

**IQ-MATH SAYI STANDARTLARINDA FPGA-TABANLI KAOTİK
OSİLATÖRÜN TASARIMI VE GERÇEKLENMESİ**

YÜKSEK LİSANS TEZİ

Halil İbrahim ŞEKER

Danışman

Dr. Öğretim Üyesi İsmail KOYUNCU

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ

ANABİLİM DALI

Haziran 2019

Bu tez çalışması 18.FEN.BİL.50 numaralı proje ile Afyon Kocatepe Üniversitesi Bilimsel Araştırma Projeleri Koordinasyon Birimi tarafından desteklenmiştir.

AFYON KOCATEPE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

**IQ-MATH SAYI STANDARTLARINDA FPGA-TABANLI KAOTİK
OSİLATÖRÜN TASARIMI VE GERÇEKLENMESİ**

Halil İbrahim ŞEKER

Danışman

Dr. Öğretim Üyesi İsmail KOYUNCU

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

Haziran 2019

TEZ ONAY SAYFASI

Halil İbrahim ŞEKER tarafından hazırlanan "IQ-Math Sayı Standartlarında FPGA-Tabanlı Kaotik Osilatörün Tasarımı Ve Gerçeklenmesi" adlı tez çalışması lisansüstü eğitim ve öğretim yönetmeliğinin ilgili maddeleri uyarınca 12/06/2019 tarihinde aşağıdaki jüri tarafından **oy birliği** ile Afyon Kocatepe Üniversitesi Fen Bilimleri Enstitüsü **Elektrik Elektronik Mühendisliği Anabilim Dalı'nda YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman : Dr. Öğretim Üyesi İsmail KOYUNCU

İmza

Başkan : Prof. Dr. Yüksel OĞUZ
Afyon Kocatepe Üniversitesi, Teknoloji Fakültesi

Üye : Dr. Öğretim Üyesi İsmail KOYUNCU
Afyon Kocatepe Üniversitesi, Teknoloji Fakültesi

Üye : Dr. Öğretim Üyesi Murat TUNA
Kırklareli Üniversitesi, Teknik Bilimler MYO

Afyon Kocatepe Üniversitesi
Fen Bilimleri Enstitüsü Yönetim Kurulu'nun
...../...../..... tarih ve
..... sayılı kararıyla onaylanmıştır.

.....
Prof. Dr. İbrahim EROL
Enstitü Müdürü

BİLİMSEL ETİK BİLDİRİM SAYFASI
Afyon Kocatepe Üniversitesi

Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- Görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- Başkalarının eserlerinden yararlanması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- Atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- Kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- Ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

12/06/2019


Halil İbrahim ŞEKER

ÖZET
Yüksek Lisans Tezi

**IQ-MATH SAYI STANDARTLARINDA FPGA-TABANLI KAOTİK OSİLATÖRÜN
TASARIMI VE GERÇEKLENMESİ**

Halil İbrahim ŞEKER
Afyon Kocatepe Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik-Elektronik Mühendisliği Anabilim Dalı
Danışman: Dr. Öğretim Üyesi İsmail KOYUNCU

Kaos veya kaotik sistemlerin periyodik olmayan karakteristikleri, başlangıç şartlarına ve sistem parametrelerine oldukça hassas bağımlı olmaları, durum uzayında periyodik olmayan özellikler taşımaları ve gürültü benzeri özelliklerinden dolayı son yıllarda mühendisliğin; kriptoloji, güvenli haberleşme, endüstriyel kontrol, yapay sinir ağları, rasgele sayı üreteçleri ve görüntü işleme gibi alanlarında kaotik osilatör tasarımı oldukça büyük öneme sahiptir.

Sunulan bu tez çalışmasında gerçek zamanlı ve yüksek çalışma frekansına sahip FPGA tabanlı farklı IQ-Math sabit noktalı sayı standartlarında kaotik osilatörler tasarlanmış ve gerçekleştirilmiştir.

Tezin ilk aşamasında, SEA kaotik sistemi Matlab tabanlı Euler, Heun, dördüncü dereceden Runge Kutta, beşinci dereceden Runge Kutta Butcher ve Dormand-Prince (DP) nümerik algoritmaları kullanılarak modellenmiştir. Çalışmadan elde edilen sonuçlar kullanılarak SEA kaotik sistemin kaos analizleri yapılmıştır. Modelleme sonucu kaotik osilatörün her bir nümerik algoritmaya ait zaman serileri ve faz portreleri incelenmiştir. SEA kaotik sisteminin DP-tabanlı nümerik modelinden diğer nümerik algoritmalara göre daha hassas çözümler elde edilmiştir.

Tezin ikinci aşamasında, SEA kaotik sistemi 32-bit IEEE 754-1985 kayan noktalı sayı standardında DP nümerik algoritması kullanılarak FPGA çipi üzerinde VHDL dili ile

kodlanmıştır. Yapılan tasarım Xilinx ISE 14.7 benzetim programı kullanılarak sentezlenmiş ve test edilmiştir. Ardından DP-tabanlı SEA kaotik osilatörü Xilinx Virtex-6 ailesi XC6VLX240T-1FF1156 FPGA çipi üzerinde gerçekleştirilmiştir. Tasarımdan elde edilen çip istatistikleri ve çalışma frekansı sunulmuştur.

Tezin üçüncü aşamasında, SEA kaotik sistemi 32-bit (16I-16Q), 28-bit (14I-14Q), 24-bit (12I-12Q), 20-bit (10I-10), ve 16-bit (8I-8Q) IQ-Math sabit noktalı sayı standartlarına uygun biçimde DP nümerik algoritması kullanılarak FPGA çipi üzerinde VHDL dilinde kodlanmıştır. Yapılan tasarım Xilinx Virtex-6 ailesi XC6VLX240T-1FF1156 FPGA çipi üzerinde Xilinx ISE 14.7 benzetim programı kullanılarak sentezlenmiş ve test edilmiştir. Tasarımı yapılan beş farklı sabit noktalı sayı standardındaki SEA kaotik osilatörlerinin çip istatistikleri ve çalışma frekansları sunulmuştur.

Tezin son aşamasında ise beş farklı IQ-Math sabit noktalı sayı formatındaki ve 32-bit IEEE 754-1985 kayan noktalı sayı formatındaki SEA kaotik osilatörlerin çip istatistikleri, çalışma frekansları karşılaştırılmıştır. Ayrıca FPGA üzerinde gerçekleştirilmesinden elde edilen sonuçlar kullanılarak RMSE ile MSE hata analizleri yapılmış ve çalışmalardan elde edilen sonuçlar değerlendirilmiştir.

2019, xiv + 82 sayfa

Anahtar Kelimeler: Dormand-Prince, IQ-Math sayı standardı, IEEE 754-1985 standardı, FPGA, Kaotik osilatör, VHDL

ABSTRACT
M.Sc. Thesis

DESIGN AND IMPLEMENTATION OF FPGA-BASED CHAOTIC OSCILLATOR
IN IQ-MATH NUMBER STANDARD

Halil İbrahim ŞEKER

Afyon Kocatepe University

Graduate School of Natural and Applied Sciences

Department of Electrical-Electronics Engineering

Supervisor: Asst. Prof. İsmail KOYUNCU

Due to the non-periodic characteristics of chaos or chaotic systems, their dependence on initial conditions and system parameters, their non-periodicity in the state space, and their noise-like characteristics, engineering in recent years; The design of chaotic oscillators in areas such as cryptology, secure communication, industrial control, artificial neural networks, random number generators and image processing is of quite great importance.

In this thesis, chaotic oscillators have been designed and implemented in accordance with FPGA based IQ-Math fixed point number standards with real time and maximum operating frequency.

In the first stage of the thesis, the SEA chaotic system was modelled using Matlab based Euler, Heun, fourth order Runge Kutta, fifth order Runge Kutta Butcher, Dormand-Prince (DP) numerical algorithms and chaos analysis of the SEA chaotic system. Time series and phase portraits of each numerical algorithm of the SEA chaotic oscillator were investigated. The DP-based numerical model of the SEA chaotic system is more sensitive than other numerical algorithms.

In the second stage of the thesis, the SEA chaotic system was coded in the VHDL language on the FPGA chip using a 32-bit IEEE 754-1985 floating-point number standard DP numerical algorithm. The design was synthesized and tested using Xilinx

ISE 14.7 simulation program. The DP-based SEA chaotic oscillator was then implemented on the Xilinx Virtex-6 family XC6VLX240T-1FF1156 FPGA chip. Chip statistics and maximum operating frequency of the DP-based SEA chaotic oscillator are presented.

In the third stage of the thesis, the SEA chaotic system was encoded in the VHDL language on the FPGA chip using the DP numerical algorithm in accordance with 32-bit (16I-16Q), 28-bit (14I-14Q), 24-bit (12I-12Q), 20-bit (10I-10) and 16-bit (8I- 8Q) IQ-Math fixed-point number standards. The Xilinx Virtex-6 family was tested and synthesized using Xilinx ISE 14.7 simulation program on XC6VLX240T-1FF1156 FPGA chip. The chip statistics and maximum operating frequencies of the SEA chaotic oscillators in five different fixed-point number standards were presented.

In the last stage of the thesis, chip statistics and maximum operating frequencies of the SEA chaotic oscillators in five different IQ-Math fixed-point number standards and 32-bit IEEE 754-1985 floating point number standard were compared. In addition, RMSE and MSE error on the results obtained from the implementation of FPGA were performed and the results obtained from the studies were evaluated.

2019, xiv + 82 pages

Keywords: Dormand-Prince, IQ-Math fixed point, IEEE 754-1985 standard, FPGA, Chaotic oscillator, VHDL

TEŐEKKÜR

Tez alıŐması boyunca maddi ve manevi her tŸrlŸ desteęini esirgemeyen aynı zamanda tez amacının belirlenmesinde ve tez aŐamalarında bilgi birikimini ve tecrŸbelerini benimle paylaŐan ok saygı deęer danıŐman hocam Sayın Dr. Őđretim Őyesi İsmail KOYUNCU 'ya en iten samimi duygularıyla sonsuz teŐekkŸrlerimi sunarım.

Tez alıŐmalarım boyunca takıldıęım noktalarda yardımlarını esirgemeyen Sayın Dr. Őđretim Őyesi Murat TUNA hocama ve ayrıca yŸksek lisans eęitimim boyunca her konuda öneri ve eleŐtirileriyle yardımlarını gŸrdŸęŸm hocalarıma ve arkadaşlarıma en iten duygularıyla teŐekkŸrlerimi sunarım.

Tez alıŐmasının yapılmasında 18.FEN.BİL.50 numaralı proje ile maddi destek saęlayan Afyon Kocatepe Őniversitesi Bilimsel AraŐtırma Projeleri Koordinasyon birimine teŐekkŸr ederim.

TŸm hayatım boyunca maddi ve manevi desteklerinden dolayı aileme teŐekkŸr ederim.

Halil İbrahim ŐEKER
AFYONKARAHİSAR, 2019

İÇİNDEKİLER DİZİNİ

	Sayfa
ÖZET	i
ABSTRACT	iii
TEŞEKKÜR	v
İÇİNDEKİLER DİZİNİ.....	vi
SİMGELER ve KISALTMALAR DİZİNİ	viii
ŞEKİLLER DİZİNİ	x
ÇİZELGELER DİZİNİ.....	xiii
RESİMLER DİZİNİ	xiv
1. GİRİŞ	1
2. LİTERATÜR BİLGİLERİ	3
3. MATERYAL VE YÖNTEM	10
3.1 Alan Programlanabilir Kapı Dizileri (FPGA).....	10
3.2 Programlanabilir Mantık Cihazları	10
3.2.1 Basit Programlanabilir Mantık Devreleri	10
3.2.2 Karmaşık Programlanabilir Mantık Devre	12
3.2.3 FPGA Çipleri ve Genel Yapısı	13
3.2.3.1 Giriş- Çıkış Blokları.....	15
3.2.3.2 Yapılandırılabilir Mantıksal Bloklar.....	15
3.2.3.3 Ara Bağlantılar (Interconnections)	16
3.3 Kayan Noktalı Sayı Standardı.....	16
3.4 Sabit Noktalı Sayı Standardı.....	19
3.5 VHDL	21
3.5.1 VHDL Veri Nesneleri.....	22
3.5.2 VHDL Temel Tasarım Yapıları.....	23
3.6 Nümerik Analiz Yöntemleri	26
3.6.1 Euler Nümerik Analiz Yöntemi.....	26
3.6.2 Heun Nümerik Analiz Yöntemi.....	26
3.6.3 RK4 Nümerik Analiz Yöntemi.....	27
3.6.4 RK5-Butcher Nümerik Analiz Yöntemi.....	27
3.6.5 DP Nümerik Analiz Yöntemi	28
3.7 Kaos ve Kaos Analiz Yöntemleri	29
3.7.1 Lyapunov Üstelleri	30

3.7.2 Sistemin Faz Portrelerinin İncelenmesi	31
3.7.3 Zaman Serileri Analizi İncelemesi	32
3.8 SEA Kaotik Sistemi ve Kaos Analizi	32
3.9 SEA Kaotik Sisteminin Ayrıklaştırılmış Modelleri	35
3.9.1 Euler Tabanlı Ayrıklaştırılmış Modeli	35
3.9.2 Heun Tabanlı Ayrıklaştırılmış Modeli	36
3.9.3 RK4 Tabanlı Ayrıklaştırılmış Modeli	38
3.9.4 RK5-Butcher Tabanlı Ayrıklaştırılmış Modeli	40
3.9.5 DP Tabanlı Ayrıklaştırılmış Modeli	42
4. BULGULAR	46
4.1 Kayan Noktalı Sayı Tabanlı SEA Kaotik Sistemi Tasarımı	46
4.2 Kayan Noktalı Sayı Tabanlı SEA Kaotik Sistemi Tasarımının Sonuçları	49
4.3 Sabit Noktalı Sayı Tabanlı SEA Kaotik Sistemi Tasarımı ve Sonuçları	51
4.3.1 32-bit (16I-16Q) Sabit Noktalı Sayı Tabanlı SEA Kaotik Sistemi Tasarımı ve Sonuçları	51
4.3.2 28-bit (14I-14Q) Sabit Noktalı Sayı Tabanlı SEA Kaotik Sistemi Tasarımı ve Sonuçları	54
4.3.3 24-bit (12I-12Q) Sabit Noktalı Sayı Tabanlı SEA Kaotik Sistemi Tasarımı ve Sonuçları	56
4.3.4 20-bit (10I-10Q) Sabit Noktalı Sayı Tabanlı SEA Kaotik Sistemi Tasarımı ve Sonuçları	59
4.3.5 16-bit (8I-8Q) Sabit Noktalı Sayı Tabanlı SEA Kaotik Sistemi Tasarımı ve Sonuçları	61
4.4 FGPA Tabanlı SEA Kaotik Sistemlerin Hata Analizleri	64
5. TARTIŞMA ve SONUÇ	66
6. KAYNAKLAR	70
ÖZGEÇMİŞ	81

SİMGELER ve KISALTMALAR DİZİNİ

Simgeler

x	Kaotik sistem durum değişkenleri
y	Kaotik sistem durum değişkenleri
z	Kaotik sistem durum değişkenleri
a	Kaotik sistem parametreleri
b	Kaotik sistem parametreleri
c	Kaotik sistem parametreleri
d	Kaotik sistem parametreleri
e	Kaotik sistem parametreleri
f	Kaotik sistem parametreleri
t	Zaman
Λ	Lyapunov üstelleri
x_0	Kaotik sistem başlangıç şartı
y_0	Kaotik sistem başlangıç şartı
z_0	Kaotik sistem başlangıç şartı

Kısaltmalar

ASIC	Uygulamaya Özel Tümdevre (Application Specific Integrated Circuit)
CLB	Yapılandırılabilir Mantıksal Bloklar (Configurable Logic Blocks)
CMOS	Bütünleyici Metal Oksit Yarı İletken (Complementary Metal Oxide Semiconductor)
CPLD	Karmaşık Programlanabilir Mantık Cihaz (Complex Programmable Logic Device)
DSPs	Sayısal Sinyal İşlemciler (Digital Signal Processors)
DP	Dormand-Prince Nümerik Algoritması
Exp	Üs Bitleri (Exponent)
EEPROM	Elektrikle Silinip Programlanabilen Salt Okunur Bellek (Electrically Erasable Programmable Read Only Memory)
EEPLD	Elektriksel-Silinip Programlanabilir Lojik Aygıt (Electrically Erasable Programmable Logic Device)
EPROM	Silinip Programlanabilir Salt Okunur Bellek (Erasable Programmable Read Only Memory)
FPAAs	Alan Programlanabilir Analog Diziler (Field Programmable Analog Array)
FPGA	Alan Programlanabilir Kapı Dizileri (Field Programmable Gate Array)
GRSÜ	Gerçek Rasgele Sayı Üreteçleri
IEEE	The Institute of Electrical and Electronical Engineers
IEEE 754	Kayan Noktalı Sayı Formatı (IEEE Floating Point Number)
IOB	Giriş Çıkış Blokları (Input/Output Blocks)
IQ-Math	Sabit-Noktalı Sayı Formatı (Fixed-Point Number)

Kısaltmalar (Devam)

IP-core	Intellectual Properties core
ISE	Entegre Yazılım Ortamı (Integrated Software Environment)
LET	Lyapunov Exponent Toolbox
LUT	Değer Tablosu (Look-up table)
MLC	Murali Lakshmanan Chua
PAL	Programlanabilir Dizi Mantıkları (Programmable Array Logic)
PEEL	Programlanabilir Elektriksel Silinebilir Lojik (Programmable Electrically-Erasable Logic)
PLA	Programlanabilir Mantık Dizisi (Programmable Logic Array)
PLD	Programlanabilir Mantık Cihazları (Programmable Logic Device)
PLL	Faz Kilitli Döngü
PROM	Programlanabilir Salt Okunabilir Bellek
ROM	Read Only Memory (Salt Okunur Bellek)
RK4	Dördüncü Dereceden Runge-Kutta Algoritması
RK5	Beşinci dereceden Runge-Kutta algoritması
SPLD	Basit Programlanabilir Mantık Cihaz (Simple Programmable Logic Device)
SEA	Kendinden Uyarımlı Çeker (Self Excited Attractor)
VHDL	Çok Yüksek Hızlı Tümeleşik Devre Donanım Tanımlama Dili (Very High Speed Integrated Circuit Hardware Description Language)

ŞEKİLLER DİZİNİ

	Sayfa
Şekil 3.1 SPLD mimari yapısı.....	12
Şekil 3.2 CPLD mimari yapısı.	13
Şekil 3.3 FPGA genel yapısı.	15
Şekil 3.4 Örnek bir FPGA çipinin CLB yapısı.....	16
Şekil 3.5 32-bit IEEE 754-1985 kayan noktalı sayı standardı gösterimi.	17
Şekil 3.6 13 sayısının 32-bit IEEE 754-1985 kayan noktalı sayı standardı gösterimi. ...	18
Şekil 3.7 32-bit IEEE 754-1985 kayan noktalı sayı standardı dönüşüm programı ekran görüntüsü.	19
Şekil 3.8 Sabit noktalı sayı standardı gösterimi.	19
Şekil 3.9 99,1545 sayısının 8I-8Q sabit noktalı sayı formatına dönüşüm sonucu.	20
Şekil 3.10 VHDL tasarımın temel bölümleri.	23
Şekil 3.11 Tam toplayıcı VHDL kodları.	24
Şekil 3.12 Tam toplayıcı en üst seviye blok şeması.....	25
Şekil 3.13 Tam toplayıcı devresine ait ikinci seviye blok diyagramı.	25
Şekil 3.14 Tam toplayıcı Xilinx ISE 14.7 simülasyon sonuçları.	25
Şekil 3.15 Örnek Lyapunov üstelleri grafiği.	31
Şekil 3.16 Altın orana sahip kaotik sistemin faz portreleri.	31
Şekil 3.17 Altın orana sahip kaotik sistemin zaman serisi analizi.	32
Şekil 3.18 DP algoritma tabanlı SEA kaotik sistemine ait zaman serileri.	33
Şekil 3.19 DP algoritma tabanlı SEA kaotik sistemine ait faz portreleri.	34
Şekil 3.20 SEA kaotik sistemin Lyapunov üstelleri.....	34
Şekil 3.21 Euler algoritması tabanlı SEA kaotik sistemin zaman serisi.	36
Şekil 3.22 Euler algoritması tabanlı SEA kaotik sistemin faz portreleri.	36
Şekil 3.23 Heun tabanlı SEA kaotik sistemin zaman serisi analizi.....	37
Şekil 3.24 Heun tabanlı SEA kaotik sisteminin faz portreleri.	38
Şekil 3.25 RK4 tabanlı SEA kaotik sistemin zaman serisi analizi.....	40
Şekil 3.26 RK4 tabanlı SEA kaotik sistemin faz portreleri.....	40
Şekil 3.27 RK5-Butcher tabanlı SEA kaotik sistemin x, y ve z zaman serisi analizi. ...	42
Şekil 3.28 RK5-Butcher tabanlı SEA kaotik sistemine ait faz portreleri.....	42
Şekil 4.1 FPGA-tabanlı SEA kaotik osilatörün en üst seviye blok şeması.	46
Şekil 4.2 FPGA-tabanlı kaotik osilatörün ikinci alt seviye blok şeması.	47
Şekil 4.3 DP-tabanlı SEA kaotik osilatörün üçüncü seviye blok şeması.	48

Şekil 4.4 DP-tabanlı SEA Kaotik osilatörün IEEE 754-1985 32-bit kayan noktalı sayı standartlarındaki simülasyon sonuçları.	49
Şekil 4.5 32-bit kayan nokta sayı tabanlı SEA kaotik osilatörün zaman serileri.	50
Şekil 4.6 32-bit kayan nokta sayı tabanlı SEA kaotik osilatörün faz portreleri.	50
Şekil 4.7 16I-16Q FPGA-tabanlı SEA kaotik osilatörün en üst seviye blok diyagramı. .	51
Şekil 4.8 DP-tabanlı 16I-16Q IQ-Math sabit sayı standardındaki SEA kaotik osilatörün Xilinx ISE 14.7 simülasyon sonuçları.	52
Şekil 4.9 FPGA üzerinde 32-bit (16I-16Q) DP-tabanlı SEA kaotik osilatörün zaman serileri.	53
Şekil 4.10 FPGA üzerinde 32-bit (16I-16Q) DP-tabanlı SEA kaotik osilatörün faz portreleri.	53
Şekil 4.11 14I-14Q FPGA-tabanlı SEA kaotik osilatörün en üst seviye blok diyagramı.	54
Şekil 4.12 DP-tabanlı 14I-14Q IQ-Math sabit sayı standardındaki SEA kaotik osilatörün Xilinx ISE 14.7 simülasyon sonuçları.	55
Şekil 4.13 FPGA üzerinde 28-bit (14I-14Q) DP-tabanlı SEA kaotik osilatörün zaman serileri.	56
Şekil 4.14 FPGA üzerinde 28-bit (14I-14Q) DP-tabanlı SEA kaotik osilatörün faz portreleri.	56
Şekil 4.15 12I-12Q FPGA-tabanlı SEA kaotik osilatörün en üst seviye blok diyagramı.	57
Şekil 4.16 DP-tabanlı 12I-12Q IQ-Math sabit sayı standardındaki SEA kaotik osilatörün Xilinx ISE 14.7 simülasyon sonuçları.	57
Şekil 4.17 FPGA üzerinde 24-bit (12I-12Q) DP-tabanlı SEA kaotik osilatörün zaman serileri.	58
Şekil 4.18 FPGA üzerinde 24-bit (12I-12Q) DP-tabanlı SEA kaotik osilatörün faz portreleri.	59
Şekil 4.19 10I-10Q FPGA-tabanlı SEA kaotik osilatörün en üst seviye blok diyagramı.	59
Şekil 4.20 DP-tabanlı 10I-10Q IQ-Math sabit sayı standardındaki SEA kaotik osilatörün Xilinx ISE 14.7 simülasyon sonuçları.	60
Şekil 4.21 FPGA üzerinde 20-bit (10I-10Q) DP-tabanlı SEA kaotik osilatörün zaman serileri.	61
Şekil 4.22 FPGA üzerinde 20-bit (10I-10Q) DP-tabanlı SEA kaotik osilatörün faz portreleri.	61
Şekil 4.23 8I-8Q FPGA-tabanlı SEA kaotik osilatörün en üst seviye blok diyagramı. .	62
Şekil 4.24 DP-tabanlı 8I-8Q IQ-Math sabit sayı standardındaki SEA kaotik osilatörün Xilinx ISE 14.7 simülasyon sonuçları.	62

Şekil 4.25 FPGA üzerinde 16-bit (8I-8Q) DP-tabanlı SEA kaotik osilatörün zaman serileri.....	63
Şekil 4.26 FPGA üzerinde 16-bit (8I-8Q) DP-tabanlı SEA kaotik osilatörün faz portreleri.....	64

ÇİZELGELER DİZİNİ

Sayfa

Çizelge 2.1 Literatürdeki FPGA-tabanlı kaotik osilatör tasarımları ve özellikleri.	8
Çizelge 4.1 DP-tabanlı SEA kaotik sistemlerin IEEE 754-1985 32-bit kayan noktalı sayı standartlarındaki FPGA çip kullanım istatistikleri.	49
Çizelge 4.2 DP-tabanlı 32 bit (16I-16Q) SEA kaotik sisteminin FPGA çip istatistikleri.	52
Çizelge 4.3 DP-tabanlı 28 bit (14I-14Q) SEA kaotik sisteminin FPGA çip istatistikleri.	55
Çizelge 4.4 DP-tabanlı 24 bit (12I-12Q) SEA kaotik sisteminin FPGA çip istatistikleri.	58
Çizelge 4.5 DP-tabanlı 20 bit (10I-10Q) SEA kaotik sisteminin FPGA çip istatistikleri.	60
Çizelge 4.6 DP-tabanlı 16 bit (8I-8Q) SEA kaotik sisteminin FPGA çip istatistikleri. .	63
Çizelge 4.7 FPGA tabanlı sabit noktalı sayı formatındaki kaotik osilatör tasarımlarının MSE ve RMSE hata analizleri.	65
Çizelge 5.1 Farklı sayısal platformlar kullanılarak gerçekleştirilen kaotik osilatör tasarımları ve özellikleri.	68

RESİMLER DİZİNİ

Sayfa

Resim 3.1 Xilinx Virtex-6 FPGA kartı.	14
---	----

1. GİRİŞ

Kâinata var olan tüm sistemler doğrusal olmayan (nonlinear) bir yapıya sahiptir. Böyle bir yapıya sahip sistemlerin birbiriyle olan etkileşimlerini inceleyen ve bu sistemleri modellemeye çalışan bilim dalı nonlinear bilim olarak ifade edilmektedir. Kaos bilimi ya da kaotik sistemler son zamanlarda üzerinde birçok araştırmaların ve çalışmaların yapıldığı nonlinear bilim alanlarından biridir. Kaos ve kaotik sistemler karmaşık davranışlar sergileyen veya düzensizlik gibi algılanmasına rağmen aslında kararlı bir hal davranışı sergileyen ayrıca kendine özgü bir iç düzene sahip sistemlerdir. Başlangıç koşullarına hassas bağımlılık, zaman serilerinin gürültü benzeri sınırsız sayıda periyodik olmayan davranışlar içermesi ve sistemde var olan parametre değerlerine kaotik sistemin aşırı bağımlı olması bu sistemlerin başlıca belirgin özellikleri arasında verilebilir. Ayrıca başlangıç değerlerinde ya da sistem parametrelerinde yapılacak en küçük değişikliğin sistem çıkışında öngörülemeyen tepkilere ve tahmin edilemeyen değişimlere neden olabilmektedir. Yukarıda verilen önemli özelliklerinden dolayı kaotik sistemler mühendisliğin birçok uygulama alanında kullanılmaktadır. Bu uygulama alanlarına endüstriyel kontrol Azar *et al.* (2017), Rabah *et al.* (2018), biyomedikal Xiong *et al.* (2014), Pandey *et al.* (2018), haberleşme Kaddoum *et al.* (2016), Ren *et al.* (2017), görüntü işleme Chen *et al.* (2004), Pareek *et al.* (2006), kriptoloji Pehlivan vd. (2007), Akkaya vd. (2018), yapay sinir ağları Alçın (2017), Lamamra *et al.* (2017), rasgele sayı üreteçleri Avaroğlu vd. (2015), Rozic *et al.* (2015), elektromanyetik Manfredi and Canavero (2015), gibi alanlar örnek olarak verilebilir. Kaos bilimi ilk olarak meteoroloji uzmanı olan ve aynı zamanda matematikçi Edward Norton Lorenz tarafından 1963 yılında ortaya çıkarılmıştır. Lorenz hava durumunu daha önceden belirleyebilmek için bir diferansiyel denklem modeli oluşturmuştur. Bu modeli kullanarak denklemleri çözmektedir. Bu modele fazla uğraşmamak için eski çözüm sonuçlarını yuvarlayarak sisteme verdiğinde bir önceki çözümünden çok daha farklı sonuç ürettiğini görmüştür. Bulduğu bu sonuçla Lorenz, yaptığı çalışmada başlangıç şartlarındaki en küçük bir değişikliğin sistem cevabı olarak öngörülemeyen, tahmin edilemeyen sonuçlara neden olduğu göstermiş ve kaotik sistemin temellerini atmıştır (Pehlivan 2007). Daha sonra literatüre Lorenz, Rössler, Rikitake, Burke-Shaw, Pehlivan-Wei, Sprott, Abooe, Chua, Van der Pol, Deng, Rucklidge ve Arneodo vb. gibi birçok kaotik osilatör sistemleri

sunulmuştur. Tüm bu yapılar için gereken en temel yapı kaotik sinyali üreten kaotik osilatör ünitesidir. Kaotik sistemlerin donanımsal olarak gerçekleştirilmesi analog ve sayısal tabanlı platformlarda yapılabilmektedir. Analog tabanlı tasarlanan kaotik sistemlerin sıcaklıkla analog elemanların değerlerinin değişmesi, gürültü ve kullanım ömrü ile sistemin sonuçları değişebildiği gibi birçok dezavantajları bulunmaktadır. Sayısal tabanlı sistemler sıcaklığa daha az duyarlı ve gürültüden daha az etkilenmesi gibi özelliklerinden dolayı sayısal tabanlı kaotik sistemler analog tabanlı olana göre daha avantajlıdır. Kaotik sistemler sayısal tabanlı platformlar ile Sayısal İşaret İşlemciler (Digital Signal Processors (DSPs)), Uygulamaya Özel Tümüleşik Devreler (Application Specific Integrated Circuits (ASICs)) ve Alan Programlanabilir Kapı Dizileri (Field Programmable Gate Array (FPGA)) gibi donanımlarda gerçekleştirilebilmektedir. ASIC tabanlı kaotik sistemlerin diğer donanımlara göre daha yüksek frekansta çalışabilme özelliği bulunmasına rağmen esnek yapıya sahip olmaması, ilk tasarım ve test süreçlerinin oldukça maliyetli olması gibi dezavantajları bulunmaktadır. İşlemleri sıralı (sequential) olarak yapan DSP yongası ise karmaşık matematiksel denklemleri gerçekleştirmek için geliştirilmiş bir donanım yapısına sahiptir. DSP yongaları işlemleri sıralı bir şekilde gerçekleştirmesinden dolayı sistemin cevap verme süresi uzamakta ve çalışma frekansı düşmektedir. FPGA çipleri paralel işlem yapabilme, tekrar tekrar programlanabilme ve yüksek frekansta çalışabilme özelliklerine sahiptir. FPGA çipinin tekrar tekrar programlanabilmesi ve esnek bir yapıya sahip olması ASIC çiplerinden daha üstün kılmaktadır. Ayrıca paralel işlem yapabilme özelliği sayesinde DSP çiplerinden daha hızlı sürede işlem yapabilmektedir. Tüm bu analog ve sayısal tabanlı yapılan kaotik sistem tasarımları literatür bilgileri kısmında detaylı örnekler verilerek sunulmuştur (Koyuncu 2014).

2. LİTERATÜR BİLGİLERİ

Güncel literatür taraması yapıldığında kaotik sistemler analog ve sayısal tabanlı olmak üzere bir çok farklı platformlarda modellenebilmektedir. Bu kaotik osilatör tasarımlarına örnek olarak;

Hidalgo ve arkadaşları tarafından sunulan çalışmada, güvenli haberleşme sistemleri için kaotik bir dinamik sistem tarafından üretilen, sadece vericiyle eşleşen bir alıcının bilgi sinyali kodunu çözebileceği şekilde kaotik sistem tasarımı DSP platformu üzerinde gerçekleştirilmiştir (Hidalgo *et al.* 2001).

Ren ve arkadaşları tarafından sunulan çalışmada, Chen kaotik sistemi TI TMS320C6713 DSP çipi üzerinde gerçekleştirilmiştir. Yapılan bu çalışmada güvenli haberleşmede kullanılmak üzere ses sinyalleri kaotik sinyal ile şifrelenmiş ve kaotik iletişim şemasında kullanılmıştır. Önerilen güvenli hiper-kaotik iletişim sistemi için senkronizasyon modellemesi yapılmıştır (Ren *et al.* 2017).

Vaidyanathan ve arkadaşları yaptıkları araştırmada, 3D kaotik sistemini denge, Lyapunov üstelleri ve Kaplan-Yorke boyutu gibi temel özellikleri bakımından analiz etmişlerdir. Ayrıca Matlab kullanılarak simüle edilen kaotik sisteme ait faz portreleri sunulmuş ve kaotik sistemin gerçek zamanlı elektronik devre gerçekleştirilmesi için LabVIEW tabanlı tasarımı gerçekleştirilmiştir (Vaidyanathan *et al.* 2015).

