



T.C.

**DÜZCE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**FPGA ÜZERİNDE EULER ALGORİTMASI KULLANARAK VAN
DER POL OSİLATÖRÜNÜN 32-BİT IEEE 754-1985 VE IQ-MATH
SAYI STANDARTLARINA GÖRE TASARIMI VE
GERÇEKLENMESİ**

ELİF KAŞİFOĞLU

**YÜKSEK LİSANS TEZİ
ELEKTRİK EĞİTİMİ ANABİLİM DALI**

**DANIŞMAN
DR. ÖĞR. ÜYESİ MUSTAFA DURSUN**

DÜZCE, 2019

T.C.
DÜZCE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**FPGA ÜZERİNDE EULER ALGORİTMASI KULLANARAK VAN
DER POL OSİLATÖRÜNÜN 32-BİT IEEE 754-1985 VE IQ-MATH
SAYI STANDARTLARINA GÖRE TASARIMI VE
GERÇEKLENMESİ**

Elif KAŞİFOĞLU tarafından hazırlanan tez çalışması aşağıdaki jüri tarafından Düzce Üniversitesi Fen Bilimleri Enstitüsü Elektrik Eğitimi Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Dr. Öğr. Üyesi Mustafa DURSUN

Düzce Üniversitesi

Jüri Üyeleri

Dr. Öğr. Üyesi Mustafa DURSUN

Düzce Üniversitesi

Doç. Dr. İsmail KOYUNCU

Afyon Kocatepe Üniversitesi

Dr. Öğr. Üyesi Ferzan KATIRCIOĞLU

Düzce Üniversitesi

Tez Savunma Tarihi: 07/08/2019

BEYAN

Bu tez çalışmasının kendi çalışmam olduğunu, tezin planlanmasından yazımına kadar bütün aşamalarda etik dışı davranışımın olmadığını, bu tezdeki bütün bilgileri akademik ve etik kurallar içinde elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara kaynak gösterdiğimi ve bu kaynakları da kaynaklar listesine aldığımı, yine bu tezin çalışılması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını beyan ederim.

07 Ağustos 2019

Elif KAŞİFOĞLU



TEŐEKKÜR

Yüksek lisans öğrenimimde ve bu tezin hazırlanmasında gösterdiği destek ve yardımdan dolayı değerli Hocam Dr. Öğretim Üyesi Mustafa DURSUN'a içten dileklerle teşekkür ederim. Bu çalışmada yardımını esirgemeyen, bilgisiyle bana yol gösteren, desteğini hiç esirgemeyen saygı değer hocam, sevgili eşim Sadık KAŐİFOĞLU'na sonsuz teşekkürlerimi sunarım. Ayrıca çalışmamda beni motive edip çalışmam boyunca zinde hissetmemi sağlayan mutluluk kaynaklarım çocuklarım Yağmur Adal, Orhan Efe, Mehtap Aça'ya da çok teşekkür ederim. Çalışmam boyunca da desteğini benden hiç esirgemeyen annem Hatice KAŐİFOĞLU ve ablam Mehtap KAŐİFOĞLU'na teşekkürlerimi sunarım.

07 Ağustos 2019

Elif KAŐİFOĞLU

İÇİNDEKİLER

	<u>Sayfa No</u>
ŞEKİL LİSTESİ.....	VII
ÇİZELGE LİSTESİ.....	VIII
KISALTMALAR.....	IX
ÖZET	XII
ABSTRACT	XIII
1. GİRİŞ	1
2. GENEL BİLGİLER.....	12
2.1. FPGA TEKNOLOJİSİ VE KULLANIMI.....	13
2.2. PROGRAMLANABİLİR MANTIK ELEMANLARI.....	13
2.2.1. SPLD - Basit Programlanabilir Mantık Devreleri.....	14
2.2.2. CPLD - Karmaşık Programlanabilir Mantık Devreleri.....	15
2.2.3. FPGA Çipleri.....	16
2.3. VAN DER POL SİSTEMİ	24
2.4. ALAN PROGRAMLANABİLİR KAPI DİZİLERİ.....	26
2.5. IEEE 754-1985 KAYAN NOKTALI VE IQ-MATH SABİT NOKTALI SAYI STANDARTLARI	27
2.6. NÜMERİK ALGORİTMALAR	31
2.6.1. Euler Nümerik Algoritması.....	31
2.6.2. Heun Nümerik Algoritması.....	31
2.6.3. RK4 Nümerik Algoritması	32
2.6.4. RK5-Butcher Nümerik Algoritması.....	33
2.6.5. Dormand Prince Nümerik Algoritması.....	33
2.7. KAOS VE ANALİZ YÖNTEMLERİ.....	34
3. FPGA TABANLI VAN DER POL OSİLATÖRÜ TASARIMLARI	38
3.1. FPGA ÜZERİNDE IEEE-754-1985 TABANLI VAN DER POL OSİLATÖRÜ TASARIMI	38
3.2. FPGA ÜZERİNDE IQ-MATH TABANLI VAN DER POL OSİLATÖRÜ TASARIMI	43
4. FPGA TABANLI VAN DER POL OSİLATÖRÜ TASARIMI TEST SONUÇLARI	47
4.1. IEEE 754-1985 TABANLI VAN DER POL OSİLATÖRÜ TASARIMI TEST SONUÇLARI VE FPGA ÇİP İSTATİSTİKLERİ.....	47
4.2. IQ- MATH TABANLI VAN DER POL OSİLATÖRÜ TASARIMI TEST SONUÇLARI VE FPGA ÇİP İSTATİSTİKLERİ	48
5. SONUÇLAR VE ÖNERİLER	50

6. KAYNAKLAR	52
7. EKLER.....	59
7.1. VIRTEX-6 FPGA ÇİPİ KATALOĐU.....	59
ÖZGEÇMİŐ	70



ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 2.1. SPLD mimari yapısı.	15
Şekil 2.2. CPLD mimari yapısı.	16
Şekil 2.3. Genel FPGA yapısı	18
Şekil 2.4. Örnek bir FPGA çipinin CLB yapısı	19
Şekil 2.5. FPGA çipinin genel yapısı.	20
Şekil 2.6. FPGA çipinin bağlantı çeşitler.	20
Şekil 2.7. Xilinx firmasının ürettiği Virtex-6 FPGA kartı	21
Şekil 2.8. Xilinx FPGA 'ların genel yapısı	21
Şekil 2.9. XC2000 CLB yapısı.	22
Şekil 2.10. Actel FPGA 'ların genel yapısı	23
Şekil 2.11. Act-1 ve Act-2 Lojik Modülleri.....	23
Şekil 2.12. Van Der Pol osilatörü için tasarlanan ana algoritma	25
Şekil 2.13. Van Der Pol osilatörü için tasarlanan fonksiyon algoritması.	25
Şekil 2.14. Van Der Pol osilatörü zaman serileri	26
Şekil 2.15. Van Der Pol osilatörü v - w faz portresi	26
Şekil 2.16. 32-bit IEEE 754-1985 kayan noktalı sayı standardı gösterimi.	28
Şekil 2.17. Örnek Lyapunov üstelleri grafiği.	36
Şekil 2.18. Altın orana sahip faz portreleri.	37
Şekil 2.19. Altın orana sahip kaotik sistemin zaman serisi analizi.	37
Şekil 3.1. Onluk sayı sisteminden 32-bit IEEE-754-1985 kayan noktalı sayı standardına dönüşüm için kullanılan program görüntüsü.	39
Şekil 3.2. FPGA tabanlı kaotik Van der Pol osilatörü birinci seviye blok diyagramı	40
Şekil 3.3. FPGA tabanlı kaotik Van der Pol osilatörü ikinci seviye blok diyagramı.	41
Şekil 3.4. FPGA tabanlı kaotik Van der Pol osilatörü üçüncü seviye blok diyagramı. ...	42
Şekil 3.5. Onluk sayı sisteminden 32-bit IEEE-754-1985 kayan noktalı sayı standardına dönüşüm için kullanılan program görüntüsü	43
Şekil 3.6. Euler-tabanlı kaotik VDPS ünitesinin en üst seviye blok diyagramı	44
Şekil 3.7. Euler- tabanlı Kaotik VDP osilatörü ikinci seviye blok diyagramı.....	45
Şekil 3.8. Euler- tabanlı Kaotik VDP osilatörü üçüncü seviye blok diyagramı	46
Şekil 4.1. Euler-tabanlı VDPS osilatör ünitesi Xilinx ISE Simülatörü sonuçları.	47
Şekil 4.2. Euler-tabanlı VDPS osilatör ünitesi Xilinx ISE Simülatörü sonuçları.....	49

ÇİZELGE LİSTESİ

Sayfa No

Çizelge 1.1. Literatürde son yıllarda yapılan bazı kaotik osilatör ve özellikleri	7
Çizelge 1.2. Literatürdeki FPGA tabanlı kaotik osilatör tasarımı ve özellikleri	8
Çizelge 2.1. IQ-Math Sabit noktalı sayı standardı gösterimi.....	29
Çizelge 2.2. 92,1245 sayısının 8I-8Qsabit noktalı sayı formatına dönüşüm sonucu.....	30
Çizelge 4.1. IEEE-754-1985 sayı standardı kullanılarak tasarlanan FPGA tabanlı Van Der Pol osilatörü Xilinx Virtex-6 çipi için kullanım istatistikleri	48
Çizelge 4.2. IQ- MATH sayı standardı kullanılarak tasarlanan FPGA tabanlı Van Der Pol osilatörü Xilinx Virtex-6 çipi için kullanım istatistikleri	49

KISALTMALAR

ASIC	Application Specific Integrated Circuit (Uygulamaya Özel Tümdevre)
CAD	Computer Aided Design (Bilgisayar Destekli Tasarım)
CLB	Configurable Logic Block
CMOS	Complementary Metal Oxide Semiconductor
CPLD	Karmaşık Programlanabilir Mantık Cihaz (Complex Programmable Logic Device)
DCSK	Diferansiyel Kaos Kaydırmalı Anahtarlama
DP	Dormand-Prince Nümerik Algoritması
DSP	Digital Signal Processors (Sayısal İşaret İşlemciler)
EEPL	Elektriksel Silinebilir Programlanabilir Lojik Aygıt (Electrically - Erasable Programmable Logic Device)
EEPROM	Elektrikle Silinip Programlanabilen Salt Okunur Bellek (Electrically Erasable Programmable Read Only Memory)
EPLD	Silinebilir Programlanabilir Lojik Aygıt (Erasable Programmable Logic Device)
EPROM	Silinip Programlanabilir Salt Okunur Bellek (Erasable Programmable Read Only Memory)
FPA	Field Programmable Analog Array
FPGA	Field Programmable Gate Array (Alanda Programlanabilir Kapı Dizileri)
GAL	Genel Dizi Mantığı (Generic Array Logic Lattice)
GRSÜ	Gerçek Rasgele Sayı Üreteçleri
HDL	Hardware Description Language (Donanım Tanımlama Dili)
IEEE 754	IEEE Floating Point Number Standardı (Kayan noktalı sayı formatı)
I/OB	Input / Output Blocks (Giriş çıkış Blokları)
IQ - Math	Fixed-Point Number (Sabit-Noktalı Sayı Formatı)
IP-core	Intellectual Properties core
ISE	Integrated Software Environment
MAPLD	Askeri ve Havacılık Programlanabilir Mantık Cihazları (Military and Aerospace Programmable Logic Devices)
MATLAB	Matrix Laboratory
MAX	Çoklu Dizi Matrisi (Multiple Array Matrix, Altera)
LUT	Look-Up Table (Değer Tablosu)
ODE	Ordinary Differential Equation (Adi Diferansiyel Denklem)
PAL	Programlanabilir Dizi Mantığı (Programmable Array Logic)
PEEL	Elektriksel Silinebilir Programlanabilir Logic Aygıt

PLA	(Programmable Electrically – Erasable Logics) Programmable Logic Array (Programlanabilir Mantık Dizisi)
PLD	Programlanabilir Mantık Cihazları (Programmable Logic Devices)
PROM	Programlanabilir Salt Okunur Bellek (Programmable Read Only Memory)
RK4	Dördüncü dereceden Runge-Kutta algoritması
RK5	Beşinci dereceden Runge-Kutta algoritması
RSÜ	Rasgele Sayı Üretici
SEA	Self Excited Attractor (Kendinden Uyartımlı Çeker)
SoC	System On a Chip
SPLD	Basit Programlanabilir Mantık Cihaz (Simple Programmable Logic Device)
VHDL	Çok Yüksek Hızlı Tümeşik Devre Donanım Tanımlama Dili (Very High Speed Integrated Circuit Hardware Description Language)
VLSI	Very Large Scale Integrated Circuit (Çok Geniş Ölçekli Tümeştirme)
VPDS	Van Der Pol Sistemi
XOR	Exclusive Or (Özel Veya)
2-B	2-Boyutlu
3-B	3-Boyutlu
4-B	4-Boyutlu
5-B	5-Boyutlu

SİMGELER

a	Kaotik sistem parametreleri
b	Kaotik sistem parametreleri
c	Kaotik sistem parametreleri
d	Kaotik sistem parametreleri
e	Kaotik sistem parametreleri
f	Kaotik sistem parametreleri
t	Zaman
v	Kaotik sistem durum değişkenleri
y	Kaotik sistem durum değişkenleri
y ₀	2. değişken için kaotik sistem başlangıç şartı
z	Kaotik sistem durum değişkenleri
z ₀	3. değişken için kaotik sistem başlangıç şartı
w	Kaotik sistem durum değişkenleri
x ₀	1. değişken için kaotik sistem başlangıç şartı
Λ	Lyapunov üstelleri

ÖZET

FPGA ÜZERİNDE EULER ALGORİTMASI KULLANARAK VAN DER POL OSİLATÖRÜNÜN 32-BİT IEEE 754-1985 VE IQ-MATH SAYI STANDARTLARINA GÖRE TASARIMI VE GERÇEKLENMESİ

Elif KAŞİFOĞLU

Düzce Üniversitesi

Fen Bilimleri Enstitüsü, Elektrik Eğitimi Anabilim Dalı

Yüksek Lisans Tezi

Danışman: Dr. Öğr. Üyesi Mustafa DURSUN

Ağustos 2019, 69 sayfa

Son yıllarda kaos veya kaotik mühendislik uygulamaları mühendisliğin bir çok alanında yaygın bir şekilde uygulanabilirliğini göstermiştir. Literatürde, kaotik mühendislik çalışmalarının en önemli yapılarından biri de kaos sinyal üreticidir. Kaotik işaret üreticinin çalışma frekansı ve diğer özellikleri kaotik mühendislik uygulamalarında oldukça önemlidir. Literatürde farklı dinamik özelliklere sahip 2-boyutlu, 3- boyutlu, 4- boyutlu ve 5- boyutlu gibi pek çok kaotik sistem önerilmektedir. Kullanılan veya tasarlanan platforma göre tercih edilecek kaotik sistem çeşitlilik göstermektedir. Bu tez çalışmasında, gerçek zamanlı kaos uygulamaları için kaotik Van der Pol sistemi FPGA yongasında çalışmak üzere tasarlanmıştır. Sunulan çalışmada, kaotik Van der Pol sistemi, MATLAB üzerinde Euler algoritması ODE (Adi Diferansiyel Denklem) çözücüsü kullanılarak sayısal olarak modellenmiştir. MATLAB üzerinde tasarlanan kaotik Van der Pol osilatörünün sayısal yapısı gerçek zamanlı FPGA yongası üzerindeki kaotik Van der Pol osilatör ünitesi tasarımı için referans olarak alınmıştır. Kaotik Van der Pol sistemi, 32 bit IEEE-754-1985 kayan nokta sayı standardı ve 32-bit (16I-16Q) IQ-Math sabit noktalı sayı standardı kullanılarak Çok Yüksek Hızlı Entegre Devreler Donanım Tanımlama Dili (VHDL) ile kodlanmıştır. Tasarlanan kaotik Van der Pol sistemi, Xilinx ISE Project Navigator programında sentezlenmiş ve Xilinx VIRTEX-6 yonga ailesi, XC6VLX75T cihazı, FF784 paketi üzerinde gerçekleştirilmiştir. FPGA tabanlı kaotik Van der Pol osilatör ünitesinin 32-bit IQ-Math sayı standardına göre çalışma frekansı 498,728 MHz ve 32-bit IEEE 754-1985 sayı standardına göre Van der Pol osilatör ünitesinin Place ve Route işlemlerinden elde edilen maksimum çalışma frekansı 483,074 MHz'dir. Dolayısıyla Euler algoritması kullanılarak FPGA üzerinde tasarımı yapılan 32 bit IQ-Math sabit noktalı sayı standardı tabanlı Van Der Pol osilatörü ünitesi, 32 bit IEEE-754-1985 kayan nokta sayı standardı tabanlı Van Der Pol osilatör ünitesine göre daha yüksek çalışma frekansı ve daha düşük kaynak kullanımı sağlamıştır. Çalışmada ek olarak, FPGA tabanlı Van der Pol osilatörünün IEEE 754-1985 sayı standardına ve 32-bit IQ-Math sayı standardına göre tasarımlarından elde edilen yonga istatistikleri de sunulmuştur.

Anahtar sözcükler: FPGA yongaları, Van der Pol Osilatörü, VHDL, Euler algoritması.

ABSTRACT

USING EULER ALGORITHM ON FPGA VAN DER POL OSCILLATOR OF 32-BIT IEEE 754-1985 AND IQ-MATH NUMBER STANDARD DESIGN AND REALIZATION

Elif KAŞIFOĞLU

Duzce University

Graduate School of Natural and Applied Sciences, Department of Electrical Education
Master's Thesis

Supervisor: Assist. Prof. Dr. Mustafa DURSUN

August 2019, 69 pages

In recent years, chaos or chaotic engineering applications have shown widespread applicability in many areas of engineering. In the literature, one of the most important structures of chaotic engineering studies is the chaos signal generator. The operating frequency and other characteristics of the chaotic signal generator are very important in chaotic engineering applications. Many chaotic systems have been proposed in the literature such as 2-dimensional, 3-dimensional, 4-dimensional and 5-dimensional with different dynamic properties. The chaotic system to be preferred according to the platform used or designed varies. In this thesis, the chaotic Van der Pol system is designed to work on FPGA chip for real time chaos applications. In the present study, the chaotic Van der Pol system is modeled numerically using Euler algorithm ODE (Ordinary Differential Equation) solver on MATLAB. The numerical structure of the chaotic Van der Pol oscillator designed on MATLAB is taken as reference for the design of the chaotic Van der Pol oscillator unit on the real-time FPGA chip. The Chaotic Van der Pol system is encoded in the Multi-Speed Integrated Circuits Hardware Definition Language (VHDL) using the 32-bit IEEE-754-1985 floating-point number standard and the 32-bit (16I-16Q) IQ-Math fixed-point number standard. The designed chaotic Van der Pol system was synthesized in Xilinx ISE Project Navigator program and implemented on Xilinx VIRTEX-6 chip family, XC6VLX75T device, FF784 package. The operating frequency of the FPGA-based chaotic Van der Pol oscillator unit according to the 32-bit IQ-Math number standard is 498,728 MHz and the maximum operating frequency of the Van der Pol oscillator unit according to the 32-bit IEEE 754-1985 number standard is 483,074 MHz. Therefore, the Van Der Pol oscillator unit based on 32-bit IQ-Math fixed point number standard based on the 32-bit IEEE-754-1985 floating point number standard-based Van Der Pol oscillator unit designed on FPGA using the Euler algorithm has a higher operating frequency and lower source use. In addition, chip statistics obtained from the designs of FPGA based Van der Pol oscillator according to IEEE 754-1985 number standard and 32-bit IQ-Math number standard are presented.

Keywords: FPGA chips, Van der Pol oscillator, VHDL, Euler algorithm.

1. GİRİŞ

Kaos veya kaotik sistemler son yıllarda ortaya konulan ve üzerinde yoğun bir şekilde çalışmalar yapılan alanlardan birisidir. Bu çalışma alanlarına örnek olarak osilatör tasarımı [1], biyomedikal [2], haberleşme [3], optik [4], kriptoloji [5], güç elektroniği [6], robotik [7], yapay sinir ağları (artificial Neural networks) [8], bulanık-kontrol (fuzzy-control) [9], sözde rasgele sayı üretici (pseudo random number generator) [10], gerçek rasgele sayı üretici (true random number generator) [11], senkronizasyon [12], görüntü işleme [13], optimizasyon [14] gösterilebilir. Meteoroloji uzmanı ve matematikçi Edward Norton Lorenz tarafından 1963 yılında Kaos biliminin temelleri atıldı. Edward Norton Lorenz, hava durumunu önceden belirleyebilmek için diferansiyel denklem modeli oluşturarak denklemleri çözdü ve bu modelde fazla uğraşmamak adına önceki bulduğu sonuçları yuvarlayarak sisteme geri yüklediğinde önceki çözümden çok farklı sonuçlar elde ettiğini farketti. Lorenz'in yapmış olduğu bu çalışmada başlangıç şartlarındaki en küçük bir değişimin, sonucu tahmin edilemeyen bir yapıya sürüklediği görülmüştür. Bu sayede de Kaotik sistemin temelleri atılmıştır [15]. Bu gelişme o zamana kadar çözülemeyen fiziksel ve matematiksel problemlere ışık tutmuştur. Ayrıca elektronik devre olarak gerçekleştirilen ilk kaos devresi 1984'te geliştirilen Chua devresidir [16, 17]. Basit bir devre olmasına rağmen karmaşık dalanma ve kaos sergilemesiyle kaos olayının aydınlatılmasında örnek model haline gelmiştir.

Literatürde özellikle kaotik işaret üreten kaotik osilatör devreleri özelliklerine bakıldığında [18, 19];

1. Dinamik yapı olup zamana bağlı olarak değişim göstermek,
2. Başlangıç şartlarına karşı hassas bağımlılık,
3. Bu sistemler değişken ve aperiodyk yapıda olup kendi kendilerini tekrarlamazlar. Sınırsız sayıda değişik periyodik salınımlar içermek,
4. Her ne kadar kompleks bir yapıda görünse bile basit bir yapıdan oluşmak,
5. Nonlineer yani doğrusal olmayan yapılardan meydana gelmek,
6. Gürültü benzeri geniş güç spektrumuna sahip olmak,
7. Limit kümesinin parçalı (fraktal) boyutlu olması,

8. Genliđi ve frekansı tespit edilemeyen, ancak sınırlı alanda deđiřen iřaretler iermesi,

9. Kaotik sistemler deterministik yapıda olması niteliklerini tařımaktadır.

Kaotik osilatörlerin ürettiđi sinyallerin gürültü benzeri ve periyodik olmayan davranıřlar sergilemeleri, giriş şartlarına ve sistem parametrelerine oldukça duyarlı bađlı olmaları gibi tipik özellikleri nedeniyle birçok uygulamada kullanılmaktadır. Bu nedenle literatürde farklı özelliklere sahip birçok kaotik sistem önerilmektedir. Yukarıda bu özelliklere genel olarak deđinilmiřtir. Bu kaotik sistemlerden literatürde sistem dinamikleri en iyi bilinen osilatörler Lorenz [20], Chua [21], Rössler [22], Duffing [23] sistemleridir. Ayrıca farklı karakteristik özelliklere sahip literatüre yeni sunulan kaotik sistemlerde bulunmaktadır [24-25]. Kaotik osilatörler sürekli zamanlı (continuous-time) ve ayrık zamanlı (discrete-time) olmak üzere iki bölümde incelenebilmektedir. Sürekli zamanlı kaotik osilatörlerin gereklenebilmesi amacı ile Sayısal İřaret İřlemci (Digital Signal Processors (DSPs)) [26], Uygulamaya Özel Tümleriřik Devreler (Application Specific Integrated Circuits (ASICs)) [27], mikro-kontrolör (microcontroller (μC)) [28] ve Alan Programlanabilir Kapı Dizileri (Field Programmable Gate Arrays (FPGAs)) [29] gibi çeřitli sayısal donanımsal platformlar kullanılmaktadır. Sayısal tabanlı platformlardan birisi olan FPGA ipleri paralel sinyal iřleme ve tekrar tekrar programlanabilme gibi özelliklerinden dolayı literatürde ok sık bir şekilde kullanılmaktadır. Literatürde sunulan bu alıřmalardan bazıları iki grup halinde ařađıda incelenmiřtir. Bunlardan ilki Field Programmable Gate Arrays (FPGA - Alan Programlanabilir Kapı Dizileri) ipleri olmadan yapılan alıřmalar ve diđeri de FPGA ipleri kullanılarak yapılan alıřmalardır.

