T.C.
İSTANBUL ÜNİVERSİTESİ-CERRAHPAŞA
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

GENİŞ BANTLI RF SPEKTRUM ANALİZİ İÇİN FFT TABANLI GERÇEK ZAMANLI SPEKTRUM ANALİZÖRÜ TASARIMI

Mehmet ESKİN

DANIŞMAN
Prof. Dr. Fırat KAÇAR

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Elektrik-Elektronik Mühendisliği Programı

İSTANBUL-2020

Bu çalışma, 6.01.2020 tarihinde aşağıdaki jüri tarafından Elektrik-Elektronik Mühendisliği Anabilim Dalı, Elektrik-Elektronik Mühendisliği Programında Yüksek Lisans tezi olarak kabul edilmiştir.

Tez Jürisi

Prof. Dr. Fırat KAÇAR (Danışman)
İstanbul Üniversitesi-Cerrahpaşa
Mühendislik Fakültesi

Doç. Dr. Yasin ÖZÇELEP
İstanbul Üniversitesi-Cerrahpaşa
Mühendislik Fakültesi

Doç. Dr. Serap ÇEKLİ
Maltepe Üniversitesi
Mühendislik ve Doğa Bilimleri Fakültesi

20.04.2016 tarihli Resmi Gazete'de yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince; Bu Lisansüstü teze, İstanbul Üniversitesi-Cerrahpaşa'nın abonesi olduğu intihal yazılım programı kullanılarak Lisansüstü Eğitim Enstitüsü'nün belirlemiş olduğu ölçütlere uygun rapor alınmıştır.

ÖNSÖZ

Yüksek Lisans tez çalışmalarım boyunca değerli bilgi ve deneyimlerinden yararlandığım danışman hocam Prof. Dr. Fırat Kaçar'a teşekkürlerimi sunarım.

Bu tez çalışması, "Profen İletişim firmasının geliştirilmekte olan 3171050 nolu Uydu Haberleşmesi için Çok Kullanıcılı Uzaktan Erişimli Gerçek Zamanlı Spektrum Analizör Geliştirmesi (GEZSAN)" adlı TEYDEB projesinden referans alınarak hazırlanmıştır.

Ayrıca çalışmalarım boyunca bana destek olan başta iş arkadaşlarım Mustafa DİREK ve Mert Külte olmak üzere bütün Profen İletişim Teknolojileri mesai arkadaşlarıma, yüksek lisans eğitimim boyunca sağlamış olduğu imkânlar ve vermiş olduğu desteklerinden dolayı Profen İletişim Teknolojileri firmasına teşekkür ederim.

Bu tezi her zaman yanımda olan ve desteğini hiç eksik etmeyen aileme ve sevgili Eşim **Ece'ye** ithaf ediyorum.

Ocak 2020 Mehmet ESKİN

İÇİNDEKİLER

Sayfa No

ÖNSÖZ ..... **iv**
İÇİNDEKİLER ..... **v**
ŞEKİL LİSTESİ ..... **vii**
TABLO LİSTESİ ..... **x**
SİMGE VE KISALTMA LİSTESİ ..... **xi**
ÖZET ..... **xiii**
SUMMARY ..... **xv**
1. GİRİŞ **1**
2. GENEL KISIMLAR **3**
2.1 ARM MİMARİSİ 3
2.2 GÖMÜLÜ LİNUX NEDİR? 4
2.3 FPGA NEDİR? 5
2.4 SOC NEDİR? 7
2.5 SPEKTRUM ANALİZÖR VE SPEKTRUM ANALİZ TEKNOLOJİSİ 8
2.5.1 Spektrum Analizör Nedir? 8
2.5.2 Spektrum Analizör Teknolojisi 8
2.5.2.1 Swept-Tuned (Süpürülmüş) Spektrum Analizörleri 9
2.5.2.2 FFT Tabanlı Spektrum Analizörler 10
2.5.2.3 Gerçek Zamanlı Spektrum Analizörler 12
3. MALZEME VE YÖNTEM **14**
3.1 SİSTEM GENEL MİMARİSİ 14
3.1.1 RF Alıcı Modülü 15
3.1.2 FPGA ve Mikroişlemci Birimleri 15
3.1.3 Spektrum Analiz Yazılımı 17
3.2 GÖMÜLÜ YAZILIM GELİŞTİRME ORTAMI 18
3.2.1 Gömülü Linux Derleme 20
3.2.2 Gömülü Linux Derleme ve Cihaz Sürücüsü Ekleme 21
3.2.3 USB Sunucu ve Veri Aktarımı 23
3.3 FPGA GELİŞTİRME VE SAYISAL SİNYAL İŞLEME 24
3.3.1 FMC Arayüz Bloğu 25

3.3.2 FFT 26
3.3.2.1 FFT Algoritması 26
3.3.2.2 FFT IP Bloğu 32
3.3.3 Pencereleme (Windowing) 34
3.3.3.1 Pencereleme Algoritması 34
3.3.3.2 Pencereleme IP Bloğu 38
3.3.4 Üst Üste Bindirme 39
4. BULGULAR 42
4.1 SPEKTRUM ANALİZÖR SİSTEMİNİN ÇALIŞMASI VE DEĞERLENDİRME 42
4.2 FIR FİLTRE IP BLOĞU DEĞERLENDİRMESİ 43
4.3 PENCERELEME IP BLOĞUNUN DEĞERLENDİRMESİ 44
4.4 FFT IP BLOK DEĞERLENDİRMESİ 50
5. TARTIŞMA VE SONUÇ 57
6. KAYNAKLAR 59
7. ÖZGEÇMİŞ 61

ŞEKİL LİSTESİ

Sayfa No

Şekil 2.1: Arm Mimari Ailesi [4]. 4

Şekil 2.2: FPGA Yapısı [7]. 6

Şekil 2.3: Masaüstü Spektrum Analizör [10]. 8

Şekil 2.4: Süpürülmüş spektrum analizörü [12]. 9

Şekil 2.5: Klasik FFT spektrum analizör blok diyagramı [11]. 11

Şekil 2.6: Gerçek zamanlı spektrum analizör blok diyagramı [11]. 12

Şekil 3.1: Zedboard geliştirme kartı üstten görünümü [13]. 16

Şekil 3.2: ZYNQ7000 mimarisi [14]. 17

Şekil 3.3: Spektrum grafiksel gösterim yöntemi [15]. 18

Şekil 3.4: Gömülü Linux Genel Mimarisi 20

Şekil 3.5: Petalinux-config yapılandırma paneli 22

Şekil 3.6: Sistem FPGA katmanı genel mimarisi 24

Şekil 3.7: Sinyal işleme IP blokları 25

Şekil 3.8: Frekans bant çözünürlüğü etkisi [17]. 26

Şekil 3.9: Zaman alanı ve frekans alanı arasındaki ilişki [12]. 27

Şekil 3.10: FFT Algoritması Kelebek Diyagramı [19]. 28

Şekil 3.11: FFT ayrıştırması. N noktalı sinyal N tane tek noktaya sahip sinyallere ayrıştırılır [20]. 29

Şekil 3.12: FFT bit ters sıralama işlemi [20]. 29

Şekil 3.13: FFT Sentezleme [20]. 30

Şekil 3.14: FFT Kelebek algoritması [20]. 31

Şekil 3.15: FFT akış diyagramı [20]. 31

Şekil 3.16: Xilinx FFT bloğu Radix-4 Mimarisi [21]. 33

Şekil 3.17: Pencereleme algoritması [22]. 34

Şekil 3.18: Hanning –Hamming pencereleme algoritmaları kıyaslaması [22]. 35

Şekil 3.19: Dikdörtgen Pencereleme ve Fourier dönüşümü [23]. 36

Şekil 3.20: Hanning Pencereleme ve Fourier dönüşümü [23] 36

Şekil 3.21: Hamming Pencereleme ve Fourier dönüşümü [23]. 37

Şekil 3.22: Flat top Pencereleme ve Fourier dönüşümü [23]..... 37

Şekil 3.23: Blackman-Harris Pencereleme ve Fourier dönüşümü [23]. 38

Şekil 3.24: Üst üste bindirme uygulanmamış iki FFT karesi arasındaki sinyal olayı [24]. 40

Şekil 3.25: Üst üste bindirme uygulanmış FFT kareleri arasındaki sinyal olayı [24]. 40

Şekil 3.26: Yüzde 75 üst üste bindirme [25]..... 41

Şekil 3.27: Üst üste bindirme yöntemi (Yukardan aşağıya %0, %75, %97, %99.9 üst üste bindirme uygulanmıştır) [25]. 41

Şekil 4.1: Matlab ile üretilen iki tonlu sinyalin Xilinx FFT modeli çıktısı 42

Şekil 4.2: FIR filtre (1/4) IP bloğu frekans cevabı 43

Şekil 4.3: FIR filtre (1/2) IP bloğu frekans cevabı 44

Şekil 4.4: Matlab ve Tasarlanan Hann pencereleme algoritması sonuçları 45

Şekil 4.5: Matlab ve Tasarlanan Hamming pencereleme algoritması sonuçları 45

Şekil 4.6: Matlab ve Tasarlanan IP bloğunun Blackman-Harris pencereleme sonuçları 46

Şekil 4.7: Matlab ve Tasarlanan IP bloğunun Flattop pencereleme sonuçları 46

Şekil 4.8: Rectengular Pencereleme uygulanmış sinyalin FFT sonucu..... 48

Şekil 4.9: Hann Pencereleme uygulanmış sinyalin FFT sonucu 48

Şekil 4.10: Hamming Pencereleme uygulanmış sinyalin FFT sonucu 49

Şekil 4.11: Blackman-Harris Pencereleme uygulanmış sinyalin FFT sonucu 49

Şekil 4.12: Flattop Pencereleme uygulanmış sinyalin FFT sonucu 50

Şekil 4.13: Radix-2 Mimarisi kaynak kullanımı 51

Şekil 4.14: Radix-4 Mimarisi Kaynak Kullanımı 51

Şekil 4.15: Pipelined Mimarisi kaynak kullanımı 52

Şekil 4.16: Radix-2 mimarisine sahip FFT IP bloğunun gecikme değerleri 52

Şekil 4.17: Radix-4 mimarisine sahip FFT IP bloğunun gecikme değerleri 53

Şekil 4.18 : Pipelined mimariye sahip FFT IP bloğunun gecikme değerleri 53

Şekil 4.19: Giriş bant genişliği (25 bit) Matlab FFT (üstte) ile Xilinx FFT (altta) kıyaslaması 55

Şekil 4.20: Giriş bant genişliği (20 bit) için Matlab FFT (üstte) ile Xilinx FFT (altta) kıyaslaması 55

Şekil 4.21: Faz faktör bit genişliği etkisi 56

TABLO LİSTESİ

Sayfa No

Tablo 3.1: Petalinux SDK ile gömülü işletim sistemi tasarım adımları [16]. 21

Tablo 3.2: Pencereleme Algoritmaları karşılaştırma tablosu [10]. 35

Tablo 4.1: Pencereleme Yöntemleri katsayı kıyaslama tablosu 47

SİMGE VE KISALTMA LİSTESİ

Simgeler	Açıklama
T	: örnekleme periyodu
f_s	: Örnekleme frekansı
N	: Örnek sayısı
BW	: Bant Genişliği

Kısaltmalar	Açıklama
ADC	: Analog Dijital Çevirici (Analog Digital Converter)
AXI	: Gelişmiş Uzatılabilir Arayüz (Advanced eXtensible Interface)
BRAM	: Blok Rasgele Erişim Belleği (Block Random Access Memory)
CISC	: Karmaşık Komut Seti Bilgisayarı (Complex Instruction Set Computer)
CORDIC	: Koordinat Döndüren Sayısal Bilgisayar (Coordinate Rotation Digital Computer)
CPU	: Merkezi İşlemci (Central Processing Unit)
dB	: Desibel
DC	: Doğru Akım (Direct Current)
DFT	: Ayrık Fourier Dönüşümü (Discrete Fourrier Tranformu)
DMA	: Doğrudan Bellek Erişimi (Direct Memory Access)
DSP	: Sayısal Sinyal İşlemcisi (Digital Singal Processor)
FFT	: Hızlı Fourier Dönüşümü (Fast Fourirer Transform)
FGPA	: Alanda Programlanabilir Kapı Dizileri -Field Programmable Gate Array
FIFO	: İlk Giren İlk Çıkar (First In, First Out)
FIR	: Sonlu Dürtü Yanıtı (Finite Impulse Response)
FMC	: FPGA Ara Kat Kartı (FPGA Mezzanine Card)
Gbit/s	: Gigabit / saniye
GPIO	: Genel Amaçlı Giriş Çıkış(General Purpose Input Output)
GPU	: Grafik İşlemcisi (Graphical Processing Unit)
HDF	: Donanım Tanımlama Dosyası (Hardware Description File)
HDL	: Donanım Tanımlama Dili (Hardware Description Language)
Hz	: Hertz

IF	: Ara Frekans (Intermediate Frequency)
IP	: Intellectual Property
JTAG	: Ortak Test Eylem Grubu (Joint Test Action Group)
kSPS	: Saniyede Bin adet Örnek (Kilo Sample Per Second)
LO	: Yerel Osilatör (Local Oscillator)
LUT	: Taramalı Tablo (Look-Up Table)
MB	: Milyon Byte (Mega Byte)
Mbit	: Milyon bit (Mega bit)
MHz	: Mega Hertz
MMU	: Bellek yönetim birimi (Memory Management Unit)
MSPS	: Saniyede Milyon adet örnek(Mega sample per second)
PL	: Programlanabilir Mantık (Programmable Logic)
POI	: Kesişme olasılığı (Probability of Intercept)
PS	: Alt Sistem İşleme (Processing Subsystem)
RAM	: Rasgele Erişim Belleği (Random Access Memory)
RBW	: Çözünürlük Bant Genişliği (Resolution Bandwidth)
RF	: Radio Frequency
RISC	: Azaltılmış Komut Seti Bilgisayarı (Reduced Instruction Set Computer)
ROM	: Sadece Okunabilir Bellek (Read Only Memory)
SDK	: Yazılım geliştirme Aracı (Software Development Kit)
SoC	: Çip Üzerinde Sistem (System On Chip)
SPI	: Seri Çevre Arayüzü (Serial Peripheral Interface)
VBW	: Video bant Genişliği (Video Bandwidth)
VHDL	: Çok Yüksek Hızlı Entegre Devre Donanım Tanımlama Dili (Very High Speed Integrated Circuit Hardware Description Language)

ÖZET

[YÜKSEK LİSANS TEZİ]

[GENİŞ BANTLI RF SPEKTRUM ANALİZİ İÇİN FFT TABANLI GERÇEK ZAMANLI SPEKTRUM ANALİZÖRÜ TASARIMI]

[Mehmet ESKİN]

İstanbul Üniversitesi-Cerrahpaşa

Lisansüstü Eğitim Enstitüsü

[Elektrik-Elektronik Mühendisliği Anabilim Dalı]

[Danışman : Prof. Dr. Fırat KAÇAR]

[Günümüzde kablosuz haberleşme çok yoğun bir şekilde kullanılmaktadır. Önümüzdeki yıllarda 5G ve Nesnelerin İnterneti gibi teknolojilerin yaygınlaşmasıyla kablosuz haberleşme yoğunluğu daha da artacaktır. Artan kablosuz haberleşme trafiği haberleşme sistemlerinde girişim gibi birçok ciddi soruna yol açmaktadır. Haberleşmenin sağlıklı bir şekilde sürdürülebilmesi için kablosuz haberleşme trafiğinin izlenmesi ve yönetilmesi gerekmektedir. Spektrum analizör cihazları yarım asırdan fazla süredir RF sinyallerinin izlenmesinde ve analizinden en önemli araçlardan bir tanesi olmuştur. Ancak spektrum analizör cihazları yüksek fiyatlarından dolayı sadece bir ölçüm cihazı olarak kullanılabilmektedir. Haberleşme sistemlerine yardımcı cihaz olarak kullanılamazlar. Çalışma kapsamında yapılan önermeler RF katmanı, sayısal sinyal işleme katmanı ve yazılım katmanı olarak üç kısımdan oluşmaktadır. RF katman, yüksek frekanslı RF sinyallerin frekanslarını düşürerek örnekleyen tümleşik bir yapıdan oluşmaktadır. Sayısal sinyal işleme katmanı ise FPGA üzerinde sinyallerin sayısal olarak işlenmesini sağlayan donanımsal blok tasarımlarından ve blokların yönetimini sağlayan mikroişlemciden oluşmaktadır. Çalışma kapsamında gömülü Linux işletim sistemi tercih edilerek sistem için özel olarak baştan sona bir gömülü Linux işletim sistemi derlenmiştir. Verilerin dışa aktarılmasını sağlayacak ve cihazın kontrolünü sağlayacak servisler yazılarak Linux işletim sistemi üzerinde koşturulmuştur. İşlenen sinyal verilerini grafiksel gösterimlere dönüştürerek gelişmiş analiz imkânı sunan bilgisayar tabanlı bir yazılım önerilmiştir. Çalışma sonucunda ortaya çıkan sistem amaçlanan hedefleri karşılayabilecek yapıdadır.]

Ocak 2020, 77 sayfa.

Anahtar kelimeler: Spektrum analizör, FPGA, Sayısal sinyal işleme, Hızlı Fourirer dönüşümü Gömülü Linux

SUMMARY

M.Sc. THESIS

DESIGN OF FFT BASED REAL TIME SPECTRUM ANALYZER SYSTEM FOR ANALYSIS OF WIDE BAND RF SPECTRUM

Mehmet ESKİN

Istanbul University-Cerrahpasa

Institute of Graduate Studies

Department of Electrical and Electronic Engineering

Supervisor : Prof. Dr. Fırat KAÇAR

Today, wireless communication is used extensively. With the spread of technologies such as 5G and Internet of Things in the coming years, the density of wireless communication will increase further. Increased wireless communication traffic causes many serious problems such as interference in communication systems. Wireless communication traffic must be monitored and managed in order to maintain a healthy communication. Spectrum analyzer devices have been one of the most important tools for monitoring and analyzing RF signals for more than half a century. However, spectrum analyzer devices can only be used as a measuring device because of their high prices. They cannot be used as auxiliary devices for communication systems The propositions made within the scope of the study consist of three parts as RF layer, digital signal processing layer and software layer. The RF layer consists of a structure that reduces the frequencies of high frequency RF signals.The digital signal processing layer consists of hardware block designs that allow the digital processing of signals on the FPGA and the microprocessor that manages the blocks. Embedded Linux operating system was preferred and an embedded Linux operating system was compiled specifically for the system. The services that will maintain the data transfer and control of the device are written and run on the Linux operating system. A computer-based software has been proposed to convert processed signal data into graphical representations, providing advanced analysis capabilities. As a result of the study, the system is able to meet the objectives.

January 2020, 77 pages.

Keywords: Spectrum Analyzer, FPGA, digital signal processing, fast Fourier transform, Embedded Linux

1. GİRİŞ

Günümüzde kablosuz haberleşmenin yaygınlaşması ve gittikçe daha da artması ile yoğun bir RF sinyal trafiği oluşmaktadır. Bu trafik yoğunluğu da beraberinde sorunları getirmektedir. Kablosuz haberleşmenin sağlıklı bir şekilde sürdürülebilmesi için RF trafiğinin izlenmesi ve analiz edilmesi gerekmektedir. Spektrum analizör cihazları ise RF sinyallerin izlenmesinde ve analizinde kullanılan en önemli ölçüm araçlarıdır.

Spektrum analizör teknolojisi 1950lere dayanmaktadır. Ancak günümüzün gelişen teknolojisi ile spektrum analizörlerde birçok gelişmiş özelliğe kavuşmuştur. Bu tez çalışmasında spektrum analizör teknolojisinin geldiği en güncel nokta olan FFT tabanlı gerçek zamanlı spektrum analizör önerilmesi yapılacaktır. Hedefler doğrultusunda kaynak ve literatür araştırması yapılmıştır. Yapılan araştırmalar sonucunda sistem gereksinimleri belirlenerek uygulama aşamasına geçilmiştir.

RF alıcılar tasarlaması ve üretilmesi zor olan yapılardır. O yüzden çalışma kapsamında RF sinyal örnekleri benzetim ortamında üretilecektir.

Geliştirme kartı olarak Avnet firması tarafından üretilen ve Xilinx firmasının ZYNQ 7000 mimarili SoC yongasını barındıran Zedboard kullanılması planlanmaktadır. ZYNQ 7000 mimarisi Xilinx tarafından geliştirilen, FPGA ve mikroişlemci yapılarını tek yongada birleştiren bir mimaridir. Geliştirme kartı üzerinde gömülü Linux işletim sistemi tercih edilmiştir. Gömülü Linux işletim sisteminin geliştirilmesinde ve derlenmesinde ise Xilinx firması tarafından sunulan Petalinux SDK ortamı tercih edilmiştir. FPGA üzerinde geliştirilecek sinyal işleme IP bloklarının mikroişlemci tarafından yönetilebilmesi amacıyla IP bloklarının kernel sürücüleri yazılarak Linux işletim sistemine dâhil edilmiştir.

FPGA üzerinde sinyal işleme bloklarının geliştirmesinde benzetim ortamından faydalanılarak doğrulama yapılması planlanmaktadır. İlk aşamada uygulanacak sinyal işleme algoritmalarının Matlab ortamında benzetimleri oluşturularak çıktılar Matlab kütüphanesinin çıktılarıyla karşılaştırılacak. Sonraki aşamada ise tasarlanan sinyal işleme IP blok tasarımları Vivado üzerinde benzetimleri koşturularak çıktılar tekrardan doğrulanacaktır. Son aşamada ise geliştirme kartı üzerinde tüm sistemle birlikte koşturulan IP blokların çıktıları daha önceki

çıktılar ile kıyaslanarak doğrulaması gerçekleştirilecektir. Doğrulama ve test aşamalarında farklı senaryolar uygulanarak çıktılar karşılaştırılacaktır.

Çalışma kapsamında, Linux işletim sistemi sinyal işleme IP bloklarının yönetme ve üzerinde koşan USB sunucusu ile verileri bilgisayar ortamına aktaracaktır. Bilgisayar ortamındaki yazılım ise alınan verileri görselleştirerek kullanıcıya aktaracaktır. Sinyal örnekleme hızları çok büyük boyutlara ulaşabilmektedir. Çalışmada hedeflenen 25 MHz bant genişliği için sinyal örnekleme hızı 25 MSPS olmalıdır. Her örneğin 2 byte olduğu hesaba katılırsa, üretilen veri miktarı 25 000 000*16 bit(2 byte) = 400 Mbit olmaktadır. USB 2.0 veri aktarım standardı 480 Mbit/s veri aktarım hızı çalışma için yeterlidir.

Linux işletim sisteminin bilgisayar yazılımı üzerinden gelen komutları gerçekleştirerek sinyal işleme IP bloklarının yönetimini sağlaması amacıyla IP bloklarının sürücü kütüphanesi ve sistem algoritması geliştirilmesi gereklidir. Böylece sistemin spektrum analiz yazılımı tarafından gelen komutları sistem algoritmasına uygun olarak gerçekleştirmesi sağlanacaktır.

Tez kapsamında, cihazın kontrolünü ve kullanımını sağlayan USB üzerinden haberleşen spektrum analiz yazılımı geliştirilmesi planlanmaktadır. Spektrum analiz yazılımında spektrumların daha gelişmiş analizine olanak sağlayan grafiksel gösterim yöntemleri geliştirilmesi planlanmaktadır. Ayrıca spektrum analizör cihazının yapılandırılmasını sağlamak amacıyla yazılım ara yüzüne yönetim paneli eklenmesi gerekmektedir.

[ ]

2. GENEL KISIMLAR

2.1 ARM Mimarisi

Asıl adı Acorn RISC machine olan ARM mimarisi 1983 yılında Acorn Computer Ltd. firması tarafından geliştirme projesi olarak başladı. Geliştirme ekibinin başında Steve Furber ve Roger Wilson bulunmaktaydı. İleri seviye bir MOS Technology 6502'yi ortaya çıkarmışlardır [1].

1990 yılında firma adını Advanced RISC Machine Ltd. olarak değiştirmiş ve 1998 yılına kadar bu isimle devam etmiştir. Firma 1998 yılında günümüzdeki ismi olan ARM Ltd. ismini almıştır.[2]

ARM mimarisi RISC tabanlı bir işlemci mimarisine sahiptir. ARM mimarisi düşük güç tüketimli ve yüksek performanslıdır. Ayrıca x64-86 işlemcilerine göre daha hesaplıdırlar. Bu avantajlarından dolayı gömülü sistemlerde ve taşınabilir cihazlarda genelde ARM işlemciler tercih edilir.

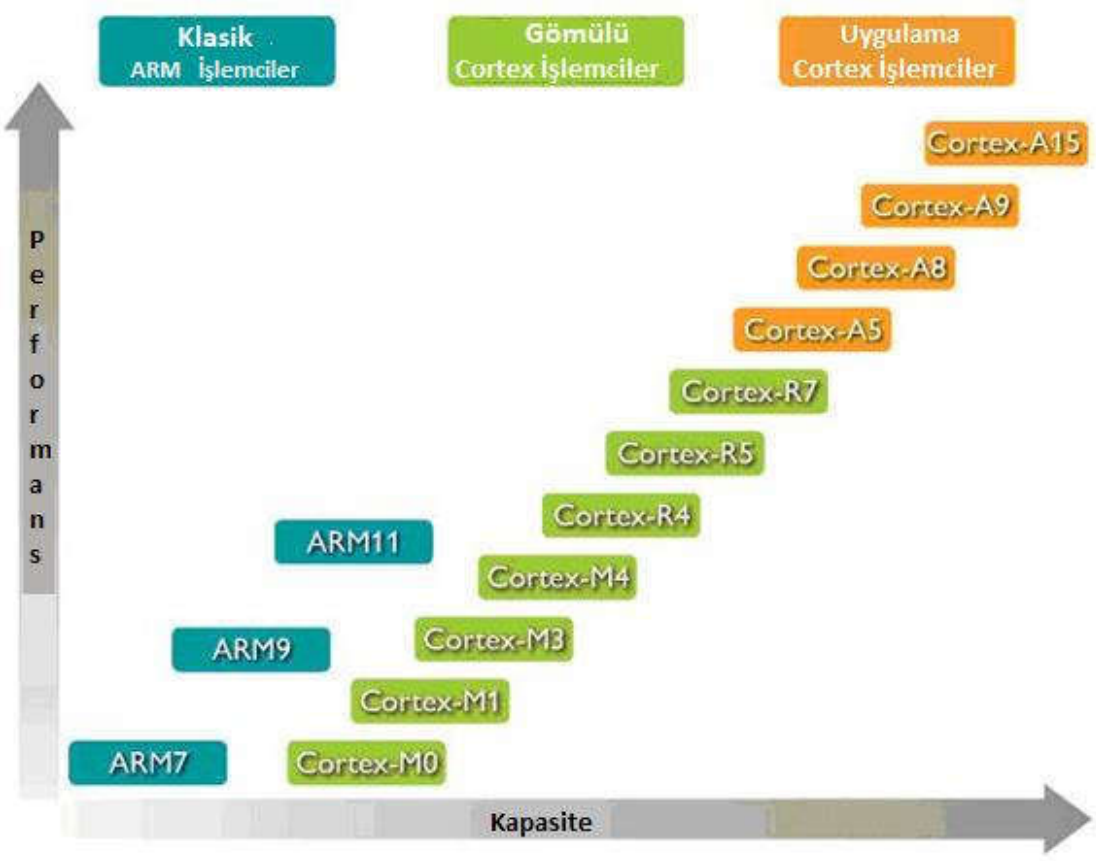
ARM mimarisinin 32 bit ve 64 bit olmak üzere iki tipi mevcuttur. ARM işlemciler diğer RISC tabanlı işlemcilere kıyasla daha performanslıdırlar. Düşük güç tüketimi ve hesaplı olmaları ARM işlemcilerin kullanımını artırmıştır. ARM firması kendi başına işlemci üretmez, tasarlar ve lisansı satar, CISC tabanlı işlemcilere göre özelleştirilebildiğinden aynı nesi işlemci farklı üreticilerden değiştirilmiş olarak çıkabilir, bağlı olarak da performans farklılıkları görülür. Bu yüzden işlemci nesil ve karakteristikleri incelenirken ARM referans tasarımı ele alınır[3].

