

KODLAMA VE HATA BULMA TEKNİKLERİ

Kodlama elektronik dünyasında çok sık kullanılan, hatta vazgeçilmesi mümkün olmayan bir kavramdır. En basit anlamda kodlama eldeki verinin (sayısal ya da analog olabilir) başka bir birimde gösterilişi olarak adlandırılabilir. Özellikle sayısal sistemlerde ağırlıklı olarak kullanılan bir çok kodlama tekniği mevcuttur. Kodlama teknikleri güvenli veri iletimi, veri sıkıştırma, hata kontrolü ve giderimi, süratli veri işleme gibi bir çok amaç için kullanılmaktadır. Örneğin karakterler 0 ve 1'den okuşan dizilere akarsü dırüşülerek bütün harfler rakamlar ve noktalam işaretleri 0 1 dizileriyle gösterilmektedir. Bu tür kodlamaya alfasayısal kodlama denir. Herkesin yakından bildiği ASCII böyle oluşturulmuş bir kodlamadır. Amaca bağlı olarak sayılar için konumsal kodlama, BCD (ikili kodlanmış onlu) kodlar kullanılırken, ölçmelerde hatayı azaltmak için gray kodu, bilgisayarlarda hatayı bulmak ve azaltmak için eşitlik (parity) ve hamming kodlamaları kullanılmaktadır. Verilen bir veriye ilave bitler katarak bu verilerin saklanması ve yada iletiminde ortaya çıkabilecek hatalar algılanıp otomatik olarak giderilebilir. Bu çeşit kodlamada hata yhapma ihtimalinin yüksek olduğu iletişim sistemlerinde ve gürültülü ortamda çalışan sanay itesislerinde yaygın olarak kullanılmaktadır. Hata bilgiyi temsil eden 0-1 dizisindeki bazı bitlerin değeri deęiştirilmesi olarak yorumlanmalıdır. Hamming kodu hata bulma ve giderme amacıyla kullanılırken, eşitlik biti teknięi yalnız hata algılama için kullanılabilir.

İkili Kodlanmış Onlu (Binary Coded Decimal)

Sayıların iki tabanında gösterilişi, bilgi işlem makinelerinde hesaplama açısından doğruluk ve kolaylık sağlar. Bir bilgisayarda, girilen sayılar üzerinde az sayıda aritmetik işlem yapıp sonuç kullanıcıya onlu biçimde gösterilecekse, o zaman onludan saf ikiliye ve saf ikiliden onluya yapılacak dönüşümlerden kaçınmak için, başka bir deyişle bilgisayarı bu dönüşümlerden kurtarmak amacıyla BCD gösterim kullanılır. Böyle bir çalışma bilgisayarda işlem hızını artırmakla birlikte ilave aritmetik işlemcilerin (yani ilave donanımın) kullanımını gerektirir. Saf ikili gösterimde, hanelerin ağırlıkları ikinin tam kuvvetleri biçiminde sıralanırken, BCD gösterimde her onlu hane kendi içinde ikinin tam kuvvetleri şeklinde sıralanır ama haneler arasında onun tam kuvvetlerine karşı düşen çarpanlar alır. Onlu sayı sisteminde 0,1,2,...,8,9 olmak üzere 10 tane rakam vardır. Bu yüzden her onlu rakam için en az dört bitlik bir ikili sayı gerekir. 4 bit ile $2^4 = 16$ farklı kod oluşturabildiği halde BCD rakamlar bunlardan yalnız 10 tanesini kullanacaklardır.

Oluşan 16 kod sözcüğünden herhangi onu seçilebileceğine göre seçim sonunda oluşturulabilen 10 elemanlı kümelerin sayısı $C = n! / p! (n-p)!$ Formülüne göre $c = 16! / 10! * 6! = 8008$ 'dir.

Ağırlıklı İkili Kodlama

Yukarıda anlatılanlara dayanılarak, BCD kodlar bitlerin konumlarına birer ağırlık vererek oluşturulursa bu ağırlıklı ikili kodlama olur. Yaygın olarak da doğal ikili kodlama kullanılır. Yani ağırlıklar ikinin tam kuvvetleriyle oluşturulmaktadır.