Sundarapandian ve arkadaşlarının sundukları çalışmada, tek bir kübik nonlineariteye sahip yeni üç boyutlu otonom kaotik sistemi tanıtılmıştır. Kaotik sisteme ait dinamik davranış, denge, Lyapunov üs spektrumu gibi yöntemleri analitik ve sayısal olarak incelenmiştir. Ayrıca kaotik sistemin Orcad-PSpice programı kullanılarak elektronik devresi gerçekleştirilmiştir. Simülasyonlar ve deneysel sonuçlar incelendiğinde başarılı sonuçlar elde edildiği gözlenmiş ve kaos tabanlı mühendislik uygulamalarında kullanılabileceğini ifade edilmiştir (Sundarapandian *et al.* 2012).

Pareschi ve arkadaşları güvenli haberleşmede kullanılabilecek gerçek rasgele sayı

üretici tasarlamışlardır. Tasarım entropi kaynağı olarak kaotik devre kullanan gerçek rasgele sayı üretici tasarımını bütünleyici metal oksit yarı iletken (Complementary Metal Oxide Semiconductor (CMOS)) üzerinde gerçekleştirmişlerdir. İki prototip 0.35 μm ve 0.18 μm CMOS teknolojisinde tasarlanmış ve prototiplerin sırasıyla 40 Mbit/s ve 100 Mbit/s bit üretim hızına sahip olduğunu ifade edilmiştir (Pareschi *et al.* 2010).

Murillo-Escobar ve arkadaşlarının yaptıkları çalışmada, yüksek güvenli haberleşme için kaos tabanlı şifreleme algoritması 32-bit mikroişlemci (micro-controller (μC)) tabanlı olarak modellemişlerdir. Çip üzerindeki düşük hafıza ve düşük frekansta çalışmasını dezavantaj olarak belirtmişlerdir. Yapılan çalışma test edilmiş ve doğrulanmıştır. Ayrıca mükemmel şifreleme özelliği sunduğu için kriptolojide kullanılabileceğini belirtmişlerdir (Murillo-Escobar *et al.* 2016).

Chiu ve arkadaşları tarafından sunulan çalışmada, üç boyutlu Lorenz kaotik sisteminin birkaç direnç ve mikroişlemci kullanarak tasarımı gerçekleştirilmiştir. Tasarımı Euler nümerik algoritması kullanarak modellemişlerdir (Chiu *et al.* 2013).

Akgül ve arkadaşları tarafından sunulan çalışmada, denge noktaları olmayan kaotik osilatör Labview-tabanlı FPGA yongası ve 32-bit kayan noktalı sayı formatına göre FPGA yongası üzerinde elektronik devre uygulamasını gerçekleştirmişler. Modelleme sonucu elde edilen faz portrelerini ve zaman serisi analizlerini sunmuşlardır. Labview-tabanlı sonuçlara ile FPGA-tabanlı sonuçlar karşılaştırılmış ve başarılı sonuçlar elde edildiği belirtilmiştir (Akgül vd. 2016).

Çiçek ve arkadaşları sundukları çalışmada, dört parametrelili ve dört doğrusal olmayan terime sahip yeni bir üç boyutlu kaotik sistemi Matlab tabanlı ve Orcad-PSpice programlarını kullanarak modellemişlerdir. Kaotik sisteme ait zaman serileri, faz portreleri, Lyapunov üstelleri analizleri verilmiştir. Çalışmanın son kısmında ise aktif kontrol senkronizasyonu ve güvenli bir kaotik maskeleyme iletişim uygulaması tasarımı sunulmuştur (Çiçek vd. 2016).

Rajagopalan ve arkadaşları sundukları çalışmada, CMOS Boolean kaotik üreticisine göre

gerçek rasgele sayı üretici tasarımının ASIC yaklaşımı 45 nm CMOS teknolojisine bağlı Cadence virtuoso aracı vasıtasıyla modellenmesi yapılmıştır. ASIC tabanlı tasarımı gerçekleştirilen rasgele sayı üreticinin kriptografik haberleşmede kullanılabileceği vurgulanmıştır (Rajagopalan *et al.* 2017).

Xu ve arkadaşlarının sunduğu çalışmada, SEA (self-excited attractor) kaotik sistemini Raspberry Pi 3 ve Orcad-PSpice üzerinde gerçekleştirmişlerdir. Sinyal gizleme uygulaması gerçekleştirmek için SEA kaotik osilatör kullanılarak RNG (rasgele sayı üreticileri) tasarlanmıştır. Tasarlanan RNG'lerin performanslarını değerlendirmek için uluslararası test olan NIST-800-22 testine tabi tutulmuş ve sonuçların başarılı olduğu gözlemlenmiştir. Daha sonra direnç, kapasitör, opamp gibi analog devre elemanları kullanılarak hem Orcad-PSpice programı üzerinde modellemişler hem de gerçek zamanlı olarak devreyi tasarlamışlardır. Kaotik sisteme ait faz portrelerinin simülasyon ve osiloskop şekilleri verilmiştir (Xu *et al.* 2018).

Kılıç sunduğu çalışmada, özerk Chua devresi, adsız özerk MLC (Murali Lakshmanan Chua) devresi ve PLL (Faz Kilitli Döngü) devresi olmak üzere üç farklı kaotik osilatör tasarım örneğini Alan Programlanabilir Analog Dizi (Field-Programmable Analog Array (FPAA)) çipi üzerinde modellemiştir. FPAA dinamik yeniden yapılandırılabilir analog fonksiyonlar içeren programlanabilir bir cihazdır. FPAA yongası üzerinde gerçekleştirilmesinden sonra elde edilen faz portreleri ve zaman serisi analizleri sunulmuştur (Kılıç 2010).

Tuna ve arkadaşlarının sundukları çalışmada, sürekli zamanlı yeni bir kaotik sistemi Matlab tabanlı ve FPGA çipi üzerinde gerçekleştirmişlerdir. Kaotik sistem 32-bit IEEE 754-1985 kayan nokta formatına uygun olarak VHDL dilinde Heun algoritması kullanılarak modellenmiştir. Tasarlanan kaotik sistem Xilinx Virtex-6 FPGA yongası üzerinde sentezlenmiş ve test edilmiştir. Kaotik osilatörün çip istatistikleri incelenmiş ve çalışma frekansı 390 MHz olarak elde edildiği ifade edilmiştir. Son olarak Matlab tabanlı sonuçlar ile FPGA tabanlı sonuçlar karşılaştırılmış başarılı sonuçlar elde edildiği belirtilmiştir (Tuna vd. 2015).

Zhang sunduğu çalışmada, Lorenz kaotik osilatörü Xilinx System Generator kullanarak FPGA yongası üzerinde gerçeklemiştir. Tasarım modelleri 32-bit ve 16-bit sabit noktalı sayı formatına uygun olarak oluşturulmuştur. Her iki tasarım içinde simülasyon sonuçları, çip istatistikleri ve zaman serileri analizi çalışmada sunulmuştur. Matlab Simulink tabanlı sonuçlar ile FPGA tabanlı sonuçlar karşılaştırılmış ve başarılı sonuçlar elde edildiği ifade edilmiştir (Zhang 2017).

Alçın ve arkadaşları sundukları çalışmada, üç boyutlu Jerk kaotik osilatör sistemini RK4 nümerik algoritması kullanarak FPGA üzerinde gerçekleştirmişlerdir. Jerk kaotik osilatör tasarımı 32-bit (16I-16Q) IQ-Math sabit sayı formatına uygun olarak VHDL dilinde kodlanmış ve Xilinx Virtex-6 FPGA yongasına sentezlenmiştir. Daha sonra tasarlanan RK4 tabanlı kaotik sistem ile literatürde var olan yapay sinir ağları tabanlı Jerk kaotik sistemi arasında karşılaştırmalı analizler yapılmıştır. Çalışma frekansı 373,128 MHz olarak elde edildiği ifade edilmiştir. RK4 tabanlı kaotik osilatör hem daha yüksek çalışma frekansına sahip hem de daha az çip kaynak tüketimine sahip olduğu belirtilmiştir (Alçın vd. 2018).

Dursun ve arkadaşları sundukları çalışmada, Van der Pol kaotik sistemini FPGA çipleri üzerinde gerçekleştirmişlerdir. Kaotik sistem Euler nümerik algoritması kullanılarak 32-bit IEEE 754-1985 kayan noktalı sayı formatına uygun olarak VHDL dilinde Xilinx Virtex-6 çipi üzerine sentezlenmiştir. Place & Route işlemlerinden sonra elde edilen çip istatistikleri verilmiştir. Ayrıca çalışma frekansı 498,728 MHz olarak belirtilmiştir (Dursun ve Kaşifoğlu 2018).

Rashtchi ve arkadaşları sundukları çalışmada, Duffing kaotik sistemini Cyclone IV GX FPGA çipi üzerinde gerçekleştirmişlerdir. Tasarım Euler ve RK4 olmak üzere iki farklı nümerik analiz yöntemi ile modellenmiş ve VHDL dilinde kodlanmıştır. Maksimum çalışma frekansı Euler tabanlı kaotik sistem 152,28 MHz, RK4 tabanlı kaotik sistem 155,69 MHz olarak elde edilmiştir. Ayrıca Euler tabanlı sistem RK4 tabanlı sisteme göre daha az çip kaynakları tükettiği belirtilmiştir (Rashtchi and Nourazar 2015).

Koyuncu ve arkadaşları sundukları çalışmada Sundarapandian-Pehlivan kaotik sistemini

Orcad-PSpice ve FPGA olmak üzere iki ayrı platformda modellemişlerdir. Kaotik sistem Xilinx Virtex-6 FPGA yongası üzerinde RK4 nümerik algoritması kullanarak gerçekleştirmişlerdir. Ayrıca tasarım 32-bit IEEE 754-1985 kayan noktalı sayı formatına göre VHDL dilinde kodlanmıştır. Tasarıma ait faz portreleri, zaman serisi analizi ve Lyapunov üstelleri verilmiştir. Kaotik sistemin çalışma frekansı 293,815 MHz olarak elde edilmiştir (Koyuncu ve Özcerit 2017).

Koyuncu ve arkadaşları sundukları çalışmada, Burke-Shaw kaotik sistemini RK5-Butcher nümerik algoritması kullanılarak FPGA tabanlı ve hem Orcad-PSpice programı üzerinden hem de analog devre elemanları kullanılarak gerçek zamanlı uygulamaları gerçekleştirmişlerdir. FPGA tabanlı kaotik sistem tasarımı 32-bit IEEE 754-1985 kayan noktalı sayı formatına uygun olarak Xilinx Virtex-6 FPGA yongası üzerine sentezlemişlerdir. Place & Route işlemlerinden sonra elde edilen çip istatistiklerini sunmuşlardır. Burke-Shaw kaotik sistemi çalışma frekansı 373,094 MHz olarak belirtilmiştir (Koyuncu vd. 2013).

Tuna ve arkadaşları sundukları çalışmada, PC kaotik sistemini FPGA çipi üzerinde Euler, Heun, RK4 ve RK5-Butcher nümerik algoritmalarını kullanarak 32-bit (16I-16Q) IQ-Math sabit noktalı sayı standardına uygun olarak modellemişlerdir. Farklı nümerik algoritmalarla tasarımı gerçekleştirilen PC kaotik sistemi VHDL dili kullanarak kodlanmış ve Xilinx Virtex-6 çipi üzerine sentezlenmiştir. Place & Route işlemlerinden sonra elde edilen çip istatistikleri ve çalışma frekansları sunulmuştur. Sabit noktalı sayı formatına uygun Euler tabanlı PC kaotik sistemi en az çip kullanımına sahip olmakla birlikte 464,688 MHz çalışma frekansına sahip olduğu görülmüştür. RK5-Butcher tabanlı PC kaotik sistemi ise diğer nümerik algoritmalarla tasarlanan kaotik sistemlerden daha fazla çip kaynağı tükettiği ve 436,143 MHz çalışma frekansına sahip olduğu ifade edilmiştir. Ayrıca RK5-Butcher tabanlı kaotik sistem diğer farklı nümerik algoritmalarla tasarlanan kaotik sistemlerden daha hassas sonuçlar ürettiği belirtilmiştir (Tuna vd. 2018).

Koyuncu ve arkadaşları sundukları çalışmada, Pehlivan-Wei (PW) kaotik sistemini Euler, Heun ve RK4 olmak üzere üç farklı algoritma FPGA yongası üzerinde modellemişlerdir. PW kaotik sistemi 32-bit IEEE 754-1985 kayan noktalı sayı

standartlarına uygun olarak VHDL dilinde kodlanmış ve Xilinx Virtex-6 çipi üzerine sentezlemiştir. Place & Route işlemlerinin ardından elde edilen çip istatistiklerini, çalışma frekanslarını ve faz portrelerini sunulmuştur. Euler tabanlı PW kaotik sisteminin en az çip kaynak tüketimine ve 436,143 MHz çalışma frekansına sahip olduğu belirtilmiştir. Heun tabanlı PW kaotik sistemi, Euler tabanlı sistemden daha fazla ama RK4 tabanlı sistemden daha az çip kaynağı tükettiği ve 463,688 MHz çalışma frekansına sahip olduğunu ifade edilmiştir. RK4 tabanlı kaotik sistem ise en fazla çip kaynağını tükettiği ve 373,094 MHz çalışma frekansına sahip olduğu belirtilmiştir. Son olarak da RMSE hata analizleri sunulmuştur (Koyuncu vd. 2014). Son zamanlarda literatüre sunulan FPGA tabanlı Euler, Heun, RK4 ve RK5-Butcher nümerik algoritmaları kullanılarak tasarımı gerçekleştirilen kaotik sistemler Çizelge 2.1’de verilmiştir.

Çizelge 2.1 Literatürdeki FPGA-tabanlı kaotik osilatör tasarımları ve özellikleri.

Literatürde yapılan çalışmalar	Kullanılan kaotik osilatör	Kullanılan yapı	Kullanılan sayı standardı	Kullanılan platform	Çalışma Frekansı (MHz)
E. Tlelo-Cuautle <i>et al.</i> (2015)	Multi-scroll	Euler ve RK4	--	XC3S1000-5FT256 FPGA Spartan-3	70.943
Azzaz <i>et al.</i> (2013)	3B Hybrid	Euler	32 bits (16I-16Q) Fixed-point	Xilinx Virtex-II XC2VP30FFG896	38.86
Tuna vd. (2016)	Altın oran	Heun	32-bit (16I-16Q) Fixed point	Virtex-6	406.736
Merah <i>et al.</i> (2013)	Lorenz	RK4	32 bits (12I-20Q) Fixed-point	Xilinx Spartan-3	18
Koyuncu (2018)	Van der Pol	Yapay sinir ağları	32 Bit IEEE-754 Floating-point	Xilinx Virtex-6 XC6VCX75T	362
Rajagopal <i>et al.</i> (2013)	Kaotik chameleon	RK5-Butcher	32-bit IEEE-754-1985 Floating point	Xilinx Virtex-6 XC6VLX240T-1FF1156	325.759
Tolba <i>et al.</i> (2017)	Liu	Grünwald-Letnikov	--	Xilinx Virtex-5 XC5VLX50T	137.561
Alçın vd. (2019)	Pehlivan-uyaroğlu	Yapay sinir ağları	32-bit IEEE-754-1985 Floating point	Xilinx Virtex-6 XC6VCX240T	231.616
Tuna vd. (2019)	Lü-Chen	Heun	32-bit IQ-Math Fixed point	Xilinx Virtex-5 XC6VLX75T-3FF484	464.688

Literatüre sunulan bütün bu kaotik tabanlı çalışmaların en temel yapısı olan kaotik

sinyali üreten bir kaotik osilatör yapısıdır. Bu kaotik osilatör yapısı diferansiyel denklemlerle ifade edilmektedir. Bu diferansiyel denklemler Euler, Heun, RK4, RK5-Butcher ve Dormand-Prince (DP) nümerik algoritmaları aracılığıyla çözülebilmektedir. Güncel literatür taraması yapıldığında Euler, Heun, RK4 ve RK5-Butcher nümerik algoritmaları kullanılarak yapılan kaotik osilatör tasarımları görülmektedir. Ama diğer nümerik algoritmalara göre daha hassas çözüm üreten DP nümerik algoritmasına rastlanmamıştır. Sunulan literatür çalışması ve çizelge 1'de belirtilen yöntemlerden farklı olarak daha önce üzerinde FPGA tabanlı çalışma yapılmamış olan SEA kaotik sistemi seçilmiş ve DP nümerik algoritması kullanılarak FPGA üzerinde modellenmiştir. Tasarımı gerçekleştirilen SEA kaotik sistemi 32-bit (16I-16Q), 28-bit (14I-14Q), 24-bit (12I-12Q), 20-bit (10I-10), ve 16-bit (8I-8Q) IQ-Math sabit noktalı sayı (fixed point number) standartlarında olmak üzere beş farklı sayı formatındaki kaotik sistem tasarımları VHDL dili kullanılarak kodlanmıştır. Ayrıca SEA kaotik sistemi FPGA üzerinde 32-bit IEEE 754-1985 kayan noktalı sayı standardına uygun olarak tasarımı gerçekleştirilmiştir.

3. MATERYAL VE YÖNTEM

3.1 Alan Programlanabilir Kapı Dizileri (FPGA)

Alan programlanabilir kapı dizileri (Field Programmable Gate Array (FPGA)) tasarımcının ihtiyaç duyduğu mantıksal fonksiyona göre donanım yapısı tekrar tekrar programlanabilen sayısal tümleşik devrelerdir. FPGA çipleri genel olarak ara bağlantılar, giriş çıkış pinleri ve mantık bloklarından oluşmaktadır. Tasarımcı FPGA içerisindeki mantık bloklarını, ara bağlantıları ve giriş-çıkış bağlantılarını ihtiyacı olan fonksiyona göre programlayabilmektedir. FPGA teknolojisi paralel işlem yapabilme, yüksek frekansta çalışabilme ve tekrar tekrar programlanabilme özelliklerinden dolayı son yıllarda popülerliği giderek artmıştır. FPGA çiplerinin gelişim süreci programlanabilir mantık elemanları ile başlamaktadır (Taşçı 2011).

3.2 Programlanabilir Mantık Cihazları

Programlanabilir mantık cihazları (Programmable Logic Device (PLD)) temelde AND, OR, NOT mantık kapıları olan ve flip-floplardan oluşan yapılardır. PLD'ler üretim sırasında belirli bir işleve ya da göreve tanımlı değillerdir. Bu nedenden dolayı elektronik devrelerde kullanılacağı zaman kesinlikle bir görev ya da işlev için yapılandırılmalıdır (Taşçı 2011).

PLD'ler üç kısma ayrılmaktadır.

- Basit PLD'ler (Simple PLD (SPLD))
- Karmaşık PLD'ler (Complex PLD (CPLD))
- Alan Programlanabilir Kapı Dizileri

3.2.1 Basit Programlanabilir Mantık Devreleri

SPLD içerisinde az sayıda mantık hücresi bulunduran PLD'lerdir. Bir SPLD aygıtında genellikle 4 ile 22 arasında programlanabilir hücre bulunmaktadır. SPLD'ler genel olarak üç ana kısımda incelenebilir (Çetin 2014).

- Programlanabilir Salt Okunur Bellek (Programmable Read Only Memory (PROM))

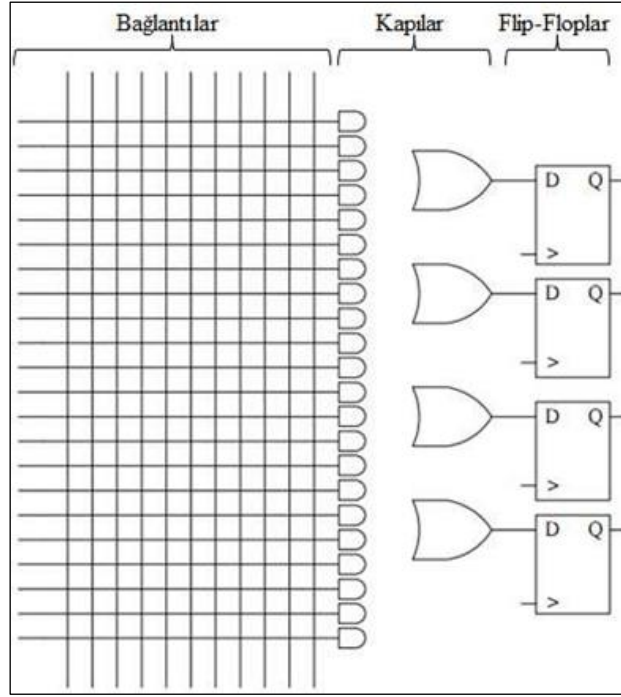
ROM (Read Only Memory) ilk programlanabilen salt okunur bellek adı verilen devrelerdir. PROM ise programlanabilir ROM aygıtlarına verilen isimdir. PROM ile ROM aynı özelliklere sahip olmasına rağmen PROM'lar üretim sırasında programlanmak zorunda değildir. Ayrıca PROM'lar bir kez programlanabilir ve daha sonra defalarca üzerinden okuma yapılabilir. PROM çeşitlerine örnek olarak EPROM (Erasable Programmable Read Only Memory-Silinip Programlanabilir Yalnızca Salt Okunur Bellek) ve EEPROM (Electrically Erasable Programmable Read Only Memory-Elektriksel Olarak Silinip Programlanabilen Salt Okunur Bellek) verilebilir. Arasındaki fark EPROM elektrikle programlanabilen ve ultraviyole ışıkla silinebilen, EEPROM ise elektrik ile yazılıp elektrik ile silinebilen donanımlardır.

- Programlanabilir Lojik Diziler (Programmable Array Logic (PAL))

PAL yapısı programlanabilen AND kapı dizilerinden ve sabit çıkışa sahip OR kapılarından oluşmaktadır (Akpolat 2015).

- Genel Dizi Mantığı (GAL)

Genel dizi mantığı (GAL), temelde PAL cihazı benzer özelliklere sahip olmasına rağmen arasındaki en belirgin iki fark; GAL ünitesinin tekrar programlanabilmesi ve bu ünitenin programlanabilir çıkış devrelerine sahip olmasıdır (Brown and Rose 1996, Gürsoy 2016). Şekil 3.1'de SPLD mimari yapısı verilmiştir (Çetin 2014).

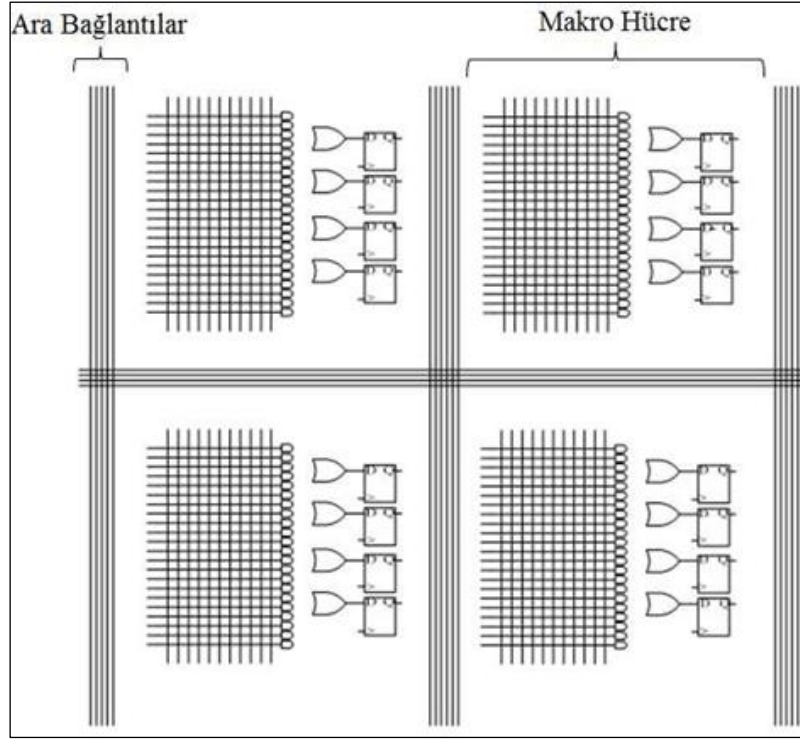


Şekil 3.1 SPLD mimari yapısı.

3.2.2 Karmaşık Programlanabilir Mantık Devre

CPLD mimarisi SPLD'lerin karmaşık devrelerde tasarım zorluğu ve yetersiz kalması durumunda birçok SPLD bloklarının bir araya getirilmesiyle oluşturulmuştur. Aynı zamanda bu bloklar arasında haberleşme mümkündür. Literatür incelendiğinde CPLD dört farklı grupta sınıflandırılabilir. Şekil 3.2'de CPLD mimari yapısı verilmiştir (Çetin 2014, Gürsoy 2016).

- Silinebilir Programlanabilir Lojik Aygıt (Erasable PLD)
- Programlanabilir Elektriksel Silinebilir Lojik (Programmable Electrically-Erasable Logic (PEEL))
- Elektriksel-Silinebilir Programlanabilir Lojik Aygıt (Electrically-Erasable PLD (EEPLD))
- Çoklu Dizi Matrisi, (Multi Array Matrix)

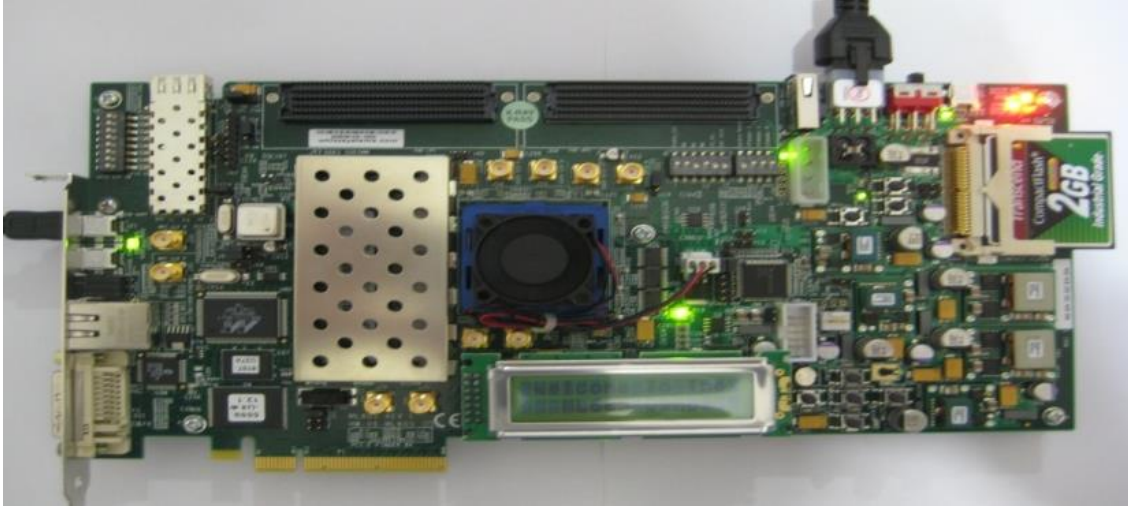


Şekil 3.2 CPLD mimari yapısı.

3.2.3 FPGA Çipleri ve Genel Yapısı

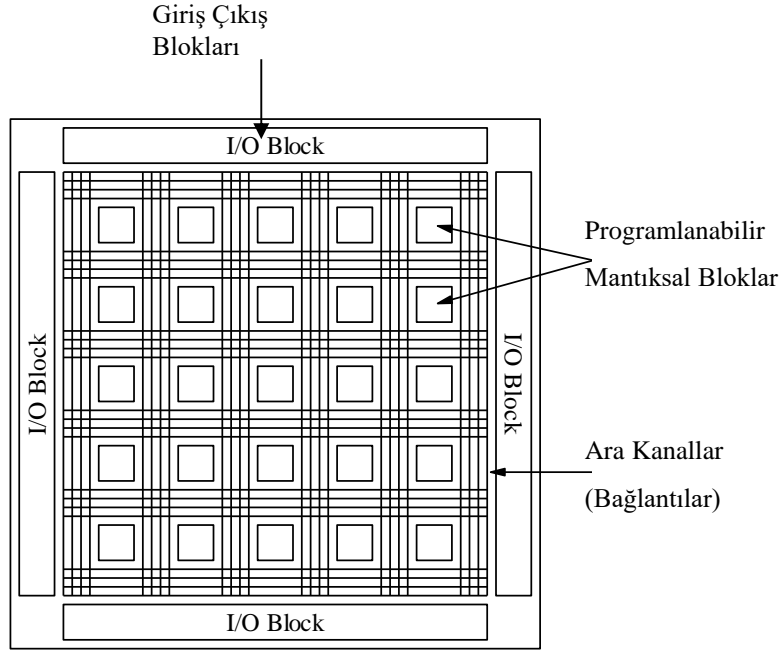
Alan programlanabilir kapı dizileri olarak adlandırılan FPGA çipleri tekrar tekrar programlanabilen sayısal tüm devrelerdir. FPGA çipleri ilk üretim aşamasından sonra tasarımcının ihtiyaç duyduğu fonksiyona göre donanım yapısı değişebilen ve sahada tekrar yapılandırılabilen tümleşik platformlardır. Başka bir ifade ile FPGA çip üretimi tamamlandıktan sonra sahada çipin üzerinde sayısal bir sistemin tasarımı gerçekleştirilebilmektedir (Şeker vd. 2018). FPGA çipleri içindeki transistör yapıları serbest olarak ve birbirinden bağımsız şekilde üretilmiş bir entegre olarak düşünülebilir. Tasarımcının belirlediği fonksiyona göre çip içerisindeki transistör yapıları birbirine bağlanmakta ve istenilen fonksiyon tasarımı gerçekleşmiş olmaktadır (Silahtar 2018). FPGA çipleri gerçek zamanlı, yüksek frekans, paralel işlem yapabilme özelliği ve içerisine işlemci gömülebilmesi gibi özelliklere sahip olmasından dolayı son yıllarda sinyal işleme Diao *et al.* (2018), görüntü işleme Koyuncu vd. (2015), ses tanıma Rodriguez-Orozco *et al.* (2018), yapay sinir ağları Libano *et al.* (2018), uzay araçları Schafer *et al.* (2009), kaotik osilatör tasarımları Tlelo-Cuautle *et al.* (2019), Rajagopal *et al.* (2018), savunma sistemleri Drozd and Kapulin (2018), güvenli haberleşme

Akkaya vd. (2018), Şahin (2012), rasgele sayı üreticileri Bakiri *et al.* (2018), Dong *et al.* (2019) gibi birçok alanda kullanılmaktadır. Xilinx firmasına ait bir Virtex-6 FPGA kartı Resim 3'te verilmiştir.



Resim 3.1 Xilinx Virtex-6 FPGA kartı.

Son zamanlarda gelişen teknoloji ile beraber MHz seviyelerinden başlayarak GHz seviyelerine kadar çalışma frekansına sahip FPGA çipleri üretilmektedir. FPGA çiplerinin başka bir avantajı ise IP core (Intellectual Properties-core) yapıları kullanılarak istenen sayısal-tabanlı tasarımının daha hızlı oluşturulmasına imkan sağlamaktadır. FPGA çipleri giriş-çıkış blokları (I-O blocks), konfigüre edilebilir mantıksal bloklar (Configurable Logic Block (CLB)) ve ara bağlantılar (interconnection network) olmak üzere üç ana bölümden oluşmaktadır. Şekil 3.3'te FPGA çip yapısı görülmektedir (Koyuncu ve Şahin 2011).



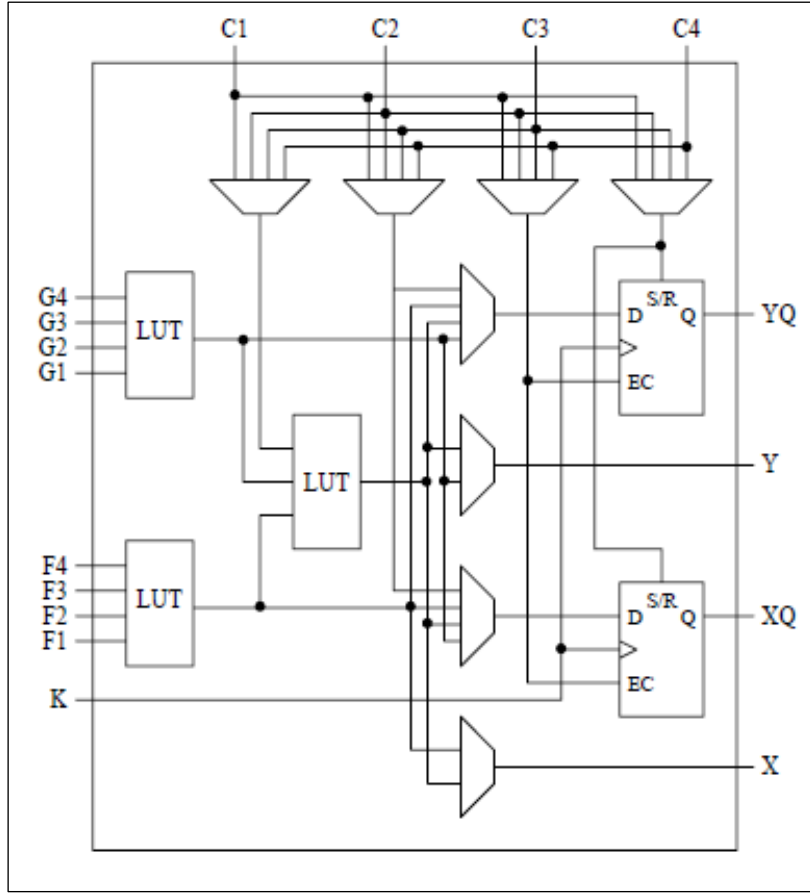
Şekil 3.3 FPGA genel yapısı.

