

FPGA KULLANILARAK DOĞRUDAN SAYISAL SENTEZ TABANLI FONKSİYON ÜRETECİ TASARIMI

DESIGN OF DIRECT DIGITAL SYNTHESIS FUNCTION GENERATOR USING FPGA

NAZİRE MERVE AYDOĞAN

Doç. Dr. Ali Ziya ALKAR

Tez Danışmanı

Hacettepe Üniversitesi

Lisansüstü Eğitim – Öğretim ve Sınav Yönetmeliğinin

Elektrik – Elektronik Mühendisliği Anabilim Dalı İçin Öngördüğü

YÜKSEK LİSANS TEZİ

olarak hazırlanmıştır.

2013

Nazire Merve AYDOĞAN'ın hazırladığı "**FPGA Kullanılarak Doğrudan Sayısal Sentez Tabanlı Fonksiyon Üreteci Tasarımı**" adlı bu çalışma aşağıdaki jüri tarafından **ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI**'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Başkan

Prof. Dr. Selçuk GEÇİM

Danışman

Doç. Dr. Ali Ziya ALKAR

Üye

Yrd. Doç. Dr. Harun ARTUNER

Üye

Yrd. Doç. Dr. Mehmet Demirer

Üye

Yrd. Doç. Dr. Umut SEZEN

Bu tez Hacettepe Üniversitesi Fen Bilimleri Enstitüsü tarafından **YÜKSEK LİSANS TEZİ** olarak onaylanmıştır.

Prof. Dr. Fatma SEVİN DÜZ

Fen Bilimleri Enstitüsü Müdürü

Her zaman yanımda olan canım aileme...

ETİK

Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada;

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

09/11/2013

Nazire Merve AYDOĞAN

ÖZET

FPGA KULLANILARAK DOĞRUDAN SAYISAL SENTEZ TABANLI FONKSİYON ÜRETECİ TASARIMI

Nazire Merve AYDOĞAN

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Danışmanı: Doç. Dr. Ali Ziya ALKAR

Kasım 2013, 94 sayfa

Periyodik dalgalar telekomünikasyon, elektromanyetik, biyomedikal gibi birçok elektronik alanda kullanılmaktadır. Doğrudan Sayısal Sentez (DDS), **istenilen frekansta ve yüksek frekans çözünürlüğünde dalga formları üretmek için kullanılan bir yöntemdir.** DDS sistemleri analog ve sayısal olmak üzere iki kısımdan oluşur. Örneklenmiş dalga verilerini içeren başvuru çizelgesi ve frekans belirleme algoritması sayısal kısımda bulunur. Bu kısımda dalga formları örneklenmiş veriler şeklinde üretilir. Sayısal olarak bulunan dalga formlarının analoga çevrilmesi için D/A çevirici kullanılır. D/A çevirici bloğu ise analog kısmı ifade eder.

Bu tez kapsamında, yüksek doğruluklu lokal osilatörlere veya farklı dalga şekillerine ihtiyaç duyulan sistemlerde kullanılmak üzere, FPGA üzerinde DDS tabanlı fonksiyon üretici tasarlanmıştır. Sinüs, kare, üçgen, testere ve sabit gerilim gibi bilinen dalga formları, frekansları ve genlikleri ayarlanabilir şekilde üretilmiştir. Frekans ve genlik değerleri, tez kapsamında geliştirilen seri kanal haberleşme arayüz programı aracılığıyla gerçek zamanlı olarak belirlenmektedir. FPGA programlanmasında VHDL programlama dili kullanılmıştır.

Fonksiyon üreticinin FPGA kullanılarak gerçekleştirilmiş olması, sistem hızının, hafıza boyutunun ve sistem büyüklüğünün kısıtlı ve önemli olduğu durumlarda kullanılabilirliğini ön plana çıkarmıştır. Üretilen dalgalar tasarlanan D/A çevirici bloğu ile analog formda kullanılabildikleri gibi FPGA çıkış hatlarından direkt alınarak sayısal olarak da kullanılabilmektedir. Sayısalan analoga çevrim işlemi en az kaynakla en yüksek doğruluk değerlerini sağlayacak şekilde tasarlanmıştır.

Anahtar Kelimeler: fonksiyon üretici, sinyal üretici, DDS, doğrudan sayısal sentez, FPGA, VHDL, D/A, sayısal, analog

ABSTRACT

DESIGN OF DIRECT DIGITAL SYNTHESIS FUNCTION GENERATOR USING FPGA

Nazire Merve AYDOĞAN

**Master of Science, Department of Electrical and Electronics
Engineering**

Supervisor: Assoc. Prof. Dr. Ali Ziya ALKAR

November 2013, 94 pages

Periodic waveforms are necessary for many applications like telecommunication, electromagnetic and biomedical. Direct Digital Synthesis (DDS) is a method of generating signals with high frequency resolution and fast frequency switching speeds by digital control. A general DDS system comprises analog and digital part. Frequency determining algorithm and look up table that holds the sampled waveform data make digital part. In this part waveforms are generated as sample points. D/A converter is used to convert digital data to analog data. D/A converter block makes up digital part of DDS.

In this thesis, we implemented a DDS based function generator by using FPGA, to be used in systems such as local oscillators or different kinds of waveforms are needed. Well known waveforms like sinus, rectangular, triangular, sawtooth and DC wave are generated with adjustable frequency and amplitude parameters. Frequency and amplitude values are determined by serial channel user interface which is implemented in this thesis. VHDL programming language is used to program FPGA.

Due to the usage of FPGA, this design will be suitable for applications where system speed, memory size and dimensions of system are important and limited. Generated waveforms can be used as digital data by directly using them from the FPGA output, also they can be used as analog signals from the output of D/A converter block. Digital to analog conversion is designed to supply highest accuracy with minimum resources.

Keywords: function generator, signal generator, DDS, direct digital synthesis, FPGA, VHDL, D/A, digital, analog, digital to analog converter

TEŞEKKÜR

Bu tezin oluşturulmasında sağladığı katkılardan ötürü tez danışmanım Doç. Dr. Ali Ziya ALKAR'a teşekkürlerimi sunuyorum.

Her zaman yanımda olan aileme, tez süresince bana moral veren tüm arkadaşlarıma, işyerindeki yöneticilerime, Aselsan A.Ş.'ye ve Sanayi Bakanlığı San-Tez Programı'na bu tez çalışmasına verdikleri destekten dolayı sonsuz teşekkürler.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	i
ABSTRACT	iii
TEŞEKKÜR	v
İÇİNDEKİLER	vi
ŞEKİLLER	viii
ÇİZELGELER	x
SİMGELER VE KISALTMALAR	xi
1. GİRİŞ	1
2. SAYISALDAN ANALOGA ÇEVİRİCİ BLOĞU TASARIMI	5
2.1 D/A Çevirici Entegre Devresi Seçimi	5
2.2 D/A Çevirici Balkon Kartı Tasarımı	7
2.3 D/A Çevirici Balkon Kartın Programlanması	12
2.3.1 D/A Çevirici Entegre Devresinin Veri Sayfalarının ve Zamanlama Şemalarının İncelenmesi	12
2.3.2 D/A Çevirici Entegre Devresi İçin Akış Diyagramı Oluşturulması ve Entegre Devresinin Programlanması	18
2.3.3 D/A Çevirici Entegre Devresi İçin Hazırlanmış Akış Diyagramının Optimize Edilmesi	25
3. BAŞVURU ÇİZELGESİNİN HAZIRLANMASI	28
3.1 BRAM Zamanlama Şemaları ve Parametreleri	28
3.2 BRAM Zamanlama Şemaları ve Parametreleri	32
3.3 BRAM Veri Kaydı ve İlkeme Parametreleri	34
3.4 Sinüs Verilerinin BRAM'a Kaydedilmesi	36
3.5 Kullanılan BRAM Boyutunun Optimize Edilmesi	38
4. DDS FONKSİYON ÜRETECİ TASARIMI	41
4.1 DDS Algoritmasının Gerçeklenmesi	41
4.2 Spektral Safılık Kavramları	45
4.3 Sinüs ve Kosinüs Dalga Formlarının Üretilmesi	48
4.4 Üçgen Dalga Formunun Üretilmesi	52
4.5 Testere Dalga Formunun Üretilmesi	54

4.6 Kare Dalga Formunun Üretilmesi	55
5. SERİ KANAL ARAYÜZÜ İLE GENLİK VE FREKANS AYARLANMASI	58
5.1 Seri Kanal Özellikleri	58
5.2 DDS Algoritmasına Değiştirilebilen Genlik ve Frekans Parametrelerinin Eklenmesi	61
5.3 Seri Kanal Arayüzünün Hazırlanması ve Baud Hızının Belirlenmesi	68
6. FONKSİYON ÜRETECİ DOĞRULAMASI VE TASARIM BELİRTİMLERİ	72
6.1 Fonksiyon Üretici Tasarım Belirtileri	72
6.2 Algoritmaların Donanım Üzerinde Doğrulanması	73
7. SONUÇLAR	81
KAYNAKLAR	83
EKLER	86
ÖZGEÇMİŞ	93

ŞEKİLLER

	Sayfa
Şekil 1. DDS Bölümleri	2
Şekil 2. ML 507 Geliştirme Kartı [18]	8
Şekil 3. ML507 J6 Konektör Bağlantıları [19]	9
Şekil 4. AD 5764 Örnek Tasarım [20]	10
Şekil 5. D/A Çevirici Balkon Kartı Ön Yüzü	11
Şekil 6. D/A Çevirici Balkon Kartı Arka Yüzü	11
Şekil 7. KN1 Konektörü Bağlantıları	13
Şekil 8. AD5764 Seri Arayüz Zamanlama Şeması [20]	17
Şekil 9. D/A Çevirici Kodu Blok Şeması	19
Şekil 10. İlk Tasarım D/A Çevirici Kodu Akış Diyagramı	20
Şekil 11. D/A Çevirici Kodu ChipScope Simülasyonu Sonuçları	24
Şekil 12. Optimize Edilmiş D/A Çevirici Kodu Akış Diyagramı	25
Şekil 13. Optimize Edilmiş D/A Çevirici Kodu ChipScope Simülasyonu Sonuçları	27
Şekil 14. İki Kanallı 36 Kbit BRAM Veri Akış Şeması [21]	29
Şekil 15. WRITE_FIRST Modu BRAM Zamanlama Şeması	30
Şekil 16. READ_FIRST Modu BRAM Zamanlama Şeması	31
Şekil 17. NO_CHANGE Modu BRAM Zamanlama Şeması	31
Şekil 18. BRAM Adresleme Haritaları	33
Şekil 19. AD5764 Değerlerine Göre Kodlanmış Sinüs Grafiği	36
Şekil 20. Örneklenmiş Sinüs Dalga Formu [11]	38
Şekil 21. Örneklenmiş Sinüs Dalga Formunun İlk Çeyrek Düzlemi	39
Şekil 22. Sinüs Verisinin Simetri Özelliğiyle Elde Edilmesi [22]	40
Şekil 23. Temel DDS Veri Akış Diyagramı	41
Şekil 24. DDS Akış Diyagramında Gürültülerin Konumları	45
Şekil 25. Sinüs ve Kosinüs Dalgalarının Simetri Özelliği ile Oluşturulması	50
Şekil 26. Sinüs/Kosinüs Dalga Üretim Simülasyonu (Referans Saati = 580 Birim Periyot)	51
Şekil 27. Sinüs/Kosinüs Dalga Üretim Simülasyonu (Referans Saati = 10 Birim Periyot)	52
Şekil 28. Üçgen Dalga Üretim Simülasyonu (Referans Saati = 10 Birim Periyot)	53

Şekil 29. Testere Dalga Üretim Simülasyonu (Referans Saati = 10 Birim Periyot).	55
Şekil 30. Kare Dalga Üretim Simülasyonu (Referans Saati = 10 Birim Periyot) ...	56
Şekil 31. Tüm Dalga Formalarının Ortak Simülasyonu.....	57
Şekil 32. Seri Kanal Modem Kablosu Bağlantıları	58
Şekil 33. RS-232 Veri İletişimi	60
Şekil 34. ADM3202 Entegre Devresi Seri Kanal Kodlama Örneği.....	61
Şekil 35. Seri kanal Arayüzü Görünümü ve Dalga Formu Seçim Ekranı	71
Şekil 36. Doğrulama Kurulum Şeması.....	74
Şekil 37. Doğrulama Kurulumu	75
Şekil 38. 5 Volt Sabit Gerilim Üretim	76
Şekil 39. 10 Volt Sabit Gerilim Üretimi.....	76
Şekil 40. 5 Volt 500 Hz Sinüs/Kosinüs Dalgası Üretimi	77
Şekil 41. 10 Volt 1000 Hz Sinüs/Kosinüs Dalgası Üretimi	77
Şekil 42. 5 Volt 500 Hz Üçgen Dalga Üretimi	78
Şekil 43. 10 Volt 1000 Hz Üçgen Dalga Üretimi	78
Şekil 44. 5 Volt 500 Hz Testere Dalga Üretimi	79
Şekil 45. 10 Volt 1000 Hz Testere Dalga Üretimi	79
Şekil 46. 5 Volt 500 Hz Kare Dalga Üretimi	80
Şekil 47. 10 Volt 1000 Hz Kare Dalga Üretimi	80
Şekil 48. AD5764 Entegre Devresi, Çevre Elemanları ve Çıkış Kanalları Tasarımı	86
Şekil 49. Gerilim Çevirici Blokları Devre Tasarımları	87
Şekil 50. DC-DC Çevirici Çıkış Filtreleme Devreleri	88
Şekil 51. DC-DC Çevirici ve Çıkış Devre Elemanları	89
Şekil 52. Test Noktaları, Ledler, Ters Akım Koruma Diyotları ve Güç Giriş Konektörü.....	89

ÇİZELGELER

	<u>Sayfa</u>
Çizelge 1. D/A Çevirici Entegre Devre Özellikleri	6
Çizelge 2. İkili Tabanda Ofsetli Kodlama	14
Çizelge 3. Giriş Yazmacı Bit Haritası [20].....	15
Çizelge 4. AD5764 Giriş Yazmacı Bit Tanımları [20]	15
Çizelge 5. AD5764 Seri Arayüz Zamanlama Parametreleri [20].....	17
Çizelge 6. AD5764 Giriş Yazmacına Yazılacak Değer	21
Çizelge 7. BRAM Bağlantı Adları ve Tanımları	29
Çizelge 8. RAMB18 Veriyolu Genişlikleri.....	32
Çizelge 9. INIT_xx Niteliği Bit Pozisyonları	35
Çizelge 10. RAMB18'e Kaydedilen Örneklenmiş Sinüs Verisi	37
Çizelge 11. DDS Algoritması Tasarım Belirtilimleri	72
Çizelge 12. Seri Kanal Arayüzü Tasarım Belirtilimleri	73
Çizelge 13. Donanım Elemanları	74

SİMGELER VE KISALTMALAR

Simgeler

μ	Mikro
\approx	Yaklaşık Olarak Eşit
Δ	Delta

Kısaltmalar

AM	Genlik Kiplenimi	Amplitude Modulation
ASIC	Uygulamaya Özel Entegre Devre	An Application Specific Integrated Circuit
ASK	Sayısal Genlik Kiplenimi	Amplitude Shift Keying
BRAM	Rastgele Erişimli Blok Bellek	Block Random Access Memory
CORDIC	Koordinat Döndüren Sayısal Bilgisayar	Coordinate Rotation Digital Computer
D/A	Sayısalan Analoga	Digital to Analog
DAC	Sayısalan Analoga Çevirici	Digital to Analog Converter
dB	Desibel	Decibel
DC	Doğru Akım	Drain Current
DDS	Doğrudan Sayısal Sentez	Direct Digital Synthesis
DNL	Türevsel Doğrusalsızlık	Differential Nonlinearity
DVI	Sayısal Görsel Arayüz	Digital Visual Interface

EPROM	Erasable Programmable Read Only Memory	Silinip Programlanabilir Salt Okunur Bellek
FKB	Frekans Kontrol Bilgisi	Frequency Control Word
FM	Frekans Kiplenimi	Frequency Modulation
FPGA	Field Programmable Gate Array	Alanda Programlanabilir Kapı Dizisi
FSK	Sayısal Frekans Kiplenimi	Frequency Shift Keying
HDL	Donanım Tanımlama Dili	Hardware Description Language
Hex	On Altılı Sayı Sistemi	Hexadecimal
I/O	Giriş Çıkış	Input Output
IC	Entegre Devre	Integrated Circuit
INL	İntegral Doğrusalsızlık	Integral Nonlinearity
JTAG	Test Eylem Grubu Eki	Joint Test Action Group
Kbit	Kilobit	Kilobit
kbps	Saniyede x Kilobit	Kilobit per Second
KHz	KiloHertz	KiloHertz
LSB	En Önemsiz Bit	Least Significant Bit
LUT	Başvuru Çizelgesi	Look-up Table
MHz	MegaHertz	MegaHertz
MSB	En Önemli Bit	Most Significant Bit
Msp/s	Saniyede x Mega Örnek	Mega Sample per Second
NCO	Sayısal Denetimli Osilatör	Numerically Controlled

		Oscillator
ns	Nanosaniye	Nanosecond
PLL	Evre Kenetleme Döngüsü	Phase Locked Loop
PSK	Evre Kaydırmalı Kiplenim	Phase Shift Keying
RTL	Yazmaç Aktarım Dili	Register Transfer Language
SNR	Sinyal Gürültü Oranı	Signal to Noise Ratio
SO-DIMM	Küçük Çerçevesi Çift Sıralı Hafıza Modülü	Small Outline Dual in-Line Memory Module
SPI	Seri Çevre Donatımı Arayüzü	Serial Peripheral Interface
UART	Universal Asynchronous Receiver/Transmitter	Genelgeçer Eşzamansız Alıcı/Verici
UCF	Kullanıcı Kısıtlamaları Dosyası	User Constraints File
USART	Universal Synchronous-Asynchronous Receiver/Transmitter	Genelgeçer Eşzamanlı-Eşzamansız Alıcı/Verici
USB	Evrensel Seri Veriyolu	Universal Serial Bus
VGA	Video Grafikleri Dizini	Video Graphics Array
VHDL	Çok Yüksek Hızlı Entegre Devrelerin Donanım Tanımlama Dili	VHSIC Hardware Description Language
V_{t-t}	Tepeden Tepeye Genlik	Voltage Peak to Peak

1. GİRİŞ

Elektronik sistemler gelişip karmaşıktıkça güvenilirlik ve düşük güç tüketimi konularındaki beklentiler artmaktadır [1]. Artan beklentiler sonucu frekans sentez teknolojisinde üçüncü nesil olan doğrudan sayısal sentez (DDS), doğrudan frekans sentezi ve evre kenetleme frekans sentezi yöntemlerinden sonra ortaya çıkmıştır [2]. İlk olarak 1971 yılında Amerikan bilim adamı J. Tierncy doğrudan sentez konseptinden yola çıkarak, tamamen sayısal teknoloji tabanlı DDS yöntemini bulmuştur [3]. DDS ilerleyen yıllarla birlikte geleneksel frekans sentezleme teknolojilerine oranla daha yüksek çözünürlük değeri, kısa frekans anahtarlama süresi, düşük faz gürültüsü gibi performans parametreleri sergilendiğinden dolayı frekans sentezleme yöntemleri arasında öne çıkmıştır [4].

Farklı pratik uygulamalarda [5][6][7] kullanılan elektronik sistemlerden faz, frekans ve genlik gibi özellikleri belirlenebilen sinüsel dalga formlarının üretilmesi beklenmektedir [8]. DDS genel olarak zamanla değişen dalga formlarının sayısal olarak üretilmesinin ardından analoga dönüştürülerek analog dalga formlarının üretilmesi tekniğidir. DDS asıl olarak sayısal olduğundan çıkış frekansları arasında hızlı anahtarlama yapabilmekle birlikte yüksek frekans çözünürlüğüne sahip dalgalar üretebilmekte ve geniş bir frekans skalasında çalışabilmektedir [9].

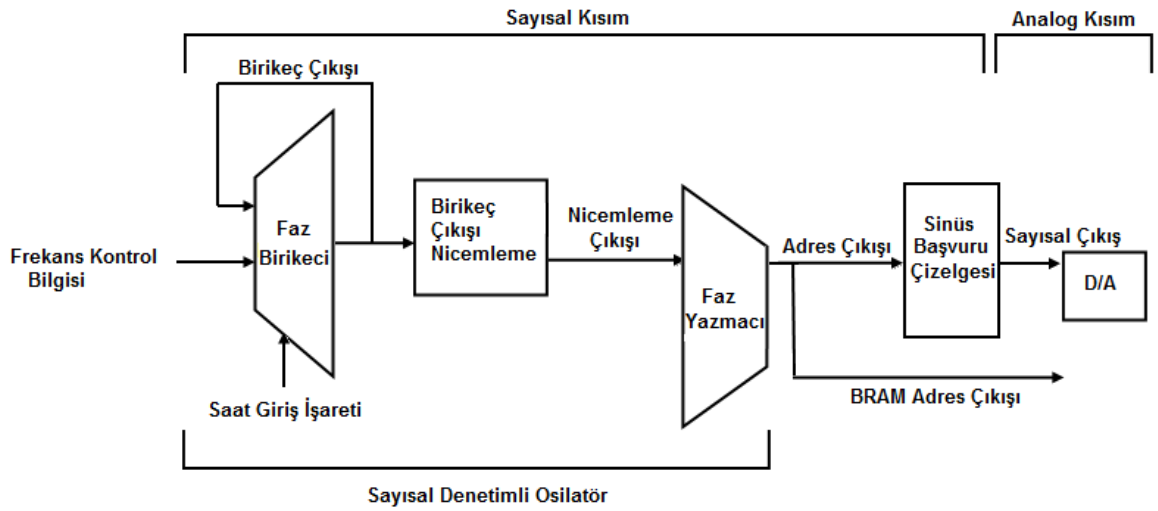
DDS yönteminin, evre kenetleme döngüsü (PLL) kullanılarak tasarlanan analog yöntemlere kıyasla birçok avantajı bulunmaktadır [10]:

- Sayısal kontrol sayesinde μHz seviyesinde frekans ve faz ayarı yapabilme kapasitesi.
- Çok yüksek değerlere çıkabilen frekans anahtarlama hızı.
- Sayısal olarak ayarlanabilen frekans ve genlik değerleri.
- Uygulama kolaylığı.

DDS genel olarak sayısal ve analog olmak üzere iki ana bölümden oluşur [11]. Bu bölümler Şekil 1'de gösterilmiştir. Sayısal kısım sayısal denetimli osilatör ve başvuru çizelgesinden oluşur. Sayısal denetimli osilatör ise faz birikeci, faz birikeci

nicemleme devresi ve faz yazmacından oluşur. Faz birikecinin girişinde frekans kontrol bilgisi verisi ve birikeç çıkışının geri beslemesi bulunur. Frekans kontrol bilgisi verisi dalgaının frekans parametresini ayarlama da kullanılan asıl veridir. Faz yazmacından çıkan ve adres verisi olarak kullanılan veri başvuru çizelgesini adreslemede kullanılır. Başvuru çizelgesinde örneklenmiş sinüs verileri bulunmaktadır ve faz yazmacından gelen adreste bulunan sinüs verileri başvuru çizelgesinin çıkışına gönderilir.

Analog kısım ise sayısal dan analoga (D/A) çevirici bloğundan oluşur. D/A çevirici bloğu başvuru çizelgesinden gelen örneklenmiş dalga verilerini DDS'nin çalışma saatiyle uyumlu bir şekilde analoga çevirir. D/A çevirici çıkışında analog formda dalgalar gözlemlenebilir.



Şekil 1. DDS Bölümleri

Xilinx firmasının DDS yöntemiyle sinüsel dalga formları üretebilen bir çekirdeği bulunmaktadır [12]. Bu çekirdekte kullanılan algoritmalar kullanıcı tarafından bilinmemekle birlikte dalga çıkış frekansı, faz birikeci genişliği ve saat işareti gibi parametreler kullanıcıya bir arayüz aracılığıyla seçtirilmektedir. Seçilen her parametre seti için bu çekirdeğin sentezlenmesi ve alanda programlanabilir kapı dizisine (FPGA) tekrar yüklenmesi gerekmektedir. Bu işlem hem zaman kaybına yol açmakta hem de çekirdeğin kullanılabilirliğini düşürmektedir. Bu tez çalışması ile belirli bir aralıkta seçilebilen frekans ve genlik parametreleri, FPGA'ya tekrar yükleme gerektirmeden ve gerçek zamanlı olarak değiştirilebilen dalga formlarının

üretimi gerçekleştirilmiştir. DDS yöntemi ile dalga üretimini konu alan birçok yayında [13][4], değiştirilebilir frekans parametresi üzerinde durulurken genlik değişimi üzerinde durulmamaktadır. Bu tez çalışmasının DDS yöntemiyle ilgili diğer yayınlardan farkı, üretilen dalgaların hem frekans hem genlik parametrelerinin gerçek zamanlı olarak değiştirilebilmesidir.

Bu çalışma bütünüyle bir fonksiyon üretici gibi kullanılabileceği gibi bir ya da birden çok bloğu değiştirilerek farklı uygulamalar elde edilebilecek bir çatı olarak da değerlendirilebilir. Örneğin elde edilen sayısal sinyaller D/A çevirici kullanılmadan direkt sayısal olarak kullanılabileceği gibi bu çalışmada kullanılan farklı bir D/A çevirici bloğu kullanılarak daha hızlı ya da daha hatasız analog dalgalar elde edilebilir. Bunun yanı sıra başvuru çizelgesine bu çalışmada kaydedilen sinüs verisi yerine genlik kiplenimi (AM), frekans kiplenimi (FM), sayısal genlik kiplenimi (ASK), sayısal frekans kiplenimi (FSK), evre kaydırmalı kiplenim (PSK) gibi yöntemlerle kiplenmiş dalga verileri kaydedilebilir [14]. Yüksek sayıda döngü gerektirdiğinden ve gecikme süresi uzun olduğundan dolayı bu çalışmada tercih edilmeyen, hiperbolik ve trigonometrik fonksiyonların hesaplanmasında kullanılan bir algoritma olan CORDIC algoritması [15], başvuru çizelgesi bloğu yerine kullanılabilir. Bu şekilde blokların değişimiyle çalışma farklı boyutlara taşınabilir.

Bu tez kapsamında anlatılan bölümlerin sıralaması yapılırken, içerdiği hesaplamalarla diğer bölümlerdeki çalışmaları etkileyen bölümlere öncelik verilmiştir. Sayısalan analoga çevrim frekansına uygun bir saat işareti seçilerek tüm DDS algoritmasında kullanılması gerektiğinden ikinci bölümde sayısalan analoga dönüştürücü bloğunun tasarımı anlatılacaktır. Bu blok tasarlanırken kullanılan entegre devre ile ilgili bilgi verilecek ve programlanma parametrelerine yer verilecektir.

DDS algoritmasında kullanılacak olan başvuru çizelgesinin boyutlarından yola çıkılarak başvuru çizelgesi adres verisinin büyüklüğü, dolayısıyla birikeç çıkışı nicemleme oranı belirleneceğinden, üçüncü bölümde DDS yönteminin gerçekleştirilmesinde kullanılmış olan başvuru çizelgelerinin oluşturulma yöntemi üzerinde durulacaktır.

Dördüncü bölümde ise önceki bölümlerde yapılan hesaplamalar ve parametreler kullanılarak oluşturulmuş olan DDS algoritmasının detaylarına yer verilecektir. Ayrıca bu bölümde çeşitli dalga formlarının oluşturulmasında kullanılan algoritmalar anlatılacaktır.

Beşinci bölümde parametrelerin gerçek zamanlı olarak değiştirilmesi için kullanılacak olan seri kanal arayüzünün hazırlanışı anlatılacaktır. Seri kanal arayüzü üzerinden genlik ve frekans bilgilerinin ayarlanma yöntemine değinilecektir.

Altıncı bölümde tasarım belirtileri verilecek, oluşturulan tüm tasarımın doğrulanması için hazırlanan kurulum ve doğrulama sonuçları anlatılacaktır.

Yedinci ve son bölümde ise tasarım sırasında karşılaşılanlar ile birlikte sonuçlara yer verilecektir. Aynı zamanda gelecek dönem çalışmaları için öneriler verilecektir.

2. SAYISALDAN ANALOGA ÇEVİRİCİ BLOĞU TASARIMI

D/A çevirici bloğu, DDS algoritması ile oluşturulan sayısal dalgaların analoga çevrilmesi için gereklidir. Bu blok sayesinde dalga formları osiloskop ile gözlemlenebilmektedir. DDS algoritmasında kullanılacak olan ve senkronizasyonu sağlayan saat işaretinin frekans seçimi, D/A çevirici bloğunun çalışma frekansına göre seçileceğinden D/A çevirici bloğunun anlatımına öncelik verilmiştir.

2.1 D/A Çevirici Entegre Devresi Seçimi

Sayısalardan analoga çevrim aşamasında kullanılmak üzere hazır analog modüller araştırılmıştır. Virtex FPGA'lar ile uyumlu çalışan MEMEC firmasına ait P160 analog modülü dönüştürme işlemi için uygun olduğu halde yüksek gürültü çıkışı, türevsel ve integral(tümlevsel) doğrusalsızlığı gibi nedenlerle sayısalardan analoga dönüştürücü kartının, seçilecek yüksek doğruluktaki bir D/A çevirici entegre devresi ile tasarlanmasına karar verilmiştir.

Ön çalışmalar sırasında değerlendirilmiş olan MEMEC P160 Analog Modülü 2 adet analogdan sayısala (A/D) çevirici, 2 adet D/A çevirici kanalı içermektedir, modülün D/A çevirici entegre devrelerinin özellikleri aşağıdaki gibidir:

- 12-bit çözünürlük
- 165 Msps çıkış güncelleme hızı
- Tek çıkışlı $2V_{t-t}$ analog çıkış
- ± 1.75 en önemsiz bit (LSB) türevsel doğrusalsızlık (DNL)
- ± 2.50 LSB integral doğrusalsızlık (INL)
- $10 \mu V/\sqrt{Hz}$ çıkış gürültüsü
- 30 ns yatışma süresi

DNL bir en önemsiz bitin ideal değeriyle gerçek kod adım yüksekliği arasındaki farktır. Bir D/A çeviricinin kod atlamamayı garanti etmesi için DNL değerinin 1 LSB'ye eşit ya da 1 LSB'den düşük olması gerekmektedir [16].

INL dönüştürücünün transfer fonksiyonunun, tam skala aralığında ideal doğrusal çizgiden ne kadar saptığını gösterir [17]. Belli bir basamaktaki INL değeri, başlangıç basamağından o basamağa kadar olan tüm DNL değerlerinin toplanmasıyla bulunur.

Sayısalan analoga dönüştürücü çalışmalarına, uygun çevirici entegre devresinin seçimiyle devam edilmiştir. Entegre devrenin öncelikle gerilim çıkışlı bir entegre devre olması gerektiğinden, Analog Devices firmasına ait gerilim çıkışlı sayısalan analoga dönüştürücüler arasında bir eleme yapılmış ve AD 5372, AD 5764 ve AD 5791 entegre devreleri belirlenmiştir. Bu entegre devrelerin özellikleri Çizelge 1 ile verilmiştir.

Çizelge 1. D/A Çevirici Entegre Devre Özellikleri

AD 5372	AD 5764	AD 5791
Gerilim Çıkışlı	Gerilim Çıkışlı	Gerilim Çıkışlı
16/14 Bit	16 Bit	20 Bit
Seri Giriş	Seri Giriş	Seri Giriş
32 Kanal	4 Kanal	1 Kanal
-4 V - +8 V Çıkış Gerilimi	± 10 V Çıkış Gerilimi	± 10 V Çıkış Gerilimi
20 μ s Yatışma Süresi	8 μ s Yatışma Süresi	1 μ s Yatışma Süresi
250 nV/ $\sqrt{\text{Hz}}$ Gürültü Yoğunluğu	80 nV/ $\sqrt{\text{Hz}}$ Gürültü Yoğunluğu	7.5 nV/ $\sqrt{\text{Hz}}$ Gürültü Yoğunluğu
± 1 LSB Maksimum DNL	± 1 LSB Maksimum DNL	± 2.5 LSB Maksimum DNL
± 4 LSB Maksimum INL	± 1 LSB Maksimum INL	± 4 LSB Maksimum INL

FPGA ile programlanma kolaylığı açısından, entegre devrenin veri genişliğinin 16 bit seçilmesine karar verildiğinden AD 5791 entegre devresi elenmiştir. AD 5372 ile AD 5764 arasında yapılan seçimde çıkış gerilimi aralığına, yatışma süresinin uzunluğuna, INL ve DNL değerlerine dikkat edilmiştir. Bu özellikler bakımından AD

5764 entegre devresi daha üstün olduğundan D/A çevirici kartı tasarımında bu entegre devre kullanılmıştır.

