UVOD U ORGANIZACIJU I ARHITEKTURU RAČUNARA

Binarno kodirani dekadni brojevi (BCD)

Danijela Simić

Binarno kodirani dekadni brojevi (BCD)

Binarno kodirani dekadni brojevi (BCD) je binarno–kodirana reprezentacija decimalnih vrednosti. U BCD-u, svaka decimalna cifra je predstavljena koristeći fiksni broj binarnih bitova. Tipično, četiri binarna bita se koriste za predstavljanje svake decimalne cifre, što omogućava kodiranje vrednosti od 0 do 9. BCD je široko korišćen metod za čuvanje i obradu decimalnih brojeva u digitalnim sistemima, naročito u aplikacijama gde je precizna decimalna reprezentacija ključna, kao što su finansijski proračuni i digitalni displeji.

Kratka istorija i evolucija

Istorija BCD-a može se pratiti unazad do ranih dana računarstva kada je prepoznata potreba za reprezentacijom koja bi efikasno obradila decimalne brojeve. Sredinom 20. veka, s pojavom digitalnih računara, koncept BCD-a stekao je na značaju.

- **Rani BCD Kodovi:** Rane forme BCD-a postojale su pre modernog 8-4-2-1 BCD koda. Ovi kodovi koristili su različite reprezentacije za kodiranje decimalnih cifara. Međutim, nisu bili standardizovani, što je dovelo do problema sa kompatibilnošću.
- **Razvoj 8-4-2-1 BCD:** 8-4-2-1 BCD kod, poznat i kao 8421 BCD, postao je široko prihvaćen standard. U ovom kodu, svaka decimalna cifra je predstavljena 4-bitnim binarnim kodom. Ovaj razvoj značajno je poboljšao efikasnost i doslednost BCD reprezentacije.
- **Integracija u hardver:** BCD kodiranje postalo je sastavni deo mnogih digitalnih sistema, uključujući kalkulatore, blagajne i rane računare. Njegovo usvajanje bilo je motivisano potrebom za tačnim decimalnim proračunima u oblastima kao što su finansije i inženjering.

Dok je binarni kod bio idealan zbog elektronske prirode ovih uređaja, predstavljao je izazov kada je trebalo reprezentovati decimalne brojeve.

Uvođenje BCD-a u digitalne računare poslužilo je kao kompromis. Omogućilo je ovim sistemima da elektronski funkcionišu koristeći binarnu logiku, dok je sačuvana jednostavnost i tačnost decimalne reprezentacije. Kao rezultat toga, rani računari poput IBM 1401, popularni šezdesetih godina prošlog veka, intenzivno su koristili BCD za različite zadatke.

BCD u Evoluciji Hardvera

Dok je važnost BCD-a rasla, proizvođači hardvera počeli su da integrišu BCD-specifične operacije u svoje proizvode:

Integrisana kola (ICs): Sa revolucijom miniaturizacije sedamdesetih godina prošlog veka i dolaskom integrisanih kola, počele su se pojavljivati specijalizovane BCD aritmetičke jedinice. Ove jedinice su dizajnirane da efikasno rukuju BCD operacijama.

Displeji i korisnički interfejsi: Korišćenje BCD-a nije bilo ograničeno samo na računske zadatke. Imao je ključnu ulogu u displejima, poput 7-segmentnog displeja, omogućavajući jednostavnu i direktnu reprezentaciju decimalnih brojeva.

Pad upotrebe

Kako je tehnologija računarske obrade napredovala i opšte namenski procesori postajali snažniji, potreba za BCD-specifičnim operacijama se smanjivala. Čisto binarne operacije bile su brže i efikasnije za većinu zadataka. Međutim, BCD je i dalje zadržavao (i još uvek zadržava) važnost u specifičnim namenama. Finansijski sistemi, na primer, koji zahtevaju preciznu decimalnu reprezentaciju da bi izbegli greške zaokruživanja, često koriste BCD. Štaviše, mnogi nasleđeni sistemi koji su još uvek u upotrebi oslanjaju se na BCD, čineći ga relevantnim čak i u modernoj eri.

