OSNOVE RAČUNALNIŠKE ARHITEKTURE I

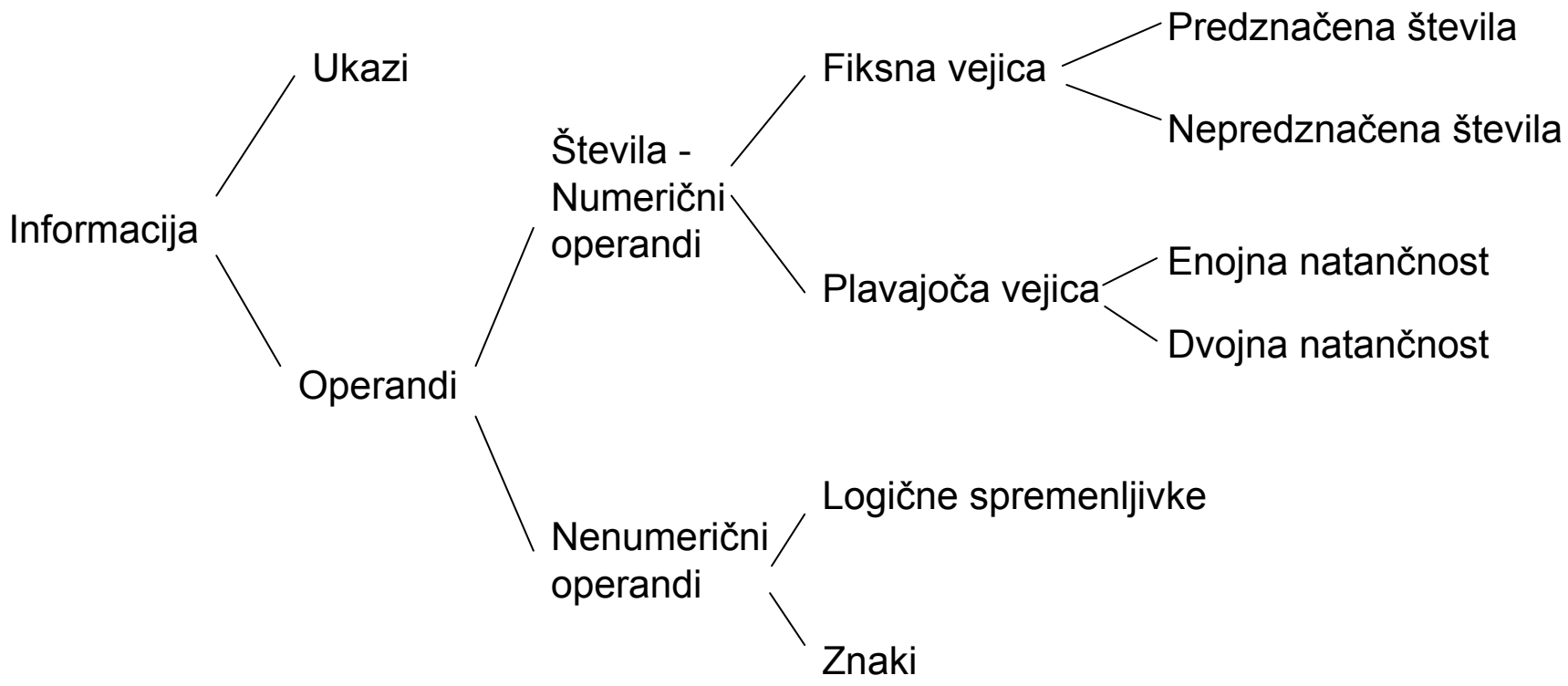
4 Predstavitev informacije in aritmetika

© Igor Škraba, FRI

- Pomen informacije, kot jo vidi uporabnik, je zelo različen, delovanje računalnika pa je neodvisno od pomena, ki ga informaciji pripisuje uporabnik
- Za delovanje bi zadoščal en sam tip informacije, vendar bi bil tak računalnik neroden za uporabo
- Osnovna enota za hranjenje informacije v računalniku je pomnilniška beseda
- Oblika oz. predstavitev v kateri je informacija shranjena, je odvisna od njenega pomena