Yapılan literatür taramasında kaotik sistemlerin farklı platformlarda modellendiđi gözlemlenmiřtir. Bu durumda FPGA ipleri kullanılmadan yapılan alıřmalar örneklendiđinde;

Koyuncu ve arkadaşları tarafından sunulan alıřmada güvenli haberleřme sistemleri için Sprott 94S sisteminin nümerik modeli ve PSpice programı ile elektronik devre modeli tasarlanmış olup devre elemanları ile modellenen kaotik sistem fiziksel olarak da gereklenmiřtir. Ayrıca Sprott 94 S sisteminin elektronik devre elemanları ile tasarlanan kaotik osilatörü ilk defa bu makale ile sunulmuş olup elektronik devre modellemesi kullanılarak, deđiřik sinyal gizleme ve güvenli haberleřme uygulamaları

yapılabileceği üzerinde durulmuştur [30].

Hidalgo ve arkadaşları tarafından sunulan çalışmada, DSP platformu üzerinde çeşitli bilgi sinyallerini kullanarak verici ve alıcısıyla eşleşerek kodunu çözebileceği güvenli iletişim için sistem tasarımını gerçekleştirmişlerdir [31].

Tavas ve arkadaşları; devre tasarımını simule edebildikleri Cadence programında Spectre benzetim aracı kullanılarak, Negatif-gm LC tank devresi tabanlı tümleştirilmiş kaotik osilatör sunmuşlardır. Devrenin tek endüktansa sahip olarak ve bunun da harici olarak bağlanmasıyla devrenin gerçekleşmesi kolaylaştırılmış olup, geniş bir parametre aralığında kaotik işaret ürettiğini gözlemlemişlerdir [32].

Vaidyanathan ve arkadaşlarının sundukları araştırmada, denge, Lyapunov üstelleri ve Kaplan-Yorke boyutu gibi temel karakteristikleri açısından 3D kaotik sistemini çözümlenmişlerdir. Bu çözümlenmeyi yaparken benzetimi yapılan sistemin faz portreleri sunularak LabVIEW tabanlı tasarımını gerçekleştirmişlerdir [33].

Sevinç tarafından sunulan çalışmada; Kaotik salınımlar üreten Lorenz sistemi üzerinde nonlinear sistemler için önerilen adaptif gözleyici tasarım yöntemi başarılı bir şekilde uygulanmıştır. Üçüncü mertebeden ve üç parametrelili bu sistemin parametrelerinden birisinin bilinmediği ve durum değişkenlerinden sadece birisinin erişilebilirliği varsayılarak hem durum değişkenleri hem de bilinmeyen parametrenin bu yöntemle tahmin edilmesi simülasyon sonuçları ile gösterilmiştir. Gizli haberleşme sistemlerine bu gözleyici uygulanırsa iki farklı bilgi sinyalinin aynı kaotik sinyal üzerinden gönderilmesi mümkün olacağı kanısına varılmıştır [34].

Xu ve arkadaşları; Raspberry Pi 3 ve Orcad-PSpice üzerinde Self-Excited Attractor (SEA) kaotik sistemini gerçekleştirmişlerdir. SEA kaotik osilatör kullanılarak Rasgele Sayı Üreteçleri (RNG), sinyal gizleme uygulamasını gerçekleştirmek için tasarlanmıştır. Tasarlanan RNG'ler NIST-800-22 testine tabi tutulmuş ve başarılı sonuçlar alınmıştır. Orcad-PSpice programı üzerinde direnç, opamp, kapasitör gibi devre elemanları kullanılarak hem modelleme hem de gerçek devre tasarımı yapılmıştır. Çalışmada faz portreleri simülasyon ve osilaskop görüntüleri verilmiştir [35].

Demirkol tarafından yapılan çalışmada; tümleştirmeye uygun, parametreleri kontrol edilebilen ve hızlı çalışabilen kaotik osilatör kullanılarak ADC tabanlı yeni bir Gerçek Rastgele Sayı Üreteci (GRSÜ) tasarlanmıştır. Çıkıştan alınan veriler; en geçerli rastgelelik testlerinden biri olan NIST 800-22 testine tabi tutulmuştur. Tasarlanan

RSÜ'nün çıkışı algoritmayla işleme konulmadığından ve de giriş işaretinin yüksek frekanslı olmasından dolayı çıkış hızı şu an halihazırda olan yapılardan daha yüksek olduğunun sonucunu gözlemlemişlerdir [36].

Rajagopalan ve arkadaşları tarafından 5gerçek sayı üretici tasarımı CMOS boolean kaotik sayı üreticine göre ASIC yaklaşımı 45 nm CMOS teknolojisine bağlı Cadence virtuoso aracı vasıtasıyla modellenmiştir. Bu tasarımın kriptografik haberleşmede kullanılabileceğine değinilmiştir [37].

Yardım ve arkadaşı tarafından sunulan çalışmada; Diferansiyel Kaos Kaydırmalı Anahtarlama (DCSK) modeli Lorenz sistemi kullanılarak incelenmiştir. Yapılsimülasyon sonuçları da Chua devresi simülasyon sonuçlarıyla karşılaştırılmış ve Chua devresi kullanılarak yapılan DCSK simülasyonunun teorik sonuca daha yakın olduğunu gözlemlemişlerdir [38].

Çiçek ve arkadaşları tarafından yapılan çalışmada, yeni 3D kaotik sistemi Matrix Laboratory (MATLAB) ve Orcad-PSpice programlarını kullanarak dört parametrelili ve dört nonlinear terim özelliklerine sahip olan bir prototip sunmuşlardır. Kaotik sisteme ait zaman serileri, faz portreleri, Lyapunov üstelleri analizleri yapmışlar ve aktif kontrol senkronizasyonu ile güvenli kaotik maskeleyme haberleşme uygulaması tasarımı sunmuşlardır [39].

Özdemir ve arkadaşları tarafından sunulan çalışmada; yeni bir rastgele sayı üreticini sürekli zamanlı kaotik işaretini; jerk osilatörü kullanarak gerçekleştirmişlerdir. Tasarlanan yapıdan elde edilen bit dizileri rastgele sayı testine tabi tutulmuş ve NIST–800–22 testinden başarıyla geçmiştir [40].

Chiu ve arkadaşları tarafından yapılan çalışmada, Euler algoritmasını kullanarak modellenen 3D Lorenz kaotik sistemini birkaç direnç ve mikroişlemci gibi az miktarda elektronik devre elemanı kullanarak gerçekleştirmişlerdir [41].

Özdemir tarafından; tümleşik yapıda sürekli-zamanlı bir kaotik işaret üretici kullanarak yeni bir rastgele sayı üretici tasarımı yani Sürekli zaman çift sarmallı kaotik osilatörden elde edilen işaret girişli kaotik işaret tabanlı yeni bir GRSÜ tasarlanmış ve nümerik analizleri üzerine çalışılmıştır. Nümerik analizden elde edilen sonuçlar kullanılarak yeni tasarlanan rastgele sayı üretici laboratuvar ortamında ayrık elemanlarla tasarlanmış ve gerçekleştirilen devreden elde edilen bit dizisi NIST–800–22 dokümanında yer alan rastgelelik testine tabi tutulmuş ve başarılı sonuç elde edilmiştir. Yani bu çalışmada

herhangi bir algoritmaya ihtiyaç duyulmadan tasarlanan GRSÜ'nün çıkışı rastgeleliği sağlamıştır. Böylece yeni bir kaos tabanlı RSÜ tasarımı yapılmıştır [42].

Murillo-Escobar ve arkadaşlarının yaptıkları çalışmada, kaos tabanlı şifreleme algoritmasını 32 bit mikroişlemci (mikrocontroller (μC)) altyapılı olarak yüksek güvenli haberleşme için prototiplemişlerdir. Bu prototip yapılırken karşılaşılan dezavantaj çip üzerindeki hafızanın düşük olması ve düşük frekansta çalışması emsal gösterilmiştir. Bu çalışma test edilmiş ve mükemmel şifreleme özelliği sunduğu için kriptolojide rahatça kullanılabilceğini sunmuşlardır [43].

Pareschi ve arkadaşları da güvenli haberleşmede kullanılmak üzere rasgele sayı 4 üretici tasarlamışlardır. Tasarım gerçek rasgele sayı üretici tasarımını CMOS üzerinde yapmışlardır. Tasarlanan prototiplerin 40 Mbit/s ve 100 Mbit/s bit üretim hızına sahip olduğunu gözlemlemişlerdir [44].

Ren ve arkadaşları tarafından yapılan çalışmada; TI TMS320C6713 DSP çipi üzerinde Chen kaotik sistemi gerçekleştirilmiştir. Kaotik sinyal, ses sinyalleri ile şifrelenerek kaotik haberleşme şemasında ve güvenli haberleşmede kullanılmıştır. Hiper-kaotik iletişim sistemi için senkronizasyon prototipi yapılmıştır [45].

Sundarapandian ve arkadaşlarının yaptıkları çalışmada, yeni 3D otonom kaotik sistem tanımlanmıştır. Bu sistemin özelliği ise; tek bir kübik nonlineeriteye sahip olmasıdır. Bu sisteme ait bazı özellikler (denge, dinamik davranış, Lyapunov üs spektrumu yöntemi)analitik ve sayısal olarak incelenmiş olup Orcad- PSpice programı kullanılarak devresi oluşturulmuş ve kaos tabanlı mühendislik uygulamalarında kullanılmak üzere literatüre geçmiştir [46].

Pehlivan yaptığı doktora tez çalışmasında yeni kaotik sistemler tanıtarak devrelerin simülasyon ve devre gerçeklemelerini yaparak 2007'de literatüre yeni garip çekerler kazandırmıştır [47, 48].

Deng vd. 2014 yılında yeni üç boyutlu 7 terimli, 3 parametrelili ve bir tane pozitif değer içeren otonom kaotik sisteminin önerdiği yeni sistemin faz portesi, denge noktası, Lyapunov üstelleri, Poincare haritalama ve çatallanma diyagramı gibi temel özelliklerini inceleyerek çalışmalarını sunmuşlardır [49].

Abooe ve arkadaşları tarafından yapılan çalışmada; üç boyutlu otonom yeni kaotik sistemin dinamik denklemlerini çıkararak bazı temel özelliklerinden olan garip çeker, denge kararsızlığı, başlangıç şartlarına duyarlılık, Lyapunov üstelleri, fraktal boyut

analizlerini inceleyerek Orcad PSpice yazılımıyla simülasyon ve donanımsal gerçekleştirilmesi yapılarak sonuçları karşılaştırılmıştır [50].

Zhoua ve arkadaşları tarafından; yeni otonom kaotik çeker önermişlerdir. Sistemin özellikleri 3 boyutlu, 6 terimli, 3 parametrelidir. Önerilen sistemde bazı dinamik davranışları (Lyapunov üstelleri, Poincare haritalama, fraktal boyut, çatallaşma diyagramı, sürekli spektrumu) hem nümerik hem de analitik olarak inceleyerek yorumlamışlardır [51].

Pehlivan ve arkadaşları tarafından sunulan çalışmada; yeni bir otonom kaotik sistem tanımlanmıştır fakat bu sistemin diğerlerinden farkı; 3D, sürekli zamanlı, 8 terim, 2 parametre, 2 kuadratik doğrusalsızlık içeren bir sistem olmasıdır. Sistemin dinamik yapı ve davranışları analitik ve nümerik olarak incelenmiş ve MATLAB, Orcad-PSpice programları kullanılarak simülasyonu yapılmış sonra devre uygulamaya koyulmuş ve somut bir devre olarak gerçekleştirilmiştir [52].

Aşağıdaki Çizelge 1.1'de bazı kaotik osilatör tasarımı ve özellikleri tablo olarak listelenmiştir.

Çizelge 1.1. Literatürde son yıllarda yapılan bazı kaotik osilatörler ve özellikleri [16].

Çalışmayı Yapan	Kaotik Sistem Özellikleri	Kaotik sistem Denklemleri	Faz Portreleri
Koyuncu ve Arkadaşları [30]	3 boyutlu 6 terimli	$\frac{dx}{dt} = -\left(\frac{1}{R_1 C_4}\right) \cdot x(t) - \left(\frac{1}{R_8 C_4}\right) \cdot y(t)$ $\frac{dy}{dt} = \left(\frac{1}{R_7 C_7}\right) \cdot x(t) + \left(\frac{1}{R_9 C_7}\right) \cdot z^2(t)$ $\frac{dz}{dt} = \left(\frac{1}{R_{14} C_8}\right) \cdot x(t) + \left(\frac{1}{R_{13} C_8}\right) \cdot V_n$	
Pehlivan, [47, 48]	3 boyutlu 8 terimli 1 parametrelili Otonom	$\dot{x} = y \cdot (1 - z)$ $\dot{y} = -a \cdot x + y \cdot (z + 1)$ $\dot{z} = a - x \cdot y - y^2$	
Abooe vd., [50]	3 boyutlu 7 terimli 6 parametrelili Otonom	$\dot{x} = a(y - x) + byz^2$ $\dot{y} = cx + dxz^2$ $\dot{z} = hz + kz^2$	
Deng vd., [49]	3 boyutlu, 7 terimli, 3 parametrelili Otonom	$\dot{x} = -x - 2 \cdot y$ $\dot{y} = -x \cdot z - b \cdot y - a \cdot x$ $\dot{z} = x \cdot y - c \cdot z$	
Zhoua vd., [51]	3 boyutlu 6 terimli 3 parametrelili Otonom	$\dot{x} = a \cdot (y - x)$ $\dot{y} = b \cdot x - x \cdot z$ $\dot{z} = x \cdot y + c \cdot z$	
Pehlivan vd., [52]	3 boyutlu 8 terimli 2 parametrelili Otonom	$\dot{x} = y - x - a \cdot z$ $\dot{y} = x \cdot z - x$ $\dot{z} = -x \cdot y - y + b$	

FPGA çipleri kullanılarak yapılan çalışmalar örneklendiğinde karşımıza aşağıdaki Çizelge 1.2'deki gibi bir tablo çıkmaktadır.

Çizelge 1.2. Literatürdeki FPGA tabanlı kaotik osilatör tasarımı ve özellikleri.

Literatürde Yapılan Çalışmalar	Kullanılan Kaotik Osilatör	Kullanılan Nümerik Yapı	Kullanılan Sayı Standardı	Kullanılan Platform	Çalışma Frekansı (MHz)
Pano-Azucena ve arkadaşları, 2019 [53]	Kesirli dereceden kaotikosilatör	Grünwald–Letnikov metodu	IQ-Math Sabit noktalı sayı standardı - 32-bit	Cyclone IV GX FPGA DE2i-150-Altera	77,59 ve 84,9
Tuna ve arkadaşları, 2019 [54]	Lü-Chen Kaotik osilatörü	Heun algoritması	IQ-Math Sabit noktalı sayı standardı - 32-bit	Virtex-6 FPGA chip.	464,688
Rajagopal ve arkadaşları, 2018 [55]	Memristor ve memkapasitör tabanlı hiper-kaotik sistem	Adomian Decomposit metodu	IQ-Math Sabit noktalı sayı standardı .	--	--
Bonny ve arkadaşları, 2018 [56]	Tek anahtarlı Jerk sistemi veiki kanatlı butterfly sistemi	RK4 algoritması	IQ-Math Sabit noktalı sayı standardı	Xilinx Zynq-7000 Xilinx Vivado	172,5
Dursun ve arkadaşları, 2018 [57]	Van Der Pol sistemi	Euler algoritması	32-bit IEEE-754-1985 kayan noktalı sayı standardı	Xilinx VIRTEX-6 chip ailesi, XC6VLX75T aygıtı, FF784 paketi	498,728
Tlelo-Cuautle ve arkadaşları, 2016 [58]	Multi-scroll kaotik osilatörü	Euler algoritması	IQ-Math Sabit noktalı sayı standardı	Xilinx XXC3S1000-5FT256 FPGA Spartan-3, Yazılım: ISE,Altera Cyclone IV GX, FPGA DE2i-150, Yazılım: Quartus II	66

Çizelge 1.2 (devam). Literatürdeki FPGA tabanlı kaotik osilatör tasarımı ve özellikleri.

Tuna ve arkadaşları, 2019	Lü-Chen	Heun algoritması	IQ-Math Sabit noktalı sayı standardı- 32-bit	Xilinx Virtex-5 XC6VLX75T-3FF484	464,688
Alçın ve arkadaşları, 2019 [54]	Pehlivan-uyaroğlu	Yapay Sinir Ağları	32-bit IEEE-754-1985 kayan noktalı sayı standardı	Xilinx Virtex-6 XC6VCX240 T	231,616
Tolba ve arkadaşları, 2017[59]	Liu sistemi	Grünwald-Letnikov metodu	-----	Xilinx Virtex-5 XC5VLX50T	137,561
Alçın ve arkadaşları, 2016 [60]	PU (Pehlivan-Uyaroglu) kaotik sistemi	YSA	32-bit IEEE-754-1985 kayan noktalı sayı standardı	Xilinx Virtex-6 XC6VCX240 T	266,429
Tuna ve arkadaşları, 2015 [61]	Yeni bir kaotik sistem	Heun algoritması	32-bit IEEE-754-1985 kayan noktalı sayı standardı	Xilinx Virtex-6 XC6VCX75 T	390
Koyuncu ve arkadaşları, 2014 [62]	Pehlivan-Wei kaotik sistemi	Euler, Heun ve RK4 algoritmaları	32-bit IEEE-754-1985 kayan noktalı sayı standardı	Xilinx Virtex-6 XC6VCX75 T	463,688
Azzaz ve arkadaşları, 2013 [63]	3-B Hibritkaotik sistem. (Lorenz, Lü, Chen ve Liu-Chen)	Euler algoritması	IQ-Math Sabit noktalı sayı standardı- 32-bit	Xilinx Virtex-II XC2VP30FF G896	38,86
Merah ve arkadaşları, 2013 [64]	Lorenz	RK4 algoritması	32 bits (12Q20) IQ-Math Sabit noktalı sayı standardı	Xilinx Spartan-3	18
Koyuncu ve arkadaşları, 2013 [65]	Burke-Shaw	RK5-Butcher algoritması	32-bit IEEE-754-1985 kayan noktalı sayı standardı	Xilinx Virtex-6 XC6VCX75 T	373,094

Çizelge 1.2 (devam). Literatürdeki FPGA tabanlı kaotik osilatör tasarımı ve özellikleri.

De Micco ve arkadaşları, 2011 [66]	Lorenz	RK4 algoritması ¹	32-bit IEEE-754-1985 kayan noktalı sayı standardı	Altera Cyclone III EP3C120F7	1
Sadoudi ve arkadaşları, 2009, [67]	Chen	RK4 algoritması ¹	32-bit IEEE-754-1985 kayan noktalı sayı standardı	Xilinx Virtex-II XCV1000FG 456-4	22,85
Eroğlu ve arkadaşları, 2007 [68]	Lorenz, Chua Rössler, Linz ve Sprott	Simulink ve Xilinx Sistem Üretici	32-bit IEEE-754-1985 kayan noktalı sayı standardı	Xilinx Virtex-4 XC4VSX35-668	-----

Literatüre sunulan kaotik tabanlı çalışmaların temelinde; kaotik sinyali üreten ve diferansiyel denklemlerle ifade edilen kaotik osilatör yapısı yer alır. Diferansiyel denklemlerin çözümlenmeleri de bazı algoritmalar aracılığıyla yapılabilmektedir. Bu nümerik algoritmalar ise; Heun, Euler, Runge Kutta4, Runge Kutta5- Butcher ve Dormand Prince (DP)'dir. Yapılan literatür taramalarında kaotik osilatör tasarımı yapımında çoğunlukla kullanılan nümerik algoritmalar; Euler, Heun, RK4 ve RK5-Butcher nümerik algoritmaları olduğu gözlenmiştir. Fakat DP algoritması diğer nümerik algoritmalara göre daha yakın ve hassas çözüm üretmesine rağmen literatür taramalarında bu algoritmaya rastlanmamıştır.

Rajagopal ve arkadaşları tarafından yapılan çalışmada, RK5-Butcher Algoritması, 32bit IEEE-754 kayan noktalı standardı kullanılarak Xilinx Virtex-6 XC6VLX240T FPGA çipi üzerinde yeni bir 3D kaotik Chameleon sistemi uygulaması gerçekleştirilmiştir [69]. Alçın ve arkadaşlarının yaptıkları çalışmada ANN yapısı kullanılarak 32-bit IEEE-754 kayan noktalı standardı ile Xilinx Virtex-6 FPGA çipi üzerinde Pehlivan Uyaroğlu Kaotik Sistemi FPGA'de donanımsal olarak tasarlanmıştır [8]. Yapılan diğer bir çalışmada Tuna ve arkadaşları tarafından Heun algoritması ile kayan noktalı sayı formatı kullanılarak Xilinx Virtex-6 FPGA çipi üzerinde tek denge noktasına sahip yeni bir 3 boyutlu kaotik sistemin tasarımı ve uygulanması gerçekleştirilmiştir [1]. Bir diğer çalışmada Azzaz ve arkadaşları tarafından 32-bit IQ-Math formatına uygun Xilinx firmasının Virtex-II ailesi FPGA çipi ile 3-Bhibrit kaotik sistemi modellenmiş ve sistem kullanılarak bir uygulama gerçekleştirilmiştir [63]. Lai ve arkadaşlarının yaptıkları çalışmada dördüncü dereceden Runge Kutta algoritması kullanılarak, 32-bit IEEE-754

numara standardı ile Xilinx Kintex-7 FPGA çipi ile multi-butterfly kaotic çekiciler (attractors) tasarımı yapılmış ve osilatör kullanılarak mühendislik uygulamaları gerçekleştirilmiştir [29]. Tlelo-Cuautle ve arkadaşlarının yaptıkları çalışmada 32-bit sabit noktalı sayı standardı kullanılarak Altera Cyclone IV FPGA çipi üzerinde kaotik sistem modellenmiştir [58]. Sadoudi ve arkadaşları tarafından RK-4 algoritması ile 32-bit IEEE-754 sayı standardı kullanılarak Xilinx Virtex-II FPGA çipi üzerinde Chen Kaotik Sistemi gerçek zamanlı olarak gerçekleştirilmiştir [67]. Rajagopal ve arkadaşları MATLAB Xilinx Sistem generator kullanarak Xilinx Virtex-7 XC7-VX980tffg FPGA çipi üzerinde 3D Zaman gecikmeli Chameleon kaotik sistemini modellemişlerdir [55]. Pano-Azucena ve arkadaşlarının yaptıkları çalışmada, Grünwald–Letnikov methodu ve 32 bit sabit noktalı sayı standardıyla Altera’ya ait Cyclone IV GX FPGA DE2i-150, DAS1612 dijital analog dönüştürücü platformu kullanılarak kesirli dereceden (fractional-order) kaotik osilatörü tasarlanmıştır [53]. Tuna ve arkadaşları Heun nümerik algoritmasını kullanarak 32 bit IQ-Math sabit noktalı sayı standardını kullanarak Xilinx Virtex-6 FPGA çipi üzerinde Lü-Chen Kaotik osilatörünü uygulamışlardır [54]. Rajagopal ve arkadaşları Adomian Decomposition metodu kullanarak kesirli mertebeden memristor ve memkapasitör bileşenleri olan yeni bir kaotik sistem önermişlerdir [55]. Bonny ve arkadaşları da RK4 algoritmasıyla IQ-Math sabit noktalı sayı standardını, Xilinx Zynq-7000 serisini ve Xilinx Vivado’yu kullanarak tek anahtarlı Jerk sistemi ve iki kanatlı kelebek sistemini modüler bir FPGA platformunda gerçekleştirmişlerdir [56]. Dursun ve arkadaşı tarafından 32bit IEEE 754-1985 sayı standardı ve Euler nümerik algoritmasıyla Xilinx Virtex-6 çip ailesinden XC6VLX75T aygıtını FF784 paketini kullanarak Van Der Pol osilatörü tasarlanmıştır [57]. Tlelo-Cuautle ve arkadaşları tarafından Euler nümerik algoritması ve Xilinx firmasına ait XC3S1000-5FT256 FPGA Spartan-3, ISE yazılımı ve Altera firmasına ait Cyclone IV GX FPGA DE2i-150 modeli ile QuartusII yazılımı kullanılarak çok kaydırmalı kaotik osilatörler gerçekleştirmişlerdir [58].