2.2 D/A Çevirici Balkon Kartı Tasarımı

Tasarlanan ve üretilen D/A çevirici balkon kartı, kullanılmakta olan Xilinx marka ML 507 model numaralı geliştirme kartının, giriş/çıkış (I/O) konektörlerinden birisine takılarak kullanıma uygun şekilde geliştirilmiştir.

Xilinx ML 507 geliştirme kartının genel özellikleri aşağıdaki gibidir;

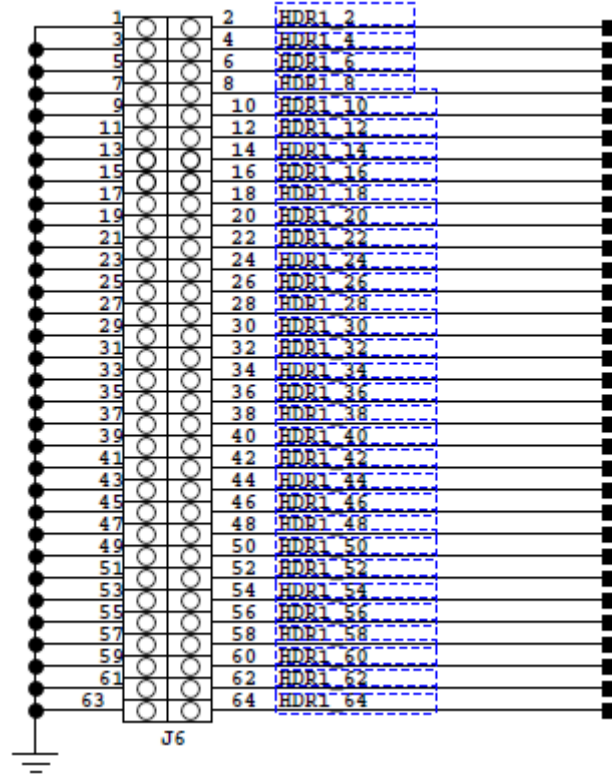
- Virtex 5 - XC5VFX70TFFG1136 FPGA
- 256 MB DDR2 SO-DIMM
- 1 MB SRAM
- 32 MB Linear Flash
- SPI Flash
- JTAG Programlama Arayüzü
- Dış saat
- Erkek RS-232 port (bağlantı noktası)
- RJ-45 10/100/1000 Ethernet
- Video girişi
- Video çıkışı (DVI/VGA)
- Tek uçlu ve türevsel uçlu I/O konektörleri
- 8 adet LED, 8 adet basma düğmesi ve 8 adet anahtar

ML 507 geliştirme kartının görünümü Şekil 2'deki gibidir.



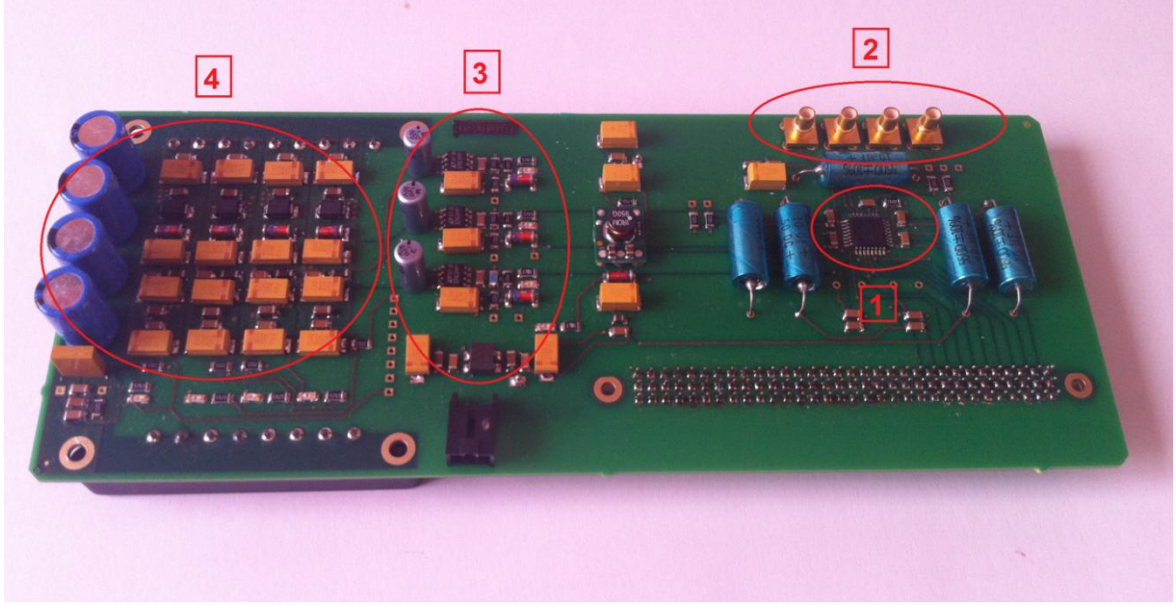
Şekil 2. ML 507 Geliştirme Kartı [18]

D/A çevirici balkon kartı "1" ile gösterilen I/O konektörüne takılarak kullanılacak şekilde tasarlanmıştır. Bu konektörün geliştirme kartı üzerindeki ismi "J6" konektörüdür ve bağlantıları Şekil 3'de gösterilmiştir. Tek sayı ile numaralandırılmış hatlar toprak hatlarıdır. Çift sayı ile numaralandırılmış HDR1 hatları FPGA I/O hatlarıdır ve programlanabilmektedirler. D/A çevirici balkon kartı tasarımında bu hatlardan "2"- "20" aralığında bulunan I/O hatları kullanılmıştır.

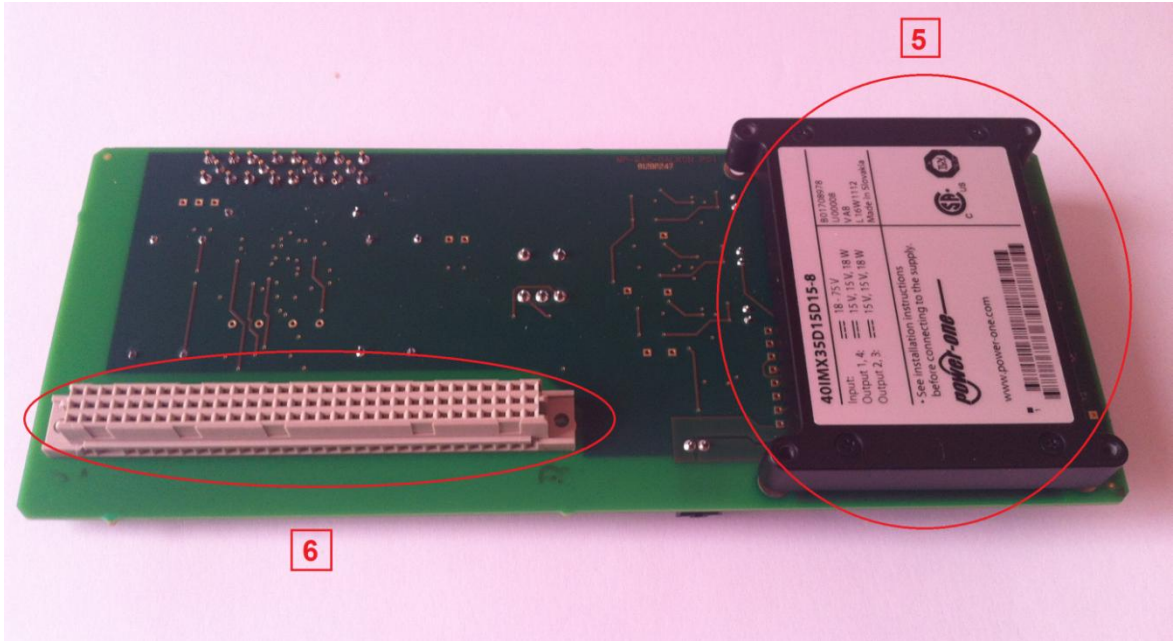


Şekil 3. ML507 J6 Konektör Bağlantıları [19]

AD 5764 entegre devresi veri sayfası incelendikten sonra entegre devre bacaklarının bağlantıları için gerekli hatlar belirlenmiş ve mevcut örnek tasarım incelenmiştir. Veri sayfasında verilen örnek tasarım Şekil 4'de gösterilmiştir.



Şekil 5. D/A Çevirici Balkon Kartı Ön Yüzü



Şekil 6. D/A Çevirici Balkon Kartı Arka Yüzü

Şekil 5'de "1" ile numaralandırılmış alanda AD 5764 entegre devresi ve çevre kapasitörleri bulunmaktadır. D/A çevirici entegre devresinin 4 adet çıkış kanalı bulunduğundan, bu çıkış hatlarını taşımak amaçlı 4 adet SubMiniature version B (SMB) koaksiyel konektör kullanılmıştır, bu konektörler Şekil 5'de "2" numaralı alanda bulunmaktadır. AD5764 entegre devresi ve çevre kapasitörleri ile çıkış kanallarının Mentor Graphics çizimleri "Ekler" bölümünde Şekil 48 ile gösterilmiştir.

Çıkış kanalları Şekil 48'de KN2, KN3, KN4 ve KN5'e denk gelmektedir. Şekil 5'de "3" ile numaralandırılmış alanda ± 14 V ve ± 5 V değerinde gerilimler elde etmek için Linear Technology markasına ait LT1743 gerilim düzenleyici entegre devreleri, 3,3 V değerinde gerilim elde etmek içinse Texas Instruments firmasına ait PTH04070W gerilim düzenleyici entegre devresi bulunmaktadır, bu blokların devre tasarım çizimleri Şekil 49'da gösterilmiştir. Bu alanda üretilen doğru akım (DC) gerilimleri, D/A çevirici entegre devresinin bacaklarına gitmektedir. Kartın arka yüzünde bulunan DC-DC çeviricinin çıkış gerilimini filtrelemek için gerekli kapasitörler ve zener diyotlar ise Şekil 5'de "4" numaralı alanda bulunmaktadır, bu alanda bulunan devre elemanlarının tasarım çizimleri Şekil 50'de gösterilmiştir.

Şekil 6'da "5" ile numaralandırılmış alanda Power-One firmasına ait 40IMX35 D15D15 model numaralı programlanabilir DC-DC çevirici bulunmaktadır. Bu çevirici ve çıkış devre elemanlarının tasarımı Şekil 51'de gösterilmiştir. DC-DC çevirici 18-75 V aralığında giriş gerilimlerine uygun olduğundan, D/A çevirici balkon kart bu aralıktaki gerilimler ile çalışabilmektedir.

Ayrıca kartta çıkabilecek sorunları gözlemlemek ve test edebilmek amacıyla test noktaları ve ledler, ters akımları engellemek için diyotlar ve DC-DC çevirici modülüne güç sağlamak için VBAT giriş konektörü kullanılmıştır. Bu elemanların tasarımı ise Şekil 52'de gösterilmiştir.

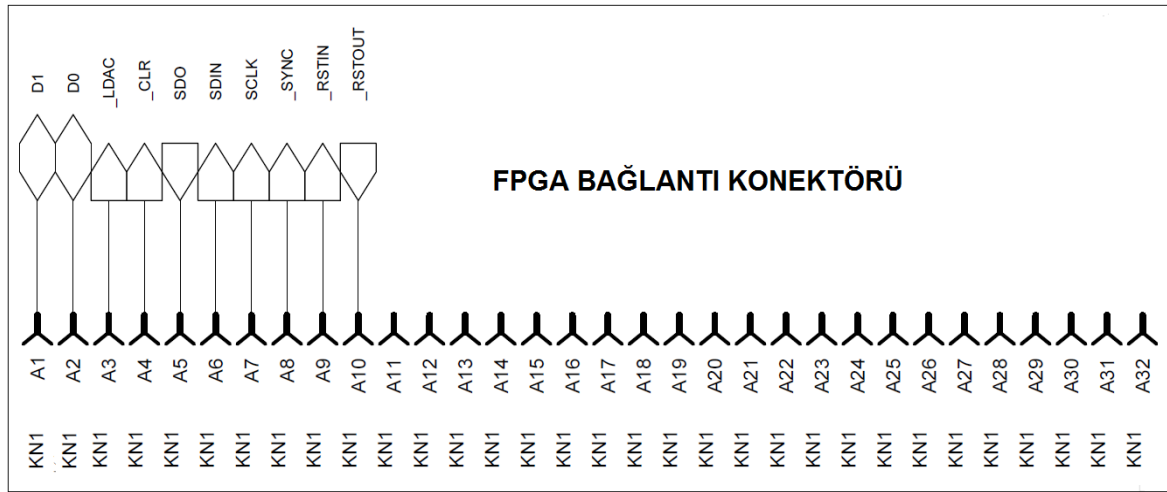
ML 507 geliştirme kartının J6 konektörüne takılarak, ML 507 üzerinde bulunan FPGA ile D/A çevirici balkon kartın iletişimini sağlamak üzere, Pancon firmasına ait 32 pimli dişi konektör seçilmiş ve kullanılmıştır. Bu konektör Şekil 6'da "6" numaralı alanda bulunmaktadır ve tasarımı Şekil 7'de gösterilmiştir.

2.3 D/A Çevirici Balkon Kartın Programlanması

2.3.1 D/A Çevirici Entegre Devresinin Veri Sayfalarının ve Zamanlama Şemalarının İncelenmesi

Balkon kartın programlanması için öncelikle AD 5764 entegre devresinin veri sayfaları ve zamanlama şemaları incelenmiştir. D/A çevirici entegre devresinin

programlanmasında Şekil 4'de gösterilen D/A çevirici bacaklarından D1, D0, _LDAC, _CLR, SDO, SDIN, SCLK, _SYNC, _RSTIN, _RSTOUT pimleri programlamada aktif olarak kullanılmaktadır. Bu pimlerden önüne "_" eklenmiş olanlar alt seviyede aktif, diğerleri üst seviyede aktiftir. Bu pimler D/A çevirici balkon kartını ML 507 geliştirme kartına bağlayan KN1 konektörüne, özelliklerine göre giriş, çıkış ya da giriş-çıkış bağlantıları olarak tanımlanmıştır. Bu bağlantılar Şekil 7'de gösterilmiştir.



Şekil 7. KN1 Konektörü Bağlantıları

Bu bağlantıların işlevleri aşağıda listelenmiştir:

D1 ve D0: Bu hatlar giriş ya da çıkış hattı olarak programlanabilen sayısal giriş/çıkış bağlantılarıdır.

_LDAC: Bu hat D/A çevirici veri yazmaçlarını ve analog çıkışları güncellemek için kullanılan giriş hattıdır. Kalıcı olarak toprağa bağlandığı durumda adreslenen veri yazmacı _SYNC hattının yükselen kenarında güncellenir. _LDAC hattı bağlantısız bırakılmamalıdır.

_CLR: Bu hat düşen kenarda tetiklenmektedir. Bu bağlantı tetiklendiğinde veri yazmaçlarına 0x0000 değeri yazılır. Bu hat herhangi bir değere atanmadığı durumda sayısal "1" değerini alır.

SDO: Bu hat sadece daisy-chain ve geri okumalı modunda seri yazmaçlar için saat işareti olarak kullanılmaktadır.

SDIN: Seri veri giriş hattıdır. Veri SCLK'nın düşen kenarında geçerli olmalıdır.

SCLK: Seri saat işareti giriş hattıdır. Veri SCLK'nın düşen kenarında giriş yazmaçlarına gönderilir. SCLK maksimum 30 MHz hızda kullanılabilir.

_SYNC: Seri arayüz için senkronizasyon sinyalidir. Bu hat alt seviyedeyken, veri SCLK sinyalinin düşen kenarında transfer edilir.

_RSTIN: Sayısal yeniden başlatma giriş sinyali. Bu giriş sinyali entegre devrede bulunan reset devresine erişimi sağlar. Bu sinyal sayısal "0" değerine çekildiğinde D/A çevirici çıkışını "0" volt değerine çeker. Normal çalışma koşulunda sayısal "1" değerine bağlı olmalıdır. Bu sinyalin değişiminden yazmaç kayıtları etkilenmez.

_RSTOUT: Entegre devrede bulunan reset devresinde kullanılan gerilim gözetleme sinyalidir. Dilenirse diğer sistem bileşenlerini kontrol etmek için de kullanılabilir.

AD5764 entegre devresinin giriş kodu ile çıkış gerilimi arasındaki ideal ilişki Çizelge 2 ile gösterilmiştir. AD5764 entegre devresinin 32. pimi sayısal toprağa bağlandığı durumda kodlama şekli ikiye tümleyen, bu pim sayısal DV_{CC} değerine bağlandığında ise kodlama ikili aritmetik tabanda ofsetli olarak yapılmaktadır. DV_{CC} değeri 2,7 volt ile 5,25 volt aralığındadır, tasarımda 32. pime 3,3 volt bağlandığından kodlama kaydırılmış ikili aritmetik tabandadır.

Çizelge 2. İkili Tabanda Ofsetli Kodlama

Sayısal Giriş		Analog Çıkış
MSB	LSB	$V_{Çıkışx}$
1111 1111 1111 1111		$+2 V_{REF} \times (32,767/32,768)$
1000 0000 0000 0001		$+2 V_{REF} \times (1/32,768)$
1000 0000 0000 0000		0 V
0111 1111 1111 1111		$-2 V_{REF} \times (1/32,768)$
0000 0000 0000 0000		$-2 V_{REF} \times (32,767/32,768)$

* $V_{REF} = 5 V$

Entegre devrenin 25. ve 26. pimlerine bağlanan gerilim değeri V_{REF} değerini belirlemektedir. Bu tasarımda V_{REF} değeri 5 voltur.

AD5764 entegre devresinin çıkış gerilim denklemi aşağıdaki gibidir;

$$V_{\text{çıkış}} = -2 \times V_{REF} + 4 \times V_{REF} \left[\frac{D}{65,536} \right] \quad (1)$$

Tasarımda V_{REF} değeri olarak 5 volt kullanıldığından, bu tasarım için çıkış gerilim denklemi aşağıdaki gibi indirgenebilir;

$$V_{\text{çıkış}} = -10 + 20 \left[\frac{D}{65,536} \right] \quad (2)$$

Entegre devrenin içerisinde bir kayan giriş yazmacı bulunmaktadır, bu yazmaçtan SDIN pimine veriler bitler halinde, seri olarak gitmektedir. Bu yazmaç 32 bittir ve yazmacın bit haritası Çizelge 3 ile verilmiştir.

Çizelge 3. Giriş Yazmacı Bit Haritası [20]

DB23	DB22	DB21	DB20	DB19	DB18	DB17	DB16	DB15:DB0
R/ \bar{W}	0	REG2	REG1	REG0	A2	A1	A0	Veri

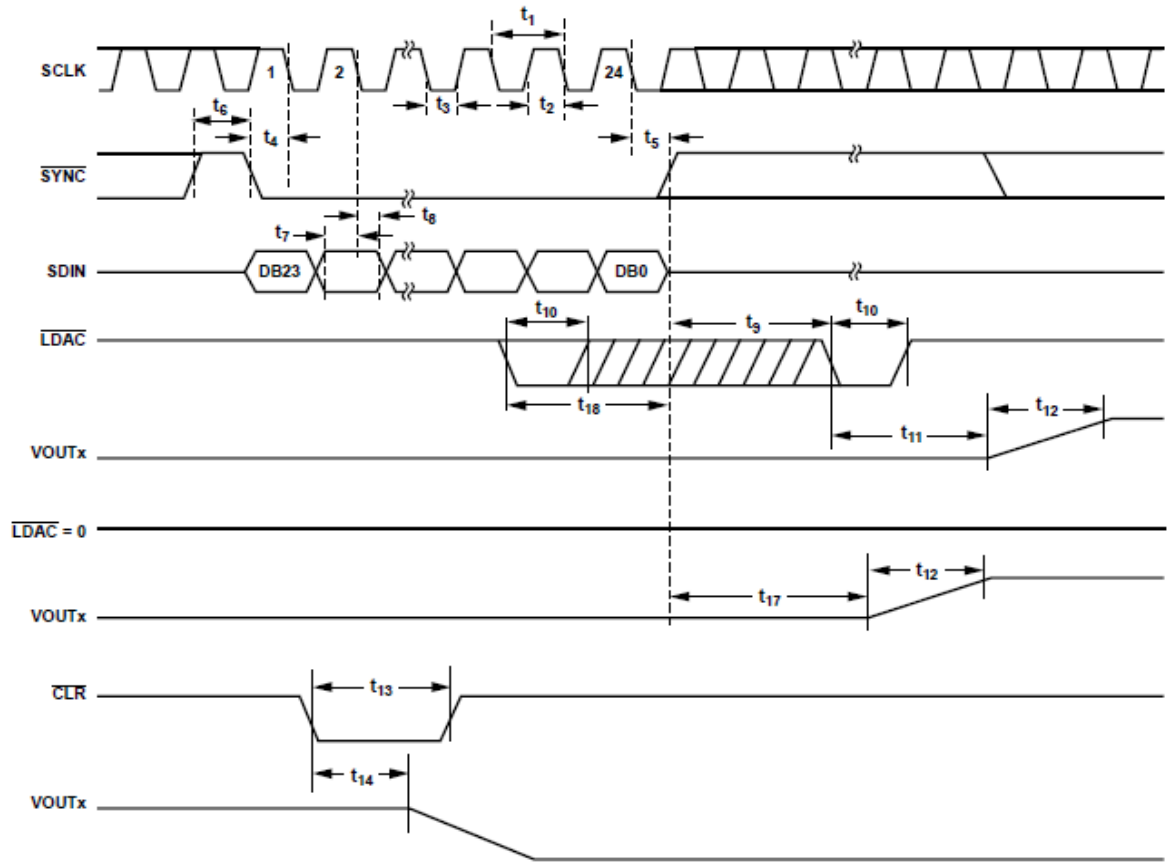
Bu giriş yazmacının bitlerinin tanımları Çizelge 4 ile verilmiştir.

Çizelge 4. AD5764 Giriş Yazmacı Bit Tanımları [20]

Bit	Tanım
R/ \bar{W}	R adreslenen yazmaçtan okuma işlemini, \bar{W} ise adreslenen yazmaca yazma işlemini ifade eder.
REG2,REG1,REG0	Adres bitleriyle birlikte yazma ya da okuma işleminin hangi yazmaca yapılacağını belirlemede kullanılır.

	REG2	REG1	REG0	Fonksiyon
	0	0	0	Fonksiyon yazmacı
	0	1	0	Veri yazmacı
	0	1	1	Kaba kazanç yazmacı
	1	0	0	İnce kazanç yazmacı
	1	0	1	Ofset yazmacı
A2,A1,A0	Bu bitler sayısaldan analoga çevirici (DAC) kanallarını kodlamak için kullanılır.			
	A2	A1	A0	Kanal Adresi
	0	0	0	DAC A
	0	0	1	DAC B
	0	1	0	DAC C
	0	1	1	DAC D
	1	0	0	Bütün DAC'lar
Veri	Veri bitleri			

AD 5764 entegre devresi seri arayüz ile programlanmış, geri okumalı modlar kullanılmamıştır. Seri arayüz için kullanılan zamanlama şeması Şekil 8'de verilmiştir. Bu şemada gösterilen zamanlama parametrelerinin açıklamaları ve limit değerleri Çizelge 5'deki gibidir (t_{15} ve t_{16} seri arayüz zamanlama şemalarında bulunmadığından çizelgede yer almamaktadır).



Şekil 8. AD5764 Seri Arayüz Zamanlama Şeması [20]

Çizelge 5. AD5764 Seri Arayüz Zamanlama Parametreleri [20]

Parametre	Limit	Birim	Tanım
t_1	33	ns min.	SCLK döngü süresi
t_2	13	ns min.	SCLK yüksek süresi
t_3	13	ns min.	SCLK düşük süresi
t_4	13	ns min.	SCLK düşen kenarı ile _SYNC düşen kenarı arası süre
t_5	13	ns min.	SCLK'nın 24. düşen kenarı ile _SYNC yükselen kenarı arası süre
t_6	90	ns min.	_SYNC yüksek süresi
t_7	2	ns min.	Veri yapılanma süresi
t_8	5	ns min.	Veri tutma süresi

t ₉	1.7	µs min.	_SYNC yükselen kenarı ile _LDAC düşen kenarı arası süre
t ₁₀	10	ns min.	_LDAC düşük süresi
t ₁₁	500	ns maks.	_LDAC düşen kenarı ile çıkış tepkisi arası süre
t ₁₂	10	µs maks.	DAC çıkış yatışma süresi
t ₁₃	10	ns min.	_CLR düşük süresi
t ₁₄	2	µs maks.	_CLR çıkış aktivasyon süresi
t ₁₇	2	µs maks.	_SYNC yükselen kenarı ile DAC çıkış tepkisi arası süre (_LDAC=0)
t ₁₈	170	ns min.	_LDAC düşen kenarı ile _SYNC yükselen kenarı arası süre

2.3.2 D/A Çevirici Entegre Devresi İçin Akış Diyagramı Oluşturulması ve Entegre Devresinin Programlanması

AD 5764 entegre devresinin veri sayfaları incelenip, zamanlama şemaları ve parametreleri çıkarıldıktan sonra programlama aşamasına geçilmiştir. Entegre devrenin programlanması için HDL Designer (2006.1) programı kullanılmıştır. Çok yüksek hızlı entegre devrelerin donanım tanımlama dili (VHDL) kullanılarak doğrudan yazımsal kodlama yapılmasındansa HDL Designer programı kullanılmasının nedeni, bu program ile akış diyagramları kullanılarak zamanlama diyagramlarının pratik şekilde kodlanabilmesidir.

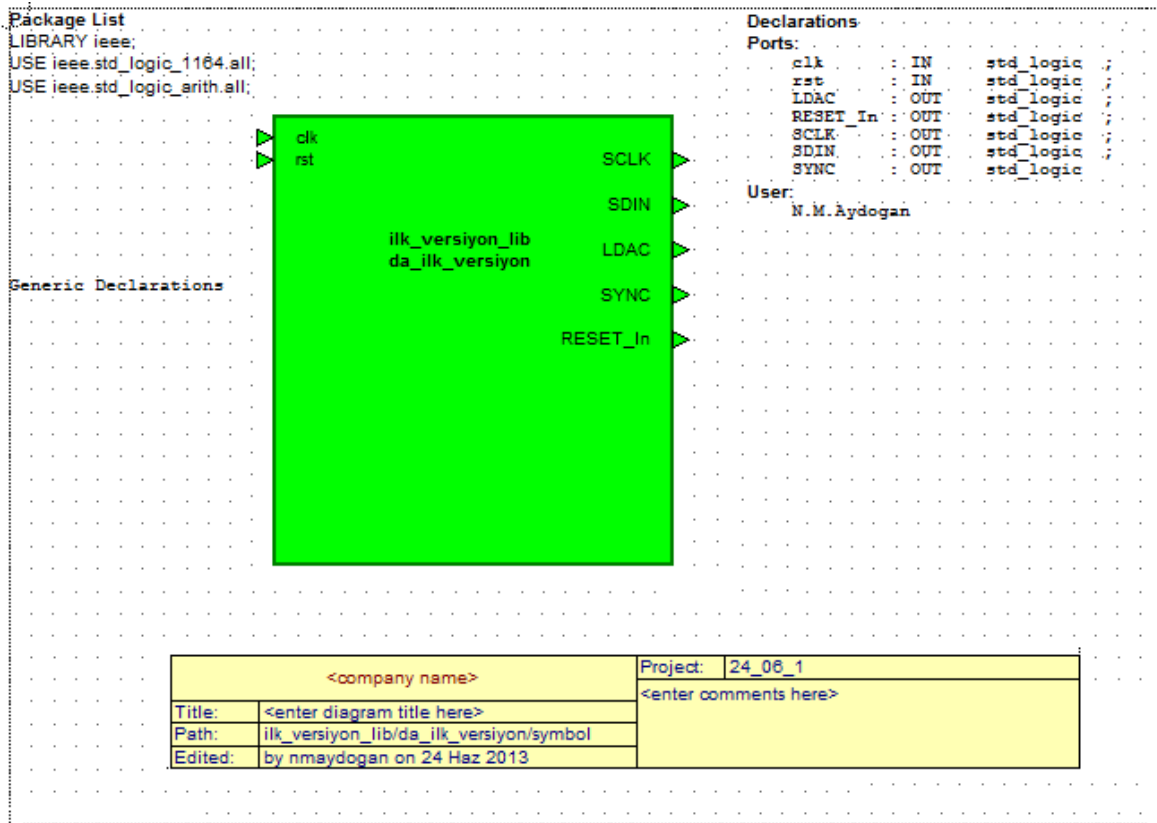
HDL Designer programı,

- FPGA ve ASIC akış diyagramları oluşturabilme
- Oluşturulan akış diyagramlarını VHDL, Verilog ve Sistem Verilog dillerine çevirebilme
- Akış diyagramlarının oluşturulmasında tablo, metin, şema gibi kullanıcı dostu araçlar sağlama
- Yukarı ve aşağı yöndeki işlemler için esneklik
- Otomasyon ile diyagram ve kod oluşturma süresini, dolayısıyla tasarım süresini kısaltma

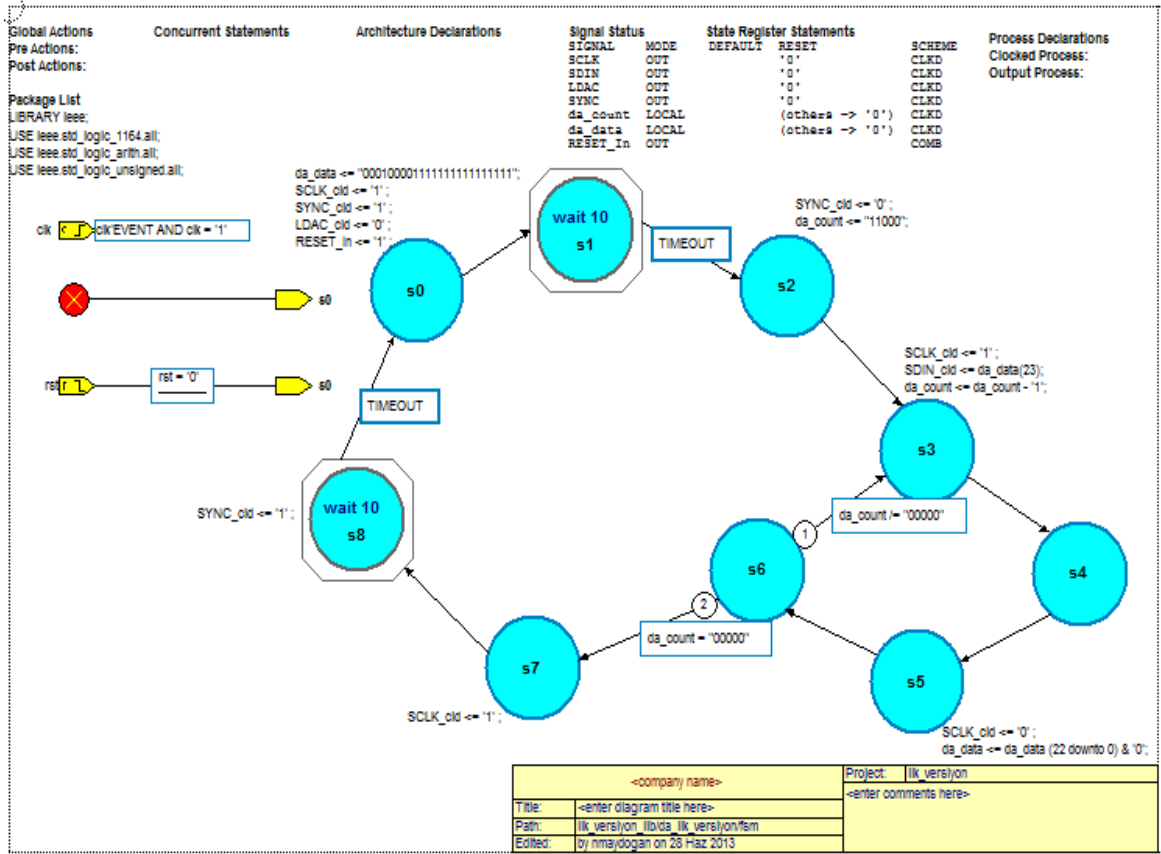
- RTL kodlama desteği ile esnek tasarım kontrolü
- Kodların ve diyagramların yeniden kullanılabilirliğini artırmak için pratik dokümantasyon

özellikleri ile kodlamada kolaylık sağlamıştır.