Istorija BCD-a nudi fascinantan uvid u izazove i rešenja koja su ponuđena u povezivanju sistema koji su razumljivih čoveku sa efikasnim računarskim sistemima. Čak i kako tehnologija napreduje, BCD ostaje kao svedočanstvo inovacija iz prošlosti.

Finansijski i monetarni proračuni: BCD naspram čisto binarnog

Scenario:

Zamislite finansijsku aplikaciju, poput bankarskog sistema, koja svakodnevno obrađuje milione transakcija. Svaka transakcija uključuje decimalne iznose, i tačnost do poslednjeg centa je imperativna. Čak i sitne greške, kada se propagiraju kroz veliki broj transakcija, mogu dovesti do značajnih neslaganja i potencijalnih finansijskih gubitaka.

Problem sa čisto binarnim:

Greške zaokruživanja: Binarna reprezentacija sa pokretnim zareyom, koja se koristi za decimalne brojeve u većini računarskih sistema, često ne može tačno predstaviti određene decimalne brojeve. Na primer, decimalni broj 0.1 se ne može tačno predstaviti u binarnoj reprezentaciji sa pokretnim zarezom. Kada se radi sa finansijskim podacima, takve aproksimacije mogu dovesti do grešaka.
$(0.1)_{10} = (0.00011001100110011....)_2$

Nagomilane greške: Proračuni koji se ponavljaju koristeći brojeve sa malim nepreciznostima mogu akumulirati greške. Vremenom, ili sa velikim obimom proračuna, ove greške mogu postati značajne.

Troškovi konverzije: Iako je unutrašnja reprezentacija možda u binarnom formatu, korisnički interfejsi, izveštaji i razmene podataka često zahtevaju decimalne vrednosti. Pretvaranje između binarnog i decimalnog može uvesti dodatne greške zaokruživanja.

Prednosti BCD-a u ovom kontekstu:

Tačna decimalna reprezentacija: BCD može tačno predstaviti decimalne brojeve. Dakle, vrednosti poput 0.1, 0.01, i tako dalje, koje su uobičajene u finansijskim kontekstima, predstavljene su i čuvane bez grešaka u aproksimaciji.

Pojednostavljena konverzija: Pošto se svaka grupa od četiri bita u BCD direktno mapira na decimalnu cifru, konverzija BCD vrednosti u decimalni broj koji ljudi mogu čitati je jednostavna i bez grešaka.

Predvidljiva aritmetika: Kada se koristi BCD, finansijski sistemi mogu osigurati dosledne i predvidljive rezultate. Na primer, kada se dodaju dve BCD vrednosti, rezultat je zagarantovano tačna decimalna suma te dve vrednosti.

Primer iz stvarnog sveta:

Reprezentacija broja 0.1 je $0.1000000000000000055511151231257827021181583404541015625$ (ovo je stvarna aproksimalcija kada se koristi dvostruka tačnost u pokretnom zarezu).

Recimo da softverski sistem banke svakodnevno obrađuje 10 miliona transakcija.

Ako svaka transakcija pretrpi čak i malu grešku od 0.01 (cent) zbog nepreciznosti binarne reprezentacije, banka bi mogla suočiti neslaganja od 100.000 dolara dnevno, što je ekvivalentno 36.5 miliona dolara godišnje. Korišćenjem BCD osigurava se da je svaka transakcija tačna do poslednjeg centa, eliminišući takve potencijalne gubitke.

Struktura i reprezentacija BCD

Binarni kodirani decimalni (BCD) je sistem koji predstavlja svaki decimalni broj njegovim odgovarajućim binarnim ekvivalentom. Međutim, za razliku od čistih binarnih sistema gde je binarna vrednost kontinuirana reprezentacija celog broja, BCD razlaže broj na cifru po cifra.