Sunulan bu çalışmada İkinci Bölüm’de kaotik sistemler ve Van der Pol osilatörü, FPGA çipleri ve nümerik algoritmalar hakkında kısaca bilgiler verilecektir. Üçüncü Bölüm’de FPGA-tabanlı Van der Pol osilatör ünitesi tasarımı sunulacaktır. Ayrıca tasarımın test edilmesinden elde edilen FPGA yonga istatistikleri ve simülasyon sonuçları verilecektir. Son Bölüm’de ise çalışmadan elde edilen sonuçlar yorumlanacaktır.

2. GENEL BİLGİLER

Doğrusallık belirli sınırlar arasında geçerli olabildiği gibi tüm Dünya’da bütün sistemler doğrusal olmayan (nonlinear) bir düzendedir. Bu nonlinear sistemler, lineer olmayan sistemlerin birbirleriyle bağlantılarını inceleyen, bu sistemin prototipini oluşturmaya çalışan bilim dalıdır. Günümüzde nonlinear sistemler üzerine birçok araştırma yapılmıştır. Bunların en başta gelen örnekleri de kaostilimi diğler bir deyişle kaotik sistemler olarak bilinmektedir.

Kaotik sistemler; başlangıç koşullarına duyarlı, karmaşık ve dağınık görünümündür ve deterministik doğrusal olmayan zamanla değışen sistemlerde ortaya çıkarlar [72].

Kaotik ve Kaos sistemi kendine özgü bir iç düzeni vardır, her ne kadar karışık bir düzene sahip olduğunu anımsatsa bile yapısı farklıdır. Bu düzen içerisinde belirgin özellikler mevcuttur. Bu özellikler arasında en önemlileri ise; başlangıç şartlarına bağılı olmaları, sonsuz sayıda periyodik olmayan davranışlara sahip olması, sistemde var olan parametre değıerlerine kaotik sisteme fazlasıyla bağlanması olarak sayılabilir. Ayrıca giriş referans değıerlerinde ya da sistem parametrelerinde değıerlerinde yapılan ufak değışiklikler sistem çıkışının tahmin edilemeyecek boyutlarda değıişimlere sebep olabilmektedir. Bu sebeplerden dolayı Kaotik sistemler bilimin her alanında özellikle de mühendisliğin birçok alanında önemli yer teşkil etmektedir. Bu uygulamalar arasında da başta gelen ana düzenek kaotik işareti üreten kaos sinyal üreteçleridir. Bu üreteçler analog ve sayısal tabanlı 2 farklı donanımsal yapıdan oluşturulabilmektedir. Kullanım ömrü ve sıcaklık gibi dış faktörlere bağılı olarak değıerleri değışen sistemler; analog kaotik üreteçlerdir. Bu dezavantajdan dolayı sayısal tabanlı kaotik üreteçler daha avantajlıdır. Sayısal tabanlı kaotik üreteçler Digital Signal Processors (DSPs) [26], Application Specific Integrated Circuits (ASICs) [27], microcontroller (μC) [28] ve FPGAs [29] gibi farklı donanımsal platformlarda kullanılmaktadırlar. Literatürde yukarıda verilen sayısal platformlar kullanılarak birçok çalışmalar gerçekleştirilmiştir. Sunulan bu çalışmalarda FPGA çiplerinin paralel sinyal işleme özelliklerinden dolayı diğler sayısal platformlara göre daha öne çıktığı görülmektedir. Bu nedenle FPGA-tabanlı kaotik osilatör tasarımları diğler platformlara göre literatürde daha fazla yer almaktadır.

2.1. FPGA TEKNOLOJİSİ VE KULLANIMI

Sayısal devre tasarımında uzun süredir programlanabilirmantık cihazları (Programmable Logic Devices: PLD) ciddi kolaylıklar sağlamaktadır. Geçmişten bu yana süregelen bir yarış içerisinde olan PLD alanında, 1970'lerde PROM (Programmable Random Memory) ve 1980'lerde TTL tümleşik devre olarak adlandırılan PAL devreleri yer alır. Süregelen zaman zarfında tümdevrelerin kapı eşdeğerlikleri binlere kadar yükselmiş ve merkezi işlem birimlerinde de bu rakam daha da yüksek hale gelmiştir. FPGA (Field Programmable Gate Array)'de Sahada Programlanabilir Kapı Dizisi olarak PLD teknolojisinin günümüzdeki en önemli uygulamalarından birisidir. Bazı üretici firmaların "Do it Yourself Processor" sloganıyla ünlendirdiği ürün sayesinde günlük yaşantımızda, eğitim ve iş hayatında da kişinin kendi işlemcisini daha kolay bir şekilde yeniden meydana getirmesini sağlamıştır. FPGA'in bu kadar ünlenmesinin sebebi de daha önce geliştirilmiş olan PLD'lerden kapasite ve hız olarak daha yüksek performans sağlamasıdır [73]. FPGA çiplerinin gelişim sürecine bakıldığında programlanabilir mantık elemanlarına dayanmaktadır [74]. Bu gelişimi daha iyi anlayabilmek ve aktif olarak kullanabilmek adına PLU, FPGA'in PLU'ya üstünlüklerini ve FPGA içyapısını bilmek gereklidir.

2.2. PROGRAMLANABİLİR MANTIK ELEMANLARI

Programlanabilir mantık aygıtları (Programmable Logic Device (PLD)); yeniden programlanabilir elektronik bir cihaz oluşturmak için yeniden konfigüre edilmiş sayısal devrelerdir. Yapı olarak Lojik kapılardan ve Flip – Flop'lardan meydana gelirler. Fabrikasyon üretim sırasında PLD'ler belirli bir göreve tanımlanmamışlardır. PLD'ler bir devrede kullanılmadan ne iş için kullanılacaksa önce özel bir program kullanılarak yeniden programlanmalıdırlar [74].

Günümüzde bilinen Programlanabilir Lojik aygıtları sınıflandırmaları aşağıdaki gibidir.

- SPLD – Simple Programmable Logic Devices (Basit PLD)
- CPLD – Complex Programmable Logic Devices (Karmaşık PLD)
- FPGA – Field Programmable Gate Array (Alan Programlanabilir Kapı Dizileri)

2.2.1. SPLD - Basit Programlanabilir Mantık Devreleri

SPLD'ler yapı olarak içerisinde az miktarda mantık hücresi bulundurlar. Dolayısıyla da SPLD ünitesinde 4 ile 22 arasında programlanabilir hücre bulunmaktadır. Genel olarak da üç ana kısımdan meydana gelir [75]. Bunlar;

1. PROM – Programmable Read Only Memory – Programlanabilir Saltokunur Bellek

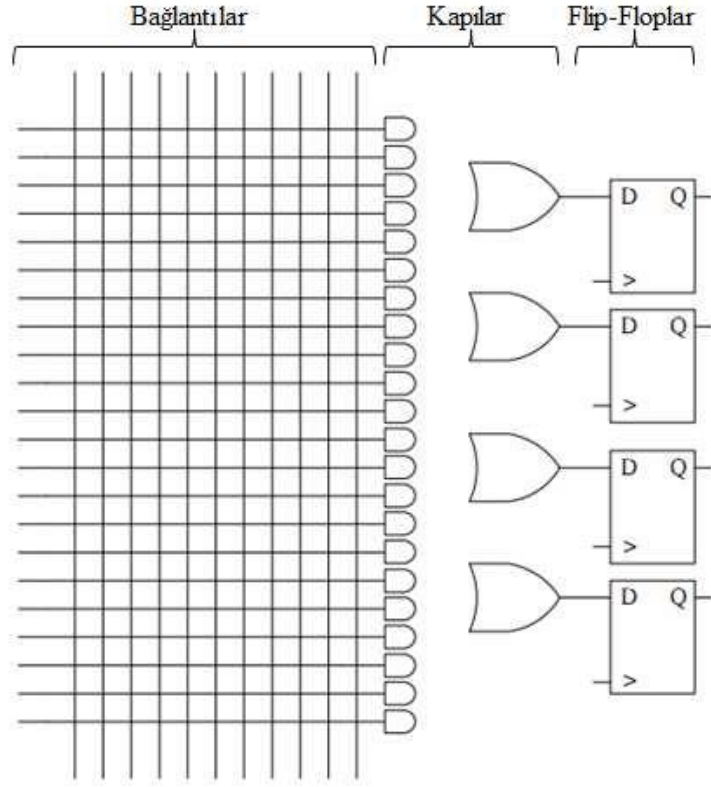
Read Only Memory (ROM) sadece okunabilir bellek adı verilen devre olup, PROM ise Programlanabilir ROM cihazı olarak adlandırılır. ROM ve PROM'lar aynı özellikleri taşımalarına rağmen PROM imalat esnasında programlanma zorunluluğu yoktur. PROM bir kez programlandığında üzerinde defalarca problemsiz olarak okuma işlemi gerçekleştirilebilir. EPROM ve Electrically Erasable Programmable Read Only Memory (EEPROM) olmak üzere iki çeşidi vardır. Bunlardan ilki olan Erasable Programmable Read Only Memory (EPROM) Silinip Programlanabilen Yalnızca Salt Okunur Bellek olup elektrikle programlanabilen ve ultraviyole ışıkla silinebilen bellek çeşidi olarak adlandırılırken; Electrically Erasable Programmable Read Only Memory (EEPROM) Elektriksel Olarak Silinip Programlanabilen Salt Okunur Bellek yani elektrikle yazılıp elektrik ile silinebilen donanım parçası olarak bilinir.

2. PAL – Programmable Array Logic – Programlanabilir Dizi Mantığı

PAL yapı olarak programlanabilen AND kapı dizisi ve OR kapılarından meydana gelmektedirler [76].

3. GAL – Generic Array Logic Lattice – Genel Dizi Mantığı

Bu cihaz; PAL ile aynı mantıksal özelliklere sahip olmasına rağmen silinebilen ve yeniden programlanabilme özelliği sayesinde tasarımın prototip aşamasında mantıkta bulunan bir hata; PAL programcısı kullanarak veya destek çiplerinde devre içi programlama tekniğiyle yeniden programlama ile düzeltilebilir. Aşağıda Şekil 2.1'de SPLD mimari yapısı verilmiştir [77].

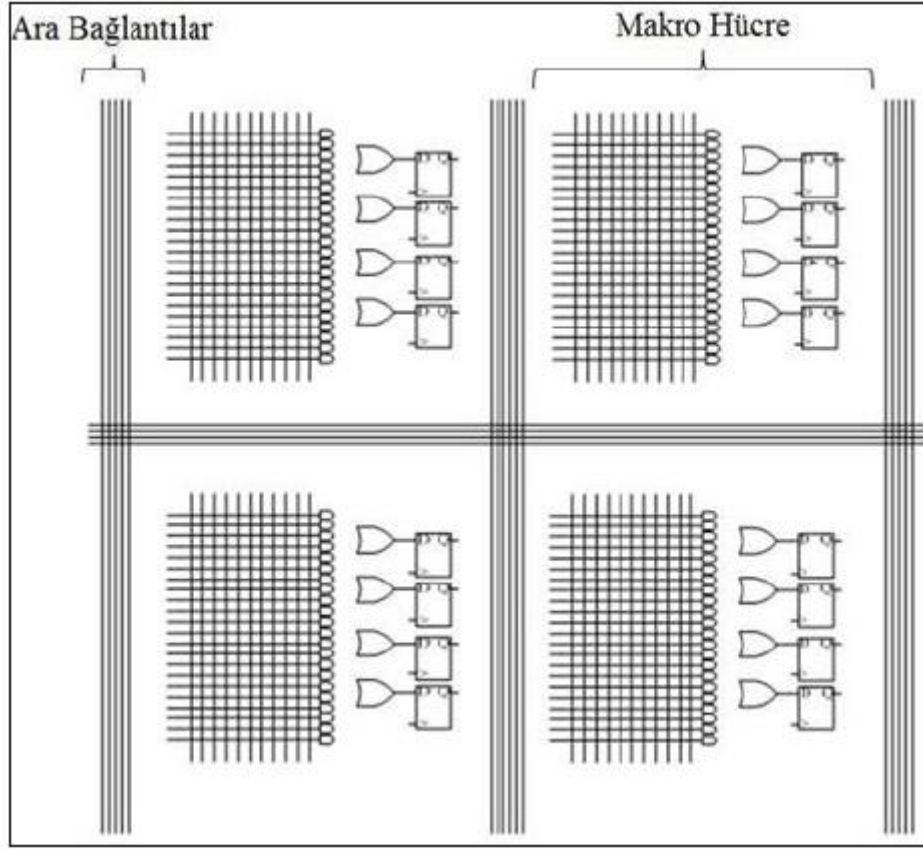


Şekil 2.1. SPLD mimari yapısı.

2.2.2. CPLD - Karmaşık Programlanabilir Mantık Devreleri

CPLD; basit programlanabilen mantık devreleri (SPLD)'de tasarımda zorlanıldığında ve tasarımın yetmediği durumlarda birden fazla SPLD yapılarının birleştirilmesiyle ve bu yapılar arasında haberleşmenin sağlanmasıyla meydana getirilebilmektedir. Literatür incelendiğinde CPLD'ler teknolojilerine göre 4 farklı sınıfta toplanabilirler. Şekil 2.2'de CPLD 'nin mimari yapısı verilmiştir [75].

1. EPLD (Erasable Programmable Logic Device) – Silinebilir Programlanabilir Lojik Aygıt
2. PEEL (Programmable Electrically-Erasable Logics) – Programlanabilir Elektriksel Silinebilir Lojik
3. EEPLD (Electrically-Erasable Programmable Logic Device) – Elektriksel Silinebilir Programlanabilir Lojik Aygıt
4. MAX (Multiple Array matrix, Altera) – Çoklu Dizi Matrisi



Şekil 2.2. CPLD mimari yapısı [75].

2.2.3. FPGA Çipleri

Field Programmable Gate Array olarak bilinen ve Alan Programlanabilir Kapı Dizileri olarak adlandırılan FPGA, tekrar tekrar elektriksel olarak programlanabilen tümleşik devre elemanıdır. Tasarımcının sahada yeniden programlayabileceği FPGA çipleri, tümleşik devrelerdir.

Günümüzde FPGA çipleri sahip olduğu yüksek hız ve kapasite avantajı sayesinde farklı alanlarda kullanımının daha da geliştirilmesiyle yapı itibarıyla daha da karmaşık bir yapıya bürünmüştür. Bu çipler genellikle SoC denilen System on a Chip yani “Çip üzerinde Sistem” olarak isimlendirilerek gerektiğinde diğer çiplerle birlikte büyük bir sistemin parçası olarak çalışabilmektedir [78].

Genel amaçlı fakat sisteme has olarak üretilen FPGA çipleri gerçek zamanlı çalışma yeteneğine sahiptirler. FPGA çipleri kişisel bilgisayarlara oranla daha düşük çalışma frekansına sahip olmalarına rağmen paralel çalışma ve kendi sistemine özgü tasarım avantajından dolayı kişisel bilgisayardan çok daha hızlı bir şekilde işlemleri gerçekleştirebilmektedir [79].

FPGA çiplerinin birçok avantajı vardır. Bunlardan bir diğeri de yüksek hız kazancıdır. Bunu da çip içerisindeki bütün bir sistem olarak tasarlanan yapının birden fazla kopyasının çalıştırmasıyla sağlar. Bu avantaja ek olarak bir de IP-core denilen Intellectual Properties-core oluşturularak modüller sayısal devre tasarımı ile hazır kütüphane dosyaları kullanılarak tasarım sürecinin hızlanmasını sağlamasıdır [80].

FPGA’larda hücreler arası iletişim mimarisi, önceki teknolojilerden çok üstündür. Paralel işlem yapabilme yeteneği sayesinde hız önünden FPGA, diğer donanım birimleri olan mikroişlemciler veya DSP’ye göre ön planda çıkmaktadır. Bunun yanı sıra maliyetinin düşük olması ve tekrar tekrar da programlanabilmesi özelliği sayesinde ilk örnek haline gelmede kullanıcıların tercihi biraz daha fazla FPGA’den yana olmaktadır.

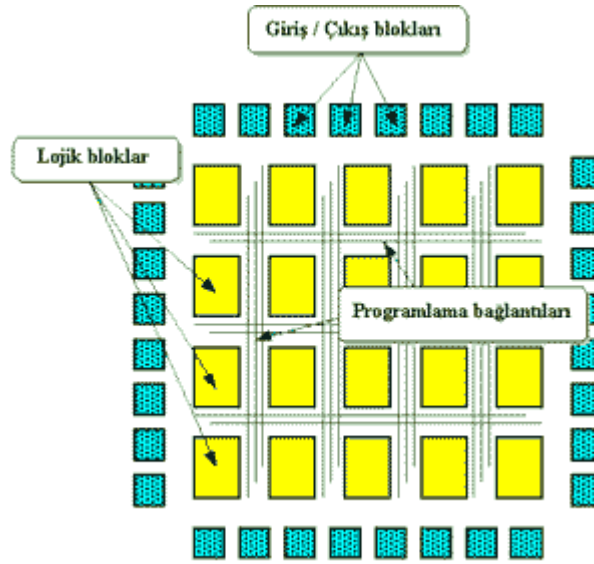
FPGA çipi ilk olarak 1985 yılında Xilinx firması tarafından üretilmiştir. Ardından farklı firmalardan farklı özelliklere sahip farklı isimlerde FPGA ‘lar üretilmeye başlanmıştır. Bu firmalar; Actel, Altera, Atmel, Chip Express, Clear Logic, Cypress, DynaChip, Fast Analog Soltions Ltd, Gatefield, HenmerGres, Lattice, Lucent Technnologies, Military Aerospace Applications of Programmable Devices and Technologies Conference (MAPLD), Motorola, Orbit, Quicklogic, QuickTurn Vantis gibi firmalardır [73]. Ayrıca Xilinx firması ürettiğiçiplere Spartan, Virtex, Kintex diye adlandırmışken, diğer bir üretici olan Altera firması imal ettiği çiplere Stratix, Cyclone gibi adlandırmıştır [80].

Üretilen bu FPGA’lar için demo kartları üreten firmalar ve kartları ise, ACS/Insight’s Demo Boards, AEE Engenharia Eletronica: AEE3298 – FPGA Evaluation Borad, Alpha Data Parllel Systems, Associated Professional Systems, Baldwin Technologies – Actel A1280CQFP Adapter, Bright Stars’ipEng’ne 1, Digitals PCI Pamette, NEF Design’s FPGA Downloader, NOVA Engineering’s Constellation FLEX 10K, Nallatech, Sun MicroSystem’s Laboratories, Reconfigurable Interface Cards, Virtual Computer Corporation firmalarıdır.

FPGA tasarım ve programlamasında kullanılan yardımcı programlar ve üretici firmaları ise, Accolade Design Automation, Acugen Software, Aldec, Alternative System Concepts (ACS), Aptix, Associated Professional Systems (APS), C2SynVHDL, Cadence Design Systems, Capilano Computing, CompiLogic Corporation, Data I/O, Design Acceleration (DAI), Embedded Solutions, Enable++ Development Environment (EDE), Envision Technologies, Escalade, Exemplar, Giga Operations, Handler, Ikos Systems, Institute for Computer Systems, InterHDL, MINC, Mentor Graphics,

MicroSim, Model Technology, Morphlogic, Olympus Synthesis System, Orcad, Phase 3, Pilkington APR, Protel, Right Track CAD, Saros Technology, Snaketech, Synario, Synopsys, Tanner Research, Translogic, Transmogripher C, VeriBest, ViewLogic, X Engineering Software Systems (XESS), Zeelan Technology ve Zycad v.b.'dir [73].

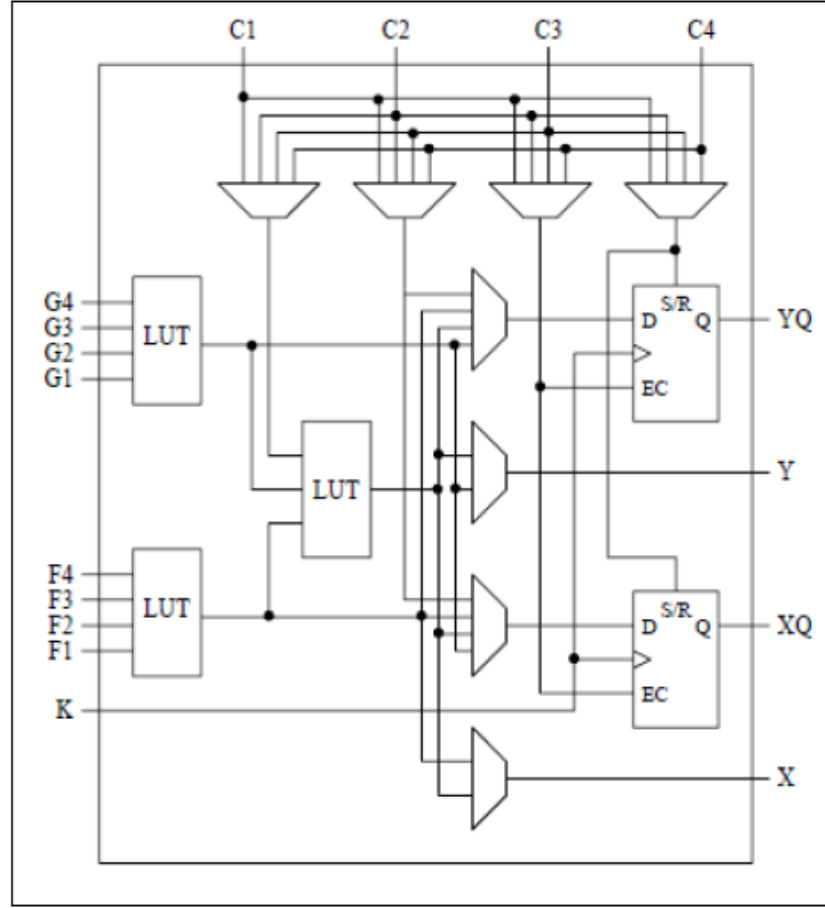
Şekil 2.3'te genel bir FPGA yapısı gösterilmiştir. Bu yapı içerisinde boole fonksiyonları ile oluşturulan konfigüre edilebilir mantıksal bloklar olan Configurable Logic Blok (CLB), giriş-çıkış blokları (I/O Blocks) ve programlama bağlantıları adı verilen ara bağlantılardan (interconnect) oluşur.



Şekil 2.3. Genel FPGA yapısı.

1. Konfigüre edilebilir Mantıksal Bloklar (CLB), mantıksal yani Boole fonksiyonların oluşturulabildiği Look-up table (LUT)'dan oluşan bir devredir. Look-up Table yapı olarak bir bitlik bilgilerin saklanabildiği Flip-Flop'lar bilgi akışını yönlendiren mux denilen çoklayıcı tabanlı yapı ile toplayıcıların tanımlanmasında kullanılan "elde-zinciri" gibi çeşitli elemanlardan oluşurlar. Oluşturulmak istenen mantıksal devreler çeşitli yazılım araçları sayesinde bir CLB'ye veya parçalara bölünerek birden fazla CLB'ye otomatik olarak uygulanır.

