T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

HÜCRESEL DÖNÜŞÜMLERLE HATA DÜZELTEN KODLAR

MEHMET EMİN KÖROĞLU

YÜKSEK LİSANS TEZİ
MATEMATİK ANABİLİM DALI

DANIŞMAN
PROF. DR. İRFAN ŞİAP

İSTANBUL, 2012

T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

HÜCRESEL DÖNÜŞÜMLERLE HATA DÜZELTEN KODLAR

Mehmet Emin KÖROĞLU tarafından hazırlanan tez çalışması 27.07.2012 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Matematik Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Prof. Dr. İrfan ŞİAP

Yıldız Teknik Üniversitesi

Jüri Üyeleri

Prof. Dr. İrfan ŞİAP

Yıldız Teknik Üniversitesi

Doç. Dr. Bayram Ali ERSOY

Yıldız Teknik Üniversitesi

Yrd. Doç. Dr. Bahattin YILDIZ

Fatih Üniversitesi

ÖNSÖZ

Gerek lisans gerekse yüksek lisans öğrenimim boyunca sevgili ailemin değerli üyelerinden gördüğüm maddi ve manevi yardımın hakkıyla yerine getirilmesi güç bir şükran borcu teşkil ettiğini teslim ederim.

Tez danışmanlığımı üstlenen ve çok yönlü desteklerini esirgemeyen tanışıklığımızı fırsat addettiğim değerli hocam Prof. Dr. Sayın İrfan ŞİAP'a şükranlarımı sunarım. Ayrıca, tez jüriliğinde yer alan ve yapıcı katkılarını esirgemeyen Doç. Dr. Sayın Bayram Ali ERSOY ve Yrd. Doç. Dr. Sayın Bahattin YILDIZ'a çok teşekkür ederim. Son olarak kısmi desteklerinden dolayı TÜBİTAK'a teşekkürlerimi bildiririm.

Mart, 2012

Mehmet Emin KÖROĞLU

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ

$\mu^{(i)}(x)$	α ilkel elemanının $\mathbb{F}_q$'daki minimal polinomunu
$R_n = \mathbb{F}_q[x] \Big/ \langle x^n - 1 \rangle$	Bölüm halkası
$C \leq_{a.v.} \mathbb{F}_q^n$	C, $\mathbb{F}_q^n$ nin alt vektör uzayıdır
$C^{\perp}$	C lineer kodunun duali
$\lvert C \rvert$	C lineer kodunun eleman sayısı
$boy(C)$	C lineer kodunun alt vektör uzayı olarak boyutu
G	C lineer kodunun üreteç matrisi
H	C lineer kodunun kontrol matrisi
$d(C)$	C kodunun minimum Hamming uzaklığı
$\varphi(C)$	C devirli kodunun φ altındaki görüntüsü
L_d	$d-$ boyutlu latis ağı
$\det(.)$	Determinant fonksiyonu
S_{aug}	Eklemeli sendrom
$\left[T_{aug}\right]$	Eklemeli T matrisi
$\forall$	Evrensel niceleyici her
$der(f(x))$	$f(x)$ polinomunun derecesi
$ekok\left[f(x), g(x)\right]$	$f(x)$ ile $g(x)$ polinomlarının en küçük ortak katı
$ebob(f(x), g(x))$	$f(x)$ ile $g(x)$ polinomlarının en büyük ortak böleni
$\mathbb{F}_q^*$	$\mathbb{F}_q - \{0\}$
x_i^t	t zaman adımındaki i. hücre
$(n, M)_q$	$\mathbb{F}_q$ alfabesi üzerinde $n-$uzunluklu, M elemanlı kod
$[n, k]_q$	$\mathbb{F}_q$ alfabesi üzerinde $n-$uzunluklu, q^k elemanlı lineer kod
$\mathbb{F}_q^n$	$\mathbb{F}_q$ cismi üzerinde $n-$boyutlu vektör uzayı
$\langle x^n - 1 \rangle$	$\mathbb{F}_q[x]$ polinom halkasının bir ideali

$C=\langle g(x)\rangle$	$g(x)$ polinomunun ürettiği devirli kod
F	Global geçiş fonksiyonu
$E=(I_e, C_e)$	Hata vektörü
B_i	i . bilgi baytı
S_i	i . hata sendromu
E_i	i . hata vektörü
C_i	i . kontrol baytı
$\mathbb{F}_q[x]$	Katsayıları $\mathbb{F}_q$ 'dan olan polinomlar halkası
CW	Kod söz
$O(.)$	Kompleksity derecesi
r	Komşuluk yarıçapı
$\oplus$	$(\mathrm{mod}\ q)$ toplama işlemi
$\mu_i(k_i)$	k_i-uzunluğunda μ_i-tane devir
f	Lokal geçiş fonksiyonu
$\mathbb{Z}_p$	$(\mathrm{mod}\ p)$ kalan sınıfları
p	Mümkün durumların sayısı
D	Mümkün bütün durumların kümesi
N	Mümkün komşuluk durumları
$I=(i_1, i_2, ..., i_n)$	$n-$bitten oluşan bilgi bitleri
$T^k[I]=(c_1, c_2, ..., c_n)$	$n-$bitten oluşan kontrol bitleri
T_n	$n\times n$ tipinde regüler matris
ə	Öyleki
$G=[I_k \mid A]$	Standart formdaki üreteç matrisi
$H=[-A \mid I_{n-k}]$	Standart formdaki kontrol matrisi
S	Sendrom
T	Temsili matris
$\circ(T)$	T matrisinin mertebesi
$\varphi(x)$	T matrisinin karakteristik polinomu
$\mathbb{F}_q$	q elemanlı ilkel cisim
$Ham(r,q)$	q 'lu Hamming kodu
$s(u)$	u 'nun sendromu
$\langle x, y\rangle$	x ile y vektörlerinin iç çarpımı
$d_H(x,y)$	x ile y vektörleri arasındaki Hamming uzaklığı
$w_H(x)$	x vektörünün Hamming ağırlığı

KISALTMA LİSTESİ

bkz.	Bakınız
BCH	Bose, Chaudhuri, Hocquenghem
CA	Cellular Automata
CAECC	Cellular Automata Based Error Correcting Code
DNA	Deoksiribonükleik asit
IEEE	Institute of Electrical and Electronics Engineers
LDPC	Low Density Parity Check
MDS	Maximum Distance Separable
PLA	Programmable Logic Arrays
vb.	ve benzeri
vd.	ve diğerleri
VLSI	Very Large Scale Integtation

ŞEKİL LİSTESİ

Sayfa

TABLO LİSTESİ

Sayfa

ÖZET

HÜCRESEL DÖNÜŞÜMLERLE HATA DÜZELTEN KODLAR

Mehmet Emin KÖROĞLU

Matematik Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Prof. Dr. İrfan ŞİAP

1948 yılında yayınladığı çalışmasında Claude E. Shannon ilk kez gürültülü bir kanal üzerinden yapılan iletişim için kanal kapasitesi denilen bir kavram ortaya koydu. Shannon, eğer uygun kodlama ve dekodlama teknikleri kullanılırsa kanal kapasitesinin altında herhangi bir oranda güvenli iletişimin teorik olarak mümkün olduğunu kanıtladı. Ancak Shannon bahsedilen uygun kodlama ve dekodlama algoritmalarına ilişkin herhangi bir metot önermiyordu. Richard W. Hamming 1950 yılında Shannon'ın varlığını kanıtladığı uygun kodlama ve dekodlama yeteneğine sahip ilk kod ailesini buldu. Aynı yıl Golay tarafından da bir kod ailesi keşfedildi. Sırasıyla Hamming ve Golay kodları olarak bilinen bu iki kod ailesi (lineer blok kod) bilinen ilk hata düzelten optimal kodlardır.

Hata düzelten kodlar dijital iletişim, haberleşme uyduları, uzay araştırmaları, dijital bilgi depolama gibi birçok alanda yaygın olarak kullanılmaktadır. Bu kodlama tekniğinin temel amacı herhangi bir bilgi kodlandıktan sonra gerek iletim esnasında gerekse depolanan bilginin geri çağrılması esnasında oluşabilecek hataları belli şartlar altında fark etmek ve hatta düzeltmektir. Bunun için temel olarak bilgi bitlerine belli sayıda kontrol bitleri eklenmektedir. Hücresel dönüşümlerle hata düzelten kodların da temel amacı diğer hata düzelten kodlarda olduğu gibi Shannon kapasitesine yakın kodlama ve dekodlama yapabilen verimli algoritmalar geliştirmektir.

Bu çalışmada ilk olarak hücresel dönüşümler ve hücresel dönüşümlerle hata düzelten kodların tarihçesi hakkında bilgi verilmiştir. İkinci bölümde hücresel dönüşümlerle ilgili bilgiler zaman zaman ayrıntılı sayılabilecek biçimde sunulmuştur. Üçüncü bölümde hata düzelten kodlar ile ilgili gerekli bilgiler sıralandıktan sonra literatürde yapılan

çalışmalar sunulmuştur. Dördüncü bölümde ise daha önce ikili cisim üzerinde yapılmış olan çalışmalar $\mathbb{F}_q$ ilkel cisimleri üzerine genellenmiştir. Son bölüm ise sonuç ve önerilere ayrılmıştır.

Anahtar Kelimeler: Hücresel Dönüşümler, Hata Düzelten Kodlar, Sonlu Cisimler

ABSTRACT

CELLULAR AUTOMATA BASED ERROR CORRECTİNG CODES

Mehmet Emin KÖROĞLU

Department of Mathematics

MSc. Thesis

Advisor: Prof. Dr. İrfan ŞİAP

In the paper published in 1948, Claude E. Shannon for the first time revealed the concept of communication through a noisy channel called channel capacity. Shannon proved that if suitable encoding and decoding techniques were used, then reliable communication theoretically could be possible at any rate below the channel capacity. However, Shannon didn't come up with any method for the suitable encoding and decoding techniques. In 1950, Richard W. Hamming found the first code family which had suitable encoding and decoding ability the existence of which had been proved by Shannon. In the same year, a code family was also discovered by Golay. These two code families (linear block codes), known as the Hamming and Golay codes, respectively, are the first known optimal error correcting codes.

Error correcting codes are widely used in many areas such as digital communication, communication satellites, space research, and storage of digital information. The main objective of the coding technique is to detect possible errors that may occur and even correct them under certain conditions, when any information has been encoded both during transmission and retrieval of information stored. Towards this end, basically certain number of check bits are added to the information bits. The main objective of the cellular automata based error correcting codes, similar to classical error correcting codes, is to develop effective algorithms which have an encoding and decoding capacity close to that of Shannon.

In this thesis, information about the history of cellular automata and cellular automata based error correcting codes is given first. In the second section, further information regarding cellular automata is introduced and some concepts are also studied in detail. In the third section after covering the information required for error correcting codes, recent studies in the literature are presented. The fourth section contains the generalization to primitive finite fields $\mathbb{F}_q$ of an original study made on binary fields. The last section is reserved for the conclusions and future research directions.

Key words: Cellular Automata, Error Correcting Codes, Finite Fields

BÖLÜM 1

HÜCRESEL DÖNÜŞÜMLER ve HÜCRESEL DÖNÜŞÜMLERLE HATA DÜZELTEN KODLARIN TARİHÇESİ

1.1 Hücresel Dönüşümlerin Tarihçesi

Literatürde cellular automata (çoğul), cellular automaton (tekil) ya da automata theory olarak adlandırılan hücresel dönüşümler ilk olarak, adı tam olarak konmasa da, 1930–1940 yılları arasında Rus araştırmacılar tarafından çalışılmıştır [1]. Automata theory ile ilgili adı konulmuş ilk çalışmalar Polonya asıllı Amerikan vatandaşı matematikçi Stanislaw Marcin Ulam ve Macar asıllı Amerikan vatandaşı matematikçi John von Neumann'ın karşılıklı çalışma ve tartışmaları sonucu ortaya çıkmıştır. Ulam 1940'lı yılların sonunda kristallerin gelişimini (crystal growth) ya da donmayı matematiksel bir model ile ifade etmek için bazı çalışmalar yaptı [3], [4], [5]. Aynı yıllarda Neumann DNA'nın (Deoksiribonükleik asit) kendini kopyalamasından (self reproducing) hareketle iki boyutlu 29-durumlu ve kendi adını taşıyan dört hücreli (merkez hücre ile beraber beş hücreli) özel komşuluğu (Neumann komşuluğu) kullanan özel bir hücresel dönüşümü bilim dünyasıyla tanıştırdı [2], [5], [6]. Neumann'nın kendi kendini yeniden üreten makine (self reproducing machine) olarak adlandırılan bu dönüşümü gerçek manada modellenememiştir [7]. Ancak sonraki yıllarda bu dönüşümün daha basit (az sayıda durum içeren) örnekleri modellenmeye çalışılmıştır [9], [10]. 1960'lı yıllara gelindiğinde bu alandaki çalışmaların sayısında ciddi bir artış yaşandı. Bu yıllarda hücresel dönüşümler bir tür özel dinamik sistem olarak çalışılmakla beraber hücresel dönüşümlerin matematiksel özellikleri de irdelenmeye başlandı [8]. 1970'li yıllarda da bu alandaki çalışmalar katlanarak arttı. Bunun yanı sıra 1970 yılında "*The Game of Life*"

isimli iki boyutlu hücresel dönüşüm, daha önce John Horton Conway tarafından tanımlanmıştı, Martin Gardner tarafından "*Scientific American*" dergisinde yayınlanan "*Matematiksel Oyunlar*" isimli makaleye konu edilerek hücresel dönüşümlerin kalabalık kitlelerce tanınması sağlandı [11]. Daha sonra bu kuralı temel alan oyunlar yaygınlaştı. 1980'li yılların başından itibaren Stephen Wolfram hücresel dönüşümlerle ilgili olarak sistematik bir şekilde makaleler yayınlamaya başladı. Bu makalelerde, bir boyutlu temel hücresel dönüşümler (daha sonra Wolfram kuralları olarak adlandırılmışlardır) detaylı bir şekilde tanıtılmakla beraber bu dönüşümlerin bir takım matematiksel özellikleri irdelenmiş ve çeşitli uygulama alanlarıyla ilişkilendirilmeye çalışılmıştır [12]. Wolfram 2002 yılında yayınladığı "*The New Kind of Science"* isimli kitabında bu çalışmalarını bir araya getirmiştir. 1980'li yıllardan itibaren hücresel dönüşümler ile birçok bilim dalı arasında irtibat kurulmaya çalışılmıştır. Bu çabanın bir sonucu olarak kodlama, kriptografi, dijital fotoğraf işleme, trafik akışı, şehir planlama, genetik gibi konularda birçok yayın yapılmıştır [1], [2], [13], [14], [15], [16], [17], [18], [19], [20], [21].

1.2 Hücresel Dönüşümlerle Hata Düzelten Kodların Tarihçesi

Automata ile kodlama teorisi ilk olarak James L. Massey'in 1967 yılında yayınladığı çalışmasıyla ilişkilendirilmeye çalışılmıştır [22], [23]. Hücresel dönüşümlerle hata düzelten kodlar ile ilgili ilk çalışma 1994 yılında bir boyutlu lineer hücresel dönüşümlerin bir geçiş matrisiyle temsil edilebileceğinin gösterilmesinden sonra [26], Dipanwita Roy Chowdhury vd. tarafından yayınlandı [24]. Bu çalışmada bir boyutlu lineer hücresel dönüşümlerin geçiş matrisleri üzerinde bir takım kısıtlamalar yapılarak lineer blok kodlar ailesine ait rastgele bit hata düzelten kodlara alternatif kodlama ve dekodlama algoritmaları verildi. Bu çalışmada temel olarak bir boyutlu lineer hücresel dönüşümlerin tersinirlik özelliği kullanılmıştır. Tersinir lineer dönüşümlerin geçiş matrislerinin terslenebilirlik özelliğini kullanan bu dekodlama algoritması, klasik bit hata düzelten lineer blok kodlara nazaran hem daha basit ve kolay bilgisayar donanımı gerektirmekte hem de daha az işlem gerektirmektedir [15], [24], [25]. Chowdhury vd.' nin bu çalışması kodlama teorisinde hata düzelten kodlar ailesinin bilinen ilk örneklerinden olan ikili Hamming kodlarını esas almıştır. Aynı yazar 1995 yılında bu

çalışmadaki mantığı blok hata düzeltebilen ve özel bir devirli kod sınıfı olan Reed-Solomon kodlara uyguladı [25]. Böylece bir boyutlu lineer hücresel dönüşümlerin geçiş matrisleri yardımıyla t-bayt hata düzeltebilen yeni bir kodlama ve dekodlama algoritması ortaya çıktı. Cellular automata ile bayt hata düzelten kodlar 2008 yılında Bhaumik vd. tarafından geliştirildi [36]. Sözkonusu tüm çalışmalarda iki durumlu bir boyutlu lineer hücresel dönüşümlerin geçiş matrisleri ile sıfır sınır şartı kullanılmıştır. 2004 yılına kadar bu konuda yeni çalışma yapılmamıştır. 2004 yılına gelindiğinde Koreli araştırmacı Sung-Jın Cho vd. bir boyutlu lineer hücresel dönüşümlerin geçiş matrisleri ile periyodik sınır şartını kullanarak 1-hata düzeltebilen bir kodlama ve dekodlama algoritması tanımladılar [37]. Bu çalışma 2006 yılında 2-hataya genişletildi [26]. Bu çalışma büyük ölçüde Chowdhury vd.'nin çalışmasına dayanmaktadır. Bahsi geçen hücresel dönüşümlerle yapılan hata düzelten tüm kodlar $\mathbb{F}_2$ ($\mathbb{Z}_2$) sonlu cismi üzerinde yapılmıştır.

1.3 Tezin Amacı

Bu çalışmanın nihai amacı lineer hücresel dönüşümlerin geçiş matrisleri ve bilinen sınır şartlarını kullanarak daha verimli bir kodlama ve dekodlama metodu geliştirmektir. Başta hedeflediğimiz hücresel dönüşümlerin bazı temel özelliklerini kullanarak (mesela belli bir konfigürasyondaki her bir hücrenin kendisinden önceki konfigürasyonların hücrelerinin durumlarına bağlı olması gibi) bilinen klasik hata düzeltme mantığından farklı bir mantık kurgulamaktı. Bu konuda bir sonuca ulaşmamakla beraber umudumuzu canlı tutmaktayız.

Bu çalışmada, hücresel dönüşümlerle yapılan ve 0.5 bilgi oranına sahip bit hata düzelten kodlar ile bayt hata düzelten kodların teorisi, ilkel (primitive) sonlu cisimlere genellenerek özgün örneklerle desteklenmiştir. Ayrıca hücresel dönüşümlerle bit hata düzelten kodların minimum Hamming uzaklığı için belli kısıtlamalar altında alt ve üst sınır tayin edilmiştir. Bunun yanı sıra maksimum grup uzunluğu olmaksızın hücresel dönüşümlerle bir-bayt hata düzeltebilen kod için kodlama ve dekodlamanın yapılabildiği örneklerle gösterilmiştir.

1.4 Hipotez

Hücresel dönüşümlerle hata düzelten kodlar, hücresel dönüşümlerin paralel ve düzenli yapılarından dolayı daha basit ve ucuz devre yapısına sahip olmaktadır [15]. Ayrıca hücresel dönüşümlerle hata düzelten kodlar klasik lineer kodlar ile karşılaştırıldığında dekodlama için daha az işlem gerektirmektedir [15].

BÖLÜM 2

HÜCRESEL DÖNÜŞÜMLER

Bu bölümde hücresel dönüşümler ile ilgili temel bilgiler verilecektir. Bu bölüm oluşturulurken; Cellular Automata: A Discrete View of the World, (2008) [7], Cellular Automata, A Discrete Universe, (2001) [1] ve Additive Cellular Automata: Theory and Applications, (1997) [15] isimli kitaplardan yararlanılmıştır.

2.1 Hücresel Dönüşümler

Genellikle karelerden oluşan ancak üçgen, beşgen, altıgen vb. gibi başka geometrik şekillerden de oluşabilen bitişik hücrelerin sonsuz bir dizisini (latis ağını) düşünelim. Bu hücre dizisini L_d ile gösterelim. Burada d hücre dizisinin boyutunu göstermektedir.

$\ldots$	x_{i-1}^t	x_i^t	x_{i+1}^t	$\ldots$

Şekil 2. 1 Bir boyutlu hücresel dönüşüm için latis kesiti

Örneğin; $d=1$ (Şekil 2.1) iken reel eksen boyunca yan yana dizilmiş tek biçimli geometrik şekillerin bir dizisi, $d=2$ (Şekil 2.2) iken düzleme yayılmış tek biçimli geometrik şekillerin iki boyutlu bir dizisini düşünebiliriz.

...	.	.	.	...
...	$x^t_{(i-1,j-1)}$	$x^t_{(i-1,j)}$	$x^t_{(i-1,j+1)}$	...
...	$x^t_{(i,j-1)}$	$x^t_{(i,j)}$	$x^t_{(i,j+1)}$	...
...	$x^t_{(i+1,j-1)}$	$x^t_{(i+1,j)}$	$x^t_{(i+1,j+1)}$	...
...	.	.	.	...

Şekil 2. 2 İki boyutlu hücresel dönüşüm için latis kesiti

Her bir hücre $p \geq 2$ olmak üzere, p farklı duruma (state) sahip olsun. Burada $p-$tane durumdan her birinin geriye kalan diğer $p-1$ durumdan farklı olduğunu belirtmemiz gerekir. Yani D mümkün bütün durumların kümesi olmak üzere; $\mathbb{Z}_p = \{0,1,2,...,p-1\} = D$ şeklinde yazılabilir. Bu kümedeki her bir eleman ile renkler, doğal süreçler veya birbirinden kesinlikle ayırt edilebilen olgular arasında bir birebir eşleşme yapılabilir. Örneğin; $\mathbb{Z}_2 = \{0,1\}$ kümesindeki mümkün iki durum 0 ve 1 dir. $0 =$ Beyaz (veya ölü), $1 =$ Siyah (veya canlı) şeklinde bir eşleşme yapılabilir. Her hücre doğal sayılar ile eşleşen ayrık zaman adımlarında durumunu güncelleyebilsin.

Tanım 2.1 Bir hücresel dönüşümde her zaman adımı (Bir ve iki boyut için sırasıyla Şekil 2.1 Şekil, 2.2'deki gibi.) aynı türden geometrik şekillerin bir dizisinden oluşur. Bu dizilerin her birine o zaman adımındaki *konfigürasyon* denir.

İki boyutlu hücresel dönüşümler için her konfigürasyon (Şekil 2.2) düzleme yayılmış tek biçimli geometrik şekillerin iki boyutlu bir dizisi olarak düşünülebilir ve her güncellenmeden sonra elde edilen yeni konfigürasyon da yine iki boyutlu bir dizidir.

…	x_{i-1}^{0}	x_{i}^{0}	x_{i+1}^{0}	…	$t=0$
…	x_{i-1}^{1}	x_{i}^{1}	x_{i+1}^{1}	…	$t=1$
		. . .			
…	x_{i-1}^{n}	x_{i}^{n}	x_{i+1}^{n}	…	$t=n$

Şekil 2. 3 Bir boyutlu hücresel dönüşüm için n adımlık evirilme

Şekil 2.3'de bulunan x_i^t ifadesinde i bulunulan hücrenin indisini, t ise bulunulan zaman adımını göstermektedir. Burada zamanın hücre durumları gibi ayrık olduğunu bir daha belirtelim. Genellikle $t=0$ anı başlangıç konfigürasyonu olarak alınır. Her zaman adımındaki her bir hücre durumunu, mevcut durumu ve belli sayıdaki komşularının durumlarına bağlı olarak yeniden belirler. Yani her bir hücre $t+1$ zaman adımında durumunu yeniden belirler. Bu durum güncellenmesine *evirilme* denir. Belli bir zaman adımındaki hücreler durumlarını eşzamanlı (synchronous) ya da eşzamanlı olmadan (asynchronous) evirebilirler. Ancak bu çalışmada evirilme eşzamanlı olarak düşünülecektir.

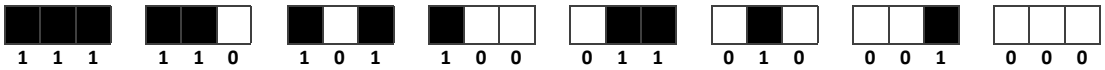
2.2 Bir Boyutlu Hücresel Dönüşümler İçin Bazı Tanımlar

Tanım 2.2 Bir hücre durumunu başka bir duruma evirirken çevresindeki eşit sayıda hücreden etkilenir. Etkileşim halinde bulunulan bu hücrelere *komşu*, sağdaki veya soldaki hücrelerin sayısına da *komşuluk yarıçapı* denir. Komşuluk yarıçapını r ile göstereceğiz.

r, komşuluk yarıçapı olmak üzere; bir boyutlu hücresel dönüşümler için bir komşulukta toplam $2r+1$ tane hücre vardır. Bu durumda p bir hücrenin bulunabileceği mümkün durumların sayısı iken toplam p^{2r+1} tane mümkün komşuluk durumu vardır. Mümkün komşuluk durumlarını N ile gösterirsek; $N=\mathbb{Z}_p^{2r+1}$ olur. O halde N kümesinin eleman sayısı p^{2r+1} dir. Örneğin; mümkün durumların sayısı $p=2$ ve komşuluk yarıçapı $r=1$ olarak alınırsa toplam $2^{(2\times1)+1}=8$ tane mümkün komşuluk durumu vardır. Bu durumlar;

$$N = \{\{1, 1, 1\}, \{1, 1, 0\}, \{1, 0, 1\}, \{1, 0, 0\}, \{0, 1, 1\}, \{0, 1, 0\}, \{0, 0, 1\}, \{0, 0, 0\}\}$$

şeklindedir. Bu sekiz komşuluk durumunu aşağıdaki gibi gösterebiliriz.



Şekil 2. 4 Bir boyutlu hücresel dönüşümler için $r = 1$ iken mümkün komşuluk durumları

Her bir hücre durumunu bir nesil sonraki durumuna dönüştürürken bu geçişi belirleyen yerel geçiş fonksiyonu *(local transition function)* adını verdiğimiz bir kural kullanır. Bu fonksiyonu (kuralı) genel olarak aşağıdaki gibi tanımlayabiliriz.

Tanım 2.3 $p \geq 2$ mümkün durumlar, r komşuluk yarıçapı ve $t = 0,1,2,...,n$ ayrık zaman adımları olmak üzere; $\mathbb{Z}_p = \{0,1,2,..., p-1\}$ olsun.

$$f : \mathbb{Z}_p^{2r+1} \to \mathbb{Z}_p,\ x_i^{t+1} := f\left(x_{i-r}^t, x_{i-r+1}^t, ..., x_{i-1}^t, x_i^t, x_{i+1}^t, ..., x_{i+r}^t\right) \tag{2.1}$$

şeklinde tanımlanan fonksiyona bir boyutlu hücresel dönüşümler için *yerel geçiş fonksiyonu* denir. Bu tanım d -boyuta genelleştirilebilir.

Geçiş fonksiyonu *belirleyici (deterministic)* ya da *olasılıklı (probabilistic)* olabilir. Ancak bu çalışmada belirleyici *(detrministic)* geçiş fonksiyonlarını tercih edeceğiz. Bu fonksiyonların sayısı sonludur ve $p \geq 2$ mümkün durumlar, r komşuluk yarıçapı olmak üzere; toplam $p^{p^{2r+1}}$ tanedir.

Tanım 2.4 $p \geq 2$ mümkün durumlar, r komşuluk yarıçapı ve t ayırık zaman adımları olmak üzere; $\mathbb{Z}_p = \{0,1,2,..., p-1\}$ olsun. $f : \mathbb{Z}_p^{2r+1} \to \mathbb{Z}_p$, $a_i \in \mathbb{Z}_p, \left(i = 1,2,...,2r+1\right)$ için

$$\begin{aligned} x_i^{t+1} &= f\left(x_{i-r}^t, x_{i-r+1}^t, ..., x_{i-1}^t, x_i^t, x_{i+1}^t, ..., x_{i+r}^t\right) \\ &= a_1 x_{i-r}^t + a_2 x_{i-r+1}^t + ... + a_r x_{i-1}^t + a_{r+1} x_i^t + a_{r+2} x_{i+1}^t + ... + a_{2r+1} x_{i+r}^t \left(\mathrm{mod}\ p\right) \end{aligned} \tag{2.2}$$

fonksiyonuyla belirli kurallara lineer kural denir. Bir boyutta toplam p^{2r+1} tane lineer kural vardır.

Aşağıda ard arda vereceğimiz üç tanım tüm hücresel dönüşümler için geçerlidir.

Tanım 2.5 Bir hücresel dönüşümde t-zaman adımındaki bir konfigürasyonu $m \in \mathbb{N}$ olmak üzere, $t+m$-zaman adımındaki başka bir konfigürasyona dönüştüren fonksiyona *global geçiş fonksiyonu* denir ve $F = \langle f_1, f_2, ..., f_n \rangle$ ile gösterilir. Burada $n \in \mathbb{Z}^+$ başlangıç dizisindeki hücrelerin sayısı ve $f_i, (i = 1,2,...,n)$'ler de yerel geçiş fonksiyonlarıdır.

Matematik diliyle ifade edecek olursak: C tüm konfigürasyonların kümesi olmak üzere; $F : C \to C$, her $c \in C$ için $F(c) = c'$, $(c' \in C)$ olarak ifade edilebilir.