ARM mimarisini gruplandıracak olursak;

Klasik ARM işlemci grubunda ARM7, ARM9 ve ARM11 yer alır. Bu mimari yapıları daha çok motor kontrolü gibi donanıma yakın işlemlerde kullanılır [2].

İkinci grup olarak gömülü işlemci grubuna bakarsak bunlar da Cortex M ve R serileri olacaktır. Bunlar matematiksel formüllere dayalı olarak kesin sonuca yönelik çalışması gereken, düşük güç tüketimi isteyen uygulamalarda kullanılır [2].

Son olarak da uygulama işlemcileri Cortex-A5, Cortex-A8, Cortex-A9, Cortex-A15 olarak sınıflandırılabilir. Bunlar da akıllı telefonlar, tablet bilgisayarlar gibi yüksek performans gerektiren, son kullanıcı uygulamalarına yönelik platformlarda kullanılır[2]. Şekil 2.1 'de Arm mikroişlemci ailesine ait işlemcilerin performans kapasite grafiği verilmiştir.



Şekil 2.1: Arm Mimari Ailesi [4].

2.2 Gömülü Linux Nedir?

Gömülü Linux, Linux işletim sisteminin cep telefonları, elde taşınabilir medya oynatıcılar ve diğer tüketici elektroniği cihazları gibi gömülü sistemlerde kullanılan adıdır [5].

Gömülü Linux sistemlerin disk üzerinde genel olarak kapladığı alan 2 MB ile sınırlıdır. Birçok cihazda kullanılması için farklı firmalar ve topluluklar Linux sistemleri geliştirmiş ve bu işletim sistemleri gömülü cihazlarda uzun yıllar boyunca kullanılmıştır [6].

Başlıca Gömülü Linux işletim sistemleri;

- VxWorks,
- QNX,
- LynxOS,
- Windows CE,
- Windows NT Embedded (Gömülü),
- ve Palm OS işletim sistemleridir.

Gömülü Linux işletim sistemlerinin diğer işletim sistemlerine oranla birçok büyük avantajı vardır. Bunlardan en önemlisiyse sistemin açık kaynak olarak kullanıcıya ulaşması ve kurulum başına lisans ücretinin talep edilmemesidir [6].

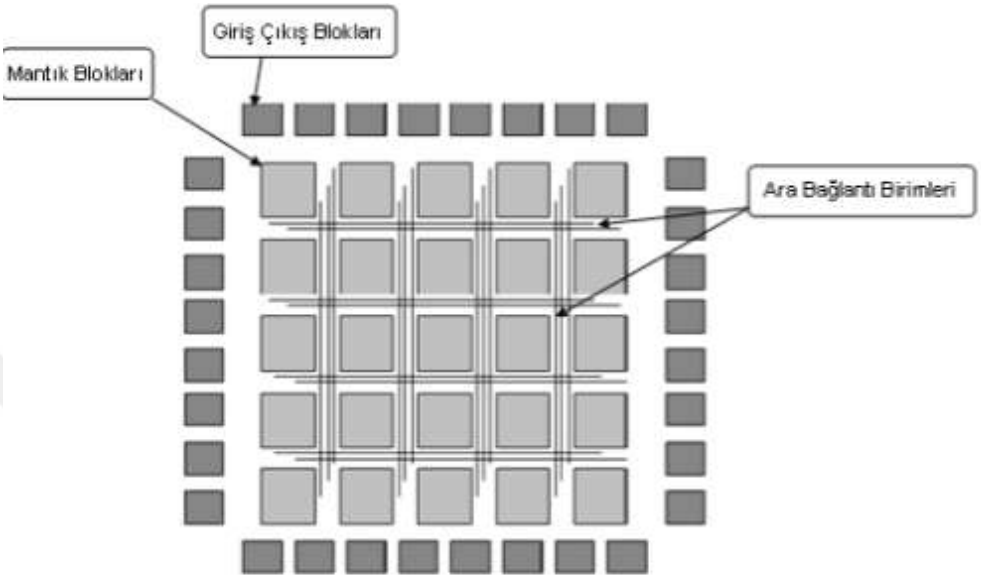
Gömülü Linux işletim sistemleri diğer sistemlere kıyasla daha kararlı bir yapıya sahiptir. Piyasa birçok şirket ve topluluk tarafından geliştirilen sistemler 10 yılı aşkın süredir kullanılmaktadır. Bu sistemler uzun yıllardır kullanılması ve geliştirilmesinin sonucu olarak kullanıcıların tüm istek ve beklentilerini karşılayacak olgunluğa erişmiş sistemlerdir.

2.3 FPGA Nedir?

FPGA (Field Programmable Gate Array - Alanda Programlanabilir Kapı Dizileri), programlanabilir mantık blokları ve bu bloklar arasındaki ara bağlantılardan oluşan ve geniş uygulama alanlarına sahip olan sayısal tümleşik devrelerdir. Tasarımcının ihtiyaç duyduğu mantık işlevlerini gerçekleştirme amacına yönelik olarak üretilmiştir. Dolayısıyla her bir mantık bloğunun işlevi kullanıcı tarafından düzenlenebilmektedir. FPGA ile temel mantık kapılarının ve yapısı daha karmaşık olan devre elemanlarının işlevselliği artırılmaktadır. Alanda programlanabilir ismi verilmesinin nedeni, mantık bloklarının ve ara bağlantıların imalat sürecinden sonra programlanabilmesidir [7].

FPGA'lar birçok firma tarafından üretilir ve birbirinden farklı yapıda bulunabilirler. Bu farklılıklara rağmen cihazlar arasında tasarım mimarisi açısından birçok benzerlik bulunmaktadır. FPGA; programlanabilir mantık blokları, bu blok dizisini çevreleyen giriş-çıkış blokları ve ara bağlantılar olmak üzere düzenlenebilir üç ana bölümden oluşmuştur. Programlanabilir mantık blokları, ara bağlantılar içerisine gömülü şekilde bulunur. Bunların

yapılandırılması ve aralarındaki iletişim ara bağlantılar sayesinde gerçekleşir. Giriş-çıkış blokları, ara bağlantılar ile bütünleşmiş devrenin paket bacakları arasındaki bağlantıyı kurar [8]. Şekil 2.2 'de FPGA iç yapısı gösterilmektedir.



Şekil 2.2: FPGA Yapısı [7].

FPGA'lerin en önemli özelliklerinden birisi de paralel işlem yapabilmesidir. Paralel işlem yeteneği sayesinde FPGA'ler aynı anda birden fazla işlemi yapabilmektedir. Paralel işlem yeteneği FPGA'leri gerçek zamanlı görüntü ve sinyal işleme gibi alanlarda çok kullanışlı yapmaktadır.

FPGA tasarımı temel olarak iki farklı yöntemle yapılabilmektedir. Birincisi donanım tanımla dili olan Verilog veya VHDL dilleri kullanılarak tasarım yapılabilir. İkincisi ise şematik tasarımla şema üzerine devre elemanlarının teker teker yerleştirilerek ara bağlantılarının şematik üzerinden yapılmasıdır. Ayrıca FPGA tasarım geliştirme ortamlarının gelişmesiyle C/C++ dilleri kullanılarak da tasarım yapılabilmektedir. Ancak derleyiciler C/C++ kodlarını donanım tanımlama dillerine (Verilog ya da VHDL) dönüştürerek donanım tasarımı yapılmasını sağlarlar.

2.4 SoC Nedir?

SoC (System On Chip – Çip Üzerinde Sistem), temel olarak bir bilgisayarı çalıştırılabilmesi için gereken tüm bileşenlerin üzerinde olduğu bütünleşmiş bir mikroçiptir.

SoC mimarisi bütünleşik yapısı sayesinde bileşenler arasında sıkı bir birleşime sahiptir. Böylece elemanlar arasındaki kablolama ihtiyacını ortadan kaldırarak daha ufak boyutlarda devre tasarımı yapmayı ve tasarım maliyetinden önemli ölçüde tasarruf etmeyi sağlar. SoC mimarisin sağladığı en önemli avantajlardan bir tanesi de düşük güç tüketimi sağlamasıdır.

SoC sistemlerin mikro denetleyicilerden en büyük farkı işlemci hızı ve yüksek belleğidir. Ayrıca SoC sistemler harici depolama, bellek ve çok çeşitli başka elemanlarla aynı kart üzerinde onları yöneterek çalışabilirken mikro denetleyicilerde bu gibi çok parçalı bir yapı yoktur. Daha doğrusu bir gömülü sistem eğer bir mikro denetleyici ile kurulduysa ROM, RAM gibi bileşenler mutlaka mikrodenetleyicinin içerisinde bulunurken yongada sistemde böyle bir zorunluluk yoktur. Ayrıca SoC sistemlerin gücü son kullanıcıların kullandığı Windows, Macintosh, iOS, Android vb. işletim sistemlerini çalıştırmaya yeterken mikrodenetleyiciler çok daha kısıtlı alanlarda ve çok kısıtlı ROM ve RAM'ler ile çalışırlar. Nihayetinde mikrodenetleyiciler örneğin bir çamaşır makinesini kontrol ederken SoC sistemler akıllı telefon, tablet ve hatta bilgisayarlarda kullanılabilirler [9].

SoC sistemler kullanım amaçlarına göre birçok farklı bileşene sahip olabilirler. Temel SoC sistem şunları içerebilir:

- Merkezi İşlemci (CPU)
- Grafik İşlemcisi (GPU)
- Bellek (RAM, ROM)
- Alanda Programlanabilir Kapı Dizileri (FPGA)
- Sayısal Sinyal İşlemcisi (DSP)
- Analog Kontrolörler
- Giriş-Çıkış birimlerinin kontrolcüleri (Ethernet, SPI, I2C, Seri Haberleşme, USB vb.)
- Kablosuz Haberleşme Kontrolcüleri (Bluetooth, Wi-Fi, Hücresel Şebeke vb.)

Bu sistem bileşenleri SoC sistemlerde bütünleşik olarak bulunabildiği gibi kullanım alanına göre kart üzerinde de bulunabilmektedir.

2.5 Spektrum Analizör ve Spektrum Analiz Teknolojisi

2.5.1 Spektrum Analizör Nedir?

Spektrum analizörler sinyallerin gücünü frekansa göre ölçmeye yarayan aletlerdir. Spektrum analizör zaman eksenindeki sinyalleri alarak frekans eksenine taşır. Spektrum analizör cihaz kapasitesine göre sinyal spektrumlarını belirli bir frekans aralığında ve bant genişliğinde gösterebilirler. Ayrıca spektrum analizör elektriksel, akustik veya optik dalga biçimlerini incelemek için en öncelikli araçtır. Şekil 2.3 'te masaüstü tipinde bir spektrum analizör cihazı gösterilmektedir.



Şekil 2.3: Masaüstü Spektrum Analizör [10].

Sinyallerin spektral analizi sayesinde zaman ekseninde gözlemlenmesi mümkün olmayan frekans, güç, bozulma, harmonikler, bant genişliği ve diğer spektral bileşenler rahatça gözlemlenebilir. Bu parametreler sinyallerin karakterize edilmesinde çok faydalıdırlar.

Elektriksel, akustik ve optik dalga spektrum analizörler temelde aynı prensiplere sahiptir. Ancak biz çalışmamızda RF sinyal spektrum analizörünü temel alacağız.

2.5.2 Spektrum Analizör Teknolojisi

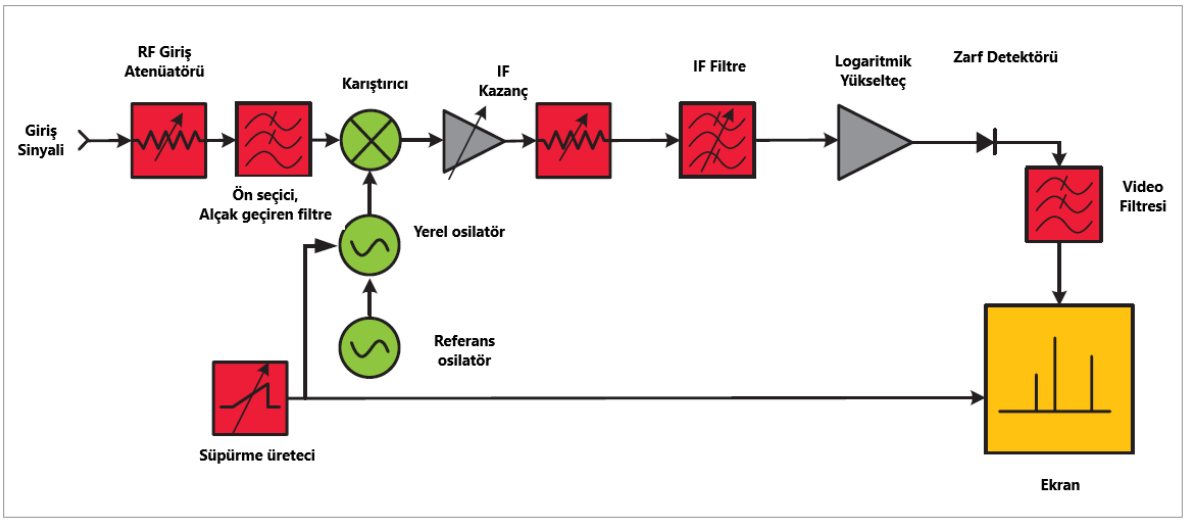
Sinyal spektrumlarını elde etmek ve göstermek için iki temel yaklaşım bulunmaktadır. Bunlar geleneksel tarama (swept) ve FFT (Fast Fourrier Transform) yaklaşımlarıdır.

Tarama analizörleri süper heterodin (superheterodyne) mimarisini kullanır. Süper heterodin mimarisi ilgili frekans aralığını uçtan uca tarayarak tüm frekans bileşenlerini gösterir. FFT analizörler ise giriş sinyallerinin verilerini bloklar şeklinde alarak hesaplar ve her bloğun

FFT'sini ekranda gösterir. FFT analizörler dinamik aralık ve frekans konusunda ADC (Analog Dijital Çevirici) teknolojisiyle sınırlıdırlar. Ancak swept analizörlerin aksine sadece sinyallerin gücünü değil aynı zamanda faz bilgisini de elde edebilirler. Spektrum analizörlerdeki en son gelişmeyle birlikte zaman ve frekans ekseninde veri kaybı olmadan analiz yapabilen gerçek zamanlı spektrum analizör yaklaşımı ortaya çıkmıştır. Gerçek zamanlı spektrum analizör yaklaşımı, spektral bilgilere göre tetikleme ve daha etkili grafiksel gösterim yöntemleri gibi birçok yeni imkânlar sunmuştur [11].

2.5.2.1 Swept-Tuned (Süpürülmüş) Spektrum Analizörleri

Süpürülmüş spektrum analizörleri yaklaşık 50 yıl önce piyasaya sürülen ilk spektrum analizör çeşididir. Süpürülmüş spektrum analizörlerde tamamen analog yöntemler kullanılır. Bu analizörler gerekli bant genişliklerine sahip filtreler ile sinyalin ilgili kısımlarını daha alçak frekanslara sahip olan sabit bir frekansa dönüştürerek spektrumları ekranda gösterir. Klasik bir süpürülmüş spektrum analizörün mimarisi Şekil 2.4 'te gösterilmektedir.



Şekil 2.4: Süpürülmüş spektrum analizörü [12].

Süpürülmüş spektrum analizörler alınan RF sinyali sırasıyla şu aşamalar geçer:

- Sinyal karıştırıcıya girmeden önce karıştırma işlemi için en uygun seviyeye gelmesini sağlamak amacıyla zayıflatıcı tarafından zayıflatılır. Böylece aşırı yükleme, bozulmalar ve kazanç sıkışmaları sorunlarının önlenmesi sağlanır.
- Giriş filtresi ise ilgili bandın izole edilmesini sağlar. Analizörler alçak frekanslarda sabit bir alçak geçiren filtre ve yüksek frekanslarda ayarlanabilir bir bant geçiren filtre

uygularlar. Böylece ara frekansta (IF-Intermediate Frequency) bant dışı istenmeyen yansımaların oluşması engellenir.

- Karıştırıcı filtreden geçen sinyali frekansı gerilimle değişen yerel osilatörden (LO) gelen sinyal ile karıştırır. Yerel osilatörün gerilimi süpürme jeneratörü yardımıyla doğrusal olarak ayarlanır. Süpürme jeneratörünün yapısı eski analog radyolarda bulunan frekans ayarlama mekanizmasına benzer bir yapıdadır.
- Karıştırıcıdan sonraki eleman değişken kazanç amplifikatörüdür. Değişken kazanç amplifikatörü sinyalin ekrandaki dikey pozisyonunu ayarlamak için kullanılır. Giriş zayıflatmasındaki herhangi bir değişimde ara frekans (IF) kazanç amplifikatörünün kazan değeri değişerek bu değişimi dengeler. Böylece karıştırıcının girişindeki sinyal seviyesini etkilemeden dengelenmiş olur.
- Ara frekans kazanç amplifikatöründen sonra artık frekans değeri daha düşük olan ara frekans kısmını geçmiş oluruz. Bu kısımda kullanıcın ekran görmek istediği ara frekans aralığını seçmeye yarayan çözünürlük bant genişliği (Resolution bandwidth-RBW) filtresi bulunur.
- Sinyalleri ekranda göstermek için en yaygın yöntem logaritmik gösterim yöntemidir. Bunun için sinyal logaritmik kazanç amplifikatöründen geçirilir.
- Sinyali ekranda gösterebilecek formata döndürmek için zarf detektörü kullanılır. Zarf detektörü basitçe diyotlar, direnç yükleri ve alçak geçiren filtreden oluşur. Bu yöntemle sinyalin faz bilgisi kaybedilir. Bu işlem, dijital olarak I ve Q elemanlarının karelerinin toplamının karekökünün alınması işlemiyle denktir.
- Son olarak analizörün iç gürültüsünü filtreleme için videobant genişliği filtresi kullanılır. Filtre sinyalin değerini yumuşatarak ya da ortalamasını alarak iç gürültüyü engellemeye çalışır.

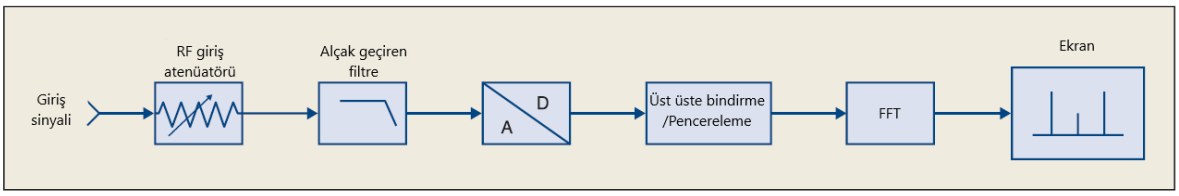
2.5.2.2 FFT Tabanlı Spektrum Analizörler

Bu yaklaşımı anlayabilmek için öncelikle DFT'yi (Discrete Fourrier Tranformu) anlamamız gerekiyor. FFT, DFT'yi temel alan ve ondan daha hızlı olan Fourrier dönüşüm algoritmasıdır. DFT dönüşümü sayısal olarak örneklenmiş zaman alanındaki sinyalleri frekans alanına taşır. Giriş sinyalleri sabit zaman aralıklarında örneklenir. Bu aralığa örnekleme periyodu (T) olarak adlandırılır. Örnekleme periyoduna karşılık gelen frekans değeri ise örnekleme frekansı (f_s) olarak ifade edilir. Her örnekleme karesinde (frame) sabit sayıda (N) örnek alınır ve FFT

hesaplaması yapılır. Sonuç olarak frekans alanında N tane karmaşık sayıdan oluşan bir sayı dizi oluşur.

Örnekleme frekansı (f_s) ve örnek sayısı (N) için bazı kısıtlamalar mevcuttur. İlk olarak örnekleme frekansı giriş sinyalinin ilgili kısmının bant genişliğini (BW) kapsayacak kadar yüksek olmalıdır. Olması gereken en az örnekleme frekansı Nyquist teoremi tarafından $f_s > 2BW$ olarak ifade edilmiştir. Bir diğer kısıtlama ise örnekleme sayısıdır. Örnekleme sayısı (N) istenen frekans çözünürlüğü ve gerekli olan örnekleme frekansı tarafından kısıtlanmaktadır. Frekans çözünürlüğü f_s/N olarak ifade edilebilir. Frekans çözünürlüğü artırmak için bir yol örnekleme frekansını azaltmaktır. Ancak örnekleme frekansı giriş sinyalinin bant genişliği tarafından kısıtlanmaktadır. Bir diğer yol ise örnekleme sayısını artırmaktır. Örnekleme sayısı (N) diziye sıfırlar eklenerek artırılabilir.

FFT tabanlı analizörlere ait blok diyagram Şekil 2.5 'te gösterilmiştir. Sinyal aynı süpürülmüş spektrum analizörde olduğu gibi ilk önce değişken bir zayıflatıcıya girer. Böylece sinyal karıştırıcıya girmek için uygun koşullara getirilir. Sinyal karıştırıcıya girdikten sonra ara frekansa düşürülür ve filtrelenir. Böylece Nyquist teoremine uygun hale getirilmiş olur. Filtrelenen ara frekans sinyali analog dijital çevirici ile sayısal verilere dönüştürülür. Sayısal verilere sayısal sinyal işleme teknikleri uygulanarak frekans alanına taşınır. Giriş sinyalinin frekans elemanları dijital bir ekranda kullanıcıya gösterilir.



Şekil 2.5: Klasik FFT spektrum analizör blok diyagramı [11].

Çalışma yapısı itibariyle FFT tabanlı spektrum analizörlerin analiz yapabildiği bant genişliği analog dijital dönüştürücülerin kapasitesine bağlıdır. Bu yüzden yüksek frekanstaki sinyalleri analiz edebilmek için sinyaller önce daha düşük frekans değerine sahip ara frekansa (IF) düşürülür.

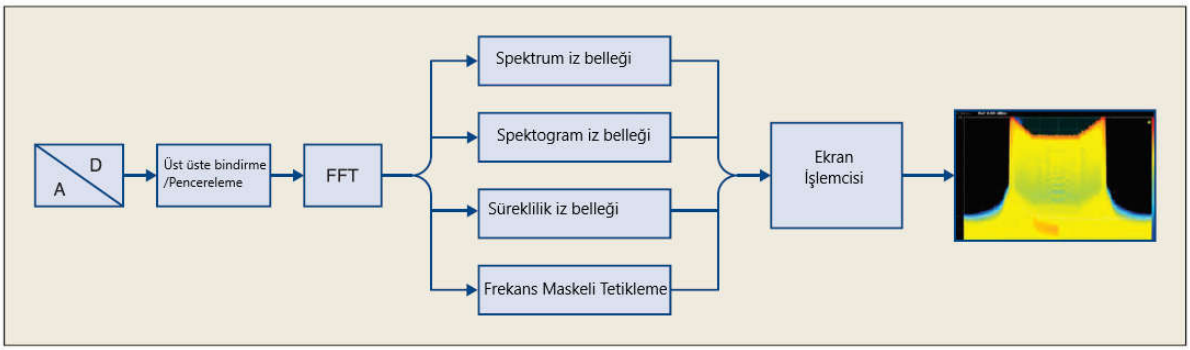
FFT spektrum analizörlerin en iyi özelliği ölçüm hızı gerektiren daha dar bant genişliğine sahip ve daha geniş aralıktaki sinyalleri analiz edebilmesidir. FFT analizörler birçok çözünürlük bant

genişliği (RBW) filtresine sahip olacak şekilde tasarlanabilmektedir. FFT analizörlerin bir diğer faydası ise bir örnekleme periyodunda tüm frekans alanını analiz edebilmesidir. Frekans aralığı taranırken karakteristikleri sürekli değişebilen dinamik sinyallerin görüntülenmesinde daha iyi sonuçlar verebilmektedirler.

2.5.2.3 Gerçek Zamanlı Spektrum Analizörler

Spektrum analizörlerde gerçek zamanlılık tüm sinyal örneklerinin kayıpsız olarak işlenebilmesini ifade etmektedir. Sayısal sinyal işleme teknolojisinin ilerlemesi sayesinde gerçek zamanlı sinyal işleme mümkün hale gelmiştir. Yüksek işlem kapasitesi ve paralel işlem yapabilme yeteneğine sahip FPGA ve DSP'ler sayesinde sinyaller aynı hem örneklenerek hem de FFT işlemi uygulanabilmektedir. Böylece ADC'den akan sinyal verileri sinyal işleme birimi tarafından eşzamanlı bir şekilde kayıpsız olarak alınarak FFT uygulanır. Şekil 2.6 'da gerçek zamanlı spektrum analizörün genel mimarisini gösterilmiştir.

Sinyallerin kayıpsız olarak işlenmesi birçok fayda sunmaktadır. Bu faydalardan bir tanesi sinyalleri sinyallerin yakalama olasılığı (POI-Probability of Intercept) kavramıdır. Sinyallerin yakalama olasılığı kavramı bir analizörün anlık bir spektrumu %100 olarak yakalayabileceği süreyi ifade eder. Gerçek zamanlı bir spektrum analizör bu sürenin üzerindeki tüm anlık olayları kaçırmadan kullanıcıya gösterebilir.



Şekil 2.6: Gerçek zamanlı spektrum analizör blok diyagramı [11].

Gerçek zamanlı bant dışındaki sinyallerin bant içerisine bozucu etkisi yapmasını önlemek amacıyla FFT dönüşümünden önce her bir FFT karesine pencereleme (windowing) olarak adlandırılan filtreleme yöntemi uygulanır. Pencereleme yönteminin olumsuz yanı ise birden fazla FFT karesi boyunca devam eden sinyallerin düzgün gösterilememesidir. Bunu önlemek üst üste bindirme (overlap) yöntemi uygulanır. Üst üste bindirme alınan sinyal örneklerinin bir

kısmının yeni bir FFT karesi hesaplaması için tekrardan kullanılmasıdır. Böylece bir FFT karesinden daha uzun süren olaylar kayıpsız olarak gösterilebilir.