HDL Designer programında VHDL diline dönüştürülmek üzere, AD5764 zamanlama şema ve parametrelerine uygun akış diyagramı hazırlanmıştır. Bu ilk tasarım örneğinde zamanlama parametreleri, tasarımın ilk doğrulaması olduğundan minimum değerlerinin üzerinde kullanılmıştır. Doğrulanmış ilk tasarımın giriş ve çıkış sinyallerini gösteren blok şeması Şekil 9 ile akış diyagramı ise Şekil 10 ile gösterilmiştir.



Şekil 9. D/A Çevirici Kodu Blok Şeması



Şekil 10. İlk Tasarım D/A Çevirici Kodu Akış Diyagramı

Şekil 9'da görüldüğü gibi giriş sinyalleri "clk" ve "rst", çıkış sinyalleri SCLK, SDIN, LDAC, SYNC, RSTIN'dir. Yine aynı şekilde gösterilen "clk" sinyali yükselen kenar ile "rst" sinyali düşen kenar ile tetiklenmektedir. Akış diyagramı "s0" durumu ile başlamaktadır. Saat girişi olarak FPGA'dan direkt çıkan 33 MHz saat kullanılmıştır, dolayısıyla bir saat işareti çevrimi yaklaşık 30,3 ns sürmektedir.

AD 5764 entegre devresi programlanırken öncelikle giriş yazmacına yazılacak değere karar verilmiştir. Çizelge 3 ve Çizelge 4'de gösterilen değerlerden yola çıkılarak, yazma işlemi yapılacağından DB23 biti "0" olarak kullanılacaktır. Verilerin sayısalardan analoga dönüştürülmesi için veri yazmacı kullanılacak, dolayısıyla DB21-DB20-DB19 bitleri "010" olacaktır. Tasarımda DAC kanallarından "A" ile isimlendirilmiş kanal kullanılmıştır, dolayısıyla DB18-DB17-DB16 bitleri "000" olacaktır. Bahsedilen ayarların kullanımıyla birlikte ilk 8 bitin "00010000" olarak kullanılacağı belirlenmiştir. İlk tasarım için 10 volt çıkış gözlemlenmek istediğinden veri bitleri için Çizelge 2'deki değerlerden "1111111111111111" seçilmiştir. Sonuç olarak giriş yazmacına yazılacak değer Çizelge 6 ile gösterilmiştir.

Çizelge 6. AD5764 Giriş Yazmacına Yazılacak Değer

DB23	DB22	DB21	DB20	DB19	DB18	DB17	DB16	DB15:DB0
0	0	0	1	0	0	0	0	1111111111111111

Çizelge 6 ile gösterilen değer Şekil 10'da gösterilen akış diyagramının "s0" durumunda "da_data" isimli dizine yazılmıştır. Aynı durumda LDAC sinyali "0" değerine çekilmiştir ve tasarım boyunca LDAC değeri düşük seviyede kullanılacaktır. Ayrıca RSTIN sinyali de sayısal "1" değerine bağlanmıştır ve reset istenmediği tüm durumlarda bu şekilde kalacaktır. SCLK ve SYNC sinyalleri de yine bu durumda sayısal "1" değerine çekilmiştir.

Bekleme durumu olan "s1" isimli durumda Şekil 8'de gösterilen "t6" değerinin sağlanması için 10 çevrimlik bir bekleme süresi tanımlanmıştır. Bu durumun ardından gelen "s2" durumunda SYNC sinyali "0" değerine çekilmiş, aynı zamanda giriş yazmacına yazılacak bitler için döngü oluşturmak amacıyla 23 sayısı atanan bir da_count dizini oluşturulmuştur. Bundan sonraki dört durum bir döngü oluşturmaktadır ve bu döngü da_count dizinine atanan sayıdan bir fazla sayıda, yani 24 kez tekrarlanmaktadır. "s3" durumunda da_count dizindeki sayı her döngüde bir azaltılmakta, giriş yazmacının son biti SDIN sinyaline atanmakta ve SCLK sinyali yüksek değerine çekilmektedir. "s4" durumu "t2", "t4" ve "t7" sürelerini tamamlamak amaçlı, bir çevrimlik bekleme sağlamak için eklenmiştir. "s5" durumunda SCLK sinyali düşük değerine çekilmiş ve da_data dizini bir bit sola kaydırılmıştır. Bu bit kaydırma işlemi sayesinde 24 döngüde da_data dizininin her biti bir kez dizinin son biti olmuş ve SDIN sinyaline atanmıştır. "s6" durumu "t3", "t5" ve "t8" sürelerini tamamlamak amaçlı eklenmiştir. "s6" durumundayken da_count dizininin değeri sıfırlanmamışsa döngü sıfırlanana kadar devam eder, bu değer sıfırlanmışsa döngü biter ve yeni durum olan "s7" ye geçilir. "s7" durumunda SCLK değeri yüksek değerine çekilir. Bir sonraki durum olan "s8" de SYNC sinyali sayısal "1" değerine çekilerek yazma işlemi tamamlanır. Bu durumda aynı zamanda 10 çevrimlik bir bekleme süresi tanımlanmıştır, bu bekleme süresi "t17"+"t12" süresinin tamamlanması içindir. Akış diyagramı, "s8" durumunda tanımlanmış olan bekleme süresi bittikten sonra otomatik olarak başlama durumu olan "s0" a geçmektedir.

Bu akış diyagramında toplam 119 çevrim bulunmaktadır, her çevrim 30,3 ns sürmektedir. Bu da bu akış diyagramı ile hazırlanmış VHDL kodu kullanıldığında D/A çevirici entegre devresinin maksimum;

$$\frac{1}{119 \times 30.3 \text{ ns}} = 277.34 \text{ KHz} \quad (3)$$

frekansıyla çalışabileceğini göstermektedir.

Akış diyagramı tamamlanıp, frekans değeri belirlendikten sonra diyagram HDL Designer ile VHDL koduna çevrilmiştir. ML 507 geliştirme kartının şematikleri incelenerek tüm sinyaller uygun değerlere atanmış, VHDL kodunun FPGA'ya yüklenebilmesi için gerekli olan kullanıcı kısıtlamaları dosyası (UCF) hazırlanmıştır.

Bu işlemlerin yapılmasının ardından hazırlanan kod FPGA'ya yüklenmiş, D/A çevirici balkon kartına güç verilerek "A" kanalı osiloskopa bağlanmıştır. Osiloskopta 10 volt değerindeki analog DC gerilim gözlemlenerek tasarım ve kod doğrulanmıştır.

Ayrıca ChipScope Pro isimli simülasyon programı kullanılarak Şekil 11'deki simülasyon elde edilmiştir. Simülasyonda "X" ve "O" ölçüm doğruları arasında kalan bölüm akış diyagramının bir tam döngüsüdür. Bu simülasyondaki zamanlama değerlerinin Şekil 8'deki zamanlama değerleri ile uyumlu olduğu doğrulanmıştır. Sinyallerin akış diyagramındaki karşılıkları aşağıdaki gibidir:

LDAC_OBUF: LDAC

RESET_In_OBUF: RESET_In

SCLK_cld: SCLK

SDIN_cld: SDIN

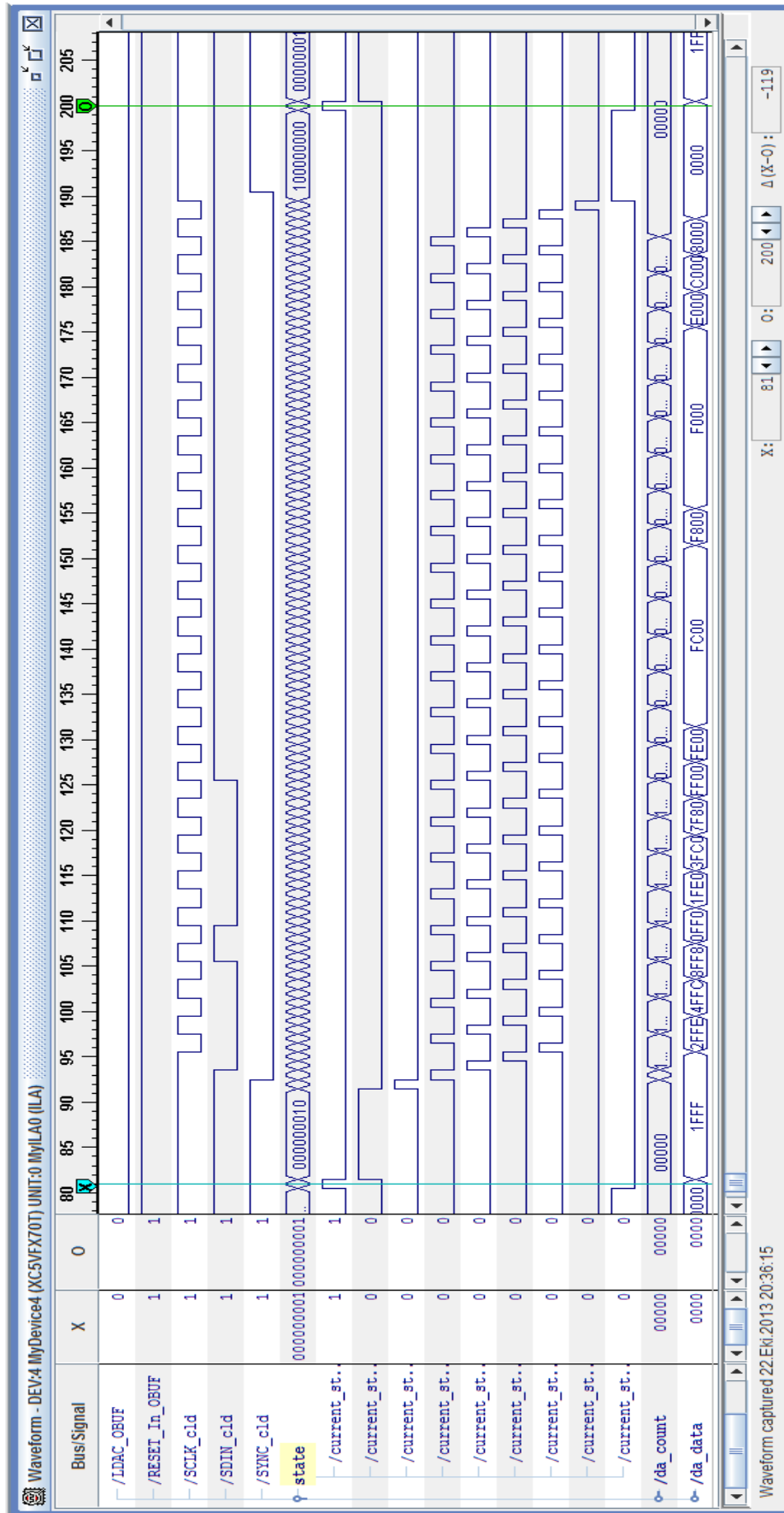
SYNC_cld: SYNC

da_count: da_count (giriş yazmacına yazılacak bit sayısı toplamını belirten dizin)

da_data: da_data (giriş yazmacına yazılacak veriyi tutan dizin)

state: Durumların kodlanmış halidir. Akış diyagramında s0-s8 arasında toplam 9 adet durum vardır ve bu durumlar simülasyonda şu şekilde kodlanmıştır;

- s0: 000000001
- s1: 000000010
- s2: 000000100
- s3: 000001000
- s4: 000010000
- s5: 000100000
- s6: 001000000
- s7: 010000000
- s8: 100000000



Şekil 11. D/A Çevirici Kodu ChipScope Simülasyonu Sonuçları

Bu şekilde optimize edilen yeni akış diyagramında toplam 57 çevrim bulunmaktadır. Her çevrim 30,3 ns olduğundan, bu akış diyagramı ile kodlanmış D/A çevirici entegre devresinin maksimum çalışma frekansı;

$$\frac{1}{57 \times 30.3 \text{ ns}} = 579 \text{ KHz} \quad (4)$$

olarak belirlenmiştir.

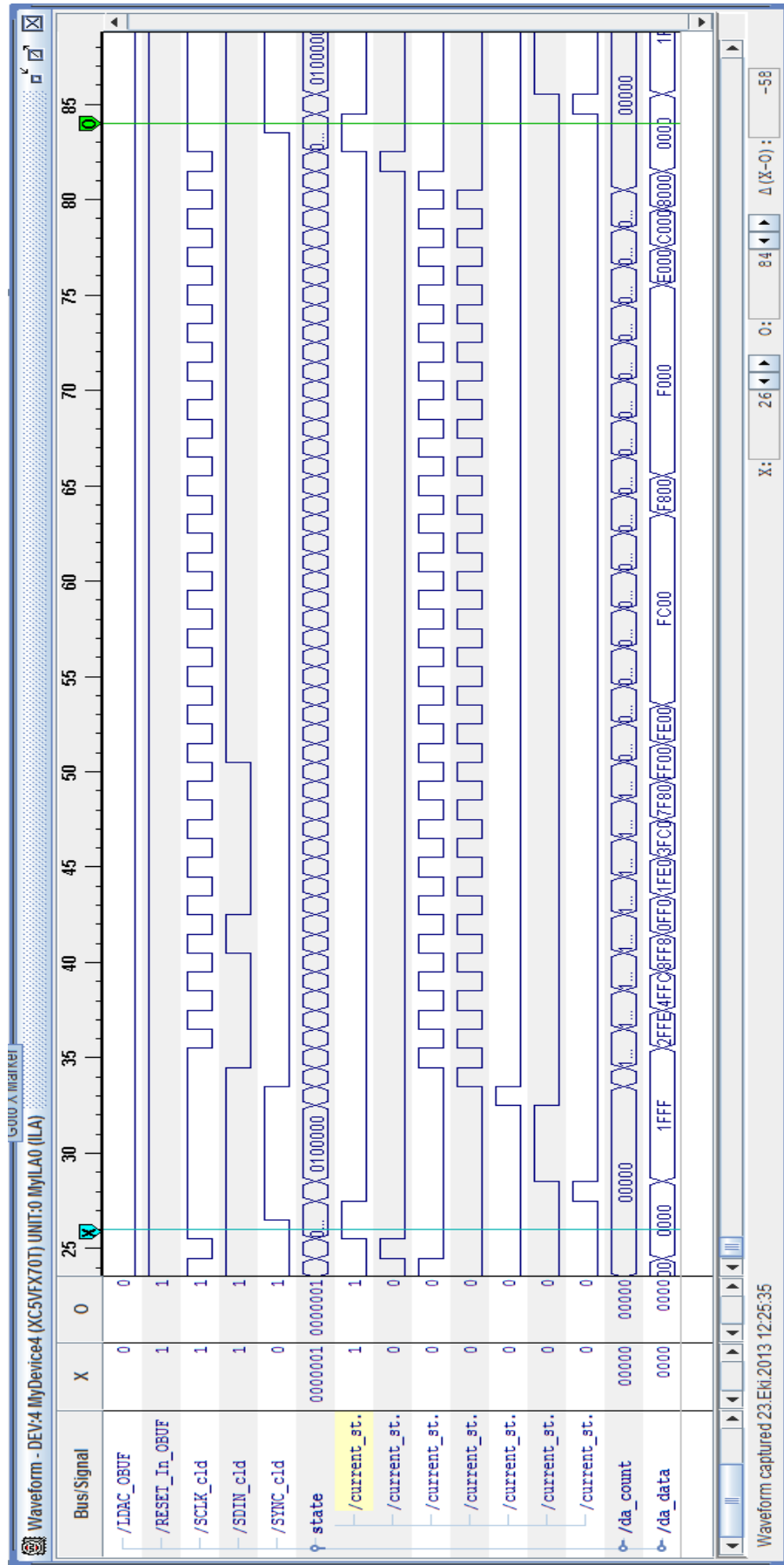
Optimize edilmiş olan bu akış diyagramı ile hazırlanan VHDL kodu FPGA'ya yüklenmiş, D/A çevirici balkon kartına güç verilerek "A" kanalı osiloskopa bağlanmıştır. Osiloskopta 10 volt değerindeki analog DC gerilim gözlemlenerek tasarım ve kod doğrulanmıştır.

Ayrıca ChipScope Pro isimli simülasyon programı kullanılarak Şekil 13'deki simülasyon elde edilmiştir. Simülasyonda "X" ve "O" ölçüm doğruları arasında kalan bölüm akış diyagramının bir tam döngüsüdür. Bu simülasyondaki zamanlama değerlerinin Şekil 8'deki zamanlama değerleri ile uyumlu olduğu doğrulanmıştır.

Bu simülasyondaki durum kodlamaları ise aşağıdaki gibidir;

- s0: 0000001
- s1: 0000010
- s2: 0000100
- s3: 0001000
- s4: kullanılmayan durum
- s5: 0010000
- s6: kullanılmayan durum
- s7: 0100000
- s8: 1000000

Çalışmaların devamında bu akış diyagramı ile hazırlanmış olan VHDL kodu kullanılmıştır ve D/A çevirici bloğunun maksimum frekansı 579 KHz olmuştur.



Şekil 13. Optimize Edilmiş D/A Çevirici Kodu ChipScope Simülasyonu Sonuçları

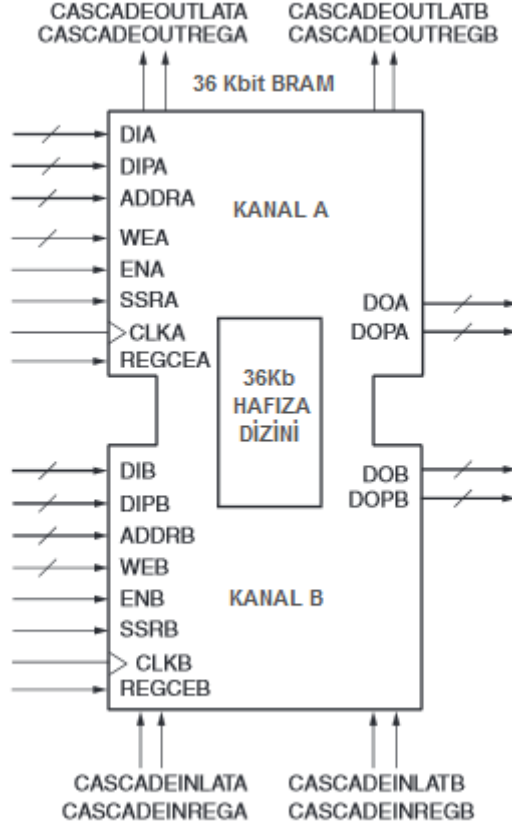
3. BAŞVURU ÇİZELGESİNİN HAZIRLANMASI

Bu çalışmadaki DDS algoritmasında başvuru çizelgesi yöntemi kullanılmıştır. Başvuru çizelgesi yöntemiyle sadece sinüsel dalgalar değil, periyodik tüm dalgalar üretilebilmektedir. DDS algoritmasının gerçekleştirilmesi için dalga verilerinin örneklenerek başvuru çizelgesine kaydedilmesi gerekmektedir. Başvuru çizelgesinin hazırlanmasında, FPGA içerisinde bulunan rastgele erişimli blok bellek (BRAM) modülleri kullanılmıştır. Bu hafıza modüllerinin büyüklükleri, kullanılan FPGA'ya göre farklılık göstermektedir. Kullanılan BRAM boyutundan yola çıkılarak adres verisinin büyüklüğü, adres verisinin büyüklüğünden de DDS algoritmasında kullanılacak olan faz birikecinin nicemlenme oranı belirleneceğinden bu bölüm DDS algoritmasından önce anlatılmıştır.

3.1 BRAM Zamanlama Şemaları ve Parametreleri

Başvuru çizelgesinin hazırlanması için öncelikle hafıza boyutlarının ve kullanılacak bit sayılarının belirlenmesi gerekmektedir. Çalışmalarda kullanılan ML507 geliştirme kartında toplam iki adet 36kbit BRAM bulunmaktadır. Her bir BRAM 18kbit'lik iki BRAM olarak kullanılabildiği gibi 36kbit'lik tek BRAM olarak da kullanılabilir. 18 kbit BRAM "RAMB18", 36 kbit BRAM "RAMB36" olarak adlandırılır.

18kbit büyüklüğündeki iki BRAM birbirinden tamamen bağımsız olarak kullanılabilmektedir ve bütün sinyal hatları "A" ve "B" olarak ayrılmıştır. BRAM yapıları simetriktir ve birbirleri ile değiştirilebilirler. Şekil 14'de iki kanallı BRAM veri akışı gösterilmiştir. Bu şemada gösterilen bağlantıların isimleri ve tanımları Çizelge 7 ile gösterilmiştir.



Şekil 14. İki Kanallı 36 Kbit BRAM Veri Akış Şeması [21]

Çizelge 7. BRAM Bağlantı Adları ve Tanımları

Bağlantı Adı	Tanım
DI[A B]	Giriş veriyolu
DIP[A B]	Giriş eşlik veriyolu (ek giriş bağlantıları olarak da kullanılabilir)
ADDR[A B]	Adres yolu
WE[A B]	Yazma etkinleştirme hattı (1 bayt)
EN[A B]	Bu hat aktif değilken BRAM'a veri yazılmaz, çıkış veriyolu önceki durumunda sabit kalır.
SSR[A B]	Senkron ayarlama / ilk durumuna getirme bağlantısı (yazmaç ya da latch modlarında)
CLK[A B]	Saat işareti girişi
DO[A B]	Çıkış veriyolu
DOP[A B]	Çıkış eşlik veriyolu (ek çıkış bağlantıları olarak da kullanılabilir)

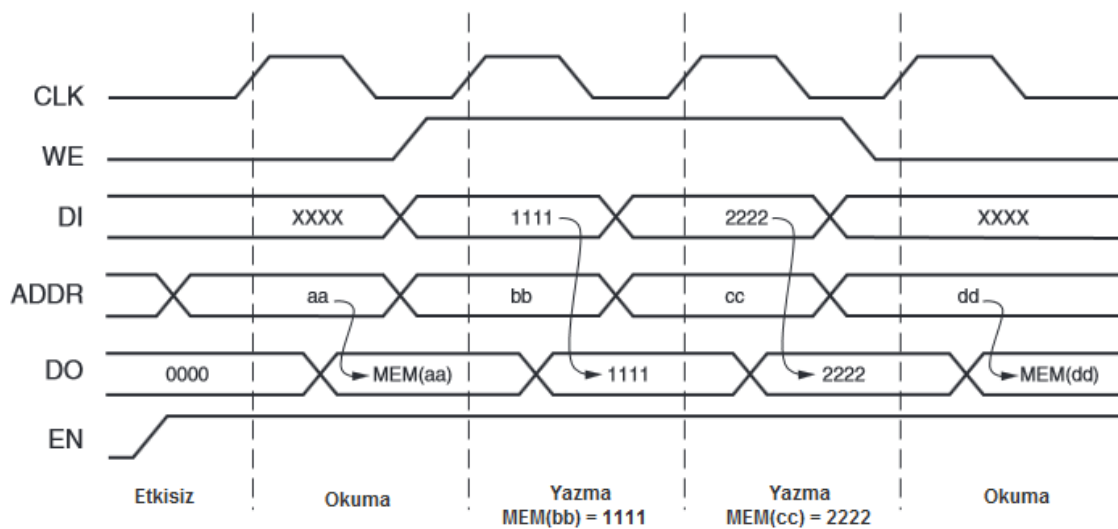
REGCE[A B]	Çıkış yazmacı etkinleştirme hattı
CASCADEINLAT[A B]	64K x 1 modu için seri giriş pimi (seçilebilir çıkış yazmaçları etkinleştirilmediğinde)
CASCADEOUTLAT[A B]	64K x 1 modu için seri çıkış pimi (seçilebilir çıkış yazmaçları etkinleştirilmediğinde)
CASCADEINREG[A B]	64K x 1 modu için seri girişi (seçilebilir giriş yazmacı etkinleştirildiğinde)
CASCADEOUTREG[A B]	64K x 1 modu için seri çıkışı (seçilebilir giriş yazmacı etkinleştirildiğinde)

Okuma işlemi latch modunda bir yazma kenarı kullanır. Okuma işlemi yapılacak adres, okuma kanalına yazılır ve saklanan veri RAM erişim süresi dolduktan sonra latching devrelerine yüklenir. Okuma işleminde çıkış yazmaçları kullanılacaksa bir çevrim gecikme yaşanır.

Yazma işlemi tek saat kenarında yapılır. Yazma işlemi yapılacak adres yazma kanalına yazılır ve giriş verisi hafızaya yazılır.

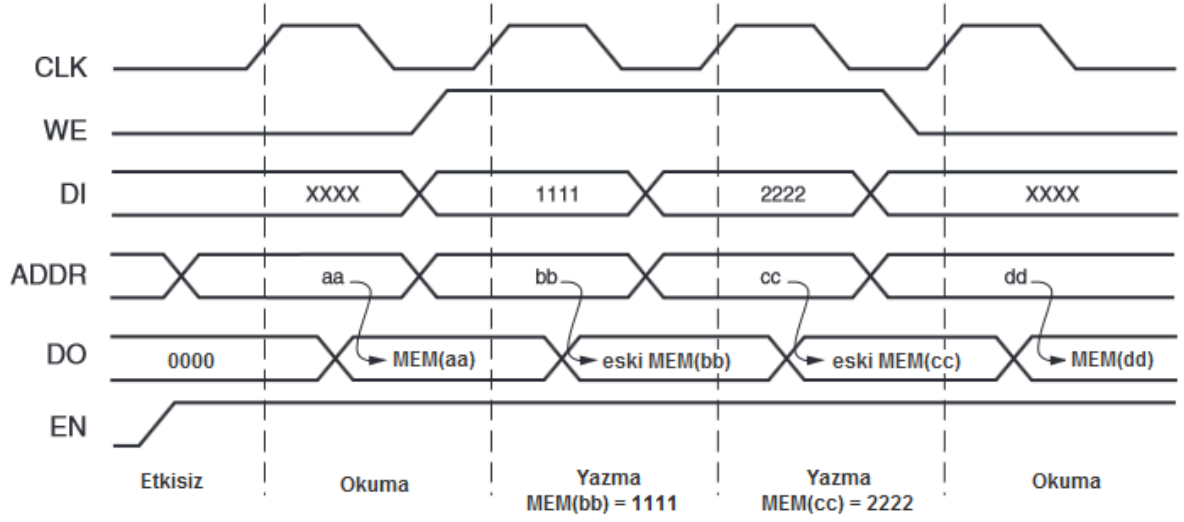
BRAM'da 3 farklı yazma modu kullanılır;

WRITE_FIRST modunda giriş veriyolundan gelen veri, çıkış veri yoluna verilirken aynı zamanda hafızaya kaydedilir. Bu mod varsayılan moddur. WRITE_FIRST modunun zamanlama şeması Şekil 15 ile gösterilmiştir.



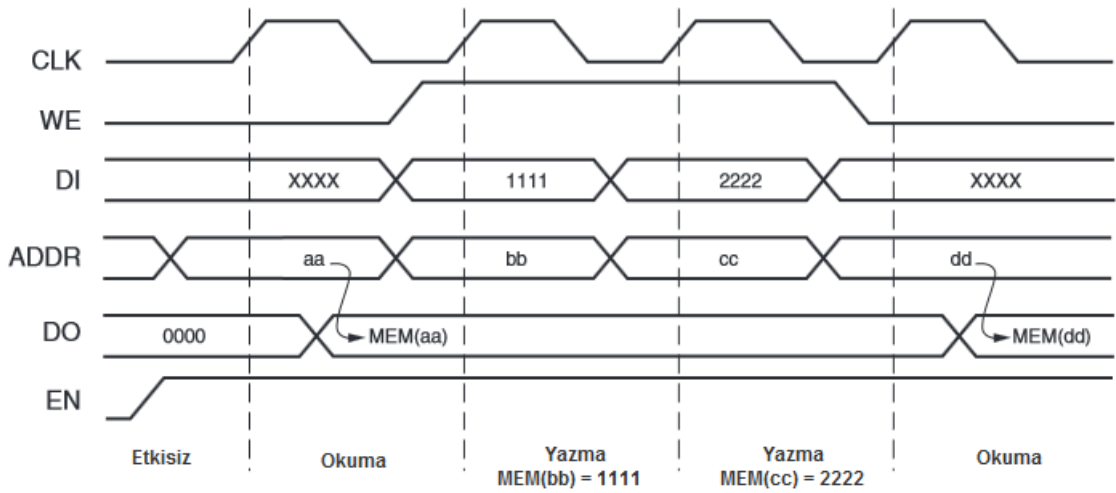
Şekil 15. WRITE_FIRST Modu BRAM Zamanlama Şeması

READ_FIRST modunda önceden saklanmış olan veri çıkış veriyoluna gönderilir, bu arada yeni veri hafızaya yazılmaktadır. Bu modun zamanlama şeması Şekil 16 ile gösterilmiştir.



Şekil 16. READ_FIRST Modu BRAM Zamanlama Şeması

NO_CHANGE modunda yazma işlemi devam ediyorsa, yazma işlemi bitene kadar çıkış veriyoluna sabit olarak son okunan veri gönderilir. Çıkış veriyolu yazma işleminden etkilenmez. Bu modun zamanlama şeması Şekil 17 ile gösterilmiştir.



Şekil 17. NO_CHANGE Modu BRAM Zamanlama Şeması

Bu çalışmada BRAM'a başvuru çizelgesi kaydedilirken, hafızada kayıtlı olan veriler yazma işlemi beklenmeksizin okunabildiğinden READ_FIRST modu kullanılmıştır [21].

3.2 BRAM Zamanlama Şemaları ve Parametreleri

Başvuru çizelgesinin hazırlanması için öncelikle hafıza boyutlarının ve kullanılacak bit sayılarının belirlenmesi gerekmektedir. Çalışmalarda kullanılan ML507 geliştirme kartında toplam iki adet 36kbit BRAM bulunmaktadır. BRAM olarak yeterli boyutta olduğundan RAMB18 kullanılmıştır. Veriyolunun genişliği giriş/çıkış veriyolunun genişliğini belirler. Bu ilişki Çizelge 8 ile verilmiştir. Tasarımda giriş/çıkış veriyolunun 16 bit olması istendiğinden veriyolu genişliği olarak "18" seçilmiştir. Veriyolu genişliği aslında 16 bittir fakat eşlik bitleri olarak 2 bit yer ayrıldığından 18 bit olarak geçmektedir. Bu tasarımda eşlik bitleri kullanılmadığından veriyolu genişliği 16 olarak alınmıştır.

Çizelge 8. RAMB18 Veriyolu Genişlikleri

Veriyolu Genişliği	Giriş/Çıkış Veriyolu (DI/DO Hatları)	Giriş/Çıkış Eşlik Veriyolu (DIP/DOP Hatları)
1	<0>	Kullanım Dışı
2	<1:0>	Kullanım Dışı
4	<3:0>	Kullanım Dışı
9	<7:0>	<0>
18 (16+2)	<15:0>	<1:0>

RAMB18 18432 hafıza hücresi içerirken RAMB36 36864 hafıza hücresi içerir. Her iki BRAM da aynı adresleme şemasını kullanır. RAMB18'in 16384 hafıza hücresi veri kaydetmek için, geri kalan 2048 hücresi ise eşlik kontrolünde kullanılmak içindir. Belli bir veriyolu genişliği için fiziksel RAM konumları aşağıdaki denklem ile belirlenir;

$$BİTİŞ = ((ADRES + 1) \times Veriyolu Genişliği) - 1 \quad (5)$$

$$BAŞLANGIÇ = ADRES \times Veriyolu Genişliği \quad (6)$$

Kullanılabilen veriyolu genişlikleri için adresleme haritaları Şekil 18 ile gösterilmiştir.

VERİYOLU GENİŞLİĞİ	EŞLİK BİTLERİ YERLERİ				VERİ BİTLERİ YERLERİ																															
1					31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
2					15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0	
4					7				6				5				4				3				2				1				0			
8+1	3	2	1	0	3								2								1								0							
16+2	1			0	1																0															

Şekil 18. BRAM Adresleme Haritaları

Veriyolu genişliği olarak (16+2) 18 bit kullanılmıştır. Bu durumda haritadan yola çıkılarak, her 16 bitlik veri bir adres aralığına denk gelmektedir. Bir başka deyişle her hafıza hücresi, 16 hücreden oluşan tek bir yeni hücreye denk gelmektedir.

RAMB18'de veri kaydetmek için bulunan 16384 hafıza hücresi, veriyolu 16 bit olarak seçildiğinde $16384 \div 16 = 1024$ yeni hafıza hücresine karşılık gelmektedir, bu hafıza hücreleri 16 bit büyüklüğünde veri saklamaktadır.

