

VTŠ: Osnovi računarske tehnike

Binarna aritmetika

mr. Veličković Zoran
Februar, 2010.

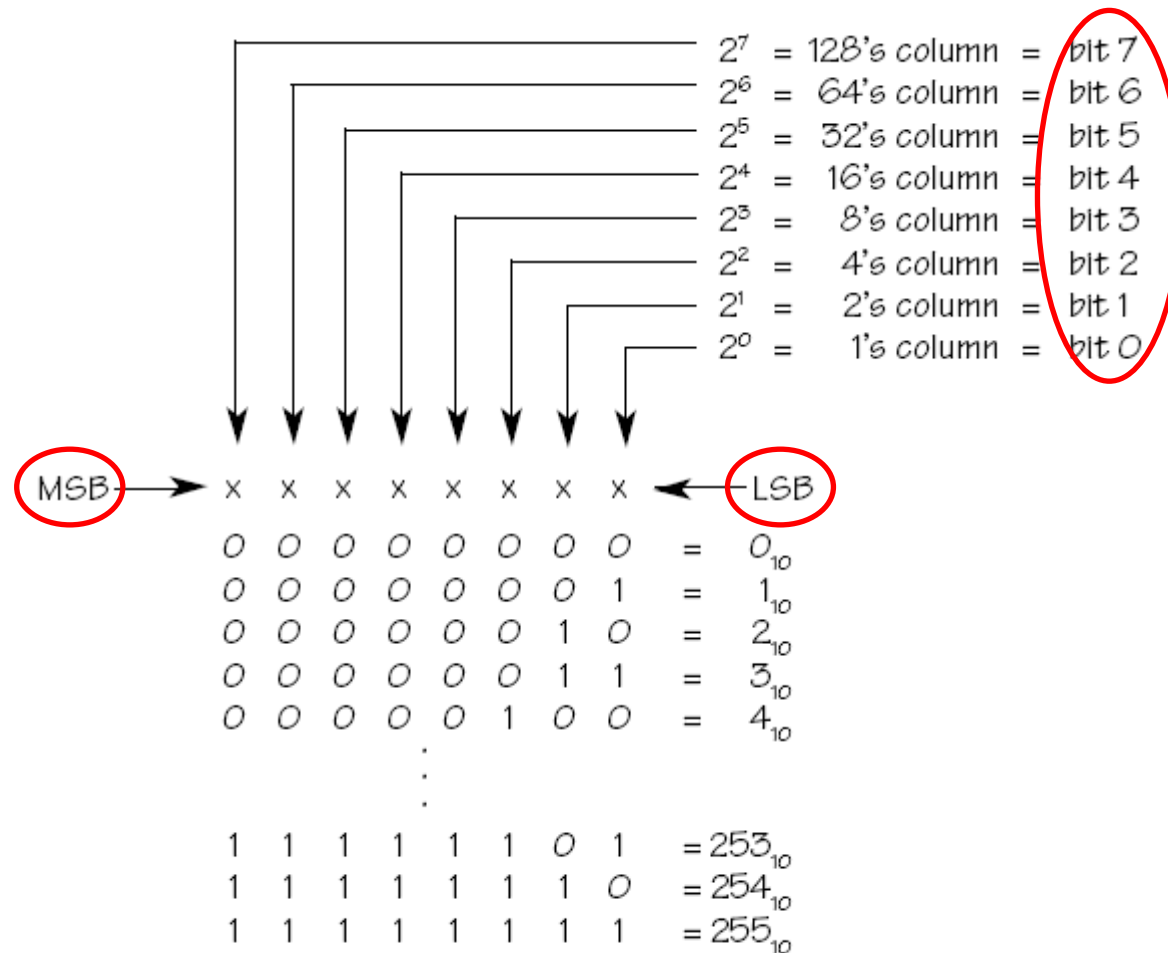
Binarna aritmetika

- ❑ Slično kao i kod aritmetike na papiru, i decimalni i binarni brojevi mogu biti bilo koje veličine, ograničenja su samo u veličini papira i dugotrajnosti olovke!
- ❑ U manipulaciji podataka u računarima se koriste fizičke logičke kapije i veze, tako da je njima određena širina podataka.
- ❑ Tako je maksimalna vrednost broja reprezentovanog u računarima određena brojem bitova koji predstavljaju neki broj.