Tanım 2.6 Eğer bir başlangıç dizisindeki (konfigürasyonundaki) tüm hücreler aynı yerel geçiş fonksiyonunu kullanarak bir sonraki zaman adımında durumlarını belirliyorlarsa ya da diğer bir ifadeyle global geçiş fonksiyonu tamamen aynı yerel geçiş fonksiyonlarından oluşuyorsa bu hücresel dönüşüme *düzenli (uniform)* hücresel dönüşüm denir.

Tanım 2.7 Eğer bir başlangıç dizisindeki (konfigürasyonundaki) hücreler bir sonraki zaman adımında durumlarını belirlerken farklı yerel geçiş fonksiyonları kullanıyorlarsa ya da global geçiş fonksiyonunda bulunan yerel geçiş fonksiyonlarından en az bir tanesi diğerlerinden farklıysa bu hücresel dönüşüme *melez (hybrid)* hücresel dönüşüm denir.

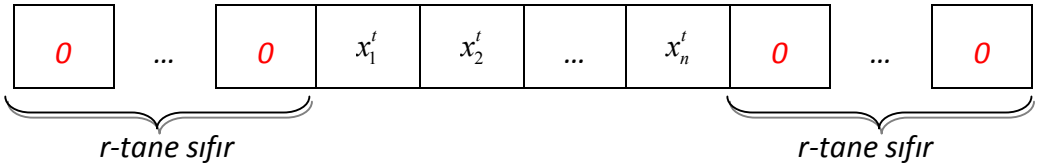
Not: Yukarıda yaptığımız tanımlamalardan sonra d boyutlu bir hücresel dönüşüm; L_d hücrelerin d-boyutlu bir dizisi (latisi), D mümkün durumların kümesi, N mümkün komşuluk durumları ve f yerel geçiş fonksiyonu olmak üzere; $CA = [L_d, D, N, f]$ şeklinde bir dörtlü ile ifade edilebilir.

2.2.1 Bir Boyutlu Hücresel Dönüşümler İçin Sınır Şartları

Bir hücresel dönüşümde verilen bir başlangıç dizisinin (konfigürasyonunun) durumunu bir başka duruma evirebilmesi için verilen her hücrenin komşularının belirli olması gerekir. Bu sebeple herhangi bir boyutta evirilmenin olabilmesi için başlangıç dizisi sonlu olmalıdır. Bu durumda verilen bir başlangıç dizisinde sınırda bulunan hücrelerin komşuları tanımlanmalıdır. İlk olarak bir boyutlu hücresel dönüşümler için sınır şartlarını *(boundary conditions)* tanımlayacağız.

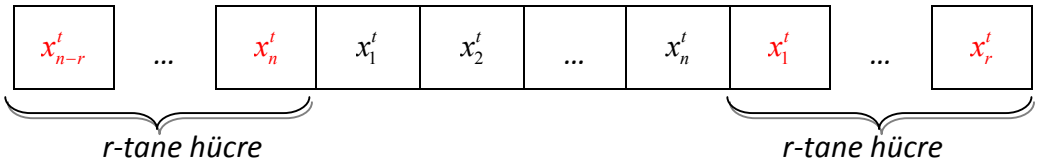
Tanım 2.8 $r < n$ komşuluk yarıçapı olmak üzere, $C^t = \left[x_1^t, x_2^t, ..., x_n^t \right]$ t-zaman adımındaki konfigürasyon olsun.

i) Eğer her t-zaman adımındaki dizinin sol baştaki teriminin soluna r-tane sıfır ve sağ baştaki teriminin yanına da r-tane sıfır eklenerek sırasıyla sol ve sağ baştaki hücrelerin sola doğru ve sağa doğru komşulukları belirleniyorsa bu sınır şartına *sıfır sınır şartı (null boundary condition)* denir.



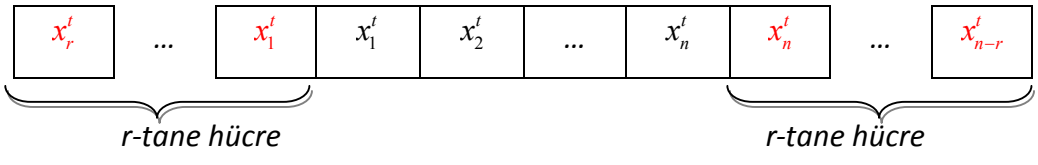
Şekil 2. 5 Bir boyutlu hücresel dönüşümler için sıfır sınır şartı

ii) Eğer her t-zaman adımındaki dizinin sol baştaki terimi ile sağ baştaki terimi bitişikmiş gibi düşünülerek dizinin uç kısmında yer alan hücrelerin r-tane komşusu belirleniyorsa bu sınır şartına *periyodik sınır şartı (periodic boundary condition)* denir.



Şekil 2. 6 Bir boyutlu hücresel dönüşümler için periyodik sınır şartı

iii) Eğer her t-zaman adımındaki dizinin sol uçtaki hücre değeri sola doğru komşuluk ve sağ uçtaki hücre değeri sağa doğru komşuluk için tekrar ediliyorsa bu sınır şartına *yansımalı sınır şartı (reflective boundary condition)* denir.



Şekil 2. 7 Bir boyutlu hücresel dönüşümler için yansımalı sınır şartı

2.2.2 Bir Boyutlu Temel Hücresel Dönüşümler (Wolfram Kuralları)

Bir boyutlu hücresel dönüşümlerde $p=2$ ve $r=1$ için toplam $p^{p^{2r+1}}=2^{2^3}=256$ tane bir boyutlu hücresel dönüşüm (mümkün yerel geçiş fonksiyonu) vardır. Bu 256 kural (hücresel dönüşüm) Stephen Wolfram tarafından *temel hücresel dönüşüm (elementary cellular automata)* olarak adlandırılmıştır. Bunlar ilk sıralı hücresel dönüşüm örnekleridir. Bu kuralların 8-tanesi lineer kurallardır. Şöyle ki; her $a,b,c\in\mathbb{Z}_2$ için $f:\mathbb{Z}_2^3\to\mathbb{Z}_2$ olmak üzere,

$$f\left(x_{i-1}^t,x_i^t,x_{i+1}^t\right)=x_i^{t+1}=ax_{i-1}^t+bx_i^t+cx_{i+1}^t\ (\mathrm{mod}\,2) \tag{2.3}$$

kuralı ile belirli mümkün sekiz tane yerel geçiş fonksiyonu vardır. Bu kurallar Tablo 2.1'de gösterilmiştir.

Kural Numarası	Geçiş Fonksiyonu
0. Kural	$x_i^{t+1}=0\ (a=b=c=0)$
60. Kural	$x_i^{t+1}=x_{i-1}^t+x_i^t(\mathrm{mod}\,2)$
90. Kural	$x_i^{t+1}=x_{i-1}^t+x_{i+1}^t(\mathrm{mod}\,2)$
102. Kural	$x_i^{t+1}=x_i^t+x_{i+1}^t(\mathrm{mod}\,2)$
150. Kural	$x_i^{t+1}=x_{i-1}^t+x_i^t+x_{i+1}^t(\mathrm{mod}\,2)$
170. Kural	$x_i^{t+1}=x_{i+1}^t$
204. Kural	$x_i^{t+1}=x_i^t$
240. Kural	$x_i^{t+1}=x_{i-1}^t$

Tablo 2. 1 Bir boyutlu temel lineer hücresel dönüşümler

Örnek 2.1 Eğer $a=b=c=1\in\mathbb{Z}_2$ olarak seçilirse (2.3)'deki fonksiyon $f\left(x_{i-1}^t,x_i^t,x_{i+1}^t\right)=x_i^{t+1}=x_{i-1}^t+x_i^t+x_{i+1}^t\ (\mathrm{mod}\,2)$ halini alır. Komşuluk yarıçapı $r=1$ olduğundan $p^{2r+1}=2^3=8$ tane mümkün komşuluk durumu vardır. Her bir komşuluk durumunun f altındaki görüntüsü Tablo 2.2'de gösterilmiştir.

7	6	5	4	3	2	1	0
111	110	101	100	011	010	001	000
1	0	0	1	0	1	1	0

Tablo 2. 2 Bir boyutlu temel lineer hücresel dönüşümler için $r=1$ iken mümkün sekiz komşuluk durumu ve her bir komşuluk durumunun f altındaki görüntüsü

Tablo 2.2'nin birinci satırı, ikinci satırda yer alan komşuluk durumunun onluk tabandaki karşılığını ve üçüncü satırı ise yine ikinci satırdaki her bir komşuluk durumunun f altındaki görüntüsünü içermektedir. Üçüncü satırda yer alan diziye f kuralına karşılık gelen *kural vektörü* denir. Bu kural vektörünün onluk tabandaki karşılığı bize kural numarasını verir. Tablo 2.2'de son satırın on tabanında 150 sayısına karşılık geldiğine dikkat ediniz.

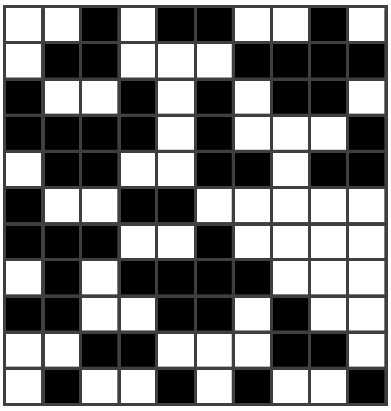
$n=10$ uzunluğunda $C^0=\left[0010110010\right]=\left[x_1^0x_2^0x_3^0x_4^0x_5^0x_6^0x_7^0x_8^0x_9^0x_{10}^0\right]$, $\left(t=0\right)$ başlangıç konfigürasyonunu sıfır sınır şartı altında $t=1,2,...,10$ adımı için evirelim. Sıfır sınır şartı kullanıldığından ve komşuluk yarıçapı $r=1$ olduğundan $t=0$ için mümkün komşuluk durumları aşağıdaki gibidir:

$0x_1^0x_2^0=000$, $x_1^0x_2^0x_3^0=001$, $x_2^0x_3^0x_4^0=010$, $x_3^0x_4^0x_5^0=101$, $x_4^0x_5^0x_6^0=011$, $x_5^0x_6^0x_7^0=110$, $x_6^0x_7^0x_8^0=100$, $x_7^0x_8^0x_9^0=001$, $x_8^0x_9^0x_{10}^0=010$ ve $x_9^0x_{10}^00=100$.

Bu komşuluklardan her biri $t=1$ anında Tablo 2.2'den de yararlanılarak;

$x_1^1=0$, $x_2^1=1$, $x_3^1=1$, $x_4^1=0$, $x_5^1=0$, $x_6^1=0$, $x_7^1=1$, $x_8^1=1$, $x_9^1=1$, $x_{10}^1=1$

olarak elde edilir. Bu şekilde $t=2,3,...,10$ için de $x_i^t,\left(i=1,2,...,10\right)$ lerin durumlarını belirlediğimizde Şekil 2.8 (a)'daki desen (ya da uzay zaman grafiği) ile Şekil 2.8 (b)'deki durum tablosu elde edilir.



t=0	0	0	1	0	1	1	0	0	1	0
t=1	0	1	1	0	0	0	1	1	1	1
t=2	1	0	0	1	0	1	0	1	1	0
t=3	1	1	1	1	0	1	0	0	0	1
t=4	0	1	1	0	0	1	1	0	1	1
t=5	1	0	0	1	1	0	0	0	0	0
t=6	1	1	1	0	0	1	0	0	0	0
t=7	0	1	0	1	1	1	1	0	0	0
t=8	1	1	0	0	1	1	0	1	0	0
t=9	0	0	1	1	0	0	0	1	1	0
t=10	0	1	0	0	1	0	1	0	0	1

(a) (b)

□ → 0, ■ → 1

Şekil 2. 8 $C^0 = [00101\,10010]$ başlangıç konfigürasyonunun 150. temel kurala göre uzay-zaman grafiği ve durum tablosu

Bir boyutlu hücresel dönüşümlerde $p = 3$ ve $\mathrm{r} = 1$ için toplam $p^{p^{2r+1}} = 3^{3^3}$ $= 7625597484987$ tane bir boyutlu hücresel dönüşüm (mümkün yerel geçiş fonksiyonu) vardır.

Örnek 2.2 Eğer $a = b = c = 1 \in \mathbb{Z}_3$ olarak seçilirse (2.3)'teki fonksiyon $f\left(x_{i-1}^t, x_i^t, x_{i+1}^t\right) = x_i^{t+1} = x_{i-1}^t + x_i^t + x_{i+1}^t \left(\mathrm{mod}\, 3\right)$ halini alır. Komşuluk yarıçapı $r = 1$ olduğundan $p^{2r+1} = 3^3 = 27$ tane mümkün komşuluk durumu vardır. Bu komşuluk durumları aşağıdaki gibidir:

$$N = \{\{2,\ 2,\ 2\},\{2,\ 2,\ 1\},\{2,\ 2,\ 0\},\{2,\ 1,\ 2\},\{2,\ 1,\ 1\},\{2,\ 1,\ 0\},$$
$$\{2,\ 0,\ 2\},\{2,\ 0,\ 1\},\{2,\ 0,\ 0\},\{1,\ 2,\ 2\},\{1,\ 2,\ 1\},\{1,\ 2,\ 0\},\{1,\ 1,\ 2\},$$
$$\{1,\ 1,\ 1\},\{1,\ 1,\ 0\},\{1,\ 0,\ 2\},\{1,\ 0,\ 1\},\{1,\ 0,\ 0\},\ \{0,\ 2,\ 2\},\{0,\ 2,\ 1\},$$
$$\{0,\ 2,\ 0\},\{0,\ 1,\ 2\},\ \{0,\ 1,\ 1\},\{0,\ 1,\ 0\},\{0,\ 0,\ 2\},\{0,0,\ 1\},\{0,\ 0,\ 0\}\}.$$

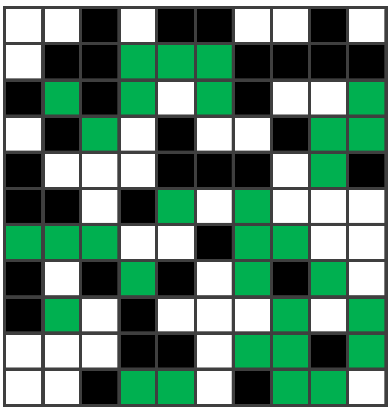
Buradaki her bir komşuluğun f fonksiyonu altındaki görüntüsü alınarak $021210102210102021102021210$ kural dizisi elde edilir. Bu kural dizisi onluk tabanda 2201243116917 sayısına karşılık gelir. Bu kural $p^{p^{2r+1}} = 3^{3^3} = 7625597484987$ tane kuraldan yalnızca bir tanesidir. Bu kurala göre $n = 10$ uzunluğunda $C^0 = [00101\,10010] = \left[x_1^0 x_2^0 x_3^0 x_4^0 x_5^0 x_6^0 x_7^0 x_8^0 x_9^0 x_{10}^0\right]$, $\left(t = 0\right)$ başlangıç konfigürasyonunu sıfır sınır şartı altında $t = 1, 2, \ldots, 10$ adım için evirelim. Sıfır sınır şartı kullanıldığından ve komşuluk yarıçapı $r = 1$ olduğundan $t = 0$ için mümkün komşuluk durumları;

$0x_1^0x_2^0 = 000$, $x_1^0x_2^0x_3^0 = 001$, $x_2^0x_3^0x_4^0 = 010$, $x_3^0x_4^0x_5^0 = 101$, $x_4^0x_5^0x_6^0 = 011$, $x_5^0x_6^0x_7^0 = 110$, $x_6^0x_7^0x_8^0 = 100$, $x_7^0x_8^0x_9^0 = 001$, $x_8^0x_9^0x_{10}^0 = 010$, $x_9^0x_{10}^0 0 = 100$

şeklindedir. Bu komşuluklardan her biri $t = 1$ anında 2201243116917 kuralına göre evrilirse aşağıdaki konfigürasyon $\left([0112221111]\right)$ elde edilir.

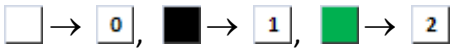
$x_1^1 = 0$, $x_2^1 = 1$, $x_3^1 = 1$, $x_4^1 = 2$, $x_5^1 = 2$, $x_6^1 = 2$, $x_7^1 = 1$, $x_8^1 = 1$, $x_9^1 = 1$, $x_{10}^1 = 1$.

Bu şekilde $t = 2,3,...,10$ için de x_i^t $(i = 1,2,...,10)$ lerin durumlarını belirlediğimizde Şekil 2.9 (a)'daki desen (ya da uzay zaman grafiği) ile Şekil 2.9 (b)'deki durum tablosu elde edilir.



t=0	0	0	1	0	1	1	0	0	1	0
t=1	0	1	1	2	2	2	1	1	1	1
t=2	1	2	1	2	0	2	1	0	0	2
t=3	0	1	2	0	1	0	0	1	2	2
t=4	1	0	0	0	1	1	1	0	2	1
t=5	1	1	0	1	2	0	2	0	0	0
t=6	2	2	2	0	0	1	2	2	0	0
t=7	1	0	1	2	1	0	2	1	2	0
t=8	1	2	0	1	0	0	0	2	0	2
t=9	0	0	0	1	1	0	2	2	1	2
t=10	0	0	1	2	2	0	1	2	2	0

(a) (b)



Şekil 2. 9 $C^0 = [00101100010]$ başlangıç konfigürasyonunun 2201243116917. kurala göre uzay-zaman grafiği ve durum tablosu

2.3 İki Boyutlu Hücresel Dönüşümler İçin Komşuluk ve Sınır Şartları

İki boyutlu hücresel dönüşümler için sınır şartlarını vermeden önce iki boyutlu hücresel dönüşümler için iyi bilinen iki komşuluğu verelim. Bu iki komşuluk aşağıda Şekil 2.10 ve Şekil 2.11'de verilmiştir.

(a) (b)

Şekil 2. 10 İki boyutlu hücresel dönüşümler için $r=1$ iken (a) Moore komşuluğu, (b) Neumann komşuluğu

(a) (b)

Şekil 2. 11 İki boyutlu hücresel dönüşümler için $r=2$ iken (a) Moore komşuluğu, (b) Neumann komşuluğu

Yukarıdaki şekillerde (Şekil 2.10, Şekil 2.11) açık gri renk ile verilen hücre merkez hücreyi koyu gri renk ile verilen hücreler ise komşu hücreleri göstermektedir. Bu iki komşuluk dışında başka komşuluklar da tanımlanabilir.

Daha önce verdiğimiz sınır şartları $d>1$ olmak üzere, d-boyutlu hücresel dönüşümlere uygulanabilir. Bu sınır şartlarını $r=1$ komşuluk yarıçapı için iki boyutlu $(d=2)$ hücresel dönüşümlere uygulayalım:

i) Sıfır (Null) Sınır şartı:

0	0	0	0	0
0	$x^t_{i-1,j-1}$	$x^t_{i-1,j}$	$x^t_{i-1,j+1}$	0
0	$x^t_{i,j-1}$	$x^t_{i,j}$	$x^t_{i,j+1}$	0
0	$x^t_{i+1,j-1}$	$x^t_{i+1,j}$	$x^t_{i+1,j+1}$	0
0	0	0	0	0

Şekil 2. 12 İki boyutlu hücresel dönüşümler için sıfır (null) sınır şartı

ii) Periyodik Sınır Şartı:

$x^t_{i+1,j+1}$	$x^t_{i+1,j-1}$	$x^t_{i+1,j}$	$x^t_{i+1,j+1}$	$x^t_{i+1,j-1}$
$x^t_{i-1,j+1}$	$x^t_{i-1,j-1}$	$x^t_{i-1,j}$	$x^t_{i-1,j+1}$	$x^t_{i-1,j-1}$
$x^t_{i,j+1}$	$x^t_{i,j-1}$	$x^t_{i,j}$	$x^t_{i,j+1}$	$x^t_{i,j-1}$
$x^t_{i+1,j+1}$	$x^t_{i+1,j-1}$	$x^t_{i+1,j}$	$x^t_{i+1,j+1}$	$x^t_{i+1,j-1}$
$x^t_{i-1,j+1}$	$x^t_{i-1,j-1}$	$x^t_{i-1,j}$	$x^t_{i-1,j+1}$	$x^t_{i-1,j-1}$

Şekil 2. 13 İki boyutlu hücresel dönüşümler için periyodik (periodic) sınır şartı

iii) Yansımalı (Reflective) Sınır Şartı:

$x^t_{i-1,j-1}$	$x^t_{i-1,j-1}$	$x^t_{i-1,j}$	$x^t_{i-1,j+1}$	$x^t_{i-1,j+1}$
$x^t_{i-1,j-1}$	$x^t_{i-1,j-1}$	$x^t_{i-1,j}$	$x^t_{i-1,j+1}$	$x^t_{i-1,j+1}$
$x^t_{i,j-1}$	$x^t_{i,j-1}$	$x^t_{i,j}$	$x^t_{i,j+1}$	$x^t_{i,j+1}$
$x^t_{i+1,j-1}$	$x^t_{i+1,j-1}$	$x^t_{i+1,j}$	$x^t_{i+1,j+1}$	$x^t_{i+1,j+1}$
$x^t_{i+1,j-1}$	$x^t_{i+1,j-1}$	$x^t_{i+1,j}$	$x^t_{i+1,j+1}$	$x^t_{i+1,j+1}$

Şekil 2. 14 İki boyutlu hücresel dönüşümler için yansımalı (reflective) sınır şartı

2.4 Bir Boyutlu Lineer Hücresel Dönüşümlerin Matris Temsili

Genel olarak r komşuluk yarıçapı ve p mümkün durumların sayısı için bir boyutlu mümkün lineer kurallar $a_i \in \mathbb{Z}_p$ olmak üzere;

$$\begin{aligned} x_i^{t+1} &= f\left(x_{i-r}^t, x_{i-r+1}^t, \ldots, x_{i-1}^t, x_i^t, x_{i+1}^t, \ldots, x_{i+r}^t\right) \\ &= a_1 x_{i-r}^t + a_2 x_{i-r+1}^t + \ldots + a_r x_{i-1}^t + a_{r+1} x_i^t + a_{r+2} x_{i+1}^t + \ldots + a_{2r+1} x_{i+r}^t \left(\mathrm{mod}\ p\right) \end{aligned} \tag{2.4}$$

ile verilebilir. $F = \langle f_1, f_2, \ldots, f_n \rangle$, *düzenli* (uniform) bir global geçiş fonksiyonu olsun. Bu durumda periyodik sınır şartı altında F aşağıdaki gibi bir T matrisi ile temsil edilebilir.

$$T = \begin{pmatrix} a_i & a_{i+1} & \cdots & a_{i+r} & 0 & \cdots & 0 & a_{i-r} & \cdots & a_{i-3} & a_{i-2} & a_{i-1} \\ a_{i-1} & a_i & a_{i+1} & \cdots & a_{i+r} & 0 & \cdots & 0 & a_{i-r} & \cdots & a_{i-3} & a_{i-2} \\ a_{i-2} & a_{i-1} & a_i & a_{i+1} & \cdots & a_{i+r} & 0 & \cdots & 0 & a_{i-r} & \cdots & a_{i-3} \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{i+3} & \cdots & a_{i+r} & 0 & \cdots & 0 & a_{i-r} & \cdots & a_{i-1} & a_i & a_{i+1} & a_{i+2} \\ a_{i+2} & a_{i+3} & \cdots & a_{i+r} & 0 & \cdots & 0 & a_{i-r} & \cdots & a_{i-1} & a_i & a_{i+1} \\ a_{i+1} & a_{i+2} & a_{i+3} & \cdots & a_{i+r} & 0 & \cdots & 0 & a_{i-r} & \cdots & a_{i-1} & a_i \end{pmatrix}$$

Eğer F *düzenli* (uniform) bir kural vektörü ve $n = k \times s$ şeklinde bir bileşik sayı ise T matrisini blok matrisler cinsinden ifade edebiliriz.

$$A=\begin{pmatrix} a_i & a_{i+1} & \cdots & a_{i+r-1} & a_{i+r} \\ a_{i-1} & a_i & \cdots & a_{i+r-2} & a_{i+r-1} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ a_{i-r+1} & a_{i-r+2} & \cdots & a_i & a_{i+1} \\ a_{i-r} & a_{i-r+1} & \cdots & a_{i-1} & a_i \end{pmatrix}, \quad B=\begin{pmatrix} a_{i-r} & a_{i-r+1} & \cdots & a_{i-2} & a_{i-1} \\ 0 & a_{i-r} & \cdots & a_{i-3} & a_{i-2} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & a_{i-r} & a_{i-r+1} \\ 0 & 0 & \cdots & 0 & a_{i-r} \end{pmatrix}$$

$$C=\begin{pmatrix} a_{i+r} & 0 & \cdots & 0 & 0 \\ a_{i-1} & a_{i+r} & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ a_{i+2} & a_{i+3} & \cdots & a_{i+r} & 0 \\ a_{i+1} & a_{i+2} & \cdots & a_{i+r-1} & a_{i+r} \end{pmatrix}, \quad O=\begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \end{pmatrix}$$

$k\times k,\ (k=2r+1)$ tipinde karesel matrisler olmak üzere;

$$T=\begin{pmatrix} A & O & \cdots & O & B \\ O & A & \cdots & O & O \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ O & O & \cdots & A & O \\ C & O & \cdots & O & A \end{pmatrix}$$

şeklinde kısaca yazılabilir. Eğer bu gösterimdeki B ve C matrislerinin tüm bileşenleri sıfır olarak alınırsa sıfır sınır şartı altında matris temsili yapılmış olur.

Ayrıca $X^t=[x_1,x_2,...,x_n]$ bir t. zaman adımındaki konfigürasyon olsun. Bu durumda $X^{t+m}=T^m\left[X^t\right],\ (m\in\mathbb{Z}^+)$ şeklinde matris yardımıyla evirilme yapılabilir.

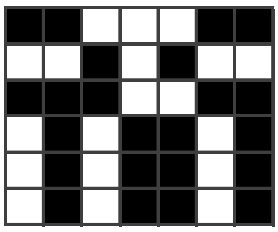
Örnek 2.3 $F=\langle 150,150,90,90,150\rangle$ melez (hybrid) global geçiş fonksiyonu 90. ve 150. Wolfram kurallarından oluşsun. Tablo 2.1'den

90. kural $\rightarrow x_i^{t+1}=x_{i-1}^t+x_{i+1}^t(\mathrm{mod}\,2)$ ve 150. kural $\rightarrow x_i^{t+1}=x_{i-1}^t+x_i^t+x_{i+1}^t(\mathrm{mod}\,2)$ idi. Şimdi F global geçiş fonksiyonuna (kural vektörüne) karşılık gelen geçiş matrisini sırasıyla periyodik ve sıfır sınır şartı altında yazalım.

$$T=\begin{pmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{pmatrix}\begin{matrix}\rightarrow 150\\ \rightarrow 150\\ \rightarrow\ 90\\ \rightarrow\ 90\\ \rightarrow 150\end{matrix}, \quad T=\begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}\begin{matrix}\rightarrow 150\\ \rightarrow 150\\ \rightarrow\ 90\\ \rightarrow\ 90\\ \rightarrow 150\end{matrix}$$

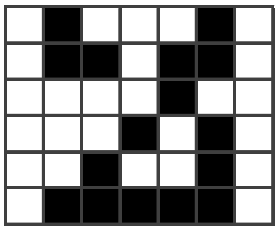
$X^0 = \left[x_1^0, x_2^0, x_3^0, x_4^0, x_5^0 \right] = [10001]$ başlangıç konfigürasyonu verilsin. X^0 başlangıç konfigürasyonunu $t = 1,2,3,4,5$ zaman adımları için sırasıyla periyodik ve sıfır sınır şartı altında evirelim. Daha sonra $t = 5$ için bu evirilmeyi matris yardımıyla yapalım.

1	1	0	0	0	1	1	t=0
0	0	1	0	1	0	0	t=1
1	1	1	0	0	1	1	t=2
0	1	0	1	1	0	1	t=3
0	1	0	1	1	0	1	t=4
0	1	0	1	1	0	1	t=5



Şekil 2. 15 $X^0 = [10001]$ başlangıç konfigürasyonunun $F = \langle 150,150,90,90,150 \rangle$ kural vektörüne göre periyodik sınır şartı altında durum tablosu ve uzay zaman grafiği

0	1	0	0	0	1	0	t=0
0	1	1	0	1	1	0	t=1
0	0	0	0	1	0	0	t=2
0	0	0	1	0	1	0	t=3
0	0	1	0	0	1	0	t=4
0	1	1	1	1	1	0	t=5



Şekil 2. 16 $X^0 = [10001]$ başlangıç konfigürasyonunun $F = \langle 150,150,90,90,150 \rangle$ kural vektörüne göre sıfır sınır şartı altında durum tablosu ve uzay zaman grafiği

Şimdi yukarıda (Şekil 2.15, Şekil 2.16) yaptığımız evirilmeyi hem periyodik hem de sıfır sınır şartı için geçiş matrisi yardımıyla gerçekleştirelim:

$$T = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{pmatrix} \Rightarrow T^5 = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$X^{t+m} = T^m \left[X^t \right], \left(m \in \mathbb{Z}^+ \right)$ olduğundan $X^{0+5} = T^5 \left[X^0 \right]$ yazılır. Değerler yerine yazılıp hesaplamalar yapılırsa;

$$T^5 = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \times \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} (\mathrm{mod}\, 2) \Rightarrow X^5 = [10110]$$

elde edilir. Aynı işlemler sıfır sınır şartı için de yapılırsa;

$$T=\begin{pmatrix}1&1&0&0&0\\1&1&1&0&0\\0&1&0&1&0\\0&0&1&0&1\\0&0&0&1&1\end{pmatrix}\Rightarrow T^5=\begin{pmatrix}0&0&1&1&1\\0&1&0&1&1\\1&0&1&1&0\\1&1&1&0&0\\1&1&0&0&0\end{pmatrix}$$

$$T^5=\begin{pmatrix}0&0&1&1&1\\0&1&0&1&1\\1&0&1&1&0\\1&1&1&0&0\\1&1&0&0&0\end{pmatrix}\times\begin{pmatrix}1\\0\\0\\0\\1\end{pmatrix}=\begin{pmatrix}1\\1\\1\\1\\1\end{pmatrix}(\mathrm{mod}\,2)\Rightarrow X^5=[11111]$$

elde edilir.