3.2.3.1 Giriş- Çıkış Blokları

Giriş/Çıkış blokları FPGA'in programlanabilir pinleridir. Bu giriş çıkış pinleri FPGA çipi ile dış dünya arasındaki veri iletişimini sağlamaktadır. Ayrıca bu pinler programlanabilmektedir bir başka ifadeyle tasarımcının ihtiyacına göre giriş, çıkış ya da hem giriş hem çıkış pini olarak programlanabilmektedir. Ek olarak güç pinleri, saat (clock) pinleri, konfigürasyon pinleri ve kullanıcı pinleri gibi gerekli olan FPGA çipi üstünde bir çok pin bulunmaktadır (Silahtar 2018).

3.2.3.2 Yapılandırılabilir Mantıksal Bloklar

Şekil 2.4'te görüldüğü gibi CLB genel olarak Look-up table (LUT), Flip-Flop ve multiplexer yapısından oluşmaktadır. Bu yapılardan LUT ünitesi mantıksal fonksiyonların oluşturulduğu ve mantık işlemini gerçekleştirebilen yapıdır. Flip-Flip ünitesi tek bitlik bilgilerin saklandığı ve multiplexer yapısı ise bilgi akışının yönlendirilmesine olanak sağlayan kısımdır. Şekil 3.4'te bir CLB yapısı örneği verilmiştir (Koyuncu 2014).



Şekil 3.4 Örnek bir FPGA çipinin CLB yapısı.

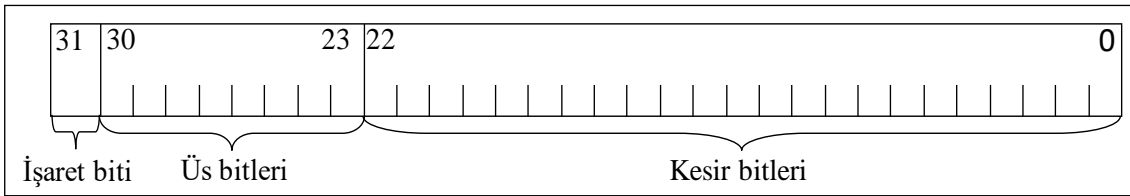
3.2.3.3 Ara Bağlantılar (Interconnections)

Programlanabilir bir yapıya sahip olan ara bağlantıların temel görevi haberleşme görevini yerine getirmektir. Bir başka ifadeyle ara bağlantılar, mantık blokları arasındaki bağlantıları veya mantık blokları ile giriş çıkış blokları arasındaki bağlantı yapılandırmasını sağlamaktadır (Koyuncu 2014, Silahtar 2018).

3.3 Kayan Noktalı Sayı Standardı

Günlük yaşamda kullanılan sayılar eksi sonsuzdan artı sonsuza kadar gitmekte ve tüm bu sayı değerleri ifade edilebilmektedir. Fakat donanımsal uygulamalarda ya da benzetim çalışmalarında tüm bu sayı değerlerinin ifade edilip gösterilmesi mümkün değildir. Bu sebeplerden dolayı benzetim çalışmalarında ve donanımsal uygulamalarda sonsuza giden değerler yerine, donanım platformlarının kapasitesine bağlı olarak

yaklaşık değerleri ifade eden bazı sayı standartları kullanılmaktadır. Bu sayı standartlarına kayan noktalı sayı standardı ve sabit noktalı sayı standardı örnek olarak verilebilir. Kayan noktalı sayı standardı dinamiklik ve hassas çözüm üretmesinden dolayı birçok algoritma uygulamalarında tercih edilmektedir. Bu avantajlarına rağmen kayan noktalı sayı formatı donanım kaynaklarını aşırı derecede tüketmektedir. Bu sayı standardı her platforma uyumlu bir gösterime sahip olsun diye IEEE 754-1985 standardı geliştirilmiştir (Kahan 1996). Yarım hassasiyetli (half-precision 16-bit), tek hassasiyetli (32-bit) ve çift hassasiyetli (64-bit) gibi farklı standartlara sahip kayan noktalı sayı gösterimleri bulunmaktadır. Şekil 3.5'te 32-bit tek duyarlı IEEE 754-1985 kayan noktalı sayı standardı gösterimi verilmiştir (Çavuşlu vd. 2006, Tuna 2017).



Şekil 3.5 32-bit IEEE 754-1985 kayan noktalı sayı standardı gösterimi.

Şekil 3.5'te görüldüğü gibi 31'inci bit işaret (significant) bitini temsil etmektedir. İşaret biti değeri 0 (sıfır) ise sayı pozitif, işaret biti değeri 1 ise sayı negatiftir. 8-bit üs (exponent) bitleri olarak adlandırılmaktadır. Bu kısım sayının üs değerini tutan bit sayısıdır. Kesirli (mantissa) bitler ise sayının kesirli kısmını ifade etmektedir (Az *et al.* 2007, Tuna 2017). Sayının gerçek değerini kayan noktalı sayılara dönüştüren eşitlik denklem (3.1)'de verilmiştir.

$$\text{sayı} = (-1)^{\text{sign}} (1.f) * 2^{(\text{exp}-127)} \quad (3.1)$$

f değeri burada sıfır ile bir (0-1) arasında bir değişkendir. $\text{exp}-127$ değeri ise sayının $\text{exp}-127$ kuvvetine eşit ve $\text{exp}-127+1$ kuvvetinden küçük olduğu değeri göstermektedir. Bu anlatılan kayan noktalı sayı standardını daha iyi anlamak için 13 sayısını 32-bit IEEE 754-1985 kayan noktalı sayı formatına dönüşümü aşağıda anlatılmıştır (Çavuşlu vd. 2012).

$13 > 0$ olduğu için işaret biti $sign=0$ olur.

$2^3 \leq 13 < 2^4$ olduğu için $exp-127=3$ olmaktadır.

$exp=127+3=130 \rightarrow$ üs bitinin (exp) onluk tabandaki karşılığı elde edilmiştir.

$130=(10000010)_2 \rightarrow exp$ değerinin ikilik tabana çevrilmektedir.

$13=(-1)^0.(1.f).2^3 \rightarrow$ kesirli kısmın onluk tabandaki değeri hesaplanmaktadır.

$13/8=1.f=1.625$

$f=0.625$

$0.625*2=1.25 \rightarrow 1$

$0.25*2=0.50 \rightarrow 0$

$0.50*2=1.00 \rightarrow 1$

$0*2=0 \rightarrow 0$

⋮ ⋮

Bu işlemler sonucunda 13 sayısının 32-bit IEEE 754-1985 kayan noktalı değeri Şekil 3.6'da verilmiştir.

İşaret biti 0	Üs bitleri 10000010	Kesir Bitleri 1010000000000000000000
------------------	------------------------	---

Şekil 3.6 13 sayısının 32-bit IEEE 754-1985 kayan noktalı sayı standardı gösterimi.

f değeri 2 ile çarpılır ve sonucun tam sayı değeri kaydedilir. Daha sonra kalan kesirli kısmı tekrar 2 ile çarpılır ve sonucun tam sayı değeri kaydedilir. Bu işlem 23 defa yapılır ya da kesirli kısım sıfır çıkıncaya kadar devam edilir. Çarpma işlemi sonucunda sıfır elde edildiği zaman elde edilen değerler en ağırlıklı bit olandan yazmaya başlanır. Bulunan sonuç yerine yazıldıktan sonra geriye kalan bitlere sıfır yazılır. Şekil 3.7'de 13 onluk sayı değerini 32-bit IEEE 754-1985 kayan noktalı sayı formatına dönüştüren programın ekran görüntüsü verilmiştir (Çavuşlu vd. 2011).

Girilen Ondalık Değer:

Tek hassasiyet (32 bit) :

İkili: Durum:

Bit 31 İşaret biti <input type="text" value="0"/> 0: + 1: -	30 - 23 arası bitler Üs alanı <input type="text" value="10000010"/> Üs alanı ve üs ögesinin ondalık değeri <input type="text" value="130"/> - <input type="text" value="127"/> = <input type="text" value="3"/>	22 - 0 bit significand <input type="text" value="1.101000000000000000000000"/> Anlamın ondalık değeri <input type="text" value="1.6250000"/>
---	---	--

Şekil 3.7 32-bit IEEE 754-1985 kayan noktalı sayı standardı dönüşüm programı ekran görüntüsü.

3.4 Sabit Noktalı Sayı Standardı

IQ-Math sabit noktalı sayı standardı (IQ-Math fixed point number) işaret biti, tamsayı kısmı (integer (I)) ve kesirli (fractional (Q)) kısımdan oluşmaktadır. Bu sabit noktalı sayı formatı kayan noktalı sayı formatından farklı olarak tam sayı kısmı ve kesirli sayı kısmı tasarımcının ihtiyaç duyduğu biçimde uyarlanabilmektedir. Diğer bir deyişle bu kısımlar ihtiyaç duyulan hassasiyet oranına göre ayarlanabilmektedir. Şekil 3.8’de sabit noktalı sayı formatı verilmiştir. Buradaki S biti işaret biti olarak adlandırılmakta ve bu değer 0 olduğunda pozitif, 1 olduğunda ise sayının negatif olduğu göstermektedir. Tam sayı kısmının bitlerinin hesaplama yöntemi sayının tam kısmının ikilik tabana dönüştürülmesiyle elde edilmektedir (Erick 2007, Özkan vd. 2011).

S 0/1	Tam Sayı Kısmı (I) $I_n \dots I_3 I_2 I_1 I_0$	Kesirli Kısmı (Q) $Q_n \dots Q_3 Q_2 Q_1 Q_0$
----------	---	--

Şekil 3.8 Sabit noktalı sayı standardı gösterimi.

Kesirli kısmı için ise iki farklı yöntem ile hesaplanabilmektedir. Birinci yöntem; kesirli kısım istenilen bit uzunluğu kadar iki ile çarpılır ve çarpım sonucu 1 sayısından küçükse bit değeri yerine 0, değilse veya eşitse bit değeri yerine 1 yazılır daha sonra 1’den çıkarılarak işleme devam edilmektedir (Şahin vd. 2013). Bu anlatılan yöntem 99,1545 sayısını kullanarak 8I-8Q sabit noktalı sayı formatına dönüşümü aşağıda anlatılmıştır.

Sayı: 99,1545

sayı > 0 olduğu için S=0

Tam sayı kısmı:99

Kesirli sayı kısmı:0,1545

Tam sayı kısmı ikilik tabana dönüştürüldüğünde;

99= (01100011)₂ elde edilmektedir.

Kesirli kısım olan 0,1545 sayısını ikilik sayı tabanına dönüşümü aşağıdaki gibidir.

1) $0,1545 * 2 = 0,309 \rightarrow 0$

2) $0,309 * 2 = 0,618 \rightarrow 0$

3) $0,618 * 2 = 1,236 \rightarrow 1 \rightarrow 1,236-1=0,236$

4) $0,236 * 2 = 0,472 \rightarrow 0$

5) $0,472 * 2 = 0,944 \rightarrow 0$

6) $0,944 * 2 = 1,888 \rightarrow 1 \rightarrow 1,888-1=0,888$

7) $0,888 * 2 = 1,776 \rightarrow 1 \rightarrow 1,776-1=0,776$

8) $0,776 * 2 = 1,552 \rightarrow 1$

Bu yöntem kullanılarak 99,1545 sayısının 8I-8Q sabit noktalı sayı formatına dönüşümünden elde edilen sonuç Şekil 3.9’da verilmiştir.

S	Tam Sayı Kısmı (I)	Kesirli Kısmı (Q)
0	1100011	00100111

Şekil 3.9 99,1545 sayısının 8I-8Q sabit noktalı sayı formatına dönüşüm sonucu.

İkinci yöntem ise; tam sayı kısmı aynı birinci yöntemdeki gibi ikilik tabanda yazılmaktadır. Kesirli kısım ise $Q*2^n$ ile temsil edilmektedir. Buradaki n sayısı kesirli kısmın bit sayısını, Q ise kesirli kısmın onluk tabandaki değerini ifade etmektedir. $Q*2^n$ işlem sonucu ikilik tabana dönüştürülerek kesirli kısmın sabit noktalı sayı formatı elde edilmiş olur (İnt.Kyn.1). Bu işlem 99,1545 sayısı örneği ile aşağıda açıklanmaktadır.

Sayı: 99,1545

S=0

Tam sayı kısmı=99

Kesirli sayı kısmı=0,1545

Tam sayı kısmı ikilik tabanda yazıldığında;

$99=(01100011)_2$ elde edilmektedir.

Kesirli kısım ise;

$0,1545*2^8 = 39,552$ elde edilmektedir.

Bu elde edilen değerin tam sayı kısmı ikilik tabana dönüşümü yapıldığında kesirli kısmın sabit sayı formatı cinsinden ifadesi $39 = (00100111)_2$ elde edilmektedir. Bu ikinci yöntemle elde edilen 99,1545 sayısının değeri 01100011.00100111 16-bit 8I-8Q sabit noktalı sayı standardında elde edilmektedir.

Negatif sayılarda hesaplama ise 2'ye tümleyen şeklinde yazılmaktadır. 99,1545 sayısını 8I-8Q sabit noktalı sayı formatındaki değerini kullanarak 2'ye tümleyen şeklinde yazıldığında negatif değeri olan -99,1545 sayısı elde edilmektedir.

1) 01100011.00100111 → 99,1545

2) 10011100.11011000 → 1'e tümleyeni

3) 10011100.11011001 → 2'e tümleyeni → -99,1545 sayısına eşittir.

3.5 VHDL

VHDL (Very High Speed Integrated Circuit **H**ardware **D**escription **L**anguage (Çok Yüksek Hızlı Tümleşik Devre Donanım Tanımlama Dili)) İngilizce isminin baş harflerinden oluşmuştur. FPGA üzerinde sayısal devrelerin tasarlanması ve denenmesi amacıyla kullanılan özel bir donanım tanımlama dilidir. Bu donanım tanımlama dili

1981'de amerikan savunma bakanlığı tarafından başlatıldı ve bu zamana kadar devam edegelmiş ve sürekli geliştirilmiştir. Daha sonra 1986'da bütün geliştirme hakları IEEE organizasyonuna bırakılmıştır. VHDL dili IEEE tarafından IEEE-1987 standardı oluşturulmuş ve daha sonra bu standart geliştirilerek IEEE 1076-2008 standardı haline almıştır. Bu versiyonda, std_logic, std_logic_vector, std_logic_arith, numeric_bit gibi veri tipleri ve alt program paketlerini içermektedir. VHDL dili FPGA çipi üzerine yüklenecek kodu sentezlemek ve bu kodun simülasyonunu gerçekleştirmek için kullanılmaktadır. FPGA üzerinde sayısal tasarım gerçekleştirmek amacıyla geliştirilmiş birçok dil vardır. Bunlar Verilog, VHDL ve Handle-C gibi donanım tanımlama dilleri bulunmaktadır. Bu donanım tanımlama dillerinin yanı sıra sayısal tasarımı lojik kapılar seviyesinde yapılmasına olanak sağlayan diğer bir yöntem ise şematik tasarım yöntemidir. Şematik tasarım yöntemi karmaşık tasarımlarda tasarım zorluğu olduğu için genellikle tercih edilmemektedir. VHDL dili bu karmaşıklığı daha kolay bir şekilde üstesinden gelebilmektedir (Çetin 2014, Savran 2017).

3.5.1 VHDL Veri Nesneleri

VHDL dilinde verilerin anlamlı bir şekilde saklanması ve tutulması için veriler nesne adı verilen yapılarda saklanmaktadır. Bu yapılara örnek olarak sinyaller, değişkenler, sabitler ve dosyalar verilebilir.

Sinyal (signal): bu nesne yapısı güncel değerleri tutar ve devrede ara bağlantıları temsil etmektedir. Sinyaller mimari kısmın içindeki tanım bölümünde tanımlanmaktadır. Sinyal genellikle, SIGNAL sinyal_adi: sinyal_türü :=ilk_değer; olarak tanımlanabilir.

Sabit (constant): bu nesne yapısının değeri tasarım boyunca sabittir. Bu nesne yapısına tasarım ilk aşamasında değer atanmaktadır. Sabit nesnesi, CONSTANT nesne_adi : türü := sabit değer; şeklinde tanımlanabilir.

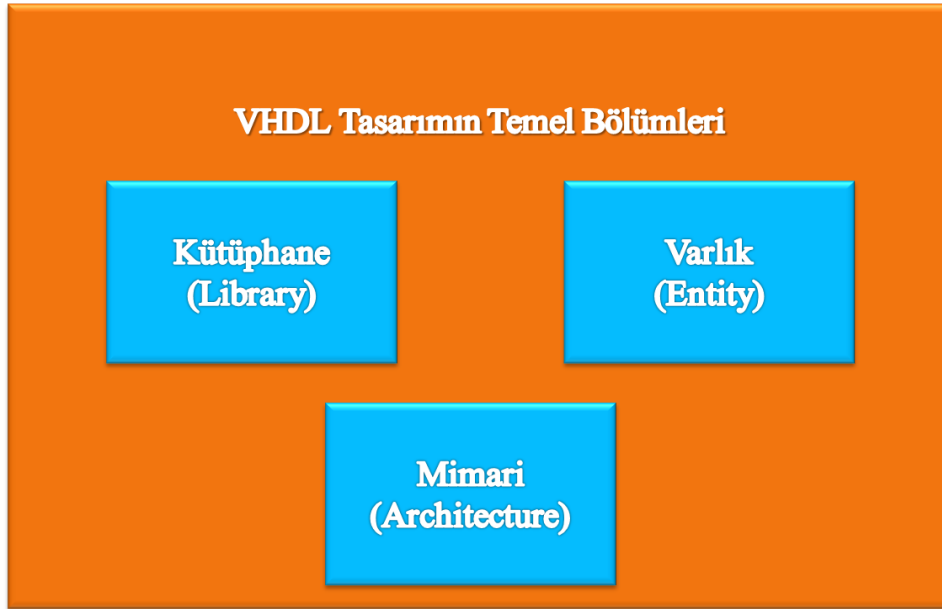
Değişken (variable): geçici değerleri saklayan ve gerektiğinde bu değerleri değiştirilebilme özelliği bulunan nesne türüdür. Değişken nesnesi, VARIABLE değişken_adi : türü := ilk_değer; şeklinde tanımlanabilmektedir (Çetin 2014, Akpolat

2015).

Dosya veri nesnesi ise VHDL dilinde bir veri yazma veya okuma işlemi yapılmasını mümkün kılmaktadır (Kösten ve Çavuşlu 2015).

3.5.2 VHDL Temel Tasarım Yapıları

VHDL dilinde varlık (entity), mimari (architecture) ve kütüphane (library) olmak üzere üç ana yapıdan oluşmaktadır. Şekil 3.10'da VHDL tasarımın temel bölümleri verilmiştir.



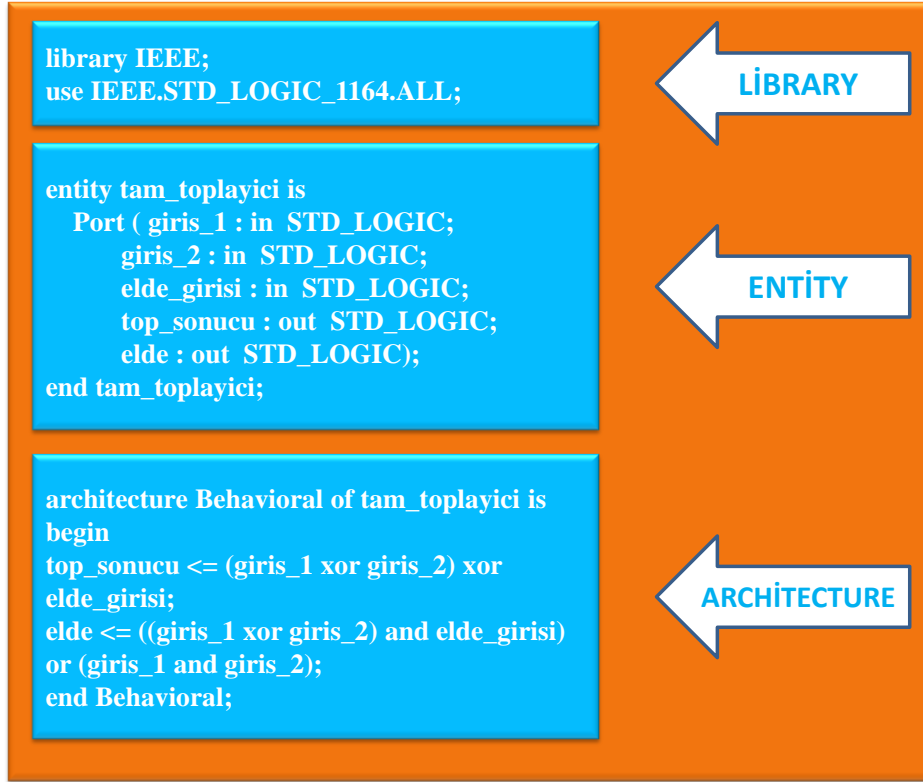
Şekil 3.10 VHDL tasarımın temel bölümleri.

Kütüphane; bu kısımda tasarımda kullanılacak olan hazır standart paket kütüphane tanımlamalarının yapıldığı gibi aynı zamanda tasarımcı tarafından oluşturulan kütüphanelerde kullanılabilir.

Entity; kısmında yapılan tanımlamalar sistemin dış dünya ile iletişimini sağlayan ara yüzüdür. Yani başka bir ifadeyle bu kısımda tasarımcı tarafından sisteme ait tüm giriş çıkış portlarının VHDL diline uygun olarak tanımlandığı yerdir. Her bir tasarım için sadece bir entity tanımlaması yapılabilir ve entity tanımlaması kütüphane tanımlaması

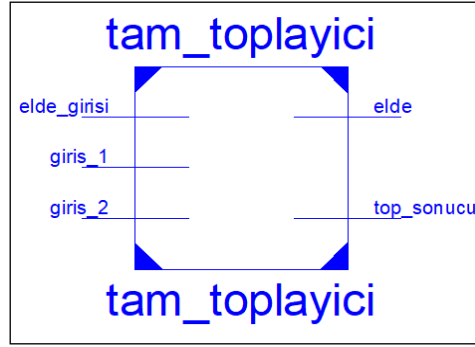
yapıldıktan sonra yapılmaktadır.

Mimari; entity’de tanımlanan giriş çıkış portlarının arasındaki ilişki bu kısımda tanımlanmaktadır. Bir başka ifadeyle tasarımın içyapısının şekillendirildiği ve tasarımın davranışını belirlendiği bölümdür. Mimari kısım iki bölümden oluşmaktadır. Birinci kısmı sinyal, sabit, vb. gibi veri nesnelerinin tanımlandığı yerdir. İkinci kısmı ise sinyal atamaları, mimari bileşen yapılarının atamaları ve *process* gibi yapıların eşzamanlı olarak paralel işlemlerin yapıldığı yerdir (Karataş ve Sarıtaş 2013, Kösten ve Çavuşlu 2015). Tüm bu anlatılan entity, mimari ve kütüphane yapılarını daha iyi anlamak için tam toplayıcı devresi VHDL kodları Şekil 3.11’de verilmiştir.

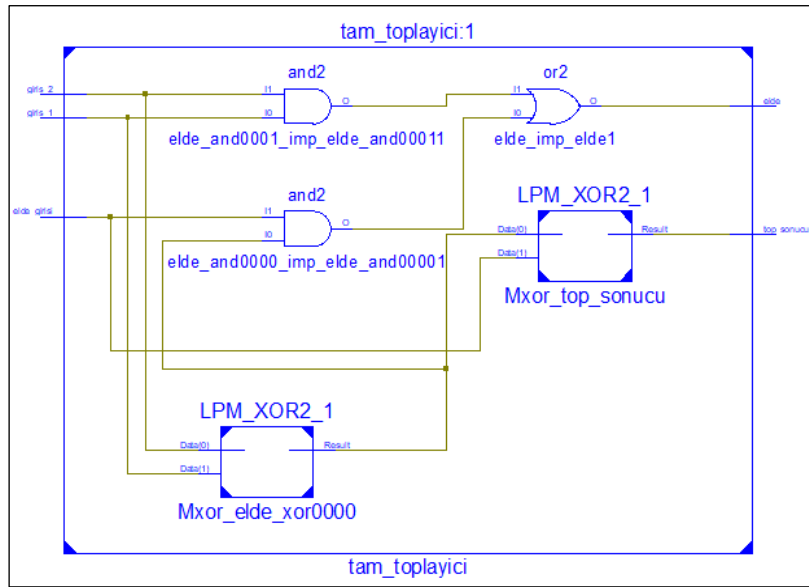


Şekil 3.11 Tam toplayıcı VHDL kodları.

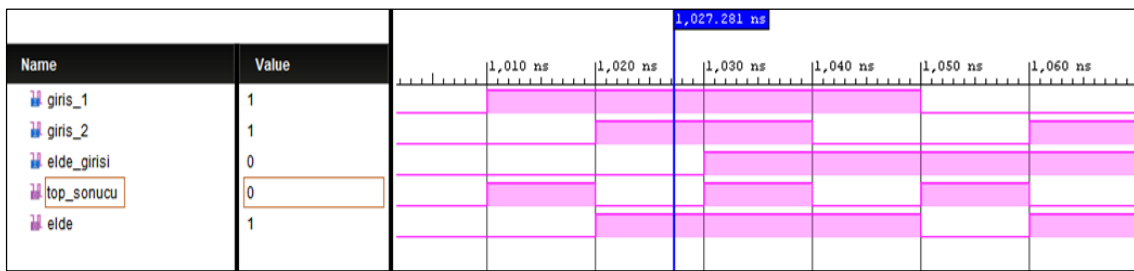
Tam toplayıcı devre tasarımı için Xilinx ISE 14.7 programı kullanılmıştır. Tasarıma ait en üst blok şeması Şekil 3.12’da, ikinci alt seviye blok şeması Şekil 3.13’te ve Xilinx ISE 14.7 simülasyon sonuçları Şekil 3.14’te verilmiştir.



Şekil 3.12 Tam toplayıcı en üst seviye blok şeması.



Şekil 3.13 Tam toplayıcı devresine ait ikinci seviye blok diyagramı.



Şekil 3.14 Tam toplayıcı Xilinx ISE 14.7 simülasyon sonuçları.

3.6 Nümerik Analiz Yöntemleri

3.6.1 Euler Nümerik Analiz Yöntemi

Kaotik sistemler adi diferansiyel denklemler olarak ifade edilmektedir. Bu diferansiyel denklemlerin sayısal çözümü için literatürde birçok nümerik analiz yöntemleri önerilmiştir. Euler algoritması yöntemi, Taylor serisinin sadece birinci terimini kullanarak elde edilen yöntemdir. Bu yöntem diğer yöntemlere göre en basit ve donanımsal uygulanması en kolay olanıdır (Koyuncu vd. 2014). Bu sebepten dolayı Euler algoritması en çok tercih edilen nümerik analiz yöntemlerinden birisi olmaktadır. Bu yöntem çok tercih edilmesine rağmen aslında çok hassas sonuçlar üretememektedir. h burada adım uzunluğunu ifade etmekte ve ayrıca çözüm hassasiyetine de etkisi olmaktadır (Dasilva *et al.* 2011). Denklem (3.2)' de Euler algoritması verilmiştir.

$$y_{\lambda+\Delta h} \approx y_{\lambda+1} = y_{\lambda} + f(y_{\lambda})h \quad (3.2)$$

3.6.2 Heun Nümerik Analiz Yöntemi

Literatürde sunulan Euler algoritması küçük adım uzunluğuna ihtiyaç duyması ve büyük adım uzunluğunda yeterince hassas sonuç vermemesi, hata oranının yüksek olmasından dolayı farklı yöntemler geliştirilmiştir. Bu sebepten dolayı Euler nümerik algoritmasından daha ileri seviyede sayısal çözümler üreten Heun nümerik algoritması yöntemi literatüre sunulmuştur. Heun nümerik analiz yöntemi Euler yöntemindeki λ noktadaki türevi yerine, λ türevi ve $\lambda + 1$ deki türevlerin aritmetik ortalaması alınarak oluşturulan bir yöntemdir (Rüemelin 1982, Tuna vd. 2015). Heun algoritmasının matematiksel eşitliği denklem (3.3)'te verilmiştir.

$$\begin{aligned} y(x_0) &= y_i = y_0 \\ f(y_{\lambda+1}^0) &= y_{\lambda} + f(y_{\lambda}) * h \\ f(y_{\lambda+1}) &= y_{\lambda} + \frac{f(y_{\lambda}) + f(y_{\lambda+1}^0)}{2} * h \end{aligned} \quad (3.3)$$

3.6.3 RK4 Nümerik Analiz Yöntemi

Heun nümerik analiz yöntemi, Euler nümerik analiz yönteminden daha hassas sonuçlar üretebilmektedir. RK4 nümerik analiz yöntemi Taylor serisinden ikiden daha fazla terim kullanılarak elde edilmiştir. Bu sebepten dolayı Euler ve Heun yöntemlerinden daha hassas çözümler elde edilmiştir. RK4 nümerik analiz yöntemi, Euler ve Heun nümerik analiz yöntemlerine göre daha iyi sonuç üretmenin yanı sıra, aynı zamanda hata oranı da oldukça düşüktür. RK4 nümerik algoritmasının matematiksel eşitliği denklem (3.4)'te verilmiştir (De Micco and Larrondo 2011, Karakaya vd. 2017).

$$\begin{aligned}y_{i+1} &= y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\k_1 &= F(y_i) \\k_2 &= F\left(y_i + \frac{h}{2}k_1\right) \\k_3 &= F\left(y_i + \frac{h}{2}k_2\right) \\k_4 &= F(y_i + hk_3)\end{aligned}\tag{3.4}$$

3.6.4 RK5-Butcher Nümerik Analiz Yöntemi

RK5-Butcher algoritması neredeyse benzer bir yapıya sahip olan RK4 algoritmasına k_5 ve k_6 parametre değerleri eklenerek elde edilen bir nümerik analiz yöntemidir. Bu yöntem RK4 ile kıyaslandığında daha hassas çözümler ürettiği görülmektedir. RK5-Butcher algoritmasının matematiksel eşitliği denklem (3.5)'te verilmiştir (Tuna vd. 2018).

$$\begin{aligned}
y_{i+1} &= y_i + \frac{h}{90} (7k_1 + 32k_3 + 32k_5 + 7k_6) \\
k_1 &= F(y_i) \\
k_2 &= F(y_i + \frac{h}{4} k_1) \\
k_3 &= F(y_i + \frac{h}{8} k_1 + \frac{h}{8} k_2) \\
k_4 &= F(y_i - \frac{h}{2} k_2 + hk_3) \\
k_5 &= F(y_i + \frac{3h}{16} k_1 + \frac{9h}{16} k_4) \\
k_6 &= F(y_i - \frac{3h}{7} k_1 + \frac{2h}{7} k_2 + \frac{12h}{7} k_3 - \frac{12h}{7} k_4 + \frac{8h}{7} k_5)
\end{aligned} \tag{3.5}$$

3.6.5 DP Nümerik Analiz Yöntemi

DP nümerik algoritması $k_1, k_2, k_3, k_4, k_5, k_6$ ve k_7 olmak üzere yedi adımdan oluşmaktadır. DP algoritması, RK5-Butcher nümerik algoritmasından yapısal olarak incelendiğinde çok fazla bir fark olmamakla beraber k_7 parametresi eklenmiş ve daha hassas sonuçlar üretmesi sağlanmıştır. Bu parametrelerin eklenmesi algoritmanın daha hassas sonuç sağlamasına rağmen algoritmanın yazılımsal ve donanımsal olarak gerçekleşmesini zorlaştırmakta ayrıca sayısal tabanlı gerçek zamanlı uygulamalarda daha çok çip kaynağı tüketmesine sebep olmaktadır. DP algoritmasının matematiksel eşitliği denklem (3.6)'da verilmiştir. DP nümerik algoritması incelendiğinde y_i algoritmanın başlangıç değerlerini, h ise adım sayısını ifade etmektedir ve adım sayısı $h=0.01$ alınmıştır (Koyuncu vd. 2018, Zhou *et al.* 2018).

$$\begin{aligned}
y_{i+1} &= y_i + h\left(\frac{35}{384}k_1 + \frac{500}{1113}k_3 + \frac{125}{192}k_4 - \frac{2187}{6784}k_5 + \frac{11}{84}k_6\right) \\
k_1 &= F(x_i, y_i) \\
k_2 &= F\left(x_i + \frac{h}{5}, y_i + \frac{h}{5}k_1\right) \\
k_3 &= F\left(x_i + \frac{3}{10}h, \left(y_i + \frac{3}{40}k_1 + \frac{9}{40}k_2\right)*h\right) \\
k_4 &= F\left(x_i + \frac{4}{5}h, \left(y_i + \frac{44}{45}k_1 - \frac{56}{15}k_2 + \frac{32}{9}k_3\right)*h\right) \\
k_5 &= F\left(x_i + \frac{8}{9}h, \left(y_i + \frac{19372}{6561}k_1 - \frac{25360}{2187}k_2 + \frac{64448}{6561}k_3 - \frac{212}{729}k_4\right)*h\right) \\
k_6 &= F\left(x_i + h, \left(y_i + \frac{9017}{3168}k_1 - \frac{355}{33}k_2 + \frac{46732}{5247}k_3 + \frac{49}{176}k_4 - \frac{5103}{18656}k_5\right)*h\right) \\
k_7 &= F\left(x_i + h, \left(y_i + \frac{35}{384}k_1 + 0*k_2 + \frac{500}{1113}k_3 + \frac{125}{192}k_4 - \frac{2187}{6784}k_5 + \frac{11}{84}k_6\right)*h\right)
\end{aligned} \tag{3.6}$$

3.7 Kaos ve Kaos Analiz Yöntemleri

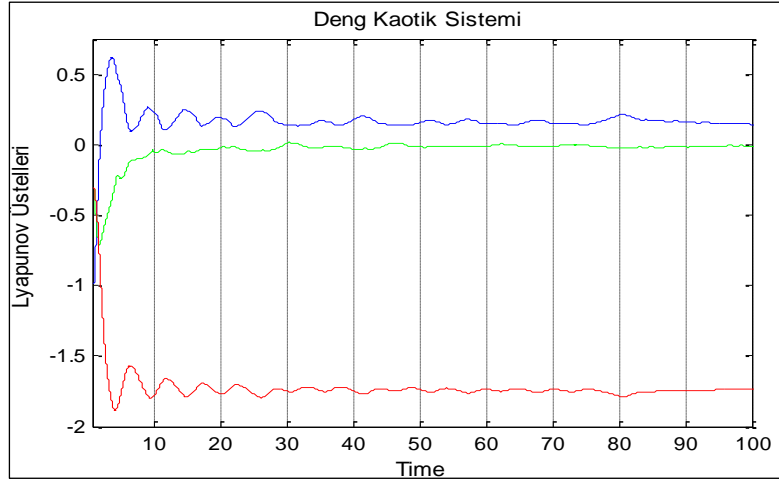
Karmaşık ve doğrusal olmayan bir davranış biçimi gösteren kaotik sistemler gürültüye benzeyen güç spektrumuna sahip olması, genliği ve frekansı tespit edilemeyen çok farklı periyodik salınımlardan oluşabilmektedir (Akgül vd. 2016). Ama bu karmaşık özelliklerinin aksine kaotik sistemler belirli bir alanda dinamik sistemlere sahiptirler. Dinamik bir sistem geçmiş ve şimdiki durumunun yanı sıra olası durumların kümesini kapsamaktadır. Kaotik sistemler, ayrık zamanlı kaotik sistemler ve sürekli zamanlı olarak iki grupta incelenebilir. Kural gereği olarak bir kaotik sistem ayrık işaretler olarak incelenirse ayrık zamanlı kaotik sistem diye adlandırılır, eğer sürekli zaman dinamikleri olarak incelenirse sürekli zamanlı kaotik sistem diye adlandırılır ve bu sistemler diferansiyel denklem kümelerinden oluşmaktadır (Pehlivan 2007). Ayrık zamanlı kaotik sistemler genellikle tek denklemden oluşabildikleri gibi iki ve üç denklemden de oluşabilmektedir. Literatüre sunulan bazı tek boyutlu ayrık kaotik sistemlere Logistic Map (Ismail *et al.* 2017), Cubic Map (Wei *et al.* 2007), Tent Map (Pareek *et al.* 2005), Gauss Map (Gandomi *et al.* 2013), Cusp Map (Akgül vd. 2018) örnek olarak verilebilir. Ayrık zamanlı kaotik sistemler ayrıklaştırma algoritmaları kullanılmadan sayısal uygulamada kullanılabilirken, sürekli zamanlı kaotik sistemler ayrıklaştırma işleminin ardından sayısal uygulamalarda kullanılabilir. Bir

sistemin kaotik bir yapıya sahip olup olmadığını anlamak için literatürde birçok yöntem önerilmiştir. Bunlar;

- Yörünge izlenmesi (zaman serileri – Time series),
- Sistemin faz portrelerinin incelenmesi (Phase portrait),
- Lyapunov üstelleri,
- Lyapunov üstelleri frekans spektrumu,
- Poincare haritalama,
- Fraktal Boyut analizi,
- Güç spektrumu,
- Çatallaşma diyagramı (Bifurcation diagram) örnek olarak verilebilir (Tuna 2017).