Neoznačen binarni broj

- ❑ Neoznačen (unsigned) binarni brojevi mogu se koristiti za prikazivanje samo **pozitivnih** vrednosti. (Na sledećem slajdu prikazan je opseg brojeva koji se mogu reprezentovati sa 8 bitova.
- ❑ "x" ("*") karakter označava jedan bit (bilo 0 bilo 1)
- ❑ Na krajnjoj desnoj strani se nalazi bit najmanje težine (*least significant bit -LSB*), indeksira se kao **bit 0**.
- ❑ Na levoj strani se nalazi najteži bit (*most significant bit -MSB*), indeksira se kao **bit 7**.
- ❑ Svaki individualni bit može uzeti vrednost 1 ili 0, tako da grupa od 8 bitova može predstaviti **2⁸** jedinstvenih kombinacija 0 i 1.
- ❑ U decimalnoj notaciji to se predstavlja 0_{10} do $+255_{10}$

Neoznačen binarni broj (2)



Binarno sabiranje (1)

- ❑ Dva binarna broja se sabiraju na sličan način kao i kod decimalne notacije.
- ❑ Prvo se sabiraju dva najmanje značajna LSB bita i ako je potrebno formiraju prenos *za sabiranje u sledećem koraku*.
- ❑ Plostupak se ponavlja sve dok se ne dođe do MSB bita.
- ❑ Pogledajmo primer.

Binarno sabiranje (2)

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

(a) Bit 0, $1 + 0 = 1_2$

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

(b) Bit 1, $0 + 1 = 1_2$

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

(c) Bit 2, $0 + 0 = 0_2$

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

(d) Bit 3, $1 + 1 = 10_2$

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

(e) Bit 4, $1 + 1 + \text{carry_in} = 11_2$

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

(f) Bit 5, $1 + 0 + \text{carry_in} = 10$,

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

(g) Bit 6, $0 + 0 + \text{carry_in} = 1_2$

$$\begin{array}{r} \boxed{\begin{array}{c} 0 \\ 0 \end{array}} \begin{array}{cccccccc} 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{array} \\ + \begin{array}{c} 0 \\ 0 \end{array} \begin{array}{cccccccc} 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{array} \\ \hline = \begin{array}{cccccccc} 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \end{array} \end{array}$$

(h) Bit 7, $0 + 0 = 0$.

$$\begin{array}{r} 57_{10} \\ + 26_{10} \\ \hline = 83_{10} \end{array}$$

Binarno oduzimanje

- ❑ Binarno oduzimanje se može obaviti slično kao kod decimalne notacije.
- ❑ Međutim, kod računara se koristi takozvana komplementarna tehnika.
- ❑ Da bi se ova tehnika realizovala kod decimalne notacije treba formirati *tens* i *nines* complement.
- ❑ Kod binarnog oduzimanja ekvivalenti u *dvojični* i *jedinični* komplement.
- ❑ Pogledajmo sledeće primere.

Nines komplement₁₀

Standard subtraction

$$\begin{array}{r} 647 \\ - 283 \\ \hline = 364 \end{array}$$

Nines complement equivalent

$$\begin{array}{r} 999 \\ - 283 \\ \hline = 716 \end{array}$$

Take nines complement

$$\begin{array}{r} 647 \\ + 716 \\ \hline = 1363 \end{array}$$

Add nines complement to minuend

End-around-carry

Formiranje
"devetičnog"
komplementa

Prenos, i
konačan
zbir

Tens complement₁₀

Standard subtraction

$$\begin{array}{r} 647 \\ - 283 \\ \hline = 364 \end{array}$$

Tens complement equivalent

$$\begin{array}{r} 1000 \\ - 283 \\ \hline = 717 \end{array} \quad + \quad \begin{array}{r} 647 \\ 717 \\ \hline = 1364 \\ \hline 364 \end{array}$$

Take tens complement

Add tens complement to minuend

Drop any carry

Formiranje
"desetičnog"
komplementa

Dodavanje
"desetičnog"
komplementa

Odbacivanje
prenosa

Jedinični komplement₂

Standard subtraction

$$\begin{array}{r} 00111001 \\ - 00011110 \\ \hline = 00011011 \end{array}$$

$$57_{10} - 30_{10} = 27_{10}$$

$$\begin{aligned} 2^4 + 2^3 + 2^2 + 2^1 &= \\ 16 + 8 + 4 + 2 &= \\ = 30 \end{aligned}$$

$$\begin{aligned} 2^5 + 2^4 + 2^3 + 2^0 &= \\ 32 + 16 + 8 + 1 &= \\ = 57 \end{aligned}$$

Ones complement equivalent

$$\begin{array}{r} 11111111 \\ - 00011110 \\ \hline = 11100001 \end{array}$$

Take ones complement

$$\begin{array}{r} 00111001 \\ + 11100001 \\ \hline 100011010 \end{array}$$

End-around-carry

$$\begin{array}{r} 00011011 \\ \hline \end{array}$$

Add ones complement to minuend

Prenos se dodaje u zadnjoj fazi

Formiranje jediničnog komplementa

$$\begin{aligned} 2^4 + 2^3 + 2^1 + 2^0 &= \\ 16 + 8 + 2 + 1 &= \\ = 27 \end{aligned}$$

