Digitalni zapis podataka

Stefan Mišković

4 Binarno kodirani dekadni brojevi

4.1 Osnovna svojstva

Budući da se nekad pri prevođenju razlomljenih ili mešovitih brojeva iz dekadne u binarni sistem može dogoditi da se rezultat ne može zapisati konačnim brojem cifara, u računaru se nekad sâm dekadni broj ne može zapisati sa stoprocentnom tačnošću. Ukoliko je taj uslov neophodan, pribegava se principu zapisa kojim se svaka dekadna cifra kodira kao određen binarni broj. Na taj način se dobijaju takozvani binarni kodovi dekadnih cifara, skraćeno BCD kod.

BCD kod predstavlja bilo koju funkciju koja svaku dekadnu cifru preslikava u neki niz binarnih cifara. U opštem slučaju kod može biti i različitih dužina za različite brojeve, a može i imati istu vrednost za različite dekadne cifre. Ovde ćemo se koncentrisati na one kodove koji imaju istu dužinu binarne vrednosti za svaku dekadnu cifru. Budući da imamo 10 različitih dekadnih cifara, najmanja moguća dužina koda je 4, što će u nastavku i biti slučaj. Dodatno, funkcija kodiranja će biti 1-1, odnosno kod će biti jednoznačan, čime će različitim dekadnim ciframa odgovarati različiti binarni kodovi.

Pored jednoznačnosti, poželjno je da BCD kod poseduje i neke od sledećih svojstava:

- Parnost. Ako je kod paran, parnim ciframa odgovaraju parni kodovi, a neparnim ciframa odgovaraju neparni kodovi.
- Komplementarnost. Neka su a i b cifre takve da je $a + b = 9$. Kod je komplementaran ako su bitovi koda za a i bitovi koda za b na odgovarajućim pozicijama komplementarne, u smislu da zbirovi cifara na istim pozicijama tih kodova daju rezultat 1.
- Da bude težinski. Ako je x dekadna cifra koja se kodira, a $x_3x_2x_1x_0$ njen BCD kod, kod je težinski ukoliko postoje celi brojevi c_3, c_2, c_1 i c_0 takvi da je $x = c_0x_0 + c_1x_1 + c_2x_2 + c_3x_3$. Brojeve c_3, c_2, c_1 i c_0 nazivamo težinama.
- Najvećoj dekadnoj cifri se pridružuje kod koji, posmatran kao binarni broj, ima najveću vrednost.

Neki primeri BCD kodova su:

- 8421. Ovaj zapis se dobija tako što se dekadna cifra prevede u binarni broj, a zatim se eventualno dodaju vodeće nule do dužine zapisa (dužina ovog zapisa, kao i ostalih koji će biti pomenuti, iznosi 4).

- Višak 3. Ovaj zapis se dobija tako što se na svaku dekadnu cifru doda vrednost 3, a zatim dobijeni broj prevede u binarni sistem i dopišu odgovarajuće nule do dužine 4.

- Ciklički kod. Ovaj zapis podrazumeva da se zapis za svake dve susedne cifre razlikuje za tačno jedan bit. Pritom se i 0 i 9 smatraju susednim ciframa. Bilo koji kod koji zadovoljava ovo svojstvo se može nazvati cikličkim.

U narednoj tabeli su prikazani zapisi 8421, višak 3 i ciklički kod. Za ciklički kod prikazana je jedna od mogućih varijanti, za koju se zapis susednih cifara razlikuje za jedan bit. Iz tabele se može videti da su svi kodovi jednoznačni. Kod 8421 je paran i težinski, a nije komplementaran. Kod višak 3 je komplementaran, a nije paran i težinski. Za oba koda važi da cifra 9 ima najveći binarni broj za zapis. Navedeni ciklički kod ne zadovoljava nijednu od pomenutih dodatnih osobina.

