

**T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**KRİPTOLOJİK UYGULAMALAR İÇİN FPGA TABANLI
YENİ KAOTİK OSİLATÖRLERİN VE GERÇEK RASGELE
SAYI ÜRETEÇLERİNİN TASARIMI VE GERÇEKLENMESİ**

DOKTORA TEZİ

İsmail KOYUNCU

**Enstitü Anabilim Dalı : ELEKTRİK-ELEKTRONİK
MÜHENDİSLİĞİ**
Enstitü Bilim Dalı : ELEKTRONİK
Tez Danışmanı : Doç. Dr. Ahmet Turan ÖZCERİT

Eylül 2014

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ


KRİPTOLOJİK UYGULAMALAR İÇİN FPGA
TABANLI YENİ KAOTİK OSİLATÖRLERİN VE
GERÇEK RASGELE SAYI ÜRETEÇLERİNİN
TASARIMI VE GERÇEKLENMESİ


DOKTORA TEZİ

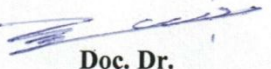
İsmail KOYUNCU

Enstitü Anabilim Dalı : ELEKTRİK-ELEKTRONİK
MÜHENDİSLİĞİ

Bu tez 18/09/2014 tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.



Prof. Dr.
Abdullah FERİKOĞLU
Jüri Başkanı


Doç. Dr.
Ahmet Turan ÖZCERİT
Üye


Doç. Dr.
Yılmaz UYAROĞLU
Üye

Doç. Dr.
Rüştü GÜNTÜRKÜN
Üye

Doç. Dr.
İbrahim ŞAHİN
Üye





ÖNSÖZ

Tez çalışması boyunca her türlü desteğini esirgemeyen ve tüm çalışmalarım boyunca gerek maddi ve gerekse manevi olarak destekleyen sayın danışmanım ve değerli hocam Doç. Dr. Ahmet Turan ÖZCERİT'e en içten teşekkürlerimi sunarım.

Tez amacının belirlenmesinde ve tez aşamalarında bilgi birikimi ve tecrübeleriyle bana yardımcı olan değerli hocam Doç. Dr. İhsan PEHLİVAN'a teşekkürü bir borç bilirim.

Doktora süreci boyunca bilgi ve birikimlerini benimle paylaşan ve tecrübeleri ile her türlü katkıda bulunan sayın Prof. Dr. Abdullah FERİKOĞLU ve Doç. Dr. Yılmaz UYAROĞLU hocalarıma teşekkürlerimi sunarım.

Çalışmalarım sırasında yapmış olduğu tatlılarla bana sürekli olarak destek olan sevgili eşim Mine KOYUNCU ve kızım Zeynep Betül KOYUNCU'ya teşekkür ederim.

Ayrıca tüm hayatım boyunca daima yanımda olduğunu hissettiren ve beni her türlü nimetlerle donatan Zat'a en içten duygularıyla hamd ederim.

İÇİNDEKİLER

ÖNSÖZ	ii
İÇİNDEKİLER.....	iii
SİMGELER VE KISALTMALAR LİSTESİ.....	vi
ŞEKİLLER LİSTESİ.....	x
TABLolar LİSTESİ.....	xx
ÖZET	xiv
SUMMARY	xv
BÖLÜM 1.	
GİRİŞ	1
1.1. Tezin Amacı	7
1.2. Tezde İzlenecek Yol	8
BÖLÜM 2.	
TEMEL KAVRAMLAR	10
2.1. Kaos ve Kaotik Sistemler	10
2.2. FPGA Çipleri	12
2.3. Nümerik Algoritmalar	14
2.4. Gerçek Rasgele Sayı Üreteçleri	17
2.5. İstatistiksel Rasgelelik Testleri	20
2.5.1. FIPS-140-1 testi.....	20
2.5.2. NIST-800-22 testi	22
BÖLÜM 3.	
REFERANS KAOTİK SİSTEMLERİN NÜMERİK VE ELEKTRONİK DEVRE MODELLERİ.....	51
3.1. Sundarapandian-Pehlivan Kaotik Sistemi	51

3.1.1. Sundarapandian-Pehlivan kaotik sistemi nümerik modeli	56
3.1.2. Sundarapandian-Pehlivan kaotik sistemi Orcad-PSpice modeli ..	57
3.2. Pehlivan-Wei Kaotik Sistemi.....	60
3.2.1. Pehlivan-Wei kaotik sistemi nümerik modeli	63
3.2.2. Pehlivan-Wei kaotik sistemi Orcad-PSpice modeli.....	63
BÖLÜM 4.	
FPGA TABANLI KAOTİK OSİLATÖRLERİN TASARIMI VE GERÇEKLENMESİ.....	68
4.1. Ayrıklaştırılmış Algoritmalar	68
4.2. FPGA Tabanlı Kaotik Osilatörlerin Gerçeklenmesi.....	72
4.3. FPGA Tabanlı Kaotik Osilatörlerin Test Sonuçları.....	78
BÖLÜM 5.	
FPGA TABANLI YENİ KAOTİK GERÇEK RASGELE SAYI ÜRETEÇLERİNİN TASARIMI VE GERÇEKLENMESİ.....	87
5.1. FPGA Tabanlı Gerçek Rasgele Sayı Üretici Tasarımları.....	87
5.1.1. Sabit eşik değer tabanlı GRSÜ.....	88
5.1.2. Adaptif eşik değer tabanlı GRSÜ	90
5.1.3. Kayan noktalı sayı tabanlı GRSÜ	91
5.2. Gerçek Rasgele Sayı Üreteçlerinin FPGA Üzerinde Gerçeklenmesi ..	92
BÖLÜM 6.	
FPGA TABANLI YENİ KAOTİK GERÇEK RASGELE SAYI ÜRETEÇLERİNİN İSTATİKSEL RASGELELİK TESTLERİ VE SONUÇLARI.....	98
6.1. FPGA Tabanlı Kaotik GRSÜ FIPS-140-1 Testleri ve Sonuçları	98
6.2. FPGA Tabanlı Kaotik GRSÜ NIST-800-22 Testleri ve Sonuçları.....	99
BÖLÜM 7.	
SONUÇLAR VE ÖNERİLER	104
KAYNAKLAR.....	107
EKLER.....	118

SİMGELER VE KISALTMALAR LİSTESİ

A	: Jakobiyen matrisi
AED	: Adaptif Eşik Değer
AFD	: Ayrık Fourier Dönüşümü
α	: Önem seviyesi
ASIC	: Application Specific Integrated Circuits
Ay	: Doğrusal terim
B	: Örtüşmeyen şablon eşleştirme testinde özel şablon
b	: Bit değerleri
β	: Sistem parametresi
BSKS	: Burke-Shaw Kaotik Sistemi
C	: Kapasite değeri
C_i^m	: Muhtemel m-bit değerlerin sayısı
CLB	: Configurable Logic Block (Konfigüre edilebilir mantıksal blok)
CMOS	: Complementary Metal Oxide Semiconductor
DFF	: Data Flip-Flop
DGKK	: Doğrusal Geri beslemeli Kayan Kaydedici
Δh	: Algoritma adım miktarı
$\Delta^2\Psi_m^2(\text{obs})$: m-bit örneğin beklenen frekansı
DSP	: Digital Signal Processor
ECAD	: Electronic Computer Aided Design
e_i	: Beklenen frekans
ε	: Bit dizisi
ε'	: Artırım dizisi
ε_i	: Bit dizisinin i. elemanı
erfc	: The Complementary Error Function
$\Phi(z)$: Olasılık yoğunluk fonksiyonu

exp	: Üs bitleri değeri
FIPS	: Federal Information Processing Standards
F-CCMs	: FPGA-based Custom Computing Machines
FPAA	: Field Programmable Analog Array
FPGA	: Field Programmable Gate Array
γ	: Sistem parametresi
GRSÜ	: Gerçek Rasgele Sayı Üreteçleri
G(y)y	: Doğrusal olmayan terim
H ₀	: Sıfır hipotezi
H _a	: Alternatif hipotez
i	: Yaklaşık entropi testi için blok değerleri
IEEE	: The Institute of Electrical and Electronical Engineers
IEEE-754	: IEEE Kayan noktalı sayı formatı
igamc	: Incomplete Complementary Gamma Function
IOB	: Giriş-Çıkış Bloğu
IP-core	: Intellectual Properties core
ISE	: Integrated Software Environment
J	: i. L-bit bloğun onluk sayı sistemindeki değeri
j	: Kesir bitlerinin sayısı
K	: Bağımsızlık katsayısı
k ₁	: RK algoritmasında ilk hesaplanan değişken
k ₂	: RK algoritmasında ikinci hesaplanan değişken
k ₃	: RK algoritmasında üçüncü hesaplanan değişken
k ₄	: RK algoritmasında dördüncü hesaplanan değişken
k ₅	: RK algoritmasında beşinci hesaplanan değişken
k ₆	: RK algoritmasında altıncı hesaplanan değişken
χ^2	: Ki-kare dağılımı
L	: Üniversal testinde her bir bloğun uzunluğu
λ	: Öz değerler
λ_σ	: Algoritma parametreleri
LFRS	: Linear Feedback Shift Register
L _i	: i. blokta üretilen en kısa DGKK dizisi
LUT	: Look Up Table

M	: Bit dizisinde belirli sayıdaki bitlerinden oluşan blok
m	: Örtüşen şablon eşleştirme testinde özel blokların bit sayısı
μ	: Beklenen değer
N	: İkili matris derece testinde matris sayısı
n	: Bit dizisinin uzunluğu
N_0	: T değerinden daha küçük beklenen değeri
NIST	: National Institute of Standards and Technology
ω_i	: Gözlemlenen frekans
π	: Bit dizisindeki 1 değerlerinin sayısı
$\phi^{(m)}$: Blokların ampirik dağılım frekansı
PLL	: Phase Locked Loop (Faz Kilitlemeli Döngü)
PWKS	: Pehlivan-Wei Kaotik Sistemi
P-değeri	: NIST-800-22 testinde rasgelelik ölçütü
κ_σ	: Algoritma parametreleri
KNS	: Kayan Noktalı Sayı
Q	: İkili matris derece testinde sütun sayısı
R	: Direnç değeri
RK4	: Dördüncü dereceden Runge-Kutta algoritması
RK5	: Beşinci dereceden Runge-Kutta algoritması
RMSE	: Root Mean Square Error (Ortalama Karese Hatanın Karekökü)
RO	: Ring Osilatörü
RSÜ	: Rasgele Sayı Üreteçleri
SED	: Sabit Eşik Değer
σ^2	: Varyans
sign	: İşaret biti
S_n	: Normalizasyon işleminden elde edilen değer
S_{obs}	: Gözlemlenen değer
SoC	: System On a Chip (Tek Çip Üzerinde Sistem)
SPKS	: Sundarapandian-Pehlivan Kaotik Sistemi
SRSÜ	: Sözde Rasgele Sayı Üreteçleri
σ_x	: x değişkeni için eşik değeri
σ_y	: y değişkeni için eşik değeri
σ_z	: z değişkeni için eşik değeri

T	: Tepe yüksekliği eşik değeri
t	: Zaman
TFF	: Toggle Flip-Flop
τ	: Test için gerekli parametre şartı
T_i	: Dağılımın rasgele değişkeni
T_j	: Muhtemel L-bit değerleri
v	: Onluk sayı değeri
var	: Varyans
VCO	: Voltage Controlled Oscillator (Gerilim Kontrollü Osilatör)
$V_{\text{exp}}(L)$: Farklı L değerleri için beklenen değer
VHDL	: VHSICircuit Hardware Description Language
VHSIC	: Very High Speed Integrated Circuit
V_i	: En uzun 1 dizisinin akış frekansı
$v_{i_1 \dots i_m}$: m bitlik $i_1 \dots i_m$ örneklerin frekansı
$v_{i_1 \dots i_{m-1}}$: m-1 bitlik $i_1 \dots i_{m-1}$ örneklerin frekansı
$v_{i_1 \dots i_{m-2}}$: m-2 bitlik $i_1 \dots i_{m-2}$ örneklerin frekansı
VLSI	: Very Large Scale Integrated Circuit
V(obs)	: Bit osilasyon sayısı
W_j	: Özel B şablonunun frekansı
x_e	: Denge noktası
\hat{x}_i	: Gerçek değerler vektörü
x_i	: Tahmin edilen değerler vektörü
XOR	: Exclusive Or (Özel Veya)
ξ	: Rasgele yürüyüşlerde ziyaret edilen durumların toplam sayısı
ξ_σ	: Algoritma parametreleri
y_λ	: Algoritma ilk değeri
$y_{\lambda+1}$: Algoritma sonraki değeri

ŞEKİLLER LİSTESİ

Şekil 2.1. FPGA çipi genel yapısı	13
Şekil 2.2. Rasgele sayı üreteçlerinin sınıflandırılması	18
Şekil 2.3. Kaos tabanlı RSÜ blok şeması	19
Şekil 3.1. $\alpha=1.5$, $\beta=0.4$ ve $\gamma=0.4$ değerleri için SPKS Lyapunov üstelleri.....	55
Şekil 3.2. α parametresi için SPKS Lyapunov üstelleri spektrumu.....	55
Şekil 3.3. β parametresine göre SPKS Lyapunov üstelleri spektrumu.....	56
Şekil 3.4. γ parametresine göre SPKS Lyapunov üstelleri spektrumu	56
Şekil 3.5. Sundarapandian-Pehlivan kaotik sistemi x , y ve z zaman serileri	57
Şekil 3.6. SPKS osilatörü nümerik faz portreleri	57
Şekil 3.7. SPKS elektronik devre şeması.....	59
Şekil 3.8. SPKS osilatörü PSpice simülasyonu x , y ve z sinyalleri zaman serileri	60
Şekil 3.9. SPKS osilatörü PSpice faz portreleri	60
Şekil 3.10. $\alpha=2$ değeri için PWKS Lyapunov üstelleri	62
Şekil 3.11. α parametresine göre PWKS Lyapunov üstelleri spektrumu	62
Şekil 3.12. Pehlivan-Wei kaotik sistemi x , y ve z zaman serileri.....	63
Şekil 3.13. PWKS osilatörü nümerik faz portreleri.....	63
Şekil 3.14. PWKS elektronik devre şeması	66
Şekil 3.15. PWKS osilatörü x , y ve z sinyalleri için PSpice zaman serileri.....	67
Şekil 3.16. PWKS osilatörü PSpice faz portreleri.....	67
Şekil 4.1. FPGA-Tabanlı YKO Ünitesi en üst seviye blok diyagramı	73
Şekil 4.2. FPGA-Tabanlı YKO Ünitesi ikinci seviye blok diyagramı	74
Şekil 4.3. Euler-Tabanlı YKO Ünitesi üçüncü seviye blok diyagramı	75
Şekil 4.4. Heun-Tabanlı YKO Ünitesi üçüncü seviye blok diyagramı	76
Şekil 4.5. RK4-Tabanlı YKO Ünitesi üçüncü seviye blok diyagramı.....	77
Şekil 4.6. RK5-Butcher-Tabanlı YKO Ünitesi üçüncü seviye blok diyagramı	78
Şekil 4.7. Euler-tabanlı SPKS osilatör ünitesi Xilinx ISE Simülatörü sonuçları.....	79
Şekil 4.8. Heun-tabanlı SPKS osilatör ünitesi Xilinx ISE Simülatörü sonuçları.....	79

Şekil 4.9. RK4-tabanlı SPKS osilatör ünitesi Xilinx ISE Simülatörü sonuçları.....	79
Şekil 4.10. RK5-Butcher-tabanlı SPKS ünitesi Xilinx ISE Simülatörü sonuçları.....	80
Şekil 4.11. Euler-tabanlı PWKS ünitesi Xilinx ISE Simülatörü sonuçları.....	80
Şekil 4.12. Heun-tabanlı PWKS ünitesi Xilinx ISE Simülatörü sonuçları.....	80
Şekil 4.13. RK4-tabanlı PWKS ünitesi Xilinx ISE Simülatörü sonuçları.....	80
Şekil 4.14. RK5-Butcher-tabanlı PWKS ünitesi Xilinx ISE Simülatörü sonuçları.....	81
Şekil 4.15. RK4-tabanlı SPKS osilatör ünitesi faz portreleri.....	81
Şekil 4.16. RK4-tabanlı PWKS osilatör ünitesi faz portreleri.....	82
Şekil 4.17. SPKS z sinyali için FPGA-tabanlı mutlak hata sonuçları.....	85
Şekil 4.18. SPKS için FPGA-tabanlı ortalama mutlak hata sonuçları.....	85
Şekil 4.19. PWKS x sinyali için FPGA-tabanlı mutlak hata sonuçları.....	85
Şekil 4.20. PWKS için FPGA-tabanlı ortalama mutlak hata sonuçları.....	86
Şekil 5.1. FPGA-tabanlı Sabit Eşik Değer GRSÜ blok şeması.....	89
Şekil 5.2. Düzeltici Fonksiyon ünitesi blok şeması.....	90
Şekil 5.3. Adaptif Sayı Üretici ünitesi blok şeması.....	91
Şekil 5.4. 32-bit IEEE 754-1985 kayan noktalı sayı standardı gösterimi.....	92
Şekil 5.5. SED-Tabanlı GRSÜ ünitesi Xilinx ISE Simülatörü sonuçları.....	93
Şekil 5.6. AED-Tabanlı GRSÜ ünitesi Xilinx ISE Simülatörü sonuçları.....	93
Şekil 5.7. KNS-Tabanlı GRSÜ ünitesi Xilinx ISE Simülatörü sonuçları.....	93

TABLolar LİSTESİ

Tablo 2.1. Koşu testi için blok uzunluklarına göre blok sayıları	21
Tablo 2.2. İstatistiksel hipotez testi sonuçları	24
Tablo 2.3. Dizi uzunluğuna göre önerilen blok uzunluğu	30
Tablo 2.4. Çeşitli blok değerleri için en uzun birlerin akış frekans değerleri.....	30
Tablo 2.5. Blok uzunluklarına göre belirlenecek K ve N parametre değerleri.....	30
Tablo 2.6. Bloklardaki en uzun birlerin frekansının incelenmesi	31
Tablo 2.7. $m=3$ için M_1 ve M_2 blokları içerisinde $B=001$ şablonunun incelenmesi...36	
Tablo 2.8. M_1 bloğu içerisinde $B=11$ özel şablonunun bulunma durumları	37
Tablo 2.9. Maurer testi L -bit uzunluğundaki blokların bölümleri.....	39
Tablo 2.10. Dört başlangıç değeri ile oluşturulan muhtemel L -bit değerleri.....	39
Tablo 2.11. Test bölümü için L -bit değerleri	40
Tablo 2.12. L değerleri için $Vexp(L)$ ve $var(f_n)$ değerleri	41
Tablo 2.13. Test için ileri ve geri yönlü metotların uygulanması	46
Tablo 2.14. Verilen ε dizisi için oluşan rasgele gezinti döngü frekansları	48
Tablo 4.1. Kaotik osilatörlerin FPGA çip istatistikleri.....	83
Tablo 5.1. SED-Tabanlı GRSÜ ünitelerinin FPGA çip istatistikleri.....	94
Tablo 5.2. AED-Tabanlı GRSÜ ünitelerinin FPGA çip istatistikleri.....	95
Tablo 5.3. KNS-Tabanlı GRSÜ ünitelerinin FPGA çip istatistikleri	96
Tablo 5.4. KNS-Tabanlı GRSÜ bit üretim hızları.....	97
Tablo 5.5. Literatürde sunulan GRSÜ çalışmaları	97
Tablo 6.1. SPKS ile gerçekleştirilen GRSÜ üniteleri FIPS-140-1 test sonuçları	98
Tablo 6.2. PWKS ile gerçekleştirilen GRSÜ üniteleri FIPS-140-1 test sonuçları.....	99
Tablo 6.3. SPKS Euler algoritması tabanlı GRSÜ ünitesi NIST testi sonuçları.....	100
Tablo 6.4. PWKS Euler algoritması tabanlı GRSÜ ünitesi NIST testi sonuçları	100
Tablo 6.5. SPKS Heun algoritması tabanlı GRSÜ ünitesi NIST testi sonuçları.....	101
Tablo 6.6. PWKS Heun algoritması tabanlı GRSÜ ünitesi NIST testi sonuçları	101
Tablo 6.7. SPKS RK4 algoritması tabanlı GRSÜ ünitesi NIST testi sonuçları.....	102

Tablo 6.8. PWKS RK4 algoritması tabanlı GRSÜ ünitesi NIST testi sonuçları	102
Tablo 6.9. SPKS RK5-Butcher tabanlı GRSÜ ünitesi NIST testi sonuçları.....	103
Tablo 6.10. PWKS RK5-Butcher tabanlı GRSÜ ünitesi NIST testi sonuçları	103

ÖZET

Anahtar kelimeler: Gerçek Rasgele Sayı Üreteci, Kaos, Kaotik Osilatör, FPGA, VHDL, Nümerik Algoritmalar

Bu tez çalışmasında, gerçek zamanlı, yüksek çalışma frekansı ve bit üretim hızına sahip Gerçek Rasgele Sayı Üreteçleri (GRSÜ), FPGA tabanlı kaotik osilatörler kullanılarak tasarlanmış ve gerçekleştirilmiştir.

Bu amaçla tezin ilk aşamasında, çeşitli sistem parametrelerinin karşılaştırılması ve değerlendirilmesi amacıyla iki farklı kaotik sistem dört farklı nümerik diferansiyel denklem çözüm metodu ile modellenerek sistemlerin dinamik davranışları incelenmiş ve kaos analizleri yapılmıştır.

İkinci aşamada, seçilen kaotik sistemler bir ECAD programında şematik giriş yapılarak analog devre elemanları ile modellenmiştir. Nümerik benzetim sonuçları ile ECAD benzetim sonuçları karşılaştırılmıştır. Elde edilen sonuçlara göre analog elemanlar kullanılarak yapılan ECAD benzetimi ile Matlab destekli nümerik model sonuçları birbiri ile uyumlu değerler üretmiştir.

Sonraki aşamada, kaotik sistemler dört farklı diferansiyel denklem çözüm metodlarından yararlanılarak, 32-bit IEEE 754-1985 kayan noktalı sayı standardında VHDL programlama dili ile FPGA üzerinde modellenmiştir. Tasarımlar Virtex-6 ailesi XC6VLX550T-2FF1759 çipi için Xilinx ISE Design Tools 14.2 benzetim programı kullanılarak sentezlenmiştir. Elde edilen sonuçlara göre FPGA-tabanlı kaotik osilatörlerin maksimum çalışma frekansları yaklaşık olarak 390-464 MHz arasında değişmektedir. Buna göre kaotik osilatör ünitesi 1 milyon veriyi 46 ms gibi çok kısa bir sürede hesaplayabilmektedir. Bu aşamada, FPGA tabanlı ünitelerin ürettiği sonuçların doğruluğunu test etmek amacıyla RMSE yöntemi kullanılarak hassasiyet analizleri de yapılmıştır.

Dördüncü aşamada, FPGA-tabanlı örnek kaotik sistemler kullanılarak GRSÜ tasarımı gerçekleştirilmiştir. Genel olarak iki farklı kaotik sistem, kaotik osilatör tasarımında dört ayrı algoritma ve kuantalama için üç değişik yöntem sunularak toplamda 24 farklı gerçek rasgele sayı üreteci ünitesi tasarlanmıştır. Tasarımlardan elde edilen sonuçlara göre, ünitelerin maksimum çalışma frekansları 339-401 MHz ve bit üretim hızları 53-132 Mbit/s arasında değişmektedir.

Son aşamada, FPGA tabanlı GRSÜ'den elde edilen sayı dizileri FIPS-140-1 ve NIST-800-22 istatistiksel rasgelelik testleri kullanılarak test edilmiş ve tüm testlerden başarılı olmuştur.

DESIGN AND IMPLEMENTATION OF FPGA BASED NEW CHAOTIC OSCILLATORS AND TRUE RANDOM NUMBER GENERATORS FOR CRYPTOGRAPHIC APPLICATIONS

SUMMARY

Key Words: True Random Number Generator, Chaos, Chaotic Oscillator, FPGA, VHDL, Numerical Algorithms

In this thesis, real-time True Random Number Generators (TRNGs) with high operating frequency and bit generation rate have been designed and implemented using FPGA-based chaotic oscillators.

In the first stage, two chaotic systems have been determined and their dynamical behavioral and chaotic analyses have been investigated to compare various system parameters using four diverse numerical differential equation solution methods.

In the second stage, the chaotic systems have been modelled using analog components in an ECAD program. After that, numerical and ECAD simulation results have been compared and the results obtained from each simulation proves that both approaches have produced compatible outcomes.

In the next stage, the chaotic systems have been modelled in VHDL in 32-bit IEEE 754-1985 floating point number standard using four diverse numerical differential equation solution methods. The designs have been synthesized for Virtex-6 using Xilinx ISE Design Tools 14.2. According to the syntheses results, the maximum operating frequency of the FPGA-based chaotic oscillators varies between 390 MHz and 464 MHz. Accordingly, the chaotic oscillator unit has been able to calculate 1 million data sets in 46 ms. In this stage, in order to test accuracy of results produced by FPGA-based units, the sensitivity analysis have been also performed by employing RMSE method.

In the fourth stage, TRNG designs have been implemented using FPGA-based chaotic systems. 24 different TRNG units have been designed and implemented by employing two distinct chaotic systems, four different algorithms in the design of the chaotic oscillators and three diverse quantification methods. According to the results, operating frequency of the units varies between 339 MHz and 401 MHz and the bit-rates varies between 53 Mbit/s and 132 Mbit/s.

In the final stage, the number sequences derived from FPGA-based TRNG have been tested with FIPS-140-1 and NIST-800-22 randomness tests, and all sequences have been verified.

BÖLÜM 1. GİRİŞ

Evrende tüm sistemler doğrusal olmayan bir yapıya sahiptir ve doğrusallık sadece belirli sınırlar arasında geçerli olabilmektedir. Doğrusal olmayan sistemlerin, birbirleriyle olan ilişkilerini ortaya koyan ve bu sistemleri modellemeye çalışan bilim doğrusal olmayan (nonlinear) bilim olarak adlandırılmaktadır. Doğrusal olmayan sistemlerde çok önemsenmeyen bir davranış veya etki sistemde öngörülemeyecek kadar büyük değişimlere ve tepkilere neden olabilmektedir. Günümüzde üzerinde pek çok araştırma ve çalışmalar yapılan doğrusal olmayan bilim alanlarından birisi de kaos bilimi veya kaotik sistemlerdir.

Kaotik sistemlerin araştırılması ve uygulanmasına yönelik bilimsel ve endüstriyel alanlarda önemli çalışmalar gerçekleştirilmektedir. Mühendisliğin pek çok alanında kaotik sistemlerin varlığının ortaya çıkarılması, bu konuda yapılan yoğun çalışmalar ve yaşanan gelişmeler kaotik sistemlerin birçok uygulama alanında kullanılabileceğini göstermiştir. Bu uygulama alanlarına biyomedikal [1-3], haberleşme [4-6], kuantum elektronığı [7-9], elektromanyetik [10, 11], görüntü işleme [12, 13], kriptoloji [14-16], optik elektronik [17-19], bulanık mantık [20-22], güç elektronığı [23-25], biyokimya [26], kontrol [27, 28], fizik [29, 30], optimizasyon [31, 32], mekatronik [33], yapay sinir ağları [34] gibi alanlar örnek olarak verilebilir.

Kaos bilimi, düzensizlik ve karmaşa gibi olumsuz durumlar çağrıştırmasına rağmen, bu sistemler belirli aralıklar içerisinde kendilerine has bir düzene sahiptirler. Kaotik sistemlerin başlıca önemli özellikleri arasında; periyodik olmayan davranış sergilemeleri [35], başlangıç koşullarına hassas bağlı olmaları [36], sistem durum uzayında periyodik olmayan davranışlar sergilemeleri [37] ve sistem parametrelerinin değişimlerine fazlasıyla duyarlı olmaları [38] sayılabilmektedir.

Günümüzde güvenli haberleşme ve kriptoloji alanlarında, kaotik sistemlerin analog ve sayısal tabanlı donanımlar ile gerçekleştirilerek kaotik üreteçler oluşturulması konusunda birçok çalışma yapılmıştır [39-43]. Özellikle kaos tabanlı mühendislik uygulamalarında kullanılması gereken en temel yapılardan birisi kaotik işareti üreten bir kaos sinyal üreticidir. Kaos üreteçleri, donanımsal olarak analog veya sayısal tabanlı olmak üzere iki farklı şekilde gerçekleştirilebilmektedir. Analog kaotik üreteç yapıları kullanan sistemlerin sıcaklık ve kullanım ömrü ile değerleri değişmektedir. Bu nedenle sayısal devre tabanlı kaotik üreteçler genel olarak analog yapılı kaotik üreteçlerden daha avantajlıdır [44, 45]. Bu problem için en iyi çözüm sayısal donanım kullanarak kaotik üreteçlerin gerçekleştirilmesidir. Sayısal devre tabanlı kaotik üreteçler literatürde Sayısal İşaret İşlemciler (Digital Signal Processors (DSPs)) [46, 47], Uygulamaya Özel Tümlşik Devreler (Application Specific Integrated Circuits (ASIC)) [48, 49] ve Alan Programlanabilir Kapı Dizileri (Field Programmable Gate Array (FPGA)) [50–52] gibi farklı entegre yapılarla gerçekleştirilebilmektedir. ASIC tabanlı kaotik üreteçlerden diğer sayısal tabanlı eşdeğerlerine göre daha yüksek performans elde edilmektedir. Ancak ASIC tabanlı uygulamalar esnek bir yapıya sahip olmamakla birlikte bu sistemlerin ilk tasarım ve test maliyeti oldukça yüksektir. Ayrıca ASIC ile yapılan tasarımların maliyetinin düşürülmesi için önemli miktarda üretim yapılmalıdır. ASIC tabanlı seri üretim aşamalarında yapılacak küçük bir hata oldukça yüksek maliyet ve uzun zaman kaybına da neden olmaktadır. DSP çipleri ise kompleks matematiksel işlemleri gerçekleştirilebilmek için optimize edilmiş yapılardır. Bu çipler, işlemleri sıralı (sequential) olarak gerçekleştirilmektedir. Sürekli zamanlı kaotik sistemler genelde karakteristik olarak en az üç diferansiyel denklemden oluşmaktadır. Bu diferansiyel denklemlerin ayrık zamanlı yöntemlerle çözümlerinin mikroişlemci veya DSP tabanlı sistemler tarafından sıralı bir şekilde gerçekleştirilmesi uzun zaman almakta ve çalışma frekansları düşük kalmaktadır. FPGA çipleri paralel işlem yapabilmekle birlikte oldukça esnek bir yapıya sahiptirler. Ayrıca FPGA çiplerinin tasarım ve test maliyetleri ASIC tabanlı uygulamalara göre oldukça düşüktür. Bir diğer önemli nokta, sayısal tabanlı olmaları, tekrar programlanabilme özellikleri sayesinde FPGA tabanlı kaotik sistemler ve bu sistemlerin kullanıldığı uygulamalar daha esnek olmaktadır. Kaotik devre modelleri tekrar programlanabilir veya yeniden yapılandırılabilir sistemler içerisinde gerçeklemeye uygundur. Bu sayede kaotik

sistemler parametre deęişimlerine göre farklı formda işaret üretebilmektedir. Bu gibi nedenlerden dolayı kaotik sistemlerin FPGA tabanlı modellenmesi ile ilgili oldukça yoğun bir şekilde çalışmalar yapılmaktadır. Bu çalışmalara aşağıdaki bilimsel yayınlar örnek olarak verilebilir.

Sadoudi ve arkadaşları güvenli kaotik haberleşme sistemleri için Chen kaotik sistemini FPGA’de modellemişlerdir. Bu çalışmada, 32-bit kesirli sayı standardı ile RK4 algoritması kullanılmış ve tasarım bir adet Virtex-II çip içeren XCV1000FG456-4 bordu ile test edilmiştir. Yapılan tasarımdan elde edilen sinyaller ile Matlab sonuçları karşılaştırılmıştır. Sistemin çalışma frekansı 22.850 MHz olarak belirtilmiştir [53].

Merah ve arkadaşları yaptıkları çalışmada bilgi güvenliği uygulamaları için Lorenz kaotik sistemini FPGA üzerinde modellemişlerdir. Lorenz kaotik sisteminin tasarım aşamasında donanım tanımlama dillerinden birisi olan VHDL (Very High Speed Integrated Circuit **H**ardware **D**escription **L**anguage (Çok Yüksek Hızlı Tümlüşik Devre Donanım Tanımlama Dili)) kullanılmıştır. Tasarlanan kaotik sistem Xilinx firmasının Spartan-3 ailesi FPGA çipinde gerçekleştirilmiştir. Tasarımın çalışma frekansı yaklaşık olarak 18 MHz olarak ifade edilmiştir [54].

De Micco ve arkadaşları yaptıkları çalışmada RK4 algoritmasını kullanarak Lorenz kaotik sistemini FPGA’de gerçekleştirmişlerdir. Kaotik sistem tasarımında IEEE-754 kayan noktalı sayı standardı kullanılmıştır. Test aşamasında kaotik sistem Altera EP3C120F7 bordu ile test edilmiş ve sistemin çalışma frekansı yaklaşık 1 MHz olarak belirtilmiştir [55].

Wang ve arkadaşlarının sunduğu çalışmada dört boyutlu hiper kaotik yeni bir sistem tanıtılmıştır. Sunulan yeni hiper kaotik sistemin nümerik analizleri yapılmış ve analog elektronik devresi gerçekleştirilmiştir. Ayrıca önerilen kaotik sistem Euler algoritmasından yararlanılarak Altera Cyclone II EP2C35F484C8 FPGA çipleri ile gerçekleştirilmiştir [56].

Qun ve arkadaşları yaptıkları çalışmada, FPGA tabanlı ağ şifrelemeli kart uygulamasını Lorenz kaotik sistemini kullanarak gerçekleştirmişlerdir. Kaotik sistem tasarımı, Matlab/Simulink ile DSP Builder Tool blokları kullanılarak yapılmış ve otomatik kod dönüştürücü ile HDL koduna çevrilmiştir [57].

Ürettiği sinyallerin rasgele görünümlü dinamik davranışlar sergilemesi gibi özellikleri sayesinde kaotik osilatörlerin kullanım alanlarından birisi de GRSÜ (Gerçek Rasgele Sayı Üreteçleri) uygulamaları olmuştur. GRSÜ, kriptolojik uygulamalarda kullanılan en temel yapılardan birisidir. Hatta kriptolojik uygulamalarda kullanılan GRSÜ'nin ürettiği sayıların rasgeleliği, kriptolojik uygulamaların güvenliğini doğrudan etkilemektedir. Bu açıdan GRSÜ, kriptolojik uygulamalar için kritik bir öneme sahiptir. Literatürde FPGA tabanlı GRSÜ yapıları için çeşitli çalışmalar gerçekleştirilmiştir. Bu çalışmalardan bir kısmında genel olarak klasik osilatör, ring osilatör ve flip-flop gibi yapılardan yararlanılmıştır. Bunlara örnek olarak aşağıdaki çalışmalar verilebilir.

Danger ve arkadaşları yaptıkları çalışmada klasik PLL (Phase Locked Loop) ve Ring osilatörlerinin sınırlı bant aralığına sahip olduklarını ve bu osilatörler ile oluşturulan GRSÜ'nin bit üretim hızlarının birkaç Mbit/s'yi geçmeyeceğini belirtmişlerdir. Bu probleme çözüm olarak daha yüksek bit üretim hızı sağlayan yeni bir yapı sunulmuş ve yapı FPGA ile gerçekleştirilmiştir. Sunulan yeni yapının doğrulanması NIST testleri ile yapılmıştır. Sonuç olarak yapının çalışma frekansı 20 MHz ve GRSÜ bit üretim hızı 20 Mbit/s olarak verilmiştir. Yapının rasgelelik testlerinden de geçtiği vurgulanmıştır [58].

Wieczorek ve arkadaşlarının yaptığı çalışmada FPGA üzerinde çift kararlı flip-flop kullanarak GRSÜ ünitesi tasarlamışlardır. Yapılan çalışmada Spartan3E FPGA çipi kullanmışlardır. Çalışma sonucunda elde edilen veriler istatistiksel testlere tabi tutulmuştur. GRSÜ tarafından üretilen rasgele diziler testlerden başarılı bir şekilde geçmiştir. Tasarlanan ünitenin çalışma frekansı 50 MHz ve bit üretim hızı ise 5 Mbit/s olarak belirtilmiştir [59].

Lozac'h ve arkadaşlarının yaptıkları çalışmada FPGA çiplerinde çalışmak üzere bir GRSÜ yapısı tasarlamışlardır. GRSÜ açık döngü gecikme zinciri üzerine çalışmaktadır. Sunulan yapı Xilinx tarafından üretilen Virtex-5 XC5VLX50T çip ailesinde gerçekleştirilmiştir. Ünite çalışma frekansı 22 MHz ve bit üretim hızı 20 Mbit/s olarak belirtilmiştir. Ünitenin ürettiği rasgele sayılar rasgelelik testlerine tabi tutulmuş ve başarılı sonuçlar elde edilmiştir [60].

Fischer ve arkadaşları PLL tabanlı osilatörü FPGA çipleriyle gerçekleştirmişlerdir. GRSÜ yapısı VHDL'de tanımlanmış ve Altera firmasının Quartus II programı kullanılarak gerçekleştirilmiştir. Kriptografik uygulamalar için tasarlanan GRSÜ'nin performans analizi NIST testleri ile yapılmıştır. Testlerden elde edilen sonuçlara göre tasarlanan sistemin bit üretim hızı 1 Mbit/s olarak elde edilmiştir [61].

Schellekens ve arkadaşları çoklu ring osilatörünü FPGA'de modellemişlerdir. FPGA çipi olarak Xilinx tarafından üretilen Virtex-2 çipi kullanılmıştır. Tasarımı yapılan sistemin örnekleme zamanı 40 MHz olarak verilmiştir. Tasarımın rasgelelik kontrolü standart NIST-800-22 testleri ile yapılmıştır. Test sonuçlarına göre GRSÜ'nin bit üretim hızı 2.5 Mbit/s olduğu ifade edilmiştir [62].

Dichtl ve arkadaşları FPGA'de Fibonacci ve Galois ring osilatörlerini modellemişlerdir. Tasarımın gerçekleştirilmesi Xilinx firmasının ürettiği FPGA çiplerinden birisi olan Spartan-3 çipini içeren Starter kit kullanılarak yapılmıştır. Sistemin ürettiği bit üretim hızı 12.5 Mbit/s olarak verilmiştir [63].

Bir diğer çalışma da István ve arkadaşları tarafından FPGA tabanlı klasik jitter osilatör yöntemi kullanılarak yapılmıştır. Önerilen yöntemin doğruluğu istatistiksel NIST-800-22 testleriyle sağlanmıştır. Sistemin çalışma frekansı 50 MHz olmakla beraber, kullanılan osilatör nedeniyle sistemin bit üretim hızı 1.92 Mbit/s'yi aşamamaktadır [64].

Literatürde GRSÜ yapıları için yapılan çalışmaların diğer bir kısmı ise kaotik osilatörlerin analog elemanlar kullanılarak veya CMOS (Complementary Metal

Oxide Semiconductor) teknolojisi ile gerçekleştirilmesi üzerine yoğunlaşmaktadır. Bunlara örnek olarak aşağıdaki çalışmalar verilebilir.

Çiçek ve arkadaşlarının sunduğu çalışmada CMOS teknolojisi ile ayırık zamanlı kaos tabanlı yeni bir tasarım metodu kullanılarak GRSÜ yapısı geliştirilmiştir. GRSÜ yapısında bulunan kaotik sistem için tek boyutlu harita kullanılmıştır. Yapılan tasarımdan elde edilen rasgele bitler NIST testlerine tabi tutulmuş ve 11 testten geçtiği belirtilmiştir [65].

Ergün ve Özoğuz'un yaptıkları çalışmada otonom olmayan kaotik sistem kullanılarak CMOS teknolojisi ile GRSÜ yapısı önerilmiştir. Önerilen yapının çalışma frekansı 1.24 MHz ve bit üretim hızı 10 Mbit/s olarak belirtilmiştir. Sunulan çalışmadan elde edilen rasgele dizinin NIST-800-22 testlerinden başarılı bir şekilde geçtiği vurgulanmıştır [66].

Farklı bir çalışma Pareschi ve arkadaşları tarafından yapılmıştır. Çalışmada kriptografik uygulamalar için yeni bir kaos tabanlı GRSÜ modeli sunulmuştur. Tasarımın prototipi 0.18 μm ve 0.35 μm CMOS teknolojisi ile gerçekleştirilmiş ve çalışmanın doğrulanması için sisteme NIST testleri uygulanmıştır. Yapılan testlerde sistemin bit üretim hızı 80 Mbit/s'ye kadar çıktığı belirtilmiştir [67].

Çiçek ve arkadaşları tek boyutlu ayırık kaotik lojistik harita ve Bernoulli haritası kullanarak GRSÜ tasarımı yapmışlardır. Yapılan tasarım Alan Programlanabilir Analog Dizileri (Field Programmable Analog Array (FPAA)) ile gerçekleştirilmiştir. FPAA donanımı kullanılarak yapılan tasarımda ana çalışma frekansı 16 MHz ve sistemin bit üretim hızı 1.5 Mbit/s olarak belirtilmiştir. Önerilen GRSÜ kullanılarak elde edilen rasgele dizilerin NIST-800-22 testlerinden geçtiği ifade edilmiştir [68].

Özoğuz ve arkadaşlarının yaptıkları çalışmada RSÜ (Rasgele Sayı Üreteçleri) devresinde kullanılabilecek ve tümleştirilmeye uygun yeni kaos üreteçleri önerilmiştir. Ayrıca tasarım ortamında yapılan benzetimlerden sonra bu devrelerin 0.35 μm CMOS prosesi ile tümleşik devre üretimleri yapılmış ve GRSÜ oluşturulmuştur. Üretilen kaotik devrelerden 16 MHz ile 25 MHz arasında değişen

bant genişlikli kaotik işaretler elde edilmiştir. GRSÜ yapısının doğrulanması için FIPS-140-2 ve NIST-800-22 gibi rastgelelik testleri uygulanmıştır. Üretimi yapılan GRSÜ devresinin ortalama bit üretim hızı 2 Mbit/s olarak verilmiştir. Ayrıca çalışmada sunulan bipolar negatif-gm kaotik osilatörünün dinamik davranışının yüksek ısı duyarlılığına sahip olduğu belirtilmiştir [69].

Literatürde ve yukarıda sunulan çalışmalardan görüleceği üzere yoğun bir biçimde GRSÜ konusunda çalışmalar yapılmaktadır. Bu çalışmaların bir kısmı, kaos tabanlı GRSÜ'nin CMOS üretim süreci ile yapılmaktadır. Yapılan çalışmaların diğer bir kısmı da, klasik osilatörler ile GRSÜ'nin FPGA tabanlı olarak gerçekleştirilmesi üzerinedir.

Yukarıda özetlenen çalışmalardan da gözlemlendiği gibi çözümlenemeyen üç önemli sorun ortaya çıkmaktadır:

1. Klasik osilatörlerin yapısından kaynaklanan ortalama bit üretim hızının belirli bir hızın üzerine çıkamaması.
2. Analog elemanlar kullanılarak gerçekleştirilen osilatörlerin çalışma frekanslarının düşük olması.
3. Ayrıca CMOS-tabanlı ve analog elemanlar kullanılarak gerçekleştirilen yapıların sistem parametrelerinin değişimine dirençli olması.

1.1. Tezin Amacı

Sunulan tezin amacı gerçek zamanlı, yüksek çalışma frekansı ve bit üretim hızına sahip, dört farklı nümerik diferansiyel denklem çözüm yöntemi kullanılarak FPGA tabanlı kaotik GRSÜ'nin tasarlanması ve gerçekleştirilmesi ile yukarıda zikredilen sorunlara çözüm bulmaktır. Birinci sorunun çözümlenmesi amacıyla tez çalışmasında klasik osilatörler yerine kaotik tabanlı osilatörler kullanılarak ortalama bit üretim hızı artırılabilecektir. İkinci sorunun çözümüne yönelik olarak da yüksek esnekliğe sahip FPGA tabanlı sistem tasarımlarından yararlanılarak osilatörlerin çalışma frekansları artırılabilecektir. Üçüncü sorunun çözümü için ise çok kısa sürede yeniden programlanabilen FPGA tabanlı sayısal entegreler kullanılarak parametre değişimleri daha esnek bir hale getirilmiş olacaktır.

1.2. Tezde İzlenecek Yol

Bu tez çalışması gerçek zamanlı, yüksek çalışma frekansı ve bit üretim hızına sahip, FPGA tabanlı yeni kaotik GRSÜ'nin tasarlanması ve gerçekleştirilmesi amacı ile yedi bölüme ayrılmıştır. Bu amaçla ikinci bölümde tez çalışması ile ilgili temel kavramların tanıtımı yapılacaktır.

Üçüncü bölümde, seçilen örnek kaotik sistemler tanıtılacaktır. Seçilen örnek yeni kaotik sistemlerin dinamik davranışlarını belirlemek amacıyla nümerik benzetim ve analiz programı (Matlab) kullanılarak kaotik sistemin zaman serileri, faz portreleri, denge noktaları ve Lyapunov spektrumu analizleri yapılacaktır. Ardından seçilen kaotik sistemler bir ECAD programında şematik giriş yapılarak analog elektronik elemanlar ile modellenecek ve benzetime tabi tutulacaktır. Nümerik analiz sonuçları ile ECAD benzetim sonuçları karşılaştırılacaktır. Üçüncü aşama sonunda elde edilecek sonuçlar gelecek aşamalarda gerçekleştirilecek modellemelerin doğrulanması için referans olarak kullanılacaktır.

Dördüncü bölümde, FPGA tabanlı modelleme için örnek kaotik sistemler FPGA üzerinde donanım tanımlama dillerinden biri olan VHDL ile Euler, Heun, RK4 ve RK5-Butcher algoritmaları kullanılarak modellenecektir. Tasarımlarda 32-bit IEEE-754-1985 kayan noktalı sayı standardı kullanılacaktır. Yapılan tasarımların gerçekleştirilmesi amacıyla Virtex-6 FPGA çipi seçilmiştir. Tasarımı sonunda elde edilen FPGA çip istatistikleri ve maksimum çalışma frekansları sunulacaktır.

Beşinci bölümde, Euler, Heun, RK4 ve RK5-Butcher algoritmaları kullanılarak FPGA-tabanlı SPKS ve PWKS kaotik sistemleri ile gerçek rasgele sayı üretici tasarımı gerçekleştirilecektir. Geliştirilen modeller VHDL donanım tanımlama dili kullanılarak kodlanacak ve modellemelerde 32-bit IEEE 754-1985 kayan noktalı sayı standardı kullanılacaktır. GRSÜ ünitelerinin verileri işleme süresi, Xilinx ISE Design Tools 14.2 simülasyon programı kullanılarak elde edilecektir. Bu aşamada yapılan tasarımların FPGA çip istatistikleri, çalışma frekansları ve bit üretim hızları incelenecektir. Ayrıca tasarlanan ünitelerden elde edilen sonuçlar yorumlanacaktır.

