1. ŠTEVILSKI SESTAVI

1.1 DESETIŠKI (DECIMALNI) ŠTEVILSKI SESTAV

Desetiški številski sestav ima osnovo 10, ker uporabljamo deset cifer (od 0 do 9) in ker so koeficienti cifer množeni s potencami osnove 10. Neko desetiško število lahko napišemo tudi na daljši način, kot npr.:

$$1984 = 1 \cdot 10^3 + 9 \cdot 10^2 + 8 \cdot 10^1 + 4 \cdot 10^0$$

Dogovor pri pisanju števil je torej v tem, da pišemo le koeficiente k potencam, vrednost (težo) pa ugotovimo iz mesta koeficienta v številu.

1.2 DVOJIŠKI (BINARNI) ŠTEVILSKI SESTAV

V digitalni tehniki uporabljamo dvojiški oz. binarni številski sestav. Koeficienti tega sestava imajo samo dve vrednosti in sicer 0 ali 1, vsak koeficient pa je pomnožen z 2^i.

$$11010_2 = 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 16 + 8 + 0 + +2 + 0 = 26_{10}$$

Da bi ločili števila z različnimi osnovami, jim kot indeks pripišemo osnovo.

$$\mathbf{11011{,}011_2 = 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-2} + 1 \cdot 2^{-3} = 16 + 8 + 2 + 1 + \frac{1}{4} + \frac{1}{8} = 27\frac{3}{8} = 27{,}375_{10}}$$

Torej pretvorimo binarno število v desetiško tako, da napišemo vsoto potenc koeficientov z vrednostjo 1.

Pretvarjanje desetiškega števila v binarno ali kakšno drugo število z osnovo r naredimo tako, da število razdelimo na celi del števila ter njegov ulomčni del. Nato opravimo pretvorbo za vsak del posebej.

1.2.1 Pretvorba celega števila

Možna sta dva načina pretvorbe. Po prvem iščemo zaporedoma najvišje potence z osnovo 2, ki so manjše od desetiškega števila.

$43 = 2^5 + 2^3 + 2^1 + 2^0 \qquad 43_{10} = 1\,0\,1\,01\,1_2$

Druga metoda je bolj sistematična. Desetiško število delimo z 2 in napišemo celoštevilčni količnik, ostanek je lahko le 0 ali 1. Nato celoštevilčni količnik ponovno delimo z 2 in zopet napišemo celoštevilčni količnik in ostanek. To ponavljamo tako dolgo, dokler ne dobimo količnik 0. Ostanki, ki smo jih dobili pri deljenju sestavljajo binarno število. Zapišemo jih v nasprotnem vrstnem redu (glej primer).

					ostanek
43	:	2	=	21	**1**
21	:	2	=	10	**1**
10	:	2	=	5	**0**
5	:	2	=	2	**1**
2	:	2	=	1	**0**
1	:	2	=	0	**1**

1 0 1 0 1 1

$43_{10} = 101011_2$

1.2.1 Pretvorba decimalnega dela števila

Decimalni del števila pretvorimo v binarno tako, da ga množimo z 2. Če je rezultat manjši od 1, zapišemo 0, dobljen rezultat ponovno množimo z 2. Če je rezultat večji od 1, zapišemo 1, decimalni ostanek pa ponovno množimo z 2. To ponavljamo tako dolgo, dokler ne dobimo na decimalnih mestih same 0, oziroma da je vrednost produkta točno 1. Če pa točno 1 ne moremo dobiti, ponavljamo množenje tako dolgo, da dosežemo želeno točnost.

