Brojni sistemi, binarna aritmetika i aritmetička kola

U svakodnevnom životu, za izražavanje brojnih vrednosti u međuljudskoj komunikaciji uobičajena je praksa korišćenja isključivo decimalnog sistema. Sa druge strane, budući da digitalna kola zasnivaju svoj rad na binarnim signalima koji imaju svega dva stanja, u svetu digitalne elektronike se sama po sebi nameće potreba za korišćenjem sistema sa osnovom 2 (binarnog sistema). Ovde će biti iznesen pregled i osobine najčešće korišćenih brojnih sistema, među koje pored decimalnog i binarnog spada i heksadecimalni. Nakon toga, biće reči o osnovnim aritmetičkim operacijama u binarnom sistemu i o konstrukciji digitalnih kola koja se koriste za obavljanje tih operacija.

Brojni sistemi

Pozicioni brojni sistem sa osnovom b ima skup od b cifara $\{0, 1, ..., b-1\}$. Ceo broj u pozicionom brojnom sistemu sa osnovom b predstavlja se nizom od n cifara:

$$c_{n-1}c_{n-2}...c_1c_0, \text{ pri čemu } c_i \in \{0, 1, ..., b-1\}, i = 0, 1, ..., n-1$$

Vrednost broja dobija se množenjem svake cifre sa težinskim faktorom koji odgovara mestu (poziciji)[1] na kome se cifra nalazi u zapisu broja i potom sumiranjem dobijenih proizvoda. Težinski faktor koji odgovara poziciji i u brojnom sistemu sa osnovom b je b^i, pa je vrednost broja:

$$X = \sum_{i=0}^{n-1} c_i b^i = c_{n-1}b^{n-1} + c_{n-2}b^{n-2} + ... + c_1b^1 + c_0b^0$$

U okviru ovog predmeta, biće korišćena sledeća 3 brojna sistema:

- Decimalni: $b = 10$, sa ciframa $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- Binarni: $b = 2$, sa ciframa $\{0, 1\}$
- Heksadecimalni: $b = 16$, sa ciframa $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$

[1] Krajnja desna cifra ima poziciju 0, a pozicije rastu sa desna na levo.

Budući da se neke od cifara pojavljuju u više brojnih sistema, to znači da jedan isti niz cifara koji predstavlja pozicioni zapis broja može imati različite vrednosti, u zavisnosti od toga u kom sistemu je predstavljen. Da bi se izbegla konfuzija i mogućnost različitih interpretacija istog zapisa, obično se u takvim situacijama osnova brojnog sistema navodi u indeksu nakon cifarskog zapisa broja[2].

Primer: Vrednost broja 101 u različitim brojnim sistemima.

$$101_2 = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 4 + 0 + 1 = 5$$
$$101_{10} = 1 \cdot 10^2 + 0 \cdot 10^1 + 1 \cdot 10^0 = 100 + 0 + 1 = 101$$
$$101_{16} = 1 \cdot 16^2 + 0 \cdot 16^1 + 1 \cdot 16^0 = 256 + 0 + 1 = 257$$

U svim pozicionim brojnim sistemima, cifra koja ima poziciju 0 (krajnja cifra s desne strane) se naziva još i *najmanje značajnom cifrom*, budući da daje najmanji doprinos ukupnoj vrednosti broja. U skladu sa tim, krajnja cifra sa leve strane se naziva *najznačajnijom cifrom* pošto ona najviše doprinosi ukupnoj vrednosti.

Pošto je u digitalnoj elektronici i programiranju binarni sistem od posebne važnosti, razvijena je posebna terminologija koja se odnosi na njega. U nastavku su navedena objašnjenja nekoliko najčešće korišćenih termina:

- **Bit** je cifra binarnog brojnog sistema. Može imati vrednost 0 ili 1 i ujedno predstavlja najmanju moguću količinu digitalne informacije.
- **MSB** (engl. Most Significant Bit) je najznačajniji bit u binarnom zapisu. Nalazi se na krajnjoj poziciji sa leve strane, tj. na poziciji $n-1$, ako binarni zapis broja ima n bita.
- **LSB** (engl. Least Significant Bit) je najmanje značajan bit u binarnom zapisu. Nalazi se na krajnjoj poziciji sa desne strane, tj. na poziciji 0.
- **Nibl** (engl. Nibble) je grupa od 4 bita.
- **Bajt** (engl. Byte) je grupa od 8 bita.