Altıncı bölümde, FPGA tabanlı GRSÜ'nden elde edilen sayı dizileri test edilecektir. GRSÜ'nin oluşturduğu sayı dizilerinin rastgeleliğini test edecek veya kıyaslama imkânı sunacak matematiksel model veya benzetim sonucu bulunmamakla birlikte, uluslararası düzeyde uygulanan geçerli istatistiksel testleri bu diziler üzerinde uygulanarak, sayı dizilerinin rasgele olup olmadığı söylenebilmektedir. Bu testler, Rasgele Sayı Üreteci (RSÜ (Random Number Generator (RNG))) çıkışının gerçek bir rasgele diziden beklenenleri karşılayıp karşılamadığını belirlemektedir. Ayrıca testlerin sonuçlarına bakılarak RSÜ'nin kalitesi hakkında yorum yapılabilmektedir. Bir sayı dizisinin rasgele olduğunu söyleyebilmek için, test standardına (NIST, FIPS, vb.) ait tüm alt testlerden geçmesi gerekmektedir. Bu amaçla tasarımı yapılan GRSÜ'nin doğrulanması, uluslararası düzeyde uygulanarak kabul görmüş olan ve en çok kullanılan test sistemi olan FIPS-140-1 (Federal Information Processing Standard) testleri [70] ve National Institute of Standard and Technology 800-22 (NIST) testleri [71] ile sağlanacaktır.

Son bölümde ise gerçek zamanlı, yüksek çalışma frekansı ve bit üretim hızına sahip, FPGA tabanlı yeni kaotik GRSÜ'nin gerçekleşmesinden elde edilen sonuçlar irdelenecektir. Ayrıca ileride yapılabilecek çalışmalar hakkında öneriler sunulacaktır.

BÖLÜM 2. TEMEL KAVRAMLAR

Bu bölümde, FPGA-tabanlı kaotik osilatörlerin ve gerçek rasgele sayı üreteçlerinin tasarımı için gerekli bazı literatürel bilgiler verilmiştir. İlk olarak kaos ve kaotik sistemlerden bahsedilmiştir. Ardından tasarımların gerçekleştirileceği donanım ortamları olan FPGA çiplerine değinilmiştir. FPGA-tabanlı donanımlar sayısal sistemlerdir. Sürekli zamanlı kaotik sistemler diferansiyel denklemler ile ifade edilmektedir. Sürekli zamanlı kaotik sistemlerin FPGA donanımları kullanılarak modellenebilmesi için nümerik algoritmaların kullanılması gerekmektedir. Bu nedenle sonraki bölümde kaotik sistemlerin FPGA üzerinde modellenmesi amacıyla kullanılan nümerik algoritmalar sunulmuştur. Bir sonraki bölümde literatürde sunulan GRSÜ çeşitlerine ve ardından GRSÜ için geliştirilen istatistiksel rasgelelik testleri olan FIPS-140-1 ve NIST-800-22 testlerine değinilmiştir.

2.1. Kaos ve Kaotik Sistemler

Kaotik sistemler, ilk olarak bir matematikçi ve meteorolog olan Edward Norton Lorenz tarafından keşfedilmiştir. Lorenz, 1963 yılında meteorolog olarak çalışırken üç değişkenli bir sistemde başlangıç şartlarındaki çok küçük değişikliklerin belirli bir süre sonunda öngörülemez sonuçlar doğurabileceğini göstermiştir [72]. Literatürde kaos kelimesi ilk olarak 1975 yılında T. Y. Li ve J. A. Yorke tarafından “Üç periyot kaos anlamına gelir” isimli makalede kullanılmıştır [73]. Rössler 1976 yılında, yeni bir yedi terimli ikinci dereceden doğrusal olmayan ve Lorenz sisteminden daha basit olan bir kaotik sistem önermiştir [74]. Yine Rössler 1979 yılında, daha önceden 1975 yılında bulunduğu sistemden daha basit bir kaotik sistem geliştirmiştir [75]. 1986 yılında Leon Chua çok basit bir devre yapısına sahip olan otonom bir kaotik devre geliştirmiştir. Chua devresi olarak isimlendirilen devre, basit bir yapıya sahip olmasına rağmen üç değişkenli kaotik sistemlerin elektronik olarak gerçekleşmesi ve kaos olayının açıklanması için örnek bir model devre olmuştur [76].

Literatürde, geliştirilen ve üzerinde bilimsel çalışmalar yapılan kaotik sistemlere örnek olarak Lorenz [77], Chua [78], Sprott [79], Rössler [80], Rabinovich [81], Rikitake [82], Burke-Shaw [83] ve Chen [84] sistemleri verilebilir.

Kaotik sistemler genellikle adi diferansiyel denklemler ile ifade edilmektedir. Sürekli zamanlı n tane birinci dereceden adi diferansiyel denklem sistemi $i=1, 2, 3, \dots, n$ olmak üzere Denklem (2.1) ile verilebilir [85].

$$\left. \begin{aligned} dx^{(i)} / dt &= f_1(x^{(i)}, x^{(i+1)}, \dots, x^{(n)}) \\ dx^{(i+1)} / dt &= f_2(x^{(i)}, x^{(i+1)}, \dots, x^{(n)}) \\ &\vdots \\ dx^{(n)} / dt &= f_n(x^{(i)}, x^{(i+1)}, \dots, x^{(n)}) \end{aligned} \right\} \quad (2.1)$$

Bu bağıntılar düzenlenirse adi diferansiyel denklemler vektörel formda Denklem (2.2)'deki gibi verilebilir.

$$dx(t)/dt = F[x(t)] \quad (2.2)$$

Burada x , n boyutlu bir vektördür. Sürekli zamanlı kaotik sistemlerde $n \geq 3$ olmalıdır [36]. Denklem (2.3)'te örnek olarak literatürde sunulan ve üzerinde birçok çalışmalar yapılmış kaotik sistemlerden biri olan üç değişkenli Burke-Shaw Kaotik Sistemi (BSKS) diferansiyel denklemleri verilmektedir [86]. BSKS diferansiyel denklem takımlarındaki α ve β parametreleri sistem parametreleri ve $x \in R$ olmak üzere $x^{(1)}$, $x^{(2)}$ ve $x^{(3)}$ ise sistemin dinamik değişkenleri olarak adlandırılmaktadır. Sistem parametrelerinin değişimi kaotik sistemin dinamik karakteristiğini değiştirmektedir. Bu nedenle bu parametreler sistem davranışının belirlenmesinde oldukça büyük önem taşımaktadır.

$$\begin{aligned} dx^{(1)} / dt &= -\alpha \cdot x^{(1)} - \alpha \cdot x^{(2)} \\ dx^{(2)} / dt &= -\alpha \cdot x^{(1)} \cdot x^{(3)} - x^{(2)} \\ dx^{(3)} / dt &= \alpha \cdot x^{(1)} \cdot x^{(2)} + \beta \end{aligned} \quad (2.3)$$

2.2. FPGA Çipleri

FPGA çipleri, herhangi bir sayısal devre veya sistem tasarımları için elektriksel olarak programlanabilir tümleşik devre araçlarıdır. Bu çipler, bir saniyeden daha kısa bir süre içerisinde konfigüre edilebilmektedirler. Programlanabilir terimi, FPGA çiplerinin silikon üretiminin tamamlanmasından sonra çip içerisine bir sistem tasarımının yapılabileceğini ifade etmektedir. Bu sayede çipin üretimi yapıldıktan sonra bile sistem tasarımcısının istediği yerde çipin yapısını değiştirerek istediği tasarımı yapmasına imkân sağlamaktadır [87].

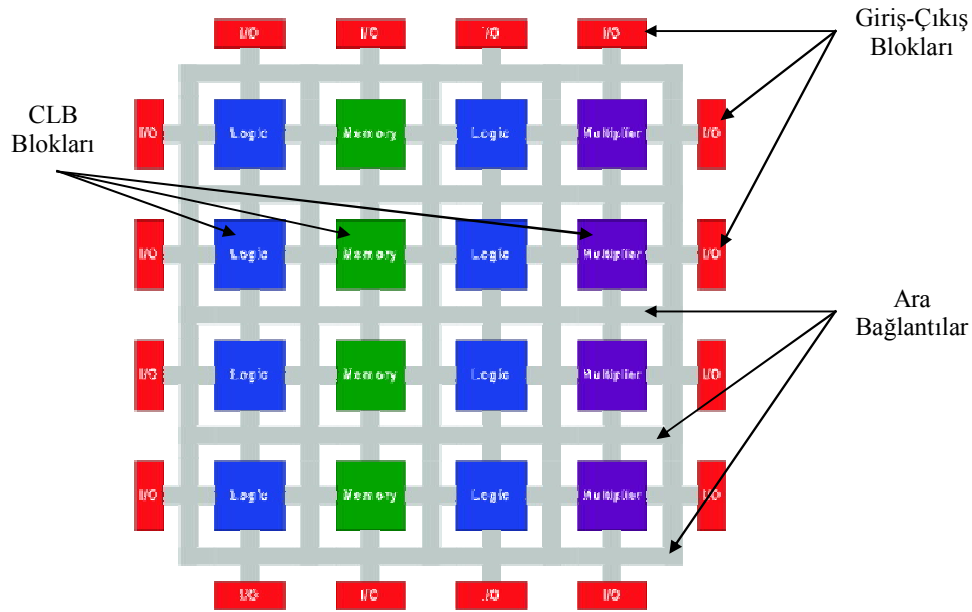
Günümüzde FPGA çipleri oldukça yüksek kapasiteye ve hıza sahip olmakla birlikte farklı uygulamaların gerçekleştirilmesine olanak sağlanabilmesi amacıyla her geçen gün daha kompleks bir yapıya sahip olmaktadır. Bu çipler genellikle “çip üzerinde sistem” (System On a Chip (SoC)) olarak da adlandırılmakta ve gereksinim duyulduğunda diğer çiplerle birlikte daha büyük sistemlerin parçaları olarak çalışabilmektedirler [88].

Genel amaçlı üretilerek sisteme özel programlanabilen FPGA çipleri gerçek zamanlı çalışma yeteneğine sahiptirler. Günümüzde yaklaşık olarak 1 GHz çalışma frekansına sahip FPGA çipleri bulunmaktadır [89]. FPGA çipleri, kişisel bilgisayarlara oranla daha düşük çalışma frekansına sahip olmalarına rağmen paralel çalışma ve sisteme özgü tasarım gibi avantajlarından dolayı kişisel bilgisayarlardan çok daha hızlı bir şekilde işlemleri gerçekleştirebilmektedirler [90].

FPGA çiplerinin bir diğer avantajı, çip içerisinde tasarımı yapılan bir sistemin birden fazla kopyası çalıştırılabilmekte ve bu sayede oldukça yüksek hız kazançları elde edilebilmektedir. FPGA-tabanlı tasarımların sağlamış olduğu diğer bir önemli avantaj ise IP-core (Intellectual Properties-core) yapıları oluşturularak modüler bir sayısal devre tasarımı ve hazır kütüphane dosyaları ile hızlı bir tasarım sürecinin sağlanmakta olmasıdır. Örneğin, bir mikroişlemci veya bir mikrodenetleyici IP-core olarak bir FPGA'in içine gömülebilmektedir [91]. Ayrıca, genel amaçlı bir bilgisayar ve buna bağlı üzerinde bir ya da daha fazla sayıda FPGA çipi bulunduran FPGA tabanlı Özel Amaçlı Bilgisayarlar (FPGA-based Custom Computing Machines (F-

CCMs)) ile sistem tasarımları yapılarak oldukça yüksek hızlı platformlar elde edilebilmektedir [92].

FPGA çipleri genel olarak üç farklı bölümden oluşmaktadır. Bunlar giriş-çıkış blokları (I-O blocks), ara bağlantılar (interconnection network) ve konfigüre edilebilir mantıksal bloklar (Configurable Logic Block (CLB))'dır. Şekil 2.1'de FPGA çiplerinin genel yapısı görülmektedir [87]. Giriş-çıkış blokları sistem tasarımına göre ister giriş ister çıkış birimi olarak veya hem giriş hem de çıkış bloğu olarak da kullanılabilirler. CLB'ler içerisinde hafıza, mantıksal blok ve çarpıcı elemanlarından oluşan programlanabilir bloklardır. Ara bağlantılar ise giriş-çıkış blokları ile hafıza, mantıksal blok ve çarpıcı elemanları arasındaki bağlantıyı sağlayan programlanabilir yapılarıdır [93].



Şekil 2.1. FPGA çipi genel yapısı

FPGA çiplerinin en büyük avantajlarından birisi paralel olarak işlem yapabilmeye kabiliyetleridir. Bu özellikleri sayesinde birçok programlanabilir donanımlara göre hız bakımından (mikroişlemci, DSP) daha yüksek başarıma sahiptirler. Yapılan tasarımların çipe yüklenerek gerçekleştirilme süresi bir kaç ms ile sınırlıdır. Ayrıca, FPGA çipleri maliyetleri düşük ve tekrar programlanabilme gibi özelliklerinden dolayı ASIC tabanlı uygulamalar için prototip olarak kullanılmaktadır. Bu gibi

avantajlarından dolayı, FPGA çipleri günümüzde endüstriyel otomasyon ve kontrol sistemlerinde motor kontrolünden endüstriyel görüntülemeye, uzay ve savunma sanayisinde kriptolu iletişimden elektronik harbe, tüketici elektroniğinde sayısal kameralardan uydu alıcılarına, tıbbi elektronikte bilgisayarlı tomografiden ultrason görüntülemeye ve otomotiv endüstrisinde görüntü işlemeden araç içi bilgi sistemlerine kadar çok geniş bir yelpazede kullanılabilir [94].

Günümüzde FPGA çipleri birçok firma tarafından üretilebilmektedir. Bu üreticilere örnek olarak Xilinx, Altera, Atmel, SiliconBlue, Microsemi ve Lattice örnek olarak verilebilir. Her üretici üretmiş olduğu FPGA çiplerine farklı isimler vermektedir. Örneğin Xilinx firması Spartan, Virtex, Kintex gibi isimler kullanırken, Altera firması ürettiği çiplere Cyclone ve Stratix gibi isimler vermektedir.

FPGA çiplerinde genellikle şematik yöntemler veya HDL (Hardware Description Language (Donanım Tanımlama Dili)) kullanılarak tasarımlar yapılmaktadır. Şematik yöntemler genel olarak küçük ölçekli tasarımlar için tercih edilmektedir. HDL ise orta ölçekli ve büyük ölçekli tasarımlarda kullanılmaktadır. Günümüzde HDL olarak en çok tercih edilen donanım tanımlama dilleri VHDL ve Verilog dilleridir. Bu diller sürekli olarak geliştirilmektedir. Örneğin VHDL dili 1987 tarihinde resmi olarak IEEE standart olarak kabul edilmiş, 1993 ve 2000 yıllarında güncellenmiştir.

2.3. Nümerik Algoritmalar

Bir fonksiyonun türevini bilinmeyen olarak bulunduran ifadelerle diferansiyel denklemler denilmektedir. Bir diferansiyel denklemi Denklem (2.4)'de $x_0=0$ için y_0 başlangıç şartını belirtmek üzere Denklem (2.5) formunda yazılabilir.

$$y(x_0) = y_0 \quad (2.4)$$

$$dy / dx = f(y, x) \quad (2.5)$$

$f: \mathfrak{R} \rightarrow \mathfrak{R}$ ve $x, y \in \mathfrak{R}$ olarak verilen bir $f(x, y)$ fonksiyonun $y=y_\lambda$ 'deki değeri ile türevleri tanımlı ve bu değerler biliniyorsa Taylor serisi açılımıyla (2.6), fonksiyonun y_λ 'deki tanımlı olan değeri ve türevleri ile $\lambda=l$ için y_λ 'den $\Delta h = y_{\lambda+1} - y_\lambda$ kadar uzaklıktaki fonksiyon değerleri hesaplanabilmektedir.

$$f(y_{\lambda+1}) = f(y_\lambda) + \frac{f'(y_\lambda)\tau}{1!} + \frac{f''(y_\lambda)\tau^2}{2!} + \dots + \frac{f^n(y_\lambda)\tau^n}{n!} \quad (2.6)$$

Taylor serisinin ilk iki terimi kullanıldığında aşağıdaki Denklem (2.7) elde edilmektedir.

$$\frac{df(y_\lambda)}{dx} = \lim_{\tau \rightarrow 0} \frac{f(y_{\lambda+1}) - f(y_\lambda)}{\tau} \quad (2.7)$$

Buradan $f(y_{\lambda+1})$ ifadesi çekilirse, Euler algoritması olarak adlandırılan denklem (2.8) elde edilmektedir. Euler algoritması, diferansiyel denklemlerin sayısal çözümü için geliştirilmiş en basit yöntemlerden birisidir. Bu yöntem, sayısal çözümünün kolay olması nedeniyle çok sık olarak tercih edilmektedir. Ancak, Euler algoritması genel olarak çok hassas çözümler üretememektedir. Ayrıca, kaotik sistemler başlangıç şartlarına çok hassas bağımlı olduklarından başlangıç şartlarındaki çok az bir değişim sistem dinamik davranışını oldukça değiştirmektedir. Sürekli zamanlı bir kaotik sistemin ayrık zamanlı olarak modellenmesi amacıyla kullanılan nümerik algoritmalarda, sistemin bir sonraki $y_{\lambda+1}$ değerinin hesaplanması için bir önceki değeri olan y_λ değeri kullanılmaktadır. Burada Δh adım sayısını belirtmektedir.

$$y_{\lambda+\Delta h} \approx y_{\lambda+1} = y_\lambda + f(y_\lambda)\Delta h \quad (2.8)$$

Sonuç olarak, Euler algoritması çok hassas sonuçlar üretemediğinden dolayı sayısal tabanlı kaotik sistemin dinamik davranışı değişebilmektedir. Literatürde sunulan bir diğer nümerik algoritma ise Euler yönteminden daha gelişmiş bir yöntem olan Heun yöntemidir. Bu yönteme ait denklemler (2.9) aşağıda verilmiştir [95].

$$\begin{aligned}
y_{\lambda+1}^0 &= y_{\lambda} + f(y_{\lambda})\Delta h \\
y_{\lambda+1} &= y_{\lambda} + \frac{f(y_{\lambda}) + f(y_{\lambda+1}^0)}{2} \Delta h
\end{aligned} \tag{2.9}$$

Heun algoritması iki adımdan oluşmaktadır. Birinci adımda, y_{λ} değeri kullanılarak $y_{\lambda+1}^0$ fonksiyonu hesaplanmaktadır. İkinci adımda ise $y_{\lambda+1}^0$ fonksiyonu ve y_{λ} değerleri kullanılarak sistemin bir sonraki değeri $y_{\lambda+1}$ hesaplanmaktadır. Heun algoritması, Euler algoritmasına göre daha hassas sonuçlar üretmesine rağmen yüksek frekanslı fonksiyonların eğimini yakalayamamaktadır. Bu nedenle, literatürde daha hassas sonuçlar üreten nümerik yöntemler geliştirilmiştir. Bu yöntemlerden bazılarına örnek olarak dördüncü dereceden Runge Kutta (RK4) algoritması [96], beşinci dereceden Runge Kutta Butcher (RK5-Butcher) algoritması [97], Dormand-Prince metodu [98] verilebilir. RK4 algoritması; Euler, Heun ve klasik RK algoritmasına göre oldukça iyi sonuçlar üretmekte ve bu algoritmada hata oranı oldukça düşük olmaktadır. Aşağıda RK4 algoritmasına ait denklemler (2.10) verilmiştir. Denklemden $y_{\lambda+1}$ değerinin hesaplanabilmesi için öncelikle k_1 , k_2 , k_3 ve k_4 değerleri hesaplanmalıdır. Burada k_1 , Δh kadar aralık sonundaki başlangıç eğimi, k_2 değeri Δh aralığının orta noktasındaki k_1 değeri kullanılarak hesaplanan eğimi, k_3 değeri Δh aralığının orta noktasındaki k_2 değeri kullanılarak hesaplanan eğimi ve k_4 değeri ise Δh aralığının sonundaki k_3 değeri kullanılarak hesaplanan eğimidir. Bu şekilde sayısal olarak y_{λ} değeri ve Δh aralık değerleri kullanılarak sistemin bir sonraki değeri olan $y_{\lambda+1}$ değeri hesaplanmaktadır [99].

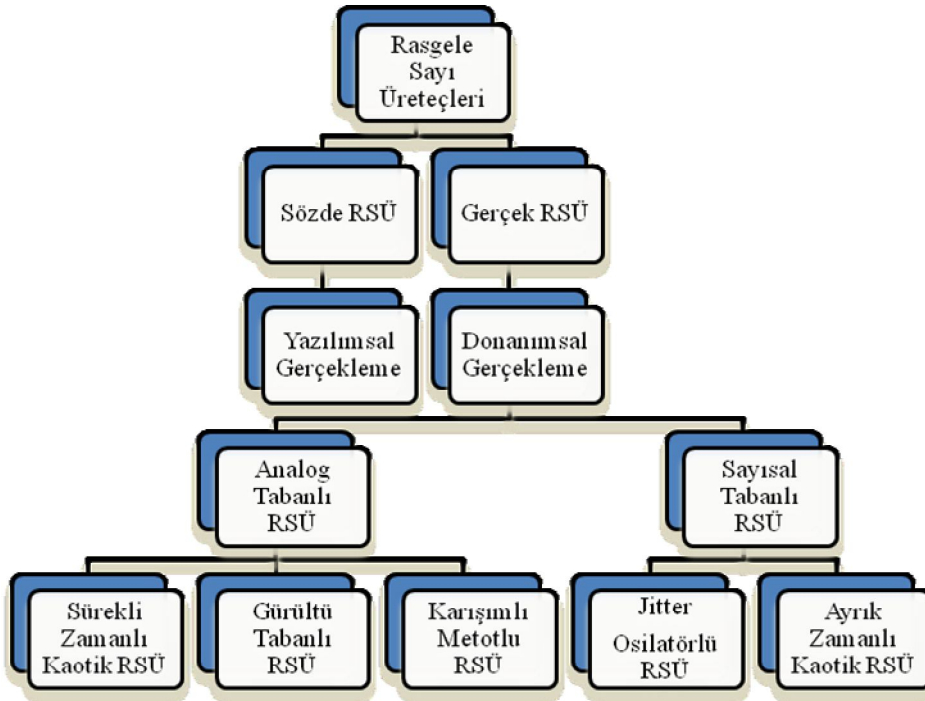
$$\begin{aligned}
y_{\lambda+1} &= y_{\lambda} + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)\Delta h \\
k_1 &= f(y_{\lambda}) \\
k_2 &= f\left(y_{\lambda} + \frac{\Delta h}{2}k_1\right) \\
k_3 &= f\left(y_{\lambda} + \frac{\Delta h}{2}k_2\right) \\
k_4 &= f(y_{\lambda} + \Delta hk_3)
\end{aligned} \tag{2.10}$$

Aşağıda RK5-Butcher algoritmasına ait denklemler (2.11) sunulmuştur. RK5-Butcher algoritması; k_1, k_2, k_3 ve k_4, k_5 ve k_6 olmak üzere altı adımdan oluşan bir algoritmadır. Yapısal olarak RK4 algoritmasından çok fazla bir farkı bulunmamakla beraber k_5 ve k_6 parametreleri eklenmiştir. Bu parametrelerin eklenmesi algoritmanın yazılımsal veya donanımsal olarak gerçekleşmesini zorlaştırmış ve sonuç üretim hızını düşürmüştür. Ancak bu parametrelerin eklenmesiyle RK5-Butcher algoritması, RK4 algoritmasına göre daha hassas sonuçlar üretmektedir.

$$\begin{aligned}
y_{\lambda+1} &= y_{\lambda} + \frac{1}{90}(7k_1 + 32k_3 + 12k_4 + 32k_5 + 7k_6)\Delta h \\
k_1 &= f(y_{\lambda}) \\
k_2 &= f\left(y_{\lambda} + \frac{\Delta h}{4}k_1\right) \\
k_3 &= f\left(y_{\lambda} + \frac{\Delta h}{8}k_1 + \frac{\Delta h}{8}k_2\right) \\
k_4 &= f\left(y_{\lambda} - \frac{\Delta h}{2}k_2 + \Delta h k_3\right) \\
k_5 &= f\left(y_{\lambda} + \frac{3\Delta h}{16}k_1 + \frac{9\Delta h}{16}k_4\right) \\
k_6 &= f\left(y_{\lambda} - \frac{3\Delta h}{7}k_1 + \frac{2\Delta h}{7}k_2 + \frac{12\Delta h}{7}k_3 - \frac{12\Delta h}{7}k_4 + \frac{8\Delta h}{7}k_5\right)
\end{aligned} \tag{2.11}$$

2.4. Gerçek Rasgele Sayı Üreteçleri

Kriptografik uygulamalarda kullanılması gereken en temel yapılardan birisi RSÜ'dür [100]. Günümüzde gerçek rasgele sayı üreteçlerinde kaotik sistemler rasgelelik kaynağı olarak kullanılmaktadır [101–104]. Kaotik sistemlerin analog veya sayısal devreler ile kolay bir şekilde gerçekleştirilmeleri, oldukça düşük güçlerde çalışmaları ve ASIC/FPGA gibi sayısal tabanlı sistemlerde yüksek frekanslarda çalışabilmeleri bu sistemleri RSÜ çalışmalarında kullanımlarını daha çekici bir hale getirmektedir. RSÜ, genel olarak Sözde RSÜ ((SRSÜ) Pseudo RNG) ve Gerçek RSÜ (True RNG) olarak iki kısma ayrılmaktadır [102]. Şekil 2.2'de RSÜ'ler genel olarak sınıflandırılması görülmektedir [103].

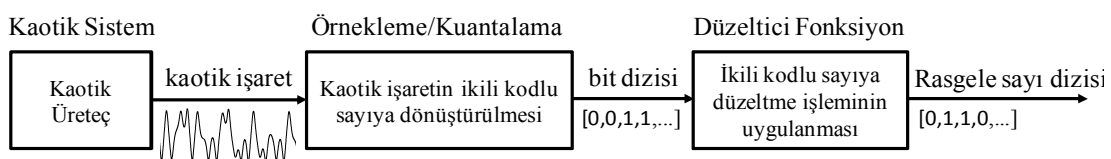


Şekil 2.2. Rasgele sayı üreteçlerinin sınıflandırılması

SRSÜ, yapay rasgele sayılar üreten ve fiziksel olmayan bir yapıya sahip olmakla birlikte bu üreteçlerin ürettiği sayılar tahmin edilebilirler. Çünkü SRSÜ’nde yapısal olarak deterministik bir algoritma veya formül çalışmaktadır [105]. SRSÜ sonlu durum makinelerinde kullanılan algoritmalar olup, rasgele görünen sayılar üretebilmektedirler. Ayrıca bu yapılar, basit ve hızlı bir şekilde gerçekleştirilerek periyotları çok uzun olduğundan pek çok istatistik testlerden geçebilmektedir. Buna rağmen kriptolojik uygulamalarda, SRSÜ, GRSÜ’ne göre güvenilirlik açısından yetersiz kalmaktadır [106]. GRSÜ’lerin tasarımında ise, gürültü kaynaklarının (ısı gürültü ve saçılma gürültüsü) doğrudan yükseltilmesi ve örneklenmesi, osilatör örnekleme yöntemi ve kaotik sistemler kullanılmaktadır [107]. Gürültü kaynaklarının doğrudan yükseltilmesi ve örneklenmesi-tabanlı GRSÜ gerçeklemelerinin çeşitli olumsuz yönleri bulunmaktadır. Kullanılan gürültü kaynaklarının ürettiği işaretlerin oldukça düşük güçlü olmaları ve bu nedenle sistemdeki istenmeyen işaretlerden etkilenmeleri ve ayrıca gürültü tabanlı GRSÜ’lerinde gürültü sinyalinin kuvvetlendirilmesi için kullanılacak olan kuvvetlendiricilerin kazancındaki bant sınırlaması sebebiyle, kuvvetlendirici çıkışında gürültü bandının da sınırlı olması, bu olumsuzluklara örnek olarak verilebilir [69].

Osilatör örnekleme yöntemine dayalı RSÜ’nde genel olarak yavaş osilatörün yükselen kenarlarında hızlı osilatörün çıkışları örneklenmektedir. Literatürde seçirmeli yavaş osilatör sinyalinin üretmek için kullanılan alt devreler değişiklik göstermektedir. Bu şekilde gerçekleştirilen RSÜ’nin bit üretim hızı, seçirme gürültülü yavaş osilatörün ortalama frekansına diğer bir deyişle gerilim kontrollü osilatör (Voltage Control Oscillator (VCO)) serbest salınım frekansına eşit olmaktadır. Osilatör örnekleme yöntemine dayalı RSÜ’lerin besleme gerilimlerinden gelen istenmeyen işaretlere ve gürültünün sınırlı bant genişliğinden gelen olumsuzluklara, gürültünün doğrudan kuvvetlendirilmesine göre daha az duyarlıdır [69, 108].

GRSÜ devrelerinin gerçekleştirilmesinde gürültü kaynağı yerine kaotik sistem yapıları kullanılabilir. Şekil 2.3’te kaos tabanlı RSÜ blok şeması görülmektedir. Düzensiz davranışları ve başlangıç koşullarına hassas oluşları nedeniyle, kaotik işaretler de rastgelelik kaynağı olarak değerlendirilmekte ve GRSÜ yapımında kullanılmaktadırlar. Kaotik sistemleri ayrık ve sürekli olarak ikiye ayırmak mümkündür. Her iki tip sistemle oluşturulan GRSÜ yapıları mevcuttur [39, 109]. Bu yapılarda kullanılan kaotik tabanlı üreteçler başlangıç koşullarına ve sistem parametrelerine oldukça hassas bağımlı olup, bu sistemlerin çözümlerinin uzun zaman aralıkları için öngörülmesi mümkün olmamaktadır. Bunun yanında, kaotik işaretlerin periyodik olmamaları nedeniyle, frekans yayılımları geniş ve sürekli olmaktadır [69]. Kaotik sistemlerin yukarıda ifade edilen önemli özellikleri sayesinde yüksek bit üretim hızına sahip GRSÜ uygulamalarında kullanılabilirliği görülmektedir [106]. Kaotik sistemler, dinamik sistemler olduklarından, osilatör çıkışları zamana bağlı olarak değişim göstermektedir. İkinci olarak bu sistemler, değişken ve periyodik olmayan yapılar olduklarından kendilerini tekrarlamazlar. Ayrıca kaotik sistemler, kararsız ve giriş koşullarına üstel duyarlı yapılar olduklarından dolayı yapılacak dinamik sistem davranış analizi sonuçları incelendiğinde bu sistemlerin rasgele davranışlar sergilediği gözlemlenmektedir.



Şekil 2.3. Kaos tabanlı RSÜ blok şeması

2.5. İstatistiksel Rasgelelik Testleri

RSÜ yapılarının üretmiş olduğu ikili kodlu bit dizilerinin rasgele olduğu matematiksel olarak kanıtlanamamakla birlikte, uluslararası düzeyde kabul görmüş bazı istatistiksel testleri uygulayarak, sayı dizilerinin rasgele olduğu veya bunun tersi söylenebilmektedir [110]. Bu testler, üreticinin çıkışının gerçek bir rasgele diziden beklenen rasgelelik düzeyini karşılayıp karşılamadığını söyleyebilmektedir. Ayrıca testlerin sonuçlarına bakılarak RSÜ'nin kalitesi hakkında yorum yapılabilir. Bir sayı dizisinin rasgele olduğunu söyleyebilmek için, tüm bu istatistiksel testlerden geçmesi gerekmektedir. Bilinen testlerden başlıca iki tanesi, "Kriptografik modüller için güvenlik araçları" isimli çalışma olarak bilinen Federal Bilgi İşleme Standartları 140-1 (Federal Information Processing Standards (FIPS 140-1)) testi [70] ve Ulusal Standartlar ve Teknoloji Enstitüsü (National Institute of Standards and Technology (NIST)) tarafından yayınlanan "Kriptografik uygulamalar için rasgele ve sözde rasgele sayı üreteçleri için bir istatistiksel test aracı" (A statistical test suite for random and pseudo random number generators for cryptographic applications) ismiyle yayınlanan NIST-800-22 testidir [71]. FIPS-140-1 testi, blok uzunluğu 20 Kbit gibi küçük boyutlu bit dizilerini test etmek amacıyla kullanılmaktadır. NIST-800-22 testi, FIPS-140-1 testine göre 100 Kbit, 500 Kbit ve 1 Mbit gibi daha büyük boyutlu bit dizilerini test etmek için kullanılmaktadır. NIST-800-22 testleri, FIPS-140-1 testlerine göre daha zor ölçütler kullanılarak yapılan testler olarak görülmektedir.

2.5.1. FIPS-140-1 testi

FIPS-140-1 testi dört testten oluşmaktadır. RSÜ'nin çıkışından alınan ve genel olarak ikili sayı sisteminde 20 Kbit'lik bit dizisi dört teste tabi tutulmaktadır. Bit dizinin rasgele kabul edilebilmesi için tanımlı dört testten geçmesi gerekmektedir. Bu testler Monobit, Poker, Koşu ve Uzun Koşu testleridir.

Monobit Testi (Monobit Test): Bu testin amacı, RSÜ tarafından üretilen bit dizisindeki '0' ve '1' dağılım oranının rasgele bir diziden beklendiği gibi olup olmadığını tespit etmektir. Testin başarılı olabilmesi için 20 Kbit'lik bir dizideki '1'

sayısının $9654 < n < 10346$ aralığında olması gerekmektedir. Eğer dizideki '1' sayısını ifade eden n belirtilen aralıklar içerisinde ise test başarılıdır. Aksi takdirde bit dizisi için monobit testi başarısız sayılmaktadır [111].

Poker Testi (Poker Test): Bu testte, ' k ' bit dizisinin uzunluğunu belirtmek üzere, $k \geq 5 \cdot 2^m$ olacak şekilde, üst üste çakışmayan m bitlik parçalara ayrılmaktadır. Burada i . parça n_i diye adlandırılmaktadır. Rasgele bir bit dizisinden beklenen, k uzunluklu bir bit dizisinde tüm m bitlik blok parçalarının aynı sayıda birbirini tekrar etmesidir. Test için Denklem (2.12)'de verilen denklem kullanılmaktadır [112].

$$X = \frac{2^m}{k} \left(\sum_i^{2^m} n_i^2 \right) - k \quad (2.12)$$

Poker testinde bit dizisinin başarılı sayılabilmesi için m bitlik blok parçalarının birbirini tekrar etme sayısı olan X değerinin, $k=20000$ ve $m=4$ için, $1.03 < X < 57.4$ aralığında olması gerekmektedir.

Koşu Testi (Run Test): Bu testte RSÜ'den elde edilen bit dizisinin başarılı sayılabilmesi için, bit dizisinde ardı ardına gelen '1' ve '0'lerden oluşan çeşitli uzunluktaki bit blok sayısının Tablo 2.1'de belirtildiği aralıklar içerisinde olması beklenmektedir. Burada x bit dizisinin koşu uzunluğunu göstermektedir. Eğer bit blok uzunluğu 6 bitten daha fazla ise bu uzun bloklar 6 bit olarak kabul edilmekte ve blok sayısı artırılmaktadır [70].

Tablo 2.1. Koşu testi için blok uzunluklarına göre blok sayıları

Blok Uzunluğu	Blok sayısı aralığı
1	$2267 \leq x \leq 2733$
2	$1079 \leq x \leq 1421$
3	$502 \leq x \leq 748$
4	$223 \leq x \leq 402$
5	$90 \leq x \leq 223$
6 ve 6+	$90 \leq x \leq 223$

Uzun Koşu Testi (Long Runs Test): Bu testte amaç, 34 veya daha fazla ardışık '0' veya '1' değerlerinin olup olmadığının tespit edilmesidir. Uzun koşu testinin başarılı kabul edilebilmesi için 20 Kbit uzunluğundaki bit dizisinin içerisindeki ardışık '0' veya '1' değerlerinin uzunluklarının 34'ten küçük olması beklenmektedir. Bu şart sağlanmadığı takdirde bit dizisi testi başarısız olmaktadır [113].

2.5.2. NIST-800-22 testi

Uluslararası düzeyde kabul görmüş olan testlerden bir diğeri ise NIST-800-22 testidir. NIST-800-22 testi, FIPS-140-1 testine göre hem test sayısı açısından daha fazla test içermekte hem de bit dizileri daha güçlü bir şekilde testlere tabi tutulmaktadır. FIPS-140-1 testinde başarılı olabilen bir bit dizisi NIST-800-22 testinden başarısız olabilmektedir. Bu nedenle NIST-800-22 testi daha güvenilir bir test olarak tercih edilmektedir. NIST-800-22 testi 15 farklı test barındırmaktadır. Bit dizisinin başarılı sayılabilmesi için 15 testin tamamından başarıyla geçmesi gerekmektedir. Bu testler aşağıda verilmektedir [71]. NIST-800-22 testinde, test edilecek rasgele bit dizisinin bazı parametreleri dışarıdan belirlenmektedir. Bu testlerde en önemli parametrelerden birisi olan *P-değeri* teste tabi tutulan rasgele dizilerin rasgeleliğinin bir ölçütü olarak kabul edilmektedir. *P-değeri* gerçekten rasgele bir dizi için 1'e yakın, bunun tersi durumunda ise *P-değeri* 0'a yakın olmaktadır.

1. Frekans Testi (The Frequency Test)
2. Bir Blok içerisinde Frekans Testi (Frequency Test within a Block)
3. Akış Testi (The Runs Test)
4. Bir Blok içerisinde En Uzun Birler Akış Testi (Tests for the Longest-Run-of-Ones in a Block)
5. İkili Matris Derece Testi (The Binary Matrix Rank Test)
6. Ayrık Fourier Dönüşüm Testi (The Discrete Fourier Transform (Spectral) Test)
7. Örtüşmeyen Şablon Eşleştirme Testi (The Non-overlapping Template Matching Test)
8. Örtüşen Şablon Eşleştirme Testi (The Overlapping Template Matching Test)
9. Maurer'in "Evrensel İstatistik" Testi (Maurer's "Universal Statistical" Test)

10. Doğrusal Karmaşıklık Testi (The Linear Complexity Test)
11. Seri Testi (The Serial Test)
12. Yaklaşık Entropi Testi (The Aproximate Entropy Test)
13. Birikimli Toplamlar Testi (The Cumulative Sums Test)
14. Rasgele Gezinimler Testi (The Random Excursions Test)
15. Rasgele Gezinimler Değişken Testi (The Random Excursions Variant Test)

NIST-800-22 testlerinde bulunan testler rasgele üreteçler tarafından üretilen verilerin rasgelelik ölçüsünün belirlenebilmesi amacıyla geliştirilmiş istatistiksel testlerdir. Bu testlerin her biri rasgele olduğu varsayılan verilerin, farklı istatistikî yöntemlerle incelenerek hipotezin karara bağlanmasını sağlamaktadır. Aşağıda bu testler ayrıntılı bir biçimde anlatılmaktadır.

İstatistiksel hipotez testleri, rasgele sayı üreteçlerinin ürettiği sayı dizilerine uygulanan istatistiksel testlerin sonuçlarını yorumlamak amacıyla kullanılmaktadır. Tablo 2.2’de istatistiksel hipotez testi sonuçları verilmektedir. Hipotez testlerinde öncelikle bir sıfır hipotezi (H_0) öne sürülür. Bu hipotezin tersi ise alternatif hipotez (H_a) olarak adlandırılmaktadır. H_0 hipotezi, üzerinde istatistiksel testler yapılan verilerin rasgele olduğu, H_a hipotezi ise üzerinde istatistiksel testler yapılan verilerin rasgele olmadığı anlamına gelmektedir [114].

Tip I hatası, test edilen veriler gerçekte rasgele iken verilerin rasgele olmadığına karar verilmesi olarak adlandırılmaktadır. Bu durum, veriler iyi bir üreteç ile üretilse bile verilerin rasgele olmayan özelliklerinin tespit edilebileceğini göstermektedir [115].

Tip II hatası ise gerçekte rasgele olmayan verilerin rasgele olduğu kararının verildiği durumu ifade etmektedir. Bu durumda veriler kötü bir rasgele sayı üreteci tarafından üretilse bile dizinin rasgele özelliklerinin tespit edilebileceğini belirtmektedir [115].

Her istatistiksel testte, öne sürülen hipotez için bir test istatistik *P-değeri* hesaplanır. Bu değere göre hipotez kabul edilmekte veya reddedilmektedir. Her istatistiksel test için bir α önem seviyesi (significance level) belirlenmektedir. *P-değeri* $\geq \alpha$ ise H_0

hipotezi kabul edilmekte diğere bir deęişle üzerinde rasgelelik testi yapılan veriler rasgele olduđu anlamına gelmektedir. P -deęeri $< \alpha$ ise H_0 hipotezi reddedilir. Bir diğere deyişle verilerin rasgele olmadığı anlamına gelmektedir. Genellikle önem seviyesi deęeri $0.01 < \alpha < 0.001$ aralığındadır. Eđer $\alpha = 0.01$ için P -deęeri $\geq \alpha$ ise H_0 hipotezi kabul edilerek üzerinde rasgelelik testi yapılan verilerin %99 doğrulukta rasgele olduđu kararına varılmaktadır. Eđer $\alpha = 0.001$ için P -deęeri $\geq \alpha$ ise H_0 hipotezi kabul edilerek üzerinde rasgelelik testi yapılan verilerin %99.9 doğrulukta rasgele olduđu kararı verilmektedir. Yapılan bu tez çalışmasında P -deęeri = 0.001 olarak alınmaktadır.

Tablo 2.2. İstatistiksel hipotez testi sonuçları

Gerçek Durum	Sonuçlar	
	H_0 kabul	H_a kabul (H_0 red)
Veri rasgele (H_0 doğru)	Hata yok	Tip I hatası
Veri rasgele deęil (H_a doğru)	Tip II hatası	Hata yok

NIST-800-22 testlerinden bazılarının hesaplanması sürecinde referans olarak kullanılan iki dağılım vardır. Bunlar standart normal dağılım ve Ki-kare (X^2) dağılımıdır. Testlerin hesaplanmasından önce bu dağılımlardan bahsedilecek ardından testlerin anlatılmasına geçilecektir.

Standart Normal Dağılım: Bu dağılım, gerçekten rasgele üretilmiş sayılar ile testi yapılan ve rasgele olduđu hipotezi savunulan veriler arasındaki karşılaştırma işlemini yapmak için kullanılan istatistiksel bir yöntemdir. Bu yöntem için x örnek test istatistik deęerleri ($-\infty < x < \infty$), μ beklenen deęer ($-\infty < \mu < \infty$) ve σ^2 test istatistiğinin varyansı ($\sigma^2 > 0$) olmak üzere olasılık yoğunluk fonksiyonu Denklem (2.13) verilmektedir.

$$\Phi(z) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad (2.13)$$

Burada x rasgele deęişkenine normal daęılıma sahip denilmekte ve $x \sim N(\mu, \sigma^2)$ şeklinde gösterilmektedir. Eęer burada $\sigma^2=1$ ve $\mu=0$ deęerini alır ise bu daęılıma standart normal daęılım denilmektedir. Bu daęılımda rasgele deęişken z olmak üzere Denklem (2.14) kullanılarak hesaplanmaktadır.

$$z = \frac{x - \mu}{\sigma} \quad (2.14)$$

Sonuç olarak z rasgele deęişkeninin olasılık yoğunluk fonksiyonu

$$\Phi(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2} \quad (2.15)$$

eşitlięi kullanılarak hesaplanmakta ve z deęişkeni standart normal daęılıma sahip bir rasgele deęişken olmaktadır [115].

Ki-kare (χ^2) Daęılımı: Rasgele olduęu hipotezi öne sürülen verilerin daęılımı ile beklenen verilerin daęılımını karşılaştırmak amacıyla kullanılan istatistiksel yöntem ise χ^2 daęılımıdır. Rasgele sayıların gözlemlenen (observed) frekansı o_i , beklenen (expected) frekans e_i ve $i=1 < k < \infty$ olmak üzere gözlemlenen χ^2 istatistiksel test daęılımı Denklem (2.16) ile hesaplanmaktadır.

$$\chi^2 (obs) = \sum_{i=1}^k \left((o_i - e_i)^2 / e_i \right) \quad (2.16)$$

NIST-800-22 testlerinde bulunan ve verilerin rasgelelik ölçüsünün belirlenebilmesi amacıyla geliştirilmiş 15 istatistiksel test aşağıda ayrıntılı bir biçimde anlatılmaktadır.

1. Frekans Testi (Frequency Test): Bu testin amacı bit dizisindeki '1' ve '0' dengesini incelemektir. Burada $\varepsilon = \varepsilon_1, \varepsilon_2, \dots, \varepsilon_n$ olmak üzere üretilen bit dizisi, n bit dizisinin uzunluęu, $S_n = X_1 + X_2 + \dots + X_n$ olmak üzere bit dizisinin '0' ve '1' deęerlerinin

$X_i = 2\varepsilon_i - 1$ dönüşümü kullanılarak -1 ve $+1$ değerlerine normalize edilmek suretiyle bit dizisinin toplanması ile elde edilen değerdir. S_{obs} gözlemlenen değer (2.17) kullanılarak hata fonksiyonu (error function) olan erf fonksiyonu (2.18) hesaplanmaktadır. Buradan erf fonksiyonu değeri kullanılarak Gauss hata fonksiyonu olarak isimlendirilen ve özel bir hata fonksiyonu olan $erfc$ fonksiyonu (complementary error function) (2.19) hesaplanmaktadır.

$$S_{obs} = \frac{|S_n|}{\sqrt{n}} \quad (2.17)$$

$$erf(z) = \frac{2}{\sqrt{\pi}} \int_z^{\infty} e^{-u^2} du \quad (2.18)$$

$$erfc(u) = 1 - erf(u) \quad (2.19)$$

Testte kullanılan P -değeri'nin hesaplanabilmesi için yukarıdaki denklemler yardımıyla Denklem (2.20) kullanılmaktadır.

$$P - value = erfc\left(\frac{S_{obs}}{\sqrt{2}}\right) \quad (2.20)$$

Örneğin bit dizisi $\varepsilon = \varepsilon_1, \varepsilon_2, \dots, \varepsilon_{12} = 011010011101$ olsun. Buradan $n=12$ olmaktadır.

$S_n = (-1) + 1 + 1 + (-1) + 1 + (-1) + (-1) + 1 + 1 + 1 + (-1) + 1 = 1$ olarak bulunur. Buradan

$$S_{obs} = \frac{|1|}{\sqrt{12}} = 0.289 \text{ ve } P - value = erfc\left(\frac{0,289}{\sqrt{2}}\right) = 0.773 \text{ olarak elde}$$

edilmektedir. Sonuç olarak P -değeri $= 0.773 \geq 0.001$ olduğundan H_0 hipotezi kabul edilir veya diğer bir deyişle verilerin rasgele olduğu kararına varılmaktadır [71, 115].