Gerçek zamanlı spektrum analizörler, spektrumların ekran üzerinde yeni ve daha etkili gösterim yöntemleri, değişken sinyaller için tetikleme mekanizması ve dinamik sinyallerin daha etkili analizi gibi birçok yeni imkânlar sunmaktadır.[ ]

3. MALZEME VE YÖNTEM

3.1 Sistem Genel Mimarisi

Geniş bantlı RF spektrumlarının analizini sağlayacak gerçek zamanlı spektrum analizör gerçeklemesi RF alıcı, analog dijital çevirici, FPGA ve spektrum analiz yazılımı olmak üzere 4 kısımdan oluşmaktadır.

RF kısmı temel olarak yüksek frekanstaki sinyalleri alarak daha düşük frekanslara sahip ara frekans (IF) bandına dönüştürecektir. Böylece sinyal frekans değeri ADC'nin örnekleme yapabileceği hızlara indirgenmiş olacaktır. ADC yüksek örnekleme hızlarına sahip olacağı için çok hızlı veri aktarımı gerekmektedir. Hızlı veri transferinin sağlanması amacıyla RF katman ile sayısal sinyal işleme katmanı arasında, özelleştirilebilir giriş-çıkış pin yapısına ve 10Gbit/s veri transfer hızına sahip olan FMC (FPGA Mezzanine Card) konektör yapısının kullanılması planlanmaktadır.

Geliştirilecek sistemin gerçek zamanlı olabilmesi için giriş sinyalinin tamamının kayıpsız olarak işlenmesi ve belleğe alınması gerekir. Sinyalleri kayıpsız olarak işlemek ve bellek taşmalarını önlemek amacıyla örnekleme ve FFT hesaplaması işlemleri FPGA üzerinde paralel olarak tasarlanacaktır. Böylece FFT hesaplaması gerçekleştirilirken aynı anda sistem paralel olarak veri almaya devam edebilecektir.

Bir sinyale FFT uygulandığında spektral sızıntı ve işlem kayıpları oluşmaktadır. Spektral sızıntılar çözünürlüğü ve sonuçların doğruluğunu düşürücü bir etki yapar. Spektral sızıntıları önlemek için sinyale FFT uygulamadan önce pencereleme (windowing) tekniklerini uygulayacak yapı FPGA yine üzerinde tasarlanacaktır. Yapılmak istenen ölçüm şekline göre farklı karakteristiklere sahip Rectangular, Blackman-Harris, Hanning, Hamming, Flattop gibi pencereleme (windowing) tekniklerinden herhangi bir tanesi seçilerek uygulanabilecektir.

Sinyal spektrumunun kullanıcıya tarafından analizini yapılmasını sağlayacak gösterim yöntemlerinin ve cihaz ayar ara yüzünün olacağı spektrum analiz yazılımı bilgisayar ortamında çalışacak. Spektrum analizör cihazı USB üzerinden haberleşerek işlenmiş verileri bu yazılıma aktaracaktır. Veriler spektrum analiz yazılımı ile gerçek zamanlı gösterim yöntemlerini kullanarak kullanıcıya gösterecektir.

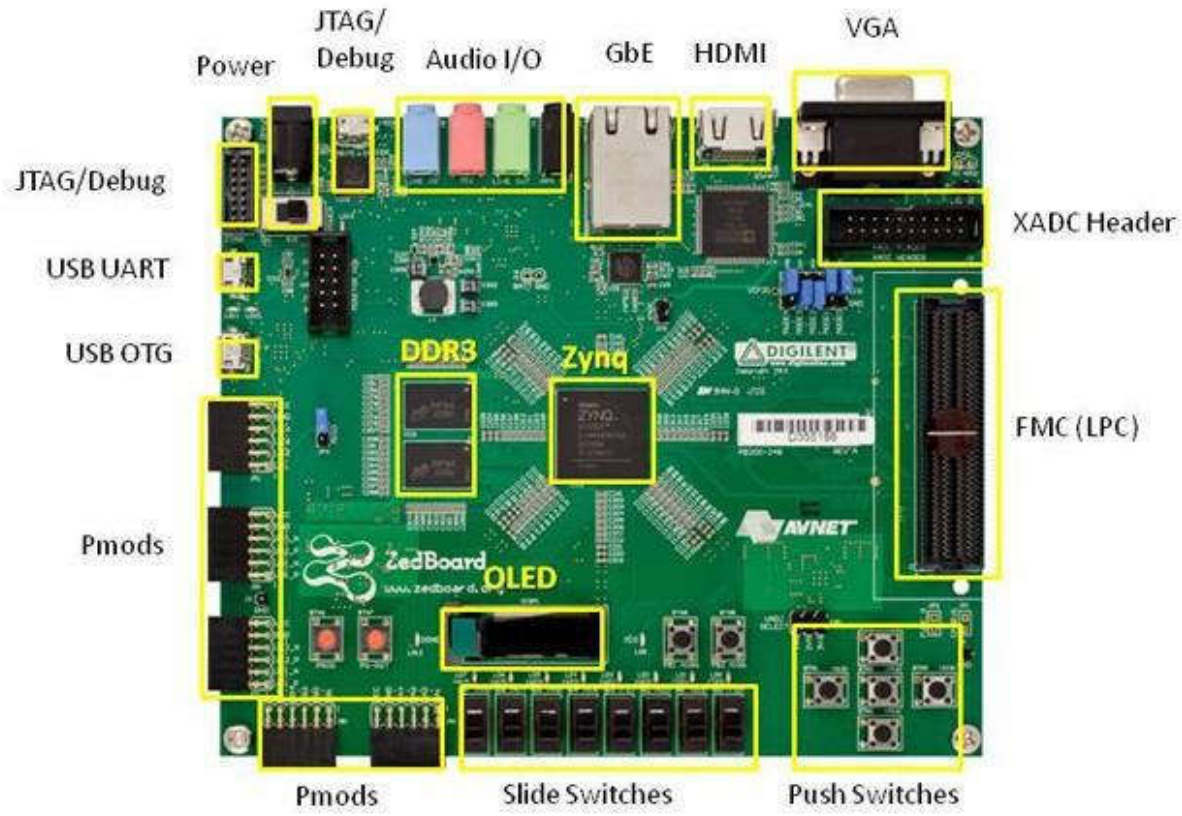
3.1.1 RF Alıcı Modülü

Çalışmada RF alıcı ve analog dijital dönüştürücü olmadığı için geniş bantlı sinyal örnekleri Matlab yardımıyla üretilerek benzetim ortamında ve IP blok testlerinde girdi olarak kullanılacaktır.

3.1.2 FPGA ve Mikroişlemci Birimleri

Sayısal sinyal işleme katmanı FPGA ve mikroişlemci birimlerinden oluşmaktadır. Geliştirme kartı olarak XC7Z020 yongasına sahip Avnet firması tarafından geliştirilen Zedboard kullanılmıştır. Şekil 3.1 'de Zedboard geliştirme kartı üzerindeki birimler gösterilmektedir. Zedboard geliştirme kartının özellikleri aşağıda verilmiştir:

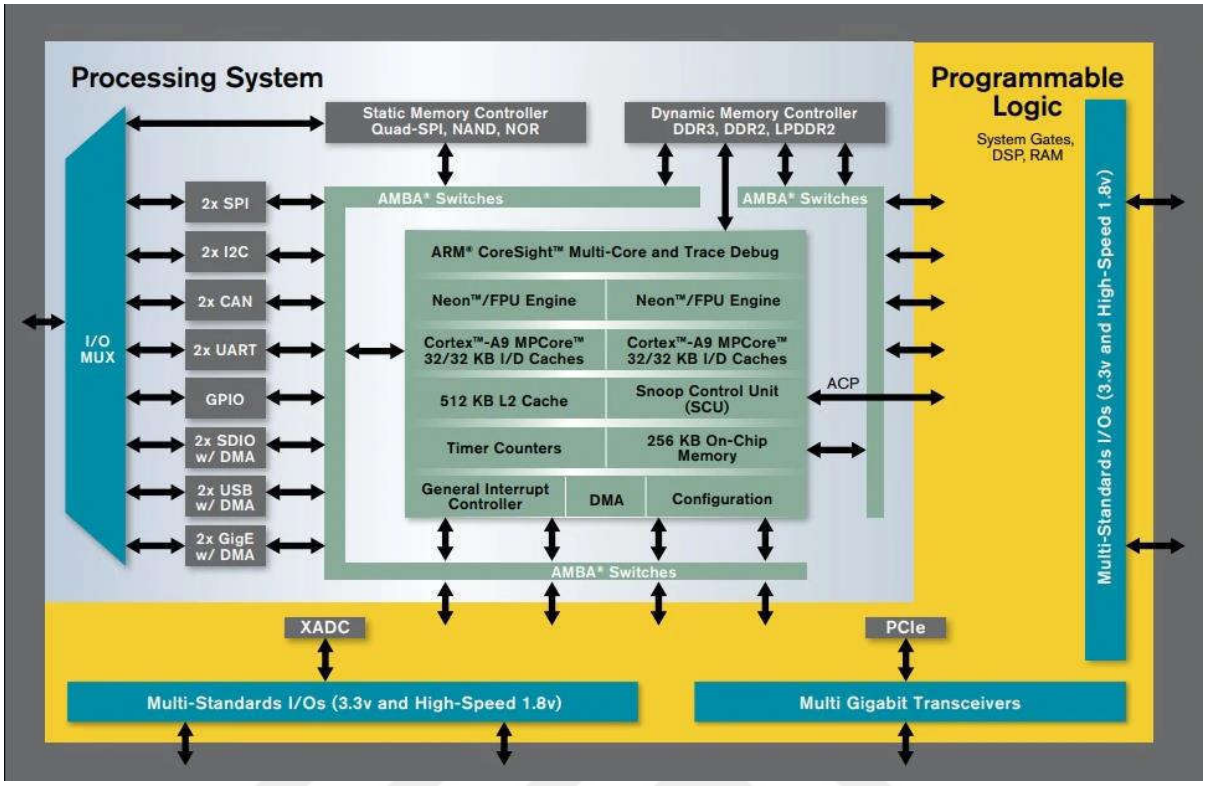
- Zynq™-7000 AP SoC XC7Z020-CLG484-1 mikro işlemci birimi
- 512 MB DDR3 bellek
- 256 Mbit Quad-SPI Flash bellek
- SD kart yuvası
- USB-JTAG programlama birimi
- 10/100/1000 Mbit Ethernet bağlantısı
- USB OTG 2.0 ve USB-UART ara yüzü
- FMC-LPC konektörü
- Pmod™ uyumlu pin başlıkları (2x6)
- PS (Processing system) için 33.33333 MHz saat kaynağı
- PL (Programmable Logic) için 100 MHz saat kaynağı
- HDMI çıkışı (1080p destekli 16-bit renk çözünürlüğü)
- VGA çıkışı (12-bit renk çözünürlüğü)
- 128x32 OLED ekran
- 8x LED, 7x buton, 8x DIP anahtar



Şekil 3.1: Zedboard geliştirme kartı üstten görünümü [13].

XC7Z020 yongası Xilinx firması tarafından geliştirilmiş ZYNQ 7000 serisi olarak adlandırılan ve FPGA ile mikro işlemci birimlerini tek yongada birleştiren SoC (System on Chip) bir yongadır. İçerisinde Xilinx Artix veya Kintex serisi FPGA'lere eşdeğer bir FPGA yapısı ve ARM Cortex-A9 MPCore mikro işlemci birimleri bulunmaktadır. ZYNQ 7000 mimarisi Şekil 3.2 'de gösterilmektedir. ZYNQ XC7Z020 yongasının özellikleri aşağıda verilmiştir:

- İki çekirdekli ARM Cortex-A9 mikro işlemci
- DDR3, DDR3L, DDR2, LPDDR2 harici hafıza desteği.
- 2x UART, 2x CAN, 2x I2C, 2x SPI, 4x 32 bit GPIO, 2x USB 2.0, 2x Ethernet, 2x IO DMA giriş çıkış ara yüz sürücüleri
- Artix-7 serisi programlanabilir mantık kapısı birimleri
- 85k mantıksal hücre, 53,2k LUT (Look-Up Table), 4.9Mb blok RAM, 220 DSP bloğu.

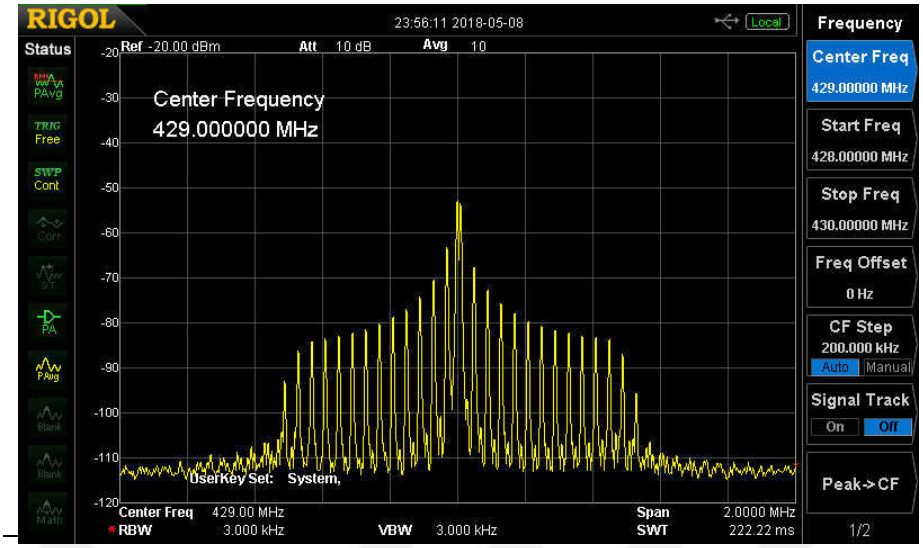


Şekil 3.2: ZYNQ7000 mimarisi [14].

3.1.3 Spektrum Analiz Yazılımı

Spektrum analiz yazılımı, sayısal olarak işlenen verileri kullanıcıya grafiksel gösterim yöntemleri kullanarak aktarılması, gözlemlemek istenen frekans bant aralığını seçilmesi, ölçüm ve cihaz parametrelerini ayarlanması gibi işlevleri yerine getiren ve bilgisayar üzerinde çalışan bir yazılımdır.

Spektrum gösterimi, dikey eksenin sinyal şiddetini yatay eksenin ise frekans değerine karşılık geldiği iki eksenli bir grafik üzerinde noktasal olarak gelen spektrum verilerinin ardışık olarak birleştirilmesiyle oluşan grafiksel gösterim yöntemidir. Spektrum gösteriminde gelen veriler değiştikçe grafik tekrardan çizdirilir. Böylece sinyallerin anlık olarak değişimi gözlenir. Şekil 3.3 'te spektrum grafiksel gösterimine ait bir görsel verilmiştir.



Şekil 3.3: Spektrum grafiksel gösterim yöntemi [15].

Mikroişlemci ile yazılım arasında USB bağlantısı üzerinden haberleşecektir. Mikroişlemci üzerindeki USB sunucu yazılımı FPGA tarafından işlenerek belleğe yazılan verileri spektrum analiz yazılımından gelen isteklere göre paketler halinde gönderecektir.

3.2 Gömülü Yazılım Geliştirme Ortamı

Spektrum analizörler yüksek sinyal örnekleme hızlarından dolayı yüksek boyutlarda veri üretmektedir. Mikroişlemcilerin hızları FPGA ler tarafından üretilen bu verilerle başa çıkabilecek kadar hızlı değildir. Bu yüzden üretilen veriler FPGA tarafından belleğe yazılır. Belleği mikroişlemci ve FPGA birimleri ortak olarak kullanırlar. Veri boyutunun yüksek olması ve sınırlı kapasitedeki belleğin daha etkili bir şekilde yönetilebilmesi amacıyla ARM A9 mimarisine sahip bir mikroişlemci ve Linux işletim sistemi tercih edilmiştir.

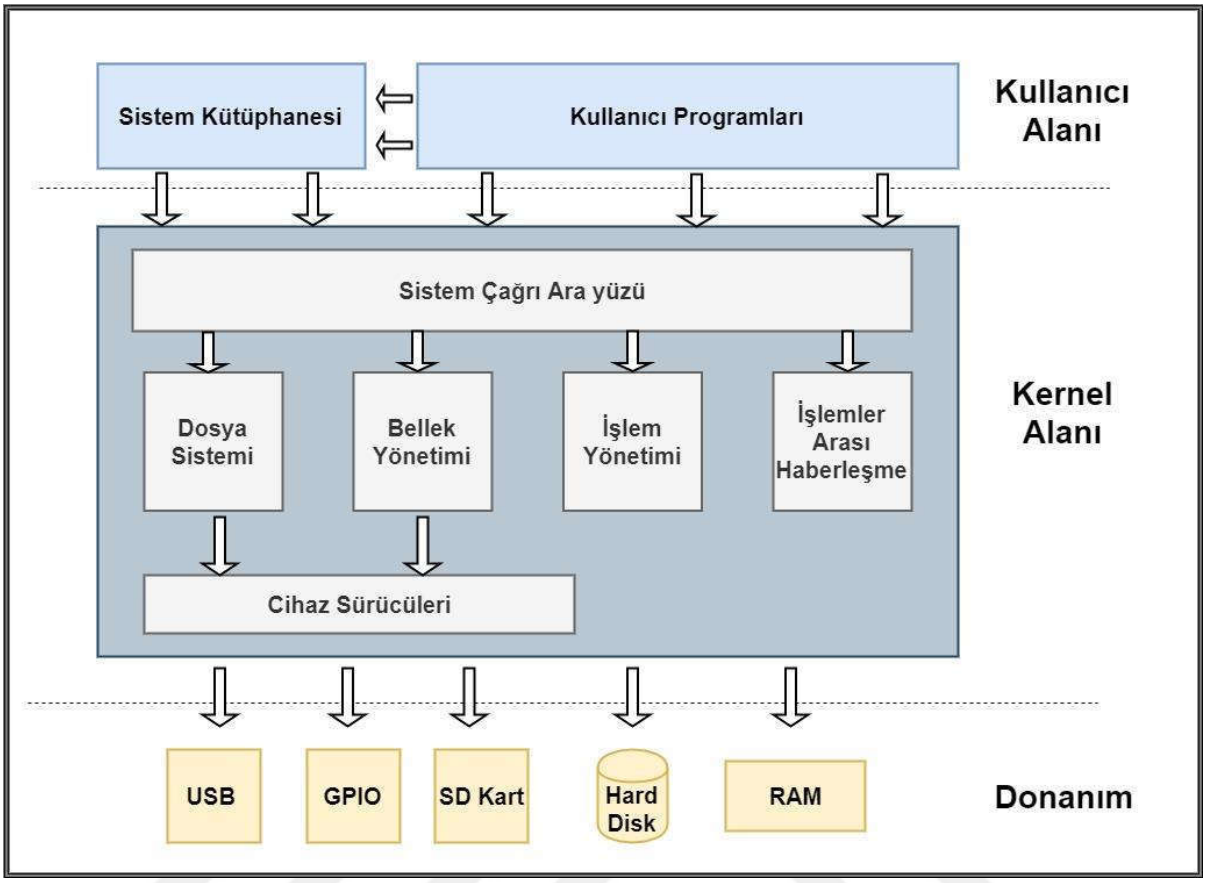
ARM Cortex-A serisi mikroişlemciler mikro kontrolörlere göre daha büyük boyutlarda belleğe ve daha yüksek saat hızlarına sahiptir. Böylece mikro kontrolörlere göre daha yüksek miktardaki verileri daha kısa sürede transfer edebilmektedir. Ayrıca mikro işlemci ve FGPA birimlerini tek bir yongada barındıran SOC yapısı sayesinden FPGA tarafından tasarlanan yapılar mikroişlemci birimi tarafından daha etkili bir şekilde yönetilebilmektedir.

Linux işletim sistemi sahip olduğu MMU (Memory Management Unit- Bellek yönetim birimi) ile mikro kontrolörlere göre çok daha sağlıklı ve verimli bir bellek yönetimi sunmaktadır. MMU fiziksel adreslere sahip bellek referanslarını sanal bellek adreslerine dönüştürerek bellek

koruması, bellek tahsisi, önbellek kontrolü, veri yolu çakışmaları ve yığın anahtarlama gibi bir görevi yerine getirir. MMU sahip olduğu bu fonksiyonlar sayesinde bellek yönetimini daha etkili ve verimli bir hale getirmektedir.

Örneklenen sinyallere ait büyük boyutlardaki veriler FPGA birimleri tarafından direk olarak belleğe yazılmaktadır. Belleğe yazma ve bellekten okuma işlemleri için hem mikroişlemci hem de FPGA kısımlarında DMA (Direct Memory Access) olarak adlandırılan ve yüksek veri yazma okuma hızlarına sahip bellek erişim birimleri kullanılır. DMA birimi merkezi işlem birimiyle paralel ve bağımsız çalışır. Böylece mikroişlemci üzerinde işlem yükü oluşması önlenir. FPGA tarafında kullanılan DMA birimi ile mikroişlemci tarafında kullanılan DMA birimleri birbirinden bağımsız yapılardır. FPGA tarafındaki DMA mantıksal kapılar kullanılarak tasarlanır ve görevi sinyal işleme blokları tarafından üretilen verileri mikroişlemci birimi tarafından belirtilen adreslere yazmaktır. Mikroişlemci tarafında ise mikroişlemci yongasının donanımsal olarak sahip olduğu DMA yapısı kullanılır. Mikroişlemci FPGA tarafından belleğe yazılan verileri DMA birimini kullanarak direk USB sürücüye aktarılmasını sağlar.

Gömülü Linux ortamı ADC yönetimi, FPGA tasarımlarının ve harici donanımların sürülmesi ile sunucu algoritmasını koşturarak spektrum analiz yazılımıyla haberleşmeyi sağlar. Şekil 3.4 ‘te gömülü Linux genel mimarisi gösterilmektedir. Şekilde en altta donanımsal birimler gösterilmektedir. Ortada ise donanımsal birimlerin kullanıcı uygulamaları tarafından sürülmesini sağlayan kernel alanı bulunmaktadır. Kernel alanı donanımlara ait sürücüleri barındırır ve kullanıcı donanımları sürmek istediğinde kernel bu sürücüleri kullanır. En üstte ise kullanıcı uygulama katmanı bulunmaktadır.



Şekil 3.4: Gömülü Linux Genel Mimarisi

3.2.1 Gömülü Linux Derleme

ZYNQ 7000 mimarisine sahip ARM Cortex-A9 yongasında gömülü Linux işletim sistemi geliştirmesi Xilinx firması tarafından sunulan Petalinux SDK ile yapılmıştır. Petalinux SDK Xilinx firması tarafından geliştirilen yongalar için gömülü Linux işletim sistemi derlemeye yarayan YOCTO tabanlı bir geliştirme ortamıdır. Ancak Petalinux SDK, YOCTO ya kıyasla daha basit bir geliştirme ortamı sunmaktadır. YOCTO da ki uzun ve karışık geliştirme aşamaları Petalinux SDK in sahip olduğu komut dizileri sayesinde otomatik olarak yapılabilmektedir. Petalinux sayesinde sistem gereksinimlerine göre özelleştirilmiş bir Linux işletim sistemi derlenmiştir. Böylece standart Linux işletim sistemlerindeki gereksiz yapılardan kurtularak daha hafif ve hızlı çalışan bir Linux işletim sistemi derlenmiştir. Petalinux SDK sadece Linux işletim sistemi üzerinde çalışmaktadır. Bu yüzden gömülü işletim sistemi geliştirmesi Ubuntu 16.04 LTS işletim sistemine sahip bir bilgisayar ile yapılmıştır.

ZYNQ 7000 mimarisinde işleciyle ile birlikte tümleşik olarak bulunan FPGA yongası üzerinde geliştirilen donanımsal birimlerin yönetimi mikroişlemci işlemci ile gerçekleştirilmesi

gerekmektedir. FPGA üzerinde tasarlanan donanımsal birimleri yönetilebilmesi için Linux işletim sistemi tarafından bu donanımların tanınması gerekmektedir. Petalinux SDK bu donanımsal tasarımların kolayca tanınabilmesi için gerekli ortamı sunmaktadır. FPGA tasarımlarına ait VHDL kodların derlenmesiyle oluşturulan HDF (Hardware Description File-Donanım Tanımla Dosyası) dosyasını Petalinux SDK sına girdi olarak vererek gömülü işletim sisteminin donanımı tanıması otomatik olarak yapılmaktadır. Ancak bu donanımların Linux işletim sistemi tarafından kullanılabilmesi için gerekli işlemler bununla yeterli değildir. Ayrıca Linux işletim sisteminde donanıma ait bir cihaz sürücüsünün geliştirilmesi gerekmektedir.

3.2.2 Gömülü Linux Derleme ve Cihaz Sürücüsü Ekleme

Linux işletim sistemiyle harici bir donanımı sürebilmek önce bu donanımı işletim sistemine tanıtmamız ve donanımı sürecek yazılımı kernele eklememiz gerekiyor. Spektrum analizörü çalışmasında sürmemiz gereken RF, ADC ve FPGA ile tasarlanan donanımların hepsini Linux işletim sistemine tanıtmamız gerekiyor. Petalinux geliştirme ortamı, işletim sistemini derlerken donanımları sisteme eklemek için standart bir yöntem sunmaktadır. Tablo 3.1 'de Petalinux SDK ile gömülü bir işletim sistemi derlenmesi için gereken tasarım adımları sırasıyla verilmiştir.

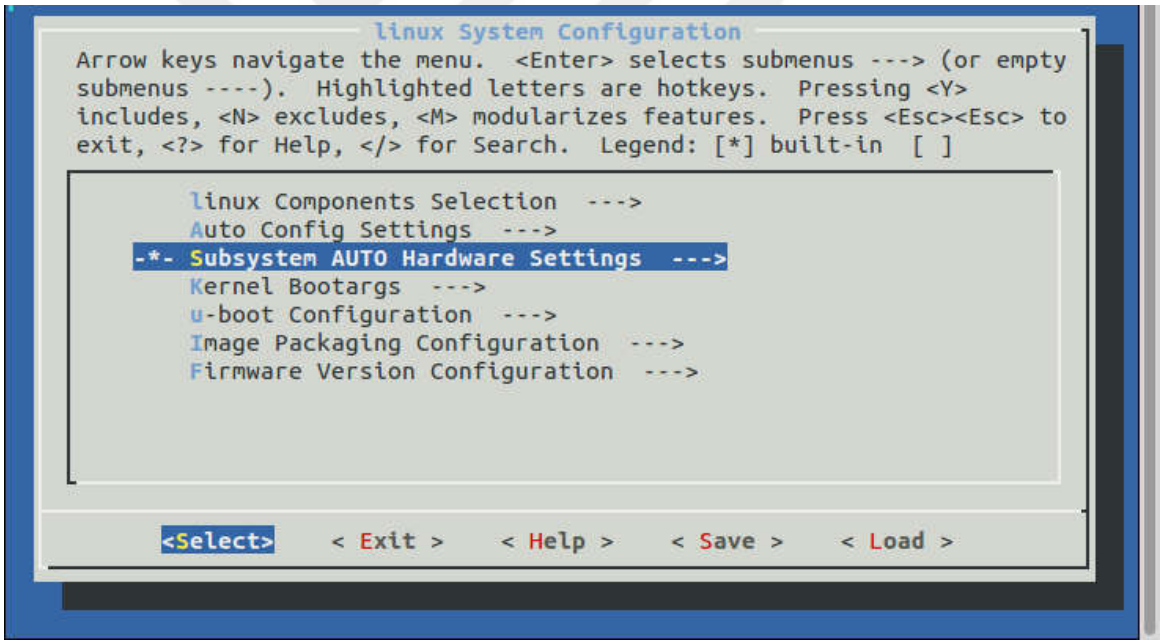
Tablo 3.1: Petalinux SDK ile gömülü işletim sistemi tasarım adımları [16].