Örneğin 0-15 aralığında bulunan hücrelerde bulunan değer, 16 bitlik (2 baytlık) bir veri olarak okunacaktır. Bu veri 0. veri olacaktır ve adres aralığı yukarıdaki denklem üzerinden gidilerek;

$$BİTİŞ = ((0 + 1) \times 16) - 1 = 15 \quad (7)$$

$$BAŞLANGIÇ = 0 \times 16 = 0 \quad (8)$$

0-15 olacaktır. Eşlik bitleri kullanılmadığı için veriyolu genişliği 16 bit alınmıştır.

Aynı şekilde 1. veri 16-31 adresleme aralığına, 2. veri 32-47 adresleme aralığına, sonuncu veri olan 1023. veri 16368-16383 adresleme aralığına denk gelecektir.

RAMB18'de

Adres yolu 14 bit olduğundan;

0000000000000000 - 000000000011111 adres aralığı 0. veriyi

00000000010000 - 000000000111111 adres aralığı 1. veriyi

00000000100000 - 000000001111111 adres aralığı 2. veriyi

...

11111111110000 - 111111111111111 adres aralığı 1023. veriyi ifade etmektedir.

3.3 BRAM Veri Kaydı ve İlkeme Parametreleri

Başvuru çizelgeleri BRAM'a ilkeme yöntemi ile kaydedilmiştir. İlkeme yapılması için belli parametrelerin bilinmesi ve bu parametrelere uygun olarak VHDL kodunun yazılması gerekmektedir.

INIT_xx niteliği başlangıç hafıza hücrelerini programlar. Hafıza hücrelerinin varsayılan değerlerinin tümü sıfırdır.

RAMB18 18kbit büyüklüğündedir. Bunun 2 kbit büyüklüğündeki kısmı eşlik doğrulaması yapılacağı zamanlarda kullanılmak üzere ayrılmıştır, veri için ayrılmış kısım 16 kbit büyüklüğündedir. Her INIT_xx niteliği on altılı aritmetik sistemde 64 bit vektörünü programlar. On altılı aritmetik sistemde her bit vektörü ikili aritmetik sistemde dört bite denk geldiğinden INIT_xx ikili aritmetik sistemde 256 biti programlamaktadır. RAMB18 ikili aritmetik sistemde 16384 bit büyüklüğünde veri saklayabildiğinden, 64 adet INIT_xx niteliğine ihtiyaç duyulmaktadır. Bu nitelikler INIT_00 ile INIT_3F aralığında adlandırılmaktadır.

INIT_xx niteliğinde bit pozisyonlarını belirlemek için aşağıdaki denklem kullanılır;

$$BAŞLANGIÇ = (yy) \times 256 \quad (9)$$

$$BİTİŞ = [(yy + 1) \times 256] - 1 \quad (10)$$

(yy = xx değerinin altılı aritmetik sistemden ondalığa çevrilmiş hali)

Örneğin, INIT_1F niteliği için dönüşüm şu şekilde olacaktır;

$$yy = \text{ondalık sistemde } (1F) = 31$$

$$BAŞLANGIÇ = 31 \times 256 = 7936$$

$$BİTİŞ = [(31 + 1) \times 256] - 1 = 8191$$

Daha fazla örnek Çizelge 9 ile gösterilmiştir.

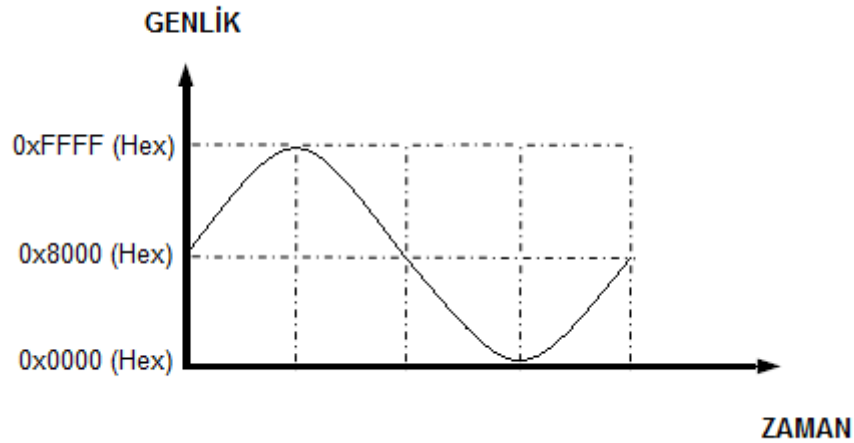
Çizelge 9. INIT_xx Niteliği Bit Pozisyonları

Nitelik	Hafıza Hücreleri Konumu	
	Başlangıç	Bitiş
INIT_00	0	255
INIT_01	256	511
INIT_02	512	767
...
INIT_0F	3840	4095
INIT_10	4096	4351
...
INIT_1F	7936	8191
INIT_20	8192	8447
...
INIT_2F	12032	12287
INIT_30	12288	12543
...
INIT_3F	16128	16383

3.4 Sinüs Verilerinin BRAM'a Kaydedilmesi

INIT_xx niteliğini kullanarak BRAM'a veri kaydı yaparken RAMB18'in "A" kanalı seçilmiştir. 16 kbit veri alanının tamamı kullanılmıştır. Veriyolu genişliği 18 seçildiğinden, her hücre 16 bit boyutundadır ve toplam 1024 hücreye veri kaydı yapılmıştır.

Kaydedilecek olan sinüs verileri oluşturulurken, AD5764 entegre devresinin Çizelge 2 ile gösterilen kodlama değerleri dikkate alınmıştır. 0 volt değeri için 0x8000 değeri, 10 volt için 0xFFFF değeri ve -10 volt için 0x0000 değeri kullanılmıştır. Aynı şekilde diğer kodlamalar da (2) numaralı denkleme uygun olarak yapılmıştır. Bu kodlamalar ile çıkarılan sinüs verisinin genel grafiği Şekil 19 ile verilmiştir.



Şekil 19. AD5764 Değerlerine Göre Kodlanmış Sinüs Grafiği

Sinüsün ilk çeyreği INIT_00 ile INIT_0F arasına, ikinci çeyreği INIT_10 ile INIT_1F arasına, üçüncü çeyreği INIT_20 ile INIT_2F arasına, dördüncü çeyreği ise INIT_30 ile INIT_3F arasına kaydedilmiştir.

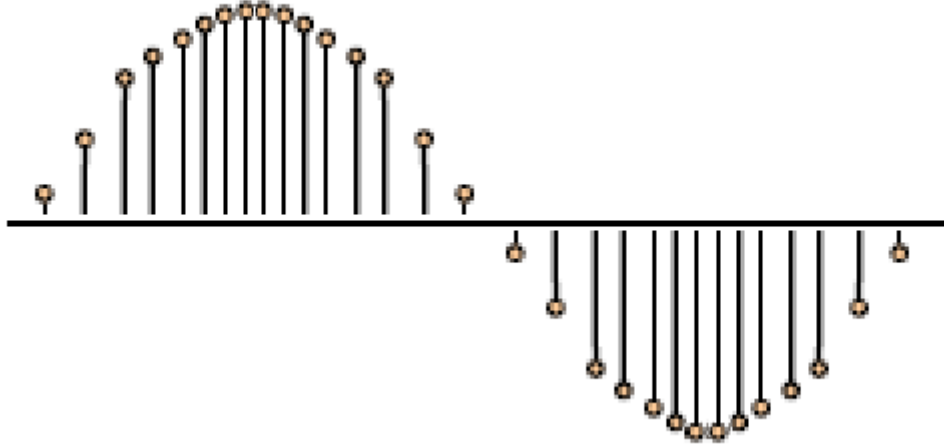
Sinüs dalga formunun sayısal veri olarak kaydedilebilmesi için belirli bir frekansla örneklenmesi gerekmektedir. Bu örnekleme sıklığına hafıza boyutu dikkate alınarak karar verilmiştir. Sinüs dalga formunun bir çeyreği 16 bitlik 256 hafıza hücresine kaydedilecek, tüm sinüs dalgası ise 16 bitlik 1024 hafıza hücresine kaydedilmiş olacaktır. Her bir çeyrekte sinüs dalgasının maksimum genlik değişimi

0x7FFF kadar olmaktadır. Bu 0x7FFF boyutundaki genlik deęiřimi 256 hafıza hücresine doruk noktalarda örnekleme frekansı artırılarak kodlanmıřtır. Kodlama örneklere Çizelge 10 ile verilmiřtir.

Çizelge 10. RAMB18'e Kaydedilen Örneklenmiř Sinüs Verisi

Hafıza Hücresi Konum	Sinüs Verisi (On Altılı Sayı Sistemi (Hex))
INIT_00	8bc48afb8a33896a88a287d987118648857f84b683ed8324825b819280c98000
INIT_01	9833976e96a895e2951c9455938f92c89201913a90728fab8ee38e1c8d548c8c
...	...
INIT_0F	ffefffdfff9fff5fff0ffe9ffe1ffd8ffcdffc1ffb4ffa6ff97ff86ff74ff61
INIT_10	ff74ff86ff97ffa6ffb4ffc1ffcdffd8ffe1ffe9fff0fff5fff9ffdfffeffff
INIT_11	fdb0fdd5fdafe1dfe3efe5ffe7efe9cfefb9fed5feefff09ff21ff37ff4dff61
...	...
INIT_1F	80c98192825b832483ed84b6857f8648871187d988a2896a8a338afb8bc48c8c
INIT_20	743b750475cc7695775d782678ee79b77a807b497c127cdb7da47e6d7f367fff
INIT_21	67cc689169576a1d6ae36baa6c706d376dfe6ec56f8d7054711c71e372ab7373
...	...
INIT_2F	000100020006000a000f0016001e00270032003e004b005900680079008b009e
INIT_30	008b007900680059004b003e00320027001e0016000f000a0006000200010000
INIT_31	024f022a020501e201c101a0018101630146012a011000f600de00c800b2009e
...	...
INIT_3F	7f367e6d7da47cdb7c127b497a8079b778ee7826775d769575cc7504743b7373

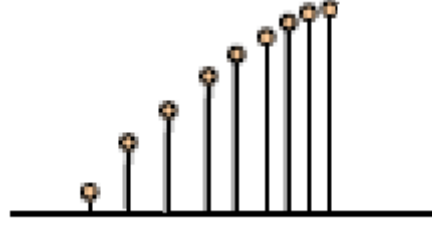
Bu şekilde 1024 adet 16 bitlik hafıza hücresi, bir başka deyişle 16 kbit hafıza kullanılarak, örneklenmiş sinüs verisi BRAM'a kaydedilmiştir. Örneklenmiş sinüs verisi Şekil 20 ile gösterilebilir. 1024 hafıza hücresi kullanıldığından BRAM adres yolunun 14 bitinin tamamı, 0-1023 aralığındaki hücreleri kodlamak için kullanılmıştır.



Şekil 20. Örneklenmiş Sinüs Dalga Formu [11]

3.5 Kullanılan BRAM Boyutunun Optimize Edilmesi

Sinüs dalga formu simetrik olduğundan , kullanılan hafıza boyutunun küçültülmesi mümkündür. Bunun için 0-90° aralığında bulunan sinüs bilgilerinden, sinüs fonksiyonunun çeyrek dalga simetrisi özelliğinden faydalanılarak 0-360° aralığındaki tüm skala bilgisi elde edilmiştir [22]. Örneklenmiş sinüs verilerinin 0-90° aralığına denk gelen ilk çeyreği Şekil 21 ile gösterilmiştir.



Şekil 21. Örneklenmiş Sinüs Dalga Formunun İlk Çeyrek Düzlemi

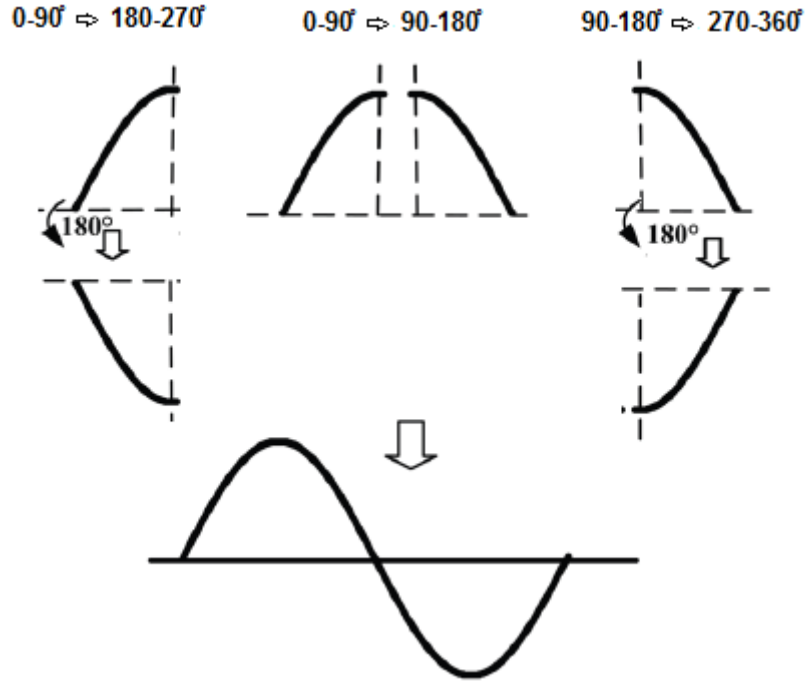
Bu çeyrek dalga verisi INIT_00 ile INIT_0F arasına yüklenerek, 256 hafıza hücresi ve $256 \times 16 = 4096$ bit hafıza kullanılmıştır. İlk çeyrek düzlemin verileri Çizelge 10'da INIT_00 ile INIT_0F arasında olduğu şekildedir. BRAM'a kaydedilen bu çeyrek veriden tüm sinüs dalgasının oluşturulabilmesi için bir algoritma hazırlanmıştır.

Bu algoritma ile ikinci çeyrek verinin oluşturulabilmesi için, ilk çeyrek veri son hafıza hücresinden başlanarak tersten okunmuş ve fonksiyon üretici bloğuna bu şekilde aktarılmıştır. Son hafıza hücresindeki veri INIT_0F niteliğinin son hücresinde bulunan 0xFFFE değeridir. Bu değerden başlanarak INIT_00 niteliğinin ilk hafıza hücresinde bulunan 0x8000 değerine kadar olan veriler okunmuştur.

Üçüncü çeyrek sinüs verisinin oluşturulabilmesi içinse INIT_00 ile INIT_0F arasına kaydedilmiş veri, BRAM'a kaydedildiği sıra ile okunmuş, fonksiyon üretici bloğuna gönderilirken 0xFFFF değerinden çıkartılarak gönderilmiştir. Bu şekilde ilk çeyrekteki verinin yatay düzlem eksenine göre simetriği alınarak üçüncü çeyrek düzlem verisi elde edilmiştir.

Dördüncü çeyrek sinüs verisi oluşturulurken, ikinci çeyrek verinin elde edilmesi için kullanılan okuma yöntemi kullanılmış ve ikinci çeyrek düzlem verileri aynı şekilde elde edilmiştir. Bu kez elde edilen veriler 0xFFFF değerinden çıkartılarak fonksiyon üretici bloğuna gönderilmiştir. Bu yöntem ile ikinci çeyrek sinüs verisinin yatay düzlem eksenine göre simetriği alınarak dördüncü çeyrek düzlem verisi elde edilmiştir.

0-90° aralığındaki veriden tüm skala verisini oluşturmak için yapılan bu işlemler Şekil 22 ile gösterilmiştir.



Şekil 22. Sinüs Verisinin Simetri Özelliğiyle Elde Edilmesi [22]

Uygulanan bu algoritma sayesinde sinüs verilerinin kaydedilmesi için yalnızca 4 kbit büyüklüğünde BRAM kullanılmıştır ve hafıza boyutundan %75 oranında tasarruf yapılmıştır.

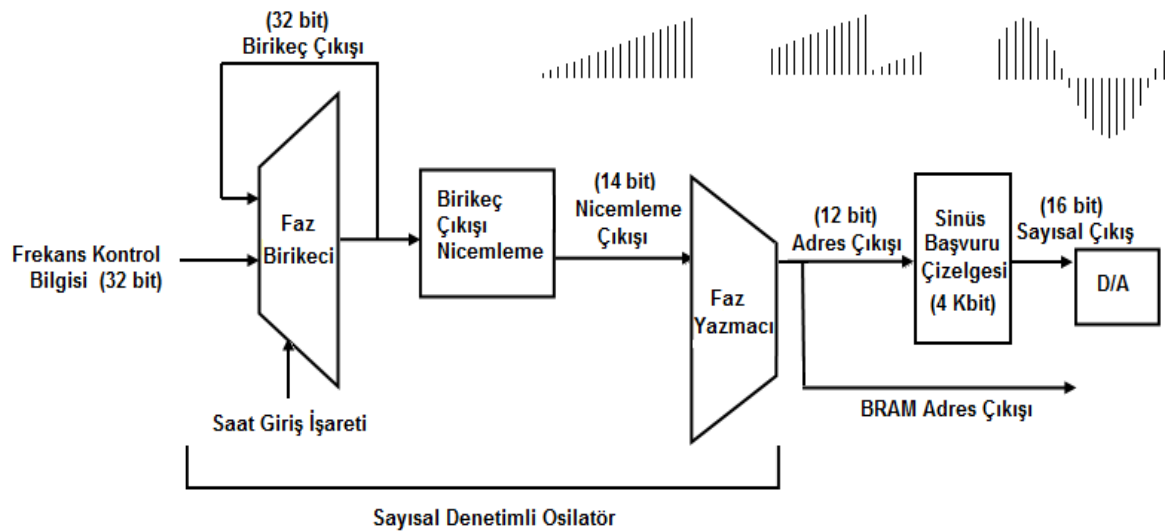
BRAM'ın yalnızca 256 hafıza hücresi kullanılmıştır. 0-255 aralığındaki hücreler toplam 4096 bit veri saklamaktadır. Tüm veri ($2^{12} = 4096$) 12 bit ile ifade edilebildiğinden, BRAM'ın 14 bit olan adres yolunun sadece 12 biti kullanılmıştır.

4. DDS FONKSİYON ÜRETECİ TASARIMI

DDS algoritması bu çalışmanın temelini oluşturmaktadır. Bu bölümde önceki bölümlerden elde edilen parametreler kullanılarak DDS algoritmasının tasarlanması üzerinde durulmuştur.

4.1 DDS Algoritmasının Gerçeklenmesi

DDS genel olarak sayısal denetimli osilatör (NCO), sinüs başvuru çizelgesi ve sayısaldan analoga çevirici bloklarından oluşur [11]. Temel DDS veri akış diyagramı Şekil 23 ile gösterilmiştir. Sinüs başvuru çizelgesi oluşturulması ve sayısaldan analoga çevirici tasarımları önceki bölümlerde anlatılmıştır. Bu bölümde sayısal denetimli osilatör anlatılacaktır.



Şekil 23. Temel DDS Veri Akış Diyagramı

Bu akışta görülen sayısal denetimli osilatör DDS algoritmasının tabanını oluşturur.

Sayısal denetimli osilatör genel olarak faz birikeci ve faz yazmacından oluşur.

Frekans kontrol bilgisi 32 bittir, frekans artışının değerini verir ve frekans ayarlaması bu veri ile yapılır. Frekans kontrol bilgisi ile birikeç çıkış verisi aynı büyüklüktedir. Frekans kontrol bilgisi ile kullanılan saat işaretinin frekansı DDS'nin frekans çözünürlüğünü belirler.

Her saat çevriminde faz birikeci frekans kontrol bilgisini alır, faz birikeci çıkışında bir önceki saat çevriminde oluşan veri ile toplar ve faz yazmacına gönderir. Bu arada faz birikecinin yeni çıkış değeri bir sonraki saat çevriminde kullanılmak üzere faz birikecine geri beslenir. Bu şekilde faz birikeci frekans kontrol verisini her yeni saat çevriminde toplamaya devam eder. Faz birikeci bu toplama işlemine, toplanan değer 32 bit boyutunu aşana kadar devam eder, bu taşma yaşandıktan sonra birikeç taşan biti ihmal ederek toplama işlemine devam eder ve periyodik bir döngü yaratır. Faz birikecinin taşma süresi DDS'nin çıkış frekansını belirler [4]. Yüksek frekanslarda dalgalar elde edilmek istendiğinde frekans kontrol bilgisi büyük seçilir, böylece toplama adımları büyüyeceğinden taşma noktasına daha çabuk ulaşılır ve yüksek bir çıkış frekansı elde edilmiş olur. Aynı şekilde düşük frekanslar elde edilmesi için frekans kontrol bilgisi değeri küçük seçilmelidir ki taşma noktasına daha uzun sürede ulaşılsın.

Faz birikecinden çıkan verinin bütün bitleri başvuru çizelgesini adreslemede kullanıldığı durumda çok büyük hafıza boyutları gerekeceğinden faz birikeci çıkışı nicemlenmektedir. Nicemleme derecesi başvuru çizelgesinin boyutlarına dolayısıyla da adres yolunun genişliğine bağlıdır. Kullanılan BRAM'in adres yolu 14 bit olduğundan, 32 bit olan birikeç çıkış verisinin soldan (en önemli) 14 biti alınarak nicemlenmiştir. Faz yazmacına yazılan bu 14 bitlik verinin sağdan (en önemsiz) 12 bitlik bölümü başvuru çizelgesi için adres yolu olarak kullanılmıştır. Başvuru çizelgesi adres yolu normalde 14 bittir fakat başvuru çizelgesi boyutları optimize edilerek 4 kbit değerine indirildiğinden 12 bit adres yolu yeterli olmaktadır. Başvuru çizelgesinin bu 12 bit veri ile adreslenen hafıza hücresinde bulunan 16 bit örnekleme verisi analog forma dönüştürülmek üzere D/A çevirici bloğuna gönderilir.

Dalga formunun frekansı referans saat işaretine, frekans kontrol bilgisi değerine ve faz birikecinin genişliğine bağlıdır. Dalga'nın frekansı aşağıdaki denklem ile hesaplanır;

$$f_{\text{çıkış}} = \frac{F.K.B \times f_{\text{referans}}}{2^N} \quad (11)$$

$f_{\text{çıkış}}$ = DDS çıkış frekansı

$F.K.B$ = Frekans kontrol bilgisi verisi

f_{referans} = Referans saat girişi frekansı

N = Faz birikecinde kullanılan bit sayısı [13]

Bu tasarımda f_{referans} değeri belirlenirken (4) numaralı denklem ile belirlenen 579 KHz değerine eşit veya bu frekanstan bir miktar daha düşük bir frekans seçilmek amaçlanmıştır. Kullanılan FPGA saat işareti 33 MHz olduğundan, bu saatin bir çevrimi 30,3 ns sürmektedir. Bu saat kullanılarak gereklere uygun bir referans saati frekansı yaratılmıştır. 33 MHz asıl saat işaretinin 29 çevrimi boyunca düşük seviyede tutulan, bir sonraki 29 çevrimi boyunca yüksek değer alan bir referans saati oluşturulmuştur. Bu referans saatinin bir çevrimi 33 MHz saat girişinin 29 + 29 = 58 çevrimine denk gelmektedir ve referans saat işaretinin frekansı;

$$\frac{33 \text{ MHz}}{58 \text{ çevrim}} = 568966 \text{ Hz} \quad (12)$$

olarak hesaplanmıştır. Çalışmalar boyunca DDS referans saat frekansı olan $f_{\text{referans}} = 568,966 \text{ KHz}$ olarak kullanılmıştır.

Ayrıca faz birikeci genişliği 32 bit olduğundan, $N = 32$ olarak alınmıştır. Bu durumda DDS çıkış frekansı denklemi;

$$f_{\text{çıkış}} = \frac{F.K.B \times 568,966 \text{ KHz}}{2^{32}} \quad (13)$$

şeklinde indirgenmiştir.

Örneğin 1 KHz frekans değerinde dalgalar elde edilmek isteniyorsa, frekans kontrol kelimesinin;

$$F.K.B = \frac{1 \text{ KHz} \times 2^{32}}{568,966 \text{ KHz}} = 7548724 \quad (14)$$

değerinde olması gerekmektedir.

DDS'nin frekans çözünürlüğü, frekans kontrol bilgisi $F.K.B = 1$ alınarak hesaplanır;

$$\Delta F = \frac{f_{referans}}{2^N} \quad (15)$$

ΔF = Frekans çözünürlüğü

$f_{referans} = 568,966 \text{ KHz}$ ve $N = 32$ olduğundan frekans çözünürlüğü;

$$\Delta F = \frac{568,966 \text{ KHz}}{2^{32}} = 0,0001325 \text{ Hz} \quad (16)$$

olarak hesaplanmıştır.

Frekans çözünürlüğü değeri aynı zamanda DDS'nin minimum frekans değerine eşittir. Maksimum DDS frekansı ise Nyquist Teoremi ile belirlenir. Nyquist Teoremi'ne göre hatasız bir şekilde üretilebilecek maksimum frekans, örnekleme hızının yarısından az olmalıdır. Buna göre DDS modülünde üretilebilecek maksimum frekans $f_{çıkış-maks} = f_{referans} / 2$ değerindedir. Fakat sürekliliğin sağlanması ve örneklenen veriden yeterince saf bir dalgaın elde edilebilmesi için genellikle $f_{çıkış-maks} = \%30 f_{referans}$ şeklinde kullanılır [23].

Bu tasarımda maksimum frekans;

$$f_{çıkış-maks} = \frac{30}{100} \times 568966 = 170689,8 \text{ Hz} \quad (17)$$

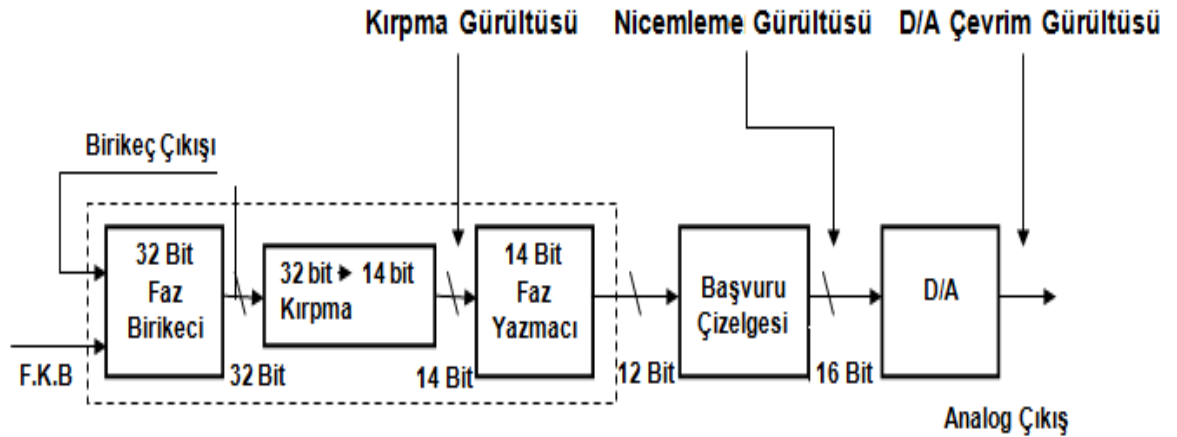
olarak kullanılmıştır.

Ayrıca başvuru çizelgesinin boyutları dolayısıyla adres yolu genişliği de dalganın saflığını etkileyen iki önemli parametredir. Bu parametreler artırıldığında çıkış dalgasının çözünürlüğü de artar.

4.2 Spektral Saflık Kavramları

DDS çıkış sinyalinde üç ana gürültü bulunur ve bunlar şu şekilde sıralanır:

- Faz kırpma gürültüsü
- Nicemleme gürültüsü
- Sayısalan analoga çevrim gürültüsü [24]. Bu gürültülerin hangi blok çıkışlarında gözlemlendiği Şekil 24 ile gösterilmiştir.



Şekil 24. DDS Akış Diyagramında Gürültülerin Konumları

Kırpma gürültüsü, faz birikeci çıkışında oluşan verinin en önemsiz bitlerinin kırılmasıyla meydana gelir. Faz birikecinde gerekli frekans çözünürlüğünün sağlanması için çok sayıda bit bulunur. Fakat faz birikecinde kullanılan tüm bitlerin başvuru çizelgesi için kullanılması başvuru çizelgesinin boyutlarını çok büyütecektir. Başvuru çizelgesinin boyutlarını yönetilebilir seviyelerde tutmak için

faz birikeci çıkışının en önemsiz bitlerinin bir kısmı kırpılır (nicemlenir). Kaç bitin kırılacağı, başvuru çizelgesinin boyutlarına göre belirlenir. Bu çalışmada başvuru çizelgesi $2^{12} = 4096$ bit, dolayısıyla da adres yolu 12 bit büyüklüğünde olduğundan, 32 bitlik faz birikeci çıkışının toplam 20 biti kırılmaktadır. Faz birikeci çıkışında bulunan veri 32 bit olarak kırılmadan kullanıldığı durumda $2^{32} = 4$ Gigabit boyutunda bir hafıza modülüne ihtiyaç duyulacaktır.

Nicemleme gürültüsü dalga formlarının belirli bir frekansla örneklenebilmesinden kaynaklanır. Dalganın tüm değerleri başvuru çizelgesine yüklenemeyeceğinden belli bir örnekleme frekansı seçilir ve dalga formu örnek noktalarla yuvarlanarak başvuru çizelgesine kaydedilir. Örnekleme frekansı ne kadar hızlı, dolayısıyla başvuru çizelgesi boyutu ne kadar büyük olursa bu gürültü o kadar küçük olacaktır.

Bu çalışmada sinüs dalgasının bir tam periyodu $256 \times 4 = 1024$ örneklenmiş veri ile ifade edilmiştir. Başvuru çizelgesi girişinde kullanılan adres yolu 12 bit olduğundan bu noktadaki çıkış frekansı denklemi;

$$f_{\text{çıkış}} = \frac{F.K.B \times 568,966 \text{ KHz}}{2^{12}} \quad (18)$$

şeklinde olur.

Örnekleme frekansının hesaplanması için $F.K.B$ değeri 16 olarak alınır, bu sayının seçilmesinin nedeni, kullanılan başvuru çizelgesinde her bir hücrenin adres aralığının 16 olmasıdır. Bu şekilde her hücredeki örnek veri bir kez okunarak dalga formuna işlenecektir. Çıkış frekansı ve periyodu bu parametre ile hesaplandığında;

$$f_{\text{çıkış}} = \frac{16 \times 568,966 \text{ KHz}}{2^{12}} = 2222,52 \text{ Hz} \quad (19)$$

$$T_{\text{çıkış}} = \frac{1}{2222,52} = 0,00045 \text{ s} \quad (20)$$

olarak belirlenir. 0.00045 saniyelik periyot boyunca 1024 örnekleme yapıldığından örnekleme frekansı;

$$\frac{1024 \text{ örnek}}{0,00045 \text{ s}} = 2275555,6 \text{ örnek/s} \quad (21)$$

olarak hesaplanmıştır.

D/A çevrim gürültüsü ise, sayısal dalganın analoga çevrilmesi sırasında oluşan gürültüdür. Bu gürültü dalganın ideal çıkış seviyesi ile gerçek D/A çevirici çıkış seviyesi arasındaki farka eşittir. D/A çevirici bloğunun anahtarlama özellikleri ne kadar iyi olursa D/A çevrim gürültüsü o kadar az olur [25][26][27]. Çıkış dalgasında bulunan DDS gürültülerinin en baskını D/A çevrim gürültüsüdür. Bu gürültü D/A çevirici bloğunun çözünürlüğü ile doğru orantılıdır ve aşağıdaki denklem ile ifade edilir.

$$SNR = 6,02N + 1,76 \text{ dB} \quad (22)$$

Bu denklemdeki N değeri D/A çevrim bloğunun bit sayısıdır [28].

Bu tasarımda kullanılan D/A çevirici entegre devresinin veriyolu 16 bit olduğundan D/A çevrimi sinyal gürültü oranı (SNR);

$$SNR = (6,02 \times 16) + 1,76 \text{ dB} = 98,08 \text{ dB} \quad (23)$$

olarak hesaplanır.

Faz birikeci çıkışında oluşan kırpma gürültüsü, çıkış dalgasında zaman tabanlı seçirmeye yol açar. Seçirme bir olayın gerçekleşmesi gereken ideal zamanla,

gerçekleştirdiği zaman arasındaki farktır ve hatalı faz bilgisinden kaynaklanır. Kırpma gürültüsü ve neden olduğu gürültülerle ilgili ayrıntılı bilgiye Xilinx'in DDS Compiler V4.0 veri sayfalarından ulaşılabilir [12]. Bu veri sayfalarında farklı büyüklükteki başvuru çizelgeleri için dalga çıkışında bulunan parazit oranları belirtilmiştir.

