



**T.C.**  
**SELÇUK ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**



**SIKIŞTIRILMIŞ ROM TABANLI DOĞRUDAN**  
**SAYISAL FREKANS SENTEZLEYİCİ**  
**TASARIMI VE GERÇEKLEŞTİRİLMESİ**

**Yunus Emre ACAR**

**YÜKSEK LİSANS TEZİ**

**Elektrik-Elektronik Mühendisliği Anabilim Dalı**

**Ağustos-2017**  
**KONYA**  
**Her Hakkı Saklıdır**

## TEZ KABUL VE ONAYI

Yunus Emre ACAR tarafından hazırlanan “*Sıkıştırılmış Rom Tabanlı Doğrudan Sayısal Frekans Sentezleyici Tasarımı ve Gerçekleştirilmesi*” adlı tez çalışması 22/08/2017 tarihinde aşağıdaki jüri tarafından oy birliği / ~~oy çokluğu~~ ile Selçuk Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

### Jüri Üyeleri

#### Başkan

Prof. Dr. Mehmet ÇUNKAŞ

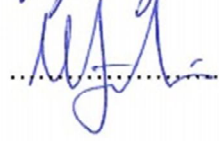
#### Danışman

Doç. Dr. Ercan YALDIZ

#### Üye

Yrd. Doç. Dr. M. Fahri ÜNLERŞEN

### İmza



Yukarıdaki sonucu onaylarım.

Prof. Dr. Mustafa YILMAZ  
FBE Müdürü

Bu tez çalışması ÖYP tarafından 2015-ÖYP 004 nolu proje ile desteklenmiştir.

## **TEZ BİLDİRİMİ**

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

## **DECLARATION PAGE**

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.



Yunus Emre ACAR

01.08.2017

## ÖZET

### YÜKSEK LİSANS TEZİ

#### SIKIŞTIRILMIŞ ROM TABANLI DOĞRUDAN SAYISAL FREKANS SENTEZLEYİCİ TASARIMI VE GERÇEKLEŞTİRİLMESİ

Yunus Emre ACAR

Selçuk Üniversitesi Fen Bilimleri Enstitüsü  
Elektrik-Elektronik Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Ercan YALDIZ

2017, 84 Sayfa

Jüri

Doç. Dr. Ercan YALDIZ

Prof. Dr. Mehmet ÇUNKAS

Yrd. Doç. Dr. M. Fahri ÜNLERŞEN

Özel bir frekansa ihtiyaç duyan sistemlerde, hedeflenen frekansın üretilmesi için çeşitli yöntemler geliştirilmiştir. Günümüzde çok özel frekanslara ihtiyaç duyulması, daha kararlı ve hızlı frekans geçişlerine ihtiyacı bulunan sistemlerin artması ile birlikte analog çözümler yetersiz kalmaktadır. Doğrudan analog frekans sentezleyicilerine ve faz kilit döngülü sentezleyicilere kıyasla daha hızlı frekans geçişlerine imkân veren, dinamik performansı yüksek, daha az güç tüketen ve maliyeti düşük olan doğrudan sayısal frekans sentezleyicileri bu tez çalışmasına konu olmuştur.

Bu tez çalışmasında, iki parçalı tablo yöntemi (BTM) kullanılarak ROM boyutu küçültülmüş bir doğrudan sayısal frekans sentezleyici (DDFS) tasarlanması, gerçekleştirilmesi ve tasarımın performansını belirleyen unsurların incelenmesi hedeflenmiştir. Farklı özelliklerde gerçekleştirilen 4 farklı tasarım FPGA üzerinde VHDL dili ile tasarlanmıştır. Gerçekleştirilen tasarımlarda ROM boyutu 208 kata kadar küçültülmüş, 71.17 dBc SFDR elde edilmiştir. Tasarımlarda sabit 32 bit ve 18-32 bit arasında değiştirilebilir faz çözünürlüğü kullanılmıştır. 10 ya da 16 bit genlik çözünürlüklü olarak tasarlanan yapılarda 14 ya da 20 bitlik faz kırpması gerçekleştirilmiştir. Faz kırpması, örnekleme frekansı, faz çözünürlüğü, ROM boyutu ve kullanılan alçak geçiren filtrelerin harcanan güç, spektral saflık ve maksimum çıkış frekansını nasıl etkilediği incelenmiştir.

**Anahtar Kelimeler:** DDFS, DDS, BTM, Frekans sentezleyici, VHDL, FPGA

## **ABSTRACT**

## **MS THESIS**

# **DESIGN AND IMPLEMENTATION OF COMPRESSED ROM BASED DIRECT DIGITAL FREQUENCY SYNTHESIZER**

**Yunus Emre ACAR**

**THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF  
SELÇUK UNIVERSITY  
THE DEGREE OF MASTER OF SCIENCE  
IN ELECTRICAL-ELECTRONICS ENGINEERING**

**Advisor: Assoc. Prof. Dr. Ercan YALDIZ**

**2017, 84 Pages**

**Jury**

**Assoc. Prof. Dr. Ercan YALDIZ**

**Prof. Dr. Mehmet ÇUNKAŞ**

**Asst. Prof. Dr. M. Fahri ÜNLERŞEN**

Various methods have been developed to produce the targeted frequency for the systems that require a specific frequency. Today, analogue solutions have become inadequate for the need for very specific frequencies, stable and fast frequency transitions. Direct digital frequency synthesizers, which have higher dynamic performance, lower power consumption and lower cost, which allow faster frequency transitions than direct analog frequency synthesizers and PLL synthesizers, have been the subject of this thesis study.

In this thesis study, it was aimed to design and implement a Direct Digital Frequency Synthesis (DDFS) with a reduced ROM size using Bipartite Table Method (BTM) and to examine the elements determining the performance of the design. Four different designs implemented with different features are designed with VHDL language on FPGA. In the implemented designs the ROM size was reduced to up to 208 times and 71.17 dBc SFDR level was obtained. Fixed 32 bit and 18-32 bit interchangeable phase resolution is used in these designs. 14 or 20 bit phase truncation has been performed in the designed structures which have amplitude resolution of 10 or 16 bits. The effects of phase truncation, sampling frequency, phase resolution, ROM size and low pass filters on power dissipation, spectral purity and maximum output frequency was examined.

**Keywords:** DDFS, DDS, BTM, Frequency Synthesizers, VHDL, FPGA

## ÖNSÖZ

Bu tez çalışmasında bilgi, tecrübe ve yardımlarını esirgemeyen danışmanım Sayın Doç. Dr. Ercan YALDIZ'a teşekkürlerimi sunarım. Ayrıca, verdikleri destekten dolayı değerli araştırma görevlisi arkadaşlarıma teşekkür ederim.

Son olarak, bana duydukları sonsuz güven ve gösterdikleri sabır için aileme teşekkür ederim.

Yunus Emre ACAR  
KONYA-2017

# İÇİNDEKİLER

|                                                                  |            |
|------------------------------------------------------------------|------------|
| <b>ÖZET .....</b>                                                | <b>iv</b>  |
| <b>ABSTRACT.....</b>                                             | <b>v</b>   |
| <b>ÖNSÖZ .....</b>                                               | <b>vi</b>  |
| <b>İÇİNDEKİLER .....</b>                                         | <b>vii</b> |
| <b>SİMGELER VE KISALTMALAR.....</b>                              | <b>ix</b>  |
| <b>1. GİRİŞ.....</b>                                             | <b>1</b>   |
| 1.1. Frekans Sentezleyici .....                                  | 2          |
| 1.1.1. Doğrudan analog frekans sentezleyicileri .....            | 3          |
| 1.1.2. Faz kilit döngülü (dolaylı) frekans sentezleyiciler ..... | 4          |
| 1.1.3. Doğrudan sayısal frekans sentezleyici (DDFS).....         | 6          |
| 1.2. Kaynak Araştırması .....                                    | 7          |
| <b>2. DDFS ÖZELLİKLERİ.....</b>                                  | <b>11</b>  |
| 2.1. Çözünürlük ve Frekans Aralığı .....                         | 12         |
| 2.2. Çıkış Frekansının Spektral Saflığı.....                     | 12         |
| 2.2.1. Parazitsiz dinamik aralık (SFDR).....                     | 13         |
| 2.3. Çıkış Frekansı.....                                         | 17         |
| 2.4. Anahtarlama Hızı.....                                       | 18         |
| 2.5. Harcanan Güç ve ROM Boyutu .....                            | 19         |
| <b>3. MATERYAL .....</b>                                         | <b>21</b>  |
| 3.1. DAC5687EVM Geliştirme Kartı.....                            | 21         |
| 3.1.1. DAC5687 için saat sinyali sağlama yöntemleri .....        | 22         |
| 3.1.2. DAC5687EVM çıkış seçenekleri.....                         | 23         |
| 3.2. Nexys4 DDR™ FPGA Kartı.....                                 | 23         |
| 3.2.1. Nexys4 DDRTM saat sinyali.....                            | 24         |
| 3.2.2. Nexys4 DDRTM giriş ve çıkışları .....                     | 25         |
| 3.2.3. Nexys4 DDRTM USB-UART köprüsü .....                       | 25         |
| 3.3. Vivado 2016.3 yazılımı.....                                 | 25         |
| <b>4. YÖNTEM.....</b>                                            | <b>26</b>  |
| 4.1. Faz Akümülatörü .....                                       | 26         |
| 4.2. Faz-Genlik Dönüştürücü .....                                | 28         |
| 4.2.1. ROM sıkıştırma yöntemleri.....                            | 28         |
| 4.3. Sayısal-Analog Dönüştürücü.....                             | 32         |
| 4.3.1. Sayısal-analog dönüştürücü parametreleri .....            | 33         |

|                                                                                  |           |
|----------------------------------------------------------------------------------|-----------|
| <b>5. GERÇEKLENEN DDFS TASARIMLARI .....</b>                                     | <b>36</b> |
| 5.1.1. Tasarım 1 .....                                                           | 36        |
| 5.1.2. Tasarım 2 .....                                                           | 37        |
| 5.1.3. Tasarım 3 .....                                                           | 37        |
| 5.1.4. Tasarım 4 .....                                                           | 39        |
| 5.1.5. UART alıcı modülü ve MATLAB GUI ile DDFS arayüzü .....                    | 49        |
| 5.1.6. Sinüs, kare, testere dişi ve üçgen sinyal formlarının elde edilmesi ..... | 53        |
| 5.1.7. Sistem saat sinyali kullanımı .....                                       | 61        |
| <b>6. SONUÇLAR VE GERÇEKLENEN DDFS'NİN ÖZELLİKLERİ.....</b>                      | <b>63</b> |
| 6.1. Çözünürlük ve Frekans Aralığı .....                                         | 63        |
| 6.2. Elde Edilen Sinüzoidal Sinyalin Spektral Saflığı.....                       | 67        |
| 6.2.1. Faz kırpmasının spektral saflığa etkisi.....                              | 70        |
| 6.2.2. Sayısal-analog dönüştürücü ve filtrenin spektral saflığa etkisi .....     | 75        |
| 6.3. Harcanan güç ve ROM boyutu.....                                             | 78        |
| <b>7. SONUÇLAR VE ÖNERİLER .....</b>                                             | <b>79</b> |
| 7.1. Sonuçlar .....                                                              | 79        |
| 7.2. Öneriler .....                                                              | 80        |
| <b>KAYNAKLAR .....</b>                                                           | <b>82</b> |
| <b>ÖZGEÇMİŞ.....</b>                                                             | <b>84</b> |

## SİMGELER VE KISALTMALAR

### Simgeler

|                    |                                                                     |
|--------------------|---------------------------------------------------------------------|
| $a$                | : İlk değer tablosunu adresleyen bit sayısı                         |
| $b$                | : Faz sözcüğünün offset tablosunu adresleyen en önemsiz bit sayısı  |
| $BG$               | : Bant genişliği                                                    |
| $c$                | : Faz sözcüğünün offset tablosunu adresleyen en önemli bit sayısı   |
| $F_{\text{çıkış}}$ | : DDFS çıkış frekansı                                               |
| $f_{dac}$          | : Nyquist örnekleme frekansı                                        |
| $F_{\text{giriş}}$ | : DDFS giriş frekansı                                               |
| $f_G$              | : Geribesleme frekansı                                              |
| $f_k$              | : Kristal frekansı                                                  |
| $f_o$              | : PLL sentezleyici çıkış frekansı                                   |
| $f_{os}$           | : Artırılmış örnekleme frekansı                                     |
| $f_r$              | : Referans frekansı                                                 |
| $f_s$              | : Örnekleme frekansı                                                |
| $f_x$              | : Karıştırıcı frekansı                                              |
| $G$                | : Geribesleme yolu bölücü oranı                                     |
| $K$                | : İnterpolasyon için aynı eğim bilgisini kullanacak komşu sayısı    |
| $M_i$              | : İnterpolasyon için komşu parçaların kullandığı ortak eğim bilgisi |
| $N$                | : Faz akümülatörü çözünürlüğü                                       |
| $N_b$              | : Sistem arka plan gürültüsü                                        |
| $N_i$              | : Giriş gürültü seviyesi                                            |
| $N_s$              | : Sistemin gürültü seviyesi                                         |
| $P$                | : Ön bölücü oranı                                                   |
| $P_g$              | : Gürültü gücü                                                      |
| $P_s$              | : Sinyal gücü                                                       |
| $Q$                | : Kırpılmış faz sözcüğü çözünürlüğü                                 |
| $R$                | : DAC çözünürlüğü                                                   |
| $S$                | : Seri haberleşme ile alınan toplam bit sayısı                      |
| $W_{TIV}$          | : İlk değer tablosu genişliği                                       |
| $W_{TO}$           | : Ofset tablosu genişliği                                           |
| $\Omega_m$         | : Sinyal bant genişliği                                             |
| $\Omega_s$         | : Örnekleme frekansı                                                |

### Kısaltmalar

|        |                                                                                                                    |
|--------|--------------------------------------------------------------------------------------------------------------------|
| BTM    | : İki parçalı tablo yöntemi ( <i>Bipartite Table Method</i> )                                                      |
| CORDIC | : Eksen döndürme sayısal bilgisayarı ( <i>C</i> oordinate <i>R</i> otation <i>D</i> igital <i>C</i> omputer )      |
| CMOS   | : Tamamlayıcı metal oksit yarı iletken ( <i>C</i> omplementary <i>M</i> etal <i>O</i> xide <i>S</i> emiconductor ) |
| DAC    | : Sayısal analog dönüştürücü ( <i>D</i> igital <i>A</i> nalog <i>C</i> onverter )                                  |
| DDFS   | : Doğrudan sayısal frekans sentezleyici ( <i>D</i> irect <i>D</i> igital <i>F</i> requency <i>S</i> ynthesizer )   |

|      |                                                                                                  |
|------|--------------------------------------------------------------------------------------------------|
| DDS  | : Doğrudan sayısal sentez ( <b><i>Direct Digital Synthesis</i></b> )                             |
| FPGA | : Sahada programlanabilir kapı dizisi ( <b><i>Field Programmable Gate Array</i></b> )            |
| FIR  | : Sonlu dürtü cevabı ( <b><i>Finite Impuls Response</i></b> )                                    |
| FCW  | : Frekans denetim sayısal sözcüğü ( <b><i>Frequency Control Word</i></b> )                       |
| LUT  | : Başvuru çizelgesi ( <b><i>Look Up Table</i></b> )                                              |
| LVDS | : Düşük voltajlı diferansiyel sinyalizasyon ( <b><i>Low Voltage Differential Signaling</i></b> ) |
| MDS  | : Tespit edilebilen minimum sinyal ( <b><i>Minimum Detectable Signal</i></b> )                   |
| MTM  | : Çok parçalı tablo yöntemi ( <b><i>Multipartite Table Method</i></b> )                          |
| NCO  | : Sayısal kontrollü osilatör ( <b><i>Numerically Controlled Oscillator</i></b> )                 |
| PLL  | : Faz kilitli döngü ( <b><i>Phase Locked Loop</i></b> )                                          |
| ROM  | : Salt okunur bellek ( <b><i>Read Only Memory</i></b> )                                          |
| SFDR | : Parazitsiz dinamik aralık ( <b><i>Spurious Free Dynamic Range</i></b> )                        |
| SNR  | : Sinyal-gürültü oranı ( <b><i>Signal to Noise Ratio</i></b> )                                   |
| SQR  | : Sinyal-kuantalama hata oranı ( <b><i>Signal to Quantization Error Ratio</i></b> )              |
| TIV  | : İlk değer tablosu ( <b><i>Table of Initial Values</i></b> )                                    |
| TO   | : Ofset tablosu ( <b><i>Table of Offsets</i></b> )                                               |
| VCO  | : Gerilim kontrollü osilatör ( <b><i>Voltage Controlled Oscillator</i></b> )                     |
| UART | : Evrensel asenkron alıcı/verici ( <b><i>Universal Asynchronous Rec./ Trans.</i></b> )           |

## 1. GİRİŞ

Günümüz teknolojisinde, elektronik sistemlerdeki analog çözümler yerini hızlı bir şekilde sayısal olanlara bırakmaktadır. Modern elektronik sistemler, daha yüksek hızlarda ve daha yüksek çözünürlükte işlem yapmakta ve teknolojinin gelişim hızı düşünüldüğünde tasarım kolaylığı ve maliyet açısından da beraberinde bazı kısıtlamalar getirmektedir. Bu kısıtlamalar sayısal tasarımın kullanımını kaçınılmaz kılarken analog tasarımların daha az tercih edilmesine neden olmaktadır.

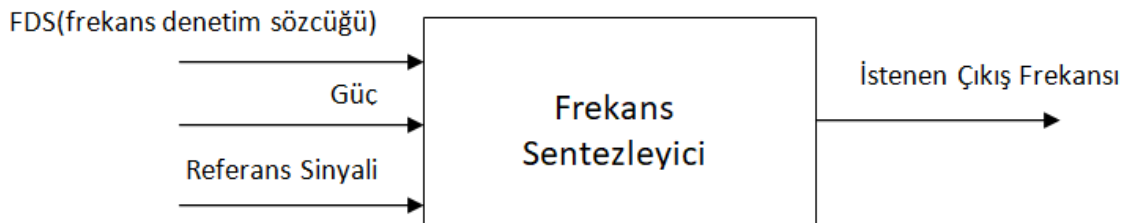
Geleneksel analog yapıların yerine tamamen sayısal çözümlerin kullanılmaya başlandığı yapılardan birisi de frekans sentezleyicileridir. Kablolu ve kablosuz haberleşme sistemleri, uzay ve uydu teknolojileri, savunma sanayi sistemleri, test ve ölçüm cihazları, medikal uygulamalar ve görüntü işleme gibi birçok alan yüksek çözünürlüklü, düşük güç tüketen, yüksek anahtarlama hızı ve yüksek bant genişliği sağlayan frekans sentezleyicilerine ihtiyaç duymaktadır (Madisetti ve ark., 1999). Frekans sentezlemek için kullanılan doğrudan analog sentezleyiciler ve klasik faz kilitleme döngülü sentezleyiciler tüm bu ihtiyaçları karşılamakta zamanla yetersiz kalmışlardır. Doğrudan sayısal frekans sentezleyicileri, bu ihtiyaçları karşılamanın yanı sıra, referans frekansı ve anahtarlama hızı gibi değişkenleri donanımsal bir yenilemeye ihtiyaç duymadan değiştirme imkânı vermesiyle tasarım süresi ve maliyetini de azaltmaktadır (Kester, 1999). Hızlı bir şekilde gelişen sayısal teknoloji ile birlikte doğrudan sayısal frekans sentezleyiciler diğer sentezleyiciler arasından öne çıkmış ve hızlı bir şekilde gelişirken geniş bir uygulama alanına da yayılmıştır. Doğrudan sayısal sentezleyicilerin hızını sınırlandıran sayısal-analog dönüştürücülerin de günümüzde yüksek hızlara ulaşmasıyla yüksek frekans seviyelerinde frekans sentezlemek doğrudan sayısal frekans sentezleyiciler ile mümkün kılınmıştır.

Doğrudan sayısal frekans sentezleyici tasarımında yaygın olarak FPGA kullanılmaktadır. Sayısal frekans sentezleyicinin yüksek çözünürlük ve hız gereksinimlerini karşılamak için gerekli mantıksal işlemleri yüksek sayıda bitle paralel olarak kolayca yapabilmesi açısından programlanabilir mantık cihazları klasik mikrodenetleyicilere üstünlük sağlamaktadır. Programlanabilir mantık cihazları içinde ise FPGA, çok sayıda mantık bloğu içermesi, bloklar arası bağlantıların kolay olması, tasarıma özgü olarak sahada programlanabilmesi nedeniyle öne çıkmaktadır.

### 1.1. Frekans Sentezleyici

Frekans sentezleyici, bir referans frekanslı sinyal girişini bir denetim yöntemi ile kumanda ederek bir ya da daha fazla yeni frekanslı sinyaller üreten elektronik aygıtlardır (Tierney ve ark., 1971).

Günümüzdeki gelişmiş frekans sentezleyicilerinden önce, özel bir frekansa gerek duyan sistemlerde, klasik tank devreleri referans frekansı üretmek için kullanılmış ve frekans ayarlama alıcı tarafındaki bir değişken kondansatör ile yapılmıştır. Bu sistemlerde referans frekansının çok kararlı olmaması, kristal osilatörlerin geliştirilip kullanılmasına kadar geçen sürede, frekans ayarlamasında önemli bir sorun oluşturmuştur. Kristal osilatörlerin referans frekans kaynağı olarak kullanılmasıyla kararlılık sorunu aşılmış, çok sayıda kristal osilatör ile frekans ayarlama işlemi sağlanmıştır. Çok sayıda kristal osilatör kullanımının yüksek maliyeti ve çok özel frekanslara duyulan ihtiyaçların da artmasından dolayı, tek bir referans frekans kaynağı ile istenilen frekansı sentezleyebilmek, gelişen teknoloji ile birlikte en önemli ihtiyaçlardan olmuştur. Frekans sentezleyici blok şeması basit olarak Şekil 1.1’de verilmiştir.



**Şekil 1.1.** Frekans sentezleyici blok şeması

Frekans sentezleyicileri, sayısal ve analog teknolojilerin her ikisini kullanan sistemlerdir. Bir frekans sentezleyici tasarlamak için osilatörler, sayısal kontrollü osilatörler, yükselteçler, filtreler, faz detektörleri, mantık, düşük gürültülü dc yükseltme ve filtreme gibi birçok disiplin kullanılmaktadır (Goldberg, 1999). Askeri radyolarda, uydu haberleşme terminallerinde ve radar sistemlerinde yaygın olarak kullanılan frekans sentezleyicilerindeki gelişmeler, cep telefonları, kablosuz haberleşme ve kişisel haberleşme servislerinin gelişimine de doğrudan etki etmektedir. Frekanslar arasındaki geçişin hızlı olması, parazitsiz dinamik aralık, harcanan güç ve alan, yapılan akademik

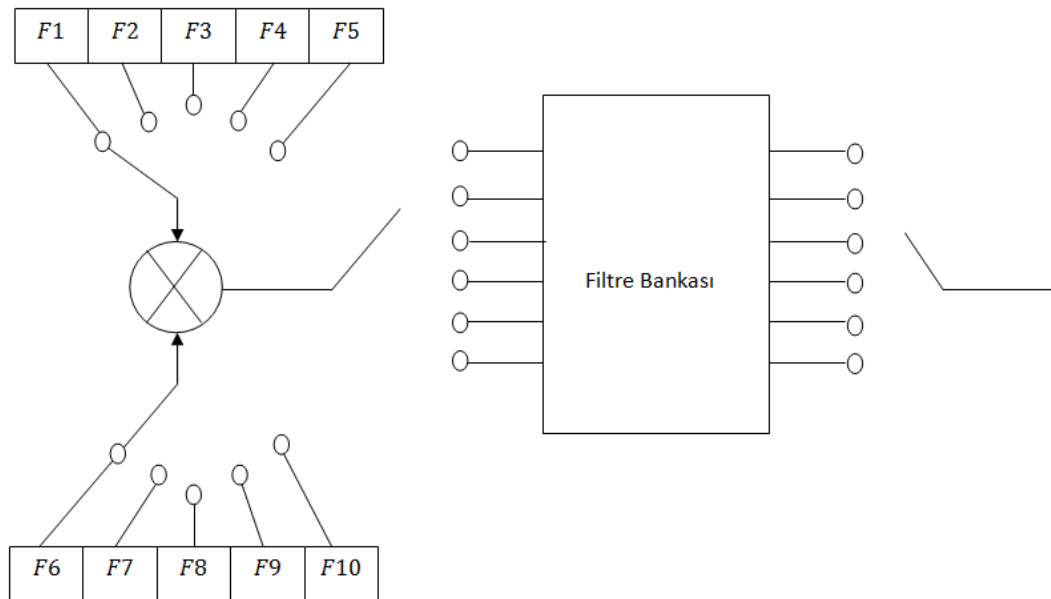
ve endüstriyel çalışmalarla frekans sentezleyicilerinde geliştirilmek istenen temel parametrelerdir.

Frekans sentezleyicileri, faz kilit döngülü (PLL ya da dolaylı), doğrudan analog (karıştırma/filtreleme/bölme) ve doğrudan sayısal olmak üzere üç ana gruba ayrılmaktadır.

### 1.1.1. Doğrudan analog frekans sentezleyicileri

Doğrudan analog frekans sentezleyicileri, frekans sentezleyicileri arasında en eski olanlardır. Referans frekans kaynakları, karıştırıcı, filtre ve bölücüler, doğrudan analog sentezleyicilerin bloklarını oluşturmaktadır. Şekil 1.2’de doğrudan analog sentezleyicinin bir basamağı verilmiştir. Şekilde verilen yapının ardından bölücü kullanılarak bir sonraki bloğun girişi oluşturulabilir. Birbirini takip eden basamaklar ile yüksek çözünürlük elde edilebilmektedir.

Doğrudan analog frekans sentezleyicileri, istenen yeni frekansı elde etmek için çarpma, bölme ve diğer matematiksel işlemleri kullanırlar. Bu sentezleyicinin “doğrudan” olarak adlandırılmasının sebebi hata düzeltme işleminden kaçınılması, dolayısıyla çıkış frekansının doğrudan giriş frekansı ile ilişkili olmasıdır. Doğrudan olması sebebiyle anahtarlama hızı çok yüksek olabilmekte ve faz gürültüsü çok düşük seviyelere inmektedir (Anonymous).



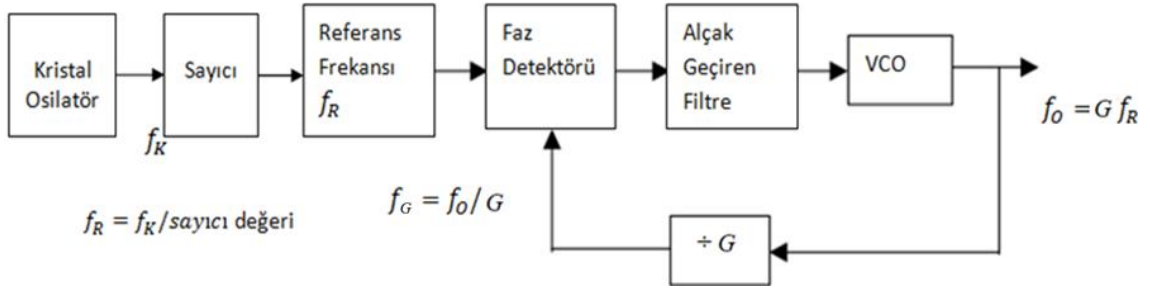
Şekil 1.2. Doğrudan analog frekans sentezleyici

Referans frekans kaynağı olarak, yüksek kararlılığa sahip kristal osilatörün doğrudan kullanılmasından dolayı, doğrudan analog sentezleyicileri ile sentezlenen sinyaller spektral olarak daha saftır. Ayrıca hata düzeltme işleminden kaçınılması nedeniyle yüksek anahtarlama hızına ulaşılabilmesi mümkündür. Bu avantajların yanında doğrudan analog frekans sentezleyicilerinin bazı olumsuz yanları da bulunmaktadır. Örneğin, genişbant uygulamalar ele alındığında, doğrudan analog frekans sentezleyicilerinin çok sayıda referans frekans ihtiyacı bulunmaktadır ve bu da karmaşıklığı ve maliyeti artırmaktadır. Frekans bölme yapılarak referans frekans ihtiyacı düşürüldüğünde ise faz belirsizliği ortaya çıkmaktadır (Goldberg, 1999). Tüm bu olumlu ve olumsuz özelliklerle birlikte, sayısal teknolojinin hızlı bir şekilde gelişmesiyle, karmaşık analog tasarımların yerini sayısal olanlar, doğrudan analog sentezleyicilerin yerini ise doğrudan sayısal sentezleyiciler almaktadır.

### 1.1.2. Faz kilit döngülü (dolaylı) frekans sentezleyiciler

Faz kilit döngülü frekans sentezleyiciler, belli bir referans frekansını gerilim kontrollü osilatör ile karşılaştırmaktadır. Çıkışta bir sapma olduğunda, tespit edilen hataya bağlı olarak gerilim kontrollü osilatörü kontrol edecek düzeltme komutu oluşturulur. Bu aşamadan dolayı faz kilit döngülü bir sentezleyicide hızlı anahtarlama seviyelerine çıkılması zordur. Basit ve ekonomik yapısından dolayı faz kilitlemeli frekans sentezleyicileri en çok kullanılan sentezleyicilerdendir. Karmaşık radar sistemlerinden uydu haberleşme terminallerine, araç radyolarından sinema sistemlerine kadar birçok uygulamada faz kilit döngülü frekans sentezleyiciler kullanılmaktadır.

Klasik bir faz kitleme döngülü frekans sentezleyici blok şeması Şekil 1.3'te verilmiştir. Şekilde verilen blok şemada, hata detektörü kısmında negatif geribesleme sinyali ile referans sinyali arasındaki farkla orantılı bir gerilim üretilir. Bu gerilim, filtrelendikten sonra VCO (gerilim kontrollü osilatör) girişine uygulanır. Çıkışta  $f_R$  referans frekansının G katı olan  $f_O$  frekansı elde edilir. Frekans sentezleyicilerinde, kararlı yapısından dolayı referans frekansı olarak genellikle kristal osilatörler kullanılır.



Şekil 1.3. Faz kilitleme döngülü frekans sentezleyici

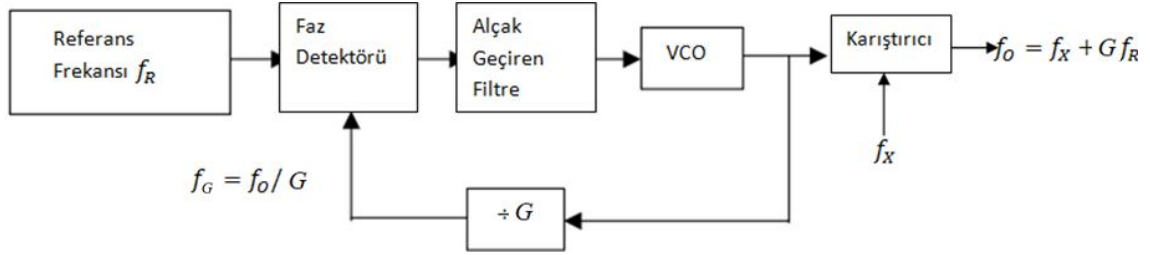
Şekil 1.3'te verilen faz kilitleme döngülü frekans sentezleyici çıkış frekansını, girişte kullanılan sayıcılar sınırlandırmaktadır. Bu nedenle VCO çıkışına frekans çarpıcı ilave edilerek daha yüksek frekanslara çıkılabilmektedir. Frekans çarpıcı kullanılarak gerçekleştirilen faz kilitleme döngülü frekans sentezleyicilerinde de, gerilim kontrollü osilatördeki ufak değişimler çıkışta çok büyük değişimlere sebep olmaktadır. Bu durumu ortadan kaldırmak için, çıkışta bir frekans çarpıcı kullanmak yerine negatif geribesleme kısmında,  $1/G$  bölücüsünden önce bir ön bölücü kullanılır. Bu ön bölücü sayesinde çıkış frekansı çok yüksek frekanslara çıkabilmektedir.  $1/P$  oranında ön bölme gerçekleştiren ön bölücü içeren faz kilitleme döngülü sentezleyicilerin,  $f_O$  çıkış frekansı ve  $f_G$  geri besleme frekansına ait eşitlikler Eşitlik (1.1), Eşitlik (1.2) ve Eşitlik (1.3)'te verilmiştir.

$$f_R = \frac{f_K}{\text{sayıcı değeri}} \quad (1.1)$$

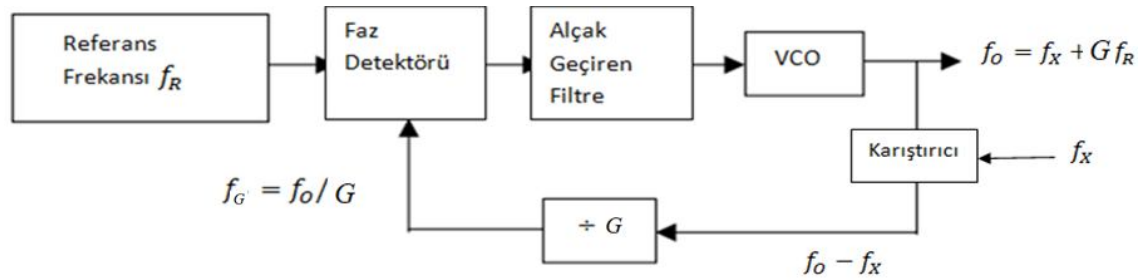
$$f_O = G \times P \times f_R \quad (1.2)$$

$$f_G = \frac{f_O}{G \times P} \quad (1.3)$$

Faz kilitlemeli frekans sentezleyicilerinde frekans artırıp azaltmak için ön bölücü ya da çarpıcı yerine karıştırıcı kullanılabilir. Kullanılacak ilave bir osilatör ve karıştırıcı ile de çıkış frekansı artırılabilir. Kullanılan ilave osilatörün ve karıştırıcının yerleşimine göre iki farklı yapı bulunmaktadır. Şekil 1.4'te verilen yapıda osilatör faz kilitleme döngüsü dışında kaldığı için osilatör kaynaklı hataların düzeltilmesi mümkün değildir. Şekil 1.5'te verilen yapıda ise osilatör kaynaklı hatalar da faz kilitleme sırasında bertaraf edilir.



Şekil 1.4. Çıkışında karıştırıcı içeren faz kilitleme döngülü frekans sentezleyici



Şekil 1.5. Faz kilitleme döngüsünde karıştırıcı içeren frekans sentezleyici

### 1.1.3. Doğrudan sayısal frekans sentezleyici (DDFS)

Doğrudan Sayısal frekans sentezleyicilerin temeli 1970'li yıllara dayanmaktadır. Sayısal elektronik teknolojisindeki hızlı gelişmelere paralel olarak günümüzde en hızlı gelişen frekans sentezleme tekniğidir. Sayısal frekans sentezleyicilerin, mantıksal yapıları ve hafıza birimleri kullanarak istenen çıkış sinyalini sayısal olarak üretir. Sayısal-analog dönüştürücüler ile de analog çıkış sinyali üretilir.

Doğrudan sayısal frekans sentezleyicilerini, doğrudan analog ve faz kilit döngülü sentezleyicilerden ayıran en önemli özellik tüm sentezleme işleminin sayısal olarak yapılması ve sayısal-analog dönüştürücü ile analog çıkış elde edilmesidir. Doğrudan sayısal frekans sentezleyiciyi diğer sentezleyicilerden ayıran özellikleri ve bazı üstünlükleri aşağıdaki gibidir (Kester, 1999).

- Doğrudan sayısal frekans sentezleyicilerde diğer sentezleyicilerin aksine osilatör kullanılmaz. Bunun yerine frekans ayarlama sayısal sözcüğü ve faz akümülatörü ile faz-genlik dönüştürücü kullanılır.
- Doğrudan sayısal frekans sentezleyicileri, PLL yapıların aksine hata düzeltme geri dönüş yoluna ihtiyaç duymazlar ve bu sayede daha yüksek hızlarda frekans sentezleyebilirler.

- Tamamen sayısal kontrollü, mikro-hertz düzeylerinde çıkış frekansı uyumlama çözünürlüğü sağlamaktadır.
- Çıkış frekans ya da faz uyumlamasında çok hızlı frekans değişimi ve sürekli fazlı frekans değişimi sağlamaktadır.
- Sayısal frekans sentezleyici yapısı, analog frekans sentezleyicilerindeki elemanların yılanması ve sıcaklık sapmaları kaynaklı manuel ince ayar yapma ve uyumlama ihtiyaçlarını ortadan kaldırmaktadır.

Gelişen sayısal elektronik teknolojisi ile birlikte sinyal üretiminde doğrudan sayısal sentezleyiciler geniş uygulama alanları bulmuş ve birçok uygulamada belirleyici bir rol üstlenmiştir. Empedans ölçümü, sensör uyartımı, sayısal modülasyon ve demodülasyon, test ve ölçüm cihazları, sıvı ve gaz akış ölçümleri, medikal cihazlar, kablosuz haberleşme ve uydu teknolojisi, geniş bant haberleşme, VHF/UHF lokal osilatör sentezi, radar, PSK/FSK modülasyonu, doğrudan sayısal sentez sistemlerinin uygulama alanlarındandır (Chimakurthy, 2005).