Tasarım Adımları	Araç/ İş Akışı
Donanım Tasarlama	Vivado
Petalinux Projesi Oluşturma	petalinux-create -t project
Petalinux Projesini Başlatma	petalinux-config --get-hw-description
Sistem Seviyesinde Konfigürasyon	petalinux-config
Kullanıcı Elemanları Oluşturma	petalinux-create -t COMPONENT
Linux Kernel Konfigürasyonu	petalinux-config -c kernel
Kök Dosya Sistemi Konfigürasyonu	petalinux-config -c rootfs
Sistem Oluşturma	petalinux-build
Sistemi Paketleme	petalinux-package
Sistemi Test Etme	petalinux-boot

Tabloda ki adımları kısaca şu şekilde açıklayabiliriz;

- Donanım tasarlama adımında Vivado geliştirme aracıyla FPGA üzerinde HDL dilleri kullanılarak donanım tasarımı yapılarak sentezlenir. Sentezleme işlemi bittikten sonra HDF dosyası oluşturulur.

- Petalinux SDK kurulu olan bir bilgisayarda *petalinux-create –t project* komutuna geliştirme kartına ait .bsp(board support package) uzantılı dosyanın konumu verilerek petalinux projesi oluşturulur. Petalinux, BSP dosyası yardımıyla işletim sistemi derlenecek olan karta ait işlemci, bellek, donanımsal sürücüler vs. gibi bilgilere sahip olur.
- *Petalinux-config –get-hw-description* komutuna Vivado ile üretilen HDF dosyasının bulunduğu klasörün uzantısı verilir. Petalinux HDF dosyası yardımıyla tasarlanan FPGA donanımına ait bilgileri alarak donanımı tanır.
- *Petalinux-config* komutu ile işletim sistemine ait yapılandırma paneli açılır. Yapılandırma paneli üzerinde işletim sistemine ait kernel yükleneceği yer, boot olma seçenekleri, kök dosya sistemi tipi ve yükleneceği fiziksel konum gibi ayarlar yapılabilir. Şekil 3.5 'te Petalinux yapılandırma paneli verilmiştir.



Şekil 3.5: Petalinux-config yapılandırma paneli

- *Petalinux-create –t* COMPONENT komut ile yeni veya önceden oluşturulmuş uygulamaları, modülleri ve kütüphaneleri dosya sistemine ekleyebiliriz.
- *Petalinux-config –c kernel* komutu ile kernel yapılandırma paneli açılır. Bu panel üzerinden kernel sürücü ayarları, işletim sistemine dâhil edilecek kernel sürümü gibi ayarlar yapılır.

- *Petalinux-config –c rootfs* komutu ile kök dosya sistemi yapılandırma paneli açılır. Bu panel üzerinden servis paketleri ve uygulama kütüphaneleri ekleme, ağ, ara yüz multimedya gibi birçok kök dosya sistemine ait yapılandırma ayarları yapılabilir.
- *Petalinux-build* komutu ile işletim sistemi derlenerek çıktılar oluşur.
- *Petalinux-package* komutu ile işletim sistemi çıktıları kullanılarak mikroişlemci tarafından boot edilebilir ve yüklenebilir paket dosyalar oluşturulur. Paket dosyalar içerisinden kök dosya sistemi, kernel ve önyükleyici gibi işletim sistemi elemanları bulunmaktadır.
- Petalinux-boot komutu ise boot edilebilir paket dosyaları JTAG üzerinden geliştirme kartına veya emülatöre yüklenerek hedef sistem tarafından boot edilmesini sağlanır. Bu komutu kullanmak zorunlu değildir. Eğer geliştirme kartı SD kart, QSPI bellek veya ağ gibi ortamlardan boot edilebilir olarak ayarlandıysa paket dosyalar ilgili konuma taşınarak açılış durumunda boot edilebilir.

3.2.3 USB Sunucu ve Veri Aktarımı

Çalışma kapsamında tasarlanacak spektrum analizör cihazı, spektrum analiz yazılımı ile USB bağlantısı üzerinden haberleşecektir. Spektrum analiz yazılımı istemci olarak çalışır ve mikroişlemci üzerinde koşan sunucuya istekler göndererek cihaz ayarlarının yapılandırılmasını ve verilerin çekilmesini sağlar. Sunucu ise kullanıcıdan gelen isteklere göre cihaz yapılandırmasını yaparak verilerin kullanıcıya aktarılmasını sağlar.

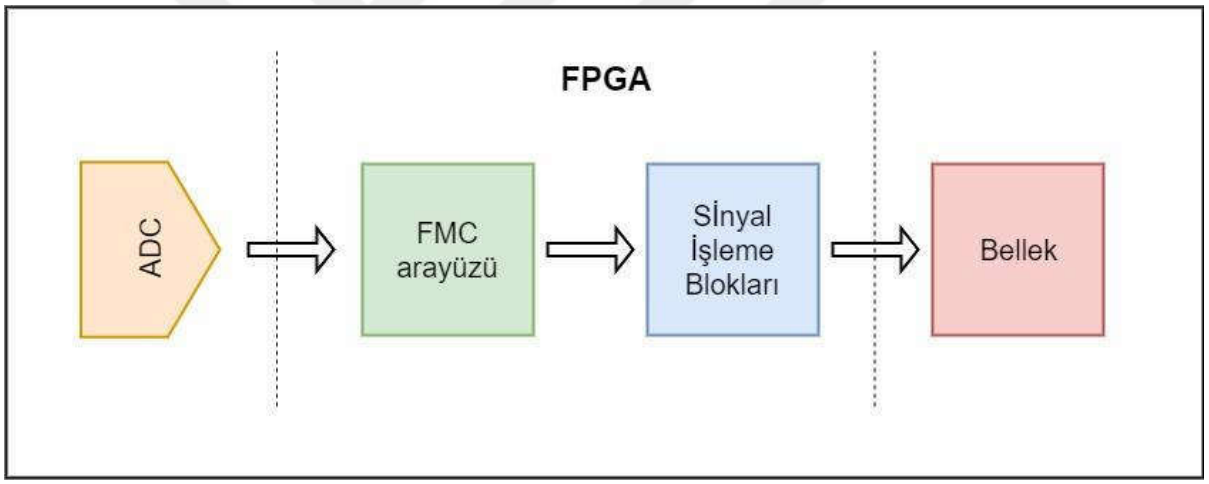
Kullanıcı ölçüm almak istediğinde öncelikle RF alıcı, ADC ve FPGA yapısının kullanıcıdan gelen parametrelere uygun bir şekilde yapılandırılması gerekir. Bu işlemin yapılması için spektrum analiz yazılımı veri talebinde bulunmadan önce ölçüm parametrelerini mikroişlemci üzerinde koşan sunucuya iletir. Sunucu ölçüm parametrelerini alarak donanımsal yapılandırmaları gerçekleştirir. Ayrıca FPGA tasarım bloklarına ait FFT boyutu, veri paketi büyüklüğü, pencereleme yöntemi ve üst üste bindirme (overlapping) gibi donanımlarının yapılandırılması sunucu tarafından gerçekleştirilerek veri üretmeye hazır hale getirilir.

FPGA, ADC ve RF alıcıların bazı donanımsal kısıtları bulunmaktadır. Bu yüzden kullanıcı tarafından gönderilen yapılandırma ayarlarına en yakın uygulanabilir ayarlar donanımsal olarak gerçekleştirilir. Yapılandırmalar yapıldıktan sonra spektrum analiz yazılımı sunucudan yapılandırma bilgilerini tekrar isteyerek donanım üzerinde gerçekleştirilen ayarları ara yüzünde

kullanıcıya gösterir. Donanımsal yapılandırma işlemi tamamlandıktan sonra spektrum analiz yazılımı sunucuya veri alma isteklerini göndererek FPGA tarafından belleğe yazılan işlenmiş spektrum veri paketlerini çeker ve grafiksel görüntüye dönüştürerek kullanıcıya gösterir.

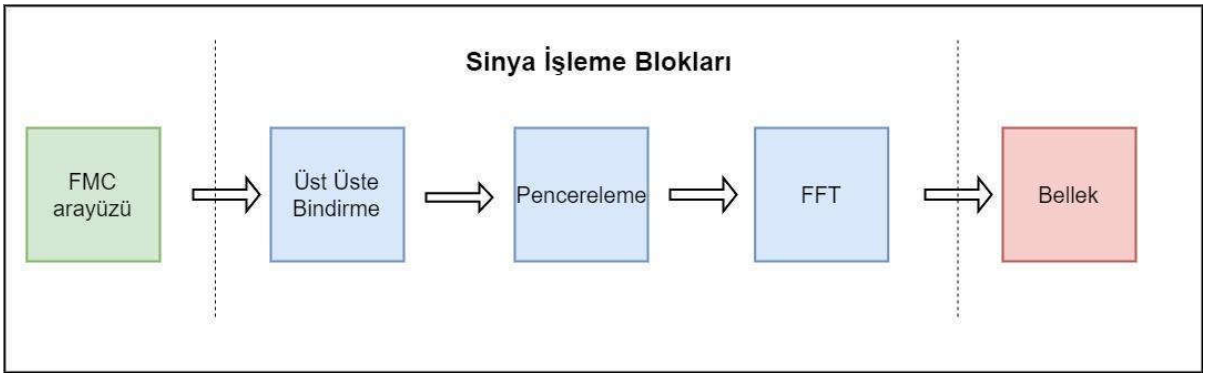
3.3 FPGA Geliştirme ve Sayısal Sinyal İşleme

Gerçek zamanlı spektrum analizi yapılabilmesi için sinyal verilerinin kayıpsız olarak toplanması ve işlenmesi gerekmektedir. RF alıcının ara frekansa düşürdüğü sinyaller yüksek hızlı bir ADC (50 MSPS örnekleme hızı öngörülmüştür) ile sayısal verilere dönüştürür. FPGA katmanı RF kartından çıkan verileri toplama, veriler üzerinde sayısal sinyal işleme ve verilerin belleğe yazılması gibi yüksek işlem hızı ve paralel işlem gerektiren görevleri yerine getirerek gerçek zamanlığı sağlar.



Şekil 3.6: Sistem FPGA katmanı genel mimarisi

Çalışma kapsamında geliştirilen FPGA tasarımına ait genel mimari Şekil 3.6 'da gösterilmektedir. ADC üzerinden gelen sinyal verileri, FMC ara yüzü ile sabit saat hızında AXI stream formatına dönüştürülerek sinyal işleme bloklarına iletilir. Sinyal işleme blokları ham verileri bir takım sinyal işleme yöntemi uygulayarak spektrum verilerine dönüştürür. Sinyal işleme IP bloklarına ait detaylı tasarım mimarisi Şekil 3.7 'de verilmiştir. Sinyal işleme bloklarından çıkan veriler PL tarafından oluşturulan DMA IP bloğuyla DDR belleğe yazılır.

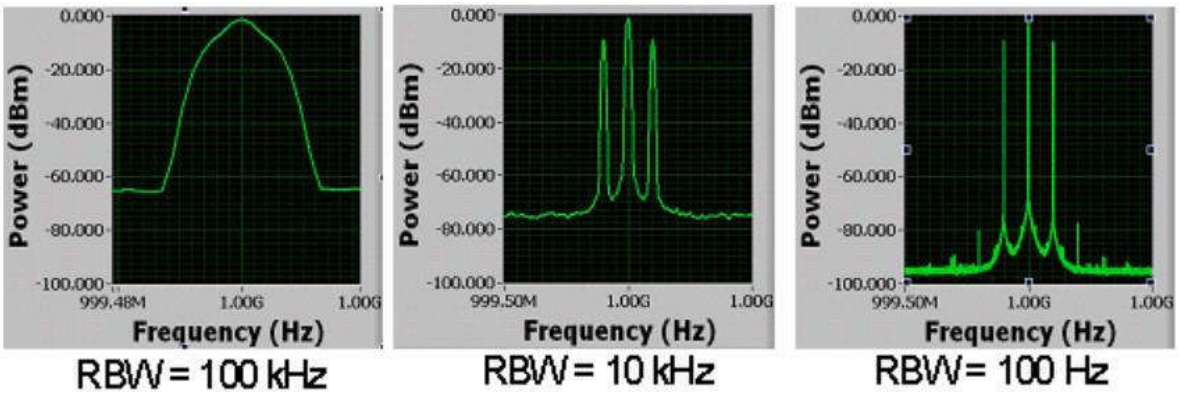


Şekil 3.7: Sinyal işleme IP blokları

3.3.1 FMC Arayüz Bloğu

RF sinyaller ilk olarak RF alıcıda ara frekansa düşürülerek ADC ile sayısal verilere dönüştürülür ve FMC konektörüne aktarılır. Bu aşamadan sonra FPGA tasarım blokları devreye girer. FMC konektöründen gelen veriler ilk FMC ara yüz IP bloğuna girer. FMC ara yüz IP bloğu CPU tarafından AXI hattı üzerinden yönetilerek ADC'den aldığı verilerin FIFO (First In, First Out) arabelleğine yazılmasını sağlar. FIFO sürekli akan verileri arabellekte tutarak havuz görevi görür. Böylece verilerin üst üste yazılmasını, verilerin taşmasını veya kaybolmasını önler.

Veriler FIFO dan sonra FIR filtreden geçer. FIR filtre IP si sinyal verilerinin seyreltilerek örnekleme hızlarının azaltılmasını sağlar. FIR filtre IP si AXI yapılandırma ara yüzü üzerinden yapılandırılabilecektir. Verilere seyreltme uygulanmasının amacı çözünürlük bant genişliğini düşürerek frekans çözünürlüğü artırmaktır. ADC lerde örnekleme hızı belirli değerlere kadar düşebilmektedir. Örneğin minimum 250 kSPS örnekleme yapan bir analog dijital çeviriciden çıkan sinyallere 8192 boyutunda FFT uygularsak, elde edeceğimiz frekans çözünürlüğü 250000/8192 ≈ 31 Hz olacaktır. FIR filtre ile 1 / 32 seyreltme uygularsak 1 Hz frekans çözünürlüğü elde edilebilir. Böylece sinyal spektrumları daha iyi analiz edilebilir. Şekil 3.8 'de aynı sinyalin farklı frekans çözünürlüklerindeki görüntüsü verilmektedir. Seyreltme işlemi için FIR seyreltme algoritması kullanılmasının amacı bant dışı sinyal ve gürültülerin bozucu etkisini ve bant dışı örnekleme gürültüsünü bastırabilmesidir.



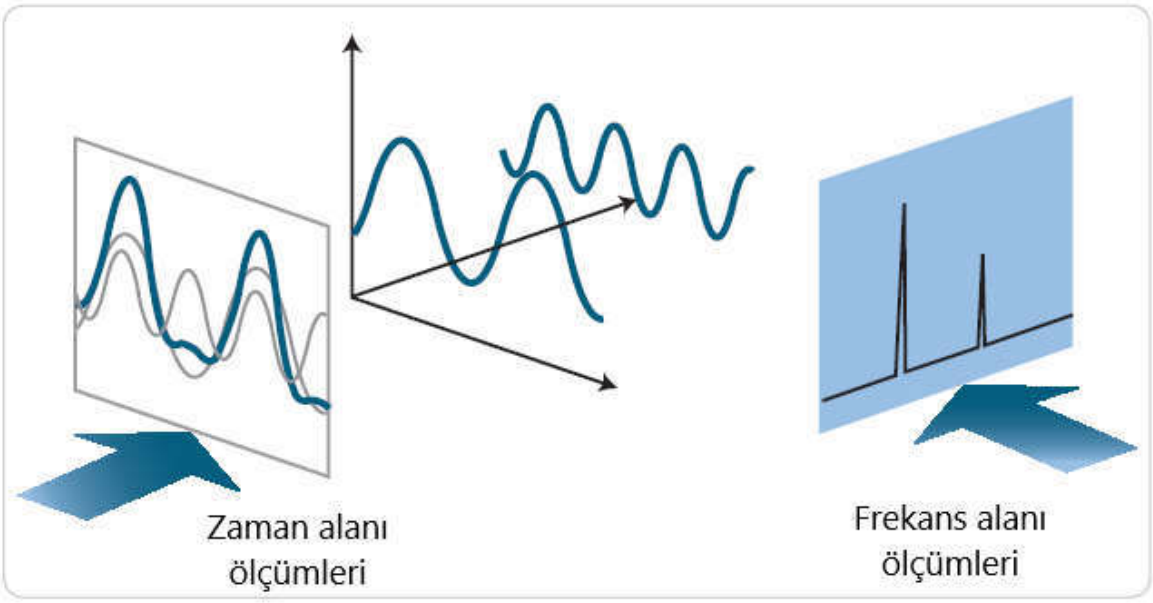
Şekil 3.8: Frekans bant çözünürlüğü etkisi [17].

FMC ara yüz bloklarından önceki IP blokları örnekleme hızı ADC nin örnekleme hızına bağlı olarak değişmektedir. Ancak FPGA saat hızı 100 MHz de çalışmaktadır. FMC ara yüz bloklarında saat hızı aynı değere çekilerek, sinyal işleme IP bloklarının aynı saat hızında ve uyumlu bir şekilde çalışmasını sağlanır. FMC ara yüz IP bloklarının bir diğer önemli görevi ise gelen verileri AXI stream veri formatına dönüştürmesidir. Böylece sinyal verileri DMA yapısı üzerinden IP bloklarına veya belleğe direk olarak aktarılabilir bir formata dönüşür. Ayrıca FMC ara yüz IP bloklarında verilerin paketlenmesi yapılır. Çünkü pencereleme, üst üste bindirme ve FFT gibi sinyal işleme blokları belirli sayıdaki sinyal verisi üzerinde çalışan yapılardır. Bu yapılara girecek veri paket büyüklüğü CPU tarafından kullanıcının istediği ölçüm değerlerine göre hesaplanarak AXI hattı üzerinden ayarlanır.

3.3.2 FFT

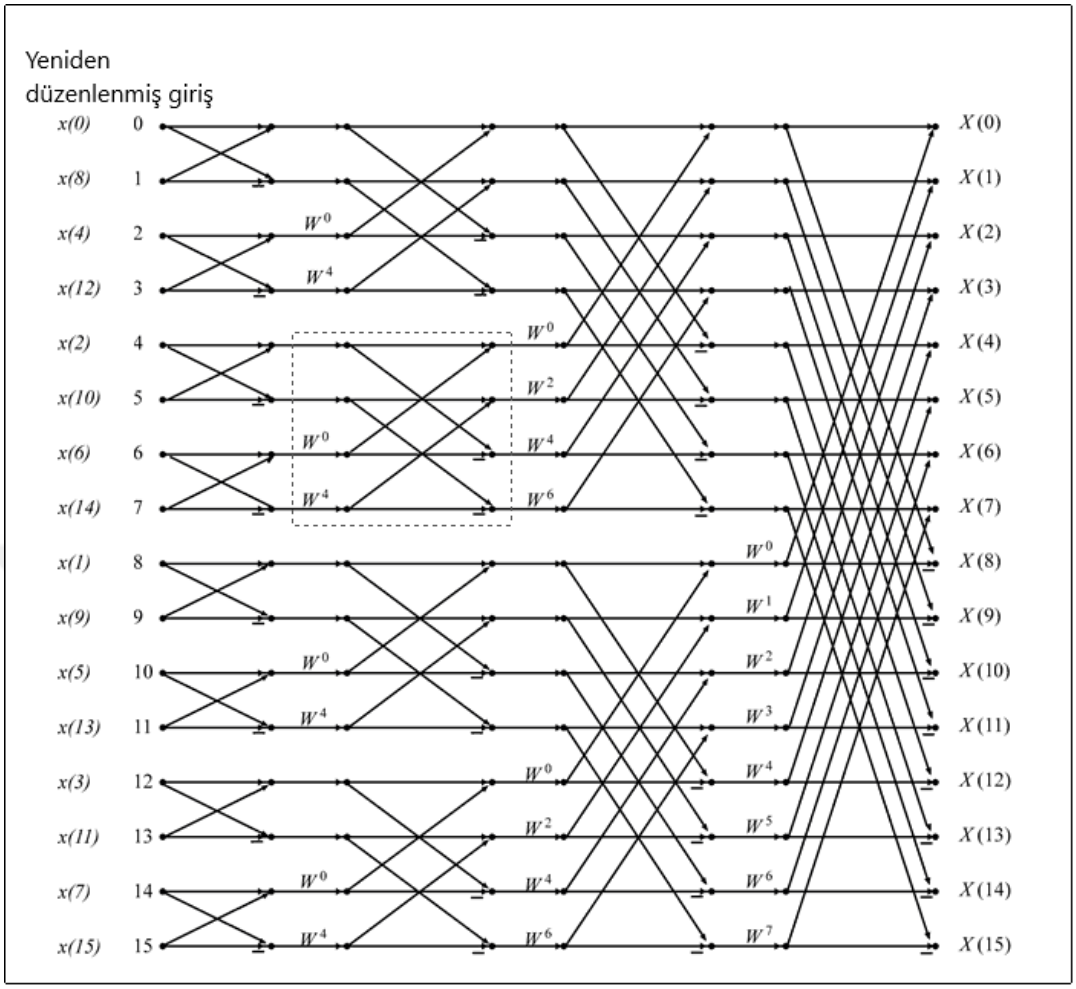
3.3.2.1 FFT Algoritması

Fourier analizi bir sinyalin zaman uzayından frekans uzayına veya frekans uzayından zaman uzayına dönüşümünü sağlar. Şekil 3.9 ‘da bir sinyalin zaman alanındaki ve frekans alanındaki ölçümleri gösterilmektedir. FFT ise bir ayrık Fourier dönüşüm (Discrete Fourier Transform) dizisini hesaplayan algoritmadır. Ayrık Fourier dönüşümü bir dizi değerin farklı frekanslardaki bileşenlere ayrıştırılmasıyla elde edilir. Ancak pratikte DFT uygulaması çok yavaş kalmaktadır. Hızlı Fourier dönüşümü böyle dönüşümleri matris ayrıştırma yöntemiyle DFT matrisini çarpanlarına ayırarak çok hızlı bir şekilde seyrek matrisler üretir. Sonuç olarak, FFT yöntemi DFT hesaplamasının karmaşıklığının azaltılmasını sağlar [18].



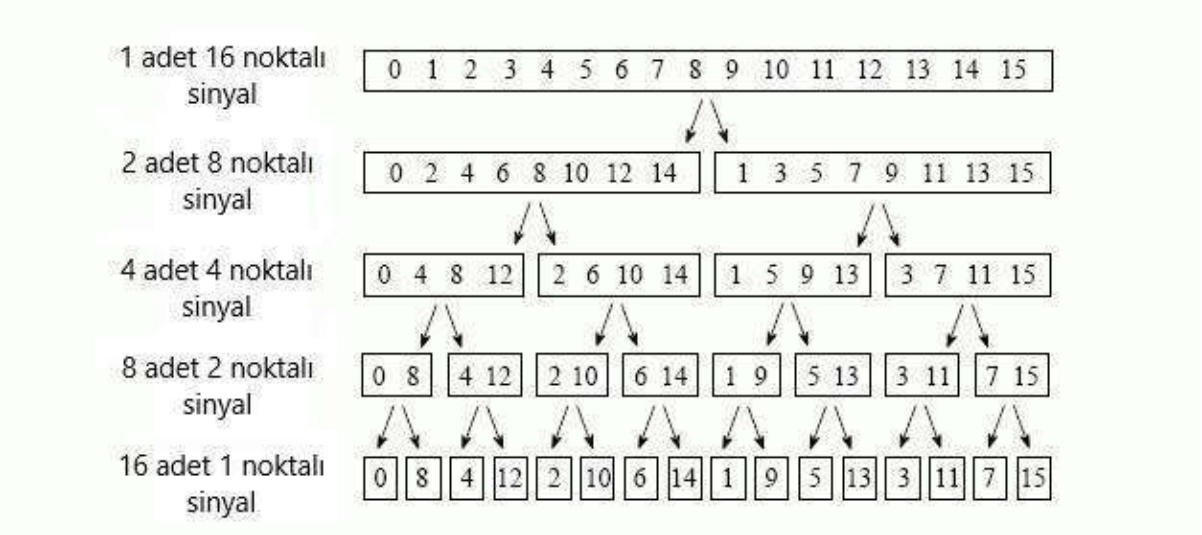
Şekil 3.9: Zaman alanı ve frekans alanı arasındaki ilişki [12].

N tane noktalı bir sinyal içi DFT yöntemi kullanılırsa N^2 sayıda hesaplama ile yapılacak işlem FFT algoritmasıyla $N\log_2 N$ sayıda hesaplama ile yapılabilmektedir. Veri boyutunun milyonlara ulaştığını düşünürsek FFT hesaplaması hız açısından çok avantajlı olmaktadır. Ayrıca dijital sistemlerde ondalık hassasiyeti sınırlı olduğundan dolayı yuvarlama hatası meydana gelmektedir. Yuvarlama hatalarında FFT algoritması DFT algoritmasına göre daha doğru sonuçlar üretmektedir. FFT algoritması diyagram üzerinde kelebeğe benzer çarpımların gerçekleşmesiyle oluştuğu için kelebek algoritması olarak adlandırılan yapılardan oluşur. Şekil 3.10 'da 16 noktalı bir sinyale ait FFT kelebek diyagramı gösterilmektedir.



Şekil 3.10: FFT Algoritması Kelebek Diyagramı [19].

FFT işlemi üç aşamadan oluşur. İlk aşamada N sayıda noktadan oluşan zaman alanındaki sinyal tek noktadan oluşan N tane zaman alanındaki sinyale dönüştürülür. İkinci adımda ise N tane zaman alanındaki sinyale karşılık gelen frekans spektrumları hesaplanır. Son olarak ise N tane frekans spektrumu sentezlenerek tek bir frekans spektrumu elde edilir. Şekil 3.11 'de FFT işleminde kullanılan zaman alanındaki sinyal ayrıştırılmasına ait bir örnek verilmiştir.