3.7.1 Lyapunov Üstelleri

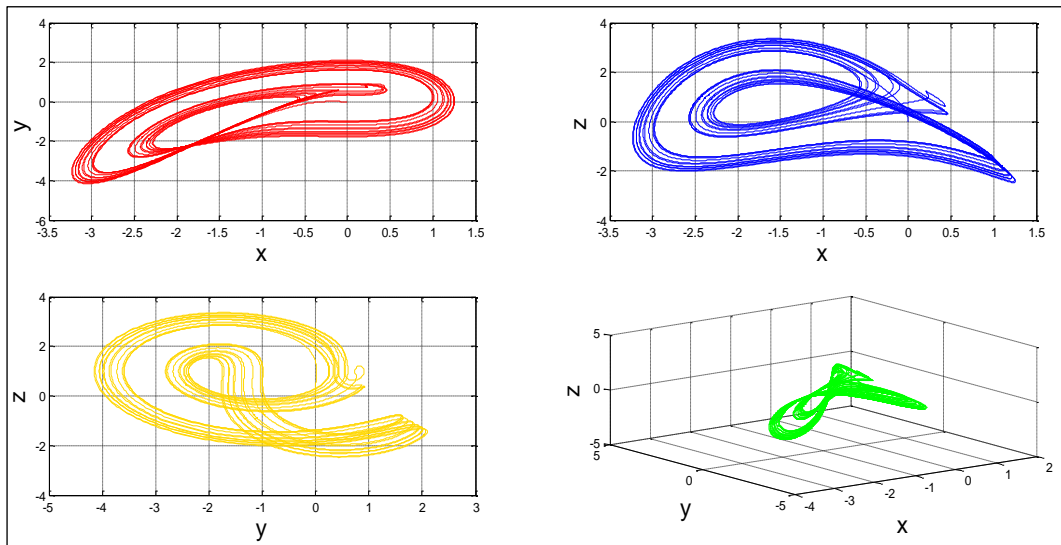
Lyapunov üstelleri yöntemi, sistemin zaman serisinin kaotik bileşenlere sahip olup olmadığını gösteren matematiksel bir analiz yöntemidir. Ayrıca bu yöntem kaotik sistemlerin en önemli özelliklerinden biri olan başlangıç şartlarına hassas bağıllık özelliğinin sayısal ifadesini göstermektedir. Üç boyutlu bir denklem düşünüldüğünde Lyapunov yöntemi ile kaotik analiz yapıldığında sistemin üstelleri $\lambda_1, \lambda_2, \lambda_3$ olarak kabul edilsin. Bu üstellerden birinin pozitif, birinin sıfır, birinin negatif olması sistemin kaotik olması için yeterlidir. Başka bir ifadeyle kaotik bir sistem için Lyapunov üstellerinin alacağı tek durum (+,0,-)'dur. Eğer işaretleri (-, -, -) ise denge durumunda, (0, -, -) ise kendini sürekli tekrarlayan, (0, 0, -) ise iki torus (halka) diye isimlendirilir. Şekil 3.15'te örnek bir kaotik sisteme ait Lyapunov grafiği verilmiştir (Pehlivan 2007, Akgül 2015).



Şekil 3.15 Örnek Lyapunov üstelleri grafiği.

3.7.2 Sistemin Faz Portrelerinin İncelenmesi

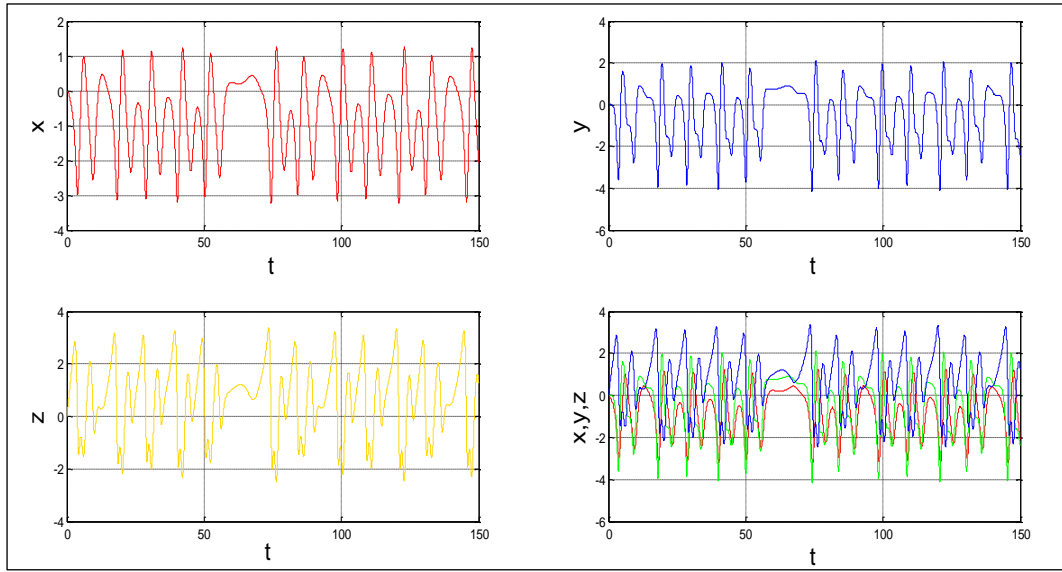
Bir sistemin kaotik davranışını belirleyen diğer bir yöntem ise faz portre analizidir. Kaotik bir sistemin faz portresi, faz uzayı bölgesinde zamanla yörüngesini doldurmaya başlar, asla üzerine kapanmaz ve sürekli tekrar etmektedir. Eğer sistem faz uzayını bu şekilde dolduruyorsa kaotik işaret taşıdığını göstermektedir (Pehlivan vd. 2007). Üç boyutlu kaotik bir sistemin x-y, x-z, y-z, x-y-z olmak üzere kaotik faz portreleri incelenebilmektedir. Bu faz portreleri işlemleri bilgisayar programları kullanılarak gerçekleştirilebilmektedir (Pehlivan 2007). Şekil 3.16’da Matlab programı kullanılarak elde edilen örnek bir kaotik sistemin faz portreleri verilmiştir.



Şekil 3.16 Altın orana sahip kaotik sistemin faz portreleri.

3.7.3 Zaman Serileri Analizi İncelemesi

Kaotik sistemlerin başlangıç şartlarına hassas bağımlı olması, sisteme verilen farklı başlangıç değerlerinin belirli bir zaman içerisinde farklı kaotik işaret üretebilmektedir. Farklı başlangıç değerleri ile üretilen kaotik işaretleri gözlemlemek için sistemin zaman serisi analizi yapılmaktadır (Tuna 2017). Şekil 3.17’de Matlab tabanlı altın orana sahip kaotik sistemin x, y, z ve x-y-z zaman serisi analizi verilmiştir.



Şekil 3.17 Altın orana sahip kaotik sistemin zaman serisi analizi.

3.8 SEA Kaotik Sistemi ve Kaos Analizi

Kaotik sistemlerin diferansiyel denklemleri doğrusal olmayan bir yapıya sahiptir. Kaotik sistem literatürde sürekli zaman diferansiyel denklem ve ayrık zamanlı diferansiyel denklem olarak belirtilmiştir. Sürekli zamanlı kaotik sistem FPGA çipleri üzerinde gerçekleştirilecek ise ayrık zamanlı matematiksel modeli kullanılmaktadır. Bu tez çalışmasında literatürde üzerinde FPGA tabanlı çalışma yapılmamış olan sürekli zamanlı SEA kaotik sistemi seçilmiştir. SEA kaotik sistemi diğer analiz yöntemlerine göre daha hassas sonuçlar üreten ayrıklaştırılmış DP nümerik algoritması kullanılarak modellenmiştir. Ayrıca SEA kaotik sistemin kaotik değer sonuçları kullanılarak faz portreleri, zaman serileri ve Lyapunov üstelleri sunulmuştur (Koyuncu 2014, Tuna 2017). Diferansiyel denklemler şeklinde ifade edilen SEA kaotik sisteminin

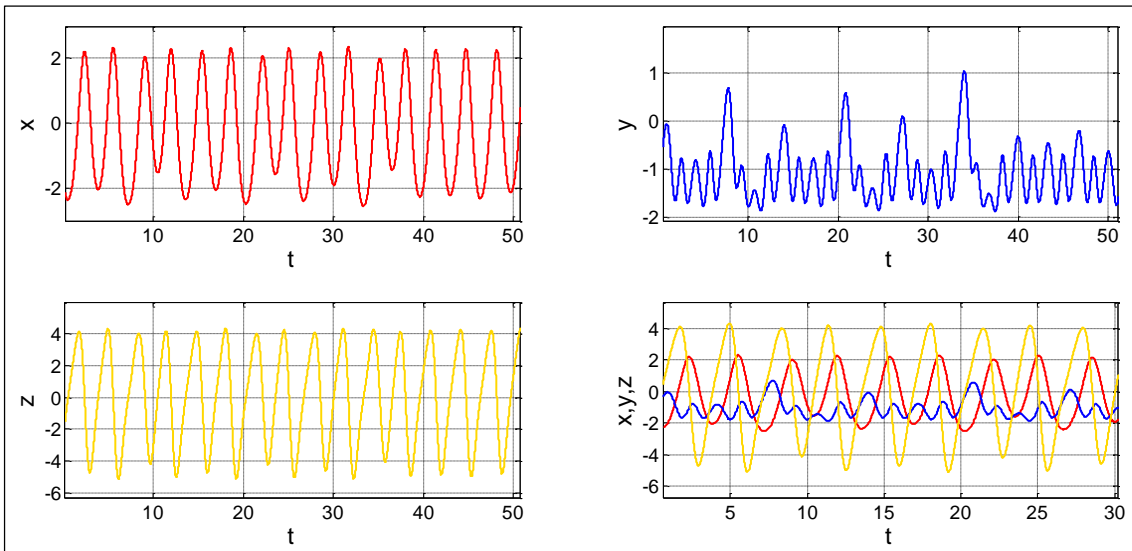
matematiksel ifadesi denklem (3.7)'de, sistem parametreleri ve başlangıç şartları ise denklem (3.8)'de verilmiştir. Bu denklemdeki x , y ve z kaotik durum değişkenini, a , b , c , d , e , f ve g ise sistem parametrelerini, x_0 , y_0 ve z_0 başlangıç şartlarını ifade etmektedir (Xu *et al.* 2018).

$$\begin{aligned} dx / dt &= gz \\ dy / dt &= dx^2 + ey^2 - f \\ dz / dt &= -ax - bx^2 + cy^2 \end{aligned} \quad (3.7)$$

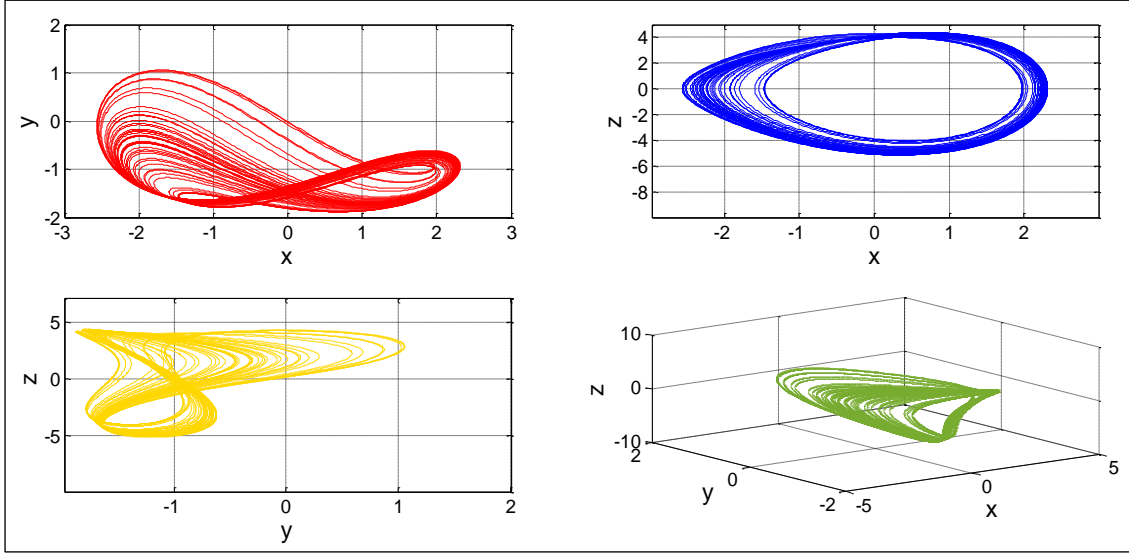
$$\begin{aligned} a = f = 4, b = c = d = e = g = 1 \\ x_0 = -1.8, y_0 = -1.5, z_0 = -2.5 \end{aligned} \quad (3.8)$$

Başlangıç değerleri olarak $x_0=-1,8$, $y_0=-1,5$, $z_0=-2,5$ olarak alınmıştır. Başlangıç şartları değerleri ile kaotik sistemin h adım sayısı kadar sonraki değerleri hesaplanmaktadır.

DP nümerik yöntemiyle Matlab-tabanlı modellenen SEA kaotik sistemin x , y ve z kaotik durum değişkenlerine ait zaman serisi Şekil 3.18'de ve faz portreleri ise Şekil 3.19'da verilmiştir.

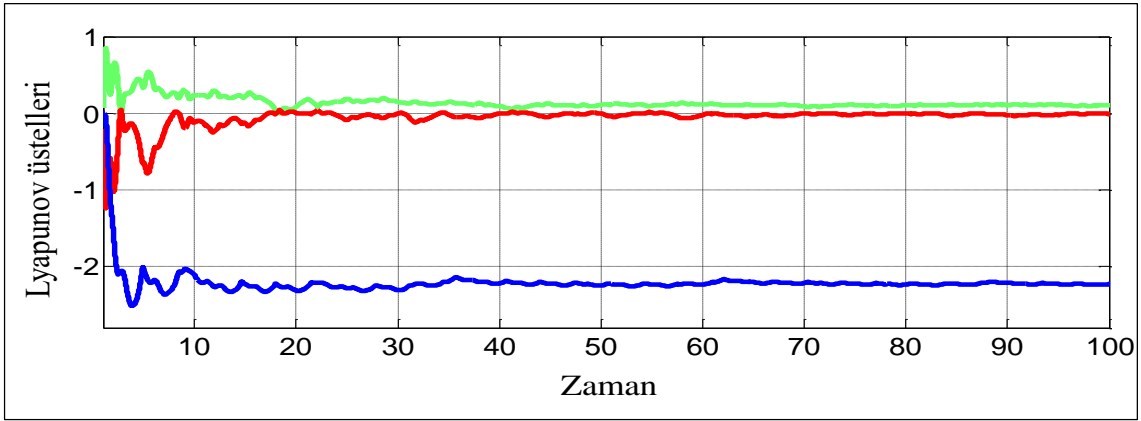


Şekil 3.18 DP algoritma tabanlı SEA kaotik sistemine ait zaman serileri.



Şekil 3.19 DP algoritma tabanlı SEA kaotik sistemine ait faz portreleri.

Kaotik sistemin kaos davranışını incelemek için gerekli olan Lyapunov üstelleri Lyapunov Exponent Toolbox (LET) aracılığıyla Matlab programını kullanılarak elde edilmiştir. Elde edilen Lyapunov üstelleri Şekil 3.20’de verilmiştir.



Şekil 3.20 SEA kaotik sistemin Lyapunov üstelleri.

Sonuçlar analiz edildiğinde SEA kaotik sisteminin Lyapunov üstellerinin işaretleri (λ_1 , λ_2 , λ_3) sırasıyla (+, 0, -) olduğu için kaotik davranış sergilediği görülmektedir.

3.9 SEA Kaotik Sisteminin Ayrıklaştırılmış Modelleri

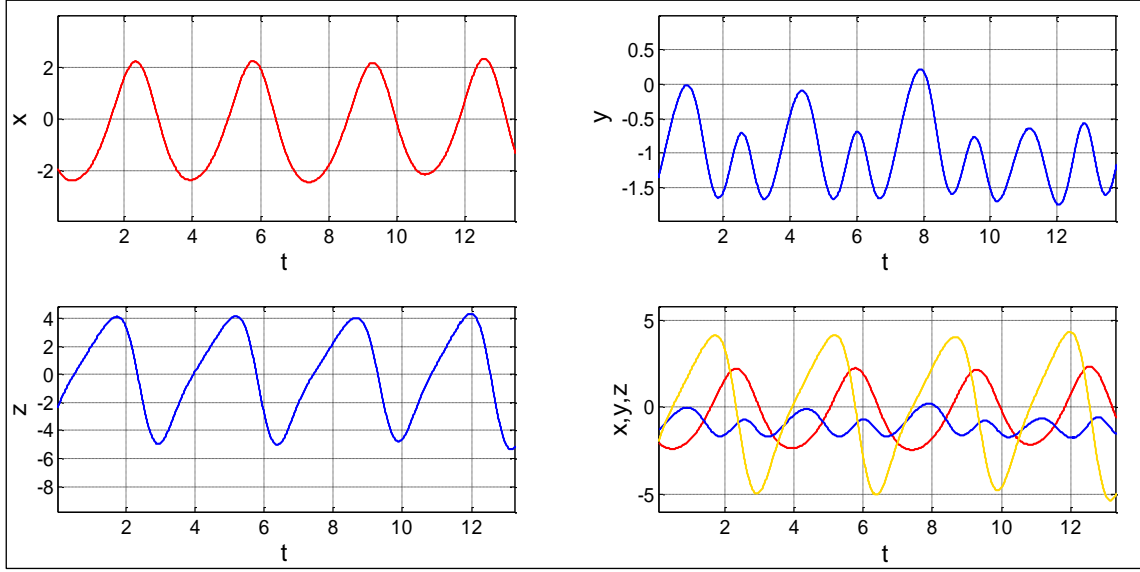
SEA kaotik sistemine ait diferansiyel denklemlerin Euler, Heun, RK4, RK5-Butcher ve DP gibi nümerik analiz algoritmaları yardımıyla sayısal uygulamalarda kullanılabilmesi için ayrık matematiksel modelleri verilmiştir.

3.9.1 Euler Tabanlı Ayrıklaştırılmış Modeli

Sürekli zamanlı kaotik sistem için kullanılacak nümerik algoritmaların ayrıklaştırılmış modelleri çıkartılarak sayısal bir donanım üzerinde modelleme yapılmaktadır. SEA kaotik sistemi Euler algoritması kullanılarak ayrıklaştırılmış matematiksel eşitliği denklem (3.9)'da verilmiştir. $x(k)$, $y(k)$ ve $z(k)$ 'nin başlangıç değerleri SEA kaotik sistem için $x(k)=-1.5$, $y(k)=-1.8$ ve $z(k)=-2.5$ alınmıştır. Δh ise adım sayısını ifade etmektedir.

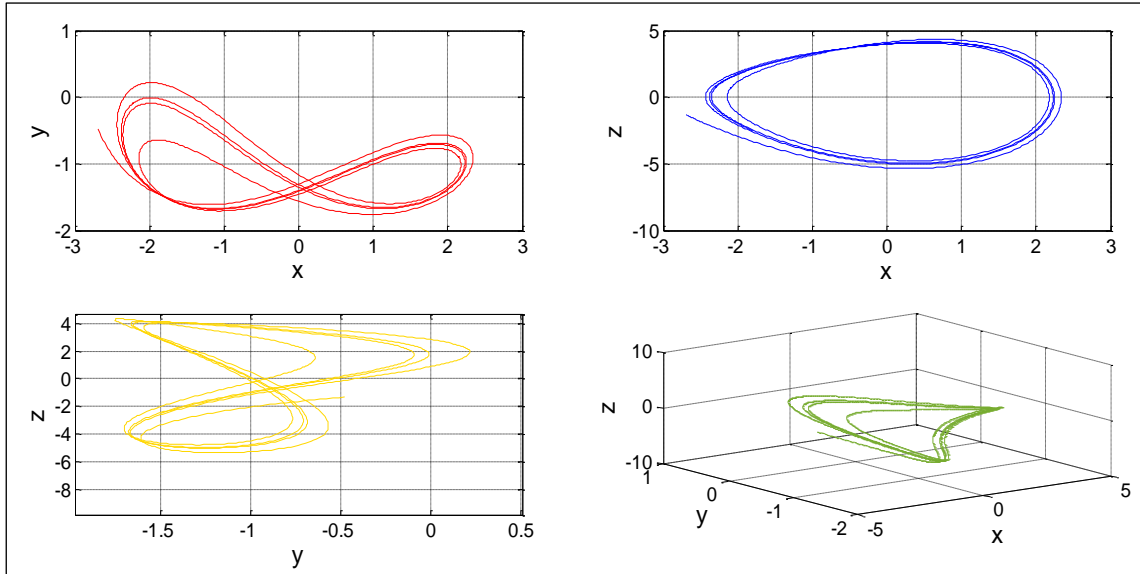
$$\begin{aligned}x(k+1) &= x(k) + \Delta h(g.z(k)) \\y(k+1) &= y(k) + \Delta h((d.x(k)^2) + (e.y(k)^2) - f) \\z(k+1) &= z(k) + \Delta h((-a.x(k)) - (b.x(k).x(k)) + (c.y(k)^2))\end{aligned}\tag{3.9}$$

SEA kaotik sistemi Matlab üzerinde Euler nümerik algoritması kullanılarak modellenmiştir. Euler algoritması kullanılarak elde edilen x, y, z kaotik durum değişkenlerine ait zaman serisi analizi Şekil 3.21'de verilmiştir.



Şekil 3.21 Euler algoritması tabanlı SEA kaotik sistemin zaman serisi.

Aynı başlangıç ve sistem parametrelerine sahip kaotik sisteminin x-y, x-z, y-z faz portreleri ve x-y-z üç boyutlu halı Şekil 3.22’de görülmektedir.



Şekil 3.22 Euler algoritması tabanlı SEA kaotik sistemin faz portreleri.

3.9.2 Heun Tabanlı Ayırıklaştırılmış Modeli

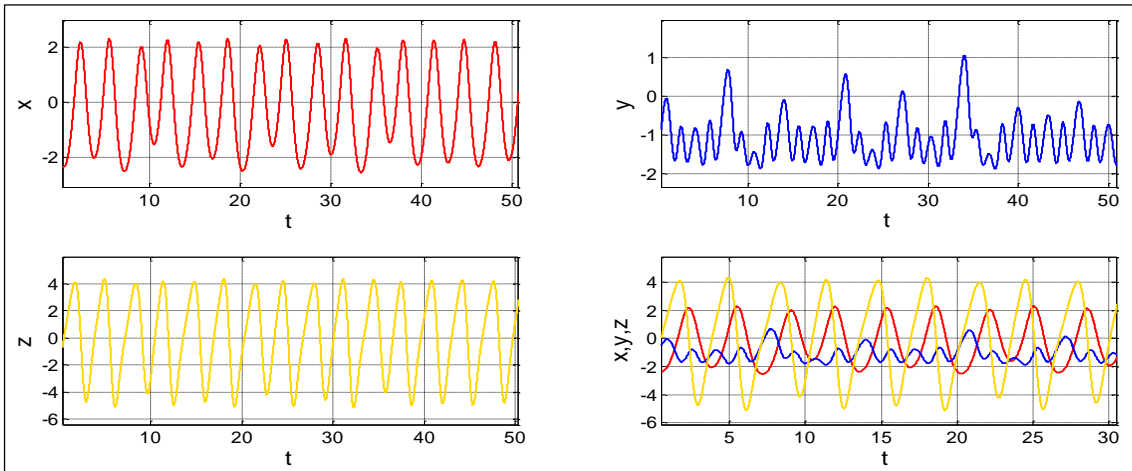
Denklem (3.10)’da Heun nümerik algoritması verilmiştir.

$$\begin{aligned}
y(x_0) &= y_i = y_0 \\
f(y_{\lambda+1}^0) &= y_{\lambda} + f(y_{\lambda}) * h \\
f(y_{\lambda+1}) &= y_{\lambda} + \frac{f(y_{\lambda}) + f(y_{\lambda+1}^0)}{2} * h
\end{aligned}
\tag{3.10}$$

Matematiksel ifadesine bakıldığı zaman birinci kısımda $f(y_{\lambda})$ fonksiyonun değeri h adım sayısı kadar sonundaki ifadesi ve y_{λ} ifadesi toplanarak, $f(y_{\lambda+1}^0)$ değeri hesaplanmaktadır (Koyuncu vd. 2013). Heun algoritmasının ikinci kısmında ise birinci kısımda elde edilen $f(y_{\lambda+1}^0)$ değeri ve $f(y_{\lambda})$ değeri toplamının ikiye bölümden elde edilen ifadenin, h adım sayısı kadar sonraki değeri Heun algoritmasının sonucudur (Tuna vd. 2015). SEA kaotik sistemi Heun algoritması yöntemi kullanılarak ayrıklaştırılmış matematiksel eşitliği denklem (3.11)'te verilmiştir.

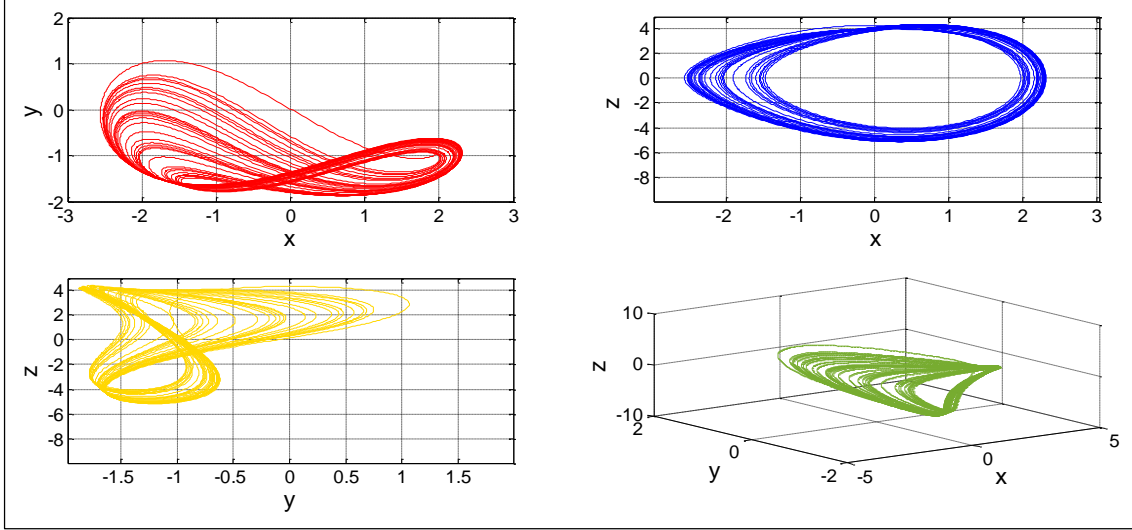
$$\begin{aligned}
x(k+1) &= x(k) + \Delta h(g.z(k)) \\
x(k+1) &= x(k) + \Delta h((g.z(k)) + x(k+1)) / 2 \\
y(k+1) &= y(k) + \Delta h((d.x(k)^2) + (e.y(k)^2) - f) \\
y(k+1) &= y(k) + \Delta h(((d.x(k)^2) + (e.y(k)^2) - f) + y(k+1)) / 2 \\
z(k+1) &= z(k) + \Delta h((-a.x(k)) - (b.x(k).x(k)) + (c.y(k)^2)) \\
z(k+1) &= z(k) + \Delta h((-a.x(k)) - (b.x(k).x(k)) + (c.y(k)^2) + z(k+1)) / 2
\end{aligned}
\tag{3.11}$$

SEA kaotik sistemine ait x , y , z kaotik durum değişkenlerinin zaman serileri Şekil 3.23'te verilmiştir.



Şekil 3.23 Heun tabanlı SEA kaotik sistemin zaman serisi analizi.

SEA kaotik sisteminin Matlab üzerinde Heun yöntemi kullanılarak elde edilen x-y, x-z, y-z faz portreleri ve x-y-z üç boyutlu hali Şekil 3.24'te görülmektedir.



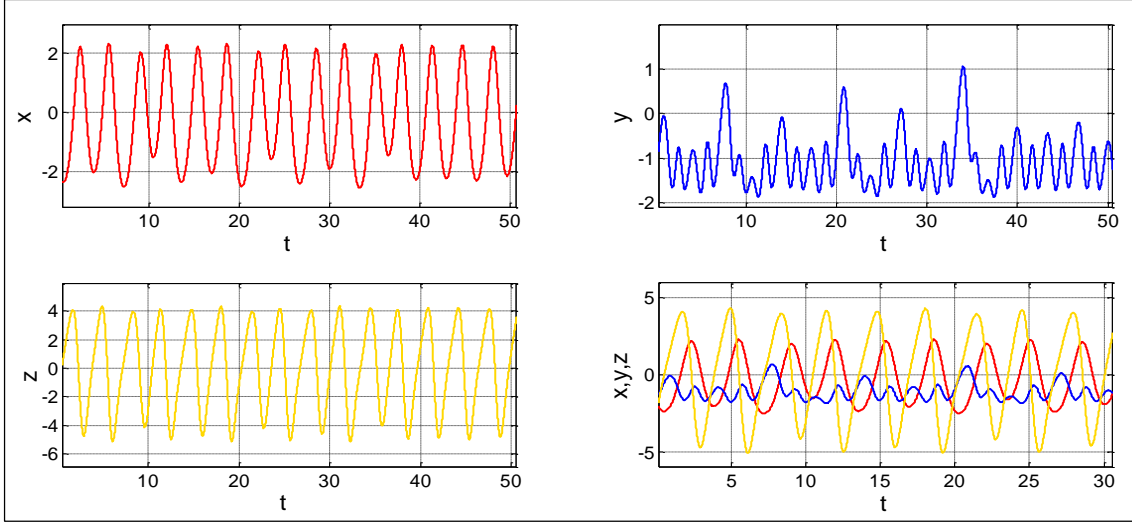
Şekil 3.24 Heun tabanlı SEA kaotik sisteminin faz portreleri.

