



**PROGRAMLANABİLİR KAOTİK
OSİLATÖR TASARIMI**

Fatma ÇULCU

**Yüksek Lisans Tezi
Elektronik Anabilim Dalı
Danışman: Prof. Dr. Mustafa TÜRK**

TEMMUZ-2019

T.C
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

PROGRAMLANABİLİR KAOTİK OSİLATÖR TASARIMI

YÜKSEK LİSANS TEZİ

Fatma ÇULCU

(161113103)

Tezin Enstitüye Verildiği Tarih : 26 Haziran 2019

Tezin Savunulduğu Tarih : 26 Temmuz 2019

Tez Danışmanı : Prof. Dr. Mustafa TÜRK (F.Ü)

Diğer Jüri Üyeleri : Prof. Dr. Arif GÜLTEN (F.Ü)

Dr. Öğr. Üyesi Mehmet ÜSTÜNDAĞ (B.Ü)

TEMMUZ-2019

ÖNSÖZ

Bu tez çalışması, Fırat Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği Elektronik Anabilim Dalı Yüksek Lisans Programı'nda hazırlanmıştır.

Bu tez çalışmasında, çalışmalarına yön veren, ilgi ve katkılarını esirgemeyen tez danışmanım, Sayın Prof. Dr. Mustafa TÜRK'e, Dr. Barış KARAKAYA'ya ve yardımlarıyla çalışmalarımda ilerlememe katkı sağlayan dostum Aybike ÜSTÜNDAĞ'a teşekkürlerimi sunarım.

Fatma ÇULCU
ELAĞIĞ-2019

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ.....	I
İÇİNDEKİLER	II
ÖZET.....	III
SUMMARY	IV
ŞEKİLLER LİSTESİ	V
TABLolar LİSTESİ	VI
KISALTMALAR LİSTESİ	VII
SEMBOLLER LİSTESİ	VIII
1.GİRİŞ	1
1.1. Kaos.....	4
1.2. Tezde İzlenecek Yol	7
2. CHUA DEVRESİ.....	8
2.1.Chua Devresinin Simulink Modeli	11
2.2.Benzetim Sonuçları	17
3. DİFERANSİYEL DENKLEMLERİN AYRIKLAŞTIRMA YÖNTEMLERİ.....	20
3.1.Taylor Serisi Genişleme Metodu.....	21
3.2. Runge-Kutta Ayırıklaştırma Yöntemi	23
3.2.1. İkinci Dereceden Runge-Kutta Yöntemi	24
3.3. Euler Ayırıklaştırma Yöntemi	27
4. XILINX SYSTEM GENERATOR TEKNOLOJİSİ.....	30
4.1. Sabit Nokta Sayı Temsil Biçimi	31
4.2.FPGA.....	33
5. XILINX SYSTEM GENERATOR'ÜN SIMULINK GÖSTERİMİ.....	38
5.1.Xilinx System Generator'ün Donanım Temelli Benzetimi	44
5.1.1. Kintex-7 FPGA Geliştirme Kartı	45
6. SONUÇLAR.....	53
KAYNAKLAR.....	55
ÖZGEÇMİŞ.....	59

ÖZET

Kaotik sistemler doğrusal olmayan dinamik sistemler olup en önemli özellikleri başlangıç şartlarına hassas duyarlılıkları, periyodik olmayan davranışlar sergilemeleri, gürültü benzeri geniş ve dağınık spektruma sahip olmalarıdır. Sistemlerin istenmeyen kaotik davranışlarını kontrol altına alan ve sistemlerin kaotik davranışlarını engelleyen çeşitli kontrol mekanizmaları ve algoritmaları geliştirilmiştir.

Bu tez çalışmasında, Kintex 7 FPGA geliştirme kartında Chua kaotik sistemin Xilinx System Generator teknolojisi kullanılarak uygulanması sunulmuştur. Herhangi bir doğrusal olmayan sistemin diferansiyel denklemleri, FPGA editöründe kodlama ve tasarım işlemi öncesinde ayrık hale getirilmelidir. Chua kaotik sistemi, sürekli oluşturulmuş sinyalleri ayırmak için Euler ayrıklaştırma yöntemi kullanılarak ayrık hale getirilmiştir.

Bu tezde, doğrusal olmayan devrelerde çok parçalı doğrusal direnç fonksiyonlarını gerçekleştirmek için sistematik bir yaklaşım önerilmiş ve geliştirilmiştir. Çift ve tek sayıda kaydırmalı kaotik çekiciler, önerilen gerçekleştirme şeması ve devre tasarımı izlenerek Chua devresinde deneysel olarak kolayca üretilebilir. Çift veya tek sayıda kaydırma yapan çekicilerin üretilmesi için tek farkın bir rezistörün bağlantısını kesmesi olduğu bulunmuştur. Yalnızca kaotik çeker sayısını giriş parametresi olarak ele eden ve 5-10 arası değişen çoklu kaotik çeker davranışları gösteren bir Matlab/Simulink modelinden yola çıkılarak XSG platformunda yeniden derlenerek bu platformda model yeniden incelenmiştir. Daha sonra FPGA geliştirme kartı sisteme tanıtılarak tek ve çift çeker değerleri için donanımsal benzetim yoluna gidilmiştir. Ele alınan kaotik sistemde her davranış için ayrı bir doğrusal olmayan eleman kullanımı gerekmektedir. Bu doğrusal olmayan eleman değerleri Simulink ve XSG platformunda değişiklik göstermektedir. Bu değerlerde modelin göstermiş olduğu kaotik davranışa göre düzenlemeler yapılmıştır. Gerçekleştirilen modelde tek bir PPD yapısı kullanılmıştır. Girdi olarak sadece çeker sayısının girilmesi yeterli olmaktadır. Modelde mümkün olabildiğince basit yapıdaki bloklar kullanılarak daha iyi anlaşılabilmesi açısından alt sistem blokları kullanılmıştır.

Anahtar Kelimeler: Kaos, FPGA, MATLAB/Simulink, Xilinx System Generator, Çoklu kaotik çekerler.

SUMMARY

Programmable Chaotic Oscillator Design

Chaotic systems are nonlinear dynamical systems and the most important features of the chaotic systems are sensitivity on initial conditions, having nonperiodic behavior and noise like wideband spread spectrum. Several control mechanisms and algorithms have been developed to control the unwanted chaotic behavior of the systems and to prevent the chaotic behavior of the systems.

In this thesis, application of Chua chaotic system using Xilinx System Generator technology in Kintex 7 FPGA development board is presented. Differential equations of any nonlinear system should be discrete in the FPGA editor prior to coding and design. The Chua chaotic system was discrete using the Euler discretization method to separate continuously generated signals.

In this thesis, a systematic approach to perform multi-part linear resistance functions in nonlinear circuits is proposed and developed. The even and odd number of sliding chaotic attractors can be easily produced experimentally in the Chua circuit following the proposed implementation scheme and circuit design. It has been found that the only difference is to disconnect a resistor to produce even or even number of shifting attractors. The model was re-compiled in XSG platform based on a Matlab / Simulink model that handled the chaotic fume number as an input parameter and showed multiple chaotic fume behaviors ranging from 5-10. Then, FPGA development card was introduced to the system and hardware simulation path was applied for single and double drive values. In the chaotic system discussed, a separate nonlinear element is required for each behavior. These nonlinear element values vary on the Simulink and XSG platforms. In these values, adjustments were made according to the chaotic behavior of the model. In the realized model, a single PPD structure was used. It is sufficient to enter only the number of fumes as input. In the model, subsystem blocks are used in order to better understand them by using blocks as simple as possible.

Keywords: Chaos, FPGA, Matlab/Simulink, Xilinx System Generator, Multi-Scroll Chaotic Attractors.

ŞEKİLLER LİSTESİ

Sayfa No

Şekil 2.1. Chua Devresi	9
Şekil 2.2. Doğrusal olmayan direncin karakteristiği	9
Şekil 2.3. Tüm Sistemin Simulink Modeli	12
Şekil 2.4. Diferansiyel denklemlerin çözümü için gerçekleştirilen Tank Resonator modeli	13
Şekil 2.5. PPD Elemanın Simulink Modeli	15
Şekil 2.6. PPD elemanların alt yapı bloğu	16
Şekil 2.7. Kontrol Bloğu	17
Şekil 2.8. (a) 10 çeker (b) 9 çeker davranışı için elde edilmiş olan PPD elemanın	19
karakteristiği, Faz uzay diyagramları (c) 10-çeker, (d) 9-çeker	19
Şekil 3.1. Runge- Kutta yöntemi	23
Şekil 3.2. Euler Yöntemi	28
Şekil 3.3. Euler Yönteminin XSG 'de uygulanması	29
Şekil 4.1. Q m.n sabit nokta sayı gösterim biçimi	32
Şekil 4.2. Xilinx System Generator ile tasarım metodolojisi	37
Şekil 5.1. Tüm sistemin XSG Simulink modeli	40
Şekil 5.2. Euler Ayırıklaştırma Yöntemi Uygulanmış Tank Resonator Modeli	40
Şekil 5.3. PPD elemanın PWL_Çift Simulink modeli	41
Şekil 5.4. PPD elemanın PWL_Tek Simulink modeli	42
Şekil 5.5. PPD elemanın alt yapı bloğu	42
Şekil 5.6. Kontrol bloğu yapısı	43
Şekil 5.7. Sistemin XSG çıkışları	44
Şekil 5.8. Programlanabilir kaotik osilatör tasarımı uygulamasının tasarım akış şeması	45
Şekil 5.9. Kintex-7 FPGA Geliştirme Kartı	46
Şekil 5.10. System Generator Blok Penceresi	47
Şekil 5.11. Kaynak Kullanımı	48
Şekil 5.12. Tüm sistemin tek bir yapıda XSG Simulink modeli	49
Şekil 5.13. Modelin 5 çeker için XSG simulink modeli	50
Şekil 5.14. Modelin 5 çeker için XSG çıkışı	50
Şekil 5.15. XSG platformunda 5 çeker için FPGA'da uygulama görünümü	51
Şekil 5.16. Modelin 5 çeker için FPGA çıkışı	52

TABLÖLAR LİSTESİ

Sayfa No

Tablo 1.1. Kaos konusundaki çalışmalarda tarihsel dönüm noktaları	6
Tablo 2.1. Denklem (2.1-2)' deki yapı için kullanılan parametreler	11
Tablo 5.1. XSG Platformunda Denklem (2.1-2)' deki yapı için kullanılan parametreler ..	39
Tablo 5.2. Ürün Özellikleri.....	46



KISALTMALAR LİSTESİ

ASIC	:	Uygulamaya özel tümleşik devreler
ASSPs	:	Uygulamaya özel standart parçalar
CLB	:	Konfigüre edilebilir mantık blokları
CSP	:	İletişim ardışık süreci
DSP	:	Dijital sinyal işleme
FFT	:	Hata düzeltim blokları
FPGA	:	Alanda programlanabilir kapı dizileri
HDL	:	Donanım tanımlama dili
PPD	:	Parça-parça doğrusal
RSÜ	:	Rasgele sayı üreteçleri
SCM	:	Tek sabit çarpımlı blok
UCF	:	Kullanıcı kısıtlamaları dosyası
XSG	:	Xilinx System Generator

SEMBOLLER LİSTESİ

C_1, C_2	:	Kondansatör
I_L	:	İndüktör akımı
N_R	:	Doğrusal olmayan direnç
V_1, V_2	:	Kondansatörlerin gerilimleri
f	:	Sabit noktalı sayının ondalık kısmı
i	:	Sabit noktalı sayının tam kısmı
m	:	n bitlik sabit noktalı sayının noktadan sonraki bit sayısı
n	:	n bitlik sabit noktalı sayının tam sayı değeri
N	:	Kaydırma çekici
$Q_{m,n}$	:	Sabit nokta sayı gösterim biçimi
R	:	Direnç
Δh	:	Adım sayısı

1.GİRİŞ

Kaos, en kısa ifadeyle, düzensizliğin düzeni olarak ifade edilen ve doğrusal olmayan olayları açıklayan bir bilim dalı olarak adlandırılabilir. Karmaşıklığı ifade etse de kendi iç düzeninde bir sürece sahiptir. Burada dikkat edilmesi gereken husus kaosun rastgele olmadığıdır. Dinamik sistemlerde bilinen en karmaşık kararlı durum davranışı kaostur. Kaos üzerine yapılan çalışmalar doğrusal olmayan dinamik sistemler teorisinin bir parçasıdır. Bu durum “deterministik kaos” olarak bilinir. Hesaplanamayan nedeni ve seyri bilinmeyen rastlatısal kaos olan kavramı da vardır. Ancak bilim, deterministik kaos ile daha fazla ilgilidir [1].

Günümüzde kaotik sistemler ve kaos tabanlı uygulamalar mühendislikte yaygın olarak kullanılmaktadır. Bu uygulamalarda kullanılan ana yapılardan biri kaos temelli sinyal üreticileridir. Kaotik sinyal üreticileri, bilhassa kaotik iletişim ve kriptolojide önemli bir rol oynamaktadır [2]. Kaos teorisi, matematik, mühendislik, şifreleme, iletişim, görüntü ve sinyal işleme gibi birçok farklı alanda uygulama bulduğu için son yıllarda en çok çalışılan konulardan biri haline gelmiştir [3].

Kaotik sistemin temel özelliği, başlangıç koşullarına son derece hassas olması ve başlangıç durumundaki küçük farkın, sistem davranışında olağanüstü farklılıklara yol açabilmesidir. Kaotik davranış karmaşık ve düzensizdir. Genel olarak mekanik sistemlerde istenmez. Birçok mekanik sistem uygulaması, karmaşıklığı en aza indiren ve sistem performansını iyileştirmek için istenmeyen davranışları ortadan kaldıran bir kontrol ünitesi gerektirir. Bununla birlikte, kaotik davranış, güvenli iletişim ve kriptografik sistemler gibi karmaşıklığın gerekli olduğu bazı alanlarda yararlı olabilir [4]. Kaotik alıcılar ve vericiler karmaşık olmayan bir yapıya, düşük dereceye sahip olmalı ve birden fazla kaotik davranışlar sergilemelidir. En yaygın kullanılan kaotik osilatör, Chua devresidir, ancak çeşitli kaotik davranışlar göstermez [7].

Yapılması gereken davranış tam olarak tespit edilebiliyorsa, sistemin tüm durumlarının ve parametrelerinin belirlenebileceği deterministik sistemlerde bilindik en kompleks dinamiklere sahip kaotik sistemler, tahmin edilmesi sistemin en zor sınıfını oluşturur [5].

Ayrık zamanda test edilen sinyallere bu sistemlerde kaotik zaman serisi denir. Kaotik zaman serilerinin tahmini cebirsel fonksiyona bir yaklaşım sorunu olarak görülebilir [6].

Sınırsız periyodik olmayan çözümlerden oluşan kaotik sistemler, başlangıç koşullarına ve sistem parametrelerine bağımlı oldukları için artan bir ilgi konusu haline geldi. Sonuç olarak, kaotik sistemlerin uygulanması ile ilgili bilimsel ve endüstriyel alanlarda önemli araştırmalar yapılmaktadır [2]. Bugün fizikçiler galaksinin oluşumunu kaos yardımıyla açıklamakta olup bu konu hakkında çeşitli çalışmalar mevcuttur. Bu teoriyle metalurjik tahminleri, kanserli hücrelerinin denetlenmesi, teşhis ve tedavi edilmesi gibi çeşitli pratik problemlerde analiz edilebilir. Bu tür sistemler doğrusal değildir ve sistemleri daha önceki girdi ve çıktı verilerinden modellemeye yöneliktir [6].

Kaotik sistemin analizinde, sürekli zaman ve ayrık zaman modelleme olmak üzere iki tür temsille karşılaşılmaktadır. Dijital uygulamalarda, dijital işlemcilerde sistem davranışını işlemek için ayrık zamanlı modelleme kullanılmalıdır. Bu amaçla, literatürde birçok ayrıklaştırma yöntemi vardır [4].

Dijital işleme için bir ayrıklaştırma yöntemi kullanılırken, tasarımın doğruluğu ve kaynak kullanımını gerektirip gerektirmediği kullanıcı tarafından dikkate alınmalıdır. Bu çalışmada diferansiyel denklemlerin ayrıklaştırma yöntemleri olan Taylor Serisi Genişletme, İleri Euler ve Runge-Kutta kullanılmış ve sonuçlar karşılaştırılmış ve denklemler İleri Euler kullanılarak ayrıklaştırılmıştır. Optimum parçalanma yönteminin seçimi, istenen performansa sahip olması bakımından önemlidir.

Alanda Programlanabilir Kapı Dizileri (FPGA) teknolojisi, hız performansı yüksek paralel işleme ve uygun maliyete sahip saha yeteneklerine dayanan tasarımcılar tarafından çoğunlukla tercih edilir. FPGA, programlanabilir lojik kapılardan oluşmuş olup bu kapılar arasındaki ara bağlantılardan oluşan dijital bir entegre devredir. Ayrıca FPGA'nın uygulama alanı oldukça geniştir. Matematiksel denklemlerin gerçekleştirilmesi için FPGA kullanıldığında, matematiksel modeli tasarlamak için kesirli sayı gösterim formatı düzenlenmelidir. İki tür kesirli sayı gösterimi vardır: sabit nokta ve kayan nokta.

Kaos temelli teknolojide ve bilgi sistemlerinde bilhassa komplike davranışlar gösteren kaotik osilatörlere duyulan ihtiyaç nedeniyle ilgili kişiler, çoklu kaotik çekiciler, iki, üç ve dört boyut gibi değişik davranış türlerinin olduğunu araştırmışlardır. Son yıllarda, birden fazla kaotik çekicinin meydana getirilmesi ve bunların elektronik devreler olarak gerçekleştirilmesi üzerine birçok çalışma yapıldı. Polinom fonksiyonları, zaman

gecikmeli sistemler veya parça parça lineer (PPD) fonksiyonlar yaygın olarak kullanılan yaklaşımlardır [23]. En azından PPD sistemlerinin doğrusal bölgelerinde analitik çözümler bulunduğundan, bu konudaki çalışmalara literatürde daha sık rastlanmaktadır. Bu tür yapıların modellenmesi de kolaydır[8]. Bu durum yeni PPD bazlı kaotik osilatörlerin gelişmesine ve yeni tiplerde çok sayıda kaotik çekicinin meydana gelmesine fırsat verdi [3].

Kaos tabanlı bilgi sistemlerinin gelecekteki uygulamaları için ana bileşen, çeşitli kaotik sinyaller üretmek için güvenilir doğrusal olmayan devrelerin donanım uygulamasıdır. Bu, düşük dereceli polinomlu doğrusal olmayan elektronik cihazlar kullanarak karmaşık çoklu kaotik çekerlerin üretilmesine yönelik araştırmaları teşvik eder. Bu amaçla, çeşitli farklı yaklaşımlar önerilmiştir ve literatürde bulunabilir. Özellikle, n-scroll kaotik çekerlerin oluşturulması araştırma ve tasarım için ilginç bir konudur [9].

Birçok araştırmacı, sistemi mümkün olduğunca basit tutarak kaotik sinyallerin karmaşıklığını arttırmaya çalıştı. N çift çoklu kaotik çeker denilen daha karmaşık kaotik sinyaller Suykens ve Vandewalle (1993) tarafından önerilmiştir. Daha sonra Suykens ve diğ. (1997). Bu çalışmalarda ana fikir doğrusal olmayanlara ilave sınır değerleri getirmektir. Bazı araştırmacılar, genelleştirilmiş Chua'nın devre denklemlerini kullanarak tüm durum değişkenleri için kaotik çekiciler meydana getirmenin mümkün olduğunu göstermiştir [7].

N kaotik çekerleri üretmek için Chua devresine dayalı yeni bir devre tasarımı önerildi ($n = 1; 2; 3; \dots$). Bu tasarımda, Chua devresindeki doğrusal olmayan direnç, ana yapı taşları kullanılarak sistematik bir prosedürle üretilir. Önerilen yapım şemasıyla, devrenin karakteristiğinin V_i eğimi ve kırılma noktaları birbirinden bağımsız olarak ayarlanabilir ve eşit veya tek sayıda kaymalı kaotik çekerler kolayca oluşturulabilir. Bu basit deneysel kurulumla elde edilen n-kaymalı ($n = 5; 6; 7; 8; 9; 10$) farklı çeker sayıları gösterilmektedir [9].

En iyi bilinen elektronik devrelerden biri, Chua'nın osilatörüdür. Chua devresi literatürdeki en basitlerinden biri olmasına rağmen, onu ayrıntılı bir çalışmaya konu yapan çeşitli karmaşık kaotik dinamik özelliklere sahip olmasıdır. Devrenin değiştirilmiş bir versiyonu da dikkat çekicidir. Teorik analizi ve sayısal simülasyonları deneysel sonuçlarla iyi bir şekilde eşleştirilmiştir [22].

Bu çalışmada, sadece kaotik çeker sayısını dışarıdan girdi olarak uygulayıp 5 ile 10 arasında değişen birden fazla kaotik davranışı ele alan bir model geliştirilmiştir. Model

Xilinx Sistem Üreticisi (XSG) platformunda Xilinx Blokset kütüphanesi kullanılarak derlendi. Modelde, tek ve çift çeker sayısına bağlı olarak değişen iki farklı doğrusal olmayan direnç modeli oluşturulmuştur aynı zamanda bu ayrı direnç modelleri aynı yapıyla çalışmaktadır. Simulink modelinde, voltaj kontrollü anahtarlar kullanılarak tek ve çift çeker sayılarına bağlı olarak kaotik davranışlar elde edilmiştir. Xilinx Sistem Üreticisi'nde uygulanan modelde, Xilinx Blockset kütüphanesinde gerilim kontrollü anahtarlar yerine Relation parametresi ile istenen durum sağlanmaya çalışılmıştır. Modelin sağladığı avantaj, tek bir yapıyla altı değişik çoklu kaotik davranışı gösterebilmesidir. Bu bakımdan uygulanan model aktif bir yapıya sahiptir denilebilir.

