

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**İKİNCİ NESİL SAYISAL VİDEO YAYINI (DVB-S2) İLERİ HATA KODLAMA
BİRİMİ TASARIMI VE GERÇEKLEMESİ**

YÜKSEK LİSANS TEZİ

Şakir BALTA

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Telekomünikasyon Mühendisliği Programı

EKİM 2013

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**İKİNCİ NESİL SAYISAL VİDEO YAYINI (DVB-S2) İLERİ HATA KODLAMA
BİRİMİ TASARIMI VE GERÇEKLEMESİ**

YÜKSEK LİSANS TEZİ

**Şakir BALTA
(504101324)**

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Telekomünikasyon Mühendisliği Programı

Tez Danışmanı: Doç. Dr. Mesut KARTAL

EKİM 2013

İTÜ, Fen Bilimleri Enstitüsü'nün 504101324 numaralı Yüksek Lisans Öğrencisi **Şakir BALTA**, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “**İKİNCİ NESİL SAYISAL VIDEO YAYINI (DVB-S2) İLERİ HATA KODLAMA BİRİMİ TASARIMI VE GERÇEKLEMESİ**” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Doç. Dr. Mesut KARTAL**
İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Prof. Dr. Tülay YILDIRIM**
Yıldız Teknik Üniversitesi

Doç. Dr. S. Berna ÖRS YALÇIN
İstanbul Teknik Üniversitesi

Teslim Tarihi : **17 Eylül 2013**
Savunma Tarihi : **07 Ekim 2013**

Aileme,

ÖNSÖZ

Yüksek lisans eğitimim boyunca ve bu tezin hazırlanması süresince desteğini hiçbir zaman eksik etmeyen, Telekomünikasyon Mühendisliği alanında bana kazandırmış olduğu bilinçten dolayı Doç. Dr. Mesut KARTAL'a sonsuz saygı ve şükranlarımı sunarım. Bu çalışmanın gerçekleşmesi boyunca her türlü yardımını esirgemeyen Seyyid M.Dilek'e çok teşekkür ederim. İyi bir eğitim almamda bana hayatım boyunca destek olan aileme ve emeği geçen değerli hocalarıma da şükranlarımı sunarım.

Ekim 2013

Şakir Balta
(Elektronik Mühendisi)

İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	vii
İÇİNDEKİLER	ix
KISALTMALAR	xi
ÇİZELGE LİSTESİ.....	xiii
ŞEKİL LİSTESİ.....	xv
ÖZET.....	xvii
SUMMARY	xix
1. GİRİŞ	1
2. İKİNCİ NESİL SAYISAL VIDEO YAYINI (DVB-S2)	3
2.1 Dış Kod: DVB-S2 de BCH Kodu	4
2.2 İç Kod: DVB-S2 de LDPC Kodu	4
3. BOSE CHAUDHURI AND HOCQUENGHEM (BCH) KODLARI	7
3.1 Minimal Polinomlar	10
3.1.1 GF(2 ^m) 'de Aritmetik İşlemler	14
3.1.1.1 GF(2 ^m)'de Toplama/Çıkarma İşlemi	14
3.1.1.2 GF(2 ^m)'de Çarpma İşlemi.....	14
3.1.1.3 GF(2 ^m)'de Çarpma İşlemi Donanım İmplementasyonu	14
3.2 BCH Kodları	15
3.2.1 Tasarlanmış Mesafe	16
3.2.2 BCH Kodunun Kuruluşu.....	17
3.2.3 n=63 İçin Tüm BCH Kodlarının Oluşturulması	20
3.2.4 İki-Hata Düzeltten BCH Kodunu Çözmek	25
3.2.5 GF(2 ¹⁴)'ün Elemanları.....	30
3.2.6 GF(2 ¹⁶)'nın Elemanları	30
4. LDPC (Low Density Parity Check) KODLARI	33
4.1 LDPC Kodların Kodlanması	33
4.2 LDPC Kodları Çözülmesi	35
4.3 Tanner Grafi	39
4.4 Mesaj Aktarma Kod Çözücü Algoritması	41
4.5 Olasılık Bölgesi Kod Çözücüsü	43
4.6 Mesaj Aktarma Algoritmasının Özeti	46
4.6.1 Örnek Uygulama	47
4.7 Kod Çözme Hatasının Analizi.....	50
4.7.1 Uzaklık Fonksiyonları.....	53
4.7.2 LDPC'nin Uzaklık Özellikleri.....	53
4.7.3 Kod Çözücü Hata Olasılığı	58
4.7.4 Simetrik İkili Giriş Kanalları	58
4.7.5 Kod Çözme Hatasının Olasılığının Üst Sınırı	59
4.7.6 Kodlar İçin Ortalama Hata Terimi	61
5. BCH KOD ÇÖZÜCÜ GERÇEKLEMESİ.....	67
5.1 FPGA Modül Yapıları	67

5.1.1 bch_decoder Modülü.....	67
5.1.2 bch_receiver Modülü.....	69
5.1.3 bch_syndrome_calc Modülü	69
5.1.4 ribm_short Modülü.....	71
5.1.5 chien_search_short Modülü	75
5.2 FPGA Benzetim Sonuçları	77
6. LDPC KOD ÇÖZÜCÜ GERÇEKLEMESİ.....	81
6.1 FPGA Modül Yapıları	83
6.1.1 ldpc_decoder Modülü.....	83
6.1.2 ldpc_ram_behav Modülü.....	88
6.1.3 ldpc_shuffle Modülü	88
6.1.4 ldpc_cn Modülü.....	90
6.1.5 ldpc_vn Modülü	91
6.1.6 ldpc_iocontrol Modülü.....	92
6.2 FPGA Benzetim Sonuçları	93
7. SONUÇ VE ÖNERİLER.....	97
KAYNAKLAR.....	99
EKLER.....	105
ÖZGEÇMİŞ.....	111

KISALTMALAR

DVB	: Sayısal Video Yayını
DVB-S2	: İkinci Nesil Sayısal Video Yayını - Uydu
ITU	: Uluslararası Telekomünikasyon Birliği
BCH	: Bose-Chaudri-Hocquenghem kodları
LDPC	: Low Density Parity Check
BP	: Belief Propagation
LLR	: Log-Likelihood Ratios
GF	: Sonlu Alan
FEC	: İleri Hata Düzeltme
VHDL	: VHSIC Donanım Tanımlama Dili
FPGA	: Sahada Programlanabilen Kapı Dizileri
MODELSIM	: VHDL Benzetim Aracı
$F_q, GF(q)$	: Mertebesi $q = p^h$ (p asal, h pozitif tam sayı) olan Galois cismi
KES	: Anahtar Denklem Çözücü (Key Equation Solver)
$F_q(x)$	: Katsayıları F_q da olan polinomlar halkası
F_q^*	: F_q sonlu cisminin sıfırdan farklı elemanlarının çarpım grubu
F_{q^m}	: $F[x]$ deki m dereceli indirgenemez f polinomunun q F üzerinde parçalanış cismi
$F^m[x]$	: $F[x]$ deki derecesi m den daha küçük olan tüm polinomların kümesi
F	: F cisminin eleman sayısı
f_t	: F_q üzerinde $\alpha^t \in F_{q^m}$ nin karakteristik polinomu
G^t	: G üreteç matrisinin transpozisi
Ham(r, 2)	: İkili (binary) Hamming kodu
RS	: Reed-Solomon kodu
ESA	: Avrupa Uzay Ajansı

ÇİZELGE LİSTESİ

Sayfa

Çizelge 2.1 : DVB-S2 data uzunlukları.	5
Çizelge 3.1 : $GF(2^4)$ elemanları.	10
Çizelge 3.2 : $GF(2^4)$ elemanları minimal polinomları.	13
Çizelge 3.3 : $GF(2^6)$ elemanları.	20
Çizelge 3.4 : $GF(2^6)$ elemanları minimal polinomları.	22
Çizelge 3.5 : $n=63$ uzunluklu ikili primitif BCH kod parametreleri.	22
Çizelge 3.6 : $GF(2^{14})$ elemanları.	30
Çizelge 3.7 : $GF(2^{16})$ elemanları.	30
Çizelge 5.1 : KES bloğu donanımsal olarak karşılaştırılması.	72
Çizelge 6.1 : Mode değerleri.	93

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 : FEC yapısı.	4
Şekil 3.1 : $GF(2^m)$ 'de hücresel-array çarpıcı blok şeması.	15
Şekil 3.2 : $GF(2^m)$ ' de hücresel-array çarpıcının hücre yapısı.	15
Şekil 4.1 : G üreteç matrisi.	34
Şekil 4.2 : H matrisi gösterilimi.	35
Şekil 4.3 : (10,5) kodu H matrisi.	39
Şekil 4.4 : (10,5) kodu H matrisine ait Tanner grafi.	40
Şekil 4.5 : (7,4) kodu H matrisi.	41
Şekil 4.6 : (7,4) kodu H matrisine ait Tanner grafi.	41
Şekil 4.7 : Değiştirilmiş Tanner grafi.	41
Şekil 4.8 : H'n ilk sütununa ilişkin alt graf.	42
Şekil 4.9 : $X_0 \rightarrow F_2$ için graf.	42
Şekil 4.10 : $(F_j \rightarrow X_i)$ ile ilgili alt graf.	43
Şekil 4.11 : $F_0 \rightarrow X_4$ durumu.	43
Şekil 4.12 : (8,4) kodu H matrisi.	47
Şekil 4.13 : (8,4) kodu H matrisine ait Tanner grafi.	48
Şekil 4.14 : R_{00} durumu.	49
Şekil 4.15 : q_{00} gösterilimi.	50
Şekil 4.16 : Genel haberleşme şeması.	51
Şekil 4.17 : Eşlik kontrol matrisine ilişkin kontrol denklemleri.	52
Şekil 4.18 : (10,3,4) kodu H matrisi gösterilimi.	54
Şekil 4.19 : $B_{jk}(\lambda)$ 'nın dağılımı.	56
Şekil 4.20 : Tipik (n,j,k) için minimum uzaklığın blok uzunluğuna oranı.	57
Şekil 4.21 : BSC için üstel ifadenin geometrik kestirimi.	64
Şekil 5.1 : DVB-S2 BCH FPGA tasarımı hierarşisi.	67
Şekil 5.2 : bch_decoder modülü RTL şeması.	67
Şekil 5.3 : DVB-S2 BCH kod çözücü blok yapısı.	68
Şekil 5.4 : bch_receiver modülü RTL şeması.	69
Şekil 5.5 : bch_syndrome_calc modülü hierarşisi.	69
Şekil 5.6 : Hata belirteci hücresi.	70
Şekil 5.7 : ribm_short modülü hierarşisi.	71
Şekil 5.8 : RiBM devre şeması.	72
Şekil 5.9 : PE1 devresinin yapısı.	73
Şekil 5.10 : Kontrol devresinin yapısı.	73
Şekil 5.11 : ribm_short modülü RTL şeması.	74
Şekil 5.12 : chien_search_short modülü hierarşisi.	75
Şekil 5.13 : Chien araştırma hücresi.	75
Şekil 5.14 : csee_cell_short modülü RTL şeması.	76
Şekil 5.15 : $t=8$ için Chien araştırma yapısı.	76
Şekil 5.16 : chien_search_short modülü RTL şeması.	77

Şekil 5.17 : GF çarpıcı benzetimi.....	77
Şekil 5.18 : GF(2^m) çarpıcı benzetimi.....	78
Şekil 5.19 : Syndrome hesaplayıcı benzetimi.....	79
Şekil 5.20 : RIBM benzetimi.....	80
Şekil 6.1 : Min-Sum BP yaklaşımı ile ideal BP karşılaştırması.....	82
Şekil 6.2 : DVB-S2 LDPC FPGA tasarımı hierarşisi.....	83
Şekil 6.3 : ldpc_decoder modülü RTL şeması.....	83
Şekil 6.4 : DVB-S2 LDPC kod çözücü blok yapısı.....	84
Şekil 6.5 : ldpc_decoder modülü iç yapısı RTL şeması.....	84
Şekil 6.6 : ldpc_decoder blok şeması.....	86
Şekil 6.7 : VN RAM organizasyonu.....	86
Şekil 6.8 : CN RAM organizasyonu.....	87
Şekil 6.9 : Karar verilen VN RAM organizasyonu.....	87
Şekil 6.10 : ldpc_ram_behav modülü RTL şeması.....	88
Şekil 6.11 : ldpc_shuffle modülü hierarşisi.....	88
Şekil 6.12 : ldpc_shuffle modülü RTL şeması.....	89
Şekil 6.13 : ldpc_cn modülü RTL şeması.....	90
Şekil 6.14 : ldpc_vn modülü RTL şeması.....	91
Şekil 6.15 : ldpc_iocontrol modülü RTL şeması.....	92
Şekil 6.16 : ldpc_decoder modülü benzetimi.....	94
Şekil 6.17 : io_control modülü benzetimi.....	95
Şekil 6.18 : ldpc_shuffle modülü benzetimi.....	95
Şekil 6.19 : ldpc_cn modülü benzetimi.....	96
Şekil 6.20 : ldpc_vn modülü benzetimi.....	96

İKİNCİ NESİL SAYISAL VIDEO YAYINI (DVB-S2) İLERİ HATA KODLAMA BİRİMİ TASARIMI VE GERÇEKLEMESİ

ÖZET

Tez çalışması, uydu haberleşme sistemleri aracılığıyla yapılan sayısal video yayınlarının alıcı tarafında kullanılan ileri hata düzeltme birimi tasarımı ve gerçeklemesini kapsamaktadır. Bu kapsamda Avrupa Telekomünikasyon Standardı Enstitüsü (ETSI) tarafından İkinci Nesil Sayısal Video Yayını (DVB-S2) için belirtilen standartlar göz önüne alınmıştır. Alıcı tarafı DVB-S2 ileri hata kod çözücü birimi standartlar incelenerek ve daha önceki çalışmalar dikkate alınarak tasarlanmış ve VHDL kodları yazılarak donanımsal olarak gerçekleştirilmiş MODELSIM yardımıyla benzetimleri yapılmış ve MATLAB yardımıyla da doğrulanmıştır.

DVB-S2 gibi yüksek hız ve büyük uzunluktaki veriler için yüksek başarımlı sağlayan ve hızlı, esnek bir yapı üzerinde durularak oldukça kullanışlı ve karmaşıklıktan uzak bir tasarım gerçekleştirilmiştir. Literatürde geçen bir çok farklı yaklaşım incelenmiş ve en uygun modellerde karar kılınarak mümkün olduğunca hızlı olması için gerekli paralelleştirmeler yapılmıştır. DVB-S2 standartlarında uydu sistemleri dijital video yayını için kullanılacak ileri hata düzeltme birimi, daha hızlı ve daha yüksek hata düzeltme performansı sağlamak amacıyla ardışıl kodlama tekniğini içermektedir. Bu kodlama türü, klasik kodlama türlerinin seri bir şekilde ard arda bağlanması ve bir kodlama türünden çıkan verinin diğeri için girdi kabul edilmesiyle oluşturulmaktadır.

DVB-S2 için ardışıl kodlama yapısı iç kodlama ve dış kodlama birimlerinden oluşmaktadır. Dış kod olarak BCH(Bose-Chaudhri-Hocquenghem) kodu iç kod olarak LDPC (Low Density Parity Check) kodu kullanılmıştır.

BCH kod çözücüsü tasarımında hızlı aynı zamanda kapladığı alan açısından da az ve esnek bir yapı oluşturulmuştur. Dışarıdan set edilebilen parametrelerle değişik veri uzunluklarına uyumlu bir tasarım yapılmıştır. Hata belirtici(syndrome) hesaplamada paralel GF çarpıcıları kullanılarak hızlı olması sağlanmıştır. Anahtar eşitlik çözücüde gecikmenin ve kritik yol beklemesinin(Critical path delay) en az olması için RiBM (Reformulated inverse-free Berlekamp-Massey) algoritması kullanılmıştır. Fifo büyüklükleri dışarıdan set edilebilecek şekilde esnek yapılmıştır.

LDPC kod çözücü yapısında hızlı haberleşme sistemleri için ideal olan belief propagation yöntemi seçilmiş literatürden farklı olarak karmaşıklığı ve kapladığı alanı azaltmak amacıyla min-sum algoritması kullanılmıştır. Tekrar hız açısından verimli olması amacıyla düzenli H matrisi güvenilir bir kaynaktan kenar tablosu alınarak oluşturulmuştur. İterasyon sayısı, mod kodu, veri uzunlukları dışarıdan girilebilecek şekilde yapılarak hertürlü(yer ya da hız) kısıt için optimum ayarlanabilecek bir tasarım oluşturulmuştur. Gerek hız gerek yer açısından bit ve kontrol noktaları RAM leri 360 bitlik gruplara ayrılarak kısmi paralel bir yapı oluşturulmuştur. Hesaplanan LLR değerlerinden sadece 2 si tutularak literatürdekilere göre yerden kazanç sağlanmıştır. Yumuşak karar LDPC kodu kullanılarak yerden tasarruf sağlanmış kodlama performansı artırılmıştır.

Sentez sonucunda BCH kod çözücü algoritması kullanılan Virtex-6 (XC6VLX240T – 1FFG1156) FPGA de %13 kadar yer kaplamaktadır ve minimum periodu 4.345 ns olmak üzere maksimum frekansı 230 MHz'dir. LDPC kod çözücüsünün kapladığı alan ise %45 ve minimum periodu 4.49 ns olmak üzere maksimum frekansı 222 MHz seviyedindedir. Elde edilen sonuçlar düşünüldüğünde gerekleri (100MHz frekansında) hem kapladığı alan hem de hız açısından gerekleri fazlasıyla yerine getiren bir tasarım yapılmış olduğu görülmektedir. Herbir modül işlevsel bir şekilde tasarlanmıştır ve başka amaçlarda da kullanılabilecek çok esnek bir yapı oluşturulmuştur.

Tasarım her yönüyle incelenmiş, benzetimi ve sentezi yapılmıştır. Hem LDPC hem de BCH blokları FPGA gerçeklemesi yapılarak DVB-S2 ileri hata düzeltme birimi tasarımı tamamlanmıştır.

DESIGN AND IMPLEMENTATION OF FORWARD ERROR CORRECTION UNIT FOR SECOND GENERATION DIGITAL VIDEO BROADCASTING (DVB-S2)

SUMMARY

The need for reliability and efficiency of digital communication systems in recent years has been growing rapidly. This is because the need for long-distance communication and data processors increase in speed is needed to increase the communication speed. Traditional techniques of modulation and audio communication is generally low data rate and high probability of error.

To ensure reliable communications in noisy environments, low power consuming error correction codes are used. Adding that extra bit of data transmitted error-correcting codes to detect and fix the error resulting from the buyer provides a degree of correction of the well. These codes can increase the reliability of the communication receiver to increase the rate of correct detection. In this context, there are various error correction codes. On each of the areas of application of these codes also provide useful and high-performance.

The ideal error correcting codes for communication systems are still being investigated by researchers. As it provides the gain of each area covered by error correcting code, complexity, such as energy efficiency brings constraint.

This thesis, the digital video broadcasting via satellite communication systems at the receiving end, and in the implementation of the driven unit design includes forward error correction. In this context, the The European Telecommunications Standards Institute (ETSI) by the Second Generation Digital Video Broadcasting (DVB-S2) were taken into account to the specified standards. DVB-S2 standard for many different applications designed to be a flexible communication protocol. DVB-S2, MPEG-2 and MPEG-4 (Moving Pictures Expert Group) video standards and compatible high-definition broadcast service (HDTV) support. DVB-S2 supports single or multi-pack or a continuous flow of data.

It includes powerful forward error correction (FEC) based on BCH and LDPC coding. Limits based on the mode of transmission is almost Shannon (Quasi-Error Free) signal to noise ratio provides an improvement of about 0.7 dB up to 1 dB(AWGN channel). Many types include point sequence (QPSK, 8PSK, 16APSK, 32APSK). Depending on the nature channel adaptive coding and modulation with channel coding provides error protection (ACM). Receiver DVB-S2 forward error decoder side, and previous studies have examined taking into consideration the standards of the unit is designed and implemented in donanim, ModelSim VHDL code written with the help of MATLAB with the help of simulations made and confirmed.

High-speed and large-length data, such as DVB-S2 provides a high performance and fast, flexible structure, with emphasis on the design was quite convenient and uncomplicated.

In the literature, many different approaches are examined and the decision-making them the most appropriate models parallelism is required to be made as quickly as possible. Digital video broadcasting standard DVB-S2 satellite systems used for forward error correction unit, in order to provide faster performance and higher error correction coding technique involves sequential. This type of coding, coding types of classical repeatedly in a series connection and the other is the data input to the adoption of an encoding type is created.

As part of the sequential coding BCH(Bose-Chaudri-Hocquenghem) code is used for outer code and LDPC(Low Density Parity Check) code is used for inner code for DVB-S2 system.

BCH decoder design fast and flexible at the same time a structure has been established in terms of the area covered. Compatible with different data length can be set from outside the design parameters were made. BCH coding is done to protect the data packet errors that may occur. It is quite systematic and after the encoding of the data BCH coding bits of information is encoded and added. BCH error correction coding is applied for $t = 12$ bit error correction for all short-frame (short-frame) ($N = 16,200$), 7 of 11 different code types with normal frame (standard frame) ($N = 64,800$). The remaining four types of code in the normal frame $t = 10$ or $t = 8$ -bit BCH error correction coding is applied. $t = 12$ -bit error correction code has 192 parity bits(12×16), $t = 10$ -bit error correction code has 160 parity bits(10×16), and $t = 8$ bit parity error correction code produces 128 parity bits(8×16). Error indication (syndrome) have been quick calculation using parallel GF multipliers. Key equality and critical path delay to expect solvent to be at least RiBM (Reformulated inverse-free Berlekamp-Massey) algorithm is used. FIFO sizes were flexible enough to be set from the outside.

LDPC decoder structure of the belief propagation method that is ideal for high-speed communication systems, as opposed to the complexity of the selected literature and min-sum algorithm is used in order to reduce the footprint. LDPC can be fully expressed by parity-check matrix or by matrix code generator. DVB-S2 standard describes a method for encoding data therefore matrix is effectively described. Encoding and decoding operations are performed in a special way so no need to store the parity-check matrix or the manufacturer matrix. This standard method is defined as "standardized matrix". This standardized matrices "H" is the so-called parameter specifies the encoding step of classified and matrices for all DVB-S2, this value is equal to 360. Again regularly in order to be efficient in terms of speed table on the edge of the H matrix formed from a trusted source.

In this algorithm, the parity bits and the probability (likelihood) values, like a message comes back and forth between VN and CN's. The first possibility is the quality of the values and symbols from the channel with the constellation points between the Euclidean distances are calculated. Post the log-likelihood ratios provide a simple arithmetic is passed to the form. Other connected to the CN's been set to zero according to the indications on the basis of parity equations and VN hostels expected parity bits are connected to the CN determines.

The parity bits are expected to go along with all the VN's as LLR values. VN keeps parity bits of these values and use them to update the LLR values, and the cycle begins again. In this way, the equations of parity bits that are compatible with the right message corrected the bits with other corrupted bits.

VN depends on each one of these two LLR value of RAM, and each one holds the sum of the total coming from CN and VN contains messages from the collectors to create, combine and output messages and input and output messages that keeps the registers.

Each CN depends on the RAM and one big iteration, and each of the smaller of the two LLR value of the input message with the location of input messages with the sign of the pair holds bits. VN messages will go to any of the LLR value is generated by reading .

Shift values to be used in the mixing module, the control module and the CN's read and write addresses for reads from ROM . To move a message to VN , these values are used directly from the CN . In order to deliver the message to the CN VN in either direction in order to ensure the flow of data values in the scroll function is disabled. At the same time the control module delays may occur during all these operations are waste.

Number of iterations, mode code, making all kinds of data to be entered from the outside lengths (location or speed) created a design constraint that can be set for the optimum. In terms of both speed and in the s-bit and 360-bit RAM divided into groups partial checkpoints created a parallel structure. LLR is calculated according to the values from just 2 percent gain in the literature have been kept. Using soft decision LDPC code encoding to save space provided increased performance.

In this study, different from the literature data for different lengths of 21 pieces of the DVB-S2 standard, the analysis and hardware implementation of the BCH error correction codes and LDPC codes are made. High-speed and large-length data, such as DVB-S2 provides a high performance and fast, flexible structure, with emphasis on the design which was quite convenient and uncomplicated. In the literature, many different approaches are examined and the most appropriate models are chosen and parallelised is to make the system quickly as possible.

Examined all aspects of the design, simulation and synthesis is done. BCH and LDPC blocks as well as DVB-S2 forward error correction unit has been completed by FPGA design implementation.

1. GİRİŞ

Son yıllarda sayısal haberleşme sistemlerinde güvenilirlik ve verimliliğe olan gereksinim hızlı bir şekilde artmaktadır[1]. Bunun nedeni, uzun mesafeli haberleşme gereksinimi ve veri işlemcilerindeki hız artışıyla birlikte iletişim hızında artışa gereksinim duyulmasıdır. Geleneksel modülasyon ve ses haberleşme teknikleri genellikle düşük veri hızı ve yüksek hata olasılığı vermektedir.

Gürültülü ortamlarda düşük güç harcayarak güvenilir haberleşmeyi sağlayabilmek için hata düzeltme kodları kullanılmaktadır. Hata düzeltme kodları iletilen veriye ekstra bit ekleyerek alıcının oluşabilecek hatayı algılamasını ve düzeltilebilir derecede ise de düzeltilmesini sağlar. Bu kodlar alıcıdaki doğru algılama oranını artırarak haberleşmenin güvenilirliğini artırmaktadır. Bu bağlamda çeşitli hata düzeltme kodları vardır. Bu kodların herbiri de ilgili uygulama alanlarında kullanışlıdır ve yüksek başarımlar sağlar[2,3].

Haberleşme sistemleri için en ideal hata düzeltici kodlar hala araştırmacılar tarafından araştırılmaktadır[4]. Herbir hata düzeltici kod kazanım sağladığı gibi kapladığı alan, karmaşıklık, enerji verimliliği gibi birçok kısıtı da beraberinde getirmektedir[5]. Bu çalışmada DVB-S2 (ikinci nesil sayısal video yayını) için BCH (Bose Chaudhuri and Hocquenghem) ve LDPC (Low Density Parity Check) hata düzeltme kodları incelenmiştir ve hata düzeltme kod çözümleri tasarımı yapılarak FPGA ortamında gerçekleştirilmiştir. Tasarımın günümüz yüksek hızlı vericilerine karşılık gelecek yüksek veri hızını sağlayacak şekilde hızlı, daha sonra oluşabilecek gereksinimlere uyumlu olabilmesi açısından da esnek bir yapıda olmasına dikkat edilmiştir[6].

İkinci bölümde ilk olarak DVB-S2 hakkında bilgiler verilmektedir. DVB-S2 ile DVB-S arasındaki farklardan yola çıkılarak gerekliliği ön plana çıkarılmak istenmiştir. DVB-S2 sistem parametreleri olan kaynak kodlayıcı yapısı, modülasyon şekilleri, hata düzeltme kod yapıları hakkında bilgiler verilmektedir[7,8].

Üçüncü bölümde BCH kodu, kodlayıcı ve çözücü yapısı hakkında bilgiler verilmektedir. BCH kodunda kullanılan en temel yapı birimi olan GF cismi ayrıntılı olarak anlatılmaktadır[9,10]. Gerçekleme kısmına yönelik, tasarımda kullanılan yöntemlere kısaca değinilmiştir. Konu birçok örnek ve tanımla desteklenmektedir[11-14].

Dördüncü bölümde LDPC kodu, kodlayıcı ve çözücü yapısı hakkında matematiksel ifadelerle teknik bilgiler verilmektedir[15,16]. LDPC kodu kod çözme algoritmalarından bahsedilerek genel bilgiler verilmiştir[17,18]. Bunlara ek olarak örnek bir uygulama yapılmış ve konunun daha iyi anlaşılması sağlanmak istenmiştir[19-22].

Beşinci bölümde BCH kodu FPGA tasarımı benzetim sonuçları verilmektedir[8-13]. Kullanılan algoritmalar hakkında kısa bilgilendirmeler yapılmaktadır[12].

Altıncı bölümde LDPC kodu FPGA tasarımı benzetim sonuçları verilmektedir[23,24]. Kullanılan algoritmalar hakkında kısa bilgilendirmeler yapılmaktadır[25]. Bu tez çalışmasında Xilinx firması tarafından üretilen Virtex-6 (XC6VLX240T – 1FFG1156) FPGA kullanılmaktadır. İleriki bir çalışma olarak Xilinx ISE 12.4 ortamında tasarımı yapılan kod ML605 programabilir geliştirme kartı üzerinde bulunan Virtex-6 FPGA JTAG yardımıyla programlanarak donanımsal olarak test edilecektir.

Sonuç bölümünde çalışmanın gelinen safhası hakkında bilgiler verilerek bundan sonra yapılabilecek çalışmalar tartışılmıştır.

2. İKİNCİ NESİL SAYISAL VİDEO YAYINI (DVB-S2)

İkinci nesil Sayısal Video Yayını-Uydu (DVB-S2) standardı ETSI EN 302-307 olarak Avrupa Telekomünikasyon Standardı Enstitüsü (ETSI) tarafından sayısal video yayın standardı olarak 2005 Mart ayında yayınlanmıştır[3]. Yayınlanan bu standart modülasyon ve kanal kodlama sistemi hakkında bilgiler vermektedir. DVB-S2 de sayısal video yayınının yanında, internet gibi interaktif veri servislerine de yer verilmektedir[7].

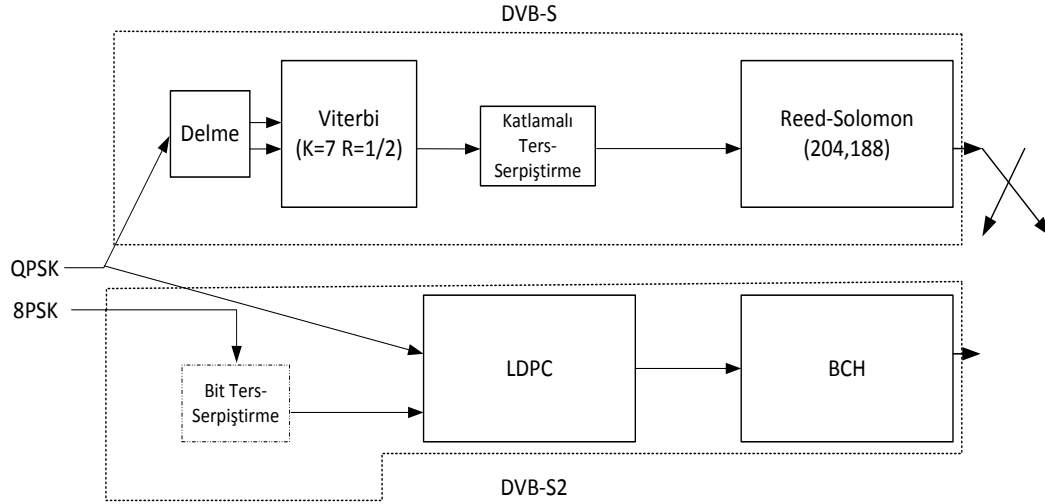
DVB-S2 birçok farklı uygulama için haberleşme protokolü sağlayan esnek bir standart olarak tasarlanmıştır[12]. DVB-S2, MPEG-2 ve MPEG-4 (Hareket Eden Resimler Uzman Grubu) video standartlarıyla uyumludur ve yüksek çözünürlüklü yayın servisini (HDTV) desteklemektedir.

DVB-S2 de farkedilen en belirgin özellikleri şu şekilde sıralayabiliriz:

- Tek yada çoklu paket yada sürekli veri akışını desteklemektedir.
- BCH ve LDPC kodlamaya dayanan güçlü bir ileri hata düzeltme (FEC) yeteneğine sahiptir. İletim moduna göre neredeyse Shannon limitlerine dayanan (Quasi-Error Free) sinyal gürültü oranında yaklaşık 0.7 dB'den 1 dB'lere varan bir iyileştirme sağlamaktadır[1].(AWGN kanalda)
- Birçok çeşit işaret dizilimini içermektedir(QPSK, 8PSK , 16APSK , 32APSK).
- Kanal yapısına bağlı olarak hata korumayı sağlayan kanal kodlaması olan uyarlabilir kodlama ve modülasyon (ACM) yeteneğine sahiptir.

Uydu haberleşmesinde verici tarafında oluşabilecek gürültü ve sinyal karışmalarını engellemek için DVB-S2 standardı güçlü bir hata düzeltme sunar[11]. Verici tarafında hata düzeltme sistemi üç basamaktan oluşur: BCH kodlama, LDPC kodlama ve bit serpiştirme[10]. Buna bağlı olarak da alıcı tarafında da sırasıyla bit ters serpiştirme, LDPC kod çözme ve BCH kod çözme işlemleri gerçekleştirilmektedir. BCH kod çözme işleminde giren veri bir ve sıfırlardan oluşan veri dizisidir bu nedenle çok karmaşık işlemlerle uğraşılmaz. Diğer taraftan LDPC kod çözme

işleminde giren veri bir ya da sıfır değerine sahip olduğunu belirten olasılıklardan bahsedilir, bu nedenle LDPC kodlama iç kod olarak algılanabilir. Bu safhada dış kod olarak BCH kodlamadan iç kod olarak da LDPC kodlamadan bahsedeceğiz. Aşağıdaki Şekil 2.1'de FEC yapısı verilmektedir.



Şekil 2.1 : FEC yapısı.

2.1 Dış Kod: DVB-S2 de BCH Kodu

Veri paketini oluşabilecek hatalardan korumak için BCH kodlama yapılır. Bu kodlama oldukça sistematiktir ve BCH kodlama sonrasında oluşan bilgi bitleri kodlanmış şekilde veriye eklenir. BCH kodlama $t=12$ bit hata düzeltme bütün kısa çerçeve (short-frame)($N=16200$) kod çeşidi ve 11 tane normal çerçeve(normal frame)($N=64800$)den 7 tanesi için uygulanmaktadır. Kalan dört normal çerçeve kod çeşidi ise $t=10$ veya $t=8$ bit hata düzeltme BCH kodlama uygulanmaktadır. $t=12$ bit hata düzeltme kodu $12 \times 16=192$ parite biti, $t=10$ bit hata düzeltme kodu $10 \times 16=160$ parite biti, $t=8$ bit hata düzeltme kodu $8 \times 16=128$ parite biti üretmektedir. Veri bit uzunluğu ile karşılaştırsak ekstra olan bu bitler kısa çerçeveler için %1 normal çerçeveler için ise %0.25 ine karşılık gelmektedir. Bu kodlar için üretici polinomlar DVB-S2 standardında verilmektedir[3].

2.2 İç Kod: DVB-S2 de LDPC Kodu

Daha önceden de bahsedildiği gibi DVB-S2 standardı güçlü bir hata koruma sağlamaktadır. Bunu da birçok kod çeşidi ve birçok farklı boyuttaki çerçeve yapısı için desteklemektedir. İki çeşit çerçeve uzunluğu desteklenmektedir, bunlar normal

çerçeve (N=64800) ve kısa çerçeve (N=16200) olarak adlandırılır. Normal çerçeve 11 tane kod oranını kısa çerçeve de 10 tane kod oranını destekler[9]. Bu yapı Çizelge 2.1' de verilmektedir.

Çizelge 2.1 : DVB-S2 data uzunlukları.

	Normal Çerçeve			Kısa Çerçeve		
Nominal Oran	Gerçek Oran	N	K	Gerçek Oran	N	K
1/4	1/4	64800	16200	1/5	16200	3240
1/3	1/3	64800	21600	1/3	16200	5400
2/5	2/5	64800	25920	2/5	16200	6480
1/2	1/2	64800	32400	4/9	16200	7200
3/5	3/5	64800	38880	3/5	16200	9720
2/3	2/3	64800	43200	2/3	16200	10800
3/4	3/4	64800	48600	11/15	16200	11880
4/5	4/5	64800	51840	7/9	16200	12600
5/6	5/6	64800	54000	37/45	16200	13320
8/9	8/9	64800	57600	8/9	16200	14400
9/10	9/10	64800	58320			

LDPC kod üretici matrisle ya da parite-çek matrisiyle tamamen ifade edilebilir. DVB-S2 standardı veri kodlamada kullanılacak bir metod tanımlar bu nedenle de üretici matrisi efektif bir şekilde tanımlanabilir[4]. Kodlama ve kod çözme işlemleri özel bir şekilde yapılmaktadır bu nedenle ayrıca üretici matrisleri ya da parite-kontrol matrislerini depolamaya gerek kalmaz[26,27]. Bu metod standartta

“standartlaştırılmış”(standardized) matris olarak tanımlanmıştır. Bu standartlaştırılmış matrisler "M" diye tabir edilen kodlamada sınıflandırılmış bir basamağı belirten parametreye sahiptir ve bütün DVB-S2 matrislerinde bu değer 360 a eşittir[14].

