

TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**ALÇAK YÖRÜNGE UYDU HABERLEŞMESİ İÇİN UYARLAMALI
KODLAMA VE KİPLEME KULLANAN FPGA TABANLI VERİCİ TASARIMI**

YÜKSEK LİSANS TEZİ

ERKAN İNCEÖZ

Elektrik ve Elektronik Mühendisliği Anabilim Dalı

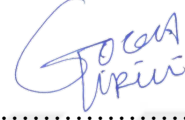
Tez Danışmanı: Doç. Dr. Ayşe Melda YÜKSEL TURGUT

EYLÜL 2020

Fen Bilimleri Enstitüsü Onayı

.....
Prof. Dr. Osman EROĞUL
Müdür

Bu tezin Yüksek Lisans derecesinin tüm gereksinimlerini sağladığını onaylıyorum.



.....
Prof. Dr. Tolga GİRİCİ
Anabilim Dalı Başkanı

TOBB ETÜ, Fen Bilimleri Enstitüsü'nün 171211124 numaralı Yüksek Lisans Öğrencisi **Erkan İNCEÖZ** 'ün ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı "**ALÇAK YÖRÜNGE UYDULARI İÇİN UYARLAMALI KODLAMA VE KİPLEME KULLANAN FPGA TABANLI VERİCİ TASARIMI**" başlıklı tezi **24.09.2020** tarihinde aşağıda imzaları olan jüri tarafından kabul edilmiştir.

Tez Danışmanı : **Doç. Dr. Ayşe Melda YÜKSEL TURGUT**.....
TOBB Ekonomi ve Teknoloji Üniversitesi



Jüri Üyeleri : **Doç. Dr. Ali BOZBEY (Başkan)**
TOBB Ekonomi ve Teknoloji Üniversitesi



Dr. Öğr. Üyesi Barış YÜKSEKKAYA
Hacettepe Üniversitesi



TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, alıntı yapılan kaynaklara eksiksiz atıf yapıldığını, referansların tam olarak belirtildiğini ve ayrıca bu tezin TOBB ETÜ Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırlandığını bildiririm.

Erkan İnceöz



ÖZET

Yüksek Lisans Tezi

ALÇAK YÖRÜNGE UYDU HABERLEŞMESİ İÇİN UYARLAMALI KODLAMA VE KİPLEME KULLANAN FPGA TABANLI VERİCİ TASARIMI

Erkan İnceöz

TOBB Ekonomi ve Teknoloji Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik ve Elektronik Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Ayşe Melda Yüksel Turgut

Tarih: Eylül 2020

Bu tez çalışmasında, alçak yörünge uydu haberleşmesi için yüksek hızlı ve uyarlamalı kodlama ve kipleme kullanan alan-programlanabilir kapı dizisi (field-programmable gate array, FPGA) tabanlı verici tasarımı yapılmıştır. Alçak yörünge uydularının yörünge hareketleri ve dünyanın kendi eksenini etrafında dönmesi nedeniyle haberleşme kanalı değişkenlik göstermektedir. Bu değişken ortamda verimli ve kesintisiz bir haberleşme sağlamak için uyarlamalı kodlama ve kipleme kullanan bir verici yapısı tasarlanmıştır. Tasarlanan uyarlamalı kodlama ve kipleme kullanan verici yapısı MATLAB ile modellenmiştir. MATLAB ile modellenen tasarımın, MATLAB modeli referans alınarak sayısal tasarımı yapılmıştır. Yapılan sayısal tasarım davranışsal benzetim ortamında referans MATLAB modeli ile doğrulanmıştır. Davranışsal olarak yapılan doğrulamalardan sonra sayısal tasarım, programlanabilir bir donanım olan FPGA üzerinde gerçekleştirilmiştir. Yüksek performans ve uzay uygulamalarına uygun eşleşiminin 2020 sonuna kadar üretilecek olması sebebiyle Xilinx Kintex Ultrascale xcku060 FPGA üzerinde gerçekleştirilmiştir. Tasarlanan uyarlamalı kodlama ve kipleme kullanan verici yapısının teorik başarımları, ulaşılan kod oranları ve spektral verimlilik sonuçları çıkarılmıştır. Yapılan tasarımın çalışabildiği en yüksek hızda spektral verimlilik olarak 7,068

değerine ulaşabildiği görülmüştür. Donanım gerçeklemeleri sonucu tasarımın zamanlama analizleri yapılmıştır. Yapılan zamanlama analizleri ile çalışılabilecek en yüksek sistem saat sinyali frekansı belirlenmiştir. Çalışılan saat sinyali frekansları kullanılarak sistemin veri hızları çıkarılmıştır. Geliştirilen uyarlamalı kodlama ve kipleme kullanan ve xcku060 FPGA üzerinde gerçekleştirilen verici yapısı çıkış sembol hızı olarak 75 Megasembol/saniye (megasymbols per second, Msps), çıkış örnek hızı olarak 300 Megaörnek/saniye (megasamples per second, MSps) hızlarına ulaşabildiği görülmüştür. Tasarım çalışma moduna göre 50,7225 ve 530,1 Megabit/saniye (megabits per second, Mbps) arasında değişebilen giriş hızlarını desteklemektedir.

Anahtar Kelimeler: Alçak yörünge uydu haberleşmesi, Uyarlamalı kodlama ve kipleme, Uzay veri sistemleri danışma komitesi 131.2-B-1, Alan-programlanabilir kapı dizileri.

ABSTRACT

Master of Science

FPGA BASED TRANSMITTER DESIGN USING ADAPTIVE CODING AND MODULATION FOR LOW EARTH ORBIT SATELLITE COMMUNICATIONS

Erkan İnceöz

TOBB University of Economics and Technology
Institute of Natural and Applied Sciences
Electrical and Electronics Engineering Science Program

Supervisor: Assoc. Dr. Ayşe Melda Yüksel Turgut

Date: September 2020

In this thesis, a field-programmable gate array (FPGA) based design of high rate transmitter using adaptive coding and modulation for low earth orbit satellite communications is implemented. The communication channel varies due to orbital movements of low earth orbit satellites and the rotation of the earth around its own axis. To provide efficient and uninterrupted communication in this varying environment, a transmitter structure employing adaptive coding and modulation has been designed. The developed transmitter structure with adaptive coding and modulation is modelled in MATLAB. Then, the digital design of the developed structure is done based on the MATLAB model. This design is verified with reference MATLAB design in behavioral simulations. After behavioral simulations are done, the design is implemented on an FPGA. The FPGA implementation is done on Xilinx Kintex Ultrascale xcku060 FPGA because it provides high performance, and its space grade model will be on product line by the end of 2020. The coding rates achieved, and the spectral efficiency results are obtained for the designed transmitter structure employing adaptive modulation and coding. It is observed that its spectral efficiency is 7,068 at the highest speed it can operate. Timing analysis of the design was also made as a result of the hardware implementation. With the

timing analysis, the highest system clock signal frequency that it can operate on has been determined. The data rates of the system are derived by using the clock signal frequencies. It is observed that the new transmitter structure implemented on xcku060 FPGA can reach an output rate of 75 Megasymbols per second (MSPs) and output sample rate of 300 Megasamples per second (MSps). It supports input speeds varying between 50.7225 and 530.1 Megabits per second (Mbps) according to the design's operating mode.

Keywords: Low earth orbit satellite communications, Adaptive coding and modulation, Consultative committee for space data systems 131.2-B-1, Field-programmable gate array.



TEŞEKKÜR

Çalışmalarım ve yüksek lisans eğitimim boyunca değerli yardım ve katkılarıyla beni yönlendiren, cesaretlendiren ve destekleyen saygı değer hocam Doç. Dr. Ayşe Melda YÜKSEL TURGUT'a sonsuz teşekkürü bir borç bilirim. Yüksek lisans eğitimim boyunca kıymetli tecrübelerinden faydalandığım TOBB Ekonomi ve Teknoloji Üniversitesi Elektrik ve Elektronik Mühendisliği Bölümü öğretim üyelerine ve sağladığı araştırma bursu ile çalışmalarına destek olan TOBB Ekonomi ve Teknoloji Üniversitesine çok ederim.

Yüksek lisans çalışmamın fikri olarak ortaya çıkmasında ve olgunlaşmasında öncü olan ve tüm çalışmam boyunca desteklerini esirgemeyen iş arkadaşlarım Dr. Raşit TUTGUN, Koray KARAKUŞ, Dr. Ahmed Yasir DOĞAN ve Dr. Ayşe Neslin İSMAİLOĞLU'na çok teşekkür ederim. Kurum kapsamında geliştirilen iç projelerle çalışma imkanlarımı artıran ve çalışanı olarak desteklerinden faydalandığım TÜBİTAK Uzun Teknolojileri Araştırma Enstitüsü'ne teşekkür ederim.

Uzun yıllardır yanımda olan ve beni destekleyen sevgili dostlarım Ufuk KAYABAŞI, Sezgin DEMİRTAŞ, Hikmet Kaan KIRCI ve Mehmet BAYAR'a çok teşekkür ederim.

Son olarak, tüm çalışmalarım boyunca beni destekleyen ve her daim yanımda olan sevgili eşim Özge İNCEÖZ'e, benim bu günlere gelmemde emeklerini esirgemeyen değerli babam Nihat İNCEÖZ, annem Emine İNCEÖZ ve abim Serkan İNCEÖZ'e çok teşekkür ederim.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	vii
ABSTRACT	ix
TEŞEKKÜR	xi
İÇİNDEKİLER	xiii
ŞEKİL LİSTESİ	xvii
ÇİZELGE LİSTESİ	xix
KISALTMALAR	xxi
SEMBOL LİSTESİ	xxiii
1. GİRİŞ	1
1.1 Motivasyon.....	1
1.2 Tezin Katkısı	2
1.3 Tez Organizasyonu.....	3
2. TEORİK ALTYAPI	5
2.1 Alçak Yörünge Uyduları	5
2.2 Sayısal Haberleşme Sistemi	8
2.3 Alan-Programlanabilir Kapı Dizileri.....	9
2.3.1 FPGA iç yapısı	10
2.3.2 FPGA çeşitleri.....	10
2.3.3 FPGA ile tasarım.....	12
3. UYARLAMALI KODLAMA VE KİPLEME KULLANAN VERİCİ YAPISI	15
3.1 Mod Uyarlama.....	15
3.1.1 Rastgeleleştirme	15
3.1.2 Bölücü	16
3.2 Seri Birleştirilmiş Evrişimsel Kodlayıcı	17
3.2.1 Sistematik evrişimsel kodlayıcı	17
3.2.1.1 Vektör gösterimi.....	19
3.2.1.2 Durum diyagramı gösterimi	19
3.2.1.3 Kafes yapısı.....	20
3.2.2 Delme	20
3.2.3 Serpiştirme	21
3.3 Döngüsel blok kodlayıcı.....	22
3.3.1 Galois cismi.....	22
3.3.2 Doğrusal blok kodlar.....	25
3.3.2.1 Döngüsel blok kodlar	26
Sistematik döngüsel blok kodlar	26
Sistematik döngüsel blok kodlar için kodlayıcı yapısı.....	27
3.3.3 Reed-Solomon kodları	28
3.4 Fiziksel Katman Çerçeveleme.....	29
3.4.1 Kipleme	29
3.4.1.1 M-PSK	29
QPSK	30

8-PSK	31
3.4.1.2 M-APSK.....	31
16-APSK	32
32-APSK	33
64-APSK	34
3.4.1.3 M-QCI.....	36
128-QCI.....	37
256-QCI.....	37
3.4.2 Çerçeve başlığı ekleme.....	37
3.4.2.1 Çerçeve belirteci.....	37
3.4.2.2 Çerçeve tanımlayıcı.....	39
3.4.2.3 Çerçeve başlığı kiplemesi.....	41
3.4.3 Pilot işareti ekleme	41
3.4.4 Fiziksel katman rastgeleleştirme	41
3.5 Tabanbant Süzme	43
3.5.1 Darbe şekillendirici süzgeç	43
3.5.1.1 Kök yükseltilmiş kosinüs süzgeci	43
4. UYARLAMALI KODLAMA VE KİPLEME KULLANAN VERİCİ FPGA	
TASARIMI	47
4.1 Mod Uyarlama.....	47
4.1.1 Giriş arayüzü	47
4.1.2 Rastgeleleştirme	48
4.1.3 Bölücü	49
4.2 Kodlayıcı	50
4.2.1 Kodlayıcı bilgi blokları oluşturma	50
4.2.2 Seri birleştirilmiş evrişimsel kodlayıcı.....	53
4.2.2.1 Dış evrişimsel kodlayıcı	53
4.2.2.2 Sabit oranlı delme.....	55
4.2.2.3 Rastgele serpiştirme	57
4.2.2.4 İç evrişimsel kodlayıcı	57
4.2.2.5 Sistematik bitler için delme.....	59
4.2.2.6 Eşlik denetim bitleri için delme	62
4.2.2.7 Satır-sütun serpiştirme.....	64
4.2.3 Döngüsel blok kodlayıcı.....	67
4.2.3.1 Reed-Solomon kodlayıcı	67
Sembol üretici.....	67
Reed-Solomon (127,120)	69
Galois cismi çarpma	70
Bit üretici	73
Kod sözcüğü doldurma.....	73
4.2.3.2 RS satır-sütun serpiştirme	76
4.2.4 Kod sözcüğü bloklarını birleştirme	78
4.3 Fiziksel Katman Çerçeveleme	81
4.3.1 Kipleme	81
4.3.2 Pilot işareti ekleme	83
4.3.3 Fiziksel katman rastgeleleştirme	85
4.3.4 Çerçeve başlığı ekleme.....	85
4.4 Tabanbant Süzme	88
4.5 FPGA Tasarım İyileştirmeleri	90
5. TASARIM BAŞARIMLARI VE FPGA GERÇEKLEME SONUÇLARI.....	95

5.1 Tasarım Başarımları	95
5.2 FPGA Davranışsal Benzetim Sonuçları	100
5.3 FPGA Seçimi ve Gerçekleme Sonuçları	100
5.3.1 FPGA seçimi	100
5.3.2 Kaynak kullanımı	102
5.3.3 Güç tüketimi.....	102
5.3.4 Zamanlama analizi	103
6. DEĞERLENDİRME VE GELECEK ÇALIŞMALAR.....	105
KAYNAKLAR	107
EKLER.....	109
ÖZGEÇMİŞ.....	125



ŞEKİL LİSTESİ

Sayfa

Şekil 1.1 : Alçak yörünge uydusu ve yer istasyonu haberleşmesi.	2
Şekil 2.1 : Dairesel bir alçak yörünge.	7
Şekil 2.2 : Yükselme açısına bağlı boş uzay kaybı.	7
Şekil 2.3 : Sayısal haberleşme sistemi temel elemanları.....	9
Şekil 2.4 : FPGA iç yapısı.....	11
Şekil 2.5 : FPGA ters sigorta yapısı (a) programlamadan önce (b) programlamadan sonra.	13
Şekil 2.6 : FPGA tasarım akışı.....	14
Şekil 3.1 : Uyarlamalı kodlama ve kipleme kullanan verici yapısı.....	16
Şekil 3.2 : Rastgeleleştirici yapısı.	17
Şekil 3.3 : Bölücü yapısı.	18
Şekil 3.4 : Sistemik evrimsel kodlayıcı yapısı.....	19
Şekil 3.5 : Sistemik evrimsel durum diyagramı.	20
Şekil 3.6 : Sistemik evrimsel kodlayıcı kafes yapısı.	21
Şekil 3.7 : Rastgele serpiştirici yapısı.	23
Şekil 3.8 : Satır-sütun serpiştirici yapısı.	23
Şekil 3.9 : Polinom bölme işlemi.	28
Şekil 3.10 : Sistemik döngüsel blok kodlayıcı yapısı.	29
Şekil 3.11 : QPSK yıldız diyagramı.....	31
Şekil 3.12 : 8-PSK yıldız diyagramı.	32
Şekil 3.13 : 16-APSK yıldız diyagramı.	33
Şekil 3.14 : 32-APSK yıldız diyagramı.	34
Şekil 3.15 : 64-APSK yıldız diyagramı.	35
Şekil 3.16 : Kare ve daire dairesel eşörüntülü eşleme.	38
Şekil 3.17 : 128-QAM ve 128-QCI yıldız diyagramları.	38
Şekil 3.18 : 256-QAM ve 256-QCI yıldız diyagramları.	39
Şekil 3.19 : Fiziksel katman çerçeve yapısı.	40
Şekil 3.20 : Çerçeve belirteci üretimi.....	40
Şekil 3.21 : Pilot sembollerinin yerleşimi.	42
Şekil 3.22 : Fiziksel katman rastgeleleştirici yapısı.	44
Şekil 3.23 : RRC süzgeci farklı azalma faktörlerine göre frekansı yanıtı.....	45
Şekil 4.1 : Fiziksel katman rastgeleleştirici yapısı.	48
Şekil 4.2 : Rastgeleleştirme bloğu FPGA tasarımı arayüz sinyalleri.	49
Şekil 4.3 : Bölücü bloğu FPGA tasarımı arayüz sinyalleri.	51
Şekil 4.4 : Kodlayıcı bilgi blokları oluşturma bloğu FPGA tasarımı arayüz sinyalleri.....	52
Şekil 4.5 : Seri birleştirilmiş evrimsel kodlayıcı bloğu FPGA tasarımı arayüz sinyalleri.....	54
Şekil 4.6 : Dış evrimsel kodlayıcı bloğu FPGA tasarımı arayüz sinyalleri.....	54
Şekil 4.7 : Sabit oranlı delme bloğu FPGA tasarımı arayüz sinyalleri.	56
Şekil 4.8 : Sabit oranlı delme bloğu zamanlama diyagramı.	56

Şekil 4.9 : Rastgele serpiştirme bloğu FPGA tasarımı arayüz sinyalleri.	58
Şekil 4.10 : İç evrişimsel kodlayıcı bloğu FPGA tasarımı arayüz sinyalleri.	60
Şekil 4.11 : Sistematik bitler için delme bloğu FPGA tasarımı arayüz sinyalleri.....	61
Şekil 4.12 : Eşlik denetim bitleri için delme bloğu FPGA tasarımı arayüz sinyalleri.	64
Şekil 4.13 : Satır-sütun serpiştirme bloğu FPGA tasarımı arayüz sinyalleri.	66
Şekil 4.14 : Reed-Solomon kodlayıcı bloğu FPGA tasarımı arayüz sinyalleri.....	68
Şekil 4.15 : Sembol üretici bloğu FPGA tasarımı arayüz sinyalleri.....	69
Şekil 4.16 : Reed-Solomon (127,120) bloğu FPGA tasarımı arayüz sinyalleri.	70
Şekil 4.17 : Galois cismi çarpma bloğu FPGA tasarımı arayüz sinyalleri.	71
Şekil 4.18 : Bit üretici bloğu FPGA tasarımı arayüz sinyalleri.	74
Şekil 4.19 : Kod sözcüğü doldurma bloğu FPGA tasarımı arayüz sinyalleri.	76
Şekil 4.20 : RS satır sütun serpiştirme bloğu FPGA tasarımı arayüz sinyalleri.....	77
Şekil 4.21 : Kod sözcüğü bloklarını birleştirme bloğu FPGA tasarımı arayüz sinyalleri.	80
Şekil 4.22 : Sabit-nokta $Q(v, w)$ formatındaki ikili sayı.....	82
Şekil 4.23 : Kipleme bloğu FPGA tasarımı arayüz sinyalleri.	84
Şekil 4.24 : Pilot işareti ekleme bloğu FPGA tasarımı arayüz sinyalleri.	84
Şekil 4.25 : Fiziksel katman rastgeleleştirme bloğu FPGA tasarımı arayüz sinyalleri.	86
Şekil 4.26 : Çerçeve başlığı ekleme bloğu FPGA tasarımı arayüz sinyalleri.	86
Şekil 4.27 : Kök yükseltilmiş kosinüs süzgeci dürtü yanıtı.	89
Şekil 4.28 : Kök yükseltilmiş kosinüs süzgeci frekans yanıtı.	89
Şekil 4.29 : Tabanbant süzme bloğu FPGA tasarımı arayüz sinyalleri.....	90
Şekil 4.30 : FPGA tasarımı üstblokları.	92
Şekil 4.31 : İyileştirilmiş FPGA tasarımı.	93
Şekil 5.1 : FPGA davranışsal benzetim ortamında çıkış sinyalleri.	101
Şekil 5.2 : FPGA davranışsal benzetim sonuçları.	101
Şekil 5.3 : FPGA davranışsal benzetim sonuçları ve MATLAB sonuçları karşılaştırılması.	102

ÇİZELGE LİSTESİ

Sayfa

Çizelge 3.1 : Sistematik evrişimsel kodlayıcı durum çizelgesi.....	21
Çizelge 3.2 : İkili Galois cisminde toplama işlemi.	24
Çizelge 3.3 : İkili Galois cisminde çarpma işlemi.	24
Çizelge 3.4 : Fiziksel katman rastgeleleştirici.....	44
Çizelge 4.1 : Giriş arayüzü bloğu FPGA tasarımı arayüz sinyal bilgileri.....	48
Çizelge 4.2 : İletim çerçevesi uzunluğu.	48
Çizelge 4.3 : Rastgeleleştirici bloğu FPGA tasarımı arayüz sinyal bilgileri.....	49
Çizelge 4.4 : Bölücü bloğu FPGA tasarımı arayüz sinyal bilgileri.....	51
Çizelge 4.5 : Kodlayıcılar için bilgi blok uzunlukları.....	51
Çizelge 4.6 : Kodlayıcı bilgi blokları oluşturma bloğu FPGA tasarımı arayüz sinyal bilgileri.	52
Çizelge 4.7 : Seri birleştirilmiş evrişimsel kodlayıcı bloğu FPGA tasarımı arayüz sinyal bilgileri.	54
Çizelge 4.8 : Dış evrişimsel kodlayıcı bloğu FPGA tasarımı arayüz sinyal bilgileri.	54
Çizelge 4.9 : Sabit oranlı delme bloğu FPGA tasarımı arayüz sinyal bilgileri.	56
Çizelge 4.10 : Rastgele serpiştirici blok uzunluğu.....	58
Çizelge 4.11 : Rastgele serpiştirme bloğu FPGA tasarımı arayüz sinyal bilgileri.	58
Çizelge 4.12 : İç evrişimsel kodlayıcı bilgi blok uzunluğu.....	59
Çizelge 4.13 : İç evrişimsel kodlayıcı bloğu FPGA tasarımı arayüz sinyal bilgileri.	60
Çizelge 4.14 : Sistematik bitler için delme bloğu silinecek bit sayısı.....	60
Çizelge 4.15 : Sistematik bitler için delme bloğu silinecek bit pozisyonları.	61
Çizelge 4.16 : Sistematik bitler için delme bloğu FPGA tasarımı arayüz sinyal bilgileri.	62
Çizelge 4.17 : Eşlik denetim bitleri için delme bloğu silinecek bit sayısı.	63
Çizelge 4.18 : Eşlik denetim bitleri için delme bloğu FPGA tasarımı arayüz sinyal bilgileri.	64
Çizelge 4.19 : Satır-sütun serpiştirme giriş verisi blok uzunluğu.	65
Çizelge 4.20 : Satır-sütun serpiştirme bloğu FPGA tasarımı arayüz sinyal bilgileri.	66
Çizelge 4.21 : Satır-sütun serpiştirici için sistematik bit ve eşlik denetim bitleri giriş verisi blok uzunlukları.....	67
Çizelge 4.22 : Reed-Solomon kodlayıcı bloğu FPGA tasarımı arayüz sinyal bilgileri.	68
Çizelge 4.23 : Sembol üretici bloğu FPGA tasarımı arayüz sinyal bilgileri.	69
Çizelge 4.24 : Reed-Solomon (127,120) bloğu FPGA tasarımı arayüz sinyal bilgileri.	70
Çizelge 4.25 : Galois cismi çarpma bloğu FPGA tasarımı arayüz sinyal bilgileri. ...	71
Çizelge 4.26 : Bit üretici bloğu FPGA tasarımı arayüz sinyal bilgileri.....	74
Çizelge 4.27 : Reed-Solomon kodlayıcı kod sözcüğü olması gereken blok uzunluğu.	75
Çizelge 4.28 : Reed-Solomon kodlayıcı kod sözcüğü blok uzunluğu.	75
Çizelge 4.29 : Kod sözcüğü doldurma bloğu FPGA tasarımı arayüz sinyal bilgileri.76	

Çizelge 4.30 : RS satır-sütun serpiştirme giriş verisi blok uzunluğu.....	77
Çizelge 4.31 : RS satır-sütun serpiştirme bloğu FPGA tasarımı arayüz sinyal bilgileri.....	77
Çizelge 4.32 : Kiplemede bir sembole eşlenecek bit sayısı.....	79
Çizelge 4.33 : Kiplemede bir sembol için seri birleştirilmiş evrişimsel kodlayıcı ve döngüsel blok kodlayıcıdan alınacak bit sayıları.....	79
Çizelge 4.34 : Kod sözcüğü blokları birleştirme bloğu FPGA tasarımı arayüz sinyal bilgileri.....	80
Çizelge 4.35 : ACM formatına göre kullanılan kipleme.....	82
Çizelge 4.36 : ACM formatına göre kullanılan APSK kiplemelerindeki halkaların yarıçap oranları.....	82
Çizelge 4.37 : Kipleme bloğu FPGA tasarımı arayüz sinyal bilgileri.....	84
Çizelge 4.38 : Pilot işareti ekleme bloğu FPGA tasarımı arayüz sinyal bilgileri.....	84
Çizelge 4.39 : Fiziksel katman rastgeleleştirme bloğu FPGA tasarımı arayüz sinyal bilgileri.....	86
Çizelge 4.40 : Çerçeve başlığı ekleme bloğu FPGA tasarımı arayüz sinyal bilgileri.....	87
Çizelge 4.41 : Tabanbant süzme bloğu FPGA tasarımı arayüz sinyal bilgileri.....	90
Çizelge 5.1 : Uyarlamalı kodlama yapısı değişkenleri.....	96
Çizelge 5.2 : Uyarlamalı kodlama yapısı ile elde edilen kod oranları.....	97
Çizelge 5.3 : Uyarlamalı kipleme yapısı değişkenleri.....	98
Çizelge 5.4 : Uyarlamalı kodlama ve kipleme yapısı ile elde edilen spektral verimlilik.....	99
Çizelge 5.5 : Uyarlamalı kodlama ve kodlama yapısı kullanan verici FPGA tasarımı kaynak kullanımı.....	103
Çizelge 5.6 : Uyarlamalı kodlama ve kodlama yapısı kullanan verici FPGA tasarımı güç tüketimi.....	103
Çizelge 5.7 : Uyarlamalı kodlama ve kipleme kullanan verici yapısı FPGA tasarımı ile elde edilen veri hızları.....	104
Çizelge Ek.1 : Rastgele serpiştirme değişkenleri 1 ($I = 25200, I = 29160, I = 33120, I = 37080, I = 41040$).....	110
Çizelge Ek.2 : Rastgele serpiştirme değişkenleri 2 ($I = 24840, I = 32760, I = 36720$).....	118

KISALTMALAR

ACM	: Adaptive coding and modulation
LEO	: Low earth orbit
Mbps	: Megabits per second
Msps	: Megasymbols per second
MSps	: Megasamples per second
MHz	: Mega Hertz
SNR	: Signal-to-noise ratio
BER	: Bit error rate
CCSDS	: Consultative Committee for Space Data Systems
ESA	: European Space Agency
FPGA	: Field-programmable gate array
PLA	: Programmable logic array
PAL	: Programmable array logic
PLD	: Programmable logic device
SPLD	: Simple programmable logic device
CPLD	: Complex programmable logic device
IOB	: Input/output block
SC	: Switching circuit
SRAM	: Static random-access memory
RTL	: Register transfer level
HDL	: Hardware description language
XOR	: Exclusive OR
SCCC	: Serial concatenated convolutional code
BCH	: Bose-Chadhuri-Hocquenghem
RS	: Reed-Solomon
PSK	: Phase shift keying
QPSK	: Quadrature shift keying
APSK	: Amplitude-phase shift keying
QAM	: Quadrature amplitude modulation
QCI	: QAM to circular isomorphic
BPSK	: Binary phase shift keying
ISI	: Inter symbol interference
RRC	: Root raised cosine
FIFO	: First-in first-out
LUT	: Look-up table
FF	: Flip-flop
BRAM	: Block random access memory
DSP	: Digital signal processor
IO	: Input/output
BUFG	: Global buffer
W	: Watt

SEMBOL LİSTESİ

Bu çalışmada kullanılmış olan simgeler açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	Açıklama
$f(x)$	Rastgeleleştirici kayan yazmaç fonksiyonu
R	Kod oranı
k	Evrişimsel kodlayıcı bilgi biti uzunluğu
n	Evrişimsel kodlayıcı kod sözcüğü bit uzunluğu
c_1	Evrişimsel kodlayıcı sistematik bit çıkışı
c_2	Evrişimsel kodlayıcı eşlik denetim bit çıkışı
z	Evrişimsel kodlayıcı hafıza derecesi
x_1^k	Evrişimsel kodlayıcının k. giriş biti için 1. yazmaç değeri
x_2^k	Evrişimsel kodlayıcının k. giriş biti için 2. yazmaç değeri
c_1^k	Evrişimsel kodlayıcının k. giriş için 1. kanal çıkışı
c_2^k	Evrişimsel kodlayıcının k. giriş için 2. kanal çıkışı
u_k	Evrişimsel kodlayıcının k. giriş biti
$g_1(D)$	Evrişimsel kodlayıcı ileribesleme polinomu
$g_2(D)$	Evrişimsel kodlayıcı geribesleme polinomu
g_1	Evrişimsel kodlayıcı ileribesleme vektörü
g_2	Evrişimsel kodlayıcı geribesleme vektörü
G_R	Evrişimsel kodlayıcı üreteç vektörü
z	Evrişimsel kodlayıcı hafıza derecesi
P	Delme matrisi
I	Serpiştirme blok uzunluğu
α	Serpiştirme adres değişkeni
β	Serpiştirme kaydırma değişkeni
k_s	Serpiştirici giriş blok uzunluğu
m_s	Serpiştirici sütun sayısı
n_s	Serpiştirici satır sayısı
m	Kipleme derecesi
p	Galois cismi eleman sayısı
p^a	Genişletilmiş Galois cismi eleman sayısı
$d(x)$	Galois cismi üzerinde bir polinom
$p(x)$	İlkel polinom
k_b	Blok kodlama bilgi bitleri uzunluğu
n_b	Blok kodlama kod sözcüğü bit uzunluğu
m_i	Blok kodlama i.bilgi serisi
c_i	Blok kodlama i.kod sözcüğü serisi
G	Blok kodlama üreteç matrisi

c_j^i	Döngüsel blok kodlamada j.kod sözcüğü serisinin i bit kaydırılması
$m(x)$	Döngüsel blok kodlama bilgi polinomu
$c(x)$	Döngüsel blok kodlama kod sözcüğü polinomu
$g(x)$	Döngüsel blok kodlama üreteç polinomu
$Q(x)$	Polinom bölmesi bölüm polinomu
$r(x)$	Polinom bölmesi kalan polinomu
k_{RS}	Reed-Solomon kodlama bilgi sembol uzunluğu
n_{RS}	Reed-Solomon kodlama kod sözcüğü sembol uzunluğu
q	Reed-Solomon kodlama sembol bit uzunluğu
$s_i(t)$	Kiplemede i.iletilem sinyal
f_c	Taşıyıcı frekansı
φ_i	Taşıyıcı fazı
T	İletilen sinyalin periyodu
$b_1(t)$	Kiplemede iletilen sinyal için tanımlanan 1.taban sinyali
$b_2(t)$	Kiplemede iletilen sinyal için tanımlanan 2.taban sinyali
s_{i1}	Kiplemede i.iletin sinyalin 1.taban sinyali üzerindeki izdüşümü
s_{i2}	Kiplemede i.iletin sinyalin 2.taban sinyali üzerindeki izdüşümü
E	Kiplemede iletim sinyal enerjisi
A	Kiplemede iletim sinyal genliği
M	Kiplemede iletim farklı sembol sayısı
R_i	APSK kiplemesinde i.halka yarıçapı
A_i	Dördün genlik kiplemesinde eş fazlı genlik bileşeni
A_q	Dördün genlik kiplemesinde dördün genlik bileşeni
$f(x, y)$	Dairesel eşörüntülü eşleme fonksiyonu
$h_1(x)$	Çerçeve belirteci 1.geribesleme polinomu
$h_2(x)$	Çerçeve belirteci 2.geribesleme polinomu
m_{ct}	Çerçeve tanımlayıcı bilgi bit serisi
b_i	Çerçeve tanımlayıcı i.bilgi biti
G_{ct}	Çerçeve tanımlayıcı üreteç matrisi
c_{ct}	Çerçeve tanımlayıcı kod sözcüğü bit serisi
t_i	Çerçeve başlığı i.biti
C_I	Fiziksel katman rastgeleleştirme karmaşık seri gerçel katsayısı
C_Q	Fiziksel katman rastgeleleştirme karmaşık seri karmaşık katsayısı
$g_x(x)$	Fiziksel katman rastgeleleştirme x rastgele serisi üreteç polinomu
$g_y(y)$	Fiziksel katman rastgeleleştirme y rastgele serisi üreteç polinomu
Z	Altın seri
α	Kök yükseltilmiş kosinüs süzgeci azaltma faktörü
f_N	Nyquist frekansı
T_s	Sembol süresi
R_s	Sembol hızı
Δ	Silinecek eşlik denetim biti sayısı

S	İletilecek sistematik bit blok uzunluğu
P	İletilecek eşlik denetim bit blok uzunluğu
$s_1(x)$	Galois cismi çarpma 1.çarpan
$s_2(x)$	Galois cismi çarpma 2.çarpan
α_i	Galois cismi çarpma 1.çarpanın i.derece değişkeni katsayısı
β_i	Galois cismi çarpma 2.çarpanın i.derece değişkeni katsayısı
$m_{GF}(x)$	Galois cismi çarpma bloğunda çarpanlarının çarpım ara polinomu
γ_i	Galois cismi çarpma bloğunda çarpım sonucunun i.derece değişkeni katsayısı
m_{SCCC}	Kipleme için seri birleştirilmiş evrişimsel kodlayıcıdan alınan bit sayısı
m_{RS}	Kipleme için döngüsel blok kodlayıcıdan alınan bit sayısı
X_{kn}	X değişkeni kayan nokta gösterimi
X_{sn}	X değişkeni sabit-nokta gösterimi
v	Sabit-nokta gösterimi tamsayı bit uzunluğu
w	Sabit-nokta gösterimi ondalık bit uzunluğu
sps	Kök yükseltilmiş kosinüs süzgeci için yukarı örnekleme değeri
$span$	Kök yükseltilmiş kosinüs süzgeci için örnek tarama genişliği
K_{SCCC}	Seri birleştirilmiş evrişimsel kodlayıcı bilgi blok uzunluğu
K_{RS}	Döngüsel blok kodlayıcı bilgi blok uzunluğu
N_{SCCC}	Seri birleştirilmiş evrişimsel kodlayıcı kod sözcüğü blok uzunluğu
N_{RS}	Döngüsel blok kodlayıcı kod sözcüğü blok uzunluğu
r_{SCCC}	Seri birleştirilmiş evrişimsel kodlayıcı kod oranı
r_{RS}	Döngüsel blok kodlayıcı kod oranı
r	Kodlayıcı toplam kod oranı
M_{ic}	İletim çerçevesi uzunluğu

1. GİRİŞ

Günümüzde alçak yörünge uydularının önemi ve kullanım alanları giderek artmaktadır. Alçak yörünge uyduları hava olaylarını inceleme, hava durumu tahminleri, coğrafi değişimleri gözleme, doğal afet sonrası hasar tespitleri, askeri ve sivil amaçlı yer görüntüleme ve gözleme amaçlarıyla kullanılmaktadır [1]. Alçak yörünge uydularından alınan veriler yer istasyonları tarafından indirilerek işlenir.

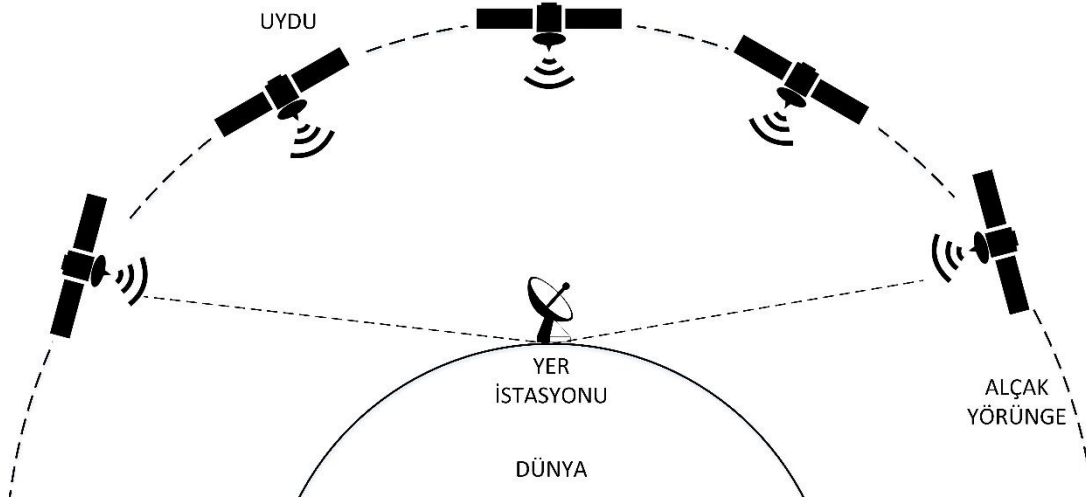
Alçak yörünge uyduları dünyaya yakın olması sebebiyle yüksek bir hızda dünya etrafında tur atar. Yörünge üzerindeki hareketinin periyodunun düşük olması ve dünyanın kendi eksenini etrafındaki dönüşü sayesinde tüm dünya üzerinde bir kapsama alanı oluşturur.

1.1 Motivasyon

Alçak yörünge uyduları ve yer istasyonu arasındaki haberleşme uydunun yer istasyonu yakınından geçtiği sürelerde olur. Uydu ufuk çizgisinden belirli bir yükselme açısına sahip olduğunda haberleşme başlar. Bu haberleşme süresi boyunca uydu yer istasyonuna yaklaşır ve daha sonra tekrar uzaklaşır. Yükselme açısı tekrar belirli bir değerin altına geldiğinde haberleşme sona erer. Uydu yörüngesi boyunca topladığı tüm verileri bu kısıtlı sürede yer istasyonuna gönderir.

Bir alçak yörünge uydusunun sabit bir noktada yer alan yer istasyonu ile haberleşmesi Şekil 1.1’de gösterilmiştir.

Uydu ve yer istasyonu arasındaki haberleşme başarımı için en önemli kriter bit hata olasılığı (bit error rate, BER) değeri ve bu değeri doğrudan etkileyen sinyal-gürültü oranı (signal-to-noise ratio, SNR) değeridir. Uydu haberleşmesi için SNR değeri boş uzay kaybına ve atmosferik kayıplara işlem kayıplarına bağlıdır. Boş uzay kaybı uydu ve yer istasyonu arasındaki uzaklığa bağlı olarak haberleşme başlangıcında artmaya ve belirli bir seviyeden sonra azalmaya başlar. Bunun yanı sıra, uydu yörüngesinin dünyaya yakın olması sebebiyle atmosferik kayıpların etkisi daha fazla



Şekil 1.1 : Alçak yörünge uydusu ve yer istasyonu haberleşmesi.

görülmür. Bu sebeplerden dolayı uydu haberleşmesi süresinde SNR değeri değişkendir [2].

Gelişen uydu ve görüntüleme teknolojileri ile alçak yörünge uydularında görev kapsamında toplanan verilerin büyüklüğü giderek artmaktadır. Veri yükünün artmasıyla birlikte kısa süreli olan haberleşmede tüm verinin aktarılması zorlaşmaktadır. Değişken SNR değerine sahip bir haberleşme sisteminde verimli haberleşebilmek için uyarlamalı kodlama ve kiplmeli yapıların kullanımı gerekmektedir.

1.2 Tezin Katkısı

Alçak yörünge uyduları için uyarlamalı kodlama ve kiplmeli haberleşme yapıları üzerine çalışmalar çok eskiye dayanmamaktadır. Alçak yörünge uydu haberleşmesi için kullanılacak yüksek veri hızı ve uyarlamalı yapıya sahip haberleşme sistemi gelişimi [3]'te verilen çalışma ile somutlaşmıştır. Bu çalışmayla birlikte 27-modlu uyarlamalı kodlama ve kiplleme yapısı kullanan alçak yörünge uyduları için bir verici yapısı önerilmiştir.

[3]'te verilen çalışmanın ışığında Consultative Committee for Space Data Systems (CCSDS) [4] tarafından önerilen yapının detayları belirlenerek standartlaştırılmıştır [5]. Bu standartlaşma doğrultusunda bu yapının bazı bölümlerinin veya tamamının geliştirme çalışmaları yapılmıştır [6-9].

[10]'da verilen çalışma ile standartlaşmış uyarlamalı kodlama ve kiplmeli yapının

veriminin artırılması ve genişletilmesi önerilmiştir. Bu öneri kapsamında yeni kodlama ve kipleme şemaları eklenerek daha önce standart ile tanımlanmış 27 modlu verici yapısı genişletilip 37 modlu bir verici yapısı önerilmiştir. Avrupa Uzay Ajansı, bu öneri doğrultusunda standardı güncelleme çalışmaları başlatmıştır.

Bu tez çalışması kapsamında, CCSDS 131.B-2.1 standardı [5] ile tanımlanan verici yapısının, [10]'da önerilen yapıya göre güncellenmesi ile oluşturulan uyarlamalı kodlama ve kipleme yöntemleri kullanan 37-modlu bir verici yapısı geliştirilmiştir. İlk aşama olarak [10]'da önerilen verici yapısı incelenmiştir. Verici yapısındaki belirlenmemiş parametreler ve tanımlanmamış yapılar üzerine çalışılmıştır. Bu çalışmada, [10]'da verilen tavsiyelere göre seri birleştirilmiş evrimsel kodlayıcı için rastgele serpiştirme bloğu parametreleri, sistematik bitleri ve eşlik denetim bitlerini delme bloklarındaki kod oranları ve delme parametreleri önerilmiştir. [10]'da verilen tavsiyeler ve kod sözcüğü blok uzunları dikkate alınarak döngüsel blok kodlayıcı olarak, verilen sabit kod sözcüğü uzunlukları için en yüksek kod oranını sağlayan Reed-Solomon kodlayıcı önerilmiştir. Oluşturulan yeni modlar sebebiyle, fiziksel katman çerçeveleme bloğundaki çerçeve tanımlayıcı için yeni modları da kapsayan bir yapı önerilmiş ve uyarlamalı kodlama ve kipleme kullanan 37-modlu verici yapısı tamamlanmıştır. Oluşturulan yeni yapının MATLAB ortamında modellenmesi ve benzetimleri yapılmıştır. MATLAB modeli referans alınarak Verilog donanım tanımlama dili aracılığıyla uyarlamalı kodlama ve kipleme kullanan verici yapısının sayısal tasarımı yapılmıştır. Sayısal tasarım ve MATLAB modeli benzetim ortamında birebir doğrulanmıştır. Sayısal tasarım programlanabilir bir donanım olan FPGA üzerinde gerçekleştirilmiş ve bu yapıya ait teorik ve donanımsal başarımlar çıkarılmıştır.

Bu çalışma, [10]'da önerilen yapının gerçekleştirilmesi için gereken parametrelerin ve blokların önerilmesi, bu yapının bilindiği kadarıyla ilk kez MATLAB ile modellenmesi, ilk kez sayısal tasarımının yapılması ve veri hızları türünden performansının ilk kez sunulması açısından önem taşımaktadır.

1.3 Tez Organizasyonu

Bu tez çalışmasının ikinci bölümünde yapılan çalışma için gerekli olan teorik altyapı verilmiştir. Tasarlanan sistemin gerekliliği, çalışma senaryosu ve kullanılan donanım hakkında genel bilgiler verilmiştir.

Üçüncü bölümde geliştirilen uyarlamalı kodlama ve kipleme kullanan verici yapısı açıklanmıştır. Kullanılan blokların kullanım amaçları, kullanım şekilleri ve yapıları detaylı olarak açıklanmıştır.

Bu tez çalışması kapsamında yapılan FPGA tasarımları dördüncü bölümde açıklanmıştır. Tasarımda kullanılan bloklar, bu bloklara ait arayüzler ve blokların çalışma prensipleri detaylı olarak bu bölümde açıklanmıştır.

Beşinci bölümde geliştirilen yapının teorik başarımları, sayısal tasarım ve referans MATLAB modeli ile yapılan doğruma sonuçları ve FPGA tasarımının gerçekleştirme sonuçları verilmiştir.

Altıncı bölümde yapılan çalışmanın genel olarak değerlendirilmesi ve gelecekte yapılabilecek çalışmalardan bahsedilmesiyle birlikte tez sonlandırılmıştır.

2. TEORİK ALTYAPI

2.1 Alçak Yörünge Uyduları

Uydu haberleşme sistemleri, uzay teknolojileri ve haberleşme teknolojileri alanlarının ortak çalışmaları sonucu ortaya çıkan sistemlerdir. Uydu haberleşme sistemlerinin temeli 1945'te Arthur C. Clarke tarafından yazılan ve Wireless World dergisinde yayınlanan bir makaleye dayanmaktadır [11]. Makalenin yazıldığı dönemde bu bulguları gerçekleyecek teknoloji olmasa da yaptığı çalışmaların ışığında 1957 yılında Sputnik 1 uydusu Sovyet Sosyalist Cumhuriyetler Birliği tarafından uzaya gönderilmiş ve dünya ülkeleri arasındaki uzay yarışı başlamıştır [12].

Uydular genellikle televizyon yayınları, haberleşme sistemleri, konumlama sistemleri, hava durumu tahminleri, yer gözlem ve görüntüleme görevlerinde kullanılmaktadır [13].

Uyduların konumlandırıldığı yörüngeler temel olarak alçak yörünge, orta yörünge ve yerdurağan yörünge olarak 3 gruba ayrılabilir.