2. Blok Frekans Testi (Block Frequency Test): Frekans testi tüm bit dizisini incelerken, blok frekans testi ise bit dizisini M bitlik bloklara ayırarak blok içerisindeki '1' oranını incelemektedir. Rasgele üretilen M bitlik bloklardaki '1' oranının $M/2$ olması beklenmektedir. Blok uzunluğu $M=1$ olarak alındığında blok

frekans testi, frekans testi ile aynı işlevi görmektedir. Testin geçerli sonuçlar üretebilmesi için veri bit uzunluğu en az $n=100$ ve blok uzunluğu $M=20$ olmalıdır.

Test istatistiklerinin ve referans dağılımının hesaplanması için ki-kare (χ^2) dağılımı kullanılmaktadır. M bitlik her bir bloktaki '1' değerlerinin oranını belirlemek amacıyla $1 \leq i \leq N$ ve $N=n/M$ olmak üzere Denklem (2.21) kullanılarak

$$\pi_i = \frac{\sum_{j=1}^M \mathcal{E}_{(i-1)M+j}}{M} \quad (2.21)$$

χ^2 dağılım istatistiği $\chi^2(obs)$ aşağıdaki şekilde (2.22) hesaplanmaktadır.

$$\chi^2(obs) = 4M \sum_{i=1}^N (\pi_i - 1/2)^2 \quad (2.22)$$

Elde edilen $\chi^2(obs)$ değeri Denklem (2.23)'te yerine konularak P -değeri elde edilmektedir. Burada $igamc$, bir yaklaşım formülü olarak a ve x değişkenlerine bağımlı olan tamamlanmamış gama fonksiyonunu ifade etmektedir.

$$P - value = igamc \left(\frac{N}{2}, \frac{\chi^2}{2} \right) \quad (2.23)$$

Örneğin bit dizisi $\varepsilon = \varepsilon_1, \varepsilon_2, \dots, \varepsilon_{100} = 10110110101101100010101010101011100111000110110001010001100100110101010011010100101011001100111001110$ ve $M=10$ olsun.

Buradan $n=100$, $N=100/10=10$,

$\pi_1=3/5$, $\pi_2=1/2$, $\pi_3=1/2$, $\pi_4=3/5$, $\pi_5=1/2$, $\pi_6=2/5$, $\pi_7=1/2$, $\pi_8=2/5$, $\pi_9=3/5$, $\pi_{10}=1/2$ ve $\chi^2(obs)=2$ olarak bulunur.

Buradan $igamc$ $Q(a,0)=1$ ve $Q(a,\infty)=0$ olmak üzere Denklem (2.24)

$$Q(a, x) \equiv \frac{\Gamma(a, x)}{\Gamma(a)} \equiv \frac{1}{\Gamma(a)} \int_x^{\infty} e^{-t} t^{a-1} dt \quad (2.24)$$

kullanılarak bulunur. Bulunan değerler P -değerinde yerine konulursa;

$$P - value = igamc \left(\frac{N}{2}, \frac{\chi^2}{2} \right) = igamc \left(\frac{10}{2}, \frac{2}{2} \right) = igamc (5, 1) = 0.99$$

olarak elde edilmektedir [116]. Sonuç olarak P -değeri $= 0.99 > 0,001$ olduğundan H_0 hipotezi kabul edilir veya diğer bir ifade ile gözlemlenen veri dizileri rasgele olarak kabul edilmektedir [71, 115].

3. Akış Testi (Runs Test): Dizideki 0 ve 1 bloklarının osilasyonunun değişimini belirlemektedir. Bu şekilde '0' ve '1' değerleri arasındaki değişimlerin yavaş veya hızlı olması hakkında fikir edinilebilmektedir [117]. Testin yapılabilmesi için $\tau = 2/\sqrt{n}$ olmak üzere Denklem (2.25) şartının sağlanması gerekmektedir.

$$\left| \pi - \frac{1}{2} \right| \geq \tau \quad (2.25)$$

Burada ε üretilen bit dizisi, n bit dizisinin uzunluğu, π bit dizisindeki '1' değerlerinin sayısı, $V_n(obs)$ bit osilasyon sayısını göstermektedir. Eğer $\varepsilon_k = \varepsilon_{k+1}$ ise $r(k)=0$ ve diğer durumda $r(k)=1$ olarak kabul edilerek akışların sayısı olarak ifade edilen $V_n(obs)$ aşağıdaki şekilde Denklem (2.26) hesaplanmaktadır.

$$V_n(obs) = \sum_{k=1}^{n-1} r(k) + 1 \quad (2.26)$$

Bu testte kullanılan P -değerinin hesaplanabilmesi için Denklem (2.26) yardımıyla Denklem (2.27) kullanılmaktadır.

$$P - value = \operatorname{erfc} \left(\frac{|V_n(\text{obs}) - 2n\pi(1 - \pi)|}{2\sqrt{2n\pi(1 - \pi)}} \right) \quad (2.27)$$

Örneğin bit dizisi $\varepsilon = \varepsilon_1, \varepsilon_2, \dots, \varepsilon_{10} = 1001101011$ olsun.

$n=10$ ve $\pi=6/10=3/5$ bulunur ve $\tau = \frac{2}{\sqrt{10}} = 0.63$ olarak hesaplanmaktadır. Buradan, $|\pi-1/2|=|0,6-0,5|=0.1$ değeri elde edilmektedir.

Denklem (2.25)'e göre $0.1 \geq 0.63$ şartı sağlanamadığından akış testi yapılamaz.

Farklı bir bit dizisi $\varepsilon = \varepsilon_1, \varepsilon_2, \dots, \varepsilon_{100} = 110010010000111111011010101000100010001001010001100001000110100110001001100011001100010100010111000$ olsun.

Buradan $n=100$, $\tau = \frac{2}{\sqrt{100}} = 0.02$ ve $\pi=0.42$ değeri elde edilmektedir. Denklem (2.25)'e göre $0.42 \geq 0.02$ şartı sağlandığından akış testi yapılabilir. Bit osilasyon sayısı Denklem (2.26) yardımıyla aşağıdaki gibi hesaplanmaktadır.

$$V_n(\text{obs}) = \sum_{k=1}^{99} r(k) + 1 = 52$$

Buna göre elde edilen değerler P -değeri için yerine konulursa;

$$P - value = \operatorname{erfc} \left(\frac{|52 - 2 \cdot 100 \cdot 0.42(1 - 0.42)|}{2\sqrt{2 \cdot 100 \cdot 0.42(1 - 0.42)}} \right) = 0.501$$

değeri elde edilir. Sonuç olarak P -değeri $= 0.501 \geq 0.001$ olduğundan dizi rasgele olarak kabul edilmektedir [71, 115].

4. En Uzun Akış Testi (Longest Run Test): Testin amacı M bitlik blokların içerisinde en uzun '1' akışını incelemektir. Test edilecek n bitlik dizi M adet bloğa bölünmekte ve her bir blok içerisindeki en uzun ardışık birlerin akışına bakılmaktadır. En uzun birlerin beklenen uzunlukta olması, en uzun sıfırların da düzensizliğini ve beklenen uzunlukta olduğunu göstermektedir. Elde edilen sonuçların frekansları beklenen değer frekansları ile karşılaştırılmaktadır. En uzun koşu testi için referans olarak χ^2

dağılımı kullanılmaktadır. Testte n dizi uzunluğuna göre önerilen M blok uzunluğu değerleri Tablo 2.3'te görülmektedir.

Tablo 2.3. Dizi uzunluğuna göre önerilen blok uzunluğu

Minimum n	M
128	8
6272	128
750.000	10^4

Tablo 2.4'te V_i en uzun birlerin akış frekansı olmak üzere çeşitli uzunluktaki blok için en uzun birlerin olması gereken akışının değerleri verilmektedir.

Tablo 2.4. Çeşitli blok değerleri için en uzun birlerin akış frekans değerleri

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

K ve N parametrelerinin belirlenmesi amacıyla M blok uzunlukları referans alınarak Tablo 2.5'te verilen değerler kullanılmaktadır.

Tablo 2.5. Blok uzunluklarına göre belirlenecek K ve N parametre değerleri

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

Burada verilen K ve N parametreleri kullanılarak gözlemlenen dağılım istatistiği $\chi^2(obs)$ Denklem (2.28)'de verildiği gibi hesaplanmaktadır.

$$\chi^2(obs) = \sum_{i=0}^K \frac{(V_i - N\pi_i)^2}{N\pi_i} \quad (2.28)$$

Elde edilen $\chi^2(obs)$ ve N değeri Denklem (2.29)'da yerine konarak P -değeri hesaplanmaktadır.

$$P - value = igamc\left(\frac{K}{2}, \frac{\chi^2(obs)}{2}\right) \quad (2.29)$$

Örneğin dizimiz $\varepsilon = \varepsilon_1, \varepsilon_2, \dots, \varepsilon_{128} = 110011000001010101101100010011001110000$
 $00000001001001101010100010001001111010110100000001101011111001100111$
 001101101100010110010 olsun.

Buradan, $n=128$, $\tau = \frac{2}{\sqrt{128}} = 0.02$ olarak hesaplanır ve $\pi=0.42$ değeri elde edilir. Tablo 2.5'ten yararlanılarak $N=128/8=16$ olarak bulunur. Buna göre $M=8$ ve $N=16$ için $K=3$ değeri bulunmaktadır. Sonuç olarak $n=128$ bitlik dizi $M=8$ bitlik bloklara ayrılarak bloklardaki en uzun birlerin frekansı Tablo 2.6'da incelenmektedir.

Tablo 2.6. Bloklardaki en uzun birlerin frekansının incelenmesi

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

Tablo 2.5'ten yararlanılarak $K=3$ olduğundan V_i değişkeni $0 \leq i \leq 3$ aralığında değişecektir. Buradan V_0 bloklardaki ardışık sadece bir tane gelen birlerin akışı sayısını, V_1 bloklardaki ardışık gelen iki tane birlerin akışı sayısını, V_2 bloklardaki ardışık gelen üç tane uzun birlerin akışı sayısını göstermek üzere; $V_0=4$, $V_1=9$, $V_2=3$, $V_3=0$ olarak bulunmaktadır.

Buradan $\pi_0=0.2148$, $\pi_1=0.3672$, $\pi_2=0.2305$, $\pi_3=0.1875$ değerleri elde edilir. Bulunan değerler $\chi^2(obs)$ denkleminde yerine konulursa;

$$\chi^2(obs) = \frac{(4 - 16 \cdot 0.2148)^2}{16 \cdot 0.2148} + \frac{(9 - 16 \cdot 0.3672)^2}{16 \cdot 0.3672} + \frac{(3 - 16 \cdot 0.2305)^2}{16 \cdot 0.2305} + \frac{(0 - 16 \cdot 0.1875)^2}{16 \cdot 0.1875} = 4.882 \text{ olarak hesaplanmaktadır.}$$

Buna göre elde edilen değerler P -değeri için yerine konulursa;

$$P - \text{value} = igamc \left(\frac{K}{2}, \frac{\chi^2(obs)}{2} \right) = igamc \left(\frac{3}{2}, \frac{4.882}{2} \right) = 0,180$$

değeri elde edilmektedir. Sonuç olarak P -değeri $=0.180 \geq 0.001$ olduğundan dizi rasgele olarak kabul edilmektedir [71, 115].

5. İkili Matris Derece Testi (The Binary Matrix Rank Test): Bu testte, n bit uzunluklu diziler M bit uzunluğundaki bloklara bölünür ve bu bloklar bir satırı belirtecek şekilde kullanılarak bir matris oluşturulur. Bu matrisin derecesi hesaplanarak bloklar arasındaki doğrusal bağımlılığın olup olmadığı incelenmektedir. Bu test için M satır sayısı ve Q sütun sayısı olmak üzere $M=Q=32$ olarak sabitlenmektedir. Test istatistiği referans dağılımı olarak χ^2 dağılımı kullanılmaktadır. Meydana gelecek matris sayısı $N=|n/MQ|$ şeklinde hesaplanmaktadır. Oluşturulan matrislerden kalan bit sayıları ihmal edilmektedir.

Örneğin dizimiz $n=20$ olmak üzere $\varepsilon=\varepsilon_1, \varepsilon_2, \dots, \varepsilon_{20}=11010110100011100110$ olsun. $M=Q=3$ ve $N=\lfloor 20/3 \rfloor=2$ olarak bulunur. Bir matris için kullanılacak bit sayısı $M \cdot Q=3 \cdot 3=9$ bit olur. Test içerisinde 2 matris oluşturulacağından $2 \cdot 9=18$ bit kullanılmaktadır. Buradan $20-18=2$ bit göz ardı edilmektedir. Sonuç olarak aşağıdaki matrisler oluşacaktır.

$$N_1 = \begin{vmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{vmatrix} \text{ ve } N_2 = \begin{vmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{vmatrix} \text{ olmaktadır. Elde edilen değerler kullanılarak}$$

P_m, P_{m-1} ve P_{m-2} değerleri aşağıdaki gibi hesaplanmaktadır.

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

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

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

Ki-kare dağılımının hesaplanabilmesi için Denklem (2.30) kullanılmaktadır.

$$\begin{aligned} \chi^2(\text{obs}) &= \frac{(F_M - P_m \cdot N)^2}{P_m N} + \frac{(F_{M-1} - P_{m-1} \cdot N)^2}{P_{m-1} N} + \\ &+ \frac{(N - F_M - F_{M-1} - P_{m-2} N)^2}{P_{m-2} N} \end{aligned} \quad (2.30)$$

Bulunan değerler Denklem (2.30) kullanılarak yerine konulursa;

$$\begin{aligned} \chi^2(\text{obs}) &= \frac{(1 - 0.2888 \cdot 2)^2}{0.2888 \cdot 2} + \frac{(1 - 0.5776 \cdot 2)^2}{0.5776 \cdot 2} + \\ &+ \frac{(2 - 1 - 1 - 0.1336 \cdot 2)^2}{0.1336 \cdot 2} = 0.597 \end{aligned}$$

olarak hesaplanmaktadır. Buradan $P - value = e^{-\chi^2(obs)/2} = e^{-0.5969/2} = 0.742$ değeri elde edilmektedir. Sonuç olarak $P-değeri=0.742 \geq 0.001$ olduğundan dizi rasgele olarak kabul edilmektedir [71, 115].

6. Ayrık Fourier Dönüşüm Testi (The Discrete Fourier Transform (Spectral) Test): Literatürde spektral test olarak da isimlendirilen bu test, dizinin tepe yüksekliklerine odaklanmaktadır. Bu testte amaç, dizinin periyodikliğinin incelenmesidir. Bu amaçla d gözlemlenen ve beklenen %95 eşik değerinin üstündeki frekans bileşenlerinin gözlemlenen ve beklenen sayıları arasındaki standart farklılığı ve $x_i=2\varepsilon-1$ göstermek üzere önce $X=x_1, x_2, \dots, x_n$ dönüşümü yapılmaktadır. Örneğin, dizimiz $n=10$ olmak üzere $\varepsilon=\varepsilon_1, \varepsilon_2, \dots, \varepsilon_{10}=1001010011$ olsun. Buradan dönüşüm sonucunda $X=1, -1, -1, 1, -1, 1, -1, -1, 1, 1$ elde edilir. Elde edilen X dizisine ayrık Fourier dönüşümü (AFD) uygulanarak $S=AFD(X)$ hesaplanır. Burada S aşağıda Denklem (2.31)'de verildiği gibi hesaplanmaktadır.

$$S_j = \sum_{k=1}^n x_k e^{(2\pi_i(k-1)j/n)} \quad (2.31)$$

Burada, $j=0, 1, \dots, n-1$ ve $i = \sqrt{-1}$ olmak üzere $e^{(2\pi_i(k-1)j/n)} = \cos(2\pi_i(k-1)j/n) + i \sin(2\pi_i(k-1)j/n)$ olmaktadır. Bir sonraki adımda, $M=modulus(S_j')$ fonksiyonu kullanılarak dizinin tepe yükseklikleri hesaplanmaktadır. Burada S' , S dizisindeki ilk $n/2$ elemandan oluşan bir alt diziyi ifade etmektedir. Ardından tepe yüksekliği eşik değeri olan $T = \sqrt{3n} = \%95$ değerini aşmamalıdır. T değerinden daha küçük tepe yüksekliklerinin beklenen teorik değeri olan $N_0=0.95*n/2=0.95*10/2=4.75$ olarak hesaplanmaktadır. Bu örnek için T 'den daha az gerçek gözlemlenen tepe sayısı $N_1=4$ olarak bulunmaktadır. Buradan test istatistiğinin hesaplanabilmesi amacıyla $d = (N_1 - N_0) / \sqrt{n(0.95)(0.05)/4}$ ifadesi kullanılarak $d = (4.75 - 4) / \sqrt{10(0.95)(0.05)/4} = -2.176$ olarak hesaplanmaktadır. Buradan, $P - value = erfc(|d|/\sqrt{2}) = erfc(2.176/\sqrt{2}) = 0.029$ elde edilmektedir. Sonuç olarak, $P-değeri=0.029 \geq 0.001$ olduğundan dizi rasgele olarak kabul edilmektedir [71, 115].

7. Örtüşmeyen Şablon Eşleştirme Testi (The Non-overlapping Template Matching Test): Bu testte, rasgele sayı üreticinin ürettiği n bitlik dizideki m bitlik blokların içerisinde, periyodik olmayan önceden belirlenmiş örnek dizinin bulunma sıklığının tespit edilmesi ve incelenmesi amaçlanmaktadır. Seçilen özel blokların tekrar edilmesi durumunda, gözlemlenen bloktan sonraki ilk bitten aramaya devam edilmektedir. Eğer belirlenen m bitlik özel bloklar bulunmaz ise pencere bir bit kaydırılarak arama işlemine devam edilir.

Örneğin $\varepsilon=10100100101110010110$ olsun. Buradan B istatistiksel testin içerisinde bulunan periyodik olmayan şablon örnekleri, M test edilecek alt dizilerin bit uzunluğu ve N bağımsız blokların sayısını göstermek üzere $n=20$, $N=2$ ve $M=n/N=20/2=10$ olarak hesaplanmaktadır. Sonuç olarak $M_1=1010010010$ ve $M_2=1110010110$ şeklinde iki blok elde edilmektedir. $j=1,2,\dots,N$ olmak üzere W_j blok içerisinde m bitlik özel B bloğunun kaç defa bulunduğunu ifade etmektedir.

Tablo 2.7’de $m=3$ ve $B=001$ örnek şablonu için B şablonunun M_1 ve M_2 blokları içerisinde kaç defa bulunduğunu göstermektedir. Burada verilen rasgele ε dizisi M_1 ve M_2 blokları içerisinde B şablonundan kaç tane olduğu araştırılmaktadır. Örneğin M_1 bloğu içerisinde üç bitlik B şablonundan araştırıldığında bit pozisyonları 1-3 için “101”, 2-4 için “010”, 3-5 için “100” olduğundan $W_1=0$ olmaktadır. Ancak bit pozisyonu 4-6 için “001”, araştırılan B şablonu ile aynı olduğundan W_1 değeri 1 arttırılarak $W_1=1$ olmaktadır. Bundan sonra şablon 3 bit olduğu için 5-7 ve 6-8 bit pozisyonları atlanarak 7-9 pozisyonundan araştırmaya devam edilmektedir. 7-9 için “001”, araştırılan B şablonu ile aynı olduğundan W_1 değeri 1 arttırılarak $W_1=2$ olmaktadır. Ardından şablon 3 bit olduğu için 8-10 bit pozisyonları atlanarak M_1 bloğu için işlem tamamlanmakta ve en son $W_1=2$ olmaktadır. Aynı işlemler M_2 bloğu için de tekrarlanmaktadır.

Tablo 2.7. $m=3$ için M_1 ve M_2 blokları içerisinde $B=001$ şablonunun incelenmesi

Bit Pozisyonları	M_1 Bloğu		M_2 Bloğu	
	Bitler	W_1	Bitler	W_2
1-3	101	0	111	0
2-4	010	0	110	0
3-5	100	0	100	0
4-6	001 (Bulundu)	Arttır 1	001 (Bulundu)	Arttır 1
5-7	Test Edilmedi	1	Test Edilmedi	1
6-8	Test Edilmedi	1	Test Edilmedi	1
7-9	001	Arttır 2	011	1
8-10	010 (Bulundu)	2	110	1

Tablo 2.7'den $W_1=2$ ve $W_2=1$ olarak elde edilmektedir. Buradan, rasgelelik testi için ortalama değeri $\mu = (M - m + 1) / 2^m = (10 - 3 + 1) / 2^3 = 1.00$ ve varyans değeri $\sigma^2 = M(1/2^m - (2m - 1/2^{2m})) = 10(1/2^3 - (2 \cdot 3 - 1/2^{2 \cdot 3})) = 0.468$ olarak hesaplanmaktadır. Buna göre χ^2 dağılımı;

$$\chi^2(\text{obs}) = \sum_{j=1}^N \frac{(W_j - \mu)^2}{\sigma^2} = \frac{(2 - 1)^2 + (1 - 1)^2}{0.468} = 2.133$$

olarak elde edildikten sonra P -değeri aşağıdaki şekilde hesaplanmaktadır.

$$P - \text{value} = \text{igamc} \left(\frac{N}{2}, \frac{\chi^2(\text{obs})}{2} \right) = \text{igamc} \left(\frac{2}{2}, \frac{2.133}{2} \right) = 0.344$$

Sonuç olarak P -değeri $= 0.344 \geq 0.001$ olduğundan dizi rasgele olarak kabul edilmektedir [71, 115].

8. Örtüşen Şablon Eşleştirme Testi (The Overlapping Template Matching Test): Bu testte amaç üretilen n bitlik dizi içerisindeki m bitlik blokların içerisinde, periyodik olmayan önceden belirlenmiş örnek dizinin bulunma sıklığının tespit edilmesi ve incelenmesidir. Bu testin örtüşmeyen şablon eşleştirme testinden farkı, bu testte eğer önceden belirlenen şablon tespit edilmiş ise arama işlemine bir bit sonra devam

edilmesidir. Eğer belirlenen m bitlik özel bloklar bulunmaz ise pencere bir bit kaydırılarak arama işlemine devam edilir.

Örneğin

$\varepsilon = \varepsilon_1, \varepsilon_2, \dots, \varepsilon_{50} = 10111011110010110100011100101110111110000101101001$ olsun. Buradan, $n=50$, N bağımsız blok sayısını (bu test için $N=5$ olarak alınmıştır), K bağımsızlık katsayısı (bu test için $K=2$ olarak alınmıştır), M test edilecek olan ε bitlerinin uzunluğunu (bu test için $M=10$ olarak alınmıştır) ifade etmek üzere üretilen sayılar 5 farklı bloğa bölünerek, $M_1=1011101111$, $M_2=0010110100$, $M_3=0111001011$, $M_4=1011111000$, $M_5=0101101001$ blokları oluşturulur. Testte m şablon bit uzunluğu ve B eşleştirilecek m bitlik şablonu göstermek üzere $m=2$ ve $B=11$ olarak alınırsa örnek olarak M_1 bloğu için B şablonunun bulunma durumları Tablo 2.8'de verildiği gibi olmaktadır.

Tablo 2.8. M_1 bloğu içerisinde $B=11$ özel şablonunun bulunma durumları

Bit Pozisyonları	M_1 Bloğu	
	Bitler	V_i
1-2	10	0
2-3	01	0
3-4	11 (Bulundu)	Arttır 1
4-5	11 (Bulundu)	Arttır 2
5-6	10	2
6-7	01	2
7-8	11 (Bulundu)	Arttır 3
8-9	11 (Bulundu)	Arttır 4
9-10	11 (Bulundu)	Arttır 5

Bu aşamadan sonra λ değeri $\lambda = (M - m + 1) / 2^m = (10 - 2 + 1) / 2^2 = 2.25$ ve η değeri $\eta = \lambda / 2 = 2.25 / 2 = 1.125$ olarak hesaplanmaktadır.

Buradan, π_i için $\pi_1=0.3246$, $\pi_2=0.1826$, $\pi_3=0.1426$, $\pi_4=0.1066$, $\pi_5=0.0771$, $\pi_6=0.1662$ olarak elde edilmektedir. $\chi^2(obs)$ dağılımı için hesaplanan değerler yerine konulursa;

$$\chi^2(\text{obs}) = \sum_{j=1}^N \frac{(v_j - N\pi_j)^2}{N\pi_j} = \frac{(0 - 5 \cdot 0.3246)^2}{5 \cdot 0.3246} + \frac{(1 - 5 \cdot 0.1826)^2}{5 \cdot 0.1826} + \frac{(1 - 5 \cdot 0.1426)^2}{5 \cdot 0.1426} + \frac{(1 - 5 \cdot 0.1066)^2}{5 \cdot 0.1066} + \frac{(1 - 5 \cdot 0.0771)^2}{5 \cdot 0.0771} + \frac{(1 - 5 \cdot 0.1662)^2}{5 \cdot 0.1662} = 3.1667$$

olarak bulunmaktadır. Elde edilen Ki-kare dağılımı ve N değerleri kullanılarak P -değeri hesaplanırsa;

$$P - \text{value} = \text{igamc} \left(\frac{N}{2}, \frac{\chi^2}{2} \right) = \text{igamc} \left(\frac{5}{2}, \frac{3.1667}{2} \right) = 0.274$$

olarak bulunmaktadır. Sonuç olarak $P\text{-değeri}=0.274 \geq 0.001$ olduğundan dizi rasgele kabul edilmektedir [71, 115].

9. Maurer “Evrensel İstatistik” Testi (Maurer’s “Universal Statistical” Test): Bu test 1992 yılında Princeton Üniversitesi bilgisayar bilimleri bölümünde bulunan U. Maurer tarafından geliştirilmiştir. Test rasgele dizinin veri kaybı olmadan sıkıştırılabilirliğine odaklanmaktadır. Ayrıca sunulan çalışmada bu testin kriptografik uygulamada gizli anahtar kaynağı için bir kalite ölçütü olduğu belirtilmektedir [118]. Ünlüml testte L her bir bloğun uzunluğu, Q başlangıç bölümü ve $K=[n/L]-Q$ olmak üzere test bölümünü ifade etmektedir. $Q \times L$ -bit ve $K \times L$ -bit değerlerinin kalan bitleri atılmakta ve testte kullanılmamaktadır.

Örneğin dizimiz $n=20$ olmak üzere $\varepsilon = \varepsilon_1, \varepsilon_2, \dots, \varepsilon_{20} = 01011010011101010111$ olsun. Buradan eğer $L=2$ ve $Q=4$ olarak alınırsa o zaman $K=[20/2]-4=6$ değeri elde edilmektedir. Başlangıç bloğu $Q=01011010$ ve test bloğu $K=011101010111$ olmaktadır. Tablo 2.9’da Maurer testinde L -bit uzunluğundaki blokların bölümleri verilmektedir.

Tablo 2.9. Maurer testi L -bit uzunluğundaki blokların bölümleri

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

Aşağıda Tablo 2.10’da başlangıç bölümü için muhtemel L -bit değerleri verilmektedir. Burada J bölümlerdeki i . L -bit bloğun onluk sayı sistemindeki değeri olmak üzere $i=1$ ’den Q değerine kadar $T_j=i$ olarak tabloya kaydedilmektedir. Ayrıca başlangıç blok değerleri, blok içerisindeki muhtemel L -bit ondalık değerleri ve başlangıç bloğunda bulunma sayısı ile çarpılarak oluşturulmaktadır.

Tablo 2.10. Dört başlangıç değeri ile oluşturulan muhtemel L -bit değerleri

Başlangıç	Muhtemel L -bit Değerleri			
	00 (T_0 'a kaydedilmiş)	01 (T_1 'a kaydedilmiş)	10 (T_2 'a kaydedilmiş)	11 (T_3 'a kaydedilmiş)
	0	2	4	0

Tablo 2.11’de başlangıç bölümü kullanılarak (başlangıç bölümünün aldığı değerler 4 nolu satırda gösterilmektedir) test bölümündeki L -bit bloklarının aldığı değerler verilmektedir. Burada test bloğunun her L -bit değerleri, başlangıç bloğuyla örtüştüğünde test bloğu tekrar blok numarası değerini almaktadır. Örneğin 5 nolu tekrar adımında, test bölümünde “01” değerleri bulunmaktadır. Başlangıç bölümünde “01” değerleri T_2 ’de kayıtlı olduğundan, 5 nolu tekrar bloğunun yeni değeri 5 olmaktadır. Sonraki satırda (6 nolu tekrar bloğu) test bölümünde “11” değerleri bulunmaktadır. Bu nedenle muhtemel L değerlerinde “11” değerleri T_4 ’te kayıtlı olduğundan buraya tekrar bloğunun değeri olan 6 değeri atanmaktadır.

Tablo 2.11. Test bölümü için L -bit değerleri

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

Test istatistiği Denklem (2.32)'deki gibi hesaplanmaktadır.

$$f_n = \frac{1}{K} \sum_{i=Q+1}^{Q+K} \log(i - T_j) = \frac{1}{6} \sum_{i=4+1}^{10} \log(i - T_j) = \frac{1}{6} \sum_{i=4+1}^{10} \begin{pmatrix} \log(5 - 2) + \\ \log(6 - 0) + \\ \log(7 - 5) + \\ \log(8 - 7) + \\ \log(9 - 8) + \\ \log(10 - 6) \end{pmatrix} = \quad (2.32)$$

$$\frac{1}{6} \sum_{i=4+1}^{10} (1.5849 + 4.1699 + 5.1699 + 5.1699 + 5.1699 + 7.1699) = 1.1949$$

Fonksiyonda beklenen değer $V_{exp}(L)$ ve varyans $var(f_n)$ değerleri Tablo 2.12'de verilmektedir.

Tablo 2.12. L değerleri için $V_{exp}(L)$ ve $var(f_n)$ değerleri

L	$V_{exp}(L)$	$Var(f_n)$
6	5,2177052	2,954
7	6,1962507	3,125
8	7,1836656	3,238
9	8,1764248	3,311
10	9,1723243	3,356
11	10,1700320	3,384
12	11,1687650	3,401
13	12,1680700	3,410
14	13,1676930	3,416
15	14,1674880	3,419
16	15,1673790	3,421

Buradan elde edilen değerler $erfc$ fonksiyonunda yerine konulursa P -değeri

$$P - value = erfc \left(\left\| \frac{f_n - V_{exp}(L)}{\sqrt{2 \text{var}(f_n)}} \right\| \right) = erfc \left(\left\| \frac{1.1949 - 1.5374}{\sqrt{21.338}} \right\| \right) = 0.767$$

olarak elde edilmektedir. Sonuç olarak P -değeri $=0.767 \geq 0.001$ olduğundan dizi rasgele olarak kabul edilmektedir [71, 115].

10. Doğrusal Karmaşıklık Testi (The Linear Complexity Test): Bu testte, rasgele bit dizisinin Doğrusal Geri beslemeli Kayan Kaydedici (DGKK) (Linear Feedback Shift Register) (LFRS) uzunluğuna bakılarak dizinin kompleksliğinin incelenmesi amaçlanmaktadır. Dizi içerisinde DGKK uzunluğunun yüksek olması dizinin daha rasgele olduğunu göstermektedir.

Örneğin $\varepsilon = \varepsilon_1, \varepsilon_2, \dots, \varepsilon_{13} = 1101011110001$ olmak üzere bit dizisi verilmektedir. Blok içerisindeki bit uzunluğu $M=13$, N ise M bit blokların sayısını belirtmek üzere $n=M*N$ ifadesiyle belirlenmektedir. Burada $N=1$ olmaktadır.

N katsayısı, blokların her birinin doğrusal karmaşıklığı polinomun en düşük derecesini hesaplayan Berlekamp-Massey algoritması kullanılarak belirlenmektedir [119]. L_i , $i=1 \dots N$ kadar i . blokta üretilen tüm bit dizilerinin en kısa DGKK dizisini belirtmektedir. Bu örnekteki dizi için $L_i=4$ olmaktadır. Bir sonraki bitten itibaren işlem yenilenmektedir.

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

$$\frac{13}{2} + \frac{(9 + (-1)^{13+1})}{36} - \frac{(13/3 + 2/9)}{2^{13}} = 6.7772$$

Buradan T_i dağılımın rasgele değişkeni olmak üzere değerler yerine konulursa; $T_i = (-1)^M \cdot (L_i - \mu) + 2/9 = (-1)^{13} \cdot (4 - 6.777) + 2/9 = 2.999$ olarak hesaplanmaktadır. π_i değerleri önceden hesaplanarak $\pi_0=0.0104$, $\pi_1=0.03125$, $\pi_2=0.125$, $\pi_3=0.5$, $\pi_4=0.25$, $\pi_5=0.0625$ ve $\pi_6=0.020833$ olarak elde edilmektedir. Ki-kare testi için hesaplanan değerler denklemde yerine konulursa;

$$\chi^2(\text{obs}) = \sum_{i=1}^K \frac{(v_i - N\pi_i)^2}{N\pi_i} = 47.0008$$

$$P - \text{value} = \text{igamc} \left(\frac{K}{2}, \frac{\chi^2(\text{obs})}{2} \right) = \text{igamc} \left(\frac{6}{2}, \frac{47.008}{2} \right) = 0.993$$

olarak elde edilmektedir. Sonuç olarak $P\text{-değeri}=0.993 \geq 0.001$ olduğundan dizi rasgele olarak kabul edilmektedir [71, 115].

11. Seri Testi (The Serial Test): Bu testte n bit dizisinin uzunluğu, m her bir bloktaki bitlerin uzunluğu olmak üzere verilen dizideki her m bit örneğin dizideki diğer m bit örnekler ile aynı değişimi ve tekdüzelilik (uniformity) seviyesini incelemektedir. Eğer seri test için $m=1$ olursa frekans testi ile aynı işlevi görmektedir. Bu testte $P\text{-değeri}1$ ve $P\text{-değeri}2$ olmak üzere iki test sonucu elde edilmektedir.

Örneğin $\varepsilon = \varepsilon_1, \varepsilon_2, \dots, \varepsilon_{10} = 0011011101$ olmak üzere bir bit dizisi verilmektedir. Blok içerisindeki bit uzunluğu $n=10$ olmaktadır. ε' dizisi n bitlik dizinin sonuna eklenen $m-1$ bit '0' eklenmesi ile oluşan artırım dizisidir. Örneğin $m=1$ için $\varepsilon' = 0011011101$, $m=2$ için $\varepsilon' = 00110111010$, $m=3$ için $\varepsilon' = 001101110100$ olmaktadır. Dizideki tüm muhtemel örtüşen m , $m-1$ ve $m-2$ bitlik blokların frekansının belirlenebilmesi için; $v_{i_1 \dots i_m}$ m bitlik $i_1 \dots i_m$ örneklerin frekansını, $v_{i_1 \dots i_{m-1}}$ $m-1$ bitlik $i_1 \dots i_{m-1}$ örneklerin frekansını, $v_{i_1 \dots i_{m-2}}$ $m-2$ bitlik $i_1 \dots i_{m-2}$ örneklerin frekansını göstermektedir. Bu bölümdeki örnek için $m=3$ olarak alındığında $m-1=2$ ve $m-2=1$ olmaktadır. Bütün 3-bitlik blokların frekansları $v_{000}=0$, $v_{001}=1$, $v_{010}=1$, $v_{011}=2$, $v_{100}=1$, $v_{101}=2$, $v_{110}=2$, $v_{111}=1$, tüm 2-bitlik blokların frekansları $v_{00}=1$, $v_{01}=3$, $v_{10}=3$, $v_{11}=3$ ve bütün 1-bitlik blokların frekansları $v_0=4$, $v_1=6$ olarak elde edilmektedir. Dizideki $\Delta\Psi_m^2(obs)$, m bit örneğin gözlemlenen frekansının ve $\Delta^2\Psi_m^2(obs)$ ise m bit örneğin beklenen frekansının ne kadar iyi olduğunun bir ölçüsü olmak üzere Ψ_m^2 , Ψ_{m-1}^2 ve Ψ_{m-2}^2 değerleri aşağıdaki denklemler kullanılarak hesaplanmaktadır.

$$\Psi_m^2 = \frac{2^m}{n} \sum_{i_1 \dots i_m} (v_{i_1 \dots i_m}^2 - n) \quad (2.33)$$

$$\Psi_{m-1}^2 = \frac{2^{m-1}}{n} \sum_{i_1 \dots i_{m-1}} (v_{i_1 \dots i_{m-1}}^2 - n) \quad (2.34)$$

$$\Psi_{m-2}^2 = \frac{2^{m-2}}{n} \sum_{i_1 \dots i_{m-2}} (v_{i_1 \dots i_{m-2}}^2 - n) \quad (2.35)$$

Buradan örnek dizideki değerler denklemlerde yerine konulursa;

$$\Psi_m^2 = \frac{2^3}{10} (0 + 1 + 1 + 4 + 1 + 4 + 4 + 1) - 10 = 2.8$$

$$\Psi_{m-1}^2 = \frac{2^{3-1}}{10} (1 + 9 + 9 + 9) - 10 = 1.2$$

$$\Psi_{m-2}^2 = \frac{2^{3-2}}{10} (16 + 36) - 10 = 0.4$$

olarak bulunmaktadır. Rasgelelik testi için genelleştirilmiş seri istatistiklerinin cevabı $\Delta\Psi_m^2 = \Psi_m^2 - \Psi_{m-1}^2$ ve $\Delta^2\Psi_m^2 = \Psi_m^2 - 2\Psi_{m-1}^2 + \Psi_{m-2}^2$ olmaktadır. Buradan hesaplanan frekans değerleri yerine konularak $\Delta\Psi_m^2 = 2.8 - 1.2 = 1.6$ ve $\Delta^2\Psi_m^2 = 2.8 - 2 \cdot 1.2 + 0.4 = 0.8$ olarak hesaplanmaktadır. Test istatistiği referans χ^2 dağılımı $P - value 1 = igamc(2^{m-2}, \nabla\Psi_m^2 / 2)$ ve $P - value 2 = igamc(2^{m-3}, \nabla^2\Psi_m^2 / 2)$ denklemleri kullanılarak

$$P - value 1 = igamc\left(2^{3-2}, \frac{1.6}{2}\right) = 0.905$$

$$P - value 2 = igamc\left(2^{3-3}, \frac{0.8}{2}\right) = 0.880$$

elde edilmektedir. Buradan sonuç olarak $P - de\tilde{g}eri 1 = 0.905 \geq 0.001$ ve $P - de\tilde{g}eri 2 = 0.880 \geq 0.001$ olduğundan dizi rasgele olarak kabul edilmektedir [71, 115].

12. Yaklaşık Entropi Testi (The Aproximate Entropy Test): Bu testte amaç, seri testinde olduğu gibi tüm muhtemel örtüşen m bitlik örnek dizinin frekansının incelenmesidir. Test, rasgele bir dizi için beklenen frekansın, iki ardışık veya bitişik uzunluktaki (m ve $m+1$) örtüşen blokların frekanslarını karşılaştırmaktadır.

Örneğin $\varepsilon = \varepsilon_1, \varepsilon_2, \dots, \varepsilon_{10} = 0100110101$ olmak üzere bir bit dizisi verilmektedir. Blok içerisindeki bit uzunluğu $n=10$ ve $m=3$ olmaktadır. Dizinin sonuna dizinin başından $m-1$ bit eklenmektedir. Buna göre $m-1=2$ olduğundan yeni dizimiz $\varepsilon = 010011010101$ olmaktadır. Ardından $m=3$ olduğundan elde edilen dizi ardışık olarak önce m sonra $m+1$ bitlik bloklara bölünmektedir. Elde edilen bloklar 3 bit için $010, 100, 001, 011, 110, 101, 010, 101, 010, 101$ ve 4 bit için $0100, 1001, 0011, 0110, 1101, 1010, 0101, 1010, 0101$, ve 1010 olarak bulunur. $2^m = 2^3 = 8$ olduğundan 3 bit için 8 muhtemel durum ve $2^m = 2^4 = 16$ olduğundan 4 bit için 16 muhtemel durum bulunmaktadır. Bu durumların dizide 3 bit için bulunma frekansları $\#000=0, \#001=1, \#010=3, \#011=1, \#100=1, \#101=3, \#110=1, \#111=0$ ve 4 bit için bulunma frekansları $\#0011=1, \#0100=1, \#0101=2, \#0110=1, \#1001=1, \#1010=3, \#1101=1$ ve diğer tüm durumlar sıfır olmaktadır. Bir sonraki aşamada i, m bit blokların değerleri ve C_i^m muhtemel m

bit değerlerin sayısı olmak üzere $C_i^m = \#i/n$ eşitliği kullanılarak 3 bit her bir i değeri için $C^3_{000}=0$, $C^3_{001}=0.1$, $C^3_{010}=0.3$, $C^3_{011}=0.1$, $C^3_{100}=0.1$, $C^3_{101}=0.3$, $C^3_{110}=0.1$, $C^3_{111}=0$, ve 4 bit her bir i değeri için $C^4_{0011}=C^4_{0100}=C^4_{0110}=C^4_{1001}=C^4_{1101}=0.1$, $C^4_{0101}=0.2$, $C^4_{1010}=0.3$ ve diğer tüm değerler sıfır olarak hesaplanmaktadır. Buradan, $\pi_i = C_j^m$ ve $j = \log_2 i$ olmak üzere m bit uzunluktaki bütün 2^m muhtemel blokların dizi üzerindeki ampirik dağılımın frekansı $\varphi^{(m)}$ Denklem (2.36) kullanılarak hesaplanmaktadır.

$$\varphi^{(m)} = \sum_{i=0}^{2^m-1} \pi_i \log \pi_i \quad (2.36)$$

Elde edilen değerler yerine konulursa; $\varphi^{(3)} = 0(\log 0) + 0.1(\log 0.1) + 0.3(\log 0.3) + 0.1(\log 0.1) + 0.1(\log 0.1) + 0.3(\log 0.3) + 0.1(\log 0.1) + 0(\log 0) = -1.6434$ ve $\varphi^{(4)} = 0+0+0+0.1(\log 0.01) + 0.1(\log 0.01) + 0.2(\log 0.02) + 0.1(\log 0.01) + 0+0+0.1(\log 0.01) + 0.3(\log 0.03) + 0+0+0.1(\log 0.01) + 0+0 = -1.8343$ olarak elde edilmektedir. Buradan, dizinin Ki-kare test istatistiği için $ApEn(m) = \varphi^{(m)} - \varphi^{(m+1)}$ olmak üzere $\chi^2 = 2n[\log 2 - ApEn(m)]$ eşitliği kullanılarak $ApEn(3) = -1.6434 - (-1.8343) = 0.1909$ ve $\chi^2 = 2*10(0.6931 - 0.1909) = 0.5021$ olarak bulunmaktadır.

$$P - value = igamc\left(2^{m-1}, \frac{\chi^2}{2}\right) = igamc\left(2^{3-1}, \frac{0.5021}{2}\right) = 0.261 \quad (2.37)$$

Buradan P -değeri $= 0.261 \geq 0,001$ olduğundan dizi rasgele olarak kabul edilmektedir [71, 115].

13. Birikimli Toplamlar Testi (The Cumulative Sums Test): Bu testte amaç, rasgele bir dizi için kümülâtif toplamın beklenen davranışı için kısmi alt blokların kümülâtif toplamının çok büyük veya çok küçük olup olmadığının belirlenmesidir. Bu amaçla dizi öncelikli olarak $X_i = 2\varepsilon_i - 1$ dönüşümü kullanılarak giriş dizisi -1 ve $+1$ değerlerine normalize edilmekte ve yeni X_i dizisi elde edilmektedir. Rasgele bir dizide sonuçların sıfıra yakın çıkması beklenmektedir. Birikimli toplamlar testi için örneğin $\varepsilon = \varepsilon_1, \varepsilon_2, \dots, \varepsilon_{10} = 1011010111$ olmak üzere bir bit dizisi verilmektedir. Buradan

normalize işlemi yapıldığında elde edilen yeni dizi $X=1, -1, 1, 1, -1, 1, -1, 1, 1, 1$ olmaktadır. Bu testin uygulanmasında ileri ve geri yönlü olmak üzere iki farklı metot kullanılmaktadır. Test aşamasında 0 seçilirse test ileri yönlü ve 1 seçilirse test geri yönlü çalışmakta ve bu metotların çalışması Tablo 2.13'te verilmektedir.

Tablo 2.13. Test için ileri ve geri yönlü metotların uygulanması

Metot1=0 (İleri yönlü)	Metot2=1 (Geri yönlü)
$S_1=X_1$	$S_1=X_n$
$S_2=X_1+X_2$	$S_2=X_n+X_{n-1}$
$S_3=X_1+X_2+X_3$	$S_3=X_n+X_{n-1}+X_{n-2}$
:	:
$S_k=X_1+X_2+X_3+\dots+X_k$	$S_k=X_n+X_{n-1}+X_{n-2}+\dots+X_{n-k+1}$
:	:
$S_n=X_1+X_2+X_3+\dots+X_k+\dots+X_n$	$S_n=X_n+X_{n-1}+X_{n-2}+\dots+X_{n-k+1}+\dots+X_1$

Bu test için verilen örneğe göre uygulama metodu için 1 seçilerek ileri yönlü metot uygulanırsa aşağıdaki değerler elde edilmektedir.

$$S_1=1$$

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

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

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

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

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

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

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

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

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

Buradan $\max_{1 \leq k \leq n} |S_k|$ değeri, S_k kısmi toplamının mutlak değerinin en büyüğü ve z test istatistiği olmak üzere $z=\max_{1 \leq k \leq n} |4|=4$ olmaktadır. P -değerinin hesaplanabilmesi için Φ standart normal kümülâtif olasılık dağılım fonksiyonu olmak üzere Denklem (2.38) kullanılmaktadır.

$$\begin{aligned}
P - value = 1 - & \sum_{k=\left(\frac{-n}{z}+1\right)/4}^{\left(\frac{n-1}{z}\right)/4} \left[\Phi\left(\frac{(4k+1)z}{\sqrt{n}}\right) - \Phi\left(\frac{(4k-1)z}{\sqrt{n}}\right) \right] + \\
& \sum_{k=\left(\frac{-n}{z}-3\right)/4}^{\left(\frac{n-1}{z}\right)/4} \left[\Phi\left(\frac{(4k+3)z}{\sqrt{n}}\right) - \Phi\left(\frac{(4k+1)z}{\sqrt{n}}\right) \right]
\end{aligned} \tag{2.38}$$

Buradan elde edilen değerler yerine konulursa P -değeri=0.411 olarak hesaplanmaktadır. P -değeri=0.411 \geq 0,001 olduğundan dizi rasgele olarak kabul edilmektedir [71, 115].

14. Rasgele Gezinimler Testi (The Random Excursions Test): Bu testte amaç, birikimli toplam rasgele yürüyüşünde K adet döngünün sayısının belirlenmesidir. Birikimli toplam rasgele yürüyüşü, '0' ve '1' değerlerinden oluşan rasgele dizinin $X_i=2\varepsilon_i-1$ dönüşümü ile uygun -1 ve +1 normalize edilmiş dizisi elde edildikten sonra kısmi toplamlarının hesaplanması ile elde edilmektedir. Bu test -4, -3, -2, -1 ve +1, +2, +3, +4 olmak üzere sekiz P -değerinin hesaplandığı serisel bir testtir.