3. BOSE CHAUDHURI AND HOCQUENGHEM (BCH) KODLARI

Bu bölümde $GF(2^m)$ Galois cisimlerinden faydalanarak, devresel kodların BCH kodları denen, özel bir sınıfı incelenecektir. Bu kodun üreteç polinomu, $GF(2^m)$ Galois cismindeki kökleri cinsinden belirlenir. $GF(2^m)$ 'deki bir primitif eleman α olsun. $2^m - 1$ uzunluklu, t -hata düzelten BCH kodunun $g(x)$ üreteç polinomu, $\alpha, \alpha^2, \dots, \alpha^{2t}$ leri kök kabul eden, $GF(2)$ üzerindeki en küçük dereceli polinomdur[10]. (yani $(\alpha^i)^{2^l} = 0, 1 \leq i \leq 2t$) Katsayıları $GF(2)$ 'de bulunan bir $p(x)$ polinomu ele alınsa; α , $GF(2)$ 'nin bir cisim genişlemesinde bulunsun, herhangi $l \geq 0$ için α^{2^l} nin, $p(x)$ polinomunun bir kökü olduğu bilinmektedir. Bu nedenle, $g(x)$ polinomunun köklerinin hepsi $\alpha, \alpha^2, \dots, \alpha^{2t}$ ve bunların eşlenikleridir[11,28].

Öncelikle, sıkça kullanılacak olan bazı kavramlar açıklanmalıdır.

Tanım 3.1. (Primitif Eleman) $GF(q)$ cisminde, bir $a \neq 0$ elemanının mertebesi $q - 1$ ise yani, $a^n = 1$ şeklindeki en küçük pozitif n tam sayısı $q - 1$ ise a ya, bir “primitif eleman” denir.

O zaman, $GF(q)$ nun sıfırdan farklı elemanları, a nın kuvvetleriyle elde edilir.

Örnek 3.1. $GF(7)$ de,

$$3^1 = 3$$

$$3^2 = 2$$

$$3^3 = 6$$

$$3^4 = 4$$

$$3^5 = 5$$

$$3^6 = 1$$

dir.

Yani $3^{q-1} = 3^{7-1} = 1$ olup, 3 bir primitif elemandır.

$GF(2)$ üzerinde bir $p(x)$ polinomu ele alınsın. $p(x)$ in derecesi m olsun. Eğer $p(x)$ polinomu; $GF(2)$ üzerinde, derecesi m den küçük, sıfırdan büyük bir polinomla bölünemiyorsa; $p(x)$ e, $GF(2)$ de indirgenemez bir polinom denir[14].

Tanım 3.2. (Primitif Polinom) İndirgenemez bir $p(x)$ polinomu ele alınsın. $p(x)$ in derecesi m olsun. Eğer $p(x) \mid x^n + 1$ şeklindeki en küçük pozitif n tam sayısı $n = 2m - 1$ ise $p(x)$ polinomuna, bir “primitif polinom” denir[10][29].

Daha genel bir anlatımla; $p(x) \in F[x]$ $q \in R$, $\deg p(x) \geq 1$ olan bir $p(x)$ polinomu, F_{q^m} nin bir primitif elemanının F_q üzerindeki minimal polinomu ise $p(x)$ polinomuna, F_q üzerinde bir primitif polinomdur denir.

Örnek 3.2. $p(x) = x^4 + x + 1$ olsun. $p(x) \mid x^{15} + 1$ dir. Fakat $1 \leq n < 15$ için, $p(x) \nmid x^n + 1$ dir.

Bu durumda;

$n = 15 = 2^4 - 1$, $m = 4$ olup, $p(x) = x^4 + x + 1$ polinomu, primitiftir.

Örnek 3.3. $x^4 + x^3 + x^2 + x + 1$ polinomu, $GF(2)$ de indirgenemezdir. Fakat primitif değildir. Çünkü;

$m = 4$, $2^4 - 1 = 15$ ve $5 < 15$ için,

$$x^4 + x^3 + x^2 + x + 1 \mid x^5 + 1$$

dir.

Örnek 3.4. $x^3 + x + 1$ polinomu, $GF(2)$ de indirgenemezdir. Bu nedenle

$$F_2/(x^3 + x + 1) = \{0, 1, x, x^2, x^3 = x + 1, x^4 = x^2 + x, x^5 = x^2 + x + 1, x^6 = x^2 + 1\}$$

primitif değildir. Çünkü;

$m = 4$, $2^4 - 1 = 15$ ve $5 < 15$ için,

$$x^4 + x^3 + x^2 + x + 1 \mid x^5 + 1$$

dir.

Not. Verilen bir m sayısı için, m dereceli primitif polinomların sayısı, birden fazla olabilir.

Tanım 3.3. Derecesi m den daha küçük olan $F[x]$ deki tüm polinomların kümesi, $F^m[x]$ olarak tanımlanır. F^m deki her bir sözcük, $F^{(m)}[x]$ deki bir polinoma karşılık gelir.

Örnek 3.5 $h(x) = 1 + x + x^4$ polinomu ele alınsın. $GF(2^2)$ de $(1101)(0101)$ çarpımının sonucunu bulalım.

$$(1101)(0101) \leftrightarrow (1 + x + x^3)(x + x^3)$$

$$= x + x^2 + x^3 + x^6$$

$$x \equiv x + x^2 + x^3 + x^6 \pmod{(1 + x + x^4)} \text{ bulunur.}$$

Bu nedenle $(1101)(0101) = 0100 \leftrightarrow x^4$ tir.

Örnek 3.6. Çarpmanın tanımı ile $h(x) = 1 + x + x^3$ primitif polinomunu kullanarak, $GF(2^3)$ cismini kuralım.

$$\text{Sözcük} \leftrightarrow x_i \pmod{h(x)}$$

$$100 \quad 1$$

$$010 \quad x$$

$$001 \quad x^2$$

$$110 \quad x^3 \equiv 1 + x$$

$$011 \quad x^4 \equiv x + x^2$$

$$111 \quad x^5 \equiv 1 + x + x^2$$

$$101 \quad x^6 \equiv 1 + x^2$$

Tabloya göre

$$(110)(001) \leftrightarrow (1 + x) x^2$$

$$\equiv x^3 x^2$$

$$\equiv x^5$$

$$\equiv 1 + x + x^2 \pmod{(h(x))} \text{ olarak bulunur.}$$

Dolayısıyla sonuç 111 e eşit olur.

Örnek 3.7. $p(x) = 1 + x + x^4$ polinomunu kullanarak, $GF(2^4)$ cismini kuralım.

$p(x) = 1 + x + x^4$ polinomunun, $GF(2)$ üzerinde primitif bir polinom olduğu bilinmektedir.

$$p(\alpha) = 1 + \alpha + \alpha^4 = 0$$

dır. Buradan, $\alpha = 1 + \alpha^4$ eşitliği kullanılarak, $GF(2^4)$ cismi kurulabilir.

Örneğin;

$$\alpha^5 = \alpha\alpha^4 = (1 + \alpha) = \alpha + \alpha^2$$

$$\alpha^6 = \alpha^2\alpha^4 = \alpha^2(1 + \alpha) = \alpha^2 + \alpha^3$$

dır. İşlemlere bu şekilde devam edildiğinde, $p(x) = 1 + x + x^4$ ile üretilen $GF(2^4)$ ün elemanları, aşağıdaki Çizelge 3.1 'de verildiği gibi bulunur.

Çizelge 3.1 : $GF(2^4)$ elemanları.

Binary	Polinom	α katsayısı
0000	0	-
1000	1	1
0100	α	α
0010	α^2	α^2
0001	α^3	α^3
1100	$1 + \alpha$	α^4
0110	$\alpha + \alpha^2$	α^5
0011	$\alpha^2 + \alpha^3$	α^6
1101	$1 + \alpha + \alpha^3$	α^7
1010	$1 + \alpha^2$	α^8
0101	$\alpha + \alpha^3$	α^9
1110	$1 + \alpha + \alpha^2$	α^{10}
0111	$\alpha + \alpha^2 + \alpha^3$	α^{11}
1111	$1 + \alpha + \alpha^2 + \alpha^3$	α^{12}
1011	$1 + \alpha^2 + \alpha^3$	α^{13}
1001	$1 + \alpha^3$	α^{14}

BCH kodlarının kurulumunda minimal polinomlar, büyük önem taşımaktadır.

Bu nedenle, minimal polinomları ayrıntılı olarak incelemek, yararlı olacaktır.

3.1 Minimal Polinomlar

$GF(2^m)$ nin sıfırdan farklı 2^{m-1} elemanı, $2^{m-1}+1$ x in tüm köklerini oluşturur. $GF(2^m)$ nin 0 elemanı, x polinomunun kökü olduğundan, $GF(2^m)$ nin elemanları, $x^{2^{m-1}}+1$ polinomunun tüm köklerini oluşturur[30]. Buna göre $GF(2^m)$ nin herhangi bir β elemanı, $x^{2^{m-1}}+1$ polinomunun bir kökü olduğundan; β , $GF(2)$ üzerinde,

derecesi 2^m den küçük olan bir polinomun kökü olabilir. $\varphi(x)$, $GF(2)$ üzerinde $\varphi(\beta) = 0$ şeklindeki en küçük dereceli polinom olsun. (Bunun, bir tek olduğu bellidir.) Bu $\varphi(x)$ polinomuna, β nın “minimal polinomu” denir [10].

Örneğin; $GF(2^m)$ nin 0 nın minimal polinomu x , 1 inin ise $x+1$ dir.

Minimal polinomun özelliklerine geçmeden önce, eşlenik kavramını inceleyelim. Eğer β , $p(x)$ in bir kökü ise $(\beta, GF(2)$ nin genişlemesi), β^{2^l} nin de $p(x)$ in bir kökü olduğu bilinmektedir. Bu β^{2^l} elemanına, β nın bir eşleniği denmektedir. Böylece β nın tüm eşlenikleri, $GF(2^m)$ nin de elemanıdır ve aynı zamanda $p(x)$ in kökleridir.

Örnek 3.8. $p(x) = 1 + x^3 + x^4 + x^5 + x^6$ polinomunun bir kökü α^4 tür ve α^4 , $GF(2^4)$ ün elemanıdır. Bu, aşağıdaki yolla kolaylıkla gösterilebilir:

$$\begin{aligned} p(\alpha^4) &= 1 + \alpha^{12} + \alpha^{16} + \alpha^{20} + \alpha^{24} \\ &= 1 + \alpha^{12} + \alpha^{14} \alpha^2 + \alpha^{14} \alpha^6 + \alpha^{14} \alpha^{10} \\ &= 1 + \alpha^{12} + (1 + \alpha^3) \alpha^2 + (1 + \alpha^3) \alpha^6 + (1 + \alpha^3) \alpha^{10} \\ &= 1 + \alpha^{12} + \alpha^2 + \alpha^5 + \alpha^6 + \alpha^9 + \alpha^{10} + \alpha^{13} \\ &= 1 + 1 + \alpha + \alpha^2 + \alpha^3 + \alpha^2 + \alpha + \alpha^2 + \alpha^2 + \alpha^3 + 1 + \alpha + \alpha^2 + 1 + \alpha^2 + \alpha^3 \\ &= 0 \end{aligned}$$

α^4 ün eşlenikleri aşağıdaki gibidir;

$$\begin{aligned} (\alpha^4)^2 &= \alpha^8, \\ (\alpha^4)^{2^2} &= \alpha^{16}, \\ (\alpha^4)^{2^3} &= \alpha^{32} = \alpha^2. \end{aligned}$$

Örnek 3.9. Örnek 3.7 de $p(x) = 1 + x + x^4$ polinomu kullanılarak, $GF(2^4)$ cismini kurmuştuk.

$$p(\alpha) = 1 + \alpha + \alpha^4 = 0$$

olduğundan α , α^3 ve α^5 in minimal polinomları sırasıyla,

$$\phi_1(x) = 1 + x + x^4$$

$$\phi_3(x) = 1 + x + x^2 + x^3 + x^4$$

$$\phi_5(x) = 1 + x + x^2$$

dir. Bunların nasıl elde edildiğini kısaca açıklayalım.

Biliyoruz ki, $GF(2^m)$ de bir β elemanının minimal polinomu $\phi(x)$ olduğunda,

$\beta^{2^e} = \beta$ şeklindeki en küçük tam sayı e ise

$$\phi(x) = \prod_{i=0}^{e-1} (x + \beta^{2^i})$$

dir.

$\beta = \alpha^3$ alalım. β nın eşlenikleri,

$$\beta^2 = \alpha^6$$

$$\beta^{2^2} = \alpha^{12}$$

$$\beta^{2^3} = \alpha^{24} = \alpha^9$$

dır. Minimal polinomu,

$$\phi(x) = (x + \alpha^3)(x + \alpha^6)(x + \alpha^{12})(x + \alpha^9)$$

olur.

Gerekli çarpımlar yapılnca,

$$\phi(x) = x^4 + x^3 + x^2 + x + 1$$

polinomu elde edilir.

Teorem 3.1. $GF(2^m)$ nin bir elemanı, $\alpha \neq 0$ olsun. $\phi_\alpha(x)$, α nın minimal polinomu olsun. Bu durumda,

(i) $\phi_\alpha(x)$, F üzerinde indirgenemezdir.

(ii) $p(x)$, F üzerinde herhangi bir polinom ise $p(\alpha) = 0$ dır. O zaman $\phi_\alpha(x)$, $p(x)$ polinomunun bir çarpanıdır.

(iii) Minimal polinom tektir.

(iv) $\phi_\alpha(x)$ minimal polinomu, $1 + x^{2^m-1}$ in bir çarpanıdır.

Teorem 3.2. $\phi_\alpha(x)$ minimal polinomu ile $GF(2^m)$ de bir eleman α olsun. Bu

durumda $\{\alpha, \alpha^2, \alpha^4, \dots, \alpha^{2^m-1}\}$ α kümesi, $\phi_\alpha(x)$ polinomunun tüm köklerinin

kümesidir.

$\phi_\alpha(x)$ polinomunun derecesi,

$$|\{\alpha, \alpha^2, \alpha^4, \dots, \alpha^{2^m-1}\}|$$

dir.

Örnek 3.10. $p(x) = 1 + x + x^4$ polinomu ile üretilen $GF(2^m)$ cismindeki tüm elemanların minimal polinomları aşağıdaki Çizelge 3.2 de verilmiştir.

$p(x) = 1 + x + x^4$ polinomu ile üretilen $GF(2^4)$ cismindeki tüm elemanların minimal polinomları Çizelge 3.2 'de verilmiştir.

Çizelge 3.2 : $GF(2^4)$ elemanları minimal polinomları.

Eşlenik Kökler	Minimal Polinomlar
0	x
1	$1 + x$
$\alpha, \alpha^2, \alpha^4, \alpha^8$	$1 + x + x^4$
$\alpha^3, \alpha^6, \alpha^9, \alpha^{12}$	$1 + x + x^2 + x^3 + x^4$
α^5, α^{10}	$1 + x + x^2$
$\alpha^7, \alpha^{11}, \alpha^{13}, \alpha^{14}$	$1 + x^3 + x^4$

Örnek 3.11. $\alpha = \beta^5, \beta^5 \in GF(2)$ in minimal polinomu, $\phi_5(x)$ olsun. Zira,

$$\{\alpha, \alpha^2, \alpha^4, \alpha^8\} = \{\beta^5, \beta^{10}\}$$

dır.

$\phi_5(x)$ polinomunun kökleri β^5 ve β^{10} elemanlarıdır.

Dolayısıyla,

$$\deg(\phi_5(x)) = 2$$

dir.

Bu nedenle,

$$\phi_5(x) = a_0 + a_1x + a_2x^2$$

$$0 = a_0 + a_1\beta^5 + a_2\beta^{10}$$

$$= a_0(1000) + a_1(0110) + a_2(1110)$$

dır.

$$a_0 = a_1 = a_2 = 1$$

ve

$$\phi_5(x) = 1 + x + x^2$$

dir.

3.1.1 GF(2^m) 'de Aritmetik İşlemler

Bu bölümde BCH kodlamasında da faydalanacağımız GF(2^m)'de aritmetik işlemlerden bahsedeceğiz. Donanım implemantasyon tarafında da sonlu bölge işlemleri kullanılmaktadır[11].

3.1.1.1 GF(2^m)'de Toplama/Çıkarma İşlemi

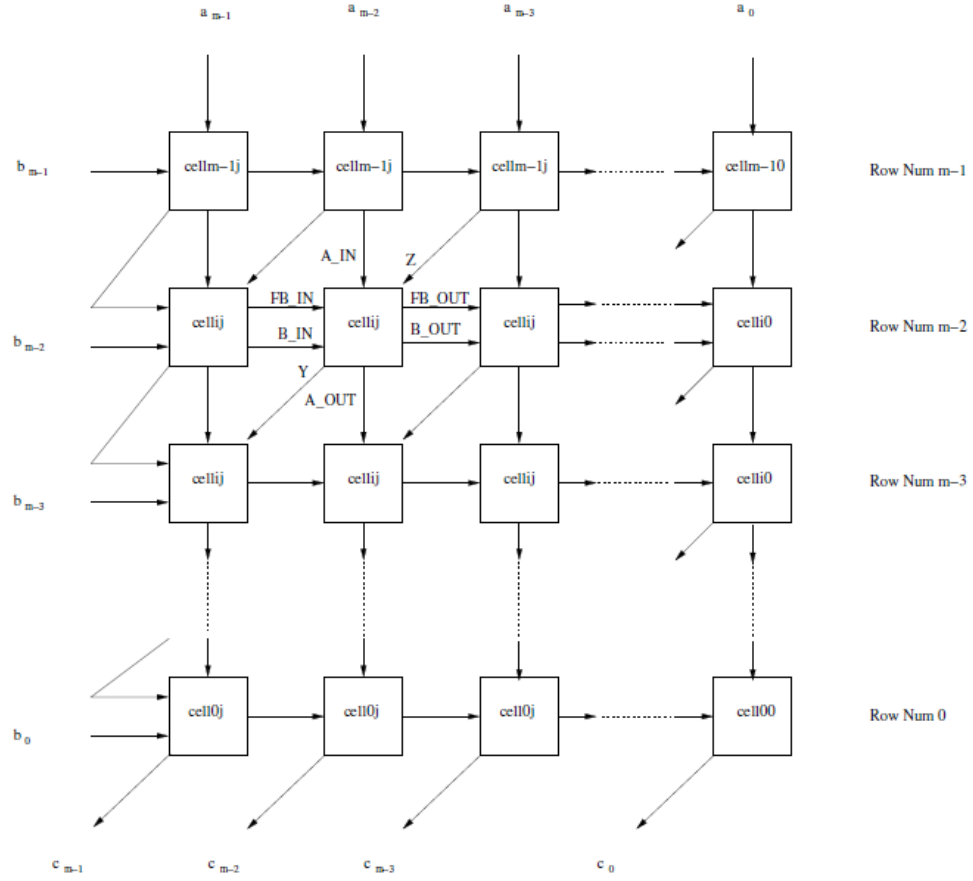
GF(2^m) 'de a ve b olmak üzere iki elemanın toplama ve çıkarma işlemleri polinom toplama ve çıkarma işlemleri olarak yapılmaktadır. Bu polinomların katsayıları GF(2) üzerinde tanımlanmıştır[10]. GF(2^m) 'de toplama işlemi a ve b vektörlerinde bit bit XOR işlemine karşılık gelmektedir[11].

3.1.1.2 GF(2^m)'de Çarpma İşlemi

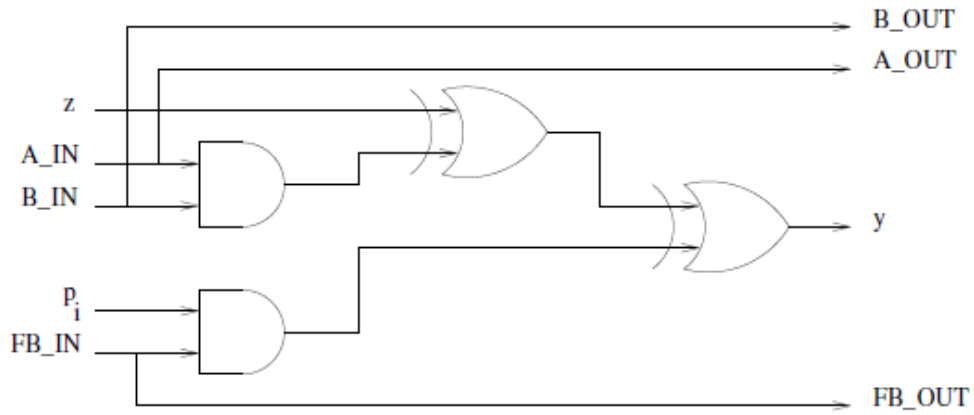
GF(2^m)'de a ve b olmak üzere iki elemanın çarpma işlemini yapabilmek için derecesi m olan indirgenemez bir polinom seçilmelidir[31]. Farklı seçilen polinomlar farklı işlemleri de beraberinde getirir ancak sonuçta aynı değerler elde edilir. Çarpma işlemini anlamak için P(x) GF(2)'de tanımlı m derecesinde indirgenemez bir polinom olsun, p_m ve p_0 değerleri de 1' e eşit diyelim. a ve b polinomlarının GF(2^m)'de çarpım sonucu $c = a.b$, $c(x) = a(x)b(x) \mod(p(x))$ işlemi yapılarak elde edilir. Çarpma işleminin donanım mimarisi üzerine Mastrovito'nun 1991 yılındaki tezi iyi bir kaynak olarak ele alınabilir.

3.1.1.3 GF(2^m)'de Çarpma İşlemi Donanım İmplemantasyonu

1971 yılında Laws and Rushforth hücresel-array çarpıcıyı yayınladılar [32]. Bu array oldukça anlaşılır ve zaman yerine uzayda çarpma işlemlerini yerine getiren bir çözüm yolu olarak düşünülebilir. Hücresel-array çarpıcının blok şeması Şekil 3.1 'de Şekil 3.2 'de de hücre yapısı görülmektedir.



Şekil 3.1 : $GF(2^m)$ 'de hücresel-array çarpıcı blok şeması.



Şekil 3.2 : $GF(2^m)$ 'de hücresel-array çarpıcının hücre yapısı.

3.2 BCH Kodları

1960 yılında R. C. Bose ve D. K. Ray-Chaudri ile bunlardan bağımsız olarak 1959 yılında A. Hocquenghem tarafından bulunmuştur[14]. Bu nedenle, Bose-Chaudri-

Hocquenghem'in isimlerinin baş harfleriyle, yani BCH ile isimlendirilmiştir. BCH kodları, devresel kodların önemli bir sınıfını oluşturmaktadır[12]. Uygulama alanı oldukça geniştir.

Öncelikle, anlaşılması güçlük gösteren ve BCH kodlarının minimum mesafesi için oldukça önemli olan, “tasarlanmış mesafe” kavramı açıklanacaktır[15].

3.2.1 Tasarlanmış Mesafe

BCH kodlarının kuruluşunda önemli bir problem, kodun boyutu k ve minimum d mesafesinin belirtilmesindeki güçluktur[33,34]. Tasarlanmış mesafe, gerçek d mesafesi hakkında verilen bir alt sınırdır. Bu alt sınırın, yani “tasarlanmış mesafenin” bilinmesi, oldukça yararlı olmaktadır.

Örneğin, bir devresel kod verilmişse ve onun üreteç polinomunun $d - 1$ elemanlı bir kök dizisine sahip olduğunu (ki bunlar, verilen bir kökün ardışık kuvvetlerinden oluşmuştur.) gösterebilirsek, o zaman kodun minimum mesafesi, en azından d dir. Bu sınır, sözkonusu kodun tipi göz önüne alınmaksızın, BCH sınırı olarak düşünülür.

Daha açık bir anlatımla; t -hata düzelten BCH kodunun minimum mesafesi, en az $2t + 1$ dir. Bu, daha sonra bir teorem olarak ispatlanacaktır. Buradaki $2t + 1$ mesafesi, çoğunlukla t -hata düzelten BCH kodunun tasarlanmış mesafesi olarak adlandırılır. Bir BCH kodunun gerçek minimum mesafesi, tasarlanmış mesafeye eşit olur veya olmayabilir. Gerçek minimum mesafenin, tasarlanmış mesafeye eşit olduğu birçok durum var olmasına karşın; gerçek minimum mesafenin, tasarlanmış mesafeden büyük olduğu durumlar da vardır.

Tanım 3.4. b , negatif olmayan bir tam sayı ve $\alpha \in F_{q^m}$, birimin n . primitive kökü olsun. Burada m , q nun (modül n) e göre çarpımsal mertebesidir. F_q üzerinde n uzunluklu ve d tasarlanmış mesafeli ($2 \leq d \leq n$) olan bir BCH kodu, üreteç polinomun $\alpha^b, \alpha^{b+1}, \dots, \alpha^{b+d-2}$ kökleriyle tanımlı bir devresel koddur. $b = 1$ ise BCH kodları, “Dar anlamda BCH kodları” olarak adlandırılan kodlara karşılık gelir.

$n = q^m - 1$ ise BCH kodları, “primitif” olarak adlandırılır. $n = q - 1$ ise F_q üzerinde n uzunluklu bir BCH kodu, “Reed-Solomon Kodu” olarak bilinir.

Teorem 3.3. Tasarlanmış mesafesi d olan bir BCH kodunun minimum uzaklığı

en az d dir. Buradan, kodun minimum uzaklığının en az d olduğu görülür.

3.2.2 BCH Kodunun Kuruluşu

Herhangi $m \geq 3$ ve $t < 2^{m-1}$ pozitif tam sayıları ele alınsın. Bu durumda,

Sözcük Uzunluğu : $n = 2^m - 1$

Eşlik-Denetim Sembollerinin Sayısı : $n - k \leq mt$

Minimum Mesafe : $d_{min} \geq 2t + 1$

parametrelerine sahip ikili (binary) BCH kodu mevcuttur. Bu kod, t veya daha az hatayı düzeltebilir. Bu nedenle, t -hata düzelten BCH kodu adını alır. Bu kodun üreteç polinomu, $GF(2^m)$ Galois cismindeki kökleri cinsinden belirlenir.

$GF(2^m)$ deki bir primitif eleman α olsun. $2^m - 1$ uzunluklu, t -hata düzelten BCH kodunun $g(x)$ üreteç polinomu, $\alpha, \alpha^2, \dots, \alpha^{2t}$ leri kök kabul eden, $GF(2)$ üzerindeki en küçük dereceli polinomdur. (yani $(\alpha^i)^{2^l} = 0, 1 \leq i \leq 2t$) Katsayıları $GF(2)$ de bulunan bir $p(x)$ polinomu ele alınsa; $\alpha, GF(2)$ nin bir cisim genişlemesinde bulunsun, herhangi $l \geq 0$ için α^{2^l} nin, $p(x)$ polinomunun bir kökü olduğu bilinmektedir. Bu nedenle, $g(x)$ polinomunun köklerinin hepsi $\alpha, \alpha^2, \dots, \alpha^{2t}$ ve bunların eşlenikleridir. α^i nin minimal polinomu $\phi_i(x)$ olsun. O zaman $g(x)$ polinomu; $\phi_1(x), \phi_2(x), \dots, \phi_{2t}(x)$ polinomlarının en küçük ortak katı olur. Yani,

$$g(x) = \text{ekok}\{\phi_1(x), \phi_2(x), \dots, \phi_{2t}(x)\} \quad (3.1)$$

dir.

Eğer i bir çift sayı ise,

$$i = i'2^l \quad (3.2)$$

şeklinde yazılabilir. Burada i' bir tek sayı ve $l \geq 1$ dir.

O zaman,

$$\alpha^i = (\alpha^{i'})^{2^l}$$

ifadesi, $\alpha^{i'}$ nün bir eşleniğidir. Böylece de α^i ve $\alpha^{i'}$ elemanları, aynı minimal polinoma sahiptir. Yani minimal polinomlar,

$$\phi_i(x) = \phi_{i'}(x) \text{ tir.}$$

Böylece,

$$\alpha, \alpha^2, \alpha^3, \dots, \alpha^{2t} \quad (3.3)$$

dizisinde α nın her bir çift kuvveti, dizide kendinden önce gelen tek kuvvetli elemanla aynı minimal polinoma sahiptir.

Böyle olunca, (3.2) ile verilen $2^m - 1$ uzunluklu t -hata düzelten ikili (binary) BCH kodunun $g(x)$ üreteç polinomu,

$$g(x) = \text{ekok}\{\phi_1(x), \phi_2(x), \dots, \phi_{2t}(x)\} \quad (3.4)$$

ifadesine indirgenir.

Herbir minimal polinomun derecesi m veya daha az olduğuna göre, $g(x)$ polinomunun derecesi en fazla mt ye eşit olabilir. Yani, $n - k$ olan eşlik-denetim sembollerinin sayısı en fazla mt olabilir. $n - k$ yı saymak için basit bir formül yoktur. Fakat, eğer t küçükse, kesinlikle $n - k = mt$ olur. Yukarıdaki gibi tanımlanan BCH kodlarına, primitif (veya dar-anlamda) BCH kodları denmektedir. (3.4) ten dolayı, $2^m - 1$ uzunluklu tek-hata düzelten BCH kodu,

$$g(x) = \phi_1(x)$$

ile üretilir.

$GF(2^m)$ nin primitif elemanı α olduğundan; $\phi_1(x)$ polinomu, derecesi m olan primitif bir polinomdur. O zaman $2^m - 1$ uzunluklu, tek-hata düzelten BCH kodu, bir Hamming kod adını alır.

Örnek 3.12. $q = 2$, $n = 15$ ve $d = 4$ olsun. Bu durumda $x^4 + x + 1$ polinomu, $GF(2)$ üzerinde indirgenemezdir ve $x^4 + x + 1$ in kökleri, $GF(2^4)$ ün primitive elemanlarıdır. Köklerden biri α ise bu durumda α^2 de bir köktür ve α^3 , $(x^4 + x^3 + x^2 + x + 1)$ polinomunun bir köküdür. Bu nedenle $d = 4$ olacak şekilde dar anlamda bir BCH kodu, $g(x) = (x^4 + x + 1)(x^4 + x^3 + x^2 + x + 1)$ ile üretilir. Üstelik $g(x)$ polinomu, $d = 5$ olacak şekilde bir BCH kodu için üreteçtir. α^2 , $(x^4 + x + 1)$ in bir köküydü. Bu kodun boyutu,

$$15 - \deg(g(x)) = 7$$

dir.

Örnek 3.13. $n = 31$, $m = 5$, $q = 2$, $d = 8$ olsun. α , $GF(2^5)$ in bir primitive elemanı olsun. α nın minimal polinomu, $(x - \alpha)(x - \alpha^2)(x - \alpha^4)(x - \alpha^8)(x - \alpha^{16})$ dır.

Aynı yolla $\phi_3(x)$ polinomunu da buluruz. Fakat,

$$\phi_5(x) = (x - a^5)(x - a^{10})(x - a^{20})(x - a^9)(x - a^{18}) = \phi_9(x)$$

tir.

$g(x)$; $\phi_1(x)$, $\phi_1(x)$, $\phi_3(x)$, $\phi_5(x)$, $\phi_7(x)$ ve $\phi_9(x)$ polinomlarının, en küçük ortak katıdır. Buradan, tasarlanmış mesafesi 8 olan bir primitif BCH kodunun minimum uzaklığının, en az 11 olduğu görülür.

Örnek 3.14. Örnek 3.7 de $p(x) = 1 + x + x^4$ polinomu kullanılarak, $GF(2^4)$

cismi kurulmuştu. α , bu cismin bir primitif elemanı idi. Bu polinom için, $n = 2^4 - 1 = 15$ uzunluklu iki-hata düzelten BCH kodu, $g(x) = \text{ekok}\{\phi_1(x), \phi_3(x)\}$ polinomu ile üretilir.

Örnek 3.1.2 de α , α^3 ve α^5 elemanlarının minimal polinomları sırasıyla,

$$\phi_1(x) = 1 + x + x^4$$

$$\phi_3(x) = 1 + x + x^2 + x^3 + x^4$$

$$\phi_5(x) = 1 + x + x^2$$

idi. $\phi_1(x)$ ve $\phi_3(x)$, iki farklı indirgenemeyen polinom olduğundan,

$$\begin{aligned} g(x) &= (\phi_1(x))(\phi_3(x)) \\ &= (1 + x + x^4)(1 + x + x^2 + x^3 + x^4) \\ &= 1 + x^4 + x^6 + x^7 + x^8 \end{aligned}$$

olarak bulunur.

Sözkonusu kod, $5 \leq d \leq 15$ olan (15, 7)-parametrelili devresel bir koddur. Üreteç polinomu, 5 ağırlıklı bir kod polinomu olduğundan, minimum mesafe kesinlikle 5 tir.

Üç-hata düzelten 15 uzunluklu BCH kodu ise,

$$\begin{aligned} g(x) &= \text{ekok}\{\phi_1(x), \phi_3(x), \phi_5(x)\} \\ &= 1 + x + x^2 + x^4 + x^5 + x^8 + x^{10} \end{aligned}$$

polinomu ile üretilir.

Bu kod, (15, 5)-parametrelili bir devresel koddur. $\min d \geq 7$ dir. Fakat üreteç polinom 7 ağırlıklı olduğundan, minimum mesafe kesinlikle 7 olarak bulunur.

Teorem 3.4. Herhangi bir $m \geq 4$ tam sayısı için $k = 2^m - 2m - 1$ boyutlu,

$n = 2^m - 1$ uzunluklu iki-hata düzelten bir BCH kodu ve $d = 5$ uzaklıklı

$g(x) = \phi_1(x)\phi_2(x)$ üreteç polinomu vardır.

3.2.3 n=63 İçin Tüm BCH Kodlarının Oluşturulması

$p(x) = 1 + x + x^6$ primitif polinomu kullanılarak $GF(2^6)$ Galois cismi kolaylıkla kurulabilir.

- $p(\alpha) = 1 + \alpha + \alpha^6$ alınarak $GF(2^6)$ nın elemanlarını α nın kuvveti olarak, polinom olarak, sıralı 6 lı olarak yazalım.

$p(x) = 1 + x + x^6$ primitif polinomu kullanılarak oluşturulan $GF(2^6)$ nın elemanları Çizelge 3.3 'te verilmiştir.

Çizelge 3.3 : $GF(2^6)$ elemanları.

α nın Kuvveti	α Cinsinden Polinom Olarak	Binary
0	0	000000
1	1	100000
α	α	010000
α^2	α^2	001000
α^3	α^3	000100
α^4	α^4	000010
α^5	α^5	000001
α^6	$1 + \alpha$	110000
α^7	$\alpha + \alpha^2$	011000
α^8	$\alpha^2 + \alpha^3$	001100
α^9	$\alpha^3 + \alpha^4$	000110
α^{10}	$\alpha^4 + \alpha^5$	000011
α^{11}	$1 + \alpha + \alpha^5$	110001
α^{12}	$1 + \alpha^2$	101000
α^{13}	$\alpha + \alpha^3$	010100
α^{14}	$\alpha^2 + \alpha^4$	001010
α^{15}	$\alpha^3 + \alpha^5$	000101
α^{16}	$1 + \alpha + \alpha^4$	110010
α^{17}	$\alpha + \alpha^2 + \alpha^5$	011001
α^{18}	$1 + \alpha + \alpha^2 + \alpha^3$	111100
α^{19}	$\alpha + \alpha^2 + \alpha^3 + \alpha^4$	011110
α^{20}	$\alpha^2 + \alpha^3 + \alpha^4 + \alpha^5$	001111
α^{21}	$1 + \alpha + \alpha^3 + \alpha^4 + \alpha^5$	110111
α^{22}	$1 + \alpha^2 + \alpha^4 + \alpha^5$	101011
α^{23}	$1 + \alpha^3 + \alpha^5$	100101
α^{24}	$1 + \alpha^4$	100010
α^{25}	$\alpha + \alpha^5$	010001
α^{26}	$1 + \alpha + \alpha^2$	111000

Çizelge 3.3 (devam) : GF(2⁶) elemanları.

α^{27}	$\alpha + \alpha^2 + \alpha^3$	011100
α^{28}	$\alpha^2 + \alpha^3 + \alpha^4$	001110
α^{29}	$\alpha^3 + \alpha^4 + \alpha^5$	000111
α^{30}	$1 + \alpha + \alpha^4 + \alpha^5$	110011
α^{31}	$1 + \alpha^2 + \alpha^5$	101001
α^{32}	$1 + \alpha^3$	100100
α^{33}	$\alpha + \alpha^4$	010010
α^{34}	$\alpha^2 + \alpha^5$	001001
α^{35}	$1 + \alpha + \alpha^3$	110100
α^{36}	$\alpha + \alpha^2 + \alpha^4$	011010
α^{37}	$\alpha^2 + \alpha^3 + \alpha^5$	001101
α^{38}	$1 + \alpha + \alpha^3 + \alpha^4$	110110
α^{39}	$\alpha + \alpha^2 + \alpha^4 + \alpha^5$	011011
α^{40}	$1 + \alpha + \alpha^2 + \alpha^3 + \alpha^5$	111101
α^{41}	$1 + \alpha^2 + \alpha^3 + \alpha^4$	101110
α^{42}	$\alpha + \alpha^3 + \alpha^4 + \alpha^5$	010111
α^{43}	$1 + \alpha + \alpha^2 + \alpha^4 + \alpha^5$	111011
α^{44}	$1 + \alpha^2 + \alpha^3 + \alpha^5$	101101
α^{45}	$1 + \alpha^3 + \alpha^4$	100101
α^{46}	$\alpha + \alpha^4 + \alpha^5$	010011
α^{47}	$1 + \alpha + \alpha^2 + \alpha^5$	111001
α^{48}	$1 + \alpha^2 + \alpha^3$	101100
α^{49}	$\alpha + \alpha^3 + \alpha^4$	010110
α^{50}	$\alpha^2 + \alpha^4 + \alpha^5$	001011
α^{51}	$1 + \alpha + \alpha^3 + \alpha^5$	110101
α^{52}	$1 + \alpha^2 + \alpha^4$	101010
α^{53}	$\alpha + \alpha^3 + \alpha^5$	010101
α^{54}	$1 + \alpha + \alpha^2 + \alpha^4$	111010
α^{55}	$\alpha + \alpha^2 + \alpha^3 + \alpha^5$	011101
α^{56}	$1 + \alpha + \alpha^2 + \alpha^3 + \alpha^4$	001001
α^{57}	$\alpha + \alpha^2 + \alpha^3 + \alpha^4 + \alpha^5$	110100
α^{58}	$1 + \alpha + \alpha^2 + \alpha^3 + \alpha^4 + \alpha^5$	111111
α^{59}	$1 + \alpha^2 + \alpha^3 + \alpha^4 + \alpha^5$	101111
α^{60}	$1 + \alpha^3 + \alpha^4 + \alpha^5$	100111
α^{61}	$1 + \alpha^4 + \alpha^5$	100011
α^{62}	$1 + \alpha^5$	100001
α^{63}	1	100000

• GF(2⁶) daki elemanların minimal polinomları, çizelge 3.4 'te verildiği gibidir:

Çizelge 3.4 : GF(2⁶) elemanları minimal polinomları.