## 1.2. Kaynak Araştırması

Jinshan ve arkadaşları (2009), CORDIC tabanlı bir sayısal sinyal sentezleyici tasarımı gerçekleştirmiştir. Tasarımda 10 bit çözünürlüklü *current steering* tip DAC kullanılmıştır. 40-360 MHz arasındaki çıkış frekanslarına karşılık 57-46 dBc arasında SFDR değerleri elde edilmiştir.

Asok ve Sahoo (2014), tasarladıkları DDFS yapısının sayısal donanım analizi ve optimizasyonunu yaptıkları çalışmalarında, 0-500 MHz arasında 13 bit doğrulukla sinüs dalga formu elde etmişlerdir. 1.5 GHz'lik giriş sinyali kullanılarak gerçekleştirilen CORDIC tabanlı DDFS yapısında 1 mHz'lik frekans uyumlama çözünürlüğü elde edilmiştir. Hedeflenen SFDR değeri için sayısal donanım bit sayılarını elde etmek için analiz yapılmıştır. Donanım zamanlama kısıtları ve hedeflenen SFDR ile örtüşen değerler nihai tasarımda kullanılmıştır.

Menakadevi ve Madheswaran (2012), boru hattı veri yolu mimarisi tabanlı bir CORDIC tasarımı önermiştir. ROM kullanımı gerektirmeyen bu yöntemde, faz-genlik dönüşümünü daha hızlı gerçekleştirmek ve FPGA'de kullanılan alanı azaltmak amaçlanmıştır. 184.76 MHz'lik maksimum çalışma frekansı, 2.3 mHz'lik frekans çözünürlüğü ve 112 dB'lik SFDR değerleri elde edilmiştir.

Madisetti ve arkadaşları (1999) tarafından CORDIC yapısına benzer bir açı döndürme algoritması tabanlı bir DDFS yapısının anlatıldığı çalışmada, 100 dBc üzerinde 16 bit doğruluklu sinüs ve kosinüs dalga formları elde edilmiştir. 36 bitlik bir frekans ayarlama sözcüğü ile 100 MHz'lik örnekleme hızında 1.5 mHz'lik çözünürlük elde edilmiştir. Tasarlanan yapı, 14 mW/MHz güç tüketimi gerektirmektedir.

Yeoh ve arkadaşları (2010), 1.3 GHz'lik maksimum frekansta işlem yapabilen düşük güç tüketen bir DDFS yapısı ortaya koymuştur. Önerilen hibrit tasarımda, doğrusal ve doğrusal olmayan tipte sayısal-analog dönüştürücüler birlikte kullanılmıştır. 11 bitlik sayısal-analog dönüştürücü ile 1.3 GHz giriş saat sinyali ile 0 - Nyquist frekansı arasında 52 dBc üzerinde SFDR değerine ulaşılmıştır. 270  $\mu$ W/MHz'lik bir güç tüketimi sağlanmıştır.

De Caro ve arkadaşları (2008), yüksek performanslı bir DDFS gerçekleştirmek için Çok Parçalı Tablo Yöntemi (MTM) kullanılmasını araştırmıştır. Çok parçalı tablo yaklaşımında tek bir ofset tablosu kullanılması halinde, SFDR değeri için kapalı-form ifadeleri elde edilmiştir. Bu ifadeleri kullanarak hedef SFDR değeri için gereken hafıza boyutunun optimum olduğu tasarım analitik olarak ortaya konmuştur. Ofset tablo sayısının iki veya daha fazla olduğu durumlar için sayısal bir algoritma önerilmiştir. Elde edilen sonuçlarla ve önerilen diğer DDFS yapılarıyla yapılan karşılaştırmalarda, yüksek performanslı DDFS tasarımında, MTM'nin etkinliği kanıtlanmıştır.

Omran ve arkadaşları (2009), tamamen sayısal, darbe çıkışlı bir sayısal sinyal sentezleyici yapısı tanımlamıştır. İlave analog bir eleman kullanmadan tamamen sayısal olarak FPGA üzerinde gerçekleştirilen sentezleyicide, seçici aşırı örnekleme kullanılarak spektral performans iyileştirilmiştir. Gecikme hattı FPGA sayısal saat yöneticisi (DCM) ile oluşturulmuş,  $\mu$ s seviyelerinde anahtarlama hızına ulaşılmıştır. Deneysel ölçümlerle sistemin 40 dBc SFDR üzerinde çalıştığı doğrulanmıştır.

Strollo ve arkadaşları (2007), öne çıkan tablo sıkıştırma tekniklerinden olan çok parçalı tablo yöntemi tabanlı DDFS yapılarının detaylı bir tanımını yapmıştır. Hedef SFDR değeri için optimum MTM ayrıştırmasını bulan yeni bir algoritma sunulmuştur. Öngörülen tasarım, yüksek frekanslı saat sinyaliyle çalışabilmekte; yalnızca küçük bir başvuru çizelgesi (Look Up Table) ve basit toplayıcılar gerektirmektedir. 630 MHz maksimum çalışma frekansında 90 dBc SFDR değerine ulaşan tasarım, 76 mW güç tüketimi gerektirmektedir. Besleme gerilmi 2.5V 'dan 1.8V'a düşürüldüğünde ise 430 MHz maksimum çalışma frekansında ve 24.9 mW güç harcandığı görülmüştür.

Dinechin ve Tisserand (2005), tarafından mevcut tablo tabanlı yöntemler ve temel yaklaşımların sunulduğu çalışmada, ilave çarpıcı gerektiren yöntemlere kıyasla daha iyi hız/alan ödünleşmesi sunan çok parçalı tablo yöntemi (MTM) ortaya konmuştur. Bu yöntemin Virtex FPGA'ları üzerinde gerçekleştirilmesi ele alınmıştır.

Patel ve Sharma (2013), parçalı doğrusal yaklaşım kullanan optimize edilmiş bir DDFS yapısı ortaya koymuştur. Önerilen yöntem, zaman paylaşımı ile her saat sinyal periyodunda hafıza hücrelerinden ardışık okuma erişimine imkân sağlamaktadır. Çıkış değerleri geçici olarak kaydedilmekte ve daha sonra okunmaktadır. Sistem çıkışı, örneklerden elde edilmiş bir sinyaldir. Sistemde, sinyal formuna ait genlik bilgilerini içeren değerleri tutmak yerine sadece sinyal formunu oluşturmak için gerekli bazı katsayılar tutulmaktadır. Bu sayede kullanılan hafıza birimi boyutu düşürülmüştür. MATLAB'da yapılan analizler ile 84.2 dBc'lik SFDR değerine ulaşılmıştır.

Nicholas ve Samueli (1991), 150 MSPS örnekleme hızında 12 bit çözünürlükte bir sinüs sinyali sentezleyebilen bir DDFS yapısı sunmuştur. Sentezlenen sinüs sinyali 90.3 dBc üzerinde spektral saflığa sahiptir. 0-75 MHz arasında 35mHz adımlarla frekans sentezleyebilen sistemde anahtarlama hızı 6.7 ns ve uyumlama gecikmesi (tuning latency) 13 saat sinyali periyodudur. Başvuru çizelgesi (Look Up Table) boyutunu 128 kat düşürebilen etkili bir ROM sıkıştırma tekniği ortaya konmuştur. 100 MHz'de 950 mW güç tüketimi gerektiren bu yöntem, literatüre Nicholas yöntemi olarak geçmiştir.

Chimakurthy (2005) tarafında kablosuz uygulamalar için DDFS tasarımının yapıldığı yüksek lisans tez çalışmasında, sentezleyici çıkış spektrumunda gürültü analizi yapılmış, faz kırpması ve aşırı örnekleme etkileri vurgulanmıştır. Doğrusal olmayan bir ROM adresleme önerilmiştir. Simülasyonu MATLAB ile yapılmış sistem, Spartan 2 FPGA ile gerçekleştirilmiştir. Gerçeklenen sistemde, 25 MHz'lik saat sinyal girişi, 15 bitlik faz akümülatörü, 9 bitlik faz çözünürlüğü ve 8 bitlik genlik çözünürlüğü ile 763 Hz frekans çözünürlüğü elde edilmiştir. Ayrıca ROM boyutu yaklaşık 11 kat küçültülebilmektedir.

Kılıç (2006), yüksek lisans tez çalışmasında, sayısal sinyal sentezleyici yapısı, önemli parametreleri ve FPGA üzerinde gerçekleştirilmesi üzerinde durmuş, özellikle sistem parametreleri detaylıca incelemiştir. 60dBc SFDR değerinde 50-100 MHz arası sinyal sentezleyen bir sayısal sinyal sentezleyici tasarlanmıştır. Spartan XC3S200 üzerinde gerçekleştirilen tasarımda yüksek SFDR değerli AD9777 sayısal-analog dönüştürücüsü kullanılmıştır. Tasarlanan sistem farklı frekans ve faz modülasyonları

yapabilmektedir. Gerçeklenen sistem, modülasyon tipi ve taşıyıcı frekansı değişkenlerinin donanımsal bir değişiklik gerektirmeksizin programlanabilmesine imkân tanımaktadır.

Kazancıoğlu (2007), tarafından DDFS tabanlı bir fonksiyon jeneratörü tasarlanan yüksek lisans tez çalışmasında, sayısal donanım modelleme ve tasarımda SystemC ve VHDL dilleri kullanılmıştır. Sayısal tasarımın her aşamasında bu iki dilin avantajları ve uygulanabilirlik performansları karşılaştırılmıştır. Fonksiyon jeneratörü, I2C ve UART haberleşme protokollerini desteklerken sayısal uygulamalarda kullanıma hazır şekilde tasarlanmıştır. Yazılan VHDL kodu Virtex 4 serisi V4FX12 FPGA üzerine uygulanmıştır.

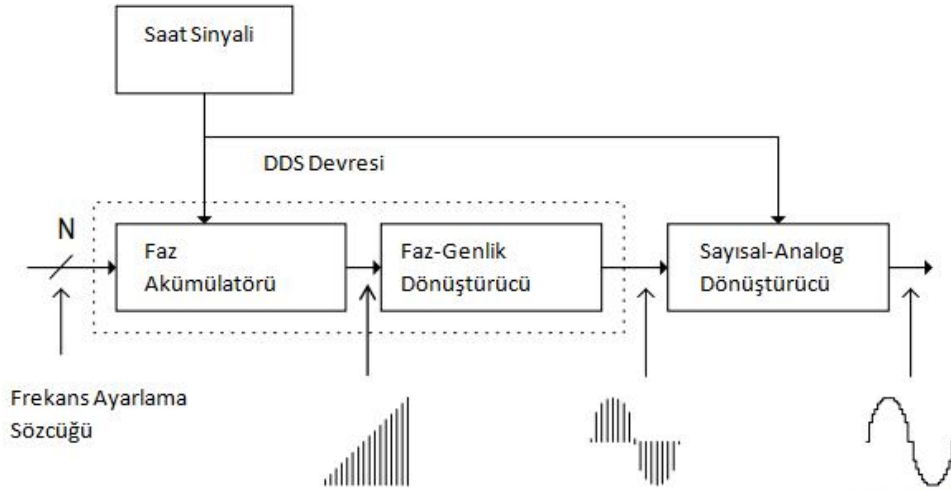
Aydoğan (2013) tarafından gerçekleştirilen tez kapsamında, yüksek doğruluklu lokal osilatörlere veya farklı dalga şekillerine ihtiyaç duyulan sistemlerde kullanılmak üzere, FPGA üzerinde DDS tabanlı fonksiyon üretici tasarlanmıştır. Genlik ve frekansı değiştirilebilen kosinüs, sinüs, kare dalga ve testere dişi dalga formları 0-171 kHz aralığında üretilmiştir. Kuadrant sıkıştırma tekniği kullanılarak ROM boyutu geleneksel yöntemlere göre 4 kat küçültülmüştür.

## 2. DDFS ÖZELLİKLERİ

Doğrudan sayısal frekans sentezleyicilerde örnekleme teoreminden faydalanılır. Bu teoreme göre bant genişliği  $\Omega_m$  olan sınırlı bantlı bir analog  $x(t)$  sinyalinin  $\Omega_s > 2\Omega_m$  olmak üzere örnekler alınır, alınan örneklerden  $x(t)$  analog sinyali bozulmadan oluşturulabilir. Doğrudan sayısal frekans sentezleyicilerde tam bir periyodu bilinen sinüs sinyalinin örnekleri oluşturulmakta ve oluşturulan bu örneklerden tam bir sinüs sinyali elde edilmektedir.

Şekil 2.1’de gösterildiği üzere geleneksel doğrudan sayısal frekans sentezleyicilerinde 3 ana kısım bulunmaktadır. Bu kısımlar faz akümülatörü, sinüs değer tablosu ve sayısal-analog dönüştürücü kısımlarıdır. Faz akümülatörü ile doğrusal olan faz değeri, bu değerinin zamana göre değişiminden de açısal hızı elde edilmektedir. Kullanılan sinüs değer tablosu ise faz ve açı bilgisinden genlik bilgisinin üretilmesini sağlamaktadır. Sinüs değer tablosu olarak ROM kullanılabileceği gibi iteratif hesaplamalarla da bu genlik bilgisinin elde edildiği yöntemler bulunmaktadır. Elde edilen genlik bilgisi ile oluşturulmak istenen analog sinyale ait faz, açı ve genlik bilgilerini içeren sayısal bilgi elde edilir. Sayısal-analog çevirici ile doğrudan sayısal frekans sentezleyicinin son aşaması tamamlanmış, sayısal sinyal üretilmiş olur. Buradan alınan analog çıkış filtrelenerek istenilen sinüs sinyali oluşturulur.

Tüm sistemlerde olduğu gibi sayısal frekans sentezleyicilerinde de bazı kısıtlamalar bulunmaktadır. Bu kısıtlamalardan en önemlileri parazitsiz dinamik aralık (SFDR), sistemin frekans atlama hızı ve sistemde harcanan güçtür. Bu kısıtlamaların yanında Şekil 2.1’de verilen elemanlardan kaynaklanan ilave birçok kısıtlama bulunmaktadır. Çözünürlük ve frekans aralığı, çıkış frekansının spektral saflığı, anahtarlama hızı, harcanan güç ve ROM boyutu gibi kavramlar bu bölümde anlatılacaktır.



Şekil 2.1. Frekans sentezleyici sinyal akışı (Kester, 1999)

## 2.1. Çözünürlük ve Frekans Aralığı

Frekans aralığı, frekans sentezleyici çıkışında elde edilen minimum frekans ile maksimum frekans arasındaki aralığı ifade etmektedir. Frekans çözünürlüğü ise çıkışta elde edilen iki frekans arasındaki en küçük farkı ifade etmektedir.

Bir sinüzoidal sinyal, sinüs değer tablolarından elde edilen örneklerinden oluşturulmak istendiğinde oluşturulan sinyalin doğruluğu faz ve genlik kuantalamasıyla doğrudan ilgilidir. Başka bir deyişle, kullanılan hafıza biriminin boyutu faz ve genlik çözünürlüğüne etki etmektedir. Bu çözünürlük sınırlandırması da zaman domeninde seçirme olarak kendini göstermekte ve spektruma geniş bant beyaz gürültü eklemektedir (Kılıç, 2006). Sistem saat sinyali ile birlikte faz akümülatörü de sentezleyicinin çözünürlüğünü belirleyen kısımlardandır. Belirlenen çözünürlük hedefi doğrultusunda birçok pratik uygulamada faz akümülatörü bit sayısı yüksek tutulmaktadır.

## 2.2. Çıkış Frekansının Spektral Saflığı

Spektral saflık, elde edilen çıkış sinyalinin spektrumunun ne kadar gürültüsüz olduğunu ifade etmektedir. Sayısal frekans sentezleyicinin spektral saflığı, kullanılan yöntem, faz ve genlik çözünürlüğüne, referans saat sinyaline, örnekleme frekansına, faz-genlik dönüşümündeki kırpma işlemine, kullanılan aritmetik devrelere ve sayısal-

analog dönüştürücünün çözünürlüğüne göre değişmektedir. Spektral saflık, parazitsiz dinamik aralık ile ölçülmektedir.

### 2.2.1. Parazitsiz dinamik aralık (SFDR)

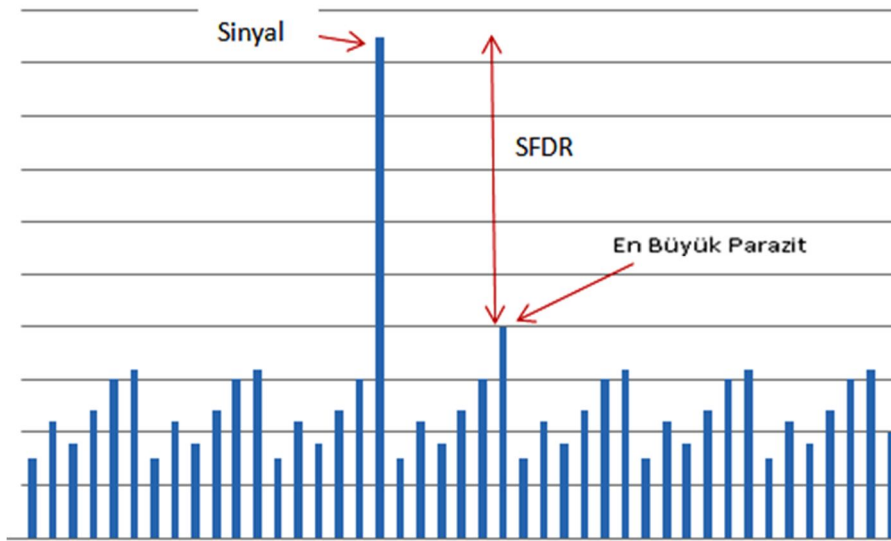
Frekans sentezleyicilerde amaç, tek bir referans frekanslı sinyal kaynağından hedeflenen yeni bir ya da daha fazla frekanslı sinyaller üretebilmektir. Parazitsiz dinamik aralık, tasarlanan frekans sentezleyicinin dinamik performansını belirtmekte kullanılan, üretilen sinüzoidal sinyalin saflığını ifade eden bir kavramdır. Hedeflenen frekanstaki sinyalin genliğinin, oluşan en büyük parazitin genliğine oranı olarak tanımlanmaktadır (Anonymous).

Sayısal frekans sentezleyici uygulamalarının büyük kısmında en önemli parametre olan parazitsiz dinamik aralığın, birbiri ile etkileşimli birçok unsura bağlı olmasından dolayı ölçülmesi, hesaplanması ya da analiz edilmesi kolay değildir.

Şekil 2.2’de görüldüğü üzere parazitsiz dinamik aralığın alt sınırını sistemdeki gürültüler belirler. Sistemin gürültü miktarı  $N_s$ , giriş gürültü seviyesi  $N_i$  ve sistem geri plan gürültüsü  $N_b$  -174dBm/Hz @ 25°C olmak üzere, sistemdeki gürültü miktarı,

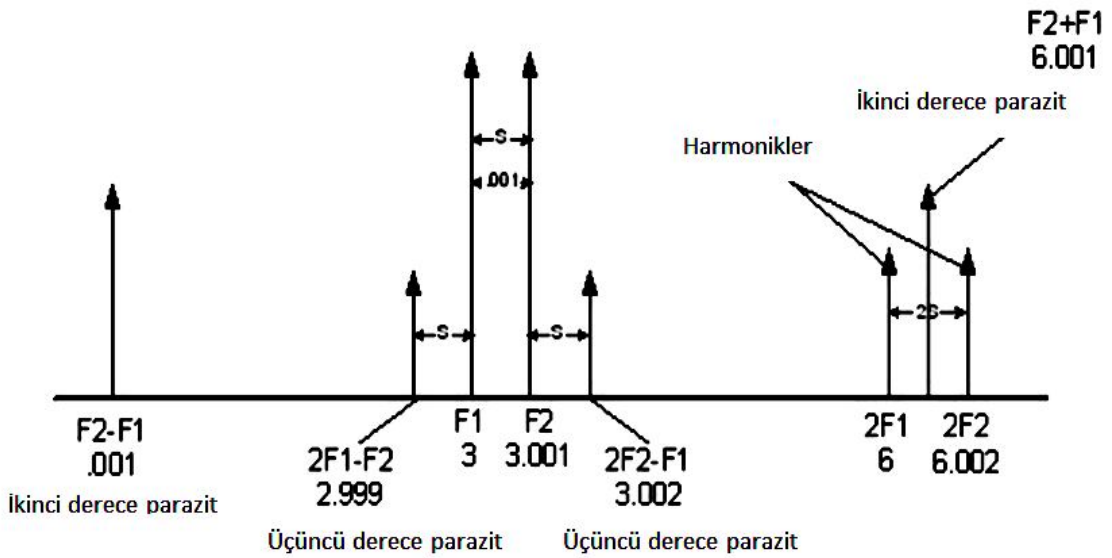
$$N_s = N_i - N_b \quad (2.1)$$

olarak hesaplanır (Anonymous).

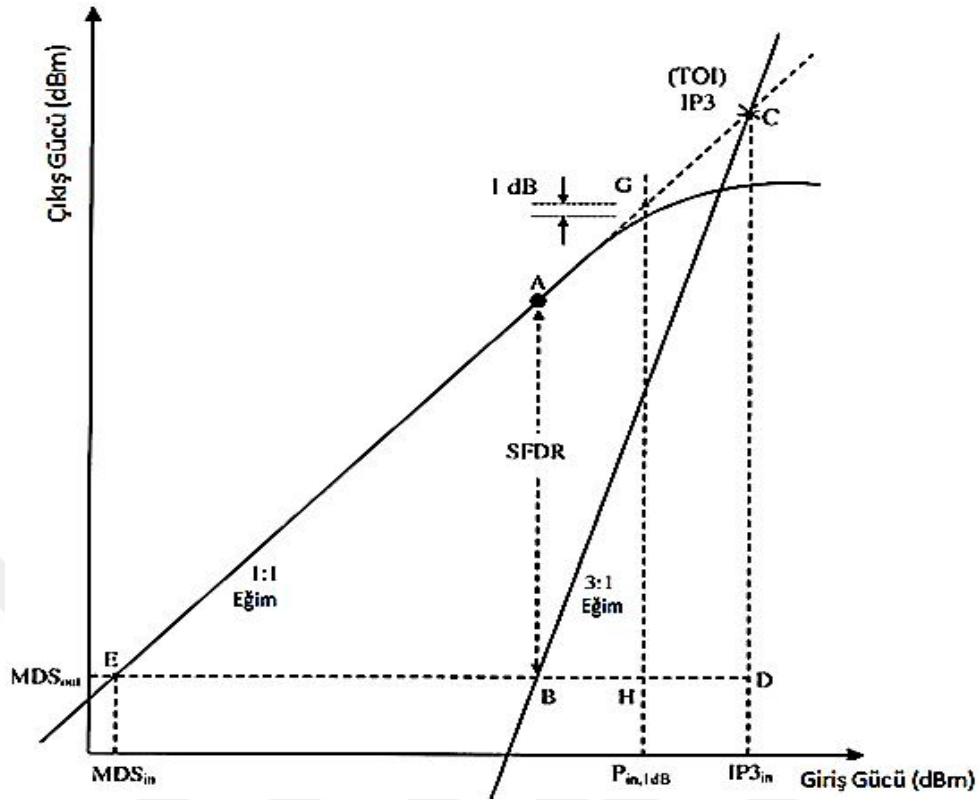


Şekil 2.2. SFDR (parazitsiz dinamik aralık)

Sayısal frekans sentezleyicileri gibi kusursuz lineerliğe sahip olmayan sistemlerde,  $f_0$  frekanslı bir sinyal girişi  $f_0$  frekansının katlarında harmonikler oluşturur. Parazitsiz dinamik aralığın üst sınırını bu harmonik bozulmaları belirler. Lineer olmayan bir sisteme  $f_0$  ve  $f_1$  frekanslı iki sinyal uygulandığında,  $f_0 + f_1$  ve  $f_0 - f_1$  frekanslarında ikinci derece parazitler ve bu parazitlerin  $f_0$  ve  $f_1$  temel frekanslı sinyaller ile etkileşimi ile de üçüncü derece parazitler oluşur. Giriş sinyallerindeki her bir dB artışı üçüncü derece harmoniğe üç dB olarak yansımaktadır (Anonymous). Şekil 2.3'te ikinci ve üçüncü derece harmoniklerin oluşumu verilmiştir. Temel frekans bileşeni ve üçüncü derece harmonik bileşenine ait giriş gücü-çıkış gücü eğrileri çizildiğinde bu eğrilerin bir noktada (IP3) kesiştiği görülür ve bu nokta sayesinde parazitsiz dinamik aralık üst sınırı belirlenir. Parazitsiz dinamik aralık üst sınırının belirlenmesini gösteren grafik Şekil 2.4'te verilmiştir.



Şekil 2.3. İkinci ve üçüncü derece parazit oluşumu (Anonymous)



Şekil 2.4. Parazitsiz dinamik aralık tespiti (Chang, 2004)

Şekil 2.4'te de görüldüğü üzere parazitsiz dinamik aralık, A ve B noktaları arasındaki aralık olarak tanımlanmıştır. Burada B noktası, aynı giriş gücü için, çıkıştaki üçüncü derece harmoniği IP3 gücünün tespit edilebilir en düşük seviyedeki sinyal (MDS) seviyesine eşit olduğu noktadır. A noktası ise aynı giriş gücü, için temel frekansa ait eğrinin çıkış gücü seviyesidir. A ve B noktaları arasındaki mesafe yani parazitsiz dinamik aralık hesaplanırken, şekilde verilen EBA, BDC ve EDC üçgenleri arasındaki trigonometrik hesaplamalardan yararlanılır. EC eğrisi eğimi bir olduğu için, ED ve CD uzunlukları eşittir. BC eğrisi eğimi üç olduğu için de BD uzunluğu CD ve ED uzunluklarının üçte biridir. Bu durumda EB uzunluğu, CD uzunluğunun üçte ikisine eşit olur. IP3 noktası C noktasındaki çıkış gücü seviyesi ve MDS tespit edilebilir en düşük sinyal seviyesi olmak üzere, parazitsiz dinamik aralık aşağıdaki denklem ile hesaplanır (Chang, 2004).

$$SFDR = \frac{2}{3}(IP3 - MDS) \quad (2.2)$$

Sayısal frekans sentezleyicileri gibi kuantalama işlemine ihtiyaç duyulan sayısal sistemlerde, kuantalama hatasını azaltmak ya da bant genişliğine yaymak için en az önemli bitin yarısı büyüklüğünde sayısal bir gürültü sisteme ilave edilir. Sistem çıkış gürültü tabanı bir miktar yükselse de parazitsiz dinamik aralık iyileştirilmiş olur.

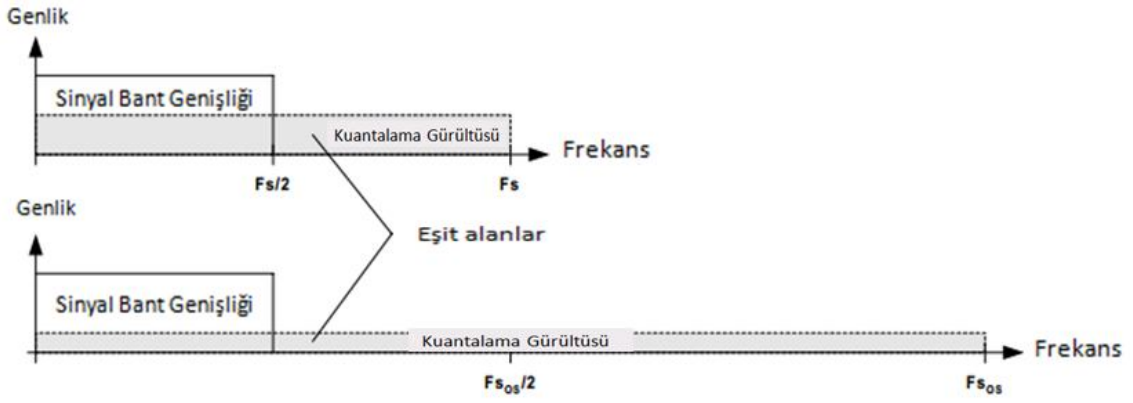
#### **2.2.1.1. Faz kırpmasının dinamik performansa etkisi**

LUT tabanlı sayısal frekans sentezleyicilerinde, ihtiyaç duyulan hafızanın boyutunu düşürmek için faz kırpması uygulanır.  $N$  bitlik bir faz akümülatörü bulunan bir sentezleyicide, girişler  $k$  bitle ifade edilirse,  $2^N \times k$  boyutunda bir ROM'a ihtiyaç duyulmaktadır. Kullanılan ROM boyutunun, faz bilgisinin ifade edildiği bit sayısı ile üstel olarak artmasından dolayı,  $Q < N$  olmak koşuluyla faz bilgisi  $Q$  bit ile ifade edilerek ROM boyutu düşürülebilir. Göz ardı edilen  $N - Q$  bitlik faz bilgisi, faz-genlik dönüştürücü kısmıyla birlikte sentezleyici çıkış sinyali genliğine periyodik bir gürültü olarak etki eder. Bu gürültü faz akümülatör boyutuna, kırpılmış faz bilgisi boyutuna ve ayarlama sözcüğüne bağlıdır. Faz kırpması kaynaklı harmonik genliği, kırılan bit sayısı 4 veya daha fazla olmak üzere  $6.02 \times Q$  dBc üst sınırına sahiptir (Kester, 1999).

#### **2.2.1.2. Sayısal-analog dönüştürücünün dinamik performansa etkisi**

Sayısal-analog dönüştürücüler, kusursuz bir lineerliği olmamasından dolayı dinamik performansa olumsuz etki etmektedir ve bu etki kuantalama hatalarından kaynaklanan gürültüler olarak spektruma yansımaktadır. Sinyalin kuantalama gürültülerine oranı (SQR), spektrumdaki kuantalama gürültülerinin dinamik performansa etkisini gösteren bir büyüklüktür. Sayısal-analog dönüştürücünün çözünürlüğü bu büyüklüğü doğrudan etkilemektedir. Çözünürlük arttıkça SQR'de artmaktadır.

Kuantalama hatalarının çözünürlükle alakalı ve sabit olduğu düşünüldüğünde, sayısal analog dönüştürücünün çalıştığı ölçeğin tam skala ölçeğine oranı da SQR değerine etki etmektedir. Benzer şekilde, örnekleme frekansının artırılmasıyla kuantalama gürültüsü spektrumda daha geniş bir yer tutarken gürültü genlik değeri azaltılabilir. Bu sayede SQR değeri iyileştirilebilir. Örnekleme frekansının artırılmasıyla gürültü genliğinin değişimi Şekil 2.5'te verilmiştir.



Şekil 2.5. Örnekleme frekansının artırılmasının gürültü genliğine etkisi (Kılıç, 2006)

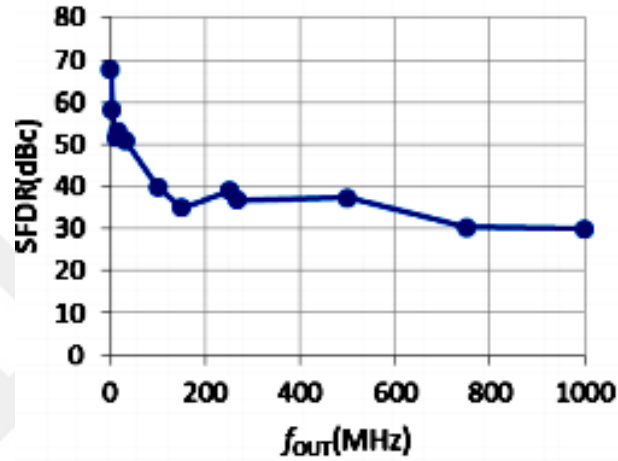
### 2.3. Çıkış Frekansı

Sayısal frekans sentezleyicilerinin önemli özelliklerinden birisi de çıkış frekansıdır. Yayılı spektrum haberleşme sistemleri ve benzeri birçok uygulamada, geniş bir bant aralığında, küçük frekans basamaklarında hızlı frekans uyulmaması gerektiğinden geleneksel frekans sentezleyicileri ihtiyaçları karşılayamamaktadır. Geleneksel frekans sentezleyicileri ele alındığında, yüksek frekansta bir sentezleme işlemi yapılmak istendiğinde kullanılan faz akümülatörü, sinüs değer tablosu ve sayısal-analog dönüştürücü kısımlarının her biri sisteme hız açısından bir kısıtlama getirmektedir.

Doğrudan sayısal frekans sentezleyici tasarımı, harcanan güç, gerek duyulan çalışma frekansı ve SFDR gibi parametreleri olan bir problem olarak düşünüldüğünde, bu değişkenlerden bazıları bakımından kayıplar verilirken diğerleri açısından çok iyi sonuçlar elde edilen çalışmalar bulunmaktadır. 2000’li yılların başından bu yana yapılan çalışmalarla sayısal frekans sentezleyicileri bazen daha çok güç gerektiren yöntemlerle, bazen de düşük SFDR değerlerine inilerek daha hızlı hale getirilmiştir. 2003-2007 yılları arasında farklı yöntemler kullanılarak gerçekleştirilen doğrudan sayısal frekans sentezleyici tasarımlarının kıyası (Strollo ve ark., 2007) Çizelge 2.1’de verilmiştir. Yuan ve arkadaşları (2015) çalışmalarında oda sıcaklığında 2 GHz maksimum operasyon frekansına kadar çıkabilmiş ve frekans-SFDR ilişkisine ait Şekil 2.6’da verilen sonuca ulaşmışlardır.

**Çizelge 2.1.** 2003-2007 yılları arasında farklı yöntemler kullanılarak gerçekleştirilen doğrudan sayısal frekans sentezleyicilerin kıyası (Strollo ve ark., 2007)

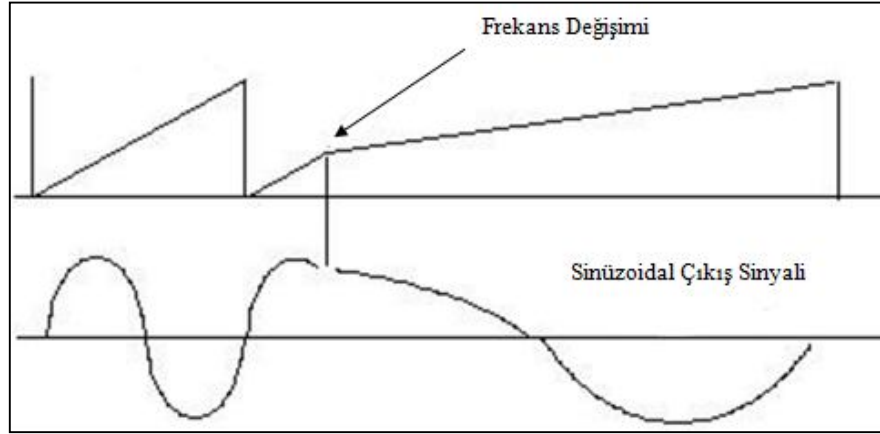
| Devre         | Teknik                            | Aküm. | SFDR | Çıkış | Maks. saat (MHz) | Güç tüketimi ( $\mu\text{W/MHz}$ ) |
|---------------|-----------------------------------|-------|------|-------|------------------|------------------------------------|
| Strollo 2005  | MTM                               | 32    | 90   | 13    | 630              | 121                                |
| Strollo 2007  | Parçalı doğrusal yaklaşım         | 24    | 80   | 12    | 600              | 120                                |
| Langlois 2003 | Parçalı doğrusal yaklaşım         | 32    | 84   | 12    | 150              | 600                                |
| Song 2004     | İnterpolasyon tabanlı aç döndürme | 32    | 100  | 16    | 150              | 2333                               |
| Song 2004     | Parçalı doğrusal yaklaşım         | 30    | 110  | 14    | 250              | 400                                |
| Yang 2004     | Kuad ROM sıkıştırma               | 32    | ~60  | 9     | 820              | 153                                |
| De Caro 2007  | Hibrit DDFS                       | 32    | 90   | 13    | 385              | 400                                |



**Şekil 2.6.** SFDR-frekans ilişkisi (Yuan ve ark., 2015)

#### 2.4. Anahtarlama Hızı

Birden fazla frekans üreten sayısal frekans sentezleyicilerinde bir frekanstan başka bir frekansa geçme ve yerleşme için geçen süre anahtarlama hızı olarak tanımlanmaktadır. Sayısal frekans sentezleyicilerin çok hızlı anahtarlama hızına sahip olmalarının yanı sıra bir frekanstan başka bir frekansa sıçrama sayısal olarak gerçekleştirildiği için kolaydır. Frekans değişimi frekans kontrol sözcüğü ile gerçekleştirilir. Faz akümülatörünü kontrol eden frekans kontrol sözcüğü ile, faz akümülatörü çıkışındaki ayırık rampa sinyalin eğimi değiştirilerek, çıkış frekansının değişimi ilave enerji ihtiyacı duymadan kolaylıkla sağlanabilir. Frekans ayarlama sözcüğü ile değiştirilen faz akümülatörü eğimine bağlı olarak sinüzoidal sinyalin frekansının değişimi Şekil 2.7’de verilmiştir.



