

**KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

FPGA TABANLI DAR BANT BPSK PLC MODEM TASARIMI VE BENZETİMİ

YÜKSEK LİSANS TEZİ

Elektrik - Elektronik Müh. Reyhan SAĞ

**OCAK 2022
TRABZON**



KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ ANABİLİM DALI

FPGA TABANLI DAR BANT BPSK PLC MODEM TASARIMI VE BENZETİMİ

Elektrik-Elektronik Müh. Reyhan SAĞ

Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünde
“ELEKTRONİK YÜKSEK MÜHENDİSİ”
Unvanı Verilmesi İçin Kabul Edilen Tezdir.

Tezin Enstitüye Verildiği Tarih : 03 / 01 / 2022

Tezin Savunma Tarihi : 31 / 01 / 2022

Tez Danışmanı : Prof. Dr. İsmail Hakkı ÇAVDAR

Trabzon 2022

ÖNSÖZ

Karadeniz Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik-Elektronik Mühendisliği Anabilim Dalı, Yüksek Lisans programı kapsamında hazırlanan bu tezde, ikili faz kaydırmalı anahtarlama modülasyonunun FPGA tabanlı benzetim çalışmaları incelenmiştir.

Başta bu tez çalışması olmak üzere lisans ve yüksek lisans öğrenimim boyunca bana yol gösteren tez danışmanım sayın Prof. Dr. İsmail Hakkı ÇAVDAR'a ve KTÜ Elektrik Elektronik Mühendisliği bölümündeki değerli hocalarıma teşekkürlerimi sunarım.

Çalışmam boyunca, her daim beni cesaretlendiren değerli hocam Dr.Öğr.Üyesi Zeynep HASIRCI TUĞCU' ya; başta Arş.Gör. Büşra ÖZGENÇ, Arş .Gör. Besra ÇETİNDERE ve Arş.Gör. Ahmet Yahya BOĞA olmak üzere bütün Elektrik Elektronik Mühendisliği bölüm arkadaşlarıma yanımda oldukları için teşekkür ederim. Hayatımdaki yerini buraya yazdıklarımla sınırlamak istemediğim her zaman motivasyon kaynağım olan sevgili hocam Arş.Gör. Yeşim Aysel BAYSAL'a , dostum Dr. Pınar KELEŞ' e ve yeğenim Zeynep İnci'me sonsuz sevgilerimi sunarım. Son olarak varlıkları ve destekleriyle bana güç veren ve her zaman her koşulda yanımda olan başta annem babam olmak üzere tüm aileme ve yol arkadaşım Nuri ÖRNEK'e teşekkür ederim.

Reyhan SAĞ

Trabzon 2022

TEZ ETİK BEYANNAMESİ

Yüksek Lisans Tezi olarak sunduğum “FPGA TABANLI DAR BANT BPSK PLC MODEM TASARIMI VE BENZETİMİ” başlıklı bu çalışmayı baştan sona kadar danışmanım Prof. Dr. İsmail Hakkı ÇAVDAR’ın sorumluluğunda tamamladığımı, verileri/örnekleri kendim topladığımı, deneyleri/analizleri ilgili laboratuvarlarda yaptığımı/yaptırdığımı, başka kaynaklardan aldığım bilgileri metinde ve kaynakçada eksiksiz olarak gösterdiğimi, çalışma sürecinde bilimsel araştırma ve etik kurallara uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim. 31/01/2022

Reyhan SAĞ

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ.....	III
TEZ ETİK BEYANNAMESİ.....	IV
İÇİNDEKİLER.....	V
ÖZET.....	VIII
SUMMARY	IX
ŞEKİLLER DİZİNİ.....	X
TABLolar DİZİNİ.....	XII
SEMBOLLER VE KISALTMALAR DİZİNİ	XIII
1. GENEL BİLGİLER	1
1.1. Giriş.....	1
1.2. Literatür Araştırması.....	2
1.3. Tezin Amacı ve Hedefleri	7
1.4. Tezin Kapsamı	8
1.5. Haberleşme Teorisi	9
1.6. İkili Faz Kaydırmalı Anahtarlama Yöntemi (BPSK).....	11
1.7. BPSK Modülasyonu.....	11
1.7.1. Dengeli Modülatör	13
1.8. BPSK Demodülasyonu	16
1.8.1. Çarpım Blok Yapısı	17
1.8.2. İntegral Alıcı Blok Yapısı	17
1.8.3. FIR Filtre Yapısı	18
1.8.3.1. FIR Filtre Tasarım Yöntemi.....	20
1.9. Karar Verici Blok Yapısı	20
1.10. Enerji Hattı Üzerinden Haberleşme (PLC)	21
2. KULLANILAN PLATFORM VE YAZILIM	24
2.1. Programlanabilir Mantık Aygıtları (Programmable Logic Devices- PLD)	24
2.1.1. Alanda Programlanabilir Kapı Dizileri	24
2.1.2. FPGA Yapısı	25
2.2. Xilinx Vivado Ortamı	27

2.3.	Donanım Tanımlama Dili (Hardware Description Language- HDL).....	28
2.3.1.	VHDL Tarihçesi.....	28
2.3.2.	VHDL Özellikleri	28
2.3.3.	VHDL Dilinin Yazım Kuralları	29
2.3.4.	VHDL Temel Bileşenleri	30
2.3.4.1.	Kütüphane (Library)	31
2.3.4.2.	Varlık (Entity).....	31
2.3.4.3.	Mimari (Architecture).....	32
2.3.4.4.	Paket (Package).....	32
2.3.4.5.	İşlem (Process).....	32
2.4.	VHDL Veri Nesneleri	33
2.4.1.	Sinyal (Signal) Veri Nesnesi.....	34
2.4.2.	Değişken (Variable) Veri Nesnesi	34
2.4.3.	Sabit (Constant) Veri Nesnesi.....	34
2.5.	VHDL Veri Tipleri	35
2.5.1.	Skaler Tip.....	35
2.5.2.	Sıralı (Listeli) Tip	35
2.5.3.	Sayı Tipi.....	36
2.5.4.	Fiziksel Tip	36
2.5.5.	Bileşik (Kompozit) Tip	36
2.5.6.	Access (Giriş) Tipi ve Dosya Tipi	37
2.5.7.	Ön Tanımlamalı (Synopsys) Tip.....	37
2.6.	VHDL Operatörleri	37
2.6.1.	Mantıksal Operatörler	37
2.6.2.	İlişkisel Operatörler	38
2.6.3.	Aritmetik Operatörler.....	38
2.6.4.	Kayıdırma Operatörleri	39
2.7.	Tip Dönüşümü	39
3.	YAPILAN ÇALIŞMALAR, BULGULAR VE İRDELEME.....	40
3.1.	Giriş.....	40
3.2.	VHDL Tasarım Süreci	41
3.3.	Taşıyıcı Sinüs İşaretinin Elde Edilmesi	43

3.4.	BPSK Modölatör Mimarisi	44
3.5.	BPSK Modölatör Benzetimi	47
3.6.	BPSK Eşevreli Demodölasyon Tekniğı ile Tasarım.....	48
3.6.1.	Giriş.....	48
3.6.2.	BPSK Eşevreli Demodölatör Mimarisi.....	49
3.6.2.1.	Çarpıcı Blok Mimarisi ve Benzetimi	49
3.6.2.2.	FIR Filtre Blok Mimarisi ve Benzetimi	52
3.6.2.3.	Karar Verici Blok Mimarisi ve Benzetimi	56
3.7.	BPSK PLC MODEM Mimarisi	58
3.8.	BPSK PLC MODEM Benzetim Bulguları	60
4.	SONUÇLAR VE TARTIŞMA	66
5.	DEĞERLENDİRME VE ÖNERİLER.....	68
6.	KAYNAKLAR	69
ÖZGEÇMİŞ		

Yüksek Lisans Tezi

ÖZET

FPGA TABANLI DAR BANT BPSK PLC MODEM TASARIMI VE BENZETİMİ

Reyhan SAĞ

Karadeniz Teknik Üniversitesi

Fen Bilimleri Enstitüsü

Elektrik- Elektronik Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. İsmail Hakkı ÇAVDAR

2022, 72 Sayfa

Haberleşme sistemlerinde gelişen teknolojiye göre her seviyede kullanıcıya kesinti olmaksızın cevap verebilecek ekonomik altyapı ihtiyacı günden güne artmaktadır. Bu durum haberleşme teknolojilerinde çeşitli yöntemlerin kullanılmasını ve geliştirilmesini zorunlu kılmaktadır. Enerji hattı üzerinden haberleşme (Power Line Communication-PLC) teknolojisi, günümüzde birçok alanda yaygın olarak kullanılan güçlü haberleşme alternatiflerinden birisidir. PLC kanalı, gösterdiği yüksek zayıflama ve değişken karakteristiği nedeniyle esnek ve adaptif yapıda PLC modem tasarımını gerektirir. Bu nedenle PLC modem tasarımı için esnek yapıya sahip kapı dizileri, kısa tasarım süresi, kullanıcı tarafından yeniden programlanabilme özelliklerine sahip FPGA (Field Programmable Gate Array) tabanlı sistemler düşük maliyetli, esnek ve pratik bir çözüm olarak düşünülmektedir.

Bu tez çalışmasında; ikili faz kaydırmalı anahtarlama (Binary Phase Shift Keying-BPSK) modülasyon tekniği ile CENELEC standardında yer alan dar banda göre 100 kHz frekanslı taşıyıcıya sahip, esnek yapıya sahip BPSK-PLC modem tasarımı yapılmıştır. BPSK modem tasarımı için benzetim çalışmaları, önce MATLAB® ortamında daha sonra ise Xilinx destekli Vivado ortamında VHDL (Very High Speed Integrated Circuit Hardware Description Language) yazılım dili kullanılarak gerçekleştirilmiştir. Sonuçlar, oluşturulan FPGA tabanlı BPSK-PLC modemin, PLC iletişim sistemine uyumlu olduğunu göstermiştir.

Anahtar Kelimeler: BPSK, Enerji Hatları Üzerinden Haberleşme, PLC Modem, FPGA, MATLAB®, Vivado, VHDL

Master Thesis

SUMMARY

FPGA BASED NARROW BAND BPSK PLC MODEM DESIGN AND SIMULATION

Reyhan SAĞ

Karadeniz Technical University
The Graduate School of Natural and Applied Sciences
Electrical and Electronics Engineering Graduate Program
Supervisor: Prof. Dr. İsmail Hakkı ÇAVDAR
2022, 72 Pages

According to the developing technology in communication systems, the need for economic infrastructure that can respond to users at all levels without interruption is increasing day by day. This situation necessitates the use and development of various methods in communication technologies. Communication over the power line (PLC) technology is one of the powerful communication alternatives that is widely used in many fields today. PLC channel requires flexible and adaptive PLC modem design due to its high attenuation and variable characteristics. For this reason, FPGA (Field Programmable Gate Array) based systems with flexible structured gate arrays, short design time, and user reprogrammability are considered as a low-cost, flexible and practical solution for PLC modem design.

In this thesis study; With the binary phase shift keying (BPSK) modulation technique, a flexible BPSK-PLC modem design with 100 kHz frequency carrier according to the narrow band in the CENELEC standard has been made. Simulation studies for BPSK modem design were carried out first in the MATLAB® environment and then in the Xilinx supported Vivado environment using the VHDL (Very High Speed Integrated Circuit Hardware Description Language) software language. The results showed that the created FPGA-based BPSK-PLC modem is compatible with the PLC communication system.

Keywords: BPSK, Communication over Power Lines, PLC Modem, FPGA, MATLAB® Vivado, VHDL.

ŞEKİLLER DİZİNİ

	<u>Sayfa No</u>
Şekil 1. Sayısal bir haberleşme sisteminin genel görünümü	9
Şekil 2. BPSK modülasyonu anahtarlamaşı	12
Şekil 3. BPSK modülatör blok yapısı.....	12
Şekil 4. Örnek bir BPSK modüleli işaret	13
Şekil 5. Denge modülatörü çalışma ilkesi	14
Şekil 6. Denge modülatörü pozitif çevrimi	15
Şekil 7. Denge modülatörü negatif çevrimi	15
Şekil 8. Eşevreli BPSK demodülatör blok yapısı.....	16
Şekil 9. Örnek bir çarpıcı çıkışı.....	17
Şekil 10. Genel FIR filtre yapısı.....	19
Şekil 11. Pencere tasarım yöntemi default görünümü.....	20
Şekil 12. Örnek olarak tekrar elde edilen bilgi işareti.....	21
Şekil 13. ITU frekans bantları	22
Şekil 14. FPGA yapısı	25
Şekil 15. FPGA mimarisi	26
Şekil 16. Vivado ara yüzü ekranı	27
Şekil 17. VHDL bileşenleri.....	30
Şekil 18. Sayısal haberleşme sistemi blok yapısı	40
Şekil 19. VHDL tasarım süreci	42
Şekil 20. Vivado ayarları.....	42
Şekil 21. BPSK modülasyonu için elde edilen sinüs işaretleri	44
Şekil 22. BPSK modülatörde process sıralı işlemi akış diyagramı	45
Şekil 23. FPGA tabanlı BPSK modülatör schematic görünümü.....	47
Şekil 24. Örnek bir BPSK modülatör benzetim sonucu.....	48
Şekil 25. BPSK çarpıcıda process sıralı işlemi akış diyagramı.....	50
Şekil 26. FPGA tabanlı çarpıcı blok schematic görünümü	51
Şekil 27. Örnek bir çarpıcı blok benzetim sonucu	51
Şekil 28. FPGA tabanlı FIR filtre çalışma prensibi	52

Şekil 29.	BPSK FIR filtrede process sıralı işlemi akış diyagramı	53
Şekil 30.	FPGA tabanlı FIR filtre blok schematic görünümü	55
Şekil 31.	Örnek bir FIR filtre benzetim sonucu	55
Şekil 32.	BPSK karar vericide process sıralı işlemi akış diyagramı	56
Şekil 33.	FPGA tabanlı karar verici blok schematic görünümü.....	57
Şekil 34.	Örnek bir karar verici blok benzetim sonucu	58
Şekil 35.	FPGA tabanlı BPSK modulator schematic görünümü.....	59
Şekil 36.	FPGA tabanlı BPSK demodulator schematic görünümü	59
Şekil 37.	FPGA yerleşiminde bazı modüllerin kullanım yüzdeleri	60
Şekil 38.	FPGA yerleşiminde harcanan güç yüzdeleri.....	60
Şekil 39.	rising_edge_clk = '1' , trailing_edge_clk = '0' ve reset = '0' olduğunda, giriş_veri:= [10110100101101001011010010110100] için benzetim sonucu	62
Şekil 40.	rising_edge_clk = '1' , trailing_edge_clk = '0' ve reset = '1' olduğunda, giriş_veri:= [10110100101101001011010010110100] için benzetim sonucu	62
Şekil 41.	rising_edge_clk = '1' , trailing_edge_clk = '0' ve reset = '0' olduğunda, giriş_veri:= [00100111001001110010011100100111] için benzetim sonucu	63
Şekil 42.	rising_edge_clk = '1' , trailing_edge_clk = '0' ve reset = '0' olduğunda, giriş_veri:= [11001010110010101100101011001010] için benzetim sonucu	64
Şekil 43.	rising_edge_clk = '0' , trailing_edge_clk = '1' ve reset = '0' olduğunda, giriş_veri:= [11001010110010101100101011001010] için benzetim sonucu	64
Şekil 44.	rising_edge_clk = '1' , trailing_edge_clk = '0' ve reset = '0' olduğunda, giriş_veri:= [11001010110010101100101011001010] için benzetim sonucu	65

TABLÖLAR DİZİNİ

	<u>Sayfa No</u>
Tablo 1. VHDL ile Verilog karşılaştırılması.....	29
Tablo 2 Port çeşitleri	31
Tablo 3. VHDL yapıları ve kullanıldıkları alanlar	33
Tablo 4. Modülatör yapısında kullanılan portlar ve özellikleri.....	46
Tablo 5. Çarpıcı blokta kullanılan portlar ve özellikleri	49
Tablo 6. FIR filtrede kullanılan sabitler ve özellikleri	53
Tablo 7. FIR filtrede kullanılan portlar ve özellikleri	54
Tablo 8. Karar verici yapıda kullanılan portlar ve özellikleri	56
Tablo 9. Modem bloklarında kullanılan hücre ve bağlantı sayıları.....	59
Tablo 10. Benzetim sonuçları için oluşturulan senaryolar	61

SEMBOLLER VE KISALTMALAR DİZİNİ

A	Taşıyıcı İşaretin Genliği
ADC	Analog Digital Converter (Analog Sayısal Dönüştürücü)
AM	Amplitude Modulation (Genlik Modülasyonu)
ARIB	Association of Radio Industries and Businesses (Radyo Endüstrileri ve İşletmeler Birliği)
ASIC	Application Specific Integrated Circuit (Uygulamaya Özel Tümleşik Devre)
ASK	Amplitude Shift Keying (Genlik Kaydırmalı Anahtarlama)
b_N	FIR Filtre Katsayıları
BER	Bit Error Rate (Bit Hata Oranı)
BFSK	Binary Frequency Shift Keying (İkili Frekans Kaydırmalı Anahtarlama)
BPSK	Binary Phase Shift Keying (İkili Faz Kaydırmalı Anahtarlama)
CENELEC	European Committee for Electrotechnical Standardization (Avrupa Elektroteknik Standartlaştırma Komitesi)
CLB	Configurable Logic Block (Konfigüre Lojik Blok)
CPLD	Complex Programmable Logic Device (Karmaşık Programlanabilir Mantık Aygıtı)
DAC	Digital Analog Converter (Sayısal Analog Dönüştürücü)
DDS	Direct Digital Synthesis (Doğrudan Dijital Sentez)
DPSK	Differential Phase Shift Keying (Diferansiyel Faz Kaydırmalı Anahtarlama)
DSP	Digital Signal Processing (Sayısal İşaret İşleme)
EHF	Extremely High Frequency (Aşırı Yüksek Frekans)
f_t	Taşıyıcı İşaretin Frekansı
FCC	Federal Communications Commission (Federal İletişim Komisyonu)
FF	Flip Flop
FIR	Finite Impulse Response (Sınırlı Sinyal Tepkisi)
FPGA	Field Programmable Gate Array (Alanda Programlanabilir Kapı Dizileri)
FSK	Frequency Shift Keying (Frekans Kaydırmalı Anahtarlama)
GHz	Gigahertz
GPP	General Purpose Processors (Genel Amaçlı İşlemciler)

HDL	Hardware Description Language (Donanım Tanımlama Dili)
HF	High Frequency (Yüksek Frekans)
Hz	Hertz
IEEE	The Institute of Electrical and Electronics Engineers (Elektrik ve Elektronik Mühendisleri Enstitüsü)
I/O	Input/Output (Giriş/Çıkış)
ITU	International Telecommunication Union (Uluslararası Telekomünikasyon Birliği)
kHz	Kilohertz
LF	Low Frequency (Düşük Frekans)
LFSR	Linear Feedback Shift Register (Doğrusal Geri Beslemeli Kaydırma Yazmacı)
LUT	Look Up Table (Başvuru Çizelgesi)
Mpbs	Mega Bits Per Second
MF	Medim Frequency (Orta Frekans)
M-FSK	Multilevel Frequency Shift Keying (Çoklu Frekans Kaydırmalı Anahtarlama)
MHz	Megahertz
M-PSK	Multilevel Phase Shift Keying (Çok seviyeli Faz Kaydırmalı Anahtarlama)
MUX	Multiplexer (Çoğullayıcı)
N	FIR Filtrenin Derecesi
OFDM	Orthogonal Frequency Division Multiplexing (Dikgen Frekans Bölmeli Çoğullama)
PLC	Power Line Communication (Güç Hattı Haberleşmesi)
PLD	Programmable Logic Devices (Programlanabilir Mantık Aygıtları)
PLL	Phase Locked Loop (Faz Kilitli Çevrim)
PSK	Phase Shift Keying (Faz Kaydırmalı Anahtarlama)
PWM	Pulse Width Modulation (Darbe Genişlik Modülasyonu)
QAM	Quadrature Amplitude Modulation (Dörtlü Genlik Modülasyonu)
QPSK	Quadrature Phase Shift Keying (Dikgen Faz Kaydırmalı Anahtarlama)
RTL	Register Transfer Language (Kaydedici Transfer Seviyesi)

$s_i(t)$	Taşıyıcı İşaret
SDR	Software Defined Radio (Yazılım Tabanlı Radyo)
SHF	Super High Frequency (Süper Yüksek Frekans)
SLF	Super Low Frequency (Süper Düşük Frekans)
SNR	Signal Noise Ratio (İşaret Gürültü Oranı)
SPLD	Simple Programmable Logic Device (Basit Programlanabilir Mantık Aygıtı)
THF	Terahertz High Frequency
THz	Terahertz
ULF	Ultra Low Frequency (Ultra Düşük Frekans)
UHF	Ultra High Frequency (Ultra Yüksek Frekans)
VHDL	Very High Speed Integrated Circuit Hardware Description Language (Yüksek Hızlı Tümleşik Devre Donanım Tanımlama Dili)
VHF	Very High Frequency (Çok Yüksek Frekans)
VLF	Very Low Frequency (Çok Düşük Frekans)
$x(n)$	FIR Filtre Girişi
$y(n)$	FIR Filtre Çıkışı
ϕ_1, ϕ_0	Taşıyıcı İşaretin Frekansı için Belirlenen Çıkış Fazları
μW	Mikrowatt

1. GENEL BİLGİLER

1.1. Giriş

İnsanlığın varoluşuna dayanan haberleşme, ilk başlarda insanlar arası temel bir ihtiyaç olarak ortaya çıksa da bugün makine-makine, insan-makine arası iletişimi de kapsayan geniş bir kavramı temsil etmektedir. Haberleşmenin mühendislikteki karşılığı ise alıcı ve verici arasında bir iletişim ortamı oluşturup, burada bilgiyi doğru, hızlı ve etkin bir biçimde iletmek ya da almak olarak tanımlanmaktadır. Bilgi çağı olarak adlandırılan günümüzde telefon, televizyon, bilgisayar gibi iletişim araçlarının hayatımızın temel parçaları konumuna gelmesi haberleşmeyi daha da önemli kılmaktadır. Haberleşme sistemlerinde gelişen teknolojiye göre her seviyede kullanıcıya kesinti olmaksızın cevap verebilecek ekonomik altyapı ihtiyacı günden güne artmaktadır. Bu durum haberleşme teknolojilerinde çeşitli yöntemlerin kullanılmasını ve geliştirilmesini gerekli kılmıştır.

Enerji hatlarını veri iletiminde kullanmak yani enerji hattı haberleşmesi (Power Line Communication-PLC) bu yöntemlerden biri olarak karşımıza çıkmaktadır. PLC haberleşmesi, yaygın kullanılan haberleşme alternatiflerinden olup mevcut güç hattı üzerinden veri iletimi yapılması ilkesine dayanan bir teknolojidir [1]. Özellikle son yıllarda, akıllı şebeke kapsamındaki çalışmaların hız kazanması ile PLC alanındaki araştırmalar farklı bir boyut kazanarak dikkat çekici hale gelmiştir [2]. Şebekedeki yükler nedeniyle kanal empedansının sürekli değişir ve enerji hattı yüksek zayıflama özelliği gösterir. Bu yüzden enerji hatları üzerinden veri iletişimi yapılırken modülasyon tekniklerine, haberleşme işaretinin gücüne, protokollere, empedans karakteristiklerine ve filtreleme tekniklerine dikkat edilmelidir.

FPGA, güçlü bir simülasyon ve yazılım desteği sunması, sahada programlanabilmesi, tasarımın test edilip doğrulanabilmesi, daha fazla örnekleme oranı sağlaması, daha kararlı olması, paralel işlem yapabilme yeteneği ve giriş/çıkış sayısının fazla olması gibi özellikleri nedeniyle avantajlıdır. Bununla birlikte FPGA'ların, yüksek yoğunluklu ve esnek yapılı kapı dizileri, kısa tasarım süresi, kullanıcı tarafından programlanabilir özellikleri ve standart ürün özelliği olan programlanabilir lojik elemanların avantajlarını birleştiren bir yapıda olması diğer özellikleri olarak verilebilir.

1.2. Literatür Araştırması

Güncel literatür taraması yapıldığında, FPGA tabanlı modemlerin haberleşme sistemlerinde yaygın olarak kullanıldığı görülmektedir. Verilen kaynaklarda, FPGA tabanlı çalışmalar için kullanılan kart, donanım tanımlama dili ve yazılım paketi gibi unsurların farklılık gösterdiği tasarımlar yer almaktadır.

[3] Numaralı çalışmada, bir BPSK modülatörü sırasıyla; MATLAB/Simulink ortamında blok yapıları kullanılarak, MATLAB ara yüzünde kod yazılarak ve System Generator ile üç farklı senaryo uygulanarak tasarlanmıştır. System Generator 'da birinci durumda taşıyıcı, modüle edici ve harici olarak üretilen modüle edilmiş sinyaller ile; ikinci durumda harici olarak üretilen taşıyıcı ve LFSR tarafından dahili olarak üretilen modüle edici sinyaller ile; üçüncü durumda ise tümü dahili olarak üretilen üç sinyal ile tasarlanan modülatör yapıları önerilmiştir. Son olarak Spartan 3E Starter FPGA kartı ile önerilen üçüncü durumun uygulaması gerçekleştirilmiştir.

[4] Numaralı diğer bir çalışmada, iletişim sistemlerinde yaygın olarak kullanılan iki modülasyon türü olan BPSK ve QPSK modülasyon tekniklerinin Spartan 3E Starter FPGA kartı ile uygulaması gerçekleştirilmiş ve iki tekniğin performansı karşılaştırılmıştır. Aynı iletim hızında aynı bit hata oranında haberleşme yapılırken, QPSK tekniğinin kullanılan bant genişliğini yarıya düşürmesinden dolayı BPSK' ya göre daha uygun bir seçenek olduğu anlaşılmıştır. Ancak az miktarda veri iletimi istendiğinde BPSK tekniğinin de tercih edilebilir bir yöntem olduğu sonucuna varılmıştır.

[5] Numaralı çalışmada, ikili genlik kaydırmalı anahtarlama (BASK), ikili frekans kaydırmalı anahtarlama (BFSK) ve ikili faz kaydırmalı anahtarlama (BPSK) sayısal modülatörlerinin sahada programlanabilir kapı dizisi (FPGA) uygulamaları sunulmuştur. Tasarlanan BASK, BFSK ve BPSK sayısal modülatörlerinin işlevselliği, Quartus II yazılımı kullanılarak yapılan simülasyonlar ve Altera DE2-115 Geliştirme ve Eğitim Kiti üzerinde gerçekleştirilen modüle edilmiş sinyalin deneysel ölçümler aracılığıyla gösterilmiştir. Modelde, giriş taşıyıcı sinyali ve bit akışı kullanıcı tarafından kontrol edilebilir olarak ayarlanmıştır. Sonuçta gerçekleştirilen FPGA tasarımlarının, yazılım tabanlı radyo sistemlerinde temel bantta modülasyon yapmak için uygun olduğu görülmüştür.

[6] Numaralı çalışmada, Xilinx System Generator 'da oluşturulan ikili genlik kaydırmalı anahtarlama (BASK), ikili frekans kaydırmalı anahtarlama (BFSK), ikili faz kaydırmalı anahtarlama (BPSK) ve diferansiyel faz kaydırmalı anahtarlama (DPSK) sayısal

modölatörlerinin Nexys2 Spartan 3E FPGA kartında uygulamaları gerçekleştirilmiştir. Modölatörler, tasarıma en az sayısal blok sayısı dahil olacak ve kullanıcı tarafında giriş frekansı kontrol edilebilir olacak şekilde tasarlanmıştır. Elde edilen dalga şekilleri Xilinx ISE tasarım paketi 14.7 ile sistem üreticinde görüntülenmiştir.