Yerdurağan yörünge veya yere eşzamanlı yörünge olarak adlandırılan yörüngelerin dünya yüzeyinden uzaklığı yaklaşık olarak 35786 kilometredir [14]. Bu yörünge ve dolayısıyla uydunun periyodu dünyanın periyodu ile aynıdır. Bu sayede bu uyduların kapsama alanları dünya üzerinde hep aynı bölgedir. Bu yörüngeler genellikle televizyon yayınları ve bazı haberleşme sistemleri için kullanılmaktadır.

Orta yörünge olarak adlandırılan yörüngelerin dünya yüzeyinden uzaklığı yaklaşık 10000 km ile 20000 km arasındadır [15]. Orta yörünge uyduları genellikle konumlama sistemlerinde kullanılmaktadır.

Alçak yörüngelerin dünya yüzeyinden uzaklığı yaklaşık olarak 500 km ile 1200 km arasındadır [15]. Alçak yörüngelerde uyduların dünya üzerindeki turları ortalama 120 dakikaya kadar çıkabilmektedir. Van Allen radyasyon kuşağı hemen altında konumlandırıldığı için uydu görev ömürleri genellikle kısadır. Alçak yörünge uyduları genellikle hava durumu tahminleri, doğa olayları sonrası hasar tespitleri, yer

gözlem ve görüntüleme görevlerinde kullanılmaktadır.

Alçak yörünge sınıfına giren güneşe eş zamanlı yörünge uydularının dünya üzerindeki turları yaklaşık 90 dakika civarındadır. Diğer alçak yörüngelerden farklı olarak bu uyduların yörünge hareketleri ve dünyanın kendi eksenini etrafında dönüş hareketi ile tüm dünya üzerinde kapsama sağlar.

Dairesel bir alçak yörünge Şekil 2.1’de verilmiştir.

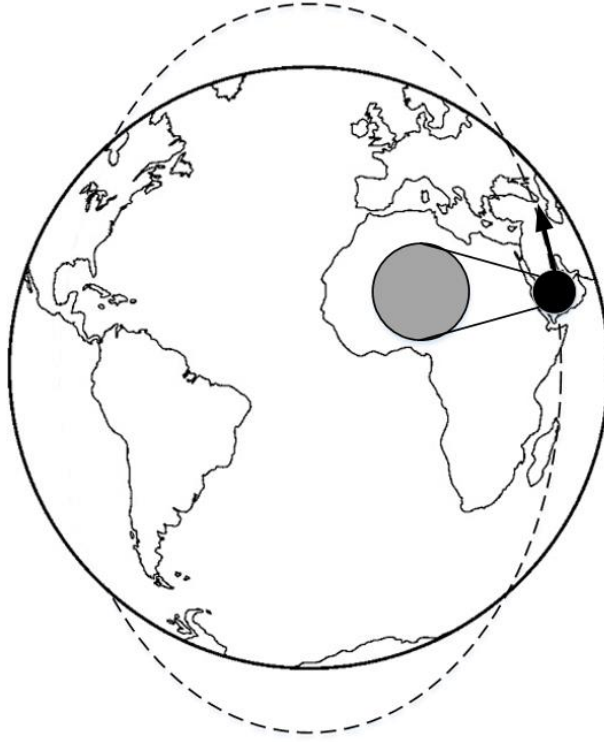
Uydu haberleşme sisteminin başarımı haberleşme süresi boyunca belirli bir BER oranının altına düşmemesi ile belirlenir. Yeterli BER oranının sağlanması SNR değerine bağlıdır. Uydu haberleşme sistemlerinde SNR değerini etkileyen faktörler boş uzay kaybı, atmosferik kayıplar ve işlem kayıplarıdır. Boş uzay kaybı uydu ile yer istasyonu arasındaki görüş hattı uzaklığı ile ilişkilidir.

Uydu ile yer istasyonu arasındaki haberleşme, Şekil 1.1’te gösterildiği gibi uydunun yükselme açısının çok düşük olduğu yerlerde başlar ve tekrar yükselme açısının çok düşük olduğu yerlere kadar devam eder. Bu haberleşme süresi boyunca uydu ile yer istasyonu arasındaki uzaklık önce azalır daha sonra artar. Dünya yüzüne 680 km uzaklıkta bir alçak yörünge uydusu için yükselme açısına bağlı olarak değişen boş uzay kaybı grafiği yaklaşık olarak Şekil 2.2’de verilmiştir.

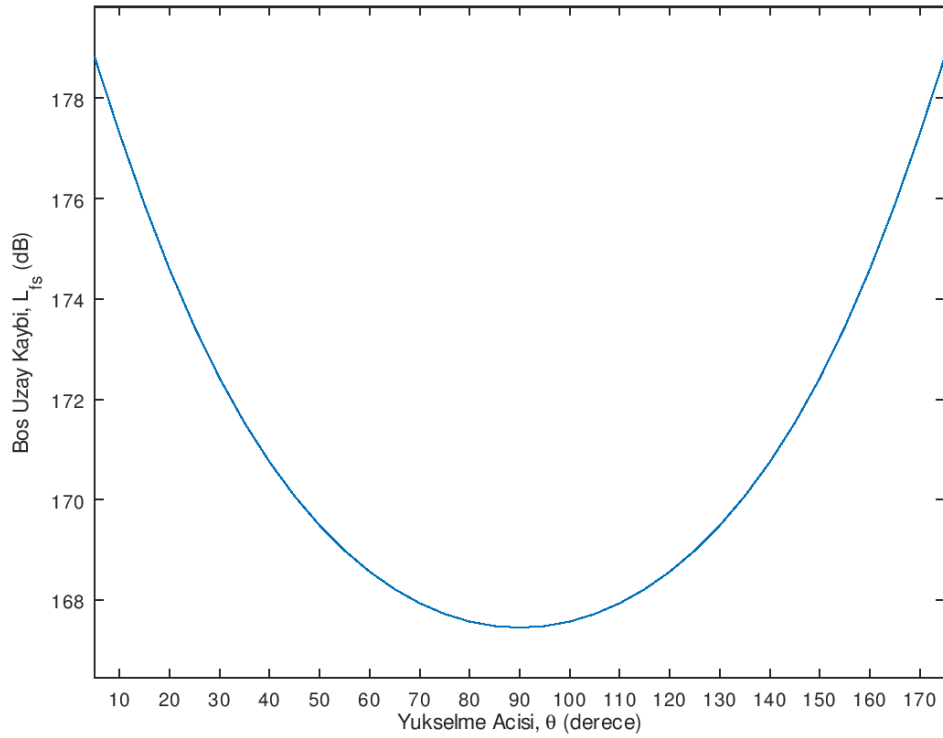
Alçak yörünge uydu haberleşme sisteminde SNR değerini etkileyen diğer etken atmosferik kayıplardır. Atmosferik kayıplar uydu yörüngesinin alçak olması sebebiyle önem kazanmaktadır. Yüksek frekanslarda ise bu etki daha da önemli hale gelmektedir.

Uydu haberleşme sisteminin belirlenen haberleşme süresi boyunca kesintisiz olarak haberleşebilmesi için, bu süre boyunca oluşan en düşük SNR değerinde başarılı haberleşecek şekilde tasarlanması gerekmektedir. En düşük SNR değerinde kesintisiz olarak haberleşebilen bir sistemden yüksek SNR değerlerinde verim alınamamaktadır.

Gelişen uydu platform teknolojileri ve görüntüleme teknolojileri doğrultusunda uydu haberleşmesindeki veri yükü giderek artmaktadır. Dünyanın kendi eksenini etrafındaki dönüşü ve uydunun yörünge üzerindeki hareketi sebebiyle sabit bir yer istasyonu ile haberleşme süresi çok kısadır. Uydu ile alınan ve depolanan tüm veriler kısa haberleşme süresinde yer istasyonuna iletilmek zorundadır. Bu kısa süre içerisinde yüksek miktarda verinin yüksek hızlı ve güvenilir bir haberleşme sistemi üzerinden



Şekil 2.1 : Dairesel bir alçak yörünge.



Şekil 2.2 : Yükselme açısına bağlı boş uzay kaybı.

iletilmesi gerekmektedir. Bu gereksinimler doğrultusunda uyarlamalı haberleşme sistemleri önem kazanmaktadır.

2.2 Sayısal Haberleşme Sistemi

Sayısal haberleşme sistemi fonksiyonel diyagramı ve temel elemanları Şekil 2.3'te verilmiştir [16].

Bilgi kaynağı, iletilecek bilgiyi analog veya sayısal olarak oluşturan bloktur.

Kaynak kodlama, analog veya sayısal olarak oluşturulan bilgiyi ikili sayı dizileri haline getiren bloktur. Bu bloğun çıkışı, bilgi bitleri olarak ifade edilir.

Kanal kodlama, kanaldan kaynaklı kayıplara karşı bilgi bitlerini korumak amacıyla yapılan işlemdir. Bilgi bitlerini korumak amacıyla belirli algoritmalara göre fazladan bitler eklenir. Eklenmiş bitler ve bilgi bitleriyle birlikte kanal kodlama çıkışında kod sözcükleri oluşturulur. Kod sözcüğü uzunluğu bilgi uzunluğundan daha fazladır. Kod sözcüğü uzunluğunun bilgi uzunluğuna oranı kod oranı olarak adlandırılır ve yapılan kodlamanın verimliliğini belirler.

Sayısal kipleme, kanal kodlaması ile oluşturulan kod sözcüğü bitlerinin bir kanal üzerinden iletilebilmesi için belirli dalga formlarına eşlenmesi işlemidir.

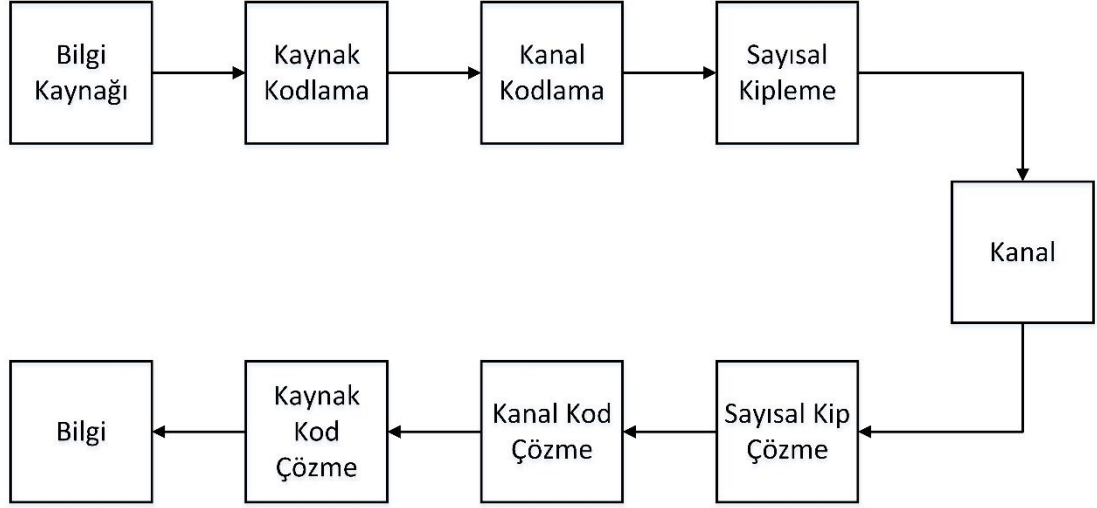
Kanal, dalga formlarının vericiden alıcıya iletildiği ortam olarak tanımlanır. Haberleşme sisteminin kullanım alanına göre bu kanal kablolu ya da kablosuz olabilmektedir. Her kanalın kendisine ve bulunduğu ortama bağlı olarak bozucu etkileri bulunmaktadır.

Sayısal kip çözme, kanal tarafından etkilenmiş dalgaları alarak bu dalga formlarına karşılık gelen tahmini kod sözcüğü bit dizilerini oluşturur.

Kanal kod çözme, elde edilen tahmini kod sözcüğü bit dizilerini alarak, kanal kodlaması sırasında eklenen fazla bitler yardımıyla hatalı bilgi bitlerini olabildiğince düzeltir ve tahmini bilgi bitlerini üretir.

Kaynak kod çözme, elde edilen tahmini bilgi bitlerini kullanarak tahmin edilen bilgiye dönüştürme işlemidir.

Haberleşme sisteminin başarımı, gönderilen bilgi ve tahmin edilen bilgi arasındaki farkla ölçülebilir.



Şekil 2.3 : Sayısal haberleşme sistemi temel elemanları.

2.3 Alan-Programlanabilir Kapı Dizileri

Sayısal elektronik devre tasarımında programlanabilir devrelerin önemli bir yeri vardır. Programlanabilir devreler programlanabilir bileşimli mantık devreleri ve programlanabilir sıralı mantık devreleri olarak 2 ana grupta incelenebilir.

Programlanabilir mantık dizisi (programmable logic array, PLA), programlanabilir sabit VE kapı dizisi ve programlanabilir VEYA kapı dizisinden oluşan bir programlanabilir bileşimli mantık devresidir. Herhangi bir mantıksal fonksiyon çarpımların toplamı olarak yazılabilir. Sabit VE kapıları ve programlanabilir VEYA kapıları kullanılarak PLA devreleri ile çarpımların toplamı işlemi gerçekleştirilebilir [17].

Programlanabilir diziler (programmable array logic, PAL), programlanabilir VE kapı dizisi ve sabit VEYA kapı dizisinden oluşan bir programlanabilir bileşimli mantık devresidir. PAL devreleri PLA devreleri kadar esnek değildir ancak PLA devrelerine göre daha kolay programlanabilirler.

Sıralı mantık devreleri temel olarak flip-floplar ve kapılardan oluşan programlanabilir mantık devreleridir. Programlanabilir mantık aracı (programmable logic device, PLD), karmaşık programlanabilir mantık aracı (complex programmable logic device, CPLD) ve alan-programlanabilir kapı dizileri (field-programmable gate array, FPGA) olarak 3 ana grupta incelenebilir.

PLD veya basit programlanabilir mantık aracı (simple programmable logic device,

SPLD) olarak da bilenen programlanabilir sıralı mantık devreleri, temel olarak PLA veya PAL bileşimli mantık devrelerine flip-flop dizilerinin eklenmesi ile oluşturulur. Kullanılan flip-floplar D veya JK tipi olabilir. Devrenin çıkış bileşimli devre çıkışından veya flip-flop çıkışından alınabilir.

CPLD, üreticilerine göre farklı olarak tasarlanan ve birden fazla PLD devresi, giriş/çıkış bloğu ve programlanabilir arabağlantı bloğundan oluşan programlanabilir sıralı mantık devreleridir [18].

FPGA, mantık devresi dizileri, giriş/çıkış bloğu dizileri, hafıza birimi dizileri ve bunlar arasındaki etkileşimi kontrol eden anahtarlama devresi dizilerinden oluşan ve son kullanıcı tarafından programlanabilen genel amaçlı programlanabilir mantık devreleridir [19].

2.3.1 FPGA iç yapısı

Alan-programlanabilir kapı dizilerinin temel elemanları olan programlanabilir mantık devreleri, programlanabilir anahtarlama devreleri (switching circuits, SC), giriş/çıkış blokları (input/output blocks, IOB) ve sabit birimler Şekil 2.4'te verilmiştir.

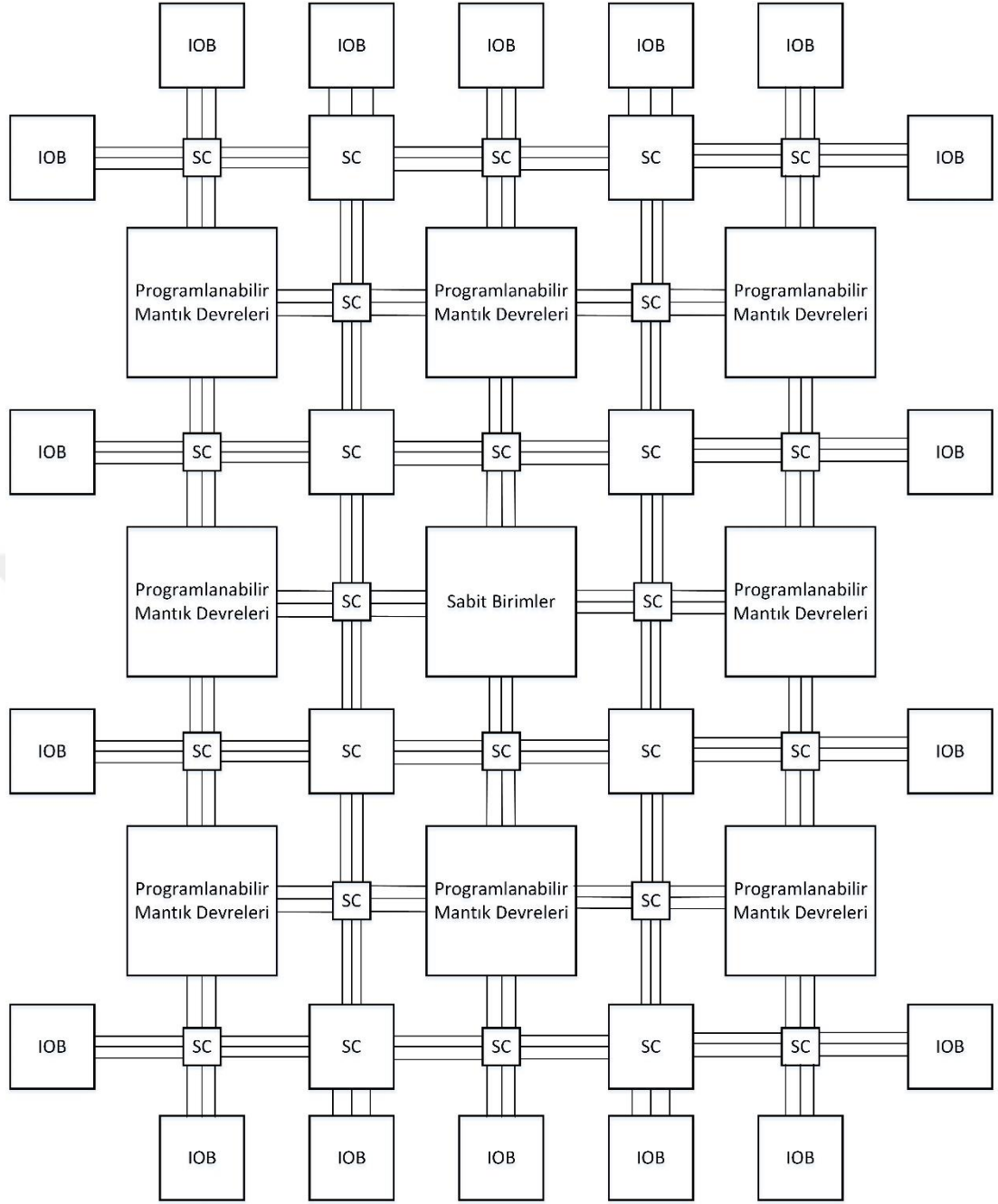
Genel olarak FPGA iç yapısı programlanabilir mantık devreleri ve programlanabilir anahtarlama devrelerinden oluşur. Programlanabilir mantık devreleri içerisinde taramalı tablolar, flip-floplar, bazı kapılar ve çoğullayıcılar gibi temel sayısal tasarım blokları bulunur. SC birimleri aracılığıyla istenilen fonksiyona göre uygun anahtarlama yapılarak farklı bağlantılar sağlanır. IOB birimleri FPGA ile dış arayüzleri sağlar. Programlanabilir anahtarlama devreleri aracılığıyla dışarıdan giriş olarak alınan sinyaller programlanabilir mantık devrelerine aktarılırken, istenilen sinyaller çıkış olarak IOB birimleri üzerinden dışarı çıkılabilir.

FPGA iç yapısında Şekil 2.4'te gösterilen birimlere ek olarak saat sinyali dağıtımı için özel hatlar bulunmaktadır. Bu hatlar FPGA içerisindeki tüm birimlere eş zamanlı ve düzgün saat sinyali ulaşmasını sağlamak üzere özel olarak tasarlanmış hatlardır.

2.3.2 FPGA çeşitleri

Üretim teknolojilerine göre FPGA'lar temel olarak, flash-tabanlı, ters sigorta ve SRAM-tabanlı olarak 3 çeşitle gruplandırılabilir.

Flash-tabanlı FPGA'larda program hafızası kayan kapılı transistörler aracılığıyla



Şekil 2.4 : FPGA iç yapısı.

tutulur. Güç kesilmesi durumunda tekrar programlanma gerektirmeyen bu FPGA türünde, güç tüketimi düşüktür. Tasarlanan elektronik kartlar üzerinde programlanabilen bu FPGA'lar geliştirme çalışmalarında esneklik sağlamaktadır. Programı silinip tekrar yüklenilebilir olsa da silinip yazılma sayısı transistörlerin yıpranması sebebiyle sınırlıdır [20].

Ters sigorta teknolojisi ile üretilen FPGA'lar sadece bir kez programlanabilen FPGA'lardır. Ters sigorta yapısı ile başlangıçta iletim yapılmazken, yakma olarak

adlandırılan programlama işleminden sonra iletim yapılır. Bu yakma işlemi geri dönüşü olmayan bir yakma işlemidir. Kullanım öncesi programlanmak zorunda olan bu FPGA'lar geliştirme çalışmalarında esneklik açısından dezavantajlıdır. Programlama öncesi ve sonrası sigorta yapısı Şekil 2.5'te verilmiştir [21].

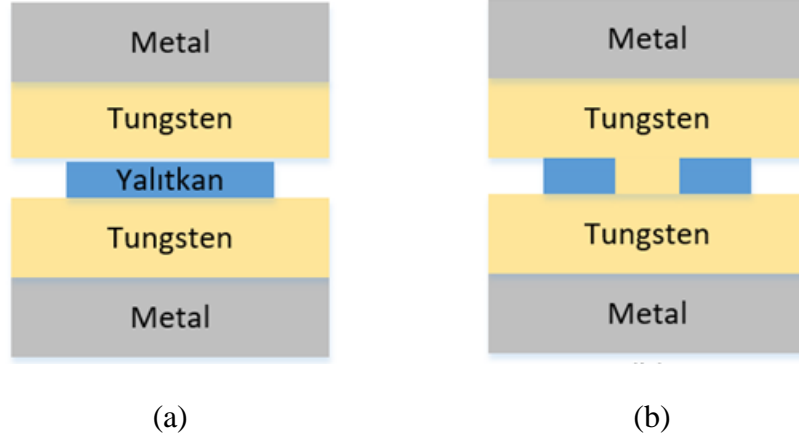
Ters sigorta teknolojisi ile üretilen FPGA'ların güç tüketimi düşüktür. Ayrıca radyasyon ve elektromanyetik dalgalara karşı dayanıklılığın yüksek olması sebebiyle uzay uygulamaları gibi yüksek güvenlik gerektiren uygulamalarda yaygın olarak kullanılmaktadır.

SRAM-tabanlı FPGA türlerinde, program durağan bir hafıza olan SRAM'da tutulur. Bu sebeple FPGA gücü kesildiğinde durağan hafızada tutulan program silinir. Güç verilmesi durumunda SRAM-tabanlı FPGA'lar uçucu olmayan bir hafıza veya başka işlemci vb. başka bir devre veya eleman aracılığıyla yeniden programlanmak zorundadır. Bu sebeple genellikle uçucu olmayan bir hafıza birimiyle birlikte kullanılırlar. SRAM-tabanlı FPGA'lar kullanım sırasında kart üzerinde programlanabildiği için geliştirme çalışmaları için uygundur. Tekrar programlanmasında herhangi bir yıpranma yoktur. FPGA içerisindeki programın uçucu olması sebebiyle uzay uygulamaları için fazladan güvenlik önlemi gerektirse de kısa süreli uygulamalar için kullanılabilir. FPGA içerisinde programın tutulduğu hafıza birimi durağan olduğu için güç tüketimi diğer FPGA türlerine göre daha yüksektir. Hız performansı olarak en yüksek performansı SRAM-tabanlı FPGA'lar sunmaktadır.

2.3.3 FPGA ile tasarım

FPGA üzerinde en yaygın tasarım yöntemi yazmaçlar ara veri aktarılması seviyesi (register transfer level, RTL) tasarımıdır. Sayısal tasarım blokları yazmaç seti ve işlem yapılan verilerin yazmaçlarda tutulduğu sistem olarak tanımlanır. Yazmaçlar sayısal tasarımın temel elemanlarıdır. Yazmaçlar üzerinde ikili sayılar tutulur ve yazmaçlar üzerinde tutulan verilerle 2-ölçekli işlemler yapılır. Yazmaçlar ile kaydırma, yer değiştirme, yazmaçtaki veriyi yeniden yazma ve yazmaçtaki veriyi okuma işlemleri yapılabilir.

Sayısal tasarımda kontrol işlemleri sonlu durum makineleri kullanılarak yapılır. Sonlu durum makinelerindeki durumlar, durum geçişleri ve durumlara göre çıkışlar yazmaçlar kullanılarak gerçekleştirilir.

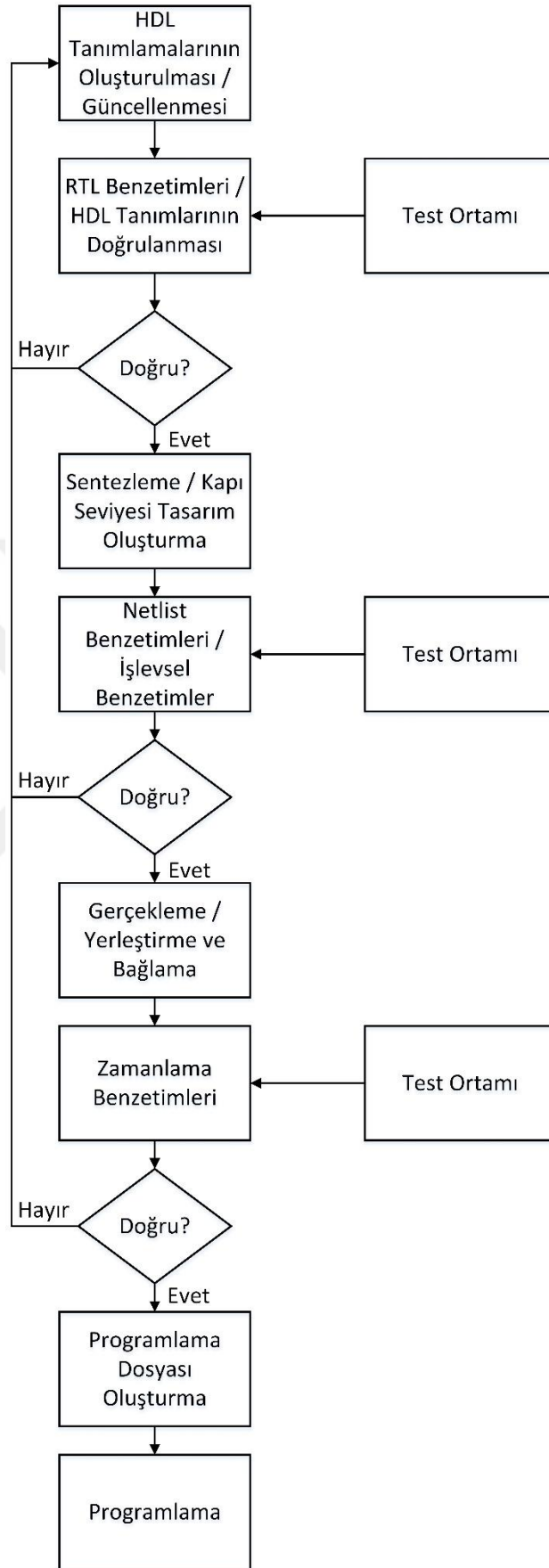


Şekil 2.5 : FPGA ters sigorta yapısı (a) programlamadan önce (b) programlamadan sonra.

FPGA'lar üzerinde RTL tasarımı donanım tanımlama dilleri (hardware description language, HDL) aracılığıyla yapılabilir. RTL seviyesindeki tüm tanımlamalar ve işlemler HDL kullanılarak yapılabilir.

RTL tasarımı yapılan sistemin mantıksal olarak sentezlenmesi ve kapı seviyesi tasarıma taşınması bilgisayar tabanlı programlar aracılığıyla gerçekleşir. Kapı seviyesi sentezleme işlemi kullanılan FPGA üreticisinin sunduğu yöntemlerle uygun bilgisayar programları aracılığıyla yapılır. Kapı seviyesinde yapılan tasarım daha sonra FPGA üzerine yerleştirilir ve yerleşimden sonra bağlantılar yapılır. Böylece FPGA üzerinde gerçekleştirme işlemi tamamlanır. Gerçekleşmesi tamamlanan tasarım için FPGA'ya yüklenebilecek formatta bit dosyası oluşturulur. Bu bit dosyası FPGA'ya yüklenmek amacıyla uçucu olmayan bir hafızaya yüklenir veya bilgisayar aracılığıyla direk olarak FPGA'ya yüklenir.

FPGA tasarımındaki tasarım aşamaları Şekil 2.6'da verilmiştir.



Şekil 2.6 : FPGA tasarım akışı.

3. UYARLAMALI KODLAMA VE KİPLEME KULLANAN VERİCİ YAPISI

Bu bölümde, deęişken hat koşullarına karşı kesintisiz ve verimli haberleşme için geliştirilen uyarlamalı kodlama ve kipleme kullanan verici yapısı açıklanmaktadır. Bu verici yapısında kodlama ve kipleme uyarlamaları uyarlamalı kodlama ve kipleme (adaptive coding and modulation, ACM) formatı ile belirlenmektedir. İlgili bloklar ACM formatını giriş olarak almakta ve gerekli uyarlamaları yaparak vericinin istenilen kod oranı ve kipleme şeması ile çalışması sağlamaktadır. Geliştirilen uyarlamalı kodlama ve kipleme kullanan verici yapısının blok diyagramı Şekil 3.1’de verilmiştir.

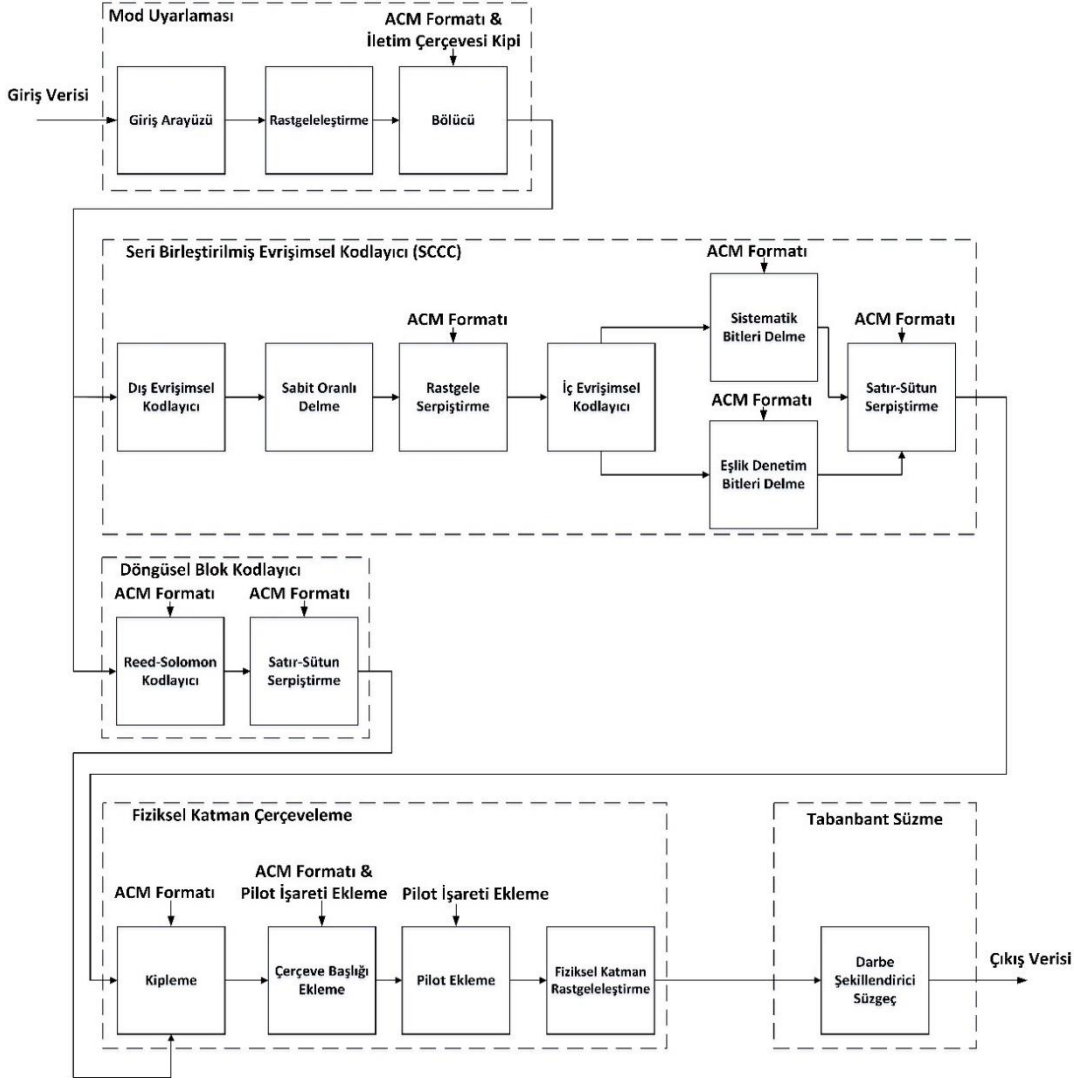
3.1 Mod Uyarlama

3.1.1 Rastgeleleştirme

Haberleşme sistemlerinde verici giriş verisi olarak uzun 1 serisine veya uzun 0 serisine rastlanma ihtimali vardır. Bu seriler verici çıkışında iletilen güç seviyesinin aynı kalmasına sebep olur. Seviye deęişiminin olmaması alıcı tarafında zaman kestirimi ve frekans kestirimi gibi eşzamanlama algoritmalarının çalışmamasına ve hatalı bir durumda kalmasına sebep olabilir. Uzun süreli 1 veya uzun süreli 0 serilerini önlemek için kanal kodlamaları öncesinde rastgeleleştirme işlemi yapılır [22].

Rastgeleleştirme işlemi için en yaygın kullanılan yöntemlerden biri toplanır rastgeleleştiricilerdir. Bu rastgeleleştiricilerde giriş verisi sözde-rastgele bir seri ile toplanır ve rastgele seri elde edilir. Sözde-rastgele seriler herhangi bir hafızada depolanıp kullanılabilirdiği gibi, doğrusal geribeslemeli kayan yazmaç (linear feedback shift register, LFSR) kullanılarak üretilebilmektedir. Doğrusal geribeslemeli kayan yazmaçlar, bir yazmaç dizisinin durumunun doğrusal bir kombinasyonunun geribesleme olarak verilerek kaydırıldığı yapılardır.

Bu çalışmada rastgeleleştirici olarak doğrusal geribeslemeli kayan yazmaç kullanılan bir toplanır rastgeleleştirici kullanılmıştır. Kullanılan doğrusal geribeslemeli kayan



Şekil 3.1 : Uyarlamalı kodlama ve kipleme kullanan verici yapısı.

yazmacın fonksiyonu (3.1)'de gösterilmiştir.

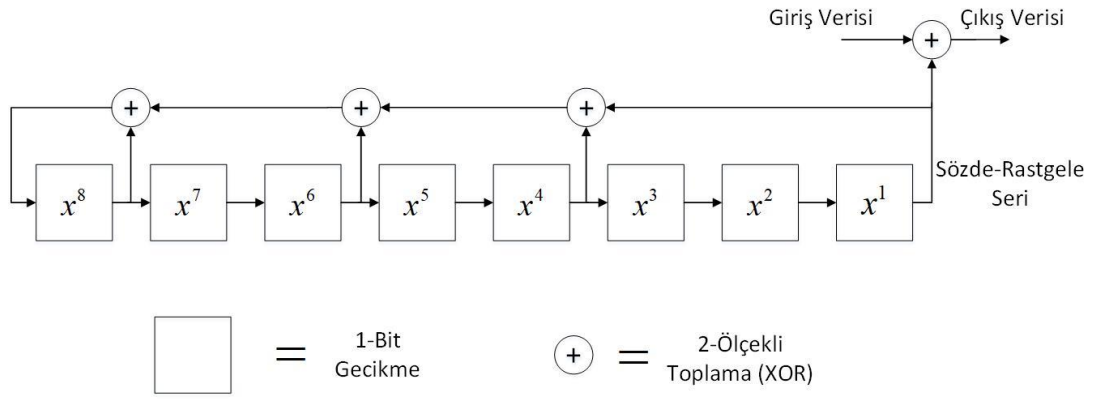
$$f(x) = x^8 + x^7 + x^5 + x^3 + 1 \quad (3.1)$$

Denklemi verilen doğrusal geribeslemeli kayan yazmaç kullanılarak oluşturulan rastgeleştirici yapısı Şekil 3.2'de verilmiştir.

Rastgeleleştiricide kullanılan yazmaçlara başlangıç değeri olarak 1 tanımlanır. Bir iletim çerçevesi verisi bittiğinde bu yazmaçlara tekrar 1 değeri tanımlanır ve böylece sözde-rastgele seri her iletim çerçevesi verisi için baştan başlamış olur.

3.1.2 Bölücü

İletim çerçeveleri eşzamanlaması için sabit uzunluklu eşzamanlama işaretlerine ihtiyaç duyulmaktadır. Bölücü devresi her iletim çerçevesi arasına sabit uzunluklu



Şekil 3.2 : Rastgeleleştirici yapısı.

eşzamanlama işareti eklemektedir. Eşzamanlama işareti olarak 32-bit uzunluğunda “00011010110011111111110000011101” bit dizisi tanımlanmıştır. Bölücü devresi bir iletim çerçevesini ilettikten sonra eşzamanlama işaretini iletip daha sonra diğer iletim çerçevesini ileterek devam etmektedir. Bölücü devresi yapısı Şekil 3.3’te verilmiştir.

Bölücü devresi iletim çerçevesi uzunluğu boyunca anahtar 1 konumunda tutar ve çıkış verisi olarak iletim çerçevesini iletir. İletim çerçevesi iletimi tamamlandıktan sonra anahtar 2 konumuna getirilir ve eşzamanlama işareti iletimi başlar.

Eşzamanlama işareti iletimi tamamlandıktan sonra anahtar tekrar iletim çerçevesi iletimi için 1 konumuna getirilir ve bu şekilde devam eder.

3.2 Seri Birleştirilmiş Evrişimsel Kodlayıcı

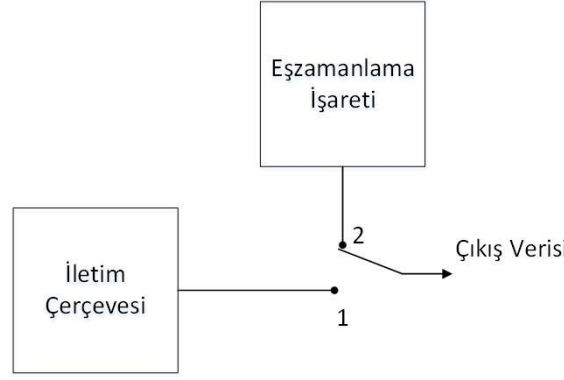
3.2.1 Sistemik evrişimsel kodlayıcı

Evrişimsel kodlayıcılar genel bilgi bitlerini sürekli olarak alır ve sürekli bir çıkış üretir. Her bilgi bitine karşılık üretilen kod sözcüğü biti hem o anki bilgi bitlerine hem de z önceki bilgi bitlerine bağlıdır. Bu yüzden bu kodlayıcılar z derece hafızaya sahiptir. Kodlayıcılar k-bitlik bilgi biti dizilerine karşılık n-bitlik kod sözcüğü dizileri üretmektedir. Bu kodlayıcılar için kod oranı denklemi (3.2)’de verilmiştir.

$$R = k/n \quad (3.2)$$

Sistemik evrişimsel kodlayıcılarda, n-bitlik kod sözcüğü dizilerininin k-biti, giriş k-bitlik bilgi bitleri ile aynıdır. Geriye kalan n-k bit ise eşlik denetim bitleridir [23].

Bu çalışmada 2 hafıza dereceli, $1/2$ kod oranlı sistemik evrişimsel kodlayıcı



Şekil 3.3 : Bölücü yapısı.

kullanılmıştır. Kullanılan kodlayıcının yapısı Şekil 3.4'te verilmiştir.

Kodlayıcıdaki hafıza birimlerinin başlangıç değeri 0 olarak tanımlanır. Giriş veri uzunluğu olan k -bit periyodu boyunca anahtar 1 konumunda olur ve c_1 çıkışından sistematik bitler, c_2 çıkışından eşlik denetim bitleri çıktısını verir. Giriş veri uzunluğu k -bit periyodu tamamlandıktan sonra anahtar 2 konumuna alınır ve kafes sonlandırması başlar. 2-bit periyodu uzunluğunda sonlandırma yapılır. Sonlandırma tamamlandığında hafıza birimlerinde 0 değeri görülür.

Sonlandırma ile kodlayıcı çıkışında k -bit giriş uzunluğuna karşılık $2(k+2)$ -bit kod sözcüğü elde edilir. Kafes sonlandırması için eklenen ekstra 2-bit spektral verimliliği düşürse de büyük veri uzunlukları için ihmal edilebilecek seviyededir.

Şekil 3.3'te gösterilen sistematik evrişimsel kodlayıcı yapısında x_1^k ve x_2^k kodlayıcının mevcut durumunu gösterirken, x_1^{k+1} ve x_2^{k+1} kodlayıcının sonraki durumunu gösterirse, u_k giriş biti için durum geçişleri ve c_1^k ve c_2^k çıkışlarının fonksiyonu anahtar 1 konumundayken (3.3), (3.4), (3.5) ve (3.6)'da verilmiştir.

$$x_1^{k+1} = u_k \oplus x_1^k \oplus x_2^k \quad (3.3)$$

$$x_2^{k+1} = x_1^k \quad (3.4)$$

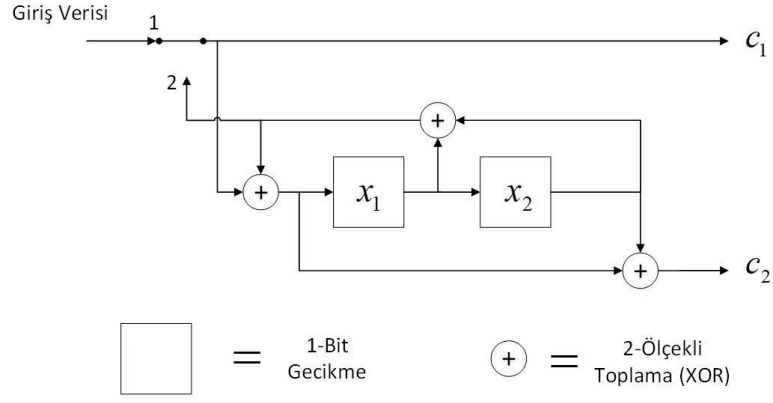
$$c_1^k = u_k \quad (3.5)$$

$$c_2^k = x_1^{k+1} \oplus x_2^k \quad (3.6)$$

Anahtarın 2 konumunda olduğu kafes sonlandırması durumunda kodlayıcının durum geçişleri ve çıkışlarının fonksiyonu (3.7), (3.8), (3.9) ve (3.10)'da verilmiştir.

$$x_1^{k+1} = 0 \quad (3.7)$$

$$x_2^{k+1} = x_1^k \quad (3.8)$$



Şekil 3.4 : Sistemik evrişimsel kodlayıcı yapısı.

$$c_1^k = x_1^k \oplus x_2^k \quad (3.9)$$

$$c_2^k = x_2^k \quad (3.10)$$

3.2.1.1 Vektör gösterimi

Sistemik evrişimsel kodlayıcılar için üreteç vektörü (3.11)'de gösterilmiştir.

$$G_R(D) = \begin{bmatrix} 1 & \frac{g_2(D)}{g_1(D)} \end{bmatrix} \quad (3.11)$$

Bu eşitlikte gösterilen $g_1(D)$ ve $g_2(D)$ polinomları sistemik evrişimsel kodlayıcının geribesleme ve ileribesleme bağlantılarını göstermektedir. Kullanılan kodlayıcı yapısı için $g_1(D)$ ve $g_2(D)$ polinomları ve bu polinomlara karşılık gelen vektör gösterimi sırası ile (3.12), (3.13), (3.14) ve (3.15)'te verilmiştir.

$$g_1(D) = D^2 + D + 1 \quad (3.12)$$

$$g_2(D) = D^2 + 1 \quad (3.13)$$

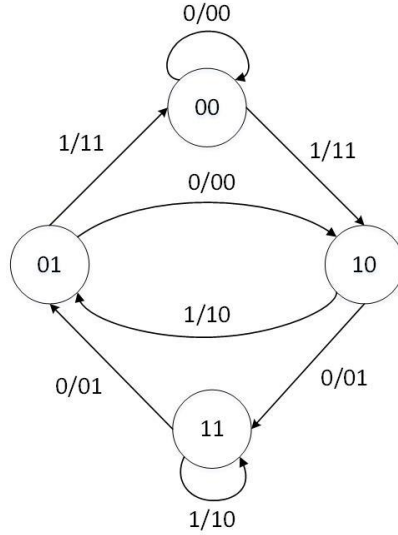
$$g_1 = [1 \ 1 \ 1] \quad (3.14)$$

$$g_2 = [1 \ 0 \ 1] \quad (3.15)$$

3.2.1.2 Durum diyagramı gösterimi

Evrişimsel kodlayıcılar için diğer bir gösterim durum diyagramı gösterimidir. Durum sayısı, durumlar arasındaki geçişler ve çıktılar her evrişimsel kodlayıcı yapısı için farklıdır. Durum sayısı kodlayıcının hafıza birimi sayısı z kullanılarak 2^z olarak bulunur. Durum geçişi koşulları ve çıktıları durum diyagramında gösterilir.

Bu çalışmada kullanılan sistemik evrişimsel kodlayıcı durum diyagramı Şekil 3.5'te verilmiştir.



Şekil 3.5 : Sistemik evrişimsel durum diyagramı.

Durum diyagramı verilen sistemik evrişimsel kodlayıcı için durum çizelgesi Çizelge 3.1’de verilmiştir.

3.2.1.3 Kafes yapısı

Bu çalışmada kullanılan sistemik evrişimsel kodlayıcı için kafes yapısı Şekil 3.6’da verilmiştir. Kafes diyagramı “00” durumundan başlayıp kafes sonlandırması sebebiyle yine “00” durumunda sonlanmıştır.