Aşağıda Şekil 2.4'te örnek bir FPGA çipinin CLB yapısı gösterilmiştir.

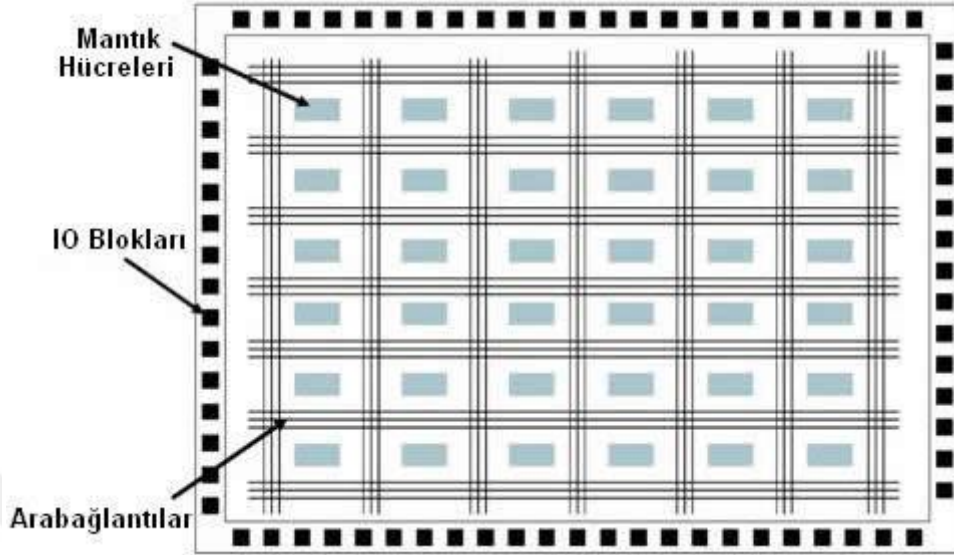


Şekil 2.4. Örnek bir FPGA çipinin CLB yapısı.

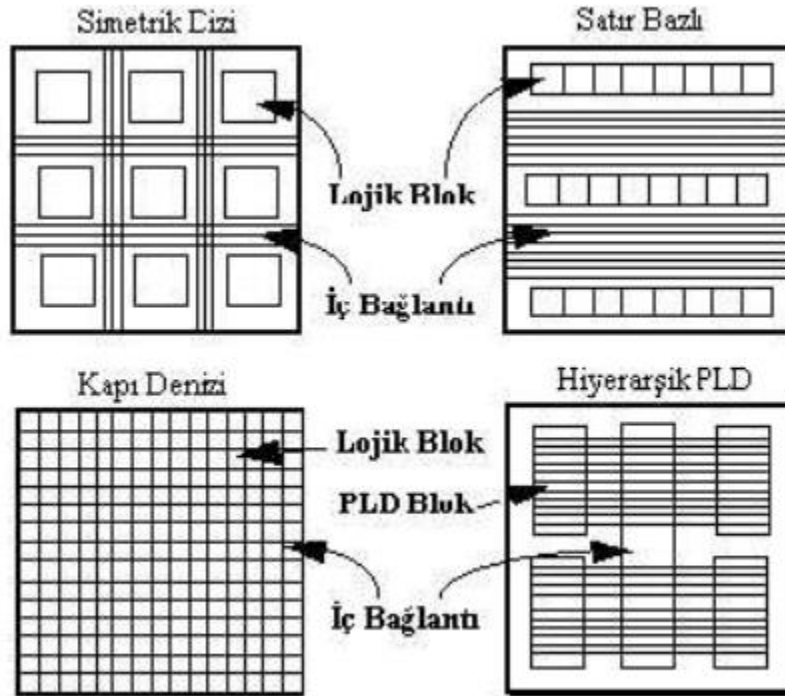
2. Giriş Çıkış Blokları (Input/Output Blocks (I/OB)), FPGA çiplerinin programlanabilir giriş/çıkış uçlarıdır. Bu blokların görevi; çip ile dış dünya arasındaki veri iletimini sağlamaktır. Bloкта yer alan pinler; tasarımcının istek ve ihtiyacına göre giriş, çıkış ya da çift yönlü olarak yani hem giriş hem çıkış pini olacak şekilde programlanabilir. Ek olarak güç pinleri, saat (clock) pinleri, konfigürasyon pinleri ve kullanıcı pinleri gibi gerekli olan FPGA çipi üstünde bir çok pin bulunmaktadır [77, 81]. Paket türüne göre de FPGA çipindeki giriş/çıkış pin sayısı yani I/OB sayısı 1000'li sayılara ulaşabilmektedir [16].

3. Ara Bağlantılar (Interconnections), yapılandırılabilir mantıksal bloklar arasında ve yapılandırılabilir mantıksal bloklar ile giriş/çıkış blokları arasında bağlantıları yapılandırmada kullanılırlar. Programlanabilir bir yapıya sahiptirler. Ara bağlantılar; haberleşme görevini yerine getirmekle yükümlüdürler. Bu nedenle çip üretimi; sayısal sinyal işleme için tasarlanmış özel bloklar, RAM hafıza blokları, hatta işlemci çekirdek üniteleri içerir [16].

Şekil 2.5'te FPGA çiplerinin genel yapısı görülmektedir.



Şekil 2.5. Bir FPGA çipinin genel yapısı [82].



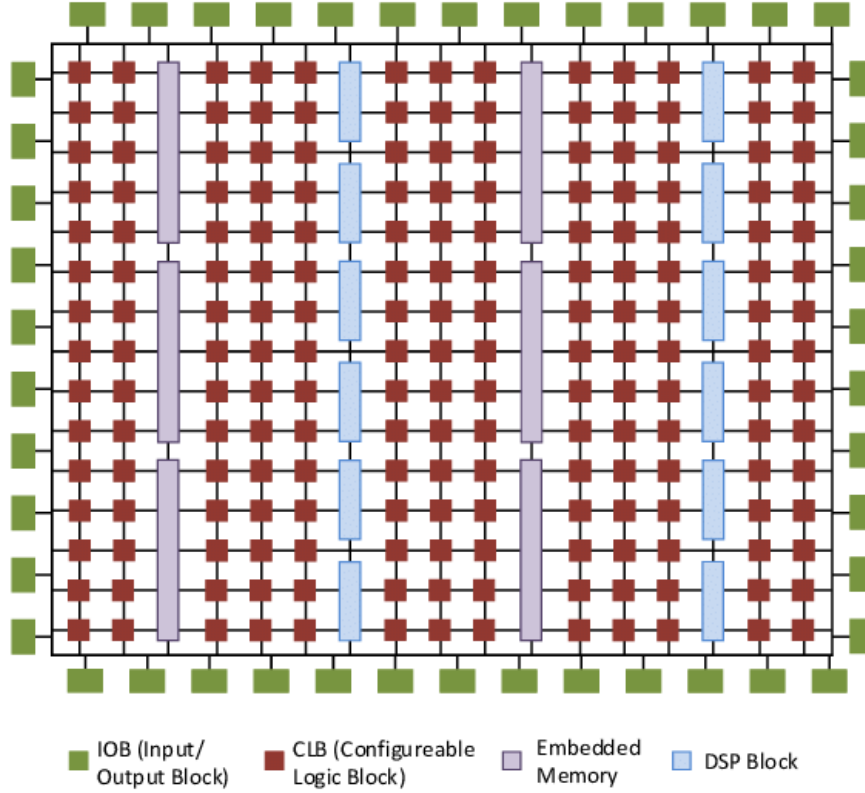
Şekil 2.6. FPGA çipinin bağlantı çeşitleri [74].

2.2.3.1 Xilinx Firması Tarafından Üretilen FPGA'ların Mimarisi

Şekil 2.7'de Xilinx firmasının ürettiği Virtex-6 FPGA kartı görülmektedir.

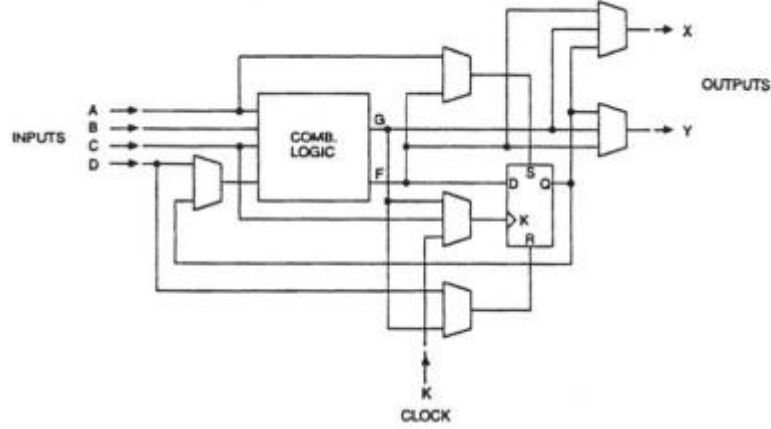


Şekil 2.7. Xilinx firmasının ürettiği Virtex-6 FPGA kartı.



Şekil 2.8. Xilinx FPGA'ların genel yapısı [83].

Şekil 2.8’de üretimi Xilinx firması tarafından yapılmış olan Look-Up-Table-tabanlı FPGA’ların genel yapısı verilmiştir. Bu yapıdan da CLB denilen Yapılandırılabilir Mantık Blok dizileri ile satır ve sütunlar arasındaki bağlantı kanallarından oluşan bir yapı anlaşılmaktadır.



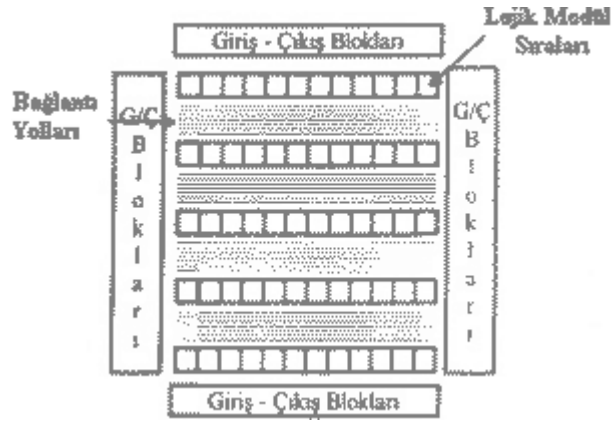
Şekil 2.9. XC2000 CLB yapısı [84].

FPGA’ların çeşitli modelleri bulunmaktadır. Bunlar; XC2000, XC3000, XC4000, XC5200, XCS olarak örneklenebilir. Şekil 2.8’de yapısı görülen XC2000 CLB ile 4 değişkenli herhangi bir boole fonksiyonu veya toplam değişken sayısı en fazla 4 olan 3 değişkenli herhangi iki Boole fonksiyonu gerçekleştirilebilir. CLB’nin çıkışlarının ikisi de doğrudan veya biri flip-flop üzerinden olabilir.

FPGA serisi olan XC3000 serisi, XC2000 serisinin geliştirilmiş şeklidir. Bu FPGA’lar ile 5 değişkenli herhangi bir Boole fonksiyonunu veya toplam değişken sayısı 5 olan 4 değişkenli iki Boole fonksiyonu gerçekleştirilebilir. XC3000 serisinin yapılandırılabilir mantık blokları (CLB’leri) ikisi de kombinezonsal ya da ardışıl olabilecek iki çıkışa sahiptir.

XC7000 serisini her CLB’si ile beş değişkenli herhangi bir fonksiyon, değişkenleri birbirinden farklı 4 değişkenli herhangi iki fonksiyon ve 9 değişkenliye kadar bazı fonksiyonlar gerçekleştirilebilir [73].

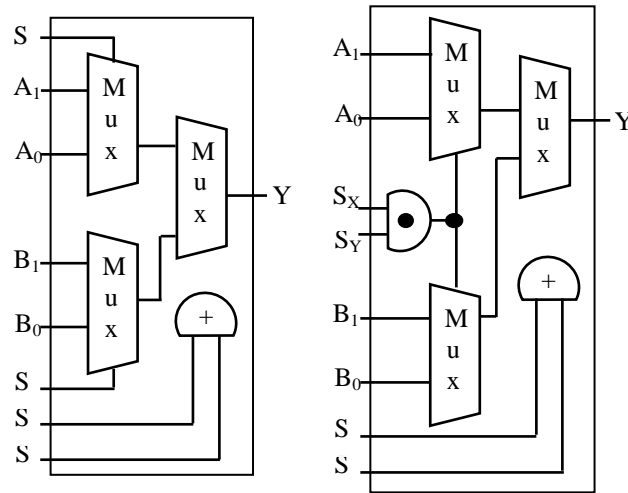
Actel firması tarafından üretilen FPGA'ların mimari yapısı aşağıda Şekil 2.10'da gösterilmiştir.



Şekil 2.10. Actel FPGA'ların genel yapısı.

Şekil 2.10'da MUX tabanlı Actel firması tarafından imal edilmiş olan FPGA'ların genel yapısı görülmektedir. Şekilden de anlaşılacağı gibi FPGA'lar "Lojik Modül" adı verilen programlanabilir bloklar ile aralarındaki yatay bağlantı kanallarından oluşmaktadır.

Bu FPGA'ların Act-1, Act-2 gibi ilk üretilen 2 modellerinin içyapıları Şekil 2.11'de gösterilmektedir. Act-1 lojik modülü ile 2 değişkenli herhangi bir fonksiyon, 3 değişkenli pek çok fonksiyon ve 4 değişkenli bazı fonksiyonlar gerçekleştirilebilir.



Şekil 2.11. Act-1 ve Act-2 lojik modülleri.

2.3. VAN DER POL SİSTEMİ

Kaos terimi ilk olarak M.I.T. bilimcisi Edward Lorentz'in 1963 yılında hava tahmini yapmak amacı ile oluşturduğu matematiksel meteorolojik modelin sonuçlarını yuvarlamak istemiş ancak istediği sonuçları alamamıştır. Bu sayede kaos bilim alanının ilk temelleri atılmıştır. Kaotik sistemler veya kaos tanım olarak incelendiğinde sistem parametreleri ile sistem başlangıç koşullarına oldukça hassas duyarlı, deterministik, non-lineer ve non-periyodik dinamik sistemler olarak ifade edilebilir [11].

Kaotik sistemler adi diferansiyel denklemler kullanılarak ifade edilmektedirler. Literatürde ayrık-zamanlı tek değişkenli kaotik sistemler bulunmaktadır. Sürekli-zamanlı kaotik sistemler ise içerisinde en az iki bağımlı değişken bulunan diferansiyel bir denklem takımı ile ifade edilmektedir. Eşitlik 2.1'de Van der Pol sistemi için ikinci dereceden diferansiyel denklemi verilmiştir [85].

$$\frac{d^2v}{dt^2} - \mu(1 - v^2) \frac{dv}{dt} + v = 0 \quad (2.1)$$

Van der Pol osilatörü birinci dereceden diferansiyel denklem kullanılarak Eşitlik 2.2'de verildiği gibi ifade edilebilmektedir.

$$\begin{aligned} \frac{dv}{dt} &= w \\ \frac{dw}{dt} &= \mu(1 - v^2)w - v \end{aligned} \quad (2.2)$$

Van der Pol osilatörü belirli şartlar altında kaotik osilatör karakteristiği göstermektedir. Van der Pol sisteminin kaotik özellik sergileyebilmesi için bu çalışmada sistem parametresi $\mu=0,5$, sistemin başlangıç şartları $v_0=1,0$ ve $w_0=-0,97$ olarak alınmıştır. Ardından sistemin zaman serilerinin elde edilebilmesi için aşağıda verilen kod yazılmıştır. Verilen kod sistemi iki bölümden oluşmaktadır. İlk bölümde algoritma çalıştığında başlangıç şartları ve adım değerlerini alarak ikinci bölüm olan Vander Pol fonksiyonuna değerleri göndermektedir. Burada gelen değerlere göre ilk fonksiyon değeri elde edilmekte ve ana-algoritmaya gönderilmektedir. Burada gelen değer h adım değeri ile çarpılmaktadır. Ardından başlangıç değeri ile toplanarak algoritma ilk iterasyon değerini elde etmektedir. Algoritmanın her iterasyon sonunda aldığı değerler ye değişkenine kaydedilmektedir. Bu aşamada ye değeri yk_1 ve yk_2 olmak üzere iki

değişkene aktarılmaktadır. Bu değişkenler Van der Pol sisteminin iki değişkenini ifade etmektedir.

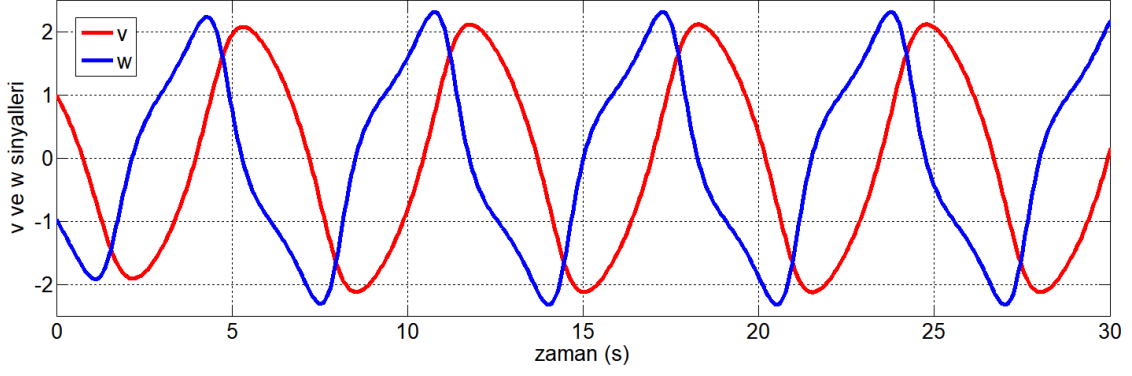
```
1 % Euler Metodu ile Van der Pol sistemi
2 - clc
3 - clear all
4 - h=0.05;
5 - time_son=500;
6 - ye=[];
7 - y0=[1,-0.97];
8 - for time=0:h:time_son
9 -     ye=[ye; y0];
10 -    yturev=vanderpol(time,y0);
11 -    yt1=y0+h*yturev';
12 -    y0=yt1;
13 - end
14 - time=0:h:time_son;
15 - yk1=ye(:,1);
16 - yk2=ye(:,2);
17 - time=time';
```

Şekil 2.12. Van der Pol osilatörü için tasarlanan ana-algoritma.

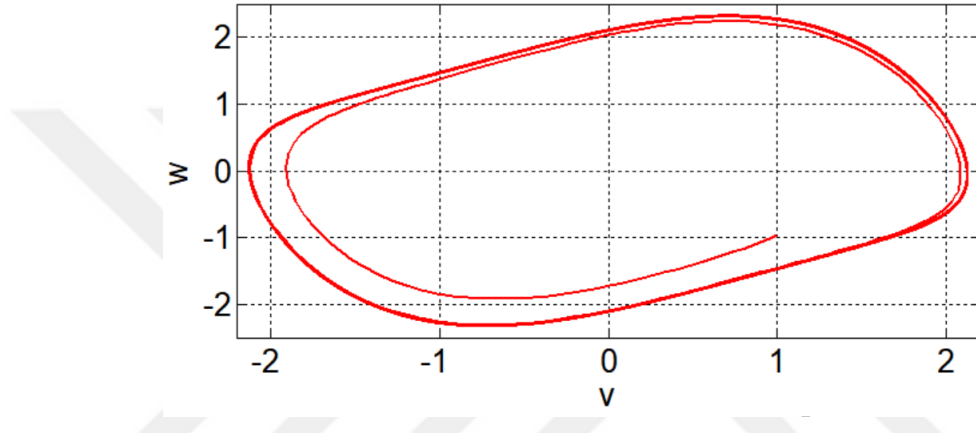
```
1
2 - function yp=vanderpol(t,y)
3 -     M=0.5;
4 -     yp=[y(2);
5 -         M*(1-y(1)^2)*y(2)-y(1)];
```

Şekil 2.13. Van der Pol osilatörü için tasarlanan fonksiyon algoritması.

Van der Pol sistemi yukarıda verilen algoritma kullanılarak adım değeri 0,05 ve iterasyon son değeri time_son=500 olmak üzere çalıştırılmıştır. Algoritma çalıştırıldığında elde edilen ye değeri yk1 ve yk2 olmak üzere iki değişkene atanmıştır. yk1 ve yk2 değişkenleri Van der Pol sisteminin v ve w bağımlı değişkenlerini ifade etmektedir. Kaotik sistemlerin kaotik olup olmadığının incelenebilmesi amacı ile literatürde farklı yöntemler kullanılmaktadır. Bu çalışmada literatürde çok sık bir şekilde kullanılan zaman serileri ve faz portreleri yöntemleri sunulmuştur. Elde edilen yk1 ve yk2 değerleri 30 s süre ile çizdirilmiştir. Bu işlemin ardından aşağıda Şekil 2.14'te verilen Van der Pol sistemine ait zaman serileri elde edilmiştir. Şekil 2.15'te verilen Van der Pol sisteminin v ve w sinyallerine ait faz portresi verilmiştir.



Şekil 2.14. Van der Pol osilatörü zaman serileri.



Şekil 2.15. Van der Pol osilatörü v-w faz portresi.

2.4. ALAN PROGRAMLANABİLİR KAPI DİZİLERİ

FPGA çipleri, sayısal bir sistem/devre tasarımları için tekrar tekrar programlanabilir özelliğe sahip tümleşik devre araçları olarak tanımlanabilir. Bu çipler, ilk silikon üretimi işleminin ardından tasarımcının istediği zaman çip içerisinde sistem tasarımının yapılabilmesine imkân sağlamaktadır [86]. Bu çipler genellikle System On a Chip (SoC) olarak da adlandırılmakta ve ihtiyaç olduğunda diğer donanımsal platformlar ile birlikte daha büyük tasarımların parçaları olarak çalışabilmektedirler [78]. Son yıllarda FPGA çipleri oldukça yüksek kapasite ve yüksek performansa sahip olmakla birlikte her geçen gün daha kompleks bir yapıya sahip olmaktadır. Genel amaçlı üretilen ve tasarıma özel programlanabilen FPGA çipleri yaklaşık olarak 1 GHz çalışma frekansına sahiptirler [36]. DSP, μ C ve PC gibi sequential çalışan platformlara göre FPGA çiplerinden parallel processing ve sisteme özgü tasarım gibi avantajları nedeni ile çok yüksek performans elde edilebilmektedir. Bundan dolayı kişisel bilgisayarlardan çok daha hızlı bir şekilde işlemleri gerçekleştirebilmektedirler. FPGA çiplerinin içerisinde

yapılan tasarımın birden fazla kopyası çip içerisinde çalıştırılabilmektedir. Ayrıca, standart bir bilgisayar ve buna bağlı bir ya da daha FPGA çipi bulunduran FPGA tabanlı Özel Bilgi işlem makinaları (FPGA based Custom Computing Machines) ile de tasarımlar gerçekleştirilerek yüksek performanslı platformlar elde edilebilmektedir. Belirtilen bu gibi avantajlardan dolayı, FPGA donanımları son yıllarda, motor kontrolünden endüstriyel görüntülemeye, uzay ve savunma sanayisinde kriptolu haberleşmeden elektronik harp uygulamalarına, tüketici elektroniğinde sayısal kameralardan uydu alıcılarına, biyomedikalde bilgisayarlı tomografiden ultrason görüntülemeye ve otomotiv endüstrisinde görüntü işlemeden araç içi bilgi sistemlerine kadar çok geniş bir yelpazede kullanılabilmektedir [87]. Günümüzde FPGA çipleri Xilinx, Altera, Atmel, SiliconBlue, Microsemi ve Lattice gibi birçok şirket tarafından üretilmektedir.

