersi durumunda ise *P-değeri* 0'a yakın olmaktadır. Teste tabi tutulan bloklardan testi geçenlerin sayısı testi geçme oranını belirler ve bu oran teste sokulan blok sayısına göre değişir. *P-değeri* ise testi geçen blokların performansının bir ölçütüdür. Test programı, hesaplanan *P-değeri* ve testi geçme oranı değerlerini belirleyerek test edilen veri için başarılı veya başarısız şeklinde sonuç belirtir. Bunun yanı sıra kullanıcı, test geçme oranı ve *P-değeri*'ni aynı anda

yorumlayarak test edilen dizilerin başarısı hakkında yorum yapılabilir. Test paketindeki bazı testler, hesaplama aşamasında referans dağılım olarak Standart normal dağılım ve Ki-kare dağılımlarını kullanmaktadırlar [132,162].

Standart Normal Dağılımda, RSÜ'den elde edilen test istatistik değeri ile rasgelelik varsayımı altındaki istatistiğin beklenen değerinin karşılaştırılmasında kullanılan istatistiksel bir yöntemdir. Bu yöntem için x , örnek test istatistik değerleri ($-\infty < x < \infty$), μ beklenen değer ($-\infty < \mu < \infty$) ve σ^2 test istatistiğinin varyansı ($\sigma^2 > 0$) olmak üzere olasılık yoğunluk fonksiyonu Eşitlik 6.2'de verilmektedir.

$$y(z) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad (6.2)$$

Burada x rasgele değişkenine normal dağılıma sahip denilmekte ve $x \sim N(\mu, \sigma^2)$ şeklinde gösterilmektedir. Eğer burada $\sigma^2=1$ ve $\mu=0$ değerini alır ise bu dağılıma standart normal dağılım denilmektedir. Bu dağılımda rasgele değişken z olmak üzere Eşitlik 6.3 kullanılarak hesaplanmaktadır.

$$z = \frac{x - \mu}{\sigma} \quad (6.3)$$

Sonuç olarak rasgele değişkeninin olasılık yoğunluk fonksiyonu;

$$f(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2}, \quad -\infty < z < \infty \quad (6.4)$$

eşitliği kullanılarak hesaplanmaktadır.

Ki-kare (χ^2) Dağılımında rasgele olduğu hipotezi öne sürülen verilerin dağılımı ile beklenen verilerin dağılımını karşılaştırmak amacıyla kullanılan bilinen en iyi istatistiksel dağılım yöntemidir. Rasgele sayıların gözlemlenen frekansı N_i , beklenen frekans Ne_i ve $i=1 < k < \infty$ olmak üzere gözlemlenen χ^2 istatistiksel test dağılımı Eşitlik 6.5 ile hesaplanmaktadır.

$$\chi^2 = \sum_{i=1}^k \frac{(N_i - Ne_i)^2}{Ne_i} \quad (6.5)$$

Örneğin zar atma olayında ki-kare dağılımı aşağıdaki gibi hesaplanmaktadır [164].

Değer:	1	2	3	4	5	6
Gözlenen:	15	19	22	21	17	26

Değerlerin sayısı $n=120$, her bir atılan zar için beklenen olasılık $e_i = 1/6$, değerlerin beklenen frekans sayısı ($Ne_i = n \cdot e_i = 120 \cdot 1/6 = 20$) hesaplanmaktadır. Gözlemlenen istatistiksel referans dağılım değeri;

$$\chi^2 = \sum_{i=1}^{k=6} \frac{(N_i - Ne_i)^2}{Ne_i} = \frac{(15-20)^2}{20} + \frac{(19-20)^2}{20} + \frac{(22-20)^2}{20} + \frac{(21-20)^2}{20} + \frac{(17-20)^2}{20} + \frac{(26-20)^2}{20} = 3,8$$

olarak hesaplanmaktadır. Bu çalışmada kullanılan NIST 800-22 testlerinin uygulanma amacı, yöntemi ve tanımı kısaca aşağıda açıklanmıştır [133,158,163,165,166].

6.1.2.1. Frekans (Monobit) Testi

Bu testin amacı bit dizisindeki “1” ve “0” sayılarının yaklaşık olarak dengeli olup olmadığının ölçülmesidir. Tüm dizideki sıfırlarla birlerinin oranın gözlemlenmesi ilkesine dayanır. Diğer testlerin uygulanma ön koşulu bu testi geçme ilkesine dayanır. Çünkü bu testin sonucu, tüm dizinin rasgele olup olmadığı ile ilgili ciddi bilgi verir. Dizi uzunluğu en az $n=100$ bit olmalı ve $P < 0,01$ küçük olmamalıdır. Eşitlik 6.6’da kullanılan üretilen bit dizisi $\varepsilon = \varepsilon_1, \varepsilon_2, \dots, \varepsilon_n$, bit dizisinin uzunluğu n , Eşitlik 6.8’deki S_{obs} gözlemlenen değer ise Eşitlik 6.6’da ki X_i ’lerin mutlak değer toplamı ve Eşitlik 6.7’de ki S_n rasgele değişkenlerinin toplamını temsil etmektedir. Daha sonra S_{obs} değeri kullanılarak Eşitlik 6.9’da ki hata fonksiyonu(error function- $erf(z)$) hesaplanarak Eşitlik 6.10’da ki tamamlayıcı hata fonksiyonu(complementary error function- $erfc(z)$) hesaplanır.

$$X_i = 2\varepsilon_i - 1, \quad X_i = \{-1, +1\} \quad (6.6)$$

$$S_n = \sum_{i=1}^n X_i, \quad i = 1, \dots, n, \quad (6.7)$$

$$S_{obs} = \frac{|S_n|}{\sqrt{n}} \quad (6.8)$$

$$\operatorname{erf}(z) = \frac{2}{\sqrt{\pi}} \int_z^{\infty} e^{-u^2} du \quad (6.9)$$

$$\operatorname{erfc}(u) = 1 - \operatorname{erf}(u) \quad (6.10)$$

Frekans testi değeri (*P-değeri*) hesaplanabilmesi için Eşitlik 6.6, Eşitlik 6.7, Eşitlik 6.8, Eşitlik 6.9 ve Eşitlik 6.10 yardımıyla Eşitlik 6.11'deki değer hesaplanmaktadır.

$$P - value = \operatorname{erfc}\left(\frac{S_{obs}}{\sqrt{2}}\right) \quad (6.11)$$

Örneğin; üretilen bit dizisi $\varepsilon = 1011010101$ ise $n=10$ olur.

$$\varepsilon_1 = 1 \text{ için } \Rightarrow X_1 = 2 \cdot \varepsilon_1 - 1 = 2 \cdot 1 - 1 = 1$$

$$\varepsilon_2 = 0 \text{ için } \Rightarrow X_2 = 2 \cdot \varepsilon_2 - 1 = 2 \cdot 0 - 1 = -1$$

$$S_n = 1 + (-1) + 1 + 1 + (-1) + 1 + (-1) + 1 + (-1) + 1 = 2$$

$$S_{obs} = \frac{|S_n|}{\sqrt{n}} = \frac{|2|}{\sqrt{10}} = 0,632455532$$

erfc tamamlayıcı hata fonksiyonu Eşitlik 6.10 ile hesaplandıktan sonra;

$P - \text{değeri} = \text{erfc}\left(\frac{S_{obs}}{\sqrt{2}}\right) = 0,527089 \geq 0,01$ olduğundan üretilen bit dizisinin rasgele olduğuna karar verilir.

Diğer bir örnekte üretilen bit dizisi ε minimum 100 bit olarak alınırsa ($n \geq 100$);

$\varepsilon = 11001001000011111101101010100010001000010110100011$
 $00001000110100110001001100011001100010100010111000,$

$n=100, S_{100} = -16, S_{obs} = 1,60$ ise $P - \text{değeri} = 0,109599 \geq 0,01$ olduğundan dizi rasgeledir.

6.1.2.2. Blok Frekans Testi

M -bit bloklar içindeki “1” lerin oranını test etmeye odaklanmıştır. Testin asıl amacı iyi bir rasgele varsayımdan beklenen M -bit bloklar içindeki “1” lerin frekansının $M/2$ olup olmadığının tespit edilmesidir. Blok uzunluğu $M=1$ için yapılan test Frekans (Monobit) testine dönüşür. Test istatistiklerinin referans dağılımı için ki-kare (χ^2) dağılımı kullanılmakta ve testin geçerli sonuçlar üretebilmesi için veri bit uzunluğu en az $n=100$ olmalıdır.

Hesaplama blok uzunluğu M , bit uzunluğu n , üretilen bit dizisi ε , referans dağılım istatistiği $\chi^2(obs)$ ve $N=n/M$ ($1 \leq i \leq N$) olmak üzere $P - \text{değeri}$ Eşitlik 6.12, Eşitlik 6.13 ve Eşitlik 6.14 kullanılarak hesaplanır.

$$\pi_i = \frac{\sum_{j=1}^M \varepsilon_{(i-1)M+j}}{M} \quad (6.12)$$

$$\chi^2(obs) = 4M \sum_{i=1}^N (\pi_i - 1/2)^2 \quad (6.13)$$

$$P - \text{değeri} = \text{igamc}\left(\frac{N}{2}, \frac{\chi^2(obs)}{2}\right) \quad (6.14)$$

Eşitlik 6.14'te kullanılan *igamc* fonksiyonu bir yaklaşım formülü olarak (a, x) değişkenlerine bağımlı tamamlanmamış gama fonksiyonu olarak Eşitlik 6.15'te tanımlanmaktadır. Burada *igamc*, $Q(a, 0)=1$ ve $Q(a, \infty)=0$ olmak üzere hesaplanır.

$$Q(a, x) \equiv \frac{\Gamma(a, x)}{\Gamma(a)} \equiv \frac{1}{\Gamma(a)} \int_x^{\infty} e^{-t} t^{a-1} dt \quad (6.15)$$

Örneğin üretilen bit dizisi $(\varepsilon = \varepsilon_1, \varepsilon_2, \dots, \varepsilon_n)$;

$$\varepsilon = 11001001000011111101101010100010001000010110100011 \quad n=100, M=10$$

$$00001000110100110001001100011001100010100010111000,$$

ise $N = \frac{n}{M} = \frac{100}{10} = 10$ ve referans dağılım istatistiği $\chi^2(obs) = 7,20$ olarak bulunur.

Hesaplamalar sonucunda alınan değerler Eşitlik 6.14'te yerine yazılarak $P - değeri = 0,706438$ hesaplanmaktadır.

Sonuç olarak $P - değeri = 0,706438 \geq 0,01$ olduğundan üretilen bit dizileri rasgele olarak kabul edilir.

6.1.2.3. Akış Testi

Testin amacı dizi içindeki “0” ve “1” lerin dengesini ve aralarındaki osilasyonun çok hızlı veya çok yavaş olup olmadığı hakkında önemli bilgiler verir. Bu test için istatistiklerinin referans dağılımında ki-kare (χ^2) dağılımı kullanılmaktadır. Testin hesaplanma aşamasında kullanılan terimlerden üretilen bit dizisi ε , bit dizisinin uzunluğu n , π bit dizisindeki “1” değerlerinin sayısını ve akışkanların toplam tekrar sayısını (bit osilasyon sayısı) $V_n(obs)$ ile tanımlanmıştır. Bu testin gerçekleştirebilmesi için $\tau = 2/\sqrt{n}$ olmak üzere Eşitlik 6.16 şartının sağlanması gerekir.

$$\left| \pi - \frac{1}{2} \right| \geq \tau \quad (6.16)$$

$$V_n(obs) = \sum_{k=1}^{n-1} r(k) + 1 \quad (6.17)$$

Eğer $\varepsilon_k = \varepsilon_{k+1}$ ise $r(k)=0$ ve diğer durumda $r(k)=1$ olarak kabul edilerek akışkanların toplam tekrar sayısı Eşitlik 6.17’de hesaplandıktan sonra Eşitlik 6.18’de yerine konularak bu test sonucu olan *P-değeri* hesaplanmaktadır.

$$P-değeri = \operatorname{erfc} \left(\frac{|V_n - 2n\pi(1-\pi)|}{2\sqrt{2n\pi(1-\pi)}} \right) \quad (6.18)$$

Örneğin üretilen bit dizisi ($\varepsilon = \varepsilon_1, \varepsilon_2, \dots, \varepsilon_n$) ve $n=100$ ise;

$\varepsilon = 11001001000011111101101010100010001000010110100011$
 $00001000110100110001001100011001100010100010111000'$

$\tau = 0,020$, $\pi = 0,42$ ve $V_n(obs) = 52$ olarak hesaplanır. Alınan bu sonuçlar Eşitlik 6.18’de yerine yazılarak $P-değeri = 0,500798$ hesaplanmaktadır. Sonuç olarak $P-değeri = 0,500798 \geq 0,01$ olduğundan üretilen bit dizileri rasgele olarak kabul edilir.

6.1.2.4. Blok içindeki Birlerin En Uzun Akış Testi

Bu test, M -bitlik bir blok içindeki en uzun “1” lerin akışına odaklanmıştır. Amaç olarak, bir dizideki “1” lerden oluşan en uzun yineleme uzunluğunun, rasgele olduğu beklenen bir dizideki değerlerle tutarlı olup olmadığının incelenmesidir. Birlerdeki tutarlı uzunluk, aynı oranda sıfırlarında beklenen uzunlukta olduğunu gösterir. Alınan sonuçların frekansları ile beklenen değer frekansları ile karşılaştırılmak suretiyle sapma olup olmadığına bakılır. Bu test için istatistiklerinin referans dağılımında ki-kare (χ^2) dağılımı kullanılmaktadır. Hesaplamalarda blok uzunluğu M , bit uzunluğu n , üretilen bit dizisi ε , referans dağılım χ^2 ve N örtüşmeyen blok sayısını ifade etmektedir. Testte önerilen n ve M sayıları Çizelge 6.2’de verilmiştir.

Çizelge 6.2. Önerilen minimum dizi ve blok uzunlukları.

Minimum n	M
128	8
6272	128
750 000	10^4

Önerilen her bir blok için en uzun birlerin beklenen akış frekans değerleri V_i Çizelge 6.3'te verilmiştir.

Çizelge 6.3. Her bir M -bit blok için en uzun birlerin beklenen akış frekans değerleri.

V_i	Önerilen M blok uzunluğu		
	8	128	10^4
V_0	≤ 1	≤ 4	≤ 10
V_1	2	5	11
V_2	3	6	12
V_3	≥ 4	7	13
V_4	---	8	14
V_5	---	≥ 9	15
V_6	---	---	≥ 16

M blok uzunlukları değeri tarafından belirlenen K ve N değerleri Çizelge 6.4'te verilmiştir. Burada verilen K ve N değerlerine göre istatistiklerinin referans dağılımı ki-kare (χ^2) Eşitlik 6.19'da ki gibi hesaplanmaktadır.

Çizelge 6.4. M değerine göre belirlenen K ve N parametre değerleri.

M	K	N
8	3	16
128	5	49
10^4	6	75

$$\chi^2(obs) = \sum_{i=0}^K \frac{(v_i - N\pi_i)^2}{N\pi_i} \quad (6.19)$$

Eşitlik 6.19'da hesaplanan $\chi^2(obs)$ değeri ve Çizelge 6.4'te verilen K değeri Eşitlik 6.20'de yerine konularak P -değeri hesaplanır.

$$P - \text{değeri} = \text{igamc} \left(\frac{K}{2}, \frac{\chi^2(obs)}{2} \right) \quad (6.20)$$

Örneğin $K=3$ ve $M=8$ üretilen bit dizisi ($\varepsilon = \varepsilon_1, \varepsilon_2, \dots, \varepsilon_n$);

$\varepsilon = 11001100000101010110110001001100111000000000001001001101010100010$
 $001001111010110100000001101011111001100111001101101100010110010$

$n=128$, $\tau = \frac{2}{\sqrt{128}} = 0,020$ ve $\pi_0 = 0,2148$, $\pi_1 = 0,3672$, $\pi_2 = 0,2305$, $\pi_3 = 0,1875$

değerleri hesaplanmaktadır. Bulunan bu değerler Eşitlik 6.19'da yerine konulursa;

$$\chi^2 = \frac{(4-16 \cdot 0,214)^2}{16 \cdot 0,214} + \frac{(9-16 \cdot 0,367)^2}{16 \cdot 0,367} + \frac{(3-16 \cdot 0,23)^2}{16 \cdot 0,23} + \frac{(0-16 \cdot 0,187)^2}{16 \cdot 0,187} = 4,882$$

olarak hesaplanmaktadır. Çizelge 6.4'ten faydalanılarak $M=8$, $N = 128/8 = 16$ ve $K=3$ değeri bulunmaktadır. $n=128$ bitlik bir dizi $M=8$ bitlik bloklara ayrılarak tüm bloklardaki en uzun akış gösteren birlerin frekansı Çizelge 6.5'te verilmiştir.

Çizelge 6.5. Alt bloklara ayrılan en uzun birlerin akış frekanslarının incelenmesi.

Blok Adresi	Alt Bloklar	Maksimum Akış
1	11001100	2
2	00010101	1
3	01101100	2
4	01001100	2
5	11100000	3
6	00000010	1
7	01001101	2
8	01010001	1
9	00010011	2
10	11010110	2
11	10000000	1
12	11010111	3
13	11001100	2
14	11100110	3
15	11011000	2
16	10110010	2

Ayrıca Çizelge 6.4'ten alınan $K=3$ değerine göre V_i () değişkeni $0 \leq i \leq 3$ aralığında toplam dört adet değişken değeri alacaktır. V_0 bloklardaki ardışık bir tane gelen en uzun birlerin akış sayısını, V_1 bloklardaki ardışık gelen iki tane birlerin akış sayısını,

V2 ise üç tane gelen uzun birlerin akış sayısını ve V3 ise dört tane gelen birlerin akışını göstermektedir. Alınan bütün sonuçlar Eşitlik 6.20’de yerine konulursa P-değeri hesaplanmaktadır.

$$P-değeri = igamc\left(\frac{K}{2}, \frac{\chi^2(obs)}{2}\right) = igamc\left(\frac{3}{2}, \frac{4.882}{2}\right) = 0.180$$

Sonuç olarak $P-değeri=0,180 \geq 0,01$ olduğundan üretilen bit dizileri rasgele olarak kabul edilir.

6.1.2.5. İkili Matris Rankı Testi

Bu test, tüm dizinin ayrışık alt matrislerinin ranklarına odaklı bir testtir. Testin amacı ise orijinal dizinin sabit uzunluktaki alt dizileri arasında doğrusal bir bağıntının varlığını araştırmaktır. Bu test ayrıca DIEHARD test suiteinde de yer almaktadır. Test içerisinde n bit uzunluktaki dizi $M \times M$ -bitlik matrisler halinde parçalanır ve oluşan her bir matrisin rankı hesaplanıp frekans değerleri belirlenir. Hesaplanan frekans değerleri beklenen frekansla karşılaştırılıp sapma olup olmadığı kontrol edilir. Buda bize üretilen n bit dizisi hakkında istatistiksel olarak rasgelelik hakkında bilgi verir.

Bu test için her bir matrisin satır sayısı M , her bir matrisin sütun sayısı Q olmak üzere $M=Q=32$ olarak alınmıştır. Bu test için istatistiklerinin referans dağılımında ki-kare (χ^2) dağılımı kullanılmaktadır. Üretilen bit dizisi $M \times Q$ bit ayrık matrislere bölünerek oluşan matris sayısı $N = \lfloor n / MQ \rfloor$ şeklinde hesaplanmaktadır. Oluşan bu matrislerden sonra geriye kalan bitler ihmal edilir.

Örneğin üretilen bit dizisi $\varepsilon = 01011001001010101101$, $n=20$, $M=Q=3$ ve oluşturulacak matris sayısı $N = \lfloor n / MQ \rfloor = 2$ olarak bulunur. Matris oluşturmada kullanılacak bit sayısı $M \times Q \times N = 3 \times 3 \times 2 = 18$ bit olarak hesaplanır ve geriye kalan $20 - 18 = 2$ en son seviyeli iki bit (0 ve 1) ihmal edilir. İlk 3 bit ilk satırda, sonraki 3 bit ikinci satırda ve diğer 3 bitte üçüncü satıra gelecek şekilde matris oluşturulur. Aynı yöntemle 2.matriste oluşturulur. Oluşan matrisler:

$$N_1 = \begin{vmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{vmatrix} \quad \text{ve} \quad N_2 = \begin{vmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{vmatrix} \quad \text{olmaktadır. Alınan sonuçlar kullanılarak } P_m$$

değerleri şu şekilde hesaplanmaktadır.

$$P_m = \prod_{j=1}^{\infty} \left[1 - \frac{1}{2^j} \right] = 0.2888$$

$$P_{m-1} \approx 2 \cdot P_m = 2 \cdot 0.2888 = 0.5776$$

$$P_{m-2} \approx \frac{4 \cdot P_m}{9} = \frac{4 \cdot 0.2888}{9} = 0.1284$$

Referans dağılımı Ki-kare değerinin hesaplanabilmesi için Eşitlik 6.21 kullanılmaktadır.

$$\chi^2(obs) = \frac{(F_M - P_m \cdot N)^2}{P_m \cdot N} + \frac{(F_{M-1} - P_{m-1} \cdot N)^2}{P_{m-1} \cdot N} + \frac{(N - F_M - F_{M-1} - P_{m-2} \cdot N)^2}{P_{m-2} \cdot N} \quad (6.21)$$

Bulunan değerler Eşitlik 6.21'de yerine yazıldığında $\chi^2(obs)=0,597$ olarak hesaplanmaktadır. Elde edilen veriler ışığında P -değeri aşağıdaki gibi hesaplanmaktadır.

$$P - \text{değeri} = e^{-\chi^2(obs)/2} = igamc\left(1, \frac{\chi^2(obs)}{2}\right) = 0,741948$$

Sonuç olarak $P\text{-değeri}=0,741948 \geq 0,01$ olduğundan üretilen bit dizileri rasgele olarak kabul edilir.