Cifra	8421	Višak 3	Ciklički kod
0	0000	0011	0001
1	0001	0100	0101
2	0010	0101	0111
3	0011	0110	1111
4	0100	0111	1110
5	0101	1000	1100
6	0110	1001	1000
7	0111	1010	1001
8	1000	1011	1011
9	1001	1100	0011

4.2 Sabiranje i oduzimanje u BCD kodu

Pretpostavimo da su dva dekadna broja zapisana u nekom od zapisa u BCD kodu i da je potrebno izračunati njihov zbir ili razliku. Ako se radi o označenim brojevima, oni se mogu, slično pravilima za sabiranje i oduzimanje u znaku i apsolutnoj vrednosti, uvek svesti na sabiranje dva pozitivna broja ili oduzimanje manjeg pozitivnog broja od većeg, pri čemu na kraju jedino treba voditi računa o znaku rezultata. Na primer, ako se sabiraju dva broja istog znaka, dovoljno je sabrati dva odgovarajuća pozitivna broja, a rezultat je onog znaka kao sabirci. Sa druge strane, ako se vrši sabiranje dva broja različitog znaka, sabiranje se može svesti na oduzimanje, gde se oduzima broj sa manjom apsolutnom vrednošću od broja sa većom. Slična situacija je i kod oduzimanja dva označena broja. Zbog toga možemo u nastavku pretpostaviti da vršimo sabiranje dva neoznačena broja ili oduzimanje dva neoznačena broja, od kojih je prvi veći po apsolutnoj vrednosti.

Ako su dva n-tocifrena neoznačena cela dekadna broja $A = a_{n-1}a_{n-2}\dots a_1a_0$ i $B = b_{n-1}b_{n-2}\dots b_1b_0$ zapisana u BCD kodu, za njihovo kodiranje bilo kojim od standardnih pomenutih zapisa (poput, na primer, 8421 ili višak 3) je potrebno $4n$ bitova. Neka je α funkcija koja predstavlja preslikavanje dekadne cifre u konkretan BCD zapis. Kodovi $\alpha(A) = \alpha(a_{n-1})\alpha(a_{n-2})\dots\alpha(a_1)\alpha(a_0)$ i $\alpha(B) = \alpha(b_{n-1})\alpha(b_{n-2})\dots\alpha(b_1)\alpha(b_0)$ su nizovi od $4n$ bitova i predstavljaju zapise brojeva A i B, dok $\alpha(a_i)$ i $\alpha(b_i)$ predstavlja četvorku bitova koja kodira jednu dekadnu cifru.

Sabiranje $\alpha(A)+\alpha(B)$ se vrši u dve faze. U prvoj fazi se vrši sabiranje dva neoznačena binarna broja, pri čemu se dobija međurezultat:

$$
\begin{array}{r}
\alpha(a_{n-1})\alpha(a_{n-2})\ldots\alpha(a_1)\alpha(a_0) \\
+\ \alpha(b_{n-1})\alpha(b_{n-2})\ldots\alpha(b_1)\alpha(b_0) \\
\hline
\alpha(c'_{n-1})\alpha(c'_{n-2})\ldots\alpha(c'_1)\alpha(c'_0)
\end{array}
$$

Dobijeni međurezultat je potrebno korigovati, budući da dobijene četvorke bitova $\alpha(c'_i)$ ne predstavljaju tačne zapise zbira. Zbog toga se u drugoj fazi vrši dodavanje korekcije $\alpha(k_i)$ za svaku grupu bitova $\alpha(c'_i)$. Konačan rezultat se dobija u drugoj fazi, sabiranjem međurezultata i broja koji se dobija nadovezivanjem korekcija:

$$
\begin{array}{r}
\alpha(c'_{n-1})\alpha(c'_{n-2})\ldots\alpha(c'_1)\alpha(c'_0) \\
+\ \alpha(k_{n-1})\alpha(k_{n-2})\ldots\alpha(k_1)\alpha(k_0) \\
\hline
\alpha(c_{n-1})\alpha(c_{n-2})\ldots\alpha(c_1)\alpha(c_0)
\end{array}
$$

Prva faza je identična za sve zapise, a u drugoj fazi način određivanja korekcije zavisi od vrste zapisa. Oduzimanje se može raditi na analogan način kao sabiranje, s tim što se u obe faze umesto sabiranja binarnih brojeva vrši njihovo oduzimanje. Drugi način da se oduzmu dva broja je da se zapišu u potpunom komplementu, a zatim svedu na sabiranje promenom znaka umanjiocu. U nastavku će biti opisani konkretni algoritmi sabiranja i oduzimanja u zapisima 8421 i višak 3.

