

Številski sistemi

Desetiški (okoli 3000 p.n.š. – Egipt)

Maji – dvajsetiški, Babilonci in Sumerci – šestdesetiški

Desetiški z ničlo – Indija 500 p.n.š, nato Perzija

Prvi računalniki – desetiški sistem, nato dvojiški.

Splošni zapis:

$$S = a_n r^n + a_{n-1} r^{n-1} + \dots a_2 r^2 + a_1 r + a_0 + a_{-1} r^{-1} + a_{-2} r^{-2} + \dots + a_{-m} r^{-m}$$

Primeri:

$$108,59375_{(10)} =$$

$$1101100,10011_{(2)}$$

$$1100101,01001_{(2)}$$

$$=101,28125_{(10)}$$

$$B3C,8_{(16)} = 2876,5_{(10)}$$

$$S = \sum_{i=-m}^n a_i r^i$$

Primer:

$$123,45 = 1 * 10^2 + 2 * 10^1 + 3 * 10^0 + 4 * 10^{-1} + 5 * 10^{-2}$$



Organizacija števil v računalniku

Nepredznačena cela števila (NBC-natural bin.code):

- vrednosti med 0 in 2^n-1 , vseh števil je 2^n
- 4-bitna beseda (nibble): $2^4=16$ različnih kombinacij - možno je predstaviti 16 vrednosti
- 8-bitna beseda - bajt (byte): 256 vrednosti
- 16-bitna beseda (double-word):
65536 vrednosti
- 32-bitna beseda (quad-word):
 $4.29 \cdot 10^9$ vrednosti



Organizacija števil v računalniku

Predznačena cela števila:

- predstavitev z dvojiškim komplementom (analitični kod)
- pogojena z dolžino besede
- pozitivna: najvišji bit vedno 0, negativna vedno 1

– Primer: $-74_{(10)} = B6_{(16)}$ v 8-bitni besedi

- Razširitev predznaka



Primeri aritmetičnih operacij nad dvojiškimi (šestnajstiškimi) števili

Predpostavimo šestnajstbitno besedo:

- Seštej šestnajstiški števili $\$0B5A + \$03B6 = \$F10$
- Odštej $\$B74 - \$3A2 = \$7D2$
- Katero je največje desetiško število, ki ga lahko še predstavimo v besedi?
- Katero je najbolj negativno število?
- Zmnoži $\$BC \times \13

$$\begin{array}{r} 234 \\ \hline =\$DF4 \end{array}$$



Podatki v več besedah

Kadar procesorska arhitektura ne omogoča vzdrževanja in računskih operacij nad dovolj dolgimi besedami, lahko zasedajo podatki po več (običajno sledečih si) besed.

- v prvi besedi najvišji del podatka (z največjo težo, most significant word - MSW, npr. pri Motorolinskih mikroprocesorjih) ali
- njegov najnižji del (least significant word - LSW, kot npr. pri Intelovih mikroprocesorjih),

ostali pa po vrsti sledijo v nadaljnjih besedah.



Aritmetika nad števili v več besedah

Običajno ni direktno mogoča, zato programiranje

Aritmetični ukazi v mikroprocesorjih

Posebni statusni biti kot rezultat aritm. operacij

Prenos, preliv, zero, negativ

1. primer:

$$\begin{array}{r} 3A = 0011\ 1010 \\ 33 = 0011\ 0011 \\ \hline 6D = 0110\ 1101 \end{array}$$

(OK, rezultat ni presegel 7 bitov)

2. primer:

$$\begin{array}{r} 3A = 0011\ 1010 \\ 37 = 0011\ 0111 \\ \hline 71 = 0111\ 0001 \end{array}$$

(prišlo je do prenosa med besedama)

3. primer:

$$\begin{array}{r} 3A = 0011\ 1010 \\ -2C = 0010\ 1100 \\ \hline 0E = 0000\ 1110 \end{array}$$

(prenos med besedama (borrow) pri odštevanju)

4. primer:

$$\begin{array}{r} 3A = 0011\ 1010 \\ 59 = 0101\ 1001 \\ \hline 93 = 1001\ 0011 \end{array}$$

