

3- Sayısal Kodlamalar

Sayısal ve Alfa Sayısal Kodlamalar

- Günlük hayatta kullanılan verilerin ifade edilebilmesi, yazınsal ve hesapsal işlemler için kabul görmüş rakam, harf ve noktalama işaretlerini belirten 100'e yakın sembol kullanılır.
- Örneğin büyük ve küçük Latin alfabesi harfleri, On tabanlı sayı sisteminin rakamları v.b.
- Ancak sayısal sistemlerde ise sadece 2 tabanlı sayı sistemi kullanılabildiği için, bütün bu sembollerin bu tabanda kullanılan 0 ve 1'leri kullanarak sisteme tanıtılması gerekir.
- Belirli sayıda 0 ve 1'lerin yan yana yazılarak oluşturulacak kelimelerle (sözcüklerle), günlük hayatta kullanılan sembollerin ifade edilebilmesini sağlayan dönüşüme **kodlama** denilir.
- Kodlamanın sistematik olması için a) Dönüştürülecek sistemde kullanılacak sembollerin var olması, b) Bu sembollerin yan yana getirilmesi ile oluşturulacak sözcüklerin kullanılabilir standartta olması gerekir.
- Örnek olarak 2'lik sistemin sembollerinin (0,1) dizilimiyle oluşturulan ve günlük hayatta kullandığımız alfa sayısal sembolleri ifade eden kodlamaya ikilik kodlama denir. Burada oluşturulacak aynı uzunluktaki her bir sözcüğe, bir alfasayısal sembolün karşı düşeceğini unutmamamız gerekir.

Önemli alfa sayısal kodlar

- **ASCII** (*American National Standard Code for Information Interchange*) kodlama : 7 bitlik bir kodlamadır. Halen kullanılmaktadır. Rakam, harf, noktalama v.b. sembollerin binary sözcüklerle ifade edilmesidir.
- **EBDIC** (*Extended Binary Coded Decimal Interchange Code*) kodlama: 4 bitlik BCD kodlamasının 8 bit'e genişletilmiş şeklidir. IBM geliştirmiştir.

Karakter	7-Bit ASCII			EBCDIC		
	10'luk	16'lık	2'lik	10'luk	16'lık	2'lik
0	48	30	0110000	240	F0	11110000
1	49	31	0110001	241	F1	11110001
9	57	39	0111001	249	F9	11111001
A	65	41	1000001	193	C1	11000001
a	97	61	1100001	129	81	10000001
M	77	4D	1001101	212	D4	11010100
(40	28	010100	77	4D	01001101
@	64	40	1000000	124	7C	01111100
DEL	127	7F	1111111	7	07	00000111

Sayısal Kodlar

- Sadece rakamsal sembollerin kodlanması sayısal kodlamadır. Bir sayısal kodun her basamağı bir ağırlık ifade ediyorsa buna ağırlıklı kodlar denir. Değilse ağırlıksız koddur.
- **BCD (Binary Code Decimal) (8-4-2-1) Kodu.**
- 10 tabanlı sayıların 2'lik tabanda kolayca yazılması için geliştirilmiş ağırlıklı bir kodlamadır. Burada 10 farklı rakamın her bir 4 bitlik 10 tane sözcüğe karşılık düşürülür. Sayılar, bu rakamların yan yana yazılmasıyla oluşturulur.
- Örneğin $(526)_{10} = (0101\ 0010\ 0110)_{BCD}$
- **3-İlave Kodu (Three-excess Decimal –TEC)**
- Tablo 3.2'den de görüldüğü gibi, her bir rakama 3 ilave edilerek 4 bitlik sözcükler şeklinde ifade edilirler. Rakamların 4'er bitlik sözcük karşılıkları yan yana getirilerek sayılar oluşturulur.
- Örneğin $(526)_{10} = (1000\ 0101\ 1001)_{TEC}$
- BCD koduna göre ara sonuç işlemlerini düzeltmek oldukça kolaydır. Bu kodlamada 0-9'un, 1-8'in v.b. 1'e tümlenmiş halidir. Dolayısıyla 1'e tümleme işlemi (9'a tümleme işlemi) kolaylıkla yapılır. Ağırlıksız bir kodlama türüdür.