- 256 bit başvuru çizelgesi => Yaklaşık SNR = 48 dB
- 512 bit başvuru çizelgesi => Yaklaşık SNR = 54 dB
- 1024 bit başvuru çizelgesi => Yaklaşık SNR = 60 dB
- 2048 bit başvuru çizelgesi => Yaklaşık SNR = 66 dB [11]

S. Cheng ve J.R. Jensen [29]'in çalışmalarından DDS algoritmalarında spektral saflık konusunda detaylı bilgiye ulaşılabilir.

4.3 Sinüs ve Kosinüs Dalga Formlarının Üretilmesi

Sinüs dalgasının üretilmesi için daha önce anlatılan başvuru çizelgesi ve DDS algoritması kullanılmıştır. Şekil 23'de gösterilen faz birikecinden çıkıp, 14 bite indirgenen faz bilgisi verisinin en önemsiz 12 biti başvuru çizelgesini adreslemede kullanılırken, en önemli 2 biti sinüs dalgasının hangi çeyrek düzleminde olunduğunun bilgisini verir.

Bu çeyrek düzlem bilgisi kullanılarak sinüs üretilirken ilk çeyrek düzlem için başvuru çizelgesindeki veriler aynı sırayla kullanılır. İkinci çeyrek düzlem için çizelge verileri tersten okunarak kullanılır, üçüncü çeyrek düzlem için ilk çeyrek düzlem verisi 0xFFFF değerinden çıkarılarak yatay eksene göre simetriği alınır, dördüncü ve son çeyrek düzlem içinse ikinci çeyrek düzlem verisi 0xFFFF değerinden çıkarılarak yatay eksene göre simetriği alınır.

Aynı çeyrek düzlem bilgisi kullanılarak kosinüs üretilirken ilk çeyrek düzlem için başvuru çizelgesi verileri tersten okunarak kullanılır. İkinci çeyrek düzlem için başvuru çizelgesindeki sırayla okunan veriler 0xFFFF değerinden çıkarılarak yatay eksene göre simetriği alınır, üçüncü çeyrek düzlem için ilk çeyrek düzlem verisi 0xFFFF değerinden çıkarılarak yatay eksene göre simetriği alınır, dördüncü ve son çeyrek düzlem içinse başvuru çizelgesindeki veriler aynı sırayla kullanılır.

Bu yapı için kullanılan algoritma örneği aşağıda verilmiştir.

Kırpılarak 14 bite indirgenmiş frekans birikeci verisi "FAZ" ile, 12 bit başvuru çizelgesi adres bilgisi "ADRES" ile, 16 bitlik başvuru çizelgesi çıkışı verisi "ÇİZELGE" olarak, başvuru çizelgesinin ters sırayla okunan çıkış verisi "TERS_ÇİZELGE" olarak, sinüs dalga formu "SİNÜS" olarak, kosinüs dalga formu "KOSİNÜS" olarak ifade edilmiştir.

ADRES <= FAZ (11 downto 0);

if FAZ >= "00000000000000" and FAZ <= "01000000000000" then # 0-90°

SİNÜS <= ÇİZELGE;

KOSİNÜS <= TERS_ÇİZELGE;

elsif FAZ >= "01000000000000" and FAZ <= "10000000000000" then # 90-180°

SİNÜS <= TERS_ÇİZELGE;

KOSİNÜS <= "11111111111111" - ÇİZELGE;

elsif FAZ >= "10000000000000" and FAZ <= "11000000000000" then # 180-270°

SİNÜS <= "11111111111111" - ÇİZELGE;

KOSİNÜS <= "11111111111111" - TERS_ÇİZELGE;

elsif FAZ >= "11000000000000" then # 270-360°

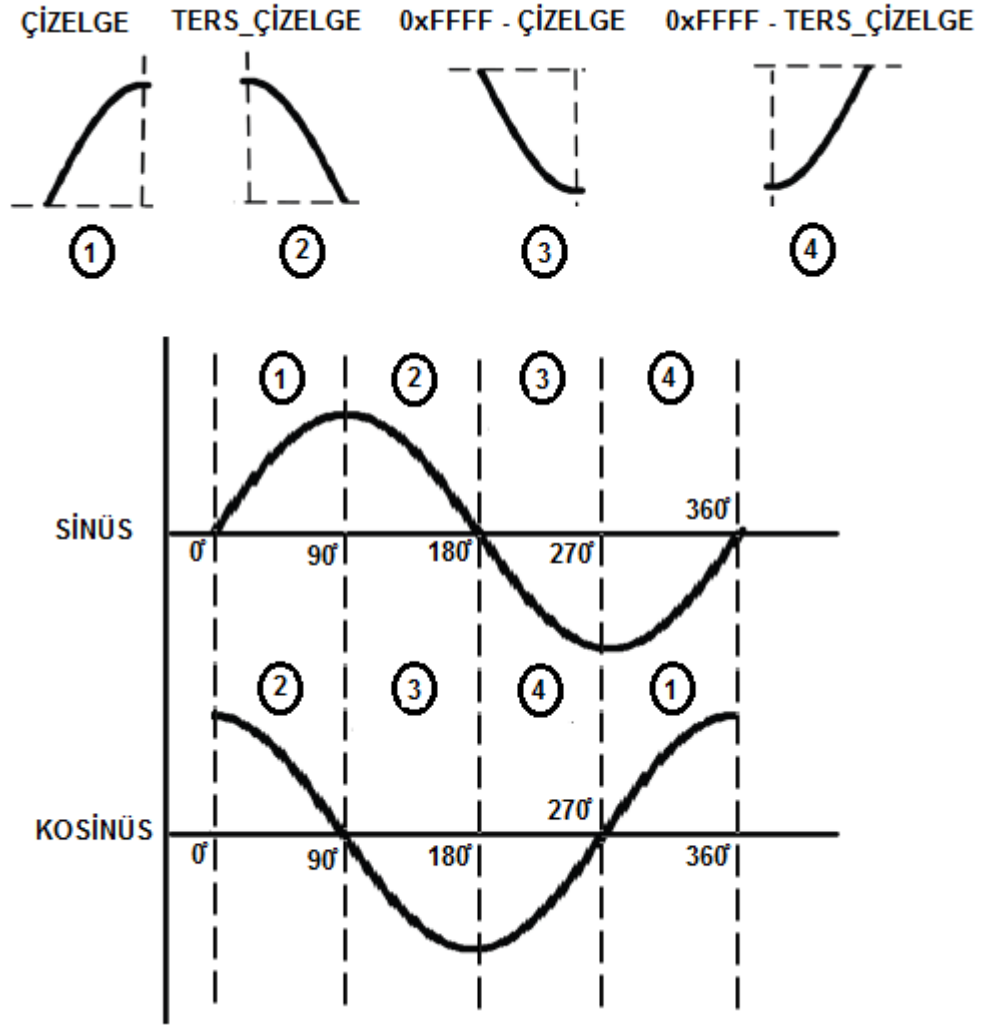
SİNÜS <= "11111111111111" - TERS_ÇİZELGE;

KOSİNÜS <= ÇİZELGE;

end if;

Bu algoritmanın ilk satırında 14 bite kırpılmış faz birikeci bilgisinin, en önemsiz 12 biti adres verisi olarak atanmıştır. Algoritmada bulunan dört koşul dört çeyrek düzlemi ifade etmektedir ve faz birikeci bilgisinin ilk iki biti ile çeyrek düzlem seçimi yapılmıştır.

Bu algoritma kullanılarak, sinüs verisinin ilk çeyrek düzlemi ile sinüs ve kosinüs dalga formlarının bir tam periyotlarının oluşturulduğu Şekil 25 ile gösterilmiştir;



Şekil 25. Sinüs ve Kosinüs Dalgalarının Simetri Özelliği ile Oluşturulması

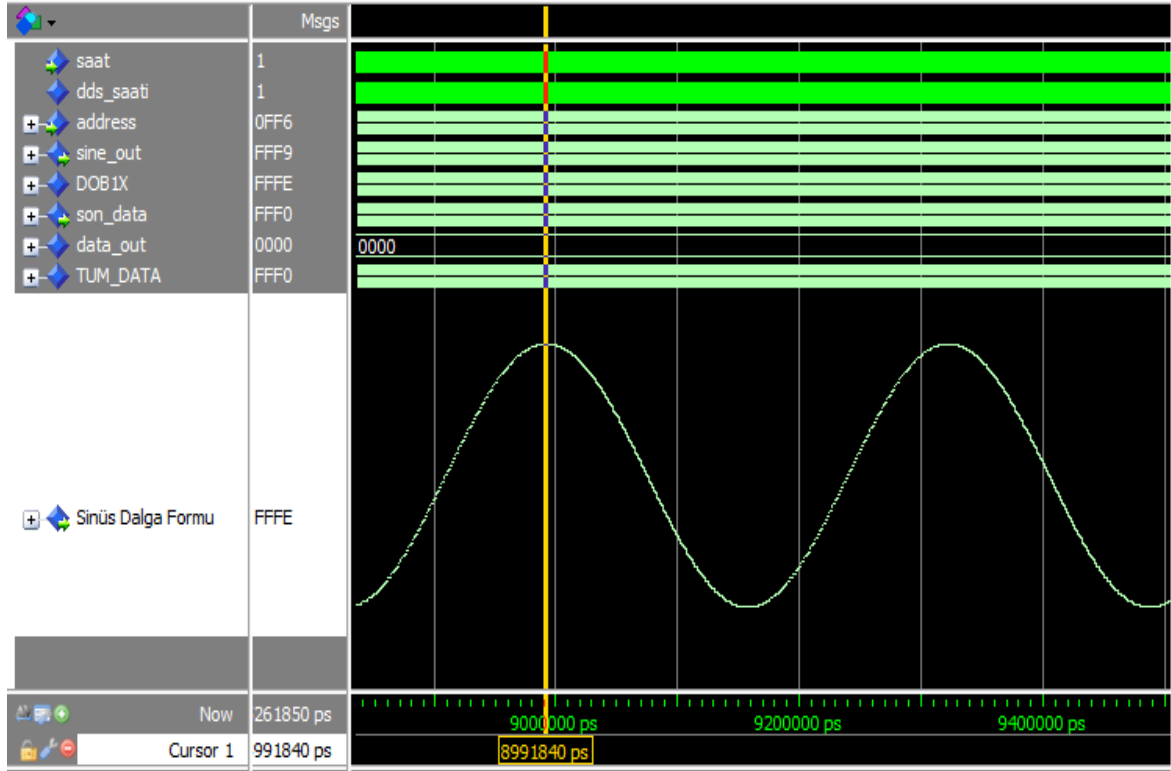
Bu algoritmayla hazırlanan kod ile sinüs ve kosinüs dalgalarının üretilmesi için dalgaların frekansı 1 KHz olarak belirlenmiştir. Bu durumda (13) numaralı denklemde görüldüğü gibi:

$$F.K.B = \frac{1 \text{ KHz} \times 2^{32}}{568,966 \text{ KHz}} = 7548724 \quad (24)$$

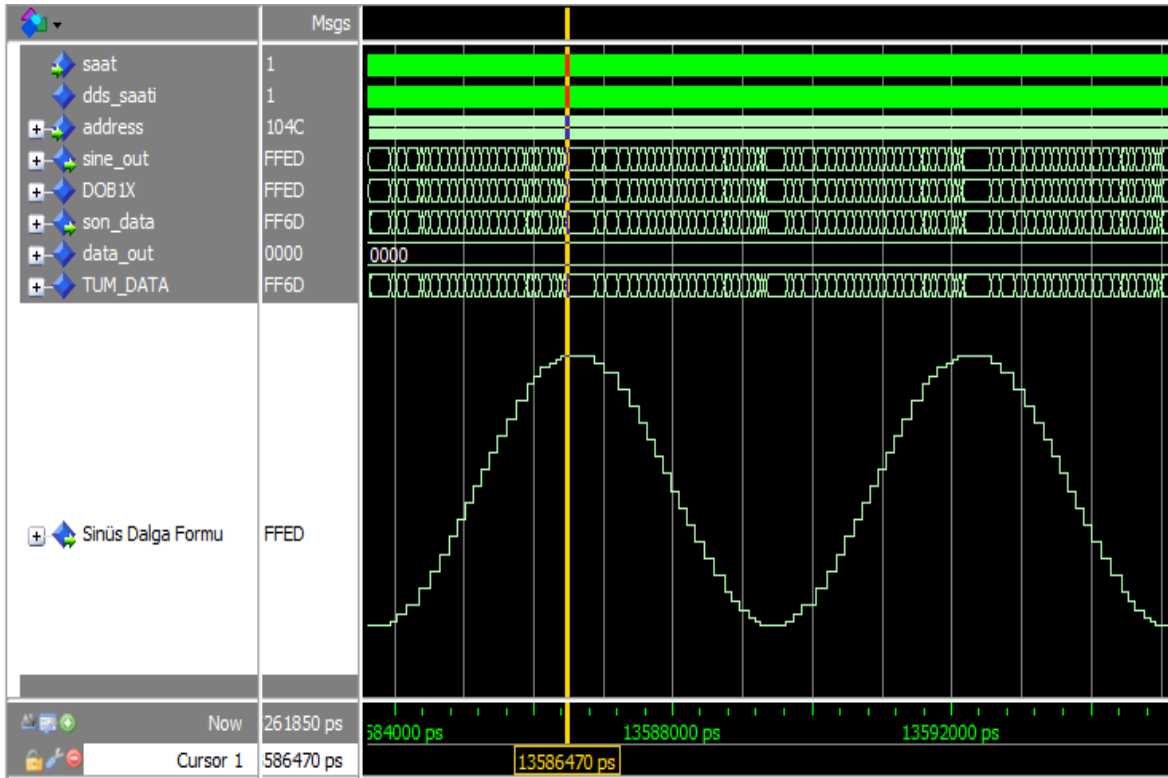
olarak hesaplanmıştır. 7548724 değeri ikili aritmetiğe dönüştürüldüğünde 32 bit F.K.B verisi "00000000011100110010111100110100" olarak elde edilmiş ve frekans kontrol bilgisi dizinine atanmıştır.

Algoritmanın benzetiminin yapılması ve üretilen dalgaların simülasyon ile gözlemlenebilmesi için Xilinx firmasının kod düzenleyicisiyle birlikte çalışan ChipScope Pro isimli program yeterli olmamıştır. Bu programda sinyallerin analog gösterim özelliği olmadığından, analog formatta gösterim yapabilen bir simülasyon programı araştırılmıştır. ModelSim isimli simülasyon programı analog formda görüntüleme yapabildiği için bu program seçilmiştir [30].

Sinüs ve kosinüs dalgalarını gözlemek için simülasyon yapılırken asıl saat işareti olarak 10 birim periyotluk bir saat kullanılırken, (12) numaralı denkleme uygun olarak DDS referans saat girişi asıl saatin 58 katı periyodunda kullanılmıştır. Bu saat periyotları ile yapılan simülasyon Şekil 26 ile gösterilmiştir. Bu şekilde elde edilen simülasyonda dalga formunun oluşması için uzun süre geçtiğinden ve sinyal geçişleri tam gözlemlenemediğinden asıl saat işareti 2 birim periyotla, DDS referans saati 10 birim periyotla çalışan bir başka simülasyon yapılarak Şekil 27 ile gösterilmiştir. Bu simülasyonda sinyal geçişleri ve basamaklı dalga formu daha net görülmektedir.



Şekil 26. Sinüs/Kosinüs Dalga Üretim Simülasyonu (Referans Saati = 580 Birim Periyot)



Şekil 27. Sinüs/Kosinüs Dalga Üretim Simülasyonu (Referans Saati = 10 Birim Periyot)

Bu simülasyonlar ile DDS algoritmasının sinüs ve kosinüs dalgaları için hatasız çalıştığı doğrulanmıştır.

4.4 Üçgen Dalga Formunun Üretilmesi

Üçgen dalga üretimi için başvuru çizelgesine ihtiyaç duyulmamıştır. Bu dalga formunun üretiminde DDS algoritmasında kullanılan ve frekans kontrol bilgisinin katları şeklinde doğrusal olarak artan faz birikeci çıkış verisi kullanılmıştır.

Üçgen dalga üretiminde kullanılan algoritma genel olarak aşağıdaki gibidir.

32 bit olan faz birikeci çıkış verisinin sağdan (en önemli) 16 biti alınmış ve "ÜÇGEN" dizinine atanmıştır. Sinüs algoritmasında da kullanılan, faz birikeci çıkışından kırılarak 14 bite indirgenmiş frekans birikeci verisi "FAZ" ile ifade edilmiştir. Oluşturulan 16 bit büyüklüğündeki üçgen dalga verisi ise "ÜÇGEN_ÇIKIŞ" ile ifade edilmiştir. Sinüs algoritmasında da görüldüğü üzere,

"FAZ" verisi "10000000000000" değerinden küçükse, bu veri 0-180° arası sinüs değerlerini barındıran hücreleri adresler , eğer "10000000000000" değerinden büyükse 180-360° arası sinüs değerlerini barındıran hücreleri adresler. Sinüs verileri 0-180° aralığında artı , 180-270° aralığında eksi değerlerde olduğundan, "FAZ" verisi "10000000000000" değerinden küçükse "ÜÇGEN" verisi aynen kullanılmış, bu değerden büyükse "ÜÇGEN" verisi "1111111111111111" değerinden çıkarılarak kullanılmıştır.

```

if FAZ < "10000000000000"      then                                # 0-180°

    ÜÇGEN_ÇIKIŞ <= ÜÇGEN;

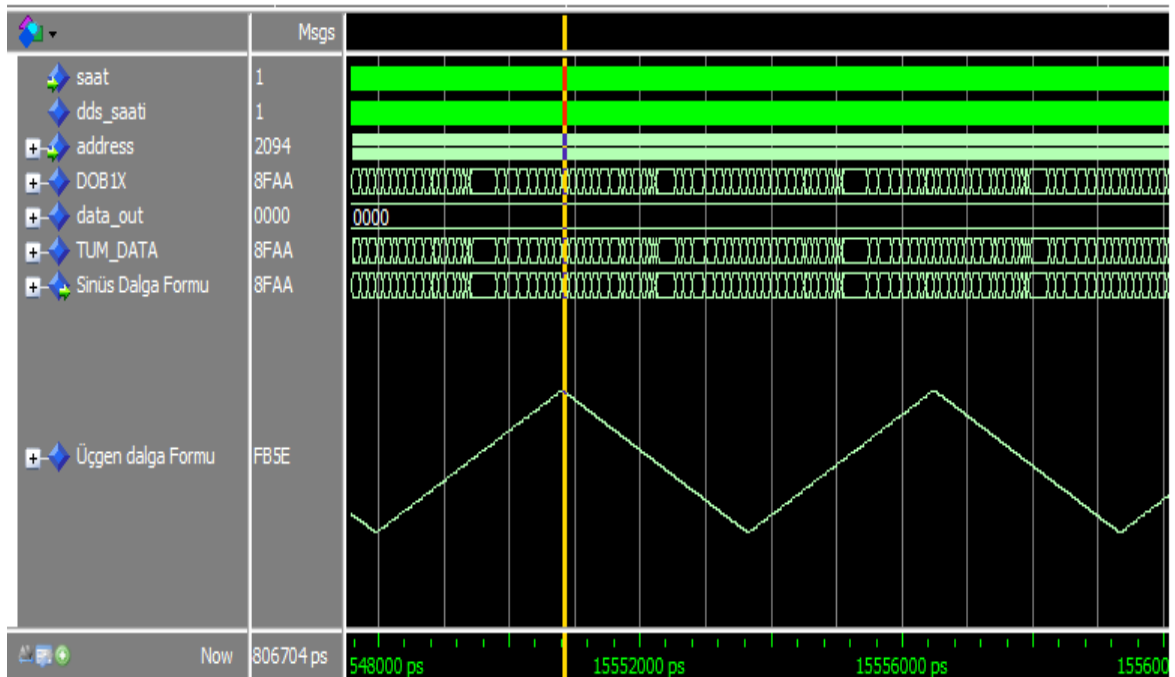
elsif FAZ >= "10000000000000"  then                                # 180-360°

    ÜÇGEN_ÇIKIŞ <= "1111111111111111" - ÜÇGEN;

end if;

```

Bu algoritma kullanılarak VHDL kodu hazırlanmış ve ModelSim programı kullanılarak asıl saat işareti 2 birim periyotla, DDS referans saati 10 birim periyotla çalışan bir simülasyon oluşturulmuş ve Şekil 28 ile gösterilmiştir.



Şekil 28. Üçgen Dalga Üretim Simülasyonu (Referans Saati = 10 Birim Periyot)

4.5 Testere Dalga Formunun Üretilmesi

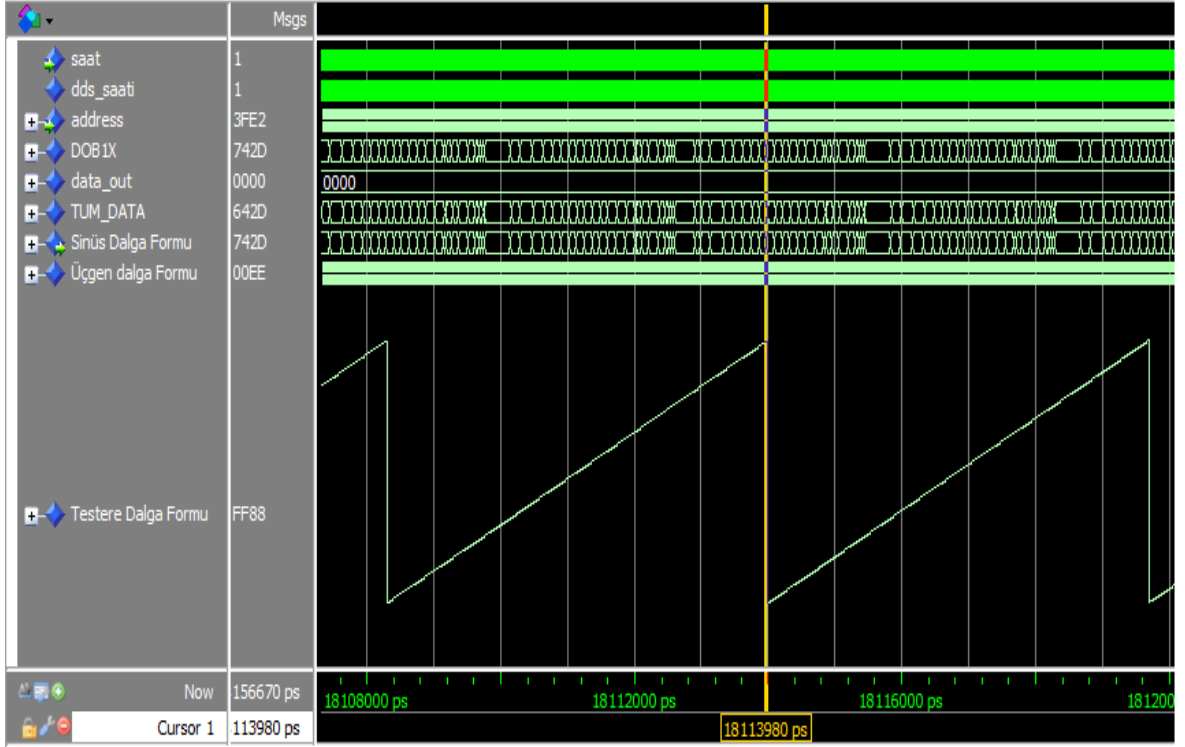
Testere dalgaının üretilmesi için de başvuru çizelgesine ihtiyaç duyulmamıştır. Bu dalga formunun üretiminde üçgen dalga üretiminde de kullanılan ve frekans kontrol bilgisinin katları şeklinde doğrusal olarak artan faz birikeci çıkış verisi kullanılmıştır.

Testere dalga üretiminde kullanılan algoritma genel olarak aşağıdaki gibidir.

32 bit olan faz birikeci çıkış verisinin sağdan (en önemli) 16 biti alınmış ve "TESTERE" dizinine atanmıştır. Testere dalga, periyodu boyunca doğrusal olarak artıp, maksimum genliğine ulaştıktan sonra sıfır değerine düştüğünden, "TESTERE" verisi periyot boyunca aynı şekilde kullanılmıştır. Oluşturulan 16 bit büyüklüğündeki testere dalga verisi ise "TESTERE_ÇIKIŞ" ile ifade edilmiştir.

```
if clk'event and clk = '1' then          #saat işaretinin her yükselen kenarında
    TESTERE_ÇIKIŞ <= TESTERE;
end if;
```

Bu algoritma kullanılarak VHDL kodu hazırlanmış ve ModelSim programı kullanılarak asıl saat işareti 2 birim periyotla, DDS referans saati 10 birim periyotla çalışan bir simülasyon oluşturulmuş ve Şekil 29 ile gösterilmiştir.



Şekil 29. Testere Dalga Üretim Simülasyonu (Referans Saati = 10 Birim Periyot).

4.6 Kare Dalga Formunun Üretilmesi

Kare dalga formunun üretiminde, dalga verisine ihtiyaç duyulmamaktadır fakat faz birikeci bilgisiyle senkronizasyonunun sağlanması için faz birikeci çıkış verisi kullanılmıştır.

Kare dalga üretiminde kullanılan algoritma genel olarak aşağıdaki gibidir.

Sinüs ve üçgen dalga algoritmalarında da kullanılan, faz birikeci çıkışından kırılarak 14 bite indirgenmiş frekans birikeci verisi "FAZ" ile ifade edilmiştir. Oluşturulan 16 bit büyüklüğündeki kare dalga verisi ise "KARE_ÇIKIŞ" ile ifade edilmiştir. Sinüs algoritmasında da görüldüğü üzere, "FAZ" verisi "10000000000000" değerinden küçükse, bu veri 0-180° arası sinüs değerlerini barındıran hücreleri adresler, eğer "10000000000000" değerinden büyükse 180-360° arası sinüs değerlerini barındıran hücreleri adresler. Sinüs verileri 0-180° aralığında artı, 180-270° aralığında eksi değerlerde olduğundan, "FAZ" verisi "10000000000000" değerinden küçükse "KARE" dizinine "11111111111111", bu değerden büyükse "KARE" dizinine "00000000000000" verileri atanmıştır.

if FAZ < "10000000000000" then

0-180°

KARE_ÇIKIŞ <= "1111111111111111";

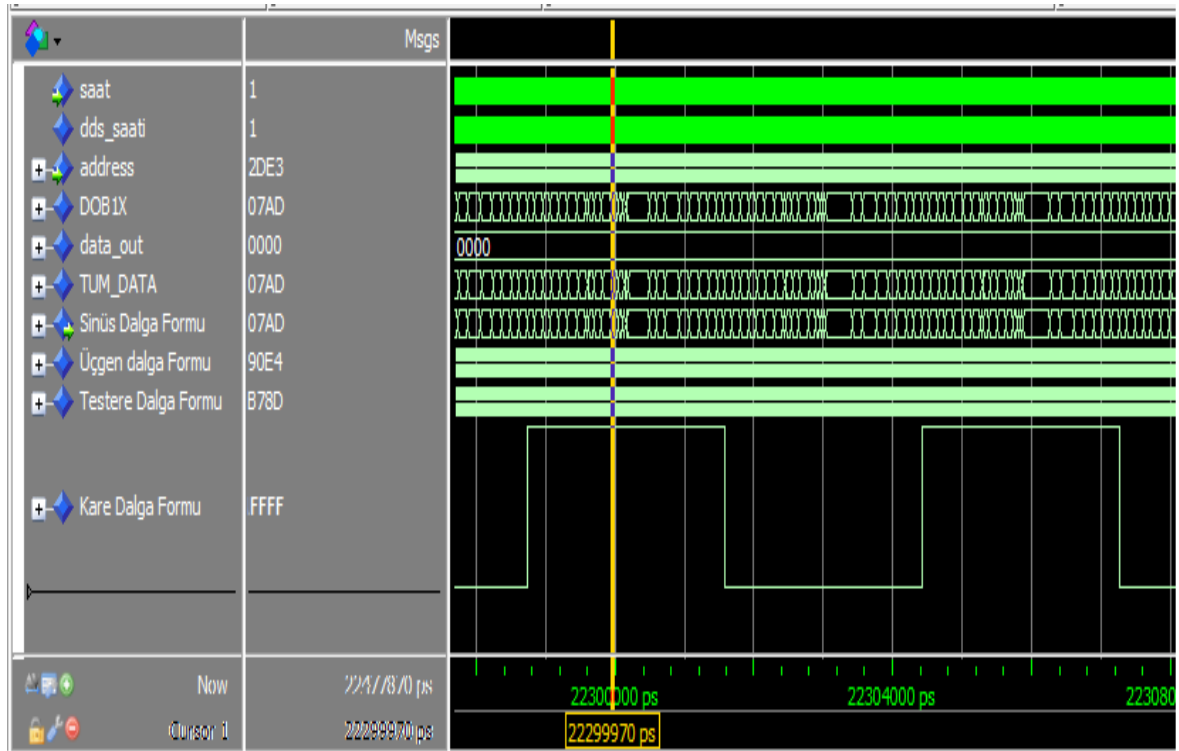
elsif FAZ >= "10000000000000" then

180-360°

KARE_ÇIKIŞ <= "0000000000000000";

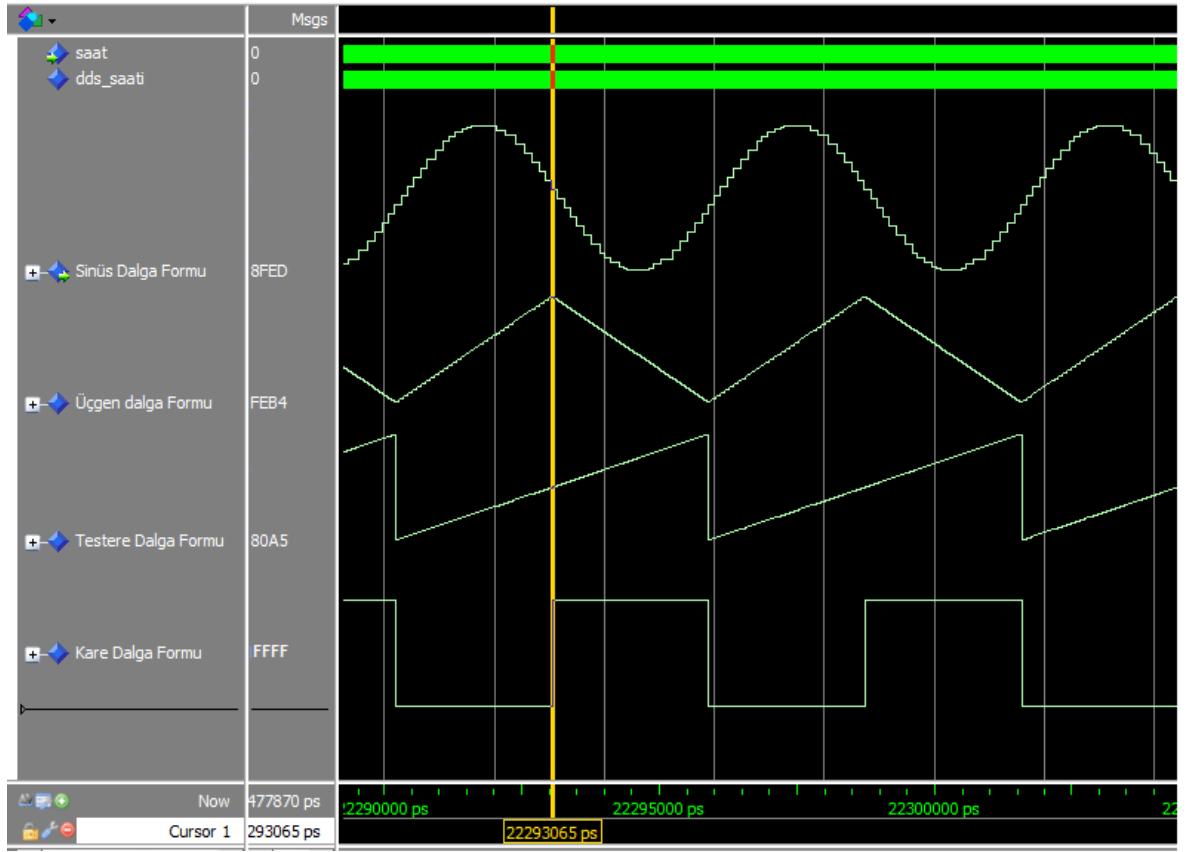
end if;

Bu algoritma kullanılarak VHDL kodu hazırlanmış ve ModelSim programı kullanılarak asıl saat işareti 2 birim periyotla, DDS referans saati 10 birim periyotla çalışan bir simülasyon oluşturulmuş ve Şekil 30 ile gösterilmiştir.



Şekil 30. Kare Dalga Üretim Simülasyonu (Referans Saati = 10 Birim Periyot)

Ayrıca üretilen tüm dalgaların aynı algoritma altında, aynı saat işareti (asıl saat işareti 2 birim periyotla, DDS referans saati 10 birim periyot) ile yapılmış simülasyonları Şekil 31 ile gösterilmiştir.