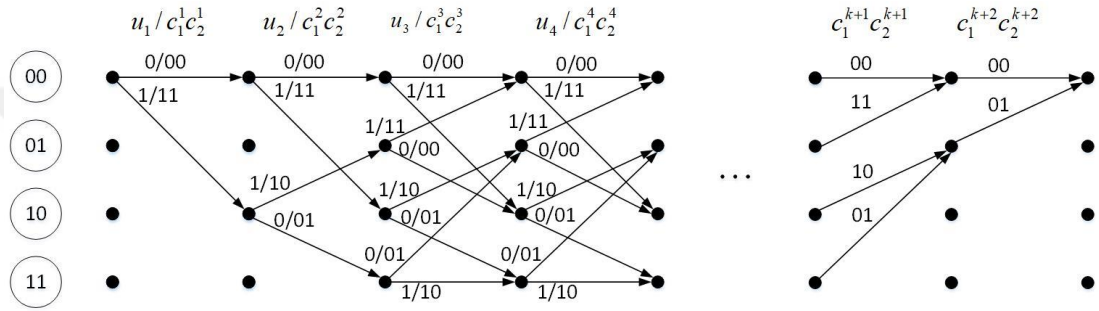
3.2.2 Delme

Bazı haberleşme sistemlerinde yüksek oranlı kodlayıcı gereksinimi duyulmaktadır. Evrişimsel kodlarda $n - 1/n$ gibi yüksek oranlı kodların elde edilmesi için kurulan yapıların kafes diyagramlarında her durum için 2^{n-1} farklı dal ortaya çıkmaktadır. Bu yapıların kodçözücülerinde her bit için 2^{n-1} işlem gerekmektedir. Bu işlem yoğunluğu pratikte uygulaması çok zordur. Bu sebeple yüksek kod oranlı kodlayıcılar düşük kod oranlı kodlayıcılarla birlikte delme işlemi uygulanması ile elde edilmektedir.

Delme işlemi evrişimsel kodlayıcı çıkışındaki bitlerin bazılarının iletilmemesi işlemidir. Bu işlem için delme matrisi tanımlanır, bu matrise göre matris elemanından 1’e karşılık gelen bitler iletilirken 0’a karşılık gelen bitler iletilmemektedir. Böylece düşük oranlı kodlayıcılar kullanılarak daha yüksek kod oranlı elde edilebilmektedir.

Çizelge 3.1 : Sistemik evrişimsel kodlayıcı durum çizelgesi.

Giriş	Mevcut Durum	Sıradaki Durum	Çıkış
u_k	$x_1^k x_2^k$	$x_1^{k+1} x_2^{k+1}$	$c_1^k c_2^k$
0	00	00	00
1	00	10	11
0	01	10	00
1	01	00	11
0	10	11	01
1	10	01	10
0	11	01	01
1	11	11	10



Şekil 3.6 : Sistemik evrişimsel kodlayıcı kafes yapısı.

Bu çalışmada farklı delme işlemleri uygulanmıştır. Sabit oranlı delme işlemi için kullanılan delme matrisi (3.16)'da verilmiştir.

$$P = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \quad (3.16)$$

Bu delme işlemi sonucunda c_1 ve c_2 girişleri alınarak örnekleme zamanına göre elde edilen çıkış verileri (3.17)'de verilmiştir.

$$c_1^1 c_2^1 c_1^2 c_2^2 c_1^3 c_2^3 c_1^4 c_2^4 \dots \quad (3.17)$$

Farklı oranlı delme işlemleri için farklı delme matrisi veya delme fonksiyonları tanımlanarak farklı bitler silinerek farklı kod oranları elde edilebilmektedir.

3.2.3 Serpiştirme

Kablosuz haberleşme sistemlerinde verilerin kanaldan iletimi sırasında hata patlamaları oluşması durumunda uzun süreli veri kaybı görülebilmektedir. Hata patlamalarının etkisini azaltmak için kod sözcüklerine serpiştirme işlemi uygulanır [24]. Bu serpiştirme işlemi ile kod sözcüklerindeki bitler rastgele veya belli bir

modele göre dağıtılır. Kullanılan yapılara göre serpiştiriciler farklı gruplara ayrılabilir. Bu çalışmada rastgele serpiştirici ve satır-sütun serpiştirici kullanılmıştır.

Rastgele serpiştiriciler seri olarak aldığı giriş verilerinin belirli bir modele göre permütasyonunu alarak çıktı vermektedir. Bu çalışmada kullanılan rastgele serpiştirici bloğunda I -bit uzunluğundaki giriş verisini doğal sırası ile bir hafıza bloğuna yazılır. Hafıza bloğundan i zamanında okunacak verinin adresi $\pi(i)$ (3.18)'de verilmiştir.

$$\pi(i) = W \times \left[\left(\lfloor i/W \rfloor + \beta(i_W) \right) \bmod 120 \right] + \alpha(i_W) \quad i = 0, 1, \dots, I - 1 \quad (3.18)$$

$$W = I/120 \quad (3.19)$$

$$i_W = i \bmod W \quad (3.20)$$

(3.18)'de kullanılan β ve α önceden tanımlı W uzunluğundaki vektörler, $[x]$ işlemi x 'ten küçük veya eşit en büyük tamsayı değerini ifade etmektedir. Bu serpiştirici yapısının görsel olarak ifadesi Şekil 3.7'de verilmiştir.

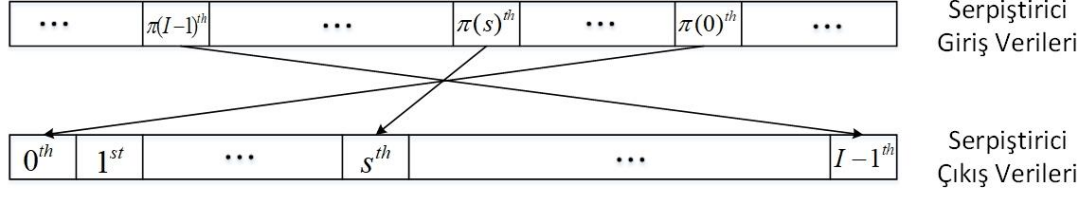
Satır-sütun serpiştiriciler blok serpiştirici olarak da adlandırılırlar. Bu serpiştirme işlemi bir sembole atanan bitlerin alıcıda ters etkisi yaratmaması için yapılır. Bu serpiştiricilerde seri olarak alınan $k_s = m_s \cdot n_s$ uzunluğundaki giriş verileri $n_s \times m_s$ boyutundaki bir hafıza bloğuna ilk sütunda yukarıdan aşağı doğru başlayıp, sütunlarda soldan sağa doğru ilerleyerek yazılır. Yazma işleminde k_s bit tamamlandıktan sonra veriler ilk satırda soldan sağa doğru başlayıp, satırlarda yukarıdan aşağı doğru ilerleyerek okunur.

Bu çalışmada kullanılan satır-sütun serpiştirici bloğunun büyüklüğü yapılacak kipleme göre değişken olarak tanımlanmıştır. Kipleme sonrası her kodbloğu 8100 sembole karşılık gelecek şekilde sabitlenmiştir. Böylece satır-sütun serpiştirici bloğu hafızasının büyüklüğü kipleme derecesi m ve 8100'e bağlıdır. Satır-sütun serpiştirici bloğu hafıza birimi $8100 \times m$ olarak belirlenmiştir. Satır-sütun serpiştiricinin görsel olarak ifadesi Şekil 3.8'de verilmiştir.

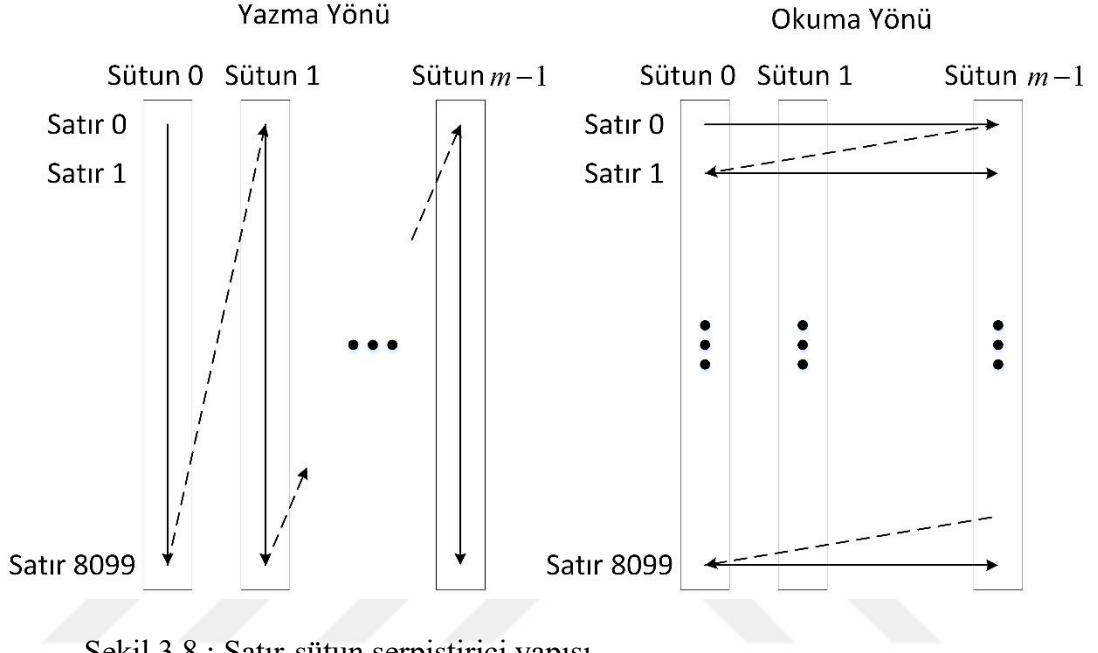
3.3 Döngüsel blok kodlayıcı

3.3.1 Galois cismi

Bir cisim kabaca elemanları ile toplama, çıkarma ve çarpma işlemleri yapıldığında



Şekil 3.7 : Rastgele serpiştirici yapısı.



Şekil 3.8 : Satır-sütun serpiştirici yapısı.

sonucun setin içinde kaldığı bir set olarak tanımlanabilir. Bu set içerisinde tanımlı toplama ve çarpma işlemleri değişme özelliğini, birleşme özelliğini ve dağılma özelliğini sağlamak zorundadır.

Bir cisimdeki eleman sayısı o cismin derecesini belirler. Sonlu eleman sayısına sahip olan cisimler sonlu cisimler olarak tanımlanır. Eleman sayısı p olan sonlu cisim veya Galois cismi $GF(p)$ kısaltması ile gösterilir. Elemanları $\{0,1\}$ olan iki elemanlı Galois cismi ikili Galois cismi olarak isimlendirilir ve $GF(2)$ kısaltması ile gösterilir. İkili Galois cismi kodlama kuramında çok temel ve önemli bir rol oynamaktadır. İkili Galois cisminde yapılan toplama ve çarpma işlemleri sırasıyla Çizelge 3.2 ve Çizelge 3.3'te verilmiştir.

Her asal p sayısı için p elemanlı bir sonlu cisim $GF(p)$ tanımlanabilir. Her a sayısı için $GF(p)$ cismi p^a elemana genişletilebilir. Elde edilen p^a elemanlı cisimler $GF(p)$ cisminin genişletilmiş cismi olarak isimlendirilir ve $GF(p^a)$ ile gösterilir.

Çizelge 3.2 : İkili Galois cisminde toplama işlemi.

+	0	1
0	0	1
1	1	0

Çizelge 3.3 : İkili Galois cisminde çarpma işlemi.

×	0	1
0	0	0
1	0	1

İkili Galois cisimlerinde ve genişletilmiş ikili Galois cisimlerinde Çizelge 3.2 ve Çizelge 3.3'te verildiği gibi toplama ve çarpma işlemleri 2-ölçekli toplama ve 2-ölçekli çarpma işlemi olarak yapılır. İkili Galois cisimlerinde çıkarma işlemi toplama işlemi ile aynıdır.

Katsayıları ikili Galois cismi elemanlarından olan polinomlar ikili Galois cismi üzerindeki polinomlar olarak tanımlanabilir. İkili Galois cismi üzerinde olan ve tek değişkenli $d(X)$ polinomu (3.21)'de verildiği gibi yazılabilir.

$$d(X) = d_0 + d_1X + d_2X^2 + \dots + d_bX^b \quad d_i = \{0,1\} \quad 0 \leq i \leq b, \quad (3.21)$$

En yüksek dereceli elemanın katsayısı olan d_b değerinin 1 olması durumunda $d(X)$ polinomu b dereceli bir polinom olarak tanımlanır. İkili Galois cismi üzerinde b dereceli 2^b polinom bulunmaktadır.

Herhangi bir polinomun kökü gerçel sayılarda olduğu gibi polinomu 0 yapan değer olarak tanımlanır. Bir $e(X)$ polinomunun kökü e_r ise $e(e_r)$ değeri 0 olarak görülür ve $e(X)$ polinomu $X - e_r$ değerine Öklit'in bölme algoritmasına göre tam bölünür.

İkili Galois cismi üzerinde b dereceli herhangi polinom olan $d(X)$, ikili Galois cismi üzerinde 0'dan büyük ve b 'den küçük dereceli yine ikili Galois cismi üzerinde başka bir polinoma tam bölünemiyorsa, indirgenemez polinom olarak tanımlanır. 1'den büyük her b değeri için b dereceli bir indirgenemez polinom olduğu ispatlanmıştır. İkili Galois cismi üzerindeki b dereceli herhangi bir polinom, $X^{2^b-1} + 1$ polinomunu tam bölmektedir.

En küçük pozitif tamsayı $c = 2^b - 1$ değeri için $X^c + 1$ polinomunu tam bölebilen b dereceli indirgenemez polinom $p(X)$ ilkel polinom olarak tanımlanır. Herhangi bir a değeri için birden çok ilkel polinom tanımlanabilir [23].

İkili Galois cismi üzerinden genişletilmiş Galois cismi $GF(2^a)$, a dereceli ilkel polinom kullanarak toplama ve çarpma işlemleriyle birlikte tanımlanabilir.

3.3.2 Doğrusal blok kodlar

Doğrusal blok kodlar k_b -bit uzunluğundaki bilgi bitlerinden n_b -bit uzunluğundaki kod sözcüklerini oluşturur ve (n_b, k_b) blok kodlama olarak ifade edilir. Bu kodlayıcıda 2^{k_b} olası bilgi dizisi ve 2^{n_b} olası kod sözcüğü dizisi bulunmaktadır. Her bilgi dizisi tek ve farklı bir kod sözcüğü dizisine eşlenmektedir. Kodlayıcının kod oranı k_b/n_b olarak tanımlanabilir.

Doğrusal blok kodlar bilgi dizisinin kod sözcüğü serisine eşlenmesi işleminin doğrusal olduğu blok kodlar olarak tanımlanır [25].

Doğrusal blok kodlarda kod sözcüğü bilgi bitlerinin doğrusal kombinasyonu ile oluşturulur. Bir (n_b, k_b) doğrusal blok kodlama için bilgi biti dizisi (3.22)'de gösterildiği gibi tanımlanabilir.

$$\mathbf{m}_i = \{m_{i1}, m_{i2}, \dots, m_{ik}\} \quad (3.22)$$

(3.22)'de verilen bilgi biti dizisinin kodlandığı kod sözcüğü dizisi (3.23)'teki gibi tanımlanabilir.

$$\mathbf{c}_i = \{c_{i1}, c_{i2}, \dots, c_{in}\} \quad (3.23)$$

(3.22)'de verilen bilgi dizisinin (3.23)'te verilen kod sözcüğü dizisine kodlama işlemi (3.24)'teki gibi tanımlanabilir.

$$c_{ij} = m_{i1}g_{1j} + m_{i2}g_{2j} + \dots + m_{ik}g_{kj} \quad j = 1, 2, \dots, n \quad (3.24)$$

Bu doğrusal blok kodlama için üreteç matrisi (3.25)'teki gibi tanımlanır.

$$\mathbf{G} = \begin{bmatrix} g_{11} & \dots & g_{1n} \\ \vdots & \ddots & \vdots \\ g_{k1} & \dots & g_{kn} \end{bmatrix} \quad (3.25)$$

Üreteç matrisinin l .satırı $\mathbf{g}_l = [g_{l1} \ g_{l2} \ \dots \ g_{ln}]$ olarak tanımlanırsa, (3.24)'te verilen ifade herhangi bir kod sözcüğü dizisi için (3.26)'da verilen şekilde yeniden ifade edilebilir.

$$\mathbf{c}_i = m_{i1}\mathbf{g}_1 + m_{i2}\mathbf{g}_2 + \dots + m_{ik}\mathbf{g}_k \quad (3.26)$$

Herhangi bir bilgi dizisi \mathbf{m}_i için (n_b, k_b) blok kodlaması sonucunda elde edilen kod

sözcüğü dizisi \mathbf{c}_i , (3.27)'de verilen matris çarpımı şeklinde elde edilebilir.

$$\mathbf{c}_i = \mathbf{m}_i \cdot \mathbf{G} \quad (3.27)$$

3.3.2.1 Döngüsel blok kodlar

Döngüsel blok kodlar, doğrusal blok kodların bir alt sınıfıdır. Döngüsel blok kodlarda herhangi bir kod sözcüğünün döngüsel olarak kayması ile yeni bir kod sözcüğü elde edilir. Herhangi bir kod sözcüğü olan $\mathbf{c}_j = \{c_0 c_1 \dots c_{n-1}\}$ dizisinin 1-bit kaydırılması ile edilen $\mathbf{c}_j^1 = \{c_{n-1} c_0 c_1 \dots c_{n-2}\}$ dizi de bir kod sözcüğüdür. Genel olarak $\mathbf{c}_j^i = \{c_{n-i} c_{n-i+1} \dots c_{n-1}\}$ dizisi bir kod sözcüğüdür. Bu kodlamaların döngüsel yapısı, kodlayıcı ve kodçözücü açısından diğer doğrusal blok kodlamalara göre daha az karmaşık bir yapı oluşturmaktadır.

Döngüsel blok kodlamalarda kod sözcüğü oluşturulmasında üreteç matrisi yerine üreteç polinomu kullanılmaktadır. Döngüsel bir (n_b, k_b) blok kodlayıcı için üreteç matrisi $n_b - k_b$ derece bir polinom olarak ifade edilir.

Döngüsel bir blok kodlayıcı (n_b, k_b) için (3.28)'de verilen k_b -bit uzunluğundaki bilgi dizisi ve (3.29)'da verilen n_b -bit uzunluğundaki kod sözcüğü dizisi bilgi polinomu ve kod sözcüğü polinomu olarak sırasıyla (3.30) ve (3.31)'de verilmiştir.

$$\mathbf{m} = \{m_0 m_1 \dots m_{k-1}\} \quad (3.28)$$

$$\mathbf{c} = \{c_0 c_1 \dots c_{n-1}\} \quad (3.29)$$

$$m(X) = m_0 + m_1X + m_2X^2 + \dots + m_{k-1}X^{k-1} \quad (3.30)$$

$$c(X) = c_0 + c_1X + c_2X^2 + \dots + c_{n-1}X^{n-1} \quad (3.31)$$

Doğrusal blok kodlamalarda $c(X)$ kod sözcüğü polinomu $m(X)$ bilgi polinomunun üreteç polinomu adı verilen ve (3.32)'de tanımlanan $g(X)$ polinomu ile çarpılmasıyla (3.33)'de gösterildiği gibi oluşturulur.

$$g(X) = g_0 + g_1X + g_2X^2 + \dots + g_{n-k-1}X^{n-k-1} \quad (3.32)$$

$$c(X) = m(X) \cdot g(X) \quad (3.33)$$

Sistematik döngüsel blok kodlar

Sistematik döngüsel bloklarda kod sözcüğü olarak önce bilgi bitleri olduğu gibi iletilir, arkasından eşlik denetim bitleri iletilir. Sistematik döngüsel blok kodlar direk

olarak üreteç polinomu ile oluşturulabilirler.

Sistemik döngüsel bir (n_b, k_b) blok kodlayıcı kod sözcüğünü oluşturmak için öncelikle bilgi polinomu $m(X)$ ile X^{n-k} ile (3.34)'te gösterildiği gibi çarpılmaktadır.

$$X^{n-k} \cdot m(X) = m_{k-1}X^{n-1} + m_{k-2}X^{n-2} + \dots + m_1X^{n-k-1} + m_0X^{n-k} \quad (3.34)$$

Sistemik bir döngüsel kodlayıcıda (3.34)'te kod sözcüğünün bilgi bitleri üretilmiştir. Kod sözcüğünün tamamlanması için $n_b - k_b$ dereceli eşlik denetim bitlerini ifade eden bir polinomun eklenmesi gerekir. Bu polinomun elde edilmesi için $X^{n-k}m(X)$ polinomu üreteç polinomuna (3.35)'te verilen şekilde bölünür ve bu işlem (3.36)'da verilen şekilde gösterilebilir.

$$\frac{X^{n-k}m(X)}{g(X)} = Q(X) + \frac{r(X)}{g(X)} \quad (3.35)$$

$$X^{n-k}m(X) = g(X)Q(X) + r(X) \quad (3.36)$$

(3.35)'te $Q(X)$ polinomu bölüm, $r(X)$ polinomu kalan olarak ifade edilebilir. Kalan polinomu $n_b - k_b$ veya daha düşük dereceli bir polinomdur.

(3.36)'da $g(X)Q(X)$ bir döngüsel kod sözcüğüdür. (3.36)'da verilen denklemin iki tarafına 2-ölçekli toplama işlemi ile $r(X)$ polinomu eklenirse, istenilen sistemik kod (3.37)'de verilen şekilde elde edilebilir.

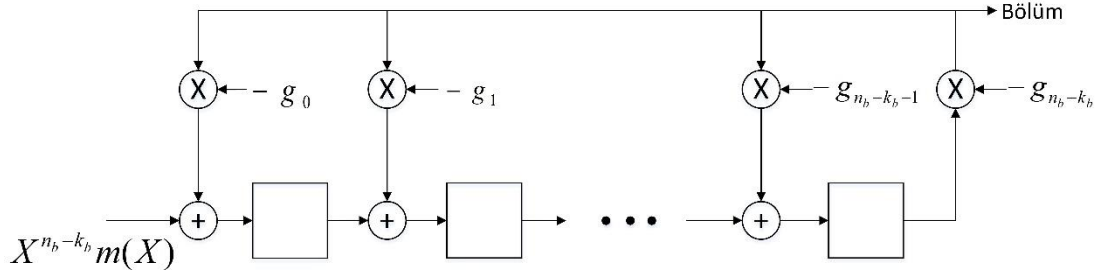
$$g(X)Q(X) = X^{n-k}m(X) + r(X) \quad (3.37)$$

Sistemik döngüsel blok kodlar için kodlayıcı yapısı

Sistemik döngüsel blok kodlayıcılarda kod sözcüğü oluşturulması (3.35)'te verilen polinom bölmesi işlemi yapılması ile başlar. Bu eşitlikteki $(n_b - 1)$ dereceli $X^{n_b-k_b}m(X)$ polinomunun (3.38)'de verilen $(n_b - k_b)$ dereceli $g(X)$ polinomuna bölme işlemi $(n_b - k_b)$ elemanlı bir kayan yazmaç kullanımı ile Şekil 3.9'da gösterilen yapı ile gerçekleştirilebilir [16].

$$g(X) = g_{n_b-k_b}X^{n_b-k_b} + g_{n_b-k_b-1}X^{n_b-k_b-1} + \dots + g_2X^2 + g_1X + g_0 \quad (3.38)$$

İkili sistemik döngüsel blok kodlarda üreteç polinomları $g(X)$ için g_0 ve $g_{n_b-k_b}$ katsayıları 1 olmak zorundadır. Yapılan çarpma ve bölme işlemleri ikili sistemik döngüsel kodlarda 2-ölçekli toplama ve 2-ölçekli çarpma işlemleridir. Kayan yazmaç kullanımı ile oluşturulan sistemik döngüsel blok kodlayıcı yapısı Şekil 3.10'da verilmiştir.



Şekil 3.9 : Polinom bölme işlemi.

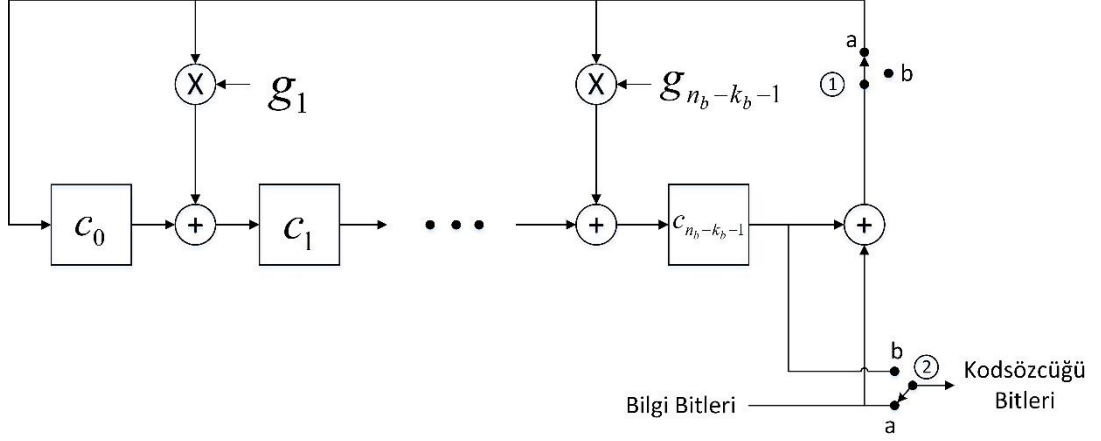
Polinom gösterimlerinde bulunan bilgi bitlerini $X^{n_b-k_b}$ polinomu ile çarpma işlemi fiziksel olarak yapılmamış, bu işlemi yazmaçlarla gelen zamanda bekleme işlemi sağlamıştır. Kodlayıcı çalışması sırasında öncelikle 1 ve 2. anahtarlar a konumuna getirilir. Bilgi bitleri kod sözcüğü bitleri olarak çıkarken, eşlik denetim bitlerini oluşturmak üzere çalışan kayan yazmaç yapısını da besler. Bilgi bit dizisinin tamamlanması durumunda 1 ve 2. anahtarlar b konumuna getirilir. Kayan yazmaç yapısı beslemesi durur ve yazmaçlardaki değerler eşlik denetim bitleri olarak kod sözcüğünün kalan kısmını oluşturur.

3.3.3 Reed-Solomon kodları

Döngüsel kodların güçlü bir sınıfı Bose-Chadhuri-Hocquenghem (BCH) kodlamalarıdır. Bu kodlamalar diğer blok kodlamalara göre aynı n ve k değerleri için daha yüksek performans sunmaktadır. BCH kodlamaları ikili kodlamalar olabileceği gibi ikili olmayan kodlamalar olarak da tanımlanabilir.

İkili olmayan kodlarda bilgi ve kod sözcüğü dizileri bitler yerine semboller ile ifade edilebilir. Kodlayıcıda yapılan işlemler için 2-ölçekli işlemler yine çalışılan Galois cisminde tanımlı olan işlemler kullanılır.

İkili olmayan kodlarda en yaygın kullanılan kodlama Reed-Solomon kodlarıdır. Bilgi dizisi uzunluğu k_{RS} sembol ve kod sözcüğü dizisi uzunluğu n_{RS} sembol olan Reed-Solomon kodları $RS(n_{RS}, k_{RS})$ sembolü ile gösterilebilir. Reed-Solomon kodlarında kod sözcüğü dizisi uzunluğu n_{RS} değeri $2^q - 1$ olarak seçilebilir. Bilgi ve kod sözcüğü sembolleri q -bit uzunluğunda olur. Bilgi dizisi uzunluğu k_{RS} değeri $1, 2, \dots, n_{RS} - 1$ olarak seçilebilir. Reed-Solomon kodlayıcı seçilen n_{RS} ve k_{RS} değerleri için $(n_{RS} - k_{RS})/2$ sembol hatasını düzeltebilecek kapasitededir. Reed-Solomon kodlayıcı için $GF(2^q)$ Galois cismi oluşturulur ve kodlama bu cisim üzerinde tanımlanır [25].



Şekil 3.10 : Sistematik döngüsel blok kodlayıcı yapısı.

Bu çalışmada kullanılacak döngüsel blok kodlayıcı seçiminde kod sözcüğü dizisi uzunluğu ve kod oranına kodlayıcı seçilmiştir. Toplam kod sözcüğü dizisi uzunluğunun sabit olması sebebiyle bu uzunlukta en yüksek kod oranını sağlayan $RS(127,120)$ kodlayıcı kullanılmıştır. Bu kodlayıcının tanımlı olduğu $GF(2^7)$ Galois cismi oluşturulmasında kullanılan ilkel polinom ve kodlayıcının kullandığı üreteç polinomu sırasıyla (3.39) ve (3.40)'da verilmiştir.

$$p(X) = X^7 + X^3 + 1 \quad (3.39)$$

$$g(X) = 51X^7 + 77X^6 + 80X^5 + 36X^4 + 32X^3 + 113X^2 + 119X + 1 \quad (3.40)$$

3.4 Fiziksel Katman Çerçeveleme

3.4.1 Kipleme

3.4.1.1 M-PSK

Sayısal haberleşme sistemlerinde kullanılan faz kaydırmalı anahtarlama kipleme yönteminde gönderilecek bilgi iletilen sinyalin fazı ile gönderilir. İletilen sinyalin genliği sabittir. M-PSK kipleme yönteminde iletilen sinyalin formülü (3.41)'de verilmiştir [25].

$$s_i(t) = A \cos(2\pi f_c t + \varphi_i), \quad 0 \leq t \leq T, \quad i = 0, 1, \dots, M \quad (3.41)$$

$$\varphi_i = \frac{(2i-1)\pi}{M} \quad (3.42)$$

(3.41)'de A taşıyıcı sinyalin genliğini, f_c taşıyıcı sinyalin frekansını, T gönderilen işaretin periyodunu ve φ_i gönderilen işaretin fazını göstermektedir.

M-PSK ile gönderilen sinyaller iki boyutlu koordinat sisteminde yıldız diyagramı ile ifade edilebilmektedir. İki boyutlu koordinat sistemi için taban sinyalleri (3.43) ve (3.44)'te verilmiştir.

$$b_1(t) = \sqrt{\frac{2}{T}} \cos 2\pi f_c t, \quad 0 \leq t \leq T \quad (3.43)$$

$$b_2(t) = -\sqrt{\frac{2}{T}} \sin 2\pi f_c t, \quad 0 \leq t \leq T \quad (3.44)$$

(3.41)'de verilen formül (3.45)'te belirtilen şekilde yazılıp, iki boyutlu koordinat sistemi taban sinyalleri cinsinden (3.46)'daki gibi yazılabilir.

$$s_i(t) = A \cos \varphi_i \cos(2\pi f_c t) - A \sin \varphi_i \sin(2\pi f_c t) \quad (3.45)$$

$$s_i(t) = s_{i1} b_1(t) + s_{i2} b_2(t) \quad (3.46)$$

Yıldız diyagramında sinyaller $b_1(t)$ ve $b_2(t)$ ile oluşturulan koordinat sisteminde (s_{i1}, s_{i2}) şeklinde gösterilir. Bu değerlerin formülleri (3.47) ve (3.48)'de verilmiştir.

$$s_{i1} = \int_0^T s_i(t) b_1(t) dt = \sqrt{E} \cos \varphi_i \quad (3.47)$$

$$s_{i2} = \int_0^T s_i(t) b_2(t) dt = \sqrt{E} \sin \varphi_i \quad (3.48)$$

(3.47) ve (3.48)'de geçen E sinyalin enerjisini ifade eder ve (3.49)'da verilen denklem ile hesaplanır.

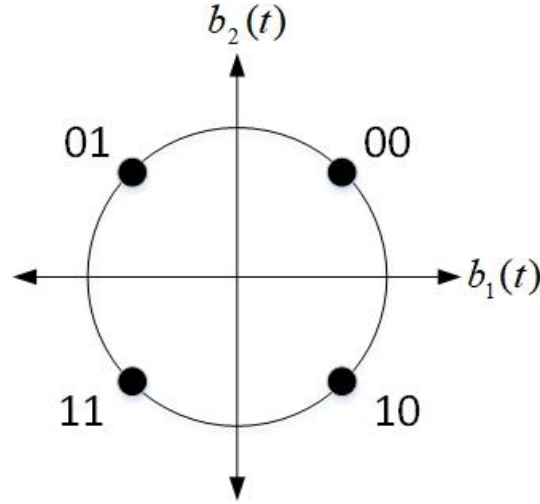
$$E = \frac{1}{2} A^2 T \quad (3.49)$$

Bu çalışma kapsamında yapılan kipleme işlemlerindeki sinyaller birim enerjili olarak tasarlanmıştır.

QPSK

QPSK kiplemesinde her sembolde 2 bit gönderilir. Bitler sembollere Gray kodlamalı olarak yerleştirilmiştir. Komşu iki sembol arasında 1-bit fark bulunmaktadır. Bu yerleşim ile bit hata oranı düşürülmüştür. QPSK kiplemesine ait yıldız diyagramı Şekil 3.11'de verilmiştir.

QPSK kiplemesinde M değeri 4 ve ortalama sinyal enerjisi E 1 olarak alınarak gönderilen işaretlerin fazı ve bu işaretler için (s_{i1}, s_{i2}) değerleri sırasıyla (3.50) ve (3.51)'de verilmiştir.



Şekil 3.11 : QPSK yıldız diyagramı.

$$\varphi_i = \frac{(2i-1)\pi}{4}, \quad i = 1,2,3,4 \quad (3.50)$$

$$s_{i1} = \cos \frac{(2i-1)\pi}{4}, \quad s_{i2} = \sin \frac{(2i-1)\pi}{4}, \quad i = 1,2,3,4 \quad (3.51)$$

8-PSK

8-PSK kiplemesinde her sembolde 3 bit gönderilir. Bitler sembollere Gray kodlamalı olarak yerleştirilmiştir. 8-PSK kiplemesine ait yıldız diyagramı Şekil 3.12'da verilmiştir.

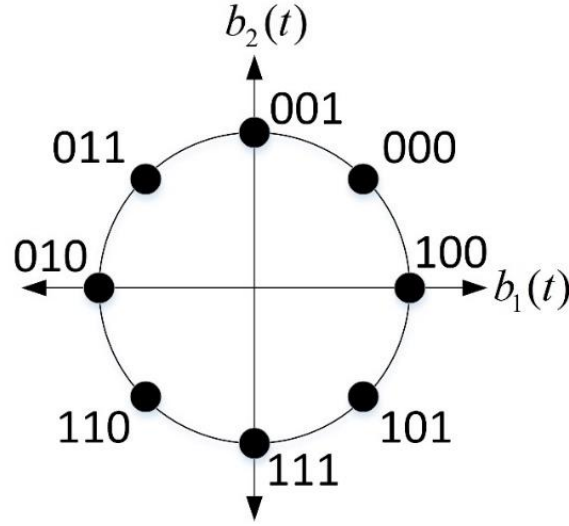
8-PSK kiplemesinde M değeri 8 ve ortalama sinyal enerjisi 1 olarak alınmıştır. Semboller yıldız diyagramına $\pi/8$ faz ofseti ile yerleştirilmiştir. Gönderilen işaretlerin fazı ve bu işaretler için (s_{i1}, s_{i2}) değerleri sırasıyla (3.52) ve (3.53)'de verilmiştir.

$$\varphi_i = \frac{(2i-1)\pi}{8} + \frac{\pi}{8}, \quad i = 1,2, \dots, 8 \quad (3.52)$$

$$s_{i1} = \cos \frac{(2i-1)\pi}{8}, \quad s_{i2} = \sin \frac{(2i-1)\pi}{8}, \quad i = 1,2, \dots, 8 \quad (3.53)$$

3.4.1.2 M-APSK

Haberleşme sistemlerinde kullanılan güç yükselteçleri doğrusal olmayan bir karaktere sahiptir. Özellikle uydu haberleşme sistemlerinde kullanılan ve yüksek güce ihtiyaç duyulan güç yükselteçlerinde bu doğrusal olmayan karakter alıcıda düzeltilemeyen hatalara sebep olur. Yüksek güç seviyelerinde giriş gücü doğrusal olarak yükseltilemez ve çıkış gücü belirli bir değere yakınsar. Güç yükselteçlerinin



Şekil 3.12 : 8-PSK yıldız diyagramı.

bu davranışı göz önüne alındığında yaygın olarak kullanılan dördün genlik kiplemesi, farklı enerji seviyeleri sayısının ve dolayısıyla tepe-ortalama güç oranının yüksek olması sebebiyle dezavantajlı olmaktadır. Güç yükselteçlerinde oluşan bu hatalara karşı genlik-faz kaydırmalı anahtarlama kiplemesi uygulanmaktadır.

Genlik-faz kaydırmalı anahtarlama kiplemesinde iletilen sinyaller farklı genlik ve farklı faz değerlerine sahiptir. Bu kipleme farklı genliklere sahip iç içe geçmiş faz kaydırmalı anahtarlama kiplemesi olarak düşünülebilir.

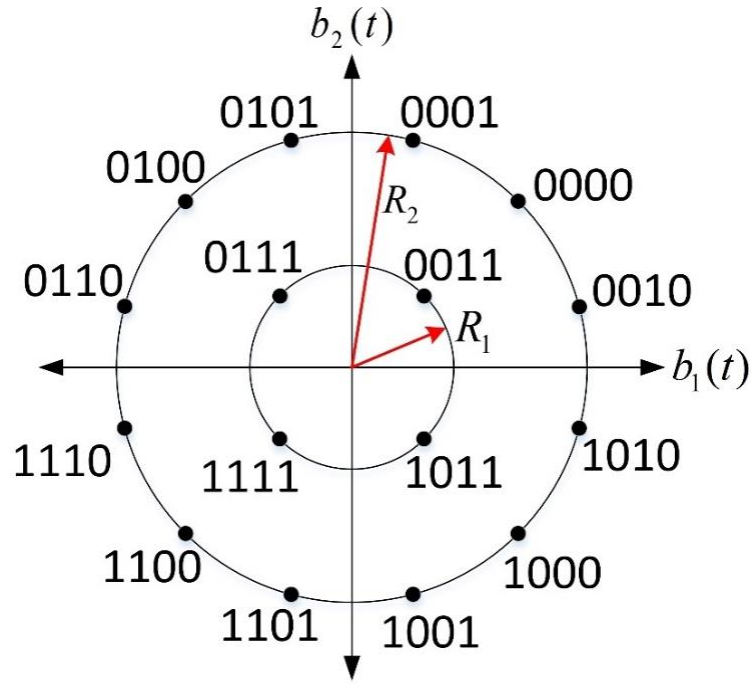
16-APSK

16-APSK kiplemesinde bir sembol ile 4 bit iletilmektedir. Bu kipleme için Gray kodlanmış yıldız diyagramı Şekil 3.13'te verilmiştir.

16-APSK yıldız diyagramı eş merkezli farklı yarıçaplı iki halkadan oluşmaktadır. Dış halka birbirlerine eşit uzaklıkta 12 sembol bulunurken, iç halkada birbirlerine eşit uzaklıkta 4 sembol bulunmaktadır. Halkaların yarıçapları oranı R_2/R_1 kipleme ile kullanılan kodlamanın kod oranına göre belirlenmektedir. Ortalama sinyal enerjisinin birim enerjiye eşit olduğu durumda R_1 ve R_2 yarıçapları (3.54)'de verilen denklem ile bulunur.

$$4 \times R_1^2 + 12 \times R_2^2 = 16 \quad (3.54)$$

16-APSK sembolleri parçalı fonksiyonu şeklinde yazılarak (3.55)'te verilen denklem ile bulunabilir.



Şekil 3.13 : 16-APSK yıldız diyagramı.

$$s_i(t) = s_i^1 + s_i^2 \quad (3.55)$$

$$s_i^1(t) = R_1 \cos(2\pi f_c t + \varphi_i), \quad 0 \leq t \leq T, \quad i = 1,2,3,4 \quad (3.56)$$

$$\varphi_i = \frac{(2i-1)\pi}{4}, \quad i = 1,2,3,4 \quad (3.57)$$

$$s_i^2(t) = R_2 \cos(2\pi f_c t + \varphi_i), \quad 0 \leq t \leq T, \quad i = 5,6, \dots, 16 \quad (3.58)$$

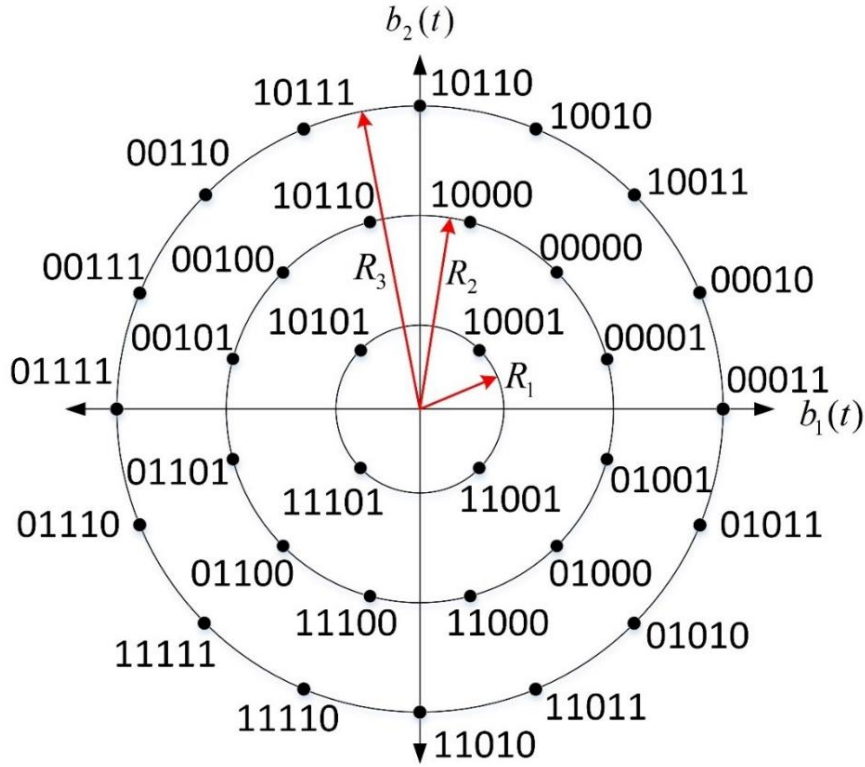
$$\varphi_i = \frac{(2(i-4)-1)\pi}{12}, \quad i = 5,6, \dots, 16 \quad (3.59)$$

32-APSK

32-APSK kiplemesinde bir sembol ile 5 bit iletilmektedir. Bu kipleme için kullanılan Gray kodlamalı yıldız diyagramı Şekil 3.14'te verilmiştir.

32-APSK yıldız diyagramı eş merkezli farklı yarıçaplı üç halkadan oluşmaktadır. Dış halkada birbirlerine eşit uzaklıkta 16 sembol, orta halkada birbirine eşit uzaklıkta 12 sembol ve iç halkada birbirlerine eşit uzaklıkta 4 sembol bulunmaktadır. Halkaların yarıçapları oranları R_2/R_1 ve R_3/R_1 kipleme ile kullanılan kodlamanın kod oranlarına göre belirlenmektedir. Ortalama sinyal enerjisinin birim enerjiye eşit olduğu durumda R_1 , R_2 ve R_3 yarıçapları (3.60)'da verilen denklem ile bulunur.

$$4 \times R_1^2 + 12 \times R_2^2 + 16 \times R_3^2 = 32 \quad (3.60)$$



Şekil 3.14 : 32-APSK yıldız diyagramı.

32-APSK sembolleri parçalı fonksiyonu şeklinde yazılarak (3.61)'de verilen denklem ile bulunabilir.

$$s_i(t) = s_i^1 + s_i^2 + s_i^3 \quad (3.61)$$

$$s_i^1(t) = R_1 \cos(2\pi f_c t + \varphi_i), \quad 0 \leq t \leq T, \quad i = 1,2,3,4 \quad (3.62)$$

$$\varphi_i = \frac{(2i-1)\pi}{4}, \quad i = 1,2,3,4 \quad (3.63)$$

$$s_i^2(t) = R_2 \cos(2\pi f_c t + \varphi_i), \quad 0 \leq t \leq T, \quad i = 5,6, \dots, 16 \quad (3.64)$$

$$\varphi_i = \frac{(2(i-4)-1)\pi}{12}, \quad i = 5,6, \dots, 16 \quad (3.65)$$

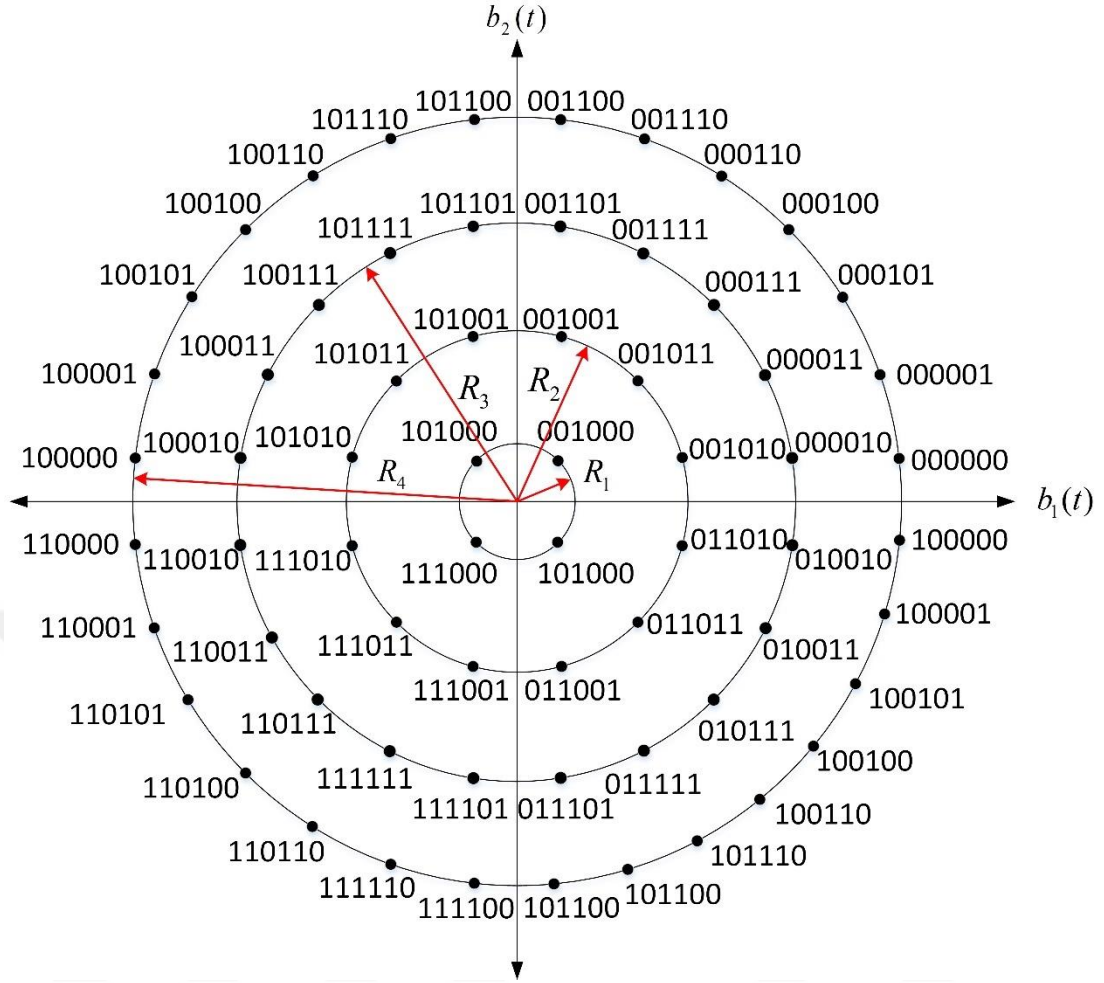
$$s_i^3(t) = R_3 \cos(2\pi f_c t + \varphi_i), \quad 0 \leq t \leq T, \quad i = 16,17, \dots, 32 \quad (3.66)$$

$$\varphi_i = \frac{(2(i-16)-1)\pi}{16}, \quad i = 17,18, \dots, 32 \quad (3.67)$$

64-APSK

64-APSK kiplemesinde bir sembol ile 6 bit iletilmektedir. Bu kipleme için kullanılan Gray kodlamalı yıldız diyagramı Şekil 3.15'te verilmiştir.