U BCD-u, svaka pojedinačna decimalna cifra, u rasponu od 0 do 9, predstavljena je grupom od četiri binarna bita. Reprezentacije su sledeće:

Decimalna 0: 0000
Decimalna 1: 0001
Decimalna 2: 0010
Decimalna 3: 0011
Decimalna 4: 0100
Decimalna 5: 0101
Decimalna 6: 0110
Decimalna 7: 0111
Decimalna 8: 1000
Decimalna 9: 1001

Kao što možemo primetiti, BCD reprezentacija za decimalne brojeve od 0 do 9 se poklapa sa standardnim binarnim vrednostima za te brojeve. Međutim, značaj BCD-a postaje očigledan kada predstavljamo brojeve sa više cifara.

Na primer, decimalni broj 19 bio bi predstavljen u čistom binarnom obliku kao 10011.

U BCD-u, ovaj broj je razbijen na njegove pojedinačne cifre, 1 i 9. Koristeći BCD reprezentaciju, 1 je 0001, a 9 je 1001. Kombinujući ih, 19 u BCD-u je 00011001.

Prednosti ovakve reprezentacije:

1. **Čitljivo za ljude:** Jedna od osnovnih prednosti BCD-a je njegova čitljivost za ljude. Budući da svaka decimalna cifra ima svoju jedinstvenu 4-bitnu reprezentaciju, čitanje i tumačenje BCD vrednosti postaje intuitivno. Nema potrebe za složenom binarno-decimalnom konverzijom kao kod čistih binarnih reprezentacija.

2. **Lakoća konverzije:** Konverzija između BCD-a i dekadne vrednosti je jednostavna, zahteva samo grupisanje ili deljenje svaka četiri bita.

3. **Preciznost u decimalnim proračunima:** BCD osigurava da se decimalne vrednosti predstavljaju bez grešaka zaokruživanja, što je ključno za aplikacije koje zahtevaju tačne decimalne proračune, poput finansijskih sistema.

4. **Kompatibilnost sa digitalnim displejima:** Mnogi digitalni displeji, poput 7-segmentnih displeja, dizajnirani su da prikazuju decimalne cifre. BCD je prirodno kompatibilan s ovim displejima, omogućavajući lako povezivanje i tačnu reprezentaciju broja.

Vrste BCD kodova

Zapis korišćenjem binarno kodiranih dekadnih brojeva (BCD) doživeo je različite adaptacije da bi zadovoljio različite zahteve i scenarije. Iako osnovna ideja ostaje konstantna – reprezentacija decimalnih cifara binarnim kodovima – različiti BCD kodovi nude jedinstvene karakteristike koje su korisne u specifičnim situacijama.

8-4-2-1 BCD (najčešći tip):

Ovo je standardna BCD reprezentacija s kojom je većina ljudi upoznata. Svaka decimalna cifra je predstavljena njenim odgovarajućim 4-bitnim binarnim brojem. Naziv ″8-4-2-1″ukazuje na binarnu težinu svake pozicije:

Vrednost	8-4-2-1 BCD reprezentacija
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Upotreba:

Zbog svoje jednostavnosti i direktnog mapiranja na decimalne brojeve, široko se koristi u aplikacijama koje se kreću od jednostavnih kalkulatora do složenih računarskih uređaja.

Primer: $56.01 = 01010110.00000001$

2-4-2-1 BCD kôd

Vrednost	2-4-2-1 BCD reprezentacija
0	0000
1	0001
2	0010
3	0011
4	0100
5	1011
6	1100
7	1101
8	1110
9	1111

Primena: 2-4-2-1 kod ima istorijski značaj zbog njegove upotrebe u nekim ranim računarskim i brojčanim sistemima, naročito u određenim mehaničkim računarima i abakusima.

U dobu savremene digitalne elektronike i računanja, upotreba 2-4-2-1 je umanjena. Kod se i dalje danas koristi u digitalnim satovima i malim kalkulatorima.

Višak 3 kôd

Višak 3 je BCD kod u kojem je svaka decimalna cifra predstavljena svojom odgovarajućom 4-bitnom binarnom reprezentacijom plus pomakom od 3. Drugim rečima, binarna reprezentacija za svaki broj je broj plus tri.