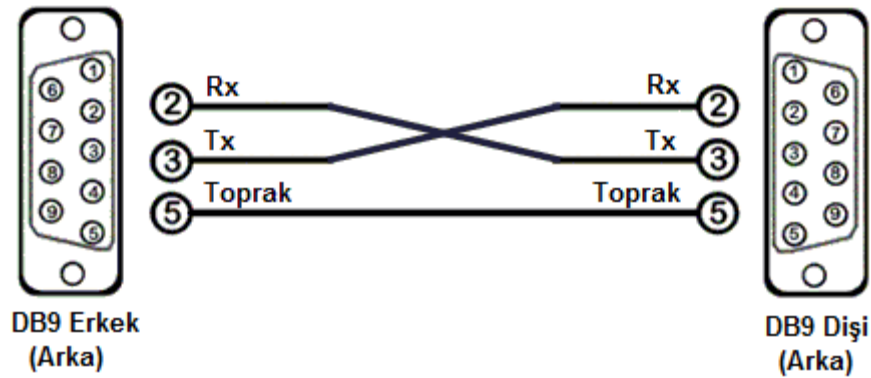
Şekil 31. Tüm Dalga Formalarının Ortak Simülasyonu

5. SERİ KANAL ARAYÜZÜ İLE GENLİK VE FREKANS AYARLANMASI

DDS algoritması ile üretilen dalgaların genlik ve frekans parametrelerinin, tekrar sentezleme gerektirmeden, gerçek zamanlı olarak değiştirilebilmesi için bir seri kanal arayüzü tasarlanmıştır. Bu bölümde öncelikle kullanılan seri kanalın özelliklerinden bahsedilmiş, ardından genlik ve frekans parametrelerinin güncellenme algoritması ve hazırlanan seri kanal arayüzü anlatılmıştır.

5.1 Seri Kanal Özellikleri

Tasarımda kullanılan ML 507 geliştirme kartının üzerinde bir adet erkek RS-232 seri kanal konektörü bulunmaktadır. Bu seri kanal portu ana eleman olduğundan, bu portu bilgisayara bağlamak için seri kanal kukla modem kablosu kullanılmalıdır [31]. Bu kablounun bağlantıları Şekil 32'deki gibi olmalıdır. Bu bağlantılardan "Rx" alıcı hattını, "Tx" verici hattını, "Toprak" ise referans toprak bağlantısını ifade eder. Bilgisayar ile geliştirme kartı arasında veri transferinin yapılabilmesi için alıcı ve verici hatlarının çaprazlanması gerekmektedir.



Şekil 32. Seri Kanal Modem Kablosu Bağlantıları

Geliştirme kartında seri kanal entegre devresi olarak Analog Devices firmasına ait ADM3202 RS-232 alıcı-verici entegre devresi kullanılmıştır. Bu entegre devre 460

kbps veri iletişim hızına kadar çıkabilen iki kanallı bir seri kanal bağlantısı sağlar [32]. Bu iki kanaldan biri ML 507 geliştirme kartında DB9 tipi erkek konektörle kullanılmaktadır, diğer kanalın giriş çıkış hatları ise kart üzerinde bulunan I/O konektörlerinden birine çıkmaktadır.

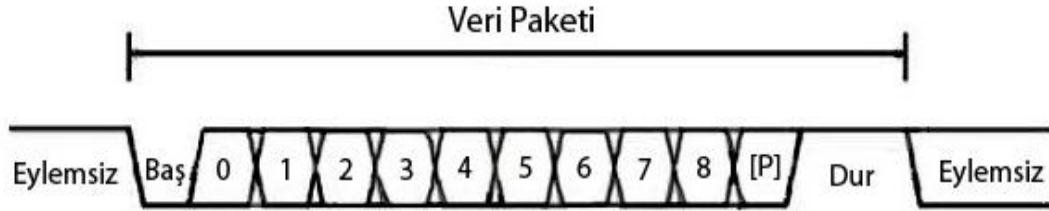
Bu tasarımda DB9 tipi konektöre modem kablosu ile bağlanılmış, modem kablusunun diğer ucu bilgisayara bağlanmıştır. Kullanılan bilgisayarda seri kanal portu bulunmadığından, bilgisayar tarafındaki bağlantının yapılabilmesi için bir USB - RS 232 çevirici kullanılmıştır. Bu çevirici Digitus firmasından temin edilmiştir.

Seri iletişim genel olarak genelgeçer eşzamansız alıcı/verici (UART) ve genelgeçer eşzamanlı-eşzamansız alıcı/verici (USART) olmak üzere iki formdan oluşur. USART iletişimde alıcı ve verici aynı saat işaretini paylaşır. Verici alıcıya veriyle birlikte saat işaretini de gönderir, böylece alıcı veriyi okuyacağı zamanı bilir. UART iletişimde ise alıcı ve vericinin aynı saat işaretini kullanması gerekmez. Bu iletişim formunda vericiyle alıcı, veri paketine eklenen zamanlama bitleriyle senkronize olurlar.

UART vericisi, veriyi baytlar halinde alır ve bitler halinde seri olarak gönderir. UART alıcısı ise bitler halinde aldığı veriyi tekrar bayt şekline getirir. UART modülleri seri ve paralel modlar arasında geçiş yapabilmek için kayan yazmaçlar içerirler.

Paketin gönderimi her zaman "BAŞLANGIÇ" bitiyle başlar. Bu bit alıcıyı paket gönderimi yapılacağı konusunda bilgilendirir. Bu biti aldıktan sonra alıcı veri gönderiminin başlamasını beklemeye başlar ve senkronize olmaya çalışır. Senkronizasyon kurulduktan sonra veri, bitler halinde seri olarak alınmaya başlanır. Genellikle en önemsiz bit en önce gönderilir. Bütün verinin gönderimi tamamlandıktan sonra verici alıcıya, alınan verinin hata kontrolünü yapması için, eşlik biti gönderebilir. İletişim vericinin "DUR" bitini göndermesiyle sonlanır. UART iletişimi RS-232 protokolünü destekler ve RS-232 devrelerinde genelde "BAŞLANGIÇ" biti mantıksal sıfır değeriyle, "DUR" biti mantıksal bir değeriyle ifade edilir. Bu devreler iletişim yokken sayısal bir değerine bağlı olurlar. "BAŞLANGIÇ" bitinin hemen ardından en önemsiz bit yollanır, en önemli bit seçime bağlı olarak

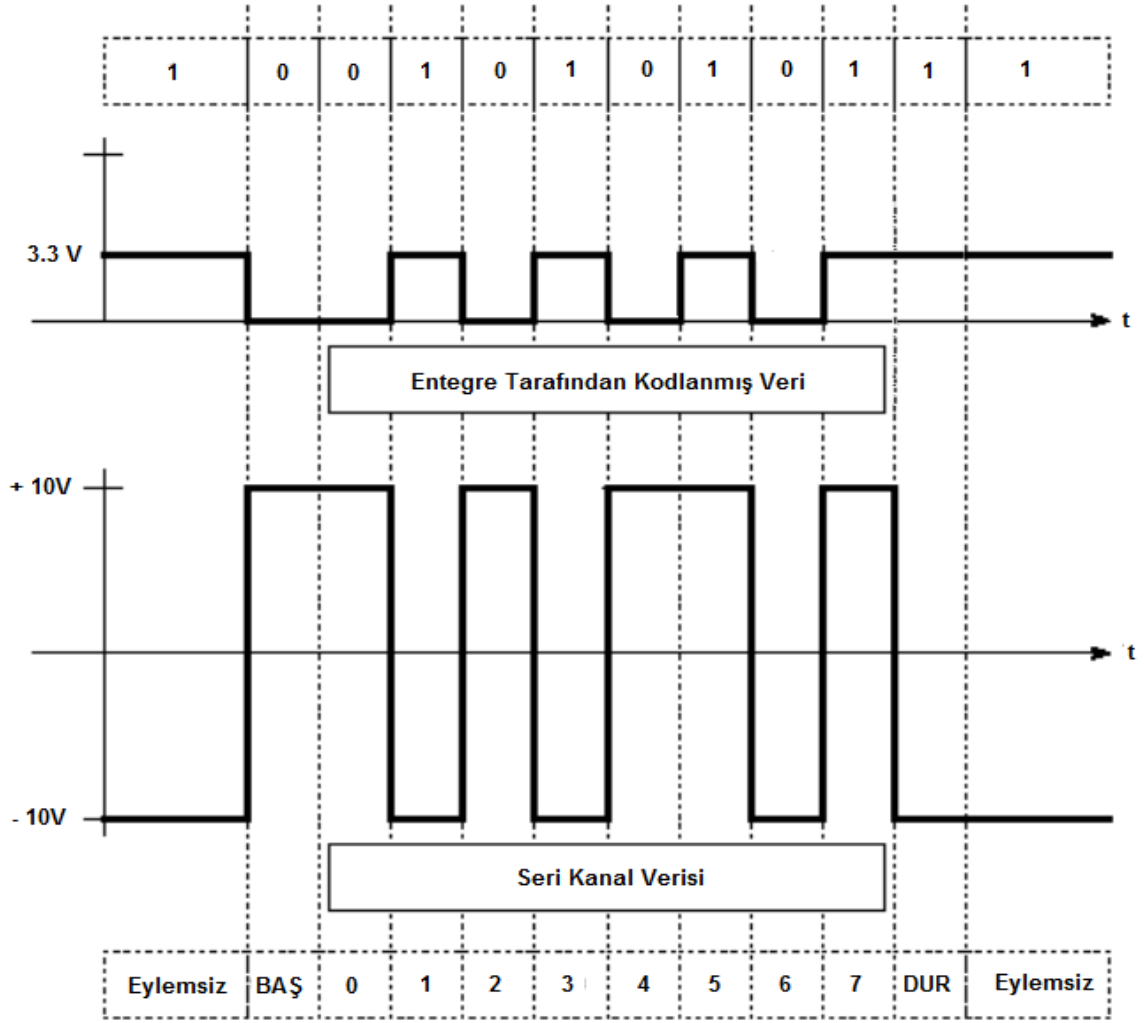
olarak ardında bir eşlik biti eklenerek en son gönderilir. RS-232 seri kanal iletişimi Şekil 33 ile gösterilmiştir [11][33][34].



Şekil 33. RS-232 Veri İletişimi

Bu şekilde iletişim olmayan durumlar "Eylemsiz" ile ifade edilmiştir bu durumlarda RS-232 devresi sayısal yüksek değerdedir. "Baş" başlangıç bitini ifade etmektedir ve bu bit sayısal düşük değerdedir. Ardından en önemsiz bittten başlanarak veri bitleri 0-8 ile gösterilmiştir. En önemli bitin ardından gelen [P] biti eşlik bitidir ve seçime bağlı olarak pakete eklenir. "DUR" tek veya iki bittten oluşan ve paketin sonlandığını haber veren sayısal yüksek değerdeki bölümdür.

Bilgisayardan gelen seri kanal verisinin gerilim değerleri +10V ile -10 V aralığındadır. ADM 3202 entegre devresi seri kanaldan gelen verileri alır ve FPGA giriş hatlarına sayısal düşük ve yüksek değerlerine uygun olarak kodlayarak gönderir. Seri kanal entegre devresine gelen seri kanal verisi ve entegre devre tarafından kodlanmış veri, bir örnek ile Şekil 34'de gösterilmiştir. Bu örnekte altılı tabanda "AA", ikili tabanda "10101010" verisi kodlanmıştır.



Şekil 34. ADM3202 Entegre Devresi Seri Kanal Kodlama Örneği

5.2 DDS Algoritmasına Değiştirilebilen Genlik ve Frekans Parametrelerinin Eklenmesi

Sabit frekans ve genlik değeriyle hazırlanan VHDL kodunda, seri kanaldan gelecek olan parametreleri alıp, bu parametrelere uygun şekilde dalga üretimi gerçekleştirilmesi amacıyla bazı değişiklikler yapılmıştır.

Koda dalga formu çeşidini seçmek için "MOD", genlik değerini seçmek için "GENLİK" ve frekans değerini seçmek için "FREKANS" ile ifade edilecek parametreler eklenmiştir.

Bunlardan "MOD" parametresi yedi farklı değer alabilen ve dört bitle ifade edilen bir parametredir. Bu parametrenin yedi farklı değeri ve karşılık geldikleri dalga formları şu şekildedir:

"0000" → 0 V Sabit Gerilim

"0001" → Sabit Gerilim

"0010" → Sinüs

"0011" → Kosinüs

"0100" → Üçgen

"0101" → Testere

"0110" → Kare

Frekans seçiminde kullanılan frekans kontrol bilgisi "F.K.B" parametresinin ayarlanabilir olması içinse yapılması gereken bir hesap vardır. Bunun için (13) numaralı denklemden F.K.B değeri çekilir ve;

$$F.K.B = \frac{2^{32} \times f_{\text{çıkış}}}{568,966 \text{ KHz}} \quad (25)$$

denklemini elde edilir. $f_{\text{çıkış}}$ parametresi seri kanaldan gelen frekans seçimine göre ayarlanabilir olacağından, dalga üretiminde kullanılan $F.K.B$ parametresinin seçilen frekansa göre güncellenmesi gerekmektedir. Bu durumda $F.K.B$ parametresi aşağıdaki denkleme uygun şekilde güncellenir;

$$F.K.B = 7548,72 \times \text{Seri Kanaldan Girilen Frekans} \quad (26)$$

Seri kanal arayüzüne girilen frekans bilgisi, geliştirme kartının seri kanal portuna 7548,72 değeriyle çarpılarak gönderilir ve VHDL kodunda 32 bit olan "F.K.B" değerine atanır.

"GENLİK" parametresi ise VHDL kodunda 8 bit büyüklüğündedir, dolayısıyla 1 ile 255 aralığındaki değerleri ifade edebilmektedir. Oluşturulan dalga formunun genliğinin "GENLİK" parametresine göre haritalanması gerekmektedir. "GENLİK" parametresinin maksimum değeri 255, D/A çevirici tarafından analoga çevrilebilen

maksimum genlik değeri 10 Volttur. Seri kanal arayüzü üzerinden genlik seçimi yapılırken volt birimi üzerinden yapılacağından;

$$\frac{GENLİK}{255} = \frac{Seri Kanaldan Girilen Genlik Değeri}{10} \quad (27)$$

$$GENLİK = 25,5 \times Seri Kanaldan Girilen Genlik Değeri \quad (28)$$

denklemini elde edilir. Seri kanal arayüzüne girilen genlik bilgisi, geliştirme kartının seri kanal portuna 25,5 değeriyle çarpılarak gönderilir ve VHDL kodunda 8 bit olan *GENLİK* değerine atanır.

0V sabit gerilim üretiminde;

Seri kanal verisinde "MOD" = "0000" ve "GENLİK" = "00000000" olarak gelmektedir. Çizelge 3 ile gösterilmiş olan bit haritasına uygun olarak D/A çevirici entegre devresinin SDIN hattına periyodik olarak "000100000000000000000000" değeri gönderilir.

Genliği ayarlanabilir sabit gerilim üretiminde;

Seri kanal verisinde "MOD" = "0001" ve "GENLİK" değeri seri kanal arayüzünden seçilen değere uygun olarak gelmektedir. "GENLİK" değeri 8 bit, D/A çevirici entegre devresi giriş sinyali 16 bit olduğundan, "GENLİK" verisi giriş verisinin en önemli 8 biti olarak atanır, en önemsiz bitler sıfır olarak atanır. Frekans bilgisine ihtiyaç duyulmadığından SDIN hattına periyodik olarak "00010000" & "GENLİK" & "00000000" olarak gönderilir. "&" işareti bitlerin birbirine seri olarak eklenmesini ifade eder. Bu şekilde yapılan tasarımda seri kanal arayüzünden 10 volt genlik değeri seçildiğinde;

$$GENLİK = 25,5 \times 10 = 255 \rightarrow "11111111" \quad (29)$$

"GENLİK" değeri 255 olarak hesaplanır ve ikili aritmetik tabanda "11111111" değerine karşılık gelir, D/A çevirici entegre devresi giriş değeri ise "1111111100000000" olur, bu değer onlu tabanda 65280 değerine karşılık gelir. (2) numaralı denklemden yola çıkılarak;

$$V_{\text{çıkış}} = -10 + 20 \left[\frac{65280}{65536} \right] = 9,92 \text{ V} \quad (30)$$

sonucu hesaplanır. Bu durumda sabit gerilim üretilirken oluşan en büyük hata değeri $10 - 9.92 = 0,08$ volt değerinde olacaktır. Buradan genlik kontrolü hata yüzdesi;

$$\text{Genlik Kontrolü Hata Yüzdesi} = \frac{0,08}{10} \times 100 = \% 0,8 \quad (31)$$

olarak hesaplanır.

Genlik çözünürlüğünün hesaplanması için "GENLİK" parametresinin seri kanaldan gelen ilk 8 bitinden en önemsiz bitin değişmesiyle, bu parametrenin ne kadar değişeceği hesaplanır ve bu fark (2) numaralı denkleme yerleştirilir Bu hesabın yapılması için "GENLİK" parametresinin bir artırılmasıyla elde edilen iki değer olan "0000000100000000" = 256 ve "0000001000000000" = 512 seçilir;

$$V_{\text{çözünürlük}} = \left(-10 + 20 \left[\frac{512}{65536} \right] \right) - \left(-10 + 20 \left[\frac{256}{65536} \right] \right) \quad (32)$$

$$V_{\text{çözünürlük}} = 0,08 \text{ Volt} \quad (33)$$

olarak hesaplanır.

Genliği ve frekansı ayarlanabilir sinüs/kosinüs dalgası üretiminde;

Seri kanal verisinde "MOD" = "0010" veya "0011", "GENLİK" ve "F.K.B" değerleri seri kanal arayüzünden seçilen değere uygun olarak gelmektedir. Seri kanaldan

seçilen frekans değeri 7548,72 değeriyle çarpılarak DDS algoritmasında 32 bit "F.K.B" değerine atanır. Aynı şekilde seçilen genlik değeri 25,5 değeriyle çarpılarak 8 bit "GENLİK" parametresine atanır. Sinüs dalgasında genlik değeri her saat çevriminde değiştiğinden, her saat çevriminde güncellenen bir genlik algoritması yazılmıştır.

```
if ÇİZELGE(15) = '1'           then
    pozitif_seviye <= ÇİZELGE - "10000000000000000";
    ölçeklenmiş_seviye <= pozitif_seviye * GENLİK;
    sinüs_çıkış <= "10000000000000000000000000" + ölçeklenmiş_seviye;
elsif ÇİZELGE(15) = '0'           then
    negatif_seviye <= "10000000000000000000000000" - ÇİZELGE;
    ölçeklenmiş_seviye <= negatif_seviye * GENLİK;
    sinüs_çıkış <= "10000000000000000000000000" - ölçeklenmiş_seviye;
end if;

çıkış <= sinüs_çıkış(23 downto 8);
```

Yukarıda gösterilen algorithmada 16 bitlik başvuru çizelgesi çıkışı verisi "ÇİZELGE" olarak ifade edilmektedir. "ÇİZELGE" verisinin en önemli biti bir ile başlıyorsa bu değer on altılı aritmetik sistemde "8000" değerine eşit veya bu değerden büyük, sıfır ile başlıyorsa bu değerden küçük demektir. On altılı aritmetik sistemde "8000" değeri kullanılan D/A çevirici entegre devresinde sıfır değerine karşılık gelmektedir, bu durumda bu değerden büyük olanlar pozitif, küçük olanlar negatif olarak adlandırılabilirler. Pozitif verilerden sıfıra denk gelen "10000000000000000000" değeri çıkarılarak "pozitif_seviye" parametresine, negatif veriler de "10000000000000000000000000" değerinden çıkarılarak "negatif_seviye" parametresine atanır ve ofset değerleri hesaplanır. Ardından bu ofset değerleri pozitif veriler için ve negatif veriler için ayrı ayrı "GENLİK" değeri ile çarpılarak ölçeklenir ve "ölçeklenmiş_seviye" parametresine atanır. Daha sonra bu ölçeklenmiş değer pozitif veriler için "10000000000000000000000000" değeriyle toplanır, negatif veriler

için "10000000000000000000000000" değerinden çıkarılır ve "sinüs_çıkış" parametresine atanır. Son olarak "sinüs_çıkış" parametresinin en önemli 16 biti alınarak "çıkış" parametresine atanır. "00010000" & "çıkış" verisi D/A çevirici entegre devresinin SDIN hattına her saat çevriminde güncellenerek gönderilir.

Genliği ve frekansı ayarlanabilir üçgen dalga üretiminde;

Seri kanal verisinde "MOD" = "0100", "GENLİK" ve "F.K.B" değerleri seri kanal arayüzünden seçilen değere uygun olarak gelmektedir. Seri kanaldan seçilen frekans değeri 7548,72 değeriyle çarpılarak DDS algoritmasında 32 bit "F.K.B" değerine atanır. Aynı şekilde seçilen genlik değeri 25,5 değeriyle çarpılarak 8 bit "GENLİK" parametresine atanır. Üçgen dalgada genlik değeri her saat çevriminde değiştiğinden, her saat çevriminde güncellenen bir genlik algoritması yazılmıştır.

FAZ <= FAZ_BİRİKECİ(31 downto 18)

ÜÇGEN_ADRES <= FAZ_BİRİKECİ (31 downto 16)

if FAZ <= "1000000000000000" then # 0-180°

üçgen_çıkış <= ÜÇGEN_ADRES(14 downto 0) & '0';

else # 180-360°

üçgen_çıkış <= "1111111111111111" - ÜÇGEN_ADRES(14 downto 0) & '0';

end if;

Yukarıdaki algorithma kırılarak 14 bite indirgenmiş faz birkeci verisi "FAZ" ile, 16 bite indirgenmiş faz birkeci verisi "ÜÇGEN_ADRES" ile ifade edilmiştir . FAZ değerinin "1000000000000000" den küçük olması 0-180° aralığına, dolayısıyla pozitif verilere, bu değerden büyük olması ise negatif verilere işaret eder. Veriler pozitifken faz birkeci çıkışının 15 biti direk alınıp, son bit olarak bir adet '0' biti eklenerek "üçgen_çıkış" parametresine atanırken, negatif skalada pozitif veriler için hesaplanan değer "1111111111111111" değerinden çıkarılarak atama yapılır. Son olarak "üçgen_çıkış" parametresi sinüs dalgası için hazırlanmış algorithmadan geçirilir ve bu algorithmadaki "ÇİZELGE" parametresine atanır. Ölçekleme algoritmasından geçerek "çıkış" parametresi ile algorithmadan çıkan veri

"00010000" & "çıkış" şeklinde D/A çevirici entegre devresinin SDIN hattına her saat çevriminde güncellenerek gönderilir.

Genliği ve frekansı ayarlanabilir testere dalga üretiminde;

Seri kanal verisinde "MOD" = "0101", "GENLİK" ve "F.K.B" değerleri seri kanal arayüzünden seçilen değere uygun olarak gelmektedir. Seri kanaldan seçilen frekans değeri 7548,72 değeriyle çarpılarak DDS algoritmasında 32 bit "F.K.B" değerine atanır. Aynı şekilde seçilen genlik değeri 25,5 değeriyle çarpılarak 8 bit "GENLİK" parametresine atanır. Testere dalgada genlik değeri her saat çevriminde değiştiğinden, her saat çevriminde güncellenen bir genlik algoritması yazılmıştır.

ÜÇGEN_ADRES <= FAZ_BİRİKECİ (31 downto 16)

testere_çıkış <= ÜÇGEN_ADRES(15 downto 0);

Yukarıdaki algoritmada kırılarak 16 bite indirgenmiş faz birikeci verisi "ÜÇGEN_ADRES" ile ifade edilmiştir. Bu değer doğrusal olarak artıp, taşma noktasından sonra sıfırlandığından testere dalga formundadır ve her saat çevriminde "testere_çıkış" parametresine atanmaktadır. Son olarak "testere_çıkış" parametresi sinüs dalgası için hazırlanmış algorithmadan geçirilir ve bu algorithmadaki "ÇİZELGE" parametresine atanır. Ölçekleme algoritmasından geçerek "çıkış" parametresi ile algorithmadan çıkan veri "00010000" & "çıkış" şeklinde D/A çevirici entegre devresinin SDIN hattına her saat çevriminde güncellenerek gönderilir.

Genliği ve frekansı ayarlanabilir kare dalga üretiminde;

Seri kanal verisinde "MOD" = "0110", "GENLİK" ve "F.K.B" değerleri seri kanal arayüzünden seçilen değere uygun olarak gelmektedir. Seri kanaldan seçilen frekans değeri 7548,72 değeriyle çarpılarak DDS algoritmasında 32 bit "F.K.B" değerine atanır. Aynı şekilde seçilen genlik değeri 25,5 değeriyle çarpılarak 8 bit "GENLİK" parametresine atanır. Kare dalgada genlik değeri her saat çevriminde değiştiğinden, her saat çevriminde güncellenen bir genlik algoritması yazılmıştır.

```

FAZ <= FAZ_BİRİKECİ( 31 downto 18 )
if FAZ <= "10000000000000" then                                # 0-180°
    kare_çıkış <= '1' & "GENLİK" & "0000000";
else                                                                # 180-360°
    kare_çıkış <= '0' & NOT "GENLİK" & "0000000";
end if;

```

Yukarıdaki algorithma kırpılarak 14 bite indirgenmiş faz birikeci verisi "FAZ" ile ifade edilmiştir . FAZ değerinin "10000000000000" den küçük olması 0-180° aralığına dolayısıyla pozitif verilere, bu değerden büyük olması ise negatif verilere işaret eder. Veriler pozitifken '1' bitine 8 bitlik "GENLİK" parametresi ve "0000000" değeri eklenerek "kare_çıkış" parametresine atanırken, negatif skalada '0' bitine 8 bitlik "GENLİK" parametresinin ikili aritmetik tabandaki tersi ve "0000000" değeri eklenerek atama yapılır. İşlemler sonucunda "00010000" & "kare_çıkış" şeklindeki veri D/A çevirici entegre devresinin SDIN hattına her saat çevriminde güncellenerek gönderilir.

5.3 Seri Kanal Arayüzünün Hazırlanması ve Baud Hızının Belirlenmesi

Kullanılan VHDL kodunda, seri kanal iletişimi parametrelerinden biri olan baud hızına uygun bir saat işareti kullanılması gerekmektedir. Bu nedenle kullanılan 33 MHz frekansındaki asıl saat giriş işaretinden seri kanal baud hızlarına uygun frekansta saat işaretleri üretilmiştir.

33 MHz asıl saat işaretinin 429 çevrimi boyunca düşük seviyede tutulan, bir sonraki 429 çevrimi boyunca yüksek değer alan bir seri kanal saati oluşturulmuştur. Bu saatin bir çevrimi 33 MHz saat girişinin $429 + 429 = 858$ çevrimine denk gelmektedir ve referans saat girişinin frekansı;

$$\frac{33 \text{ MHz}}{858 \text{ çevrim}} = 38461 \text{ Hz} \quad (34)$$

olarak hesaplanır.

Üretilen bu saat, seri kanalın sabit baud hızlarından 38400 değerine denk gelmektedir. Bu saat işareti kullanılarak diğer seri kanal baud hızlarından olan 1200, 2400, 4800, 9600 ve 19200 saatleri de elde edilmiştir ve seri kanal arayüzünde kullanılan baud değerine göre bu saat işaretlerinden uygun olan seçilip kullanılabilir. Bu tasarımda seri kanal arayüzünde 9600 baud hızı kullanılmaktadır. Bu değer 9600 bit/s hızına denk gelmektedir.

Seri kanal arayüzü hazırlanırken Microsoft Visual Studio 2010 Express programı kullanılmıştır. Öncelikle arayüz üzerinden seçimi yapılacak parametreler belirlenmiş ve arayüze "Frekans Ayarı", " Genlik Ayarı" ve "Dalga Formu Seçimi" kutuları eklenmiştir.

"Frekans Ayarı" seçim kutusunun minimum değeri 0 Hz, maksimum değeri ise (17) numaralı denklemdeki hesaptan $170689,8$ Hz'dir. DDS algoritmasının çözünürlüğü (16) numaralı denklemde gösterildiği üzere $0,0001325$ Hz olduğu halde bu boyuttaki frekans değişimi gözle görülemeyeceğinden bu kutuda yukarı-aşağı 1 Hz değerinde değişimler yapılabilir. İstenilen değer direkt olarak girilebilir. 1 Hz'lik frekans çözünürlüğünden dolayı seri kanal arayüzünde seçilebilecek frekans değerleri 0-170689 Hz aralığında sınırlandırılmıştır. Bu sınırlamalarla girilen frekans değeri (26) numaralı denklemden yola çıkılarak $7548,72$ değeri ile çarpılır ve bayt formatına dönüştürülür. Frekans değeri 4 bayt, dolayısıyla 32 bit büyüklüğünde bir parametreye atanır ve gönderilecek olan 7 bayt büyüklüğündeki pakete işlenir.

"Genlik Ayarı" seçim kutusunun minimum değeri 0 volt, maksimum değeri 10 voltur. Genlik çözünürlüğü (33) numaralı denklemde hesaplandığı üzere 0,08 voltur. Seri kanal arayüzü genlik ayarı kutusunda bu değer yuvarlanarak 0,1 volt olarak kullanılmıştır. Bu kutuda yukarı aşağı 0,1 volt değerinde değişimler yapılabilir gibi, istenilen değer direkt olarak da girilebilir. Girilen genlik değeri (28) numaralı denklemdeki hesaptan dolayı 25,5 ile çarpılarak bayt formatına dönüştürülür. Genlik değeri 1 bayt, dolayısıyla 8 bit büyüklüğünde bir parametreye atanır ve gönderilecek pakete işlenir.

"Dalga Formu Seçimi" kutusunda seçilebilecek dalga tipleri gösterilmektedir. Bu dalga tipleri ve ikili aritmetik taban karşılıkları "MOD" parametresi açıklanırken

verilmiştir. Bu değerler 4 bit olduğundan "0000" boş verisine eklenerek 1 bayt halinde gönderilecek pakete işlenmektedir.

Bu üç kutudan yapılan seçimler sonucunda toplam 6 bayt veri elde edilir. Seri kanal paketinin VHDL kodu tarafından tanınması için 1 bayt büyüklüğünde önceden seçilmiş ve hem FPGA tarafında hem arayüz tarafında bilinen bir veri de pakete eklenerek 7 baytlık paket tamamlanır. Oluşturulan bu paket 9600 baud hızı ve bilgisayarda kullanılan seri kanal port numarasıyla seri kanal üzerinden gönderilir. Bu 7 baytlık, dolayısıyla $7 \times 8 = 56$ bitlik parametre seti paketinin gönderimi;

$$Paket\ Gönderim\ Süresi = \frac{1\ s}{9600\ bit} \times 56\ bit = 0,00583\ s \quad (35)$$

sürecektir.

Seri kanal arayüzünün görünümü ve dalga formu seçim ekranı Şekil 35'deki gibidir.

Form1

**Fonksiyon Üretici
Seri Kanal Arayüzü**

Frekans Ayarı
500 Hz

Genlik Ayarı
10,0 V

Dalga Formu
0 V Sabit Gerilim

GÖNDER

0 V Sabit Gerilim
0 V Sabit Gerilim
Sabit Gerilim
Sinüs
Kosinüs
Üçgen
Testere
Kare

Şekil 35. Seri kanal Arayüzü Görünümü ve Dalga Formu Seçim Ekranı

6. FONKSİYON ÜRETECİ DOĞRULAMASI VE TASARIM BELİRTİMLERİ

6.1 Fonksiyon Üreteci Tasarım Belirtilimleri

Tasarlanan fonksiyon üreticinin skala, çözünürlük, hata oranı gibi tasarım parametreleri önceki bölümler boyunca hesaplanmıştır. DDS algoritmasının gerçek belirtilimleri Çizelge 11 ile verilmiştir. Seri kanal arayüzünde belirli kısıtlamalar kullanıldığından bu arayüzde kullanılan belirtilimler de Çizelge 12 ile verilmiştir.