Şekil 2.7. Frekans değişimi (Anonymous)

Sayısal frekans sentezleyicilerinin anahtarlama hızını belirleyen kısımlardan birisi şüphesiz sayısal analog dönüştürücü kısmıdır. Sayısal-analog dönüştürücüde sinyalin yerleşme zamanı dört ayrı kısım olarak ele alınır. İlk kısım olan ölü zaman sayısal giriş anahtarlama süresini, yükselme zamanı beklenen seviye artışının gerçekleştiği kısmı, toparlanma zamanı yükselme süresi sonrası kararlı duruma geçene kadar geçen süreyi ve oturma süresi de sinyalin nihai değerini aldığı kısmı ifade etmektedir. Sayısal-analog dönüştürücünün oturma zamanı frekans sentezleyicinin anahtarlama hızına doğrudan etki etmektedir.

Sentezleyicinin anahtarlama hızını belirleyen kısımlardan bir başkası da giriş verisinin paralel ya da seri olarak sisteme verilmesi ve kontrol arayüzüdür. Frekans ayarlama sözcüğünün kontrol arayüzü tarafından girişe uygulanma hızı bir frekanstan diğerine geçme hızına etki etmektedir. Örneğin kontrol arayüzü 200 MHz'lık bir saat sinyali ile veri gönderiyorsa, bir sonraki veri ile birlikte frekans değişmesi için gereken süre 5 ns'dir. Yeni bir veri gönderimi için geçen süre sayısal-analog dönüştürücü oturma süresinden fazla olmalıdır.

## 2.5. Harcanan Güç ve ROM Boyutu

Sayısal frekans sentezleyicilerinin temeli 1970'li yıllara dayansa da son yıllarda özellikle askeri radar uygulamalarında, sayısal haberleşme terminallerinde vb. uygulamalardaki kullanımı hızla artmaktadır. Bu süre zarfında, özellikle güç tüketiminin azaltılması ve daha geniş bantlarda frekans sentezleyebilmek için sayısal frekans sentezleyicileri geliştirilmeye çalışılmıştır. Sayısal elektronik teknolojisindeki

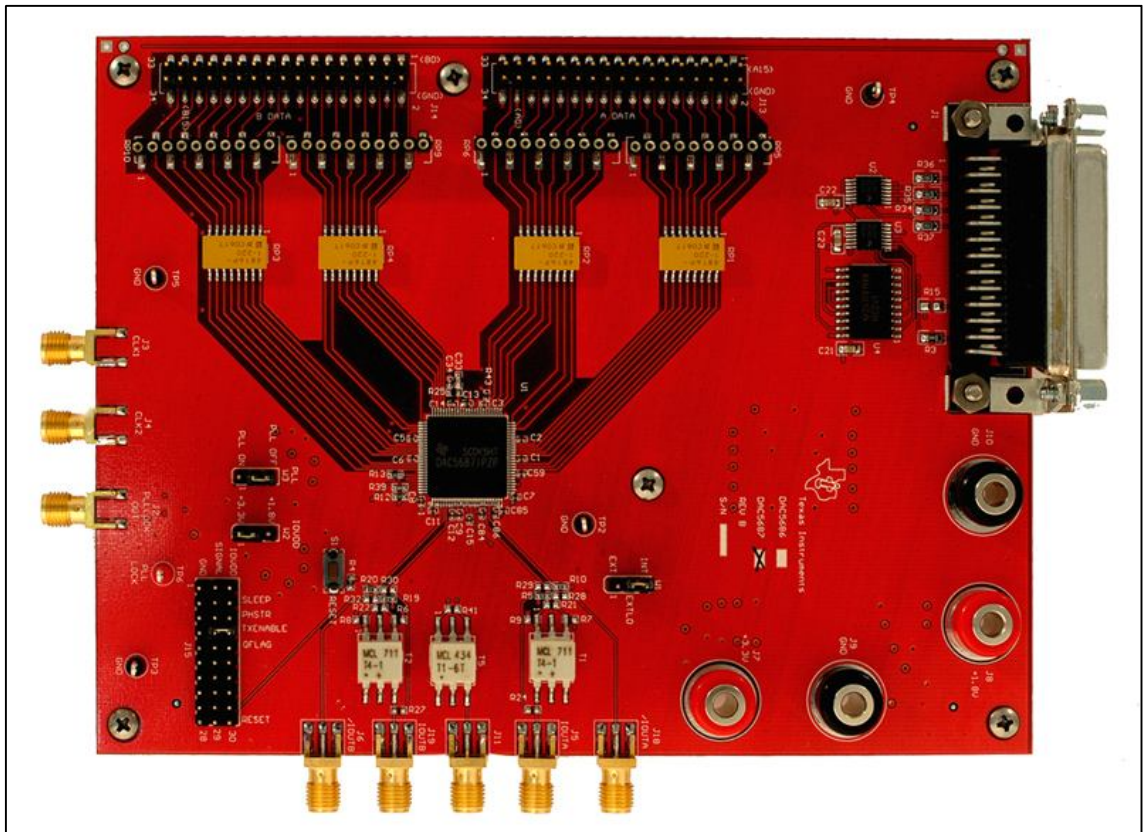
gelişmelere paralel olarak birçok gelişme kaydedilmişse de bu alanda çalışmalar halen devam etmektedir. Güç tüketimini azaltmak için genellikle ROM sıkıştırma teknikleri ile ROM boyutunun indirgenmesi yöntemine gidilmiş ya da ROM tabanlı olmayan sentezleyiciler ile bu amaca ulaşılmıştır.



### 3. MATERYAL

#### 3.1. DAC5687EVM Geliştirme Kartı

DAC5687EVM, TI firmasının DAC5687 tümleşik devresi ile geliştirdiği bir sayısal-analog dönüştürücü kartıdır. CMOS, LVDS ve paralel veri arayüzleri ile veri aktarımı sağlayan geliştirme kartı üzerinde, sayısal girişler için 2 adet 34-pin konektör ile saat giriş ve çıkışları için 8 adet SMA konektör bulunmaktadır. DAC5687EVM geliştirme kartının üstten görünüşü Şekil 3.1’de verilmiştir.

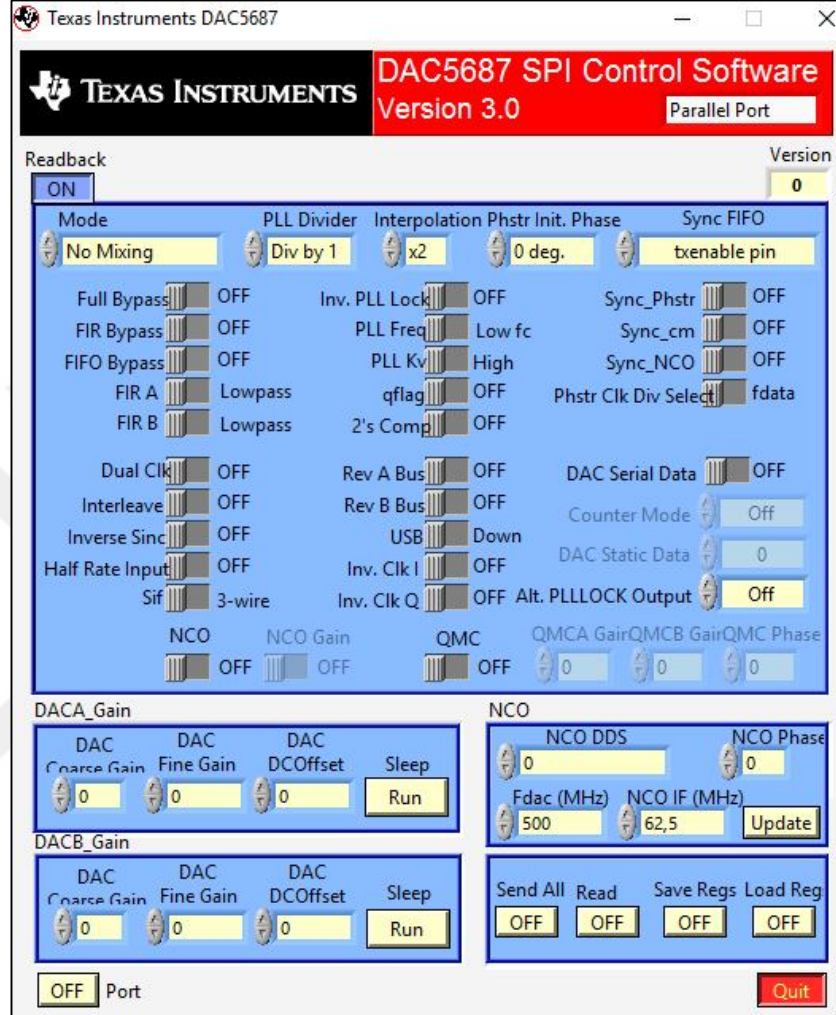


Şekil 3.1. DAC5687EVM geliştirme kartı

500 MSPS maksimum örnekleme hızına sahip 16 bitlik çift kanallı bir sayısal-analog dönüştürücü olan DAC5687 tümleşik devresi; 2x, 4x ve 8x olarak seçilebilen interpolasyon filtresi, dâhili PLL/VCO çarpıcı, 32 bit NCO ve FIR filtreleri bulundurmaktadır.

DAC5687EVM geliştirme kartı 250 MSPS örnekleme hızına kadar CMOS seviyesindeki sayısal girişleri analoğa dönüştürülebilmektedir (Anonymous, 2013). Sayısal girişler için 2 adet 34 pin header bulunurken, giriş saat sinyali ve analog

çıkışlar için SMA konnektörler bulunmaktadır. Seri ve paralel olarak veri aktarımına müsaade eden geliştirme kartı için, gerekli ayarlar ve değişikliklerin bilgisayar üzerinden yapılması için TI firmasının tasarladığı arayüz Şekil 3.2’te verilmiştir.



Şekil 3.2. DAC5687 Kontrol Arayüzü

### 3.1.1. DAC5687 için saat sinyali sağlama yöntemleri

Barındırdığı mantıksal devreler, FIR filtreleri ve sayısal analog dönüştürücü için gerekli olan saat sinyali, DAC5687 tümleşik devresine üç farklı biçimde sağlanabilmektedir. Bu çalışmada harici saat yöntemi kullanılmıştır.

Bu yöntemde, tümleşik devreye sayısal analog dönüştürücünün örnekleme frekansında bir sinyal CLK2/CLK2C bacaklarından sağlanır. Tümleşik devre bu sinyalin frekansını seçilen interpolasyon filtresi oranında bölerek PLLLOCK çıkışı

olarak verir. Bu çıkış tümleşik devreye gönderilecek olan sayısal veri kaynağının saat sinyal ihtiyacı için kullanılır.

### 3.1.2. DAC5687EVM çıkış seçenekleri

Geliştirme kartının fabrika çıkış yapılandırması trafo kuplajlı sinyal çıkışı sağlamaktadır. Bu yapılandırma ile çift sonlandırmalı kabloyu sürebilmeye imkân tanınmıştır. A ve B olarak adlandırılan kanalların her biri maksimum 20 mA çıkış sağlayabilmektedir. Ayrıca kart, kanallara özdeş sayısal bilgi uygulayarak çıkışları toplamaya ve 40 mA çıkış elde etmeye de imkân tanımaktadır.

Geliştirme kartı üzerindeki bazı elemanları söküp, yeni elemanlar ilave ederek farklı şekillerde çıkış almak mümkündür. Bu çalışmada, tavsiye edilen eleman değişiklikleri gerçekleştirilerek çıkış seçeneği değiştirilmiştir. Tamponlanmamış diferansiyel sinyal çıkışı olarak adlandırılan bu yapılandırmada tam skala değerinde  $0.5V_{pp}$  çıkış gerilimi elde edilmektedir. Her kanal için birbirinin eşleniği olan  $I_{outX1}$  ve  $I_{outX2}$  akımlarının yalnızca biri kullanılarak çıkış elde edilir. Çıkış gerilimi 2.8V minimum, 3.3V maksimum değerine sahiptir.

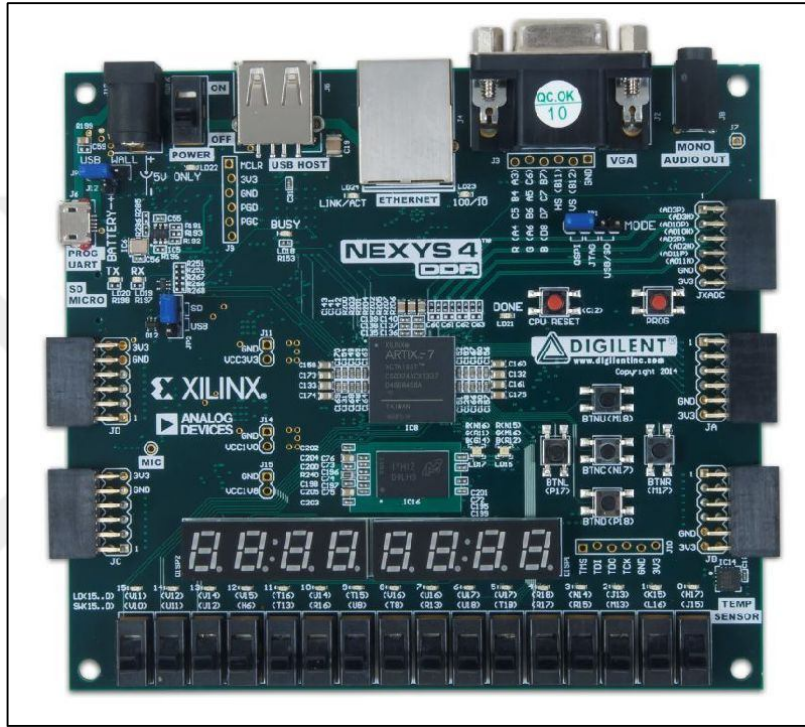
### 3.2. Nexys4 DDR™ FPGA Kartı

Nexys4 DDR™, Digilent firmasının ürettiği bir FPGA geliştirme kartıdır ve Xilinx firmasının ürettiği 7. nesil FPGA ailelerinden biri olan Artix7 tabanlı bir FPGA (XC7A100T-1CSG324C ) kullanılmaktadır. Kullanılan FPGA'nın sahip olduğu özelliklerden bazıları şunlardır (Anonymous, 2016).

- Her birinde dört adet 6-giriş LUT ve 8 flip-flop bulunan 15,850 mantık bloğu,
- 4,860 Kbit hızlı BRAM,
- Her biri PLL içeren 6 adet saat yönetim bloğu,
- 240 DSP bloğu,
- 450 MHz üzeri dâhili saat hızı,
- Dâhili analog-sayısal dönüştürücü.
- Nexys4 DDR™ FPGA kartının üzerinde bulunan donanımlar şunlardır.
  - 16 adet anahtar,
  - 16 adet LED,
  - 2 adet 4 basamaklı 7 parçalı gösterge,
  - USB-UART köprüsü,
  - PDM mikrofon,
  - 3 eksenli ivmeölçer,
  - Sıcaklık sensörü,
  - 10/100 PHY Ethernet,

- 2 adet 3-renk LED,
- Mikro SD kart konektörü,
- 12-bit VGA çıkışı,
- PWM audio çıkışı,
- USB HID sunucusu.
- 128MiB DDR2,
- Seri flash,
- 5 adet Pmod portu,
- USB-JTAG PORTU,

Nexys4 DDR™ geliştirme kartının üstten görünümü Şekil 3.3'te verilmiştir.



Şekil 3.3. Nexys4 DDR™ FPGA Geliştirme Kartı

### 3.2.1. Nexys4 DDR™ saat sinyali

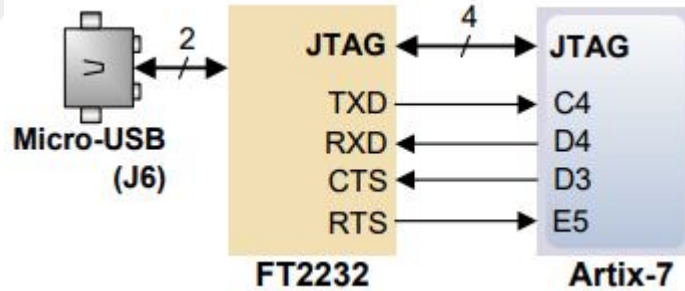
Nexys4 DDR™ geliştirme kartı üzerinde 100 MHz'lık bir kristal osilatör bulunmaktadır. Bu osilatör FPGA'nın E3 bacağına bağlıdır. Farklı frekans ihtiyaçları için karma mod saat yönetici (MMCM) ve faz kilitli döngü (PLL) birimleri kullanılabilir. Bu çalışmada *Clocking Wizard IP* ile MMCM birimi kullanılmıştır.

### 3.2.2. Nexys4 DDRTM giriş ve çıkışları

Geliştirme kartı giriş ve çıkış bağlantıları üzerinde bulunan çevresel modül (PMOD) konektörleri aracılığı ile yapılmaktadır. Bu konektörler düşük frekanslı giriş ve çıkış sinyallerinde kullanılmak üzere tasarlanmıştır (Anonymous, 2011). Bu çalışmada, kart üzerinde bulunan beş adet PMOD konektörden üç tanesi kullanılmıştır. İki konektör 16 bit sayısal veri iletimi için kullanılırken üçüncü konektör saat sinyali iletiminde kullanılmıştır.

### 3.2.3. Nexys4 DDRTM USB-UART köprüsü

Nexys4 DDR™ üzerinde PC uygulamalarında kullanılmaya olanak tanıyan bir FTDI FT2232HQ USB-UART köprüsü bulunmaktadır. Bu köprü sayesinde Windows haberleşme bağlantı noktası komutları kullanılarak kolaylıkla kart ile iletişim sağlanabilmektedir. Kartın köprü ile bağlantısı Şekil 3.4’te verilmiştir.



Şekil 3.4. Nexys4-FT2232 bağlantısı

### 3.3. Vivado 2016.3 yazılımı

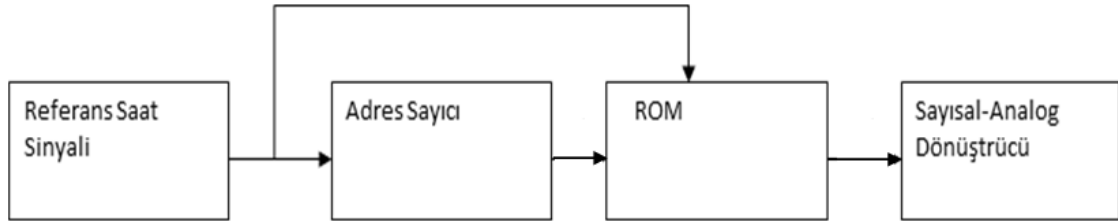
Bu çalışmada tasarımların VHDL kodlarını sentezlemek, simüle etmek ve oluşturulan kodun FPGA'ya aktarılmasında Vivado 2016.3 yazılımı kullanılmıştır. Yazılım VHDL ve Verilog dillerinde tasarım yapma seçeneği sunmaktadır. Bu çalışmada VHDL dilinde kodlama yapılmıştır.

Vivado yazılımı sahip olduğu geniş IP kataloğu ile birçok hazır tasarım bloğunun da kullanımına imkân tanımaktadır. Bu çalışmada blok hafızalar için *blok memory generator*, saat sinyal ihtiyacı için ise *clocking wizard* hazır blok sağlayıcıları kullanılmıştır.

## 4. YÖNTEM

Bu tez çalışmasında frekans sentezlemek için DDFS yönteminden, kullanılan ROM boyutunu sıkıştırmak için ise iki parçalı tablo yönteminden faydalanılmıştır. Bu kısımda DDFS yöntemi ile tasarlanan bir frekans sentezleyicinin blokları ele alınacaktır.

Sayısal frekans sentezleyici referans saat sinyali, adres sayıcı, ROM ve sayısal-analog dönüştürücü ile basitçe gerçekleştirilebilir. Burada, referans saat sinyali tetiklemesi ile adres sayıcı, ROM'daki sinüs tablosundan bir periyotluk sinüs genlik bilgisine erişir ve bu sayısal bilgi, sayısal-analog dönüştürücüye iletilir. Analog-sayısal dönüştürücü çıkışı, sayısal frekans sentezleyicinin de çıkışıdır. Bu basit sayısal frekans sisteminde çıkış frekansını ayarlamak için saat sinyali frekansı değiştirilmeli ya da ROM'a kaydedilen sinüs değerleri adımı değiştirilmelidir. Her iki yöntemde de yüksek hızlı çıkış frekans atlama sağlanamamaktadır (Kester, 1999). Basit bir sayısal sinyal sentezleyicinin blok şeması Şekil 4.1'de verilmiştir.



Şekil 4.1. Basit sayısal sinyal sentezleyici

### 4.1. Faz Akümülatörü

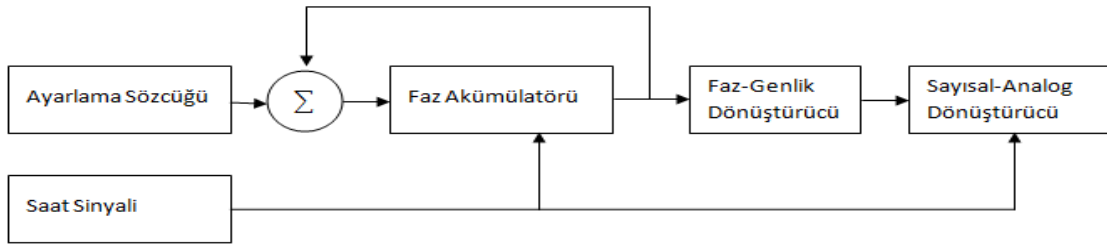
Ayarlanabilir bir çıkış frekansı elde edebilmek için sayısal frekans sentezleyicisine faz akümülatörü ilave edilir.  $N$  bitlik bir ayarlama sözcüğü ile faz akümülatörü kontrol edilir. Faz akümülatörü sayesinde referans saat sinyali sabit kalacak şekilde çıkış frekansı hedeflenen şekilde ayarlanabilir. Faz akümülatörü ile sayısal bir faz çemberi oluşturulur ve çember üzerindeki her nokta o faza ait genlik bilgisini içerir. Çember üzerinde oluşturulacak nokta sayısını ayarlama sözcüğünün bit sayısı belirler. Ayarlama sözcüğünün değeri ise faz çemberi üzerindeki atlama adımını ayarlamaktadır. Ayarlanabilir frekanslı bir sayısal sinyal sentezleyicinin blok şeması Şekil 4.2'de verilmiştir.

Faz akümülatörü, referans saat sinyali ile değerini artıran bir sayıcı olarak çalışmaktadır. Her saat sinyalinde artımın ne kadar olacağını ise atlama adımı

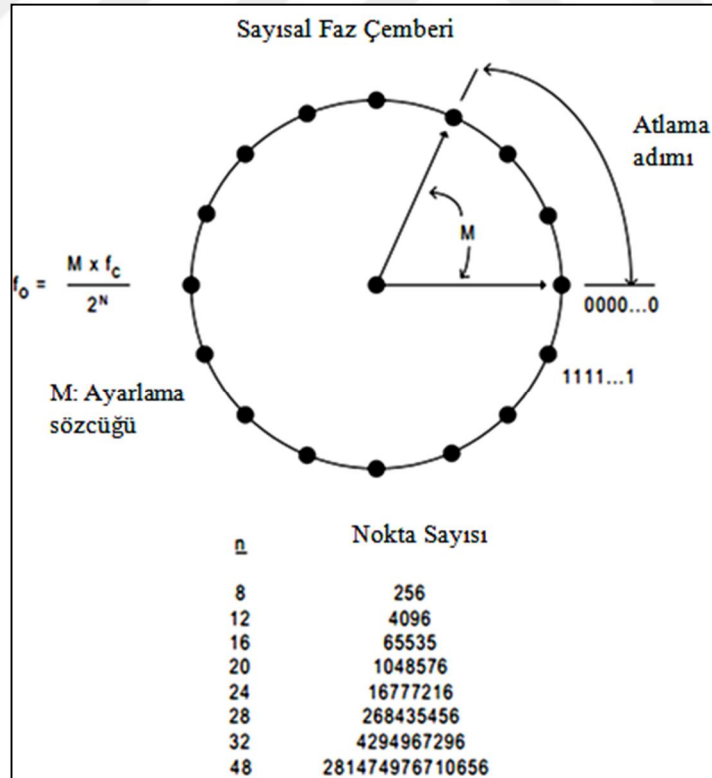
belirlemektedir. Atlama adımı ne kadar fazla olursa sayısal faz çemberi üzerindeki dönüş daha hızlı olur ve sinüs sinyalinin periyodu daha çabuk tamamlanır. Elde edilecek sinyalin frekansı Eşitlik (4.1) ile hesaplanır.

$$F_{\text{Çıkış}} = \frac{\text{Ayarlama Sözcüğü} \times \text{Giriş Frekansı}}{2^N} \quad (4.1)$$

$N$  bitlik ayarlama sözcüğü ile çıkış frekansı arasındaki ilişki Şekil 4.3'te verilmiştir.



Şekil 4.2. Ayarlanabilir frekanslı sayısal frekans sentezleyici



Şekil 4.3. Faz çemberi ve ayarlama sözcüğü (Kester, 1999)

## 4.2. Faz-Genlik Dönüştürücü

Tipik bir sayısal frekans sentezleyicide kullanılan bir başka yapı ise hafıza (RAM/ROM) birimleridir. Bu hafıza birimleri genellikle LUT (Look Up Table) olarak kullanılır ve  $x$  girişine karşılık gelen  $\sin(x)$  değerini çıkış olarak verir.

Yüksek faz çözünürlüğü gerektiren sistemlerde daha çok sayıda  $x$  girişine karşılık gelen  $\sin(x)$  değerleri içeren tabloları barındıran yüksek bitli hafıza birimleri kullanılmalıdır. Daha büyük kapasiteli bir hafıza birimi kullanmak, sistemde harcanan gücü artırmakla beraber daha fazla alan gerektirdiği için mobil uygulamalar gibi düşük güç ve küçük alan gerektiren uygulamalarda, ROM içermeyen sayısal frekans sentezleyici yapısı ya da ROM sıkıştırma yöntemleri kullanılmaktadır.

Hedeflenen çıkış sinyal frekansı çok yüksek olduğunda sistemi hız açısından sınırlandıran yapı hafıza birimleri olmaktadır. Hafıza biriminden kaynaklı yavaşlamayı önlemek için bazı çözümler geliştirilmiştir. Bu çözümler:

- Sistem saat sinyal frekansında çalışan tek bir hafıza birimi kullanmak yerine  $A$  sayıda hafıza birimi kullanarak her bir hafıza biriminin çalışma frekansını  $1/A$  seviyesine indirmek,
- Sinüs fonksiyonun simetrik ve monoton yapısı kullanılarak hafıza birimini sıkıştırmak,
- Çok sayıda mantıksal dizilerle birlikte küçük hafıza birimleri kullanmak ya da hafıza birimi içermeyen yapılar kullanmak,

olarak sıralanabilir. Bu çözümlere ilave olarak veri dönüştürme elemanlarındaki bitlerin ağırlıklandırılması da henüz verimli bir şekilde gerçekleştirilemeyen bir yöntemdir (Anonymous).

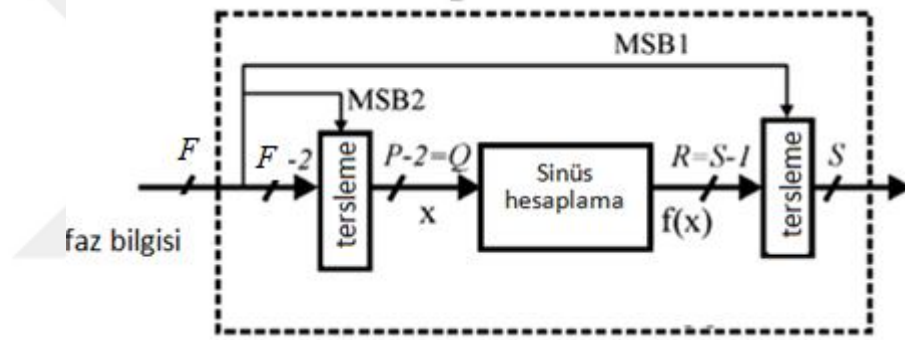
### 4.2.1. ROM sıkıştırma yöntemleri

Sayısal frekans sentezleyicilerinin en basit şekli Şekil 4.1’de verildiği gibidir ve bu yapıda hafıza birimlerinden LUT olarak faydalanılır. LUT yapıları, faz çemberleri gibi düşünülecek olursa, herhangi bir faza ait sinüs değerini gösteren tablolar olarak kullanılır. Bu yapılar kullanılırken faz çözünürlüğünü artırmak için faz tekerleği üzerinde çok sayıda nokta oluşturulabilir. Ancak bu işlem kullanılacak ROM boyutunu ve harcanan gücü artırmakla beraber sistem hızını da sınırlayacaktır. Tüm bu olumsuzlukları gidermek adına bazı ROM sıkıştırma teknikleri geliştirilmiştir.

#### 4.2.1.1. Kuadrant sıkıştırma tekniği

Bu teknikte sinüs dalgasının simetri özelliği kullanılmaktadır.  $0-2\pi$  aralığındaki sinüs değerlerini içeren LUT kullanmak yerine  $0- \pi/2$  aralığındaki sinüs değerlerini içeren bir LUT kullanılarak gerekli ROM boyutu düşürülmektedir. Sistemin blok şeması Şekil 4.4'te, sinüs simetrisinin nasıl kullanıldığı Çizelge 4.1'de verilmiştir.  $F$  bitlik giriş sinyalinin en önemli 2 biti, bu sinyalin fazının bulunduğu çeyreği ifade etmektedir. Giriş sinyalinin diğer bitleri ise  $0- \pi/2$  arasındaki açı bilgisini tutmaktadır. Sinüs hesaplama bloğu Eşitlik (4.2) ile faza ait sinüs değerini hesaplamaktadır (De Caro ve ark., 2008).

$$f(x) = \sin \left[ \frac{\pi}{2} \left( x + \frac{1}{2^{(2^Q)}} \right) \right] \quad (4.2)$$



Şekil 4.4. Sinüs simetrisi kullanan sinüs üretici (De Caro ve ark., 2011)

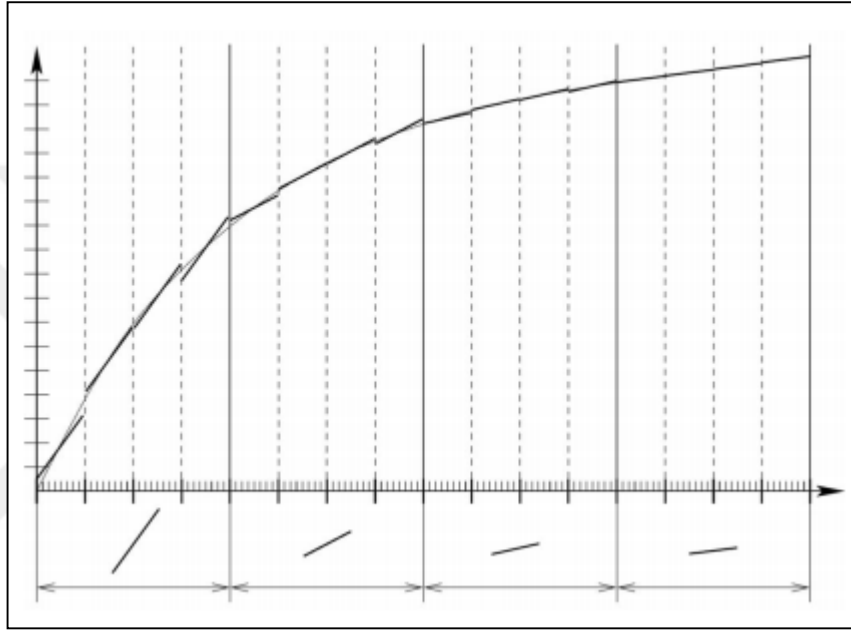
Çizelge 4.1. Sinüs üreticinin sinüs simetrisini kullanması

| Faz Çeyreği     | Sinüs Üretici Çıktısı ( $0 < \theta < 90$ ) | MSB_1 | MSB_2 | Elde Edilen Faz Açısı |
|-----------------|---------------------------------------------|-------|-------|-----------------------|
| Birinci Çeyrek  | $\sin(\theta)$                              | 0     | 0     | $0 < \theta < 90$     |
| İkinci Çeyrek   | $\sin(90 - \theta)$                         | 0     | 1     | $90 < \theta < 180$   |
| Üçüncü Çeyrek   | $-\sin(\theta)$                             | 1     | 0     | $180 < \theta < 270$  |
| Dördüncü Çeyrek | $-\sin(90 - \theta)$                        | 1     | 1     | $270 < \theta < 360$  |

Sinüs hesaplama bloğu öncesindeki tersleme bloğu ile açı değeri terslenmekte, ikinci blok ile de sinüs değeri terslenerek 4 farklı çeyreğe ait sinüs değerleri elde edilmektedir. Başka bir deyişle,  $0-90$  derecelik faz bilgisi ve fazın hangi çeyrekte olduğuna dair bilgi kullanılarak tüm bir sinüs üretilebilmektedir.

#### 4.2.1.2. BTM (İki parçalı tablo yöntemi)

Lineer yaklaşımlı tablo yöntemlerinde, sayısal frekans sentezleyici çıkış sinyaline kısmi parçalı lineer yaklaşım uygulanır.  $x$  eksenini  $2^a$  sayıda eşit uzunlukta parçaya bölünür. Her parça için  $f(x)$  fonksiyonu lineer interpolasyon ile hesaplanır. Sinüs dalgasının çeyrek periyodunun BTM ile interpolasyonu Şekil 4.5'te, interpolasyon için kullanılan eşitlikler Eşitlik (4.3) ve Eşitlik (4.4)'te verilmiştir.



**Şekil 4.5.** Sinüs dalgasının çeyrek periyodunun BTM ile interpolasyonu (De Dinechin ve Tisserand, 2005)

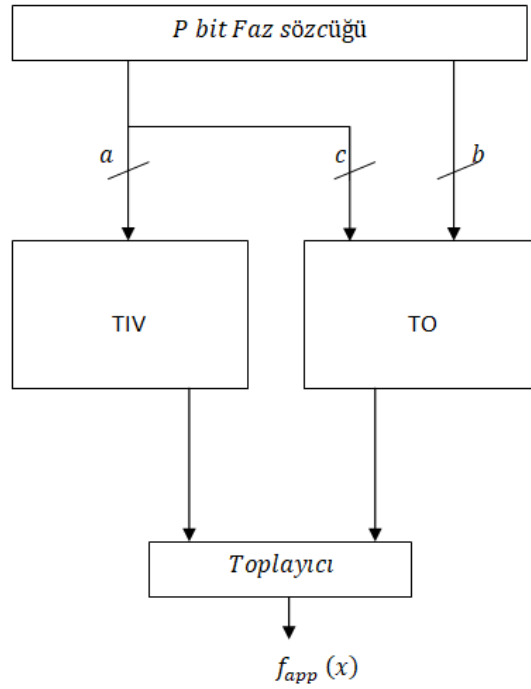
$$m_i = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} \quad (4.3)$$

$$f_{app}(x) = y_i + m_i (x - x_i) \quad (4.4)$$

BTM tekniğinde,  $Q$  bitlik faz bilgisi iki ayrı kısma ayrılır.  $a$  bitlik ilk kısım ilk değer tablosu için gerekli bilgileri tutar. İlk değer tablosunda (TIV)  $x_i$  değerleri tutulur.  $Q - a = b$  bitlik ikinci kısımda ise  $x - x_i$  ve  $m_i$  bilgisi tutulur. Bu bilgiler de ofset tablosunda tutulur (TO).

Bu yöntemde,  $c < a$  olmak üzere eksen  $2^c$  tane daha geniş kısımlara bölünür.  $2^{a-c}$  kadar komşu kısımda aynı eğim bilgisi kullanılır. Bu sayede,  $2^{a+b}$  bitlik bir ROM

yerine,  $2^a + 2^{c+b}$  bitlik bir ROM kullanılmış olur. BTM blok şeması Şekil 4.6'da verilmiştir.