[7] Numaralı çalışmada, uydu telemetri sistemi için model tabanlı programlama kullanılarak FPGA tabanlı BPSK demodölatörünün tasarımı önerilmiştir. Sistem performansı için önce FPGA üzerinde bir modölasyon işlemi yapılmış ve bu modüleli işaret, başka bir FPGA kartına giriş olarak verilerek demodüle edilmesi sağlanmıştır. Uygulama için Zynq 7000 FPGA tabanlı ZED kartı kullanılmıştır. Önerilen BPSK demodölatörünün %99,7'lik bir doğrulukla bilgi işaretlerini yeniden elde ettiği gözlemlenmiştir.

[8] Numaralı çalışmada, BPSK tekniğine dayalı bir iletişimin etkin bir fiziksel katmanını oluşturmak için sembol senkronizasyon süreçleriyle uyumlu bir demodölasyon tasarlanmasının mümkün olduğu görülmüştür. Teorik çalışma ve simölasyon için MATLAB, uygulama için Spartan 3 geliştirme kiti ve yazılım aracı Xilinx ISE 9.2'i kullanılmıştır. Gürültünün fazla olduğu ortamlarda, uygun giriş işareti ve korelatör tasarımı ile haberleşme verimliliğinin artması sağlanmıştır.

[9] Numaralı çalışmada, su altı haberleşmesi için bir BPSK akustik modem geliştirilmiştir. Simölasyon için MATLAB ortamı, donanımı gerçeklemek için Xilinx Virtex5 FPGA kartı kullanılmıştır. Modemin performansını değerlendirmek için yapılan simölasyon ve deneysel çalışmalar sonuçlar BSPK akustik modemin doğru çalıştığını göstermektedir.

[10] Numaralı çalışmada, dijital sinyal işleme (DSP) tabanlı BPSK vericisi geliştirilmiş ve Verilog donanım tanımlama diline (HDL) göre derlenmiştir, 5 Mbps giriş biti için bit başına taşıyıcı işaretin periyodu denk gelecek şekilde tasarlanmıştır.

[11] Numaralı çalışmada, sayısal modölasyon teknikleri kullanılarak MATLAB/Simulink ortamında geliştirilen modölatörlerin, FPGA uygulaması için Xilinx13.1 aracılığıyla Verilog' da önerilen tasarımları anlatılmıştır. FPGA yapısına gömülen ARM ile işlevselliği yapısal olarak değiştirilmiş ve zaman, maliyet, alan gibi unsurlar azaltılmıştır. Yeniden yapılandırmayı gözlemlemek için ASK,FSK,BPSK gibi farklı sayısal modölasyon uygulamaları seçilerek en iyi SNR değeri elde edilmiştir.

[12] Numaralı diğer bir çalışmada, uydu telemetri bağlantısı için model tabanlı programlama kullanılarak FPGA tabanlı bir BPSK demodölatörünün üç farklı algoritmaya göre tasarımı ve değerlendirilmesi önerilmiştir. Tasarımın performansı, modellerin Zynq

7000 FPGA tabanlı ZED kartı üzerindeki donanım uygulamasıyla elde edilmiştir. Taşıyıcının farklı frekans değerleri çeşitli veri hızları ve kullanılan üç farklı yaklaşım için bulunan sonuçlar bir tabloda verilmiştir.

[13] Numaralı çalışmada, bir nano uydu için FPGA tabanlı SDR tipi iletişim sisteminin tasarımı ve donanım uygulaması açıklanmıştır. Sistemin donanımsal değerlendirmesi için yüksek hızlı analog ek kartlı Virtex-6 FPGA kullanılmıştır. SDR tabanlı sistemlerin önemli yönlerinden biri olan çeşitli yeniden yapılandırılabilirlik seçenekleri de gösterilmiştir.

[14] Numaralı çalışmada, ters adres tekniği ile akümülatörler kullanılarak BPSK ve QPSK modülasyonunun bir uygulaması sunulmuştur. BPSK modülatörü için, tek bir Look Up Table (LUT) kullanılarak ve herhangi bir çarpma talimatı olmadan 180 derecelik faz kaymasına sahip iki sinüzoidal sinyal üretilmiştir. İlk sinyal, DDS tekniğine dayalı LUT kullanılarak oluşturulmuştur ikinci sinyal ise aynı LUT kullanılarak; fakat akümülatörün en önemli bitini ters çevirdikten sonra elde edilmiştir. QPSK modülatörü için, yalnızca bir LUT kullanılarak aralarında 90 derece faz farkı olan dört sinüzoidal dalga üretilmiştir. İlk iki dalga, saatin yükselen kenarında ve düşen kenarında çalışan iki akümülatör kullanılarak; diğer dalgalar ise, en anlamlı biti tersine çevirdikten sonra ilk iki akümülatörden oluşturulmuştur. Tüm sistemlerin uygulaması Xilinx System Generator veya DSP Builder araçlarının yardımı olmadan VHDL ile yapılmıştır.

[15] Numaralı çalışmada, hibrit bir yazılım radyo platformu kullanan bir kablosuz sayısal alıcı-vericinin tasarımı ve uygulaması sunulmuştur. General Purpose Processors (GPP), DSP, FPGA hibrit mimarisinin, yazılım tanımlı radyo teknolojisi için uygun bir çözüm olduğu görülmüştür. BPSK alıcısının FPGA uygulaması için çeşitli tasarım parametreleri tartışılmıştır.

[16] Numaralı çalışmada Spartan 3E başlangıç kiti üzerinde tasarlanan BPSK sistemi, Modelsim kullanılarak yeniden oluşturulmuştur. System Generator 'da VHDL kodunu üretme seçeneği olmasına rağmen, bu yapı için oluşturulan kodun okunması zor olduğu için başlangıç kullanılarak üretilmiştir.

[17] Numaralı çalışmada, sayısal sinyallerin donanım tanımlama dili (VHDL) kullanılarak tasarlanan verici, alıcı ve geciktirici gibi seri iletişim modüllerinin algoritması, sayısal filtre yapımı, modül örnekleme, filtre mimarilerinin farklılıkları ve tasarım süreci gibi konular anlatılmış ve sonuçlar karşılaştırılmıştır. Ayrıca tasarım sürecinde gerekli olan çeşitli yazılımların nasıl kullanılacağından bahsedilmiştir. FPGA'da oluşturulan modülleri

test etmek için bilgisayardan alınan sayısal verilerin iletimi FPGA'nın seri portu ile yapılmıştır.

[18] Numaralı çalışmada, SDR (yazılım tanımlı radyo) teknikleri kullanılarak FPGA'da bir AM (genlik modüleli) alıcısı tasarlanmış ve uygulanmıştır. SDR eğitimi için FPGA tabanlı ucuz ve basit bir ortam sağlamak amaçlanmıştır. FPGA simülasyonu ve uygulamasında kullanılan test sinyali, MATLAB'de genlik modülasyonu tekniği ile elde edilmiştir. Sonuçlar, tasarlanan ve uygulanan FPGA tabanlı AM alıcısının sinyallerin demodülasyonunda başarılı olduğunu göstermiştir.

[19] Numaralı çalışmada, PLC kanalında, düşük SNR değerinde yüksek bit hata oranında (Bit Error Rate, BER) veri iletimi sağlayabilmesi amacı ile modülasyon yöntemi olarak BPSK tercih edilmiştir. Tasarlanan modem verici ve alıcı olarak FPGA ara yüzünde benzetimi yapılmıştır. FPGA'nın paralel işlem gücünün yüksek olması ve esnek yapılı kapı dizilerine sahip olması bu çalışmada tercih edilme sebebidir.

[20] Numaralı PLC üzerine yapılmış bir diğer çalışma ise, PLC teknolojisinin gerekleri ve geliştirilmesi ile ilgili trendlerin derlendiği bir araştırma makalesidir. Burada, PLC kanaldaki yayılma modelleri, çok yolluluk etkileri, kanal ve gürültü etkileri detaylı olarak incelenmiş olup herhangi bir deneysel analize yer verilmemiştir.

[21] Numaralı BPSK modülasyonu kullanılarak yapılan bir diğer çalışmada ise Nakagami-m gürültüsü ile PLC sistemin performans analizi gerçekleştirilmiştir. Yapılan çalışmada teorik olarak BER analizi gerçekleştirilmiş ve Maximum Likelihood detektörü kullanılmıştır. Ayrıca Monte Carlo benzetim yöntemi kullanılarak sayısal BER analizi gerçekleştirilmiştir. Ancak bu çalışmada deneysel analiz bulunmamaktadır.

[22] Numaralı diğer bir deneysel bir çalışmada ise PLC kanalındaki veri iletimi Digital Signal Processor (DSP) kullanılarak gerçekleştirilmiştir. Bu çalışmada BPSK modülasyonu kullanılarak veri iletimi gerçekleştirilmiş ancak ulaşılan veri hızından bahsedilmemiştir. Bunun yerine DSP'de kullanılan hafıza alanı optimize edilmeye çalışılmıştır.

[23] Numaralı tez çalışmasında, akıllı şebekeler için PLC dar bandında tasarlanan bir PLC modem anlatılmıştır. Bu modem tasarımı Texas Instruments tarafından üretilen bir Analog Front End (AFE03x) Chip kullanılmış ve referans tasarım esas alınarak bir PLC modem tasarlanmıştır. Akıllı ev uygulaması için farklı elektrik yükleri altında veri hızları, BER ve SNR oranları çizdirilmiştir. Empedans bara sistemleri ve güç hattı sonuçları PLC'nin uygunluğunu göstermiştir.

[24] Nolu çalışmada, güç hattı iletişimi (PLC) için tam entegre bir dijital modülatör/demodülatör tasarımı sunulmuştur. Önerilen tasarımda darbe genişliği modülasyonu (PWM) ve frekans kaydırmalı anahtarlama (FSK) modülasyonu yöntemleri kullanılmıştır. Burada PWM'nin işlevi, giriş voltajı salınımına göre aynı darbe genişliğini sergileyen sayısal darbeleri üretmektir. Haberleşmede FSK, sayısal bilginin taşıyıcının ayrı frekans değişiklikleri yoluyla iletilebildiği bir frekans modülasyon şemasıdır. Öğretme bloğunun detaylı tasarımları açıklanmıştır. Ölçülen veriler, önerilen tasarımın tamamen işlevsel olduğunu ve $155\mu W$ tükettiğini göstermiştir.

[25] Nolu çalışmada, düşük sinyal bit hızı gerektiren tüketici noktasına yakın alçak gerilim dağıtım sisteminde karmaşık programlanabilir mantık aygıtı (CPLD) tabanlı dar bant güç hattı haberleşmesi (NB-PLC) üzerinde yoğunlaşmıştır. Modemi tasarlamak için analog sistemle birlikte dijital sinyal işleme tekniği de dahil edilmiştir. CPLD, iletişim sistemindeki modemi etik olmayan bilgisayar korsanlarından korumak için tasarlandığında yeniden yapılandırılabilirliği nedeniyle kullanılmıştır. Burada, frekans kaydırmalı anahtarlama (FSK) modülasyon yöntemini uygulayan CPLD tabanlı bir modem tasarlanmıştır.

[26] Nolu çalışmada, Lite-PLC adlı güç hattı iletişimi için yeni bir fiziksel katmana genel bir bakış ve tasarlanmış PLC modemin beklenen performansını gösteren benzetim ve laboratuvar test sonuçları sunulmaktadır. Veri iletimi için OFDM kullanılmış ve ARIB bant planında 100 - 450 kHz'lik düşük frekansta çalışması amaçlanmıştır. Burada, güç hattı kanalında 500 kHz'in altındaki düşük frekans bölgesinde iletim için uygun olan bir paket vardır. Simülasyon ve FPGA laboratuvar test sonuçları, beklenen performansı gösterebilmiştir.

[27] Nolu çalışmada, güvenilir ve uygun maliyetli veri iletişim yetenekleri sağlayabilen bir dar bantlı güç hattı iletişim (NB-PLC) modemi, akıllı şebekelerin iletişim ihtiyaçları için ideal ve doğal bir çözümdür. Burada yüksek verimli dijital sinyal işlemcisi (DSP) kullanan bir NB-PLC modem tasarımı önerilmiştir. Bu tasarım, çok seviyeli frekans kaydırmalı anahtarlama modülasyonu (M-FSK) ile birleştirilmiştir. Doğrulama testleri, sistemin sağlam ve dar bantlı güç hattı iletişimleri için önemli ölçüde verimli hale getirildiğini göstermiştir.

Bu tez çalışmasında, BPSK modülasyon yöntemi kullanılarak, PLC dar bandına uyumlanabilecek şekilde Vivado ara yüzünde VHDL ile modem tasarımı ve benzetimi yapılmıştır. Literatür taramasında PLC'nin diğer frekans bantları kullanılarak başka uygulama alanlarında yapılmış çalışmaları karşımıza çıkmaktadır. FPGA paralel ve hızlı

işlem yapma , uygulamada yeniden yapılandırılabilme gibi avantajlara sahip olmasına rağmen FPGA tabanlı modem mimarilerinin, PLC kanalında kullanımının yaygın olmadığı görülmüştür. Bir sayısal haberleşme sisteminde verici ve alıcı yapılarının sorunsuz çalışması işaret özelliklerine yani sisteme uygun modülasyon yöntemine bağlıdır. FPGA tabanlı bu modemin en basit yapıda kolaylıkla tasarlanıp alanda değiştirilebilir olması ile modemin fiziksel olarak gerçekleştirilmesi gerekliliği ortadan kaldırılabilir ve ekonomik yönden bakıldığında donanım problemleri aşılabılır. Aynı zamanda sistem deneysel olarak gerçekleştirilmesi durumunda FPGA yapısında minimum alan kaplayacak dolayısıyla kullanılan lojik blok sayısı en az seviyede tutularak tasarımın basitliği sayesinde harcanan gücün de makul seviyede kalması sağlanacaktır.

1.3. Tezin Amacı ve Hedefleri

Bu tez çalışmasında, bir haberleşme alternatifi olan enerji hattı haberleşmesine ait dar bantta (100 kHz- 500 kHz) kullanılabilecek; bir modemin donanımsal gerçeklemede gerekli olan sabit fiziksel yapıların kullanımını gerektirmeme ve Vivado ortamında VHDL kodlamayla ekonomik, esnek ve uyarlanabilir olma gibi avantajlara sahip bir BPSK PLC modem tasarımı ve benzetiminin yapılması amaçlanmıştır. Bu amaç doğrultusunda tezin hedefleri sırasıyla

- ✓ İlk olarak MATLAB ara yüzünde, BPSK modülasyonunda kullanılacak bir sinüs işaretine ait örnekleri örnekleme teoremi kriterine göre elde etmek ve sonrasında bu örnekleri kullanarak Vivado programında VHDL ile 100 kHz frekansa sahip bir taşıyıcı işaret oluşturmak,
- ✓ Ardından Vivado ortamında VHDL ile 100 kHz bir referans taşıyıcı işaret ve onun 180° fazı kaydırılarak oluşturulan diğer bir işaret elde edilerek FPGA tabanlı BPSK modülatör mimarisi oluşturmak ve farklı bilgi işaretlerine göre sonuçları test etmek,
- ✓ Sonrasında Vivado ortamında VHDL ile BPSK demodülatör mimarisini çarpıcı blok, FIR filtre blok ve karar verici blok yapılarını içerecek şekilde adım adım tasarlamak,
- ✓ Son olarak tasarlanan bütün blokların haritalama ile birbiriyle bağlantısı ve uyumu sağlanarak FPGA tabanlı BPSK PLC modem mimarisi oluşturmak ve farklı bilgi işaretlerine göre benzetim sonuçlarını incelemek olarak verilebilir.

Ayrıca tasarımın, FPGA platformunda gerçekleştirildiğinde hafızada minimum yer kaplaması ve buna bağlı olarak harcanan gücün en az olması da tezin hedefleri arasındadır.

1.4. Tezin Kapsamı

Bu tez çalışmasında dar bant PLC haberleşmesi için FPGA tabanlı adaptif bir BPSK modem tasarımı ve benzetimi yapılmıştır. İlerleyen kısımlarda yer alan başlık ve anlatımlara şu şekilde yer verilmiştir.

Birinci bölümde, haberleşme sistemleri ile genel bilgilerden bahsedilerek giriş yapılmış, farklı modülasyon türlerinin, farklı FPGA kartının kullanıldığı FPGA tabanlı yapıların çeşitli alanlardaki literatür özetine yer verilmiş, tezin amacı ve hedefleri anlatılmıştır. Burada sayısal modülasyon genel teorisinden bahsedilmiş, tezde kullanılan ikili faz kaydırmalı anahtarlama modülasyon tekniği matematiksel denklemlerle birlikte detaylı olarak anlatılmış ve son olarak bu modülasyon tekniğinin MATLAB benzetimi gerçekleştirilerek çıktıları verilmiştir.

İkinci bölümde tasarımda kullanılan donanım tanımlama dili VHDL hakkında ve kullanılan platform alanda programlanabilir kapı dizileri (FPGA) hakkında bilgiler yer almaktadır.

Üçüncü bölümde, öncelikle VHDL ile nasıl tasarım yapılacağına dair bir akış verilerek yapılan çalışmalara yer verilmiştir. Bu kısımda FPGA derleyicisi Vivado programı üzerinde temel blok tasarımlarına yer verilmiş ve tezde kullanılan BPSK sayısal modülasyon tekniğinin FPGA uygulamaları anlatılmıştır. Bu bölümde Vivado ortamında oluşturulan modülatör ve demodülatör mimarileri için oluşturulan kodlama akış diyagramları ve elde edilen bulgular verilmiştir ve irdelenmiştir.

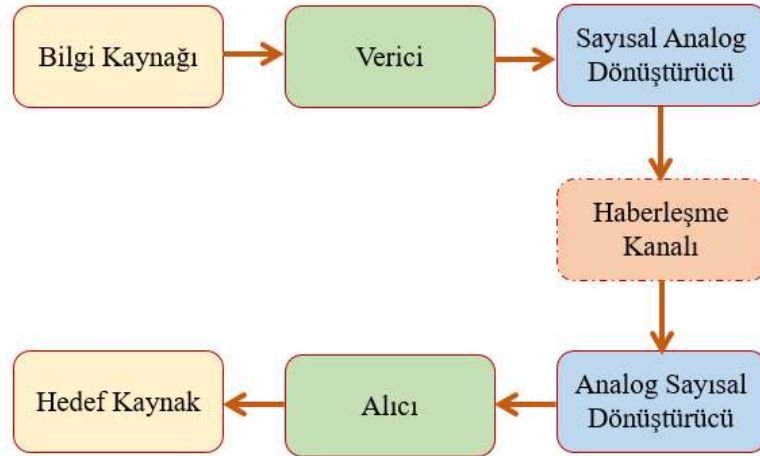
Dördüncü bölümde, tasarlanan modem mimarisine ait sonuçlar ve tartışma kısmı sunulmuştur.

Beşinci bölümde ise tez çalışmasının genel bir değerlendirmesi yapılmış ve gelecek çalışmalarda yapılabilecek önerilere yer verilmiştir.

1.5. Haberleşme Teorisi

Elektronikte haberleşme, analog veya sayısal herhangi bir bilginin, uzay ve zaman içinde bir noktadan başka bir noktaya aktarılması sürecinin tamamıdır. Haberleşmede amaç, verilen bir zaman aralığı içinde, bilginin, iletim ortamındaki gürültü, girişim, buruşum gibi bozucu etkilerden en az etkilenecek şekilde hedefe aktarılmasıdır. Bilgi kaynağı ile hedef kaynak arasındaki veri aktarımı işlemini sağlayan tüm sistem haberleşme sistemidir. Sayısal bir haberleşme sistemine ait yapı Şekil’1 de gösterilmektedir.

Haberleşme sistemlerinde ses, görüntü, sıcaklık gibi alçak frekanslı analog bilgi işaretlerinin elektriksel işarete dönüştürülmesinden sonra ortamda iletilmeleri için modülasyon adımının gerçekleştirilmesi gerekmektedir. Etkin bir iletişim için verici yapı içerisinde gerekli işlemler yapılmalıdır. Bu işlemlerin en önemlisi modülasyondur. Modülasyon işlemi, periyodik bir elektromanyetik, akustik veya optik dalganın kullanılacak modülasyon tekniğine göre o özelliği taşıyan parametresinin değiştirilerek bilginin iletilmesidir.



Şekil 1. Sayısal bir haberleşme sisteminin genel görünümü

Haberleşmede kullanılacak anten boyutunun belirlenmesinde modülasyon işleminin rolü büyüktür. Bilgi işaretini modülasyon olmadan iletmek için kullanılan anten boyunun çok büyük olması gerekmektedir. Ancak taşıyıcı işaretin yüksek frekansı sayesinde anten boyu küçük seçilebilmektedir. Böylece sınırlayıcı etki ortadan kaldırılarak anten tasarımı

kolaylıkla yapılabilmektedir. Modülasyon işlemi gürültü, girişim, buruşum gibi olumsuz etkileri azaltmaktadır. Örneğin, elektromanyetik dalgaların girişimi oluştuğunda alıcı ve verici birimlerin taşıyıcı frekansları farklı belirlenerek sorun çözülebilir. Modülasyon, haberleşme için tasarlanan alıcının ve vericinin kendileri için belirlenen frekans aralığında diğer sistemlerle girişim olmadan iletim yaparak frekans bandının tümünün kullanımına imkan sağlamaktadır. Ayrıca televizyon, telefon vb. iletim kanallarında olduğu gibi bazı sistemlerde aynı anda birden çok bilgi işareti gönderilmek istendiğinde yine modülasyon yapılması gerekmektedir.

Sayısal modülasyon, sayısal işaretleri iletim yapılacak haberleşme kanalına uygun dalga şekillerine dönüştürmek için kullanılan modülasyondur. Bu modülasyonda, analog taşıyıcının genlik, açı veya faz parametreleri sayısal bilgi işaretine göre değiştirilmektedir. İletilecek bilgi doğrudan sayısal çıktı veren bir kaynak olabileceği gibi analog veri de olabilmektedir. Bu durumda analog sayısal dönüşümü yapılarak bilgi işareti sürekli dalga formundan sayısal veri haline getirilmelidir [28].

Analog modülasyon, gürültüye oldukça duyarlı olan modülasyon türlerinden oluşmaktadır. Gürültünün olumsuz etkilerini gidermek için sayısal modülasyon teknikleri kullanılmaktadır. Sayısal modülasyonun analog modülasyona göre birçok üstünlüğü bulunmaktadır.

- Sayısal alıcı-verici yapıları analog yapılara göre daha ucuz, verimli ve hızlı olmaktadır.
- Zayıf sinyal gücüne yani gürültüye karşı daha dayanıklıdır.
- Yüksek güvenlik ile gizli iletişime imkan sağlamaktadır.
- Sahip olduğu çoklu erişim teknolojileri ile bilginin çoğullanması daha kolaydır.
- Kaynak ve denetim kodlama ile daha gelişmiş hata düzeltme teknikleri kullanılmaktadır.

Analog veya sayısal bilgileri, veri iletmek için hangi modülasyon tekniğinin kullanıldığına bakılmaksızın alıcı kısmında tekrar elde etmek için demodülasyon işleminin yapılması gerekmektedir. Kısaca demodülasyon bir taşıyıcı üzerine bindirilip gönderilen bilgi işaretini ayıklamaktır. Bir haberleşme sisteminin sağlıklı bir şekilde çalışması için vericide kullanılan modülasyon tekniği ile alıcıda tarafında yapılan demodülasyon işleminin uyumlu olması sağlanmalıdır. Bu şekilde modülasyon ve demodülasyonun eş zamanlı ve uyumlu bir şekilde gerçekleştirildiği yapılara modem denir. İsmi uygulanan işlemlerin kısaltmasından alan modem kullanım amacı, mesafe, gürültü, hız gibi parametreler göz

önüne alınarak tasarlanmalıdır. Tasarlanan sistemin yapısına uygun modülasyon tekniği uygulanmalıdır.

1.6. İkili Faz Kaydırmalı Anahtarlama Yöntemi (BPSK)

Faz kaydırmalı anahtarlama gönderilmek istenen bilgi, taşıyıcı işaretin fazında saklanmaktadır. Bilgi işareti bitler halinde veya bitler birlikte kullanılarak semboller halinde iletebilir. Sembol sayısına göre modülasyon M-PSK olarak adlandırılır. Sembol sayısının iki olduğu durum özel olarak BPSK olarak isimlendirilir [29].

BPSK modülasyonunda ikili sistem halindeki veriler tek bir taşıyıcı frekansı için farklı fazlarda iki sinyal ile ifade edilmektedir. Girişteki sayısal veriler değiştikçe taşıyıcının fazı iki açı değeri arasında kaymaktadır. Eşitlik (1) ve (2) 'de geçen $s_i(t)$ terimi periyodik taşıyıcı işaretini, f_t bu işaretin frekansını ve A bu işaretin genliğini temsil etmektedir. İletilen işaretin genel ifadesi Eşitlik (1)'de verilmektedir.

$$s_i(t) = \begin{cases} A \sin(2\pi f_t t + \phi_1), & \text{sayısal sembol 1 ise } (s_1(t) \text{ taşıyıcı işareti}) \\ A \sin(2\pi f_t t + \phi_0), & \text{sayısal sembol 0 ise } (s_0(t) \text{ taşıyıcı işareti}) \end{cases} \quad (1)$$

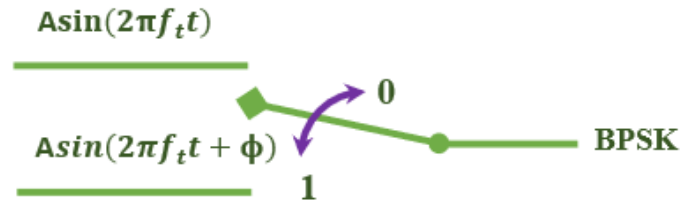
Taşıyıcı frekans için belirlenen iki çıkış fazı (ϕ_1 ve ϕ_0) mevcut olabilmektedir (0 ve π) veya ($\pi/2$ ve $3\pi/2$). Tipik kullanım olarak ikili sayısal sembollerden biri için "0" diğer sembol için " $180^\circ(\pi)$ " fazını taşıyan işaretler Eşitlik (2)'de gösterildiği gibi oluşturulmaktadır.

$$s_i(t) = \begin{cases} A \sin(2\pi f_t t), & \text{sayısal sembol 1 ise } (s_1(t) \text{ taşıyıcı işareti}) \\ A \sin(2\pi f_t t + \pi), & \text{sayısal sembol 0 ise } (s_0(t) \text{ taşıyıcı işareti}) \end{cases} \quad (2)$$

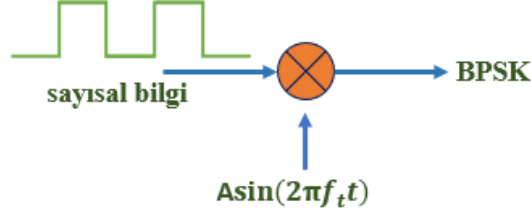
1.7. BPSK Modülasyonu

Faz tersinim anahtarlama da denilen ikili faz kaydırmalı anahtarlama, tek bir taşıyıcı frekansı için iki çıkış fazı mümkündür. Bu yöntem, sinüzoidal taşıyıcı işaretinin fazının, gönderilecek bilgi işaretine bağlı olarak değiştirilmesi ile yapılan modülasyondur.