1.1. Kaos

Doğrusal olmayan bilim olarak nitelendirebileceğimiz kaos terimi, dünyadaki doğrusal olmayan sistemlerin ilişkilerini ortaya çıkaran ve bu sistemleri modellemeye çalışan bilimdir. Bu sistemlerdeki düşük tahmin edilen davranış veya etki, sistemde öngörülemeyen değişikliklere ve reaksiyonlara neden olabilir. Doğrusallık sadece belirli sınırlar arasında geçerli olabilir. Günümüzde, araştırılmış ve çalışılmış doğrusal olmayan bilimlerden biri kaos bilimi veya kaotik sistemlerdir [10].

Kaos, başlangıç koşullarına duyarlı deterministik bir sistemde aperiodik olarak uzun vadeli davranış olarak tanımlanabilir. Aperiodik uzun vadeli davranış, sistemin yörüngelerinin sabit noktalara veya periyodik yörüngelere yerleşmediği anlamına gelir, çünkü zaman sonsuzluğa meyillidir [11]. Konseptin en tanımlayıcı tanımlarından birini veren teorik fizikçi Jensen, kaosu karmaşık, doğrusal olmayan dinamik sistemlerin düzensiz ve öngörülemeyen davranışlarını kapsayacak şekilde tanımlar. Tanımdaki karmaşık ifade, karmaşıklığa, doğrusal olmayan ifade, benzersiz bir matematiksel yapıya ve dinamik ifadeye, sabit olmayan bir değişken yapıya atıfta bulunur [12]. Kaotik sistemlerin başlangıç ve giriş koşullarına bu duyarlılığı nedeniyle, bu verilerdeki küçük değişiklikler sistemin çıktısının çok farklı noktalara ulaşmasına neden olur. Bu davranış nedeniyle, kaotik sistemler deterministik sistemler olsa bile, sistemin davranışını yalnızca kısa bir süre için ölçebilir. Daha sonraki yinelemelerde, kaotik sistemlerin davranışı tahmin edilemez. Bu nedenle, kaotik sistemler rastgele davranış sergilemektedir. Kaotik sistemlerin rastgele davranışları nedeniyle yeni analiz ve modelleme teknikleri geliştirilmiştir. Bu gelişmeler sayesinde dinamik sistem teorisi daha da geliştirilmiştir ve anlaşılması kolay olmayan bazı fiziksel ve doğal olaylar açıklanmıştır [13].

1900'lerde matematiksel olarak araştırılmaya başlanan ilk ciddi ve bilimsel çalışma, 1961'de Edward Lorenz'in matematiksel meteorolojik modelinin 1961'de hava tahmini için elde edilen sonuçlarından elde edildi. Edward Lorenz, sonuçlarını hızlandırmak için elde edilen verileri toplayarak sayısal analiz sonuçlarını modellemek için bilgisayarını kullandı. Ancak, sonuçlar çok hızlı bir şekilde değişti ve tahmin edilemez hale geldi. Lorenz böylece istemeden kaos teorisinin temellerini attı. Bu gelişme doğrultusunda, kaos teorisi o zamana kadar çözülemeyen birçok fiziksel ve matematik problemini çözmüştür. Günümüzde kaos kavramı astronomi, biyofizik, biyoloji, fizik, jeoloji, kimya, matematik, meteoroloji, mühendislik, plazma, tıp ve hatta sosyal bilimlerdeki fenomeni açıklamak zor modellemesinde ve çözümlemesinde kullanılmaktadır.

Kaotik davranışın elektronik modellenmesi ilk olarak 1983 yılında elektronik elemanlar ve devreler kullanılarak Leon O. Chua tarafından gerçekleştirildi. Tasarımcısının ismini taşıyan bu özerk Chua devresi, anlaşılması ve tasarlanması çok basit bir elektronik devredir. Bu özellikler nedeniyle, Chua devresi birçok uygulamada birçok kez kullanılmıştır ve günümüzün kaotik devre çalışmalarına ışık tutmuştur [13].

Kaos tabanlı sistemler gürültü benzeri sinyaller üreterek, periyodik davranış sergileyen ve başlangıç koşullarına hassas bağımlılık göstererek kararsız görünmektedir. Bu özellikleri nedeniyle güvenli iletişim, şifreleme ve rasgele sayı üretimi gibi elektronik mühendisliği uygulamalarının dikkatini çekmiştir [14].

Kısaca, kaosa ilişkin dönüm noktaları Tablo 1'de gösterilmiştir [12].

Tablo 1.1. Kaos konusundaki çalışmalarda tarihsel dönüm noktaları

1890	Henri Poincare n –cisim problemi ile gezegenleri izleyeceği yörüngelerin karmaşık ve tahmin edilemez olabileceğini ilk vurgulayan kişi oldu.(Kaos terimi henüz kullanılmamaktaydı.)
1963	Edward Lorenz atmosferik olaylara yönelik basit bir dinamik sistemde ilk kaotik ve garip çakeri elde etti.
1975	Biyolojik popölasyon artışlarında kaosu tespit edilmesi(Robert May)
1976	Tien-Yien Li ve James Yorke “Priyot Three Implies Shaos” aslı makaleleri ile kaos kelimesini literatüre kazandırdı.
1978	Mitchell Feigenbaum kaosa geçişteki evrensel sabiti keşfetti.
1980	Benoit Mandelbrot fraktal geometrinin bilgisayar grafikleri ile ilgili çalışmalarda bulundu.
1990	Kaos teorisinin başlangıcı
1990	Kaos sistemlerinin senkronizasyonu

Kaotik sistemlerin en önemli özelliği başlangıç koşullarına hassas bağımlılıktır. Bu özellik, başlangıç koşullarındaki en küçük bir değişikliğin çıktıda tamamen farklı bir davranışa neden olabileceği anlamına gelir. Diğer özellikler olarak; (Parker ve Chua 1987).

- i) Yörüngeler sınırlı,
- ii) kaotik bir davranış için belirlenen sınır, fraktaller ve Cantor kümeleriyle ilgilidir,
- iii) kaotik sistemler ses benzeri spektrumlara sahiptir[11].

Kaotik sistemlerin küçük bozulmalara karşı hassasiyetinin sistemleri dengelemek veya kontrol etmek için kullanılabileceği görülmüştür. Kaotik bir sistem sonsuz sayıda dengersiz periyodik yörüngeden oluştuğundan, yörüngeler istenen bir periyodik yörüngeye yönlendirilebilir. Bu, yörünge istenen periyodik yörüngeye yaklaştığında sistemi kapalı yörüngede yapmak için yeterli giriş parametrelerini bozmak suretiyle yapılır. Kaotik davranış sergileyen epileptik beyin nöbetlerini kontrol etmek için benzer çabalar gösteriliyor [11].

1.2. Tezde İzlenecek Yol

Bu çalışmada, literatürde sunulan sürekli zaman özerk 3 boyuta sahip kaotik sistemi, FPGA'da sabit nokta sayı biçiminde Euler ayrıklaştırma yöntemi kullanılarak Chua devresinin XSG platformunda modelleme yoluna gidilmiştir. Bu sistemler Xilinx Kintex-7 ailesi FPGA yongasında sentezlenir ve test edilir. FPGA'da tasarlanan kaotik osilatörlerin performans analizi Matlab / Simulink sonuçları ile karşılaştırıldı ve başarılı sonuçların elde edildiği tespit edildi. Bilgisayar ortamındaki kaotik osilatörlerin simülasyonu ile FPGA'da gerçekleştirilen tasarım aynı sonuçları verir ve bu sistemlerin donanım sentezi sağlanır. Bu şekilde, Denklem (2.1) 'deki diferansiyel denklem, ayrık zamanlı FPGA tabanlı kaotik osilatörler ve gömülü kaos tabanlı mühendislik uygulamaları ile tasarlanmıştır.

Referans model olarak [9] 'daki model kullanıldı. Model, çeker sayısının değişimine bağlı olarak daima değişmesi gerekli olan bir dizi diferansiyel denklem ve doğrusal olmayan dirençten meydana gelir. Chua devresindeki doğrusal olmayan direnç, temel yapı taşları kullanılarak sistematik bir prosedürle oluşturulur. Önerilen yapım şemasıyla, devrenin karakteristiğinin eğimleri ve kırılma noktaları bağımsız olarak ayarlanabilir ve bir çift veya hatta birkaç kaotik çeker kaydırılarak kolayca oluşturulabilir.

İkinci bölümde, öncelikle bu modele ilişkin matematiksel ilişki kurulmuş olup ilgili parametreler verilmiştir. Tek bir modelde tüm davranışları elde etmek için kullanılan PPD direncinin modellenmesi verilmiştir. Modellenen PPD direncinin doğruluğu tespit edildi ve gözlemler sonucu oluşan kaotik çekerler Matlab Simulink ortamında gösterildi.

Üçüncü bölümde referans denkleminde özelleştirme yöntemleri uygulanmış ve özelleştirme yöntemleri hakkında bilgi verilmiştir. Bu çalışmada, Xilinx System Generator'da kullanılan Euler ayrıklaştırma yönteminin sonuçları incelenmiştir.

Dördüncü bölümde, FPGA hakkında bilgi verilmiş ve XSG teknolojisi anlatılmıştır.

Beşinci bölümde, referans modelimiz XSG Platformunda uygulanmıştır. Ortaya çıkan kaotik davranışlar gözlemlendi.

Altıncı bölümde, modelin FPGA çıkışı FPGA kullanılarak test edildi ve XSG çıkışı ile karşılaştırıldı.

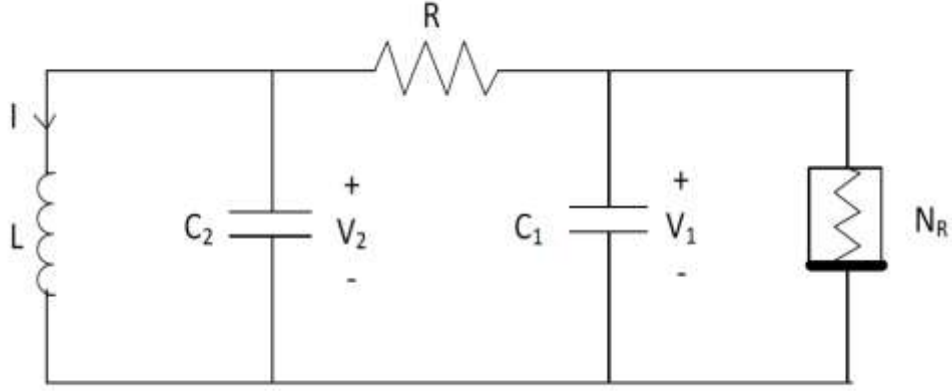
2. CHUA DEVRESİ

Chua devresi, kaos gibi çok çeşitli doğrusal olmayan dinamik olayları gösteren basit bir elektronik devredir. Sadeliği ve evrenselliği nedeniyle, Chua devresi büyük ilgi gördü ve kaos araştırmalarında standart bir öncelik haline geldi. 1983 yılında, daha sonra Japonya'daki Waseda Üniversitesi'nde misafir olan Leon O. Chua tarafından tanıtıldı. Chua devresi, otonom bir üçüncü dereceden doğrusal olmayan elektronik devredir. Standart bileşenlerden (dirençler, kapasitörler, indüktörler) oluşturulan basit otonom bir devreden kaotik bir davranış elde eder [11].

Son yıllarda, birçok Chua devreleri gerçekleştirildi. Chua devresi ile senkronizasyon kullanarak güvenli iletişim alanındaki pratik kaotik sistemleri açıklamak gibi birçok önemli çalışma vardır [24]. Tasarımcısının adını taşıyan bu sürekli özerk Chua devresi literatürde birçok kez çalışılmıştır, çünkü elektronik olarak oldukça kolay anlaşılan ve elektronik olarak tasarlanan ve günümüzün kaotik devre çalışmalarına ışık tutan çok basit bir devredir.

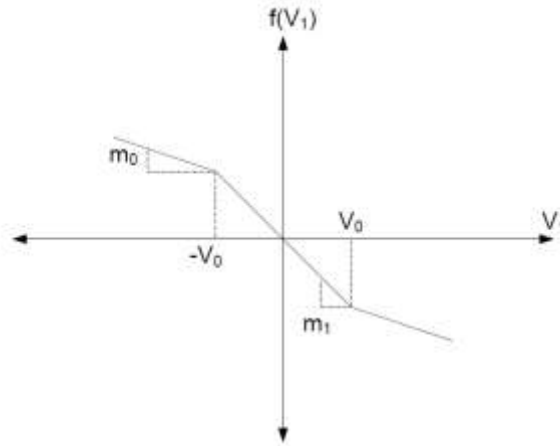
Chua devresi en az bir doğrusal olmayan eleman, en az bir yerel olarak aktif direnç ve en az üç enerji depolama elemanı taşıma kriterlerini karşılayan en basit elektronik devredir. Ek olarak, bu olağanüstü devre, kaos varlığının matematiksel olarak kanıtlandığı tek fiziksel sistemdir. Standart elektronik bileşenler kullanılarak devre kolayca düşük maliyetle yapılır ve çok çeşitli çatallanma ve kaos sergiler [24].

Literatürdeki en iyi tanınan kaotik osilatör olarak adlandırılanlardan biri, Chua devresidir. Chua devresi bir tank rezonatörü, bir alçak geçiren filtre ve doğrusal olmayan bir dirençten oluşur. Chua devresinde kullanılan doğrusal olmayan direnç elemanı, üç bölgeye sahip bir element olup PPD yapılıdır. Bu devrede elde edilebilecek davranış, bu doğrusal olmayan elemandaki sınırlı kırılma noktalarından dolayı sınır döngüsü, periyodik pencereler, spiral çekme ve çift çekme gibi belirli sayıda davranıştır. Bu devrede kırılma noktalarının sayısını arttırarak veya PPD yapısının içerisindeki doğrusal bölgelerin sayısı arttırılarak davranış çeşitliliği arttırılabilir. Bu yaklaşımla, Denklem (2.1) 'de verilen yapı iki farklı tip PPD yapısı kullanılarak geliştirilmiştir [9].



Şekil 2.1. Chua Devresi

Devrede bulunan R direnci N_R ile C_2 arasındaki iletkenliği sağlarken, devrenin doğrusal olmayan bileşeni olan N_R direnci, üç segmentli parçalı doğrusal direnç özelliği gösterir. Bu bileşene ait transfer I - V karakteristiği Şekil 2.2’de gösterilmiştir. Chua devresi kaosu araması için gerekli olan şartları barındıran en basit elektronik devredir [12].



Şekil 2.2. Doğrusal olmayan direncin karakteristiği

Paralel RLC devresi, dinamik bir sistemde salınımın başlangıcını modelleyebilen en düşük dereceli sistem olduğu için, Chua devresi, yüksek dereceli sistemlerin zengin periyodik olmayan dinamiklerini yakalayabilen en düşük dereceli sistemdir [25].

Chua devresi, adım fonksiyonlarını, delta fonksiyonlarını ve bunların x türevleriyle ürünlerini içeren dört terimden daha fazlasıyla daha karmaşık bir temsile sahiptir. Chua

devresini inşa etmek, rastgele frekanslara ölçeklendirmek ve frekans bağımlı direnç kayıpları ile indüktör nedeniyle analiz etmek daha zordur[26] .

Bu sistemde lazım olan diferansiyel denklemleri elde etmek için değeri zamanla değişen üç parametre olarak V_1 ve V_2 kondansatörlerin gerilimleri ve I_L ise indüktör akımı seçilir. Otonom Chua devresinin dinamiği, Denklem (2.1)'de üç diferansiyel denklem takımı ile açıklanmıştır:

$$\begin{aligned}\frac{dI_L}{dt} &= -\frac{1}{L}V_2 \\ \frac{dV_2}{dt} &= \frac{1}{C_2}I_L - \frac{1}{RC_2}(V_2 - V_1) \\ \frac{dV_1}{dt} &= \frac{1}{RC_1}(V_2 - V_1) - \frac{1}{C_1}f(V_1)\end{aligned}\tag{2.1}$$

$f(V_1)$ olarak adlandırılan PPD eleman tek ve çift sayıda çeker elde edebilmek için iki farklı matematiksel işlem olarak Denklem (2.2)'de verildiği gibi seçilir. Birinci ifade çift sayıda, ikinci ifade ise tek sayıda çeker elde etmek için kullanılır.

$$\begin{aligned}f(V_1) &= m_{2n-1}V_1 + \frac{1}{2} \sum_{i=1}^{2n-1} (m_{i-1} - m_i)(|V_1 + b_i| - |V_1 - b_i|) \\ f(V_1) &= m_{2n-1}V_1 + \frac{1}{2} \sum_{i=2}^{2n-1} (m_{i-1} - m_i)(|V_1 + b_i| - |V_1 - b_i|)\end{aligned}\tag{2.2}$$

Denklem (2.2)'de verilen parametreler; $m_0 = -3.2$ mS, $m_1=m_3=m_5=m_7=m_9 = -0.2$ mS, $m_2=m_4 = m_6=m_8 = -2.2$ mS, $C_1=10$ nF, $C_2=100$ nF, $L=18.68$ mH olarak seçilir. Elde edilen çekerler ve kullanılan parametreler Tablo 2.1'de özetlenmiştir.

Tablo 2.1. Denklem (2.1-2)' deki yapı için kullanılan parametreler [3]

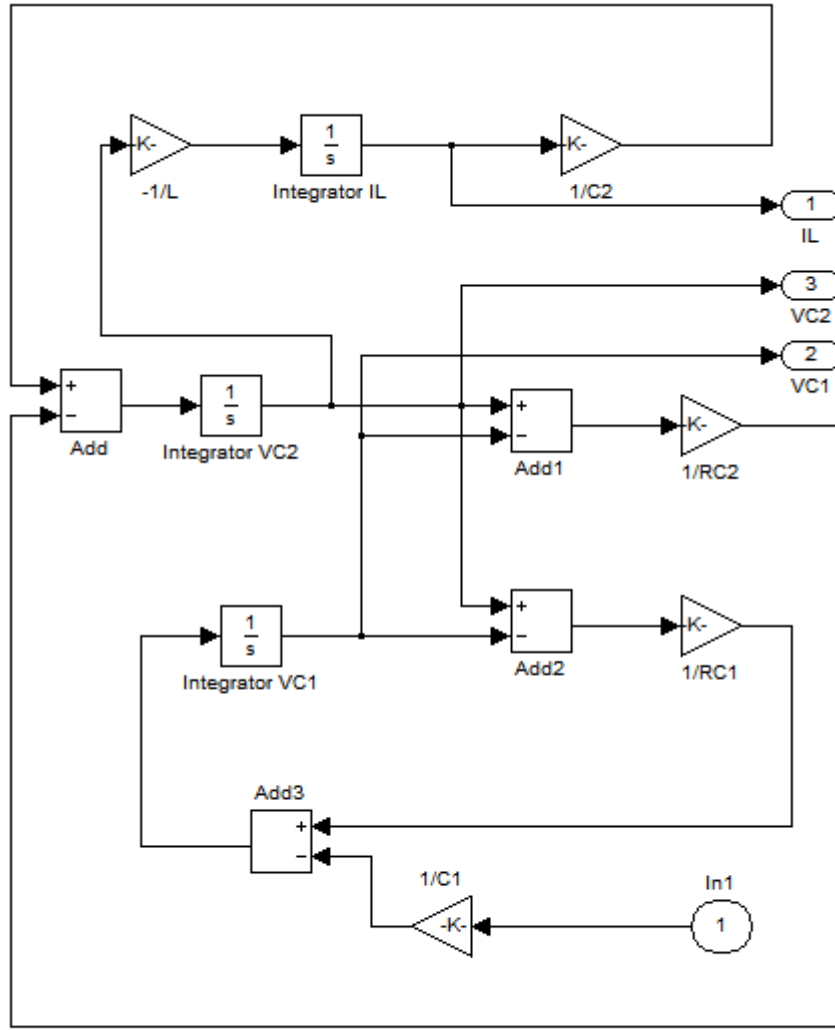
Devre Parametreleri	Çeker Sayısı					
	5	6	7	8	9	10
R(Ω)	1515	1572	1515	1552	1381	1381
b ₁ (V)	-	0.1	-	0.1	-	0.1
b ₂ (V)	0.8	1.1	0.8	1.1	0.8	1.1
b ₃ (V)	1.4	1.55	1.4	1.55	1.4	1.55
b ₄ (V)	3.2	3.2	3.2	3.2	3.2	3.2
b ₅ (V)	3.9	3.85	3.9	3.85	3.9	3.85
b ₆ (V)	-	-	5.8	5.84	5.8	5.84
b ₇ (V)	-	-	6.4	6.6	6.4	6.6
b ₈ (V)	-	-	-	-	8.3	8.7
b ₉ (V)	-	-	-	-	9.2	9.45

2.1.Chua Devresinin Simulink Modeli

Simulink, MATLAB ortamında da dahil olmak üzere dinamik sistemleri modellemek, simule etmek ve analiz etmek için bir yazılım paketidir. Model analizi araçları, MATLAB komut satırından erişilebilen doğrusallaştırma ve düzeltme araçlarını ve ayrıca MATLAB ve uygulama araç kutularındaki birçok aracı içerir. MATLAB ve Simulink'in entegre olması nedeniyle her iki ortamda modellerimizi herhangi bir noktada simule edebilir, analiz edebilir ve revize edilebilir[31].

