



**FOURIER BÖLGESİNDE ÇAPRAZ İLİNTİ YÖNTEMİ İLE SES
KAYNAĞININ KONUMUNUN FPGA KULLANARAK TESPİTİ**

Merve ÖZTÜRK LAFCI

**YÜKSEK LİSANS TEZİ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANA BİLİM DALI**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

TEMMUZ 2019

Merve ÖZTÜRK LAFCI tarafından hazırlanan “FOURIER BÖLGESİNDE ÇAPRAZ İLİNTİ YÖNTEMİ İLE SES KAYNAĞININ KONUMUNUN FPGA KULLANARAK TESPİTİ” adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Gazi Üniversitesi Elektrik Elektronik Mühendisliği Ana Bilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Danışman: Doç. Dr. Hasan Şakir BİLGE

Elektrik Elektronik Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi
Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum.

.....

Başkan: Prof. Dr. Şerafettin EREL

Elektrik Elektronik Mühendisliği Ana Bilim Dalı, Ankara Yıldırım Beyazıt
Üniversitesi
Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum.

.....

Üye: Doç. Dr. Ertuğrul AKSOY

Elektrik Elektronik Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi
Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum.

.....

Tez Savunma Tarihi: 03/07/2019

Jüri tarafından kabul edilen bu tezin Yüksek Lisans Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

.....
Prof. Dr. Sena YAŞYERLİ

Fen Bilimleri Enstitüsü Müdürü

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Merve ÖZTÜRK LAFCI

03/07/2019

FOURIER BÖLGESİNDE ÇAPRAZ İLİNTİ YÖNTEMİ İLE SES KAYNAĞININ KONUMUNUN FPGA KULLANARAK TESPİTİ

(Yüksek Lisans Tezi)

Merve ÖZTÜRK LAFCI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Temmuz 2019

ÖZET

Akustik kaynak konumlandırma ile ilgili, robot uygulamaları, navigasyon sistemleri, konferans uygulamaları, işitme cihazları ve ayrıca askeri uygulamalara yönelik çalışmalar son yıllarda oldukça artmıştır. Yüksek kapasiteli verilerin gerçek zamanlı olarak işlenebilmesi için gelişmiş donanım yapılarına ihtiyaç duyulmaktadır. Günümüzde ihtiyaç duyulan donanım yapısı, Alanda Programlanabilir Kapı Dizileri (FPGA) kullanılarak tasarlanabilmektedir. FPGA hızları, kapasiteleri ve tasarım araçları son yıllarda iyileştirilmiştir. Bu sayede yüksek kapasite veri işleyebilen donanımlar düşük maliyetler ile tasarlanıp üretilebilmektedir. Bu çalışmada konumları bilinen mikrofon dizisi kullanıldığı durumda akustik ses kaynağının konumunun açısal olarak tespit edilmesi üzerine çalışmıştır. Mikrofon dizileri ile ses kaynağının konumunun gerçek zamanlı olarak belirlenmesi amacıyla farklı yöntemler geliştirilmiştir. Bu tezde, akustik kaynak konumlandırmada yaygın olarak kullanılan ve hassasiyeti yüksek olan TDOA (Time Difference of Arrival) tabanlı akustik kaynak konumlandırmaya yönelik bir çalışma yapılmıştır. İki mikrofon kullanılan bir durumda akustik kaynağın konumunun tespit edilebilmesi için bir donanım mimarisi tasarlanmıştır. Sinyaller arasındaki gecikmenin hesaplanması için çapraz ilinti temelli bir yöntem sunulmaktadır. Yöntem Verilog donanım tanımlama dili kullanılarak Modelsim simülasyon ortamında test edilmiştir. Çapraz ilinti yöntemi donanım üzerinde daha az yer kaplayacak şekilde sadeleştirilmiş ve bu sadeleştirmenin tespit sonucunu etkilemediği gözlemlenmiştir. Yapılan çalışmalarda bu yöntem sayesinde çok az bir hatayla iki sayısal sinyal arasındaki gecikmenin hesaplandığı ve akustik kaynakların konumunun açısal olarak tespit edilebildiği görülmüştür.

Bilim Kodu : 90520
Anahtar Kelimeler : Çapraz İlinti, Ses Konum Tespiti, FPGA
Sayfa Adedi : 75
Danışman : Doç. Dr. Hasan ŞAKİR BİLGE

SOUND SOURCE LOCALIZATION IN THE FOURIER REGION WITH CROSS
CORRELATION METHOD USING FPGA

(M. Sc. Thesis)

Merve ÖZTÜRK LAFCI

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

July 2019

ABSTRACT

Research on sound source localization, robot selection, navigation systems, conference settings, hearing aids, and central military applications has increased considerably in recent years. Advanced hardware structures are needed for real-time processing of high-capacity data. Today, the required hardware structure can be designed using Field Programmable Gate Arrays. FPGA speeds, capacities and design tools have been improved in recent years. In this way, high capacity data processing equipment can be designed and produced with low costs. In this study, it has been worked on angular determination of the position of the acoustic sound source where known positions of the microphone array are used. Different methods have been developed to determine the sound source in real time in microphone arrays. In this thesis, a study has been done for positioning of acoustical sound source based on TDOA (Time Difference of Arrival) which is widely used in acoustic sound source positioning. In the case of using two microphones, a hardware architecture is designed to determine the location of the acoustic source. A cross-correlation based method is used to calculate the delay between signals. Method has been tested in Modelsim simulation environment using Verilog the hardware definition language. The cross-correlation method was simplified to take up less space on the hardware and it was observed that this simplification did not affect the detection result. In this method, it is seen that the delay between two digital signals is calculated with a very little error and the position of the acoustic sources can be determined angularly.

Science Code : 90520

Key Words : Cross Correlation, Sound Source Location, FPGA

Page Number : 75

Supervisor : Assoc. Prof. Dr. Hasan Şakir BİLGE

TEŞEKKÜR

Tez çalışmam sırasında kıymetli bilgi, birikim ve tecrübeleri ile bana yol gösterici ve destek olan değerli danışman hocam Doç. Dr. Hasan Şakir BİLGE 'ye, bana yaşamım boyu katkılar sağlayan tüm hocalarıma, bu güne kadar bana hem maddi hem manevi katkıda bulunan biricik aileme ve son olarak çalışma süresince tüm zorlukları benimle göğüsleyen ve hayatımın her evresinde bana destek olan değerli eşim Mehmet LAFCI 'ya ve değerli evlatlarım Sümeyye LAFCI ve Zeynep LAFCI 'ya sonsuz teşekkür ve saygılarımı sunarım.



İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ.....	x
ŞEKİLLERİN LİSTESİ.....	xv
RESİMLERİN LİSTESİ	xviii
SİMGELER VE KISALTMALAR.....	xix
1. GİRİŞ.....	1
2. TEORİK BİLGİLER	3
2.1. Ses	3
2.2. Mikrofon	3
2.2.1. Dinamik mikrofonlar.....	5
2.2.2. Kapasitif (Kondansatör) mikrofonlar	5
2.2.3. Şeritli mikrofonlar	6
2.2.4. Piezoelektrik kristalli mikrofonlar	6
2.2.5. Elektret mikrofonlar.....	7
2.2.6. Karbon tozlu mikrofonlar.....	7
2.3. Mikrofon Dizileri.....	8
3. DAHA ÖNCE YAPILMIŞ ÇALIŞMALAR.....	11
4. SES KAYNAĞININ YERİNİN BULUNMASI.....	19
4.1. Çapraz İlinti Yönteminin Sadeleştirilmesi.....	22

	Sayfa
5. ALAN PROGRAMLANABİLİR KAPI DİZİLERİ (FPGA).....	27
5.1. FPGA ile Tasarım.....	29
5.2. FPGA Türleri.....	30
5.2.1. SRAM tabanlı.....	30
5.2.2. Anti sigorta tabanlı.....	31
5.2.3. EEPROM / FLASH tabanlı.....	32
5.3. FPGA İç Yapısı	33
5.3.1. Mantık hücresi (Logic-Cell).....	33
5.3.2. Matris anahtarlama yönlendirme yolları.....	34
5.3.3. Giriş/Çıkış blokları.....	34
5.3.4. Clock ve global lines.....	34
5.3.5. RAM blokları	35
5.4. Programlama ve Tasarım	35
5.4.1. FPGA tasarım akış şeması	38
5.5. İkili Sayı Sistemleri	39
5.5.1. Sabit noktalı sayı sistemleri.....	39
5.5.2. Kayan noktalı sayı sistemleri.....	40
6. VERİLOG DONANIM TANIMLAMA DİLİ.....	43
6.1. Modüller (Modules).....	44
7. DONANIM MİMARİSİ.....	47
7.1. HFD ve Ters HFD.....	47
7.1.1. FFT megacore projeye eklenmesi	47
7.1.2. FFT megacore simulasyon ayarları	53
7.1.3. Simulasyon sonuçlarının değerlendirilmesi	56

	Sayfa
7.2. Çapraz İlinti.....	58
7.3. Maksimum Noktanın Bulunması	59
7.4. Zaman ve Mesafe Hesapları.....	59
8. DENEYSEL SONUÇLAR.....	63
9. SONUÇ VE ÖNERİLER.....	71
KAYNAKLAR	73
ÖZGEÇMİŞ	75

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 3.1. Ses kaynağının yerini belirlemek için MUSIC ve geliştirilmiş MUSIC algoritmasının karşılaştırılması.....	14
Çizelge 4.1. Çapraz ilinti yöntemi ve sadeleştirilmiş çapraz ilinti yöntemi ile gecikme hesaplaması sonuçları.....	25
Çizelge 5.1. Üretici firmalar ve tercih ettikleri üretim teknolojileri	30
Çizelge 5.2. Programlama teknolojilerinin karşılaştırılması	32
Çizelge 7.1. FFT megacore sinyalleri ve açıklamaları.....	52
Çizelge 8.1. Pencere boyutu 50000, 16384, 8192, 1024, 512 ve 256 örnek içeren ses verileri ile sadeleştirilmiş çapraz ilinti yöntemi kullanılarak gecikme hesaplamaları	65
Çizelge 8.2. Pencere boyutu 50000, 16384, 8192, 1024, 512 ve 256 örnek içeren ses verileri ile sadeleştirilmiş çapraz ilinti yöntemi kullanılarak açı hesaplamaları	65
Çizelge 8.3. Ses işaretleri arasındaki hesaplanan gecikmenin MATLAB ve MODELSİM den elde edilen sonuçları	69
Çizelge 8.4. Ses kaynaklarının konumunun açısal olarak MATLAB ve MODELSİM den elde edilen sonuçları	69

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. Her yönlü mikrofon	4
Şekil 2.2. Kalp biçimli mikrofon	4
Şekil 2.3. Dinamik mikrofonun yapısı	5
Şekil 2.4. Kapasitif mikrofonun yapısı	5
Şekil 2.5. Şeritli mikrofonun yapısı	6
Şekil 2.6. Piezoelektrik mikrofonun yapısı.....	6
Şekil 2.7. Elektret mikrofonun yapısı	7
Şekil 2.8. Karbon tozlu mikrofonun yapısı	8
Şekil 3.1. Çapraz mikrofon dizisinin sinyal iletim modeli	13
Şekil 3.2. Akustik sinyaller	15
Şekil 3.3. Mikrofon dizisinin çıkış sinyali ve spektrumu	15
Şekil 3.4. Dizinin uzaysal güç spektrumu.....	16
Şekil 3.5. (a) ve (b) Ana hoparlöre işaret eden dizinin önceki ve sonraki çıkış sinyallerinin karşılaştırılması	17
Şekil 4.1. Ses kaynağının yayılımı ve mikrofonlara ulaşma süreleri arasındaki farklılık.....	19
Şekil 4.2. İki mikrofondan alınan işaretle zaman farkı	20
Şekil 4.3. İki mikrofondan alınan ses işareti ve çapraz ilinti sonucu.....	21
Şekil 4.4. Ses kaynağının mikrofonlara gelişi ve mikrofonlara göre ses kaynağının açısal kestiriminin gösterimi	22
Şekil 4.5. İki işaret arasındaki gecikmenin çapraz ilinti fonksiyonu ile hesaplanması	23
Şekil 4.6. İki işaret arasındaki gecikmenin sadeleştirilmiş çapraz ilinti fonksiyonu ile hesaplanması	24

Şekil	Sayfa
Şekil 5.1. FPGA temel yapısı.....	28
Şekil 5.2. 2010 yılı FPGA üreticileri ve pazar payları.....	30
Şekil 5.3. Temel Mantık Bloğu İç Yapısı	33
Şekil 5.4. Yarım Toplayıcı (Half Adder)	35
Şekil 5.5. Half Adder'ın Şematik Tasarımı	36
Şekil 5.6. Half Adder'ın RTL Şeması.....	37
Şekil 5.7. FPGA Tasarım Akış Şeması.....	38
Şekil 5.8. Kayan noktalı sayı formatı bit gösterimi	41
Şekil 7.1. Ses konum tespiti donanımının temel mimarisi.....	47
Şekil 7.2. Kullanılan HFD öbeğinin sinyalleşme ara yüzü.....	52
Şekil 7.3. FFT Streaming Veri Akışı Mimarisi Simülasyon Dalga Formu	57
Şekil 7.4. Modelsim Simulasyon Sonuçları.....	57
Şekil 7.5. Kullanılan çapraz ilinti öbeğinin sinyalleşme ara yüzü.....	58
Şekil 7.6. İlinti öbeğinin mimarisi	58
Şekil 7.7. Maksimum bulma öbeğinin mimarisi.....	59
Şekil 7.8. Uzaklık ve Açık Bulma Öbeğinin Mimarisi.....	60
Şekil 7.9. Ses Konum Tespiti Donanım Mimarisi	60
Şekil 7.10. Ses Konum Tespiti Donanım Mimarisinin RTL Şeması.....	61
Şekil 8.1. Pencere boyutu 50000, 16384, 8192, 1024, 512 ve 256 örnek içeren ses verileri	64
Şekil 8.2. Ses kayıt örneği.....	66
Şekil 8.3. s1 işaretinin Fourier dönüşümünün MATLAB ve MODELSİM karşılaştırması	66
Şekil 8.4. s2 işaretinin Fourier dönüşümünün MATLAB ve MODELSİM karşılaştırması	67

Şekil	Sayfa
Şekil 8.5. Sadeleştirilmiş çapraz ilinti yönteminin MATLAB ve MODELSİM karşılaştırması	67
Şekil 8.6. Ters Fourier işleminin MATLAB ve MODELSİM karşılaştırması	68



RESİMLERİN LİSTESİ

Resim	Sayfa
Resim 2.1. MIT laboratuvarlarında geliştirilen 1020 mikrofona düzlemsel mikrofona.....	9
Resim 6.1. Kullanılan derleyici Altera Quartus 13.1 Web Edition	45
Resim 7.1. MegaWizard Plug-In Manager'a ait ekran görüntüsü.....	48
Resim 7.2. FFT Megacore ayarlarının yapılması.....	48
Resim 7.3. FFT Megacore parametre ayarları	49
Resim 7.4. FFT Megacore mimari ayarları.....	50
Resim 7.5. FFT Megacore Simülasyon Ayarları	51
Resim 7.6. FFT Megacore raporu	51
Resim 7.7. Eda Tool Simulation ayarları.....	54
Resim 7.8. Test bench ve simülasyon dosyalarının eklenmesi.....	55
Resim 7.9. Eda Tool Seçenekleri Modelsim-Altera dosya yolu.....	55
Resim 7.10. Test bench içinde kullanılan FFT giriş dosyalarının simülasyon klasörüne kopyalanması.....	56
Resim 8.1. Donanım kaynak tüketim tablosu	70

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler

Açıklamalar

cm	Santimetre
hz	Hertz
kb	Kilobayt
khz	Kilohertz
µs	Mikro saniye

Kısaltmalar

Açıklamalar

AFD	Ayrık Fourier Dönüşümü
APKD	Alan Programlanabilir Kapı Dizisi
ASIC	Application Specific Integrated Circuit
Clk	Clock
CMOS	Complementary Metal Oxide Semiconductor
CORDIC	COordinate Rotation DIgital Computer
CPLD	Complex Programmable Logic Device
Dm	2 mikrofon arasındaki uzaklık miktarı
DOA	Direction of Arrival
EEPROM	Electronically Erasable Programmable Read-Only Memory
FFT	Fast Fourier Transform
FPGA	Field Programmable Gate Array
Fs	Örnekleme Frekansı
HDL	Hardware Description Language
HFD	Hızlı Fourier Dönüştürücü
I/O	Giriş/Çıkış
ITD	Gecikme miktarının zaman düzleminde karşılığı
LSB	Least Significant Bit
LUT	Look Up Table

Kısaltmalar**Açıklamalar**

m	Gecikme miktarı
maks_i	Maksimum tepenin oluştuğu yer
MIT	Massachusetts Institute of Technology
MSB	Most Significant Bit
MUSIC	MUltiple SIgnal Classification
MUX	Multiplexer
PLD	Programmable Logic Device
PROM	Programmable Read-Only Memory
RAM	Random Access Memory
RTL	Register Transfer Level
SNR	Signal to Noise Ratio
SPLD	Simple Programmable Logic Device
SRAM	Static Random Access Memory
STFT	Short-Time Fourier Transform
TDOA	Time Difference of Arrival
THFD	Ters Hızlı Fourier Dönüştürücü
USB	Universal Serial Bus
V_ses	Ses hızı

1. GİRİŞ

Ses konum tespiti, ses kaynağının nerede bulunduğunu belirlemek için yapılan çalışmalardır. Ses kaynağının konumunun belirlenmesi için birbirleri arasında belirli mesafelerle yerleştirilmiş mikrofon dizileri kullanılmaktadır.

Ses konum tespitinin geliştirilmesinde memelilerin sesin konumunu bulmasında kullandığı yöntemler örnek alınmıştır. Memelilerde işitme sistemi ses kaynağının konumunu iki kulağa gelen sinyaller arasındaki zaman farkından faydalanarak kestirir. Ses kaynağının hareketine göre kulaklara gelen seslerin şiddeti ve fazı değişecektir ve bu sayede ses kaynağının yönü ve uzaklığını algılanacaktır. Memelilerde konum kestirimi yatay düzlemde $\pm 1^\circ$ sapma ile hesaplanmaktadır [1].

Üç boyutlu uzayda ilerleyen ses dalgaları küresel yörüngeli bir yayılım izlemektedir. Bu küresel yayılımının bir sonucu olarak, nokta şeklindeki bir yapıdan başlayarak giderek büyüyen ve büyüdükçe zayıflayan bir yapı ses işaretlerini ifade edebilmek için sıkça kullanılmaktadır. Ses dalgaları dairesel bir şekilde yayıldığından dolayı, aynı ses kaynağından çıkan işaretlerin farklı ses işareti alıcılarına farklı zaman dilimlerinde ulaşmaktadır. Bu bilgilere göre birden fazla ses alıcısı kullanarak ses kaynağının konumunun tespit edilmesi mümkün olmaktadır.

Çalışmanın ilk bölümünde sesin yayılımı ve ses işaretlerinin elektriksel sinyallere dönüştürülmesi konusunda teorik bilgiler verilmekte olup ikinci bölümde daha önce yapılmış çalışmalara yer verilmiştir. Tezin üçüncü bölümünde ses kaynağının yerinin bulunması ile ilgili algoritma açıklanmıştır. Daha sonra tezin dördüncü bölümünde FPGA ile ilgili genel bilgiler verilmekte olup beşinci bölümde ise Verilog donanım tanımlama dili anlatılmıştır. Tezin altıncı bölümünde geliştirilmiş olan donanım mimarisine yer verilmiş olup tezin son iki bölümünde deneysel sonuçlar ile sonuç ve öneriler bölümlerine değinilmiştir.

2. TEORİK BİLGİLER

Bu bölümde sesin yayılımı ve ses işaretlerinin elektriksel sinyallere dönüştürülmesi konusunda bilgi verilmektedir.

2.1. Ses

Ses fiziksel bir olaydır ve bir objenin (cismin) titreşimleriyle oluşan bir enerji türüdür. Sesin bir frekansı, şiddeti, boyu ve hızı bulunmaktadır. Bir cismin ses çıkarması için titreşim yapması gerekir. Sesler insanın kulağının duyduğu ve insan kulağının duymadığı sesler olmak üzere iki gruba ayrılmaktadır.

Sesin doğması için titreşim hareketi yetmez aynı zamanda titreşen cismin sesi iletecek bir ortamda bulunması gerekmektedir. Söz konusu ortam katı, sıvı ya da gaz olabilir. Ancak sesi iletebilecek nitelikte, yani esnek bir ortam olması gerekmektedir. Ses hızı frekansa bağlı olarak değişmez. Ses hızı her frekansta aynıdır. Sesin yayılma hızı havanın sıcaklığı, yoğunluğu gibi durumlara göre değişmektedir.

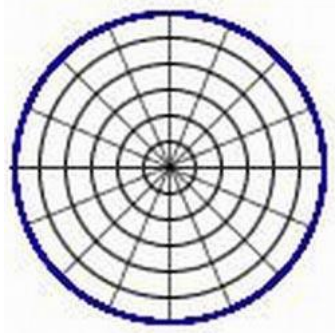
Sesin yayılma hızı, hareketli bir ses kaynağının hızından daha az olduğu zaman, ses, patlama sesi olarak duyulur. Bu durumda ses işareti şok dalgası olarak isimlendirilir ve ışın gibi konik bir alana yayılmaktadır.

Frekansları 20 Hz den az olan ses işaretleri ses beri ya da infra sound olarak nitelendirilir. Bu ses dalgasının frekansı insan kulağının duyamayacağı kadar düşüktür. Teorik olarak insan kulağının 20 Hz ile 20000 Hz arasını duyduğu bilinmektedir. Frekansı 20000Hz in üstündeki yüksek frekanslı sesler ses öte ya da ultra sound olarak adlandırılır.

2.2. Mikrofon

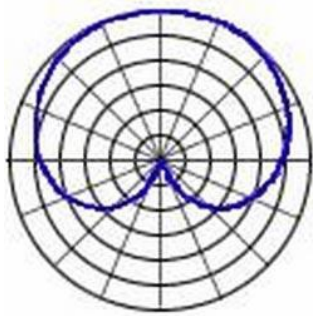
Ses işaretlerini elektriksel sinyallere dönüştüren elektro akustik cihazlara mikrofon denir. Bir sesin saptanması yani depolanması için bu cihazların kullanılması gereklidir. Ses sinyalleri ve elektrik sinyalleri arasında dönüşüm yaptıkları için algılayıcı (transduser) olarak tanımlanabilirler. Algı tiplerine göre mikrofonlar her yönlü (omnidirectional), bir yönlü (unidirectional), çift yönlü (bidirectional) gibi sınıflandırılmaktadır.

Her yönden gelen sesi eşit olarak algılayan mikrofonlara her yönlü (omnidirectional) mikrofonlar denir. Ses kaynağının bariz şekilde belli olmadığı sesin birçok yönden geldiği durumlarda ya da mikrofonun sabit, kaynağın hareketli olduğu durumlarda bu mikrofonlar tercih edilir.



Şekil 2.1. Her yönlü mikrofon

Cardioid algı tipinde ses en çok diyaframın karşısından geldiğinde mikrofon tarafından algılanır, yanlara doğru gidildikçe azalır. Ortam seslerinden çok, belirli bir yönden gelen sesin mikrofon tarafından algılanmak istendiğinde ama yanlardan ve arkadan gelen seslerin belirli düzeyde algılandığı tiptir.