Konverzije između brojnih sistema

U ovom delu će biti prikazani primeri prevođenja brojeva iz jednog u drugi brojni sistem. U sledećoj tabeli prikazani su zapisi prvih 16 brojeva u sva 3 brojna sistema o kojima je do sada bilo reči. Pri tome, binarnim brojevima kojima je zapis kraći od 4 bita dodate su vodeće nule, kako bi bili svedeni na 4-bitnu formu. Ovo je učinjeno kako bi bila olakšana konverzija iz binarnog u heksadecimalni sistem (i obratno)[3].

[2] Pošto je decimalni sistem najviše odomaćen u svakodnevnoj upotrebi, ako osnova sistema nije eksplicitno naglašena, podrazumevaće se da se radi o decimalnom zapisu.

[3] Iz ljudske perspektive, zapis binarnih brojeva sa većim brojem bita postaje zamoran i teško čitljiv. Heksadecimalni sistem upravo i jeste uveden kao kompaktniji način za zapisivanje brojeva, koji se lako prevodi u binarnu formu, kao što će biti prikazano u nastavku.

Decimalni	Binarni	Heksadecimalni
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Heksadecimalni $<->$ Binarni

Konverzija iz heksadecimalnog sistema u binarni je jednostavna zahvaljujući činjenici da je $16 = 2^4$. To znači da svakoj heksadecimalnoj cifri odgovara grupa od tačno 4 bita, a konverzija se vrši tako što se cifre zamene odgovarajućim grupama bita, u skladu sa gore prikazanom tabelom.

Primer: (HEX$-$ >BIN)

$$A5C3_{16} = 1010\ 0101\ 1100\ 0011_2 = 1010010111000011_2$$

Prilikom konverzije iz binarnog u heksadecimalni sistem, svaka grupa od 4 uzastopna bita se zamenjuje odgovarajućom heksadecimalnom cifrom. Ukoliko ukupan broj bita u zapisu binarnog broja nije deljiv sa 4, zapis broja se dopunjava (sa leve strane) sa onoliko nula koliko je potrebno da broj bita postane deljiv sa 4.

Primer: (BIN$-$ >HEX)

$$1110001100010101011001111111_2 = 0111\ 0001\ 1000\ 1010\ 1011\ 0011\ 1111_2 = 718AB3F_{16}$$

Binarni ili Heksadecimalni $->$ Decimalni

U slučaju konverzije iz binarnog ili heksadecimalnog sistema u decimalni, koristi se formula prikazana na početku, po kojoj se svaka cifra množi sa težinskim faktorom, a zatim se dobijeni proizvodi sumiraju.

Primer: (BIN$-$ >DEC)

$$1100111_2 = 1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 64 + 32 + 4 + 2 + 1 = 103_{10}$$

Primer: (HEX$-$ $>$DEC)

$$A39F_{16} = A \cdot 16^3 + 3 \cdot 16^2 + 9 \cdot 16^1 + F \cdot 16^0 = 10 \cdot 4096 + 3 \cdot 256 + 9 \cdot 16 + 15 \cdot 1 = 41887_{10}$$

Decimalni $->$ Binarni ili Heksadecimalni

Kada su u pitanju konverzije iz decimalnog u druge pozicione sisteme, koristi se princip sukscesivnog celobrojnog deljenja decimalnog broja sa brojnom osnovom sistema u koji se vrši konverzija, uz pamćenje ostataka pri svakom pojedinačnom deljenju. Postupak se prekida nakon koraka u kojem je količnik pri deljenju jednak nuli. Na kraju se formira konvertovani broj, ispisivanjem ostataka u obrnutom redosledu od onog kojim su izračunavani.