Onlu Sayı	Kod sözcüğü
0	0000
1	0001
2	0010
3	0011 doğal ikili
4	0100 kodlamaya
5	0101 göre geçerli
6	0110 BCD
7	0111 haneler
8	1000
9	1001

Onlu Sayı	Kod sözcüğü
10	1010
11	1011
12	1100
13	1101 doğal ikili
14	1110 kodlamaya
15	1111 göre geçersiz BCD haneler

3 Fazlı Kod

BCD kodlamada kullanılır. Bu kodlamada her onlu sayının 9'a tümleyeni ikili gösterilişinin bit bit tümleyeni olarak bulunur. Burada her sözcüğe 3 eklenmiştir.

Sayı	Kod Sözcüğü	Sayı	Kod Sözcüğü
0	0011	9	1100
1	0100	8	1011
2	0101	7	1010
3	0110	6	1001
4	0111	5	1000

Bitişik Kodlar

Birbirini izleyen sayılara karşılık alınan ikili kod sözcükleri arasındaki uzaklık Hamming'e göre '1' ise, söz konusu sözcükler 'bitişik' bir kodlama oluşturur. Örneğin 0'dan 3'e kadar olan sayıları kodlamada 00, 01, 11, 10, sözcükleri kullanılırsa bitişik bir kodlama yapılmış olur. Çünkü birbirini izleyen her sözcük arasındaki fark '1' dir. Kod sözcüklerinin, '1'.si ile 'sonuncusu' arasındaki uzaklık yine '1' ise kodlama 'çevrimli' olur. Yukarıdaki örnek çevrimli bir kodlamadır.

Gray Kod

2n elamanlı bir küme için 2 tabanında 'artıksız' ve 'çevrimli' bir kodlama yapılırsa Gray Kodu elde edilir. Yani bir kod sözcüğünden diğerine geçişte sadece 1 bit değişmelidir.

Kodlanacak Sayılar	Doğal İkili Kodlama	Gray Kodlama
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1001
14	1111	1000

Bu kodlamanın yapısı gereği b r s zc kten diğereine geiştir bitlerden yalnızca biri deęer deęiştirildiğinden olacan hata en fazla bu s zc klerin deęerleri farkı kadar olur. Bu nedenle  lme tekniğinde sıka kullanılır.

	00	01	11	10
00	0	1	2	3
01	7	6	5	4
11	8	9	10	11
10	15	14	13	12

Karnaugh diyagramı kullanılarak Gray kodun elde edilmesi

Eşitlik

İkili veriler üzerinde işlem g ren sistemler iin aynı anda ancak bir bitin hatalı olabileceđi iki bitin aynı anda hatalı olma olasılığının ihmal edilir d zeyde olduđu varsayımı yaygındır.  z m  basitleştiren bu varsayım gerek hayatta da ođu kez geerliliğini korur. Verinin bir bitinde hata olabileceđi kabul edildiğinde hatanın varlığını ortaya ıkarmak iin kullanılan en yaygın uygulama veriye bir bit daha ekleyip (eşitlik biti) yeni bir kod s zc đ  oluřturmaktır. Eşitlik biti, kodlanan verinin lojik 1 deęerli bitlerinin sayısının tek ya da ift yapılmasını saęlar. B ylece eklenen eşitlik bitiyle n bitlik veri n+1 bitlik kod s zc đ ne d n řm ř olur. Kod s zc đ  alıcıya iletildiğinde, alıcı tarafta incelenerek iletim sırasında hatanın olup olmadıđı algılanmış olur. Seri veri iletiminde ve manyetik bilgi saklama iřlemlerinde eşitlik y ntemi yaygın olarak kullanılır.

 rnek: Ařađıda tek eşitliđe g re oluřturulan tabloda eřitli durumlar g sterimiřtir. Altı izili olanlar hatalı olarak iletilen bitleri kutu iindekiler ise veriye eklenen eşitlik bitini g stermektedir.