Şekil 2.2. Kalp biçimli mikrofon

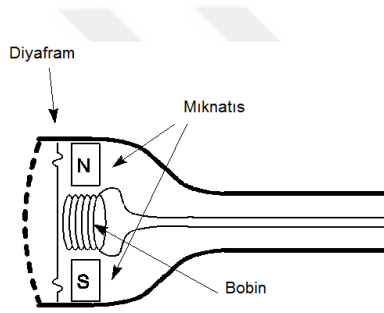
Sekiz şekline benzeyen yapıda algılama karakteristiğine sahip mikrofonlar bidirectional olarak isimlendirilmektedir. Bu algı yapısında önden ve arkadan gelen sesler eşit şekilde algılanırken yanlardan gelen sesler daha az algılanacaktır.

Mikrofona gelen ses dalgaları diyaframa çarpar ve ses basıncındaki değişikliklere göre diyafram, içe veya dışa doğru hareket ederek mekanik titreşim yapar. Diyafram arkasında

bulunan yapılar sayesinde bu titreşimlerin elektriksel işarete dönüştürülmesi gerçekleştirilir. Mikrofonlar, diyaframın arkasında kullanılan yapılara göre sınıflara ayrılmaktadır.

2.2.1. Dinamik mikrofonlar

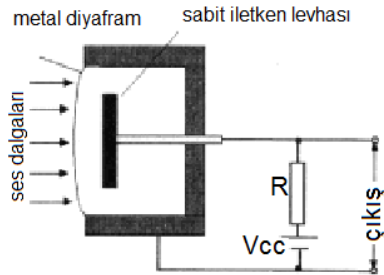
Ses dalgalarının diyaframı titreştirmesi ile diyaframa tutturulmuş olan silindirik yapılı bir doğal mıknatısın içine yerleştirilmiş olan bobin gelen ses dalgalarının frekansında elektrik işareti üretir. Bu üretilen elektrik sinyallerinin değeri 1-10mV düzeyinde oldukça düşük genlikli işaretlerdir. Bunun için dinamik mikrofonlar bir ön yükselteç ile kullanılmalıdır.



Şekil 2.3. Dinamik mikrofonun yapısı

2.2.2. Kapasitif (Kondansatör) mikrofonlar

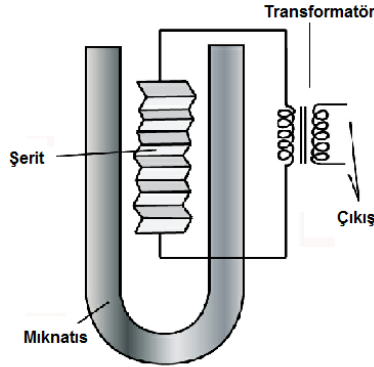
Diyaframın titreşmesi ile birlikte gelen işaretin şiddeti ile orantılı olarak içerisindeki kapasitenin değeri değişen mikrofonlardır. Bu değişim ile sesin özelliğine göre çıkışta elektrik sinyali oluşturulur. Hafif ve küçük yapılı üretilmesinin yanında yüksek ses kalitesi gerektiren sistemlerde bu mikrofonlar tercih edilir.



Şekil 2.4. Kapasitif mikrofonun yapısı

2.2.3. Şeritli mikrofonlar

Karşılıklı yerleştirilmiş iki mıknatıs arasında oluşan manyetik alana ince bir kalay veya alüminyum levhaya çarpan ses sinyalleri ile, manyetik alan içerisinde hareket eden levhada elektrik akımı oluşmaktadır.



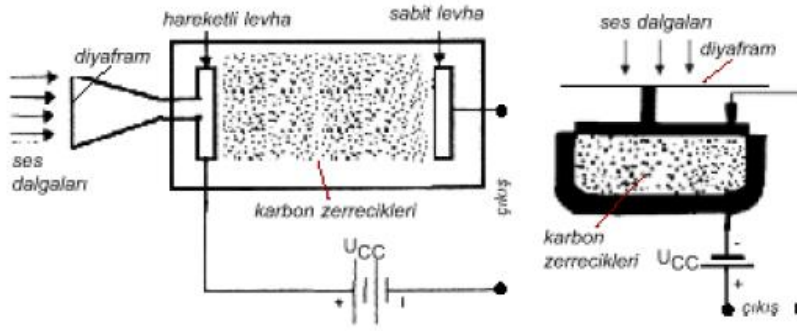
Şekil 2.5. Şeritli mikrofonun yapısı

2.2.4. Piezoelektrik kristalli mikrofonlar

Doğada bulunan kristal yapıları roşel tuzu, kuartz, baryum, turmalin gibi maddelere basınç uygulandığında üzerlerinde elektrik akımı oluşur. Oluşan elektrik akımı, uygulanan basıncın frekansına ve kuvvetine göre değişmektedir. Bu bilgiden yararlanılarak kristal mikrofonlar yapılmıştır. Ses titreşimleri diyaframı titreştirdiğinde kristal de titreşmekte ve alternatif elektrik akımı oluşmaktadır. Bu mikrofonlar içlerindeki kristalden dolayı sarsıntı, güneş ışığı, nem gibi etkenlere karşı hassastır.



Şekil 2.6. Piezoelektrik kristalli mikrofonun yapısı



Şekil 2.8. Karbon tozlu mikrofonun yapısı

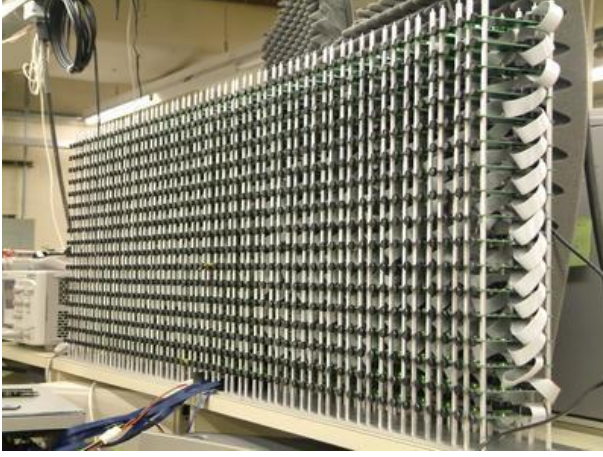
2.3 Mikrofon Dizileri

Mikrofon dizisi, eşzamanlı çalışan birden fazla mikrofonun geometrik olarak farklı noktalara yerleştirilmesiyle oluşan mikrofonlar kümesidir.

Mikrofon dizilerinin normal mikrofonlara göre temelde iki üstünlüğü bulunmaktadır. Bunlardan ilki kaynağa yakın olma zorunluluğunu ortadan kaldırmalarıdır. Ses sinyali elde edilir iken, en çok dikkat edilen husus ses sinyali ile gürültünün etkileşiminin en az seviyede tutulmasıdır. Normal mikrofonlarda bu soruna çözüm olarak mikrofon, ses kaynağına olabildiğince yakın bir yere yerleştirilmektedir. Bu sayede mikrofon doğrudan istenilen ses sinyalini alırken, gürültü sinyalini daha az almaktadır. Fakat birden fazla konuşmacının bulunduğu konferans salonları gibi ortamlarda bu tip uygulamalar yetersiz kalmaktadır. Buna karşılık mikrofon dizileri uzaysal olarak istenildiği gibi yerleştirilebilmektedir. Bu mikrofonlardan elde edilen sinyaller ile sinyal işleme süreci gerçekleştirildiğinde sinyal kalitesini bozan gürültü ortadan kaldırılabilen ve metrelerce uzaktan ses sinyali elde edilebilmektedir.

Diğer bir üstünlüğü ise mikrofon dizileri sayesinde geliş açısı hesaplama işlemleri yapılabilmektedir ve bu sayede sesin kaynağı veya konuşmacının takibi gerçekleştirilebilmektedir. Sağladığı üstünlükler sayesinde birçok farklı alanda kendilerine kullanım alanı bulmuşlardır.

Resim 2.1’de MIT laboratuvarlarında ses iyileştirme çalışmaları için kullanılan 1020 mikrofonlu bir mikrofon dizisi örneği görülmektedir.



Resim 2.1. MIT laboratuvarlarında geliştirilen 1020 mikrofonlu düzlemsel mikrofon [2]

Bir kulağı duymayan insanlarda sesin yönünün belirlenmesi mümkün değildir. İşitme cihazlarında mikrofon dizileri kullanılarak ses kaynağının yönünün belirlenmesi mümkün olmaktadır. Mikrofon sayısına bağlı olarak ses algılama ve tanıma başarısı da artırılabilir. Ses sinyalinin gürültülü ortamlarda gürültüden ayrıştırılması, çok konuşmacılı ortamlarda mikrofonun yönlendirilmesi, ses kaynağının yerinin tespit edilmesi, ses tanıma sistemleri, işitme cihazlarında başarının artırılması, eller serbest telefon görüşmesi kullanım amaçlarına örnek olarak verilebilir.

Ses kaynağının hareketli veya değişken olduğu ortamlarda ses kaynağının yerinin gerçek zamanlı olarak tespit edilmesi amacı ile mikrofon dizileri kullanılmaktadır. Geometrik olarak yeri tespit edilen ses kaynağına demetleme yapılarak, alınan ses sinyalinin kalitesi (SNR), mikrofon sayısına göre değişen miktarlarda artırılabilir.

Mikrofon dizisinin yapısı biliniyorsa, ses kaynağının yönünü tahmin edebiliriz. Mikrofon dizisinin geometrisi ve eleman sayısı, ses kaynağının mikrofonlara varış zamanları arasındaki farkın hesaplanma tahmininin doğruluğunu etkiler.



3. DAHA ÖNCE YAPILMIŞ ÇALIŞMALAR

Bu bölümde mikrofon dizileri ile ses konum tespiti ile ilgili daha önce yapılan ve bu çalışmada yol gösterici olarak kullanılmış olan akademik çalışmalardan bahsedilecektir. Kullandıkları yöntemler ve elde ettikleri sonuçlar incelenecektir.

Mikrofon dizileri ile ses konum tespiti konusunda yapılan çalışmalardan biri Alfonso Chacon-Rodriguez ve ark. tarafından yapılan çalışmadır [3]. Bu çalışmada [20, 300] Hz aralığında merkez frekanslı iki dijital sinyal arasındaki gecikmenin hesaplanması için bir yöntem sunulmaktadır. Yöntem, bir ses kaynağının geliş açısını belirlemek için bir gecikme hesaplaması gerçekleştirir. Hesaplama doğruluğu Çapraz Korelasyon Türev yönteminin daha önceki uygulamasına karşı test edilmiştir. Verilog RTL modelinin çalışması, yöntemin gerçek performansını değerlendirmek için bir Xilinx® FPGA üzerinde test edilmektedir. Bir FPGA üzerinde gerçekleştirilen dijital simülasyonların ve testlerin sonuçları, yöntemin işlevsel ve verimli olduğunu gösterdi ve örnekleme hızı 200 kHz olan $\pm 640 \mu\text{S}$ 'ye kadar olan gecikmelerin ölçülmesine izin veren genişletilmiş bir aralık sergilenmektedir. Bu çalışmada standart bir CMOS 0.5 teknolojisi kullanan bu yöntemin simülasyon sonuçları da sunulmuştur. Analog simülasyonların sonuçları, diğer uygulamalara göre toplam güç tüketiminde önemli bir gelişme olduğunu göstermektedir. Standart bir CMOS teknolojisinde bir ASIC tasarımının simülasyonları, önceki uygulamalara göre gecikme aşaması başına yaklaşık 25 kat daha fazla enerji tasarrufu öngörmektedir.

Diğer bir çalışmada Kefeli ve Ertürk ses dalgaları yayan cisimlerin konumlarını tespit edebilmek için bir yöntem geliştirmiştir. Geliştirdikleri bu yöntemi Texas Instruments'in DaVinci işlemcisini kullanarak gerçekleştirmişlerdir [4]. Çalışmada, iki eşdeğer mikrofon farklı konumlara yerleştirilmiştir ve bu mikrofonlardan alınan ses işaretleri arasındaki gecikme çapraz ilinti ile hesaplanmaktadır. Bulunan gecikme bilgisinden yararlanılarak ses kaynağının yönü açısal olarak tespit edilmektedir. TI DaVinci platformu sistemin gerçekleştirilmesi aşamasında kullanılmıştır. Gerçekleştirilen sistem ile üç boyutlu uzayda konum için bir çember verecek şekilde konum tespiti yapılabilmektedir. Sistem TI DaVinci platformunda gerçekleştirilmiş olup gerçek zamanlı olarak çalışmaktadır. Mikrofon sayısının artırılması ile daha fazla alıcı ile çalışılarak bu konum bilgisinin noktasal olarak

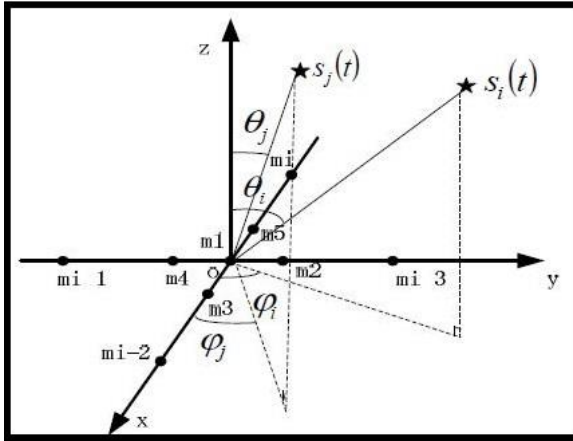
tespit edilebilmesi de mümkündür ve geliştirilmiş olan sistemin bu duruma adapte edilebilme yeteneği mevcuttur. Ayrıca araç veya cisimlerin geliş yönünün tespiti için de önerilen yaklaşımın kullanılması mümkündür.

Zhu, Guo ve Lin yaptıkları çalışmada, bilinmeyen sayıda ses kaynağı için varış yönlerini (DOA) tahmin etme sorununu bir mikrofon dizisi kullanarak çözmüştür [5]. Ses sinyali genellikle geniş bantlı ve durağan olmadığından, Kısa Süreli Fourier Dönüşümü (STFT) kullanılmaktadır. Bununla birlikte, pratik uygulamalarda, kaynakların eksik olması veya sık sık görünmesi meydana gelmekte ve bu da zamanla değişen sayıda hedefle sonuçlanmaktadır. Bu problemle başa çıkmak için, tüm zaman alanını bazı küçük zaman pencerelere bölmek için yeni bir ön işleme yöntemi sunulmaktadır. Her zaman penceresi için, örnek kovaryansı hesaplanmakta ve DOA tahminini üretmek için kullanılmaktadır. Ayrıca, bağımsız kaynakların sayısını tahmin etmek için iki yöntem (Özdeğer tabanlı ve Bilgi Kuramı tabanlı) kullanılmaktadır. Bazı seçilmiş simülasyon sonuçları, DOA'ların tahmin performansını ve kaynak sayısındaki değişimin tespitinde zaman doğruluğunu göstermek için verilmiştir. Algoritma, gürültüye ve durağan olmayan sinyallere karşı dayanıklıdır. Fakat önerilen yaklaşım kendi problemlerine sahiptir. Mikrofonların ve ses kaynaklarının sayısı arttıkça, hesaplama yükü önemli ölçüde artabilmektedir.

M.Davidson Kamala Das ve ark. yaptıkları çalışmada akustik sinyalin konumunu elde etmeyi ve azimuth açısını tahmin etmeyi amaçlamaktadır [6]. Akustik kaynak lokalizasyonu için tüm fonksiyonlar, FPGA 'de özel bir paralel mimari kullanılarak uygulanmaktadır. Ses konumlandırmasının tüm modülleri tek bir Virtex-5 FPGA 'ye entegre edilmiştir. Standart bir Google Veri tabanından bir ses sinyali ve gecikmeli ses sinyali toplanmıştır. Her bir orijinal sinyal çifti ve veri tabanındaki gecikme süreleri, istenen örnekleme hızında (16kHz) önceden işlendi ve gürültü önleme amacıyla Bandpass FIR Filtresinden geçirilmiştir. Tüm ses lokalizasyon modülleri tek bir Virtex-5 FPGA 'ya entegre edilmiştir. 36Kb blok RAM bağımsız örnek akış modülü olarak kullanılmıştır. Ön İşleme için kullanılan Filtre, gürültüyü ortadan kaldırmış ve sistemi güvenilir ve gerçek zamanlı uygulamalar için uygun hale getirmiştir.

YU Chunhe ve YU Yuanyuan yaptıkları çalışmada gürültülü bir ortamda ana konuşmacının akustik sinyalini çıkarmak için, çapraz mikrofon dizisine dayanan bir sinyal iletim modeli kurmaktadır. Bu sinyal iletim modeline göre, ana konuşmacının yönü

geliştirilmiş MUSIC kullanarak kestirilmekte ve bir uzamsal filtreleme yöntemi kullanarak mikrofon dizilerinden gelen ana konuşmacının sesi toparlanmakta ve izlenmektedir [7]. Mikrofon dizisinin yapısı ve kaynakların yönü, sinyalin geldiği dizi elemanları arasındaki zaman gecikmelerini belirlemektedir. Çapraz mikrofon dizisine dayanan alıcı sinyal modeli Şekil 3.1’de verilmiştir. Bu çapraz dizi iletim modelinde dizi elemanlarının eleman aralığı eşittir. Üç boyutlu uzayda, ses kaynağının yönü, atalet açısı θ ve azimut açısı ϕ ile tanımlanabilmektedir.



Şekil 3.1. Çapraz mikrofon dizisinin sinyal iletim modeli [7]

MUSIC, geleneksel uzay spektrum tahmin yöntemidir. Tüm uzayda güç spektrumunu arayarak kaynağın yönünü tahmin edebilmektedir. MUSIC (ve geliştirilmiş MUSIC) birden çok kaynağın yerini belirleyebilmektedir, pozisyon için yüksek doğruluk, daha az hesaplama ve mükemmel kararlılık gibi pek çok avantaja sahiptir. Dizinin yapısı bilindiği sürece yönü doğru olarak tahmin edilebilmektedir. Gürültülü ortamlarda, alınan sinyaller eş kanal paraziti ve çoklu yol yayın sinyali içermektedir, bu nedenle klasik MUSIC algoritması ana konuşmacının konumunu doğru bir şekilde bulamamaktadır. Bu nedenle, geliştirilmiş MUSIC ana konuşmacının yönünü belirleyebilmek için kullanılabilir.

Alan filtreleme (hüzmeleme (beamforming)), dizi sinyal işleminin önemli bir yönüdür. Uzamsal filtreleme, esas olarak, her elemanın sinyalinin ağırlıklandırma işlemi vasıtasıyla arzulanan sinyali güçlendirmektedir. Aynı zamanda girişim sinyallerini de azaltmaktadır. Ve her elemanın ağırlık katsayısı kaynak yönü ile değiştirilmektedir. Ana konuşmacının yönünü elde ettikten sonra, mikrofon dizisi, ana konuşmacının yönünü uzamsal filtreleme ile işaret eder, böylece ana konuşmacının sinyalini artırır ve diğerlerini azaltır.

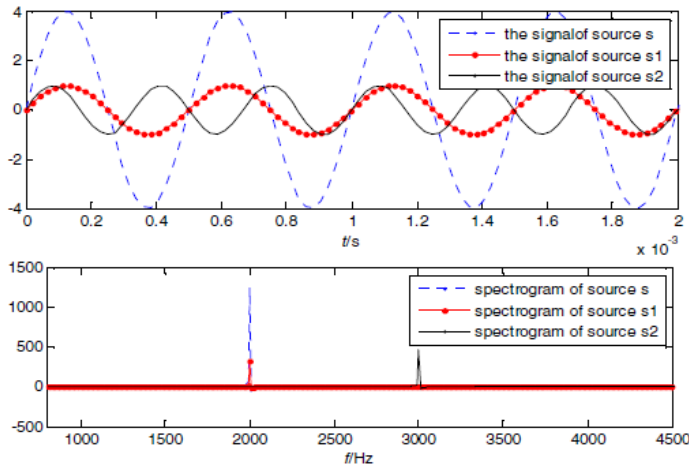
Bu çalışmada 17 çok yönlü mikrofon kullanılmaktadır, eleman aralığı $d = 5\text{cm}$ 'dir. Çapraz mikrofon dizisinin referans ögesi koordinatların kaynağında bulunmakta ve dizi eksenini ile çakışmaktadır. Çizelge 3.1, MUSIC ve iyileştirilmiş MUSIC'in iki kaynaklı ses kaynağının yerini belirleme sonuçlarını içermektedir.

Çizelge 3.1. Ses kaynağının yerini belirlemek için MUSIC ve geliştirilmiş MUSIC algoritmasının karşılaştırılması [7]

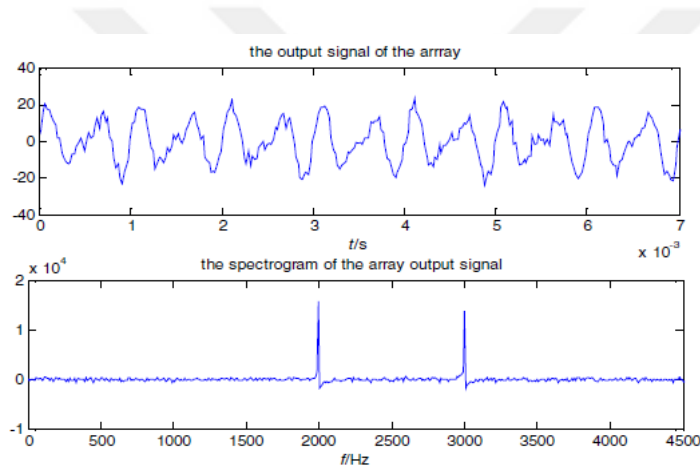
Algorithm Source	Classical MUSIC	Improved MUSIC
$s_1=(30,200)$ $s_2=(70,110)$	$s_1'=(47,210)$ $s_2'=(90,113)$	$s_1'=(25,200)$ $s_2'=(72,114)$
$s_1=(40,70)$ $s_2=(30,150)$	$s_1'=(22,134)$ $s_2'=(48,128)$	$s_1'=(31,57)$ $s_2'=(41,138)$
$s_1=(70,30)$ $s_2=(30,150)$	$s_1'=(31,107)$ $s_2'=(72,113)$	$s_1'=(60,23)$ $s_2'=(33,136)$
$s_1=(30,30)$ $s_2=(50,170)$	$s_1'=(57,82)$ $s_2'=(30,124)$	$s_1'=(28,28)$ $s_2'=(40,160)$

Ana ses kaynağının yönü, merkez frekansı kullanılarak tahmin edilmektedir. Çizelge 3.1, geliştirilmiş MUSIC'in, klasik algoritmaya kıyasla aynı frekansta olan çoklu kaynakları bile doğru bir şekilde lokalize ettiğini göstermektedir.

Bu çalışmadaki diğer bir deneyde, konuşmacılar sinüsoidal sinyallerle temsil edilmektedir. Üç ses kaynağı bulunmaktadır ve maksimum sinyal enerjisine sahip olan sinyal ana ses kaynağını temsil etmektedir. Yön parametreleri sırasıyla $s_1(\theta_1, \phi_1) = (30,30)$, $s_2(\theta_2, \phi_2) = (50,170)$, $s_3(\theta_3, \phi_3) = (80,90)$ ve merkez frekansı ayrı olarak $f_1 = f_2 = 2000\text{Hz}$, $f_3 = 3000\text{Hz}$ 'dir. Şekil 3.2, ses kaynağının sinyallerini göstermektedir. Ana ses kaynağının sinyal alanında s_1 sinyali olduğu bilinmektedir.