0,3125	.	2	=	**0**,625
0,625	.	2	=	**1**,25
0,25	.	2	=	**0**,50
0,50	.	2	=	**1**,000

$0{,}3125_{10} = 0{,}0101_2$

0,35	.	2	=	**0**,70
0,70	.	2	=	**1**,40
0,40	.	2	=	**0**,80
0,80	.	2	=	**1**,60
0,60	.	2	=	**1**,20
0,20	.	2	=	**0**,40
0,40	.	2	=	**0**,80
0,80	.	2	=	**1**,6o

$0{,}35_{10} = 0{,}01011001\ldots_2$

1.3 OSMIŠKI (OKTALNI) ŠTEVILSKI SESTAV

Osmiški (oktalni) številski sestav ima osnovo 8 in koeficiente oz. cifre od 0 do 7, utežne vrednosti so potence števila 8. Ker je $2^3 = 8$, ustreza vsaka oktalna enota trem binarnim. Zato pri pretvarjanju binarnega števila v osmiško, binarno število razdelimo od desne proti levi v skupine po tri enote ter poiščemo ustrezno osmiško vrednost.

$$1\,1\,0\,0\,1\,0\,0\,1\,1_2 = \underbrace{1\,1\,0}_{6}\;\;\underbrace{0\,1\,0}_{2}\;\;\underbrace{0\,1\,1}_{3} = 6\,2\,3_8$$

Pretvorbo v nasprotni smeri, to je iz osmiškega v binarno število izvedemo po nasprotnem postopku kot prej. Vsako osmiško cifro pretvorimo v trimestno binarno cifro.

$$4\,5\,7_8 = 1\,0\,0\;\;1\,0\,1\;\;1\,1\,1 = 100101111_2$$

Osmiško število pretvorimo v desetiško takole:

$$2\,4\,6_8 = 2 \cdot 8^2 + 4 \cdot 8^1 + 6 \cdot 8^0 = 128 + 32 + 6 = 166_{10}$$

Postopek pretvorbe iz desetiškega števila v osmiško je enak postopku pretvorbe desetiškega števila v binarno, le da desetiško število delimo oziroma množimo z 8.

$199{,}5_{10} = ?$ ostanek

199	:	8	=	24	**7**		0,5	·	8	=	**4**,0
24	:	8	=	3	**0**						
3	:	8	=	0	**3**						

$1\,9\,9,\,5_{10} = 3\,0\,7,\,4_8$

1.4 ŠESTNAJSTIŠKI (HEKSADECIMALNI) ŠTEVILSKI SESTAV

Šestnajstiški (heksadecimalni) številski sestav ima osnovo 16 in koeficiente oz. cifre od 0 do 15, utežne vrednosti so potence števila 16. Zaradi razločnosti pri pisanju so dvomestne številke zamenjane s črkami.

A = 10 B = 11 C = 12 D = 13 E = 14 F = 15

Ker je $2^4 = 16$, ustreza vsaka šestnajstiška enota štirim binarnim. Zato pri pretvarjanju binarnega števila v šestnajstiško, binarno število razdelimo od desne proti levi v skupine po štiri enote ter poiščemo ustrezno šestnajstiško vrednost.

$1110100110_2 = 0011\ 1010\ 0110 = 3A6_{16}$

3 A 6

Šestnajstiško število spremenimo v binarno tako, da vsaki šestnajstiški oznaki priredimo binarno število.

$F3B4_{16} = 1111\ 0011\ 1011\ 0100 = 1111001110110100_2$

Šestnajstiško število pretvorimo v desetiško takole:

$2C8_{16} = 2 \cdot 16^2 + 12 \cdot 16^1 + 8 \cdot 16^0 = 512 + 192 + 8 = 712_{10}$

Pretvorbo iz desetiškega v šestnajstiško naredimo z zaporednim deljenjem (množenjem) s 16.