Faz kaydırmalı modülasyonda taşıyıcının genliği ve frekansı korunur, değiştirilmez [30]. Taşıyıcı işaretin genlik değeri harcanan güç ve beklenen hata oranına göre değiştirilirken, frekans değeri ise sistemin iletim yapacağı band seviyesine göre ayarlanmaktadır. BPSK modülasyon genel çalışma mantığı Şekil 2’de verilmektedir. Bu modülasyonda, Şekil 2’de görüldüğü üzere gelen bilgi işareti sembolüne göre anahtarlama yapılmaktadır. Anahtar konumuna göre belirlenen taşıyıcı işaretin, sayısal bilgi işareti ile çarpılması ile BPSK modüleli işaretin elde edilmesi Şekil 3’te gösterilmektedir.

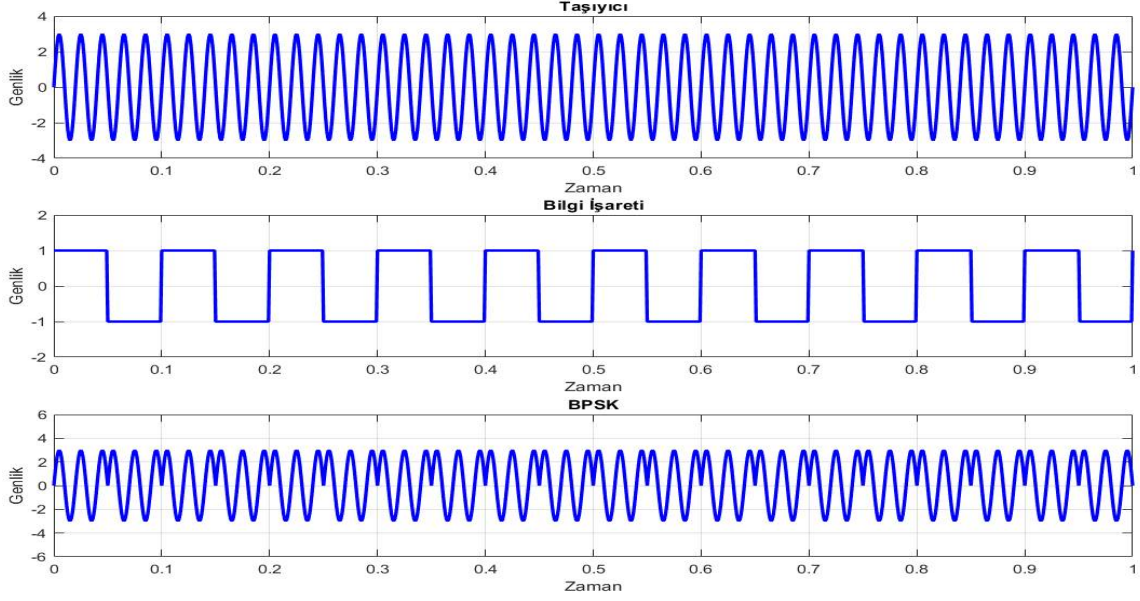


Şekil 2. BPSK modülasyonu anahtarlama



Şekil 3. BPSK modülatör blok yapısı

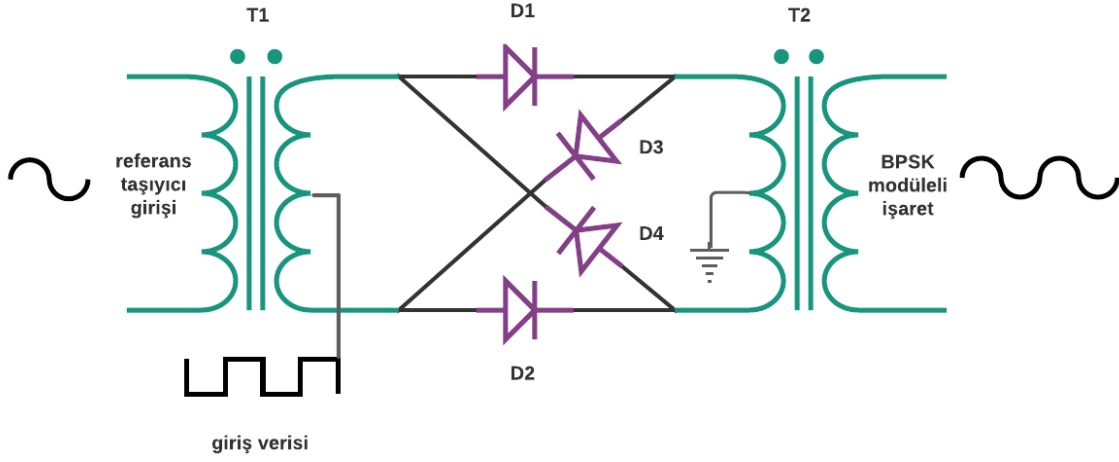
Şekil 4’te MATLAB ile elde edilen bulguya göre sayısal bilgi işareti 1 olduğunda 0° fazına sahip sinüzoidal işaret iletilirken, 0 olduğunda ise 180° fazına sahip taşıyıcı işaret iletilmektedir. Kısaca gelen bilgi işareti 1 ise taşıyıcı sinyalin fazı değiştirilmez iken 0 olduğunda ise taşıyıcı sinyalin fazı 180 derece kaydırılmaktadır.



Şekil 4. Örnek bir BPSK modüleli işaret

1.7.1. Dengeli Modülatör

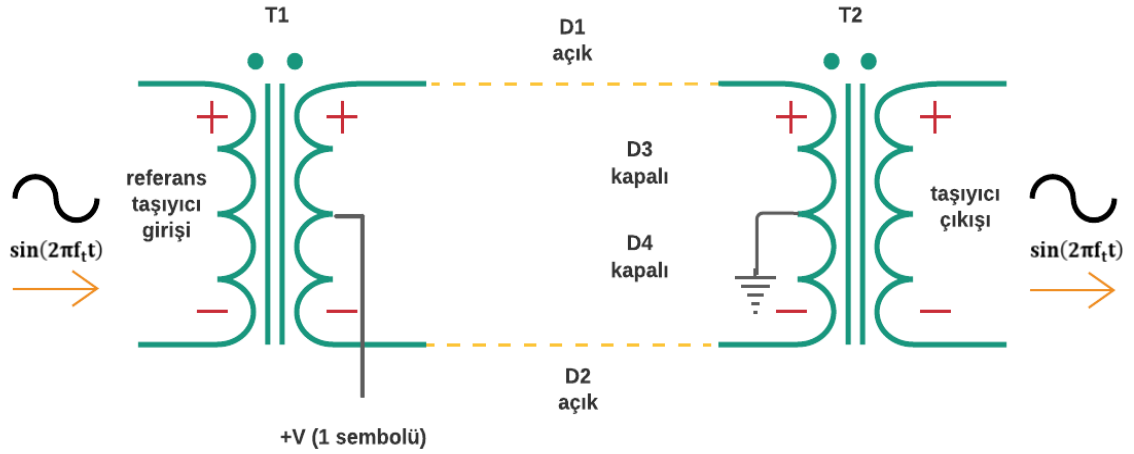
Elektronik iletişimde, dengeli bir modülatör, çift yan bantlı bastırılmış taşıyıcı sinyalleri üreten bir devredir. Kafes modülatörü olarak da bilinen diyot halka modülatörü en yaygın dengeli modülatörlerden biridir. Ayrıca diyot halka modülatörü, elektronik haberleşmede en yaygın olarak kullanılan devrelerden biridir. Çift yan bantlı bastırılmış taşıyıcı sinyalleri üretmenin yanı sıra frekans ve faz modülasyon sistemlerinde ve PSK ve QAM gibi sayısal modülasyon sistemlerinde de kullanılmaktadır. Başlangıçta bir "halka" olarak biçimlendirilmiş dört diyot, veri ögesi, giriş ve çıkış transformatörlerinden oluşmaktadır. Bir faz çevirme anahtarı gibi hareket dengeli modülatörün yapısı Şekil 5'te gösterilmektedir. Halka modülatörün tüm diyotları saat yönünde veya saat yönünün tersine bakacak şekilde yerleştirilmektedir. Dengeli modülatörün referans taşıyıcı dalga ile aynı fazda olan taşıyıcı ve ikili bilgi işareti olmak üzere iki girişi vardır. Taşıyıcı sinyal, giriş ve çıkış transformatörlerinin orta kademelerinde; modülasyon sinyali giriş transformatörünün primerinde uygulanmaktadır. Çıkış sinyali ise, çıkış transformatörünün sekonderinde ölçülmektedir. Denge modülatöründeki diyotların açık/kapalı durumları bu sayısal giriş işaretine göre kontrol edilerek modülatörün çıkışında BPSK modüleli işaret elde edilmektedir.



Şekil 5. Denge modülatörü çalışma ilkesi

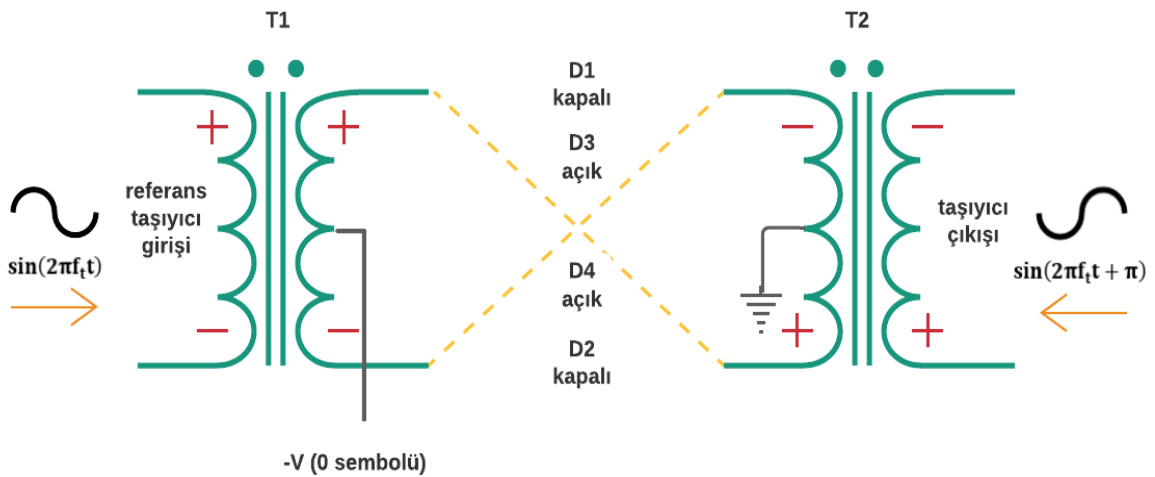
Dengeli modülatör, giriş sinyalinin fazının referans olarak belirlenen değerde veya 180° çevrilmesiyle oluşturulan değerde iletilmesini kontrol eden bir anahtar olarak görev yapmaktadır. Taşıyıcı sinyal, diyotları yüksek hızda açıp kapatan sinyaldir. Modülatörün çalışması için, taşıyıcının genliğinin, modüle edici sinyalinkinden yeterince büyük olması gerekmektedir.

Dengeli modülatörün çalışma prensibi iki kısımda incelenmektedir. İlk durum olarak pozitif yarı çevrim çalışması Şekil 6'da gösterilmektedir. Modülatörün pozitif çevrim sırasında, D1 ve D2 ileri yönlüdür yani açık konumdadır ve kısa devre gibi davranmaktadır. D3 ve D4 ters yönlüdür yani kapalı konumdadır ve açık devre olmaktadır. Sonrasında taşıyıcı akımı giriş transformatörünün sekonderinde eşit olarak bölünüp sargının üst ve alt kısımları boyunca zıt yönlerde akmaktadır. Alt ve üst kısımlardaki akımlar birbirine eşit ve zıt yönde birer manyetik alan oluşturmaktadır ve bu manyetik alanlar birbirini götürmektedir. Böylece modüledi işaret, girişten çıkış transformatörlerine faz değişimi olmadan D1 ve D2 üzerinden iletilmektedir.



Şekil 6. Denge modülatörü pozitif çevrimi

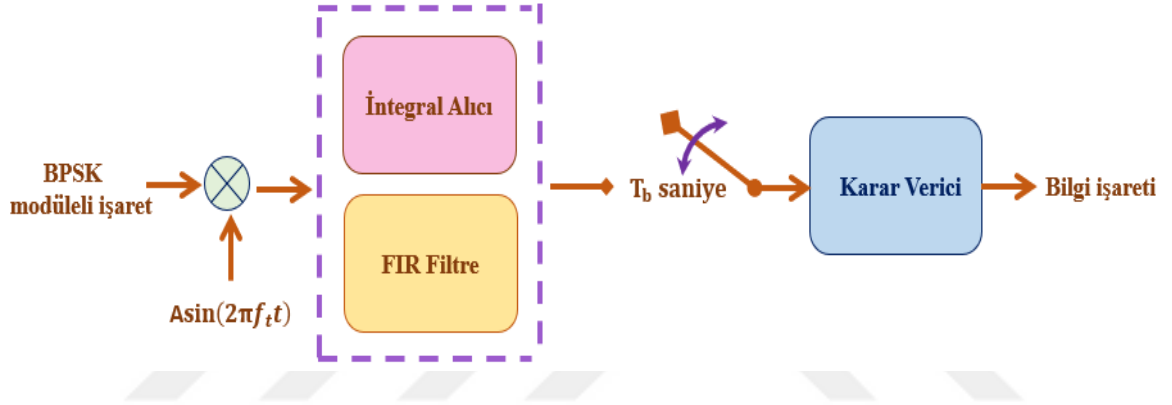
Modülatörün diğer durum olarak negatif çevrim çalışması Şekil 7’de gösterilmektedir. Modülatörün negatif çevrimde olmasında ise, D3 ve D4 ileri yönlüdür yani açık durumdadır ve kısa devre gibi davranmaktadır. D1 ve D2 ters yönlüdür yani kapalı durumdadır ve açık devre olmaktadır. Taşıyıcı akım aynı şekilde giriş transformatörünün sekonderinde eşit olarak bölünüp sargının üst ve alt kısımları boyunca zıt yönlerde akmaktadır. Çıkış transformatörünün primerinde eşit olarak bölünüp her iki akım da birbirine eşit ve zıt manyetik alanlar üretmektedir. Giriş transformatörünün sekonderinde iki akım birleşip manyetik alanlar nötrlenmektedir. Sinyal, giriş transformatöründen geçmekte ve çıkış transformatörüne ulaşmadan önce faz 180° tersine dönmektedir.



Şekil 7. Denge modülatörü negatif çevrimi

1.8. BPSK Demodülasyonu

Faz kaydırmalı anahtarlama modülasyon tekniğinde kullanılan farklı fazlara sahip taşıyıcı işaretler genlik ve frekans olarak aynı değerlere sahiptir. Bu yüzden zarf sezici kullanılarak eşvreli olmayan demodülasyon işlemi gerçekleştirmek mümkün değildir. Kısaca BPSK demodülasyonu için Şekil 8’de gösterilen eşvreli demodülasyon yöntemi kullanılmaktadır. Buna göre taşıyıcı frekansına ve fazına sabitlenmiş bir sinüzoidal demodülasyon işareti gerekmektedir.



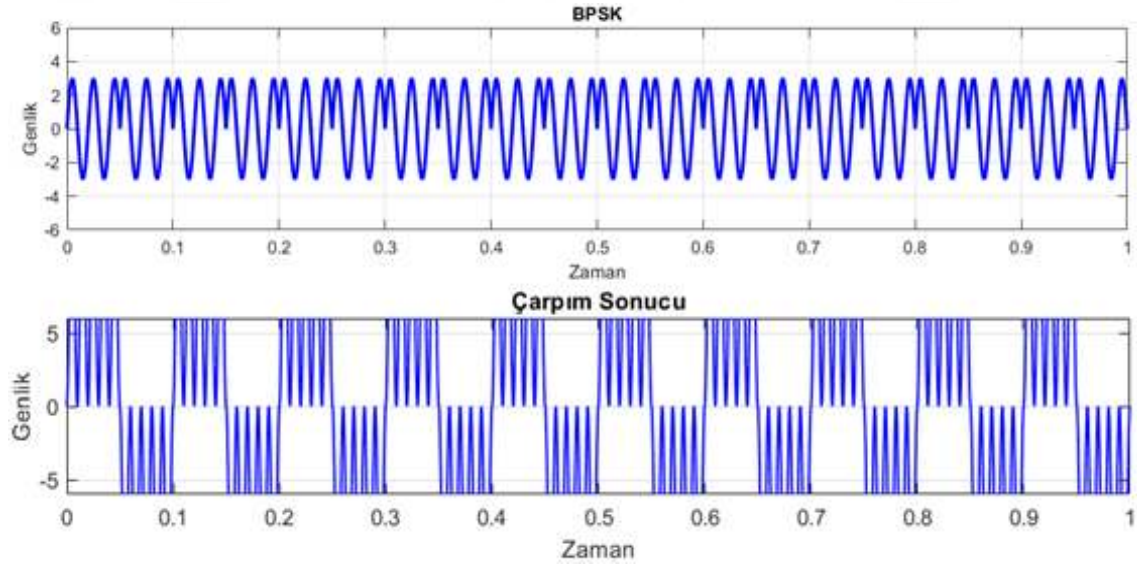
Şekil 8. Eşvreli BPSK Demodülatör blok yapısı

Şekil 8’de gösterilen yapıya göre BPSK demodülasyonu için öncelikle çarpım modülatöründe, gelen BPSK modüleli işaretin, sistemde daha önce üretilmiş var olan referans taşıyıcı işaret ile çarpılması gerçekleştirilmektedir. Çarpım bloğu çıkışı integral alıcı veya sisteme uygun tasarlanan bir FIR filtre bloğuna iletilmektedir. Bu işlemten sonra modüleli işaret içerisinde daha görünür hale gelmesi beklenen bilgi işaretinin tam olarak ayırt edilebilmesi için karar verici blok kullanılmaktadır. Uygun eşik değeri belirlenen karar verici blok ile iletilen bilgi işaretinin çıkışta tekrar elde edilmesi sağlanmalıdır. Eşvreli BPSK demodülatör yapısı hakkında daha ayrıntılı bilgiler alt başlıklarda verilmektedir.

1.8.1. Çarpım Blok Yapısı

BPSK demodülatörün çalışma ilkesine göre taşıyıcı devresi frekansı ve fazı başlangıçtaki taşıyıcı işaret ile aynı olan bir taşıyıcı sinyali üretmektedir. Bir çarpım dedektörü olarak görev yapan denge modülatörü, girişine gelen BPSK modüleli işaret ile taşıyıcı işareti çarpmaktadır.

MATLAB’de elde edilen BSPK modüleli işaretin, referans taşıyıcı sinyal ile çarpılması sonucu elde edilen işaret Şekil 9’da verilmektedir. Bu çarpma işlemine göre bilgi işaretinin, modüleli işaret içinde daha belirgin hale getirildiği görülmektedir.



Şekil 9. Örnek bir çarpıcı çıkışı

1.8.2. İntegral Alıcı Blok Yapısı

Sayısal sembol olarak “0” geldiğinde oluşturulan $s_0(t)$ taşıyıcısı ile sembol olarak “1” geldiğinde oluşturulan $s_1(t)$ taşıyıcısı ile alıcı yapısında bulunan referans taşıyıcı işareti $s(t) = A\sin(2\pi f_t t)$ olarak alındığında, çarpım modülatöründe gerçekleştirilen işlemlerin ardından integral alıcı blok çıkışında elde edilen sonuçlar Eşitlik (3) ve Eşitlik (4) ‘te verilmektedir.

$$s_{d0}(t) = \int_0^{T_b} (s_0(t) * s(t))dt = \int_0^{T_b} (-A\sin(2\pi f_t t)) * (A\sin(2\pi f_t t))dt$$

$$s_{d0}(t) = -\frac{1}{2}A^2T_b, \text{ çıkış 0 ise} \quad (3)$$

$$s_{d1}(t) = \int_0^{T_b} (s_1(t) * s(t))dt = \int_0^{T_b} (A\sin(2\pi f_t t)) * (A\sin(2\pi f_t t))dt$$

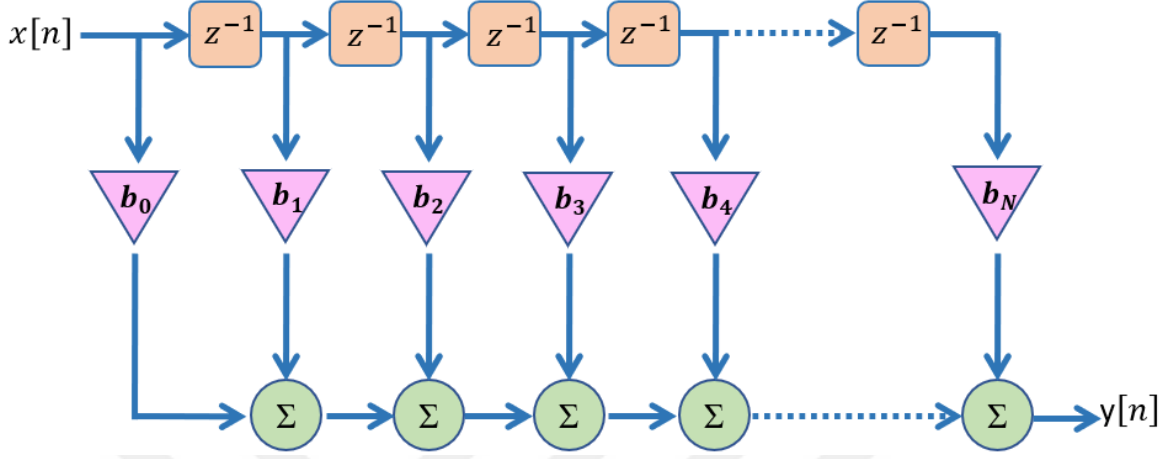
$$s_{d1}(t) = \frac{1}{2}A^2T_b, \text{ çıkış 1 ise} \quad (4)$$

1.8.3. FIR Filtre Yapısı

Sayısal filtreler elektroniğin her alanında kullanılmaktadır. Ancak ses ve video uygulamaları ile sayısal işaret işleme için daha yaygın kullanım alanı bulunmaktadır. Bunun nedeni, sayısal filtrelerin analog filtrelerden daha iyi SNR'ler elde edebilmesidir [15]. Daha iyi SNR yeteneği gürültüsüz matematiksel işlemlerden gelir, ancak bu özellik analog filtreler için doğru değildir çünkü analog filtre sinyale daha fazla gürültü eklemektedir. Sonuç olarak, sayısal filtreler, iletişim sistemlerinde gürültüyü gidermek ve spektrumunu şekillendirmek için tercih edilen seçenek haline gelmiş bulunmaktadır. Sayısal filtreler, hassas tekrarlanabilirlik özelliğine sahip olduğu ve tasarımcıların analog filtrelerle elde edilmesi zor olan yüksek performans seviyelerine ulaşmalarına olanak sağladığı için daha yaygın kullanıma sahiptir [15]. Sayısal filtreler FIR (Finite Impulse Response) ve IIR (Infinite Impulse Response) olmak üzere iki yaygın filtre formuna sahiptir. Bir sistemde tasarlanmak istenen filtrenin IIR veya FIR seçilmesi için faz, gecikme, kararlılık, tasarıma göre hesaplama hızı, yineleme gibi özellikler dikkate alınmalıdır.

Birim zamanda yapılan her işlem FPGA kartında aktif edilen veri yolu ve mantık kapısı kullanımını etkilemektedir. Filtre türlerinin matematiksel yapısı incelenirse, FIR filtrelerin sadece paylara sahipken, IIR filtrelerin hem pay hem de payda ifadelerine sahip olduğu görülmektedir. Bu nedenle IIR filtreleri, FIR filtrelerine göre daha karmaşık bir yapıya sahiptir ve aynı dereceden bir filtre tasarlandığında FIR filtrelerinden çok daha fazla kaynak kullanılmaktadırlar [15]. Demodülatör yapısında kullanılacak filtrenin verilen bilgilere

göre Şekil 10’da gösterilen FIR filtre olarak tasarlanmasının daha uygun olduğuna karar verilmiştir.



Şekil 10. Genel FIR filtre yapısı

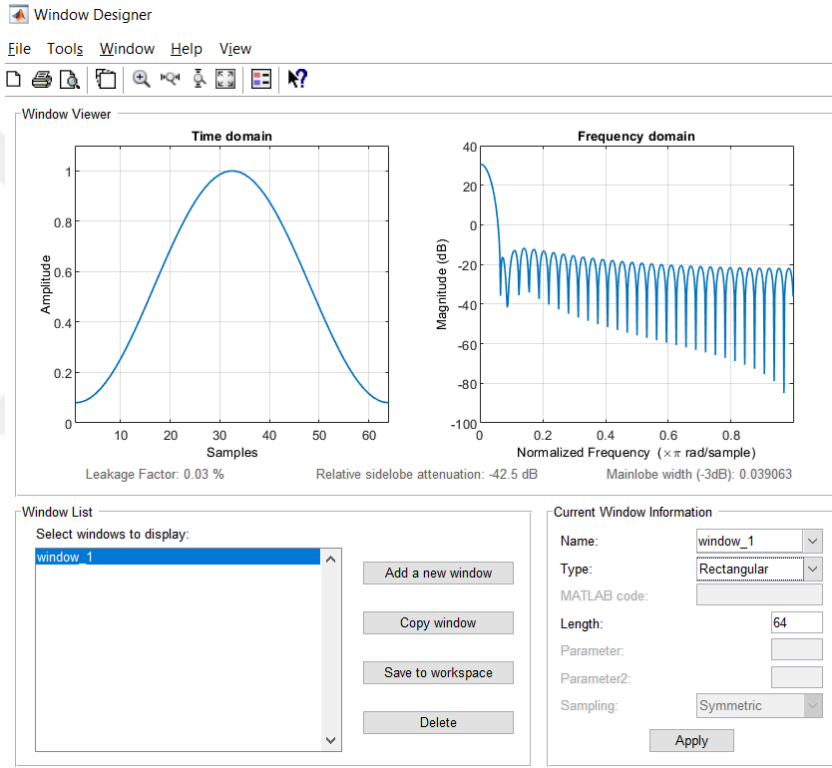
FIR filtreleri, ileri beslemeli veya özyinelemeli olmayan veya çapraz filtreler olarak da bilinmektedir. Şekil 10’da gösterilen bir FIR filtresi genellikle, filtrenin çıktısını oluşturmak için bir dizi gecikme, çarpan ve toplayıcı kullanılarak uygulanmaktadır. FIR filtresinin dürtü yanıtı; $x(n)$ filtre girişi, $y(n)$ filtre çıkışı, b_N ise filtre katsayıları, N geciktirici eleman sayısı ve filtrenin derecesi olmak üzere Eşitlik (5)’teki gibi verilebilir.

$$y[n] = b_n * x[n] + b_{n-1} * x[n - 1] + b_{n-2} * x[n - 2] \dots \dots \dots + b_{n-N} * x[N] + b_N \quad (5)$$

Şekilde ve eşitlikte belirtilen FIR filtre çalışma prensibine göre girişine gelen $x[n]$ işareti ilgili filtre katsayısı ile çarpılmakta, sonra gecikmeye uğratarak elde edilen $x[n - 1]$ işareti kendisine denk gelen diğer ilgili filtre katsayısı ile çarpılmaktadır. Ardından bu iki işaret toplanarak diğer toplayıcı girişine aktarılmaktadır. Çarpma işlemi filtre derecesi kadar tekrarlanmaktadır ve bütün çarpım işlemlerinin sonuçları her seferinde birbiri ile toplanarak filtre çıkışı $y[n]$ oluşturulmaktadır.