Şekil 4.6. BTM blok şeması

#### 4.2.1.3. MTM (Çok parçalı tablo yöntemi)

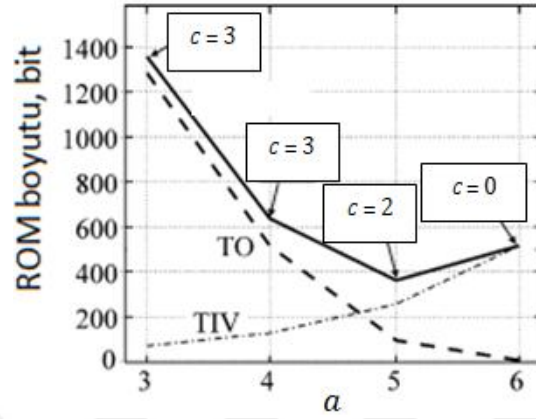
MTM, Dinechin ve Tisserand (2005) tarafından ortaya konmuş, BTM tekniğini temel alan bir yöntemdir. Bu yöntemde BTM tekniğinde kullanılan ofset tablosu yerine daha küçük boyutlarda birden fazla ofset tablosu kullanılmaktadır. BTM tekniğinde ofset tablosunda hesaplanan  $m_i(x-x_i)$  işleminin lineer olma özelliğinden yola çıkarak,  $(x-x_i)$  kısmını ifade eden B sözcüğü k tane alt sözcüğe ayrılarak aynı işlem yapılmaktadır.

$$TO(m_i, x_i) = m_i(x - x_i) \quad (4.5)$$

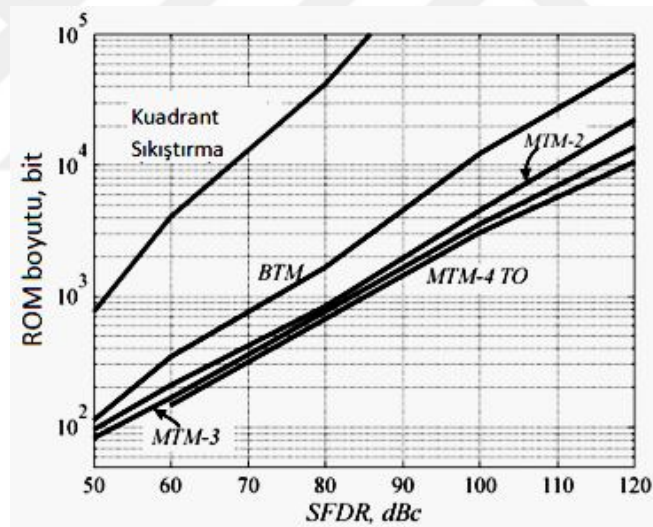
$$m_i(x - x_i) = m_i x_1 + m_i x_2 + \dots + m_i x_k \quad (4.6)$$

Tabloların daha da küçülmesi ile genellikle toplam kullanılan alan azalsa da ilave toplayıcılara ihtiyaç duyulmaktadır. Ayrıca oluşturulan her küçük tablodaki hata oranı çok daha az görünse de toplamdaki hata daha çok olabilmektedir. Bu yöntem çok daha küçük bir alana imkân sağlarken doğruluk oranı daha az ve daha karmaşık bir

çözüm ortaya koymaktadır. De Caro ve arkadaşları (2008), 60 dBc SFDR için BTM optimizasyonu ile  $a = 5$ ,  $c = 2$  olmak üzere ROM boyutunu 352 bit seviyesine indirmişlerdir. Aynı çalışmada farklı sayıda ofset tabloları kullanarak SFDR-ROM boyutu değişimini göstermişlerdir. İlgili grafikler Şekil 4.7 ve Şekil 4.8’de verilmiştir.



Şekil 4.7. 60 dBc SFDR için optimize BTM tasarımı (De Caro ve ark., 2008)

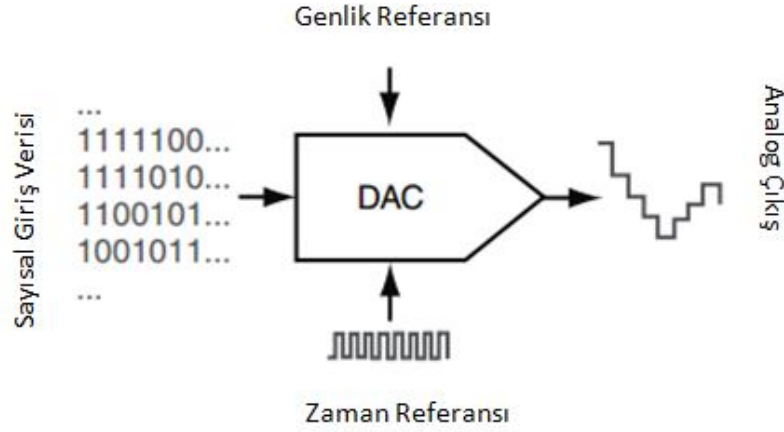


Şekil 4.8. Optimize edilmiş MTM algoritmaları için ROM boyutu-SFDR grafiği (De Caro ve ark., 2008)

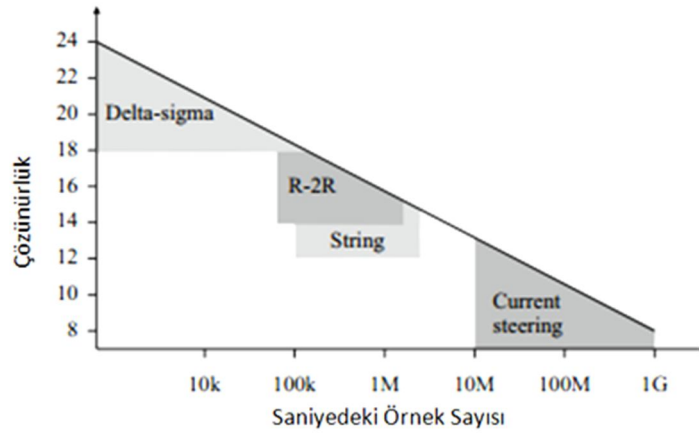
### 4.3. Sayısal-Analog Dönüştürücü

Sayısal-analog dönüştürücüler, sayısal verileri analog değerlere dönüştüren elemanlardır. Sayısal verilerle birlikte referans bir saat sinyali ve referans genlik bilgisini kullanarak analog çıkış sağlarlar. Sayısal giriş verisi, zaman referansı ile belirlenen zaman aralıklarıyla genlik referansı ile karşılaştırılarak bir analog çıkış üretilir (Radulov ve ark., 2011). Sayısal-analog dönüşümü basit olarak Şekil 4.9’da

verilmiştir. Çıkışların daha kararlı olması için örnekle-tut blokları kullanılır. Bu işlemler birçok farklı mimari ile yapılabilmektedir. Kullanılan bazı sayısal-analog dönüştürücü mimarilerinin çözünürlük ve saniyedeki örnekleme sayısı açısından karşılaştırıldığı grafik Şekil 4.10'da verilmiştir.



Şekil 4.9. Sayısal-analog dönüştürümü (Radulov ve ark., 2011)



Şekil 4.10. DAC mimarilerinin karşılaştırılması (Ohnhäuser, 2015)

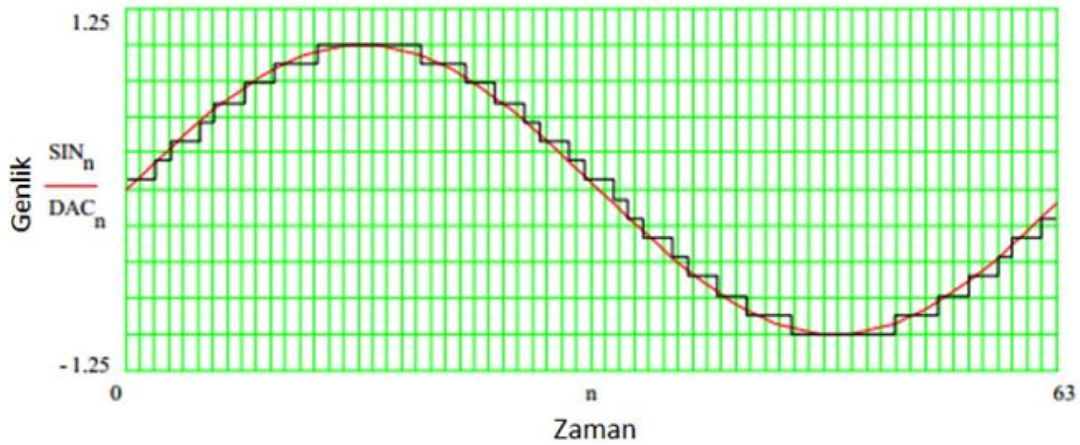
Sayısal-analog dönüştürme kısmı, frekans sentezleyicinin limitlerini belirleyen en önemli kısımlardan birisidir. Kullanılan sayısal-analog dönüştürücünün; sentezleyici frekans çözünürlüğü ve spektral saflığı üzerinde etkisi büyüktür. Bu nedenle kullanılacak dönüştürücünün özelliklerinin de dikkatli incelenmesi gerekmektedir.

#### 4.3.1. Sayısal-analog dönüştürücü parametreleri

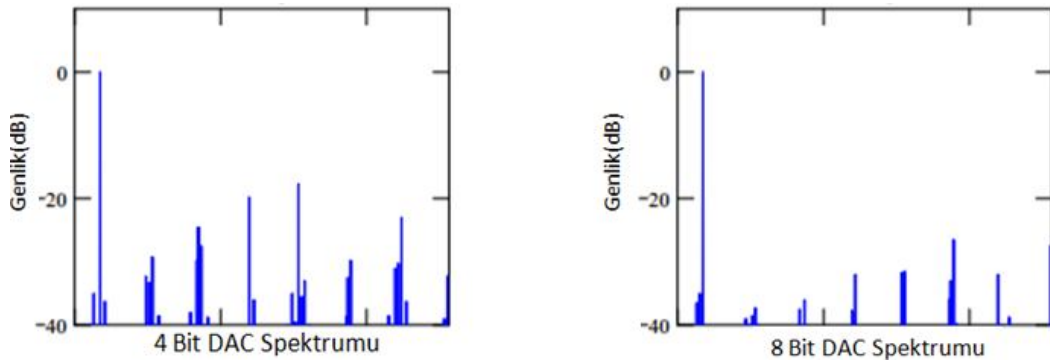
Sayısal-analog dönüştürücülerin en önemli parametreleri örnekleme hızları ve çözünürlük değerleridir. Çözünürlük kavramı kuantalama hataları ile birlikte

düşünülebilir. Çözünürlük arttıkça kuantalama kaynaklı gürültü seviyesi azalır. Çözünürlük değeri, sayısal-analog dönüştürücü giriş bit sayısı ile orantılıdır. Sayısal veri ne kadar çok bit ile ifade edilirse çözünürlük o derece yüksek, kuantalama hataları o kadar az olur. N giriş bit sayısı olmak üzere bir sayısal-analog dönüştürücünün çözünürlüğü  $\frac{\text{Tam Ölçek Değeri}}{2^N}$  olarak hesaplanabilir.

Sayısal-analog dönüştürücü kullanılarak örneklerden elde edilen bir sinüs ile orijinal bir sinüs sinyali arasındaki fark Şekil 4.11’de açıkça görülmektedir. Bir periyodun 64 örnekle örneklendiği şekilde, iki örnekleme arasında dönüştürücünün sınırlı çözünürlüğü nedeniyle genliğin sabit kalması frekans düzleminde yüksek frekanslı parazitler olarak gözlemlenir (Kester,1999). 4 bit ve 8 bit çözünürlükteki sayısal-analog dönüştürücülerin çözünürlük farklarının spektruma nasıl yansıtacağı Şekil 4.12’de verilmiştir.

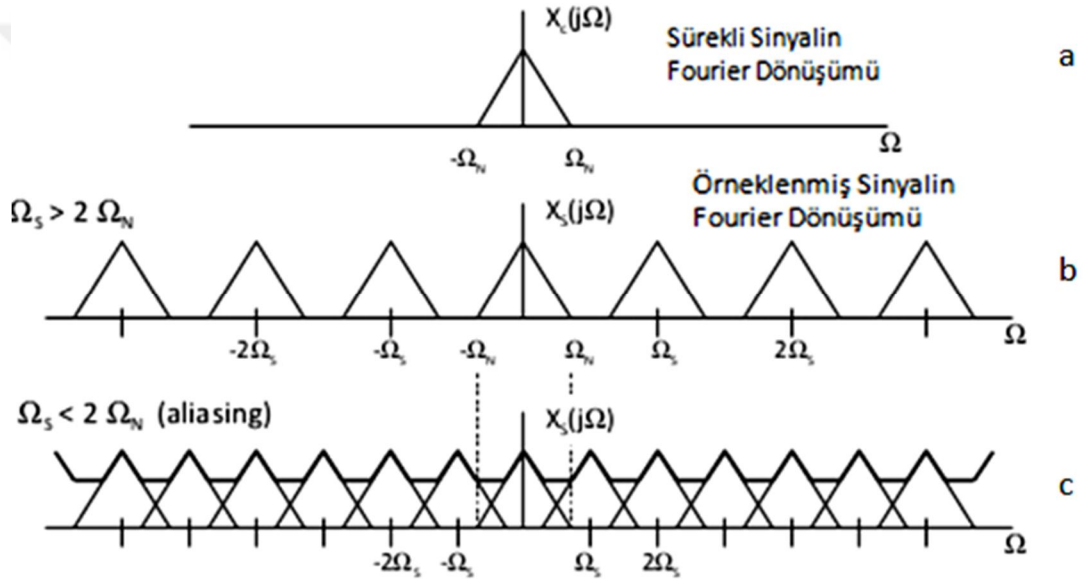


Şekil 4.11. Orijinal Sinüs ile Örneklerden Elde Edilenin Karşılaştırılması (Kester, 1999)



Şekil 4.12. Dört bit ve sekiz bit çözünürlüklü DAC spektrumları (Kester, 1999)

Örnekleme hızı, sayısal-analog dönüştürücünün bir saniyedeki örnekleyebildiği maksimum örnek sayısını ifade etmektedir. Örnekleme teoremine göre, bir  $f_0$  merkez frekanslı sinyal  $f_s$  örnekleme frekansı ile örneklenirse, spektrum üzerinde  $K = 1, 2, 3, 4, \dots$  olmak üzere  $|\pm K f_s \pm f_0|$ 'da görüntüler oluşur. Oluşan bu görüntülerin ana sinyale girişim oluşturmaması için bant genişliği  $f_b - f_a$  olan sinyalin örnekleme frekansı Nyquist teoremine göre  $f_s > 2(f_b - f_a)$  olacak şekilde seçilmelidir (Kester, 1999). Fourier dönüşümü Şekil 4.13.a'da verilen sinyalin  $f_s > 2(f_b - f_a)$  koşuluna uygun örneklenmesi Şekil 4.13.b'de, koşulu sağlamayan örneklenmesi de Şekil 4.13.c'de verilmiştir. Şekilden de kolayca anlaşılabacağı üzere, koşulu sağlamayan örneklemede, örneklenen sinyalden orijinal sinyalin yeniden elde edilmesi mümkün değildir.



Şekil 4.13. Örnekleme teoremi

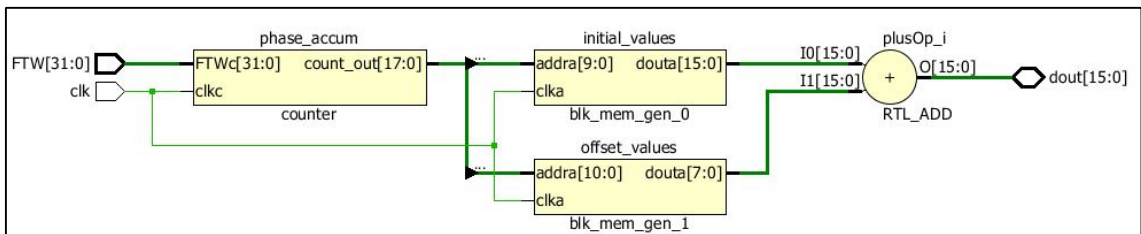
Sayısal-analog dönüştürme kısmı, doğrudan sayısal frekans sentezleyicilerinin en önemli limitlerini belirleyen kısımdır. Bu nedenle tasarımda kullanılacak sayısal-analog dönüştürücünün seçimi çok önemlidir. Dönüştürücüler, sahip olduğu mimariye göre de farklı AC ve DC tepkiler vermektedir. Tasarımda, dönüştürücünün yalnızca çözünürlük ve örnekleme hızı değil sağladığı parazitsiz dinamik aralık gibi özellikleri de dikkate alınmalıdır.

## 5. GERÇEKKLENEN DDFS TASARIMLARI

Bu çalışmada BTM tabanlı 4 farklı frekans sentezleyici VHDL dilinde Vivado 2016.3 yazılımı kullanılarak tasarlanmıştır. İlk olarak, 32 bit sabit faz çözünürlüklü klasik BTM tabanlı DDFS; BTM yönteminin harcanan güç ve spektral saflık performansını ölçmek için tasarlanmıştır. Yüksek SFDR hedeflenen bu ilk tasarımda 4 kB değerinde ROM kullanılarak 71.17 dBc SFDR elde edilmiştir. Bu tasarımla yalnızca sinüs dalgası çıkışı elde edilmektedir. Aynı yöntem ile faz ve genlik çözünürlüğü daha düşük tutularak tasarlanan ikinci sistemde 320 bayt ROM kullanılarak 64.69 dBc SFDR elde edilmiştir. Üçüncü olarak BTM ve kuadrant sıkıştırma tekniğinin birlikte kullanıldığı bir tasarım gerçekleştirilmiştir. Bu tasarımla yöntemlerin birlikte kullanılabilirliği test edilmiştir. Son olarak tasarlanan yapının faz çözünürlüğü kullanıcı tarafından 18-32 bit arasında değiştirilebilmekte olup, genlik çözünürlüğü 16 bittir. Gerçekleştirilen bu son tasarım ile sinüs, testere dişi, kare ve üçgen dalgaları elde etmek mümkündür.

### 5.1.1. Tasarım 1

Bu tasarımda faz akümülatörü, kullanıcı tarafından belirlenen atlama adımı ile  $0-2^{32}$  arasında sayan bir sayıcı olarak çalışır. 32 bitlik sayıcı çıkışı kırpılarak 18 bitlik nihai faz sözcüğüne dönüştürülür. 18 bitlik faz sözcüğü faz genlik dönüştürücü kısmında tabloları adreslemek için kullanılır. En önemli 10 bit ile ilk değer tablosu adreslenirken en önemli 3 bit ve en önemsiz 8 bit ile offset tablosu adreslenir. Tasarım 1'in Vivado yazılımı ile oluşturulmuş blok şeması Şekil 5.1'de verilmiştir.



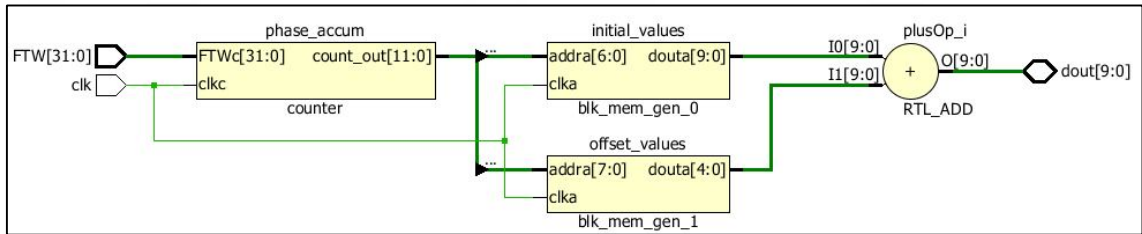
Şekil 5.1. Tasarım 1 blok şeması

Bu tasarımda  $a$ ,  $b$  ve  $c$  parametreleri 10, 8 ve 3 olarak belirlenmiştir. 100 dBc SFDR değeri hedeflenen bu tasarımda, 71.17 dBc SFDR değeri elde edilmiş olup,

tasarımlar arasındaki en yüksek SFDR değeri sağlayan tasarımıdır. Tasarım toplam 4 kB LUT boyutu gerektirmektedir.

### 5.1.2. Tasarım 2

Bu tasarımda da faz akümülatörü, kullanıcı tarafından belirlenen atlama adımı ile  $0-2^{32}$  arasında sayan bir sayıcı olarak çalışır. 32 bitlik sayıcı çıkışı kırpılarak 12 bitlik nihai faz sözcüğüne dönüştürülür. 12 bitlik faz sözcüğünün en önemli 7 biti ile ilk değer tablosu adreslenirken en önemli 3 biti ve en önemsiz 5 biti ile ofset tablosu adreslenir. Tasarım 2'nin Vivado yazılımı ile oluşturulmuş blok şeması Şekil 5.2'de verilmiştir.

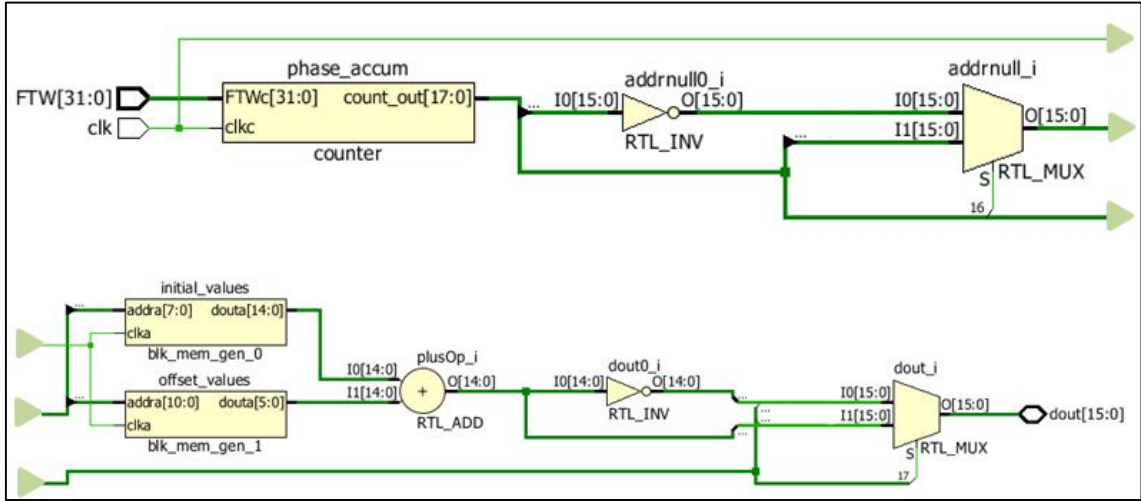


Şekil 5.2. Tasarım 2 blok şeması

Bu tasarımda  $a, b$  ve  $c$  parametreleri 7, 5 ve 3 olarak belirlenmiştir. 70 dBc SFDR değeri hedeflenen bu tasarımda, 64.69 dBc SFDR değeri elde edilmiştir. Tasarım toplam 320 bayt LUT boyutu gerektirmektedir. Bu tasarım, gerçekleştirilen tasarımlar arasında en küçük LUT boyutu gerektiren tasarımıdır.

### 5.1.3. Tasarım 3

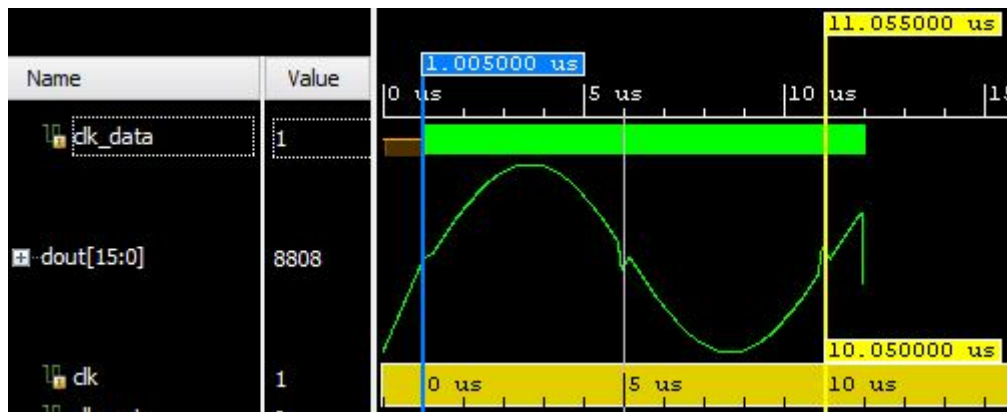
Bu tasarımda BTM ile kuadrant sıkıştırma tekniği birlikte kullanılmıştır. 32 bitlik sayıcı çıkışı kırpılarak 18 bitlik nihai faz sözcüğüne dönüştürülür. Bu tasarımda diğer tasarımların aksine faz sözcüğünün tüm bitleri tablo adreslemekte kullanılmaz. Kırpılmış faz sözcüğünün en önemli 2 biti sinüs çeyreğini belirlemek için kullanılır. Geriye kalan 16 bitin en önemli 8 biti ilk değer tablosunu adreslerken, en önemli 3 biti ve en önemsiz 8 biti ofset tablosunu adresler. Tasarım 3'ün Vivado yazılımı ile oluşturulmuş blok şeması Şekil 5.3'te verilmiştir.



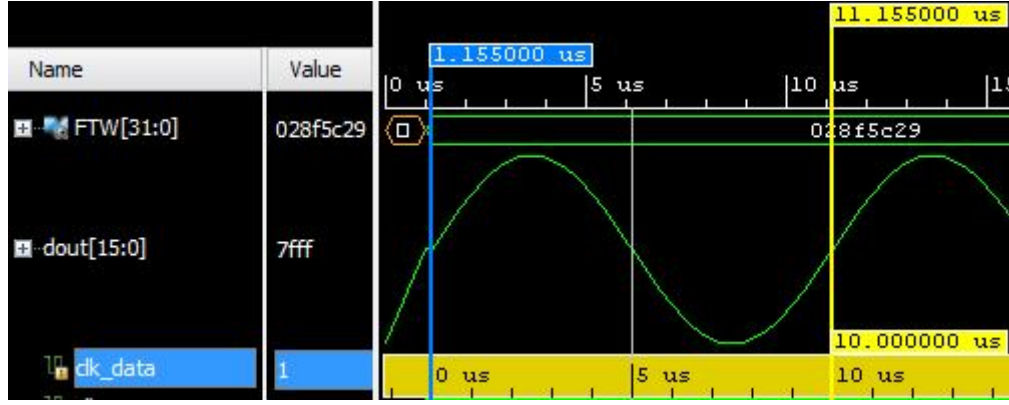
Şekil 5.3. Tasarım 3 blok şeması

Bu tasarımda  $a, b$  ve  $c$  parametreleri 8, 8 ve 3 olarak belirlenmiştir. Tasarım yaklaşık olarak 2 kB LUT boyutu gerektirmektedir. Bu tasarım, gerçekleştirilen tasarımlar arasında en yüksek ROM sıkıştırma oranına sahip tasarımdır. Bu tasarımın ROM sıkıştırma oranı yaklaşık 208:1'dir.

Bu tasarımda unutulmaması gereken en önemli noktalardan birisi de kullanılan ROM'dan kaynaklanan gecikmelerdir. Artix 7 ailesi için ROM gecikmesi 1 saat sinylidir. Yani adres değeri uygulandıktan bir saat sinyali sonra çıkış alınmaktadır. Kuadrant sıkıştırma tekniği kullanarak sinüs simetrisinden faydalanırken bu gecikme göz önünde bulundurulmalı, ikinci ve dördüncü çeyrek sonundaki genlik tersleme bir saat sinyali gecikme ile yapılmalıdır. Gecikme dikkate alınmayan kuadrant sıkıştırması ile elde edilen sinüs Şekil 5.4'te, gecikme dikkate alınarak kuadrant sıkıştırması ile elde edilen sinüs ise Şekil 5.5'te verilmiştir.



Şekil 5.4. ROM gecikmesi dikkate alınmayan kuadrant sıkıştırma



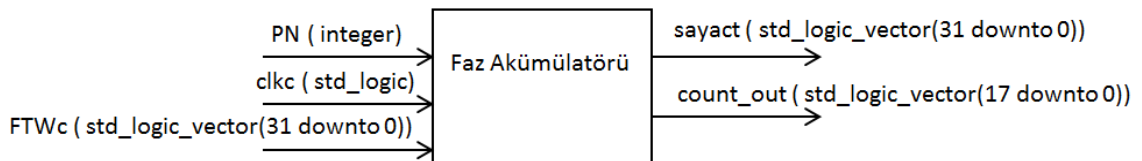
Şekil 5.5. ROM gecikmesi dikkate alınan kuadrant sıkıştırma

#### 5.1.4. Tasarım 4

Bu tasarımda faz çözünürlüğü değişken tutulmuştur. Kullanıcı tarafından 18-32 arasında faz çözünürlüğü girilir.  $N$  faz çözünürlüğü olmak üzere, sayıcı 0 –  $2^N$  arası sayar. Sayıcı çıkışı kırpılarak 18 bitlik kısmı tabloları adreslemek için kullanılır. En önemli 10 bit ile ilk değer tablosu adreslenirken en önemli 3 bit ve en önemsiz 8 bit ile ofset tablosu adreslenir. Değişken faz çözünürlüklü bu tasarımın faz akümülatörü giriş ve çıkışları Çizelge 5.1’de, genel yapısı Şekil 5.6’da, VHDL kodu ise Şekil 5.7de verilmiştir.

Çizelge 5.1. Faz Çözünürlüğü değişken faz akümülatörün giriş ve çıkışları

|          | Değişken adı | Açıklama                                                                                                                          |
|----------|--------------|-----------------------------------------------------------------------------------------------------------------------------------|
| Girişler | PN           | <i>integer</i> tipindedir. Yapının faz çözünürlüğünü belirlemektedir. Kullanıcı tarafından 18-32 arasında değiştirilebilmektedir. |
|          | clk          | Faz akümülatörü saat sinyalıdır. <i>std_logic</i> tipindedir.                                                                     |
|          | FTWc         | Frekans ayarlama sözcüğüdür. <i>std_logic_vector</i> tipindedir. Faz akümülatörü adım sayısını belirlemektedir.                   |
| Çıkışlar | count_out    | Kırpılmış faz sözcüğünü ifade etmektedir. <i>std_logic_vector</i> tipindedir.                                                     |
|          | sayact       | Faz akümülatörünün her saat sinyaliyle artan sayaç değeridir. <i>std_logic_vector</i> tipindedir.                                 |



Şekil 5.6. Faz Çözünürlüğü değişken faz akümülatörü genel yapısı

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.numeric_std.all;
entity counter is
    port(
        PN : in integer:=32;
        FTWc:in STD_LOGIC_VECTOR ( 31 downto 0 );
        clk : in STD_LOGIC;
        sayact: out STD_LOGIC_VECTOR ( 31 downto 0 );
        count_out : out STD_LOGIC_VECTOR ( 17 downto 0 )
    );
end counter;

architecture Davranis of counter is
    signal max:STD_LOGIC_VECTOR(33 downto 0):=(others=>'0');
    signal FTW:STD_LOGIC_VECTOR ( 31 downto 0 );
    signal sayac:std_logic_vector(31 downto 0):=(others=>'0');
begin
    FTW<=FTWc;
    process(clk)
        variable pr:integer:=32;
    begin
        pr:=PN;
        max<=(others=>'0');
        max(pr)<='1';
        if clk='1' and clk'event then
            if sayac+FTW < max then
                sayac<=sayac+FTW;
            else
                sayac<= (others=>'0');
            end if;
        end if;
        count_out<=sayac(pr-1 downto pr-18);
    end process;
    sayact<=sayac;
end Davranis;

```

Şekil 5.7. Faz çözünürlüğü değişken faz akümülatörü VHDL kodu

Bu tasarımda, sinüs, kare, testere dişi ve üçgen dalga formları elde edilebilmektedir. Dalga formu seçimi kullanıcı tarafından gönderilen 2 bitlik *formsec* bilgisi ile gerçekleştirilir. Diğer tasarımların faz akümülatörlerinden farklı olarak bu tasarımın faz akümülatöründe *sayact* çıkışı bulunmaktadır. Bu çıkışın en önemli 16 biti testere dişi, üçgen ve kare dalga üretimi için kullanılmaktadır. Dalga formlarının nasıl üretildiği Bölüm 5.1.6’da anlatılacaktır.

#### 5.1.4.1. BTM parametrelerinin (a, b, c) belirlenmesi

Kırılmış faz sözcüğü bit sayısı olan  $Q$  değeri 18 olan tasarım 1 ve tasarım 4 için  $a$ ,  $b$  ve  $c$  parametreleri sırasıyla 10, 8 ve 3; tasarım 3 için ise 8, 8 ve 3 seçilmiştir.  $Q$  değeri 12 olan tasarım 2 için ise  $a$ ,  $b$  ve  $c$  parametreleri 7, 5 ve 3 seçilmiştir.  $Q$

değerinin  $a$ ,  $b$  ve  $c$  değerlerine nasıl ayrıştığı, tasarlanan yapının spektral saflığına ve ROM boyutuna doğrudan etki etmektedir. TIV genişliği  $W_{TIV}$ , TO genişliği  $W_{TO}$  olmak üzere ROM boyutu Eşitlik (5.1)'deki gibi hesaplanırken; De Caro ve arkadaşları (2008)  $a$ ,  $b$  ve  $c$  değerlerine bağlı olarak SFDR değerini Eşitlik (5.2)'deki gibi formülize etmiştir.

$$ROM \text{ Boyutu} = W_{TIV} \cdot 2^a + W_{TO} \cdot 2^{b+c} \quad (5.1)$$

$$SFDR \approx \frac{16(2^{a-c}-1)2^{2a}}{2^{2(a-c)}} \quad (5.2)$$

BTM parametrelerinin belirlenmesi çok amaçlı bir optimizasyon problemi olarak ele alınabilir. Yapılan çalışmalarda genellikle hedef bir SFDR değeri belirlenerek, bu değeri sağlayan minimum ROM boyutunu bulmak üzere problem tek bir amaca indirgenmiştir (De Dinechin ve Tisserand, 2005).

Bu çalışmada tasarlanan yapılarda kırılmış faz sözcüğü 18 bit ve 12 olarak seçilmiş ve parametrelerin belirlenmesinde Matlab R2016a yazılımı ile oluşturulan ve Şekil 5.8'de verilen kod parçası kullanılmıştır.