Elemanlar	Minimal Polinomlar
$\alpha, \alpha^2, \alpha^4, \alpha^8, \alpha^{16}, \alpha^{32}$	$1 + x + x^6$
$\alpha^3, \alpha^6, \alpha^{12}, \alpha^{24}, \alpha^{48}$	$1 + x + x^2 + x^4 + x^6$
$\alpha^5, \alpha^{10}, \alpha^{20}, \alpha^{40}, \alpha^{17}, \alpha^{34}$	$1 + x + x^2 + x^5 + x^6$
$\alpha^7, \alpha^{14}, \alpha^{28}, \alpha^{56}, \alpha^{49}, \alpha^{35}$	$1 + x^3 + x^6$
$\alpha^9, \alpha^{18}, \alpha^{36}$	$1 + x^2 + x^3$
$\alpha^{11}, \alpha^{22}, \alpha^{44}, \alpha^{25}, \alpha^{50}, \alpha^{37}$	$1 + x^2 + x^3 + x^4 + x^5 + x^6$
$\alpha^{13}, \alpha^{26}, \alpha^{52}, \alpha^{41}, \alpha^{19}, \alpha^{38}$	$1 + x + x^3 + x^4 + x^6$
$\alpha^{15}, \alpha^{30}, \alpha^{60}, \alpha^{57}, \alpha^{51}, \alpha^{39}$	$1 + x^2 + x^4 + x^5 + x^6$
α^{21}, α^{42}	$1 + x + x^2$
$\alpha^{23}, \alpha^{46}, \alpha^{29}, \alpha^{58}, \alpha^{53}, \alpha^{43}$	$1 + x + x^4 + x^5 + x^6$
$\alpha^{27}, \alpha^{54}, \alpha^{45}$	$1 + x + x^3$
$\alpha^{31}, \alpha^{62}, \alpha^{61}, \alpha^{59}, \alpha^{55}, \alpha^{47}$	$1 + x^5 + x^6$

- O halde, n=63 uzunluklu tüm ikili primitif BCH kodlarının parametreleri ve üreteç polinomları, aşağıdaki gibidir.

n=63 uzunluklu tüm ikili primitif BCH kodlarının parametreleri ve üreteç

polinomları Çizelge 3.5 'te verilmiştir.

Çizelge 3.5 : n=63 uzunluklu ikili primitif BCH kod parametreleri.

n	k	t	g(x)
63	57	1	$g_1(x) = 1 + x + x^6$
	51	2	$g_2(x) = (1 + x + x^6)(1 + x + x^2 + x^4 + x^6)$
	45	3	$g_3(x) = (1 + x + x^2 + x^5 + x^6)g_2(x)$
	39	4	$g_4(x) = (1 + x^3 + x^6)g_3(x)$
	36	5	$g_5(x) = (1 + x^2 + x^3)g_4(x)$
	30	6	$g_6(x) = (1 + x^2 + x^3 + x^5 + x^6)g_5(x)$
	24	7	$g_7(x) = (1 + x + x^3 + x^4 + x^6)g_6(x)$
	18	10	$g_{10}(x) = (1 + x^2 + x^4 + x^5 + x^6)g_7(x)$
	16	11	$g_{11}(x) = (1 + x + x^2)g_{10}(x)$
	10	13	$g_{13}(x) = (1 + x + x^4 + x^5 + x^6)g_{11}(x)$
	7	15	$g_{15}(x) = (1 + x + x^3)g_{13}(x)$

- t-hata düzelten, $n = 2m - 1$ uzunluklu bir BCH kodunun tanımından da anlaşılacağı gibi, her kod polinomunun kökleri $\alpha, \alpha^2, \dots, \alpha^{2^t}$ ve bunların eşlenikleridir.
- Katsayıları GF(2) den alınan,

$$v(x) = v_0 + v_1x + \dots + v_{n-1}x^{n-1}$$

polinomu ele alınsın.

Eğer $v(x)$ polinomu,

$\alpha, \alpha^2, \dots, \alpha^{2^t}$ yi kök kabul ederse, $v(x)$ polinomu, $\alpha, \alpha^2, \dots, \alpha^{2^t}$ nin $\phi_1(x), \phi_2(x), \dots, \phi_{2^t}(x)$ minimal polinomlarıyla bölünebilir. Çünkü, örneğin $v(x)$ polinomu, $\phi_1(x)$ ile bölünürse,

$$v(x) = \phi_1(x)q(x) + r(x), \text{ } \text{degr}(x) < \text{deg}\phi_1(x) \quad (3.5)$$

olur.

Denklemde α yerleştirilerek,

$v(\alpha) = \phi_1(\alpha)q(\alpha) + r(\alpha)$ elde edilir.

$$v(\alpha) = \phi_1(\alpha) = 0$$

olduğundan,

$r(\alpha) = 0$ elde edilir.

Eğer $r(x) \neq 0$ ise $\text{degr}(x) < \text{deg}\phi_1(x)$ olmalıdır. Bu ise $\phi_1(x)$ in tanımına aykırıdır. Yani, $\phi_1(x)$ in minimal oluşuna aykırıdır. Dolayısıyla, $r(x) = 0$ olmak zorundadır.

Sonuç olarak, $\phi_1(x)|v(x)$ bulunur. Diğerleri de benzer tarzda gösterilebilir.

$\phi_1(x)|v(x), \phi_2(x)|v(x), \dots$, ten $\phi_1(x), \phi_2(x), \dots, \phi_{2^t}(x)$ lerin en küçük ortak katı da $v(x)$ polinomunu böler.

$g(x) = \text{ekok}\{\phi_1(x), \phi_2(x), \dots, \phi_{2^t}(x)\}$ olup $g(x)$ üreteç polinomu, $v(x)$ polinomunu böler. Böylece $v(x)$, bir kod polinomudur. Buradan yararlanılarak, $n = 2^m - 1$ uzunluklu, t -hata düzelten BCH kodu tanımlanabilir:

İkili bir

$$v = (v_0, v_1, \dots, v_{n-1})$$

sıralı $n - 1$ lisinin bir kodsözcüğü olması için gerek ve yeter koşul,

$$v(x) = (v_0, v_1x, \dots, v_{n-1}x^{n-1})$$

polinomunun

$$\alpha, \alpha^2, \dots, \alpha^{2t}$$

leri, kök kabul etmesidir.

Bu tanım yardımıyla, kodun minimum mesafesi de kolaylıkla belirlenebilir:

$n = 2^m - 1$ uzunluklu, t -hata düzelten bir BCH kodunun bir kod polinomu,

$$v(x) = (v_0, v_1x, \dots, v_{n-1}x^{n-1})$$

olsun. $1 \leq i \leq 2t$ için α_i elemanı $v(x)$ polinomunun bir kökü olduğundan,

$$v(\alpha^i) = (v_0 + v_1\alpha^i + v_2\alpha^{2i} + \dots + v_{n-1}\alpha^{(n-1)i}) = 0 \quad (3.6)$$

olur.

$$(v_0, v_1, \dots, v_{n-1}) \begin{bmatrix} 1 \\ \alpha^i \\ \alpha^{2i} \\ \vdots \\ \alpha^{(n-1)i} \end{bmatrix} = 0, \quad (1 \leq i \leq 2t) \quad (3.7)$$

ise $(v_0, v_1, \dots, v_{n-1})$ ile $(1, \alpha^i, \alpha^{2i}, \dots, \alpha^{(n-1)i})$ iç çarpımının 0 olması demektir.

Bütün bunlar biraraya getirildiğinde H matrisi şu şekilde yazılabilir;

$$H = \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \dots & \alpha^{n-1} \\ 1 & \alpha^2 & (\alpha^2)^2 & (\alpha^2)^3 & \dots & (\alpha^2)^{n-1} \\ 1 & \alpha^3 & (\alpha^3)^2 & (\alpha^3)^3 & \dots & (\alpha^3)^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha^{2t} & (\alpha^{2t})^2 & (\alpha^{2t})^3 & \dots & (\alpha^{2t})^{n-1} \end{bmatrix} \quad (3.8)$$

gibi $2t \times n$ matris elde edilir. Bu matriste $(\alpha^3)^2$ dendiğinde; 3. satır, 3. sütun elemanı anlaşılır. Yani $(\alpha^i)^j$ elemanı; i . satır, $(j + 1)$. sütun elemanıdır. O zaman (3.8) eşitliğinden dolayı,

$$v = (v_0, v_1, \dots, v_{n-1})$$

bir kodsözcüğü ise t -hata düzelten BCH kodunda,

$$v.H^T = 0 \quad (3.9)$$

olur.

Diğer yandan, eğer $v = (v_0, v_1, \dots, v_{n-1})$ sıralı $n - l$ isi, $v.H^T = 0$ koşulunu, yani koşul (3.7)'i sağlarsa, (3.5) ve (3.6)'den dolayı, $1 \leq i \leq 2t$ için α_i ler, $v(x)$ polinomunun kökü olurlar. Böylece de v , t -hata düzelten BCH kodunun bir

kodsözcüğü olur. Böylece sözkonusu kod, H matrisinin null (sıfır) uzayıdır. Burada H , kodun eşlik-denetim matrisidir. Eğer i ve j değerleri (pozitif tam sayılar) için α^j , α^i nin bir eşleniği ise,

$$v(\alpha^j) = 0 \Leftrightarrow v(\alpha^i) = 0 \quad (3.10)$$

dır. Bunun anlamı; $v = (v_0, v_1, \dots, v_{n-1})$ ile H nin i . satırının iç çarpımı sıfır ise, v ile H nin j . sütununun iç çarpımının sıfır olması demektir. Bu nedenle, H nin j . satırı atılabilir. O zaman (3.8) ile verilen H matrisi, aşağıdaki şekle indirgenmiş olur:

$$H = \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \dots & \alpha^{n-1} \\ 1 & \alpha^3 & (\alpha^3)^2 & (\alpha^3)^3 & \dots & (\alpha^3)^{n-1} \\ 1 & \alpha^5 & (\alpha^5)^2 & (\alpha^5)^3 & \dots & (\alpha^5)^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha^{2t-1} & (\alpha^{2t-1})^2 & (\alpha^{2t-1})^3 & \dots & (\alpha^{2t-1})^{n-1} \end{bmatrix} \quad (3.11)$$

olur. Burada j ile, α^j nin eşleniği olan α^i deki j anlatılmaktadır. Ayrıca belirtmeli ki; H deki elemanlar, $GF(2^m)$ nin elemanlarıdır. $GF(2^m)$ deki her eleman, $GF(2)$ deki elemanların sıralı m - lisi olarak yazılır ve sütunlara yerleştirilirse, kodun bir binary eşlik-denetim matrisi elde edilir.

Bunu aşağıdaki örnekle açıklayalım:

Örnek 3.16. $n = 2^4 - 1 = 15$ uzunluklu, çift-hata düzelten BCH kodu ele alınsın.

Bu, bir (15, 7)-koddur. $GF(2^4)$ te primitif bir eleman α olsun. Eşlik-denetim matrisi,

$$H = \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 & \dots & \alpha^{14} \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & \alpha^{15} & \alpha^{18} & \dots & \alpha^{42} \end{bmatrix}$$

olur.

$\alpha^{15} = 1$ olduğu bilindiğine göre, sütunlardaki elemanların sıralı-dörtlüler halindeki gösterilişleri H de yazılırsa, kodun binary eşlik-denetim matrisi elde edilir. ($\alpha^{12} = \alpha^{27} = \alpha^{42}$)v.b. (mod n)

3.2.4 İki-Hata Düzelten BCH Kodunu Çözmek

BCH kodları için genel kod çözme algoritması aşağıdaki gibi tanımlanabilir:

Gönderilen Kod Polinomu : $w(x)$

Alınan Polinom : $v(x)$

Hata Polinomu : $e(x)$

olsun.

Dolayısıyla,

$$v(x) = w(x) + e(x) \quad (3.12)$$

tir.

İlk olarak v nin sendromunu bulalım.

$$S(v) = Hv^T = (S_b, S_{b+1}, \dots, S_{b+d-2})^T \quad (3.13)$$

dir.

Burada

$$S_j = v(\alpha^j) = w(\alpha^j) + e(\alpha^j) \quad , \quad (b \leq j \leq b + d - 2) \quad (3.14)$$

dir.

$r \leq t$ tane hata ortaya çıkıyorsa bu durumda,

$$e(x) = \sum_{i=1}^r c_i x^{a_i} \quad (3.15)$$

olur. $a_1, \dots, a_r, \{0, 1, \dots, n-1\}$ in farklı elemanlarıdır.

$$\eta_i = \alpha^{a_i} \in F_{q^m}$$

elemanları, "hata yerleştirme sayıları" olarak adlandırılır.

$$c_i \in F_q^*$$

elemanları, "hata değerleri" adını alır.

Bu nedenle v nin sendromu için,

$$S_j = e(\alpha^j) = \sum_{i=1}^r c_i \eta_i^j \quad (b \leq j \leq b + d - 2) \quad (3.16)$$

elde edilir.

(η_i, c_i) çiftleri $S(v)$ sendromunun koordinantlarıdır. Bu bilinmeyenler, alınan v vektöründen hesaplanarak bulunabilir.

İkili durumda herhangi bir hata, yalnız η_i ile karakterize edilir. Bu durumda c_i lerin tümü 1 dir.

Kod çözme algoritmasının sonraki aşamasında,

$$\prod_{i=1}^r (\eta_i - x) = \sum_{i=0}^r (-1)^i \sigma_{r-i} x^i = \sigma_r - \sigma_{r-1} x + \dots + (-1)^r \sigma_0 x^r \quad (3.17)$$

özdeş polinomu ile tanımlı σ_i katsayıları tanımlanır. Bu nedenle $\sigma_0 = 1$ ve $\sigma_1, \dots, \sigma_r, \eta_1, \dots, \eta_r$ de tanımlı temel simetrik polinomlardır. x in yerine η_i yazarak,

$$(-1)^r \sigma_r S_j + (-1)^{r-1} \sigma_{r-1} S_{j+1} + \dots + (-1) \sigma_1 \eta_i^{r-1} + \eta_i^r = 0, \quad (3.18)$$

elde edilir.

Bu denklem $c_i \eta_i^j$ ile çarpılıp, sonra toplanarak

$$(-1)^r \sigma_r S_j + (-1)^{r-1} \sigma_{r-1} S_{j+1} + \dots + (-1) \sigma_1 S_{j+r-1} + S_{j+r-1} = 0, \\ (i = 1, \dots, r), \quad (J = b, b+1, \dots, b+r-1) \quad (3.19)$$

elde edilir.

BCH kodunun kod çözme algoritması şu şekilde açıklanabilir:

Diyelim ki tasarlanmış mesafesi $d \geq 2t - 1$ olan bir BCH kodu kullanarak gönderilen bir w kodsözcüğünde, en fazla t hata ortaya çıktığı kabul edilsin;

1. Adım: v alıcı sözcüğünün sendromu tanımlanır.

$$S(v) = (S_b, S_{b+1}, \dots, S_{b+d-2})^T, \quad S_j = \sum_{i=1}^r c_i \eta_i^j, \quad (b \leq j \leq b+d-2) \quad (3.20)$$

2. Adım: $S_{j+r} + S_{j+r-1} \tau_1 + \dots + S_j \tau_r = 0, \quad (b \leq j \leq b+r-1)$ denklemler

sisteminde $r \leq t$ olan maximum sayı tanımlanır. τ_i singüler olmayan bir katsayımatrisine sahiptir. Bu nedenle bulunan r sayısı, ortaya çıkan hataların sayısıdır.

Bu durumda hata yerleştiren polinom,

$$S(x) = \prod_{i=1}^r (1 - \eta_i x) = \sum_{i=0}^r \tau_i x^i \quad (3.21)$$

S_j den τ_i katsayıları bulunur.

3. Adım: $S(x)$ deki α nın kuvvetlerinin yerine konulması ile $S(x) = 0$ denklemini çözülür. Buradan η_i hata düzeltme sayısı hata düzeltme sayısı bulunur. (Chien Araştırması)

4. Adım: c_i hata değerlerini tanımlamak için ilk olarak, 1. adımın r tane denklemindeki η_i ifadesi tanımlanır. Bu durumda, $w(x) = v(x) - e(x)$ ifadesinden, gönderilen w kodsözcüğü bulunur.

Örnek 3.17. Tasarlanmış mesafesi $d = 5$ olan bir BCH kodu hesaplınsın. Bu

kod, herhangi tek veya çift hatayı düzeltebilir. $b=1$, $n=15$, $q=2$ $\phi_i(x)$; $\alpha \in F_{16}$ primitif olduğunda, F_2 üzerinde α^i nin minimal polinomunu belirtiyorsa α , $x^4 + x + 1$ in bir köküdür.

Bu durumda;

$$\phi_1(x) = \phi_2(x) = \phi_4(x) = \phi_8(x) = x^4 + x + 1$$

$$\phi_3(x) = \phi_6(x) = \phi_{12}(x) = \phi_9(x) = x^4 + x^3 + x^2 + x + 1$$

olur. Buradan üreteç polinomu

$$g(x) = \phi_1(x)\phi_3(x) = x^8 + x^7 + x^6 + 1$$

olacaktır.

(15, 7)-kodu, eşlik-denetim polinomu ile

$$h(x) = \frac{(x^{15} - 1)}{g(x)} = x^7 + x^6 + x^4 + 1$$

dir. Vektörleri eşleyerek elde ettiğimiz $g(x)$, $xg(x)$, $x^2g(x)$, $x^3g(x)$, $x^4g(x)$, $x^5g(x)$, $x^6g(x)$ taban olarak düşünülebilir ve üreteç matrisi,

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

v alıcı vektörünü 100100110000100 şeklinde veya $v(x) = 1 + x^3 + x^6 + x^7 + x^{12}$ şeklinde bir polinom olduğunu kabul edelim ilk adımı uygulayarak sendromu hesaplayalım.

$$S_1 = e(\alpha) = v(\alpha) = 1,$$

$$S_2 = e(\alpha^2) = v(\alpha^2) = 1,$$

$$S_3 = e(\alpha^3) = v(\alpha^3) = \alpha^4,$$

$$S_4 = e(\alpha^4) = v(\alpha^4) = 1$$

bulunur.

τ_i bilinmeyenlerinde ikinci adım olarak linear denklemler sisteminin en büyük ihtimali

$$S_2\tau_1 + S_1\tau_2 = S_3$$

$$S_3\tau_1 + S_2\tau_2 = S_4$$

veya

$$\tau_1 + \tau_2 = \alpha^4$$

$$\alpha^4\tau_1 + \tau_2 = 1$$

dir.

Bu sistem, singüler olmayan bir katsayı matrisine sahiptir. Buradan 2 hata ortayan çıkmalıdır. Zira, $r = 2$ dir. Bu denklemler sistemini çözelim. $\tau_1 = 1$, $\tau_2 = \alpha$ olarak bulalım. $S(x)$ deki bu değerleri yerine yazıp $\tau_0 = 1$ olduğunu hatırlarsak,

$$S(x) = 1 + x + ax^2$$

elde ederiz.

F_{16} daki kökler gibi, $\eta_1^{-1} = \alpha^8$, $\eta_2^{-1} = \alpha^6$ buluruz. Bununla birlikte $\eta_1 = \alpha^7$, $\eta_2 = \alpha^9$ dur. Kodsözcüğünün hatalarının 8 ve 10 konumlarında ortaya çıkması gerektiğini biliyoruz. Alıcı polinomdaki bu hataları düzeltirsek,

$$\begin{aligned} w(x) &= v(x) - e(x) \\ &= (1 + x^3 + x^6 + x^7 + x^{12}) - (x^7 + x^9) \\ &= (1 + x^3 + x^6 + x^9 + x^{12}) \end{aligned}$$

bulunur. Vektörel formda ise 100100100100100 şeklindedir.

İlk baştaki mesaj, hata düzelten polinomla tekrar elde edilebilir. Gönderilen kod polinomu,

$$w(x)/g(x)$$

tir.

$$\frac{w(x)}{g(x)} = 1 + x^3 + x^4$$

Vektörel olarak da 1001100 mesaj sözcüğü elde edilir.

3.2.5 GF(2¹⁴)'ün Elemanları

$p(x) = 1 + x + x^3 + x^5 + x^{14}$ primitif polinomu kullanılarak oluşturulan $GF(2^{14})$ nin elemanları Çizelge 3.6 'da verildiği gibidir.

Çizelge 3.6 : GF(2¹⁴) elemanları.

α nın Kuvveti	α Cinsinden Polinom Olarak	Binary
0	0	00000000000000
1	1	10000000000000
α	α	01000000000000
α^2	α^2	00100000000000
α^3	α^3	00010000000000
α^4	α^4	00001000000000
α^5	α^5	00000100000000
α^6	α^6	00000010000000
α^7	α^7	00000001000000
α^8	α^8	00000000100000
α^9	α^9	00000000010000
α^{10}	α^{10}	00000000001000
α^{11}	α^{11}	00000000000100
α^{12}	α^{12}	00000000000010
α^{13}	α^{13}	00000000000001
α^{14}	$\alpha^5 + \alpha^3 + \alpha + 1$	11010100000000
α^{15}	$\alpha^6 + \alpha^4 + \alpha^2 + \alpha$	01101010000000
α^{16}	$\alpha^7 + \alpha^5 + \alpha^3 + \alpha^2$	00110101000000
α^{17}	$\alpha^8 + \alpha^6 + \alpha^4 + \alpha^3$	00011010100000
α^{18}	$\alpha^9 + \alpha^7 + \alpha^5 + \alpha^4$	00001101010000
α^{19}	$\alpha^{10} + \alpha^8 + \alpha^6 + \alpha^5$	00000110101000
α^{20}	$\alpha^{11} + \alpha^9 + \alpha^7 + \alpha^6$	00000011010100
α^{21}	$\alpha^{12} + \alpha^{10} + \alpha^8 + \alpha^7$	00000001101010
α^{22}	$\alpha^{13} + \alpha^{11} + \alpha^9 + \alpha^8$	00000000110101
α^{23}	$\alpha^{12} + \alpha^{10} + \alpha^9 + \alpha^5 + \alpha^3 + \alpha + 1$	11010100011010
α^{24}	$\alpha^{13} + \alpha^{11} + \alpha^{10} + \alpha^6 + \alpha^4 + \alpha^2 + \alpha$	01101010001101

3.2.6 GF(2¹⁶)'nin Elemanları

$p(x) = 1 + x^2 + x^3 + x^5 + x^{16}$ primitif polinomu kullanılarak oluşturulan $GF(2^{16})$ nin elemanları Çizelge 3.7 'de verildiği gibidir.

Çizelge 3.7 : GF(2¹⁶) elemanları.

α nın Kuvveti	α Cinsinden Polinom Olarak	Binary
0	0	0000000000000000
1	1	1000000000000000
α	α	0100000000000000
α^2	α^2	0010000000000000

Çizelge 3.7 (devam) : GF(2¹⁶) elemanları.

α^3	α^3	0001000000000000
α^4	α^4	0000100000000000
α^5	α^5	0000010000000000
α^6	α^6	0000001000000000
α^7	α^7	0000000100000000
α^8	α^8	0000000010000000
α^9	α^9	0000000001000000
α^{10}	α^{10}	0000000000100000
α^{11}	α^{11}	0000000000010000
α^{12}	α^{12}	0000000000001000
α^{13}	α^{13}	0000000000000100
α^{14}	α^{14}	0000000000000010
α^{15}	α^{15}	0000000000000001
α^{16}	$\alpha^5 + \alpha^3 + \alpha^2 + 1$	1011010000000000
α^{17}	$\alpha^6 + \alpha^4 + \alpha^3 + \alpha$	0101101000000000
α^{18}	$\alpha^7 + \alpha^5 + \alpha^4 + \alpha^2$	0010110100000000
α^{19}	$\alpha^8 + \alpha^6 + \alpha^5 + \alpha^3$	0001011010000000
α^{20}	$\alpha^9 + \alpha^7 + \alpha^6 + \alpha^4$	0000101101000000
α^{21}	$\alpha^{10} + \alpha^8 + \alpha^7 + \alpha^5$	0000010110100000
α^{22}	$\alpha^{11} + \alpha^9 + \alpha^8 + \alpha^6$	0000001011010000
α^{23}	$\alpha^{12} + \alpha^{10} + \alpha^9 + \alpha^8$	0000000101101000
α^{24}	$\alpha^{13} + \alpha^{11} + \alpha^{10} + \alpha^9$	0000000010110100

4. LDPC (Low Density Parity Check) KODLARI

LDPC (Low-Density Parity-Check)kodları, 1960'lı yıllarda Robert Gallager [19] [20] tarafından doktora tezinde sunulmuş blok kodlardır. LDPC kodları özyinelemeli yöntemlerle çözülür[35,36]. 1960'lı yıllardaki hesaplama gücü yetersizliği nedeniyle, iyi sonuçlar vermesine rağmen uygulanabilir bulunmadığı için kullanılamamıştır. Ancak 2000'li yıllarda hesaplama gücünün artması ile birlikte özellikle Mackay [25] tarafından gündeme getirilmiştir. LDPC kodları günümüz hesaplama gücüne göre uygulanabilir durumdadır.

LDPC kodu, kodlama ve çözme karmaşıklığı bakımından ve elde ettiği sonuçlara göre çok başarılıdır[35,37].

LDPC kodlarının önemli artılarının yanında bazı dezavantajları vardır. Bunlar uygulanacak sistemlere göre değişmektedir[38]. Bu durum, LDPC'nin uygulama ortamına göre uyarlanması gereğini ortaya çıkarmaktadır. Şu anda yapılan çalışmaların çoğu bu yaklaşım çerçevesindedir. Örneğin kullanılan modülasyon türüne , kullanılan kanal türüne ya da haberleşme hiyerarşisine göre LDPC üzerinde değişiklikler yapılmaya gereksinim duyulmaktadır[26,39]. Bu konulara sonuç bölümünde değinilecektir.

Bu bölümde LDPC kodlarına ilişkin kodlayıcı ve kod çözücü yapıları açıklanacak, bunlara ilişkin bir örnek verilecektir. Kod çözücü algoritması olarak mesaj aktarma algoritması kullanılacaktır.

4.1 LDPC Kodların Kodlanması

LDPC kodları doğrusal blok kodlardır[40]. LDPC kodların kodlanması, bilgi dizisi ile G üreteç matrisinin basit anlamda çarpılarak kodlanmış bilginin oluşması esasına dayanır. Bunun için bilgi dizisi k bitlik bloklara ayrılır ve her k bitlik blok, G matrisi ile çarpılarak n bitlik kodlanmış bilgi dizisi elde edilir[37,41]. Buna göre 2^k adet veri bloğu $u=[u_0, \dots, u_{k-1}]$ için, uygun çarpım sonucunda 2^k adet n bitlik kod sözcüğü $c=[c_0, \dots, c_{n-1}]$ elde edilir. Kod sözcükleri n bitlik olmalarına ve 2^n adet farklı bilgi içerebilmelerine karşın, kodlama sonucunda 2^n seçeneğin 2^k adeti

kullanılmaktadır[42]. u'dan c'ye geçiş $c = u_0 * g_0 + \dots + u_{k-1} * g_{k-1}$ dir ve matris biçiminde ise $C = u * G$ şeklinde gösterilebilir. Burada G matrisi Şekil 4.1 'de verilmektedir. Burada $g_0, \dots, g_{k-1}$, G matrisi satırlarını oluşturan k adet n bitlik vektörleri göstermektedir.

$$G = \begin{bmatrix} \text{--} & g_0 & \text{--} \\ \text{--} & g_1 & \text{--} \\ & \vdots & \\ \text{--} & g_{k-1} & \text{--} \end{bmatrix} \quad k \times n$$

Şekil 4.1 : G üreteç matrisi.

G matrisine üreteç matrisi denir ve k rankına sahiptir. Ayrıca

$$G = [I \mid P] \quad (4.1)$$

şeklinde iki matrisin birleşiminden oluşan sistematik biçimde gösterilebilir. Burada I matrisi k x k boyutlarında birim matristir. P matrisi ise k x (n-k) boyutlarında eşlik matrisidir. Bu yapı matrisler üzerinde yapılabilecek ve matris denkleğini bozmayacak işlemler sonucunda elde edilebilir.

Kodlama işlemi sonucunda elde edilen kod sözcüğünün ilk k biti kodlanan veri, sonraki n-k biti ise eşlik amaçlı kullanılan veridir:

$$C = u * G = [u \mid u * P] \quad (4.2)$$

LDPC kodları yukarıda anlatıldığı biçimde kod sözcüğü üretimini gerçekleştirir. LDPC kodlarını diğer doğrusal blok kodlardan ayıran özellik, G üreteç matrisinin yapısıdır[43]. LDPC kodları üreteç matrisinin n ve k değerleri diğer doğrusal blok kodlarının n ve k değerlerine kıyasla çok büyük değerler alır ve üreteç matrisi çok az sayıda '1' içerir.

Genel olarak yapılan çalışmalarda H matrisi üretimi üzerine incelemeler yapılmış ve elde edilen H matrislerinden G üreteç matrisi bulunmuştur[40,44]. H matrisi sistematik biçimde

$$H = [P^T \mid I] \quad (n-k) \times n \quad (4.3)$$

Şeklinde gösterilebilir[45,46]. Burada P^T matrisi , üreteç matrisinin sistematik biçimde gösterildiği yapıda bulunan P eşlik matrisinin evriğidir. H matrisinin özellikleri ve üretilme biçimi bir sonraki bölümde anlatılmaktadır.

4.2 LDPC Kodları Çözülmesi

LDPC kod sözcüklerinin çözümünde H matrisi kullanılır[26][45]. H matrisi

$$C \cdot H^T = 0 \quad (4.4)$$

denklemini sağlayan bir matristir. Bu matrise eşlik kontrol matrisi denir ve genel yapısı Şekil 4.2 'de verilmektedir. Burada $h_0, \dots, h_{n-k-1}$, H matrisi satırlarını oluşturan , n-k adet , Şekil 4.2 'de verilen n bitlik birbirlerinden bağımsız vektörlerdir.

$$H = \begin{bmatrix} - & - & h_0 & - & - \\ - & - & h_1 & - & - \\ & & \vdots & & \\ - & - & h_{n-k-1} & - & - \end{bmatrix} \quad (n-k) \times n$$

Şekil 4.2 : H matrisi gösterilimi.

H matrisi, G üreteç matrisine benzer bir biçimde çok az sayıda 1 içerir. Bazı temel tanımlamalar aşağıda verilmektedir.

Tanım 4.1. Düzenli bir (n,k) LDPC kodu , H matrisinin her bir sütununda W_c adet 1 ve her bir satırında W_r adet 1 içeren matristir. Bu durumda $W_r = W_c \cdot n / (n-k = m)$ olduğu görülecektir.

Tanım 4.2. Kodlama oranı $R = k / n$ dir. Bu 1 sayıları bakımından $R = (W_r - W_c) / W_r$ olarak gösterilebilir. $W_c \geq 3$ olduğu kodlamalar iyi kod olarak kabul edilir.

Tanım 4.3. H matrisi satırlarında ve sütunlarında eşit sayıda 1 içermiyorsa bu kodlamaya düzensiz LDPC kodu denir.

H matrisi üretimi genel olarak iki şekilde yapılmaktadır.

Birinci yöntemde, H matrisi belirli kurallara uygun olarak, rasgele olmadan elde edilmektedir. Bu şekilde elde edilen matrisler düzenli[23] ya da düzensiz matrisler [30,46] olabilirler. Genel olarak düzenli matrislerin analizi daha kolaydır. Ancak çalışma başarımı, düzensiz matrisler kadar iyi değildir. Genel olarak yapılan çalışmalarda, matris üretimi için kullanılan algoritmalar matrisin minimum çevrim

sayısının (girth) maksimum olmasına[24], satırlarda yada sütunlarda bulunan 1'lerin yerlerinin , iki satır yada sütunda birden fazla bir şekilde aynı olmamasına , düzenli matris olacak ise satır ve sütunlarda eşit sayıda '1' olmasına dikkat edilmektedir[47].

Örnek olarak [23] de, H matrisi elde etmek için, temel matrislerin çeşitli permütasyonları birleştirilerek kullanılmaktadır[48]. $W_r=4$ parametresi(satırlardaki 1 sayısı) için H_0 temel matrisi aşağıdaki şekilde oluşturulabilir(Temel matrisler sütunlarında bir adet 1 içermektedir).

$$H_0 = \begin{pmatrix} 11110000000000000000 \\ 00001111000000000000 \\ 00000000111100000000 \\ 00000000000011110000 \\ 00000000000000001111 \end{pmatrix} .$$

Benzer şekilde $W_r=4$ için H_1 ve H_2 temel matrisleri ;

$$H_1 = \begin{pmatrix} 10001000100010000000 \\ 01000100010000001000 \\ 00100010000001000100 \\ 00010000001000100010 \\ 00000001000100010001 \end{pmatrix} ,$$

$$H_2 = \begin{pmatrix} 10000100000100000100 \\ 01000010001000010000 \\ 00100001000010000010 \\ 00010000100001001000 \\ 00001000010000100001 \end{pmatrix}$$

olarak elde edilebilir. $n=20$, $W_c=3$ (sütunlarda bulunan 1 sayısı), $W_r=4$ (satırlarda bulunan 1 sayısı) parametrelerine sahip düzenli H matrisi ;

$$H = \begin{pmatrix} 11110000000000000000 \\ 00001111000000000000 \\ 00000000111100000000 \\ 00000000000011110000 \\ 00000000000000001111 \\ 10001000100010000000 \\ 01000100010000001000 \\ 00100010000001000100 \\ 00010000001000100010 \\ 00000001000100010001 \end{pmatrix}$$

```
00010000001000100010
000000001000100010001
10000100000100000100
01000010001000010000
00100001000010000010
00010000100001001000
00001000010000100001
```

şeklinde , H_0 , H_1 ve H_2 temel matrislerinin birleştirilmesiyle elde edilir[49].

Matris üretiminde önemli noktalardan biri de, matrisin minimum uzunlukta olması çevrim sayısının da maksimum olmasıdır[24]. Bununla ilgili olarak [24] da Bit-Filling algoritması verilmiştir. Algoritmada temel amaç mümkün olan maksimum kodlama oranında mümkün olan maksimum girth'i (minimum çevrim sayısı) vermektir. Bu şekilde, matris elemanlarını kullanarak ilerleyen özyinelemeler birbirlerinden daha bağımsız olacaktır. Bu tür matris kullanan kod çözme algoritmaları daha iyi sonuç vermektedir.

Düzensiz matris üretimi genel anlamda, uygun bir dağılımla tasarlanması durumunda, düzenli matris kullanımına göre hata başarımı açısından daha iyi sonuç vermektedir. Ancak düzensiz matrislerin analizi, düzenli matrislere göre daha zordur.

İkinci yöntemde, üreteç matrisi rasgele bir şekilde üretilmektedir. Genel olarak bu yöntem ilkinde göre daha iyi sonuçlar verir. Ancak üretim şekli, bir çok çalışma ortamı için uygun olmayabilir. Bu matrisler, düzenli[21] ya da düzensiz matrisler olabilir. Düzenli matrislerde rasgele üretime belirli sınırlandırmalar getirilmektedir.

Mackay [25] tarafından yapılan çalışmada, matrislerin rasgele üretim yöntemi verilmektedir. Bu çalışmaya göre, tüm sıfır matristen başlanır ve her bir sütunda W_c adet bit rasgele ters çevrilir[50,51]. Bu sırada mümkün olduğu kadar uniform bir şekilde, W_r ağırlıklı satırlar oluşturulmaya çalışılır. Üretimde sütunlar ve satırlar arasında ikiden fazla bitin aynı olmamasına dikkat edilmelidir[28,52]. Kısa çevrimlerden kaçınılmalıdır. Bu yöntem algoritmik olarak ,

1. $n, k, W_c, W_r, L_{\min}$ (minimum çevrim sayısı) parametrelerini seçin. Bu durumda H matrisi $(n-k) \times n$ şeklinde bir matris olacaktır. Matrisin her bir sütunu W_c adet '1' , her bir satırı W_r adet '1' içerecektir.
2. Sütun sayıcısı i_c 'yi , '0'a ayarlayın.