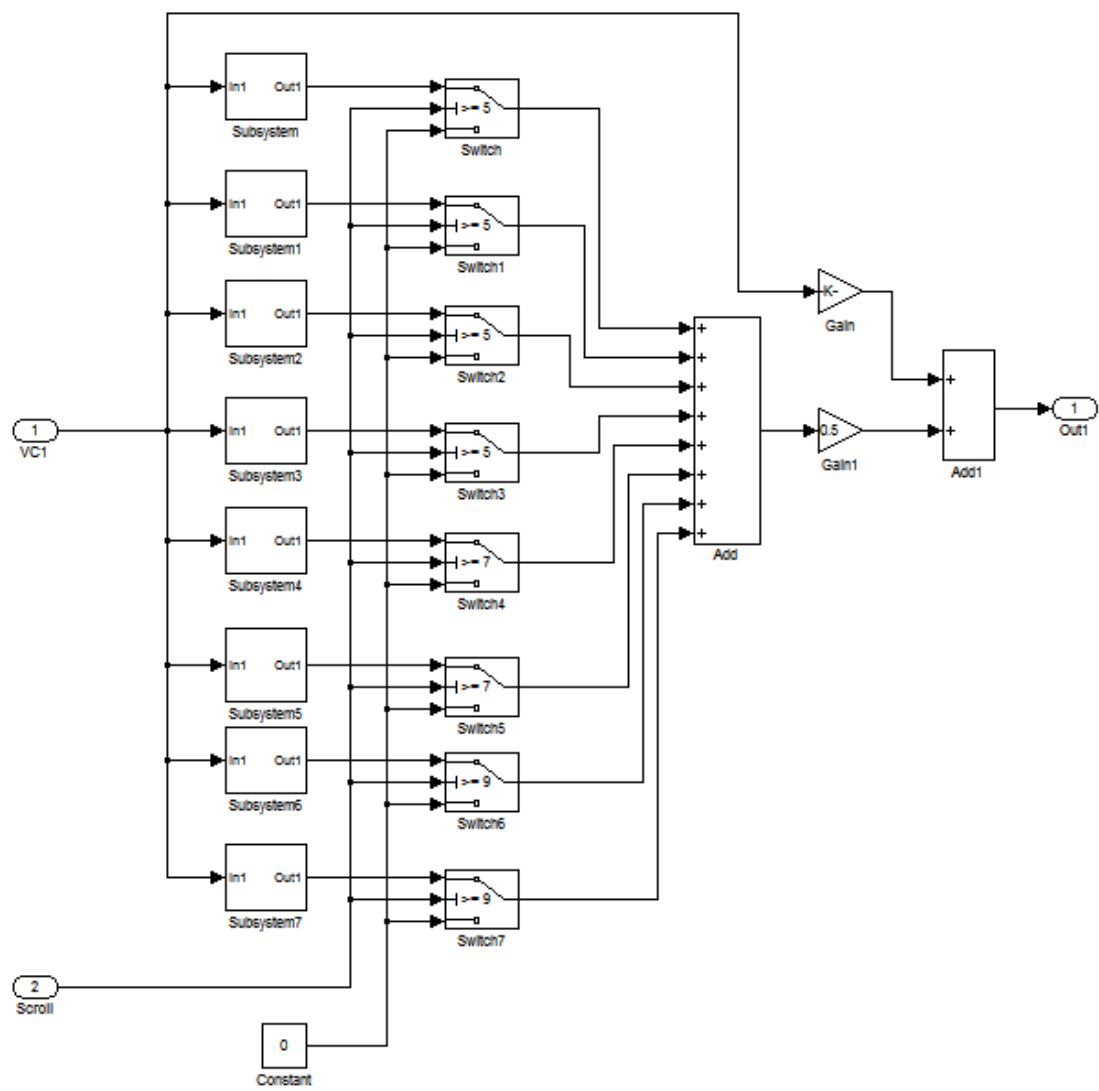
Chua devresinin gösterdiği davranışını görebilmek için Chua devresine ait diferansiyel denklemler MATLAB/Simulink ortamında uygun bloklar kullanılarak modellenirse, bu sisteme ait simulink modeli Şekil 2.3'deki gibi olur.

Bu modelin Simulink ile modellenmesinde en önemli sorun, döngüsel bir yapıya sahip ve karşılaştırma yapan komutları olan PPD ögesi gibi görünmektedir. Tüm sistemi tek bir yapıda göstermek yerine, kolayca anlaşılabilmesi ve kontrol edilebilmesi için subsystem diye adlandırılan bloklara bölünerek modellenmiştir. Model bir bütün olarak Şekil 2.3'de gösterilmektedir. Model, Denklem (2.1) 'de verilen diferansiyel denklemlerin ele alındığı bir tank rezonator alt bloğu ve Denklem (2.2)' de verilen PPD yapılarının olduğu tek ve çift çeker sayısının ayrı ayrı değerlendirildiği PWL alt sistem bloklarından oluşur.

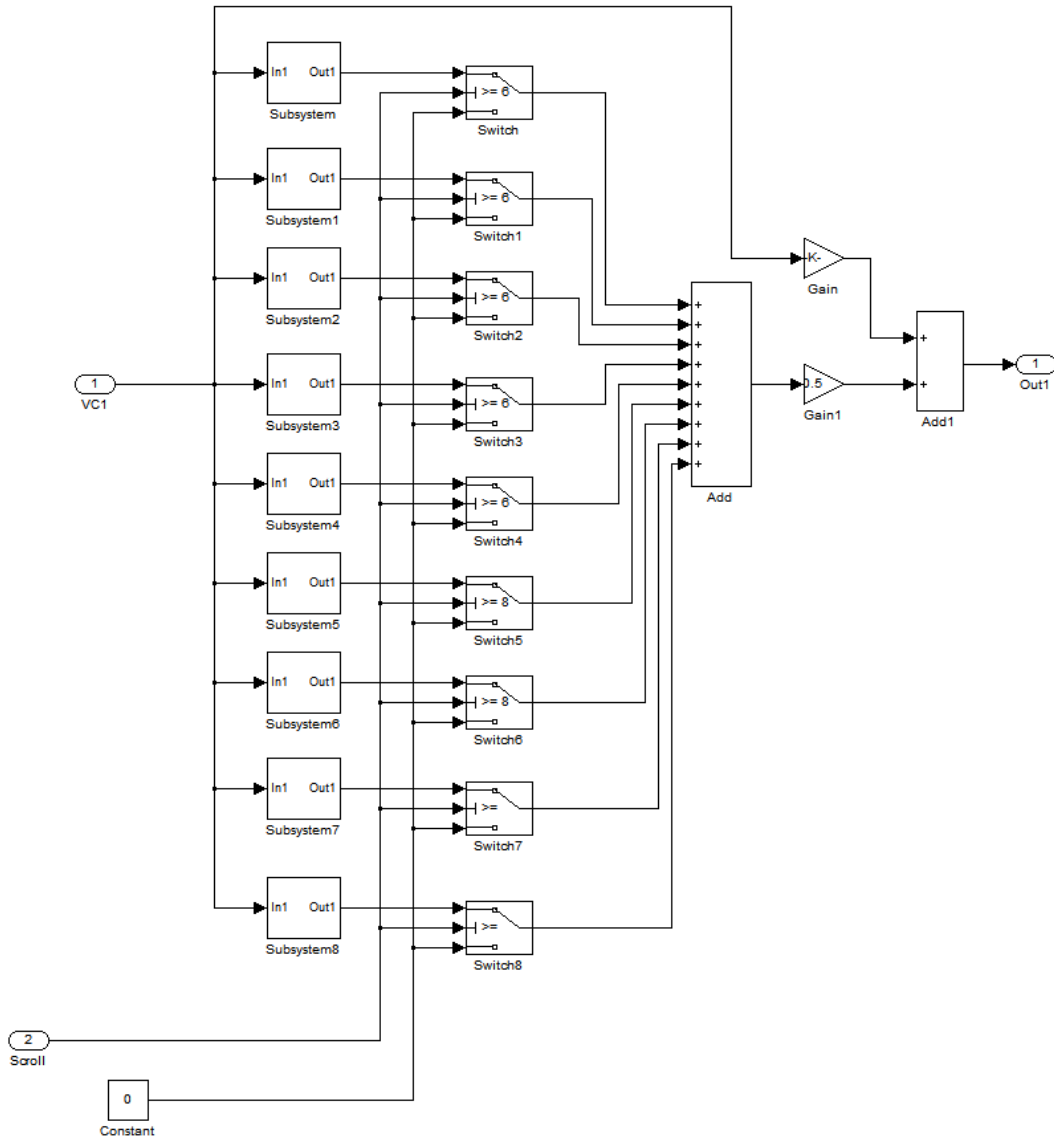


Şekil 2.4. Diferansiyel denklemlerdeki matematiksel işlemler için Tank Resonator modeli

Diferansiyel denklemlerdeki matematiksel işlemler için Tank Resonator modeli Matlab blokları kullanılarak Simulink ortamında uygulanması ile elde edilmiş olup $C_1=10$ nF, $C_2=100$ nF, $L=18.68$ mH parametre değerleri kullanılmıştır.



a)



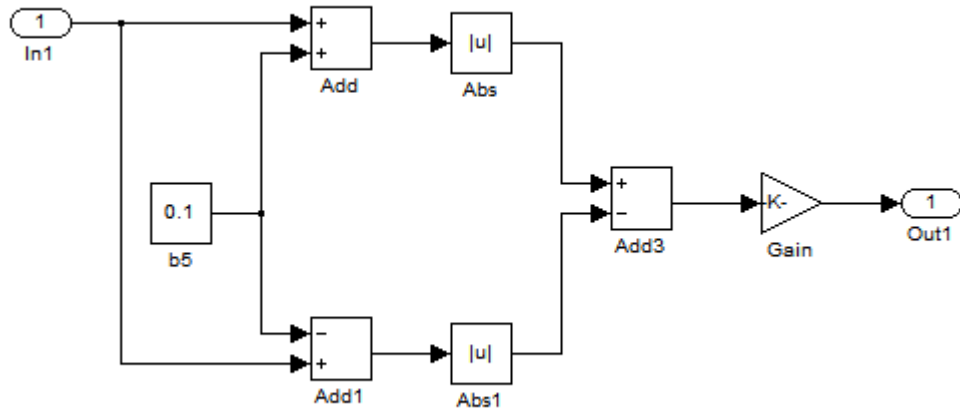
b)

Şekil 2.5. PPD Elemanın Simulink Modeli a) PWL Tek b) PWL Çift

Şekil 2.5'te gösterilen subsystem blokları alt bloklar olup çok fazla bloğun içiçe geçip kontrol edilme zorluğunu önlemek için kullanılmıştır. Subsystem blokları basit bir matematiksel işlemden oluşmaktadır. Sisteme dış girdi olarak kullanılan scroll değeri Şekil 2.5'deki switch diye adlandırılan anahtarlar ile karşılaştırılarak sonucu çıkışa aktarır. Bu karşılaştırma için değerler sırasıyla tek ppd elemanı için 5,5,5,5,7,7,9,9 olup 8 adet anahtar kullanılmıştır. Çift ppd elemanı için 6,6,6,6,6,8,8,10,10 olup 9 adet anahtar kullanılmıştır. Çeker sayısı tek ise ve anahtarlara tanımlı bu değerlerden büyük ise bağlı olunan subsystem bloğunu switchlerin çıkışındaki toplamın içine atmaktadır. Aynı şekilde çeker sayısı çift ise ve anahtarlara tanımlı bu değerlerden büyük ise bağlı olunan subsystem

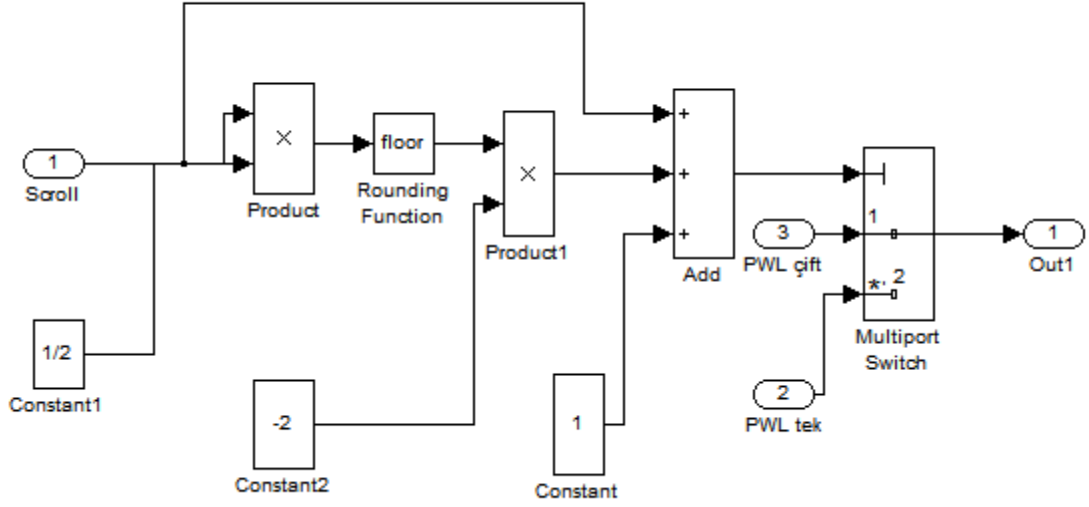
bloğunu switchlerin çıkışındaki toplamın içine atmaktadır. Eğer küçük ise diğer 0 değerini toplam bloğuna göndermektedir.

Denklem (2.2) 'deki $|V_1 + b_i| - |V_1 - b_i|$ matematiksel işlemlerini gerçekleştirmek için Matlab kütüphanesinden yararlanılarak Şekil 2.6'daki model oluşturulmuştur. Tablo 2.1'de verilen kırılma noktası gerilimleri ile istenen karakteristik eğimi sağlayacak şekilde tasarlanmıştır.



Şekil 2.6. PPD elemanların alt yapı bloğu

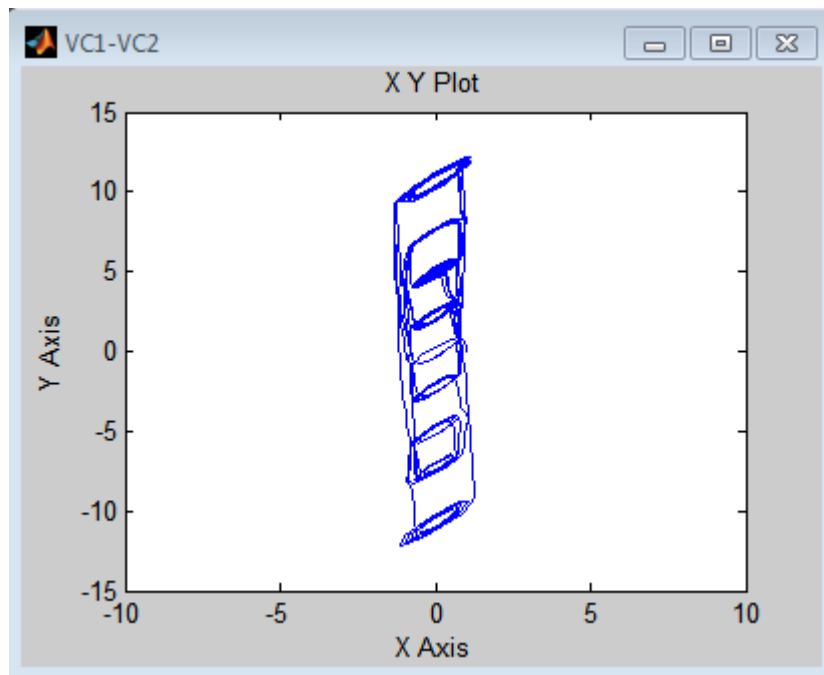
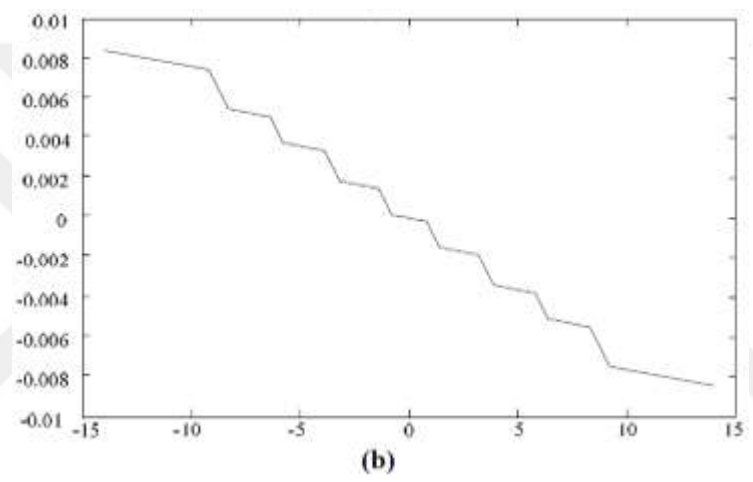
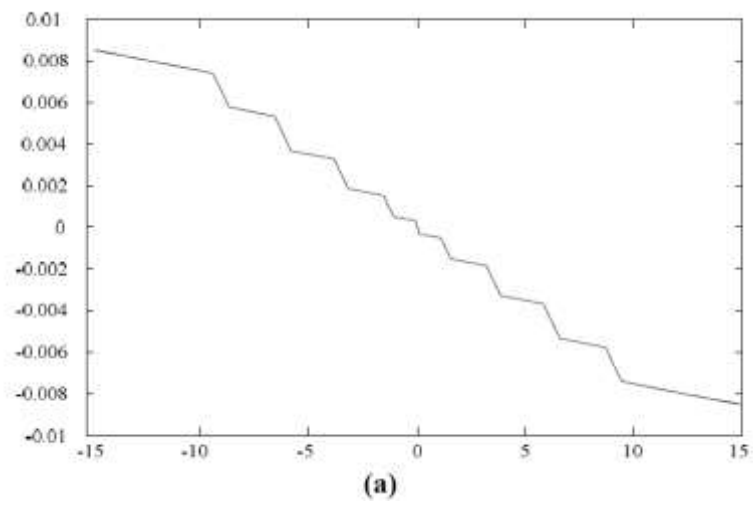
Çeker sayısını Matlab satırına dışarıdan girerek PWL alt sistem blokları tek ve çift çeker değerlerine bağlı olarak oluşturdukları sonucu tek bir modelde elde etmek için Şekil 2.7'de gösterildiği gibi bir kontrol bloğu tasarlanmıştır. Bu kontrol bloğu, Matlab ortamında HDL Kodlayıcı ile düzenlenebilecek matematiksel terimler kullanılarak özel olarak tasarlanmıştır. Bu tezde daha sonra gerçekleştirildi. Tek sayıda çeker isteniyorsa, çok fonksiyonlu anahtar, çeker sayısına bağlı olarak tek girişi veya çift girişi çıkışa iletir. Aslında, burada kullanılan çoklu giriş anahtarının girişindeki ifade basit bir modül alma işlemidir.

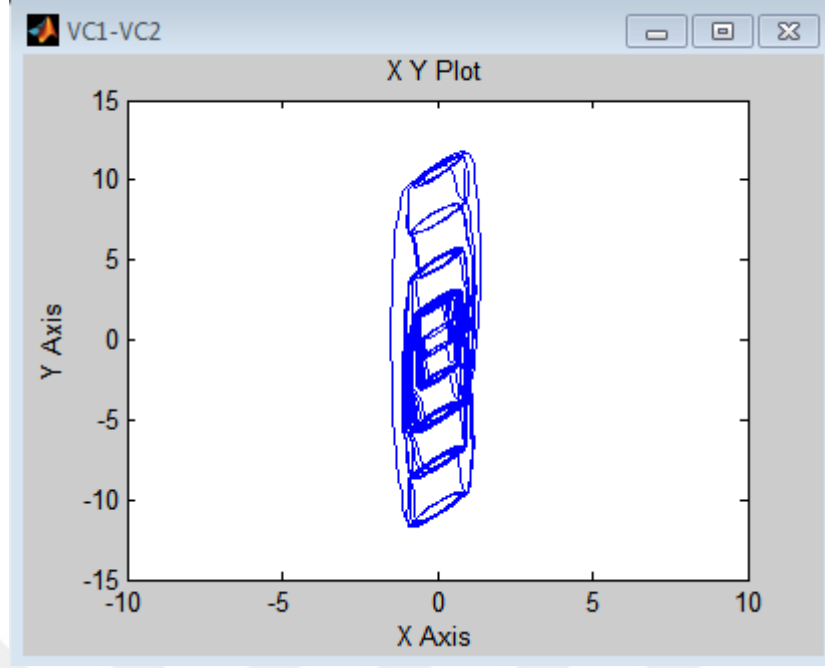


Şekil 2.7. Kontrol Blok Yapısı

2.2. Benzetim Sonuçları

PPD elemanı, ± 15 V, 10 Hz sinüzoidal bir kaynağa bağlı olan tank rezonatoründen ayrılır ve giriş-çıkış karakteristiği, Şekil 2.8a'da gösterildiği gibi elde edilir. Şekil 2.8a'da görüldüğü gibi 19 doğrusal bölgeden meydana gelmiştir. Şekil 2.8 b'de ise 17 doğrusal bölgeden oluşan doğrusal olmayan karakteristik 9 çeker için elde edilmiştir. Chua sisteminin Matlab ortamında hazırlanan blok diyagramı için başlangıç değerleri sırasıyla, (1, 0, 0) olarak seçilmiştir ve 0.02 saniyelik bir benzetim yapılmıştır. Bu benzetim sonucunda gösterilen Şekil 2.8 c ve 2.8 d'de elde edilmiş olan 10 çeker ve 9 çeker kaotik davranışları gözlenmiştir. Bu sonuçların elde edilmesinde R için Tablo 2.1 deki değerler çeker sayısına göre seçilmiş olup, $L=18$ mH, $C_1=10$ nF, $C_2=100$ nF değerleri alınmıştır[3].