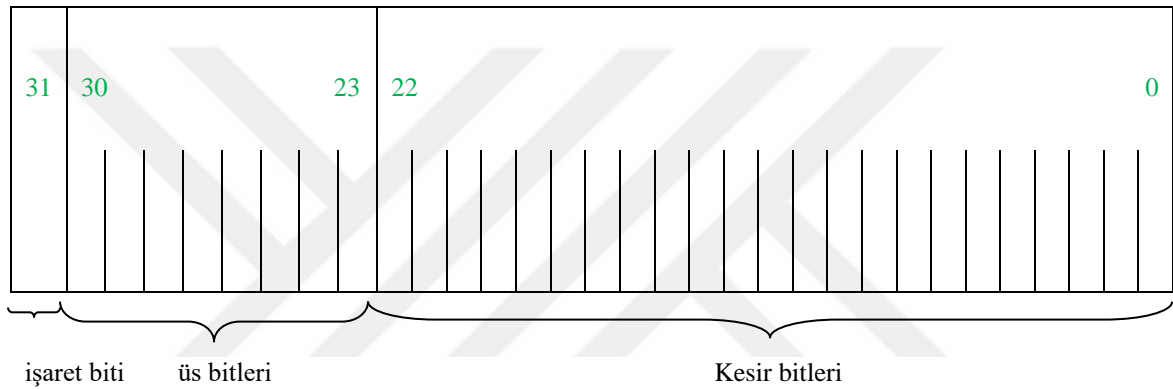
FPGA donanımları genel olarak giriş-çıkış blokları (I-O blocks), interconnection network ve Configurable Logic Block (CLB) olmak üzere üç bölümden meydana gelmektedir. Giriş-çıkış blokları (I-O blocks), interconnection network ve Configurable Logic Block (CLB) yapılarının tümü configure edilebilmektedir. Giriş-çıkış blokları giriş, çıkış veya hem giriş hem de çıkış portu olarak kullanılabilir. CLB birimleri çip türüne göre yapıları değişmekle birlikte genel olarak hafıza, mantıksal blok ve çarpıcı elemanlarından oluşmaktadır. Interconnection network ise giriş-çıkış portları ile CLB arasındaki bağlantıyı sağlayan programlanabilir yapılardır.

FPGA çiplerinde tasarımlar şematik ve Hardware Description Language (HDL) ile yapılmaktadır. VHDL ve Verilog dilleri en çok tercih edilen donanım tanımlama dilleridir.

2.5. IEEE 754-1985 KAYAN NOKTALI VE IQ-MATH SABİT NOKTALI SAYI STANDARTLARI

Gerçel sayıların ifade edilebilmesi için kullanılan yöntemlerin bir tanesi de kayan noktalı sayı standardıdır. Tüm sayıların gösterilmesi günümüzde donanımsal gerçeklemlerde mümkün olamamakla birlikte aslında sayıların sonsuza kadar gittiği bilinmektedir. Bu nedenle uygulamalarda bu değerler yaklaşık değerlerle temsil edilirler. Kayan noktalı sayı standardı da sayıların gerçeğe en yakın ve en yüksek değerlerde temsil edilmesini sağlayan, donanımsal kısıtlamaların en aza indirildiği sayı

standardıdır. IEEE 754-1975 standardı; tüm farklı donanımlara uyum sağlaması için geliştirilmiştir. 32 bit (Tek duyarlı) ve 64 bit (Çift duyarlı) olmak üzere iki farklı şekilde kayan noktalı sayı gösterimi bu standarda göre gerçekleştirilmektedir. Aşağıda Şekil 2.16'da tek duyarlı 32 bit IEEE 754-1985 kayan noktalı sayı standardı gösterilmiştir ve 3 kısımdan oluşmaktadır. Birinci kısım işaret biti olarak adlandırılan 31'inci bitin olduğu kısımdır. Eğer 31'inci bit '0' ise sayı pozitif, '1' ise sayı negatiftir. İkinci kısım ise; üstel kısmı belirtmek için kullanılan 8 bitlik üs bitleridir. Üçüncü ve son kısım da kayan noktalı sayı standardında sayının kesirli kısmına ait kesir bitleri olarak adlandırılan mantissa bitlerini işaret eder.



Şekil 2.16. 32-bit IEEE 754-1985 kayan noktalı sayı standardı gösterimi.

Şekil 2.16'da Kayan noktalı sayı standardının sayıların gösterilmesi Eşitlik 2.3'te tanımlanmıştır. Eşitlikte hesaplanacak olan 'v' onluk sayı değerini, 'j' kesir bit sayısını (tek duyarlı için j=23), kesirli bitleri b, üs bitlerin değerini de exp ile gösterilmiştir.

$$V = (-1)^{sign} \left[1 + \sum b_{23-i} 2^{-i} \right] \times 2^{(exp-127)} \quad (2.3)$$

iki kısımdan oluşmaktadır. Baş harflerinden de anlaşılacağı üzere I integer yani tam sayı terimini, Q ise fractional yani kesirli terimi ifade eder. Tasarımlarda ihtiyaç halinde IQ-Math sabit noktalı sayı standardı değerleri değiştirilebilmektedir. Fakat kayan noktalı sayı formatında böyle bir değişiklik söz konusu olamamaktadır. Çizelge 2.1'de IQ-Math fixed point number standardı gösterilmiştir. Kayan noktalı sayı standardında işaret biti 31'inci bit iken burada S olarak gösterilmiştir. Değer olarak kayan noktalı sayı standardında olduğu gibi S biti değeri '0' olduğunda sayı pozitif, '1' olduğunda ise sayı negatiftir. Yapılan hesaplamalarda ise sayının Integer denilen tam kısmı ikilik tabana dönüştürülerek işlem yapılır [88, 89].

Çizelge 2.1. IQ-Math Sabit noktalı sayı standardı gösterimi.

S	Integer (Tam Sayı Kısmı)	Q(Fractional=Kesirli Kısmı)
0/1	$I_n \dots \dots \dots I_2 I_1 I_0$	$Q_n \dots \dots \dots Q_2 Q_1 Q_0$

IQ-Math sabit noktalı sayı standardında kesirli kısım için farklı yöntemler geliştirilmiştir. Bunlardan ilki; kesirli kısım arzu edilen bit uzunluğu kadar 2 ile çarpılarak sonuç eğer 1'den küçük ise bit değeri yerine '0', eşitlik ya da büyüklük halinde ise '1' yazılarak bu değerden (1'den) çıkarılır [88]. 92,1245 sayısı bu yöntemle 8I-8Q sabit noktalı sayı formatına dönüştürme işlemi aşağıda gösterilmiştir.

Sayı : 92,1245

Sayı>0 nedeniyle S=0

I(Tam kısım) : 92

Q(kesirli kısım) :0,1245

Sayının Integer kısmı ikilik tabanda;

$$92=(1011100)_2$$

Q(kesirli kısım) olan 0,1245 ise;

1. $0,1245*2=0,249 \longrightarrow 0$
2. $0,249*2=0,498 \longrightarrow 0$
3. $0,498*2=0,996 \longrightarrow 0$
4. $0,996*2=1,992 \longrightarrow 1 \longrightarrow 1,992-1=0,992$
5. $0,992*2=1,984 \longrightarrow 1 \longrightarrow 1,984-1=0,984$
6. $0,984*2=1,968 \longrightarrow 1 \longrightarrow 1,968-1=0,968$
7. $0,968*2=1,936 \longrightarrow 1 \longrightarrow 1,936-1=0,936$
8. $0,936*2=1,872 \longrightarrow 1$

92,1245 sayısı 8I-8Q sabit noktalı sayı formatına birinci yöntem kullanılarak yapılan dönüşümde Çizelge 2.2'de verilen sonuçlar alınmıştır.

Çizelge 2.2. 92,1245 sayısının 8I-8Q sabit noktalı sayı formatına dönüşüm sonucu.

S	Integer (Tam Sayı Kısmı)	Q(Fractional=Kesirli Kısmı)
0/1	1011100	00011111

IQ-Math sabit noktalı sayı standardında kesirli kısım için geliştirilen bir diğer yöntem ise; integer kısmı yine ikilik tabana dönüştürülmektedir. Fractional kısım ise $Q \cdot 2^n$ ile gösterilmektedir. Q kesirli kısmın onluk tabandaki değeri ifade ederken n ise fractional kısmın bit sayısını temsil eder. Böylece $Q \cdot 2^n$ işlem sonucu binary tabana dönüştürülerek fractional kısmın IQ-Math sabit noktalı sayı standardının matematiksel değeri elde edilir. 92,1245 sayısı bu yöntemle sabit noktalı sayı formatına dönüştürme işlemi aşağıda gösterilmiştir.

Sayı: 92,1245

S=0

I(Tam kısım) : 92

Q(kesirli kısım) :0,1245

Sayının Integer kısmı ikilik tabanda;

$$92=(1011100)_2$$

Q(kesirli kısım) olan 0,1245 ise;

$$0,1245 \cdot 2^8 = 31,872 \text{ sonucu bulunur.}$$

Bulunan bu sonucun tam sayı ile ifade edilen kısmı ikilik tabana dönüştürüldüğünde IQ-Math sabit sayı formatı değeri olarak $31=(00011111)_2$ bulunur. 92,1245 sayısının bu yöntemle 16 bit 8I-8Q sabit noktalı sayı standardında 01011100.00011111 sonucu bulunur.

Pozitif sayılarda hesaplama bu şekilde yapılırken negatif sayılarda hesaplama ise 2'ye tümleyerek yapılır. 8I-8Q sabit noktalı sayı formatındaki 92,1245 sayının değeri 2'ye tümleyen şeklinde kendi değerinin yine negatifi olarak diğer bir deyişle -92,1245 olarak hesaplanır.

$$1. \ 01011100.00011111 \longrightarrow 92,1245$$

$$2. \ 10100011.11100000 \longrightarrow 1'e \ tümleyeni$$

3. 10100011.11100001 \longrightarrow 2'ye tümleyeni \longrightarrow -92,1245 değerine bulunur.

2.6. NÜMERİK ALGORİTMALAR

Literatürde diferansiyel denklemlerin sayısal çözümleri için çeşitli algoritmalar geliştirilmiştir. Bu algoritmalara örnek olarak Euler, Heun, dördüncü dereceden Runge-Kutta (RK-4), beşinci dereceden Runge-Kutta Butcher ve Dormand-Prince metotları örnek olarak verilebilir. Aşağıda belirtilen bu algoritmalar ayrıntılı bir şekilde incelenmiştir. Verilen bu algoritmaların tümü kaotik osilatörlerin nümerik olarak modellenmesi için kullanılabilir. Ayrıca ilgili kaotik sistemler farklı platformlarda gerçekleştirilmek istendiğinde bu nümerik modeller referans olarak alınmaktadır. Sunulan bu tez çalışmasında Euler, Heun, dördüncü dereceden Runge-Kutta (RK-4), beşinci dereceden Runge-Kutta Butcher ve Dormand-Prince metotları içerisinde Van der Pol osilatörünün modellenmesi için Euler algoritması kullanılmıştır. Bunun en temel sebebi Euler algoritmasının diğer algoritmalara göre daha basit yapıda olmasıdır. Bunun bir sonucu olarak ilgili algoritma FPGA çipleri üzerinde daha kolay bir şekilde tasarlanacaktır. Tasarımdan elde edilen FPGA maksimum çalışma frekansı yüksek ve çip kaynak kullanımını düşürecektir.

2.6.1. Euler Nümerik Algoritması

Yukarıda bahsedilen algoritmalarından birisi olan Euler algoritması sayısal çözümün kısa sürede hesaplanabilmesi ve sayısal işlemciler üzerinde kolay bir şekilde modellenmesi gibi avantajlarından dolayı tercih edilmektedir. Euler algoritması Eşitlik 2.4'te verilmektedir.

$$v_{\lambda+\Delta h} \approx v_{\lambda+1} = v_{\lambda} + f(v_{\lambda})\Delta h \quad (2.4)$$

Eşitlikte Δh adım aralığı, v_{λ} başlangıç değerleri ve $v_{\lambda+1}$ ise sistemin bir iterasyon sonar alacağı değerleri ifade etmektedir. Sürekli zamanlı sistemlerin sayısal işlemciler üzerinde gerçek zamanlı olarak modellenmesi amacıyla kullanılan nümerik algoritmalarda, sistemin bir sonraki $v_{\lambda+1}$ değerinin hesaplanması için bir önceki değeri olan v_{λ} değeri kullanılmaktadır.

2.6.2. Heun Nümerik Algoritması

Heun nümerik algoritmasının geliştirilme sebebi; Euler algoritması'nın küçük adım

uzunluđuna gereksinimi, büyük adım uzunluđunda yeterince hassas sonuç vermemesi, yüksek hata oranıdır. Heun nümerik algoritması; Euler algoritmasından daha ileri seviyede nümerik çözümler üretmişlerdir. Heun nümerik algoritması Euler yöntemindeki noktadaki türevi yerine, türevi ve +1 deki türevlerin aritmetik ortalaması alınarak oluşturulan bir yöntemdir [61, 90]. Yani i inci noktadaki türev yerine i ve (i+1) inci noktadaki türevlerin aritmetik ortalaması alınır. Heun algoritmasının matematiksel eşitliđi Eşitlik (2.5)'te verilmiştir [77].

$$v(v_0) = v_i = v_0$$

$$f(v_{\lambda+1}^0) = v_{\lambda} + f(v_{\lambda}) * h$$

$$f(v_{\lambda+1}) = v_{\lambda} + \frac{f(v_{\lambda}) + f(v_{\lambda+1}^0)}{2} * h \quad (2.5)$$

2.6.3. RK4 Nümerik Algoritması

Runge-Kutta yöntemlerinden biri olan RK4 nümerik algoritması; ikiden çok terim kullanılarak Taylor serisi kullanılarak gerçekleştirilir. Dolayısıyla da Euler ve Heun nümerik analiz yöntemlerinden daha da hassas çözümler ürettiđi için hata oranı sayısal deđeri de çok düşüktür. Runge-Kutta-4 nümerik algoritmasının matematiksel eşitlik denklemleri; Eşitlik (2.6)'da verilmiştir [17, 76].

$$v_{i+1} = v_i + \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4)$$

$$k_1 = f(v_i)$$

$$k_2 = f(v_i + \frac{h}{2} k_1)$$

$$k_3 = f(v_i + \frac{h}{2} k_2)$$

$$k_4 = f(v_i + h k_3) \quad (2.6)$$

2.6.4. RK5-Butcher Nümerik Algoritması

Runge Kutta 4 algoritmasına ek olarak k5 ve k6 olmak üzere iki parametre daha eklendiğinde elde edilen nümerik analiz yöntemlerinden biri olan RK5-Butcher algoritması, RK4'e göre daha duyarlı sonuçlar elde edilmektedir. RK5- Butcher algoritmasının matematiksel eşitlik denklemi; Eşitlik (2.7)'de verilmiştir [76,91].

$$V_{i+1} = v_i + \frac{h}{90} (7k_1 + 32k_3 + 32k_5 + 7k_6)$$

$$k_1 = f(v_i)$$

$$k_2 = f\left(v_i + \frac{h}{4}k_1\right)$$

$$k_3 = f\left(v_i + \frac{h}{8}k_1 + \frac{h}{8}k_2\right)$$

$$k_4 = f\left(v_i - \frac{h}{2}k_2 + hk_3\right)$$

$$k_5 = f\left(v_i + \frac{3h}{16}k_1 + \frac{9h}{16}k_4\right)$$

$$k_6 = f\left(v_i - \frac{3h}{7}k_1 + \frac{2h}{7}k_2 + \frac{12h}{7}k_3 - \frac{12h}{7}k_4 + \frac{8h}{7}k_5\right) \quad (2.7)$$

2.6.5. Dormand Prince Nümerik Algoritması

DP nümerik algoritması yedi parametreden oluşur. Bu terimler; k1, k2, k3, k4, k5, k6 ve k7 olarak adlandırılır. Yapı itibarıyla RK5-Butcher nümerik algoritmasına benzemesine rağmen denklemde sadece k7 parametresi eklenerek daha hassas sonuç elde edilmiştir. Bu avantaj olarak görülse de sistemin tek dezavantajı parametre sayısının artışıyla yazılımsal ve donanımsal gerekleştirmenin güçleşmesi ve sayısal tabanlı gerçek zamanlı uygulamalarda daha çok çip kaynağı tüketmesine sebep olmaktadır [77].

Bu algoritmaya ait olan denklem, Eşitlik 2.8'de görülmektedir. Algoritmanın başlangıç değeri v_i , adım sayısı ise h ile gösterilmiştir. Eşitlikte ele alınan $h=0,01$ olarak kabul edilmiştir [77, 92, 93].

$$v_{i+1} = v_i + h \left(\frac{35}{384} k_1 + \frac{500}{384} k_3 + \frac{125}{192} k_4 - \frac{2187}{6784} k_5 + \frac{11}{84} k_6 \right)$$

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

$$k_2 = f\left(x_i + \frac{h}{5}, v_i + \frac{h}{5} k_1\right)$$

$$k_3 = f\left(x_i + \frac{3}{10} h, v_i + \frac{3}{40} k_1 + \frac{9}{40} k_2 * h\right)$$

$$k_4 = f\left(x_i + \frac{4}{5} h, \left[v_i + \frac{44}{45} k_1 - \frac{56}{15} k_2 + \frac{32}{9} k_3\right] * h\right)$$

$$k_5 = f\left(x_i + \frac{8}{9} h, \left[v_i + \frac{19372}{6561} k_1 - \frac{25360}{2187} k_2 + \frac{64448}{6561} k_3 - \frac{212}{729} k_4\right] * h\right)$$

$$k_6 = f\left(x_i + h, v_i + \frac{9017}{3168} k_1 - \frac{355}{33} k_2 + \frac{46732}{5247} k_3 + \frac{49}{176} k_4 - \frac{5103}{18656} k_5 * h\right)$$

$$k_7 = f\left(x_i + h, v_i + \frac{35}{384} k_1 + 0 * k_2 + \frac{500}{1113} k_3 + \frac{125}{192} k_4 - \frac{2187}{6784} k_5 + \frac{11}{84} k_6 * h\right) \quad (2.8)$$

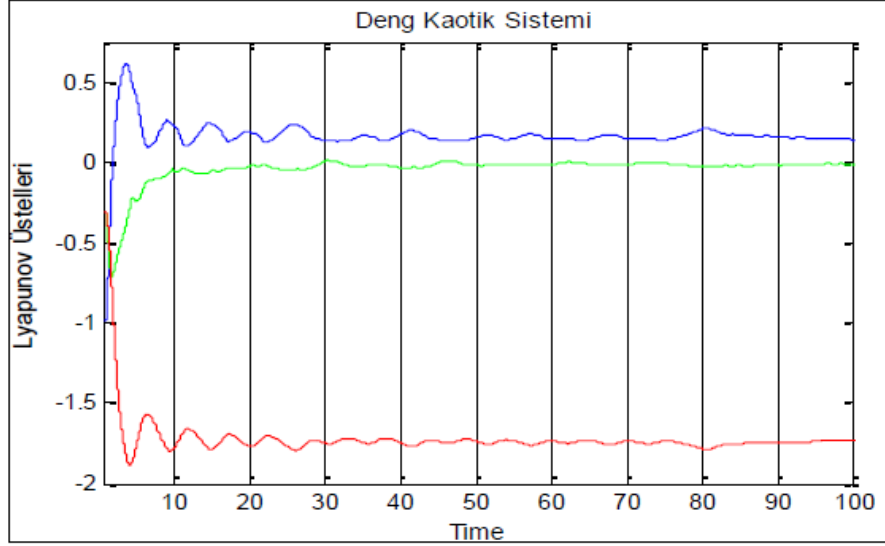
2.7. KAOS VE ANALİZ YÖNTEMLERİ

Kaotik ve Kaos sistemin temel karakteristiği; başlangıç şartlarına hassasiyet, sınırsız sayıda değişik periyodik salınımlar içermesi yani aperiodyklik, zaman boyutunda düzensizliğinin rasgele olmayıp deterministik şekilde olması, gürültü benzeri güç spektrumuna sahip olması, genliği ve frekansı tespit edilemeyen ancak sınırlı bir alan içerisinde değişen kompleks davranışlar sergilemesi olarak belirtilebilir [38]. Kaos analizinin yapılabilmesi için sistemin 2 şartı sağlaması gereklidir. Bunlardan ilki; sistem içerisinde bir ya da daha fazla doğrusal olmayan eleman bulundurma şartıdır. Diğeri ise; sürekli zamanlı sistemlerde sistemin en az üçüncü dereceden bir sistem olma şartıdır [16]. Bu şartlar sağlandıktan sonra literatürde yer alan kaotik analiz yöntemleri mevcuttur [47]. Bunlardan en yaygın olanları ise aşağıda listelenmiştir.

1. Yörünge'nin izlenmesi (zaman serileri – Time serious),
2. Sistemin faz uzayının incelenmesi (Phaseportrait),
3. Lyapunov üstelleri,
4. Lyapunov üstelleri frekans spektrumu,
5. Poincare haritalama,
6. Fraktal boyut analizi,
7. Güç spektrumu,
8. Çatallaşma diyagramı (Bifurcation diagram) örnek olarak verilebilir [38, 47].

En temel ve en önemli özellik başlangıç şartlarına bağlıdır. İki yörünge arasındaki uzaklık başlangıçta her ne kadar birbirine yakın olsa da zamanla birbirinden uzaklaşır ve bu mesafe genellikle zamana göre üssel olarak artar. Sistemin limit kümesinin fraktal (parçalı) olması sistemin ikincil bir özelliğidir. Denge noktası, limit çevrim ve torusda boyut bir tamsayı iken kaotik bir sistemin boyutu parçalıdır. Kaotik sistemler temelde deterministik olsalar bile, kaotik sinyallerin rastgele süreçlerle benzer karakteristiklere sahip olduğu görülmüştür. Kaotik bir davranışı diğer davranışlardan ayıran diğer göstergeler, faz resmi görünümü ve frekans spektrumdur. Kaotik yapıya sahip sistemlerde faz resminin zaman gelişimi, dinamik sistemin yapısının belirlediği faz uzayı bölgesinde, sayılamayacak kadar yörüngeyle dolması şeklinde olur. Zaman ilerledikçe, yörüngeler faz uzayını doldurmaya başlar ve hiçbir zaman üzerine kapanmaz, tekrar eder. Faz uzayının bu şekilde dolması kaotik işaretlerden biridir [16].

Adından da anlaşılacağı üzere Aleksandra Mikhailovich Lyapunov tarafından geliştirilen Lyapunov üstelleri yöntemi; sistemin zaman serisinin kaotik bileşenler içerip içermediğini gösteren matematiksel bir analiz yöntemidir [47, 94]. Lyapunov üstelinde; başlangıçta birbirine yakın olan yörüngelerin ıraksama ya da yakınsama oranını ölçülür. Eğer bazı kaotik sistemlerde aynı denemeler üst üste iki kez tekrarlandığında bir sonrakinde tamamen farklı sistem değişimleri gözlenebilmektedir. Lyapunov üsteli yöntemi, “başlangıç şartlarına hassas bağlılık” özelliğinin sayısal bir ifadesi olup eğer üç boyutlu bir faz uzayında Lyapunov yöntemi ile kaotik analizi yapılan sistemin üstellerinin ($\lambda_1, \lambda_2, \lambda_3$) işaretleri sırasıyla (+, 0, -) ise sistem kaos durumundadır. Sistemin dengede olması için sistem işaretleri (-,-,-) olmalı, kendini sürekli tekrarlayan durumda olunca da (0,-,-) ve halka olunca da (0,0,-) işaretleri sırasıyla kullanılmaktadır. Dört boyutlu bir sistemde (+, 0, -, -) ve (+, +, 0, -) olmak üzere iki mümkün durum vardır. (+, +, 0, -) durumu hiper Kaos olarak adlandırılmaktadır [47]. Aşağıda Şekil 2.17’de Deng Kaotik Sistemi’ne ait örnek Lyapunov üstelleri grafiği verilmiştir.