Bir boyutlu lineer Wolfram kurallarının matris temsilleri için A. K. Das vd. tarafından yazılan makaleye bakılabilir [27]. İki boyutlu lineer hücresel dönüşümlerin matris temsilleri de bir boyutlu lineer hücresel dönüşümlerin matris temsillerine benzemektedir. Bu konuda detaylı bilgi için [28], [29], [30], [31] referans numarası ile verilen makalelere başvurulabilir.

2.5 Hücresel Dönüşümlerin Grup Özellikleri

Tanım 2.9 Bir hücresel dönüşümün $k-$uzunluğundaki mümkün tüm konfigürasyonları bir takım döngüler içine giriyorsa bu tür hücresel dönüşüme *grup hücresel dönüşüm* denir [27].

Teorem 2.1 Bir lineer hücresel dönüşümün grup olabilmesi için gerek ve yeter şart temsili matrisinin determinantının sıfırdan farklı olmasıdır [27].

İspat: $(\Rightarrow)$ Verilen hücresel dönüşüm grup olsun. Bu durumda T temsili matris olmak üzere, $T^n=I$ (I birim matris) olacak şekilde bir $n\in\mathbb{Z}^+$ vardır. O halde $\det\left(T^n\right)=\det\left(T\right)^n=\det\left(I\right)\neq 0$ olur.

$(\Leftarrow)$ $\det\left(T\right)\neq 0$ olsun. T temsili matrisin kuvvetleri bir grup oluşturur. Bu grup birimi I olan ve tersinir karesel matrislerin oluşturduğu değişmeli olmayan grubun devirli dolayısıyla değişmeli bir alt grubudur. Bu durumda T'nin temsil ettiği hücresel

dönüşümün konfigürasyonları döngü ya da döngülerden oluşur. O halde bu hücresel dönüşüm gruptur.

Tanım 2.10 T, $k\times k$ tipinde temsili matris ve $D=\mathbb{Z}_p=\{0,1,2,...,p-1\}$ mümkün durumların kümesi olsun. Eğer $T_{k\times k}$ temsili matrisine sahip hücresel dönüşümün sıfır dışındaki $k-$uzunluklu tüm konfigürasyonları tek bir döngü içinde yer alıyorsa, yani $T^{p^k-1}=I$ oluyorsa bu hücresel dönüşüme *maksimal (devir) uzunluklu* hücresel dönüşüm denir. Aksi halde *maksimal uzunluklu olmayan* hücresel dönüşüm denir.

Teorem 2.2 Temsili matrisi T olan bir hücresel dönüşüm için $T^n=I$ olsun. T'nin $s-$uzunluğunda bir devir üretebilmesi için gerek ve yeter şart $\det\left(T^s-I\right)=0$ olmasıdır.

İspat: $(\Rightarrow)$ T'nin temsil ettiği hücresel dönüşüm $s-$uzunluğunda bir devir üretsin. Bu durumda bir $X\neq 0$ konfigürasyonu için $T^s[X]=[X]$ olur. Buradan $T^s[X]-[X]=0$ $\Rightarrow\left(T^s-I\right)[X]=0\ \Rightarrow\ \det\left(T^s-I\right)=0$ olarak bulunur.

$(\Leftarrow)$ Şimdi $\det\left(T^s-I\right)=0$ olacak şekilde bir $s\in\mathbb{Z}^+$ olsun. Bu durumda $\left(T^s-I\right)[X]=0\ \Rightarrow\ T^s[X]-[X]=0\ \Rightarrow\ T^s[X]=[X]$ olduğundan T, $s-$uzunluğunda bir devir üretir.

Lemma 2.1 Eğer T'nin grup mertebesi $\circ(T)=n$ şeklinde bileşik bir sayı ise bu durumda T'nin üreteceği devir uzunlukları ancak n'nin bir böleni olabilirler [27].

İspat: Bu ispatı çelişki bulma yöntemiyle yapacağız. $\circ(T)=n$ ve $s\nmid n,\ \left(s,n\in\mathbb{Z}^+\right)$ bir devir uzunluğu olsun. n ve s ye bölme algoritması uygulanırsa $n=qs+r,\ (r<s)$ olacak şekilde tek türlü belirli $q,r\in\mathbb{Z}^+$ vardır. Aynı zamanda s devir uzunluğu olduğundan bir $X\neq 0$ konfigürasyonu için $T^s[X]=[X]$ olacak şekilde en küçük s vardır. Buradan hareketle;

$$
\begin{aligned}
T^n[X] &= T^{qs+r}[X] \\
&= T^r T^{(q-1)s} T^s\ [X] \\
&= T^r T^{(q-1)s} [X] \\
&= T^r T^{(q-2)s} T^s\ [X] \\
&= T^r T^{(q-2)s}\ [X] \\
&\ \ \vdots \\
&= T^r T^s\ [X] \\
&= T^r [X] \\
\Rightarrow T^n[X] &= T^r [X]
\end{aligned}
$$

Bu durumda s'den daha küçük bir r devir uzunluğu bulunmuş olur. Ancak bu bir çelişkidir. Bu çelişki $s \nmid n$ olarak seçmemizden kaynaklandı. Yani $s \mid n$ dir.

Örnek 2.4 $F = \langle 90,150,90,150 \rangle$ kural vektörüne sıfır sınır şartı altında karşılık gelen temsili matris;

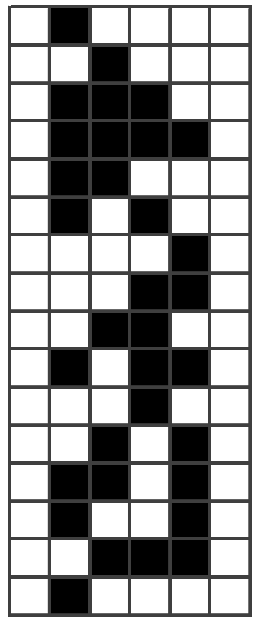
$$
T = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}
$$

şeklindedir. $k = 4$ uzunluğundaki tüm konfigürasyonların kümesi;

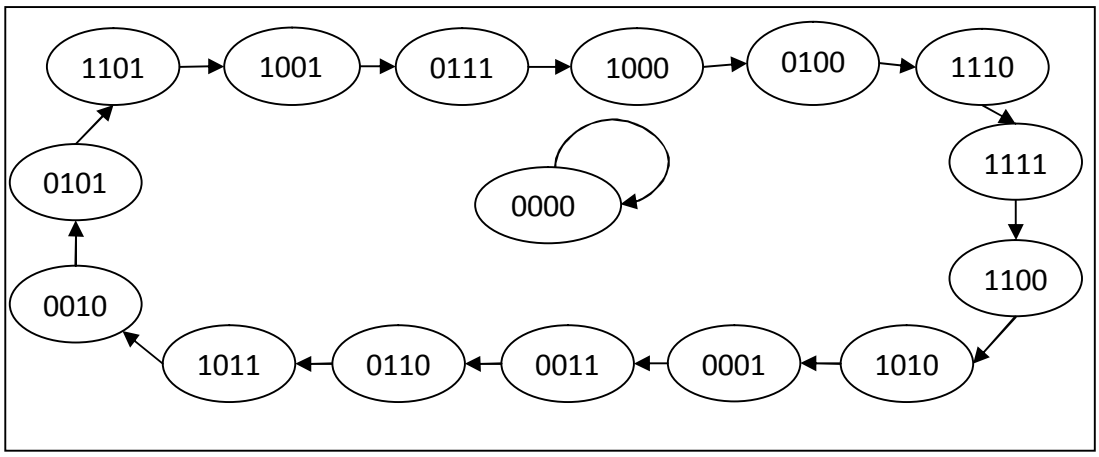
$$
\begin{aligned}
C = \{&0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, \\
&1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111\}
\end{aligned}
$$

şeklindedir. $X^0 = [1000]$ konfigürasyonu sıfır sınır şartı altında $F = \langle 90,150,90,150 \rangle$ global geçiş fonksiyonu kullanılarak $t = 1,2,..,15$ adım evirildikten sonra $k = 4$ uzunluğundaki mümkün tüm konfigürasyonlar gözlendikten sonra $t = 15$. adımda başlangıç konfigürasyonuna tekrar ulaşılır. Yani verilen kural maksimal uzunluklu bir hücresel dönüşümdür.

0	1	0	0	0	0	t=0
0	0	1	0	0	0	t=1
0	1	1	1	0	0	t=2
0	1	1	1	1	0	t=3
0	1	1	0	0	0	t=4
0	1	0	1	0	0	t=5
0	0	0	0	1	0	t=6
0	0	0	1	1	0	t=7
0	0	1	1	0	0	t=8
0	1	0	1	1	0	t=9
0	0	0	1	0	0	t=10
0	0	1	0	1	0	t=11
0	1	1	0	1	0	t=12
0	1	0	0	1	0	t=13
0	0	1	1	1	0	t=14
0	1	0	0	0	0	t=15



Şekil 2. 17 $F=\langle 90,150,90,150\rangle$ kural vektörüne karşılık gelen durum tablosu ve uzay-zaman grafiği



Şekil 2. 18 $F=\langle 90,150,90,150\rangle$ kural vektörüne karşılık gelen devir diyagramı

Örnek 2.5 $F=\langle 90,90,90,90\rangle$ kural vektörüne sıfır sınır şartı altında karşılık gelen temsili matris;

$$T=\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

şeklinde olur. $k=4$ uzunluğundaki tüm konfigürasyonların kümesi olmak üzere;

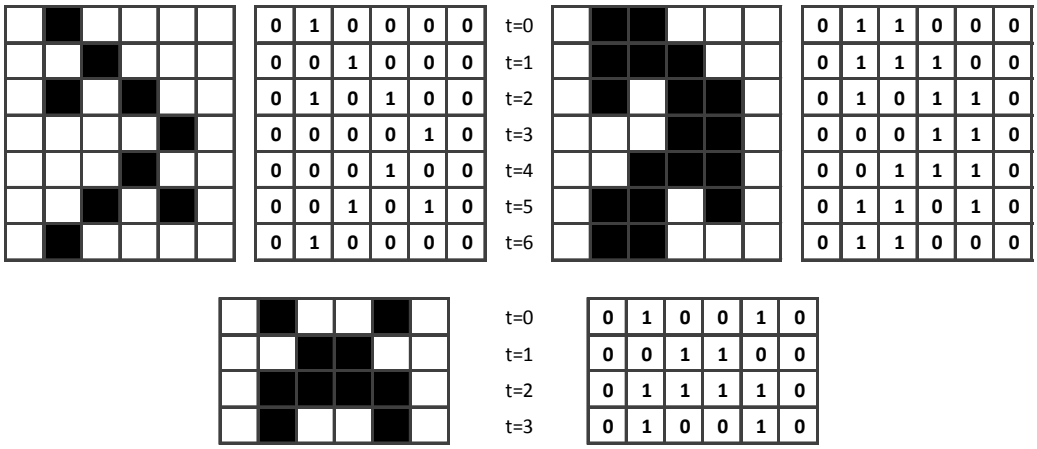
$U_0 = \{0000\}$

$U_1 = \{0001, 0010, 0101, 1000, 0100, 1010\}$

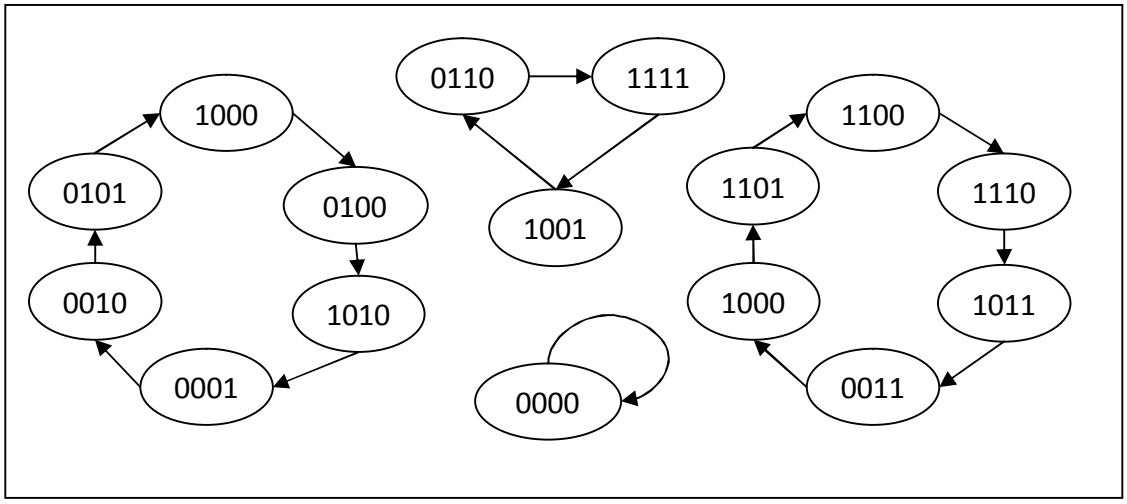
$U_2 = \{0111, 1101, 1100, 1110, 1011, 0011\}$

$U_3 = \{1001, 0110, 1111\}$

dört tane döngüden oluşur. Bu döngülerin uzay-zaman grafikleri ile devir diyagramları sırasıyla Şekil 2.19 ve Şekil 2.20'de gösterilmiştir.



Şekil 2. 19 $F = \langle 90, 90, 90, 90 \rangle$ kural vektörüne karşılık gelen durum tablosu ve uzay-zaman grafiği



Şekil 2. 20 $F = \langle 90, 90, 90, 90 \rangle$ kural vektörüne karşılık gelen devir diyagramı

2.6 İki Durumlu Bir Boyutlu Lineer Hücresel Dönüşümlerin Devir Yapısının Cebirsel İfadesi

Bu alt başlık altında durum kümesi $\mathbb{Z}_2 = \{0,1\}$ olan *(binary)* hücresel dönüşümlerin temsili matrislerinin karakteristik polinomu yardımıyla devir uzunlukları hesaplanacaktır.

Tanım 2.11 $n \times n$ tipinde T matrisi $\mathbb{F}_q$ cismi üzerinde tanımlı olsun. I_n, $n \times n$ tipinde birim matris ve x bilinmeyen olmak üzere, $(xI_n - T)$ matrisine T'nin *karakteristik matrisi* ve $\det(xI_n - T) = \varphi(x)$ polinomuna ise T'nin *karakteristik polinomu* denir [32].

T $(\det(T) \neq 0)$ iki durumlu bir boyutlu bir lineer hücresel dönüşümün temsili matrisi olsun. $\varphi(x) = \varphi_1(x)\varphi_2(x)...\varphi_r(x)$ T matrisinin karakteristik polinomu olmak üzere, T 'nin temsil ettiği iki durumlu bir boyutlu lineer hücresel dönüşümün devir yapısı aşağıdaki gibi hesaplanır [33].

i) Eğer T matrisinin karakteristik polinomu $\varphi(x) = \varphi_1(x)\varphi_2(x)...\varphi_r(x)$ ilkel polinomların çarpımı şeklinde ise, bu durumda $\varphi_i(x), (i = 1,2,...,r)$ çarpan polinomuna karşılık gelen devir yapısı $[1, \mu_i(k_i)]$ şeklindedir. Burada μ_i, $k_i -$ uzunluğundaki devirlerin sayısıdır. $der(\varphi_i(x)) = n_i$ olmak üzere; $\mu_i = (2^{n_i} - 1)/k_i$, $(k_i \in \mathbb{Z}^+)$ ve $\varphi_i(x) | x^{k_i} + 1$ şartını sağlayan en küçük sayıdır. Eğer $\varphi_1(x) \to [1, \mu_1(k_1)]$ ve $\varphi_2(x) \to [1, \mu_2(k_2)]$ devir yapılarına sahip iseler, bu durumda $\varphi_1(x)\varphi_2(x) \to [1, \mu_1(k_1), \mu_2(k_2), \mu(k)]$ devir yapısına sahiptir. Burada $\mu = \mu_1\mu_2 ebob(k_1, k_2)$ ve $k = ekok(k_1, k_2)$ dir.

ii) Eğer T matrisinin karakteristik polinomu $[\varphi(x)]^r$, $(r \in \mathbb{Z}^+)$ şeklinde ise, bu durumda devir yapısı $[1, \mu'(k')]$ şeklindedir. Burada $2^{r_1 - 1} < r < 2^{r_1}$ ve $der(\varphi(x)) = d$ olmak üzere; $\mu' = 2^{d(r-1)}(2^d - 1)/k'$, $(k' = k2^{r_1})$ şeklindedir.

Örnek 2.6 $F=\langle 90,150,90,150\rangle$ kural vektörüne sıfır sınır şartı altında karşılık gelen temsili matris ve karakteristik polinomu aşağıdaki gibidir:

$$T=\begin{pmatrix}0&1&0&0\\1&1&1&0\\0&1&0&1\\0&0&1&1\end{pmatrix},\ \varphi(x)=x^4+x+1\in\mathbb{F}_2[x].$$

$T^{2^4-1}=T^{15}=I$ olduğundan $\circ(T)=15$ dir. $\varphi(x)\to[1,\mu(k)]$ devir yapısı olsun. $der(\varphi(x))=4$, $\varphi(x)\big|x^{15}+1$ olduğundan $k=15$ ve $\mu=(2^4-1)/15=1\Rightarrow[1,1(15)]$ devir yapısı elde edilir. Yani bu devir yapısında biri sıfırdan oluşan diğeri ise $15-$ uzunluğunda bir döngüden oluşan iki döngü vardır. Sıfır dışındaki tüm konfigürasyonlar tek döngü içerisinde yer aldığından bu hücresel dönüşüm maksimal uzunlukludur. Bu hücresel dönüşümün döngü yapısını Şekil 2.18'de gösterdik.

Örnek 2.7 $F=\langle 90,90,90,90\rangle$ kural vektörüne sıfır sınır şartı altında karşılık gelen temsili matris;

$$T=\begin{pmatrix}0&1&0&0\\1&0&1&0\\0&1&0&1\\0&0&1&0\end{pmatrix},\ \varphi(x)=\left[x^2+x+1\right]^2=x^4+x^2+1\in\mathbb{F}_2[x].$$

$T^6=I$ olduğundan $\circ(T)=6$ dır. x^2+x+1 polinomuna karşılık gelen devir yapısı; $der(x^2+x+1)=2$ ve $x^2+x+1\big|x^3+1$ buradan $k=3$ ve $\mu=(2^2-1)/3=1$ yani x^2+x+1 polinomuna karşılık gelen devir yapısı $[1,1(3)]$ şeklindedir.

$\left[x^2+x+1\right]^2$ polinomuna karşılık gelen devir yapısı ise $[1,\mu'(k')]$ şeklindedir. Burada $2^{r_1-1}<r<2^{r_1}$ ve $der(\varphi(x))=2$, $r=2$, $r_1=1$, $k'=k2^{r_1}$ olmak üzere; $\mu'=2^{d(r-1)}(2^d-1)/k'$ dir. Buradan $k'=3\times 2^1=6$, $\mu'=2^{2(2-1)}(2^2-1)/6=2$ olduğundan $\left[x^2+x+1\right]^2$ polinomuna karşılık gelen devir yapısı $[1,2(6)]$ olur. O halde

$F = \langle 90, 90, 90, 90 \rangle$ kural vektörüne karşılık gelen devir yapısı $[1, 1(3), 2(6)]$ olarak bulunur. Bu hücresel dönüşümün döngü yapısını Şekil 2.20'de gösterdik.

BÖLÜM 3

İKİ DURUMLU HÜCRESEL DÖNÜŞÜMLERLE HATA DÜZELTEN KODLAR

Bu bölümde hata düzelten kodlar hakkında gerekli önbilgiler verilecektir. Daha sonra hücresel dönüşümlerle hata düzelten kodlar ile ilgili yapılan çalışmalar sırasıyla hücresel dönüşümlerle bit hata düzelten kodlar ve hücresel dönüşümlerle bayt hata düzelten kodlar başlıkları altında verilecektir. Hata düzelten kodlarla ilgili bilgiler Coding Theory, A First Course (2004) [34], The Art of Error Correcting Coding (2006) [35] isimli kitaplardan yararlanılarak hazırlanmıştır. Hücresel dönüşümlerle bit hata düzelten kodlar konusunda Design of CAECC Cellular Automata Based Error Correcting Code, (1994) [24] isimli makaleden ve hücresel dönüşümlerle bayt hata düzelten kodlar konusunda CA Based Byte Error Correcting Code, (1994) [25] makalelerinden yararlanılmıştır.

3.1 Hata Düzelten Kodlar

Tanım 3.1 $\mathbb{F}_q = \{0,1,\ldots,q-1\}$, q (burada q asal bir sayı) elemanlı sonlu bir cisim olmak üzere, $\mathbb{F}_q$ üzerinde $n-$uzunluğunda ve eleman sayısı M olan bir C kodu, $\mathbb{F}_q^n$ uzayının bir alt kümesidir. Böyle bir kod $(n,M)_q -$ kod şeklinde gösterilir.

Tanım 3.2 (Lineer Kod) $\mathbb{F}_q$ üzerinde $n-$uzunluğunda ve boyutu k olan bir C lineer kodu, $\mathbb{F}_q^n$ vektör uzayının bir alt uzayıdır. Böyle bir kod $[n,k]_q -$kod şeklinde gösterilir.

Özel olarak $q=2$ için $\mathbb{F}_2$ üzerinde bir koda ikili kod, $q=3$ için $\mathbb{F}_3$ üzerinde bir koda üçlü kod denir.

Örnek 3.1 $q=2$ için $C=\{0000,1000,0100,1100\}\subseteq\mathbb{F}_2^4$ kodu bir ikili lineer koddur.

Örnek 3.2 $q=3$ için $C=\{000,100,001,200,002,101,202,201,102,\}\subseteq\mathbb{F}_3^3$ kodu bir üçlü lineer koddur.

Tanım 3.3 C, $\mathbb{F}_q$ üzerinde bir lineer kod olsun.

i) C lineer kodun duali $C^{\perp}$ ile gösterilir ve $C^{\perp}=\{x\in\mathbb{F}_q^n:\langle x,c\rangle=0,\forall c\in C\}$ şeklinde tanımlanır.

ii) C lineer kodun boyutu, C'nin vektör uzayı olarak boyutudur ve $boy(C)$ ile gösterilir.

Teorem 3.1 C, $\mathbb{F}_q$ üzerinde $n-$uzunluğunda ve boyutu k olan bir lineer kod olsun. Bu durumda;

i) $|C|=q^{boy(C)}$ yani $M=q^k$ dır.

ii) $C^{\perp}$ bir lineer koddur ve $boy(C)+boy(C^{\perp})=n$ dir.

iii) $(C^{\perp})^{\perp}=C$ dir.

İspat:

i) C kodunun vektör uzayı olarak bir tabanı $\{c_1,c_2,\ldots,c_k\}$ olsun. Bu durumda $C=\left\{\sum_{i=1}^{k}\lambda_i c_i:\lambda_i\in\mathbb{F}_q,\ i=1,2,...,k\right\}$ yazılır. $|\mathbb{F}_q|=q$ olduğundan her bir λ_i için q tane seçim yapılabilir. Dolayısıyla C kodunun q^k elemanı vardır.

ii) İlk olarak $C^{\perp}=\{x\in\mathbb{F}_q^n:\langle x,c\rangle=0,\forall c\in C\}$ kümesinin $\mathbb{F}_q^n$ vektör uzayının bir alt uzayı olduğunu gösterelim.

a) Her $x,y\in C^{\perp}$ ve her $c\in C$ için $\langle x,c\rangle=0$, $\langle y,c\rangle=0$ olduğundan $\langle x+y,c\rangle=\langle x,c\rangle+\langle y,c\rangle=0\Rightarrow x+y\in C^{\perp}$ olur.

b) Her $x \in C^{\perp}$ ve her $c \in C$ için $\langle x,c \rangle = 0$ olduğundan her $\alpha \in \mathbb{F}_q$ için $\langle \alpha x,c \rangle = \alpha \langle x,c \rangle = 0 \Rightarrow \alpha x \in C^{\perp}$ olur. *(a)* ve *(b)*'den $C^{\perp}$, $\mathbb{F}_q^n$ vektör uzayının bir alt vektör uzayıdır. Yani $C^{\perp}$ bir lineer koddur.

Şimdi $boy(C) + boy(C^{\perp}) = n$ olduğunu gösterelim. Eğer $C = \{0\}$ ise $\mathbb{F}_q^n$ deki tüm vektörler C'ye diktir. Dolayısıyla $boy(C) + boy(C^{\perp}) = 0 + n = n$. $boy(C) = k \geq 1$ olarak alalım ve $\{c_1, c_2, \ldots, c_k\}$, C'nin bir tabanı olsun. Bu durumda $x \in C^{\perp} \Leftrightarrow \langle c_1, x \rangle = \langle c_2, x \rangle = \ldots = \langle c_k, x \rangle = 0$ yazılabilir. $\{c_1, c_2, \ldots, c_k\}$ taban elemanlarını satır kabul eden matris, $A = \begin{pmatrix} -c_1 - \\ -c_2 - \\ \vdots \\ -c_k - \end{pmatrix}$ matrisi olsun. Bu durumda $Ax^T = 0$ dır. Bu $n-$bilinmeyen ve $k-$tane denklemden oluşan bir lineer homojen denklem sistemidir. Böyle bir sistemin $n-k$ tane çözümü vardır. Dolayısıyla $C^{\perp}$ uzayının boyutu $n-k$ dır.

iii) $c \in C$ olsun. Bu durumda $\langle c,d \rangle = c.d = 0 \Rightarrow d \in C^{\perp} \Rightarrow \langle d,c \rangle = d.c = 0$ $\Rightarrow c \in (C^{\perp})^{\perp} \Rightarrow C \subseteq (C^{\perp})^{\perp}$ yazılır. Diğer taraftan $C \rightarrow [n,k]_q -$ kod olduğundan $C^{\perp} \rightarrow [n, n-k] -$ koddur. $|(C^{\perp})^{\perp}| = q^k = |C|$ olup $boy(C) = k$ dır. Buradan $(C^{\perp})^{\perp} = C$ elde edilir.

Örnek 3.3 $q = 2$ için $C = \{\overset{c_1}{0000}, \overset{c_2}{1000}, \overset{c_3}{0100}, \overset{c_4}{1100}\} \subseteq \mathbb{F}_2^4$ kodunun dual kodunu bulalım.

$C^{\perp} = \{x \in \mathbb{F}_2^4 : \langle x,c \rangle = 0, \forall c \in C\}$, burada $x = (x_1, x_2, x_3, x_4) \in C^{\perp} \subseteq \mathbb{F}_2^4$ olup $\langle x, c_1 \rangle = 0$, $\langle x, c_2 \rangle = 0$, $\langle x, c_3 \rangle = 0$, $\langle x, c_4 \rangle = 0 \Rightarrow x_1 = 0$, $x_2 = 0$, $x_3 = r$, $x_4 = s$, $r,s \in \mathbb{F}_2$ olmak üzere; $C^{\perp} = \{x = (0,0,r,s) \in \mathbb{F}_2^4 : r,s \in \mathbb{F}_2\} \Rightarrow C^{\perp} = \{0000, 0010, 0001, 0011\} \subseteq \mathbb{F}_2^4$ olarak bulunur.

Tanım 3.4 C, $\mathbb{F}_q$ cismi üzerinde bir lineer kod olsun. Eğer $C \subseteq C^{\perp}$ ise C'ye *kendine dik kod*, eğer $C = C^{\perp}$ ise C'ye *kendine dual kod* denir.