3.9.3 RK4 Tabanlı Ayrıklaştırılmış Modeli

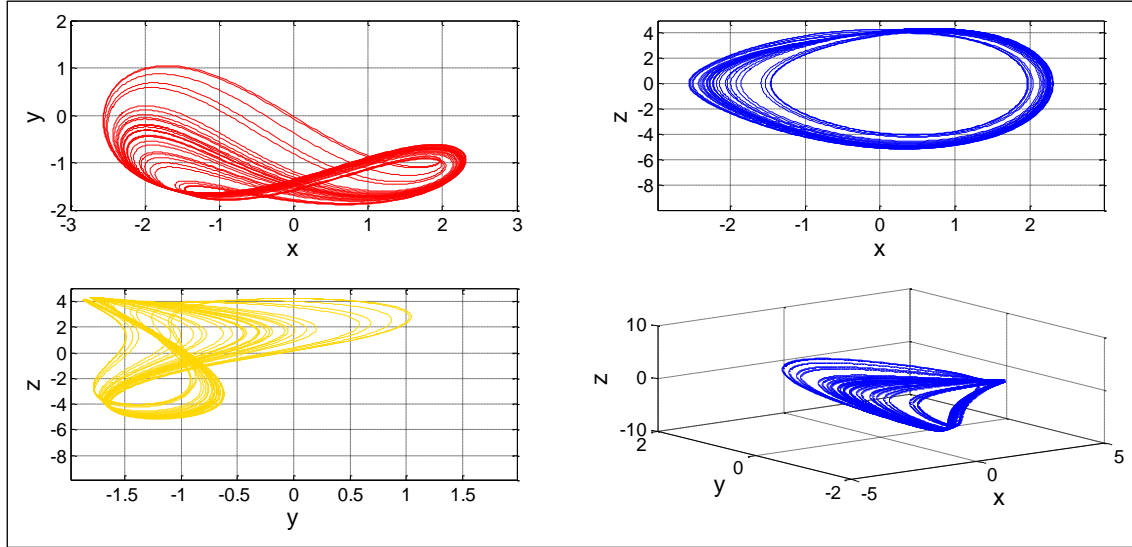
RK4 nümerik analiz yöntemi kullanılarak SEA kaotik sistemin ayrıklaştırılmış matematiksel eşitliği denklem (3.12)'de sunulmuştur. Bu denklemde SEA kaotik sistemin birinci denkleme ait değerleri κ_1 , κ_2 , κ_3 ve κ_4 değişkenleri, ikinci denkleme ait değerleri λ_1 , λ_2 , λ_3 ve λ_4 değişkenleri ve üçüncü denkleme ait değerleri ξ_1 , ξ_2 , ξ_3 ve ξ_4 değişkenleri temsil etmektedir (Alçın vd. 2018). Bu katsayılar RK4 nümerik analiz yönteminde yerine konulduğunda ve Δh adım sayısı kadar sonraki değeri olan $x(k+1)$, $y(k+1)$ ve $z(k+1)$ yaklaşık değerleri bulunmaktadır. κ_1 , λ_1 , ξ_1 parametre değerleri bir sonraki adımda κ_2 , λ_2 , ξ_2 , sisteminde giriş değerleri olarak kullanılmakta ve hesaplamalar sonucunda κ_3 , λ_3 , ξ_3 değerleri bulunmaktadır. Bulunan bu değerler bir sonraki adımın κ_4 , λ_4 , ξ_4 sistemine giriş değerleri olarak verilmektedir (Koyuncu vd. 2014).

$$\begin{aligned}
x(k+1) &= x(k) + \frac{1}{6} \Delta h [\kappa_1(k) + 2\kappa_2(k) + 2\kappa_3(k) + \kappa_4(k)] \\
y(k+1) &= \rho(k) + \frac{1}{6} \Delta h [\lambda_1(k) + 2\lambda_2(k) + 2\lambda_3(k) + \lambda_4(k)] \\
z(k+1) &= \delta(k) + \frac{1}{6} \Delta h [\xi_1(k) + 2\xi_2(k) + 2\xi_3(k) + \xi_4(k)] \\
\kappa_1 &= f(x(k), y(k), z(k)) \\
\lambda_1 &= \rho(x(k), y(k), z(k)) \\
\xi_1 &= \delta(x(k), y(k), z(k)) \\
\kappa_2 &= f(x(k) + \frac{1}{2} \Delta h \kappa_1, y(k) + \frac{1}{2} \Delta h \lambda_1, z(k) + \frac{1}{2} \Delta h \xi_1) \\
\lambda_2 &= \rho(x(k) + \frac{1}{2} \Delta h \kappa_1, y(k) + \frac{1}{2} \Delta h \lambda_1, z(k) + \frac{1}{2} \Delta h \xi_1) \\
\xi_2 &= \delta(x(k) + \frac{1}{2} \Delta h \kappa_1, y(k) + \frac{1}{2} \Delta h \lambda_1, z(k) + \frac{1}{2} \Delta h \xi_1) \\
\kappa_3 &= f(x(k) + \frac{1}{2} \Delta h \kappa_2, y(k) + \frac{1}{2} \Delta h \lambda_2, z(k) + \frac{1}{2} \Delta h \xi_2) \\
\lambda_3 &= \rho(x(k) + \frac{1}{2} \Delta h \kappa_2, y(k) + \frac{1}{2} \Delta h \lambda_2, z(k) + \frac{1}{2} \Delta h \xi_2) \\
\xi_3 &= \delta(x(k) + \frac{1}{2} \Delta h \kappa_2, y(k) + \frac{1}{2} \Delta h \lambda_2, z(k) + \frac{1}{2} \Delta h \xi_2) \\
\kappa_4 &= f(x(k) + \Delta h \kappa_3, y(k) + \Delta h \lambda_3, z(k) + \Delta h \xi_3) \\
\lambda_4 &= \rho(x(k) + \Delta h \kappa_3, y(k) + \Delta h \lambda_3, z(k) + \Delta h \xi_3) \\
\xi_4 &= \delta(x(k) + \Delta h \kappa_3, y(k) + \Delta h \lambda_3, z(k) + \Delta h \xi_3)
\end{aligned} \tag{3.12}$$

RK4 nümerik yöntemiyle Matlab-tabanlı modellenen SEA kaotik sistemin x, y ve z kaotik durum değişkenlerine ait zaman serisi Şekil 3.25'te ve faz portreleri ise Şekil 3.26'da verilmiştir.



Şekil 3.25 RK4 tabanlı SEA kaotik sistemin zaman serisi analizi.



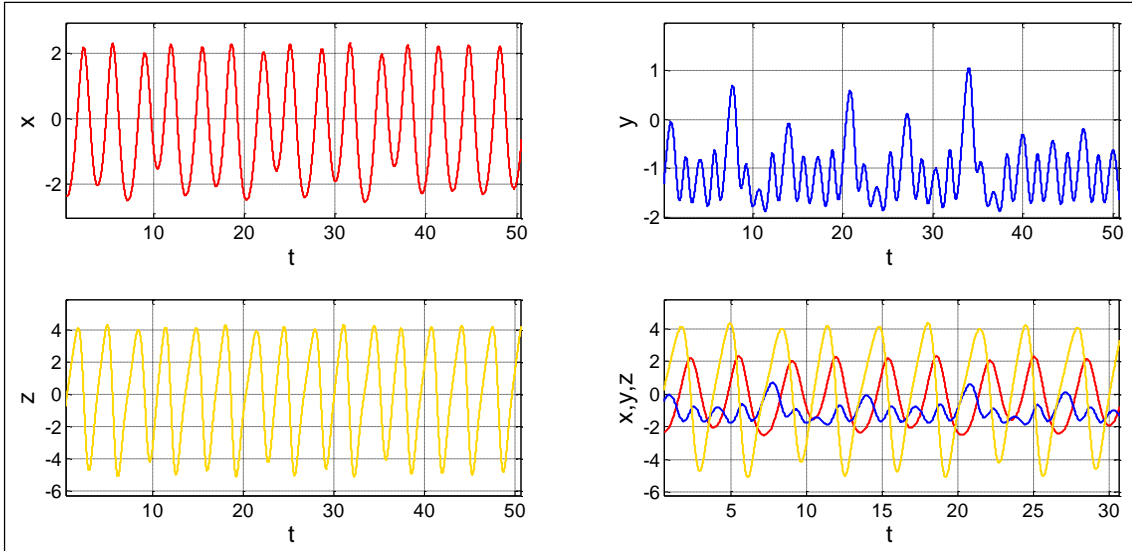
Şekil 3.26 RK4 tabanlı SEA kaotik sistemin faz portreleri.

3.9.4 RK5-Butcher Tabanlı Ayırıklaştırılmış Modeli

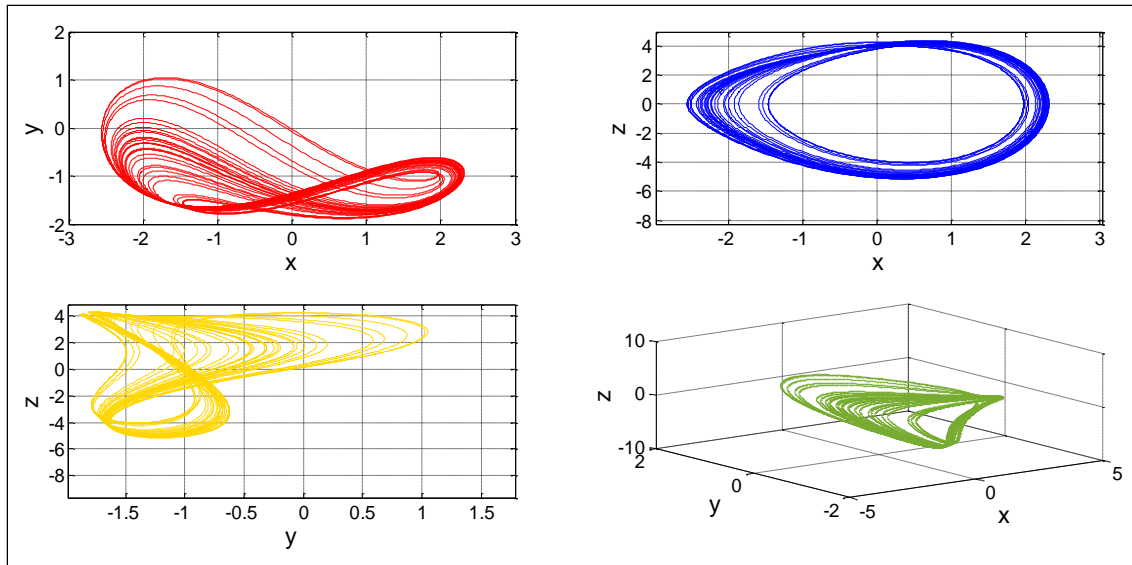
RK5-Butcher nümerik analiz yöntemi kullanılarak SEA kaotik sistemin ayırıklaştırılmış matematiksel eşitliği denklem (3.13)'te sunulmuştur.

$$\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)] \\
\kappa_1 &= f(x(k), y(k), z(k)) \\
\lambda_1 &= \rho(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 &= \rho(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 &= \rho(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 &= \rho(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 &= \rho(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 &= \rho(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{3.13}$$

RK5-Butcher nümerik analiz yöntemiyle Matlab-tabanlı modellenen SEA kaotik sistemin x, y, ve z kaotik durum değişkenlerine ait zaman serisi Şekil 3.27’de ve faz portreleri ise Şekil 3.28’de verilmiştir.



Şekil 3.27 RK5-Butcher tabanlı SEA kaotik sistemin x, y ve z zaman serisi analizi.



Şekil 3.28 RK5-Butcher tabanlı SEA kaotik sistemine ait faz portreleri.

3.9.5 DP Tabanlı Ayrıklaştırılmış Modeli

DP algoritması $k_1, k_2, k_3, k_4, k_5, k_6$ ve k_7 olmak üzere yedi parametre içerir ve y_{i+1} değerinin hesaplanabilmesi için ilk önce bu yedi parametrenin değerinin hesaplanması

gerekmektedir. Başlangıç şartları ve Δh adım sayısı kullanılarak elde edilen sonuç, k_1 parametresini, k_1 ve Δh adım sayısı kullanılarak elde edilen sonuç, k_2 parametresini, k_1 , k_2 ve Δh adım sayısı kullanılarak elde edilen sonuç, k_3 parametresini, k_1 , k_2 , k_3 ve Δh adım sayısı kullanılarak elde edilen sonuç, k_4 parametresini, k_1 , k_2 , k_3 , k_4 ve Δh adım sayısı kullanılarak elde edilen sonuç, k_5 parametresini, k_1 , k_2 , k_3 , k_4 , k_5 ve Δh adım sayısı kullanılarak elde edilen sonuç, k_6 parametresini, k_1 , k_2 , k_3 , k_4 , k_5 , k_6 ve Δh adım sayısı kullanılarak elde edilen sonuç, k_7 parametresinin değerini vermektedir (Koyuncu 2014). DP nümerik analiz yöntemi kullanılarak SEA kaotik sistemin ayrıklaştırılmış matematiksel eşitliği denklem (3.14)'de sunulmuştur. Bu denklemde SEA kaotik sistemin birinci denkleme ait değerleri κ_1 , κ_2 , κ_3 , κ_4 , κ_5 , κ_6 ve κ_7 değişkenleri, ikinci denkleme ait değerleri λ_1 , λ_2 , λ_3 , λ_4 , λ_5 , λ_6 ve λ_7 değişkenleri ve üçüncü denkleme ait değerleri ξ_1 , ξ_2 , ξ_3 , ξ_4 , ξ_5 , ξ_6 ve ξ_7 değişkenleri temsil etmektedir.

$$\begin{aligned}
x(k+1) &= x(k) + \Delta h \left[\left(\frac{35}{384} \kappa_1(k) + \frac{500}{1113} \kappa_3(k) + \frac{125}{192} \kappa_4(k) - \frac{2187}{6784} \kappa_5(k) + \frac{11}{84} \kappa_6(k) \right) \right] \\
y(k+1) &= y(k) + \Delta h \left[\left(\frac{35}{384} \lambda_1(k) + \frac{500}{1113} \lambda_3(k) + \frac{125}{192} \lambda_4(k) - \frac{2187}{6784} \lambda_5(k) + \frac{11}{84} \lambda_6(k) \right) \right] \\
z(k+1) &= z(k) + \Delta h \left[\left(\frac{35}{384} \zeta_1(k) + \frac{500}{1113} \zeta_3(k) + \frac{125}{192} \zeta_4(k) - \frac{2187}{6784} \zeta_5(k) + \frac{11}{84} \zeta_6(k) \right) \right] \\
\kappa_1 &= f(x(k), y(k), z(k)) \\
\lambda_1 &= \rho(x(k), y(k), z(k)) \\
\xi_1 &= \delta(x(k), y(k), z(k)) \\
\kappa_2 &= f\left(x(k) + \frac{1}{5} \Delta h \kappa_1, y(k) + \frac{1}{5} \Delta h \lambda_1, z(k) + \frac{1}{5} \Delta h \xi_1\right) \\
\lambda_2 &= \rho\left(x(k) + \frac{1}{5} \Delta h \kappa_1, y(k) + \frac{1}{5} \Delta h \lambda_1, z(k) + \frac{1}{5} \Delta h \xi_1\right) \\
\xi_2 &= \delta\left(x(k) + \frac{1}{5} \Delta h \kappa_1, y(k) + \frac{1}{5} \Delta h \lambda_1, z(k) + \frac{1}{5} \Delta h \xi_1\right) \\
\kappa_3 &= f\left(x(k) + \frac{3}{40} (\Delta h (\kappa_1 + 3\kappa_2)), y(k) + \frac{3}{40} (\Delta h (\lambda_1 + 3\lambda_2)), z(k) + \frac{3}{40} (\Delta h (\xi_1 + 3\xi_2))\right) \\
\lambda_3 &= \rho\left(x(k) + \frac{3}{40} (\Delta h (\kappa_1 + 3\kappa_2)), y(k) + \frac{3}{40} (\Delta h (\lambda_1 + 3\lambda_2)), z(k) + \frac{3}{40} (\Delta h (\xi_1 + 3\xi_2))\right) \\
\xi_3 &= \delta\left(x(k) + \frac{3}{40} (\Delta h (\kappa_1 + 3\kappa_2)), y(k) + \frac{3}{40} (\Delta h (\lambda_1 + 3\lambda_2)), z(k) + \frac{3}{40} (\Delta h (\xi_1 + 3\xi_2))\right)
\end{aligned} \tag{3.14}$$

(Devam) (3.14)

$$\begin{aligned}\kappa_4 &= f(x(k) + (\frac{44}{45}\Delta h\kappa_1 - \frac{56}{15}\Delta h\kappa_2 + \frac{32}{9}\kappa_3, y(k) + \frac{44}{45}\Delta h\lambda_1 - \frac{56}{15}\Delta h\lambda_2 + \frac{32}{9}\lambda_3, \\ & z(k) + \frac{44}{45}\Delta h\xi_1 - \frac{56}{15}\Delta h\xi_2 + \frac{32}{9}\xi_3)) \\ \lambda_4 &= \rho(x(k) + (\frac{44}{45}\Delta h\kappa_1 - \frac{56}{15}\Delta h\kappa_2 + \frac{32}{9}\kappa_3, y(k) + \frac{44}{45}\Delta h\lambda_1 - \frac{56}{15}\Delta h\lambda_2 + \frac{32}{9}\lambda_3, \\ & z(k) + \frac{44}{45}\Delta h\xi_1 - \frac{56}{15}\Delta h\xi_2 + \frac{32}{9}\xi_3)) \\ \xi_4 &= \delta(x(k) + (\frac{44}{45}\Delta h\kappa_1 - \frac{56}{15}\Delta h\kappa_2 + \frac{32}{9}\kappa_3, y(k) + \frac{44}{45}\Delta h\lambda_1 - \frac{56}{15}\Delta h\lambda_2 + \frac{32}{9}\lambda_3, \\ & z(k) + \frac{44}{45}\Delta h\xi_1 - \frac{56}{15}\Delta h\xi_2 + \frac{32}{9}\xi_3))\end{aligned}$$

$$\begin{aligned}\kappa_5 &= f(x(k) + (\Delta h \frac{19372}{6561}\kappa_1 - \Delta h \frac{25360}{2187}\kappa_2 + \Delta h \frac{64448}{6561}\kappa_3 - \Delta h \frac{212}{729}\kappa_4, y(k) \\ & + \Delta h \frac{19372}{6561}\lambda_1 - \Delta h \frac{25360}{2187}\lambda_2 + \Delta h \frac{64448}{6561}\lambda_3 - \Delta h \frac{212}{729}\lambda_4, z(k) + \Delta h \frac{19372}{6561}\xi_1 \\ & - \Delta h \frac{25360}{2187}\xi_2 + \Delta h \frac{64448}{6561}\xi_3 - \Delta h \frac{212}{729}\xi_4)) \\ \lambda_5 &= \rho(x(k) + (\Delta h \frac{19372}{6561}\kappa_1 - \Delta h \frac{25360}{2187}\kappa_2 + \Delta h \frac{64448}{6561}\kappa_3 - \Delta h \frac{212}{729}\kappa_4, y(k) \\ & + \Delta h \frac{19372}{6561}\lambda_1 - \Delta h \frac{25360}{2187}\lambda_2 + \Delta h \frac{64448}{6561}\lambda_3 - \Delta h \frac{212}{729}\lambda_4, z(k) + \Delta h \frac{19372}{6561}\xi_1 \\ & - \Delta h \frac{25360}{2187}\xi_2 + \Delta h \frac{64448}{6561}\xi_3 - \Delta h \frac{212}{729}\xi_4)) \\ \xi_5 &= \delta(x(k) + (\Delta h \frac{19372}{6561}\kappa_1 - \Delta h \frac{25360}{2187}\kappa_2 + \Delta h \frac{64448}{6561}\kappa_3 - \Delta h \frac{212}{729}\kappa_4, y(k) \\ & + \Delta h \frac{19372}{6561}\lambda_1 - \Delta h \frac{25360}{2187}\lambda_2 + \Delta h \frac{64448}{6561}\lambda_3 - \Delta h \frac{212}{729}\lambda_4, z(k) + \Delta h \frac{19372}{6561}\xi_1 \\ & - \Delta h \frac{25360}{2187}\xi_2 + \Delta h \frac{64448}{6561}\xi_3 - \Delta h \frac{212}{729}\xi_4))\end{aligned}$$

$$\begin{aligned}\kappa_6 &= f(x(k) + (\Delta h \frac{9017}{3168}\kappa_1 - \Delta h \frac{355}{33}\kappa_2 + \Delta h \frac{46732}{5247}\kappa_3 + \Delta h \frac{49}{176}\kappa_4 - \Delta h \frac{5103}{18656}\kappa_5, \\ & y(k) + \Delta h \frac{9017}{3168}\lambda_1 - \Delta h \frac{355}{33}\lambda_2 + \Delta h \frac{46732}{5247}\lambda_3 + \Delta h \frac{49}{176}\lambda_4 - \Delta h \frac{5103}{18656}\lambda_5, \\ & z(k) + \Delta h \frac{9017}{3168}\xi_1 - \Delta h \frac{355}{33}\xi_2 + \Delta h \frac{46732}{5247}\xi_3 + \Delta h \frac{49}{176}\xi_4 - \Delta h \frac{5103}{18656}\xi_5))\end{aligned}$$

(Devam) (3.14)

$$\lambda_6 = \rho(x(k) + (\Delta h \frac{9017}{3168} \kappa_1 - \Delta h \frac{355}{33} \kappa_2 + \Delta h \frac{46732}{5247} \kappa_3 + \Delta h \frac{49}{176} \kappa_4 - \Delta h \frac{5103}{18656} \kappa_5,$$

$$y(k) + \Delta h \frac{9017}{3168} \lambda_1 - \Delta h \frac{355}{33} \lambda_2 + \Delta h \frac{46732}{5247} \lambda_3 + \Delta h \frac{49}{176} \lambda_4 - \Delta h \frac{5103}{18656} \lambda_5,$$

$$z(k) + \Delta h \frac{9017}{3168} \xi_1 - \Delta h \frac{355}{33} \xi_2 + \Delta h \frac{46732}{5247} \xi_3 + \Delta h \frac{49}{176} \xi_4 - \Delta h \frac{5103}{18656} \xi_5)$$

$$\xi_6 = \delta(x(k) + (\Delta h \frac{9017}{3168} \kappa_1 - \Delta h \frac{355}{33} \kappa_2 + \Delta h \frac{46732}{5247} \kappa_3 + \Delta h \frac{49}{176} \kappa_4 - \Delta h \frac{5103}{18656} \kappa_5,$$

$$y(k) + \Delta h \frac{9017}{3168} \lambda_1 - \Delta h \frac{355}{33} \lambda_2 + \Delta h \frac{46732}{5247} \lambda_3 + \Delta h \frac{49}{176} \lambda_4 - \Delta h \frac{5103}{18656} \lambda_5,$$

$$z(k) + \Delta h \frac{9017}{3168} \xi_1 - \Delta h \frac{355}{33} \xi_2 + \Delta h \frac{46732}{5247} \xi_3 + \Delta h \frac{49}{176} \xi_4 - \Delta h \frac{5103}{18656} \xi_5)$$

$$\kappa_7 = f(x(k) + (\Delta h \frac{35}{384} \kappa_1 + \Delta h \frac{500}{1113} \kappa_3 + \Delta h \frac{125}{192} \kappa_4 - \Delta h \frac{2187}{6784} \kappa_5 + \Delta h \frac{11}{84} \kappa_6,$$

$$y(k) + \Delta h \frac{35}{384} \lambda_1 + \Delta h \frac{500}{1113} \lambda_3 + \Delta h \frac{125}{192} \lambda_4 - \Delta h \frac{2187}{6784} \lambda_5 + \Delta h \frac{11}{84} \lambda_6,$$

$$z(k) + \Delta h \frac{35}{384} \xi_1 + \Delta h \frac{500}{1113} \xi_3 + \Delta h \frac{125}{192} \xi_4 - \Delta h \frac{2187}{6784} \xi_5 + \Delta h \frac{11}{84} \xi_6)$$

$$\lambda_7 = \rho(x(k) + (\Delta h \frac{35}{384} \kappa_1 + \Delta h \frac{500}{1113} \kappa_3 + \Delta h \frac{125}{192} \kappa_4 - \Delta h \frac{2187}{6784} \kappa_5 + \Delta h \frac{11}{84} \kappa_6,$$

$$y(k) + \Delta h \frac{35}{384} \lambda_1 + \Delta h \frac{500}{1113} \lambda_3 + \Delta h \frac{125}{192} \lambda_4 - \Delta h \frac{2187}{6784} \lambda_5 + \Delta h \frac{11}{84} \lambda_6,$$

$$z(k) + \Delta h \frac{35}{384} \xi_1 + \Delta h \frac{500}{1113} \xi_3 + \Delta h \frac{125}{192} \xi_4 - \Delta h \frac{2187}{6784} \xi_5 + \Delta h \frac{11}{84} \xi_6)$$

$$\xi_7 = \delta(x(k) + (\Delta h \frac{35}{384} \kappa_1 + \Delta h \frac{500}{1113} \kappa_3 + \Delta h \frac{125}{192} \kappa_4 - \Delta h \frac{2187}{6784} \kappa_5 + \Delta h \frac{11}{84} \kappa_6,$$

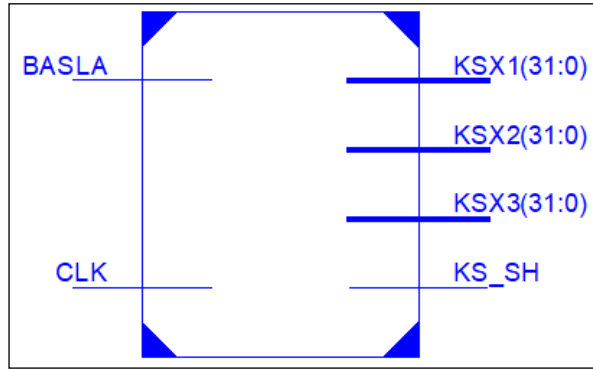
$$y(k) + \Delta h \frac{35}{384} \lambda_1 + \Delta h \frac{500}{1113} \lambda_3 + \Delta h \frac{125}{192} \lambda_4 - \Delta h \frac{2187}{6784} \lambda_5 + \Delta h \frac{11}{84} \lambda_6,$$

$$z(k) + \Delta h \frac{35}{384} \xi_1 + \Delta h \frac{500}{1113} \xi_3 + \Delta h \frac{125}{192} \xi_4 - \Delta h \frac{2187}{6784} \xi_5 + \Delta h \frac{11}{84} \xi_6)$$

4. BULGULAR

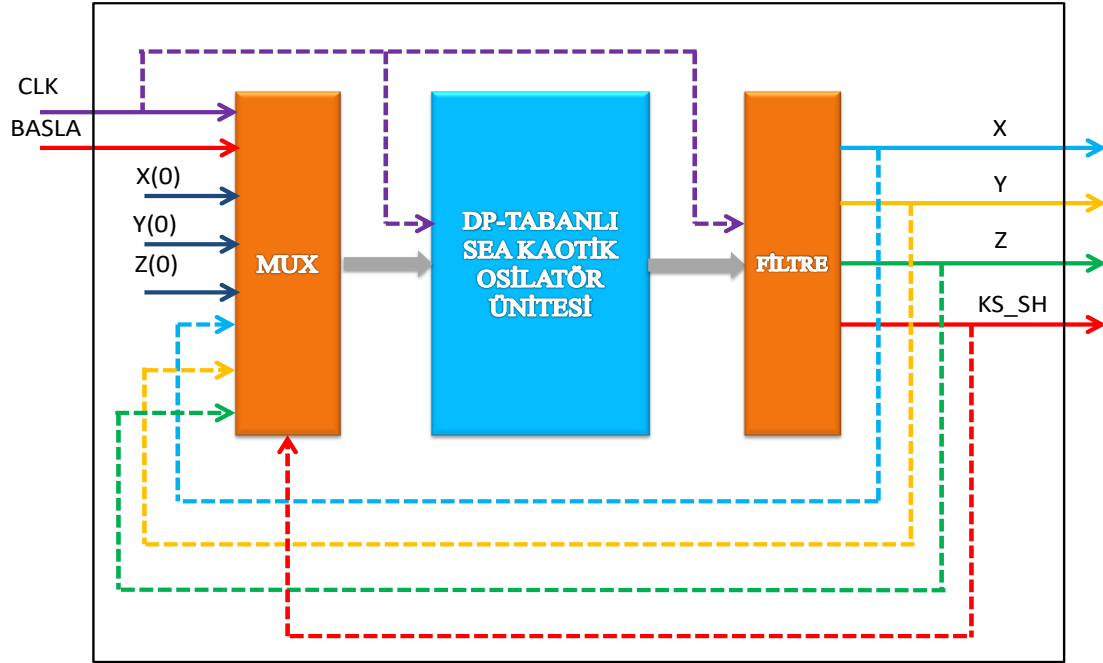
4.1 Kayan Noktalı Sayı Tabanlı SEA Kaotik Sistemi Tasarımı

Yapılan bu tez çalışmasında son zamanlarda literatüre sunulan ve üzerinde FPGA-tabanlı çalışma yapılmamış olan SEA kaotik sistemi seçilmiştir. İlk başta SEA kaotik sistemi ayrıklaştırılmış DP nümerik algoritması kullanılarak 32-bit IEEE 754-1985 kayan noktalı sayı formatına uygun olarak FPGA çipi üzerinde modellenmiştir. Tasarımı gerçekleştirilen SEA kaotik sistemi bir donanım tanımlama dili olan VHDL dili kullanılarak kodlanmıştır. Tasarlanan bu kaotik osilatörün Xilinx ISE 14.7 kullanılarak elde edilen en üst seviye blok diyagramı Şekil 4.1’de verilmiştir.



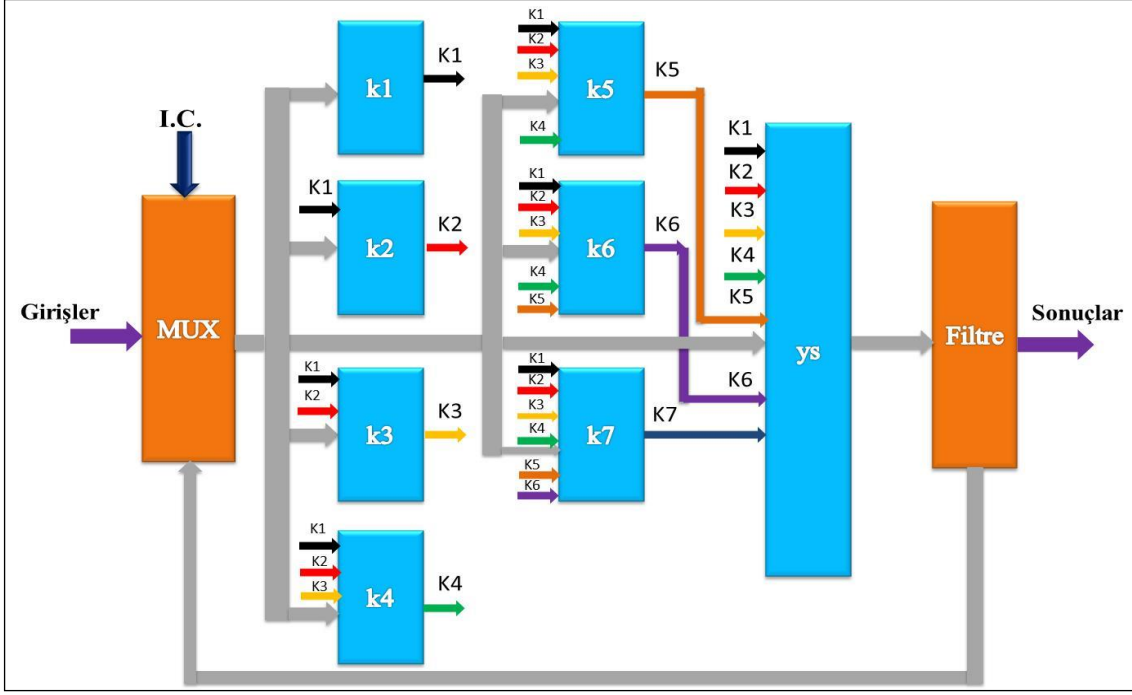
Şekil 4.1 FPGA-tabanlı SEA kaotik osilatörün en üst seviye blok şeması.

Tasarlanan kaotik sistemin üzerinde 1-bit BASLA ve 1-bit CLK giriş sinyalleri bulunmaktadır. 1-bit BASLA sinyali sistemin ve alt ünitelerin çalışması için kontrol sinyalini ifade etmektedir. 1-bit CLK sinyali ise sistem için gerekli olan ünitelerin içindeki her bir alt ünitenin zamanlamasını ve senkronizasyonunu sağlamaktadır. DP nümerik algoritmanın hassasiyetini belirleyen Δh adım sayısı parametresi ve sistemin ilk başlangıç şartları, tasarımda kullanılan çip kaynakları azaltmak için FPGA çipi içerisine sabit veri nesnesi olarak tanımlanmıştır. Sistem çıkışlarında ise 1-bit KS_SH sinyali ve 3 adet 32-bit sonuç değeri sinyali bulunmaktadır. 1-bit KS_SH sinyali ünitenin sonuç üretinceye kadar 0 değerini ve ilk sonuç değerini ürettiği zaman 1 değerini göstermektedir. 3 adet 32-bit kayan noktalı sayı standartlarında kaotik sinyal değerini taşıyan KXS1, KXS2 ve KXS3 sinyalleri bulunmaktadır. Tasarımın ikinci alt seviye blok yapısında Şekil 4.2’de verilmiştir.



Şekil 4.2 FPGA-tabanlı kaotik osilatörün ikinci alt seviye blok şeması.