Vrednosti u kodu višak 3:

vrednost	reprezentacija višak 3	objašnjenje
0	0011	(0 + 3 = 3 u binarnom)
1	0100	(1 + 3 = 4 u binarnom)
2	0101	(2 + 3 = 5 u binarnom)
3	0110	(3 + 3 = 6 u binarnom)
4	0111	(4 + 3 = 7 u binarnom)
5	1000	(5 + 3 = 8 u binarnom)
6	1001	(6 + 3 = 9 u binarnom)
7	1010	(7 + 3 = 10 u binarnom)
8	1011	(8 + 3 = 11 u binarnom)
9	1100	(9 + 3 = 12 u binarnom)

Primena:

- **Pojednostavljenje u konstrukciji hardvera:** Upotreba koda višak 3 u određenim računarskim hardverskim sistemima (poput ranih digitalnih računara ili kalkulatora) je zbog činjenice da može da pojednostavi određene aritmetičke operacije. Na primer, sabiranje dva broja u ovom sistemu ponekad može biti izvedeno bez potrebe za prenosom.

- **Samo-komplementiranje:** Zanimljiva karakteristika koda višak 3 je što je samo-komplementirajući. To znači da se 9-ti komplement decimalnog broja može pronaći jednostavnim invertovanjem (promenom 1 u 0 i obrnuto) njegovog višak 3 koda. Ovo svojstvo ga je učinilo korisnim u nekim operacijama oduzimanja koristeći komplementarnu aritmetiku.

U savremenim računarskim sistemima, kod višak 3 se ređe koristi, prvenstveno zbog napretka tehnologije i efikasnijih binarnih aritmetičkih operacija. Međutim, razumevanje koda višak 3 je vredno za one koji proučavaju istoriju računarstva i dizajn digitalne logike jer pruža uvid u metode binarne aritmetike koji su nekada korišćeni.

Neki binarni kodovi dekadnih cifara:

Dekadna cifra	Binarni kod						
	8421	2421	5421	753-6	84-2-1	višak 3	ciklički
0	0000	0000	0000	0000	0000	0011	0001
1	0001	0001	0001	1001	0111	0100	0101
2	0010	0010	0010	0111	0110	0101	0111
3	0011	0011	0011	0010	0101	0110	1111
4	0100	0100	0100	1011	0100	0111	1110
5	0101	1011	1000	0100	1011	1000	1100
6	0110	1100	1001	1101	1010	1001	1000
7	0111	1101	1010	1000	1001	1010	1001
8	1000	1110	1011	0110	1000	1011	1011
9	1001	1111	1100	1111	1111	1100	0011

Pakovani i nepakovani zapis BCD brojeva

Pakovani zapis BCD brojeva

U formatu pakovanog BCD zapisa, dve BCD cifre kombinuu se ili ˝pakujužajedno u jedan bajt. Bajt se sastoji od 8 bita, pa u ovom formatu, leva 4 bita (najznačajnija četvorka) predstavljaju jednu BCD cifru, a desna 4 bita (manje značajna četvorka) predstavljaju drugu.

Primer:
- Razmotrimo decimalni broj 29.
- U BCD 8421 kodu, 2 je predstavljeno kao 0010, a 9 kao 1001.
- U pakovanom BCD zapisu, oni se kombinuju u jedan bajt kao: 00101001.

Primer 2:
- 5799 se zapisuje kao $01\ 57\ 99$, tj. $00000001\ 01010111\ 10011001$

Označeni brojevi se zapisuju tako što se cifra za znak zapisuje u poslednji polubajt. C za pozitivne brojeve, a D za negativne.

Primer 3:
- $+1579$ se zapisuje kao $01\ 57\ 9C$
- -15799 se zapisuje kao $15\ 79\ 9D$

Nepakovani BCD zapis

U formatu nepakovanog BCD zapisa, svaka BCD cifra zauzima ceo bajt. Najmanje značajna četvorka (desna 4 bita) bajta koristi se za BCD znamenke, dok je najznačajnija četvorka može biti postavljena na 0000.

Primer:
- 29
- 2 je 00000010, a 9 zapisujemo kao 00001001.
- U dva odvojena bajta bi pisalo 0000001000001001

Nepakovani zapis može biti zapisan u ASCII ili EBCDIC kodu.