3.1.1 Üreteç ve Kontrol Matrisleri

Tanım 3.5 $C \to [n,k]_q$ parametrelerine sahip lineer kodun bir alt vektör uzayı olduğunu ve bu alt vektör uzayının $\{c_1, c_2, \ldots, c_k\}$ şeklinde bir tabana sahip olduğunu daha önce söylemiştik. Tabandaki vektörleri satır kabul eden G matrisine, C lineer kodun *üreteç matrisi* denir. Yani

$$G = \begin{pmatrix} -c_1 - \\ -c_2 - \\ \vdots \\ -c_k - \end{pmatrix}_{k \times n}$$

matrisine $C \to [n,k]_q$ parametrelerine sahip lineer kodun *üreteç matrisi* denir

Örnek 3.4 $q = 2$ için $C = \left\{ \overset{c_1}{0000}, \overset{c_2}{1000}, \overset{c_3}{0100}, \overset{c_4}{1100} \right\} \subseteq \mathbb{F}_2^4$ kodunun üreteç matrisini bulalım. $\{c_2, c_3\}$, C'nin bir tabanıdır. Dolayısıyla C'nin parametreleri $n = 4$, $k = 2$ olduğundan $[4,2]_2$ şeklindedir. C'nin üreteç matrisi ise,

$$G = \begin{pmatrix} -c_2 - \\ -c_3 - \end{pmatrix}_{k \times n} \Rightarrow G = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}_{2 \times 4}.$$

Örnek 3.5 $q = 3$ için $G = \begin{pmatrix} 1 & 0 & 2 \\ 2 & 1 & 0 \end{pmatrix}_{2 \times 3}$ matrisinin ürettiği lineer kodu bulalım. $C \to [3,2]_3$ koddur. $C = \{\alpha(102) + \beta(210) : \alpha, \beta \in \mathbb{F}_3\} \subseteq \mathbb{F}_3^3$, $|C| = q^k \Rightarrow |C| = 3^2 = 9$ $C = \{000, 102, 210, 201, 120, 012, 021, 111, 222\} \subseteq \mathbb{F}_3^3$ olarak bulunur.

- Bir C lineer kodun kendine dik olduğunu göstermek için üreteç matrisindeki satırların ikişerli birbirlerine dik olduğunu göstermek yeterlidir.

Tanım 3.6 Bir C lineer kodunun *parite kontrol matrisi* $C^{\perp}$ dualinin üreteç matrisidir ve genellikle H ile gösterilir. Yani $C = \{x \in \mathbb{F}_q^n : Hx^T = 0\}$ dir.

Örnek 3.6 $q=2$ için $C=\{0000,1000,0100,1100\}$ kodunun $C^{\perp}$ kodunu ve parite kontrol matrisini bulalım. $C^{\perp}=\{x\in\mathbb{F}_2^4 : \langle x,c\rangle=0, \forall c\in C\}$ olduğundan $C^{\perp}=\{0000,0001,0010,0011\}$ şeklindedir. Dolayısıyla $C^{\perp}$ nin üreteç matrisi,

$H=\begin{pmatrix}0&0&1&0\\0&0&0&1\end{pmatrix}_{2\times 4}$ olarak bulunur.

Tanım 3.7 $C\to[n,k]_q$ lineer kodun üreteç matrisi $G=[I_k \mid A]_{k\times n}$ olsun. Bu durumda G'nin *standart formda* olduğu söylenir. Burada I_k, $k\times k$ tipinde birim matris, A ise $k\times(n-k)$ tipinde bir matristir.

Teorem 3.2 Bir $C\to[n,k]_q$ kodunun üreteç matrisi standart formda verilmişse $H=[-A \mid I_{n-k}]$ matrisi, C için bir kontrol matrisidir.

- $C\to[n,k]_q$ lineer kodunun üreteç ve kontrol matrisleri sırasıyla aşağıdaki gibi olsun.

$$G=\begin{pmatrix}-c_1-\\-c_2-\\\vdots\\-c_k-\end{pmatrix}_{k\times n},\quad H=\begin{pmatrix}-h_1-\\-h_2-\\\vdots\\-h_{n-k}-\end{pmatrix}_{(n-k)\times n}$$

Bu durumda $G.H^T=0$ dır. Burada 0, $k\times(n-k)$ tipinde sıfır matristir.

Örnek 3.7 $G=\begin{pmatrix}1&0&0&1\\0&1&1&1\end{pmatrix}_{2\times 4}$ üreteç matrisiyle verilen $C\to[4,2]_3$ parametrelerine sahip lineer kodun kontrol matrisini bulalım. Burada $n=4$ ve $k=2$ olduğundan $n-k=2$ dir. Dolayısıyla $H=[-A \mid I_{n-k}]$ ve $-A^T=\begin{pmatrix}0&2\\2&2\end{pmatrix}$ şeklindedir. O halde

$H=\begin{pmatrix}0&2&1&0\\2&2&0&1\end{pmatrix}_{2\times 4}$ olur.

3.1.2 Hamming Uzaklık- Hamming Ağırlık

Tanım 3.8 (Hamming Uzaklığı) $x=(x_1,x_2,...,x_n),\ y=(y_1,y_2,...,y_n)\in\mathbb{F}_q^n$ olsun. $d:\mathbb{F}_q^n\times\mathbb{F}_q^n\to\mathbb{N}$, $d_H(x,y)=\left|\{i:x_i\neq y_i,i=1,2,...,n\}\right|$ şeklinde tanımlanan fonksiyona *Hamming uzaklığı* denir.

Örnek 3.8 $\mathbb{F}_2$ üzerinde $x=(10011),\quad y=(10110)\Rightarrow d_H(x,y)=2$ dir. $\mathbb{F}_3$ üzerinde $x=(22101),\quad y=(12210)\Rightarrow d_H(x,y)=4$ olur.

Teorem 3.3 $d:\mathbb{F}_q^n\times\mathbb{F}_q^n\to\mathbb{N}$, $d_H(x,y)=\left|\{i:x_i\neq y_i,i=1,2,...,n\}\right|$ şeklinde tanımlanan fonksiyon $\mathbb{F}_q^n$ vektör uzayı üzerinde bir metriktir.

Tanım 3.9 Bir $C\leq_{a.v}\mathbb{F}_q^n$ lineer kodunun minimum Hamming uzaklığı C'nin birbirinden farklı mümkün tüm kod söz çiftleri arasındaki en küçük Hamming uzaklığıdır ve $d(C)$ ile gösterilir. Yani $d(C)=\min\{d_H(x,y):x\neq y,x,y\in C\}$ olarak tanımlanır. Minimum uzaklığı d olan bir $[n,k]_q-$kod, $[n,k,d]_q-$kod ile gösterilir.

Tanım 3.10 (Hamming Ağırlığı) Bir $x\in\mathbb{F}_q^n$ vektörünün Hamming ağırlığı $w_H(x)$ ile gösterilir ve x vektörünün sıfırdan farklı koordinatlarının sayısıdır. Yani $w_H(x)=\left|\{i:x_i\neq 0,\ x=(x_1,x_2,...,x_n)\in\mathbb{F}_q^n\}\right|$. Diğer bir ifadeyle $w_H(x)=d_H(x,0)$ dır.

Örnek 3.9 $\mathbb{F}_2$ üzerinde $x=(11101)\Rightarrow w_H(x)=4$. $\mathbb{F}_3$ üzerinde $x=(10220)\Rightarrow w_H(x)=3$.

Teorem 3.4 $x,y\in\mathbb{F}_q^n$ için $d_H(x,y)=w_H(x-y)$ dir.

İspat: $x,y\in\mathbb{F}_q^n$ olsun.

$$\begin{aligned}d_H(x,y)&=d_H(x_1,y_1)+d_H(x_2,y_2)+\ldots+d_H(x_n,y_n)\\&=d_H(x_1-y_1,0)+d_H(x_2-y_2,0)+\ldots+d_H(x_n-y_n,0)\\&=w_H(x_1-y_1)+w_H(x_2-y_2)+\ldots+w_H(x_n-y_n)\\&=w_H(x-y).\end{aligned}$$

Teorem 3.5 Eğer $C \subseteq \mathbb{F}_q^n$ bir lineer kod ise $d_{\min}(C) = w_{\min}(C)$.

İspat:

$$\begin{aligned} d_{\min}(C) &= \min\{d_H(x,y) : x,y \in C, x \neq y\} \\ &= \min\{w_H(x-y) : x,y \in C, x \neq y\} \\ &= \min\{w_H(c) : c \in C, c \neq 0\} \\ &= w_{\min}(C). \end{aligned}$$

Teorem 3.6 C lineer kodunun minimum uzaklığının d olabilmesi için gerek ve yeter şart C'nin kontrol matrisinin en az $d-$tane sütununun lineer bağımlı ve herhangi $(d-1)-$tane sütununun lineer bağımsız olmasıdır.

İspat: $c = (c_1, c_2, \ldots c_n) \in C$ kod sözünün ağırlığı sıfırdan faklı olsun. c kod sözünün sıfırdan farklı koordinatları $\{i_1, i_2, \ldots, i_e\}$ ile gösterilsin. Bu durumda eğer $c_i \notin \{i_1, i_2, \ldots, i_e\}$ ise $c_i = 0$ dır. $h_i, (i = 1,2,\ldots,n)$ ile C'nin H kontrol matrisinin i. sütununu gösterelim. H, $C^{\perp}$'i ürettiğinden c kod sözü H kontrol matrisinin tüm satırlarına diktir. Dolayısıyla $cH^T = c_{i_1} h_{i_1}^T + c_{i_2} h_{i_2}^T + \ldots + c_{i_e} h_{i_e}^T = 0$ yazılır. Bu da kontrol matrisinin en az $e-$ tane sütununun lineer bağımlı olduğunu gösterir. Eğer en küçük e sayısı d ise C'de ağırlığı $\leq d-1$ olan kod söz yoktur. Yani H kontrol matrisinin $\leq d-1$ sütunları lineer bağımsızdır. Benzer şekilde eğer C'de ağırlığı $\leq d$ olan sıfırdan farklı kod söz varsa bunun anlamı C'nin $\leq d$ tane sütunu lineer bağımlıdır.

Örnek 3.10 $H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}_{3\times 7}$ kontrol matrisine sahip C lineer kodunun minimum uzaklığı $d(C) = 3$ olarak bulunur. Çünkü H matrisinin herhangi iki sütunu lineer bağımsız olmasına karşın ilk üç sütunu lineer bağımlıdır.

3.1.3 Lineer Kodlarda Kodlama

Tanım 3.11 $C,\ [n,k,d]_q$ parametrelerine sahip bir lineer kod ve $G=\begin{pmatrix}-c_1-\\-c_2-\\\vdots\\-c_k-\end{pmatrix}$ C'nin üreteç matrisi olsun. Bu durumda bir $u=(u_1,u_2,...,u_k)\in\mathbb{F}_q^k$ için $u.G=c\in C\subseteq\mathbb{F}_q^n$ şeklindeki ifadeye bir u sözünün kodlanması ve $c\in C$ vektörüne de bir *kod söz* denir.

Örnek 3.11 $G=\begin{pmatrix}1&1&0&1&0&0&1\\0&1&0&1&0&1&0\\1&1&1&0&0&0&0\\1&0&0&1&1&0&0\end{pmatrix}_{4\times7}$ $C\to[7,4,3]_2$ parametrelerine sahip lineer kodun üreteç matrisi olsun. Bir $u=1011\in\mathbb{F}_2^4$ elemanını kodlayalım. Bu durumda $u.G=1010101\in C\subseteq\mathbb{F}_2^7$ olarak elde edilir. Burada $|C|=\left|\mathbb{F}_2^4\right|=2^4=16$ olduğunu görmek kolaydır.

3.1.4 Lineer Kodlarda Dekodlama

Tanım 3.12 $C\subseteq\mathbb{F}_q^n$ bir $\mathbb{F}_q$ lineer kod olsun. C'nin herhangi bir $u\in\mathbb{F}_q^n$ sözüyle belirli koseti (sol veya sağ yan kümesi) $C+u=\{c+u:c\in C\}=u+C$ şeklindeki kümeye denir.

Örnek 3.12 $\mathbb{F}_2$ üzerinde $C=\{000,011,100,111\}\subseteq\mathbb{F}_2^3$ lineer kodunun tüm kosetlerini yazalım.

$$\begin{array}{l|l}000+C=\{000,011,100,111\}=C+000 & 010+C=\{010,001,110,101\}=C+010\\100+C=\{000,011,100,111\}=C+100 & 001+C=\{001,010,101,110\}=C+001\\011+C=\{000,011,100,111\}=C+011 & 110+C=\{010,001,110,101\}=C+110\\111+C=\{000,011,100,111\}=C+111 & 101+C=\{001,010,101,110\}=C+101\end{array}$$

Teorem 3.7 $C\to[n,k,d]_q$ parametrelerine sahip bir lineer kod olsun.

i) $\mathbb{F}_q^n$ nin her vektörü C'nin bir kosetindedir.

ii) Her $u \in \mathbb{F}_q^n$ için $|C+u| = |C| = q^k$.

iii) Her $u, v \in \mathbb{F}_q^n$ için $u \in v+C \Rightarrow u+C = v+C$.

iv) İki koset ya aynı ya da ayrıktır.

v) C'nin q^{n-k} tane koseti vardır.

vi) Her $u, v \in \mathbb{F}_q^n$ için $u-v \in C \Leftrightarrow u$ ve v C'nin aynı kosetindedir.

İspat:

i) $u \in \mathbb{F}_q^n$ olsun. $00...0 \in C$ olduğundan $u+C = \{u, ...\}$ şeklinde u vektörünü içerir.

ii) Her $u \in \mathbb{F}_q^n$ için $|C+u| \le q^k$, $u+c_1, u+c_2 \in u+C$ olsun. $c_1 \neq c_2$ iken $u+c_1 \neq u+c_2$ dir. Eğer $u+c_1 = u+c_2 \Rightarrow c_1 = c_2$ olmak zorundadır. O halde kosetin herhangi iki elemanı aynı olamaz.

iii) $u \in v+C \Rightarrow u+C \subseteq v+C$ dir. Aynı zamanda $|u+C| = q^k = |v+C|$ olduğundan $|u+C| = |v+C|$.

iv) $u+C$ ve $v+C$ C'nin iki koseti olsun. $x \in (u+C) \cap (v+C)$ alalım. Bu durumda $x \in (u+C)$ ve $x \in (v+C) \Rightarrow x+C = u+C$ ve $x+C = v+C \Rightarrow u+C = v+C$.

v) $|\mathbb{F}_q^n| = q^n \Rightarrow \mathbb{F}_q^n = \{u_1, u_2, ..., u_{q^n}\}$ yazılır. $|C| = q^k$ olduğunu biliyoruz. Buradan hareketle birbirinden farklı kosetlerin sayısı s olsun. $s.q^k = q^n \Rightarrow s = q^{n-k}$ olur.

vi) $(\Rightarrow)$ $u-v = c \in C$ olsun. $u = c+v \in v+C \Rightarrow u+C = v+C \Rightarrow u-v \in C$.

$$(\Leftarrow) \left.\begin{array}{l} u \in x+C \;\; u = c_1 + x \\ v \in x+C \;\; v = c_2 + x \end{array}\right\} \Rightarrow u-v = c_1 - c_2 = c_3 \in C.$$

Tanım 3.13 Bir kosetteki en küçük ağırlığa sahip söze *koset lideri* denir.

Örnek 3.13 (Slepian Standart Dizisi) $C=\{0000,1011,0101,1110\}\subseteq\mathbb{F}_2^4$ lineer kodunun tüm kosetlerini bulalım. $n=4$ ve $k=2$ olduğundan C'nin toplam $q^{n-k}=2^{4-2}=4$ tane koseti vardır. Bunlar;

$$0000+C=\{0000,1011,0101,1110\}$$
$$0001+C=\{0001,1010,0100,1111\}$$
$$0010+C=\{0010,1001,0111,1100\}$$
$$1000+C=\{1000,0011,1101,0110\}$$

şeklindedir.

3.1.4.1 En Yakın Komşu Dekodlaması (Nearest Neighbour Decoding)

v gönderilen kod söz ve w alınan söz (hatalı olabilir) olsun. e hata vektörü w vektörünü içeren kosetteki en küçük ağırlıklı vektör olmak üzere, $v+e=w\Rightarrow v=w-e$.

Örnek 3.14 $C=\{0000,1011,0101,1110\}\subseteq\mathbb{F}_2^4$ lineer kodu için $w=1101$ alınan sözünü dekodlayalım. Örnek 3.13'ten w vektörünü içeren kosetteki en küçük ağırlıklı vektör $e=1000$ olduğundan $v=w-e=1101-1000=0101\in C$ olarak dekodlanır. Eğer en küçük ağırlıklı birden fazla koset lideri varsa en yakın komşu dekodlaması yapılamaz. Bu durumda alınan söz tekrar istenir.

3.1.4.2 Sendrom Dekodlaması

Tanım 3.14 $C\to[n,k,d]_q$ parametrelerine sahip bir lineer kod olsun. H, C'nin bir kontrol matrisi olmak üzere, $w\in\mathbb{F}_q^n$ nin sendromu $s(w)$ ile gösterilir ve $s(w)=w.H^T$ şeklinde hesaplanır.

Teorem 3.8 $C \to [n,k,d]_q$ parametrelerine sahip bir lineer kod ve H, C'nin bir kontrol matrisi olsun. $u, v \in \mathbb{F}_q^n$ için;

i) $s(u+v) = s(u) + s(v)$.

ii) $s(u) = 0 \Leftrightarrow u \in C$.

iii) $s(u) = s(v) \Leftrightarrow$ u ile v C'nin aynı kosetindedir.

İspat:

i) $s(u+v) = (u+v)H^T = uH^T + vH^T = s(u) + s(v)$.

ii) $s(u) = 0 \Leftrightarrow uH^T = 0 \Leftrightarrow u \in C$.

iii) $s(u) = s(v) \Leftrightarrow uH^T = vH^T \Leftrightarrow uH^T - vH^T = 0 \Leftrightarrow (u-v)H^T \Leftrightarrow u - v \in C$.

Sonuç 3.1 $C \to [n,k,d]_q$ parametrelerine sahip bir lineer kod olsun. C kodu için mümkün sendromların sayısı kosetlerin sayısına eşittir. Yani q^{n-k} tane sendrom vardır.

Sendrom Tablosu Düzenleme

i) Verilen kodun tüm kosetleri yazılır ve her bir kosetin koset lideri seçilir.

ii) Verilen kodun H kontrol matrisi belirlenir ve tüm koset liderlerinin sendromları hesaplanır.

iii) Her bir koset lideri ile karşılık gelen sendrom tablo haline getirilir.

Teorem 3.9 Bir C lineer kodunun $t-$tane hata düzeltebilmesi için gerek ve yeter şart $d(C) = d \geq 2t+1$ olmasıdır $\left(t = \left\lfloor \frac{d-1}{2} \right\rfloor \right)$.

İspat: $(\Rightarrow)$ $d(C) \geq 2t+1$ olsun. c gönderilen kod söz ve x alınan söz olsun. Eğer iletimde t veya daha az hata meydana gelmişse $d(x,c) \leq t$ olur. Bu yüzden herhangi $c' \in C, (c' \neq c)$ kod sözü için $d(x,c') \geq d(c,c') - d(x,c) \geq 2t+1-t = t+1 > d(x,c)$ olur.

Bu durumda en yakın komşu dekodlaması kullanılırsa x doğru bir şekilde c olarak dekodlanacaktır. Bu da C lineer kodunun $t-$tane hata düzeltebildiğini gösterir.

$(\Leftarrow)$ C lineer kodu $t-$tane hata düzeltebilsin. Eğer $d(C)<2t+1$ ise bu durumda $d(c,c')=d(C)\le 2t$ olacak şekilde farklı $c,c'\in C$ vardır. c kod sözü gönderilsin ve $t-$tane hata meydana gelmiş olsun. Bu durumda c ya yanlış bir şekilde c' olarak dekodlanır ya da tamamlanmamış dekodlama kuralı kullanıldığında düzeltilemeyecek hata meydana gelmiştir. Bu durum C lineer kodunun $t-$tane hata düzeltebildiği kabulüyle çelişir. Bu da $d(C)\ge 2t+1$ olması gerektiğini gösterir.

Örnek 3.15 $H=\begin{pmatrix}0&0&0&1&1&1&1\\0&1&1&0&0&1&1\\1&0&1&0&1&0&1\end{pmatrix}_{3\times 7}$ kontrol matrisine sahip $C\to[7,4,3]_2$ parametrelerine sahip lineer kod için sendrom tablosu oluşturunuz.

H uygun permütasyonlarla standart hale getirilerek G' elde edilir. Permütasyonlar geri alınarak G elde edilir. Bu durumda

$$G=\begin{pmatrix}1&1&0&1&0&0&1\\0&1&0&1&0&1&0\\1&1&1&0&0&0&0\\1&0&0&1&1&0&0\end{pmatrix}_{4\times 7}$$

olarak bulunur. Buradan G'nin mümkün bütün satır kombinasyonlarıyla $16-$elemanlı C lineer kodu elde edilir.

$$C=\left\{\begin{matrix}0000000,1101001,0101010,1110000,1001100,1000011,0110011,0011001,\\1011010,0100101,1100110,0111100,0001111,1010101,0010110,1111111\end{matrix}\right\}$$

$n=7,\ n-k=3\Rightarrow k=4$ olduğundan toplam $q^{n-k}=2^{7-4}=8$ tane sendrom vardır. Sendrom tablosu aşağıdaki gibidir:

Koset Lideri	Sendrom
0000000	000
1000000	001
0100000	010
0010000	011
0001000	100
0000100	101
0000010	110
0000001	111

Tablo 3. 1 İkili Hamming kod için sendrom tablosu

Sendrom Dekodlama Aşamaları

i) Alınan bir w sözü için $s(w)$ sendromu hesaplanır.

ii) $s(w)=s(u)$ olacak şekilde u koset lideri bulunur.

iii) v gönderilen kod söz olmak üzere, $v=w-u$ hesaplanır.

Örnek 3.16 $H=\begin{pmatrix}0&0&0&1&1&1&1\\0&1&1&0&0&1&1\\1&0&1&0&1&0&1\end{pmatrix}_{3\times 7}$ kontrol matrisine sahip $C\to[7,4,3]_2$ parametrelerine sahip lineer kod için $w=1111001$ sözünü dekodlayınız.

i) Alınan w sözü için $s(w)=w.H^T=011$ olarak hesaplanır.

ii) Tablo 3.1'den $s(w)=s(u)=011\Rightarrow u=0010000$ dir.

iii) $v=w-u=1111001-0010000=1101001\in C$ olarak bulunur.

3.1.5 Hamming Kodları

Tanım 3.15 $\mathbb{F}_q$ üzerinde sütun vektörlerinin herhangi iki tanesi lineer bağımsız olan $H=\begin{pmatrix} h_1 & h_2 & \dots & h_n \end{pmatrix}_{r\times n}$ kontrol matrisine sahip lineer koda q'lu Hamming kod denir. $Ham(r,q)$ ile gösterilir. q'lu Hamming kodu $\left[n=\frac{q^r-1}{q-1}, k=\frac{q^r-1}{q-1}-r, d=3 \right]_q$ parametrelerine sahiptir. Özel olarak $q=2$ iken $Ham(r,q)=Ham(3,2)\to[7,4,3]_2$ ikili Hamming kodu adını alır ve Örnek 3.16'da verilen kontrol matrisine sahiptir.

3.1.6 Devirli Kodlar

Tanım 3.16 $C\subseteq\mathbb{F}_q^n$ bir lineer kod olsun. Eğer $c=(c_0,c_1,c_2,...,c_{n-1})\in C$ iken $c=(c_{n-1},c_0,c_1,...,c_{n-2})\in C$ oluyorsa C'ye bir devirli kod denir.

Örnek 3.17 $C=\{000,110,011,101\}\subseteq\mathbb{F}_2^3$ ile verilen kodu $[3,2,2]_2$ parametrelerine sahip bir devirli koddur.

3.1.6.1 Devirli Kodun Polinom ile Temsili

$C\subseteq\mathbb{F}_q^n$ bir devirli kod olmak üzere;

$$\varphi: C\to \mathbb{F}_q[x]\Big/\langle x^n-1\rangle = R_n,\quad \varphi(c)=\varphi(c_0,c_1,...,c_{n-1})=c_0+c_1x+...+c_{n-1}x^{n-1} \tag{3.1}$$

şeklindeki dönüşüm her devirli koda bir polinom karşılık getirir. Burada $\mathbb{F}_q[x]\Big/\langle x^n-1\rangle = R_n$ bölüm halkası ve $\langle x^n-1\rangle$ ise $\mathbb{F}_q[x]$ polinom halkasının bir idealidir.

Örnek 3.18 $C=\{000,110,011,101\}\subseteq\mathbb{F}_2^3$ ile verilen koda karşılık $\varphi(C)=\{0,1+x,x+x^2,1+x^2\}\subseteq R_3$ gelir. Burada $R_3=\mathbb{F}_2[x]\Big/\langle x^3-1\rangle$

$= \{0,1,x,x^2,1+x,x+x^2,1+x^2,1+x+x^2\}$ şeklindedir. Burada $\varphi(C)$'nin R_3'ün bir ideali olduğunu belirtelim.

- Her devirli kodun φ altındaki görüntüsü bir ideal oluşturur ve her ideale karşılık bir devirli kod gelir.

Teorem 3.10 $\varphi(C)$, $\mathbb{F}_q[x]/\langle x^n-1\rangle = R_n$ bölüm halkasının sıfırdan farklı bir ideali olsun. Bu durumda bir en küçük dereceli monik $g(x) \in \varphi(C)$ polinomu vardır ve $g(x) \mid x^n-1$ dir.

İspat: R_n Euclid bölgesi olduğundan bölme algoritması uygulanabilir. Dolayısıyla $x^n-1 = q(x)g(x)+r(x),\ der(r(x)) < der(g(x))$ olacak şekilde tek türlü belirli $r(x)$ ve $q(x)$ polinomları vardır. Bu durumda $r(x) = (x^n-1) - q(x)g(x) \in \varphi(C)$ dir. Ancak bu durum $g(x)$'in en küçük dereceli oluşuyla çelişir. O halde $r(x) = 0$ ve $x^n-1 = q(x)g(x)$ olur. Buradan $g(x) \mid x^n-1$ yazılır.

Tanım 3.17 Teorem 3.10'da verilen $g(x) \in \varphi(C)$ polinomuna $\varphi(C)$'nin ve dolayısıyla C devirli kodun üreteç polinomu denir ve $C = \langle g(x) \rangle$ şeklinde gösterilir.

Tanım 3.18 $\varphi(C)$, $\mathbb{F}_q[x]/\langle x^n-1\rangle = R_n$ bölüm halkasının sıfırdan farklı bir ideali ve $C = \langle g(x) \rangle$, $der(g(x)) = r$ olmak üzere, $G = \begin{pmatrix} g(x) \\ xg(x) \\ \vdots \\ x^{n-r-1}g(x) \end{pmatrix}_{(n-r)\times n}$ ile verilen matrise C devirli kodun üreteç matrisi denir.

Örnek 3.19 $g(x)=1+x+x^3 \mid x^7-1$ olarak seçilirse $C=\langle g(x)\rangle$ bir $[7,4]_2$ devirli ikili koddur ve üreteç matrisi;

$$G=\begin{pmatrix} g(x) \\ xg(x) \\ x^2g(x) \\ x^3g(x) \end{pmatrix}_{4\times 7} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}_{4\times 7}$$

şeklindedir. G'nin ürettiği devirli kod Örnek 3.11'deki üreteç matrisiyle verilen ikili Hamming koda denktir. Yani G matrisine elementer satır ve sütun işlemleri uygulanarak Örnek 3.11'deki üreteç matrisi elde edilebilir.

3.1.7 BCH Kodlar

BCH (Bose, Chaudhuri, Hocquenghem) kodları devirli kodların özel bir sınıfı olmakla beraber Hamming kodların çoklu hataya genelleştirilmesidir.

Tanım 3.19 $\mathbb{F}_{q^m}-\{0\}=\mathbb{F}_{q^m}^*=\langle\alpha\rangle$ ve $\mu^{(i)}(x)$ ile α ilkel elemanının $\mathbb{F}_q$'daki minimal polinomunu gösterelim. $g(x)=ekok\left[\mu^{(a)}(x),\mu^{(a+1)}(x),...,\mu^{(a+\delta-2)}(x)\right],(a\in\mathbb{N})$ polinomu tarafından üretilen devirli koda $n=q^m-1$ uzunluğunda ve $\delta-$tasarlanmış minimum uzaklığa sahip q'lu BCH kod denir. Eğer $a=1$ olarak seçilirse dar anlamda (narrow sense) BCH kod adını alır.

Örnek 3.20 $\alpha\in\mathbb{F}_{2^3}$, $1+x+x^3$ polinomunun bir kökü olsun. Bu durumda $\mu^{(1)}(x)=\mu^{(2)}(x)=1+x+x^3$ olup $7-$uzunluğunda dar anlamda 2'li BCH kodu $g(x)=ekok\left[\mu^{(1)}(x),\mu^{(2)}(x)\right]=1+x+x^3$ tarafından üretilir ve $[7,4,3]_2$ Hamming koda denktir.

3.1.8 Reed-Solomon Kodlar

Tanım 3.20 BCH kodlarda $n=q^m-1$ ifadesindeki $m=1$ olarak alındığında elde edilen kodlara Reed-Solomon kodlar denir. Yani q'lu bir Reed-Solomon kod $q-1$

uzunluğunda ve $g(x)=ekok\left[\left(x-\alpha^{a}\right),\left(x-\alpha^{a+1}\right),...,\left(x-\alpha^{a+\delta-2}\right)\right]$

$=\left(x-\alpha^{a}\right)\left(x-\alpha^{a+1}\right)...\left(x-\alpha^{a+\delta-2}\right)$ polinomu tarafından üretilen devirli bir koddur.