Örneğin $\varepsilon = \varepsilon_1, \varepsilon_2, \dots, \varepsilon_{10} = 0110110101$ olmak üzere bir bit dizisi verilmektedir. Buradan $n=10$ ve $X_i=2\varepsilon_i-1$ dönüşümü kullanılarak giriş dizisi -1 ve +1 değerlerine normalize edilmektedir. Buradan yeni X_i dizisi $X = -1, 1, 1, -1, 1, 1, -1, 1, -1, 1$ olmaktadır. Bu test için verilen örneğe göre birikimli toplamlar testinde verildiği gibi uygulama metodu için 1 seçilerek ileri yönlü metot uygulanırsa S_i kısmi toplam olmak üzere $S_1 = -1, S_2 = 0, S_3 = 1, S_4 = 0, S_5 = 1, S_6 = 2, S_7 = 1, S_8 = 2, S_9 = 1, S_{10} = 2$ değerleri elde edilmektedir. Buradan $S = \{-1, 0, 1, 0, 1, 2, 1, 2, 1, 2\}$ olmaktadır. S' kümesi S kümesinin başına ve sonuna 0 elemanlarının eklenmesi ile oluşturulmaktadır. Sonuç olarak $S' = \{0, -1, 0, 1, 0, 1, 2, 1, 2, 1, 2, 0\}$ olarak elde edilmektedir. Rasgele dizideki döngü sayısını ifade eden J değeri, S' kümesindeki ilk sıfır elemanı hariç kümedeki sıfır elemanlarının sayısıdır. Buna göre S' kümesinde kümenin başlangıcındaki 0 hariç 3 tane 0 bulunduğundan dolayı 3 tane $J_1 = \{0, -1, 0\}, J_2 = \{0, 1, 0\}$ ve $J_3 = \{0, 1, 2, 1, 2, 1, 2, 0\}$ döngüsü bulunmaktadır. Elde edilen döngüler kullanılarak x durum değerlerinin frekansları Tablo 2.14'te verildiği gibi hesaplanmaktadır. Her bir döngü için ve her bir sıfır olmayan durum değeri için x $-4 \leq x \leq -1$ ve $4 \leq x \leq 1$ değerlerini

alabilmekte ve her bir döngü içerisinde her bir x değerinin frekansı hesaplanmaktadır.

Tablo 2.14. Verilen ε dizisi için oluşan rasgele gezinti döngü frekansları

Durum x	Döngüler (J)		
	Döngü 1 (J_1)	Döngü 2 (J_2)	Döngü 3 (J_3)
-4	0	0	0
-3	0	0	0
-2	0	0	0
-1	1	0	0
1	0	1	3
2	0	0	3
3	0	0	0
4	0	0	0

Sonraki aşamada $k=0,1,\dots,5$ ($k=5$ ve tüm frekanslar ≥ 5 $v_5(x)$ 'e kaydedilmektedir) olmak üzere x değerinin 8 farklı durumu için bütün döngüler arasında k defa meydana gelen x durumundaki döngünün toplam sayısı $v_k(x)$ hesaplanmaktadır.

Dikkat edilirse $\sum_{k=0}^5 v_k(x) = J$ olduğu görülmektedir. Verilen örneğe göre;

$V_0(-1)=2$ ($x=-1$ durumu 2 döngüde de 0 kez meydana gelmektedir.)

$V_1(-1)=1$ ($x=-1$ durumu 1 döngüde de 1 kez meydana gelmektedir.)

$V_2(-1)=V_3(-1)=V_4(-1)=V_5(-1)=0$ ($x=-1$ durumu 0 döngüde de $\{2, 3, 4, \geq 5\}$ kez meydana gelmektedir.)

$V_0(1)=1$ ($x=1$ durumu 1 döngüde de 0 kez meydana gelmektedir.)

$V_1(1)=2$ ($x=1$ durumu 1 döngüde de 1 kez meydana gelmektedir.)

$V_3(1)=2$ ($x=1$ durumu 1 döngüde de 3 kez meydana gelmektedir.)

$V_2(1)=V_4(1)=V_5(1)=0$ ($x=1$ durumu 0 döngüde $\{2, 4, \geq 5\}$ kez meydana gelmektedir.)

$V_0(2)=2$ ($x=2$ durumu 2 döngüde de 0 kez meydana gelmektedir.)

$V_3(2)=1$ ($x=2$ durumu 1 döngüde de 3 kez meydana gelmektedir.)

$V_1(2)=V_2(2)=V_4(2)=V_5(2)=0$ ($x=2$ durumu 0 döngüde $\{1, 2, 4, \geq 5\}$ kez meydana gelmektedir.)

$V_0(-4)=3$ ($x=-4$ durumu 3 döngüde de 0 kez meydana gelmektedir.)

$V_1(-4)=V_2(-4)=V_3(-4)=V_4(-4)=V_5(-4)=0$ ($x=-4$ durumu 0 döngüde $\{1, 2, 3, 4, \geq 5\}$ kez meydana gelmektedir.)

Bu aşamadan sonra x değerinin 8 durumu için aşağıda verilen Ki-kare istatistiği Denklem (2.39) kullanılarak hesaplanması gerekmektedir.

$$\chi^2(\text{obs}) = \sum_{k=0}^5 \frac{(v_k(x) - J\pi_k(x))^2}{J\pi_k(x)} \quad (2.39)$$

Yukarıda elde edilen değerler kullanılarak sadece $x=1$ durumu için;

$$\begin{aligned} \chi^2 &= \frac{(1 - 3(0.5))^2}{3(0.5)} + \frac{(1 - 3(0.25))^2}{3(0.25)} + \frac{(0 - 3(0.125))^2}{3(0.125)} \\ &+ \frac{(1 - 3(0.0625))^2}{3(0.0625)} + \frac{(0 - 3(0.0312))^2}{3(0.0312)} + \frac{(0 - 3(0.0312))^2}{3(0.0312)} = 4.3330 \end{aligned}$$

olarak hesaplanmaktadır. Buradan $P\text{-değeri} = \text{igamc}(5/2, 4.3330/2) = 0.502$ değeri bulunmaktadır. $P\text{-değeri} = 0.502 \geq 0.001$ olduğundan dizi rasgele olarak kabul edilmektedir [71, 115].

15. Rasgele Gezinimler Değişken Testi (The Random Excursions Variant Test): Bu testte amaç, birikimli toplam rasgele yürüyüşte belirli durumların toplam meydana gelme sayısının incelenmesidir. Bu test, rasgele bir yürüyüşte çeşitli durumlar için ziyaretin beklenen sayıdaki sapmalarını belirlemektedir. Bu testte -9, -8, -7, -6, -5, -4, -3, -2, -1 ve +1, +2, +3, +4, +5, +6, +7, +8, +9 olmak üzere on sekiz $P\text{-değerinin}$ hesaplandığı serisel bir testtir.

Bu test için örneğin $\varepsilon = \varepsilon_1, \varepsilon_2, \dots, \varepsilon_{10} = 0110110101$ olmak üzere bir bit dizisi verilmektedir. Buradan $n=10$ ve $X_i = 2\varepsilon_i - 1$ dönüşümü kullanılarak giriş dizisi -1 ve +1 değerlerine normalize edilmektedir. Yeni X_i dizisi $X = -1, 1, 1, -1, 1, 1, -1, 1, -1, 1$ olmaktadır. Bu test için verilen örneğe göre birikimli toplamlar testinde verildiği gibi

uygulama metodu için 1 seçilerek ileri yönlü metot uygulanırsa S_i kısmi toplam olmak üzere $S_1=-1, S_2=0, S_3=1, S_4=0, S_5=1, S_6=2, S_7=1, S_8=2, S_9=1, S_{10}=2$ değerleri elde edilmektedir. Buradan $S=\{-1, 0, 1, 0, 1, 2, 1, 2, 1, 2\}$ olmaktadır. S' kümesi S kümesinin başına ve sonuna 0 elemanlarının eklenmesi ile oluşturulmaktadır. Sonuç olarak $S'=\{0, -1, 0, 1, 0, 1, 2, 1, 2, 1, 2, 0\}$ kümesi elde edilmektedir. Rasgele dizideki döngü sayısını ifade eden J değeri, S' kümesindeki ilk sıfır elemanı hariç kümedeki sıfır elemanlarının sayısı ve ζ ise testte bütün rasgele yürüyüşler süresince ziyaret edilen durumların toplam sayısını temsil etmektedir. Bu bölümde verilen örnek için: $\zeta(-1)=1, \zeta(1)=4, \zeta(2)=3$ ve diğerleri $\zeta(x)=0$ olmaktadır. Her bir döngü için ve her bir sıfır olmayan durum değeri için x $-9 \leq x \leq -1$ ve $9 \leq x \leq 1$ değerlerini alabilmekte ve her bir döngü içerisinde her bir x değerinin frekansı hesaplanmaktadır. Sonraki aşamada ise P -değerinin hesaplanabilmesi amacıyla her bir $\zeta(x)$ değeri için (18 $\zeta(x)$ değeri ile) 18 tane P -değeri Denklem (2.40) kullanılarak hesaplanmaktadır.

$$P - value = \operatorname{erfc} \left(\frac{|\zeta(x) - J|}{\sqrt{2J(4|x| - 2)}} \right) \quad (2.40)$$

Yukarıda elde edilen değerler kullanılarak sadece $x=1$ durumu için;

$$P - value = \operatorname{erfc} \left(\frac{|4 - 3|}{\sqrt{23(4|1| - 2)}} \right) = 0.683$$

olarak elde edilmektedir. Buradan P -değeri $=0.502 \geq 0,001$ olduğundan dizi rasgele kabul edilmektedir [71, 115].

BÖLÜM 3. REFERANS KAOTİK SİSTEMLERİN NÜMERİK VE ELEKTRONİK DEVRE MODELLERİ

GRSÜ tasarımlarında kullanılmak üzere FPGA çipleri üzerinde gerçekleştirilmeye uygun kaotik osilatörler tasarlanması amacıyla sürekli zamanlı kaotik sistem/sistemler kullanılacaktır. Literatürde sürekli zamanlı bir çok örnek kaotik sistem sunulmuş ve bu sistemler ile ilgili çalışmalar yapılmıştır [77–84]. Literatüre sunulan her farklı yapıdaki kaotik sistem, kaos tabanlı bilimsel araştırmalarda kullanılmak üzere önemli bir referans çalışma modeli olarak kabul edilmektedir. Bu çalışma kapsamında referans olarak literatüre yeni sunulan ve üzerinde FPGA-tabanlı herhangi bir çalışma yapılmamış olan sürekli zamanlı örnek iki yeni kaotik sistem belirlenmiştir. Öncelikle belirlenen otonom yeni kaotik sistemlerin kaos analizleri yapılmıştır. Ardından diferansiyel denklemler ile ifade edilen kaotik sistemler nümerik diferansiyel denklem çözüm yöntemleri kullanılarak modellenmiştir. Nümerik yöntemleri ile yapılan modellemelerden elde edilen sinyal değerleri kullanılarak kaotik sistemlerin zaman serileri ve faz portreleri çıkarılmıştır. Ayrıca çeşitli analiz yöntemleri ile sistemlerin denge noktaları ve Lyapunov spektrumu analizleri sunulmuştur. Sonraki aşamada ise analog devre elemanları kullanılarak kaotik sistemlerin bir ECAD programında simülasyonları yapılmıştır. ECAD programında analog devre elemanları kullanılarak elde edilen modelleme sonuçları ile kaotik sistemin zaman serileri ve faz portreleri çıkarılmıştır. Son olarak nümerik ve elektronik devre modelleri karşılaştırılmıştır.

3.1. Sundarapandian-Pehlivan Kaotik Sistemi

Kaotik sistemler genellikle fark denklemleri ile ifade edilmektedir. Aşağıda Sundarapandian-Pehlivan Kaotik Sistemi (SPKS) için diferansiyel denklemler (3.1) verilmiştir [120]. Burada Denklem (3.2)'deki α , β ve γ sistem parametrelerini ve Denklem (3.3) kaotik sistemin başlangıç şartlarını ifade etmektedir. Kaotik sistemler

başlangıç şartlarına ve sistem parametrelerini oldukça duyarlı sistemlerdir. Başlangıç şartlarındaki veya sistem parametrelerindeki küçük bir değişim sistemin dinamik davranışının değişmesine neden olabilmektedir.

$$\begin{aligned} dx/dt &= \alpha \cdot y - x \\ dy/dt &= -\beta \cdot x - z \end{aligned} \quad (3.1)$$

$$\begin{aligned} dz/dt &= \gamma \cdot z + x \cdot y^2 - x \\ \alpha &= 1.5, \beta = 0.4, \gamma = 0.4 \end{aligned} \quad (3.2)$$

$$x_0 = 0, y_0 = 0, z_0 = 0.1 \quad (3.3)$$

Nonlinear dinamik bir sistemin kaotik olduğunun belirlenebilmesi için çeşitli şartlar bulunmaktadır. Bu şartlardan ilki sistemde mutlaka bir veya daha fazla doğrusal olmayan bir elemanın bulunmasıdır. Diğer bir şart ise sürekli zamanlı sistemlerde sistemin en az üçüncü dereceden bir sistem olmasıdır [46]. Sistem ayrık zamanlı ise böyle bir şart bulunmamaktadır. Eğer bir sistem bu iki şartı sağlıyorsa sistemde kaos analizi yapılabilmektedir. Literatürde bir sistemin kaotik analizinin yapılabilmesi için geliştirilmiş çeşitli yöntemler bulunmaktadır. Bu yöntemlerden bazılarında sistemin faz uzayının incelenmesi (phase portrait), yörüngenin izlenmesi (zaman serileri), Poincare haritalama, güç spektrumu, çatallaşma (bifurcation) diyagramı ve Lyapunov üstelleri frekans spektrumu örnek olarak verilebilir [121, 122]. Kaotik analiz yöntemlerinden birisi olan Lyapunov üstelleri yöntemi, Aleksandr Mikhailovich Lyapunov tarafından geliştirilmiş olup, sistemin zaman serisinin kaotik bileşenler içerip içermediğini gösteren matematiksel bir analiz yöntemidir [123]. Lyapunov kararlılık yöntemleri hem doğrusal hem de doğrusal olmayan sistemlerin kararlılık analizinde kullanılmaktadır. Lyapunov'un kararlılık yöntemi için $\dot{x} = f(x)$ ve $x \in \mathcal{R}^n$ olmak üzere doğrusal olmayan bir sistemi göz önüne alalım. $f(x)$ fonksiyonunun x_1, x_2, \dots, x_n değişkenlerine göre sürekli ve türevlenebilir fonksiyonlar olduğunu varsayalım. $f(x)$ fonksiyonu x_e denge noktası için $f(x_e) = 0$ olmaktadır. $f(x)$ fonksiyonunu x_e denge noktası civarında Taylor serisine açtıktan sonra $y = x - x_e$ dönüşümü sayesinde denge noktası orijine kaydırılabilmektedir. $\dot{y} = Ay + xG(y)$ ifadesinde A , Denklem (3.4)'te verilen Jakobiyen matrisi, Ay

doğrusal ve $G(y)y$ doğrusal olmayan terimi belirtmek üzere Taylor serisindeki yüksek dereceli türevlerden oluşan $G(y)y$ 'nin tüm elemanları 0 olarak kabul edilmektedir [124].

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}_{x=x_e} \quad : \text{Jakobiyen matrisi} \quad (3.4)$$

Bu şekilde denklem $\dot{y} = Ay$ olarak yazılabilmektedir. Lyapunov bu metot ile göstermiştir ki A matrisinin bütün özdeğerlerinin (köklerinin) gerçel kısımları sıfırdan farklı ise $\dot{x} = f(x)$ nonlieer sisteminin denge durumundaki kararlılığı sistemin kararlılığına eşit olmaktadır. Sonuç olarak, A matrisinin özdeğerlerinden birisinin gerçel kısmı sağ yarı düzlemde ise “*sistem kararsız*”, özdeğerlerinin tümü sol yarı düzlemde ise “*sistem asimptotik kararlı*” olduğu sonucuna varılmaktadır [124]. SPKS'nin x_e denge noktasının bulunabilmesi amacı ile Denklem (3.5)'te görüldüğü gibi sistem durum değişkenlerinin türevleri sıfıra eşitlenmektedir.

$$\begin{aligned} dx/dt = 0 & \quad 0 = \alpha \cdot y - x \\ dy/dt = 0 & \quad 0 = -\beta \cdot x - z \\ dz/dt = 0 & \quad 0 = \gamma \cdot z + x \cdot y - x \end{aligned} \quad (3.5)$$

Buna göre sistemin denge noktaları için $x_{1e}=x_{2e}=x_{3e}=0$ olarak alınırsa A jakobiyen matrisi aşağıdaki gibi olmaktadır.

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} \end{bmatrix}_{x=x_e} = \begin{bmatrix} -1 & \alpha & 0 \\ -\beta & 0 & -1 \\ y^2 - 1 & 2xy & \gamma \end{bmatrix}_{x=x_e} = \begin{bmatrix} -1 & \alpha & 0 \\ -\beta & 0 & -1 \\ -1 & 0 & \gamma \end{bmatrix}_{\substack{x_1=0 \\ x_2=0 \\ x_3=0}}$$

Kaotik sistemin karakteristik denklemi üzerinde $|\lambda I - A| = 0$ dönüşümü ile sistemin öz değerleri Denklem (3.6) kullanılarak elde edilmektedir.

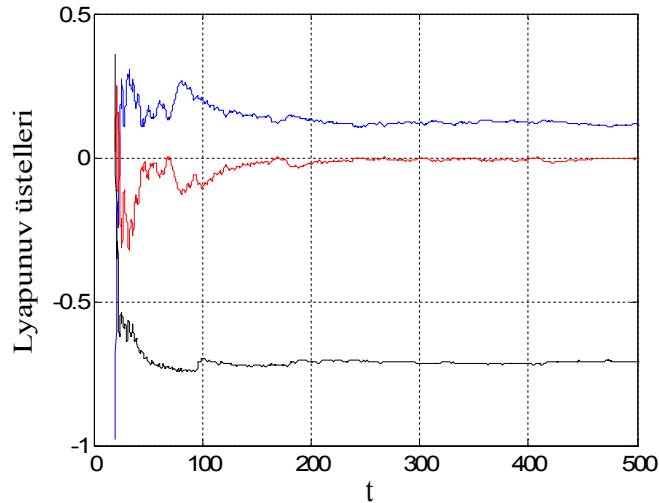
$$\det \begin{vmatrix} \lambda + 1 & -\alpha & 0 \\ \beta & \lambda & 1 \\ 1 & 0 & \lambda - \gamma \end{vmatrix} = \lambda^3 - \lambda^2 \gamma + \lambda^2 - \lambda \gamma - \alpha + \lambda \alpha \beta - \alpha \beta \gamma = 0 \quad (3.6)$$

Kaotik sistemin $\alpha=1.5$, $\beta=1.5$ ve $\gamma=0.4$ olmak üzere parametre değerleri yerine konulursa;

$\lambda^3 - 0.4\lambda^2 + \lambda^2 - 0.4\lambda - 1.5 + 1.5 \cdot 0.4\lambda - 1.5 \cdot 0.4 \cdot 0.4 = \lambda^3 + 0.6\lambda^2 + 0.2\lambda - 1.74 = 0$ denklemi elde edilmektedir. Buradan kaotik sistemin öz değerleri hesaplanırsa

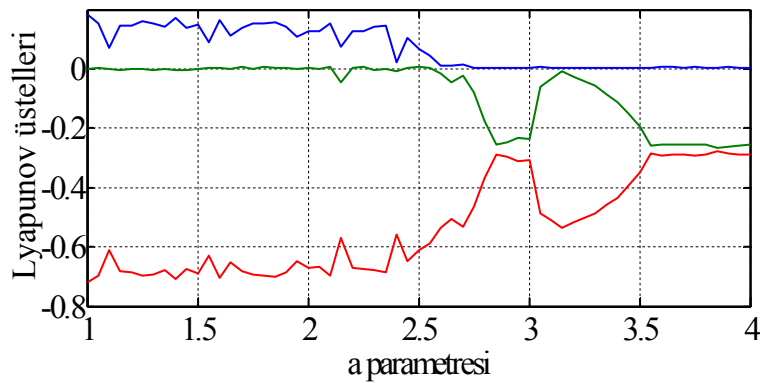
$\lambda_1 = 0.9862$, $\lambda_2 = -0.7931 + 1.0655i$ ve $\lambda_3 = -0.7931 - 1.0655i$ olarak

bulunmaktadır. Sonuç olarak, A matrisinin özdeğerlerinden birisinin gerçel kısmı sağ yarı düzlemde olduğundan “sistem kararsızdır” sonucuna varılmaktadır. Eğer üç boyutlu bir faz uzayında Lyapunov yöntemi ile kaotik analizi yapılan sistemin üstellerinin $(\lambda_1, \lambda_2, \lambda_3)$ işaretleri $(-, -, -)$ ise denge durumunda, $(0, 0, -)$ ise iki torus (halka), $(0, -, -)$ ise limit çevrim (kendini sürekli tekrarlayan), $(+, 0, -)$ ise garip çekici yani kaos durumundadır. Bir diğer ifade ile Lyapunov üstellerinin en büyüğünün değeri pozitif ise sistem yine kaotiktir. Sundarapandian-Pehlivan kaotik sisteminin kaos analizi için Lyapunov üstelleri [125] ve faz portresi yöntemleri kullanılmıştır. Yapılan Lyapunov üstelleri analiz çalışmalarından elde edilen sonuçlar Şekil 3.1’de verilmiştir. Sonuçlardan da görüleceği üzere Lyapunov üstellerinin işaretlerinin birisi pozitif, diğeri sıfır ve sonuncusu negatiftir. Sonuçlara göre sistemin işaretleri $(\lambda_1, \lambda_2, \lambda_3)$ sırasıyla $(+, 0, -)$ olduğundan Sundarapandian-Pehlivan sistemi kaotiktir.

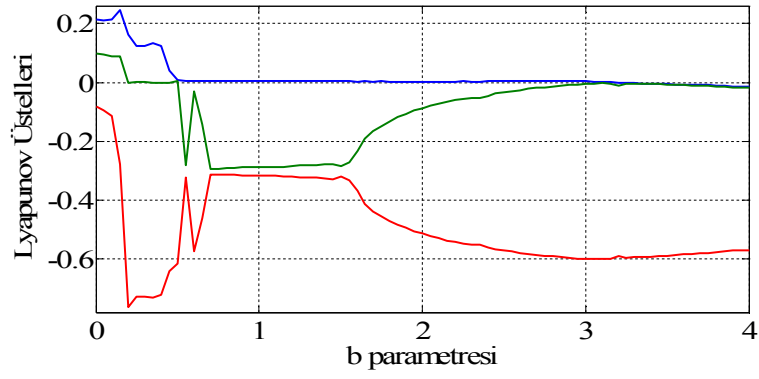


Şekil 3.1. $\alpha=1.5$, $\beta=0.4$ ve $\gamma=0.4$ değerleri için SPKS Lyapunov üstelleri

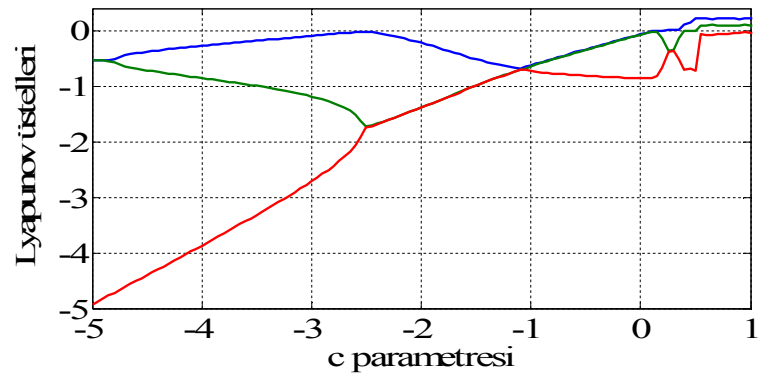
SPKS için Lyapunov üstelleri spektrumu $\beta=0.4$, $\gamma=0.4$ sabit değerleri ve $\alpha=1$ ile $\alpha=4$ arasındaki değerler için incelenmiştir. İnceleme sonuçları Şekil 3.2’de verilmektedir. Şekilde parametre değerleri $\alpha=1$ ile yaklaşık olarak $\alpha=2.8$ değerleri arasında sistemin kökleri olan λ_1 , λ_2 ve λ_3 değişkenlerinin değerleri birisi pozitif, ikincisi negatif ve üçüncüsü sıfır değerlerinde değişmektedir. SPKS bu değer aralıkları içerisinde kaotik bir davranış göstermektedir. SPKS $\alpha=2.8$ değerlerinden sonra λ_1 , λ_2 ve λ_3 değişkenlerinin değerleri birincisi sıfır, ikincisi ve üçüncüsü negatif değerlerinde değişmektedir. Sistem $\alpha=2.8$ değerinden sonra kaotik davranışını kaybetmektedir. SPKS için Lyapunov üstelleri spektrumu β ve γ parametrelerine göre değişimleri Şekil 3.3 ve Şekil 3.4’te verilmektedir.



Şekil 3.2. α parametresi için SPKS Lyapunov üstelleri spektrumu



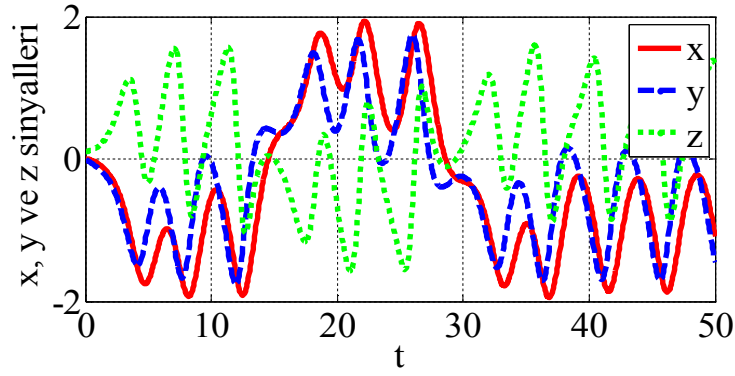
Şekil 3.3. β parametresine göre SPKS Lyapunov üstelleri spektrumu



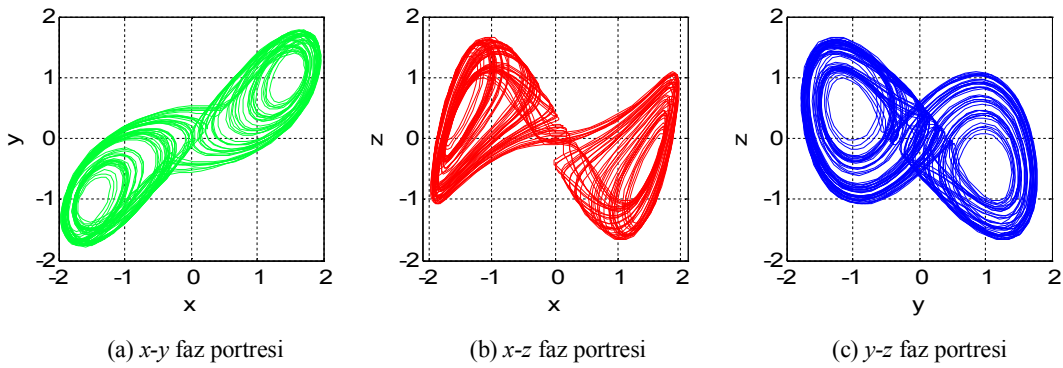
Şekil 3.4. γ parametresine göre SPKS Lyapunov üstelleri spektrumu

3.1.1. Sundarapandian-Pehlivan kaotik sistemi nümerik modeli

SPKS, hem faz portresinin oluşturulması hem de elektronik devre tasarımı modelinin sonuçlarını doğrulamak amacıyla Matlab programında nümerik olarak modellenmiştir. Sistemin x , y ve z çıkışlarından elde edilen nümerik sonuçlar Şekil 3.5'te grafiksel olarak verilmiştir. Şekil 3.6 (a)'da x - y sinyallerinin, 3.6 (b)'de x - z sinyallerinin ve 3.6 (c)'de y - z sinyallerinin faz portreleri verilmiştir.



Şekil 3.5. Sundarapandian-Pehlivan kaotik sistemi x , y ve z zaman serileri



Şekil 3.6. SPKS osilatörü nümerik faz portreleri

3.1.2. Sundarapandian-Pehlivan kaotik sistemi Orcad-PSpice modeli

SPKS, Orcad-PSpice modelinin elde edilebilmesi için x , y ve z sinyallerinin değerleri simetrik olarak beslenen op-amp'ın besleme değerleri arasında yer almalıdır. SPKS nümerik modelinden elde edilen sonuçlardan görüleceği üzere sistemin x , y ve z sinyallerinin alt ve üst aralık değerleri -2 V ile 2 V aralığındadır. Sonuç olarak, PSpice programında tasarımın yapılabilmesi için herhangi bir ölçeklendirmeye gerek kalmadan devre modellenenmektedir. Aşağıda kaotik sistemin x , y ve z sistem değişkenlerine göre analizi yapılmıştır. Devrede ilk sistem çıkışı olarak x değişkeni alınırsa buna göre ilk op-amp devresinin çıkışı;

$$x(t) = \left(-\frac{1}{R_1 C_1} \right) \cdot \int_0^t (x(t) dt) - \left(-\frac{1}{R_2 C_1} \right) \cdot \int_0^t (y(t) dt) \quad (3.7)$$

olarak bulunur. Burada ilk op-amp devresi için, eşitliğin her iki tarafının türevi alındığında Denklem (3.8) elde edilmektedir.

$$dx/dt = -\frac{1}{C_1} \left[\left(\frac{x}{R_1} \right) - \left(\frac{y}{R_2} \right) \right] \quad (3.8)$$

Sistemin y çıkışı analiz edilirse; buna göre op-amp devresinin Denklemi (3.9);

$$y(t) = \left(-\frac{1}{R_3 C_2} \right) \cdot \int_0^t x(t) dt - \left(-\frac{1}{R_4 C_2} \right) \cdot \int_0^t y(t) dt \quad (3.9)$$

olarak bulunmaktadır. Buradan y çıkışlı op-amp devresi için eşitliğin her iki tarafının türevi alındığında Denklem (3.10) elde edilmektedir.

$$dy/dt = -\frac{1}{C_2} \left[\left(\frac{x}{R_3} \right) + \left(\frac{y}{R_4} \right) \right] \quad (3.10)$$

Kaotik sistemin z çıkışı için op-amp devresi analiz edilirse buna göre devrenin Denklemi (3.11);

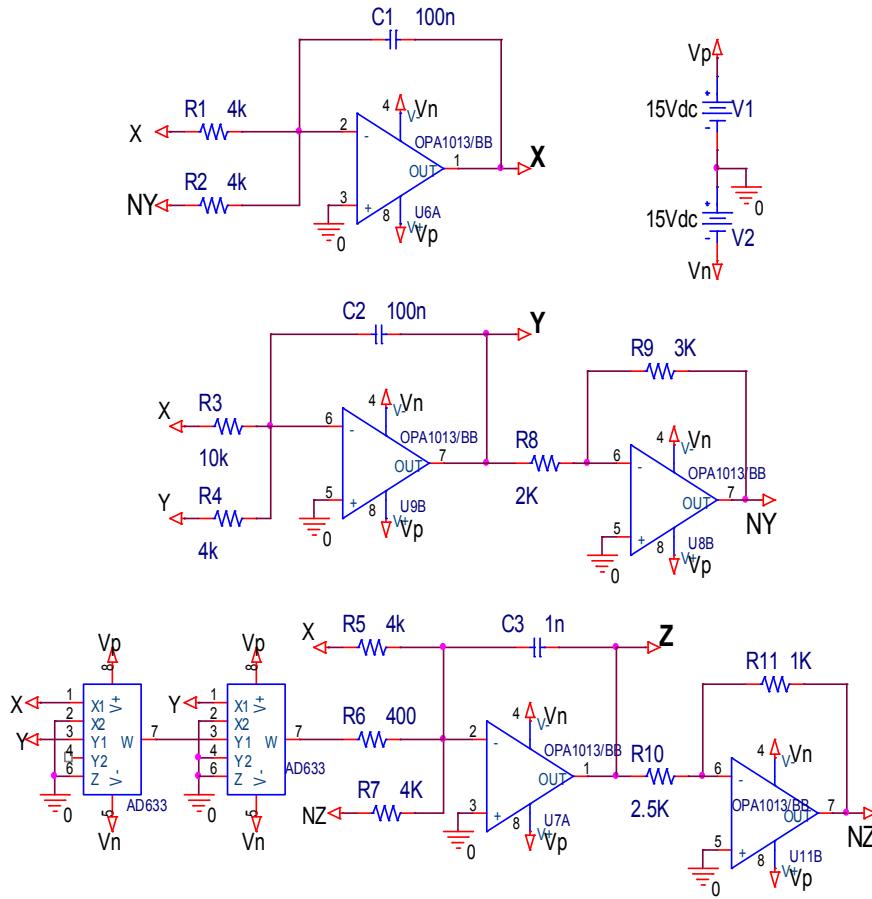
$$z(t) = \left(-\frac{1}{R_5 C_3} \right) \cdot \int_0^t x(t) dt + \left(-\frac{1}{R_6 C_3} \right) \cdot \int_0^t xy^2(t) dt - \left(-\frac{1}{R_7 C_3} \right) \cdot \int_0^t z(t) dt \quad (3.11)$$

olarak bulunur. Burada z çıkışlı op-amp devresi için eşitliğin her iki tarafının türevi alındığında Denklem (3.12) elde edilmektedir.

$$dz/dt = -\frac{1}{C_3} \left[\left(\frac{x}{R_5} \right) - \left(\frac{xy^2}{R_6} \right) + \left(\frac{z}{R_7} \right) \right] \quad (3.12)$$

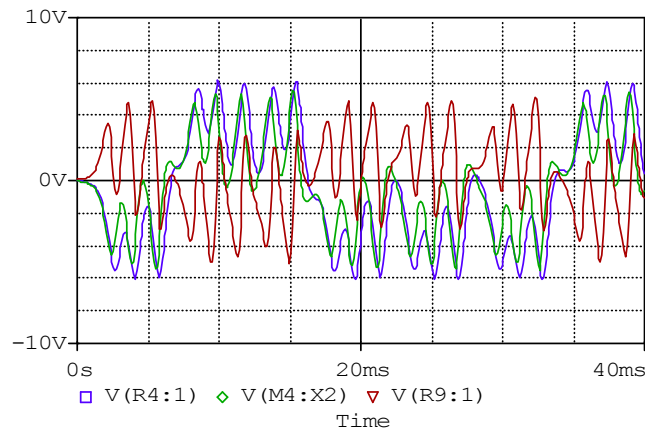
SPKS osilatörünün denklemleri analog elemanlar kullanılarak modellendiğinde, sistemde beş adet TL081 op-amp, iki adet AD633 çarpıcı, üç kapasitör ve değişik değerlere sahip on bir adet direnç elemanı kullanılmaktadır. Devre bağlantılarının

karmaşık görünümünden kurtarılması amacıyla bazı sinyaller sadece isimleri kullanılarak verilmiştir. Devrede kullanılan işlemsel yükselteçler (op-amp) ve çarpıcı üniteleri simetrik $\pm 15 V$ güç kaynağı ile beslenmiştir. Kaotik sistemin Orcad-PSpice modeli için kullanılan direnç elemanlarının değerleri $R_1=R_2=R_4=R_5=R_7=4 K\Omega$, $R_3=10 K\Omega$, $R_6=400 \Omega$, $R_9=3 K\Omega$, $R_{10}=2.5 K\Omega$ ve $R_{11}=1 K\Omega$ olarak belirlenmiştir. Kullanılan kapasitör elemanlarının değerleri ise $C_1=C_2=100 nF$ ve $C_3=1 nF$ olarak seçilmiştir. Besleme gerilim değerleri ise $V_P=15 V$, $V_N=-15 V$ olarak verilmektedir. Şekil 3.7’de ORCAD-PSpice programında tasarımı yapılan SPKS elektronik devre şeması verilmiştir.

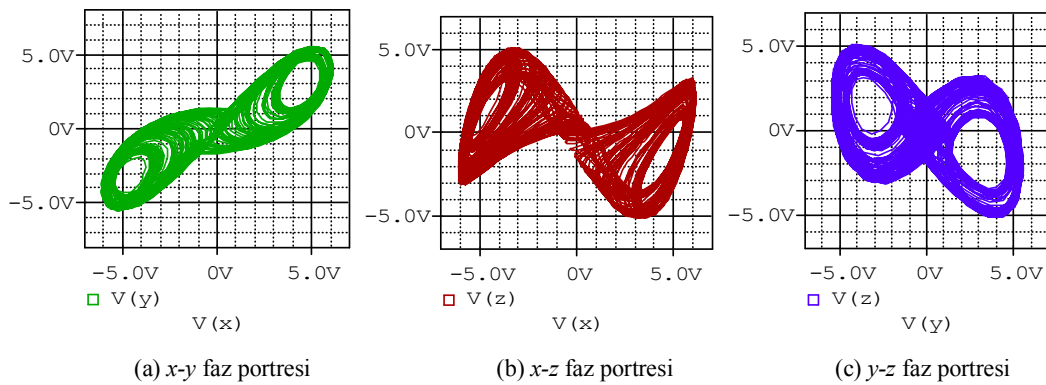


Şekil 3.7. SPKS elektronik devre şeması

Elektronik devre elemanları ile tasarımı yapılan kaotik sistemin Şekil 3.8’de PSpice simülasyon sonuçları verilmiştir. Orcad-PSpice simülasyonundan elde edilen sonuçlara göre Şekil 3.9 (a)’da x - y sinyallerinin, Şekil 3.9 (b)’de x - z sinyallerinin ve Şekil 3.9 (c)’de y - z sinyallerinin faz portreleri verilmiştir.



Şekil 3.8. SPKS osilatörü PSpice simülasyonu x , y ve z sinyalleri zaman serileri



Şekil 3.9. SPKS osilatörü PSpice faz portreleri

3.2. Pehlivan-Wei Kaotik Sistemi

Aşağıda Pehlivan-Wei Kaotik Sistemi (PWKS) için diferansiyel denklemler (3.13) verilmiştir [126]. Buradaki denklemden $\alpha=2$ olmak üzere sistem parametresini ve Denklem (3.14) kaotik denklemin başlangıç şartlarını ifade etmektedir.

$$\begin{aligned} dx/dt &= y(1-z) \\ dy/dt &= y(1+z) - \alpha x \end{aligned} \quad (3.13)$$

$$\begin{aligned} dz/dt &= \alpha - x.y - y^2 \\ x_0 &= -4.0, y_0 = 1.0, z_0 = -4.0 \end{aligned} \quad (3.14)$$

PWKS'nin Lyapunov kararlılık analizinde x_e denge noktasının bulunabilmesi amacı ile Denklem (3.15)'te görüldüğü gibi sistem durum değişkenlerinin türevleri sıfıra eşitlenmektedir.

$$\begin{aligned} dx/dt = 0 & \quad 0 = y(1-z) \\ dy/dt = 0 & \quad 0 = y(1+z) - \alpha x \\ dz/dt = 0 & \quad 0 = \alpha - x y - y^2 \end{aligned} \quad (3.15)$$

Buna göre sistemin denge noktaları için $x_{1e}=x_{2e}=x_{3e}=0$ olarak alınırsa A jakobiyen matrisi aşağıdaki gibi olmaktadır.

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} \end{bmatrix}_{x=x_e} = \begin{bmatrix} 0 & 1-z & -y \\ -\alpha & 1+z & y \\ -y & -x-2y & 0 \end{bmatrix}_{x=x_e} = \begin{bmatrix} 0 & 1 & 0 \\ -\alpha & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}_{\substack{x_1=0 \\ x_2=0 \\ x_3=0}}$$

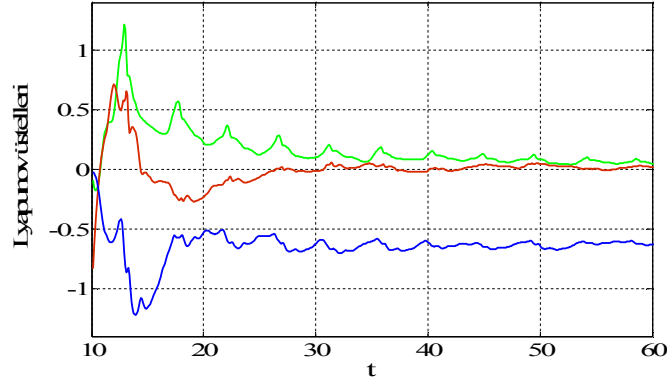
Kaotik sistemin karakteristik denklemi üzerinde $|\lambda I - A| = 0$ dönüşümü yapılarak sistemin öz değerleri Denklem (3.16)'daki gibi elde edilmektedir.

$$\det \begin{vmatrix} \lambda & -1 & 0 \\ \alpha & \lambda - 1 & 0 \\ 0 & 0 & \lambda \end{vmatrix} = \lambda^3 - \lambda^2 + \lambda \alpha = 0 \quad (3.16)$$

Kaotik sistemin parametre değeri yerine konulursa $\lambda^3 - \lambda^2 + 2\lambda = 0$ denklemi elde edilmektedir. Buradan kaotik sistemin öz değerleri hesaplanırsa $\lambda_1 = 0.5000 + 1.3229i$, $\lambda_2 = 0.0000$ ve $\lambda_3 = 0.5000 - 1.3229i$ olarak bulunmaktadır. Sonuç olarak A matrisinin özdeğerlerinden birisinin gerçel kısmı sağ yarı düzlemde olduğundan “sistem kararsızdır” sonucuna varılmaktadır.

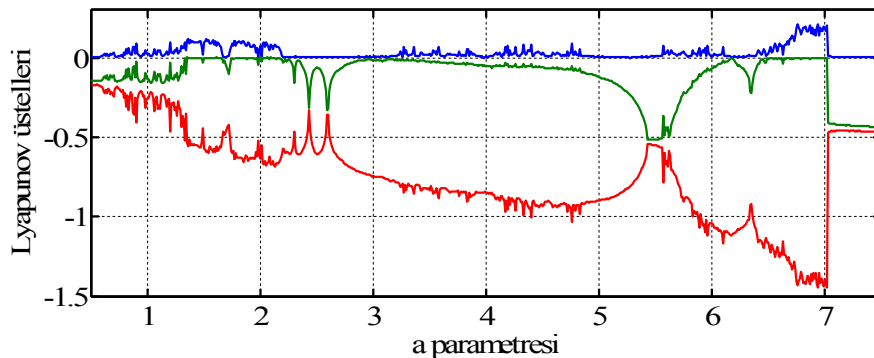
Aşağıda Şekil 3.10'da PWKS için Lyapunov üsteli yönteminden elde edilen sonuçlar verilmiştir. Sonuçlardan da görüleceği üzere Lyapunov üstellerinin işaretlerinin

birincisi pozitif, ikincisi sıfır ve sonuncusu negatiftir. Sonuçlara göre sistemin işaretleri $(\lambda_1, \lambda_2, \lambda_3)$ sırasıyla $(+, 0, -)$ olduğundan PWS kaotiktir.



Şekil 3.10. $\alpha=2$ değeri için PWKS Lyapunov üstelleri

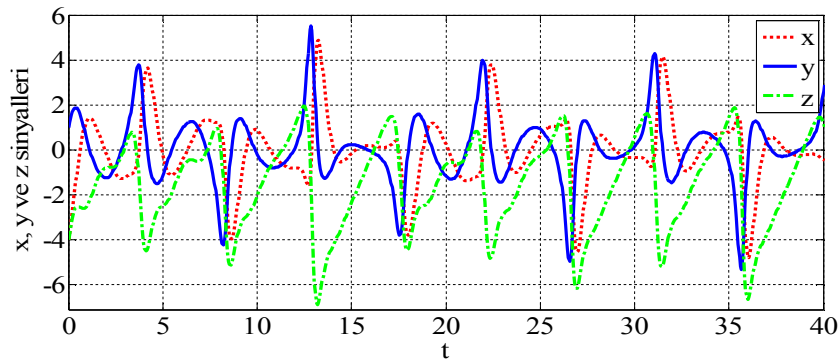
PWKS için Lyapunov üstelleri spektrumu $\beta=0.4$, $\gamma=0.4$ sabit değerleri ve $\alpha=1$ ile $\alpha=4$ arasındaki değerler için incelenmiştir. İnceleme sonuçları Şekil 3.11'de verilmektedir. Şekilde parametre değerleri $\alpha=1$ ile yaklaşık olarak $\alpha=2.8$ değerleri arasında sistemin kökleri olan $\lambda_1, \lambda_2, \lambda_3$ değişkenlerinin değerleri birisi pozitif, ikincisi negatif ve üçüncüsü sıfır değerlerinde değişmektedir. PWKS bu değer aralıkları içerisinde kaotik bir davranış göstermektedir. PWKS $\alpha=2.8$ değerlerinden sonra λ_1, λ_2 ve λ_3 değişkenlerinin değerleri birincisi sıfır, ikincisi ve üçüncüsü negatif değerlerinde değişmektedir. Sonuç olarak, α bu değer aralığından sonra kaotik davranışını kaybetmektedir.



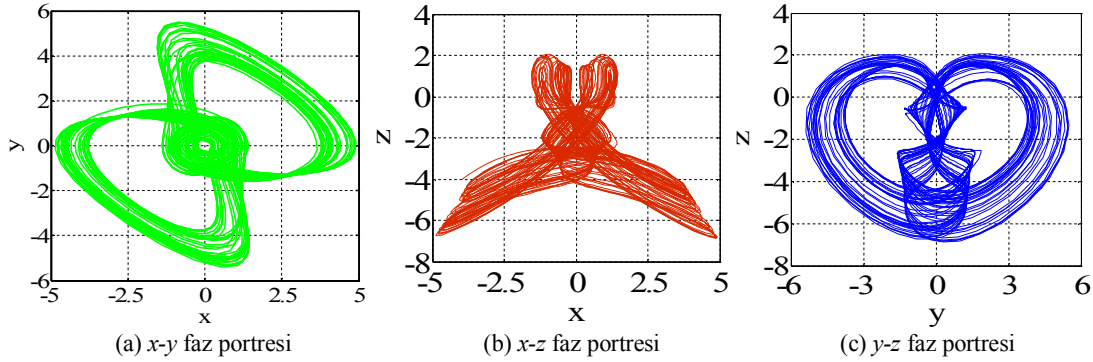
Şekil 3.11. α parametresine göre PWKS Lyapunov üstelleri spektrumu

3.2.1. Pehlivan-Wei kaotik sistemi nümerik modeli

PWKS faz portresinin oluşturulması hem de elektronik devre tasarımı modelinin sonuçlarını doğrulamak amacıyla Matlab programında nümerik olarak modellenmiştir. Sistemin x , y ve z çıkışlarından elde edilen nümerik sonuçlar Şekil 3.12’de grafiksel olarak verilmiştir. Şekil 3.13 (a)’da x - y sinyallerinin, 3.13 (b)’de x - z sinyallerinin ve 3.13 (c)’de y - z sinyallerinin faz portreleri verilmiştir.