Farklı bir örnekte üretilen bit dizisi $\varepsilon = \varepsilon_1, \varepsilon_2, \dots, \varepsilon_{100000}$ ise $n=100.000$, $M=Q=32$

ve oluşacak $M \times Q=32 \times 32$ boyutlu matris sayısı $N = \left\lfloor \frac{n}{MQ} \right\rfloor = \left\lfloor \frac{100000}{32 \cdot 32} \right\rfloor = 97$ olarak

hesaplanmaktadır. Kullanılacak bit sayısı $M*Q*N=32*32*97=99.328$ adet olup en son seviyedeki atılan bit sayısı $100.000-99.328=672$ tanedir. Hesaplanan diğer değerler $F_M=23$, $F_{M-1}=60$, $N-F_M-F_{M-1}=14$ Eşitlik 6.21'de yerine konularak referans dağılım değeri $\chi^2=1.2619656$ hesaplanır. Tüm bu elde edilen sonuçlardan sonra $P\text{-değeri}=0,532069$ olarak bulunur ve bu değer $0,532069 \geq 0,01$ olduğundan üretilen $n=100\ 000$ adet bit dizisi rasgele kabul edilir.

6.1.2.6. Ayırık Fourier Dönüşüm Testi

Bu test bir dizinin ayırık Fourier dönüşümünün tepe noktalarına odaklıdır ve Spektral yöntem olarakta adlandırılır. Bu testin amacı test edilen dizinin içindeki rasgelelik varsayımından sapma gösteren periyodik özelliklerin tespit edilmesidir. Tepe değerinin %95 i aştığı durumlar rasgelelik için iyi bir ölçüt anlamına gelmektedir. Testin hesaplamasında kullanılan bit dizisi ε , bit uzunluğu n , %95 eşik değerinin üstündeki frekans bileşenlerinin gözlemlenen ve beklenen sayıları arasındaki standart farklılığı d olmak üzere $x_i=2\varepsilon_i-1$ hesaplanır. Bu test için istatistiklerinin referans dağılımında standart normal dağılım yöntemi kullanılmaktadır. Burada bit dizisindeki 0 ve 1'ler için -1 ve +1 dönüşümü yapılarak $n=10$, $\varepsilon=1001010011$ için dönüşüm $X=1,-1,-1,1,-1,1,-1,1,1$ olmaktadır. Dönüşüm yapılan bit dizisi X 'e ayırık Fourier dönüşümü (AFD) uygulanarak Eşitlik 6.22'deki $S=AFD(X)$ elde edilir.

$$S_j = \sum_{k=1}^n X_k e^{(2\pi_i(k-1)j/n)} \quad (6.22)$$

Burada, $e^{(2\pi_i(k-1)j/n)} = \cos(2\pi_i k j / n) + i \sin(2\pi_i k j / n)$, $j=0, 1, \dots, n-1$ ve $i=\sqrt{-1}$ olarak alınmaktadır. Sonraki adımda $M = \text{modulus}(S_j) \equiv |S_j|$ burada S' , S dizisindeki ilk $n/2$ elemandan oluşan bir alt diziyi iade etmekte ve modulus fonksiyonu tepe yüksekliklerinin bir dizisini üretir. $T = \sqrt{3n} = \%95$ değerini aşmamalıdır. $N_0=0,95*n/2$ değeri ise T 'den daha küçük tepe yüksekliklerinin beklenen teorik sayıdır. Bu örnek için $N_0=4.75$ olarak bulunur ve T den daha az gerçek gözlemlenen tepe sayısı $N_1=4$ olarak bulunur. Test istatistik değeri P 'nin hesaplanabilmesi için;

$$d = \frac{N_1 - N_0}{\sqrt{\frac{n(0.95)(0.05)}{4}}} = \frac{(4 - 4.75)}{\sqrt{\frac{10(0.95)(0.05)}{10}}} = -2.176429$$

$$P\text{-değeri} = \operatorname{erfc}\left(\frac{|d|}{\sqrt{2}}\right) = \left(\frac{|-2.176429|}{\sqrt{2}}\right) = 0.029523$$

olarak hesaplanmaktadır. Sonuç olarak $P\text{-değeri}=0,029523 \geq 0,01$ olduğundan üretilen bit dizileri rasgele olarak kabul edilir.

Diğer bir örnekte üretilen bit dizisi ($\varepsilon = \varepsilon_1, \varepsilon_2, \dots, \varepsilon_n$);

$$\varepsilon = 11001001000011111101101010100010001000010110100011 \quad n=100, \quad N_1=46, \\ 00001000110100110001001100011001100010100010111000,$$

$N_0=47.5$, $d=-1.376494$ ve hesaplanan test istatistik değeri $P\text{-değeri}$:

$$P\text{-değeri} = \operatorname{erfc}\left(\frac{|d|}{\sqrt{2}}\right) = 0,168669 \geq 0,01 \text{ olduğundan üretilen dizi rasgele kabul edilir.}$$

6.1.2.7. Örtüşmeyen Şablon Eşleştirme Testi

Bu test n bitlik bir dizideki m -bitlik bloklar içerisinde periyodik olmayan (aperiyodik) önceden belirlenmiş (pre-specified) hedef dizisinin bulunma sıklığının gözlenmesine dayanır. Bu ve bundan sonraki Örtüşen Şablon Eşleştirme testlerinde özel m -bitlik bir örneği aramak için m -bitlik bir pencere kullanılmaktadır. Eğer bu örnek bulunamaz ise pencere bir bit kaydırılarak tarama işlemine devam edilir. Eğer örnek bulunur ise, pencere bulunan örnekten sonraki ilk bite yeniden konumlanır ve arama işlemlerine devam eder. Testin hesaplama aşamasında formüllerde kullanılan terimlerden üretilen bit dizisi ε ($\varepsilon = \varepsilon_1, \varepsilon_2, \dots, \varepsilon_n$), bit dizisi uzunluğu n , her bir şablonun bit uzunluğu m , istatistiksel testin içerisinde bulunan aperiodyik şablon örnekleri B , bağımsız blok sayısı N ve dizi içerisinde test edilecek alt dizilerin bit

uzunluğu M ile tanımlanmaktadır. Bu test için istatistiklerinin referans dağılımında ki-kare (χ^2) dağılımı kullanılmaktadır.

Örneğin üretilen bit dizisi $\varepsilon=10100100101110010110$, bit dizi uzunluğu $n=20$, Eğer $N=2$ ve $M=n/N=20/2=10$ olur ve $M_1=1010010010$ ile $M_2=11100010110$ olmak üzere iki blok elde edilir. $j=1, 2, \dots, N$, olmak üzere W_j , j blok içinde seçilen B şablonunun kaç kez bulunduğu sayısıdır. Burada eğer $m=3$ ve $B=001$ olarak seçilen şablonun M_1 ve M_2 blokları içinde kaç kez bulunduğu Çizelge 6.6'da gösterilmektedir. Amaç üretilen ε dizisi içinde M_1 ve M_2 blokları içerisinde seçilen B şablonunun kaç tane olduğunun bulunmasıdır. Bu örnekte M_1 bloğu içerisinde üç bitlik B bloğu araştırıldığında bit pozisyonları 1-3 için "101", 2-4 için "010", 3-5 için "100" olduğundan $W_1=0$ olmaktadır. Fakat bit pozisyonu 4-6 için "001" B şablonu ile eşit olduğundan $W_1=1$ olur. Daha sonra şablon üç bit olduğundan uygun bitlerden kurtulana kadar 5-7 ve 6-8 bit pozisyonları atlanarak diğer pozisyonlara bakılmaya devam edilir. 7-9 pozisyonu için "001" B şablonu ile tekrar benzer olduğundan $W_1=2$ değerini alır. Son olarak şablon üç bit olduğu için 8-10 pozisyonu atlanarak M_1 bloğu için işlem tamamlanarak $W_1=2$ değerini almaktadır. Tüm bu aşamalar seçilen B şablonu için bir sonraki blok M_2 olmak üzere bütün bloklara sırasıyla uygulanmaktadır. Bu aşamalar M_1 ve M_2 blokları için Çizelge 6.6'da tüm bit pozisyonları için incelenmiştir.

Çizelge 6.6. $m=3$ bitlik $B=001$ şablonunun M_1 ve M_2 bloklarının içinde incelenmesi.

Bit Pozisyonları	B=001			
	M_1 1010010010		M_2 11100010110	
	Bitler	W_1	Bitler	W_2
1-3	101	0	111	0
2-4	010	0	110	0
3-5	100	0	100	0
4-6	001 (B)	1	001 (B)	1
5-7	Test edilmedi		Test edilmedi	
6-8	Test edilmedi		Test edilmedi	
7-9	001 (B)	2	011	1
8-10	Test edilmedi		110	1

Çizelge 6.6’da gösterilen üretilen bit dizisinden parçalanmış M1 ve M2 blokları içerisinde yapılan incelemeye göre $W_1=2$ ve $W_2=1$ olarak elde edilmektedir. Rasgelelik testi için ortalama değeri (μ), varyans (σ^2) ve ki-kare (χ^2) değerleri hesaplanır.

$$\mu = \frac{(M - m + 1)}{2^m} = \frac{(10 - 3 + 1)}{2^3} = 1$$

$$\sigma^2 = M \left(\frac{1}{2^m} - \frac{2m-1}{2^{2m}} \right) = 10 \left(\frac{1}{2^3} - \frac{2 \cdot 3 - 1}{2^{2 \cdot 3}} \right) = 0.46875$$

$$\chi^2(obs) = \sum_{j=1}^N \frac{(W_j - \mu)^2}{\sigma^2} = \frac{(2-1)^2 + (1-1)^2}{0.46875} = 2.13333$$

Bulunan değerlerden sonra P-değeri aşağıdaki gibi hesaplanmaktadır.

$$P\text{-değeri} = igamc \left(\frac{N}{2}, \frac{\chi^2(obs)}{2} \right) = igamc \left(\frac{2}{2}, \frac{2.1333}{2} \right) = 0.344154$$

Sonuç olarak $P\text{-değeri} = 0.344154 \geq 0.01$ olduğundan üretilen bit dizileri rasgele kabul edilir. Bu testin başarılı sonuçlar verebilmesi için m değeri 2,3,...,10 arasında alınabilir. Fakat tavsiye edilen $m=9$ ve $m=10$ değerlerinde daha başarılı sonuçlar alınmıştır. N değeri ise P-değerinin geçerli olabilmesi için $N \leq 100$ olarak seçilmelidir.

Ayrıca kesinlikle $M > 0.01 \cdot n$ ve $N = \left\lfloor \frac{n}{M} \right\rfloor$ olmalıdır.

Diğer bir örnekte üretilen bit dizisi $\varepsilon=2^{20}$ ve üretilen bit sayısı $n=2^{20}$ olarak alınmaktadır. Bu test için $N=8$, $B=000000001$ ve $m=9$ olarak seçildiğinde $\mu=255.984375$, $\sigma^2=247.4999$, $W_1=259$, $W_2=229$, $W_3=271$, $W_4=245$, $W_5=272$, $W_6=262$, $W_7=259$, $W_8=246$, $\chi^2(obs)=5.999377$ ve $P\text{-değeri}=0.647302$ olarak hesaplanmaktadır. Sonuç olarak hesaplanan $P\text{-değeri} \geq 0.01$ olduğundan üretilen dizi rasgele kabul edilir.

6.1.2.8. Örtüşen Şablon Eşleştirme Testi

Bu test önceden belirlenmiş (pre-specified) hedef dizisinin n bitlik bir dizideki m -bitlik bloklar içerisinde periyodik olmayan (aperiyodik) örnek dizinin bulunma sıklığının gözlenmesine dayanır. Bu testin bir önceki Örtüşmeyen Şablon Eşleştirme Testinden tek farkı eğer hedef şablon bulunursa pencere bulunan şablondan sonraki ilk bit yerine sadece o an bulunan pozisyondan bir sonraki bite yeniden konumlanarak taramaya devam edilir. Bir önceki testteki hesaplamalarda kullanılan terimlerden farklı olarak bağımsızlık derecesi K ($K=5$ olarak alınmalı), test edilecek alt dizilerinin bit uzunluğu M ($M=1032$ alınmalı) ve n 'in içindeki bağımsız blok sayısı N ($N=968$ alınmalı) ile ifade edilmektedir. Bu test için istatistiklerinin referans dağılımında ki-kare (χ^2) dağılımı kullanılmaktadır. Örneğin bu test için üretilen bit dizisi; $\varepsilon = 10111011110010110100011100101110111110000101101001$, $n=50$, $K=2$, $M=10$ ve $N=5$ olarak alınırsa bit dizisi 5 ayrı bloğa bölünerek $M_1=1011101111$, $M_2=0010110100$, $M_3=0111001011$, $M_4=1011111000$ ve $M_5=0101101001$ blokları oluşturulur. Daha sonra her bir N bloktaki seçilen B örnek şablonunun bulunma sayısı Çizelge 6.7'de gösterildiği gibi bulunur. $m=2$ için $B=11$ olarak alındığında M_1 bloğu içinde B şablonunun varlığı hesaplanır.

Çizelge 6.7. Önerilen $B=11$ şablonunun M_1 bloğu içinde bulunma durumu.

Bit Pozisyonu	M ₁ Bloğu M ₁ =1011101111	
	Bitler	B=11 bulunma sayısı (V _i)
1-2	10	0
2-3	01	0
3-4	11 (B)	1
4-5	11 (B)	2
5-6	10	2
6-7	01	2
7-8	11 (B)	3
8-9	11 (B)	4
9-10	11 (B)	5

Bu işlemden sonra λ değeri $\lambda=(M-m+1)/2^m=(10-2+1)/2^2=2.25$ ve η değeri $\eta=\lambda/2=2.25/2=1.125$ olarak hesaplanır. Daha sonra $\pi_1=0,3246$, $\pi_2=0,1826$, $\pi_3=0,1426$, $\pi_4=0,1066$, $\pi_5=0,077$ ve $\pi_6=0,1662$ olasılık değerlerine göre $\chi^2(obs)$ ki-kare dağılımı hesaplanır.

$$\chi^2(obs) = \sum_{i=1}^N \frac{(v_i - N\pi_i)^2}{N\pi_i} = \frac{(0-5 \cdot 0.3246)^2}{5 \cdot 0.3246} + \frac{(1-5 \cdot 0.1826)^2}{5 \cdot 0.1826} + \frac{(1-5 \cdot 0.1426)^2}{5 \cdot 0.1426} + \frac{(1-5 \cdot 0.1066)^2}{5 \cdot 0.1066} + \frac{(1-5 \cdot 0.0771)^2}{5 \cdot 0.0771} + \frac{(1-5 \cdot 0.1662)^2}{5 \cdot 0.1662} = 3.167729$$

Elde edilen ki-kare değeri ve diğer alınan sonuçlar kullanılarak bu testin istatistiksel değeri;

$$P-değeri = igamc\left(\frac{N}{2}, \frac{\chi^2(obs)}{2}\right) = igamc\left(\frac{5}{2}, \frac{3.1667}{2}\right) = 0.274932$$

olarak hesaplanmaktadır. Sonuç olarak $P-değeri=0,274932 \geq 0,01$ olduğundan üretilen bit dizileri rasgele kabul edilir.

Diğer bir başka örnekte rasgele üretici tarafından üretilen bit dizisi $\varepsilon=10^6$, $n=1000000$, $B=111111111$, $m=9$, $K=5$ olarak alındığında $\chi^2(obs)=8.965859$ ve $P-değeri=0,110434$ olarak hesaplanmaktadır. Sonuç olarak hesaplanan $P-değeri \geq 0,01$ olduğundan üretilen dizi rasgele kabul edilir.

6.1.2.9. Maurer “Evrensel İstatistik” Testi

Princeton Üniversitesinde bilgisayar bilimcisi Maurer tarafından 1992 yılında geliştirilen bu test bir dizide eşleşen şablonların bit dizilerinin gözlenmesine ve rasgele dizinin veri kaybı olmadan sıkıştırılıp sıkıştırılamayacağını tespit edilmesine odaklanmıştır. Anlamlı bir şekilde sıkıştırılabilen bir dizinin rasgele olduğu kabul edilemez. Ayrıca önerilen bu test istatistiğinin kriptografik uygulamalarda gizli anahtar kaynağı için güvenli kalite ölçütü olduğu belirtilmiştir. Hesaplamalarda kullanılan her bir bloğun uzunluğu L , başlangıç dizisindeki blok sayısı Q , üretilen bit dizisi ε , üretilen bit dizi uzunluğu n , L -bit örnekleri arasındaki bitlerin sayılarının toplamı f_n olmak üzere $K=[n/L]-Q$ test bölümünü ifade etmektedir.

Örneğin rasgele sayı üretici tarafından üretilen bit dizisi $\varepsilon=01011010011101010111$ olursa $n=20$ ve $L=2$ ile $Q=4$ alındığında $K=[20/2]-4=6$ sonucu elde edilir. Başlangıç

segmenti $Q=01011010$ ve test segmenti $K=011101010111$ olur. Maurer testine göre başlangıç ve test bölümlerindeki L-bit uzunluğundaki blokların değerleri Çizelge 6.8’de gösterilmiştir.

Çizelge 6.8. L-bit uzunluğundaki blokların Maurer testine göre gösterimi.

Blok	Blok Tipi	İçerik
1	Başlangıç Segmenti Q	01
2		01
3		10
4		10
5	Test Segmenti K	01
6		11
7		01
8		01
9		01
10		11

Başlangıç segmenti kullanılarak olası her bir L-bit değerleri Çizelge 6.9’da verilmektedir. j burada i. L-bit bloğunun onluk sayı sistemindeki değeri olmak üzere $i=1$ den başlayıp Q değerine kadar $T_j=i$ olarak her bir L-bit bloğunun en son blok numarası değeri çizelgeye kaydedilir.

Çizelge 6.9. Dört başlangıç değeri olası L-bit değerleri.

	Olası L-bit değerleri			
	00 T_0 ’da kayıtlı	01 T_1 ’da kayıtlı	10 T_2 ’da kayıtlı	11 T_3 ’da kayıtlı
Başlangıç	0	2	4	0

Başlangıç bölümünün Çizelge 6.9’de ilk satırdaki aldığı değerler kullanılarak test bölümündeki diğer L-bit bloklarının aldığı değerler Çizelge 6.10’da verilmektedir. Eğer test bloğunun her bir L-bit değerleri başlangıç bölümüyle eşleştiğinde test bloğu tekrar blok numarası değerini almaktadır. Örneğin 5 nolu tekrar adımı, test bölümünde “01” değeri bulunmakta ve bu değer T_2 kayıtlı olduğundan bu bloğun yeni değeri 5 olmaktadır. 6 nolu satırdaki test bölümünde “11” değeri bulunmakta olup bu değerde olası L-bit değeri T_4 ’te kayıtlı olduğundan buraya tekrar bloğunun değeri olan 6 değeri atanmaktadır. Bu durum Çizelge 6.10’de gösterilmektedir.

Çizelge 6.10. Tekrar bloğu ve test bölümü için L-bit değerleri.

Tekrar Bloğu	Olası L-bit Değerleri			
	00	01	10	11
4	0	2	4	0
5	0	5	4	0
6	0	5	4	6
7	0	7	4	6
8	0	8	4	6
9	0	9	4	6
10	0	9	4	10

Tüm bu aşamalardan sonra test istatistiği Eşitlik 6.23 ve Eşitlik 6.24 kullanılarak hesaplanmaktadır.

$$f_n = \frac{1}{K} \sum_{i=Q+1}^{Q+K} \log_2(i - T_j) = \frac{1}{6} \sum_{i=4+1}^{10} \left[\log(5-2) + \log(6-0) + \log(7-5) + \log(8-7) + \log(9-8) + \log(10-6) \right] \quad (6.23)$$

Bu örnek için $f_n = 1,1949875$ değeri hesaplanır.

$$P - \text{değeri} = \text{erfc} \left(\left\| \frac{f_n - V_{\text{exp}}(L)(\text{beklenen değer})}{\sqrt{2} \sigma} \right\| \right) \quad (6.24)$$

Burada $\sigma = c \sqrt{\frac{\text{var}(L)}{K}}$ ve $c = 0,70 - \frac{0,80}{L} + (4 + \frac{32}{L}) \frac{K^{-3/L}}{15}$ olarak hesaplanmaktadır.

L'nin beklenen ($V_{\text{exp}}(L)$) ve varyans ($\text{var}(L)$) değerleri Çizelge 6.11'de verilmektedir. Alınan bütün değerler bu testin istatistik değerini hesaplamak için Eşitlik 6.24'te yerine konulursa P -değeri;

$$P - \text{değeri} = \text{erfc} \left(\left\| \frac{f_n - V_{\text{exp}}(L)}{\sqrt{2} \sigma} \right\| \right) = \text{erfc} \left(\left\| \frac{1,19498 - 1,53743}{\sqrt{2} \cdot \sqrt{1,338}} \right\| \right) = 0,767189$$

olarak hesaplanmaktadır. Sonuç olarak $P\text{-değeri} = 0,767189 \geq 0,01$ olduğundan üretilen bit dizileri rasgele kabul edilir.

Çizelge 6.11. L değerleri için beklenen ve varyans değerleri.

L	Beklenen Değer $V_{exp}(L)$	Varyans değeri $var(L)$
6	5,2177052	2,954
7	6,1962507	3,125
8	7,1836656	3,328
9	8,1766424	3,311
10	9,1723243	3,356
11	10,170032	3,384
12	11,168765	3,401
13	12,168070	3,410
14	13,167693	3,416
15	14,167488	3,419
16	15,167379	3,421

6.1.2.10. Lempel-Ziv Sıkıştırma Testi

Genel olarak bu test bir dizideki kümülatif olarak ayrık örnek kelimelerin sayısına odaklanır. Burada amaç test edilecek dizinin ne kadar sıkıştırıp sıkıştırılamayacağını belirlemesidir. Eğer dizi anlamlı bir manada sıkıştırılmaz ise sonuç olarak bu dizi rasgele olarak kabul edilir. Bu test üretilen rasgele diziyi Lempel-Ziv algoritması kullanarak sıkıştırır. Bu testin hesaplamasında üretilen bit dizisi ε , üretilen bit dizisinin uzunluğu n , dizi içinde birbirinden ayrılmış kümülatif kelimelerin sayısı W_{obs} olarak tanımlanmıştır. Örneğin üretilen bit dizisi $\varepsilon=010110010$ olarak alındığında Çizelge 6.12’de ki kurala göre sözlük oluşturulur.