$199{,}5_{10} = ?$

					ostanek						
199	:	16	=	12	**7**		0,5	·	16	=	**8**,0
12	:	16	=	0	**12=C**						

$199{,}5_{10} = C7{,}8_{16}$

1.5 ARITMETIČNE OPERACIJE

Osnovna pravila za binarni številski sestav so:

Za seštevanje:	0	+	0	=	0	Za množenje:	0	·	0	=	0
	0	+	1	=	1		0	·	1	=	0
	1	+	0	=	1		1	·	0	=	0
	1	+	1	=	10		1	·	1	=	1

Seštevanje

$$\begin{array}{ccccccc} & 1 & 0 & 1 & 1 & 0 & 1 \\ + & 1 & 0 & 0 & 1 & 1 & 1 \\ \hline 1 & 0 & 1 & 0 & 1 & 0 & 0 \end{array}$$

Odštevanje

$$\begin{array}{ccccccc} & 1 & 0 & 1 & 1 & 0 & 1 \\ - & 1 & 0 & 0 & 1 & 1 & 1 \\ \hline & 0 & 0 & 0 & 1 & 1 & 0 \end{array}$$

Množenje

$$\begin{array}{|c|c|c|c|c|c|c|c|} 1 & 0 & 1 & 1 & \times & 1 & 0 & 1 \end{array}$$

$$\begin{array}{cccccccc} 1 & 0 & 1 & 1 & \times & 1 & 0 & 1 \\ \hline & 0 & 0 & 0 & 0 & & & \\ & & 1 & 0 & 1 & 1 & & \\ \hline 1 & 1 & 0 & 1 & 1 & 1 & & \end{array}$$

1.6 KOMPLEMENTARNE VREDNOSTI ŠTEVIL

Za številčni sestav z osnovo r poznamo dve vrsti komplementarnih vrednosti oz. skrajšano komplementov:

r komplement

$r^n - N$ za $N \neq 0$

N … pozitivno število

r … osnova

n … število celih mest

r-1 komplement

$r^n - r^{-m} - N$

m … število ulomčnih mest

Če je osnova 2, dobimo dvojiški in eniški komplement. Poiščimo dvojiški in eniški komplement za število 101100_2.

$N = 101100$

$N_{2K} = 2^6 - 101100 = 1000000 - 101100 = 0010100$ dvojiški komlement

$N_{1K} = 2^6 - 2^0 - 101100 = 1000000 - 1 - 101100 = 010011$ eniški komlement

Vidimo, da eniški komplement dobimo enostavno **z zamenjavo enk z ničlami in ničel z enkami**. Iz definicije komplementa tudi sledi, da dobimo dvojiški komplement tako, da eniškemu komplementu prištejemo 1 na zadnjem mestu.

Pravo binarno vrednost dobimo nazaj iz eniškega komplementa tako, da enke zamenjamo z ničlami in ničle z enkami.

Iz dvojiškega komplementa dobimo pravo binarno vrednost tako, da najprej naredimo eniški komplement in temu na zadnjem mestu prištejemo 1.

Oba komplementa uporabljamo v digitalnih sistemih in računalništvu **za odštevanje**.

1.7 PREDZNAČENA ŠTEVILA

V računalništvu je dvojiški zapis osnova vsem predstavitvam. V dvojiškem ali binarnem zapisu imenujemo eno binarno mesto ali eno števko (cifro) enoto binarne informacije ali ***bit*** *(**bi**nary digi**t**)*. Glede na število bitov ločimo med drugim naslednje tipe podatkov:

- bajt (angl. Byte) je skupina osmih bitov

- beseda (angl. Word) je skupina bitov (običajno dva bajta)

Nepredznačena števila zapišemo v računalniški bajt enostavno kot naravno binarno število.