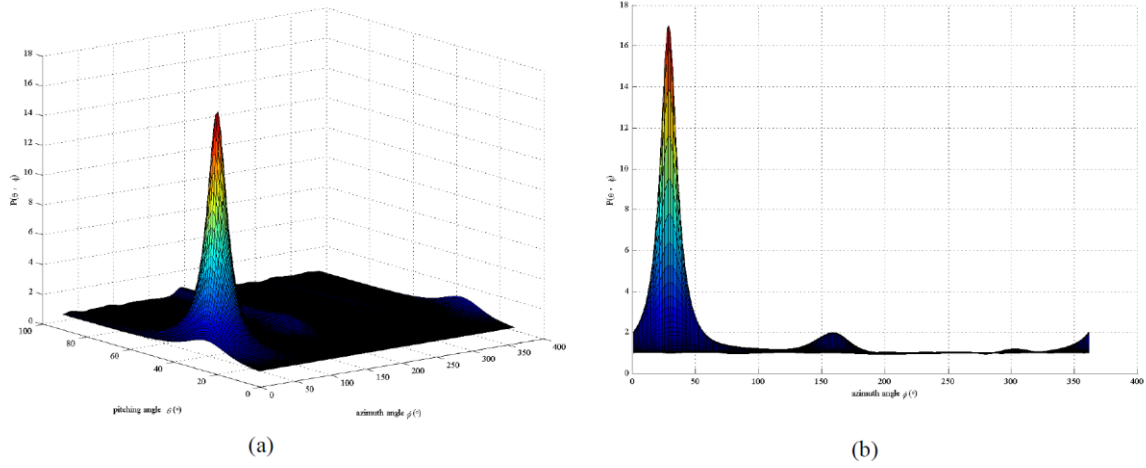
Şekil 3.2. Akustik sinyaller [7]



Şekil 3.3. Mikrofon dizisinin çıkış sinyali ve spektrumu [7]

Şekil 3.4’de diziden alınan sinyal ve onun spektrogramı vardır. Frekansı 2000Hz olan sinyalin maksimum enerji sinyali olduğu görülebilmektedir. Yani, istenen sinyalin bilgisi bilinebilmektedir.

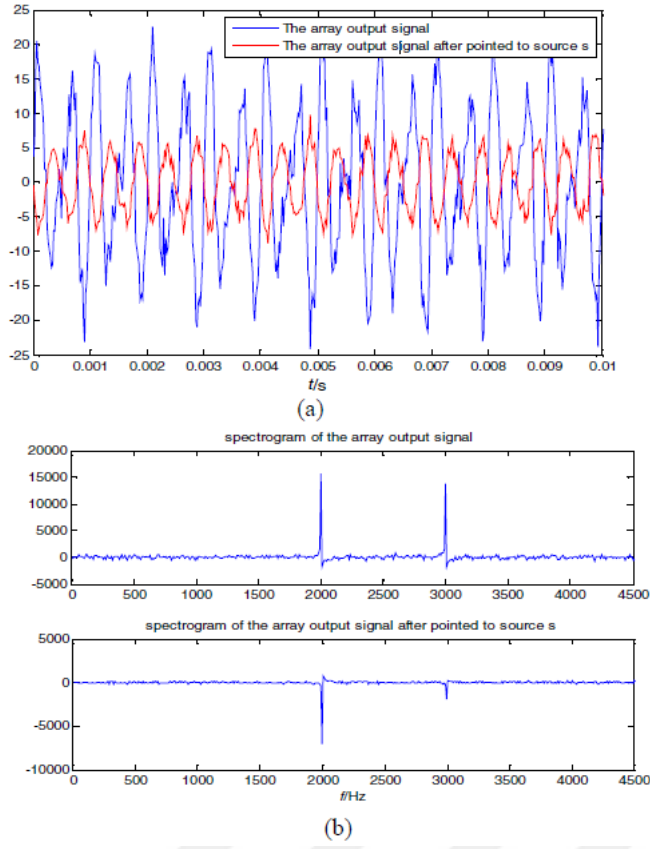
Şekil 3.5’de ana ses kaynağının, yani $f = 2000\text{Hz}$ frekansındaki sinüzoidal sinyalin, yönünü tahmin etmek için alınan sinyaller ele alınmaktadır.



Şekil 3.4. Dizinin uzaysal güç spektrumu [7]

Şekil 3.4’de frekansları $f = 2000\text{Hz}$ olan ve yönleri sırasıyla $s1'$ (28,28), $s2'$ (40,160) olan iki sinyal görülmektedir. Ek olarak, dizinin alan spektrumu $s1$ yönünde en büyük olmaktadır. Yani maksimum enerji sinyalinin yönü $s1'$ tahmin edilmiştir. Bu nedenle, $s2(50,170)$ yönünde ortak kanal girişim kaynağı olsaydı bile, gelişmiş MUSIC yine de ana hoparlörün yönünü tahmin edebildi ve kestirilen değer aynı anda elde edilmiştir. Sonuç olarak, testte belirlenen ana hoparlörün yönü $s1$ (30,30) için, geliştirilmiş MUSIC klasik algoritma ile karşılaştırıldığında doğru bir şekilde lokalize etmiştir.

Hedef kaynağı $s1'$ in kestirilen değeriyle, uzamsal filtreleme yöntemini kullanarak dizi $s1'$ olarak belirlenmiştir ve kaynak hedefe ulaşılması sağlanmıştır. Şekil 3.5, çıkış sinyalinin ana hoparlöre işaret etmeden önce ve sonra karşılaştırılmasını göstermektedir. Şekil 3.5 (a) 'da, dizi ana hoparlöre işaret ettikten sonra, dizinin çıkış sinyalinin tamamen zayıfladığı görülmektedir. Şekil 3.5 (b) 'de, alınan dizi sinyalinde hedef kaynağın SNR'sinin arttığı görülmektedir. Burada, hedef kaynağın ($s1$) arttırılması ve girişim sinyalinin ($s2$ ve $s3$) azaltılması sağlanmaktadır.



Şekil 3.5. (a) ve (b) Ana hoparlöre işaret eden dizinin önceki ve sonraki çıkış sinyallerinin karşılaştırılması [7]

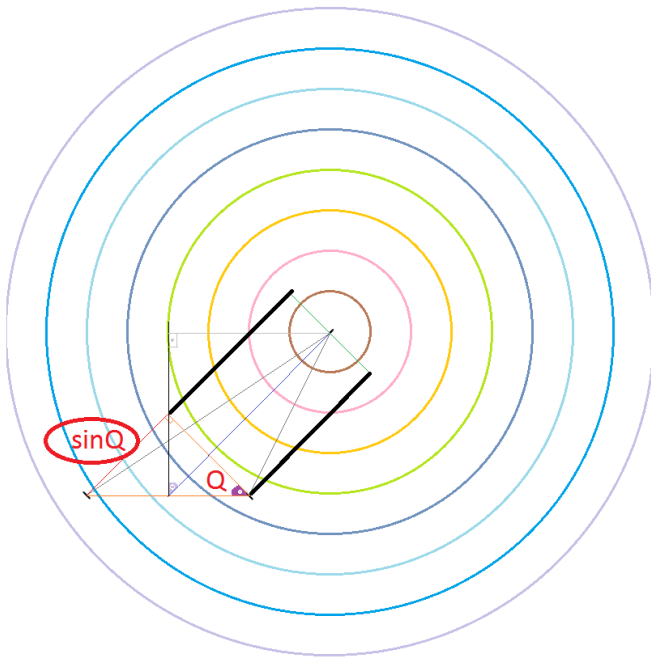


4. SES KAYNAĞININ YERİNİN BULUNMASI

Ses kaynağının yönü ses kaynağının yerini belirleme teknolojisi ile tahmin edilebilmektedir. Mikrofon dizisine dayanan ses kaynağının yerini belirleme yöntemi üç kategoriye ayrılabilir [7]:

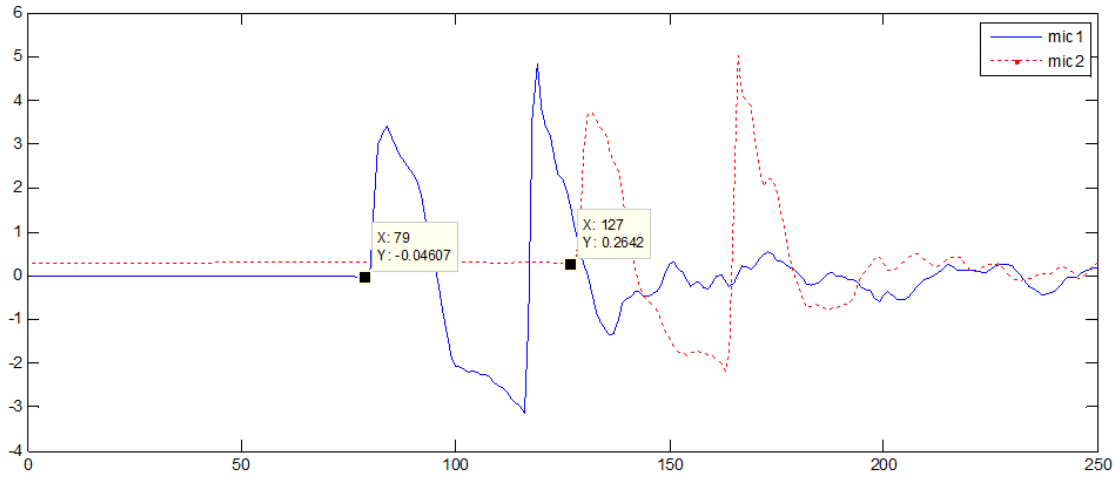
- Çapraz korelasyon zaman gecikmesi kestirimine dayalı ses kaynağının yerini belirleme teknolojisi
- Yönetilebilir hüzmleme (beam forming) teknolojisine dayalı ses kaynağının yerini belirleme teknolojisi
- Uzaklık spektrum kestirimi

Mikrofonlara gelen seslerden konum tespiti için seslerin mikrofonlara geliş zamanları arasındaki farkın (Time Difference of Arrival - TDOA) hesaplanması gerekmektedir. Bunun için literatürde birçok çalışmada çapraz ilintinin kullanımı önerilmektedir [8]. Bu çalışma kapsamında, ses konum tespiti için çapraz ilintinin kullanılması üzerinde çalışılmıştır. Şekil 4.1 'de görüldüğü gibi bir ses kaynağının dairesel olarak ses dalgaları yaydığı bilindiğinden, aynı ses dalgasının iki ayrı mikrofona ulaşması için farklı uzunlukta yollar gitmesi gerekmektedir.



Şekil 4.1. Ses kaynağının yayılımı ve mikrofonlara ulaşma süreleri arasındaki farklılık

Dolayısıyla, aynı ses dalgası iki ayrı mikrofona farklı zamanlarda ulaşmaktadır. İdeal bir yayılım ortamında eş mikrofonlar eşdeğer kabul edildiğinde, iki mikrofona ulaşan ses işaretlerinin arasında sadece öteleme olması gerekmektedir. Pratikte ise yayılım ortamının homojen olmaması, mikrofonların eşdeğer karakteristikte olamaması, mikrofonlar arası mesafe gibi etkenlerden dolayı, mikrofonlara ulaşan ses sinyalleri arasında zamansal farklılığın yanında genlik olarak farklılık da oluşmaktadır. Şekil 4.2 'de iki farklı mikrofona ulaşan aynı ses görülmektedir.



Şekil 4.2. İki mikrofondan alınan işaretle zaman farkı

Mikrofonlar arasındaki bu zaman farkının otomatik olarak tespit edilebilmesi için kullanılan çapraz ilintinin eşitlikleri aşağıda verilmektedir. Eğer $x_1[n]$ ve $x_2[n]$ sol ve sağ mikrofonlardan alınan 2 farklı sayısal işaret ve aralarında yalnızca n_0 örnek kadar bir öteleme (ilerleme ya da gecikme) var ise;

$$x_1[n] \xrightarrow{\text{AZFD}} X_1(e^{j\Omega}) \quad (4.1)$$

$$x_2[n] \xrightarrow{\text{AZFD}} X_2(e^{j\Omega}) \quad (4.2)$$

$$x_1[n] = x[n - n_0] \quad (4.3)$$

$$X_2(e^{j\Omega}) = X(e^{j\Omega}) = |X(e^{j\Omega})| \cdot e^{j\angle X(e^{j\Omega})} \quad (4.4)$$

$$X_1(e^{j\Omega}) = e^{-j\Omega n_0} \cdot X(e^{j\Omega}) = |X(e^{j\Omega})| \cdot e^{j\angle X(e^{j\Omega})} \cdot e^{-j\Omega n_0} \quad (4.5)$$

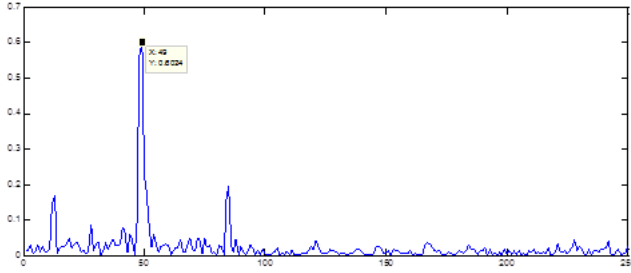
$$X_2^*(e^{j\Omega}) = |X(e^{j\Omega})| \cdot e^{-j\angle X(e^{j\Omega})} \quad (4.6)$$

$$Y(e^{j\Omega}) = \frac{X_1(e^{j\Omega}) \cdot X_2^*(e^{j\Omega})}{|X_1(e^{j\Omega}) \cdot X_2^*(e^{j\Omega})|} \quad (\text{Çapraz İlinti}) \quad (4.7)$$

$$Y(e^{j\Omega}) = \frac{|X(e^{j\Omega})| \cdot e^{j\angle X(e^{j\Omega})} \cdot e^{-j\Omega n_0} \cdot |X(e^{j\Omega})| \cdot e^{-j\angle X(e^{j\Omega})}}{|X(e^{j\Omega})| \cdot |X(e^{j\Omega})|} = e^{-j\Omega n_0} \quad (4.8)$$

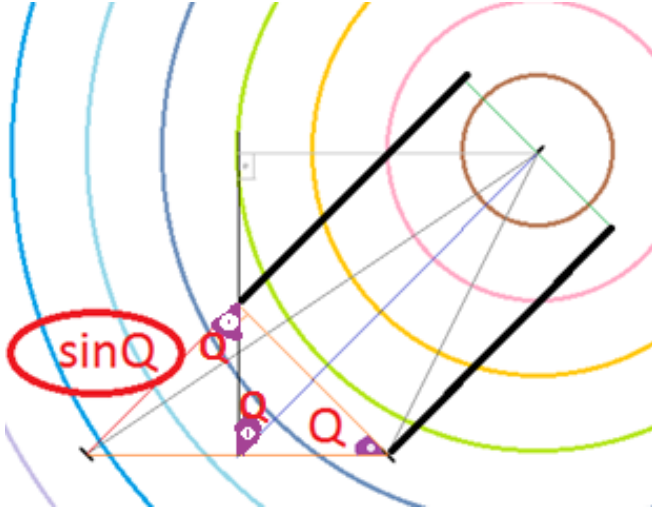
$$e^{-j\Omega n_0} \xleftrightarrow{AZFD} \delta(n - n_0) \quad (4.9)$$

İki işaretin çapraz ilintisi sonrasında öteleme sayısı kadar mesafede bir dürtü oluşmaktadır. Pratikte, çapraz ilintisi uygulanan iki işaret genlik olarak farklılıklar göstereceğinden ideal dürtü işareti elde edilememekte fakat öteleme sayısının olduğu konumda bir tepe elde edilmektedir. Ayrıca bu tepe değerinin 1 değerine yakın olması iki işaretinin benzerliğinin yüksek olmasını; 0 değerine yakın olması da iki işaretin farklılaştığını göstermektedir. İki mikrofondan alınan ve Şekil 4.2’de verilen ses işareti için çapraz ilintisi sonucu Şekil 4.3’de verilmektedir.



Şekil 4.3. İki mikrofondan alınan ses işareti ve çapraz ilinti sonucu

Şekil 4.2 ’de işaretlerin arasındaki örnek farkı incelendiğinde 1. mikrofondan (mic1) alınan işaretin 2. mikrofondan (mic2) alınan işarete göre 48 örnek (127-79) ileride olduğu görülmektedir. Şekil 4.3 ’deki çapraz ilintinin sonucuna bakıldığında, en büyük değere sahip bileşenin bulunduğu konumun başlangıç noktasına (1. değer) göre 48 örnek (49-1) ileride olduğu bilgisini elde edilmektedir ki bu değer örnek uzayındaki işaretler arasındaki öteleme farkıdır. Ayrıca Tepenin 0.6024 değeri, bu iki işaret arasındaki benzerlik hakkında bilgi vermektedir. Çapraz korelasyonu ile elde edilen gecikme miktarından sesin geliş yönünün elde edilebilmesi için kullanılan trigonometrik ilişki Şekil 4.4 ’de gösterilmektedir.



Şekil 4.4. Ses kaynağının mikrofona geliş ve mikrofona göre ses kaynağının açısal kestiriminin gösterimi

Yukarıda şekilde anlaşıldığı gibi 2 mikrofon kullanıldığında tam ortalarından dik eksen ile yapılan açı ses kaynağının konumu hakkında bilgi vermektedir. $f_{\text{ö}}$ örnekleme frekansıdır ve her iki örnek arası mesafe $1/f_{\text{ö}}$ olduğundan, maksimum tepenin olduğu yer maks_i ise, gecikme miktarının zaman düzleminde karşılığına ITD dendiği taktirde;

$$\text{ITD} = \text{maks}_i \cdot 1/f_{\text{ö}} \quad (4.10)$$

olmaktadır. Gecikme miktarı m ile gösterilirse;

$$m = \text{ITD} \cdot V_{\text{ses}} \quad (4.11)$$

olmaktadır. 2 mikrofon arasındaki uzaklığa D_m ile gösterilirse;

$$\sin Q = \frac{m}{D_m} \quad (4.12)$$

$$Q = \sin^{-1} \frac{m}{D_m} \quad (4.13)$$

Eşitliklerinden Q açısı bulunabilmektedir.

4.1. Çapraz İlinti Yönteminin Sadeleştirilmesi

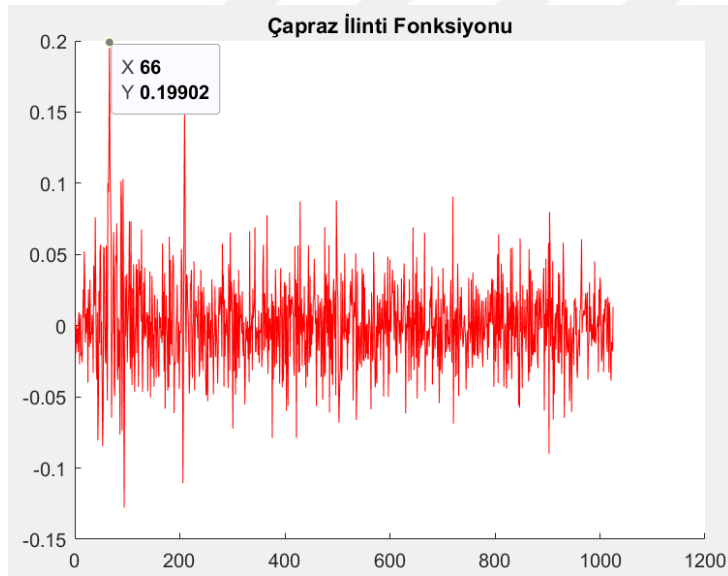
Çapraz ilinti yöntemi algoritmanın performansını etkilemeyecek şekilde sadeleştirilmiş ve bu sayede çapraz ilintinin bulunması için gereken bölme işlemi ortadan kaldırılmıştır. Eşitlik 4.7’de belirtilen çapraz ilinti fonksiyonunda bölme operatörü teorik olarak sabit bir

katsayı çarpımını sadeleştirmektedir. Bu bölme operatörü kullanılmadığında sonuca sabit bir çarpım gelmektedir.

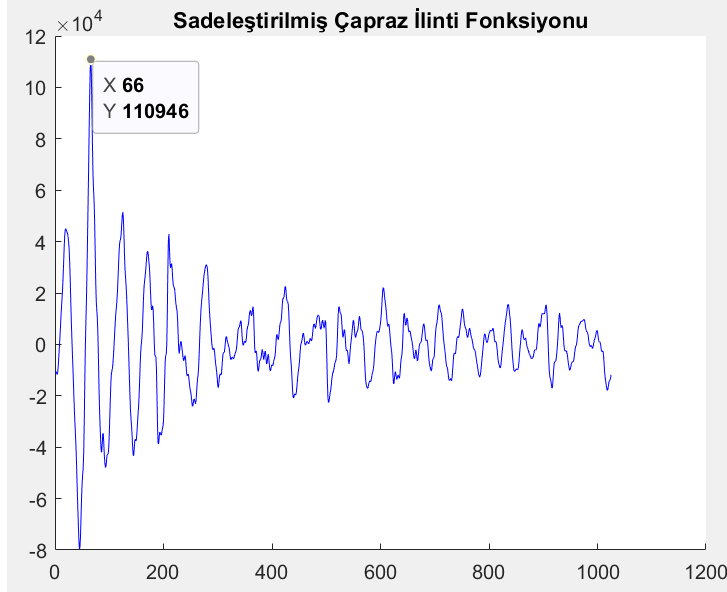
$$Y(e^{j\Omega}) = X_1(e^{j\Omega}) \cdot X_2^*(e^{j\Omega}) \quad (\text{Çapraz İlinti}) \quad (4.14)$$

$$Y(e^{j\Omega}) = |X(e^{j\Omega})| \cdot e^{j\angle X(e^{j\Omega})} \cdot e^{-j\angle X(e^{j\Omega})} \cdot |X(e^{j\Omega})| \cdot e^{-j\angle X(e^{j\Omega})} = |X(e^{j\Omega})|^2 e^{-j\angle X(e^{j\Omega})} \quad (4.15)$$

Çapraz ilintideki bölme işlemi sonucu normalize etmektedir. Bu işlemin donanım maliyeti göz önüne alındığında bu bölme işlemi olmadan da başarılı sonuçlar alınabileceği incelenmiştir. Şekil 4.5 'de çapraz ilinti fonksiyonu ile elde edilen sonuç görülmektedir. En büyük noktanın bulunduğu yer 66. örnekte gözlemlenmiştir ve değeri 0.19902'dir. Şekil 4.6'da sadeleştirilmiş çapraz ilinti fonksiyonu ile elde edilen sonuç görülmektedir. En büyük noktanın bulunduğu yer 66. örnekte gözlemlenmiştir ve değeri 110946'dır. Bu örnekte de görüldüğü gibi fonksiyonda kullanılan bölme işlemi genlik bazında normalizasyon işlemi yapmaktadır.