Sabiranje u zapisu 8421. Neka su data dva neoznačena cela broja sa n cifara $A = a_{n-1}a_{n-2}\ldots a_1a_0$ i $B = b_{n-1}b_{n-2}\ldots b_1b_0$, čiji su BCD kodovi u zapisu 8421 $\alpha(A) = \alpha(a_{n-1})\alpha(a_{n-2})\ldots\alpha(a_1)\alpha(a_0)$ i $\alpha(B) = \alpha(b_{n-1})\alpha(b_{n-2})\ldots\alpha(b_1)\alpha(b_0)$. Potrebno je izračunati zbir $\alpha(A) + \alpha(B)$, gde funkcija α svakoj cifri dodeljuje četvorobitni kod u skladu sa zapisom 8421.

Prva faza sabiranja je identična kao opisana prva faza u opštem slučaju, gde se vrši sabiranje dva neoznačena binarna broja od $4n$ cifara. Neka $p'_i \in \{0,1\}$ označava prenos sa i-te četvorke bitova na narednu četvorku bitova. Na primer, p'_1 označava eventualni prenos sa četvrtog na peti par pitova gledajući zdesna, a p'_2 označava eventualni prenos sa osmog na deveti par pitova gledajući zdesna. Na početku je uvek $p'_0 = 0$.

Druga faza sabiranja je specifična samo za zapis 8421. Konkretno, specifičan je način na koji se određuju korekcije pri sabiranju. Najpre se određuje korekcija za četvorku bitova $\alpha(c'_0)$. Ako je $\alpha(c'_0) \geq (1010)_2$ ili je $p'_1 = 1$, korekcija je $\alpha(k_0) = (0110)_2$. Inače, korekcija je $\alpha(k_0) = (0000)_2$. Izvršimo sada sabiranje $\alpha(c'_0) + \alpha(k_0)$ i neka je $p''_1 \in \{0,1\}$ odgovarajući prenos. Za određivanje korekcije za četvorku bitova $\alpha(c'_1)$, dovoljno je proveriti da li je $\alpha(c'_1) + p''_1 \geq (1010)_2$ ili je $p'_2 = 1$. Ukoliko je bilo koji od ova dva uslova ispunjen, korekcija je $\alpha(k_1) = (0110)_2$, a inače je $\alpha(k_1) = (0000)_2$. Ovaj postupak se ponavlja, gde se redom dobijaju korekcije $\alpha(k_2), \alpha(k_3), \ldots, \alpha(k_{n-1})$ i prenosi $p''_3, p''_4, \ldots, p''_n$. Bitovi p''_{i+1} predstavljaju dakle prenos pri sabiranju četvorki bitova $\alpha(c'_i)$ i $\alpha(k_i)$. Uzima se da je $p''_0 = 0$.

Na osnovu prethodno opisanog, može se izvesti pravilo za određivanje korekcije $\alpha(k_i)$ za četvorku bitova međurezultata $\alpha(c'_i)$:

- Ako je $\alpha(c'_i) + p''_i \geq (1010)_2$, korekcija je $\alpha(k_i) = (0110)_2$.
- Ako je $p'_{i+1} = 1$, korekcija je $\alpha(k_i) = (0110)_2$.
- Inače, ako nijedan od prethodna dva uslova nije ispunjen, korekcija je $\alpha(k_i) = (0000)_2$.

Sabiranjem međurezultata i korekcije, dobija se rezulat $\alpha(c_{n-1})\alpha(c_{n-2})\ldots\alpha(c_1)\alpha(c_0)$. Prekoračenje pri sabiranju se javlja ako je $p'_n = 1$ ili $p''_n = 1$. Sabiranje u 8421 se šematski može prikazati na sledeći način:

$$
\begin{array}{c||c||c||c||c||c}
 & \alpha(a_{n-1}) & \alpha(a_{n-2}) & \ldots & \alpha(a_1) & \alpha(a_0) \\
+ & \alpha(b_{n-1}) & \alpha(b_{n-2}) & \ldots & \alpha(b_1) & \alpha(b_0) \\
\hline
p'_n & p'_{n-1} & p'_{n-2} & \ldots & p'_1 & p'_0 \\
 & \alpha(c'_{n-1}) & \alpha(c'_{n-2}) & \ldots & \alpha(c'_1) & \alpha(c'_0) \\
p''_n & p''_{n-1} & p''_{n-2} & \ldots & p''_1 & p''_0 \\
+ & \alpha(k_{n-1}) & \alpha(k_{n-2}) & \ldots & \alpha(k_1) & \alpha(k_0) \\
\hline
 & \alpha(c_{n-1}) & \alpha(c_{n-2}) & \ldots & \alpha(c_1) & \alpha(c_0)
\end{array}
$$

Svaka kolona, odvojena dvostrukom uspravnom crtom predstavlja četvorku bitova, pri čemu su prenosi p'_i i p''_i desno poravnati. U prvoj fazi se dobija međurezultat

$$\alpha(c'_{n-1})\alpha(c'_{n-2})\ldots\alpha(c'_1)\alpha(c'_0)$$

i prenosi p'_i. Druga faza se izvodi u n koraka. Svaki korak se sastoji od određivanja korekcije $\alpha(k_i)$, sabiranja $\alpha(c'_i) + \alpha(k_i) = \alpha(c_i)$ i određivanje prenosa p''_i.

Na primer, ako je dekadne brojeve 23492 i 5189 potrebno sabrati u BCD kodu u zapisu 8421 na 5 mesta, ceo postupak izgleda ovako (broju 5189 se dodaju vodeće nule do 5 mesta):

```
  0010 0011 0100 1001 0010
+ 0000 0101 0001 1000 1001
--------------------------
0    0    0    1    0    0
  0010 1000 0110 0001 1011
0    0    0    0    1    0
+ 0000 0000 0000 0110 0110
--------------------------
  0010 1000 0110 1000 0001
```

Rešenje je 28681. Kako su poslednji prenosi u obe faze jednaki 0, nije došlo do prekoračenja. Radi lakšeg razumevanja prethodnog primera, prikažimo kako se on radi postupno. Najpre se u prvoj fazi odradi sabiranje prva dva binarna broja, dobije rezultat i upišu odgovarajući prvi prenosi:

```
  0010 0011 0100 1001 0010
+ 0000 0101 0001 1000 1001
--------------------------
0    0    0    1    0    0
  0010 1000 0110 0001 1011
```

Zatim se druga faza radi korak po korak. Nulti prenos u drugoj fazi se postavi na nulu, odredi se korekcija, a zatim se sabiraju poslednja četiri bita međurezultata sa korekcijom i pritom dobijaju prva četiri bita rezultata i prvi prenos u drugoj fazi:

```
  0010 0011 0100 1001 0010
+ 0000 0101 0001 1000 1001
--------------------------
0    0    0    1    0    0
  0010 1000 0110 0001 1011
                    1    0
+                     0110
                    ------
                      0001
```

Nakon toga se određuje druga korekcija, pa vrši sabiranje odgovarajuće druge četvorke bitova, nakon čega se određuju naredna četiri bita rezultata i drugi prenos u drugoj fazi:

```
  0010 0011 0100 1001 0010
+ 0000 0101 0001 1000 1001
--------------------------
0    0    0    1    0    0
  0010 1000 0110 0001 1011
               0    1    0
+                0110 0110
               -----------
                 1000 0001
```

Isti postupak se ponavlja naredna tri koraka, nakon čega se završava druga faza i dobija konačan rezultat.