Örnek 3.21

i) 7'li Reed-Solomon kod $6-$uzunluğunda ve $\mathbb{F}_7^*=\langle 3\rangle$, $\delta=4\Rightarrow a+\delta-2=3$ $g(x)=(x-3)\left(x-3^2\right)\left(x-3^3\right)=6+x+3x^2+x^3$ polinomu tarafından üretilen $[6,3]_7-$koddur. Üreteç ve kontrol matrisleri sırasıyla; $G=\begin{pmatrix}6&1&3&1&0&0\\0&6&1&3&1&0\\0&0&6&1&3&1\end{pmatrix}$, $H=\begin{pmatrix}1&4&1&1&0&0\\0&1&4&1&1&0\\0&0&1&4&1&1\end{pmatrix}$ şeklindedir. Bu kodun minimum uzaklığı kontrol matrisinden 4 olarak bulunur. O halde $[6,3,4]_7-$koddur.

ii) 8'li Reed-Solomon kod $7-$uzunluğunda ve $\mathbb{F}_8^*=\langle\alpha\rangle$, $\delta=3\Rightarrow a+\delta-2=2$ $g(x)=(x-\alpha)\left(x-\alpha^2\right)=1+\alpha+\left(\alpha^2+\alpha\right)x+x^2$ polinomu tarafından üretilen $[7,5]_8-$ koddur. Burada α, $1+x+x^3\in\mathbb{F}_2[x]$ polinomunun bir köküdür. Üreteç ve kontrol matrisleri sırasıyla;

$$G=\begin{pmatrix}1+\alpha&\alpha^2+\alpha&1&0&0&0&0\\0&1+\alpha&\alpha^2+\alpha&1&0&0&0\\0&0&1+\alpha&\alpha^2+\alpha&1&0&0\\0&0&0&1+\alpha&\alpha^2+\alpha&1&0\\0&0&0&0&1+\alpha&\alpha^2+\alpha&1\end{pmatrix}$$ ve

$H=\begin{pmatrix}1&\alpha^4&1&1+\alpha^4&1+\alpha^4&\alpha^4&0\\0&1&\alpha^4&1&1+\alpha^4&1+\alpha^4&\alpha^4\end{pmatrix}$ olarak bulunur.

Teorem 3.11 $g(x)=\prod_{i=a+1}^{a+\delta-2}\left(x-\alpha^{i}\right)$ polinomu tarafından üretilen $(q-1)-$uzunluğunda q'lu Reed-Solomon kod herhangi $2\le\delta\le q-1$ için $[q-1,q-\delta,\delta]_q$ parametrelerine sahip bir devirli koddur.

İspat: $der\left(g\left(x\right)\right)=\delta-1$ olduğundan söz konusu kodun boyutu tam olarak $q-1-\left(\delta-1\right)=q-\delta$ olur. Minimum uzaklığı ise $\delta\le d\le\left(q-1\right)+1-k=\delta$ olur.

3.2 Hücresel Dönüşümlerle Bit Hata Düzelten Kodlar

3.2.1 Kodlama

Tanım 3.21 $I=\left(i_1,i_2,...,i_n\right)\to n-$bitten oluşan bilgi bitleri ve T bir boyutlu lineer temel hücresel dönüşümlerin temsili matrisi olmak üzere, bir $k\in\mathbb{Z}^+$ için $T^k\left[I\right]=\left(c_1,c_2,...,c_n\right)$ olsun. $CW=\left(I,T^k\left[I\right]\right)=\left(i_1,i_2,...,i_n,c_{n+1},c_{n+2},...,c_{2n}\right)$ şeklindeki ifadeye bir kod söz denir.

Teorem 3.12 $n-$hücreli bir boyutlu temel lineer hücresel dönüşümün temsili matrisi T olsun. Bazı $k\in\mathbb{Z}^+$ değerleri için T^k matrisinin herhangi $i\left(0<i<d\right)$ sayıdaki sütununun toplamı $\left(d-i\right)-$tane 1 içeriyorsa, bu durumda T matrisi $t-$uzaklığa sahip kod üretir [24].

İspat: T^k $(k\in\mathbb{Z}^+)$ herhangi $i-$tane sütununun toplamı $\left(d-i\right)-$tane 1 içeren temsili matris ve I_1, I_2 farklı sözler olsun. $CW_1=\left(I_1,T^k[I_1]\right)$, $CW_2=\left(I_2,T^k[I_2]\right)$ olmak üzere; $d_H\left(CW_1,CW_2\right)\ge d$ olduğunu göstermeliyiz. $d_H(I_1,I_2)\ge i$ olarak kabul edelim. Yani I_1 ve I_2'nin $i-$tane koordinatı birbirinden farklı olsun. Bu farklı koordinatlara $n_1,n_2,...,n_i$ diyelim. Eğer $i\ge d$ ise ispat tamamdır. $i<d$ olsun. T^k da $n_1,n_2,...,n_d$ sütunlarının koordinat koordinat toplamını düşünelim. Bu toplam en az $\left(d-i\right)-$tane 1 içermelidir. Bu toplamdaki sıfırdan farklı yerler $r_1,r_2,...,r_{t-i},...,r_j$ olsun. Şimdi $T^k[I_1]$ ve $T^k[I_2]$'nin r_1. koordinatını düşünelim. Bu koordinatları sırasıyla b_1 ve b_2 ile gösterelim.

$$b_1=t_{r_1,1}I_{11}+t_{r_1,2}I_{12}+...+t_{r_1,k}I_{1k} \tag{3.1}$$

$$b_2=t_{r_1,1}I_{21}+t_{r_1,2}I_{22}+...+t_{r_1,k}I_{2k} \tag{3.2}$$

Burada $t_{i,j}=T^k_{i,j}$ ve $I_{i,j}$, I_i'nin j. koordinatıdır. (3.1) ve (3.2) taraf tarafa toplanırsa;

$$b_1+b_2=t_{r_1,1}(I_{11}+I_{21})+t_{r_1,2}(I_{12}+I_{22})+...+t_{r_1,k}(I_{1k}+\ I_{2k}) \tag{3.3}$$

$$I_{1j} + I_{2j} = \begin{cases} 1, \ j = n_1, n_2, ..., n_t \\ 0, \ \text{diğer yerlerde} \end{cases} \tag{3.4}$$

şeklindedir. Bu yüzden $b_1 + b_2 = t_{r_1 n_1}, t_{r_1 n_2}, ..., t_{r_1 n_i} = 1$ dir. Çünkü $n_1, n_2, ..., n_i$ sütunlarının koordinat koordinat toplamı 1 dir. Buradan $T^k[I_1]$ ve $T^k[I_2]$'nin r_1. koordinatları birbirinden farklıdır. Benzer şekilde $r_2, ..., r_{t-i}, ..., r_j$ koordinatlarının da birbirinden farklı olduğu gösterilebilir. Bu durumda $d_H\left(T^k\left[I_1\right], T^k\left[I_2\right]\right) \geq d - i$ olarak elde edilir. Ayrıca $d_h\left(I_1, I_2\right) = i$ olduğundan $d_H\left(CW_1, CW_2\right) \geq d$ elde edilir.

Sonuç 3.1 $n-$hücreli bir boyutlu temel lineer hücresel dönüşümün temsili matrisinin $k-$devirde bir $\left[2n, n, 4\right]_2 -$kod üretebilmesi için T^k aşağıdaki şartları sağlamalıdır [24].

i) T^k nin her sütunu en az üç tane 1 içermelidir.

ii) T^k nin herhangi iki sütununun toplamının en az iki koordinatı birbirinden farklı olmalıdır.

iii) T^k nin herhangi üç sütununun toplamı sıfırdan farklı (lineer bağımsız) olmalıdır.

Lemma 3.1 $n-$hücreli bir boyutlu temel lineer hücresel dönüşümün temsili matrisi T olsun. Bu durumda T^k matrisinin ilk ve son sütunu en az $\left(k+1\right) -$tane 1 içerir [15].

İspat: Sadece üç hücreli komşuluk durumunu düşüneceğiz. Bu yüzden T'nin ilk ve son sütununda en çok iki tane 1 olabilir. Benzer şekilde T^2 nin ilk ve son sütununda en çok üç tane 1 olabilir. Bu şekilde devam ederek tümevarım ilkesinden T^k nın ilk ve son sütununda en çok $\left(k+1\right) -$tane 1 olabilir.

Teorem 3.13 Eğer $n-$hücreli bir boyutlu bir hücresel dönüşüm $k-$devirde $\left[2n, n, d\right]_2 -$kod üretiyorsa bu durumda $k \geq d - 2$ dir [15].

İspat: Teorem 3.12'den T^k nın her sütunu $\left(d-1\right) -$tane 1 içermelidir. Fakat Lemma 3.1'den T^k nın ilk ve son sütunlarında en çok $\left(k+1\right) -$tane 1 olabilir. Buradan $k+1 \geq d-1 \Rightarrow k \geq d-2$ olarak yazılır.

Örnek 3.22 $F=\langle 90,150,150,90\rangle$ Wolfram kurallarından oluşan global geçiş fonksiyonunu kullanarak $[8,4,4]_2-$kod üretelim.

$$T=\begin{pmatrix}0&1&0&0\\1&1&1&0\\0&1&1&1\\0&0&1&0\end{pmatrix}_{4\times 4},\ k=2 \text{ için } T^2=\begin{pmatrix}1&1&1&0\\1&1&0&1\\1&0&1&1\\0&1&1&1\end{pmatrix}_{4\times 4}$$

T^2 Sonuç 3.1'deki şartları sağladığından $G=\left[I_4\mid T^2\right]$ matrisi $[8,4,4]_2-$kod üretir. Şimdi $I=0001\in\mathbb{F}_2^4$ sözünü kodlayalım. $T^2[I]=0111$ olduğundan $CW=00010111\in\mathbb{F}_2^8$ kod söz olur.

3.2.2 Keyfi Sayıda Bilgi Bitlerine Sahip Bir Hata Düzelten ve İki Hata Fark Eden Kodun Üretilmesi

$T_n\ (n\geq 4)$ ile $n-$uzunluğunda bir boyutlu lineer Wolfram kurallarının temsili matrisini gösterelim. $n>4$ için T_n aşağıdaki gibi genişletilerek $[2n,n,4]_2-$kod üretilebilir.

$$T_n=\begin{pmatrix}&&&&0\\&(T_{n-1})&&&\vdots\\&&&&0\\&&&&1\\0&0&\cdots&0&1\end{pmatrix},\ n \text{ tek ise;}\quad T_n=\begin{pmatrix}&&&&0\\&(T_{n-1})&&&\vdots\\&&&&0\\&&&&1\\0&0&\cdots&1&0\end{pmatrix},\ n \text{ çift ise}$$

Örnek 3.23 $F=\langle 90,150,150,90\rangle$ Wolfram kurallarından oluşan global geçiş fonksiyonu için

$$T_4=\begin{pmatrix}0&1&0&0\\1&1&1&0\\0&1&1&1\\0&0&1&0\end{pmatrix},\ k=2 \text{ için } T_4^2=\begin{pmatrix}1&1&1&0\\1&1&0&1\\1&0&1&1\\0&1&1&1\end{pmatrix}$$

yazılır. $[10,5,4]_2-$kod üretmek için $n=5$ tek olduğundan

$T_5 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$ olur. $G = \left[I_5 \mid T_5^2 \right]$ matrisi bir $[10,5,4]_2 -$ kod üretir.

Teorem 3.14 Verilen herhangi $n-$uzunlukta I bilgi bitleri için $\left(I, T_n^2 [I]\right)$ şeklindeki kod sözler bir $[2n,n,4]_2 -$kod üretir [24].

İspat: n üzerine tümevarım uygulayalım. $n=4$ için Örnek 3.22'den ifade doğrudur. $n=s$ için ifade doğru olsun. $n=s+1$ için ifadenin doğru olduğunu gösterelim. Eğer s tek ise $s+1$ çift olur.

$$T_{s+1} = \begin{pmatrix} T_s & B \\ C & D \end{pmatrix} \Rightarrow T_{s+1}^2 = \begin{pmatrix} T_s^2 + BC & T_s B + BD \\ CT_s + DC & CB + D^2 \end{pmatrix}$$

Burada T_s, B, C, D sırasıyla $s \times s, s \times 1, 1 \times s, 1 \times 1$ ebatlı matrislerdir. $s+1$ çift olduğundan $B^T = (00...01)$, $C^T = (00...01)$, $D = (0)$. Ayrıca $t_{ss} \in T_s$, $t_{ss} = 1$ ve $t_{(s-1)s} = 1$ dir. Buradan $\left(T_s B + BD\right)^T = (00...011)$, $\left(CT_s + DC\right) = (00...01)$ ve $\left(CB + D^2\right) = (1)$.

$BC = \begin{pmatrix} 0 & \cdots & 0 \\ \vdots & \cdots & \vdots \\ 0 & \cdots & 1 \end{pmatrix}$. Bu durumda $T_{s+1}^2 = \begin{pmatrix} & & & & 0 \\ & A & & & \vdots \\ & & & & 1 \\ & & & & 1 \\ 0 & 0 & \cdots & 1 & 1 \end{pmatrix}$ olur. Burada A, T_s^2 matrisi gibi (s,s). bileşeninde sıfır bulunan bir matristir. T_{s+1}^2 matrisi Teorem 3.12'deki şartları sağlar. s 'nin çift olması halinde de ispat benzer şekilde yapılır.

Teorem 3.15 Bir $[2n,n,4]_2 -$kodda T_n^2 matrisi aşağıdaki şartları sağlarsa c_i ve c_j kontrol bitleri $c_i \oplus c_j$ ile değiştirilebilir [24].

i) $i.$ ve $j.$ satırların aynı koordinatlarında 1 bulunmamalıdır.

ii) $i.$ ve $j.$ satırlar silinerek yerine $i \oplus j$ satırının yazılmasıyla elde edilen satırları azaltılmış (sıkıştırılmış) T_n^2 matrisi Teorem 3.12'deki şartları sağlamalıdır.

- T_n^2 matrisine $r-$defa yukarıdaki sıkıştırma işlemi uygulanarak elde edilen yeni kodun parametreleri $[2n-r,k,4]_2$ şeklindedir.

Örnek 3.24 $n=8$ için $F=<90,150,150,90,102,90,102,90>$ olarak alalım. Teorem 3.12'den $[16,8,4]_2-$kod vardır. Teorem 3.15'den T_8^2 matrisinin satır sayısı azaltılarak bu kodun kontrol bitlerinin sayısı 8'den 5'e düşürülebilir. Yani $[13,8,4]_2-$kod üretilir. Şöyle ki;

$$T_8=\begin{pmatrix}0&1&0&0&0&0&0&0\\1&1&1&0&0&0&0&0\\0&1&1&1&0&0&0&0\\0&0&1&0&1&0&0&0\\0&0&0&0&1&1&0&0\\0&0&0&0&1&0&1&0\\0&0&0&0&0&0&1&1\\0&0&0&0&0&0&1&0\end{pmatrix},$$

$$T_8^2=\begin{pmatrix}1&1&1&0&0&0&0&0\\1&1&0&1&0&0&0&0\\1&0&1&1&1&0&0&0\\0&1&1&1&1&1&0&0\\0&0&0&0&0&1&1&0\\0&0&0&0&1&1&1&1\\0&0&0&0&0&0&0&1\\0&0&0&0&0&0&1&1\end{pmatrix}\Rightarrow\begin{cases}c_0=i_0\oplus i_1\oplus i_2\\c_1=i_0\oplus i_1\oplus i_3\\c_2=i_0\oplus i_2\oplus i_3\oplus i_4\\c_3=i_1\oplus i_2\oplus i_3\oplus i_4\oplus i_5\\c_4=i_5\oplus i_6\\c_5=i_4\oplus i_5\oplus i_6\oplus i_7\\c_6=i_7\\c_7=i_6\oplus i_7\end{cases},$$

$$\left.\begin{array}{c}c_0\oplus c_4=i_0\oplus i_1\oplus i_2\oplus i_5\oplus i_6\\c_1\oplus c_7=i_0\oplus i_1\oplus i_3\oplus i_6\oplus i_7\\c_2\oplus c_6=i_0\oplus i_2\oplus i_3\oplus i_4\oplus i_7\\c_3=i_1\oplus i_2\oplus i_3\oplus i_4\oplus i_5\\c_5=i_4\oplus i_5\oplus i_6\oplus i_7\end{array}\right\}\Rightarrow\left(T_8^2\right)'=\begin{pmatrix}1&1&1&0&0&1&1&0\\1&1&0&1&0&0&1&1\\1&0&1&1&1&0&0&1\\0&1&1&1&1&1&0&0\\0&0&0&0&1&1&1&1\end{pmatrix}.$$

Örnek 3.25 $n=8$ için $F=<90,150,90,90,90,90,150,90>$ kural vektörü ile $d=5$ minimum uzaklığa sahip kod üretelim. $k=d-2\Rightarrow k=5-2=3$ tür.

$$T_8=\begin{pmatrix}0&1&0&0&0&0&0&0\\1&1&1&0&0&0&0&0\\0&1&0&1&0&0&0&0\\0&0&1&0&1&0&0&0\\0&0&0&0&0&1&0&0\\0&0&0&0&1&0&1&0\\0&0&0&0&0&1&1&1\\0&0&0&0&0&0&1&0\end{pmatrix}\Rightarrow T_8^3=\begin{pmatrix}1&1&1&1&0&0&0&0\\1&1&0&1&1&0&0&0\\1&0&1&1&0&1&0&0\\1&1&1&0&1&0&1&0\\0&1&0&1&0&1&1&1\\0&0&1&0&1&1&0&1\\0&0&0&1&1&0&1&1\\0&0&0&0&1&1&1&1\end{pmatrix}.$$

Teorik olarak $8-$bilgi bitine $8-$kontrol biti ekleyerek $d=5$ minimum uzaklığa sahip kod üretmek mümkün değildir. Bunun için T_8^3 matrisine $\begin{pmatrix}1&0&0&0&0&0&0&1\end{pmatrix}$ satırı eklenerek $[17,8,5]_2-$kod elde edilir. Bu durumda kontrol bitlerinin sayısı bilgi bitlerinin sayısından fazladır.

- Hücresel dönüşümlerle yapılan ve $8-$bilgi biti içeren kod için $11-$tane iki girişli XOR mantık kapısı gerekirken, geleneksel Hamming kodu aynı uzunluk için $21-$tane iki girişli XOR mantık kapısı gerektirir.

3.2.3 Keyfi t-Uzaklığa Sahip Kodun Üretilmesi

Bu durumda kontrol bitlerinin sayısı bilgi bitlerinin sayısından fazladır. $I\rightarrow$bilgi bitleri, $T^{p_i}[I]\rightarrow$kontrol bitleri olmak üzere; $CW=\left(I,T^{p_1}[I],T^{p_2}[I],...,T^{p_k}[I]\right)$ kod sözdür. Buradaki $T^{p_i}[I]$ lerin her biri Teorem 3.12'deki şartları sağlamalıdır.

Örnek 3.26 $F=\langle 90,150,150,90,60,90,60,90\rangle$ kural vektörüne sahip hücresel dönüşüm ile $d=9$ uzaklığa sahip kod üretelim. Kontrol bitleri $\left(T_8^2[I],\ T_8^3[I],\ T_8^5[I],\ T_8^9[I]\right)$ şeklinde üretilirse $8-$bilgi biti ve $32-$kontrol biti olan kod sözlerden oluşan kodun minimum uzaklığı 9'dur. $32-$kontrol biti sıkıştırılmak suretiyle minimum uzaklık korunarak kontrol bitlerinin sayısı 18'e kadar düşürülebilir.

3.2.4 Hücresel Dönüşümlerle Bit Hata Düzelten Kodlarda Dekodlama

$I=\left(i_1,i_2,...,i_k\right)$ $k-$ bitten oluşan hatasız söz,

$C=\left(c_{k+1},c_{k+2},...,c_n\right)$ $n-k$ bitten oluşan hatasız kontrol haneleri,

$I^{'}=\left(i_1^{'},i_2^{'},...,i_k^{'}\right)$ $k-$ bitten oluşan hatalı söz,

$C^{'}=\left(c_{k+1}^{'},c_{k+2}^{'},...,c_n^{'}\right)$ $n-k$ bitten oluşan hatalı kontrol haneleri,

$E=\left(I_e,C_e\right)=\left(e_1,e_2,...,e_k,e_{k+1},e_{k+2},...,e_n\right)$ hata vektörü ve I_n, $n\times n$ tipinde birim matris olsun.

$$H'=\begin{cases}\left(T_k^p\right)^*, & n-k<k\\ \left(T_k^p\right), & n-k=k\\ \\ \begin{pmatrix}T_k^p\\ H_0\end{pmatrix}, & n-k>k\end{cases} \tag{3.5}$$

H_0 fazla kontrol bitlerini üretmesi için $\left(T_k^p\right)$ matrisine eklenen alt matrisi ve $\left(T_k^p\right)^*$ da $\left(T_k^p\right)$ matrisinin sıkıştırılmış halini göstermektedir. Dekodlama için ilk adım olarak $n-k$ bitten oluşan sendrom hesaplanır.

$S=\left[H^{'}\right]\left[I^{'}\right]\oplus\left[C\right]$ veya $H=\left[H^{'}\mid I_{n-k}\right]$ olmak üzere; $S=\left[\mathrm{H}\right]\begin{bmatrix}I'\\ C'\end{bmatrix}$, $I^{'}=I\oplus I_e$ ve $C^{'}=C\oplus C_e$ olduğundan

$$S=\left[\mathrm{H}\right]\begin{bmatrix}I'\\ C'\end{bmatrix}=\left[\mathrm{H}\right]\begin{bmatrix}I\oplus I_e\\ C\oplus C_e\end{bmatrix}=\left[\mathrm{H}\right]\begin{bmatrix}I_e\\ C_e\end{bmatrix} \tag{3.6}$$

Eğer hata yoksa sendrom sıfırdır. Aksi halde hata vektörünü bulmak için $n-$ bilinmeyen ve $\left(n-k\right)$ denklemden oluşan sendrom denklemlerinin çözülmesi gerekir. Ancak bu denklem sisteminin birden fazla çözümü vardır. Bu sebeple mümkün bütün hata vektörlerini hesaplamamız gerekir. Daha sonra bu hata vektörleri alınan sendrom ile karşılaştırılır. Bu şekilde hata vektörü bulunabilir. Yani $t'-$ tane hata düzeltebilen kod

için $w(E) \le t'$ olan ve $\le \sum_{i=1}^{t'} \binom{n}{i}$ sayıda bulunan mümkün bütün hata vektörlerinin hesaplanması gerekir ki bu da $O\left(n^{t'}\right)$ karmaşıklık (complexity) derecesine sahiptir. Dolayısıyla bu karmaşık durumundan kurtulmak için hata vektörünü doğrudan sendromdan elde etmemize olanak tanıyan ve kodlamada kullanılan hücresel dönüşümlerin karakteristik matrislerinin tersinirlik özelliğinden faydalanan iki tane dekodlama algoritması vereceğiz.

$[H][E]=[S]$ iken eğer H'nin tersi varsa yani H regüler karesel bir matris ise, $[H]^{-1}[S]=[E]$ olacak şekilde H^{-1} matrisi vardır.

3.2.4.1 Dekodlama–1

$n-k \ge k$ yani kontrol bitlerinin sayısı bilgi bitlerinin sayısından büyük ya da eşit olan kodlara uygulanabilir. H matrisi aşağıdaki gibi alt matrislere ayrılabilir.

$$[H]_{(n-k)\times n} = \left[\begin{array}{c|c} T_k^p & [I_{n-k}] \\ [H_0]_{(n-2k)\times k} & \end{array}\right] \Rightarrow \left[\begin{array}{c|c} T_k^p & [I_{n-k}] \\ [H_0]_{(n-2k)\times k} & \end{array}\right][E] = S \qquad (3.7)$$

$[H][E]$ matris çarpımı aşağıdaki adımları içerir.

1. T_k^p ile $[E]_{n\times 1}$ in ilk k koordinatı çarpılır ve $[S_1]_{k\times 1}$ elde edilir.
2. $[I_{n-k}]$ ile $[E]_{n\times 1}$ in alt $(n-k)$ koordinatı çarpılır ve $[S_2]_{(n-k)\times 1}$ elde edilir.
3. $[S_1]_{k\times 1}$ ile $[S_2]_{(n-k)\times 1}$ in ilk k koordinatı bileşen bileşen toplanarak $[S]_{(n-k)\times 1}$ in ilk k koordinatı elde edilir.
4. $[H_0]_{(n-2k)\times k}$ ile $[E]_{n\times 1}$ in ilk k koordinatı çarpılır ve elde edilen vektör $[S_2]_{(n-k)\times 1}$ nin kalan $(n-2k)-$uzunluğundaki vektör ile toplanarak $[S]_{(n-k)\times 1}$ nın son $(n-2k)$ bitlik kısmı hesaplanır.

- Eğer yukarıdaki işlemlerin her biri tersinir ise bu durumda $[H]^{-1}[S]$ bu işlemlere denktir.

Dekodlama Algoritması–1

1. k bitten oluşan S_{aug} eklemeli sendromunu, $n-k$ bitten oluşan S bilgi bitlerinin ürettiği sendrom ile bitiştir.
2. $\left[I_{n-k}\right]^{-1}=\left[I_{n-k}\right]$ olduğundan $\left[I_{n-k}\right]\left[S_{aug}\right]$ yerine $S_{aug}\oplus S_k$ şeklinde sendromun ilk k bitini hesapla.
3. $\left[T_k^{-p}\right]\left[S_k\right]$ ifadesini hesapla.
4. $\left[H_0\right]_{(n-2k)\times k}\left[S_k\right]$ ifadesini hesapla ve S'nin geriye kalan $(n-2k)$ bitiyle topla daha sonra bu $(n-2k)$ bit ile k bitten oluşan S_{aug}'u bitiştir (concatenate). Eğer $n=2k$ ise bu adımı atla.
5. 3. adımda güncellenen $\left[S_k\right]=\left[I_e\right]$ ve 4. adımdaki $\left[S_{aug},S_{n-2k}\right]=\left[C_e\right]$ dir.

$I=I'+I_e$ ve $C=C'+C_e$ alarak sendromu yeniden hesapla eğer sendrom sıfır ise $\left[E\right]_{n\times 1}=\left[I_e,C_e\right]$ hata vektörüdür. Aksi halde S_{aug} eklemeli sendromu mümkün başka bir hata vektörü olarak seç ve algoritmadaki adımları tekrar izle.

Örnek 3.27 $n=8$ için $F=<90,150,90,90,90,90,150,90>$ kural vektörü ile $d=5$ minimum uzaklığa sahip kod üretebilmek için T_8^3 matrisine $H_0=\begin{pmatrix}1&1&1&1&1&1&1&1\end{pmatrix}_{1\times 8}$ satırını ekleyelim (bkz. Örnek 3.25). Bu durumda

$$T_8^3=\begin{pmatrix}1&1&1&1&0&0&0&0\\1&1&0&1&1&0&0&0\\1&0&1&1&0&1&0&0\\1&1&1&0&1&0&1&0\\0&1&0&1&0&1&1&1\\0&0&1&0&1&1&0&1\\0&0&0&1&1&0&1&1\\0&0&0&0&1&1&1&1\end{pmatrix}\Rightarrow\begin{pmatrix}T_8^3\\H_0\end{pmatrix}=\begin{pmatrix}1&1&1&1&0&0&0&0\\1&1&0&1&1&0&0&0\\1&0&1&1&0&1&0&0\\1&1&1&0&1&0&1&0\\0&1&0&1&0&1&1&1\\0&0&1&0&1&1&0&1\\0&0&0&1&1&0&1&1\\0&0&0&0&1&1&1&1\\1&1&1&1&1&1&1&1\end{pmatrix}_{9\times 8}$$

olacağından

$$[H]_{(n-k)\times n}=\left[\begin{array}{c|c} T_k^p & \\ [H_0]_{(n-2k)\times k} & [I_{n-k}] \end{array}\right]$$

$$=\begin{pmatrix} 1&1&1&1&0&0&0&0&1&0&0&0&0&0&0&0&0\\ 1&1&0&1&1&0&0&0&0&1&0&0&0&0&0&0&0\\ 1&0&1&1&0&1&0&0&0&0&1&0&0&0&0&0&0\\ 1&1&1&0&1&0&1&0&0&0&0&1&0&0&0&0&0\\ 0&1&0&1&0&1&1&1&0&0&0&0&1&0&0&0&0\\ 0&0&1&0&1&1&0&1&0&0&0&0&0&1&0&0&0\\ 0&0&0&1&1&0&1&1&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&1&0\\ 1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&1 \end{pmatrix}_{9\times 17}$$

$I=10100100\in\mathbb{F}_2^8$ sözünü kodlayalım. Bu durumda $CW=\left(I,\begin{bmatrix}T_8^3\\ H_0\end{bmatrix}[I]\right)$

$=10100100011010011\in\mathbb{F}_2^{17}$ kod söz olur. CW kod sözünü gönderelim ve CW' alınan hatalı söz olsun. Bu halde üç durum söz konusudur.

1. Hata sadece bilgi bitlerinde oluşabilir.
2. Hata sadece kontrol bitlerinde oluşabilir.
3. Hem bilgi hem de kontrol bitleri hatalı olabilir.