Şekil 4.5. İki işaret arasındaki gecikmenin çapraz ilinti fonksiyonu ile hesaplanması



Şekil 4.6. İki işaret arasındaki gecikmenin sadeleştirilmiş çapraz ilinti fonksiyonu ile hesaplanması

İncelenen ses örnekleri için çapraz ilinti fonksiyonunda bölme işleminin kullanılmaması maksimum noktanın yerinin bulunma performansını etkilememektedir. Bölme işlemi ile sonuç genlik bazında normalize edilerek sonucun daha belirgin, genlik değerlerinin daha küçük sayılar olması sağlanmaktadır. Bölme işlemi donanımı tasarlanacak olan sistem için ek olarak alan maliyeti ve hesaplama maliyeti getireceğinden ve bu işlem sistemin çalışma performansını etkilemediğinden çapraz ilinti fonksiyonunun sadeleştirilerek kullanılması tercih edilmiştir.

Çizelge 4.1 'de çapraz ilinti yöntemi ve sadeleştirilmiş çapraz ilinti yöntemi ile gecikme hesaplaması sonuçları verilmiştir. Bu sonuçların oluşturulması için çapraz ilinti yönteminde kullanılan Fourier işlemi için 1024 pencere boyutlu diziler kullanılmıştır. Ses kaydı üzerinden anlamlı ses verisinin olduğu dizi aralığından 1024 pencere boyutlu dizi kullanılarak çapraz ilinti yöntemi ve sadeleştirilmiş çapraz ilinti yöntemi ile gecikme hesaplaması yapılmıştır. Ters fourier işlemi yapılarak maksimum dürtünün olduğu dizi incelenmiştir. Ses kaynağının mikrofonlara göre sağ veya sol yönde olmasına göre gecikme miktarı 1024 ün yarısından büyük veya küçük olarak sonuç vermektedir. 512 değerinden büyük olan sonuçlardan 1024 çıkarılarak negatif bir gecikme değeri bulunur. Negatif olan gecikme değeri işaretin diğer mikrofondan alınan işarete göre ileride veya geride olduğu bilgisini bize vermektedir.

Çizelge 4.1. Çapraz ilinti yöntemi ve sadeleştirilmiş çapraz ilinti yöntemi ile gecikme hesaplaması sonuçları

SES KAYITLARI	Çapraz ilinti yöntemi ile gecikme hesaplaması	Sadeleştirilmiş Çapraz İlinti yöntemi ile gecikme hesaplaması
Kayıt 1	24	19
Kayıt 2	66	66
Kayıt 3	54	57
Kayıt 4	1023 / -1	1023 / -1
Kayıt 5	65	67
Kayıt 6	1002 -> -22	1007 -> -17

Mevcut ses verileri üzerinde çapraz ilinti yöntemi ve sadeleştirilmiş çapraz ilinti yöntemi ile gecikme hesaplaması ile elde edilen sonuçlar incelendiğinde hesaplanan değerler arasındaki maksimum hata farkı 5 örnektir ve toplam ortalama hata farkı 2.5 örnektir. Bu gecikme miktarı yaklaşık olarak 3 dereceye karşılık gelmektedir. Sonuç olarak sadeleştirmeden kaynaklı hata farkının sistem çalışma performansını büyük ölçüde etkilemediği görülmektedir.



5. ALAN PROGRAMLANABİLİR KAPI DİZİLERİ (FPGA)

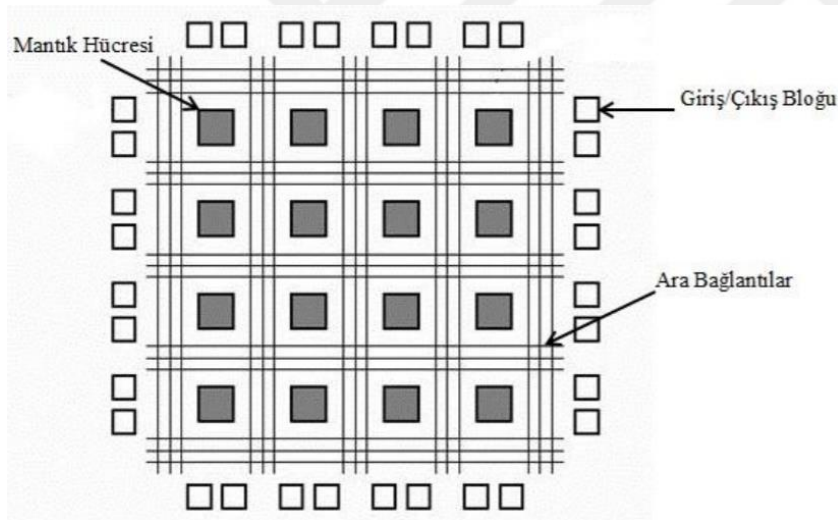
1980'lerin başlarında, PLD (Programmable Logic Device)'lerle ASIC (Application Specific Integrated Circuit)'ler arasında bir boşluğun olduğu açığa çıktı. Bir tarafta, programlanabilir, hızlı tasarımlar yapılabilen SPLD (Simple Programmable Logic Device) ve CPLD (Complex Programmable Logic Device)'ler gibi entegreler vardı. Ancak bunlar büyük ve kompleks fonksiyonları desteklemiyorlardı. Diğer tarafta ise ASIC'ler bulunuyordu. Bunlar, aşırı büyük ve kompleks fonksiyonları destekliyorlardı, ancak oldukça pahalı idi ve tasarımları çok zaman alıyordu. Dahası, bir tasarım ASIC olarak bir kez yapıldığında silikon içinde donduruluyordu. Bu aradaki boşluğu kapatmak için Xilinx firması 1984 de alan programlanabilir kapı dizileri (FPGA) adı altında yeni bir tümleşik devreyi piyasaya sürdü.

FPGA, Alan Programlanabilir Kapı Dizisi anlamına gelen Field Programmable Gate Array'in kısaltılmış halidir. Alan programlanabilir kapı dizileri istenen muameleye göre donanım üreten ve daha sonrada tasarımcı tarafından değiştirilebilen entegre yongalardır. Tasarım sırasında kullanıcıya esneklik sağlaması ve düşük maliyetli olması nedeniyle kullanımı gittikçe yaygınlaşmıştır. Savunma, uzay, sayısal işaret işleme, tıbbi görüntüleme ve otomotiv, FPGA' nın uygulama alanlarından bazılarıdır. Tasarımcının isteğine göre FPGA içerisindeki bağlantılar birbirlerine bağlanır ve istenen işlev gerçekleştirilir. Teorik olarak aklımıza gelen herhangi bir entegrenin yaptığı fonksiyonları ve işleri FPGA yapabilir.

FPGA'lerin en önemli özellikleri arasında paralel ve hızlı işlem yapabilme kabiliyeti gelir. Aynı anda birden fazla işlemi yapmaya paralel işlem yapabilmek denilebilir. Bilinen entegrelerin tamamı eş zamanlı işlem yapamazlar, bazıları sınırlı olarak gerçekleştirebilirler. FPGA ise kapasiteye ve uygulamaya bağlı olarak, birbiri ile eş zamanlı yüzlerce işlemi aynı anda yapabilir. Bu durum eş zamanlı işlem gerektiren uygulamalarda bu yongaları eşsiz ve değerli kılmaktadır. Eş zamanlı işleme bir örnek olarak; bir mikrofon dizisi ile ses kaynağının konumunun tespit edilmesi için dizideki tüm mikrofonlardan ses eş zamanlı olarak alınarak irdelenmektedir. Bir akustik işareti giriş portlarından almak, onu irdelemek ve gerekli çıkış portlarından bilgiyi göndermek gerekmektedir. Sonra ikinci akustik işaret için de aynı işlemleri gerçek zamanlı olarak

tekrarlamak gerekecektir. Standart entegreler (örneğin bir mikroişlemci) ile bu işlemleri sırasıyla yapıp bitirdikten sonra ikinci akustik işareti alması gerekecektir. Eğer bu işlemler yeterince hızlı yapılamaz ise gelen akustik bilgiler kaçırılabilir. FPGA’de ise bu işlemler paralel olarak devam eder. Tüm mikrofon dizilerinden akustik işaret eş zamanlı olarak alınır ve gerekli işlemleri yaparken ikinci akustik işaret bilgisi alınmaya başlanır. Mikrofon dizisinden alınan ilk akustik işaret dizisinin çıkış bilgisi gönderilirken, ikinci akustik işaret işlenmeye ve üçüncü akustik işaret alınmaya başlanabilir.

FPGA temel olarak mantık hücreleri, giriş/çıkış blokları ve ara bağlantılardan oluşur. Bir FPGA yongası, programcı tarafından yazılan kodlar ile programlanır. Derleyici program yardımı ile yonganın içerisindeki bloklar ve yollar arasında bağlantı kurularak, kodları yazılan devre gerçekleştirilir. Alan programlanabilir diye isimlendirilmesinin sebebi, ara bağlantıların ve mantık bloklarının imalat sürecinden sonra programlanabilmesidir.



Şekil 5.1. FPGA temel yapısı

Mikroişlemci ile FPGA arasındaki en temel ve büyük fark FPGA ’in donanımsal olarak sabit bir yapıya sahip olmamasıdır. Kullanıcının uygulamasına göre programlanabilir bir entegredir. İşlemcilerin donanım yapısı ise sabit bir yapıya sahiptir. Yani içerisindeki tüm transistör, bellek, çevre birimi yapıları ve bağlantıları sabittir. İşlemcinin yapabileceği işlemler (toplama, çarpma, I/O kontrol, vs) önceden tanımlı işlemlerdir ve kullanıcı yazılımla bu işlemleri kendi amacına uygun olarak “sıralı bir şekilde” işlemciye yaptırabilir.

FPGA 'de ise sabit olmayan bir donanım yapısı vardır ve kullanıcı tarafından tanımlanır. FPGA içerisindeki mantık hücrelerinin sabit olmasıyla birlikte bunların gerçekleştirdiği fonksiyonlar ve aralarındaki bağlantılar kullanıcı tarafından değiştirilebilir. Bu durumdan dolayı FPGA 'ın yapabileceği işlemler önceden tanımlanmış işlemler değildir. Yazılan HDL koduna göre istenilen işlemler “paralel olarak” yani aynı anda yaptırılabilir. FPGA 'yı işlemciden ayıran ve birçok alanda üstün kılan en önemli özellik paralel işlem yapabilme yeteneğidir.

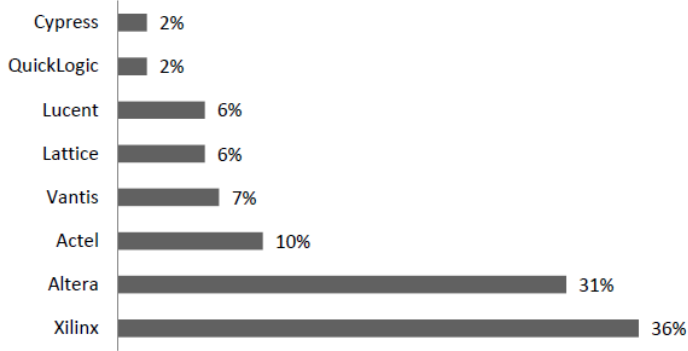
İşlemciler belli devrelerin rutin kontrol işlemlerinde genellikle daha kullanışlıdır. Örneğin bilgisayardan herhangi bir cihazı açıp kapamak gibi basit fonksiyonlar için FPGA kullanmak tercih edilmeyecektir. Yoğun veri işleme gibi performans gerektiren uygulamalarda FPGA ön plana çıkmaktadır.

5.1. FPGA ile Tasarım

Kullanıcı tasarladığı devreyi VHDL veya Verilog gibi donanım tanımlama dillerinden birini kullanarak yazılım diline döker. Projenin gerçekleştirileceği aygıtın türüne uygun olan bir derleyici program ile yazılan komutlar sentezlenir. İsteğe bağlı olarak bir benzetim programında devrenin güvenilirliği incelenir. Doğrudan bilgisayara USB veya seri port yardımı ile bağlanabilen FPGA donanımına, sentezlenen komutlar uygun bir ara yazılım ile aktarılır. Yazılımın çalışmasına uygun olan devre FPGA içerisinde gerçekleştirilir ve giriş çıkış birimleri ile kullanıma uygun hale gelir.

Resim 19’da SourceTech411.com web sitesinin yayınladığı 2013 yılında yayınladığı FPGA üreticileri ve 2010 yılı için pazar payları verilmiştir [9].

2010 Yılı FPGA Pazar Payları



Şekil 5.2. 2010 yılı FPGA üreticileri ve pazar payları [9]

5.2. FPGA Türleri

FPGA'lar "tekrar programlanabilir" ve "bir kez programlanabilir" olmak üzere iki farklı türde üretilmektedir. Yapılandırılma biçimlerindeki bu ayrım üretimde farklı yöntemlerin temel alınmasına sebep olmuştur. FPGA 'lar temel aldıkları üretim teknolojileri bakımından SRAM tabanlı, Anti-Sigorta tabanlı, EEPROM tabanlı ve Flash tabanlı olarak sınıflandırılabilir. Çizelge 5.1'de üretici firmalar ve tercih ettikleri üretim teknolojileri verilmiştir [10].

Çizelge 5.1. Üretici firmalar ve tercih ettikleri üretim teknolojileri [10]

Üretici Firma	Teknoloji
Altera	SRAM, FLASH
Actel	Anti-Sigorta
Lattice	SRAM, FLASH
QuickLogic	Anti-Sigorta
Xilinx	SRAM

5.2.1. SRAM tabanlı

FPGA üretiminde en çok tercih teknoloji edilen CMOS tabanlı üretimdir. SRAM tipi olan FPGA'lar defalarca programlanabilmektedir. Ayrıca SRAM tabanlı FPGA'lar devre üzerinde yerleşik durumdayken programlanabilir. FPGA'ların araştırma ve geliştirme

alanlarındaki kullanım amacı göz önüne alındığında ilk örnek tasarımı ve geliştirme aşamaları için en iyi seçenek olarak değerlendirilebilir.

Statik yapısı sebebiyle güç kesildiğinde içerisindeki veriler silinir bu yüzden her açılışta SRAM tabanlı bir FPGA 'nın tekrar yapılandırılması gerekir. Her açılışta tekrar yapılandırma verilerinin saklanması için PROM tipi bir bellek içerir. Ancak her türlü kar getiren ürünün kopyalandığı bir pazarın olduğu günümüzde PROM içerisindeki yapılandırma programı rahatlıkla alınıp kopyalanabilir. Buradaki asıl sorun programın alınıp kopyalanmasından ziyade, ters mühendislik uygulanarak mevcut program yapısının geliştirilmesi ve asıl üretimi yapan firmanın haksız rekabete uğramasıdır. Bu ciddi güvenlik sorununa önlem olarak PROM içerisine yeniden yapılandırma programı şifrelenerek yazılır. Şifre anahtarı da FPGA 'nın JTAG portu aracılığıyla SRAM tabanlı kaydedici içerisinde barındırılır. Şifre anahtarının SRAM tabanlı kaydedici içerisinde silinmemesi için küçük bir güç kaynağı olarak devreye bir pil ilave edilmesi gerekir. Bu durumda devrenin boyut, ağırlık ve maliyet gibi faktörlerden yana olumsuz etkilenmesine sebep olur.

5.2.2. Anti sigorta tabanlı

SRAM tabanlı FPGA 'ların aksine, yerleşik bulunduğu devre üzerinden ayrı bir başka programlayıcı aygıt ile programlanabilirler. Sistemden güç kesildiğinde içerisindeki program silinmez. Bu sebeple devre üzerinde harici bir hafıza birimi ve güç kaynağı gerektirmez. Olağan yapısından dolayı ortamdaki radyasyondan etkilenmezler. En önemli avantajlarından biriside güvenlik yönündendir. İçerisine yazılan yapılandırma programı FPGA 'nın ulaşamayacak kadar derinlerinde yer alır. SRAM 'ın aksine ulaşılması zordur ve kopyalanamaz. Sistem bekleme durumundayken aynı özelliklerde bir SRAM tabanlı ürüne göre yaklaşık yüzde 20 oranında az güç tüketir.

Ancak üretim aşamasından sonra programlama için ayrı bir aygıt gerektirmesi ve sadece tek sefer programlanabilmesi gerekçelerinden dolayı tercih olarak SRAM tabanlı mimarini gerisinde kalmıştır.

5.2.3 EEPROM / FLASH tabanlı

SRAM tabanlı mimariye benzer bir yapısı vardır. Devreden ayrı başka bir programlayıcı aygıt aracılığıyla programlanabilir. Ancak bazı çeşitleri devre üzerinde programlanabilmesine rağmen programlama süresi SRAM tabanlı mimariye göre oldukça uzundur. Üzerindeki veri kalıcı olarak saklanır. Sistem aktif duruma getirildiği gibi kullanıma hazırdır.

Program güvenliği için çoklu anahtar ile şifreleme kullanılır. Bu anahtarın bit genişliği 50 bitten fazladır. Bu da programın şifre anahtarının kırılmasını neredeyse imkânsız kılmaktadır. Yapısında kullanılan transistor boyutları SRAM içerisinde kullanılan transistörlerle karşılaştırıldığında daha küçüktür. Bu durumda içerisindeki mantık birimleri birbirine daha yakın paketlenir, bu da paket içerisindeki yollardan kaynaklanan gecikmeleri azaltır.

Çizelge 5.2. Programlama teknolojilerinin karşılaştırılması [11]

	SRAM	Anti-Sigorta	EEPROM/FLASH
Teknoloji	En yüksek düzeyde	Bir veya daha fazla nesil geride	Bir veya daha fazla nesil geride
Tekrar Programlanabilirlik	Var (sistem içerisinde)	Yok	Var (sistem içerisinde veya dışında)
Tekrar Programlama Hızı	Hızlı	-	SRAM'in 3 katı daha yavaş
Geçicilik (Sistem açılışında yeniden programlanma gerekliliği)	Var	Yok	Yok
Ar-Ge kullanımı	Prototip geliştirmede sık tercih edilir.	Yok	Kabul edilebilir düzeyde kullanılabilir.
Anlık hızlı açılış	Yok	Var	Var
Program güvenliği	Bitstream şifreleme kullanıldığı zaman kabul edilebilir düzeyde	Çok Yüksek	Çok Yüksek
Yeniden yapılandırma hücre boyu	Büyük (6 transistor)	Çok Küçük	Orta-Küçük (2 transistor)
Güç tüketimi	Orta	Düşük	Orta
Hızlı Uygulama Geliştirme zorluğu	Yok	Var	Yok

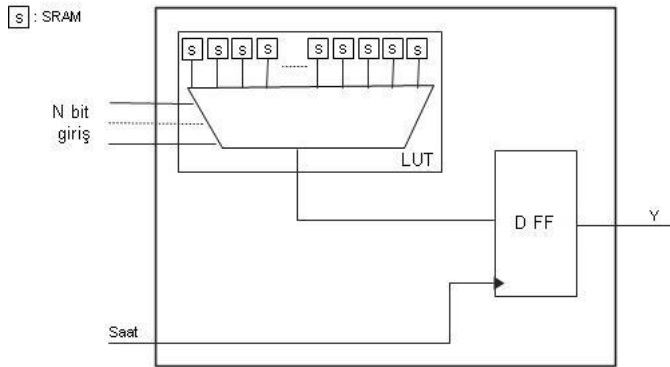
5.3. FPGA İç Yapısı

FPGA mimarisi, Resim 18’de olduğu gibi temel olarak mantık hücreleri(Logic Cells), giriş/çıkış blokları (I/O Block) ve ara bağlantılar olmak üzere 3 kısımdan oluşmaktadır.

5.3.1. Mantık hücresi (Logic-Cell)

Mantık Hücresi bir FPGA 'yı oluşturan ana bileşenlerdir. FPGA 'nın programlandığı işlevi yerine getirmesini sağlayan ana devre elemanıdır. Firmalara göre "mantık birimi", "mantık blokları", "makro hücresi", "dilim" gibi farklı şekillerde isimlendirilebilir, mimari yapıda farklılık gösterse de gerçekleştirdiği fonksiyon temelde aynıdır.

Genel mimari bakımından çoklayıcı (MUX) tabanlı ve Look Up Tablosu (LUT) tabanlı olmak üzere kullanılan iki farklı tip mantık bloğu türü vardır. Günümüz FPGA 'larında LUT tabanlı mantık blokları kullanılmaktadır. Tipik bir mantık bloğu N bit girişli Look-Up tablosu ve D tipi Flip Flop'tan oluşur. LUT, basit bir hafıza birimidir ve bu biriminin büyüklüğü FPGA'nın kapasitesini belirleyen en önemli etken unsurdur. Her LUT birimi n bit veya daha az girişli olan herhangi bir mantık işlemini yerine getirebilir.



Şekil 5.3. Temel Mantık Bloğu İç Yapısı [10]

LUT 'lar aslında bir mantık işlemi yerine getiren küçük belleklerdir. N girişli bir LUT, 2^N 'li bir belleğe işaret eder. Binlerce Mantık Hücrelerinin birleşimi sonucu olarak karmaşık ve büyük programlar oluşturulur. Mantık hücrelerinin ara bağlantıları matris şeklindeki veri yolları ve programlanabilir anahtarlarla sağlanmaktadır.

5.3.2. Matris anahtarlamaalı yönlendirme yolları

Yatay ve dikey olarak iki boyutlu bir düzlem üzerinde mantık blokları arasındaki bağlantıyı sağlayan bir yapıdır. Şekil 5.1 'de görüldüğü gibi her mantık bloğunun etrafını saracak şekilde FPGA 'nın içerisinde yer alır. FPGA 'nın türüne ve üreticisine göre bu yolların büyüklüğü değişim gösterir. Bağlantı noktalarında bulunan programlanabilir statik anahtarlar sayesinde yüklenen programa göre mantık blokları arasında ihtiyaç duyulan şekilde yönlendirme yapar. Bloklar arasında yapılan bağlantıda oluşacak gereksiz bir uzunluk veya yanlış yönlendirme, doğrudan FPGA 'nın hızını etkileyecek bir unsurdur.

5.3.3. Giriş/Çıkış blokları

Mantık bileşenlerinin harici bileşenler ile bağlantı kurması kullanılan yapılardır. Aygıtın bağlantı pin sayısına denk gelecek kadar bulunur. Bir giriş/çıkış bloğu, kaydediciler, kontrol sinyalleri, mux devreleri ve saat sinyallerini içerir [10]. Bir G/Ç bloğu sadece giriş veya sadece çıkış bloğu olarak programlanabilir. FPGA 'nın harici bağlantı noktasında bulunduğu için oluşabilecek elektrostatik deşarja karşı korumalıdır.

FPGA pinleri genel olarak ayrılmış pinler ve kullanıcı pinleri olmak üzere 2 kategoriye ayrılmaktadır. Ayrılmış pinler; FPGA 'ın enerji gereksinimini (güç ve ground (toprak)) sağlayan güç Pinleri, programın FPGA 'e yüklenmesi için kullanılan konfigürasyon pinleri ve saat sinyalleri için özel ayrılmış clock pinleridir. Kullanıcı pinleri; tasarımcı tarafından ayarlanabilen standart giriş/çıkış pinleridir. Giriş, Çıkış, Giriş/Çıkış olarak üç kategoriye ayrılır.

5.3.4. Clock ve global lines

FPGA tasarımları saat tabanlıdır ve FPGA içerisindeki D Flip-Floplar, saat sinyalinin yardımıyla durum değiştirirler. Senkronize olmuş tasarımlarda bir saat sinyalinin, bütün flip-flop'ları aynı anda tetiklemesi gerekir. Aksi takdirde FPGA 'de elektriksel ve zamansal problemler ortaya çıkmaktadır.