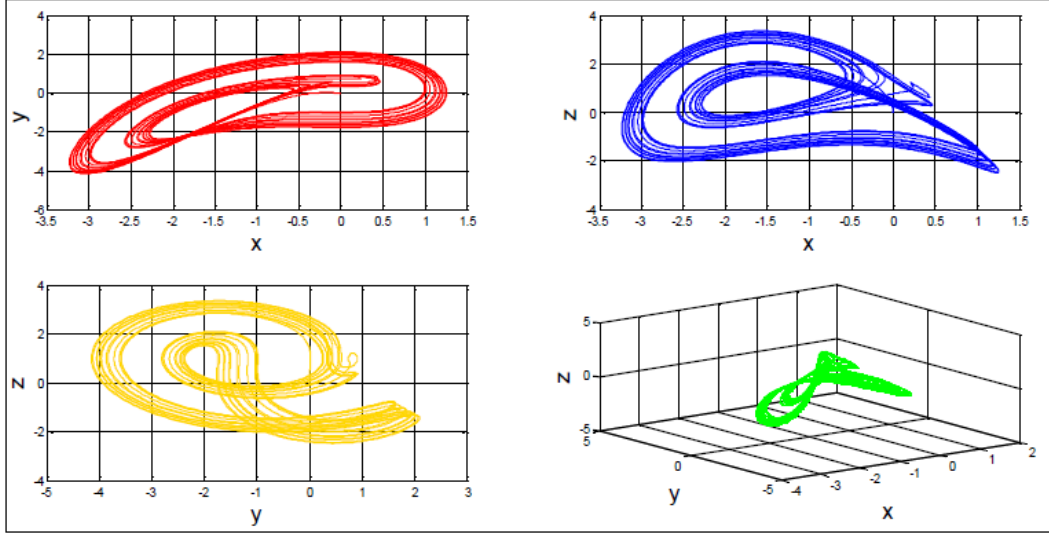


Şekil 2.17. Örnek Lyapunov üstelleri grafiği [77, 94].

Poincare kesit; faz uzayı içerisindeki yaklaşık bütün yörüngelere karşılık gelen eğriler planına verilen isimdir. Dinamik bir sistemin faz uzayındaki davranışı, faz uzayı belirli bir düzlemlle kesilerek (Poincare kesit), yörüngelerin o düzlemi kestiği noktaların oluşturduğu bir geometrik harita şeklinde de gözlemlenebilir. Poincare kesitteki noktaların dağılımı tek ve küçük bir bölgede sonlu sayı da ise hareket periyodik, kapalı bir eğri ise hareket yarı periyodik, belirli alanlarda yoğunlaşmış kümeler şeklinde ise hareket kaotiktir [16, 95].

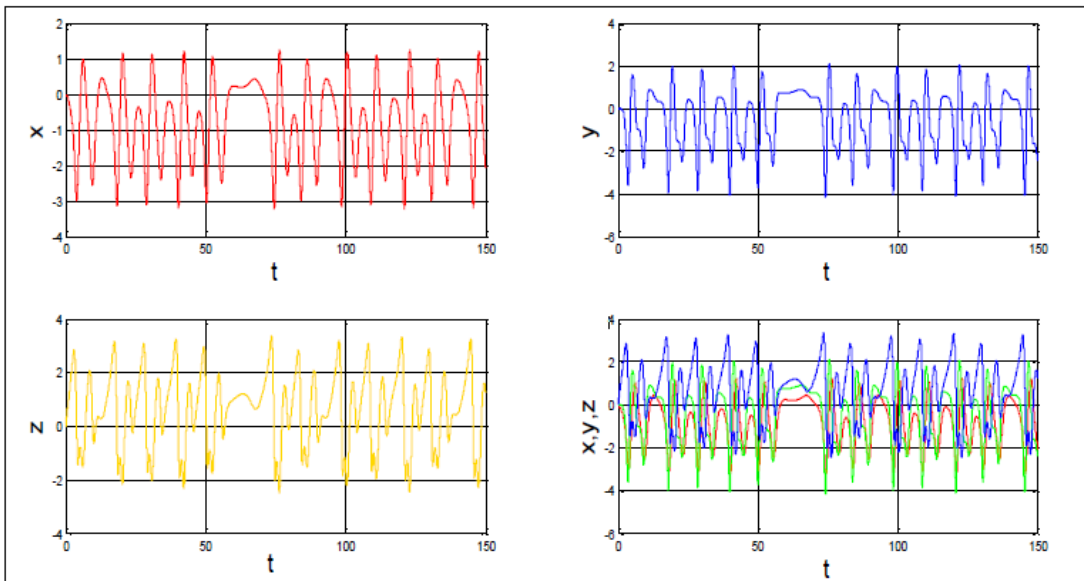
Mandebrot tarafından ortaya konan diğer bir terim de Fraktal geometridir. Doğal nesnelerin yapısal bütünlükleri incelenerek matematiksel obje olan nokta, çizgi, küpün boyutları, geometrik yapısı hakkında kesin bilgi verdiği için Mandebrot, tamsayı ile ifade edilebilen boyutların kaotik yapılara uygun olmadığını belirtmiştir. Kaotik sistem çekerleri fraktal boyutlara sahipken, doğrusal sistem çekerleri tamsayı boyutlara sahiptir [95].

Diğer bir analiz yöntemi olarak geçen sistemin faz portreleri incelendiğinde; faz portreleri faz uzayı bölgesinde zamanla yörüngesini doldurmaya başlar ve periyodiktir. Eğer faz uzayı bu şekilde doldurulmuş ise sistemin kaotik işaret taşıdığına göstergesidir. Faz portreleri işlemleri genellikle MATLAB programı kullanılarak örneklendirilmektedir ve aşağıda Şekil 2.18’de altın orana sahip faz portreleri örneklendirilmiştir.



Şekil 2.18. Altın orana sahip faz portreleri [77].

Zaman serileri analizi incelemesi yapıldığında; sistemin en temel ve en önemli özelliği olan başlangıç şartlarına bağıllığı, sisteme verilen değişik başlangıç değerleri belirli zaman dilimi sonrasında farklı kaotik işaretler üretebilmektedir. Farklı başlangıç değerleri ile üretilen kaotik işaretleri gözlemlemek için sistemin zaman serisi analizi yapılmaktadır [16]. Aşağıda Şekil 2.19’da MATLAB tabanlı altın orana sahip kaotik sistemin x, y, z ve x-y-z zaman serisi analizi verilmiştir.



Şekil 2.19. Altın orana sahip kaotik sistemin zaman serisi analizi [77].

3. FPGA TABANLI VAN DER POL OSİLATÖRÜ TASARIMLARI

Bu bölümde Van Der Pol sistemi (VDPS) Euler nümerik diferansiyel denklem çözüm yöntemi kullanılarak ayrıklaştırılmış modeli çıkarılmıştır. Ayrıca Van Der Pol sistemi VHDL dili kullanılarak FPGA üzerinde tasarımları ve gerçeklemeleri yapılmıştır. Yapılan çalışmalar sonucunda elde edilen FPGA tabanlı Van Der Pol sistemi yapısının performans ve çip istatistikleri incelenmiştir. Ayrıca FPGA tabanlı Van Der Pol osilatörünün ürettiği sinyaller kullanılarak hassasiyet analizi yapılmıştır.

Bu çalışmada VDPS sisteminin Euler algoritması kullanılarak FPGA tabanlı olarak modellenmesini sağlamak amacıyla sistemin ayrıklaştırılmış modeli oluşturulmuştur. Algoritmanın ayrıklaştırılmış modelindeki $v(k)$ ve $w(k)$ 'nin başlangıç değerleri $v(t_0)=v(k)=0,0$ ve $w(t_0)=w(k)=0,1$ olarak alınmıştır. Aşağıda Denklem (3.1)'de Euler algoritması kullanılarak VDPS ayrıklaştırılmış matematiksel modeli sunulmaktadır.

$$\begin{aligned} v(k+1) &= v(k) + \Delta h(y(k)) \\ w(k+1) &= w(k) + \Delta h(\mu(1-v(k)^2)w(k) - v(k)) \end{aligned} \quad (3.1)$$

Eşitlikte Δh adım aralığı, w^λ başlangıç değerleri ve $w^{\lambda+1}$ ise sistemin bir iterasyon sonar alacağı değerleri ifade etmektedir. Sürekli zamanlı sistemlerin sayısal işlemciler üzerinde gerçek zamanlı olarak modellenmesi amacıyla kullanılan nümerik algoritmalarda, sistemin bir sonraki $w^{\lambda+1}$ değerinin hesaplanması için bir önceki değeri olan w^λ değeri kullanılmaktadır.

3.1. FPGA ÜZERİNDE IEEE-754-1985 TABANLI VAN DER POL OSİLATÖRÜ TASARIMI

Bu bölümde Van der Pol osilatörü, Euler nümerik algoritması ile 32-bit IEEE-754-1985 kayan noktalı sayı standardı kullanılarak FPGA çipleri üzerinde çalışmak üzere modellenmiştir. Yapılan tasarım bir donanım tanımlama dili olan VHDL'de kodlanmıştır. FPGA tabanlı Van der Pol osilatörü tasarımında kullanılan kayan noktalı sayı standardındaki çarpıcı, toplayıcı, bölücü ve çıkarıcı gibi birimler, Xilinx ISE

Project Navigator program ile sunulan IP CORE üretici ile gerçekleştirilmiştir. Van der Pol osilatörünün Euler nümerik algoritması ile FPGA çipleri üzerinde çalışmak üzere modellenmesi 32-bit IEEE-754-1985 kayan noktalı sayı standardı kullanılarak gerçekleştirildiğinden kullanılan onluk sayı sistemindeki sayıların 32-bit IEEE-754-1985 kayan noktalı sayı standardına dönüştürülmesi gerekmektedir. Bu amaç ile kullanılan program ekranı Şekil 3.1’de verilmiştir. Tasarlanan programa onluk sayı sistemindeki değer girildiğinde sayının 32-bit IEEE-754-1985 kayan noktalı sayı standardındaki değeri onaltılık (hexadecimal) ve ikilik (binary) sayı sistemindeki karşılığı görülmektedir.

IEEE-754 Floating-Point Conversion

From Decimal Floating-Point
To 32-bit and 64-bit Hexadecimal Representations
Along with Their Binary Equivalents

Enter a decimal floating-point number here,
then click either the **Rounded** or the **Not Rounded** button.

Decimal Floating-Point:

Rounding from floating-point to 32-bit representation uses the IEEE-754 round-to-nearest-value mode.

Results:

Decimal Value Entered:

Single precision (32 bits):

Binary: *Status:*

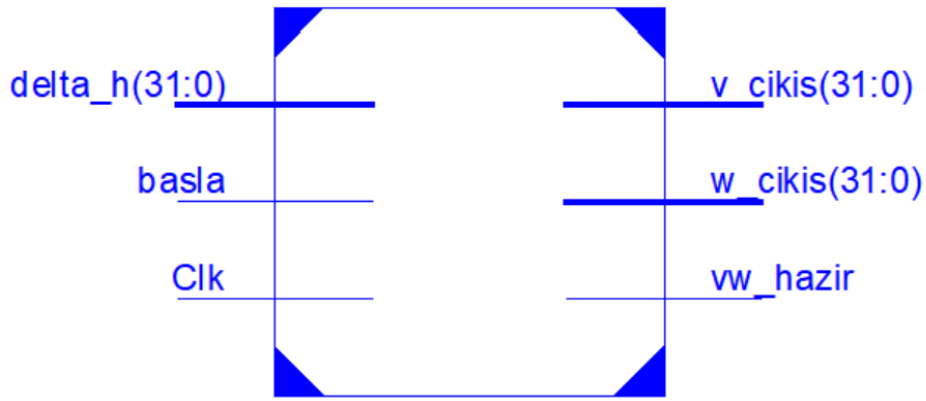
Bit 31 Sign Bit <input type="text" value="1"/> 0: + 1: -	Bits 30 - 23 Exponent Field <input type="text" value="01111110"/> Decimal value of exponent field and exponent <input type="text" value="126"/> - <input type="text" value="127"/> = <input type="text" value="-1"/>	Bits 22 - 0 Significand <input type="text" value="1.1111000010100011110110"/> Decimal value of the significand <input type="text" value="1.9400001"/>
--	--	---

Hexadecimal: *Decimal:*

Şekil 3.1. Onluk sayı sisteminden 32-bit IEEE-754-1985 kayan noktalı sayı standardına dönüşüm için kullanılan program görüntüsü.

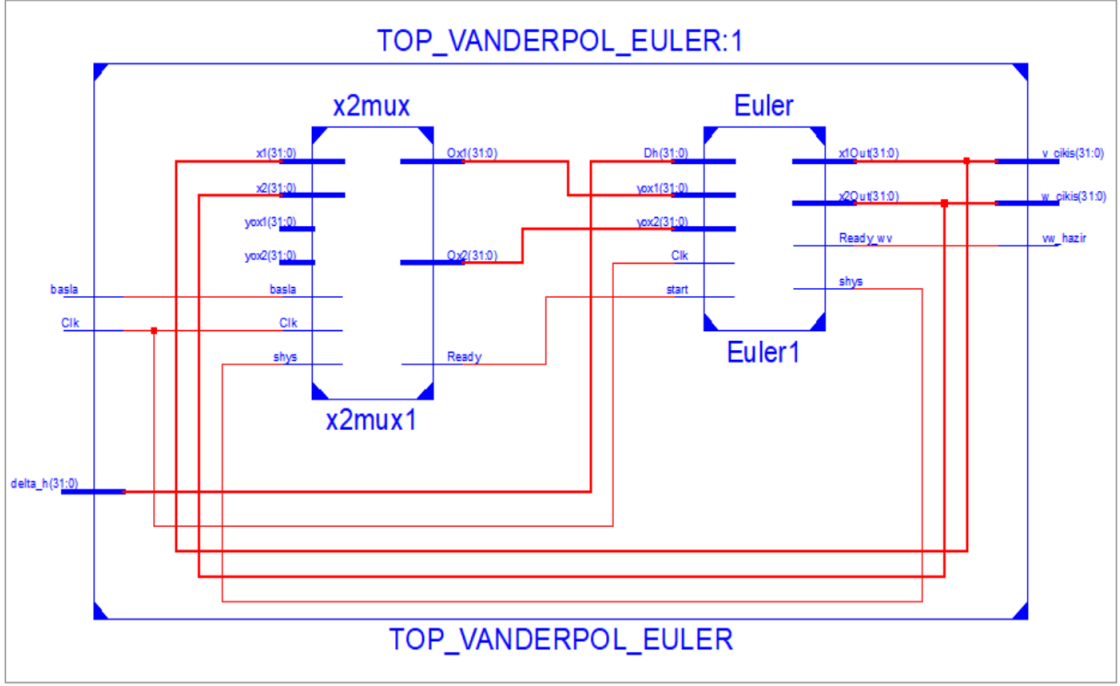
Van der Pol osilatörünün Euler algoritması ile FPGA tabanlı olarak modellenmesine birinci seviye blok şeması Şekil 3.2’de verilmiştir. Tasarlanan sistemin girişinde bulunan 1-bitlik Basla ve Clk sinyalleri, tüm ünitelerin zamanlaması ve ünitelerin bağlı bulunduğu sistem ile senkronizasyonu sağlamak amacıyla kullanılmaktadır. Tasarımda osilatörün başlangıç değerleri FPGA çipinin kaynak kullanımını azaltmak amacıyla ise

32-bitlik başlangıç şartları tasarımın içerisine gömülmüştür. Ancak ihtiyaç duyulduğunda ilgili sinyaller tasarımcı tarafından tasarımın dışarısına alınabilir. 32-bit delta_h sinyali Euler algoritmasının adım sayısını ifade etmektedir. Bu sinyal dışarıdan verildiği için oldukça kolay bir şekilde algoritmanın adım değeri gerçek zamanlı olarak değiştirilebilmektedir. Fakat istenirse bu sinyal FPGA giriş-çıkış sayısını azaltmak amacıyla ile çip içerisine gömülerek giriş-çıkış kaynak kullanımını azaltılabilir. FPGA-tabanlı Van der Pol osilatörü ünitesinde 2 adet w_cikis ve v_cikis kayan nokta sayı standardında 32 bit çıkış sinyalleri ve bu çıkış sinyallerinin hazır olduğunu göstermek amacıyla kullanılan 1-bit vw_hazir sinyali bulunmaktadır.



Şekil 3.2. FPGA tabanlı kaotik Van der Pol osilatörü birinci seviye blok diyagramı.

Şekil 3.3'te FPGA çiplerinde çalışmak üzere tasarımı yapılan kaotik Van der Pol osilatörü ikinci seviye blok diyagramı sunulmuştur. Tasarlanan sistem X2MUX ve Euler ünitesi olmak üzere 2 üniteden oluşmaktadır. X2MUX ünitesi, sistemin başlangıç şartları ile osilatörün ürettiği sinyaller için seçici görevi yapmaktadır. Euler ünitesi ise kaotik Van der Pol sinyallerini üreten ünite dir. Tasarımın Start girişine '1' sinyali uygulandığında ve sistemin ihtiyaç duyduğu başlangıç koşulları X2MUX aracılığı ile Euler ünitesine aktarılmaktadır. Euler ünitesi ilk değerini ürettiğinde ise Ready_wv sinyali '1' olmakta ve bu anda Van der Pol osilatör ünitesi w_cikis ve v_cikis olarak ilk çıkışlarını üretmektedir. Bu andan itibaren X2MUX ünitesi başlangıç şartları yerine Euler ünitesinin ürettiği sinyalleri sisteme göndermektedir.



Şekil 3.3. FPGA tabanlı kaotik Van der Pol osilatörü ikinci seviye blok diyagramı.

Şekil 3.4'te Euler tabanlı kaotik Van der Pol osilatör ünitesi üçüncü seviye blok diyagramı görülmektedir. Osilatör ünitesi yapısında X2MUX, V, En_Mult, En_adder ve Counter üniteleri olmak üzere 7 ünite bulunmaktadır. Buradaki V ünitesi X2MUX ünitesinden gelen kontrol sinyalleri ile Van der Pol sistemi denklemlerindeki $w(k+1)$ ve $v(k+1)$ değerlerinin hesaplanmasını sağlamaktadır. Üniteden elde edilen sinyaller başlangıç değeri sinyalleri ($w(k)$ ve $v(k)$) ile En_adder ünitesinde toplanarak sonuçlar Counter Ünitesi'ne gönderilmektedir. Counter ünitesi sistem pipeline olarak tasarlandığından istenmeyen sinyallerin filtrelenmesi amacı ile kullanılmaktadır. FPGA tabanlı tasarımı yapılan Van der Pol osilatörü içerisinde kullanılan 32 bit kayan noktalı sayı formatındaki çarpma, bölme, toplama, çıkarma ve diğer modüller Xilinx firmasının geliştirdiği IP Core üretici ile gerçekleştirilmiştir.

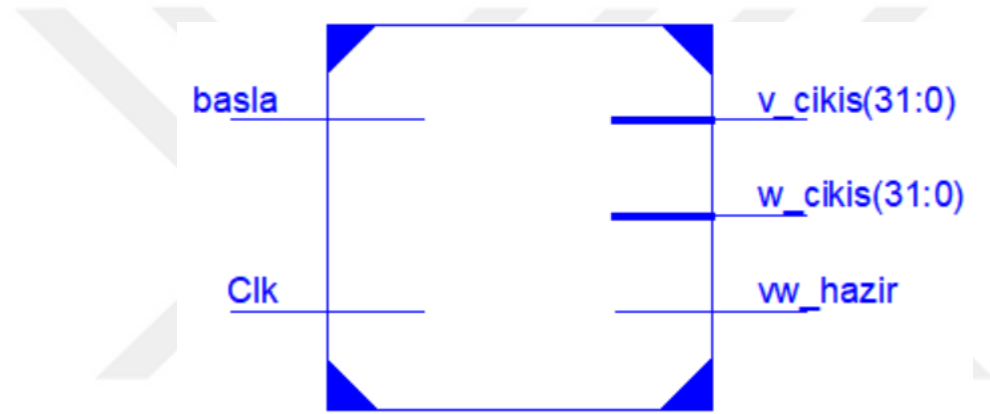
3.2. FPGA ÜZERİNDE IQ-MATH TABANLI VAN DER POL OSİLATÖRÜ TASARIMI

Bu bölümde Van der Pol osilatörü, Euler nümerik algoritması ile 32-bit IQ-Math sabit noktalı sayı standardı kullanılarak FPGA çipleri üzerinde çalışmak üzere modellenmiştir. Literatürde farklı tam sayı ve ondalıklı bölümleri bulunan IQ-Math sayı formatları bulunmaktadır. Sunulan bu çalışmada yapılan tasarımlar 32-bit IQ-Math sayı standardı olarak 16-bit tam sayı (integer) ve 16-bit ondalıklı (fractional) sayı standardı kullanılmıştır. Yapılan IQ-Math-tabanlı tasarım bir donanım tanımlama dili olan VHDL dilinde kodlanmıştır. FPGA tabanlı Van der Pol osilatörü tasarımında kullanılan IQ-Math sabit noktalı sayı standardındaki toplayıcı, bölücü ve çıkarıcı gibi birimler, Xilinx ISE Project Navigator program ile sunulan IP-CORE üretici ile gerçekleştirilmiştir. Van der Pol osilatörünün Euler nümerik algoritması ile FPGA çipleri üzerinde çalışmak üzere modellenmesi 32-bit IQ-Math sabit noktalı sayı standardı kullanılarak gerçekleştirildiğinden kullanılan onluk sayı sistemindeki sayıların 32-bit IQ-Math sabit noktalı sayı standardına dönüştürülmesi gerekmektedir. Bu amaç ile kullanılan program ekranı Şekil 3.5'te verilmiştir. Tasarlanan programa onluk sayı sistemindeki değer girildiğinde sayının 32-bit IQ-Math sabit noktalı sayı standardındaki değeri ikilik sayı sisteminde görülmektedir.

	A	E	DH	DI
	Sayı Değeri		32 bitlik IQ-Math değeri	
1				
2				
3	-0,9700000000		11111111111111110000011110101111	
4	-1,5000000000		11111111111111110100000000000000	
5	30,7500000000		00000000000111101100000000000000	
6	-30,7500000000		11111111111000010100000000000000	
7	-30,0000000000		11111111111000100000000000000000	
8	15,0000000000		00000000000011110000000000000000	
9	-15,0000000000		11111111111100010000000000000000	
10	2,7500000000		00000000000000010110000000000000	
11	-2,7500000000		11111111111110101000000000000000	

Şekil 3.5. Onluk sayı sisteminden 32-bit IEEE-754-1985 kayan noktalı sayı standardına dönüşüm için kullanılan program görüntüsü.