© Igor Škraba, FRI

Osnovni tipi informacije v računalniku



 © Igor Škraba, FRI

4.1 Predstavitev nenumeričnih operandov

- Nenumerični operandi so običajno v obliki različno dolgih nizov (angl. string), ki jih sestavljajo znaki (angl. character)
- Vsak znak je predstavljen z neko abecedo
- Abeceda je predpis, ki določa preslikavo elementov ene množice v elemente druge množice

© Igor Škraba, FRI

Vrste abeced, ki se uporabljajo v računalnikih

- **BCD abeceda**
  - 6-bitna (2^6 = 64 različnih znakov)
  - 26 črk angleške abecede, 10 številk, 28 posebnih znakov
  - V uporabi do leta 1964 (6-bitne besede)
- Danes v uporabi 8-bitni abecedi EBCDIC in ASCII ter 16-bitna abeceda Unicode

© Igor Škraba, FRI

- **EBCDIC abeceda** (8-bitna)
  - Uporablja predvsem firma IBM na velikih računalnikih
- **ASCII abeceda** (8-bitna)
  - V osnovi 7-bitna, vendar se danes v računalnikih uporablja 8-bitna oblika
  - Bit 7 = 0 - osnovna oblika
  - Bit 7 = 1 - razširjena ASCII abeceda, definiranih je dodatnih 128 znakov (IBM PC)
  - Dodatnih 128 znakov je za različne države različnih in tvorijo nacionalne ASCII abecede (npr. Latin2 = ISO 8859-2)

 © Igor Škraba, FRI

Osnovna 7-bitna ASCII abeceda

Biti 3210	Biti 645							
	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	‘	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	o
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	’	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	I	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	–	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

NUL	Null/Idle	VT	Vertical tab	EM	End of medium
SOH	Start of header	FF	Form feed	SUB	Substitute
STX	Start of text	CR	Carriage return	ESC	Escape
ETX	End of text	SO	Shift out	FS	File separator
EOT	End of transmission	SI	Shift in	GS	Group separator
ENQ	Enquiry	DLE	Data link escape	RS	Record separator
ACK	Acknowledge	DC1-DC4	Device control	US	Unit separator
BEL	Audible signal	NAK	Negative acknowledge	SP	Space
BS	Backspace	SYN	Synchronous idle	DEL	Delete
HT	Horizontal tab	ETB	End of transmitted block		
LF	Line feed	CAN	Cancel		

Bitni format = 6 5 4 3 2 1 0

D.Kodek: Arhitektura računalniških sistemov, str.83

Razširjena 8-bitna ASCII abeceda Latin2 dodatni znaki(b_7=1)

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
128	**8**	€	□	‚	*ƒ*	„	…	†	‡	ˆ	‰	Š	‹	Œ	□	Ž	□
144	**9**	□	‘	’	“	”	•	–	—	˜	™	š	›	œ	□	ž	Ÿ
160	**A**		¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	-	®	¯
176	**B**	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
192	**C**	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
208	**D**	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
224	**E**	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
240	**F**	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

© Igor Škraba, FRI

- **Unicode abeceda**
  - BMP (Basic Multilingual Plane) 16-bitna abeceda (2^{16}=65 536 različnih znakov)
  - Omogoča predstavitev znakov v večini svetovnih jezikov
  - Uporaba v programskem jeziku Java
  - Razširitve:
  - UTF-8 za HTML -Unicode UTF-8 ustreza ASCII znakom
  - UTF-32

 © Igor Škraba, FRI

4.2 Predstavitev števil v fiksni vejici

- Predstavitev števil (numeričnih operandov) v fiksni vejici je izpeljana iz desetiške predstavitve
- Pozicijska notacija - pri kateri ima vsaka številka svojo težo v skladu z njenim položajem glede na decimalno vejico
- Številka desno od vejice ima težo 10^{-i}, številka levo od vejice 10^{i}