Çizelge 6.12. Üretilen bit dizisine göre Lempel-Ziv sözlük oluşumu.

Sözlük	Son Durum	Geçerli Durum	Giriş	Kod Çıkışı	Giriş kısmında elde edilen değer sözlükte yoksa Son durumdaki değerlerin sözlükteki karşılık gelen değerleri kod çıkışına yazılır.
0 = 1		$\varepsilon_0=0$			
1 = 2	0	$\varepsilon_1=0$	01=3 (Sözlükte yok ekle)	1	
01 = 3	1	$\varepsilon_2=0$	10=4 (Sözlükte yok ekle)	2	
10 = 4	0	$\varepsilon_3=0$	01 (Sözlükte Mevcut)		
011 = 5	01	$\varepsilon_4=0$	011=5 (Sözlükte yok ekle)	3	
100 = 6	1	$\varepsilon_5=0$	10 (Sözlükte Mevcut)		
010 = 7	10	$\varepsilon_6=0$	100=6 (Sözlükte yok ekle)	4	
	0	$\varepsilon_7=0$	01 (Sözlükte Mevcut)		
	01	$\varepsilon_8=0$	010=7 (Sözlükte yok ekle)	3	
	0	Son		1	

Çizelge 6.12’te gösterilen kurallara göre oluşturulan Lempel-Ziv sıkıştırma kuralının kod çıkış örneği Çizelge 6.13’te verilmiştir.

Çizelge 6.13. Seçilen örneğin Lempel-Ziv kod çıkış örneği.

Kod Çıkışı	1	2	3	4	3	1
Sözlük Değeri	0	1	01	10	01	0
Örnek Dizi	010110010					

Sıkıştırma kuralına göre sözlüğe yeni eklenen değerlerin sayısı $W_{obs}=5$ olarak bulunur. Test istatistik P-değeri ise Eşitlik 6.25’teki hesaplanabilir.

$$P - \text{değeri} = \frac{1}{2} \operatorname{erfc} \left(\frac{\mu - W_{obs}}{\sqrt{2\sigma^2}} \right) \quad (6.25)$$

6.1.2.11. Doğrusal Karmaşıklık Testi

Bu testin amacı üretilen bit dizinin Doğrusal Geri Beslemeli Kayan Kaydedici (DGBKK) (Linear Feedback Shift Register (LFSR)) uzunluğuna bakılarak rasgele olduğu varsayılan dizinin karmaşıklığının belirlenmesidir. Rasgele olan diziler daha uzun DGBKK ile karakterize edilmekte olup çok kısa olan DGBKK’lar olasılıkla rasgele olmayan dizileri belirlemektedir. Testin hesaplanma aşamasında üretilen bit dizisi ε , üretilen bit dizisinin uzunluğu n , blok içerisindeki bit uzunluğu M ve serbestlik derecesi sayısı K ($K=6$) ile ifade edilmektedir. Bu test için istatistiklerinin referans dağılımında ki-kare (χ^2) dağılımı kullanılmaktadır.

Örneğin rasgele üreteç tarafından üretilen bit dizisi $\varepsilon=1101011110001$ olmak üzere dizi her biri M uzunlukta N tane bloğa bölünür. Burada $M=13$ ve $N=1$ olmaktadır. Daha sonra N tane her bloğa Berlekamp-Massey algoritması uygulanarak belirtilen ikili dizinin DGBKK’nın geri beslemeli polinomunun en düşük derecesi hesaplanır. Ayrıca Berlekamp-Massey algoritması kullanılarak N bloklarının her birinin doğrusal karmaşıklık değeri L_i belirlenmektedir. L_i , i . bloklarının içinde üretilen bütün bitlerin en kısa DGBKK’nın uzunluğunun derecesini belirtir. Bu örnekteki dizi için $L_i=4$ olarak alınmaktadır ve bir sonraki bitten itibaren tüm işlemler

yenilenmektedir. Rasgelelik varsayımı altında teorik ortalama değeri Eşitlik 6.26'da ve her alt dizi için gerekli T_i değeri Eşitlik 6.27'deki gibi hesaplanmaktadır.

$$\mu = \frac{M}{2} + \frac{(9 + (-1)^{M+1})}{36} - \frac{(M/3 + 2/9)}{2^M} \quad (6.26)$$

$$T_i = (-1)^M * (L - \mu) + \frac{2}{9} \quad (6.27)$$

Bu örnek için teorik ortalama değer Eşitlik 6.26'da yerine konularak;

$$\mu = \frac{13}{2} + \frac{(9 + (-1)^{13+1})}{36} - \frac{(13/3 + 2/9)}{2^{13}} = 6,7772$$

hesaplanmaktadır. Daha sonra dağılımın her altdizinin rasgele değişken değeri T_i Eşitlik 6.27'da yerine konulduğunda;

$$T_i = (-1)^{13} \cdot (4 - 6,7772) + \frac{2}{9} = 2,999444$$

olarak bulunmaktadır. Olasılık değerleri π_i önceden hesaplanarak $\pi_0=0,0104$, $\pi_1=0,03125$, $\pi_2=0,125$, $\pi_3=0,50$, $\pi_4=0,25$, $\pi_5=0,0625$ ve $\pi_6=0,020833$ olarak elde edilmektedir. Hesaplanan değerler referans dağılım değeri için ki-kare dağılımında yerine konulduğunda bu test istatistik değeri;

$$\chi^2(obs) = \sum_{i=1}^K \frac{(V_i - N\pi_i)^2}{N\pi_i} = 47,0008$$

$$P-değeri = igamc\left(\frac{K}{2}, \frac{\chi^2}{2}\right) = igamc\left(\frac{6}{2}, \frac{47,0008}{2}\right) = 0,993$$

olarak hesaplanmaktadır. Sonuç olarak $P-değeri=0,993 \geq 0,01$ olduğundan üretilen bit dizileri rasgele kabul edilir.

Farklı bir diğer örnekte üretilen bit dizisinden “ ε =ilk 1 000 000 bit” alındığında $n=10^6$ ve $M=1\,000$ olduğunda $v_0=11$, $v_1=31$, $v_2=116$, $v_3=501$, $v_4=258$, $v_5=57$ ve $v_6=26$ sonuçları elde edilir. Bu değerlerin referans dağılım değeri $\chi^2=2,700348$ ve test istatistik değeri $P\text{-değeri}=0,845406$ olarak hesaplanmaktadır. Sonuç olarak hesaplanan $P\text{-değeri}\geq 0,01$ olduğundan üretilen dizi rasgele kabul edilir.

6.1.2.12. Seri Testi

Bu test, bir dizi içindeki m -bitlik olası bütün örtüşmelerin frekansına odaklanmış olup 2^m adet m -bitlik bütün örtüşen şablonların sayısının, rasgele olduğu kabul edilen bir dizide beklenen sayıya ne kadar yakın olduğunun belirlenmesidir. Rasgele bir dizide tekdüzelik (uniformity) önemli bir özellik olup her m -bitlik şablonun diğer m -bitlik şablonlar gibi ortaya çıkma olasılığının aynı olması beklenir. $m=1$ için uygulanan bu testin sonucu ilk testte açıklanan Frekans testine eşit olmaktadır. Bu testin için üretilen bit dizisi ε , bit dizisinin uzunluğu n , her bir bit bloğunun uzunluğu m , bir dizideki m -bit örneğinin gözlemlenen frekansını $\nabla \psi^2(\text{obs})$ ve aynı dizide m -bit örneğin beklenen frekans değerini $\nabla^2 \psi^2(\text{obs})$ temsil etmektedirler. Bu test için istatistiklerinin referans dağılımında ki-kare (χ^2) dağılımı kullanılmaktadır.

Örneğin rasgele üreteç tarafından üretilen bit dizisi $\varepsilon=0011011101$ olarak alındığında içerisindeki bit uzunluğu $n=10$ olur. ε' dizisi, üretilen n bit uzunluğundaki dizinin sonuna $m-1$ bit “0” biti eklenmesi ile oluşan genişletilmiş bir dizidir. Eğer $m=3$ ise $\varepsilon'=001101110100$, $m=2$ ise $\varepsilon'=00110111010$ ve $m=1$ ise $\varepsilon'=0011011101$ olur ve $m=1$ için dizi orijinal halini almaktadır. Üretilen dizideki tüm olası örtüşen m , $m-1$ ve $m-2$ bitlik örüntülerin frekanslarının belirlenmesi gerekmektedir. Burada $v_{i_1 \dots i_m}$ m -bitlik örneklerin $i_1 \dots i_m$ kadar örneklerin frekansını, $v_{i_1 \dots i_{m-1}}$ $m-1$ bitlik örneklerin $i_1 \dots i_{m-1}$ kadar örneklerin frekansını ve $v_{i_1 \dots i_{m-2}}$ $m-2$ bitlik örneklerin $i_1 \dots i_{m-2}$ kadar olan örneklerin frekansını ifade etmektedir.

Bu bölümdeki örnek eğer $m=3$ alınırsa $m-1=2$ ve $m-2=1$ olmaktadır. Bütün 3-bitlik örüntülerin frekansları $v_{000}=0$, $v_{001}=1$, $v_{010}=1$, $v_{011}=2$, $v_{100}=1$, $v_{101}=2$, $v_{110}=2$, $v_{111}=1$, tüm 2-bitlik örneklerin frekansları $v_{00}=1$, $v_{01}=3$, $v_{10}=3$, $v_{11}=3$ ve bütün 1-

bitlik örneklerin frekansları $v_0=4$ ve $v_1=6$ olarak elde edilmektedir. Beklenen ve gözlemlenen frekansların durumları hakkında karar verebilmek için Eşitlik 6.28, Eşitlik 6.29 ve Eşitlik 6.30 değerleri hesaplanır.

$$\psi_m^2 = \frac{2^m}{n} \sum_{i_1 \dots i_m} \left(v_{i_1 \dots i_m} - \frac{n}{2^m} \right)^2 = \frac{2^m}{n} \sum_{i_1 \dots i_m} (v_{i_1 \dots i_m}^2 - n) \quad (6.28)$$

$$\psi_{m-1}^2 = \frac{2^{m-1}}{n} \sum_{i_1 \dots i_{m-1}} \left(v_{i_1 \dots i_{m-1}} - \frac{n}{2^{m-1}} \right)^2 = \frac{2^{m-1}}{n} \sum_{i_1 \dots i_{m-1}} (v_{i_1 \dots i_{m-1}}^2 - n) \quad (6.29)$$

$$\psi_{m-2}^2 = \frac{2^{m-2}}{n} \sum_{i_1 \dots i_{m-2}} \left(v_{i_1 \dots i_{m-2}} - \frac{n}{2^{m-2}} \right)^2 = \frac{2^{m-2}}{n} \sum_{i_1 \dots i_{m-2}} (v_{i_1 \dots i_{m-2}}^2 - n) \quad (6.30)$$

Bu örnek için alınan değerler Eşitlik 6.28, Eşitlik 6.29 ve Eşitlik 6.30'da yerlerine yazıldığında;

$$\psi_3^2 = \frac{2^3}{10} (0+1+1+4+1+4+4+1) - 10 = 2,80$$

$$\psi_2^2 = \frac{2^{3-1}}{10} (1+9+9+9) - 10 = 1,20$$

$$\psi_1^2 = \frac{2^{3-2}}{10} (16+36) - 10 = 0,40$$

olarak hesaplanmaktadır. Buradan da gözlemlenen ve beklenen frekans değerleri

$$\nabla \psi_m^2 = \psi_m^2 - \psi_{m-1}^2 \quad \text{ve} \quad \nabla^2 \psi_m^2 = \psi_m^2 - 2\psi_{m-1}^2 + \psi_{m-2}^2 \quad \text{formülleri hesaplanmaktadır.}$$

Hesaplanan frekans değerleri yerlerine konulduğunda $\nabla \psi_m^2 = 2,80 - 1,20 = 1,60$ ve

$$\nabla^2 \psi_m^2 = 2,80 - 2 \cdot 1,20 + 0,40 = 0,80 \quad \text{olarak hesaplanmaktadır. Alınan bu sonuçlardan}$$

sonra bu testin istatistik değerleri aşağıdaki gibi hesaplanmaktadır.

$$P-değeri_1 = igamc\left(2^{m-2}, \frac{\nabla \psi_m^2}{2}\right) = igamc(2^{3-2}, \frac{1,60}{2}) = 0,9057$$

$$P-değeri_2 = igamc\left(2^{m-3}, \frac{\nabla^2 \psi_m^2}{2}\right) = igamc(2^{3-3}, \frac{0,80}{2}) = 0,8805$$

Sonuç olarak $P1_değeri=0,9057 \geq 0,01$ ve $P2_değeri=0,8805 \geq 0,01$ olduğundan üretilen bit dizisinin rasgele olduğu kabul edilir.

6.1.2.13. Yaklaşık Entropi Testi

Bu testte beklenen, bir önceki testte olduğu gibi tüm dizi içindeki m-bitlik örtüşen olası şablonlarının frekansının gözlemlenmesine dayanır. Amaç ise rasgele olması beklenen bir dizinin frekansı ile iki ardışık veya komşu uzunluktaki (m ve m+1 bitler) örtüşen bloklarının frekansı ile karşılaştırmaktır. Bu testin hesaplamasında üretilen bit dizisi ε , üretilen bit dizisinin uzunluğu n ve her bir bit bloğunun uzunluğu m ($m+1$ ise 2. blok uzunluğu) ile ifade edilmekle birlikte bu test için istatistiklerinin referans dağılımında ki-kare (χ^2) dağılımı kullanılmaktadır.

Örneğin üretilen bit dizisi $\varepsilon=0100110101$ olarak alındığında $n=10$ ve $m=3$ olsun. Örtüşen diziler oluşturulurken m-1 bitlik dizi dizinin başından dizinin sonuna eklenir. $m-1=3-1=2$ olduğundan dizi başlangıcındaki 0 ve 1 bitleri üretilen dizinin sonuna eklenerek test edilecek yeni dizi $\varepsilon=010011010101$ olur. Daha sonra $m=3$ olduğundan üretilen yeni dizi önce m ve sonrasında m+1 bitlik bloklara bölünür. 3 bit için elde edilen bloklar 010, 100, 001, 011, 110, 101, 010, 101, 010 ve 101 olarak bulunur. $2^m=2^3=8$ olduğundan olası m-bit dizi için hesaplanan bulunma frekansları #000=0, #001=1, #010=3, #011=1, #100=1, #101=3, #110=1, #111=0 sonuçları bulunur. Bundan sonraki aşamada her bir “i” değeri için $C_i^m = \#i/n$ formülü kullanılarak $C_{000}^3=0$, $C_{001}^3=0,10$, $C_{010}^3=0,30$, $C_{011}^3=0,10$, $C_{100}^3=0,10$, $C_{101}^3=0,30$, $C_{110}^3=0,10$ ve $C_{111}^3=0$ olası m-bit değerlerinin sayısı hesaplanır. Buradan da Eşitlik 6.31 kullanılarak ampirik dağılım frekansı;

$$\varphi^{(m)} = \sum_{i=0}^{2^m-1} \pi_i \log \pi_i \quad (6.31)$$

olarak hesaplanır. Eşitlik 6.31'deki terimlerden $\pi_i = C_j^3$ ve $j = \log_2 i$ olarak tanımlanır. Alınan sonuçlar Eşitlik 6.31'de yerine konulduğunda;

$$\begin{aligned} \varphi^{(3)} = & 0(\log 0) + 0,1(\log 0,1) + 0,3(\log 0,3) + 0,1(\log 0,1) + 0,1(\log 0,1) + 0,3(\log 0,3) + \\ & + 0,1(\log 0,1) + 0(\log 0) = -1,64341772 \end{aligned}$$

sonucu elde edilir. Üretilen bit dizisinde ilk 1-4 adım $m+1=4$ m blokları için de tekrarlanır. $m-1=4-1=3$ ilk bit bloğu üretilen bit dizisine eklenerek yeni bit dizisi $\varepsilon=010011010101010$ olarak elde edilir. 4 bit için elde edilen bloklar 0100, 1001, 0011, 0110, 1101, 1010, 0101, 1010, 0101 ve 1010 olarak bulunur. $m=4$ bit için olası $2^m=2^4=16$ durum elde edilir. Bulunma frekansları #0011=1, #0100=1, #0101=2, #0110=1, #1001=1, #1010=3, #1101=1 ve diğer tüm durumlar sıfırdır. Her bir "i" değeri için $C_{0011}^4 = C_{0100}^4 = C_{0110}^4 = C_{1001}^4 = C_{1101}^4 = 0,10$, $C_{0101}^4 = 0,20$, $C_{1010}^4 = 0,30$ ve diğer tüm değerler sıfırdır. Eşitlik 6.31 kullanılarak $m=4$ için ampirik dağılım frekansı hesaplanmaktadır.

$$\begin{aligned} \varphi^{(4)} = & 0 + 0 + 0 + 0,1(\log 0,01) + 0,1(\log 0,01) + 0,2(\log 0,03) + 0,1(\log 0,01) + 0 + 0 + \\ & + 0,1(\log 0,01) + 0,3(\log 0,03) + 0 + 0 + 0,1(\log 0,01) + 0 + 0 = -1,83437197 \end{aligned}$$

Dizinin referans dağılım istatistiği hesaplanırken $ApEn(m) = \varphi^{(m)} - \varphi^{(m+1)}$ olmak üzere $\chi^2(obs) = 2n[\log 2 - ApEn(m)]$ ifadeleri kullanılarak $ApEn(3) = 0,190954$ ve $\chi^2(obs) = 2 \cdot 10[0,693147 - 0,190954] = 0,502193$ sonuçları elde edilir. Alınan bu sonuçlardan sonra test istatistik değeri;

$$P\text{-değeri} = igamc\left(2^{m-1}, \frac{\chi^2(obs)}{2}\right) = igamc\left(2^{3-1}, \frac{0,5021}{2}\right) = 0,261961$$

olarak hesaplanmaktadır. Sonuç olarak $P\text{-değeri} = 0,261961 \geq 0,01$ olduğundan üretilen bit dizileri rasgele kabul edilir.

6.1.2.14. Kümülatif Toplamlar Testi

Bu teste ilk olarak dizi içindeki bitler $X_i = 2\varepsilon_i - 1$ dönüşümü ile normalize edilmiş -1 ve +1 değerlerinin kümülatif toplamı olarak tanımlanmış olup rasgele yürüyüş maksimal gezinimlerin araştırılmasına odaklanmıştır. Bu testin amacı test edilecek bir dizide kısmi alt dizilerinin kümülatif toplamlarının rasgele olduğu bilinen bir dizinin beklenen değerine göre çok büyük veya çok küçük olup olmadığının incelenmesidir. Elde edilen bu kümülatif toplam rasgele bir yürüyüş olarak kabul edilebilir. Rasgele dizilerde rasgele yürüyüş gezinimleri sıfıra yakın değerde olup rasgele olmayan dizilerde bu rasgele yürüyüş gezinim değeri sıfırdan çok uzak olmaktadır. Bu testin hesaplamasında üretilen bit dizisi ε , üretilen bit dizisinin uzunluğu n , dizi içindeki kümülatif toplamların en büyük gezinim değeri ve $mode=0$ için giriş dizisi ileri yönde, $mode=1$ için dizi geri yönde uygulanır. Test istatistiği için uygulanan referans dağılım normal dağılımdır.

Örneğin bu testin hesaplanmasını göstermek için üretilen bit dizisi $\varepsilon=1011010111$ olarak verildiğinde ve $X_i = 2\varepsilon_i - 1$ denklemi kullanılarak normalize işlemi yapıldığında $X=1, (-1), 1, 1, (-1), 1, (-1), 1, 1, 1$ yeni bir dizi elde edilir. S_i kısmi toplam değerleri hesaplanır. Eğer $mode=1$ ise X_1 den $mode=0$ ise X_n den başlanır. Bu hesaplama yönteminin detayları Çizelge 6.14'te gösterilmektedir.

Çizelge 6.14. İleri ($mode=0$) ve geri ($mode=1$) yönlü metot uygulaması.