$169_{10} = 10101001_2$

MSB LSB

1	0	1	0	1	0	0	1
7	6	5	4	3	2	1	0

MSB (**M**ost **S**ignificant **B**it)… najbolj pomemben bit oz. bit z največjo težo

LSB (**L**east **S**ignificant **B**it)… najmanj pomemben bit oz. bit z najmanjšo težo

Cifre od 0 do 7 (v splošnem od **0** do **n-1**) pomenijo dvojiške uteži i (2^i). Če število nima uporabljenih vseh osem mest, manjkajoče zapolnimo z vodilnimi ničlami. Z nepredznačenimi števili označujemo pomnilniške naslove, vrednosti registrov…

Predznačena števila, ki potrebujejo še informacijo o predznaku, pa zapisujemo na naslednje načine:

- **Zamenjava predznaka:** zapis nepredznačenega celega števila dopolnimo tako, da mu dodamo predznak. Predznak **0 za pozitivne** in **1 za negativne** vrednosti se nahaja na mestu bita z največjo utežjo (MSB). Vrednost števila pri tem ne spremenimo. Tako npr. števili +345 in -345 zapišemo na naslednji način:

MSB LSB

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

+ predznak

MSB LSB

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

- predznak

Tovrstni način zapisovanja predznačenih števil se redko uporablja.

- **Zapis predznačenega števila z dvojiškim komplementom:** zapisana pozitivna števila imajo najvišji bit vedno 0, negativna pa 1.

Zgled: Zapišimo +25 in -25 v računalniški bajt.

+ 25 =	**0**	0	0	1	1	0	0	1		MSB je 0, pomeni, da je število pozitivno.
	1	1	1	0	0	1	1	0		
								1		
– 25 =	**1**	1	1	0	0	1	1	1		MSB je 1, pomeni, da je število negativno.

Največje pozitivno in negativno (po absolutni vrednosti) predznačeno število zapisano v bajt je:

$+N_{max}$ = | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | = + 1 2 7 **$+N_{max} = 2^{n-1} - 1$**

$-N_{max}$ = | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | = − 1 2 8 **$-N_{max} = 2^{n-1}$**

n…..število bitov

1.8 ODŠTEVANJE S POMOČJO DVOJIŠKEGA KOMPLEMENTA

Direktna metoda odštevanja za računalnik ni najbolj primerna. Bolj enostavno je delati s komplementi števil. V bistvu je odštevanje poseben primer seštevanja:

$A - B = A + (-B)$

Odštevanje dveh predznačenih števil opravimo tako, da negativnemu številu naredimo dvojiški komplement, dobimo vsoto. Če pride do končnega prenosa, ga zanemarimo. Če je MSB 0, je rezultat pozitivno število, ki ga lahko direktno pretvorimo v desetiško. Če pa je MSB 1, je rezultat negativno število. Naredimo dvojiški komplement dobljene vsote in dodamo znak minus.

8 – 3 = 8 + (– 3) = 5

```
+3  =  0 0 0 0 0 0 1 1            0   0 0 0 1 0 0 0
       1 1 1 1 1 1 0 0       +    1   1 1 1 1 1 0 1
                     1           [0]  0 0 0 0 1 0 1  =  +5
–3  =  1 1 1 1 1 1 0 1           MSB
```

– 12 + 9 = (– 12) + 9 = – 3

```
+12  =  0 0 0 0 1 1 0 0           1   1 1 1 0 1 0 0
        1 1 1 1 0 0 1 1      +    0   0 0 0 1 0 0 1
                      1          [1]  1 1 1 1 1 0 1
–12  =  1 1 1 1 0 1 0 0          MSB

                                  0   0 0 0 0 0 1 0
                                                  1
                                  0   0 0 0 0 0 1 1  =  – 3
```

Pri seštevanju dveh predznačenih števil z enakim predznakom lahko dobimo rezultat nasprotnega predznaka. Temu pravimo *preliv*.

+ 64	0 1 0 0 0 0 0 0	– 127	1 0 0 0 0 0 0 1
+ 96	0 1 1 0 0 0 0 0	– 3	1 1 1 1 1 1 0 1
+ 160	**1** 0 1 0 0 0 0 0	– 130	$\boxed{1}$ **0** 1 1 1 1 1 1 0
	–		+

prenos zanemarimo

Bit, ki označuje predznak se je spremenil. To pa zaradi tega, ker +160 in –130 ne moremo zapisati samo z osmimi biti. Rešitev je, da obe števili izrazimo kot 16 bitni predznačeni števili.