Oduzimanje u zapisu 8421. Neka su uvedene iste oznake kao kod sabiranja u zapisu 8421. Oduzimanje se radi na sličan način kao sabiranje, a osnovna razlika je što se umesto sabiranja u obe faze vrši oduzimanje:

$$
\begin{array}{c||c||c||c||c||c}
 & \alpha(a_{n-1}) & \alpha(a_{n-2}) & \ldots & \alpha(a_1) & \alpha(a_0) \\
- & \alpha(b_{n-1}) & \alpha(b_{n-2}) & \ldots & \alpha(b_1) & \alpha(b_0) \\
\hline
p'_n & p'_{n-1} & p'_{n-2} & \ldots & p'_1 & p'_0 \\
 & \alpha(c'_{n-1}) & \alpha(c'_{n-2}) & \ldots & \alpha(c'_1) & \alpha(c'_0) \\
- & \alpha(k_{n-1}) & \alpha(k_{n-2}) & \ldots & \alpha(k_1) & \alpha(k_0) \\
\hline
 & \alpha(c_{n-1}) & \alpha(c_{n-2}) & \ldots & \alpha(c_1) & \alpha(c_0)
\end{array}
$$

Ovde nisu od značaja prenosi u drugoj fazi, pa će biti izostavljeni. Korekcija $\alpha(k_i)$ za $\alpha(c'_i)$ se određuje jedino na osnovu vrednosti pozajmice p'_{i+1}. Ako je $p'_{i+1} = 1$, tada je $\alpha(k_i) = (0110)_2$, a ako je $p'_{i+1} = 0$, tada je $\alpha(k_i) = (0000)_2$. Slično kao i kod sabiranja, na početku se uvek postavlja $p'_0 = 0$. Budući da se uvek oduzimaju neoznačeni celi brojevi, takvi da je umanjilac manji od umanjenika, kod oduzimanja nikad ne dolazi do prekoračenja.

Na primer, neka je potrebno izvršiti oduzimanje brojeva 52629 i 2634, ako su oni zapisani u BCD kodu u zapisu 8421 na 5 mesta. Nakon što se umanjilac dopuni jednom vodećom nulom, oduzimanje se može prikazati na sledeći način:

```
  0101 0010 0110 0010 1001
- 0000 0010 0110 0011 0100
```

```
-------------------------
0    1    1    1    0    0
  0100 1111 1111 1111 0101
- 0000 0110 0110 0110 0000
-------------------------
  0100 1001 1001 1001 0101
```

Rešenje je 49995. Primetimo da prenosi u drugoj fazi nisu ni zapisivani, jer, za razliku od sabiranja u zapisu 8421, ovde nisu od značaja. Redosled izvođenja operacija je takođe pravolinijski – najpre se oduzmu dva broja u prvoj fazi, a zatim odrede prenosi. Nakon toga se na osnovu prenosa odrede vrednosti korekcija, i na kraju se u drugoj fazi može odjednom izvršiti oduzimanje međurezultata i broja dobijenog na osnovu korekcija.

Oduzimanje u zapisu 8421 se može svesti i na sabiranje, ali je potrebno da se oba broja zapišu u potpunom komplementu. Tada se umanjilac komplementira, nakon čega se izvrši sabiranje u kodu 8421 umanjenika i komplementiranog umanjioca. Na primer, neka je potrebno izvršiti oduzimanje brojeva 9463 i 6423, ako su oni zapisani na 5 mesta. Oduzimanje se može vršiti opisanim algoritmom (s tim što treba kod oba broja dodati vodeću nulu, jer je u zadatku naglašeno da je zapis na 5 mesta), a može se vršiti i svođenjem na potpuni komplement. Vrednosti brojeva u potpunom komplementu su 09463 i 06423. Komplementirani umanjilac je 93577. Nakon toga je potrebno primeniti algoritam za sabiranje brojeva 09463 i 93577 u zapisu 8421. Kada se četvorke bitova rezultata prevedu u dekadne cifre, on je u potpunom komplementu i treba ga nazad prevesti u dekadni broj (budući da će rezultat uvek biti pozitivan, ovo će se svoditi na brisanje vodeće nule za znak broja). Ukoliko se oduzimanje na ovaj način svodi na sabiranje, nikad ne može doći do prekoračenja. Ovo je posledica činjenice da su brojevi zapisani u potpunom komplementu, u kome se kod sabiranja poslednji prenosi uvek ignorišu. Primetimo još i da je potrebno da u zadatku bude naglašeno da su brojevi zapisani na bar jedno mesto više od njihovog broja cifara u dekadnom zapisu, da bi oduzimanje moglo da se izvrši. Inače, ne može se izvršiti svođenjem na potpuni komplement, jer nema mesta za zapis cifre za znak.