© Igor Škraba, FRI

- Dvojiška predstavitev (osnova $r = 2$)

- Zaporedje $b_{N-1}\ldots b_2 b_1 b_0$, $b_{-1}b_{-2}\ldots b_{-M}$ predstavlja število, ki ima vrednost:

$$V(b) = \sum_{i=-M}^{N-1} b_i 2^i, \qquad b_i = 0.ali.1$$

- Vejica je na vnaprej določenem fiksnem mestu - zapis s fiksno vejico

 © Igor Škraba, FRI

- Pri vseh številih, ki nastopajo v aritmetični operaciji, mora biti vejica na istem mestu – zato na računanje ne vpliva in jo lahko postavimo na poljubno mesto
- Če je vejica desno od bita z najnižjo težo, je število celo število (integer)
- Cela števila ali integerji so pravzaprav sinonim za predstavitev s fiksno vejico

 © Igor Škraba, FRI

- Za cela števila s predznakom se uporabljajo (ali so se uporabljali) štirje načini predstavitev:
  - Predznak in velikost
  - Predstavitev z odmikom
  - Eniški komplement
  - Dvojiški komplement
- *n*-bitno zaporedje $b_{n-1}\ldots b_2 b_1 b_0$ v vsakem od načinov predstavlja neko predznačeno celo število

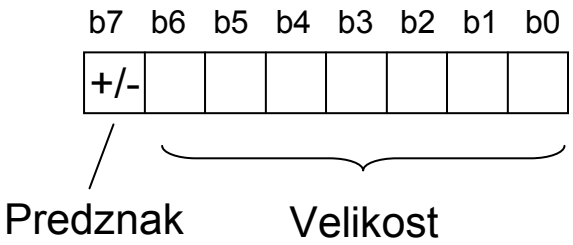
b7	b6	b5	b4	b3	b2	b1	b0	
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	uteži posameznih bitov

© Igor Škraba, FRI

Predstavitev s predznakom in velikostjo

- Bit b_{n-1} (najbolj levi bit) predstavlja predznak po pravilu
  - b_{n-1}= 0 - pozitivno število
  - b_{n-1} = 1 - negativno število

- Preostali biti predstavljajo vrednost (velikost) števila



- *n*-bitno zaporedje predstavlja desetiško število

$$V(b) = (-1)^{b_{n-1}} \sum_{i=0}^{n-2} b_i 2^i, \qquad b_i = 0.ali.1$$

- *8*-bitno zaporedje npr.

b7	b6	b5	b4	b3	b2	b1	b0
1	0	1	1	0	0	1	0

predstavlja desetiško število

$$V(b) = (-1)^{b_7} \sum_{i=0}^{6} b_i 2^i = (-1)^1 \sum_{i=0}^{6} b_i 2^i$$
$$= -(0 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 + 0 \cdot 2^3 + 1 \cdot 2^4 + 1 \cdot 2^5 + 0 \cdot 2^6) =$$
$$= -(2 + 16 + 32) = -50$$

 © Igor Škraba, FRI

- Za računalnike ta predstavitev ni najboljša, ker je treba bit za predznak pri seštevanju in odštevanju obravnavati drugače kot ostale bite
- Slabost je tudi, da imamo za ničlo dve predstavitvi +0 in -0
- Uporablja se pogosto za podajanje mantise v predstavitvi števil s plavajočo vejico

 © Igor Škraba, FRI

Predstavitev z odmikom

- Številu se najprej prišteje konstanta, kar zagotovi, da je vrednost vedno pozitivna
- Tako dobljeno število se predstavi v dvojiški obliki
- Konstanta - odmik (angl. bias) - je pri *n*-bitnem zaporedju enaka 2^{n-1}, lahko pa tudi $2^{n-1}-1$

© Igor Škraba, FRI

- *n*-bitno zaporedje predstavlja desetiško število

$$V(b) = \sum_{i=0}^{n-1} b_i 2^i - 2^{n-1}, \qquad b_i = 0.ali.1$$

- *8*-bitno zaporedje npr.