5.3.5. RAM blokları

Günümüzde FPGA 'lerde bellek üniteleri yani RAM ayrılmış bir şekilde bulunmaktadır. Bunlar ayrılan bellek üniteleri mantık devrelerinin çalışması sırasında duyulan geçici hafıza gereksinimi için hazırdırlar. Bu bellek üniteleri tekli ve çoklu erişimi destekleyebilirler.

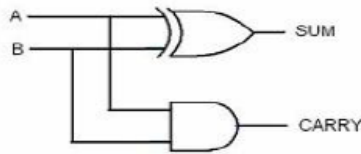
5.4. Programlama ve tasarım

FPGA 'lerin kullanıma hazır hale getirilebilmesi için programlanmaları gerekir. (Altera 'da Quartus II ara yüzü, Xilinx 'de ise ISE Design Suite derleyici programları kullanılır).

FPGA'in içerisine gömülecek devrenin tasarımı 2 yolla yapılmaktadır. Birincisi şematik tasarımla devre elemanlarının teker teker yerleştirilip aralarındaki bağlantıların şematik üzerinde yapılmasıdır. Şematik tasarımda, derleyici programın (ISE, Quartus vs.) kütüphanesinde yer alan araç ve mantık kapıları kullanılarak yapılır. İkincisi ise donanım tanımlama dillerinden (HDL - Hardware Description Language) her hangi bir tanesinin kullanılarak yapılmasıdır. HDL bir donanım parçasını modellemek için kullanılan yazılım dilidir. VHDL ile Verilog en yaygın kullanılan iki türüdür.

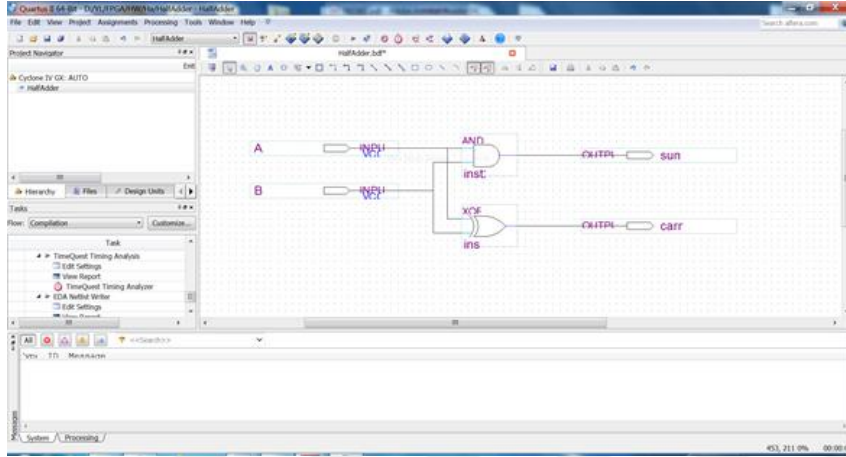
Verilog ile VHDL arasında herhangi bir üstünlük ya da yetersizlik mevcut değildir. Her iki yazılım dilinin de kendi karakteristiğine göre yapısı vardır. Kullanıcı FPGA programlamak için her iki dili de kullanabilmektedir. Program tasarlanırken hem grafik hem de VHDL/Verilog kodu yazılabilir.

Örneğin Şekil 5.4 'de verilen bir adet yarım toplayıcının şematik tasarımı Şekil 5.5 'dedir. Verilog ile tasarımını irdelemek için ayrıca Verilog kodlarının yazılması gerekmektedir.



Şekil 5.4. Yarım Toplayıcı (Half Adder)

Şematik tasarım kullandığımız programda bir tane Ve kapısı ile bir tane Xor kapısı componentler arasından seçilerek gerekli bağlantılar ve pin bağlantıları yapılarak yarım toplayıcı tasarlanabilir.



Şekil 5.5. Half Adder'ın Şematik Tasarımı

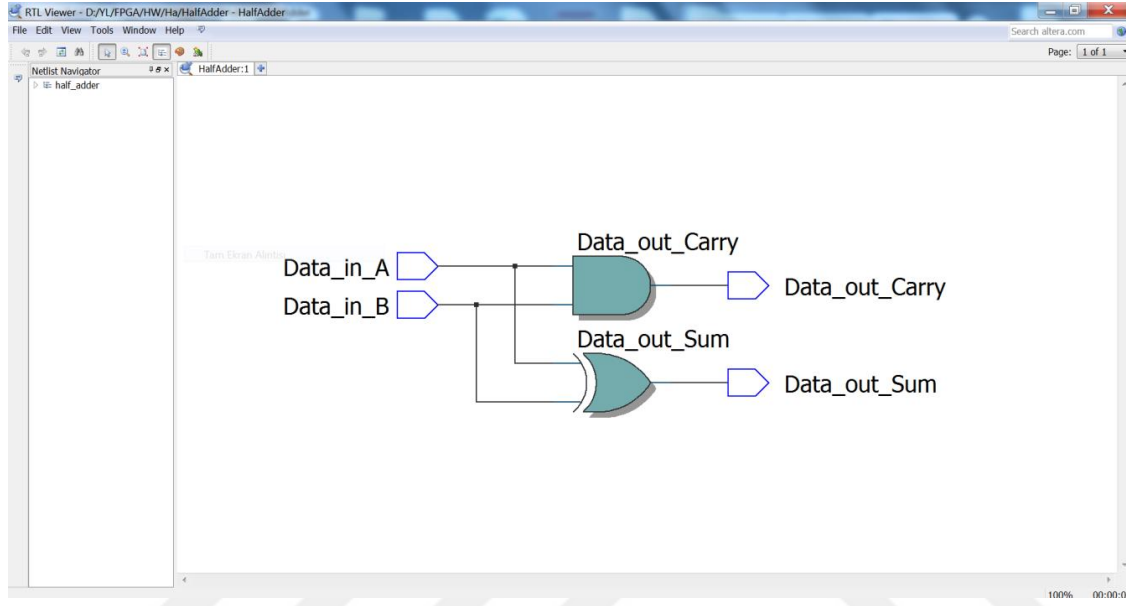
HDL dillerinden birisi olan Verilog kullanarak oluşturmak için aşağıdaki kodları yazmamız gerekecektir.

```
module half_adder(
    Data_in_A,
    Data_in_B,
    Data_out_Sum,
    Data_out_Carry );
    input Data_in_A;
    input Data_in_B;
    output Data_out_Sum;
    output Data_out_Carry;
    assign Data_out_Sum = Data_in_A ^ Data_in_B; //XOR
    assign Data_out_Carry = Data_in_A & Data_in_B; //AND
endmodule
```

Tasarım aşamasından sonra analiz ve sentezleme aşaması gelmektedir. Bu aşamada tercih edilen derleyici program, HDL dili ile yazılan kodu analiz edip, donanımsal olarak gerçekleşip gerçekleşmeyeceğine karar vermektedir. Eğer kod parçasının gerçekleşmesi

mümkün ise kod sentezlenir ve Şekil 5.6'da gösterildiği gibi Register Transfer Level şeması çıkarılır.

RTL; sentezlenen kodun, register cinsinden tasarımında belirtilmesine karşı düşmektedir. Kısacası HDL koduna karşılık gelen lojik kapılarından oluşan devre kısmıdır.



Şekil 5.6. Half Adder'ın RTL Şeması

Program dosyasını oluşturmadan önce tasarımda kısıtlamaların girilmesi gerekir. Bu kısımda pin atamaları belirlenmektedir. Yani çıkış veya giriş olmasını istediğimiz kısmın FPGA 'da hangi pine karşılık geldiği belirtilmektedir.

Yerleşim ve bağlantıların yapılması (Place ve Routing); derleyici ara yüz programı kısıtlamaları da göz önünde bulundurarak tasarıma ait programın yerleştirileceği mantık elemanları belirler. Mantık elemanları ile atanan pinler arasındaki bağlantı bu kısımda yapılmaktadır.

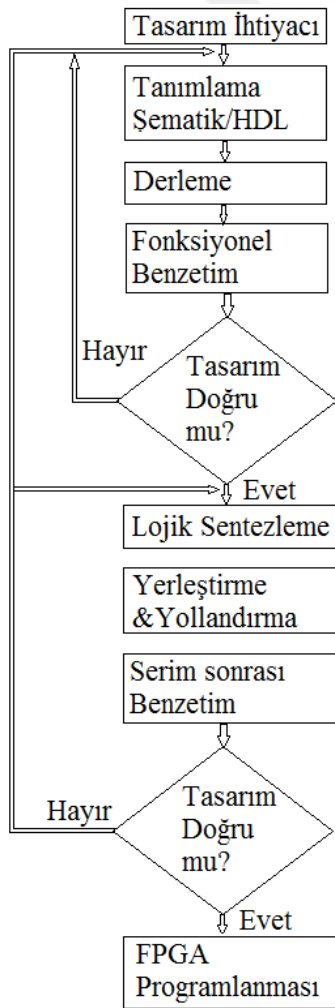
Program dosyasının oluşturulması, yerleşim ve bağlantıların yapılması aşamasından sonra derleyici ara yüz programı, programın yükleneceği FPGA 'lere uygun programlama dosya formatını oluşturmaktadır. Program direkt FPGA 'e yüklenecek ise JTAG uzantılı, konfigürasyon elemanı veya flash kullanılacaksa, bu cihazların desteklediği program uzantılı dosyalar oluşturulmaktadır.

Programın Yükleme; yukarıdaki aşamalardan sonra program dosyası FPGA 'e yongasına veya konfigürasyon elemanlarına aktararak işlemler tamamlanmış olur.

5.4.1. FPGA tasarım akış şeması

Oluşturulan tasarımlar, derlenme boyunca Şekil 5.7 'deki adımlardan geçmektedir.

- Tasarım girişi
- Sentezleme
- İşlevsel benzetim
- Yerleştirme
- Zaman analizi ve benzetimi
- Programlama ve yapılandırma



Şekil 5.7. FPGA Tasarım Akış Şeması

5.5. İkili Sayı Sistemleri

Günlük hayatta sayılar onluk tabanda ifade edilirler. Bu tabanda sayılar 0, 1, 2, 3, 4, 5, 6, 7, 8 ve 9 değerlerini almaktadırlar. İkilik sistemlerde ise sayılar yalnızca 0 ve 1 değerlerini almaktadırlar. İkilik sayı sistemi, sayısal elektronik devrelerinde uygulanabilirliğinin kolay olması açısından vazgeçilmez hal almıştır. Hemen hemen bütün bilgisayar mimarileri bu sayı sistemine göre çalışmaktadırlar.

Temel ikilik sistemlerde iki tabanındaki her kelime pozitif tamsayı olarak yorumlanır. İkili tabanda 32 bit uzunluklu kelimenin değeri olan “S” Eşitlik (5.1)’de gösterilmiştir. Burada en soldaki bit (b_{31}) en fazla değerlikli bit (Most Significant Bit –MSB), en sağdaki bit (b_0) ise en az değerlikli bit (Least Significant Bit –LSB) olarak adlandırılır.

$$S = b_{31} \times 2^{31} + b_{30} \times 2^{30} + \dots + b_1 \times 2^1 + b_0 \times 2^0 \quad (5.1)$$

Ancak kullanılan sistemlerde her zaman pozitif tam sayılarla karşılaşmamaktadır. Negatif tam sayıların kullanılma ihtimali gibi kesirli sayıların da kullanılma ihtimali vardır. Bu durumda kesirli sayıların ikilik sayı sisteminde yazılması ihtiyacı vardır. Bunun için literatürde çeşitli yöntemler vardır. Bu yöntemler genel olarak iki gruba ayrılabilirler. Bunlar sabit noktalı sayı (fixed point number) sistemleri ve kayan noktalı sayı (floating point number) sayı sistemleridir.

5.5.1. Sabit noktalı sayı sistemleri

Sabit noktalı sayı biçimlerinde sayıların bitleri arasına nokta koyulabilmektedir. Böylece kullanılacak sayının büyüklüğü ve hassasiyeti uygulamaya göre değiştirilebilmektedir.

Sabit noktalı sayı sistemlerinin birçok örneği mevcuttur. Ancak hem yaygın kullanılma oranı hem de kullanılan diğer sayı formatı ile karşılaştırılabilme yapılabilmesi adıyla 32 bit kelime uzunluklu I-Q Math adı verilen sabit noktalı sayı formatı anlatılacaktır. Bu sayı formatında sayıların tam kısmı (integer part) “I” ile kesirli kısmı (fractional part) “Q” ile temsil edilmektedir. Kelime uzunluğu 32 bit olan sabit noktalı sayı I8Q24 olarak ifade edilir. Bu gösterimde en değerlikli bit (MSB) işaret biti olarak adlandırılır. “0” ise sayı pozitif, “1” ise negatiftir. I8Q24 sabit noktalı sayı formatındaki N sayısının bit temsili Denklem (5.2)’de verilmiştir.

$$N = b_7b_6b_5 \dots b_1b_0b_{-1}b_{-2} \dots \dots b_{-22}b_{-23}b_{-24} \quad (5.2)$$

Eşitlik (5.2)' de gösterimi verilen N sayısının değeri, Nt tam sayı kısmını ve Nk kesir kısmını temsil etmek üzere:

$$N = Nt + Nk \quad (5.3)$$

olarak verilir. Burada;

$$N_t = \sum_{t=0}^6 b_t \times 2^t \quad (5.4.a)$$

$$N_k = \sum_{k=1}^{24} b_{-k} \times 2^{-k} \quad (5.4.b)$$

olarak tanımlanır. Sayısının işareti ise $(-1)^{b_7}$ ile elde edilir.

Eşitlik (5.3), Eşitlik (5.4a) ve (5.4b) de verilenlere göre örneğin 10'luk tabanda verilen

10.1875 sayısı;

$$10.1875 = 00001010.001100000000000000000000 \quad (5.5)$$

olarak gösterilecektir.

I-Q sayı formatında işlemler yapılırken eğer sayı negatif ise sayının 2'nin tümleyeni alınır ve işlemler 2'lik tabanda pozitif gibi devam ettirilir.

5.5.2. Kayan noktalı sayı sistemleri

Gerçek dünyada sayılarla ilgili işlemler yapılırken bazı durumlarda sonsuza kadar giden değerler ile karşılaşmak mümkündür. Bu tarz işlemleri sayısal donanımlarda gerçekleştirilirken uygulamanın ihtiyaç duyduğu büyüklük ve çözünürlüğe göre yaklaşık değerler ile temsil etmek gerekecektir. Bu temsillerin gerçeğe en yakın şekilde yapılmasını sağlayan sistemler kayan noktalı sayı sistemleridir.

IEEE 754 standartlarında belirlenmiş 4 adet kayan noktalı sayı formatı mevcuttur. Bunlardan en yaygın olarak kullanılanı 32-bit tek duyarlıklı kayan noktalı sayı (32-bit single precision floating point number) sistemidir.



6. VERİLOG DONANIM TANIMLAMA DİLİ

Verilog donanım tanımlama dili sayısal sistemleri tanımlamak için kullanılan bir dildir. Verilog HDL kullanan biri herhangi seviyedeki herhangi bir sayısal donanımı tanımlayabilir. Verilog; C programlama diline çok benzemesi nedeniyle, birçok C kullanıcısı tarafından tercih edilen, ve bu kişiler tarafından kolayca öğrenilebilen bir dildir. Verilog HDL büyük küçük harf duyarlı(case-sensitive) bir dildir. Tüm anahtar sözcükler küçük harfle yazılmaktadır.

Verilog tasarım döngüsü aşağıdaki gibi ifade edilmektedir:

- Belirtilimler (Specifications) ; tasarlanması düşünülen sistemin/tasarımın önemli parametrelerinin tanımlandığı adımdır.
- Yüksek seviyeli tasarım (High level design) ; birçok blok ve bunlar arasındaki haberleşmenin tanımlandığı adımdır.
- Düşük seviyeli tasarım(Low level (micro) design) ; her bir blok uygulandığında nasıl tanımlanacağı belirtilir. Ayrıntı içerir. Her zaman, farklı ara yüzlerde dalga formu çizmek gerekebilmektedir. Bu adım çok zaman harcanan bölümlerden biridir.
- RTL kodlama; mikro tasarımda dilin sentezlenebilir yapıları kullanılarak Verilog/VHDL kodun dönüştürülmesi aşamasıdır.
- Doğrulama (Verification) ; herhangi bir seviyedeki soyutlamanın fonksiyonel karakteristiğinin simülasyon ortamında test edilmesine denir. Simülatörler donanım modellerini doğrulamak için kullanılmaktadır. RTL kodun belirtilen fonksiyonel gereklilikleri karşılayıp karşılamadığını test etmek için tüm RTL bloklarının fonksiyonel olarak doğruluğu kontrol edilmelidir. Bunu yapmak için bir clk(saat), reset(yeniden başlatma) ve gerekli test vektörlerini üretecek bir testbenç yazmamız gerekmektedir. Test altındaki aracın (DUT (Device Under Test)) fonksiyonel olarak doğru çalışıp çalışmadığını anlamak için simülatörden çıkan dalga formu kullanılmaktadır. Bu alanda başka bir simülasyon daha vardır ve buna zamanlı simülasyon(timing simulation) denir. Bu simülasyon Sentez veya Alan ve Rota(P&R-Place and Route)'dan sonra yapılır. Burada kapı gecikmeleri ve tel gecikmeleri eklenmektedir ve test altındaki tasarımın beklenen saat hızında çalışıp çalışmadığı incelenmektedir. Buna aynı zamanda SDF simülasyon veya kapı seviyesi simülasyon da denmektedir.

- Sentez (Synthesis); tasarım derleyici veya Synplify gibi sentez araçlarının Verilog veya VHDL ‘deki RTL’den aldıkları, hedef teknolojiyi, girişlere göre kısıtlamak ve RTL’i hedef teknolojinin kısıtlarıyla eşleştirmektir. Sentez araçları, RTL’i kapılarla eşleştirdikten sonra eşleştirilmiş tasarımın zamanlama ihtiyacına uyup uymadığını en düşük miktarda zamanlama analizi ile yapmalıdır.

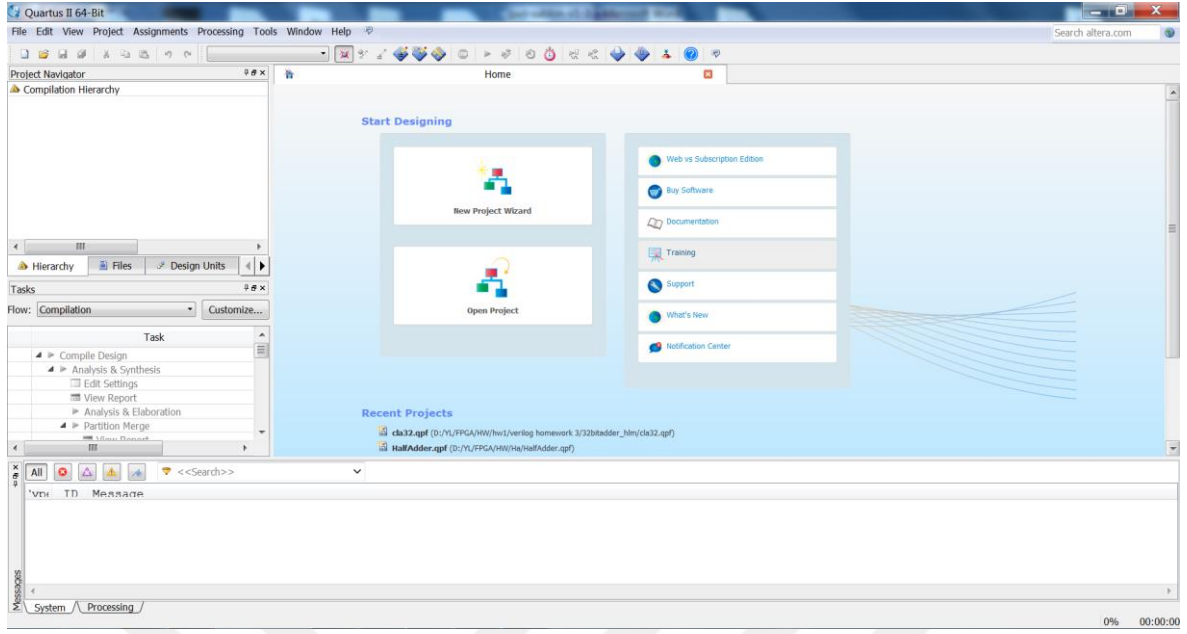
6.1. Modüller(Modules)

Verilog programlama dili, sayısal bir sistemi, modül setleri ile tanımlar. Bu modüller birbirleriyle tasarımı yapan kişinin tanımladığı kurallar çerçevesinde ilişkilidir. Modüller, giriş/çıkış portları ve içerik kısmından oluşur. Bu içerik kısmı giriş sinyalleri ile çıkış sinyallerini belirlemektedir. Modülün içeriği üç değişik tarzda ifade edilebilir:

- Yapısal (Düşük seviyeli lojik ve alt modüller ile),
- Veri akışı ile (Çıkış sinyalleri giriş sinyallerinin bir fonksiyonu olarak ifadesi ile)
- Davranışsal olarak (Devreden beklenen davranışın programsal ifadesi şeklinde ifade edilir. Tasarlanmak istenen sistemin davranışı «always» blokları içinde programlama dillerinde de kullanılan «if/else» veya «case» gibi yapılar ile ifade edilerek yapılır. Always içerisinde sadece reg tipinde sinyallere atama yapılabilir.)

Bu üç stil, üst seviyeli modüllerde soyutlamaya gidilebilmesini ve alt seviyeli modüllerinde de donanımı daha yakın ve verimli olarak ifade etme imkânı sağlar. Modül içinde birden fazla tarz ifade beraber kullanılabilir.

Bu tezde yazılan Verilog kodları, Altera Quartus 13.1 (Resim 6.1) Web versiyonu kullanılarak derlenmiştir.