1.8.3.1. FIR Filtre Tasarım Yöntemi

FIR filtre tasarım aşamasında çok sayıda yöntem bulunmaktadır. Bunlardan en yaygın olanı “Window Design Method” yani “Pencere Tasarım Metot” olarak bilinmektedir. Sayısal filtre tasarımı için pencere yöntemi hızlı, kullanışlı ve sağlamdır, ancak genellikle optimal değildir. Şekil 11’de verilen ekranda BPSK demodülatör yapısına uygun FIR filtre tasarımı için gerekli frekans ayarlamaları yapılabilmektedir. Bu tez kapsamında bu yöntem ile FIR filtre katsayıları elde edilerek demodülatör tasarımında kullanılmıştır.

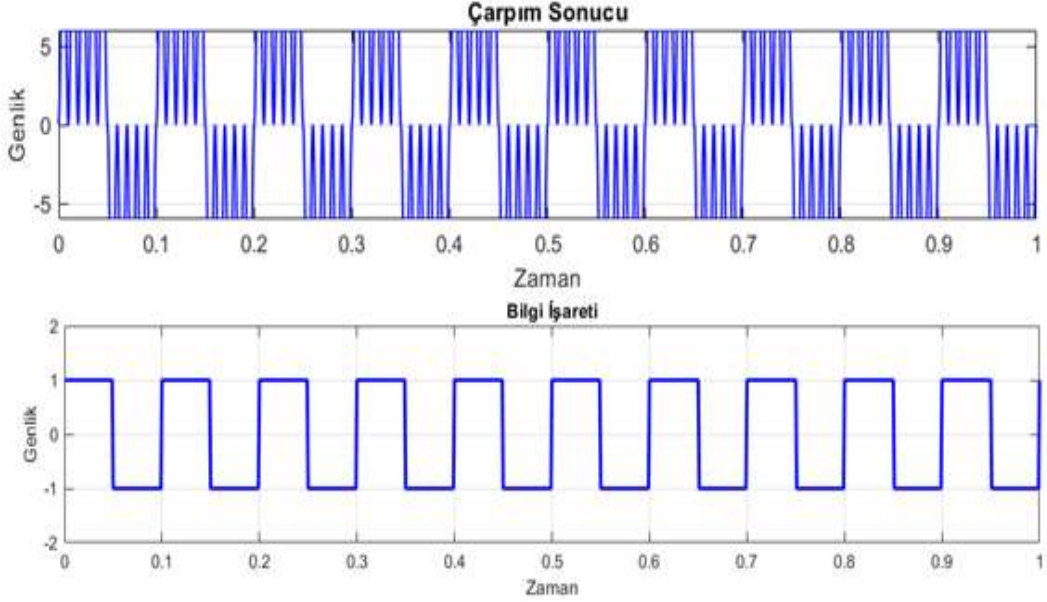


Şekil 11. Pencere tasarım yöntemi default görünümü

1.9. Karar Verici Blok Yapısı

BPSK demodülatör tasarımında karar verici bloğunda eşik değeri belirlenirken, çıkışların ortalaması olarak sıfır değeri uygun olmaktadır. Bu karar seviyesine göre Eşitlik (3) ve (4)’te verilen integral alıcı bloğu çıkışlarına bakıldığında, bu bloktan gelen negatif değerler karar verici bloktan “0” bilgi sinyali olarak; pozitif değerler ise karar verici bloktan “1” bilgi sinyali olarak tekrar elde edilebilmektedir.

MATLAB’de elde edilen örnek bir BSPK modüleli işaretin, referans taşıyıcı sinyal ile çarpılması sonucu elde edilen işaret Şekil 12’de verilmektedir. Bu çarpma işleminde oluşturulan sinyal FIR filtreden geçirilip ardından karar verici eşiği uygulandığında elde edilen bilgi işareti Şekil 12’de net olarak görülmektedir.



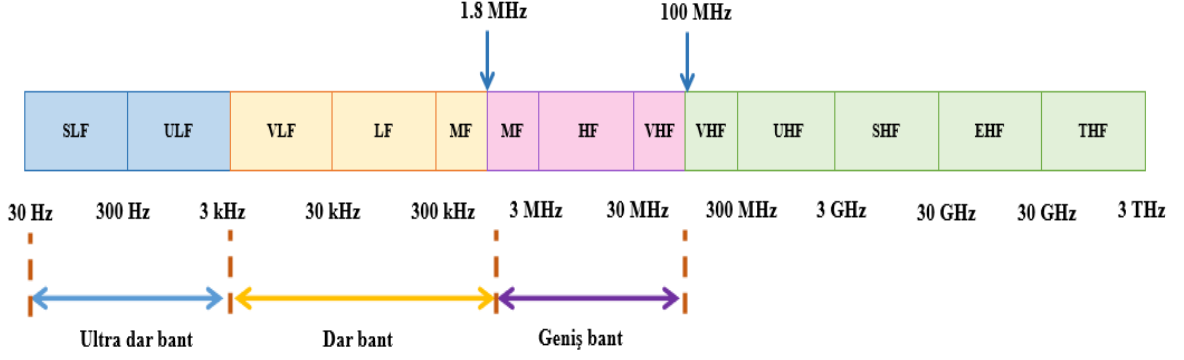
Şekil 12. Örnek olarak tekrar elde edilen bilgi işareti

1.10. Enerji Hattı Üzerinden Haberleşme (PLC)

Enerji hattı üzerinde haberleşmede mevcut şebeke üzerinden hem enerji hem de veri iletimi birlikte gerçekleştirilmektedir. Enerji iletimi büyük ve yavaş değişen dalga ile sağlanırken, veri iletimi küçük ve hızlı değişen dalga ile sağlanmaktadır. İki iletim de farklı hızlarda sağlanmasına rağmen elektrik akımı ve radyo dalgaları kendilerine ait frekanslarda gönderildiği için birbirlerine engel olmazlar yani girişim etkisi göstermezler. PLC de klasik haberleşme sistemlerinde olduğu gibi bir iletişim ortamı aracılığıyla gönderilecek verileri modüle eden bir göndericiden ve verileri daha sonra kullanmak üzere demodüle eden bir alıcıdan oluşmaktadır. Bu haberleşmede ASK, FSK, BPSK, OFDM gibi modülasyon türleri kullanılmaktadır [17].

Enerji hattı haberleşmesi frekans bant genişliğine göre ultra dar bant, dar bant ve geniş bant olmak üzere 3 grupta incelenmektedir. ITU (International Telecommunication Union) tarafından belirlenen frekans bantları ve PLC sistemlerindeki kullanımları Şekil 13’te

görüldüğü gibidir. Burada 30 Hz ile 3 THz frekans aralığı SLF süper düşük, ULF ultra düşük, VLF çok düşük, LF düşük, MF orta, HF yüksek, VHF çok yüksek, UHF ultra yüksek, SHF süper yüksek, EHF aşırı yüksek ve THF müthiş yüksek frekans olmak üzere 11 bölgeye ayrılmıştır.



Şekil 13. ITU frekans bantları

Ultra dar bant PLC 30-300 Hz/ 0.3-3 kHz frekans aralıklarında tek yönlü iletişimde kullanılmaktadır. Dar bant PLC 3-500 kHz bant aralığında, geniş bant PLC 1.8-100 MHz ya da 1.8- 250 MHz bant aralığında kullanılmaktadır.

PLC teknolojisi eski bir haberleşme sistemi olmasına rağmen belirli bir standarta sahip olmayıp dünya üzerinde ülkeden ülkeye değişen frekans bantı kullanımına sahiptir. Dar bant PLC sistemleri için kullanılacak bant genişlikleri Avrupa ülkelerinin kullandığı CENELEC standartlara göre belirlenmiştir. Ülkeler ihtiyaçlarına göre frekans bant genişliğini ayarladığından bir ülkede PLC için kullanılan frekans bandı başka bir ülkede farklı bir amaçla kullanılabilir. Avrupa ülkelerinin kullandığı CENELEC standartına göre dar bant 3-148.5 kHz olarak, Amerika için FCC standartıyla 10-490 kHz, Japonya için ARIB standartıyla 10-450 kHz, Çin için FCC standartıyla 3-500 kHz olarak belirlenmiştir. Bant genişliğindeki sınıflandırmanın yanı sıra, kullanılacak güçler için de sınıflandırma yapılmıştır [31]. CENELEC standartına göre frekans bandı 4 gruba ayrılmaktadır.

- A-Bandı (3 kHz–95 kHz) elektrik üreticileri için,
- B-Bandı (95 kHz–125 kHz) erişim protokolü olmadan bütün uygulamalar için kullanılabilir,
- C-Bandı (125 kHz–140 kHz) ev ağırları için ayrılmıştır,
- D-Bandı (140 kHz–148.5 kHz) alarm ve güvenlik sistemleri için ayrılmıştır.

Dar bant PLC daha düşük veri hızına sahip olup daha düşük güç tüketir [31]. İletim sektöründe dar bant PLC, kırık izolatörleri, kısa devreleri ve açık / kapalı besleyici kesicileri, gerçek zamanlı sarkma izleme, uzak istasyon gözetimi, arıza tespiti, trafo merkezi otomasyonu ve akıllı ölçüm uygulamalarını tespit etmek için kullanılır.

Günümüzde en yaygın kullanılan dar bant PLC CENELEC A bandıdır [31]. Bu tez kapsamında kullanılan frekans bandı ise CENELEC B bandı olup taşıyıcı frekans 100 kHz seçilmiştir.



2. KULLANILAN PLATFORM VE YAZILIM

2.1. Programlanabilir Mantık Aygıtları (Programmable Logic Devices- PLD)

Mantık kapıları, flip-floplar gibi yapılar içeren ve bu elemanları yazılım ile programlayarak istenilen mantık işlemini gerçekleyen tüm devreler PLD olarak isimlendirilmektedir. Bu mantık elemanları tasarlanan baskı devre kartını optimum hale getirebilme, tasarımda esneklik sağlama, üretim sonrasında kolayca test edilebilme, güç tüketimini azaltma gibi birçok avantaj özelliklere sahiptir. Kullanılacak devreye göre mantık blokları arasında bağlantı kurularak karmaşık işlevleri yerine getirebilen PLD' ler üç gruba ayrılmaktadır:

- Basit Programlanabilir Mantık Aygıtı (Simple Programmable Logic Device, SPLD)
- Karmaşık Programlanabilir Mantık Aygıtı (Complex Programmable Logic Device, CPLD)
- Alanda Programlanabilir Kapı Dizisi (Field Programmable Gate Array, FPGA)

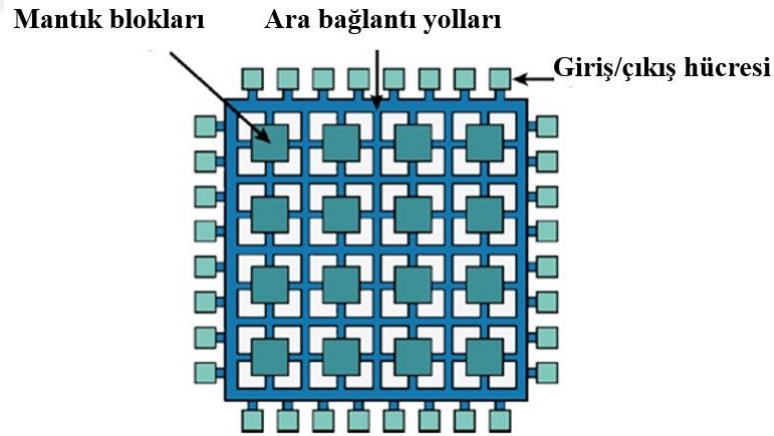
2.1.1. Alanda Programlanabilir Kapı Dizileri

Sahada programlanabilir kapı dizileri (Field Programmable Gate Array - FPGA), üretimden sonra istenen fonksiyona göre kullanıcı tarafından donanım yapısı değiştirilebilen yarı iletken cihazlardır. İçerisinde bulunan çok sayıdaki mantık devresi sayesinde basit tasarımlardan karmaşık tasarımlara kadar pek çok donanımın gerçekleştirilebilmektedir. Ayrıca FPGA yeniden programlanarak tekrar tekrar kullanılırken, hata durumunda ise kolayca düzeltilebilmektedir. Paralel işlem yeteneği yani eş zamanlı çalışabilmesi birden fazla işlemi aynı anda yapması diğer entegrelere göre daha hızlı olmasını sağlamaktadır. Bu özellik sinyal işleme gibi uygulamalarda kullanımını artırmış, çok sayıda entegrenin yaptığı işi yapabilmesi devre kartlarının boyutunu düşürmüştür. Böylece havacılık, medikal, otomotiv, tüketici elektroniği, dijital sinyal işleme, bulanık mantık denetleyicileri, motor denetleyicileri, sinir ağı uygulamaları gibi çeşitli uygulamalarda FPGA kullanım alanı bulmuştur.

FPGA diğer tüm PLD'lerin özelliklerini taşımakla birlikte hız, enerji tüketimi, güvenilirlik, kullanım kolaylığı, çok yönlülük ve yazılım editörü gibi birçok yönden daha üstün oluğu görülmektedir [32]. FPGA kullanımının avantajı olarak yüksek hız, esneklik, düşük maliyet ve düşük güç tüketimi, mikro denetleyicilere kıyasla zaman gecikmesinin daha az olması söylenebilir. FPGA tabanlı denetleyici, paralel ve donanım tabanlı mimarisi nedeniyle kısa bir yürütme süresine sahiptir. Bu avantajlara ek olarak, bu cihazın alternatiflerine kıyasla küçük boyutu ve düşük maliyeti vardır. Hız özelliklerinin yanı sıra algoritmalar, kolayca revize edilebilen veya iyileştirilebilen yeniden yapılandırılabilir bir platformda kodlanmıştır. FPGA, yazılım işlevselliği ve donanım performansı ile uygulamalarda hem esneklik hem de güvenlik sunmaktadır [33].

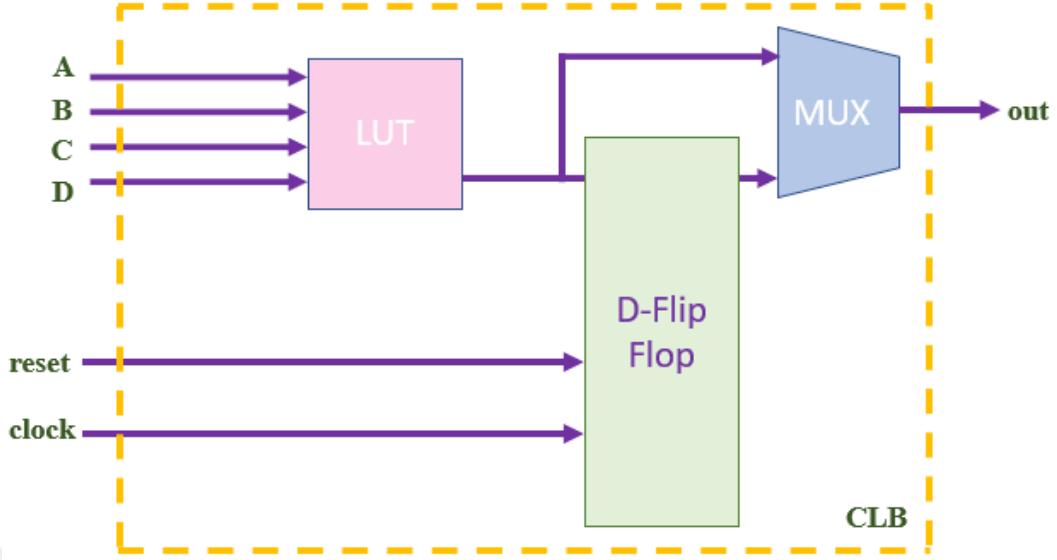
2.1.2. FPGA Yapısı

Bir FPGA donanımı, temelde mantık hücreleri (logic cell), I/O giriş çıkış hücreleri ve bağlantı bloklarından oluşur. Genel yapısı Şekil 14'te gösterilmektedir.



Şekil 14. FPGA temel yapısı

FPGA'daki yeniden yapılandırılabilir birincil yapının adı, Xilinx için yapılandırılabilir mantık bloğu (CLB), Altera için mantık ögesidir (LE) [33]. FPGA kapasitesi bu mantık hücrelerinin sayısı ile belirlenmektedir. Şekil 15'te gösterilen FPGA mimarisi basit fonksiyonları gerçekleştiren mantık hücreleri LUT (Look Up Table), çoklayıcı (MUX) ve flip-flop yapılarından oluşur.



Şekil 15. FPGA mimarisi

Mimaride yer alan yapılara ait görevler kısaca şöyle verilebilir.

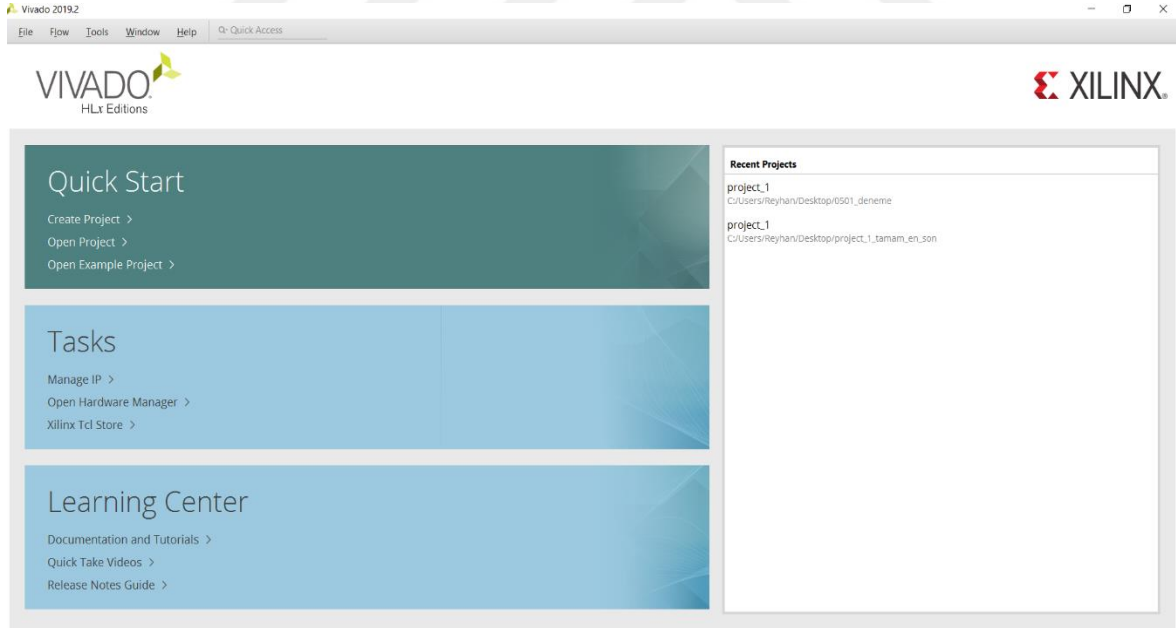
- Arama tablosu (LUT): Bu bileşen birçok mantıksal işlemi gerçekleştirir. LUT'lar mantık işlemlerini gerçekleştiren belleklerdir. Her LUT sabit sayıda girişe sahip olup N girişli LUT 2^N bellek anlamına gelir [32]. LUT'lar sıralı devreler oluşturmak için bir çoklayıcıya ve bir flip-flop'a bağlanır. Benzer şekilde, karmaşık fonksiyonlar için birkaç LUT birleştirilebilir.

- Flip-Flop (FF): Bu kayıt bileşeni, LUT sonucunu saklar.
- Veri yolları: Bu bileşenler diğer bileşenleri birbirine bağlar.
- Giriş/çıkış (I/O) : Bu portlar fizikseldir ve FPGA ile çevresinin bağlantısı giriş/çıkış blokları ile sağlanır.

Ayrıca FPGA içerisinde çok sayıda çarpma işlemi yapan DSP blokları ile farklı frekans ve fazlarda saatler üreten PLL modülleri de bulunur. Mantık hücrelerinin ara bağlantıları veri yolları ve programlanabilir anahtarlarla FPGA içerisine yüklenen program aracılığıyla ile yapılmaktadır [34].

2.2. Xilinx Vivado Ortamı

FPGA kartları üretiminde piyasanın büyük bir yüzdesi Xilinx ve Altera (Intel) firmalarına aittir [35]. Bu durum farklı üreticilerin piyasaya sunduğu FPGA kartlarının tasarımı için de farklı programların geliştirilmesini beraberinde getirmiştir. Vivado programı Xilinx şirketinin FPGA çipleri için kullanımı için uygun olurken, Quartus programı Altera şirketinin çipleri için kullanılmaktadır. Vivado, 2012 yılında ortaya çıkan sayısal sistem ve donanım tasarımı yapmak amacıyla üretilmiş bir geliştirme ortamıdır. Eski model FPGA'ler Xilinx ISE adlı program ile kullanılırken, Zynq®-7000, Kintex®-7, Artix®-7, Kintex UltraScale™ gibi çeşitlerinin ise sadece Vivado ile kullanılmasına imkan sağlanmaktadır. Vivado ortamında, Verilog, VHDL gibi donanım tanımlama dilleri ile tasarım yapılmakta, yazılan test kodları (test bence) ile benzetim sonuçları elde edilebilmektedir. Şekil 16'da gösterilen Vivado ara yüzünde tasarım, sentezleme, programlama ve gerçekleştirme yapılabilmektedir.



Şekil 16. Vivado ara yüzü ekranı

2.3. Donanım Tanımlama Dili (Hardware Description Language- HDL)

Elektronikte donanım tanımlama dili, elektronik devrelerin ve sayısal mantık devrelerinin yapısını, tasarımını ve çalışmasını tanımlamak için kullanılan özel bir bilgisayar dilidir. Bir donanım tanımlama dili, bir elektronik devrenin otomatik analizine, simülasyonuna ve simüle edilmiş testine izin veren bir elektronik devrenin gerçekleştirilmesini sağlamaktadır. HDL, bir sistemi donanımsal olarak gerçeklemeden önce tasarımdaki hataları belirlemek için kullanılmaktadır. HDL, C gibi bir programlama diline benzer ifadeler ve kontrol yapılarından oluşmaktadır. Ancak HDL zaman kavramını içermesi yönünden diğer programlama dillerinden farklıdır. Ayrıca HDL kodundaki ifadeler paralel işlem yapabilirken, diğer programlama dilleri işlemleri sırayla yapar. En çok kullanılan yazılım dilleri Verilog ve VHDL'dir.

2.3.1. VHDL Tarihçesi

FPGA üzerinde sayısal devrelerin tasarlanmasında, test edilmesinde ve sentezlenmesinde kullanılan bir donanım tanımlama dilidir. VHDL çalışmaları, 1981 civarında ABD Savunma Bakanlığı tarafından başlatıldı. IBM ve Texas Instruments gibi şirketlerin iş birliği, 1985'te VHDL'nin ilk versiyonunun piyasaya sürülmesine yol açtı. VHDL'in ilk sürümü, IEEE standartlarına uyacak şekilde oluşturuldu. 1984'te ilk FPGA'yı icat eden Xilinx, kısa sürede ürünlerinde VHDL'yi destekledi. Sonrasında VHDL, dijital devre tasarımı, simülasyonu ve sentezinde yaygınca kullanılan bir dile dönüşmüştür [35].

2.3.2. VHDL Özellikleri

Yüksek hızlı tümeşik devreler için donanım tanımlama dilinin sahip olduğu özellikler şu şekilde sıralanabilir.

- ASIC'ler, FPGA'lar gibi sayısal elektronik donanım tasarımlarının ve geleneksel sayısal devrelerin yapısını ve davranışını tanımlamaya uygundur.
- VHDL, IEEE tarafından düzenlenen uluslararası bir standarttır.

- VHDL, kullanıcıyı tek bir tasarım şekliyle kısıtlamaz. VHDL, tasarımların yukarıdan aşağıya, aşağıdan yukarıya veya ortadan dışarıya tanımlanmasına izin vermektedir.
- VHDL ile büyük karmaşık tasarımlar esnek olarak modellenmekte ve paralel işlem özelliği ile hızlı sonuçlar elde edilmektedir.

VHDL, diğer bir donanım tanımlama dili olan Verilog ile karşılaştırıldığında daha fazla satır kod yazılmasına ve daha ayrıntılı bir tasarım yapılmasına izin vermektedir. Tablo 1 'de VHDL'in Verilog ile kıyaslandığında birçok yönden üstün olduğu görülmektedir.

Tablo 1. VHDL ile Verilog karşılaştırılması

VHDL	Verilog
Kullanıcının veri tiplerini tanımlamasını sağlar.	Kullanıcının veri tiplerini tanımlamasına izin vermez.
Çok boyutlu diziyi destekler.	Çok boyutlu diziyi desteklemez.
Eşzamanlı prosedür çağrılarına izin verir.	Eşzamanlı aramalara izin vermez.
Bir mod operatörü mevcuttur.	Bir mod operatörü mevcut değil.
Tekli indirgeme operatörü mevcut değil.	Tekli indirgeme operatörü mevcut.
Öğrenmesi daha zordur.	Öğrenmesi kolaydır.
C diline benzemeyen dizime sahiptir	C diline benzer dizime sahiptir

2.3.3. VHDL Dilinin Yazım Kuralları

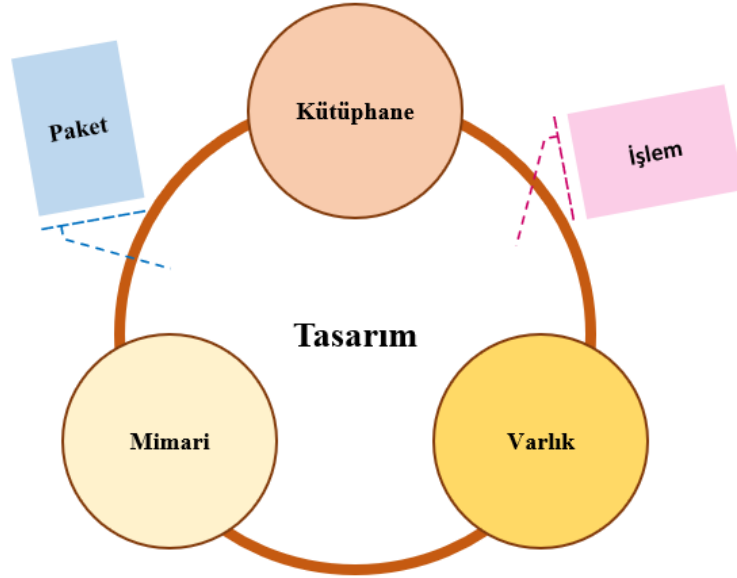
VHDL ile kodlama yapılırken diğer bütün programlama dillerinde olduğu gibi dikkat edilmesi gereken bazı önemli hususlar bulunmaktadır. Bu kurallar şu şekilde sıralanabilir.

- VHDL'de kodlar küçük harfe dönüştürülüp derlendiği için büyük harf/ küçük harf duyarlılığı bulunmamaktadır. Hem kullanıcının tanımladığı değişkenler ve komutlar hem hazır komutlar büyük/küçük harf dikkat edilmeksizin yazılabilmektedir.

- Bütün yazılım dillerinde bulunduğu gibi VHDL için de ona has kullanılan anahtar kelimeler mevcuttur. Bu dile özgü bazı anahtar kelimeler begin, with, end, entity vb. olarak örnek verilebilir.
- Yazılan kodlara açıklama yapmak istendiğinde kullanılan yorum satırları, VHDL için “—“ (çift çizgi) kullanılarak eklenmektedir.
- Yapılan atama işlemlerinden, tanımlanan ifadelerden ve kullanılan komutlardan sonra “;” (noktalı virgül) kullanılmaktadır.
- VHDL’de, FPGA kartına yüklenmeden sadece test amaçlı yazılan sentezlenemeyen kodlama yapılabildiği gibi FPGA üzerinde fiziksel karşılığı olacak şekilde sentezlenebilir kodlama da yapılmaktadır.