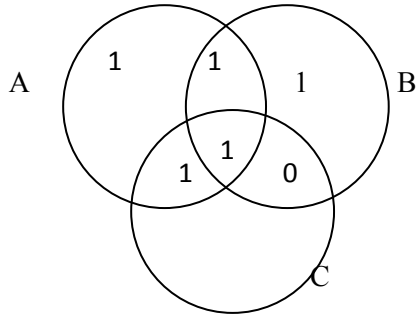
Gönderilen Veri	Alınan Veri	
10110	11110	Çift eşitlik, hata algılandı.
10110	11010	Tek eşitlik, hata algılanamadı.
10011	10010	Çift eşitlik, hata algılandı.
10011	01101	Tek eşitlik, hata algılanamadı

Hamming Kodu

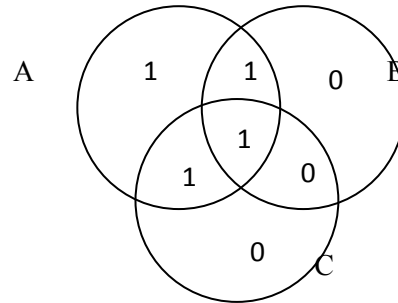
Hammin Kodu kullanılarak hata araştırıldığında aşağıdaki üç sonuçtan biri elde edilir.

- 1) Hata algılanılmadı
- 2) Hata algılandı ve bu hatayı düzeltmek mümkündür.
- 3) Hata algılanır ama hatayı düzeltmek imkansızdır. Bu durum sadece rapor edilir.

Hamming Kodunu anlayabilmek için aşağıdaki Venn diyagramları kullanılsın. Elimizdeki 4 bitlik veriyi Hamming kodlayabilmek için 3 düzeltme biti kullanılır. Bu üç düzeltme bitini A,B,C kümeleri karakterize eder. Eldeki 4 bitlik veriyi de bu kümelerin kesişim notlarına koyalım. (şekil 1)



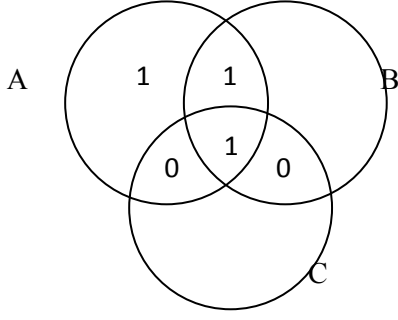
Şekil 1 : Veri bitlerini gösteren A,B,C kümeleri



Şekil 2: düzeltme bitlerinin belirlenmiş hali

Veriyi Hamming kodlayabilmek için 3 düzeltme bitini A,B,C kümelerine bakarak tayin edebiliriz. A kümesinde tek sayıda '1' olduğu için çift eşitlik kavramından giderek, düzeltme biti olarak '1' koyulmasına karar verilir. Yine aynı şekilde B kümesinde çift sayıda '1' olduğu için düzeltme biti '0' olmalıdır. Düzeltme bitinin belirlenmesi basit bir XOR işlemidir(şekil 2).

Hamming kodlu sözcükte bozulma olduğu varsayılınsın (şekil 3). Buna göre; A ve C kümesinde tek sayıda "1" vardır. B kümesi ise normaldir. A ve C kümelerindeki "1"lerin sayısı çift yapılırsa hata giderilmiş olacaktır. Bu da şekil 3'teki taralı alandaki bit "1" yapılarak gerçekleştirilir.



şekil 3: Hatalı bit ve hatanın giderilmesi

Bu kodlamayı bir örnekle inceleyelim;

a_0, a_1, a_2, a_3 biçiminde 4 bitlik verimiz olsun. Bu veriye a_4, a_5, a_6 düzeltme bitleri eklenerek 7 bitlik Hamming kodlu sözcük oluşturulsun.

Düzeltilme bitleri, veri bitlerinden giderek eşitlik hesabıyla şöyle bulunur..

$$a_4 = a_0 \oplus a_1 \oplus a_2$$

$$a_5 = a_1 \oplus a_2 \oplus a_3$$

$$a_6 = a_2 \oplus a_3 \oplus a_0$$

Bu işlem sonunda;

$$a_0, a_1, a_2, a_3, \quad a_4, a_5, a_6$$

Veri bitleri düzeltme bitleri

Hamming kodlu sözcük oluşturulur.