Primer: (DEC$-$ $>$BIN)

$53 : 2 = 26$,	ostatak 1
$26 : 2 = 13$,	ostatak 0
$13 : 2 = 6$,	ostatak 1
$6 : 2 = 3$,	ostatak 0
$3 : 2 = 1$,	ostatak 1
$1 : 2 = 0$,	ostatak 1

Broj se formira zapisivanjem ostataka odozdo prema gore, pa je tako:

$$53_{10} = 110101_2.$$

Primer: (DEC$-$ $>$HEX)

$42895 : 16 = 2680$,	ostatak 15
$2680 : 16 = 167$,	ostatak 8
$167 : 16 = 10$,	ostatak 7
$10 : 16 = 0$,	ostatak 10

Pri formiranju heksadecimalnog broja, ostaci se konvertuju u odgovarajuće heksadecimalne cifre, u skladu sa tabelom konverzije. U ovom primeru dobijen je broj:

$$42895_{10} = A78F_{16}$$

Binarna aritmetika

U ovom delu će kroz primere biti predstavljene osnovne aritmetičke operacije sabiranja i oduzimanja celih brojeva u binarnom sistemu. Takođe, biće reči o uobičajenom načinu za predstavljanje negativnih brojeva.

Sabiranje

Sabiranje celih brojeva u binarnom sistemu vrši se na sličan način kao u decimalnom: sukcesivno se sabiraju parovi bita, počevši od para bita najmanje važnosti. Prilikom sabiranja dva bita, javljaju se sledeće 4 mogućnosti:

$0 + 0 = 0$, prenos 0
$0 + 1 = 1$, prenos 0
$1 + 0 = 1$, prenos 0
$1 + 1 = 0$, prenos 1

Dakle, u poslednjem slučaju dolazi do situacije da rezultat sabiranja ne može biti predstavljen jednim bitom. U tom slučaju, (pošto je u binarnom sistemu $1 + 1 = 10$), kaže se da je rezultat sabiranja 0, ali postoji izlazni prenos 1, koji mora biti uračunat pri sabiranju sledećeg para bita. Generalna forma za binarno sabiranje para bita na poziciji n, tj. $A_n + B_n$ uključuje ulazni prenos C_{in} nastao pri sabiranju prethodnog para bita. Na izlazu se dobija rezultat S_n i izlazni prenos C_{out}. Sledeća tabela prikazuje zbir S_n i izlazni prenos C_{out} za sve moguće kombinacije ulaza A_n, B_n i ulaznog prenosa C_{in}. Sa desne strane uporedo je prikazano nekoliko primera sabiranja u decimalnom i binarnom sistemu.

A_n	B_n	C_{in}	S_n	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

DECIMALNO	BINARNO
5	0000 0101
+2	+0000 0010
7	0000 0111
18	0001 0010
+ 2	+0000 0010
20	0001 0100
147	1001 0011
+ 75	+0100 1011
222	1101 1110

Oduzimanje

Prilikom binarnog oduzimanja dva bita javljaju se sledeće 4 mogućnosti:

$0 - 0 = 0$, pozajmica $= 0$
$0 - 1 = 1$, pozajmica $= 1$
$1 - 0 = 1$, pozajmica $= 0$
$1 - 1 = 0$, pozajmica $= 0$

Slično kao kod sabiranja, pri oduzimanju proizvoljnog para bita na poziciji n (tj. $A_n - B_n$), potrebno je voditi računa o ulaznoj pozajmici B_{in}. Na izlazu se dobijaju razlika R_n i izlazna pozajmica B_{out}. Svi mogući slučajevi su prikazani u tabeli (levo), a sa desne strane je dat uporedni prikaz nekoliko primera oduzimanja u decimalnom i binarnom sistemu.