Tablo 3.2. BCD ve 3-ilave kodundaki sözcükler

On tabanlı rakam	BCD kodu	3-İlave kodu
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

BCD ve 3-İlave kodunda Toplama İşlemi

- 10 tabanlı sayı sistemi tam kodlamalı bir sistem olmadığından, gerek BCD gerekse 3-İlave koduyla kodlanmış sayıların toplanmasında sonuç her zaman doğru olamayabilir. Bunların belirli kurallara göre düzeltilmesi gerekir

BCD kodunda yazılmış rakamların toplanması ve düzeltilmesi kuralları:

- 1- $0 \leq \text{Ara sonuç} \leq 9$ ise doğrudur. Düzeltilmeye gerek yoktur.
- 2- $9 < \text{Ara sonuç} \leq 15$ ise yanlışdır. Düzeltilmek için ara sonuç 6 ile toplanır. Oluşan elde biti bir üst basamağa verilir. İşlem basamağındaki sonuç düzeltilmiştir.
- 3- $\text{Ara sonuç} > 15$ ise yanlışdır. Oluşan elde biti bir üst basamağa verilip, kalan 6 (0110) ile toplanır. İşlem basamağındaki sonuç düzeltilmiştir.

Örnekler

$$\begin{array}{r}
 5 \quad 0000 \, 0101 \\
 + 3 \quad 0000 \, 0011 \\
 \hline
 1000 \, 8 \quad 0000 \, 1000 \quad \text{Ara sonuç} \\
 \text{Doğru}
 \end{array}$$

$$\begin{array}{r}
 9 \quad 0000 \, 1001 \\
 + 5 \quad 0000 \, 0101 \\
 \hline
 00010100 \quad 14 \quad 0000 \, 1110 \quad \text{Ara sonuç} \quad \text{Yanlış}
 \end{array}
 \begin{array}{l}
 \nearrow \\
 \begin{array}{r}
 0000 \, 1110 \\
 + \quad 0110 \\
 \hline
 0001 \, 0100 \quad \text{Düzeltilmiş sonuç}
 \end{array}
 \end{array}$$

$$\begin{array}{r}
 8 \quad 0000 \, 1000 \\
 + 9 \quad 0000 \, 1001 \\
 \hline
 00010111 \quad 17 \quad 0001 \, 0001 \quad \text{Ara sonuç} \quad \text{Yanlış}
 \end{array}
 \begin{array}{l}
 \nearrow \\
 \begin{array}{r}
 0001 \, 0001 \\
 + \quad 0110 \\
 \hline
 0001 \, 0111 \quad \text{Düzeltilmiş sonuç}
 \end{array}
 \end{array}$$

3-İlave kodunda yazılmış rakamaların toplanması ve düzeltilmesi

- 1- Ara sonuç 4 bitlik ise yanlıştır. Düzeltmek için 3(0011) çıkarılır veya 13 (1101) eklenir.
Oluşan elde biti gözardı edilir.
- 2- Ara sonuç 5 bitlik ise yanlıştır. Düzeltmek için 3(0011) ilave edilir. Oluşan elde biti üst basamağa verilir.

Örnekler

$$\begin{array}{r} 05 \\ +03 \\ \hline 08 \end{array} \quad \begin{array}{r} 0011 \ 1000 \\ + \ 0011 \ 0110 \\ \hline 0110 \ 1110 \text{ Ara sonuç 4 bit; Yanlış} \end{array}$$

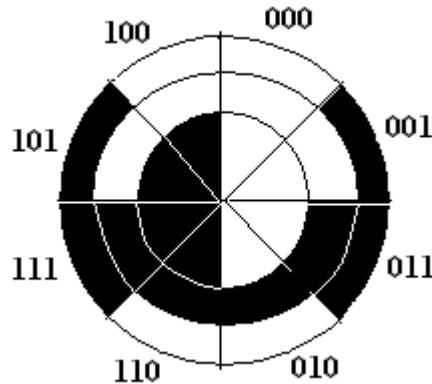
$$\begin{array}{r} 0110 \ 1110 \\ + 1101 \ 1101 \\ \hline \cancel{10011}1011 \text{ Düzeltilmiş sonuç} \end{array}$$

$$\begin{array}{r} 07 \\ +09 \\ \hline 16 \end{array} \quad \begin{array}{r} 0011 \ 1010 \\ +0011 \ 1100 \\ \hline 0110 \ 10110 \text{ Ara sonuç 5 bit; Yanlış} \\ + \underline{1} \\ 0111 \end{array}$$

$$\begin{array}{r} 0111 \ 0110 \\ + 1101 \ 0011 \\ \hline \cancel{10100} \ 1001 \text{ Düzeltilmiş sonuç} \end{array}$$