Şekil 4.2’de SEA kaotik osilatör ünitesinin ikinci seviye blok şeması incelendiğinde bir adet multiplexer ünitesi (MUX), bir adet DP-tabanlı SEA kaotik osilatör ünitesi ve bir adet filtre ünitesi bulunmaktadır. Sunulan tez çalışmasındaki MUX ünitesinin amacı, kullanıcı tarafından verilen ilk başlangıç şartları $X(0)$, $Y(0)$ ve $Z(0)$ değerlerini ilk çalışma zamanında almasını ve bundan sonraki diğer tüm aşamalarda filtre ünitesinin ürettiği çıkışları almasını sağlamaktır. MUX ünitesi bu işlemi filtre ünitesinin 1-bitlik KS_SH çıkış sinyalinin değeri 1 olduğu zaman bu değeri almakta ve bundan sonraki aşamada ise ilk başlangıç değerleri yerine filtrenin ünitesinden çıkan sonuçları kullanmaktadır. Tasarlanan filtre ünitesinin amacı ise istenmeyen sinyalleri filtrelemektir. Başka bir ifadeyle kayan noktalı sayı formatındaki kaotik osilatör ilk değeri 416 clock darbesi sonunda üretmekte ve bu zamana kadar kaotik osilatörün üretmediği sinyallerin dışarı çıkmasını engellemektedir. Kaotik osilatör sistemi çalıştırıldığında her 416 clock darbesi sonundaki sistem çıkışları olan X, Y ve Z 32-bit kayan noktalı sayı formatındaki kaotik değerleri dışarı aktarmaktadır. Şekil 4.3’te DP-tabanlı SEA kaotik osilatörün üçüncü seviye blok şeması verilmiştir.

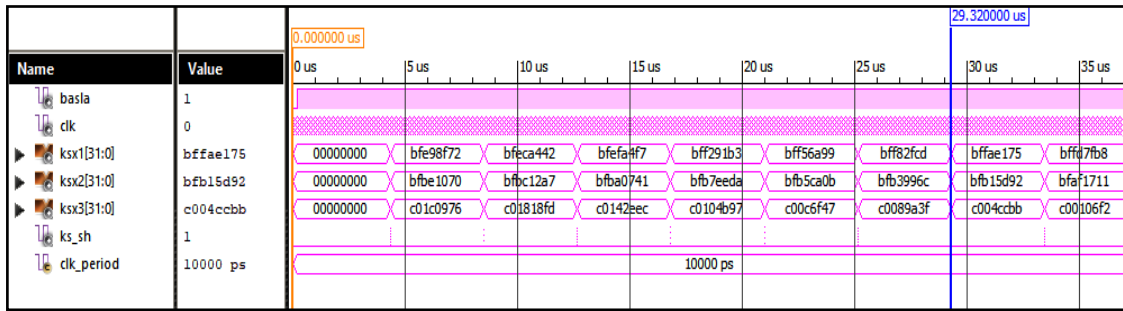


Şekil 4.3 DP-tabanlı SEA kaotik osilatörün üçüncü seviye blok şeması.

Şekil 4.3'te DP-tabanlı SEA kaotik osilatörün üçüncü alt seviye blok şeması incelendiğinde DP-tabanlı SEA kaotik osilatör ünitesinin altındaki $k_1, k_2, k_3, k_4, k_5, k_6, k_7, ys$ ünitelerinden oluştuğu görülmektedir. Bu $k_1, k_2, k_3, k_4, k_5, k_6, k_7$ ünitelerin her biri DP-tabanlı SEA kaotik sistemin ayrıklaştırılmış matematiksel denklemindeki $\kappa_\sigma, \zeta_\sigma$ ve λ_σ değerlerini hesaplamaktadır. Ayrıklaştırılmış DP nümerik algoritmasında verilen $x(k+1), y(k+1)$ ve $z(k+1)$ değerleri ise ys ünitesinde hesaplanmaktadır. Kaotik sistem sırası ile çalıştırıldığında k_1 ünitesi ayrıklaştırılmış modeldeki MUX ünitesinin ürettiği sonuçları kullanarak $\kappa_1, \zeta_1, \lambda_1$ değerleri hesaplamaktadır. k_2 ünitesi, k_1 ve MUX ünitelerinin ürettiği sonuçları kullanarak $\kappa_2, \zeta_2, \lambda_2$ değerlerini hesaplamaktadır. k_3 ünitesi, k_1, k_2 ve MUX ünitelerinin ürettiği sonuçları kullanarak $\kappa_3, \zeta_3, \lambda_3$ değerlerini hesaplamaktadır. k_4 ünitesi, k_1, k_2, k_3 ve MUX ünitelerinin ürettiği sonuçları kullanarak $\kappa_4, \zeta_4, \lambda_4$ değerlerini hesaplamaktadır. k_5 ünitesi, k_1, k_2, k_3, k_4 ve MUX ünitelerinin ürettiği sonuçları kullanarak $\kappa_5, \zeta_5, \lambda_5$ değerlerini hesaplamaktadır. k_6 ünitesi, k_1, k_2, k_3, k_4, k_5 ve MUX ünitelerinin ürettiği sonuçları kullanarak $\kappa_6, \zeta_6, \lambda_6$ değerlerini hesaplamaktadır. k_7 ünitesi, $k_1, k_2, k_3, k_4, k_5, k_6$ ve MUX ünitelerinin ürettiği sonuçları kullanarak $\kappa_7, \zeta_7, \lambda_7$ değerlerini hesaplamaktadır. ys ünitesi ise $k_1, k_2, k_3, k_4, k_5, k_6$ ve k_7 ünitelerinden elde edilen sonuçları kullanarak $x(k+1), y(k+1)$ ve $z(k+1)$ değerlerini hesaplamaktadır.

4.2 Kayan Noktalı Sayı Tabanlı SEA Kaotik Sistemi Tasarımının Sonuçları

Tasarımı gerçekleştirilmiş olan DP-tabanlı SEA kaotik sistem Xilinx Virtex-6 ailesi xc6vlx240t-ff1156-1 çipi üzerinde sentezlenmiştir. Tasarım 32-bit IEEE 754-1985 kayan noktalı sayı standartlarına uygun olarak FPGA üzerinde VHDL dili kullanılarak kodlanmıştır. Tasarımı gerçekleştirilen SEA kaotik osilatör ünitesi için testbench ünitesi oluşturulmuş ve test edilmiştir. Xilinx ISE Design Tools 14.7 programı kullanılarak elde edilen simülasyon sonuçları Şekil 4.4'te verilmiştir.



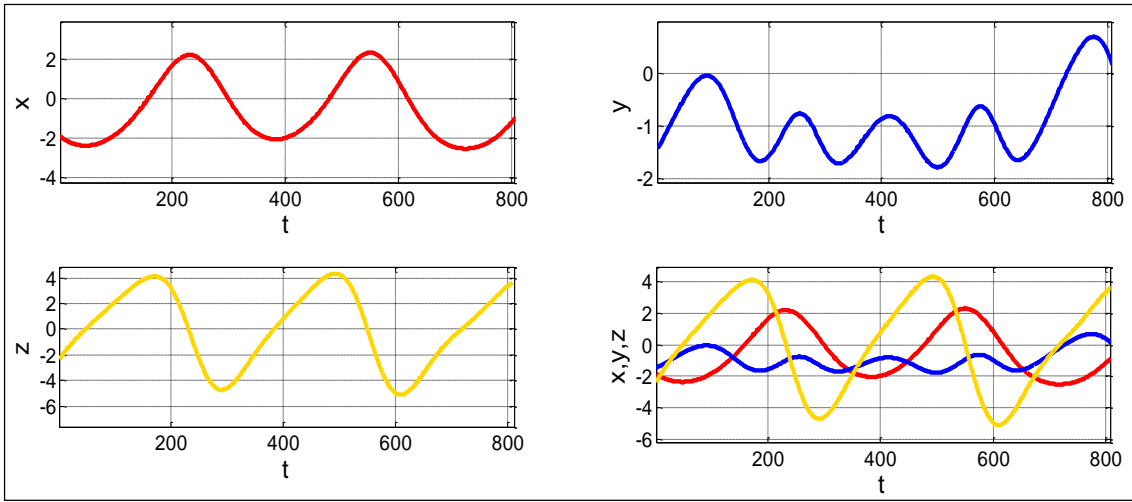
Şekil 4.4 DP-tabanlı SEA Kaotik osilatörün IEEE 754-1985 32-bit kayan noktalı sayı standartlarındaki simülasyon sonuçları.

Tasarımı yapılan sistemin Xilinx ISE Design Tools 14.7 benzetim programı kullanılarak yerleştirme ve bağlama (Place & Route) işlemlerinin ardından elde edilen çip istatistikleri Çizelge 4.1'de verilmiştir.

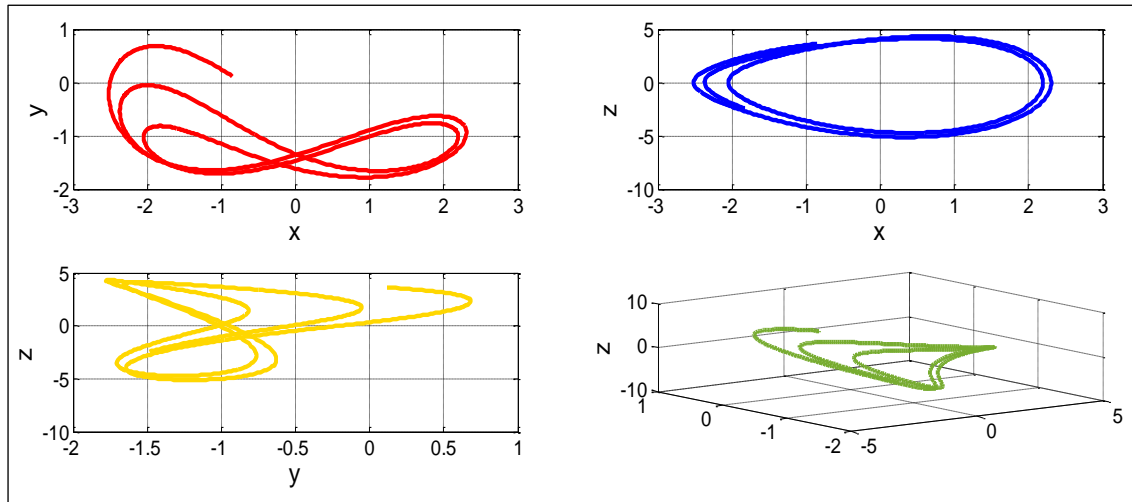
Çizelge 4.1 DP-tabanlı SEA kaotik sistemlerin IEEE 754-1985 32-bit kayan noktalı sayı standartlarındaki FPGA çip kullanım istatistikleri.

Lojik Kullanım	Kullanılan	Mevcut	Kul. Oranı	Çalışma Frekansı (MHz)
Slice Registers Sayısı	115,700	301,440	%37	316.756
Slice LUTs Sayısı	103,747	150,720	%68	
Occupied Slices Sayısı	31,945	37,680	%84	
IOBs Sayısı	99	600	%16	
BUFG/BUFGCTRLs Sayısı	1	32	%3	
DSP48E1s Sayısı	168	768	%21	

SEA kaotik sistemi FPGA üzerinde gerçekleştirilmesinden sonra benzetim aşamasında elde edilen 32-bit IEEE 754-1985 kayan noktalı sayı standartlarına uygun KXS1, KXS2 ve KXS3 ikilik (binary) formatında kaotik sonuç değerleri bir dosyaya kaydedilmiştir. Kaydedilen bu değerler onluk (decimal) sayı sistemine dönüştürülmüştür. FPGA tabanlı kaotik sistemin ürettiği ilk 3x810 veri seti değerleri yardımıyla kaotik sistemin zaman serileri ve faz portreleri elde çizdirilmiştir. FPGA yongası üzerinde gerçekleştirilen kayan noktalı sayı formatına uygun DP-tabanlı SEA kaotik sistemin zaman serisi Şekil 4.5'te ve faz portreleri Şekil 4.6'da verilmiştir.



Şekil 4.5 32-bit kayan nokta sayı tabanlı SEA kaotik osilatörün zaman serileri.



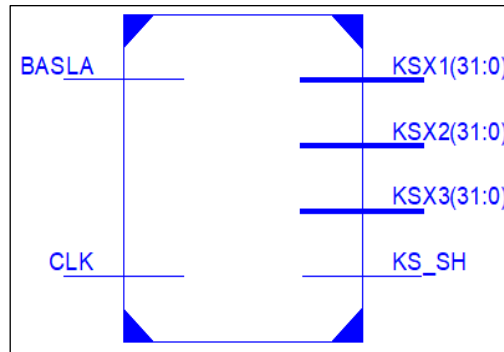
Şekil 4.6 32-bit kayan nokta sayı tabanlı SEA kaotik osilatörün faz portreleri.

4.3 Sabit Noktalı Sayı Tabanlı SEA Kaotik Sistemi Tasarımı ve Sonuçları

Sunulan tez çalışmasında daha önce üzerinde FPGA tabanlı çalışma yapılmamış olan SEA kaotik sistemi 16I-16Q, 14I-14Q, 12I-12Q, 10I-10Q ve 8I-8Q IQ-Math sabit noktalı sayı standartlarına uygun olmak üzere beş farklı sayı formatındaki tasarımları gerçekleştirilmiştir. Her bir tasarım donanım tanımlama dili olan VHDL dilinde kodlanmıştır. Gerçeklenen bu kaotik osilatörler için bir testbench ünitesi oluşturulmuş ve test edilmiştir. Farklı IQ-Math sayı standartlarındaki kaotik osilatörlerin Xilinx ISE Design Tools 14.7 programı kullanılarak elde edilen simülasyon sonuçları ve Place & Route işlemlerinin ardından elde edilen çip istatistikleri verilmiştir. DP-tabanlı farklı IQ-Math sayı standartlarındaki kaotik osilatörlerin ürettiği sonuçlar benzetim aşamasında bir dosya formatına IQ-Math sayı standardına uygun bir şekilde kaydedilmiş ve daha sonra gerçek sayı sistemine dönüştürülmüştür. Kaotik sistemin ürettiği ilk 3x909 veri değeri kullanılarak zaman serileri ve faz portreleri elde edilmiştir.

4.3.1 32-bit (16I-16Q) Sabit Noktalı Sayı Tabanlı SEA Kaotik Sistemi Tasarımı ve Sonuçları

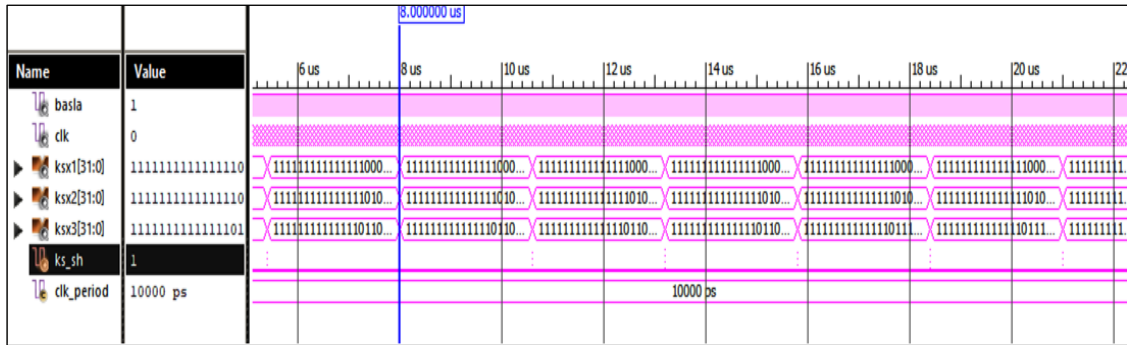
SEA kaotik osilatörü ayrılaştırılmış DP nümerik algoritması kullanılarak 32-bit (16I-16Q) IQ-Math sabit noktalı sayı standardında FPGA yongası üzerinde modellenmiştir. Tasarım sırasında ihtiyaç duyulan toplayıcı, çıkarıcı, çarpıcı ve bölücü gibi temel matematiksel işlemler 16I-16Q sabit noktalı sayı standardına uygun olarak Xilinx ISE Design Tools tarafından geliştirilen IP CORE Generator kullanılarak oluşturulmuş ve tasarım VHDL dilinde kodlanmıştır. Tasarıma ait en üst seviye blok şeması Şekil 4.7’de verilmiştir.



Şekil 4.7 16I-16Q FPGA-tabanlı SEA kaotik osilatörün en üst seviye blok diyagramı.

Kaotik sistem ünitesi üzerinde sistemin başlaması için 1-bitlik BASLA sinyali bulunmaktadır. 1-bit CLK sinyali kaotik sisteme ait ünitelerin senkronizasyonu ve üniteler içindeki her bir alt ünitenin zamanlamasını sağlamaktadır. Tasarlan DP-tabanlı SEA kaotik osilatöründe 3 adet 16I-16Q sabit noktalı sayı standardına uygun çıkış sinyalleri KSX1, KSX2, KSX3 ve bir bitlik KS_SH sonuç hazır sinyali bulunmaktadır. En üst seviyedeki üniteden en alt seviyedeki üniteye kadar her birinde 16I-16Q IQ-Math sabit noktalı sayı standardına uygun olarak VHDL dilinde tasarımı gerçekleştirilmiştir.

Tasarımı yapılan SEA kaotik osilatör sistemi için testbench ünitesi oluşturulmuş ve test edilmiştir. DP-tabanlı SEA kaotik osilatörü Xilinx Virtex-6 ailesi xc6vlx240t-ff1156-1 çipi üzerinde sentezlenmiş ve test edilmiştir. Xilinx ISE Design Tools 14.7 programı kullanılarak elde edilen simülasyon sonuçları Şekil 4.8’de ve Place & Route işlemlerinin ardından elde edilen çip istatistikleri Çizelge 4.2’de verilmiştir.

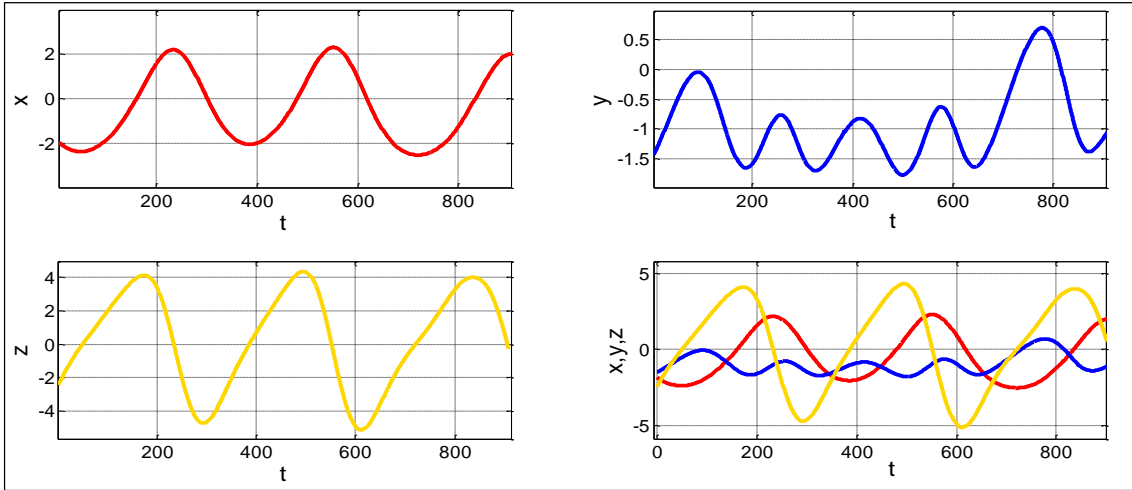


Şekil 4.8 DP-tabanlı 16I-16Q IQ-Math sabit sayı standardındaki SEA kaotik osilatörün Xilinx ISE 14.7 simülasyon sonuçları.

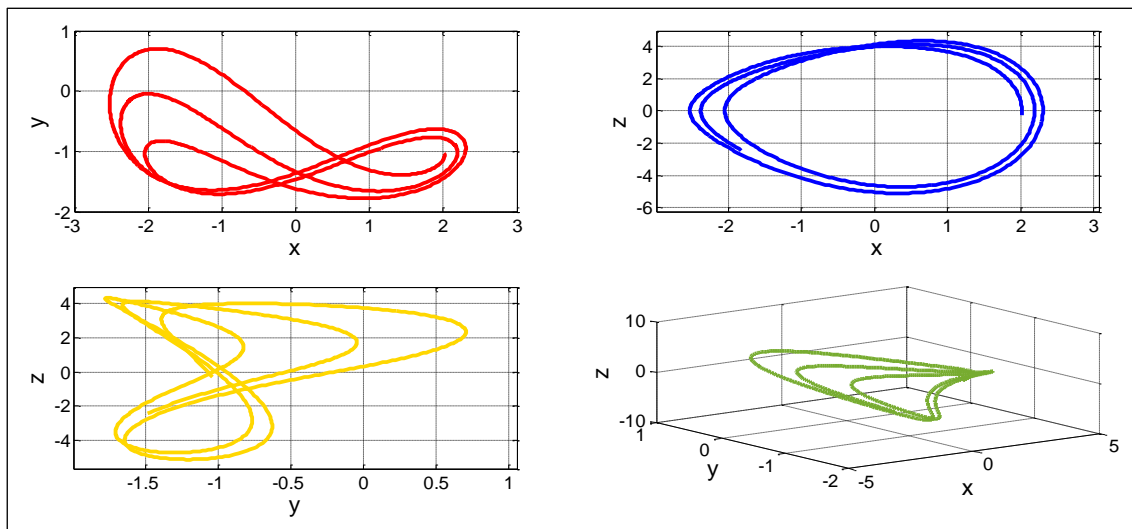
Çizelge 4.2 DP-tabanlı 32 bit (16I-16Q) SEA kaotik sisteminin FPGA çip istatistikleri.

Lojik Kullanım	Kullanılan	Mevcut	Kul. Oranı	Çalışma Frekansı (MHz)
Slice Registers Sayısı	18,280	301,440	%6	
Slice LUTs Sayısı	14,473	150,720	%9	
Memory Sayısı	7,953	58,400	%13	344,585
BUFG/BUFGCTRLs Sayısı	1	32	%3	
DSP48E1s Sayısı	712	768	%92	

DP-tabanlı SEA kaotik sistemin FPGA üzerinde gerçekleştirilmesinden sonra elde edilen KSX1, KSX2 ve KSX3 kaotik sonuç değerini taşıyan 32-bit (16I-16Q) IQ-Math sabit noktalı sayı formatına uygun değerler benzetim aşamasında bir dosyaya kaydedilmiştir. Kaydedilen bu 16I-16Q IQ-Math formatına uygun değerler onluk sayı sistemine dönüştürülmüştür. FPGA tabanlı kaotik sistemin ürettiği ilk 3x909 veri seti değerleri yardımıyla kaotik sistemin zaman serileri ve faz portreleri elde edilmiştir. 32-bit (16I-16Q) IQ-Math sabit noktalı sayı formatında DP-tabanlı SEA kaotik sistemin FPGA üzerinde gerçekleştirilmesinden elde edilen zaman serisi Şekil 4.9’da faz portreleri Şekil 4.10’da verilmiştir.



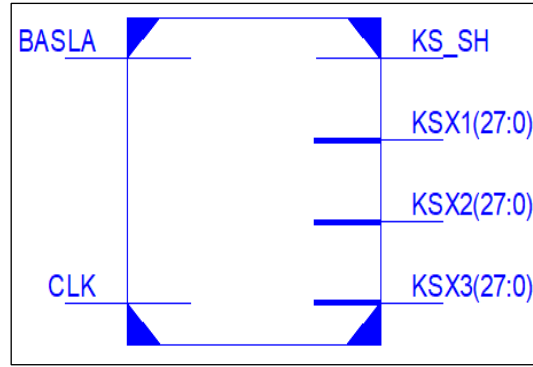
Şekil 4.9 FPGA üzerinde 32-bit (16I-16Q) DP-tabanlı SEA kaotik osilatörün zaman serileri.



Şekil 4.10 FPGA üzerinde 32-bit (16I-16Q) DP-tabanlı SEA kaotik osilatörün faz portreleri.

4.3.2 28-bit (14I-14Q) Sabit Noktalı Sayı Tabanlı SEA Kaotik Sistemi Tasarımı ve Sonuçları

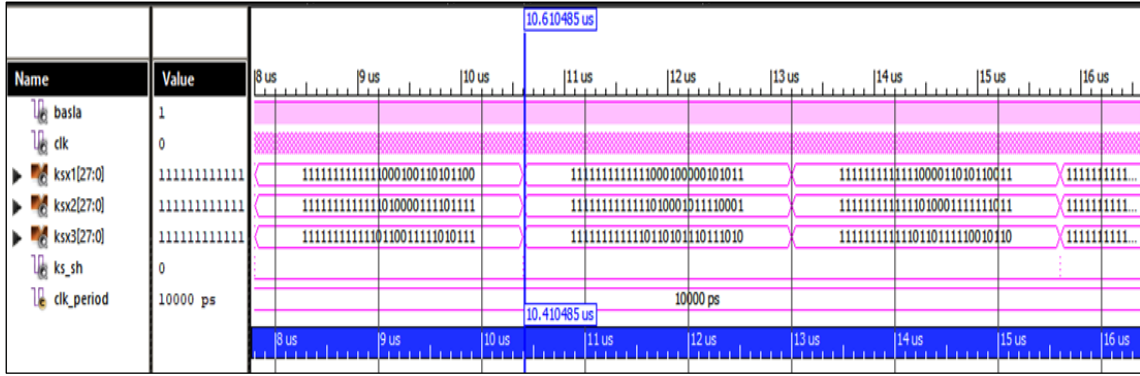
Sunulan çalışmada 28-bit (14I-14Q) IQ-Math sabit noktalı sayı standardına uygun SEA kaotik sistemi FPGA yongası üzerinde modellenmiştir. Tasarım gerçekleştirilecek olan kaotik osilatörün hem tam sayı kısmı hem de kesirli kısmı hassasiyeti ikişer bit azaltılmıştır. Başka bir ifade ile 14-bit tam sayı kısmı 14-bit kesirli kısmı olmak üzere 28-bit hassasiyetine sahip bir kaotik sistem tasarımı yapılmıştır. Tasarıma ait en üst seviye blok şeması Şekil 4.11’de verilmiştir.



Şekil 4.11 14I-14Q FPGA-tabanlı SEA kaotik osilatörün en üst seviye blok diyagramı.

Kaotik sistem ünitesi üzerindeki 2 giriş sinyali ve 4 çıkış sinyali bulunmaktadır. 1 bitlik CLK sinyali sistemin senkronizasyonunu sağlamaktadır. 1 bitlik BASLA sinyali sistemin başlaması için gereken sinyaldir. Tasarlan DP-tabanlı SEA kaotik osilatörün KSX1(27:0), KSX2(27:0), KSX3(27:0), 3 adet 28 bit 14I-14Q sabit noktalı sayı standardına uygun çıkış sinyalleri ve bir bitlik KS_SH sonuç hazır sinyali bulunmaktadır. Ünitenin ve alt ünitelerin her birimi 14I-14Q IQ-Math sabit noktalı sayı standardına uygun olarak VHDL dilinde tasarımı gerçekleştirilmiştir.

28-bit (14I-14Q) IQ-Math sabit noktalı sayı standartlarına uygun DP-tabanlı SEA kaotik sistemi Xilinx Virtex-6 ailesi xc6vlx240t-ff1156-1 çipi üzerinde sentezlenmiştir. Tasarımı gerçekleştirilen bu kaotik osilatör sistemi için testbench ünitesi oluşturulmuş ve test edilmiştir. Xilinx ISE Design Tools 14.7 programı kullanılarak elde edilen 28-bit (14I-14Q) IQ-Math sabit sayı formatındaki KSX1, KSX2 ve KSX3 sinyallerinin simülasyon sonuçları Şekil 4.12’de verilmiştir.



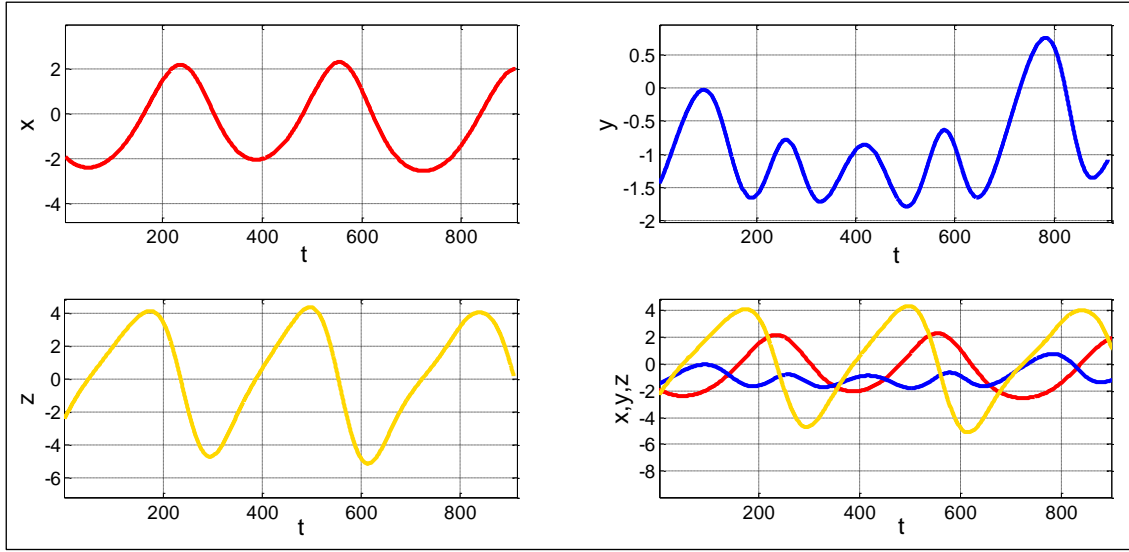
Şekil 4.12 DP-tabanlı 14I-14Q IQ-Math sabit sayı standardındaki SEA kaotik osilatörün Xilinx ISE 14.7 simülasyon sonuçları.

Tasarımı gerçekleştirilen 14I-14Q IQ-Math sabit sayı formatındaki DP-tabanlı SEA kaotik sistemi Xilinx ISE Design Tools 14.7 benzetim programı kullanılarak Place & Route işlemlerinin ardından elde edilen çip istatistikleri Çizelge 4.3'te verilmiştir.

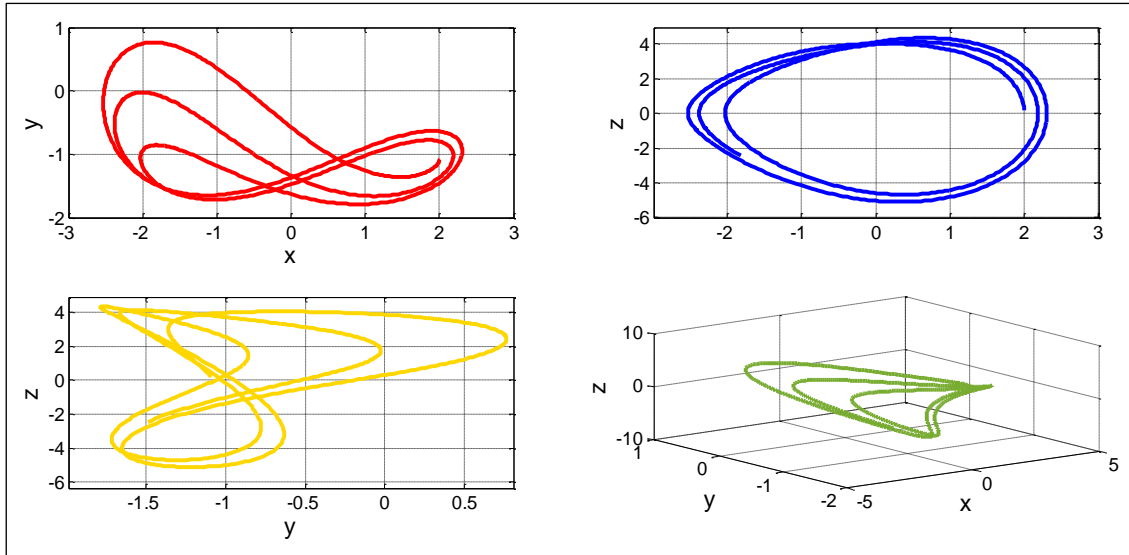
Çizelge 4.3 DP-tabanlı 28 bit (14I-14Q) SEA kaotik sisteminin FPGA çip istatistikleri.

Lojik Kullanım	Kullanılan	Mevcut	Kul. Oranı	Çalışma Frekansı (MHz)
Slice Registers Sayısı	16,529	301,440	%5	349,599
Slice LUTs Sayısı	13,020	150,720	%8	
Memory Sayısı	7,247	58,400	%12	
BUFG/BUFGCTRLs Sayısı	1	32	%3	
DSP48E1s Sayısı	712	768	%92	

DP-tabanlı SEA kaotik sistemin FPGA üzerinde gerçekleştirilmesinden sonra elde edilen KSX1, KSX2 ve KSX3 kaotik sonuç değerini taşıyan 28-bit (14I-14Q) IQ-Math sabit noktalı sayı formatına uygun değerler benzetim aşamasında bir dosyaya kaydedilmiştir. Kaydedilen 14I-14Q IQ-Math sabit sayı formatına uygun değerler onluk sayı sistemine dönüştürülmüştür. SEA kaotik osilatörün ürettiği ilk 3x909 veri seti değerleri yardımıyla kaotik sistemin zaman serileri ve faz portreleri elde edilmiştir. DP-tabanlı SEA kaotik sistemin 28-bit (14I-14Q) IQ-Math sabit noktalı sayı formatına uygun olarak FPGA üzerinde gerçekleştirilmesinden elde edilen zaman serisi Şekil 4.13'te faz portreleri Şekil 4.14'te verilmiştir.



Şekil 4.13 FPGA üzerinde 28-bit (14I-14Q) DP-tabanlı SEA kaotik osilatörün zaman serileri.