64-APSK yıldız diyagramı eş merkezli farklı yarıçaplı dört halkadan oluşmaktadır.



Şekil 3.15 : 64-APSK yıldız diyagramı.

Dördüncü halka birbirlerine eşit uzaklıkta 28 sembol, üçüncü halka birbirine eşit uzaklıkta 20 sembol, ikinci halka birbirine eşit uzaklıkta 12 sembol ve birinci halka birbirlerine eşit uzaklıkta 4 sembol bulunmaktadır. Halkaların yarıçapları oranları R_2/R_1 , R_3/R_1 ve R_4/R_1 kipleme ile kullanılan kodlamanın kod oranlarına göre belirlenmektedir. Ortalama sinyal enerjisinin birim enerjiye eşit olduğu durumda R_1 , R_2 , R_3 ve R_4 yarıçapları (3.68)'de verilen denklem ile bulunur.

$$4 \times R_1^2 + 12 \times R_2^2 + 20 \times R_3^2 + 28 \times R_4^2 = 64 \quad (3.68)$$

64-APSK sembolleri parçalı fonksiyonu şeklinde yazılarak (3.69)'da verilen denklem ile bulunabilir.

$$s_i(t) = s_i^1 + s_i^2 + s_i^3 + s_i^4 \quad (3.69)$$

$$s_i^1(t) = R_1 \cos(2\pi f_c t + \varphi_i), \quad 0 \leq t \leq T, \quad i = 1,2,3,4 \quad (3.70)$$

$$\varphi_i = \frac{(2i-1)\pi}{4}, \quad i = 1,2,3,4 \quad (3.71)$$

$$s_i^2(t) = R_2 \cos(2\pi f_c t + \varphi_i), \quad 0 \leq t \leq T, \quad i = 5,6, \dots, 16 \quad (3.72)$$

$$\varphi_i = \frac{(2(i-4)-1)\pi}{12}, \quad i = 5,6, \dots, 16 \quad (3.73)$$

$$s_i^3(t) = R_3 \cos(2\pi f_c t + \varphi_i), \quad 0 \leq t \leq T, \quad i = 17,18, \dots, 36 \quad (3.74)$$

$$\varphi_i = \frac{(2(i-16)-1)\pi}{20}, \quad i = 17,18, \dots, 36 \quad (3.75)$$

$$s_i^4(t) = R_4 \cos(2\pi f_c t + \varphi_i), \quad 0 \leq t \leq T, \quad i = 37,38, \dots, 64 \quad (3.76)$$

$$\varphi_i = \frac{(2(i-36)-1)\pi}{28}, \quad i = 37,38, \dots, 64 \quad (3.77)$$

3.4.1.3 M-QCI

QAM dairesel eşörüntülü kipleme yöntemi temel olarak düzgün dağılımı olmayan APSK kipleme yöntemi olarak nitelendirilebilir. Bu kipleme yöntemi kare ve dairelerin dairesel eşörüntülü eşleşmesi temelinde geliştirilmiştir. Elde edilen yıldız diyagramları M-QAM yıldız diyagramlarının eşlenmiş halidir. Tepe-ortalama güç kısıtı olan sistemlerdeki yüksek dereceli kiplemelerde QAM kiplemelerinden daha yüksek başarımlar göstermektedir.

M-QAM kiplemelerinde gönderilen sinyalin denklemi ve (3.43) ve (3.44)'te verilen taban fonksiyonları cinsinden ifadesi sırası ile (3.78) ve (3.79)'da verilmiştir [16].

$$s_i(t) = A_i \cos \varphi_i \cos(2\pi f_c t) - A_q \sin \varphi_i \sin(2\pi f_c t) \quad (3.78)$$

$$s_i(t) = s_{i1} b_1(t) + s_{i2} b_2(t) \quad (3.79)$$

(3.79)'da verilen s_{i1} ve s_{i2} değerleri $s_i(t)$ sinyalinin yıldız diyagramı üzerindeki koordinatlarını (s_{i1}, s_{i2}) belirtmektedir.

Kare ve daire eşörüntülü yapılarıdır. Bu yapılar arasında sürekli ve birebir örten eşleme ve ters eşleme vardır. Bu kipleme yönteminde dairesel eşörüntülü eşleme ile kare üzerindeki noktalar daire üzerine eşlenmiştir. Bu eşlemenin fonksiyonu (3.80)'de ve görsel gösterimi Şekil 3.16'da verilmiştir [26].

$$f(x, y) = \begin{cases} \frac{\sqrt{2} \max(|x|, |y|)}{\sqrt{x^2 + y^2}}(x, y) & \text{if } (x, y) \neq (0, 0) \\ (0, 0) & \text{if } (x, y) = (0, 0) \end{cases} \quad (3.80)$$

Bu dairesel eşleme ile M-QAM yıldız diyagramının Gray kodlama özelliği korunmaktadır. Ayrıca her M-QAM yıldız diyagramı için tek bir M-QCI yıldız diyagramı elde edilmektedir.

128-QCI

128-QCI kiplemesinde bir sembol ile 7 bit iletilmektedir. Bu kipleme için kullanılan 128-QAM yıldız diyagramı ve 128-QAM işaretlerine dairesel eşörüntü eşlemesi uygulanması ile elde edilen 128-QCI yıldız diyagramı Şekil 3.17’de verilmektedir. Yıldız diyagramında derecesinin artması sebebiyle sembollerin bit gösterimlerine yer verilmemiştir.

128-QAM kiplemesi yapılırken sinyallerin ortalama enerjisi birim enerji olacak şekilde tasarlanmış, 128-QCI dönüşümü ile sinyallerin ortalama enerjisi korunmuştur.

256-QCI

256-QCI kiplemesinde bir sembol ile 8 bit iletilmektedir. Bu kipleme için kullanılan 256-QAM yıldız diyagramı ve 256-QAM işaretlerine dairesel eşörüntü eşlemesi uygulanması ile elde edilen 256-QCI yıldız diyagramı Şekil 3.18’de verilmektedir. Yıldız diyagramında derecesinin artması sebebiyle sembollerin bit gösterimlerine yer verilmemiştir.

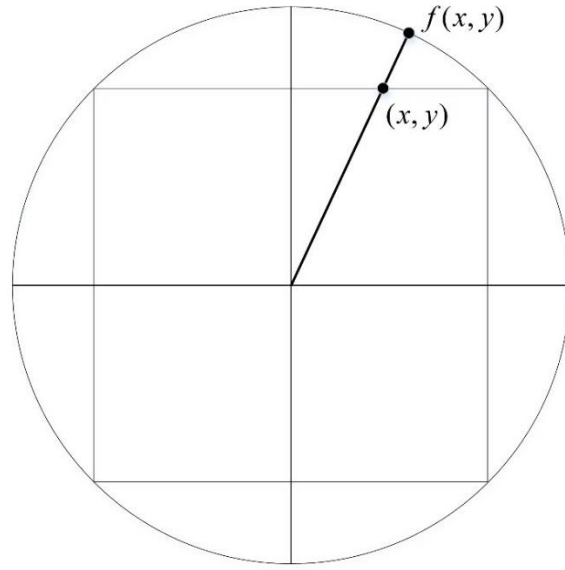
256-QAM kiplemesi yapılırken sinyallerin ortalama enerjisi birim enerji olacak şekilde tasarlanmış, 256-QCI dönüşümü ile sinyallerin ortalama enerjisi korunmuştur.

3.4.2 Çerçeve başlığı ekleme

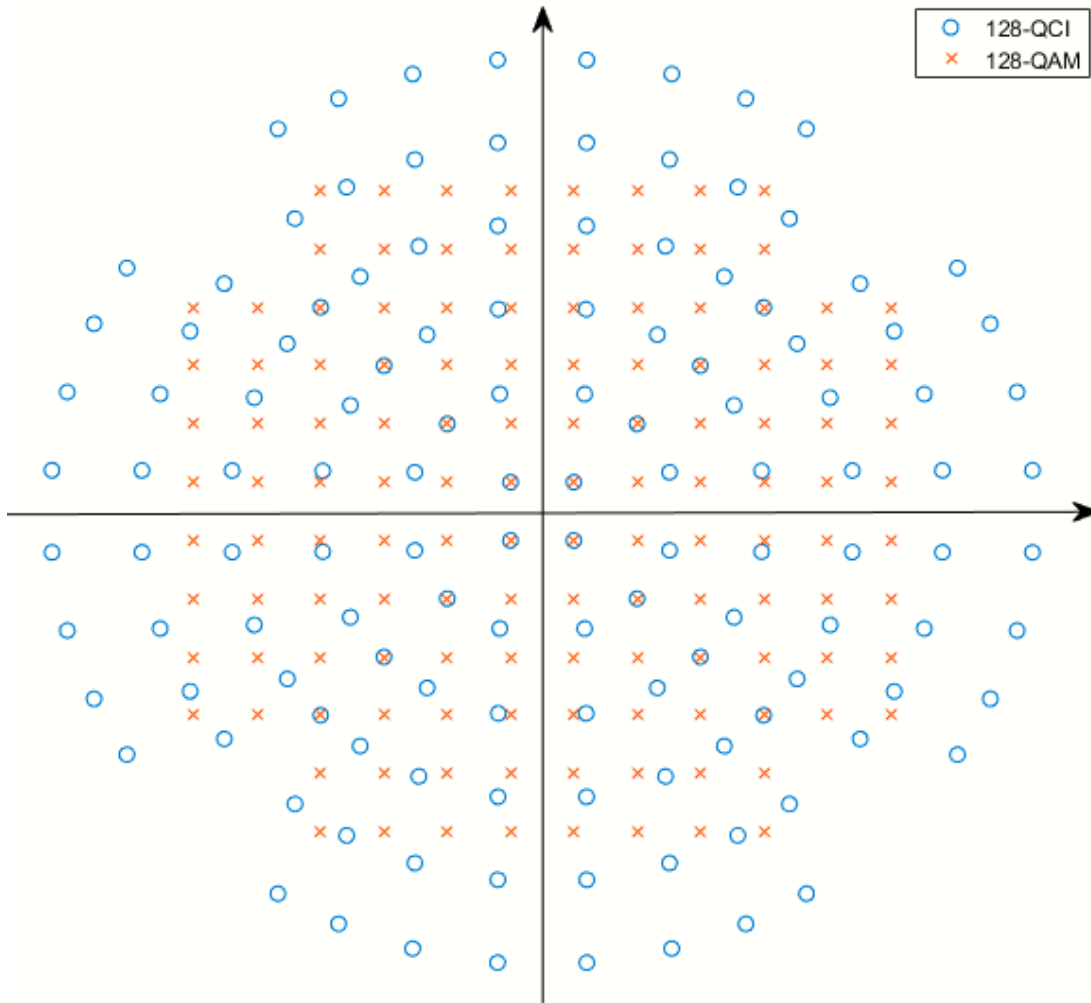
Bir fiziksel katman çerçevesi çerçeve başlığı ve kod sözcüğü bloklarından oluşmaktadır. Çerçeve başlığı çerçeve belirteci ve çerçeve tanımlayıcı bölümlerinden oluşmaktadır. Bir çerçeve içerisinde 256 sembol çerçeve belirteci, 64 sembol çerçeve tanımlayıcı ve 16 kod sözcüğü bloğu bulunmaktadır. Her kod sözcüğü bloğunda 8100 veri sembolü ve eklenirse 240 pilot sembolü bulunmaktadır. Bir fiziksel katman çerçevesinin yapısı Şekil 3.19’da verilmiştir.

3.4.2.1 Çerçeve belirteci

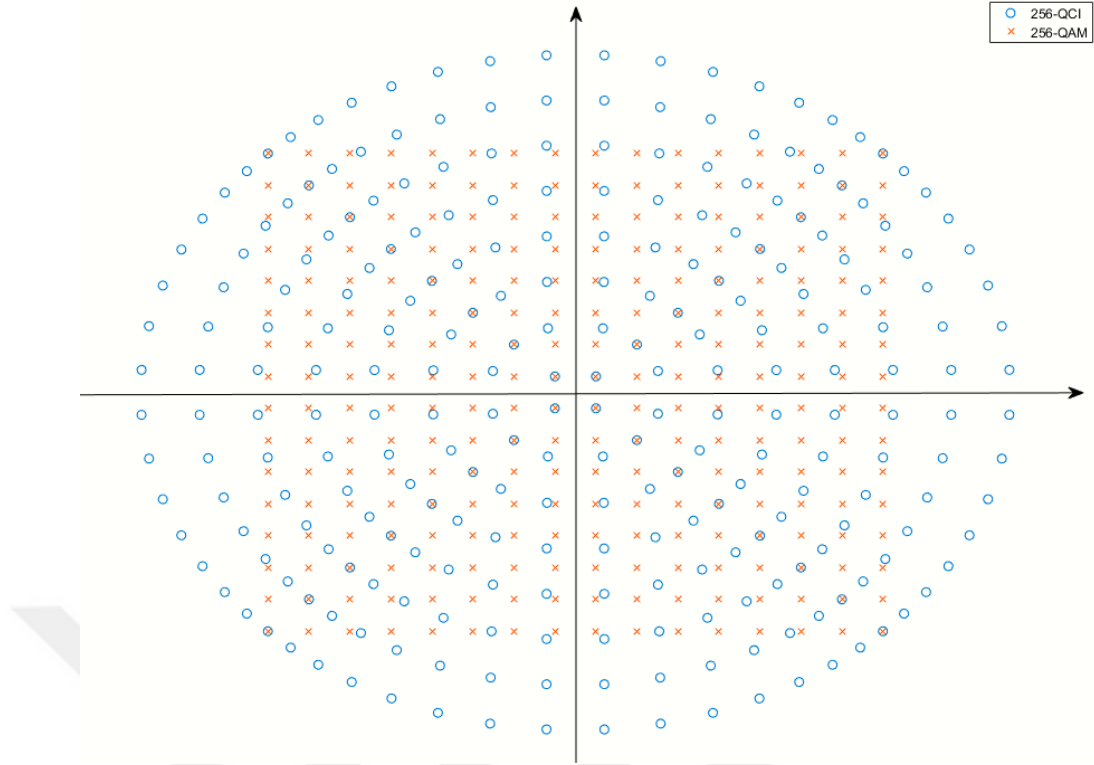
Çerçeve belirteci çerçeve başlangıcını yakalamak ve eşzamanlama için bilinen 256 sembol uzunluğunda bir dizidir. Çerçeve belirteci sembolleri $\pi/2$ -BPSK kiplemesi ile kiplenmiştir. Kipleme detayları ilerleyen bölümlerde anlatılmaktadır.



Şekil 3.16 : Kare ve daire dairesel eşörüntülü eşleme.



Şekil 3.17 : 128-QAM ve 128-QCI yıldız diyagramları.



Şekil 3.18 : 256-QAM ve 256-QCI yıldız diyagramları.

İşaret belirteci (3.81) ve (3.82)'de verilen geri besleme polinomları kullanılarak üretilen altın seriden oluşmaktadır.

$$h_1(x) = x^8 + x^6 + x^5 + x^4 + 1 \quad (3.81)$$

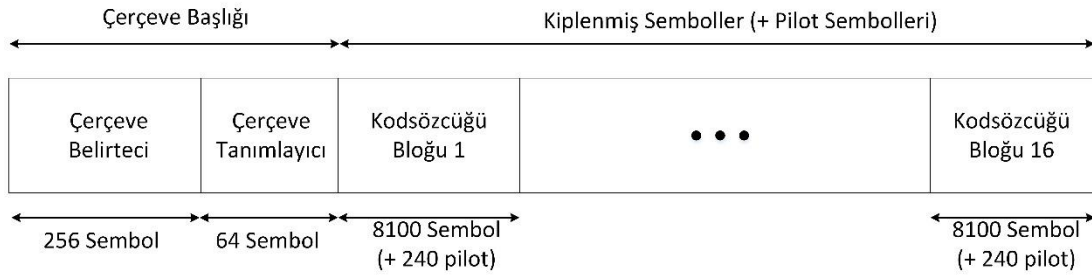
$$h_2(x) = x^8 + x^6 + x^5 + x^4 + x^3 + x + 1 \quad (3.82)$$

Altın serinin üretilmesinde kullanılan kayan yazmaçlı yapı ve yazmaçların başlangıç değerleri Şekil 3.20'de verilmiştir.

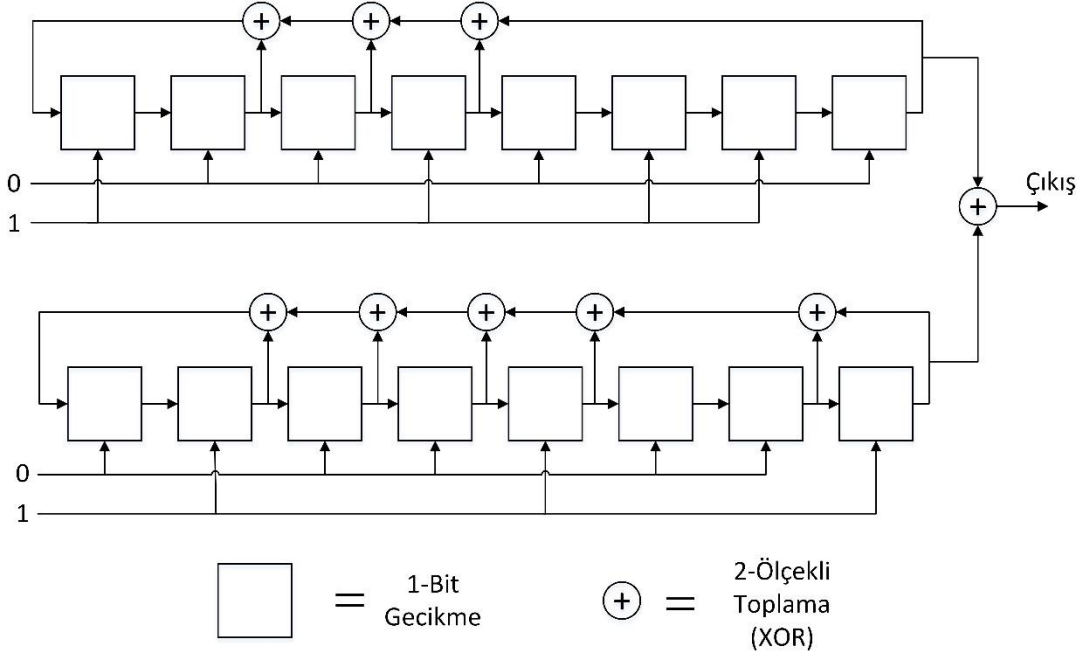
3.4.2.2 Çerçeve tanımlayıcı

Çerçeve tanımlayıcı 6 bit uzunluğunda ACM formatını ve 1 bit uzunluğunda pilot eklenip eklenmeyeceği bilgisini alarak 64 bit uzunluğunda bir dizi oluşturur. Bu dizi(64,7) blok kodlama ile oluşturulmaktadır. Burada kullanılan blok kodlama üreteç matrisi kullanarak kodlama yapmaktadır. Bu kodlayıcıda (64,7) blok kodlama 7 bit uzunluğundaki bilgiyi alarak 64 bit uzunluğundaki kod sözcüğü oluşturur. Bu kodlamada bilgi içeriğinin yapısı ve kullanılan üreteç matrisi sırasıyla (3.83) ve (3.84)'te verilmiştir.

$$m_{ct} = [b_1 \ b_2 \ b_3 \ b_4 \ b_5 \ b_6 \ b_7] \quad (3.83)$$



Şekil 3.19 : Fiziksel katman çerçeve yapısı.



Şekil 3.20 : Çerçeve belirteci üretimi.

$$G_{ct} = \begin{bmatrix} 01 \\ 0011 \\ 000011110000111100001111000011110000111100001111000011110000111100001111 \\ 0000000011111111000000001111111100000000111111110000000011111111 \\ 000000000000000011111111111110000000000000001111111111111111 \\ 0011111111111111111111111111111111 \\ 11 \end{bmatrix} \quad (3.84)$$

Bilgi bitleri b_1, b_2, \dots, b_6 ACM formatı hakkında bilgi içerirken, b_7 biti pilot eklenip eklenmeyeceği bilgisini içermektedir. Blok kodlamalarda kod sözcüğü bilgi bitleri ile üreteç matrisinin çarpılmasıyla elde edilir. Bilgi bitleri 1×7 boyutunda vektör olarak tanımlanır ve üreteç matrisi 7×64 boyutunda bir matris olduğu için kod sözcüğü 1×64 boyutunda vektör olarak elde edilir. Bu kodlayıcı yapısı (3.85)'te verilmiştir.

$$c_{ct} = m_{ct} \times G_{ct} \quad (3.85)$$

Blok kodlayıcı sonrası elde edilen 64 bit uzunluğundaki kod sözcüğüne 64-bit

uzunluğundaki bir dizi ile rastgeleliği artırmak amacıyla bitset XOR işlemi uygulanır. Bu dizi (3.86)'da verilmiştir.

$$[0111100011001110110000011110010010101001101000010001011011111010] \quad (3.86)$$

3.4.2.3 Çerçeve başlığı kiplemesi

Çerçeve başlığı 256 bit uzunluğundaki çerçeve belirteci ve 64 bit uzunluğundaki çerçeve tanımlayıcıdan oluşmaktadır. 320 bit uzunluğundaki çerçeve başlığına $\pi/2$ -BPSK kiplemesi uygulanır.

Çerçeve başlığı bitlerini $(t_1, t_2, \dots, t_{320})$ olarak tanımlarsak, $\pi/2$ -BPSK kiplenmiş sembollere ait eş fazlı (I) ve dördün (Q) bileşenler (3.87)'de verilen denklem kullanılarak hesaplanır.

$$\begin{aligned} I_{2i-1} &= Q_{2i-1} = \frac{1}{\sqrt{2}}(1 - 2t_{2i-1}) \\ I_{2i} &= -Q_{2i} = -\frac{1}{\sqrt{2}}(1 - 2t_{2i-1}) \end{aligned} \quad i = 1, 2, \dots, 160 \quad (3.87)$$

3.4.3 Pilot işareti ekleme

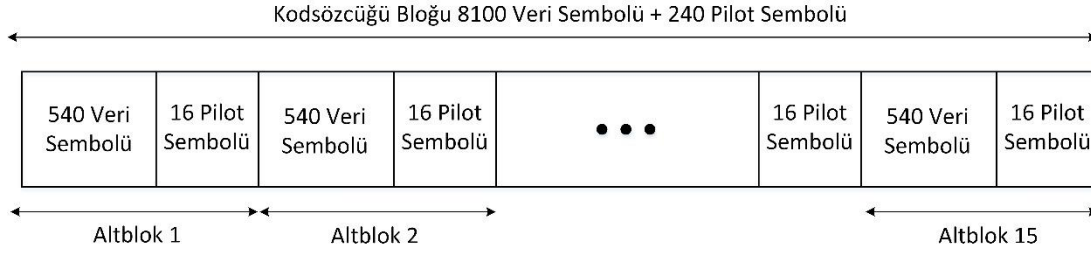
Pilot sembolleri alıcısındaki taşıyıcı, faz ve zamanlama kestirimlerini ve eşzamanlamayı kolaylaştırmak amacıyla isteğe bağlı olarak eklenebilmektedir. Pilot eklenip eklenmeyeceği bilgisi çerçeve tanımlayıcıdaki bilgi bitlerinden b_7 ile iletilmektedir.

Pilot sembollerinin eklenmesi durumunda, her kod sözcüğü bloğu 8100 veri sembolüne ek olarak 240 pilot sembolü içerir. Pilot sembolleri veri sembolleri arasına dağıtılmaktadır. 540 veri sembolü ve sonrasında 16 pilot sembolü eklenerek 556 sembolük altbloklar oluşturulmakta ve her kod sözcüğü bloğu 15 altblok ile tamamlanmaktadır. Pilot dağılımı görsel olarak Şekil 3.21'de verilmiştir.

Pilot sembollerinin eş fazlı (I) ve dördün (Q) bileşenleri $1/\sqrt{2}$ olarak tanımlanmıştır.

3.4.4 Fiziksel katman rastgeleştirme

Fiziksel katman rastgeleştirme, tekrarlı veri ve pilot sembollerinin dağılımı için yapılmaktadır. Çerçeve başlığı dışındaki 16 kod sözcüğü bloğuna uygulanır. Fiziksel katman rastgeleştirme, eş fazlı ve dördün bileşenlerin karmaşık rastgele bir seri $(C_I + jC_Q)$ ile çarpılmasıyla elde edilir. Bu işlem eş fazlı ve dördün bileşenler için sırasıyla (3.88) ve (3.89)'da verilmiştir.



Şekil 3.21 : Pilot sembollerinin yerleşimi.

$$I_{rastgele} = \{I \cdot C_I + Q \cdot C_Q\} \quad (3.88)$$

$$Q_{rastgele} = \{Q \cdot C_I + I \cdot C_Q\} \quad (3.89)$$

Karmaşık rastgele $(C_I + jC_Q)$ serisi, 18 dereceli üreteç polinomlar ile tanımlanan gerçel ve farklı x ve y serilerinin kombinasyonu ile bir altın seri oluşturularak elde edilir. Gerçel x ve y serilerinin üreteç polinomları sırasıyla (3.90) ve (3.91)'de verilmiştir.

$$g_x(x) = x^{18} + x^7 + 1 \quad (3.90)$$

$$g_y(y) = y^{18} + y^{10} + y^7 + y^5 + 1 \quad (3.91)$$

Gerçel x ve y serilerinin başlangıç değerleri sırasıyla (3.92) ve (3.93)'te verilmiştir.

$$x(0) = 1, \quad x(1) = x(2) = \dots = x(16) = x(17) = 0 \quad (3.92)$$

$$y(0) = y(1) = \dots = y(16) = y(17) = 1 \quad (3.93)$$

Serinin oluşturulmasındaki adımlar x ve y serileri için sırasıyla (3.94) ve (3.95)'te verilmiştir.

$$\begin{aligned} x_{k+1}(i) &= x_k(7) + x_k(0), & i &= 17 \\ x_{k+1}(i) &= x_k(i + 1) & 0 \leq i < 17 \end{aligned} \quad k = 1, 2, \dots \quad (3.94)$$

$$\begin{aligned} y_{k+1}(i) &= y_k(10) + y_k(7) + y_k(5) + y_k(0), & i &= 17 \\ y_{k+1}(i) &= y_k(i + 1) & 0 \leq i < 17 \end{aligned} \quad k = 1, 2, \dots \quad (3.95)$$

Bu gerçel seriler ile oluşturulan altın seri Z denklemi (3.96)'da verilmiştir.

$$\begin{aligned} Z(k) &= (x_k(15) + x_k(6) + x_k(4) + y_k(15) + y_k(14) + y_k(13) + y_k(12) + y_k(11) + \\ & \quad y_k(10) + y_k(9) + y_k(8) + y_k(7) + y_k(6) + y_k(4)) \bmod 2 \quad k = 1, 2, \dots \end{aligned} \quad (3.96)$$

Elde edilen Z altın serisi kullanılarak tamsayı $R_{rastgele}$ serisi (3.97)'de belirtilen

şekilde tanımlanmıştır.

$$R_{rastgele}(k) = (2 \cdot Z(k) + x_k(0) + y_k(0)) \bmod 4 \quad i = 1, 2, \dots \quad (3.97)$$

Gerçel R serisinin oluşturulduğu yapı görsel olarak Şekil 3.22’de verilmiştir.

Gerçel tamsayı serisi R kullanılarak karmaşık rastgele $(C_I + jC_Q)$ serisi (3.98)’de verilen şekilde tanımlanmıştır.

$$(C_I(i) + jC_Q(i)) = e^{-jR(i)\frac{\pi}{2}} \quad (3.98)$$

(3.97)’de tanımlanan R serisinin olası değerleri, bu değerlere karşılık gelen karmaşık rastgele seri ve eş fazlı (I) ve dördün (Q) bileşenlerle çarpımı sonucunda elde edilen $I_{rastgele}$ ve $Q_{rastgele}$ değerleri Çizelge 3.4’te verilmiştir.

Bu rastgeleleştirme işlemi çerçeve sonunda başlangıç değerlerine alınır ve her çerçeve için baştan uygulanır.

3.5 Tabanbant Süzme

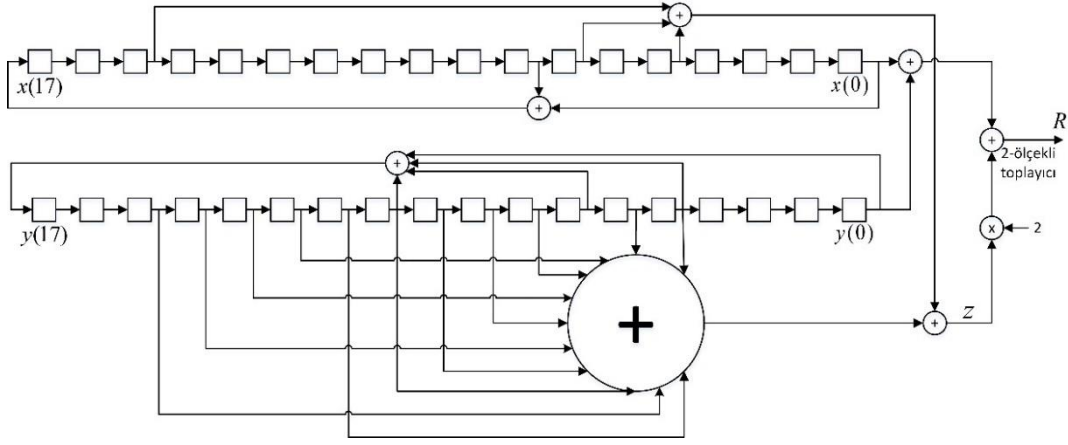
3.5.1 Darbe şekillendirici süzgeç

Haberleşme sistemlerinde kipleme kaynaklı faz atlamalarını önlemek, bant-sınırlı kanal oluşturmak ve semboller arası girişimleri önlemek amacıyla darbe şekillendirici süzgeç kullanılmaktadır. Sayısal olarak gönderilecek sinyaller kare dalga formunda olduğu için bant genişlikleri sınırsız çıkmaktadır. İletilecek her sembol darbe şekillendirici süzgeçten geçerek iletildiğinde, iletilen sinyallerin bant genişliği sınırlandırılabilir. Ayrıca alıcıda uyumlu süzgeç kullanılması durumunda iletilen semboller arasındaki girişimin önüne geçilmektedir [27].

Yükseltilmiş kosinüs veya kök yükseltilmiş kosinüs süzgeçleri en yaygın olarak kullanılan darbe şekillendirici süzgeçlerdir. Bu süzgeçlerde artık bant genişliği ayarlanabilmektedir. Böylece elde edilen spektral verimlilik veya daha basit süzgeç tasarımı arasındaki ödünleşmede amaca uygun olarak tasarlanabilirler. Yükseltilmiş kosinüs süzgeci kullanıldığında bir sembolün örnekleme anında diğer sembollerin değerleri 0 olur ve böylece semboller arası girişimin önüne geçilmiş olur. Yükseltilmiş kosinüs süzgeci kullanılan sistemlerde, alıcıda uyumlu süzgeç kullanılmaması durumunda alınan gürültü toplanır beyaz Gauss gürültüsü özelliğini kaybeder. Bu yüzden alıcılarda da uyumlu süzgeç kullanımı gerektirir.

3.5.1.1 Kök yükseltilmiş kosinüs süzgeci

Kullanılan kök yükseltilmiş kosinüs süzgecine ait frekans yanıtı (3.99)’da verilmiştir.



Şekil 3.22 : Fiziksel katman rastgeleştirici yapısı.

Çizelge 3.4 : Fiziksel katman rastgeleştirici.

$R(i)$	$e^{-jR(i)\frac{\pi}{2}}$	$I_{rastgele}(i)$	$Q_{rastgele}(i)$
0	1	$I(i)$	$Q(i)$
1	j	$-Q(i)$	$I(i)$
2	-1	$-I(i)$	$-Q(i)$
3	$-j$	$Q(i)$	$-I(i)$

$$H(f) = \begin{cases} 1, & |f| < f_N \cdot (1 - \alpha) \\ \left[\frac{1}{2} + \frac{1}{2} \sin \frac{\pi}{2f_N} \left(\frac{f_N - |f|}{\alpha} \right) \right]^{1/2}, & f_N \cdot (1 - \alpha) < |f| < f_N \cdot (1 + \alpha) \\ 0, & |f| > f_N \cdot (1 + \alpha) \end{cases} \quad (3.99)$$

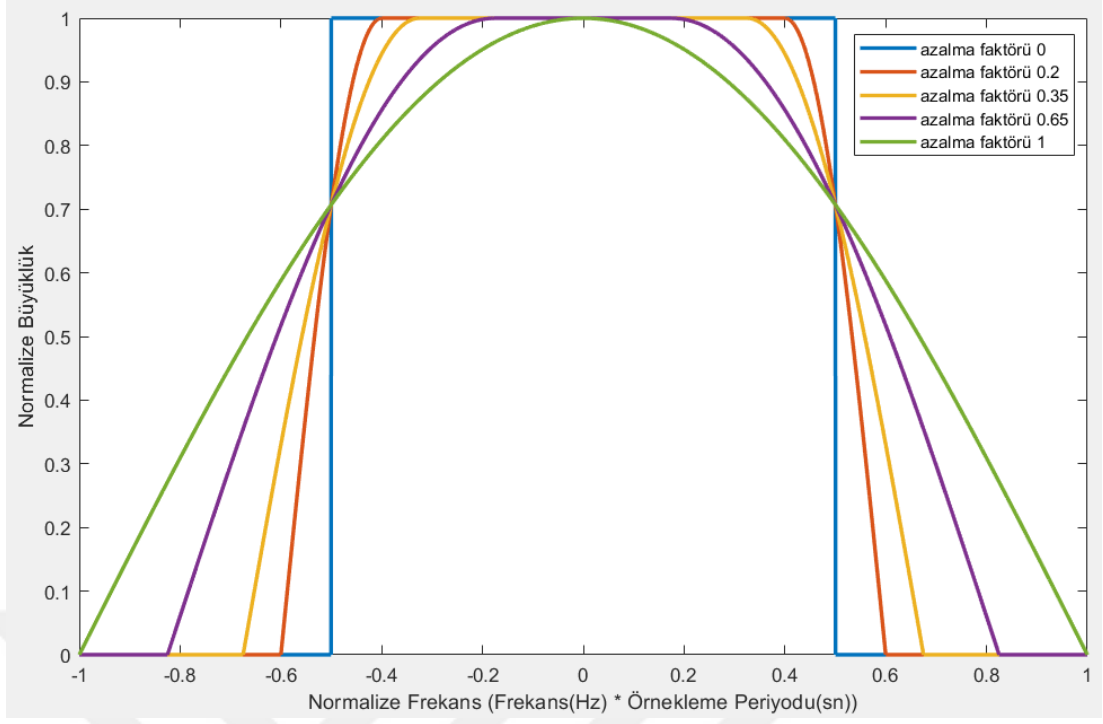
$$f_N = \frac{1}{2T_s} = \frac{R_s}{2} \quad (3.100)$$

(3.100)'de Nyquist bantgeniřlięi f_N sembol hızı R_s cinsinden verilmiřtir.

Kök yükseltilmiř kosinüs süzgeçlerinin en önemli deęiřkeni azalma faktörüdür. Azalma faktörünün 0 olduęu durumu Nyquist bantgeniřlięine yaklařmakta ancak zamanlama kaymalarından oluřabilecek semboller arası giriřimi artırmakta ve alıcı yapısını zorlařtırmaktadır. Azalma faktörünün 1 olması durumunda iřaretin bantgeniřlięinin 2 katı kadar bantgeniřlięine ihtiyaç duyulmaktadır. Kullanılan sistem gereksinimlerine uygun olan azalma faktörü deęeri seçilebilir.

Farklı azalma faktörlerine göre frekans yanıtı grafikleri Şekil 3.23'te verilmiřtir.

Bu sistemde kullanılan azalma faktörü $\alpha = 0,35$ olarak seçilmiřtir.



Şekil 3.23 : RRC süzgeci farklı azalma faktörlerine göre frekansı yanıtı.

4. UYARLAMALI KODLAMA VE KİPLEME KULLANAN VERİCİ FPGA TASARIMI

Bu bölümde, geliştirilen uyarlamalı kodlama ve kipleme kullanan verici yapısının FPGA tasarımı açıklanmıştır. FPGA tasarımı için referans olması açısından öncelikle yapıdaki her blok MATLAB ile modellenmiştir. MATLAB modeli kaynak alınarak FPGA tasarımı yapılmıştır.

4.1 Mod Uyarlama

4.1.1 Giriş arayüzü

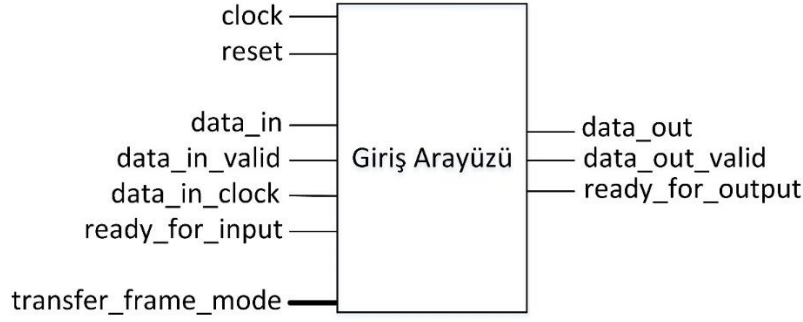
Giriş arayüzü bloğu FPGA dışından gelen sinyallerin alınması ve farklı saat sinyali alanından gelmesi durumunda FPGA tasarımının sistem saat sinyaline eşzamanlamasının yapılmasını sağlamaktadır.

Giriş arayüzü bloğu FPGA tasarımının arayüz diyagramı Şekil 4.1'de ve arayüz sinyallerinin açıklamaları Çizelge 4.1'de verilmiştir.

Giriş arayüzü bloğuna başlangıçta sistem yeniden başlatma sinyali gelir ve tasarımda kullanılan iç sinyaller başlangıç değerlerini alır. Sistem yeniden başlatma sinyalinden sonra bileşimli bir devre ile iletim çerçeve kipine göre giriş verisi olarak gelecek iletim çerçeve uzunluğu Çizelge 4.2'de belirtilen şekilde seçilir.

Giriş arayüzü bloğunda sistem başlangıç durumuna getirildiğinde ve sistem veri girişine hazır olduğunda girişe hazır sinyali 1 yapılır. Dışarıdan gelecek olan veri bu sinyale göre gönderilir. Girişe hazır sinyali 0 ise veri akışı durdurulur.

Dışarıdan gelecek olan giriş verisi, giriş verisi geçerli ve giriş verisi saat sinyaliyle birlikte alınır. Farklı saat sinyali alanından gelen bu sinyallerin eşzamanlaması için eşzamanlamasız ilk-giren ilk-çıkartıcı hafıza birimi kullanılmaktadır. Bu birimde farklı saat sinyali alanından gelen giriş verisi ve giriş verisi geçerli sinyallerini giriş verisi saat sinyaline göre alınır ve bir hafızaya yazılır. Çıkışa hazır sinyalinin 1 gelmesi durumunda, bir iletim çerçevesi uzunluğundaki veri hafızadan okunur ve çıkış verisi



Şekil 4.1 : Fiziksel katman rastgeleleştirici yapısı.

Çizelge 4.1 : Giriş arayüzü bloğu FPGA tasarımı arayüz sinyal bilgileri.

Sinyal	Giriş/ Çıkış	Bit Genişliği	Tanım
clock	giriş	1	sistem saat sinyali
reset	giriş	1	sistem yeniden başlatma
data_in	giriş	1	giriş verisi
data_in_valid	giriş	1	giriş verisi geçerli
data_in_clock	giriş	1	giriş verisi saat sinyali
ready_for_input	çıkış	1	girişe hazır
transfer_frame_mode	giriş	2	iletim çerçevesi kipi
data_out	çıkış	1	çıkış verisi
data_out_valid	çıkış	1	çıkış verisi geçerli
ready_for_output	giriş	1	çıkışa hazır

Çizelge 4.2 : İletim çerçevesi uzunluğu.

İletim Çerçevesi Kipi	İletim Çerçevesi Uzunluğu
0	1738-bit
1	3568-bit
2	7136-bit
3	8920-bit

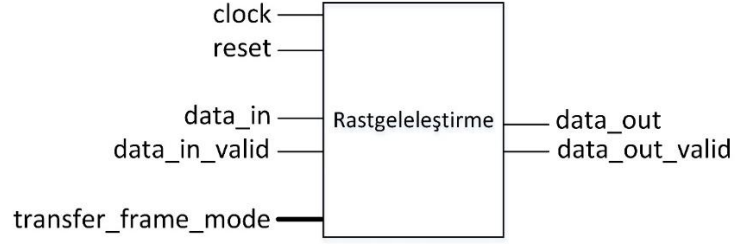
geçerli sinyali 1 yapılarak gönderilir. Her iletim çerçeveleri arasında 32-bit uzunluğundaki eşzamanlama işareti eklenmesi için boşluk bırakılarak çıkışa hazır sinyali geldiği sürece iletme devam edilir.

4.1.2 Rastgeleleştirme

Rastgeleleştirme bloğu tekrarlı verilerin dağıtılması işlemi için kullanılmaktadır.

Rastgeleleştirme bloğu FPGA tasarımının arayüz diyagramı Şekil 4.2’de ve arayüz sinyallerinin açıklamaları Çizelge 4.3’te verilmiştir.

Rastgeleleştirme bloğunda başlangıçta sistem yeniden başlatma sinyali gelir ve



Şekil 4.2 : Rastgeleleştirme bloğu FPGA tasarımı arayüz sinyalleri.

Çizelge 4.3 : Rastgeleleştirme bloğu FPGA tasarımı arayüz sinyal bilgileri.

Sinyal	Giriş/ Çıkış	Bit Genişliği	Tanım
clock	giriş	1	sistem saat sinyali
reset	giriş	1	sistem yeniden başlatma
data_in	giriş	1	giriş verisi
data_in_valid	giriş	1	giriş verisi geçerli
transfer_frame_mode	giriş	2	iletim çerçevesi kipi
data_out	çıkış	1	çıkış verisi
data_out_valid	çıkış	1	çıkış verisi geçerli

tasarımda kullanılan iç sinyaller başlangıç değerleri alır. Sistem yeniden başlatma sinyalinden sonra bileşimli bir devre ile iletim çerçeve kipine göre giriş verisi olarak gelecek iletim çerçeve uzunluğu Çizelge 4.2’de belirtilen şekilde seçilir.

Rastgeleleştirme bloğunda sistem başlangıç durumuna getirildiğinde ve sistem veri girişine hazır olduğunda giriş verisi geçerli sinyalini takip etmeye başlanır. Giriş verisi geçerli sinyalinin 1 olduğu durumda giriş verisi alınır ve o anki kayan yazmaç çıkışı ile 2-ölçekli toplama (XOR) işlemi yapılır. XOR işlemi sonucunu çıkış verisi geçerli sinyalini 1 yapılarak iletir. Aynı saat sinyalinde kayan yazmaç değerleri uygun şekilde güncellenir. Böylece rastgeleleştirme bloğunda başlangıçta bir saat sinyali gecikmeli olmakla beraber her saat sinyalinde giriş alınıp her saat sinyalinde çıkış verilebilmektedir. İletim çerçevesi boyunca bu işlem devam eder. İletim çerçevesinin bittiği iletim çerçevesi uzunluğuna kadar sayan bir sayaç ile anlaşılır ve kayan yazmaç değerlerini başlangıç değerlerine getirilir. Yeni iletim çerçevesi için rastgeleleştirme işlemi baştan başlatılır.

4.1.3 Bölücü

Bölücü bloğunda rastgeleleştirilmiş iletim çerçeveleri alınır ve aralarına eşzamanlama işaretleri eklenir.

Bölücü bloğu FPGA tasarımının arayüz diyagramı Şekil 4.3'te ve arayüz sinyallerinin açıklamaları Çizelge 4.4'te verilmiştir.