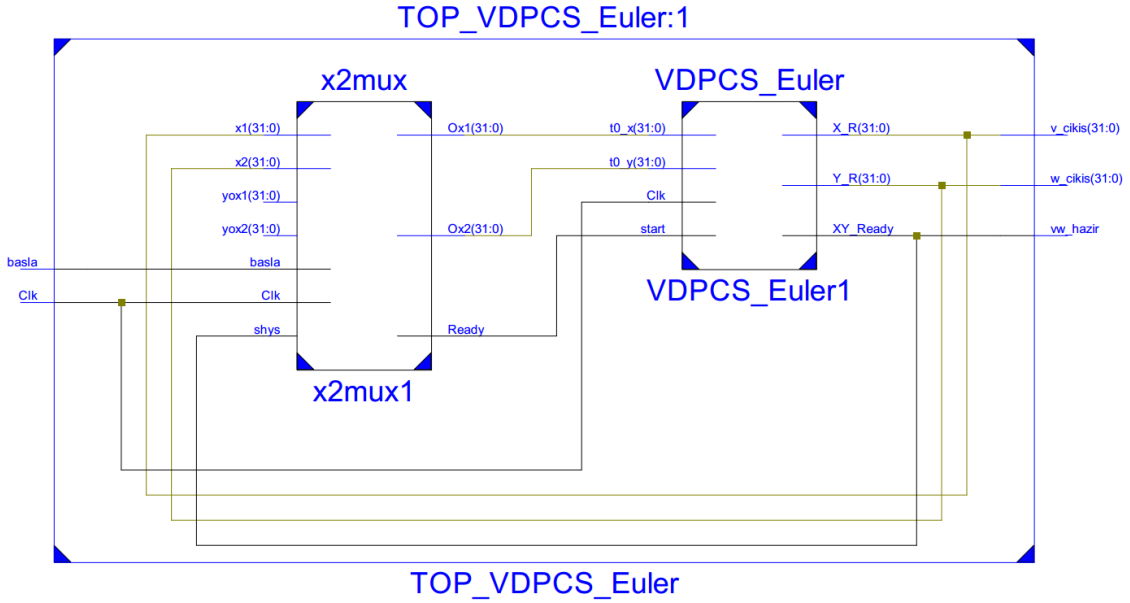
Van der Pol osilatörünün Euler algoritması ile FPGA tabanlı olarak modellenmesine birinci seviye blok şeması Şekil 3.6’da verilmiştir. Tasarlanan sistemin girişinde bulunan 1-bitlik basla ve Clk sinyalleri, tüm birimlerin zamanlaması ve birimlerin bulunduğu yapı ile arasındaki eş-zamanlı sinyal akışını sağlamak için tasarlanmıştır. Tasarımda Van der Pol osilatörün başlangıç şartı değerleri FPGA çipinin kaynak kullanımını azaltmak amacıyla ise 32-bitlik sabit olarak tasarımın içerisine gömülmüştür. Ancak ihtiyaç duyulduğunda ilgili sinyaller tasarımcı tarafından tasarımın dışarısına alınabilir. FPGA-tabanlı Van der Pol osilatörü ünitesinde 2 adet w_cikis ve v_cikis IQ-Math sabit nokta sayı standardında 32-bit çıkış sinyalleri ve bu çıkış sinyallerinin hazır olduğunu göstermek amacıyla kullanılan 1-bit vw_hazir sinyali bulunmaktadır.



Şekil 3.6. Euler tabanlı kaotik VDPS ünitesinin en üst seviye blok diyagramı.

Şekil 3.7’de FPGA çiplerinde çalışmak üzere IQ-Math sabit noktalı sayı standardı kullanılarak tasarımı yapılan kaotik Van der Pol osilatörü ikinci seviye blok diyagramı sunulmuştur. Tasarlanan sistem X2MUX ve VDPCS_Euler ünitesi olmak üzere 2 üniteden oluşmaktadır. X2MUX ünitesi, sistemin başlangıç şartları ve Van der Pol osilatörünün ürettiği sinyaller arasında seçici görevini gerçekleştirmektedir. E VDPCS_Euler ünitesi ise kaotik Van der Pol sinyallerini üreten ünitedir. Tasarımın 1-bit basla girişine ‘1’ sinyali uygulandığında ve sistemin ihtiyaç duyduğu başlangıç koşulları X2MUX aracılığı ile VDPCS_Euler ünitesine aktarılmaktadır. VDPCS_Euler ünitesi ilk değerini ürettiğinde ise XY_Ready sinyali ‘1’ değerini almaktadır. XY_Ready sinyalinden gelen ‘1’ değeri shys sinyali aracılığı ile X2MUX ünitesine gönderilmektedir. Bu sinyal X2MUX ünitesine ulaştığında X2MUX ünitesi çıkışına sistemin başlangıç için ihtiyaç duyduğu sabit olarak tanımlanan ilk değerlerini değilde VDPCS_Euler osilatör ünitesinin ilk iterasyon sonunda üretmiş olduğu sinyalleri

göndermektedir. Bu şekilde VDPCS_Euler osilatör ünitesi her ürettiği w_cikis ve v_cikis sinyallerini hem X2MUX ünitesine hem de çıkışlara aktarmaktadır. Bu andan itibaren VDPCS_Euler ünitesinin ürettiği vw_hazir sinyal değeri '1' olmaktadır. Diğer tüm durumlarda ilgili sinyal değeri '0' olmaktadır.



Şekil 3.7. Eulertabanlı kaotik Van der Pol osilatörü ikinci seviye blok diyagramı.

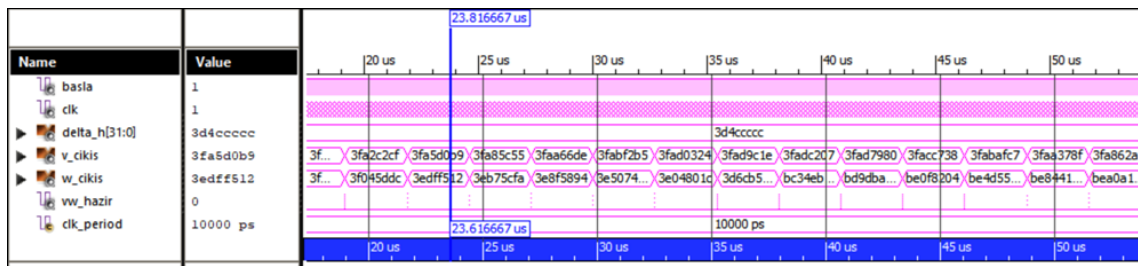
Şekil 3.8'de Euler tabanlı kaotik Van der Pol osilatör ünitesi üçüncü seviye blok diyagramı görülmektedir. Osilatör ünitesi yapısında X2MUX, VDPCS, i16_q16_mult, i16_q16_adder, ft5L ve filter üniteleri olmak üzere 8 ünite bulunmaktadır. Buradaki VDPCS ünitesi X2MUX ünitesinden gelen kontrol sinyalleri ile Van der Pol sistemi denklemlerindeki $w(k+1)$ ve $v(k+1)$ değerlerinin hesaplanmasını sağlamaktadır. Üniteden elde edilen sinyaller başlangıç değeri sinyalleri ($w(k)$ ve $v(k)$) ile i16_q16_adder ünitesinde toplanarak sonuçlar filter ünitesi'ne gönderilmektedir. Counter ünitesi sistem pipeline olarak tasarlandığından istenmeyen sinyallerin filtrelenmesi amacı ile kullanılmaktadır. FPGA tabanlı tasarımı yapılan Van der Pol osilatörü içerisinde kullanılan 32 bit kayan noktalı sayı formatındaki çarpma, toplama, çıkarma ve diğer modüller Xilinx firmasının geliştirdiği IP Core üretici ile gerçekleştirilmiştir. Tasarımda bulunan ft5L ünitesi tasarım pipeline olarak çalıştığından ilgili sinyallerde gecikme olmaması açısından sinyal senkronizasyonunu sağlamak amacı ile geciktirme ünitesi olarak tasarlanmıştır.

4. FPGA TABANLI VAN DER POL OSİLATÖRÜ TASARIMI TEST SONUÇLARI

Tez çalışmasının bu bölümünde, 32-bit IEEE-754-1985 kayan noktalı sayı standardı ve 32-bit IQ-Math sabit noktalı sayı standardı kullanılarak tasarlanan Euler-Tabanlı VDPS 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 ünitelerin verileri işleme süresi, Xilinx ISE Design Tools 14.2 benzetim programı kullanılarak elde edilmiştir.

4.1. IEEE 754-1985 TABANLI VAN DER POL OSİLATÖRÜ TASARIMI TEST SONUÇLARI VE FPGA ÇİP İSTATİSTİKLERİ

Bu bölümde, 32-bit IEEE-754-1985 kayan noktalı sayı standardı kullanılarak tasarlanan Euler-Tabanlı VDPS osilatör ünitesi Xilinx ISE Design Tools 14.2 benzetim programı kullanılarak test edilmiştir. Test dosyası VHDL dili kullanılarak oluşturulmuştur. Tasarımı yapılan ünitelerin verileri işleme süresi, Xilinx ISE Design Tools 14.2 benzetim programı kullanılarak elde edilmiştir. 32-bit IEEE-754-1985 kayan noktalı sayı standardı kullanılarak tasarlanan Euler-Tabanlı VDPS osilatör ünitesi Xilinx ISE Simülöründe elde edilen sonuçlar Şekil 4.1'de görülmektedir. Burada kaotik osilatörün ISE Design Tools kullanılarak FPGA'de gerçekleştirilmesinden elde edilen v ve w sinyallerinin kaotik osilatör ünitesinde karşılıkları olan v_çıkış ve w_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.



Şekil 4.1. Euler-tabanlı VDPS osilatör ünitesi Xilinx ISE Simülöründe sonuçları.

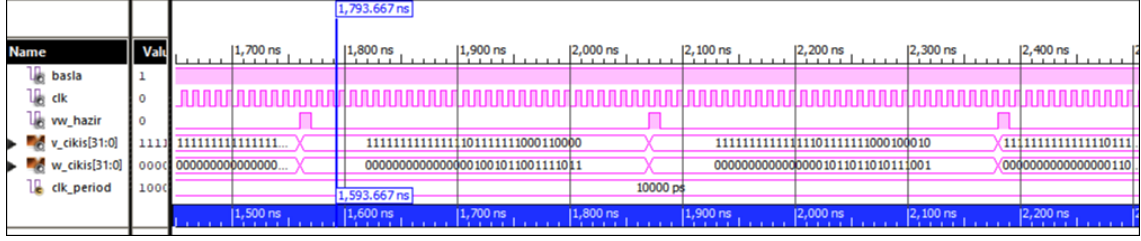
32-bit IEEE-754-1985 kayan noktalı sayı standardı kullanılarak tasarlanan Euler-Tabanlı VDPS osilatör ünitesi Xilinx Virtex-6 ailesi XC6VLX550T-2FF1759 çipi için sentezlenmiştir. Tasarımın Place-Route işleminin ardından elde edilen FPGA çip kaynak kullanımı ve ünitenin saat hızlarına ait parametrelerin istatistikleri incelenmiştir. Place ve Route işleminden elde edilen FPGA çip istatistikleri Çizelge 4.1’de verilmiştir. 32-bit IEEE-754-1985 kayan noktalı sayı standardı kullanılarak tasarlanan FPGA tabanlı Van der Pol osilatör ünitesinin maksimum çalışma frekansı 483,074 MHz olarak elde edilmiştir. Diğer bir deyişle tasarlanan ünitenin minimum çalışma periyodu 2,070 ns’dir.

Çizelge 4.1. IEEE-754-1985 sayı standardı kullanılarak tasarlanan FPGA-tabanlı Van der Pol osilatörü Xilinx Virtex-6 çipi için kullanım istatistikleri.

Sayısal Araçlar	Kullanılan	Kullanılabilir	Kullanım Oranı (%)
Slice Registers Sayısı	4.702	93.120	5
Occupied slices Sayısı	1.326	11.640	11
Slice LUTs Sayısı	4.160	46.560	8
Giriş-Çıkışlar Sayısı	99	360	27

4.2. IQ- MATH TABANLI VAN DER POL OSİLATÖRÜ TASARIMI TEST SONUÇLARI VE FPGA ÇİP İSTATİSTİKLERİ

Tasarımı Bölüm 3.2’de sunulan 32-bit IQ-Math sabit noktalı sayı standardı kullanılarak tasarlanan Euler-Tabanlı VDPS osilatör ünitesi Xilinx ISE Design Tools 14.2 benzetim programı kullanılarak test edilmiştir. Test dosyası VHDL dili kullanılarak oluşturulmuştur. Tasarımı yapılan ünitelerin verileri işleme süresi, Xilinx ISE Design Tools 14.2 benzetim programı kullanılarak elde edilmiştir. 32-bit IQ-Math sabit noktalı sayı standardı kullanılarak tasarlanan Euler-Tabanlı VDPS osilatör ünitesi Xilinx ISE Simülöründe elde edilen sonuçlar Şekil 4.2’de görülmektedir. Burada kaotik osilatörün ISE Design Tools programı kullanılarak FPGA çipi üzerinde gerçekleşmesinden elde edilen v ve w sinyallerinin kaotik osilatör ünitesinde karşılıkları v_çıkış ve w_çıkış sinyalleri olarak gösterilmiştir.



Şekil 4.2. Euler-tabanlı VDPS osilatör ünitesi Xilinx ISE Simülatorü sonuçları.

32-bit IQ-Math sabit noktalı sayı standardı kullanılarak tasarlanan Euler-Tabanlı VDPS osilatör ünitesi Xilinx Virtex-6 ailesi XC6VLX550T-2FF1759 çipi için sentezlenmiştir. Tasarımın Place-Route işleminin ardından elde edilen FPGA çip kaynak kullanımı ve ünitenin saat hızlarına ait parametrelerin istatistikleri incelenmiştir. Place ve Route işleminden elde edilen FPGA çip istatistikleri Çizelge 4.2’de verilmiştir. 32-bit IQ-Math sabit noktalı sayı standardı kullanılarak tasarlanan FPGA tabanlı Van der Pol osilatör ünitesinin maksimum çalışma frekansı 498,728 MHz olarak elde edilmiştir. Diğer bir deyişle tasarlanan ünitenin minimum çalışma periyodu 2,005 ns’dir.

Çizelge 4.2. IQ-Math sayı standardı kullanılarak tasarlanan FPGA-tabanlı Van der Pol osilatörü Xilinx Virtex-6 çipi için kullanım istatistikleri.

Sayısal Araçlar	Kullanılan	Kullanılabilir	Kullanım Oranı (%)
Slice Registers Sayısı	698	93.120	1
Occupied slices Sayısı	237	11.640	2
Slice LUTs Sayısı	549	46.560	1
Giriş-Çıkışlar Sayısı	67	360	18
Slice Registers Sayısı	16	288	5

5. SONUÇLAR VE ÖNERİLER

Sunulan bu tez çalışmasında, Van der Pol osilatörü kaos-tabanlı mühendislik uygulamaları için FPGA yongaları üzerinde çalışmak üzere Euler algoritması kullanılarak gerçekleştirilmiştir. Çalışmada ilk olarak Van der Pol osilatörü nümerik olarak MATLAB programı üzerinde modellenmiş ve zaman serileri elde edilmiştir. Ardından nümerik model referans alınarak VHDL dilinde kodlanmıştır. FPGA-tabanlı Van der Pol osilatörü tasarımlarında 32-bit IEEE-754-1985 standardı ve 32-bit IQ-Math sabit noktalı sayı standardı kullanılmıştır. Tasarımı yapılan FPGA-tabanlı Van der Pol osilatörü Xilinx ISE Project Navigator programı kullanılarak test edilmiştir. 32-bit IEEE-754-1985 standardı ve 32-bit IQ-Math sabit noktalı sayı standardı kullanılarak gerçekleştirilen FPGA-tabanlı Van der Pol osilatörü test sonuçları sunulmuştur. 32-bit IEEE-754-1985 standardı ve 32-bit IQ-Math sabit noktalı sayı standardı kullanılarak tasarlanan FPGA-tabanlı Van der Pol osilatörü ünitesi Xilinx VIRTEX-6 chip ailesi için sentezlenmiştir. Place&Route işleminin ardından 32-bit IEEE-754-1985 standardı kullanılarak tasarlanan FPGA-tabanlı Van der Pol osilatör ünitesinin maksimum çalışma frekansı 483,074 MHz olarak elde edilmiştir. Yine Place&Route işleminin ardından 32-bit IQ-Math sabit noktalı sayı standardı kullanılarak tasarlanan FPGA-tabanlı Van der Pol osilatör ünitesinin maksimum çalışma frekansı 498,728 MHz olarak elde edilmiştir. Elde edilen sonuçlara göre 32-bit IQ-Math sabit noktalı sayı standardı kullanılarak tasarlanan FPGA-tabanlı Van der Pol osilatör ünitesinin maksimum çalışma frekansı 32-bit IEEE-754-1985 standardı kullanılarak tasarlanan FPGA-tabanlı Van der Pol osilatör ünitesinin maksimum çalışma frekansından daha yüksektir. Ayrıca 32-bit IQ-Math sabit noktalı sayı standardı kullanılarak tasarlanan FPGA-tabanlı Van der Pol osilatör ünitesinin FPGA kaynak kullanımı 32-bit IEEE-754-1985 standardı kullanılarak tasarlanan FPGA-tabanlı Van der Pol osilatör ünitesinin FPGA kaynak kullanımından daha azdır. Dolayısıyla Euler algoritması kullanılarak FPGA üzerinde tasarımı yapılan 32 bit sabit noktalı sayı standardı tabanlı Van Der Pol osilatörü ünitesi, 32 bit IEEE-754-1985 kayan nokta sayı standardı tabanlı Van Der Pol osilatör ünitesine göre daha yüksek çalışma frekansı ve daha düşük kaynak kullanımı sağlamıştır. Sonuç olarak FPGA yongaları üzerinde yapılan tasarımlarda IQ-Math sabit noktalı sayı standardı tabanlı tasarımlar IEEE 754-1985 sayı standartlarına göre daha yüksek maksimum

alıřma frekansını ve daha dūřük FPGA yongası kaynak kullanımını sunmaktadır.

Diđer bir deyiřle IQ-Math sabit noktalı sayı standardını aynı bit boyutundaki IEEE 754-1985 sayı standardına gre daha avantajlı ozmler sunmaktadır. İleride yapılacak alıřmalar ile bu alıřmada sunulan FPGA-tabanlı Van der Pol osilatr kullanılarak rasgele sayı reteci tasarımı, sinyal gizleme ve kaotik senkronizasyon uygulamaları gerekleřtirilebilir.



6. KAYNAKLAR

- [1] M. Tuna and C. B. Fidan, "Electronic circuit design, implementation and FPGA-based realization of a new 3D chaotic system with single equilibrium point", *Optik-International Journal for Light and Electron Optics*, c. 127, sayı 24, ss. 11786-11799, 2016.
- [2] T. Belkhouja, A. Mohamed, A. K. Al-Ali, X. Du and M. Guizani, "Light-weight encryption of wireless communication for implantable medical devices using henon chaotic system", *In Wireless Networks and Mobile Communications (WINCOM), IEEE 2017 International Conference on*, Rabat, Morocco, 2017, ss. 1-6.
- [3] X. Wang, A. Akgul, S. Cicek, V. T. Pham and D. V. Hoang, "A chaotic system with two stable equilibrium points: Dynamics, circuit realization and communication application", *International Journal of Bifurcation and Chaos*, c. 27, sayı 08, 2017.
- [4] Y. Ji, M. Zhang, Y. Wang, P. Wang, A. Wang, Y. Wu, et al., "Microwave-photonic sensor for remote water-level monitoring based on chaotic laser", *International Journal of Bifurcation and Chaos*, c. 24, sayı 03, 2014.
- [5] E. Avaroğlu, İ. Koyuncu, A. B. Özer and M. Türk, "Hybrid pseudo-random number generator for cryptographic systems", *Springer, Nonlinear Dynamics*, c. 82, sayı 1-2, ss. 239-248, 2015.
- [6] S. Ashita, G. Uma and P. Deivasundari, "Chaotic dynamics of a zero average dynamics controlled DC-DC Cuk converter", *IET Power Electronics*, c. 7, sayı 2, ss. 289-298, 2014.
- [7] J. Pomares, I. Perea and F. Torres, "Dynamic visual servoing with chaos control for redundant robots", *IEEE/ASME Transactions on Mechatronics*, c. 19, sayı 2, ss. 423-431, 2014.
- [8] M. Alçın, İ. Pehlivan and İ. Koyuncu, "Hardware design and implementation of a novel ANN-based chaotic generator in FPGA", *Elsevier, Optik-International Journal for Light and Electron Optics*, c.127, sayı 13, ss. 5500-5505, 2016.
- [9] Z-G. Wu, P. Shi, H. Su and J. Chu, "Sampled-data fuzzy control of chaotic systems based on a T-S fuzzy model", *IEEE Transactions on Fuzzy Systems*, c. 22, sayı 1, ss. 153-163, 2014.
- [10] E. Avaroğlu, "Pseudorandom number generator based on Arnold cat map and statistical analysis", *Turkish Journal of Electrical Engineering & Computer Sciences*, c. 25, sayı 1, ss. 633-643, 2017.
- [11] C. B. Fidan and M. Tuna, "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*, c. 33, sayı 2, ss. 473-491, 2018.
- [12] K. Shi, Y. Tang, X. Liu and S. Zhong, "Non-fragile sampled-data robust synchronization of uncertain delayed chaotic Lurie systems with randomly occurring controller gain fluctuation", *ISA transactions*, c. 66, ss. 185-199, 2017.

- [13] Ü. Çavuşoğlu, S. Kaçar, İ. Pehlivan and A. Zengin, “Secure image encryption algorithm design using a novel chaos based S-Box”, *Chaos, Solitons Fractals*, c. 95, ss. 92-101, 2017.
- [14] A. A. Heidari, R. A. Abbaspour and A. R. Jordehi, “An efficient chaotic water cycle algorithm for optimization tasks”, *Neural Computing and Applications*, c. 28, s.1, ss. 57-85, 2017.
- [15] İ. Pehlivan, Y. Uyaroğlu, M. A. Yalçın ve A. Ferikoğlu, “Sprott_94_A Kaotik Sisteminin Senkronizasyonu ve Bilgi Gizlemede Kullanılması”, *Uluslararası Katılımlı Bilgi güvenliği ve Kriptoloji Konferansı*, Ankara, Türkiye, 2007.
- [16] M. Tuna, “Kaos ve çift entropi çekirdekli gerçek rasgele sayı üreticinin FPGA tabanlı tasarımı ve gerçekleşmesi”, Doktora tezi, Elektrik – Elektronik Mühendisliği, Fen Bilimleri Enstitüsü, Karabük Üniversitesi, Karabük, Türkiye, 2017.
- [17] K. Murali, M. Lakshmanan and L. O. Chua, “Controlling and synchronization of chaos in the simplest dissipative non-autonomous circuit”, *International Journal of Bifurcation and Chaos*, c. 5, s. 2, ss. 563-571, 1995.
- [18] R. Kılıç, “A practical guide for studying Chua’s circuits”, 71, Chua, L. O., World Scientific, Singapore, ss. 1-44, 2010.
- [19] G. Qi, G. Chen, S. Du, Z. Chen and Z. Yuan, “Analysis of a new chaotic system”, *Physica A: Statistical Mechanics And Its Applications*, c. 352, sayı 2, ss. 295-308, 2005.
- [20] N. Sharma, I. Saini, A. K. Yadav and P. Singh, “Phase-image encryption based on 3d-Lorenz Chaotic System and double random phase encoding”, *3D Research*, c. 8, sayı 4, 2017.
- [21] S. Vaidyanathan and S. Rasappan, “Global chaos synchronization of n-scroll Chua circuit and Lur’e system using backstepping control design with recursive feedback”, *Arabian Journal for Science and Engineering*, c. 39, sayı 4, ss. 3351-3364, 2014.
- [22] İ. Koyuncu, İ. Şahin, C. Gloster ve N. K. Sarıtekin, “A neuron library for rapid realization of artificial neural networks on FPGA: a case study of Rössler chaotic system”, *Journal of Circuits, Systems and Computers*, c. 26, sayı 01, 2017.
- [23] S. H. Yu, H. S. Kang, Y. T. Kim, C. H. Hyun and M. Park, “Fuzzy adaptive modular design of uncertain chaotic duffing oscillators”, *International Journal of Control Automation and Systems*, c. 12, sayı 1, ss. 188-194, 2014.
- [24] S. Jafari, A. Ahmadi, A. J. M. Khalaf, H. R. Abdolmohammadi, V. T. Pham and F. E. Alsaadi, “A new hidden chaotic attractor with extreme multi-stability”, *AEU-International Journal of Electronics and Communications*, c. 89, ss.131-135, 2018.
- [25] M. Tuna, C. B. Fidan, İ. Koyuncu ve İ. Pehlivan, "Real time hardware implementation of the 3D chaotic oscillator which having golden-section equilibria", *IEEE 24 th In Signal Processing and Communication Application Conference (SIU)*, ss. 1309-1312, 2016.
- [26] Z. Zhong, C. Guanrong and Y. Simin, “Hyperchaotic signal generation via DSP for efficient perturbations to liquid mixing”, *International Journal of Circuit Theory and Applications*, cc. 37, sayı 1, ss. 31-41, 2009.