2.3.4. VHDL Temel Bileşenleri

Tasarım sürecinde kullanılan kütüphane (library), varlık (entity), mimari (architecture) ve olmak üzere üç temel bileşenden oluşan VHDL yapısı Şekil 17’de gösterilmektedir.



Şekil 17. VHDL bileşenleri

2.3.4.1. Kütüphane (Library)

Pek çok programlama dilinde olduğu gibi bu kısımda kullanılacak kütüphaneler tanımlanmaktadır. Böylece hazırlanan paketlerin isteyen kullanıcılar tarafından başka tasarımlarda da kolaylıkla kullanılabilmesi sağlanmaktadır. Paket içerisindeki öğelerin aktif hale gelmesi için ilk olarak “use” kelimesinin, ardından kütüphane ismi ve paketin isminin yazılması gerekmektedir. Ayrıca pakette bulunan tüm öğelerin kullanılması için de en sona “all” kelimesi eklenmektedir. Belli standartlara uygun olarak hazırlanan kütüphane ve paketler olduğu gibi tasarımcı kendine özgü tanımlamalar yapılmasına da izin verilmektedir. En güncel paket olan IEEE kütüphanesi ile yapılan bir tanımlama şu şekilde verilebilir.

```
Library ieee;
Use ieee.std_logic_1164.all;
```

2.3.4.2. Varlık (Entity)

Tasarlanan yapının ana hatlarının oluşturulduğu kısımdır. Entity, devrenin giriş ve çıkış portlarını tanımlamak için kullanılmaktadır. Bu kısımda VHDL kurallarına göre bir isimlendirme yapılmalıdır. Entity ismi bir harfle başlamak şartıyla içerisinde rakam ve alt çizgi bulunabilir. Varlık kısmı “entity deneme is” ile başlar “port” bölümünde tasarıma ait giriş ve çıkışlar belirtilir ve “end deneme” şeklinde biter. Tablo 2’de görüldüğü üzere bir portun tanımlanabileceği dört farklı şekil vardır.

Tablo 2. Port çeşitleri

Port Tanımı	Tasarımdaki Karşılığı
in	giriş portu
out	çıkış portu (veri okunamaz ama yazılabilir)
inout	hem giriş hem çıkış portu
buffer	çıkış portu (veri okunabilir ve yazılabilir)

2.3.4.3. Mimari (Architecture)

Kütüphane ve entity kısmında genel hatlarıyla oluşturulan tasarımın iç yapısı ve davranışı mimari kısmında oluşturulmaktadır. Bu kısımda tanımlama ve atama gibi işlemler yapılmaktadır. Mimari yapısı “architecture mimari_deneme of entity_deneme is” başlar, ilk kısımda çeşitli tanımlamalar yer alır, “begin” ifadesi yazıldıktan sonra ikinci kısımda atamalar ,“process” , “generate” gibi işlemler yapılır, son olarak “end mimari_deneme” ile biter. Mimari, hangi entity için tasarlanıyorsa ise oluşturulurken entity bölümüne verilen ismi ile aynı olmasına dikkat edilmelidir.

VHDL yapısında üç temel bileşenden başka tasarım ihtiyacına göre zaman zaman kullanılan Paket (Package), Bileşen (Component) ve İşlem (Process) gibi bileşenler de mevcuttur. Paket kısmında kullanıcının kendi tasarımına göre oluşturduğu veri türleri, alt programlar, bileşenler yer almaktadır.

2.3.4.4. Paket (Package)

Paket kısmında kullanıcının kendi tasarımına göre oluşturduğu veri türleri, alt programlar, bileşenler yer almaktadır. Paket sayesinde üretilen bu modüller bir araya getirilip bir modelin bütünlüğü sağlanmaktadır. Örneğin, tasarlanan sistemde herhangi bir değişiklik yapıldığında bütün tasarımcılar bu durumu anında takip edebilir.

2.3.4.5. İşlem (Process)

VHDL program satırları diğer programlama dillerindeki gibi sırayla algılanmaz. Paralel işlem yapma özelliğinden dolayı bütün satırlar aynı anda çalışmaktadır. VHDL ara yüzünde kodlama yaparken if ve for gibi ifadeler kullanılmak istendiğinde program satırlarının sırayla işlem yapması beklenmektedir. Bu duruma çözüm olarak Process yapısı oluşturulur ve işlemin satır yürütülmesi sağlanmaktadır.

VHDL programlamada farklı yapıların tanımlanabilecekleri alanlar Tablo 3’te gösterilmektedir. Tasarım yapılırken bu durum göz önüne alındığında kodlama mantık hatalarının azaltılması sağlanmaktadır.

Tablo 3. VHDL yapıları ve kullanıldıkları alanlar

	Varlık	Mimari	Paket	İşlem	Altprogram
Port	✓	✗	✗	✗	✗
Signal	✗	✓	✓	✓	✗
Variable	✗	✗	✗	✗	✓
Constant	✓	✓	✓	✓	✓
Datatype	✓	✓	✓	✓	✓
Component	✗	✓	✓	✗	✗
Subprogram	✓	✓	✓	✓	✓
Configuration	✗	✓	✗	✗	✗

2.4. VHDL Veri Nesneleri

VHDL içerisinde oluşturulan veri türündeki değerleri tutan yapılar veri nesneleridir. Tasarım yapılırken kullanılan verilerin yönetilebileceği üç tane temel veri nesnesi vardır. Program içerisine dahil edilmek istendiğinde şu biçimde tanımlanmalıdır.

veri_nesnesi_sınıfı veri_nesnesi_ismi := veri_nesnesi_türü

VHDL’ de bir veri nesnesi isimlendirilirken dikkat edilmesi gereken bazı kurallar şöyle sıralanabilir.

- Mutlaka bir harfle başlamalıdır. Harften sonra rakam ve alt çizgi “_” kullanılabilir.
- Verilecek isim VHDL programlamaya ait anahtar bir kelime olmamalıdır.
- Türkçe ’ye ait özel karakterler kullanılmamalıdır.
- Alt çizgi “_” ile bitmemelidir.
- &, ?, +, !, \$ gibi özel karakterler ve boşluk içermemelidir.

2.4.1. Sinyal (Signal) Veri Nesnesi

Bir sistemde, veri akışını açıklamak için içeriği değişebilen signal nesnesi kullanılır. Bu yapı, bir devrede ara bağlantıların fiziksel iletişimini ifade etmektedir. Sinyaller entity, mimari, paket ve blok bölümlerinde tanımlanabilir. Bir sinyal ilk değer ataması olmadan da belirtilebilir. Signal nesne tanımlaması şu şekilde yapılmaktadır.

```
signal _adı: singal_tipi_adı [:= başlangıç değeri];
```

2.4.2. Değişken (Variable) Veri Nesnesi

Değişkenler sadece tanımlandığı process, alt program ve fonksiyonlarda kullanılabilir. Signal veri nesnesi ile arasındaki en belirgin fark içeriğin anlık güncellenip değişmesidir. Bir tasarım yapılırken bellek elemanları için sinyallerin yerine variable nesnesinin kullanımı tavsiye edilmektedir. Çünkü tasarımın simülasyonu çalıştırılırken bazı durumlarda sinyallerin yerine değişkenlerin tanımlanması kullanılan bellek alanının azaltılmasını sağlamaktadır. Variable nesne tanımlaması şu şekilde yapılmaktadır.

```
variable _adı: variable_tipi_adı [:= başlangıç değeri];
```

2.4.3. Sabit (Constant) Veri Nesnesi

Tasarım süresince değeri bir kez belirlendikten sonra değiştirilemeyen nesnedir. Sabitler, kodlama yapılırken tasarım boyunca çokça yerde kullanılacak olan bir verinin değerini belirlemektedir. Bu durum programın okunabilirliğini artırmakta ve kod düzeninin sağlanmasına yardımcı olmaktadır. Ayrıca kullanıcı kendi müdahalesiyle sabitin değerini değiştirerek sistem için en uygun değere ulaşabilmektedir. Modele esnek kazandırmak için kullanılan sabitler; VHDL’de programlama yaparken entity, architecture, process, procedure, function, package, block, body gibi birçok kısımda tanımlanmasına müsaade edilmektedir. Constant nesne tanımlaması şu şekilde yapılmaktadır.

```
constant _adı: constant_tipi_adı [:= başlangıç değeri];
```

2.5. VHDL Veri Tipleri

VHDL tasarımı yapılırken veri nesnelerinin, FPGA bellek bloklarına kaydedilecek değerlerini belirlemek için veri tipleri kullanılmaktadır. Programlamada veri nesnesi tanımlanırken veri türünün de yazılması gerekmektedir. Bu durum kodun daha etkin hale getirilmesini sağlar. Yeni bir veri tipini bildirmek için “Type” komutu kullanılmalıdır VHDL veri tipleri yönünden oldukça zengin içeriğe sahiptir. Skaler, bileşik (kompozit) , access ve dosya olmak üzere başlıca dört veri tipi bulunmaktadır.

2.5.1. Skaler Tip

Tanımlandığı anda sadece bir veri değerini tutabilen nesneler için kullanılmaktadır. Tipin değer aralığını göstermek için “range”, “to”, “downto” ifadeleri yazılmalıdır. Burada “range” değer aralığının ne olacağını gösterirken; bu aralığı küçükten büyüğe doğru ifade etmek için “to”, büyükten küçüğe ifade etmek için ise “downto” kullanılır. Skaler tip sıralı (listeli) tip, fiziksel, tamsayı, reel sayı başlıklarında incelenmeye devam edecektir.

2.5.2. Sıralı (Listeli) Tip

Sembolik ifadeler kullanılarak elemanların sıralanmış değerlerinin listelenerek gösterildiği tiptir. Durum makinelerinde okunabilirliği artırmak için kullanılır. Genel gösterimi şu şekildedir.

Type tip_ismi is (liste_degerleri);

Boolean Tipi: Koşullu ifadeler oluşturmak için kullanılır. “True” (Doğru) ve “False” (Yanlış) durumları olmak üzere iki değere sahiptir. Burada sonuç “True” ise çıkış 1, “False” ise çıkış 0 olmaktadır.

Bit Tipi: Nesnelerin sadece ‘0’ ve ‘1’ değerlerini alabildiği tiptir. IEEE standart kütüphanesinde tanımlıdır.

Karakter (Character) Tip: VHDL dilinde kullanılamayan karakterlerin gösterimi için kullanılan tiptir. Karakterin üç harfle isimlendirmesi yapılır. VHDL’ de karakter setinde 256 sembol yani 8 bitlik kodlama yapılmaktadır.

2.5.3. Sayı Tipi

VHDL’ de bir nesne tamsayı veya reel sayı olarak tanımlanabilmektedir. Bu tip toplama, çıkarma, çarpma, bölme, mutlak değer, üs alma gibi matematiksel fonksiyonlarda; eşit, büyük, küçük, büyük eşit gibi karşılaştırmalı işlemlerde kullanılmaktadır.

Tamsayı Tipi: Kullanıcı tarafından range komutu ile belirtilen artan ya da azalan değer aralığındaki tamsayıları içeren bir tiptir. Type olarak tanımlanabilen bir tamsayı nesnesi için “integer” ifadesi kullanılır.

Real Tip (Kayan Nokta): Gerçel sayıları kullanmak için tanımlanan bu nesne tipi için “real” ifadesi yazılmalıdır. Sentezlenemeyen bir tip olduğu için kullanımı yaygın değildir.

Sayı tipi bir nesne tanımlanırken kullanılan sayı kümesine dikkat edilerek bellekte daha az yer kaplayacak şekilde tasarım yapılabilir. Örneğin doğal sayılar kümesi ile bir işlem yapılacaksa nesneyi “integer” yerine “natural” tipinde tanımlamak daha uygundur.

2.5.4. Fiziksel Tip

Uzunluk, sıcaklık, ağırlık, zaman gibi nicelikleri ifade etmek için tanımlanan tiptir. Sentezlenemeyip sadece simülasyon üzerinde kullanılan fiziksel tiple aritmetik ve karşılaştırmalı işlemler yapılabilir.

2.5.5. Bileşik (Kompozit) Tip

Aynı veya farklı türden nesneleri tanımlamak için kullanılan tiptir. Aşağıda bileşik tip sınıfına giren türler anlatılmıştır.

Array (Dizi) Tip: Aynı türden bir ya da daha fazla nesneyi bir arada tutarak bütün verilerin tek bir nesne yapısı olarak tanımlanmasını sağlamaktadır. Dizi boyutuna uygun indeks değeri belirtilerek tek boyutlu ya da çok boyutlu diziler tanımlanabilir. Sınırlı veya sınırsız boyutta tanımlanabilen bir dizide bulunan elemanlara birbirinden bağımsız olacak şekilde uygun türde ve genişlikte atama işlemleri yapılabilir.

bit_vector Tip: Bu dizide, bit tipinde nesneler yer alır yani dizi elemanları ‘0’ ve ‘1’ değerlerinden oluşur.

`std_logic_vector` Tip ve `std_ulogic_vector` Tip: IEEE kütüphanesinde tanımlı olan `std_logic_1164` paketinde yer alan dizi türleridir. RTL seviyesinde tasarım yapmak için gereklidir.

Record (Kayıt) Tip: Farklı türdeki nesnelerin oluşturduğu tiptir.

2.5.6. Access (Giriş) Tipi ve Dosya Tipi

Dosya tipi bir dosyadan veri okumak veya dosyaya veri yazmak için kullanılmaktadır. Texio paketi içerisinde bu dosya tipinin tanımlanmasıyla okuma yazma işlemi gerçekleştirilebilir. Access tipi, tasarım sırasında daha önce bilinmeyen değişken nesnelere ulaşmayı sağlayan veri tipidir.

2.5.7. Ön Tanımlı (Synopsys) Tip

IEEE kütüphanesi kapsamında `std_logic_1164` paketinde bulunan ön tanımlı türler `bit`, `bit_vector`, `string`, `integer`, `real`, `time`, `boolean`, `character`, `std_logic`, `std_logic_vector`, `std_ulogic`, `std_ulogic_vector` tipleridir.

IEEE kütüphanesi kapsamında `std_logic_arith` paketinde bulunan ön tanımlı türler `signed` ve `unsigned` tipleridir. İşaretli sayılar `signed`, işaretli sayılar `unsigned` ile gösterilir.

2.6. VHDL Operatörleri

VHDL ile programlama yapılırken ifadeleri oluşturmak için VHDL operatörleri kullanılmaktadır. VHDL'de aşağıdaki operatör türleri mevcuttur.

2.6.1. Mantıksal Operatörler

Program akışını kontrol etmek için mantıksal operatörler kullanımı gerekmektedir. Mantıksal operatörler sinyaller veya değişkenlerle birleştiğinde, birleşimsel mantık oluşturmaktadır. Bu nedenle veri tipleri uyumlu olmalıdır yani işlenenlerin aynı uzunluk ve tipte olmasına dikkat edilmelidir. VHDL aşağıdaki mantıksal operatörleri desteklemektedir.

- and : ve
- or : veya
- nane: ve değil
- nor : veya değil
- xor : özel veya
- xnor : özel veya değil

2.6.2. İlişkisel Operatörler

VHDL'de, aynı veri tipindeki iki işareti karşılaştırmak için ilişkisel operatörler kullanılmaktadır. Bu durumda alınan sonuç her zaman Boolean tipindedir. VHDL aşağıdaki ilişkisel operatörleri desteklemektedir.

- + : toplama
- : çıkarma
- & : ekleme
- * : çarpma
- / : bölme
- mod : mod alma
- rem : artan
- ** : üs alma
- abs : mutlak değer
- not : tersi

2.6.3. Aritmetik Operatörler

Aritmetik işlemleri gerçekleştirmek için kullanılan operatörlerdir. Burada işlem yapılan işaretler tamsayı veya gerçel sayı olabilir. İşlenen veriler aynı tipte olmalıdır. Klasik matematiksel anlam ve önceliğe sahiptir. VHDL'de aşağıda verilen aritmetik operatörler kullanılmaktadır.

= : eşittir
 /= : eşit değil
 < : küçük
 <= : küçük eşit
 > : büyük
 >= : büyük eşit

2.6.4. Kaydırma Operatörleri

VHDL'de, kaydırma operatörü, birinci işlenenin bitlerini sağa veya sola kaydırarak ve döndürerek veriler üzerinde bit manipülasyonunu gerçekleştirmek için kullanılmaktadır. VHDL aşağıdaki operatörleri desteklemektedir.

- sll :mantıksal sola kaydırma
- srl :mantıksal sağa kaydırma
- sla :shift aritmetiği sola
- sra :kaydırma aritmetiği sağa
- rol :sola döndür
- ror :sağa döndür

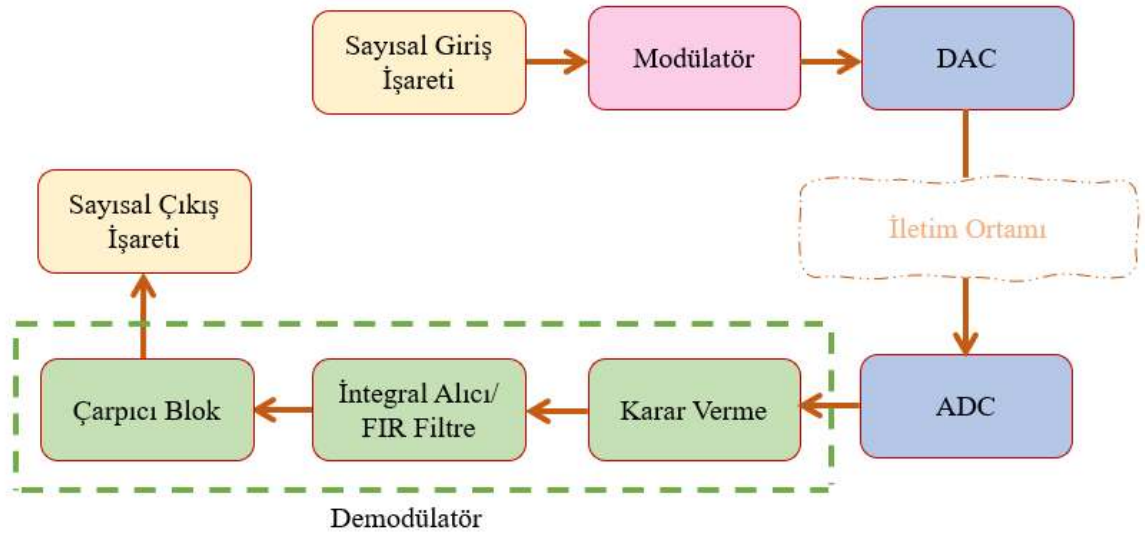
2.7. Tip Dönüşümü

Birbiriyle uyuşan veri tiplerine sahip sinyallerin birbirine ataması yapılabilmektedir. Aksi takdirde tip dönüşümü gereklidir. Bu dönüşümler birbiriyle yakın ilişkili veri tipleri için geçerlidir. Örneğin `std_logic_vector` tipindeki bir işaret integer tipine dönüştürülebilirken, tam tersi bir dönüşüm yapılamamaktadır.

3. YAPILAN ÇALIŞMALAR, BULGULAR VE İRDELEME

3.1. Giriş

Sayısal bir haberleşme sisteminin en önemli kısımları verici, iletim ortamı ve alıcı bloklarıdır. Uygulanan modülasyon yöntemine göre değişiklik göstermesine rağmen, sayısal bir bilginin bir yerden alınıp diğer bir yere aktarımı için kullanılacak sayısal bir haberleşme sistemi, genel olarak Şekil 18’de verilen yapıya sahiptir. Haberleşme sisteminde verici blokta modülasyon işlemi yapılarak sayısal bilgi işareti iletim ortamına uygun bir hale getirilmektedir. Bir sonraki adımda DAC’a iletilen sayısal modüleli işaret analog işarete dönüştürülmekte ve haberleşme ortamına aktarılmaktadır. İçerisinde iletilen bilgi işaretini barındıran analog modüleli işaret vericiden alıcıya göndermek için kullanılan fiziksel ortam haberleşme kanalı yani iletim ortamıdır. Fiziksel ortamdan gelen analog işaret ADC ile sayısal işarete çevrilmekte ve alıcı kısımda bu işaret demodüle edilerek girişteki bilgi işaretinin yeniden elde edilmesi hedeflenmektedir.



Şekil 18. Sayısal haberleşme sistemi blok yapısı

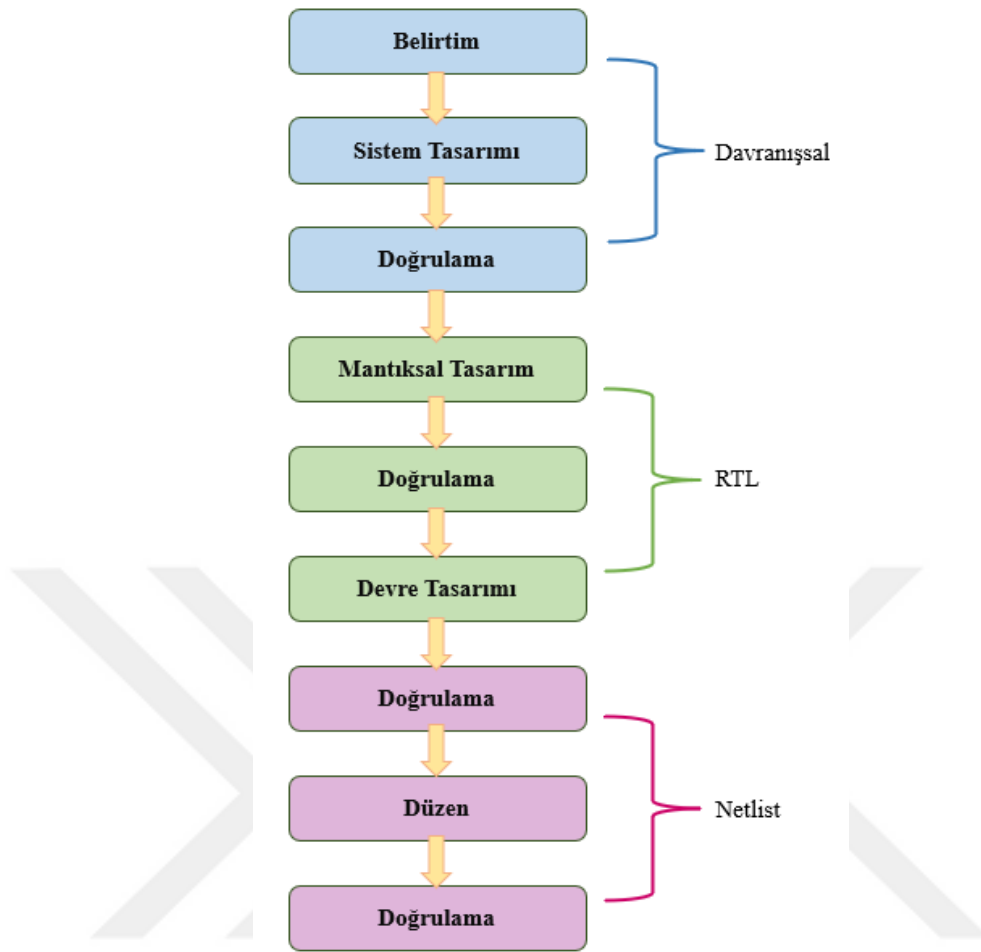
FPGA, standart ürün özelliği olan yüksek yoğunluklu ve esnek yapıli kapı dizileri, gömülü işlemciler, DSP blokları, yüksek saat frekansı gibi özelliklere sahip olmasının

yanında kısa tasarım süresi, kullanıcı tarafından programlanabilir olması nedeniyle de tercih edilmektedir. Bu kadar nitelikli özellikleri taşımasından dolayı kullanım alanı oldukça geniş olan FPGA kablolu ve kablosuz haberleşme sistemlerinde de yaygın olarak karşımıza çıkmaktadır.

Haberleşme sistemlerinde bir verinin gönderilmesi ve aynı verinin elde edilmesi için alıcı-verici yapılarının hızlı ve uygun frekansta çalışabilmeleri önemlidir. Bu tez çalışmasında, FPGA'nın yüksek clock hızı ve paralel işlem yapma özelliği sayesinde PLC dar bant standardına uygun şekilde çalışacak BPSK modülatör ve demodülatör mimarilerinin tasarımı yapılmış ve sonraki başlıklarda adım adım ayrıntılı bir şekilde açıklanmıştır.

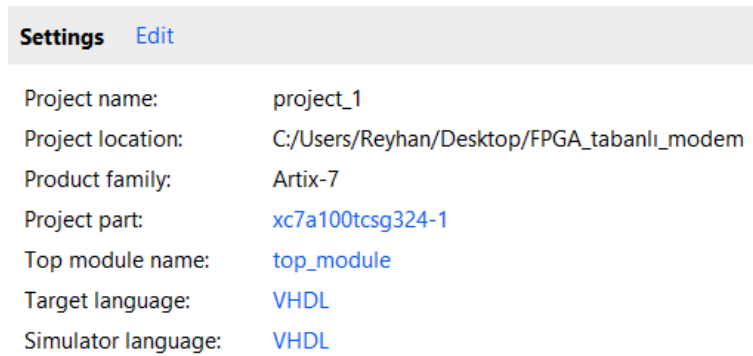
3.2. VHDL Tasarım Süreci

Vivado ara yüzünde VHDL ile kodlama yapılarak tasarlanan BPSK modem yapısı için programlama akış şeması Şekil 19'da gösterilmektedir. Belirleme aşamasında elektronik sistemin benzetimi için gereken her şey tespit edilir. Tasarımı yapılacak her bir eleman kullanılan ara yüze uygun tanımlanmalıdır. Sonraki adımda sistemin davranış tasarımı yapılır. Sistemde iç içe oluşturulacak bütün yapılar belirlenir. İç içe modüllerin bütün etkileşimleri tanımlanmalıdır. Bu adımda ara yüzde sistemin girişi ve çıkışı, saat hızı (clock), sıfırlama (reset) gibi temel taşları oluşturulur. Bu adımlar tamamlandığında bir devrenin simülasyonu gerçekleştirilebilir. Doğrulama sağlandıktan sonra mantık tasarımında (logic) önceki adımdaki tasarımların bütün sentezleri tanımlanır. Bu adımda tanımlanan kütüphanelerle FPGA yapısındaki mantık bloklarının eşlenerek çalışması sağlanmaktadır. Eşleşmeden sonra bir doğrulama adımı daha gerçekleşir. Devre tasarımı adımı yapısal tasarımın otomatik dönüştürülmesi sağlanır. Son adımda devrenin yerleşimi ara yüzdeki diğer araçlar sayesinde gerçekleştirilmiş olur. Her bir doğrulama adımı ara yüzdeki fonksiyonlar kontrol edilir, gerekli olan girişler sağlanır ve sistemin cevabı kontrol edilir.



Şekil 19. VHDL tasarım süreci

Vivado programında tasarım yapılmadan önce kodlamaya uygun ayarlamalar yapılmalıdır. Bu tez çalışmasında Şekil 20’de görüldüğü gibi FPGA kartı Nexys4DDR cihazı seçilmiş dolayısıyla Artix-7 işlemci ailesinden ve ‘xca100tcs324-1’ numaralı kısımda işlem yapılmıştır. Tasarım dili olarak da VHDL seçilmiştir.



Şekil 20. Vivado ayarları

Hazırlanan bu tezde VHDL tasarımında kullanılan IEEE standart kütüphaneleri ve kullanıcı tarafından filtre kısmı için oluşturulan kütüphane komutları aşağıdaki şekilde sıralanabilir.