Hamming kodlu sözcük, başka bir ortama iletdikten ya da saklandıktan sonra yeniden okunduğunda elde edilen 7 bitlik dizi $a'0, a'1, a'2, a'3, a'4, a'5, a'6$ olsun.

Alıcı tarafta düzeltme bitlerinin değeri yeniden hesaplanır.

$$a''4 = a'0 \oplus a'1 \oplus a'2 \quad \text{Hesaplanan } a''4, a''5, a''6 \text{ ile alınan ya da okunan}$$

$$a''5 = a'1 \oplus a'2 \oplus a'3 \quad a'4, a'5, a'6 \text{ karşılaştırılır.}$$

$$a''6 = a'2 \oplus a'3 \oplus a'0$$

$S_4 = a'4 \oplus a''4$ Bu hesap sonunda S_4, S_5, S_6 sıfır değeri alırsa hata yoktur. Bu

$S_5 = a'5 \oplus a''5$ yöntemle hata algılandığı gibi hangi bitte hata olduğu algılanıp

$S_6 = a'6 \oplus a''6$ düzeltilir.

Aşağıdaki tabloda her S 'ye ilişkin satırda, o S terimi ile ilgili değişkenlerin karşısına "x" işareti koyulmuştur.

	a0	a1	a2	a3	a4	a5	a6
S_4	x	x	x		x		
S_5		x	x	x		x	
S_6	x		x	x			x

S_4	S_5	S_6	Hatalı bit
0	0	0	Hatalı bit yok
0	0	1	a6
0	1	0	a5
0	1	1	a3
1	0	0	a4
1	0	1	a0
1	1	0	a1
1	1	1	a2

Genel olarak k düzeltici değişken varsa 1 durum hatasızlığı, geri kalan 2^k-1 durum ise hatalı olabilecek bitleri göstermeye yarar. Bu örnekte 3 düzeltici değişken var ve ancak bu değişkenle 8 durum kodlanır. ($2^3=8$). Bu durumlardan biri hatasızlığı geri kalan durumlar ise, 7 bitten hatalı olanını gösterir. Bu açıklanan kod, 'tek hata-düzeltilme' olarak (Single Error Correction-SEC) bilinir ve eğer aynı anda ancak bir bitte hata olmuşsa geçerlidir. Bitlerden 2 tanesinde hata oluşmuşsa Hamming kod hatayı algılar ama düzeltemez. (Çift hata algılama: Double Error Detection, DED). Hata düzeltme, ilave bir lojik gerektirirken, belleğin ya da manyetik saklama ortamının güvenilirliğini iyileştirir. Örneğin; IBM 30XX bilgisayarları ana bellekteki her 64 bitlik data için 8 bitlik bir SEC-DED kodu kullanır. Bu nedenle, ana belleğin gerçek boyu, kullanıcıya görünenden daha büyüktür. VAX bilgisayarlar ise, belleğin her 32 biti için 7 bitlik SEC-DED kullanırlar.

DENEYİN YAPILIŞI

- 1) Öncelikle Exor Entegresi (74LS86) doğru çalıştığından emin olunması için test edilir.
- 2) 4 bitlik veri için a4, a5, a6 değerlerini oluşturunuz.
- 3) Ardından karşı tarafta alınmış veriyi kendiniz hatalı ve hatasız olarak üreterek, hatanın hangi bitte oluştuğunu gözlemleyiniz.

DENEYLE İLGİLİ SORULAR

- 1) Kodlama işlemi nerelerde kullanılır.
- 2) Analog ve dijital işaretin farkları nelerdir?
- 3) Analog işareten dijital işaret nasıl elde edilir?
- 4) İyi kodlamanın özellikleri nelerdir? Avantaj ve dezavantajlarını araştırınız.
- 5) Eşitlik bitinin hata algılayamadığı durumları söyleyin ve buna rağmen neden güvenilir olduğunu açıklayın.
- 6) İlk sayfada 8008 sayısı nerden elde edilmiştir? İrdeleyiniz.
- 7) Hata algılama yöntemlerini karşılaştırınız.