Bölücü bloğunda başlangıçta sistem yeniden başlatma sinyali gelir ve tasarımda kullanılan iç sinyaller başlangıç değerleri alır. Sistem yeniden başlatma sinyalinden sonra bileşimli bir devre ile iletim çerçeve kipine göre iletim çerçeve uzunluğu Çizelge 4.2'de belirtilen şekilde seçilir. Bölücü bloğunda sistem başlangıç durumuna getirildiğinde ve sistem veri girişine hazır olduğunda giriş verisi geçerli sinyali takip edilmeye başlanır. Giriş verisi geçerli sinyali 1 olduğunda giriş verisi alınır ve çıkış verisi geçerli sinyali 1 yapılarak iletilir. Bir iletim çerçevesi uzunluğu boyunca giriş verileri çıkış olarak verilir. İletim çerçevesinin bittiği iletim çerçevesi uzunluğuna kadar sayan bir sayaç ile anlaşılır. İletim çerçevesinin ardından 32-bit uzunluğundaki eşzamanlama işareti çıkış geçerli sinyaliyle birlikte çıkış olarak verilir. Eşzamanlama işaretinin bitimi bir sayaç ile anlaşılır. Eşzamanlama işaretinden sonra tekrar giriş verisi geçerli sinyali takip edilmeye başlanır ve diğer iletim çerçevesi iletilmeye başlanır.

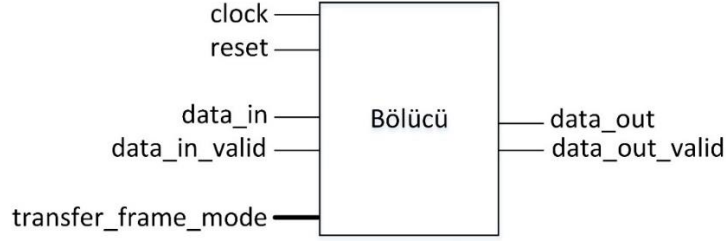
4.2 Kodlayıcı

4.2.1 Kodlayıcı bilgi blokları oluşturma

Geliştirilen uyarlamalı kodlama ve kiplenmeli kullanan verici yapısında iki farklı kodlayıcı türü birlikte kullanılmaktadır. Bölücü bloğundan alınan veriler seri birleştirilmiş evrişimsel kodlayıcı ve döngüsel blok kodlayıcı bloklarına paylaştırılır. Paylaştırma işleminde kullanılan seri birleştirilmiş evrişimsel kodlayıcı (SCCC) bilgi bloğu uzunluğu ve döngüsel blok kodlayıcı (RS) bilgi bloğu uzunluğu bit cinsinden 37 farklı ACM formatına göre Çizelge 4.5'te verilmiştir.

Kodlayıcı bilgi blokları oluşturma bloğu FPGA tasarımının arayüz diyagramı Şekil 4.4'te ve arayüz sinyallerinin açıklamaları Çizelge 4.6'da verilmiştir.

Kodlayıcı bilgi blokları oluşturma bloğunda başlangıçta sistem yeniden başlatma sinyali gelir ve tasarımda kullanılan iç sinyaller başlangıç değerleri alır. Sistem yeniden başlatma sinyalinden sonra bileşimli bir devre ile ACM formatına göre seri birleştirilmiş evrişimsel kodlayıcı ve döngüsel blok kodlayıcı için bilgi blok uzunlukları verisi Çizelge 4.5'te belirtilen şekilde seçilir.



Şekil 4.3 : Bölücü bloğu FPGA tasarımı arayüz sinyalleri.

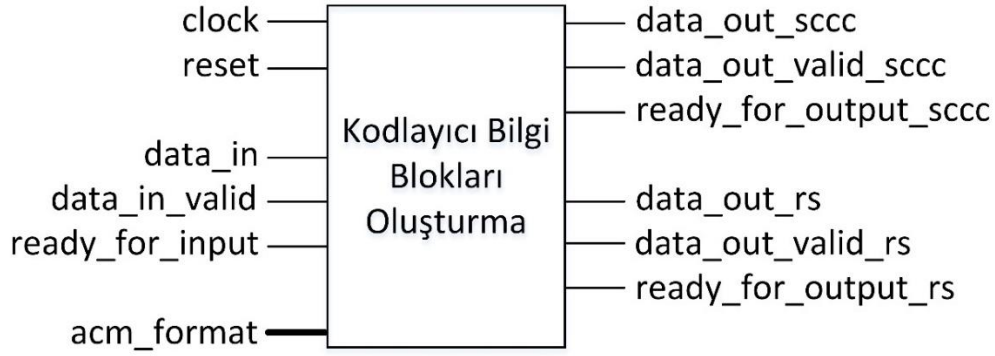
Çizelge 4.4 : Bölücü bloğu FPGA tasarımı arayüz sinyal bilgileri.

Sinyal	Giriş/ Çıkış	Bit Genişliği	Tanım
clock	giriş	1	sistem saat sinyali
reset	giriş	1	sistem yeniden başlatma
data_in	giriş	1	giriş verisi
data_in_valid	giriş	1	giriş verisi geçerli
transfer_frame_mode	giriş	2	iletim çerçevesi kipi
data_out	çıkış	1	çıkış verisi
data_out_valid	çıkış	1	çıkış verisi geçerli

Çizelge 4.5 : Kodlayıcılar için bilgi blok uzunlukları.

ACM Formatı	SCCC	RS	ACM Formatı	SCCC	RS
1	5758	0	20	30958	0
2	6958	0	21	33358	0
3	8398	0	22	35998	0
4	9838	0	23	33358	0
5	11278	0	24	35998	0
6	13198	0	25	38638	0
7	11278	0	26	41038	0
8	13198	0	27	43678	0
9	14878	0	28	16798	22680
10	17038	0	29	19438	22680
11	19198	0	30	22078	22680
12	21358	0	31	24718	22680
13	19198	0	32	27358	22680
14	21358	0	33	16558	30240
15	23518	0	34	19198	30240
16	25918	0	35	21838	30240
17	28318	0	36	24478	30240
18	25918	0	37	27358	30240
19	28318	0			

Kodlayıcı bilgi blokları oluşturma bloğunda sistem başlangıç durumuna getirildiğinde ve sistem veri girişine hazır olduğunda girişe hazır sinyali 1 yapılır.



Şekil 4.4 : Kodlayıcı bilgi blokları oluşturma bloğu FPGA tasarımı arayüz sinyalleri.

Çizelge 4.6 : Kodlayıcı bilgi blokları oluşturma bloğu FPGA tasarımı arayüz sinyal bilgileri.

Sinyal	Giriş/Çıkış	Bit Genişliği	Tanım
clock	giriş	1	sistem saat sinyali
reset	giriş	1	sistem yeniden başlatma
data_in	giriş	1	giriş verisi
data_in_valid	giriş	1	giriş verisi geçerli
ready_for_input	çıkış	1	girişe hazır
acm_format	giriş	6	ACM formatı
data_out_sccc	çıkış	1	SCCC için çıkış verisi
data_out_valid_sccc	çıkış	1	SCCC için çıkış verisi geçerli
ready_for_output_sccc	giriş	1	SCCC için çıkışa hazır
data_out_rs	çıkış	1	RS için çıkış verisi
data_out_valid_rs	çıkış	1	RS için çıkış verisi geçerli
ready_for_output_rs	giriş	1	RS için çıkışa hazır

Dışarıdan gelecek olan veri bu sinyale göre gönderilir. Girişe hazır sinyali 0 ise veri akışı önceki blok tarafından durdurulur.

Giriş verisi ve giriş verisi geçerli sinyaliyle birlikte alınır. Alınan veriler öncelikle eşzamanlamalı ilk-giren ilk-çıkış hafıza birimine yazılır. Öncelikle SCCC için çıkışa hazır sinyalinin 1 gelmesi durumunda, ilk-giren ilk-çıkış hafıza biriminden SCCC bilgi blok uzunluğu kadar bilgi okunur ve çıkış verisi geçerli sinyali 1 yapılarak gönderilir. Çıkışa hazır sinyalinin 0 gelmesi durumunda veri akışı kesilir ve sistem tekrar çıkışa hazır sinyali 1 gelene kadar sistem aynı durumda bekletilir. SCCC bilgi bloğu uzunluğu kadar veri okunduktan sonra RS kodlayıcı için çıkışa hazır sinyaline bakılır. RS kodlayıcı için çıkışa hazır sinyali gelmesi durumunda ilk-giren ilk-çıkış hafıza biriminden RS bilgi blok uzunluğu kadar bilgi okunur ve çıkış verisi geçerli

sinyali 1 yapılarak gönderilir. Çıkışa hazır sinyalinin 0 gelmesi durumunda veri akışı kesilir ve sistem tekrar çıkışa hazır sinyali 1 gelene kadar sistem aynı durumda bekletilir. RS bilgi bloğu uzunluğu kadar veri okunduktan sonra tekrar SCCC kodlayıcı için çıkışa hazır sinyaline bakılır ve sistem bu şekilde çalışmaya devam eder. Böylece ilk-giren ilk-çıkar hafıza birimine yazılan giriş verileri Çizelge 4.5'te verilen bilgi blok uzunluklarına göre kodlayıcılara paylaştırılır.

4.2.2 Seri birleştirilmiş evrişimsel kodlayıcı

Seri birleştirilmiş evrişimsel kodlayıcı bloğu, seri birleştirilmiş kodlama için gerekli olan blokları birleştirilen bir üstblok olarak tanımlanabilir.

Seri birleştirilmiş evrişimsel kodlayıcı bloğu FPGA tasarımının arayüz diyagramı Şekil 4.5'te ve arayüz sinyallerinin açıklamaları Çizelge 4.7'de verilmiştir.

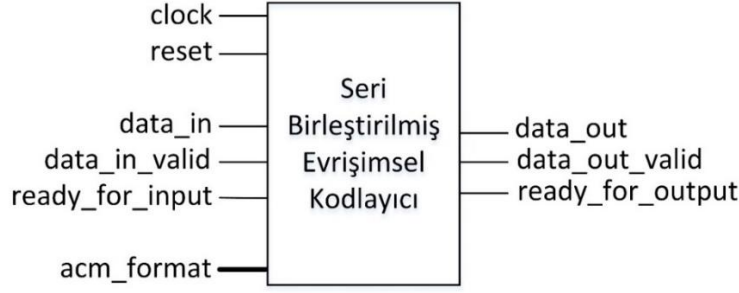
4.2.2.1 Dış evrişimsel kodlayıcı

Dış evrişimsel kodlayıcı bloğunda, geliştirilen uyarlamalı kodlama ve kipleme kullanan verici yapısındaki dış evrişimsel kodlayıcı yapısı gerçekleştirilmiştir. ACM formatına göre seçilen bilgi bloğu uzunluğu kadar bilgi evrişimsel kodlayıcıdan geçirilir ve bilgi bloğu uzunluğunun 2 fazlasının 2 katı kadar biti 2 ayrı kanaldan çıkarılır.

Dış evrişimsel kodlayıcı bloğu FPGA tasarımının arayüz diyagramı Şekil 4.6'da ve arayüz sinyallerinin açıklamaları Çizelge 4.8'de verilmiştir.

Dış evrişimsel kodlayıcı bloğunda başlangıçta sistem yeniden başlatma sinyali gelir ve tasarımda kullanılan iç sinyaller başlangıç değerleri alır. Sistem yeniden başlatma sinyalinden sonra bileşimli bir devre ile ACM formatına göre kodlayıcı bilgi bloğu uzunluğu Çizelge 4.5'te belirtilen şekilde seçilir.

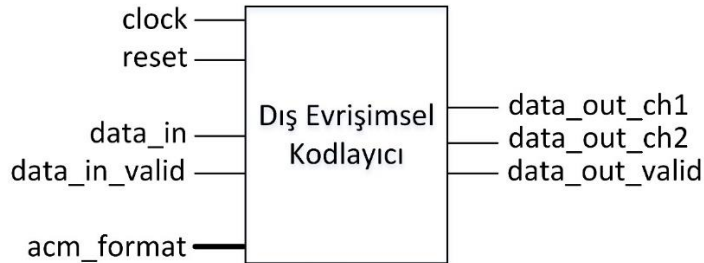
Dış evrişimsel kodlayıcı bloğunda sistem başlangıç durumuna getirildiğinde ve sistem veri girişine hazır olduğunda giriş verisi geçerli sinyalini takip etmeye başlanır. Giriş verisi geçerli sinyalinin 1 olduğu durumda giriş verisi alınır ve kodlayıcı yapısı için tanımlanan kayan yazmaçlara beslenir. Kayan yazmaçların çıkışları 1.kanal ve 2.kanal olarak ayrı ayrı çıkış verisi geçerli sinyaliyle birlikte çıkarılır. Seçilen bilgi bloğu uzunluğu tamamlandığında ilgili anahtarlar sonlandırma için gerekli konuma getirilir. Giriş verisi ve giriş verisi geçerli sinyalleri kontrol



Şekil 4.5 : Seri birleştirilmiş evrişimsel kodlayıcı bloğu FPGA tasarımı arayüz sinyalleri.

Çizelge 4.7 : Seri birleştirilmiş evrişimsel kodlayıcı bloğu FPGA tasarımı arayüz sinyal bilgileri.

Sinyal	Giriş/Çıkış	Bit Genişliği	Tanım
clock	giriş	1	sistem saat sinyali
reset	giriş	1	sistem yeniden başlatma
data_in	giriş	1	giriş verisi
data_in_valid	giriş	1	giriş verisi geçerli
ready_for_input	çıkış	1	girişe hazır
acm_format	giriş	6	ACM formatı
data_out	çıkış	1	çıkış verisi
data_out_valid	çıkış	1	çıkış verisi geçerli
ready_for_output	giriş	1	çıkışa hazır



Şekil 4.6 : Dış evrişimsel kodlayıcı bloğu FPGA tasarımı arayüz sinyalleri.

Çizelge 4.8 : Dış evrişimsel kodlayıcı bloğu FPGA tasarımı arayüz sinyal bilgileri.

Sinyal	Giriş/Çıkış	Bit Genişliği	Tanım
clock	giriş	1	sistem saat sinyali
reset	giriş	1	sistem yeniden başlatma
data_in	giriş	1	giriş verisi
data_in_valid	giriş	1	giriş verisi geçerli
acm_format	giriş	6	ACM format
data_out_ch1	çıkış	1	1.kanal çıkış verisi
data_out_ch2	çıkış	1	2.kanal çıkış verisi
data_out_valid	çıkış	1	çıkış verisi geçerli

edilmeden sonlandırma yapılır ve kodlanmış bitlerden sonra sonlandırma bitleri her iki kanaldan da çıkarılır. Sonlandırma ile kayan yazmaçların durumu başlangıç durumuna geldiği için tekrar giriş verisi geçerli sinyali takip etmeye başlanır ve sistem bu şekilde çalışmaya devam eder.

4.2.2.2 Sabit oranlı delme

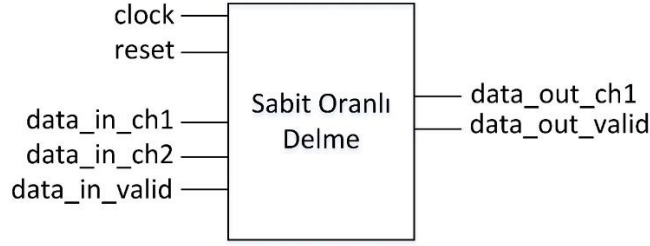
Sabit oranlı delme bloğunda iki ayrı kanaldan gelen veriler, delme matrisine göre silinerek seri bir şekilde çıkarılır.

Sabit oranlı delme bloğu FPGA tasarımının arayüz diyagramı Şekil 4.7’de ve arayüz sinyallerinin açıklamaları Çizelge 4.9’da verilmiştir.

Sabit oranlı delme bloğunda başlangıçta sistem yeniden başlatma sinyali gelir ve tasarımda kullanılan iç sinyaller başlangıç değerleri alır. Sistem başlangıç durumuna getirildiğinde ve sistem veri girişine hazır olduğunda giriş verisi geçerli sinyali takip etmeye başlanır. Giriş verisi geçerli sinyalinin 1 olduğu durumda her iki kanaldaki veri alınır. Delme matrisine göre 1.kanaldan gelen veri çıkış geçerli sinyaliyle birlikte çıkarılır. Alınan 2.kanal verisi ise bir yazmaçta depolanır. Bir sonraki saat işaretinde yazmaçta depolanan 2.kanal verisi geçerli sinyaliyle birlikte çıkarılır. Bir sonraki saat işaretinde giriş verisi geçerli sinyali takip etmeye başlanır. Giriş verisi geçerli sinyalinin 1 olduğu durumda her iki kanaldaki veri alınır. Delme matrisine göre 1.kanaldan gelen veri çıkış geçerli sinyaliyle birlikte çıkarılır. 2.kanaldan gelen veri silineceği için depolanmaz. Sistem başlangıç durumuna döner ve giriş verisi geçerli sinyali takip etmeye başlanır.

Bu bloktaki kısıt giriş verisi sıklığıdır. Paralel olarak 2 ayrı kanaldan alınan verinin seri olarak tek kanaldan çıkması için 2 saat periyoduna ihtiyaç vardır. Kullanılan delme matrisine göre bu durum her 2 giriş örneğinde 1 olmaktadır. Bu yüzden evrişimsel kodlayıcının bir saat sinyalinde çıktı verip, bir saat sinyalinde durup, sonraki iki saat sinyali boyunca çıkış verip, bir saat sinyalinde durup, sonraki iki saat sinyali boyunca çıkış vermesi ve bu şekilde devam etmesi gerekir. Bu zamanlama diyagramı Şekil 4.8’de verilmiştir.

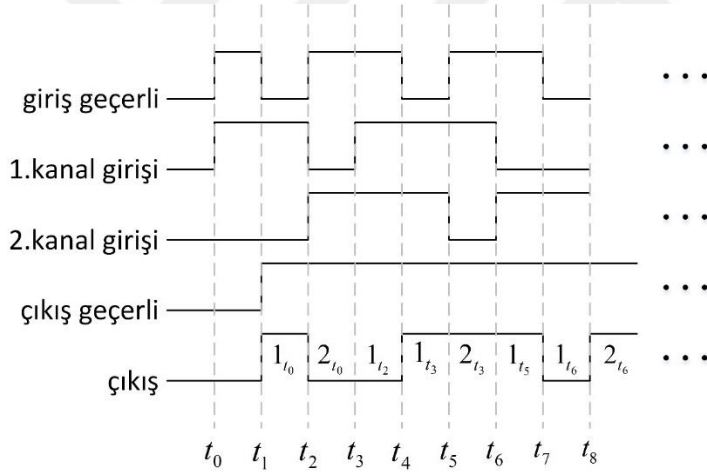
Sabit oranlı delme bloğunda t_0 anında 1 olan giriş geçerli sinyali sebebiyle 1 ve 2.kanal girişlerinin t_0 zamanındaki değerleri alınır. t_0 zamanında alınan 1.kanal verisi 1_{t_0} , t_1 anında çıkış geçerli sinyali 1 yapılarak çıkarılır. t_0 zamanında alınan



Şekil 4.7 : Sabit oranlı delme bloğu FPGA tasarımı arayüz sinyalleri.

Çizelge 4.9 : Sabit oranlı delme bloğu FPGA tasarımı arayüz sinyal bilgileri.

Sinyal	Giriş/ Çıkış	Bit Genişliği	Tanım
clock	giriş	1	sistem saat sinyali
reset	giriş	1	sistem yeniden başlatma
data_in_ch1	giriş	1	1.kanal giriş verisi
data_in_ch2	giriş	1	2.kanal giriş verisi
data_in_valid	giriş	1	giriş verisi geçerli
data_out	çıkış	1	çıkış verisi
data_out_valid	çıkış	1	çıkış verisi geçerli



Şekil 4.8 : Sabit oranlı delme bloğu zamanlama diyagramı.

2.kanal verisi 2_{t_0} , t_2 anında çıkış geçerli sinyali 1 yapılarak çıkılır. Bu sebeple t_1 zamanında yeni giriş verisi gelmemesi gerekir. Şekil 4.8’de verilen zamanlama diyagramı ile dış evrişimsel kodlayıcı bloğunun zamanlama diyagramında verildiği şekilde çıkış verdiğinde sabit oranlı delme bloğunda her saat sinyalinde çıkış verilmektedir.

Evrişimsel kodlayıcının bu sıklıkta çıkış vermesi kodlayıcı için bilgi blokları oluşturma bloğunda ayarlanmaktadır. Bu blokta hafızadan okunan ve dış evrişimsel kodlayıcıya gelen giriş verileri Şekil 4.8’de verilen zamanlama diyagramını sağlayacak sıklıkta okunmaktadır.

4.2.2.3 Rastgele serpiştirme

Rastgele serpiştirme bloęu belirli bir uzunluktaki veri setinin tanımlanan bir fonksiyon veya permütasyona göre karıştırılması işlemidir. Farklı ACM formatları için rastgele serpiştirme blok uzunluęu Çizelge 4.10'da verilmiştir.

Rastgele serpiştirme bloęu FPGA tasarımının arayüz diyagramı Şekil 4.9'da ve arayüz sinyallerinin açıklamaları Çizelge 4.11'de verilmiştir.

Rastgele serpiştirme bloęunda başlangıçta sistem yeniden başlatma sinyali gelir ve tasarımda kullanılan iç sinyaller başlangıç deęerleri alır. Sistem yeniden başlatma sinyalinden sonra bileşimli bir devre ile ACM formatına göre rastgele serpiştirme blok uzunluęu Çizelge 4.10'da belirtilen şekilde seçilir.

Rastgele serpiştirme bloęunda sistem başlangıç durumuna getirildiğinde ve sistem veri girişine hazır olduğunda girişe hazır sinyali 1 yapılır. Dışarıdan gelecek olan veri bu sinyale göre gönderilir. Girişe hazır sinyali 0 ise veri akışı önceki blok tarafından durdurulur.

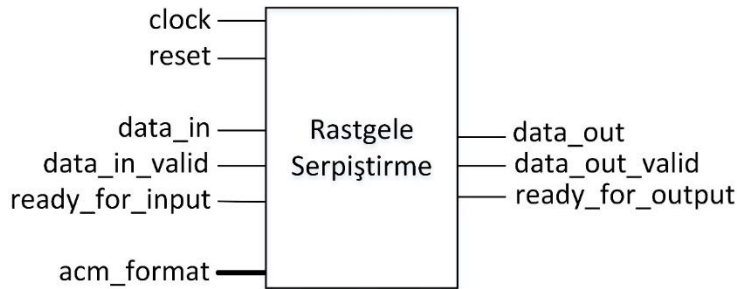
Giriş verisi ve giriş verisi geçerli sinyaliyle birlikte alınır. Alınan veriler sırayla en az rastgele serpiştirme bloęu uzunluęunda olan bir blok hafıza birimine yazılır. Rastgele serpiştirme bloęu uzunluęu kadar veri sırayla blok hafızaya yazılır. Blok uzunluęu bir sayaç ile kontrol edilir. Sayaç blok uzunluęuna ulaştığında girişe hazır sinyali 0 yapılır ve veri akışı durur. Daha sonra serpiştirme işlemi başlatılır. Serpiştirme bloęunda ilk okunacak verinin adresi hesaplanır. Hesaplama kullanılan α ve β deęişkenleri daha önce tanımlanmış bir taramalı tablo üzerinden okunur. 27 mod için kullanılan serpiştirme blok uzunluklarına göre α ve β deęerlerini gösteren tablo [5]'te, dięer modlar için kullanılan serpiştirme blok uzunluklarına göre α ve β deęerlerini gösteren tablo Ek 1'de verilmiştir. Hesaplamalar yapıldıktan sonra çıkışa hazır sinyalinin 1 olması durumunda, çıkış geçerli sinyaliyle birlikte çıkılır. Çıkışa hazır sinyali 1 olduęu sürece serpiştirme bloęu bitine kadar işlem devam eder. Bir serpiştirme bloęu tamamlandıktan sonra tekrar girişe hazır sinyali 1 yapılır ve dięer serpiştirme bloęu için giriş verisi geçerli sinyali takip edilmeye başlanır.

4.2.2.4 İç evrişimsel kodlayıcı

İç evrişimsel kodlayıcı bloęunda, dış evrişimsel kodlayıcı bloęundaki işlemlerin aynı uygulanmakta ancak bilgi blok uzunluęu farklı seçilmektedir. İç evrişimsel kodlayıcı

Çizelge 4.10 : Rastgele serpiştirici blok uzunluğu.

ACM Formatı	Rastgele Serpiştirici Blok Uzunluğu (I)	ACM Formatı	Rastgele Serpiştirici Blok Uzunluğu(I)
1	8640	20	46440
2	10440	21	50040
3	12600	22	54000
4	14760	23	50040
5	16920	24	54000
6	19800	25	57960
7	16920	26	61560
8	19800	27	65520
9	22320	28	25200
10	25560	29	29160
11	28800	30	33120
12	32040	31	37080
13	28800	32	41040
14	32040	33	24840
15	35280	34	28800
16	38880	35	32760
17	42480	36	36720
18	38880	37	41040
19	42480		



Şekil 4.9 : Rastgele serpiştirme bloğu FPGA tasarımı arayüz sinyalleri.

Çizelge 4.11 : Rastgele serpiştirme bloğu FPGA tasarımı arayüz sinyal bilgileri.

Sinyal	Giriş/Çıkış	Bit Genişliği	Tanım
clock	giriş	1	sistem saat sinyali
reset	giriş	1	sistem yeniden başlatma
data_in	giriş	1	giriş verisi
data_in_valid	giriş	1	giriş verisi geçerli
ready_for_input	çıkış	1	girişe hazır
acm_format	giriş	6	ACM formatı
data_out	çıkış	1	çıkış verisi
data_out_valid	çıkış	1	çıkış verisi geçerli
ready_for_output	giriş	1	çıkışa hazır

için ACM formatına göre bilgi blok uzunluğu Çizelge 4.12’de verilmiştir.

Çizelge 4.12 : İç evrişimsel kodlayıcı bilgi blok uzunluğu.

ACM Formatı	İç Evrişimsel Kodlayıcı Blok Uzunluğu	ACM Formatı	İç Evrişimsel Kodlayıcı Blok Uzunluğu
1	8640	20	46440
2	10440	21	50040
3	12600	22	54000
4	14760	23	50040
5	16920	24	54000
6	19800	25	57960
7	16920	26	61560
8	19800	27	65520
9	22320	28	25200
10	25560	29	29160
11	28800	30	33120
12	32040	31	37080
13	28800	32	41040
14	32040	33	24840
15	35280	34	28800
16	38880	35	32760
17	42480	36	36720
18	38880	37	41040
19	42480		

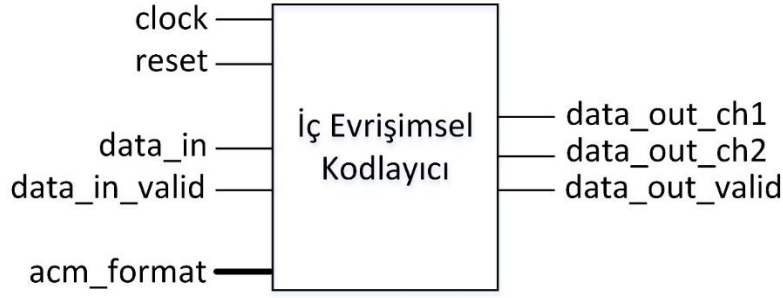
İç evrişimsel kodlayıcı bloğu FPGA tasarımının arayüz diyagramı Şekil 4.10'da ve arayüz sinyallerinin açıklamaları Çizelge 4.13'te verilmiştir.

4.2.2.5 Sistematik bitler için delme

Sistematik bitler için delme bloğunda, iç evrişimsel kodlayıcı bloğu 1.kanalından çıkan sistematik bitlere farklı oranlı bir delme işlemi uygulanır. Bu delme işleminde giriş verileri 300'er bitlik setler halinde alınır. Bu 300 bit içerisindeki bazı bitler silinir. ACM formatına göre silinecek bit sayısı ve bit sayılarına göre silinecek bit pozisyonları Çizelge 4.14 ve Çizelge 4.15'te verilmiştir.

Sistematik bitler için delme bloğu FPGA tasarımının arayüz diyagramı Şekil 4.11'de ve arayüz sinyallerinin açıklamaları Çizelge 4.16'da verilmiştir.

Sistematik bitler için delme bloğunda başlangıçta sistem yeniden başlatma sinyali gelir ve tasarımda kullanılan iç sinyaller başlangıç değerleri alır. Sistem yeniden başlatma sinyalinden sonra bileşimli bir devre ile ACM formatına göre giriş verisi blok uzunluğunu sonlandırma bitleri sebebiyle Çizelge 4.12'de verilen iç evrişimsel kodlayıcı bloğu bilgi blok uzunluğunun 2 fazlası olarak belirlenir.



Şekil 4.10 : İç evrişimsel kodlayıcı bloğu FPGA tasarımı arayüz sinyalleri.

Çizelge 4.13 : İç evrişimsel kodlayıcı bloğu FPGA tasarımı arayüz sinyal bilgileri.

Sinyal	Giriş/ Çıkış	Bit Genişliği	Tanım
clock	giriş	1	sistem saat sinyali
reset	giriş	1	sistem yeniden başlatma
data_in	giriş	1	giriş verisi
data_in_valid	giriş	1	giriş verisi geçerli
acm_format	giriş	6	ACM formatı
data_out_ch1	çıkış	1	1.kanal çıkış verisi
data_out_ch2	çıkış	1	2.kanal çıkış verisi
data_out_valid	çıkış	1	çıkış verisi geçerli

Çizelge 4.14 : Sistematik bitler için delme bloğu silinecek bit sayıları.

ACM Formatı	Silinecek Bit Sayısı	ACM Formatı	Silinecek Bit Sayısı
1	0	20	76
2	0	21	83
3	26	22	90
4	49	23	64
5	66	24	72
6	79	25	80
7	8	26	86
8	60	27	92
9	50	28	26
10	66	29	49
11	79	30	64
12	86	31	76
13	45	32	86
14	59	33	26
15	70	34	45
16	80	35	64
17	89	36	76
18	55	37	86
19	66		

Çizelge 4.15 : Sistematik bitler için delme bloğu silinecek bit pozisyonları.

Bit Sayısı	Bit Pozisyonu	Bit Sayısı	Bit Pozisyonu	Bit Sayısı	Bit Pozisyonu
1	76	32	207	63	84
2	1	33	259	64	229
3	145	34	292	65	193
4	214	35	232	66	283
5	256	36	67	67	262
6	37	37	280	68	25
7	109	38	247	69	238
8	181	39	147	70	60
9	277	40	30	71	201
10	235	41	111	72	294
11	55	42	183	73	132
12	127	43	6	74	96
13	163	44	48	75	159
14	19	45	93	76	34
15	199	46	165	77	265
16	91	47	129	78	114
17	289	48	219	79	177
18	244	49	195	80	225
19	64	50	270	81	79
20	268	51	72	82	12
21	223	52	15	83	151
22	136	53	297	84	51
23	172	54	211	85	274
24	28	55	138	86	204
25	100	56	102	87	105
26	190	57	174	88	4
27	10	58	39	89	241
28	46	59	250	90	169
29	118	60	57	91	69
30	154	61	120	92	124
31	81	62	156		



Şekil 4.11 : Sistematik bitler için delme bloğu FPGA tasarımı arayüz sinyalleri.

Çizelge 4.16 : Sistematik bitler için delme bloğu FPGA tasarımı arayüz sinyal bilgileri.

Sinyal	Giriş/ Çıkış	Bit Genişliği	Tanım
clock	giriş	1	sistem saat sinyali
reset	giriş	1	sistem yeniden başlatma
data_in	giriş	1	giriş verisi
data_in_valid	giriş	1	giriş verisi geçerli
acm_format	giriş	6	ACM formatı
data_out	çıkış	1	çıkış verisi
data_out_valid	çıkış	1	çıkış verisi geçerli

Sistematik bitler için delme bloğunda sistem başlangıç durumuna getirildiğinde ve sistem veri girişine hazır olduğunda giriş verisi geçerli sinyali takip edilir. Giriş verisi geçerli sinyaliyle birlikte giriş verisi alınır. Silinecek bit sayısı ve bit pozisyonları Çizelge 4.14 ve Çizelge 4.15'te verildiği şekilde bir taramalı tabloda tutulur. Her gelen bit için bulunduğu 300-bit uzunluğundaki setteki pozisyonu sayaç ile tutulur. Gelen bitin pozisyonu silinecek bit pozisyonları içerisinde aranır. Aranan bit pozisyonu silinecek pozisyonlarında ise çıkış verilmez. Aranan bit pozisyonu silinecek bit pozisyonlarında değilse alınan bit, çıkış geçerli sinyaliyle birlikte çıkarılır. Giriş verisi bloğundaki son iki sonlandırma bitlerine delme işlemi uygulanmaz. Her giriş verisi bloğu için 300-bitlik setler halinde bu işlem uygulanır.

4.2.2.6 Eşlik denetim bitleri için delme

Eşlik denetim bitleri için delme bloğunda, iç evrişimsel kodlayıcı bloğu 2.kanalından çıkan eşlik denetim bitlerine farklı oranlı bir delme işlemi uygulanır. Bu delme işleminde silinecek bit sayısına bağlı olan bir hız ayarlama algoritması uygulanır. Silinecek eşlik denetim bit sayısı Çizelge 4.17'de verilmiştir.

Eşlik denetim bitleri delme işlemi iç evrişimsel kodlayıcıdan gelen sonlandırma bitlerine uygulanmaz. Bu yüzden delme işlemi uygulanacak giriş verisi blok uzunluğu rastgele serpiştirici blok uzunluğu (I) ile aynıdır. Silinecek eşlik denetim bit sayısına (Δ) ve delme işlemi uygulanacak blok uzunluğuna (I) bağlı olarak uygulanan hız ayarlama algoritması işlem adımları şu şekilde tanımlanır:

1. $e = 1$ değişkeni tanımlanır.
2. Her bit pozisyonu için $i = 0, 1, \dots, I - 1$

Çizelge 4.17 : Eşlik denetim bitleri için delme bloğu silinecek bit sayıları.

ACM Formatı	Silinecek Bit Sayısı (Δ)	ACM Formatı	Silinecek Bit Sayısı (Δ)
1	1084	20	40619
2	4684	21	45739
3	7912	22	51304
4	10913	23	40808
5	13922	24	46444
6	17992	25	51869
7	9092	26	56877
8	11344	27	62351
9	16624	28	15820
10	21201	29	21161
11	25720	30	26778
12	30599	31	32371
13	20884	32	37919
14	25383	33	15131
15	29933	34	20884
16	34997	35	26135
17	39962	36	31742
18	30137	37	37919
19	35119		

- a. Eğer $e > 0$ ise i .bit iletilir, değilse $e = e + 1$ güncellemesi yapılır ve bit iletilmez.
- b. $e = e - \Delta$ güncellemesi yapılır ve devam eder.

Eşlik denetim bitleri için delme bloğu FPGA tasarımının arayüz diyagramı Şekil 4.12’de, arayüz sinyallerinin açıklamaları Çizelge 4.18’de verilmiştir.

Eşlik denetim bitleri için delme bloğunda başlangıçta sistem yeniden başlatma sinyali gelir ve tasarımda kullanılan iç sinyaller başlangıç değerleri alır. Sistem yeniden başlatma sinyalinden sonra bileşimli bir devre ile ACM formatına göre giriş verisi blok uzunluğunu sonlandırma bitleri sebebiyle Çizelge 4.12’de verilen iç evrişimsel kodlayıcı bloğu bilgi blok uzunluğunun 2 fazlası olarak belirlenir.

Silinecek eşlik denetim bit sayısı Çizelge 4.17’de verildiği şekilde bir taramalı tabloda tutulur. Sistem yeniden başlatma sinyalinden sonra bileşimli bir devre ile ACM formatına göre silinecek eşlik denetim bit sayısı taramalı tablodan okunur.

Eşlik denetim bitleri için delme bloğunda giriş verisi geçerli sinyali takip edilir. Giriş verisi geçerli sinyaliyle birlikte giriş verisi alınır. Belirlenen giriş verisi blok



Şekil 4.12 : Eşlik denetim bitleri için delme bloğu FPGA tasarımı arayüz sinyalleri.

Çizelge 4.18 : Eşlik denetim bitleri için delme bloğu FPGA tasarımı arayüz sinyal bilgileri.

Sinyal	Giriş/Çıkış	Bit Genişliği	Tanım
clock	giriş	1	sistem saat sinyali
reset	giriş	1	sistem yeniden başlatma
data_in	giriş	1	giriş verisi
data_in_valid	giriş	1	giriş verisi geçerli
acm_format	giriş	6	ACM formatı
data_out	çıkış	1	çıkış verisi
data_out_valid	çıkış	1	çıkış verisi geçerli

uzunluğu ve silinecek eşlik denetim bit sayısı kullanılarak hız ayarlama algoritması tasarlanan bir sonlu durum makinesi üzerinde işletilir. Çıkış sinyalleri çıkış geçerli sinyaliyle birlikte çıkarılır.

4.2.2.7 Satır-sütun serpiştirme

Satır-sütun serpiştirme bloğu belirli bir uzunluktaki veri setinin bir blok halinde karıştırılması işlemidir. Farklı ACM formatları için satır-sütun serpiştirme giriş verisi blok uzunluğu Çizelge 4.19’da verilmiştir.

Satır-sütun serpiştirme bloğu giriş verisi blok uzunluğu kipleme derecesi ve kiplemede 1 sembol ile gönderilen bit sayısına göre belirlenir. Her bilgi bloğunun uzunluğu kipleme sonrasında 8100 olacak şekilde sabitlenmiştir. Kiplemede seri birleştirilmiş evrişimsel kodlayıcı çıkışından m -bit 1 sembole eşlenecekse, bu ACM formatı için satır-sütun serpiştirme bloğu giriş verisi blok uzunluğu $m \times 8100$ olarak hesaplanır. ACM formatına göre değişen m değerine göre, satır-sütun serpiştirme işlemi için tamamlanan blok hafızanın boyutları $8100 \times m$ olarak ayarlanır.

Çizelge 4.19 : Satır-sütun serpiştirme giriş verisi blok uzunluğu.

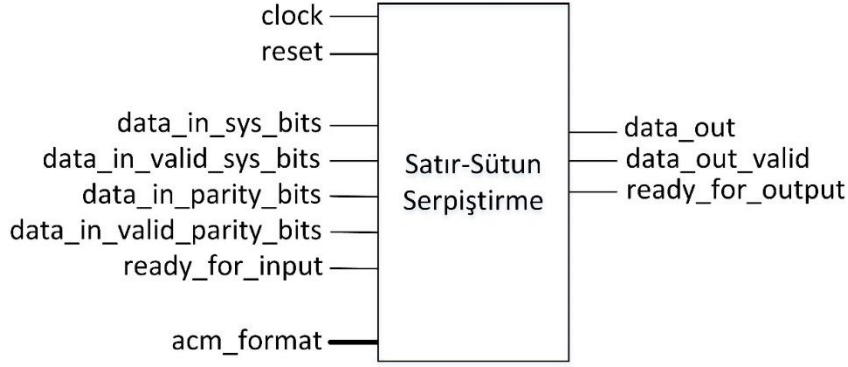
ACM Formatı	Blok Uzunluğu	ACM Formatı	Blok Uzunluğu
1	16200	20	40500
2	16200	21	40500
3	16200	22	40500
4	16200	23	48600
5	16200	24	48600
6	16200	25	48600
7	24300	26	48600
8	24300	27	48600
9	24300	28	32400
10	24300	29	32400
11	24300	30	32400
12	24300	31	32400
13	32400	32	32400
14	32400	33	32400
15	32400	34	32400
16	32400	35	32400
17	32400	36	32400
18	40500	37	32400
19	40500		

Giriş verisi blok uzunluğu, sistematik bitler için delme ve eşlik denetim bitleri için delme bloklarından gelen verilerin uzunluğunun toplamıdır. ACM formatına göre tanımlanan blok hafızaya önce sistematik bitler için delme bloğu çıkışları, devamında eşlik denetim bitleri için delme bloğu çıkışları yazılır. Yapılan okuma işlemi ile sistematik bitler ve eşlik denetim bitleri serpiştirilmiş olur.

Satır-sütun serpiştirme bloğu FPGA tasarımının arayüz diyagramı Şekil 4.13'te ve arayüz sinyallerinin açıklamaları Çizelge 4.20'de verilmiştir.

Satır-sütun serpiştirme bloğunda başlangıçta sistem yeniden başlatma sinyali gelir ve tasarımda kullanılan iç sinyaller başlangıç değerleri alır. Sistem yeniden başlatma sinyalinden sonra bileşimli bir devre ile ACM formatına göre sistematik bitler ve eşlik denetim bitleri için giriş verisi blok uzunluğu Çizelge 4.21'de belirtilen şekilde seçilir.

Satır-sütun serpiştirme bloğunda sistem başlangıç durumuna getirildiğinde ve sistem veri girişine hazır olduğunda girişe hazır sinyali 1 yapılır. Dışarıdan gelecek olan veri bu sinyale göre gönderilir. Girişe hazır sinyali 0 ise veri akışı önceki blok tarafından



Şekil 4.13 : Satır-sütun serpiştirme bloğu FPGA tasarımı arayüz sinyalleri.

Çizelge 4.20 : Satır-sütun serpiştirme bloğu FPGA tasarımı arayüz sinyal bilgileri.

Sinyal	Giriş/Çıkış	Bit Genişliği	Tanım
clock	giriş	1	sistem saat sinyali
reset	giriş	1	sistem yeniden başlatma
data_in_sys_bits	giriş	1	sistematik bit giriş verisi
data_in_valid_sys_bits	giriş	1	sistematik bit giriş verisi geçerli
data_in_parity_bits	giriş	1	eşlik denetim bit giriş verisi
data_in_valid_parity_bits	giriş	1	eşlik denetim bit giriş verisi geçerli
ready_for_input	çıkış	1	girişe hazır
acm_format	giriş	6	ACM formatı
data_out	çıkış	1	çıkış verisi
data_out_valid	çıkış	1	çıkış verisi geçerli
ready_for_output	giriş	1	çıkışa hazır

durdurulur.

Öncelikle sistematik bitler giriş verisi geçerli sinyali kontrol edilir. Giriş geçerli sinyaliyle birlikte blok uzunluğu boyunca sistematik bitler alınır ve uygun şekilde blok hafızaya yazılır. Bu sırada eşlik denetim bitleri de sinyal göndereceği için, eşlik denetim bitleri giriş sinyali giriş geçerli sinyaliyle birlikte bir ilk-giren ilk-çıkış hafıza birimine yazılır. Sistematik bitler blok uzunluğunun tamamlanmasından sonra eşlik denetim bitleri yazıldığı ilk-giren ilk-çıkış hafıza biriminden okunarak blok hafızaya uygun şekilde yazılır. Yazma işlemi tamamlandıktan sonra girişe hazır sinyali 0 yapılır ve serpiştirme işlemi başlatılır.

Çıkışa hazır sinyalinin 1 gelmesi durumunda okunacak bit adresleri hesaplanarak blok hafızadan okunur ve çıkış geçerli sinyaliyle birlikte çıkılır. Çıkışa hazır sinyalinin 0 gelmesi durumunda sistem durumunu korur ve çıkış vermez. Serpiştirme

Çizelge 4.21 : Satır-sütun serpiştirici için sistematik bit ve eşlik denetim bitleri giriş verisi blok uzunlukları.

ACM Formatı	Sistematik Bitler Blok Uzunluğu (S)	Eşlik Denetim Bitleri Blok Uzunluğu (P)	ACM Formatı	Sistematik Bitler Blok Uzunluğu (S)	Eşlik Denetim Bitleri Blok Uzunluğu (P)
1	8642	7558	20	34677	5823
2	10442	5758	21	36197	4303
3	11510	4690	22	37802	2698
4	12351	3849	23	39366	9234
5	13200	3000	24	41042	7558
6	14390	1810	25	42507	6093
7	16470	7830	26	43915	4685
8	15842	8458	27	45429	3171
9	18602	5698	28	23018	9382
10	19939	4361	29	24399	8001
11	21218	3082	30	26056	6344
12	22857	1443	31	27689	4711
13	24482	7918	32	29277	3123
14	25741	6659	33	22689	9711
15	27051	5349	34	24482	7918
16	28515	3885	35	25773	6627
17	29880	2520	36	27420	4980
18	31755	8745	37	29277	3123
19	33137	7363			

işleminin tamamlanmasının ardından girişe hazır işareti 1 yapılır ve tekrar sistematik bitler giriş verisi geçerli sinyali kontrol edilmeye başlanır. Bu işlem her bilgi bloğu için bu şekilde devam eder.

4.2.3 Döngüsel blok kodlayıcı

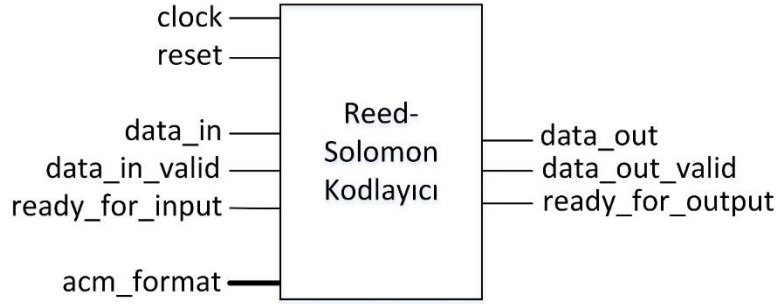
4.2.3.1 Reed-Solomon kodlayıcı

Reed-Solomon kodlayıcı bloğu, döngüsel blok kodlayıcı için gerekli olan blokları birleştirilen bir üstblok olarak tanımlanabilir.

Reed-Solomon kodlayıcı bloğu FPGA tasarımının arayüz diyagramı Şekil 4.14'te ve arayüz sinyallerinin açıklamaları Çizelge 4.22'de verilmiştir.

Sembol üretici

Sembol üretici bloğu temel olarak seri bitleri paralelleştirerek Reed-Solomon



Şekil 4.14 : Reed-Solomon kodlayıcı bloğu FPGA tasarımı arayüz sinyalleri.

Çizelge 4.22 : Reed-Solomon kodlayıcı bloğu FPGA tasarımı arayüz sinyal bilgileri.

Sinyal	Giriş/Çıkış	Bit Genişliği	Tanım
clock	giriş	1	sistem saat sinyali
reset	giriş	1	sistem yeniden başlatma
data_in	giriş	1	giriş verisi
data_in_valid	giriş	1	giriş verisi geçerli
ready_for_input	çıkış	1	girişe hazır
acm_format	giriş	6	ACM formatı
data_out	çıkış	1	çıkış verisi
data_out_valid	çıkış	1	çıkış verisi geçerli
ready_for_output	giriş	1	çıkışa hazır

kodlayıcının çalıştığı Galois cismi elemanları olan sembollere dönüştürür. Tasarımda kullanılan Reed-Solomon kodlayıcı $GF(2^7)$ Galois cismi üzerinde çalıştığı için, 7 biti bir sembole dönüştüren sembol üreticine ihtiyaç duyulmuştur.