EBCDIC kod:
- u levi polubajt se smešta cifra F (oznaka da je u pitanju cifra), a u desni sama cifra broja.

ASCII koda:
- u levi polubajt se smešta cifra 3 (oznaka da je u pitanju cifra), a u desni cifra broja.

Kod označenih brojeva se na mesto poslednje cifre stavlja kod za znak broja (C ili D), umesto znaka F ili 3.

Primer:

- neoznačen dekadni broj 1579
- u EBCDIC kodu zapisuje kao $F1\ F5\ F7\ F9$
- u ASCII kodu kao $31\ 35\ 37\ 39$
- $+1579$ u EBCDIC kodu zapisuje kao $F1\ F5\ F7\ C9$
- -1579 u ASCII kodu kao $31\ 35\ 37\ D9$

Upotreba i poređenje:

- **Efikasnost memorije:**
  - **Pakovani BCD zapis:** Ovaj format je efikasniji u pogledu prostora.
  - **Nepakovani BCD zapis:** Zahteva dvostruko više memorije za isti broj.
- **Obrada:**
  - **Pakovani BCD zapis:** Operacije koje uključuju pakovani BCD zapis imaju dodatne korake za razdvajanje cifara pre obrade, a zatim ih ponovo pakovanje. To može dodati složenost aritmetičkim operacijama.
  - **Nepakovani BCD zapis:** Čini aritmetičke i manipulativne operacije jednostavnijim. Svaka BCD cifra izolovana je u svom bajtu. To može biti posebno korisno u sistemima gde je brzina obrade prioritet.
- **Primene:**
  - **Pakovani BCD zapis:** Koristi se u mnogim mikrokontrolerima, protokolima prenosa podataka i aplikacijama gde je korisna kompresija podataka.
  - **Nepakovani BCD zapis:** Uu nekim starijim računarskim sistemima i određenim specijalizovanim hardverskim okruženjima gde je bitna jednostavnost manipulacije ciframa.

Decimalna aritmetika

Nepakovani zapis brojeva se najcešce koristi pri ulazno-izlaznim operacijama za direktno unošenje brojeva, dok se u aritmetičkim operacijama koristi pakovani zapis. Operacije sa BCD brojevima su sporije u odnosu na operacije sa binarnim brojevima zbog specifičnosti zapisa.

Promena znaka

Pošto su BCD brojevi u računaru zapisani u obliku znak i aspolutna vrednost promena znaka je jednostavna. Samo se menja vrednost poslednjeg polubajta.

$$08\ 96\ 31\ C2 \rightarrow 08\ 96\ 31\ D2$$

Sabiranje i oduzimanje

Neka su A i B dekadni brojevi sa n cifara $A = a_{n-1}a_{n-2}\ldots a_1a_0$ i $B = b_{n-1}b_{n-2}\ldots b_1b_0$, i neka je α funkcija kodiranja koja svakoj cifri u broju pridružuje binarnu kodnu reč.
$A_\alpha = \alpha(a_{n-1})\alpha(a_{n-2})\ldots\alpha(a_1)\alpha(a_0)$
$B_\alpha = \alpha(b_{n-1})\alpha(b_{n-2})\ldots\alpha(b_1)\alpha(b_0)$
Sabiranje se realizuje u dve faze:

1. Odredi se međurezultat $C'_\alpha = A_\alpha + B_\alpha$:
   $$\begin{array}{ccccccc} A_\alpha & = & \alpha(a_{n-1}) & \alpha(a_{n-2}) & \ldots & \alpha(a_1) & \alpha(a_0) \\ B_\alpha & = & \alpha(b_{n-1}) & \alpha(b_{n-2}) & \ldots & \alpha(b_1) & \alpha(b_0) \\ \hline C'_\alpha & = & \alpha(c'_{n-1}) & \alpha(c'_{n-2}) & \ldots & \alpha(c'_1) & \alpha(c'_0) \end{array}$$