```

R=16; %% genlik çözünürlüğü
Q=18; %% kırılmış faz sözcüğü
a_max=Q-2;
a_min=round(Q/2);
desired_SFDR=100; %% hedeflenen SFDR değeri
parameters=[R Q 0 0]; %% [R Q a c]
c_min=3;
min_size=R*2^Q;
%%her (a,c) çifti için SFDR ve toplam tablo boyutunun hesaplanması
for i=a_min:a_max
    parameters(3)=i;
    for j=c_min:parameters(3)
        parameters(4)=j;
        [SFDRx,size]=calculations(parameters);
        if SFDRx>=desired_SFDR
            if size<min_size
                min_size=size;
                best_decomposition=parameters;
                obtained_SFDR=SFDRx;
            end
        end
    end
end
end

```

Şekil 5.8. BTM parametrelerinin belirlenmesinde kullanılan kod parçası

Bu kod parçası, her bir  $a, c$  çifti için Eşitlik (5.2)'de verilen eşitliği kullanarak SFDR değerini hesaplamaktadır. Hedef olarak belirlenen SFDR değerinin üzerinde SFDR sağlayan her  $a, c$  çifti için tabloların boyutlarını Eşitlik (5.1)'de verilen eşitliği kullanarak hesaplamaktadır. Hesaplanan bu boyutlardan en küçüğünü sağlayan  $a, b, c$  değerlerini ise  $Q$  değerinin en iyi ayrışması olarak belirlemektedir. Bu kod parçası ile hedef SFDR 100 dBc seçildiğinde  $a, b, c$  parametreleri sırasıyla 10, 8 ve 3 olarak belirlenmiştir. Gerçeklenen sistemde 71.17 maksimum SFDR elde edilmiştir. Kırpılmış faz sözcüğü 12 ve hedef SFDR 70 dBc seçilerek belirlenen 7, 5, 3 değerleri ile ROM boyutu 12,8 kat daha küçültülerek 64.69 dBc değeri elde edilmiştir. Tasarımların LUT boyutları Çizelge 5.2'de verilmiştir.

**Çizelge 5.2.** LUT Boyutları

| <i>Çalışma</i> | <i>a, b, c</i> | Ofset tablosu  |                | İlk değer tablosu |                |
|----------------|----------------|----------------|----------------|-------------------|----------------|
|                |                | Genişlik (Bit) | Derinlik (Bit) | Genişlik (Bit)    | Derinlik (Bit) |
| Tasarım 1 ve 4 | 10, 8, 3       | 8              | 2048           | 16                | 1024           |
| Tasarım 2      | 7, 5, 3        | 5              | 256            | 10                | 128            |
| Tasarım 3      | 8, 8, 3        | 8              | 2048           | 15                | 256            |

#### 5.1.4.2. LUT İçeriklerinin Hesaplanması

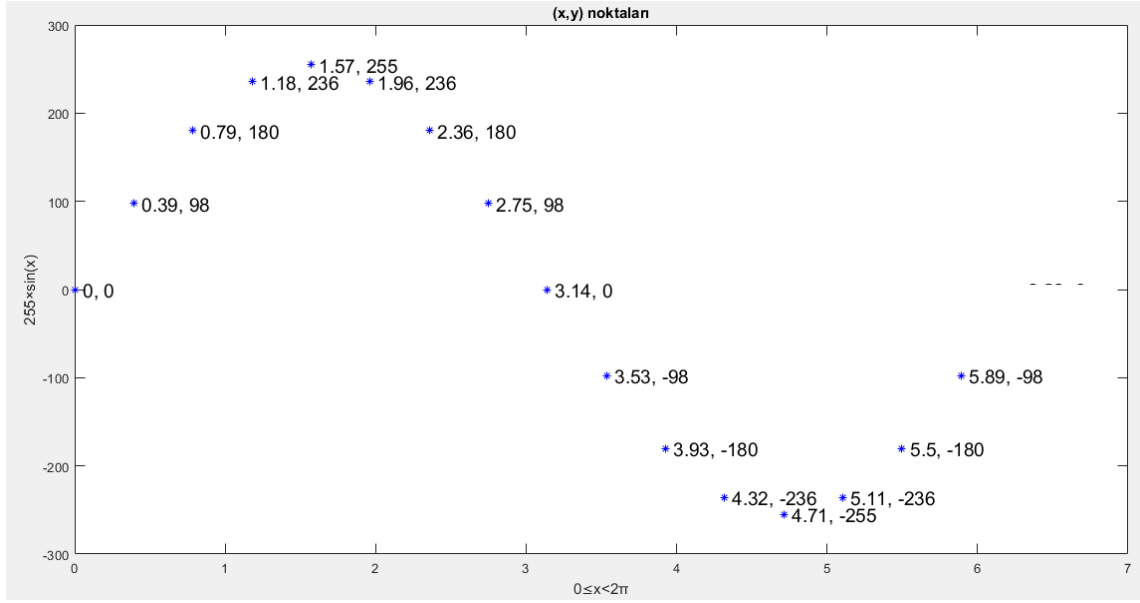
BTM ile fonksiyon hesaplanırken, öncelikle  $x$  eksen  $2^a$  eşit parçaya bölünür. Her  $x$  noktası için  $y = f(x)$  değeri hesaplanır. BTM tekniği ile kuadrant sıkıştırma tekniği birleştirilmek istenirse, hesaplanacak fonksiyonun çeyrek periyoduna ait değerler ilk değer tablosu için yeterlidir. Genlik çözünürlüğü  $R$ ,  $x$  eksenindeki artış miktarı  $x_{inc}$  olmak üzere,  $y = \sin(x)$  fonksiyonunun bir periyodu için  $x_1 = 0$  ve  $y_1 = 0$  olmak üzere  $(x, y)$  değerleri aşağıdaki eşitlikler ile hesaplanır.

$$x_{inc} = \frac{2\pi}{2^a} \quad (5.3)$$

$$x_{i+1} = x_i + x_{inc} \quad i = 1 \dots 2^a \quad (5.4)$$

$$y_{i+1} = (2^{R-1} - 1) \times \sin(x_{i+1}) \quad i = 1 \dots 2^a \quad (5.5)$$

$a = 4$  ve  $R = 9$  için hesaplanan örnek  $(x, y)$  değerleri Şekil 5.9'da verilmiştir.



Şekil 5.9.  $a=4$  ve  $R=9$  için hesaplanan  $(x, y)$  değerleri

Şekil 5.9'da görüldüğü üzere  $x$  eksenini 16 eşit parçaya ayrılmıştır. Her parça için eğim bilgisi Eşitlik (5.6) kullanılarak hesaplanır.

$$m_i = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \quad i = 1 \dots 2^a \quad (5.6)$$

İlk değer tablosu için  $2^a$  eşit parçaya ayrılan  $x$  eksenini,  $c < a$  olmak üzere ofset tablosu için de  $2^c$  aralığa ayırılır.  $K = 2^{a-c}$  olmak üzere her  $K$  komşu nokta için aynı eğim bilgisi  $M_i$  kullanılarak  $f_{app}(x)$  hesaplanır. Bu çalışmada  $M_i$  değerleri Eşitlik (5.7) kullanılarak hesaplanmıştır.

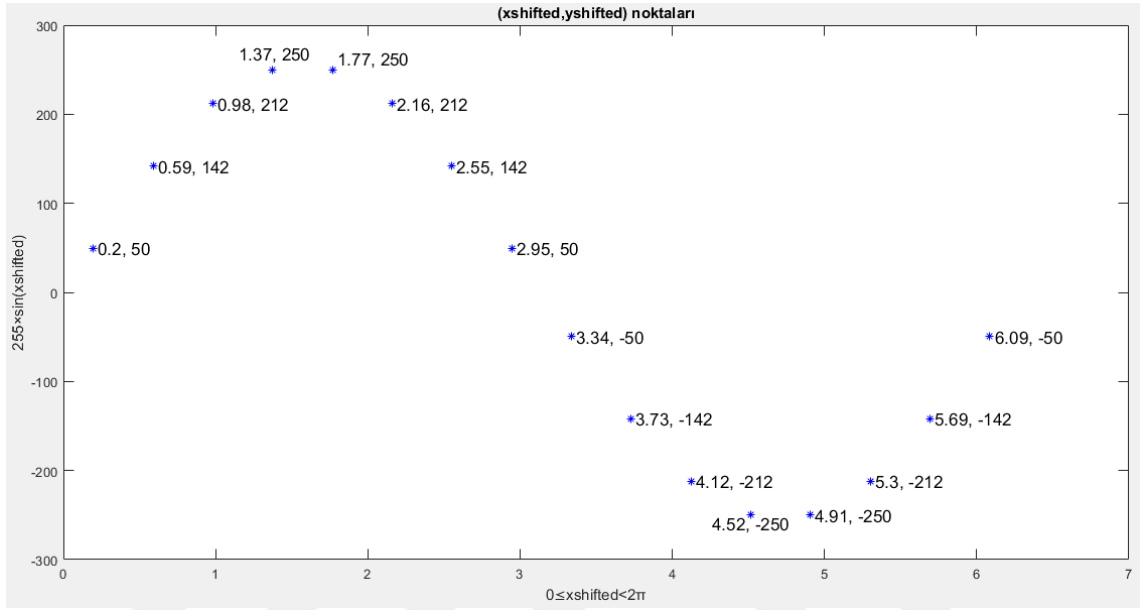
$$M_i = \frac{\sum_{j=1}^{2^{a-c}} m_{(i-1)K+j}}{2^c} \quad i = 1, 2 \dots 2^c \quad (5.7)$$

$a = 4, c = 2$  ve  $R = 9$  için hesaplanan  $m$  ve  $M$  değerleri Çizelge 5.3'te verilmiştir.

Çizelge 5.3.  $a = 4, c = 2$  ve  $R = 9$  için hesaplanan  $m$  ve  $M$  değerleri

| $i$ | $m_{(i-1)K+1}$ | $m_{(i-1)K+2}$ | $m_{(i-1)K+3}$ | $m_{(i-1)K+4}$ | $M_i$   |
|-----|----------------|----------------|----------------|----------------|---------|
| 1   | 123.504        | 106.952        | 68.755         | 25.465         | 81,169  |
| 2   | -25,465        | -68,755        | -106,952       | -122,231       | -80,851 |
| 3   | -124,777       | -106,952       | -68,755        | -25,465        | -81,487 |
| 4   | 25,465         | 68,755         | 106,952        | 124,778        | 81,487  |

Belirlenen  $x$  noktaları  $x_{inc}/2$  kadar sağa ötelenerek ilk değer tablosunda kullanılmak üzere  $xshifted_1 = x_{inc}/2$  ve  $yshifted_1 = (2^R - 1) \times \sin(xshifted_1)$  olmak üzere  $(xshifted, yshifted)$  noktaları Eşitlik (5.3), Eşitlik (5.4) ve Eşitlik (5.5) kullanılarak hesaplanır.  $a = 4$  ve  $R = 9$  için hesaplanan  $(xshifted, yshifted)$  noktaları Şekil 5.10'da verilmiştir.

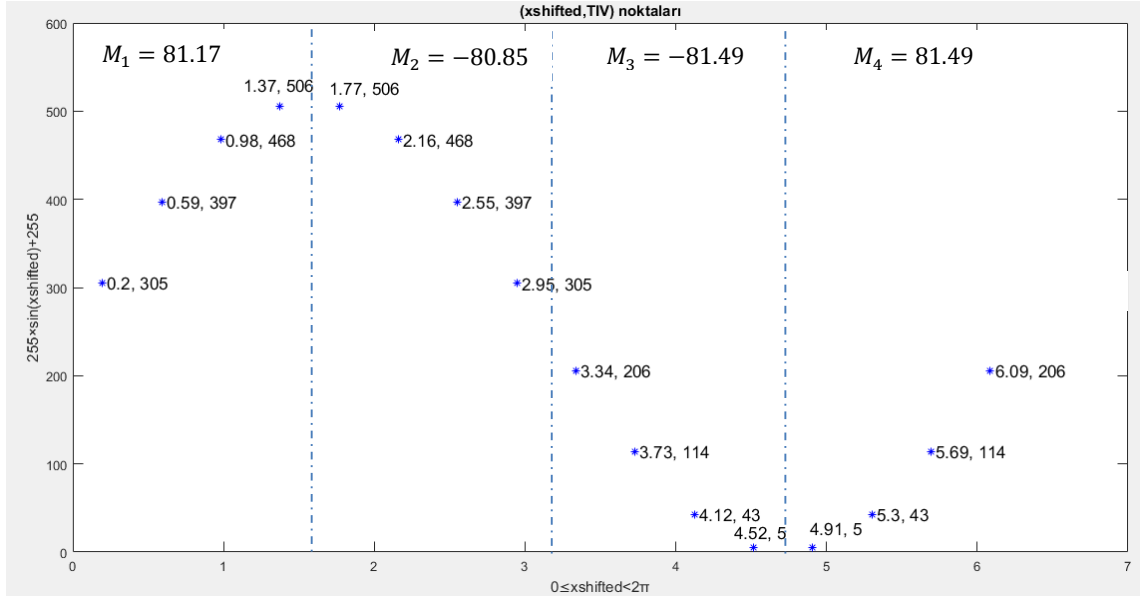


Şekil 5.10.  $a=4$  ve  $R=9$  için hesaplanan  $(xshifted, yshifted)$  değerleri

İlk değer tablosu içeriğinin yalnızca pozitif sayılardan oluşması istenirse, içerik Eşitlik (5.8) ile hesaplanır.

$$TIV_{i+1} = (2^{R-1} - 1) + (2^{R-1} - 1) \times \sin(xshifted_{i+1}) \quad i = 1, 2, \dots, 2^a \quad (5.8)$$

$a = 4$  ve  $R = 9$  için hesaplanan ilk değer tablosu değerleri Şekil 5.11'de verilmiştir.



Şekil 5.11.  $a=4$  ve  $R=9$  için hesaplanmış ilk değer tablosu içerikleri

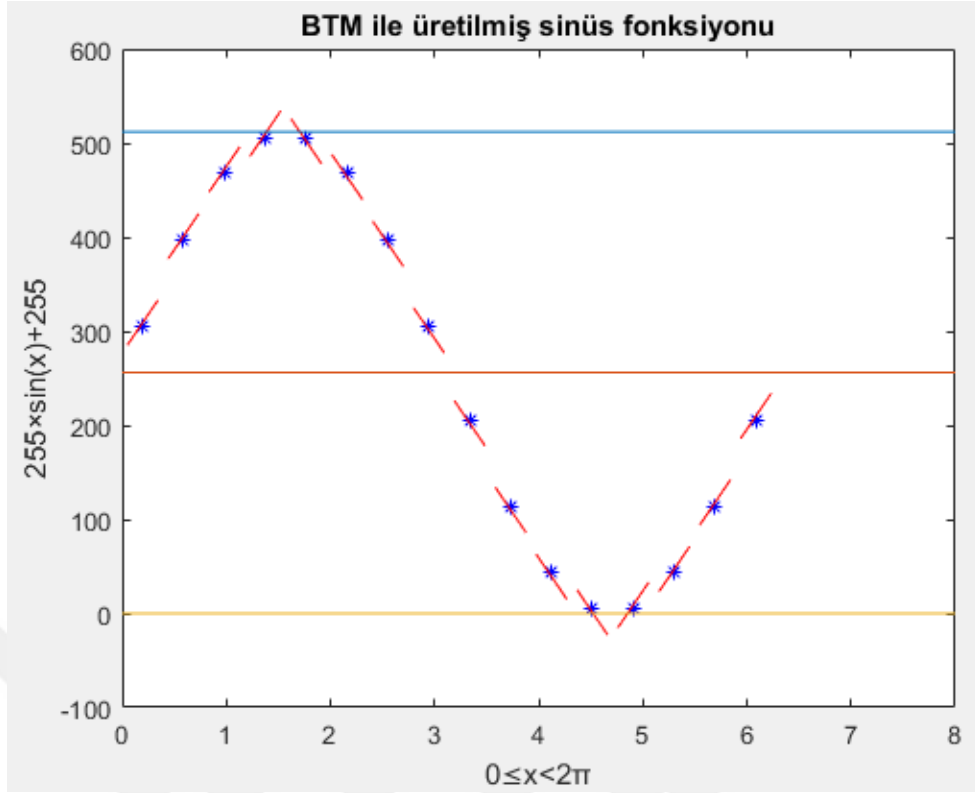
$Q$  kırılmış faz sözcüğünün bit sayısı,  $b=Q-a$  olmak üzere her TIV değeri için  $2^b$  ofset değeri hesaplanır.  $x_{TOinc} = \frac{x_{inc}}{2^b}$  olmak üzere ofset tablo içeriği Eşitlik (5.9) ile hesaplanır.

$$TO_{(i-1)K+j} = -\frac{M_i \times x_{inc}}{2} + M_i \times j \times x_{TOinc} \quad i = 1, 2 \dots 2^c, \quad j = 1, 2 \dots 2^b \quad (5.9)$$

$a=4$ ,  $b=2$ ,  $c=2$  ve  $R=9$  için oluşturulan TIV ve TO içerikleri Çizelge 5.4'te, BTM ile elde edilen sinüs eğrisi Şekil 5.12'de verilmiştir.

Çizelge 5.4.  $a=4$ ,  $b=2$ ,  $c=2$  ve  $R=9$  için oluşturulan TIV ve TO içerikleri

|                           |     | TIV içeriği |     |     |     | TO içeriği |     |     |  |
|---------------------------|-----|-------------|-----|-----|-----|------------|-----|-----|--|
| $a=4, b=2,$<br>$c=2, R=9$ | 305 | 397         | 468 | 506 | -16 | 0          | 16  | 32  |  |
|                           | 506 | 468         | 397 | 305 | 16  | 0          | -16 | -32 |  |
|                           | 206 | 114         | 43  | 5   | 16  | 0          | -16 | -32 |  |
|                           | 5   | 43          | 114 | 206 | -16 | 0          | 16  | 32  |  |



Şekil 5.12. BTM ile elde edilen sinüs eğrisi ( $a=4$ ,  $b=2$ ,  $c=2$  ve  $R=9$ )

Şekil 5.12’de görüldüğü üzere üretilen sinüs eğrisi, tepe noktalarında maksimum genlik değerlerini aşmaktadır. Bu sorunu çözmek için bu çalışmada TIV değerleri Eşitlik (5.10), Eşitlik (5.11), Eşitlik (5.12) ve Eşitlik (5.13) yardımı ile normalize edilmesi önerilmiştir. Normalize edilmiş TIV ile hesaplanan sinüs eğrisi Şekil 5.13’te verilmiştir.

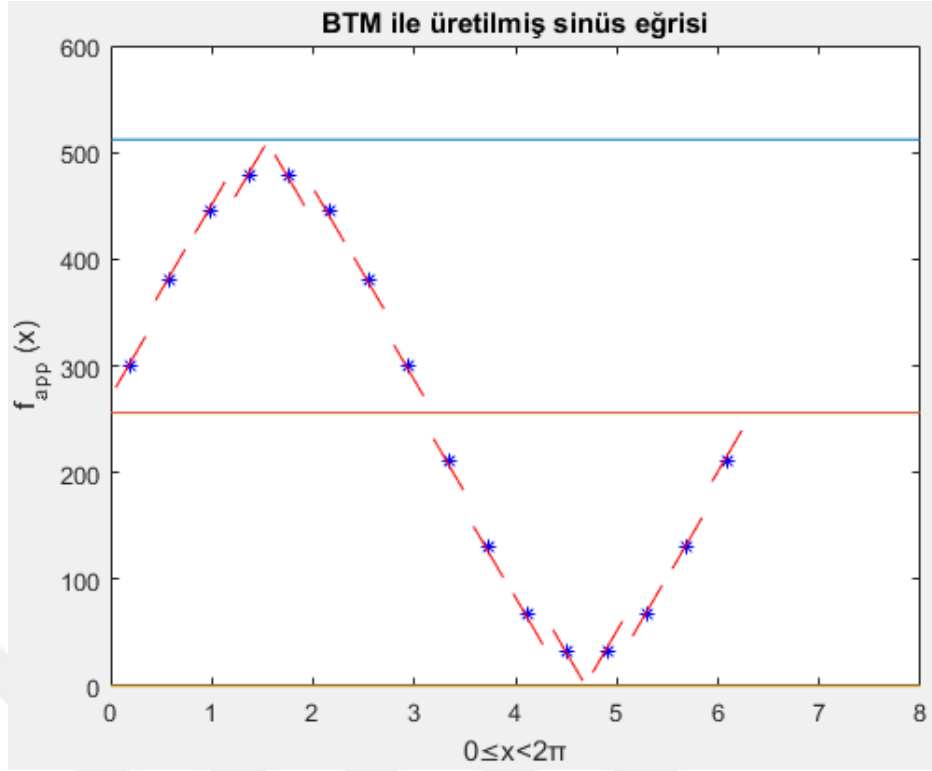
$$M_{tepe} = M_t \quad t = 2^{c-2} \quad (5.10)$$

$$fazlalık = M_{tepe} \times x_{inc} \quad (5.11)$$

$$TIV' = TIV + \frac{fazlalık}{2} \quad (5.12)$$

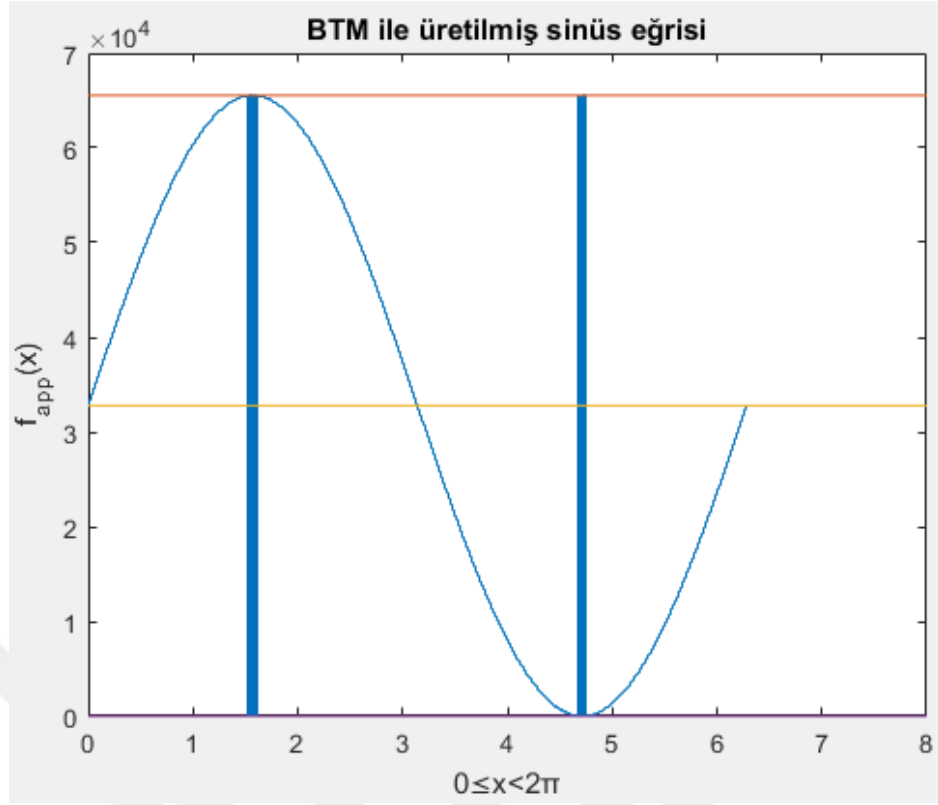
$$TIV'' = TIV' \times \frac{(2^R - 1)}{(2^R - 1 + fazlalık)} \quad (5.13)$$

$t = 2^{c-2}$  değeri elde edilecek sinüs eğrisinin tepe noktadaki eğimini belirlemek için klasik BTM tabanlı tasarımda kullanılmaktadır. BTM ve kuadrant sıkıştırma tekniği birlikte kullanılmak istenirse bu değer  $t = 2^c$  olacaktır.

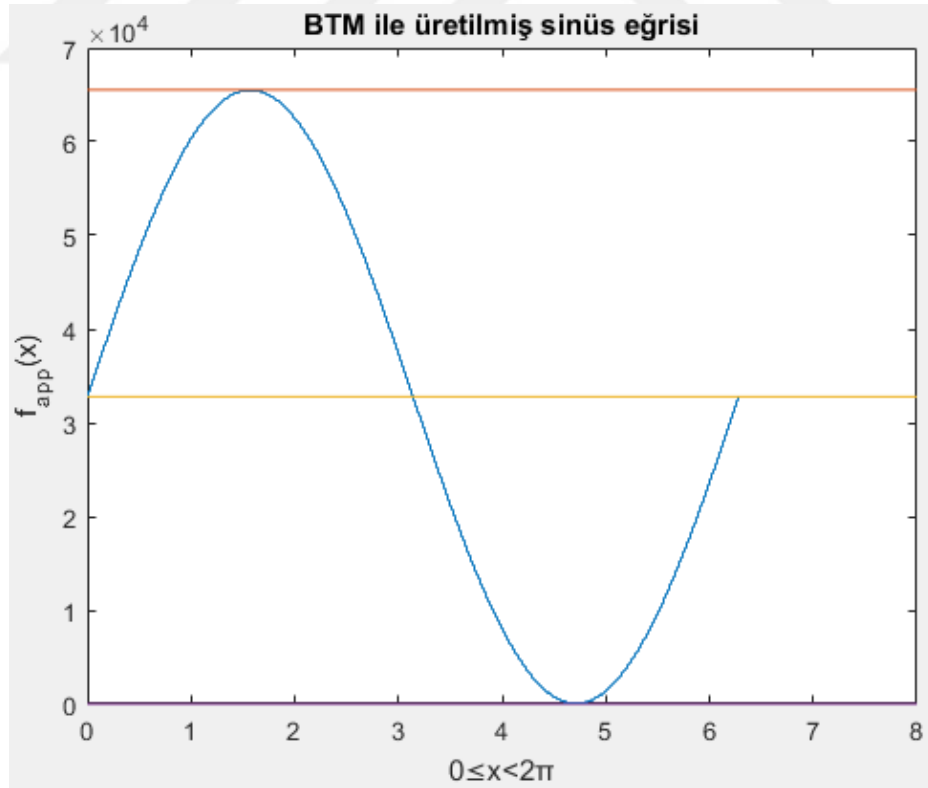


Şekil 5.13. Normalize edilmiş TIV kullanılarak elde edilen sinüs eğrisi ( $a=4$ ,  $b=2$ ,  $c=2$  ve  $R=9$  )

$a=10$ ,  $b=8$ ,  $c=3$  ve  $R=16$  için normalize edilmemiş TIV ile elde edilen sinüs dalgasının 16 bit çözünürlüklü bir DAC kullanılarak analoğa dönüştürülmüş hali Şekil 5.14'te, normalize edilmiş TIV ile elde edilen sinüs eğrisinin analog hali Şekil 5.15'te verilmiştir.



Şekil 5.14. Normalize edilmemiş TIV ile elde edilen analog sinüs eğrisi ( $a=10$ ,  $b=8$ ,  $c=3$  ve  $R=16$ )



Şekil 5.15. Normalize edilmiş TIV ile elde edilen analog sinüs eğrisi ( $a=10$ ,  $b=8$ ,  $c=3$  ve  $R=16$ )

### 5.1.5. UART alıcı modülü ve MATLAB GUI ile DDFS arayüzü

Bu çalışmada, frekans ayarlama sayısal sözcüğü MATLAB GUI (Graphical User Interfaces) ile oluşturulan arayüzler ile UART haberleşme protokolü aracılığıyla kullanıcıdan alınmıştır.

#### 5.1.5.1. Sabit faz çözünürlüklü DDFS arayüzü

Faz çözünürlüğü sabit olan tasarımlar için MATLAB GUI aracılığı ile hazırlanan arayüz Şekil 5.16'da verilmiştir.

Şekil 5.16. Sabit Faz Çözünürlüklü DDFS Arayüzü

Hazırlanan arayüz, kullanıcı tarafından girilen giriş frekansı, istenilen çıkış frekansı ve faz çözünürlüğü değerlerini kullanılarak “hesapla” butonuna basılmasıyla sayısal frekans ayarlama sözcüğü ve elde edilen gerçek çıkış frekansını hesaplar. Hesaplanan frekans ayarlama sayısal sözcüğü, “Gönder” butonu ile UART haberleşmesi kullanılarak birer bit start ve stop bitleri ile birlikte 10 bitlik paketler halinde 4800 baud hızı ile gönderilir. 10 MHz giriş frekansı, 1 kHz çıkış frekansı ve 32 bitlik faz çözünürlüğü ile arayüz çalıştırıldığında hesaplanan çıkışlar Şekil 5.17’de verilmiştir.

**Şekil 5.17.** 10 MHz giriş frekansı, 1 kHz çıkış frekansı ve 32 bitlik faz çözünürlüğü ile çalıştırılmış sabit faz çözünürlüklü DDFS arayüzü

### 5.1.5.2. Değiştirilebilir faz çözünürlüklü DDFS arayüzü

Faz çözünürlüğü değiştirilebilir olan, farklı sinyal formlarını sentezleyebilen DDS tasarımları için MATLAB GUI aracılığı ile hazırlanan arayüz Şekil 5.18’de verilmiştir.