b7	b6	b5	b4	b3	b2	b1	b0
1	0	1	1	0	0	1	0

predstavlja desetiško število (odmik = $2^{8-1} = 2^7$)

$$V(b) = \sum_{i=0}^{7} b_i 2^i - 2^7 =$$
$$= (0 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 + 0 \cdot 2^3 + 1 \cdot 2^4 + 1 \cdot 2^5 + 0 \cdot 2^6 + 1 \cdot 2^7) - 2^7 =$$
$$= (2 + 16 + 32 + 128) - 128 = 50$$

 © Igor Škraba, FRI

- Dobra lastnost je, da vrednost zaporedja bitov odraža velikost števila (če ga obravnavamo kot pozitivno število)
- Pri seštevanju in odštevanju je treba odmik odšteti oziroma prišteti
- Danes se ne uporablja več, razen za podajanje eksponenta v predstavitvi s plavajočo vejico

 © Igor Škraba, FRI

Eniški komplement

- Bit b_{n-1} predstavlja predznak
- Pozitivna števila (b_{n-1}=0) so predstavljena enako kot pri predstavitvi s predznakom in velikostjo
- Predstavitev negativnega števila pa dobimo tako, da v ustreznem pozitivnem številu invertiramo vse bite, vključno z bitom za predznak (zamenjamo 0 z 1 in 1 z 0)

© Igor Škraba, FRI

- *n*-bitno zaporedje predstavlja desetiško število

$$V(b) = \sum_{i=0}^{n-1} b_i 2^i - b_{n-1}(2^n - 1), \qquad b_i = 0.ali.1$$

- *8*-bitno zaporedje npr.

b7	b6	b5	b4	b3	b2	b1	b0
1	0	1	1	0	0	1	0

predstavlja desetiško število

$$V(b) = \sum_{i=0}^{7} b_i 2^i - b_7(2^8 - 1) =$$
$$= (0 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 + 0 \cdot 2^3 + 1 \cdot 2^4 + 1 \cdot 2^5 + 0 \cdot 2^6 + 1 \cdot 2^7) - (2^8 - 1) =$$
$$= (2 + 16 + 32 + 128) - 255 = -77$$

 © Igor Škraba, FRI

- Taki predstavitvi rečemo **komplement**, aritmetično pa je invertiranje ekvivalentno odštevanju od števila 2^n-1 (=11111…111)

- Prednost predstavitve negativnih števil s komplementom je, da pri seštevanju in odštevanju bit za predznak upoštevamo enako kot ostale bite

- Slabost eniškega komplementa sta dve predstavitvi za ničlo (+0 in -0) in korekcija rezultata v primeru prenosa pri seštevanju

 © Igor Škraba, FRI

Dvojiški komplement

- Predstavitev podobna eniškemu komplementu, vsi biti b_i se invertirajo, nato pa se k temu prišteje 1
- Aritmetično je to enako odštevanju od 2^n
- Prednost dvojiškega komplementa pred eniškim je, da nimamo dveh predstavitev za ničlo

© Igor Škraba, FRI

- *n*-bitno zaporedje predstavlja desetiško število

$$V(b) = \sum_{i=0}^{n-1} b_i 2^i - b_{n-1} 2^n, \qquad b_i = 0.ali.1$$

- *8*-bitno zaporedje npr.

b7	b6	b5	b4	b3	b2	b1	b0
1	0	1	1	0	0	1	0

predstavlja desetiško število

$$V(b) = \sum_{i=0}^{7} b_i 2^i - b_7 2^8 =$$
$$= (0 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 + 0 \cdot 2^3 + 1 \cdot 2^4 + 1 \cdot 2^5 + 0 \cdot 2^6 + 1 \cdot 2^7) - 2^8 =$$
$$= (2 + 16 + 32 + 128) - 256 = -78$$