Dvojični komplement₂

Standardno oduzimanje

$$\begin{array}{r} 00111001 \\ - 00011110 \\ \hline = 00011011 \end{array}$$

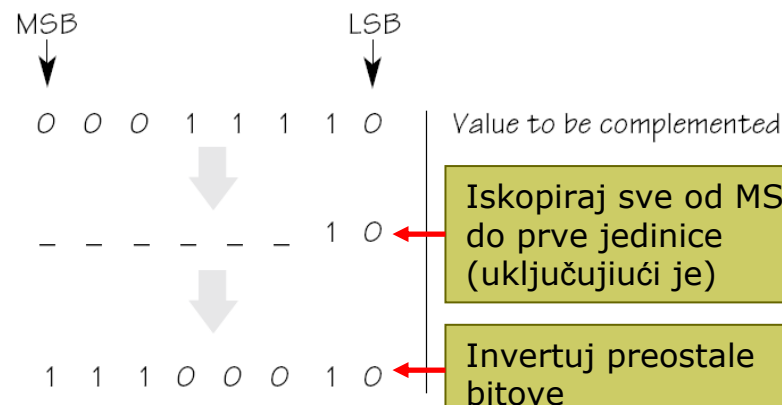


Oduzimanje dvojičnim komplementom

$\begin{array}{r} 10000000 \\ - 00011110 \\ \hline = 11100010 \end{array}$ Take twos complement	$\begin{array}{r} 00111001 \\ + 11100010 \\ \hline 100011011 \\ 00011011 \end{array}$ Add twos complement to minuend
--	---

Drop any carry

Formiranje
dvoičnog
komplementa



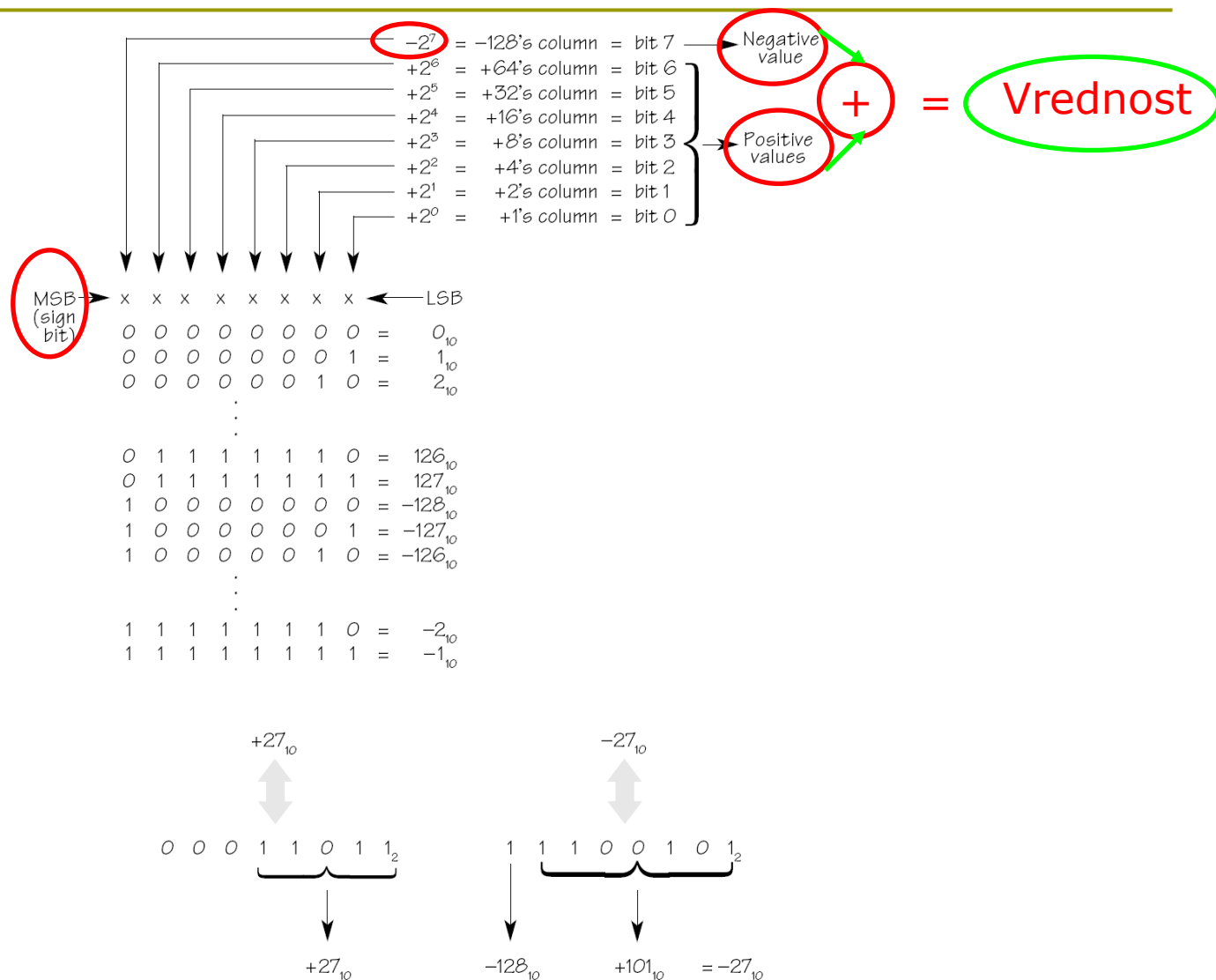
Formiranje dvoičnog komplementa, jedan od načina