Çizelge 11. DDS Algoritması Tasarım Belirtilimleri

Veriyolu Genişliği	16 Bit
Frekans Çözünürlüğü	0,0001325 Hz \approx 132,5 μ Hz
Frekans Aralığı	0 - 170689,8 Hz \approx 0 - 170,79 KHz
Örnekleme Frekansı	2275555,6 Örnek/s \approx 2,28 MÖrnek/s
Genlik Çözünürlüğü	0,08 Volt
Genlik Aralığı (Tepeden Tepeye)	- 10 +10 Volt
Genlik Hata Oranı	%0,8
SNR	98,08 dB
Başvuru Çizelgesi Boyutu	4096 Bit \approx 4 Kbit
Adres Yolu Genişliği	12 Bit
Toplam Örnek Sayısı	1024

D/A Çevrim Periyodu	$57 \times 30,3 \text{ ns} \approx 1727 \text{ ns} \approx 1,73 \text{ } \mu\text{s}$
----------------------------	---

Çizelge 12. Seri Kanal Arayüzü Tasarım Belirtilimleri

Frekans Çözünürlüğü	1 Hz
Frekans Aralığı	0 - 170689 Hz
Genlik Çözünürlüğü	0,1 Volt
Genlik Aralığı (Tepe)	10 Volt
Baud Frekansı	9600 Hz
Paket Gönderim Süresi	$0,00583\text{s} = 5,83 \text{ ms}$

Fonksiyon üreticinin tepki süresi ise D/A çevrim periyodu ile seri kanal paket gönderim süresinin toplanmasıyla bulunur. D/A çevrim süresi bu toplamda ihmal edilebilir olduğundan paket gönderim süresi yaklaşık olarak tepki süresine eşittir.

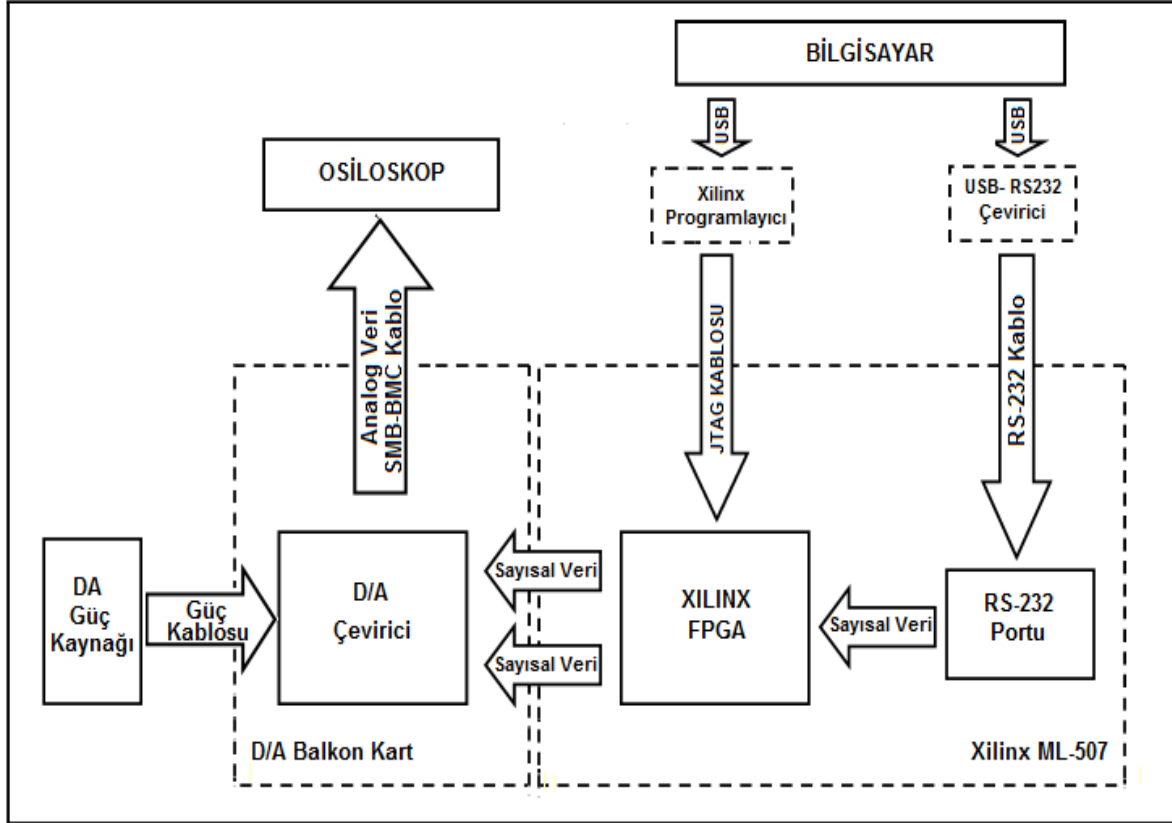
DDS fonksiyon üretici algoritması ile üretilen dalgalar D/A çevirici bloğu kullanılmadan, sayısal veriler halinde kullanılmak istenirse, analoga çevrim frekans limiti ortadan kalkacağından frekans aralığı artacaktır. Üretilen dalga formlarının sayısal olarak kullanıldığı durumda, VHDL sentez raporu verilerinden yola çıkılarak maksimum frekans değeri 109,182 MHz, çıkış tepki süresi ise 3,264 ns olarak bulunmuştur.

6.2 Algoritmaların Donanım Üzerinde Doğrulanması

Hazırlanan algoritmaların ve kodların donanımlar üzerinde doğrulanabilmesi için donanımsal bir kurulum yapılması gerekmektedir. Bu kurulum için gerekli donanım elemanları Çizelge 13 ile belirtilmiştir. Yapılan doğrulama kurulumunun şeması Şekil 36 ile gösterilmiştir.

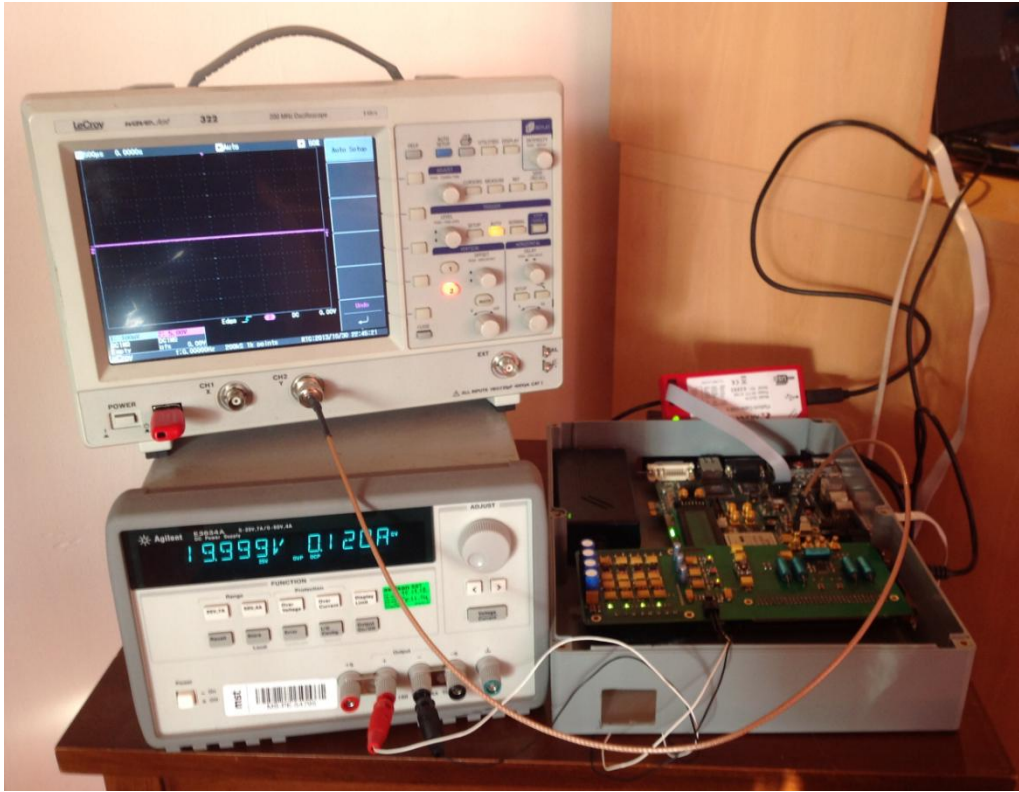
Çizelge 13. Donanım Elemanları

Donanım	Kullanımı
Xilinx ML-507	Geliştirme Kartı
D/A Çevirici Balkon Kartı	D/A Çevirici
Xilinx JTAG Kablosu	FPGA Programlama Kablosu
DC Güç Kaynağı (20 V - 0,2 A)	D/A Çevirici Balkon Kartı Güç Beslemesi
Osiloskop	Dalgaların Gözlemlenmesi
SMB-BNC Kablo	D/A Çevirici Kart - Osiloskop Bağlantı Kablosu
Muz - Dişi Pim Kablo (2 Adet)	D/A Çevirici Balkon Kartı Güç Beslemesi (+/-)
Seri Kanal USB Çevirici	Bilgisayarda Seri Kanal Portu Oluşturma
Seri Kanal Kukla Modem Kablosu	Bilgisayar- ML 507 Seri Kanal Bağlantısı

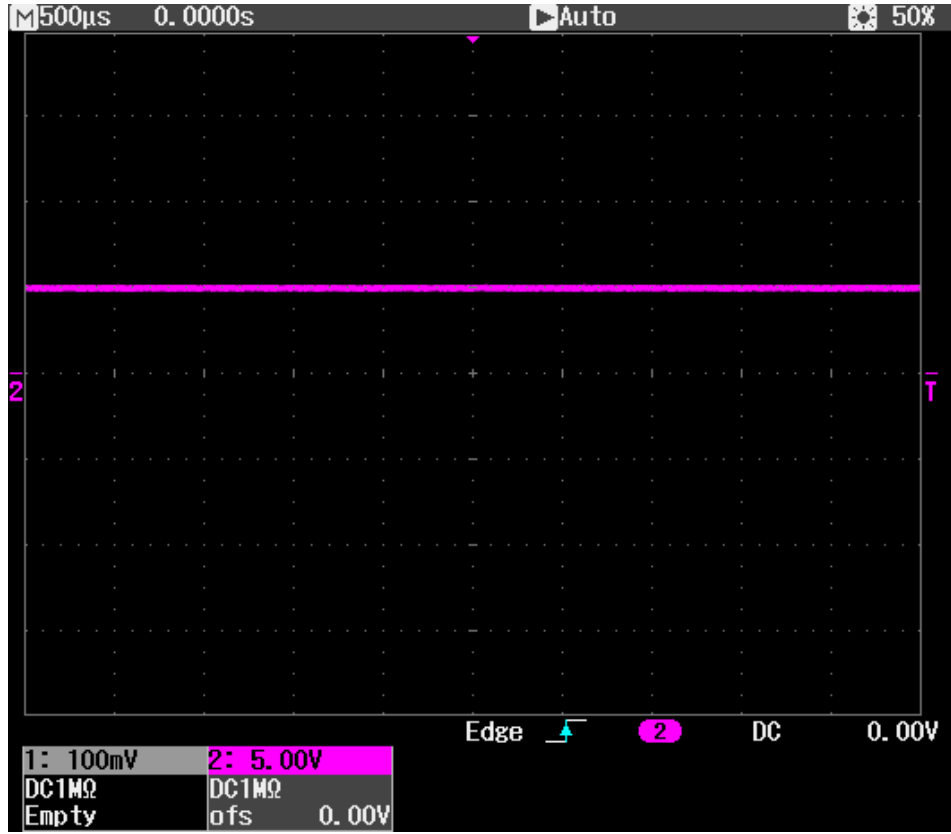


Şekil 36. Doğrulama Kurulum Şeması

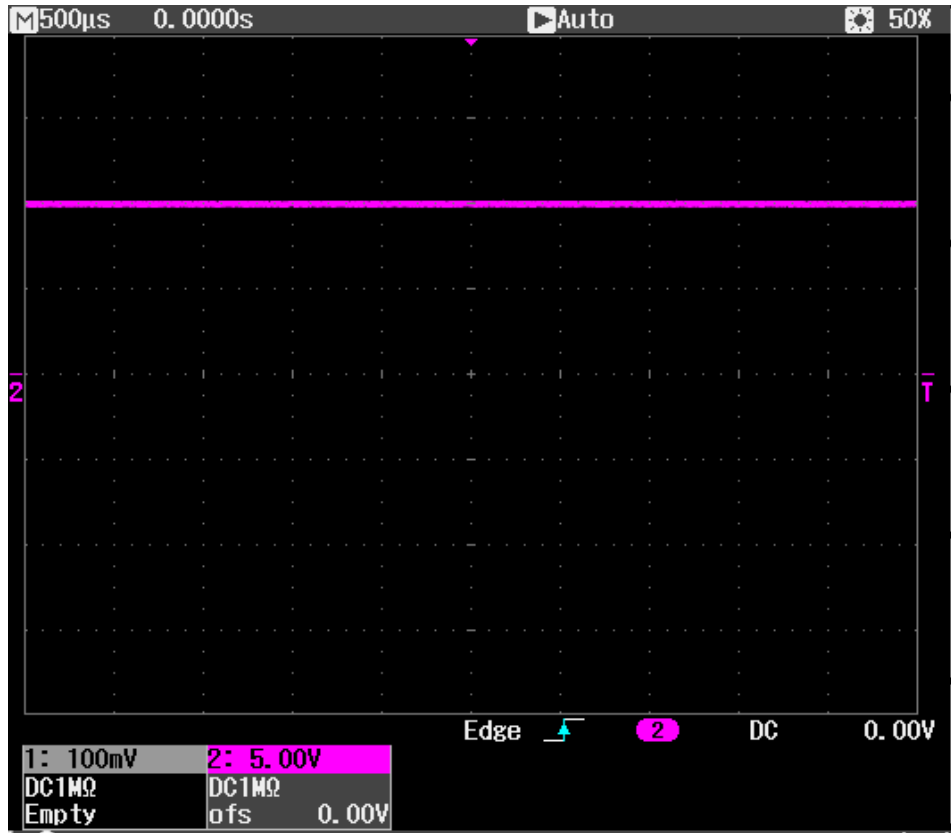
Donanımsal doğrulama yapılması için seri kanal arayüzünden bazı seçimler yapılmış ve bu seçimlere uygun dalga formlarının üretildiği donanım üzerinde doğrulanmıştır. Doğrulama kurulumu Şekil 37 ile, çeşitli parametre setleriyle üretilen örnek dalga formlarının osiloskop ekran görüntüleri Şekil 38, Şekil 39, Şekil 40, Şekil 41, Şekil 42, Şekil 43 , Şekil 44, Şekil 45, Şekil 46 ve Şekil 47 ile gösterilmiştir. Her şeklin seri kanal arayüz parametreleri şekil bilgisinde yer almaktadır.



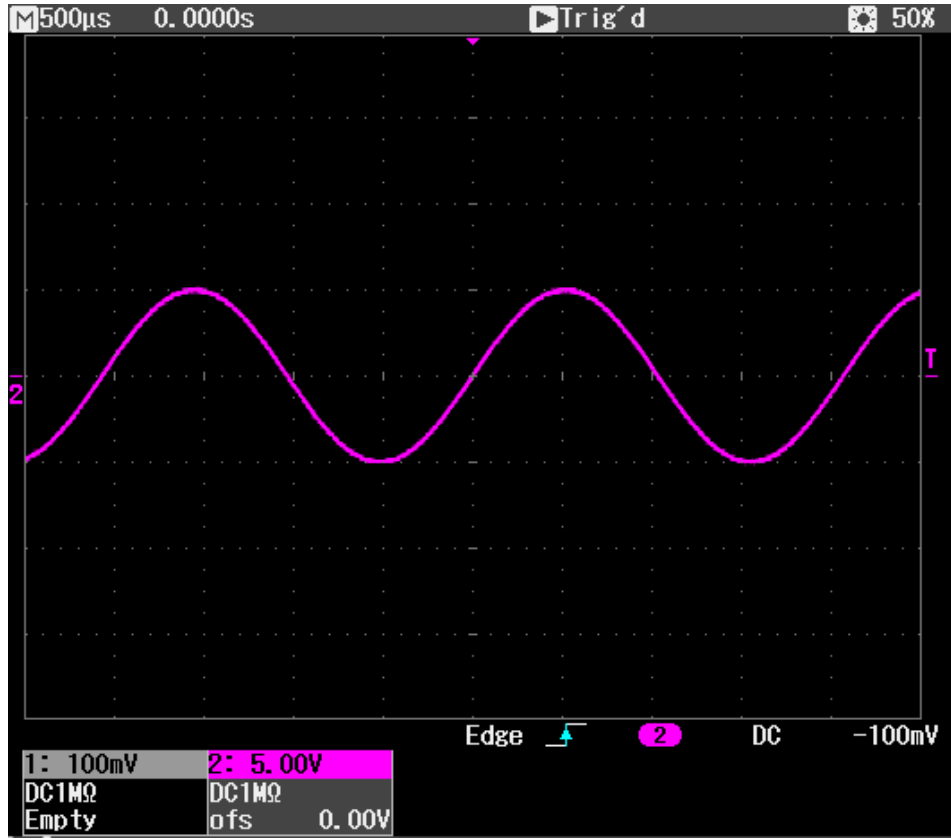
Şekil 37. Doğrulama Kurulumu



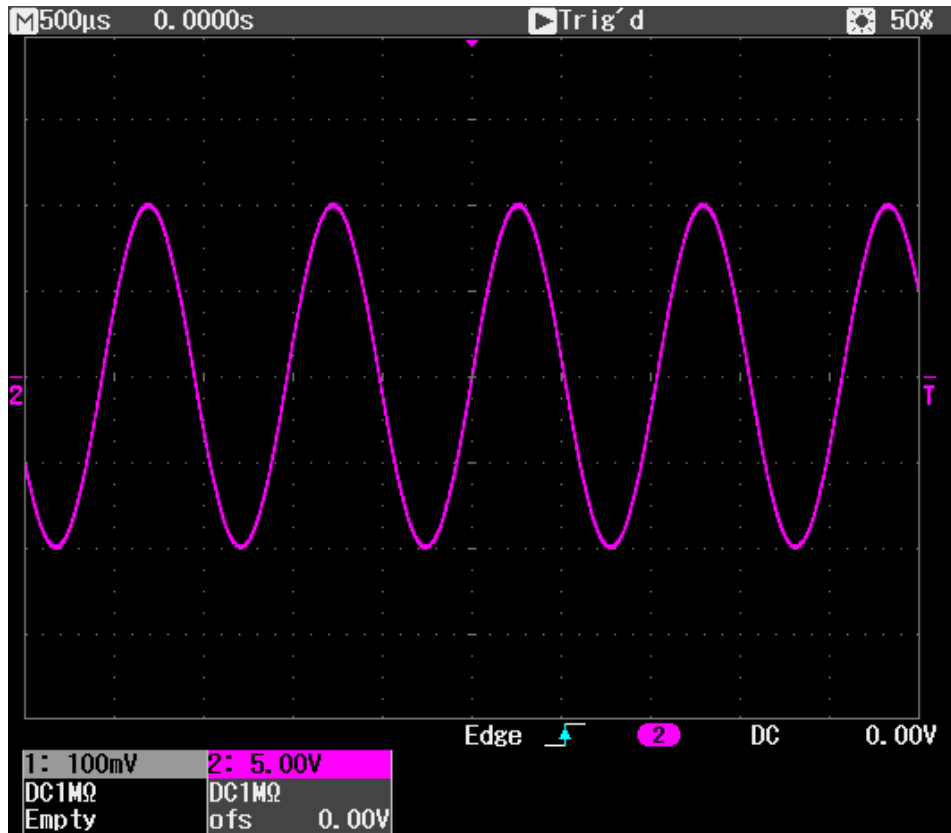
Şekil 38. 5 Volt Sabit Gerilim Üretim



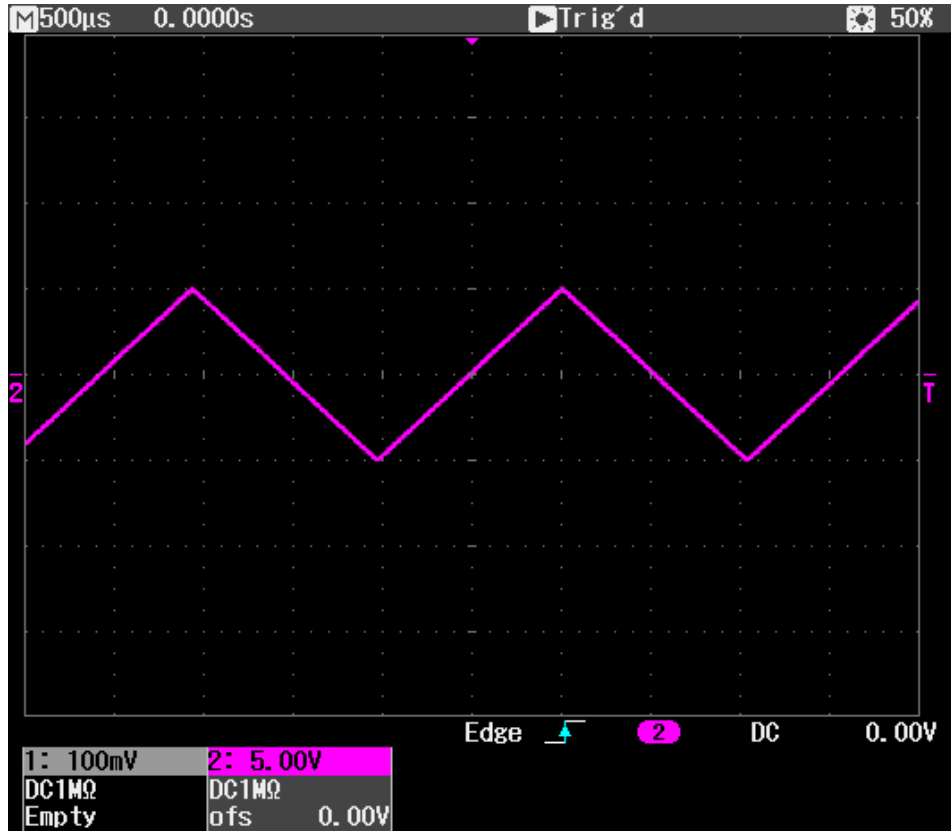
Şekil 39. 10 Volt Sabit Gerilim Üretimi



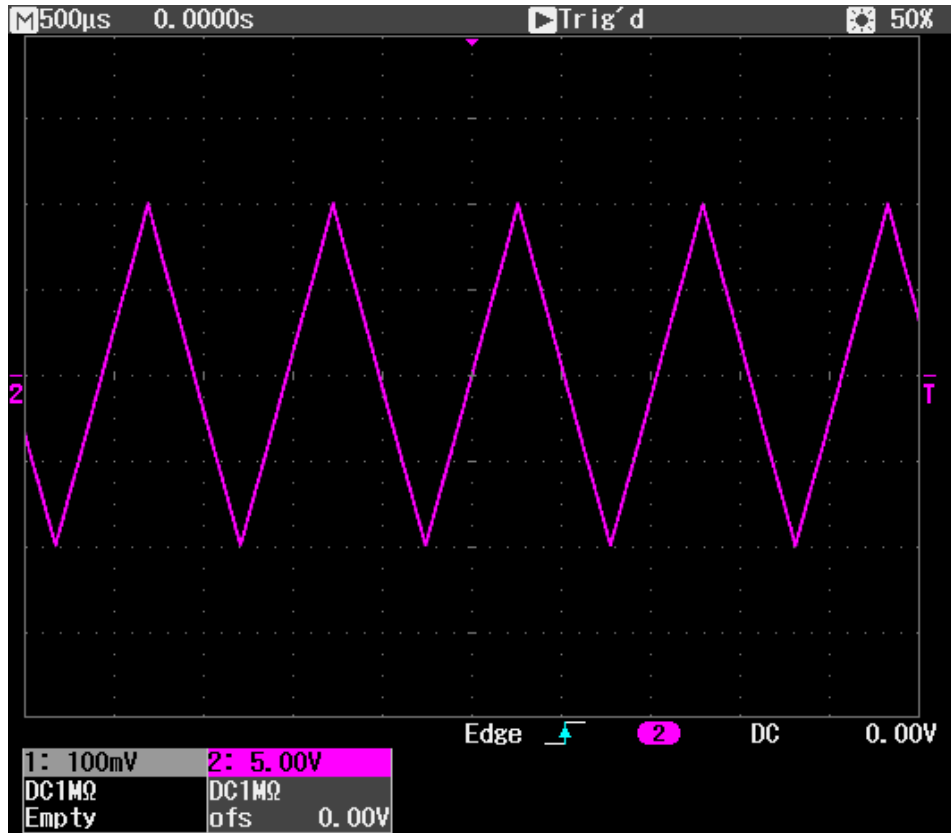
Şekil 40. 5 Volt 500 Hz Sinüs/Kosinüs Dalgası Üretimi



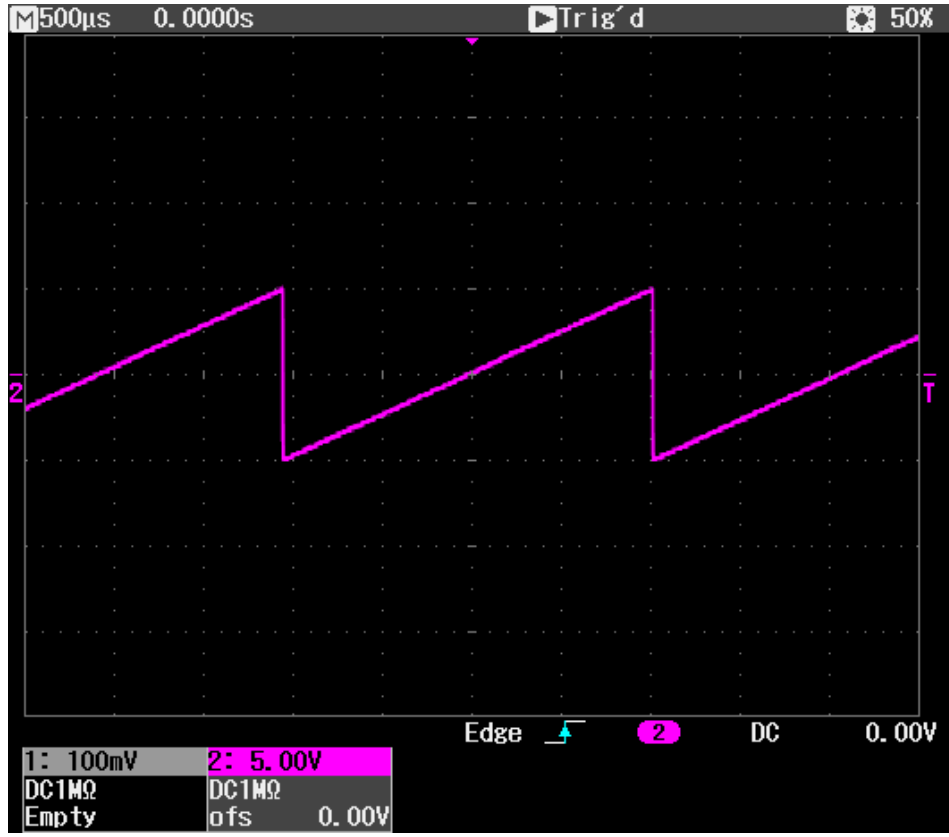
Şekil 41. 10 Volt 1000 Hz Sinüs/Kosinüs Dalgası Üretimi



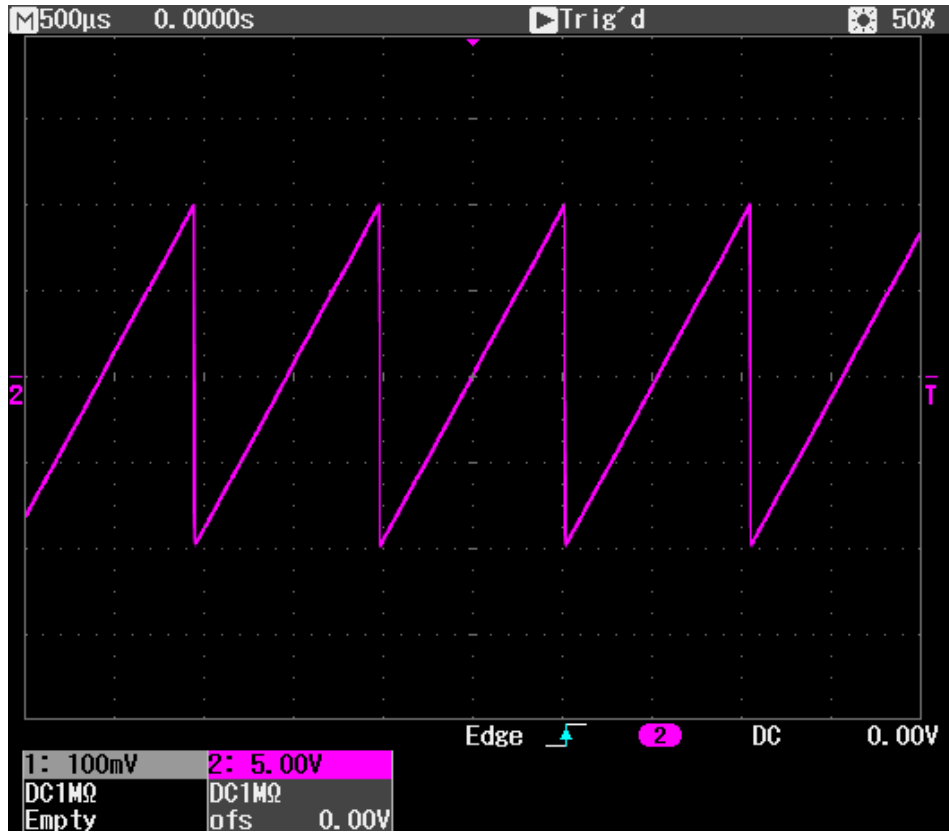
Şekil 42. 5 Volt 500 Hz Üçgen Dalga Üretimi



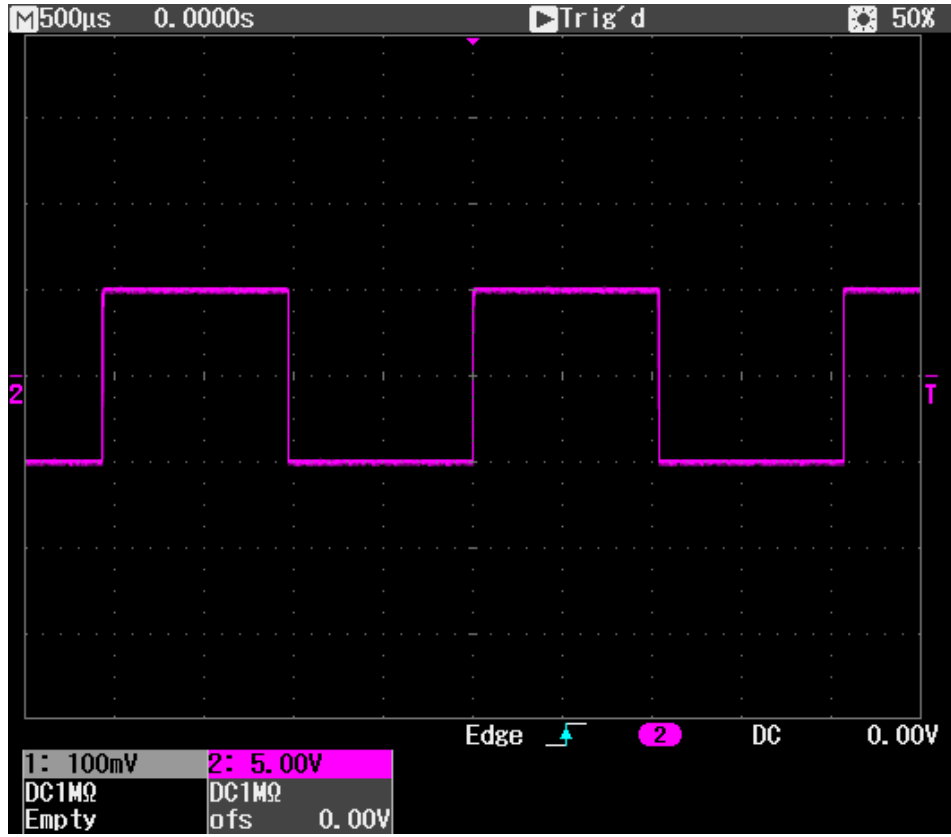
Şekil 43. 10 Volt 1000 Hz Üçgen Dalga Üretimi



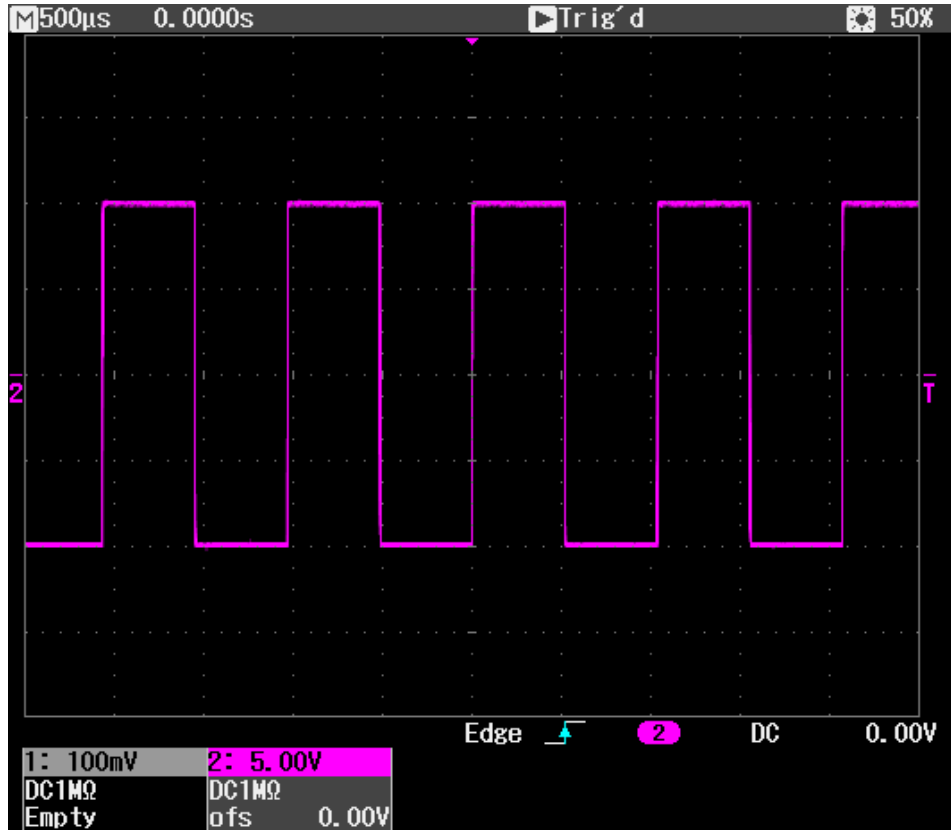
Şekil 44. 5 Volt 500 Hz Testere Dalga Üretimi



Şekil 45. 10 Volt 1000 Hz Testere Dalga Üretimi



Şekil 46. 5 Volt 500 Hz Kare Dalga Üretimi



Şekil 47. 10 Volt 1000 Hz Kare Dalga Üretimi

7. SONUÇLAR

Giriş bölümünde genel olarak yer verilen bloklar tez kapsamında sırayla detaylandırılmıştır. Öncelikle sayısalardan analoga geçişi sağlayan ve üretilen dalgaların osiloskop ile gözlemlenmesine fırsat sağlayan D/A çevirici bloğu tasarlanmıştır. Gerekli parametrelere uygun olarak seçilen bir D/A çevirici entegre devresi için balkon kart hazırlanmış ve ML507 geliştirme kartıyla birlikte kullanılmıştır. Bu blok HDL Designer programı kullanılarak kodlanmış ve optimize edilerek sayısal değerleri analoga çevirebildiği örnek bir DC değeri ile gözlemlenmiştir.