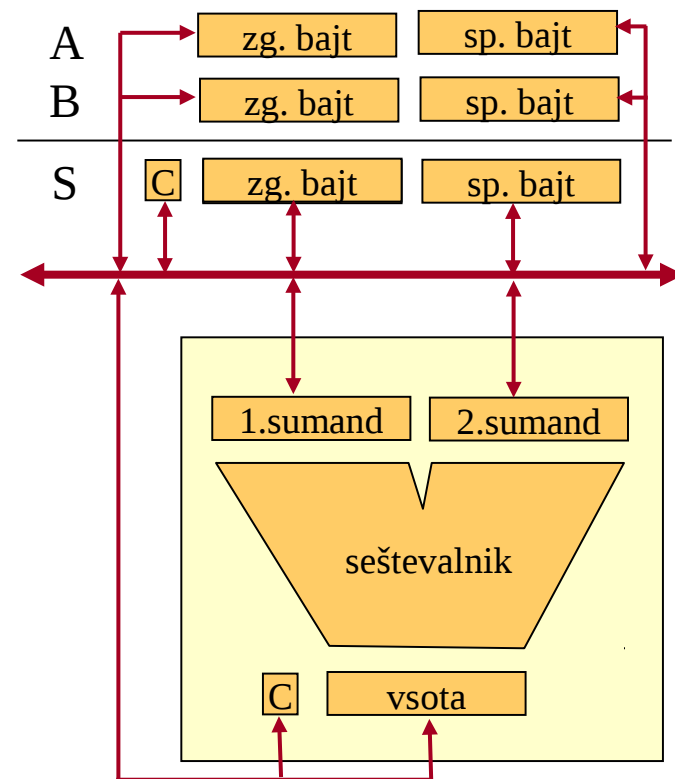
(najvišji bit rezultata je postal 1, pomeni, da bi bil negativen; → preliv.)



Aritmetika nad števili v več besedah

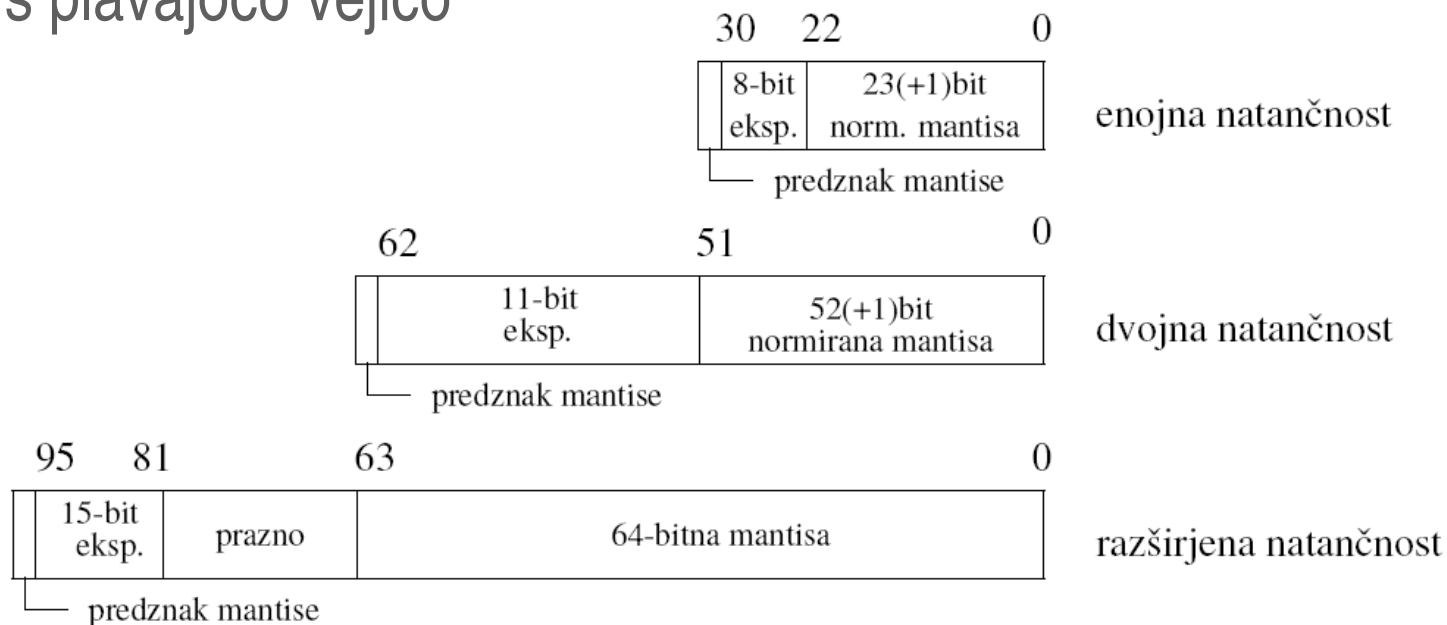
Primer programa v simboličnem strojnem jeziku za seštevanje dveh nepredznačenih podatkov $S=A+B$, predstavljenih v dveh osembitnih besedah:

naloži spodnji bajt A v prvi sumand seštevalnika
naloži spodnji bajt B v drugi sumand seštevalnika
seštej brez prenosa
shrani rezultat v spodnji bajt S
naloži zgornji bajt A v prvi sumand seštevalnika
naloži zgornji bajt B v drugi sumand seštevalnika
seštej s prenosom
shrani rezultat v zgornji bajt S
vpiši prenos v skupni prenos C(S)



Kodiranje števil

Števila s plavajočo vejico



$$x := (-1)^s * 1, f * 2^{e-127}$$



Kodiranje števil

Binarno kodirana desetiška števila (BCD kod)

Algoritem za seštevanje dveh osembitnih nepredznačenih BCD števil:

- šestnajstiško seštej podatka (dva bajta).
- če je spodnji kvartet (nibble) rezultata večji kot 9, prištej 6.
- če je zgornji kvartet (nibble) novega rezultata večji kot 9, celotni vrednosti prištej 60.

$$\begin{array}{r} 47 \\ + 53 \\ \hline 9A \end{array} \implies \begin{array}{r} 9A \\ + 06 \\ \hline A0 \end{array} \implies \begin{array}{r} A0 \\ + 60 \\ \hline 100 \end{array}$$



Kodiranje števil

Alfanumerični ASCII kod

zg. 3 biti→ sp. 4 biti↓	000 (0)	001 (1)	010 (2)	011 (3)	100 (4)	101 (5)	110 (6)	111 (7)
0000 (0)	NUL	DLE	SP	0	@	P	'	p
0001 (1)	SOH	DC1	!	1	A	Q	a	q
0010 (2)	STX	DC2	“	2	B	R	b	r
0011 (3)	ETX	DC3	#	3	C	S	c	s
0100 (4)	EOT	DC4	\$	4	D	T	d	t
0101 (5)	ENQ	NAK	%	5	E	U	e	u
0110 (6)	ACK	SYN	&	6	F	V	f	v
0111 (7)	BEL	ETB	‘	7	G	W	g	w
1000 (8)	BS	CAN	(8	H	X	h	x
1001 (9)	HT	EM)	9	I	Y	i	y
1010 (A)	LF	SUB	*	:	J	Z	j	z
1011 (B)	VT	ESC	+	;	K	[k	{
1100 (C)	FF	FS	,	<	L	\	l	
1101 (D)	CR	GS	-	=	M]	m	}
1110 (E)	SO	RS	.	>	N	^	n	~
1111 (F)	SI	US	/	?	O	_	o	DEL

Drugi kodi: ISO8859-n, MS CP (1250, 852,..)



Kodiranje števil

UNICODE kod

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
U-0x000?																
U-0x001?																
U-0x002?		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
U-0x003?	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
U-0x004?	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
U-0x005?	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
U-0x006?	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
U-0x007?	p	q	r	s	t	u	v	w	x	y	z	{		}	~	

.....