$mode=0$ (İleri yönlü)	$mode=1$ (Geri yönlü)
$S_1=X_1$	$S_1=X_n$
$S_2=X_1+X_2$	$S_2=X_n+X_{n-1}$
$S_3=X_1+X_2+X_3$	$S_3=X_n+X_{n-1}+X_{n-2}$
.	.
.	.
$S_k=X_1+X_2+X_3+\dots+X_k$	$S_k=X_n+X_{n-1}+X_{n-2}+\dots+X_{n-k+1}$
.	.
.	.
$S_n=X_1+X_2+X_3+\dots+X_k\dots+X_n$	$S_n=X_n+X_{n-1}+X_{n-2}+\dots+X_{n-k+1}\dots+X_1$

Bu bölümdeki örnek için ileri yönlü ($mode=0$) uygulandığında $X=1,-1,1,1,-1,1,-1,1,1$ aşağıdaki sonuçlar elde edilir.

$$S_1 = 1$$

$$S_2 = 1 + (-1) = 0$$

$$S_3 = 1 + (-1) + 1 = 1$$

$$S_4 = 1 + (-1) + 1 + 1 = 2$$

$$S_5 = 1 + (-1) + 1 + 1 + (-1) = 1$$

$$S_6 = 1 + (-1) + 1 + 1 + (-1) + 1 = 2$$

$$S_7 = 1 + (-1) + 1 + 1 + (-1) + 1 + (-1) = 1$$

$$S_8 = 1 + (-1) + 1 + 1 + (-1) + 1 + (-1) + 1 = 2$$

$$S_9 = 1 + (-1) + 1 + 1 + (-1) + 1 + (-1) + 1 + 1 = 3$$

$$S_{10} = 1 + (-1) + 1 + 1 + (-1) + 1 + (-1) + 1 + 1 + 1 = 4$$

Elde edilen S değerlerinin mutlak değeri en büyük değeri aynı zamanda test istatistik değeri z olmak üzere $z = \max |S_k| \Rightarrow S_k = 4$ ve $z = 4$ sonuçları elde edilir. Bu testicin kullanılan standart normal kümülatif olasılık dağılım fonksiyonu (φ) Eşitlik 6.32 ile hesaplanmaktadır. Test istatistik değeri P 'nin hesaplanması için Eşitlik 6.33 kullanılmaktadır.

$$\varphi = \frac{1}{2\pi} \int_{-\infty}^z e^{-u^2/2} \cdot du \quad (6.32)$$

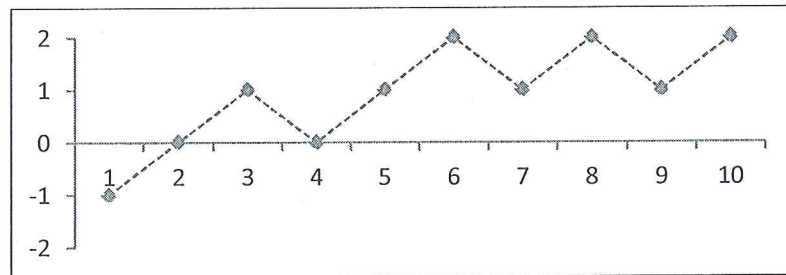
$$P - \text{değeri} = 1 - \sum_{k=\left(\frac{-n}{z}+1\right)/4}^{\left(\frac{n-1}{z}\right)/4} \left[\varphi\left(\frac{(4k+1)z}{\sqrt{n}}\right) - \varphi\left(\frac{(4k-1)z}{\sqrt{n}}\right) \right] + \sum_{k=\left(\frac{-n}{z}-3\right)/4}^{\left(\frac{n-1}{z}\right)/4} \left[\varphi\left(\frac{(4k+3)z}{\sqrt{n}}\right) - \varphi\left(\frac{(4k+1)z}{\sqrt{n}}\right) \right] \quad (6.33)$$

Alınan sonuçların tamamı Eşitlik 6.33'te yerine konulduğunda $P\text{-değeri}=0,41165$ olarak bulunmaktadır. Sonuç olarak $P\text{-değeri}=0,41165 \geq 0,01$ olduğundan üretilen bit dizileri rasgele kabul edilir.

6.1.2.15. Rasgele Gezinimler Testi

Bu test bir önceki testte uygulanan kümülatif toplam rasgele yürüyüşündeki K adet döngünün belirlenmesine odaklanmıştır. Kümülatif toplam rasgele yürüyüşünde 0 ve 1'lerden oluşan dizinin $X_i = 2\varepsilon_i - 1$ dönüşümü kullanılarak -1 ve +1 olarak normalize edildikten sonraki kısmi toplamlarının hesaplaması ile elde edilmektedir. Bu test gerçekte 8 adet (-4, -3, -2, -1 ve +1, +2, +3, +4) testten oluşan bir seridir. Bu testin hesaplamasında üretilen bit dizisi ε ve üretilen bit dizisinin uzunluğu n ile ifade edilmekle birlikte bu test için istatistiklerinin referans dağılımında ki-kare (χ^2) dağılımı kullanılmaktadır.

Örneğin üretilen bit dizisi $\varepsilon=0110110101$ olarak alındığında $n=10$ olmaktadır. Üretilen bit dizisi $X_i = 2\varepsilon_i - 1$ dönüşümü kullanılarak -1 ve +1 olarak normalize edilen yeni dizi $X=-1, 1, 1, -1, 1, 1, -1, 1, -1, 1$ olarak bulunmaktadır. Bir önceki kümülatif toplamlar testinde olduğu gibi Çizelge 6.14'teki ileri yönlü ($mode=0$) yöntem uygulandığında S_i kısmi toplam değerleri $S_1=-1, S_2=0, S_3=1, S_4=0, S_5=1, S_6=2, S_7=1, S_8=2, S_9=1, S_{10}=2$ olarak bulunurlar. Sonuç olarak S kümesi $S=\{-1, 0, 1, 0, 1, 2, 1, 2, 1, 2\}$ olmaktadır. S' kümesi bulunurken S kümesinin başına ve sonuna 0 eklenerek $S' = \{0, -1, 0, 1, 0, 1, 2, 1, 2, 1, 2, 0\}$ elde edilmektedir. Bu aşamada rasgele yürüyüş sonuçları Şekil 6.1'de verilmektedir.



Şekil 6.1. Örnekteki S' kümesi için rasgele yürüyüş sonucu.

Rasgele dizideki döngü sayısını ve S' deki sıfır geçişlerinin toplam sayısını ifade eden J değeri başlangıçtaki sıfır ile sonraki sıfır arası ilk döngüyü oluşturur. Sonrasında ilk döngünün sonu ikinci döngünün başlangıcı olacak ve sonraki sıfır son olarak diğer döngü oluşturulur. Eğer $J < 500$ test işlemi sonlandırılır. Bu test için alınan örnekte $S' = \{0, -1, 0, 1, 0, 1, 2, 1, 2, 1, 2, 0\}$ için başlangıçtaki sıfır hariç 3 tane sıfır bulunduğu (sıfırların pozisyonları 3, 5 ve 12) $J=3$ olur ve burada $\{0, -1, 0\}$, $\{0, 1, 0\}$ ile $\{0, 1, 2, 1, 2, 1, 2, 0\}$ olmak üzere 3 tane döngü vardır. Elde edilen tüm döngüler kullanılarak x durum değerlerinin frekansları Çizelge 6.15'te gösterildiği gibi bulunmaktadır. Her bir döngü ve sıfır olmayan durum değeri için $-4 \leq x \leq -1$ ve $1 \leq x \leq 4$ değerlerini alabilirken her bir döngü içinde her bir x değerinin frekansı hesaplanmaktadır. Bu örnek için ilk döngü 1 tane -1'in oluşumuna, ikinci döngü 1 tane 1'in oluşumuna ve üçüncü döngü 3'er tane 1 ile 2'nin oluşumuna sahiptir.

Çizelge 6.15. Üretilen ε dizinde oluşan rasgele yürüyüş döngü frekansları.

Durum x	Döngüler		
	Döngü 1 $\{0, -1, 0\}$	Döngü 2 $\{0, 1, 0\}$	Döngü 3 $\{0, 1, 2, 1, 2, 1, 2, 0\}$
-4	0	0	0
-3	0	0	0
-2	0	0	0
-1	1	0	0
1	0	1	3
2	0	0	3
3	0	0	0
4	0	0	0

Bundan sonraki aşamada x sekiz durum değerinin her biri için $k=0, 1, \dots, 5$ için tüm döngüler arasında k defa meydana gelen x durumundaki döngünün toplam sayısı $v_k(x)$ değeri hesaplanır. Ayrıca dikkatle incelendiğinde $\sum_{k=0}^5 v_k(x) = J$ olduğu görülmektedir. Bu bölümde verilen örnek için:

1. $v_0(-1)=2$ (-1 durumu 2 döngüde de 0 kez meydana gelir)
 $v_1(-1)=1$ (-1 durumu 1 döngüde de 1 kez meydana gelir)
 $v_2(-1)= v_3(-1)= v_4(-1)= v_5(-1)=0$ (-1 durumu 0 döngüde de $\{2, 3, 4, \geq 5\}$ kez meydana gelir)
2. $v_0(1)=1$ (1 durumu 1 döngüde de 0 kez meydana gelir)

- $v_1(1)=2$ (1 durumu 1 döngüde de 1 kez meydana gelir)
 $v_3(1)=2$ (1 durumu 1 döngüde de 3 kez meydana gelir)
 $v_2(1)=v_4(1)=v_5(1)=0$ (1 durumu 0 döngüde $\{2, 4, \geq 5\}$ kez meydana gelir)
 3. $v_0(2)=2$ (2 durumu 2 döngüde de 0 kez meydana gelir)
 $v_3(2)=1$ (2 durumu 1 döngüde de 3 kez meydana gelir)
 $v_1(2)=v_2(2)=v_4(2)=v_5(2)=0$ (2 durumu 0 döngüde $\{1, 2, 4, \geq 5\}$ kez meydana gelir)
 4. $v_0(-4)=3$ (-4 durumu 3 döngüde de 0 kez meydana gelir)
 $v_1(-4)=v_2(-4)=v_3(-4)=v_4(-4)=v_5(-4)=0$ (-4 durumu 0 döngüde $\{1, 2, 3, 4, \geq 5\}$ kez meydana gelir)

Daha sonraki aşamada x 'in her bir durum değerinin ki-kare istatistik değeri Eşitlik 6.34 kullanılarak hesaplanmaktadır.

$$\chi^2(obs) = \sum_{k=0}^5 \frac{(v_k(x) - J\pi_k(x))^2}{J\pi_k(x)} \quad (6.33)$$

Eşitlik 6.33'te verilen $\pi_k(x)$ değeri, x durum değerinin rasgele dağılım içerisindeki k kez meydana çıkma olasılığını ifade etmekle birlikte Çizelge 6.16'da gösterilmektedir.

Çizelge 6.16. Üretilen ε dizi içerisinde x durum değişkeninin oluşma olasılığı.

	$\pi_0(x)$	$\pi_1(x)$	$\pi_2(x)$	$\pi_3(x)$	$\pi_4(x)$	$\pi_5(x)$
$x=0$	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
$x=1$	0,5000	0,2500	0,1250	0,0625	0,0312	0,0312
$x=2$	0,7500	0,0625	0,0469	0,0352	0,0264	0,0791
$x=3$	0,8333	0,0278	0,0231	0,0193	0,0161	0,0804
$x=4$	0,8750	0,0156	0,0137	0,0120	0,0105	0,0733
$x=5$	0,9000	0,0100	0,0090	0,0081	0,0073	0,0656
$x=6$	0,9167	0,0069	0,0064	0,0058	0,0053	0,0588
$x=7$	0,9286	0,0051	0,0047	0,0044	0,0041	0,0531

Burada sadece $x=1$ durumu için Eşitlik 6.33 kullanılarak ki-kare ve sonrasında test istatistik değeri hesaplanmaktadır.

$$\chi^2 = \frac{(1-3(0,50))^2}{3(0,50)} + \frac{(1-3(0,25))^2}{3(0,25)} + \frac{(0-3(0,125))^2}{3(0,125)} + \frac{(1-3(0,0625))^2}{3(0,0625)} + \frac{(0-3(0,0312))^2}{3(0,0312)} + \frac{(0-3(0,0312))^2}{3(0,0312)} = 4,333033$$

$$P-değeri = igamc\left(\frac{5}{2}, \frac{\chi^2(obs)}{2}\right) = igamc\left(\frac{5}{2}, \frac{4,333033}{2}\right) = 0,502529$$

sonucu elde edilmektedir. $x=1$ durum değeri için hesaplanan $P-değeri=0,50252 \geq 0,01$ olduğundan üretilen bit dizileri rasgele kabul edilir. Her bir x durum değeri için toplamda 8 adet ki-kare değeri ve $P-değeri$ hesaplanmaktadır.

Farklı bir örnekte üretilen bit dizisi $\varepsilon=$ “1 000 000 tane 0 ve 1” olarak alındığında bit dizi uzunluğu $n=10^6$ ve $J=1490$ olarak bulunur. Tüm durum değişkenlerini test istatistik sonuçları Çizelge 6.17’de gösterildiği gibi 8 adet test istatistik değeri hesaplanmakta ve x durum değerinin bir tane değeri haricindeki ($x=-1$ durumu $P-değeri < 0,01$) bütün değerler $P-değeri \geq 0,01$ olduğundan üretilen dizinin rasgele olduğu kabul edilebilir. Fakat bit tane değerinin ($x=-1$ için) rasgele sonuç vermemesinden dolayı üretilen bit dizisi rasgele değildir.

Çizelge 6.17. $n=10^6$ adet örnekte verilen bit dizisinin test istatistik sonuçları.

Durum x	$\chi^2(obs)$	J	$P-değeri$	Sonuç
-4	3,835698	1490	0,573306	Başarılı-Rasgele
-3	7,318709		0,197996	Başarılı-Rasgele
-2	7,8611927		0,164011	Başarılı-Rasgele
-1	15,692617		0,007779	Başarısız- Rasgele değil
+1	2,485906		0,778616	Başarılı-Rasgele
+2	5,429381		0,365752	Başarılı-Rasgele
+3	2,404171		0,790853	Başarılı-Rasgele
+4	2,393928		0,792378	Başarılı-Rasgele

6.1.2.16. Rasgele Gezinimler Değişken Testi

Bu testte, kümülatif toplam rasgele yürüyüşünde özel bir durumun toplam ziyaret edilme sayısının ölçülmesidir. Testin asıl amacı özel durumların beklenen ziyaret sayısındaki sapmaların belirlenmesidir. Bu test gerçekte 18 seriden (-9, -8, ..., -1 ve

+1, +2, ..., +9) oluşan bir testtir. Bu testin hesaplamasında üretilen bit dizisi ε , üretilen bit dizisinin uzunluğu n ve testte bütün rasgele yürüyüşler süresince ziyaret edilen durumların toplam sayısı ξ ile tanımlanmaktadır.

Bu testin için üretilen bit dizisi $\varepsilon=0110110101$ olarak alındığında $n=10$ olmaktadır. Bit dizisindeki sıfırlar ve birler $X_i = 2\varepsilon_i - 1$ dönüşümü kullanılarak -1 ve +1 değerlerine normalize işlemi yapılmaktadır. Bu işlemden sonra X_i yeni dizi $X=-1, 1, 1, -1, 1, 1, -1, 1, -1, 1$ olur. Kümülatif toplamlar testinde olduğu gibi S_i kısmi toplamları hesaplanırken uygulama metodu için $mode=0$ ileri yönlü seçildiğinde $S_1=-1, S_2=0, S_3=1, S_4=0, S_5=1, S_6=2, S_7=1, S_8=2, S_9=1, S_{10}=2$ değerleri elde edilir. Bu işlemden sonra $S=\{-1, 0, 1, 0, 1, 2, 1, 2, 1, 2\}$ kümesi olur. S' kümesi bulunurken S kümesinin başına ve sonuna 0 eklenerek $S' = \{0, -1, 0, 1, 0, 1, 2, 1, 2, 1, 2, 0\}$ kümesi elde edilmektedir. 18 farklı x durumu için tüm J döngüleri içerisinde oluşan x durum değerlerinin toplam sayısı $\xi(x)$ hesaplanmaktadır. Bu bölümdeki örnek uygulama için $\xi(-1)=1, \xi(1)=4, \xi(2)=3$ ve diğer tüm durumlarda $\xi(x)=0$ olarak hesaplanır. Daha sonra her bir $\xi(x)$ için toplamda 18 tane P -değeri Eşitlik 6.34'teki gibi hesaplanmaktadır.

$$P - \text{değeri} = \text{erfc} \left(\frac{|\xi(x) - J|}{\sqrt{2 \cdot J \cdot (4 \cdot |x| - 2)}} \right) \quad (6.34)$$

Sadece $x=1$ durumu için elde edilen sonuçlar Eşitlik 6.34'te yerine konulduğunda;

$$P - \text{değeri} = \text{erfc} \left(\frac{|4 - 3|}{\sqrt{2 \cdot 3 \cdot (4 \cdot 1 - 2)}} \right) = 0,683091$$

olarak hesaplanmaktadır. $x=1$ durum değeri için hesaplanan $P\text{-değeri}=0,68309 \geq 0,01$ olduğundan üretilen bit dizileri rasgele kabul edilir. Bundan sonraki aşamada her bir x durum değeri için ($-9 \leq x \leq -1$ ve $9 \leq x \leq 1$) toplamda 18 adet $\xi(x)$ değeri ve $P\text{-değeri}$ hesaplanmaktadır.

Farklı bir örnekte üretilen bit dizisi ε ="1 000 000 tane 0 ve 1" olarak alındığında bit dizi uzunluğu $n=10^6$ ve $J=1490$ olarak bulunur. Tüm durum değişkenlerini test istatistik sonuçları Çizelge 6.18'de gösterildiği gibi 18 adet test istatistik değeri hesaplanmakta ve bütün değerler $P\text{-değeri} \geq 0,01$ olduğundan üretilen dizinin rasgele olduğu kabul edilmektedir.

Çizelge 6.18. $n=10^6$ adet örnekte verilen bit dizisinin test istatistik sonuçları.

Durum x	$\xi(x)$	J	$P\text{-değeri}$	Sonuç
-9	1450	1490	0,858946	Başarılı-Rasgele
-8	1435		0,794755	Başarılı-Rasgele
-7	1380		0,576249	Başarılı-Rasgele
-6	1366		0,493417	Başarılı-Rasgele
-5	1412		0,633873	Başarılı-Rasgele
-4	1475		0,917283	Başarılı-Rasgele
-3	1480		0,934708	Başarılı-Rasgele
-2	1468		0,816012	Başarılı-Rasgele
-1	1502		0,822609	Başarılı-Rasgele
+1	1409		0,137861	Başarılı-Rasgele
+2	1369		0,200642	Başarılı-Rasgele
+3	1396		0,441254	Başarılı-Rasgele
+4	1479		0,939291	Başarılı-Rasgele
+5	1599		0,505683	Başarılı-Rasgele
+6	1628		0,445593	Başarılı-Rasgele
+7	1619		0,512207	Başarılı-Rasgele
+8	1620		0,538635	Başarılı-Rasgele
+9	1610		0,593930	Başarılı-Rasgele

6.2. TASARLANAN GRSÜ FIPS 140-1 TEST SONUÇLARI

Genel olarak FIPS-140-1 testlerinde dört adet bulunmaktadır. Tasarlanan PCKS ve P3BKS ile ring osilatör kullanılarak geliştirilen çift entropi çekirdekli GRSÜ ünitelerinden 20 000 bit veri alınarak bilgisayar ortamında bir dosyaya kaydedilmiş ve FIPS 140-1 testlerindeki dört adet teste tabi tutulmuştur. Çizelge 6.19'da tasarımı yapılan FPGA-tabanlı PCKS osilatörü kullanılarak gerçekleştirilen çift entropi çekirdekli GRSÜ ünitelerinin FIPS 140-1 testlerinden elde edilen sonuçları verilmiştir. PCKS tabanlı çift entropi çekirdekli GRSÜ üniteleri FIPS 140-1 testlerindeki dört testten de başarılı olmuştur. Koşu testinde çok fazla test olduğundan burada sadece blok uzunluğu 1 için sonuçlar verilmiştir.

Çizelge 6.19. FPGA üzerinde PCKS osilatörün Euler, Heun, RK4 ve RK5-Butcher tabanlı çift entropi çekirdekli GRSÜ ünite tasarımının FIPS 140-1 test sonuçları.

FIPS_140_1 Testleri		Euler Tabanlı	Heun Tabanlı	RK4 Tabanlı	RK5-Butcher Tabanlı
Monobit Testi		9 931 / Başarılı	9 941 / Başarılı	10 079 / Başarılı	10 041 / Başarılı
Poker Testi		12,3456 / Başarılı	17,849 / Başarılı	15,315 / Başarılı	12,582 / Başarılı
Koşu Testi	Run_1	4 996	5 046	5 038	5 049
	Run_2	1 266	1 264	1 247	1 281
	Run_3	311	324	303	327
	Run_4	70	110	68	76
	Run_5	11	35	11	20
	Run_6	3	12	0	8
Uzun Koşu Testi		Başarılı	Başarılı	Başarılı	Başarılı

Çizelge 6.20’de tasarımı yapılan FPGA-tabanlı P3BKS osilatörü kullanılarak gerçekleştirilen çift entropi çekirdekli GRSÜ ünitelerinin FIPS 140-1 testlerinden elde edilen sonuçları verilmiştir. Koşu testinde çok fazla test olduğundan burada sadece blok uzunluğu 1 için sonuçlar verilmiştir. P3BKS tabanlı çift entropi çekirdekli GRSÜ üniteleri FIPS 140-1 testlerindeki dört testten de başarılı olmuştur.

Çizelge 6.20. FPGA üzerinde P3BKS osilatörün Euler, Heun, RK4 ve RK5-Butcher tabanlı çift entropi çekirdekli GRSÜ ünite tasarımının FIPS 140-1 test sonuçları.

FIPS_140_1 Testleri		Euler Tabanlı	Heun Tabanlı	RK4 Tabanlı	RK5-Butcher Tabanlı
Monobit Testi		9 931 / Başarılı	9 941 / Başarılı	10 079 / Başarılı	10 041 / Başarılı
Poker Testi		12,3456 / Başarılı	17,849 / Başarılı	15,315 / Başarılı	12,582 / Başarılı
Koşu Testi	Run_1	4 996	5 046	5 038	5 049
	Run_2	1 266	1 264	1 247	1 281
	Run_3	311	324	303	327
	Run_4	70	110	68	76
	Run_5	11	35	11	20
	Run_6	3	12	0	8
Uzun Koşu Testi		Başarılı	Başarılı	Başarılı	Başarılı

6.3. TASARLANAN GRSÜ NIST 800-22 TEST SONUÇLARI

Uluslararası geçerliliğe sahip diğer bir istatistiksel test olan NIST Test Suite’de 16 adet test bulunmaktadır. Random-Excursions ve Random Excursions Variant testleri için genel olarak 1 milyon bit veriye ihtiyaç duyulmaktadır. Bu nedenle bu testlerin

yapılabilmesi amacıyla genel olarak her bir tasarım için 1 milyon bitlik veriler toplanarak dosyaya kaydedilmiştir. Ardından bit dosyası NIST Test Suite’de bulunan 16 adet teste tabi tutulmuştur. Çizelge 6.21’de FPGA üzerinde PCKS kullanılarak Euler ve Heun algoritması ile tasarlanan çift entropi çekirdekli GRSÜ ünitesi için 1 milyon bit ile yapılan testlerin sonuçları verilmiştir. Test sonuçlarından da görüldüğü gibi $P\text{-değeri} \geq 0,01$ olduğundan elde edilen diziler rasgele olarak kabul edilmektedir.