Prenos se odbacuje u zadnjoj fazi

Označeni binarni brojevi (1)

- ❑ Označeni brojevi se koriste za predstavljanje **pozitivnih i negativnih** veličina.
- ❑ Kod decimalne notacije se koristi **sign-magnitude** način za predstavu negativnih brojeva. Primer **+27** i **-27**.
- ❑ Iz razloga efikasnosti u računarima se ne koristi ovaj pristup!
- ❑ Naime, u računarima se koristi **sign-bit** način predstave negativnih brojeva. Vrednost MSB bita određuje znak binarnog broja.
- ❑ Pogledajte sledeću tabelu u kojoj je predstavljeno korišćenje bita za označavanje znaka (sign bita)

Označeni binarni brojevi (2)



Oduzimanje i označeni brojevi

Decimalno: signed-magnitude

Binarno: signed-binarno

$$\begin{array}{r} 5 \ 7 \\ + \ 3 \ 0 \\ \hline = \ 8 \ 7 \end{array} \quad \longleftrightarrow \quad \begin{array}{r} 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 1 \\ + \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \\ \hline 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1 \end{array}$$

$$\begin{array}{r} 5 \ 7 \\ + \ -3 \ 0 \\ \hline = \ 2 \ 7 \end{array} \quad \longleftrightarrow \quad \begin{array}{r} 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 1 \\ + \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \\ \hline 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \end{array}$$

$$\begin{array}{r} -5 \ 7 \\ + \ 3 \ 0 \\ \hline = \ -2 \ 7 \end{array} \quad \longleftrightarrow \quad \begin{array}{r} 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \\ + \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \\ \hline 1 \ 1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \end{array}$$

$$\begin{array}{r} -5 \ 7 \\ + \ -3 \ 0 \\ \hline = \ -8 \ 7 \end{array} \quad \longleftrightarrow \quad \begin{array}{r} 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \\ + \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \\ \hline 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \end{array}$$

Negativni brojevi

$$\begin{aligned} & -2^7 + 2^5 + 2^3 + 2^0 = \\ & -128 + 32 + 8 + 1 = \\ & = -87 \end{aligned}$$

Prednosti signed-binary predstave

- ❑ Brojevi mogu biti uvek dodati direktno bez obzira da li oni predstavljaju pozitivne ili negativne brojeve.
- ❑ Aritmetička operacije:
$$a + b, a + (-b), (-a) + b, \text{ i } (-a) + (-b)$$
se obavljaju uvek na **isti način**, jednostavno sabiranjem ovih dveju veličina.
- ❑ Ovo ubrzava rad računara i može se realizovati sa manjim brojem logičkih kapija.
- ❑ Dakle, računari ne zahtevaju različite blokove za sabiranje i oduzimanje, već samo zahtevaju sabirač i dvojični komplementator (koji se realizuje sa par logičkih kola).
- ❑ Realizaciju operacija binarnog oduzimanja i binarnog sabiranja učićemo kasnije.

Binarno množenje

- Jedan od načina realizacije množenja je prevođenje u **višestruko sabiranje**:

$$6 \times 4 = 6 + 6 + 6 + 6 = 24.$$

- Iako je to jednostavno za računare, u slučaju dugih nizova brojeva to može biti vremenski zahtevan proces.
- Još jedan način za realizaciju množenja je tehnika **“pomeri i saberi”** (shift-and-add).
- Pogledajmo sledeći primer ove tehnike:

Tehnika “pomeri i saberi”