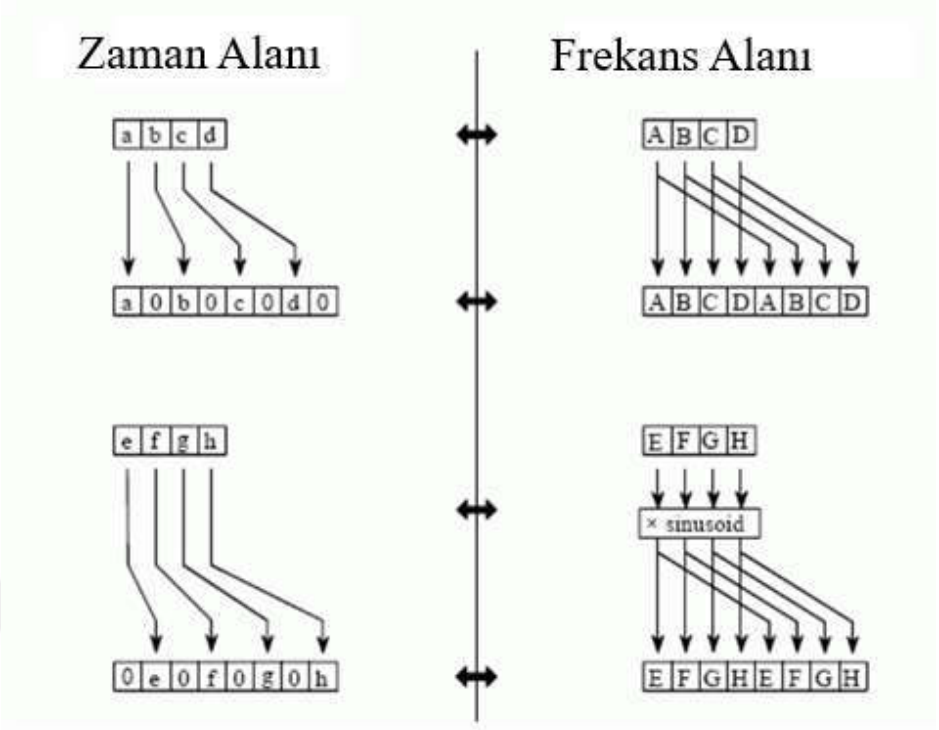
Şekil 3.11: FFT ayrıştırması. N noktalı sinyal N tane tek noktaya sahip sinyallere ayrıştırılır [20].

FFT zaman alanı ayrıştırması dijital sistemlerde bit ters sıralama algoritması ile gerçekleştirilir. Şekil 3.12 'de bit ters sıralama algoritmasına ait bir örnek verilmiştir.



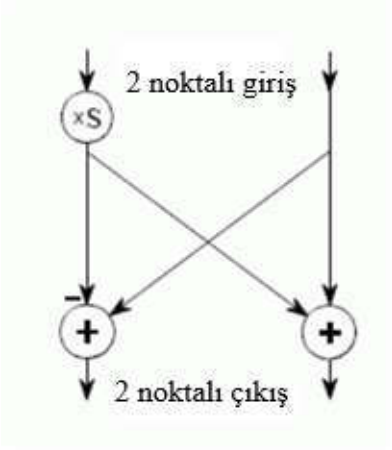
Şekil 3.12: FFT bit ters sıralama işlemi [20].

Sonraki adımda ise ayrıştırılan zaman alanındaki tek noktalı sinyalin frekansı bulunur. Bu aşamada hiçbir işleme yapmaya gerek yoktur. Çünkü sinyalin kendisi spektrumuna eşittir.



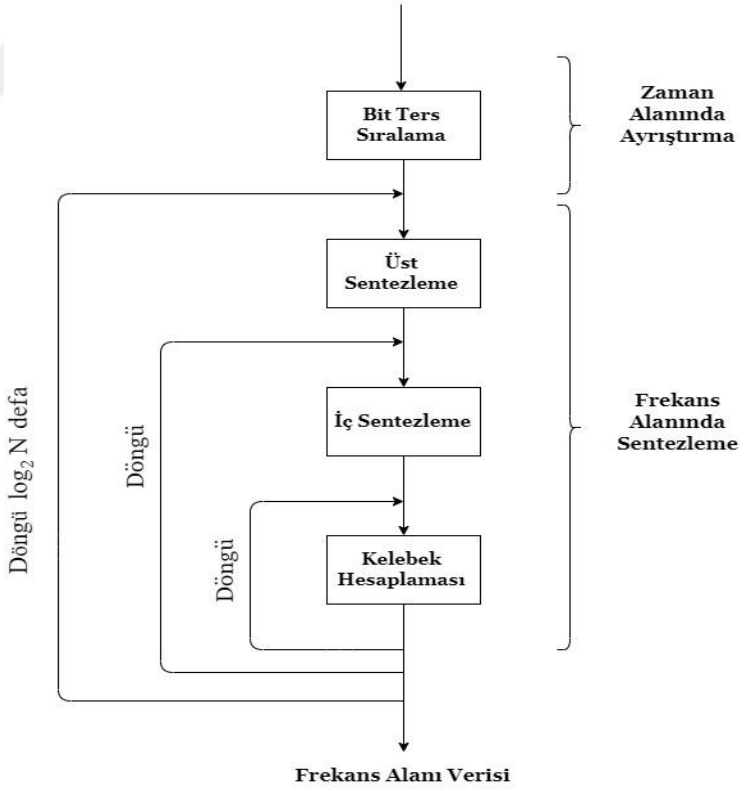
Şekil 3.13: FFT Sentezleme [20].

Son aşama ise N tane frekans spektrumunun zaman alanı ayrıştırmasının tam tersi sırada birleştirilmesi gerekir. Bu aşama bit ters sıralama algoritması mümkün olmadığı için zaman alanında birer aşama geri giderek bu işlemi yapmamız gerekir. Şekil 3.13 'te FFT sentezlemeye ait bir örnek verilmiştir. Zaman alanındaki 8 noktalı abcdefgh sinyali zaman ayrıştırması ile a0b0c0d0 ve 0e0f0g0h olacak şekilde iki adet 4 noktalı sinyale ayrıştırılmıştır. Bu sinyale ait frekans spektrumları sentezlenirken zaman alanındaki ayrıştırmaya karşılık gelecek şekilde birleştirilmesi gerekir. efgh sinyali zaman alanında ayrıştırılırken bir bit kaydırma işlemi uygulanarak seyreltilmiştir. Eğer zaman alanındaki sinyal ayrıştırma esnasında bir bit kaydırıldıysa, frekans spektrumu sentezlenirken bir sinüzoit ile çarpılması gerekir. Zaman alanındaki efgh sinyaline karşılık gelen frekans alanındaki EFGH spektrumu bir sinüzoit ile çarpılarak sentezlenir[20].



Şekil 3.14: FFT Kelebek algoritması [20].

FFT algoritmasının en temel hesaplama işlemi olan kelebek algoritmasına ait görsel Şekil 3.14 'te verilmiştir. Kelebek algoritmasında 2 tane karmaşık sayı 2 farklı karmaşık sayıya dönüştürülür.



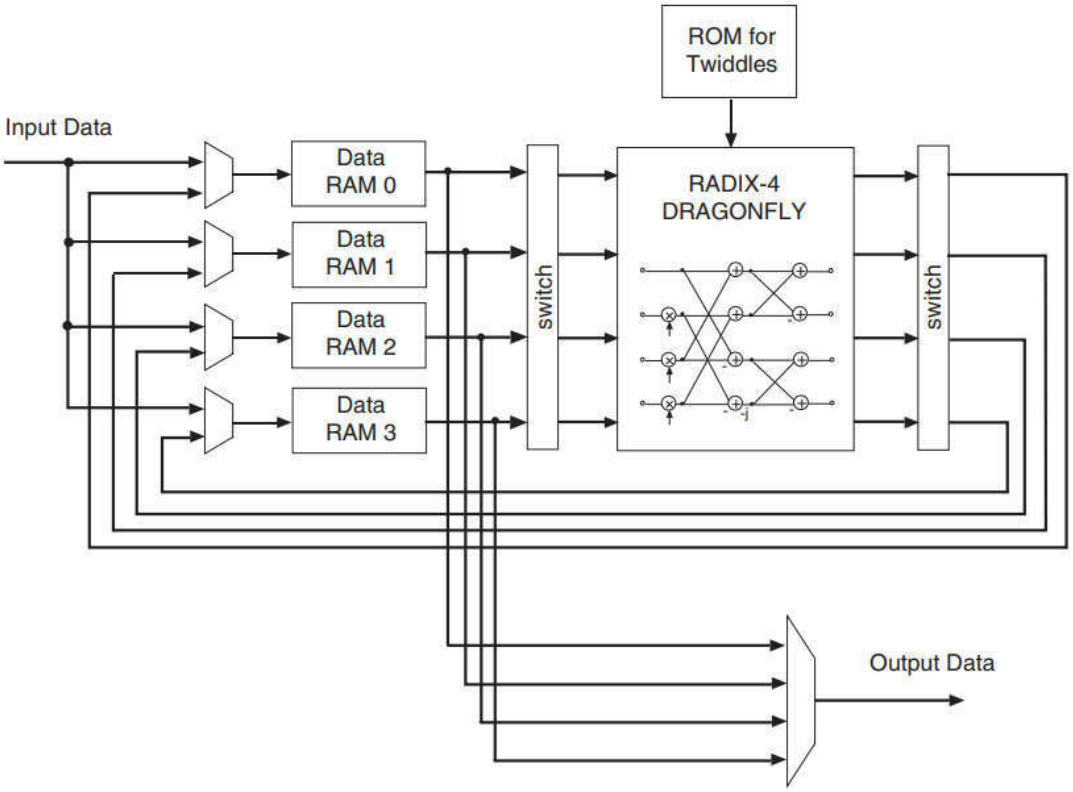
Şekil 3.15: FFT akış diyagramı [20].

Şekil 3.15 'de FFT algoritmasının aşamalarını gösteren bir akış diyagramı verilmiştir. İlk olarak bit ters sıralama işlemi ile zaman alanında ayrıştırma yapılır. Zaman ayrıştırmasından sonra

ayrıştırılan sinyallerin spektrumları bulunur. Bu aşamada bir işlem yapılmasına gerek olmadığı için akış diyagramında gösterilmemiştir. Son olarak sinyal spektrumlarının sentezlenerek tek bir spektrum elde edilmesi için döngü yapısı kullanılır. Üst sentezleme döngüsü ayrışık spektrumların sentezlenerek bir üst aşamaya geçişi için uygulanır. Üst sentezleme döngüsü her uygulandığında spektrumların nokta sayısı iki katına çıkar ve ayrışık sinyal sayısı yarıya düşer. İç sentezleme döngüsü ise bir seviyedeki birbirinden bağımsız sentezlenmiş ayrışık frekans spektrumlarına (Şekil 3.11'deki kutulardan her biri) uygulanır. İç ve dış sentezleme işlemleri hem döngülerin başlangıç ve bitiş indekslerine karar verir hem de kelebek hesaplaması için gereken sinüzoitlerin hesaplamasını gerçekleştirir. En içte ise her bir frekans spektrumu içerisindeki noktaları hesaplayan kelebek algoritması bulunur [20].

3.3.2.2 FFT IP Bloğu

FPGA üzerinde Fourier dönüşümü FFT IP bloğu tarafından gerçekleştirilmektedir. FFT IP bloğu olarak Xilinx firması tarafından sunulan özelleştirilebilir IP bloğu kullanılacaktır. Xilinx FFT IP bloğu mikroişlemci tarafından AXI ile yönetilebilmektedir. IP bloğu veri girişi AXI Stream formatındadır ve veriler uygulanacak FFT uzunluğuyla aynı boyutlarda paketler halinde alınmalıdır. Xilinx FFT IP bloğu sabit dönüşüm uzunluğuna sahip olabildiği gibi çalışma esnasından dönüşüm uzunluğu değiştirilebilir olarak da ayarlanabilmektedir. Spektrum analizör çalışmasında, uygulanacak FFT uzunluğu seçilen ölçüm frekans aralığı ve bant genişliği çözünürlüğü (RBW) değerlerine bağlı olarak değişeceği için FFT IP bloğu çalışma esnasından değiştirilebilir olarak yapılandırılacaktır. FFT IP bloğu giriş veri tipi sabit noktalı veya kayan noktalı olarak seçilebilmektedir. IP bloğunun sabit ölçeklendirme ve kayan ölçeklendirme olmak üzere iki adet seçeneğe sahiptir. Bu seçenek dinamik aralığın en iyi değerde tutulması amacıyla veri ile değişen ölçeklendirme uygulayan kayan noktalı ölçeklendirme seçilmiştir.



Şekil 3.16: Xilinx FFT bloğu Radix-4 Mimarisi [21].

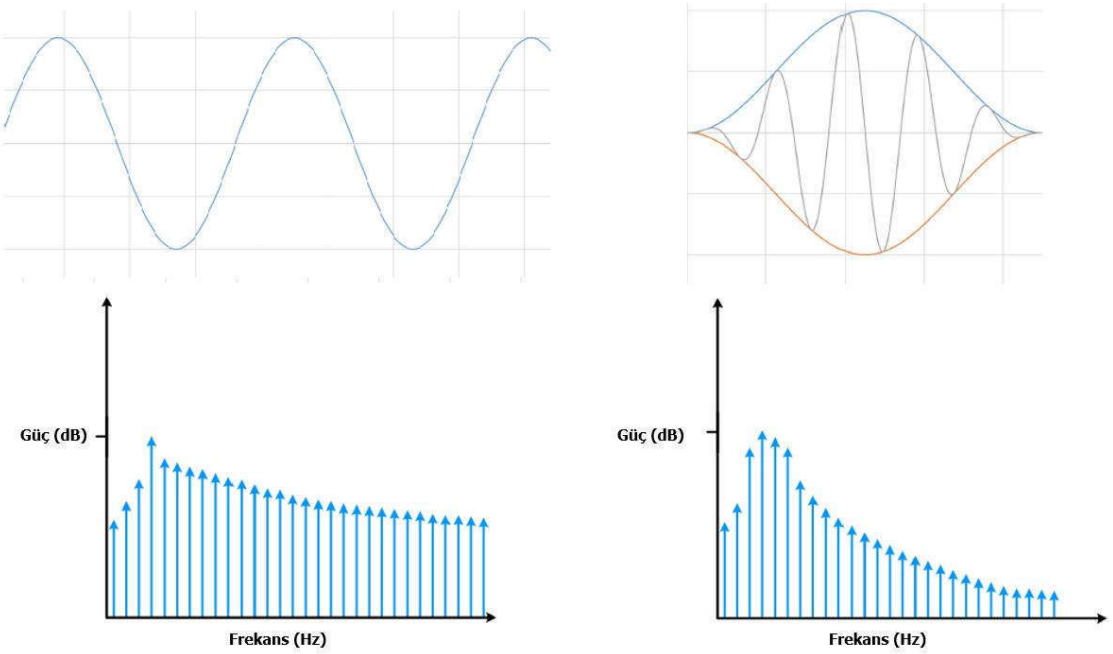
IP bloğuna ait mimari seçiminde ise Pipelined, Radix-2 ve Radix-4 seçenekleri mevcuttur. Mimarilerde farklı ayrıştırma yöntemleri kullanılmaktadır. Radix-4 ayrıştırması uygulandığında N noktalı FFT işlemi, her biri N/4 tane Radix-4 kelebek yapısı kullanan $\log_4 N$ tane aşama gerçekleştirilir. Nokta sayısı 4 ün kuvvetlerine eşit değilse ekstradan Radix-2 mimarisi uygulanabilir. Radix-2 ayrıştırması uygulandığında N noktalı FFT işlemi, her biri N/2 tane Radix-2 kelebek yapısı kullanan $\log_2 N$ tane aşama gerçekleştirilir.

Giriş veri saat hızı çıkış veri saat hızına eşit veya yakın olduğu durumlarda Pipelined mimarisi kullanılabilir. Giriş veri saat hızının çıkış saat hızının 4 katı olduğu durumlarda Radix-4 mimarisi de kullanılabilir. Şekil 3.16'da Xilinx FFT IP bloğuna ait Radix-4 mimarisi görülmektedir. Ayrıca FFT IP bloğuna ait Matlab C modeli Xilinx firması tarafından sunulmaktadır. IP bloğunun C modeli kullanılarak Matlab üzerinden benzetimi gerçekleştirilebilmektedir.

3.3.3 Pencereleme (Windowing)

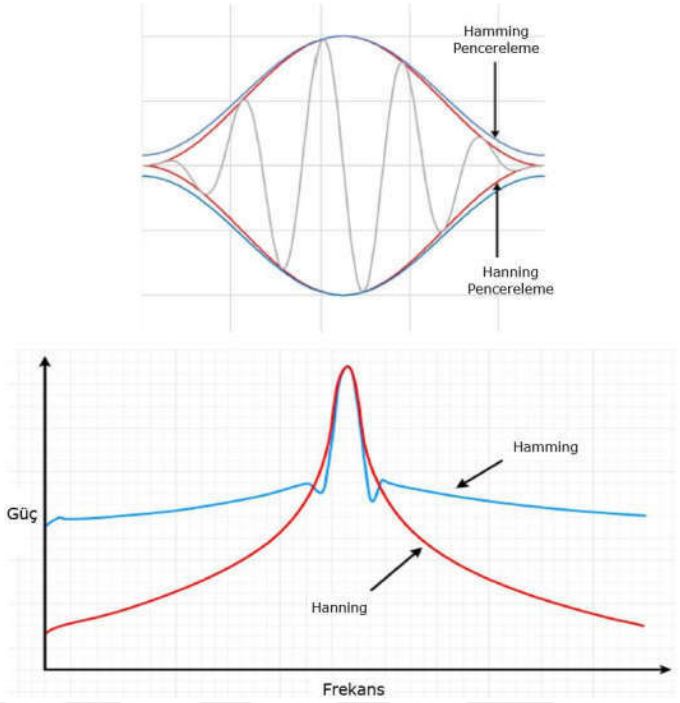
3.3.3.1 Pencereleme Algoritması

Zaman eksenindeki bir sinyali örneklerken uygulanan DFT dönüşümü sonucunda sinyalin frekans elemanları 0-1/2f_s arasında oluşur. Eğer giriş sinyali Nyquist frekansından daha büyük frekans elemanlarına sahipse bu elemanlar bant içerisinde bir yerlere bozucu etki yapacaktır. Bant dışındaki elemanların bozucu etkiyi ortadan kaldırmak için bir filtre uygulanması gerekir. Bu işlem için en iyi sayısal filtreleme yöntemi pencereleme (Windowing) teknikleridir.



Şekil 3.17: Pencereleme algoritması [22].

Gerçek hayatta sinyaller süreklidirler. Ancak dijital sistemlerde sinyaller verilerle ifade edilirler ve süreklidirler. Pencereleme yöntemi ise veri bloklarının sınırlarındaki süreksizliği azaltmak için uygulanır. Pencereleme veri bloklarına ağırlıklı çarpımlar uygulayarak sınırları sorunsuzca sıfıra eşitler. Böylece veri blokları uygulanacak türev işlemleri için sürekli bir hale gelmiş olur[11].



Şekil 3.18: Hanning –Hamming pencereleme algoritmaları kıyaslaması [22].

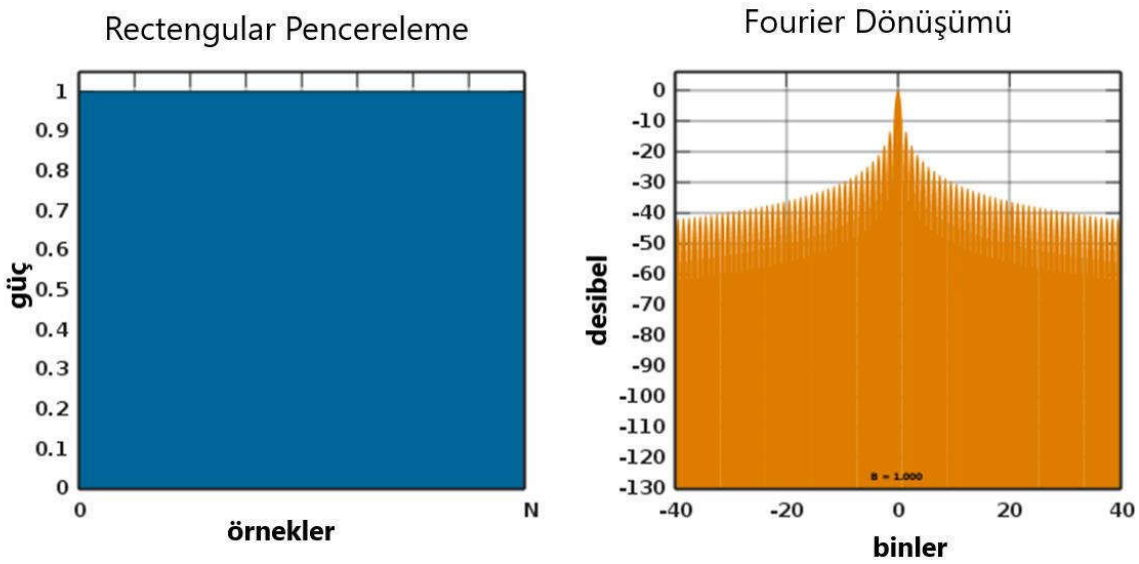
Şekil 3.17 'de üstte bir sinyale zaman ekseninde pencereleme uygulanmamış ve uygulanmamış halleri gösterilirken altta FFT uygulandıktan sonra frekans alanındaki spektrum grafikleri gösterilmektedir. Frekans alanındaki görseller karşılaştırıldığında pencereleme uygulanan sinyalin daha az gürültüye sahip olduğu ve daha net olduğu görülmektedir.

Tablo 3.2: Pencereleme Algoritmaları karşılaştırma tablosu [10].

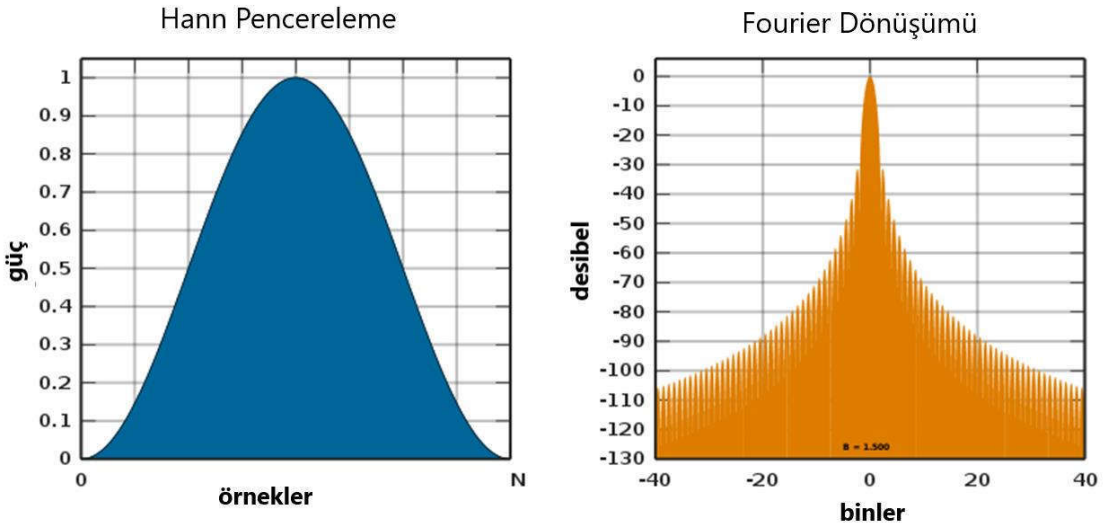
Pencere	Spektral Sızıntı	Güç Doğruluğu	Frekans Çözünürlüğü
Blackman	En iyi	İyi	Orta
Flattop	İyi	En iyi	Zayıf
Gaussian	Orta	İyi	Orta
Rectangle(yok)	Zayıf	Zayıf	En iyi
Hann(ing)	İyi	Orta	İyi
Hamming	Orta	Orta	İyi
Kaiser	İyi	İyi	Orta

Farklı karakteristiklere sahip birçok pencereleme algoritması bulunmaktadır. Şekil 3.18 'de Hann ve Hamming pencereleme algoritmalarının farklı karakteristiklere sahip olduğu görülmektedir. Tablo 3.2'de ise en çok kullanılan pencereleme algoritmalarına ait kıyaslama yapılmaktadır. Spektrum analizörlerde kullanıcılara yapacakları ölçüm karakteristiklerine

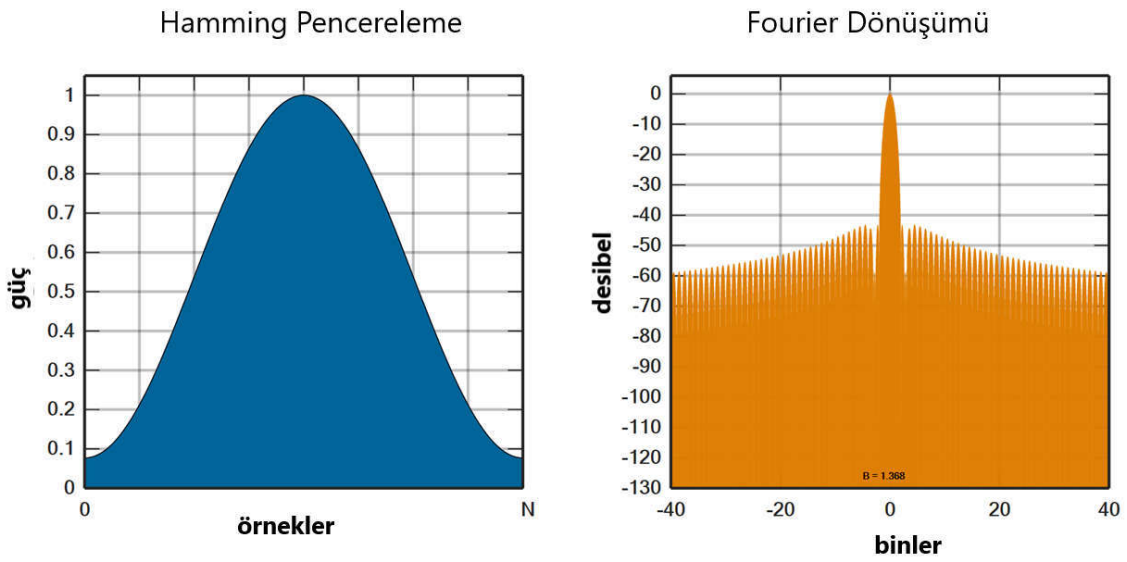
uygun seçim yapabilecekleri birkaç tane pencereleme algoritması seçenek olarak sunulmaktadır. Şekil 3.19 Şekil 3.20 Şekil 3.21 Şekil 3.22 ve Şekil 3.23 'te en çok kullanılan pencereleme yöntemlerinin aynı sinyallere uygulanmış zaman alanındaki ve pencereleme uygulandıktan sonra Fourier dönüşüm alınmış halleri gösterilmektedir.