Şekil 3.12. PWKS x , y ve z zaman serileri



Şekil 3.13. PWKS osilatörü nümerik faz portreleri

3.2.2. Pehlivan-Wei kaotik sistemi Orcad-PSpice modeli

PWKS nümerik modelinden elde edilen sonuçlardan görüleceği üzere sistemin x , y ve z sinyallerinin alt ve üst aralık değerleri -6 V ile 6 V aralığındadır. Sonuç olarak PSpice programında tasarımın yapılabilmesi için herhangi bir ölçeklendirmeye gerek kalmadan devre modellenenmektedir. Aşağıda kaotik sistemin x , y ve z durum

değişkenlerine göre analizi yapılmıştır. Devrede ilk sistem çıkışı olarak x alınırsa buna göre ilk op-amp devresinin çıkışı Denklem (3.18)'deki gibi bulunmaktadır.

$$x(t) = \left(-\frac{1}{R_3 C_1} \right) \cdot \int_0^t y(t) dt + \left(-\frac{1}{R_4 C_1} \right) \cdot \int_0^t y z(t) dt \quad (3.17)$$

Burada ilk op-amp devresi için eşitliğin her iki tarafının türevi alındığında Denklem (3.19) elde edilmektedir.

$$dx/dt = \frac{1}{C_1} \left[\left(\frac{y}{R_3} \right) - \left(\frac{y z}{R_4} \right) \right] \quad (3.18)$$

Kaotik sistemin y çıkışı analiz edilirse, buradan op-amp çıkışına göre Denklem (3.20)

$$y(t) = -\left(-\frac{1}{R_5 C_2} \right) \cdot \int_0^t y(t) dt + \left(-\frac{1}{R_6 C_2} \right) \cdot \int_0^t x(t) dt + \left(-\frac{1}{R_7 C_2} \right) \cdot \int_0^t y z(t) dt \quad (3.19)$$

olarak bulunmaktadır. Burada y çıkışlı op-amp devresi için eşitliğin her iki tarafının türevi alındığında Denklem (3.21) elde edilmektedir.

$$dy/dt = \frac{1}{C_2} \left[\left(\frac{y}{R_5} \right) - \left(\frac{x}{R_6} \right) - \left(\frac{y z}{R_7} \right) \right] \quad (3.20)$$

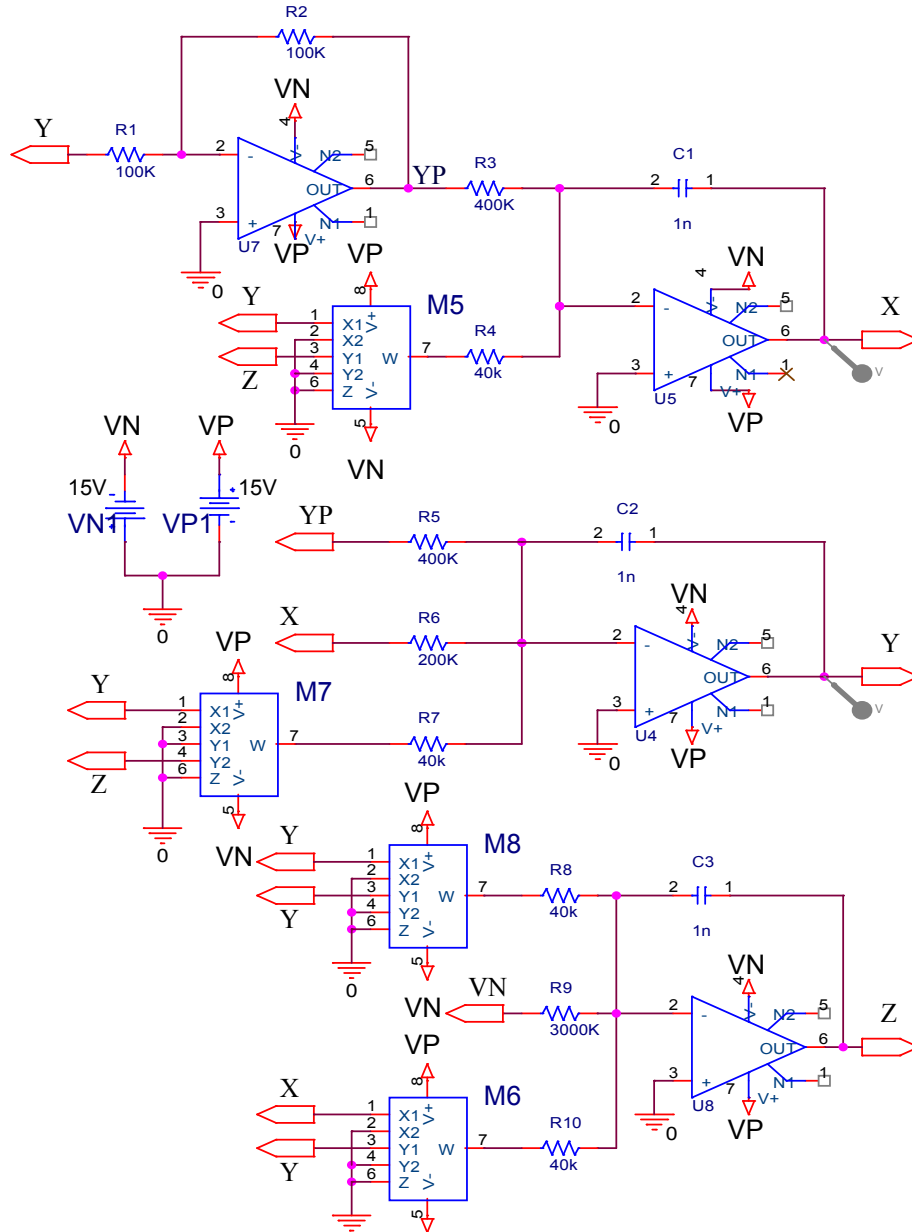
z çıkışı için op-amp devresi analiz edilirse; buna göre devrenin Denklemi (3.22);

$$z(t) = \left(-\frac{1}{R_8 C_3} \right) \cdot \int_0^t y^2(t) dt + \left(-\frac{1}{R_9 C_3} \right) \cdot \int_0^t Vn(t) dt + \left(-\frac{1}{R_{10} C_3} \right) \cdot \int_0^t x y(t) dt \quad (3.21)$$

olarak bulunur. Buradan z çıkışlı op-amp devresi için eşitliğin her iki tarafının türevi alındığında Denklem (3.23) elde edilmektedir.

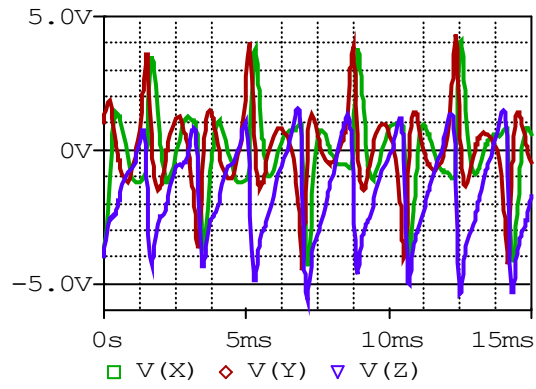
$$dz / dt = -\frac{1}{C_3} \left[\left(\frac{y^2}{R_8} \right) + \left(\frac{Vn}{R_9} \right) + \left(\frac{xy}{R_{10}} \right) \right] \quad (3.22)$$

PWKS osilatörünün denklemleri analog elemanlar kullanılarak modellendiğinde, sistemde dört adet TL081 op-amp, dört adet AD633 çarpıcı, üç kapasitör ve değişik değerlere sahip on adet direnç elemanı kullanılmıştır. Devre bağlantılarının karmaşık görünümünden kurtarılması amacıyla bazı sinyaller sadece isimleri kullanılarak verilmiştir. Devrede kullanılan işlemsel yükselteçler (op-amp) ve çarpıcı üniteleri simetrik $\pm 15V$ güç kaynağı ile beslenmiştir. Kaotik sistemin Orcad-PSpice modeli için kullanılan direnç elemanlarının değerleri $R_1=R_2=100 K\Omega$, $R_3=R_5=400 K\Omega$, $R_6=200 K\Omega$, $R_4=R_7=R_8=R_{10}=40 K\Omega$ ve $R_9=3 M\Omega$ olarak belirlenmiştir. Kullanılan kapasitör elemanlarının değerleri ise $C_1=C_2=C_3=1 nF$ olarak seçilmiştir. Besleme gerilim değerleri ise $V_P=15 V$, $V_N=-15 V$ olarak verilmektedir. Şekil 3.14'te ORCAD-PSpice programında tasarımı yapılan PWKS elektronik devre şeması verilmiştir.

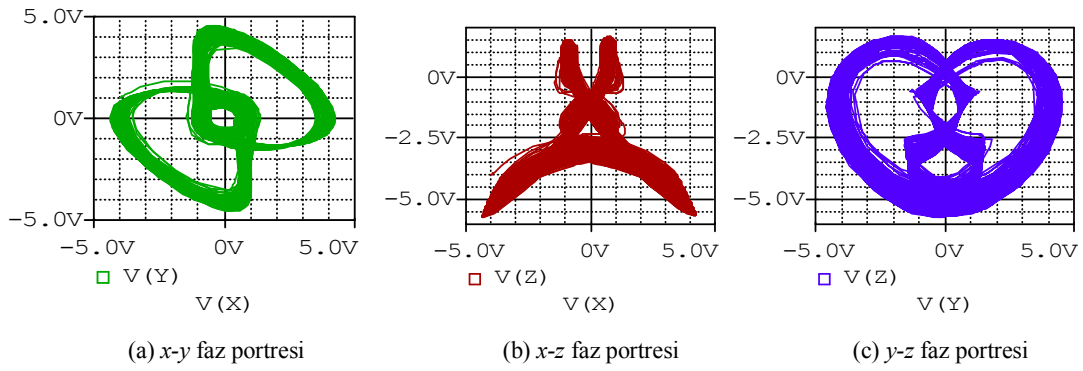


Şekil 3.14. PWKS elektronik devre şeması

Elektronik devre elemanları ile tasarımı yapılan kaotik sistemin Şekil 3.15'te PSpice simülasyon sonuçları verilmiştir. Orcad-PSpice simülasyonundan elde edilen sonuçlara göre Şekil 3.16 (a)'da x - y sinyallerinin, 3.16 (b)'de x - z sinyallerinin ve 3.16 (c)'de y - z sinyallerinin faz portreleri verilmiştir.



Şekil 3.15. PWKS osilatörü x , y ve z sinyalleri için PSpice zaman serileri



Şekil 3.16. PWKS osilatörü PSpice faz portreleri

BÖLÜM 4. FPGA TABANLI KAOTİK OSİLATÖRLERİN TASARIMI VE GERÇEKLENMESİ

Bu bölümde örnek olarak seçilen SP ve PW kaotik sistemlerinin Euler, Heun, RK4 ve RK5-Butcher nümerik diferansiyel denklem çözüm yöntemleri kullanılarak ayrıklaştırılmış modelleri çıkarılmıştır. Ayrıca ilgili kaotik sistemler VHDL dili kullanılarak FPGA üzerinde dört farklı nümerik algoritma kullanılarak tasarımları ve gerçeklemeleri yapılmıştır. Yapılan çalışmalar sonucunda elde edilen FPGA tabanlı kaotik sistem yapılarının performans ve çip istatistikleri incelenmiştir. Ayrıca FPGA tabanlı kaotik osilatörlerin ürettiği sinyaller kullanılarak algoritmaların hassasiyet analizi yapılmıştır.

4.1. Ayrıklaştırılmış Algoritmalar

Bu çalışmada SP ve PW kaotik sistemlerinin Euler, Heun, RK4 ve RK5-Butcher algoritmaları kullanılarak FPGA tabanlı olarak modellenmesini sağlamak amacıyla sistemin ayrıklaştırılmış modelleri oluşturulmuştur. Algoritmaların ayrıklaştırılmış modellerindeki $x(k)$, $y(k)$ ve $z(k)$ 'nin başlangıç değerleri SPKS için $x(t_0)=x(k)=0.0$, $y(t_0)=y(k)=0.0$ ve $z(t_0)=z(k)=0.1$ ve PWKS için $x(t_0)=x(k)=-4.0$, $y(t_0)=y(k)=1.0$ ve $z(t_0)=z(k)=-4.0$ olarak alınmıştır. Çalışmada sadece SPKS ayrıklaştırılmış algoritmaları sunulmuştur. PWKS ayrıklaştırılmış modelleri de benzer şekilde elde edilebilmektedir. Aşağıda Denklem (4.1)'de Euler algoritması kullanılarak SPKS ayrıklaştırılmış matematiksel modeli sunulmaktadır.

$$\begin{aligned}x(k+1) &= x(k) + \Delta h(\alpha \cdot y(k) - x(k)) \\y(k+1) &= y(k) + \Delta h(-\beta \cdot x(k) - z(k)) \\z(k+1) &= z(k) + \Delta h(\gamma \cdot z(k) + x(k) \cdot y(k)^2 - x(k))\end{aligned}\tag{4.1}$$

Heun tabanlı SPKS ayrıklaştırılmış matematiksel modeli Denklem (4.2)'de verilmektedir. Matematiksel modelden de görüldüğü gibi Heun algoritması iki

aşamalı bir algoritmadır. Birinci aşamada $x(k)$, $y(k)$ ve $z(k)$ değerleri kullanılarak sırasıyla $x(k^0+1)$, $y(k^0+1)$ ve $z(k^0+1)$ değerleri hesaplanmaktadır. İkinci aşamada ise hesaplanan $x(k^0+1)$, $y(k^0+1)$ ve $z(k^0+1)$ değerleri $x(k+1)$, $y(k+1)$ ve $z(k+1)$ ifadelerinde yerlerine konularak kaotik sistemin Δh adımı kadar sonraki değerleri hesaplanmaktadır.

$$\begin{aligned}
x(k^0+1) &= x(k) + \Delta h(\alpha \cdot y(k) - x(k)) \\
x(k+1) &= x(k) + \Delta h\left(\left(\alpha \cdot y(k) - x(k)\right) + x(k^0+1)\right) / 2 \\
y(k^0+1) &= y(k) + \Delta h(-\beta \cdot x(k) - z(k)) \\
y(k+1) &= y(k) + \Delta h\left(\left(-\beta \cdot x(k) - z(k)\right) + y(k^0+1)\right) / 2 \\
z(k^0+1) &= z(k) + \Delta h\left(\gamma \cdot z(k) + x(k) \cdot y(k)^2 - x(k)\right) \\
z(k+1) &= z(k) + \Delta h\left(\left(\gamma \cdot z(k) + x(k) \cdot y(k)^2 - x(k)\right) + z(k^0+1)\right) / 2
\end{aligned} \tag{4.2}$$

Aşağıda RK4 algoritması kullanılarak SPKS ayrıklaştırılmış matematiksel modeli Denklem (4.3)'teki f , g ve ζ fonksiyonlarına göre Denklem (4.4)'te verilmektedir.

$$\begin{aligned}
\dot{x} &= f(t, x, y, z) = \alpha \cdot y - x \\
\dot{y} &= g(t, x, y, z) = -\beta \cdot x - z \\
\dot{z} &= \delta(t, x, y, z) = \gamma \cdot z + x \cdot y^2 - x
\end{aligned} \tag{4.3}$$

$$\begin{aligned}
x(k+1) &= x(k) + \frac{1}{6} \Delta h [\kappa_1(k) + 2\kappa_2(k) + 2\kappa_3(k) + \kappa_4(k)] \\
y(k+1) &= y(k) + \frac{1}{6} \Delta h [\lambda_1(k) + 2\lambda_2(k) + 2\lambda_3(k) + \lambda_4(k)] \\
z(k+1) &= z(k) + \frac{1}{6} \Delta h [\xi_1(k) + 2\xi_2(k) + 2\xi_3(k) + \xi_4(k)]
\end{aligned} \tag{4.4}$$

Bu denklemlerde bulunan κ_1 , κ_2 , κ_3 ve κ_4 parametreleri, Denklem (4.3)'teki kaotik sistemin birinci denklemine ait değerleri, λ_1 , λ_2 , λ_3 ve λ_4 parametreleri ikinci denkleme ait değerleri ve ξ_1 , ξ_2 , ξ_3 ve ξ_4 parametreleri ise üçüncü denkleme ait değerleri Denklem (4.5)'te verildiği gibi hesaplanmaktadır. Bu katsayılar Denklem (4.4)'te sunulan RK4 algoritmasında yerine konularak, kaotik sistemin Δh kadar adım sonraki değeri olan $x(k+1)$, $y(k+1)$ ve $z(k+1)$ değerleri hesaplanmaktadır. Her iterasyon sonunda sistemin çıkışları olan $x(k+1)$, $y(k+1)$ ve $z(k+1)$ değerleri hem

çıkış olarak hem de bir sonraki iterasyonda algoritmanın başlangıç şartları olarak kullanılmaktadır.

$$\begin{aligned}
\kappa_1 &= f(x(k), y(k), z(k)) \\
\lambda_1 &= g(x(k), y(k), z(k)) \\
\xi_1 &= \delta(x(k), y(k), z(k)) \\
\kappa_2 &= f(x(k) + \frac{1}{2} \Delta h \kappa_1, y(k) + \frac{1}{2} \Delta h \lambda_1, z(k) + \frac{1}{2} \Delta h \xi_1) \\
\lambda_2 &= g(x(k) + \frac{1}{2} \Delta h \kappa_1, y(k) + \frac{1}{2} \Delta h \lambda_1, z(k) + \frac{1}{2} \Delta h \xi_1) \\
\xi_2 &= \delta(x(k) + \frac{1}{2} \Delta h \kappa_1, y(k) + \frac{1}{2} \Delta h \lambda_1, z(k) + \frac{1}{2} \Delta h \xi_1) \\
\kappa_3 &= f(x(k) + \frac{1}{2} \Delta h \kappa_2, y(k) + \frac{1}{2} \Delta h \lambda_2, z(k) + \frac{1}{2} \Delta h \xi_2) \\
\lambda_3 &= g(x(k) + \frac{1}{2} \Delta h \kappa_2, y(k) + \frac{1}{2} \Delta h \lambda_2, z(k) + \frac{1}{2} \Delta h \xi_2) \\
\xi_3 &= \delta(x(k) + \frac{1}{2} \Delta h \kappa_2, y(k) + \frac{1}{2} \Delta h \lambda_2, z(k) + \frac{1}{2} \Delta h \xi_2) \\
\kappa_4 &= f(x(k) + \Delta h \kappa_3, y(k) + \Delta h \lambda_3, z(k) + \Delta h \xi_3) \\
\lambda_4 &= g(x(k) + \Delta h \kappa_3, y(k) + \Delta h \lambda_3, z(k) + \Delta h \xi_3) \\
\xi_4 &= \delta(x(k) + \Delta h \kappa_3, y(k) + \Delta h \lambda_3, z(k) + \Delta h \xi_3)
\end{aligned} \tag{4.5}$$

Aşağıda RK5-Butcher algoritması kullanılarak SPKS ayrıklaştırılmış matematiksel modeli Denklem (4.6)'da verilmiştir. Buradaki $\kappa_1 \dots \kappa_6$, $\lambda_1 \dots \lambda_6$ ve $\xi_1 \dots \xi_6$ parametrelerinin açılımı Denklem (4.7)'de verilmektedir. RK5-Butcher algoritmasında beşinci ve altıncı dereceden terimler bulunduğundan RK4 algoritmasına göre daha hassas çözümler üretmektedir.

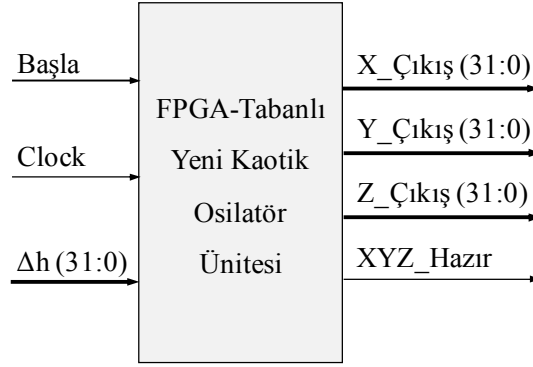
$$\begin{aligned}
x(k+1) &= x(k) + \frac{1}{90} \Delta h [7\kappa_1(k) + 32\kappa_3(k) + 12\kappa_4(k) + 32\kappa_5(k) + 7\kappa_6(k)] \\
y(k+1) &= y(k) + \frac{1}{90} \Delta h [7\lambda_1(k) + 32\lambda_3(k) + 12\lambda_4(k) + 32\lambda_5(k) + 7\lambda_6(k)] \\
z(k+1) &= z(k) + \frac{1}{90} \Delta h [7\xi_1(k) + 32\xi_3(k) + 12\xi_4(k) + 32\xi_5(k) + 7\xi_6(k)]
\end{aligned} \tag{4.6}$$

$$\begin{aligned}
\kappa_1 &= f(x(k), y(k), z(k)) \\
\lambda_1 &= g(x(k), y(k), z(k)) \\
\xi_1 &= \delta(x(k), y(k), z(k)) \\
\kappa_2 &= f(x(k) + \frac{1}{4}\Delta h\kappa_1, y(k) + \frac{1}{4}\Delta h\lambda_1, z(k) + \frac{1}{4}\Delta h\xi_1) \\
\lambda_2 &= g(x(k) + \frac{1}{4}\Delta h\kappa_1, y(k) + \frac{1}{4}\Delta h\lambda_1, z(k) + \frac{1}{4}\Delta h\xi_1) \\
\xi_2 &= \delta(x(k) + \frac{1}{4}\Delta h\kappa_1, y(k) + \frac{1}{4}\Delta h\lambda_1, z(k) + \frac{1}{4}\Delta h\xi_1) \\
\kappa_3 &= f(x(k) + \frac{1}{8}(\Delta h(\kappa_1 + \kappa_2), y(k) + \frac{1}{8}(\Delta h(\lambda_1 + \lambda_2), z(k) + \frac{1}{8}(\Delta h(\xi_1 + \xi_2))) \\
\lambda_3 &= g(x(k) + \frac{1}{8}(\Delta h(\kappa_1 + \kappa_2), y(k) + \frac{1}{8}(\Delta h(\lambda_1 + \lambda_2), z(k) + \frac{1}{8}(\Delta h(\xi_1 + \xi_2))) \\
\xi_3 &= \delta(x(k) + \frac{1}{8}(\Delta h(\kappa_1 + \kappa_2), y(k) + \frac{1}{8}(\Delta h(\lambda_1 + \lambda_2), z(k) + \frac{1}{8}(\Delta h(\xi_1 + \xi_2))) \\
\kappa_4 &= f(x(k) - \frac{1}{2}\Delta h\kappa_2 + \Delta h\kappa_3, y(k) - \frac{1}{2}\Delta h\lambda_2 + \Delta h\lambda_3, z(k) - \frac{1}{2}\Delta h\xi_2 + \Delta h\xi_3) \\
\lambda_4 &= g(x(k) - \frac{1}{2}\Delta h\kappa_2 + \Delta h\kappa_3, y(k) - \frac{1}{2}\Delta h\lambda_2 + \Delta h\lambda_3, z(k) - \frac{1}{2}\Delta h\xi_2 + \Delta h\xi_3) \\
\xi_4 &= \delta(x(k) - \frac{1}{2}\Delta h\kappa_2 + \Delta h\kappa_3, y(k) - \frac{1}{2}\Delta h\lambda_2 + \Delta h\lambda_3, z(k) - \frac{1}{2}\Delta h\xi_2 + \Delta h\xi_3) \\
\kappa_5 &= f(x(k) + \frac{3}{16}\Delta h\kappa_1 + \frac{9}{16}\Delta h\kappa_4, y(k) + \frac{3}{16}\Delta h\lambda_1 + \frac{9}{16}\Delta h\lambda_4, z(k) + \frac{3}{16}\Delta h\xi_1 + \frac{9}{16}\Delta h\xi_4) \\
\lambda_5 &= g(x(k) + \frac{3}{16}\Delta h\kappa_1 + \frac{9}{16}\Delta h\kappa_4, y(k) + \frac{3}{16}\Delta h\lambda_1 + \frac{9}{16}\Delta h\lambda_4, z(k) + \frac{3}{16}\Delta h\xi_1 + \frac{9}{16}\Delta h\xi_4) \\
\xi_5 &= \delta(x(k) + \frac{3}{16}\Delta h\kappa_1 + \frac{9}{16}\Delta h\kappa_4, y(k) + \frac{3}{16}\Delta h\lambda_1 + \frac{9}{16}\Delta h\lambda_4, z(k) + \frac{3}{16}\Delta h\xi_1 + \frac{9}{16}\Delta h\xi_4) \\
\kappa_6 &= f(x(k) - \frac{3}{7}\Delta h\kappa_1 + \frac{2}{7}\Delta h\kappa_2 + \frac{12}{7}\Delta h\kappa_3 - \frac{12}{7}\Delta h\kappa_4 + \frac{8}{7}\Delta h\kappa_5, y(k) + -\frac{3}{7}\Delta h\lambda_1 + \frac{2}{7}\Delta h\lambda_2 + \\
&\quad \frac{12}{7}\Delta h\lambda_3 - \frac{12}{7}\Delta h\lambda_4 + \frac{8}{7}\Delta h\lambda_5, z(k) - \frac{3}{7}\Delta h\xi_1 + \frac{2}{7}\Delta h\xi_2 + \frac{12}{7}\Delta h\xi_3 - \frac{12}{7}\Delta h\xi_4 + \frac{8}{7}\Delta h\xi_5) \\
\lambda_6 &= g(x(k) - \frac{3}{7}\Delta h\kappa_1 + \frac{2}{7}\Delta h\kappa_2 + \frac{12}{7}\Delta h\kappa_3 - \frac{12}{7}\Delta h\kappa_4 + \frac{8}{7}\Delta h\kappa_5, y(k) + -\frac{3}{7}\Delta h\lambda_1 + \frac{2}{7}\Delta h\lambda_2 + \\
&\quad \frac{12}{7}\Delta h\lambda_3 - \frac{12}{7}\Delta h\lambda_4 + \frac{8}{7}\Delta h\lambda_5, z(k) - \frac{3}{7}\Delta h\xi_1 + \frac{2}{7}\Delta h\xi_2 + \frac{12}{7}\Delta h\xi_3 - \frac{12}{7}\Delta h\xi_4 + \frac{8}{7}\Delta h\xi_5) \\
\xi_6 &= \delta(x(k) - \frac{3}{7}\Delta h\kappa_1 + \frac{2}{7}\Delta h\kappa_2 + \frac{12}{7}\Delta h\kappa_3 - \frac{12}{7}\Delta h\kappa_4 + \frac{8}{7}\Delta h\kappa_5, y(k) + -\frac{3}{7}\Delta h\lambda_1 + \frac{2}{7}\Delta h\lambda_2 + \\
&\quad \frac{12}{7}\Delta h\lambda_3 - \frac{12}{7}\Delta h\lambda_4 + \frac{8}{7}\Delta h\lambda_5, z(k) - \frac{3}{7}\Delta h\xi_1 + \frac{2}{7}\Delta h\xi_2 + \frac{12}{7}\Delta h\xi_3 - \frac{12}{7}\Delta h\xi_4 + \frac{8}{7}\Delta h\xi_5)
\end{aligned} \tag{4.7}$$

4.2. FPGA Tabanlı Kaotik Osilatörlerin Gerçeklenmesi

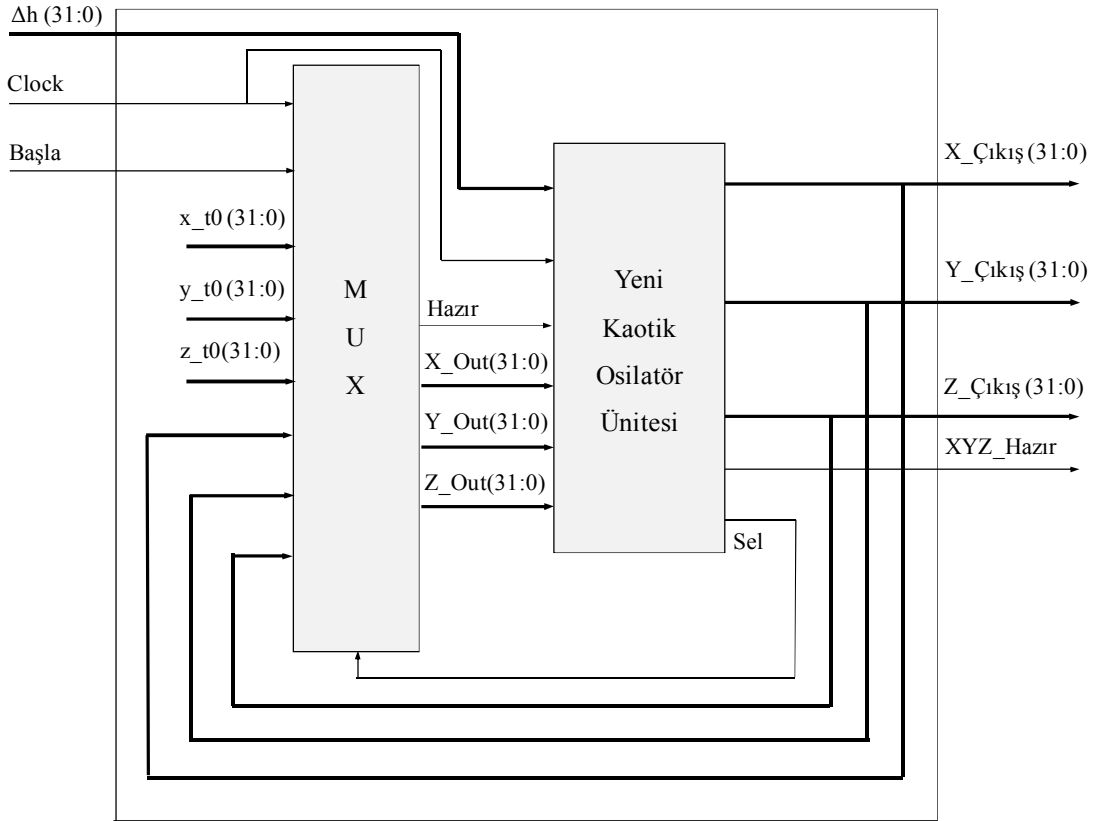
Sunulan örnek yeni kaotik sistemler Euler, Heun, RK4 ve RK5-Butcher algoritmaları kullanılarak, 32-bit IEEE 754-1985 kayan noktalı sayı standardı ile FPGA üzerinde çalışmak üzere modellenmiş ve VHDL'de kodlanmıştır. Yapılan tasarımlarda kullanılan kayan noktalı sayı standardına uygun çarpıcı, toplayıcı ve çıkarıcı gibi üniteler, Xilinx ISE Design Tools ile geliştirilen IP Core Generator kullanılarak oluşturulmuştur.

Tüm Euler, Heun, RK4 ve RK5-Butcher tabanlı *Yeni Kaotik Osilatör* (YKO) ünitelerinin en üst seviye blok diyagramları, iki yeni kaotik sistem için aynı olup Şekil 4.1'de görülmektedir. Ünitelerin girişlerinde bulunan 1-bitlik *Başla* ve *Clock* sinyalleri, ünitelerin içerisindeki alt modüllerin zamanlaması ve senkronizasyonu sağlamak amacıyla kullanılmaktadır. Algoritmanın hesaplanma hassasiyetini belirleyen Δh adım sayısı yine aynı sinyal ismi kullanılarak 32-bit giriş sinyali olarak belirlenmiştir. Bu sinyal, tasarımın daha esnek olmasını sağlamak amacıyla dışarıdan uygulanmaktadır. Sistemin ilk çalışma anında ihtiyaç duyduğu başlangıç şartları olan x_0 , y_0 ve z_0 sinyalleri 32-bit olarak tasarımın içerisinde tanımlanmıştır. Bunun en büyük nedeni, tasarımda kullanılan FPGA çipi kaynak kullanımını azaltmaktır. Ancak ihtiyaç duyulduğunda bu sinyaller, 32-bitlik 3 farklı sinyal tanımlaması yapılarak tasarımda küçük değişiklikler ile değerleri kullanıcı tarafından ayarlanacak şekilde de tasarlanabilir. Tasarlanan Euler, Heun, RK4 ve RK5-Butcher tabanlı kaotik osilatörlerde 3 adet kayan noktalı sayı standardında 32-bit çıkış sinyalleri (*X_Çıkış*, *Y_Çıkış* ve *Z_Çıkış*) ve bu çıkış sinyallerinin hazır olduğunu göstermek amacıyla kullanılan 1-bit *XYZ_Hazır* sinyali bulunmaktadır. Bu sinyaller sürekli zamanlı kaotik sistemin x , y ve z değişkenlerine karşılık gelmektedir.



Şekil 4.1. FPGA-Tabanlı YKO Ünitesi en üst seviye blok diyagramı

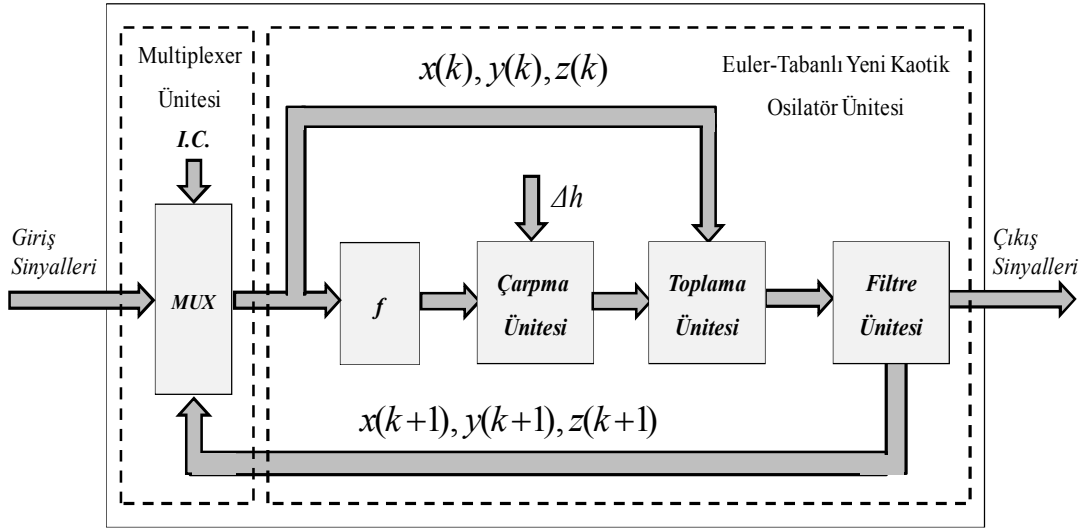
Şekil 4.2’de *YKO* ünitesinin ikinci seviye blok diyagramı görülmektedir. Sistemin ikinci seviye blok diyagramı *Multiplexer Ünitesi* ve *FPGA- Tabanlı YKO Ünitesi*’nden oluşmaktadır. Tasarımda *Multiplexer (MUX) Ünitesi* kullanılmasının amacı, başlangıç koşulu değerlerini ilk çalışma anında, kullanıcı tarafından atanan başlangıç sinyalleri olan 32-bit kayan noktalı sayı formatında x_{t0} , y_{t0} ve z_{t0} sinyallerinden almasını ve bundan sonraki tüm aşamalar için bu değerlerin *YKO Ünitesi* çıkışından almasını sağlamaktır. 1-bitlik *Sel* sinyali kaotik sistem sonuç ürettiği durumlarda ‘1’, bunun dışındaki tüm durumlarda ‘0’ değeri vermektedir. Bu şekilde *YKO Ünitesi* ilk değerlerini ürettiğinde, *Sel* sinyali ‘1’ olmakta ve bu sinyali *Multiplexer Ünitesi*’ne göndererek kullanıcı tarafından atanan başlangıç değerleri yerine kaotik sistemin ürettiği değerleri kullanmasını sağlamaktadır.



Şekil 4.2. FPGA-Tabanlı YKO Ünitesi ikinci seviye blok diyagramı

Şekil 4.3'te *Euler-Tabanlı YKO Ünitesi* üçüncü seviye blok diyagramı verilmektedir. Osilatör yapısında *MUX*, *Çarpma*, *Toplama*, *Filtre* ve *f* üniteleri olmak üzere 5 ünite kullanılmıştır. *MUX* ünitesinin işlevi *Başla* sinyali geldiğinde sistemin ihtiyaç duyduğu başlangıç şartlarının atanmasını sağlamaktır. Diğer bir ifade ile sisteme kullanıcı tarafından atanan başlangıç şartları ile sistemin çıkışından elde edilen ve bir sonraki algoritmanın hesaplanmasında başlangıç şartları olarak kullanılan $x(k+1)$, $y(k+1)$ ve $z(k+1)$ sinyalleri arasında seçim yaparak, bu sinyalleri sisteme göndermektir. *MUX* ünitesinin girişlerinden birisi olan *I.C.* kaotik sistemin başlangıç şartlarını ifade etmektedir. Sistemdeki *f* ünitesi *MUX* ünitesinden gelen kontrol sinyalleri ile PWKS veya SPKS denklemlerinin hesaplanmasını sağlamaktadır. Bu ünitelerden çıkan sinyaller ile algoritma adım sayısı olan Δh değeri *Çarpma Ünitesi* tarafından çarpılmaktadır. Ardından *Toplama Ünitesi* başlangıç şartı değerlerini toplayarak sonuçları *Filtre Ünitesi*'ne göndermektedir. Sistemde *Filtre Ünitesi*, kaotik osilatörün istenmeyen sinyaller üretmesini engellemek diğer bir ifade ile filtreleme işlemini gerçekleştirmek amacıyla kullanılmaktadır. *Euler-Tabanlı YKO*

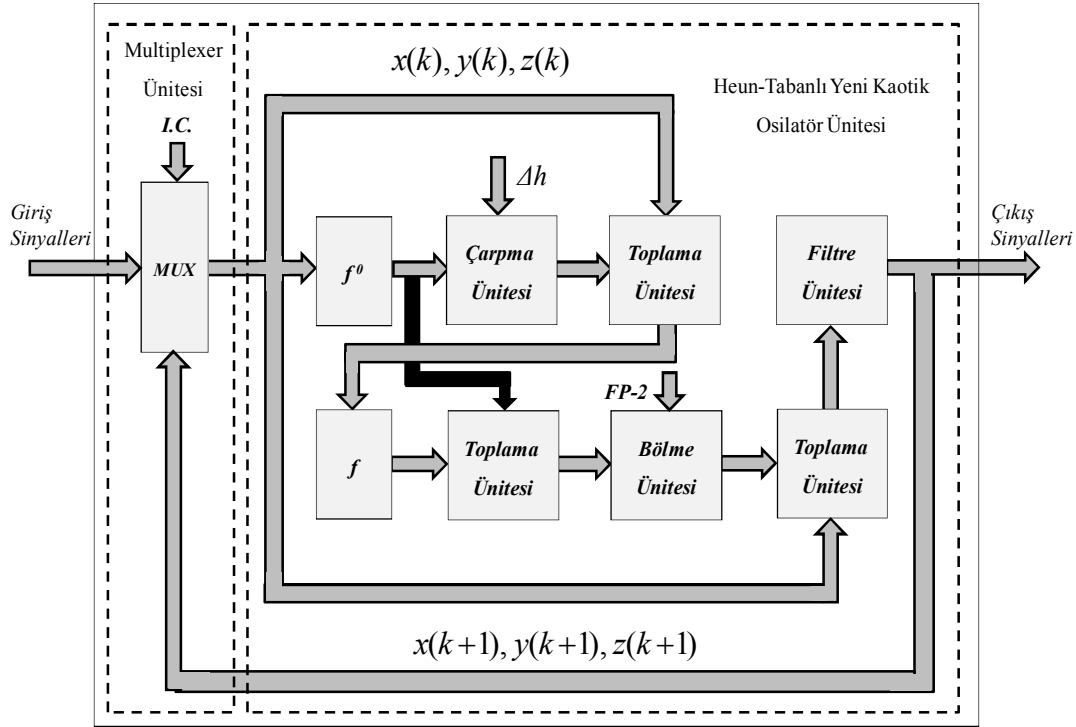
Ünitesi iş hattı tabanlı olarak çalışmakta ve 54 saat darbesi sonunda ilk değerini üretmektedir. Kaotik osilatör ünitesinin ürettiği sinyaller olan $x(k+1)$, $y(k+1)$ ve $z(k+1)$ sinyalleri sistemin hem çıkışlarını hem de bir sonraki iterasyonun hesaplanması için MUX ünitesine gönderilerek başlangıç şartlarını oluşturmaktadır.



Şekil 4.3. Euler-Tabanlı YKO Ünitesi üçüncü seviye blok diyagramı

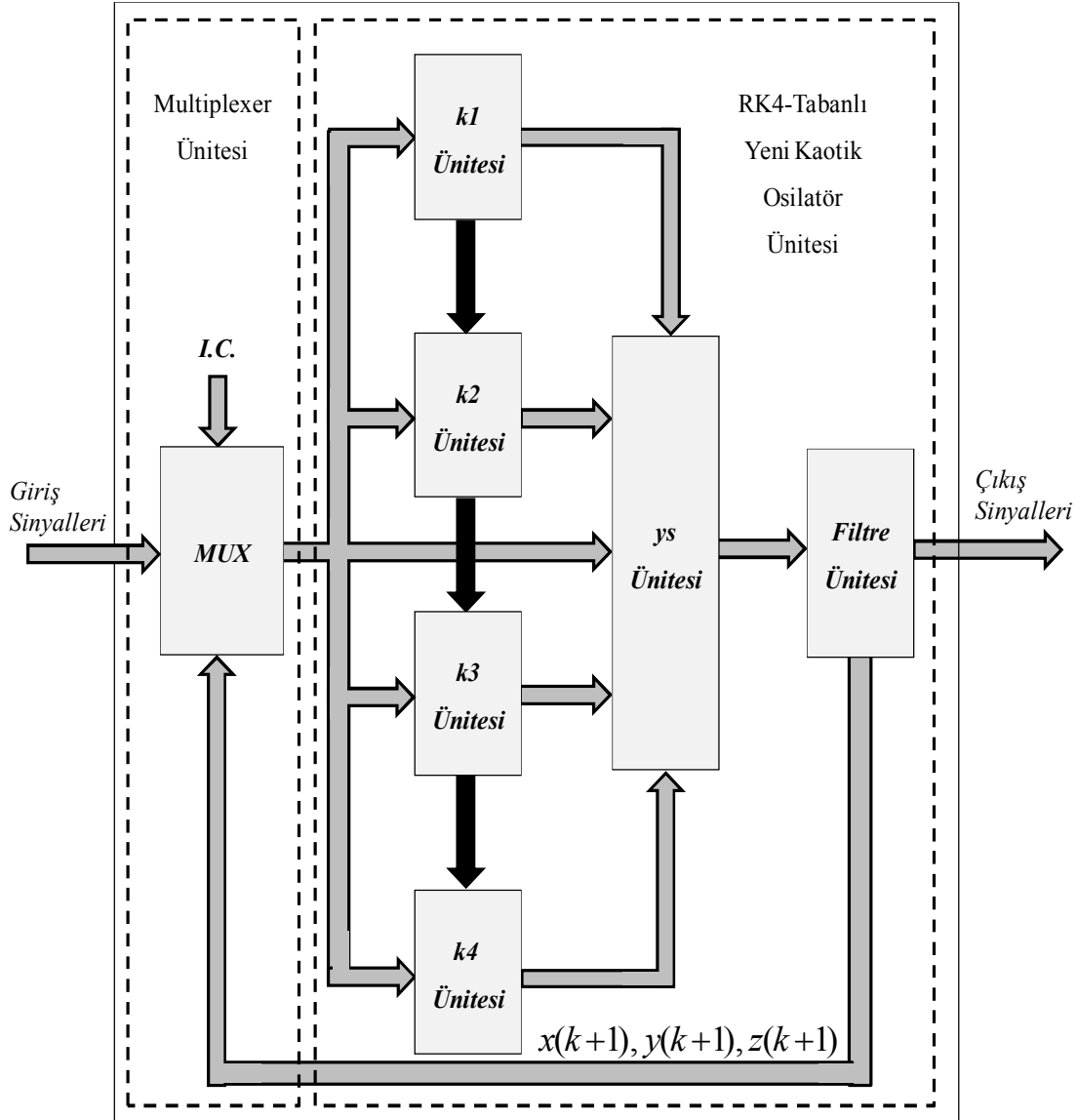
Şekil 4.4'te *Heun-Tabanlı YKO Ünitesi* üçüncü seviye blok diyagramı görülmektedir. Osilatör yapısında MUX, f^0 , Çarpma, Bölme, Toplama, f ve Filtre üniteleri olmak üzere 9 ünite bulunmaktadır. Buradaki f^0 ünitesi MUX ünitesinden gelen kontrol sinyalleri ile PWKS veya SPKS denklemlerindeki $x(k^0 + 1)$, $y(k^0 + 1)$ ve $z(k^0 + 1)$ değerlerinin hesaplanmasını sağlamaktadır. Bu üniteden elde edilen sinyaller Çarpma Ünitesi yardımıyla Δh adım sayısı ile çarpılmakta ve kaotik osilatörün başlangıç şartı değerleri ile $(x(k)$, $y(k)$ ve $z(k)$) toplanmaktadır. Bu şekilde algoritmanın ilk aşaması tamamlanmaktadır. İkinci aşamada, Toplama Ünitesi'nden çıkan sinyaller f^0 ünitesinden elde edilen sinyaller ile toplanmakta ve kayan noktalı sayı formatında 2.0 değerine (FP-2) Bölme Ünitesi yardımı ile bölünmektedir. Bölme Ünitesi'nden çıkan sinyaller ile kaotik osilatörün ürettiği bir önceki sinyaller ($x(k)$, $y(k)$ ve $z(k)$) Toplama Ünitesi'nde toplanarak sonuçlar Filtre Ünitesi'ne gönderilmektedir. Filtre Ünitesi'nden çıkan $x(k+1)$, $y(k+1)$ ve $z(k+1)$ sinyalleri sistemin hem çıkışlarını hem de bir sonraki iterasyonun hesaplanması için MUX ünitesine gönderilerek başlangıç şartlarını oluşturmaktadır. *Heun-Tabanlı Kaotik Osilatör Ünitesi* ilk değerini 132 saat darbesi sonunda üretmektedir. Kaotik Osilatör

Ünitesi iş hattı tabanlı olarak çalıştığından her 132 saat darbesi sonunda sistemin 32-bitlik çıkış sinyalleri olan $X_Çıkış$, $Y_Çıkış$ ve $Z_Çıkış$ değerlerini çıkışa aktarmaktadır.