2. KODIRANJE

Kodiranje je prirejanje števil, črk in nekaterih posebnih znakov v obliko, ki je primerna za predstavitev v binarnem sistemu. Gre za transformacijo informacije iz ene oblike v drugo po določenem pravilu, ki mu pravimo tudi *kodirno pravilo*.

Vsako informacijo, ki jo posredujemo digitalni napravi, je treba predhodno binarno kodirati, ker lahko digitalni sistemi neposredno sprejmejo le binarno obliko informacije.

Za posredovanje desetiških števil digitalnim napravam potrebujemo binarno kodirana decimalna števila. Od tod izvira skupno ime za pravila kodiranja. Tem pravilom pravimo **BCD kode** (**B**inary **C**oded **D**ecimal).

Za binarno kodiranje desetiških cifer, ki jih je deset, moramo imeti na voljo vsaj 10 različnih možnosti. Zato pa potrebujemo 4 bite. Glede na to, da lahko s štirimi biti kodiramo 16 cifer oz. znakov, jih je pri predstavitvi 10 cifer 6 odveč (število možnosti = 2^n, n je število bitov). To nam daje številne možnosti za tvorbo različnih kod.

2.1 UTEŽNOSTNE BCD KODE

Vse BCD kode, pri katerih ima položaj bita določeno številsko vrednost, imenujemo *utežnostne kode*, ker je vsak bit utežen s to vrednostjo. Kot najbolj preprosta možnost se ponuja kar pretvorba desetiškega števila v binarno, saj vemo, da so v binarnem številskem sistemu mesta utežena z 2^i, šteto od desne proti levi. Tej kodi pravimo ***naravna BCD koda*** ali tudi 8 – 4 – 2 – 1 koda, kar ustreza utežem na posameznih mestih.

- *Kodiranje*: Vsaki desetiški števki priredimo ustrezno uteženo skupino štirih binarnih števk.

  $785{,}3_{10} = 0111\ 1000\ 0101,\ 0011_{BCD}$

- *Dekodiranje*: Kodirano število razdelimo levo in desno od decimalne vejice v skupine s 4 biti in te skupine pretvorimo v desetiško vrednost.

  $0001\,\vdots\,1001\,\vdots\,1000\,\vdots\,0010{,}0101_{BCD} = 1982{,}5_{10}$

2.1.1 BCD seštevanje

Seštejmo dve števili:

$$\begin{array}{r} 23 \\ +15 \\ \hline 38 \end{array} \qquad\qquad \begin{array}{r} 0010\ 0011 \\ 0001\ 0101 \\ \hline 0011\ 1000 \end{array}$$

Seštevamo po pravilih binarnega seštevanja. Rezultat je pravilen, ker vsota znotraj skupine 4 bitov ni večja od 9. Povečajmo sedaj drugo decimalno število za 2:

23 + 17 = 40

$$
\begin{array}{rl}
 & 0010\ 0011 \\
+ & \underline{0001\ 0111} \\
 & 0011\ \underbrace{1010}_{\text{ni naravna BCD koda}}
\end{array}
$$

Druga skupina v seštevku ne pripada naravni BCD kodi. Številska vrednost te skupine je 10, kar pomeni, da je potrebno na nižjem mestu potrebno vpisati samo kodirano 0, 1 pa prenesti na naslednje višje mesto. Prenos enice na višje mesto je možen tudi s prištevanjem binarnega števila 6 vsem tistim skupinam, pri katerih je vsota večja od 9.

$$
\begin{array}{rll}
 & 0010\ 0011 & \\
+ & \underline{0001\ 0111} & \\
 & 0011\ 1010 & \\
 & \underline{\quad\ +\ 0110} & \text{prištejemo binarno 6 (0110)} \\
 & 0100\ 0000 &
\end{array}
$$