Şekil 2.8. (a) 10 çeker (b) 9 çekerin kaotik davranışı için PPD elemanın gösterdiği karakteristik, Faz uzay diyagramları (c) 10-çeker, (d) 9-çeker

3. DİFERANSİYEL DENKLEMLERİN AYRIKLAŞTIRMA YÖNTEMLERİ

Kaotik davranışın analog devre temelli modellenmesi ilk olarak 1983 yılında elektronik elemanlar ve devreler kullanılarak Leon O. Chua tarafından gerçekleştirildi. Tasarımcısının adını taşıyan bu sürekli özerk Chua devresi, literatürde birçok kez çalışılmıştır, çünkü elektronik olarak anlaşılması ve tasarlanması ve günümüzün kaotik devre çalışmalarına ışık tutması çok basit bir devredir. Literatürde birçok özerk kaotik devre geliştirilse de, en çok çalışılan özerk sistemler Lorenz, Rössler, Chua, Liu, Lü, Duffing, Chen, Rabinovich, Rikitake ve Burke-Shaw'dur. Kaotik sistemler, ayrık zamanlı olarak kaotik sistemlere ve sürekli zamanda kaotik sistemlere ayrılır. Ayrık zamandaki kaos, genellikle uygun bir $f(x)$ doğrusal olmayan fonksiyonun bir yinelemesinin geri besleme karakteristiğine sahip kaotik sekanslar sistemidir. Ayrık zamanlı kaotik fonksiyonlar incelendiğinde, bu davranışa basit bir boyutlu ifade ile ulaşılabilir. $x_{n+1} = f(x_n)$ eşitliğinde gösterilen x_{n+1} ve $f(x_n)$ değişkenlerin kullanımıyla tek boyutlu haritalar oluşturulur ve bu fonksiyonlar ters V biçimli düzlem formunda bir geri besleme yapısına sahiptir. Çadır, lojistik, çadır, Bernoulli sürgüsü gibi işlevler bu davranışla kaotik haritalara örnektir [14].

Seçilen fonksiyonun $f(x)$ kaotik olması için, fonksiyon ilk koşullara yüksek hassasiyet göstermeli ve buna göre x_n değerleri çok farklı değerlere gitmelidir. Ayrık zamanlı kaotik fonksiyonlar incelendiğinde, bu davranış basit bir boyutlu denklem ile sağlanabilir. Bu x_{n+1} , $f(x_n)$ değişkenlerinin kullanımıyla bir boyutlu haritalar oluşturulmakta ve bu fonksiyonlar geri besleme yapısına sahip olduğundan düzlemde V şeklinde bir form almaktadır [14].

Bu çok kaymalı kaotik osilatörleri gerçekleştirmenin avantajı, çarpan varlıkları kullanmaktan kaçınmak, böylece tek sabit çarpma (SCM) blokları gerçekleştirerek gösterdiğimiz gibi FPGA kaynaklarını optimize etmek ve işlem hızını arttırmaktır [15].

Lineer olmayan bir sistemin diferansiyel denklemleri kodlama ve tasarımdan önce FPGA editöründe ayrılmalıdır. Chua kaotik sistemi, sürekli üretilen sinyalleri ayrıştırmak için çoğunlukla tercih edilen Taylor serisi uzantısı, Runge Kutta ve Euler ayrıklaştırma yöntemleri kullanılarak ayrıştırılabilir. Uygulama alanına göre en uygun sayısal yöntem, yöntemlerin doğruluğunu ve karmaşıklığını kanıtlayarak ve tasarımları FPGA kartındaki kaynak kullanımı açısından karşılaştırarak kullanılabilir. Bu çalışmada diferansiyel denklemler Euler ayrıklaştırma metodu kullanılarak ayrıştırılmıştır.

Denklem (2.1) 'de verilen diferansiyel denklemin matematiksel modeli, üç boyutlu basit bir otonom sistemdir. Bu nedenle, FPGA'yı devreye uygulamak için mevcut dijital çözünürlük yöntemlerinden biri kullanılmalıdır. Çünkü çoğu adi diferansiyel denklem analitik olarak çözülebilir.

Uygulanacak olan metotlarda Denklem (2.1) deki I_L , V_1 ve V_2 sırasıyla x, y, z olarak adlandırılmış olup ilgili işlemler bu harflendirilmeler ile yapılmıştır.

3.1. Taylor Serisi Genişleme Metodu

Taylor serisi çoğu fonksiyonu güç serisi olarak ifade etmenin bir yoludur. $x=a$ etrafındaki Taylor genişlemesinde, büyüklük $(x-a)$ üstlerinin katsayıları, $x=a$ 'daki fonksiyonun türevlerinin değerlerini içerir. Bunun anlamı, eğer bir fonksiyonun ve $x=a$ noktasındaki türevlerinin değerleri biliniyorsa, x noktasındaki tüm fonksiyonların değerleriyle aynı değerleri vermek için bir dizi kuvvet yazılabilir demektir. $x=a$ yerine x_0 kullanılacaktır.

Bir fonksiyonun ilk türevi $y(x)$, $\dot{y}=f(x,y)$ olarak verilir ve fonksiyonun ilk değeri $y(x_0)$ olarak verilir. Bu bilgiyi kullanarak, $y(x)$ 'in $x=x_0$ civarında genişlemesi yazılabilir. Bu genişleme için gereken $\dot{y}=f(x,y)$ büyüklüğünde gereken türev derecesi alınabilir ve $x=x_0$ değerleri hesaplanabilir. Bununla birlikte, eğer y bilinmeyen (x) türevi x 'in yanında bilinmeyen boyutu y içeriyorsa türev değerini hesaplamak kolay değildir. Bu gibi durumlarda, yüksek mertebeden türevler x cinsinden ve daha küçük mertebeden türevlerde ifade edilir.

Algoritmanın hassasiyetini belirleyen “ Δh ” adım sayısı parametresini belirtmektedir. Bu sinyal, tasarımı daha esnek hale getirmek için harici olarak uygulanır. Bu çalışmada adım aralığı dediğimiz h değeri 10^{-6} olarak alınmıştır. İlk işletim sırasındaki sistemin ilk gereksinimleri, tasarımda kullanılan FPGA yongasının kaynaklarını azaltmak için tasarıma gömüldü. Bununla birlikte, gerektiğinde, bu sinyaller, kullanıcının 3 farklı 32-bit sinyali [21] tanımlayarak tasarımdaki küçük değişikliklerle değerlerini ayarlayabilmesi için tasarlanabilir.

$$y(x) = y(x_0) + \dot{y}(x_0)(x - x_0) + \frac{\ddot{y}(x_0)}{2!}(x - x_0)^2 + \frac{\ddot{\ddot{y}}(x_0)}{3!}(x - x_0)^3 + \dots \quad (3.1)$$

Şayet $x-x_0=h$ denilirse

$$y(x) = y(x_0) + \dot{y}(x_0)h + \frac{\ddot{y}(x_0)}{2!}h^2 + \frac{\ddot{y}(x_0)}{3!}h^3 + \dots \quad (3.2)$$

Herhangi bir sistemi bir mikroişlemci üzerinde programlamak için, sistem modeli ayrık hale getirilmelidir. Taylor serisi genişleme sayısal yönteminde, Chua devresinin durum değişkenleri Denklem (3.3) 'teki gibi herhangi bir m terimi için genişletilir.

$$\begin{aligned} x(t+h) &= x(t) + \sum_{m=1}^{\infty} \frac{1}{m!} \cdot h^m \cdot x^{(m)}(t) \\ y(t+h) &= y(t) + \sum_{m=1}^{\infty} \frac{1}{m!} \cdot h^m \cdot y^{(m)}(t) \\ z(t+h) &= z(t) + \sum_{m=1}^{\infty} \frac{1}{m!} \cdot h^m \cdot z^{(m)}(t) \end{aligned} \quad (3.3)$$

Bu çalışmanın sayısal yöntemlerin karşılaştırılmasını amaçladığı için, Taylor serisi genişleme yöntemi $m=2$ ve $h=10^{-6}$ için yürütülür. Bu durumda, Chua devresinin Taylor serisi genişleme yöntemi ile ayrık zaman hal denklemleri Denklem (3.4) ve (3.5)' teki gibi elde edilir:

$$\begin{aligned} x[n+1] &= x[n] + 10^{-6} \cdot \dot{x}[n] + \frac{1}{2} \cdot (10^{-6})^2 \cdot \ddot{x}[n] \\ y[n+1] &= y[n] + 10^{-6} \cdot \dot{y}[n] + \frac{1}{2} \cdot (10^{-6})^2 \cdot \ddot{y}[n] \\ z[n+1] &= z[n] + 10^{-6} \cdot \dot{z}[n] + \frac{1}{2} \cdot (10^{-6})^2 \cdot \ddot{z}[n] \end{aligned} \quad (3.4)$$

$$\dot{x}[n] = -\frac{1}{L} \cdot y[n]$$

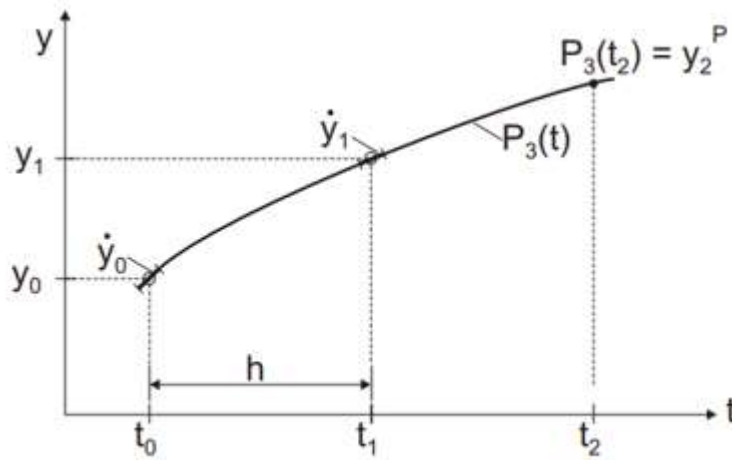
$$\ddot{x}[n] = -\frac{1}{L} \cdot \left(\frac{1}{C_2} \cdot x[n] - \frac{1}{RC_2} \cdot (y[n] - z[n]) \right)$$

$$\dot{y}[n] = \frac{1}{C_2} x[n] - \frac{1}{RC_2} \cdot (y[n] - z[n])$$

$$\begin{aligned}
\ddot{y}[n] &= \frac{1}{C_2} \left(-\frac{1}{L} \cdot y[n] \right) - \frac{1}{RC_2} \left(\left(\frac{1}{C_2} x[n] - \frac{1}{RC_2} \cdot (y[n] - z[n]) \right) \right. \\
&\quad \left. - \left(\frac{1}{RC_1} \cdot (y[n] - z[n]) - \frac{1}{C_1} f(z[n]) \right) \right) \\
\dot{z}[n] &= \frac{1}{RC_1} \cdot (y[n] - z[n]) - \frac{1}{C_1} f(z[n]) \\
\ddot{z}[n] &= \frac{1}{RC_1} \left(\left(\frac{1}{C_2} x[n] - \frac{1}{RC_2} \cdot (y[n] - z[n]) \right) \right) - \left(\frac{1}{RC_1} \cdot (y[n] - z[n]) - \frac{1}{C_1} f(z[n]) \right) - \\
&\quad \frac{1}{C_1} \left(f \left(\frac{1}{RC_1} \cdot (y[n] - z[n]) - \frac{1}{C_1} f(z[n]) \right) \right)
\end{aligned} \tag{3.5}$$

3.2. Runge-Kutta Ayırıklaştırma Yöntemi

Runge-Kutta metodu, Taylor serileri ile yaklaşımdaki hassasiyeti, yüksek mertebeden türevlere ihtiyaç duymadan yakalayabildiğinden, yüksek hassasiyetin arandığı durumlarda tercih edilir. Runge-Kutta metodu Denklem (3.6)'daki formda yazılabilir.



Şekil 3.1. Runge- Kutta yöntemi

$$y_2 = y_0 + \frac{2h}{6} \cdot [y_0 + 4y_1 + f(u_2, y_2^P)]$$

$$y_{i+1} = y_i + \phi(x_i, y_i, h)h \tag{3.6}$$

Burada $\emptyset(x_i, y_i, h)$ fonksiyonuna artım fonksiyonu denir. Bu söz konusu aralıktaki eğimi gösterir. Artım fonksiyonu genel formda Denklem (3.7)'deki gibi yazılabilir.

$$\emptyset = a_1, k_1 + a_2, k_2 + \dots a_n, k_n \quad (3.7)$$

Burada a'lar sabit k'ler ise Denklem (3.8) deki gibidir.

$$k_1 = f(x_i, y_i)$$

$$k_2 = f(x_i + p_1 h, y_i + q_{11} k_1 h)$$

$$k_3 = f(x_i + p_2 h, y_i + q_{21} k_1 h + q_{22} k_2 h)$$

.

.

.

$$k_n = f(x_i + p_{n-1} h, y_i + q_{n-1} k_1 h + q_{n-1,2} k_2 h + \dots + q_{n-1} k_{n-1} h) \quad (3.8)$$

Burada p ve q sabitlerdir.

3.2.1. İkinci Dereceden Runge-Kutta Yöntemi

Literatürde, en çok tercih edilen Runge-Kutta metodu 4. dereceden Runge-Kutta yöntemidir. Bu çalışma, ayırıklaştırma yöntemlerini karşılaştırmayı amaçladığından, 2. derece Runge-Kutta yöntemi tercih edilmektedir. 2. dereceden Runge-Kutta yöntemi denklem (3.10) da verilen bir örnek için genişletilmiştir. Chua devresinin durum değişkenleri, $h = 10^{-6}$ için 2. dereceden Runge-Kutta yöntemi ile sistemin ayrık zamanı modeli ise Denklem (3.19) 'daki gibi temsil edilmektedir.

$$y_{i+1} = y_i + (a_1, k_1 + a_2, k_2)h$$

$$k_1 = f(x_i, y_i)$$

$$k_2 = f(x_i + p_1 h, y_i + q_{11} k_1 h) \quad (3.9)$$

y_{i+1} için y_i ve $f(x_i, y_i)$ terimleri ile ikinci mertebeden Taylor serisi yazılırsa

$$y_{i+1} = y_i + f(x_i, y_i)h + \frac{\dot{f}(x_i, y_i)}{2!} h^2 \quad (3.10)$$

Burada $\dot{f}(x_i, y_i)$ zincir kuralı ile belirlenmelidir.

$$\dot{f}(x_i, y_i) = \frac{\partial f(x, y)}{\partial x} + \frac{\partial f(x, y)}{\partial y} \frac{dy}{dx} \quad (3.11)$$

Bu ikinci türev Taylor formülünde yerine yazılırsa

$$y_{i+1} = y_i + f(x_i, y_i)h + \left(\frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} \frac{dy}{dx} \right) \frac{h^2}{2!} \quad (3.12)$$

İki değişkenli fonksiyonlarda Taylor serisi

$$g(x + r, y + s) = g(x, y) + r \frac{\partial g}{\partial x} + s \frac{\partial g}{\partial y} + \dots \quad (3.13)$$

Bu formül yukarıdaki iki değişkenli fonksiyon içeren k_2 eşitliği için uygulanırsa

$$k_2 = f(x_i + p_1 h, y_i + q_{11} k_1 h) = f(x_i, y_i) + p_1 h \frac{\partial f}{\partial x} + q_{11} k_1 h \frac{\partial f}{\partial y} + 0(h^2) \quad (3.14)$$

Bu k_2 eşitliği $k_1 = f(x_i, y_i)$ eşitliğiyle birlikte Denklem (3.9) daki y_{i+1} de yerinde yazılırsa

$$y_{i+1} = y_i + a_1 h f(x_i, y_i) + a_2 h f(x_i, y_i) + a_2 p_1 h^2 \frac{\partial f}{\partial x} + a_2 q_{11} f(x_i, y_i) h^2 \frac{\partial f}{\partial y} + 0(h^2) \quad (3.15)$$

ve terimler bir araya toplanırsa

$$y_{i+1} = y_i + [a_1 f(x_i, y_i) + a_2 f(x_i, y_i)]h + \left[a_2 p_1 \frac{\partial f}{\partial x} + a_2 q_{11} f(x_i, y_i) \frac{\partial f}{\partial y} \right] h^2 + 0(h^3) \quad (3.16)$$

Bu denklem $\frac{dy}{dx} = f(x, y)$ olduğu göz önüne alınarak değerlendirilirse

$$\begin{aligned}
a_1 + a_2 &= 1 \\
a_2 p_1 &= \frac{1}{2} \\
a_2 q_{11} &= \frac{1}{2}
\end{aligned} \tag{3.17}$$

bulunur. Burada üç denklem dört bilinmeyen olduğundan çok sayıda çözüm elde edilebilir.

$$\begin{aligned}
\dot{x}(t) &= \frac{dx(t)}{dt} = f(x(t), t) \\
k_1 &= h \cdot f(x(t_0), t_0) \\
x_1 \left(t_0 + \frac{h}{2} \right) &= x(t_0) + k_1 \cdot \frac{h}{2} \\
k_2 &= f \left(x_1 \left(t_0 + \frac{h}{2} \right), t_0 + \frac{h}{2} \right) \\
x(t_0 + h) &= x(t_0) + k_2 \cdot h
\end{aligned} \tag{3.18}$$

$$k_{1x} = h \cdot \left(-\frac{1}{L} \cdot y[n] \right)$$

$$k_{2x} = h \cdot \left(-\frac{1}{L} \cdot y[n] - \frac{k_{1x}}{2} \right)$$

$$x[n+1] = x[n] + \frac{(k_{1x} + k_{2x})}{2}$$

$$k_{1y} = h \cdot \left(\frac{1}{C_2} x[n] - \frac{1}{RC_2} \cdot (y[n] - z[n]) \right)$$

$$k_{2y} = h \cdot \left(\frac{1}{C_2} x[n] + \frac{1}{RC_2} \cdot \left(y[n] + \frac{k_{1y}}{2} \right) - \frac{1}{RC_2} \cdot z[n] \right)$$

$$y[n+1] = y[n] + \frac{(k_{1y} + k_{2y})}{2}$$

$$k_{1z} = h \cdot \left(\frac{1}{RC_1} \cdot (y[n] - z[n]) - \frac{1}{C_1} f(z[n]) \right)$$

$$k_{2z} = h \cdot \left(\frac{1}{RC_1} \cdot (y[n]) - \frac{1}{RC_1} \cdot \left(z[n] + \frac{k_{1z}}{2} \right) - \frac{1}{C_1} f \left(z[n] + \frac{k_{1z}}{2} \right) \right)$$

$$z[n + 1] = z[n] + \frac{k_{1z} + k_{2z}}{2} \quad (3.19)$$

3.3. Euler Ayırıklaştırma Yöntemi

Euler yöntemi İleri ve Geri Euler olmak üzere iki şekilde düzenlenebilir. Bu çalışmada Euler yöntemi tercih edilmiştir. Euler yönteminin ifadesi Denklem (3.22) ve ayırık model, $h = 10^{-6}$ için Denklem (3.23) ve (3.24)'te ifade edilmiştir.

Başlangıç değer problemleri için kullanılan en basit metottur. Taylor serisinin ilk iki terimi (yani birinci türeve kadar olan terimleri dikkate alındığında) Euler metodu elde edilir.