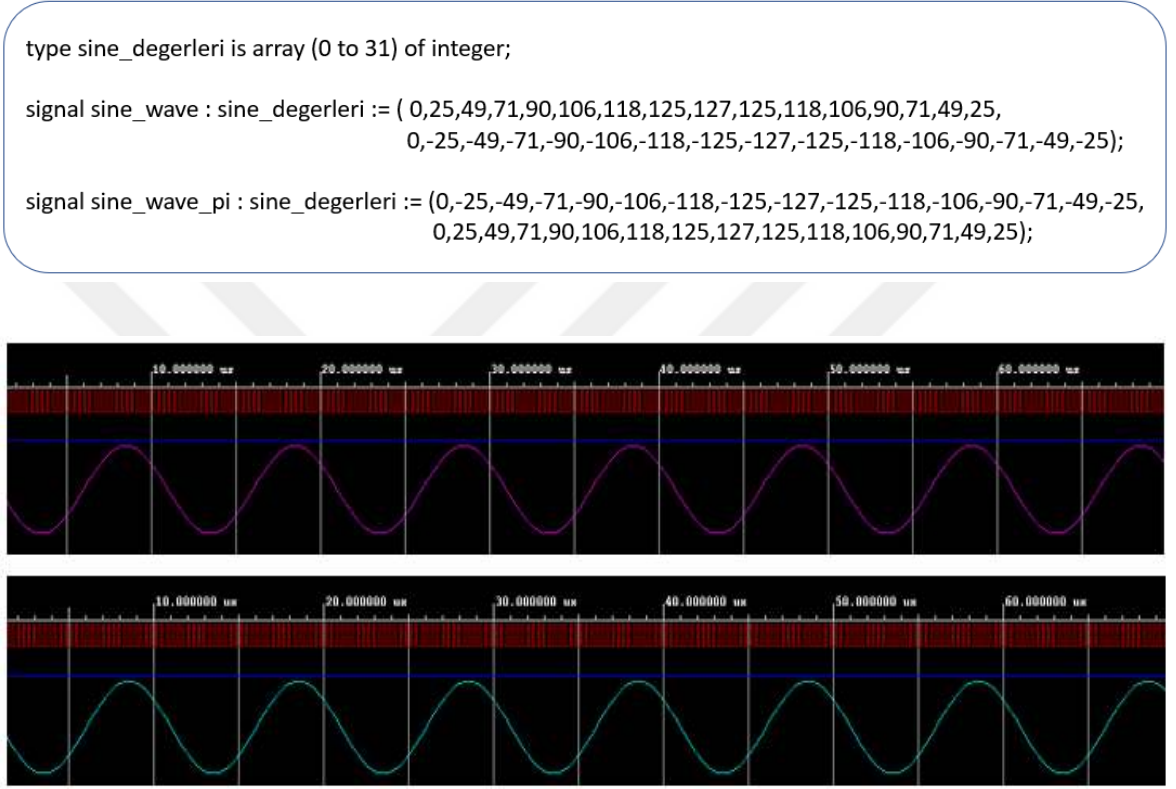
- library IEEE;
- use IEEE.std_logic_1164.all;
- use IEEE.std_logic_signed.all;
- use IEEE.numeric_std.all;
- use IEEE.std_logic_arith.all;
- use IEEE.math_real.all;
- use work.types.all;

3.3. Taşıyıcı Sinüs İşaretinin Elde Edilmesi

Bir modülasyon işleminin temeli bilgi işaretinin taşıyıcı sinyal üzerinde saklanarak iletilmesidir. Bu durum sistem içinde, modülasyonun ana unsurlarından olan taşıyıcının üretilmesini önemli bir adım haline getirmiştir. Bu çalışmada, BPSK modülasyon yöntemi ve çalışılan PLC dar bant genişliği gibi özellikler dikkat edilerek bir taşıyıcı işaret elde edilmiştir. Kullanılan Vivado ara yüzünde diğer bazı programlar gibi önceden kütüphanesine tanımlanmış hazır fonksiyonların kullanımı oldukça sınırlıdır. Örneğin bir sinüs işaretinin üretilmesi için bu fonksiyonu sağlayacak hazır komut bulunmamaktadır. Bu nedenle BPSK modülasyonunda gereken taşıyıcı sinüs işaretleri MATLAB ve VHDL kullanılarak oluşturulmuştur.

Günümüzde analog sistemlerin yerine tasarlanması daha kolay ve FPGA ile daha uyumlu olan sayısal sistemler tercih edilmektedir. Bu durum analog sinyallerden sayısal sinyallere geçişi sağlayan örnekleme teorisini göz önünde bulundurulması gerektirmektedir. Nyquist örnekleme teoremine göre sürekli bir sinyalin herhangi bir noktadaki değerini kaçırmayacak kayıpsız bir şekilde yeniden oluşturulmasını sağlayacak mümkün olan en düşük örnekleme oranı uygulanmalıdır. Bu teoremin kriterleri dikkate alınarak MATLAB ile 1 Hz frekansa sahip bir analog sinüs işaretinden, 32 ayrık değer alınarak sayısal sinüs işaretinin kayıpsız olarak elde edilmesi sağlanmıştır. Kısaca VHDL kodlamada kullanılan değerler MATLAB’de bir sinüs işaretinin 32 örneği alınarak oluşturulmuştur. Aşağıda gösterilen kod parçasında olduğu gibi integer tipinde 32 bitlik bir dizi tanımlanmıştır. Bu

dizinin her bir elemanı da 8 bitten oluşmaktadır. Burada bir referans taşıyıcı işaret ve onun 180° fazının kaydırılmasıyla diğer bir taşıyıcı işaret elde edilmiştir. Taşıyıcıların çalışılacak banda uygunluğu için 100 kHz frekansa sahip olacak şekilde programlamada clock hızı ayarlanmıştır. Bu çalışmada elde edilen taşıyıcı sinüs işaretleri Şekil 21’de gösterilmektedir.

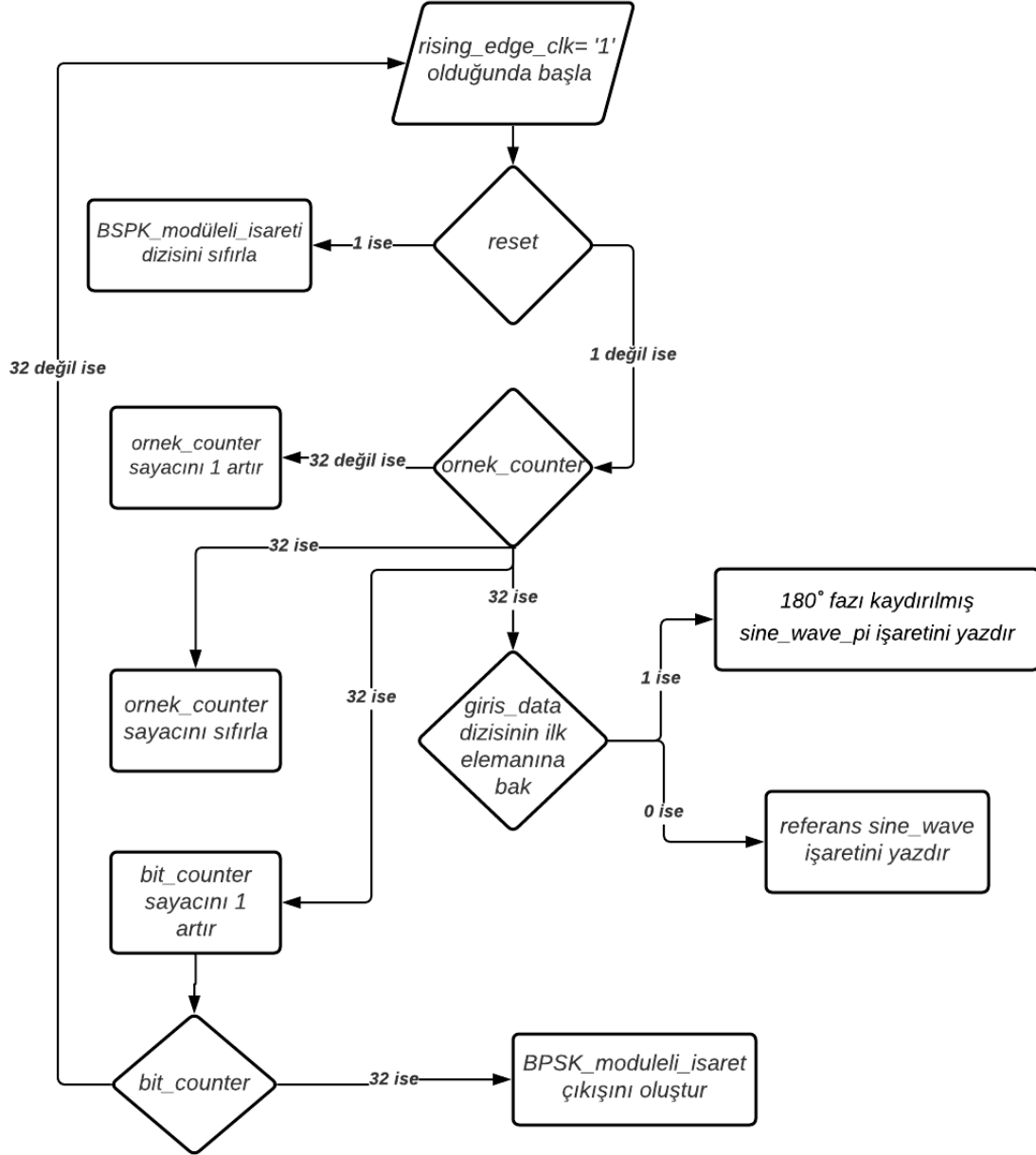


Şekil 21. BPSK modülasyonu için elde edilen sinüs işaretleri

3.4. BPSK Modülatör Mimarisi

BPSK modülatörü için aralarında 180° derece faz farkı olan iki ayrı sinüzoidal taşıyıcı sinyalin üretilmesi gerektiği bilinmektedir. BPSK modülasyonunun fiziksel bir ortamda gerçekleştirilmesi teorik kısımda bahsedilen dengeli modülatör kullanılarak mümkün olmamaktadır. Bu çalışmada VHDL ile Vivado ortamında dijital yapıda bir BPSK modülatörü tasarlanarak denge modülatörünün donanımsal olarak gerçekleştirilmesinin gerekliliği ortadan kaldırılmıştır. BPSK modülatörü VHDL ile kodlanırken process kısmında yapılan sıralı işlemler Şekil 22’de bir akış diyagramı olarak gösterilmiştir. Process bölümüne yazılan işlemlerin yapılması için dışarıdan bit tetikleme işaretinin olması

gerekmektedir. Bu işaret, üretilen veya değişime uğrayan herhangi bir sinyal olabilmektedir. Bu çalışmada tetikleme işareti olarak FPGA yapısında üretilen clock darbesi kullanılmıştır. Şekilde verilen akış diyagramında clock'un yükselen kenarı kullanarak process başlamaktadır.



Şekil 22. BPSK modülatörde process sıralı işlemi akış diyagramı

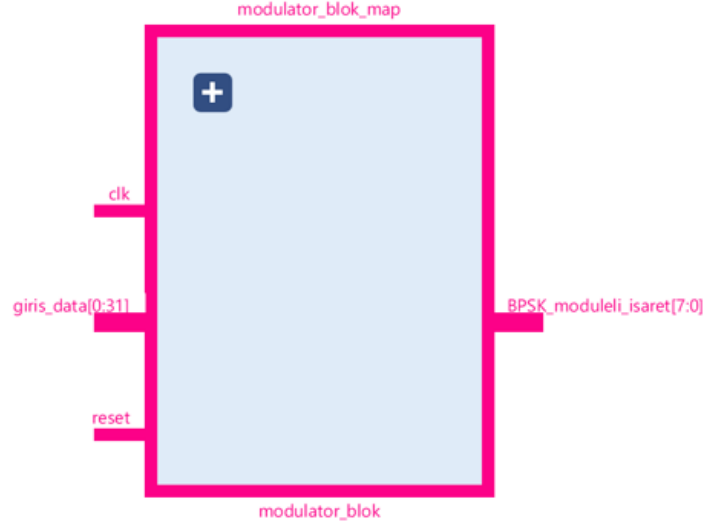
Modülatör mimarisinde blok içerisine taşıyıcı işaret olarak bir sinüzoidal işaret ve onun 180 derecelik faz farkıyla oluşturulan başka bir sinüzoidal işaret yerleştirilmiştir. Blok içinde bulunan bu taşıyıcı işaretin değerleri MATLAB ortamında örnekleme yapılarak elde

edilmiştir [36]. Bu taşıyıcı işaretlerin elde edilme süreci önceki kısımda ayrıntılı bir şekilde anlatılmıştır. Modülatör bloğunda “process” kısmının oluşturulması “clock” ve “reset” olarak tanımlanan hassasiyet ayarlarına bağlı olarak sağlanmaktadır. Process içerisinde if söz dizimi ve lop söz dizimi kullanılmıştır. Modülatör yapısında kullanılan portlar ve VHDL ile tanımlanmasında kullanılan bilgiler Tablo 4’te verilmiştir.

Tablo 4. Modülatör yapısında kullanılan portlar ve özellikleri

Port	Port Tipi	Data Tipi	Uzunluk	Tanım
clk	in	std logic	1	Tüm sisteme ait clock
reset	in	std logic	1	Modulator bloğunda çıkış sıfırlanmak istendiğinde kullanılan reset
giris_data	in	std logic vector	32	İletilecek bilgi işareti
BPSK_modüleli_isaret	out	std logic vector	8	BPSK modülatör çıkışı

Vivado ortamında tasarımı yapılan modülatör blok yapısı Şekil 23’te gösterilmektedir. Bu yapıda process kısmı clock ve reset parametreleriyle hassasiyet oluşturularak kontrol edilmektedir. Örneğin clock ’un her bir yükselen kenarında işlem başlamaktadır ve reset ’in “1” olma durumunda modüleli işaret çıkışı sıfır yapılarak temizlenmektedir. Process içerisinde if söz dizimi ve lop söz dizimi kullanılmıştır. Tablo 4’ten görüldüğü üzere sistemde iletilecek bilgi işareti 32 bitlik bir giriş verisi tanımlanmıştır. Önerilen modülatör tasarımında gelen bilgi işaretinin ilk bitinin ‘0’ veya ‘1’ gelme durumlarına karşılık hangi bit değerinin olduğu tespit edilir. Bir sonraki kısımda girişin ilk biti 0 olarak geldiğinde sistemde referans olarak üretilen sinüs işaretinin,1 olarak geldiğinde ise 180 derece fazı kaydırılan sinüs işaretinin iletilmesini sağlayacak bir anahtarlama işlemi yapılır. Giriş verisinin her biti için aynı adımlar uygulanır. Böylece blok çıkışında BPSK modülasyonlu işaret elde edilmiş olur.

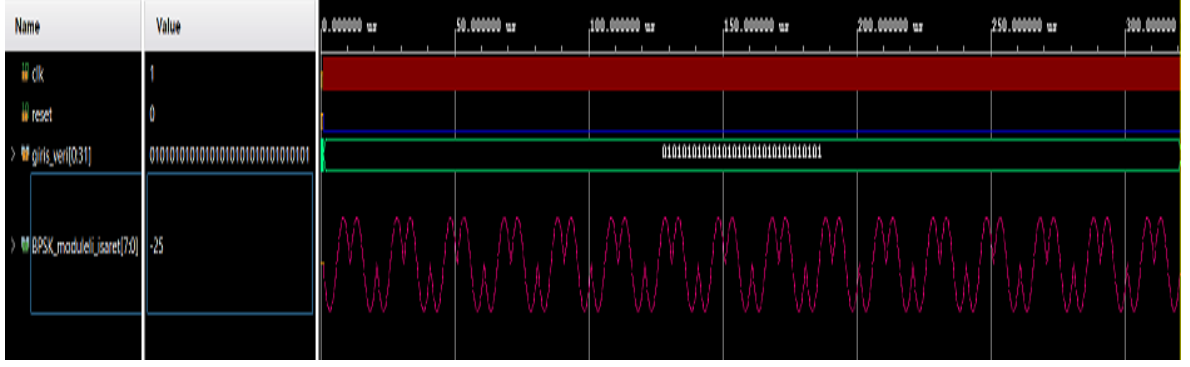


Şekil 23. FPGA tabanlı BPSK modulator schematic görünümü

3.5. BPSK Modulator Benzetimi

Tasarıma başlamadan önce Vivado ara yüzünde FPGA kartı olarak NexysDDR4 cihazı ve programlama dili olarak VHDL seçilmiştir.

Benzetim sonuçları için bilgi işareti olarak giriş_data kısmına 32 bitlik bir dizi atanacaktır. Modulator tasarımında Şekil 23'te BPSK_moduleli_isaret kısmı ikili faz kaydırma tekniği uygulanmış 8 bitlik modülasyonlu işareti göstermektedir. Alınan bilgi sinyalinin durumuna göre "1" ise modulator içine kaydedilmiş referans sinüs değerleri benzetim ekranına yazdırılmaktadır. "0" bilgi biti için ise 180 derecelik faz kaydırılmasıyla oluşturulan sinüs değerleri ekranda görülmektedir. Böylece giriş işaretinin her bir biti için 0 ve 1 olma durumuna göre uygun sinüzoidal işaret oluşmaktadır. Burada bilgi işareti giriş_data:= [01] olarak uygulanarak örnek bir benzetim yapılmıştır. Bunun sonucunda çıkış verisi olarak Şekil 24'te belirtilen 8 bitlik BPSK moduleli işaret elde edilmektedir.



Şekil 24. Örnek bir BPSK modülatör benzetim sonucu

Şekilden de görüldüğü gibi 0-1 bit değişiminin gerçekleştiği zaman anında sinyal pozitif alternanstan negatif alternansa geçiyorken, “1” bitinin gelmesi nedeniyle fazı kaydırılmamış sinüs işaretinin iletimi gerçekleştirilmiştir. Bu nedenle 0-1 bilgi biti değişiminde 180 derecelik faz geçişi sağlanmış olmaktadır.

3.6. BPSK Eşvreli Demodülasyon Tekniği ile Tasarım

3.6.1. Giriş

Programlanabilir bir sayısal sistemde BPSK modem uygulamasının en kritik kısmı demodülatör kısmının tasarımıdır. Giriş işareti olarak verilen bilgi sinyalinin, tasarlanan modülatör yapıları ile modüle edilip iletimi yapıldıktan sonra tekrar elde edilmesi için demodülasyonu işlemi gerçekleştirecek demodülatör yapıları oluşturulmalıdır.

Modülasyon işlemine benzer adımlarla demodülatör yapıları tasarlanacaktır. Demodülasyon işleminde öncelikle modüleli işaretin, referans olan taşıyıcı sinyal ile bir çarpma işlemine sokularak seviyelenmesi sağlanacaktır. Böylece bilgi işareti bitlerinin daha kolay ayırt edilmesi mümkündür. Çarpma işleminin ardından çalışılmak istenilen frekansa uygun olarak filtre tasarımı yapılacaktır. Demodülatörün son aşaması olan karar verme kısmında bilgi işaretinin keskin bir şekilde elde edilmesi gerekmektedir. Karar verme bloğunda uygun bir eşik değeri belirlenmelidir. Filtre çıkışından gelen değer eşik değerinden büyük ise bilgi işaretinin “1”, küçük ise “0” olduğu anlaşılmaktadır.

3.6.2. BPSK Eşevreli Demodülatör Mimarisi

İkili faz kaydırmalı anahtarlama demodülasyon teorisinde belirtilen adımlar baz alınarak BPSK demodülatör blok tasarımı gerçekleştirilmiştir. BPSK demodülatörün yapısı çarpıcı modülatörü bloğu, FIR filtre bloğu ve karar verici bloğundan oluşmaktadır. Modülatördeki yazılıma benzer olarak demodülatör bloğu için de “process” bölümü “clock” ve “reset” için hassasiyetlik ayarı yapılmıştır. Process içerisinde if söz dizimi ve loop söz dizimi kullanılmıştır.

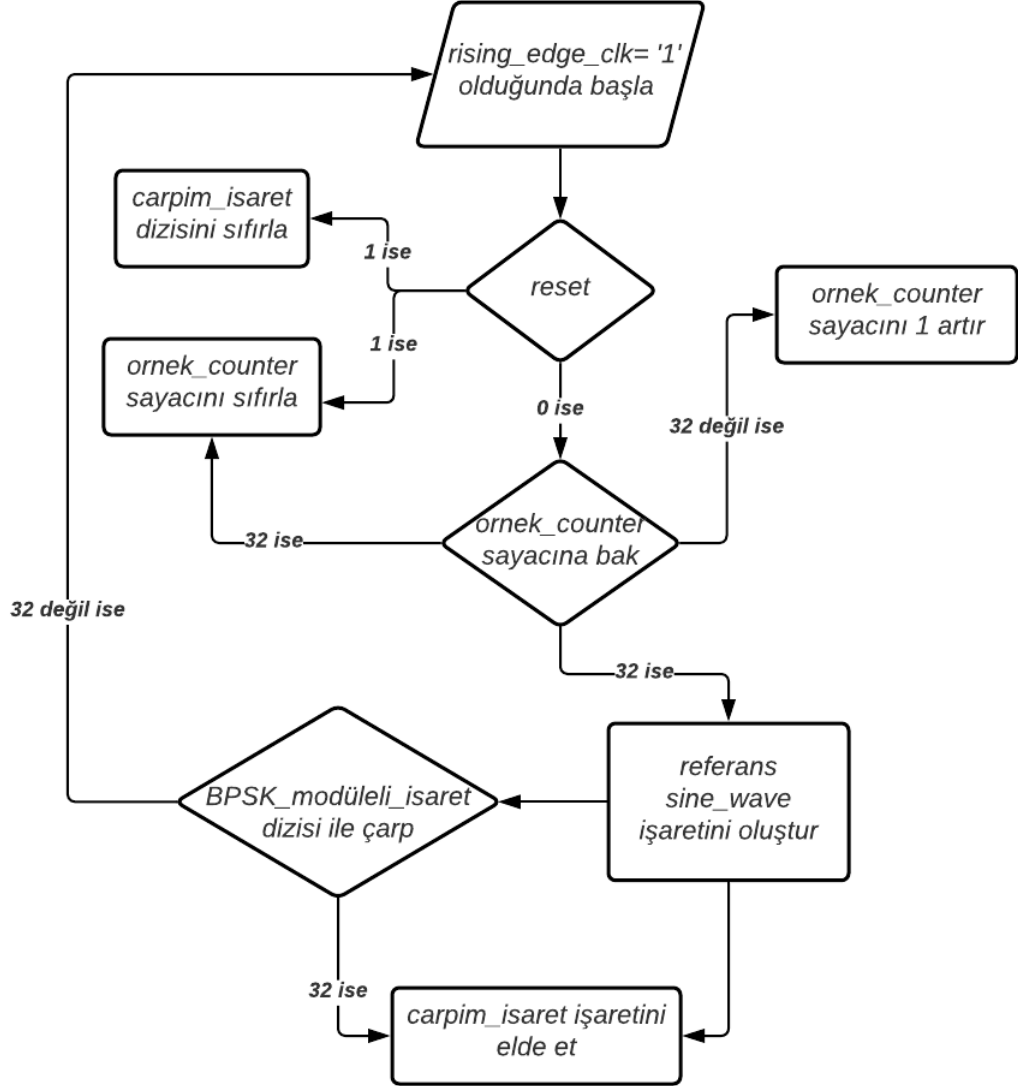
3.6.2.1. Çarpıcı Blok Mimarisi ve Benzetimi

BPSK modülatörünün çıkışında elde edilen 8 bitlik modüleli işaret ilk olarak “carpici_blok” olarak isimlendirilen çarpım modülatörünün girişine uygulanmaktadır. Çarpım modülatöründe referans olarak oluşturulan sinüzoidal taşıyıcı işaret bulunmaktadır. Bu blokta BPSK modüleli işaret ile taşıyıcı işaretin çarpımı gerçekleştirilmektedir. Aynı işlem önce MATLAB ortamında yapılarak Vivado ortamında sonucun doğruluğu sağlanmıştır. Bu mimaride bulunan portlar ve VHDL ile tanımlanmasında kullanılan bilgiler Tablo 5’te verilmektedir.

Tablo 5. Çarpıcı blokta kullanılan portlar ve özellikleri

Port	Port Tipi	Data Tipi	Uzunluk	Tanım
clk	in	std logic	1	Tüm sisteme ait clock
reset	in	std logic	1	Çarpıcı bloğunda çıkış sıfırlanmak istendiğinde kullanılan reset
BPSK_moduleli_isaret	in	std logic vector	8	Modulatörden gelen işaret
carpim_isaret	out	std logic vector	16	Çarpıcı blok çıkışı

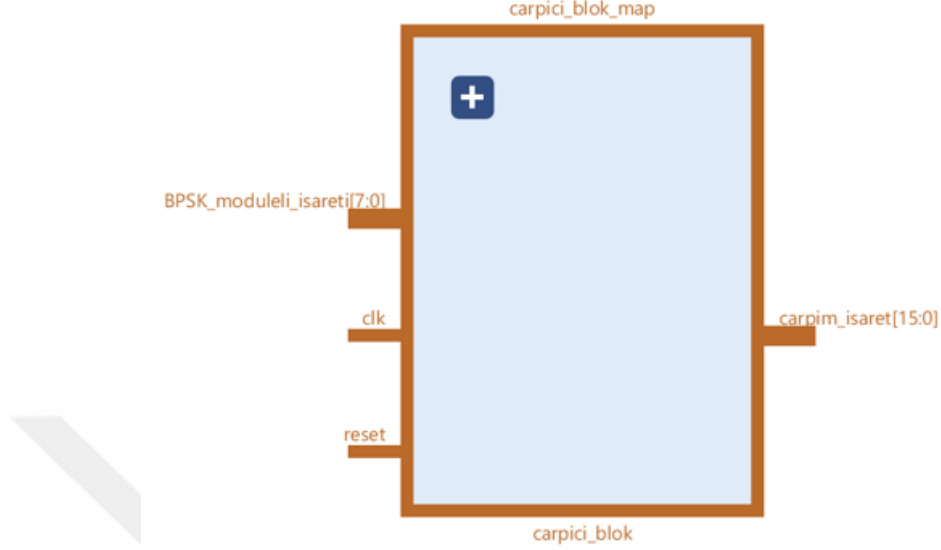
FPGA tabanlı çarpım bloğu tasarlanırken VHDL kodunda process kısmında yapılan sıralı işlemler Şekil 25’te bir akış diyagramı olarak gösterilmiştir. Bu çalışmada tetikleme işareti olarak FPGA yapısında üretilen clock darbesi kullanılmıştır. Şekilde verilen akış diyagramında clock’un yükselen kenarı kullanarak process başlamaktadır.