**Şekil 5.18.** Değiştirilebilir faz çözünürlüklü ve farklı sinyal formlarını sentezleyebilen DDS arayüzü

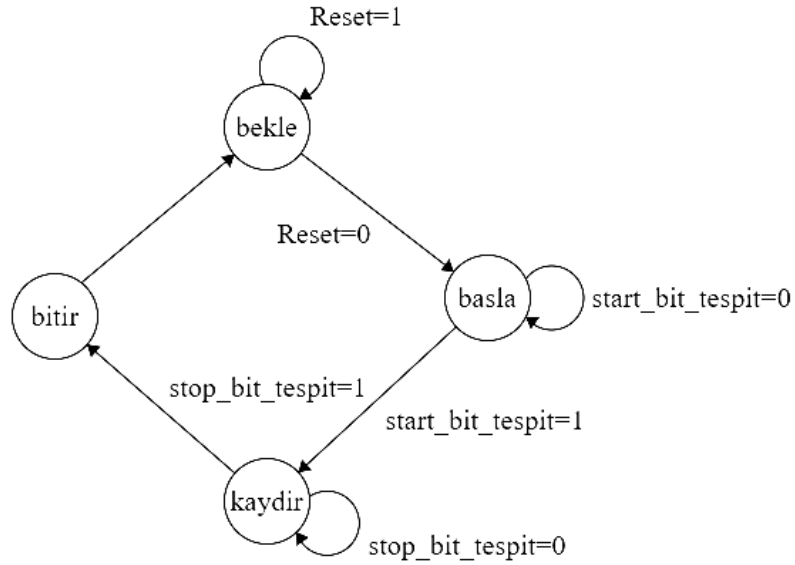
Hazırlanan arayüz, “Hesapla” butonuna basılmasıyla kullanıcı tarafından girilen giriş frekansı ve istenilen çıkış frekansı ile birlikte 18-32 arasında seçilen faz çözünürlüğü değerini kullanarak sayısal frekans ayarlama sözcüğünü ve elde edilen çıkış frekansını hesaplamaktadır. Ayrıca seçilen sinyal formu için 16’lık sayı tabanında oluşturulan kod ile birlikte faz çözünürlüğü değerinin 16’lık tabanındaki değerini birleştirerek ikincil bir kod oluşturmaktadır. “Gönder” butonuna basılmasıyla frekans ayarlama sayısal sözcüğü ile birlikte oluşturulan ikincil kod 10 bitlik paketler halinde

4800 baud hızı ile gönderilmektedir. Kare dalga seçimi, 10 MHz giriş frekansı, 1 kHz çıkış frekansı ve 32 bitlik faz çözünürlüğü ile arayüz çalıştırıldığında hesaplanan çıkışlar Şekil 5.19'da verilmiştir.

**Şekil 5.19.** Kare dalga seçimi, 10 MHz giriş frekansı, 1 kHz çıkış frekansı ve 32 bitlik faz çözünürlüğü ile çalıştırılmış DDS arayüzü

### 5.1.5.3. VHDL ile UART alıcı modül tasarımı

MATLAB GUI aracılığı ile oluşturulmuş arayüz ile gönderilen verileri almak için VHDL ile bir UART alıcı modülü tasarlanmıştır. Tasarlanan modülün sonlu durum makinesi Şekil 5.20'de verilmiştir.



Şekil 5.20. UART alıcı modül sonlu durum makinesi

**bekle:** Önerilen sonlu durum makinesinde, herhangi bir durumda iken  $Reset = '1'$  olması durumunda *bekle* durumuna geçilmektedir. Bu durumda iken  $Reset = '0'$  olması durumunda ise *basla* durumuna geçilmektedir.

**basla:** Bu durumda iken seri bilgi girişi olan *giriş* değişkeninin '1' seviyesinden '0' seviyesine geçmesi yani *start* bitinin tespit edilmesi beklenmektedir. *Start* bitinin tespit edilmesiyle *kaydir* durumuna geçilmektedir.

**kaydir:** Bu durum verinin alındığı durumdur. Transfer edilmesi beklenen toplam bit sayısı  $N$  olmak üzere, gelen toplam bit sayısı  $S$  olana kadar veri alma işlemi devam etmektedir.  $S$  değeri aşağıdaki eşitlik kullanılarak hesaplanmaktadır.

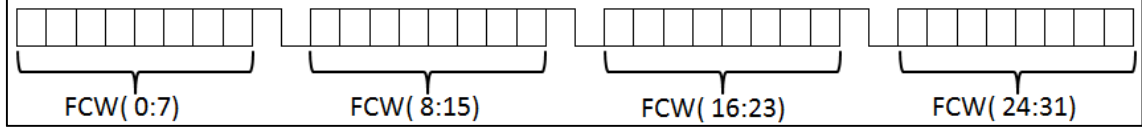
$$S = N - 1 + \left(\frac{N}{8} - 1\right) \times 2 \quad (5.14)$$

Gelen bit sayısı  $S$  değerine ulaştığında, başlangıç seviyesi '0' olan *stop\_bit\_tespit* değişkeni seviyesi '1' olarak değiştirilir. *stop\_bit\_tespit* seviyesi '0' olduğu sürece bu durumda beklenirken '1' olduğunda ise *bitir* durumuna geçilir.

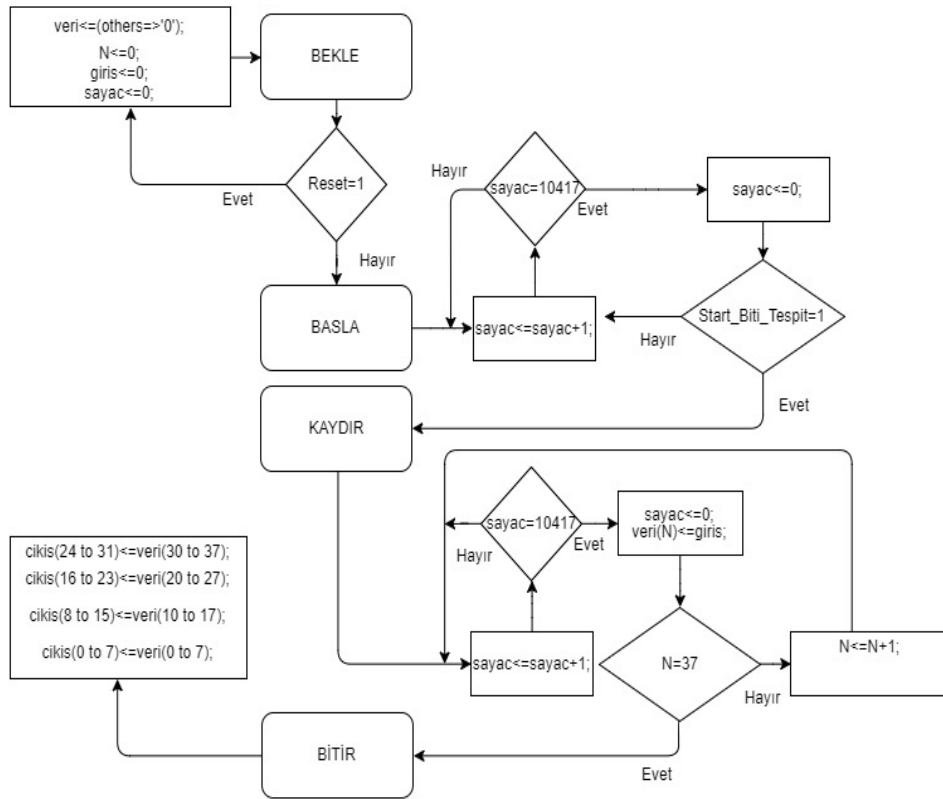
**bitir:** Bu durum veri alışının sonlandığı durumdur. Bu durum sonrasında *bekle* durumuna geçilir.

Sabit faz çözünürlüklü DDFS arayüzünden gönderilen 32 FCW bilgisini almak için tasarlanan UART alıcı modülünde baud hızını 4800 olarak ayarlamak üzere 100 MHz olan sistem saati frekansı bir sayıcı kullanılarak bölünmüştür. 32 bitlik FCW bilgisi arayüzden 10 bitlik (8 bit veri, 1 bit start biti, 1bit stop biti) paketler halinde

gönderildiği için 38 bit olarak alınan verinin anlamlı 32 biti FCW bilgisi olarak belirlenmiştir. 38 bit olarak alınan verinin içerdiği 32 bitlik FCW bilgisi Şekil 5.21’de, tasarlanan sistemin akış diyagramı Şekil 5.22’de verilmiştir.



Şekil 5.21. Gelen veriden FCW bilgisinin tespit edilmesi



Şekil 5.22. UART alıcı modül tasarımı akış diyagramı

### 5.1.6. Sinüs, kare, testere dişi ve üçgen sinyal formlarının elde edilmesi

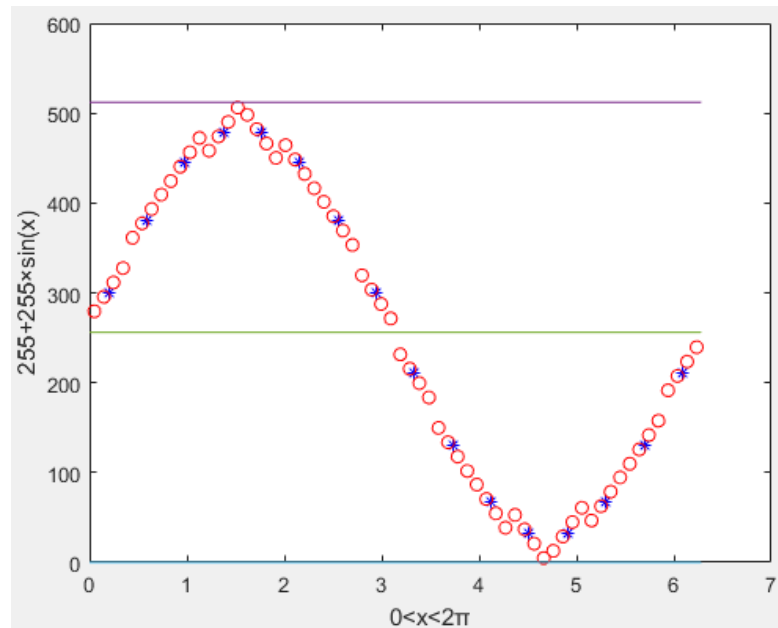
#### 5.1.6.1. Sinüs dalga formunun elde edilmesi

BTM tabanlı olarak tasarlanan DDFS yapısında örneklenmiş sinüs sinyali genlik değerleri ilk değer tablosu (TIV) ve ofset değerleri tablosu (TO) olarak kullanılan hafıza birimlerine (RAM/ROM) kaydedilir. Her saat sinyali ile tabloların adresleri

değiştirilerek adreslerdeki değerler çıkış olarak alınır ve toplanır. Elde edilen toplam sinüs sinyalinin vermektedir. Tabloların adresleri faz akümülatörü olarak kullanılan sayıcı ile değiştirilir. Sayıcı adım sayısı frekans ayarlama sözcüğü ile belirlenir.  $a=4$ ,  $b=2$ ,  $c=2$ ,  $R=9$  ve adım sayısı 1 olmak üzere 6 bitlik faz sözcüğünün değişimi ile elde edilen sinüs sinyalinin ilk çeyreğine ait tablo değerleri Çizelge 5.1’de, toplam değerleri ile elde edilen bir periyotluk sinüs eğrisi de Şekil 5.23’te verilmiştir.

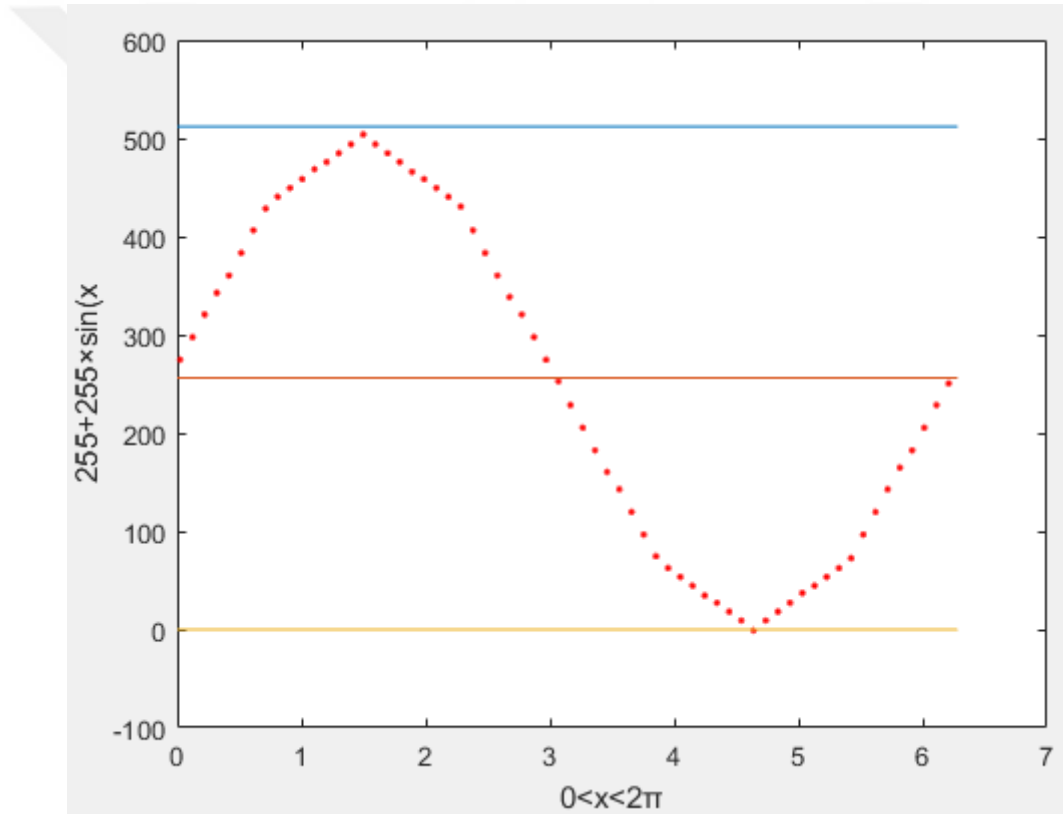
**Çizelge 5.5** Klasik BTM ile elde edilmiş sinüs sinyalinin çeyrek periyodunun değerleri ( $a=4, b=2, c=2, R=9$  ve adım sayısı=1)

| Faz (adres) sözcüğü | TIV adresi | TO adresi | TIV çıkışı | TO çıkışı | Toplam |
|---------------------|------------|-----------|------------|-----------|--------|
| 00000               | 0000       | 0000      | 305        | -16       | 289    |
| 00001               | 0000       | 0001      | 305        | 0         | 305    |
| 00010               | 0000       | 0010      | 305        | 16        | 321    |
| 00011               | 0000       | 0011      | 305        | 32        | 337    |
| 00100               | 0001       | 0000      | 397        | -16       | 381    |
| 00101               | 0001       | 0001      | 397        | 0         | 397    |
| 00110               | 0001       | 0010      | 397        | 16        | 413    |
| 00111               | 0001       | 0011      | 397        | 32        | 429    |
| 01000               | 0010       | 0000      | 468        | -16       | 452    |
| 01001               | 0010       | 0001      | 468        | 0         | 468    |
| 01010               | 0010       | 0010      | 468        | 16        | 484    |
| 01011               | 0010       | 0011      | 468        | 32        | 500    |
| 01100               | 0011       | 0000      | 506        | -16       | 490    |
| 01101               | 0011       | 0001      | 506        | 0         | 506    |
| 01110               | 0011       | 0010      | 506        | 16        | 522    |
| 01111               | 0011       | 0011      | 506        | 32        | 538    |
| ⋮                   | ⋮          | ⋮         | ⋮          | ⋮         | ⋮      |

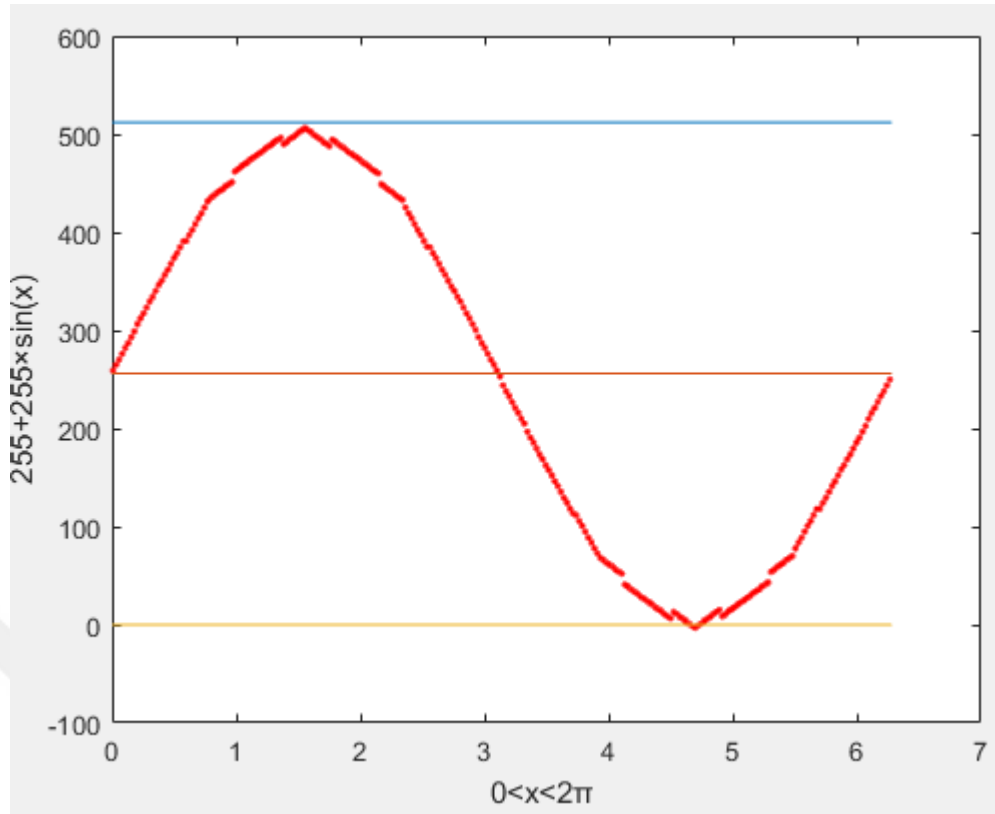


**Şekil 5.23.** BTM ile elde edilen bir periyotluk sinüs sinyali

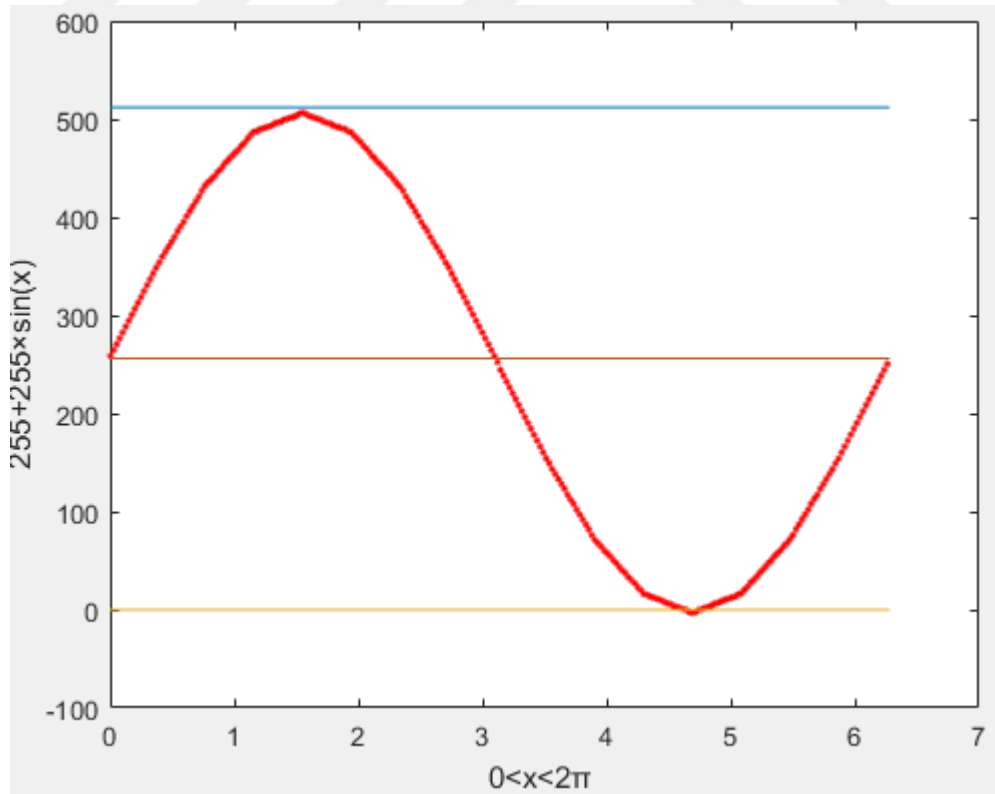
Şekil 5.23'te elde edilen sinyalde  $x$  eksenini 16 eşit parçaya ayrılmıştır. Bir tam periyot sinüs için ilk değer tablosuna kaydedilmek üzere alınan 16 örnek Şekil 5.23'te yıldız şeklinde çizdirilmiştir. Her parça için 4 adet ofset değeri kullanılmıştır. Bu değerler kullanılarak örnek sayısı 64'e çıkarılmıştır. Ofset eklenerek çoğaltılan değerler Şekil 5.23'te daire şeklinde çizdirilmiştir. Her 4 komşu ilk değer için aynı eğimli ofsetler kullanılarak oluşturulan sinyal sinüs sinyalinden oldukça uzaktır. Sinüs sinyaline daha benzer sinyaller elde etmek için aynı eğimi kullanan komşu sayısı azaltılmalı, ilk değer ve ofset değerleri sayıları ile tablo genişlikleri artırılmalıdır. Farklı  $a$ ,  $b$  ve  $c$  değerleri ile elde edilmiş sinüs sinyalleri Şekil 5.24-Şekil 5.28'de verilmiştir.



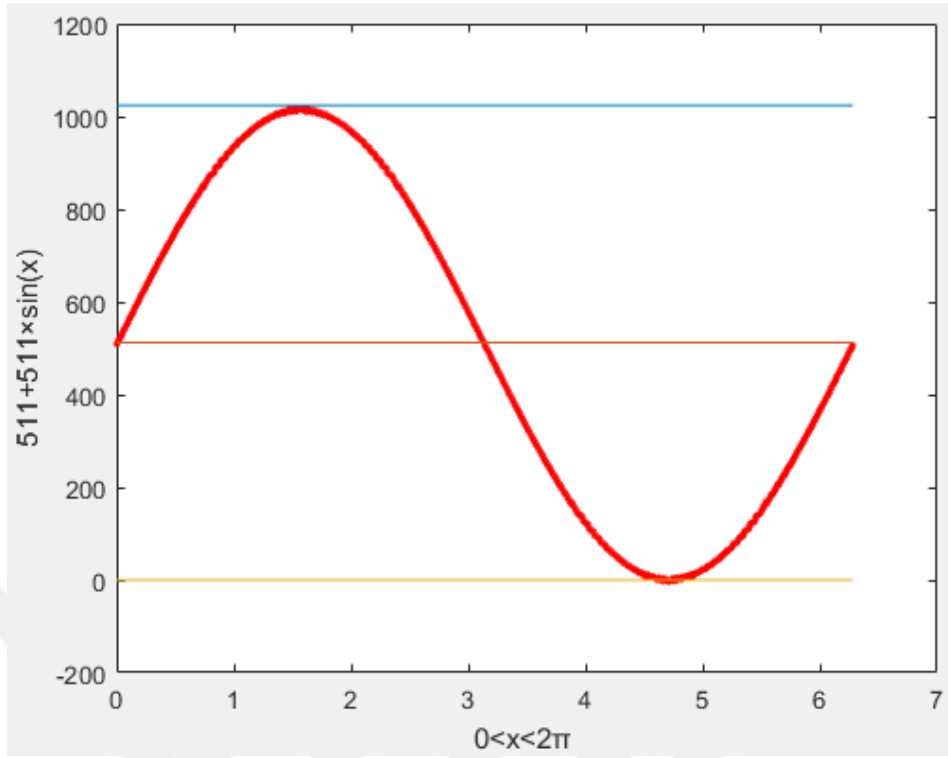
Şekil 5.24. BTM ile elde edilen sinüs sinyali( $a=4, b=2, c=3, R=9$ )



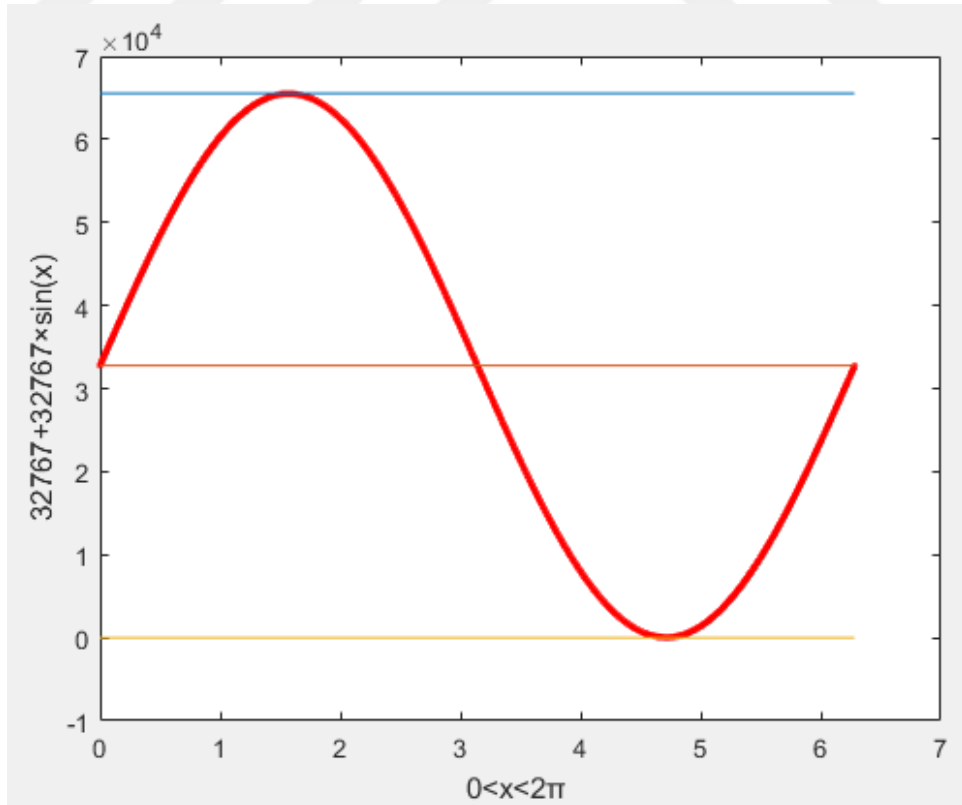
Şekil 5.25. BTM ile elde edilen sinüs sinyali ( $a=5, b=3, c=3, R=9$ )



Şekil 5.26. BTM ile elde edilen sinüs sinyali ( $a=5, b=3, c=4, R=9$ )



Şekil 5.27. BTM ile elde edilen sinüs sinyali ( $a=7, b=5, c=3, R=10$ )



Şekil 5.28. BTM ile elde edilen sinüs sinyali ( $a=10, b=8, c=3, R=16$ )

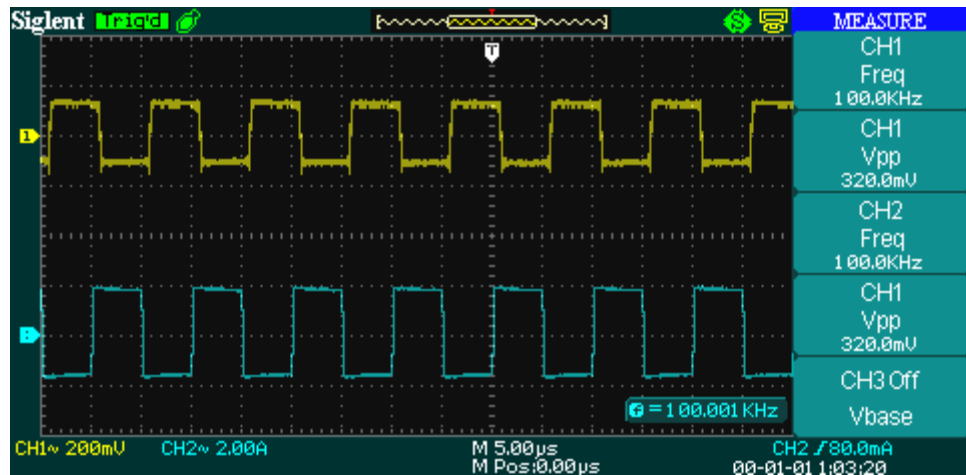
BTM ile birlikte kuadrant sıkıştırma tekniği kullanılmak istenirse sinüs sinyaline ait çeyrek periyotluk verinin tablolara kaydedilmesi yeterli olacaktır. Faz sözcüğünün ilk iki bitine göre sinüsün tamamının nasıl elde edileceği Şekil 4.4'te verilmiştir.

Kuadrant sıkıştırma tekniği kullanılırken dikkat edilmesi gereken en önemli hususlardan birisi kullanılan hafıza biriminin adres değişimine müteakiben kaç saat sinyali sonra çıkış verdiğidir. Örneğin, Artix7 FPGA ailesinde blok hafızalar adres değişimi ile çıkış değerini bir saat sinyali gecikme ile güncellemektedir.

#### 5.1.6.2. Kare dalga formunun elde edilmesi

DDS tabanlı sinyal üreteçlerinde kare dalga formu elde etmek kırpılmış faz sözcüğü sayıcısı kullanılmaktadır. Tabloları adreslemek için kullanılan kırpılmış faz sözcüğü sayıcısı saymaya başladığında kare dalga mantıksal seviyesi 0 ya da 1 olarak atanır. Sayıcının sayacağı tam değer yarısına gelindiğinde ise mantıksal seviye terslenir.

Bu çalışmada kare dalga çıkışı iki farklı şekilde alınmıştır. 16 bitlik kare dalga çıkışının sayısalan analog forma dönüştürülmesinde sinyalde bozulmalar gözlemlendiği için tek bitlik bir çıkış FPGA kartından alınarak kare dalga çıkışı elde edilmiştir. 16 bitlik ve 1 bitlik olarak alınan kare dalgalar arasındaki fark Şekil 5.29'da verilmiştir.



Şekil 5.29. 16 bit ve 1 bit ile elde edilen 100 kHz kare dalgalar

#### 5.1.6.3. Testere dişi dalga formunun elde edilmesi

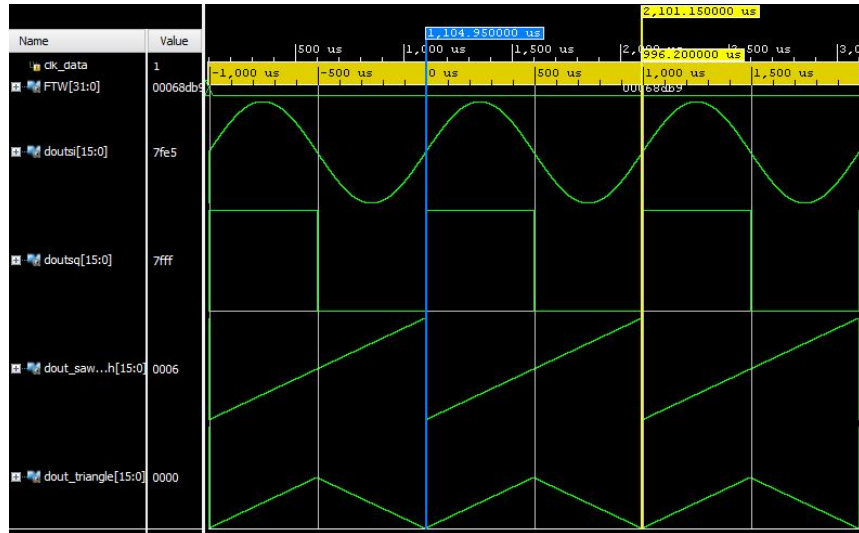
DDS tabanlı sinyal üreteçlerinde testere dişi dalga formu elde etmek için ilave hafıza birimine ihtiyaç duyulmamaktadır. Tabloları adreslemek için kullanılan kırpılmış faz sözcüğü, aynı zamanda bir sayıcı olduğu için testere dişi dalga formu çıkışı olarak doğrudan kullanılabilir.

#### 5.1.6.4. Üçgen dalga formunun elde edilmesi

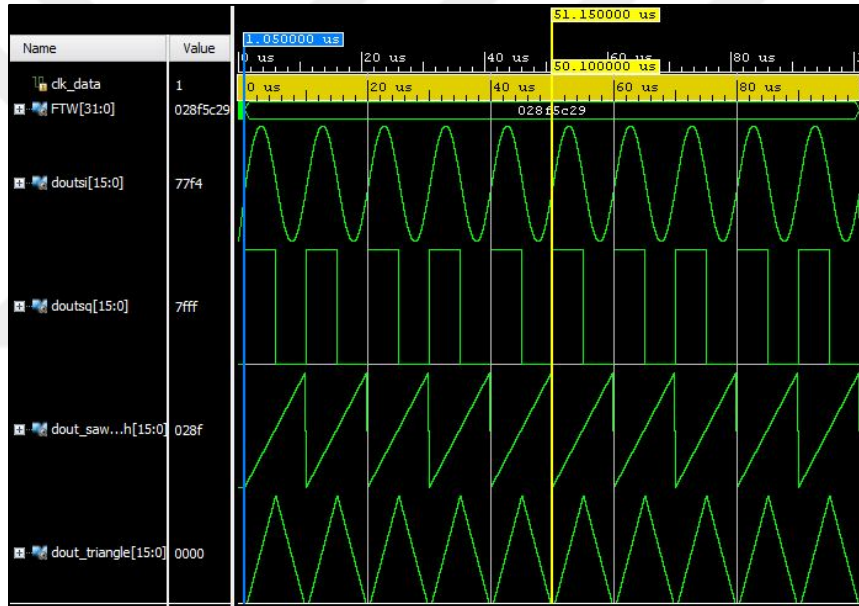
Üçgen dalga formu testere dişi dalga formundan yararlanılarak üretilir. Sayıcının yarı değere geldiği anda testere dişi sinyalinin tersi kullanılarak üçgen dalga elde edilir. Dalga formlarının elde edilmesi için kullanılan VHDL kodu Şekil 5.30'da, elde 1kHz, 100 kHz ve 1 MHz frekansında elde edilen sinyallerin simülasyonu da Şekil 5.31, Şekil 5.32 ve Şekil 5.33'de verilmiştir.