Çizelge 6.21. FPGA üzerinde PCKS osilatörün Euler ve Heun algoritmaları kullanılarak gerçekleştirilen çift entropi çekirdekli GRSÜ NIST 800-22 istatistiksel test sonuçları.

NIST 800-22 İstatistiksel testler			Euler Tabanlı		Heun Tabanlı	
			P_değeri	Sonuç	P_değeri	Sonuç
1.	Frekans Test		0,72184	Başarılı	0,98882	Başarılı
2.	Blok Frekans Testi		0,06038	Başarılı	0,70946	Başarılı
3.	Akış Testi		0,30368	Başarılı	0,87900	Başarılı
4.	En Uzun Birler Akış Testi		0,19640	Başarılı	0,02197	Başarılı
5.	İkili Matris Rankı Testi		0,99834	Başarılı	0,98444	Başarılı
6.	Ayrık Fourier Dönüşümü Testi		0,12786	Başarılı	0,97395	Başarılı
7.	Örtüşmeyen Şablon Eşleştirme Testi		0,69314	Başarılı	0,52063	Başarılı
8.	Örtüşen Şablon Eşleştirme Testi		0,90598	Başarılı	0,02496	Başarılı
9.	Maurer'in Evrensel İstatistik Testi		0,02262	Başarılı	0,70434	Başarılı
10.	Doğrusal Karmaşıklık Testi		0,03501	Başarılı	0,44137	Başarılı
11.	Seri Test		0,05060	Başarılı	0,53727	Başarılı
			0,87105	Başarılı	0,10461	Başarılı
12.	Lempel Ziv Testi		0,94369	Başarılı	0,21454	Başarılı
13.	Yaklaşık Entropi Testi		0,15224	Başarılı	0,53757	Başarılı
14.	Kümülatif Toplam Testi		0,56254	Başarılı	0,48443	Başarılı
15.	Rasgele Yürüyüş Testi (x=-4 için)	x=-4	0,49514	Başarılı	0,21081	Başarılı
		x=-3	0,37854	Başarılı	0,54789	Başarılı
		x=-2	0,74589	Başarılı	0,24578	Başarılı
		x=-1	0,27548	Başarılı	0,87541	Başarılı
		x=1	0,47895	Başarılı	0,14785	Başarılı
		x=2	0,12547	Başarılı	0,6789	Başarılı
		x=3	0,89654	Başarılı	0,23578	Başarılı
		x=4	0,78951	Başarılı	0,8456	Başarılı
16.	Rasgele Yürüyüş Değişken Testi (x=-9 için)		0,58504	Başarılı	0,67895	Başarılı

Çizelge 6.22’de FPGA üzerinde gerçekleştirilen PCKS kullanılarak RK4 ve RK5-Butcher algoritması ile tasarlanan çift entropi çekirdekli GRSÜ ünitesi için 1 milyon bit ile yapılan testlerin sonuçları verilmiştir. Test sonuçlarından da görüldüğü gibi $P\text{-değeri} \geq 0,01$ olduğundan elde edilen diziler rasgele olarak kabul edilmektedir.

Çizelge 6.22. FPGA üzerinde PCKS osilatörün RK4 ve RK5-Butcher algoritmaları kullanılarak gerçekleştirilen çift entropi çekirdekli GRSÜ NIST 800-22 istatistiksel test sonuçları.

NIST 800-22 İstatistiksel testler			RK4 Tabanlı		RK5-Butcher Tabanlı	
			P_değeri	Sonuç	P_değeri	Sonuç
1.	Frekans Test		0,30207	Başarılı	0,86312	Başarılı
2.	Blok Frekans Testi		0,85206	Başarılı	0,70593	Başarılı
3.	Akış Testi		0,13751	Başarılı	0,74812	Başarılı
4.	En Uzun Birler Akış Testi		0,19603	Başarılı	0,88342	Başarılı
5.	İkili Matris Rankı Testi		0,46454	Başarılı	0,83629	Başarılı
6.	Ayrık Fourier Dönüşümü Testi		0,61942	Başarılı	0,16101	Başarılı
7.	Örtüşmeyen Şablon Eşleştirme Testi		0,78308	Başarılı	0,69314	Başarılı
8.	Örtüşen Şablon Eşleştirme Testi		0,36351	Başarılı	0,86020	Başarılı
9.	Maurer’in Evrensel İstatistik Testi		0,14804	Başarılı	0,60430	Başarılı
10.	Doğrusal Karmaşıklık Testi		0,93967	Başarılı	0,38490	Başarılı
11.	Seri Test		0,62612	Başarılı	0,91870	Başarılı
			0,10229	Başarılı	0,24154	Başarılı
12.	Lempel Ziv Testi		0,78051	Başarılı	0,34521	Başarılı
13.	Yaklaşık Entropi Testi		0,80941	Başarılı	0,84871	Başarılı
14.	Kümülatif Toplam Testi		0,81520	Başarılı	0,50798	Başarılı
15.	Rasgele Yürüyüş Testi ($x=-4$ için)	$x=-4$	0,06158	Başarılı	0,19941	Başarılı
		$x=-3$	0,12458	Başarılı	0,41247	Başarılı
		$x=-2$	0,78952	Başarılı	0,12547	Başarılı
		$x=-1$	0,32145	Başarılı	0,65875	Başarılı
		$x=1$	0,65789	Başarılı	0,45785	Başarılı
		$x=2$	0,47852	Başarılı	0,45698	Başarılı
		$x=3$	0,98754	Başarılı	0,78954	Başarılı
		$x=4$	0,24785	Başarılı	0,86548	Başarılı
16.	Rasgele Yürüyüş Değişken Testi ($x=-9$ için)		0,78521	Başarılı	0,45689	Başarılı

Çizelge 6.23’te FPGA üzerinde P3BKS tasarımı kullanılarak Euler ve Heun algoritmaları ile tasarlanan çift entropi çekirdekli GRSÜ ünitesi için 1 milyon bit ile

yapılan testlerin sonuçları verilmiştir. Test sonuçlarından da görüldüğü gibi $P\text{-değeri} \geq 0,01$ olduğundan elde edilen diziler rasgele olarak kabul edilmektedir.

Çizelge 6.23. FPGA üzerinde P3BKS osilatörün Euler ve Heun algoritmaları kullanılarak gerçekleştirilen çift entropi çekirdekli GRSÜ NIST 800-22 istatistiksel test sonuçları.

NIST 800-22 İstatistiksel testler			Euler Tabanlı		Heun Tabanlı	
			P_değeri	Sonuç	P_değeri	Sonuç
1.	Frekans Test		0,72184	Başarılı	0,98882	Başarılı
2.	Blok Frekans Testi		0,00638	Başarılı	0,70946	Başarılı
3.	Akış Testi		0,30368	Başarılı	0,87900	Başarılı
4.	En Uzun Birler Akış Testi		0,19640	Başarılı	0,02197	Başarılı
5.	İkili Matris Rankı Testi		0,99834	Başarılı	0,98444	Başarılı
6.	Ayrık Fourier Dönüşümü Testi		0,12786	Başarılı	0,97395	Başarılı
7.	Örtüşmeyen Şablon Eşleştirme Testi		0,69314	Başarılı	0,52063	Başarılı
8.	Örtüşen Şablon Eşleştirme Testi		0,90598	Başarılı	0,02496	Başarılı
9.	Maurer'in Evrensel İstatistik Testi		0,02262	Başarılı	0,70434	Başarılı
10.	Doğrusal Karmaşıklık Testi		0,01101	Başarılı	0,44137	Başarılı
11.	Seri Test		0,05060	Başarılı	0,53727	Başarılı
			0,87105	Başarılı	0,10461	Başarılı
12.	Lempel Ziv Testi		0,94369	Başarılı	0,21454	Başarılı
13.	Yaklaşık Entropi Testi		0,15224	Başarılı	0,53757	Başarılı
14.	Kümülatif Toplam Testi		0,56254	Başarılı	0,48443	Başarılı
15.	Rasgele Yürüyüş Testi (x=-4 için)	x=-4	0,49514	Başarılı	0,21081	Başarılı
		x=-3	0,54785	Başarılı	0,25478	Başarılı
		x=-2	0,15987	Başarılı	0,58791	Başarılı
		x=-1	0,07354	Başarılı	0,09785	Başarılı
		x=1	0,35479	Başarılı	0,65874	Başarılı
		x=2	0,74585	Başarılı	0,47862	Başarılı
		x=3	0,35478	Başarılı	0,12548	Başarılı
		x=4	0,86541	Başarılı	0,78546	Başarılı
16.	Rasgele Yürüyüş Değişken Testi (x=-9 için)		0.35789	Başarılı	0.65047	Başarılı

Çizelge 6.24'te FPGA üzerinde P3BKS tasarımı kullanılarak RK4 ve RK5-Butcher algoritmaları ile tasarlanan çift entropi çekirdekli GRSÜ ünitesi için 1 milyon bit ile yapılan testlerin sonuçları verilmiştir. Test sonuçlarından da görüldüğü gibi $P\text{-değeri} \geq 0,01$ olduğundan elde edilen diziler rasgele olarak kabul edilmektedir.

Çizelge 6.24. FPGA üzerinde P3BKS osilatörün RK4 ve RK5-Butcher algoritmaları kullanılarak gerçekleştirilen çift entropi çekirdekli GRSÜ NIST 800-22 istatistiksel test sonuçları.

NIST 800-22 İstatistiksel testler			RK4 Tabanlı		RK5-ButcherTabanlı	
			P_değeri	Sonuç	P_değeri	Sonuç
1.	Frekans Test		0,30207	Başarılı	0,86312	Başarılı
2.	Blok Frekans Testi		0,85206	Başarılı	0,70593	Başarılı
3.	Akış Testi		0,13751	Başarılı	0,74812	Başarılı
4.	En Uzun Birler Akış Testi		0,19603	Başarılı	0,88342	Başarılı
5.	İkili Matris Rankı Testi		0,46454	Başarılı	0,83629	Başarılı
6.	Ayrık Fourier Dönüşümü Testi		0,61942	Başarılı	0,16101	Başarılı
7.	Örtüşmeyen Şablon Eşleştirme Testi		0,78308	Başarılı	0,69314	Başarılı
8.	Örtüşen Şablon Eşleştirme Testi		0,36351	Başarılı	0,86020	Başarılı
9.	Maurer'in Evrensel İstatistik Testi		0,14804	Başarılı	0,60430	Başarılı
10.	Doğrusal Karmaşıklık Testi		0,93967	Başarılı	0,38490	Başarılı
11.	Seri Test		0,62612	Başarılı	0,91870	Başarılı
			0,10229	Başarılı	0,24154	Başarılı
12.	Lempel Ziv Testi		0,78051	Başarılı	0,34521	Başarılı
13.	Yaklaşık Entropi Testi		0,80941	Başarılı	0,84871	Başarılı
14.	Kümülatif Toplam Testi		0,81520	Başarılı	0,50798	Başarılı
15.	Rasgele Yürüyüş Testi (x=-4 için)	x=-4	0,06158	Başarılı	0,19941	Başarılı
		x=-3	0,12548	Başarılı	0,23547	Başarılı
		x=-2	0,26547	Başarılı	0,47585	Başarılı
		x=-1	0,32654	Başarılı	0,65485	Başarılı
		x=1	0,24785	Başarılı	0,12452	Başarılı
		x=2	0,08245	Başarılı	0,12478	Başarılı
		x=3	0,45685	Başarılı	0,65482	Başarılı
		x=4	0,14785	Başarılı	0,42354	Başarılı
16.	Rasgele Yürüyüş Değişken Testi (x=-9 için)		0.76329	Başarılı	0.89522	Başarılı

Bu aşamada her iki kaotik sistemle tasarlanan çift entropi çekirdekli GRSÜ'lerin NIST 800-22 testlerinde Rasgele Gezinimler Değişken Testi için x değişkeni $-9 \leq x \leq -1$ ve $9 \leq x \leq 1$ değerleri arasında toplam 18 test sonucu üretilmektedir. Tablo boyutlarının çok fazla büyümemesi amacıyla bu test için sadece $x=-9$ testinin sonucu verilmektedir. Diğer tüm durumlar için başarı şartını sağlamaktadır. FPGA üzerinde farklı nümerik algoritmalar kullanılarak PCKS ve P3BKS ile tasarlanan çift entropi çekirdekli GRSÜ ünitelerini yapılan tüm testlerden başarılı olduğu görülmektedir [54, 149, 159, 164–167].

BÖLÜM 7

SONUÇLAR VE ÖNERİLER

Bu tez çalışmasında, kaos tabanlı çift entropi çekirdekli 8 adet yeni GRSÜ tasarlanarak FPGA üzerinde gerçekleştirilmiştir.

Bu doğrultuda tezin ilk aşamasında, literatüre yeni sunulan PCKS ve P3BKS tasarımları dört farklı nümerik integrasyon metodu ile modellenmiş ve sistemlerin dinamik davranışları incelenerek kaos analizleri yapılmıştır. Daha sonra PCKS ve P3BKS FPGA üzerinde donanım tanımlama dili VHDL ile 32 bit IQ-Math sabit noktalı sayı standardına uygun olarak modellenmiştir. Modelleme aşamasında Euler, Heun, RK4 ve RK5-Butcher olmak üzere dört farklı algoritma kullanılmıştır. Xilinx Virtex-6 ailesi XC6VLX75T-3FF784 nolu çipi kullanılan tasarımlar Xilinx ISE Design Tools 14.2 benzetim programı kullanılarak sentezlenmiştir. FPGA çip kaynak kullanımına ve ünitelerin saat hızlarına ait parametrelerin istatistikleri incelenmiştir. Elde edilen sonuçlara göre kaotik osilatörlerin çalışma frekansı yaklaşık 390-464 MHz arasında değişmektedir.

İkinci aşamada, Euler, Heun, RK4 ve RK5-Butcher algoritmaları kullanılarak FPGA-tabanlı PCKS ve P3BKS tasarımları ile ring osilatör yapısı birleştirilerek çift entropi çekirdekli GRSÜ tasarımı gerçekleştirilmiştir. Kuantalama işlemleri için yeni Sabit Noktalı Sayı (SNS) modeli sunulmuştur. SNS modeli GRSÜ için FPGA-tabanlı Euler, Heun, RK4 ve RK5-Butcher algoritmaları kullanılarak yapılan tasarımların her birine uygulanmıştır. Genel olarak iki farklı kaotik sistem, kaotik osilatör tasarımında dört ayrı algoritma ve kuantalamada bir yöntem sunularak toplamda 8 farklı GRSÜ ünitesi tasarlanmıştır. FPGA tabanlı GRSÜ modellemelerinde 32 bit IQ-Math sabit noktalı sayı standardı kullanılmıştır. Geliştirilen modeller VHDL donanım tanımlama dili kullanılarak kodlanmıştır. Tasarımı yapılan 8 farklı GRSÜ ünitesi Xilinx firmasının ürettiği Virtex-6 ailesinin XC6VLX75T-3FF784 çipi için sentezlenerek,

FPGA ip kaynak kullanımına ve nitelerin saat hızlarına ait parametrelerin istatistikleri incelenmiştir. GRSÜ nitelerinin verileri işleme süreleri, Xilinx ISE Design Tools 14.2 simölasyon programı kullanılarak elde edilmiştir. SNS-Tabanlı GRSÜ niteleri tasarımlarında sistemin alışma frekansı hızlarında 400-464 Mbit/s arasında deėişen yüksek bit üretim hızları elde edilmiştir.

Son bölümde, FPGA tabanlı GRSÜ'den elde edilen sayı dizileri test edilmiştir. Bu amaçla tasarımı yapılan GRSÜ'nin doėrulanması, uluslararası düzeyde uygulanarak kabul görmüş olan ve en ok kullanılan NIST 800-22 ve FIPS 140-1 testleri ile sağlanmıştır. Tasarlanan PCKS ve P3BKS osilatörlerinden yararlanılarak geliştirilen ift entropi ekirdekli GRSÜ nitelerinden 20 bin veri alınarak bir dosyaya kaydedilmiş ve FIPS-140-1 testlerindeki dört adet teste tabi tutulmuştur. Tüm PCKS ve P3BKS tabanlı GRSÜ niteleri bu testlerden başarılı olmuştur. Diėer bir uluslararası geçerliliėe sahip istatistiksel test olan NIST Test Suite'de 16 adet test bulunmaktadır. Testlerin yapılabilmesi amacıyla her bir tasarım için 1 milyon bit veri toplanarak bir dosyaya kaydedilmiştir. Ardından bit dosyası NIST Test Suite'de bulunan 16 adet teste tabi tutulmuş ve elde edilen tüm diziler rasgelelik testlerinin hepsinden başarılı olmuştur.

Literatürde birçok GRSÜ tasarımları önerilmiştir (Bkz. izelge 5.1, izelge 5.2, izelge 5.3). Bu tasarımlar entropi kaynakları ve üretim tekniklerine göre önemli ölçüde deėişiklikler göstermektedirler. Her tasarımın güçlü ve zayıf yanları vardır. GRSÜ gerçek entropi kaynaklı tamamen rasgele sayılar üreten bir donanımdır. Analog CMOS tabanlı GRSÜ tasarımlarının alışma frekansları 1-25 MHz kadar, bit üretim hızları maksimum 1-40 Mbit/s'ye kadar çıkabilmesine rağmen donanımsal olarak üretilen GRSÜ sistemlerinin ok altında kalmaktadır (Bkz. izelge 5.1). FPGA tabanlı klasik osilatörlerle gerçekleştirilen GRSÜ tasarımlarının alışma frekansları 20-300 MHz seviyelerine kadar çıkabilirken kullanılan osilatörler ve yöntemden dolayı bit üretim hızları 1-40 Mbit/s'ye kadar düşmektedir (Bkz. izelge 5.2). Yukarıda özetlenen alışmalardan da gözlemlendiėi gibi geleneksel GRSÜ yöntemlerinde özümlemeyen bazı önemli sorunlar ortaya çıkmaktadır:

1. Analog elemanlar kullanılarak gerçekleştirilen osilatörlerin çalışma frekanslarının düşük olması,
2. Isıl veya saçılma gürültüsü gibi fiziksel gürültü üreten kaynaklarının ürettikleri işaretlerin çok düşük güçlü olmaları ve dolayısıyla devredeki istenmeyen işaretlerden etkilenmeleri,
3. Ayrıca analog elemanlar kullanılarak gerçekleştirilen yapıların sistem parametrelerinin değişimine dirençli olması,
4. Klasik osilatörlerin çalışma frekanslarının yüksek olmasına rağmen yapısından kaynaklanan ortalama bit üretim hızının belirli bir hızın üzerine çıkamaması.

Geleneksel rasgele sayı üretiminde yukarıda belirtilen olumsuz etkileri yok etmek, daha hızlı ve güvenli rasgele diziler elde etmek için tamamen sayısallaştırılmış gömülü kaos tabanlı GRSÜ'leri alternatif olarak görülmüş ve son bir kaç yılda bu alanda önemli çalışmalar yapılmıştır. Bu çalışmalarda kaotik işaret üreteçlerinin kendilerine has özelliklerinden dolayı GRSÜ'lerin de gürültü kaynağı yerine kullanılabileceği gösterilmiştir. FPGA üzerinde kaotik sistemleri kullanarak gömülü olarak gerçekleştirilen GRSÜ'lerinin hem çalışma frekanslarının 400 MHz seviyelerinde hem de bit üretim hızlarının diğer yöntemlere göre çok daha yüksek olduğu görülmektedir (Bkz. Çizelge 5.3). Bu tez çalışmasında gerçekleştirilen kaos tabanlı çift entropi çekirdekli FPGA-tabanlı yeni kaotik GRSÜ tasarımları, literatürde sunulan diğer çalışmalarla ve yöntemlerle karşılaştırıldığında hem çalışma frekansı (390-464 MHz) hem de bit üretim hızı açısından (390-464 Mbit/s) daha başarılı sonuçlar elde edilmiştir.

Bu çalışmada donanım tabanlı kaotik sistemlerin tasarımında Euler, Heun, RK4 ve RK5-Butcher algoritmaları olmak üzere dört nümerik integrasyon metodu üzerinde çalışmalar yapılmıştır. İleride yapılacak çalışmalar ile daha farklı kaotik sistemler, farklı nümerik integrasyon metotları ile tasarlanıp farklı yapıda ring osilatör tasarımları kullanılarak çift entropi çekirdekli GRSÜ yapıları gerçekleştirilebilir ve literatürdeki çalışmalar geliştirilebilir.

Ayrıca yapılan tez çalışmasında çift entropi çekirdekli GRSÜ yapılarından elde edilen rasgele sayı dizileri bir dosyaya kaydedilerek off-line olarak NIST 800-22 ve

FIPS 140-1 gibi istatistiksel testlere tabi tutulmuştur. İleride yapılacak çalışmalarla bu istatistiksel testler, FPGA üzerinde gerçekleştirilerek testlerin hem daha hızlı yapılmış olması hem de on-line olarak gerçekleşmesi sağlanabilir.

Rasgele sayı üreticileri kriptografi, Monte-Carlo metodunun kullanıldığı uygulamalar, bilgisayar benzetimleri ve modellemeleri ile sayısal analiz uygulamaları gibi birçok alanda kullanılmaktadır. Bu tez çalışmasında FPGA üzerinde kaos tabanlı çift entropi çekirdekli tasarımı gerçekleştirilen yeni GRSÜ ünitelerinden elde edilen yüksek çalışma ve bit üretim hızına sahip rasgele sayı dizilerinin gerekli olduğu hızlı, güvenli ve yoğun işlem gerektiren kriptografi ve güvenli haberleşme alanlarında kullanılabileceklerdir.