1. Durum: Hata sadece bilgi bitlerinde meydana gelmiş olsun ve $CW'=1\hat{1}1001\hat{1}0011010011$ alınan hatalı söz olsun. Bu durumda $[H]_{9\times 17}\begin{bmatrix}I'\\ C'\end{bmatrix}=[S]=110000110$ olur. İlk devirde $S_{aug}=00000000$ alalım. O halde

$S_8=S\oplus S_{aug}=11000011$

$I_e=T_8^{-3}[S_8]=01000010$

$S_{n-2k}=[H_0]_{1\times 8}[S_8]\oplus S_{n-2k}=0$

$I'=I'\oplus I_e=10100100$

$C'=C'\oplus(S_{aug},S_{n-2k})=011010011$

$E=[I_e,S_{aug},S_{n-2k}]=[01000010000000000]$ olarak bulunur.

2. Durum: Hata sadece kontrol bitlerinde meydana gelmiş olsun ve $CW'=10100100011\hat{1}\hat{0}0011$ alınan hatalı söz olsun. Bu durumda $S=[H]_{9\times17}\begin{bmatrix} I' \\ C' \end{bmatrix}=000110000$ olur. $S_{aug}=00011000$ olarak alalım. O halde

$S_8=S_8\oplus S_{aug}=00000000$

$S_{n-2k}=[H_0]_{1\times8}[S_8]\oplus S_{n-2k}$

$I=I'\oplus I_e=10100100$

$C'=C'\oplus(S_{aug},S_{n-2k})=011010011$

$E=[I_e,S_{aug},S_{n-2k}]=[0000000000110000]$ olarak bulunur.

3. Durum: Hata hem bilgi bitlerinde hem de kontrol bitlerinde meydana gelmiş olsun ve $CW'=10100\hat{1}00110010010\hat{0}$ alınan hatalı söz olsun. Bu durumda $S=[H]_{9\times17}\begin{bmatrix} I' \\ C' \end{bmatrix}=000110111$ olur. İlk devirde $S_{aug}=00000000$ alalım. O halde

$S_8=S\oplus S_{aug}=00011011$

$I_e=[T_8^{-3}][S_8]=00000010$

$S_{n-2k}=[H_0]_{1\times8}[S_8]\oplus S_{n-2k}=1$

$I'=I'\oplus I_e=10100100$

$C'=C'\oplus(S_{aug},S_{n-2k})=011010011$

$E=[I_e,S_{aug},S_{n-2k}]=[0000001000000001]$ olarak bulunur.

3.2.4.2 Dekodlama–2

Buradaki temel amaç hata vektörünü doğrudan sendromdan hesaplamaktır. Ancak her zaman bir sendroma karşılık tek hata vektörü gelmeyebilir. Bu kodun hata düzeltme kapasitesiyle ilgili bir durumdur. Örneğin; iki hata düzeltebilen bir kodda ağırlığı 1 ve 2 olan tüm hata vektörleri tek türlü belirlidir.

$$\left[H\right]_{(n-k)\times n}=\left[\begin{array}{c|c} T_k^p & \\ \left[H_0\right]_{(n-2k)\times k} & \left[I_{n-k}\right] \end{array}\right] \tag{3.8}$$

matrisini determinantı sıfırdan farklı olacak şekilde tersinir karesel bir matris yapmak için H matrisine $k-$tane satır ekleyelim. Bu işlemi yaparken 1 lerin sayısının az olması istenen bir durumdur. Bu yeni matris $\left[T_{aug}\right]\Rightarrow \det\left[T_{aug}\right]=1$. Sendrom ile hata vektörü arasındaki ilişki;

$$\left[T_{aug}\right]=\begin{bmatrix}\left[H\right]_{(n-k)\times n}\\ \left[A\right]_{k\times n}\end{bmatrix}_{n\times n},\ E=\begin{bmatrix}I_e\\ C_e\end{bmatrix}_{n\times 1} \tag{3.9}$$

$$\left[T_{aug}\right]\left[E\right]=\begin{bmatrix}\left[H\right]_{(n-k)\times n}\\ \left[A\right]_{k\times n}\end{bmatrix}_{n\times n}\begin{bmatrix}I_e\\ C_e\end{bmatrix}_{n\times 1}=\begin{bmatrix}S\\ S_{aug}\end{bmatrix}_{n\times 1} \tag{3.10}$$

şeklindedir.

Dekodlama Algoritması–2

1. H matrisine $k-$tane satır ekleyerek regüler karesel $\left[T_{aug}\right]$ matrisini oluştur.
2. Bütün mümkün hata vektörleri için S ile S_{aug} arasındaki ilişkiyi tablo haline getir.
3. Girdileri $n-k$ bitlik S olan ve çıktısı k bitlik S_{aug} olan PLA'yı düzenle.
4. $\left[T_{aug}\right]^{-1}$ matrisini hesapla.
5. Alınan bilgi ve kontrol bitleri için hata vektörünü doğrudan sendromdan hesapla.

Örnek 3.28 $F=\langle 90,150,150,90\rangle$ Wolfram kurallarından oluşan global geçiş fonksiyonuna sıfır sınır şartı altında karşılık gelen matris;

$$T=\begin{pmatrix}0&1&0&0\\1&1&1&0\\0&1&1&1\\0&0&1&0\end{pmatrix}_{4\times4}\Rightarrow T^2=\begin{pmatrix}1&1&1&0\\1&1&0&1\\1&0&1&1\\0&1&1&1\end{pmatrix}_{4\times4}$$

dir. Bu durumda

$$H=\left[T^2\mid I_4\right]=\begin{pmatrix}1&1&1&0&1&0&0&0\\1&1&0&1&0&1&0&0\\1&0&1&1&0&0&1&0\\0&1&1&1&0&0&0&1\end{pmatrix},$$

$$\left[T_{aug}\right]=\begin{bmatrix}[H]_{4\times8}\\[A]_{4\times8}\end{bmatrix}_{8\times8}=\begin{pmatrix}1&1&1&0&1&0&0&0\\1&1&0&1&0&1&0&0\\1&0&1&1&0&0&1&0\\0&1&1&1&0&0&0&1\\0&0&1&1&1&0&1&0\\0&0&0&0&0&0&1&0\\0&1&1&0&1&0&0&1\\1&1&1&0&1&0&0&1\end{pmatrix},$$

$$\text{ve }\left[T_{aug}\right]^{-1}=\begin{pmatrix}0&0&0&0&0&0&1&1\\1&0&1&1&0&1&1&0\\0&0&0&1&1&1&1&0\\0&0&1&1&1&0&0&1\\0&0&1&1&1&0&1&0\\0&0&1&0&1&0&1&1\\0&0&0&0&0&1&0&0\\1&0&0&0&0&0&0&1\end{pmatrix}$$

şeklindedir.

Hata vektörü	S	S_{aug}
00000000	0000	0000
10000000	1110	0110
01000000	1101	0000
00100000	1011	0000
00010000	0111	1000
00001000	1000	1000
00000100	0100	0011
00000010	0010	0101
00000001	0001	1001

Tablo 3. 2 S ile S_{aug} arasındaki ilişkiyi gösteren sendrom tablosu

Yukarıda verilen kod bir $[8,4,4]_2 -$kod olup $1-$hata düzeltebilir. Mümkün hata vektörleri $\binom{8}{1}=8$ tanedir ve $\left[T_{aug}\right]$ ile $\left[T_{aug}\right]^{-1}$ kullanılarak S ile S_{aug} arasındaki ilişki Tablo 3.2'de gösterilmiştir.

3.3 Hücresel Dönüşümlerle Bayt Hata Düzelten Kodlar

T, $b-$uzunluğundaki bir boyutlu lineer hücresel dönüşümün global geçiş fonksiyonuna karşılık gelen tersinir temsili matris olsun. T'nin üreteceği devir uzunluğu N olmak üzere, $N \le 2^b - 1$ dir. Eğer $N = 2^b - 1$ ise bu hücresel dönüşüme maksimal devir uzunluğuna sahip hücresel dönüşüm denir. Hücresel dönüşümler ile bayt hata düzelten kodların çıkış noktası Reed-Solomon kodlardır. Hücresel dönüşümlerle bayt hata düzelten kodlarda kod sözlerin her biri $k-$tane bitten oluşan $N-$tane bilgi bayt ve $t-$bayt hata düzeltebilmesi için $(2t+1)-$tane kontrol baytından oluşur.

3.3.1 Kodlama

T, $b-$uzunluğundaki bir boyutlu lineer hücresel dönüşümün global geçiş fonksiyonuna karşılık gelen N devir uzunluğuna sahip tersinir temsili matris olsun. $B=\left[B_0,B_1,...,B_{N-1}\right]$ ile her biri $b-$bitten oluşan $N-$tane bilgi baytını göstersin. $t-$bayt hata düzeltebilmesi için $\left(2t+1\right)-$tane kontrol baytı, $C=\left[C_0,C_1,...,C_{2t}\right]$ olsun. Bu kontrol baytlarının her biri $i=0,1,2,...,2t+1$ olmak üzere;

$$C_i=T^{i(N-1)}\left[B_0\right]\oplus T^{i(N-2)}\left[B_1\right]\oplus T^{i(N-3)}\left[B_2\right]\oplus ...\oplus T^{i2}\left[B_{N-3}\right]\oplus T^{i}\left[B_{N-2}\right]\oplus B_{N-1} \qquad (3.11)$$

şeklinde üretilir.

3.3.1.1 Bir Bayt Hata Düzelten-İki Bayt Hata Fark Eden Kod

Bir bayt hata düzeltebilmesi için her kod söze üç tane kontrol baytı eklenmelidir. Bu kontrol baytları (3.11) ifadesinden aşağıdaki gibidir:

$$C_0=B_0\oplus B_1\oplus ...\oplus B_{N-1}, \qquad (3.12)$$

$$C_1=T^{N-1}\left[B_0\right]\oplus T^{N-2}\left[B_1\right]\oplus ...\oplus T\left[B_{N-2}\right]\oplus B_{N-1}, \qquad (3.13)$$

$$C_2=T^{2(N-1)}\left[B_0\right]\oplus T^{2(N-2)}\left[B_1\right]\oplus ...\oplus T^{2}\left[B_{N-2}\right]\oplus B_{N-1}. \qquad (3.14)$$

Bu durumda $CW=\left[B_0,B_1,...,B_{N-1},C_0,C_1,C_2\right]$ bir kod sözdür. Bu kodun kontrol matrisi aşağıdaki gibidir:

$$H=\begin{pmatrix} I & I & I & I & \cdots & I & I & O & O \\ I & T & T^2 & T^3 & \cdots & T^{N-1} & O & I & O \\ I & T^2 & T^4 & T^6 & \cdots & T^{2(N-1)} & O & O & I \end{pmatrix}.$$

Genel olarak $t-$bayt hata düzelten kodlar için kontrol matrisi;

$$H=\begin{pmatrix} I & I & I & I & \cdots & I & I & O & O & \cdots & O \\ I & T & T^2 & T^3 & \cdots & T^{N-1} & O & I & O & \cdots & O \\ I & T^2 & T^4 & T^6 & \cdots & T^{2(N-1)} & O & O & I & \cdots & O \\ \vdots & & & & & & & & & & \vdots \\ I & T^{2t} & T^{4t} & T^{6t} & \cdots & T^{2t(N-1)} & O & O & O & \cdots & I \end{pmatrix}$$

şeklindedir. Burada T, $b\times b$ tipinde tersinir temsili matris; I, $b\times b$ tipinde birim matris ve O, $b\times b$ tipinde sıfır matristir.

Dekodlama

Tanım 3.22 (Hata sendromu) C_i, i. kontrol baytı ve $C_i^{'}$ de alınan bilgi baytlarından tekrar hesaplanan i. kontrol baytı (hatalı olabilir) olsun. Bu durumda i. kontrol baytına karşılık gelen hata sendromu $0 \le \mathrm{i} \le 2\mathrm{t}$ olmak üzere;

$$S_i = C_i \oplus C_i^{'} \tag{3.15}$$

şeklinde hesaplanır.

Tanım 3.23 (Hata baytı) B_{N-1-j} ve $B_{N-1-j}^{'}$ sırasıyla gönderilen ve alınan (Soldan sayıldığında j. bayt, sağdan sayıldığında $\left(N-1-j\right)$. bayta eşittir.) bilgi baytları olsun. Bu durumda $j \ge 0$ için

$$E_j = B_{N-1-j} \oplus B_{N-1-j}^{'} \tag{3.16}$$

şeklindeki ifadeye j. hata baytı denir.

Hata düzeltmede temel amaç alınan $B^{'} = \left[B_0^{'}, B_1^{'}, ..., B_{N-1}^{'}\right]$ bilgi baytları ve $C^{'} = \left[C_0^{'}, C_1^{'}, ..., C_{2t}^{'}\right]$ kontrol baytlarından $E = \left[E_0, E_1, ..., E_{N-1}\right]$ hata vektörünü hesaplamaktır. Bu durumda

$$\left[B\right] = \left[B^{'}\right] \oplus \left[E\right]. \tag{3.17}$$

Soldan i. ve j. baytlarda hata oluşsun ve bu hatalara karşılık gelen hata vektörleri sırasıyla E_i ve E_j olsun. O halde kontrol baytları tekrar aşağıdaki gibi üretilir.

$$C_0^{'} = B_0^{'} \oplus B_1^{'} \oplus ... \oplus B_{N-1}^{'} \oplus E_i \oplus E_j \tag{3.18}$$

$$C_1^{'} = T^{N-1}\left[B_0^{'}\right] \oplus T^{N-2}\left[B_1^{'}\right] \oplus ... \oplus T\left[B_{N-2}^{'}\right] \oplus B_{N-1}^{'} \oplus T^{i}\left[E_i\right] \oplus T^{j}\left[E_j\right] \tag{3.19}$$

$$C_2^{'} = T^{2(N-1)}\left[B_0^{'}\right] \oplus T^{2(N-2)}\left[B_1^{'}\right] \oplus ... \oplus T^{2}\left[B_{N-2}^{'}\right] \oplus B_{N-1}^{'} \oplus T^{2i}\left[E_i\right] \oplus T^{2j}\left[E_j\right] \tag{3.20}$$

Yeniden üretilen kontrol baytları $C_0^{'}$, $C_1^{'}$, $C_2^{'}$ olmak üzere, hata sendromları;

$$S_0 = C_0 \oplus C_0^{'} = E_i \oplus E_j \tag{3.21}$$

$$S_1 = C_1 \oplus C_1^{'} = T^i \left[E_i\right] \oplus T^j \left[E_j\right] \tag{3.22}$$

$$S_2 = C_2 \oplus C_2^{'} = T^{2i} \left[E_i\right] \oplus T^{2j} \left[E_j\right] \tag{3.23}$$

olarak hesaplanır.

Dekodlama Algoritması–1

1. Eğer S_0, S_1 ve S_2 sıfır ise hata yoktur.
2. Eğer S_0, S_1 ve S_2 sendromlarından sadece bir tanesi sıfırdan farklı ancak diğer iki tanesi sıfır ise bu durumda sıfırdan farklı sendromu üreten kontrol baytı hatalıdır.
3. Birden fazla sendrom sıfırdan farklı ise,

   $T^i \left[S_0\right] = S_1$ ve $T^{2i} \left[S_0\right] = S_2, \quad \left(0 \le i \le N-1\right)$

   olacak şekilde bir i varsa $\left(N-1-i\right)$. baytta hata vardır ve hata vektörü $\left[S_0\right]$ dır. Aksi halde birden fazla hata meydana gelmiştir. Yani düzeltilemez hata vardır.

Teorem 3.16 Dekodlama Algoritması–1 bir bayt hata düzelten ve iki bayt hata fark eden kodları doğru bir şekilde dekodlar.

İspat: Eğer bir bayt hatalıysa hata ya bilgi baytlarında ya da kontrol baytlarındadır. Kontrol baytlarından biri hatalıysa hatalı kontrol baytının ürettiği sendrom sıfırdan farklı iken diğer iki sendrom sıfırdır. Eğer bu bir hata bilgi baytlarından birinde meydana gelmişse algoritmanın üçüncü adımı bu hatayı düzeltir. Çünkü T matrisi regülerdir ve sıfırdan farklı bir konfigürasyonu $T^i, \left(i = 0,1,2,...,N-1\right)$ ile çarptığımızda elde edilen yeni konfigürasyon da sıfırdan farklı olacaktır. O halde bilgi baytlarında hata meydana gelmesi halinde tüm sendromlar sıfırdan farklıdır.

Eğer iki bayt hatalıysa üç durum söz konusudur. Hataların ikisi kontrol baytlarında meydana gelmiş olabilir ki bu durumda iki sendrom sıfırdan farklıdır. Bu halde iki hata

fark edilmiş olur. İkinci olarak iki hata bilgi baytlarında meydana gelmiş olabilir. Bu durumun bir hata meydana gelmesi hali ile çakışmadığını gösterelim. $(N-1-i)$ ve $(N-1-j)$. bilgi baytlarında hata meydana gelmiş olsun ve sırasıyla E_i ile E_j hata vektörleri olsun. $(N-1-k)$. bilgi baytı bir hatanın meydana geldiği durumu ve E_k da hata vektörünü göstersin. Bu durumda sendromlar aşağıdaki denklemleri sağlamalıdır.

$$S_0 = C_0 \oplus C_0^{'} = E_i \oplus E_j = E_k \tag{3.24}$$

$$S_1 = C_1 \oplus C_1^{'} = T^i\left[E_i\right] \oplus T^j\left[E_j\right] = T^k\left[E_k\right] \tag{3.25}$$

$$S_2 = C_2 \oplus C_2^{'} = T^{2i}\left[E_i\right] \oplus T^{2j}\left[E_j\right] = T^{2k}\left[E_k\right] \tag{3.26}$$

(3.24) ve (3.25) denklemlerinden

$$T^i\left[E_i\right] \oplus T^j\left[E_j\right] = T^k\left[E_i \oplus E_j\right] \Rightarrow \left(T^i \oplus T^k\right)\left[E_i\right] = \left(T^k \oplus T^j\right)\left[E_i \oplus E_j\right] = E \tag{3.27}$$

ve (3.26) denkleminden

$$T^{2k}\left[E_k\right] = T^k\left[T^i\left[E_i\right] \oplus T^j\left[E_j\right]\right] = T^{2i}\left[E_i\right] \oplus T^{2j}\left[E_j\right] \tag{3.28}$$

gerekli düzenlemelerden sonra

$$\left(T^{i+k} \oplus T^{2i}\right)\left[E_i\right] = \left(T^{2j} \oplus T^{j+k}\right)\left[E_j\right] \Rightarrow T^i\left(T^k \oplus T^i\right)\left[E_i\right] = T^j\left(T^j \oplus T^k\right)\left[E_j\right] \tag{3.29}$$

(3.27) denklemi göz önünde bulundurulursa;

$$T^i\left[E\right] = T^j\left[E\right] \tag{3.30}$$

Bu bir çelişkidir. Çünkü biz hata yerlerini farklı kabul ettik. Sonuç olarak iki hata meydana gelmesi hali ile bir hata meydana gelmesi halinde elde edilen sendromlar birbirinden farklıdır.

Hataların biri bilgi baytlarında diğeri kontrol baytlarında oluşmuşsa yukarıdaki gibi bu durumun bir hata oluşması halinden farklı sendrom ürettiği gösterilebilir.

3.3.1.2 İki Bayt Hata Düzelten-İki Bayt Hata Yerleştiren Kod

İki hata düzeltebilmek ve iki hata meydana gelmesi halinde hataların yerlerini belli baytlar arasına kısıtlamak (hata yerleştirme) için yukarıda verilen bir hata durumunda gerekli üç kontrol baytına bir tane daha bayt eklenmelidir. Eğer hataların ikisi sadece bilgi ya da kontrol baytlarında meydana gelmişse bu durumda hatalar hem fark edilir hem de düzeltilir. Ancak hataların biri bilgi baytlarında diğeri kontrol baytlarında meydana gelmişse hata en çok üç bilgi baytı arasına yerleştirilir. Bu durumda kontrol baytları aşağıdaki gibidir:

$$C_0 = B_0 \oplus B_1 \oplus ... \oplus B_{N-1} \tag{3.31}$$

$$C_1 = T^{N-1}\left[B_0\right] \oplus T^{N-2}\left[B_1\right] \oplus ... \oplus T\left[B_{N-2}\right] \oplus B_{N-1} \tag{3.32}$$

$$C_2 = T^{2(N-1)}\left[B_0\right] \oplus T^{2(N-2)}\left[B_1\right] \oplus ... \oplus T^2\left[B_{N-2}\right] \oplus B_{N-1} \tag{3.33}$$

$$C_3 = T^{3(N-1)}\left[B_0\right] \oplus T^{3(N-2)}\left[B_1\right] \oplus ... \oplus T^3\left[B_{N-2}\right] \oplus B_{N-1} \tag{3.34}$$

Hata sendromları ise,

$$S_0 = C_0 \oplus C_0^{'} = E_i \oplus E_j = E_k \oplus E_r \tag{3.35}$$

$$S_1 = C_1 \oplus C_1^{'} = T^i\left[E_i\right] \oplus T^j\left[E_j\right] = T^k\left[E_k\right] \oplus T^r\left[E_r\right] \tag{3.36}$$

$$S_2 = C_2 \oplus C_2^{'} = T^{2i}\left[E_i\right] \oplus T^{2j}\left[E_j\right] = T^{2k}\left[E_k\right] \oplus T^{2r}\left[E_r\right] \tag{3.37}$$

$$S_3 = C_3 \oplus C_3^{'} = T^{3i}\left[E_i\right] \oplus T^{3j}\left[E_j\right] = T^{3k}\left[E_k\right] \oplus T^{3r}\left[E_r\right] \tag{3.38}$$

şeklinde hesaplanır.

Dekodlama Algoritması–2

1. Eğer sendrom baytlarının tamamı sıfır ise alınan bilgi baytları hatasızdır.
2. Eğer tüm sendrom baytları sıfırdan farklı ve

   $$T^i\left[S_2\right] \oplus S_3 = T^{2j}\left[T^i\left[S_0\right] \oplus S_1\right] \tag{3.39}$$

   şartını sağlayan i ve j, $\left(i, j \le N\right)$ varsa ya da

$$T^j\left[T^i\left[S_0\right]\oplus S_1\right]=T^i\left[S_1\right]\oplus S_2 \tag{3.40}$$

$$T^j\left[T^{2i}\left[S_0\right]\oplus S_2\right]=T^{2i}\left[S_1\right]\oplus S_3 \tag{3.41}$$

şartları sağlanıyorsa i. ve j. baytlar hatalıdır.

3. Eğer $T^{i+j}+T^{i+k}+T^{j+k}=0$ veya $T^i+T^j+T^k=0$ olacak şekilde bir k sayısı varsa (5)'e git ve hata yerini belirle.

4. j. baytın E_j hata vektörünü aşağıdaki gibi hesapla:

$$E_j=T^{-x}\left[T^i\left[S_0\right]\oplus S_1\right]=T^{N-x}\left[T^i\left[S_0\right]\oplus S_1\right] \tag{3.42}$$

Burada $T^x=T^i+T^j$ dir. i. baytın hata vektörü E_i'yi aşağıdaki gibi hesapla:

$$E_i=E_j\oplus S_0 \tag{3.43}$$

5. İki hata ya i. ve j. bilgi baytlarında ya da biri k. bilgi baytında diğeri ise kontrol baytlarındadır.

Teorem 3.17 Dekodlama Algoritması–2 iki bayt hata düzelten ve iki bayt hata fark eden kodları doğru bir şekilde dekodlar.

İspat: İki hata kontrol baytlarında meydana gelmiş olsun. Bu durumda hatalı kontrol baytlarına karşılık gelen iki sendrom sıfırdan farklıdır. Şimdi hataların ikisi bilgi baytlarında meydana gelmiş olsun. O halde sendromların tamamı sıfırdan farklıdır. İki ayrı bayt çiftinde meydana gelen hataların aynı sendromu üretmeyeceğini gösterelim. $\left(N-1-i\right)$, $\left(N-1-j\right)$, $\left(N-1-k\right)$ ve $\left(N-1-r\right)$ bilgi baytlarında hata meydana gelmiş olsun ve sırasıyla E_i, E_j, E_k ve E_r hata vektörleri olsun. İlk olarak E_i ve E_j hata vektörü çiftine karşılık gelen sendromlar, E_k ve E_r hata vektörlerininki ile aynı olsun. (3.35)'teki eşitliğin her iki tarafına T^j uygulanır ve (3.36) ile taraf tarafa toplanırsa,

$$\underbrace{\left(T^i\oplus T^j\right)\left[E_i\right]}_{E_1}=\underbrace{\left(T^k\oplus T^j\right)\left[E_k\right]}_{E_2}\oplus\underbrace{\left(T^r\oplus T^j\right)\left[E_r\right]}_{E_3} \tag{3.44}$$

(3.36)'daki eşitliğin her iki tarafına T^j uygulanır ve (3.37) ile taraf tarafa toplanırsa,

$$T^i\left[E_1\right]=T^k\left[E_2\right]\oplus T^r\left[E_3\right] \tag{3.45}$$

Benzer şekilde (3.37)'deki eşitliğin her iki tarafına T^j uygulanır ve (3.38) ile taraf tarafa toplanırsa,

$$T^{2i}\left[E_1\right]=T^{2k}\left[E_2\right]\oplus T^{2r}\left[E_3\right] \tag{3.46}$$

Aynı işlemler (3.44) ve (3.45)'teki eşitliklere uygulanırsa;

$$\left(T^i\oplus T^k\right)\left[E_1\right]=\left(T^r\oplus T^k\right)\left[E_3\right]=E_4 \tag{3.47}$$

(3.45) ve (3.46) eşitliklerinden

$$T^i\left[E_4\right]=T^r\left[E_4\right] \tag{3.48}$$

yazılır. Buradan $i=yN+r,\left(y\in\mathbb{N}\right)$ şeklinde olduğu anlaşılır. b bir baytta bulunan bitlerin sayısı olmak üzere, $N=2^b-1$ hücresel dönüşümün devir uzunluğudur. O halde $y=0$ ya da $i=r$ dir. Benzer şekilde $j=k$ olduğu gösterilebilir. Yani farklı hata çiftleri aynı sendromları üretmezler.

Şimdi de hata vektörlerinin tek türlü hesaplandığını gösterelim. $\left(N-1-i\right).$ ve $\left(N-1-j\right).$ baytlar hatalı olsun. Bu durumda sendrom denklemleri aşağıdaki gibidir:

$$S_0=E_i\oplus E_j \tag{3.49}$$

$$S_1=T^i\left[E_i\right]\oplus T^j\left[E_j\right] \tag{3.50}$$

$$S_2=T^{2i}\left[E_i\right]\oplus T^{2j}\left[E_j\right] \tag{3.51}$$

$$S_3=T^{3i}\left[E_i\right]\oplus T^{3j}\left[E_j\right] \tag{3.52}$$

(3.49)'daki eşitliğin her iki tarafına T^i uygulanır ve (3.50) ile taraf tarafa toplanırsa,

$$T^i\left[S_0\right]\oplus S_1=\left(T^i\oplus T^j\right)\left[E_j\right] \tag{3.53}$$

(3.49)'daki eşitliğin her iki tarafına T^{2i} uygulanır ve (3.51) ile taraf tarafa toplanırsa,

$$T^{2i}\left[S_0\right]\oplus S_2=\left(T^{2i}\oplus T^{2j}\right)\left[E_j\right] \tag{3.54}$$

(3.50)'deki eşitliğin her iki tarafına T^i uygulanır ve (3.51) ile taraf tarafa toplanırsa,

$$T^i\left[S_1\right]\oplus S_2 = T^j\left(T^i\oplus T^j\right)\left[E_j\right] = T^j\left(T^i\left[S_0\right]\oplus S_1\right) \tag{3.55}$$

(3.50)'deki eşitliğin her iki tarafına T^{2i} uygulanır ve (3.52) ile taraf tarafa toplanırsa,

$$T^{2i}\left[S_1\right]\oplus S_3 = T^j\left(T^{3i}\oplus T^{3j}\right)\left[E_j\right] = T^j\left(T^{3i}\left[S_0\right]\oplus S_2\right) \tag{3.56}$$

Bu şekilde hata yerleştirilmiş olur. Yani i ve j bulunmuş olur. Dolayısıyla T^i ve T^j de bulunmuş olur. $T^i\oplus T^j = T^x$ diyelim. T terslenebilir olduğundan ve maksimal devir uzunluğuna sahip olduğundan T^x ve T^{-x} bu devir içine düşer. (3.53)'teki eşitliğin her iki tarafına T^{-x} uygulanırsa;

$$T^{-x}\left(T^i\left[S_0\right]\oplus S_1\right) = T^{-x}\left(T^i\oplus T^j\right)\left[E_j\right] \Rightarrow E_j = T^{-x}\left(T^i\left[S_0\right]\oplus S_1\right) \tag{3.57}$$

şeklinde E_j hata vektörü bulunur. (3.49) eşitliğinde E_j yerine yazılırsa,

$$E_i = S_0\oplus E_j \tag{3.58}$$

E_i hata vektörü bulunur.

Hataların biri bilgi baytlarında diğeri kontrol baytlarında oluşmuşsa $k\neq i,\ k\neq j,\ \left(0\le k\le N\right)$ şartını sağlayan bir k sayısı için dekodlama algoritması–2'nin üçüncü adımında belirtildiği üzere, iki hata ya i. ve j. bilgi baytlarında ya da k. bilgi baytı ile kontrol baytlarından birindedir. Bu durumda hata i, j ve k bilgi baytlarına yerleştirilmiş olur.