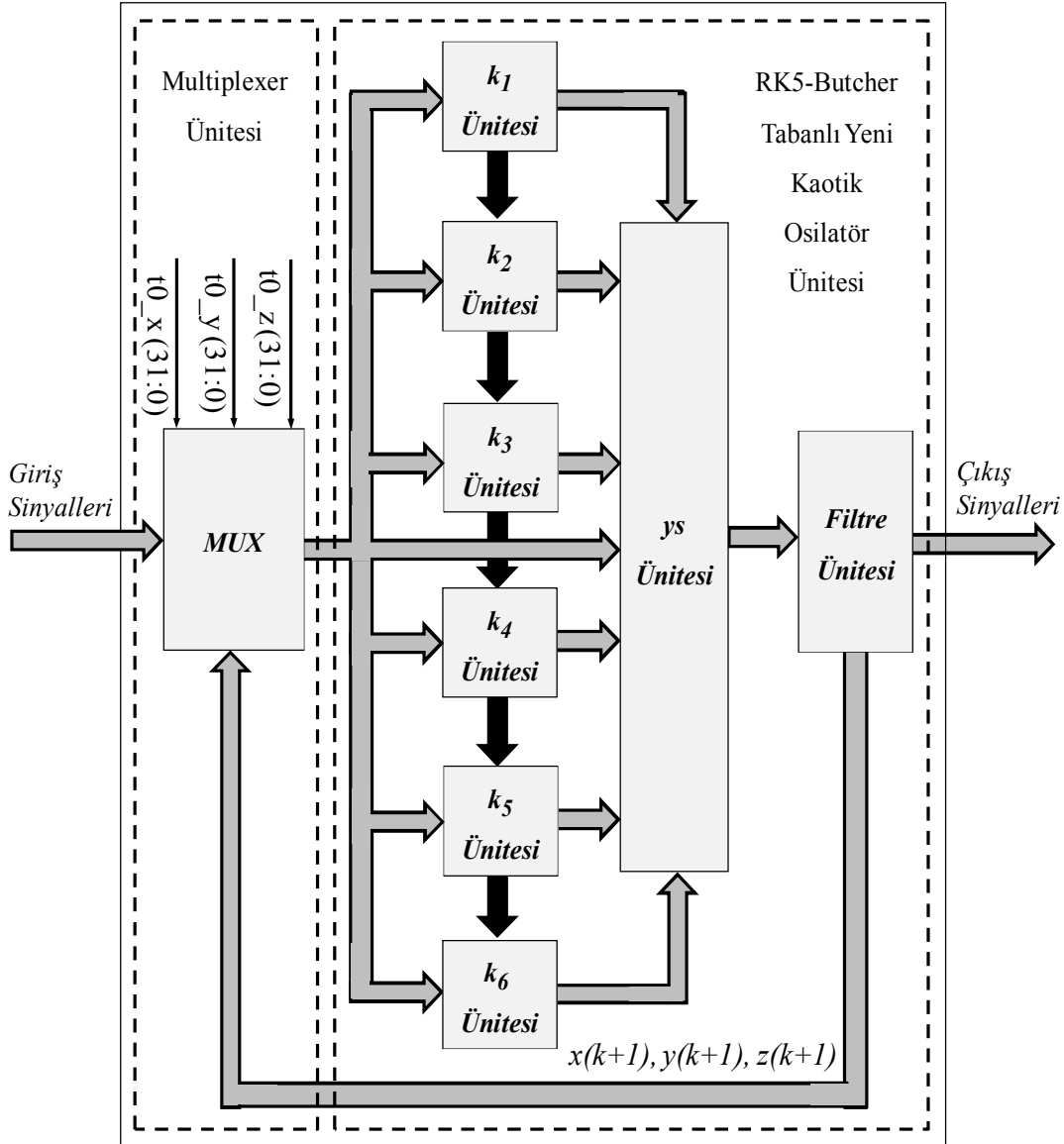
Şekil 4.4. Heun-Tabanlı YKO Ünitesi üçüncü seviye blok diyagramı

Şekil 4.5'te *RK4-Tabanlı YKO Ünitesi* üçüncü seviye blok diyagramı verilmektedir. Kaotik osilatörde *MUX*, k_1 , k_2 , k_3 , k_4 , ys ve *Filtre Ünitesi* olmak üzere toplam 7 ünite bulunmaktadır. k_1 , k_2 , k_3 ve k_4 Üniteleri kaotik sistemin ayrıklaştırılmış modelindeki $\sigma=1...4$ 'e kadar κ_σ , ζ_σ ve λ_σ değerlerini hesaplamaktadır. RK4 algoritmasında Denklem (4.4)'te verilen $x(k+1)$, $y(k+1)$ ve $z(k+1)$ değerleri ise *ys Ünitesi*'nde hesaplanmaktadır. Ünite iş hattı tabanlı olarak çalışmakta ve kaotik osilatör 178 saat darbesi sonunda ilk değerini üretmektedir. Ancak ünitenin bir sonraki $x(k+1)$, $y(k+1)$ ve $z(k+1)$ değerlerini üretebilmesi için geri besleme değerlerine ihtiyaç göstermekte ve bu değerleri 178 saat darbesi sonra üretmiş olduğu çıkış sinyallerini girişteki *MUX Ünitesi*'ne göndererek sağlamaktadır. Kaotik osilatörün sonuç üretmediği durumlarda, çıkışa istenmeyen sinyallerin ulaşmaması için bir *Filtre Ünitesi* kullanılmıştır. Bu şekilde bütün istenmeyen sinyaller filtrelenmekte ve çıkışa sadece istenen çıkış sinyalleri gönderilmektedir.



Şekil 4.5. RK4-Tabanlı YKO Ünitesi üçüncü seviye blok diyagramı

Şekil 4.6'da *RK5-Butcher-Tabanlı YKO Ünitesi* üçüncü seviye blok diyagramı görülmektedir. *Kaotik Osilatör Ünitesi*'nde, k_1 , k_2 , k_3 , k_4 , k_5 , k_6 , ys ve *Filtre Ünite*'si olmak üzere toplam 9 ünite bulunmaktadır. Ünite'ye *RK4-Tabanlı YKO Ünitesi*'nden farklı olarak sadece beşinci ve altıncı dereceden terimlerin hesaplanması için k_5 Ünitesi ve k_6 Ünitesi eklenmiştir. Bunun dışında Ünite, *RK4-Tabanlı YKO Ünitesi* ile benzer biçimde çalışmaktadır. *Kaotik Osilatör Ünitesi* iş hattı tabanlı olarak çalışmakta ve her 298 saat darbesi sonunda sistemin 32-bitlik çıkış sinyalleri olan $X_Çıkış$, $Y_Çıkış$ ve $Z_Çıkış$ değerlerini çıkışa aktarmaktadır.

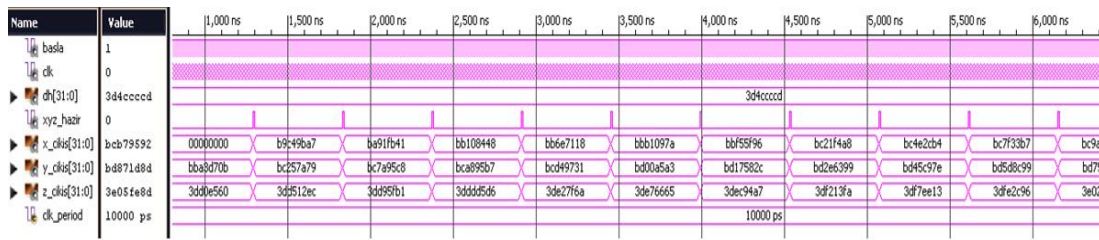


Şekil 4.6. RK5-Butcher-Tabanlı YKO Ünitesi üçüncü seviye blok diyagramı

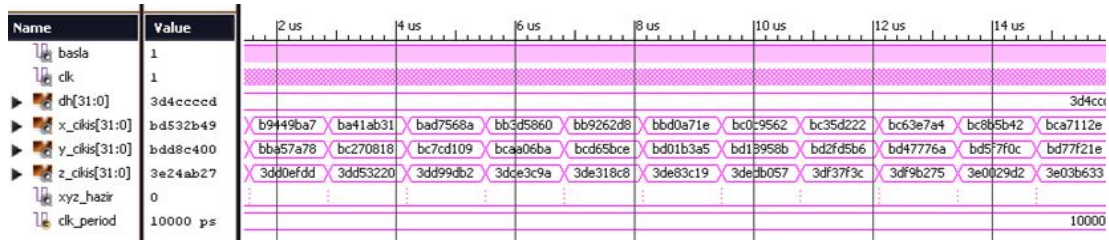
4.3. FPGA Tabanlı Kaotik Osilatörlerin Test Sonuçları

Euler, Heun, RK4 ve RK5-Tabanlı Osilatör Üniteleri, Xilinx Virtex-6 ailesi XC6VLX550T-2FF1759 çipi için sentezlenerek, FPGA çip kaynak kullanımına ve ünitelerin saat hızlarına ait parametrelerin istatistikleri incelenmiştir. Tasarımı yapılan 4 farklı yapıdaki ünitelerin verileri işleme süresi, Xilinx ISE Design Tools 14.2 benzetim programı kullanılarak elde edilmiştir. Burada kaotik osilatörün ISE Design Tools kullanılarak FPGA’de gerçekleştirilmesinden elde edilen x , y ve z sinyallerinin kaotik osilatör ünitesinde karşılıkları olan $X_{çıkış}$, $Y_{çıkış}$ ve $Z_{çıkış}$

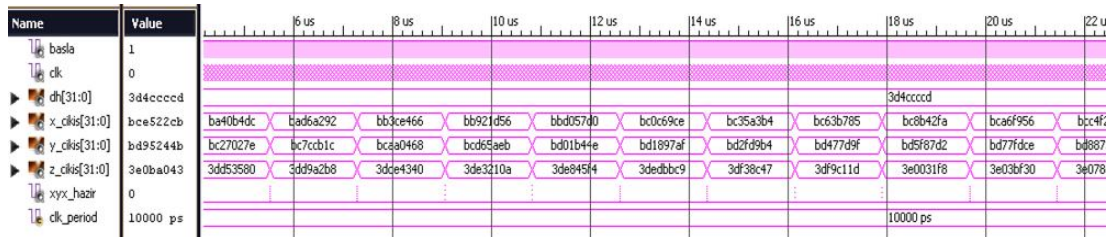
sinyallerinin zaman serilerine ait değerlerin daha kolay incelenebilmesini sağlamak amacıyla tasarımda 32-bit kayan noktalı sayı standardı kullanılmasına rağmen benzetim sonuçları onaltılık sayı formatında gösterilmiştir. SPKS osilatör ünitelerinin Xilinx ISE Simülâtöründe elde edilen sonuçlar Euler-tabanlı için Şekil 4.7, Heun-tabanlı için Şekil 4.8, RK4-tabanlı için Şekil 4.9 ve RK5-tabanlı için Şekil 4.10'da görülmektedir. PWKS osilatör ünitelerinin sonuçları ise Euler-tabanlı için Şekil 4.11, Heun-tabanlı için Şekil 4.12, RK4-tabanlı için Şekil 4.13 ve RK5-Butcher-tabanlı için Şekil 4.14'te verilmiştir.



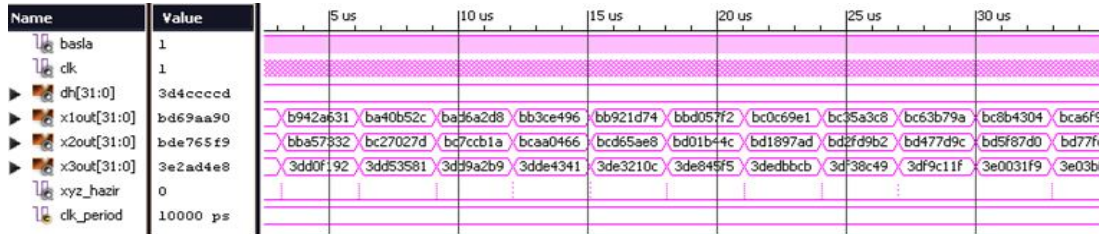
Şekil 4.7. Euler-tabanlı SPKS osilatör ünitesi Xilinx ISE Simülâtörü sonuçları



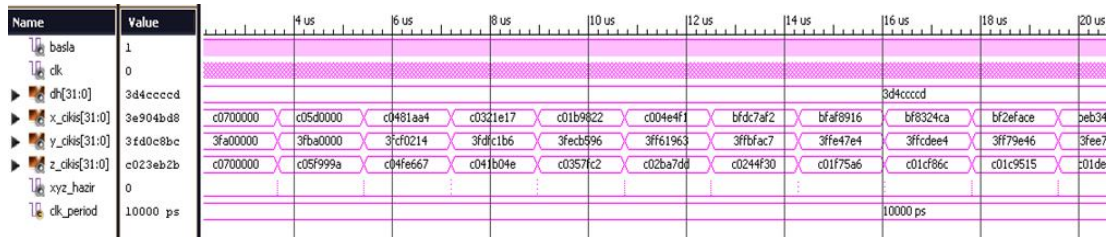
Şekil 4.8. Heun-tabanlı SPKS osilatör ünitesi Xilinx ISE Simülâtörü sonuçları



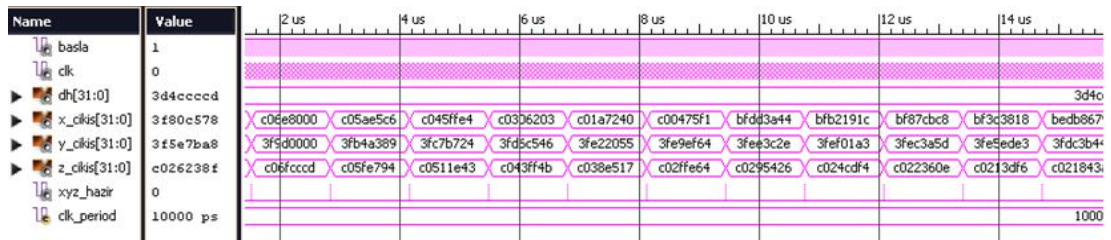
Şekil 4.9. RK4-tabanlı SPKS osilatör ünitesi Xilinx ISE Simülâtörü sonuçları



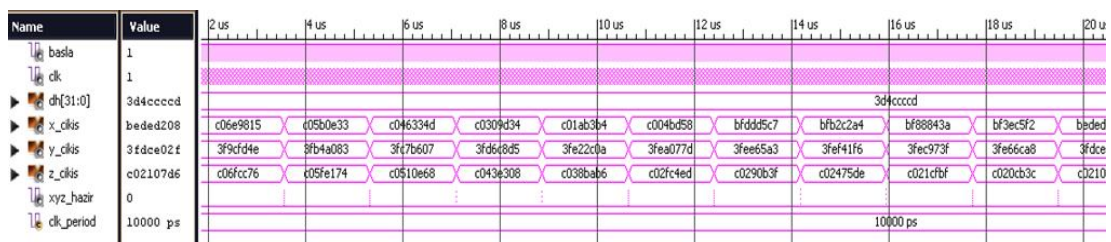
Şekil 4.10. RK5-Butcher-tabanlı SPKS ünitesi Xilinx ISE Simülatorü sonuçları



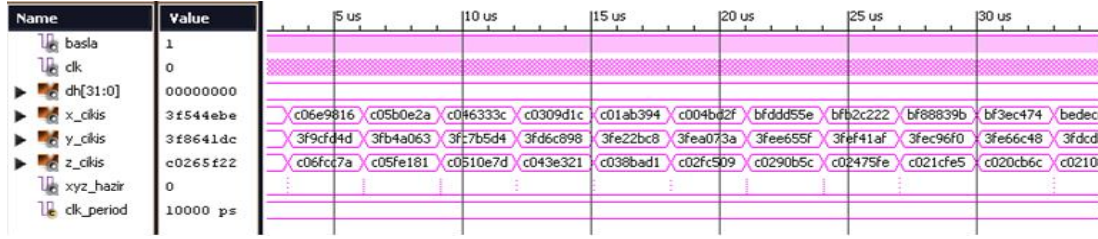
Şekil 4.11. Euler-tabanlı PWKS ünitesi Xilinx ISE Simülatorü sonuçları



Şekil 4.12. Heun-tabanlı PWKS ünitesi Xilinx ISE Simülatorü sonuçları

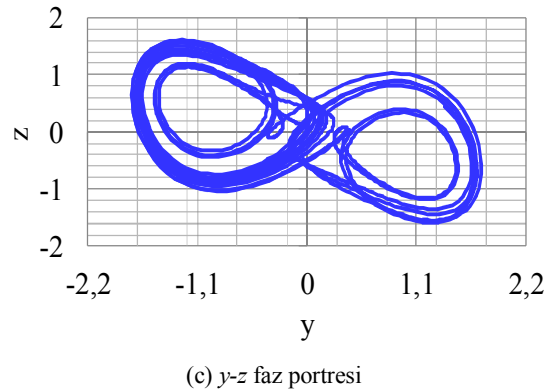
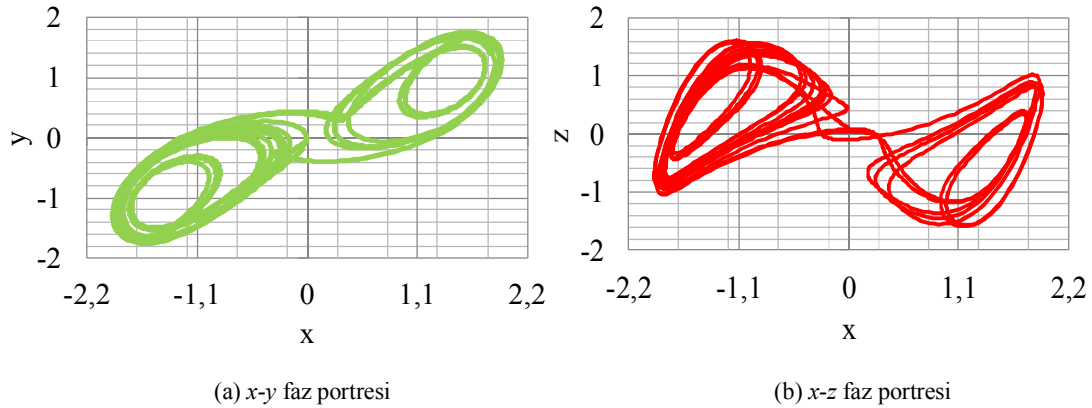


Şekil 4.13. RK4-tabanlı PWKS ünitesi Xilinx ISE Simülatorü sonuçları



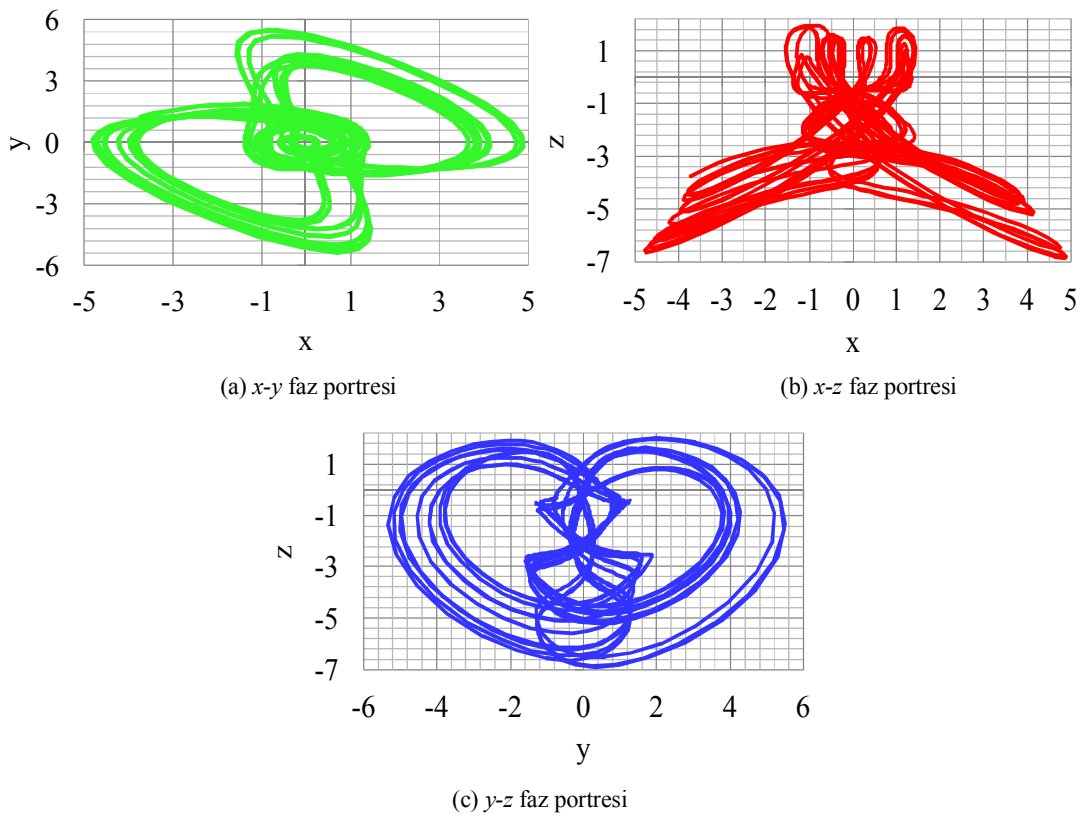
Şekil 4.14. RK5-Butcher-tabanlı PWKS ünitesi Xilinx ISE Simülatorü sonuçları

Ayrıca kaotik osilatörlerin FPGA’de gerçekleşmesinden elde edilen $X_çıkış$, $Y_çıkış$ ve $Z_çıkış$ sinyallerinin zaman serilerine ait 32-bit IEEE-754 formatındaki ikilik değerler, benzetim test aşamasında bir dosyaya kaydedilmiştir. Kaydedilen değerler gerçel sayı sistemine dönüştürüldükten sonra kaotik osilatörün ürettiği ilk 3×1750 veri seti yardımıyla $X_çıkış$, $Y_çıkış$ ve $Z_çıkış$ sinyalleri faz portreleri elde edilmiştir. Örnek olarak sadece Şekil 4.15’te SPKS RK4-tabanlı osilatörlerin FPGA üzerinde gerçekleşmesinden elde edilen faz portreleri verilmiştir.



Şekil 4.15. RK4-tabanlı SPKS osilatör ünitesi faz portreleri

Aynı şekilde, PWKS kaotik osilatörünün RK4 algoritması kullanılarak FPGA’de gerçekleştirilmesinden elde edilen $X_çıkış$, $Y_çıkış$ ve $Z_çıkış$ sinyallerinin zaman serilerine ait 32-bit IEEE–754 formatındaki ikilik değerler benzetim test aşamasında bir dosyaya kaydedilmiştir. Kaydedilen değerler gerçel sayı sistemine dönüştürüldükten sonra kaotik osilatörün ürettiği ilk 3x1750 veri seti yardımıyla $X_çıkış$, $Y_çıkış$ ve $Z_çıkış$ sinyalleri faz portreleri elde edilmiştir. Şekil 4.16’da PWKS RK4-tabanlı osilatörlerin FPGA üzerinde gerçekleştirilmesinden elde edilen faz portreleri verilmiştir.



Şekil 4.16. RK4-tabanlı PWKS osilatör ünitesi faz portreleri

Tablo 4.1’de Euler, Heun, RK4 ve RK5-Butcher-tabanlı SP ve PW kaotik osilatörlerinin sentezlenmesi işleminin ardından yapılan place&route işleminden elde edilen Xilinx Virtex–6 ailesi XC6VLX550T-2FF1759 FPGA çip istatistikleri verilmiştir. Ayrıca SPKS için tüm kaotik osilatörlerin minimum darbe periyodu 2.564 ns olarak belirlenmiştir. PWKS için Euler ile Heun-tabanlı kaotik osilatörlerin minimum darbe periyodu 2.151 ns ve RK4 ile RK5-Butcher-tabanlı kaotik osilatörlerin minimum darbe periyodu 2.292 ns olmaktadır.

Tablo 4.1. Kaotik osilatörlerin FPGA çip istatistikleri

Kaotik Osilatör	Slice Regs. Sayısı / %	LUTs Sayısı / %	Bonded IOBs Sayısı / %	Maks. Saat Frekansı (MHz)
Euler-Tabanlı SPKS	8,594 / 1.3	7,945 / 2.3	131 / 2.9	390.076
Heun-Tabanlı SPKS	18,935 / 2.8	18,282 / 5.3	131 / 2.9	390.076
RK4-Tabanlı SPKS	40,869 / 5.9	40,156 / 11.7	131 / 2.9	390.076
RK5-Butcher-Tabanlı SPKS	84,017 / 12.2	86,295 / 25.1	131 / 2.9	390.076
Euler-Tabanlı PWKS	9,315 / 1.4	8,376 / 2.4	131 / 2.9	464.688
Heun-Tabanlı PWKS	19,721 / 2.9	16,591 / 4.8	131 / 2.9	464.688
RK4-Tabanlı PWKS	42,021 / 6.1	39,309 / 11.4	131 / 2.9	436.143
RK5-Butcher-Tabanlı PWKS	85.637 / 12.5	85.030 / 24.7	131 / 2.9	436.143

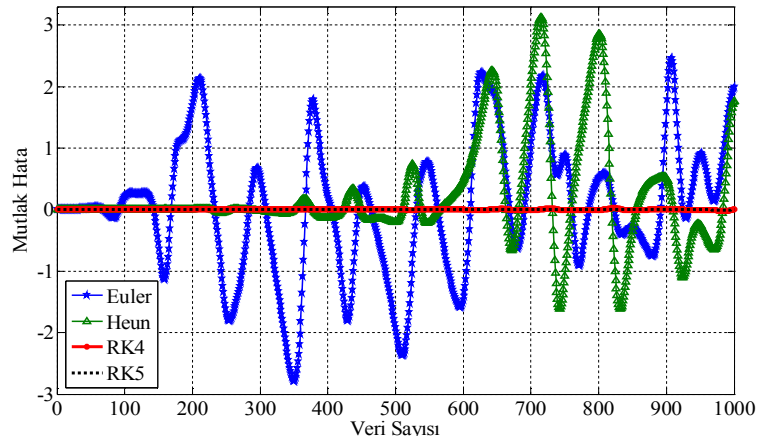
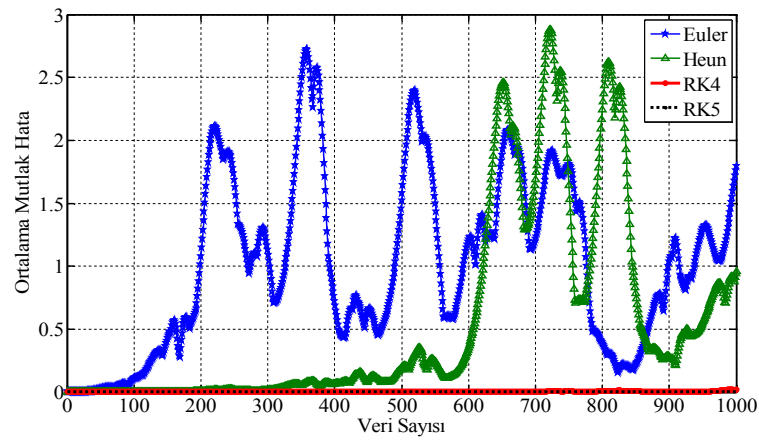
Literatürde sunulan FPGA-tabanlı kaotik osilatör tasarımı çalışmalarından Sadoudi ve arkadaşlarının yaptığı çalışmada Chen kaotik sistemini 32-bit kesirli sayı standardı ile RK4 algoritması kullanılarak Virtex-II çip içeren XCV1000FG456-4 bordu üzerinde gerçekleştirilmiş ve çalışma frekansı 22.85 MHz olarak elde edilmiştir [53]. Virtex-2 FPGA çipinin maksimum çalışma frekansı yaklaşık 400 MHz'dir [127]. Buna göre yapılan çalışmada, çalışma frekansının çipin maksimum çalışma frekansına oranı 1/20 civarındadır. Bu tez çalışmasında sunulan RK4-tabanlı SPKS ve PWKS sistemlerinin çalışma frekansı ise 390 MHz ve 436 MHz olarak elde edilmiştir. Kullanılan Virtex-6 çipinin yaklaşık maksimum çalışma frekansı ise 1000 MHz civarındadır. Sunulan çalışma için bu oran yaklaşık olarak 1/3 civarında olduğundan tez çalışması kapsamında yapılan FPGA-tabanlı kaotik osilatör uygulamasından daha başarılı sonuçlar elde edilmiştir.

De Micco ve arkadaşları yaptıkları çalışmada kayan noktalı sayı IEEE-754 sayı standardı kullanarak RK4 algoritması ile Lorenz kaotik sistemini Altera Cyclone III ailesinin EP3C120F7 FPGA bordu ile test edilmiş ve sistemin çalışma frekansı yaklaşık 1 MHz olarak belirtilmiştir [55]. Cyclone III FPGA çipinin dâhili çalışma frekansı maksimum 100 MHz'dir. Çalışma sonuçlarına göre sistemin çalışma

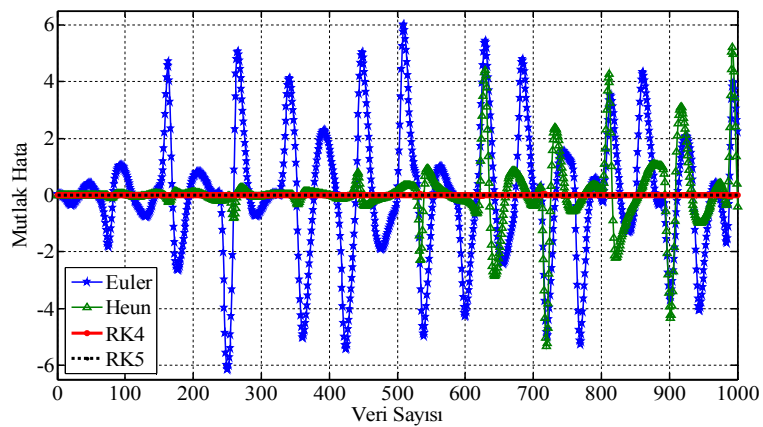
frekansı oldukça düşüktür. Bu tez çalışmasında sunulan RK4-tabanlı SPKS ve PWKS sistemlerinin maksimum çalışma frekansı ise 390 MHz ve 436 MHz olarak elde edilmiştir. Sonuç olarak tez çalışması kapsamında FPGA-tabanlı kaotik osilatör uygulamasından elde edilen sonuçlar daha başarılıdır.

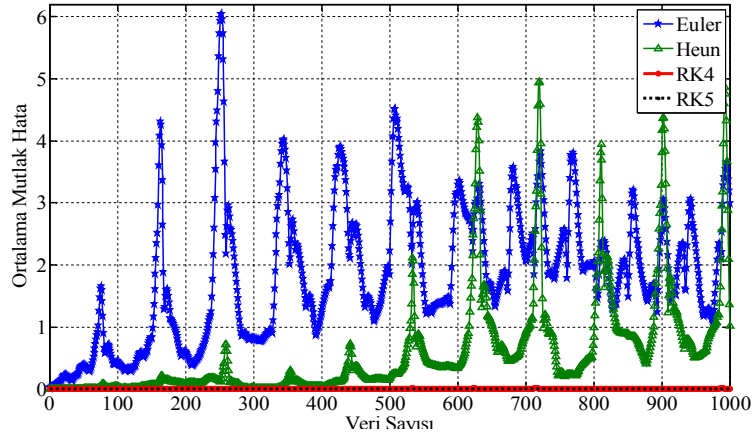
Merah ve arkadaşları yaptıkları çalışmada RK4 algoritmasını kullanarak kesirli sayı standardı ile Lorenz kaotik sistemini Xilinx firmasının Spartan-3 ailesi FPGA çipinde gerçekleştirmişler ve tasarımın çalışma frekansını yaklaşık olarak 18 MHz olarak belirtmişlerdir [54]. Spartan-3 FPGA çipinin dahili maksimum çalışma frekansı 275 MHz'dir [128]. Buna göre yapılan çalışmanın çalışma frekansının çipin maksimum çalışma frekansına oranı 1/15 civarındadır. Bu çalışmada sunulan RK4-tabanlı SPKS ve PWKS sistemlerinin çalışma frekansı ile karşılaştırıldığında bu oran yaklaşık olarak 1/3 civarında olduğundan bu tez çalışması kapsamında yapılan FPGA-tabanlı kaotik osilatör uygulamasından daha başarılı sonuçlar elde edilmiştir.

SPKS ve PWKS için yapılan FPGA-tabanlı tasarımların ve bu tasarımlarda kullanılan Euler, Heun, RK4 ve RK5-Butcher algoritmalarının hata oranlarının belirlenebilmesi amacıyla mutlak hata hesabı yapılmıştır. Yapılan çalışmada referans algoritma olarak RK5-Butcher algoritması seçilerek algoritma ve FPGA-tabanlı modellemelerin ilk 3x1000 veri seti kullanılmıştır. Şekil 4.17'de SPKS z sinyali için FPGA-tabanlı modellemelerden elde edilen mutlak hata ve 4.18'de SPKS x, y ve z sinyalleri ortalama mutlak hata grafiği verilmiştir. Şekil 4.19'da PWKS x sinyali için FPGA-tabanlı modellemelerden elde edilen mutlak hata ve 4.20'de PWKS x, y ve z sinyalleri ortalama mutlak hata sonuç grafiği sunulmuştur. Şekillerden de görüldüğü gibi FPGA tabanlı Euler ve Heun üniteleri oldukça yüksek mutlak hatalar üretmektedir. FPGA tabanlı RK4 ve RK5 algoritmalarının ürettiği sonuçlar ise diğer algoritmaların hata oranlarına göre oldukça düşüktür.

Şekil 4.17. SPKS z sinyali için FPGA-tabanlı mutlak hata sonuçları

Şekil 4.18. SPKS için FPGA-tabanlı ortalama mutlak hata sonuçları

Şekil 4.19. PWKS x sinyali için FPGA-tabanlı mutlak hata sonuçları



Şekil 4.20. PWKS için FPGA-tabanlı ortalama mutlak hata sonuçları

Ayrıca Matlab tabanlı RK5-Butcher algoritması ve FPGA tabanlı ünitelerin ürettiği sonuçların doğruluğunu test etmek amacıyla RMSE (Root Mean Square Error (Ortalama Karesel Hatanın Karekökü)) yöntemi kullanılmıştır. RMSE denklem (4.7)'de görüldüğü gibi hesaplanmaktadır. Burada \hat{x}_i gerçek değerler vektörünü, x_i tahmin edilen değerler vektörünü ve $n=4000$ olmak üzere yapılan değerlendirmede kullanılan örnek adedini ifade etmektedir.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{x}_i - x_i)^2} \quad (4.7)$$

FPGA-tabanlı algoritmalar arasında yapılan RMSE değerlendirmesi Matlab-tabanlı RK4 algoritması referans kabul edilerek yapılmıştır. FPGA-tabanlı algoritmaların ürettiği sonuçlara göre Euler-tabanlı ünite RMSE oranı 1.909, Heun-tabanlı ünite RMSE oranı 1.584 ve RK4-tabanlı ünite RMSE oranı 1.112 olarak hesaplanmıştır.

BÖLÜM 5. FPGA TABANLI YENİ KAOTİK GERÇEK RASGELE SAYI ÜRETEÇLERİNİN TASARIMI VE GERÇEKLENMESİ

Bu bölümde, Euler, Heun, RK4 ve RK5-Butcher algoritmaları kullanılarak FPGA-tabanlı SPKS ve PWKS kaotik sistemleri ile gerçek rasgele sayı üretici tasarımı gerçekleştirilmiştir. Kuantalama işlemleri için üç farklı model sunulmuştur. GRSÜ için yapılan üç farklı FPGA-tabanlı model, daha önceki bölümde sunulan Euler, Heun, RK4 ve RK5-Butcher algoritmaları kullanılarak yapılan tasarımların her birine uygulanmıştır. Genel olarak iki farklı kaotik sistem, kaotik osilatör tasarımında dört ayrı algoritma ve kuantalama işlemi için üç değişik yöntem sunularak toplamda 24 farklı GRSÜ ünitesi tasarlanmıştır. Geliştirilen modeller VHDL kullanılarak kodlanmıştır. FPGA tabanlı modellemelerde 32-bit IEEE 754-1985 kayan noktalı sayı standardı kullanılmıştır. Yapılan tasarımlarda kullanılan kayan noktalı sayı standardına uygun çarpıcı, toplayıcı ve çıkarıcı gibi üniteler, Xilinx ISE Design Tools ile geliştirilen IP Core Generator kullanılarak oluşturulmuştur. Tasarımı yapılan 24 farklı GRSÜ ünitesi Xilinx firmasının ürettiği Virtex-6 ailesinin XC6VLX550T-2FF1759 çipi için sentezlenerek, FPGA çip kaynak kullanımına ve ünitelerin saat hızlarına ait parametrelerin istatistikleri incelenmiştir. GRSÜ ünitelerinin verileri işleme süresi, Xilinx ISE Design Tools 14.2 simülasyon programı kullanılarak elde edilmiştir. Ayrıca tasarlanan üniteler ile ilgili sonuçlar yorumlanmıştır.

5.1. FPGA Tabanlı Gerçek Rasgele Sayı Üretici Tasarımları

FPGA üzerinde GRSÜ tasarımı için Sabit Eşik Değer (SED), Adaptif Eşik Değer (AED) ve Kayan Noktalı Sayı (KNS) tabanlı olmak üzere üç farklı model geliştirilmiştir. Geliştirilen GRSÜ ünitelerinin tümü genel olarak *Kaotik Osilatör Ünitesi*, *Kuantalama ünitesi* ve *Düzeltici Fonksiyon ünitesi* olmak üzere üç bölümden oluşmaktadır.

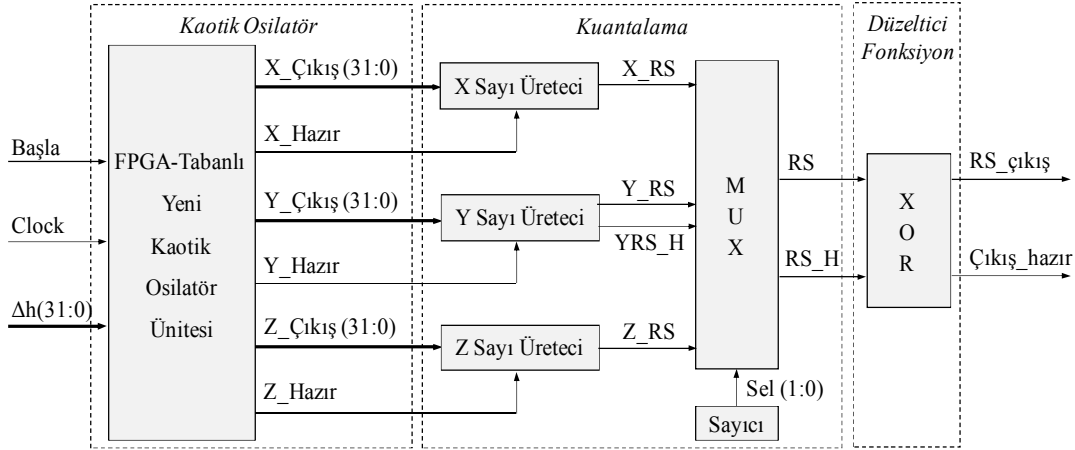
5.1.1. Sabit eşik değeri tabanlı GRSÜ

Şekil 5.1’de *FPGA-Tabanlı Yeni SED GRSÜ* ünitesi blok diyagramı verilmiştir. SED yapısı kullanılarak geliştirilen GRSÜ ünitesi, genel olarak *Kaotik Osilatör Ünitesi*, *Kuantalama* ünitesi ve *Düzeltilici Fonksiyon* ünitesi olmak üzere üç bölümden oluşmaktadır. *FPGA-Tabanlı YKO Ünitesi*’nin yapısı ayrıntılı bir şekilde Bölüm 4.2’de anlatılmıştır. Aşağıda blok şemasında verilen *FPGA-Tabanlı YKO Ünitesi* Euler, Heun, RK4 ve RK5-Butcher tasarımlarının hepsi için geçerlidir. *Kuantalama* ünitesi, üç *Sayı Üreteci*, bir *MUX* ve bir *Sayıcı* ünitesinden oluşmaktadır. *X*, *Y* ve *Z* *Sayı Üreteçleri*’nin yapısında ise 32-bit kayan noktalı sayı standardına uygun giriş sinyalleri ve bu sinyallerin kontrolünü sağlayan *X_Hazır*, *Y_Hazır* ve *Z_Hazır* sinyalleri kullanılmıştır. *Kaotik Osilatör Ünitesi*’nden çıkarak *X*, *Y* ve *Z* *Sayı Üreteçleri*’ne gelen 32-bitlik *X_Çıkış*, *Y_Çıkış* ve *Z_Çıkış* sinyalleri ile σ_x , σ_y ve σ_z sinyalleri arasında Denklem (5.1)’de verilen işlemi gerçekleştirmek amacıyla bir karşılaştırma işlemi yapılmaktadır. Burada σ değeri eşik değeri olarak adlandırılmakta ve bu değer kullanılan kaotik sistemin karakteristiğine göre değişiklik göstermektedir. Bu çalışmada (SPKS için) $\sigma_x=-0.388$, $\sigma_y=-0.387$ ve $\sigma_z=0.154$ olarak alınmıştır. Bu ünite çıkışından elde edilen *X_RS*, *Y_RS* ve *Z_RS* sinyalleri rasgele sayıların taşındığı sinyallerdir. *YRS_H* sinyali ise *Y* *Sayı Üreteci*’nden gelen rasgele sinyalin varlığını göstermektedir. Eğer *YRS_H* sinyali ‘1’ ise rasgele sayı üretilmiş anlamına gelmektedir. Eğer *YRS_H* sinyali ‘0’ ise ünite çıkışından herhangi bir rasgele sayı üretmediği anlamına gelmektedir.

$$RS(x, y, z) = \begin{cases} 0 & x, y, z < \sigma \\ 1 & x, y, z \geq \sigma \end{cases} \quad (5.1)$$

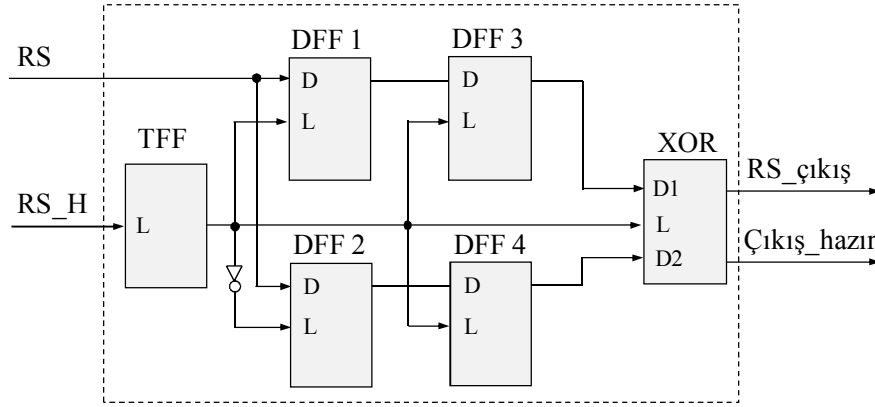
MUX ünitesine gelen *YRS_H* kontrol sinyali ile *X_RS*, *Y_RS* ve *Z_RS* sinyalleri algılanmaktadır. *Sayıcı* birimi ikili sayı sisteminde 0-2 sayıcı olarak görev yapmaktadır. *MUX Ünitesi*, *Sayıcı* biriminden gelen “00”, “01” ve “10” sinyal değerlerine göre sırasıyla *X_RS*, *Y_RS* ve *Z_RS* sinyallerinden birisini seçerek çıkışa aktarmaktadır. Tasarımda *X_RS*, *Y_RS* ve *Z_RS* sinyallerinin hepsinden sonuçların hazır olduğunu gösteren *XRS_H*, *YRS_H* ve *ZRS_H* sinyalleri üretilmektedir. Kontrol sinyali olarak sadece *YRS_H* sinyali kullanılmıştır. Bunun nedeni, tüm *Sayı Üreteci*

üniteleri eşzamanlı bir şekilde paralel olarak çalıştığından *MUX Ünitesi*'nin çalışabilmesi için tek bir denetim sinyali yeterli olmaktadır. Bu nedenle, diğer kontrol sinyalleri kullanılmamıştır. Sunulan bu üç çıkışlı yapı sayesinde ortalama bit hızı yaklaşık olarak üç kat arttırılmaktadır.



Şekil 5.1. FPGA-tabanlı Sabit Eşik Değer GRSÜ blok şeması

FPGA-Tabanlı Yeni SED GRSÜ Ünitesi'nin son bölümünde bulunan *Düzeltici Fonksiyon* ünitesinin açık blok şeması Şekil 5.2'de verilmektedir. Ünite içerisinde bir *TFF*, dört *DFF* ve bir *XOR* ünitesi kullanılmıştır. Ünitelerin tüm girişlerine global saat darbesi uygulanmış ancak şekil karmaşasından kurtarmak amacıyla gösterilmemiştir. *TFF*, *Kuantalama* ünitesi tarafından gönderilen *RS_H* sinyali ile bir toggle flip-flop gibi çalışarak global saat frekansının her yükselen kenarında çıkışı bir önceki durumun tersi olacak şekilde değiştirmektedir. *DFF*'ler data flip-flop gibi çalışarak, *TFF* ünitesi tarafından gönderilen sinyalin durumuna göre *Kuantalama* ünitesi tarafından gönderilen *RS* sinyalini tutmakta ve bir saat darbesi sonunda çıkışa aktarmaktadır. *XOR* birimi, içerisinde bir XOR kapısı bulunmaktadır. *TFF* tarafından gönderilen sinyalin durumuna göre *DFF 3* ve *DFF 4* ünitesinden *D1* ve *D2* girişlerine gelen sinyalleri XOR işlemine tabi tutarak bu değeri *RS_cikis* sinyali olarak çıkışa aktarmaktadır. *RS_cikis* sinyali üretildiği durumlarda ise *Çıkış_hazır* sinyali '1' değerini almaktadır. Diğer durumlarda *Çıkış_hazır* sinyali '0' olmaktadır.