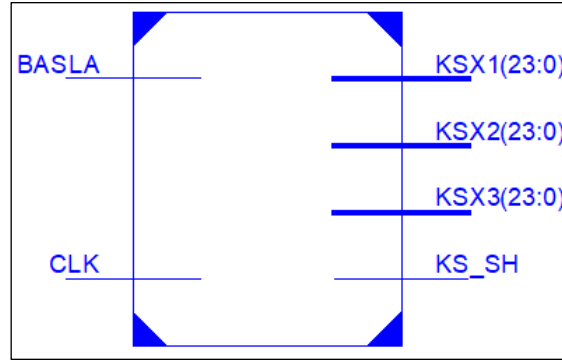


Şekil 4.14 FPGA üzerinde 28-bit (14I-14Q) DP-tabanlı SEA kaotik osilatörün faz portreleri.

4.3.3 24-bit (12I-12Q) Sabit Noktalı Sayı Tabanlı SEA Kaotik Sistemi Tasarımı ve Sonuçları

SEA kaotik sistemi 24-bit (12I-12Q) IQ-Math sabit noktalı sayı standardına uygun olarak DP nümerik algoritması kullanılarak FPGA-tabanlı tasarımı gerçekleştirilmiştir. Bir başka deyişle 12-bit tam sayı kısmı 12-bit kesirli kısmı olmak üzere 24-bit hassasiyetine sahip bir kaotik sistem tasarımı yapılmıştır. Tasarım Xilinx ISE 14.7 Design Tools programı kullanılarak donanım tanımla dili olan VHDL dilinde kodlanmış ve tasarıma ait en üst

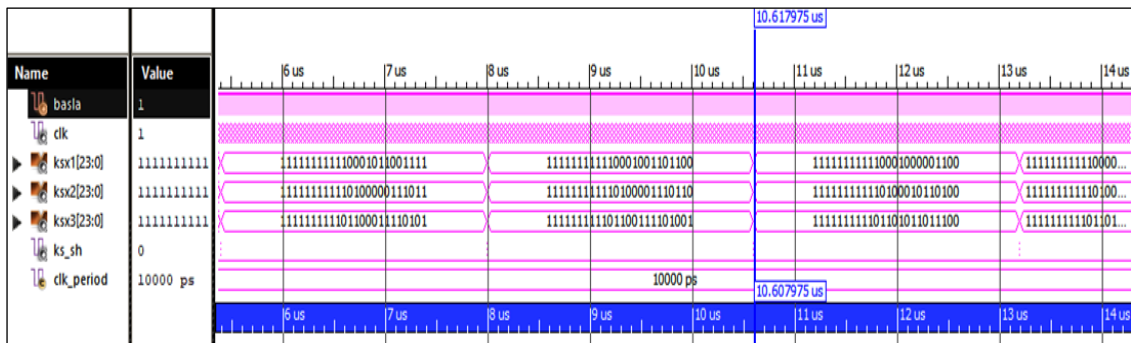
seviye blok şeması Şekil 4.15'te verilmiştir.



Şekil 4.15 12I-12Q FPGA-tabanlı SEA kaotik osilatörün en üst seviye blok diyagramı.

Sistemin çalışması için gerekli olan 1-bit BASLA sinyali ve senkronizasyon için gerekli olan 1-bit CLK saat sinyali bulunmaktadır. SEA kaotik osilatörün çıkışlarında KSX1(23:0), KSX2(23:0), KSX3(23:0) sinyalleri ise 24-bit (12I-12Q) IQ-Math sabit sayı formatına uygun kaotik sonuç değerini taşıyan sinyallerdir.

DP-tabanlı SEA kaotik sistem Xilinx Virtex-6 ailesi xc6vlx240t-ff1156-1 çipi üzerinde sentezlenmiş ve test edilmiştir. Tasarım 24-bit (12I-12Q) IQ-Math sabit noktalı sayı standartlarına uygun olarak FPGA üzerinde VHDL dili kullanılarak kodlanmıştır. Xilinx ISE Design Tools 14.7 programı kullanılarak elde edilen simülasyon sonuçları Şekil 4.16'da verilmiştir.



Şekil 4.16 DP-tabanlı 12I-12Q IQ-Math sabit sayı standardındaki SEA kaotik osilatörün Xilinx ISE 14.7 simülasyon sonuçları.

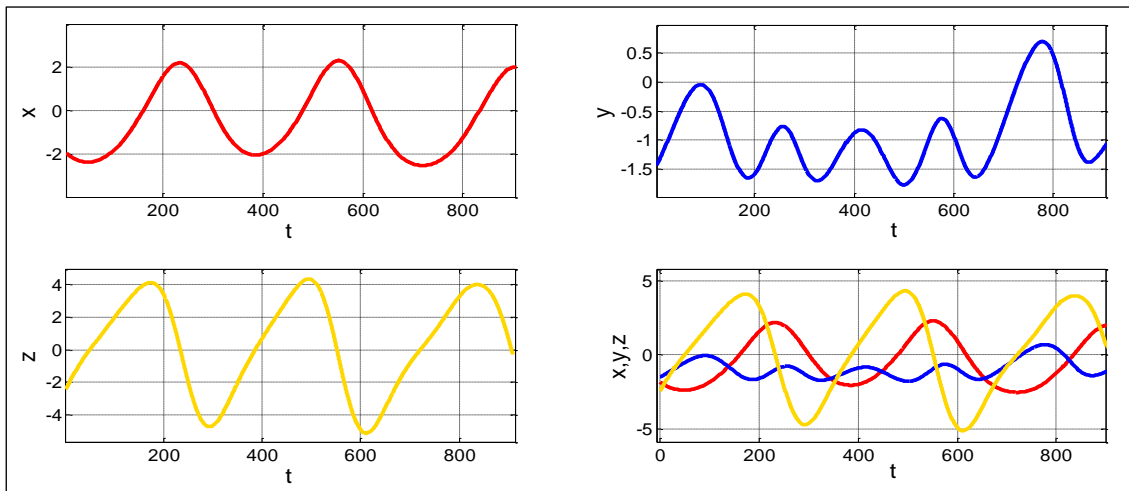
Tasarımı yapılan 12I-12Q IQ-Math sabit noktalı sayı standardındaki kaotik sistemin Xilinx ISE Design Tools 14.7 benzetim programı kullanılarak Place & Route

işlemlerinin ardından elde edilen çip istatistikleri incelenmiş ve Çizelge 4.4'te verilmiştir.

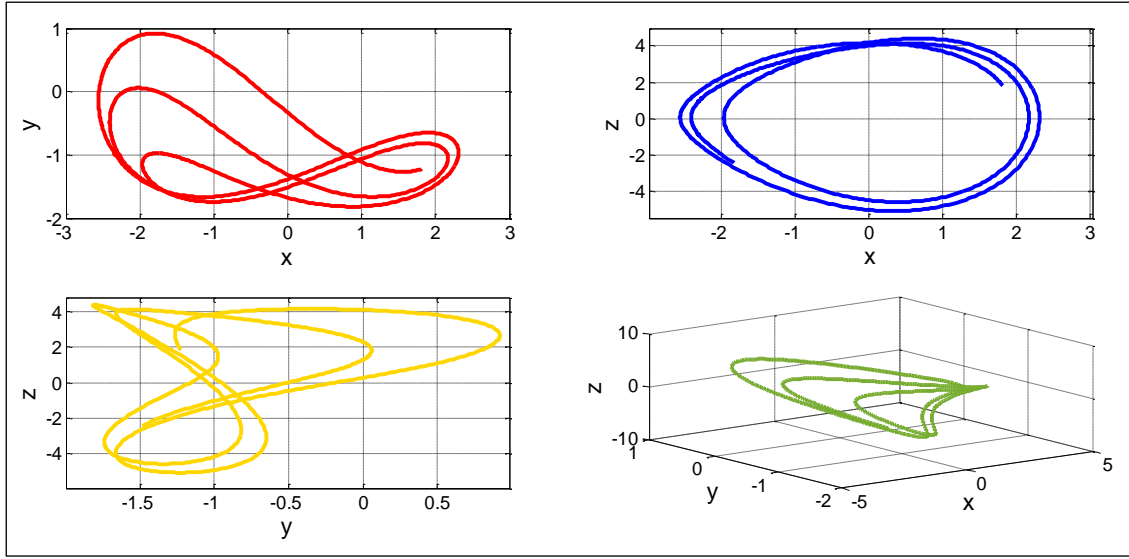
Çizelge 4.4 DP-tabanlı 24 bit (12I-12Q) SEA kaotik sisteminin FPGA çip istatistikleri.

Lojik Kullanım	Kullanılan	Mevcut	Kul. Oranı	Çalışma Frekansı (MHz)
Slice Registers Sayısı	13,888	301,440	%4	354,762
Slice LUTs Sayısı	11,773	150,720	%7	
Memory Sayısı	6,747	58,400	%11	
BUFG/BUFGCTRLs Sayısı	1	32	%3	
DSP48E1s Sayısı	356	768	%46	

DP-tabanlı SEA kaotik sisteminin FPGA üzerinde gerçekleştirilmesinden sonra elde edilen KSX1, KSX2 ve KSX3 kaotik sonuç değerini taşıyan 24-bit (12I-12Q) IQ-Math sabit noktalı sayı formatına uygun ikilik değerler benzetim aşamasında bir dosyaya kaydedilmiştir. Kaydedilen sabit noktalı sayı formatına uygun değerler onluk sayı sistemine dönüştürülmüştür. FPGA-tabanlı kaotik sistemin ürettiği ilk 3x909 veri seti değerleri yardımıyla kaotik sistemin zaman serileri ve faz portreleri elde edilmiştir. FPGA-tabanlı kaotik sistemin zaman serisi Şekil 4.17'de faz portreleri Şekil 4.18'de verilmiştir.



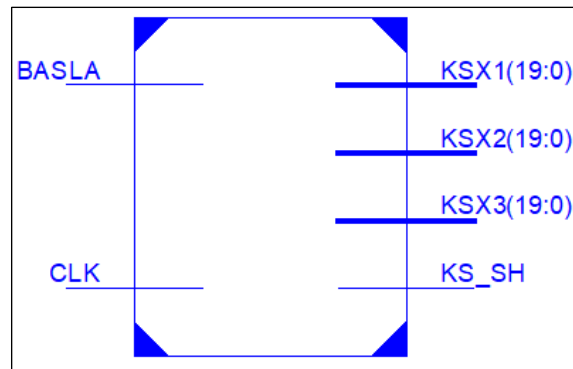
Şekil 4.17 FPGA üzerinde 24-bit (12I-12Q) DP-tabanlı SEA kaotik osilatörün zaman serileri.



Şekil 4.18 FPGA üzerinde 24-bit (12I-12Q) DP-tabanlı SEA kaotik osilatörün faz portreleri.

4.3.4 20-bit (10I-10Q) Sabit Noktalı Sayı Tabanlı SEA Kaotik Sistemi Tasarımı ve Sonuçları

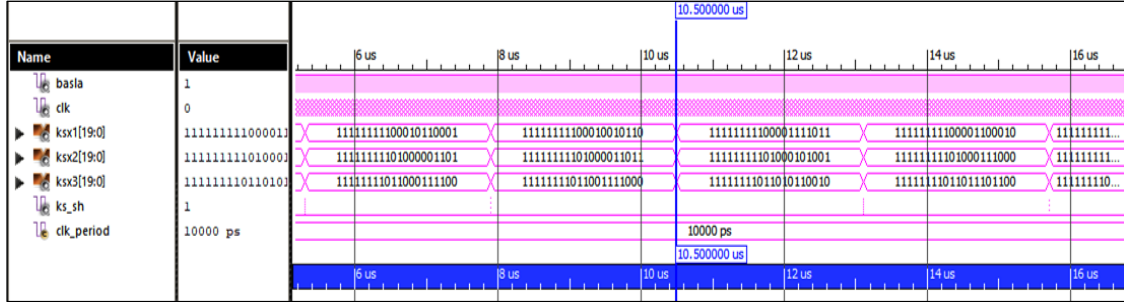
Ayrıklaştırılmış DP nümerik algoritması kullanılarak SEA kaotik osilatörü FPGA yongası üzerinde çalışmak üzere 20-bit (10I-10Q) sabit noktalı sayı formatına uygun tasarımı gerçekleştirilmiştir. Tasarım Xilinx ISE 14.7 Design Tools programı kullanılarak donanım tanımla dili olan VHDL dilinde kodlanmış ve tasarıma ait en üst seviye blok şeması Şekil 4.19'da verilmiştir.



Şekil 4.19 10I-10Q FPGA-tabanlı SEA kaotik osilatörün en üst seviye blok diyagramı.

Tasarımı gerçekleştirilen SEA kaotik osilatör ünitesinin en üst seviye blok diyagramında sistem çıkışlarındaki KSX1(19:0), KSX2(19:0), KSX3(19:0) sinyalleri 20-bit 10I-10Q sabit noktalı sayı standartlarına uygun kaotik sonuç değerlerini taşımaktadır. FPGA-tabanlı

SEA kaotik sistemi VHDL dilinde kodlanmış ve Xilinx Virtex-6 ailesi xc6vlx240t-ff1156-1 çipi üzerinde sentezlenmiştir. Tasarımı gerçekleştirilen sistem için testbench ünitesi oluşturulmuş ve test edilmiştir. Xilinx ISE 14.7 programı kullanılarak elde edilen simülasyon sonuçları Şekil 4.20’de verilmiştir.



Şekil 4.20 DP-tabanlı 10I-10Q IQ-Math sabit sayı standardındaki SEA kaotik osilatörün Xilinx ISE 14.7 simülasyon sonuçları.

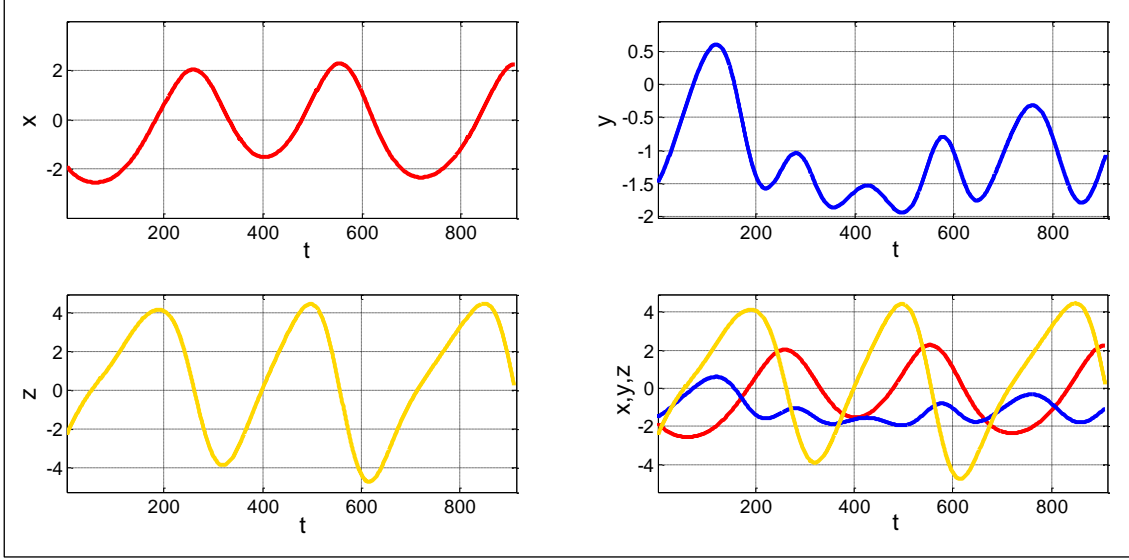
Tasarımı yapılan 10I-10Q IQ-Math sabit noktalı sayı standardındaki kaotik sistemin Place & Route işlemlerinin ardından elde edilen çip istatistikleri incelenmiş ve Çizelge 4.5’te verilmiştir.

Çizelge 4.5 DP-tabanlı 20 bit (10I-10Q) SEA kaotik sisteminin FPGA çip istatistikleri.

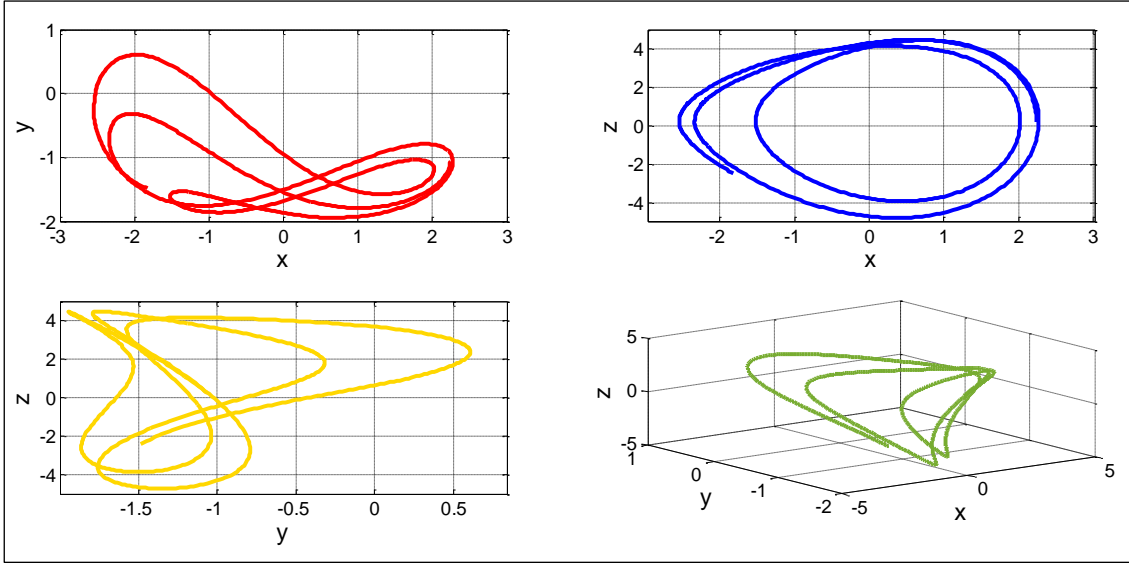
Lojik Kullanım	Kullanılan	Mevcut	Kul. Oranı	Çalışma Frekansını (MHz)
Slice Registers Sayısı	11,575	301,440	%3	360,080
Slice LUTs Sayısı	10,260	150,720	%6	
Memory Sayısı	6,144	58,400	%10	
BUFG/BUFGCTRLs Sayısı	1	32	%3	
DSP48E1s Sayısı	356	768	%46	

DP-tabanlı SEA kaotik sisteminin FPGA üzerinde gerçekleştirilmesinden sonra elde edilen KSX1, KSX2 ve KSX3 kaotik sonuç değerini taşıyan 20-bit (10I-10Q) IQ-Math sabit noktalı sayı formatına uygun bir şekilde ikilik değerler benzetim aşamasında bir dosyaya kaydedilmiştir. Kaydedilen bu 10I-10Q IQ-Math formatına uygun değerler onluk sayı sistemine dönüştürülmüştür. FPGA tabanlı kaotik sistemin ürettiği ilk 3x909 veri seti değerleri yardımıyla kaotik sistemin zaman serileri ve faz portreleri elde

edilmiştir. DP-tabanlı SEA kaotik sistemin FPGA üzerinde gerçekleştirilmesinden elde edilen zaman serisi Şekil 4.21’de faz portreleri Şekil 4.22’de verilmiştir.



Şekil 4.21 FPGA üzerinde 20-bit (10I-10Q) DP-tabanlı SEA kaotik osilatörün zaman serileri.

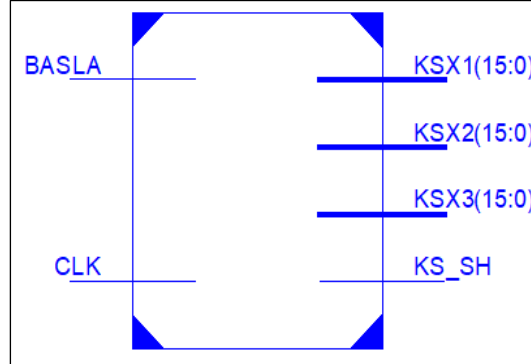


Şekil 4.22 FPGA üzerinde 20-bit (10I-10Q) DP-tabanlı SEA kaotik osilatörün faz portreleri.

4.3.5 16-bit (8I-8Q) Sabit Noktalı Sayı Tabanlı SEA Kaotik Sistemi Tasarımı ve Sonuçları

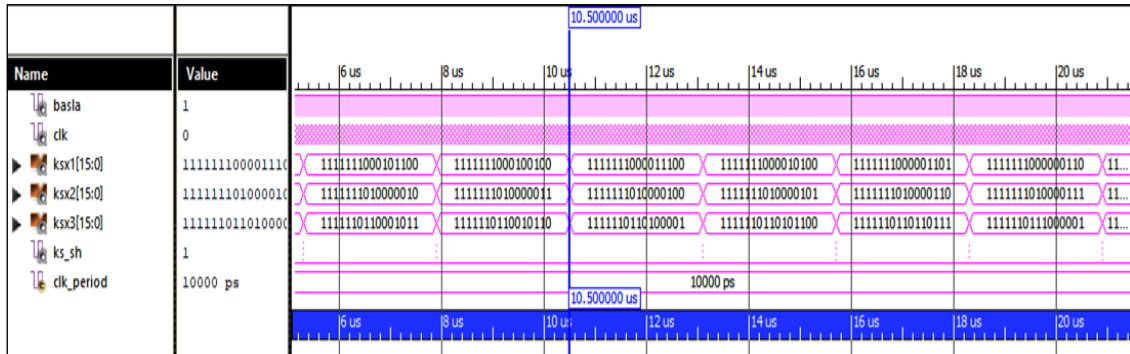
Sunulan tez çalışmasında son olarak SEA kaotik sistemi 8-bit tam sayı kısmı 8-bit kesirli kısmı olmak üzere 16-bit hassasiyetine sahip IQ-Math sabit noktalı sayı standardına uygun

olarak FPGA yongası üzerinde gerçekleştirilmiştir. Tasarım ayrıklaştırılmış DP nümerik algoritması kullanılarak donanım tanımlama dili olan VHDL ile kodlanmıştır. Tasarıma ait en üst seviye blok şeması Şekil 4.23'te verilmiştir. Şekil 4.23 incelendiğinde sistem çıkışında kaotik sistemin sonuç değerlerini taşıyan KSX1(15:0), KSX2(15:0), KSX3(15:0) sinyalleri 16-bit (8I-8Q) IQ-Math sabit sayı formatına uygun şekilde ayarlanmıştır.



Şekil 4.23 8I-8Q FPGA-tabanlı SEA kaotik osilatörün en üst seviye blok diyagramı.

SEA kaotik sistem Xilinx Virtex-6 ailesi xc6vlx240t-ff1156-1 çipi üzerinde sentezlenmiş ve testbench ünitesi oluşturularak test edilmiştir. Tasarımı yapılan SEA kaotik osilatör sistemi için Xilinx ISE Design Tools 14.7 programı kullanılarak elde edilen simülasyon sonuçları Şekil 4.24'te verilmiştir.



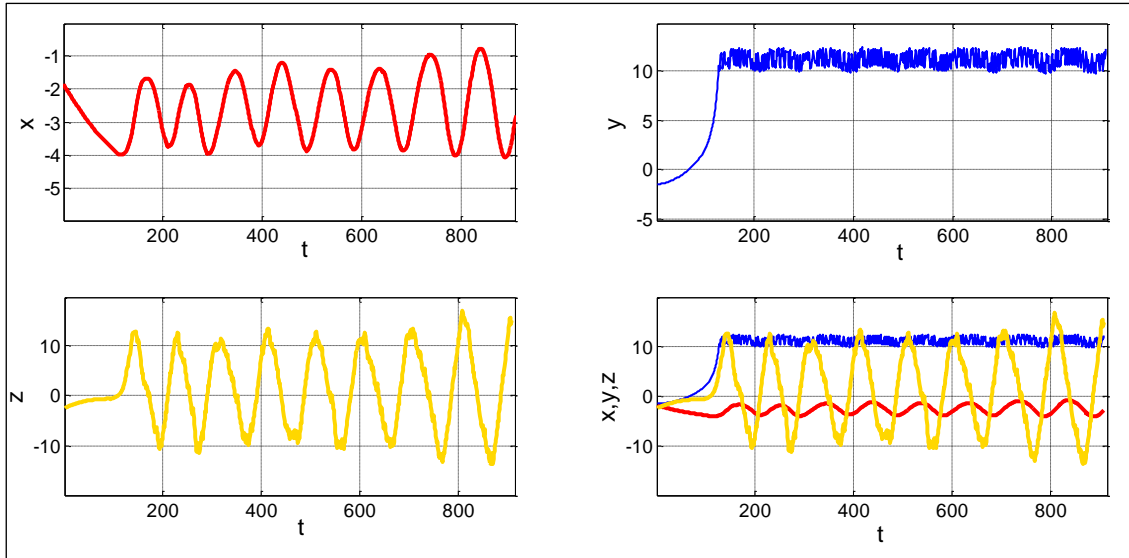
Şekil 4.24 DP-tabanlı 8I-8Q IQ-Math sabit sayı standardındaki SEA kaotik osilatörün Xilinx ISE 14.7 simülasyon sonuçları.

Tasarımı gerçekleştirilen 8I-8Q IQ-Math sabit sayı formatındaki DP-tabanlı SEA kaotik sistemi Xilinx ISE Design Tools 14.7 benzetim programı kullanılarak Place & Route işlemlerinin ardından elde edilen çip istatistikleri Çizelge 4.6'da verilmiştir.

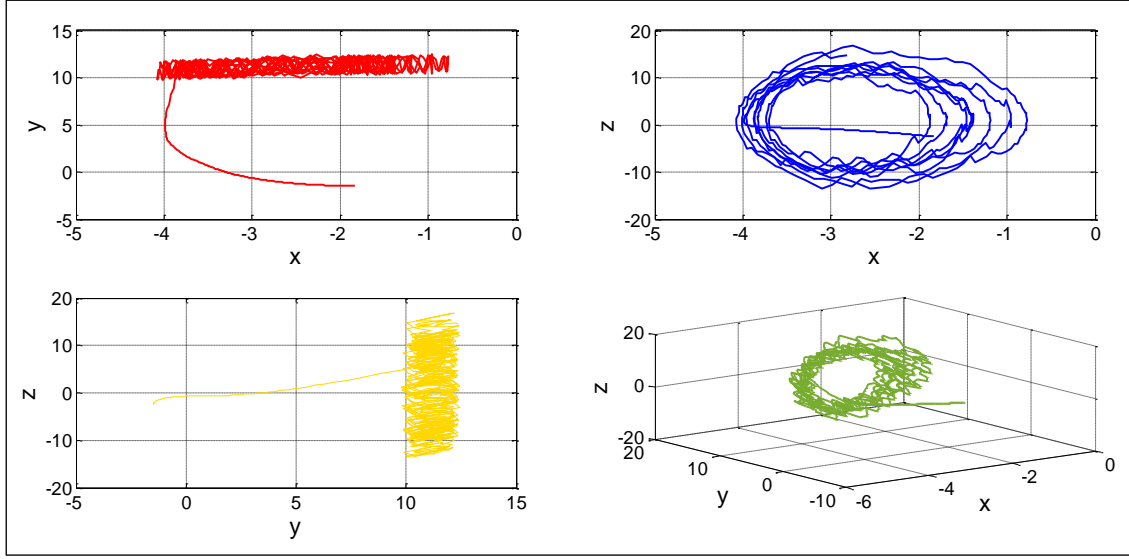
Çizelge 4.6 DP-tabanlı 16 bit (8I-8Q) SEA kaotik sisteminin FPGA çip istatistikleri.

Lojik Kullanım	Kullanılan	Mevcut	Kul. Oranı	Çalışma Frekansı (MHz)
Slice Registers Sayısı	12,747	301,440	%4	365,559
Slice LUTs Sayısı	13,550	150,720	%8	
Memory Sayısı	10,169	58,400	%17	
BUFG/BUFGCTRLs Sayısı	1	32	%3	
DSP48E1s Sayısı	178	768	%23	

DP-tabanlı SEA kaotik sisteminin FPGA üzerinde gerçekleştirilmesinden sonra elde edilen 16-bit (8I-8Q) IQ-Math formatına uygun KSX1, KSX2 ve KSX3 kaotik sonuç değerini taşıyan ikilik değerler benzetim aşamasında bir dosyaya kaydedilmiştir. Kaydedilen 8I-8Q IQ-Math sayı formatına uygun değerler onluk sayı sistemine dönüştürülmüştür. FPGA tabanlı SEA kaotik sistemin ürettiği ilk 3x909 veri seti değerleri yardımıyla kaotik sistemin zaman serileri ve faz portreleri elde edilmiştir. FPGA üzerinde gerçekleştirilmesinden elde edilen zaman serisi Şekil 4.25'te faz portreleri Şekil 4.26'da verilmiştir.



Şekil 4.25 FPGA üzerinde 16-bit (8I-8Q) DP-tabanlı SEA kaotik osilatörün zaman serileri.



Şekil 4.26 FPGA üzerinde 16-bit (8I-8Q) DP-tabanlı SEA kaotik osilatörün faz portreleri.

4.4 FPGA Tabanlı SEA Kaotik Sistemlerin Hata Analizleri

FPGA-tabanlı 16I-16Q, 14I-14Q, 12I-12Q, 10I-10Q, 8I-8Q olmak üzere beş farklı sabit sayı formatında modellenen DP-tabanlı SEA kaotik osilatörlerin sonuçlarının doğruluğunu test etmek amacıyla RMSE (Root Mean Square Error (Ortalama Karesel Hatanın Karekökü)) ve MSE (Mean Square Error (Ortalama Karesel Hata)) yöntemleri kullanılmıştır. RMSE denklemi (4.1)'de verilmiştir (Mert vd. 2014).

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{x}_i - x_i)^2} \quad (4.1)$$

Bu denkleme göre \hat{x}_i gerçek değerleri, x_i tahmini değer ya da elde edilen değerler olarak değerlendirilebilir. n ise yapılacak hata analizinde kullanılacak örnek sayısını ifade etmektedir. MSE matematiksel eşitliği ise denklem (4.2)'de verilmiştir (Ersin 2011).

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (\hat{x}_i - x_i)^2 \quad (4.2)$$

MSE matematiksel eşitliği incelendiğinde, RMSE denklemine benzer bir yapıya sahiptir. Aralarındaki tek fark MSE denkleminde kök alma işlemi yoktur. FPGA-tabanlı yapılan farklı IQ-Math sayı formatındaki kaotik osilatörlerin RMSE ve MSE hata analiz değerlendirmesi Matlab-tabanlı DP algoritması kullanılarak modellenen SEA kaotik sistemi referans kabul edilerek yapılmıştır ve Çizelge 4.7’de verilmiştir.

Çizelge 4.7 FPGA tabanlı sabit noktalı sayı formatındaki kaotik osilatör tasarımlarının MSE ve RMSE hata analizleri.

IQ-Math sabit noktalı sayı formatı	Çıkış Sinyalleri	MSE	RMSE
16I-16Q	X	1,23E-05	3,52E-03
	Y	7,08E-06	2,66E-03
	Z	1,75E-05	4,19E-03
14I-14Q	X	2,82E-04	1,68E-02
	Y	1,38E-04	1,17E-02
	Z	5,99E-04	2,45E-02
12I-12Q	X	4,38E-03	6,62E-02
	Y	1,90E-03	4,36E-02
	Z	1,04E-02	1,02E-01
10I-10Q	X	4,93E-02	2,22E-01
	Y	3,94E-02	1,99E-01
	Z	6,24E-02	2,50E-01
8I-8Q	X	8,79E-01	9,38E-01
	Y	4,60E-01	6,78E-01
	Z	1,7756069	1,33252

Elde edilen sonuçlara göre 32-bit (16I-16Q) IQ-Math sabit sayı formatının en hassas ve en doğru çözüm ürettiği görülmektedir. 16-bit (8I-8Q) IQ-Math sabit sayı formatındaki kaotik sistem ise hatalı sonuç ürettiği görülmektedir. 8I-8Q formatındaki kaotik sistemin faz portreleri ve zaman serileri incelendiği zaman kaotik davranış sergilemediği görülmektedir.

5. TARTIŞMA ve SONUÇ

Kaotik sinyal üreten kaotik osilatör yapısı kriptoloji, güvenli haberleşme, endüstriyel kontrol, yapay sinir ağları, rasgele sayı üreteçleri ve görüntü işleme gibi kaos tabanlı uygulamalarda kullanılan en temel yapıdır. Kaotik sistemler diferansiyel denklemlerle ifade edilmekte ve diferansiyel denklemlerin çözümü için literatürde Euler, Heun, RK4, RK5-Butcher, DP nümerik algoritmaları kullanılmaktadır. Güncel literatür incelendiğinde Euler, Heun, RK4 ve RK5-Butcher algoritmaları kullanılarak yapılan kaotik sistemler rastlanmıştır. Sunulan tez çalışmasında bu nümerik yöntemlerden farklı ve daha hassas çözümler üreten ayrıklaştırılmış DP nümerik algoritması kullanılarak FPGA üzerinde 32-bit IEEE 754-1985 kayan noktalı sayı standardında ve IQ-Math sabit noktalı sayı standartlarına uygun biçimde SEA kaotik sistemi modellenmiştir.

Sunulan tez çalışmasının ilk aşamasında SEA kaotik sistem DP algoritması ile Matlab-tabanlı modellenmiş ve sistemin kaos analizleri yapılmıştır. Yapılan analiz neticesinde sistemin kaotik davranış sergilediği gözlemlenmiştir. Matlab-tabanlı SEA kaotik sisteme ait zaman serileri, faz portreleri ve Lyapunov üstelleri analizi verilmiştir.