3. Oluşturulacak H matrisinin i_c . sütununa rasgele elde edilmiş W_c ağırlıklı sütun vektörünü yerleştirin.
4. Bu noktada satır ağırlığı W_r 'den küçük ise ve iki satır ya da sütun arasında birden fazla aynı '1' konumu yoksa sütun sayıcısını bir artırın. Aksi takdirde bir önceki adıma geri dönün.
5. $i_c=n$ olması durumunda algoritma tamamlanmıştır. Aksi durumda ise 3. adıma geri dönün.

şeklinde verilebilir. Görüldüğü gibi yapı tam olarak rasgele değildir. Önemli sınırlandırmalar gelmektedir. Algoritmanın yapısına bakıldığında, algoritmanın çalışma süresinin, özellikle büyük değerler için uzun süreceği ve hatta rasgele değerlere göre kilitlenebileceği görülebilir. Ayrıca üretilen matris büyük bir olasılıkla satır ağırlığı W_r 'den küçük satırlar kalacağı için düzensiz bir matris olacaktır[52]. Düzenli olabilmesi için satır ağırlığı üzerine belirli kısıtlamalar konulmalıdır. Ancak bu durumda algoritmanın kilitlenme olasılığı çok artacaktır[33,53]. Ancak genellikle düzensiz matrisler , düzenli matrislere göre daha iyi sonuçlar verdiği için , algoritma düzenli üretim için çalıştırılmaz.

Genel olarak iki yöntemde de kesin doğru bir noktaya ulaşılamamıştır. Bu konu hala çalışma alanı olarak güncelliğini korumaktadır[53,54]. Rasgele üretim geliştirerek , daha kullanışlı hale getirilebilir. Ayrıca, bir kısmı tamamen rasgele üretilmiş matrislerin , kalan kısmının belirli kurallara uygun şekilde üretilmesi düşünülebilir

Kanaldan alınan gürültülü sözcüklerin , doğru bir şekilde alınmış olması durumunda H^T ile çarpımları 0 matrisini vermelidir[55]. 0 olmaması durumu, gürültü nedeniyle gönderilen kodlanmış verinin bozulması anlamına gelmektedir. Buna göre alınan bir vektörün w adet 1 içerdiğini varsayarsak (bu Hamming ağırlığının w olduğu anlamına gelir), ve 1'lerin yerlerine $l_1 , l_2 , \dots l_w$ dersek, bu konumlara karşılık düşen H^T matrisinin satırlarının toplamı 0 değerini vermelidir[56,57].

Minimum Hamming ağırlığına sahip kod sözcüğünün ağırlığı w_{min} , kodun minimum uzaklığını verir ve d_{min} olarak gösterilir. Bu sayı, minimum kaç adet H^T matrisi satırının toplamının 0 olduğunu söyler[56]. Gürültüsüz bir ortamda, LDPC kod çözücüsü , çok basit olarak çalışır. Çünkü gürültüsüz bir ortamda alınan kodlanmış veri olan n bitlik c vektörü ile H^T matrisinin çarpımı 0 verir ve bu durumda, c vektörünün ilk k biti kodlanmış veri olarak bulunur[58]. Kod çözücülerde asıl sorun

bu çarpımın 0 çıkmaması durumudur. Bu durumda çeşitli kod çözme algoritmaları geliştirilmiştir. Bunların bazıları ;

- Toplama-çarpma algoritması (Sum-product algorithm)
- Minimum-toplam algoritması (Min-sum algorithm)
- İleri-geri algoritması (Forward-backward algorithm)
- BCJR algoritması
- APP yada MAP algoritması
- Kanı-yayılımı algoritması (Belief-propagation algorithm)
- Mesaj-aktarma algoritması (Message-passing algorithm)
-

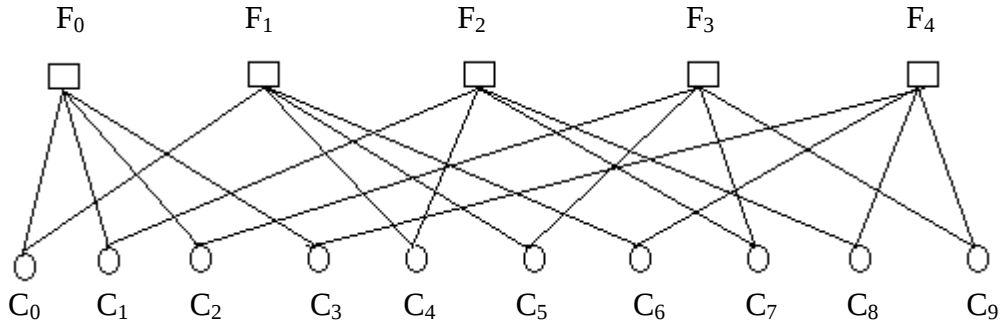
Tüm bu algoritmaların ortak özellikleri özyinelemeli olarak çalışıyor olmalarıdır[59]. Örnek bir kod çözücü algoritmasına geçmeden önce, birçok algorithmada ortak olarak kullanılan bazı kavramlar verilecektir.

4.3 Tanner Grafi

[22] Bu graf gösterilimi, H matrisinin farklı bir gösterilimidir. İki farklı düğüm kümesinden ve bunlar arasındaki dallardan oluşmaktadır. Bu düğümlerden biri, bit ya da değişken düğümü olarak ifade edilir[60]. İkincisi ise kontrol (fonksiyon) düğümü olarak verilir. H matrisindeki $h_{ji}=1$ ise j. kontrol düğümü, i. bit düğümüne bağlanır. Bu işlem 1'den m'inci kontrol düğümüne kadar yapılır. Buna göre $m=n-k$ adet kontrol düğümü ve n adet bit düğümü bulunmaktadır. Şekil 4.3 'te verilen örnek bir H matrisi için Tanner grafi Şekil 4.4 'te verilmektedir.

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Şekil 4.3 : (10,5) kodu H matrisi.



Şekil 4.4 : (10,5) kodu H matrisine ait Tanner grafi.

Verilen örnekte H matrisi $n=10, k=5, W_c=2$ ve $W_r=4$ parametrelerine sahip bir matristir. Buna göre 10 (n) adet C olarak adlandırılan bit düğümlerine, 5 ($m=n-k$) adet F ile adlandırılan kontrol düğümlerine sahip bir graf elde edilecektir. Grafta bit düğümlerine 2 (W_c) adet , kontrol düğümlerine ise 4 (W_r) adet dal bağlantısı olacaktır[37,61]. Grafta, h_{ij} değerinin 1 olduğu durumlarda F_i kontrol düğümü ile C_j bit düğümü arasında bir dal oluşturulur. Bu işlem 1 olan tüm matris elemanları dikkate alınarak yapılarak graf elde edilir.

Bu graftan eşlik kontrol denklemleri çok kolay bir şekilde görülmektedir. Buna göre;

$$F_0 = C_0 + C_1 + C_2 + C_3$$

$$F_1 = C_0 + C_4 + C_5 + C_6$$

$$F_2 = C_1 + C_4 + C_7 + C_8$$

$$F_3 = C_2 + C_5 + C_7 + C_9$$

$$F_4 = C_3 + C_6 + C_8 + C_9$$

eşlik kontrol denklemleridir. Böylece her bir kontrol düğümü bir eşlik kontrol denklemi tanımlar.

Tanım 4.4. Bir Tanner grafında , l adet dal kullanılarak tekrar aynı düğüme gelinmesi durumuna , l uzunluklu çevrim adı verilir.

Tanım 4.5. Bir Tanner grafında , minimum çevrim uzunluğuna , grafın çevrimi adı verilir. Örnekte verilen grafta mümkün olan en küçük çevrim 4 olduğu için , grafın çevrimi 4'tür.

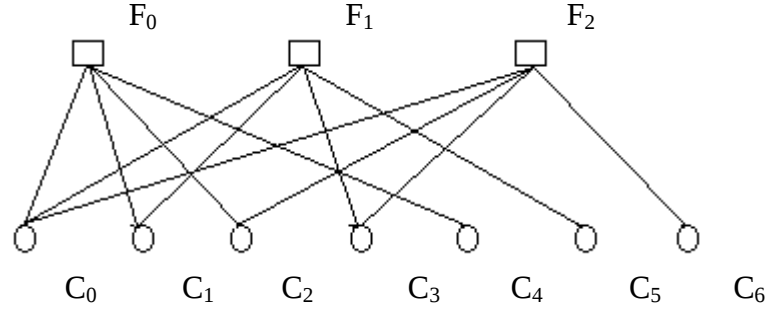
Genellikle bir graftaki küçük çevrimler yok edilmeye çalışılır. Çünkü bu tür çevrimler kod çözücünün çalışmasını olumsuz yönde etkilemektedir[62].

4.4 Mesaj Aktarma Kod Çözücü Algoritması

Kod çözücü algoritmasını basit bir blok kod örneğiyle açıklamak daha kolay olacaktır[61,63]. Bunun için (7,4) Hamming kodu kullanılabilir. Bu koda ait H matrisi ve Tanner grafi Şekil 4.5 ve Şekil 4.6 'da verilmektedir.

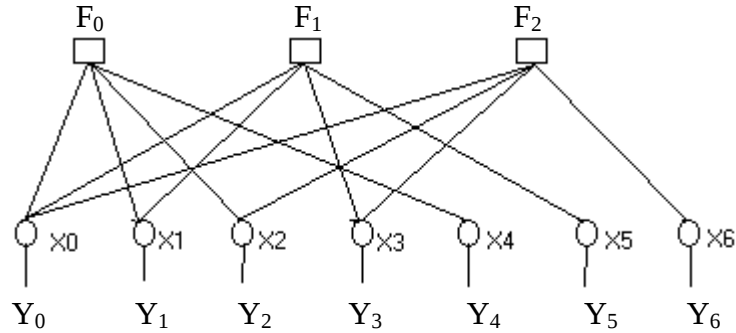
$$H^T = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} P \\ I \end{bmatrix}$$

Şekil 4.5 : (7,4) kodu H matrisi.



Şekil 4.6 : (7,4) kodu H matrisine ait Tanner grafi.

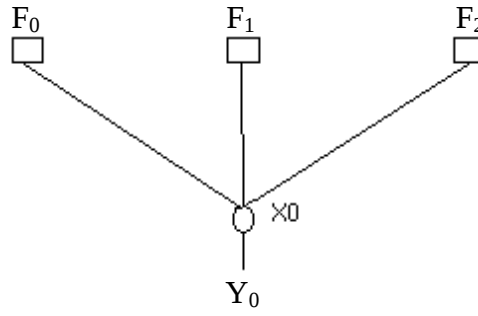
C_k bitleri AWGN kanalına $X_k = (-1)^{C_k}$ olarak gönderilsin. Bu durumda Tanner grafında biraz değişiklik yapılmalıdır. Yeni Tanner grafi Şekil 4.7 'dedir.



Şekil 4.7 : Değiştirilmiş Tanner grafi.

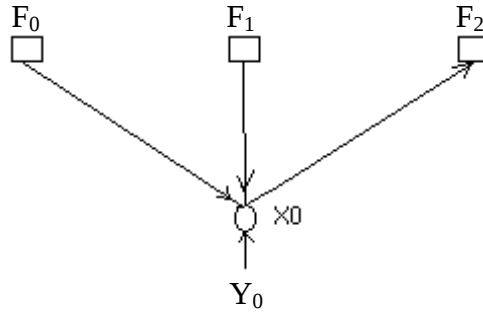
Burada Y_k kanaldan alınan simgedir. $Y_k = X_k + n_k$ dır ve buradaki n_k büyüklüğü AWGN kanal gürültüsünü ifade eder.

Şimdi H 'ın ilk sütununa ilişkin kod için grafın alt grafını düşünelim. Şekil 4.8 'de ilk sütuna ilişkin alt graf verilmektedir.



Şekil 4.8 : H'nin ilk sütununa ilişkin alt graf.

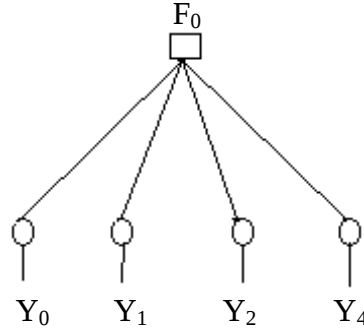
Algoritmanın ilk hesaplamasında X_0 düğümü, sahip olduğu tüm bilgiyi grafta belirtilen bağlantı düğümlerine iletir. Ancak burada daha önce bu bilgiyi almış olan düğümler hariç tutulur. Örnek olarak $X_0 \rightarrow F_2$ durumu Şekil 4.9 'da verilmektedir.



Şekil 4.9 : $X_0 \rightarrow F_2$ için graf.

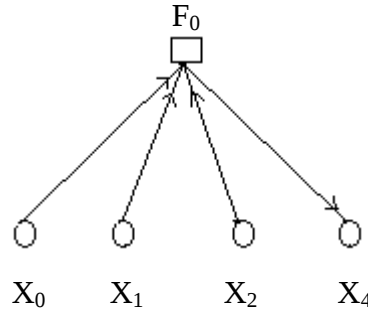
X_0 'dan F_2 'ye iletilen bilgi , X_0 düğümünün bir önceki yarım özyinelemede F_0 ve F_1 düğümlerinden aldığı dış bilgi ve kanaldan gelen bilginin toplamıdır[64,65].

Kod çözücü algoritmasının bir yarım özyinelemesinde hesaplamalar bit düğümlerinden kontrol düğümlerine yapılır ($X_i \rightarrow F_j$)[27]. Kod çözücü algoritmasının diğer yarım özyinelemesinde ise hesaplamalar zıt yönde olarak, kontrol düğümlerinden bit düğümlerine doğru yapılır ($F_j \rightarrow X_i$). Bu hesaplama için H'nin ilk satırına ilişkin alt graf Şekil 4.10'da verilmektedir.



Şekil 4.10 : ($F_j \rightarrow X_i$) ile ilgili alt graf.

Şekil 4.10'dan da anlaşılabileceği gibi, F_0 düğümü sahip olduğu tüm bilgiyi (dış bilgi), grafta belirtilen düğümlere iletir. Ancak daha önce bu bilgiyi almış olduğu düğüm hariç tutulur. Örnek olarak $F_0 \rightarrow X_4$ için durum Şekil 4.11'de gösterilebilir.



Şekil 4.11 : $F_0 \rightarrow X_4$ durumu.

F_0 düğümünden X_4 düğümüne iletilen bilgi, bir önceki yarım özyinelemede X_0 , X_1 ve X_2 noktalarından alınmış olan dış bilgidir.

Bu şekilde, bir özyineleme $X_i \rightarrow F_j$ ve onu izleyen $F_j \rightarrow X_i$ hesaplamalarının bir kümesidir[66]. Kod çözme işlemi maksimum özyineleme sayısına ulaşıldığında ya da $Y * H^T = 0$ eşitliği sağlandığında sonlandırılır.

4.5 Olasılık Bölgesi Kod Çözücü

Bu tür kod çözücülerde, olasılık işlemleri kullanılarak gürültülü sözcükteki hatalar giderilmeye çalışılacaktır[66]. Burada ilk olarak $\Pr(C_i = 1 | \bar{Y}, S_i)$ olasılığı incelenir. Burada S_i , C_i 'yi içeren W_c adet eşlik kontrol denklemini sağlayan C 'deki bitlerin kümesidir. Bundan sonra ise Log-APP oranı ya da Log-Likelihood oranı da denilen LLR'nin sayısal olarak hesaplanması incelenecektir. Bu oran ;

$$\log \left(\frac{\Pr(C_i = 0 | \bar{Y}, S_i)}{\Pr(C_i = 1 | \bar{Y}, S_i)} \right) \quad (4.5)$$

şeklindedir. Bazı temel varsayımlar ve tanımlamalar aşağıda sıralanmıştır [21].

- $\Pr(a_k = 1) = P_k$ olmak üzere bir $a = (a_1, a_2, \dots, a_m)$ ikili dizisi düşünelim ve bu dizi elemanları birbirinden istatistiksel bağımsız olsunlar.
- a 'nın çift sayıda 1 içermeye olasılığı

$$\frac{1}{2} + \frac{1}{2} \prod_{k=1}^m (1 - 2P_k) \text{ dır.}$$

- a 'nın tek sayıda 1 içermeye olasılığı ise

$$\frac{1}{2} - \frac{1}{2} \prod_{k=1}^m (1 - 2P_k) \text{ dır. Bu olasılıkların kanıtı burada verilmeyecektir.}$$

Bundan sonrası için çözücüde kullanılacak bazı notasyonlar verilecektir. H matrisi için

- $R_j = \{ i : h_{ji} = 1 \}$; j 'inci satırdaki 1'lerin sütun konumlarının kümesi
- $R_{j|i} = \{ i' : h_{ji'} = 1 \} \setminus \{i\}$; i konumu haricinde, j 'inci satırdaki 1'lerin sütun konumlarının kümesi
- $C_i = \{ j : h_{ji} = 1 \}$; i 'inci sütundaki 1'lerin satır konumlarının kümesi
- $C_{i|j} = \{ j' : h_{ji'} = 1 \} \setminus \{j\}$; j konumu haricinde, i 'inci sütundaki 1'lerin satır konumlarının kümesi
- $P_i = \Pr(C_i = 1 | Y_i)$
- $C_{kj} = C_i$ 'yi içeren j 'inci eşlik kontrol denklemindeki k 'inci bit
- $Y_{kj} = C_{kj}$ 'ye karşılık hattan alınan örnektir. Bizim için $Y_{kj} = (-1)^{C_{kj}} + n$ olarak tanımlanmaktadır.
- $P_{kj} = \Pr(C_{kj} = 1 | Y_{kj})$

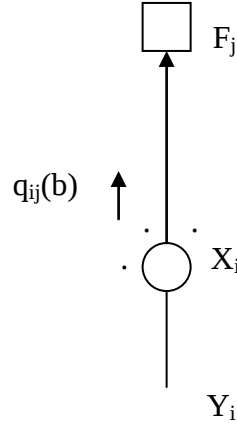
Bu noktada APP oranına ilişkin bir teorem verilmektedir. Teoremin kanıtı verilmeyecektir. Teoreme göre [53]

$$\frac{\Pr(C_i = 0 | \bar{Y}, S_i)}{\Pr(C_i = 1 | \bar{Y}, S_i)} = \frac{(1 - P_i)}{P_i} \frac{\prod_{j \in C_i} \left(1 + \prod_{i' \in R_{ji}} (1 - 2P_{i'j}) \right)}{\prod_{j \in C_j} \left(1 - \prod_{i' \in R_{ji}} (1 - 2P_{i'j}) \right)} \quad (4.6)$$

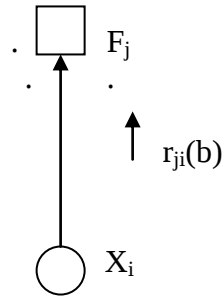
dir.

Ancak teoremden $\bar{Y}$ 'den alınan örneklerin istatistiksel olarak bağımsız olduğu varsayılmaktadır.

- $q_{ij}(b)$: $b \in \{0,1\}$ olmak üzere, $C_i=b$ olasılığına bakılmaksızın, X_i düğümünden F_j düğümüne iletilen mesaj (dışsal bilgi) olarak tanımlanır.



- $r_{ji}(b)$: F_j düğümünden X_i düğümüne iletilen mesaj olarak tanımlanır.



Verilen tanımları birleştirirsek ;

$$R_{ji}(0) = \frac{1}{2} + \frac{1}{2} \prod_{i' \in R_{ji}} (1 - 2P_{i'j}), \quad (4.7)$$

$$R_{ji}(1) = \frac{1}{2} - \frac{1}{2} \prod_{i' \in R_{ji}} (1 - 2P_{i'j}), \quad (4.8)$$

$$\frac{\Pr(C_i = 0 | \bar{Y}, S_i)}{\Pr(C_i = 1 | \bar{Y}, S_i)} = \frac{(1 - P_i)}{P_i} \frac{\prod_{j \in C_i} (r_{ji}(0))}{\prod_{j \in C_j} (r_{ji}(1))}, \quad (4.9)$$

$$q_{ij}(0) = (1 - P_i) \prod_{j' \in C_{ij}} r_{j'i}(0), \quad (4.10)$$

$$q_{ij}(1) = P_i \prod_{j' \in C_{ij}} r_{j'}(1), \quad (4.11)$$

şeklinde yazılabilirler[67]. Burada görülebildiği gibi , algoritma q_{ij} ve r_{ji} parametreleri arasında ileri ve geri özyineleme yaparak çevrimi tamamlar.

Algoritma verilmeden önce AWGN kanala özgü düzeltmeler yapılabilir. Bunun için $Y_i = X_i + n_i$ ve $\Pr(X_i = +1) = \Pr(X_i = -1) = 0.5$ olduğunu varsayarsak,

$$\Pr(X_i = x | y) = \frac{1}{1 + e^{-2yx/\sigma^2}} \text{ olarak bulunur.}$$

Bu eşitliğin kanıtını ;

$$\begin{aligned} \Pr(X_i = x | Y) &= \frac{P(Y | X_i = x) \Pr(X_i = x)}{P(Y)} \\ &= \frac{\frac{1}{2} e^{-(y-x)^2 / \sigma^2}}{\frac{1}{2} e^{-(y-1)^2 / 2\sigma^2} + \frac{1}{2} e^{-(y+1)^2 / 2\sigma^2}} \\ &= \frac{e^{xy / \sigma^2}}{e^{y / \sigma^2} + e^{-y / \sigma^2}} \\ &= \frac{1}{e^{y(1-x) / \sigma^2} + e^{-y(1-x) / \sigma^2}} \\ &= \frac{1}{1 + e^{-2yx / \sigma^2}} \end{aligned}$$

şeklinde verebiliriz.

4.6 Mesaj Aktarma Algoritmasının Özeti

(0) Başlangıç

$$q_{ij}(0) = 1 - P_i = \Pr(x_i = +1 | Y_i) = \frac{1}{1 + e^{-2y / \sigma^2}} \quad (C_i = 0) \quad (4.12)$$

$$q_{ij}(1) = P_i = \Pr(x_i = -1 | Y_i) = \frac{1}{1 + e^{2y / \sigma^2}} \quad (C_i = 1) \quad (4.13)$$

(1)

$$r_{ji}(0) = \frac{1}{2} + \frac{1}{2} \prod_{i' \in R_j} (1 - 2q_{i'j}(1)) \quad (4.14)$$

$$r_{ji}(1) = 1 - r_{ji}(0) \quad (4.15)$$

(2)

$$q_{ij}(0) = K_{ij}(1 - P_i) \prod_{j' \in C_i / j} r_{j'i}(0) \quad (4.16)$$

$$q_{ij}(1) = K_{ij} P_i \prod_{j' \in C_i / j} r_{j'i}(1) \quad (4.17)$$

olup burada K_{ij} katsayısı $q_{ij}(0) + q_{ij}(1) = 1$ eşitliğinin sağlanması için kullanılır.

(3)

Her i için

$$Q_i(0) = K_i (1 - P_i) \prod_{j \in C_i} r_{ji}(0) \quad (4.18)$$

$$Q_i(1) = K_i P_i \prod_{j \in C_i} r_{ji}(1) \quad (4.19)$$

Burada K_i katsayıları $Q_i(0) + Q_i(1) = 1$ eşitliğinin sağlanması için kullanılır.

(4)

Her i için

$$C_i = \begin{cases} 1, & Q_i(1) > 0.5 \text{ ise} \\ 0, & Q_i(1) \leq 0.5 \text{ ise} \end{cases} \quad (4.20)$$

dir.

$\hat{c}H^T = \bar{0}$ ise ya da maksimum özyineleme yapılmış ise çözücü işlemini tamamlamıştır ($\hat{c}$ vektörü 4. adımda karar verilen n adet C_i bitlerinin birleştirilmesinden oluşan vektördür). Aksi durumda birinci adıma gidilip, işlemler tekrar yapılır.

4.6.1 Örnek Uygulama

Şekil 4.12 ile verilen H matrisini kullanan $(8,4)$ kodunun ele alalım . Kodun minimum uzaklığı $d_{\min} = 4$ tür.

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Şekil 4.12 : $(8,4)$ kodu H matrisi.

Koda ait eşlik kontrol denklemleri ;

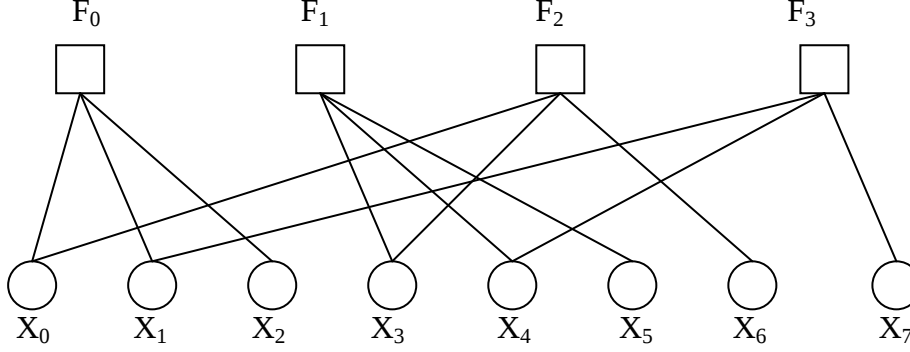
$$C_2 = C_0 + C_1$$

$$C_5 = C_3 + C_4$$

$$C_6 = C_0 + C_3$$

$$C_7 = C_1 + C_4 \text{ dır.}$$

H'a uygun Tanner grafi Şekil 4.13 'te gösterilmektedir.



Şekil 4.13 : (8,4) kodu H matrisine ait Tanner grafi.

Örnek olarak, Bir kod sözcüğü seçilecek , bu kod sözcüğüne gürültü eklenip kod çözme işlemleri uygulanarak gürültüsüz veri elde edilmeye çalışılacaktır.

$\bar{c} = (1,0,1,0,1,1,1,1)$ kod sözcüğü olarak seçilirse ve $X_i = (-1)^{c_i}$ olması durumunda, $\bar{x} = (-1,+1,-1,+1,-1,-1,-1,-1)$ olacaktır. $\bar{x}$ vektörüne kanal zayıflatmaları ve kanal gürültüsü eklenmesi durumunda $\bar{Y} = (+0.2,+0.2,-0.9,+0.6,+0.5,-1.1,-0.4,-1.2)$ olduğunu varsayalım. Bu durumda Y_0 ve Y_4 işaretlerinde , işaret hatası oluşmuştur. Oluşan $\bar{Y}$ vektörü değerlerini kod çözme algoritmasına girersek ;

($\sigma^2 = 0.5$ varsayılıyor)

- Başlangıç işlemi

$$\{q_{ij}(1)\}_{i=0}^7 = \{P_i\}_{i=0}^7 ; h_{ij}=1 \text{ olduğu her bir } j \text{ için}$$

$$= \left\{ \frac{1}{1 + e^{2Y_i / \sigma^2}} \right\}$$

$$= (0.31,0.31,0.97,0.038,0.12,0.99,0.83,0.99) \text{ bulunur.}$$

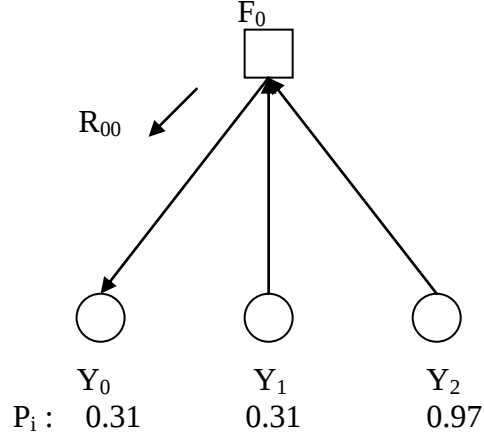
$h_{ij}=1$ olduğu tüm i ve j için , $q_{ij}(0) = 1-q_{ij}(1)$ dir.

- $\{r_{ji}\}$ 'lerin hesaplanması

$$r_{00}(0) = \frac{1}{2} + \frac{1}{2} \prod_{i' \in \{1,2\}} (1 - 2q_{i'0}(1))$$

$$= \frac{1}{2} + \frac{1}{2}(1 - 2(0.31))(1 - 2(0.97))$$

$$= 0.32$$



Şekil 4.14 : R_{00} durumu.

$$r_{01}(0) = \frac{1}{2} + \frac{1}{2}(1 - 2(0.31))(1 - 2(0.97))$$

$$= 0.32$$

$$r_{02}(0) = \frac{1}{2} + \frac{1}{2}(1 - 2(0.31))(1 - 2(0.31))$$

$$= 0.57$$

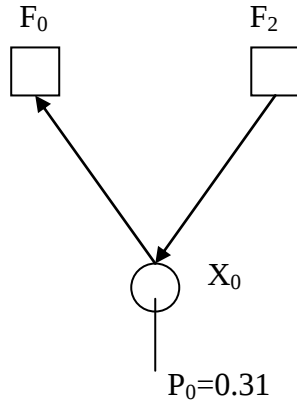
Şekil 4.14 'teki şekilde olduğu gibi devam eder. $r_{ji}(1) = 1 - r_{ji}(0)$ dir.

- $\{q_{ji}\}$ 'lerin hesaplanması

$$\tilde{q}_{00}(0) = (1 - P_0) \prod_{j' \in \{2\}} r_{j'0}(0)$$

$$= (1 - 0.31)(0.22)$$

$$= 0.15 \text{ dir.}$$



Şekil 4.15 : q_{00} gösterilimi.

$$\begin{aligned}\tilde{q}_{00}(1) &= P_0 \prod_{j' \in \{2\}} r_{j'0}(1) \\ &= (0.31)(0.78) \\ &= 0.24\end{aligned}$$

$$\Rightarrow q_{00}(0) = 0.15 / (0.15 + 0.24) = 0.38$$

$$\Rightarrow q_{00}(1) = 0.24 / (0.15 + 0.24) = 0.62$$

Şekil 4.15 'teki şekilde olduğu gibi devam eder.

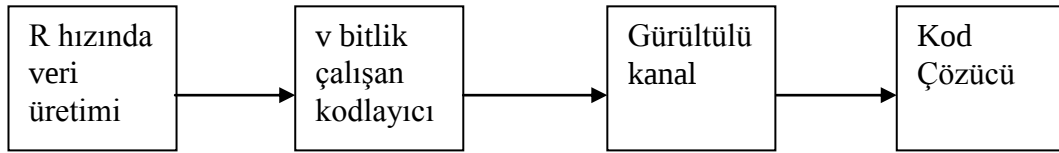
- Q_i 'ler , ek katsayı ile $\{q_{ij}\}$ 'lerin çarpımından elde edilebilir.

Tüm bu işlemleri maksimum 10 özyineleme kullanarak yaptığımızda elde edilen sonuçlar Ek 1'de verilmektedir. Sonuçlardan da anlaşılacağı üzere algoritma 7. özyineleme değerinde doğru sonucu bulmuştur. Normalde böyle bir durumda algoritma 7. özyinelemede durdurulur. Burada örnek oluşturması açısından algoritma devam ettirilmiştir. Sonuçta görüldüğü gibi algoritmanın devam etmesine rağmen , 7. özyinelemede elde edilen doğru değer, 10. özyineleme sonucunda da aynı şekilde kalmıştır.

4.7 Kod Çözme Hatasının Analizi

Son yıllarda sayısal haberleşme sistemlerinde güvenilirlik ve verimliliğe olan gereksinim hızlı bir şekilde artmaktadır. Bunun nedeni, uzun mesafeli haberleşme gereksinimi ve veri işlemcilerindeki hız artışıyla birlikte iletişim hızında artışa gereksinim duyulmasıdır. Geleneksel modülasyon ve ses haberleşme teknikleri genellikle düşük veri hızı ve yüksek hata olasılığı vermektedir.

Haberleşme sistemlerinde verimlilik ve güvenilirlik problemine en temel yaklaşım Shannon tarafından verilmiştir.



Şekil 4.16 : Genel haberleşme şeması.

Şekil 4.16 'da kodlayıcı, kaynak bitlerini v bitlik bloklara ayırır. Kanala gönderilen veri gürültü tarafından bozulur ve kod çözücü gürültülü veriyi , algoritmasından geçirerek gerçek veriye ulaşmaya çalışır. Kod çözücünün veriyi yeniden elde etmedeki hata olasılığı P_e genel olarak :

$$e^{-v[E_L(R_\&)+O(v)]} \leq P_e \leq e^{-vE(R_\&)} \quad (4.21)$$

ile sınırlıdır. $E(R_\&)$ ve $E_L(R_\&)$ fonksiyonları kanala bağlıdır. v blok uzunluğu değişkeni kanala bağımlı değildir. $R_\&=0$ olduğunda $E(R_\&)$ ve $E_L(R_\&)$ fonksiyonları pozitiftirler ve 0 olana kadar $R_\&$ ile birlikte azalır. 0 oldukları noktadaki hız $C_\&$ kanal kapasitesidir.

Kodlanacak veri uzunluğu v , haberleşme sisteminin temel değişkenidir. Kanal verimli olarak kullanılırsa $R_\&$, $C_\&$ 'ye yakın olur. Bunun içinde v değişkeni mümkün olduğunca büyük olmalıdır. Basit anlamda, v büyük olması durumunda, kod çözücü bölümü , kodlayıcı bölümünden daha karmaşık işlemler yapar. Bunun için , kod çözücü karmaşıklığını azaltmak amaçlı önemli çalışmalar yapılmıştır.

LDPC kodlarının matematiksel olarak bazı sonuçlarını elde etmek için, kanallar belirli sınırlamalar ve idealleştirmeler kullanılarak düşünülmektedir. Doğal olarak bu tür kanal yaklaşımları ile elde edilen sonuçlar ancak gerçek ortamda bu kanallara yakın özellikte olan kanallarda görülecektir. Sonuç olarak idealleştirilmiş kanallardaki analizler , gerçek kanallar hakkında sınırlı bir bakış açısı verir.

Genelde hata analizinde kullanılacak kanal modeli, BSC (Binary Symmetric Channel Input) kanallardır. Bu kanallarda giriş dizisi 0 ve 1'lerden oluşmaktadır. Çıkış dizisi ise ayırık ya da sürekli alfabeden gelen uygun dizidir. Kanal belleksizdir ve çıkışlar daha önceki girişlerden bağımsızdır. Genel olarak BSC'ler tamamıyla giriş çıkış arasındaki geçiş olasılıkları ile ifade edilirler.

BSC'de kodlama kullanılmazsa , giriş dizisi kanala gönderilecek, alıcıda alınan simgelerden , iletilen giriş dizisini kestirecektir. Kodlama yapılması durumunda ise kodlayıcı ilk olarak kaynaktan bilgi taşıyan diziyi alacak, bunlara belirli eklemeler yapıp kod sözcüğünü oluşturacak ve kanala iletecektir. Bu noktada kodun kodlama oranı tanımlanır. Bu, bir kod sözcüğü için kaynaktan alınan bitin, elde edilen kod

sözcüğündeki bite oranıdır. R ile ifade edilir. Kod sözcüğü uzunluğu n ise , 2^{nR} adet mümkün dizi vardır. Alıcıda olası kod sözcüklerinin bilgisi vardır ve alıcı aldığı gürültülü veriyi n-bitlik kod sözcüklerine ayırır ve buradan nR'lik kaynak dizisini elde eder. Çoğu kod çözücü ilk olarak alıcıda alınan veri hakkında karar vererek iletilen kod sözcüğünü bulur ve sonra hatayı düzeltmek için kod sözcüklerinin bilgisini kullanılır. Bu hızlı karar verme mekanizması çok önemli miktarda bilgi kaybına neden olur. Burada kullanılacak kod çözücü, bu tür hızlı karar vermeden kaçınır ve sonsal olasılıklar ile çalışır. Bu olasılık alınan simgelere uygun, koşullu giriş simge olasılıklarıdır.

$$x_5 = x_1 + x_2 + x_3$$

$$x_6 = x_1 + x_2 + x_4$$

$$x_7 = x_1 + x_3 + x_4$$

x_1	x_2	x_3	x_4	x_5	x_6	x_7
1	1	1	0	1	0	0
1	1	0	1	0	1	0
1	0	1	1	0	0	1

Şekil 4.17 : Eşlik kontrol matrisine ilişkin kontrol denklemleri.

Şekil 4.17 'deki matris doğrusal homojen eşlik kontrol denklemlerini ifade eder. Eşlik kontrol kodlarının kullanımı kodlamayı , kod çözmeye göre kolay gerçekleştirebilir yapar. [31]'de BSC'de kullanılan tipik blok uzunlukları ve kritik hız ile kanal kapasitesi arasındaki hız kullanılırsa, kodlama hatası olasılığı mümkün olduğunca küçük yapılabilir.

Ancak eşlik kontrol kodların kod çözücülerini gerçekleştirmek basit değildir. Bunun için çeşitli çözümler aranmaktadır.

LDPC kodları çok sayıda 0 ve az sayıda 1 içeren matris ile tanımlanır. (n,j,k) da n, kodlanmış verinin blok uzunluğu, j matrisin sütunlarındaki 1'lerin sayısı ve k'da satırlardaki 1'lerin sayısıdır. LDPC kodlar verilen blok uzunluğu için kod çözme hatası olasılığını küçültmede optimum değildir. Ancak kullanılabildikleri maksimum hızın kanal kapasitesi ile sınırlı olduğu gösterilebilir. Aynı zamanda çözme basitliği bu dezavantajını örter.