- [27] Z. Yiwei, L. Zexiang and Z. Xinjian, “A chaos-based image encryption ASIC using reconfigurable logic”, *IEEE Asia Pacific Conference on Circuits and Systems*, Macao-China, ss. 1782-1785, 2008.
- [28] S. Kaçar, “Analog circuit and microcontroller based RNG application of a new easy realizable 4D chaotic system”, *Optik*, c. 127, sayı 20, ss. 9551-9561, 2016.
- [29] İ. Koyuncu ve A. T. Özcerit, “The design and realization of a new high speed FPGA-based chaotic true random number generator”, *Elsevier, Computers & Electrical Engineering*, c. 58, ss. 203-214, 2017.
- [30] İ. Koyuncu, M. Alçın ve İ. Pehlivan, “Güvenli haberleşme sistemleri için Sprot 94 S kaotik sisteminin elektronik devre gerçekleştirilmesi ve senkronizasyon uygulaması”, 21st Signal Processing and Communications Applications Conference, Haspolat, Türkiye, 2013.
- [31] R. M. Hidalgo, J. G. Fernandez, R. R. Rivera and H. A. Larrondo, “Versatile DSP-based chaotic communication system”, *Electronics Letters*, c. 37, sayı 19, 2001.
- [32] V. Tavas, S. Özoğuz ve A. Toker, “Yeni Bir Tümlüşük Kaotik Devre”, *12. Elektrik - Elektronik-Bilgisayar-Biyomedikal Mühendisliği Ulusal Kongresi Bildiriler Kitabı*, 2007.
- [33] S. Vaidyanathan, K. Rajagopal, C. K. Volos, I. M. Kyprianidis, I. N. Stouboulos, “Analysis, adaptive control and synchronization of a seven-term novel 3-D chaotic system with three quadratic nonlinearities and its digital implementation in LabVIEW”, *Journal of Engineering Science and Technology Review*, c. 8, sayı 2, ss. 130-141, 2015.
- [34] A. Sevinç, “Lorenz Kaotik Sistemi İçin Adaptif Bir Gözleyici”, *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, c. 18, sayı 4, ss. 57-66, 2003.
- [35] G. Xu, Y. Shekofteh, A. Akgül, C. Li and S. Panahi, “A new chaotic system with a self-excited attractor: entropy measurement, signal encryption, and parameter estimation”, *Entropy*, c. 20, sayı 86, 2018.
- [36] A. Ş. Demirkol, “Kaotik osilatör girişli ADC tabanlı rastgele sayı üretici”, Yüksek lisans tezi, Elektronik ve Haberleşme Mühendisliği, Fen Bilimleri Enstitüsü, İstanbul Teknik Üniversitesi, İstanbul, Türkiye, 2007.
- [37] S. Rajagopalan, S. Rethinam, A. N. Deepika, A. Priyadarshini, M. Jyothirmai and A. Rengarajan, “Design of boolean chaotic oscillator using CMOS technology for true random number generation”, *International conference on Microelectronic Devices, Circuits and Systems*, India, ss. 1-6, 2017.
- [38] F. E. Yardım ve E. AFACAN, “Lorenz-tabanlı diferansiyel kaos kaydırmalı anahtarlama (DCSK) modeli kullanılarak kaotik bir haberleşme sisteminin simülasyonu”, *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, c. 25, sayı 1, ss. 101-110, 2010.
- [39] S. Çiçek, A. Ferikoğlu and İ. Pehlivan, “A new 3D chaotic system: dynamical analysis, electronic circuit design, active control synchronization and chaotic masking communication application”, *Optik-International Journal for Light and Electron Optics*, c. 127, ss. 4024-4030, 2016.
- [40] K. Özdemir, S. Kılınç ve S. Özoğuz, “Sürekli-zamanlı kaos ile rastgele sayı üretici tasarımı (random number generator design using continuous-time chaos)”, *IEEE 16th Signal Processing, Communication and Applications Conference*, 2008.

- [41] R. Chiu, M. M. Gonzalez and D. L. Mancilla, "Implementation of a Chaotic Oscillator into a Simple Microcontroller", *IERI Procedia*, c. 4, ss. 247–252, 2013.
- [42] K. Özdemir, "Sürekli - zamanlı kaos ile rastgele sayı üretici tasarımı", Yüksek lisans tezi, Elektronik ve Haberleşme Mühendisliği, Fen Bilimleri Enstitüsü, İstanbul Teknik Üniversitesi, İstanbul, Türkiye, 2008.
- [43] M. A. Murillo-Escobar, C. Cruz-Hernández, F. Abundiz-Pérez and R. M. López-Gutiérrez, "Implementation of an improved chaotic encryption algorithm for real-time embedded systems by using a 32-bit microcontroller", *Microprocessors and Microsystems*, c. 45, ss. 297-309, 2016.
- [44] F. Pareschi, G. Setti and R. Rovatti, "Implementation and testing of high-speed CMOS true random number generators based on chaotic systems", *IEEE transactions on circuits and systems I: regular papers*, c. 57, ss. 3124-3137, 2010.
- [45] H. P. Ren, C. Bai, Z. Z. Huang and C. Grebogi, "Secure communication based on hyperchaotic Chen system with time-delay", *International Journal of Bifurcation and Chaos*, c. 27, sayı 5, 2017.
- [46] V. Sundarapandian and İ. Pehlivan, "Analysis, control, synchronization, and circuit design of a novel chaotic system", *ScienceDirect, Elsevier, Mathematical and computer modelling*, c. 55, ss. 1904–1915, 2012.
- [47] İ. Pehlivan, "Yeni kaotik sistemler: elektronik devre gerçeklemeleri, senkronizasyon ve güvenli haberleşme uygulamaları", Doktora tezi, Elektronik, Fen Bilimleri Enstitüsü, Sakarya Üniversitesi, Sakarya, Türkiye, 2007.
- [48] İ. Pehlivan and Y. Uyaroğlu, "A new 3D chaotic system with golden proportion equilibria: Analysis and electronic circuit realization", *Computers & Electrical Engineering*, c. 38, sayı 6, ss. 1777-1784, 2012.
- [49] K. Deng, S. Yu, "Estimating ultimate bound and finding topological horseshoe for a new chaotic system", *Optik - International Journal for Light and Electron Optics*, c. 125, sayı 20, ss. 6044-6048, 2014.
- [50] A. Abooe, H. A. Yaghini-Bonabi and M. R. Jahed-Motlagh, "Analysis and circuitry realization of a novel three-dimensional chaotic system", *Communications in Nonlinear Science and Numerical Simulation*, c. 18, sayı 5, ss. 1235-1245, 2013.
- [51] W. Zhou, Y. Xu, H. Lu and L. Pan, "On dynamics analysis of a new chaotic attractor", *Physics Letters, Section A: General, Atomic and Solid State Physics*, c. 372, sayı 36, ss. 5773-5777, 2008.
- [52] İ. Pehlivan, I. M. Moroz and S. Vaidyanathan, "Analysis, synchronization and circuit design of a novel butterfly attractor", *Journal of Sound and Vibration*, c. 333, sayı 20, ss. 5077-5096, 2014.
- [53] A. D. Pano-Azucena, E. Tlelo-Cuautle, J. M. Muñoz-Pacheco and L. G. de la Fraga, "FPGA-based implementation of different families of fractional-order chaotic oscillators applying Grünwald–Letnikov method", *Communications in Nonlinear Science and Numerical Simulation*, c. 72, ss. 516-527, 2019.
- [54] M. Tuna, M. Alçın, İ. Koyuncu, C. B. Fidan and İ. Pehlivan, "High speed FPGA-based chaotic oscillator design.", *Science Direct, Microprocessors and Microsystems*, c. 66, ss. 72-80, 2019.

- [55] K. Rajagopal, A. Karthikeyan and A. Srinivasan, “Dynamical analysis and FPGA implementation of a chaotic oscillator with fractional-order memristor components”, *Nonlinear Dynamics*, c. 91, sayı 3, ss. 1491-1512, 2018.
- [56] T. Bonny and A. S. Elwakil, “FPGA realizations of high-speed switching-type chaotic oscillators using compact VHDL codes”, *Nonlinear Dynamics*, c. 93, sayı 2, ss. 819-833, 2018.
- [57] M. Dursun and E. Kaşifoğlu, “Design and implementation of the FPGA-based chaotic van der pol oscillator”, *International Advanced Researches and Engineering Journal*, c. 2, sayı 3, ss. 309-314, 2018.
- [58] E. Tlelo-Cuautle, A. D. Pano-Azucena, J. J. Rangel-Magdaleno, V. H. Carbajal-Gomez, and G. Rodriguez-Gomez, “Generating a 50-scroll chaotic attractor at 66 MHz by using FPGAs”, *Nonlinear Dynamics*, c. 85, s. 4, ss. 2143-2157, 2016.
- [59] M. F. Tolba, A. M. AbdelAty, L. A. Saida, A. S. Elwakila, A. T. Azara, A. H. Madiana, A. Ounnasf and A. G. Radwana, “FPGA Realization of Caputo and Grünwald-Letnikov Operators”, *6th International Conference on Modern Circuits and Systems Technologies (MOCAS)*, Thessaloniki, Greece, 2017.
- [60] M. Alçın, İ. Pehlivan, İ. Koyuncu, “Hardware design and implementation of a novel ANN-based chaotic generator in FPGA”, *Optik*, c. 127, sayı 13, ss. 5500-5505, 2016.
- [61] M. Tuna, İ. Koyuncu, C. B. Fidan and I. Pehlivan, “Real time implementation of a novel chaotic generator on FPGA”, *IEEE 23rd Signal Processing and Communications Applications Conference*, Malatya, Turkey, ss. 698-701, 2015.
- [62] İ. Koyuncu, A. T. Özcerit and I. Pehlivan, “Implementation of FPGA-based real time novel chaotic oscillator”, *Nonlinear Dynamics*, c. 77, sayı 1-2, ss. 49-59, 2014.
- [63] M. S. Azzaz, C. Tanougast, S. Sadoudi, R. Fellah and A. Dandache, “A new auto-switched chaotic system and its FPGA implementation”, *Communications in Nonlinear Science and Numerical Simulation*, c. 18, sayı 7, ss. 1792–1804, 2013.
- [64] L. Merah, A. Ali-Pacha, N. H. Said and M. Mamat, “Design and FPGA implementation of Lorenz chaotic system for information security issues”, *Applied Mathematical Sciences*, c. 7, sayı 5, ss. 237-246, 2013.
- [65] İ. Koyuncu, A. T. Özcerit and I. Pehlivan, “An analog circuit design and FPGA-based implementation of the Burke-Shaw chaotic system”, *Optoelectronics and Advanced Materials-Rapid Communications*, c. 7, sayı 9, ss. 635-638, 2013.
- [66] L. De Micco and H. A. Larrondo, “FPGA implementation of a chaotic oscillator using RK4 method”, *IEEE VII Southern Conference on Programmable Logic*, Cordoba, Argentina, ss. 185-190, 2011.
- [67] S. Sadoudi, M. S. Azzaz, M. Djeddou and M. Benssalah, “An FPGA real-time implementation of the chen’s chaotic system for securing chaotic communications”, *International Journal Of Nonlinear Science*, c. 7, sayı 4, ss. 467-474, 2009.
- [68] C. Eroğlu, “Implementation of synchronized chaotic systems by field programmable gate array”, M.S. thesis, Electrical and Electronics Engineering, İzmir Institute of Technology, İzmir, Turkey, 2007.

- [69] K. Rajagopal, A. Akgul, S. Jafari, A. Karthikeyan and İ. Koyuncu, “Chaotic chameleon: Dynamic analyses, circuit implementation, FPGA design and fractional-order form with basic analyses”. *Chaos, Solitons & Fractals*, c. 103,ss. 476-487, 2017.
- [70] Q. Lai, X.W. Zhao, K. Rajagopal, G. Xu, A. Akgul and E. Guleryuz, “Dynamic analyses, FPGA implementation and engineering applications of multi-butterfly chaotic attractors generated from generalised Sprott C system”, *Pramana-Journal of Physics*, c. 90, sayı 6, 2018.
- [71] K. Rajagopal, S. Jafari and G. Laarem, “Time-delayed chameleon: Analysis, synchronization and FPGA implementation”, *Pramana-Journal of Physics*, c. 89, sayı 92, 2017.
- [72] L. Jin, J. Mei and L. Li, “Chaos control of parametric driven Duffing oscillators”, *Applied Physics Letters*, c. 104, sayı 13, 2014.
- [73] Anonim, (2019, 2 Temmuz). [Online]. Erişim: <https://www.elektrik.gen.tr/2015/08/fpga-teknolojisi-ve-kullanimi/196>
- [74] M. Taşçı, “FPGA kontrollü robotik göz”, Yüksek lisans tezi, Elektrik-Elektronik Mühendisliği, Fen Bilimleri Enstitüsü Balıkesir Üniversitesi, Balıkesir, Türkiye, 2011.
- [75] O. Çetin, “Yapay sinir ağlarının uyarlanabilir donanımsal yapılarda gerçekleştirilmesi”, Doktora tezi, Elektrik-Elektronik Mühendisliği, Fen Bilimleri Enstitüsü, Sakarya Üniversitesi, , Sakarya, Türkiye, 2014.
- [76] A. N. Akpolat, “FPGA tabanlı nesne algılama”, Yüksek lisans tezi, Mekatronik Mühendisliği, Fen Bilimleri Enstitüsü, Fırat Üniversitesi, Elazığ, Türkiye, 2015.
- [77] H. İ. Şeker, “IQ-Math sayı standartlarında FPGA-tabanlı Kaotik osilatörün tasarımı ve gerçekleştirilmesi”, Yüksek lisans tezi, Elektrik-Elektronik Mühendisliği, Fen Bilimleri Enstitüsü, Afyon Kocatepe Üniversitesi, Afyon, Türkiye, 2019.
- [78] R. Munden, *ASIC and FPGA Verification: A Guide To Component Modeling* , San Francisco, USA: Morgan Kaufmann Publishers, 2005.
- [79] İ. Şahin and İ. Koyuncu, “A new module design for 3D graphic transformations using generated floating-point core units”, *International Revision On Modelling and Simulation*, c.4, sayı 2, ss. 691-698, 2011.
- [80] İ. Koyuncu, “Kriptolojik uygulamalar için FPGA tabanlı yeni kaotik osilatörlerin ve gerçek rasgele sayı üreteçlerinin tasarımı ve gerçekleştirilmesi”, Doktora tezi, Elektrik-Elektronik Mühendisliği, Fen Bilimleri Enstitüsü, Sakarya Üniversitesi, Sakarya, Türkiye, 2014.
- [81] O. Silahtar, “FPGA tabanlı kaos senkronizasyonu için denetleyici tasarımı”, Yüksek lisans tezi, Elektrik-Elektronik Mühendisliği, Fen Bilimleri Enstitüsü, Van Yüzüncü Yıl Üniversitesi, Van, Türkiye, 2018.
- [82] Anonim, (2019, 2 Temmuz). [Online]. Erişim: http://www.fpganedir.com/FPGA/fpga_yapisi.php
- [83] F. A. Suhaib, (2019, 2 Temmuz). [Online]. Erişim: https://www.researchgate.net/figure/Modern-Xilinx-FPGA-architecture-showing-different-basic-components_fig1_282398023.

- [84] H. İ. Eskikurt, Entegre devre tasarımı kitabı, *Ders Notları*, Sakarya Üniversitesi, Teknoloji Fakültesi, 2016.
- [85] S. Vaidyanathan, “Output regulation of the forced Van der Pol chaotic oscillator via adaptive control method”, *International Journal of PharmTech Research*, c. 8, ss. 106–116, 2015.
- [86] I. Kuon, R. Tessier, J. Rose, “FPGA architecture: survey and challenges”, *Foundations and Trends® in Electronic Design Automation*, c. 2, sayı 2, ss. 135–253, 2008.
- [87] E. Sarıtaş, S. Karataş, *Her Yönüyle FPGA ve VHDL*, Ankara, Türkiye: Palme Yayıncılık, 2013.
- [88] İ.A. Özkan, İ Sarıtaş ve S. Herdem, “Manyetik filtreler için FPGA tabanlı bulanık kontrolör tasarımı”, *Selçuk Teknik Dergisi*, sayı 10, ss. 271-284, 2011.
- [89] L. O. Erick, “Fixed-point representation & fractional math”, *Oberstar Consulting, Revision 1.2*, Madison, United States, 2007.
- [90] W. Rüemelin, “Numerical treatment of stochastic differential equations”, *SIAM Journal on Numerical Analysis*, c. 19, ss. 604-613, 1982.
- [91] M. Tuna, İ. Koyuncu and M. Alçın, “Fixed and Floating point-based high-speed chaotic oscillator design with different numerical algorithms on FPGA”, *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, c. 7, ss. 3179-3187, 2018.
- [92] İ. Koyuncu, “Implementation of high speed tangent sigmoid transfer function approximations for artificial neural network applications on FPGA”. *Advances in Electrical and Computer Engineering*, c. 18, ss. 79-87, 2018.
- [93] L. Zhou, C. Wang and L. Zhou, “A novel no-equilibrium hyperchaotic multi-wing system via introducing memristor”, *International Journal of Circuit Theory and Applications*, c. 46, ss. 84-98, 2018.
- [94] A. M. Lyapunov, “The general problem of the stability of motion”, *International Journal of Control*, c. 55, sayı 3, ss. 531-534, 1992.
- [95] D. Yılmaz ve N. F. Güler, “Kaotik zaman serisinin analizi üzerine bir araştırma”, *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, c. 21, sayı 4, ss. 759-779, 2006.
- [96] O. Şahin, “Kriptoloji Uygulamalarında Kullanılacak Bir İşlemcinin Tasarlanarak FPGA Üzerinde Gerçeklenmesi”, Lisans tezi, Elektronik ve Haberleşme Mühendisliği Bölümü, Mühendislik Fakültesi, İstanbul Teknik Üniversitesi, İstanbul, Türkiye, 2012.

7. EKLER

7.1. VIRTEX-6 FPGA ÇİPİ KATALOĞU

FPGA-tabanlı Van Der Pol osilatörünün tasarımı için kullanılan Virtex-6 çipi ailesinin kataloğu aşağıda verilmiştir.



Virtex-6 Family Overview

DS150 (v2.2) January 28, 2010

Advance 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

© 2009–2010 Xilinx, Inc. XILINX, the Xilinx logo, Virtex, Spartan, ISE, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. PCI, PCIe and PCI Express are trademarks of PCI-SIG and used under license. All other trademarks are the property of their respective owners.

Virtex-6 FPGA Feature Summary

Table 1: Virtex-6 FPGA Feature Summary by Device

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

Notes:

1. Each Virtex-6 FPGA slice contains four LUTs and eight flip-flops, only some slices can use their LUTs as distributed RAM or SRLs.
2. Each DSP48E1 slice contains a 25 x 18 multiplier, an adder, and an accumulator.
3. Block RAMs are fundamentally 36 Kbits in size. Each block can also be used as two independent 18 Kb blocks.
4. Each CMT contains two mixed-mode clock managers (MMCM).
5. This table lists individual Ethernet MACs per device.
6. Does not include configuration Bank 0.
7. This number does not include GTX or GTH transceivers.

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	
	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		
	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 160 Mb (2 to 20 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 ISETM 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 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 x 1, 16K x 2, 8K x 4, 4K x 9 (or 8), 2K x 18 (or 16), 1K x 36 (or 32), or 512 x 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 x 1 to 512 x 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 x 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 x 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_{CCO} output supply-voltage pin, which also powers certain input buffers. Some single-ended input buffers require an externally applied reference voltage (V_{REF}). There are two V_{REF} pins per bank (except configuration bank 0). A single bank can have only one V_{REF} voltage value.

I/O Electrical Characteristics

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

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

Digitally Controlled Impedance

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

I/O Logic

Input and Output Delay

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

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

ISERDES and OSERDES

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

System Monitor

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

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

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 9.95 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 (2500 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. 2500 Mb/s support is also available.

Virtex-6 FPGA Ordering Information

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

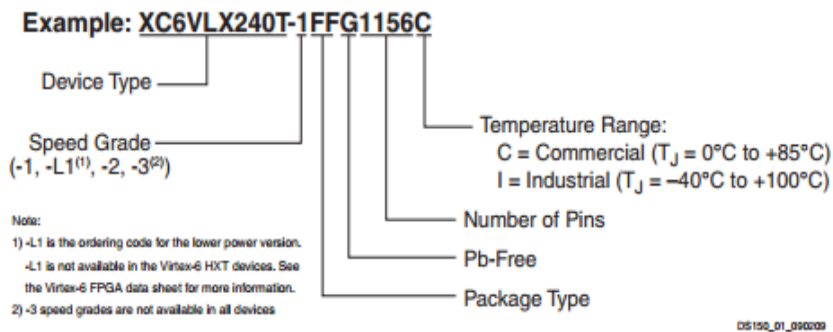


Figure 1: Virtex-6 FPGA Ordering Information

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 (2500 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.

Notice of Disclaimer

THE XILINX HARDWARE FPGA AND CPLD DEVICES REFERRED TO HEREIN ("PRODUCTS") ARE SUBJECT TO THE TERMS AND CONDITIONS OF THE XILINX LIMITED WARRANTY WHICH CAN BE VIEWED AT <http://www.xilinx.com/warranty.htm>. THIS LIMITED WARRANTY DOES NOT EXTEND TO ANY USE OF PRODUCTS IN AN APPLICATION OR ENVIRONMENT THAT IS NOT WITHIN THE SPECIFICATIONS STATED IN THE XILINX DATA SHEET. ALL SPECIFICATIONS ARE SUBJECT TO CHANGE WITHOUT NOTICE. PRODUCTS ARE NOT DESIGNED OR INTENDED TO BE FAIL-SAFE OR FOR USE IN ANY APPLICATION REQUIRING FAIL-SAFE PERFORMANCE, SUCH AS LIFE-SUPPORT OR SAFETY DEVICES OR SYSTEMS, OR ANY OTHER APPLICATION THAT INVOKES THE POTENTIAL RISKS OF DEATH, PERSONAL INJURY, OR PROPERTY OR ENVIRONMENTAL DAMAGE ("CRITICAL APPLICATIONS"). USE OF PRODUCTS IN CRITICAL APPLICATIONS IS AT THE SOLE RISK OF CUSTOMER, SUBJECT TO APPLICABLE LAWS AND REGULATIONS.

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

KİŞİSEL BİLGİLER

Adı Soyadı : Elif KAŞİFOĞLU
Doğum Tarihi ve Yeri : 31/03/1985, Isparta
Yabancı Dili : İngilizce
E-posta : elifkasifoglu@duzce.edu.tr

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Y. Lisans	Elektrik Eğitimi	Düzce Üniversitesi	2019
Lisans	Bilgisayar Sis. Öğrt.	Süleyman Demirel Üniversitesi	2007
Lise		Isparta Gazi Lisesi	2003

MESLEKİ DENEYİM

Ocak 2008 -..... **Öğretim Görevlisi**
Düzce Meslek Yüksekokulu / Düzce Üniversitesi

Uluslararası Makaleler

1. Dursun M., Kaşifoğlu E., “Design and implementation of the FPGA-based chaotic van der pol oscillator”, IAREJ (International Advanced Researches and Engineering Journal), 2018. 2(3), 309-314.