KAYNAKLAR

1. Banerjee, S. and Kurths, J., "Chaos and cryptography: A new dimension in secure communications", *The European Physical Journal Special Topics*, 223 (8): 1441-1445 (2014).
2. Xiong, A., Zhao, X., Han, J. and Liu, G., "Application of the chaos theory in the analysis of EMG on patients with facial paralysis", *In Robot Intelligence Technical and Application*, 274: 805-819 (2014).
3. Huang, Z., Dong, W., Duan, H. and Li, H., "Similarity measure between patient traces for clinical pathway analysis: problem, method and applications", *IEEE Journal of Biomedical and Health Informatics*, 18 (1): 4-14 (2014).
4. Hoang, T. M. and Tran, D., "Cryptanalysis and security improvement for selective image encryption", *The European Physical Journal Special Topics*, 223 (8): 1635-1646 (2014).
5. Anees, A., Siddiqui, A. M. and Ahmed, F., "Chaotic substitution for highly autocorrelated data in encryption algorithm", *Communications in Nonlinear Science and Numerical Simulation*, 19 (9): 3106-3118 (2014).
6. Kang, Z., Sun, J., Ma, L., Qi, Y. and Jian, S., "Multimode synchronization of chaotic semiconductor ring laser and its potential in chaos communication", *IEEE Journal of Quantum Electronics*, 50 (3): 148-157 (2014).
7. Mansingka, A. S., Barakat, M. L., Radwan, A. G. and Salama, K. N., "Hardware stream cipher with controllable chaos generator for colour image encryption", *IET Image Processing*, 8 (1): 33-43 (2014).
8. Ji, Y., Zhang, M., Wang, Y., Wang, P., Wang, A., Wu, Y., Xu, H. and Zhang, Y., "Microwave-photonic sensor for remote water-level monitoring based on chaotic laser", *International Journal of Bifurcation and Chaos*, 24 (3): 1450032-1450039 (2014).
9. Wu, Z. G., Shi, P., Su, H. and Chu, J., "Sampled-data fuzzy control of chaotic systems based on a T-S fuzzy model", *IEEE Transactions on Fuzzy Systems*, 22 (1): 153-163 (2014).
10. Yu, S. H., Kang, H. S., Kim, Y. T., Hyun, C. H. and Park, M., "Fuzzy adaptive modular design of uncertain chaotic Duffing oscillators", *International Journal of Control, Automation and Systems*, 12 (1): 188-194 (2014).

11. Ashita, S., Uma, G. and Deivasundari, P., "Chaotic dynamics of a zero average dynamics controlled DC-DC cuk converter", *IET Power Electronics*, 7 (2): 289-298 (2014).
12. Jin, L., Mei, J. and Li, L., "Chaos control of parametric driven Duffing oscillators", *Applied Physics Letters*, 104 (13): 134101-5 (2014).
13. Wei, D. Q., Wan, L., Luo, X. S., Zeng, S. Y. and Zhang, B., "Global exponential stabilization for chaotic brushless DC motors with a single input", *Nonlinear Dynamics*, 77 (1-2): 209-212 (2014).
14. Hemmati, M., Amjady, N. and Ehsan, M., "System modeling and optimization for islanded micro-grid using multi-cross learning-based chaotic differential evolution algorithm", *International Journal of Electrical Power & Energy Systems*, 56 (2014): 349-360 (2014).
15. Pomares, J., Perea, I. and Torres, F., "Dynamic visual servoing with chaos control for redundant robots", *IEEE/ASME Transactions on Mechatronics*, 19 (2): 423-431 (2014).
16. Venkadesan, A., Himavathi, S., Sedhuraman, K. and Muthuramalingam, A., "Design and field programmable gate array implementation of cascade neural network based flux estimator for speed estimation in induction motor drives", *IET Electric Power Applications*, 11 (1): 121-131 (2016).
17. Author, C., Syedul Amin, M., Mamun, M., Hanim Hashim, F., Jalil, J. and Husain, H., "Design and implementation of novel artificial neural network based stock market forecasting system on field-programmable gate arrays", *American Journal Of Applied Sciences*, 8 (10): 1054-1060 (2011).
18. Koyuncu, I., "Design and implementation of high speed artificial neural network based spott 94 S system on FPGA", *International Journal of Intelligent Systems and Applications in Engineering*, 4 (2): 33-39 (2016).
19. Koyuncu, İ., Şahin, İ., Gloster, C. and Sarıtekin, N. K., "A neuron library for rapid realization of artificial neural networks on FPGA: A case study of Rössler chaotic system", *Journal of Circuits, Systems and Computers*, 26 (1): 1750015-21 (2017).
20. Lü, J., Chen, G. and Cheng, D., "A new chaotic system and beyond: the generalized lorenz-like system", *International Journal of Bifurcation and Chaos*, 14 (5): 1507-1537 (2004).
21. Jakimoski, G. and Kocarev, L., "Chaos and cryptography: block encryption ciphers based on chaotic maps", *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 48 (2): 163-169 (2001).
22. Katz, J. and Lindell, Y., "Introduction to Modern Cryptography", Stinson, D. R., *Chapman & Hall/CRC*, New York, USA, 3-20 (2007).

23. Keyman, E. ve Yıldırım, M., "Kriptolojiye Giriş Ders Notları", *Orta Doğu Teknik Üniversitesi Uygulamalı Matematik Enstitüsü Kriptografi Bölümü*, Ankara (2016).
24. Lawande, Q., Ivan, B. and Dhodapkar, S., "Chaos Based Cryptography: A New Approach to Secure Communications, 258", *BARC Newsletter*, 1-12 (2005).
25. Kocarev, L., "Chaos-based cryptography: a brief overview", *IEEE Circuits and Systems Magazine*, 1 (3): 6-21 (2001).
26. Corrigan-Gibbs, H., Mu, W., Boneh, D. and Ford, B., "Ensuring high-quality randomness in cryptographic key generation", *Proceedings of the Conference on Computer & Communications Security*, Berlin, Germany, 685-696 (2013).
27. Kohlbrenner, P. and Gaj, K., "An embedded true random number generator for FPGAs", *12th International Symposium on Field Programmable Gate Arrays*, Colifornia, USA, 71-78 (2004).
28. Avaroğlu, E., Koyuncu, İ., Özer, A. B. and Türk, M., "Hybrid pseudo-random number generator for cryptographic systems", *Nonlinear Dynamics*, 82 (1-2): 239-248 (2015).
29. Kohno, T., Ferguson, N. and Schneier, B., "Cryptography Engineering : Design Principles and Practical Applications", *Wiley Publication*, Indianapolis, Indiana, 137-141 (2010).
30. Pande, A. and Zambreno, J., "Design and hardware implementation of a chaotic encryption scheme for real-time embedded systems", *International Conference on Signal Processing and Communications*, Bangalore, India, 1-5 (2010).
31. Zhang, Z., Chen, G. and Yu, S., "Hyperchaotic signal generation via DSP for efficient perturbations to liquid mixing", *International Journal of Circuit Theory and Applications*, 37 (1): 31-41 (2009).
32. Kharel, R., Busawon, K., Aggoune, W. and Ghassemloy, Z., "Implementation of a secure digital chaotic communication scheme on a DSP board", *Communication Systems Networks and Digital Signal Processing*, Newcastle upon Tyne, United Kingdom, 212-216 (2010).
33. Shi, T., Rui, G., Zhang, Y. and Zhang, S., "Design method for duffing system based on DSP builder", *International Conference on Systems and Informatics*, Yantai, China, 121-124 (2012).
34. Delgado-Restituto, M., Acosta, A. J. and Rodriguez-Vazquez, A., "A mixed-signal integrated circuit for FM-DCSK modulation", *IEEE Journal of Solid-State Circuits*, 40 (7): 1460-1471 (2005).

35. Güler, Ü. and Ergün, S., "A high speed, fully digital IC random number generator", *International Journal Of Electronics and Communications*, 66 (2): 143-149 (2012).
36. Zhang, Y., Liu, Z. and Zheng, X., "A chaos-based image encryption ASIC using reconfigurable logic", *IEEE Asia Pacific Conference on Circuits and Systems*, Macao, China, 1782-1785 (2008).
37. Addabbo, T., Alioto, M. and Fort, A., "A feedback strategy to improve the entropy of a chaos-based random bit generator", *IEEE Transactions on Circuits and Systems-I: Regular Papers Circuits and Systems*, 53 (2): 326-337 (2006).
38. Callegari, S., Rovatti, R. and Setti, G., "First direct implementation of a true random source on programmable hardware", *International Journal of Circuit Theory and Applications*, 33 (1): 1-16 (2005).
39. Drutarovsky, M. and Galajda, P., "A robust chaos-based true random number generator embedded in reconfigurable switched-capacitor hardware", *17th International Conference Radioelektronika*, Brno, Czech Republic, 1-6 (2007).
40. Ozoguz, S. and Ergun, S., "A non-autonomous IC chaotic oscillator and its application for random bit generation", *Proceedings of the European Conference on Circuit Theory and Design*, Cork, Ireland, 165-168 (2005).
41. Kaçar, S., "Analog circuit and microcontroller based RNG application of a new easy realizable 4D chaotic system", *Optik-International Journal for Light and Electron Optics*, 127 (20): 9551-9561 (2016).
42. Azzaz, M. S., Tanougast, C., Sadoudi, S., Fellah, R. and Dandache, A., "A new auto-switched chaotic system and its FPGA implementation", *Communications in Nonlinear Science and Numerical Simulation*, 18 (7): 1792-1804 (2013).
43. Sadoudi, S., Azzaz, M. S., Djeddou, M. and Benssalah, M., "An FPGA real-time implementation of the chen's chaotic system for securing chaotic communications", *International Journal Of Nonlinear Science*, 7 (4): 1749-3889 (2009).
44. Ergun, S. and Ozoguz, S., "Truly random number generators based on a double-scroll attractor", *49th IEEE International Midwest Symposium on Circuits and Systems*, San Juan, PR, USA, 322-326 (2006).
45. Mao, Y., Cao, L. and Liu, W., "Design and FPGA implementation of a pseudo-random bit sequence generator using spatiotemporal chaos", *International Conference on Communications, Circuits and Systems*, Guilin, China, 2114-2118 (2006).
46. Fatemi-Behbahani, E., Ansari-Asl, K. and Farshidi, E., "A new approach to analysis and design of chaos-based random number generators using algorithmic converter", *Circuits Systems and Signal Processing*, 35 (11): 3830-3846 (2016).

47. Zhang, J., Wang, Y., Liu, M., Xue, L., Li, P., Wang, A. and Zhang, M., "A robust random number generator based on differential comparison of chaotic laser signals", *Optics Express*, 20 (7): 7496-7506 (2012).
48. Zidan, M. A., Radwan, A. G. and Salama, K. N., "Random number generation based on digital differential chaos", *IEEE 54th International Midwest Symposium on Circuits and Systems*, Seoul, Korea, 1-4 (2011).
49. Çicek, İ., Pusane, A. E. ve Dünder, G., "A feasibility study of a 1D chaotic map for true random number generation", *20th Signal Processing and Communications Applications Conference*, Muğla, Turkey, 1-4 (2012).
50. Xingyuan, W., Xue, Q., Lin, T., Xingyuan, W., Xue, Q. and Lin, T., "A novel true random number generator based on mouse movement and a one-dimensional chaotic map", *Mathematical Problems in Engineering*, 2012 (2012): 1-9 (2012).
51. Sundaram, P. S., "Development of a FPGA-based true random number generator for space applications", M. Sc. Thesis, *Linköpings Universitet Linköping Institute of Technology*, Linköping, Sweden, 7-17 (2010).
52. Danger, J. L., Guilley, S. and Hoogvorst, P., "High speed true random number generator based on open loop structures in FPGAs", *Microelectronics Journal*, 40 (11): 1650-1656 (2009).
53. Stipcevic, M., "Quantum random number generators and their applications in cryptography", *Proceedings Article*, 8375 (2012): 1-6 (2012).
54. Varchola, M., "FPGA based true random number generators for embedded cryptographic applications", Ph. D. Thesis, *Technical University of Kosice Faculty of Electrical Engineering and Informatics*, Kosice, Slovakia, 12-20 (2008).
55. Tavas, V., Demirkol, A. S., Ozoguz, S., Kılınç, S., Toker, A. and Zeki, A., "An IC random number generator based on chaos", *International Conference on Applied Electronics*, Pilsen, Czech Republic, 1-4 (2010).
56. Jessa, M. and Jaworski, M., "Randomness of a combined TRNG based on the ring oscillator sampling method", *IEEE International Conferences on Signals and Electronic Systems*, Gliwice, Poland, 323-326 (2010).
57. Wold, K. and Petrovic, S., "Optimizing speed of a true random number generator in FPGA by spectral analysis", *IEEE 4th International Conference on Computer Sciences and Convergence Information Technology*, Seoul, Korea, 1105-1110 (2009).
58. Istvan, H., Suci, A. and Cret, O., "FPGA based TRNG using automatic calibration", *IEEE 5th International Conference on Intelligent Computer Communication and Processing*, Cluj-Napoca, Romania, 373-376 (2009).

59. Petrie, C. S. and Connelly, J. A., "A noise-based IC random number generator for applications in cryptography", *IEEE Transactions on Circuits and Systems I: Fundamental Theory And Applications*, 47 (5): 615-621 (2000).
60. Stojanovski, T. and Kocarev, L., "Chaos-based random number generators-part I: analysis", *IEEE Transactions on Circuits and Systems I: Fundamental Theory And Applications*, 48 (3): 281-288 (2001).
61. Stojanovski, T., Pihl, J. and Kocarev, L., "Chaos-based random number generators Part II: practical realization", *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 48 (3): 382-385 (2001).
62. Ergun, S. and Ozoguz, S., "A Chaos-Modulated Dual Oscillator-Based Truly Random Number Generator", *IEEE International Symposium on Circuits and Systems*, New Orleans, USA, 2482-2485 (2007).
63. Koyuncu, I., Ozcerit, A. T. and Pehlivan, I., "Implementation of FPGA-based real time novel chaotic oscillator", *Nonlinear Dynamics*, 77 (1-2): 49-59 (2014).
64. Chen, G. and Ueta, T., "Chaos in Circuits and Systems", Chua, L. O., *World Scientific*, London, 4-40 (2002).
65. Petrzela, J., Hrubos, Z. and Gotthans, T., "Modeling deterministic chaos using electronic circuits", *Radioengineering*, 20 (2): 438-444 (2011).
66. Pehlivan, İ. and Uyaroğlu, Y., "Chaotic circuit design and analyse of a new chaotic attractor", *6th International Advanced Technologies Symposium*, Elazığ, Turkey, 351-355 (2011).
67. Piper, J. R. and Sprott, J. C., "Simple autonomous chaotic circuits", *IEEE Transactions on Circuits and Systems II: Express Briefs*, 57 (9): 730-734 (2010).
68. Sprott, J. C., "Simple chaotic systems and circuits", *American Journal of Physics*, 68 (8): 758-763 (2000).
69. Alligood, K. T., Sauer, T. D. and Yorke, J. A., "Chaos: An Introduction to Dynamical Systems", *Springer-Verlag*, New York, USA, 67-68 (1997).
70. Chua, L. O., Wu, C. W., Huang, A. and Zhong, G. Q., "A universal circuit for studying and generating chaos I: Routes to chaos", *IEEE Transactions on Circuits and Systems I: Fundamental Theory And Applications*, 40 (10): 732-744 (1993).
71. Murali, K., Lakshmanan, M. and Chua, L. O., "Controlling and synchronization of chaos in the simplest dissipative non-autonomous circuit", *International Journal of Bifurcation and Chaos*, 5 (2): 563-571 (1995).

72. Merah, L., Ali-Pacha, A., Said, N. and Mamat, M., "A pseudo random number generator based on the chaotic system of Chua's circuit, and its real time FPGA implementation", *Applied Mathematical*, 7 (55): 2719-2734 (2013).
73. Yamaçlı, V., Abacı, K. ve Köse, E., "Chua devresinin gerçekleştirilmesi ve simülasyonu", *6th International Advanced Technologies Symposium*, Elazığ, Turkey, 82-86 (2011).
74. Tlelo-Cuautle, E. and Muñoz-Pacheco, J. M., "Simulation of Chua's circuit by automatic control of step-size", *Applied Mathematics and Computation*, 190 (2): 1526-1533 (2007).
75. Razminia, A. and Sadrnia, M. A., "Chua's circuit regulation using a nonlinear adaptive feedback technique", *International Journal of Electrical, Robotics, Electronics and Communications Engineering*, 1 (10): 1508-1512 (2007).
76. Kılıç, R., "A Practical Guide for Studying Chua's Circuits, 71", Chua, L. O., *World Scientific*, Singapore, 1-44 (2010).
77. Qi, G., Chen, G., Du, S., Chen, Z. and Yuan, Z., "Analysis of a new chaotic system", *Physica A: Statistical Mechanics And Its Applications*, 352 (2): 295-308 (2005).
78. Uyaroğlu, Y., "Kaotik lorenz sisteminin yarı-ayna yapısı", *Journal of İstanbul Kultur University*, 2006 (3): 141-146 (2006).
79. Strogatz, S., "Nonlinear Dynamics And Chaos: With Applications To Physics, Biology, Chemistry and Engineering", *Addison Wesley Publication*, Cambridge, USA, 1-11 (1994).
80. Juncu, V. D., Rafiei-Naeini, M. and Dudek, P., "Integrated circuit implementation of a compact discrete-time chaos generator", *Analog Integrated Circuits and Signal Processing*, 46 (3): 275-280 (2006).
81. Özdemir, K., "Sürekli-zamanlı kaos ile rastgele sayı üretici tasarımı", Yüksek Lisans Tezi, *İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü*, İstanbul, Türkiye, 11-14 (2008).
82. Cicek, I., Pusane, A. E. and Dundar, G., "A new dual entropy core true random number generator", *Analog Integrated Circuits and Signal Processing*, 81 (1): 61-70 (2014).
83. Toker, A., Ozoguz, S., Demirkol, A. S., Zeki, A. and Tavas, V., "Integrated cross-coupled chaos oscillator applied to random number generation", *IET Circuits, Devices & Systems*, 3 (1): 1-11 (2009).
84. Degaldo-Restituto, M., Medeiro, F. and Rodríguez-Vázquez, A., "Nonlinear switched-current CMOS IC for random signal generation", *Electronics Letters*, 29 (25): 2190-2191 (1993).

85. Yıldırım, S., "A True Random Number Generaor in FPGA for Cryptographic Application", M. Sc. Thesis, *Orta Doğu Teknik Üniversitesi Fen Bilimleri Enstitüsü*, Ankara, Turkey, 1-19 (2012).
86. Yang, Y., Bai, G. and Chen, H., "A 200Mbps random number generator with jitter-amplified oscillator", *IEEE 5th International Conference on Computing, Communication and Networking Technologies*, Hefei, China, 1-5 (2014).
87. Bochard, N., Bernard, F. and Fischer, V., "Observing the randomness in RO-based TRNG", *IEEE International Conference on Reconfigurable Computing and FPGAs*, Quintana Roo, Mexico, 237-242 (2009).
88. Valtchanov, B., Aubert, A., Bernard, F. and Fischer, V., "Modeling and observing the jitter in ring oscillators implemented in FPGAs", *IEEE 11th Workshop on Design and Diagnostics of Electronic Circuits and Systems*, Bratislava, Slovakia, 1-6 (2008).
89. Sunar, B., Martin, W. and Stinson, D., "A provably secure true random number generator with built-in tolerance to active attacks", *IEEE Transactions on Computers*, 56 (1): 109-119 (2007).
90. Wold, K. and Tan, C. H., "Analysis and enhancement of random number generator in FPGA based on oscillator rings", *International Journal of Reconfigurable Computing*, 2009 (2009): 1-8 (2009).
91. Yalcin, M. E., Suykens, J. A. K. and Vandewalle, J., "True random bit generation from a double-scroll attractor", *IEEE Transactions on Circuits and Systems I: Regular Papers*, 51 (7): 1395-1404 (2004).
92. Ozdemir, K., Kilinc, S. and Ozoguz, S., "Random number generator design using continuous-time chaos", *IEEE 16th Signal Processing, Communication and Applications Conference*, Aydin, Turkey, 1-4 (2008).
93. Cicek, I., Pusane, A. E. and Dundar, G., "A novel design method for discrete time chaos based true random number generators", *Integration, the VLSI Journal*, 47 (1): 38-47 (2014).
94. Pamuk, N., "Determination of chaotic time series in dynamic systems", *BAÜ Fen Bil. Enst. Dergisi*, 15 (1): 77-91 (2013).
95. Yardım, F. E. ve Afacan, E., "Lorenz-tabanlı diferansiyel kaos kaydırmalı anahtarlama (DCSK) modeli kullanılarak kaotik bir haberleşme sisteminin simülasyonu", *Gazi Üniv. Müh. Mim. Fak. Der.*, 25 (1): 101-110 (2010).
96. Pehlivan, İ., "Yeni kaotik sistemler: elektronik devre gerçeklemeleri, senkronizasyon ve güvenli haberleşme uygulamaları", Doktora Tezi, *Sakarya Üniversitesi Fen Bilimleri Enstitüsü*, Sakarya, Türkiye, 1-8, 10-25 (2007).