2 sözcük arasındaki uzaklık farklı oldukları bitlerin sayısıdır. Bir kod sözcüğünü diğer tüm kod sözcüklerinden ayıran uzaklık kümesi, bir kodda önemli bir

değişkendir. LDPC'de tüm kod sözcüklerinin, diğer kod sözcükleriyle aynı uzaklık kümesine sahip olduğu gösterilebilir[30,68].

4.7.1 Uzaklık Fonksiyonları

Eşlik kontrol kodların uzaklık fonksiyonu $N(l)$, l ağırlıklı kod sözcüğünün sayısı olarak tanımlanır. Eşlik kontrol kodunun grup özelliğinden, herhangi bir verilen kod sözcüğünden l uzaklığındaki kod sözcüğünün sayısı olarak ta tanımlanabilir. Bir kodun minimum uzaklığı, $N(l) \neq 0$ için $l > 0$ olmak üzere, l 'nin en küçük değeri olarak tanımlanabilir (D olarak gösterilir). Blok uzunluğu n ve kodlama oranı R olarak verilen bir kodda D 'yi mümkün olduğu kadar büyük ve D 'den az büyük l 'ler için de $N(l)$ 'yi mümkün olduğu kadar küçük yapmak arzu edilir. $N(l)$ 'nin hata olasılığına doğrudan etkisi bulunmaktadır. Büyük blok uzunluklu eşlik kontrol kodları için uzaklık fonksiyonunu ve hatta minimum uzaklığı hesaplamak uygulanamaz. Çünkü çok büyük sayıda kod sözcüğü olacaktır. Kodun ortalama uzaklık fonksiyonunu analiz etmek daha basittir.

Teorem 4.1. $\overline{N(l)}$: n ve R 'li bir kodda l ağırlıklı kod sözcüklerinin ortalama sayısı $\overline{N(l)}$ olsun [1].

$$\overline{N(l)} \left(\frac{n}{l} \right) 2^{-n(1-R)} \leq [2\pi n \lambda (1-\lambda)]^{-0.5} \exp[H(\lambda) - (1-R) \ln 2] \text{ dir. Burada}$$

$$\lambda = \frac{l}{n} ,$$

$$H(\lambda) = \lambda \ln \frac{1}{\lambda} + (1-\lambda) \ln \frac{1}{1-\lambda} \text{ şeklinde tanımlanmaktadır.}$$

Teorem 4.2. Minimum uzaklık dağılım fonksiyonu $\Pr(D \leq n\delta)$, $\delta < 0.5$ için aşağıdaki gibi sınırlıdır [19]

$$\Pr(D \leq n\delta) \leq \frac{1}{1-2\delta} \sqrt{\frac{1-\delta}{2\pi n\delta}} \exp n[H(\delta) - (1-R) \ln 2]$$

$$\Pr(D \leq n\delta) \leq 1 .$$

4.7.2 LDPC'nin Uzaklık Özellikleri

Küçük minimum uzaklığa sahip olan bir kodun mahsurlu yerleri çıkartılarak yeni bir kod elde edilebilir. (n,j,k) 'lık bir matris alalım. Burada herbir sütunda j adet '1' ve

her bir satırda k adet '1' vardır. Bu durumda PC matrisi nj/k adet satır içerir. Kodlama oranı ise $R \geq 1 - j/k$ dir. Şekil 4.18'de (20,3,4) koduna ilişkin özel bir matris verilmektedir.

1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0
0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0
0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0
0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0
0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0
0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1

Şekil 4.18 : (10,3,4) kodu H matrisi gösterilimi.

Teorem 4.3. Bir (n,j,k) 'daki her bir kod için, l ağırlıklı dizilerinin sayısı $N_1(l)$ dir ve aşağıdaki gibi sınırlandırılmıştır [1]

$$N_1\left(\frac{n}{k}\mu'(s)\right) \leq \exp \frac{n}{k} \left[\mu(s) - s\mu'(s) + (k-1)\ln 2 \right] .$$

Burada s keyfi bir parametre ve $\mu(s)$ ve $\mu'(s)$ ise aşağıdaki gibi tanımlanmıştır

$$\mu(s) = \ln 2^{-k[(1+e^s)^k + (1-e^s)^k]} \text{ ve } \mu'(s) = \frac{d\mu(s)}{ds} .$$

Referans [35]'te , $l = (n/k)\mu'(s)$ 'ya ayarlama, üstel ifadeyi minimize eder. Böylece en iyi sınırlamanın elde edildiği gösterilir.

Teorem 4.3, l ağırlıklı bazı özel dizilerin bir kod sözcüğü olduğu kod sözcüklerinin kümesinin olasılığını $P(l)$ bulmak için kullanılabilir. Bir kod sözcüğünün tüm permütasyonları eşit olduğundan dolayı $P(l)$ seçilen l ağırlıklı dizilerden bağımsızdır. l ağırlıklı dizileri rasgele seçersek, her bir kod için olasılık $N_1(l)/n_1$ dir. Öyle ki,

seçilen l ağırlıklı dizi n/k eşlik kontrollerin herhangi bir bloğunu sağlayacaktır. Eşlik kontrollerin her bir j bloğu bağımsız olarak seçildiğinden dolayı

$$P(l) = \left[\frac{N_1(l)}{\binom{n}{l}} \right]^j \text{ dir.}$$

Bu durumda uzaklık özelliği ve minimum uzaklık dağılım fonksiyonu $P(l)$ kullanılarak

$$\overline{N_{jk}(l)} \leq \binom{n}{l} P(l) = \binom{n}{l}^{-j+1} N_1(l)^j ,$$

$$\Pr(D \leq n\delta) \leq \sum_{l=2}^{n\delta} \binom{n}{l} P(l) = \sum_{l=2}^n \binom{n}{l}^{-j+1} N_1(l)^j \quad (4.22)$$

ifade edilebilir. Ayrıca

$$\overline{N_{jk}(l)} \leq C(\lambda, n) \exp - nB_{jk}(\lambda); \lambda = \frac{l}{n} \quad (4.23)$$

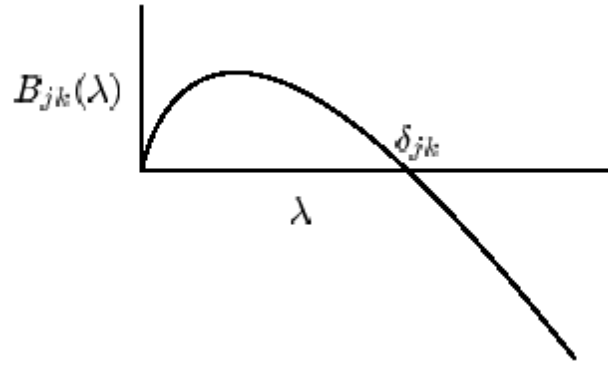
$$B_{jk}(\lambda) = (j-1)H(\lambda) - \frac{j}{k} [\mu(s) + (k-1)\ln 2] + js\lambda ,$$

$$C(\lambda, n) = [2\pi n\lambda(1-\lambda)]^{\frac{j-1}{2}} \exp \frac{j-1}{12\pi\lambda(1-\lambda)}; \lambda = \frac{\mu'(s)}{k} \text{ dir.}$$

(2.2)'i (2.1)'de yerine koyarsak

$$\Pr(D \leq n\delta) \leq \sum_{l=2}^{n\delta} C(\lambda, n) \exp - nB_{jk}(\lambda) \text{ dir.} \quad (4.24)$$

Büyük n 'ler için , (4.23) ve (4.24) denklemleri $B_{jk}(\lambda)$ 'nın davranışı ile belirlenir. $B_{jk}(\lambda)$ 'da ilerde ifade edilecek kod çözücü hata olasılığı ile sınırlıdır. $B_{jk}(\lambda)$ 'yı analiz etmek kolay değildir. Çünkü s ile verilir ve s 'te λ 'nın fonksiyonun içinde saklıdır. $B_{jk}(\lambda)$ Şkiel 4.19 'da gösterilmektedir.



Şekil 4.19 : $B_{jk}(\lambda)$ 'nın dağılımı.

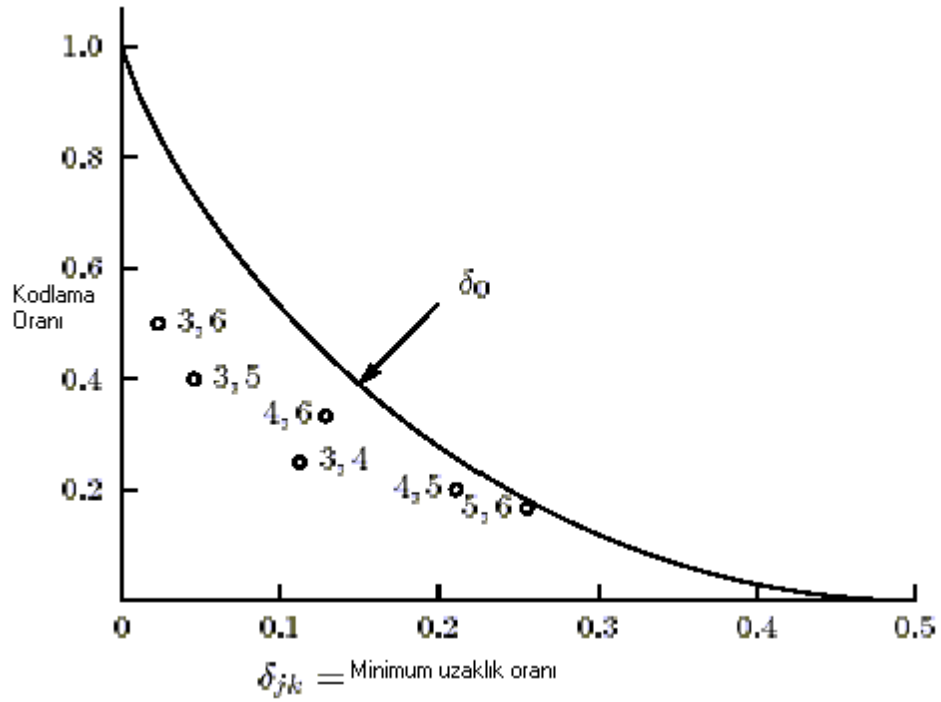
Herhangi bir $\delta > \delta_{jk}$ için 4.24 teki toplamın sınırsız olacağı açıktır, ama minimum uzaklık dağılım fonksiyonu hala 1 ile sınırlıdır.

Teorem 4.4. Bir (n,j,k) için , minimum uzaklık dağılım fonksiyonu hem 4.4 hem de $\Pr(D \leq \delta n) \leq 1$ ile sınırlıdır

$$\Pr(D \leq n\delta) \leq \frac{k-1}{2n^{j-2}} + O\left(\frac{1}{n^{j-2}}\right) + nC(n\delta, n) \exp - nB_{jk}(\delta) \quad (4.25)$$

Denklemdaki ilk ifade 2 ağırlıklı kod sözcüklerinden gelmektedir. Sonraki ifade ise 2'den büyük küçük ağırlıklılardan , son ifade ise büyük ağırlıklılardan gelmektedir.

δ_{jk} ifadesi bir (n,j,k) 'nin tipik minimum uzaklık oranıdır. Şekil 4.20 'de verilmektedir.



Şekil 4.20 : Tipik (n,j,k) için minimum uzaklığın blok uzunluğuna oranı.

Burada minimum uzaklık dağılım fonksiyonunun , kod çözücü hata olasılığı sonuçlarının elde edilmeden önce türetilme nedenini görmekteyiz. Bir gruptaki iki sözcük sadece iki bityle birbirinden farklı ise , kod çözücü hata olasılığı bu iki biti hatalı bir şekilde alma olasılığı ile sınırlıdır. Bu kod uzunluğundan bağımsızdır. Böylece , $n \rightarrow \infty$ için kod çözücü hata olasılığı $1/n$ ile orantılıdır. Bu da minimum uzaklığı 2 olan kodun olasılığıdır. Çok küçük sayıda fakir kodlar , kod çözücünün hata olasılığında baskın çıkar.

N 'ye göre minimum uzaklıklı tipik (n,j,k) kodlarının hata davranış olasılığını belirlemek için, (n,j,k) 'yı düzenleyebiliriz. Bir (n,j,k) 'dan en küçük minimum uzaklıklı kodların yarısının çıkartılması ve kalan yarıdaki her bir kodun olasılığının iki katına çıkartılması durumunda sakıncalı bölümleri çıkartılmış bir (n,j,k) elde edilir.

Sakıncalı bölümleri çıkartılmış (n,j,k) 'nın minimum uzaklığını ele alalım. Bu durumda $\delta_{nj,k}$ değeri , 4.25'teki denklemin sağ tarafındaki değer 0.5 olması için δ 'nın aldığı değer ile alttan sınırlıdır. N 'in artması ile 4.25 denkleminin sınırı δ_{jk} 'daki adım fonksiyonuna yakınsar. Sakıncalı bölümleri çıkartılmış (n,j,k) için 4.26'yı yazabiliriz:

$$\overline{N_{jk}(l)} \left\{ \begin{array}{l} \leq 2C(\lambda, n) \exp - nB_{jk}(\lambda); \lambda \geq \delta_{nj} \\ = 0; \lambda < \delta_{nj} \end{array} \right\} \quad (4.26)$$

Benzer olarak , 4.22 ve 4.26 dan yararlanarak , rasgele kümelerin sakıncalı bölümlerini çıkartabiliriz

$$\overline{N(l)} \left\{ \begin{array}{l} \leq 2[2\pi\lambda(1-\lambda)]^{-0.5} \exp n[H(\lambda) - (1-R)\ln 2]; \lambda \geq \delta_0 \\ = 0; \lambda \leq \delta_0 \end{array} \right\} \quad (4.27)$$

$H(\delta_0) = (1-R)\ln 2$ dir ve n yeteri kadar büyükse

$$\frac{1}{1-2\delta_0} \sqrt{\frac{1-\delta_0}{2\pi n \delta_0}} \leq 0.5 \text{ ifadesi geçerlidir.}$$

4.7.3 Kod Çözücü Hata Olasılığı

İkili blok kodlarda kod çözücü hata olasılığını üstten sınırlamak için bir teknik geliştirilecektir. Kod çözücünün en büyük olabilirlikli, kanalın ikili giriş alfabesi ve keyfi çıkış alfabesine sahip olduğu ve daha sonra tanımlanacağı gibi simetrik olduğu varsayılacaktır.

Bu tekniği geliştirmenin nedenleri 3 adettir. Birincisi; düşük yoğunluklu kodların kapasitesini gösterme gereksinimi, ikincisi; kodun uzaklık özelliği ile kod çözücü hatası olasılığı arasında bir bağlantı sağlaması , üçüncüsü ise analitik olarak daha karmaşık olmasına rağmen fıkırsel olarak daha basitlik oluşturmastır. Bu fıkırsel basitleşme kodun analizini, kanalın analizinden ayırmada yarar.

4.7.4 Simetrik İkili Giriş Kanalları

Aşağıda ifade edilen özelliklere sahip , ayrık zamanlı kanallardır.

1. Giriş alfabesi X , 0 ve 1den oluşur.
2. Çıkış alfabesi Y ya ayrık ya da sürekli sayı kümesinden oluşur.
3. Verilen bir ayrık zamanda çıkış y , sadece mevcut giriş x'e istatistiksel olarak

bağımlıdır.

4. (4.28) ile verilen simetri durumu tüm y çıkışları için sağlanır.

$$P_0(y) = P_1(y) \quad (4.28)$$

Bu denklemde Y sürekli bir kümeysse , $P_x(y)$ koşullu olasılık yoğunluğudur. Y ayrık bir kümeysse $P_x(y)$ koşullu olasılıktır.

4.7.5 Kod Çözme Hatasının Olasılığının Üst Sınırı

İletilen kod sözcükleri $u_0 = x_{10}, x_{20}, \dots, x_{n0}$ ve alınan dizi $v = y_1, y_2, \dots, y_{M-1}$ olsun. En büyük olabirlikli kod çözücü kullanılması durumunda, $P(v|u_j) > P(v|u_0)$ ise kod çözme hatası yapılacaktır. Ayrıca $P(v|u_j) = P(v|u_0)$ ise de kod çözme hatası yapılabilir. Kod çözme hatası olasılığının üst sınır olasılığında, böyle hataların daima yapıldığını varsayalım. Kanalin n adet kullanımları arasındaki bağımsızlığı varsayımını kullanırsak , çözücü hatası için bu durum 4.29'da verildiği gibi olur.

$$\prod_{i=1}^n P_{xij}(y_i) \geq \prod_{i=1}^n P_{xi0}(y_i) , 1 \leq j \leq M-1 \text{ için} \quad (4.29)$$

Böylece , kod çözme hatasının olasılığı (4.29)'da verilen denklemin sağladığı olasılık ile üstten sınırlanmaktadır. Her iki tarafın logaritmasını alırsak

$$\sum_{i=1}^n \ln P_{xij}(y_i) \geq \sum_{i=1}^n \ln P_{xi0}(y_i)$$

olacak ve son olarak, daha sonra gösterileceği gibi , eşitliğin her iki tarafından keyfi bir fonksiyonu çıkartırsak ;

$$\sum_{i=1}^n \ln \frac{f(y_i)}{P_{xij}(y_i)} \leq \sum_{i=1}^n \ln \frac{f(y_i)}{P_{xi0}(y_i)} \quad (4.30)$$

olacaktır.

$f(y)$ fonksiyonuna bir sınırlama konulursa , öyleki tüm y 'ler için hem $P_0(y)$ hem de $P_1(y)$ pozitif ise ve $f(y) = f(-y)$ ise $f(y)$ 'de pozitif olacaktır.

Giriş x_i ve çıkış y_i arasındaki farklılığı $\delta(x_i, y_i)$ şeklinde tanımlarsak;

$$\delta(x_i, y_i) = \ln \frac{f(y_i)}{P_{xi}(y_i)} \quad (4.31)$$

olacaktır.

Ayrıca u ile v arasındaki farklılığı,

$$D(uv) = \sum_{i=1}^n \delta(x_i, y_i) \quad (4.32)$$

şeklinde tanımlayabiliriz.

Tüm bu tanımlamalar sonrasında kod çözme hata olasılığı aşağıdaki gibi sınırlanabilir.

$$P_e \leq \Pr \left\{ \bigcup_{j=1}^{M-1} (D(u_j v) \leq D(u_0 v) \text{ olması } _ \text{ olayı}) \right\} \quad (4.33)$$

Bu sınırlandırmayı , $D(u_0 v) \geq nd$ olmak üzere ayrı sınırlandırmaların toplamı olarak gösterebiliriz. Buna göre ;

$$P_e \leq P_1 + P_2$$

$$P_1 = \Pr \left\{ \bigcup_{j=1}^{M-1} (D(u_0 v) > nd, D(u_j v) \leq D(u_0 v) \text{ olması } _ \text{ olayı}) \right\},$$

$$P_2 = \Pr \left\{ \bigcup_{j=1}^{M-1} (D(u_0 v) \geq nd, D(u_j v) \leq D(u_0 v) \text{ olması } _ \text{ olayı}) \right\} \text{ olacaktır.}$$

Bu noktada P_1 ve P_2 yi

$$P_1 \leq \Pr(D(u_0 v) > nd),$$

$$P_2 \leq \sum_{j=1}^{M-1} \Pr[D(u_0 v) \leq nd; D(u_j v) \leq D(u_0 v)] \text{ şeklinde sınırlayabiliriz.}$$

Bu sınırlandırmaları daha kullanılır hale getirmek için Chernoff kuralı kullanılacaktır. Kuralın ayrıntıları verilmeyecektir. Kural uygulanması durumunda P_1 ve P_2 için elde edilen sonuçlar;

$$g(s) = \int_{-\infty}^{+\infty} P_0(y)^{1-s} f(y)^s dy \text{ olmak üzere}$$

$$P_1 \leq \Pr[D(u_0 v) \geq nd] \leq g(s)^n \exp(-nsd) \quad (4.34)$$

dir . Burada

$$h(r) = \int_{-\infty}^{+\infty} [P_0(y)P_0(-y)]^{(1-r)/2} f(y)^r dy ,$$

$$g(r) = \int_{-\infty}^{+\infty} [P_0(y)]^{(1-r)} f(y)^r dy \text{ olmak üzere}$$

$$P_2 \leq \sum_{l=0}^n N(l) [h(r)]^l [g(r)]^{n-l} e^{-nrd} \quad (4.35)$$

dir.

4.7.6 Kodlar İçin Ortalama Hata Terimi

Şimdi de tüm kod için hata olasılığını düşünelim. $N_j(l)$, u_j kod sözcüğüne l uzaklığındaki kod sözcüklerinin sayısı olsun. Bu durumda P_1 ve P_2 eşitsizliklerinden

$$\overline{P_e} \leq \sum_{j=0}^{M-1} \frac{1}{M} \left\{ g(s)^n e^{-nsd} + \sum_{l=0}^n N_j(l) h(r)^l g(r)^{n-l} e^{-nrd} \right\}; s \geq 0, r \leq 0 \text{ için} \quad (4.36)$$

olur ve u_j kod sözcüğünden l uzaklığındaki kod sözcüklerinin sayısının, j üzerinden ortalaması alınır ;

$$\overline{N(l)} = \frac{1}{M} \sum_{j=0}^{M-1} N_j(l) \text{ olacaktır.}$$

Ortalama ifadesine , bu eşitliği yerleştirmemiz durumunda ;

$$\overline{P_e} \leq g(s)^n e^{-nsd} + \sum_{l=0}^n \overline{N(l)} h(r)^l g(r)^{n-l} e^{-nrd}; s \geq 0, r \leq 0 \text{ olacaktır. Burada}$$

$$g(s) = \int_{-\infty}^{+\infty} P_0(y)^{1-s} f(y)^s dy ,$$

$$h(r) = \int_{-\infty}^{+\infty} [P_0(y)P_0(-y)]^{(1-r)/2} f(y)^r dy \text{ dir.} \quad (4.37)$$

Ortalama hata olasılığı ifadesini hesaplamak zordur. Çünkü n adet ifadenin toplamını içermektedir ve n bir çok durumda çok büyük olabilir. İkinci neden olarak $t, r, s, d, f(y)$ gibi çok sayıda ayrık parametre içermektedir. Bu nedenlerle ifade mümkün olduğunca en uygun hale getirilmelidir. Bunun için ifade optimize edilmelidir. Ancak , bu tür optimizasyonlar, zayıflatma yapılmaksızın gerçekleştirilemez. Ortalama hata olasılığı , n parametresine göre, yaklaşık olarak üstel olarak azalan bir fonksiyondur. Özellikle büyük blok uzunluklu kodlarda n 'nin katsayısı çok önem kazanmaktadır. Sonuç olarak diğer ifade optimizasyon amaçlı olarak ihmal edilebilir. Tabi ki bu

işlem ortalama hata açısından daha kaba bir sonuç verecektir. Ancak şaşırtıcı bir şekilde yeterli olabilir.

Özel kodlar ya da tüm kodlar için $\overline{N(l)}, \overline{N(l)} \leq C(\lambda, n)e^{nB(\lambda)}$; $\lambda = \frac{l}{n}$ şekilde sınırlandırılabilir.

Burada $C(\lambda, n)$ ifadesi yaklaşımın doğru olması için yeterince küçük olması gerekmektedir.

$C_n = \max_{\lambda} C(\lambda, n)$ olmak üzere $C(\lambda, n)$ yerine C_n yi kullanırsak ortalama hata olasılığı

$$\overline{P_e} \leq \exp n[\ln g(s) - sd] + nC_n \max_{\lambda} \exp n[B(\lambda) + \lambda \ln h(r) + (1-r) \ln g(r) - rd] \quad (4.38)$$

olacaktır.

Denklemden n ile üstel olarak değişen temelde iki terim bulunmaktadır. İlk terim $s > 0$ ise d ile birlikte azalır. İkinci terim $r < 0$ ise d ile birlikte artar. Üstleri eşit yapmak için d'yi seçersek , d'deki değişiklik 2 üstten birini artıracaktır. d'nin bu seçimi en büyük üst ifadesinde n'nin katsayısını minimize eder. Bu yolla d'yi kaldırırsak ;

$$\overline{P_e} \leq (1 + C_n) \exp \left[-n \min_{\lambda} E(s, r, \lambda) \right] s \geq 0, r \leq 0 ,$$

$$E(s, r, \lambda) = \frac{r}{s-r} \ln g(s) - \frac{s}{s-r} [B(\lambda) + \lambda \ln h(r) + (1-\lambda) \ln g(r)] \text{ bulunur.} \quad (4.39)$$

Bu sonuçlar hala , $g(r)$ ve $h(r)$ fonksiyonlarının tanımlarının içindeki $f(y)$ fonksiyonuna bağlıdır. $f(y)$, bazı basitleştirmelerle ;

$$f(y) = k \left\{ \frac{\left[P_0(y)^{\frac{1-r}{2}} + P_1(y)^{\frac{1-r}{2}} \right]^2}{P_0(y)^{1-s} + P_1(y)^{1-s}} \right\}^{\frac{1}{s-r}} \text{ şeklinde verilebilir.} \quad (4.40)$$

Burada $\alpha = \frac{g(r)}{g(r) + h(r)}$ olarak alındığında ve λ üzerinden ortalama hatayı

maksimum yapmak için $\alpha = \lambda$ seçilirse, yeni elde edilen ortalama hata olasılığı ;

$$\overline{P_e} \leq (1 + C_n) \exp -nE(s, r) \text{ şekilde olacaktır. Burada}$$

$$E(s, r) = \frac{s}{s-r} \beta(\alpha) - \ln \int_0^\infty (P_0^{1-s} + P_1^{1-s})^{-\frac{r}{s-r}} (P_0^{\frac{1-r}{2}} + P_1^{\frac{1-r}{2}})^{\frac{2s}{s-r}},$$

$$\beta(\alpha) = \min_{\lambda} [-B(\lambda) - \lambda \ln \alpha - (1-\lambda) \ln(1-\alpha)],$$

$$\alpha = \frac{\int_0^\infty (P_0^{1-s} + P_1^{1-s})^{-\frac{r}{s-r}} \left(P_0^{\frac{1-r}{2}} + P_1^{\frac{1-r}{2}} \right)^{\frac{2r}{s-r}} 2(P_0 P_1)^{\frac{1-r}{2}}}{\int_0^\infty (P_0^{1-s} + P_1^{1-s})^{-\frac{r}{s-r}} \left(P_0^{\frac{1-r}{2}} + P_1^{\frac{1-r}{2}} \right)^{\frac{2s}{s-r}}} \text{ dir.} \quad (4.41)$$

Elde ettiğimiz formülleri BSC için kullanabiliriz. Bu özel kanallar için

$$\overline{P_e} \leq (1 + nC_n) \exp - nE(s, r),$$

$$E(s, r) = \frac{s}{s-r} \beta(\alpha) + \frac{r}{s-r} \ln[(1-p)^{1-s} + p^{1-s}] - \frac{2s}{s-r} \ln \left[(1-p)^{\frac{1-r}{2}} + p^{\frac{1-r}{2}} \right] \quad (4.42)$$

$$\alpha = \frac{2[(1-p)p]^{\frac{1-r}{2}}}{(1-p)^{\frac{1-r}{2}} + p^{\frac{1-r}{2}}},$$

$\beta(\alpha) = \min_{\lambda} [-B(\lambda) - \lambda \ln \alpha - (1-\lambda) \ln(1-\alpha)]$ şeklinde yazılabilir. $E(s, r)$ 'nin maksimum değeri $E = \max_{s,r} E(s, r) = p_s \ln \frac{1}{p} + (1-p_s) \ln \frac{1}{1-p} - H(p_s)$ dir. Burada kullanılan p_s ve $H(p_s)$ ifadeleri ise

$$p_s = \frac{\lambda_0}{2} + (1-\lambda_0)p_r, \quad (4.43)$$

$$H(p_s) = B(\lambda_0) + \lambda_0 \ln 2 + (1-\lambda_0)H(p_r) \text{ biçimindedir.} \quad (4.44)$$

Bu denklemlerde kullanılan λ_0 değeri $B(\lambda) + \frac{\lambda}{2} \ln 4p_r(1-p_r)$ değerini maksimum yapan λ değeridir. E değerinin maksimum olması için s ve r değerleri cinsinden

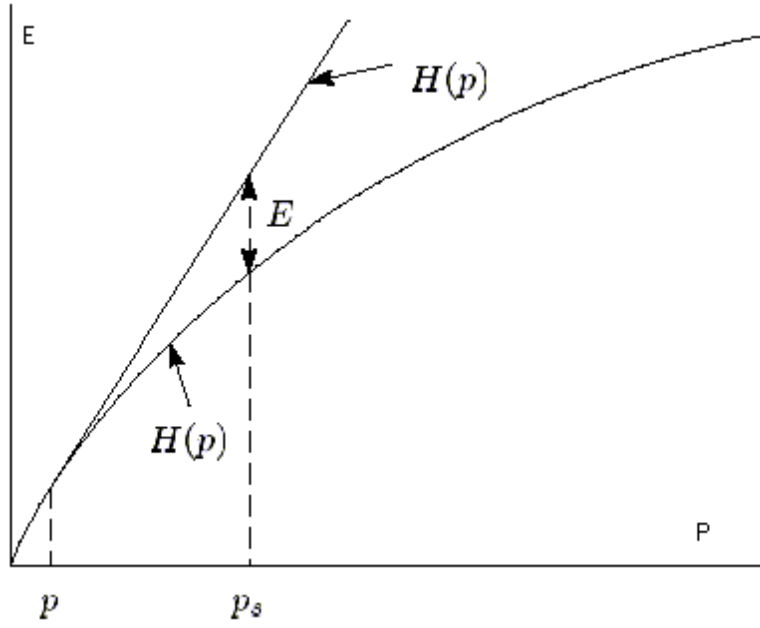
$$p_s = \frac{p^{1-s}}{p^{1-s} + (1-p)^{1-s}}$$

$$p_r = \frac{p^{1-r}}{p^{1-r} + (1-p)^{1-r}} \quad (4.45)$$

olmalıdır.

(4.43) , (4.44) ve (4.45) denklemlerinin çözümü, ikisinin çözümü çok zor olan 3 denklemin aynı zamanda çözülmesini gerektirmektedir. Ancak bu denklemlerin avantajı, kanal geçiş olasılığını kapsamamalarıdır. Böylece bir sonuç elde edilirse , Bu sonuç $p_r \leq p \leq p_s$ sınırları içerisindeki tüm geçiş olasılıkları için geçerli olacaktır.

(4.46)'daki denklemden , p 'nin sınırının $s \geq 0, r \leq 0$ olduğu görülmektedir. Şekil 4.21, E 'nin üstel olarak geometrik kestirimini vermektedir.



Şekil 4.21 : BSC için üstel ifadenin geometrik kestirimi.

$p < p_r$ iken , $E(s,r)$ değeri $r = 0$ için maksimumdur. Çünkü (4.45)'dan $p=p_r$ için $r = 0$ dır. $r = 0$ durumunu (4.42)'deki denklemde yerine koyarsak ve bazı değişiklikler yaparsak

$$\max_{s,r} E(s,r) = \min_{\lambda} -B(\lambda) + \frac{\lambda}{2} \ln \frac{1}{4p(1-p)}; p \leq p_r \quad (4.46)$$

olacaktır.

$p \leq p_r$ için (4.46)'yı minimum yapan λ değeri , p ile birlikte kod minimum uzaklık oranına doğru azalır.

LDPC kodları kodlayıcısının ve kod çözücüsünün çalışmasını etkileyen önemli faktörlerden birisi , kullanılan üreteç matrisidir. Üreteç matrisin elde edilmesine yönelik olarak birçok çalışma yapılmaktadır. Bu çalışmalar 2. bölümde de ifade

edildiği gibi genel olarak iki farklı yöntemle yapılmaktadır. Her iki yöntemin mümkün olduğunca olumlu yönlerini alarak matris üretimi elde edilmesi düşünülebilir.

LDPC kodları önemli noktalarından birisi de kod çözme algoritmalarıdır. Çok çeşitli çözme algoritmaları bulunmaktadır. Bunların bazıları ;

- Toplama-çarpma algoritması (Sum-product algorithm) [26,27]
- Minimum-toplam algoritması (Min-sum algorithm) [21]
- İleri-geri algoritması (Forward-backward algorithm)
- BCJR algoritması
- APP yada MAP algoritması
- Kanı-yayılımı algoritması (Belief-propagation algorithm) [66]
- Mesaj-aktarma algoritması (Message-passing algorithm) [21]
- İteratif kod çözücü algoritması [28,29]

.....

Tüm algoritmaların kullanım alanlarına göre belirli avantaj ve dezavantajları bulunmaktadır. Bazı algoritmalar gürültü gücünün işaret gücüne oranının (SNR) yüksek olduğu kanallarda başarılı iken, bazılara da SNR oranının düşük olduğu kanallarda diğer algoritmalara göre daha başarılı olmaktadır. Ayrıca kanal türüne göre de algoritmaların başarımları değişmektedir.

LDPC blok kodları, kullanılan kanal ve modülasyon türüne göre düzenlenir. Kullanılan kanallara göre LDPC çalışmalarını özetlersek ;

Sönümlemeli kanallar :

Özellikle günümüz GSM kanalları ve diğer telsiz haberleşme kanalları , sönümlemeli kanaldır. Çok güncel bir konu olması nedeniyle üzerinde önemli çalışmalar yapılmaktadır. [34-43] numaralı referanslar sönümlemeli kanallar üzerinde yapılan çalışmalara örnek olarak verilmektedir. Sönümlemeli kanallar üzerindeki çalışmalarda özel olarak kanal $y_i = a_i x_i + n_i$ şeklinde tanımlanmaktadır. Burada n_i AWGN kanalı değişkenini , x_i giriş, a_i sönümleme değişkenlerini ve y_i çıkış değişkenini ifade etmektedir. Bu değişkenler kompleks olabilir. Sönümleme

değişkeni, dağılımı bilindiği varsayılan bir raslantı değişkenidir. Özel bir dağılım olarak Nakagami dağılımı kullanılabilir. Bu dağılım

$$P_A(a) = \frac{2}{\Gamma(m)} \left(\frac{m}{2\sigma^2} \right)^2 a^{2m-1} e^{-ma^2/\sigma^2}$$
 şeklinde verilmektedir ve m değeri değiştirilerek

bir çok sönümlemeli kanalın modellenmesinde kullanılabilir. Denklemden $m > \frac{1}{2}$ dir ve Nakagami değişkeni olarak bilinir. σ^2 varyans büyüklüğü ve $\Gamma(m)$ gamma

fonksiyonudur. Gamma fonksiyonu, $\Gamma(m) = \int_0^{\infty} y^{m-1} e^{-y} dy, m > 0$ şeklinde

tanımlanmaktadır. m=1 için haberleşme sistemlerinde sıkça kullanılan Rayleigh sönümleme modeli tanımlanmaktadır.

Manyetik ortamlar :

Hard-disk gibi manyetik ortamlara kayıt , LDPC gibi hata kodlamaları kullanılarak yapılabilmektedir. Bu konuda yapılan çalışmalar [44-49], [53-56] numaralı referanslarda verilmektedir. Bu çalışmalarda özellikle Hard-disk’lerde kullanılan Reed-Solomon kodunu kuvvetlendirmek amacıyla LDPC uygulanmaya çalışılmaktadır.

Optik ortamlar :

Optik lif kanallara ya da CDROM gibi optik bellek birimlerine , LDPC uygulanmasına yönelik [32,33] numaralı kaynaklarda, örnek çalışmalar verilmiştir. Optik ortamlarda, özellikle fiziksel açıdan CDROM gibi bellek birimlerinde sınır değerlere çıkmıştır. LDPC kullanılarak , daha hatasız ve daha hızlı bir ulaşım gerçekleştirilmeye çalışılmaktadır.

Diğer ortamlar :

[57] ile [64] numaraları arasındaki kaynaklar, çeşitli kanal ve modülasyon türlerine göre yapılan çalışmaları vermektedir.

Katlamalı kodlar ile LDPC’nin birleştirilmesi :

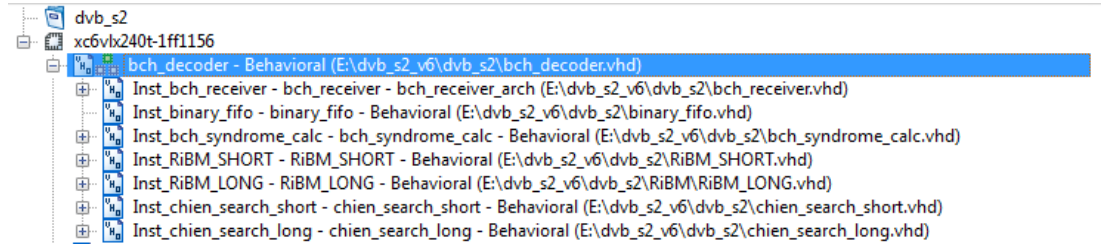
LDPC kodların, blok uzunluğunun artması ile birlikte hata başarımı artmaktadır. Ancak birçok uygulama alanında , blok uzunluğunun büyük olması sorun yaratmaktadır. Bu sorunu kapatmak amacıyla LDPC kodlar katlamalı kodlar ile birleştirilmeye çalışılmaktadır.[50-52], numaralı kaynaklar konuya değinmektedir.