Sabiranje u zapisu višak 3. Neka su uvedene iste oznake kao kod sabiranja u zapisu 8421. Sabiranje u višku 3 se radi na sličan način kao sabiranje u zapisu 8421:

	$\alpha(a_{n-1})$	$\alpha(a_{n-2})$	$\dots$	$\alpha(a_1)$	$\alpha(a_0)$
$+$	$\alpha(b_{n-1})$	$\alpha(b_{n-2})$	$\dots$	$\alpha(b_1)$	$\alpha(b_0)$
p'_n	p'_{n-1}	p'_{n-2}	$\dots$	p'_1	p'_0
	$\alpha(c'_{n-1})$	$\alpha(c'_{n-2})$	$\dots$	$\alpha(c'_1)$	$\alpha(c'_0)$
$+$	$\alpha(k_{n-1})$	$\alpha(k_{n-2})$	$\dots$	$\alpha(k_1)$	$\alpha(k_0)$
	$\alpha(c_{n-1})$	$\alpha(c_{n-2})$	$\dots$	$\alpha(c_1)$	$\alpha(c_0)$

Prva faza je identična kao kod sabiranja u 8421, a druga faza je karakteristična za ovaj zapis. Korekcija $\alpha(k_i)$ za deo međurezultata $\alpha(c'_i)$ se određuje na osnovu vrednosti prenosa p'_{i+1}. Ako je $p'_{i+1} = 1$, tada je $\alpha(k_i) = (0011)_2$, a ako je $p'_{i+1} = 0$, tada je $\alpha(k_i) = (1101)_2$. Na početku se uvek postavlja $p'_0 = 0$. U drugoj fazi, kada se vrši sabiranje jedne četvorke međurezultata sa korekcijom, ignoriše se prenos na narednu četvorku (kao kada sabiramo dva binarna broja u potpunom komplementu, zapisana na 4 mesta). Zbog toga prenosi u drugoj fazi i ne postoje. Prekoračenje kod sabiranja u višku 3 se javlja ako je $p'_n = 1$.

Na primer, neka je potrebno izvršiti sabiranje u višku 3 brojeva 28367 i 2847, ako su oni zapisani na 5 mesta. Nakon određivanja njihovih kodova u višku 3, algoritam izgleda ovako:

```
  0101 1011 0110 1001 1010
+ 0011 0101 1011 0111 1010
--------------------------
0    1    1    1    1    0
  1001 0001 0010 0001 0100
+ 1101 0011 0011 0011 0011
--------------------------
  0110 0100 0101 0100 0111
```

Rešenje je 31214. Do prekoračenja nije došlo, budući da je poslednji prenos jednak 0.

Oduzimanje u zapisu višak 3. Oduzimanje u zapisu višak 3 se svodi na sabiranje. Pritom je potrebno da se oba broja zapišu u potpunom komplementu. Tada se umanjilac komplementira, nakon čega se izvrši sabiranje u višku 3 umanjenika i komplementiranog umanjioca. Na primer, neka je potrebno izvršiti oduzimanje brojeva 9463 i 6423, ako su oni zapisani na 5 mesta. Vrednosti brojeva u potpunom komplementu su 09463 i 06423. Komplementirani umanjilac je 93577. Nakon toga je potrebno primeniti algoritam za sabiranje brojeva 09463 i 93577 u zapisu višak 3. Kada se četvorke bitova rezultata prevedu u dekadne cifre, on je u potpunom komplementu i treba ga nazad prevesti u dekadni broj (budući da će rezultat uvek biti pozitivan, ovo će se svoditi na brisanje vodeće nule za znak broja). Pritom u ovakvom sabiranju, na koje se svodi oduzimanje, nikad ne može doći do prekoračenja. Ovo je posledica činjenice da su brojevi zapisani u potpunom komplementu, u kome se kod sabiranja poslednji prenosi uvek ignorišu. Primetimo još i da je potrebno da u zadatku bude naglašeno da su brojevi zapisani na bar jedno mesto više od njihovog broja cifara u dekadnom zapisu. Inače, oduzimanje ne bi moglo da se izvrši, jer nema mesta za zapis cifre za znak.