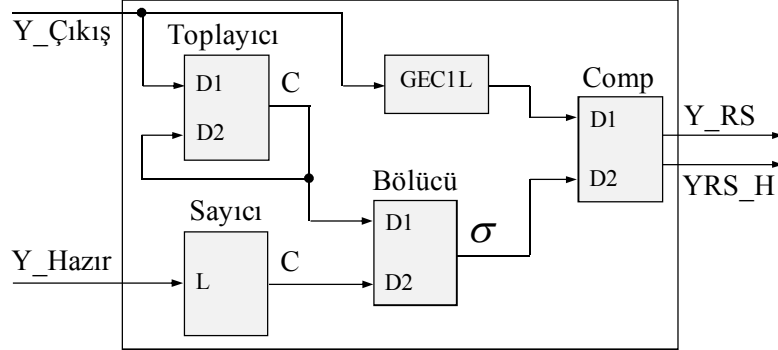


Şekil 5.2. Düzeltilici Fonksiyon ünitesi blok şeması

5.1.2. Adaptif eşik değer tabanlı GRSÜ

FPGA-Tabanlı Yeni AED GRSÜ ünitesi genel blok diyagramı üç bölümden oluşmaktadır. Ünitenin blok diyagramı Şekil 5.1’de verilen *FPGA-Tabanlı Yeni SED GRSÜ* ünitesi blok diyagramı ile aynıdır. Ancak *Kuantalama* ünitesi içerisinde bulunan adaptif yapıdaki *Sayı Üreteci* tasarımları farklılık göstermektedir. Şekil 5.3’te *Sayı Üreteci* ünitesi blok şeması verilmiştir. Blok şemasında verilen sinyaller Y *Sayı Üreteci*’ne göre verilmiştir. X *Sayı Üreteci* ve Z *Sayı Üreteci* üniteleri de aynı yapıda tasarlanmıştır. Tasarımda kayan noktalı sayı standardına uygun *Toplayıcı*, *Sayıcı*, *GECIL*, *Bölücü* ve *Comp* birimleri kullanılmıştır. *Toplayıcı* birimi kaotik osilatör ünitesinden gelen ve y sinyal değerini taşıyan $Y_Çıkış$ sinyalini, kaotik osilatör her sinyal ürettiğinde toplamakta ve çıkışa aktarmaktadır. *Sayıcı* birimi kaotik osilatör ünitesinden gelen ve y değerinin üretildiğini gösteren $Y_Hazır$ sinyalini kullanarak osilatörün kaç değer ürettiğini saymakta ve bu değeri *Bölücü* birimine göndermektedir. *Bölücü* birimi, *Toplayıcı* ve *Sayıcı* biriminden gelen sinyalleri kullanarak bölme işlemi yapmakta ve anlık ortalama değerini hesaplamaktadır. Bu yöntem ile eşik değerleri sabit olarak alınmamakta sürekli olarak anlık ortalama değerler hesaplanarak adaptif eşik değer olan σ_y bulunmaktadır. *GECIL* birimi, sistemin senkronizasyonun sağlanması amacıyla geliştirilmiş tek saat darbelik bir sinyal geciktirme ünitesidir. *GECIL* biriminin çıkışındaki $Y_Çıkış$ sinyali ve *Bölücü* birimi çıkışındaki σ_y değerleri *Comp* birimine gönderilmekte ve burada Denklem 5.1’de verilen işlem gerçekleştirilerek karşılaştırma yapılmaktadır. Eğer $Y_Çıkış$ sinyali, σ_y değerinden büyükse çıkış sinyali olan Y_RS sinyaline ‘1’,

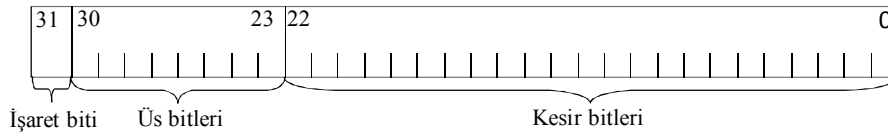
değilse '0' sinyali gönderilmektedir. YRS_H sinyali ise Y_RS sinyalinin hazır olduğunu göstermektedir.



Şekil 5.3. Adaptif Sayı Üreteci ünitesi blok şeması

5.1.3. Kayan noktalı sayı tabanlı GRSÜ

Kayan Noktalı Sayı (KNS) standardı gerçel sayıların ifade edilmesi için kullanılan yöntemlerden birisidir. Günlük hayatta kullanılan sayı değerleri, sonsuza kadar giderken, simülasyon çalışmalarında ve donanımsal uygulamalarda, donanımsal kısıtlamalardan dolayı tüm sayı değerlerinin gösterilmesi mümkün olamamaktadır. Bununla birlikte yazılımsal ve donanımsal uygulamalarda sonsuza kadar giden değerler, donanım platformunun kapasitesine bağlı olarak yaklaşık değerlerle temsil edilmektedirler. Belirtilen donanımsal sınırlamaların etkisini en aza indirgeyen, ifade edilmek istenen sayıların en hassas ve yüksek miktarda ifade edilmesini sağlayan sayı sistemlerinden birisi de kayan noktalı sayı standardıdır. Kayan noktalı sayıların her farklı donanıma uyumlu bir gösterime sahip olması için IEEE-754-1985 standardı geliştirilmiştir. Bu standarda göre kayan noktalı sayıların gösterimi tek duyarlı (32-bit) ve çift duyarlı (64-bit) olmak üzere iki farklı şekilde gerçekleştirilmektedir. Şekil 5.4'te 32-bit tek duyarlı IEEE 754-1985 KNS standardı gösterimi verilmiştir. Bu standartta 31'inci bit işaret biti olarak isimlendirilmektedir. İşaret (significant) biti değeri eğer '0' ise sayı pozitif, '1' ise sayı negatif olmaktadır. İkinci kısım olan 8-bit üs (exponent) bitleri, sayının üstel kısmını belirtmek için kullanılmaktadır. Kesirli (mantissa) bitler ise KNS standardında sayının kesirli kısmını ifade etmektedir [129].



Şekil 5.4. 32-bit IEEE 754-1985 kayan noktalı sayı standardı gösterimi

KNS standardına göre sayıların ifade edilmesi Denklem (5.2)'de verildiği gibi gerçekleştirilmektedir. Burada v hesaplanacak onluk sayı değeri, $sign$ işaret biti değeri, j kesir bitlerinin sayısı (32-bit için $j=23$), b kesirli bitleri ve exp üs bitlerinin değerini ifade etmektedir [129].

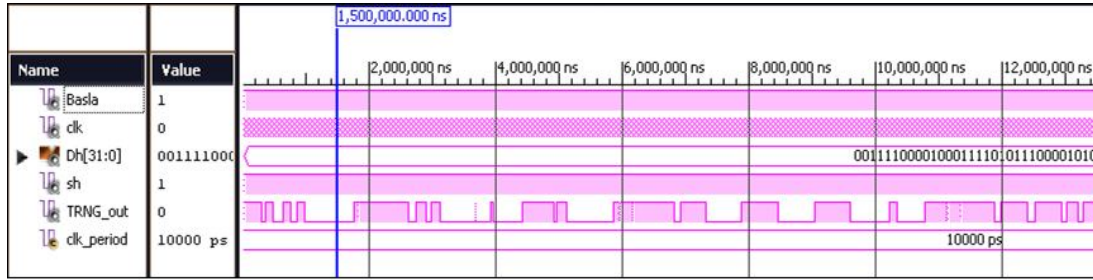
$$v = (-1)^{sign} \left(1 + \sum_{i=1}^j b_{23-i} 2^{-i} \right) x 2^{(exp-127)} \quad (5.2)$$

KNS Tabanlı GRSÜ ünitesi blok diyagramı, Şekil 5.1'de verilen *FPGA-Tabanlı Yeni SED GRSÜ* ünitesi blok diyagramı ile aynıdır. Fakat *Kuantalama* ünitesi içerisinde bulunan *Sayı Üreteci* tasarımları farklılık göstermektedir. KNS formatında kesirli kısımda en düşük değerlikli veya en yüksek hassasiyete sahip olan bit ise 0'ncı bit (b_0) olmaktadır. Bu bit değeri sayı değerinin çok az değişmesiyle değişebilmektedir. KNS Tabanlı GRSÜ yapısı kaotik osilatör ünitesinin her bir ürettiği kayan noktalı sayı standardındaki 32-bitlik sayının kesirli kısmındaki bitlerin alınması ile gerçekleştirilmiştir.

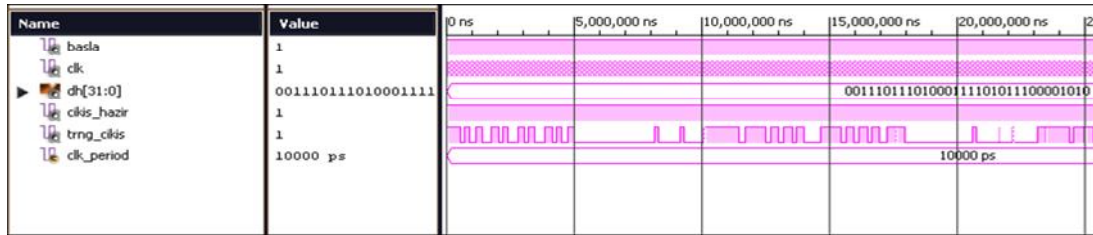
5.2. Gerçek Rasgele Sayı Üreteçlerinin FPGA Üzerinde Gerçeklenmesi

Bu bölümde SPKS ve PWKS kaotik sistemleri, Euler, Heun, RK4 ve RK5-Butcher algoritmaları kullanılarak SED, AED ve KNS tabanlı yapılar ile tasarımı yapılan 24 farklı GRSÜ ünitesi Xilinx firmasının ürettiği Virtex-6 ailesinin XC6VLX550T-2FF1759 çipi için sentezlenerek, FPGA çip kaynak kullanımına ve ünitelerin saat hızlarına ait parametrelerin istatistikleri incelenmiştir. Tasarımı yapılan 24 farklı yapıdaki GRSÜ ünitelerinin verileri işleme süresi, Xilinx ISE Design Tools 14.2 simülasyon programı kullanılarak elde edilmiştir. Tasarlanan ünitelerin tümü iş hattı tabanlı olarak çalışmaktadır.

Şekil 5.5'te *SED-Tabanlı*, Şekil 5.6'da *AED-Tabanlı* ve Şekil 5.7'de *KNS-Tabanlı* *GRSÜ* üniteleri için Xilinx ISE Simülöründe elde edilen sonuçlar verilmiştir. Xilinx ISE Design Tools 14.2 simülasyon programı kullanılarak yapılan tasarımları test etmek amacıyla VHDL dilinde bir testbench yapısı oluşturulmuştur. *GRSÜ* ünitelerinin en üst genel yapılarında bulunan 1-bit *Basla* sinyaline '1' ve 32-bit Δh sinyaline 0.01 değerleri daha kolay okunabilmesini sağlamak amacıyla onaltılık formatta verilmiştir.



Şekil 5.5. *SED-Tabanlı* *GRSÜ* ünitesi Xilinx ISE Simülöründe sonuçları



Şekil 5.6. *AED-Tabanlı* *GRSÜ* ünitesi Xilinx ISE Simülöründe sonuçları



Şekil 5.7. *KNS-Tabanlı* *GRSÜ* ünitesi Xilinx ISE Simülöründe sonuçları

Tablo 5.1'de *SED-Tabanlı* *GRSÜ* üniteleri için FPGA çip istatistikleri sunulmuştur. Tabloda sunulan çip istatistikleri kaotik osilatör, kuantalama ve düzeltici fonksiyon ünitelerinden oluşan tüm *GRSÜ* sistemi için elde edilen verilerdir. Görüldüğü gibi *SED-Tabanlı* *GRSÜ* ünitelerinin maksimum saat frekansları ve çip kullanım oranları,

AED-Tabanlı GRSÜ ünitelerinin maksimum saat frekanslarına göre yüksek ve çip kullanım oranları da düşüktür. *KNS-Tabanlı GRSÜ* ünitelerine göre ise maksimum saat frekansları düşük ve çip kullanım oranları da yüksektir.

Yapılan simülasyon çalışmalarında rasgelelik testlerinin yapılabilmesi amacıyla tasarlanan Euler-Tabanlı SPKS GRSÜ ünitesinden her 110 ns aralıklarla 20,000 bit veri alınarak bir dosyaya kaydedilmiştir. Çalışmada kullanılan 20,000 bit verinin FPGA üzerinde elde edilebilmesi için yaklaşık 2.2 ms süre geçmektedir. Sonuç olarak, Euler algoritması kullanılarak modellenen SPKS *SED-Tabanlı GRSÜ* ünitelerinin yaklaşık bit üretim hızı 9.09 Kbit/s olarak belirlenmiştir. Tasarımın rasgele sayı üretimi performansı (b/s) oldukça düşüktür. Bunun en büyük nedeni, *SED-Tabanlı Kuantalama/Örnekleme* ünitesinin yapısından kaynaklanmaktadır. Çünkü *SED-Tabanlı Kuantalama* yönteminde eşik değeri sabittir. Bunun sonucunda, FPGA-tabanlı kaotik osilatörün ürettiği sinyallerin çalışma frekansı yüksek olsa da kaotik sinyalin eşik değeri ile karşılaştırma işleminden elde edilen 0/1 değerlerinin değişimi çok yavaş kalmaktadır.

Tablo 5.1. *SED-Tabanlı GRSÜ* ünitelerinin FPGA çip istatistikleri

Kaotik Osilatör	Nümerik Algoritma	Slice Regs. Sayısı / %	LUTs Sayısı / %	Bonded IOBs Sayısı / %	Maks. Saat Frekansı (MHz)
SPKS	Euler	8,759 / 1.3	8,522 / 2.5	36 / 4.3	339.179
PWKS	Euler	9,323 / 1.4	8,417 / 2.4	36 / 4.3	399.383
SPKS	Heun	18,941 / 2.8	18,318 / 5.5	36 / 4.3	339.179
PWKS	Heun	20,071 / 2.9	18,012 / 5.2	36 / 4.3	399.383
SPKS	RK4	41,857 / 6.1	43,364 / 12.6	36 / 4.3	339.179
PWKS	RK4	42,937 / 6.2	42,587 / 12.3	36 / 4.3	373.094
SPKS	RK5-B	83,999 / 12.2	86,302 / 25.1	36 / 4.3	339.179
PWKS	RK5-B	85,645 / 12.5	85,071 / 24.8	36 / 4.3	373.094

Tablo 5.2'de *AED-Tabanlı GRSÜ* üniteleri için FPGA çip istatistikleri verilmiştir. Tabloda sunulan çip istatistikleri kaotik osilatör, kuantalama ve düzeltici fonksiyon ünitelerinden oluşan tüm GRSÜ sistemi için elde edilen verilerdir. Tasarlanan ünitelerin hepsi iş hattı tabanlı olarak çalışmaktadır. *AED-Tabanlı GRSÜ* ünitelerinin

maksimum saat frekansları ve çip kullanım oranları, *SED-Tabanlı GRSÜ* ünitelerinin maksimum saat frekanslarına yakın olup, çip kullanım oranları ise daha yüksektir. *KNS-Tabanlı GRSÜ* ünitelerine göre ise maksimum saat frekansları düşük iken, çip kullanım oranları da yüksektir. Yapılan simülasyon çalışmalarında rasgelelik testlerinin yapılabilmesi ve testlerden başarı sağlanabilmesi amacıyla tasarlanan Euler-Tabanlı SPKS GRSÜ ünitesinden her 80 ns aralıklarla 20,000 bit veri alınarak bir dosyaya kaydedilmiştir. Çalışmada kullanılan 20,000 bit verinin FPGA üzerinde elde edilebilmesi için yaklaşık 1.6 ms süre geçmektedir. Sonuç olarak Euler algoritması kullanılarak modellenen SPKS *AED-Tabanlı GRSÜ* ünitelerinin yaklaşık bit üretim hızı 10.22 Kbit/s olarak belirlenmiştir. Tasarımın rasgele sayı üretimi performansı (b/s) oldukça düşük olmasının en büyük nedeni *AED-Tabanlı Kuantalama/Örnekleme* ünitesinin yapısından kaynaklanmaktadır. FPGA üzerinde kaotik sinyalin değişimi çok hızlı olsa da kaotik sinyalin karşılaştırma işleminden elde edilen 0/1 değerlerinin değişimi *SED-Tabanlı GRSÜ* ünitesine göre yüksek olmasına rağmen istenilen bit üretim hızına çıkılamamakta ve bit üretim hızı yavaş kalmaktadır.

Tablo 5.2. AED-Tabanlı GRSÜ ünitelerinin FPGA çip istatistikleri

Kaotik Osilatör	Nümerik Algoritma	Slice Regs. Sayısı / %	LUTs Sayısı / %	Bonded IOBs Sayısı / %	Maks. Saat Frekansı (MHz)
SPKS	Euler	8,987 / 1.0	12,466 / 3.0	36 / 4.3	339.179
PWKS	Euler	9,717 / 1.4	13,106 / 3.8	36 / 4.3	394.958
SPKS	Heun	19,335 / 2.8	23,105 / 6.7	36 / 4.3	339.179
PWKS	Heun	20,465 / 3.0	22,702 / 6.6	36 / 4.3	394.958
SPKS	RK4	42,251 / 6.1	48,151 / 14.0	36 / 4.3	339.179
PWKS	RK4	43,331 / 6.3	47,179 / 13.7	36 / 4.3	371.284
SPKS	RK5-B	84,393 / 12.3	91,089 / 26.5	36 / 4.3	339.179
PWKS	RK5-B	86,039 / 12.5	89,663 / 26.1	36 / 4.3	371.284

Tablo 5.3'te *KNS-Tabanlı GRSÜ* üniteleri için FPGA çip istatistikleri verilmiştir. Tasarlanan ünitelerin tümü iş hattı tabanlı olarak çalışmaktadır. *KNS-Tabanlı GRSÜ* ünitelerinin maksimum saat frekansları ve çip kullanım oranları, *SED-Tabanlı GRSÜ* ünitelerinin maksimum saat frekanslarına göre yüksek ve çip kullanım oranları ise düşüktür. *AED-Tabanlı GRSÜ* ünitelerine göre ise maksimum saat frekansları yüksek

ve çip kullanım oranları düşüktür. Rasgelelik testlerinin yapılabilmesi amacıyla tasarlanan Euler-Tabanlı SPKS GRSÜ ünitesinden rasgelelik oranının artırılması amacı ile her 30 ns aralıklarla 20,000 bit veri alınarak bir dosyaya kaydedilmiştir. İstatistiksel testlerin yapılabilmesi için gerekli 20 Kbit verinin elde edilebilmesi için 0.6 ms süre geçmektedir. Bu çalışma için *KNS-Tabanlı GRSÜ* ünitelerinin yaklaşık bit üretim hızı 113.07 Mbit/s olarak belirlenmiştir.

Tablo 5.3. KNS-Tabanlı GRSÜ ünitelerinin FPGA çip istatistikleri

Kaotik Osilatör	Nümerik Algoritma	Slice Regs. Sayısı / %	LUTs Sayısı / %	Bonded IOBs Sayısı / %	Maks. Saat Frekansı (MHz)
SPKS	Euler	8,765 / 1.3	8,491 / 2.5	36 / 2.9	339.179
PWKS	Euler	9,329 / 1.4	8,384 / 2.4	36 / 4.3	401.290
SPKS	Heun	18,947 / 2.8	18,287 / 5.3	36 / 4.3	339.179
PWKS	Heun	20,077 / 2.9	17,979 / 5.2	36 / 4.3	401.290
SPKS	RK4	41,863 / 6.1	43,333 / 12.6	36 / 4.3	339.179
PWKS	RK4	42,943 / 6.2	42,554 / 12.4	36 / 4.3	373.094
SPKS	RK5-B	84,005 / 12.2	86,271 / 25.1	36 / 4.3	339.179
PWKS	RK5-B	42,943 / 6.2	42,554 / 12.4	36 / 4.3	373.094

Yapılan çalışmalarda KNS-Tabanlı yöntem kullanılarak yapılan 8 farklı GRSÜ ünitelerinin bit üretim hızları incelenmiştir. Elde edilen bit üretim hızları Tablo 5.4'te verilmiştir. Elde edilen sonuçlardan da görüldüğü üzere yapılan tasarımların rasgele bit üretim performansı *SED-Tabanlı GRSÜ* ve *AED-Tabanlı GRSÜ* ünitelerine göre oldukça yüksektir. *KNS-Tabanlı GRSÜ* ünitelerinin maksimum saat frekansları ile bit üretim hızları oldukça yüksek ve çip kullanım oranları düşük olduğundan bu aşamadan sonraki çalışmalarda *KNS-Tabanlı GRSÜ* üniteleri kullanılacaktır.

Tablo 5.4. *KNS-Tabanlı GRSÜ* bit üretim hızları

Kaotik Osilatör	Nümerik Algoritma	Bit Üretim Hızı (Mbit/s)
SPKS	Euler	113.07
PWKS	Euler	132.80
SPKS	Heun	84.80
PWKS	Heun	99.60
SPKS	RK4	67.84
PWKS	RK4	53.30
SPKS	RK5-B	67.84
PWKS	RK5-B	53.30

Tablo 5.5’te literatürde çeşitli yaklaşımlar kullanılarak geliştirilen GRSÜ çalışmaları ile ilgili bilgiler verilmiştir. Literatürde yapılan çalışmalar ile tez çalışmasından elde edilen maksimum bit üretim hızları karşılaştırılmıştır. Sonuçlardan da görüldüğü üzere tez çalışmasından elde edilen maksimum bit üretim hızları diğer çalışmalara göre daha başarılıdır.

Tablo 5.5. Literatürde sunulan GRSÜ çalışmaları

Yapılan Çalışma	Teknoloji	Teknoloji Özelliği	Osilatör	Maksimum Çalışma Frekansı (MHz)	Maksimum Bit Üretim Hızı (Mbit/s)
Wieczorek	FPGA	Spartan-3E	FF	50	5.0
Çiçek	FPGA	Virtex-5	Kaotik	75	3.2
Danger	FPGA	Stratix	Gecikme Zinciri	20	20
Fischer	FPGA	Stratix	PLL	250	1.0
Schellekens	FPGA	Virtex-2	Ring	40	2.5
Dichtl	FPGA	Spartan-3	Ring	--	12.5
Istvan	FPGA	Spartan-3E	Jitter	50	1.92
Çiçek	CMOS	0.25 μ m	Lojistik Harita	16	1.5
Pareschi	CMOS	0.18 μ m	Kaotik	25	100
Özoğuz	CMOS	0.35 μ m	Kaotik	25	2.3
Koyuncu	FPGA	Virtex-6	Kaotik	401	132

BÖLÜM 6. FPGA TABANLI YENİ KAOTİK GERÇEK RASGELE SAYI ÜRETEÇLERİNİN İSTATİSTİKSEL RASGELELİK TESTLERİ VE SONUÇLARI

Kriptolojik uygulamalarda kullanılmak üzere geliştirilen sözde rasgele ve gerçek rasgele sayı üreteçlerinin rasgeleliklerinin ve istatistiksel özelliklerinin incelenmesi ve test edilmesi gerekmektedir. Bu amaçla literatürde geliştirilmiş çeşitli istatistiksel testler bulunmaktadır. Bu bölümde, geliştirilen FPGA-tabanlı yeni kaotik GRSÜ üniteleri, kriptolojik uygulamalarda güvenli bir şekilde kullanılabilmesi amacıyla literatürde geliştirilmiş olan istatistiksel testlere tabi tutulmuştur. Bu amaçla Bölüm 2.5'te kapsamlı bir biçimde sunulan FIPS-140-1 ve NIST-800-22 testleri kullanılmıştır.

6.1. FPGA-Tabanlı Kaotik GRSÜ FIPS-140-1 Testleri ve Sonuçları

FIPS-140-1 testinde Monobit, Poker, Koşu ve Uzun Koşu olmak üzere dört test bulunmaktadır. Tasarlanan SPKS ve PWKS osilatörlerinden yararlanılarak geliştirilen GRSÜ ünitelerinden 20,000 bit veri alınarak bir dosyaya kaydedilmiş ve FIPS-140-1 testlerine tabi tutulmuştur. Tablo 6.1'de FPGA-tabanlı SPKS osilatörü kullanılarak gerçekleştirilen GRSÜ ünitelerinin FIPS-140-1 testlerinden elde edilen sonuçları verilmiştir. SPKS-tabanlı GRSÜ üniteleri FIPS-140-1 testlerindeki dört testten de başarılı olmuştur. Koşu testinde sadece blok uzunluğu 1 için sonuçlar verilmiştir.

Tablo 6.1. SPKS ile gerçekleştirilen GRSÜ üniteleri FIPS-140-1 test sonuçları

FIPS-140-1 Testleri	Test başarı kriterleri	Euler-Tabanlı	Heun-Tabanlı	RK4-Tabanlı	RK5-Butcher Tabanlı
Monobit Testi	$9654 < n < 10346$	10006	9966	9932	9994
Poker Testi	$1.03 < X < 57.4$	12.1	11.9	14.5	21.1
Koşu Testi	$2267 \leq x \leq 2733$	2494	2488	2401	2408
Uzun Koşu Testi	$34 > \text{Koşu}$	$34 > 24$	$34 > 31$	$34 > 11$	$34 > 18$

Tablo 6.2’de tasarımı yapılan FPGA-tabanlı PWKS osilatörü kullanılarak gerçekleştirilen GRSÜ ünitelerinin FIPS-140-1 testlerinden elde edilen sonuçları verilmiştir. Koşu testinde çok fazla test olduğundan burada sadece blok uzunluğu 1 için sonuçlar verilmiştir. PWKS-tabanlı GRSÜ üniteleri FIPS-140-1 testlerindeki dört testten de başarılı olmuştur.

Tablo 6.2. PWKS ile gerçekleştirilen GRSÜ üniteleri FIPS-140-1 test sonuçları

FIPS-140-1 Testleri	Test başarı kriterleri	Euler-Tabanlı	Heun-Tabanlı	RK4-Tabanlı	RK5-Butcher Tabanlı
Monobit Testi	$9654 < n < 10346$	9949	9973	9959	9991
Poker Testi	$1.03 < X < 57.4$	12.4	24.4	17.9	17.9
Koşu Testi	$2267 \leq x \leq 2733$	2476	2450	2408	2546
Uzun Koşu Testi	$34 > \text{Koşu}$	$34 > 12$	$34 > 15$	$34 > 17$	$34 > 13$

6.2. FPGA-Tabanlı Kaotik GRSÜ NIST-800-22 Testleri ve Sonuçları

Uluslararası geçerliliğe sahip diğer bir istatistiksel test olan NIST Test Suite’de 15 test bulunmaktadır. Random-Excursions ve Random Excursions Variant testleri için genel olarak 1 milyon bit veriye ihtiyaç duyulmaktadır. Bu nedenle, testlerin yapılabilmesi amacıyla genel olarak her bir tasarım için 1 M-bitlik veriler toplanarak dosyaya kaydedilmiştir. Ardından bit dosyası NIST Test Suite’de bulunan 15 teste tabi tutulmuştur. Tablo 6.3’te FPGA üzerinde gerçekleştirilen SPKS ile Euler algoritması kullanılarak GRSÜ ünitesi için yapılan testlerin sonuçları verilmiştir. Bütün testlerde sonuçların başarılı kabul edilebilmesi için rastgeleliğin ölçüsü olarak kabul edilen *P-değeri*nin 0.001 ’den büyük olması gerekmektedir. Sonuçlardan da görüldüğü üzere, *P-değeri* ≥ 0.001 olduğundan elde edilen diziler rasgele olarak kabul edilmektedir. Rasgele gezinimler testi için x değişkeni $-4 \leq x \leq 1$ ve $4 \leq x \leq 1$ değerlerini alabilmekte ve testte toplam 8 sonuç üretilmektedir. Tablo boyutlarının çok fazla büyümemesi amacıyla bu test için sadece $x = -4$ testinin sonucu verilmektedir. Ayrıca rasgele gezinimler değişken testi için x değişkeni $-9 \leq x \leq 1$ ve $9 \leq x \leq 1$ değerlerini alabilmekte ve bu testte toplam 18 test sonucu üretilmektedir. Tablo boyutlarının çok fazla büyümemesi amacıyla tüm NIST-800-22 testlerinde rasgele gezinimler değişken testi için sadece $x = -9$ testinin sonucu verilmektedir.

Tablo 6.3. SPKS Euler algoritması tabanlı GRSÜ ünitesi NIST testi sonuçları

İstatistiksel Testler	P-değeri	Sonuç
Frequency (Monobit) Test	0.0849	Başarılı
Block-Frequency Test	0.7819	Başarılı
Cumulative-Sums Test	0.0312	Başarılı
Runs Test	0.5842	Başarılı
Longest-Run Test	0.6455	Başarılı
Binary Matrix Rank Test	0.2804	Başarılı
Discrete Fourier Transform Test	0.9586	Başarılı
Non-Overlapping Templates Test	0.5337	Başarılı
Overlapping Templates Test	0.0019	Başarılı
Maurer's Universal Statistical Test	0.6561	Başarılı
Approximate Entropy Test	0.4578	Başarılı
Random-Excursions Test	0.9895	Başarılı
Random-Excursions Variant Test	0.5495	Başarılı
Serial Test-1	0.8222	Başarılı
Serial Test-2	0.9137	Başarılı
Linear-Complexity Test	0.3349	Başarılı

Tablo 6.4'te FPGA üzerinde gerçekleştirilen PWKS ile Euler algoritması kullanılarak GRSÜ ünitesi için yapılan testlerin sonuçları verilmiştir. Test sonuçlarından da görüldüğü gibi $P\text{-değeri} \geq 0.001$ olduğundan elde edilen diziler rasgele olarak kabul edilmektedir.

Tablo 6.4. PWKS Euler algoritması tabanlı GRSÜ ünitesi NIST testi sonuçları

İstatistiksel Testler	P-değeri	Sonuç
Frequency (Monobit) Test	0.0957	Başarılı
Block-Frequency Test	0.1504	Başarılı
Cumulative-Sums Test	0.1531	Başarılı
Runs Test	0.6332	Başarılı
Longest-Run Test	0.7264	Başarılı
Binary Matrix Rank Test	0.4926	Başarılı
Discrete Fourier Transform Test	0.7830	Başarılı
Non-Overlapping Templates Test	0.8679	Başarılı
Overlapping Templates Test	0.5866	Başarılı
Maurer's Universal Statistical Test	0.4181	Başarılı
Approximate Entropy Test	0.0035	Başarılı
Random-Excursions Test	0.9829	Başarılı
Random-Excursions Variant Test	0.2860	Başarılı
Serial Test-1	0.4767	Başarılı
Serial Test-2	0.2312	Başarılı
Linear-Complexity Test	0.0088	Başarılı

Tablo 6.5'te FPGA üzerinde gerçekleştirilen SPKS ile Heun algoritması kullanılarak GRSÜ ünitesi için yapılan testlerin sonuçları ve Tablo 6.6'da PWKS ile Heun algoritması kullanılarak GRSÜ ünitesi için yapılan testlerin sonuçları verilmiştir. Test sonuçlarından da görüldüğü gibi $P\text{-değeri} \geq 0.001$ olduğundan elde edilen diziler rasgele olarak kabul edilmektedir.

Tablo 6.5. SPKS Heun algoritması tabanlı GRSÜ ünitesi NIST testi sonuçları

İstatistiksel Testler	P-değeri	Sonuç
Frequency (Monobit) Test	0.7810	Başarılı
Block-Frequency Test	0.5564	Başarılı
Cumulative-Sums Test	0.8546	Başarılı
Runs Test	0.2135	Başarılı
Longest-Run Test	0.1759	Başarılı
Binary Matrix Rank Test	0.0207	Başarılı
Discrete Fourier Transform Test	0.0116	Başarılı
Non-Overlapping Templates Test	0.7969	Başarılı
Overlapping Templates Test	0.1186	Başarılı
Maurer's Universal Statistical Test	0.4559	Başarılı
Approximate Entropy Test	0.1237	Başarılı
Random-Excursions Test	0.4238	Başarılı
Random-Excursions Variant Test	0.6989	Başarılı
Serial Test-1	0.2399	Başarılı
Serial Test-2	0.1800	Başarılı
Linear-Complexity Test	0.5491	Başarılı

Tablo 6.6. PWKS Heun algoritması tabanlı GRSÜ ünitesi NIST testi sonuçları

İstatistiksel Testler	P-değeri	Sonuç
Frequency (Monobit) Test	0.3624	Başarılı
Block-Frequency Test	0.0055	Başarılı
Cumulative-Sums Test	0.4462	Başarılı
Runs Test	0.7352	Başarılı
Longest-Run Test	0.7541	Başarılı
Binary Matrix Rank Test	0.5124	Başarılı
Discrete Fourier Transform Test	0.2814	Başarılı
Non-Overlapping Templates Test	0.0952	Başarılı
Overlapping Templates Test	0.8135	Başarılı
Maurer's Universal Statistical Test	0.0145	Başarılı
Approximate Entropy Test	0.3453	Başarılı
Random-Excursions Test	0.0603	Başarılı
Random-Excursions Variant Test	0.5432	Başarılı
Serial Test-1	0.2906	Başarılı
Serial Test-2	0.1717	Başarılı
Linear-Complexity Test	0.8670	Başarılı

Tablo 6.7’de FPGA üzerinde gerçekleştirilen SPKS ile RK4 algoritması kullanılarak GRSÜ ünitesi için yapılan testlerin sonuçları ve Tablo 6.8’de PWKS ile RK4 algoritması kullanılarak GRSÜ ünitesi için yapılan testlerin sonuçları verilmiştir. Test sonuçlarından da görüldüğü gibi $P\text{-değeri} \geq 0.001$ olduğundan elde edilen diziler rasgele olarak kabul edilmektedir.

Tablo 6.7. SPKS RK4 algoritması tabanlı GRSÜ ünitesi NIST testi sonuçları

İstatistiksel Testler	P-değeri	Sonuç
Frequency (Monobit) Test	0.8693	Başarılı
Block-Frequency Test	0.0175	Başarılı
Cumulative-Sums Test	0.9862	Başarılı
Runs Test	0.0086	Başarılı
Longest-Run Test	0.2408	Başarılı
Binary Matrix Rank Test	0.3691	Başarılı
Discrete Fourier Transform Test	0.2229	Başarılı
Non-Overlapping Templates Test	0.0265	Başarılı
Overlapping Templates Test	0.1241	Başarılı
Maurer's Universal Statistical Test	0.9029	Başarılı
Approximate Entropy Test	0.5156	Başarılı
Random-Excursions Test	0.6304	Başarılı
Random-Excursions Variant Test	0.8238	Başarılı
Serial Test-1	0.4904	Başarılı
Serial Test-2	0.4748	Başarılı
Linear-Complexity Test	0.9699	Başarılı

Tablo 6.8. PWKS RK4 algoritması tabanlı GRSÜ ünitesi NIST testi sonuçları

İstatistiksel Testler	P-değeri	Sonuç
Frequency (Monobit) Test	0.9188	Başarılı
Block-Frequency Test	0.2053	Başarılı
Cumulative-Sums Test	0.6241	Başarılı
Runs Test	0.0012	Başarılı
Longest-Run Test	0.2061	Başarılı
Binary Matrix Rank Test	0.4456	Başarılı
Discrete Fourier Transform Test	0.0302	Başarılı
Non-Overlapping Templates Test	0.2590	Başarılı
Overlapping Templates Test	0.0353	Başarılı
Maurer's Universal Statistical Test	0.2987	Başarılı
Approximate Entropy Test	0.2963	Başarılı
Random-Excursions Test	0.7880	Başarılı
Random-Excursions Variant Test	0.1915	Başarılı
Serial Test-1	0.1092	Başarılı
Serial Test-2	0.0941	Başarılı
Linear-Complexity Test	0.9733	Başarılı

Tablo 6.9’da FPGA üzerinde gerçekleştirilen SPKS RK5-Butcher algoritması kullanılarak ve Tablo 6.10’da PWKS RK5-Butcher algoritması kullanılarak GRSÜ ünitesi için yapılan testlerin sonuçları verilmiştir. Yapılan testlerde üniversal test için $L=8$ ve $Q=4000$ olarak alınmış ve $P\text{-değeri} \geq 0.001$ olduğundan elde edilen diziler rasgele olarak kabul edilmektedir.

Tablo 6.9. SPKS RK5-Butcher tabanlı GRSÜ ünitesi NIST testi sonuçları

İstatistiksel Testler	P-değeri	Sonuç
Frequency (Monobit) Test	0.8943	Başarılı
Block-Frequency Test	0.0517	Başarılı
Cumulative-Sums Test	0.8958	Başarılı
Runs Test	0.1584	Başarılı
Longest-Run Test	0.8975	Başarılı
Binary Matrix Rank Test	0.4061	Başarılı
Discrete Fourier Transform Test	0.7059	Başarılı
Non-Overlapping Templates Test	0.8674	Başarılı
Overlapping Templates Test	0.1666	Başarılı
Maurer's Universal Statistical Test	0.1766	Başarılı
Approximate Entropy Test	0.6688	Başarılı
Random-Excursions Test	0.0958	Başarılı
Random-Excursions Variant Test	0.5318	Başarılı
Serial Test-1	0.3685	Başarılı
Serial Test-2	0.1825	Başarılı
Linear-Complexity Test	0.8456	Başarılı

Tablo 6.10. PWKS RK5-Butcher tabanlı GRSÜ ünitesi NIST testi sonuçları

İstatistiksel Testler	P-değeri	Sonuç
Frequency (Monobit) Test	0.0969	Başarılı
Block-Frequency Test	0.0808	Başarılı
Cumulative-Sums Test	0.1551	Başarılı
Runs Test	0.6882	Başarılı
Longest-Run Test	0.7819	Başarılı
Binary Matrix Rank Test	0.4853	Başarılı
Discrete Fourier Transform Test	0.8471	Başarılı
Non-Overlapping Templates Test	0.8446	Başarılı
Overlapping Templates Test	0.5084	Başarılı
Maurer's Universal Statistical Test	0.5109	Başarılı
Approximate Entropy Test	0.0084	Başarılı
Random-Excursions Test	0.0623	Başarılı
Random-Excursions Variant Test	0.1160	Başarılı
Serial Test-1	0.6017	Başarılı
Serial Test-2	0.3200	Başarılı
Linear-Complexity Test	0.0163	Başarılı

BÖLÜM 7. SONUÇLAR VE ÖNERİLER

Sunulan tez çalışmasında, SPKS ve PWKS dört farklı nümerik diferansiyel denklem çözüm yöntemi kullanılarak modellenmiş ve sistemlerin dinamik davranışları incelenerek kaos analizleri yapılmıştır.

Ayrıca seçilen kaotik sistemler bir ECAD programında şematik giriş ile analog devre elemanları kullanılarak modellenmiş ve ECAD benzetimi ile Matlab destekli nümerik model sonuçları birbiri ile uyumlu olduğu görülmüştür.

SPKS ve PWKS, FPGA üzerinde VHDL dili ile 32-bit IEEE 754-1985 kayan noktalı sayı standardına uygun olarak modellenmiştir. Modelleme aşamasında Euler, Heun, RK4 ve RK5-Butcher olmak üzere dört farklı algoritma kullanılmıştır. Xilinx Virtex-6 ailesi XC6VLX550T-2FF1759 çipi için Xilinx ISE Design Tools 14.2 benzetim programı kullanılarak sentezlenmiştir. FPGA çip kaynak kullanımına ve ünitelerin saat hızlarına ait parametrelerin istatistikleri incelenmiştir. Elde edilen sonuçlara göre kaotik osilatörlerin çalışma frekansı yaklaşık 390–464 MHz arasında değişmektedir. Ayrıca bu aşamada, nümerik tabanlı RK5-Butcher algoritması referans kabul edilerek FPGA tabanlı ünitelerin ürettiği sonuçların doğruluğunu test etmek amacıyla RMSE yöntemi kullanılarak hassasiyet analizleri yapılmıştır. FPGA-tabanlı algoritmaların ürettiği sonuçlara göre Euler-tabanlı ünite RMSE oranı 1.909, Heun-tabanlı ünite RMSE oranı 1.584 ve RK4-tabanlı ünite RMSE oranı 1.112 olarak hesaplanmıştır. Elde edilen sonuçlardan RK4 tabanlı gerçekleştirimin hassasiyetinin diğer gerçekleştirmelerden daha iyi olduğu görülmüştür.

Ardından, Euler, Heun, RK4 ve RK5-Butcher algoritmaları kullanılarak FPGA-tabanlı SPKS ve PWKS ile gerçek rasgele sayı üretici tasarımı gerçekleştirilmiştir. Kuantalama işlemleri için AED, SED ve KNS olmak üzere üç farklı model sunulmuştur. GRSÜ için yapılan üç farklı FPGA-tabanlı model Euler, Heun, RK4 ve

RK5-Butcher algoritmaları kullanılarak yapılan tasarımların her birine uygulanmıştır. Genel olarak iki farklı kaotik sistem, kaotik osilatör tasarımında dört ayrı algoritma ve kuantalama işlemi için üç değişik yöntem sunularak toplamda 24 farklı GRSÜ ünitesi tasarlanmıştır. İlk aşamada *SED-Tabanlı GRSÜ* üniteleri tasarlanmış ve yaklaşık bit üretim hızı 9.09 Kbit/s olarak belirlenmiştir. İkinci aşamada *AED-Tabanlı GRSÜ* üniteleri tasarlanmış ve yaklaşık bit üretim hızı 10.22 Kbit/s olarak belirlenmiştir. Son aşamada *KNS-Tabanlı GRSÜ* üniteleri tasarlanmış ve bu tasarımlarda 53–132 Mbit/s arasında değişen yüksek bit üretim hızları elde edilmiştir. Sonuçlara göre tasarımın rasgele sayı üretimi performansı *SED-Tabanlı GRSÜ* ve *AED-Tabanlı GRSÜ* ünitelerine göre oldukça yüksek olduğundan çalışmaların devamında *KNS-Tabanlı GRSÜ* üniteleri kullanılmıştır.

Son bölümde, FPGA tabanlı GRSÜ'den elde edilen sayı dizisi test edilmiştir. Bu amaçla tasarımı yapılan GRSÜ'nin doğrulanması, uluslararası düzeyde uygulanarak kabul görmüş olan ve en çok kullanılan test sistemleri olan NIST-800-22 ve FIPS-140-1 testleri ile sağlanmıştır. Tasarlanan SPKS ve PWKS osilatörlerinden yararlanılarak geliştirilen GRSÜ ünitelerinden 20 Kbit veri alınarak bir dosyaya kaydedilmiş ve FIPS-140-1 testlerindeki dört adet teste tabi tutulmuştur. Tüm SPKS ve PWKS-tabanlı GRSÜ üniteleri FIPS-140-1 testlerindeki dört testten de başarılı olmuştur. Diğer bir uluslararası geçerliliğe sahip istatistiksel test olan NIST Test Suite'de 15 adet test bulunmaktadır. Testlerin yapılabilmesi amacıyla her bir tasarım için veriler toplanarak bir dosyaya kaydedilmiştir. Ardından bit dosyası NIST Test Suite'de bulunan 15 adet teste tabi tutulmuş ve elde edilen tüm diziler rasgelelik testlerinden başarılı olmuştur.

Bu çalışmada Euler, Heun, RK4 ve RK5-Butcher algoritmaları olmak üzere dört nümerik diferansiyel denklem çözüm metodu üzerinde çalışmalar yapılmıştır. İleride yapılacak çalışmalar ile daha farklı nümerik diferansiyel denklem çözüm metotları kullanılarak GRSÜ yapıları gerçekleştirilebilir ve çalışmalar geliştirilebilir.

Ayrıca sunulan çalışma da GRSÜ yapılarından elde edilen rasgele sayı dizileri bir dosyaya kaydedilerek off-line olarak NIST-800-22 ve FIPS-140-1 gibi istatistiksel testlere tabi tutulmuştur. İleride yapılacak çalışmalarla bu istatistiksel testler FPGA

üzerinde gerçekleştirilerek testlerin hem gerçek zamanlı hem de daha hızlı gerçekleştirilmesi sağlanabilir.

KAYNAKLAR

- [1] ZHENGXING, H., WEI, D., HUILONG, D., HAOMIN, L., Similarity measure between patient traces for clinical pathway analysis: problem, method, and applications. *IEEE J. of Biomedical and Health Inf.*, 18(1):4–14, 2014.
- [2] XIONG, A., ZHAO, X., HAN, J., LIU, G., Application of the chaos theory in the analysis of EMG on patients with facial paralysis. In *Robot Intelligence Tech. and App.*, Springer, 274:805–819, 2014.
- [3] CHING, C., CHUN, L., SHYAN, L., YEN, C., CHENG, C., A chaotic theoretical approach to ECG-based identity recognition. *IEEE Comp. Intelligence Mag.*, 9(1):53–63, 2014.
- [4] YANG, J., YANTAO, C., FANGLAI, Z., Singular reduced-order observer-based synchronization for uncertain chaotic systems subject to channel disturbance and chaos-based secure communication. *Applied Math. and Comp.*, Elsevier, 229:227–238, 2014.
- [5] KADDOUM, G., GAGNON, F., Lower bound on the bit error rate of a decode-and-forward relay network under chaos shift keying communication system. *IET Comm.*, 8(2):227-232, 2014.
- [6] XINLEI, M., YONHHPNG, C., DDoS detection method based on chaos analysis of network traffic entropy. *IEEE Comm. Lett.*, 18(1):114–117, 2014.
- [7] YANHUA, H., SPENCER, PS., SHORE, KA., Wideband chaos with time-delay concealment in vertical-cavity surface-emitting lasers with optical feedback and injection. *IEEE J. of Quantum Electr.*, 50(4):236–242, 2014.
- [8] ZEXIN, K., JIANG, S., LIN, M., YANHUI, Q., SHUISHENG, J., Multimode synchronization of chaotic semiconductor ring laser and its potential in chaos communication. *IEEE J. of Quantum Electr.*, 50(3):148–157, 2014.
- [9] LI, H., HU, Y., Observer-based synchronization for laser systems. *IEEE J. of Quantum Electr.*, 50(5):372–378, 2014.

- [10] BARTEL, A., HULLSMANN, T., KUHN, J., PULCH, R., SCHOPS, S., Influence of measurement errors on transformer inrush currents using different material models. *IEEE Trans. on Magn.*, 50(2):485–488, 2014.
- [11] MANFREDI, P., CANAVERO, FG., Numerical calculation of polynomial chaos coefficients for stochastic per-unit-length parameters of circular conductors. *IEEE Trans. on Magn.*, 50(3):74–82, 2014.
- [12] BARAKAT, ML., MANSINGKA, AS., RADWAN, AG., SALAMA, KN., Hardware stream cipher with controllable chaos generator for colour image encryption. *IET Image Processing*, 8(1):33–43, 2014.
- [13] ANBING, Z., YICHUN, X., Chaos theory-based data-mining technique for image endmember extraction: lyapunov index and correlation dimension. *IEEE Trans. on Geoscience and Remote Sensing*, 52(4):1935–1947, 2014.
- [14] ANEES, A., SIDDIQUI, AM., AHMED, F., Chaotic substitution for highly autocorrelated data in encryption algorithm. *Communications in Nonlinear Sci. and Numerical Sim.*, Elsevier, 19(9):3106–3118, 2014.
- [15] BEIRAMI, A., NEJATI, H., Aframework for investigating the performance of chaotic-map TRNGs. *IEEE Trans. on Circuits and Sys.*, 60(7):446–450, 2013.
- [16] BELAZI, A., HERMASSI, H., RHOUMA, R., BELGHITH, S., Algebraic analysis of a RGB image encryption algorithm based on DNA encoding and chaotic map. *Nonlinear Dyn.*, Springer, 76:1989–2004, 2014.
- [17] XIAOLE, F., WETZEL, B., MORELLA, JM., DUDLEY, JM., LARGER, L., GUYEUX, C., BAHI, JM., Noise and chaos contributions in fast random bit sequence generated from broadband optoelectronic entropy sources. *IEEE Trans. on Circuits and Syst.*, 61(3):888–901, 2014.
- [18] JI, Y., ZHANG, M., WANG, Y., WU, Y., ZHANG, Y., Microwave-photonic sensor for remote water-level monitoring based on chaotic laser. *Int. J. of Bifurcation and Chaos*, 24(3):321–327, 2014.
- [19] YE, YZ., WEI, LZ., YUN, JR., ZI, NW., XIN, HJ., Output characterization of random fiber laser formed by dispersion compensated fiber. *IEEE Photonics Tech. Lett.*, 26(3):246–248, 2014.
- [20] ZHENG, GW., PENG, S., HONGYE, S., JIAN, C., Sampled-data fuzzy control of chaotic systems based on a T–S fuzzy model. *IEEE Trans. on Fuzzy Sys.*, 22(1):153–163, 2014.