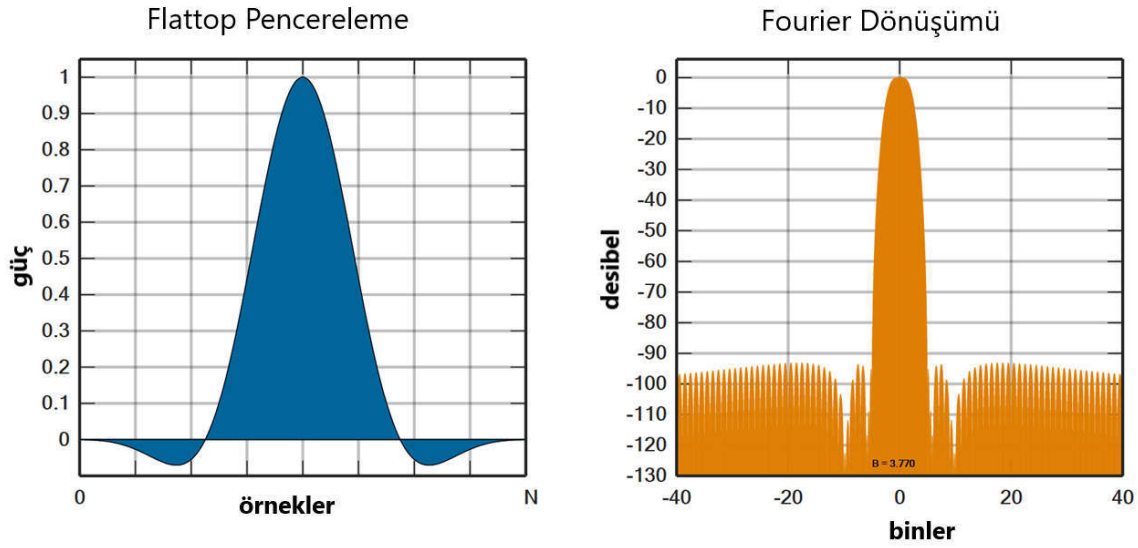
Şekil 3.19: Dikdörtgen Pencereleme ve Fourier dönüşümü [23].



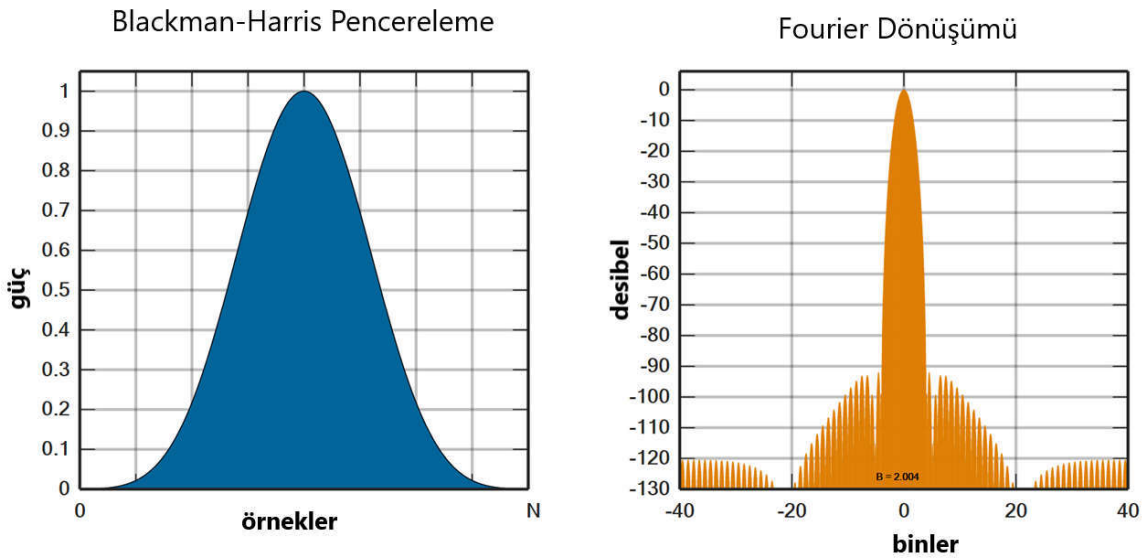
Şekil 3.20: Hanning Pencereleme ve Fourier dönüşümü [23]



Şekil 3.21: Hamming Pencereleme ve Fourier dönüşümü [23].



Şekil 3.22: Flat top Pencereleme ve Fourier dönüşümü [23].



Şekil 3.23: Blackman-Harris Pencereleme ve Fourier dönüşümü [23].

3.3.3.2 Pencereleme IP Bloğu

Çalışmada zaman alanındaki sinyallere FFT IP bloğuna girmeden önce pencereleme algoritması uygulamak amacıyla IP blok tasarım gerçekleştirilecektir. Pencereleme IP bloğu AXI yapılandırma ara yüzü ile mikroişlemci tarafından yönetilecektir. Pencereleme IP bloğu Hanning, Hamming, Flattop, Blackman-Harris ve dikdörtgen olmak üzere beş farklı pencereleme algoritması uygulayabilecek şekilde tasarlanması planlanmaktadır.

Dikdörtgen pencereleme yönteminde veri bloğuna hiçbir işlem yapılmaz. Zaman alanındaki sinyal olduğu gibi FFT IP bloğuna iletilir.

Hanning pencereleme algoritması zaman alanındaki sinyal bloğunun sınırlarını sıfıra eşitlemektedir. Hanning algoritmasına ait denklem (3.1) 'de verilmiştir. N veri bloğundaki toplam nokta sayısına karşılık gelirken n ise noktanın indeksine karşılık gelmektedir. Denklemde her bir indekse karşılık gelen bir katsayı bulunur ve o indeksteki noktanın değeri ile çarpılır.

$$w(n)\text{Hanning} = 0.5 - 0.5\cos\left(\frac{2\pi n}{N}\right) \tag{3.1}$$

Hamming pencereleme algoritması Hanning algoritmasıyla benzer bir yapıya sahiptir. Ancak sınırdaki sinyal değerlere sıfıra zayıflatılmaz. Hamming algoritmasına ait denklem (3.2) 'de

verilmiştir. Hamming pencereleme algoritmasında da yine aynı şekilde katsayılar hesaplanarak sinyal bloğuna uygulanır.

$$w(n)\text{Hamming} = 0.54 - 0.46\cos\left(\frac{2\pi n}{N}\right) \quad (3.2)$$

Blackman-Harris algoritmasına ait denklem (3.3) 'te verilmiştir. Denklemdeki katsayılar a_0=0.35875, a_1=0.48829, a_2=0.14128 ve a_3=0.01168 olacak şekilde uygulanmıştır. Blackman-Harris yönteminde de indekslere karşılık gelen katsayılar hesaplanarak veri bloğuna çarpan olarak uygulanır.

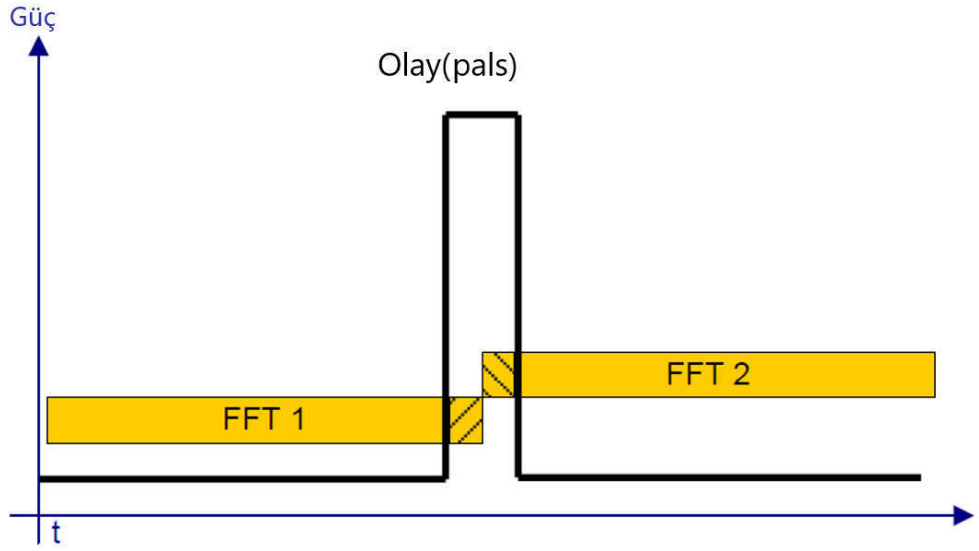
$$w(n)\text{Blackman} - \text{Harris} = a_0 - a_1\cos\left(\frac{2\pi n}{N}\right) + a_2\cos\left(\frac{4\pi n}{N}\right) - a_3\cos\left(\frac{6\pi n}{N}\right) \quad (3.3)$$

Flattop algoritmasına ait denklem (3.4) 'te verilmiştir. Denklemdeki katsayılar a_0=0.21557895, a_1=0.41663158, a_2=0.277263158, a_3=0.083578947, a_4=0.006947368 olacak şekilde uygulanmıştır.

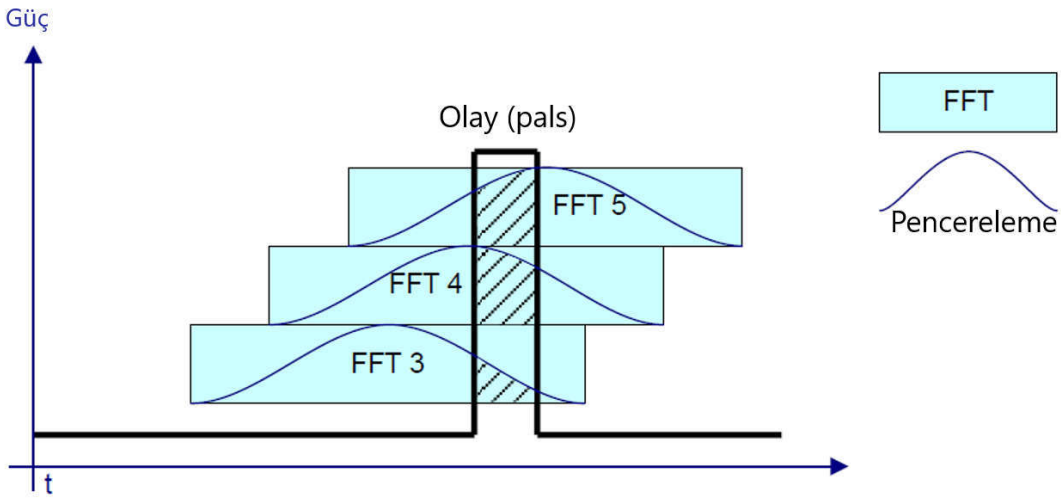
$$w(n)\text{Flattop} = a_0 - a_1\cos\left(\frac{2\pi n}{N}\right) + a_2\cos\left(\frac{4\pi n}{N}\right) - a_3\cos\left(\frac{6\pi n}{N}\right) + a_4\cos\left(\frac{8\pi n}{N}\right) \quad (3.4)$$

3.3.4 Üst Üste Bindirme

Pencereleme uygulanan veri bloklarının sınırları ağırlıklı çarpan uygulanarak sıfıra doğru yaklaştırılır. Bu yöntem veri bloklarının sınırlarındaki sinyal değerlerinin zayıflatılması sonucu veri blokları arasında boşluklar oluşmasına neden olabilir. Gerçek zamanlı spektrum analizörlerde pencerelemeden dolayı oluşabilecek boşlukların önlenerek geçiş sinyallerinin kaybedilmemesi amacıyla üst üste bindirme yöntemi uygulanır. Şekil 3.24 ve Şekil 3.25 'te üst üste bindirme yönteminin pencerelemeden dolayı oluşan sinyal kayına etkisi gösterilmektedir. Ayrıca üst üste bindirme yöntemi sayesinde daha fazla FFT karesi oluşturularak spektrumların güncellenme hızı artar.

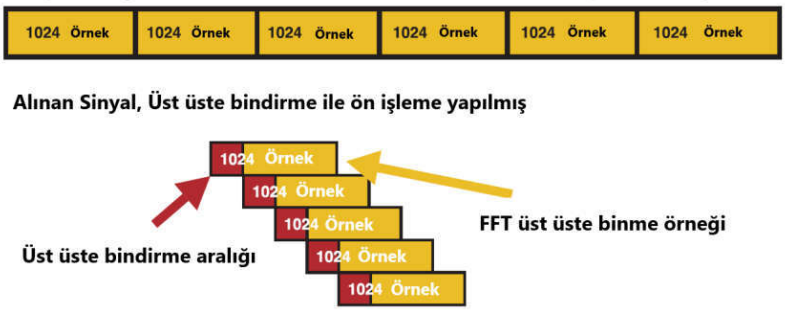


Şekil 3.24: Üst üste bindirme uygulanmamış iki FFT karesi arasındaki sinyal olayı [24].



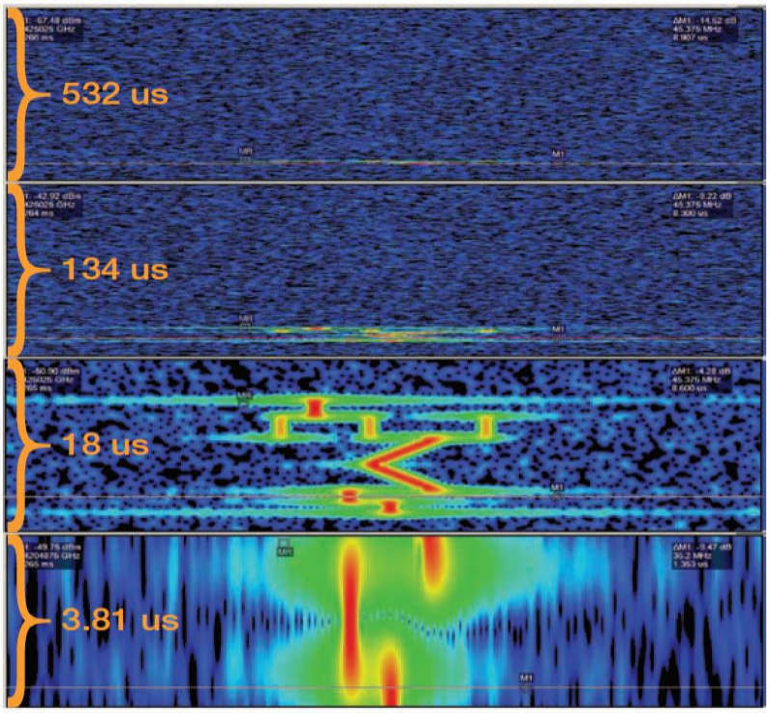
Şekil 3.25: Üst üste bindirme uygulanmış FFT kareleri arasındaki sinyal olayı [24].

Üst üste bindirme yöntemi zaman alanındaki veri bloklarının FFT uygulanması amacıyla tekrardan kullanılmasıyla gerçekleştirilir. Yüzde 75 üst üste bindirme uygulanan bir sinyalde veriler FFT işlemi için 4 kez kullanılır. Şekil 3.26 'da yüzde 75 üst üste bindirme yöntemine ait bir görsel verilmektedir. Uzunluğu 1024 olan veri bloğuna yüzde 75 üst üste bindirme uygulanması demek bir veri bloğunun 3/4 ünün (768 noktanın) bir sonraki FFT işlemi için tekrardan kullanılması anlamına gelmektedir.



Şekil 3.26: Yüzde 75 üst üste bindirme [25].

Üst üste bindirme yönteminde aynı miktardaki veriyle ait daha fazla FFT karesi oluşturulur. Daha fazla FFT karesinin oluşturulması demek sinyal spektrumlarının hareketlerinin zamana bağlı değişimlerinin daha hassas olarak gözlemlenebilmesini sağlamak demektir. Şekil 3.27 'de Aynı sinyale dört farklı üst üste bindirme yüzdeleri uygulanmıştır. Şekil incelendiğinde üst üste bindirme oranı artırıldıkça sinyal olayının daha detaylı olarak görülebildiği anlaşılmaktadır. Çalışmanın daha sonraki aşamalarında, üst üste bindirme yöntemi için IP blok tasarımı yapılması planlanmaktadır.



Şekil 3.27: Üst üste bindirme yöntemi (Yukardan aşağıya %0, %75, %97, %99.9 üst üste bindirme uygulanmıştır) [25].

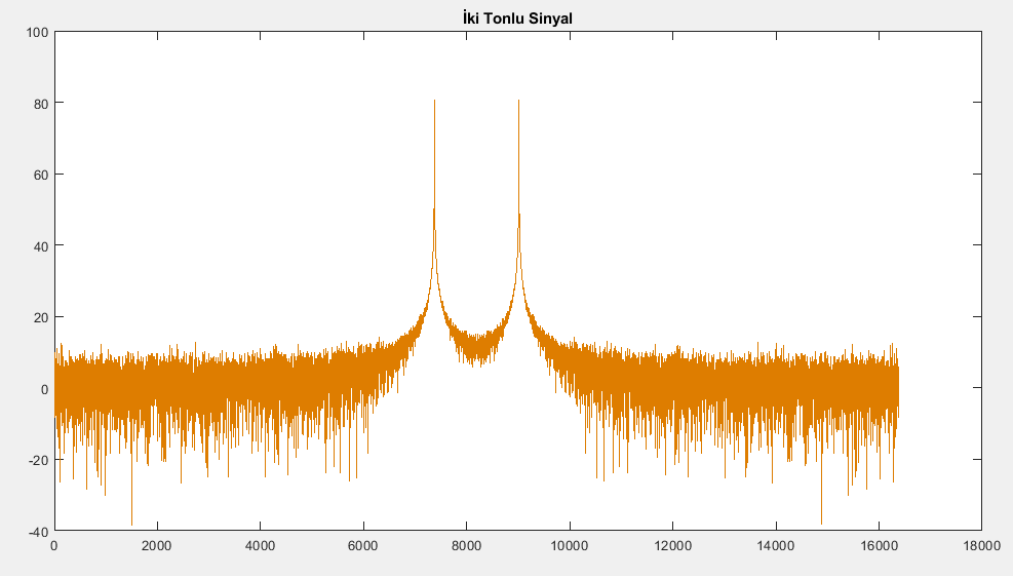
[ ]

4. BULGULAR

4.1 Spektrum Analizör Sisteminin Çalışması ve Değerlendirme

Geniş bantlı spektrum analizör sistemi çalışması geliştirme kartları üzerinden devam etmektedir. Sistemin geliştirmesi kapsamında yapılacak çalışmalarının benzetimlerinin gerçekleştirilmesi ve doğrulamasının yapılmasında Matlab yazılımından faydalanılmıştır. Matlab ile üretilen farklı karakteristikteki sinyallere geliştirilen algoritmalar uygunalarak Matlab fonksiyonlarıyla karşılaştırılmıştır. Üretilen sinyal verileri Vivado test benzetimine girdi olarak verilerek IP bloklarının çıktıları alınmıştır.

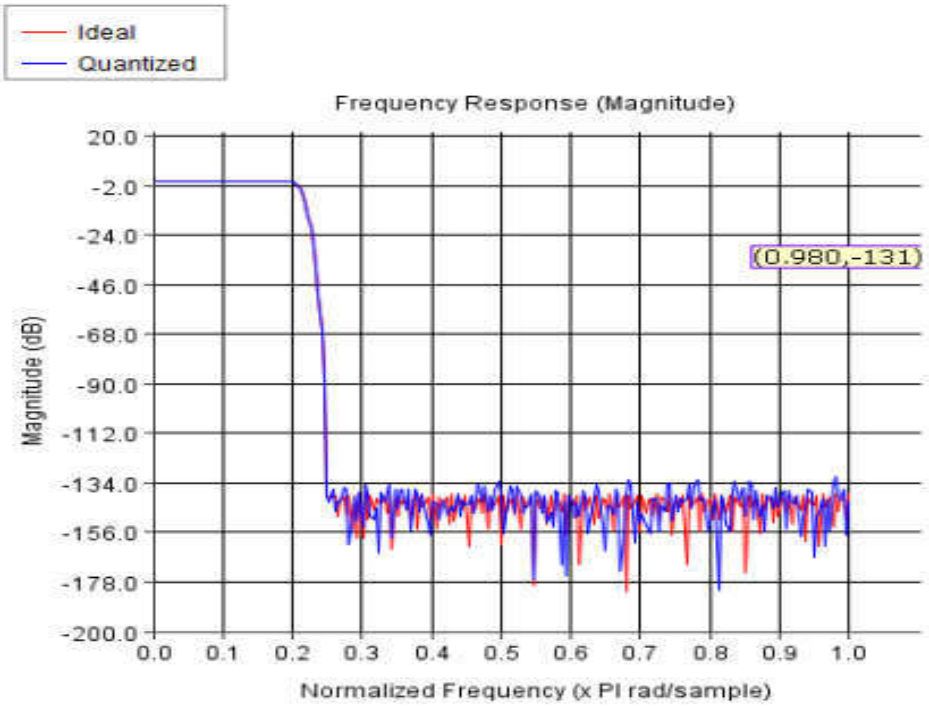
Gelinen aşamada spektrum analiz yazılımı henüz yapılamadığından işlenen sinyal verileri Matlab ortamında grafiksel çıktılara dönüştürülmüştür. Geliştirme kartı üzerindeki seri haberleşme ara yüzü ile spektrum ölçüm parametreleri girilerek cihaz kontrolü ve yapılandırmasının başarılı bir şekilde çalıştığı doğrulanmıştır. Çalışmadan geliştirilen sistemler kullanılarak 25 MHz bant genişliğinde bir sinyalin spektrum verileri elde edilebilmektedir. Şekil 4.1 ‘de sistem tarafından işlenen iki tonlu bir sinyali Matlab yazılımıyla çizdirilmiş spektrum grafiği verilmektedir.



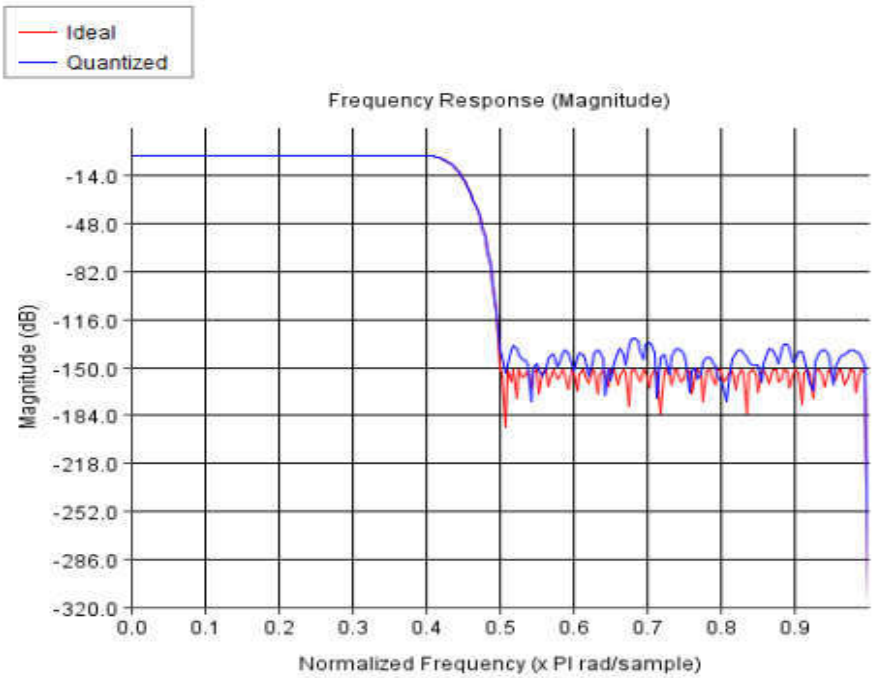
Şekil 4.1: Matlab ile üretilen iki tonlu sinyalin Xilinx FFT modeli çıktısı

4.2 FIR Filtre IP Bloğu Değerlendirmesi

FIR filtre blokları FPGA üzerinde tasarlanmadan önce Matlab kullanılarak filtre katsayıları hesaplanmıştır. Matlab ile tasarlanan filtrenin frekans cevabı ve kuantalama performansının tespit edilmesi amacıyla benzetim ortamında sinyal üretilerek filtreden geçirilmiştir. Hesaplanan filtre katsayıları Xilinx FIR filtre bloğunun Matlab C modeli üzerinde kullanılarak denenmiştir. Xilinx FIR filtre C modeli 25 bit küsurat çözünürlüğüne sahip bir FIR filtre tasarımının istenilen frekans cevabını verebileceği tespit edilmiştir. Şekil 4.2 ve Şekil 4.3 'te FPGA üzerinde tasarlanan harici FIR filtre IP bloklarının frekans cevabı gösterilmektedir. Mavi çizgi ile gösterilen kuantalama etkisi her iki filtre bloğu içinde istenilen performans kıstaslarına uymaktadır.