Sembol üretici bloğu FPGA tasarımının arayüz diyagramı Şekil 4.15'te ve arayüz sinyallerinin açıklamaları Çizelge 4.23'te verilmiştir.

Sembol üretici bloğunda başlangıçta sistem yeniden başlatma sinyali gelir ve tasarımda kullanılan iç sinyaller başlangıç değerleri alır. Sistem yeniden başlatma sinyalinden sonra giriş verisi geçerli sinyali takip edilir. Giriş verisi geçerli sinyaliyle birlikte giriş verisi alınır ve 7-bit uzunluktaki yazmaç serisine yazılır. Bit uzunluğu sayaçlar ile kontrol edilir. Sonlu durum makineleri ile yapılan tasarımda 7 bit giriş verisi alındıktan sonra 7-bit uzunluğunda bir sembol çıkış verisi geçerli sinyaliyle birlikte bir sembol olarak Reed-Solomon kodlayıcıya gönderilir.



Şekil 4.15 : Sembol üretici bloğu FPGA tasarımı arayüz sinyalleri.

Çizelge 4.23 : Sembol üretici bloğu FPGA tasarımı arayüz sinyal bilgileri.

Sinyal	Giriş/ Çıkış	Bit Genişliği	Tanım
clock	giriş	1	sistem saat sinyali
reset	giriş	1	sistem yeniden başlatma
data_in	giriş	1	giriş verisi
data_in_valid	giriş	1	giriş verisi geçerli
data_out	çıkış	7	çıkış verisi
data_out_valid	çıkış	1	çıkış verisi geçerli

Reed-Solomon (127,120)

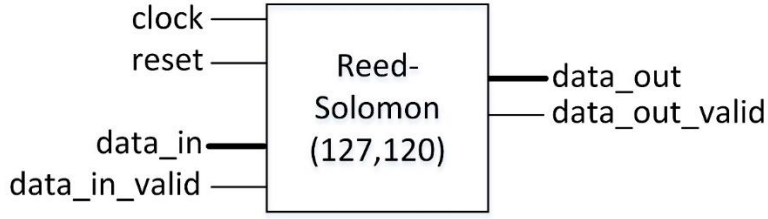
Reed-Solomon (127,120) bloğunda, alınan $GF(2^7)$ sembolleri Reed-Solomon kodlayıcı ile kodlanır. Çıkışlar yine $GF(2^7)$ sembolleri olarak çıkar.

Reed-Solomon (127,120) bloğu FPGA tasarımının arayüz diyagramı Şekil 4.16'da ve arayüz sinyallerinin açıklamaları Çizelge 4.24'te verilmiştir.

Reed-Solomon (127,120) bloğunda başlangıçta sistem yeniden başlatma sinyali gelir ve tasarımda kullanılan iç sinyaller başlangıç değerleri alır. Sistem yeniden başlatma sinyalinden sonra giriş verisi geçerli sinyali takip edilir. Giriş verisi geçerli sinyaliyle birlikte giriş verisi alınır. Alınan sembol kayan yazmaç yapısı ile geliştirilen kodlayıcıda giriş sembolü noktasından beslenir. Çıkış noktasından alınan kod sözcüğü sembolü çıkış verisi geçerli sinyaliyle birlikte çıkarılır.

Sistematik sembollerde alınan giriş sembolleri olduğu gibi çıkılırken, kayan yazmaç yapısı ile eşlik denetim sembolleri oluşturma işlemleri başlatılır. Kayan yazmaçlı yapıda belirtilen Galois cismi üzerinde çarpma ve Galois cismi üzerinde toplama işlemleri yapılarak kayan yazmaç değerleri güncellenir.

Galois cismi $GF(2^7)$ üzerindeki toplama işlemi, sembollere bitset XOR işlemi uygulanarak elde edilir. Galois cismi $GF(2^7)$ üzerindeki çarpma işlemi için ayrı bir



Şekil 4.16 : Reed-Solomon (127,120) bloğu FPGA tasarımı arayüz sinyalleri.

Çizelge 4.24 : Reed-Solomon (127,120) bloğu FPGA tasarımı arayüz sinyal bilgileri.

Sinyal	Giriş/Çıkış	Bit Genişliği	Tanım
clock	giriş	1	sistem saat sinyali
reset	giriş	1	sistem yeniden başlatma
data_in	giriş	7	giriş verisi
data_in_valid	giriş	1	giriş verisi geçerli
data_out	çıkış	7	çıkış verisi
data_out_valid	çıkış	1	çıkış verisi geçerli

blok tanımlanmıştır. Kodlayıcı yapısındaki her çarpma işlemi için bir Galois cismi çarpma bloğu kullanılarak aynı anda bütün çarpma işlemleri gerçekleştirilir.

Galois cismi çarpma

Galois cismi çarpma bloğu, alınan iki tane 7-bit uzunluğundaki $GF(2^7)$ sembollerini $GF(2^7)$ Galois cismi üzerinde çarpar ve çarpımı 7-bit uzunluğundaki $GF(2^7)$ sembolü olarak çıkar.

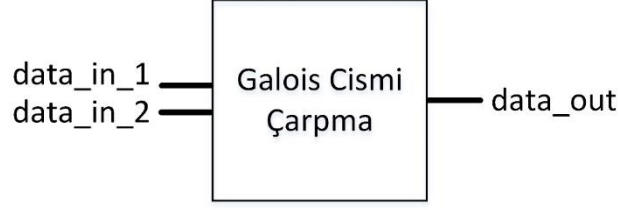
Galois cismi çarpma bloğu FPGA tasarımının arayüz diyagramı Şekil 4.17'de ve arayüz sinyallerinin açıklamaları Çizelge 4.25'te verilmiştir.

Galois cismi çarpma bloğu bileşimli devre olarak çalıştığı için saat sinyali ve yeniden başlatma sinyali kullanılmamıştır. Çıkış verisi bileşimli devre ile giriş verilerine göre anlık olarak üretilmektedir.

Galois cismi üzerindeki çarpma işlemi için öncelikle semboller 6.dereceden bir polinom olarak düşünülmüş ve ilk önce polinom çarpması işlemi tanımlanmıştır. Bu tanımlamada çarpılacak semboller (4.1) ve (4.2)'de verilmiştir.

$$s_1(x) = \alpha_6x^6 + \alpha_5x^5 + \alpha_4x^4 + \alpha_3x^3 + \alpha_2x^2 + \alpha_1x + \alpha_0 \quad (4.1)$$

$$s_2(x) = \beta_6x^6 + \beta_5x^5 + \beta_4x^4 + \beta_3x^3 + \beta_2x^2 + \beta_1x + \beta_0 \quad (4.2)$$



Şekil 4.17 : Galois cismi çarpma bloğu FPGA tasarımı arayüz sinyalleri.

Çizelge 4.25 : Galois cismi çarpma bloğu FPGA tasarımı arayüz sinyal bilgileri.

Sinyal	Giriş/ Çıkış	Bit Genişliği	Tanım
data_in_1	giriş	7	1.giriş verisi
data_in_2	giriş	7	2.giriş verisi
data_out	çıkış	7	çıkış verisi

Belirtilen 6 dereceli iki polinomun çarpımı ile elde edilen 12 dereceli polinom (4.3)'te verilmiştir.

$$\begin{aligned}
m_{GF}(x) &= s_1(x) \cdot s_2(x) \\
&= (\alpha_6\beta_6)x^{12} + (\alpha_5\beta_6 + \alpha_6\beta_5)x^{11} + (\alpha_4\beta_6 + \alpha_5\beta_5 + \alpha_6\beta_4)x^{10} + \\
&\quad (\alpha_3\beta_6 + \alpha_4\beta_5 + \alpha_5\beta_4 + \alpha_6\beta_3)x^9 + \\
&\quad (\alpha_2\beta_6 + \alpha_3\beta_5 + \alpha_4\beta_4 + \alpha_5\beta_3 + \alpha_6\beta_2)x^8 + \\
&\quad (\alpha_1\beta_6 + \alpha_2\beta_5 + \alpha_3\beta_4 + \alpha_4\beta_3 + \alpha_5\beta_2 + \alpha_6\beta_1)x^7 + \\
&\quad (\alpha_0\beta_6 + \alpha_1\beta_5 + \alpha_2\beta_4 + \alpha_3\beta_3 + \alpha_4\beta_2 + \alpha_5\beta_1 + \alpha_6\beta_0)x^6 + \\
&\quad (\alpha_0\beta_5 + \alpha_1\beta_4 + \alpha_2\beta_3 + \alpha_3\beta_2 + \alpha_4\beta_1 + \alpha_5\beta_0)x^5 + \\
&\quad (\alpha_0\beta_4 + \alpha_1\beta_3 + \alpha_2\beta_2 + \alpha_3\beta_1 + \alpha_4\beta_0)x^4 + \\
&\quad (\alpha_0\beta_3 + \alpha_1\beta_2 + \alpha_2\beta_1 + \alpha_3\beta_0)x^3 + (\alpha_0\beta_2 + \alpha_1\beta_1 + \alpha_2\beta_0)x^2 + \\
&\quad (\alpha_0\beta_1 + \alpha_1\beta_0)x + \alpha_0\beta_0
\end{aligned} \tag{4.3}$$

Yüksek dereceli bir polinomun $GF(2^7)$ elemanı olarak gösterimi, bu Galois cisminin ilkel polinomu $p(x)$ 'e bölümü ile elde edilir. Bölme işlemi Öklid'in bölme algoritmasına göre yapılır. Örnek olarak, 12 dereceli bir polinom olan $l(x)$, 7 dereceli bir polinom olan Galois cisminin ilkel polinomu $p(x)$, $l(x)$ polinomunun $p(x)$ polinomuna bölümünün sonuç polinomu $l'(x)$ ve bölme işleminin detayları sırasıyla (4.4), (4.5), (4.6) ve (4.7)'de verilmiştir.

$$l(x) = \lambda_{12}x^{12} + \lambda_{11}x^{11} + \lambda_{10}x^{10} + \lambda_9x^9 + \lambda_8x^8 + \lambda_7x^7 + \lambda_6x^6 + \lambda_5x^5 + \lambda_4x^4 + \lambda_3x^3 + \lambda_2x^2 + \lambda_1x + \lambda_0 \quad (4.4)$$

$$p(x) = x^7 + x^3 + 1 \quad (4.5)$$

$$l'(x) = \gamma_6x^6 + \gamma_5x^5 + \gamma_4x^4 + \gamma_3x^3 + \gamma_2x^2 + \gamma_1x + \gamma_0 \quad (4.6)$$

$$\begin{aligned} l'(x) &= l(x) \div p(x) \\ &= (\lambda_6 + \lambda_{10})x^6 + (\lambda_5 + \lambda_9 + \lambda_{12})x^5 + \\ &\quad (\lambda_4 + \lambda_8 + \lambda_{11} + \lambda_{12})x^4 + (\lambda_3 + \lambda_7 + \lambda_{10} + \lambda_{11})x^3 + \\ &\quad (\lambda_2 + \lambda_9)x^2 + (\lambda_1 + \lambda_8 + \lambda_{12})x + (\lambda_0 + \lambda_7 + \lambda_{11}) \end{aligned} \quad (4.7)$$

Öklid'in bölme algoritmasına göre yapılan polinom bölmesi, (4.3)'te verilen $m(x)$ ve $p(x)$ polinomları için uygulanır. Bölüm polinomu katsayıları γ_i , λ_i cinsinden (4.7)'de verilmiştir. Bölünen polinom katsayıları λ_i , α_i ve β_i cinsinden (4.3)'te verilmiştir. Bu eşitlikler birleştirildiğinde $m(x)$ polinomunun $p(x)$ polinomuna bölümü $l'(x)$ polinomu katsayıları γ_i değerleri (4.8)'de verilmiştir.

$$\begin{aligned} \gamma_6 &= \lambda_6 + \lambda_{10} \\ &= \alpha_0\beta_6 + \alpha_1\beta_5 + \alpha_2\beta_4 + \alpha_3\beta_3 + \alpha_4\beta_2 + \alpha_5\beta_1 + \alpha_6\beta_0 + \alpha_4\beta_6 + \alpha_5\beta_5 + \\ &\quad \alpha_6\beta_4 \end{aligned}$$

$$\begin{aligned} \gamma_5 &= \lambda_5 + \lambda_9 + \lambda_{12} \\ &= \alpha_0\beta_5 + \alpha_1\beta_4 + \alpha_2\beta_3 + \alpha_3\beta_2 + \alpha_4\beta_1 + \alpha_5\beta_0 + \alpha_3\beta_6 + \alpha_4\beta_5 + \alpha_5\beta_4 + \\ &\quad \alpha_6\beta_3 + \alpha_6\beta_6 \end{aligned}$$

$$\begin{aligned} \gamma_4 &= \lambda_4 + \lambda_8 + \lambda_{11} + \lambda_{12} \\ &= \alpha_0\beta_4 + \alpha_1\beta_3 + \alpha_2\beta_2 + \alpha_3\beta_3 + \alpha_4\beta_0 + \alpha_2\beta_6 + \alpha_3\beta_5 + \alpha_4\beta_4 + \alpha_5\beta_3 + \\ &\quad \alpha_6\beta_2 + \alpha_5\beta_6 + \alpha_6\beta_5 + \alpha_6\beta_6 \end{aligned}$$

$$\begin{aligned} \gamma_3 &= \lambda_3 + \lambda_7 + \lambda_{10} + \lambda_{11} \\ &= \alpha_0\beta_3 + \alpha_1\beta_2 + \alpha_2\beta_1 + \alpha_3\beta_0 + \alpha_1\beta_6 + \alpha_2\beta_5 + \alpha_3\beta_4 + \alpha_4\beta_3 + \alpha_5\beta_2 + \\ &\quad \alpha_6\beta_1 + \alpha_4\beta_6 + \alpha_5\beta_5 + \alpha_6\beta_4 + \alpha_5\beta_6 + \alpha_6\beta_5 \end{aligned}$$

$$\gamma_2 = \lambda_2 + \lambda_9$$

$$\begin{aligned}
&= \alpha_0\beta_2 + \alpha_1\beta_1 + \alpha_2\beta_0 + \alpha_3\beta_6 + \alpha_4\beta_5 + \alpha_5\beta_4 + \alpha_6\beta_3 \\
\gamma_1 &= \lambda_1 + \lambda_8 + \lambda_{12} \\
&= \alpha_0\beta_1 + \alpha_1\beta_0 + \alpha_2\beta_6 + \alpha_3\beta_5 + \alpha_4\beta_4 + \alpha_5\beta_3 + \alpha_6\beta_2 + \alpha_6\beta_6 \\
\gamma_0 &= \lambda_0 + \lambda_7 + \lambda_{11} \\
&= \alpha_0\beta_0 + \alpha_1\beta_6 + \alpha_2\beta_5 + \alpha_3\beta_4 + \alpha_4\beta_3 + \alpha_5\beta_2 + \alpha_6\beta_1 + \alpha_5\beta_6 + \\
&\quad \alpha_6\beta_5
\end{aligned} \tag{4.8}$$

(4.1) ve (4.2)'de verilen 6 dereceli iki polinomun ilkel polinomu $p(x)$ olan $GF(2^7)$ üzerindeki çarpımı sonucu elde edilen 6 dereceli polinomun katsayıları (4.8)'deki gibi bulunur. Polinom katsayıları α_i , β_i , ve γ_i bitlerle gösterildiği için, (4.8)'de verilen işlemler bitset XOR ve AND kapıları ile gerçekleştirilir. (4.8)'de görülen toplama işlemi XOR kapısı ile, çarpma işlemleri AND kapısı ile gerçekleştirilir. Böylece iki sembolün $GF(2^7)$ üzerindeki çarpımının sonucu (4.8)'de verilen XOR ve AND bağlantıları ile elde edilir.

Bit üretici

Bit üretici bloğu temel olarak Reed-Solomon kodlayıcının çalıştığı Galois cismi elemanları olan sembolleri seri hale getirerek olan bitlere dönüştürür. Tasarımda kullanılan Reed-Solomon kodlayıcı $GF(2^7)$ Galois cismi üzerinde çalıştığı için, 7 bitlik sembolleri bitlere dönüştüren bit üreticine ihtiyaç duyulmuştur.

Bit üretici bloğu FPGA tasarımının arayüz diyagramı Şekil 4.18'de ve arayüz sinyallerinin açıklamaları Çizelge 4.26'da verilmiştir.

Sembol üretici bloğunda başlangıçta sistem yeniden başlatma sinyali gelir ve tasarımda kullanılan iç sinyaller başlangıç değerleri alır. Sistem yeniden başlatma sinyalinden sonra giriş verisi geçerli sinyali takip edilir. Giriş verisi geçerli sinyaliyle birlikte giriş verisi alınır ve 7-bit uzunluktaki yazmaç serisine yazılır. Giriş verisinin her biti sırayla çıkış geçerli sinyaliyle birlikte seri olarak çıkarılır. Bu bloğa 7 saat sinyalinde bir giriş verisi gelir ve her saat sinyalinde çıkış verisi üretilir.

Kod sözcüğü doldurma

Reed-Solomon kodlayıcı çıkışında oluşturulan kod sözcüğü uzunluğunun kiplemede kullanılan bit uzunluğuna göre 8100 sembol oluşturacak uzunlukta olmak



Şekil 4.18 : Bit üretici bloğu FPGA tasarımı arayüz sinyalleri.

Çizelge 4.26 : Bit üretici bloğu FPGA tasarımı arayüz sinyal bilgileri.

Sinyal	Giriş/ Çıkış	Bit Genişliği	Tanım
clock	giriş	1	sistem saat sinyali
reset	giriş	1	sistem yeniden başlatma
data_in	giriş	7	giriş verisi
data_in_valid	giriş	1	giriş verisi geçerli
data_out	çıkış	1	çıkış verisi
data_out_valid	çıkış	1	çıkış verisi geçerli

zorundadır. ACM formatına göre 8100 sembolün kiplemesinde kullanılacak Reed-Solomon kodlayıcı çıkışı kod sözcüğü olması gereken blok uzunluğu Çizelge 4.27’de verilmiştir.

Reed-Solomon kodlayıcı blok kodlamasında tam olarak Çizelge 4.27’de verilen değerler tam olarak sağlanamamaktadır. Kodlayıcının bu değerlere en yakın çıkış kod sözcüğü blok uzunluklarını sağlayacak kodlayıcı seçilmiştir. Seçilen kodlayıcı ile giriş bilgi blok uzunluğuna göre elde edilen çıkış kod sözcüğü blok uzunlukları Çizelge 4.28’de verilmiştir.

Reed-Solomon kodlayıcı kod sözcüğü blok uzunluğun Çizelge 4.27’de verilen tablodaki değerlere ulaşması için kod sözcüğü bloklarının sonuna doldurma bitleri eklenir. Çizelge 4.28’de uzunluğu verilen kod sözcüğü bloklarının sonuna doldurma biti olarak 0 eklenerek Çizelge 4.27’de verilen değerler elde edilir.

Kod sözcüğü doldurma bloğu FPGA tasarımının arayüz diyagramı Şekil 4.19’da ve arayüz sinyallerinin açıklamaları Çizelge 4.29’da verilmiştir.

Kod sözcüğü doldurma bloğunda başlangıçta sistem yeniden başlatma sinyali gelir ve tasarımda kullanılan iç sinyaller başlangıç değerleri alır. Sistem yeniden başlatma sinyalinden sonra ACM formatına göre bileşimli bir devre ile doldurulacak bit uzunluğu seçilir. Giriş verisi geçerli sinyali takip edilir. Giriş verisi geçerli sinyaliyle

Çizelge 4.27 : Reed-Solomon kodlayıcı kod sözcüğü olması gereken blok uzunluğu.

ACM Formatı	Blok Uzunluğu	ACM Formatı	Blok Uzunluğu
1	0	20	0
2	0	21	0
3	0	22	0
4	0	23	0
5	0	24	0
6	0	25	0
7	0	26	0
8	0	27	0
9	0	28	24300
10	0	29	24300
11	0	30	24300
12	0	31	24300
13	0	32	24300
14	0	33	32400
15	0	34	32400
16	0	35	32400
17	0	36	32400
18	0	37	32400
19	0		

Çizelge 4.28 : Reed-Solomon kodlayıcı kod sözcüğü blok uzunluğu.

ACM Formatı	Blok Uzunluğu	ACM Formatı	Blok Uzunluğu
1	0	20	0
2	0	21	0
3	0	22	0
4	0	23	0
5	0	24	0
6	0	25	0
7	0	26	0
8	0	27	0
9	0	28	24003
10	0	29	24003
11	0	30	24003
12	0	31	24003
13	0	32	24003
14	0	33	32004
15	0	34	32004
16	0	35	32004
17	0	36	32004
18	0	37	32004
19	0		



Şekil 4.19 : Kod sözcüğü doldurma bloğu FPGA tasarımı arayüz sinyalleri.

Çizelge 4.29 : Kod sözcüğü doldurma bloğu FPGA tasarımı arayüz sinyal bilgileri.

Sinyal	Giriş/ Çıkış	Bit Genişliği	Tanım
clock	giriş	1	sistem saat sinyali
reset	giriş	1	sistem yeniden başlatma
data_in	giriş	1	giriş verisi
data_in_valid	giriş	1	giriş verisi geçerli
data_out	çıkış	1	çıkış verisi
data_out_valid	çıkış	1	çıkış verisi geçerli

birlikte giriş verisi alınır. Alınan giriş verileri çıkış geçerli sinyaliyle birlikte çıkarılır.

Veri sayısı sayaç ile tutulur. Sayaç kod sözcüğü bloğu uzunluğuna geldiğinde doldurma biti yeterince uzunlukta 0 biti serisi eklenir. Daha sonra yeni kod sözcüğü bloğu için giriş verisi geçerli sinyali takip edilir ve sistem bu şekilde çalışmaya devam eder.

4.2.3.2 RS satır-sütun serpiştirme

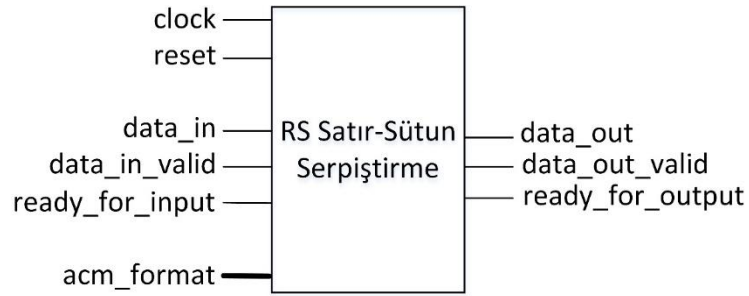
RS satır-sütun serpiştirme bloğu seri birleştirilmiş evrişimsel kodlayıcıda kullanılan satır-sütun bloğu ile aynı şekilde çalışmaktadır. Giriş arayüzü ve serpiştirme blok uzunluğu farklıdır. ACM formatlarına göre RS satır-sütun serpiştirme giriş verisi blok uzunluğu Çizelge 4.30'da verilmiştir.

Reed-Solomon kodlayıcı kod sözcüğü bloklarından bir sembolün kiplemesinde m_{rs} bitleri kullanıldığında, RS satır-sütun serpiştiricide $8100 \times m_{rs}$ boyutlarında blok hafıza kullanılır. Bu hafıza boyutlarına göre seri birleştirilmiş evrişimsel kodlayıcıdaki satır-sütun serpiştirme işlemiyle aynı işlemler bu blokta da gerçekleştirilir.

RS satır-sütun serpiştirme bloğu FPGA tasarımının arayüz diyagramı Şekil 4.20'de ve arayüz sinyallerinin açıklamaları Çizelge 4.31'de verilmiştir.

Çizelge 4.30 : RS satır-sütun serpiştirme giriş verisi blok uzunluğu.

ACM Formatı	Blok Uzunluğu	ACM Formatı	Blok Uzunluğu
1	0	20	0
2	0	21	0
3	0	22	0
4	0	23	0
5	0	24	0
6	0	25	0
7	0	26	0
8	0	27	0
9	0	28	24300
10	0	29	24300
11	0	30	24300
12	0	31	24300
13	0	32	24300
14	0	33	32400
15	0	34	32400
16	0	35	32400
17	0	36	32400
18	0	37	32400
19	0		



Şekil 4.20 : RS satır sütun serpiştirme bloğu FPGA tasarımı arayüz sinyalleri.

Çizelge 4.31 : RS satır-sütun serpiştirme bloğu FPGA tasarımı arayüz sinyal bilgileri.

Sinyal	Giriş/Çıkış	Bit Genişliği	Tanım
clock	giriş	1	sistem saat sinyali
reset	giriş	1	sistem yeniden başlatma
data_in	giriş	1	sistematik bit giriş verisi
data_in_valid	giriş	1	sistematik bit giriş verisi geçerli
ready_for_input	çıkış	1	girişe hazır
acm_format	giriş	6	ACM formatı
data_out	çıkış	1	çıkış verisi
data_out_valid	çıkış	1	çıkış verisi geçerli
ready_for_output	giriş	1	çıkışa hazır

4.2.4 Kod sözcüğü bloklarını birleştirme

Geliştirilen uyarlamalı kodlama ve kipleme kullanan verici yapısında kodlayıcı olarak seri birleştirilmiş evrişimsel kodlayıcı ve döngüsel blok kodlayıcı birlikte kullanılmaktadır. Kodlayıcı bilgi blokları oluşturma bloğu kodlayıcılar için uygun bilgi bloklarını kodlayıcılara gönderir. Kodlayıcı çıkışlarındaki kod sözcüğü blokları da kullanılan kiplemeye göre birleştirilir.

ACM formatına göre kipleme işleminde 1 sembole eşlenecek bit sayısı Çizelge 4.32'de verilmiştir.

Her bir sembol kiplemesinde Çizelge 4.32'de verilen bit sayısı seri birleştirilmiş evrişimsel kodlayıcı ve Reed-Solomon kodlayıcı kod sözcüğü bloklarının birleştirilmesiyle oluşturulur. Bir sembole eşlenecek bit sayısı (m), seri evrişimsel kodlayıcıdan (m_{SCCC}) bit ve döngüsel blok kodlayıcıdan (m_{RS}) bit alınarak oluşturulur. ACM formatına göre m_{SCCC} ve m_{RS} değerleri Çizelge 4.33'te verilmiştir.

Seri birleştirilmiş evrişimsel kodlayıcı ve döngüsel blok kodlayıcıda bulunan satır-sütun serpiştiricilerdeki m değeri her iki kodlayıcı için farklı olarak Çizelge 4.33'te verilen değerler iletilir.

Kod sözcüğü blokları birleştirme bloğunda, seri birleştirilmiş evrişimsel kodlayıcı kod sözcüğü bloklarından m_{SCCC} ve döngüsel blok kodlayıcı kod sözcüğü bloklarından m_{RS} bit alınarak m bit uzunluğunu tamamlayarak bir sembole eşlenecek bit sayısı tamamlanmış olur.

Kod sözcüğü blokları birleştirme bloğu FPGA tasarımının arayüz diyagramı Şekil 4.21'de ve arayüz sinyallerinin açıklamaları Çizelge 4.34'te verilmiştir.

Kod sözcüğü blokları birleştirme bloğunda başlangıçta sistem yeniden başlatma sinyali gelir ve tasarımda kullanılan iç sinyaller başlangıç değerleri alır. Sistem yeniden başlatma sinyalinden sonra bileşimli bir devre ile ACM formatına göre m_{SCCC} ve m_{RS} değerleri Çizelge 4.33'te belirtilen şekilde seçilir.

Sistem başlangıç durumuna getirildiğinde ve sistem veri girişine hazır olduğunda girişe hazır sinyali 1 yapılır. Dışarıdan gelecek olan veri bu sinyale göre gönderilir. Girişe hazır sinyali 0 ise veri akışı önceki blok tarafından durdurulur.

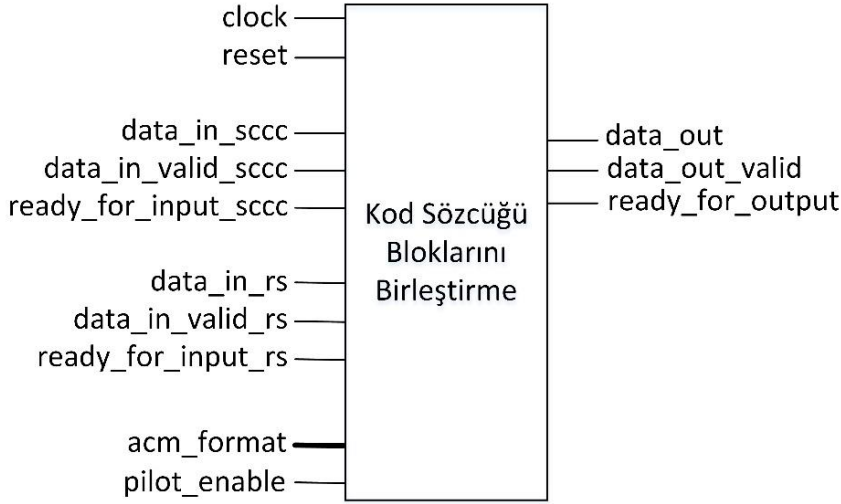
Bu blokta seri birleştirilmiş evrişimsel kodlayıcı ve döngüsel blok kodlayıcı için ayrı girişe hazır sinyalleri tanımlanmıştır. Başlangıçta her iki girişe hazır sinyali 1

Çizelge 4.32 : Kiplerede 1 sembole eşlenecek bit sayısı.

ACM Formatı	Bit Sayısı (m)	ACM Formatı	Bit Sayısı (m)
1	2	20	5
2	2	21	5
3	2	22	5
4	2	23	6
5	2	24	6
6	2	25	6
7	3	26	6
8	3	27	6
9	3	28	7
10	3	29	7
11	3	30	7
12	3	31	7
13	4	32	7
14	4	33	8
15	4	34	8
16	4	35	8
17	4	36	8
18	5	37	8
19	5		

Çizelge 4.33 : Kiplerede bir sembol için seri birleştirilmiş evrişimsel kodlayıcı ve döngüsel blok kodlayıcıdan alınacak bit sayıları.

ACM Formatı	m_{SCCC}	m_{RS}	ACM Formatı	m_{SCCC}	m_{RS}
1	2	0	20	5	0
2	2	0	21	5	0
3	2	0	22	5	0
4	2	0	23	6	0
5	2	0	24	6	0
6	2	0	25	6	0
7	3	0	26	6	0
8	3	0	27	6	0
9	3	0	28	4	3
10	3	0	29	4	3
11	3	0	30	4	3
12	3	0	31	4	3
13	4	0	32	4	3
14	4	0	33	4	4
15	4	0	34	4	4
16	4	0	35	4	4
17	4	0	36	4	4
18	5	0	37	4	4
19	5	0			



Şekil 4.21 : Kod sözcüğü bloklarını birleştirme bloğu FPGA tasarımı arayüz sinyalleri.

Çizelge 4.34 : Kod sözcüğü bloklarını birleştirme bloğu FPGA tasarımı arayüz sinyal bilgileri.

Sinyal	Giriş/Çıkış	Bit Genişliği	Tanım
clock	giriş	1	sistem saat sinyali
reset	giriş	1	sistem yeniden başlatma
data_in_sccc	giriş	1	SCCC giriş verisi
data_in_valid_sccc	giriş	1	SCCC giriş verisi geçerli
ready_for_input_sccc	çıkış	1	SCCC girişe hazır
data_in_rs	giriş	1	RS giriş verisi
data_in_valid_rs	giriş	1	RS giriş verisi geçerli
ready_for_input_rs	çıkış	1	RS girişe hazır
acm_format	giriş	6	ACM formatı
pilot_enable	giriş	1	pilot işareti ekleme bilgisi
data_out	çıkış	1	çıkış verisi
data_out_valid	çıkış	1	çıkış verisi geçerli
ready_for_output	giriş	1	çıkışa hazır

yapılır. Her iki kodlayıcının veri girişinde ilk-giren ilk-çıkır hafıza birimi bulunmaktadır. Giriş verisi geçerli sinyaliyle birlikte alınan giriş verileri hafızaya yazılır. Çıkışa hazır sinyalinin 1 gelmesi ile seri birleştirilmiş evrimsel kodlayıcı ilk-giren ilk-çıkır hafızasından m_{SCCC} ve döngüsel blok kodlayıcı ilk-giren ilk-çıkır hafızasından m_{RS} bit okunur ve birleştirilerek m -bit uzunluğunda semboller kiplenmek üzere kipleme bloğuna iletilir. Bu şekilde kod sözcüğü blokları birleştirilerek kipleme bloğuna iletilir.

İlk-giren ilk-çıkar hafıza biriminden okunan verilerin sıklığı pilot işareti eklenme bilgisine göre değişir. Pilot işareti eklenmesi durumunda 540 sembole eşlenecek kadar bit okunduktan sonra 16 sembol süresi kadar beklenir, daha sonra 540 sembole eşlenecek kadar bit okunur. Pilot işareti eklenmemesi durumunda hafızadan veriler sürekli olarak okunur.

Giriş hazır sinyalleri ilk-giren ilk-çıkar hafıza birimleri doluluk oranıyla kontrol edilir. İlgili kodlayıcıya ait hafıza biriminin dolması halinde o kodlayıcıya ait giriş hazır sinyali 0 olarak gönderilir ve veri akışı durdurulur. Hafızada yer açılması halinde giriş hazır sinyali tekrar 1 olarak gönderilir ve işlem devam eder.

4.3 Fiziksel Katman Çerçeveleme

4.3.1 Kipleme

Geliştirilen uyarlamalı kodlama ve kipleme kullanan verici yapısında kullanılan kipler ACM formatına göre Çizelge 4.35'te verildiği gibi seçilmektedir.

Kullanılan APSK kiplerinde halkaların yarıçapları oranları ACM formatına göre Çizelge 4.36'da verildiği gibi seçilmektedir.

Kipleme çıkışlarında eş fazlı ve dördün bileşenler kayan nokta sayı formatındadır. FPGA tasarımında sabit-nokta sayı formatında çalıştığı için birebir doğrulamaların yapılabilmesi için MATLAB modeli eş fazlı ve dördün bileşenleri sabit-nokta sayı formatına geçirilmiştir.

Sabit-nokta formatında sayıların tamsayı ve ondalık kısımları farklı bit uzunlukları ile ifade edilebilir. Tamsayı bit uzunluğu v ve ondalık bit uzunluğu w olarak seçildiğinde sabit-nokta formatı $Q(v.w)$ ifadesi ile gösterilebilir. Sabit-nokta formatı ve kayan nokta formatı arasındaki ilişki için sabit-nokta $Q(v.w)$ formatındaki iki sayı Şekil 4.22'de, bu sayının kayan nokta formatındaki karşılığının hesaplaması ise (4.9)'da verilmiştir.

$$\begin{aligned} & bit_{v-1} \cdot 2^{v-1} + bit_{v-2} \cdot 2^{v-2} + \dots + bit_1 \cdot 2 + bit_0 + bit_{-1} \cdot 2^{-1} + bit_{-2} \cdot 2^{-2} + \\ & \dots + bit_{-w+1} \cdot 2^{-w+1} + bit_{-w} \cdot 2^{-w} \end{aligned} \quad (4.9)$$

Kayan nokta formatındaki bir X_{kn} sayısının $Q(v.w)$ formatında sabit-nokta X_{sn} dönüştürülmesi için MATLAB *floor* komutu ile (4.10)'da verildiği şekilde yapılmıştır.

Çizelge 4.35 : ACM formatına göre kullanılan kipleme.

ACM Formatı	Kipleme	Kipleme Derecesi (m)	ACM Formatı	Kipleme	Kipleme Derecesi (m)
1	QPSK	2	20	32-APSK	5
2	QPSK	2	21	32-APSK	5
3	QPSK	2	22	32-APSK	5
4	QPSK	2	23	64-APSK	6
5	QPSK	2	24	64-APSK	6
6	QPSK	2	25	64-APSK	6
7	8-PSK	3	26	64-APSK	6
8	8-PSK	3	27	64-APSK	6
9	8-PSK	3	28	128-QCI	7
10	8-PSK	3	29	128-QCI	7
11	8-PSK	3	30	128-QCI	7
12	8-PSK	3	31	128-QCI	7
13	16-APSK	4	32	128-QCI	7
14	16-APSK	4	33	256-QCI	8
15	16-APSK	4	34	256-QCI	8
16	16-APSK	4	35	256-QCI	8
17	16-APSK	4	36	256-QCI	8
18	32-APSK	5	37	256-QCI	8
19	32-APSK	5			

Çizelge 4.36 : ACM formatına göre kullanılan APSK kiplemelerindeki halkaların yarıçap oranları.

ACM Formatı	Kipleme	R_2/R_1	R_3/R_1	R_4/R_1
13	16-APSK	3,15	UD	UD
14	16-APSK	3,15	UD	UD
15	16-APSK	2,85	UD	UD
16	16-APSK	2,75	UD	UD
17	16-APSK	2,60	UD	UD
18	32-APSK	2,84	5,27	UD
19	32-APSK	2,84	5,27	UD
20	32-APSK	2,84	5,27	UD
21	32-APSK	2,72	4,87	UD
22	32-APSK	2,54	4,33	UD
23	64-APSK	2,73	4,52	6,31
24	64-APSK	2,73	4,52	6,31
25	64-APSK	2,73	4,52	6,31
26	64-APSK	2,73	4,52	6,31
27	64-APSK	2,73	4,52	6,31

\underline{bit}_{v-1} \underline{bit}_{v-2} ... \underline{bit}_1 \underline{bit}_0 \underline{bit}_{-1} \underline{bit}_{-2} ... \underline{bit}_{-w+1} \underline{bit}_{-w}

Şekil 4.22 : Sabit-nokta $Q(v.w)$ formatındaki ikili sayı.

$$X_{sn} = \text{floor}(2^k \times X_{kn}) \div 2^k \quad (4.10)$$

Kipleme bloğu MATLAB modelinde kipleme yapıldıktan sonra elde edilen eş fazlı ve dördün bileşenler (4.10) kullanılarak $Q(2.12)$ formatına taşınmıştır.

Kipleme bloğu FPGA tasarımının arayüz diyagramı Şekil 4.23'te ve arayüz sinyallerinin açıklamaları Çizelge 4.37'de verilmiştir.

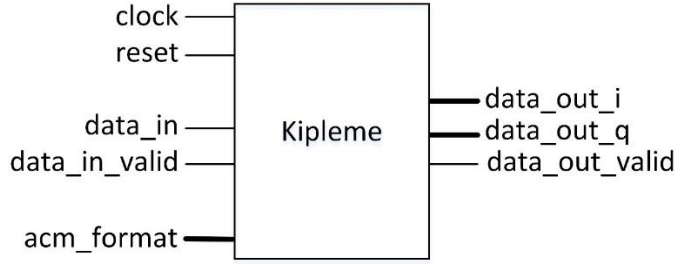
Kipleme bloğunda başlangıçta sistem yeniden başlatma sinyali gelir ve tasarımda kullanılan iç sinyaller başlangıç değerleri alır. Sonlu durum makineleri kullanılarak tasarlanan kipleme bloğunda sistem yeniden başlatma sinyalinden sonra ACM formatına göre yapılacak kipleme için uygun duruma geçilir. Giriş geçerli sinyali takip edilmeye başlanır. Giriş geçerli sinyaliyle birlikte alınan giriş verisi bir yazmaç serisine yazılır. Bu yazmaç serisinde kiplemede kullanılacak bit sayısı tamamlanınca kipleme yapılır. Her sembole ait MATLAB ile $Q(2.12)$ formatında oluşturulan eş fazlı ve dördün bileşenler bir taramalı tabloda tutulur. Kipleme işlemi sırasında alınan bitlere göre eşlenecek sembol belirlenir ve taramalı tablodan bu sembole ait eş fazlı ve dördün bileşenler çıkış geçerli sinyaliyle birlikte çıkarılır.

4.3.2 Pilot işareti ekleme

Pilot işaretleri, alıcıdaki taşıyıcı kestirimi, faz kestirimi ve eşzamanlama problemlerinin çözümünü kolaylaştırmak amacıyla veri sembollerinin aralarına eklenen ve alıcı tarafından bilinen sabit sembollerdir.

Pilot işareti ekleme bloğu FPGA tasarımının arayüz diyagramı Şekil 4.24'te ve arayüz sinyallerinin açıklamaları Çizelge 4.38'de verilmiştir.

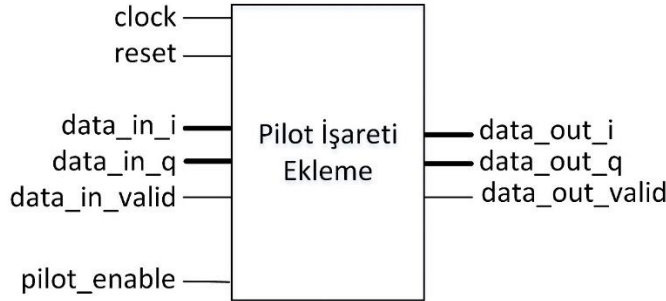
Pilot işareti ekleme bloğunda başlangıçta sistem yeniden başlatma sinyali gelir ve tasarımda kullanılan iç sinyaller başlangıç değerleri alır. Sonlu durum makineleri kullanılarak tasarlanan pilot işareti ekleme bloğunda sistem yeniden başlatma sinyalinden sonra pilot işareti ekleme bilgisine göre uygun duruma geçilir. Pilot işareti eklenmemesi durumunda giriş geçerli sinyaliyle birlikte alınan giriş verileri çıkış geçerli sinyaliyle birlikte çıkarılır. Pilot eklenmesi durumunda giriş sinyali geçerli sinyaliyle birlikte alınan 540 sembol çıkış geçerli sinyaliyle birlikte çıkarılır. 540 veri sembolü çıkıldıktan sonra 16 pilot işareti sembolü eş fazlı ve dördün bileşenleri çıkış geçerli sinyaliyle birlikte çıkarılır ve tekrar veri sembolleri için sayaçlar sıfırlanır.



Şekil 4.23 : Kipleme bloğu FPGA tasarımı arayüz sinyalleri.

Çizelge 4.37 : Kipleme bloğu FPGA tasarımı arayüz sinyal bilgileri.

Sinyal	Giriş/ Çıkış	Bit Genişliği	Tanım
clock	giriş	1	sistem saat sinyali
reset	giriş	1	sistem yeniden başlatma
data_in	giriş	1	giriş verisi
data_in_valid	giriş	1	giriş verisi geçerli
acm_format	giriş	6	ACM formatı
data_out_i	çıkış	14	eş fazlı çıkış verisi
data_out_q	çıkış	14	dördün çıkış verisi
data_out_valid	çıkış	1	çıkış verisi geçerli



Şekil 4.24 : Pilot işareti ekleme bloğu FPGA tasarımı arayüz sinyalleri.

Çizelge 4.38 : Pilot işareti ekleme bloğu FPGA tasarımı arayüz sinyal bilgileri.

Sinyal	Giriş/ Çıkış	Bit Genişliği	Tanım
clock	giriş	1	sistem saat sinyali
reset	giriş	1	sistem yeniden başlatma
data_in_i	giriş	14	eş fazlı giriş verisi
data_in_q	giriş	14	dördün giriş verisi
data_in_valid	giriş	1	giriş verisi geçerli
pilot_enable	giriş	1	pilot işareti ekleme bilgisi
data_out_i	çıkış	14	eş fazlı çıkış verisi
data_out_q	çıkış	14	dördün çıkış verisi
data_out_valid	çıkış	1	çıkış verisi geçerli

4.3.3 Fiziksel katman rastgeleleştirme

Fiziksel katman rastgeleleştirme bloğunda tekrarlı gelen veri sembolleri ve pilot işareti sembollerinin dağılımı yapılmaktadır.

Fiziksel katman rastgeleleştirme bloğu FPGA tasarımının arayüz diyagramı Şekil 4.25'te ve arayüz sinyallerinin açıklamaları Çizelge 4.39'da verilmiştir.

Fiziksel katman rastgeleleştirme bloğunda başlangıçta sistem yeniden başlatma sinyali gelir ve tasarımda kullanılan iç sinyaller başlangıç değerleri alır. Giriş verisi geçerli sinyaliyle birlikte alınan giriş verisi, kayan yazmaçlar ile oluşturulan rastgeleleştirici yapısına beslenir ve çıkan sonucuna göre rastgeleleştirme işlemi yapılır. Aynı saat sinyali içerisinde rastgeleleştirici durumları belirtilen şekilde güncellenir ve tüm çerçeve rastgeleleştirmesi bu şekilde devam eder.

Pilot işareti ekleme bilgisi ile çerçevedeki veri uzunluğu kontrol edilir. Pilot işareti eklenip eklenmemesine göre seçilen veri uzunluğu sayaçlar ile kontrol edilir. Çerçevenin tamamlandığı sayaçlar ile belirlenir ve çerçeve tamamlandıktan sonra rastgeleleştirici başlangıç durumuna getirilir ve yeni çerçeve için işlem devam eder.

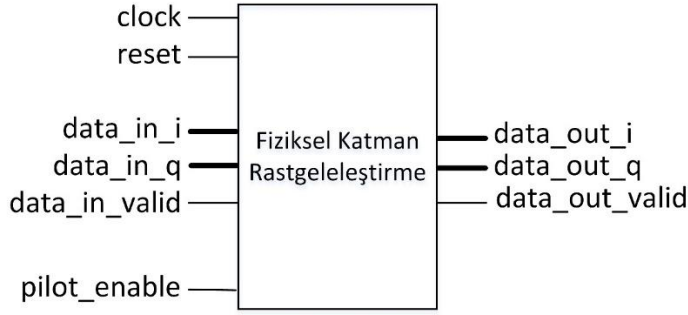
4.3.4 Çerçeve başlığı ekleme

Çerçeve başlığı ekleme bloğunda, alıcıya çerçevenin başladığı bilgisini ve çerçeve içerisindeki verinin oluşturulmasında kullanılan kodlama ve kipleme hakkında bilgi veren çerçeve bilgisini iletmek amacıyla çerçeve başlığı oluşturulur ve çerçevenin başına eklenir.

Çerçeve başlığı ekleme bloğu FPGA tasarımının arayüz diyagramı Şekil 4.26'da ve arayüz sinyallerinin açıklamaları Çizelge 4.40'ta verilmiştir.