Tako smo dobili pravilen rezultat. Seštejmo sedaj naslednji dve števili:

28 + 19 = 47

$$
\begin{array}{ll}
0010\ 1000 & \\
\underline{0001\ 1001} & \\
0100\ 0001 \leftarrow & \text{Prišlo je do prenosa v višjo tetrado (tetrada je skupina 4 bitov)} \\
\underline{\quad\ +\ 0110} & \\
0100\ 0111 & \text{Prišteli smo 6 (0110) in dobili smo pravilen rezultat.}
\end{array}
$$

Torej je potrebno prišteti 6 tudi v primeru, če pride do prenosa v višjo tetrado.

2.2 NEUTEŽNOSTNE KODE

Skupna značilnost teh kod je, da posamezno mesto znotraj skupine nima stalne vrednosti, kar pomeni, da ni uteženo. Neutežnostne kode običajno delimo še v dve podskupini:

- *Reflektivne kode* so tiste pri katerih se skupine, ki so enako oddaljene od horizontalne simetrale, razlikujejo vedno le za en bit.
- *Enokoračne kode* so tiste, pri katerih se dve sosedni skupini razlikujeta za enako število bitov, običajno je to le en bit.

2.2.1 Grayeva koda

Grayeva koda je hkrati *reflektivna* in *enokoračna*. Zelo pogosto se uporablja povsod tam, kjer se v digitalni obliki prenaša informacija o fizikalnih veličinah (npr. AD pretvorba geometrijskega kota v digitalni zapis).

- *Kodiranje* je zelo enostavno. Pred bit z največjo težo (skrajni levi) pripišemo ničlo ali pa to enico enostavno prepišemo, nato pa med posameznimi mesti tvorimo vsoto po modulu 2 (vsota po modulu 2 ima vrednost 1, če sta seštevanca <u>različna</u> in 0, če sta <u>enaka).</u>

$$
\begin{array}{llcccccccc}
99_{10} = 1100011_2 & \quad & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1_2} \\
 & & & \oplus\ \ \mathbf{1} & \oplus\ \ \mathbf{0} & \oplus\ \ \mathbf{1} & \oplus\ \ \mathbf{0} & \oplus\ \ \mathbf{0} & \oplus\ \ \mathbf{1} & \oplus\ \ \mathbf{0_{Gray}}
\end{array}
$$

- *Dekodiranje* pa poteka takole: prvo enico na skrajni levi (MSB) prepišemo, potem pa tvorimo vsoto po modulu 2, kot je razvidno iz slike spodaj.

$$
\begin{array}{ccccccccccccccl}
\mathbf{1} & & \mathbf{0} & & \mathbf{1} & & \mathbf{0} & & \mathbf{0} & & \mathbf{1} & & \mathbf{0}_{\mathrm{Gray}} & & \\
\downarrow & \oplus & & \oplus & & \oplus & & \oplus & & \oplus & & \oplus & & & \\
\mathbf{1} & & \mathbf{1} & & \mathbf{0} & & \mathbf{0} & & \mathbf{0} & & \mathbf{1} & & \mathbf{1_2} & & \qquad \textbf{rezultat}
\end{array}
$$

2.3 ALFANUMERIČNE KODE

To so kode, ki poleg številk vključujejo tudi črke in druge znake. Najbolj znana je *ASCII* koda (American Standard for Information Interchange). V osnovi je to 7 bitna koda, torej je mogoče v njej tvoriti 128 (2^7) različnih besed ali kodirati 128 različnih znakov. Kodo v tabeli čitamo tako, da najprej pogledamo zgornje bite (b_7, b_6, b_5) in poiščemo stolpec, ki ima to kombinacijo bitov. Na levi strani poiščemo še kombinacijo spodnjih bitov (b_4, b_3, b_2, b_1) in na ustreznem mestu odčitamo črtko, številko ali znak. Prvih 32 kod (od 0 do 31) in 127 predstavlja kontrolne znake, ki jih nikoli ne vidimo na zaslonu. Ostalih 95 kod predstavlja številke, velike in male črke ter posebne znake.