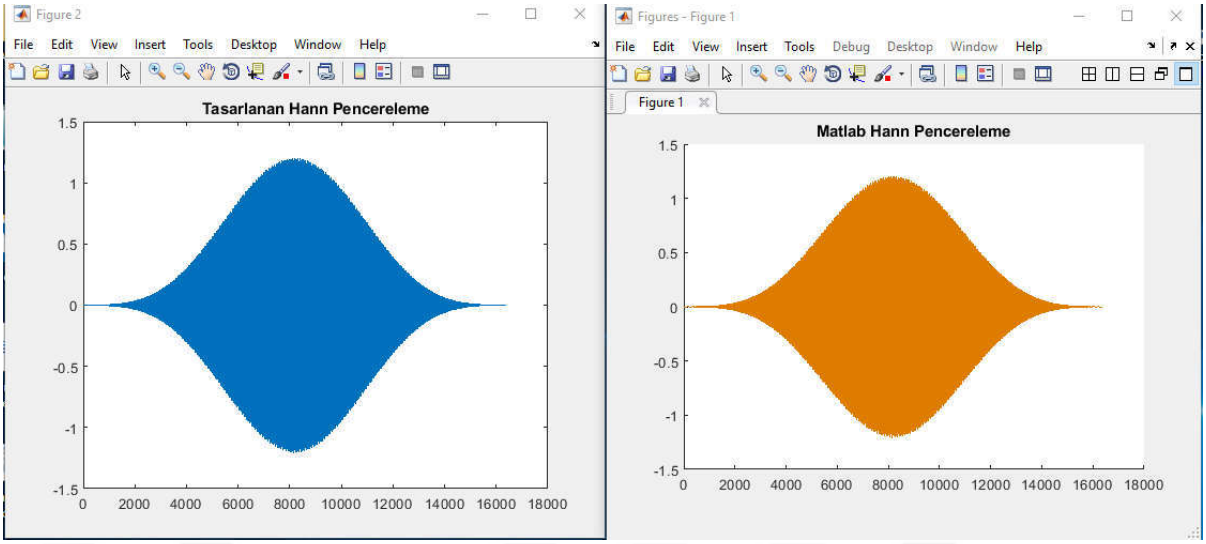
Şekil 4.2: FIR filtre (1/4) IP bloğu frekans cevabı



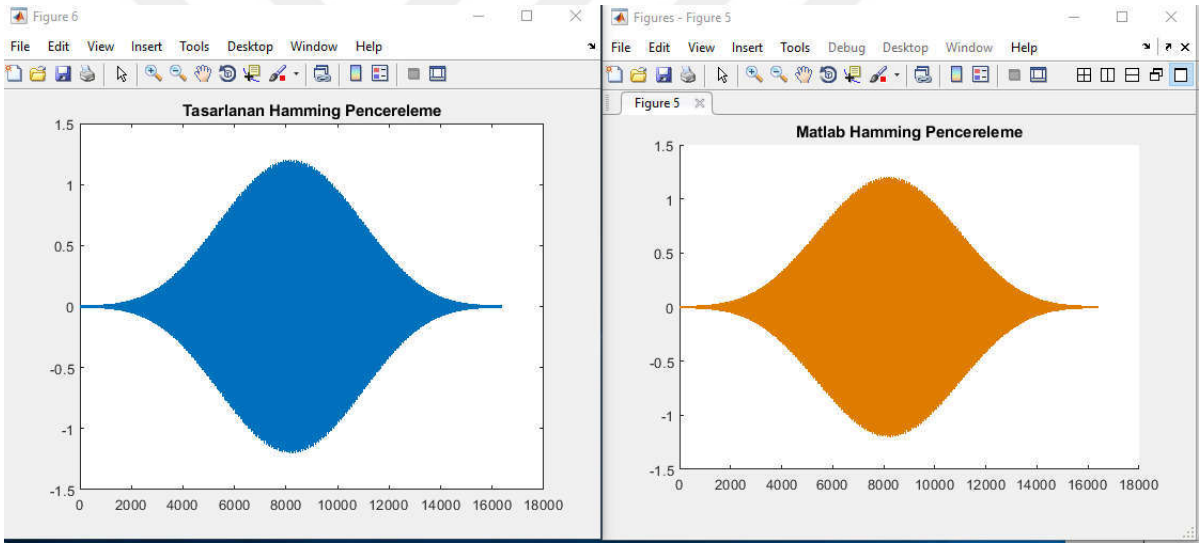
Şekil 4.3: FIR filtre (1/2) IP bloğu frekans cevabı

4.3 Pencereleme IP Bloğunun Değerlendirmesi

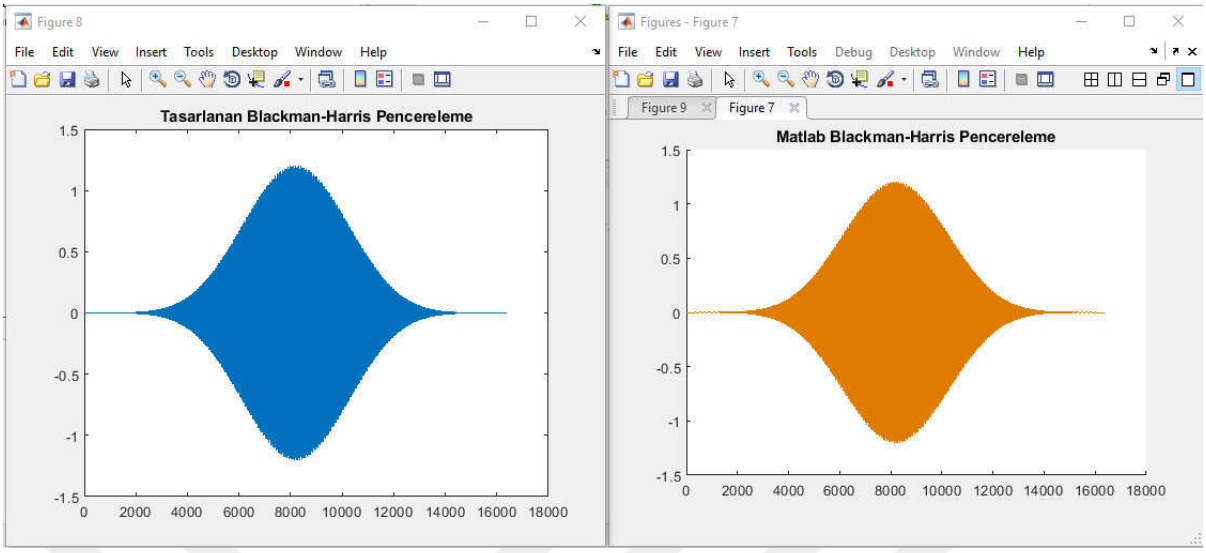
Geliştirilen pencereleme IP bloğunun çalışmasının doğrulanması ve performansının test edilmesi amacıyla Matlab yazılımından çok tonlu bir sinyal üretilerek, sinyalin zaman alanındaki örnekleri kaydedilmiştir. Kaydedilen verilere önce Matlab pencereleme uygulanmıştır. Daha sonra ise Matlab ortamında tasarlanan Pencereleme IP blok benzetimi ile kaydedilen verilere pencereleme uygulanmıştır. Matlab ile IP blok benzetimlerinin sonuçları kıyaslandığında pencereleme IP bloklarının istenilen performansı verdiği gözlemlenmiştir. Benzetimde aynı sinyal örneğine sırasıyla dört farklı pencereleme yöntemi uygulanmıştır. Şekil 4.4, Şekil 4.5, Şekil 4.6 ve Şekil 4.7 'de zaman alanındaki sinyalin Matlab pencereleme fonksiyonları ve IP blok pencereleme algoritmaları uygulanmış çıktıları kıyaslanmıştır. Tablo 4.1'de ise Matlab ve IP blok benzetimlerinin pencereleme algoritması katsayıları kıyaslanmaktadır. Tabloda incelendiğinde değerlerin birbirleriyle neredeyse aynı olduğu gözlemlenmiştir.



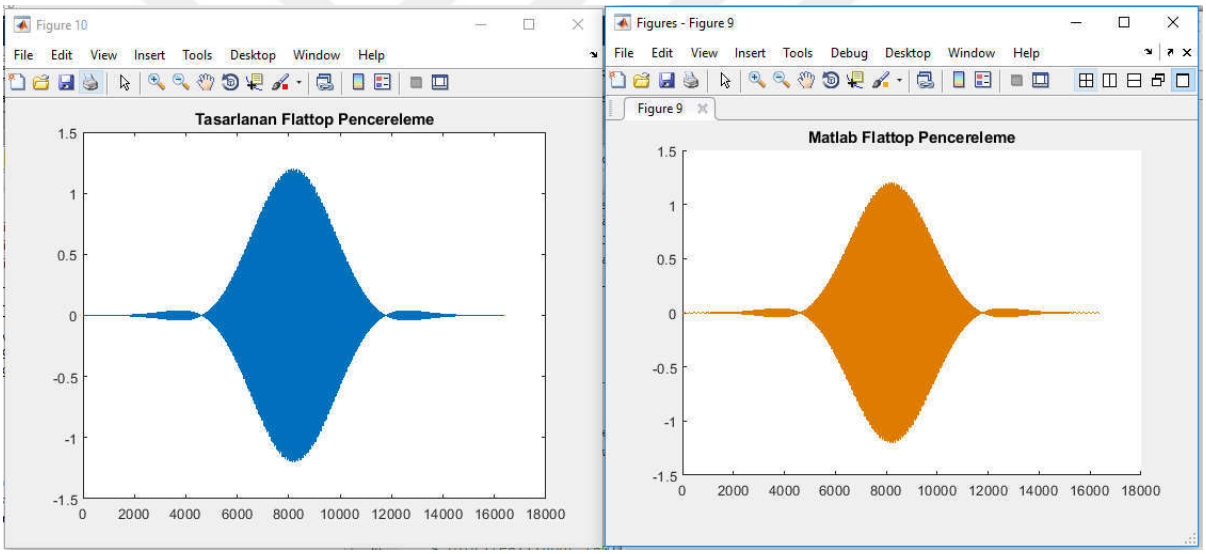
Şekil 4.4: Matlab ve Tasarlanan Hann pencereleme algoritması sonuçları



Şekil 4.5: Matlab ve Tasarlanan Hamming pencereleme algoritması sonuçları



Şekil 4.6: Matlab ve Tasarlanan IP bloğunun Blackman-Harris pencereleme sonuçları

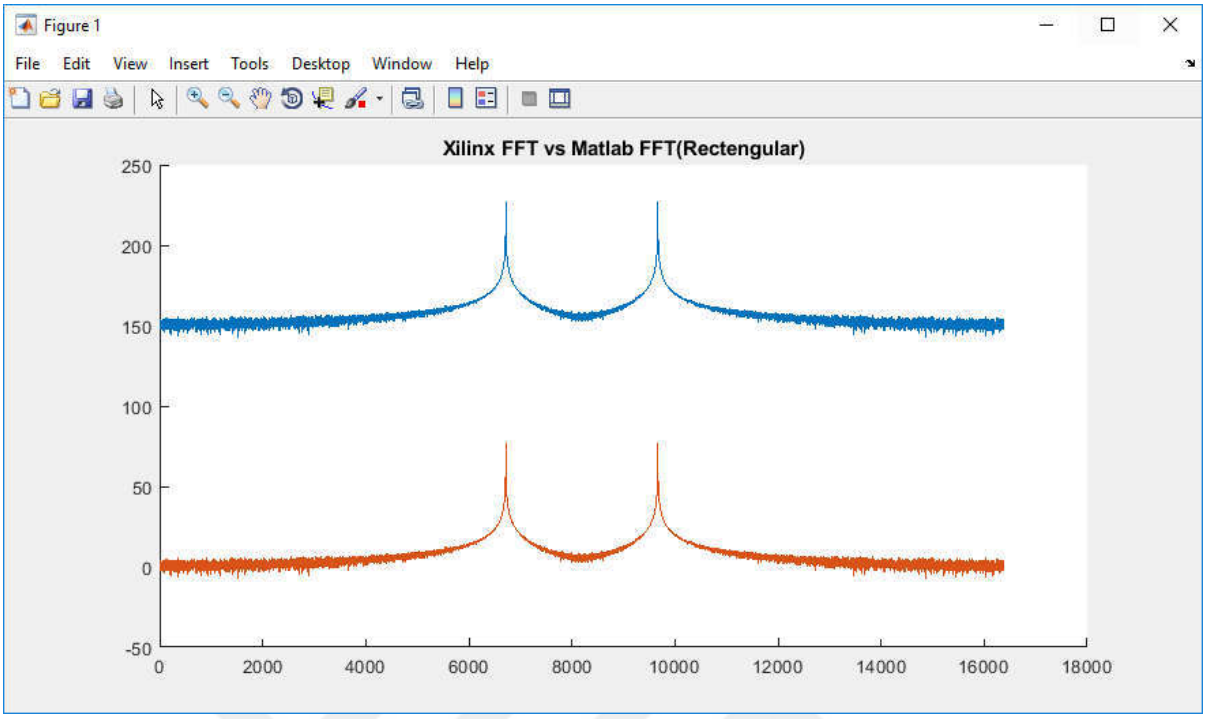


Şekil 4.7: Matlab ve Tasarlanan IP bloğunun Flattop pencereleme sonuçları

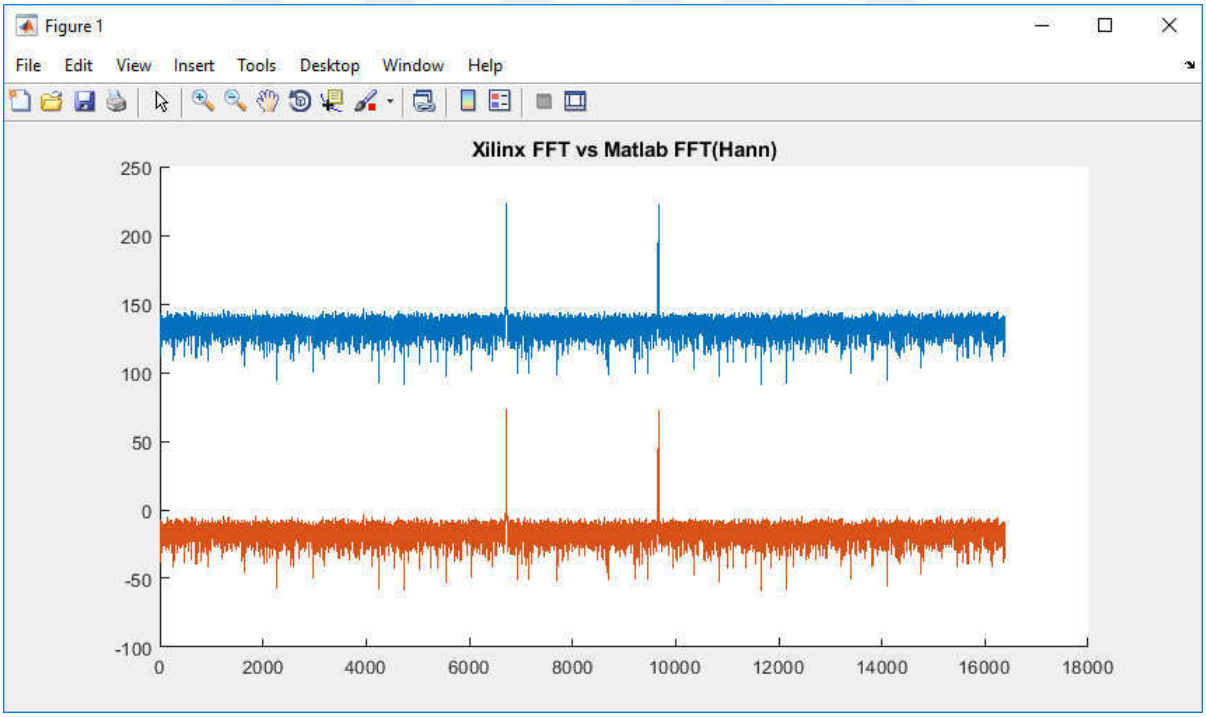
Tablo 4.1: Pencereleme Yöntemleri katsayı kıyaslama tablosu

Pencereleme Yöntemi	Katsayı	Matlab	IP blok
Hanning	a0	0.5	0.5
	a1	0.5	0.5
Hamming	a0	0.54	0.53832323656677
	a1	0.46	0.4661236775
Blackman-Harris	a0	0.35875	0.35874623729
	a1	0.48829	0.48828537
	a2	0.14128	0.14123643467
	a3	0.01168	0.01165771484375
Flat Top	a0	0.21557895	0.21557896737686
	a1	0.41663158	0.416627853859
	a2	0.277263158	0.277220654322
	a3	0.083578947	0.0835579846235
	a4	0.006947368	0.0068969723129

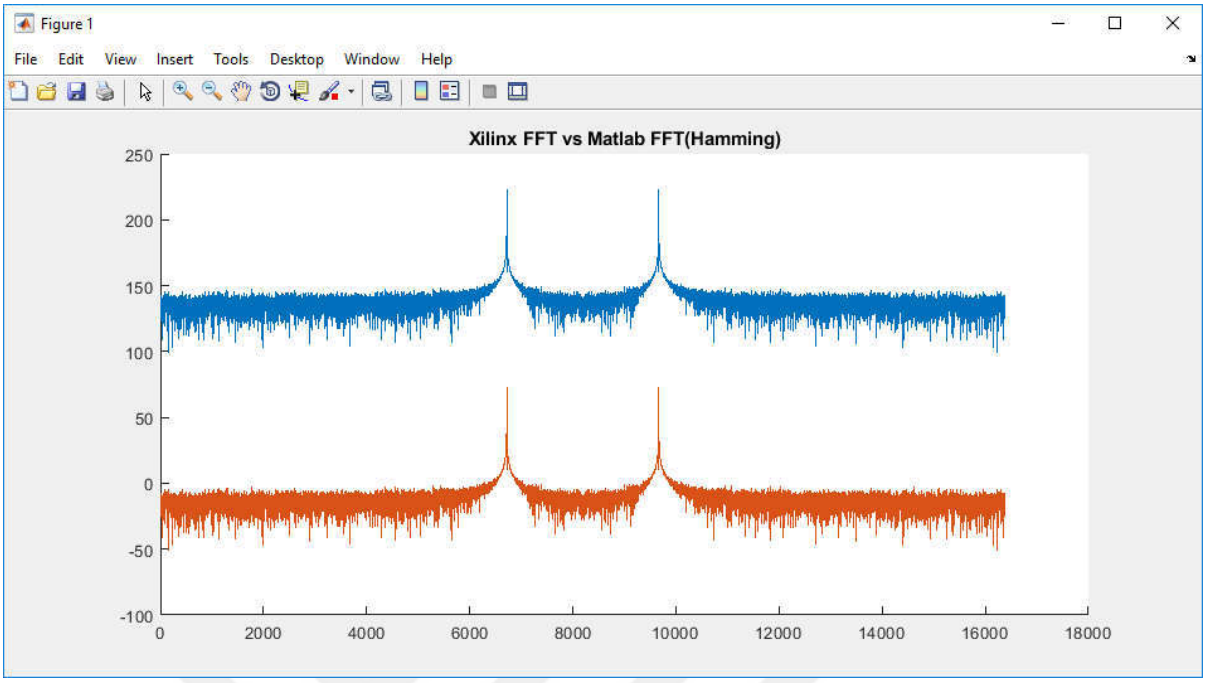
Şekil 4.8, Şekil 4.9, Şekil 4.10, Şekil 4.11 ve Şekil 4.12 'de IP bloğu tarafından pencereleme uygulanmış sinyallerin FFT çıktıları verilmiştir. Pencereleme uygulanmış FFT çıktıları ile Matlab çıktıları kıyaslanmıştır. Çıktıların tamamı incelendiğinde pencereleme fonksiyonlarının çalışması istenen doğrulukta çalıştığı tespit edilerek IP blok tasarımları projeye dâhil edilmiştir.



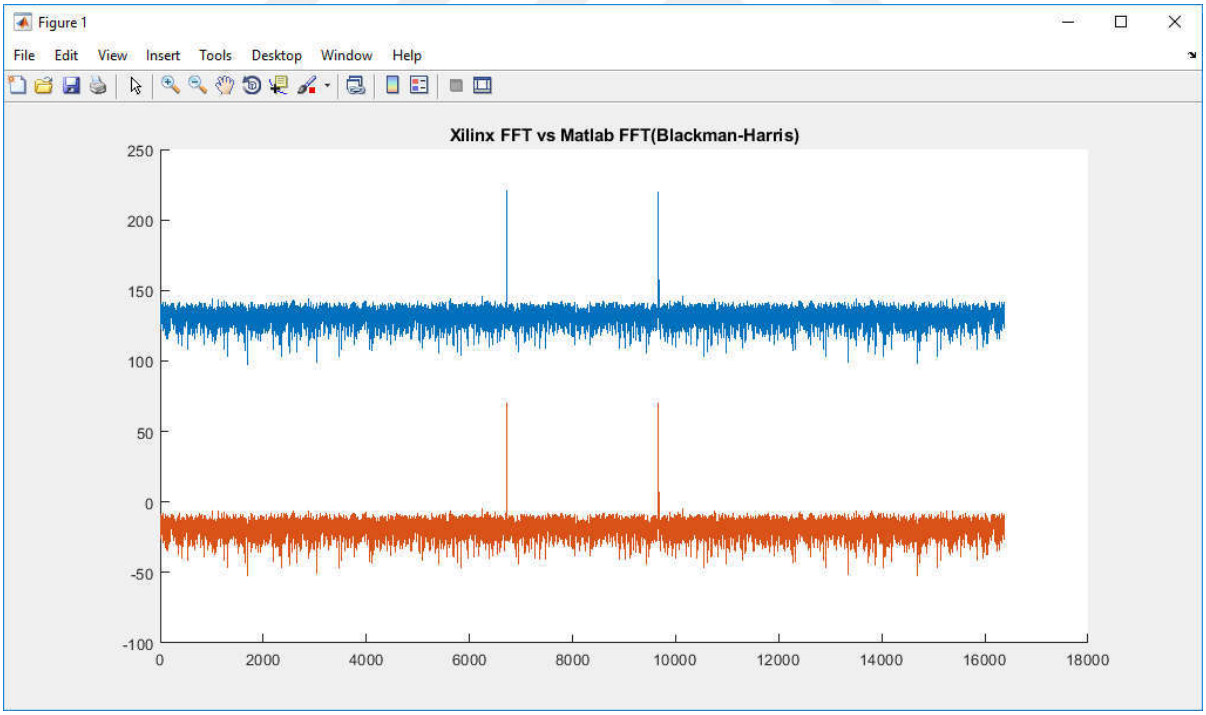
Şekil 4.8: Rectengular Pencereleme uygulanmış sinyalin FFT sonucu



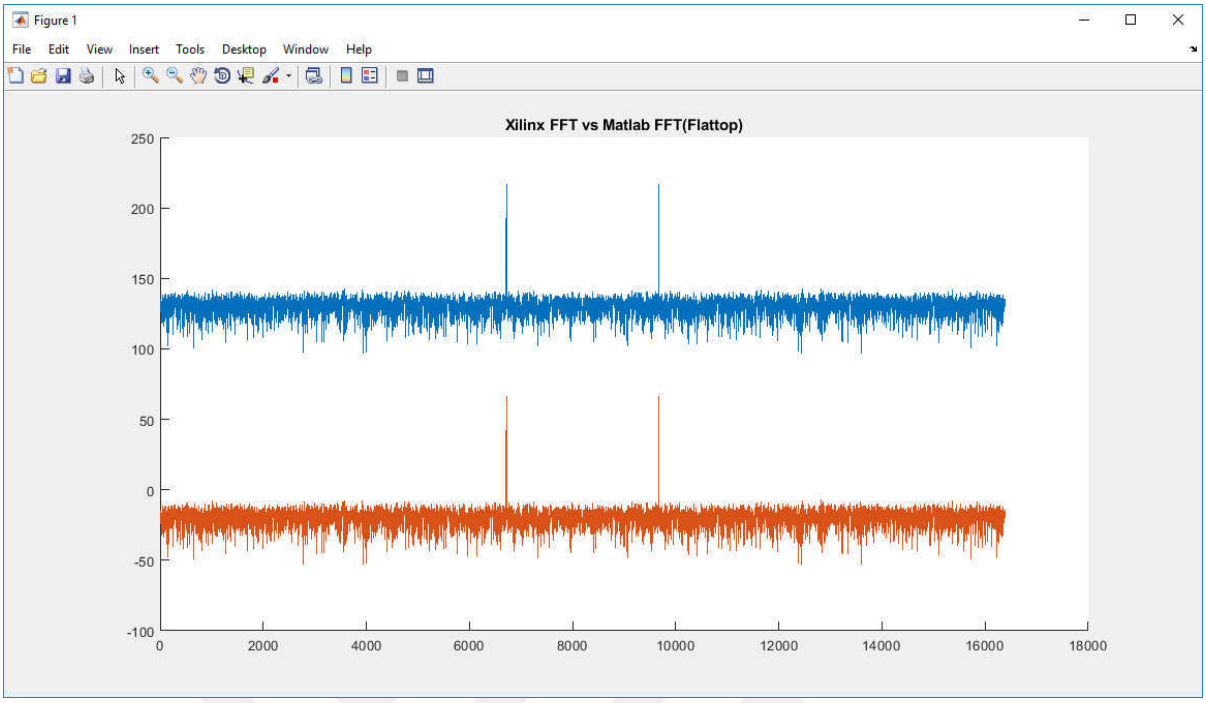
Şekil 4.9: Hann Pencereleme uygulanmış sinyalin FFT sonucu



Şekil 4.10: Hamming Pencereleme uygulanmış sinyalin FFT sonucu



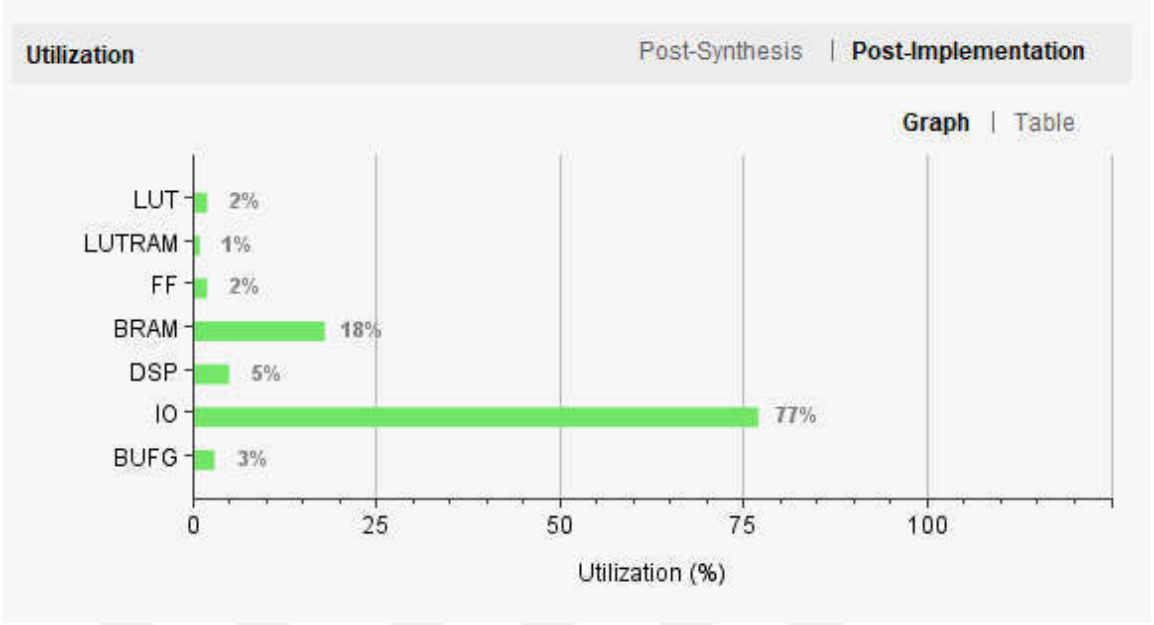
Şekil 4.11: Blackman-Harris Pencereleme uygulanmış sinyalin FFT sonucu



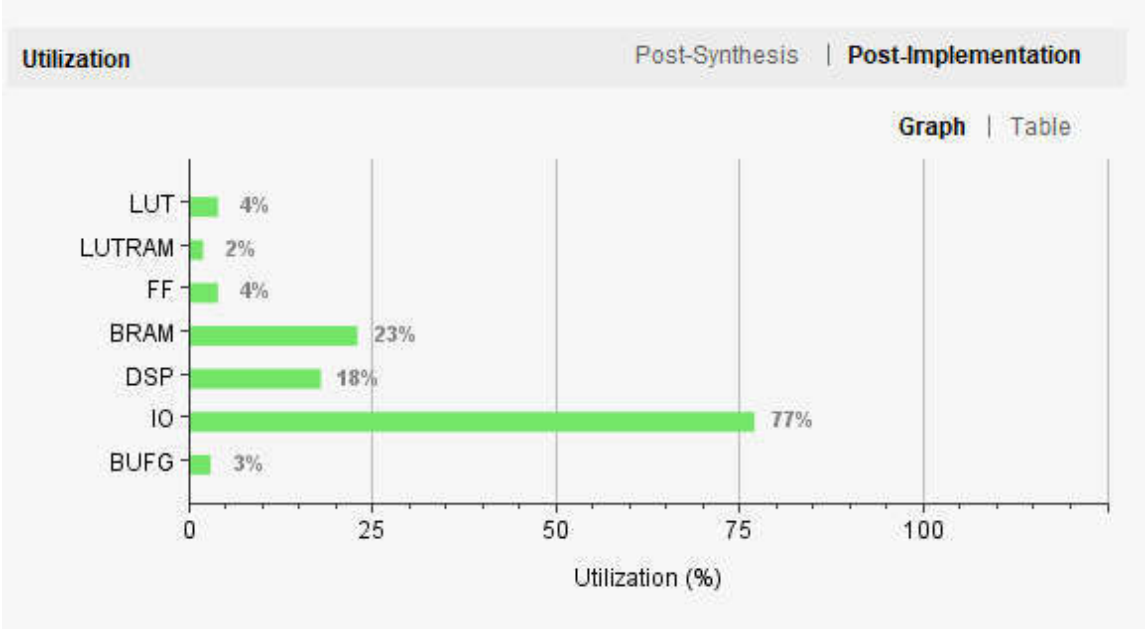
Şekil 4.12: Flattop Pencereleme uygulanmış sinyalin FFT sonucu