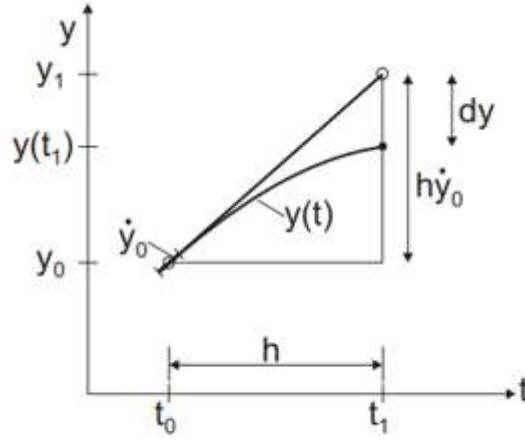
$$y_{n+1} = y(x_n) + \dot{y}(x_n) \cdot (x_{n+1} - x_n) \rightarrow y(x_n + \Delta x) = y_n + \left(\frac{dy}{dx} \Big|_{x=x_n} \right) \cdot \Delta x \quad (3.20)$$

Denklemde $\Delta x = h$ artış değeri olarak gösterildiğinde,

$$y(x_n + h) = y_n + \left(\frac{dy}{dx} \Big|_{x=x_n} \right) h \quad (3.21)$$

haline gelir. Yani başlangıç değeri bilinen birinci dereceden adi diferansiyel denklemin sayısal çözümü böylece hesaplanır.

$$\begin{aligned} \dot{x}(t) &= \frac{x(t+h) - x(t)}{h} \\ \dot{y}(t) &= \frac{y(t+h) - y(t)}{h} \\ \dot{z}(t) &= \frac{z(t+h) - z(t)}{h} \end{aligned} \quad (3.22)$$



Şekil 3.2. Euler Yöntemi

$$y_1 = y_0 + h \cdot \dot{y}_0$$

$$x[n + 1] = x[n] + h \cdot -\frac{1}{L} \cdot y[n]$$

$$y[n + 1] = y[n] + h \cdot \left(\frac{1}{C_2} x[n] - \frac{1}{RC_2} \cdot (y[n] - z[n]) \right)$$

$$z[n + 1] = z[n] + h \cdot \left(\frac{1}{RC_1} \cdot (y[n] - z[n]) - \frac{1}{C_1} f(z[n]) \right) \quad (3.23)$$

$$f(z[n]) = m_{2n-1}y[n] + \frac{1}{2} \sum_{i=1}^{2n-1} (m_{i-1} - m_i)(|y[n] + b_i| - |y[n] - b_i|)$$

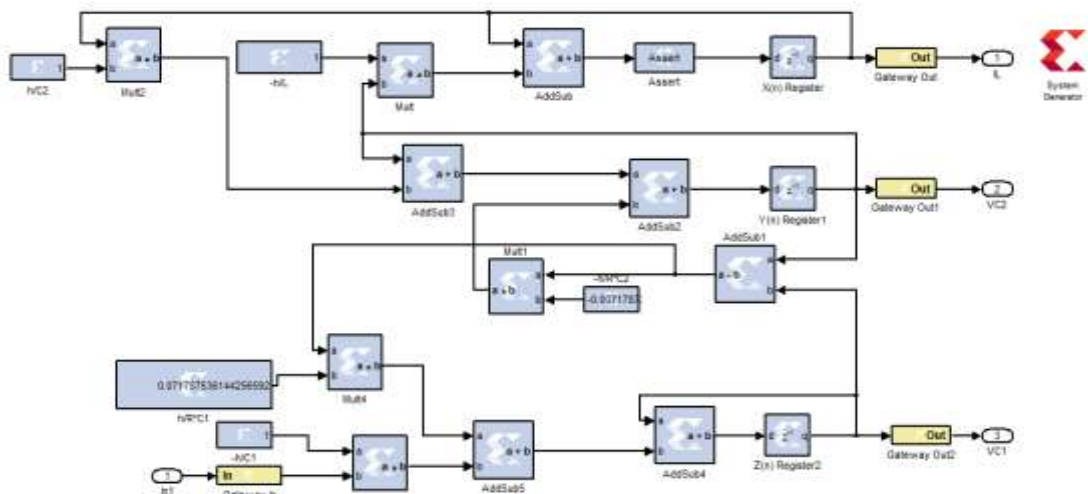
$$f(z[n]) = m_{2n-1}y[n] + \frac{1}{2} \sum_{i=2}^{2n-1} (m_{i-1} - m_i)(|y[n] + b_i| - |y[n] - b_i|) \quad (3.24)$$

Taylor Serisi ve Runge-Kutta denklemleri ile Euler denklemini karşılaştırdığımızda Euler denklemini daha kısa olup hata yapma ihtimali daha düşüktür. Bu da hem zamandan tasarruf hem de denklemin kontrolü açısından bizim için bir avantajdır.

Bu denklemler FPGA ortamında yapılacak işlemlerde integralden kurtulmasına yarayacaktır. Bölüm 3'te elde edilen sonuçlarda da görüldüğü gibi bu ayrıklaştırma metotlarında elde edilen denklemlerle sadece çarpma, toplama, çıkarma parametreleri

kullanılmıştır. Performans ve donanım kaynakları arasında iyi bir denge sağladığı göstermektedir. Bu da dijital ortamda yani FPGA ortamında da sistemin çalışmasının görülmesine olanak sağlayacaktır.

Bu çalışmada Denklem (2.1) üzerine Euler Yöntemi uygulanarak denklem ayrıklaştırma yoluna gidilmiştir. Elde edilen yeni denklemler Denklem (3.23)'te belirtilmiş olup Xilinx System Generator üzerinde elde edilen model Şekil 3.3'te gösterilmiştir.



Şekil 3.3. Euler Yönteminin XSG 'de uygulanması

4. XILINX SYSTEM GENERATOR TEKNOLOJİSİ

Sistem Üreticisi, ISE® Design Suite'in bir parçasıdır ve uygulamaya özel tasarım uygulamaları için ekleyiciler, çarpanlar, kayıtlar, filtreler ve hafızalar gibi Xilinx DSP(Dijital sinyal işleme) Blocksetinde sağlar. Bu bloklar, seçilen cihaz için optimize edilmiş sonuçlar sağlamak üzere Xilinx IP çekirdek jeneratörlerinden yararlanır. Sistem Üreticisi kullanılırken Xilinx FPGA'larda önceki deneyim gerekli değildir. Tasarımlar, Xilinx'e özgü bir Blockset kullanılarak DSP dostu Simulink modelleme ortamında yakalanır. Bir FPGA programlama dosyası oluşturmak için sentez, yer ve rota dahil olmak üzere tüm FPGA uygulama adımları otomatik olarak gerçekleştirilir. Donanım uygulaması için Xilinx sistem üreticisi kullanmanın avantajı, Xilinx Blockset'in, FPGA modülünün MATLAB Simulink Blocks tarafından sağlanan piksel vektörüyle birlikte simüle edilmesine yardımcı olan MATLAB Simulink ile yakın entegrasyon sağlamasıdır[16].

Xilinx, donanım programcılarının tasarım görevlerini basitleştirmek için birçok yazılım aracı üretmektedir. Böyle bir program DSP Sistem Üreticisi'dir. Matlab Simulink ile arayüz oluşturur ve dijital olarak gerçekleştirilebilen Simulink bloklarının ve yaygın olarak kullanılan dijital programlama bloklarının verimli uygulamalarını sağlar. Sistem Üreticisinin bir diğer önemli özelliği Gateway Giriş / Çıkış bloklarıdır. Bunlar, FPGA'nın kayan nokta mimarisiyle Simulink'in çift hassasiyeti arasındaki arayüzü sağlar [31].

XSG, çeşitli Xilinx FPGA'lara uygulanabilecek birkaç donanım işlemine yönelik bir dizi Simulink blokları sağlamaktadır. Bu bloklar, Simulink ortamını kullanarak donanım sisteminin fonksiyonelliğini ve benzetimini yapmak üzere kullanılmaktadır [19].

Xilinx System Generator, FPGA boardlarında hızlı ve kolay tasarımlar oluşturmak, Hardware Co Simulation tasarımını gerçekleştirmek ve gerçek zamanlı tümleşik uygulamaları uygulamak için kullanılan yüksek seviyeli bir MATLAB-Simulink tabanlı yazılım platformudur. XSG, aritmetik, mantıksal, bellek ve DSP uygulamaları için MATLAB-Simulink içerisindeki bit veya döngü tabanlı bloklardan oluşan kütüphanelere sahiptir. XSG blokları ile ortak Simulink blokları arasındaki tek fark, XSG bloklarının sabit nokta sayı gösterim formatıyla ayrık zaman alanlarında kullanılabilmesidir.

XSG, FPGA uygulaması için temel DSP algoritmalarını anlamak için yararlı bir araçtır. FPGA'lar, uygulamaya özel donanımdaki programlanabilir DSP'lerden önemli ölçüde daha yüksek performansa sahip gerçek zamanlı algoritmalar için daha iyi bir

platform sunar. Çalışma, burada sunulan donanım uygulama sonuçlarının döngü içinde donanım simülasyonu gerçekleştirmek için uzatılacağını göstermektedir [29].

Xilinx ile birlikte kullanılabilen System Generatorün Matlab olarak yapılandırılması gerekir. Bu, Xilinx Blockset'in algoritmik model oluşturmak için doğrudan kullanılacak Matlab Simulink ortamına eklenmesiyle sonuçlanır. Algoritmalar geliştirilir ve modeller, Xilinx Blockset tarafından sağlanan kütüphane kullanılarak kaotik davranışın elde edilmesi sağlanır[16].

Burada tasarlanan Chua devresine ait kaotik davranış sonuçları, Xilinx ISE Simülatörü ile elde edildi [27].

4.1. Sabit Nokta Sayı Temsil Biçimi

Çoğu DSP uygulamasının kalitesi, veri gösterimi için bir kayan noktalı sayı formatı gerektirir. Bunu Simulink gibi yüksek seviye modelleme yazılımı çalıştıran çeşitli bilgisayar sistemlerinde uygulamak basit olsa da, bellekdeki bu işlem kayan nokta aritmetik uygulamasının karmaşıklığı nedeniyle çok daha zordur. Bu zorluklar, sistem tasarımında daha kısıtlayıcı kısıtların bulunduğu taşınabilir DSP sistemlerinde artmaktadır. Bu nedenlerden dolayı, XSG sistemdeki tüm sayısal değerleri görüntülemek için sabit noktalı bir sayı biçimi kullanır. Sistem jeneratörü simülasyon ortamı tarafından sağlanan verileri yazılım ve donanıma dönüştürmek için bazı bloklar sağlar [19].

Sistem üreticisi, Xilinx'ten DSP tasarım aracıdır ve FPGA tasarımı için MathWorks model tabanlı Simulink tasarım ortamının kullanılmasını sağlar. Tasarımlar, Xilinx'e özel blok seti kullanılarak DSP dostu Simulink modelleme ortamında yakalanır. FPGA'nın sentez, yerleştirme ve yönlendirme gibi tüm gelişmiş uygulama adımları, bir FPGA programlama dosyası oluşturmak için otomatik olarak gerçekleştirilir [13]. Simulink için belirlenen Xilinx DSP bloğunda 90'dan fazla DSP yapı bloğu bulunmaktadır. Bu bloklar; toplayıcılar, çoklayıcılar ve kaydediciler gibi genel DSP yapı taşları. İletim yönünde hata düzeltme blokları, FFT'ler (hata düzeltim blokları), filtreler ve anılar gibi karmaşık bir DSP yapı bloğu dizisi de dahil edilmiştir. Bu bloklar, Xilinx IP çekirdek üreticilerini seçilen cihaz için optimize edilmiş sonuçlar elde etmek üzere güçlendirir [13].

Sabit nokta sayı biçimi, tam sayı ve sayının ondalık kısmı ile yazılmıştır. Bu sayı biçiminde tasarlanacak sisteme göre, istenen sayının azami büyüklüğü ve duyarlılığı dikkate alınarak bir uzunluk belirlenir.

i : Tam kısım

f : Ondalık kısım

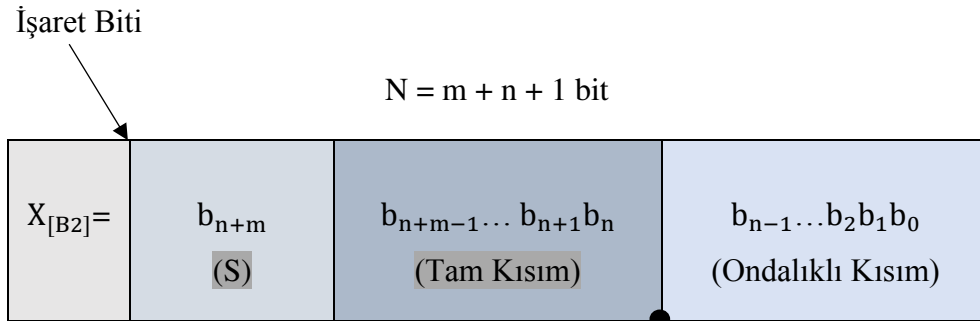
N-bit sabit nokta sayısının matematiksel ifadesi, $Number = n2^{(-m)}$ olarak verilir:

N, n bit'in tamsayı değerini temsil ettiğinde, m noktadan sonraki bit sayısını temsil eder.

Texas Instruments tarafından kullanılan IQmath sabit nokta sayı biçimi, Şekil 10'da gösterilmiştir. Şekil 10, S sinyal bitinin değerini gösterir. Bu değer 0 olduğunda, numaramız pozitif ve 1 olduğunda sayımız negatiftir. Tam bölümünde, sayının tamamı ikili sayı sistemine yazılmıştır. Negatif sayılar 2 tamamlayıcı olarak yazılmıştır. Ondalık kısım tam kısımdaki gibi ikili sayı sistemine dönüştürülür. Denklem (4.1), sistemin sayı sayısının 0,3 sayısının sayı sistemiyle verilmesi [20].

$$\begin{array}{ll} 0.3 \times 0.2 = 0.6 & 0 \\ 0.6 \times 0.2 = 1.2 & 1 \\ 0.2 \times 0.2 = 0.4 & 0 \end{array} \quad (4.1)$$

Bu çalışmada 32-bit işaretli kesirli sayılar, toplama ve çarpma gibi aritmetik işlemlerde kullanılmıştır. Bir tasarımda kesirli sayıları kullanabilmek için, literatürde kayan nokta ve sabit sayı sayı gösterim biçimleri olmak üzere iki yol vardır. Ayrık zamanlı model denklemlerinin tasarım sürecinde, 2'nin tamamlayıcı 32 bitlik sabit sayı sayı biçimi tercih edilir. Q m,n sabit sayı sayı biçimini temsil etmek için kullanılır, burada kesirli kısım için n iken m, tamsayı kısmı için düzenlenmiş bit sayısını belirtir. Bu nedenle, format $Q_{16.16}$ olarak düzenlenmiş ve Şekil 4.1'de gösterilmiştir. Formun çözünürlüğü $2^{-n} = 2^{-16} = 1.5259 \times 10^{-5}$ olarak elde edilmiştir [18].



Şekil 4.1. Q m,n sabit nokta sayı gösterim biçimi

Literatürdeki FPGA üzerinde sabit noktalı sayı tabanlı tasarlanan kaotik osilatör kullanılarak daha az maliyetli ve daha hızlı işlem gücü gerektiren kaos tabanlı mühendislik uygulamaları gerçekleştirilebilir[21].

4.2.FPGA

FPGA bugün sektörde yeni bir lider teknolojidir. Programlanabilen ve binlerce kapıyı bile kaldırabilen bir mantık yongası (yüksek kapasite). Bu nedenle, FPGA entegre devre ve gömülü sistem prototipleme için popülerdir. FPGA'nın yapılandırması, yonganın bir mantık devresi veya kaynak koduyla nasıl çalıştığını belirten bir donanım tanımlama dili (HDL) kullanılarak yapılır. Xilinx System Generator'ün avantajlarından biri, döngüde FPGA donanımını Simulink simülasyonlarıyla kullanma yeteneğidir. Bu tasarım için, Xilinx panosuna programlanan tasarımı uygulamak için donanım koordinasyonu kullanılır. Donanım haritalama, Sistem Üreticisi araçları tarafından sağlanır; Simulink'in bilgisayardaki tasarımıyla bağlantılı olduğu simülasyonu sırasında tasarım donanımını kontrol eder. Tasarım ve FPGA kartı, JTAG kablosu kullanılarak bağlanır. Giriş bilgisayardan beslendi ve çıkış bilgisayara geri verildi. Donanım eşzamanlı simülasyonu, gerçek donanımdaki tasarımı doğrulamanın etkili bir yoludur. Donanım eşzamanlı simülasyonu, FPGA kartına paralel olarak çalışan özel bir eşzamanlı simülasyon bloğu oluşturur. Bu belirli bloğun bağlantı noktaları, özgün tasarım bağlantı noktalarına benzer adlara ve türlere sahiptir [32].

FPGA, sistemlerde dijital mantık uygulamasına bir alternatiftir. FPGA'lar, herhangi bir dijital tasarımı uygulamak için elektriksel olarak programlanabilir silikon yongalardır. İlk statik bellek tabanlı FPGA (genellikle SRAM tabanlı FPGA olarak adlandırılır), 1967'de Wahlstrom tarafından tasarlanmıştır. Bu mimari, bir dizi yapılandırma bitini kullanarak mantık ve ara bağlantı konfigürasyonuna izin verir. Daha sonra, modern çağımızın ilk ticari FPGA'sı 1984 yılında Xilinx tarafından tanıtıldı. Bu FPGA az sayıda klasik Konfigüre Edilebilir Mantık Bloku (CLB) ve giriş / çıkış içerir. 64 CLB ve 58 giriş / çıkış içeren ilk FPGA'dan bu yana, FPGA'lar karmaşıklığı oldukça geliştirmiştir. Günümüzün modern FPGA'ları yaklaşık 330.000 mantık bloğu ve 1100 giriş ve çıkış içerebilir. FPGA'nın temel mimarisi üç ana bileşenden oluşur. Bunlar; mantık fonksiyonlarını yerine getiren programlanabilir mantık blokları, bu fonksiyonları gerçekleştirmek için programlanabilir yönlendirme blokları (ara bağlantı) ve çipin

dışındaki bağlantıları yapmak için giriş / çıkış blokları [19]. FPGA modülü aynı anda maksimum 40 MHz hızda birden fazla işlem yapabilir [27].

VHDL, Verilog, SystemC ve Handle-C gibi donanım tanımlama dilleri FPGA programlaması için sıklıkla kullanılır. VHDL ve Verilog endüstri standardı haline geldi. Donanım Tanımlama Dilleri (HDL), çok sayıda sağlayıcıya sahip, simülasyon ve sentez araçları sağlar. RTL ve yapısal seviyeler bu dillerde alternatif olarak kullanılabilir. SystemC, sistem düzeyinde davranışı modellemek için kullanılan C ++ tabanlı bir kütüphanedir. Temel dil C ++ olduğundan, yazılım işlemleri geleneksel bir HDL'den daha kolay modellenebilir. SystemC için sentez araçları ortaya çıkmakta ve VHDL veya Verilog sentez ürünlerinin vadelerine yaklaşmamıştır. Handel-C ayrıca VHDL veya Verilog ile karşılaştırıldığında yeni bir üründür. Handel-C, İletişim Ardışık İşlem (CSP) modelini takip eder. Handel-C, tasarımcıdan bir işlemdeki paralel işlem bloklarını açıkça tanımlamasını gerektirir. Bu, SystemC 2.0'ın yaptığı gibi süreçler arası iletişim için dahili bir yapı içerir.

FPGA'lar, yüksek programlanabilir lojik ürün sınıfında en gelişmiş ünitelerdir. FPGA'lar, yüksek kapı miktarlarına sahip programlanabilir mantık ürünleridir. Ek olarak, FPGA'lar aynı anda birçok işlemi gerçekleştirmek için programlanabilen yüksek hızlı entegre programlardır. Farklı üreticiler tarafından sunulan FPGA'lar farklı yapılarla sahiptir. İçerdikleri programlanabilir bloklar ve bu bloklar arasındaki bağlantılar düzen mimarisine göre değişir. Ek olarak, bloklar arası bağlantıların programlandığı teknoloji FPGA'lar için önemli bir sınıflandırma özelliğidir [17].

Kaotik sinyallerin elektronik mühendisliğinde önemli araştırma ve uygulama alanları arasında güvenli iletişim cihazları, gürültü jeneratörleri, şifreleme ve rasgele sayı üreteçleri (RSU) oluşturulmaktadır. Bilgi güvenliği sistemlerinde RSU'lar tarafından üretilen rasgele sayılar anahtar olarak kullanılır. Bununla birlikte, anahtarların sistem dışındaki kontrolsüz ortamlarda üretilmesi sistemin güvenilirliğini azaltır. Bu dezavantajın üstesinden gelmek için, donanım kriptolojisi ve güvenli iletişimin geliştirilmesi, DSP, ASIC (Uygulamaya Özel Entegre Devreler) ve FPGA gibi entegre sistemlerde programlanmış şifreleme aritmetiği uygulamaktır. FPGA yongaları bu sorunun üstesinden gelebilir, ancak bu işlemleri yüksek frekanslarda da yapabilir. Yüksek hızı ve kapasitesi nedeniyle FPGA, özellikle yüksek performans ve işlem gücü gerektiren kriptoloji ve güvenli iletişim gibi uygulamalarda bilgi güvenliği kapasitesini geliştirme konusunda önemli bir potansiyele sahiptir [10].