U-0x068?	ا	ب	ت	ث	ج	ح	خ	د	ذ	ر	ز	س	ش	ص	ض	ط
U-0x069?	ظ	ي	ك	ق	ف	ق	ف	ك	ن	هـ	و	ز	ح	ط	ظ	ع
U-0x06A?	غ	ف	ق	ك	ن	هـ	و	ز	ح	ط	ظ	ع	غ	ف	ق	ك
U-0x06B?	ن	هـ	و	ز	ح	ط	ظ	ع	غ	ف	ق	ك	ن	هـ	و	ز
U-0x06C?	ح	ط	ظ	ع	غ	ف	ق	ك	ن	هـ	و	ز	ح	ط	ظ	ع
U-0x06D?	غ	ف	ق	ك	ن	هـ	و	ز	ح	ط	ظ	ع	غ	ف	ق	ك



Predstavitev nekaterih vgrajenih tipov ANSI C v računalniku

char: 8 bitov (0..255, nepredznačeno)

signed char: 8 bitov (-128..127)

short: 16 bitov (-32768..32767) – tudi unsigned

int: 16(/32) bitov – tudi unsigned

long: 32(/64) bitov ($\sim -2 \cdot 10^9 \dots 2 \cdot 10^9$) – tudi unsigned

(double int: 64 bitov)

float: 32 bitov ($\pm 3,4 \cdot 10^{-38} \dots 3,4 \cdot 10^{38}$, natančnost 7 števk)

long: 64 bitov ($\pm 1,7 \cdot 10^{-308} \dots 1,7 \cdot 10^{308}$, natančnost 15 števk)

long double: 80 bitov ($\pm 3,4 \cdot 10^{-4932} \dots 1,1 \cdot 10^{4932}$, natančnost 19 št.)



Predstavitev polj (array) v računalniku

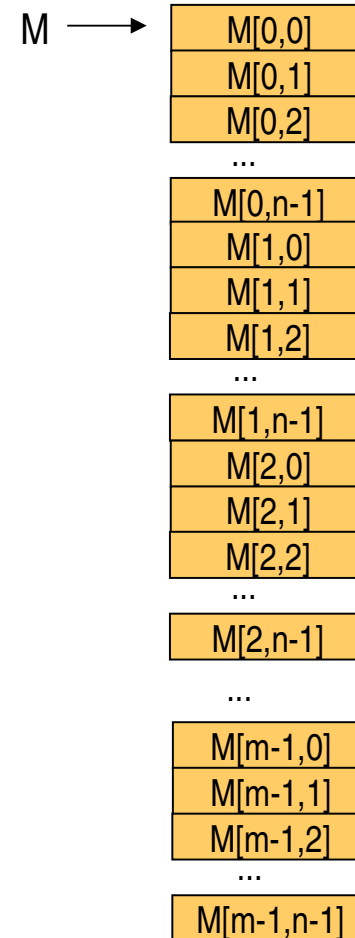
Matrika M

M[0,0]	M[0,1]	M[0,2]	...	M[0,n-1]
M[1,0]	M[1,1]	M[1,2]	...	M[1,n-1]
M[2,0]	M[2,1]	M[2,2]	...	M[2,n-1]
...				
M[m-1,0]	M[m-1,1]	M[m-1,2]	...	M[m-1,n-1]

Izračun fizičnega naslova
elementa M[v,s]:

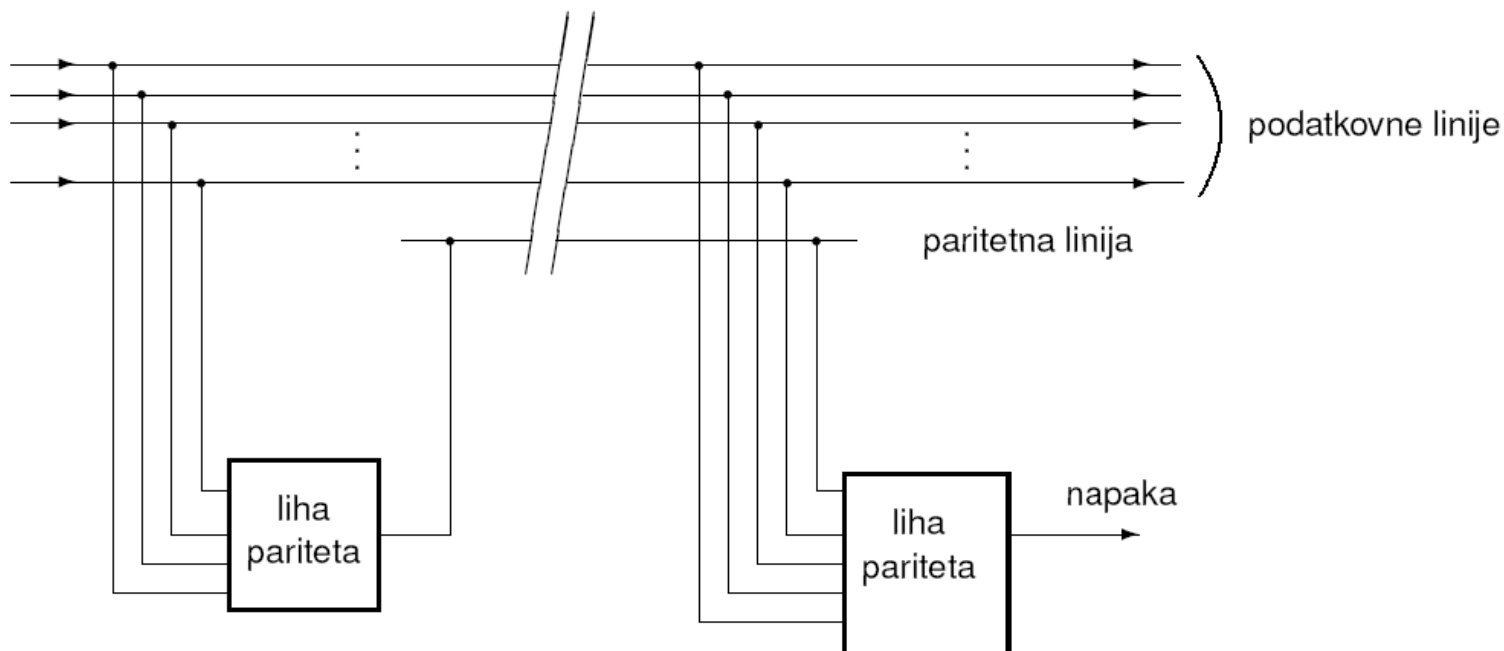
$$\text{addr} = M + (v) * n + s$$

pomnilnik



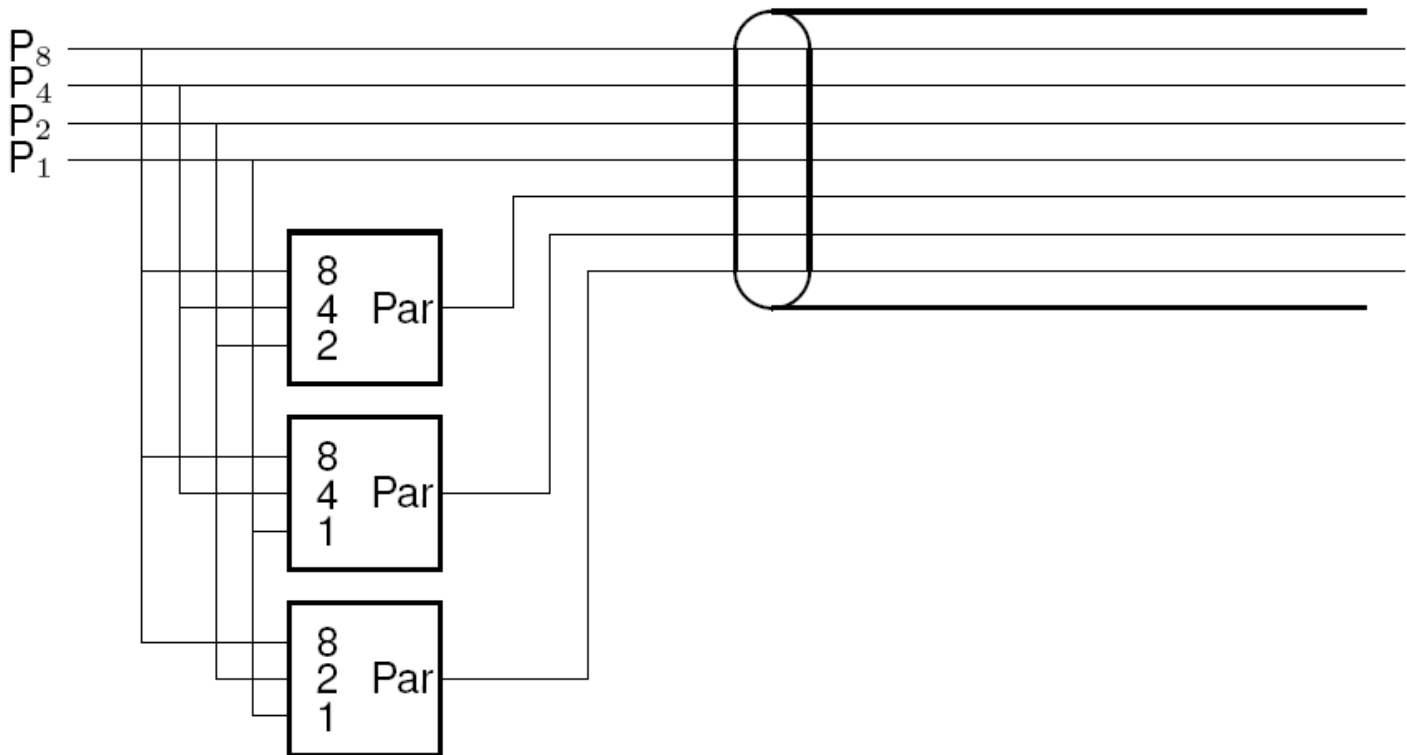
Kodiranje števil

Kod za detekcijo napak pri prenosu



Kodiranje števil

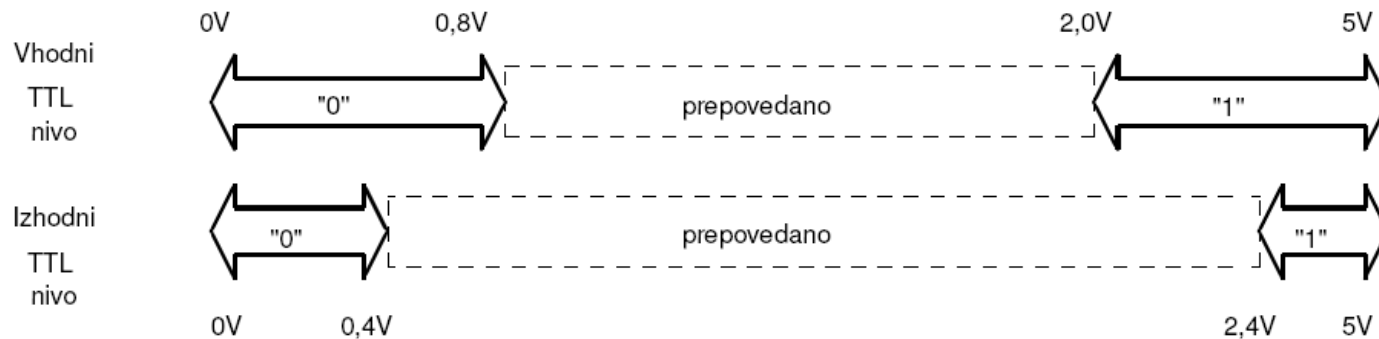
Kod za korekcijo napak pri prenosu



Fizikalna predstavitev dvojiških podatkov v računalniku

Za predstavljanje podatkov v računalniku običajno uporabljamo električno napetost
(včasih tudi katero drugo fizikalno veličino, npr. električni ali svetlobni tok za prenos podatkov)

Primer: Mejni napetostni nivoji v TTL tehnologiji (tradicionalni napetostni nivoji):



(Pozor: merilo za prikaz napetosti je zelo popačeno!)