5. BCH KOD ÇÖZÜCÜ GERÇEKLEMESİ

Bu bölümde ikinci bölümden itibaren başlayan ve teorik açıklaması yapılan DVB-S2 BCH kod çözücü kısmı FPGA gerçekleştirilmesi anlatılacaktır.

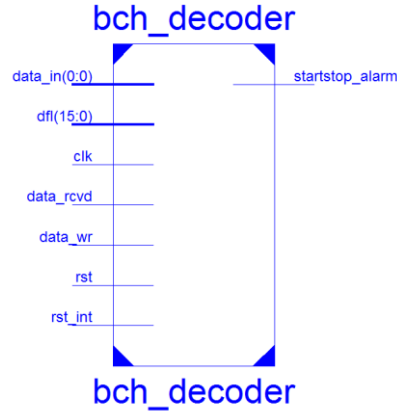
5.1 FPGA Modül Yapıları

5.1.1 bch_decoder Modülü



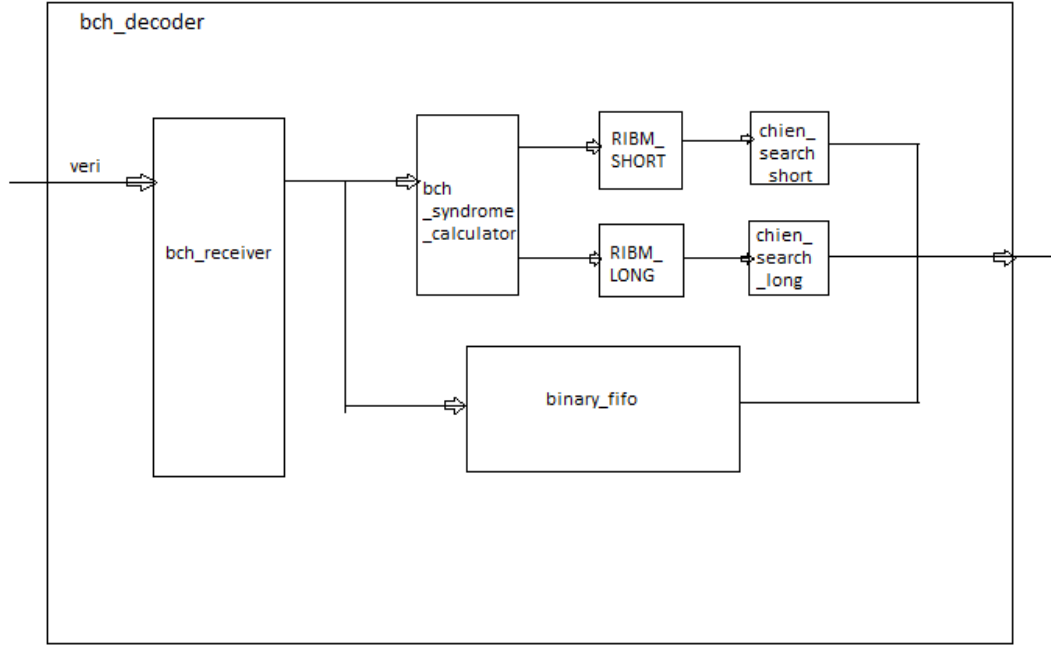
Şekil 5.1 : DVB-S2 BCH FPGA tasarımı hiyerarşisi.

DVB-S2 BCH kod çözücü FPGA tasarımı hiyerarşik yapısı Şekil 5.1 ve Şekil 5.2 'de gösterildiği gibidir.



Şekil 5.2 : bch_decoder modülü RTL şeması.

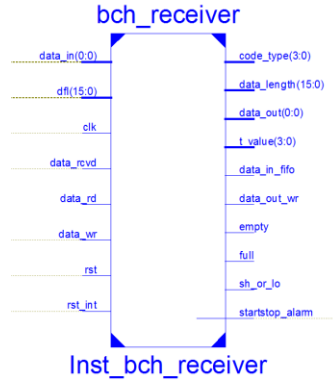
Bu yapıyı blok şemaları Şekil 5.3 'te gösterecek olursak bch_decoder modülü en üst modül olmak üzere :



Şekil 5.3 : DVB-S2 BCH kod çözücü blok yapısı.

BCH kod çözücü bloğuna bölüm 2.2 de bahsedilen bütün uzunluklardaki verilerin gelmesi muhtemeldir. Veri bit halinde her bir saat darbesinde bir bit olmak üzere gelir, gelen verilerin hangi uzunlukta ve hangi kod yapısında olduğu bch_receiver bloğu tarafından ayırt edilir ve bch_syndrome_calculator bloğuna gerekli bilgilerle birlikte bu veri iletilir aynı zamanda da kullanılmak üzere fifo'ya yazılır. BCH kod çözücü aşamalarında daha önce belirtilen sendrom hesaplama aşaması daha önce bilgileri verilen GF çarpıcı yardımıyla yapılır ve $t=12$ için 24 , $t=10$ için 20 ve $t=8$ değeri için 16 tane olma üzere hesaplanır. Bu hesaplama işleminde kısa çerçeveli (short frame) veri için üçüncü bölümde elemanları verilen $GF(2^{16})$ elemanlarından, normal çerçeveli(normal frame) veri için de aynı şekilde $GF(2^{14})$ elemanlarından faydalanılır. Daha sonra kısa ya da normal çerçeve yapısına göre RIBM_short ya da RIBM_long bloğuna giren sendrom değerleri RIBM [21] algoritmasına göre hata bulma ve düzeltme bloğuna gelmeden önce gerekli olan değerler elde edilir. Sonraki aşamada kısa ya da normal çerçeve yapısına göre chien_search_short ya da chien_search_long bloğuna gelerek ilgili veri biti uzunluğunca hata bulma ve düzeltme işlemleri gerçekleşir. Şimdi kısaca anlattığımız bu blokları inceleyelim.

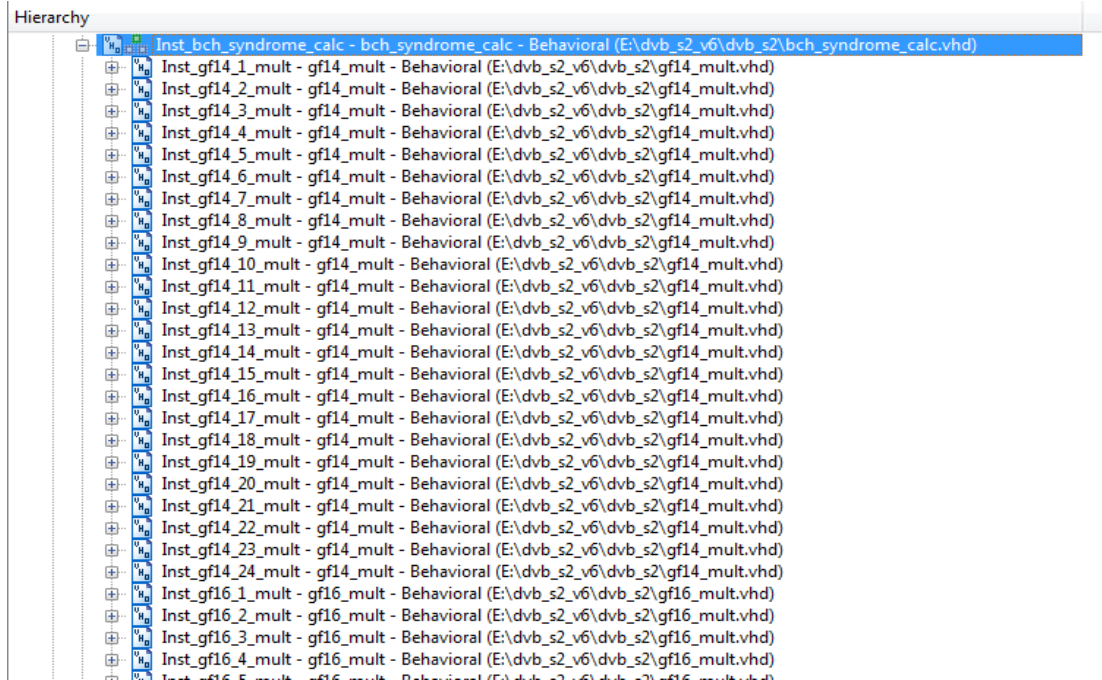
5.1.2 bch_receiver Modülü



Şekil 5.4 : bch_receiver modülü RTL şeması.

Şekil 5.4 'te verilen bch_receiver modülü gelen veriyi depolayarak sağlıklı bir veri akışının olmasını sağlar. Aynı zamanda verinin boyunu ve kodlanma şeklini de ortaya çıkararak ilgili modüllere bildirir. Veri tamamen fifoya doldurulunca gerekli işaretlerle ilgili modülü bilgilendirir.mnnmmmm

5.1.3 bch_syndrome_calc Modülü



Şekil 5.5 : bch_syndrome_calc modülü hierarşisi.

Şekil 5.5 'te verilen modül 24 adet kısa çerçeveli 24 adet de normal çerçeveli olmak üzere toplamda 48 adet GF çarpıcısı içerir. Bu çarpıcıların alt modüllerinde de kısa çerçeveli yapı için $14 \times 14 = 196$ adet, normal çerçeve için de $16 \times 16 = 256$ adet Şekil 3.2 'de gösterilen hücrelerden vardır. Bu hücreler de sendrom hesaplamak için kullanılmaktadırlar.

Hata belirteci hesaplayıcı giriş kod kelimesini polinom katsayısı serisiymiş gibi işler ve $2t$ katsayılarının hata belirteci polinomunu hesaplar. Hata belirteci polinomları geçersiz (invalid) bir kod kelimesi içindeki t hataları için konum ve genliği içerir. Geçerli (valid) bir kod kelimesi hepsinin sıfır katsayılı olduğu sendrom polinomunu üretir.

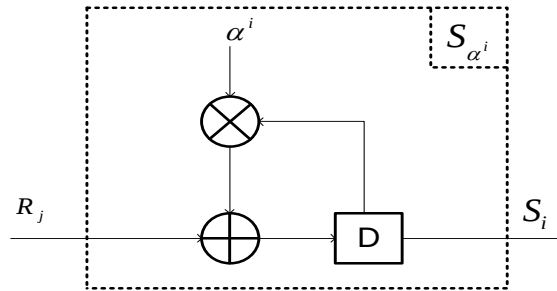
Gürültülü bir gönderimde alıcıdaki R kod kelimesi orijinal kod kelimesine E hatasının eklenmesiyle oluşur. Hata belirteci polinomunun tanımlanması ile $S(x)$

$$S(x) = \sum_{i=1}^{2t} S_i x^{(i-1)}, S_i = R(\alpha^i) \quad (5.1)$$

$$R(x) = R_0 + R_1 x + R_2 x^2 \dots \dots \dots + R_{N-1} x^{N-1} \quad (5.2)$$

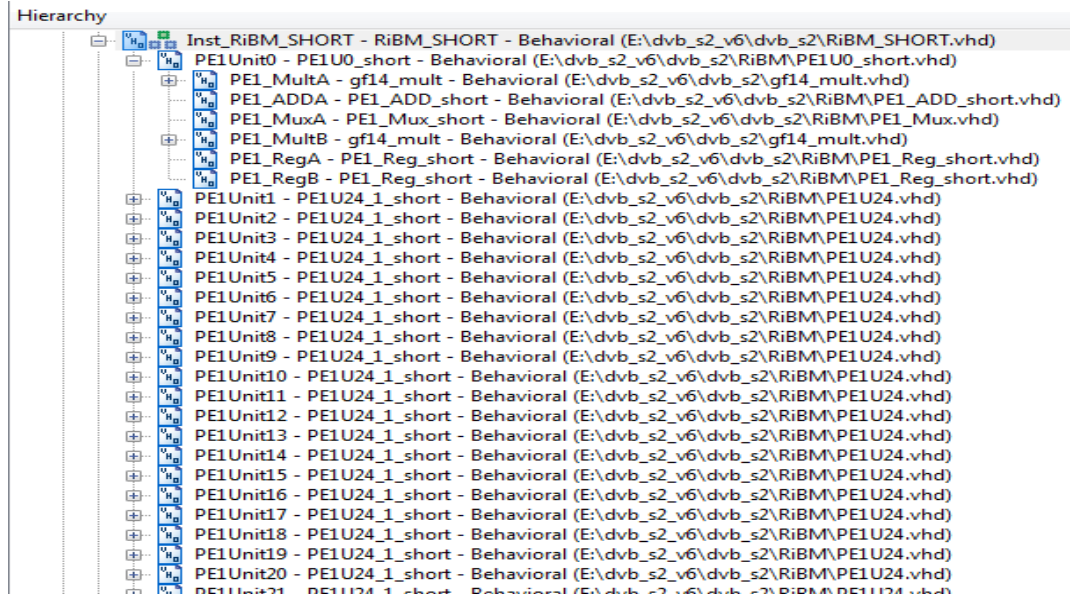
$$S_i = R_0 + \alpha^i \left(R_1 + \alpha^i \left(R_2 \dots \dots \dots + \alpha^i (R_{N-1}) \right) \right) \quad (5.3)$$

3. eşitlik literatürde Horney yöntemi olarak bilinmektedir. İşlem R_{n-1} katsayısının α^i ile çarpılması ile başlamaktadır. Daha sonra gelen her bir katsayı Şekil 5.6 'daki gibi bir önceki çarpım işlemi ile toplanmakta ve elde edilen sonuç α^i ile tekrardan çarpılır taki R_0 katsayı son olarak toplanana kadar. Bu yöntemin uygulanmasında en önemli faktör; her bir aşamada sürekli aynı α^i katsayısı ile çarpma işlemi yapmasıdır. Hata belirteci devresi paralel yapı tercih edilmiştir.



Şekil 5.6 : Hata belirteci hücresi.

5.1.4 ribm_short Modülü



Şekil 5.7 : ribm_short modülü hierarşisi.

Şekil 5.7 'de verilen BCH kod çözücünün temel parçası KES bloğudur. 2t lineer bağımlı eşitliğin kümesini çözer. Hata belirteç polinomundan anahtar eşiliğini ($\sigma(x)$ konumlayıcı polinomu ve $\Omega(x)$ değerlendirici polinomu) üretir. Konumlayıcı (Locator) polinomu kod kelimesinde ki hatalı simgelerin konumu hakkında bilgi içerir. Değerlendirici polinomu hatalı simgelerin hata genlikleri hakkında bilgi içerir. Bu $\sigma(x)$ ve $\Omega(x)$ polinomlarının matematiksel modelleri aşağıdaki gibidir.

$$\sigma(x) = \prod_{i=1}^w (1 - x * X_i) \quad (5.4)$$

w, R(x)'te oluşan hata sayısını, X, hata yerini ve Y, hata değerini belirtmektedir.

$$\Omega(x) = \sum_{i=1}^w Y_i * X_i * x \prod_{j=1, j \neq i}^w (1 - X_j * x) \quad (5.5)$$

İki polinom anahtar eşitlik'ten kaynaklı S(x) ile doğrudan ilişkilidir. Böylece anahtar eşitlik çözümünü kullanarak bilinmeyen $\sigma(x)$ ve $\Omega(x)$ polinomları aşağıdaki gibi belirlenebilir.

$$S(x) * \sigma(x) = \Omega(x) \bmod x^{2t} \quad (5.6)$$

KES bloğu farklı yöntemler kullanılarak yapılabilmektedir. Bunlar;

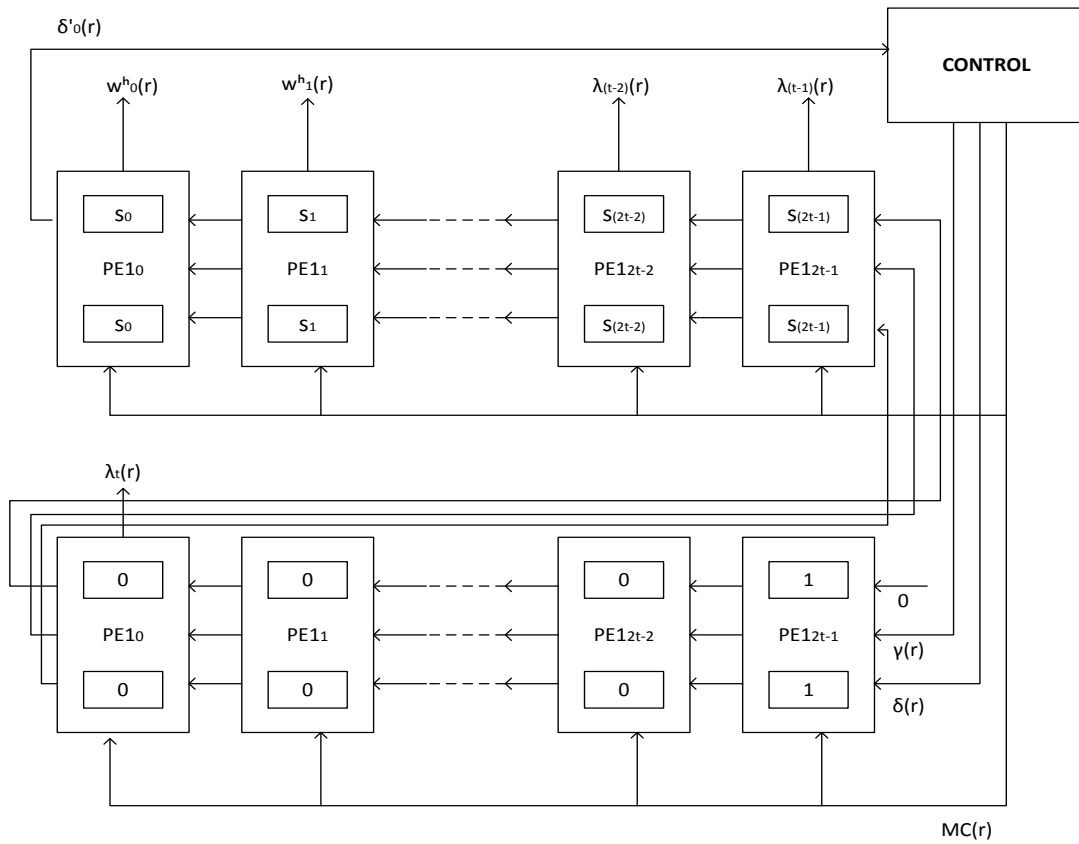
- a) Berlekamp-Messay algoritması
- b) Euclidean Algoritması

ve türevleridir.

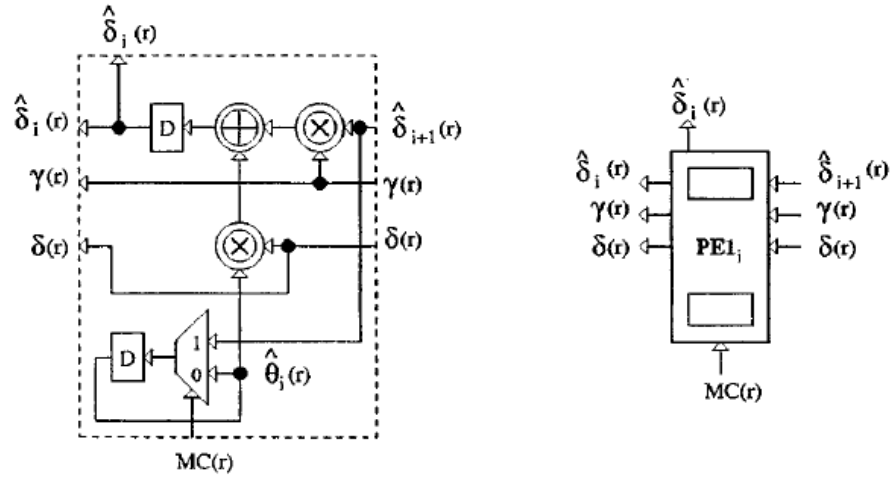
Çizelge 5.1 : KES bloğu donanımsal olarak karşılaştırılması.

Algorithm	Multipliers	Adders	D-FFs	Mux	Latency	Critical path delay
ME	64	48	628	322	60	$T_{ff}+T_M+T_A+T_{mux}$
DCME	48	24	48	48	15	$T_M+T_A+T_{mux}$
RiBM	50	25	50	25	16	T_M+T_A
pRiBM	50	24	100	25	32	$T_M+T_A+T_{ff}$
PF-RiBM	4	2	62	6	225	T_M+T_A
UiBM	2	1	50	1	384	T_M+T_A
FRiBM	12	6	78	18	65	T_M+T_A

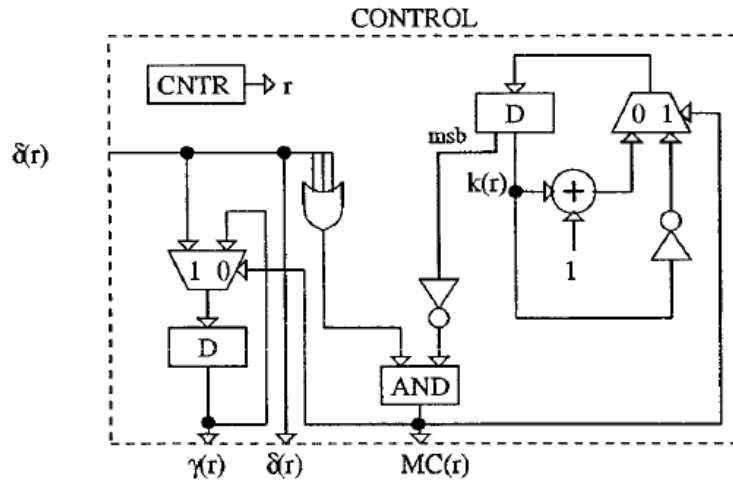
Çizelge 5.1 'de görüldüğü üzere gecikmenin(Latency) ve kritik yol beklmesinin(Critical path delay) en az olduğu RiBM (Reformulated inverse-free Berlekamp-Massey) algoritmasına karar verilmiştir.



Şekil 5.8 : RiBM devre şeması.



Şekil 5.9 : PE1 devresinin yapısı.

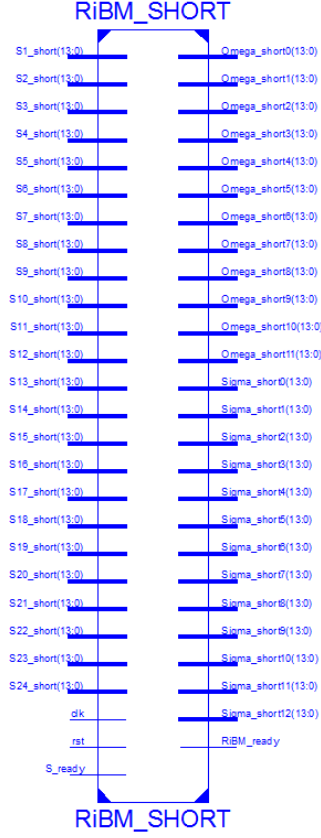


Şekil 5.10 : Kontrol devresinin yapısı.

Şekil 5.8 'de verilen KES bloğunda karmaşıklığın en aza indirgenmesi ve hızlı bir şekilde çalışabilmesi için 36 adet PE1 ve bir adet kontrol altbloğundan oluşmaktadır.

KES bloğunda en önemli devre yapısı PE1 yapısıdır. Şekil 5.9 'daki yapıda parametreler, Cf_Sigmaout, Cf_Sigmain, Sendrome, Gamma, Sigma temel parametreleri ve Load, Init, Hold ve MC kontrol parametreleri bulunmaktadır. Kontrol parametreleri Şekil 5.10 'da verilen kontrol devre yapısı tarafından üretilmektedir.

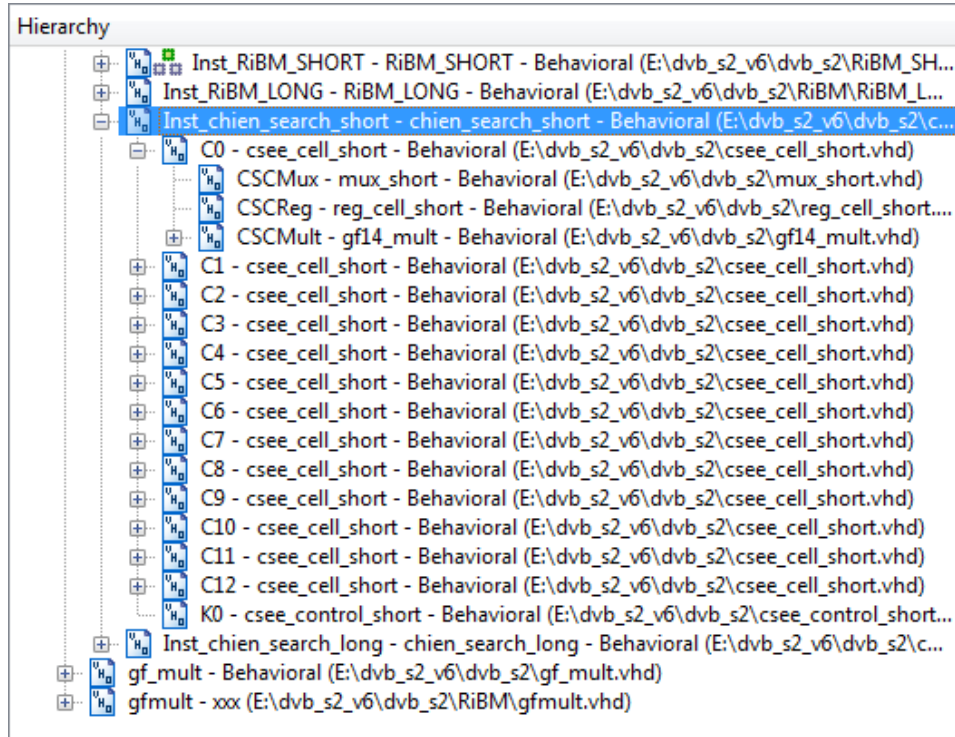
Kısaca kontrol parametrelerini açıklamak gerekirse, Init parametresi aktif olması ile beraber hata belirteci degerleri kayıtlara yüklenir. Load parametreleri Init parametresinin devre dışı olması ile beraber aktif olmakta ve CF_Sigmain değerini iterasyona dahil eder. Her bir bloğun görevini yerine getirmesi sonucunda Hold parametresi hesaplamının sonlanmasıyla aktif olup bloğun bir çerçevenin işlenmesini bitirir.



Şekil 5.11 : ribm_short modülü RTL şeması.

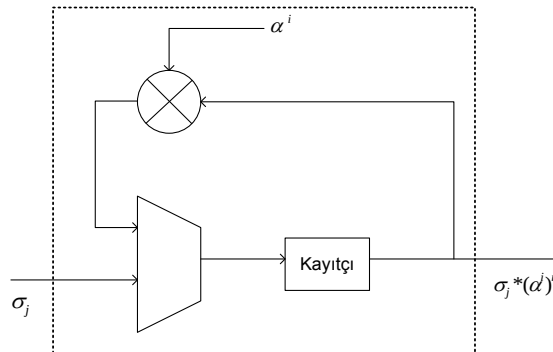
RIBM_LONG isimli modül de Şekil 5.11 'de gösterilen RIBM_SHORT isimli modülle aynı yapıdadır. Farklılık olarak sadece sendrome, omega ve sigma değerleri veri uzunlukları RIBM_SHORT'ta 14 bit iken RIBM_LONG'da 16 bittir ve gf14_mult yerine gf16_mult bloğu kullanılmıştır.

5.1.5 chien_search_short Modülü

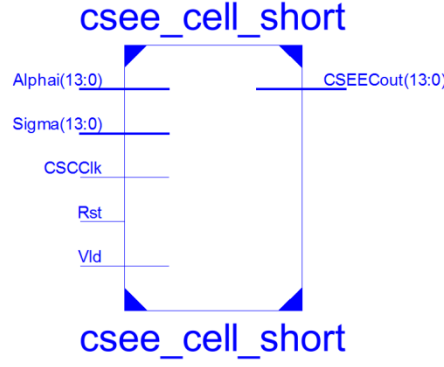


Şekil 5.12 : chien_search_short modülü hiyerarşisi.

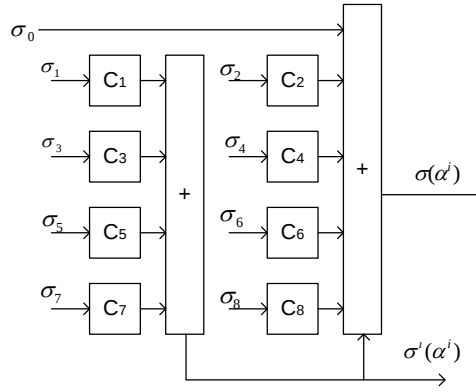
Hata konumlayıcı polinomun bilinmesi ile hata ile konumlayıcı polinomun sıfıra eşit olup olmadığının kontrolü yapılarak hatanın yerinin belirtilmesi mümkündür. Hata konumlayıcı polinomun kökleri kod kelimesinin hatalı yerlerinin tersidir. Polinomun köklerini bulmak için Şekil 5.12' deki yapıda ve Şekil 5.13' de verilen Chien araştırması yapılır. Tüm giriş değerlerini kullanır ve daha sonra çıkıştan sıfırı gördüğünü kontrol eder. Bu sadece bir hata oluştuğunda meydana gelir. Sıfıra eşit polinomlar içindeki yedeklenen elemanların ve hafızada saklanmış elemanların her biri için polinomun kökleri gibidirler ve bu yüzden hata konumunun tersidirler.



Şekil 5.13 : Chien araştırma hücresi.

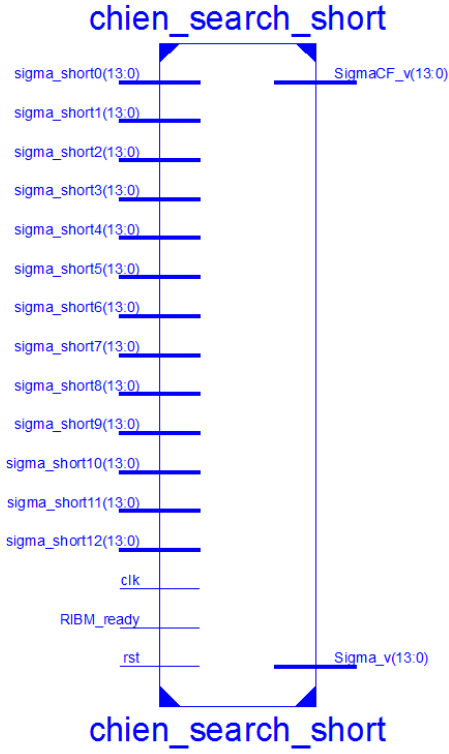


Şekil 5.14 : csee_cell_short modülü RTL şeması.



Şekil 5.15 : t=8 için Chien araştırma yapısı.

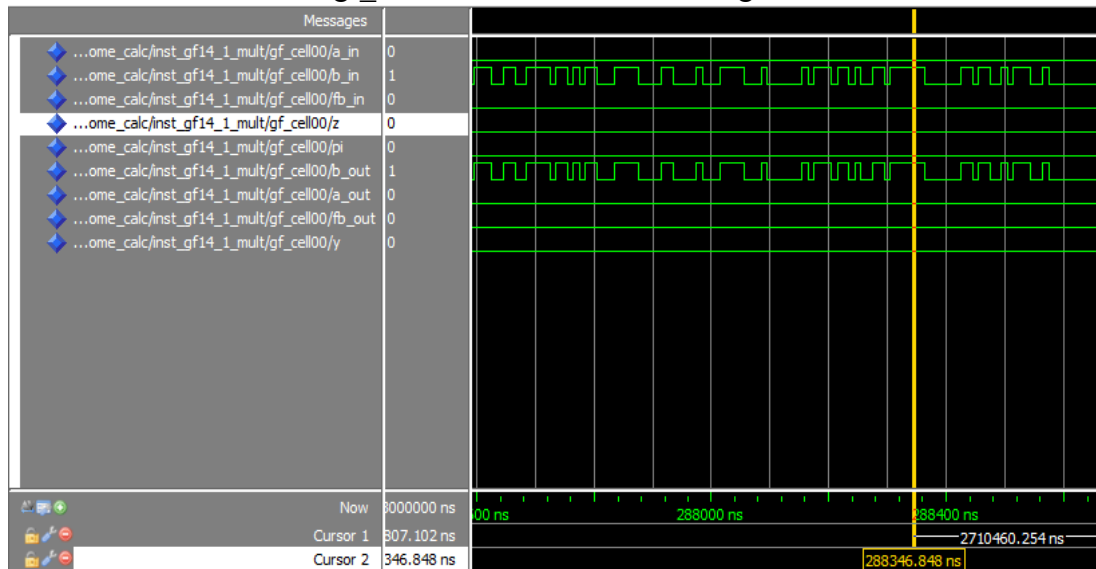
Donanımda Şekil 5.14 'te verilen yapıda Şekil 5.15 'te uygulanacak CA t+1 aşamaya sahiptir. Bu aşamalardan (bir aşama bir adet çarpıcı, çoğullayıcı (mux) ve kayıtçı içerir) her biri CA eşitliği üzerinde ki j değerleri için farklı bir değeri yansıtır. Arama n saat darbesi (her saat darbesi eşitlik üzerindeki i'nin farklı değerlerini gösterir) ile çalışır ve toplayıcı çıkışı sıfır eşitliği görmek için inceleme yapar. Şayet çıkış 0 ise sıfır algılayıcı devre çıkışı 1 yapar değilse 0 yapar. Her bir “1” değeri hatadaki simgenin konumunu gösterir. İlk saat darbesi için çoğullayıcı hata konumlayıcı polinom katsayılarını kayıtçıya yönlendirir. Geriye kalan n-1 saat darbesi için çarpıcının çıkışı kayıtçı içindeki çoğullayıcıya yönlendirilmiştir. RTL şeması Şekil 5.16 'da verilmiştir.



Şekil 5.16 : chien_search_short modülü RTL şeması.

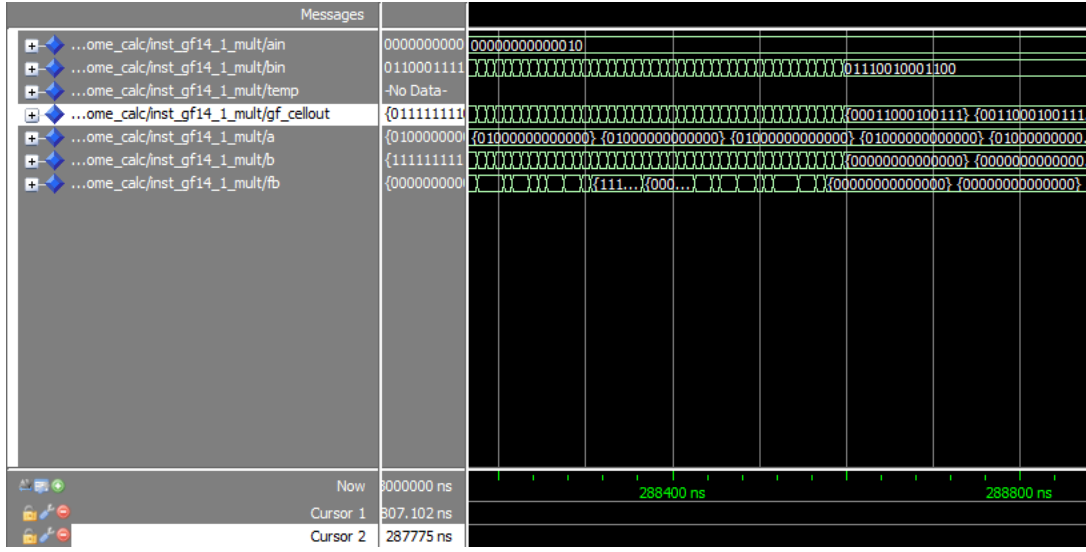
5.2 FPGA Benzetim Sonuçları

$GF(2^{14})$ çarpıcısı için 0'te verilen yapıya göre toplam 14x14 (mxm) olmak üzere 196 tane gf_cell modülü kullanılmaktadır. Ain değerleri $GF(2^{14})$ ve $GF(2^{16})$ çarpıcıları için tablolarda verilen α değerleridir. Aşağıda Şekil 5.17 'de 196 tane hücrenin ilk hücresi olan gf_cell00 hücresinin benzetimi görülmektedir.