2. Dobijeni međurezultat C'_α se koriguje zbog specifičnosti zapisa binarno kodiranih dekadnih brojeva. Konačan rezultat je jednak zbiru međurezultata i korekcije: $C_\alpha = C'_\alpha + K_\alpha$:
   $$\begin{array}{ccccccc} C'_\alpha & = & \alpha(c'_{n-1}) & \alpha(c'_{n-2}) & \ldots & \alpha(c'_1) & \alpha(c'_0) \\ K_\alpha & = & \alpha(k_{n-1}) & \alpha(k_{n-2}) & \ldots & \alpha(k_1) & \alpha(k_0) \\ \hline C_\alpha & = & \alpha(c_{n-1}) & \alpha(c_{n-2}) & \ldots & \alpha(c_1) & \alpha(c_0) \end{array}$$

Oduzimanje binarno kodiranih dekadnih brojeva može da se realizuje na dva načina:

1. Po sličnom principu kao i sabiranje, pri čemu se u obe faze umesto sabiranja vrši oduzimanje brojeva.

2. Kao sabiranje brojeva u potpunom komplementu.

Sabiranje i oduzimanje u kodu 8421

Funkcija kodiranja je definisana kao prevođenje cifre u binarni sistem, tj. $\alpha(c) \to c_3c_2c_1c_0$ gde $\forall c \in \{0,1,2,3,4,5,6,7,8,9\}$ važi $c = c_3 \cdot 2^3 + c_2 \cdot 2^2 + c_1 \cdot 2^1 + c_0 \cdot 2^0$ gde je $c_i \in \{0,1\}$ za $i \in [0,3]$.
Prva faza je nezavisna od funkcije kodiranja tako da se primenjuje prethodni algoritam.
U prikazu druge faze se uvode sledeće oznake:

- $\alpha(c_i')$ označava zbir dobijen sabiranjem kodova za dekadno cifarsko mesto i
- p_i' označava binarni prenos izmedju zbirova $\alpha(c_i')$ i $\alpha(c_{i+1}')$ u međurezultatu prve faze sabiranja
- $\alpha(k_i)$ označava korekciju na dekadnom cifarskom mestu i.
- p_i'' označava binarni prenos u drugoj fazi sabiranja sa dekadnog cifarskog mesta $i-1$ na dekadno cifarsko mesto i. Važi $p_0'' = 0$.

Druga faza se izvodi u n koraka (n maksimum broja cifara dekadnih brojeva koji se sabiraju). Sabiranje se vrši zdesna u levo; postupak u i-tom koraku je sledeći:

1. Određuje se privremeni zbir $t_i = \alpha(c_i') + p_i''$.
2. Na osnovu vrednosti t_i i p_i' određuje se korekcija $\alpha(k_i)$.
3. Krajnja vrednost $\alpha(c_i)$ se dobija kao zbir $t_i + \alpha(k_i)$. Pri tome se određuje i p_{i+1}''.

Korekcija međurezultata je:

1. $p_{i+1}' = 1 \Rightarrow \alpha(k_i) = (0110)_2$
2. $t_i \geq (1010)_2 \Rightarrow \alpha(k_i) = (0110)_2$.
3. $\alpha(k_i) = (0000)_2$.

Prekoračenje se javlja kada je $p_n' = 1$ ili $p_n'' = 1$.

Primeri :

1. A = 18345, B = 9567 A+B=?

A_α	=		0001	1000	0011	0100	0101
B_α	=		0000	1001	0101	0110	0111
P′	=	0	1	0	0	0	0
C'_α	=		0010	0001	1000	1010	1100
P″	=	0	0	0	1	1	0
T	=		0010	0001	1001	1011	1100
K_α	=		0000	0110	0000	0110	0110
C_α	=		0010	0111	1001	0001	0010

Rezultat je 27912.

2. A = -23492, B = -5189, A+B=?

A_α	=		0010	0011	0100	1001	0010
B_α	=		0000	0101	0001	1000	1001
P′	=	0	0	0	1	0	0
C'_α	=		0010	1000	0110	0001	1011
P″	=	0	0	0	0	1	0
T	=		0010	1000	0110	0010	1011
K_α	=		0000	0000	0000	0110	0110
C_α	=		0010	1000	0110	1000	0001

Rezultat je -28681.