A_n	B_n	B_{in}	R_n	B_{out}
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

DECIMALNO	BINARNO
27	0001 1011
-10	- 0000 1010
17	0001 0001
172	1010 1100
- 42	- 0010 1010
130	1000 0010
192	1100 0000
- 3	- 0000 0011
189	1011 1101

Predstavljanje negativnih brojeva

Za prikaz negativnih brojeva u binarnom brojnom sistemu najčešće se koristi predstava broja u *komplementu dvojke*. Ovaj način predstavljanja brojeva biće prikazan na primeru binarnih brojeva u osmobitnoj označenoj formi[4]. Ovakva forma dozvoljava predstavljanje i pozitivnih i negativnih vrednosti, koje se nalaze u intervalu od -128 do 127[5].

Prevođenje broja u njegov drugi komplement (komplement dvojke), koji predstavlja broj koji je isti po apsolutnoj vrednosti, a suprotan po predznaku, vrši se u dva koraka:

1. Invertuje se vrednost svih n bita u binarnom zapisu broja.
2. Dobijeni rezultat se sabere sa 1.

Ovakav postupak konverzije je nezavistan od toga da li se pozitivna vrednost prevodi u negativnu, ili obrnuto. Iako možda na prvi pogled deluje čudno, ovde se krije poželjno svojstvo komplementa dvojke kao reprezentacije suprotne vrednosti: komplement dvojke nekog broja je tačno ona vrednost sa kojom treba sabrati taj broj da bi se dobila nula. Ova osobina upravo dovodi do toga da se pri sabiranju i oduzimanju uvek dobijaju korektne vrednosti, bez obzira na to da li u operacijama učestvuju pozitivne ili negativne vrednosti, dokle god rezultat ne izlazi iz zadatog opsega.

Primer: (Komplement dvojke broja 35)
$35_{10} = 00100011_2$
1.korak (invertovanje): 11011100_2
2.korak (+1): 11011101_2

Dakle, $-35_{10} = 11011101_2$

Primer: (Komplement dvojke broja -35)
$-35_{10} = 11011101_2$
1.korak (invertovanje): 00100010_2
2.korak (+1): 00100011_2

Dakle, $-(-35_{10}) = 00100011_2$

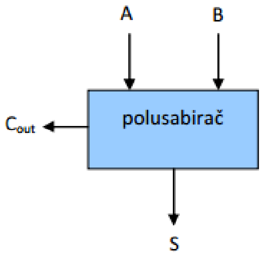
[4] U najvećem broju programskih jezika, za svaki celobrojni tip podataka unapred se definišu 2 svojstva: širina (tj. broj bita koji u memoriji zauzima varijabla datog tipa) i način tretiranja vrendosti broja, koji može biti označen ili neoznačen.

[5] Generalno, označeni tip širine n bita uzima vrednosti iz opsega $-2^{n-1}..2^{n-1}-1$

Aritmetička kola

Polusabirač

Polisabirač je kolo koje obavlja operaciju sabiranja dva bita, bez vođenja računa o ulaznom prenosu. Kao takav, pogodan je za korišćenje u situacijama u kojim je unapred poznato da je ulazni prenos jednak nuli, kao što je recimo pri sabiranju najmanje značajnih bita. U nastaku prikazan je simbol polusabirača (levo), funkcionalna tabela (u sredini) i logičke funkcije kojima se vrši njegova realizacija (desno).



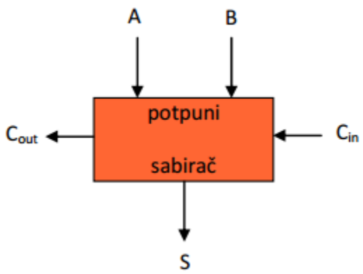
A	B	S	C_{out}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$S = \overline{A} \cdot B + A \cdot \overline{B} = A \bigoplus B$$
$$C_{out} = A \cdot B$$

<u>Zadatak</u> U simulatoru nacrtati logičku šemu polusabirača i potvrditi funkcionalnost simulacijom.