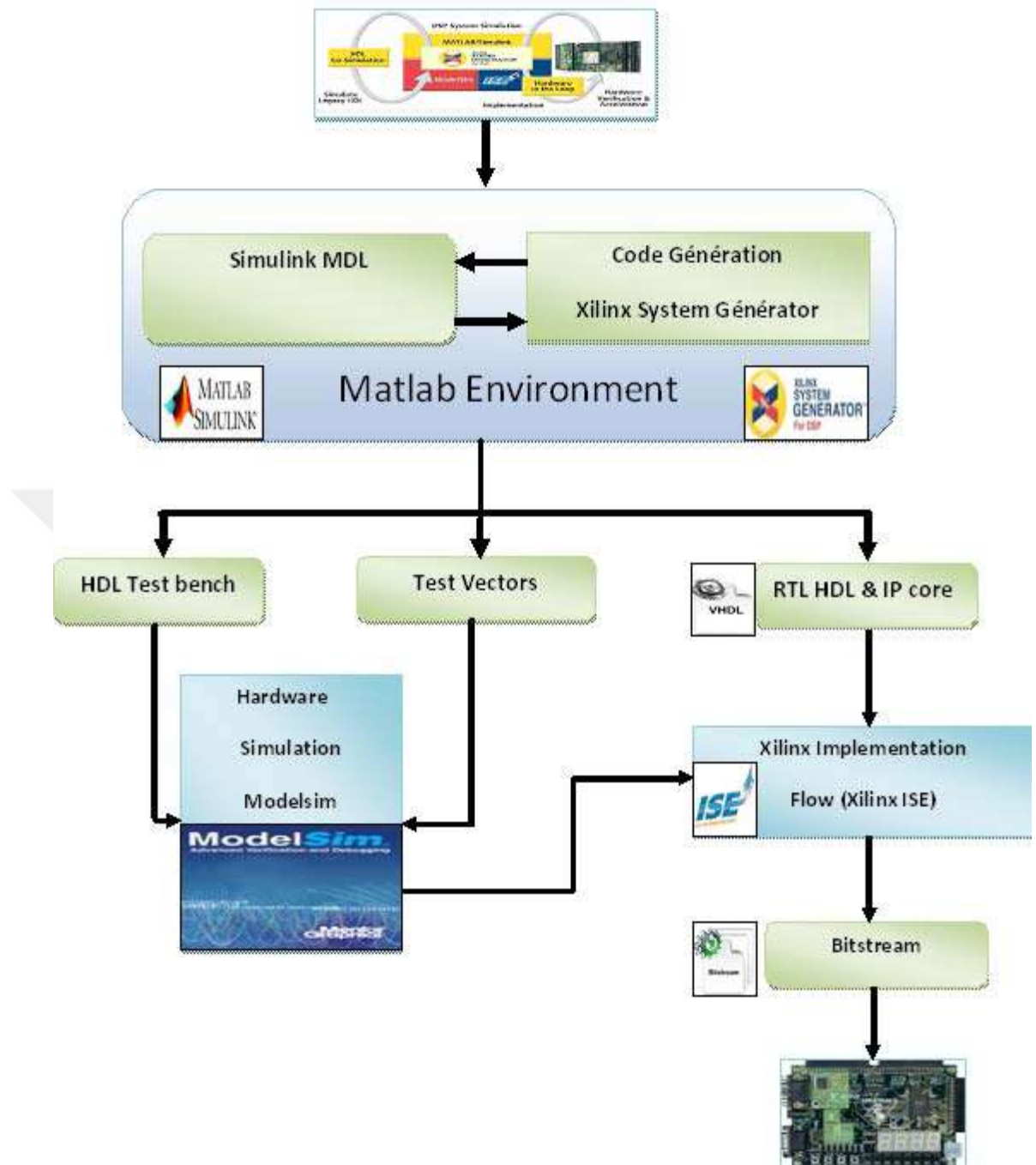
FPGA'lar, birim hesaplama başına düşük enerji tüketimi, yüksek performans ve yeniden yapılandırılabilirlik nedeniyle çekici bir seçimdir. FPGA'nın paralel işlem gücü, DSP, görüntü ve video işleme vb. Bu gibi zorlu uygulamaların modern dünyasında son derece yararlıdır. FPGA'da özel DSP otobüsleri oluşturmak için, Sistem Üreticisi oldukça uygun bir tasarım aracı olarak kullanılır [29].

Dijital sistemlerde FPGA kullanmanın avantajları öge olarak listelenebilir [14]:

- FPGA'lar yüksek hızda çalıştığından dolayı aynı anda birçok işlemi yapabilir durumda programlanabilirler.
- FPGA tasarım süreci kısadır. Prototipin üretimi için çok fazla bekleme gerektirmez, entegre hazır olarak tasarımcıda mevcuttur.
- FPGA'lar defalarca programlanabilmektedir. Tasarımı ilerletmek için tekrar programlama kifayetli olmakta, entegreyi değiştirmeye gerek kalmamaktadır. Bu durumda FPGA zamanlama ve maliyet konusunda avantajlı olmaktadır.
- FPGA ile elde edilen hızlı tasarım ve doğrulama süreçleri, ürünlerin piyasaya daha hızlı sunulmasını sağlamaktadır.
- FPGA'lar meydana getirildikten sonra yapısal olarak test edilirler. Bu sebeple, test için ayrıca bir düzeneğe ihtiyaç duyulmamaktadır. Yalnızca işlevsel olarak arzu edilen şekilde çalışıp çalışmadığının incelenmesi yeterli gelmektedir.
- FPGA'lar firmaların ürün olarak piyasaya takdim ettikleri entegrelerdir. Üretici firmalar FPGA'yı yıllık olarak milyonlar seviyesinde üretilip satışa sunmaktadırlar. Üretilen FPGA sayısının fazla olması, maliyetinin düşük olmasına olanak sağlamakta, dolayısıyla FPGA kullanıcısına maliyet bakımından bir avantaj sağlamaktadır.
- FPGA'lar herhangi bir entegre değişikliği yapılmadan sistem içerisinde tekrar tekrar programlanabilme özelliğine sahiptirler. FPGA'ların bu özelliğinden faydalanılarak, farklı zamanlarda farklı işlevleri yerine getirecek bir tasarım söz konusu ise, FPGA sistem içinde gerekli durumlarda tekrar programlanarak kullanılabilme imkanına sahiptir.
- Mevcut devrede değişiklik yapmaya ihtiyaç duymadan aynı paket tipindeki daha büyük veya daha küçük kapasiteye sahip FPGA geliştirme kartları birbirinin yerine kullanılabilmeyi sağlamaktadır. Bu da, baskı devre kartı değişikliğinin meydana getireceği maliyet ve zamanlama sıkıntılarını ortadan kaldırarak avantaj sağlar.

Beklenen sonuçlar elde edildiğinde, Sistem Üreticisi uygun FPGA kartı için yapılandırılmıştır. Bu tez çalışmasında kullanılan FPGA kartı Kintex-7 xc7k325t-2ffg900 FPGA geliştirme kartıdır. G / Ç planlama ve saat planlama yapılır ve JTAG donanım eş simülasyonu için model uygulanır. Sistem üretici parametreleri ayarlandı ve üretildi.

Verilog HDL'deki model ve programlama dosyası için bir taslak oluşturulur ve Xilinx ISE kullanılarak erişilebilir. Modül FPGA üzerinde davranışsal sözdizimi kontrolü için kontrol edilir, sentezlenir ve uygulanır. Xilinx System Generator'ün kendisi, mimariyi test etmek için Kullanıcı kısıtlamaları dosyası (UCF), Test tezgahı ve Test vektörleri oluşturma özelliğine sahiptir. Xilinx System Generator, öncelikle karmaşık DSP uygulamalarıyla başa çıkmak için oluşturuldu, ancak görüntü teması gibi bununla da çalışan bu temanın başka uygulamaları da vardır. FPGA girişi için uygun olan bir FPGA bit dosyası oluşturmak için gerekli olan bit akımı derlemesi yapılır[16]. Şekil 4.2 Xilinx System Generator için tasarım akışını göstermektedir [30].



Şekil 4.2. Xilinx System Generator ile tasarım metodolojisi

5. XILINX SYSTEM GENERATOR'ÜN SIMULINK GÖSTERİMİ

Xilinx Sistem Üreticisi, FPGA kartlarında hızlı ve kolay tasarımlar oluşturmak, tasarımın Donanım Eş Simülasyonunu gerçekleştirmek ve gerçek zamanlı tümleşik uygulamaları gerçekleştirmek için kullanılan yüksek düzeyde MATLAB-Simulink tabanlı bir yazılım platformudur. XSG, aritmetik, mantıksal, bellek ve Dijital Sinyal gibi uygulamalar için MATLAB-Simulink içinde bit veya döngü tabanlı bloklardan oluşan kütüphanelere sahiptir. XSG blokları ve yaygın Simulink blokları arasındaki tek fark, XSG bloklarının sabit nokta sayı sunum formatı ile ayrık zamanlı alanda kullanılabileceğidir[4].

Bu çalışmada, Bölüm 2'de yapılmış olan Simulink modeli Xilinx System Generator platformunda Xilinx Blokset kütüphanesi kullanılarak, uygun adım aralığı ve direnç değerleri tekrar belirlenerek kaotik davranış gözlemlenmiştir. Dışarıdan sadece çeker sayısı girilerek 5 ile 10 arasında değişen çoklu kaotik davranışlarını XSG platformunda sergileyen bir Simulink model geliştirilmiştir. Modelde farklı parametre değerlerine sahip iki farklı doğrusal olmayan direnç tek ve çift sayıdaki çekerlerin oluşturulması amacıyla kullanılmıştır. Modelde, gerilim kontrollü anahtarlar yerine kaotik çeker sayısı ile karşılaştırma yaparak kontrol bloğuna gönderen Relation yapısı kullanılmıştır. Relation yapısı ile elde edilen sonuçlar iki bloktan alınıp kontrol bloğunda tek bir yapı şeklinde sonuçlanarak çeker sayısına bağlı olarak altı farklı kaotik davranış XSG platformunda da gerçekleştirilmiştir[3].

Denklem (2.1)'deki yapıda kullanılan parametreler Matlab/Simulink modelinde kullanıldığı gibi; $m_0 = -3.2 \text{ mS}$, $m_1 = m_3 = m_5 = m_7 = m_9 = -0.2 \text{ mS}$, $m_2 = m_4 = m_6 = m_8 = -2.2 \text{ mS}$, $C_1 = 10 \text{ nF}$, $C_2 = 100 \text{ nF}$, $L = 18.68 \text{ mH}$ olarak seçilmiş olup elde edilen çekerler ve kullanılan parametreler Tablo 5.1'de özetlenmiştir. Daha önce Tablo 2.1'de verilen parametrelerden farklı olarak Tablo 5.1'de çeker sayısı ile değişen direnç değerleri değişmiştir.

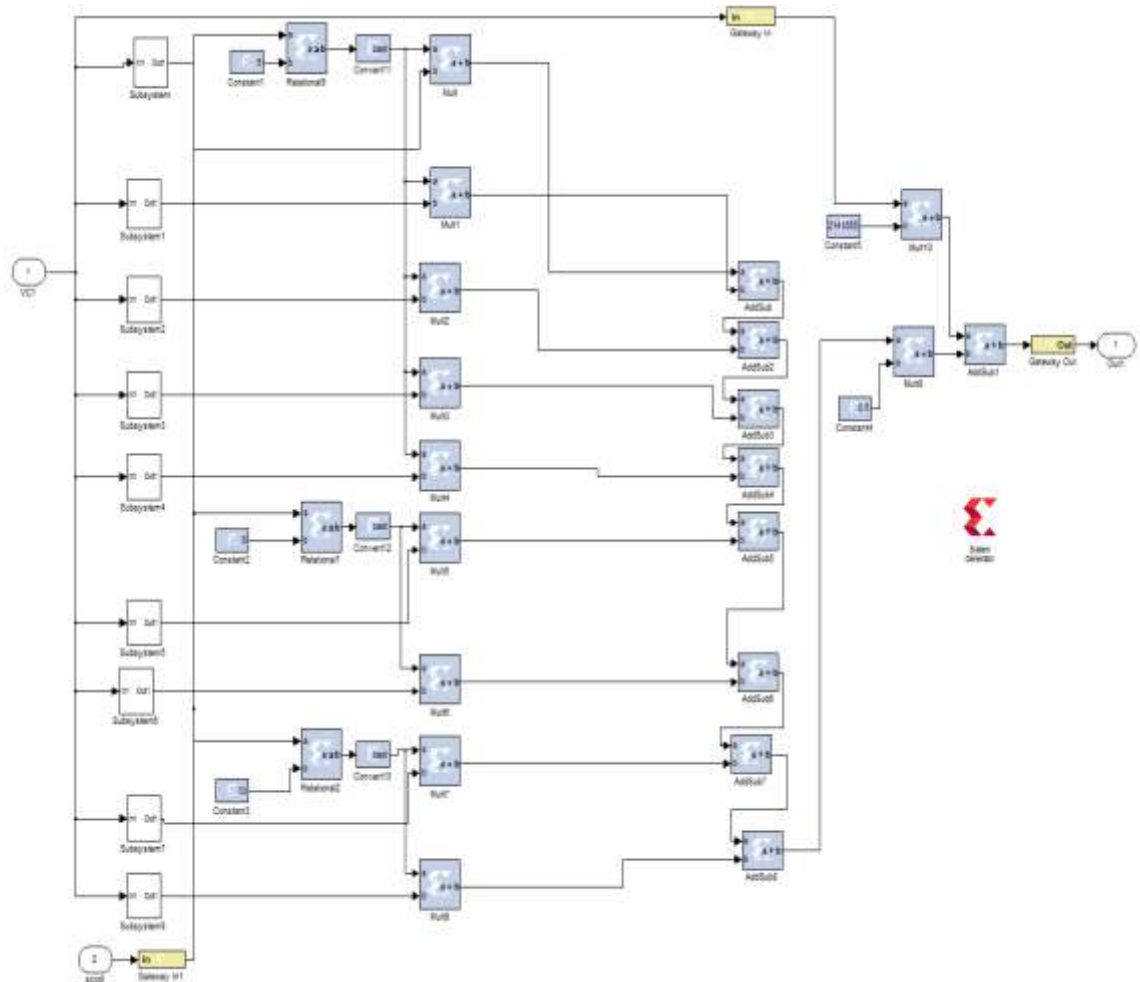
Tablo 5.1. XSG Platformunda Denklem (2.1-2)' deki yapı için kullanılan parametreler

Devre Parametreleri	Çeker Sayısı					
	5	6	7	8	9	10
$R(\Omega)$	1515	1572	1552	1552	1390	1393
$b_1(V)$	-	0.1	-	0.1	-	0.1
$b_2(V)$	0.8	1.1	0.8	1.1	0.8	1.1
$b_3(V)$	1.4	1.55	1.4	1.55	1.4	1.55
$b_4(V)$	3.2	3.2	3.2	3.2	3.2	3.2
$b_5(V)$	3.9	3.85	3.9	3.85	3.9	3.85
$b_6(V)$	-	-	5.8	5.84	5.8	5.84
$b_7(V)$	-	-	6.4	6.6	6.4	6.6
$b_8(V)$	-	-	-	-	8.3	8.7
$b_9(V)$	-	-	-	-	9.2	9.45

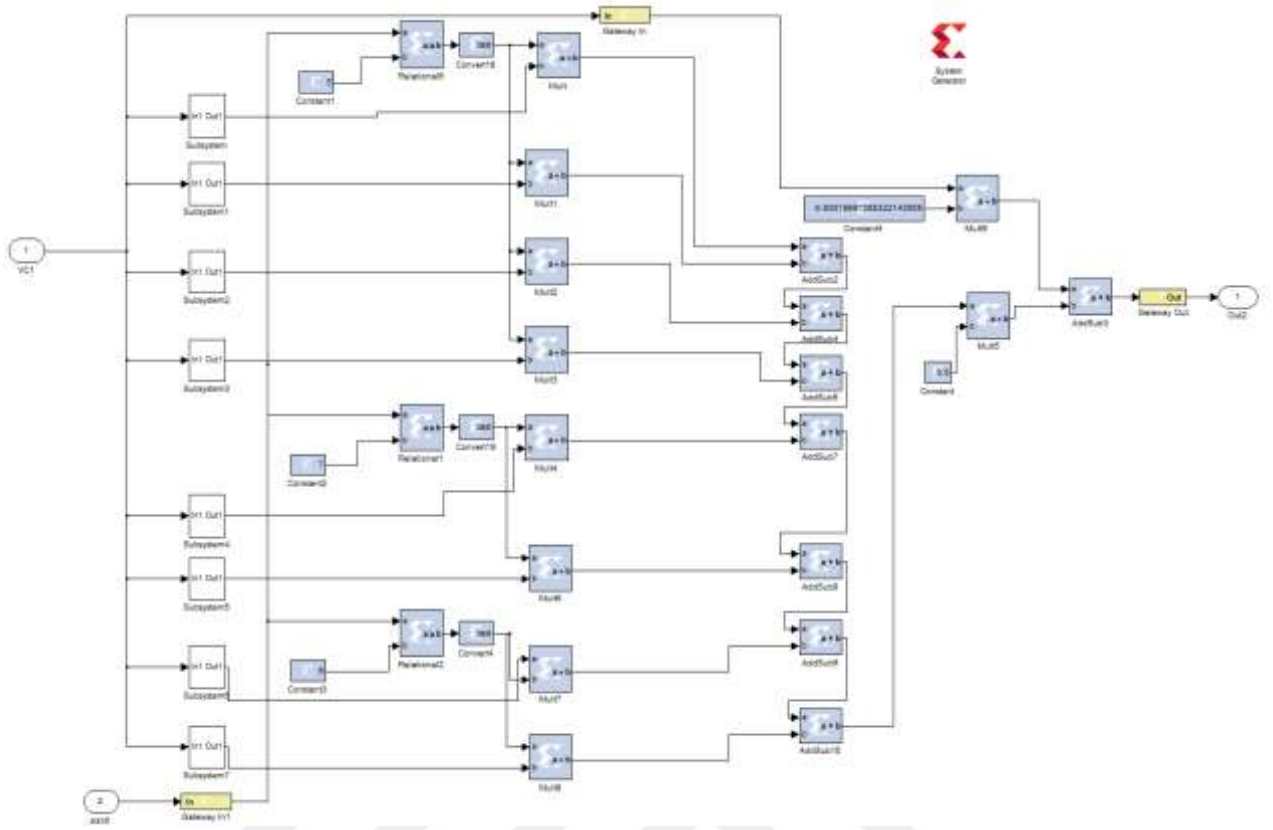
Sistem bir bütün yapı olarak oluşturulmamış anlaşılması ve kolayca kontrol edilebilmesi için Matlab / Simulink modelinde olduğu gibi alt sistemlere bölünerek ele alınmıştır. Modelin bütün olarak yapısı Şekil 5.1'de gösterilmiştir. Matlab / Simulink modelinde olduğu gibi, XSG Platformunda sistem, Denklem (2.1) 'de verilen diferansiyel denklemi baz alınarak denklem çözümü için bu platformda modellemenin yapıldığı Tank_Resonator'dan, Denklem (2.2)' den tek ve çift çeker için oluşturulan PWL alt sistemleri ve çıkışları kontrol eden bir kontrol blok yapısından oluşur. Yapının karmakarışık davranışı, kullanıcı Scroll olarak belirlenen çekme sayısının 5 ila 10'u arasında rakamlar, modelin başka herhangi bir harici girişine ihtiyaç duymadan dışarıdan girilerek gözlemlenmiştir.

Diferansiyel denklemi ayırklaştırmak için uygulanabilecek methotlardan Euler Ayırklaştırma metodu seçilerek denklem ayırklaştırılmıştır ve gerçekleştirilen XSG Simulink modeli Şekil 5.2'de gösterilmiştir. PPD elemanı için kontrol edilen parametre V_{C1} ile gösterilen çıkış değişkenidir.

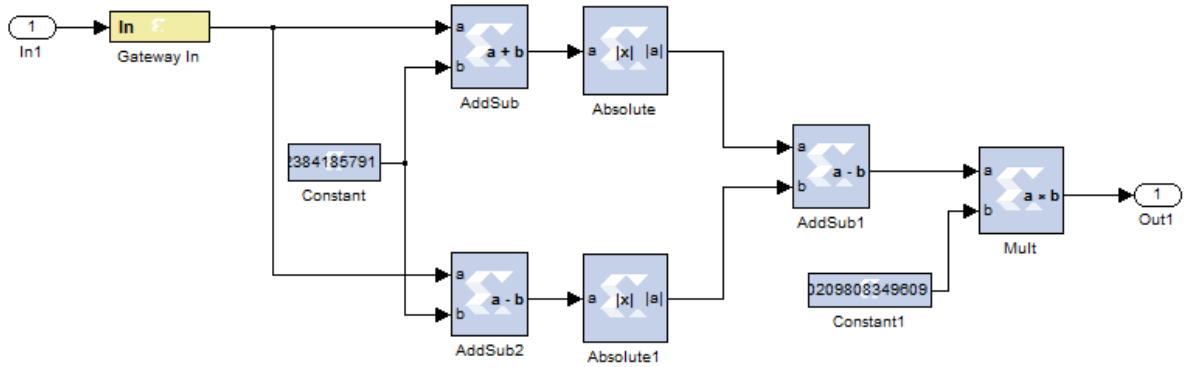
çift çeker sayısı ile bağlı olarak sonuç veren Relation yapısı ile elde edilmiştir. Uygulama gerilim kontrollü olup Tablo 5.1'deki gerilim seviyeleri ile kontrol edilmektedirler.



Şekil 5.3. PPD elemanın PWL_Çift Simulink modeli



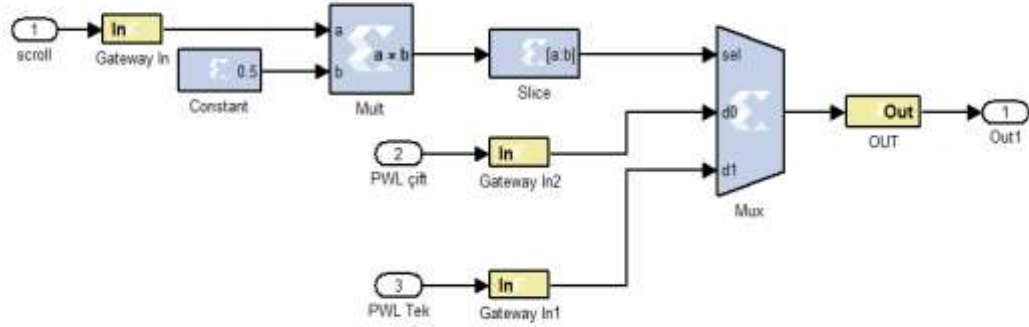
Şekil 5.4. PPD elemanın PWL_Tek Simulink modeli



Şekil 5.5. PPD elemanın alt yapı bloğu

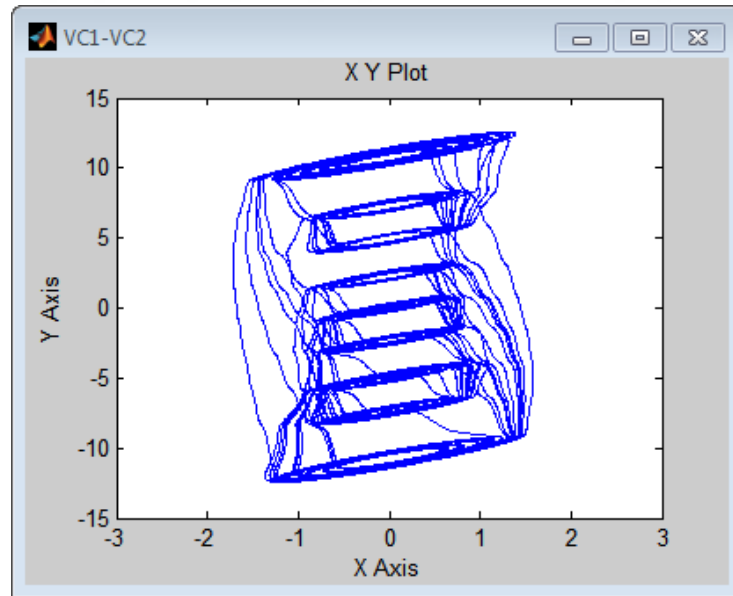
Girilen tek ve çift değerlerin sadece bir yapı ile değerlendirilip modellenmesi için Şekil 5.6'da gösterildiği gibi bir kontrol blok yapısı tasarlanmıştır. Bu kontrol blok yapısı Matlab bünyesinde bulunan Xilinx Blokset ile derlenmiştir. Matlab/Simulink'te kullanılmış olan

çok girişli anahtar (Multiport Switch) yerine Mux kullanılmış olup PWL alt sistem bloklarından elde edilen sonuçlardan biri kontrol blok yapısı ile çıkışa aktarır.