 © Igor Škraba, FRI

- Seštevanje in odštevanje potekata enako kot pri *n*-bitnih pozitivnih številih, korekcija ni potrebna, prenos iz *n-1* mesta enostavno zanemarimo
- Dvojiški komplement je danes najbolj razširjen način za predstavitev števil v fiksni vejici
- Lepotna napaka predstavitve v dvojiškem komplementu je, da lahko predstavimo eno negativno število več kot pozitivno

 © Igor Škraba, FRI

- ! V komplementu so zapisana samo negativna števila

- Pri vsaki predstavitvi števil s končnim številom bitov obstaja najmanjše in največje število, ki ga še lahko predstavimo

- Največje in najmanjše predstavljivo *n*-bitno število predstavljeno v dvojiškem komplementu

$$-2^{n-1} \le x \le 2^{n-1} - 1$$

$$n = 8 \qquad -128_D \le x \le +127_D$$

© Igor Škraba, FRI

- **Preliv** (angl. overflow) - če je rezultat operacije izven tega področja

- Najmanjše in največje predstavljivo nepredznačeno (pozitivno) število

$$0 \le x \le 2^n - 1$$

$$n = 8 \qquad 0_D \le x \le 255_D$$

- **Prenos** (angl. carry) - če je rezultat seštevanja ali odštevanja pozitivnih (nepredznačenih) števil izven območja, pride do prenosa iz najvišjega mesta

 © Igor Škraba, FRI

- Preliv ali prenos je lahko vzrok za napako
- V CPE mora biti vgrajen mehanizem, s pomočjo katerega lahko programer ugotovi, ali je pri rezultatu operacije prišlo do prenosa ali preliva
- Bita C (carry) in V (overflow) v registru pogojev v CPE, ki se postavita na 1, kadar pride pri operaciji do prenosa oziroma preliva

 © Igor Škraba, FRI

4.3 Aritmetika s števili v fiksni vejici

- Aritmetika - štiri osnovne operacije: seštevanje, odštevanje, množenje in deljenje
- Aritmetične operacije se izvajajo v aritmetično-logični enoti (ALE), ki je del CPE
- Vrsta in število operacij, ki jih zna izvajati ALE se med računalniki razlikujeta - pri najpreprostejših samo seštevanje in logične operacije, druge operacije so realizirane s programi

© Igor Škraba, FRI

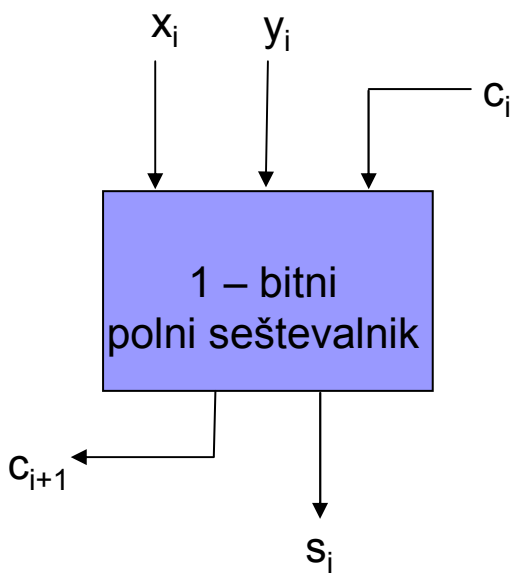
- Pri hitrejših računalnikih želimo pogosto doseči, da se vsaka aritmetična operacija opravi v eni urini periodi
- S programsko izvedbo operacij to ni mogoče
- Tudi ostale aritmetične operacije moramo realizirati z digitalnimi vezji
- Ključno vezje za realizacijo aritmetičnih operacij je paralelni dvojiški seštevalnik

 © Igor Škraba, FRI

- Z njim so narejene vse operacije, tudi odšte-vanje, množenje in deljenje
- *n*-bitni paralelni seštevalnik je vezje, ki iz dveh nepredznačenih *n*-bitnih števil tvori njuno vsoto
- Osnovni element, s katerim zgradimo *n*-bitni seštevalnik, je *1*-bitni polni seštevalnik