Obstaja tudi razširjena 8 bitna ASCII koda, ki jo je prvi uporabil IBM v svojih PC-jih. Ima še dodatnih 128 znakov (tuje abecede, simboli za denarne enote, grške črke, matematični simboli,…)

Decimalno število	**BCD kode**															
	Naravna BCD koda				**Aiken koda**											
	8	**4**	**2**	**1**	**2**	**4**	**2**	**1**	**4**	**2**	**2**	**1**	**8**	**4**	**-2**	**-1**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1	0	0	0	1	0	1	1	1
2	0	0	1	0	0	0	1	0	0	0	1	0	0	1	1	0
3	0	0	1	1	0	0	1	1	0	0	1	1	0	1	0	1
4	0	1	0	0	0	1	0	0	1	0	0	0	0	1	0	0
5	0	1	0	1	1	0	1	1	0	1	1	1	1	0	1	1
6	0	1	1	0	1	1	0	0	1	1	0	0	1	0	1	0
7	0	1	1	1	1	1	0	1	1	1	0	1	1	0	0	1
8	1	0	0	0	1	1	1	0	1	1	1	0	1	0	0	0
9	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1

Tabela 2.1 Nekatere utežnostne BCD kode

Decimalno število	**BCD reflektivna koda**			
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	1	1	0	0
6	1	0	1	1
7	1	0	1	0
8	1	0	0	1
9	1	0	0	0

Horizontalna simetrala

Tabela 2.2 Princip reflektivne kode

Decimalno število	**3 bitna Gray-eva koda**			**4 bitna Gray-eva koda**			
0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	1
2	0	1	1	0	0	1	1
3	0	1	0	0	0	1	0
4	1	1	0	0	1	1	0
5	1	1	1	0	1	1	1
6	1	0	1	0	1	0	1
7	1	0	0	0	1	0	0
8				1	1	0	0
9				1	1	0	1
A				1	1	1	1
B				1	1	1	0
C				1	0	1	0
D				1	0	1	1
E				1	0	0	1
F				1	0	0	0

Tabela 2.3 3 in 4 bitna Gray-eva koda

	$b_7b_6b_5$	$b_7b_6b_5$	$b_7b_6b_5$	$b_7b_6b_5$	$b_7b_6b_5$	$b_7b_6b_5$	$b_7b_6b_5$	$b_7b_6b_5$
$b_4b_3b_2b_1$	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
0 0 0 0	NUL	DLE	SP	0	@	P	`	p
0 0 0 1	SOH	DC1	!	1	A	Q	a	q
0 0 1 0	STX	DC2	"	2	B	R	b	r
0 0 1 1	ETX	DC3	#	3	C	S	c	s
0 1 0 0	EOT	DC4	$	4	D	T	d	t
0 1 0 1	ENQ	NAK	%	5	E	U	e	u
0 1 1 0	ACK	SYN	8	6	F	V	f	v
0 1 1 1	BEL	ETB	′	7	G	W	g	w
1 0 0 0	BS	CAN	(	8	H	X	h	x
1 0 0 1	HT	EM	)	9	I	Y	i	y
1 0 1 0	LF	SUB	*	:	J	Z	j	z
1 0 1 1	VT	ESC	+	:	K	[	k	{
1 1 0 0	FF	FS	.	<	L	\	l	|
1 1 0 1	CR	GS	-	=	M	]	m	}
1 1 1 0	SO	RS	.	>	N	^	n	~
1 1 1 1	SI	US	/	?	O	_	o	DEL

Tabela 2.4 7 bitna ASCII koda