- [21] U, SH., KANG, HS., KIM, YT., HYUN, CH., PARK, M., Fuzzy adaptive modular design of uncertain chaotic duffing oscillators. *Int. J. of Control Automation and Sys.*, 12(1):188–194, 2014.
- [22] CHING, HL., FENG, YC., CHIH, ML., An efficient interval type-2 fuzzy CMAC for chaos time-series prediction and synchronization. *IEEE Trans. on Cybernetics*, 44(3):329–341, 2014.
- [23] DEIVASUNDARI, P., UMA, G., ASHITA, S., Chaotic dynamics of a zero average dynamics controlled DC-DC Cuk converter, *IET Power Electr.*, 7(2):289–298, 2014.
- [24] NATARAJAN, S., RAJASEKAR, N., An FPGA chaos-based PWM technique combined with simple passive filter for effective EMI spectral peak reduction in DC-DC converter. *Advances in Power Electr.*, 1–11, 2014.
- [25] XIA, C., ZHAO, YD., KE, M., YAN, X., KIT, PW., NGAN, HW., Electricity price forecasting with extreme learning machine and bootstrapping. *IEEE Trans. on Power Syst.*, 27(4):2055–2062, 2012.
- [26] LONES, MA., FUENTE, LA., TURNER, AP., CAVES, LSD., STEPNEY, S., SMITH, SL., TYRRELL, AM., Artificial biochemical networks: evolving dynamical systems to control dynamical systems. *IEEE Trans. on Evolutionary Comp.*, 18(2):145–166, 2014.
- [27] JIN, L., MEI, J., LI, L., Chaos control of parametric driven Duffing oscillators. *Applied Physics Lett.*, 104(13):1011–1015, 2014.
- [28] WAN, L., LUO, XS., ZENG, SY., ZHANG, B., Global exponential stabilization for chaotic brushless DC motors with a single input. *Springer Nonlinear Dyn.*, 77(1–2):209–212, 2014.
- [29] ARASHAR, A., SINGH, R., PANIGRAHI, PK., MURALIDHAR, K., Chaotic flow in an aortic aneurysm. *J. of Applied Phys.*, 113(21):1–14, 2013.
- [30] PIGNOLET, A., Ferroelectrics chaotic memory. *Nature Phys.*, 10(1):9–11, 2014.
- [31] HEMMATI, M., AMJADY, N., EHSAN, M., System modeling and optimization for islanded micro-grid using multi-cross learning-based chaotic differential evolution algorithm. *Int. J. of Elect. Power and Energy Syst.*, Elsevier, 56:349–360, 2014.
- [32] KAVEH, A., SHEILHOESLAMI, R., TALATAHARI, S., KESHVARI, IM., Chaotic swarming of particles: A new method for size optimization of truss structures. *Advances in Eng. Softw.*, 67:136–147, 2014.

- [33] POMARES, J., PEREA, I., TORRES, F., Dynamic visual servoing with chaos control for redundant robots. *IEEE Trans. on Mechatronics*, 19(2):423–431, 2014.
- [34] DECAI, L., MIN, H., JUN, W., Chaotic time series prediction based on a novel robust echo state network. *IEEE Trans. on Neural Networks and Learning Syst.*, 23(5):787–799, 2012.
- [35] STROGATZ, SH., *Nonlinear dynamics and chaos with applications to physics, biology, chemistry*. Perseus Publ., LLC, ABD, 2006.
- [36] OTT, E., *Chaos in dynamical systems*. Cambridge University Press, sec. ed., USA, 2002.
- [37] ALLIGOOD, KT., SAUER, TD., YORKE, JA., CRAWFORD, JD., *Chaos: An introduction to dynamical systems*. Physics Today, Springer, 50(11):67–68, 2008.
- [38] KOYUNCU, I., OZCERIT, AT., PEHLIVAN, I., Implementation of FPGA-based real time novel chaotic oscillator. *Nonlinear Dyn.*, Springer, 75(1–2):49–59, 2014.
- [39] AKIZAWA, Y., YAMAZAKI, T., UCHIDA, A., HARAYAMA, T., SUNADA, S., ARAI, K., YOSHIMURA, K., DAVIS, P., Fast RNG with bandwidth-enhanced chaotic semiconductor lasers at 8 times 50Gb/s. *IEEE Photonics Tech. Lett.*, 24(12):1042–1044, 2012.
- [40] UYAROĞLU, Y., PEHLIVAN, I., Nonlinear Sprott94 case a chaotic equation: synchronization and masking communication applications. *Computers and Elect. Eng.*, Elsevier, 36:1093–1100, 2010.
- [41] PARESCHI, F., SETTI, G., ROVATTI, R., Implementation and testing of high-speed CMOS TRNGs based on chaotic systems. *IEEE Trans. on Circuits and Syst.*, 57(12):3124–3137, 2010.
- [42] GALAJDA, MDP., Chaos-based true RNG embedded in a mixed-signal reconfigurable hardware. *J. of Elect. Eng.*, 57(4):218–225, 2006.
- [43] DRUTAVOSKY, M., GALAJDA, P., A robust chaos-based TRNG embedded in reconfigurable switched-capacitor hardware. *Radioelektronika*, 17th Int. Conf., 1–6, 2007.
- [44] PANDE, A., ZAMBRENO, J., Design and hardware implementation of a chaotic encryption scheme for real-time embedded systems. *Int. Conf. on Signal Processing and Comm.*, 1–5, 2010.
- [45] EROĞLU, C., Implementation of synchronized chaotic systems by FPGA. Graduate Sch. of Eng. and Sci. of Izmir Inst. of Tech., Izmir, Turkey, 2007.

- [46] ZHONG, Z., GUANRONG, C., SIMIN, Y., Hyperchaotic signal generation via DSP for efficient perturbations to liquid mixing. *Int. J. of Circuit Theory and App.*, 37:31–41, 2009.
- [47] KHAREL, R., BUSAWON, K., AGGOUNE, W., GHASSEMLOY, Z., Implementation of a secure digital chaotic communication scheme on a DSP board. *7th Int. Symp. on Comm. Syst. Networks and Digital Signal Process.*, 212–216, 2010.
- [48] DELGADO, RM., ACOSTA, AJ., RODRIGUEZ, VA., A mixed-signal integrated circuit for FM-DCSK modulation. *IEEE J. of Solid-State Circ.*, 40(7):1460–1471, 2005.
- [49] YIWEI, Z., ZEXIANG, L., XINJIAN, Z., A chaos-based image encryption ASIC using reconfigurable logic. *IEEE Asia Pacific Conf. on Circuits and Syst.*, 1782–1785, 2008.
- [50] AZZAZ, MS., TANOUGAST, C., SADOUDI, S., FELLAH, R., DANDACHE, A., A new auto-switched chaotic system and its FPGA implementation. *Comm. in Nonlinear Sci. and Numerical Sim.*, 1007–5704, 2012.
- [51] PAOLO, A., SEBASTIANO, DF., LUIGI, F., MATTIA, F., LUCA, P., GUIDO, V., Reactive navigation through multiscroll systems: from theory to real-time implementation. *Autonomous Robots*, Springer, 25(1–2):123–146, 2008.
- [52] TE, S., GUOSHENG, R., YANG, Z., SONG, Z., Design method for Duffing system based on DSP builder. *Int. Conf. on Syst. and Inform.*, 121–124, 2012.
- [53] SADOUDI, S., MOHAMED, SA., MUSTAPHA, D., MUSTAPHA, B., An FPGA real-time implementation of the Chen's chaotic system for securing chaotic communications. *Int. J. of Nonlinear Sci.*, 7(4):467–474, 2009.
- [54] MERAH, L., PASCHA, A., SAID, A., MAMAT, NH., Design and FPGA implementation of Lorenz chaotic system for information security issues. *Appl. Math. Sci.*, 7(5):237–246, 2013.
- [55] DEMICCO, L., LARRONDO, HA., FPGA implementation of a chaotic oscillator using RK4 method. *VII Southern Conf. on Programm. Logic*, 185–190, 2011.
- [56] WANG, GY., BAO, XL., WANG, ZL., Design and FPGA implementation of a new hyperchaotic system. *Chinese Phys. B*, 10:3596–3602, 2008.

- [57] QUN, D., JING, P., JINQUING, F., XIYUAN, P., Designin of chotic system output sequence circuit based on FPGA and its applications in network encryption card. *Int. J. Innov. Comput. Inform. Control*, 3(2):449–456, 2007.
- [58] DANGER, JL., GUILLEY, S., HOOGVORST, P., High speed true random number generator based on open loop structures in FPGAs. *Microelectronics J.*, Elsevier, 40(11):1650–1656, 2009.
- [59] WIECZOREK, PZ., GOLOFIT, K., Dual-metastability time-competitive TRNG. *IEEE Trans. on Circuits and Syst.*, 61(1):134–145, 2014.
- [60] LOZACH, F., BEN, RM., GRABA, T., DANGER, JL., FPGA design of an open-loop TRNG. *Euromicro Conf. on Digital Sys. Design*, 615–622, 4-6 Sept. 2013.
- [61] FISCHER, V., DRUTAVOSKY, M., SIMKA, M., BOCHARD, N., High performance TRNG in altera stratix FPLDs. *Field Programmable Logic and App.*, Springer, 3203:555–564, 2004.
- [62] SCHELLEKENS, B., PRENEEL, I., VERBAUWHEDE, I., FPGA vendor agnostic TRNG. *Proc. of Int. Conf. on Field Programmable Logic and App.*, 1–6, 2006.
- [63] DICHTL, M., GOLIC, J., High-speed TRNG with logic gates only. *Lect. Notes in Computer Sci.*, 4727:45–62, 2007.
- [64] ISTVAN, H., SUCIU, A., CRET, O., FPGA based TRNG using automatic calibration. *Intelligent Comp. Comm. and Proc.*, IEEE 5th Int. Conf. on ICCP, 373–376, 2009.
- [65] CİCEK, İ., PUSANE, AE., DUNDAR, G., A novel design method for discrete time chaos based true random number generators. *The VLSI J. Integr.*, Elsevier, 47(1):38–47, 2014.
- [66] ERGÜN, S., ÖZOĞUZ. S., Truly RNGs based on non-autonomous continuous-time chaos. *Int. J. of Circuit Theory and App.*, 38(1):1–24, 2010.
- [67] PARESCHI, F., SETTI, G., ROVATTI, R., Implementation and testing of high-speed CMOS TRNGs sased on chaotic systems. *IEEE Trans. on Circuits and Sys.*, 57(12):3124–3137, 2010.
- [68] CİCEK, I., PUSANE, AE., DUNDAR, G., A novel dual entropy core TRNG. *IEEE 8th Int. Conf. on Elec. and Electronics Eng.*, 332–335, 2013.
- [69] ÖZOĞUZ, S., ZEKİ, A., Sürekli zamanlı kaotik sistemlerin tümleşik olarak gerçekleştirilmesi ve rasgele sayı üretiminde kullanılması. *Tübitak Projesi Sonuç Raporu (106E093)*, 2008.

- [70] Federal information processing standards publication, Security requirements for cryptographic modules. FIPS PUB 140-1, 1994. <http://csrc.nist.gov/publications/fips/fips1401.htm>, Eriřim Tarihi: 06.06.2014.
- [71] A statistical test suite for random and pseudo RNGs for cryptographic applications. National institute of stand. and tech.,NIST-800-22, 2001. <http://csrc.nist.gov/publications/nistpubs/800-22/sp-800-22-051501.pdf>, Eriřim Tarihi: 06.06.2014.
- [72] LORENZ, EN., Deterministic nonperiodic flow. *J. of the Atmospheric Sci.*, 20:130–141, 1963.
- [73] LI, TY., YORKE, JA., Period three implies chaos. *The American Math. Monthly*, 82(10):985–992, 1975.
- [74] RÖSSLER, OE., An equation for continuous chaos. *Physics Lett.*, 57(5):397–398, 1976.
- [75] RÖSSLER, OE., Continuous chaos-four prototype equations. *Annals of the New York Academy of Sci.*, 316:376–392, 1979.
- [76] MATSUMOTO, T., CHUA, LO., TANAMA, S., Simplest chaotic nonautonomous circuit. *Physical Rev.*, 30:1155–1157, 1984.
- [77] ZHANG, W., GUI, Z., WANG, K., Impulsive control for synchronization of Lorenz chaotic system. *J. of Software Eng. and Appl.*, 5(12B):23–25, 2013.
- [78] KUETCHE, M., FOTSIN, ES., KENGNE, HB., WOAFU, P., Parameters estimation based adaptive GPS of chaotic Chua's circuit with application to chaos communication by parametric modulation. *Chaos, Solitons & Fract.*, Elsevier, 61:27–37, 2014.
- [79] ÇİÇEK, S., UYAROĞLU, Y., PEHLİVAN, İ., Simulation and circuit implementation of sprott case H chaotic system and its synchronization application for secure communication systems. *J. of Circuits, Syst. and Comp.*, 22(04):1–15, 2013.
- [80] PRECUP, RE., TOMESCU, ML., DRAGOS, CA., Stabilization of Rössler chaotic dynamical system using fuzzy logic control algorithm. *Int. J. of General Syst.*, 43(5):413–433, 2014.
- [81] KOCAMAZ, UE., UYAROĞLU, Y., KIZMAZ, H., Control of Rabinovich chaotic system using sliding mode control. *Int. J. of Adaptive Control and Signal Proc.*, Wiley, 1–9, 2013.

- [82] VEMBARASAN, V., BALASUBRAMANIAM, P., Chaotic synchronization of Rikitake system based on TS fuzzy control techniques. *Nonlinear Dyn.*, 74(1–2):31–44, 2013.
- [83] KOYUNCU, I., OZCERIT, AT., PEHLIVAN, P., An analog circuit design and FPGA-based implementation of the Burke-Shaw chaotic system. *Optoelectronics and Advanced Materials-Rapid Comm.*, 7(9–10):635–638, 2013.
- [84] HU, HP., LIU, LF., DING, ND., Pseudorandom sequence generator based on the Chen chaotic system. *Computer Physics Comm.*, 184(3):765–768, 2013.
- [85] BUTCHER, JC., Numerical methods for ordinary differential equations. 2nd ed., L. John Wiley & Sons, Ed., 2008.
- [86] PANCHEV, S., SPASSOVA, T., VITANOV, NK., Analytical and numerical investigation of two families of Lorenz-like dynamical systems. *Chaos, Solitons & Fract.*, 33:1658–1671, 2007.
- [87] KUON, I., TESSIER, R., ROSE, J., FPGA architecture: survey and challenges. *Foundations and Trends in Electr. Design Autom.*, 2(2):135–253, 2007.
- [88] MUNDEN, R., ASIC and FPGA verification: a guide to component modeling. Morgan Kaufmann Publ., Elsevier, San Francisco, USA, 2005.
- [89] http://www.xilinx.com/publications/prod_mktg/zynq7000/Zynq-7000-combined-product-table.pdf, Erişim Tarihi: 09.06.2014.
- [90] ŞAHİN, İ., KOYUNCU, İ., A new module design for 3D graphic transformations using generated floating-point core units. *I. Rev. on Modelling and Sim.*, 4(2):691–698, 2011.
- [91] ÇAKIROĞLU, M., ÖZCEİT, AT., ÇETİN Ö., ESKİKURT, HA., Integration of real-time counter unit into a microcontroller through reconfiguration. *Proc. of Twelfth Int. Symp. on Artificial Intelligence and Neural Netw.*, 2003.
- [92] SAHİN, I., GLOSTER, C., DOSS, C., Feasibility of floating-point arithmetic in reconfigurable computing systems. *Military and Aerospace Applications of Programmable Devices and Techn. Conf.*, Washington, DC, 2000.
- [93] SAHİN, I., A 32-bit floating-point module design for 3D graphic transformations. *Sci. Research and Ess.*, 5(20):3070–3081, 2010.
- [94] SARITAŞ, E., KARATAŞ, S., Her yönüyle FPGA ve VHDL. Palme Yayıncılık, Ankara, 2013.

- [95] DASILVA, RC., KUROKAWA, S., JOSE, A., PISSOLATO, J., Integration methods used in numerical simulations of transient electromagnetic. *IEEE Latin America Trans.*, 9(7):1060–1065, 2011.
- [96] BENNETT, A., MILNE, W., BATEMANN, H., Numerical integration of differential equations. Dover, 1956.
- [97] SENTHILKUMAR, S., PIAH, A., An improved fuzzy cellular neural network (IFCNN) for an edge detection based on parallel RK (5, 6) approach. *Int. J. of Computational Systems Eng.*, 1(1):70–78, 2012.
- [98] DORMAND, JR., PETER, JP., A family of embedded Runge-Kutta formulae. *J. of Computational and Applied Math.*, 6(1):19–26, 1980.
- [99] BUTCHER, JC., Numerical methods for ordinary differential equations. 2nd ed., L. John Wiley & Sons, Ed., 2008.
- [100] MENEZES, AJ., PAUL, C., VAN, O., SCOTT, AV., Handbook of applied cryptography. CRC press, 1996.
- [101] BEIRAMI, A., NEJATI, H., ALI, WH., Zigzag map: a variability-aware discrete-time chaotic-map truly random number generator. *Electronics Lett.*, 48(24):1537–1538, 2012.
- [102] ZHAO, L., LIAO, X., XIAO, D., XIANG, T., ZHOU, Q., DUAN, S., TRNG from mobile telephone photo based on chaotic cryptography. *Chaos, Solitons & Fract.*, Elsevier, 42(3):1692–1699, 2009.
- [103] ERGÜN, S., ÖZOĞUZ, S., TRNGs based on a non-autonomous chaotic oscillator. *Int. J. of Electronics and Comm.*, 61(4):235–242, 2007.
- [104] LI, Q., LIU, Q., NIU, J., Chaotic oscillator with potentials in TRNG and ADC. 35th Int. Conf. on Telecomm. and Signal Proc., 397–400, 3–4 July 2012.
- [105] ANGULO, JAA., KUSSENAR, E., BARTHELEMY, H., DUVAL, B., A new oscillator-based RNG. *IEEE Faible Tension Faible Cons.*, 1–4, June 2012.
- [106] STOJANOVSKI, T., KOCAREV, L., Chaos-based random number generators-part I: analysis. *IEEE Trans. on Circuits and Syst. I: Fundamental Theory and Appl.*, 48(3):281–288, 2001.
- [107] PETRIE, CS., CONNELLY, JA., A noise-based IC RNG for applications in cryptography. *IEEE Trans. on Circuits and Syst. I: Fundamental Theory and Appl.*, 47(5):615–621, May 2000.

- [108] BUCCI, M., GERMANI, L., LUZZI, R., TOMMASINO, P., TRIFILETTI, A., VARANONUOVO M., A high-speed IC random-number source for smart card microcontrollers. *IEEE Trans. on Circuits and Sys.I: Fundamental Theory and Appl.*, 50(11):1373–1380, 2003.
- [109] NIEN, HH., HUANG, CK., CHANGCHIEN, SK., SHIEH, HW., CHEN, CT., TUAN, YY., Digital color image encoding and decoding using a novel chaotic random generator. *Chaos, Solitons & Fract.*, 32(3):1070–1080, 2007.
- [110] AVAROGLU, E., TURK, M., RNG using multi-mode chaotic attractor. *IEEE Signal Processing and Comm. Appl. Conf.*, 1–4, April 2013.
- [111] DEMİRKOL, A., Kaotik osilatör girişli ADC tabanlı rasgele sayı üretici. Yüksek lisans tezi, İstanbul Teknik Üniversitesi, 2007.
- [112] GÜVEN, P., Otonom olmayan kaotik sistemlerde rasgele sayı üretiminin incelenmesi. Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, 2006.
- [113] <http://www.csm.ornl.gov/~dunigan/fips140.txt>, Erişim Tarihi: 26.05.2014.
- [114] YAYIK, A., KUTLU, Y., Improving PNRG using artificial neural networks. *IEEE 21st Signal Processing and Comm. App. Conf.*, 1–4, 2013.
- [115] AVAROĞLU, E., Donanım tabanlı rasgele sayı üreticinin gerçekleştirilmesi. Doktora Tezi, Fırat Üniversitesi, 2014.
- [116] BÜYÜKSARAÇOĞLU, F., BULUŞ, E., Sözde rastsal sayı üretiminin kriptografik açıdan incelenmesi. *TMMOB Elektrik Müh. Odası IV. İletişim Tekn. Ul. Semp.*, Adana, 2009.
- [117] Kriptografiye giriş ders notları. Uygulamalı Matematik Enstitüsü Kript. Böl., ODTÜ, 2004.
- [118] MAURER, UM., A universal statistical test for random bit generators. *J. of Cryptology*, 5(2):89–105, 1992.
- [119] MENEZES, AJ., ORSCHOT PC., VANSTANO, SA., *Handbook of app. Cryp.*, CRC press, 1996.
- [120] SUNDARAPANDIAN, V., PEHLIVAN, I., Analysis, control, synchronization, and circuit design of a novel chaotic system. *Math. Comp. Modelling*, 55(7–8):1904–1915, 2012.
- [121] YARDIM, FE., AFACAN, E., Simulation of a communication system using Lorenz-based differential chaos shift keying model. *J. Fac. Eng. Arch. Gazi Univ.*, 25(1):101–110, 2010.

- [122] OZKAYNAK, F., Doğrusal olmayan sistemlerde Lyapunov üstellerini hesaplayan yazılımın gerçekleştirilmesi. Yüksek Lisans Tezi, Fırat Üniversitesi, 2007.
- [123] LYAPUNOV, AM., The general problem of the stability of motion. *Int. J. of Control*, 55(3):531–534, 1992.
- [124] AKSOY, S., Lineer sistem teorisi. Yüksek Lisans Ders Notları, Sakarya Üniversitesi, 2011.
- [125] KAM, SW., Lyapunov exponents toolbox. Dep. of Electronic Eng., City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong, 1999.
- [126] PEHLIVAN, I., WEI, Z., Analysis, nonlinear control and circuit design of another strange chaotic system. *Turkish J. of Electrical Eng. and Comp. Sci.*, 20(2):1229–1239, 2012.
- [127] http://www.xilinx.com/support/documentation/data_sheets/ds031.pdf, Erişim Tarihi: 24.05.2014.
- [128] http://read.pudn.com/downloads125/doc/fileformat/529590/xilinx_spartan3e_fa105.pdf, Erişim Tarihi: 24.05.2014.
- [129] http://en.wikipedia.org/wiki/Single-precision_floating-point_format, Erişim Tarihi: 09.06.2014.

EKLER

EK-A: Doktora Tez Kapsamında Yapılan Bilimsel Çalışmalar

Doktora tez kapsamında yapılan bilimsel yayınlar aşağıda verilmiştir.

- [1] KOYUNCU, I., OZCERIT, AT., PEHLIVAN, I., Implementation of FPGA-based real time novel chaotic oscillator, *Nonlinear Dynamics*, Springer, 77(1–2):49–59, 2014.
- [2] KOYUNCU, I., OZCERIT, AT., PEHLIVAN, I., AVAROGLU, E., Design and implementation of chaos based true random number generator on FPGA, 22st IEEE Signal Processing and Communications Applications Conf., Trabzon, Turkey, 2014.
- [3] KOYUNCU, I., OZCERIT, AT., PEHLIVAN, I., An analog circuit design and FPGA-based implementation of the Burke-Shaw chaotic system, *Optoelectronics and Advanced Materials– Rapid Communications*, 7(9-10):635–638, 2013.
- [4] KOYUNCU, I., OZCERIT, AT., PEHLIVAN, I., Implementation of Burke-Shaw chaotic system on FPGA, 2nd International Eurasian Conference on Mathematical Sciences and Applications, IECMSA-2013, Sarajevo, Bosnia and Herzegovina, 2013.
- [5] KOYUNCU, I., OZCERIT, AT., PEHLIVAN, I., FPGA-based a chaotic oscillator design and implementation, 1st International Symposium on Innovative Technologies in Engineering and Science, ISITES2013, 873-879, Sakarya University, Turkey, June 2013.

Doktora tez kapsamında yapılan bilimsel projeler aşağıda verilmiştir.

- [1] ÖZCERİT AT, PEHLİVAN İ, KOYUNCU İ, FPGA üzerinde gerçek zamanlı yeni kaotik osilatörlerin tasarımı ve gerçekleştirilmesi, Sakarya Üniversitesi BAP, 2014-09-10-001, 2014.

EK-B: Virtex-6 FPGA Çip Ailesi Katalog Bilgileri

Tezin bu bölümünde FPGA-tabanlı kaotik osilatör ve GRSÜ gerçeklemelerinde kullanılan Virtex-6 FPGA çip ailesinin genel katalog bilgileri verilmiştir [127].



Virtex-6 Family Overview

DS150 (v2.4) January 19, 2012

Product Specification

General Description

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

Summary of Virtex-6 FPGA Features

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

Virtex-6 FPGA Feature Summary

Table 1: Virtex-6 FPGA Feature Summary by Device

Device	Logic Cells	Configurable Logic Blocks (CLBs) ¹		DSP48E ¹ Slices ⁽²⁾	Block RAM Blocks			MMCMs ⁽⁴⁾	Interface Blocks for PCI Express	Ethernet ³ MACs ⁽⁵⁾	Maximum Transceivers		Total I/O Banks ⁽⁶⁾	Max User I/O ⁽⁷⁾
		Slices ⁽¹⁾	Max Distributed RAM (Kb)		18 Kb ⁽²⁾	36 Kb	Max (Kb)				GTX	GTH		
XC6VLX75T	74,496	11,640	1,045	288	312	156	5,616	6	1	4	12	0	9	360
XC6VLX130T	128,000	20,000	1,740	480	528	264	9,504	10	2	4	20	0	15	600
XC6VLX195T	199,680	31,200	3,040	640	688	344	12,384	10	2	4	20	0	15	600
XC6VLX240T	241,152	37,680	3,650	768	832	416	14,976	12	2	4	24	0	18	720
XC6VLX365T	364,032	56,880	4,130	576	832	416	14,976	12	2	4	24	0	18	720
XC6VLX550T	549,888	85,920	6,200	864	1,264	632	22,752	18	2	4	36	0	30	1200
XC6VLX760	758,784	118,560	8,280	864	1,440	720	25,920	18	0	0	0	0	30	1200
XC6VSX315T	314,880	49,200	5,090	1,344	1,408	704	25,344	12	2	4	24	0	18	720
XC6VSX475T	476,160	74,400	7,640	2,016	2,128	1,064	36,304	18	2	4	36	0	21	840
XC6VHX250T	251,904	39,360	3,040	576	1,008	504	18,144	12	4	4	48	0	8	320
XC6VHX255T	253,440	39,600	3,050	576	1,032	516	18,576	12	2	2	24	24	12	480
XC6VHX380T	382,464	59,760	4,570	864	1,536	768	27,648	18	4	4	48	24	18	720
XC6VHX565T	566,784	88,560	6,370	864	1,824	912	32,832	18	4	4	48	24	18	720

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

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

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

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

Notes:

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

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

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

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

Notes:

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

Configuration

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

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

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

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

Dynamic Reconfiguration Port

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

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

Encryption, Readback, and Partial Reconfiguration

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

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

CLBs, Slices, and LUTs

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

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

Clock Management

Each Virtex-6 FPGA has up to nine clock management tiles (CMTs), each consisting of two mixed-mode clock managers (MMCMs), which are PLL based.

Phase-Locked Loop

The MMCM can serve as a frequency synthesizer for a wider range of frequencies and as a jitter filter for incoming clocks. The heart of the MMCM is a voltage-controlled oscillator (VCO) with a frequency from 600 MHz up to 1600 MHz, spanning more than one octave. There are three sets of programmable frequency dividers (D, M, and O).

The pre-divider D (programmable by configuration) reduces the input frequency and feeds one input of the traditional PLL phase/frequency comparator. The feedback divider (programmable by configuration) acts as a multiplier because it divides the VCO output frequency before feeding the other input of the phase comparator. D and M must be chosen appropriately to keep the VCO within its specified frequency range.

The VCO has eight equally-spaced output phases (0°, 45°, 90°, 135°, 180°, 225°, 270°, and 315°). Each can be selected to drive one of the seven output dividers, O0 to O6 (each programmable by configuration to divide by any integer from 1 to 128).

MMCM Programmable Features

The MMCM has three input-jitter filter options: low bandwidth, high bandwidth, or optimized mode. Low-bandwidth mode has the best jitter attenuation but not the smallest phase offset. High-bandwidth mode has the best phase offset, but not the best jitter attenuation. Optimized mode allows the tools to find the best setting.

The MMCM can have a fractional counter in either the feedback path (acting as a multiplier) or in one output path. Fractional counters allow non-integer increments of 1/8 and can thus increase frequency synthesis capabilities by a factor of 8.

The MMCM can also provide fixed or dynamic phase shift in small increments that depend on the VCO frequency. At 600 MHz the phase-shift timing increment is 30 ps; at 1600 MHz, it is 11.5 ps.

Clock Distribution

Each Virtex-6 FPGA provides five different types of clock lines (BUFG, BUFR, BUFIO, BUFH, and the high-performance clock) to address the different clocking requirements of high fanout, short propagation delay, and extremely low skew.

Global Clock Lines

In each Virtex-6 FPGA, 32 global-clock lines have the highest fanout and can reach every flip-flop clock, clock enable, set/reset, as well as many logic inputs. There are 12 global clock lines within any region. Global clock lines can be driven by global clock buffers, which can also perform glitchless clock multiplexing and the clock enable function. Global clocks are often driven from the CMT, which can completely eliminate the basic clock distribution delay.

Regional Clocks

Regional clocks can drive all clock destinations in their region as well as the region above and below. A region is defined as any area that is 40 I/O and 40 CLB high and half the chip wide. Virtex-6 FPGAs have between 6 and 18 regions. There are 6 regional clock tracks in every region. Each regional clock buffer can be driven from either of four clock-capable input pins, and its frequency can optionally be divided by any integer from 1 to 8.

I/O Clocks

I/O clocks are especially fast and serve only I/O logic and serializer/deserializer (SerDes) circuits, as described in the [I/O Logic](#) section. Virtex-6 devices have a high-performance direct connection from the MMCM to the I/O directly for low-jitter, high-performance interfaces.

Block RAM

Every Virtex-6 FPGA has between 156 and 1064 dual-port block RAMs, each storing 36 Kbits. Each block RAM has two completely independent ports that share nothing but the stored data.

Synchronous Operation

Each memory access, read and write, is controlled by the clock. All inputs, data, address, clock enables, and write enables are registered. *Nothing happens without a clock.* The input address is always clocked, retaining data until the next operation. An optional output data pipeline register allows higher clock rates at the cost of an extra cycle of latency.

During a write operation, the data output can reflect either the previously stored data, the newly written data, or remain unchanged.

Programmable Data Width

- Each port can be configured as $32K \times 1$, $16K \times 2$, $8K \times 4$, $4K \times 9$ (or 8), $2K \times 18$ (or 16), $1K \times 36$ (or 32), or 512×72 (or 64). The two ports can have different aspect ratios, without any constraints.
- Each block RAM can be divided into two completely independent 18 Kb block RAMs that can each be configured to any aspect ratio from $16K \times 1$ to 512×36 . Everything described previously for the full 36 Kb block RAM also applies to each of the smaller 18 Kb block RAMs.
- In 18 Kb block RAMs, only simple dual-port mode can provide data width of >36 bits. In this mode, one port is dedicated to read and the other port is dedicated to write operation. In SDP mode one side (read or write) can be variable while the other is fixed to 32/36 or 64/72. There is no read output during write. The dual-port 36 Kb RAM both sides can be of variable width.
- Two adjacent 36 Kb block RAMs can be configured as one cascaded $64K \times 1$ dual-port RAM without any additional logic.

Error Detection and Correction

Each 64 bit-wide block RAM can generate, store, and utilize eight additional Hamming-code bits, and perform single-bit error correction and double-bit error detection (ECC) during the read process. The ECC logic can also be used when writing to, or reading from external 64/72-wide memories. This works in simple dual-port mode and does not support read-during-write.

FIFO Controller

The built-in FIFO controller for single-clock (synchronous) or dual-clock (asynchronous or multirate) operation increments the internal addresses and provides four handshaking flags: full, empty, almost full, and almost empty. The almost full and almost empty flags are freely programmable. Similar to the block RAM, the FIFO width and depth are programmable, but the write and read ports always have identical width. First-word fall-through mode presents the first-written word on the data output even before the first read operation. After the first word has been read, there is no difference between this mode and the standard mode.

Digital Signal Processing—DSP48E1 Slice

DSP applications use many binary multipliers and accumulators, best implemented in dedicated DSP slices. All Virtex-6 FPGAs have many dedicated, full-custom, low-power DSP slices combining high speed with small size, while retaining system design flexibility.

Each DSP48E1 slice fundamentally consists of a dedicated 25×18 bit two's complement multiplier and a 48-bit accumulator, both capable of operating at 600 MHz. The multiplier can be dynamically bypassed, and two 48-bit inputs can feed a single-instruction-multiple-data (SIMD) arithmetic unit (dual 24-bit add/subtract/accumulate or quad 12-bit add/subtract/accumulate), or a logic unit that can generate any one of 10 different logic functions of the two operands.

The DSP48E1 includes an additional pre-adder, typically used in symmetrical filters. This new pre-adder improves performance in densely packed designs and reduces the logic slice count by up to 50%.

The DSP48E1 slice provides extensive pipelining and extension capabilities that enhance speed and efficiency of many applications, even beyond digital signal processing, such as wide dynamic bus shifters, memory address generators, wide bus multiplexers, and memory-mapped I/O register files. The accumulator can also be used as a synchronous up/down counter. The multiplier can perform logic functions (AND, OR) and barrel shifting.

Input/Output

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

All I/O pins are organized in banks, with 40 pins per bank. Each bank has one common V_{CC0} output supply-voltage pin, which also powers certain input buffers. Some single-ended input buffers require an externally applied reference voltage (V_{REF}). There are two V_{REF} pins per bank (except configuration bank 0). A single bank can have only one V_{REF} voltage value.

I/O Electrical Characteristics

Single-ended outputs use a conventional CMOS push/pull output structure driving High towards V_{CC0} or Low towards ground, and can be put into high-Z state. The system designer can specify the slew rate and the output strength. The input is always active but is usually ignored while the output is active. Each pin can optionally have a weak pull-up or a weak pull-down resistor.

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

Digitally Controlled Impedance

Digitally controlled impedance (DCI) can control the output drive impedance (series termination) or can provide parallel termination of input signals to V_{CC0} , or split (Thevenin) termination to $V_{CC0}/2$. DCI uses two pins per bank as reference pins, but one such pair can also control multiple banks. VRN must be resistively pulled to V_{CC0} , while VRP must be resistively connected to ground. The resistor must be either 1 \times or 2 \times the characteristic trace impedance, typically close to 50 Ω .

I/O Logic

Input and Output Delay

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

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

ISERDES and OSERDES

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

System Monitor

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

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

The System Monitor does not require explicit instantiation in a design. Once the appropriate power supply connections are made, measurement data can be accessed at any time, even pre-configuration or during power down, through the JTAG test access port (TAP).

Low-Power Gigabit Transceivers

Ultra-fast serial data transmission between ICs, over the backplane, or over longer distances is becoming increasingly popular and important. It requires specialized dedicated on-chip circuitry and differential I/O capable of coping with the signal integrity issues at these high data rates.

All but one Virtex-6 device has between 8 to 72 gigabit transceiver circuits. Each GTX transceiver is a combined transmitter and receiver capable of operating at a data rate between 480 Mb/s and 6.6 Gb/s. Lower data rates can be achieved using FPGA logic-based oversampling. Each GTH transceiver is a combined transmitter and receiver capable of operating at a rate between 2.488 Gb/s and 11.18 Gb/s. The GTX transmitter and receiver are independent circuits that use separate PLLs to multiply the reference frequency input by certain programmable numbers between 4 and 25, to become the bit-serial data clock. The GTH transceiver is a purpose-built design for 10 Gb/s rates and shares a single high-performance PLL between four transmitter and receiver circuits. Each GTX and GTH transceiver has a large number of user-definable features and parameters. All of these can be defined during device configuration, and many can also be modified during operation.

Transmitter

The GTX transmitter is fundamentally a parallel-to-serial converter with a conversion ratio of 8, 10, 16, 20, 32, or 40. The GTH transmitter offers bit widths of 16, 20, 32, 40, 64, or 80 to allow additional timing margin for high-performance designs. These transmitter outputs drive the PC board with a single-channel differential current-mode logic (CML) output signal.

TXOUTCLK is the appropriately divided serial data clock and can be used directly to register the parallel data coming from the internal logic. The incoming parallel data is fed through a small FIFO and can optionally be modified with the 8B/10B, 64B/66B, or the 64B/67B (GTX only) algorithm to guarantee a sufficient number of transitions. The bit-serial output signal drives two package pins with complementary CML signals. This output signal pair has programmable signal swing as well as programmable pre-emphasis to compensate for PC board losses and other interconnect characteristics.

Receiver

The receiver is fundamentally a serial-to-parallel converter, changing the incoming bit serial differential signal into a parallel stream of words, each 8, 10, 16, 20, 32, or 40 bits wide. The GTH transceiver offers 16, 20, 32, 40, 64, and 80 bit widths to allow greater timing margin. The receiver takes the incoming differential data stream, feeds it through a programmable equalizer (to compensate for PC board and other interconnect characteristics), and uses the F_{REF} input to initiate clock recognition. There is no need for a separate clock line. The data pattern uses non-return-to-zero (NRZ) encoding and optionally guarantees sufficient data transitions by using the selected encoding scheme. Parallel data is then transferred into the FPGA logic using the RXUSRCLK clock. The serial-to-parallel conversion ratio for GTX transceivers can be 8, 10, 16, 20, 32, or 40. The serial-to-parallel conversion ratio for GTH transceivers can be 16, 20, 32, 40, 64, or 80 for GTH.

Out-of-Band Signaling

The GTX transceivers provide Out-of-Band (OOB) signaling, often used to send low-speed signals from the transmitter to the receiver, while high-speed serial data transmission is not active, typically when the link is in a power-down state or has not been initialized. This benefits PCI Express and SATA/SAS applications.

Integrated Interface Blocks for PCI Express Designs

The PCI Express standard is a packet-based, point-to-point serial interface standard. The differential signal transmission uses an embedded clock, which eliminates the clock-to-data skew problems of traditional wide parallel buses.

The PCI Express Base Specification Revision 2.0 is backwards compatible with Revision 1.1 and defines a configurable raw data rate of 2.5 Gb/s, or 5.0 Gb/s per lane in each direction. To scale bandwidth, the specification allows multiple lanes to be joined to form a larger link between PCI Express devices.

All Virtex-6 devices (except the XC6VLX760) include at least one integrated interface block for PCI Express technology that can be configured as an Endpoint or Root Port, compliant to the PCI Express Base Specification Revision 2.0. The Root Port can be used to build the basis for a compatible Root Complex, to allow custom FPGA-FPGA communication via the PCI Express protocol, and to attach ASSP Endpoint devices such as Fibre Channel HBAs to the FPGA.

This block is highly configurable to system design requirements and can operate 1, 2, 4, or 8 lanes at the 2.5 Gb/s data rate and the 5.0 Gb/s data rate. For high-performance applications, advanced buffering techniques of the block offer a flexible maximum payload size of up to 1024 bytes. The integrated block interfaces to the GTX transceivers for serial connectivity, and to block RAMs for data buffering. Combined, these elements implement the Physical Layer, Data Link Layer, and Transaction Layer of the PCI Express protocol.

Xilinx provides a light-weight, configurable, easy-to-use LogiCORE™ wrapper that ties the various building blocks (the integrated block for PCI Express, the GTX transceivers, block RAM, and clocking resources) into an Endpoint or Root Port solution. The system designer has control over many configurable parameters: lane width, maximum payload size, FPGA logic interface speeds, reference clock frequency, and base address register decoding and filtering.

More information and documentation on solutions for PCI Express designs can be found at:

<http://www.xilinx.com/technology/protocols/pciexpress.htm>

10/100/1000 Mb/s Ethernet Controller (2,500 Mb/s Supported)

An integrated Tri-mode Ethernet MAC (TEMAC) block is easily connected to the FPGA logic, the GTX transceivers, and the SelectIO resources. This TEMAC block saves logic resources and design effort. All of the Virtex-6 devices (except the XC6VLX760) have four TEMAC blocks, implementing the link layer of the OSI protocol stack. The CORE Generator™ software GUI helps to configure flexible interfaces to GTX transceiver or SelectIO technology, to the FPGA logic, and to a microprocessor (when required). The TEMAC is designed to the IEEE Std 802.3-2005 specification. 2,500 Mb/s support is also available.

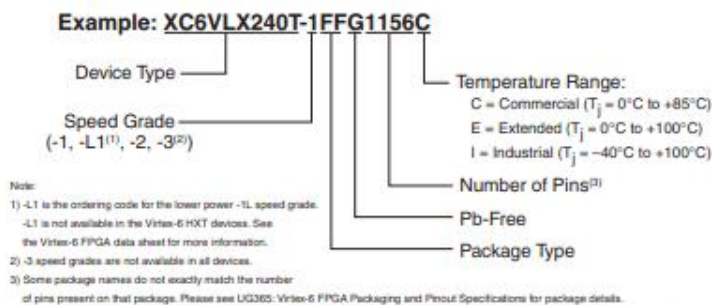
Virtex-6 FPGA Ordering Information

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

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

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

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



DS162_01_102111

Figure 1: Virtex-6 FPGA Ordering Information

Revision History

The following table shows the revision history for this document:

Date	Version	Description of Revisions
02/02/09	1.0	Initial Xilinx release.
05/05/09	1.1	Added the FF1156 package for both the XC6VVSX315T and XC6VVSX475T devices in Table 2, page 3 . Updated the PCI Express design discussion on page 9 to remove the LogiCORE wrapper (<100 LUT) description and clarify 8 lanes at the 5.0 Gb/s data rate. Clerical edits to Global Clock Lines and 10/100/1000 Mb/s Ethernet Controller (2,500 Mb/s Supported) sections. Overall clarifications made in text.
06/24/09	1.2	Added ordering information and FPGA documentation sections.
09/16/09	2.0	Added Virtex-6 HXT family information. Updated number to 26 Mb in Configuration section.
11/06/09	2.1	Clarified distributed RAM features on page 1 . Updated CLB slice number for the XC6VHX565T in Table 1 . Updated compliance to the PCI Express Base Specification Revision 2.0. Updated Integrated Interface Blocks for PCI Express Designs section with link to documentation.
01/28/10	2.2	In Table 1 , there are two Ethernet MACs in the XC6VHX255T. Under Clock Management, page 5 , revised the VCO frequency minimum to 600 MHz which also revised the phase-shift timing increment. Updated GTX transceivers operating data rate range to 6.6 Gb/s. Changed GTX PLL input reference clock frequency divider.
03/24/11	2.3	Changed document classification to Preliminary Product Specification from Advance Product Specification. Updated Figure 1 .
01/19/12	2.4	Changed document classification to Product Specification from Preliminary Product Specification. Updated Configuration, CLBs, Slices, and LUTs, Low-Power Gigabit Transceivers, and Virtex-6 FPGA Ordering Information (including Figure 1) .

Notice of Disclaimer

The information disclosed to you hereunder (the "Materials") is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials, or to advise you of any corrections or update. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of the Limited Warranties which can be viewed at <http://www.xilinx.com/warranty.htm>; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in Critical Applications: <http://www.xilinx.com/warranty.htm#critapps>.

Virtex-6 FPGA Documentation

Complete and up-to-date documentation of the Virtex-6 family of FPGAs is available on the Xilinx website. In addition to the most recent *Virtex-6 Family Overview*, the following files are also available for download:

Virtex-6 FPGA Data Sheet: DC and Switching Characteristics ([DS152](#))

This data sheet contains the DC and Switching Characteristic specifications for the Virtex-6 family.

Virtex-6 FPGA Packaging and Pinout Specifications ([UG365](#))

These specifications includes the tables for device/package combinations and maximum I/Os, pin definitions, pinout tables, pinout diagrams, mechanical drawings, and thermal specifications.

Virtex-6 FPGA Configuration Guide ([UG360](#))

This all-encompassing configuration guide includes chapters on configuration interfaces (serial and parallel), multi-bitstream management, bitstream encryption, boundary-scan and JTAG configuration, and reconfiguration techniques.

Virtex-6 FPGA SelectIO Resources User Guide ([UG361](#))

This guide describes the SelectIO™ resources available in all the Virtex-6 devices.

Virtex-6 FPGA Clocking Resources User Guide ([UG362](#))

This guide describes the clocking resources available in all the Virtex-6 devices, including the MMCM and clock buffers.

Virtex-6 FPGA Memory Resources User Guide ([UG363](#))

This guide describes the Virtex-6 device block RAM and FIFO capabilities.

Virtex-6 FPGA CLB User Guide ([UG364](#))

This guide describes the capabilities of the configurable logic blocks (CLB) available in all Virtex-6 devices.

Virtex-6 FPGA GTX Transceivers User Guide ([UG366](#))

This guide describes the GTX transceivers available in all the Virtex-6 FPGAs except the XC6VLX760.

Virtex-6 FPGA GTH Transceivers User Guide ([UG371](#))

This guide describes the GTH transceivers available in all Virtex-6 HXT FPGAs except the XC6VHX250T and the XC6VHX380T in the FF1154 package.

Virtex-6 FPGA DSP48E1 Slice User Guide ([UG369](#))

This guide describes the architecture of the DSP48E1 slice in Virtex-6 FPGAs and provides configuration examples.

Virtex-6 FPGA Tri-Mode Ethernet MAC User Guide ([UG368](#))

This guide describes the dedicated tri-mode Ethernet media access controller (TEMAC) available in all the Virtex-6 FPGAs except the XC6VLX760.

Virtex-6 FPGA System Monitor User Guide ([UG370](#))

This guide describes the System Monitor functionality.

Virtex-6 FPGA PCB Design Guide ([UG373](#))

This guide provides information on PCB design for Virtex-6 devices, with a focus on strategies for making design decisions at the PCB and interface level.

ÖZGEÇMİŞ

İsmail Koyuncu, 12.10.1981 yılında Bursa Mustafa Kemalpaşa'da doğdu. İlk ve orta eğitimini Mustafa Kemalpaşa'da tamamladı. 1998 yılında Bursa Atatürk E. M. Lisesi Elektrik Bölümünden mezun oldu. 2000 yılında başladığı Kocaeli Üniversitesi Elektrik Eğitimi bölümünü 2004 yılında bitirdi. 2006 yılında Abant İzzet Baysal Üniversitesi'nde Elektrik Eğitiminde Yüksek lisansa ve 2007 yılında Düzce Üniversitesi Düzce Meslek Yüksekokulu Kontrol ve Otomasyon Bölümü'nde öğretim görevlisi olarak çalışmaya başladı. Yüksek lisansını 2008 yılında tamamladı. Ardından Düzce Meslek Yüksekokulu'nda Teknik programlar bölüm başkan yardımcılığı ve bölüm koordinatörlüğü görevlerinde aktif rol aldı. Halen Düzce Üniversitesi, Düzce Meslek Yüksekokulu Kontrol ve Otomasyon Bölümü'nde öğretim görevlisi olarak görev yapmaya devam etmektedir.