 © Igor Škraba, FRI

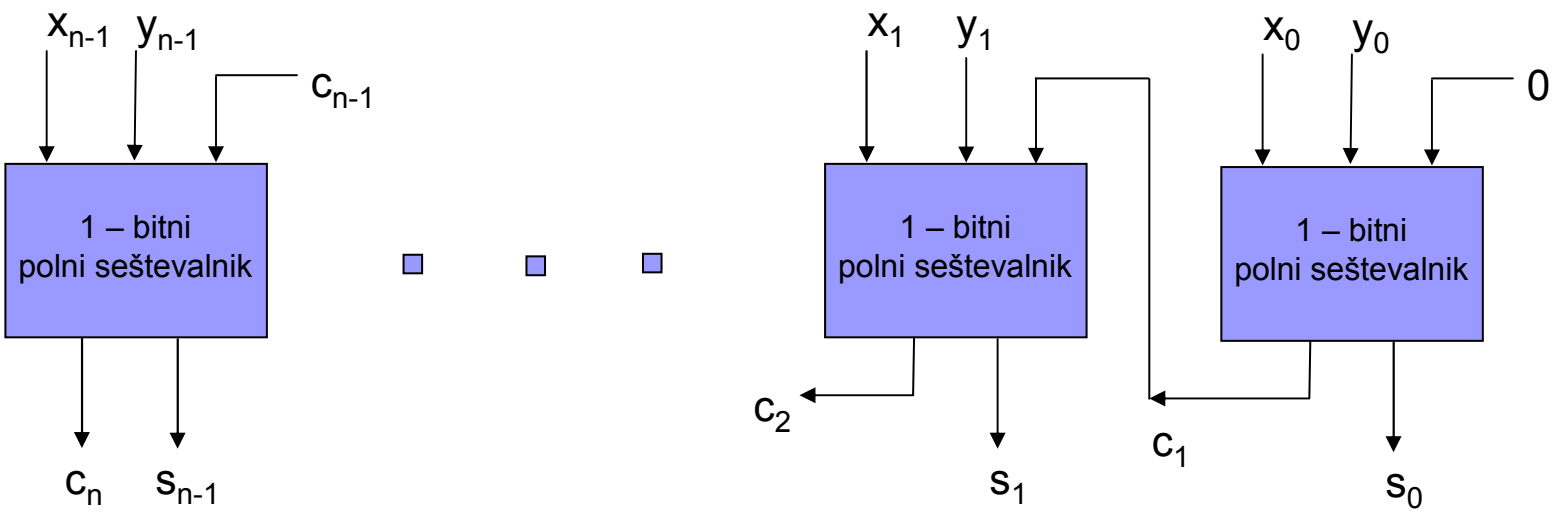
- *1*-bitni polni seštevalnik ima tri vhode
  - Dva sumanda x_i in y_i
  - Vhodni prenos c_i
- in dva izhoda
  - Vsota s_i
  - Izhodni prenos c_{i+1}



x_i	y_i	c_i	s_i	c_{i+1}
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

 © Igor Škraba, FRI

- *n*-bitni seštevalnik dobimo, če povežemo *n* eno-bitnih seštevalnikov - seštevalnik s plazovitim prenosom



4.4 Predstavitev števil v plavajoči vejici

- Obseg števil, ki jih lahko predstavimo v predstavitvi s fiksno vejico, je za tehnične probleme običajno premajhen

- Ta števila pišemo običajno v znanstveni notaciji, ki omogoča predstavitev z razmeroma malo številkami

$$3.200.000 = 3{,}2 \cdot 10^6 = 0{,}32 \cdot 10^7 = 32 \cdot 10^5 =$$

- Predstavitev števil v plavajoči vejici je samo za računalnik prirejena oblika znanstvene notacije

© Igor Škraba, FRI

- Splošna oblika $m \cdot r^e$
  - *m* – mantisa (koeficient, fraction, significand)
  - *r* – baza (osnova, radiks)
  - *e* – eksponent (karakteristika)