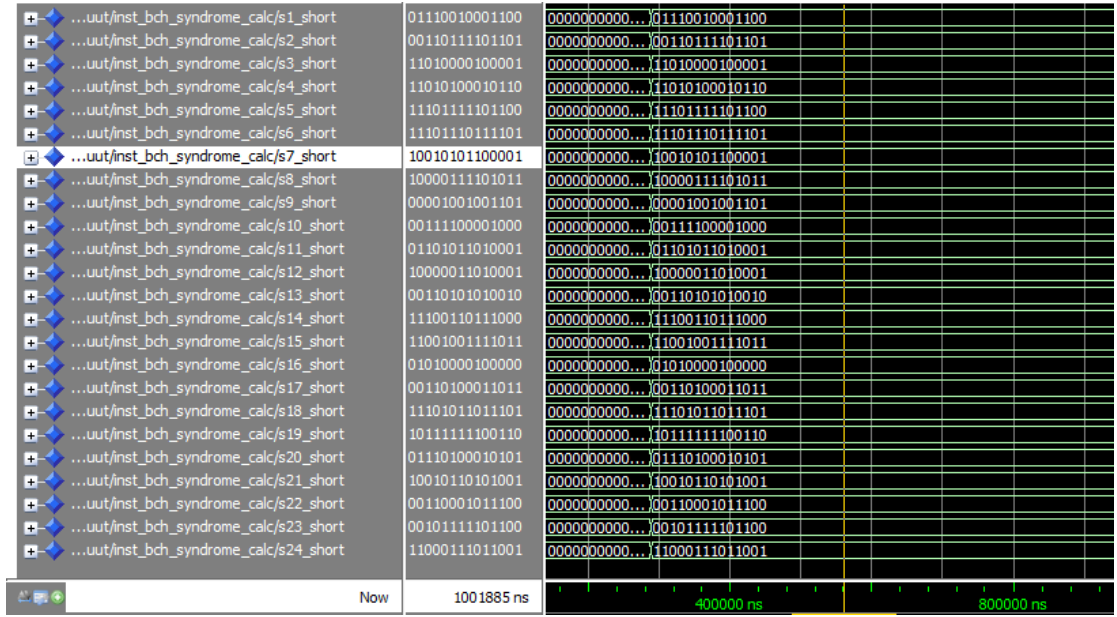
Şekil 5.17 : GF çarpıcı benzetimi.

Aynı şekilde $GF(2^{16})$ çarpıcısı için de Şekil 3.1 'de verilen yapıya göre toplam 16×16 (mxm) olmak üzere 256 tane gf_cell modülü kullanılmaktadır. Giriş ve çıkış portları da Şekil 3.2 'de verilen $GF(2^{14})$ çarpıcısıyla aynı şekildedir. Ain değerleri $GF(2^{14})$ ve GF (çarpıcıları için tablolarda verilen α değerleridir. Aşağıda Şekil 5.18 'de 24 tane $GF(2^{14})$ çarpıcısının ilki olan gf14_1_mult hücresinin benzetimi görülmektedir. Ain değeri $GF(2^{14})$ elemanları tablosunda verilmiştir. Aynı şekilde $GF(2^{16})$ çarpıcısı için benzetim yapacak olursak da ain ve bin değerlerinin 14 bit yerine 16 bit olacaktır.



Şekil 5.18 : $GF(2^{14})$ çarpıcı benzetimi.

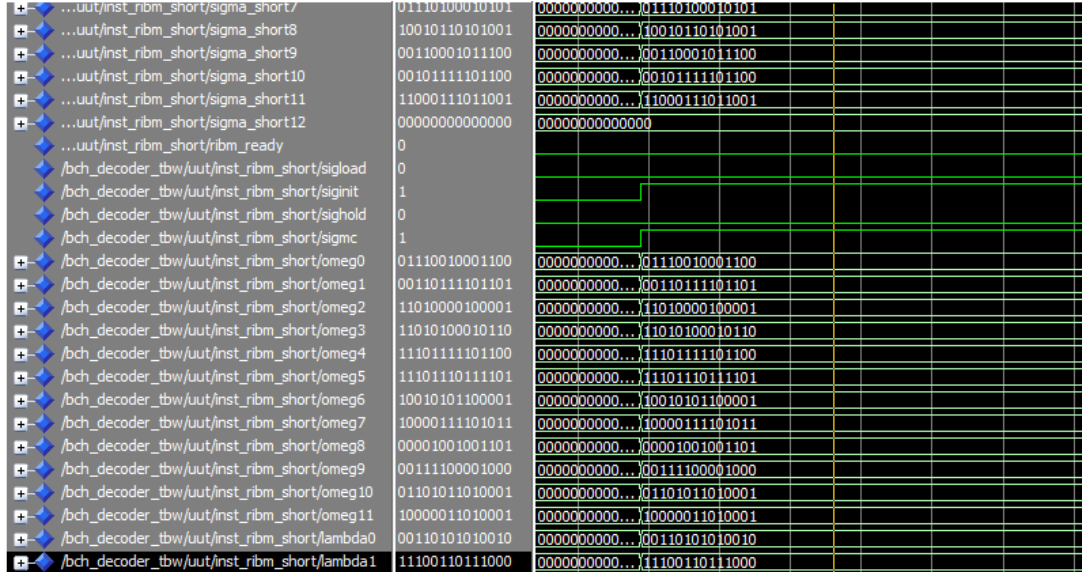
Syndrome hesaplayıcı modülde 12 bit hata düzeltici ($t=12$) BCH kod çözücü için 24 tane $GF(2^{14})$, 24 tane de $GF(2^{16})$ çarpıcısı kullanılmaktadır. Toplamda 24 tane syndrome değeri hesaplanmaktadır. DVB-S2 paket yapısı düşünüldüğünde normal ve kısa olmak üzere iki farklı data yapısı için syndrome hesaplama şekli $GF(2^{14})$ ve $GF(2^{16})$ çarpıcılarıyla farklılaştırılmıştır. Aşağıda Şekil 5.19 'da syndrome hesaplayıcısının benzetimi görülmektedir. Gelen veri benzetimde DVB-S2 paket yapılarından kısa paket yapısına denk gelmektedir ve $t=12$ seçilmiştir. Oluşan syndrome değerleri de 24 tane 14 bitlik değerden oluşmaktadır. Aynı şekilde paket yapısına göre değişmekle birlikte normal paket yapısı için de 16 bitlik syndrome değerleri oluşacaktır. Aşağıda 500 us koşturulan benzetim sonucunda oluşan syndrome değerleri görülmektedir. Oluşan syndrome değerleri MATLAB yardımıyla doğrulanmıştır.



Şekil 5.19 : Syndrome hesaplayıcı benzetimi.

Oluşan syndrome değerleri RIBM modülüne giriş olarak alınır. RIBM modülünden çıkan sonuçlar ise hata ayıklama ve düzeltme modülünde kullanılır.

Şekil 5.20 'de kısa paket yapısı için oluşturulan RIBM modülü görülmektedir. Benzer şekilde normal paket yapısı için oluşturulan RIBM modülünde tek fark data boyunun 14 bit yerine 16 bit olmasıdır. Aşağıda verilen şekilde RIBM modülünün benzetimi görülmektedir. Gelen veri benzetimde DVB-S2 paket yapılarından kısa paket yapısına denk gelmektedir ve t=12 seçilmiştir. Oluşan syndrome değerleri de 24 tane 14 bitlik değerden oluşmaktadır. Aynı şekilde paket yapısına göre değişmekle birlikte normal paket yapısı için de 16 bitlik syndrome değerleri oluşacaktır.



Şekil 5.20 : RIBM benzetimi.

Chien araştırması blok yapıları da RIBM deki blok yapılarıyla aynı özelliktedir bu nedenle ayrı bir benzetim şeması şeklinde gösterilmesine yer verilmeyecektir. BCH kod çözücüsü tasarımında da hızlı aynı zamanda kapladığı alan açısından da az ve esnek bir yapı oluşturulmuştur. Dışarıdan set edilebilen parametrelerle değişik veri uzunluklarına uyumlu bir tasarım yapılmıştır. Hata belirteci(syndrome) hesaplamada paralel GF çarpıcıları kullanılarak hızlı olması sağlanmıştır. Anahtar eşitlik çözücünde gecikmenin ve kritik yol beklemesinin(Critical path delay) en az olması için RiBM (Reformulated inverse-free Berlekamp-Massey) algoritması kullanılmıştır. Fifo büyüklükleri dışarıdan set edilebilecek şekilde esnek yapılmıştır.

6. LDPC KOD ÇÖZÜCÜ GERÇEKLEMESİ

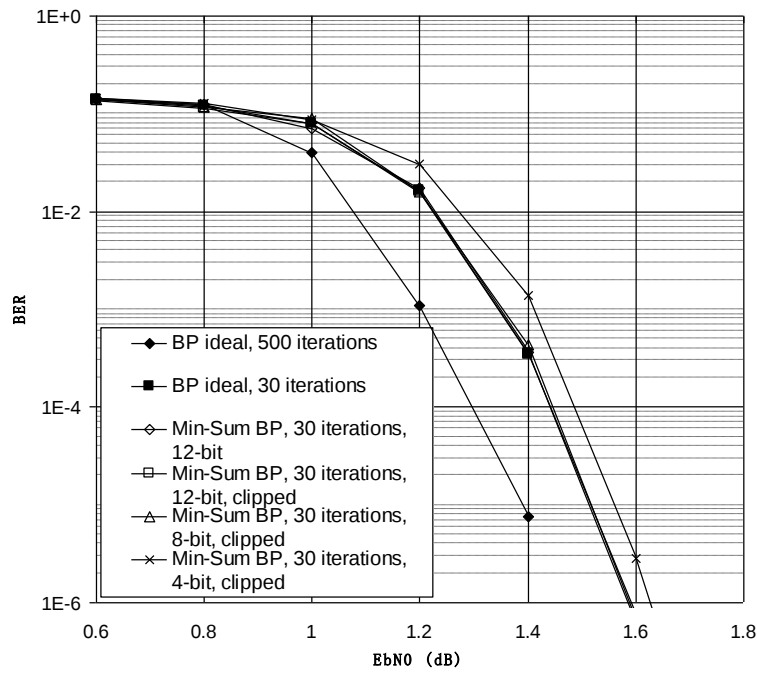
DVB-S2 için LDPC kod çözücüsü kabul edilebilir performans elde etmek için milyonlarca mesajın geçmesini gerektiren, hesaplama açısından yoğun bir işlemdir. Bu uygulamada yüksek verimlilik ve hata düzeltme performansı elde etmek için paralel blok yapıları kullanılmıştır. Standartlarda yer alan LDPC kod yapısının donanımsal olarak en az kompleksliğe sahip olacak şekilde tasarlanılmaya çalışılmıştır.

Tasarlanan bu kod çözücüde Belief-propagation(BP)[68] algoritması kullanılmıştır. LDPC kod çözücülerde oldukça sık kullanılan bu algoritma Tanner grafikteki bütün kenarlardan tekrarlamalı mesaj geçişi mantığına dayanır. Bu algoritmada parite bitler ve olasılık(likelihood) değerleri mesaj gibi VN ve CN ler arasında gidip gelir. İlk olasılık değerleri kanal kalitesinden ve alınan semboller ile takımyıldızı noktaları (constellation points) arasındaki Öklid uzaklığından hesaplanır. Mesajlar daha basit bir aritmetik sağlayan log-olasılık oranları(log-likelihood ratios LLR's) şeklinde geçirilir. Diğer bağlanan CN lerin işaretlerine göre parite denklemlerinin sıfıra set edilmesinden yola çıkılarak CN ler bağlanan VN lerdeki beklenen parite bitleri belirler. Beklenen bu parite bitleri bütün VN lere LLR değerleri ile birlikte giderler. VN ler bu değerleri VN parite bitlerini ve LLR değerlerini güncelleştirmek için kullanırlar ve döngü yeniden başlar. Bu şekilde parite denklemleriyle uyumlu olan doğru olan mesaj bitleri sağlamlaştırılırken diğer bozuk olan bitler de düzeltilir.

BP nin doğruluğu test edilmiştir ve donanım açısından uygun olan modeliyle de karşılaştırılmıştır[69]. Donanımla uyumlu modelinde ideal BP ye yakın özellik gösteren Min-Sum algoritması kullanılmıştır ve hemen hemen aynı performansı sağlamanın yanında çok daha az kompleksliğe sahiptir[65]. Bu algoritmada CN lerde bütün LLR değerlerini tutmak yerine en küçük LLR değerleri tutulmaktadır. Bu nedenle hem komplekslik azalmakta hem de RAM büyüklüğünde kaydadeğer bir azalma sağlamaktadır.

Aşağıdaki Şekil 6.1 'de ideal BP ve Min-Sum kullanılan BP arasındaki performans farklılıkları gösterilmektedir. Matlab benzetiminden yola çıkılarak oluşturulan bu

şekilden tasarımda kullandığımız iterasyon sayısını 30 değerine sabitlememizdeki neden de görülebilmektedir. İterasyon sayısı bakımından 30 ve 500 iterasyon karşılaştırıldığında performans açısından çok fazla bir kazanç sağlamamasına rağmen donanımsal devreye gerek hız gerekse kullanılan RAM açısından oldukça yük getirmektedir. Performans farklılıkları en iyi 4 bit kuantalamada görülmektedir. ((n,k)=(163200, 7200)). DVB-S2 sistemi mümkün olduğunca az gürültülü bir akışı sağlamak amacıyla tasarlanmıştır bu nedenle en fazla hata oranı 10^{-4} seviyelerindedir. Bu sebeplerden 30 iterasyon seviyesi yeterlidir ve önemli bir kazanım sağlamaktadır.



Şekil 6.1 : Min-Sum BP yaklaşımı ile ideal BP karşılaştırması.

Bu bölümde üçüncü bölümden itibaren başlayan ve teorik açıklaması yapılan DVB-S2 LDPC kod çözücü kısmı FPGA gerçekleştirilmesi anlatılacaktır.

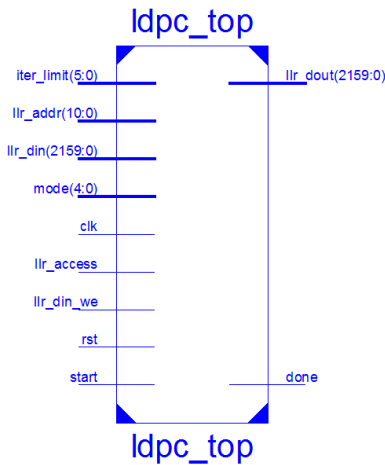
6.1 FPGA Modül Yapıları

6.1.1 ldpc_decoder Modülü



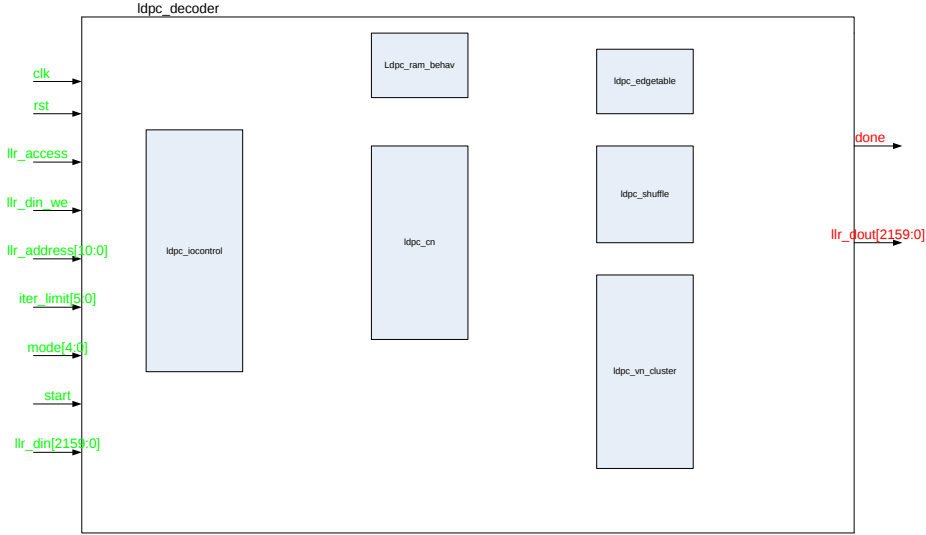
Şekil 6.2 : DVB-S2 LDPC FPGA tasarımı hiyerarşisi.

DVB-S2 LDPC kod çözücü FPGA tasarımı hiyerarşik yapısı Şekil 6.2 ' de gösterildiği gibidir.



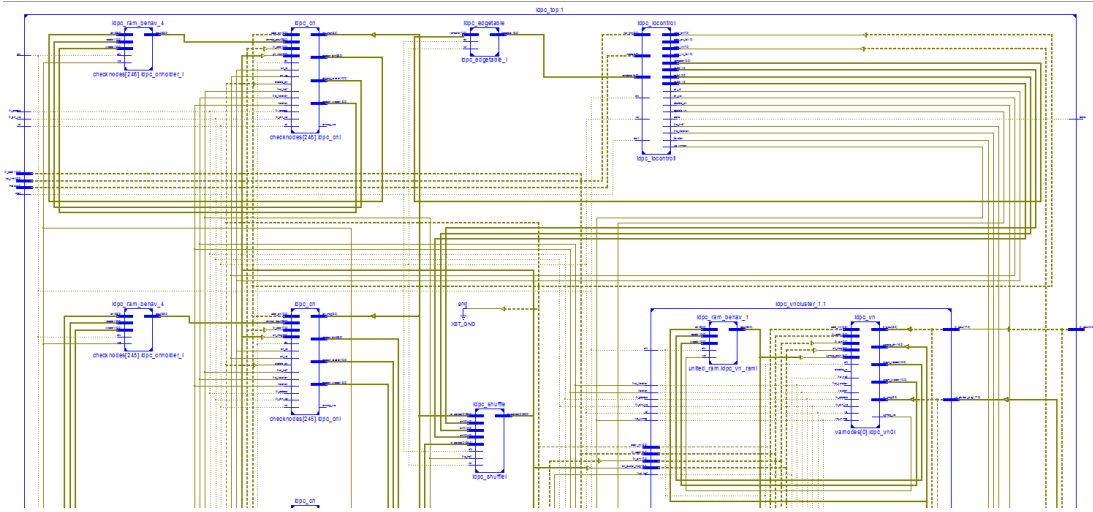
Şekil 6.3 : ldpc_decoder modülü RTL şeması.

Bu yapıyı Şekil 6.3 'teki gibi blok şemalar şeklinde gösterecek olursak `ldpc_decoder` modülü en üst modül olmak üzere :



Şekil 6.4 : DVB-S2 LDPC kod çözücü blok yapısı.

ldpc_decoder modülü Şekil 6.4 'te en üst modülü oluşturur. Bu modülün altında hierarşik yapıda ise şu modüller bulunmaktadır; ldpc_vn_cluster, ldpc_vn, ldpc_edgetable, ldpc_iocontrol, ldpc_shuffle, ldpc_cn .



Şekil 6.5 : ldpc_decoder modülü iç yapısı RTL şeması.

Şekil 6.5 'te verilen LDPC kod çözücü bloğuna bölüm 2.2 de bahsedilen bütün uzunluklardaki verilerin gelmesi muhtemeldir. DVB-S2 standardında 64800 bit veri uzunluğuna karşılık kullanılması gereken Tanner grafikteki kenar uzunluğu yaklaşık olarak 280000 dolaylarındadır. Projede hedeflenen performans 100Mbps olarak öngörülmektedir bu nedenle bu kadar yoğun miktardaki veriyi tamamen paralel işlemek yerine bu hız düzeyini yakalamamızı sağlayacak şekilde optimize etmek hem kullanılacak alan ve kaynak açısından hem de pratik olması açısından önem arz etmektedir. Bu nedenle daha esnek yapı sağlayan 2 dönmede veriyi gönderen tek

portlu RAM ler tercih edilmiştir. Bu varsayımdan yola çıkarak gerekli olan minimum paralellik derecesi şu şekilde hesaplanabilir:

$$\frac{30^{iterasyon} / kod\ s\ddot{o}zc\ddot{u}g\ddot{u} * 2^{233} kmesaj / iterasyon 100 Mbit/s}{200 M d\ddot{o}nme / s 0.5^{mesaj} / d\ddot{o}nme 64 kbit / kod\ mesajı} \approx 300 \quad (6.1)$$

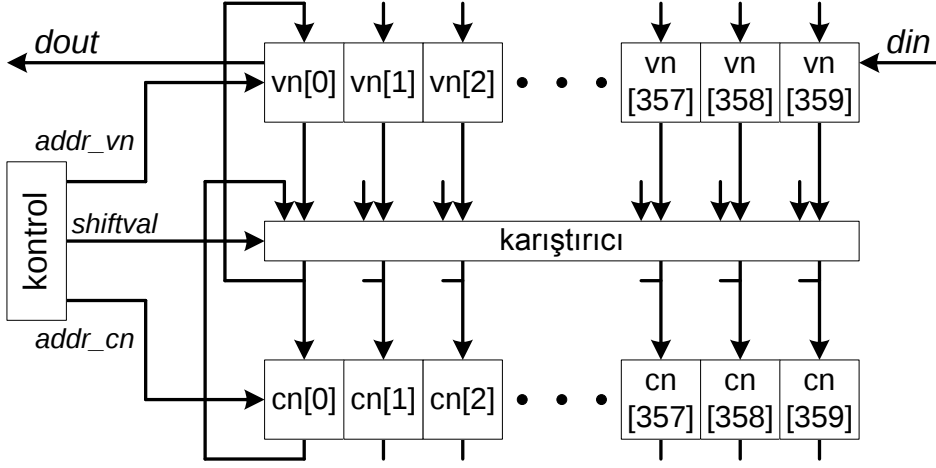
DVB-S2 standardı 360 bitlik gruplar halinde düzenlenmiştir bu nedenle bu çalışmada paralellik derecesi 360 olarak ele alınmıştır. Frekans arttırılmak istenirse bir parça karmaşıklığı artırmakla birlikte iki portlu bir RAM yapısı kullanılarak yeniden gerçekleştirilebilir.

Şekil 6.6 'da verilen tasarımda karıştırma modülü(ldpc_shuffle) olarak geçen ve 360 tane giriş mesajını 360 tane çıkış mesajına doğru kaydıran bir modül kullanılmıştır. DVB-S2 standardında giriş çıkış bağlantıları rastgele olmadığından dolayı karıştırma modülünde her girişinde bir çoklayıcının yer aldığı rotasyonel bir yapı oluşturulmuştur.

Herbir VN iki tane RAM e bağlıdır bunlardan bir tanesi LLR değerini, diğeri ise CN den gelen mesajların toplamını tutmaktadır ve herbir VN gelen mesajları birleştirmek ve çıkış mesajlarını oluşturmak için toplayıcılar içermektedir ve registerlarda bu giriş-çıkış mesajlarını tutmaktadır.

Herbir CN ise bir tane büyük RAM e bağlıdır ve bu RAM ilgili iterasyondaki en küçük iki LLR değerini yerleri ile birlikte ve herbir giriş mesajının işareti ile giriş mesajlarının parite bitlerini tutmaktadır. VN e gidecek mesajlar da herhangi bir LLR değerinin okunması ile oluşturulur.

Kontrol modülü ise karıştırma modülünde kullanılmak üzere kayma değerlerini ve CN lar için de okuma-yazma adreslerini ROM dan okur. VN den CN e mesaj taşımak için bu değerler doğrudan kullanılır. CN de VN e mesaj taşımak için ise her iki yönde veri akışını sağlamak amacıyla kaydırma değerleri etkisiz hale getirilir. Aynı zamanda kontrol modülünde tüm bu işlemler sırasında oluşabilecek gecikmeler telefı edilmektedir.



Şekil 6.6 : ldpc_decoder blok şeması.

Mesaj grubundaki 360 tane değişken noktasına(VN) paralel erişime izin vermek amacıyla Şekil 6.7 'deki VN RAM yapısı organize edilmiştir.

	<i>vn</i> [0]	<i>vn</i> [1]	<i>vn</i> [2]	• • •	<i>vn</i> [357]	<i>vn</i> [358]	<i>vn</i> [359]
179	n-360	n-359	n-358		n-3	n-2	n-1
•							
•							
•							
3	1080	1081	1082		1437	1438	1439
2	720	721	722		1077	1078	1079
1	360	361	362		717	718	719
<i>addr</i> =0	0	1	2		357	358	359

Şekil 6.7 : VN RAM organizasyonu.

DVB-S2 standardında [2] ldpc için rastsal düzensiz bir kenar tablosu yaratmak amacıyla q dediğimiz şifre faktörü yayınlanmıştır. Bu formül standartta $q = \frac{n-k}{360}$ şeklinde karakterize edilmiştir. Önceden belirtildiği gibi ldpc yapısında 20 farklı çeşit modülasyon yapısı vardır ve n , k değerleri değişkendir. Bu nedenledir ki kenar değerleri her bir çeşit için farklılık göstermektedir. Buna göre standartta belirtildiği gibi her bir $vn[m]$ (kenar başlama referans noktası) olmak üzere;

$$cn[y] = x + (m * q) \bmod (n - k) \quad (6.2)$$

denklemini ile ifade edilen Şekil 6.8 'deki CN RAM organizasyonu ifade edilmektedir.

	$cn[0]$	$cn[1]$	$cn[2]$	$\cdot \cdot \cdot$	cn $[357]$	cn $[358]$	cn $[359]$
$q-1$	q-1	2q-1	3q-1		358q-1	359q-1	360q-1
$\cdot$							
$\cdot$							
$\cdot$							
3	3	q+3	2q+3		357q+3	358q+3	359q+3
2	2	q+2	2q+2		357q+2	358q+2	359q+2
1	1	q+1	2q+1		357q+1	358q+1	359q+1
addr=0	0	q	2q		357q	358q	359q

Şekil 6.8 : CN RAM organizasyonu.

Bu yapı ldpc kodlama aşamasında aynı şekilde kullanılır. Bu CN RAM organizasyonuna uyum sağlaması açısından VN RAM organizasyonu yeniden Şekil 6.9 'da verildiği gibi yapılabilir:

	$vn[0]$	$vn[1]$	$vn[2]$	$\cdot \cdot \cdot$	vn $[357]$	vn $[358]$	vn $[359]$
179	k+q-1	k+2q-1	k+3q-1		k+358q-1	k+359q-1	n-1
$\cdot$							
$\cdot$							
$\cdot$							
$k/360+1$	k+1	k+q+1	k+2q+1		k+357q+1	k+358q+1	k+359q+1
$k/360$	k	k+q	k+2q		k+357q	k+358q	k+359q
$k/360-1$	k-360	k-359	k-358		k-3	k-2	k-1
$\cdot$							
$\cdot$							
$\cdot$							
2	720	721	722		1077	1078	1079
1	360	361	362		717	718	719
addr=0	0	1	2		357	358	359

Şekil 6.9 : Karar verilen VN RAM organizasyonu.

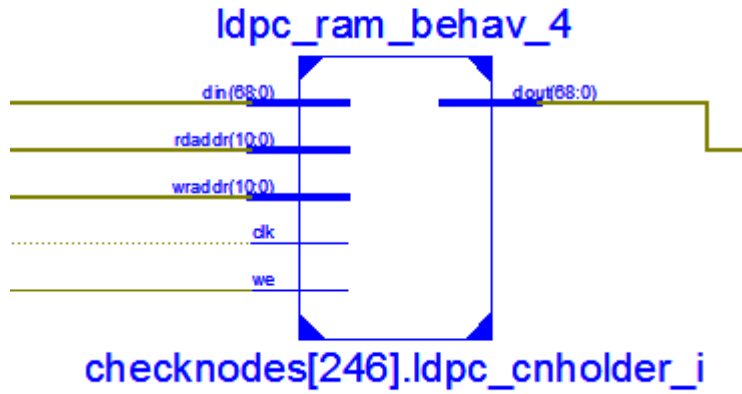
VN RAM de 30 kadar gelen mesajı saklamak için oldukça büyük bir RAM gerekmektedir. Büyük bir RAM den kaçınmak için CN de bir önceki mesaj ile yeni gelen mesaj saklanır ve birbirinden çıkarılır. Bu yaklaşım bütün mesajları saklayarak elde edilen LLR değeri ile aynı sonucu verir. Karşımıza çıkabilecek bir problem olarak gözüken herhangi bir mesajın doygunluğa ulaşması ve bozulmasıdır. Bu durumu engellemek için ise giden mesajı gelen mesajdan bir bit daha az kabul

etmekle çözülmüştür. Bu bize daha VN de daha az RAM kullanmamızı sağladığı gibi getirdiği dezavantaj ise etkili mesaj biti sayısını bir bit azaltarak performans düşüklüğüne yol açmasıdır.

Bu projede katmanlı kod çözme kullanılmıştır ve bu da VN den çıkan ve CN e giden herbir mesajı yenileyerek iterasyon sayısında azalmaya neden olur. Bu yaklaşım da herbir iterasyonda mesaj geçme yönlerinin sürekli kaydırılmasına ve VN RAM indeki değerlerin sürekli son gelen mesajlarla yenilenmesine neden olur. Muhtemel bir parite matris satır düzeni yerine sütun içinden geçen mesaj düzenine dönmek RAM organizasyonunda birtakım düzenlemeler gerektirecektir.

Şimdi kısaca anlattığımız bu blokları inceleyelim.

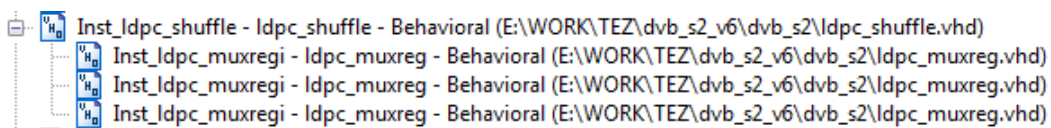
6.1.2 ldpc_ram_behav Modülü



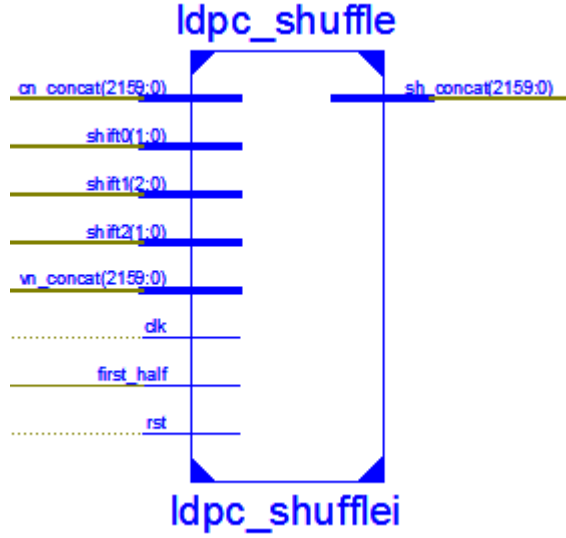
Şekil 6.10 : ldpc_ram_behav modülü RTL şeması.

Tek port girişli ldpc_ram_behav modülü projede birçok modülün alt birimi olarak kullanılmıştır. RTL şeması Şekil 6.10 'da verilen modül CN RAM olarak kullanılmaktadır, veri yazma işlemi belirtilen adrese "we" sinyali veri boyunca kaldırılarak yapılır. Okuma işlemi de aynı şekilde gerçekleşmektedir. Oldukça esnek olması açısından veri boyu "WIDTH" adı verilen ve bütün modüllerde ortak olarak kullanılan data port genişliği ile değiştirilebilmektedir.

6.1.3 ldpc_shuffle Modülü



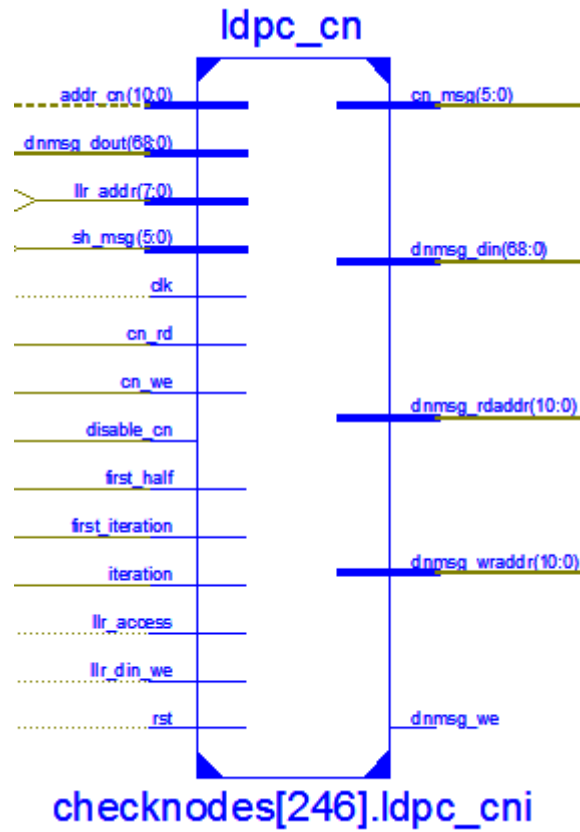
Şekil 6.11 : ldpc_shuffle modülü hiyerarşisi.



Şekil 6.12 : `ldpc_shuffle` modülü RTL şeması.

Şekil 6.11 ve Şekil 6.12 'de verilen `ldpc_shuffle` modülünde de daha önceki modüllerde olduğu gibi esnek bir yapıda olmasına dikkat edilmiştir ve gerek data boyunu gerekse kullanılmak istenen karıştırıcı sayısı değiştirilerek direk olarak modül yapısı değiştirilebilmektedir. Karıştırıcı CN ve VN mesajlarını kaydırıcılara yükleyerek muxlar. Bu yapıda multiplexerları "first_half" kontrol eder, işaret kalktığı zaman "vn_concat" mesajları ilk olarak multiplexerlara girer daha sonraki saat darbesinde de "cn_concat" mesajları kullanılır.

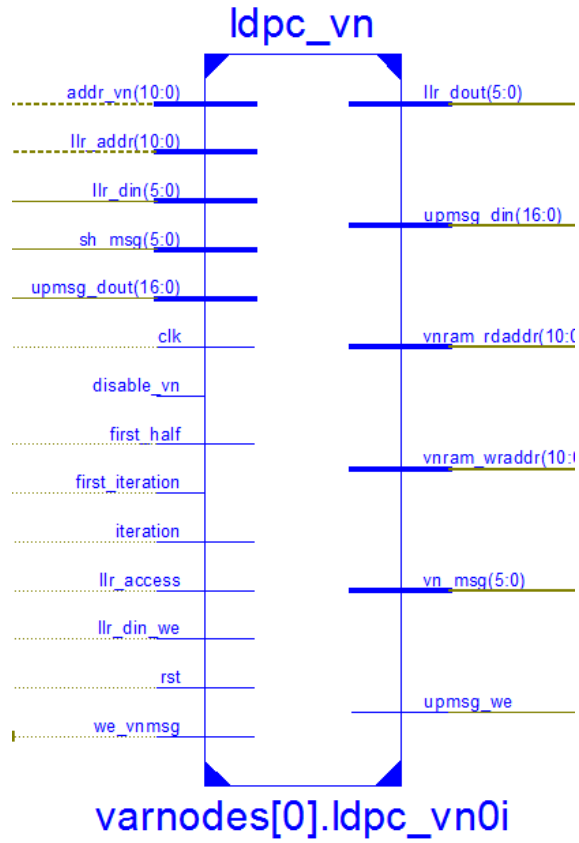
6.1.4 ldpc_cn Modülü



Şekil 6.13 : ldpc_cn modülü RTL şeması.

Sistem kontrol merkezi olan "ldpc_iocontrol" modülü CN ve VN arasındaki mesaj alışverişini okuma ve yazma adreslerini kullanarak yönetir. Bu kontrol merkezi kenar hedeflerini de ROM dan okur ve "dnmsg_we" işaretiyle de CN mesajlarının Şekil 6.13 'te verilen CN RAM e yazılmasını sağlar.

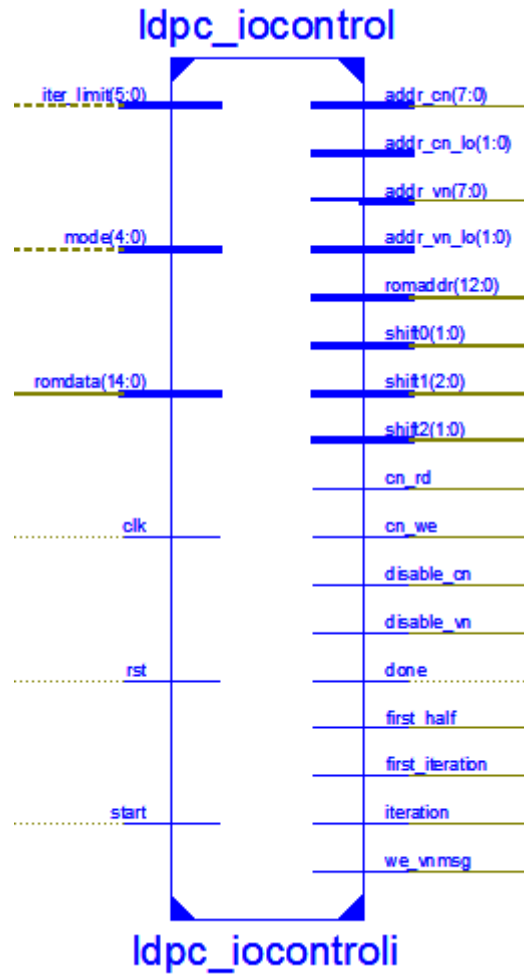
6.1.5 ldpc_vn Modülü



Şekil 6.14 : ldpc_vn modülü RTL şeması.

Sistem kontrol merkezi olan "ldpc_iocontrol" modülü CN ve VN arasındaki mesaj alışverişini okuma ve yazma adreslerini kullanarak yönetir. Bu kontrol merkezi kenar hedeflerini de ROM dan okur ve "upmsg_we" işaretiyle de VN mesajlarının Şekil 6.14 'te verilen VN RAM e yazılmasını sağlar.

6.1.6 ldpc_iocontrol Modülü



Şekil 6.15 : ldpc_iocontrol modülü RTL şeması.