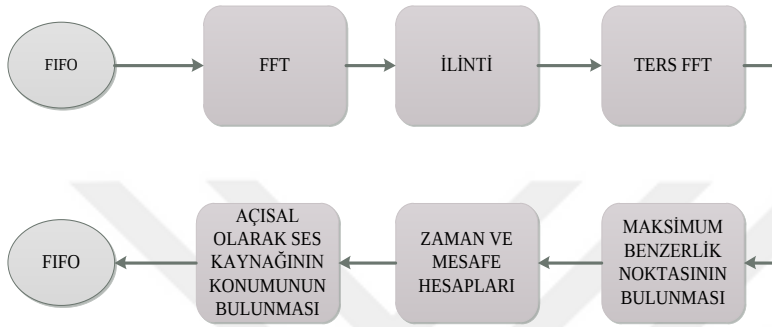


Resim 6.1. Kullanılan derleyici Altera Quartus 13.1 Web Edition



7. DONANIM MİMARİSİ

FPGA ile ses konum tespiti gerçekleştirebilmek için öncelikle geliştirme kartından ses girişinin sağlanması gerekmektedir. Mikrofondan gerçek zamanlı olarak veya kaydedilmiş bir ses verisi üzerinde ses işleme yapabilmek için gerekli donanım tasarlanmalıdır. Ses girişi sağlandıktan sonra sırası ile aşağıdaki işlemlerin gerçekleştirilmesi gerekmektedir:



Şekil 7.1. Ses konum tespiti donanımının temel mimarisi

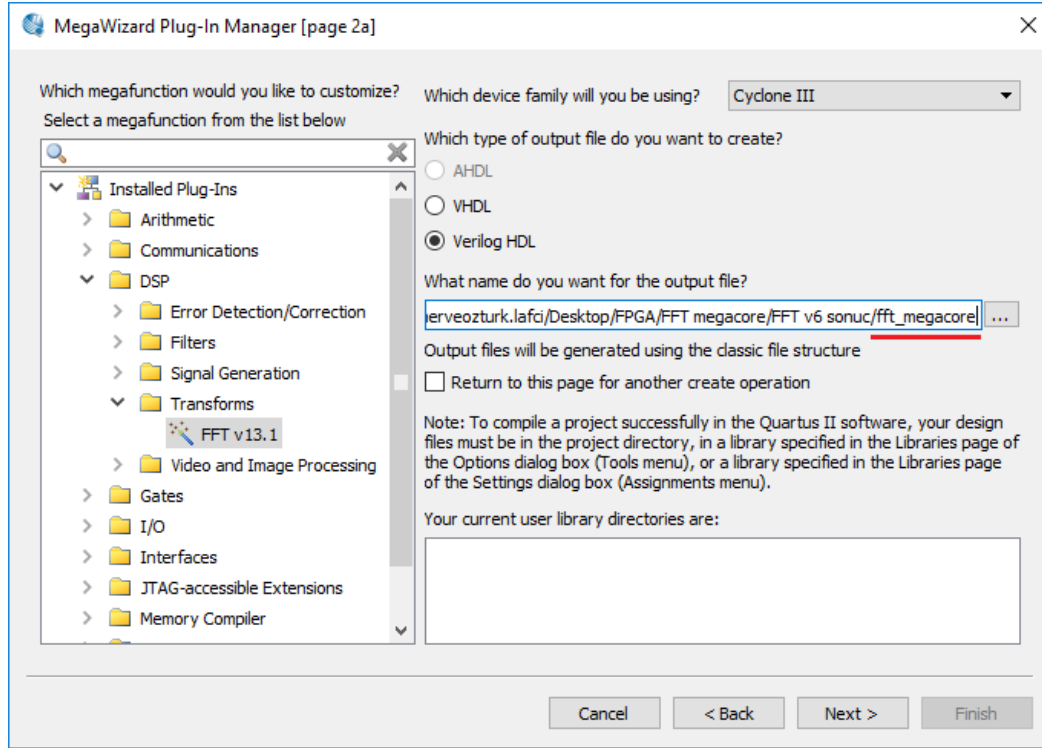
7.1. HFD ve Ters HFD

Her iki mikrofondan gelen ses işaretinin ayrık Fourier dönüşümünün (AFD) hesaplanabilmesi için Altera firmasının sağlamış olduğu hızlı Fourier dönüştürücü (HFD) öbeği kullanılmıştır. Bu öbek hem HFD hem de ters HFD (THFD) işlemini gerçekleştirebilmektedir.

HFD matematiksel olarak son derece yoğun işlemlerin olduğu bir bloktur. Bu nedenle bu bloğun Verilog kodunun yazılması yerine “Quartus MegaWizard Plug-In Manager” aracının sunduğu “FFT MegaCore” kütüphanesi kullanılmıştır.

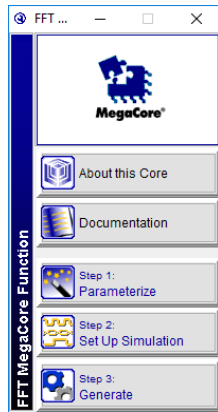
7.1.1. FFT megacore projeye eklenmesi

MegaWizard Plug-In Manager aracının kullanıcının ihtiyacının olduğu kütüphaneyi seçmesine olanak sağlayan ara yüz Resim 7.1’de gösterilmektedir.



Resim 7.1. MegaWizard Plug-In Manager’a ait ekran görüntüsü

FFT Core parametre ayarlarının yapılması için yeni bir pencere açılmaktadır. Bu pencereden ilk önce “step 1(Parameterize)” ile parametre ayarları yapılmalıdır.



Resim 7.2. FFT Megacore ayarlarının yapılması

Açılan yeni ekrandan kullanılacak ürün ailesi seçilir. Proje gereksinimlere göre veri uzunluğu ve her bir verinin bit uzunluğu ayarlanır.

Parameterize - FFT MegaCore Function

Parameters Architecture Implementation Options

Target Device Family Cyclone III

Transform Length 1024 points

Data Input Precision 20 bits

Twiddle Precision 20 bits

Data Output Precision 29 bits

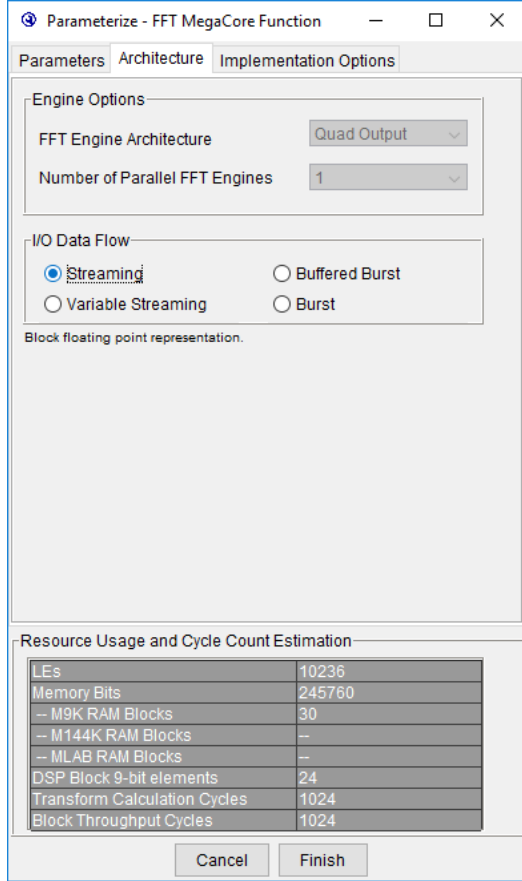
Resource Usage and Cycle Count Estimation

LEs	10236
Memory Bits	245760
-- M9K RAM Blocks	30
-- M144K RAM Blocks	--
-- MLAB RAM Blocks	--
DSP Block 9-bit elements	24
Transform Calculation Cycles	1024
Block Throughput Cycles	1024

Cancel Finish

Resim 7.3. FFT Megacore parametre ayarları

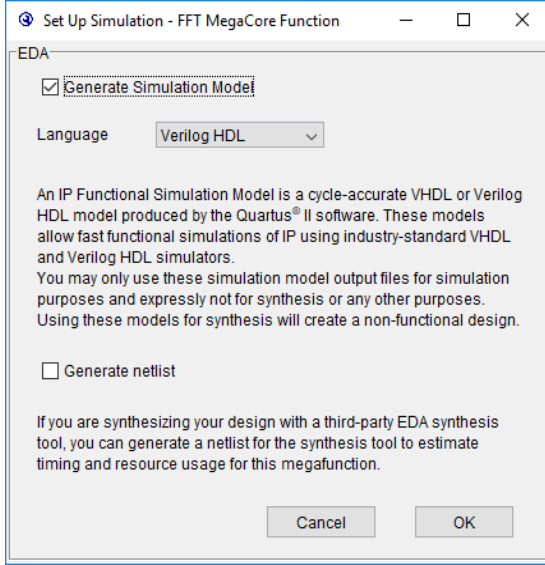
FFT Megacore için Architecture penceresinden FFT mimari ayarları yapılır.



Resim 7.4. FFT Megacore mimari ayarları

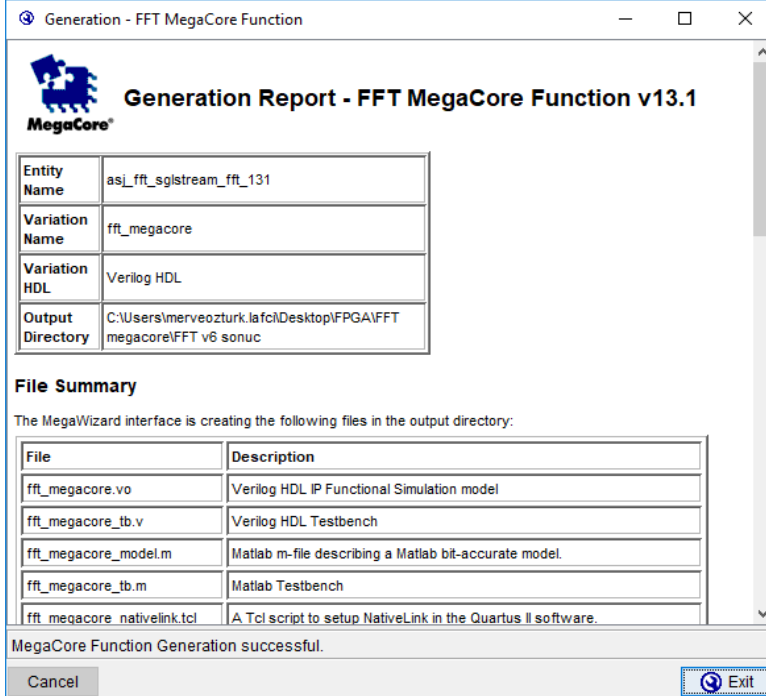
Implementation Options penceresinden tercih edilen mimari yapısı seçilmektedir. Tasarıya global saat etkinleştirmeyi eklemek için Global Clock Enable'ı açılabilir. Spesifik hafıza ayarlamaları yapabilir. Gerekli düzenlemelerden sonra Finish ile ayarlamaları kaydedilerek bu pencere kapatılmaktadır.

Şekil 7.2 'deki pencere sayfasından “step 2(Set Up Simulation)” kısmından simülasyon dosyalarının üretilmesi için ayarlar yapılmaktadır. Testbench in yazılacağı dilin seçilmesi bu pencereden yapılmaktadır.



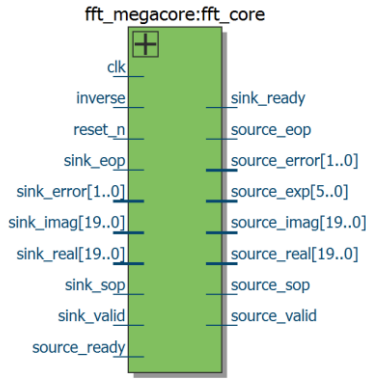
Resim 7.5. FFT Megacore Simulasyon Ayarları

Resim 7.2 'deki pencere sayfasından “step 3(Generate)” kısmından kod üretilmektedir. Başarılı bir kod üretimi sonucunda çıkan pencereden Exit ile Megacore üretim sayfası kapatılmaktadır.



Resim 7.6. FFT Megacore raporu

FFT Megacore üretildikten sonra kodun derlenmesi gerekmektedir. Quartus RTL Viewer kısmından RTL şemasına ulaşılmaktadır. Resim 7.2 'de kullanılan HFD öbeğinin RTL şeması görülmektedir.



Şekil 7.2. Kullanılan HFD öbeğinin sinyalleşme ara yüzü

Reset_n sinyali mantıksal 1 olduktan sonra veri kaynağı, geçerli veri girişi için uygun olduğunu HFD işlemine göstermek için sink_valid ögesinde mantıksal 1 çıkışı görülür. Hem sink_valid hem de sink_ready sinyallerinde mantıksal 1 değeri görüldüğünde başarılı bir veri aktarımı gerçekleşir. Veri aktarımı tamamlandığında, sink_sop mantıksal 0 değerini gösterir ve veri örnekleri doğal sırada yüklenir. HFD, giriş bloğunun dönüşümünü tamamladığında, source_valid ögesine mantıksal 1 değerini atar ve karmaşık dönüşüm alanı veri bloğunu doğal sırada gönderir. HFD işlevi, ilk çıktı örneğini belirtmek için source_sop ögesine mantıksal 1 seviyesini atmaktadır [12]. HFD öbeğindeki sinyaller ve açıklamaları Çizelge 7.1’de verilmiştir.

Çizelge 7.1. FFT megacore sinyalleri ve açıklamaları

Sinyal İsmi	Yön	Boyut	Açıklama
clk	Giriş	1	Tüm dahili fft bileşenlerinin saat sinyalıdır.
inverse	Giriş	1	1 ise ters FFT, 0 ise FFT hesaplaması yapılır.
reset_n	Giriş	1	Aktif-düşük eş zamansız sıfırlama sinyalıdır. Sinyal clk'nin yükselen kenarında algılanır ve en azından bir clk saat devri ağlanmalıdır.
sink_eop	Giriş	1	Gelen FFT çerçevesinin sonunu gösterir
sink_error	Giriş	2	Avolon-ST protokolünün yasadışı bir şekilde kullanılması nedeniyle üst bir modülde bir hata olduğunu gösterir. Aşağıdaki hatalar tanımlanır: 00 = hata yok 01 = eksik paket başlangıcı (SOP) 10 = eksik paket sonu (EOP) 11 = beklenmedik EOP Bu sinyal üst modüllerde kullanılmazsa, sıfıra ayarlayın.
sink_imag	Giriş	Veri genişlik hassasiyeti[20]	Sanal giriş verileri, veri bitleri hassasiyetinin sayısını temsil eder.

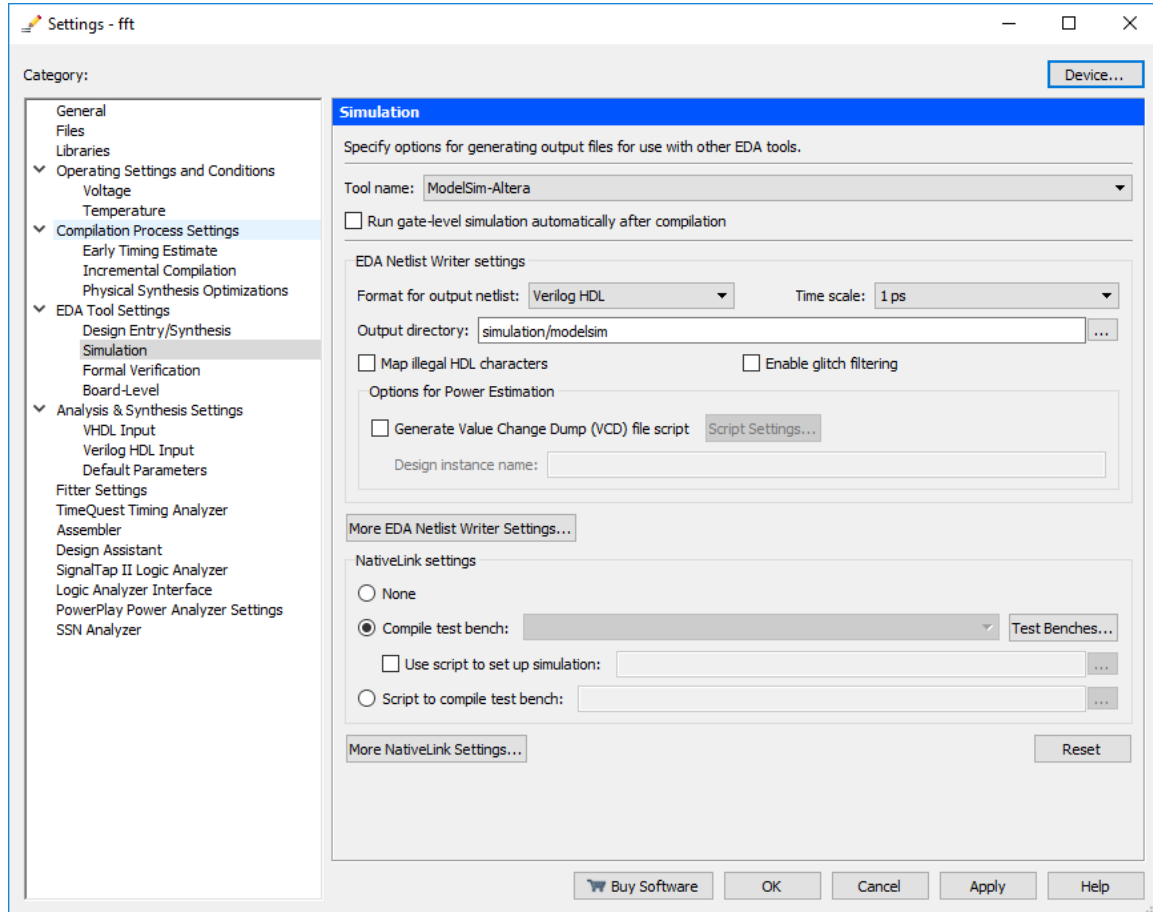
Çizelge 7.1. (Devamı) FFT megacore sinyalleri ve açıklamaları

sink_real	Giriş	Veri genişlik hassasiyeti[20]	Gerçek giriş verileri, veri bitleri hassasiyetinin sayısını temsil eder.
sink_sop	Giriş	1	Gelen FFT çerçevesinin başlangıcını gösterir.
sink_valid	Giriş	1	Veri yolundaki veriler geçerli olduğunda onaylanır. sink_valid ve sink_ready belirtildiğinde, bir veri aktarımı gerçekleşir.
source_ready	Giriş	1	Veri kabul edilebiliyorsa, alt modül tarafından onaylanır.
sink_ready	Çıkış	1	Veriyi kabul edebildiğinde FFT tarafından onaylanır. Hazır çevrimler sırasında FFT'ye bilgi vermek zorunlu değildir.
source_eop	Çıkış	1	Giden FFT çerçevesinin sonunu belirtir. Yalnızca source_valid onaylandığı zaman geçerlidir.
source_error	Çıkış	2	Bir üst modülde veya bir FFT modülünde hata olduğunu gösterir.
source_exp	Çıkış	6	Yalnızca streaming, burst, ve buffered burst mimarileridir. işaretli blok üssü: FFT hesaplaması sırasında dahili sinyal değerlerinin ölçeklenmesi için hesaplanır.
source_imag	Çıkış	Veri genişlik hassasiyeti + büyüme	Sanal çıkış verileridir. Burst, buffered burst, and streaming FFT'ler için çıkış verisi genişliği girdi verileri genişliğine eşittir. variable streaming FFT'ler için çıktı verisinin boyutu, FFT için tanımlanan aşamaların sayısına bağlıdır.
source_real	Çıkış	Veri genişlik hassasiyeti + büyüme	Gerçek çıkış verileridir. Burst, buffered burst, and streaming FFT'ler için çıkış verisi genişliği girdi verileri genişliğine eşittir. variable streaming FFT'ler için çıktı verisinin boyutu, FFT için tanımlanan aşamaların sayısına bağlıdır.
source_sop	Çıkış	1	Giden FFT çerçevesinin başlangıcını işaretler. Yalnızca source_valid onaylandığında geçerlidir.
source_valid	Çıkış	1	Çıktı için geçerli veri olduğunda FFT tarafından onaylanır.

7.1.2. FFT megacore simülasyon ayarları

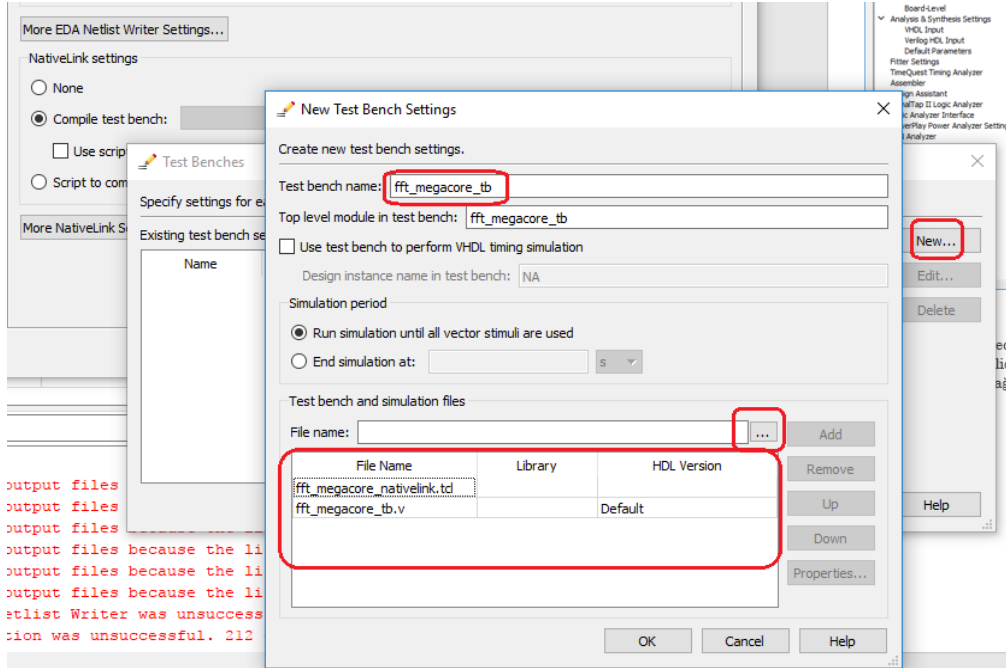
Bu bölümde FFT Megacore simülasyonu için gerekli olan konfigürasyonlar anlatılmaktadır.

1. Assignments->Settings kısmından Category bölümünden Eda Tool Settings kısmından Simulation ayarlarına gidilmelidir. Tool name, Format for output netlist seçenekleri Resim 7.7’de gösterildiği gibi belirlenmelidir.



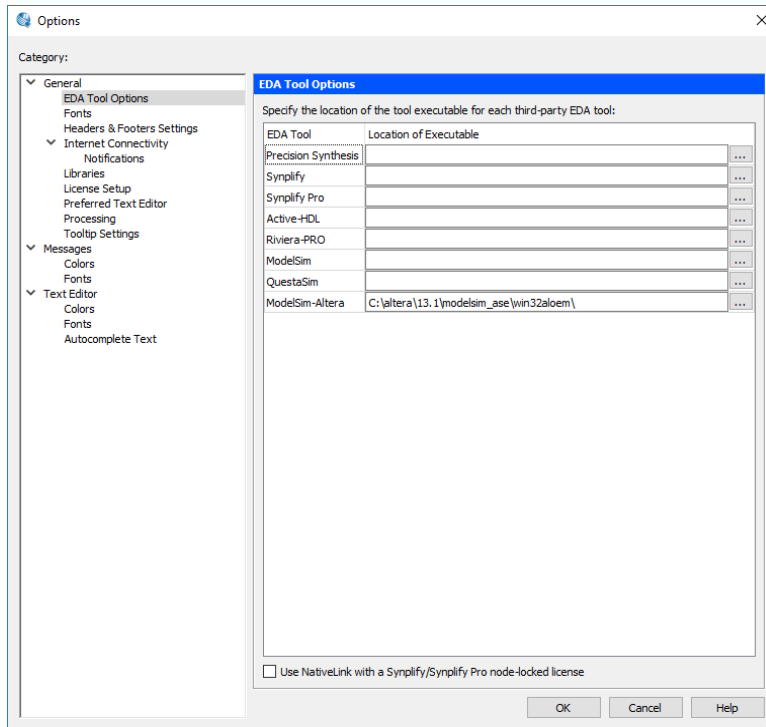
Resim 7.7. Eda Tool simulation ayarları

2. Aynı pencereden NativeLink settings bölümünden compile test bench işaretlenmelidir. Test Bences... kısmından FFT Megacore un testbench lerinin eklenmesi sağlanmalıdır. <Megacore ismi>_nativelink.tcl ve <Megacore ismi>_tb.v dosyaları testbench and simulation files kısmına eklenmelidir. OK ile bu pencereler kapatılmalıdır.