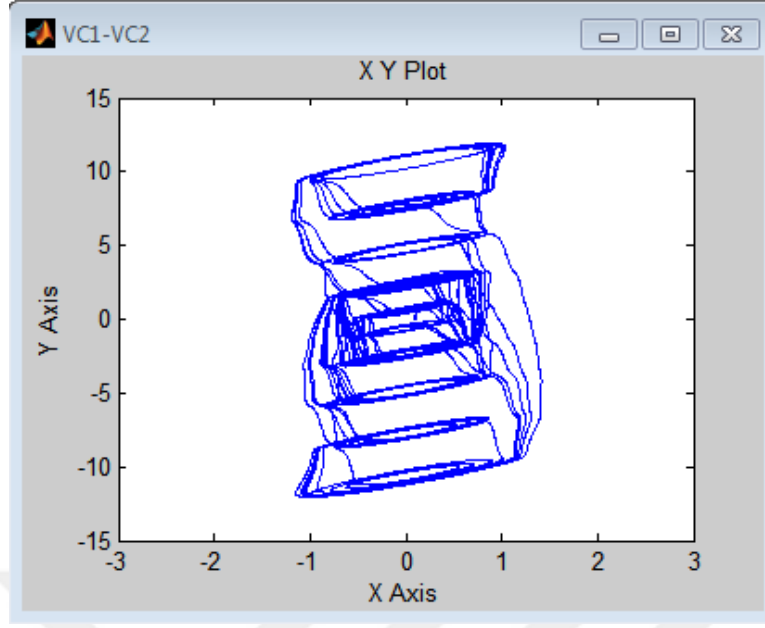


Şekil 5.6. Kontrol bloğu yapısı

XSG platformunda uygulanmış olan modelin tüm alt sistemleri ikinci bölümde Matlab/Simulink ortamında uygulamış olan modelle karşılaştırılarak XSG çıkışları 9 ve 10 çeker için Şekil 5.7’de elde edilmiştir.



a)



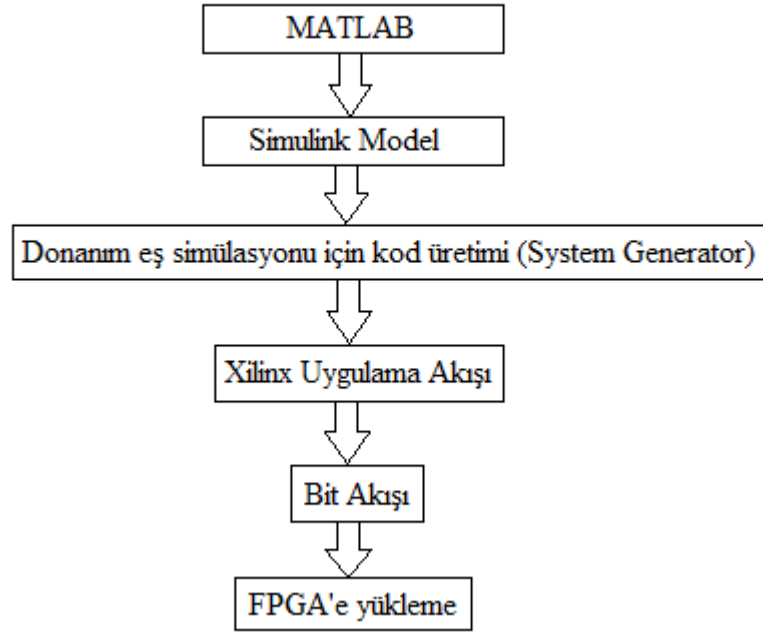
b)

Şekil 5.7. Sistemin XSG çıkışları a) 9 çeker b) 10 çeker

5.1.Xilinx System Generator'ün Donanım Temelli Benzetimi

Bu çalışmada, literatüre sunulan sürekli zamanlı otonom 3-Boyutlu kaotik sistem, FPGA üzerinde IQ-Math sabit noktalı sayı formatında Euler ayrıklaştırma yöntemi kullanılarak Simulink ve XSG platformunda elde edilen kaotik sonuçları Xilinx Kintex-7 ailesi FPGA çipinde sentezlenmiş ve test edilmiştir. Ayrıca FPGA üzerinden alınan kaotik davranış sonuçları Matlab programından alınan ve XSG platformunda elde edilen sonuçlarla karşılaştırılarak sadece 5 scroll değeri için başarılı bir sonuç elde edildiği görülmüştür. Kaotik osilatörlerin bilgisayar ortamında kurgulanan benzetimi ile FPGA üzerinde gerçekleştirilen tasarımın aynı sonuçlar vermesi ile bu sistemlerin donanımsal olarak sentezlenmesi sağlanmıştır.

Uygulama aşamasında, bölüm 5'te gösterilmiş olunan Xilinx System Generator'ün Simulink modeli, Kintex-7 xc7k325t-2ffg900 FPGA geliştirme kartında 32 bitlik 2'lik tamsayı noktalı sayı gösterimi formatlı XSG platformunu kullanarak 5 scroll için donanım temelli benzetimi gerçekleştirilmiştir.



Şekil 5.8. Programlanabilir kaotik osilatör tasarımı uygulamasının tasarım akış şeması

5.1.1. Kintex-7 FPGA Geliştirme Kartı

Kintex-7 ailesi, düşük maliyetli, düşük güçlü ve yüksek performanslı uygulamalarda ASSP'leri (Uygulamaya özel standart parçalar) ve ASIC'leri çıkarmak için tasarlanmış FPGA'lar için yeni bir fiyat / performans noktası belirler. Bu aile, daha önce FPGA'lara erişilemeyen pazar segmentlerinde ve uygulamalarda önemli ölçüde nüfuz ve büyüme potansiyeli yaratır. Kintex-7 cihazları fiyatın yarısı kadardır ve Virtex-6 LXT cihazına eşdeğer performans sunar. Bu fiyatı / performansını elde etmek için, Xilinx bir dizi mimari yenilik kullandı. Kintex-7 ailesi, gerekli bant genişliğini elde etmek için benzersiz bir alıcı-verici teknolojisi ve ambalaj kombinasyonunu içerir, yani, yüksek performanslı alıcı-verici teknolojisi daha düşük maliyetli ambalajlama ile eşleştirilirken, daha yüksek performanslı alıcı-verici teknolojisi daha yüksek performanslı ambalajlama ile eşleştirilir. Bu nedenle, Kintex-7 cihazları en düşük fiyat noktasında en yüksek seviyede sinyal bütünlüğü sunar. "Optimize edilmiş" kaynakların bu birleşimi, maliyet optimizasyonlu paketleme ile birlikte, herhangi bir FPGA ailesinde mevcut olan dolar başına en yüksek performansla sonuçlanır. Ürün özelliklerini Tablo 5.2'de özetlenebilir [33].

Tablo 5.2. Ürün Özellikleri [33]

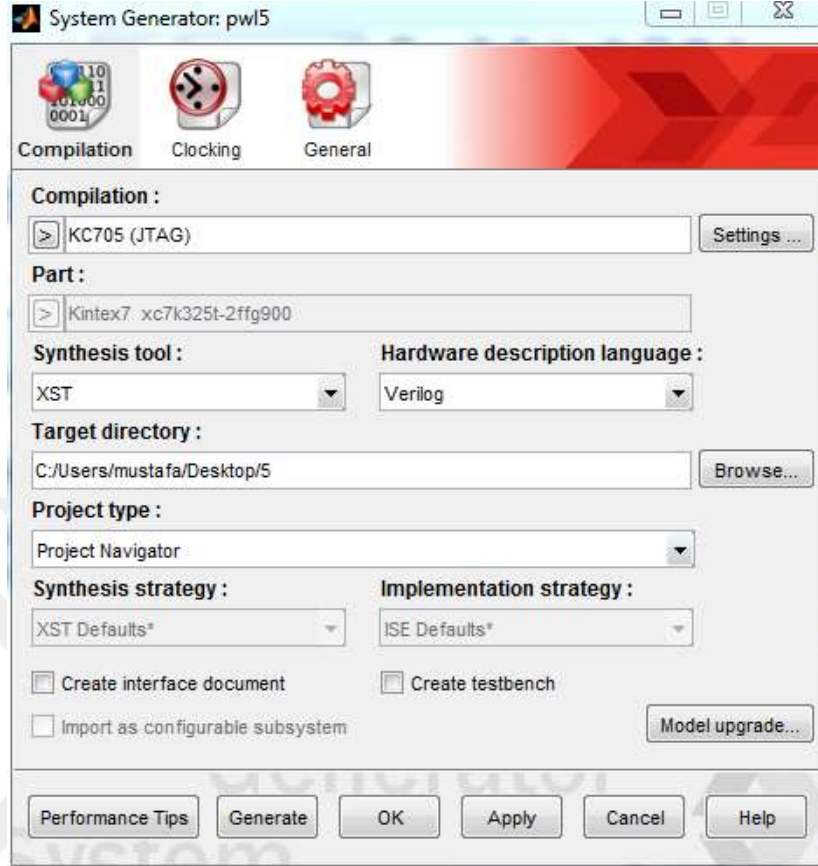
Cihaz	Özellikleri	Kullanım Yerleri
Kintex-7	Endüstrinin en iyi fiyat / performans FPGA'ları	Yüksek hacimli 10G optik kablolu iletişim ekipmanı,
	Önceki nesil Virtex cihazlarından% 50 daha düşük güç	LTE kablosuz ağları, yeni nesil yüksek çözünürlüklü
	Virtex-6 FPGA performansına sahip, yarı fiyatına değerinde	3D düz panel ekranlar ve talep üzerine video
	16'ya kadar yüksek hızlı (10.3 Gb / s) seri alıcı-verici	sistemleri yayınları gibi kablosuz, endüstriyel, tıbbi,
	3.3 V özellikli G / Ç	yayın ve askeri uygulamalar.

Bu tez çalışmasında FPGA kartı olarak Kintex 7 modeli kullanılmıştır. System Generator kullanılarak Matlab 2012a başlatılıp modeli oluşturulduktan sonra Hardware Co-simulation bloğunu çalıştırabilmek için Şekil 5.9'da görüldüğü gibi Kintex 7 kartı JTAG kablosu ile bilgisayara takılmıştır.



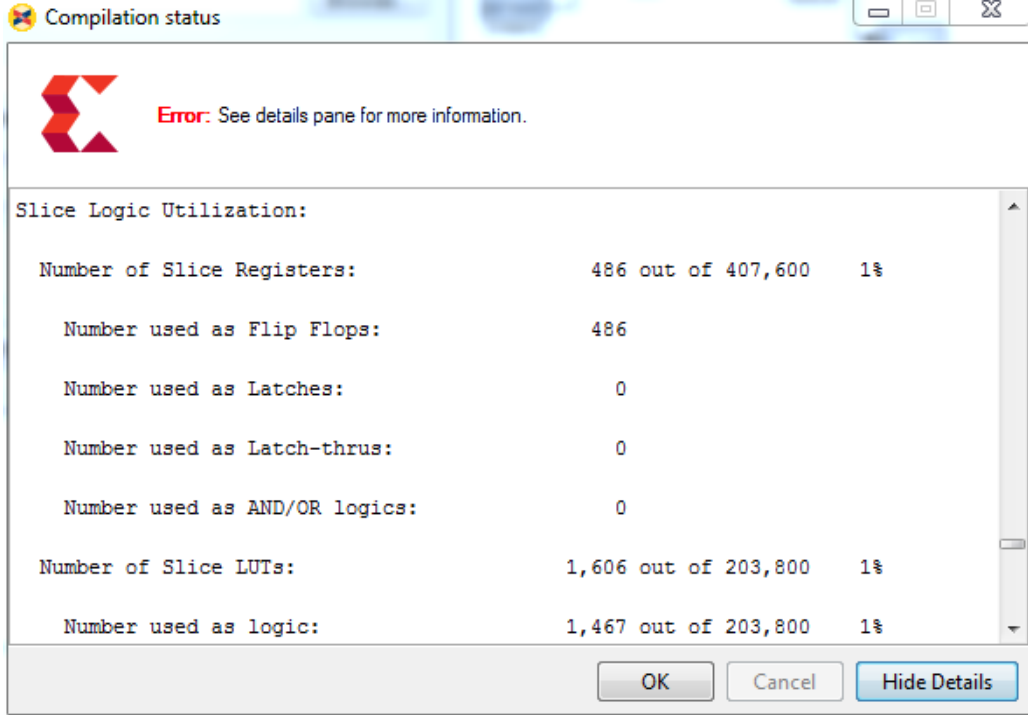
Şekil 5.9. Kintex-7 FPGA Geliştirme Kartı

FPGA kartını tanıtmak için modelde yer alan System Generator bloğuna tıklayıp açılan Şekil 5.10'daki pencereden gerekli veriler girilerek Kintex 7. zip dosyası kaydedilmiştir.



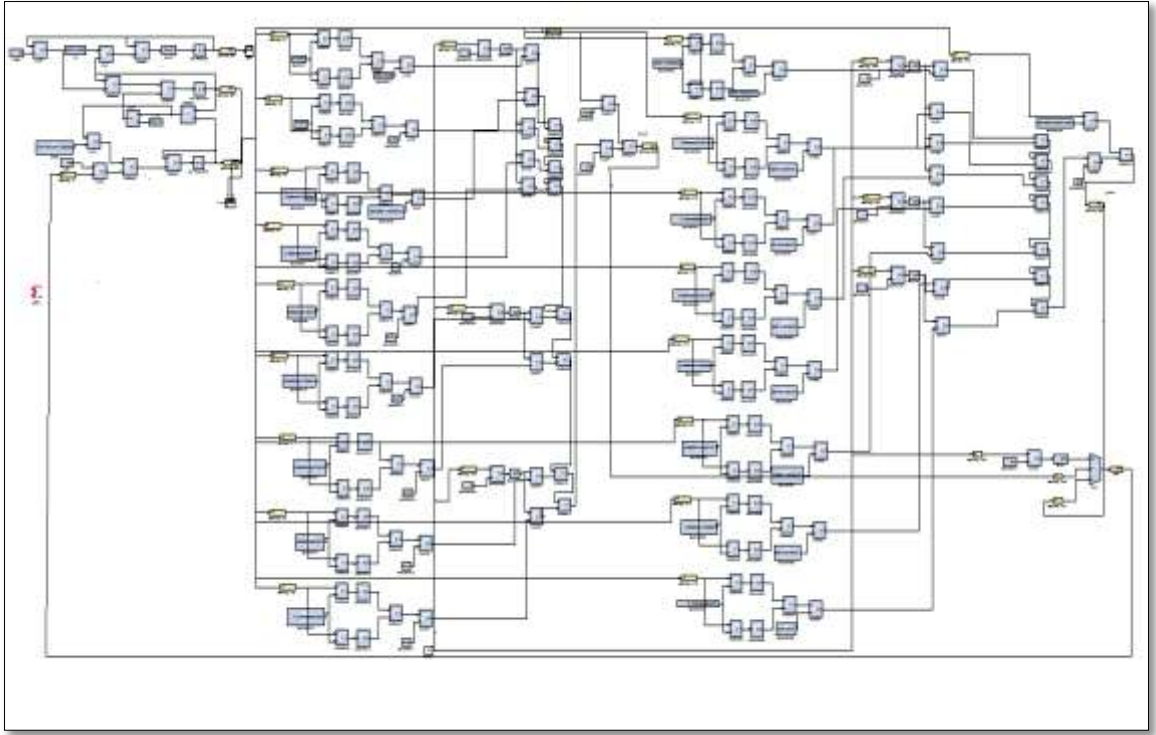
Şekil 5.10. System Generator Blok Penceresi

Xilinx System Generator’de tasarlanan modeli FPGA’e yüklemek için Hardware Co-simulation bloğu oluşturulmuştur. Bu bloğu oluşturmak için model üzerindeki System Generator bloğuna tıklayıp Hardware Co-simulation listesine kaydolan Kintex 7 modeli seçilerek Generate butonuna basılmıştır. Modelin 5 scroll için göstermiş olduğu kaotik davranış FPGA üzerinde de incelenmiştir. Elde edilen bu kaotik davranış modelde Hardware Co-simulation bloğu oluşturulmadan önce bağlı olan Scope bloklarında görülen kaotik davranışlarla aynı olduğu teyit edilmiştir. Şekil 5.11’de kaynak kullanımına ait bilgiler gösterilmiştir.



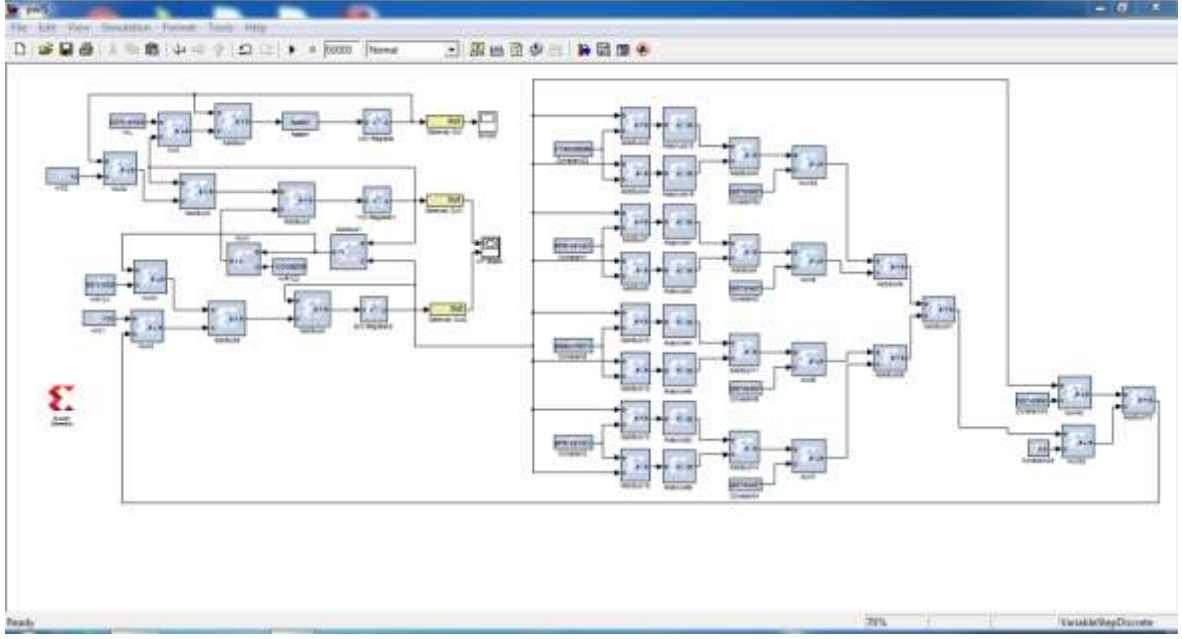
Şekil 5.11. Kaynak Kullanımı

Daha önceki bölümlerde sistemin tamamı bir bütün olarak ele alınmayıp anlaşılabilmesi ve daha kolay kontrol edilebilir olması açısından alt sistem blokları yardımıyla incelenmiştir. Şekil 5.12’de ise sistem tek bir yapı haline getirilerek 5 çeker değeri için simulink modeli Şekil 5.13 ve XSG çıkışı Şekil 5.14 ‘te gösterilmiştir. Sistem tek bir yapı haline getirilmesi durumunda bloklardaki eleman sayılarının fazla olması modelin bağlantılarının kontrol edilmesi açısından birtakım sıkıntılara neden olduğu görülmüştür.