Örnek 3.29 $F=\langle 90,150,90,150\rangle$ Wolfram kurallarından oluşan global geçiş fonksiyonuna sıfır sınır şartı altında karşılık gelen matris;

$$T=\begin{pmatrix}0&1&0&0\\1&1&1&0\\0&1&0&1\\0&0&1&1\end{pmatrix}_{4\times 4}$$

şeklindedir. Ayrıca Örnek 2.6'dan T matrisinin karakteristik polinomu $\varphi(x)=x^4+x+1$ ve bu polinoma karşılık gelen devir yapısı $\left[1,1(15)\right]$ (bkz. Şekil 2.18) şeklinde olur. $b=4$ ve $N=15$ olduğundan $B=\left[B_0,B_1,...,B_{N-1}\right]=\left[B_0,B_1,...,B_{14}\right]$ olur ve her bir $B_i,\ (i=0,1,2,...,14)$ baytı $4-$bitten oluşur. $t=1$ bayt hata düzelten kod üretelim. Bu durumda $2t+1=3$ tane kontrol baytı üretmeliyiz.

$$B=\left[B_0,B_1,...,B_{14}\right]=\underbrace{\left[1111,1111,...,1111\right]}_{15-\tan e}$$

$$C_0=B_0\oplus B_1\oplus B_2\oplus...\oplus B_{14}=\left[1111\right]$$

$$C_1=T^{14}\left[B_0\right]\oplus T^{13}\left[B_1\right]\oplus...\oplus T\left[B_{13}\right]\oplus B_{14}=\left[0000\right]$$

$$C_2=T^{28}\left[B_0\right]\oplus T^{26}\left[B_1\right]\oplus...\oplus T^2\left[B_{13}\right]\oplus B_{14}=\left[0000\right]$$

Bu durumda $CW=\left[B_0,B_1,...,B_{14},C_0,C_1,C_2\right]=\underbrace{\left[1111,1111,...,1111,1111,0000,0000\right]}_{18-\tan e}$ olur. Soldan 4 . bilgi baytında hata meydana gelsin ve alınan söz;

$$CW'=\left[B_0',B_1',...,B_{14}',C_0',C_1',C_2'\right]=\left[\overset{0}{1111},\overset{1}{1111},\overset{2}{1111},\overset{3}{1111},\overset{4}{0000},...,\overset{14}{1111},\overset{15}{1111},\overset{16}{0000},\overset{17}{0000}\right]$$

şeklinde olsun. Kontrol baytlarını yeniden üretelim.

$$C_0'=B_0'\oplus B_1'\oplus B_2'\oplus...\oplus B_{14}'=\left[0000\right]$$

$$C_1'=T^{14}\left[B_0'\right]\oplus T^{13}\left[B_1'\right]\oplus...\oplus T\left[B_{13}'\right]\oplus B_{14}'=\left[1001\right]$$

$$C_2'=T^{28}\left[B_0'\right]\oplus T^{26}\left[B_1'\right]\oplus...\oplus T^2\left[B_{13}'\right]\oplus B_{14}'=\left[0110\right]$$

Bu durumda hata sendromları aşağıdaki gibidir:

$$S_0=C_0\oplus C_0'=E_i=\left[1111\right]$$

$$S_1=C_1\oplus C_1'=T^i\left[E_i\right]=\left[1001\right]$$

$$S_2=C_2'\oplus C_2=T^{2i}\left[E_i\right]=\left[0110\right]$$

$T^i\left[S_0\right]=S_1,\ T^{2i}\left[S_0\right]=S_2,\quad i=10$ için $T^{10}\left[S_0\right]=\left[1001\right],\ T^{20}\left[S_0\right]=\left[0110\right]$ şartı sağlandığından soldan $N-1-i=15-1-10=4$. bilgi baytında hata vardır ve hata vektörü $E=S_0=\left[1111\right]$ dir. $B_4=B_4^{'}\oplus E=\left[1111\right]$ şeklinde dekodlanır.

Şimdi $t=2$ bayt hata düzelten kod üretelim. Bu durumda yukarıdaki kontrol baytlarına ilave olarak bir tane daha kontrol baytı üretmeliyiz.

$$B=\left[B_0,B_1,...,B_{14}\right]=\underbrace{\left[1111,1111,...,1111\right]}_{15-\tan e}$$

$$C_0=B_0\oplus B_1\oplus B_2\oplus...\oplus B_{14}=\left[1111\right]$$

$$C_1=T^{14}\left[B_0\right]\oplus T^{13}\left[B_1\right]\oplus...\oplus T\left[B_{13}\right]\oplus B_{14}=\left[0000\right]$$

$$C_2=T^{28}\left[B_0\right]\oplus T^{26}\left[B_1\right]\oplus...\oplus T^2\left[B_{13}\right]\oplus B_{14}=\left[0000\right]$$

$$C_3=T^{42}\left[B_0\right]\oplus T^{39}\left[B_1\right]\oplus...\oplus T^3\left[B_{13}\right]\oplus B_{14}=\left[0000\right]$$

Bu durumda

$$CW=\left[B_0,B_1,...,B_{14},C_0,C_1,C_2,C_3\right]=\underbrace{\left[1111,1111,...,1111,1111,0000,0000,0000\right]}_{19-\tan e}$$

olur. Soldan 4. ve 5. bilgi baytlarında hata meydana gelsin ve alınan söz;

$$CW'=\left[B_0^{'},B_1^{'},...,B_{14}^{'},C_0^{'},...,C_3^{'}\right]=\left[\overset{0}{1111},...,\overset{3}{1111},\overset{4}{0000},\overset{5}{1101},...,,\overset{15}{1111},\overset{16}{0000},\overset{17}{0000},\overset{18}{0000}\right]$$

şeklinde olsun. Kontrol baytlarını yeniden üretelim.

$$C_0^{'}=B_0^{'}\oplus B_1^{'}\oplus B_2^{'}\oplus...\oplus B_{14}^{'}=\left[0010\right]$$

$$C_1^{'}=T^{14}\left[B_0^{'}\right]\oplus T^{13}\left[B_1^{'}\right]\oplus...\oplus T\left[B_{13}^{'}\right]\oplus B_{14}^{'}=\left[0101\right]$$

$$C_2^{'}=T^{28}\left[B_0^{'}\right]\oplus T^{26}\left[B_1^{'}\right]\oplus...\oplus T^2\left[B_{13}^{'}\right]\oplus B_{14}^{'}=\left[1111\right]$$

$$C_3^{'}=T^{42}\left[B_0\right]\oplus T^{39}\left[B_1\right]\oplus...\oplus T^3\left[B_{13}\right]\oplus B_{14}=\left[1100\right]$$

Bu durumda hata sendromları aşağıdaki gibidir:

$$S_0=C_0\oplus C_0^{'}=E_i\oplus E_j=\left[1101\right]$$

$S_1 = C_1 \oplus C_1' = T^i\left[E_i\right] \oplus T^j\left[E_j\right] = \left[0101\right]$

$S_2 = C_2' \oplus C_2 = T^{2i}\left[E_i\right] \oplus T^{2j}\left[E_j\right] = \left[1111\right]$

$S_3 = C_3' \oplus C_3 = T^{3i}\left[E_i\right] \oplus T^{3j}\left[E_j\right] = \left[1100\right]$

$E_j = T^{-x}\left[T^{-i}\left[S_0\right] \oplus S_1\right] = T^{N-x}\left[T^{-i}\left[S_0\right] \oplus S_1\right]$

$i = 10$, $j = 9$ için $T^i + T^j = T^x \Rightarrow x = 13, -x \equiv 2(\mathrm{mod}15)$ $E_j = T^2\left[T^{10}\left[S_0\right] \oplus S_1\right]$ $= \left[0010\right]$ ve $E_i = S_0 \oplus E_j = \left[1111\right]$ şartları sağlandığından soldan $N-1-i = 15-1-10 = 4$ ve $N-1-j = 15-1-9 = 5$. bilgi baytlarında hata vardır. $B_4 = B_4' \oplus E_4 = \left[1111\right]$, $B_5 = B_5' \oplus E_5 = \left[1111\right]$ şeklinde dekodlanır.

BÖLÜM 4

İKİDEN FAZLA DURUMA SAHİP HÜCRESEL DÖNÜŞÜMLERLE HATA DÜZELTEN KODLAR

Bu bölümde hücresel dönüşümlerle bit ve bayt hata düzelten kodlar için sonlu cisimler üzerinde kodlama ve dekodlama algoritmaları verilecektir. Bit hata düzelten kodlar için $G=\left[I_n \mid T_n^k \right]_{n\times 2n}$ matrisi tarafından üretilen ve $C \to \left[2n, n, d\right]_q$ parametrelerine sahip sistematik lineer kodlar düşünülecektir.

4.1 İlkel Sonlu Cisimler Üzerinde Hücresel Dönüşümlerle Bit Hata Düzelten Kodlar

4.1.1 Kodlama

Tanım 4.1 $I=\left(i_1, i_2, ..., i_n\right) \to n$ -bitten oluşan bilgi bitleri ve T bir boyutlu lineer temel hücresel dönüşümlerin temsili matrisi olmak üzere, bir $k \in \mathbb{Z}^+$ için $T^k\left[I\right]=\left(c_1, c_2, ..., c_n\right)$ olsun. $CW=\left(I, T^k\left[I\right]\right)=\left(i_1, i_2, ..., i_n, c_{n+1}, c_{n+2}, ..., c_{2n}\right)$ şeklindeki ifadeye bir kod söz denir.

Teorem 4.1 T $l-$diyagonal ($l \geq 2$), $n \times n$ tipinde periyodik sınır şartı ile verilen $\mathbb{F}_q$ (q asal) ilkel sonlu cismi üzerinde bir ya da iki boyutlu lineer hücresel dönüşümler için temsili matris ve C, $G=\left[I_n \mid T_n^k \right]_{n\times 2n}$ matrisi tarafından üretilen, $\left[2n, n, d\right]_q$ parametrelerine sahip sistematik lineer kod olsun. Bu durumda $1 \leq k < s,\ \left(k,\ s \in \mathbb{Z}^+\right)$ ve $T^s = I$ şartını sağlayan bir k tamsayısı için C sistematik lineer kodun d minimum uzaklığı $3 \leq d \leq n$ şeklindedir.

İspat: T^k simetrik matris olduğundan $G=\left[I_n \mid T^k\right] \Rightarrow H=\left[-T^k \mid I_n\right]$ dir. T matrisi tersinir olduğundan T^k matrisi de tersinirdir. O halde T^k matrisinin satır ve sütun rankları birbirine eşittir. Dolayısıyla T^k matrisinin herhangi iki sütunu lineer bağımsızdır. Bu durumda herhangi iki sütunun toplamı en az bir tane sıfırdan farklı koordinat içerir. H kontrol matrisini ve Teorem 3.6'yı bir arada düşündüğümüzde H matrisinin lineer bağımlı sütunlarının sayısı en az 3 olur. Dolayısıyla $d \geq 3$ dir. Eğer T^k matrisinin herhangi iki sütunu lineer bağımsız ve toplamlarında en az iki tane sıfırdan farklı koordinat varsa ve T^k matrisinin herhangi üç sütunu lineer bağımsız ise bu sütunların toplamında en az bir tane sıfırdan farklı koordinat vardır. H kontrol matrisini ve Teorem 3.6'yı bir arada düşündüğümüzde H matrisinin lineer bağımlı sütunlarının sayısı en az 4 olur. Benzer muhakeme ile eğer T^k matrisinin herhangi $i, \left(1 \leq i < d\right)$ sayıda sütununun toplamı $\left(d-i\right)-$ tane sıfırdan farklı koordinat içeriyorsa bu durumda H matrisinin herhangi $i, \left(i < d\right)$ sayıda sütunu lineer bağımsız ve lineer bağımlı sütunlarının minimum sayısı d dir. Singleton sınırından C lineer kodun minimum uzaklığı $k+d \leq n+1 \Rightarrow n+d \leq 2n+1 \Rightarrow d \leq n+1$ dir. Ancak aşikâr lineer kodlar dışında MDS kod bulunmadığından $d \leq n$ yazarız. Yani C lineer kodunun d minimum uzaklığı için $3 \leq d \leq n$ sınırları vardır.

4.1.2 Dekodlama

Bu dekodlama yöntemindeki temel düşünce bilgi ve kontrol haneleri için ayrı sendromlar üreterek bilinen klasik sendrom dekodlamasına nazaran işlem yükünün oldukça hafifletilmesine dayanmaktadır. Bu kolaylığı sağlayan esas etken ise kodlamada kullanılan tersinir matristir. Eğer bilgi kısmını oluşturan vektör bir lineer hücresel dönüşümün $t=r$ zaman adımındaki konfigürasyonu ise kontrol kısmı da $t=r+k$ zaman adımındaki konfigürasyona karşılık gelir. $CW=(I,T^k[I])$ gönderilen kod söz ve $CW'=\left(I',C'\right)=\left(i_1',i_2',...,i_n',c_{n+1}',c_{n+2}',\ldots,c_{2n}'\right)=(I\oplus I_e, T^k[I]\oplus C_e)$ alınan söz olsun. Burada $\oplus$ modülo q toplamayı ve I_e ile C_e sırasıyla bilgi ve kontrol kısmına etki eden hata vektörlerini göstermektedir. C lineer kodunun minimum uzaklığı $d=2t+1$ olmak üzere, C kodu bilgi ve kontrol kısmında oluşabilecek $t-$ tane ve daha az sayıdaki

hataları düzeltebilir. Yani $w_H(I_e)+w_H(C_e)\le t$ şeklindedir. S ile genel hata sendromunu gösterelim. Bu durumda

$$S=(q-1)T^k\left[I'\right]\oplus C'=(q-1)T^k[I_e]\oplus C_e \tag{4.1}$$

Ayrıca $n-$tane bilgi bitinden sorumlu hata sendromu da S_n olmak üzere;

$$S_n=(q-1)T^k[I']\oplus C' \tag{4.2}$$

$n-$tane kontrol bitinden sorumlu hata sendromunu da S_c ile gösterelim.

$$S_c=T^k[I']\oplus(q-1)C' \tag{4.3}$$

Eğer $S=0$ ise alınan söz hatasızdır. Eğer $S\neq 0$ ise alınan sözde hata oluşmuştur. Yani $C'\neq T^k[I']$ şeklindedir. Bu halde üç durum söz konusudur.

1. Hataların tamamı bilgi kısmında meydana gelmiş olabilir. Bu durumda $\left(T^{-k}\left[S_n\right]\oplus I',C'\right)$ kod sözdür ve bilgi kısmının hata vektörü $T^{-k}(S_n)$ şeklinde hesaplanır. Hata vektörü $\left(T^{-k}(S_n),0\right)$, $0=\underbrace{00\ldots0}_{n-\text{tane}}$ şeklindedir.

2. Hataların tamamı kontrol kısmında meydana gelmiş olabilir. Bu durumda $(I',C'\oplus S_c)$ kod sözdür ve hata vektörü $(0,S_c)$ dir.

3. Hem bilgi kısmı hem de kontrol kısmında hata oluşmuş olabilir. Yani $CW'=(I',C')=(I\oplus I_e,T^k[I]\oplus C_e)$ ə $I_e\neq 0$, $C_e\neq 0$ dir. Bu durumda kontrol kısmında oluşması muhtemel olan ve Hamming ağırlığı $w_H(C_e)\le t-1$ olan tüm hata vektörleri için $S_n=S\oplus C_e,\left(C_e=S_c\right)$ bilgi sendromu hesaplanır. Daha sonra $T^{-k}\left[S_n\right]$ bilgi kısmının muhtemel hata vektörü bulunur. $CW''=(T^{-k}\left[S_n\right]\oplus I',C'\oplus C_e)$ muhtemel kod söz için yeniden sendrom hesaplanır. Eğer bu sendrom sıfır ise $C_e\neq 0$ vektörü kontrol kısmının hata vektörü ve $T^{-k}\left[S_n\right]$ bilgi kısmının hata vektörüdür. Dolayısıyla $CW''=CW$ gönderilen kod sözdür. Eğer yeni hesaplanan sendrom sıfırdan farklı ise kontrol kısmında oluşabilecek muhtemel hatalardan başka bir tane seçilir ve aynı

işlemler tekrar edilir. Bu işlemlere yeni sendrom sıfır bulununcaya kadar devam edilir.

Teorem 4.2 Sistematik lineer kodlar için yukarıda verilen dekodlama planı $C \to [2n, n, d]_q$, $(d = 2t+1)$ parametrelerine sahip kodu doğru bir şekilde dekodlar.

İspat: Hataların oluşmasına bağlı olarak mümkün üç durumu ele alalım.

1. Eğer hataların tamamı bilgi kısmında meydana gelmişse bu durumda $C = C^{'}$ ve

$$\begin{aligned} &T^{-k}[S_n] \oplus I^{'} = T^{-k}\left[(q-1)T^k[I^{'}] \oplus C^{'}\right] \oplus I^{'} \\ &= (q-1)I^{'} \oplus T^{-k}\left[C^{'}\right] \oplus I^{'} \\ &= T^{-k}\left[C^{'}\right] = T^{-k}\left[T^k[I]\right] = I \end{aligned} \tag{4.4}$$

2. Eğer hataların tamamı kontrol kısmında meydana gelmişse bu durumda $I = I^{'}$

$$\begin{aligned} &(0, S_c) \oplus (I^{'}, C^{'}) = (I^{'}, S_c \oplus C^{'}) \\ &= (I^{'}, T^k[I^{'}] \oplus C^{'} \oplus (q-1)C^{'}) \\ &= \left(I, T^k[I]\right) \end{aligned} \tag{4.5}$$

3. Hem bilgi kısmı hem de kontrol kısmında hata oluşmuşsa $CW^{"} = (T^{-k}[S_n] \oplus I^{'}, C^{'} \oplus C_e)$ yukarıdaki gibi hesaplanan muhtemel kod söz olsun. Eğer C_e kontrol kısmına etki eden hata vektörü ise $C^{'} \oplus C_e = C$ ve S_N, $CW^{"}$ için yeniden hesaplanan sendrom olsun. Bu durumda

$$\begin{aligned} &S_N = (q-1)T^k\left[T^{-k}[S_n] \oplus I^{'}\right] \oplus \left(C^{'} \oplus C_e\right) \\ &= (q-1)\left(\left(S \oplus C_e\right) \oplus T^k\left[I^{'}\right]\right) \oplus \left(C^{'} \oplus C_e\right) \quad \left\{C^{'} \oplus C_e = T^k[I]\right\} \\ &= (q-1)\left(\left[(q-1)T^k\left[I^{'}\right] \oplus C^{'} \oplus C_e\right] \oplus T^k\left[I^{'}\right]\right) \oplus T^k[I] \\ &= (q-1)\left(\left[qT^k\left[I^{'}\right] \oplus T^k[I]\right]\right) \oplus T^k[I] \\ &= qT^k[I] = 0 \end{aligned} \tag{4.6}$$

$CW^{"}$ için yeniden hesaplanan sendrom sıfırdır. Yani $CW^{"} = CW$ dir.

Örnek 4.1 $F=\langle f_1,f_2,f_3,f_4\rangle$ bir boyutlu lineer hücresel dönüşümün global geçiş fonksiyonu ve $f_j\left(x_{i-1}^t,x_i^t,x_{i+1}^t\right)=x_i^{t+1}=x_{i-1}^t+x_i^t+2x_{i+1}^t\ (\bmod 3),\ (j=1,2,3,4)$ olsun. Bu durumda F'nin periyodik sınır şartı altında temsili matrisi; $T=\begin{pmatrix}1&2&0&1\\1&1&2&0\\0&1&1&2\\2&0&1&1\end{pmatrix}$ tersinirdir ve $T^2=\begin{pmatrix}2&1&2&2\\2&2&1&2\\2&2&2&1\\1&2&2&2\end{pmatrix}(\bmod 3)$, $k=1$ veya 2 için $G=\left[I_4\mid T^k\right]$ sistematik matrisi $C\to[8,4,4]_3-$kod üretir. $d(C)=4$ olduğundan bu kod $t=1$ tane hata düzeltebilir. $I=1111\Rightarrow C=T^2[I]=1111\Rightarrow CW=11111111$ kod sözdür. C bir hata düzeltebildiğinden hata ya bilgi kısmında ya da kontrol kısmında meydana gelebilir. $CW'=\hat{2}1111111=\left(I'\mid C'\right)$ alınan söz olsun. Yani hata bilgi kısmında meydana gelmiş olsun. Bu durumda $S=2T^2\left[I'\right]\oplus C'=0001\oplus 1111=1112$ olacak şekilde genel hata sendromu hesaplanır. Kontrol kısmı hatasız olduğundan $S_c=C_e=0000$ dir. O halde bilgi kısmının hata sendromu $S_4=S\oplus S_c=1112$ olarak hesaplanır. $I_e=T^{-2}[S_4]=2000$ ve $I=I'\oplus I_e=2111\oplus 2000=1111$ bulunur. $C=C'=1111$ olduğundan hata vektörü $E=20000000$.

Şimdi $CW'=1111111\hat{0}=\left(I'\mid C'\right)$ alınan söz olsun. Yani hata kontrol kısmında meydana gelmiş olsun. Kontrol kısmının hata sendromu $S=T^2\left[I'\right]\oplus 2C'=1111\oplus 2220=0001=C_e$. Bilgi kısmı hatasız olduğundan $S_4=0000$ ve $I_e=T^{-2}[S_4]=0000$. $C=C'\oplus C_e=1110\oplus 0001=1111$ ve $E=00000001$ hata vektörüdür.

Örnek 4.2 $F=\langle f_1,f_2,f_3,f_4,f_5,f_6\rangle$ bir boyutlu lineer hücresel dönüşümün global geçiş fonksiyonu ve $f_1=f_3=f_5=x_i^{t+1}=x_{i-2}^t+x_{i-1}^t+x_i^t+x_{i+1}^t+x_{i+2}^t\ (\bmod 3)$,

$f_2 = f_4 = f_6 = x_i^{t+1} = x_i^{t+1} = x_{i-2}^t + x_{i-1}^t + x_{i+1}^t + x_{i+2}^t \pmod 3$ olsun. Bu durumda F'nin periyodik sınır şartı altında temsili matrisi;

$$T = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}. \; k = 6 \text{ için } T^6 = \begin{pmatrix} 0 & 2 & 1 & 1 & 1 & 2 \\ 2 & 0 & 2 & 2 & 1 & 2 \\ 1 & 2 & 0 & 2 & 1 & 1 \\ 1 & 2 & 2 & 0 & 2 & 2 \\ 1 & 1 & 1 & 2 & 0 & 2 \\ 2 & 2 & 1 & 2 & 2 & 0 \end{pmatrix} \pmod 3$$

dir. Bu durumda $G = \left[I_6 \mid T^6 \right]$ sistematik matrisi bir $[12,6,6]_3 -$ kod üretir. $d(C) = 6$ olduğundan bu kod $t = 2$ tane hata düzeltebilir. $I = 111111 \Rightarrow C = T^6 [I]$ $= 1101010 \Rightarrow CW = 111111101010$ kod sözdür. $CW' = \hat{2}111\hat{0}1101010 = \left(I' \mid C' \right)$ alınan söz olsun. Yani iki hata bilgi kısmında meydana gelmiş olsun. Bu durumda $S = 2T^6 \left[I' \right] \oplus C' = 022110 \oplus 101010 = 120120$ olacak şekilde genel hata sendromu hesaplanır. Kontrol kısmı hatasız olduğundan $S_c = C_e = 000000$ dir. O halde bilgi kısmının hata sendromu $S_6 = S \oplus S_c = 120120$ olarak hesaplanır. $I_e = T^{-6} \left[S_6 \right] = 200010$ ve $I = I' \oplus I_e = 211101 \oplus 200010 = 111111$ bulunur. $C = C' = 101010$ olduğundan hata vektörü $E = 20001000000$.

Şimdi $CW' = 11111\hat{1}00101\hat{1} = \left(I' \mid C' \right)$ alınan söz olsun. Yani iki hata kontrol kısmında meydana gelmiş olsun. Kontrol kısmının hata sendromu $S_c = T^6 \left[I' \right] \oplus 2C'$ $= 101010 \oplus 002022 = 100002$. Bilgi kısmı hatasız olduğundan $S_6 = 000000$ ve $C_e = S_c = 200001$. $C = C' \oplus C_e = 001011 \oplus 100002 = 101010$ ve $E = 000000100002$ hata vektörüdür.

Son olarak hataların biri bilgi kısmında diğeri ise kontrol kısmında meydana gelmiş olsun. Yani $CW' = \hat{0}11111\hat{1}1010$ alınan söz olsun. Genel hata sendromu $S = 2T^6 \left[I' \right] \oplus C' = 220102 \oplus 111010 = 001112$ olarak hesaplanır. Her iki kısım hatalı olduğundan kontrol kısmının hata sendromu dolayısıyla hata vektörü de sıfırdan farklı olacaktır. Hata kısmına etkiyen hata vektörü $S_c = 020000$ olarak alındığında

$S_6 = S \oplus S_c = 021112$ bilgi kısmının sendromu olur. $I_e = T^{-6}\left[S_6\right] = 100000$ bilgi kısmının muhtemel hata vektörüdür. $I^{"} = I^{'} \oplus I_e = 011111 \oplus 100000 = 111111$ ve $C^{"} = C^{'} \oplus C_e = 111010 \oplus 020000 = 101010$ şeklinde hesaplanır. $CW^{''} = 111111101010$ muhtemel kod sözün sendromu sıfır olduğundan $CW^{''} = CW$ ve $E = 100000020000$ hata vektörüdür.

Hatanın hem bilgi kısmında hem de kontrol kısmında meydana geldiği durum için hücresel dönüşümlerle yapılan dekodlamada denenmesi gereken sendromların sayısı $\leq \sum_{i=0}^{1} \binom{6}{i} 2^i = 13$ dir. Ancak geleneksel sendrom dekodlaması kullanılırsa $\leq \sum_{i=0}^{2} \binom{12}{i} 2^i = 289$ tane sendromun denenmesi gerekir. Genel olarak hücresel dönüşümlerle yapılan dekodlamada denenmesi gereken sendromların sayısı $\leq \sum_{i=0}^{t-1} \binom{n}{i} (q-1)^i$ iken geleneksel sendrom dekodlamasında $\leq \sum_{i=0}^{t} \binom{2n}{i} (q-1)^i$ dir. Aşağıdaki tablo hücresel dönüşümlerle bit hata düzelten kodların bu konudaki üstünlüğünü gözler önüne sermesi bakımından önemlidir.

q	n	t	Geleneksel Metot	Hücresel Dönüşüm Tabanlı
3	5	2	201	11
3	10	2	801	21
3	15	2	1801	31
3	20	3	82241	801
3	20	4	1544481	9921

Tablo 4. 1 Hücresel dönüşüm tabanlı dekodlama ile geleneksel sendrom dekodlaması arasında bir karşılaştırma

Tablo 4.1'de verilen q sonlu cismin karakteristiğini, n bilgi bitlerinin sayısını ve $t,\ \left(d = 2t+1\right)$ de kodun düzeltebildiği hata sayısını göstermektedir.

4.2 İlkel Sonlu Cisimler Üzerinde Hücresel Dönüşümlerle Bayt Hata Düzelten Kodlar

4.2.1 Kodlama

T , $b-$uzunluğunda bir boyutlu lineer hücresel dönüşümün global geçiş fonksiyonuna karşılık gelen N devir uzunluğuna sahip tersinir temsili matris olsun. $B=\left[B_0,B_1,...,B_{N-1}\right]$ ile her biri $b-$bitten oluşan $N-$tane $\left(N\le q^b-1\right)$ bilgi baytını göstersin. $t-$bayt hata düzeltebilmesi için $\left(2t+1\right)-$tane kontrol baytı üretilmelidir. $C=\left[C_0,C_1,...,C_{2t}\right]$ kontrol baytları olsun. Bu kontrol baytlarının her biri $i=0,1,2,...,2t+1$ olmak üzere;

$$C_i=T^i\left[B_{N-1}\right]\oplus T^{2i}\left[B_{N-2}\right]\oplus...\oplus T^{(N-1)i}\left[B_1\right]\oplus T^{Ni}\left[B_0\right] \tag{4.7}$$

şeklinde üretilir.

4.2.1.1 İki Bayt Hata Düzelten-İki Bayt Hata Fark Eden Kod

İki bayt hata düzeltebilmesi için her kod söze dört tane kontrol baytı eklenmelidir. Bu kontrol baytları (4.7) ifadesinden aşağıdaki gibidir:

$$C_0=B_{N-1}\oplus B_{N-2}\oplus...\oplus B_1\oplus B_0 \tag{4.8}$$

$$C_1=T\left[B_{N-1}\right]\oplus T^2\left[B_{N-2}\right]\oplus...\oplus T^{N-1}\left[B_1\right]\oplus T^N\left[B_0\right] \tag{4.9}$$

$$C_2=T^2\left[B_{N-1}\right]\oplus T^4\left[B_{N-2}\right]\oplus...\oplus T^{2(N-1)}\left[B_1\right]\oplus T^{2N}\left[B_0\right] \tag{4.10}$$

$$C_3=T^3\left[B_{N-1}\right]\oplus T^6\left[B_{N-2}\right]\oplus...\oplus T^{3(N-1)}\left[B_1\right]\oplus T^{3N}\left[B_0\right] \tag{4.11}$$

Bu durumda $CW=\left(B_0,B_1,...,B_{N-1},C_0,C_1,C_2,C_3\right)$ bir kod sözdür.