Çerçeve başlığı ekleme bloğunda başlangıçta sistem yeniden başlatma sinyali gelir ve tasarımda kullanılan iç sinyaller başlangıç değerleri alır. Sistem başlangıç durumuna getirildiğinde ve sistem veri girişine hazır olduğunda girişe hazır sinyali 1 yapılır. Dışarıdan gelecek olan veri bu sinyale göre gönderilir. Girişe hazır sinyali 0 ise veri akışı önceki blok tarafından durdurulur.

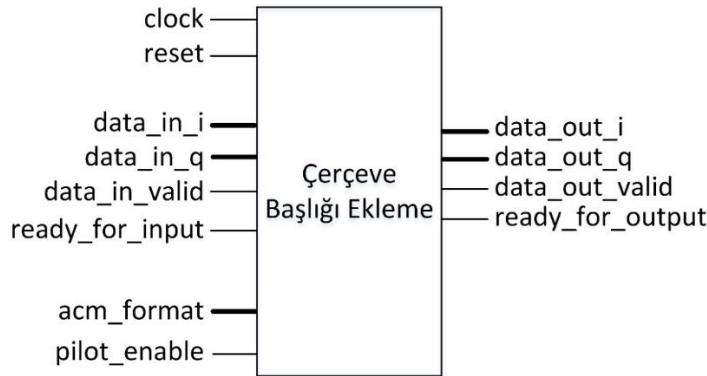
Sistem başlangıç durumuna getirildiğinde otomatik olarak çerçeve belirtecini üretmeye başlar. Tanımlanan kayan yazmaç yapısı ve uygun geribesleme bağlantıları ile 256-bit uzunluğundaki çerçeve belirteci üretilip yazmaç serisinde depolanır.



Şekil 4.25 : Fiziksel katman rastgeleleştirme bloğu FPGA tasarımı arayüz sinyalleri.

Çizelge 4.39 : Fiziksel katman rastgeleleştirme bloğu FPGA tasarımı arayüz sinyal bilgileri.

Sinyal	Giriş/Çıkış	Bit Genişliği	Tanım
clock	giriş	1	sistem saat sinyali
reset	giriş	1	sistem yeniden başlatma
data_in_i	giriş	14	eş fazlı giriş verisi
data_in_q	giriş	14	dördün giriş verisi
data_in_valid	giriş	1	giriş verisi geçerli
pilot_enable	giriş	1	pilot işareti ekleme bilgisi
data_out_i	çıkış	14	eş fazlı çıkış verisi
data_out_q	çıkış	14	dördün çıkış verisi
data_out_valid	çıkış	1	çıkış verisi geçerli



Şekil 4.26 : Çerçeve başlığı ekleme bloğu FPGA tasarımı arayüz sinyalleri.

Giriş verisi geçerli sinyaliyle birlikte alınan giriş verisi ilk-giren ilk-çıkış hafıza birimine yazılır. İlk giriş verisinin gelmesiyle birlikte, çerçeve belirteci bitleri belirtilen kiplemeye göre durum makineleri yapısı kullanılarak kiplenir ve kiplenen sembolün eş fazlı ve dördün bileşenleri çıkış verisi geçerli sinyaliyle birlikte çıkarılır. Bu işleme paralel olarak gelen giriş verisine ait ACM formatı ve pilot işareti ekleme

Çizelge 4.40 : Çerçeve başlığı ekleme bloğu FPGA tasarımı arayüz sinyal bilgileri.

Sinyal	Giriş/ Çıkış	Bit Genişliği	Tanım
clock	giriş	1	sistem saat sinyali
reset	giriş	1	sistem yeniden başlatma
data_in_i	giriş	14	eş fazlı giriş verisi
data_in_q	giriş	14	dördün giriş verisi
data_in_valid	giriş	1	giriş verisi geçerli
ready_for_input	çıkış	1	girişe hazır
acm_format	giriş	6	ACM formatı
pilot_enable	giriş	1	pilot işareti ekleme
data_out_i	çıkış	14	eş fazlı çıkış verisi
data_out_q	çıkış	14	eş fazlı çıkış verisi
data_out_valid	çıkış	1	çıkış verisi geçerli
ready_for_output	giriş	1	çıkışa hazır

bilgisi blok kodlama için bilgi vektörü olarak birleştirilir. Hazırlanan bilgi vektörü taramalı tabloda tutulan üreteç matrisi ile kod sözcüğünü oluşturmak amacıyla çarpılır. Matris çarpımı işlemleri bilgi bitleri ve üreteç matrisi bitlerinin uygun şekilde çarpılıp toplanması ile edilir. Yapılan çarpma ve toplama işlemlerinde AND ve XOR kapıları kullanılır. Kod sözcüğü oluşturulduktan sonra taramalı tabloda tutulan bir seri ile bitsel XOR işlemi uygulanır ve çerçeve tanımlayıcı oluşturulur.

Çerçeve belirteci sembollerinin kiplemesi ve çıkışı bittikten sonra 64-bit uzunluğundaki çerçeve tanımlayıcı bitleri kiplenir ve kiplenen sembolün eş fazlı ve dördün bileşenleri çıkış verisi geçerli sinyaliyle birlikte çıkarılır.

Çerçeve tanımlayıcı bitlerinin çıkılmasıyla çerçeve başlığı tamamlanmış olur. Çerçeve verisi ilk-giren ilk-çıkış hafıza biriminden okunmaya başlanır. Hafızadan okunan eş fazlı ve dördün bileşenler çıkış verisi geçerli sinyaliyle birlikte çıkarılır.

Çıkılan verilerin sıklığı 4 saat sinyalinde 1 olarak ayarlanır. Bunun sebebi tabanbant süzme bloğunda veri örnekleme hızının 4 katına çıkarılacak olmasıdır.

Girişe hazır sinyalleri ilk-giren ilk-çıkış hafıza birimleri doluluk oranıyla kontrol edilir. İlgili kodlayıcıya ait hafıza biriminin dolması halinde o kodlayıcıya ait girişe hazır sinyali 0 olarak gönderilir ve veri akışı durdurulur. Hafızada yer açılması halinde girişe hazır sinyali tekrar 1 olarak gönderilir ve işlem devam eder.

4.4 Tabanbant Süzme

Tabanbant süzme bloğunda yukarı örnekleme ve darbe şekillendirici süzgeçten geçirme işlemleri uygulanmaktadır. Semboller arası girişimi önlemek ve bant-kısıtlı kanal oluşturmak amacıyla kullanılan darbe şekillendirici süzgeç ile örneklere bu şeklin verilmesi için yukarı örnekleme yapılır. Asıl semboller arasına 0 sembolleri eklenerek örnekleme hızı artırılır [28].

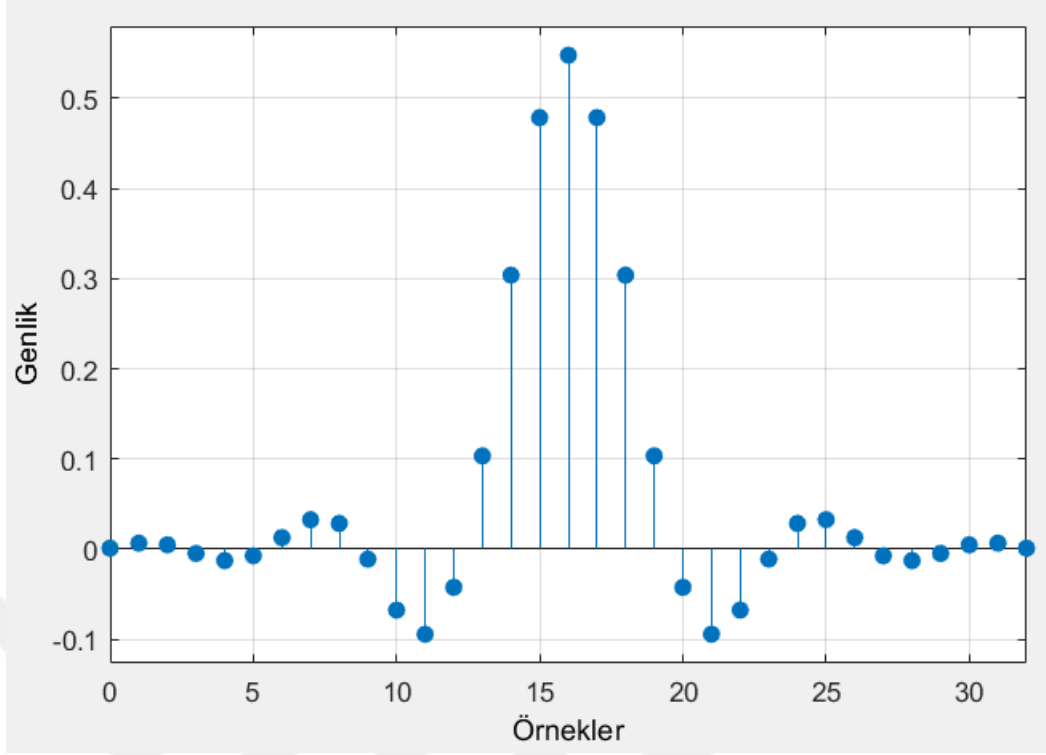
Bu tasarımda yukarı örnekleme oranı 4 olarak seçilmiştir. Gelen her sembolün arkasına üç tane 0 eklenerek yukarı örnekleme sağlanmış ve hızı sembol hızının 4 katına çıkarılmıştır. Yukarı örnekleme işleminden sonra, yukarı örnekleme ile çıkılan örnekleme hızında çalışan bir darbe şekillendirici süzgeç kullanılmıştır. Yukarı örneklenmiş veriler darbe şekillendirici süzgeçten geçerek sayısal-analog dönüşüme hazır hale gelir.

Darbe şekillendirici süzgeç tasarımında MATLAB *rcosdesing* komutu kullanılmıştır. Bu süzgeç için azalma faktörü $\alpha = 0,35$, yukarı örnekleme değeri $sps = 4$ ve örnek tarama genişliği $span = 8$ olarak belirlenmiş ve bu değişkenlerle darbe şekillendirici süzgeç oluşmuştur. Oluşan darbe şekillendirici süzgecin analizleri MATLAB süzgeç görselleştirme aracı ile yapılmıştır. Bu araç ile yapılan analizlere göre tasarlanan kök yükseltilmiş kosinüs süzgecinin dürtü yanıtı ve frekans yanıtı sırasıyla Şekil 4.27 ve 4.28’de verilmiştir.

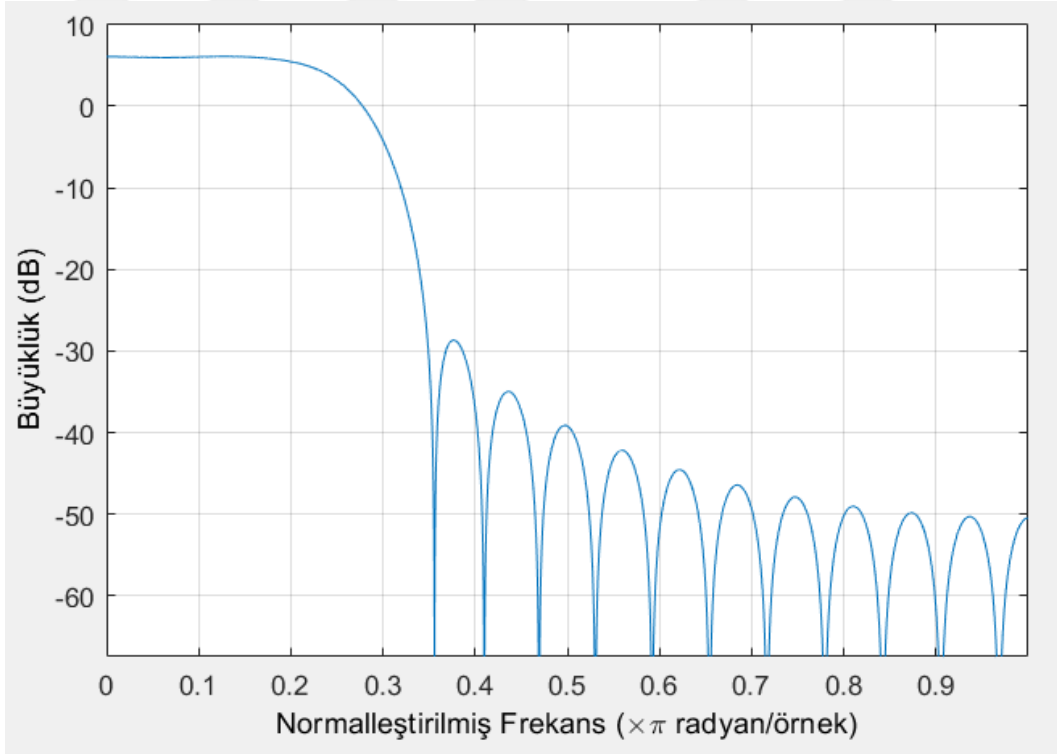
Tasarlanan süzgeç 33-basamaklı bir sonlu dürtü cevaplı süzgeç olarak elde edilmiştir.

Tabanbant süzme bloğu FPGA tasarımının arayüz diyagramı Şekil 4.29’da ve arayüz sinyallerinin açıklamaları Çizelge 4.41’de verilmiştir.

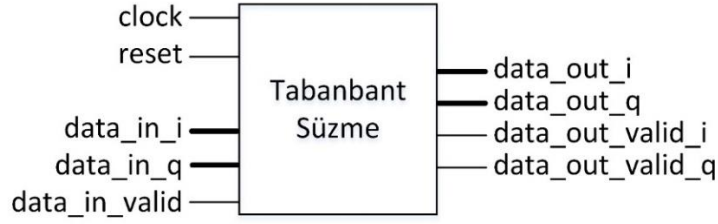
Tabanbant süzme bloğunda başlangıçta sistem yeniden başlatma sinyali gelir ve tasarımda kullanılan iç sinyaller başlangıç değerleri alır. Daha sonra giriş verisi geçerli sinyali takip edilir. Giriş verisi geçerli sinyaliyle alınan giriş verileri eş iki darbe şekillendirici filtreye gönderilir. Darbe şekillendirici filtre gerçekleştirilmesi hazır sonlu dürtü cevaplı süzgeç bloğu kullanılarak yapılmıştır. MATLAB üzerinde oluşturulan süzgece ait süzgeç katsayıları bu bloğa tasarım sırasında eklenmiştir. Giriş verisi almasıyla birlikte katsayıları verilen süzgeç ile süzme işlemini gerçekleştirmektedir. Alınan eş fazlı ve dördün bileşenler ayrı sonlu dürtü cevaplı süzgeç bloğuna girer ve çıkan değerler çıkış geçerli sinyalleriyle birlikte çıkarılır. İlk



Şekil 4.27 : Kök yükseltilmiş kosinüs süzgeci dürtü yanıtı.



Şekil 4.28 : Kök yükseltilmiş kosinüs süzgeci frekans yanıtı.



Şekil 4.29 : Tabanbant süzme bloğu FPGA tasarımı arayüz sinyalleri.

Çizelge 4.41 : Tabanbant süzme bloğu FPGA tasarımı arayüz sinyal bilgileri.

Sinyal	Giriş/ Çıkış	Bit Genişliği	Tanım
clock	giriş	1	sistem saat sinyali
reset	giriş	1	sistem yeniden başlatma
data_in_i	giriş	14	eş fazlı giriş verisi
data_in_q	giriş	14	dördün giriş verisi
data_in_valid	giriş	1	giriş verisi geçerli
data_out_i	çıkış	12	eş fazlı çıkış verisi
data_out_q	çıkış	12	eş fazlı çıkış verisi
data_out_valid_i	çıkış	1	eş fazlı çıkış verisi geçerli
data_out_valid_q	çıkış	1	dördün çıkış verisi geçerli

verinin sonlu dürtü cevaplı süzgeç bloğuna iletilmesinden sonra bu bloğa art arda üç 0 değeri iletilir ve böylece yukarı örnekleme sağlanmış olur. Yukarı örnekleme işleminden sonra tekrar giriş verisi geçerli sinyali takip edilir ve sistem bu şekilde devam eder.

4.5 FPGA Tasarım İyileştirmeleri

Uyarlamalı kodlama ve kipleme kullanan verici yapısı FPGA tasarımı üretilen çıkış verisi miktarını artırmak için tasarım iyileştirme çalışmaları yapılmıştır. Sistemi tam gücüyle çalıştırılabilmek için her saat sinyalinde çıkış verilebilmesi yani çıkış örnekleme hızının sistem saat sinyali frekansına ulaştırılması hedeflenmiştir.

Yapılan iyileştirme çalışmaları kapsamında, sistemin en yüksek veri hızlarında çalıştığı durumda sistemi her saat sinyalinde çıkış üretebilecek ve düşük veri hızlarında çıkış veri hızını sabit tutarak giriş veri hızını düşürecek şekilde çalışması için sistem blokları incelenmiştir. Her adımda çıkışta istenen veri hızını elde edebilecek giriş verileri hesaplanarak çıkıştan girişe doğru ilerlenmiştir.

İlk darboğaz olan kipleme bloğunda, giriş verisi örnek hızının saat sinyali frekansına

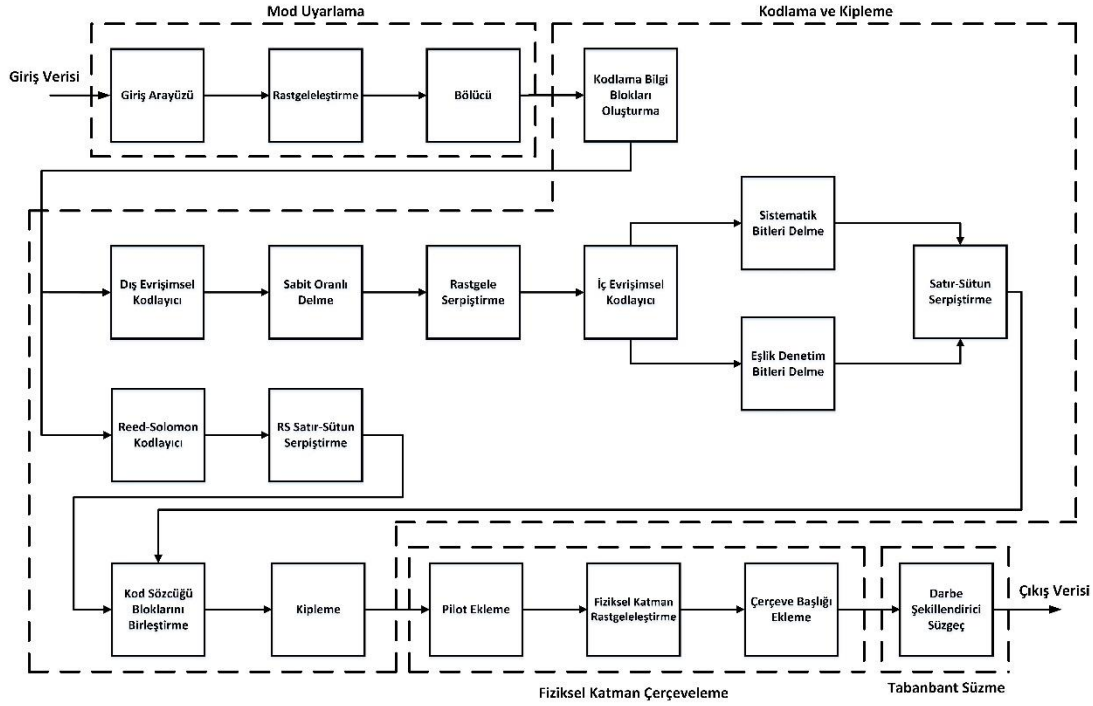
eşit olması durumunda bile bitlerden sembollere geçiş sebebiyle çıkış verisi örnek hızı saat sinyalinin $1/8$ 'ine inmektedir. Bu sebeple kipleme bloğunun en az 2 kat veri örnek hızına çıkması gerekmektedir. Giriş örnek hızı en yüksekte çalıştığı için, kipleme bloğu sistem saat frekansının yükseltilmesi veya kipleme bloğunun paralel bir şekilde çoklanması gerekmektedir.

Kipleme bloğuna veri akışı sağlayan kodlama blokları çıkışı, alabileceği en yüksek giriş örnek hızında bile blok yapılan işlemlerden ve yapılan işlem süresinden dolayı çıkış örnek hızı sistem saat sinyali frekansının yaklaşık $2/5$ 'i örnek hızındadır. Bu sebeple kodlayıcı blokları çıkış örnek hızının en az $5/2$ katına çıkması gerekir. Kodlayıcı bloklarının çıkış örnek hızının artırılması için sistem saat frekansının yükseltilmesi veya kipleme bloğunun paralel bir şekilde çoklanması gerekmektedir.

Yapılan çalışmalar doğrultusunda kodlayıcılar ve kipleme bloğunun birlikte iyileştirilmesine karar verilmiştir. FPGA tasarım blokları, blokların birbiri ile olan ilişkileri, yapısı ve örnekleme hızları göz önüne alınarak Şekil 4.30'da verildiği şekilde üst bloklara ayrılarak gruplandırılmıştır.

Özellikle kodlayıcılarda bulunan işlem gücünün yüksek olması sebebiyle kodlama ve kipleme üstbloğu sistem saat sinyali frekans artırımı yerine paralel bir şekilde çoklama işlemi uygun görülmüştür. Kodlama ve kipleme üstbloğu çoklanabilmesi için sistemin bloklar halinde çoklanması gerekmektedir. Seri birleştirilmiş evrişimsel kodlayıcıda bulunan evrişimsel kodlayıcıların kafes sonlandırması yapısı sebebiyle farklı bilgi blokları birbirinden bağımsız ve paralel çalışabileceği görülmüştür. Döngüsel blok kodlayıcıların da blok olarak çalışması sebebiyle, kodlama ve kipleme üstbloğunun bilgi blokları temelinde çoklanabileceği görülmüştür.

Yapılan analizler doğrultusunda, kodlayıcı çıkış örnek hızlarının yaklaşık $5/2$ kat artırılması gereksinimi sebebiyle kodlama ve kipleme üstbloğundan üç adet kullanılarak paralel çoklama işlemi yapılmıştır. Böylece kodlama ve kipleme üstbloğundan, sistem çıkış örnek hızını en yükseğe getirecek çıkış örnek hızı elde edilmiştir. Paralel olarak çoklanmış kodlama ve kipleme yapısında bilgi blok bütünlüğü bozulmadan, her bir kodlama ve kipleme üstbloğuna bir bilgi bloğu gelecek şekilde çoklama yapılmıştır. Alınan giriş bilgi blokları sırasıyla kodlama ve kipleme üstbloklarına dağıtılmıştır.

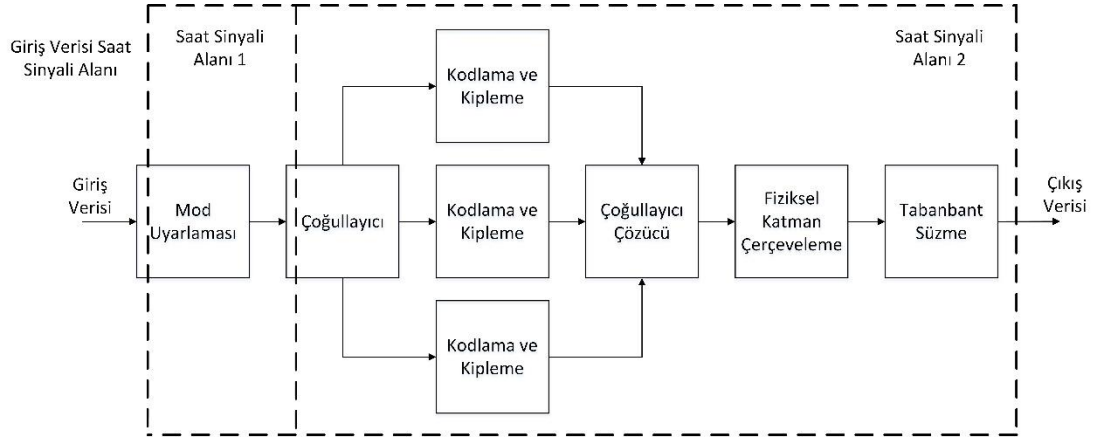


Şekil 4.30 : FPGA tasarımı üstblokları.

Yapılan paralel çoklama yapısının istenilen çıkış örnek hızlarında çalışabilmesi için yeterli giriş örnek hızlarında giriş oluşturulması gerekmektedir. Bu sebeple mod uyarlaması üstbloğunun çıkış örnek hızının, sistem saat frekansı artırılarak veya paralel çoklama işlemi yapılarak artırılması gerekmektedir. Mod uyarlaması üstbloğunda bulunan rastgeleleştirme ve bölücü blokları hem farklı ACM formatlarına hem de farklı iletim çerçevesi uzunluklarına uyumlu olarak çalışmak zorundadır. Bu üstbloğun paralel olarak çoklanması durumunda bu uyumlandırma sağlanamayacağı için, sistem saat sinyali frekansı artırılma yoluna gidilmiştir.

Mod uyarlaması üstbloğu için farklı bir saat sinyali alanı tanımlanmıştır. Kodlama ve kipleme çoklu bloklarına yeterli hızda veri sağlayabilmek üzere daha yüksek bir saat sinyali frekansı tanımlanması amaçlanmıştır. Yapılan farklı saat sinyali bölgeleri ve paralel çoklama işlemleri ile Şekil 4.30'da yapılan gruplandırmalara göre iyileştirilmiş FPGA tasarımı Şekil 4.31'de verilmiştir.

Yapılan bu iyileştirme çalışmaları doğrultusunda geliştirilen uyarlamalı kodlama ve kipleme kullanan verici FPGA tasarımı çıkış veri miktarı her saat sinyalinde bir veri olacak şekilde en fazlaya getirilmiştir. Bu çıkışı sağlayacak uygun hızda veri girişi desteklenebilir hale getirilmiştir. Bu çalışmaların sonucunda FPGA tasarımının işlem gücü en fazlaya getirilmiş ve iyileştirme işlemleri tamamlanmıştır.



Şekil 4.31 : İyileştirilmiş FPGA tasarımı.

5. TASARIM BAŞARIMLARI VE FPGA GERÇEKLEME SONUÇLARI

5.1 Tasarım Başarımları

Geliştirilen uyarlamalı kodlama ve kipleme kullanan verici yapısında mod uyarlama, seri birleştirilmiş evrişimsel kodlama, döngüsel blok kodlama, fiziksel katman çerçeveleme ve tabanbant süzme blokları birleştirilmiştir. ACM formatına göre uyarlama yapan verici yapısında kodlama için seçilen seri birleştirilmiş evrişimsel kodlama için bilgi blok uzunluğu K_{SCCC} , rastgele serpiştirme blok uzunluğu I , sistematik bitler delme sonrası blok uzunluğu S , eşlik denetim bitleri delme sonrası blok uzunluğu P , satır-sütun serpiştirme ve aynı zamanda kod sözcüğü blok uzunluğu N_{SCCC} , döngüsel blok kodlayıcı için bilgi blok uzunluğu K_{RS} , satır-sütun serpiştirme ve aynı zamanda kod sözcüğü blok uzunluğu N_{RS} değişkenleri Çizelge 5.1’de verilmiştir.

Çizelge 5.1’de seçilen değişkenlere göre seri birleştirilmiş evrişimsel kodlayıcı kod oranı r_{SCCC} , döngüsel blok kodlayıcı kod oranı r_{RS} ve genel kodlayıcı kod oranı r ACM formatına göre Çizelge 5.2’de verilmiştir.

ACM formatına göre uyarlama yapan verici yapısında, kullanılan kipleme yöntemi, kipleme derecesini belirten ve 1 sembole eşlenecek bit sayısı m , seri birleştirilmiş evrişimsel kodlayıcıdan alınan bit sayısı m_{SCCC} , döngüsel blok kodlayıcıdan alınan bit sayısı m_{RS} değerleri Çizelge 5.3’te verilmiştir.

Çizelge 5.1’de verilen kodlama değişkenleri ve Çizelge 5.3’te verilen kipleme değişkenlerine göre, Çizelge 5.2’de verilen kod oranlarıyla elde edilen spektral verimlilik olarak pilot eklenme durumuna ve iletim çerçevesi uzunluğu M değerlerinin farklı seçilmesi durumlarında Çizelge 5.4’teki değerler elde edilir.

Çizelge 5.1 : Uyarlamalı kodlama yapısı değişkenleri.

ACM Formatı	K_{SCCC}	I	S	P	N_{SCCC}	K_{RS}	N_{RS}
1	5758	8640	8642	7558	16200	0	0
2	6958	10440	10442	5758	16200	0	0
3	8398	12600	11510	4690	16200	0	0
4	9838	14760	12351	3849	16200	0	0
5	11278	16920	13200	3000	16200	0	0
6	13198	19800	14390	1810	16200	0	0
7	11278	16920	16470	7830	24300	0	0
8	13198	19800	15842	8458	24300	0	0
9	14878	22320	18602	5698	24300	0	0
10	17038	25560	19939	4361	24300	0	0
11	19198	28800	21218	3082	24300	0	0
12	21358	32040	22857	1443	24300	0	0
13	19198	28800	24482	7918	32400	0	0
14	21358	32040	25741	6659	32400	0	0
15	23518	35280	27051	5349	32400	0	0
16	25918	38880	28515	3885	32400	0	0
17	28318	42480	29880	2520	32400	0	0
18	25918	38880	31755	8745	40500	0	0
19	28318	42480	33137	7363	40500	0	0
20	30958	46440	34677	5823	40500	0	0
21	33358	50040	36197	4303	40500	0	0
22	35998	54000	37802	2698	40500	0	0
23	33358	50040	39366	9234	48600	0	0
24	35998	54000	41042	7558	48600	0	0
25	38638	57960	42507	6093	48600	0	0
26	41038	61560	43915	4685	48600	0	0
27	43678	65520	45429	3171	48600	0	0
28	16798	25200	23018	9382	32400	22680	24300
29	19438	29160	24399	8001	32400	22680	24300
30	22078	33120	26056	6344	32400	22680	24300
31	24718	37080	27689	4711	32400	22680	24300
32	27358	41040	29277	3123	32400	22680	24300
33	16558	24840	22689	9711	32400	30240	24300
34	19198	28800	24482	7918	32400	30240	32400
35	21838	32760	25773	6627	32400	30240	32400
36	24478	36720	27420	4980	32400	30240	32400
37	27358	41040	29277	3123	32400	30240	32400

Çizelge 5.2 : Uyarlamalı kodlama yapısı ile elde edilen kod oranları.

ACM Formatı	r_{SCCC}	r_{RS}	r
1	0,3554	0	0,3554
2	0,4295	0	0,4295
3	0,5184	0	0,5184
4	0,6073	0	0,6073
5	0,6962	0	0,6962
6	0,8147	0	0,8147
7	0,4641	0	0,4641
8	0,5431	0	0,5431
9	0,6123	0	0,6123
10	0,7012	0	0,7012
11	0,7900	0	0,7900
12	0,8789	0	0,8789
13	0,5925	0	0,5925
14	0,6592	0	0,6592
15	0,7259	0	0,7259
16	0,7999	0	0,7999
17	0,8740	0	0,8740
18	0,6400	0	0,6400
19	0,6992	0	0,6992
20	0,7644	0	0,7644
21	0,8237	0	0,8237
22	0,8888	0	0,8888
23	0,6864	0	0,6864
24	0,7407	0	0,7407
25	0,7950	0	0,7950
26	0,8444	0	0,8444
27	0,8987	0	0,8987
28	0,5185	0,9333	0,6963
29	0,5999	0,9333	0,7428
30	0,6814	0,9333	0,7894
31	0,7629	0,9333	0,8359
32	0,8444	0,9333	0,8825
33	0,5111	0,9333	0,7222
34	0,5925	0,9333	0,7629
35	0,6740	0,9333	0,8037
36	0,7555	0,9333	0,8444
37	0,8444	0,9333	0,8889

Çizelge 5.3 : Uyarlamalı kipleme yapısı değişkenleri.

ACM Formatı	Kipleme	m	m_{SCC}	m_{RS}
1	QPSK	2	2	0
2	QPSK	2	2	0
3	QPSK	2	2	0
4	QPSK	2	2	0
5	QPSK	2	2	0
6	QPSK	2	2	0
7	8-PSK	3	3	0
8	8-PSK	3	3	0
9	8-PSK	3	3	0
10	8-PSK	3	3	0
11	8-PSK	3	3	0
12	8-PSK	3	3	0
13	16-APSK	4	4	0
14	16-APSK	4	4	0
15	16-APSK	4	4	0
16	16-APSK	4	4	0
17	16-APSK	4	4	0
18	32-APSK	5	5	0
19	32-APSK	5	5	0
20	32-APSK	5	5	0
21	32-APSK	5	5	0
22	32-APSK	5	5	0
23	64-APSK	6	6	0
24	64-APSK	6	6	0
25	64-APSK	6	6	0
26	64-APSK	6	6	0
27	64-APSK	6	6	0
28	128-QCI	7	4	3
29	128-QCI	7	4	3
30	128-QCI	7	4	3
31	128-QCI	7	4	3
32	128-QCI	7	4	3
33	256-QCI	8	4	4
34	256-QCI	8	4	4
35	256-QCI	8	4	4
36	256-QCI	8	4	4
37	256-QCI	8	4	4

Çizelge 5.4 : Uyarlamalı kodlama ve kipleme yapısı ile elde edilen spektral verimlilik.

ACM Formatı	Pilot İşareti Eklemeli				Pilot İşareti Eklemez			
	$M_{ic} =$ 1738	$M_{ic} =$ 3568	$M_{ic} =$ 7136	$M_{ic} =$ 8920	$M_{ic} =$ 1738	$M_{ic} =$ 3568	$M_{ic} =$ 7136	$M_{ic} =$ 8920
1	0,6763	0,6826	0,6857	0,6863	0,6966	0,7028	0,7059	0,7066
2	0,8172	0,8249	0,8286	0,8293	0,8418	0,8493	0,8531	0,8538
3	0,9864	0,9956	1,0001	1,0010	1,0160	1,0250	1,0296	1,0305
4	1,1555	1,1663	1,1715	1,1726	1,1902	1,2008	1,2062	1,2072
5	1,3247	1,3371	1,3430	1,3442	1,3644	1,3766	1,3827	1,3840
6	1,5502	1,5647	1,5717	1,5731	1,5967	1,6109	1,6181	1,6196
7	1,3247	1,3371	1,3430	1,3442	1,3644	1,3766	1,3827	1,3840
8	1,5502	1,5647	1,5717	1,5731	1,5967	1,6109	1,6181	1,6196
9	1,7475	1,7639	1,7717	1,7733	1,8000	1,8160	1,8241	1,8257
10	2,0012	2,0199	2,0289	2,0308	2,0613	2,0796	2,0889	2,0908
11	2,2549	2,2760	2,2862	2,2882	2,3226	2,3433	2,3537	2,3558
12	2,5086	2,5321	2,5434	2,5457	2,5839	2,6069	2,6186	2,6209
13	2,2549	2,2760	2,2862	2,2882	2,3226	2,3433	2,3537	2,3558
14	2,5086	2,5321	2,5434	2,5457	2,5839	2,6069	2,6186	2,6209
15	2,7623	2,7882	2,8006	2,8031	2,8453	2,8706	2,8834	2,8860
16	3,0442	3,0727	3,0864	3,0892	3,1356	3,1635	3,1776	3,1805
17	3,3261	3,3572	3,3722	3,3752	3,4260	3,4564	3,4719	3,4750
18	3,0442	3,0727	3,0864	3,0892	3,1356	3,1635	3,1776	3,1805
19	3,3261	3,3572	3,3722	3,3752	3,4260	3,4564	3,4719	3,4750
20	3,6362	3,6702	3,6866	3,6899	3,7454	3,7787	3,7955	3,7989
21	3,9181	3,9547	3,9724	3,9760	4,0357	4,0716	4,0898	4,0934
22	4,2281	4,2677	4,2868	4,2906	4,3551	4,3938	4,4135	4,4174
23	3,9181	3,9547	3,9724	3,9760	4,0357	4,0716	4,0898	4,0934
24	4,2281	4,2677	4,2868	4,2906	4,3551	4,3938	4,4135	4,4174
25	4,5382	4,5807	4,6011	4,6052	4,6745	4,7161	4,7371	4,7414
26	4,8201	4,8652	4,8869	4,8913	4,9649	5,0090	5,0314	5,0359
27	5,1302	5,1782	5,2013	5,2060	5,2843	5,3313	5,3551	5,3598
28	4,6369	4,6803	4,7011	4,7054	4,7762	4,8186	4,8401	4,8444
29	4,9470	4,9933	5,0155	5,0200	5,0955	5,1408	5,1638	5,1684
30	5,2570	5,3062	5,3299	5,3347	5,4149	5,4631	5,4875	5,4924
31	5,5671	5,6192	5,6443	5,6494	5,7343	5,7853	5,8111	5,8163
32	5,8772	5,9322	5,9587	5,9640	6,0537	6,1075	6,1348	6,1403
33	5,4966	5,5481	5,5729	5,5778	5,6617	5,7121	5,7376	5,7427
34	5,8067	5,8611	5,8872	5,8925	5,9811	6,0343	6,0612	6,0667
35	6,1168	6,1741	6,2016	6,2072	6,3005	6,3565	6,3849	6,3906
36	6,4269	6,4870	6,5160	6,5218	6,6199	6,6788	6,7086	6,7146
37	6,7652	6,8285	6,8590	6,8651	6,9684	7,0303	7,0617	7,0680

5.2 FPGA Davranışsal Benzetim Sonuçları

Uyarlamalı kodlama ve kipleme kullanan verici MATLAB sabit-nokta modeli ile FPGA tasarımı davranışsal benzetim sonuçları karşılaştırılmıştır.

Giriş verisi olarak MATLAB üzerinde rastgele ikili sayı serisi üretilmiştir. Üretilen rastgele seri dosyaya yazdırılarak çıktısı alınmıştır. Oluşturulan dosya FPGA davranışsal benzetimlerine giriş olarak verilmiştir. Uyarlamalı kodlama ve kipleme kullanan verici FPGA tasarımı farklı ACM formatlarında çalıştırılarak her durum için davranışsal benzetimler tekrarlanmıştır. Uyarlamalı kodlama ve kipleme yapısının en yüksek hızla çalıştığı modda yapılan, sistem saat hızlarının ve her saat sinyalinde üretilen çıkış hızlarının görülebileceği gösteren ekran görüntüsü Şekil 5.1’de verilmiştir.

Uyarlamalı kodlama ve kipleme kullanan verici FPGA tasarımının en yüksek hızla çalıştığı modda yapılan ve bu modda her saat sinyalinde çıkış üretilbildiğini gösteren uzun süreli benzetim sonuçları ekran görüntüsü Şekil 5.2’de verilmiştir.

Tasarlanan her blok için çıktılar dosyaya yazdırılmış ve bir dosya karşılaştırma programı ile doğrulanmıştır. Daha sonra bloklar birleştirilmiş ve benzetimler tekrarlanmış, çıkış verileri dosyaya yazdırıp dosyalar karşılaştırılarak doğrulamalar yapılmıştır. Dosya karşılaştırma örneği olan ekran görüntüsü Şekil 5.3’te verilmiştir.

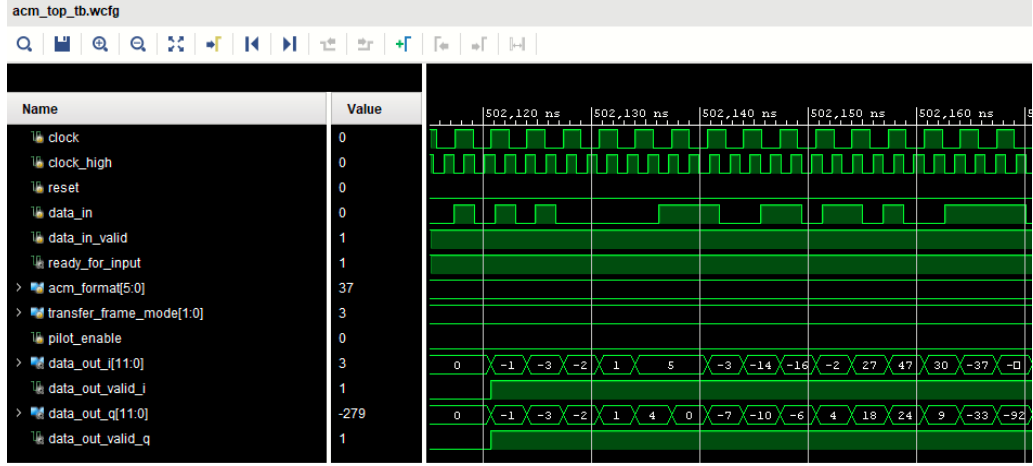
5.3 FPGA Seçimi ve Gerçekleme Sonuçları

5.3.1 FPGA seçimi

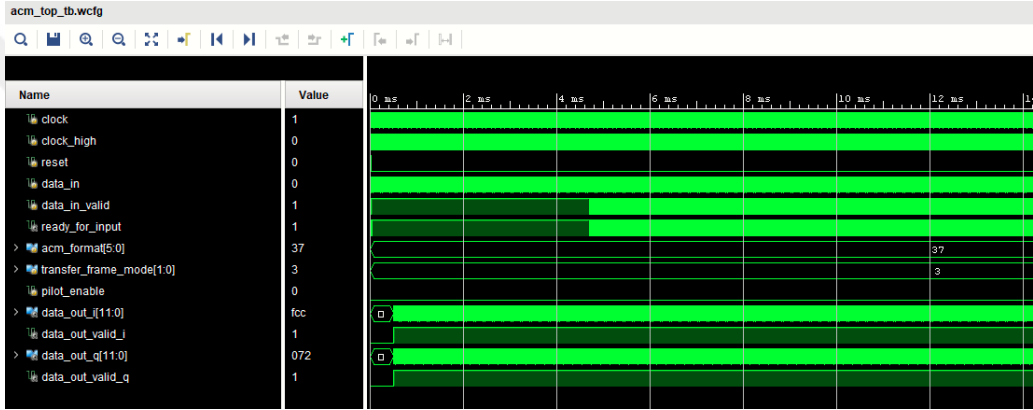
FPGA seçiminde öncelikle FPGA teknolojileri arasındaki ödünleşmeler incelenmiştir. Yüksek kapasite ve yüksek performans gereksinimleri sebebiyle SRAM-tabanlı FPGA türleri üzerine yoğunlaşmıştır.

SRAM-tabanlı FPGA türlerinde en yüksek kapasite ve performansı sunan ve sektör lideri olan Xilinx firması SRAM-tabanlı FPGA türleri incelenmiştir. Yapılan incelemeler ve performans karşılaştırmaları sonucunda güncel bir FPGA ailesi olan Xilinx Ultrascale ailesi üzerine yoğunlaşmıştır [29].

Uzay uygulamalarında güvenilirlik son derece önemlidir. Mümkün olduğu sürece radyasyon dayanımı yüksek ürünlerin kullanımı hedeflenmektedir. Bu kapsamda yüksek kapasite ve performans sunan güvenilir FPGA türleri araştırılmıştır. Xilinx



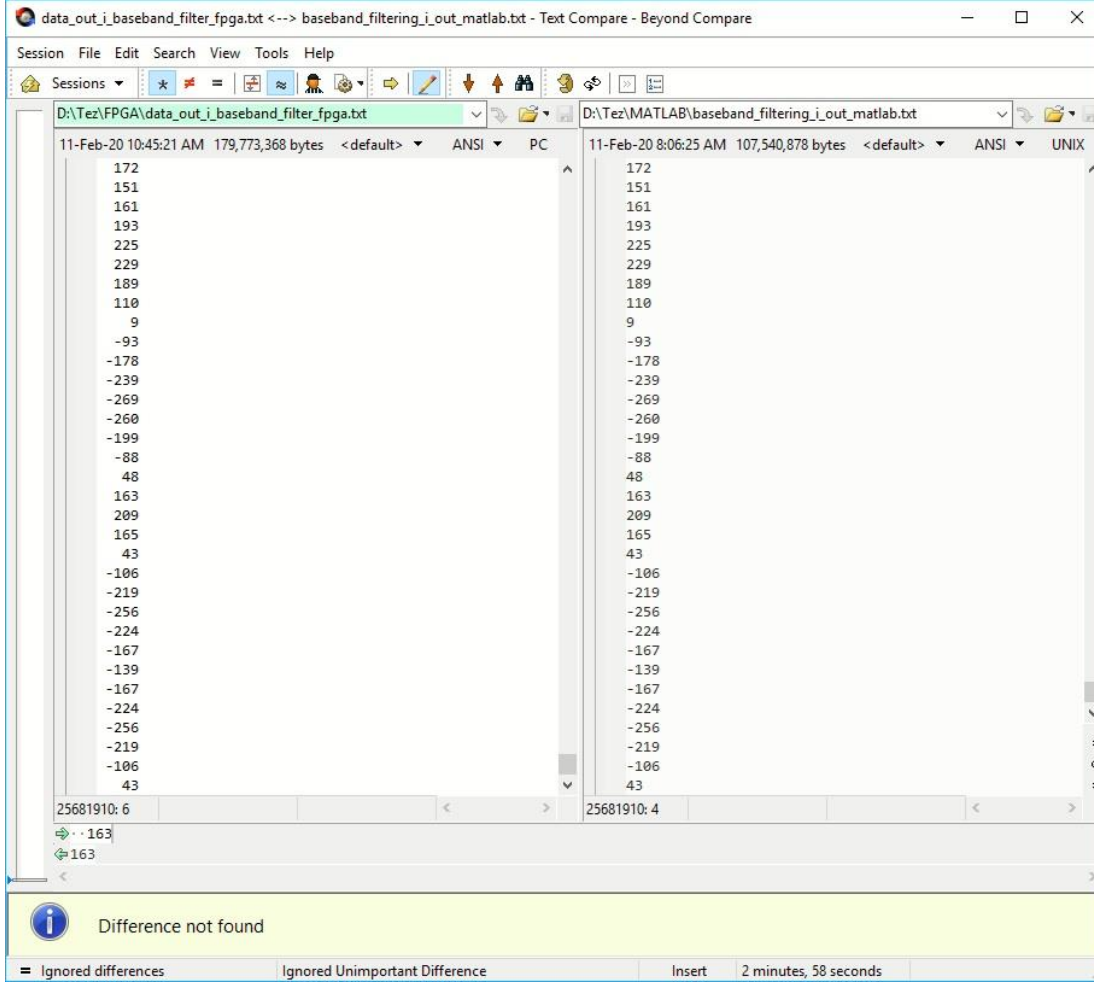
Şekil 5.1 : FPGA davranışsal benzetim ortamında çıkış sinyalleri.