Şekil 25. BPSK çarpıcıda process sıralı işlemi akış diyagramı

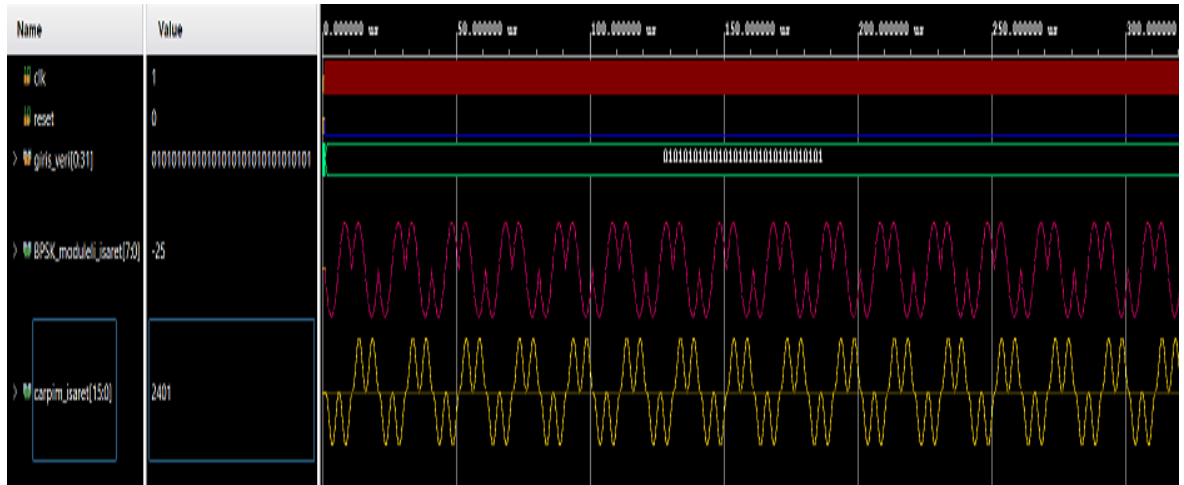
Vivado ortamında VHDL ile tasarlanan BPSK çarpım blok yapısı Şekil 26’da gösterilmektedir. Çarpıcı blok mimarisinde giriş sinyali, modulatordan elde edilen 8 bitlik modüleri işareti “BPSK_modüleri_isaret” ile temsil edilmektedir. Tasarımda sinüzoidal formdaki taşıyıcı işaretlerin her bir örnek değeri 8 bit uzunluktadır. Çarpım bloğunda bu iki

işaretin çarpılması ile blok çıkışında 16 bit uzunlukta çarpım işareti elde edilmiştir. Elde edilen bu işaret aynı zamanda FIR filtrenin girişini de oluşturmaktadır.



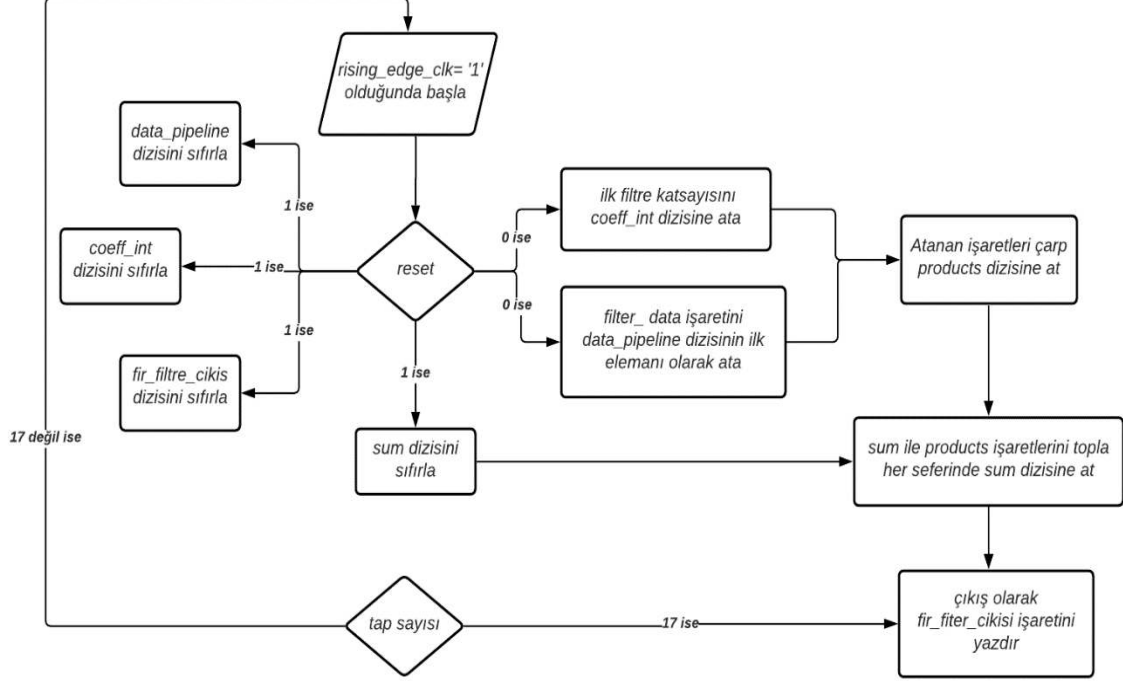
Şekil 26. FPGA tabanlı çarpıcı blok schematic görünümü

Tasarlanan çarpıcı blok için örnek olarak çarpım bloğunun girişine, giriş_data:= [01] için elde edilen BPSK_moduleli_isaret uygulanmıştır. Bu işaret ile referans taşıyıcının çarpılması sağlanarak elde edilen çıkış 16 bitlik 'carpim_isaret' olarak Şekil 27'de gösterilmektedir. Çarpıcı blok ile bilgi işaretinin, modüleli işaret içinde belirlenir seviyeye gelmesi sağlanmıştır.



Şekil 27. Örnek bir çarpıcı blok benzetim sonucu

olarak FPGA yapısında üretilen clock darbesi kullanılmıştır. Şekilde verilen akış diyagramında clock'un yükselen kenarı kullanarak process başlamaktadır.



Şekil 29. BPSK FIR filtrede process sıralı işlemi akış diyagramı

FIR filtre tasarlanırken, yapılandırılabilir özellikteki giriş verilerinin boyutu, katsayıların boyutu ve tap sayısı değerleri sabit parametreler ayarlanarak bir paket (package) tanımlanmakta ve kütüphane kısmına ‘use work.types.all’ ifadesi eklenerek kullanılabilir. Oluşturulan pakette bulunan sabitler ve VHDL ile tanımlanmasında kullanılan bilgiler Tablo 6’ da verilmektedir.

Tablo 6. FIR filtrede kullanılan sabitler ve özellikleri

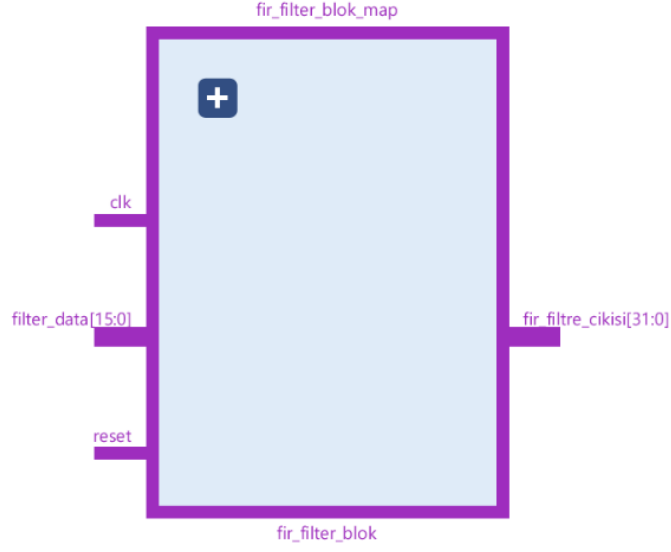
Constant	Data Tipi	Değeri	Tanım
taps	integer	17	FIR filtre terimlerinin sayısı
data_width	integer	16	FIR filtre girişinin bit uzunluğu
coeff_width	integer	4	FIR filtre katsayılar girişinin bit uzunluğu

Çarpma işleminden elde edilen 16 bitlik çarpım işareti, “fir_filter_blok” olarak isimlendirilen filtre girişine uygulanmaktadır. FIR filtre tasarımı için öncelikle MATLAB ara yüzünde ‘Window Design Method’ kullanılarak katsayı dizisi elde edilmiştir. Burada oluşturulan katsayılar fir filtre için yazılan VHDL kodu içerisine yerleştirilmiştir. Bu mimaride bulunan portlar ve VHDL ile tanımlanmasında kullanılan bilgiler Tablo 7’ de verilmektedir.

Tablo 7. FIR filtrede kullanılan portlar ve özellikleri

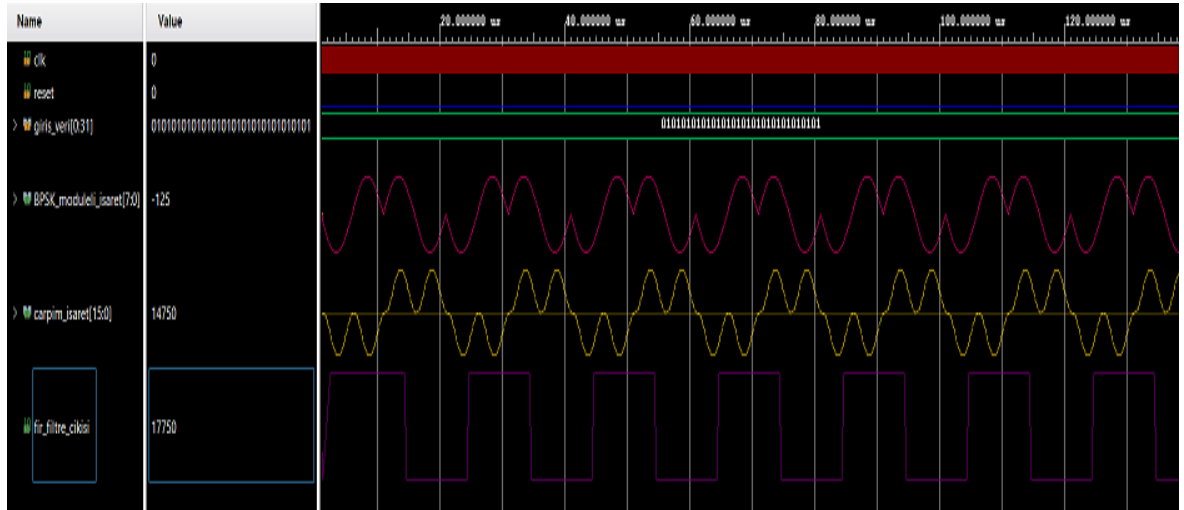
Port	Port Tipi	Data Tipi	Uzunluk	Tanım
clk	in	std logic	1	Tüm sisteme ait clock
reset	in	std logic	1	Filtre bloğunda diziler sıfırlanmak istendiğinde kullanılan reset
filter_data	in	std logic vector	16	Filtrelenecek veri
fir_filtre_cikisi	out	std logic vector	32	FIR filtre çıkışı

Vivado ortamında VHDL ile tasarlanan FIR filtre blok yapısı Şekil 30’da gösterilmektedir. Bu mimaride giriş verisi, çarpıcıdan elde edilen 16 bitlik ‘carpim_isaret’ ifadesinin yine 16 bitlik ‘filter_data’ olarak isimlendirilen porta aktarılmasıyla oluşturulmaktadır. Şekil 10’da verilen temel FIR filtre ilkesine uyularak oluşturulan Şekil 28’e göre VHDL kodlaması yapılmıştır. Çıkış işaretinin en başta uygulanan 32 bitlik bilgi işareti ile aynı bit uzunluğunda olmasına dikkat edilmiştir. Sonuç olarak filtre blok çıkışında ‘fir_filtre_cikisi’ olarak ifade edilen 32 bitlik işaret elde edilmiştir.



Şekil 30. FPGA tabanlı FIR filtre blok schematic görünümü

Tasarlanan FIR filtre yapısı için örnek olarak filtre bloğunun girişine, giriş_data:= [01010101010101010101010101010101010101] uygulanarak elde edilen carpim_isaret filter_data olarak uygulanmıştır. FIR filtre çalışma prensibine göre süzülen işaret 32 bitlik bir çıkış sinyali oluşturmaktadır. Elde edilen çıkış, Şekil 31’de ‘fir_filtre_cikisi’ olarak belirtilmektedir. FIR filtre blok ile bilgi işaretinin, modüleli işaret içinde ayırt edilecek şekilde görülmesi sağlanmıştır.



Şekil 31. Örnek bir FIR filtre benzetim sonucu

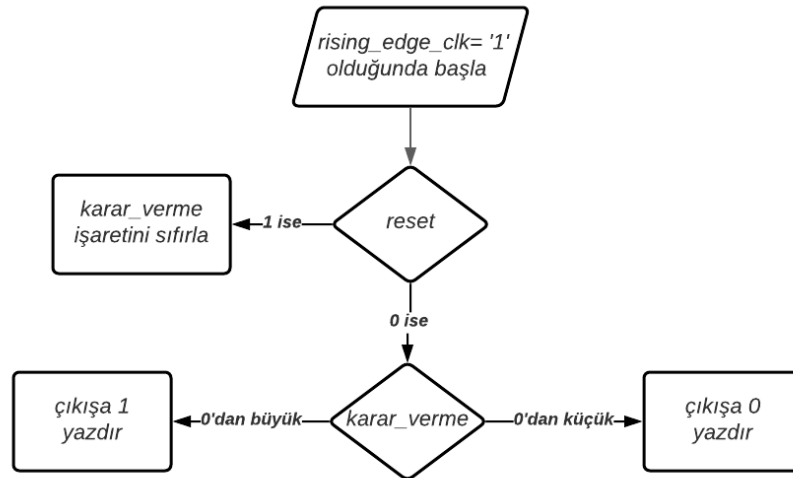
3.6.2.3. Karar Verici Blok Mimarisi ve Benzetimi

FIR filtre çıkışında elde edilen 32 bitlik işaret “filter_data” olarak isimlendirilen karar verici blok girişine uygulanmaktadır. Bu blokta gelen verinin 0 ‘dan büyük ya da küçük olmasına göre işlem gerçekleştirilmektedir. Aynı işlem önce MATLAB ortamında yapılarak Vivado ortamında sonucun doğruluğu sağlanmıştır. Bu mimaride bulunan portlar ve VHDL ile tanımlanmasında kullanılan bilgiler Tablo 8’de verilmektedir.

Tablo 8. Karar verici yapıda kullanılan portlar ve özellikleri

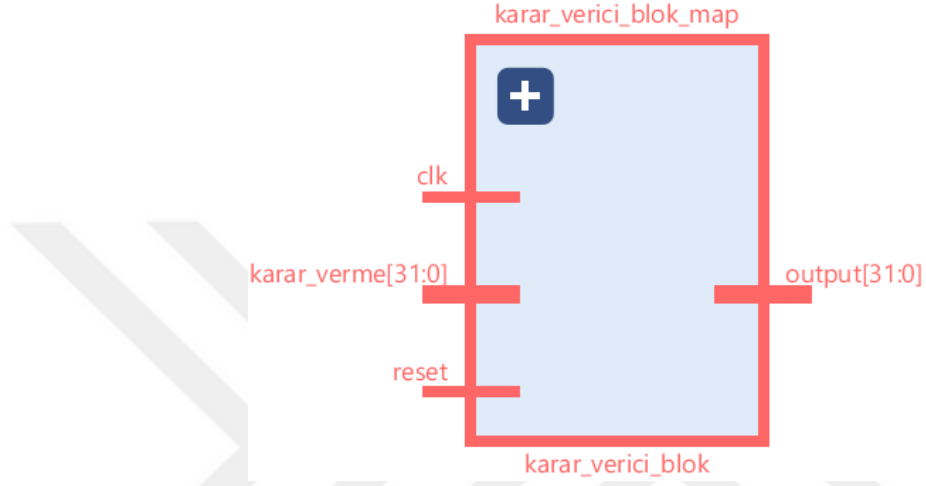
Port	Port Tipi	Data Tipi	Uzunluk	Tanım
clk	in	std logic	1	Tüm sisteme ait clock
reset	in	std logic	1	Çarpıcı bloğunda çıkış sıfırlanmak istendiğinde kullanılan reset
karar_verme	in	integer	32	Filtre çıkışından gelen işaret
output	out	integer	32	Karar verici blok çıkışı

FPGA tabanlı bir karar verici blok tasarlanırken VHDL kodunda process kısmında yapılan sıralı işlemler Şekil 32’de bir akış diyagramı olarak gösterilmiştir. Bu çalışmada tetikleme işareti olarak FPGA yapısında üretilen clock darbesi kullanılmıştır. Şekilde verilen akış diyagramında clock’un yükselen kenarı kullanarak process başlamaktadır.



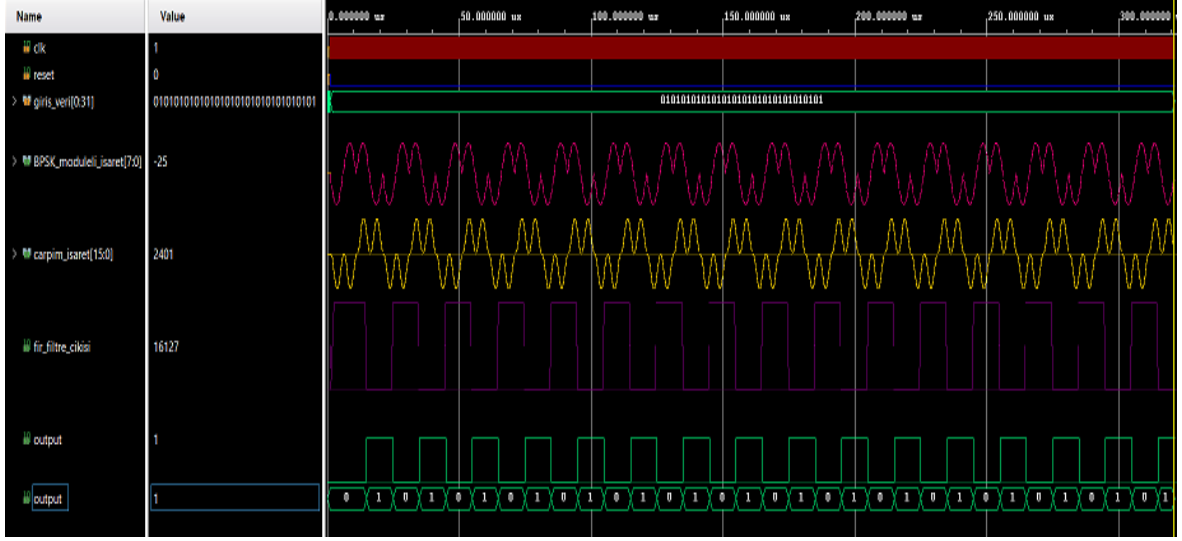
Şekil 32. BPSK karar vericide process sıralı işlemi akış diyagramı

Vivado ortamında VHDL ile tasarlanan karar verici blok yapısı Şekil 33'te gösterilmektedir. Bu mimaride giriş, filtreden elde edilen 32 bitlik 'fir_filtre_cikisi' ifadesinin yine 32 bitlik "karar_verme" olarak isimlendirilen porta aktarılmasıyla oluşturulmaktadır. Karar verme blok çıkışında 'output' olarak ifade edilen 32 bitlik işaret elde edilmiştir. Bu işaretin ilk başta sisteme uygulanan 32 bitlik bilgi işareti ile aynı olması beklenmektedir.



Şekil 33. FPGA tabanlı karar verici blok schematic görünümü

Tasarlanan karar verici blok için örnek olarak fir filtre girişine, giriş_data:= [01] için elde edilen karar_verme işareti uygulanmıştır. Bu işaretin 0 seviyesinden büyük/küçük olma durumlarına göre 32 bitlik 'output' elde edilmiştir. Karar verme bloğu ile demodülasyon işlemi tamamlanmıştır. Sisteme ilk uygulanan bilgi işaretinin (giris_data), Şekil 34'te gösterildiği gibi tekrar elde edilebildiği gösterilmiştir.

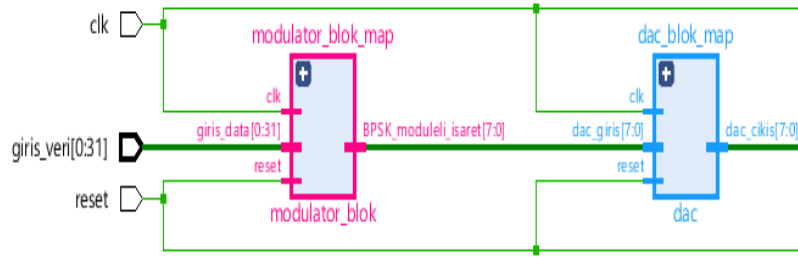


Şekil 34. Örnek bir karar verici blok benzetim sonucu

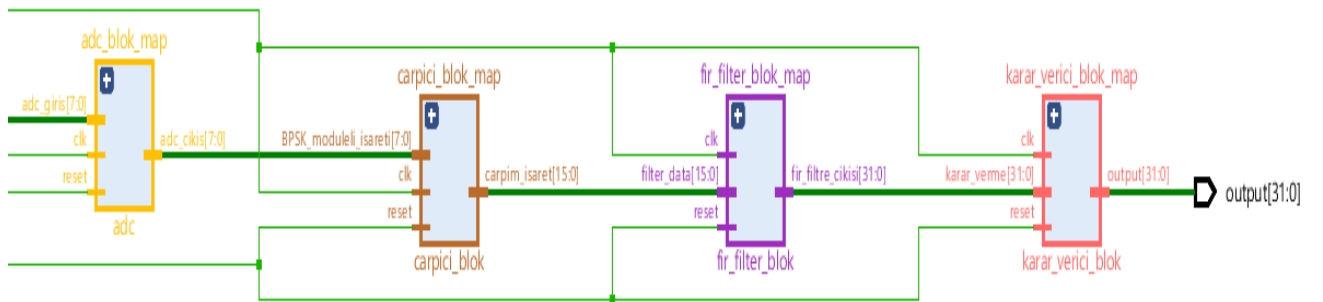
3.7. BPSK PLC MODEM Mimarisi

İkili faz kaydırmalı anahtarlama modülasyon ve demodülasyon temel teknik ilkelerine göre FPGA tabanlı modülatör ve demodülatör yapılarının tasarımı adım adım gerçekleştirilmiş ve örnek benzetim sonuçları verilmiştir. Ayrı bloklar halinde tasarlanan bu yapıların birbirileri ile bağlantısını sağlanmalıdır. Bu nedenle ‘top_module’ kaynağı oluşturulmuştur. VHDL ile önce her bir yapının portları bildirilmiş ardından her bir blok için haritalandırma yapılmıştır. Böylece Şekil 35’te ve 36’da gösterilen BPSK Modem mimarisi ortaya çıkmıştır.

Tasarlanan modemde her bir blok içinde bulunan process, clock’ un yükselen kenarı geldiğinde işlem yapmaktadır. Taşıyıcı işaretin 100 kHz frekansa sahip olması için kullanılan clock periyodu 312,5 nanosaniyedir. Sistemde 8 bitlik modüleli işaret modülatör yani vericiden DAC’a iletilmekte ve yine 8 bitlik işaret demodülatör yani alıcı tarafında ADC ile alınmaktadır. Bu durumda clock frekansı 3,2 MHz olduğu bilinmesi üzerine 8 bitlik bir işaretin iletimi sağlandığında modem data hızı 25,6 mega bit/saniye olmaktadır.



Şekil 35. FPGA tabanlı BPSK modulator schematic görünümü



Şekil 36. FPGA tabanlı BPSK demodulator schematic görünümü

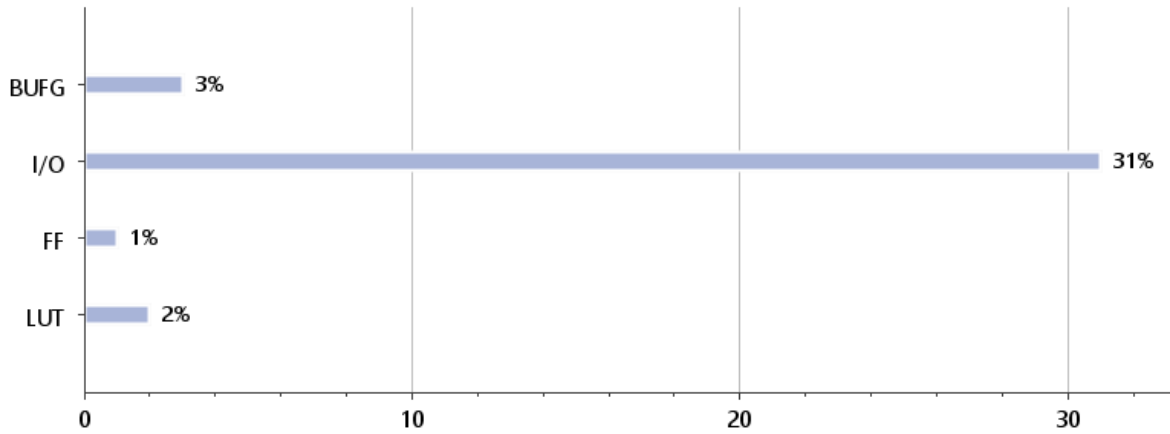
VHDL modem tasarlanırken oluşturulan blok yapılarında kullanılan hücre (cell) ve ara bağlantı (net) sayıları Tablo 9’da verilmektedir. Ayrıca tüm sistemde 1049 Look Up Table ve 409 flip-flop kullanılmıştır.

Tablo 9. Modem bloklarında kullanılan hücre ve bağlantı sayıları

Blok	hücre (cell) sayısı	ara bağlantı (net) sayısı
Modulatör	84	268
Çarpıcı	37	156
FIR Filtre	440	1309
Karar Verici	3	37

Vivado programında modem mimarisi için elde edilen benzetimlerin FPGA kartı için yerleşimi yapıldığı durumda kullanılacak global saat belleğinin (BUFG), giriş/çıkış modüllerinin (I/O), flip-flop ve başvuru çizelgesinin (LUT) kullanım yüzde oranları Şekil 37’de gösterilmektedir.

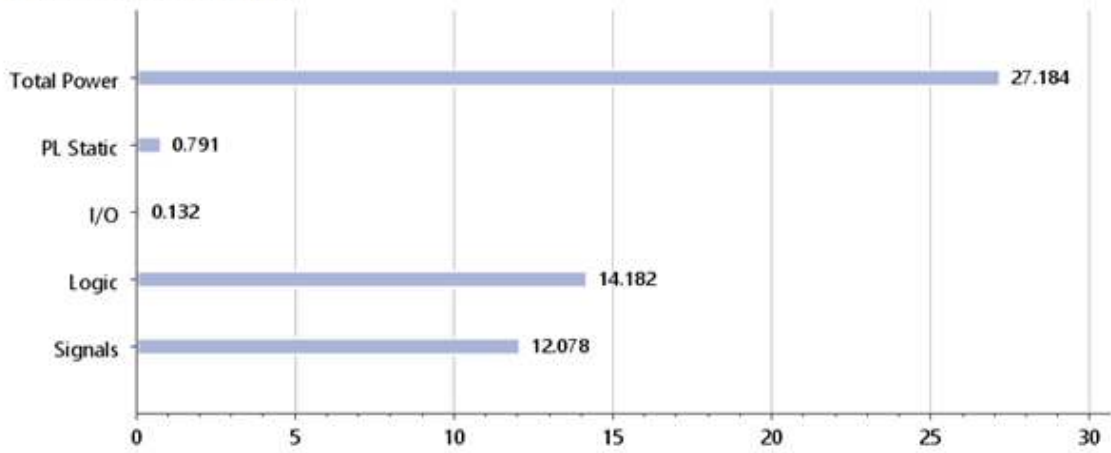
Utilization (impl_1, Place Design)



Şekil 37. FPGA yerleşiminde bazı modüllerin kullanım yüzdeleri

Tasarlanan yapının FPGA kartına yerleştirilmesi durumunda giriş/çıkış modüllerinde (I/O), mantık (logic), programlanabilir mantık kapılarında, sinyallerde güç harcamalarının yüzde oranları Şekil 38’de verildiği gibi olacağı öngörülmektedir.

Power (impl_1, Route Design)



Şekil 38. FPGA yerleşiminde harcanan güç yüzdeleri

3.8. BPSK PLC MODEM Benzetim Bulguları

VHDL ile modem mimarisi tasarlanırken kodlanan processlerin hassasiyetlikleri clock ve reset ayarları ile kontrol edilmiştir. Benzetim sonuçları için oluşturulan senaryolar bu

ayarların farklı durumları ve girişe verilen rastgele bilgi işaretleri ile elde edilmiştir. Bu senaryo durumları Tablo 10’ daki gibi oluşturulmuştur.