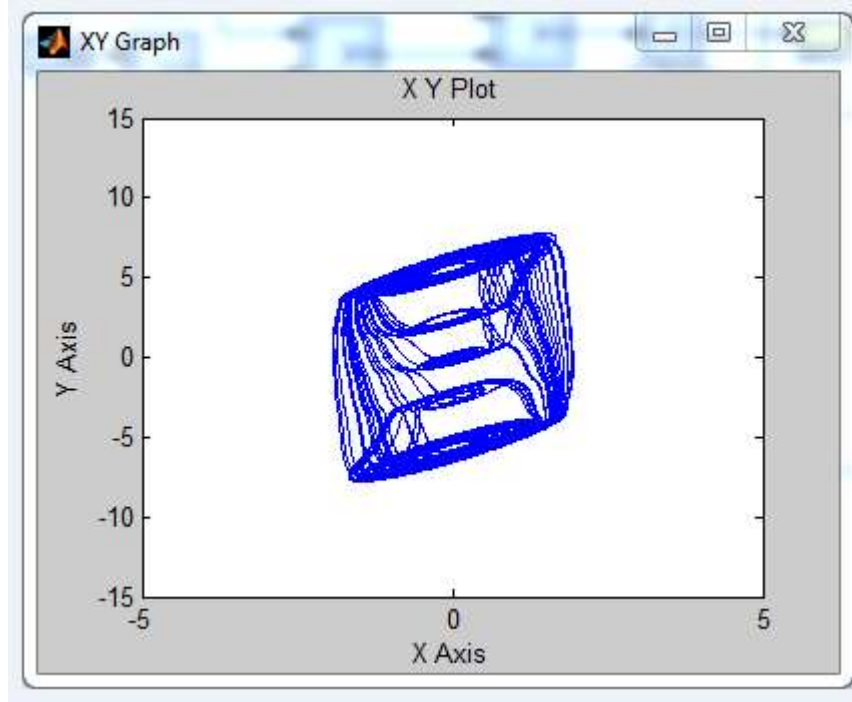


Şekil 5.12. Tüm sistemin tek bir yapıda XSG Simulink modeli

Şekil 5.12 'de görüldüğü gibi model tek bir yapı haline getirildiğinde System Generator'de çok fazla eleman bulunmasından dolayı aynı zamanda FPGA 'ya aktarım konusunda da birtakım sıkıntılar yaşanmıştır. FPGA'nın hafızasının yeterli gelmemesinden dolayı sadece beş scroll için kaotik davranışlar incelenebilmiştir. Matlab/Simulink'te Tank Rezonatör alt sisteminde Euler yöntemi kullanılarak denklemlerdeki parametreler mümkün olduğunca azaltılmaya çalışılmıştır. Ancak yine de diğer bloklarda kullanılan elemanların da çokluğuyla birlikte FPGA'nın hafızası yeterli gelmemiştir.



Şekil 5.13. Modelin 5 çeker için XSG simulink modeli

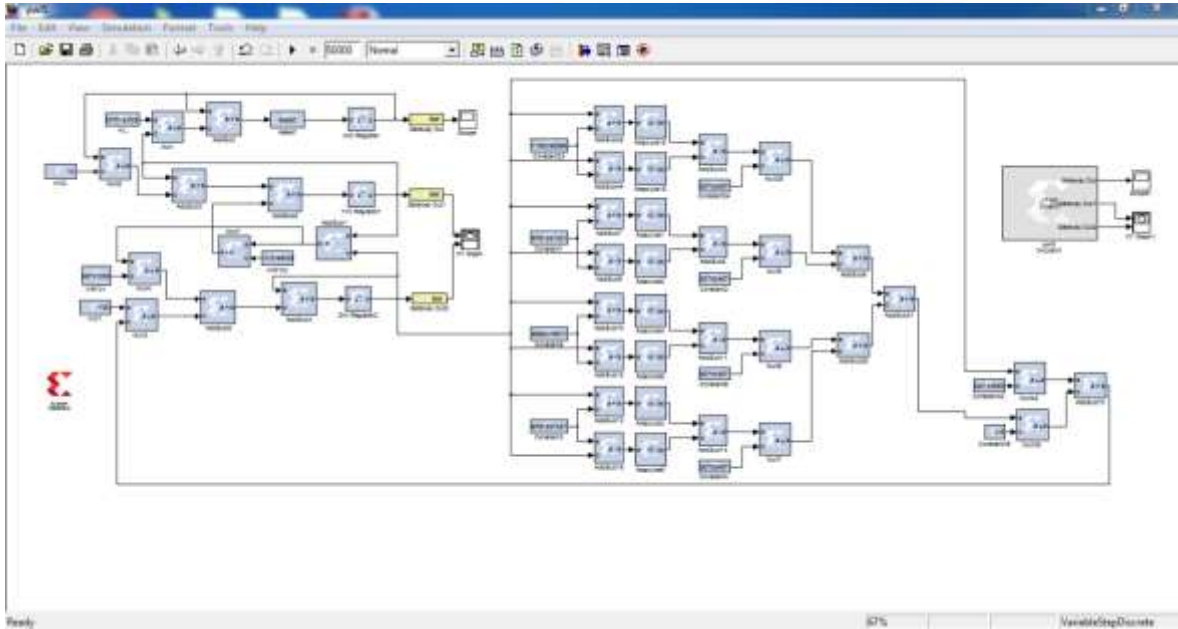


Şekil 5.14. Modelin 5 çeker için XSG çıkışı

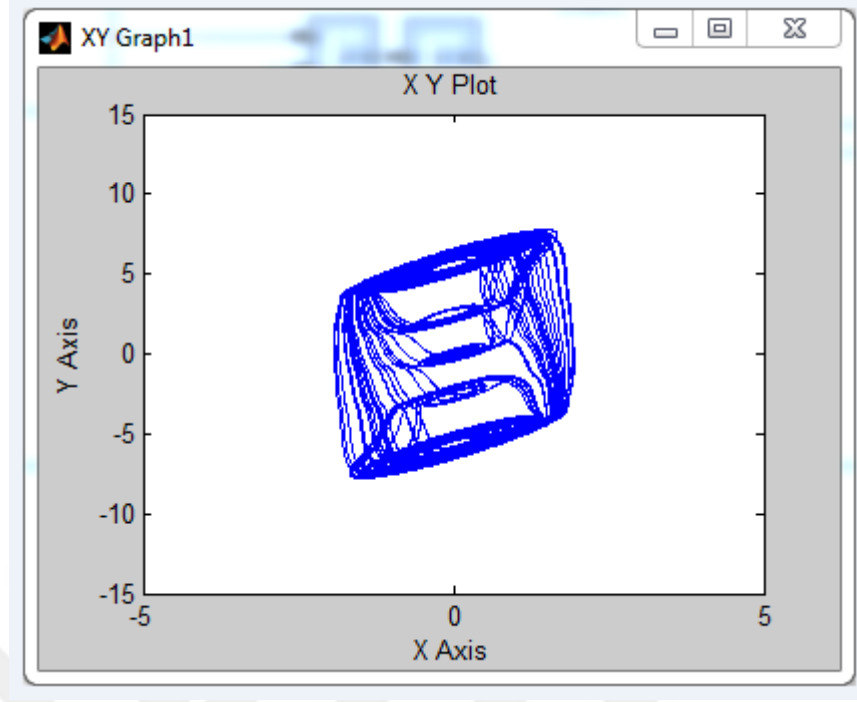
Modele dış girdi olarak beş çeker değeri uygulanıp direnç değeri 1515Ω olarak ele alınarak daha önce Matlab/Simulink ve XSG platformunda elde edilen sonucun elde edilmesi amacıyla FPGA’da gerçekleştirilmesi yapılmıştır. Şekil 5.15’te daha önce XSG

platformunda oluşturulan modelin FPGA uygulama görünümü ve Şekil 5.16'da ise FPGA'da gerçekleştirilmesi sonucu elde edilen kaotik davranış gösterilmiştir.

Bu bölümde elde edilen sonuçlar Xilinx System Generatorde elde edilen sonuçlarla karşılaştırılmıştır. Aynı kaotik davranışı veren model üzerinde daha büyük çeker değerleri için FPGA'nın hafıza yetersizliği nedeniyle sonuç alınamamıştır. Virtex 5, Spartan 3e XC3S1600e gibi FPGA geliştirme kartlarında da model üzerinde çalışma yapılmış olup ancak yine de eleman sayısının fazla olmasından dolayı hafıza yeterli gelmemiştir. Modeli farklı FPGA geliştirme kartlarında test edilerek farklı çeker değerlerinde kaotik davranışları elde etmede kullanılabilir.



Şekil 5.15. XSG platformunda 5 çeker için FPGA'da uygulama görünümü



Şekil 5.16. Modelin 5 çeker için FPGA çıkışı

6. SONUÇLAR

Bu yazıda, doğrusal olmayan devrelerde çok parçalı doğrusal direnç fonksiyonlarını gerçekleştirmek için sistematik bir yaklaşım önerilmiş ve geliştirilmiştir. Çift ve tek sayıda kaydırmalı kaotik çekiciler, önerilen gerçekleştirme şeması ve devre tasarımı izlenerek Chua devresinde deneysel olarak kolayca üretilebilir. Çift veya tek sayıda kaydırma yapan çekicilerin üretilmesi için tek farkın bir rezistörün bağlantısını kesmesi olduğu bulunmuştur. Ayrıca, bölümlerin eğimleri ve parça parça-lineer özelliğin kırılma noktalarının bağımsız olarak ayarlanabildiği, böylece cihazın dinamik aralığının verimli bir şekilde kullanılabileceği bulunmuştur [9].

Bu tez çalışmasında, Kintex 7 FPGA geliştirme kartında Chua kaotik sistemin Xilinx System Generator teknolojisi kullanılarak uygulanması sunulmuştur. Herhangi bir doğrusal olmayan sistemin diferansiyel denklemleri, FPGA editöründe kodlama ve tasarım işlemi öncesinde ayrık hale getirilmelidir. Chua kaotik sistemi, sürekli oluşturulmuş sinyalleri ayırmak için çoğunlukla tercih edilen Taylor serisi genişleme, Runge-Kutta ve Euler ayrıklaştırma yöntemleri kullanılarak ayrık hale getirilir. Uygulama alanına dayalı optimum sayısal yöntem, yöntemlerin doğruluğunu kanıtlayarak ve FPGA kartı üzerindeki kaynak kullanımları açısından tasarımları karşılaştırarak bir yöntem seçilir. Daha sonrasında kodlama ve tasarım işlemi yapılabilir. Bu tez çalışmasında bu ayrıklaştırma yöntemlerinden Euler ayrıklaştırma yöntemi kullanılarak Denklem 2.1'deki diferansiyel denklemi ayrıklaştırılmıştır.

Yalnızca dışarıdan kaotik çeker değeri girilerek 5 ile 10 arasında değişen kaotik davranışları gözlemleyen bir Matlab/Simulink modelinden yola çıkılarak XSG platformunda yeniden derlenerek bu platformda model yeniden incelenmiştir. Modelde mümkün olabildiğince basit yapıdaki bloklar kullanılarak daha iyi anlaşılabilmesi açısından alt sistem blokları kullanılmıştır. Modelde, gerilim kontrollü anahtarlar yerine Xilinx Blockset içerisinde bulunan Relation karşılaştırma elemanı kullanılarak farklı kaotik çeker değerlerinin elde edilmesi amaçlanmıştır. Modeldeki avantaj uygulanan tek bir yapının altı farklı kaotik davranış gözlenme olanağı vermesidir. [3].

Kaos tabanlı bilgi sistemlerinin gelecekteki uygulamaları için önemli bir bileşen, çeşitli kaotik sinyallerin üretilmesi için güvenilir doğrusal olmayan devrelerin donanım uygulamasıdır. Bu, düşük dereceli polinom doğrusal olmayan özelliklere sahip elektronik

cihazlar kullanarak karmaşık çekicilerin üretilmesi konusundaki araştırmaları teşvik etmektedir. Bu amaç için birkaç farklı yaklaşım önerilmiş ve literatürde bulunabilir[9]. Özellikle, n-scroll'lu çekiciler oluşturmak, araştırma ve tasarım için ilginç bir konudur. Önerilen sistematik tasarım şemasını göstermek için, n-kaydırmalı bazı çekiciler deneysel olarak üretildi ve bu bölümde bildirildi [9].

Burada 5 kaydırma, 6 kaydırma, 7 kaydırma, 8 kaydırma, 9 kaydırma ve 10 kaydırma olmak üzere altı farklı çoklu kaotik çekici türü kullanılır. Bu altı farklı çoklu kaydırma kaotik çekicinin her biri için Şekil 2.1 deki elamanların matematiksel değerleri, farklı başlangıç koşulları ve direnç değerleri ile simule edilir [34].

Bu tez çalışmasında, birden fazla çeker sayısı ile kaotik davranışların modellenmesi için bir Matlab/Simulink modeli üzerine çalışma yapılmış olup bu model üzerinde gerekli değişimler yapıp Xilinx System Generator platformunda sonuçlar karşılaştırılmıştır. Daha sonra FPGA geliştirme kartı sisteme tanıtılarak tek ve çift çeker değerleri için donanımsal benzetim yoluna gidilmiştir. Model alınan kaotik sistemde her davranış için farklı bir doğrusal olmayan eleman kullanılması gerekmektedir. Bu doğrusal olmayan eleman değerleri Simulink ve XSG platformunda değişiklik göstermektedir. Bu değerlerde modelin göstermiş olduğu kaotik davranışa göre düzenlemeler yapılmıştır. Gerçekleştirilen modelde tek bir PPD yapısı kullanılmıştır. Dışarıdan tek veya çift çeker değeri girilmesi kaotik davranışı gözlemlemek için yeterli olmaktadır. Çeker sayısına bağlı olarak direnç değerleri, Matlab/Simulink ortamında uygulanan modelde Tablo 2.1'deki ve XSG platformundaki model için ise Tablo 5.1'deki değerler dikkate alınarak Matlab comment satırına girilmektedir. Simulink'te simülasyonu elde edilen modeli FPGA ile gerçeklemek için System Generator araçları kullanılarak Şekil 6.7'de görülen 5 scroll için blok yapısı oluşturulmuştur. Döngüsel komutlar kolay işlemsel bloklara çevrilerek model gerçekleştirilmiştir. Elde edilen XSG çıkışı ve FPGA çıkışlarından da görüldüğü üzere FPGA üzerinde donanım tabanlı gerçekleştirilen tasarımın oldukça başarılı olduğu görülmektedir. Görsellik olarak düşünüldüğünde de Simulink modellerinin bu şekilde alt bloklara ayrılarak karmaşıklığı azalttığı düşünülürse avantajlı bir modelleme olduğu söylenebilir.

KAYNAKLAR

- [1] PEHLIVAN, İ. Yeni kaotik sistemler: elektronik devre gerçeklemeleri, senkronizasyon ve güvenli haberleşme uygulamaları. Doktora Tezi, Fen Bilimleri Enstitüsü, Sakarya Üniversitesi, 2010.
- [2] KOYUNCU, İsmail; OZCERİT, Ahmet Turan; PEHLIVAN, İhsan. Implementation of FPGA-based real time novel chaotic oscillator. *Nonlinear Dynamics*, 2014, 77.1-2: 49-59.
- [3] KAYA, Duygu; TÜRK, Mustafa. A Matlab/Simulink model for multi-scroll chaotic attractors. In: *Signal Processing and Communications Applications Conference (SIU)*, 2015 23th. IEEE, 2015. p. 164-167.
- [4] KARAKAYA, Barış, et al. Selection of optimal numerical method for implementation of Lorenz Chaotic system on FPGA. *International Advanced Researches and Engineering Journal, IAREJ*, 2018.
- [5] Türk, M., and Ata, F. “The multi-mode chaotic behaviors: N+N and 2D N-scroll chaotic attractors”, *COMPEL - The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, 25(4), 929-939, 2006.
- [6] KAPITANIAK, T. *Chaos for Engineering*. 1998.
- [7] ZHONG, Guo-Qun; MAN, Kim-Fung; CHEN, Guanrong. A systematic approach to generating n-scroll attractors. *International Journal of Bifurcation and Chaos*, 2002, 12.12: 2907-2915.
- [8] YALÇIN, MÜŞTAK E., et al. Families of scroll grid attractors. *International Journal of Bifurcation and Chaos*, 2002, 12.01: 23-41.
- [9] GÜLTEN, Arif; TÜRK, Mustafa. Examination of chaotic behaviours using bond graph model. *Journal of the Franklin Institute*, 2003, 340.6-7: 415-422.
- [10] FİDAN, Can Bülent; TUNA, MURAT. Kaotik sistemler ve FPGA tabanlı kaotik osilatörlerin gerçek rasgele sayı üretimindeki (GRSÜ) önemi üzerine bir araştırma. *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, 2018, 2018.2018.

- [11] MERAH, Lahcene, et al. A pseudo random number generator based on the chaotic system of Chua's circuit, and its real time FPGA implementation. Applied Mathematical Sciences, 2013, 7.55: 2719-2734.
- [12] YILDIZ, HAVVA. Farklı Türdeki Kaotik Sistemlerin Modellemeleri ve Senkronizasyonlarının Gerçekleştirilmesi. Fırat Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Tezi. Eylül 2013.
- [13] ÖZDEMİR, Koray. Sürekli-Zamanlı Kaos ile Rastgele Sayı Üretici Tasarımı. 2008. PhD Thesis. Fen Bilimleri Enstitüsü.
- [14] TUNA, Murat; FIDAN, Can Bulent. A Study on the importance of chaotic oscillators based on FPGA for true random number generating (TRNG) and chaotic systems. Journal of the Faculty of Engineering and Architecture of Gazi University, 2018, 33.2: 469-486.
- [15] TLELO-CUAUTLE, E., et al. FPGA realization of multi-scroll chaotic oscillators. Communications in Nonlinear Science and Numerical Simulation, 2015, 27.1-3: 66-80.
- [16] RAUT, Neha P.; GOKHALE, A. V. FPGA implementation for image processing algorithms using xilinx system generator. IOSR Journal of VLSI and Signal Processing (IOSR-JVSP), 2013, 2.4: 26-36.
- [17] AKSEBZECİ, B. HAKAN. Bir Asansör Kontrol Sisteminin Petri Netler Yardımıyla Modellenmesi ve XILINX XC2S200E FPGA'sı ile Gerçekleştirilmesi. Niğde Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Ana Bilim Dalı Yüksek Lisans Tezi. Temmuz 2005.
- [18] KARAKAYA, Barış; YENİÇERİ, Ramazan; YALÇIN, Müştak E. Wave computer core using fixed-point arithmetic. In: Circuits and Systems (ISCAS), 2015 IEEE International Symposium on. IEEE, 2015. p. 1514-1517.
- [19] ÖZPOLAT, Erman; KARAKAYA, Barış; GÜLTEN, Arif. FIR Filtre Tasarımı ve FPGA Ortamında Gerçeklenmesi. Fırat Üniversitesi Mühendislik Bilimleri Dergisi, 2017, 29.2: 269-275.

- [20] ŞAHİN, Suhap; KAVAK, Adnan; ÇAVUŞLU, Mehmet Ali. Comparison of number formats on FPGA-based OFDM modem architecture. In: 2013 21st Signal Processing and Communications Applications Conference (SIU). IEEE, 2013. p. 1-4.
- [21] KOYUNCU, İsmail; TUNA, Murat; ALÇIN, Murat. FPGA tabanlı farklı nümerik algoritmalar ile kaotik osilatör tasarımları.
- [22] Chua, L.O., “The genesis of Chua’s circuit”, Archiv fur Elektronik and Ubertragungstechnik, special issue on Nonlinear Networks and Systems, 46, 250-257, 1992.
- [23] YALÇIN, M. ALİ; UYAROĞLU, YILMAZ. ELEKTRİK GÜÇ SİSTEMLERİNDE KAOTİK OLAYLARIN BAŞLANGIÇ ŞARTLARINA HASSAS BAĞIMLILIGI. Sakarya Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 2002, 6.1: 97-100.
- [24] YAMAÇLI, V.; ABACI, K.; KÖSE, E. Chua Devresinin Gerçeklenmesi ve Simülasyonu. In: 6th International Advanced Technologies Symposium, Elazığ-Türkiye. p. 82-86.
- [25] KENNEDY, Michael Peter. Three steps to chaos. I. Evolution. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 1993, 40.10: 640-656.
- [26] SPROTT, J. C. A new class of chaotic circuit. Physics Letters A, 2000, 266.1: 19-23.
- [27] AKGUL, Akif, et al. Chaos-based engineering applications with a 3D chaotic system without equilibrium points. Nonlinear dynamics, 2016, 84.2: 481-495.
- [28] YILMAZ, Serpil. Stochastic resonance in chua's circuit driven by alpha-stable noise. 2012. Master's Thesis. Izmir Institute of Technology.
- [29] ELAMARAN, V.; RAJKUMAR, G. FPGA implementation of point processes using xilinx system generator. Journal of Theoretical and Applied Information Technology, 2012, 41.2: 201-206.
- [30] SAIDANI, Taoufik, et al. Hardware co-simulation for video processing using xilinx system generator. In: Proceedings of the World Congress on Engineering. 2009. p. 3-7.

- [31] GAIKWAD, Suraj R.; GAWANDE, Gopal S. Design and Implementation of Efficient FIR Filter Structures using Xilinx System Generator. International Journal of scientific research and management (IJSRM), 2014, 2.3: 599-604.
- [32] ZAWAWI, N. M., et al. Implementing WCDMA digital up converter in FPGA. In: 2008 IEEE International RF and Microwave Conference. IEEE, 2008. p. 91-95.
- [33] PRZYBUS, Brent. Xilinx redefines power, performance, and design productivity with three new 28 nm fpga families: Virtex-7, kintex-7, and artix-7 devices. Xilinx White Paper, 2010.
- [34] TÜRK, Mustafa; OĞRAŞ, Hidayet. Recognition of multi-scroll chaotic attractors using wavelet-based neural network and performance comparison of wavelet families. Expert Systems with Applications, 2010, 37.12: 8667-8672.

ÖZGEÇMİŞ

1991 yılında Elazığ'da doğdum. İlk ve orta öğrenimimi Elazığ'da tamamladım. Orta öğrenimimi İstiklal İlköğretim Okulunda, lise öğrenimimi ise Korgeneral Hulusi Sayın Lisesin'de tamamlayarak mezun oldum. Lisans eğitimimi 2009 yılında girdiğim Fırat Üniversitesi Mühendislik Fakültesi Elektrik-Elektronik Mühendisliği Bölümünde tamamlayarak 2013 yılında mezun oldum.

Fatma ÇULCU