- Vloga eksponenta je, da določa na katerem mestu je decimalna vejica

- Pri predstavitvi s fiksno vejico je vejica vedno na istem mestu, tu pa se njen položaj spreminja (plava vzdolž mantise levo ali desno) odvisno od vrednosti eksponenta *e*

 © Igor Škraba, FRI

- V računalniku sta mantisa in eksponent predstavljena s fiksno vejico, baza pa je 2 ali 10 in seveda pri obeh enaka

- Ker je baza konstanta, je število v plavajoči vejici določeno s parom *(m,e),* kjer sta *m* in *e* predznačeni števili v fiksni vejici

- Natančnost števila je določena s številom bitov za predstavitev mantise, največje in najmanjše predstavljivo število pa z eksponentom *e*

 © Igor Škraba, FRI

- Z *n*-bitnim zaporedjem lahko v fiksni in plavajoči vejici predstavimo največ 2^n števil, vendar lahko v plavajoči vejici predstavimo tudi ulomke (v fiksni vejici samo cela števila)
- Povečanje števila bitov za eksponent na račun mantise poveča velikost intervala, števila pa so porazdeljena redkeje
- Števila predstavljena v plavajoči vejici imenujemo tudi realna števila

 © Igor Škraba, FRI

- **Normalizirano število** - število v plavajoči vejici je normalizirano, kadar mantisa na levi strani nima ničel ali, prvi bit mantise b_{n-1} mora biti različen od nič

- Dogovor o položaju vejice - 3 možnosti
  - Vejica je tik pred prvo številko mantise
  - Vejica je tik za prvo številko mantise
  - Vejica je desno od zadnje številke mantise (mantisa je celo število

 © Igor Škraba, FRI

- Za predstavitev mantise se danes najpogosteje uporablja predstavitev **predznak in velikost**
- Prvi bit normalizirane mantise je vedno ena - imenuje se **normalni bit** (seveda če je baza 2)
- Ker je normalni bit vedno ena, ga pri predstavitvi lahko spustimo - **implicitna predstavitev normalnega bita** - pri računanju ga seveda upoštevamo

 © Igor Škraba, FRI

- Tako pridobimo en bit pri predstavitvi mantise - večja natančnost (standard IEEE 754)
- Implicitna predstavitev normalnega bita je možna samo pri bazi $r = 2$
- Števila, ki niso normalizirana, so **denormalizirana**. Števil, ki so po absolutni vrednosti zelo majhna, v normalizirani obliki ni mogoče predstaviti - **podliv** (eksponent doseže svojo najmanjšo vrednost)

 © Igor Škraba, FRI

- Eksponent je predstavljen z odmikom, kar omogoča, da bitni vzorec samih ničel predstavlja najbolj negativen eksponent (najmanjši)
- Na različnih računalnikih so bili za predstavitev števil v plavajoči vejici uporabljeni različni formati
- Posledica: isti program, napisan v nekem višjem programskem jeziku, je na različnih računalnikih dal nekoliko različne rezultate

 © Igor Škraba, FRI

Standard za predstavitev v plavajoči vejici

- Leta 1981 je bil v okviru organizacije IEEE predlagan standard za aritmetiko s plavajočo vejico, leta 1985 pa sprejet v končni obliki z oznako IEEE 754

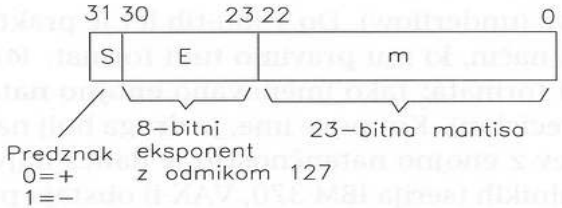
- Poleg formata za predstavitev števil določa standard še načine izvajanja aritmetičnih operacij (zaokroževanje) in postopke v primeru napak (preliv, deljenje z 0, itn.)