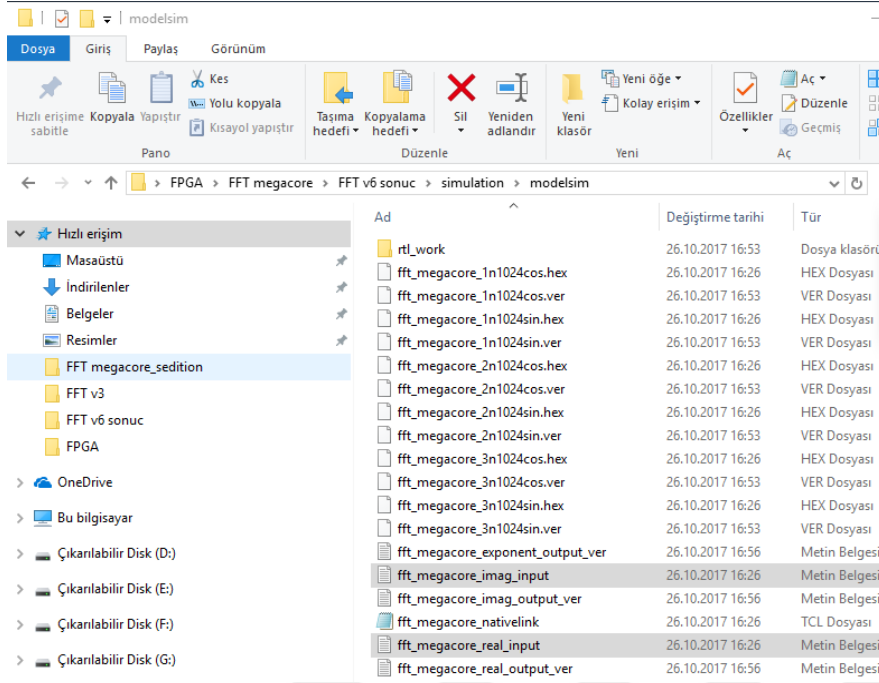
Resim 7.8. Test bench ve simulasyon dosyalarının eklenmesi

3) Tools-> Options kısmından General başlığı altında Eda Tool Options sekmesinde simülasyon yolunun doğru olarak eklendiğinden emin olunmalıdır. Modelsim-Altera seçeneğinde çalışan uygulama dosyasının yolu “C:\altera\13.1\modelsim_ase\win32aloem\” olarak yazılmaktadır.



Resim 7.9. Eda Tool Seçenekleri Modelsim-Altera dosya yolu

4) FFT Megacore sanal ve reel giriş dosyaları testbencin kullanacağı dosyalar olduğundan bu dosyalar modelsim klasörüne kopyalanmalıdır.



Resim 7.10. Test bench içinde kullanılan FFT giriş dosyalarının simülasyon klasörüne kopyalanması

5) Tools-> Run Simulation Tool-> RTL Simulation Tool kısmından Modelsim-Altera ya bağlanılmaktadır.

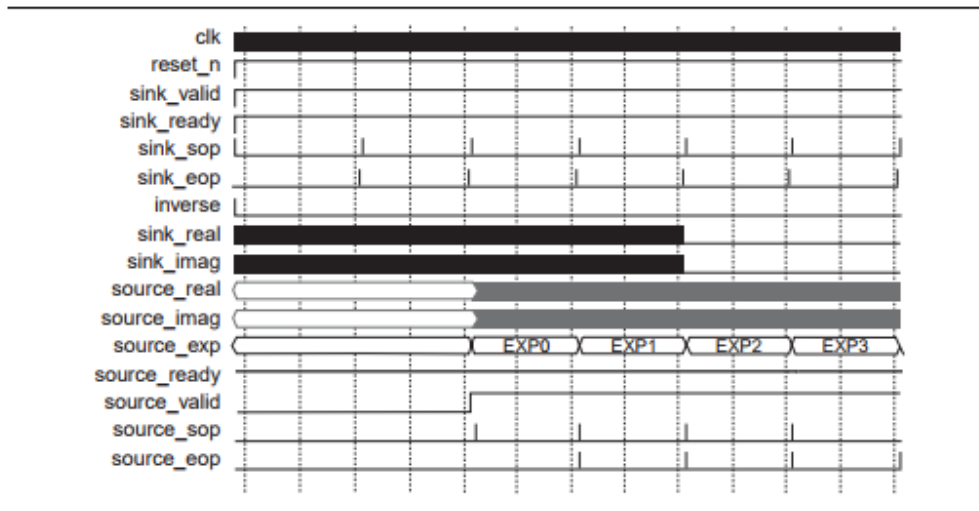
7.1.3. Simülasyon sonuçlarının değerlendirilmesi

FFT Megacore Mimari ayarlarında oluşturulurken, Streaming mimari seçeneği tercih edilmiştir. Simülasyon sonuçlarını doğru değerlendirebilmek için FFT Megacore User Guide referans kaynağından bu mimarinin I/O veri akış mimarisinin nasıl çalıştığı analiz edilmiştir.

Streaming G / Ç veri akışı FFT mimarisi, giriş verisinin sürekli olarak işlenmesine izin verir ve FFT işlevinin içerisindeki veya dışındaki veri akışını durdurmak zorunda kalmadan sürekli bir karmaşık veri akışı sağlar. Resim 31 ve Resim 32’de örnek bir simülasyon dalga şekli gösterilmektedir.

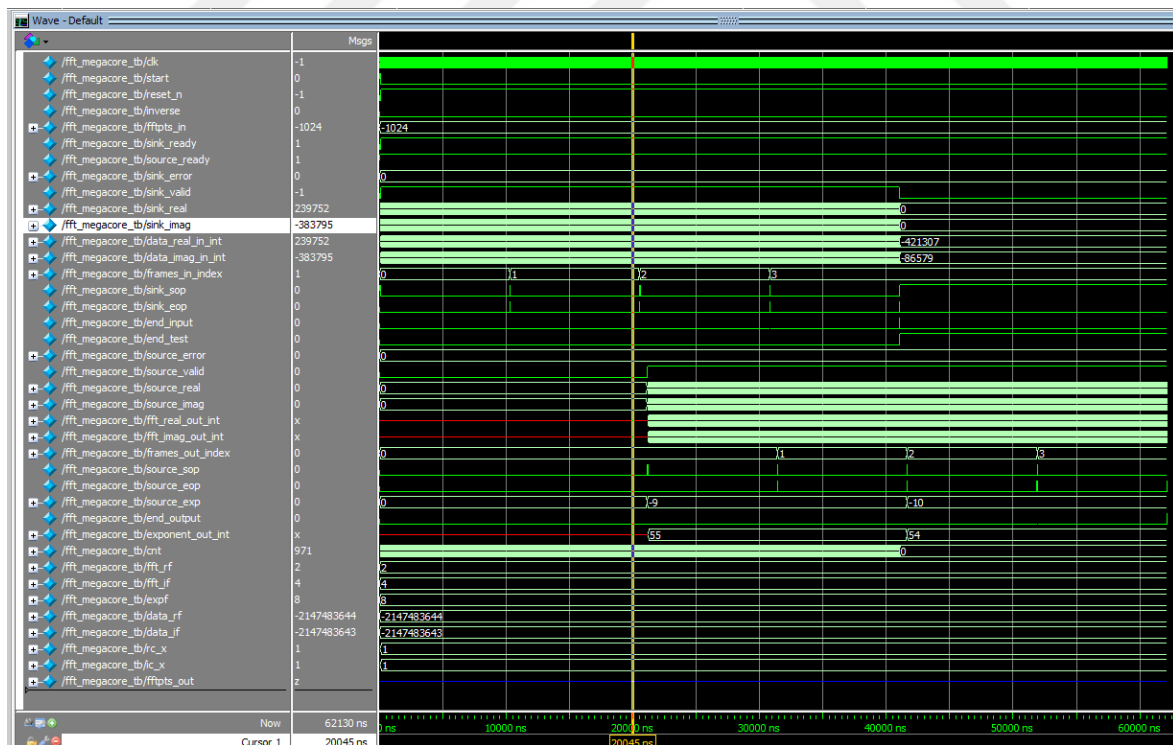
Sistem reset işleminin onaylanmasını takiben veri kaynağı, geçerli veri girişi için uygun olduğunu FFT işlevine göstermek için sink_valid ögesini teyit(1) eder. Hem sink_valid

hem de sink_ready atandığında(1) başarılı bir veri aktarımı gerçekleşir. Veri aktarımı tamamlandığında, sink_sop açık gösterilir(0) ve veri örnekleri doğal sırada yüklenir.



Şekil 7.3. FFT Streaming Veri Akışı Mimarisi Simülasyon Dalga Formu [10]

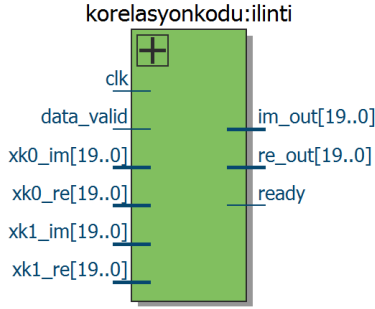
Streaming FFT İşlemleri değerlendirilerek Modelsim de alınan sonuçların doğruluğu teyit edilebilmektedir.



Şekil 7.4. Modelsim Simulasyon Sonuçları

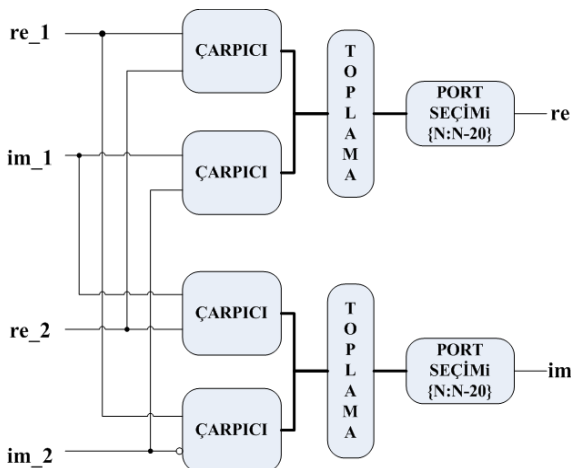
7.2. Çapraz İlinti

Sayısal ses işareti için örnek uzayındaki bağıl gecikme miktarı çapraz ilintisi ile bulunur. İlinti için clk, data_valid, xk0_im, xk0_re, xk1_re, xk1_im, re_out, im_out, ready sinyallerini içeren bir öbek tasarlanmıştır. Resim 21 'de bu öbeğin sinyalleşme ara yüzü görülmektedir.



Şekil 7.5. Kullanılan çapraz ilinti öbeğinin sinyalleşme ara yüzü

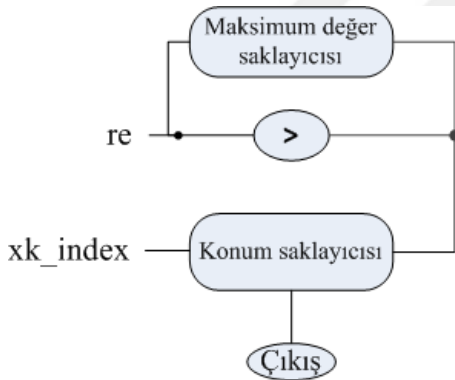
İlinti işlemlerinin başladığının bilgisini ters HFD öbeğine gönderebilmek için ready sinyali kullanılmaktadır. İlintiye gelen veriler HFD işlemi gerçekleştikten sonra geçerli veri olduğunu belirten source_valid sinyali mantıksal 1 seviyesine çekilerek HFD öbeğinin çıkışından alınmaktadır. Bu nedenle ilinti hesaplama öbeğinde saatin her yükselen kenarında bu source_valid sinyalinin durumu kontrol edilir. İlinti öbeğinde HFD öbeğinin iki kanalından alınan, gerçek ve sanal kısımlardan oluşan dört adet giriş verisi saatin her yükselen kenarında işlemlere tabi tutulur. Eşleniğini hesaplamak için ikinci kanalın sanal kısmının ikiye tümleyenini alınıp bir eklenir. Sonra çarpma ve toplama işlemleri uygulanır. İlinti sonucunda gerçek ve sanal olmak üzere bir sonuç elde edilir.



Şekil 7.6. İlinti öbeğinin mimarisi

7.3. Maksimum Noktanın Bulunması

İlinti işleminin sonuçları ters HFD işlemine tabi tutulduktan sonraki aşamada, çıkan sonuçtaki en büyük değerin hangi adreste olduğunu bulma işlemi yapılmaktadır. source_valid sinyali kontrol edilerek mantıksal 1 olduğunda, her yükselen kenarda alınan ters HFD sonuçlarının gerçek kısmı üzerinde tepe noktasını bulma işlemi yapılır. Bu işlem için değeri sıfır olarak ilklendirilmiş ve re_buyuk olarak isimlendirilmiş bir yerel kaydedici tanımlanmıştır. Gelen her yeni veri bu kaydedicinin içerdiği sayıyla karşılaştırılarak bu değer güncellenmektedir. Bütün veriler bittiğindeki re_buyuk değeri maksimum nokta olarak bulunmuştur. Bu öbeğin içerisine bir de her veri geldiğinde değeri bir artan sayıcı yerleştirilmiştir. Bu sayıcının değeri en büyük veri bulunduğunda kaydedilerek en büyük verinin yerinin bilgisi re_buyuk_yeri olarak bulunur. İşlemlerin sonunda maksimum noktanın bulunduğu bilgisini bir sonraki bloğa bildirmek için ise re_buyuk_hazir sinyali 1'e çekilir.



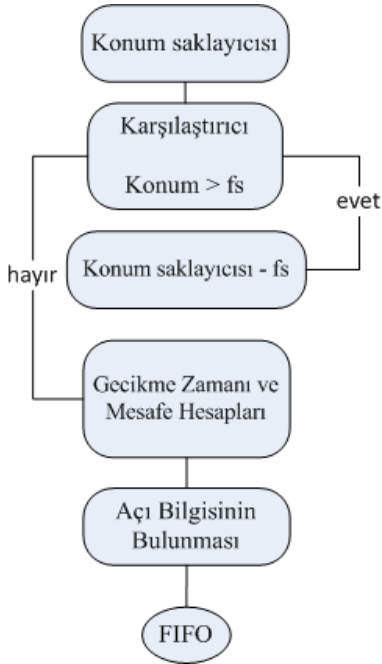
Şekil 7.7. Maksimum bulma öbeğinin mimarisi

7.4. Zaman ve Mesafe Hesapları

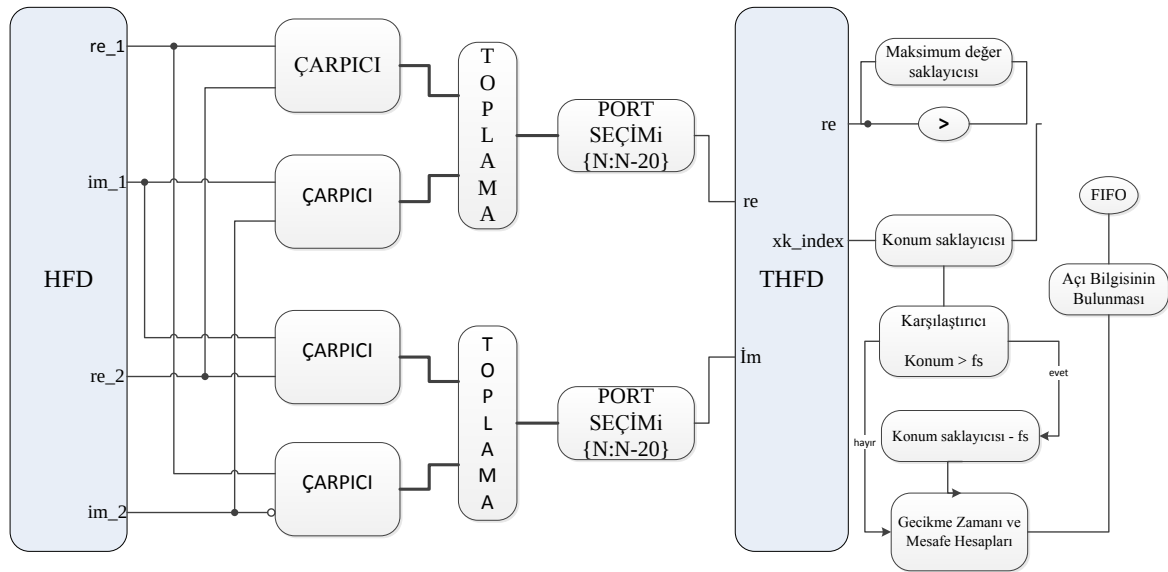
Maksimum noktanın kaçınıcı örnekte elde edildiği bulunduktan sonra ki aşamalar ise mesafe ve zaman hesaplarıdır. Maksimum noktanın $fs/2$ ile kontrolü yapılır. Eğer bulunan nokta $fs/2$ den büyükse bu noktanın değeri fs den çıkarıp maksimum noktanın değeri güncellenir. Böylece işlem için daha uygun hale getirilmiş olur. Ardından güncellenen maksimum noktayı fs 'e bölerek de gecikme miktarının zamandaki karşılığı ITD bulunur ve $m=ITD.V_{ses}$ eşitliğinde de görüldüğü gibi ses hızını da kullanarak mesafe bulunur.

Bu mesafe bilgisinin iki mikrofon arası uzaklığa(D_m) oranından açı bilgisi bulunur. Sesin gecikme miktarı ve mikrofonlar arası uzaklık bilgileri elde edildikten sonra mikrofonlara

gelen sesin konumu 14 numaralı denklem kullanılarak tespit edilmiş olmaktadır. Q ses kaynağının konumunu açısal olarak vermektedir. Bunun FPGA ile gerçekleştirilebilmesi için sayısal koordinat dönüşümü bilgisayar (COordinate Rotation DIgital Computer-CORDIC) veya okuma tablosu (Look Up Table - LUT) kullanılabilmektedir. Bu çalışmada daha az yer kapladığı için LUT kullanımı tercih edilmiştir.



Şekil 7.8. Uzaklık ve Açı Bulma Öbeğinin Mimarisi



Şekil 7.9. Ses Konum Tespiti Donanım Mimarisi

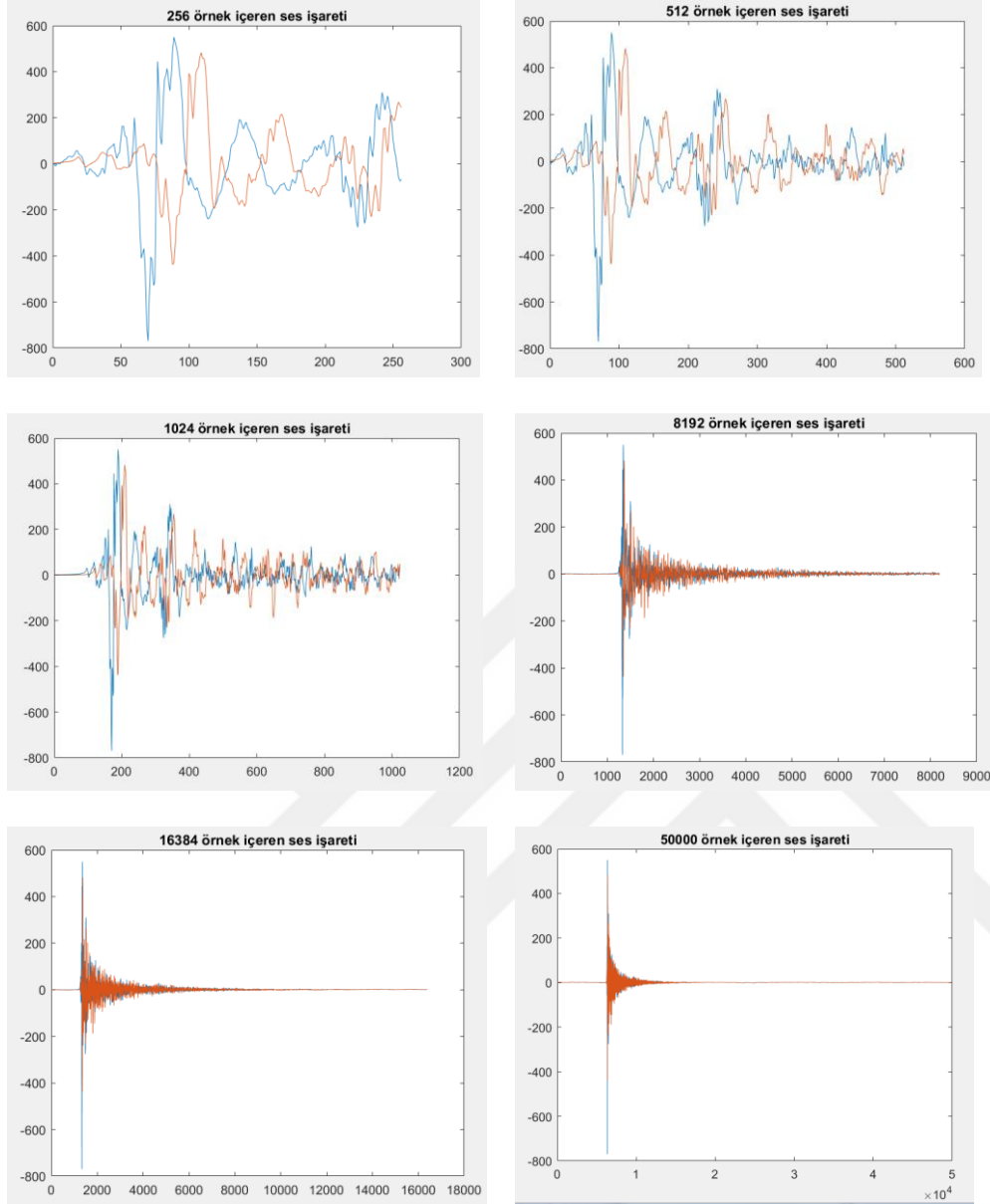


8. DENEYSEL SONUÇLAR

Sayısal ses işareti için örnek uzayındaki bağıl gecikme miktarını bulmak için kullanılan çapraz ilinti yöntemi Verilog donanım tanımlama dili ile geliştirilmiştir. Bu donanım mimarisinin doğruluğunun test edilmesi için aralarında 62cm mesafe bulunan iki mikrofondan alınan ses kayıtları kullanılmıştır. Ses kayıtları mikrofonlara göre farklı yönlerden alınmıştır. Farklı açılardan el çırpma sesleri kayıt edilerek sesler arasındaki gecikmeler MATLAB ortamında incelenmiştir. Mevcut ses kayıtları 50000 örnek içeren ses kayıtlarıdır. Ses kayıtlarında örnekleme frekansı 50kHz dir ve 1 sn'lik ses kayıtlarından oluşmaktadır.

Ses işareti pencereleme yöntemi ile işlenmektedir. Pencere boyutunu belirlemek için farklı pencere kanalları kullanarak algoritmanın çalışma performansı test edilmiştir. Pencere boyutu 50000, 16384, 8192, 1024, 512 ve 256 örnek içeren ses verileri kullanılarak gecikme ve açı hesaplamaları yapılmıştır. Şekil 8.1'de pencere boyutları 50000, 16384, 8192, 1024, 512 ve 256 örnek içeren ses verileri görülmektedir. Pencere boyutunun küçük olması donanım üzerindeki hafıza gereksinimini azaltmaktadır. Pencere boyutu daraldıkça ses verisinin içerdiği veri miktarı azalmaktadır. Bununla birlikte çok küçük pencere boyutlarının alınması sistemin çalışma performansını olumsuz yönde etkilemektedir.