3. A = 39492, B = 5782, A+B=?

A_α	=		0011	1001	0100	1001	0010
B_α	=		0000	0101	0111	1000	0010
P′	=	0	0	0	1	0	0
C'_α	=		0011	1110	1100	0001	0100
P″	=	0	1	1	0	0	0
T	=		0100	1111	1100	0001	0100
K_α	=		0000	0110	0110	0110	0000
C_α	=		0100	0101	0010	0111	0100

Rezultat je 45274.

4. A = 3276, B = 4915, A+B=?

A_α	=		0011	0010	0111	0110
B_α	=		0100	1001	0001	0101
P′	=	0	0	0	0	0
C'_α	=		0111	1011	1000	1011
P″	=	0	1	0	1	0
T	=		1000	1011	1001	1011
K_α	=		0000	0110	0000	0110
C_α	=		1000	0001	1001	0001

Rezultat je 8191.

5. A = 1275, B = 452, A-B=?

A_α	=		0000	0001	0010	0111	0101
$-B_\alpha$	=		1001	1001	0101	0100	1000
P'	=	0	0	0	0	0	0
C'_α	=		1001	1010	0111	1011	1101
P"	=	1	1	0	1	1	0
T	=		1010	1010	1000	1100	1101
K_α	=		0110	0110	0000	0110	0110
C_α	=		0000	0000	1000	0010	0011

Rezultat je 823.

6. A = 4231, B = 5348, A-B=?
   $B > A$, pa A-B = -(B-A)

B_α	=		0000	0101	0011	0100	1000
$-A_\alpha$	=		1001	0101	0111	0110	1001
P'	=	0	0	0	0	1	0
C'_α	=		1001	1010	1010	1011	0001
P"	=	1	1	1	1	0	0
T	=		1010	1011	1011	1011	0001
K_α	=		0110	0110	0110	0110	0110
C_α	=		0000	0001	0001	0001	0111

Rezultat je -1117.

7. A = -4257, B = 2369, A+B=?
   $B < |A|$, pa $-|A| + B = -(|A| - B)$

$-A_\alpha$	=		0000	0100	0010	0101	0111
B_α	=		1001	0111	0110	0011	0001
P'	=	0	0	0	0	0	0
C'_α	=		1001	1011	1000	1000	1000
P"	=	1	1	0	0	0	0
T	=		1010	1011	1000	1000	1000
K_α	=		0110	0110	0000	0000	0000
C_α	=		0000	0001	1000	1000	1000

Rezultat je -1888.

8. A = -4254, B = 3365, A+B=?
   $B < |A|$, pa $-|A| + B = -(|A| - B)$

$-A_\alpha$	=		0000	0100	0010	0101	0100
B_α	=		1001	0110	0110	0011	0101
P'	=	0	0	0	0	0	0
C'_α	=		1001	1010	1000	1000	1001
P"	=	1	1	0	0	0	0
T	=		1010	1010	1000	1000	1001
K_α	=		0110	0110	0000	0000	0000
C_α	=		0000	0000	1000	1000	1001

Rezultat je -889.

9. A = -4178, B = 2451, A+B=?
   $B < |A|$, pa $-|A| + B = -(|A| - B)$

A_α	=		0000	0100	0001	0111	1000
$-B_\alpha$	=		1001	0111	0101	0100	1001
P'	=	0	0	0	0	1	0
C'_α	=		1001	1011	0110	1100	0001
P"	=	1	1	0	1	0	0
T	=		1010	1011	0111	1100	0001
K_α	=		0110	0110	0000	0110	0110
C_α	=		0000	0001	0111	0010	0111

Rezultat je -1727.

Prednosti i nedostaci BCD koda

Prednosti:

- Preciznost u finansijskim proračunima.
- Lakoća konverzije u zapis koji je čitljiv ljudima.
- Kompatibilnost sa decimalno baziranim sistemima.

Nedostaci:

- Neefikasno skladištenje u poređenju sa binarnim sistemom.
- Složenije aritmetičke operacije.