Daha sonra sinüs dalga formunun üretilmesinde kullanılacak olan başvuru çizelgesi hazırlanmıştır. Başvuru çizelgesi FPGA içerisinde bulunan BRAM'a kaydedilmiştir. Bir tam periyot için hazırlanan sinüs verisi öncelikle 16 kbit hafıza kullanılarak başvuru çizelgesine kaydedilmiştir. Daha sonra sinüs dalgasının simetri özelliği kullanılarak bu hafıza boyutu 4 kbit'e kadar düşürülmüştür.

DDS algoritmasında bulunan faz birikeci ve nicemleme devreleri VHDL kodu ile gerçekleştirilmiştir. Bu algoritmalar gerçekleştirirken başvuru çizelgesinin adres yolu genişliğine uygun biçimde nicemleme yapılmıştır. DDS algoritmasında frekans kontrol bilgisiyle kontrol edilen faz birikeci çıkışı nicemlenerek başvuru çizelgesi adreslenmiş ve başvuru çizelgesinde adreslenen hücrede bulunan veri D/A çevirici bloğuna gönderilmiştir. Genlik kontrolü de aynı şekilde VHDL kodlaması ile gerçekleştirilerek DDS algoritmasına eklenmiştir.

Genlik ve frekans parametrelerini belirlemek ve üretilecek dalga formunu seçmek üzere, gerçek zamanlı iletişim sağlayan seri kanal arayüzü tasarlanmıştır. Fonksiyon üretici VHDL kodu FPGA'nın silinip programlanabilir salt okunur belleğine (EPROM) yüklenerek geliştirme kartına her güç verildiğinde FPGA'yı programlayacak şekilde gömülü hale getirilmiştir. Hazırlanan doğrulama kurulumunda bilgisayarın seri kanalından gönderilen parametre setlerine uygun olarak dalga formlarının üretildiği gözlemlenmiştir.

Bu çalışma bütünüyle bir fonksiyon üretici gibi kullanılabileceği gibi bir ya da birden çok bloğu değiştirilerek farklı uygulamalar elde edilebilir. Örneğin doğrusalsızlık ve hata oranı gibi parametrelerden vazgeçilerek daha hızlı bir D/A çevirici bloğu kullanılabilir. Bu şekilde frekans aralığı daha geniş bir fonksiyon üretici elde edilebilir. Tezde gerçekleştirilen DDS algoritmasında başvuru çizelgesine sinüs verileri kaydedilmiştir. İleride yapılacak olan çalışmalarda birden çok başvuru çizelgesi kullanılarak, bu çizelgelere periyodik/periyodik olmayan dalga formları yüklenebilir. Aynı zamanda başvuru çizelgesine AM, FM, ASK, PSK ve FSK gibi kiplenmiş dalga formu verileri yüklenerek telekomünikasyon uygulamalarında kullanılmak üzere geliştirilebilir. Başvuru çizelgesinin boyutları büyütülerek frekans çözünürlüğü artırılabilir. Hafıza boyutlarının sınırlı olduğu durumlarda frekans çözünürlüğünden vazgeçilerek hafıza boyutları düşürülebilir. Nicemlemeden kaynaklanan zaman tabanlı seğirmeyi azaltmak için gürültü minimizasyon çalışmaları yapılabilir.

Tasarlanan fonksiyon üretici VHDL kodu, üzerinde FPGA bulunan bütün kart ve sistemlere gömülerek, haricen kullanılacak bir fonksiyon üreticine olan ihtiyacı ortadan kaldırmaktadır. Ayrıca bloklar halinde tasarlanan bu fonksiyon üreticinin blokları değiştirilerek farklı uygulamalarda kullanıma uygun hale getirilebilir. Tez sonucunda, donanım gerekleri minimize edildiği ve yeniden tasarlanabilir olduğu için pratik bir fonksiyon üretici elde edilmiştir.

KAYNAKLAR

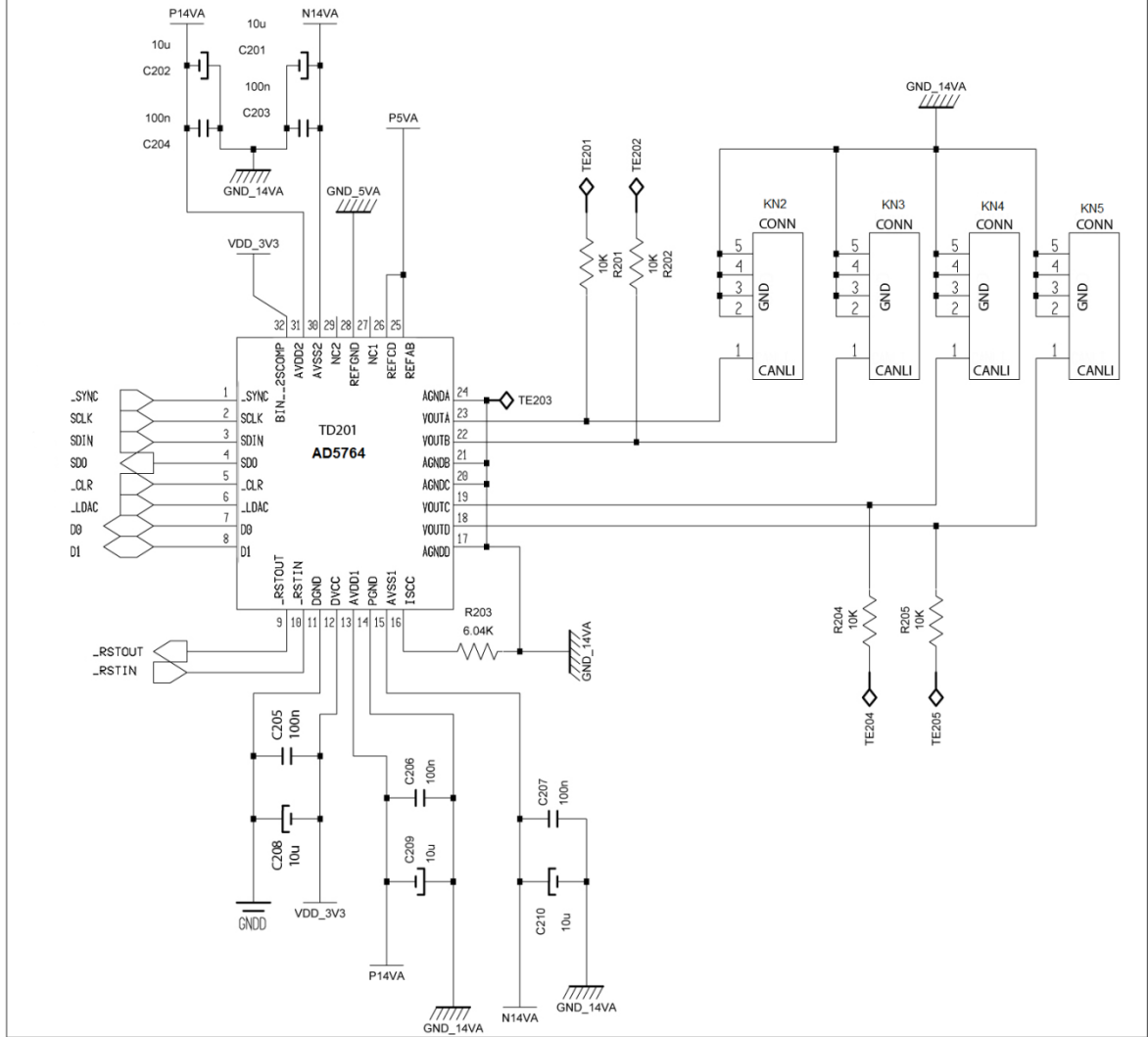
- [1] R. Yue, T. Ying, Y. Wen-ji, W. Zhang-quan, X. Sen, A Single Chip Multi-functional DDS Waveform Generator based on FPGA with SOPC Design Flow, *World Congress on Intellignet Control and Automation*, 4206–4210, Beijing, Çin, **2012**.
- [2] D. C. Larson, High speed direct digital synthesis techniques and applications, *IEEE Gallium Arsenide Integrated Circuit Symposium 20th Annual*, NO. 6, 209–212, **1998**.
- [3] D. Sunderland, CMOS/SOS frequency synthesizer LSI circuit for spread spectrum communications, *IEEE Journal of Solid-State Circuits*, NO. 4, **1984**.
- [4] Q. Wang, S. He, Z. Zhong, Design and Simulation of an Optimized DDS, *2010 International Conference on Computational Intelligence and Software Engineering*, 1–3, Eylül, **2010**.
- [5] J. Farina, J. Rodriguez-Andina, Multifrequency equipment for sensing the state of electrodes in electric-arc furnaces, *Industrial Electronics Magazine*, 4–8, **2007**.
- [6] K. Elliott, High Speed Direct Digital Synthesis for Next Generation RF Systems, *Radio Wireless Symposium 2007 IEEE*, 423–426, **2007**.
- [7] G. Xu, S. Zhang, H. Wu, S. Yang, The acquisition hardware system with direct digital synthesis and filtered back-projection imaging in electrical impedance tomography, *IEEE Engineering in Medicine and Biology 27th Annual Conference*, Vol. 7, 7758–61, Ocak, **2006**.
- [8] M. Xu, J. Hu, Y. Gao, FPGA-based Design and Implementation of Arbitrary Waveform Generator, *Control, Automation and System Engineering (CASE) International Conference 2011 IEEE*, 1–4, **2011**.
- [9] F. Yi-yuan, C. Xue-jun, Design and Simulation of DDS Based on Quartus II, *Computer Science and Automation Engineering (CSAE) International Conference 2011 IEEE*, **2011**.
- [10] Analog Devices, A technical tutorial on digital signal synthesis, *Application Note*, 1–122, **1999**.
- [11] U. Kazancıoğlu, *The Implementation of a Direct Digital Synthesis Based Function Generator Using SYSTEMC and VHDL*, Yüksek Lisans Tezi, Orta Doğu Teknik Üniversitesi, **2007**.
- [12] D. S. March, LogiCORE IP DDS Compiler v4.0, *Application Note*, 1–31, **2011**.
- [13] J. Zhang, X.-G. Li, The Front Design and Implement of Direct Digital Frequency Synthesizer Based on FPGA, *2010 International Conference on Electrical and Control Engineering*, 4816–4819, Haziran, **2010**.

- [14] J. Hsieh, G. Tsai, M. Lin, Using FPGA to implement a n-channel arbitrary waveform generator with various add-on functions, *IEEE International Conference on Field-Programmable Technology (FPT)*, 296–298, **2003**.
- [15] J. Valls, M. Kuhlmann, K. K. Parhi, Efficient mapping of CORDIC algorithms on FPGA, *Signal Processing Systems, SIPS 2000 IEEE Workshop*, 336–345, **2000**.
- [16] Maxim, INL / DNL Measurements for High-Speed Analog-to-Digital Converters (ADCs), *Maxim*, Vol. 283, 1–9, **2000**.
- [17] BİLEŞİM Yayıncılık, Fuarcılık ve Tanıtım Hizmetleri, <http://www.bilesim.com.tr/yazdir.php?t=3&id=553&sn=0.html> (erişim tarihi: Kasım, **2013**).
- [18] Xilinx, ML507 QuickStart, *Application Note*, **2008**.
- [19] Xilinx, ML 50x Schematics, *Application Note*, **2008**.
- [20] Analog Devices, Complete Quad, 16-Bit, High Accuracy, Serial Input Bipolar Voltage Output DAC, *Datasheet*, **2011**.
- [21] Xilinx, UG190 Virtex-5 FPGA User Guide, Vol. 190, *Application Note*, **2007**.
- [22] Z. Zhang, A harmonic signal generator based on DDS and SOPC, *2010 Chinese Control and Decision Conference*, 1542–1547, Mayıs, **2010**.
- [23] X. Wang, Q. Mei, High-Precision Design of DDS Based on FPGA, *2012 Third Global Congress on Intelligent Systems*, 386–389, Kasım, **2012**.
- [24] Y. Yang, J. Cai, L. Liu, A Novel DDS Array Structure with Low Phase Noise and Spurs, *Communication Technology (ICCT) IEEE 13th International Conference*, 302–306, **2011**.
- [25] D. Brandon, Determining if a Spur is Related to the DDS/DAC or to Some Other Source, *Analog Devices*, **2007**.
- [26] V. F. Kroupa, Discrete Spurious Signals and Background Noise in Direct Digital Frequency Synthesizers, *IEEE International Frequency Control Symposium*, NO. 2, **1993**.
- [27] H. T. Nicholas, H. Samuelli, An Analysis of the Output Spectrum of Direct Digital Frequency Synthesizers in the Presence of Phase-Accumulator Truncation, *41st Annual Symposium on Frequency Control*, 495–502, **1987**.
- [28] W. Kester, MT-001: Taking the Mystery out of the Infamous Formula, 'SNR= 6.02 N+ 1.76 dB and Why You Should Care, *REV. 0 Tutorial*, 1–7, **2005**.
- [29] S. Cheng and J. Jensen, Further enhancements to the analysis of spectral purity in the application of practical direct digital synthesis, *Frequency Control Symposium and Exposition 2004 IEEE International*, 462–470, **2004**.
- [30] M. G. Corporation, ModelSim ® PE Tutorial, *Datasheet*, **2013**.
- [31] Xilinx, ML507 Evaluation Platform User Guide, *Datasheet*, Vol. 347, 1–60, **2008**.

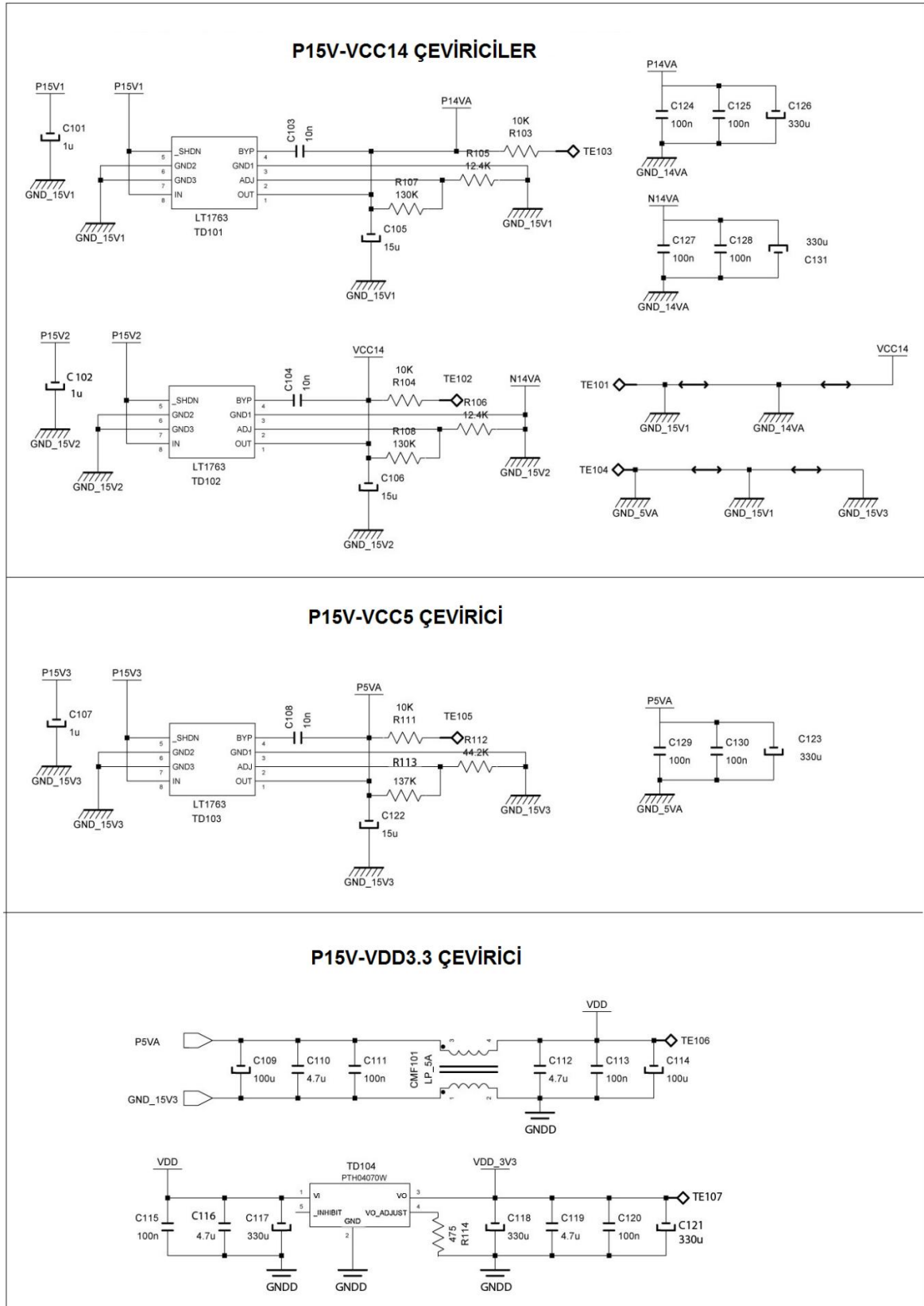
- [32] Analog Devices, ADM 3202 Line Driver/Receiver Datasheet, *Datasheet*, 1–16, **2006**.
- [33] Anonim, Serial and UART Tutorial, <http://www.freebsd.org/doc/en/articles/serial-uart.html> (erişim tarihi: Kasım, **2013**)
- [34] Chris Kuethe, AVR: Input and Output, http://www.lizarum.com/assignments/physical_computing/2010/avr/serial.html (erişim tarihi: Kasım, **2013**).

EKLER

Ek 1

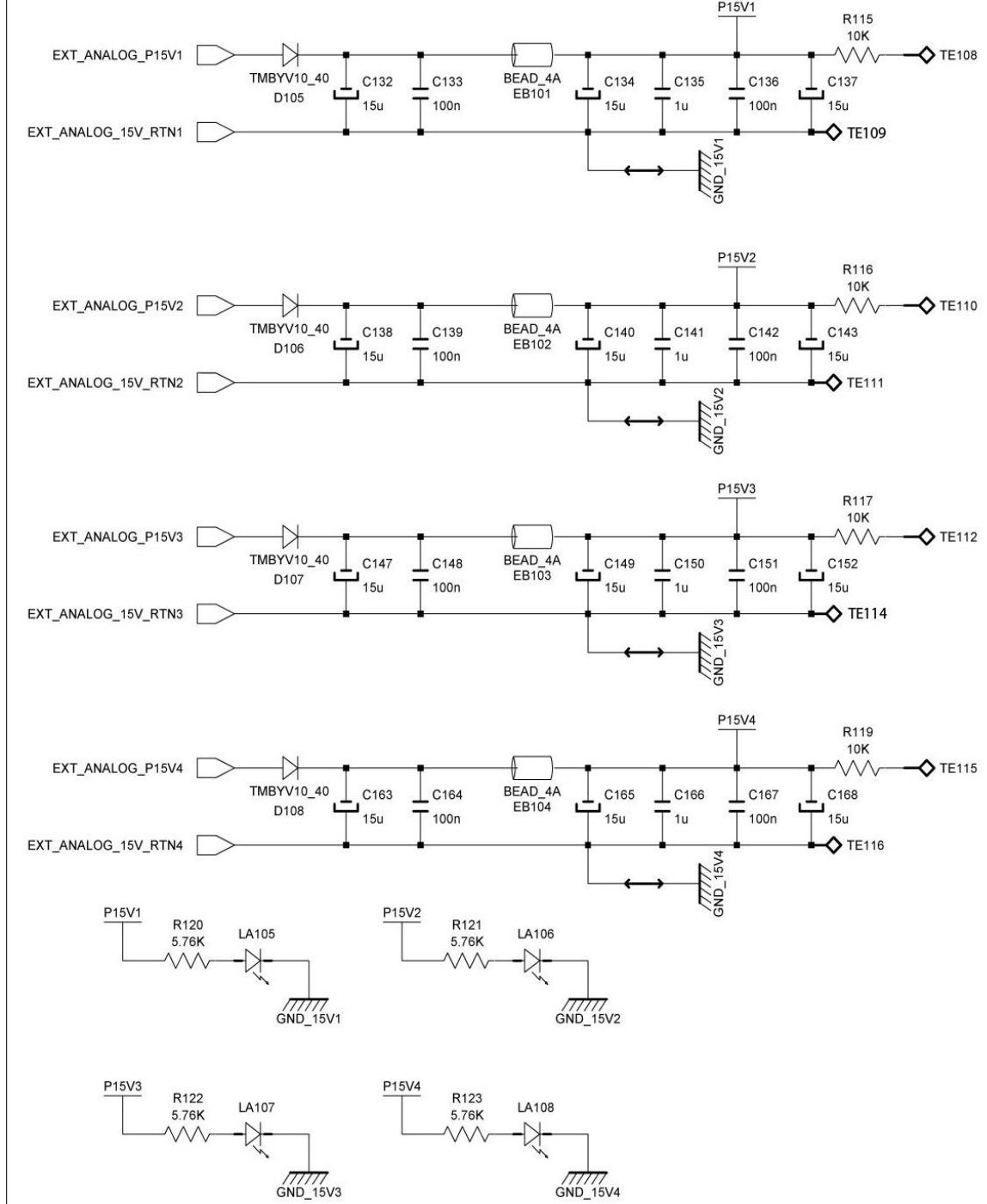


Şekil 48. AD5764 Entegre Devresi, Çevre Elemanları ve Çıkış Kanalları Tasarımı

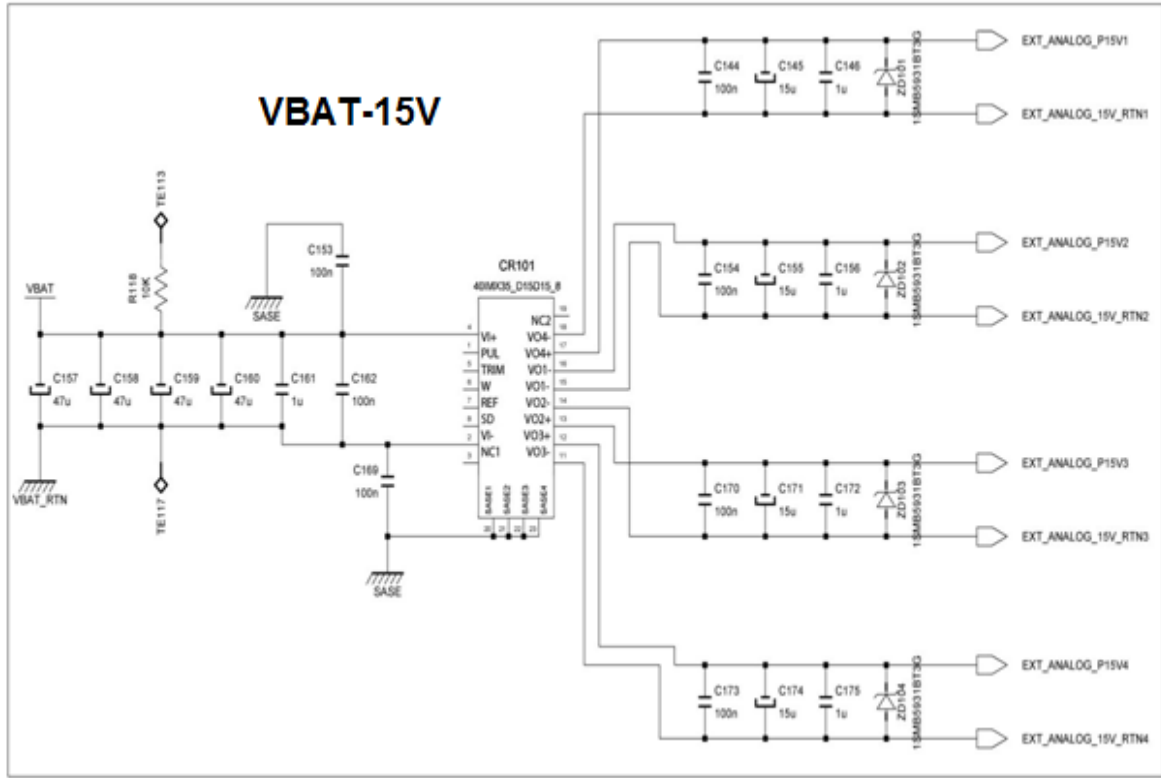


Şekil 49. Gerilim Çevirici Blokları Devre Tasarımları

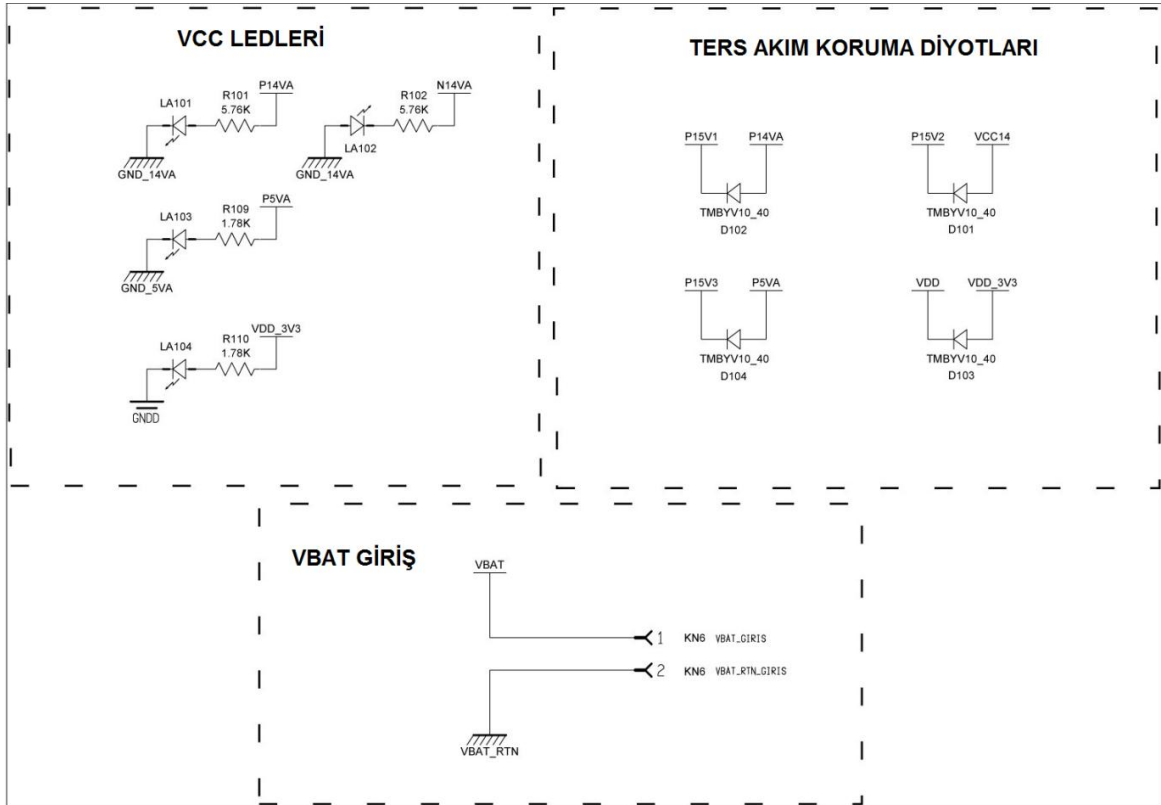
15V FİLTRE DEVRELERİ



Şekil 50. DC-DC Çevirici Çıkış Filtreleme Devreleri



Şekil 51. DC-DC Çevirici ve Çıkış Devre Elemanları



Şekil 52. Test Noktaları, Ledler, Ters Akım Koruma Diyotları ve Güç Giriş Konektörü

Ek 2

Terimler Sözlüğü

adres yolu	address bus
alıcı - verici	receiver - transmitter
ana eleman	host
baud hızı	baudrate
çevrim	cycle
çoklu zincir bağlama	daisy chain
çözünürlük	resolution
dalga formu	waveform
denklem	equation
doğrulama	validation
entegre devre	integrated circuit
faz birikeci	phase accumulator
frekans	frequency
genlik	amplitude
geri besleme	feedback
hız	rate
ikili aritmetik taban	binary

ikiye tümleyen	two's complement
ilkleme	initialize
ince kazanç	fine gain
kaba kazanç	coarse gain
kanal	port
kayan yazmaç	shift register
kırpma	truncation
kiplenim	modulation
konektör	connector
kukla modem	null modem
kullanıcı kılavuzu	user guide
mandal	latch
mantık	logic
nicemleme	quantization
nitelik	attribute
örnekleme	sampling
saat	clock
safılık	impurity
sayısal	digital
signal	şinyal - işaret
spektral	spectral

tasarım belirtilmeleri	design specifications
veri	data
veri sayfası	datasheet
veriyolu	bus
yazmaç	register
zaman tabanlı seçirme	time jitter

ÖZGEÇMİŞ

Kimlik Bilgileri

Adı Soyadı : Nazire Merve AYDOĞAN
Doğum Yeri : ANKARA
Medeni Hali : Bekar
E-posta : merve_aydogan@mynet.com
Adresi : Mehmet Akif Ersoy Mah. 296. Cad. NO:16, ASELSAN A.Ş
06370, Yenimahalle/ANKARA/TÜRKİYE

Eğitim

Lise : Hacı Ömer Tarman Anadolu Lisesi
Lisans : Hacettepe Üniversitesi

Yabancı Dil ve Düzeyi

İngilizce : Çok iyi düzeyde
Almanca : Orta düzeyde

İş Deneyimi

Aselsan A.Ş Elektronik Test Tasarım Mühendisliği (2010)

Deneyim Alanları

Gömülü yazılımlar, network veri iletişimi, otomatize test sistemleri

Tezden Üretilmiş Projeler ve Bütçesi

-

Tezden Üretilmiş Yayınlar

-

Tezden Üretilmiş Tebliğ ve/veya Poster Sunumu ile Katıldığı Toplantılar

-