Tablo 10. Benzetim sonuçları için oluşturulan senaryolar

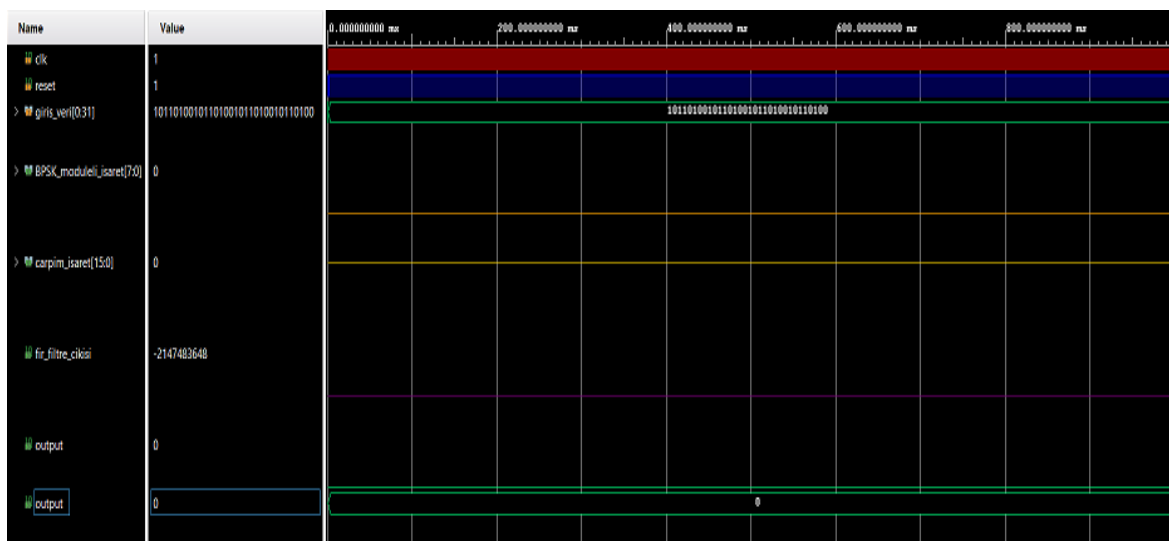
	bilgi işareti	yükselen kenar	düşen kenar	reset
1.durum	10110100101101001011010010110100	1	0	0
2.durum	10110100101101001011010010110100	1	0	1
3.durum	00100111001001110010011100100111	1	0	0
4.durum	11001010110010101100101011001010	1	0	0
5.durum	11001010110010101100101011001010	0	1	0

Oluşturulan senaryolar içinde 1.durumda process kısmında tanımlanan clock’ un yükselen kenarı 1, düşen kenarı 0 ve reset değeri 0 olarak ayarlanıp girişte giriş_veri:=[10110100101101001011010010110100] olarak uygulandığında elde edilen benzetim sonucu Şekil 39’da verilmektedir. Burada pembe renkteki işarete, sayısal bilginin 0 ve 1 gelme durumlarına göre BPSK modülasyonu gerçekleştirilmiştir. Modüleli işaretin çarpıcı bloktan geçirilmesiyle birlikte saklanan bilginin belirginleştirildiği sarı renkteki işarete ve bu sinyalin FIR filtreden süzülerek daha da bilgi sinyali seviyesine getirildiği mor renkli işarete görülmektedir. Son olarak karar verici devrede eşik değerine göre çıkışta elde edilen yeşil renkteki işaretin, başlangıçta girişte uygulanan sayısal bilgi ile aynı olduğu gösterilmiştir.



Şekil 39. rising_edge_clk = '1', trailing_edge_clk = '0' ve reset = '0' olduğunda, giriş_veri:= [10110100101101001011010010110100] için benzetim sonucu

Verilen senaryoların 2.durumunda process kısmında tanımlanan clock' un yükselen kenarı 1, düşen kenarı 0 ve reset değeri 1 olarak ayarlanıp girişte giriş_veri:=[10110100101101001011010010110100] olarak uygulandığında elde edilen benzetim sonucu Şekil 40'ta verilmektedir. Reset ayarı aktif olduğunda bütün blokların sıfırlandığı görülmektedir.



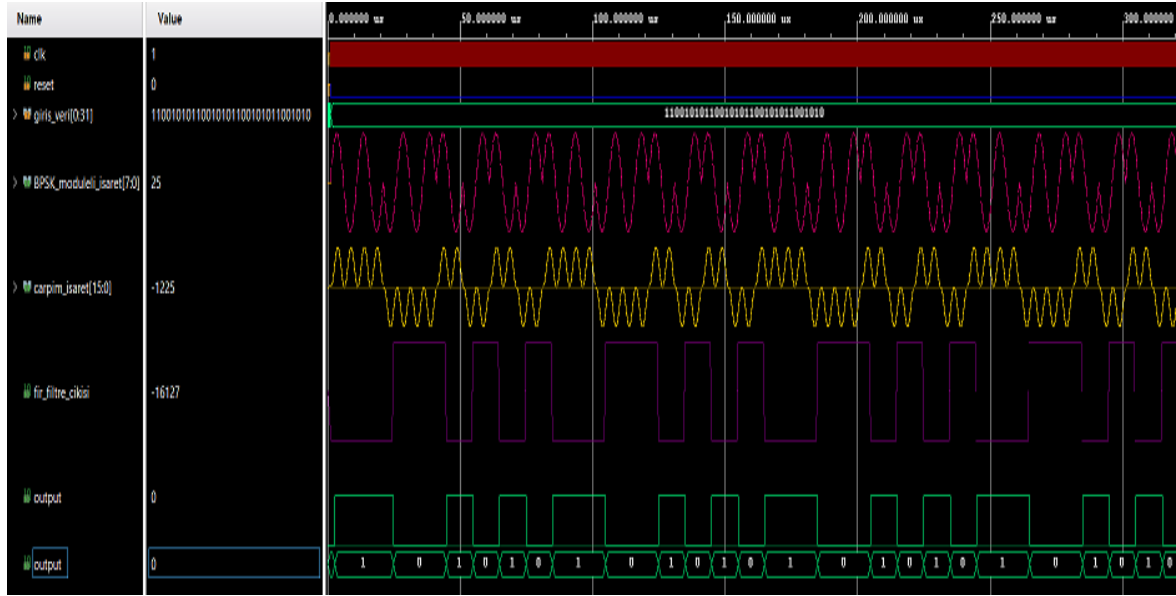
Şekil 40. rising_edge_clk = '1', trailing_edge_clk = '0' ve reset = '1' olduğunda, giriş_veri:= [10110100101101001011010010110100] için benzetim sonucu

Oluşturulan senaryoların 3.durumunda process kısmında tanımlanan clock' un yükselen kenarı 1, düşen kenarı 0 ve reset değeri 1 olarak ayarlanıp girişte giriş_veri:= [00100111001001110010011100100111] olarak uygulandığında elde edilen benzetim sonucu Şekil 41'de verilmektedir. Burada, 1.durumda verilen senaryoya girişte başka bir bilgi işareti verilerek benzetim sonucu elde edilen işaretler gösterilmektedir. Farklı bir bilgi işaretinin uygulanması ile modem test edilmiş ve sonuç başarılı olmuştur.



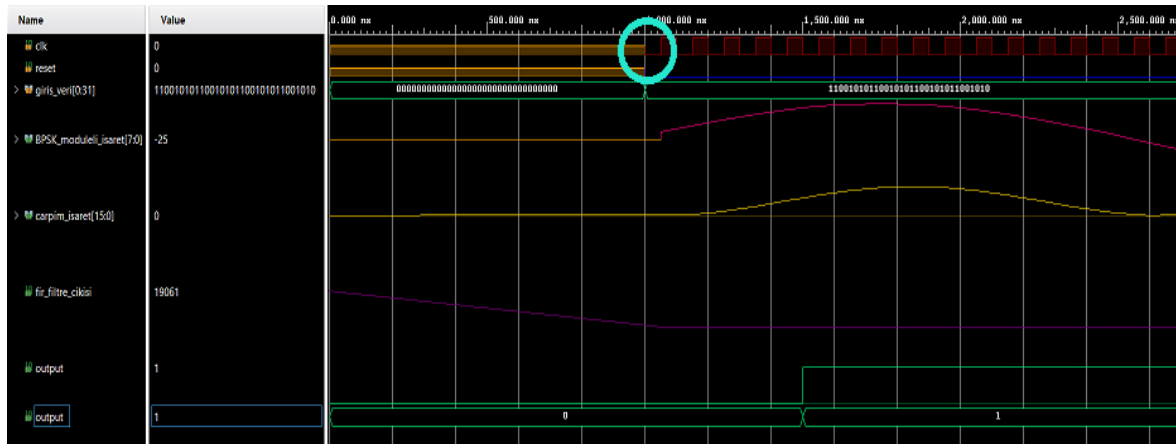
Şekil 41. rising_edge_clk = '1' , trailing_edge_clk = '0' ve reset = '0' olduğunda, giriş_veri:= [00100111001001110010011100100111] için benzetim sonucu

Senaryoların 4.durumunda process kısmında tanımlanan clock' un yükselen kenarı 1, düşen kenarı 0 ve reset değeri 1 olarak ayarlanıp girişte giriş_veri:= [11001010110010101100101011001010] olarak uygulandığında elde edilen benzetim sonucu Şekil 42'de verilmektedir. Burada, 1. durumda ve 2.durumda verilen senaryoya girişte başka bir bilgi işareti verilerek benzetim sonucu elde edilen işaretler gösterilmektedir. Farklı bir bilgi işaretinin uygulanması ile modem test edilmiş ve sonuç başarılı olmuştur.



Şekil 42. rising_edge_clk = '1' , trailing_edge_clk = '0' ve reset = '0' olduğunda, giriş_veri:= [11001010110010101100101011001010] için benzetim sonucu

4. durumda verilen bilgi işaretinin aynı kalarak process hassasiyet ayarında clock'un yükselen kenarı 0, düşen kenarı 1 yapılarak oluşturulan 5. durum için elde edilen benzetim çıktısı Şekil 43'te gösterilmektedir. VHDL tasarımı process işlemi, clock yükselen kenarı ile başlamakta yani tasarlanan bloklardaki işlemler Şekil 44' te görüldüğü gibi paralel olarak yürütülmektedir. Clock düşen kenarı 1 değerini aldığı Şekil 43'te görüldüğü üzere process işlemi başlamamakta ve bir saat darbesi kadar bekletilerek diğer bir clock darbesinin yükselen kenarında haberleşme işlemleri başlatılmaktadır.



Şekil 43. rising_edge_clk = '0' , trailing_edge_clk = '1' ve reset = '0' olduğunda, giriş_veri:= [11001010110010101100101011001010] için benzetim sonucu



Şekil 44. rising_edge_clk = '1', trailing_edge_clk = '0' ve reset = '0' olduğunda, giriş_veri:= [11001010110010101100101011001010] için benzetim sonucu

4. SONUÇLAR VE TARTIŞMA

Bu tez çalışmasında, iletilmek istenilen bir sayısal bilgi işaretinin, PLC dar bant kanalına uyumlanabilir olarak üretilen 100 kHz frekansa sahip bir taşıyıcı sinüs ile BPSK yöntemine göre modüle edilerek iletim ortamına uygun hale getirilmesi ve sonrasında demodüle edilerek başlangıçta gönderilen bilgi işaretinin tekrar elde edilmesini gerçekleştirecek bir modem tasarlanmış ve benzetimi gerçekleştirilmiştir. Modem tasarımına sistem benzetim çalışmaları MATLAB ve Xilinx destekli Vivado ortamlarında gerçekleştirilmiştir. MATLAB tabanlı sonuçlar, referans tasarım olarak kabul edilen bir simülasyon çalışmasıdır ve FPGA platformunda VHDL tercih edilerek tasarlanan mimarilerin benzetim sonuçlarının doğruluğunu göstermek için kullanılmıştır.

Çalışmanın ilk aşamasında modülasyon işleminin gerçekleştirilmesi için gerekli olan sinüs formundaki taşıyıcı işaret MATLAB’de örnekleme kriterlerine göre 32 değer olacak şekilde oluşturulmuş ve bu örnek değerleri VHDL ile kodlamada kullanılmıştır. Böylece FPGA tabanlı modülatör mimarisi için 100 kHz frekansta referans taşıyıcı işaret ve yine aynı frekansta 180° fazı kaydırılan diğer sinüs taşıyıcısı üretilmiştir. 32 bitlik bir sayısal bilgi girişi bulunan ve 8 bitlik bir BPSK modüleli işaretin elde edildiği önerilen modülatör mimarisi Şekil 23’te verilmiştir. Process hassasiyetinin clock ve reset ile sağlandığı modülatör bloğunun gerçekleştirilen benzetimi Şekil 24’te görüldüğü gibidir.

Çalışmanın ikinci aşamasında iletilen BPSK modüleli işaret üzerinden tekrar bilgi işaretini elde etmek için demodülatör tasarımı yapılmıştır. Eşvreli BPSK demodülasyon teorisine göre oluşturulan FPGA tabanlı mimari çarpıcı blok, FIR filtre blok ve karar verici blok yapılarını içermektedir. Her bir blok içerisindeki process hassasiyeti yine clock ve reset ile ayarlanmıştır. Öncelikle 8 bit olarak gönderilen modüleli işaret çarpıcı blok yapısına giriş olarak verilmiştir. Şekil 26’da görülen çarpıcı blok içerisinde modüleli işaret ile referans sinüs çarpılarak 16 bitlik bir çarpım işaret edilmiş ve Şekil 27’deki benzetim sonucunda belirtildiği üzere yapılan işlem ile bilgi işareti daha seçilebilir duruma getirilmiştir. Ardından 16 bitlik çarpma sonucu FIR filtre bloğuna giriş olarak uygulanmıştır. Şekil 30’da gösterilen blokta filtre giriş verisinin, ona denk gelen katsayı değeri ile çarpılıp toplama işlemine aktarılması ve her seferinde verinin geciktirilerek yine gecikmeyle ona karşılık gelen katsayıyla çarpılması ve her seferinde toplama işleminin bir önceki toplama sonucuna eklenerek bu işlemin de filtrenin tap sayısı kadar toplamaya devam edilmesiyle filtreleme

işlemi gerçekleştirilmektedir. Şekil 31'deki benzetim sonucunda verildiği üzere yapılan filtreleme ile bilgi işareti daha da bilgi işaretine benzer olarak seviyelenmiştir. Sonra filtrelenen işaret eşik seviyesi sıfır olarak belirlenen karar verici bloktan geçirilerek sıfırdan büyük değerlerin 1, sıfırdan küçük değerlerin 0 sayısal bilgi sinyallerinin elde edilmesi sağlanmıştır. Karar verici blok çıkışı ile demodülasyon işlemi tamamlanmış ve en başta giriş verisi olarak tanımlanan 32 bitlik sayısal dizi yeniden elde edilmiştir. Şekil 34'te benzetim sonucunda görüldüğü gibi tasarımın doğruluğu kontrol edilmiştir.

Çalışmanın üçüncü aşamasında ise bir haberleşme sistemi yapısının gerektirdiği ADC ve DAC bloklarının sanal olarak eklenmesiyle tasarlanan modülatör ve demodülatör yapılarının, alıcı verici durumları belirtilerek haritalandırma yapılmış ve BPSK PLC modem mimarisi tamamlanmıştır. Bu yapının test edilmesi için Tablo 10'da anlatılan farklı senaryolar uygulanmış Şekil 39, 40, 41, 42, 43 ve 44'te görüldüğü gibi benzetimlerde farklı durumların sonuçları incelenmiş ve girişe uygulanan bilgi işaretleri başarı ile tekrar elde edilmiştir.

Yapılan literatür taramasına göre, paralel ve hızlı işlem yapma, yeniden yapılandırılabilme gibi avantajlara sahip olmasına rağmen FPGA tabanlı modemlerin, PLC kanalında uygulamalarına pek fazla rastlanılmamıştır. Bu çalışmada, günümüzde kullanılan PLC modemlerin yerini alabilecek tamamen sayısal işlemlerin gerçekleştirildiği PLC kanalında uyumlu olarak modellenen adaptif bir BPSK modem tasarımı ve benzetimi yapılmıştır. BPSK modülasyon ve demodülasyon tekniğine göre Vivado ortamında VHDL ile modülatör/demodülatör tasarlanarak modem fiziksel devre yapısında yer alan denge modülatörünün donanımsal olarak gerçekleştirilmesinin gerekliliği ortadan kaldırılmıştır. Ayrıca yine demodülatör kısmının içinde yer alan integral alıcı, FIR filtre, karar verici sabit devre yapılarının fiziksel olarak gerçekleştirilme karmaşıklığı ve maliyeti yok edilerek tamamen sayısal tabanlı bir tasarım yapılmıştır. CENELEC standartlarına uygun frekanslarda haberleşme yapabilecek şekilde tasarlanan bu BPSK-PLC modem, donanımsal tasarımda gerekli olan yapıların kullanımını gerektirmeme ve Vivado ortamında VHDL kodlamayla tasarlanması ile ekonomik, esnek ve uyarlanabilir olma gibi avantajlara sahiptir.

5. DEĞERLENDİRME VE ÖNERİLER

PLC dar bandında kullanılmak üzere FPGA tabanlı bir BPSK modem tasarımından elde edilen sonuçlara göre önerilen modemin FPGA kartı üzerinde deneysel olarak gerçekleştirilmesi durumunda kendine özgü avantaj ve dezavantajları olabileceği düşünülmektedir. Örneğin, mimarının FPGA platformuna yerleştirilmesi halinde düşük SNR değerinde yüksek BER sağlayabilmek amacı ile modülasyon yöntemi olarak BPSK tercih edilmiştir. Ancak BPSK yöntemi az miktarda veri iletimi yapmak istenildiğinde kullanılan PSK modülasyon türüdür. Böylece FPGA ile yapılan bu tasarımda kullanılan basit yapılı PSK tekniği olan BPSK modülasyonunun deneysel çalışmalar için de uygun bir yöntem olduğu düşünülebilir.

VHDL ile tasarlanan BPSK PLC modem, paralel işlem gücünün yüksek olması ve esnek yapıya sahip olması gibi avantajlarından dolayı FPGA’da gerçekleştirilerek deneysel çalışmalar yapılabilir. Böylece gelecek çalışmalarda, tasarım gereksinimlerine ve donanım kullanılabilirliğine göre farklı yaklaşımlar seçilerek diğer sayısal modülasyon teknikleri ile farklı senaryolar oluşturularak istenilen özelliklerde modem mimarileri oluşturulabilir.

6. KAYNAKLAR

1. Hasirci, Z., Cavdar, İ.H. ve Öztürk, M., Modeling and Link Performance Analysis of Busbar Distribution Systems for Narrowband PLC, Radioengineering, 2017, C.26, 611-620.
2. Prakash S., Bansal, A. ve Jha, S. K., Performance analysis of narrowband PLC system under Gaussian Laplacian noise model. 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), 2016, 3597-3600, Chennai: IEEE.
3. Popescu, S.O., Gontean, A.S. ve Budura, G., Simulation and Implementation of a BPSK Modulator on FPGA, 6th IEEE International Symposium on Applied Computational Intelligence and Informatics, May 2011, Timisoara Romania, 459-463.
4. Popescu, S.O., Gontean, A.S. ve Budura, G., Performance Comparison of the BPSK and QPSK Modulation Techniques on FPGA, 17th International Symposium for Design and Technology in Electronic Packaging (SIITME), October 2011, Timisoara, Romania, 257-260.
5. Erdoğan, C., Myderrizi, I. ve Minaei, S., FPGA Implementation of BASK-BFSK-BPSK Digital Modulators, IEEE Antennas and Propagation Magazine, C. 54, No.2, April 2012, 262-269.
6. Jammu, B.R., Botcha, H.K., Sowjanya, A.V. ve Bodasingi, N., FPGA Implementation of BASK-BFSK-BPSK-DPSK Digital Modulators using System Generator, International Conference on Circuits Power and Computing Technologies [ICCPCT], April 2017, Kollam, India.
7. Chye., Y. H., Ain, M. F. ve Zawawi, N. M., Design of BPSK Transmitter Using FPGA with DAC, 9th Malaysia International Conference on Communications (MICC), December 2009, Kuala Lumpur Malaysia, 451-456.
8. Sourabh, K.S., Lekshmi, V., Sudhakar, S., ve Manikandan, J., FPGA Implementation of BPSK Demodulator using Model based Programming, 2020 IEEE International Conference for Innovation in Technology (INOCON), November 2020, Bengaluru, India, 1-6.
9. Krivić, P. ve Štimac, G., FPGA Implementation of BPSK Modem for Telemetry Systems Operating in Noisy Environments, MIPRO 2011, May 2011, Opatija, Croatia, 1727-1731.

10. Wu, L., Cui, X. ve Yu, D., Design and Implementation of a BPSK Acoustic Modem for Underwater Communication, IET International Conference on Information Science and Control Engineering (ICISCE), December 2012, Shenzhen.
11. Karthikalakshmi, B., Anand, Dr.M., ve Vijaykanth, An Autonomous Reconfiguration of FPGA with Embedded Arm, 2nd International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB16), February 2016, Chennai, India.
12. Sourabh, K.S, Deepika, R. ve Lekshmi, V., Sudhakar, S., ve Manikandan, J., Design and Evaluation of BPSK Demodulator using Model based Programming, 16th India Council International Conference (INDICON), December 2019, Rajkot, India.
13. Nivin, R., Sheeba Rani, J. ve Vidhya, P., Design and Hardware Implementation of Reconfigurable Nano Satellite Communication System Using FPGA Based SDR for FM/FSK Demodulation and BPSK Modulation, International Conference on Communication Systems and Networks (ComNet), July 2016, Trivandrum, 1-6.
14. Safi., A. A. ve Bazuin, B., FPGA Based Implementation of BPSK and QPSK Modulators using Address Reverse Accumulators, 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), October 2016, New York USA.
15. Tachwali, Y. ve Refai, H., Implementation of a BPSK Transceiver on Hybrid Software Defined Radio Platforms, Information and Communication Technologies: From Theory to Applications (ICTTA), May 2008.
16. Thasleem Sulthana, A. K., Simulation and Implementation of BPSK Modulator and Demodulator System on Spartan-3E FPGA, Second International Conference on Smart Systems and Inventive Technology (ICSSIT 2019), November 2019, India, 126-128.
17. Belen., S., Dijital Filtering for Communication Signals Using FPGA Technology, M.Sc Thesis, Dokuz Eylül University, Graduate School of Natural and Applied Sciences, October 2010, İzmir.
18. Hander., A.I.K., Design and Implementation of an Educational am Receiver with FPGA Using SDR Techniques, M.Sc Thesis, Karabük University, Institute of Graduate Programs, January 2021, Karabuk.
19. Satılmış., G., Designing a PLC Modem for Smart Grid Applications, M.Sc Thesis, YILDIZ Technical University, Graduate School of Natural and Applied Sciences, June 2015, İstanbul.

20. Kuo., K.C., Guo., J.W. ve Ou., Y.H., A Fully Digital Modulator/Demodulator for Power Line Communication (PLC), IEEE Asia Pacific Conference on Circuits and Systems, December 2010, Malaysia, 835-838.
21. Mishra., D. K. ve Saini., L. M., CPLD Based FSK Modem for Narrowband Power Line Communication, 2nd International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), February 2016, India.
22. Chen H., Baba E., Ishio K., ve diğerleri, A Novel PLC Physical Layer Specification - Overview, System Simulation and FPGA Lab Test Results, The 1st IEEE Global Conference on Consumer Electronics 2012, October 2012, Tokyo/ Japan, 343-346.
23. Bali., M. C. ve Rebai., C., Advanced DSP Based Narrowband PLC Modem for Smart Grids Applications, IEEE Intl Conf on High Performance Computing and Communications, IEEE 6th Intl Symp on Cyberspace Safety and Security, IEEE 11th Intl Conf on Embedded Software and Syst (HPCC,CSS,ICSS), August 2014, Paris/France, 725-728.
24. Moaveninejad, S., Kumar, A., Scazzoli, D., Piti, A., Magarini, M., Bregni, S., ve Verticale, G., BER evaluation of post-meter PLC services in CENELEC-C band, 9th Latin-American Conference on Communications (LATINCOM), 2017, 1-6.
25. Esmailian, T., Kschischang, F.R. ve Gulak, P. G. ,Characteristics of Inbuilding Power Lines at High Frequencies and their Channel Capacity, Proceeding of ISPLC 2000, 52-59, Limerick.
26. Mathur, A. ve Bhatnagar, M.R., PLC Performance Analysis Assuming BPSK Modulation Over Nakagami- m Additive Noise. IEEE Communications Letters ,2014, 909 - 912.
27. Hagen, M., Heminger, M., ve Mohammed, A., Power Line Communication for Lighting Applications Using Binary Phase Shift Keying (BPSK) with a Single DSP Controller, Twenty-First Annual IEEE Applied Power Electronics Conference and Exposition, 2006, 6.
28. Prasad, B.K.V. ve Priya, R. S., Implementation and Reconfiguration of Basic Digital Modulation Design Models. Journal of Theoretical and Applied Information Technology, 2016, 208.
29. Nair L. S. ve Arun K.L., Simulation of BASK, BPSK, BFSK Modulators using Verilog, International Journal on Cybernetics & Informatics (IJCI), C.5, No.2, April 2016, 365-376.

30. Yadav., A., Digital Communication, 188, Firewall Media, 2009, 71-72.
31. Değermenci, T., Enerji Hatları Üzerinden Delta Modülasyonlu Ses İletimi, Yüksek lisans tezi, Karadeniz Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Mayıs 2007, Trabzon.
32. Sarıtaş, E. ve Karataş, S., Her Yönüyle FPGA ve VHDL,3. Baskı, 411, Palme Yayıncılık, Ankara, 2015.
33. LaMeres B.J., Introduction to Logic Circuits & Logic Design With VHDL, Springer, Switzerland, 2017.
34. Dereli, S., FPGA ile Gömülü Sistemler ve Sayısal Devre Tasarımı, 1.Baskı, 249, Nobel Yayıncılık, Ankara, Şubat, 2020.
35. Savran., İ., Donanım Tanımlama Dili VHDL ve FPGA Uygulamaları, 1.Baskı, 306, Papatya Bilim Yayıncılık, Ocak, 2017, İstanbul.
36. Sönmez, M., Çok Yüksek Hızlı Tümlleşik Devre Donanım Tanımlama Dili Kullanılarak Gerçek Zamanlı Modülatör-Demodülatör Tasarımı, Doktora Tezi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, 2016, Elazığ.

ÖZGEÇMİŞ

Reyhan SAĞ, İlköğretim ve ortaöğretimini Lâdik' te tamamlamış olup, 2017 yılında Karadeniz Teknik Üniversitesi Elektrik-Elektronik Mühendisliği Bölümünden mezun oldu. Aynı sene Elektronik Mühendisliğinde yüksek lisans eğitimine başladı ve 2019 Nisan ayında Karadeniz Teknik Üniversitesi Elektronik Anabilim Dalında araştırma görevlisi olarak atandı. Orta derecede İngilizce bilen SAĞ, elektronik, haberleşme FPGA alanlarında çalışmalarını sürdürmektedir.

YAYINLARI

1. R.SAĞ ve diğerleri , "FPGA TABANLI DAR BANT BPSK - PLC MODEM TASARIMI VE BENZETİMİ," 2. ULUSLARARASI 19 MAYIS YENİLİKÇİ BİLİMSEL YAKLAŞIMLAR KONGRESİ , ss.1110-1117, 2019.

(Kabul edildi)