```
process( clk_veri)
begin
    if formsec="01" then
        dout<=doutkare;
    elsif formsec="10" then
        dout<=dout_testere;
    elsif formsec="11" then
        dout<=dout_ucgen;
    else
        dout<=doutsinus;
    end if;
    doutsinus<=dout_tiv+dout_to;
    dout_testere<=(adres(31 downto 16));
    if dout_testere(15)='0' then
        doutkare<="1111111111111111";
        dout_ucgen<=dout_testere;
    else
        doutkare<="0000000000000000";
        dout_ucgen<=not(dout_testere);
    end if;
end process;
```

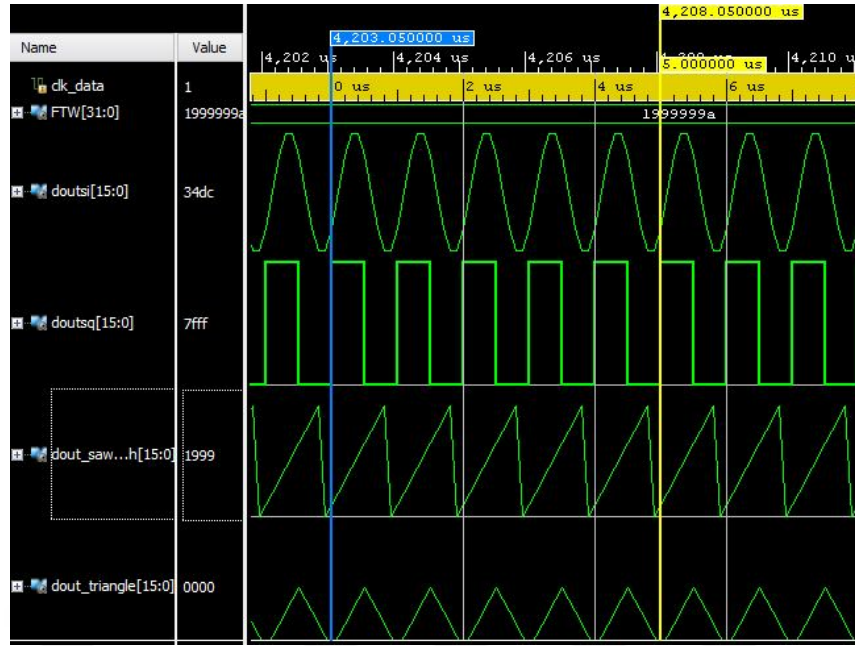
Şekil 5.30. Sinüs, kare, testere ve üçgen dalga formlarının elde edilmesi



Şekil 5.31. 1 kHz sinüs, kare, testere dişi ve üçgen dalga formları simülasyonları



Şekil 5.32. 100 kHz sinüs, kare, testere dişi ve üçgen dalga formları simülasyonları



Şekil 5.33. 1 MHz sinüs, kare, testere dişi ve üçgen dalga formları simülasyonları

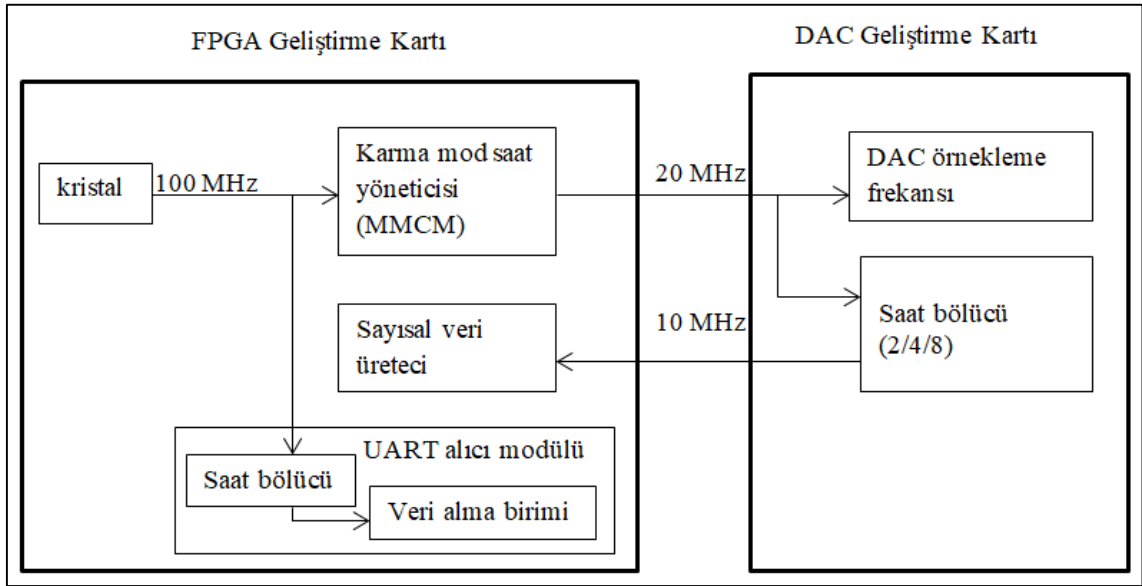
### 5.1.7. Sistem saat sinyali kullanımı

Tasarlanan DDFS sisteminde Nexys DDR™ geliştirme kartı üzerinde bulunan kristal osilatör temel frekans kaynağı olarak kullanılmıştır. Kart üzerinde bulunan PMOD konnektörleri yüksek frekansta kullanılmak üzere tasarlanmadığından 100 MHz kristal frekansının düşürülmeye ihtiyacı vardır. PMOD konnektörleri arayüz özellikleri ile ilgili detaylar (Anonymous, 2011) verilmiştir.

DAC örnekleme frekansı olarak kullanılmak amacıyla 100 MHz kristal osilatör FPGA içerisinde bulunan MMCM biriminde 20 MHz'e düşürülmüştür. DAC5687EVM geliştirme kartı örnekleme frekansında gönderilen saat sinyalinin frekansını 2, 4 veya 8'e bölüp, sayısal verilerin gönderimi için kullanılmak üzere faz uyumlu bir sinyal çıkışı sağlar. Tasarlanan DDFS sisteminin girişi olarak bu çıkış doğrudan kullanılabilceği gibi, gerekli olan frekansta saat sinyali karma mod saat yönetici birimi ile de üretilebilir. Bu çalışmada DAC kartından alınan çıkış kullanılmış olup, 2'ye bölme seçeneği kullanılarak DDFS frekans aralığının yüksek tutulması amaçlanmıştır. Ayrıca spektral saflığı iyileştirmek adına 4 ve 8'e bölme seçenekleri de denenmiş, sonuçları 6. Bölümde verilmiştir.

Kullanıcının bilgisayar üzerinden gireceği sayısal frekans ayarlama sözcüğünü almak üzere tasarlanan UART alıcı modülünde 100 MHz saat sinyali doğrudan

kullanılmıştır. Baud iletim hızı alıcı modül içerisinde kullanılan bir sayıcı ile ayarlanmıştır. Sistemin saat sinyali kullanımı Şekil 5.34’te verilmiştir.



Şekil 5.34. DDFS saat sinyali kullanımı

## 6. SONUÇLAR VE GERÇEKLENEN DDFS'NİN ÖZELLİKLERİ

### 6.1. Çözünürlük ve Frekans Aralığı

Gerçeklenen ilk 3 tasarımın faz çözünürlüğü 32 bit iken, son tasarımın faz çözünürlüğü 18-32 bit arasında değiştirilebilmektedir.  $F_{giris}$  saat sinyali 10 MHz olarak kullanıldığı için frekans çözünürlüğü, faz çözünürlüğüne bağlı olarak 2,3 mHz - 38 Hz arasında değişebilmektedir. Frekans çözünürlüğü Eşitlik (6.1) ile hesaplanabilir.

$$frekans\ çözünürlüğü = \frac{F_{giris}}{2^N} \quad (6.1)$$

DDFS'nin genlik çözünürlüğünü ilk değer tablosu genişliği belirlemektedir. İlk değer tablosu ve ofset tablosu toplamı DDFS çıkışı olarak DAC birimine uygulanacağı için, hedeflenen genlik çözünürlüğü DAC çözünürlüğünden büyük olmamalıdır. DDFS tasarımlarının sinüs sinyali için çözünürlük ve frekans aralığı değerleri Çizelge 6.1'de verilmiştir.

**Çizelge 6.1.** Tasarımların çözünürlük ve frekans aralığı değerleri

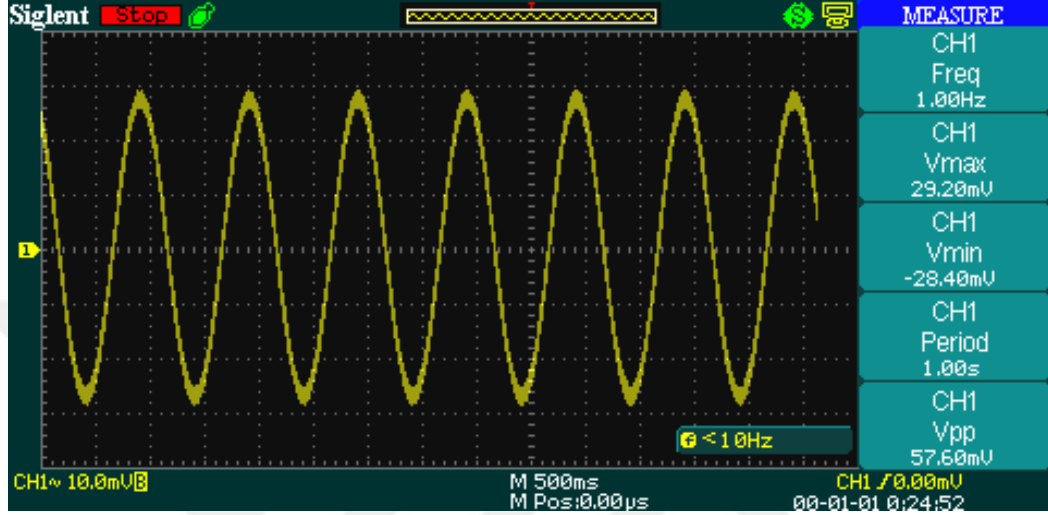
| Tasarım No | Faz Çöz. (bit)  | Frekans Çöz. (mHz) | Genlik Çözünürlüğü (bit) | Frekans Aralığı |
|------------|-----------------|--------------------|--------------------------|-----------------|
| 1          | sabit, 32       | 2,3                | 16                       | 1 Hz-4 MHz      |
| 2          | sabit, 32       | 2,3                | 10                       | 1 Hz-4 MHz      |
| 3          | sabit, 32       | 2,3                | 16                       | 1 Hz- 1 MHz     |
| 4          | değişken, 18-32 | 38000-2,3          | 16                       | 1 Hz-4 MHz      |

Çizelge 6.1'den de görüleceği üzere BTM ve kuadrant sıkıştırma tekniğinin birlikte uygulandığı tasarım 3'te frekans aralığı diğer tasarımlara oranla daha küçüktür. Bunun nedeni ilk değer tablo boyutunun sinüs simetrisinden faydalanarak kuadrant sıkıştırma ile sıkıştırılmasıdır.

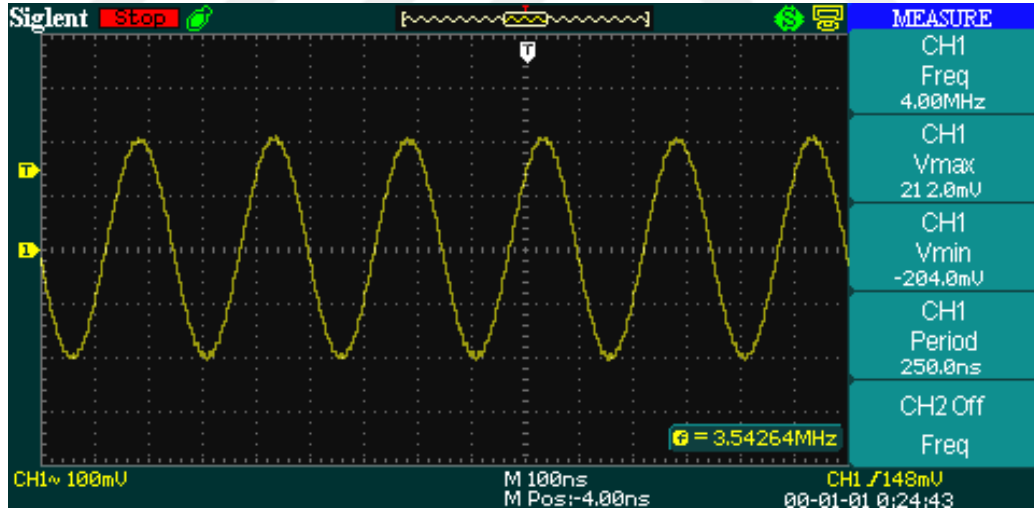
Farklı dalga formları elde etmek için tasarlanan 4 numaralı DDFS'nin sinüs, testere, üçgen ve kare dalgalar için gözlemlenen frekans aralıkları Çizelge 6.2'de verilmiştir. Şekil 6.1-Şekil 6.8'de bu tasarım ile elde edilen minimum ve maksimum frekanstaki dalga formlarını göstermektedir.

**Çizelge 6.2.** Sinüs, kosinüs, testere dişi, üçgen ve kare dalga formlarının için frekans aralıkları

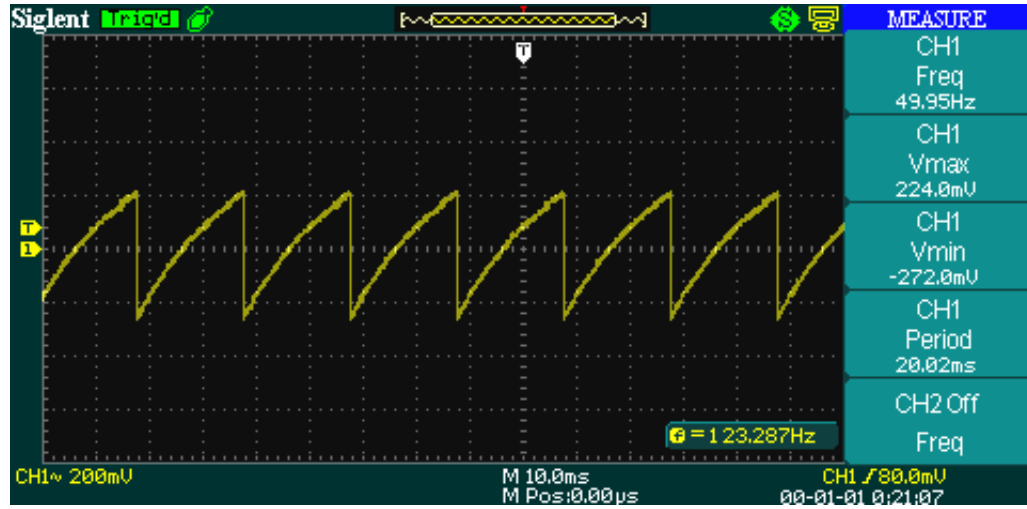
| Dalga Formu  | Minimum Frekans | Kullanılan AGF | Maksimum Frekans | Kullanılan AGF |
|--------------|-----------------|----------------|------------------|----------------|
| Sinüs        | 1 Hz            | P7LP-605L      | 4 MHz            | P7LP-605L      |
| Testere Dişi | 50 Hz           | P7LP-605L      | 625 kHz          | P7LP-306L      |
| Üçgen        | 50 Hz           | P7LP-605L      | 625 kHz          | P7LP-306L      |
| Kare         | 100 Hz          | kullanılmadı   | 1 MHz            | kullanılmadı   |



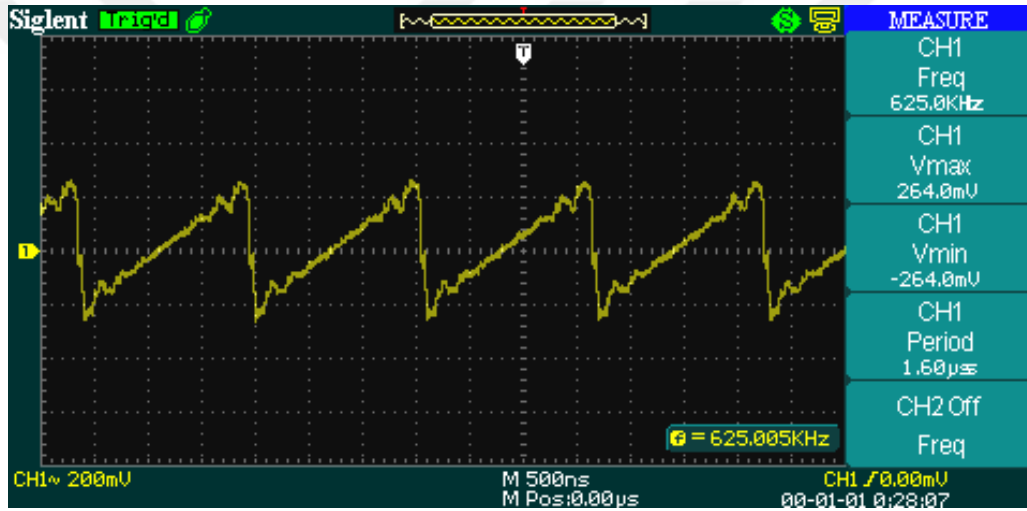
**Şekil 6.1.** Tasarım 4 ile elde edilen 1 Hz sinüs sinyali



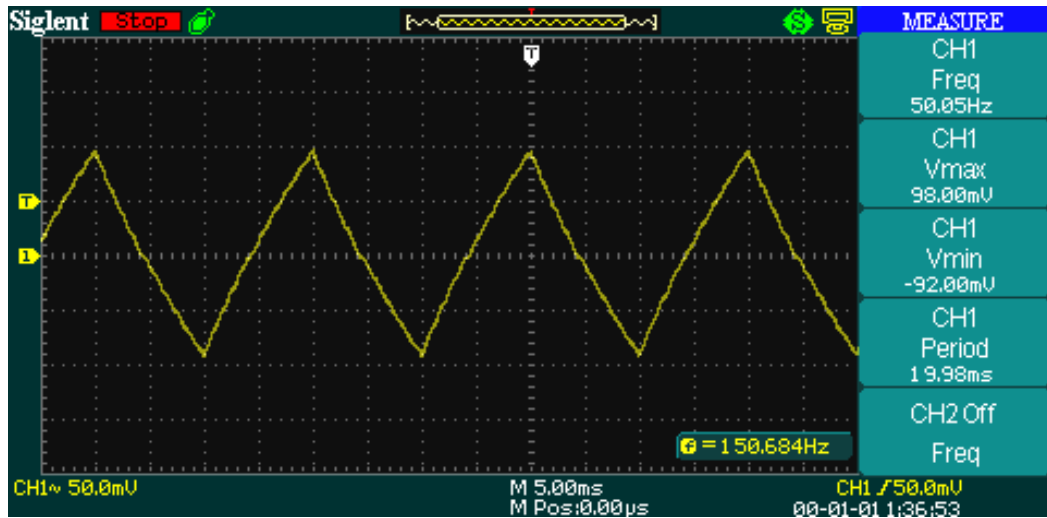
**Şekil 6.2.** Tasarım 4 ile elde edilen 4 MHz sinüs sinyali



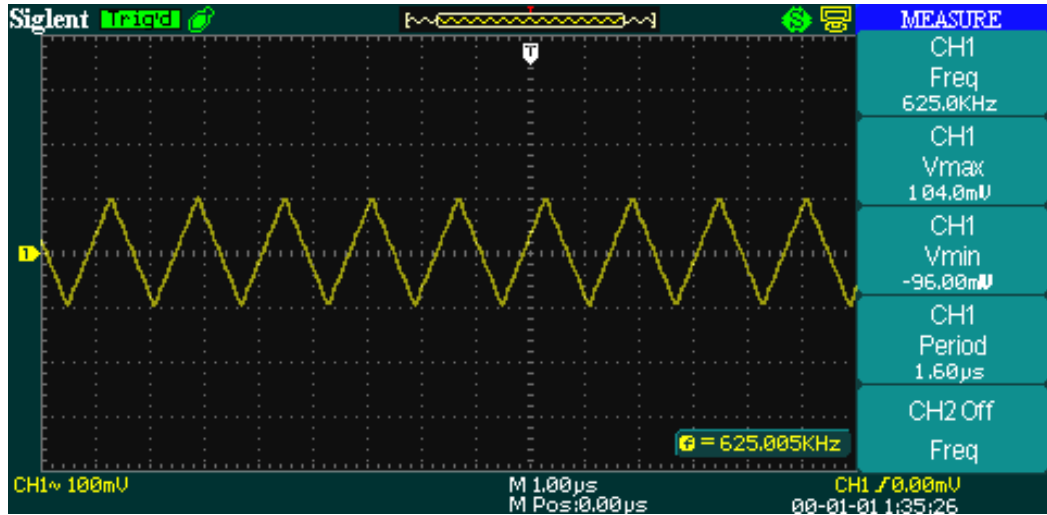
Şekil 6.3. Tasarım 4 ile elde edilen 50 Hz testere dişi sinyali



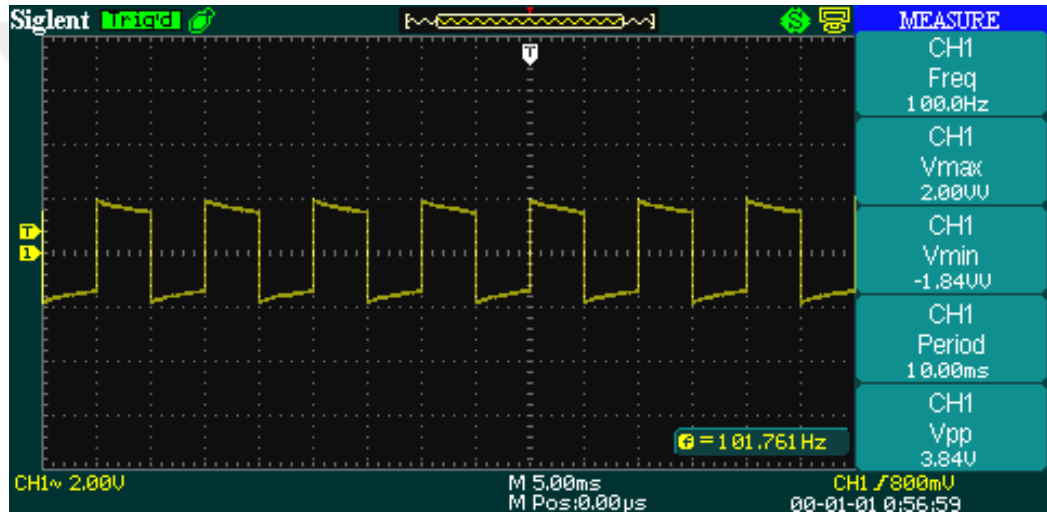
Şekil 6.4. Tasarım 4 ile elde edilen 625 kHz testere dişi sinyali



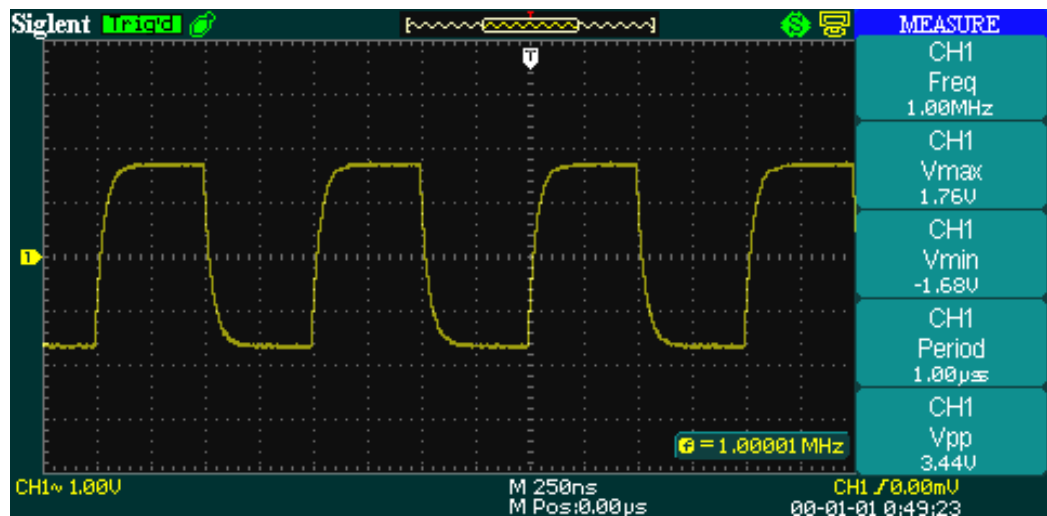
Şekil 6.5. Tasarım 4 ile elde edilen 50 Hz üçgen dalga formu



Şekil 6.6. Tasarım 4 ile elde edilen 625 kHz üçgen dalga formu



Şekil 6.7. Tasarım 4 ile elde edilen 100 Hz kare dalga formu



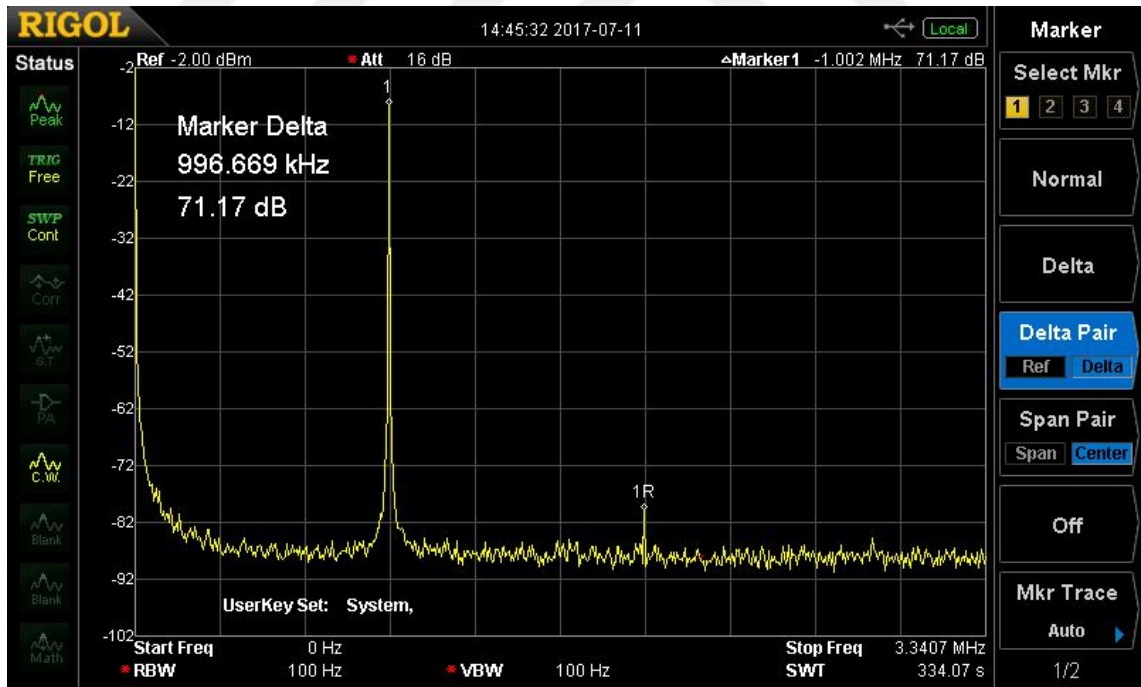
Şekil 6.8. Tasarım 4 ile elde edilen 1 MHz kare dalga formu

## 6.2. Elde Edilen Sinüzoidal Sinyalin Spektral Saflığı

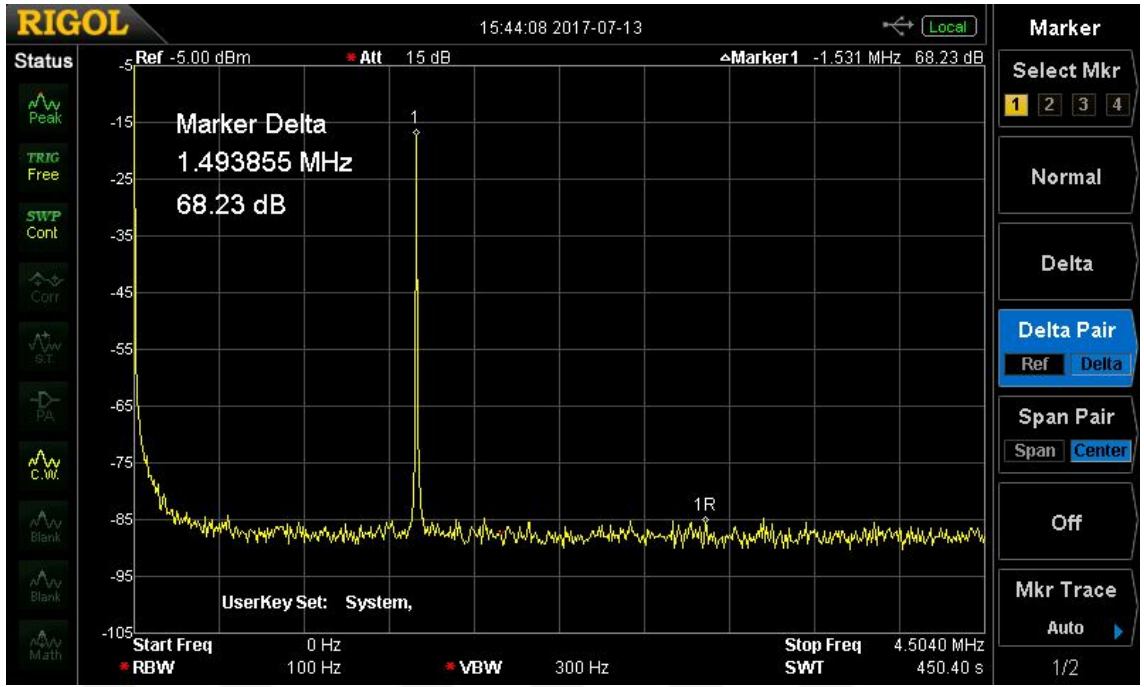
Gerçeklenen DDFS tasarımları arasında en yüksek SFDR değerine tasarım 1 ve 4'te ulaşılmıştır. Bu tasarımlarda BTM parametreleri, Şekil 5.7'de verilen kod parçası kullanılarak  $a = 10$ ,  $b = 8$ ,  $c = 3$  olarak belirlenmiştir. Bu belirlemelerde hedef SFDR 100 dBc seçilmiş, SFDR hesaplamasında De Caro ve arkadaşlarının (2008) önerdiği Eşitlik (5.2)'de verilen formül kullanılmıştır. Farklı frekanslarda elde edilen maksimum SFDR değerleri Çizelge 6.3'de, elde edilen sinyallerin spektrumdaki görüntüsü de Şekil 6.9-Şekil 6.13'te verilmiştir.

Çizelge 6.3. Farklı frekanslarda elde edilen maksimum SFDR değerleri

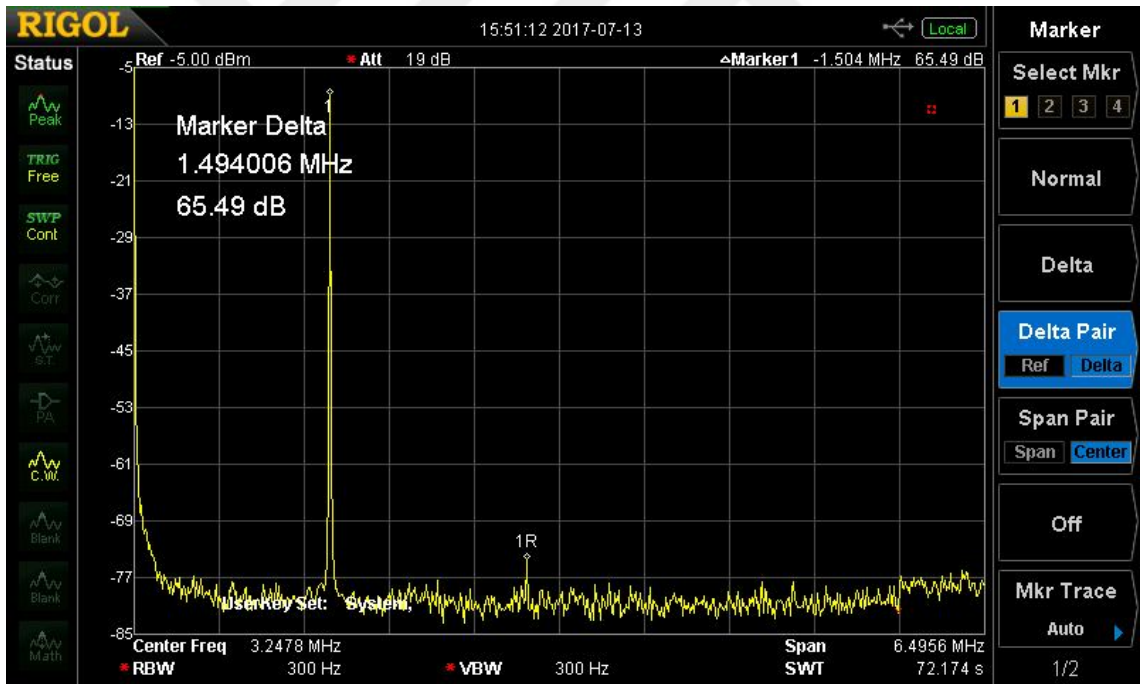
| Frekans | Kullanılan Filtre | SFDR      |
|---------|-------------------|-----------|
| 1 MHz   | P7LP-155L         | 71.17 dBc |
| 1.5 MHz | P7LP-155L         | 68.23 dBc |
| 1.5 MHz | P7LP-605L         | 65.49 dBc |
| 3 MHz   | P7LP-605L         | 67.75 dBc |
| 4 MHz   | P7LP-605L         | 70.77 dBc |



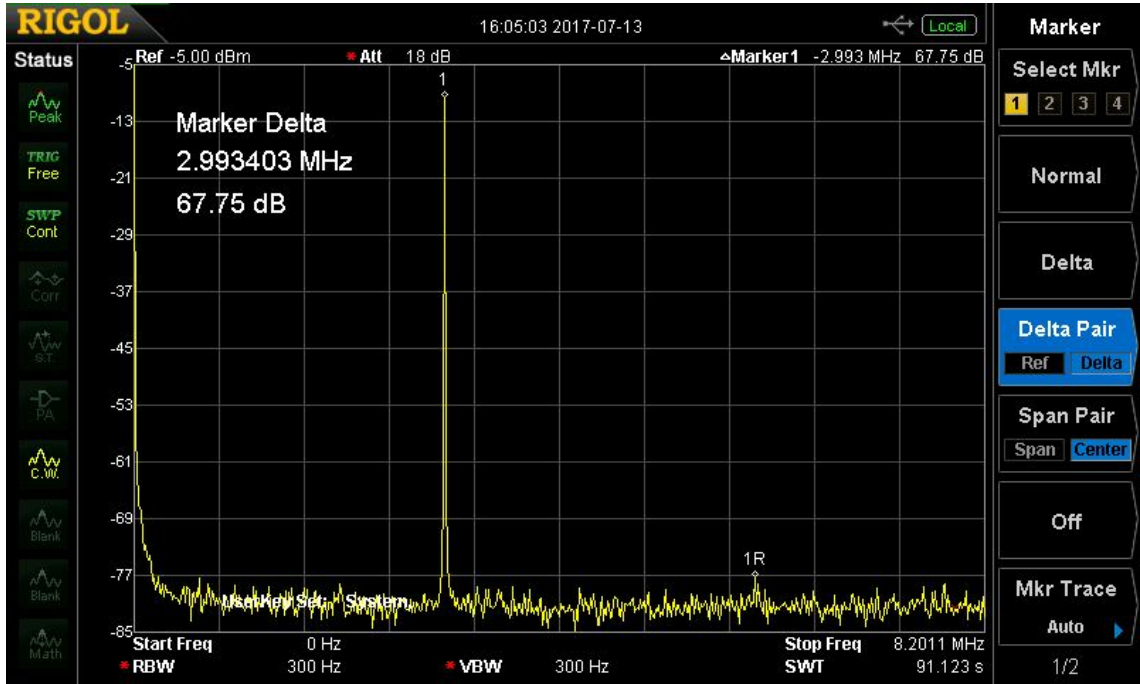
Şekil 6.9. Tasarım 4 ile üretilen 1 MHz sinüsün spektrum görüntüsü (P7LP-155L)



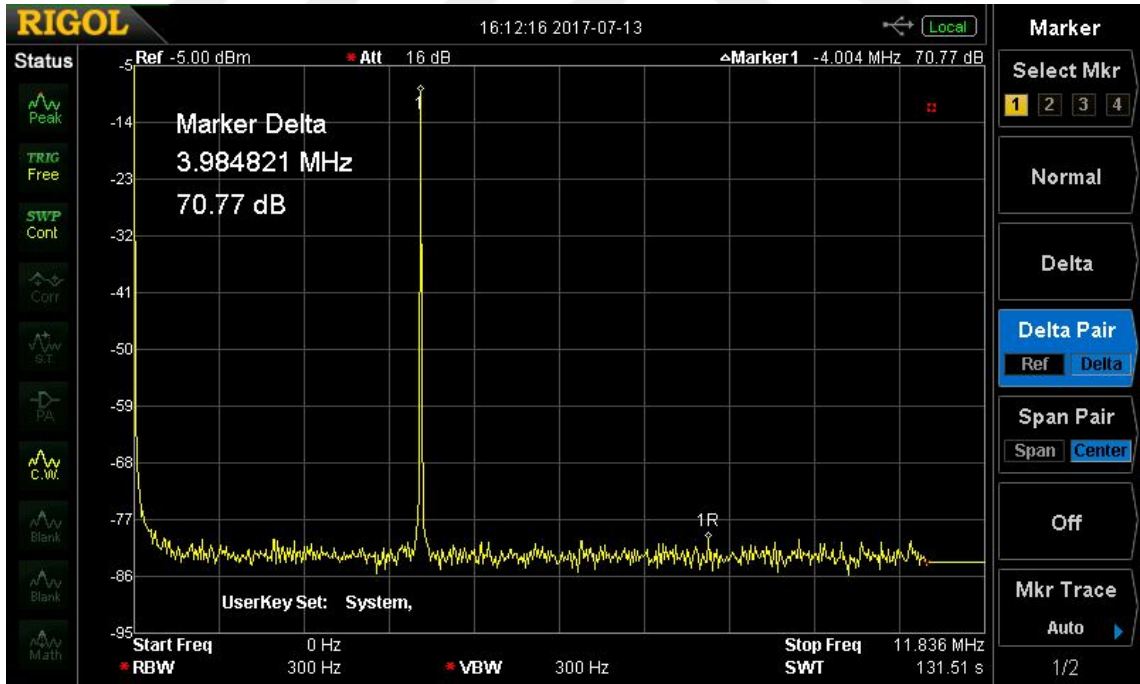
Şekil 6.10. Tasarım 4 ile üretilen 1.5 MHz sinüsün spektrum görüntüsü (P7LP-155L)



Şekil 6.11. Tasarım 4 ile üretilen 1.5 MHz sinüsün spektrum görüntüsü (P7LP-605L)



Şekil 6.12. Tasarım 4 ile üretilen 3 MHz sinüsün spektrum görüntüsü (P7LP-605L)

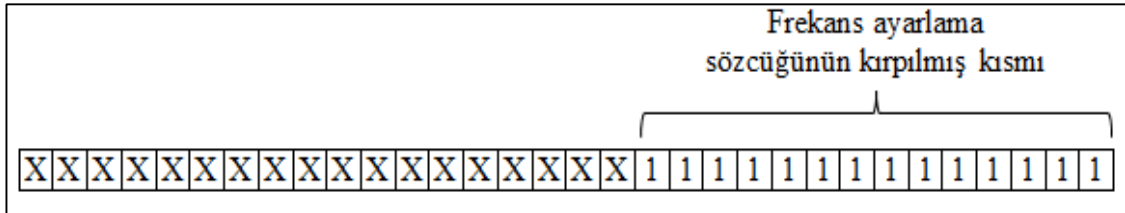


Şekil 6.13. Tasarım 4 ile üretilen 4 MHz sinüsün spektrum görüntüsü (P7LP-605L)

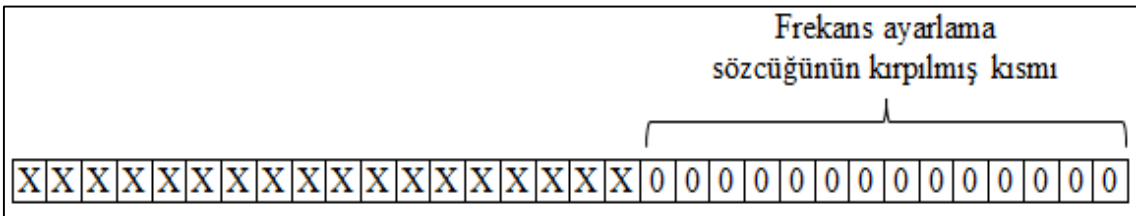
### 6.2.1. Faz kırpmasının spektral saflığa etkisi

Gerçekleştirilen tasarımlarda maksimum faz çözünürlüğü 32 bit ve genlik çözünürlüğü 16 bittir. Faz kırpması uygulanmayan bu özelliklerdeki geleneksel bir DDFS için gerekli olan LUT boyutu  $2^{32} \times 16 = 2^{36}$  bit yani 64 gigabayttır. Bu boyutlarda bir LUT içeren bir DDFS gerçekleştirilebilir değildir.

DDFS uygulamalarında, bellek küçültme yöntemleri ile birlikte faz kırpması da kullanılarak gerekli olan hafıza boyutu düşürülmektedir. Ancak, faz kırpması uygulanması elde edilen sinyalin oluşması beklenen sinyalden farklı olmasına neden olmaktadır. Örneğin, 12 bit faz çözünürlüklü bir DDFS'e 4 bit faz kırpması uygulanacak olursa, oluşturulan faz tekerleği üzerindeki çözünürlük  $\frac{360^\circ}{2^{12}} = 0,088^\circ$  yerine  $\frac{360^\circ}{2^8} = 1,406^\circ$  olacaktır. Frekans ayarlama sözcüğünün 10 seçildiği bir durum için elde edilen faz ile istenilen faz arasında maksimum  $10 \times 0,088^\circ = 0,88^\circ$  hata oluşacaktır. Periyodik olarak oluşacak bu hata spektruma periyodik bir gürültü olarak yansıyacaktır (Kester, 1999). Faz çözünürlüğü 32 bit olan ve 14 bit faz kırpması uygulanan bir DDFS için maksimum gürültüyü oluşturacak en kötü senaryo ve ilave gürültüye sebep olmayacak en iyi senaryo Şekil 6.14'te ve Şekil 6.15'te verilmiştir.



Şekil 6.14. Faz kırpması için en kötü senaryo

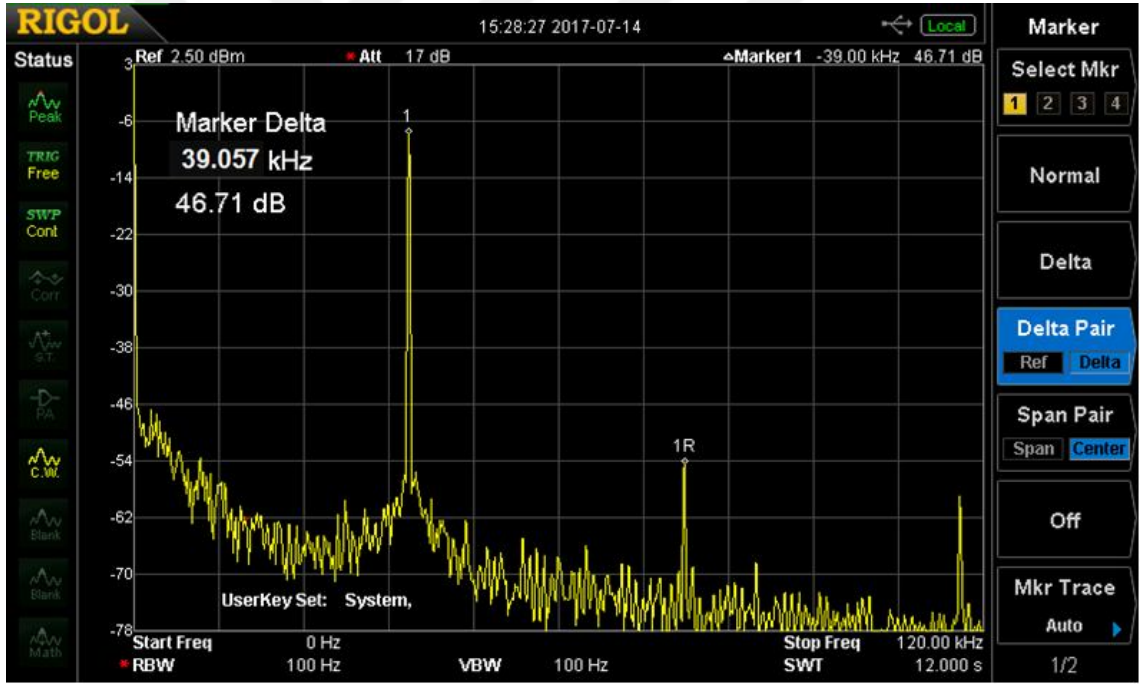


Şekil 6.15. Faz kırpması için en iyi senaryo

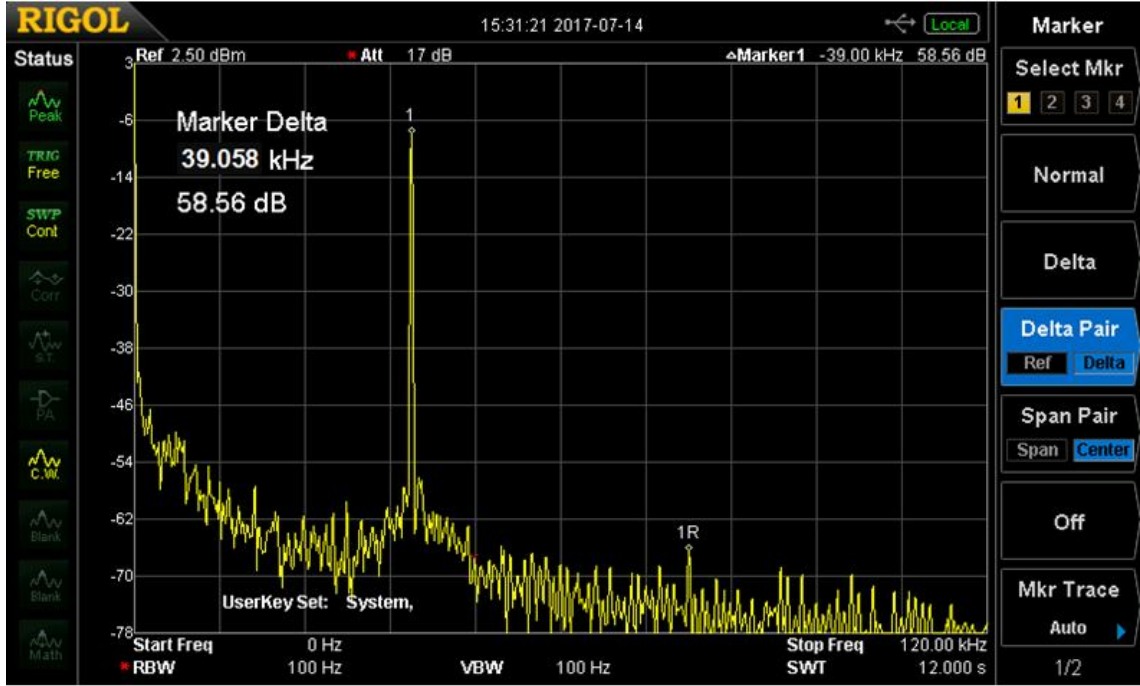
Faz kırpmasının elde edilen sinüzoidalın spektral performansına etkisi farklı faz çözünürlüklerinde ve iki ayrı frekansta test edilmiştir. Test frekansları ve faz çözünürlükleri Çizelge 6.4'te, spektrum görüntüleri ise Şekil 6.16-Şekil 6.23'te verilmiştir.

**Çizelge 6.4.** Faz kırpması test frekansları, faz çözünürlükleri ve SFDR değerleri

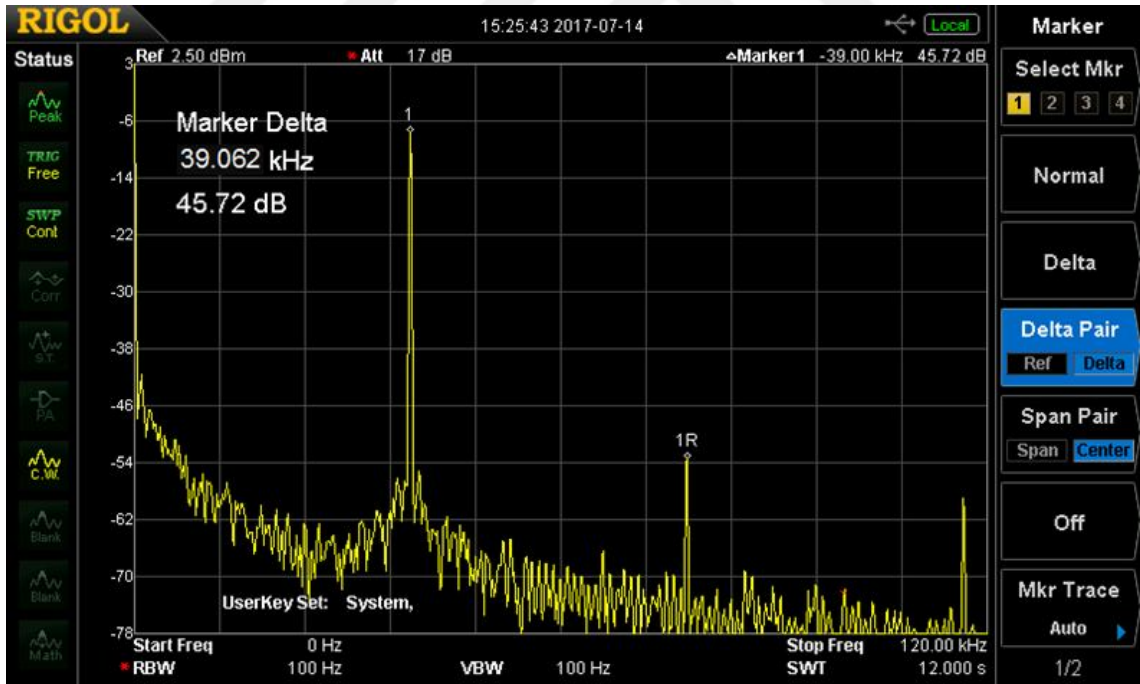
| Frekans    | Frekans Ayarlama Sözcüğü | Faz Çözünürlüğü | SFDR      |
|------------|--------------------------|-----------------|-----------|
| 39.057 kHz | FFF                      | 20 bit          | 46.71 dBc |
| 39.058 kHz | 1000                     | 20 bit          | 58.56 dBc |
| 39.062 kHz | FFFF                     | 24 bit          | 45.72 dBc |
| 39.063 kHz | 1001                     | 24 bit          | 57.26 dBc |
| 39.062 kHz | FFFF3                    | 28 bit          | 45.21 dBc |
| 39.063 kHz | 10000D                   | 28 bit          | 58.85 dBc |
| 39.062 kHz | FFFF29                   | 32 bit          | 55.58 dBc |
| 39.063 kHz | 10000D7                  | 32 bit          | 57.37 dBc |



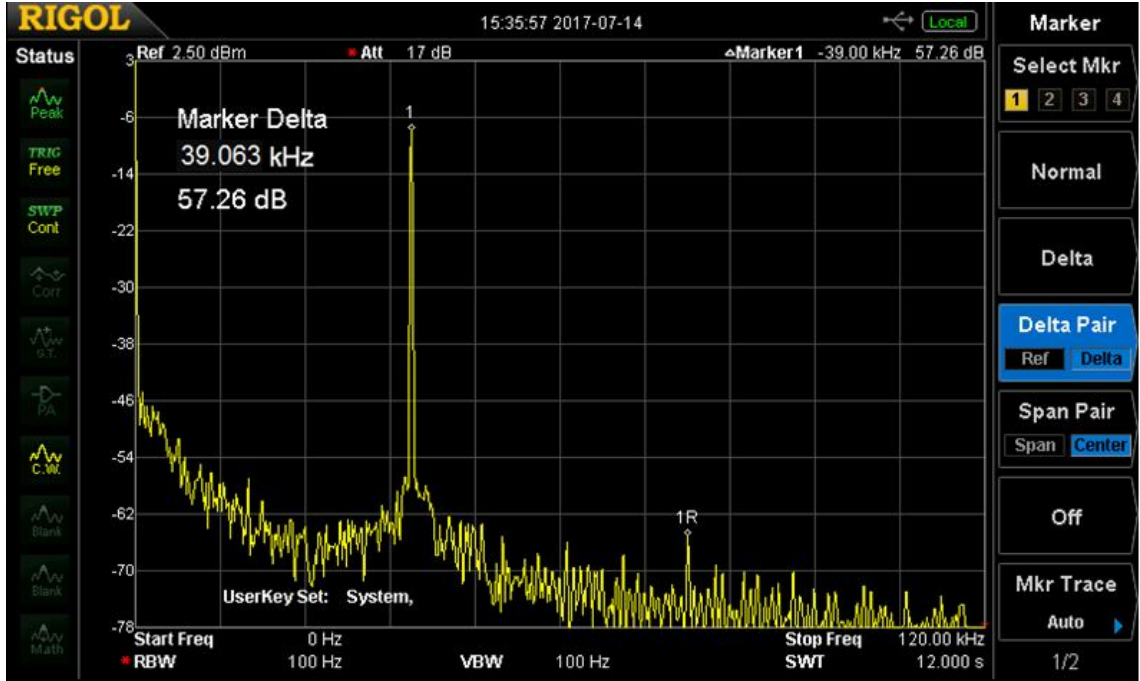
**Şekil 6.16.** 20 bit Faz Çözünürlüğü ile Elde Edilen 39.057 Hz Frekanslı Filtrenmemiş Sinüs



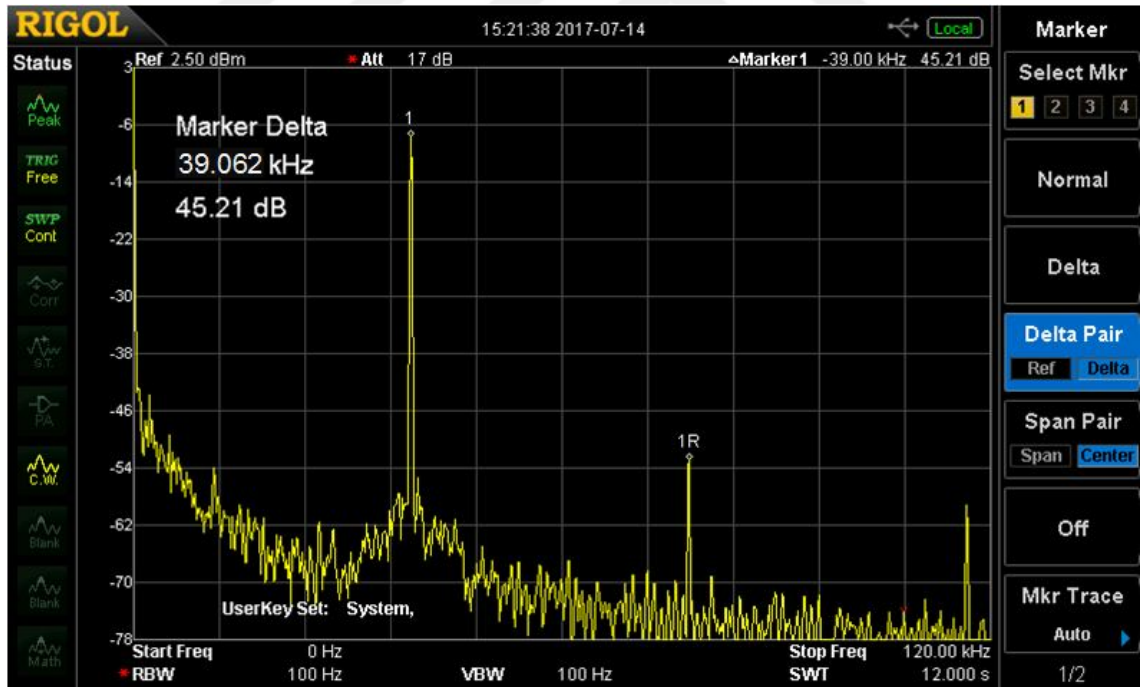
Şekil 6.17. 20 bit Faz Çözünürlüğü ile Elde Edilen 39.058 Hz Frekanslı Filtrenmemiş Sinüs



Şekil 6.18. 24 bit Faz Çözünürlüğü ile Elde Edilen 39.062 Hz Frekanslı Filtrenmemiş Sinüs

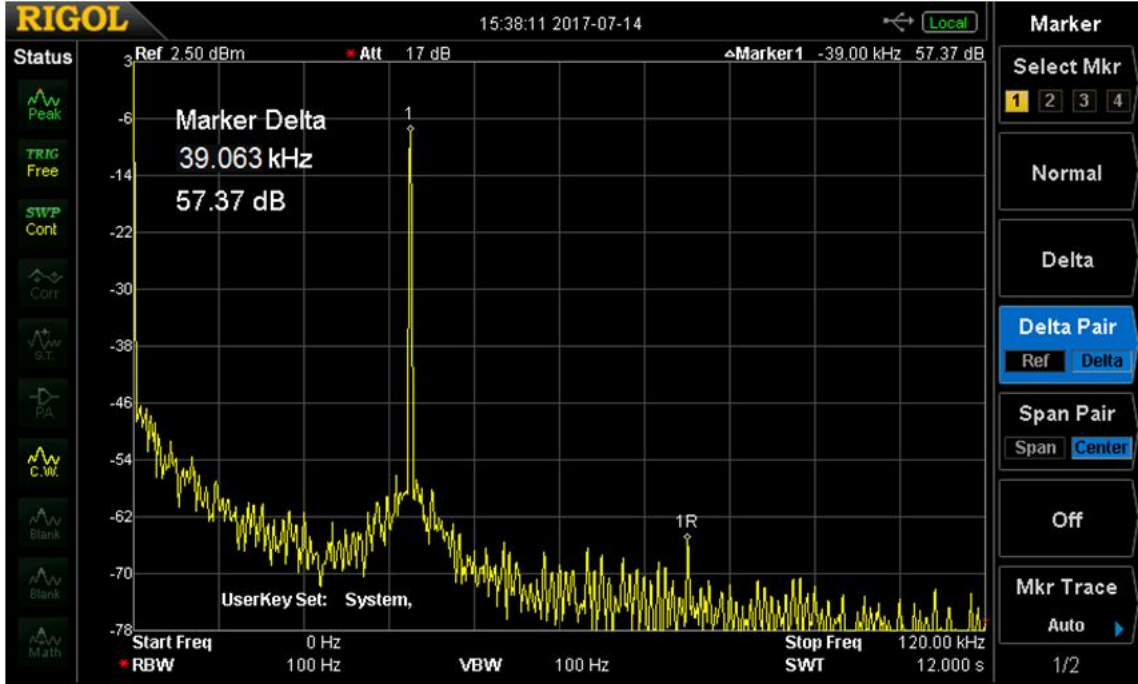


Şekil 6.19. 24 bit Faz Çözünürlüğü ile Elde Edilen 39.063 Hz Frekanslı Filtrenmemiş Sinüs



Şekil 6.20. 28 bit Faz Çözünürlüğü ile Elde Edilen 39.062 Hz Frekanslı Filtrenmemiş Sinüs





Şekil 6.23. 32 bit Faz Çözünürlüğü ile Elde Edilen 39.063 Hz Frekanslı Filtrenmemiş Sinüs

### 6.2.2. Sayısal-analog dönüştürücü ve filtrenin spektral saflığa etkisi

DDFS çıkış sinyalinin spektral saflığına etkiyen önemli unsurlardan birisi de kullanılan sayısal-analog dönüştürücüdür. Sayısal-analog dönüştürücünün örnekleme frekansı ve çözünürlüğü elde edilen analog sinyalin SNR (*Signal to Noise Ratio*) değerine etki eder. SNR değeri, sinyal gücü  $P_s$ , gürültü gücü  $P_g$  olmak üzere Eşitlik (6.2) ile hesaplanır.

$$SNR (dB) = 10 \times \log\left(\frac{P_s(\text{watt})}{P_g(\text{watt})}\right) \quad (6.2)$$

Sayısal-analog dönüşümünde oluşan kuantalama hataları spektruma gürültü olarak yansımaktadır.  $q$  genlik çözünürlüklü bir DAC için kuantalama gürültüsü ve SNR değeri Eşitlik (6.3) ve Eşitlik (6.4) ile hesaplanır (Kılıç, 2006).

$$\text{kuantalama gürültüsü}(rms) = \frac{q}{\sqrt{12}} \quad (6.3)$$

$$SNR (dB) = 20 \log_{10} \left[ \frac{q \times 2^R}{2\sqrt{2}} / \frac{q}{\sqrt{12}} \right] = 6.02R + 1.76 \text{ dB} \quad (6.4)$$

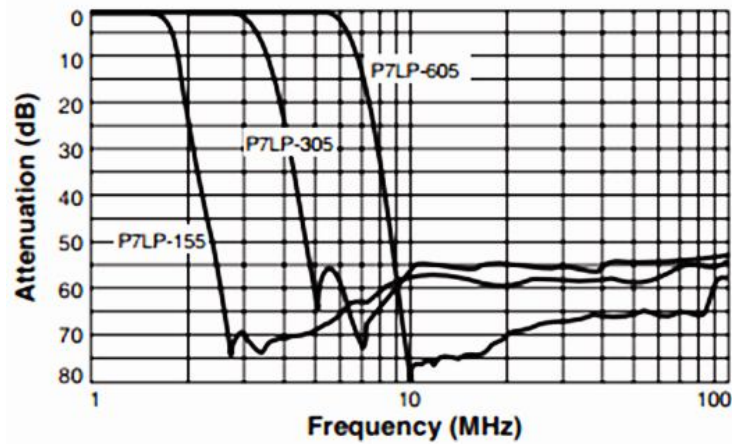
Kuantalama hatalarının genliğini düşürmek adına örnekleme frekansı olabildiğince yüksek seçilmelidir. Bir başka deyişle,  $f_{giris}$ , analog forma dönüştürülecek sayısal değerlerin değişim frekansı;  $f_{dac}$ , sayısal-analog dönüştürücünün *Nyquist* örnekleme frekansı,  $f_{os}$  artırılmış örnekleme frekansı ve  $f_{os} \geq f_{dac}$  olmak üzere,  $\frac{f_{os}}{f_{giris}}$  oranının yüksek tutulması ve DAC çıkışının filtrelenmesi SNR değerine olumlu etki etmektedir. Bu etkiler göz önünde bulundurulacak olursa; *BG* filtrenin bant genişliği,  $f_{os}$  artırılmış örnekleme frekansı olmak üzere SNR değeri aşağıdaki şekilde hesaplanabilir.

$$SNR = 20 \log_{10} \left[ \frac{q \times 2^R}{2\sqrt{2}} / \frac{q}{\sqrt{12}} \right] + 10 \log_{10} \left[ \frac{f_{dac}/2}{BG} \right] + 10 \log_{10} \left[ \frac{f_{os}}{f_{dac}} \right] \quad (6.5)$$

$\frac{f_{os}}{f_{giris}}$  oranı 2, 4 ve 8 olarak elde edilen 1.25 MHz frekanslı sinüs sinyallerinin hesaplanan SNR değerleri Çizelge 6.5'te, kullanılan filtrelerin zayıflatma- frekans eğrileri de Şekil 6.24'te verilmiştir.

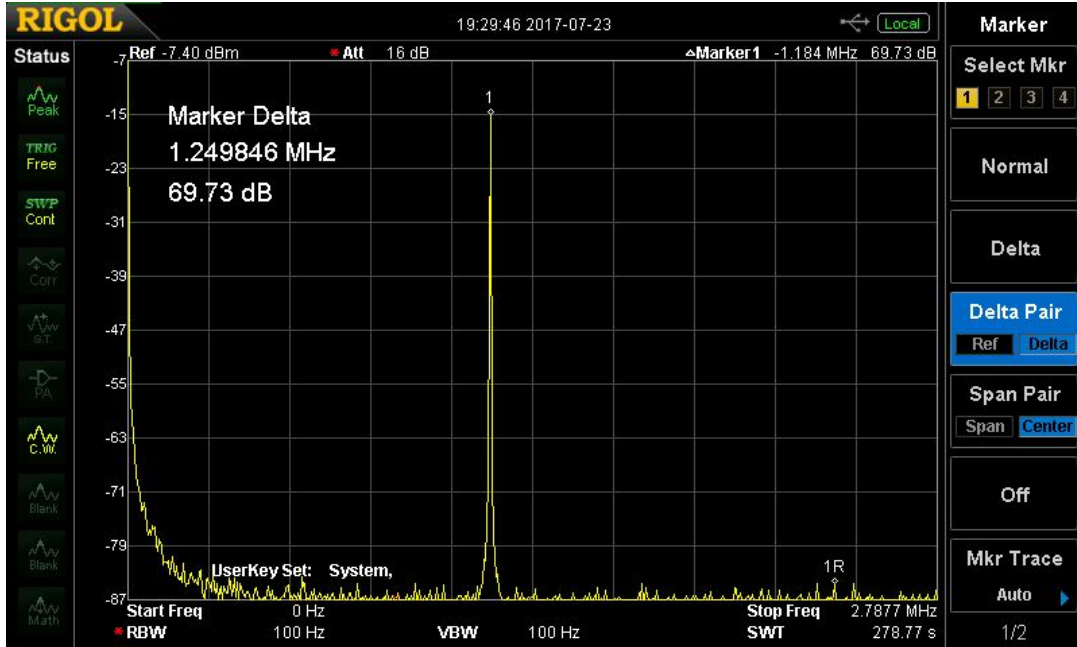
**Çizelge 6.5.** 1.25 MHz sinüs sinyalinin hesaplanan SNR değerleri

| $f_{os}$ | Kullanılan AGF | Hesaplanan SNR |
|----------|----------------|----------------|
| 5 MHz    | P7LP-155       | 100.3 dB       |
| 5 MHz    | P7LP-305       | 97.3 dB        |
| 10 MHz   | P7LP-155       | 103.31 dB      |
| 10 MHz   | P7LP-305       | 100.31 dB      |
| 20 MHz   | P7LP-155       | 106.32 dB      |
| 20 MHz   | P7LP-305       | 103.32 dB      |

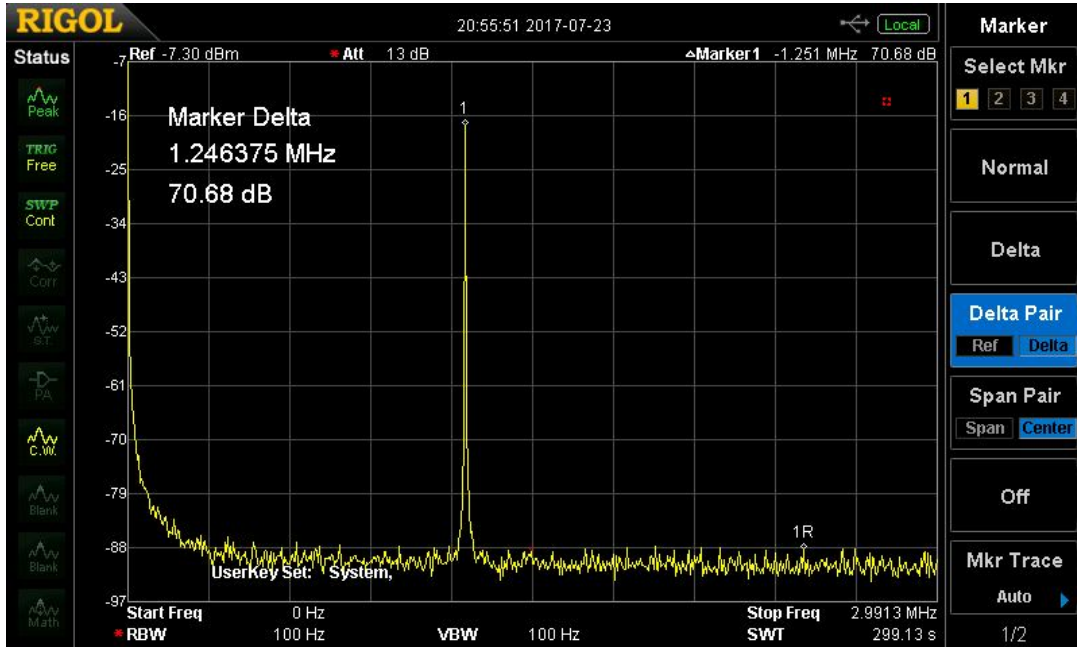


**Şekil 6.24.** Kullanılan filtrelerin zayıflatma-frekans grafikleri

P7LP-155 alçak geçiren filtresi kullanılarak,  $f_{os} = 10\text{MHz}$  ve  $f_{os} = 20\text{MHz}$  olmak üzere elde edilen 1.25 MHz frekanslı sinüs sinyallerinin spektrum görüntüleri Şekil 6.25 ve Şekil 6.26’da verilmiştir.



Şekil 6.25.  $f_{os} = 10\text{MHz}$  için P7LP-155 filtresi ile elde edilen 1.25 MHz’lik sinüsün spektrum görüntüsü



Şekil 6.26.  $f_{os} = 20\text{ MHz}$  için P7LP-155 filtresi ile elde edilen 1.25 MHz’lik sinüsün spektrum görüntüsü

### 6.3. Harcanan güç ve ROM boyutu

DDFS tasarımlarının önemli bir parametresi de harcadığı güç ve fiziksel alandır. ROM sıkıştırma teknikleri ile ROM boyutunun düşürülmesi ve güç tüketiminin azaltılması hedeflenmektedir. BTM ile ROM sıkıştırıldığında sıkıştırma oranı Eşitlik (6.6) ile hesaplanır. Gerçekleştirilen tasarımların ROM boyutları ve sıkıştırma oranları Çizelge 6.6’da verilmiştir.

$$\text{Sıkıştırma oranı} = \frac{2^{a+b} \times R}{2^a \times W_{TIV} + 2^{b+c} \times W_{TO}} \quad (6.6)$$