İkinci mikrofondan alınan ses işaretinin birinci mikrofondan alınan ses işaretine göre ileride veya geride olması durumuna göre gecikme hesabı negatif veya pozitif çıkmaktadır. Bu sayede ses kaynağının sağ tarafta veya sol tarafta olduğu bilgisine ulaşılabilir.



Şekil 8.1. Pencere boyutu 50000, 16384, 8192, 1024, 512 ve 256 örnek içeren ses verileri

Pencere boyutu 50000, 16384, 8192, 1024, 512 ve 256 örnek içeren ses verileri ile sadeleştirilmiş çapraz ilinti yöntemi kullanılarak gecikme hesaplamaları Çizelge 8.1’de verilmiştir. Pencere boyutu 512 ve 256 örnek içeren ses verilerinde çalışma performansının olumsuz yönde etkilendiği görülmektedir.

Çizelge 8.1. Pencere boyutu 50000, 16384, 8192, 1024, 512 ve 256 örnek içeren ses verileri ile sadeleştirilmiş çapraz ilinti yöntemi kullanılarak gecikme hesaplamaları

SES KAYITLARI	50000 örnek ile	16384 örnek ile	8192 örnek ile	1024 örnek ile	512 örnek ile	256 örnek ile
Kayıt 1	-17	-17	-17	-17	-17	-17
Kayıt 2	66	66	66	66	66	66
Kayıt 3	57	57	57	57	57	62
Kayıt 4	-1	-1	-1	-1	-1	-2
Kayıt 5	10	10	10	10	11	11
Kayıt 6	67	67	67	67	67	67

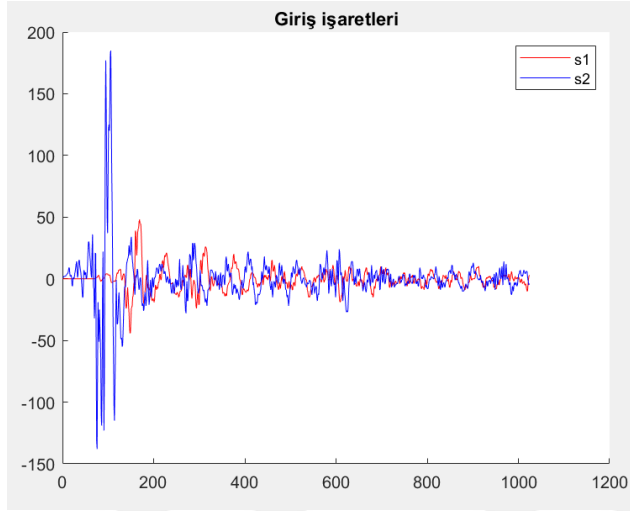
Pencere boyutu 50000, 16384, 8192, 1024, 512 ve 256 örnek içeren ses verileri ile sadeleştirilmiş çapraz ilinti yöntemi kullanılarak gecikme hesaplamaları çizelge 8.2’de verilmiştir. Pencere boyutu 512 ve 256 örnek içeren ses verilerinde çalışma performansının olumsuz yönde etkilendiği görülmektedir. Açık farklarının gecikme hesabındaki farktan kaynaklandığı görülmektedir. Çizelge 8.1’de gecikme hesaplarının bulunduğu tablo incelendiğinde bu farkların çizelge 8.2’de açık hesaplamalarındaki fark ile paralel olarak ortaya çıktığı görülmektedir.

Çizelge 8.2. Pencere boyutu 50000, 16384, 8192, 1024, 512 ve 256 örnek içeren ses verileri ile sadeleştirilmiş çapraz ilinti yöntemi kullanılarak açık hesaplamaları

SES KAYITLARI	50000 örnek ile	16384 örnek ile	8192 örnek ile	1024 örnek ile	512 örnek ile	256 örnek ile
Kayıt 1	-10.5540	-10.5540	-10.5540	-10.5540	-10.5540	-10.5540
Kayıt 2	45.3242	45.3242	45.3242	45.3242	45.3242	45.3242
Kayıt 3	37.8887	37.8887	37.8887	37.8887	37.8887	41.9129
Kayıt 4	-0.6173	-0.6173	-0.6173	-0.6173	-0.6173	-1.2347
Kayıt 5	6.1852	6.1852	6.1852	6.1852	6.8065	6.8065
Kayıt 6	46.2092	46.2092	46.2092	46.2092	46.2092	46.2092

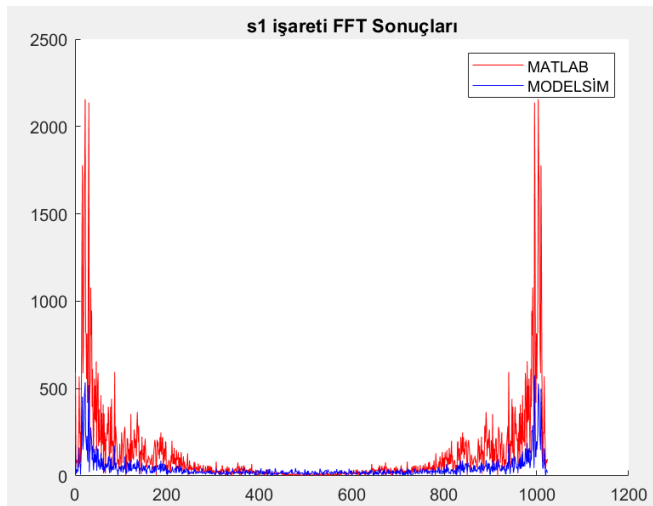
2 mikrofondan alınan ses kayıtlarında mikrofona kanalları s1 ve s2 olarak isimlendirilmiştir. Ses verilerinde el çırpma sesi kaydedilmiştir. 50000 örnek içeren bir kayıt verisinde 1024 pencere boyutu kullanılmıştır ve bu pencere bölgesi el çırpma sesinin olduğu zaman dilimi

bölgesinden alınmıştır. Mikrofonlardan alınan ses kayıtlarından bir tanesi örnek olarak Şekil 8.2 'de görülmektedir.

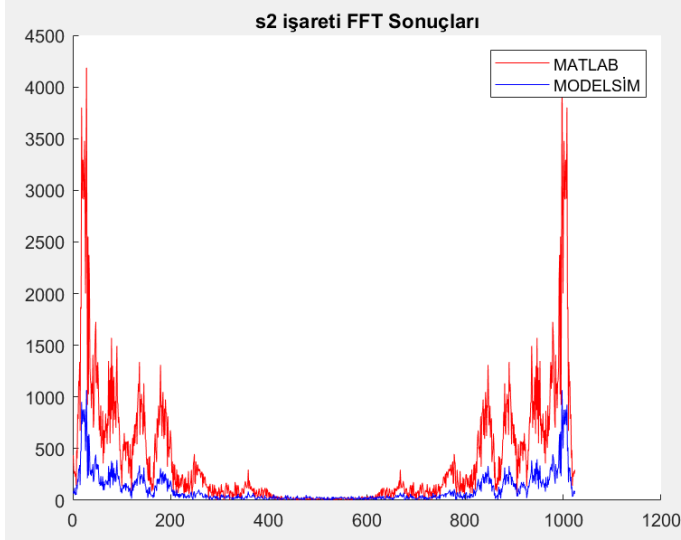


Şekil 8.2. Ses kayıt örneği

Ses işaretlerine uygulanan HFD bloğunun çıkış işareti, Matlab 'da aynı giriş işaretleri uygulanarak fourier dönüşümü alınan sonuç ile karşılaştırılmıştır. İki kanaldan kaydedilen örnek bir ses işaretinin Matlab ve Modelsim fourier dönüşümünün sonuçları Şekil 8.3 ve Şekil 8.4'de görülmektedir. Matlab ve Modelsim fourier dönüşüm sonuçları incelendiğinde her iki sonucunda aynı olduğu sadece Matlab 'dan elde edilen sonucun genlik değerlerinin Modelsim 'den elde edilen sonucun genlik değerlerinden daha büyük olduğu görülmektedir. Bu farkın Matlab'da kullanılan fonksiyonlar ile Altera Megacore 'da bulunan fonksiyonların çalışmasındaki farklılıklardan kaynaklanabileceği düşünülmektedir.

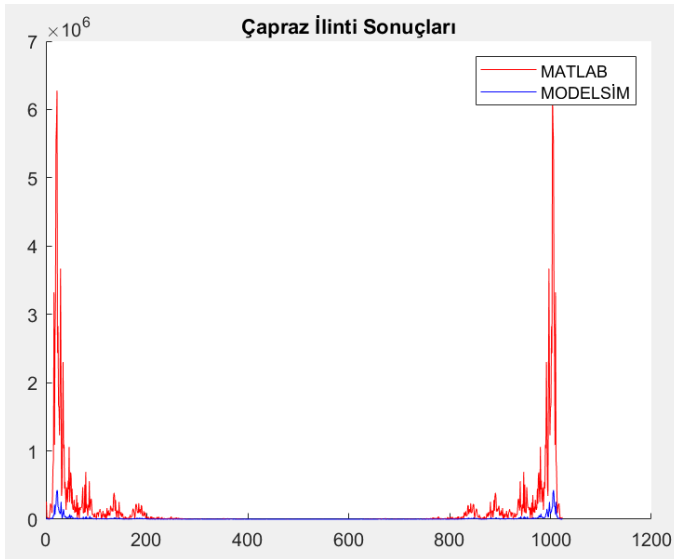


Şekil 8.3. s1 işaretinin Fourier dönüşümünün MATLAB ve MODELSİM karşılaştırması



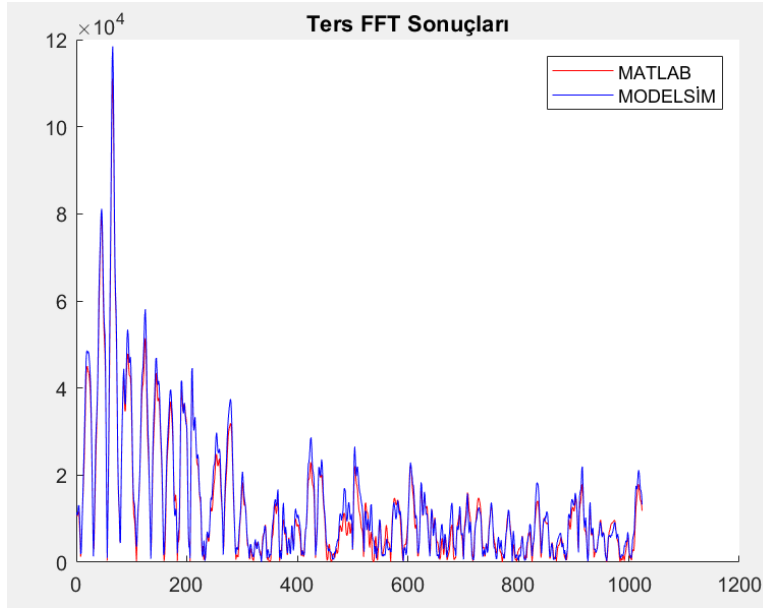
Şekil 8.4. s2 işaretinin Fourier dönüşümünün MATLAB ve MODELSİM karşılaştırması

HFD bloğundan sonra çapraz ilinti öbeği ile işleme alınan işaretlerin ilinti sonucu ile Matlab’da aynı işaretlere uygulanan çapraz ilinti yöntemi karşılaştırılmıştır. Çapraz ilinti işleminin MATLAB ve MODELSİM karşılaştırılması sonucu Şekil 8.5 ‘ de gösterilmektedir. Sadeleştirilmiş çapraz ilinti yönteminin Matlab ve Modelsim sonuçları incelendiğinde her iki sonucunda aynı olduğu sadece Matlab ’dan elde edilen sonucun genlik değerlerinin Modelsim ’den elde edilen sonucun genlik değerlerinden daha büyük olduğu görülmektedir.



Şekil 8.5. Sadeleştirilmiş çapraz ilinti yönteminin MATLAB ve MODELSİM karşılaştırması

Çapraz İlintisi alınan işaretlere ters Fourier işlemi uygulanarak iki işaret arasındaki gecikme miktarı maksimum noktanın olduğu yer olacak şekilde sonuç elde edilir. Çapraz ilinti işlemi sonrasında uygulanan ters Fourier işleminin MODELSİM sonucu, MATLAB 'da uygulanan sonuç ile karşılaştırılmıştır. Elde edilen sonuç Şekil 8.6 'da gösterilmektedir.



Şekil 8.6. Ters Fourier işleminin MATLAB ve MODELSİM karşılaştırması

Matlab ve Modelsim sonuçları incelendiğinde her iki sonucunda aynı olduğu görülmektedir. Her iki uygulamanın sonucunda da dürtü işareti dizinin aynı elamanını işaret etmektedir. Bu sonuç bize sadeleştirilmiş çapraz ilinti yönteminin donanım tasarımının başarılı bir şekilde çalıştığını göstermektedir. Geliştirilen donanım mimarisinin doğruluğu Matlab ile geliştirilen yazılım ile karşılaştırmıştır ve geliştirilen donanım mimarisinin doğruluğu test edilmiştir.

Birden fazla ses kaydı ile çapraz ilinti yöntemi kullanılarak iki ses işareti arasındaki gecikmenin bulunmasına yönelik MATLAB 'dan alınan sonuçlar ve Modelsim simülasyon ortamından elde edilen sonuçlar Çizelge 8.3'de karşılaştırılmıştır. Hesaplanan gecikme değerleri arasındaki hata farkı en çok 6 örnektir ve sadece bir kayıta böyle bir fark görülmektedir. Diğer kayıtlarda en çok 1 örnek fark vardır ya da birebir aynı gecikme değeri hesaplanmıştır. Tüm işaretler göz önüne alındığında ortalama hata farkı 1.5 örnektir.

Donanım mimarisinde gecikme hesaplamasından kaynaklanan yaklaşık olarak ortalama hata farkı açı olarak 0.9 derece olmaktadır.

Çizelge 8.3. Ses işaretleri arasındaki hesaplanan gecikmenin MATLAB ve MODELSİM’den elde edilen sonuçlar

SES KAYITLARI	MATLAB SONUCU	MODELSİM SONUCU
Kayıt 1	19	19
Kayıt 2	66	65
Kayıt 3	57	57
Kayıt 4	-1	-2
Kayıt 5	67	66
Kayıt 6	-17	-23

Açı hesaplamalarında kullanılan LUT tablosu 1 derecelik açı farkı hesaplanarak oluşturulmuştur ve ara değer açı hesaplamaları yapılmamıştır. Bu sebeple açı sonuçları tam sayılardan oluşmaktadır. İki mikrofon kullanılarak ses kaynağının konumunun açısal olarak tespit edilmesine dair MATLAB ortamında elde edilen sonuçlar Modelsim simülasyon ortamından elde edilen sonuçlar ile Çizelge 8.4 ’de karşılaştırılmıştır. Uzaklık ve açı hesaplamalarının donanım mimarisinin doğruluğunun test edilebilmesi için bu karşılaştırma tablosu incelenmiştir. Elde edilen açı değerleri incelendiğinde sadece kayıt 6’da açısal hata farkı 4 derecedir. Bu farkın sebebi Çizelge 8.3 ’de kayıt 6’da gecikme hesabında ortaya çıkan farkıdır. LUT tablosuna göre karşılaştırma yaparak oluşturulan açı hesaplaması neticesinde hesaplanan açı değerlerindeki sapma normal kabul edilmektedir.

Çizelge 8.4. Ses kaynaklarının konumunun açısal olarak MATLAB ve MODELSİM den elde edilen sonuçları

SES KAYITLARI	MATLAB SONUCU	MODELSİM SONUCU
Kayıt 1	11.8125	11
Kayıt 2	45.3242	44
Kayıt 3	37.8887	37
Kayıt 4	-0.6173	-1
Kayıt 5	46.2092	45
Kayıt 6	-10.5540	-14

Verilog donanım tanımlama dili kullanılarak geliştirilen sistem ile Matlab ortamında geliştirilen sistemin sonuçları incelenmiştir ve incelenen sonuçlara göre tasarlanan sistemin doğru olarak çalıştığı test edilmiştir.

Tasarlanan donanımın kaynak tüketimi Resim 8.1’de gösterilmektedir. Pencere boyutu 1024 örnek ve hızlı fourier dönüşümü için kullanılan girişler 20 bit kullanılan donanım tasarımında kullanılan toplam lojik eleman 30267’dir. Giriş ve çıkışlar için kullanılan toplam pin 383’dür. Toplam hafıza bitleri 584710’dur. Donanım mimarisi Altera Cyclone IV E ailesi için sentezlenmiştir.

Flow Summary	
Flow Status	In progress - Sun Apr 21 13:09:31 2019
Quartus II 64-Bit Version	13.1.0 Build 162 10/23/2013 SJ Web Edition
Revision Name	fft
Top-level Entity Name	fft
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	30,267
Total combinational functions	21,558
Dedicated logic registers	26,858
Total registers	26858
Total pins	383
Total virtual pins	0
Total memory bits	584,710
Embedded Multiplier 9-bit elements	116
Total PLLs	0

Resim 8.1. Donanım kaynak tüketim tablosu

SONUÇ VE ÖNERİLER

Yapılan araştırmalara göre çapraz ilinti yöntemi ve sesin yayılımından yola çıkarak geliştirilmiş olan geometrik yöntemlerin çalışması görülmüştür. Bu tekniklere göre sesin mikrofonlara varış zamanları arasındaki farkın yani Δt değerlerinin doğru hesaplanması ile ses kaynağının yönü küçük açı hatalarıyla bulunabilmektedir.

Ses kayıtlarında doyuma giden verilerin olması, kayıt alınan ortam şartları, mikrofonların eşdeğer karakteristikte olamaması, mikrofonlar arası mesafe sonuçların düzgün değerlendirilmesini etkilemektedir. Kazanç ayarlı mikrofonlar kullanılarak iyi kayıtlar üzerinde yüksek örnekleme frekansı alınarak daha iyi sonuçlar elde edilebilir.

Ses kaynağının yerinin belirlenmesi yöntemlerinin duyarlılığı birden çok etkene bağlıdır. Bunlardan en önemlisi dizideki mikrofon sayısıdır. Mikrofon sayısı arttıkça geliş açısının daha doğru bir şekilde kestirilmesi mümkün olmaktadır. İkinci etken ise, örnekleme frekansıdır. Örnekleme frekansı arttırıldıkça dizinin uzaysal çözünürlüğü de artacaktır. Ancak burada dikkat edilmesi gereken durum, örnekleme frekansı arttıkça, aynı zaman dilimini kullanan algoritmaların hesaplama maliyetinin artacağıdır. Bu nedenle, hesaplama için örnek sayısı da dikkate alınmalıdır. Ayrıca, analog-sayısal dönüştürücülerin sınırları da çözünürlüğü etkileyen önemli bir kısıttır.

Bu çalışmada çapraz ilinti temelli ses konum tespiti algoritması Verilog donanım tanımlama dili kullanılarak gerçekleştirilmiştir. Simülasyon sonuçları ModelSim-Altera 10.1d kullanılarak elde edilmiştir. Çapraz ilinti yöntemi algoritmanın performansını etkilemeyecek şekilde sadeleştirilmiş ve bu sayede çapraz ilintinin bulunması için gereken bölme işlemi ortadan kaldırılmıştır. Ses kayıtlarında doyuma giden verilerin olması, kayıt alınan ortam şartları, mikrofonların eşdeğer karakteristikte olamaması, mikrofonlar arası mesafe sonuçların düzgün değerlendirilmesini etkilemektedir. Kazanç ayarlı mikrofonlar kullanılarak iyi kayıtlar üzerinde yüksek örnekleme frekansı alınarak daha iyi sonuçlar elde edilebilir. Ses işaretlerinde ön işleme teknikleri kullanılarak gürültü, yankı gibi bozucu etkilerin giderilmesi ile sonuçlar daha da iyileştirilebilir.



KAYNAKLAR

1. Dağıstanlı, B. ve Ocak, H. (2008, Kasım). *Mobil robotta ses kaynağının yönünün kestirilmesi*. Otomatik Kontrol Ulusal Toplantısı, İstanbul.
2. Weinstein, E., Steele, K., Agarwal, A., and Glass, J. (2004, April). *LOUD: a 1020-node modular microphone array and beamformer for intelligent computing space*. Massachusetts Institute of Technology Cambridge United States Technical Memo, Cambridge.
3. Rodriguez, A., Julian, P., and Pirchio, F. (2007). A verilog hdl digital architecture for delay calculation. *Latin American Applied Research*, 37(1), 41-45.
4. Kefeli, H. C. and Ertürk, S. (2008, April). *Spatial position detection of acoustic sources on the ti davinci platform using microphone arrays*. 16th Signal Processing, Communication and Applications Conference, Aydın.
5. Zhu, H., Guo, K., and Lin, Y. (2015, July). *DOA Estimation based on the microphone array for the time-varying number of sound signals*. 18th International Conference on Information Fusion, Washington.
6. Davidson, K. D. M. and Priyanka, M. (2016, March). *FPGA implementation of acoustic source localization algorithm*. International Conference on Wireless Communications, Signal Processing and Networking conference, Chennai, India.
7. Chunhe, Y. and Yuanyuan, Y. (2014, October). *A design of algorithm based on microphone array*. 12th International Conference on Signal Processing, Shenyang, China.
8. Valin, J. M., Michau, F., Rouat, J., and Létourneau, D. (2003, October). *Robust sound source localization using a microphone array on a mobile robot*. Conference on Intelligent Robots and Systems, Las Vegas.
9. İnternet: URL: <https://sourcetechn411.com/top-fpga-companies-for-2013/> , Son Erişim Tarihi: 18.05.2019.
10. Harding, B. and Cofer, R. C. (2006). *Rapid system prototyping with FPGA's* (First Edition). Massachusetts USA: Elsevier, 87, 92.
11. Maxfield, C. (2009). *FPGAs world class designs* (First Edition). Massachusetts USA: Newnes, 50, 67.
12. İnternet: URL: <https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/archives/ug-fft-10.1.pdf> , Son Erişim Tarihi: 06.08.2019.
13. Veen, V. and Buckley, K. (1988). Beamforming: a versatile approach to spatial filtering. *IEEE Acoustics, Speech and Signal Processing (ASSP) Magazine*, 4-24.
14. Çontar, E. (2007). *Mikrofon Dizilerinde Ses Kaynağının Yerinin Genetik Algoritma Kullanılarak Bulunması*, Yüksek Lisans Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara, 30-33.



ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : ÖZTÜRK LAFCI, Merve
 Uyuğu : T.C.
 Doğum tarihi ve yeri : 1990, Çankırı
 Medeni hali : Evli



Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Yüksek Lisans	Gazi Üniversitesi / Elektrik Elektronik Müh.	Devam ediyor
Lisans	Kocaeli Üniversitesi / Elektronik Haberleşme Müh.	2012
Lise	Çankırı Anadolu Lisesi	2008

İş Deneyimi

Yıl	Yer	Görev
2017-Halen	Bilgi Teknolojileri ve İletişim Kurumu	Bilişim Uz. Yrd.
2013-2014	Digitest Elektronik	Ar-Ge Mühendisi

Yabancı Dil

İngilizce

Yayınlar

Öztürk Lafci, M., Lafci, M., ve Bilge, H.Ş. (2019, 26-27 Nisan). *FPGA Tabanlı Ses Kaynaklarının Konum Tespiti*, 6. Uluslararası Multidisipliner Çalışmaları Kongresi, Gaziantep.



GAZİ GELECEKTİR..