Şekil 6.15 'te verilen kontrol modülü karıştırma modülünde kullanılmak üzere kayma değerlerini ve CN lar için de okuma-yazma adreslerini ROM dan okur. VN den CN e mesaj taşımak için bu değerler doğrudan kullanılır. CN de VN e mesaj taşımak için ise her iki yönde veri akışını sağlamak amacıyla kaydırma değerleri etkisiz hale getirilir. Aynı zamanda kontrol modülünde tüm bu işlemler sırasında oluşabilecek gecikmeler telafi edilmektedir.

6.2 FPGA Benzetim Sonuçları

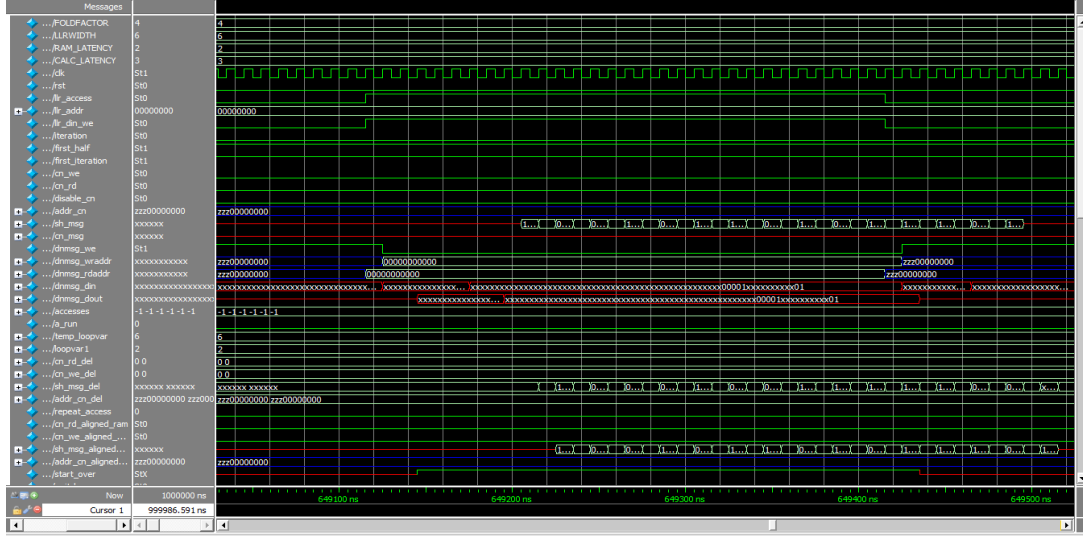
Xilinx ISE de yazılan fpga kodları ayrı ayrı testbençler yazılarak Modelsim kullanılarak simülasyonları yapılmıştır.

Sistemde 20 farklı moda karşılık gelen DVB-S2 bütün data formatları desteklenmektedir. Kod çözücünün çalışması için ldpc_decoder modülüne dışarıdan verilmesi gereken giriş sinyalleri öncelikle ortak bir saat darbesiyle birlikte rst, llr_din_we, mode, llr_din, llr_access, llr_addr, iter_limit ve start sinyalleridir. İlk olarak sisteme llr_din datası seri bir şekilde yüklenmelidir. Yükleme sırasında llr_access ve llr_din_we sinyalleri kalkık durumdadır. Mesaj 0 gösterildiği gibi yüklenmelidir. Mesajın hepsi kod çözücüye yazıldıktan sonra artık kod çözme işlemi "start" sinyali kaldırılarak yapılabilir. Aynı anda mode ve iterasyon bilgisinin de girilmesi gerekmektedir. Uygulanabilecek mode değerleri Çizelge 6.1 'de gösterilebilir:

Çizelge 6.1 : Mode değerleri.

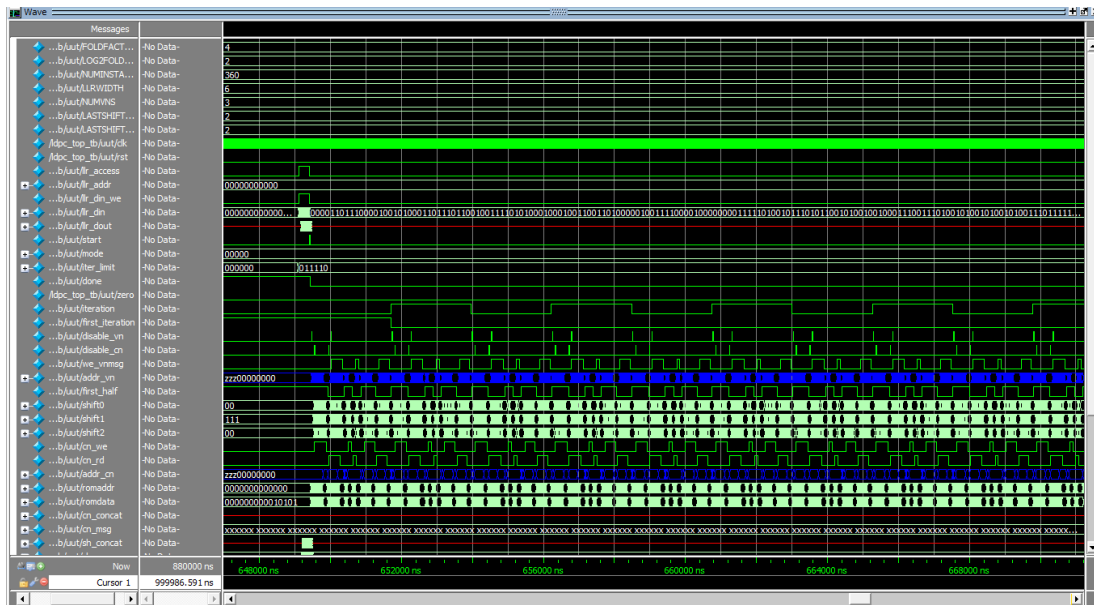
mode numarası	açıklama	n	k
0	1/4 normal	64800	16200
1	1/3 normal	64800	21600
2	2/5 normal	64800	25920
3	1/2 normal	64800	32400
4	3/5 normal	64800	38880
5	2/3 normal	64800	43200
6	3/4 normal	64800	48600
7	4/5 normal	64800	51840
8	5/6 normal	64800	54000
9	8/9 normal	64800	57600
10	9/10 normal	64800	58320
11	1/5 short	16200	3240
12	1/3 short	16200	5400
13	2/5 short	16200	6480
14	4/9 short	16200	7200
15	3/5 short	16200	9720
16	2/3 short	16200	10800
17	11/15 short	16200	11880
18	7/9 short	16200	12600
19	37/45 short	16200	13320
20	8/9 short	16200	14400

Karıştırıcı CN ve VN mesajlarını kaydırıcılara yükleyerek muxlar. Bu yapıda multiplexerları "first_half" kontrol eder, işaret kalktığı zaman "vn_concat" mesajları ilk olarak multiplexerlara girer daha sonraki saat darbesinde de "cn_concat" mesajları kullanılır.



Şekil 6.19 : ldpc_cn modülü benzetimi.

Sistem kontrol merkezi olan "ldpc_iocontrol" modülü CN ve VN arasındaki mesaj alışverişini okuma ve yazma adreslerini kullanarak yönetir. Bu kontrol merkezi kenar hedeflerini de ROM dan okur ve "dnmsg_we" işaretiyle CN mesajlarının Şekil 6.19'daki gibi CN RAM e yazılmasını sağlar ve "upmsg_we" işaretiyle de VN mesajlarının Şekil 6.20'deki gibi VN RAM e yazılmasını sağlar.



Şekil 6.20 : ldpc_vn modülü benzetimi.

7. SONUÇ VE ÖNERİLER

Bu çalışmada literatürden farklı olarak 21 adet farklı uzunlukta veri için, DVB-S2 standardı hata düzeltme kodları olan BCH ve LDPC kodları analizi ve donanım gerçeklemesi yapılmıştır. DVB-S2 gibi yüksek hız ve büyük uzunluktaki veriler için yüksek başarımlı sağlayan ve hızlı, esnek bir yapı üzerinde durularak oldukça kullanışlı ve karmaşıklıktan uzak bir tasarım gerçekleştirilmiştir. Literatürde geçen bir çok farklı yaklaşım incelenmiş ve en uygun modellerde karar kılınarak mümkün olduğunca hızlı olması için gerekli paralelleştirmeler yapılmıştır.

BCH kod çözücüsü tasarımında da hızlı aynı zamanda kapladığı alan açısından da az ve esnek bir yapı oluşturulmuştur. Dışarıdan set edilebilen parametrelerle değişik veri uzunluklarına uyumlu bir tasarım yapılmıştır. Hata belirteci(syndrome) hesaplamada paralel GF çarpıcıları kullanılarak hızlı olması sağlanmıştır. Anahtar eşitlik çözücünde gecikmenin ve kritik yol beklemesinin(Critical path delay) en az olması için RiBM (Reformulated inverse-free Berlekamp-Massey) algoritması kullanılmıştır. Fifo büyüklükleri dışarıdan set edilebilecek şekilde esnek yapılmıştır.

LDPC kod çözücü yapısında hızlı haberleşme sistemleri için ideal olan belief propagation yöntemi seçilmiş literatürden farklı olarak karmaşıklığı ve kapladığı alanı azaltmak amacıyla min-sum algoritması kullanılmıştır. Tekrar hız açısından verimli olması amacıyla düzenli H matrisi güvenilir bir kaynaktan kenar tablosu alınarak oluşturulmuştur. İterasyon sayısı, mod kodu, veri uzunlukları dışarıdan girilebilecek şekilde yapılarak hertürlü(yer ya da hız) kısıt için optimum ayarlanabilecek bir tasarım oluşturulmuştur. Gerek hız gerek yer açısından bit ve kontrol noktaları RAM leri 360 bitlik gruplara ayrılarak kısmi paralel bir yapı oluşturulmuştur. Hesaplanan LLR değerlerinden sadece 2 si tutularak literatürdekilere göre yerden kazanç sağlanmıştır. Yumuşak karar LDPC kodu kullanılarak yerden tasarruf sağlanmış kodlama performansı artırılmıştır.

Sentez sonucunda BCH kod çözücü algoritması kullanılan Virtex-6 (XC6VLX240T – 1FFG1156) FPGA de %13 kadar yer kaplamaktadır ve minimum periodu 4.345 ns olmak üzere maksimum frekansı 230 MHz'dir. LDPC kod çözücüsünün kapladığı

alan ise %45 ve minimum periodu 4.49 ns olmak üzere maksimum frekansı 222 MHz seviyedindedir. Elde edilen sonuçlar düşünüldüğünde gerekleri (100MHz frekansında) hem kapladığı alan hem de hız açısından gerekleri fazlasıyla yerine getiren bir tasarım yapılmış olduğu görülmektedir. Herbir modül işlevsel bir şekilde tasarlanmıştır ve başka amaçlarda da kullanılabilecek çok esnek bir yapı oluşturulmuştur.

Bu çalışmada tasarlanan kod çözücü yapısı bütün DVB-S2 standardı paket ve kod yapısını desteklemektedir. İleriki çalışma olarak bu tasarım blokları donanımsal olarak sentezlenerek Virtex-6 FPGA e yüklenerek gerçek çalışma ortamında kullanılacaktır.

Tasarım her yönüyle incelenmiş, benzetimi ve sentezi yapılmıştır. Hem LDPC hem de BCH blokları tasarımının yapılmış olması DVB-S2 ileri hata düzeltme birimi tasarımının tamamıyla tamamlanması sonucunu doğurmuştur.

KAYNAKLAR

- [1] **McEliece, R.J.** (1977). *The Theory of Information and Coding*, Addison-Wesley Publishing Company, New York.
- [2] **ETSI EN 300 421 v 1.1.2.** (2004). *Digital Video Broadcasting (DVB); Framingstructure, channel coding and modulation for 11/12 GHz satellite services.*
- [3] **TSI EN 302 307 Draft v1. 1.1.** (2004-06). *Digital Video Broadcasting(DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications.*
- [4] **Hanna, S. A.** (1993, 12-15 Oct 1993). *Convolutional interleaving for digital radio communications.* Paper presented at the Universal Personal Communications, 1993. Personal Communications: Gateway to the 21st Century. Conference Record., 2nd International Conference on.
- [5] **Vaupel, M., Lambrette, U., Dawid, H., Joeressen, O., Bitterlich, S., Meyr, H.** (1997). Lehrstuhl Fur Integrierte Systeme, F. Frieling, K. Müller. *An All-Digital Single-Chip Symbol Synchronizer and Channel Decoder for DVB.*
- [6] **Chaari, L., Fourati, M., Masmoudi, N., Kamoun, L.** (2003). A reconfigurable FEC system based on Reed-Solomon codec for DVB and 802.16 network, *WSEAS Transactions on Circuits and Systems*, v.8 n.8, 729-744.
- [7] **Lattice Semiconductor.** (2005). Reed-Solomon Decoder IP User's Guide .
- [8] **Altera.** (2011). Reed-Solomon Compiler User Guide .
- [9] **Reed, I. S., & Shih, M. T.** (1991). VLSI design of inverse-free Berlekamp-Massey algorithm. *Computers and Digital Techniques, IEE Proceedings E*, 138(5), 295-298.
- [10] **Zhiqiang, Y., & Yun, L.** (2008, 12-14 Oct. 2008). *An Efficient Convolutional Code Information Recovery Scheme in DVB-S.* Paper presented at the Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on.
- [11] **Hagenauer, J., & Hoeher, P.** (1989, 27-30 Nov 1989). *A Viterbi algorithm with soft-decision outputs and its applications.* Paper presented at the Global Telecommunications Conference and Exhibition 'Communications Technology for the 1990s and Beyond' (GLOBECOM), 1989. IEEE.
- [12] **Forney, G. Jr.** (1971). Burst-Correcting Codes for the Classic Bursty Channel. *Communication Technology, IEEE Transactions on*, 19(5), 772-781. doi: 10.1109/TCOM.1971.1090719

- [13] **Ramsey, J.** (1970). Realization of optimum interleavers. *Information Theory, IEEE Transactions on*, 16(3), 338-345. doi: 10.1109/TIT.1970.1054443
- [14] **Yi-Min, Lin, Jau-Yet, Wu, Chien-Ching, Lin, & Chang, Hsie-Chia.** (2009, 14-16 Dec. 2009). *A long block length bch decoder for DVB-S2 application*. Paper presented at the Integrated Circuits, ISIC '09. Proceedings of the 2009 12th International Symposium on.
- [15] **Eroz M., Sun F. W. ve Lee L. N.** (2004). DVB-S2 Low Density Parity Check Codes with Near Shannon Limit Performance, IJSC.
- [16] **Ying, Y., Dan, B., Shuangqu, H., Xiang, B., Chen, Y., & Xiaoyang, Z.** (2009, 20-23 Oct. 2009). *A cost efficient LDPC decoder for DVB-S2*. Paper presented at the ASIC, 2009. ASICON '09. IEEE 8th International Conference on.
- [17] **Seok-Min, Kim, Chang-Soo, Park, & Sun-Young, Hwang.** (2010). A novel partially parallel architecture for high-throughput LDPC Decoder for DVB-S2. *Consumer Electronics, IEEE Transactions on*, 56(2), 820-825. doi: 10.1109/TCE.2010.5506007
- [18] **Muller, S., Schreger, M., Kabutz, M., Alles, M., Kienle, F., & Wehn, N.** (2009, 20-24 April 2009). *A novel LDPC decoder for DVB-S2 IP*. Paper presented at the Design, Automation & Test in Europe Conference & Exhibition, 2009. DATE '09.
- [19] **Gallager, R. G.** (1962). Low-density parity-check codes. *Information Theory, IRE Transactions on*, 8(1), 21-28. doi: 10.1109/TIT.1962.1057683
- [20] **William, R.** (2001). An Introduction To Low-Density Parity-Check Codes, University of Arizona.
- [21] **Tanner, R. M.** (1981). A recursive approach to low complexity codes. *Information Theory, IEEE Transactions on*, 27(5), 533-547. doi: 10.1109/TIT.1981.1056404
- [22] **John, B.** (2001). Low Density Parity check Codes, Georgia Institute of Technology.
- [23] **Campello, J., & Modha, D. S.** (2001). *Extended bit-filling and LDPC code design*. Paper presented at the Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE.
- [24] **MacKay, D. J. C.** (1999). Good error-correcting codes based on very sparse matrices. *Information Theory, IEEE Transactions on*, 45(2), 399-431. doi: 10.1109/18.748992
- [25] **Wei, W., Sangjin, H., & Do-Sik, Y.** (2003, 22-25 April 2003). *Block length of LDPC codes in fading channels*. Paper presented at the Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual.
- [26] **Sae-Young, C., Richardson, T. J., & Urbanke, R. L.** (2001). Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation. *Information Theory, IEEE Transactions on*, 47(2), 657-670. doi: 10.1109/18.910580

- [27] **Fossorier, M. P. C., Mihaljevic, x, M., & Imai, H.** (1999). Reduced complexity iterative decoding of low-density parity check codes based on belief propagation. *Communications, IEEE Transactions on*, 47(5), 673-680. doi: 10.1109/26.768759
- [28] **Abrahart, R. ve See, L.** (2000). Comparing neural network and autoregressive moving average techniques for the provision of continuous river flow forecasts in two contrasting catchments, *Hydrolog. Process.*, **14**, 2157–2172.
- [29] **Lehmann, F., & Maggio, G. M.** (2003). Analysis of the iterative decoding of LDPC and product codes using the Gaussian approximation. *Information Theory, IEEE Transactions on*, 49(11), 2993-3000. doi: 10.1109/TIT.2003.819335
- [30] **Fan, J.** (1999). Constrained coding and soft iterative decoding for storage, Ph.D. dissrtation, Stanford Universiy.
- [31] **Sakib, M. N., Mahalingam, V., Gross, W. J., & Liboiron-Ladouceur, O.** (2011). Optical Front-End for Soft-Decision LDPC Codes in Optical Communication Systems. *Optical Communications and Networking, IEEE/OSA Journal of*, 3(6), 533-541. doi: 10.1364/JOCN.3.000533
- [32] **Bhagavatula, V., Hongwei, Song, & Jingfeng, Liu.** (2002, 2002). *Low density parity check (LDPC) codes for optical data storage*. Paper presented at the Optical Memory and
- [33] **Bard, M., Markus, M. ve Geir, E.** (1993). LDPC coded adaptive modulation for slowly varying rayleigh-fading channels.
- [34] **Jilei, Hou, Siegel, P. H., & Milstein, L. B.** (2001). Performance analysis and code optimization of low density parity-check codes on Rayleigh fading channels. *Selected Areas in Communications, IEEE Journal on*, 19(5), 924-934. doi: 10.1109/49.924876
- [35] **Yang, M., & Ryan, W. E.** (2002, 27-30 Oct. 2002). *Design of LDPC codes for two-state fading channel models*. Paper presented at the Wireless Personal Multimedia Communications, 2002. The 5th International Symposium on.
- [36] **Guo, F., Ng, S. X., & Hanzo, L.** (2003, 22-25 April 2003). *LPDC assisted block coded modulation for transmission over Rayleigh fading channels*. Paper presented at the Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual.
- [37] **Hagenauer, J., & Hoeher, P.** (1989, 27-30 Nov 1989). *A Viterbi algorithm with soft-decision outputs and its applications*. Paper presented at the Global Telecommunications Conference and Exhibition 'Communications Technology for the 1990s and Beyond' (GLOBECOM), 1989. IEEE.
- [38] **Jones, C., Tao, Tian, Matache, A., Wesel, R., & Villasenor, J.** (2002, 17-21 Nov. 2002). *Robustness of LDPC codes on periodic fading channels*. Paper presented at the Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE.

- [39] **Wei, W., Sangjin, H., & Do-Sik, Y.** (2003, 22-25 April 2003). *Block length of LDPC codes in fading channels*. Paper presented at the Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual.
- [40] **Bard, M.** (2008). LDPC- coded adaptive modulation on slowly varying Nakagami-Fading channels, *SINTEF Telecom and Informatics*, NORWAY.
- [41] **Chiani, M., Conti, A., & Ventura, A.** (2000). *Evaluation of low-density parity-check codes over block fading channels*. Paper presented at the Communications, 2000. ICC 2000. 2000 IEEE International Conference on.
- [42] **Bard, M., Vidar, M., Geir, E.** (2001). LDPC coded adaptive multilevel modulation for slowly varying rayleigh-fading channels, Øien Dept of Telecommunications, NTNU.
- [43] **Hongzin, S., & Cruz, J. R.** (2003). Reduced-complexity decoding of Q-ary LDPC codes for magnetic recording. *Magnetics, IEEE Transactions on*, 39(2), 1081-1087. doi: 10.1109/TMAG.2003.808600
- [44] **Hongzin, S., Todd, R. M., & Cruz, J. R.** (2000). Low density parity check codes for magnetic recording channels. *Magnetics, IEEE Transactions on*, 36(5), 2183-2186. doi: 10.1109/20.908351
- [45] **Sae-Young, C., Richardson, T. J., & Urbanke, R. L.** (2001). Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation. *Information Theory, IEEE Transactions on*, 47(2), 657-670. doi: 10.1109/18.910580
- [46] **Sridharan, A., Costello, D. J., Jr., & Tanner, R. M.** (2002). *A construction for low density parity check convolutional codes based on quasi-cyclic block codes*. Paper presented at the Information Theory, 2002. Proceedings. 2002 IEEE International Symposium on
- [47] **Saber, H., Olfat, E., & Amindavar, H.** (2010, 9-15 May 2010). *Joint Multiuser LDPC Coding Scheme for Error Control in DS-CDMA Systems*. Paper presented at the Telecommunications (AICT), 2010 Sixth Advanced International Conference on.
- [48] **Patel, Arvind M.** (1989). Two-level coding for error control in magnetic disk storage products. *IBM Journal of Research and Development*, 33(4), 470-484. doi: 10.1147/rd.334.0470
- [49] **Jimenez Felstrom, A., & Zigangirov, K. S.** (1999). Time-varying periodic convolutional codes with low-density parity-check matrix. *Information Theory, IEEE Transactions on*, 45(6), 2181-2191. doi: 10.1109/18.782171
- [50] **Sridharan, A., Costello, D. J., Jr., & Tanner, R. M.** (2002, 2002). *A construction for low density parity check convolutional codes based on quasi-cyclic block codes*. Paper presented at the Information Theory, 2002. Proceedings. 2002 IEEE International Symposium on.

- [51] **Kurkoski, B. M., Siegel, P. H., & Wolf, J. K.** (2003). Joint message-passing decoding of ldpc codes and partial-response channels. *Information Theory, IEEE Transactions on*, 49(8), 2076-2076. doi: 10.1109/TIT.2003.814494
- [52] **Mittelholzer, T., Dholakia, A., & Eleftheriou, E.** (2001). Reduced-complexity decoding of low density parity check codes for generalized partial response channels. *Magnetics, IEEE Transactions on*, 37(2), 721-728. doi: 10.1109/20.917607
- [53] **Todd, R. ve Cruz, J.** (1999). Designing good LDPC codes for partial response channels, unpublished preprint.
- [54] **Varnica, N., & Kavcic, A.** (2002, 2002). *Optimized LDPC codes for partial response channels*. Paper presented at the Information Theory, 2002. Proceedings. 2002 IEEE International Symposium on.
- [55] **Changyan, Di, Proietti, D., Telatar, I. E., Richardson, T. J., & Urbanke, R. L.** (2002). Finite-length analysis of low-density parity-check codes on the binary erasure channel. *Information Theory, IEEE Transactions on*, 48(6), 1570-1579. doi: 10.1109/TIT.2002.1003839
- [56] **Jeongseok, H., & McLaughlin, S. W.** (2003). Low-density parity-check codes over Gaussian channels with erasures. *Information Theory, IEEE Transactions on*, 49(7), 1801-1809. doi: 10.1109/TIT.2003.813480
- [57] **Ratzer, E. A., & MacKay, D. J. C.** (2003, 31 March-4 April 2003). *Sparse low-density parity-check codes for channels with cross-talk*. Paper presented at the Information Theory Workshop, 2003. Proceedings. 2003 IEEE.
- [58] **Chih-Chun, Wang, Kulkarni, S. R., & Poor, H. V.** (2005). Density evolution for asymmetric memoryless channels. *Information Theory, IEEE Transactions on*, 51(12), 4216-4236. doi: 10.1109/TIT.2005.858931
- [59] **Garcia-Frias, J.** (2004). Decoding of low-density parity-check codes over finite-state binary Markov channels. *Communications, IEEE Transactions on*, 52(11), 1840-1843. doi: 10.1109/TCOMM.2004.836568
- [60] **Saber, H., Olfat, E., & Amindavar, H.** (2010, 9-15 May 2010). *Joint Multiuser LDPC Coding Scheme for Error Control in DS-CDMA Systems*. Paper presented at the Telecommunications (AICT), 2010 Sixth Advanced International Conference on.
- [61] **Yang, M., & Ryan, W. E.** (2001, 2001). *Performance of (quasi-)cyclic LDPC codes in noise bursts on the EPR4 channel*. Paper presented at the Global Telecommunications
- [62] **Jilei, Hou, Siegel, P. H., Milstein, L. B., & Pfister, H. D.** (2001, 2001). *Design of low-density parity-check codes for bandwidth efficient modulation*. Paper presented at the Information Theory Workshop, 2001. Proceedings. 2001 IEEE.
- [63] **Fano, R.** (1961). *Transmission of Information*, The M.I.T. Press and John Wiley & Sons, Inc. New York.
- [64] **Shokrollahi, A.** (2003). *LDPC Codes: An Introduction*, Digital Fountain, Inc.

- [65] **Gallager, R.** (1963). *Low Density Parity Check Codes*. Cambridge, MA: MIT Press.
- [66] **MacKay, D. J. C., & Neal, R. M.** (1996). Near Shannon limit performance of low density parity check codes. *Electronics Letters*, 32(18), 1645. doi: 10.1049/el:19961141
- [67] **Jianguang, Z., Zarkeshvari, F., & Banihashemi, A. H.** (2005). On implementation of min-sum algorithm and its modifications for decoding low-density Parity-check (LDPC) codes. *Communications, IEEE Transactions on*, 53(4), 549-554. doi: 10.1109/TCOMM.2004.836563
- [68] **Lee, J. K. S., & Thorpe, J.** (2005, 4-9 Sept. 2005). *Memory-efficient decoding of LDPC codes*. Paper presented at the Information Theory, 2005. ISIT 2005. Proceedings. International Symposium on.
- [69] **URL-1**<<http://itpp.sourceforge.net.2013>>, alındığı tarih: 17.06.2013.

EKLER

EK A:

```
*****
***** başlangıç bölümü
*****

qij[0]
0.689974 0.689974 0.689974 0.689974
0.689974 0.689974 0.689974 0.689974
0.026597 0.026597 0.026597 0.026597
0.916827 0.916827 0.916827 0.916827
0.880797 0.880797 0.880797 0.880797
0.012128 0.012128 0.012128 0.012128
0.167982 0.167982 0.167982 0.167982
0.008163 0.008163 0.008163 0.008163
***** başlangıç bölümü sonu
*****

rji[0]
0.320131 0.320131 0.572181 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.128440 0.093284 0.817453 0.000000 0.000000
0.223211 0.000000 0.000000 0.373850 0.000000 0.000000 0.658373 0.000000
0.000000 0.125419 0.000000 0.000000 0.313127 0.000000 0.000000 0.644683
qij[0]
0.390063 0.689974 0.511705 0.689974
0.241939 0.689974 0.689974 0.511705
0.026597 0.026597 0.026597 0.026597
0.916827 0.868100 0.618969 0.916827
0.880797 0.771087 0.880797 0.431880
0.012128 0.012128 0.012128 0.012128
0.167982 0.167982 0.167982 0.167982
0.008163 0.008163 0.008163 0.008163
qi[0]
0.231436 0.130647 0.035255 0.492361 0.257362 0.052113 0.280104 0.014712
1. iterasyon ci degerleri 1 1 1 1 1 1 1
parite kontrol degerleri 1 1 1
*****
*****

rji[0]
0.744334 0.604089 0.556741 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.235489 0.140829 0.699574 0.000000 0.000000
0.421000 0.000000 0.000000 0.492227 0.000000 0.000000 0.502785 0.000000
0.000000 0.567008 0.000000 0.000000 0.488486 0.000000 0.000000 0.498405
qij[0]
0.618062 0.689974 0.866298 0.689974
0.744531 0.689974 0.689974 0.772509
0.026597 0.026597 0.026597 0.026597
```

```

0.916827 0.914425 0.772490 0.916827
0.880797 0.875875 0.880797 0.547748
0.012128 0.012128 0.012128 0.012128
0.167982 0.167982 0.167982 0.167982
0.008163 0.008163 0.008163 0.008163
qi[0]
0.824906 0.816406 0.033180 0.766979 0.536314 0.027794 0.169544 0.008111
2. iterasyon ci degerleri 0 0 1 0 0 1 1 1
parite kontrol degerleri 1 1 1 1
*****
*****
rji[0]
0.268476 0.388218 0.557740 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.133242 0.095627 0.811544 0.000000 0.000000
0.319056 0.000000 0.000000 0.256764 0.000000 0.000000 0.699625 0.000000
0.000000 0.453031 0.000000 0.000000 0.231940 0.000000 0.000000 0.526024
qij[0]
0.510471 0.689974 0.449580 0.689974
0.648299 0.689974 0.689974 0.585451
0.026597 0.026597 0.026597 0.026597
0.916827 0.792020 0.628879 0.916827
0.880797 0.690532 0.880797 0.438616
0.012128 0.012128 0.012128 0.012128
0.167982 0.167982 0.167982 0.167982
0.008163 0.008163 0.008163 0.008163
qi[0]
0.276782 0.539110 0.033310 0.369248 0.190900 0.050215 0.319845 0.009051
3. iterasyon ci degerleri 1 0 1 1 1 1 1 1
parite kontrol degerleri 0 1 1 0
*****
*****
rji[0]
0.359590 0.490086 0.503106 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.314089 0.215063 0.611279 0.000000 0.000000
0.414419 0.000000 0.000000 0.533481 0.000000 0.000000 0.487004 0.000000
0.000000 0.560382 0.000000 0.000000 0.415944 0.000000 0.000000 0.489509
qij[0]
0.611655 0.689974 0.555484 0.689974
0.739373 0.689974 0.689974 0.681428
0.026597 0.026597 0.026597 0.026597
0.916827 0.926500 0.834648 0.916827
0.880797 0.840312 0.880797 0.669368
0.012128 0.012128 0.012128 0.012128
0.167982 0.167982 0.167982 0.167982
0.008163 0.008163 0.008163 0.008163
qi[0]
0.469321 0.731659 0.026921 0.852338 0.590464 0.018941 0.160839 0.007830
4. iterasyon ci degerleri 1 0 1 0 0 1 1 1
parite kontrol degerleri 0 1 0 1

```

```

*****
*****
rji[0]
0.273360 0.394284 0.553455 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.167943 0.083846 0.790286 0.000000 0.000000
0.277782 0.000000 0.000000 0.463156 0.000000 0.000000 0.537135 0.000000
0.000000 0.333397 0.000000 0.000000 0.321534 0.000000 0.000000 0.561456
qij[0]
0.461205 0.689974 0.455706 0.689974
0.526759 0.689974 0.689974 0.591619
0.026597 0.026597 0.026597 0.026597
0.916827 0.904854 0.689915 0.916827
0.880797 0.777865 0.880797 0.403428
0.012128 0.012128 0.012128 0.012128
0.167982 0.167982 0.167982 0.167982
0.008163 0.008163 0.008163 0.008163
qi[0]
0.243583 0.420139 0.032756 0.657479 0.242700 0.044220 0.189819 0.010426
5. iterasyon ci degerleri 1 1 1 0 1 1 1 1
parite kontrol degerleri 1 0 0 1
*****
*****
rji[0]
0.474665 0.536731 0.497924 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.228875 0.104967 0.724990 0.000000 0.000000
0.373890 0.000000 0.000000 0.529413 0.000000 0.000000 0.483176 0.000000
0.000000 0.594996 0.000000 0.000000 0.409877 0.000000 0.000000 0.482304
qij[0]
0.570633 0.689974 0.667871 0.689974
0.765784 0.689974 0.689974 0.720550
0.026597 0.026597 0.026597 0.026597
0.916827 0.925379 0.765904 0.916827
0.880797 0.836925 0.880797 0.464257
0.012128 0.012128 0.012128 0.012128
0.167982 0.167982 0.167982 0.167982
0.008163 0.008163 0.008163 0.008163
qi[0]
0.545624 0.791146 0.026383 0.786358 0.375735 0.031351 0.158782 0.007609
6. iterasyon ci degerleri 0 0 1 0 1 1 1 1
parite kontrol degerleri 1 0 1 0
*****
*****
rji[0]
0.248354 0.433124 0.537546 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.171248 0.084939 0.786642 0.000000 0.000000
0.323430 0.000000 0.000000 0.388528 0.000000 0.000000 0.589275 0.000000
0.000000 0.535160 0.000000 0.000000 0.283051 0.000000 0.000000 0.484234
qij[0]
0.515482 0.689974 0.423747 0.689974
0.719276 0.689974 0.689974 0.629690

```

```

0.026597 0.026597 0.026597 0.026597
0.916827 0.875064 0.694913 0.916827
0.880797 0.744715 0.880797 0.406837
0.012128 0.012128 0.012128 0.012128
0.167982 0.167982 0.167982 0.167982
0.008163 0.008163 0.008163 0.008163
qi[0]
0.260097 0.661896 0.030783 0.591382 0.213084 0.043306 0.224605 0.007667
7. iterasyon ci degerleri 1 0 1 0 1 1 1 1
parite kontrol degerleri 0 0 0 0
*****
*****
rji[0]
0.292388 0.485342 0.506790 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.261221 0.134034 0.683568 0.000000 0.000000
0.370571 0.000000 0.000000 0.550635 0.000000 0.000000 0.470275 0.000000
0.000000 0.591642 0.000000 0.000000 0.372428 0.000000 0.000000 0.475835
qij[0]
0.567150 0.689974 0.479059 0.689974
0.763282 0.689974 0.689974 0.677291
0.026597 0.026597 0.026597 0.026597
0.916827 0.931070 0.795820 0.916827
0.880797 0.814298 0.880797 0.533512
0.012128 0.012128 0.012128 0.012128
0.167982 0.167982 0.167982 0.167982
0.008163 0.008163 0.008163 0.008163
qi[0]
0.351243 0.752522 0.027309 0.826870 0.404303 0.025837 0.151995 0.007416
8. iterasyon ci degerleri 1 0 1 0 1 1 1 1
parite kontrol degerleri 0 0 0 0
*****
*****
rji[0]
0.250723 0.436422 0.535359 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.193326 0.079387 0.770969 0.000000 0.000000
0.303565 0.000000 0.000000 0.513905 0.000000 0.000000 0.487611 0.000000
0.000000 0.467036 0.000000 0.000000 0.325603 0.000000 0.000000 0.511883
qij[0]
0.492406 0.689974 0.426839 0.689974
0.661044 0.689974 0.689974 0.632813
0.026597 0.026597 0.026597 0.026597
0.916827 0.920973 0.725410 0.916827
0.880797 0.781061 0.880797 0.389193
0.012128 0.012128 0.012128 0.012128
0.167982 0.167982 0.167982 0.167982
0.008163 0.008163 0.008163 0.008163
qi[0]
0.245059 0.601629 0.030521 0.736352 0.235260 0.039688 0.161167 0.008557
9. iterasyon ci degerleri 1 0 1 0 1 1 1 1
parite kontrol degerleri 0 0 0 0

```

```

*****
*****
rji[0]
0.347522 0.507190 0.497554 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.225756 0.089239 0.736638 0.000000 0.000000
0.350320 0.000000 0.000000 0.548582 0.000000 0.000000 0.467018 0.000000
0.000000 0.608998 0.000000 0.000000 0.369355 0.000000 0.000000 0.470567
qij[0]
0.545465 0.689974 0.542411 0.689974
0.776104 0.689974 0.689974 0.696093
0.026597 0.026597 0.026597 0.026597
0.916827 0.930535 0.762706 0.916827
0.880797 0.812298 0.880797 0.419953
0.012128 0.012128 0.012128 0.012128
0.167982 0.167982 0.167982 0.167982
0.008163 0.008163 0.008163 0.008163
qi[0]
0.389936 0.781062 0.026345 0.796168 0.297768 0.033200 0.150316 0.007262
10. iterasyon ci degerleri 1 0 1 0 1 1 1 1
parite kontrol degerleri 0 0 0 0
*****
*****

```


ÖZGEÇMİŞ

Ad Soyad: Şakir Balta

Doğum Yeri ve Tarihi: Kastamonu/Tosya 01.01.1986

Adres:

E-Posta: sakir.balta@itu.edu.tr

Lisans : Elektronik Mühendisliği/Işık Üniversitesi

Lisans(ÇAP) : Endüstri Mühendisliği/Işık Üniversitesi

Yüksek Lisans : Telekomünikasyon Mühendisliği/İstanbul Teknik Üniversitesi

Mesleki Deneyim ve Ödüller: Araştırmacı, TÜBİTAK BİLGEM/UEKAE(2010-.....)

Yayın ve Patent Listesi:

- Ş.Balta, S. Altunbay, “Elektrik Enerji Kaynakları Optimizasyonu”, Yöneylem Araştırması ve Endüstri Mühendisliği 31.Ulusal Kongresi, Sakarya, Temmuz 2011.