© Igor Škraba, FRI

- Osnovne lastnosti predstavitve števil v standardu IEEE 754
  - Standard uporablja bazo $r = 2$
  - Mantisa je predstavljena v načinu predznak in velikost
  - Implicitna predstavitev normalnega bita. Vejica je desno od normalnega bita (= levo od prvega bita mantise)
  - Eksponent je predstavljen v predstavitvi z odmikom
  - Definirana sta dva formata: *32*-bitni format ali enojna natančnost in *64*-bitni format ali dvojna natančnost

 © Igor Škraba, FRI

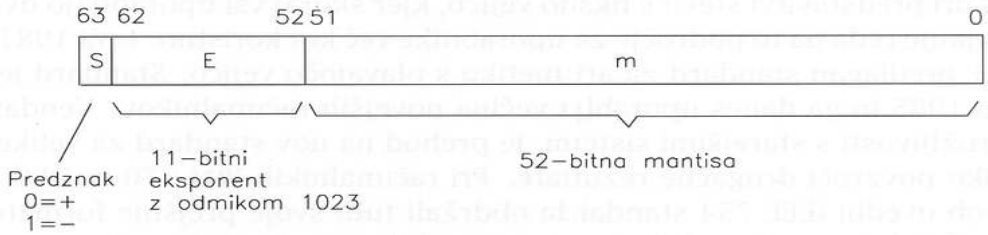
Predstavitev števil v plavajoči vejici – standard IEEE 754 32-bitni in 64-bitni format



$$\text{vrednost} = (-1)^S (1,m) 2^{E-127}, \quad 1 \le E \le 254$$

obseg: $\pm 1{,}18 \times 10^{-38}$ do $\pm 3{,}40 \times 10^{38}$ (v normalizirani obliki)

a) 32–bitni format (enojna natančnost)



$$\text{vrednost} = (-1)^S (1,m) 2^{E-1023}, \quad 1 \le E \le 2046$$

obseg: $\pm 2{,}22 \times 10^{-308}$ do $\pm 1{,}80 \times 10^{308}$ (v normalizirani obliki)

b) 64–bitni format (dvojna natančnost)

D.Kodek: Arhitektura računalniških sistemov, str.102

- Standard IEEE 754 določa predstavitev petih vrst števil:
  - Normalizirana števila - eksponent *E* nima vseh bitov enakih 0 ali vseh bitov enakih 1 (večina števil je normaliziranih, ostala so izjeme)
  - Denormalizirana števila - eksponent *E* ima vse bite enake 0
  - Ničli (+0 in -0) - eksponent *E* in mantisa *m* imata vse bite enake 0
  - Neskončnost ($+\infty$ in $-\infty$) - eksponent *E* ima vse bite enake 1, mantisa *m* vse bite enake 0
  - Neveljavna števila (NaN) - eksponent *E* ima vse bite enake 1, mantisa *m* je različna od 0

 © Igor Škraba, FRI

Aritmetika s števili v plavajoči vejici

- Aritmetika v plavajoči vejici se je v računalnikih obravnavala ločeno od aritmetike v fiksni vejici
- Največkrat je bila realizirana programsko, včasih z možnostjo dograditve dodatne logike za računanje v plavajoči vejici (matematični koprocesor)
- Enota (logika) za operacije v plavajoči vejici je bila vgrajena predvsem v računalnike, ki so bili namenjeni za reševanje znanstvenih in tehničnih problemov

© Igor Škraba, FRI

- Danes so enote za operacije v plavajoči vejici (FPU - floating point unit) v večini računalnikov standardne, nimajo jih samo najmanjši računalniki

- Osnovni razliki glede na operacije v fiksni vejici sta:
  - Pri operacijah je treba poleg mantise uporabiti še eksponent – za te operacije je potrebna aritmetika v fiksni vejici
  - Zaokroževanje – rezultat operacije mora biti enak matematično točni vrednosti, ki se zaokroži na dolžino bitov mantise

 © Igor Škraba, FRI