Şekil 5.2 : FPGA davranışsal benzetim sonuçları.

firması nispeten eski teknoloji FPGA aileleri için uzay uyumlu radyasyon dayanımı yüksek FPGA modelleri üretmiştir ancak bu nispeten eski teknoloji sebebiyle performans istenilen seviye elde edilememektedir. Xilinx firması yakında zamanda Kintex Ultrascale FPGA ailesinden bir uzay uyumlu ürün sunulacağını duyurmuştur. Uzay uyumlu radyasyon dayanımı yüksek olarak sunulacak xqrku060 FPGA modelinin uzay uyumlu olmayan xcku060-ffva1517 FPGA modeli ile birebir eş olduğu ancak üretim teknikleri ve üretimde kullanılan malzemeler ile radyasyon dayanımının artırıldığı duyurulmuştur [30].

Bu bilgiler ışığında uyarlamalı kodlama ve kipleme kullanan verici yapısının uzay uyumlu olmayan xcku060-ffva1517 modeli üzerinde geliştirilmesi uygun bulunmuştur. Uzay uyumlu xqrku060 FPGA modelinin kullanıma hazır olması durumunda, bu tasarımın kolayca xqrku060 FPGA modeline aktarılabilir olması sebebiyle geliştirmelerin uzay uyumlu olmayan xcku060-ffva1517 yapılması ve daha sonra uzay uyumlu bir FPGA olan xqrku060 üzerinde kullanımı hedeflenmiştir.



Şekil 5.3 : FPGA davranışsal benzetim sonuçları ve MATLAB sonuçları karşılaştırılması.

5.3.2 Kaynak kullanımı

Seçilen Xilinx Kintex Ultrascale xcku060-ffva1517-3-e FPGA üzerinde gerçekleştirme işlemi yapılmıştır. Yapılan gerçekleştirme işlemleri sonucunda gerçekleştirme sonuçları ve raporları incelenmiştir. Seçilen xcku060-ffva1517-3-e FPGA üzerindeki kullanılabilir kaynaklar ve uyarlamalı kodlama ve kipleme kullanan verici FPGA tasarımının kullandığı kaynak miktarı Çizelge 5.5'te verilmiştir.

5.3.3 Güç tüketimi

Seçilen Xilinx Kintex Ultrascale xcku060-ffva1517-3-e FPGA üzerinde yapılan gerçekleştirme işleminin ardından, gerçekleştirilen tasarımın yaklaşık 27 °C bağlantı sıcaklığı için ortalama güç tüketim değeri 2,074 W olarak analiz edilmiş ve güç dağılımı Çizelge 5.6'da verilmiştir.

Çizelge 5.5 : Uyarlamalı kodlama ve kipleme kullanan verici yapısı FPGA tasarımı kaynak kullanımı.

Kaynak	Kullanılabilir Miktar	Kullanılan Miktar	Kullanım Oranı (%)
LUT	331680	26518	8,00
LUTRAM	146880	292	0,20
FF	663360	7202	1,09
BRAM	1080	241	22,31
DSP	2760	40	1,45
IO	624	42	6,73
BUFG	624	2	0,32

Çizelge 5.6 : Uyarlamalı kodlama ve kipleme kullanan verici yapısı FPGA tasarımı güç tüketimi.

Güç Tipi	Güç Tüketen	Tüketim Miktarı (W)	Tüketim Oranı (%)
Dinamik Güç	Saat Sinyalleri	0,165	12
	İç Sinyaller	0,332	24
	Mantık Birimleri	0,401	29
	BRAM	0,455	33
	DSP	0,020	1
	IO	0,016	1
	Toplam	1,389	67
Statik Güç	Toplam	0,684	33

5.3.4 Zamanlama analizi

Seçilen Xilinx Kintex Ultrascale xcku060-ffva1517-3-e FPGA üzerinde yapılan gerçeklemlerde, tanımlanan iki ayrı saat sinyali alanı için ayrı ayrı zamanlama analizi yapılmıştır.

Analiz sonuçları incelenerek saat sinyali alanı 1’de tanımlı saat sinyali frekansı için en yüksek frekans yaklaşık 530,22 MHz bulunurken, saat sinyali alanı 2’de tanımlı saat sinyali frekansı için en yüksek frekans yaklaşık olarak 300,5 MHz olacak bulunmuştur.

Yapılan zamanlama analizleri sonucunda saat sinyali alanı 1’de tanımlı saat sinyali frekansı 530,2 MHz, saat sinyali alanı 2’de tanımlı saat sinyali frekansı 300 MHz olarak alınmış ve sistem veri hızı analizleri bu frekansa göre yapılmıştır. Uyarlamalı kodlama ve kipleme kullanan verici yapısının pilot işareti eklenmeyen ve iletim çerçevesi uzunluğunun 8920 olarak seçildiği en yüksek verimlilikte ulaşabildiği çıkış örnekleme hızı, çıkış sembol hızı, spektral verimlilik ve giriş veri hızları Çizelge 5.7’de verildiği gibi elde edilmiştir.

Çizelge 5.7 : Uyarlamalı kodlama ve kipleme kullanan verici yapısı FPGA tasarımı ile elde edilen veri hızları.

ACM Formatı	Çıkış Örnekleme Hızı (MSps)	Çıkış Sembol Hızı (Mps)	Spektral Verimlilik	Giriş Veri Hızı (Mbps)
1	300	75	0,7066	53,00
2	300	75	0,8538	64,04
3	300	75	1,0305	77,29
4	300	75	1,2072	90,54
5	300	75	1,3840	103,80
6	300	75	1,6196	121,47
7	300	75	1,3840	103,80
8	300	75	1,6196	121,47
9	300	75	1,8257	136,93
10	300	75	2,0908	156,81
11	300	75	2,3558	176,69
12	300	75	2,6209	196,57
13	300	75	2,3558	176,69
14	300	75	2,6209	196,57
15	300	75	2,8860	216,45
16	300	75	3,1805	238,54
17	300	75	3,4750	260,63
18	300	75	3,1805	238,54
19	300	75	3,4750	260,63
20	300	75	3,7989	284,92
21	300	75	4,0934	307,01
22	300	75	4,4174	331,31
23	300	75	4,0934	307,01
24	300	75	4,4174	331,31
25	300	75	4,7414	355,61
26	300	75	5,0359	377,69
27	300	75	5,3598	401,99
28	300	75	4,8444	363,33
29	300	75	5,1684	387,63
30	300	75	5,4924	411,93
31	300	75	5,8163	436,22
32	300	75	6,1403	460,52
33	300	75	5,7427	430,70
34	300	75	6,0667	455,00
35	300	75	6,3906	479,30
36	300	75	6,7146	503,60
37	300	75	7,0680	530,10

6. DEĞERLENDİRME VE GELECEK ÇALIŞMALAR

Bu tez kapsamında yapılan çalışmalarda alçak yörünge uydu haberleşme sistemi gereksinimleri ve operasyonu üzerine çalışılmıştır. Yapılan çalışmalar ışığında uyarlamalı kodlama ve kiplemeli haberleşme yapılarının gereksinimi görülmüştür. Daha önce yapılan çalışmalar ve tanımlanan standartlar incelenmiş, yeni önerilen yapılar göz önüne alınarak FPGA tabanlı uyarlamalı kodlama ve kipleme kullanan bir verici yapısı geliştirilmiştir.

Bu çalışmada, daha önce yapılan çalışmalar ışığında önerilen yapının gerçekleşmesi için gerekli olan bazı parametreler ve bazı kodlama blokları önerilmiştir. Önerilen parametre ve bloklar doğrultusunda geliştirilen bu yapı öncelikle MATLAB ile modellenmiş ve MATLAB ortamında benzetimleri yapılmıştır. MATLAB modeli referans alınarak geliştirilen bu yapının sayısal tasarımı Verilog donanım tanımlama dili kullanılarak yapılmıştır. Yapılan sayısal tasarım, davranışsal olarak benzetim ortamında MATLAB modeli ile doğrulanmıştır. Doğrulan tasarım yüksek performanslı programlanabilir bir sayısal donanım olan FPGA üzerinde gerçekleşmiştir. Gerçeklemeler sonucu elde edilen sistem performansları incelenmiştir.

Bu çalışmayla birlikte, daha önceki standart ve çalışmalar ışında önerilen uyarlamalı kodlama ve kipleme kullanan 37-modlu verici yapısı ilk kez oluşturulmuştur. Oluşturulan bu yapının farklı modlar aracılığıyla 0,3554 ile 0,8987 arasında kod oranlarına ve 0,6763 ile 7,068 arasında spektral verimliliğe erişebildiği görülmüştür. Bu yapının MATLAB ile modellemesi ve benzetimleri ilk kez yapılmıştır. Modellemesi yapılan bu yapının ilk kez sayısal tasarımı yapılmış ve bir FPGA üzerinde gerçekleşmiştir. Yapılan gerçekleştirme ile bu yapının 300 Megaörnek/saniye (Megasamples per second, MSps) çıkış örnekleme hızına, 75 Megasembol/saniye (Megasympols per second, Msps) çıkış sembol hızına erişebildiği görülmüştür. Gerçeklenen sayısal tasarımın farklı modlar aracılığıyla 53 ile 530,1 Megabit/saniye (Megabits per second, Mbps) arasında giriş hızlarını destekleyebildiği görülmüştür. Geliştirilen uyarlamalı kodlama ve kipleme kullanan 37-modlu verici yapısının veri

hızları açısından donanım üzerindeki performansı ilk kez bu çalışma ile gösterilmiştir.

Gelecek çalışmalar kapsamında, bu tez çalışmasında geliştirilen ve FPGA üzerinde gerçekleştirilen uyarlamalı kodlama ve kipleme kullanan verici yapısı sayısal tasarımını donanım üzerinde test etme çalışmaları başlamıştır. Yapılan donanım testleri sonucunda, teorik olarak elde edilen başarımların ve analizler sonucu çıkarılan veri hızlarının donanım üzerinde doğrulanması hedeflenmektedir. Ayrıca bu tez kapsamında geliştirilen verici yapısının, bu çalışmaları destekleyen TÜBİTAK Uzay Teknolojileri Araştırma Enstitüsü tarafından geliştirilen yeni nesil uydularda faydalı yük haberleşmesi için kullanılması planlanmaktadır.

Bu tez çalışması kapsamında geliştirilen verici yapısında önerilen serpiştirme, delme parametrelerinin ve döngüsel blok kodlayıcının performansa etkileri incelenip, bu parametre ve yapıların iyileştirmeleri gelecek çalışmalar olarak yapılabilir.

Geliştirilen uyarlamalı kodlama ve kipleme kullanan verici yapısının teorik olarak performansını görmek ve iyileştirme çalışmaları yapmak amacıyla uygun alıcı yapısı gelecek çalışma olarak geliştirilebilir ve verici tasarımıyla birlikte genel sistem performansı test edilebilir.

KAYNAKLAR

- [1] **İnceöz, E., Tutgun, R., Yüksel Turgut, A.M.,** (2020). “FPGA Based Transmitter Design Using Adaptive Coding and Modulation Schemes for Low Earth Orbit Satellite Communications”, *IEEE 5th International Symposium on Telecommunication Technologies*.
- [2] **Aydoğan, Y.,** (2017). Yüksek Hızlı Alçak İrtifa Uydu Haberleşmesi için Uyarlamalı Modülasyon ve Kodlama Yöntemlerinin Gerçeklenmesi (Yüksek Lisans Tezi).
- [3] **Benedetto, S., Garello, R., Montorsi, G., Berrou, C., Douillard, C., Giancrifofaro, D., Spazio, A., Ginesi, A., Giugno, L., Luise, M.,** (2005). “MHOMS: High-Speed ACM Modem for Satellite Applications”, *IEEE Wireless Communications*, 12(2), 66-77.
- [4] **CCSDS** <https://public.ccsds.org/>, Alındığı Tarih: 20.09.2020
- [5] **CCSDS** (2012). Flexible Advanced Coding and Modulation Scheme for High Rate Telemetry Applications, CCSDS 131.2-B-1.
- [6] **Wertz, P., Hespeler, B., Kiessling, M., Hagemanns, F.J.,** (2016). “Next generation high data rate downlink subsystems based on a flexible APSK modulator applying SCCC encoding,” *7th ESA International Workshop on Tracking, Telemetry and Command Systems for Space Applications*.
- [7] **Fukami, T., Watanabe, H., Saito, H., Tomiki, A., Mizuno, T., Ivakiri, N., Shigeta, O., Nunomura, H., Kojima, K., Shinke, T., Kawamoto, K.,** (2015). “Field Tests of 348 Mbps High Speed Downlink System for 50-kg Class Satellite”.
- [8] **Ingeniars,** CCSDS Telemetry Transmitter IP Core.
- [9] **Creonic,** CCSDS SCCC Turbo Encoder and Decoder IP Core.
- [10] **Ugolini, A., Montorsi, G., Member, S.,** (2018). “Next Generation High-Rate Telemetry”, *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 2, pp. 327–337.
- [11] **Clarke, A.C.,** (1945). “Extra-terrestrial relays,” *Electronics World*, vol. 119, no. 1924. pp. 14–19.
- [12] **Kolawole, M.O.,** (2012). Satellite Communication Engineering, Marcel Dekker Inc., USA.
- [13] **Sebestyen, G., Fujikawa, S., Galassi, N., Chuchr A.,** (2018). Low Earth Orbit Satellite Design, Springer International Publishing, USA
- [14] **Maral, G., Bousquet, M.,** (2009). Satellite Communication Systems, John Wiley & Sons Ltd.

- [15] **Pelton, J.N.**, (2012). *Satellite Communications*, Springer.
- [16] **Proakis, J.G., Salehi, M.**, (2008) *Digital Communications*, The McGraw-Hill Companies, Inc.
- [17] **Weste, N.H.E., Harris, D.M.**, (2011). *CMOS VLSI Design*, Pearson Education, Inc.
- [18] **Mano, M.M., Ciletti, M.D.**, (2013). *Digital Design*, Pearson Education, Inc.
- [19] **Trimberger, S.M.**, (1994). *Field-Programmable Gate Array Technology*, Springer Science+Business Media.
- [20] **Battezzati, N., Sterpone, L., Violante, M.**, (2011). *Reconfigurable Field Programmable Gate Arrays for Mission-Critical Applications*, Springer.
- [21] **Sarıtaş, E., Karataş, S.**, (2015). *Her Yönüyle FPGA ve VHDL*, Palme Yayınevi.
- [22] **Randiva, N.K., Borkhade, G.P.**, (2014). A review on implementation of data scrambler & descrambler system using VHDL, *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 3, Issue 12.
- [23] **Lin, S., Costello, D.J.**, (2004). *Error Control Coding*, Pearson Education, Inc.
- [24] **Das, B., Sarma, M.P., Sarma, K.K.**, (2015). Different Aspects of Interleaving Techniques in Wireless Communication, *Intelligent Applications for Heterogeneous System Modeling and Design*, pp. 335-374.
- [25] **Goldsmith, A.**, (2005). *Wireless Communications*, Cambridge University Press.
- [26] **Kayhan, F.**, (2016). “QAM to circular isomorphic constellations”, *8th Adv. Satell. Multimed. Syst. Conf. 14th Signal Process. Sp. Commun. Work. ASMS/SPSC*.
- [27] **Çankaya, Ş.**, (2014). *Tek ve Çift Katmanlı Kiplenmeler için İdeal Olmayan Bozulmuş Etkileri (Yüksek Lisans Tezi)*.
- [28] **Muratoğlu Çürük, S., Tanık, Y.**, (2005). Darbe Şekillendirici Filtre Tasarımında Yeni Bir Yaklaşım, *İletişim Teknolojileri Ulusal Sempozyumu*.
- [29] **Xilinx Inc.**, (2019). *UltraScale Architecture and Product Data Sheet: Overview*.
- [30] **Elftman, D.**, (2018). *Xilinx Space Products-Space Environment FPGA User Workshop*.

EKLER

EK 1: Rastgele Serpiştirme Değişkenleri



EK 1

Çizelge Ek.1 : Rastgele serpiştirme değişkenleri 1 ($I = 25200, I = 29160, I = 33120, I = 37080, I = 41040$).

	$I = 25200$		$I = 29160$		$I = 33120$		$I = 37080$		$I = 41040$	
	α	β	α	β	α	β	α	β	α	β
0	50	32	109	98	63	93	30	57	238	48
1	134	71	179	66	238	10	165	114	260	88
2	200	47	129	13	92	78	156	65	253	11
3	68	89	214	109	189	72	235	67	208	43
4	74	15	186	45	73	25	15	19	130	102
5	1	18	128	72	172	68	126	90	190	79
6	32	40	60	8	94	96	203	0	193	2
7	102	69	112	91	275	14	270	24	122	73
8	33	79	8	93	49	6	189	15	118	47
9	154	119	183	117	228	5	103	78	317	95
10	21	29	170	64	218	46	145	36	177	42
11	22	111	225	114	197	47	301	43	224	111
12	103	61	35	15	173	111	186	87	143	89
13	162	108	97	77	232	43	264	63	181	100
14	130	54	87	35	195	72	218	78	178	86
15	83	16	88	85	144	53	52	21	29	30
16	38	99	188	6	21	50	265	32	64	40
17	190	105	47	99	134	49	141	85	173	81
18	180	55	165	107	161	65	181	21	315	50
19	61	116	156	88	274	89	208	119	198	3
20	104	75	52	55	135	65	9	70	46	7
21	167	114	233	15	102	76	88	103	258	0
22	168	21	151	95	259	57	196	92	108	98
23	100	101	240	20	101	37	32	80	106	69
24	191	81	0	17	200	60	78	96	7	50
25	26	19	17	115	3	87	130	118	167	101
26	142	67	235	71	7	70	162	114	308	51
27	10	43	157	22	115	12	300	38	139	105
28	178	91	70	67	114	36	269	30	68	85
29	140	33	191	94	31	89	42	49	262	96
30	87	63	44	11	9	17	99	94	125	19
31	158	13	50	26	273	90	22	68	246	91
32	45	90	83	48	255	99	192	82	129	114
33	145	95	223	28	2	103	147	12	295	30
34	207	46	155	87	22	28	221	62	87	119
35	64	3	216	111	89	94	16	110	72	17
36	184	24	192	79	90	27	182	31	332	9
37	4	66	30	28	240	15	188	54	226	1
38	56	4	160	49	50	35	73	6	232	86
39	173	52	54	57	74	101	159	99	94	80
40	101	50	146	32	20	58	256	97	307	23
41	8	112	69	41	236	92	21	34	67	99

Çizelge Ek.1 (devam) : Rastgele serpiştirme değışkenleri 1 ($I = 25200, I = 29160, I = 33120, I = 37080, I = 41040$).

	$I = 25200$		$I = 29160$		$I = 33120$		$I = 37080$		$I = 41040$	
	α	β	α	β	α	β	α	β	α	β
42	85	5	26	6	98	113	10	48	329	52
43	201	58	136	102	239	45	72	60	98	89
44	29	17	126	60	1	110	61	85	282	29
45	41	105	195	100	251	38	169	103	303	110
46	42	100	41	23	155	40	232	8	142	69
47	170	34	182	74	160	21	283	12	55	18
48	43	62	49	84	150	61	271	40	1	58
49	115	74	190	18	153	1	6	95	250	107
50	138	76	27	34	80	41	302	58	10	118
51	199	1	10	47	108	38	171	22	318	27
52	108	86	33	62	86	77	29	29	274	72
53	88	117	237	113	165	107	279	42	306	87
54	105	39	217	59	167	20	115	117	233	46
55	90	30	100	4	196	69	199	26	284	38
56	179	92	4	37	67	11	285	84	21	20
57	76	22	80	46	262	29	231	4	255	107
58	204	82	196	34	29	84	272	110	289	65
59	121	51	187	11	272	56	194	41	163	84
60	106	23	75	18	104	100	129	27	6	15
61	30	38	145	83	48	4	210	36	144	93
62	54	60	43	82	43	118	19	15	124	62
63	17	54	102	25	65	59	253	66	96	4
64	70	11	68	92	61	119	299	20	221	64
65	37	97	22	10	166	74	163	37	188	103
66	165	41	103	30	143	82	263	10	111	102
67	176	6	116	1	202	118	255	86	323	37
68	156	0	117	97	19	24	259	4	172	115
69	98	9	154	53	129	97	275	79	214	111
70	125	80	193	105	33	54	282	106	169	45
71	92	107	99	47	188	44	175	98	89	10
72	120	35	236	31	132	106	83	97	247	44
73	133	20	73	20	53	64	227	25	297	94
74	65	87	161	81	229	30	288	24	279	0
75	122	110	42	96	258	114	164	64	286	14
76	110	7	105	67	209	51	108	100	335	47
77	151	78	141	70	263	93	202	61	9	113
78	149	83	231	63	201	1	82	98	325	7
79	71	114	140	39	51	10	55	23	191	80
80	117	36	162	2	23	89	3	17	293	104
81	3	73	177	42	32	70	26	91	34	92
82	19	55	72	94	235	105	89	87	121	33
83	6	98	57	12	124	8	140	72	99	40
84	114	88	32	101	193	3	20	75	264	25
85	75	27	206	79	91	110	178	33	334	59

Çizelge Ek.1 (devam) : Rastgele serpiştirme değışkenleri 1 ($I = 25200, I = 29160, I = 33120, I = 37080, I = 41040$).

	$I = 25200$		$I = 29160$		$I = 33120$		$I = 37080$		$I = 41040$	
	α	β	α	β	α	β	α	β	α	β
86	52	81	29	58	46	16	40	71	86	48
87	113	95	3	75	88	52	291	63	185	70
88	11	43	222	45	106	62	53	55	136	32
89	188	15	204	61	24	101	293	74	338	31
90	53	77	58	118	181	57	17	99	14	33
91	57	103	202	7	242	115	28	82	39	36
92	109	70	137	42	146	18	209	1	339	2
93	128	42	45	104	271	6	251	73	107	78
94	157	118	135	54	118	47	176	52	61	16
95	127	116	15	52	170	66	289	108	160	105
96	195	85	31	93	127	108	134	101	90	53
97	152	11	2	88	257	55	65	50	187	17
98	96	12	11	39	156	7	243	35	78	64
99	160	104	48	103	205	33	206	18	230	63
100	47	3	62	65	55	25	154	92	73	8
101	51	53	241	98	47	22	56	87	104	95
102	58	96	178	118	151	21	27	112	305	15
103	196	68	1	116	221	32	184	53	340	102
104	171	4	94	5	119	13	94	83	197	119
105	209	100	23	68	16	48	75	116	265	75
106	60	68	198	96	26	80	119	22	84	25
107	46	106	111	30	95	84	69	46	174	116
108	182	8	166	115	82	4	35	28	299	34
109	169	29	215	103	249	82	284	53	80	22
110	63	61	91	42	122	55	127	91	312	44
111	135	113	127	17	241	39	250	5	37	59
112	194	64	158	0	111	53	123	30	202	86
113	192	70	171	110	176	65	260	68	320	106
114	118	32	229	27	208	75	143	16	278	67
115	69	58	159	81	260	103	240	14	152	115
116	111	77	221	23	213	87	204	41	47	13
117	7	10	67	1	190	100	85	115	182	97
118	147	116	85	49	131	26	225	77	231	20
119	153	59	194	109	72	71	274	62	313	75
120	136	40	77	73	149	64	303	38	341	57
121	2	45	65	72	18	81	70	96	162	91
122	187	3	150	38	35	112	36	34	301	66
123	25	30	153	21	252	79	0	88	123	99
124	163	52	142	119	10	96	38	10	281	104
125	23	90	9	33	270	104	112	30	259	26
126	80	28	149	59	139	5	290	117	175	50
127	12	115	181	45	141	90	179	81	133	1
128	112	49	180	61	110	45	233	76	183	11
129	91	16	19	56	158	42	128	77	33	35

Çizelge Ek.1 (devam) : Rastgele serpiştirme değışkenleri 1 ($I = 25200, I = 29160, I = 33120, I = 37080, I = 41040$).

	$I = 25200$		$I = 29160$		$I = 33120$		$I = 37080$		$I = 41040$	
	α	β	α	β	α	β	α	β	α	β
130	131	1	173	66	100	25	8	107	235	42
131	84	82	56	40	168	19	247	118	3	4
132	155	111	20	7	223	12	139	69	135	52
133	185	60	95	73	217	113	193	60	113	21
134	132	62	40	47	198	63	44	67	119	83
135	97	44	234	89	222	29	220	11	314	6
136	20	79	197	86	99	36	216	104	213	95
137	144	39	218	50	17	14	60	93	292	78
138	16	71	81	3	133	59	183	101	88	85
139	77	56	219	69	269	8	80	39	12	62
140	86	109	139	85	145	99	180	40	83	19
141	175	26	205	19	59	91	239	47	52	31
142	119	9	106	91	112	34	168	119	257	114
143	0	99	213	113	116	104	298	7	242	88
144	148	9	118	13	44	51	114	95	271	66
145	55	105	133	24	246	30	14	50	69	77
146	164	0	209	6	37	86	120	32	205	109
147	39	107	119	32	70	74	205	18	290	46
148	139	34	189	90	187	107	131	51	249	96
149	49	66	55	52	45	85	117	36	298	56
150	67	57	13	29	120	116	190	73	40	53
151	31	47	59	110	140	9	93	111	140	82
152	161	74	207	107	206	78	261	102	272	118
153	28	23	64	22	138	2	62	78	330	40
154	78	25	122	37	66	119	174	9	210	79
155	197	32	152	84	137	67	170	66	154	98
156	14	50	28	15	211	42	172	94	328	101
157	198	102	132	93	233	76	34	23	217	5
158	141	112	174	64	41	31	2	104	227	29
159	206	42	210	111	256	95	276	86	201	61
160	40	28	92	9	162	44	64	44	291	13
161	66	83	232	44	216	17	74	2	141	8
162	193	69	208	0	38	109	107	60	236	43
163	15	27	224	14	56	55	213	63	74	90
164	35	94	96	70	34	49	102	100	126	99
165	183	95	61	95	36	37	304	55	77	81
166	82	97	79	80	185	35	101	87	27	37
167	166	91	125	114	182	115	155	19	13	42
168	73	29	114	25	234	83	201	3	149	76
169	205	5	101	48	212	58	76	57	54	110
170	202	21	76	112	93	118	158	51	245	45
171	48	50	86	62	175	98	110	79	43	23
172	189	113	203	3	54	0	13	65	326	65
173	79	15	108	78	107	68	226	46	19	71

Çizelge Ek.1 (devam) : Rastgele serpiştirme değışkenleri 1 ($I = 25200, I = 29160, I = 33120, I = 37080, I = 41040$).

	$I = 25200$		$I = 29160$		$I = 33120$		$I = 37080$		$I = 41040$	
	α	β	α	β	α	β	α	β	α	β
174	174	37	238	16	226	78	195	59	304	55
175	24	35	84	53	220	28	258	93	311	34
176	34	59	147	63	87	57	161	108	8	21
177	36	42	134	8	64	20	48	94	331	69
178	181	63	25	57	97	35	79	117	36	29
179	159	103	63	102	142	61	71	113	120	94
180	59	14	14	22	79	22	25	84	116	84
181	13	40	37	75	245	77	121	112	155	88
182	186	73	201	0	25	18	217	40	283	9
183	129	89	124	38	121	52	24	106	11	116
184	123	38	239	106	125	93	167	70	333	94
185	94	93	212	89	148	116	135	12	23	46
186	124	65	74	105	126	80	207	32	240	83
187	203	86	16	13	244	47	297	61	156	74
188	99	2	228	20	207	106	305	68	254	28
189	9	21	36	18	230	117	54	0	49	0
190	126	67	167	91	40	49	238	14	252	93
191	89	13	143	66	0	0	257	18	50	20
192	95	54	89	34	8	73	46	56	209	117
193	93	116	5	26	250	82	223	26	319	71
194	172	31	121	50	268	112	173	98	207	59
195	208	85	184	11	210	84	266	29	277	57
196	107	46	242	69	11	89	228	31	158	12
197	27	85	138	100	62	101	31	68	62	100
198	18	81	211	97	83	95	236	20	239	108
199	72	7	164	77	224	45	241	24	322	23
200	5	11	144	58	194	27	98	5	164	103
201	150	98	110	79	237	15	67	1	302	68
202	81	84	104	104	27	10	23	37	171	27
203	116	3	226	98	58	113	224	82	138	69
204	177	119	24	6	5	110	90	85	35	47
205	143	114	131	43	171	59	292	2	18	57
206	146	110	38	76	179	4	214	88	75	49
207	62	58	168	71	261	28	142	4	176	1
208	137	27	34	60	57	72	104	72	324	63
209	44	45	230	45	77	91	211	8	241	111
210	50	32	51	74	6	3	296	5	206	38
211			71	28	113	32	246	74	56	18
212			18	32	105	75	96	63	321	77
213			113	117	183	94	167	43	223	97
214			123	59	85	103	178	98	263	33
215			200	44	96	17	41	84	92	112
216			21	55	227	24	12	76	273	81
217			227	83	84	40	219	47	109	51

Çizelge Ek.1 (devam) : Rastgele serpiştirme değışkenleri 1 ($I = 25200, I = 29160, I = 33120, I = 37080, I = 41040$).

	$I = 25200$		$I = 29160$		$I = 33120$		$I = 37080$		$I = 41040$	
	α	β	α	β	α	β	α	β	α	β
218			53	81	253	41	306	90	300	104
219			148	4	204	71	252	54	30	45
220			172	51	218	61	148	83	91	72
221			19	113	109	13	150	13	76	114
222			107	40	254	65	273	16	296	60
223			115	41	117	86	50	36	216	37
224			82	97	248	107	87	101	251	22
225			185	10	152	54	152	14	146	35
226			93	1	71	23	92	105	4	73
227			130	17	69	5	280	66	180	23
228			120	31	68	102	295	118	93	48
229			46	64	78	9	151	25	66	85
230			12	27	215	108	215	77	16	40
231			163	30	12	19	191	58	170	84
232			176	102	184	60	37	75	179	4
233			169	107	178	33	249	44	285	26
234			98	39	225	63	106	33	24	58
235			7	25	180	92	116	102	70	59
236			39	77	266	30	245	64	112	117
237			66	105	186	56	146	35	248	43
238			175	87	264	106	149	21	192	108
239			90	52	199	45	51	73	168	107
240			6	57	265	63	109	99	15	90
241			78	15	164	6	286	41	97	15
242			220	64	174	64	58	3	229	8
243					76	117	125	80	128	14
244					13	1	100	110	211	109
245					14	11	45	70	145	42
246					28	74	68	89	196	63
247					243	25	18	48	150	96
248					191	119	1	34	151	3
249					136	3	230	27	65	48
250					231	88	248	109	131	6
251					30	37	43	91	115	67
252					130	42	33	64	25	10
253					128	79	198	114	215	31
254					60	82	200	15	161	17
255					39	100	157	85	195	27
256					103	118	47	95	101	54
257					267	31	86	20	48	78
258					218	63	95	17	17	65
259					81	6	4	71	51	64
260					123	64	237	22	316	1
261					15	117	97	94	186	106

Çizelge Ek.1 (devam) : Rastgele serpiştirme değışkenleri 1 ($I = 25200, I = 29160, I = 33120, I = 37080, I = 41040$).

	$I = 25200$		$I = 29160$		$I = 33120$		$I = 37080$		$I = 41040$	
	α	β	α	β	α	β	α	β	α	β
262			154	1	81	72	219	36		
263			75	11	262	114	71	76		
264			192	74	113	87	280	15		
265			159	25	118	57	165	28		
266			163	119	84	6	184	62		
267			247	3	63	115	189	32		
268			52	88	138	102	327	77		
269			147	37	136	113	132	2		
270			203	42	7	59	294	20		
271			169	79	137	46	85	68		
272			157	82	287	93	105	118		
273			177	100	122	28	269	95		
274			42	118	229	41	38	50		
275			4	31	242	92	127	54		
276					197	34	22	95		
277					49	10	275	53		
278					277	24	288	64		
279					307	1	204	93		
280					66	47	81	10		
281					144	116	28	30		
282					160	96	279	11		
283					294	67	2	72		
284					59	49	117	109		
285					57	75	134	39		
286					212	42	218	98		
287					105	52	26	66		
288					234	30	44	47		
289					5	39	267	61		
290					166	103	53	86		
291					177	107	148	69		
292					254	17	237	13		
293					153	7	79	114		
294					185	81	82	33		
295					132	23	337	29		
296					133	75	309	104		
297					268	8	244	39		
298					11	38	45	116		
299					91	62	225	79		
300					187	27	110	25		
301					111	104	157	113		
302					244	12	31	55		
303					124	55	220	5		
304					281	61	234	102		
305					308	56	41	17		

Çizelge Ek.1 (devam) : Rastgele serpiştirme değışkenleri 1 ($I = 25200, I = 29160, I = 33120, I = 37080, I = 41040$).

	$I = 25200$		$I = 29160$		$I = 33120$		$I = 37080$		$I = 41040$	
	α	β	α	β	α	β	α	β	α	β
306					77	118	268	12		
307					222	40	0	16		
308					39	97	203	89		
309							95	5		
310							212	105		
311							222	35		
312							199	9		
313							137	41		
314							194	82		
315							336	107		
316							276	91		
317							147	68		
318							266	4		
319							58	75		
320							57	115		
321							310	38		
322							200	81		
323							63	110		
324							228	37		
325							261	117		
326							100	50		
327							103	93		
328							287	59		
329							59	42		
330							166	57		
331							159	40		
332							60	21		
333							20	58		
334							42	36		
335							114	66		
336							256	42		
337							32	15		
338							102	56		
339							243	13		
340							153	32		
341							5	102		

Çizelge Ek.2 : Rastgele serpiştirme değışkenleri 2 ($I = 24840$, $I = 32760$, $I = 36720$).

	$I = 24840$		$I = 32760$		$I = 36720$	
	α	β	α	β	α	β
0	50	32	93	63	30	57
1	134	71	10	238	165	114
2	200	47	78	92	156	65
3	68	89	72	189	235	67
4	74	15	25	73	15	19
5	1	18	68	172	126	90
6	32	40	96	94	203	0
7	102	69	14	49	270	24
8	33	79	6	228	189	15
9	154	119	5	218	103	78
10	21	29	43	197	145	36
11	22	111	72	173	301	43
12	103	61	53	232	186	87
13	162	108	50	195	264	63
14	130	54	49	144	218	78
15	83	16	65	21	52	21
16	38	99	89	134	265	32
17	190	105	65	161	141	85
18	180	55	76	135	181	21
19	61	116	57	102	208	92
20	104	75	37	259	9	80
21	167	114	60	101	88	96
22	168	21	87	200	196	118
23	100	101	70	3	32	114
24	191	81	12	7	78	38
25	26	19	36	115	130	30
26	142	67	89	114	162	49
27	10	43	17	31	300	94
28	178	91	90	9	269	68
29	140	33	99	255	42	82
30	87	63	103	2	99	12
31	158	13	28	22	22	62
32	45	90	94	89	192	110
33	145	95	27	90	147	31
34	64	46	15	240	221	54
35	184	3	35	50	16	6
36	4	24	101	74	182	99
37	56	66	58	20	188	97
38	173	4	92	236	73	34
39	101	52	113	98	159	48
40	8	50	45	239	256	60
41	85	112	110	1	21	85
42	201	5	38	251	10	103
43	29	58	40	155	72	8

Çizelge Ek.2 (devam) : Rastgele serpiştirme değışkenleri 2 ($I = 24840$,
 $I = 32760$, $I = 36720$).

	$I = 24840$		$I = 32760$		$I = 36720$	
	α	β	α	β	α	β
44	41	17	21	160	61	12
45	42	105	61	150	169	40
46	170	100	1	153	232	95
47	43	34	41	80	283	58
48	115	62	38	108	271	22
49	138	74	77	86	6	29
50	199	76	107	165	302	42
51	108	1	20	167	171	117
52	88	86	69	196	29	26
53	105	117	11	67	279	84
54	90	39	29	262	115	4
55	179	30	84	29	199	110
56	76	92	56	272	285	41
57	204	22	100	104	231	27
58	121	82	4	48	272	36
59	106	51	118	43	194	15
60	30	23	59	65	129	66
61	54	38	119	61	210	20
62	17	60	74	166	19	37
63	70	54	82	143	253	10
64	37	11	118	202	299	86
65	165	97	24	19	163	4
66	176	41	97	129	263	79
67	156	6	54	33	255	106
68	98	0	44	188	259	98
69	125	9	106	132	275	97
70	92	80	64	53	282	25
71	120	107	30	229	175	24
72	133	35	114	258	83	64
73	65	20	51	209	227	100
74	122	87	93	263	288	61
75	110	110	1	201	164	98
76	151	7	10	51	108	23
77	149	78	89	23	202	17
78	71	83	70	32	82	91
79	117	114	105	235	55	87
80	3	36	8	124	3	72
81	19	73	3	193	26	75
82	6	55	110	91	89	33
83	114	98	16	46	140	71
84	75	88	52	88	20	63
85	52	27	62	106	178	55
86	113	81	101	24	40	74
87	11	95	57	181	291	99

Çizelge Ek.2 (devam) : Rastgele serpiştirme deęişkenleri 2 ($I = 24840$,
 $I = 32760$, $I = 36720$).

	$I = 24840$		$I = 32760$		$I = 36720$	
	α	β	α	β	α	β
88	188	43	115	242	53	82
89	53	15	18	146	293	1
90	57	77	6	271	17	73
91	109	103	47	118	28	52
92	128	70	66	170	209	108
93	157	42	108	127	251	101
94	127	118	55	257	176	50
95	195	116	7	156	289	35
96	152	85	33	205	134	18
97	96	11	25	55	65	92
98	160	12	22	47	243	87
99	47	104	21	151	206	112
100	51	3	32	221	154	53
101	58	53	13	119	56	83
102	196	96	48	16	27	116
103	171	68	80	26	184	22
104	60	4	84	95	94	46
105	46	100	4	82	75	28
106	182	68	82	249	119	53
107	169	106	55	122	69	91
108	63	8	39	241	35	5
109	135	29	53	111	284	30
110	194	61	65	176	127	68
111	192	113	75	208	250	16
112	118	64	103	260	123	14
113	69	77	87	213	260	41
114	111	10	100	190	143	115
115	7	116	26	131	240	77
116	147	59	71	72	204	62
117	153	40	64	149	85	38
118	136	45	81	18	225	96
119	2	3	112	35	274	34
120	187	30	79	252	303	88
121	25	52	96	10	70	10
122	163	90	104	270	36	30
123	23	28	5	139	0	117
124	80	115	90	141	38	81
125	12	49	45	110	112	76
126	112	16	42	158	290	77
127	91	1	25	100	179	107
128	131	82	19	168	233	118
129	84	111	12	223	128	69
130	155	60	113	217	8	60
131	185	62	63	198	247	67

Çizelge Ek.2 (devam) : Rastgele serpiştirme değışkenleri 2 ($I = 24840$,
 $I = 32760$, $I = 36720$).

	$I = 24840$		$I = 32760$		$I = 36720$	
	α	β	α	β	α	β
132	132	44	29	222	139	11
133	97	79	36	99	193	104
134	20	39	14	17	44	93
135	144	71	59	133	220	101
136	16	56	8	269	216	39
137	77	109	99	145	60	40
138	86	26	91	59	183	47
139	175	9	34	112	80	119
140	119	99	104	116	180	7
141	0	9	51	44	239	95
142	148	105	30	246	168	50
143	55	0	86	37	298	32
144	164	107	74	70	114	18
145	39	34	107	187	14	51
146	139	66	85	45	120	36
147	49	57	116	120	205	73
148	67	47	9	140	131	111
149	31	74	78	206	117	102
150	161	23	2	138	190	78
151	28	25	119	66	93	9
152	78	32	67	137	261	66
153	197	50	42	211	62	94
154	14	102	76	233	174	23
155	198	112	31	41	170	104
156	141	42	95	256	172	86
157	206	28	44	162	34	44
158	40	83	17	216	2	2
159	66	69	109	38	276	60
160	193	27	55	56	64	63
161	15	94	49	34	74	100
162	35	95	37	36	107	55
163	183	97	35	185	213	87
164	82	91	115	182	102	19
165	166	29	83	234	304	3
166	73	5	58	212	101	57
167	205	21	118	93	155	51
168	202	50	98	175	201	79
169	48	113	0	54	76	65
170	189	15	68	107	158	46
171	79	37	78	226	110	59
172	174	35	28	220	13	93
173	24	59	57	87	226	108
174	34	42	20	64	195	94
175	36	63	35	97	258	117

Çizelge Ek.2 (devam) : Rastgele serpiştirme deęişkenleri 2 ($I = 24840$,
 $I = 32760$, $I = 36720$).

	$I = 24840$		$I = 32760$		$I = 36720$	
	α	β	α	β	α	β
176	181	103	61	142	161	113
177	159	14	22	79	48	84
178	59	40	77	245	79	112
179	13	73	18	25	71	40
180	186	89	52	121	25	106
181	129	38	93	125	121	70
182	123	93	116	148	217	12
183	94	65	80	126	24	32
184	124	86	47	244	167	61
185	203	2	106	207	135	68
186	99	21	117	230	207	0
187	9	67	49	40	297	14
188	126	13	0	0	305	18
189	89	54	73	8	54	56
190	95	116	82	250	238	26
191	93	31	112	268	257	98
192	172	85	84	210	46	29
193	107	46	89	11	223	31
194	27	85	101	62	173	68
195	18	81	95	83	266	20
196	72	7	45	224	228	24
197	5	11	27	194	31	5
198	150	98	15	237	236	1
199	81	84	10	27	241	37
200	116	3	113	58	98	82
201	177	119	110	5	67	85
202	143	114	59	171	23	2
203	146	110	4	179	224	88
204	62	58	28	261	90	4
205	137	27	72	57	292	72
206	44	45	91	77	214	8
207			3	6	142	5
208			32	113	104	74
209			75	105	211	63
210			94	183	296	43
211			103	85	246	98
212			17	96	96	84
213			24	227	167	76
214			40	84	178	47
215			41	253	41	90
216			71	204	12	54
217			61	218	219	83
218			13	109	252	13
219			65	254	148	16

Çizelge Ek.2 (devam) : Rastgele serpiştirme değışkenleri 2 ($I = 24840$,
 $I = 32760$, $I = 36720$).

	$I = 24840$		$I = 32760$		$I = 36720$	
	α	β	α	β	α	β
220			86	117	150	36
221			107	248	273	101
222			54	152	50	14
223			23	71	87	105
224			5	69	152	66
225			102	68	92	118
226			9	78	280	25
227			108	215	295	77
228			19	12	151	58
229			60	184	215	75
230			33	178	191	44
231			63	225	37	33
232			92	180	249	102
233			30	266	106	64
234			56	186	116	35
235			106	264	245	21
236			45	199	146	73
237			63	265	149	99
238			6	164	51	41
239			64	174	109	3
240			117	76	286	80
241			1	13	58	110
242			11	14	125	70
243			74	28	100	89
244			25	243	45	48
245			119	191	68	34
246			3	136	18	27
247			88	231	1	109
248			37	30	230	91
249			42	130	248	64
250			79	128	43	114
251			82	60	33	15
252			100	39	198	85
253			118	103	200	95
254			31	267	157	20
255			63	218	47	17
256			6	81	86	71
257			64	123	95	22
258			117	15	4	94
259			1	154	237	72
260			11	75	97	114
261			74	192	81	87
262			25	159	262	57
263			119	163	113	6

Çizelge Ek.2 (devam) : Rastgele serpiştirme deęişkenleri 2 ($I = 24840$,
 $I = 32760$, $I = 36720$).

	$I = 24840$		$I = 32760$		$I = 36720$	
	α	β	α	β	α	β
264			3	247	118	115
265			88	52	84	102
266			37	147	63	113
267			42	203	138	59
268			79	169	136	46
269			82	157	7	93
270			100	177	137	28
271			118	42	287	41
272			31	4	122	92
273			93	63	229	34
274					242	10
275					197	24
276					49	1
277					277	47
278					66	116
279					144	96
280					160	67
281					294	49
282					59	75
283					57	42
284					212	52
285					105	30
286					234	39
287					5	103
288					166	107
289					177	17
290					254	7
291					153	81
292					185	23
293					132	75
294					133	8
295					268	38
296					11	62
297					91	27
298					187	104
299					111	12
300					244	55
301					124	61
302					281	56
303					77	118
304					222	40
305					39	97

ÖZGEÇMİŞ

Ad-Soyad : Erkan İnceöz
Uyruğu : T.C.
Doğum Tarihi ve Yeri : 11.06.1993- Ankara
E-posta : erkan.inceoz@tubitak.gov.tr,erkaninceoz@gmail.com

ÖĞRENİM DURUMU:

Lisans : 2016, Ankara Yıldırım Beyazıt Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Elektrik-Elektronik Mühendisliği

MESLEKİ DENEYİM VE ÖDÜLLER:

Yıl	Yer	Görev
2014-2015	Tera-Mikroelektronik	Sayısal Tasarım Müh. (Kısmi Zamanlı)
2016-2016	TÜBİTAK UZAY	Sayısal Tasarım Müh. (Kısmi Zamanlı)
2016-2020	TÜBİTAK UZAY	Sayısal Tasarım Müh.
2020-	TÜBİTAK UZAY	Uzman Sayısal Tasarım Müh.

YABANCI DİL: İngilizce

TEZDEN TÜRETİLEN YAYINLAR, SUNUMLAR VE PATENTLER:

- **İnceöz. E.**, Tutgun, R., Yüksel Turgut, A.M., (2020). “FPGA Based Transmitter Design Using Adaptive Coding and Modulation Schemes for Low Earth Orbit Satellite Communications”, *IEEE 5th International Symposium on Telecommunication Technologies*.

DİĞER YAYINLAR, SUNUMLAR VE PATENTLER:

- Çağlan, A., **İnceöz, E.**, Balcısoy, E., Özbek, M.E., Çavuş, E., (2016) “Box-Muller Metodu ile Toplanır Beyaz Gauss Gürültü (AWGN) Üreticinin FPGA Üzerinde Gerçeklenmesi”, *Sinyal İşleme ve İletişim Uygulamaları Kurultayı*, Zonguldak, Türkiye.
- **İnceöz. E.**, Enver Çavuş, (2017) “FPGA Implementation of Variable-Length Split-Radix FFT Algorithm”, *International Journal of Engineering Science and Computing*, Vol.7, No.7, pp. 13977- 13980.