Gray kodu

- Bir sayısal kod'da ardışıl rakamları belirten sayısal sözcüklerden, birinden diğerine geçilirken sözcüklerdeki sadece 1 bit değer değiştiriyorsa ve bu özellik birinci ve sonuncu sözcük arasında da mevcutsa bu kodlama, *bitişimli ve çevrimsel bir kodlamadır*.
- Böyle kodlamaların matematiksel bir anlamı olmamakla birlikte, pratikte uygulamalarda en az hatalı bit oluşumu açısından önem kazanmaktadırlar. GRAY kodları da bu sınıftan bir kodlamadır.
- Bu kodlama ile 0 ve 2^n-1 arasındaki ardışıl sayılar kodlanır. 2,3,4,5,...n bitlik Gray kodları oluşturulabilir. Şekilde Gray koduna göre kodlanmış bir disk görülmektedir.



Binary-Gray Dönüşümü

Gray kodlanmış sayıların akılda tutulması zahmetlidir. Bu yüzden dönüşümün pratik algoritmasını bilmek yeterlidir. Bu algoritmayı $(10111001101)_2$ sayısının Gray kodu karşılığının bulunması ile açıklayalım.

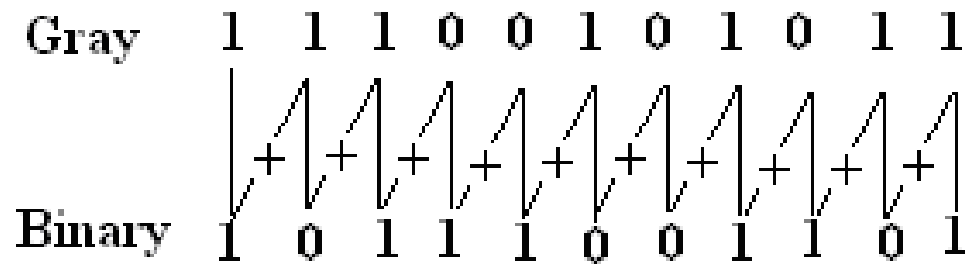
- 1- Binary sayının MSB biti GRAY sözcüğün MSB biti olarak alınır.
- 2- Bundan sonra soldan sağa doğru, ardarda gelen bitler 2'şer 2'şer toplanıp, elde biti atılır sonuç gray kodunun basamak değeridir.

Binary	1	0	1	1	1	0	0	1	1	0	1
		+	+	+	+	+	+	+	+	+	
Gray	1	1	1	0	0	1	0	1	0	1	1

Gray-Binary Dönüşümü

- Gray kodundaki sözcüğün Binary karşılığını bulmak için aşağıdaki pratik algoritma kullanılır. Bu dönüşümü $(11100101011)_{\text{gray}}$ sayısının Binary karşılığının bulunmasıyla açıklayalım

- 1- Gray kodundaki sözcüğün MSB biti Binary sözcüğün MSB bitidir.
- 2- Binary sözcüğün n . değerli biti ile gray nsözcüğün $(n-1)$.değerli biti toplanır.Elde gözönüne alınmaz, toplam sonucu Binary sözcüğün $(n-1)$. bitidir.



Onluk sayı	2 bit Gray kodu	3 bit gray kodu	4 bit gray kodu
0 (0000)	00	000	0000
1 (0001)	01	001	0001
2 (0010)	11	011	0011
3 (0011)	10	010	0010
4 (0100)		110	0110
5 (0101)		111	0111
6 (0110)		101	0101
7 (0111)		100	0100
8 (1000)			1100
9 (1001)			1101
10(1010)			1111
11(1011)			1110
12(1100)			1010
13(1101)			1011
14(1110)			1001
15(1111)			1000

Hata bulan ve düzelten kodlar

- Sayısal olarak kodlanmış sözcükler, gerek iletilirken gerek saklanırken az da olsa bozulma ihtimalleri olabilir.
- Bu bozulma genellikle 1 bit veya 2 bit'teki bozulma şeklindedir.
- İşte bu bozulan bitleri tanıyan veya düzeltebilen kodlamalar da mevcut olup, ilerideki konularda az da olsa üzerinde durulacaktır.