Sabirač

Za razliku od polusabirača, potpuni sabirač uzima u obzir i ulazni prenos (C_{in}).U nastaku prikazan je simbol potpunog sabirača (levo), funkcionalna tabela (u sredini) i logičke funkcije kojima se vrši njegova realizacija (desno).



A_n	B_n	C_{in}	S_n	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$A_n \backslash B_nC_{in}$	00	01	11	10
0	0	1	0	1
1	1	0	1	0

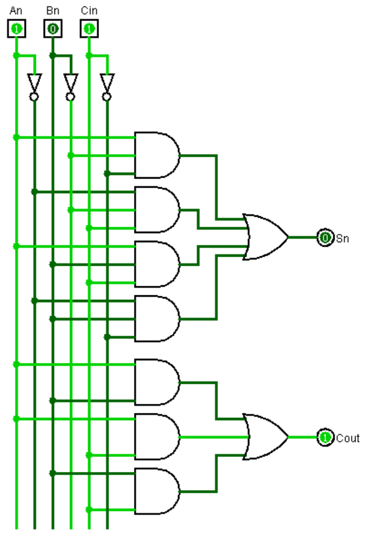
$$S_n = A_n \cdot \overline{B_n} \cdot \overline{C_{in}} + \overline{A_n} \cdot \overline{B_n} \cdot C_{in} +$$
$$+ A_n \cdot B_n \cdot C_{in} + \overline{A_n} \cdot B_n \cdot \overline{C_{in}}$$

$A_n \backslash B_nC_{in}$	00	01	11	10
0	0	0	1	0
1	0	1	1	1

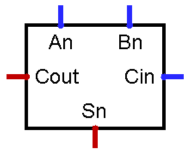
$$C_{out} = A_n \cdot B_n + A_n \cdot C_{in} + B_n \cdot C_{in}$$

Zadatak Nacrtati šemu i simulirati rad jednobitnog potpunog sabirača u programu *Logisim*. Nakon toga, kreirati šematski simbol jednobitnog sabirača koji će biti korišćen pri realizaciji višebitnih sabirača.

Rešenje:



Logička šema

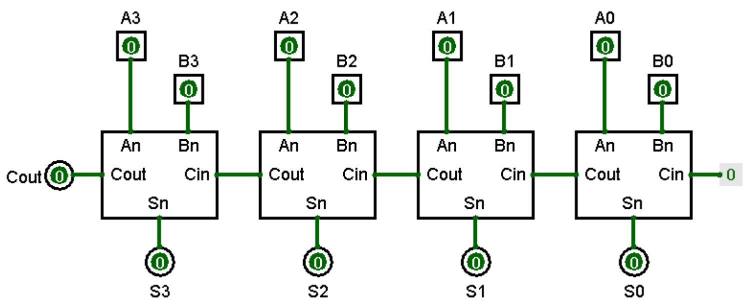


Šematski simbol

Zadatak Koristeći simbol jednobitnog potpunog sabirača kreiran u prethodnom zadatku, realizovati i simulirati rad 4-bitnog sabirača.

Rešenje:

Višebitni sabirači se dobijaju kaskadnim vezivanjem jednobitnih sabirača. Pri tome, izlazni prenos sabirača koji vrši sabiranje para bita na poziciji $i-1$ (tj. $A_{i-1}+B_{i-1}$) vezuje se na ulazni prenos sledećeg sabirača koji vrši sabiranje na poziciji i (tj. A_i+B_i). Kada je u pitanju sabiranje para bita najmanje važnosti (A_0+B_0), koristi se polusabirač, ili pun sabirač kojem se dovodi 0 kao ulazni prenos, kao što je prikazano na slici.



Zadatak Koristeći potreban broj jednobitnih potpunih sabirača, realizovati i simulirati rad kola koje računa $F = A - B$, gde su A, B i F 4-bitni brojevi. Oduzimanje realizovati kao sabiranje sa komplementom dvojke broja B (tj. $F = A + (-B) = A + \overline{B} + 1$).