Bir sonraki aşamada SEA kaotik sistem ünitesi ayrıklaştırılmış DP nümerik algoritması kullanılarak FPGA çipleri üzerinde çalışacak şekilde 32-bit IEEE 754-1985 kayan noktalı sayı standardında gerçekleştirilmiştir. Xilinx ISE 14.7 benzetim programı kullanılarak donanım tanımlama dillerinden biri olan VHDL dilinde kodlanmıştır. Xilinx Virtex-6 FPGA yongası için sentezlenmiş ve test edilmiştir. Kayan noktalı sayı formatındaki DP-tabanlı SEA kaotik sistemin çalışma frekansı 316,756 MHz elde edilmiştir. Çip istatistikleri incelendiğinde sabit noktalı sayı formatından daha fazla çip kaynaklarını tükettiği ve daha hassas sonuçlar ürettiği gözlemlenmiştir. FPGA üzerinde gerçekleştirilmesinden sonra elde edilen sonuçlar Matlab-tabanlı olan kaotik sistemle karşılaştırılmış ve başarılı sonuçlar elde edildiği görülmektedir. Ayrıca elde edilen 32-bit IEEE 754-1985 kayan noktalı sayı formatına uygun kaotik ünitenin ürettiği ikilik tabandaki değerler bir dosyaya kaydedilmiştir. Kaydedilen bu sonuçlar onluk tabana dönüştürülmüş ve SEA kaotik ünitenin ürettiği ilk 3x909 veri seti kullanılarak kaotik sisteme ait zaman serileri ile faz portreleri çizdirilmiştir.

Son aşamada ise SEA kaotik osilatörü FPGA çipleri üzerinde gerçekleştirilmek üzere 16I-16Q, 14I-14Q, 12I-12Q, 10I-10Q, 8I-8Q beş farklı IQ-Math sabit noktalı sayı standartlarında modellenmiştir. Her bir kaotik osilatör ünitesi Xilinx ISE 14.7 benzetim programında ayrıklaştırılmış DP nümerik algoritması kullanılarak VHDL dilinde kodlanmıştır. Tasarımları gerçekleştirilen bu sistemler Xilinx Virtex-6 FPGA yongası için sentezlenmiş ve test edilmiştir. Kaotik osilatörlerin performans analizleri ve çip kullanım istatistikleri gözlemlenmiştir. 32-bit (16I-16Q) IQ-Math sabit noktalı sayı formatındaki SEA kaotik sistemi en düşük çalışma frekansı 344,585 MHz sahip olduğu görülmüştür. Bunun yanı sıra çip donanım kaynakları tüketimi karşılaştırıldığında diğer sabit noktalı sayı formatındaki SEA kaotik osilatörlerden daha fazla çip kaynağı tüketmesine rağmen kayan noktalı sayı formatındaki SEA kaotik osilatörden az çip kaynağı tükettiği görülmektedir. Diğer 14I-14Q, 12I-12Q, 10I-10Q IQ-Math sabit noktalı sayı formatındaki SEA kaotik sistemlerine ait çalışma frekansları sırasıyla 349,599 MHz, 354,762 MHz, 360,080 MHz olarak elde edilmiştir. Çip kaynak tüketimi kıyaslandığında ise bit sayısı azaldıkça kullandığı çip donanım tüketimi azalmaktadır. Ayrıca bit sayısı azaldıkça elde edilen çalışma frekansı arttığı gözlemlenmiş ve kaotik ünitenin ürettiği sonuç değer hassasiyeti ise azaldığı görülmektedir. 16-bit (8I-8Q) IQ-Math sabit noktalı sayı formatındaki SEA kaotik sistemi en az çip kaynak tüketimine ve en yüksek çalışma frekansına 365,559 MHz sahip olmasına rağmen yeterince hassas çözüm üretmediği ve kaotik davranış sergilemediği görülmektedir. Beş farklı sabit noktalı sayı formatındaki kaotik osilatörlerin FPGA üzerinde gerçekleştirilmesinden elde edilen sonuçlar bir dosya içerisine kaydedilmiştir. Kaydedilen IQ-Math sabit noktalı sayı değerleri gerçek sayı sistemine dönüştürülmüştür. Elde edilen sonuçlardan SEA kaotik sistemin ürettiği 3×909 değer kullanılarak her bir kaotik sisteme ait zaman serileri ve faz portreleri çizdirilmiştir.

Matlab-tabanlı SEA kaotik sistem referans alınarak sabit noktalı sayı formatındaki kaotik sistemlerin RMSE ve MSE hata analizleri yapılmıştır. 16-bit (8I-8Q) IQ-Math sabit sayı formatındaki kaotik sistemin hata oranı yüksek olduğu görülmektedir. Ayrıca 8I-8Q kaotik sistemin ürettiği kaotik sonuç değerleri kullanılarak çizdirilen faz portresi incelendiğinde sistemin kaotik davranış sergilemediği görülmektedir. Bu durumda 8I-8Q SEA kaotik sistemi mühendislik uygulamalarında kullanılmaya uygun olmadığı

anlaşılmaktadır.

Bu tez çalışmasında FPGA-tabanlı tasarımı gerçekleştirilen 32-bit IEEE 754-1985 kayan noktalı sayı formatındaki SEA kaotik osilatörü ve 32-bit (16I-16Q), 28-bit (14I-14Q), 24-bit (12I-12Q), 20-bit (10I-10Q) IQ-Math sabit sayı formatındaki kaotik sistemler kullanılarak kaos tabanlı mühendislik uygulamaları gerçekleştirilebilir. Sunulan tez çalışmasında SEA kaotik osilatörleri kullanılarak gelecekte kriptolojik haberleşme için kaos tabanlı rasgele sayı üreticileri, güvenli haberleşme için kaotik sinyal gizleme uygulaması, görüntü maskeleyme ve senkronizasyon uygulamaları gerçekleştirilebilir.

Çizelge 5.1’de literatürdeki farklı sayısal platformlar kullanılarak gerçekleştirilen kaotik osilatör tasarımları ve özellikleri sunulmuştur. Literatürde yapılan çalışmalar incelendiğinde bu çalışmada sunulan FPGA çipi üzerindeki DP-tabanlı SEA kaotik osilatör tasarımlarının verilen çalışmalara göre daha yüksek çalışma frekansına sahip olduğu görülmektedir. Çizelge 5.1’de verilen Heun-tabanlı altın oran ve Lü-Chen kaotik sistemlerinin çalışma frekansları yüksek olmasına rağmen literatür çalışması bölümünde belirtildiği üzere heun algoritması oldukça basit bir yapıya sahip olduğundan DP algoritması kadar hassas çözümler üretememektedir.

Çizelge 5.1 Farklı sayısal platformlar kullanılarak gerçekleştirilen kaotik osilatör tasarımları ve özellikleri.

Literatürde yapılan çalışmalar	Kullanılan kaotik osilatör	Kullanılan yapı	Kullanılan sayı standardı	Kullanılan platform	Çalışma Frekansı (MHz)
E. Tlelo-Cuautle <i>et al.</i> (2015)	Multi-scroll	Euler ve RK4	--	XC3S1000-5FT256 FPGA Spartan-3	70.943
Azzaz <i>et al.</i> (2013)	3B Hybrid	Euler	32 bits (16I-16Q) Fixed-point	Xilinx Virtex-II XC2VP30FFG896	38.86
Tuna vd. (2016)	Altın oran	Heun	32-bit (16I-16Q) Fixed point	Virtex-6	406.736
Merah <i>et al.</i> (2013)	Lorenz	RK4	32 bits (12I-20Q) Fixed-point	Xilinx Spartan-3	18
Koyuncu (2018)	Van der Pol	Yapay sinir ağları	32 Bit IEEE-754 Floating-point	Xilinx Virtex-6 XC6VCX75T	362
Rajagopal <i>et al.</i> (2013)	Kaotik chameleon	RK5-Butcher	32-bit IEEE-754-1985 Floating point	Xilinx Virtex-6 XC6VLX240T-1FF1156	325.759

Çizelge 5.1 (Devam) Farklı sayısal platformlar kullanılarak gerçekleştirilen kaotik osilatör tasarımları ve özellikleri.

Literatürde yapılan çalışmalar	Kullanılan kaotik osilatör	Kullanılan yapı	Kullanılan sayı standardı	Kullanılan platform	Çalışma Frekansı (MHz)
Tolba <i>et al.</i> (2017)	Liu	Grünwald-Letnikov	--	Xilinx Virtex-5 XC5VLX50T	137.561
Alçın vd. (2019)	Pehlivan-uyaroğlu	Yapay sinir ağları	32-bit IEEE-754-1985 Floating point	Xilinx Virtex-6 XC6VCX240T	231.616
Tuna vd. (2019)	Lü-Chen	Heun	32-bit IQ-Math Fixed point	Xilinx Virtex-5 XC6VLX75T-3FF484	464.688
Xu <i>et al.</i> (2018)	SEA	RK4	-	Raspberry Pi 3	-
Murillo-Escobar (2018)	Lojistik Map	Murillo-Escobar algoritması	-	Freescale COLDFIRE 32-bit Mikroişlemci	80
Çiçek vd. (2013)	1D Lojistik Map	-	-	FPAA	16
Zhang <i>et al.</i> (2008)	Kaotik Sistem	-	-	ASIC	100
Kharel (2010)	3D Henon ve Lorenz	-	-	TMS320C6713 DSK DSP	225
Sunulan Çalışma (2019)	SEA	Dormand-Prince	32-bit IEEE-754-1985 Floating point	Xilinx Virtex-6 XC6VLX240T FF1156-1	316.756
Sunulan Çalışma (2019)	SEA	Dormand-Prince	32-bit (16I-16Q) Fixed point	Xilinx Virtex-6 XC6VLX240T FF1156-1	344.585

6. KAYNAKLAR

- Akgül, A. (2015). Yeni kaotik sistemler ile rasgele sayı üretici tasarımı ve çoklu-ortam verilerinin yüksek güvenli şifrelenmesi. Doktora Tezi, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Sakarya.
- Akgul, A., Calgan, H., Koyuncu, I., Pehlivan, I. and Istanbulu, A. (2016). Chaos-based engineering applications with a 3D chaotic system without equilibrium points. *Nonlinear dynamics*, **84**: 481-495.
- Akgul, A., Hussain, S. and Pehlivan, I. (2016). A new three-dimensional chaotic system, its dynamical analysis and electronic circuit applications. *Optik*, **127**: 7062-7071.
- Akgul, A., Kacar, S., Pehlivan, I. and Aricioglu, B. (2018). Chaos based encryption of multimedia data and design of security analysis interface as an educational tool. *Computer Applications in Engineering Education*, **26**: 1336-1349.
- Akkaya, S., Pehlivan, I., Akgül, A. and Varan, M. (2018). The design and application of bank authenticator device with a novel chaos based random number generator. *Journal of the Faculty of Engineering and Architecture of Gazi University*, **33**: 1171-1182.
- Akpolat, A.N. (2015). FPGA tabanlı nesne algılama. Yüksek Lisans Tezi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Elazığ.
- Alçın, M. (2017). The Effect on Modelling Performance of Different Activation Functions for Feed Forward and Feedback Network Structures in Modeling of Chen Chaotic System. *International Journal of Scientific and Technological Research*, **7**: 60-70.
- Alçın, M., Tuna, M. and Koyuncu, I. (2018). IQ-Math Based Designing of Fourth Order Runge-Kutta Algorithm on FPGA and Performance Analysis According to ANN Approximation. *International Journal of Advanced Research in Science, Engineering and Technology*, **5**: 6523-6530.
- Alçın, M., Koyuncu, I., Tuna, M., Varan, M. and Pehlivan, I. (2019). A novel high speed Artificial Neural Network-based chaotic True Random Number Generator on Field Programmable Gate Array. *International Journal of Circuit Theory and*

Applications, **47**: 365-378.

- Avaroğlu, E., Koyuncu, I., Özer, A. B. and Türk, M. (2015). Hybrid pseudo-random number generator for cryptographic systems. *Nonlinear Dynamics*, **82**: 239-248.
- Az, I., Sahin, S. and Cavuslu, M.A. (2007). Implementation of fast fourier and inverse fast fourier transforms in FPGA. In 2007 IEEE 15th Signal Processing and Communications Applications, Eskişehir, 11-13 June, 1-4.
- Azar, A.T., Vaidyanathan, S. and Ouannas, A. (2017). Fractional order control and synchronization of chaotic systems. 1. Edition Springer.
- Azzaz M.S., Taugast C., Sadoudi S., Fellah R. and Dandache A. (2013). A new auto-switched chaotic system and its FPGA implementation. *Communications in Nonlinear Science and Numerical Simulation*, **18**:1792-1804.
- Bakiri, M., Guyeux, C., Couchot, J.F. and Oudjida, A.K. (2018). Survey on hardware implementation of random number generators on FPGA. *Theory and experimental analyses. Computer Science Review*, **27**: 135-153.
- Chen, G., Mao, Y. and Chui, C.K. (2004). "A symmetric image encryption scheme based on 3D chaotic cat maps", *Chaos, Solitons & Fractals*, **21**: 749-761.
- Chiu, R., Gonzalez, M.M. and Mancilla, D.L. (2013). Implementation of a Chaotic Oscillator into a Simple Microcontroller, *IERI Procedia*, **4**: 247–252.
- Çavuşlu, M.A., Dikmese, S., Şahin, S., Küçük, K. ve Kavak, A. (2006). Akıllı Anten Algoritmalarının IEEE 754 Kayan Sayı Formatı ile FPGA Tabanlı Gerçeklenmesi ve Performans Analizi. *Union Radio-Scientifique Internationale*, **3**: 610-612.
- Çavuşlu, M.A., Karakuzu, C., Şahin, S. and Yakut, M. (2011). Neural network training based on FPGA with floating point number format and it's performance. *Neural Computing and Applications*, **20**: 195-202.
- Çetin, O. (2014). Yapay sinir ağlarının uyarlanabilir donanımsal yapılarda gerçekleştirilmesi. Doktora Tezi, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Sakarya.
- Çiçek, S., Ferikoğlu, A. and Pehlivan, I. (2016). A new 3D chaotic system: dynamical analysis, electronic circuit design, active control synchronization and chaotic masking communication application. *Optik-International Journal for Light and*

Electron Optics, **127**: 4024-4030.

- Dasilva, R.C., Kurokawa, S., Jose, A. and Pissolato, J. (2011). Integration methods used in numerical simulations of transient electromagnetic. *IEEE Latin America Transactions*, **9**: 1060–1065,
- De Micco, L. and Larrondo, H.A. (2011). FPGA implementation of a chaotic oscillator using RK4 method. In 2011 VII Southern Conference on Programmable Logic, Cordoba, Argentina, 13-15 April, 185-190.
- Diao, L., Tang, J., Loh, P.C., Yin, S., Wang, L. and Liu, Z. (2018). An efficient DSP–FPGA-based implementation of hybrid PWM for electric rail traction induction motor control. *IEEE Transactions on Power Electronics*, **33**: 3276-3288.
- Dong, E., Yuan, M., Du, S. and Chen, Z. (2019). A new class of Hamiltonian conservative chaotic systems with multistability and design of pseudo-random number generator. *Applied Mathematical Modelling*, **73**: 40-71.
- Drozd, O.V. and Kapulin, D.V. (2018). The device of secure data transmission based on Magma crypto algorithm with implementation on FPGA. In 2018 Moscow Workshop on Electronic and Networking Technologies, Moscow, 14-16 March, 1-5.
- Dursun, M. and Kaşifoğlu, E. (2018). Design and implementation of the FPGA-based chaotic van der pol oscillator. *International Advanced Researches and Engineering Journal*, **2**: 309-314.
- Erick, L.O. (2007). Fixed-point representation & fractional math. Oberstar Consulting, revision, Madison, United States.
- Ersin, Ö.Ö. (2011). Türkiye’de mali sürdürülebilirliğin doğrusal olmayan bir analizi: MLSTAR çoklu lojistik yumuşak geçişli otoregresif modeli. *Ege Akademik Bakış Dergisi*, **11**: 41-58.
- Gandomi, A.H., Yang, X.S., Talatahari, S. and Alavi, A.H. (2013). Firefly algorithm with chaos. *Communications in Nonlinear Science and Numerical Simulation*, **18**: 89-98.
- Gürsoy H. (2016). FPGA tabanlı otomatik kontrol sistemleri geliştirme. Yüksek Lisans Tezi, Hacettepe Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.

- Hidalgo, R.M., Fernandez, J.G., Rivera, R.R. and Larrondo, H.A. (2001). Versatile DSP-based chaotic communication system. *Electronics Letters*, **37**: 1.
- Ismail, S.M., Said, L.A., Rezk, A.A., Radwan, A.G., Madian, A.H., Abu-Elyazeed, M.F. and Soliman, A.M. (2017). Generalized fractional logistic map encryption system based on FPGA. *AEU-International Journal of Electronics and Communications*, **80**: 114-126.
- Kaddoum, G. (2016). Wireless chaos-based communication systems: A comprehensive survey. *IEEE Access*, **4**: 2621-2648.
- Kahan, W. (1996). IEEE standard 754 for binary floating-point arithmetic. *Lecture Notes on the Status of IEEE*, **754**: 94720-1776.
- Karakaya, B., Çelik, V. and Gülten, A. (2017). Implementation Of Two Cell Non-Autonomous CNN Model On FPGA. *Firat University Turkish Journal of Science & Technology*, **13**: 25-29.
- Karataş, S. ve Sarıtaş, E. (2013). Her yönüyle FPGA ve VHDL, 3. Basım, Palme Yayıncılık, Ankara.
- Kılıç, R. (2010). Universal programmable chaos generator: Design and implementation issues. *International Journal of Bifurcation and Chaos*, **20**: 419-435.
- Kharel, R., Busawon, K., Aggoune, W. and Ghassemlooy, Z. (2010). Implementation of a secure digital chaotic communication scheme on a DSP board. In 2010 7th International Symposium on Communication Systems, Networks & Digital Signal Processing, Newcastle, United Kingdom, 21-23 July, 212-216.
- Koyuncu, İ. (2014). Kriptolojik Uygulamalar için FPGA tabanlı Yeni Kaotik Osilatörlerin ve Gerçek Rasgele Sayı Üreteçlerinin Tasarımı ve Gerçeklenmesi. Doktora Tezi, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Sakarya.
- Koyuncu, I. and Özcerit, A.T. (2017). The design and realization of a new high speed FPGA-based chaotic true random number generator. *Computers & Electrical Engineering*, **58**: 203-214.
- Koyuncu, İ. ve Özcerit, AT (2017). Yeni bir yüksek hızlı FPGA tabanlı kaotik gerçek rasgele sayı üreticinin tasarımı ve gerçekleştirilmesi. *Bilgisayar ve Elektrik Mühendisliği*, **58**: 203-214.
- Koyuncu, I. (2018). Implementation of high speed tangent sigmoid transfer function

- approximations for artificial neural network applications on FPGA. *Advances in Electrical and Computer Engineering*, **18**: 79-87.
- Koyuncu, İ. ve Şahin, İ. (2011). FPGA Çipleri İçin Cordic Tabanlı Ex Hesaplama Ünitesi Tasarımı, *E-journal of New World Sciences Academy Engineering Sciences*, **4**: 1565-1572.
- Koyuncu, İ., Çetin, Ö., Katırcıoğlu, F. and Tuna, M. (2015). Edge dedection application with FPGA based Sobel operator. In 2015 23rd Signal Processing and Communications Applications Conference, Malatya, 16-19 May, 1829-1832.
- Koyuncu, I., Ozcerit, A. T. and Pehlivan, I. (2013). An analog circuit design and FPGA-based implementation of the Burke-Shaw chaotic system. *Optoelectronics and Advanced Materials-Rapid Communications*, **7**: 635-638.
- Koyuncu, I., Ozcerit, A.T. and Pehlivan, I. (2014). Implementation of FPGA-based real time novel chaotic oscillator. *Nonlinear Dynamics*, **77**: 49-59.
- Koyuncu, İ., Özcerit, A.T. ve Pehlivan, İ. (2013). FPGA üzerinde yeni bir kaotik üreticinin gerçek zamanlı gerçekleştirilmesi. *Akademik Platform*, **1**: 875-879.
- Koyuncu, İ., Özcerit, A.T., Pehlivan, İ. and Avaroğlu, E. (2014). Design and implementation of chaos based true random number generator on FPGA. In 2014 22nd Signal Processing and Communications Applications Conference, Trabzon, 23-25 April, 236-239.
- Koyuncu, İ., Şeker, H.İ., Tuna, M. ve Alçın, M. (2018). Dormand-Prince tabanlı kaotik osilatör tasarımının FPGA üzerinde gerçekleşmesi. International Eurasian Conference on Science, Engineering and Technology, Ankara, 22-23 Kasım, 1059-1065.
- Kösten, M.M. ve Çavuşlu, M.A. (2015). VHDL ile Sayısal Tasarım ve FPGA Uygulamaları, 1. Basım, KODLAB Yayıncılık, İstanbul.
- Lamamra, K., Vaidyanathan, S., Azar, A.T. and Salah, C.B. (2017). Chaotic system modelling using a neural network with optimized structure. *Fractional Order Control and Synchronization of Chaotic Systems Springer, E-Book*, 833-856.
- Libano, F., Rech, P., Tambara, L., Tonfat, J. and Kastensmidt, F. (2018). On the reliability of linear regression and pattern recognition feed forward artificial neural networks in FPGAs. *IEEE Transactions on Nuclear Science*, **65**: 288-295.

- Manfredi, P., Canavero, F.G. (2014). Numerical calculation of polynomial chaos coefficients for stochastic per-unit-length parameters of circular conductors. *IEEE Transactions on Magnetics*, **50**: 74–82.
- Merah L., Pascha A., Said A. and Mamat N.H. (2013). Design and FPGA implementation of Lorenz chaotic system for information security issues. *Application Mathematics Sciences*, **7**: 237-246.
- Mert, İ., Karakuş, C. ve Peker, F. (2014). Antakya bölgesi rüzgar karakteristiğinin incelenmesi. *Dicle Üniversitesi Mühendislik Dergisi*, **5**: 13-22.
- Murillo-Escobar, M.A., Cruz-Hernández, C., Abundiz-Pérez, F. and López-Gutiérrez, R.M. (2016). Implementation of an improved chaotic encryption algorithm for real-time embedded systems by using a 32-bit microcontroller. *Microprocessors and Microsystems*, **45**: 297-309.
- Özkan, İ.A., Sarıtaş, İ. ve Herdem, S. (2011). Manyetik filtreler için FPGA tabanlı bulanık kontrolör tasarımı. *Selçuk Teknik Dergisi*, **10**: 271-284.
- Pandey, A., Saini, B.S., Singh, B. and Sood, N. (2018). Complexity sorting and coupled chaotic map based on 2D ECG data compression-then-encryption and its OFDM transmission with impair sample correction. *Multimedia Tools and Applications*, **78**: 1-39.
- Pareek, N.K., Patidar, V. and Sud, K.K. (2006). Image encryption using chaotic logistic map. *Image and vision computing*, **24**: 9926-934.
- Pareek, N.K., Patidar, V. and Sud, K.K. (2005). Cryptography using multiple one-dimensional chaotic maps. *Communications in Nonlinear Science and Numerical Simulation*, **10**: 715-723.
- Pareschi, F., Setti, G. and Rovatti, R. (2010). 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**: 3124-3137.
- Pehlivan, İ. (2007). Yeni kaotik sistemler Elektronik devre gerçeklemeleri, senkronizasyon ve güvenli haberleşme uygulamaları. Doktora Tezi, Sakarya üniversitesi, Fen Bilimleri Enstitüsü, Sakarya.
- Pehlivan, İ., Uyaroğlu, Y., Yalçın, M.A. ve Ferikoğlu, A. (2007). Sprott_94_A Kaotik

Sisteminin Senkronizasyonu ve Bilgi Gizlemede Kullanılması. Uluslararası Katılımlı Bilgi güvenliği ve Kriptoloji Konferansı, Ankara, 13-14 Aralık.

- Rabah, K., Ladaci, S. and Lashab, M. (2018). Bifurcation-based fractional-order PI λ D μ controller design approach for nonlinear chaotic systems. *Frontiers of Information Technology & Electronic Engineering*, **19**: 180-191.
- Rajagopal, K., Karthikeyan, A. and Srinivasan, A. (2018). Dynamical analysis and FPGA implementation of a chaotic oscillator with fractional-order memristor components. *Nonlinear Dynamics*, **91**: 1491-1512.
- Rajagopal, K., Akgul, A., Jafari, S., Karthikeyan, A. and Koyuncu, I. (2017). Chaotic chameleon: Dynamic analyses, circuit implementation, FPGA design and fractional-order form with basic analyses. *Chaos, Solitons & Fractals*, **103**: 476-487.
- Rajagopalan, S., Rethinam, S., Deepika, A.N., Priyadarshini, A., Jyothirmai, M. and Rengarajan, A. (2017). Design of boolean chaotic oscillator using CMOS technology for true random number generation. In 2017 International conference on Microelectronic Devices, Circuits and Systems, India, 10-12 August, 1-6.
- Rashtchi, V. and Nourazar, M. (2015). FPGA implementation of a real-time weak signal detector using a duffing oscillator. *Circuits, Systems, and Signal Processing*, **34**: 3101-3119.
- Ren, H.P., Bai, C., Huang, Z.Z. and Grebogi, C. (2017). Secure communication based on hyperchaotic Chen system with time-delay. *International Journal of Bifurcation and Chaos*, **27**: 1750076.
- Ren, H.P., Bai, C., Kong, Q., Baptista, M.S. and Grebogi, C. (2017). A chaotic spread spectrum system for underwater acoustic communication. *Physica A: Statistical Mechanics and its Applications*, **478**: 77-92.
- Rodríguez-Orozco, E., García-Guerrero, E., Inzunza-Gonzalez, E., López-Bonilla, O., Flores-Vergara, A., Cárdenas-Valdez, J. and Tlelo-Cuautle, E. (2018). FPGA-based chaotic cryptosystem by using voice recognition as access key. *Electronics*, **7**: 414.
- Rozic, V., Yang, B., Dehaene, W. and Verbauwhede, I. (2015). Highly efficient entropy extraction for true random number generators on FPGAs. In 2015 52nd Design Automation Conference, San Francisco, 8-12 June, 1-6.

- Rüemelin, W. (1982). Numerical treatment of stochastic differential equations. *SIAM Journal on Numerical Analysis*, **19**: 604-613.
- Savran, İ. (2017). Donanım Tanımlama Dili VHDL ve FPGA Uygulamaları. 1. Basım, Papatya Yayıncılık Eğitim, İstanbul.
- Schafer, B. C., Takenaka, T. and Wakabayashi, K. (2009). Adaptive simulated annealer for high level synthesis design space exploration. In 2009 International Symposium on VLSI Design, Automation and Test, Taiwan, 28-30 April, 106-109.
- Silahtar, O. (2018). FPGA Tabanlı Kaos Senkronizasyonu İçin Denetleyicic Tasarımı. Yüksek Lisans Tezi, Van Yüzüncü Yıl Üniversitesi, Fen Bilimleri Enstitüsü, Van.
- Şahin, O. (2012). Kriptoloji Uygulamalarında Kullanılacak Bir İşlemcinin Tasarlanarak FPGA Üzerinde Gerçeklenmesi. Lisans Tezi, İstanbul Teknik Üniversitesi, Mühendislik Fakültesi, İstanbul.
- Şeker, H.I., Tuna, M. and Koyuncu, I. (2018). Design and Implementation of FPGA-Based Mexican Hat Wavelet for Real-Time Wavelet Transforming. 3rd International Conference on Engineering Technology and Applied Sciences, Skopje, Macedonia, 18-20 July, 168-173.
- Tlelo-Cuautle, E., Guillén-Fernández, O., de Jesus Rangel-Magdaleno, J., Melendez-Cano, A., Nuñez-Perez, J.C. and de la Fraga, L.G. (2019). FPGA Implementation of Chaotic Oscillators, Their Synchronization, and Application to Secure Communications. In Recent Advances in Chaotic Systems and Synchronization, Academic Press, 301-328.
- Tlelo-Cuautle, E., Rangel-Magdaleno, J.J., Pano-Azucena, A.D., Obeso-Rodelo, P.J. and Nuñez-Perez, J.C. (2015). FPGA realization of multi-scroll chaotic oscillators. *Communications in Nonlinear Science and Numerical Simulation*, **27**: 66-80.
- Tolba, M.F., AbdelAty, A.M., Soliman, N.S., Said, L.A., Madian, A.H., Azar, A.T. and Radwan, A.G. (2017). FPGA implementation of two fractional order chaotic systems. *AEU-International Journal of Electronics and Communications*, **78**: 162-172.
- Tuna, M. (2017). Kaos tabanlı çift entropi çekirdekli gerçek rasgele sayı üretici tasarımı

ve FPGA üzerinde gerçekleştirilmesi. Doktora Tezi, Karabük Üniversitesi, Fen Bilimleri Enstitüsü, Karabük.

- Tuna, M., Koyuncu, I. and Alçın, M. (2018). Fixed and Floating point-Based High-Speed Chaotic Oscillator Design with Different Numerical Algorithms on FPGA. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, **7**: 3179-3187.
- Tuna, M., Koyuncu, I., Fidan, C.B. and Pehlivan, I. (2015). Real time implementation of a novel chaotic generator on FPGA. In 2015 23rd Signal Processing and Communications Applications Conference, Malatya, 16-19 May, 698-701.
- Tuna, M., Fidan, C.B., Koyuncu, İ. and Pehlivan, İ. (2016). Real time hardware implementation of the 3D chaotic oscillator which having golden-section equilibria. 24th IEEE In Signal Processing and Communication Application Conference, Zonguldak, 16-19 May, 1309-1312.
- Tuna, M., Alçın, M., Koyuncu, İ., Fidan, C.B. and Pehlivan, İ. (2019). High speed FPGA-based Chaotic oscillator design. *Microprocessors and Microsystems*, **66**: 72-80.
- Vaidyanathan, S., Rajagopal, K., Volos, C.K., Kyprianidis, I.M. and Stouboulos, I.N. (2015). Analysis, adaptive control and synchronization of a seven-term novel 3-D chaotic system with three quadratic nonlinearities and its digital implementation in LabVIEW. *Journal of Engineering Science and Technology Review*, **8**: 130-141.
- Wei, J., Liao, X., Wong, K. W. and Zhou, T. (2007). Cryptanalysis of a cryptosystem using multiple one-dimensional chaotic maps. *Communications in Nonlinear Science and Numerical Simulation*, **12**: 814-822.
- Xiong, A., Zhao, X., Han, J. and Liu, G. (2014). Application of the chaos theory in the analysis of EMG on patients with facial paralysis. *Springer*, **274**: 805–819,
- Xu, G., Shekofteh, Y., Akgül, A., Li, C. and Panahi, S. (2018). A new chaotic system with a self-excited attractor: entropy measurement, signal encryption, and parameter estimation. *Entropy*, **20**: 86.
- Zhang, L. (2017). System generator model-based FPGA design optimization and hardware co-simulation for lorenz chaotic generator. 2nd Asia-Pacific Conference

on Intelligent Robot Systems, China, 16-18 June, 170-174.

Zhang, Y., Liu, Z. and Zheng, X. (2008). A chaos-based image encryption ASIC using reconfigurable logic. Asia Pacific Conference on Circuits and Systems, China, 30 November, 1782-1785.

Zhou, L., Wang, C. and Zhou, L. (2018). A novel no-equilibrium hyperchaotic multi-wing system via introducing memristor. *International Journal of Circuit Theory and Applications*, **46**: 84-98.

İnternet Kaynakları

1) <http://www.ni.com/innovations-library/white-papers/> 19.10.2018

ÖZGEÇMİŞ

Adı Soyadı : Halil İbrahim ŞEKER
Doğum Yeri ve Tarihi : KARAMAN/10.01.1994
Yabancı Dili : İngilizce
İletişim (Telefon/e-posta) : 0506 647 06 01/hseker5@gmail.com

Eğitim Durumu (Kurum ve Yıl)

Lise : Karaman TOKİ Anadolu Lisesi , (2008-20012)
Lisans : Bozok Üniversitesi, Elektrik Elektronik Mühendisliği Bölümü, (2013-2017)
Yüksek Lisans : Afyon Kocatepe Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik Elektronik Mühendisliği Anabilim Dalı, (2017-2019)

Yayımları (Uluslararası hakemli dergilerde yayımlanan makaleler):

[1] Koyuncu, İ., Şeker, H.İ., (2019). Implementation of Dormand-Prince based chaotic oscillator designs in different IQ-Math number standards on FPGA. *Sakarya University Journal of Science*, **23**: (859-868).

Uluslararası Bilimsel Toplantılarda Sunulan ve Bildiri Kitaplarında (Proceedings) Basılan Bildiriler:

[1] Şeker, H.I., Tuna, M. and Koyuncu, I. (2018). Design and Implementation of FPGA-Based Mexican Hat Wavelet for Real-Time Wavelet Transforming. 3rd International Conference on Engineering Technology and Applied Sciences, Skopje, Macedonia, 18-20 July, 168-173.

[2] Koyuncu, İ., Şeker, H.İ., Tuna, M. ve Alçın, M. (2018). Dormand-Prince tabanlı kaotik osilatör tasarımının FPGA üzerinde gerçekleştirilmesi. International Eurasian Conference on Science, Engineering and Technology, Ankara, 22-23 Kasım, 1059-

1065.

[3] Şeker, H. İ., Koyuncu, İ., Tuna, M., & Alçın, M., (2019). Dormand-Prince Tabanlı SEA Kaotik Osilatör Tasarımının FPGA Üzerinde Gerçeklenmesi. 5th International Academic Research Congress, Ankara, 17-21 Nisan, 1-8.

Projeler:

Farklı IQ-Math sayı standartları için FPGA-tabanlı kaotik osilatörlerin tasarımı ve gerçekleştirilmesi, Proje No: 18.FEN.BİL.50, Afyon Kocatepe Üniversitesi BAP Projesi, Araştırmacı, 2018.