**Çizelge 6.6.** Gerçekleştirilen tasarımların ROM boyutları ve sıkıştırma oranları

| Tasarım No | TIV boyutu (bayt) | TO boyutu (bayt) | Toplam ROM boyutu (kбайt) | Sıkıştırma oranı |
|------------|-------------------|------------------|---------------------------|------------------|
| 1 ve 4     | 2048              | 2048             | 4                         | 128              |
| 2          | 160               | 160              | 0.3125                    | 16               |
| 3          | 480               | 2048             | 2.4688                    | 207.39           |

Bu çalışmada, tasarlanan yapıların güç tüketim tahminleri Vivado 2016.3 yazılımının *power report* aracı kullanılarak yapılmıştır. Klasik DDFS ve ROM sıkıştırma tekniği uygulanmış DDFS tasarımlarının güç tüketimi karşılaştırıldığında güç tüketiminin yaklaşık olarak %93 azaldığı görülmektedir. Tasarımların güç tüketim tahminleri yaklaşık olarak Çizelge 6.7’de verilmiştir.

**Çizelge 6.7.** Klasik DDFS ve BTM uygulanmış DDFS tasarımlarının güç tüketimi

| Yöntem         | Fgiris  | Harcanan Güç | ROM Boyutu   |
|----------------|---------|--------------|--------------|
| Klasik DDFS    | 400 MHz | 270µW/MHz    | 512 kбайt    |
| Tasarım 1 ve 4 | 400 MHz | 20µW/MHz     | 4 kбайt      |
| Tasarım 2      | 400 MHz | 16µW/MHz     | 320 bayt     |
| Tasarım 3      | 400 MHz | 20µW/MHz     | 2.4688 kбайt |

## 7. SONUÇLAR VE ÖNERİLER

### 7.1. Sonuçlar

Bu tez çalışmasında dört farklı DDFS tasarımı başarıyla gerçekleştirilmiştir. Tasarımlara faz kırpması ve ROM sıkıştırma teknikleri uygulanarak ROM boyutları düşürülmüştür.

Gerçekleştirilen tasarımların ilki olan Tasarım 1’de yüksek SFDR değeri elde etmek temel hedef seçilmiştir. De Caro ve arkadaşları (2008) tarafından ortaya konan Eşitlik (5.2) kullanılarak 100 dBc hedef SFDR için en düşük ROM boyutunu sağlayan  $a, b$  ve  $c$  parametreleri belirlenmiştir. Kullanılan donanımlar ve çevresel etkilerin sağlayabileceği spektral saflık üst sınırını görmek adına yüksek tutulan SFDR hedefine beklenildiği üzere ulaşılamamıştır. Bu tasarım ile 71.17 dBc SFDR değeri elde edilmiştir.

Çevresel etkilerle birlikte kullanılan donanımlar ile elde edilebilecek SFDR üst sınırı olarak 71.17 dBc SFDR değeri göz önüne alınarak Tasarım 2 için hedef SFDR değeri 70 dBc seçilmiştir. Bu tasarımda amaç hedeflenen SFDR değerini sağlayan en düşük ROM boyutu ile tasarımı gerçekleştirmek olmuştur. Hedef SFDR için ROM boyutunun minimize edilmesini sağlayan bir algoritma ile  $a, b$  ve  $c$  parametreleri belirlenmiştir. Tasarım 1’e oranla 12.8 kat daha küçük bir ROM boyutu ile 64.69 dBc SFDR değerine ulaşılmıştır. 2006-2013 yıllarında yapılan benzer çalışmalarla karşılaştırıldığında bu tasarım ROM boyutu ve SFDR değeri açısından daha iyi sonuç vermektedir. Çizelgede de görüldüğü üzere tasarım 2 yapılan diğer çalışmalara SFDR ve ROM boyutu açısından üstünlük sağlasa da maksimum çıkış frekansı açısından çalışmalar arasında üçüncü sıradadır. Bu eksiklik kullanılan yöntem ya da tasarımla ilgili olmamakla beraber düşük saat sinyal giriş frekansından kaynaklanmaktadır. Dâhili 100 MHz saat sinyali barındıran Nexys4 DDR™ geliştirme kartı, düşük frekanslarda kullanılmak üzere tasarlanan PMOD konektörleri nedeniyle 20 MHz seviyelerinde çıkış verebilmektedir ve gerçekleşen sistemin çıkış frekansını sınırlayan temel unsur olmuştur. 2006-2013 yıllarında yapılan benzer çalışmalar bu tasarımın karşılaştırılması Çizelge 7.1’de verilmiştir.

**Çizelge 7.1.** Bazı DDS/DDFS çalışmalarının karşılaştırılması

|                             | Tasarım 2 | KILIÇ<br>(2006) | Kazancıoğlu<br>(2007) | Aydoğan<br>(2013) |
|-----------------------------|-----------|-----------------|-----------------------|-------------------|
| Yöntem                      | Klasik    | Kuadrant        | Kuadrant              | Kuadrant          |
|                             | BTM       | Sıkıştırma      | Sıkıştırma            | Sıkıştırma        |
| Maks. saat kaynağı (MHz)    | 10        | 100             | 125                   | 33                |
| Genlik çöz. (bit)           | 10        | 16              | 12                    | 16                |
| Kırpılmış faz sözcüğü (bit) | 12        | 10              | 10                    | 12                |
| Maks. çıkış frekansı        | 4 MHz     | 67.5 MHz        | 40 MHz                | 180 kHz           |
| ROM boyutu (Bayt)           | 320       | 2000            | 3000                  | 4000              |
| SFDR                        | 64.69 dBc | 60 dBc          | 60 dBc                | -                 |

Tasarım 3, BTM ve kuadrant sıkıştırma yöntemlerinin birlikte kullanılabilirliğini test etmek üzere tasarlanmıştır. Sıkıştırma oranı en yüksek tasarım olsa da çıkış frekansı diğer tasarımlara oranla düşük kalmıştır. Bu tasarım, yüksek referans saat kaynağı sağlanıp, düşük frekans aralığı hedeflendiğinde güç tüketimi ve ROM boyutu açısından diğer tasarımlara üstünlük sağlamaktadır.

Tasarım 4, faz kırpması ve faz çözünürlüğü gibi unsurların spektral saflığa etkisini gözlemlemek için değişken faz çözünürlüklü olarak tasarlanmıştır. Sinüs, kare, testere dişi, üçgen dalga çıkışı sağlayan bu tasarım, çıkışına ilave edilecek bir yükselteç devresi ile fonksiyon jeneratörü olarak kullanılabilir. Ayrıca çıkış sinyal çözünürlüğü düşürülerek sinyal genliği sayısal olarak da değiştirilebilir.

## 7.2. Öneriler

Sayısal elektronik teknolojisindeki hızlı gelişmeye paralel olarak birçok sistemde analog çözümler yerine sayısal olanlar kullanılmaya başlanmıştır. Doğrudan sayısal frekans sentezleyici de bu dönüşümün içinde kendine yer bulan sistemlerden olmuştur. Analog frekans sentezleyicilerine göre daha kararlı yapı, yüksek anahtarlama hızı, yüksek çözünürlük, tasarım kolaylığı ve düşük tasarım maliyeti gibi avantajlarından dolayı, doğrudan sayısal frekans sentezleyicileri gün geçtikçe daha fazla kullanılmaktadır.

Doğrudan sayısal frekans sentezleyici tasarımında çok sayıda yöntem geliştirilmiştir. Temelde ROM tabanlı ve iteratif olarak ikiye ayrılan doğrudan sayısal frekans sentezleyicilerinde günümüzde, polinomsal bazı yaklaşımlar kullanılarak

yüksek çalışma frekansı ve SFDR değerleri elde edilmiştir (Jafari ve ark., 2005; Chen ve Chau, 2010). ROM tabanlı olmayan bu yöntemlerin güç tüketimleri, parçalı tablo tabanlı yaklaşımlara oranla daha yüksektir. Polinomsal bazı yaklaşımların parçalı tablo yöntemleri ile hibrit kullanımıyla güç tüketiminin azaltılması mümkün olacaktır.

BTM ile birlikte kuadrant sıkıştırma tekniği birlikte kullanıldığında ROM boyutunun daha da küçüleceği yapılan çalışmada açıkça görülmesine rağmen yüksek frekanslarda iki yöntemin birlikte kullanılması spektral saflık açısından daha kötü sonuçlar vermektedir. Çeyrek periyot olarak tabloda tutulan sinüs değerlerinden diğer çeyreklerin elde edilmesi işlemlerinde oluşan gecikmelerin yüksek frekanslardaki bozulmalara neden olduğu sonucuna varılmıştır. Yüksek çıkış frekansının hedeflendiği çalışmalarda kuadrant sıkıştırma tekniği kullanmak yerine ofset tablo sayısı artırılarak çok parçalı tablo yönteminin kullanılmasının daha iyi sonuç vereceği öngörülmektedir.

Sayısal frekans sentezleyici tasarımında, faz ve genlik çözünürlüğü, SFDR, maksimum frekans, kullanılan alan, anahtarlama frekansı ve harcanan güç gibi çok sayıda önemli parametre dikkate alınmaktadır. Bu parametrelerden bazılarında iyileştirme sağlanmak istediğinde bazıları kötüleşmektedir. Uygulamaya yönelik olarak, bazı parametrelerin optimize edilmesine yönelik çalışmalarla, bazı parametrelerin iyileştirildiği çalışmalar bulunmaktadır. Bu çalışmalardan bazıları Çizelge 7.2’de verilmiştir.

**Çizelge 7.2.** Optimize edilmiş DDFS çalışmaları

| Çalışma               | Yöntem                            | Açıklama                                     |
|-----------------------|-----------------------------------|----------------------------------------------|
| Asok ve Sahoo, 2014   | Bit genişliklerinin optimizasyonu | 75 dBc SFDR, <i>f<sub>out</sub></i> 0-500MHz |
| Ashrafi, 2014         | İnterp. katsayıları optimizasyonu | 6.5 $\mu\text{m}^2$ silikon alan             |
| De Caro ve ark., 2008 | Ofset tablo sayısı optimizasyonu  | Hedef SFDR için min. Alan                    |
| Lin ve ark., 2005     | Bit genişlikleri optimizasyonu    | 565:1 ROM Sıkıştırma<br>96 dBc SFDR          |

Çizelgede de görüldüğü üzere çalışmalarda tek bir amaca yönelik optimizasyon hedeflenmiştir. SFDR, ROM boyutu, maksimum çalışma frekansı ve güç tüketimi gibi özelliklerden iki ya da daha fazlasının aynı anda, çok amaçlı bir optimizasyon algoritması ile optimize edilmesi DDFS tasarımları için önemli bir yenilik olacaktır.

## KAYNAKLAR

- Anonymous, Direct Digital Frequency Synthesis: a basis tutorial, *Osicom Technologies*.
- Anonymous, Spurious Free Dynamic Range, *Fiber-Span*.
- Anonymous, 2011, Digilent Pmod™ Interface Specification, *Digilent*.
- Anonymous, 2013, DAC5687 EVM, User Guide, *Texas Instrument*.
- Anonymous, 2016, Nexys4 DDR™ FPGA Board Reference Manual, *Digilent*.
- Asok, K. S. ve Sahoo, K. P., 2014, Digital hardware optimization for 1.5-GHz high-speed DDS, *Electronics, Circuits and Systems (ICECS), 2014 21st IEEE International Conference on*, 746-749.
- Aydoğan, N. M., 2013, Fpga Kullanılarak Doğrudan Sayısal Sentez Tabanlı Fonksiyon Üretici Tasarımı.
- Chang, K., 2004, RF and microwave wireless systems, John Wiley & Sons, p.
- Chen, Y.-H. ve Chau, Y. A., 2010, A direct digital frequency synthesizer based on a new form of polynomial approximations, *IEEE Transactions on Consumer Electronics*, 56 (2).
- Chimakurthy, L., 2005, Design of Direct Digital Frequency Synthesizer for Wireless Applications.
- De Caro, D., Petra, N. ve Strollo, A. G., 2008, Reducing lookup-table size in direct digital frequency synthesizers using optimized multipartite table method, *IEEE Transactions on Circuits and Systems I: Regular Papers*, 55 (7), 2116-2127.
- De Caro, D., Petra, N. ve Strollo, A. G., 2011, Direct digital frequency synthesizer using nonuniform piecewise-linear approximation, *IEEE Transactions on Circuits and Systems I: Regular Papers*, 58 (10), 2409-2419.
- De Dinechin, F. ve Tisserand, A., 2005, Multipartite table methods, *IEEE Transactions on Computers*, 54 (3), 319-330.
- Goldberg, B.-G., 1999, Digital Frequency Synthesis Demystified: DDS and Fractional-N PLLs (Demystified).
- Jafari, H., Ayatollahi, A. ve Mirzakuchaki, S., 2005, A low power, high SFDR, ROM-less direct digital frequency synthesizer, *Microelectronics, 2005. ICM 2005. The 17th International Conference on*, 5 pp.
- Jinshan, Y., Dongbing, F., Ruzhang, L., Yafeng, Y., Gang, Y., Jun, L., Ruitao, Z., Zhou, Y. ve Tun, L., 2009, A direct digital frequency synthesizer with high-speed current-steering DAC, *Journal of Semiconductors*, 30 (10), 105006.
- Kazancıoğlu, U., 2007, The Implementation of a Direct Digital Synthesis Based Function Generator Using SystemC and VHDL, *Middle east Technical University*.
- Kester, W., 1999, A technical tutorial on digital signal synthesis, *Analog Devices*, 1-122.
- Kılıç, A., 2006, Implementation Of A Digital Signal Synthesizer With High Spurious Free Dynamic Range, *Middle East Technical University*.
- Madiseti, A., Kwentus, A. Y. ve Willson, A. N., 1999, A 100-MHz, 16-b, direct digital frequency synthesizer with a 100-dBc spurious-free dynamic range, *IEEE Journal of solid-state circuits*, 34 (8), 1034-1043.
- Menakadevi, T. ve Madheswaran, M., 2012, Direct Digital Synthesizer using Pipelined CORDIC Algorithm for Software Defined Radio, *International Journal of Science and Technology*, 2 (6).
- Nicholas, H. T. ve Samueli, H., 1991, A 150-MHz direct digital frequency synthesizer in 1.25- $\mu$ m CMOS with -90-dBc spurious performance, *IEEE Journal of solid-state circuits*, 26 (12), 1959-1969.

- Ohnhäuser, F., 2015, Analog-digital converters for industrial applications including an introduction to digital-analog converters, Springer, p.
- Omran, H., Sharaf, K. ve Ibrahim, M., 2009, An all-digital direct digital synthesizer fully implemented on FPGA, *Design and Test Workshop (IDT), 2009 4th International*, 1-6.
- Patel, G. S. ve Sharma, K., 2013, FPGA-nanotechnology based DDS analysis and implementation, *Journal of Computational and Theoretical Nanoscience*, 10 (10), 2458-2461.
- Radulov, G., Quinn, P., Hegt, H. ve Roermund, A., 2011, Flexible Digital-to-Analog Converters Concept, *Smart and Flexible Digital-to-Analog Converters*, 185-209.
- Strollo, A. G. M., De Caro, D. ve Petra, N., 2007, A 630 MHz, 76 mW direct digital frequency synthesizer using enhanced ROM compression technique, *IEEE Journal of solid-state circuits*, 42 (2), 350-360.
- Tierney, J., Rader, C. ve Gold, B., 1971, A digital frequency synthesizer, *IEEE Transactions on Audio and Electroacoustics*, 19 (1), 48-57.
- Yeoh, H. C., Jung, J.-H., Jung, Y.-H. ve Baek, K.-H., 2010, A 1.3-GHz 350-mW hybrid direct digital frequency synthesizer in 90-nm CMOS, *IEEE Journal of solid-state circuits*, 45 (9), 1845-1855.
- Yuan, L., Zhang, Q. ve Shi, Y., 2015, A 2GHz direct digital frequency synthesizer based on multi-channel structure, *Circuits and Systems (ISCAS), 2015 IEEE International Symposium on*, 3064-3067.

## ÖZGEÇMİŞ

### KİŞİSEL BİLGİLER

**Adı Soyadı** : Yunus Emre ACAR  
**Uyruğu** : TC  
**Doğum Yeri ve Tarihi** : Ankara, 21.07.1990  
**Telefon** : 332 223 33 49  
**Faks** :  
**e-mail** : yacar@selcuk.edu.tr

### EĞİTİM

| Derece        | Adı, İlçe, İl                                 | Bitirme Yılı |
|---------------|-----------------------------------------------|--------------|
| Lise          | : İbn-i Sina Lisesi, Sincan, Ankara           | 2008         |
| Üniversite    | : Eskişehir Osmangazi Üniversitesi, Eskişehir | 2013         |
| Yüksek Lisans | : Selçuk Üniversitesi, Selçuklu, Konya        | -            |
| Doktora       | :                                             |              |

### İŞ DENEYİMLERİ

| Yıl  | Kurum                                                | Görevi    |
|------|------------------------------------------------------|-----------|
| 2013 | Savronik A.Ş.                                        | Mühendis  |
| 2014 | TCDD                                                 | Mühendis  |
| 2015 | Selçuk Üniversitesi Teknoloji Fak. Elk. ve Elt. Müh. | Arş. Gör. |

### UZMANLIK ALANI

Haberleşme Sistemleri

### YABANCI DİLLER

İngilizce

### YAYINLAR

Acar Y.E., Yıldız E., Direct Digital Frequency Synthesizer Designs in MATLAB, *International Journal of Applied Mathematics, Electronics and Computers (IJAMEC)*, 4(1), 2016.

Acar Y.E., Yıldız E., A high resolution DDFS design on VHDL using Bipartite Table Method, *International Conference on Engineering and Technology and Innovation (ICETI)*, March 22-26, 2016, Sarajevo, Bosnia and Herzegovina.