4.4 FFT IP Blok Değerlendirmesi

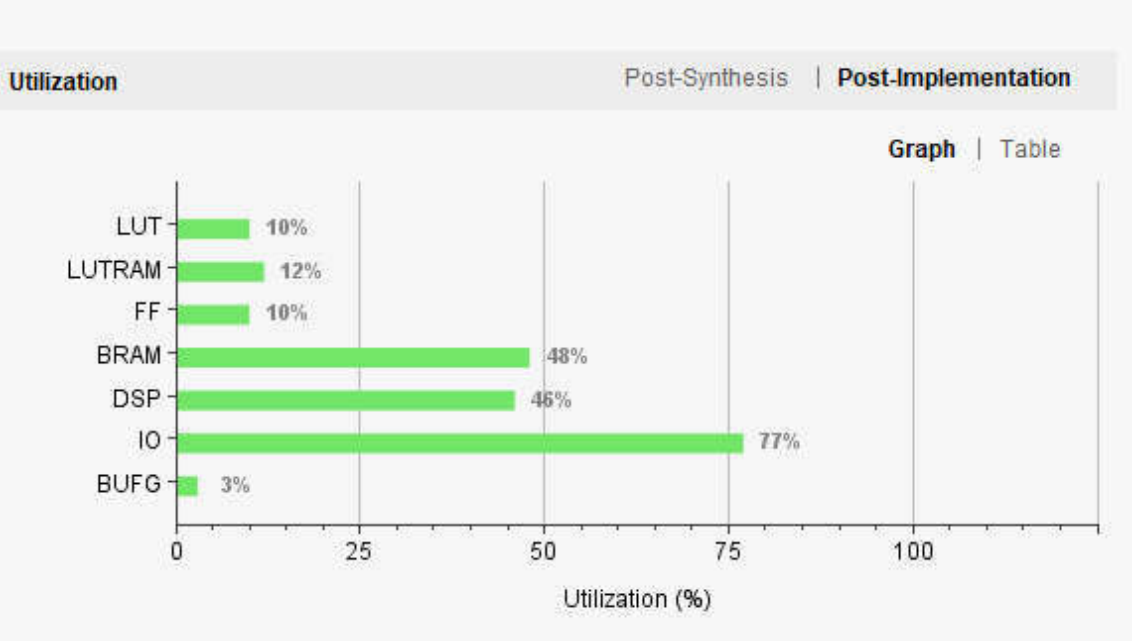
FFT IP blok tasarımı gerçekleştirilirken performansın yanı sıra FPGA üzerinde kullandığı kaynaklarda dikkate alındı. Çünkü FFT IP bloğunun 64 ten başlayarak 8192 kadar ikinin katları olacak şekilde değiştirilebilir FFT boyutuna sahip olması gerekmekteydi. FFT IP bloğunun değiştirilebilir FFT boyutuna sahip olması FPGA üzerindeki kaynak kullanımını önemli ölçüde artırmaktadır. FFT IP bloğunun kaynak kullanımını etkileyen en önemli şey ise kullanılan FFT mimarisidir. Xilinx FFT IP bloğu özelleştirilebilir bir yapıya sahiptir ve dört farklı mimari seçeneği sunmaktadır. Xilinx FFT IP bloğunun sunduğu mimari seçenekleri Pipelined, Radix-4, Radix-2 ve Radix-2 Lite'dır. FFT mimarilerinin istenilen ayarlar ile FPGA üzerinde tasarlandığında kullandıkları FGPA kaynakları Şekil 4.13, Şekil 4.14 ve Şekil 4.15 'te verilmiştir. Şekillerde ki en önemli kaynaklar blok bellekler (BRAM) ve sinyal işleme bloklarıdır (DSP). Blok bellek ve sinyal işleme blokları açısından en avantajlısının Radix-2 mimarisi olduğu görülmüştür. Şekil 4.16, Şekil 4.17 ve Şekil 4.18 'de Radix-2, Radix-4 ve Pipelined mimarisi ile FPGA üzerinde tasarlanan FFT IP bloklarının gecikme değerleri verilmiştir. Radix-4 mimarisi Pipelined mimarisine kıyasla daha az kaynak kullanmasına rağmen yakın gecikme değerleri verdiği görülmüştür. Daha az kaynak kullanmasından ve gecikme değerlerinin yakın olmasından dolayı Radix-4 mimarisi kullanılmıştır.



Şekil 4.13: Radix-2 Mimarisi kaynak kullanımı



Şekil 4.14: Radix-4 Mimarisi Kaynak Kullanımı



Şekil 4.15: Pipelined Mimarisi kaynak kullanımı

IP Symbol | Implementation Details | **Latency**

Transform Length	Transform Cycles	Latency(µs)
8	122	1.220
16	197	1.970
32	344	3.440
64	651	6.510
128	1310	13.100
256	2737	27.370
512	5828	58.280
1024	12503	125.030
2048	26858	268.580
4096	57597	575.970
8192	123152	1231.520
16384	262435	2624.350

Şekil 4.16: Radix-2 mimarisine sahip FFT IP bloğunun gecikme değerleri

IP Symbol | Implementation Details | **Latency**

Transform Length	Transform Cycles	Latency(µs)
64	266	2.660
128	499	4.990
256	883	8.830
512	1804	18.040
1024	3468	34.680
2048	7333	73.330
4096	14501	145.010
8192	30910	309.100
16384	61630	616.300

Şekil 4.17: Radix-4 mimarisine sahip FFT IP bloğunun gecikme değerleri

IP Symbol | Implementation Details | **Latency**

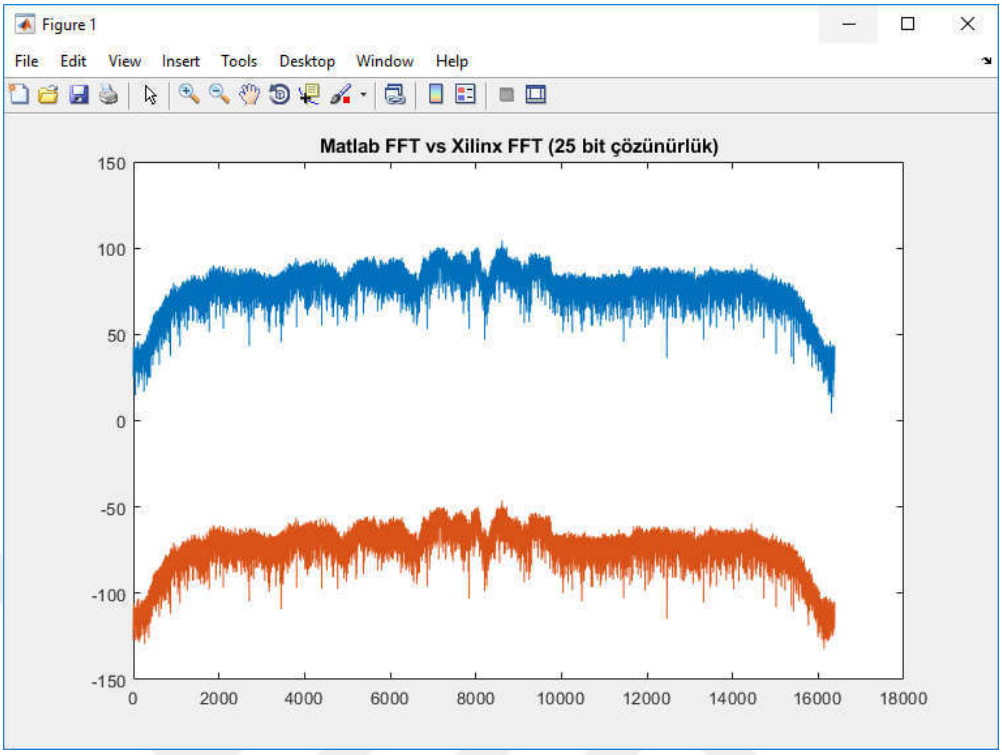
Transform Length	Transform Cycles	Latency(µs)
8	96	0.960
16	119	1.190
32	207	2.070
64	302	3.020
128	518	5.180
256	901	9.010
512	1693	16.930
1024	3228	32.280
2048	6324	63.240
4096	12467	124.670
8192	24778	247.780
16384	49353	493.530

Şekil 4.18 : Pipelined mimariye sahip FFT IP bloğunun gecikme değerleri

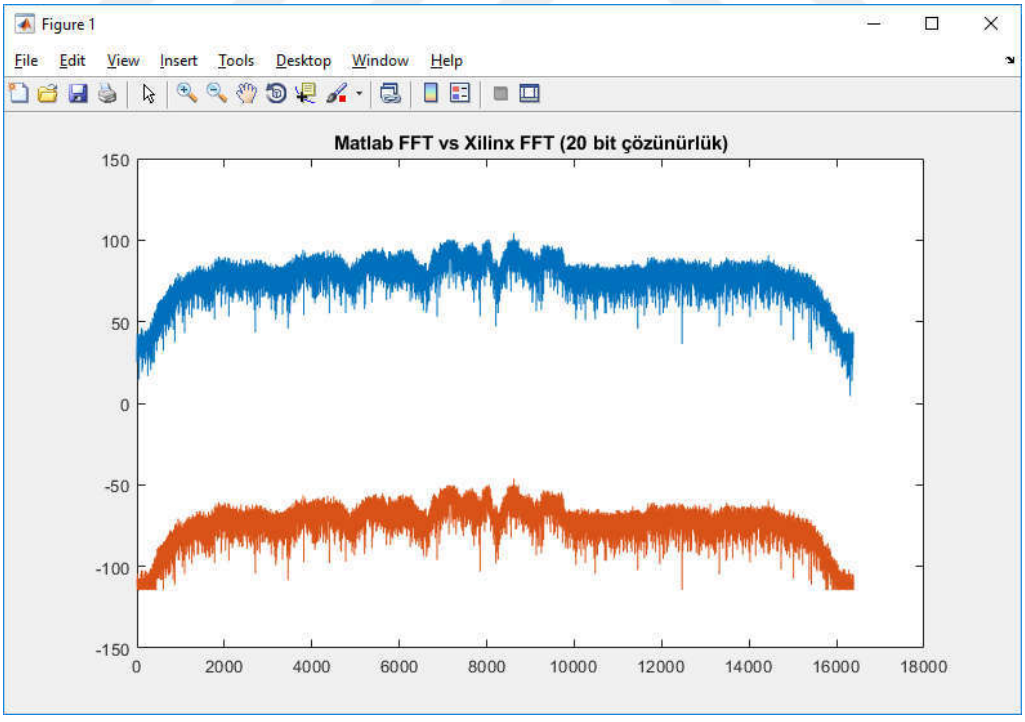
FFT IP bloğunun tasarımının derlenmesi ve geliştirme kartı üzerinde denenerek performansının incelenmesi zaman gerektiren bir yöntemdir. Bu yüzden FFT IP bloğunun tasarımı yapılırken Matlab ortamından faydalanılmıştır. Xilinx FFT IP bloğunun Matlab ortamında benzetiminin yapılmasını sağlayan C model dosyaları kullanılarak istenilen ayarlar için verdiği performansın

değerlendirilmesi yapıldı. Ayrıca FFT bloğunun C modeli ile çıkan sonuçlar kıyaslamak için Matlab FFT fonksiyonları referans olarak kullanıldı.

Xilinx tarafından sunulan FFT IP bloğunun yapılandırma seçeneklerinde FFT performansını etkileyen iki önemli parametre bulunmaktadır. Bu parametrelerden ilki, sinyallerin dinamik aralığın çözünürlüğünü belirleyen giriş verisi genişliğidir. Giriş verisinin çok yüksek olması gereksiz yere kaynak kullanımına sebep olduğu için optimize bir değer bulunması gerekmiştir. FFT IP bloğunun C modeli kullanılarak farklı giriş bant genişlikleri ile Matlab yazılımından üretilen sinyallere FFT uygulanarak Matlab FFT fonksiyonu çıktılarıyla karşılaştırılmıştır. Sinyal örneklerinin kaydedilmesinde Matlab test ortamı yazılımından faydalanılmıştır. Şekil 4.19 'da giriş bant genişliğinin 20 bit olduğu Xilinx FFT C modelinin çıktısı(altta) ile yüksek giriş bit genişliğine sahip Matlab FFT çıktısı(üstte) aynı grafik üzerinde gösterilmektedir. Giriş bant genişliğinin 20 bit olarak seçildiğinde Xilinx FFT C model çıktısının gürültü seviyesinin altında bir düzleşme görülmüştür. Gürültü seviyesinin altındaki düzleşme dinamik aralığının yetersiz olduğunu göstermektedir. Şekil 4.20 'de giriş bant genişliğinin 25 bit olduğu durumda Xilinx FFT C modelinin çıktısı gösterilmektedir. Grafik incelendiğinde 25 bit giriş bant genişliğini sahip Xilinx FFT C modeli Matlab çıktısına çok yakın bir sonuç verdiği görülmektedir. Tasarlanacak FFT IP bloğu için 25 bit giriş bant genişliğinin yeterli olduğu tespit edilmiştir.

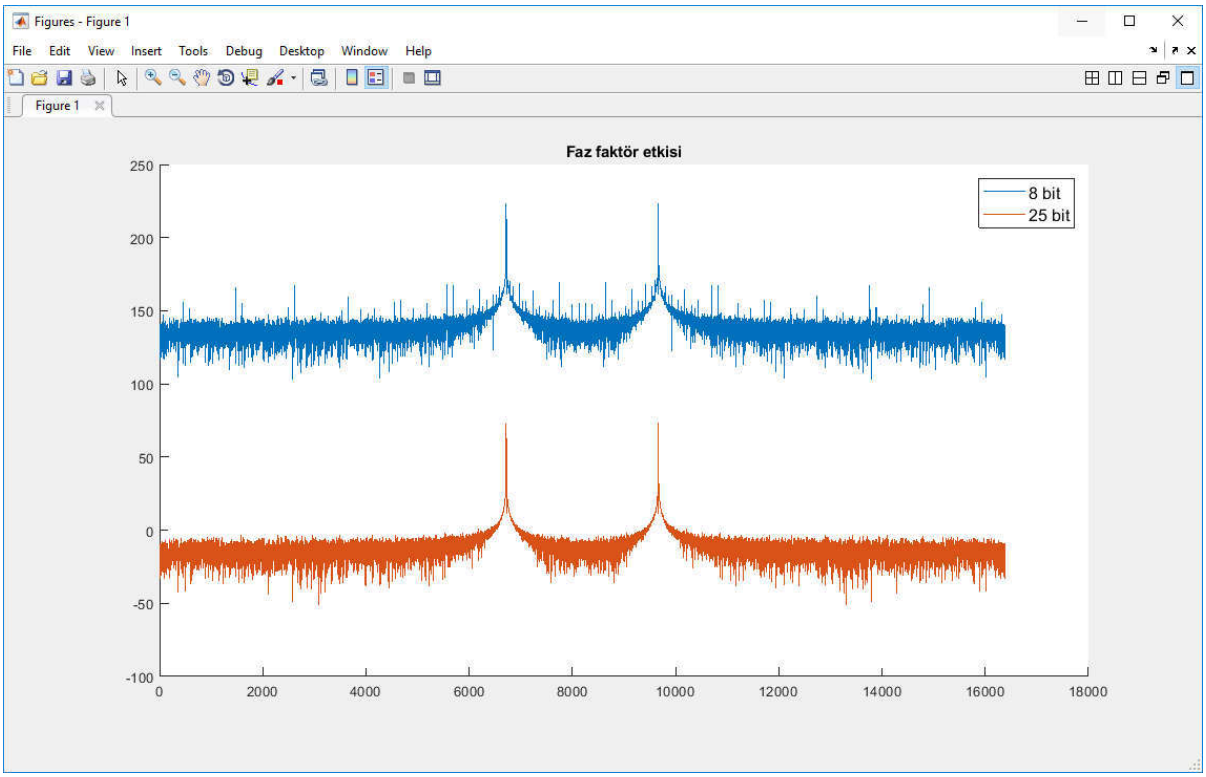


Şekil 4.19: Giriş bant genişliği (25 bit) Matlab FFT (üstte) ile Xilinx FFT (altta) kıyaslaması



Şekil 4.20: Giriş bant genişliği (20 bit) için Matlab FFT (üstte) ile Xilinx FFT (altta) kıyaslaması

FFT IP bloğunun performansı etkileyen ikinci önemli parametre faz faktörü bit genişliğidir. Şekil 4.21 'de faz faktörü bit genişliğini 25 bit ve 8 bit olduğu durumlardaki FFT C modelinin çok tonlu bir sinyale uyguladığı FFT sonuçları gösterilmektedir. Faz faktörü 8 bit olduğunda gürültüye sebep olduğu ve spektrumları bozduğu gözlemlenmiştir. Faz faktörünün 25 bit olarak ayarlandığı durumda ise yeterli performansı verdiği tespit edilmiştir.



Şekil 4.21: Faz faktör bit genişliği etkisi

Xilinx FFT IP bloğu için gerekli yapılandırma seçenekleri belirlendikten sonra FFT IP bloğu FPGA tasarımına geçilecektir. FFT IP bloğunun çıktısı ile Matlab FFT çıktısı karşılaştırıldığında C model ile tasarlanan FFT IP bloğunun istenilen performansı sağladığı görülmüştür.

[ ]

5. TARTIŞMA VE SONUÇ

Tez çalışması kapsamında, geniş bantlı RF spektrumlarının gerçek zamanlı olarak işlenmesini sağlayan FFT tabanlı bir spektrum analizör sistemi geliştirmesi yapılmıştır.

Sistemin daha yüksek frekans çözünürlüğüne sahip olması amacıyla FPGA üzerinde FIR filtre tasarımları eklenmiştir. FIR tasarımlarıyla frekans çözünürlüğü 32 kat artırılabilmektedir. Böylece geliştirilen sistem ile frekans adım aralığı düşük sinyallerin analiz edilebilmesine imkân verilmiştir.

Gerçekleştirilen sistem gerçek zamanlı olmasından dolayı sinyal verileri kayıpsız olarak alınarak işlenmektedir. Böylece çok kısa bir süreli sinyal olayları dahi kaçırılmadan işlenerek kullanıcıya aktarılabilmektedir. Sadece gerçek zamanlı sinyal analizörlerde bulunan ve kesin olarak yakalanabilmesi için anlık bir sinyalin en az ne kadar sürmesi gerektiğini ifade eden kesişme olasılığı (POI-Probability of Intercept) parametresi 2 mikro saniye olarak ölçülmüştür.

Tasarlanan sistem veri aktarımını mikroişlemci üzerinde koşan sunucu algoritması ile USB üzerinden yapabilecektir. Tasarlanan spektrum analizör cihazının veri transferini USB 2.0 bağlantısı üzerinden gerçekleştirmesi sayesinde bir bilgisayar kullanılarak kolayca kullanılabilecektir. Spektrum analiz yazılımı grafiksel gösterimlerin oluşturulmasında bilgisayarın ekran kartını kullanılacaktır. Böylece bilgisayarın işlemcisinde aşırı yük oluşması önlenecektir.

Tasarlanan FFT IP bloğunun FFT nokta sayısı 64 ten 8192 noktaya kadar 2 nin katları olacak şekilde ayarlanabilmektedir. FFT nokta sayısının ayarlanabilir olması farklı frekans çözünürlüklerinde ölçüm yapabilme imkânı sağlamıştır. Ölçülmek istenen bant genişliği ve frekans çözünürlüğü gibi parametreler örnekleme hızı, seyreltme katsayıları ve FFT boyutu gibi belirli değerlere sahip parametrelere bağlıdır. Bu yüzdende kullanıcı tarafından istenilen ölçüm parametreleri birebir olarak gerçekleştirilemez. FFT boyutunun farklı değerlere ayarlanabilmesi sayesinde cihaz üzerinde uygulanabilen ölçüm değerleri kullanıcı tarafından istenen ölçüm değerlerine olabildiğince yakın ayarlanabilmektedir. FFT IP bloğunun 8192 noktalı FFT alabilmesi ölçüm frekans çözünürlüğünü 1 Hz ‘e kadar artırmıştır.

Çalışmada tasarlanan pencereleme IP bloğu spektrum analizörlerde en çok kullanılan 5 farklı pencereleme algoritmasını gerçekleştirebilmektedir. Pencereleme yöntemleri spektrum analiz

yazılımı ara yüzü ile kullanıcı tarafından seçilebilecek şekilde tasarlanmıştır. Böylece yapılacak ölçüm için en uygun pencereleme yöntemi seçilerek spektrumların daha iyi analiz edilebilmesi sağlanmıştır.

Spektrum analiz yazılımı cihaz ölçüm parametrelerini (merkez frekansı, frekans çözünürlüğü, bant genişliği, pencereleme yöntemi) ara yüzünde bulunan yönetim paneli üzerinden kontrol edebilecektir. Spektrum analiz yazılımı spektrum gösterimi ile spektrumların frekans alanındaki değişimlerinin analiz edilebilmesine olanak sağlayacaktır.

Tez çalışması kapsamında geliştirilen spektrum analizör sistemi 25 MHz gibi geniş bir bant aralığında gerçek zamanlı olarak analiz edebilmektedir. FFT IP bloğu ve FIR bloklarında yapılan geliştirmeler ile sistem sinyal spektrumlarını 1 Hz frekans çözünürlüğü ile gösterebilmekte ve 2 mikro saniye gibi kısa bir sürede gerçekleşen anlık sinyal olaylarını kesin olarak yakalanabilmektedir. Seçilebilir pencereleme yöntemi, düşük frekans adım aralığı gibi fonksiyonel özellikler sisteme gelişmiş analiz yeteneği kazandırmıştır. Spektrum analiz yazılımının USB bağlantısına sahip olması cihazın kullanımı açısından önemli bir kolaylık sağlamaktadır. Çalışmada kullanılan geliştirme kartlarının boyutlarının ufak olması sayesinde nihai olarak ortaya çıkacak cihaz boyut olarak ufak ve taşınabilir olabilecektir. Cihazın boyutunun küçüklüğü, yüksek frekans çözünürlüğü ve geniş bantlı analiz gibi özellikleri sayesinde çok geniş bir kullanım alanına sahip olacaktır.

[ ]

6. KAYNAKLAR

[1] Türk, Ö., 2011, ARM İşlemciler İle Tek Kullanımlık Şifre Uygulamasının Gerçekleştirilmesi, Tez(Yüksek Lisans), Fırat Üniversitesi, Fen Bilimleri Enstitüsü.

[2] Elektrikport, 2013, Miladından Günümüze ARM !, https://www.elektrikport.com/teknik-kutuphane/miladindan-gunumuze-arm-!/8150#ad-image-0, [Ziyaret tarihi: 29 Ekim 2019].

[3] Cebeci, H.Y., 2012, Düşük Maliyetli, Bilgisayar Tabanlı Data Logger Özellikli Osiloskop Tasarımı ve Gerçekleştirilmesi, 66:37–39,.

[4] Taşdemir, C., 2011, Arm ailesi, https://coskuntasdemir.com/gomulu-sistemler/32-bitin-yeni-lideri-arm.html/attachment/armailesi, [Ziyaret tarihi: 29 Ekim 2019].

[5] Vikipedi, 2019, Gömülü Linux, https://tr.wikipedia.org/wiki/Gömülü_Linux,.

[6] https://wmaraci.com/nedir/gomulu-linux, [Ziyaret tarihi: 29 Ekim 2019].

[7] Vikipedi, 2018, FPGA, https://tr.wikipedia.org/wiki/FPGA, [Ziyaret tarihi: 29 Ekim 2019].

[8] Elektrikport, 2014, FPGA Nedir?, https://www.elektrikport.com/teknik-kutuphane/fpga-nedir/14702#ad-image-0, [Ziyaret tarihi: 29 Ekim 2019]..

[9] Vikipedi, 2018, Yongada sistem, https://tr.wikipedia.org/wiki/Yongada_sistem, [Ziyaret tarihi: 3 Kasım 2019].

[10] Rohde-Schwarz, 2019, Spectrum Analyzer, https://www.rohde-schwarz.com/au/product/fsu-productstartpage_63493-7993.html, [Ziyaret tarihi: 3 Kasım 2019].

[11] Adnani, A., Duplicy, J., and Philips, L., 2013, Spectrum analyzers today and tomorrow: Part 1, IEEE Instrumentation and Measurement Magazine 16(6):36–40, doi:10.1109/MIM.2013.6704970.

[12] Keysight Technologies, 2016, Spectrum Analysis Basics, Aplication Note 150, USA, ISBN: 5952-0292.

[13] http://zedboard.org/product/zedboard, [Ziyaret tarihi: 20 Kasım 2019].

[14] https://www.avnet.com/wps/portal/silica/products/product-highlights/2016/xilinx-zynq-7000/ [Ziyaret tarihi: 22 Kasım 2019].

[15] https://lists.gnu.org/archive/html/discuss-gnuradio/2018-05/msg00073.html, [Ziyaret tarihi: 20 Kasım 2019].

[16] https://www.xilinx.com/support/documentation/sw_manuals/xilinx2018_3/ug1144-petalinux-tools-reference-guide.pdf, [Ziyaret tarihi: 23 Kasım 2019].

[17] National Instruments, 2019, Understanding RF & Microwave Specifications - Part I, http://www.ni.com/tutorial/4446/en/#toc3, [Ziyaret tarihi: 3 Aralık 2019].

[18] Vikipedi, 2019, Fast Fourier transform, https://en.wikipedia.org/wiki/Fast_Fourier_transform, [Ziyaret tarihi: 6 Aralık 2019].

[19] http://www.iowahills.com/Example Code/FFTButterfly.html, [Ziyaret tarihi: 6 Aralık 2019].

[20] Brigham, E.O. and Morrow, R.E., 1967, The fast Fourier transform, IEEE Spectrum 4(12):63–70, doi:10.1109/MSPEC.1967.5217220.

[21] https://www.xilinx.com/support/documentation/ip_documentation/xfft/v9_0/pg109-xfft.pdf, [Ziyaret tarihi: 16 Aralık 2019]

[22] National Instruments, 2015, Understanding FFTs and Windowing Overview, White Paper.

[23] Vikipedi, 2019, Window function, https://en.wikipedia.org/wiki/Window_function, [Ziyaret tarihi: 9 Aralık 2019].

[24] Ramian, F., 2015, Implementation of Real-Time Spectrum Analysis, White Paper, 35.

[25] Tektronix, Inc., 2009, Understanding FFT Overlap Processing Fundamentals, USA, 1-24.

]

7. ÖZGEÇMİŞ

Kişisel Bilgiler	
Adı Soyadı	Mehmet Eskin
Doğum Yeri	Aksaray
Doğum Tarihi	28.12.1989
Uyruğu	☑ T.C. ❒ Diğer:
Telefon	0537 286 8534
E-Posta Adresi	mehmet.eskin@ogr.iu.edu.tr
Web Adresi	



Eğitim Bilgileri	
Lisans	
Üniversite	Fatih Üniversitesi
Fakülte	Mühendislik Fakültesi
Bölümü	Elektrik Elektronik Mühendisliği
Mezuniyet Yılı	2015

Yüksek Lisans	
Üniversite	İstanbul Üniversitesi-Cerrahpaşa
Enstitü Adı	Lisansüstü Eğitim Enstitüsü
Anabilim Dalı	Elektrik-Elektronik Mühendisliği Anabilim Dalı
Programı	Elektrik-Elektronik Mühendisliği

Makale ve Bildiriler