4.2.2 Dekodlama

Tanım 4.2 (Hata sendromu) C_i , i . kontrol baytı ve $C_i^{'}$ de alınan bilgi baytlarından tekrar hesaplanan i . kontrol baytı (hatalı olabilir) olsun. Bu durumda i . kontrol baytına karşılık gelen hata sendromu $0\le \mathrm{i}\le 2\mathrm{t}$ olmak üzere,

$$S_i=C_i\oplus\left(q-1\right)C_i^{'} \tag{4.12}$$

şeklinde hesaplanır.

Tanım 4.3 (Hata baytı) B_{N-1-j} ve B'_{N-1-j} sırasıyla gönderilen ve alınan (Soldan sayıldığında j. bayt, sağdan sayıldığında $(N-1-j)$. bayta eşittir.) bilgi baytları olsun. Bu durumda $j \geq 0$ için

$$E_j = B_{N-1-j} \oplus B'_{N-1-j} \tag{4.13}$$

şeklindeki ifadeye j. hata baytı denir.

Hata düzeltmede temel amaç alınan $B' = \left[B'_0, B'_1, ..., B'_{N-1}\right]$ bilgi baytları ve $C' = \left[C'_0, C'_1, ..., C'_{2t}\right]$ kontrol baytlarından $E = \left[E_0, E_1, ..., E_{N-1}\right]$ hata vektörünü hesaplamaktır. Bu durumda

$$\left[B\right] = \left[B'\right] \oplus \left[E\right] \tag{4.14}$$

Soldan k. ve l. baytlarda hata oluşsun ve bu hatalara karşılık gelen hata vektörleri sırasıyla E_k ve E_l olsun. O halde kontrol baytları tekrar aşağıdaki gibi üretilir:

$$C'_0 = B'_0 \oplus B'_1 \oplus ... \oplus B'_{N-1} \oplus E_k \oplus E_l \tag{4.15}$$

$$C'_1 = T^N\left[B'_0\right] \oplus T^{N-1}\left[B'_1\right] \oplus ... \oplus T^2\left[B'_{N-2}\right] \oplus T\left[B'_{N-1}\right] \oplus T^i\left[E_k\right] \oplus T^j\left[E_l\right] \tag{4.16}$$

$$C'_2 = T^{2(N)}\left[B'_0\right] \oplus T^{2(N-1)}\left[B'_1\right] \oplus ... \oplus T^4\left[B'_{N-2}\right] \oplus T^2\left[B'_{N-1}\right] \oplus T^{2i}\left[E_k\right] \oplus T^{2j}\left[E_l\right] \tag{4.17}$$

$$C'_3 = T^{3(N)}\left[B'_0\right] \oplus T^{3(N-1)}\left[B'_1\right] \oplus ... \oplus T^6\left[B'_{N-2}\right] \oplus T^3\left[B'_{N-1}\right] \oplus T^{2i}\left[E_k\right] \oplus T^{2j}\left[E_l\right] \tag{4.18}$$

Yeniden üretilen kontrol baytları C'_0, C'_1, C'_2, C'_3 olmak üzere, $S_i\,(i = 0,1,2,3)$ hata sendromları;

$$S_i = C_i \oplus (q-1)C'_i \tag{4.19}$$

ile hesaplanır. Bu hazırlıklardan sonra iki bayt hata düzelten kodlarda hata vektörünün hesapalnmasını gerektiren durumları tek tek ele alarak her bir durum için hata yerini tespit ederek hata vektörünü (veya hata yerlerini tespit ederek hata vektörlerini) hesaplayalım:

1. Bilgi baytlarından sadece bir tanesinin hatalı olması durumu:

$i.$ bilgi baytı hatalı olsun ve bu yere karşılık gelen hata vektörünü E_l ile gösterelim. Bu durumda hata sendromları aşağıdaki gibidir:

$$S_0 = (q-1)E_l \tag{4.20}$$

$$S_1 = (q-1)T^i[E_l] \tag{4.21}$$

$$S_2 = (q-1)T^{2i}[E_l] \tag{4.22}$$

$$S_3 = (q-1)T^{3i}[E_l] \tag{4.23}$$

Burada $i+l=N$ ve $N=q^b-1$ dir. Yukarıdaki sendrom denklemleri kullanılarak aşağıdaki eşitlikler elde edilir.

$$T^i[S_0]=S_1,\ T^i[S_1]=S_2,\ T^i[S_2]=S_3 \text{ ve } E_l=(q-1)S_0 \tag{4.24}$$

olarak elde edilir. Burada $l=N-i$ hata yeridir.

2. Bilgi baytlarından iki tanesinin hatalı olması durumu:

Bu durumda $l.$ ve $m.$ bilgi baytları hatalı olsun. Sırasıyla E_l ve E_m bu hata yerlerine karşılık gelen hata vektörleri olsun. Sendrom denklemleri aşağıdaki gibidir:

$$S_0 = (q-1)(E_l \oplus E_m) \tag{4.25}$$

$$S_1 = (q-1)\left(T^i[E_l] \oplus T^j[E_m]\right) \tag{4.26}$$

$$S_2 = (q-1)\left(T^{2i}[E_l] \oplus T^{2j}[E_m]\right) \tag{4.27}$$

$$S_3 = (q-1)\left(T^{3i}[E_l] \oplus T^{3j}[E_m]\right) \tag{4.28}$$

Burada $i+l=N$ ve $j+m=N$ dir. Yukarıdaki sendrom denklemlerinden,

$$(q-1)T^i[S_0] \oplus S_1 = \left(T^i \oplus (q-1)T^j\right)[E_m] \tag{4.29}$$

$$(q-1)T^{2i}[S_0] \oplus S_2 = \left(T^{2i} \oplus (q-1)T^{2j}\right)[E_m] \tag{4.30}$$

$$(q-1)T^i[S_1] \oplus S_2 = T^j\left(T^i \oplus (q-1)T^j\right)[E_m] = T^j\left((q-1)T^i[S_0] \oplus S_1\right) \tag{4.31}$$

$$(q-1)T^{2i}[S_1] \oplus S_3 = T^j\left(T^{2i} \oplus (q-1)T^{2j}\right)[E_m] = T^j\left((q-1)T^{2i}[S_0] \oplus S_2\right) \quad (4.32)$$

Kodlamada kullanılan hücresel dönüşüm maksimal uzunluğa sahip olduğundan $T^i \oplus (q-1)T^j = T^r$ olacak şekilde $i,\ j,\ r \in \mathbb{Z}^+$ vardır. T matrisi terslenebilir olduğundan T^{-r} vardır. Dolayısıyla (4.29) eşitliğinden;

$$E_m = T^{-r}\left((q-1)T^i[S_0] \oplus S_1\right) \text{ ve } E_l = (q-1)(S_0 \oplus E_m) \quad (4.33)$$

elde edilir.

3. Bir bilgi baytı ve kontrol baytlarından bir tanesinin hatalı olma durumu:

A. l. bilgi baytı ile birinci kontrol baytı hatalı olsun ve E_l, E_0 sırasıyla karşılık gelen hata vektörlerini göstersin. Bu durumda hata sendromları aşağıdaki gibidir:

$$S_0 = (q-1)E_l \oplus E_0 \quad (4.34)$$

$$S_1 = (q-1)T^i[E_l] \quad (4.35)$$

$$S_2 = (q-1)T^{2i}[E_l] \quad (4.36)$$

$$S_3 = (q-1)T^{3i}[E_l] \quad (4.37)$$

Bu denklemler yardımıyla aşağıdaki eşitlikleri elde edebiliriz.

$$S_3 = T^i[S_2],\ S_3 = T^{2i}[S_1],\ T^i[S_0] \neq S_1 \quad (4.38)$$

Dolayısıyla (4.35) denkleminden hata vektörü aşağıdaki gibi hesaplanır:

$$E_l = (q-1)T^{-i}[S_1] \quad (4.39)$$

Burada $l = N - i$ dir.

B. l. bilgi baytı ile ikinci kontrol baytı hatalı olsun ve E_l, E_1 sırasıyla karşılık gelen hata vektörlerini göstersin. Bu durumda hata sendromları aşağıdaki gibidir:

$$S_0 = (q-1)E_l \quad (4.40)$$

$$S_1 = (q-1)T^i[E_l] \oplus E_1 \quad (4.41)$$

$$S_2 = (q-1)T^{2i}[E_l] \quad (4.42)$$

$$S_3 = (q-1)T^{3i}[E_l] \tag{4.43}$$

Bu denklemler yardımıyla aşağıdaki eşitlikleri elde edebiliriz.

$$S_3 = T^i[S_2], S_2 = T^{2i}[S_0], T^i[S_0] \neq S_1 \tag{4.44}$$

Dolayısıyla (4.40) denkleminden hata vektörü aşağıdaki gibi hesaplanır:

$$E_l = (q-1)S_0 \tag{4.45}$$

i (4.44)'teki eşitliklerinden hesaplanır.

C. l. bilgi baytı ile üçüncü kontrol baytı hatalı olsun ve E_l, E_2 sırasıyla karşılık gelen hata vektörlerini göstersin. Bu durumda hata sendromları aşağıdaki gibidir:

$$S_0 = (q-1)E_l \tag{4.46}$$

$$S_1 = (q-1)T^i[E_l] \tag{4.47}$$

$$S_2 = (q-1)T^{2i}[E_l] \oplus E_2 \tag{4.48}$$

$$S_3 = (q-1)T^{3i}[E_l] \tag{4.49}$$

Bu denklemler yardımıyla aşağıdaki eşitlikleri elde edebiliriz.

$$S_1 = T^i[S_0], S_3 = T^{2i}[S_1], T^i[S_2] \neq S_3 \tag{4.50}$$

Dolayısıyla (4.46) denkleminden hata vektörü aşağıdaki gibi hesaplanır:

$$E_l = (q-1)S_0 \tag{4.51}$$

i (4.50)'deki eşitliklerinden hesaplanır.

C. l. bilgi baytı ile dördüncü kontrol baytı hatalı olsun ve E_l, E_3 sırasıyla karşılık gelen hata vektörlerini göstersin. Bu durumda hata sendromları aşağıdaki gibidir:

$$S_0 = (q-1)E_l \tag{4.52}$$

$$S_1 = (q-1)T^i[E_l] \tag{4.53}$$

$$S_2 = (q-1)T^{2i}[E_l] \tag{4.54}$$

$$S_3 = (q-1)T^{3i}[E_l] \oplus E_3 \tag{4.55}$$

Bu denklemler yardımıyla aşağıdaki eşitlikleri elde edebiliriz.

$$S_1 = T^i[S_0], S_2 = T^{2i}[S_0], T^i[S_2] \neq S_3 \tag{4.56}$$

Dolayısıyla (4.52) denkleminden hata vektörü aşağıdaki gibi hesaplanır:

$$E_l = (q-1)S_0 \tag{4.57}$$

i (4.56)'daki eşitliklerinden hesaplanır.

NOT: Hücresel dönüşümlerle bir bayt hata düzelten kodlarda söz konusu hücresel dönüşümün maksimal uzunluklu olmasına gerek yoktur. Ancak hata sayısının bir bayttan fazla olduğu durumlarda hata yerinin saptanabilmesi için maksimum devir uzunluğuna ihtiyaç vardır. Eğer kodlamada kullanılan hücresel dönüşüm maksimal uzunluklu değilse ve bir bayttan fazla hata meydana gelmişse bu durumda farklı hata çiftleri aynı sendromları üretebileceğinden hata yerini tam olarak tespit etmek mümkün değildir.

Örnek 4.3 $F = \langle 90,90,90,90 \rangle$ Wolfram kurallarından oluşan global geçiş fonksiyonuna sıfır sınır şartı altında karşılık gelen matris; $T = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}_{4\times 4}$ şeklindedir. Ayrıca Örnek 2.5'ten $T, \left(T^6 = I \Rightarrow \circ(T) = 6\right)$ matrisinin karakteristik polinomu $\left(x^2 + x + 1\right)^2$ ve bu polinoma karşılık gelen devir yapısı $\left[1,1(3),2(6)\right]$ (bkz. Şekil 2.17) şeklinde olur. Bu devir yapısı;

$U_0 = \{0000\}$

$U_1 = \{0001, 0010, 0101, 1000, 0100, 1010\}$

$U_2 = \{0111, 1101, 1100, 1110, 1011, 0011\}$

$U_3 = \{1001, 0110, 1111\}$

$b=4$ ve C_3 için $N=3\Rightarrow N-1=2$, $B=\left[\overset{B_0}{1111},\overset{B_1}{1111},\overset{B_2}{1111}\right]$, $\left(B_i\in\mathbb{F}_2^4, i=0,1,2\right)$ sözünü $t=1$ bayt hata düzeltecek şekilde kodlayalım.

$C_0=B_0\oplus B_1\oplus B_2=[1111]$

$C_1=T^2[B_0]\oplus T[B_1]\oplus B_2=[0000]$

$C_2=T^4[B_0]\oplus T^2[B_1]\oplus B_2=[0000]$

$CW=\left[\overset{B_0}{1111},\overset{B_1}{1111},\overset{B_2}{1111},\overset{C_0}{1111},\overset{C_1}{0000},\overset{C_2}{0000}\right]$

$B'=\left[\overset{B_0'}{1111},\overset{B_1'}{0001},\overset{B_2'}{1111}\right]$

alınan söz olsun. Kontrol baytları tekrar hesaplanırsa;

$C_0'=B_0'\oplus B_1'\oplus B_2'=[0001]$

$C_1'=T^2\left[B_0'\right]\oplus T\left[B_1'\right]\oplus B_2'=[0000]$

$C_2'=T^4\left[B_0'\right]\oplus T^2\left[B_1'\right]\oplus B_2'=[0000]$

O halde $CW'=\left[\overset{B_0'}{1111},\overset{B_1'}{0001},\overset{B_2'}{1111},\overset{C_0'}{0001},\overset{C_1'}{0000},\overset{C_2'}{0000}\right]$. Bu durumda hata sendromları aşağıdaki gibidir:

$S_0=C_0\oplus C_0'=[1110]$

$S_1=C_1\oplus C_1'=[1011]$

$S_2=C_2'\oplus C_2=[0011]$

$T[S_0]=[1011], T^2[S_0]=[0011]$

$i=1$ olup $N-1-i=2-1=1$ baytta hata vardır ve hata vektörü S_0 dır. $B_1=B_1'\oplus E=[1111]$ şeklinde hesaplanır.

Örnek 4.4 $\mathbb{F}_3$ üzerinde $F=\langle f_1,f_2,f_3,f_4\rangle$ sıfır sınır şartı altında

$f_1\left(x_{i-1}^t,x_i^t,x_{i+1}^t\right)=f_4\left(x_{i-1}^t,x_i^t,x_{i+1}^t\right)=x_i^{t+1}=x_{i+1}^t\left(\bmod 3\right)$

$f_2\left(x_{i-1}^t,x_i^t,x_{i+1}^t\right)=f_3\left(x_{i-1}^t,x_i^t,x_{i+1}^t\right)=x_i^{t+1}=x_{i-1}^t+x_i^t\left(\bmod 3\right)$ olarak alınırsa,

$$T=\begin{pmatrix}0&1&0&0\\1&1&0&0\\0&0&0&1\\0&0&1&1\end{pmatrix}$$

matrisi elde edilir. $q=3,\ b=4$ olduğundan toplam $q^b=3^4=81$ tane konfigürasyon vardır. $T,\ \left(T^8=I\Rightarrow\circ(T)=8\right)$ matrisine karşılık gelen devir yapısı aşağıdaki gibidir:

$U_0=\{0000\}$

$U_1=\{0120,1102,1222,2021,0210,2201,2111,1012\}$

$U_2=\{1000,0100,1100,1200,2000,0200,2200,2100\}$

$U_3=\{0010,0001,0011,0012,0020,0002,0022,0021\}$

$U_4=\{1111,1212,2020,0202,2222,2121,1010,0101\}$

$U_5=\{1110,1201,2011,0212,2220,2102,1022,0121\}$

$U_6=\{0111,1112,1220,2002,0222,2221,2110,1001\}$

$U_7=\{1221,2010,0201,2211,2112,1020,0102,1122\}$

$U_8=\{2122,1021,0110,1101,1211,2012,0220,2202\}$

$U_9=\{1121,1210,2001,0211,2212,2120,1002,0122\}$

$U_{10}=\{1011,0112,1120,1202,2022,0221,2210,2101\}$

$B=\left[\overset{B_0}{0120},\overset{B_1}{1102},\overset{B_2}{1222},\overset{B_3}{2021},\overset{B_4}{0210},\overset{B_5}{2201},\overset{B_6}{2111},\overset{B_7}{1012}\right]$ $4-$bitlik 8 bayttan oluşan söz olsun. Devir uzunluğu $N=8\Rightarrow N-1=7$. $t=1$ bayt hata düzelten kod için $3-$tane kontrol baytı üretelim.

$C_0 = B_0 \oplus B_1 \oplus B_2 \oplus ... \oplus B_7 = [0000]$

$C_1 = T^7[B_0] \oplus T^6[B_1] \oplus ... \oplus T[B_6] \oplus B_7 = [2021]$

$C_2 = T^{14}[B_0] \oplus T^{10}[B_1] \oplus ... \oplus T^2[B_6] \oplus B_7 = [0000]$

$CW = \left[\overset{B_0}{0120}, \overset{B_1}{1102}, \overset{B_2}{1222}, \overset{B_3}{2021}, \overset{B_4}{0210}, \overset{B_5}{2201}, \overset{B_6}{2111}, \overset{B_7}{1012}, \overset{C_0}{0000}, \overset{C_1}{2021}, \overset{C_2}{0000} \right]$ kod sözü gönderilmiş olsun ve 0. bilgi baytı değişmiş olsun. Bu durumda

$B' = \left[\overset{B_0'}{0000}, \overset{B_1'}{1102}, \overset{B_2'}{1222}, \overset{B_3'}{2021}, \overset{B_4'}{0210}, \overset{B_5'}{2201}, \overset{B_6'}{2111}, \overset{B_7'}{1012} \right]$ alınan bilgi baytları olur. Kontrol baytlarını yeniden üretelim.

$C_0' = B_0' \oplus B_1' \oplus B_2' \oplus ... \oplus B_7' = [0210]$

$C_1' = T^7[B_0'] \oplus T^6[B_1'] \oplus ... \oplus T[B_6'] \oplus B_7' = [1012]$

$C_2' = T^{14}[B_0'] \oplus T^{10}[B_1'] \oplus ... \oplus T^2[B_6'] \oplus B_7' = [1222]$

$CW' = \left[\overset{B_0'}{0000}, \overset{B_1'}{1102}, \overset{B_2'}{1222}, \overset{B_3'}{2021}, \overset{B_4'}{0210}, \overset{B_5'}{2201}, \overset{B_6'}{2111}, \overset{B_7'}{1012}, \overset{C_0'}{0210}, \overset{C_1'}{1012}, \overset{C_2'}{1222} \right]$ olarak yazılabilir. Hata sendromları aşağıdaki gibidir:

$S_0 = C_0 \oplus 2C_0' = [0120]$

$S_1 = C_1 \oplus 2C_1' = [1012]$

$S_2 = C_2 \oplus 2C_2' = [2111]$

$T^7[S_0] = [1012], T^{14}[S_0] = [2111]$

$i = 1$ olup $N - 1 - i = 7 - 7 = 0$. bilgi baytı hatalıdır ve hata baytı S_0 dır. O halde

$B_0 = B_0' \oplus E = [0120]$ şeklinde bulunmuş olur.

BÖLÜM 5

SONUÇ VE ÖNERİLER

Bu tez çalışmasında hücresel dönüşümlerin tanımlanması ve cebirsel yapıları konusunda şu ana kadar yapılmış birçok çalışma taranarak zaman zaman ayrıntılı sayılabilecek bilgiler bir bütünlük içerisinde sunulmuştur. Ardından lineer blok kodlarla ilgili temel tanım ve teoremler verilmiştir.

$\mathbb{F}_2$ cismi üzerinde tanımlanan hücresel dönüşümlerle hata düzelten kodların teorisi, "Hücresel Dönüşümlerle Bit Hata Düzelten Kodlar" ve "Hücresel Dönüşümlerle Bayt Hata Düzelten Kodlar" başlıklarıyla verildikten sonra bu teori örneklerle desteklenmiştir.

Son bölümde ise üçüncü bölümde en küçük ilkel cisim ($\mathbb{F}_2 = \{0,1\}$) üzerinde tanımlanan hücresel dönüşümlerle hata düzelten kodların teorisi ilkel sonlu cisimler ($\mathbb{F}_q$, q asal) üzerine genellenerek konu özgün örneklerle desteklenmiştir. Ayrıca hücresel dönüşümlerle bit hata düzelten kodların minimum uzaklığı için belli şartlar altında alt ve üst sınır bir teorem ile ifade edilmiştir.

Kodlama teorisinde cebirsel yapılardan büyük ölçüde istifade edilmektedir. Dolayısıyla yeni kodlama-dekodlama algoritmaları tanımlamak için kullanılan yapıların cebirsel özelliklerinin biliniyor olması büyük kolaylıklar sağlar. Bu noktada nonlineer hücresel dönüşümlerle kodlama-dekodlama yapılabilecek uygun cebirsel yöntemlerin araştırılması büyük merak konusudur.

Convolutional kodlar ile hücresel dönüşümler arasında sıkı bir ilişkinin var olduğunu öngörmekteyiz. Bu ilişkinin araştırılması ve convolutional kodların mevcut kodlama-dekodlama yöntemlerinden daha verimli bir yöntemin hücresel dönüşümler aracılığıyla tanımlanıp tanımlanamayacağı sorusu cevabını bekleyen önemli sorular arasındadır.

Aynı şekilde LDPC (Low Density Parity Check) kodlar ile hücresel dönüşümler arasında da anlamlı ilişkiler araştırmanın zaman harcamaya değer bir çaba olduğunu düşünmekteyiz.

KAYNAKLAR

[1] Ilachiski, A., (2001). Cellular Automata: *A Discrete Universe*, First Edition, World Scientific, New York.

[2] Neumann, J. von, (1951). Collected Works, Design of Computers Theory of Automata and Numerical Analysis, Pergamon Press.

[3] Ulam, S. M., (1952). "Random Processes and Transformations", Proc. International Congress of Mathematicians, Cambridge MA., 2: 264–275.

[4] Ulam, S. M., (1962). "On Some Mathematical Problems Connected with Pattern of Growth of Figures", Proc. Syp. Appl. Math. 14: 215–224.

[5] Beyer W. A. vd., (1985). "Stanislaw M. Ulam's Contributions to Theoretical Theory", Letters in Mathematical Physics, 10: 231–242.

[6] Neumann, J. von, (1966). The Theory of Self Reproducing Automata (Edited by A. W. Burks), University of Illinois Press, Urbana.

[7] Schiff, J. L., (2008). Cellular Automata: A Discrete View of the World, First Edition, Wiley &Sons, Inc. Hoboken, New Jersey.

[8] Hedlund, G. A., (1969). "Endomorphisms and Automorphisms of The Shift Dynamical System", Math. Syst. Theory 3: 320–375.

[9] Codd, E. F., (1968). Cellular Automata, First Edition, Academic Press, New York.

[10] Langton, C. G., (1984). "Self-Reproduction in Cellular Automata", Physica D: Nonlinear Phenomena 10 (1–2): 135–144.

[11] Gardner, M., (1970). "Mathematical Games: The Fantastic Fombinations of John Conway's New Solitaire Game "Life"", Scientific American 223: 120–123.

[12] Wolfram, S., (2002). A New Kind of Science, First Edition, Wolfram Media, Inc.

[13] Wolfram, S., (1986). "Cryptography with Cellular Automata", Lecture Notes in Computer Science, 218, Springer-Verlag, 19: 429–432.

[14] Siap, I. vd., (2010). "Garden of Eden Configurations for 2-D Cellular Automata with Rule 2460-N", Information Sciences, 180, 3562–3571.

[15] Chaudhuri, P.P., vd., (1997). Additive Cellular Automata: Theory and Applications, First Edition, IEEE Press, New York.

[16] Tobler, W., (1979). “Cellular Geography, Gale, S. ve Olsson, G. (eds.)”, Philosophy in Geography, 379–386.

[17] White, R. ve Engelen, G., (1993). “Fractal Urban Land Use Patterns: A Cellular Automata Approach”, Environment and Planning A 25: 1175–1199.

[18] Sirakoulis, G. C. vd., (2003). “A Cellular Automaton Model for The Study of DNA Sequence Evolution”, Comput. Biol. Med. 33: 439–453.

[19] Barlovic, R. vd., (1998). “Metastable States in Cellular Automata for Traffic Flow”, The European Physical Journal B, 793–800.

[20] Kronholm, K. ve Birkeland K. W., (2005). “Integrating Spatial Patterns into A Snow Avalanche Cellular Automata Model”, Geophysical. Research Letters, 32.

[21] Nandi, S. vd., (1994). “Theory and Applications of Cellular Automata in Cryptography”, IEEE Trans. On Computers, 43: 1346–1357.

[22] Massey, J. L. ve Codes M. K., (1967). “Automata, and Continuous Systems: Explicit Interconnections”, Adaptive Processes Symp. Natl. Electron. Conf. Chicago.

[23] Massey, J. L., (1968). Application of Automata Theory in Coding, Applied Automata Theory, First Edition, Academic Press Inc., New York.

[24] Chowdhury, D.R. vd., (1994). “Design of CAECC Cellular Automata Based Error Correcting Code”, IEEE Trans. Computers, 43: 759–764.

[25] Chowdhury, D.R. vd., (1994). “CA Based Byte Error Correcting Code”, IEEE Trans. on Computers 43: 371–382.

[26] Cho, S. J. vd., (2006). “Design of Double Error Correctıng Codes Based on Cellular Automata”, J. Appl. Math. & Computing, 21: 545–553.

[27] Das, A. K. ve Chaudhuri, P. P., (1990). “Efficient Characterization of Cellular Automata”, Proc. IEE (Pan E), 137: 81–87.

[28] Khan, A.R. vd., (1997). “VLSI Architecture of A Cellular Automata”, Comput. Math. Appl. 33: 79–94.

[29] Chattopadhyay, P. vd., (1999). “Characterisation of a Particular Hybrid Transformation of Two-Dimensional Cellular Automata”, Computers and Mathematics with Applications 38: 207–216.

[30] Dihidar, K. ve Choudhury, P. P., (2004). “Matrix Algebraic Formula Concerning Some Exceptional Rules of Two Dimensional Cellular Automata”, Information Sciences, 165: 91–101.

[31] Siap, I. vd., (2011). “Characterization of Two Dimensional Hybrid Cellular Automata with Exceptional Family of Two Rules over Ternary Fields”, Journal of the Franklin Institute, 348: 1258–1275.

[32] Lipschutz, S., (1990). Lineer Cebir, Teori ve Problemleri, (Çeviri Editörü: Hilmi Hacısalioğlu), İkinci basım, Nobel Yayın Dağıtım, Ankara.

[33] Elspas, B., (1959). “The Theory of Autonomous Linear Sequential Networks”, TRE Trans. Circuits, CT–6: 45–60.

[34] Ling, S. ve Xing, C., (2004). Coding Theory, A First Course, First Edition, Cambridge University Press, New York.

[35] Morelos-Zaragoza, R. H., (2006). The Art of Error Correcting Coding, Second Edition, John Wiley & Sons Ltd, West Sussex, England.

[36] Bhaumik, J., Chowdhury, D. R. ve Chakrabarti, I., (2008). "An Improved Double Byte Error Correcting Code Using Cellular Automata", In Proc. 8th Int. Conf. Cellular Automat for Res. Ind. (ACRI), LNCS 5191: 463–470.

[37] Cho, S. J. vd., (2004). “Single Error Correcting Code Using PBCA”, J. Appl. Math.& Computing, 14: 461–471.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Mehmet Emin KÖROĞLU

Doğum Tarihi ve Yeri : 1982, Gerger/Adıyaman

Yabancı Dili : İngilizce

E-posta : sadecesad@gmail.com

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Y. Lisans	Matematik	Yıldız Teknik Üniversitesi	2012
P.F.P.	Eğitim Bilimleri	Yıldız Teknik Üniversitesi	2011
Y. Lisans	Hazırlık	Karadeniz Teknik Üniversitesi	2009
Lisans	Matematik	Atatürk Üniversitesi	2009
Lise	Fen Bilimleri	Ankara Açıköğretim Lisesi	2006
Lise	Fen Bilimleri	Esenler İbrahim Turhan Lisesi	1995-1997

Bildiri

1. M. E. Koroglu, I. Siap, H. Akin, F. Temiz, Hybrid Quadratic Cellular Automata and Its Applications to Pseudo Random Number Generators, Procedia Computer Science, 2011 (accepted) (CPCI-S).

2. F. Temiz, I. Siap, H. Akin, M. E. Koroglu, A Family of Two Dimensional Hybrid Cellular Automata and Its Applications to Pseudo Random Number Generators, Procedia Computer Science, 2011 (accepted) (CPCI-S).

3. M. E. Koroglu, I. Siap, H. Akin, Cellular Automata Based Byte Error Correcting Codes over Ternary Fields, International Conference on Applied Analysis and Algebra, Istanbul, Turkey, pp. 167-168, June 20-24, (2012).

Proje

1. **Proje No:** 110T713

   Proje Adı: Cisim ve Halkalar Üzerinde Tanımlı 2-Boyutlu Hücresel Dönüşümlerin Cebirsel Yapıları ve Davranışları (Bursiyer)