97. Lyapunov, A. M., "The general problem of the stability of motion", *International Journal of Control*, 55 (3): 531-534 (1992).
98. Özkaynak, F., "Doğrusal olmayan sistemlerde Lyapunov üstellerini hesaplayan yazılımın gerçekleştirilmesi", Yüksek Lisans Tezi, *Fırat Üniversitesi Fen Bilimleri Enstitüsü*, Elazığ, Türkiye, 13-22 (2007).
99. Yılmaz, D. ve Güler, N. F., "Kaotik zaman serisinin analizi üzerine bir araştırma", *Gazi Üniv. Müh. Mim. Fak. Der.*, 21 (4): 759-779 (2006).
100. Pehlivan, İ. and Uyaroglu, Y., "A new 3D chaotic system with golden proportion equilibria: Analysis and electronic circuit realization", *Computers & Electrical Engineering*, 38 (6): 1777-1784 (2012).
101. Abooe, A., Yaghini-Bonabi, H. A. and Jahed-Motlagh, M. R., "Analysis and circuitry realization of a novel three-dimensional chaotic system", *Communications in Nonlinear Science and Numerical Simulation*, 18 (5): 1235-1245 (2013).
102. Deng, K. and Yu, S., "Estimating ultimate bound and finding topological horseshoe for a new chaotic system", *Optik - International Journal for Light and Electron Optics*, 125 (20): 6044-6048 (2014).
103. Zhou, W., Xu, Y., Lu, H. and Pan, L., "On dynamics analysis of a new chaotic attractor", *Physics Letters, Section A: General, Atomic and Solid State Physics*, 372 (36): 5773-5777 (2008).
104. Pehlivan, I., Moroz, I. M. and Vaidyanathan, S., "Analysis, synchronization and circuit design of a novel butterfly attractor", *Journal of Sound and Vibration*, 333 (20): 5077-5096 (2014).
105. Tuna, M., Koyuncu, I., Fidan, C. B. and Pehlivan, I., "Real time implementation of a novel chaotic generator on FPGA", *IEEE 23rd Signal Processing and Communications Applications Conference*, Malatya, Turkey, 698-701 (2015).
106. Internet: Department of Electronic Engineering Kong City University of Hong, "Lyapunov Exponents Toolbox (LET)", <https://www.mathworks.com/matlabcentral/fileexchange/233-let?focused=5039116&tab=function> (1998).
107. Wei, Z., Tang, Y., Chen, H. and Pehlivan, I., "Adaptive reduced-order function projective synchronization and circuit design of hyperchaotic DLE with no equilibria", *Optoelectronics and Advanced Materials-Rapid Communications*, 7 (11): 984-990 (2013).
108. Pehlivan, I., "Four-scroll stellate new chaotic system", *Optoelectronics and Advanced Materials-Rapid Communications*, 5 (9): 1003-1006 (2011).

109. Eroğlu, C., "Implementation of synchronized chaotic systems by field programmable gate array", M. Sc. Thesis, *İzmir Institute of Technology*, İzmir, Turkey, 27-35, 80-107 (2008).
110. Butcher, J. C., "Numerical methods for ordinary differential equations in the 20th century", *Journal of Computational and Applied Mathematics*, 125 (1-2): 1-29 (2000).
111. Senthilkumar, S. and Piah, A. R. M., "An improved fuzzy cellular neural network (IFCNN) for an edge detection based on parallel RK(5,6) approach", *International Journal of Computational Systems Engineering*, 1 (1): 70-78 (2012).
112. Koyuncu, I., Ozcerit, A. T. and Pehlivan, I., "An analog circuit design and FPGA-based implementation of the Burke-Shaw chaotic system", *Optoelectronics and Advanced Materials-Rapid Communications*, 7 (9): 635-638 (2013).
113. Xiang, F. and Chen, X., "A method to generate chaotic attractors based on FPGA", *Applied Mechanics and Materials*, 66 (68): 1301-1304 (2011).
114. Guang-Yi, W., Xu-Lei, B. and Zhong-Lin, W., "Design and FPGA implementation of a new hyperchaotic system", *Chinese Physics B*, 17 (10): 3596-3602 (2008).
115. Aseeri, M. A. and Sobhy, M. I., "A new approach to implement chaotic generators based on filed programmable gate array (FPGA)", *Greener Journal of Science, Engineering and Technology Research*, 3 (7): 205-209 (2012).
116. Merah, L., Ali-Pacha, A., Said, N. H. and Mamat, M., "Design and FPGA implementation of Lorenz chaotic system for information security issues", *Applied Mathematical Sciences*, 7 (5): 237-246 (2013).
117. Koyuncu, I., Ozcerit, A. T. ve Pehlivan, I., "FPGA Tabanlı Kaotik Osilatör Tasarımı ve Gerçeklenmesi", *1st International Symposium on Innovative Technologies in Engineering and Science*, Sakarya, Turkey, 873-879 (2013).
118. Tlelo-Cuautle, E., Pano-Azucena, A. D., Rangel-Magdaleno, J. J., Carbajal-Gomez, V. H. and Rodriguez-Gomez, G., "Generating a 50-scroll chaotic attractor at 66 MHz by using FPGAs", *Nonlinear Dynamics*, 85 (4): 2143-2157 (2016).
119. De Micco, L. and Larrondo, H. A., "FPGA implementation of a chaotic oscillator using RK4 method", *IEEE VII Southern Conference on Programmable Logic*, Cordoba, Argentina, 185-190 (2011).
120. Justin, R., Mathew, B. K. and Abe, S., "FPGA implementation of high quality random number generator using LUT based shift registers", *Procedia Technology*, 24 (2016): 1155-1162 (2016).

121. Ahadpour, S., Sadra, Y. and Arasteh-Fard, Z., "A novel chaotic encryption scheme based on pseudorandom bit padding", *International Journal of Computer Science Issues*, 9 (1): 449-456 (2012).
122. Kocarev, L. and Jakimoski, G., "Pseudorandom bits generated by chaotic maps", *IEEE Transactions on Circuits and Systems I: Fundamental Theory And Applications*, 50 (1): 123-126 (2003).
123. Ergün, S. and Özoguz, S., "Truly random number generators based on a non-autonomous chaotic oscillator", *International Journal of Electronics and Communications*, 61 (4): 235-242 (2007).
124. Özkaynak, F., "Kriptolojik rasgele sayı üreteçleri", *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendislik Dergisi*, 8 (2): 37-45 (2015).
125. Özkaynak, F., "Cryptographically secure random number generator with chaotic additional input", *Nonlinear Dynamics*, 78 (3): 2015-2020 (2014).
126. Avaroğlu, E. ve Türk, M., "Son işlemin gerçek rasgele sayı üreteçleri üzerindeki etkisinin incelenmesi", *6. Uluslararası Bilgi Güvenliği ve Kriptoloji Konferansı*, Ankara, Türkiye, 291-294 (2013).
127. Avaroglu, E. ve Türk, M., "Çok modlu kaotik sarmal kullanılarak rasgele sayı üretimi", *IEEE 21. Sinyal İşleme ve İletişim Uygulamaları Kurultayı*, Girne, Kuzey Kıbrıs Türk Cumhuriyeti, 1-4 (2013).
128. Pellicer-Lostao, C. and Lopez-Ruiz, R., "Pseudo-random bit generation based on 2D chaotic maps of logistic type and its applications in chaotic cryptography", *Computational Science and Its Applications*, Berlin, Germany, 784-796 (2008).
129. Nejati, H., Beirami, A. and Ali, W. H., "Discrete-time chaotic-map truly random number generators: design, implementation, and variability analysis of the zigzag map", *Analog Integrated Circuits and Signal Processing*, 73 (1): 363-374 (2012).
130. Cret, O., Gyorfi, T. and Suciu, A., "Implementing true random number generators based on high fanout nets", *Romanian Journal of Information Science and Technology*, 15 (3): 277-298 (2012).
131. Drutarovsky, M., Simka, M., Fischer, V. and Celle, F., "A simple PLL-based true random number generator for embedded digital systems", *Computing and Informatics*, 23 (5-6): 501-515 (2004).
132. Nejati, H., Beirami, A. and Ali, W. H., "Zigzag map: a variability-aware discrete-time chaotic-map truly random number generator", *Electronics Letters*, 48 (24): 1537-1538 (2012).

133. Koyuncu, İ., “Kriptolojik uygulamalar için FPGA tabanlı yeni kaotik osilatörlerin ve gerçek rasgele sayı üreteçlerinin tasarımı ve gerçekleştirilmesi”, Doktora Tezi, *Sakarya Üniversitesi Fen Bilimleri Enstitüsü*, Sakarya, Türkiye, 1-52, 68-78, 105-107 (2014).
134. Dhanuskodi, S. N., Vijayakumar, A. and Kundu, S., “A Chaotic ring oscillator based random number generator”, *IEEE International Symposium on Hardware-Oriented Security and Trust*, Arlington, USA, 160-165 (2014).
135. Aguilar Angulo, J. A., Kussener, E., Barthelemy, H., and Duval, B., “Discrete chaos-based random number generator”, *IEEE Faible Tension Faible Consommation*, Monaco, Monaco, 1-4 (2014).
136. Moqadasi, H. and Ghaznavi-Ghouschi, M. B., “A new Chua’s circuit with monolithic Chua’s diode and its use for efficient true random number generation in CMOS 180 nm”, *Analog Integrated Circuits and Signal Processing*, 82 (3): 719-731 (2015).
137. Park, M., Rodgers, J. C. and Lathrop, D. P., “True random number generation using CMOS boolean chaotic oscillator”, *Microelectronics Journal*, 46 (12): 1364-1370 (2015).
138. Bucci, M., Germani, L., Luzzi, R., Trifiletti, A. and Varanonuovo, M., “A high-speed oscillator-based truly random number source for cryptographic applications on a smartcard IC”, *IEEE Transactions on Computers*, 52 (4): 403-409 (2003).
139. Bucci, M. and Luzzi, R., “A Fully-Digital Chaos-Based Random Bit Generator”, *The New Codebreakers*, 9100, Ryan, P. Y. A., Naccache, D. and Quisquater, J. J., *Springer Berlin Heidelberg*, Heidelberg, Germany, 396-414 (2016).
140. Tavas, V., Demirkol, A. Ş., Kılınç, S., Özoğuz, S., Toker, A. ve Zeki, A., “Tümleştirmeye uygun kaotik osilatör tabanlı rastgele sayı üreteci”, *The 5th National Symposium on Electrical, Electronics and Computer Engineering*, İzmir, Turkey, 1-5 (2008).
141. Pareschi, F., Setti, G. and Rovatti, R., “Implementation and testing of high-speed CMOS true random number generators based on chaotic systems”, *IEEE Transactions on Circuits and Systems I: Regular Papers*, 57 (12): 3124-3137 (2010).
142. Ning, L., Ding, J., Chuang, B. and Xuecheng, Z., “Design and validation of high speed true random number generators based on prime-length ring oscillators”, *The Journal of China Universities of Posts and Telecommunications*, 22 (4): 1-6 (2015).
143. Fischer, V., Drutarovsk, M., Simka, M. and Bochar, N., “High Performance True Random Number Generator in Altera Stratix FPLDs”, *Field*

Programmable Logic and Application, 3203, Becker, J., Platzner, M. and Vernalde, S., *Springer Berlin Heidelberg*, Heidelberg, Germany, 555-564 (2004).

144. Fischer, V., Bernard, F., Bochar, N. and Varchola, M., "Enhancing security of ring oscillator-based trng implemented in FPGA", *IEEE International Conference on Field Programmable Logic and Applications*, Heidelberg, Germany, 245-250 (2008).
145. Schellekens, D., Preneel, B. and Verbauwhede, I., "FPGA vendor agnostic true random number generator", *IEEE International Conference on Field Programmable Logic and Applications*, Madrid, Spain, 1-6 (2006).
146. Dichtl, M. and Golić, J. D., "High-Speed True Random Number Generation with Logic Gates Only", *Cryptographic Hardware and Embedded Systems*, 4727, Paillier, P. and Verbauwhede, I., *Springer Berlin Heidelberg*, Heidelberg, Germany, 45-62 (2007).
147. Tuncer, T., Avaroglu, E., Türk, M. and Ozer, A. B., "Implementation of non-periodic sampling true random number generator on FPGA", *Informacije Midem*, 44 (4): 296-302 (2015).
148. Avaroğlu, E., Tuncer, T., Özer, A. B., Ergen, B. and Türk, M., "A novel chaos-based post-processing for TRNG", *Nonlinear Dynamics*, 81 (1-2): 189-199 (2015).
149. Zidan, M. A., Radwan, A. G. and Salama, K. N., "Random number generation based on digital differential chaos", *IEEE 54th International Midwest Symposium on Circuits and Systems*, Seoul, Korea, 1-4 (2011).
150. Koyuncu, I., Ozcerit, A. T., Pehlivan, I. and Avaroğlu, E., "Design and implementation of chaos based true random number generator on FPGA", *22nd Signal Processing and Communications Applications Conference*, Trabzon, Turkey, 236-239 (2014).
151. Wang, Q., Yu, S., Li, C., Lu, J., Fang, X., Guyeux, C. and Bahi, J. M., "Theoretical design and FPGA-based implementation of higher-dimensional digital chaotic systems", *IEEE Transactions on Circuits and Systems I: Regular Papers*, 63 (3): 401-412 (2016).
152. Koyuncu, I. and Turan, A. Ö., "The design and realization of a new high speed FPGA-based chaotic true random number generator", *Computers & Electrical Engineering*, In Press, Corrected Proof (2016).
153. Sahin, S., Kavak, A. and Cavuslu, M. A., "Comparison of number formats on FPGA-based OFDM modem architecture", *21st Signal Processing and Communications Applications Conference*, Girne, Kuzey Kıbrıs Türk Cumhuriyeti, 1-4 (2013).

154. Vipin Chandra, S., "A survey on CORDIC algorithm implementations using different number format", *International Journal of Innovative Research in Science, Engineering and Technology*, 3 (6): 13452-13458 (2014).
155. Az, I., Sahin, S. and Cavuslu, M. A., "Implementation of fast fourier and inverse fast fourier transforms in FPGA", *IEEE 15th Signal Processing and Communications Applications*, Eskisehir, Turkey, 1-4 (2007).
156. Kir, B., Altuncu, M. A. and Sahin, S., "FPGA based implementation of CORDIC using different number format", *IEEE The International Conference on Technological Advances in Electrical, Electronics and Computer Engineering*, Konya, Turkey, 444-448 (2013).
157. The National Institute of Standards, "Security requirements for cryptographic modules, Federal Information Processing Standards Publication (FIPS PUB) 140-2", *National Institute of Standards & Technology*, (2002).
158. Bassham, L. E., Rukhin, A. L., Soto, J., Nechvatal, J. R., Smid, M. E., Barker, E. B., Leigh, S. D., Levenson, M., Vangel, M., Banks, D. L., Heckert, N. A., Dray, J. F. and Vo, S., "SP 800-22 Rev. 1a. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications", *National Institute of Standards & Technology*, (2010).
159. Çavuşoğlu, Ü., Zengin, A., Pehlivan, I. and Kaçar, S., "A novel approach for strong S-Box generation algorithm design based on chaotic scaled Zhongtang system", *Nonlinear Dynamics*, 87 (2): 1081-1094 (2017).
160. Çavuşoğlu, Ü., Akgül, A., Kaçar, S., Pehlivan, İ. and Zengin, A., "A novel chaos-based encryption algorithm over TCP data packet for secure communication", *Security and Communication Networks*, 9 (11): 1285-1296 (2016).
161. Bonilla, L. L., Alvaro, M. and Carretero, "Chaos-based true random number generators", *Journal of Mathematics in Industry*, 7 (1): 1-8 (2016).
162. Avaroğlu, E., "Pseudorandom number generator based on Arnold cat map and statistical analysis", *Turkish Journal of Electrical Engineering & Computer Sciences*, 25 (1): 633-643 (2017).
163. Avaroğlu, E., "Donanım tabanlı rasgele sayı üreticinin gerçekleştirilmesi", Doktora Tezi, *Fırat Üniversitesi Fen Bilimleri Enstitüsü*, Elazığ, Türkiye, 1-3, 20-22, 48-81, 83-104 (2014).
164. Demirkol, A. Ş., "Kaotik osilatör girişli ADC tabanlı rastgele sayı üretici", Yüksek Lisans Tezi, *İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü*, İstanbul, Türkiye, 1-2, 9-17 (2007).

165. Erat, M., “FPGA tabanlı, PCI ara yüzölü, gerek zamanlı rasgele sayı üreteci test sistemi tasarımı ve uygulamaları”, Doktora Tezi, *Erciyes Üniversitesi Fen Bilimleri Enstitüsü*, Kayseri, Türkiye, 1-38 (2008).
166. Pareschi, F., “Chaos-based random number generators: monolithic implementation, testing and applications”, Ph. D. Thesis, *Alma Mater Studiorum Università Di Bologna Program in Information Technology*, Bologna, Italy, 47-58, 107-110 (2007).

EK AÇIKLAMALAR A.

XILINX VIRTEX-6 FPGA AİLESİ ÖZELLİKLER

General Description

Virtex®-6 FPGAs are the programmable silicon foundation for Targeted Design Platforms that deliver integrated software and hardware components to enable designers to focus on innovation as soon as their development cycle begins. Using the third-generation ASMBL™ (Advanced Silicon Modular Block) column-based architecture, the Virtex-6 family contains multiple distinct sub-families. This overview covers the devices in the LXT, SXT, and HXT sub-families. Each sub-family contains a different ratio of features to most efficiently address the needs of a wide variety of advanced logic designs. In addition to the high-performance logic fabric, Virtex-6 FPGAs contain many built-in system-level blocks. These features allow logic designers to build the highest levels of performance and functionality into their FPGA-based systems. Built on a 40 nm state-of-the-art copper process technology, Virtex-6 FPGAs are a programmable alternative to custom ASIC technology. Virtex-6 FPGAs offer the best solution for addressing the needs of high-performance logic designers, high-performance DSP designers, and high-performance embedded systems designers with unprecedented logic, DSP, connectivity, and soft microprocessor capabilities.

Summary of Virtex-6 FPGA Features

- Three sub-families:
 - Virtex-6 LXT FPGAs: High-performance logic with advanced serial connectivity
 - Virtex-6 SXT FPGAs: Highest signal processing capability with advanced serial connectivity
 - Virtex-6 HXT FPGAs: Highest bandwidth serial connectivity
- Compatibility across sub-families
 - LXT and SXT devices are footprint compatible in the same package
- Advanced, high-performance FPGA Logic
 - Real 6-input look-up table (LUT) technology
 - Dual LUT5 (5-input LUT) option
 - LUT/dual flip-flop pair for applications requiring rich register mix
 - Improved routing efficiency
 - 64-bit (or two 32-bit) distributed LUT RAM option per 6-input LUT
 - SRL32/dual SRL16 with registered outputs option
- Powerful mixed-mode clock managers (MMCM)
 - MMCM blocks provide zero-delay buffering, frequency synthesis, clock-phase shifting, input-jitter filtering, and phase-matched clock division
- 36-Kb block RAM/FIFOs
 - Dual-port RAM blocks
 - Programmable
 - Dual-port widths up to 36 bits
 - Simple dual-port widths up to 72 bits
 - Enhanced programmable FIFO logic
 - Built-in optional error-correction circuitry
 - Optionally use each block as two independent 18 Kb blocks
- High-performance parallel SelectIO™ technology
 - 1.2 to 2.5V I/O operation
 - Source-synchronous interfacing using ChipSync™ technology
 - Digitally controlled impedance (DCI) active termination
 - Flexible fine-grained I/O banking
 - High-speed memory interface support with integrated write-leveling capability
- Advanced DSP48E1 slices
 - 25 x 18, two's complement multiplier/accumulator
 - Optional pipelining
 - New optional pre-adder to assist filtering applications
 - Optional bitwise logic functionality
 - Dedicated cascade connections
- Flexible configuration options
 - SPI and Parallel Flash interface
 - Multi-bitstream support with dedicated fallback reconfiguration logic
 - Automatic bus width detection
- System Monitor capability on all devices
 - On-chip/off-chip thermal and supply voltage monitoring
 - JTAG access to all monitored quantities
- Integrated interface blocks for PCI Express® designs
 - Compliant to the PCI Express Base Specification 2.0
 - Gen1 (2.5 Gb/s) and Gen2 (5 Gb/s) support with GTX transceivers
 - Endpoint and Root Port capable
 - x1, x2, x4, or x8 lane support per block
- GTX transceivers: up to 6.6 Gb/s
 - Data rates below 480 Mb/s supported by oversampling in FPGA logic.
- GTH transceivers: 2.488 Gb/s to beyond 11 Gb/s
- Integrated 10/100/1000 Mb/s Ethernet MAC block
 - Supports 1000BASE-X PCS/PMA and SGMII using GTX transceivers
 - Supports MII, GMII, and RGMII using SelectIO technology resources
 - 2500Mb/s support available
- 40 nm copper CMOS process technology
- 1.0V core voltage (-1, -2, -3 speed grades only)
- Lower-power 0.9V core voltage option (-1L speed grade only)
- High signal-integrity flip-chip packaging available in standard or Pb-free package options

Configuration

Virtex-6 FPGAs store their customized configuration in SRAM-type internal latches. The number of configuration bits is between 26 Mb and 177 Mb, depending on device size but independent of the specific user-design implementation, unless compression mode is used. The configuration storage is volatile and must be reloaded whenever the FPGA is powered up. This storage can also be reloaded at any time by pulling the PROGRAM_B pin Low. Several methods and data formats for loading configuration are available, determined by the three mode pins.

Bit-serial configurations can be either master serial mode where the FPGA generates the configuration clock (CCLK) signal, or slave serial mode where the external configuration data source also clocks the FPGA. For byte- and word-wide configurations, master SelectMAP mode generates the CCLK signal while slave SelectMAP mode receives the CCLK signal for the 8-, 16-, or 32-bit-wide transfer. Alternatively, serial-peripheral interface (SPI) and byte-peripheral interface (BPI) modes are used with industry-standard flash memories and are clocked by the CCLK output of the FPGA. JTAG mode uses boundary-scan protocols to load bit-serial configuration data.

The bitstream configuration information is generated by the ISE® software using a program called BitGen. The configuration process typically executes the following sequence:

- Detects power-up (power-on reset) or PROGRAM_B when Low.
- Clears the whole configuration memory.
- Samples the mode pins to determine the configuration mode: master or slave, bit-serial or parallel, or bus width.
- Loads the configuration data starting with the bus-width detection pattern followed by a synchronization word, checks for the proper device code, and ends with a cyclic redundancy check (CRC) of the complete bitstream.
- Start-up executes a user-defined sequence of events: releasing the internal reset (or preset) of flip-flops, optionally waiting for the phase-locked loops (PLLs) to lock and/or the DCI to match, activating the output drivers, and transitions the DONE pin High.

Dynamic Reconfiguration Port

The dynamic reconfiguration port (DRP) gives the system designer easy access to configuration bits and status registers for three block types: 32 locations for each clock tile, 128 locations for the System Monitor, and 128 locations for each serial GTX or GTH transceiver.

The DRP behaves like memory-mapped registers, and can access and modify block-specific configuration bits as well as status and control registers.

Encryption, Readback, and Partial Reconfiguration

As a special option, the bitstream can be AES-encrypted to prevent unauthorized copying of the design. The Virtex-6 FPGA performs the decryption using the internally stored 256-bit key that can use battery backup or alternative non-volatile storage.

Most configuration data can be read back without affecting the system's operation. Typically, configuration is an all-or-nothing operation, but the Virtex-6 FPGA also supports partial reconfiguration. When applicable in certain designs, partial reconfiguration can greatly improve the versatility of the FPGA. It is even possible to reconfigure a portion of the FPGA while the rest of the logic remains active i.e., active partial reconfiguration.

CLBs, Slices, and LUTs

The look-up tables (LUTs) in Virtex-6 FPGAs can be configured as either one 6-input LUT (64-bit ROMs) with one output, or as two 5-input LUTs (32-bit ROMs) with separate outputs but common addresses or logic inputs. Each LUT output can optionally be registered in a flip-flop. Four such LUTs and their eight flip-flops as well as multiplexers and arithmetic carry logic form a slice, and two slices form a configurable logic block (CLB). Four flip-flops per slice (one per LUT) can optionally be configured as latches. In that case, the remaining four flip-flops in that slice must remain unused.

Between 25–50% of all slices can also use their LUTs as distributed 64-bit RAM or as 32-bit shift registers (SRL32) or as two SRL16s. Modern synthesis tools take advantage of these highly efficient logic, arithmetic, and memory features. Expert designers can also instantiate them.

Input/Output

The number of I/O pins varies from 240 to 1200 depending on device and package size. Each I/O pin is configurable and can comply with a large number of standards, using up to 2.5V. The *Virtex-6 FPGA SelectIO Resources User Guide* describes the I/O compatibilities of the various I/O options. With the exception of supply pins and a few dedicated configuration pins, all other package pins have the same I/O capabilities, constrained only by certain banking rules.

All I/O pins are organized in banks, with 40 pins per bank. Each bank has one common V_{CCO} output supply-voltage pin, which also powers certain input buffers. Some single-ended input buffers require an externally applied reference voltage (V_{REF}). There are two V_{REF} pins per bank (except configuration bank 0). A single bank can have only one V_{REF} voltage value.

I/O Electrical Characteristics

Single-ended outputs use a conventional CMOS push/pull output structure driving High towards V_{CCO} or Low towards ground, and can be put into high-Z state. The system designer can specify the slew rate and the output strength. The input is always active but is usually ignored while the output is active. Each pin can optionally have a weak pull-up or a weak pull-down resistor.

Any signal pin pair can be configured as differential input pair or output pair. Differential input pin pairs can optionally be terminated with a 100 Ω internal resistor. All Virtex-6 devices support differential standards beyond LVDS: HT, RSDS, BLVDS, differential SSTL, and differential HSTL.

Digitally Controlled Impedance

Digitally controlled impedance (DCI) can control the output drive impedance (series termination) or can provide parallel termination of input signals to V_{CCO} , or split (Thevenin) termination to $V_{CCO}/2$. DCI uses two pins per bank as reference pins, but one such pair can also control multiple banks. VRN must be resistively pulled to V_{CCO} , while VRP must be resistively connected to ground. The resistor must be either 1x or 2x the characteristic trace impedance, typically close to 50 Ω .

I/O Logic

Input and Output Delay

This section describes the available logic resources connected to the I/O interfaces. All inputs and outputs can be configured as either combinatorial or registered. Double data rate (DDR) is supported by all inputs and outputs. Any input or output can be individually delayed by up to 32 increments of ~78 ps each. This is implemented as IODELAY. The number of delay steps can be set by configuration and can also be incremented or decremented while in use.

For using either IODELAY, the system designer must instantiate the IODELAY control block and clock it with a frequency close to 200 MHz. Each 32-tap total IODELAY is controlled by that frequency, thus unaffected by temperature, supply voltage, and processing variations.

ISERDES and OSERDES

Many applications combine high-speed bit-serial I/O with slower parallel operation inside the device. This requires a serializer and deserializer (SerDes) inside the I/O structure. Each input has access to its own deserializer (serial-to-parallel converter) with programmable parallel width of 2, 3, 4, 5, 6, 7, 8, or 10 bits. Each output has access to its own serializer (parallel-to-serial converter) with programmable parallel width of up to 8 bits wide for single data rate (SDR), or up to 10 bits wide for double data rate (DDR).

System Monitor

Every Virtex-6 FPGA contains a System Monitor circuit providing thermal and power supply status information. Sensor outputs are digitized by a 10-bit 200kSPS analog-to-digital converter (ADC). This fully tested and specified ADC can also be used to digitize up to 17 external analog input channels. The System Monitor ADC utilizes an on-chip reference circuit thereby eliminating the need for any external active components. On-chip temperature and power supplies are monitored with a measurement accuracy of $\pm 4^{\circ}\text{C}$ and $\pm 1\%$ respectively.

By default the System Monitor continuously digitizes the output of all on-chip sensors. The most recent measurement results together with maximum and minimum readings are stored in dedicated registers for access at any time through the DRP or JTAG interfaces. User defined alarm thresholds can automatically indicate over temperature events and unacceptable power supply variation. A specified limit (for example: 125 $^{\circ}\text{C}$) can be used to initiate an automatic power down.

Package Marking

All Virtex-6 devices have package top-markings similar to the example shown in [Figure 6-1](#) and explained in [Table 6-1](#).

Note: The informational product change customer notice [XCN11022: Product Marking Change](#) outlines a change to the top marking process from an ink mark to laser marking. The previous top mark included an ink patch.

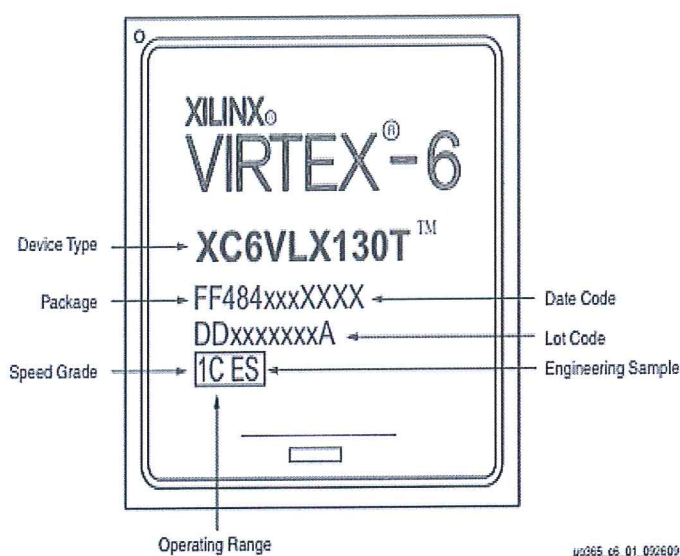


Figure 6-1: Virtex-6 Device Package Marking

Table 6-1: Xilinx Device Marking Definition—Example

Item	Definition
Xilinx Logo	Xilinx logo, Xilinx name with trademark, and trademark-registered status.
Family Brand Logo	Virtex-6 family name with trademark and trademark-registered status. This line is optional and could appear blank.
1st Line	Device type.
2nd Line	Package code, circuit design revision, the location code for the wafer fab, the geometry code, and date code. A G in the third letter of a package code indicates a Pb-free RoHS compliant package. For more information on these packages see Pb-Free and RoHS Compliant Products .
3rd Line	Ten alphanumeric characters for Assembly, Lot, and Step information. The last digit is usually an A or an M if a stepping version does not exist.

Virtex-6 FPGA Feature Summary

Table 1: Virtex-6 FPGA Feature Summary by Device

Device	Logic Cells	Configurable Logic Blocks (CLBs)		DSP48E1 Slices ⁽²⁾	Block RAM Blocks			MMCMs ⁽⁴⁾	Interface Blocks for PCI Express ⁽⁵⁾	Ethernet MACs ⁽⁶⁾	Maximum Transceivers		Total I/O Banks ⁽⁷⁾	Max User I/Os ⁽⁸⁾
		Slices ⁽¹⁾	Max Distributed RAM (Kb)		18 Kb ⁽³⁾	36 Kb	Max (Kb)				GTX	GTH		
XC6VLX75T	74,496	11,640	1,045	288	312	156	5,616	6	1	4	12	0	9	360
XC6VLX130T	128,000	20,000	1,740	480	528	264	9,504	10	2	4	20	0	15	600
XC6VLX195T	199,680	31,200	3,040	640	688	344	12,384	10	2	4	20	0	15	600
XC6VLX240T	241,152	37,680	3,650	768	832	416	14,976	12	2	4	24	0	18	720
XC6VLX365T	364,032	56,880	4,130	576	832	416	14,976	12	2	4	24	0	18	720
XC6VLX550T	549,888	85,920	6,200	864	1,264	632	22,752	18	2	4	36	0	30	1200
XC6VLX760	758,784	118,560	8,280	864	1,440	720	25,920	18	0	0	0	0	30	1200
XC6VSX315T	314,880	49,200	5,090	1,344	1,408	704	25,344	12	2	4	24	0	18	720
XC6VSX475T	476,160	74,400	7,640	2,016	2,128	1,064	38,304	18	2	4	36	0	21	840
XC6VHX250T	251,904	39,360	3,040	576	1,008	504	18,144	12	4	4	48	0	8	320
XC6VHX255T	253,440	39,600	3,050	576	1,032	516	18,576	12	2	2	24	24	12	480
XC6VHX380T	382,464	59,760	4,570	864	1,536	768	27,648	18	4	4	48	24	18	720
XC6VHX565T	566,784	88,560	6,370	864	1,824	912	32,832	18	4	4	48	24	18	720

Virtex-6 FPGA Device-Package Combinations and Maximum I/Os

Virtex-6 LXT and SXT FPGA package combinations with the maximum available I/Os per package are shown in Table 2.

Table 2: Virtex-6 LXT and SXT FPGA Device-Package Combinations and Maximum Available I/Os

Package	FF484 FFG484 FFV484		FF784 FFG784 FFV784		FF1156 FFG1156 FFV1156 ⁽¹⁾		FF1759 FFG1759 FFV1759 ⁽²⁾		FF1760 FFG1760	
Size (mm)	23 x 23		29 x 29		35 x 35		42.5 x 42.5		42.5 x 42.5	
Device	GTXs	I/O	GTXs	I/O	GTXs	I/O	GTXs	I/O	GTXs	I/O
XC6VLX75T	8	240	12	360						
XC6VLX130T	8	240	12	400	20	600				
XC6VLX195T			12	400	20	600				
XC6VLX240T			12	400	20	600	24	720		
XC6VLX365T					20	600	24	720		
XC6VLX550T							36	840	0	1200
XC6VLX760									0	1200
XC6VSX315T					20	600	24	720		
XC6VSX475T					20	600	36	840		

Notes:

- FFV1156 package is available in the LX75T, LX130T, LX195T, and LX240T devices only.
- FFV1759 package is available in the LX240T device only.

Virtex-6 HXT FPGA package combinations with the maximum available I/Os per package are shown in Table 3.

Table 3: Virtex-6 HXT FPGA Device-Package Combinations and Maximum Available I/Os

Package	FF1154 FFG1154			FF1155 FFG1155			FF1923 FFG1923			FF1924 FFG1924		
Size (mm)	35 x 35			35 x 35			45 x 45			45 x 45		
Device	GTXs	GTHs	I/O	GTXs	GTHs	I/O	GTXs	GTHs	I/O	GTXs	GTHs	I/O
XC6VHX250T	48	0	320									
XC6VHX255T				24	12	440	24	24	480			
XC6VHX380T	48	0	320	24	12	440	40	24	720	48	24	640
XC6VHX565T							40	24	720	48	24	640

Notes:

- Flip-chip packages are also available in Pb-Free versions (FFG).

Virtex-6 FPGA Ordering Information

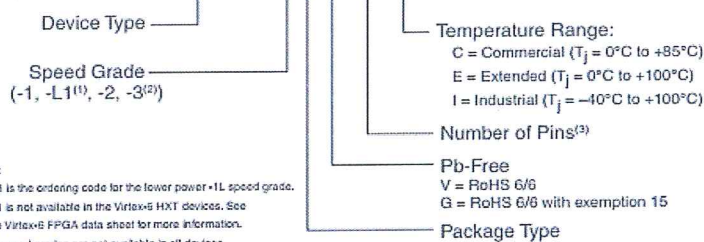
Table 4 shows the speed and temperature grades available in the different Virtex-6 devices. Some devices might not be available in every speed and temperature grade.

Table 4: Virtex-6 FPGAs Speed Grade and Temperature Ranges

Device Family	Speed Grade and Temperature Range		
	Commercial (C) 0°C to +85°C	Extended (E) 0°C to +100°C	Industrial (I) -40°C to +100°C
Virtex-6 LXT	-3, -2, -1, -1L	-2	-2, -1, -1L
Virtex-6 SXT	-3, -2, -1, -1L	-2	-2, -1, -1L
Virtex-6 HXT	-3, -2, -1	-2	-2, -1

The Virtex-6 FPGA ordering information shown in Figure 1 applies to all packages including Pb-Free.

Example: XC6VLX240T-1FFG1156C



Note:

1) -1L is the ordering code for the lower power -1L speed grade.

-1L is not available in the Virtex-6 HXT devices. See the Virtex-6 FPGA data sheet for more information.

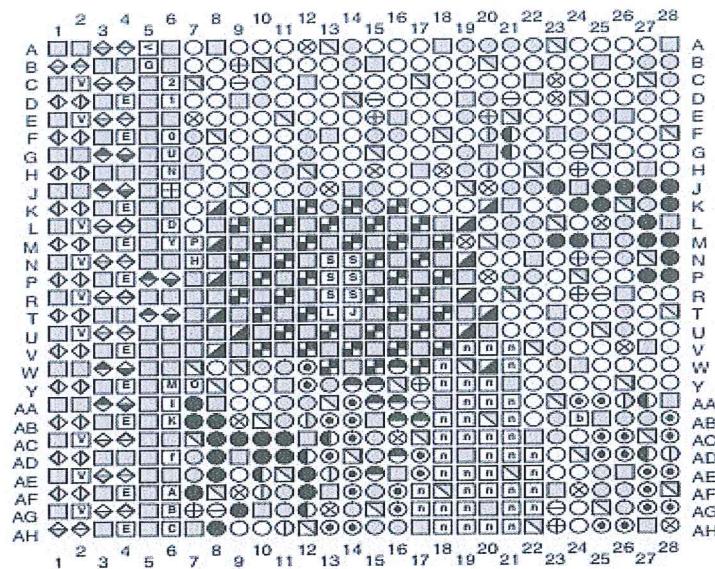
2) -3 speed grades are not available in all devices.

3) Some package names do not exactly match the number of pins present on that package. Please see UG365: Virtex-6 FPGA Packaging and Pinout Specifications for package details.

Table 1-5: Available I/O Pin/Device/Package Combinations

Virtex-6 Device	User I/O Pins	Virtex-6 FPGA Package								
		FF484	FF784 RF784	FF1154	FF1155	FF1156 RF1156	FF1759 RF1759	FF1760	FF1923	FF1924
XC6VLX75T	Available User I/Os	240	360	-	-	-	-	-	-	-
	Differential I/O Pairs	120	180	-	-	-	-	-	-	-
XC6VLX130T	Available User I/Os	240	400	-	-	600	-	-	-	-
	Differential I/O Pairs	120	200	-	-	300	-	-	-	-
XC6VLX195T	Available User I/Os	-	400	-	-	600	-	-	-	-
	Differential I/O Pairs	-	200	-	-	300	-	-	-	-
XC6VLX240T	Available User I/Os	-	400	-	-	600	720	-	-	-
	Differential I/O Pairs	-	200	-	-	300	360	-	-	-
XC6VLX365T	Available User I/Os	-	-	-	-	600	720	-	-	-
	Differential I/O Pairs	-	-	-	-	300	360	-	-	-
XC6VLX550T	Available User I/Os	-	-	-	-	-	840	1200	-	-
	Differential I/O Pairs	-	-	-	-	-	420	600	-	-
XC6VLX760	Available User I/Os	-	-	-	-	-	-	1200	-	-
	Differential I/O Pairs	-	-	-	-	-	-	600	-	-
XC6VSX315T	Available User I/Os	-	-	-	-	600	720	-	-	-
	Differential I/O Pairs	-	-	-	-	300	360	-	-	-
XC6VSX475T	Available User I/Os	-	-	-	-	600	840	-	-	-
	Differential I/O Pairs	-	-	-	-	300	420	-	-	-
XC6VHX250T	Available User I/Os	-	-	320	-	-	-	-	-	-
	Differential I/O Pairs	-	-	160	-	-	-	-	-	-
XC6VHX255T	Available User I/Os	-	-	-	440	-	-	-	480	-
	Differential I/O Pairs	-	-	-	220	-	-	-	240	-
XC6VHX380T	Available User I/Os	-	-	320	440	-	-	-	720	640
	Differential I/O Pairs	-	-	160	220	-	-	-	360	320
XC6VHX565T	Available User I/Os	-	-	-	-	-	-	-	720	640
	Differential I/O Pairs	-	-	-	-	-	-	-	360	320

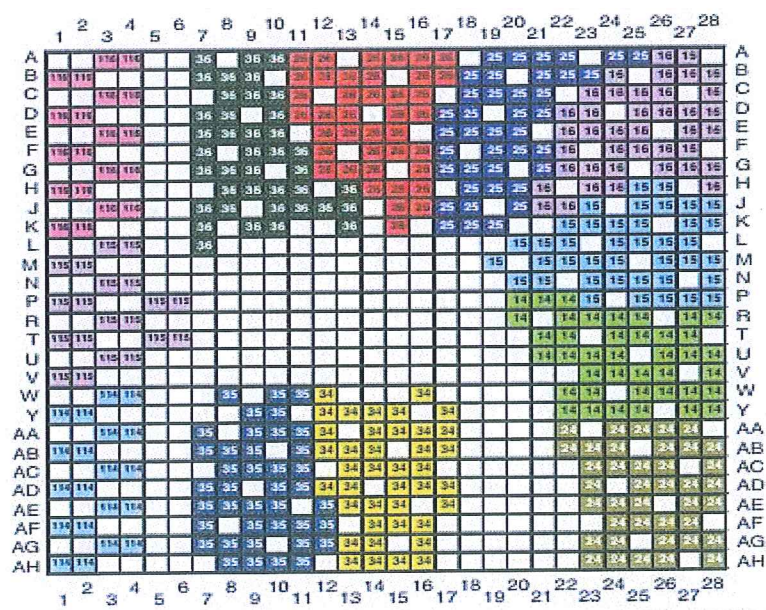
FF784 Package—LX75T



User I/O Pins	Multi-Function Pins	Transceiver Pins	Dedicated Pins	Other Pins
○ IO_LXXY_#	● SM ⊗ VREF ⊕ VRN ⊖ VRP ⊙ P_GC ⊙ N_GC ○ CC ⊙ D0 - D31 ⊙ A0 - A25 ⊞ CS0_B	⊞ MGTAVCC ⊞ MGTAVTT ⊞ MGTAVTTTICAL ⊞ MGTREFCLKP ⊞ MGTREFCLKN ⊞ MGTRREF ⊞ MGTRXP ⊞ MGTRXN ⊞ MGTTXN ⊞ MGTTPX	⊞ CCLK ⊞ CSI_B ⊞ DIN ⊞ DONE ⊞ DOUT_BUSY ⊞ HSWAPEN ⊞ INIT ⊞ M2, M1, M0 ⊞ AVDD, AVSS, VP, VN, VREFP, VREFN ⊞ PROGRAM_B ⊞ RDWR ⊞ TCK ⊞ TDI ⊞ TDO ⊞ TMS ⊞ DXP ⊞ DXN	⊞ GND ⊞ VFS ⊞ VBATT ⊞ VCCAUX ⊞ VCCINT ⊞ VCCO ⊞ NC ⊞ FLOAT

UG365_c3_03_061306

Figure 3-3: FF784 Package—LX75T Pinout Diagram



UG365_c3_04_091509

Figure 3-4: FF784 Package—LX75T I/O Bank Diagram

ÖZGEÇMİŞ

Murat TUNA 1982 yılında Bursa'nın Mustafakemalpaşa ilçesinde doğdu. İlk ve orta öğrenimini aynı şehirde tamamladı. Bursa Ali Osman Sönmez Teknik Endüstri Meslek Lisesi Elektrik Bölümü'nden 2000 yılında mezun oldu. Liseden mezun olduğu yıl girdiği Kocaeli Üniversitesi Teknik Eğitim Fakültesi Elektrik Öğretmenliği bölümünden 2004 yılında mezun olarak 2005 yılında aynı üniversitede Fen Bilimleri Enstitüsü Elektrik Eğitimi bölümünde Yüksek Lisans eğitimine başladı. 2007 yılında Kocaeli sınırları içinde bulunan Elimsan Elektromekanik Şalt Cihazları A.Ş. firmasında üretim, tasarım ve ARGE departmanlarında çeşitli görevlerde yer aldı. 2008 yılında Yüksek Lisans eğitimini tamamladı. 2009 yılında özel sektörden ayrılarak Kırklareli Üniversitesi Teknik Eğitim Fakültesinde Öğretim Görevlisi olarak göreve başladı. Halen Kırklareli Üniversitesi Teknik Bilimler Meslek Yüksekokulunda aktif görevine devam etmektedir.

ADRES BİLGİLERİ

Adres : Kırklareli Üniversitesi
Teknik Bilimler Meslek Yüksekokulu
Karahıdır Mah. Harmanlık Mevkii
39100 Merkez / KIRKLARELİ
Tel : (288) 214 1845
Faks : (288) 214 1495
E-posta : murat.tuna@klu.edu.tr