Predstavljanje podataka u računaru

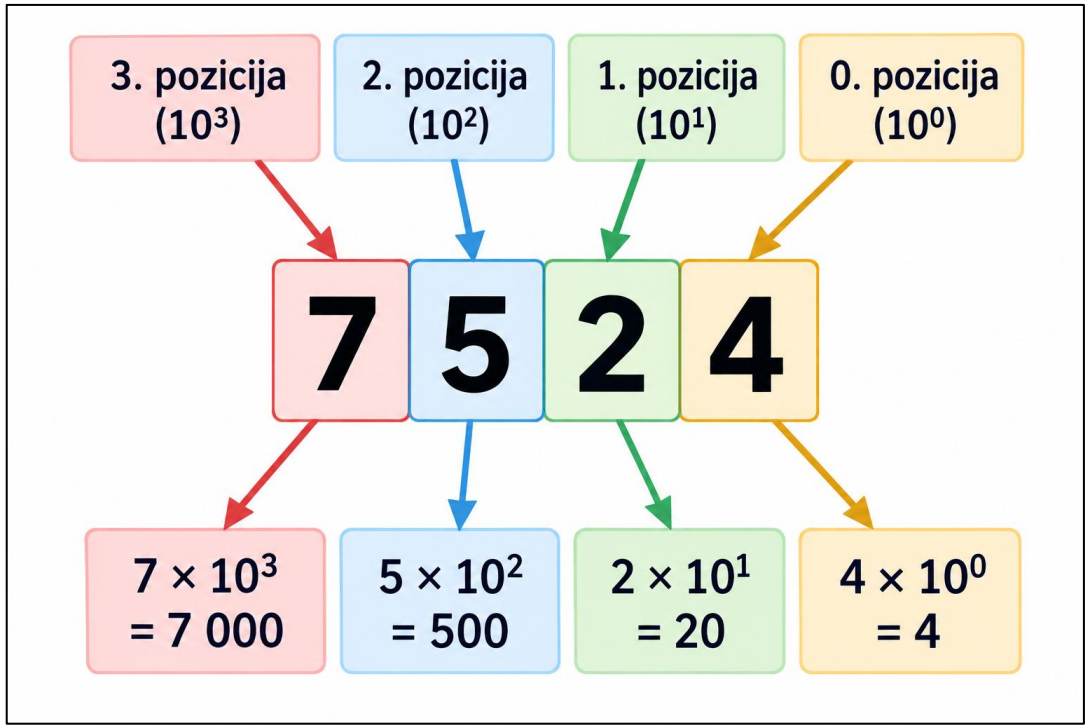
Sadržaj predavanja

- **Brojni sistemi** – pozicioni brojni sistemi i osnova brojnog sistema
- **Dekadni, binarni, oktalni i heksadekadni sistem**
- **Prevođenje brojeva između brojnih sistema**
- **Binarno predstavljanje podataka u računaru**
- **Bit, bajt i predstavljanje znakova**
- **Kodiranje znakova** – ASCII, kodne strane i Unicode
- **Primeri predstavljanja podataka u računaru**

Pozicioni brojni sistemi

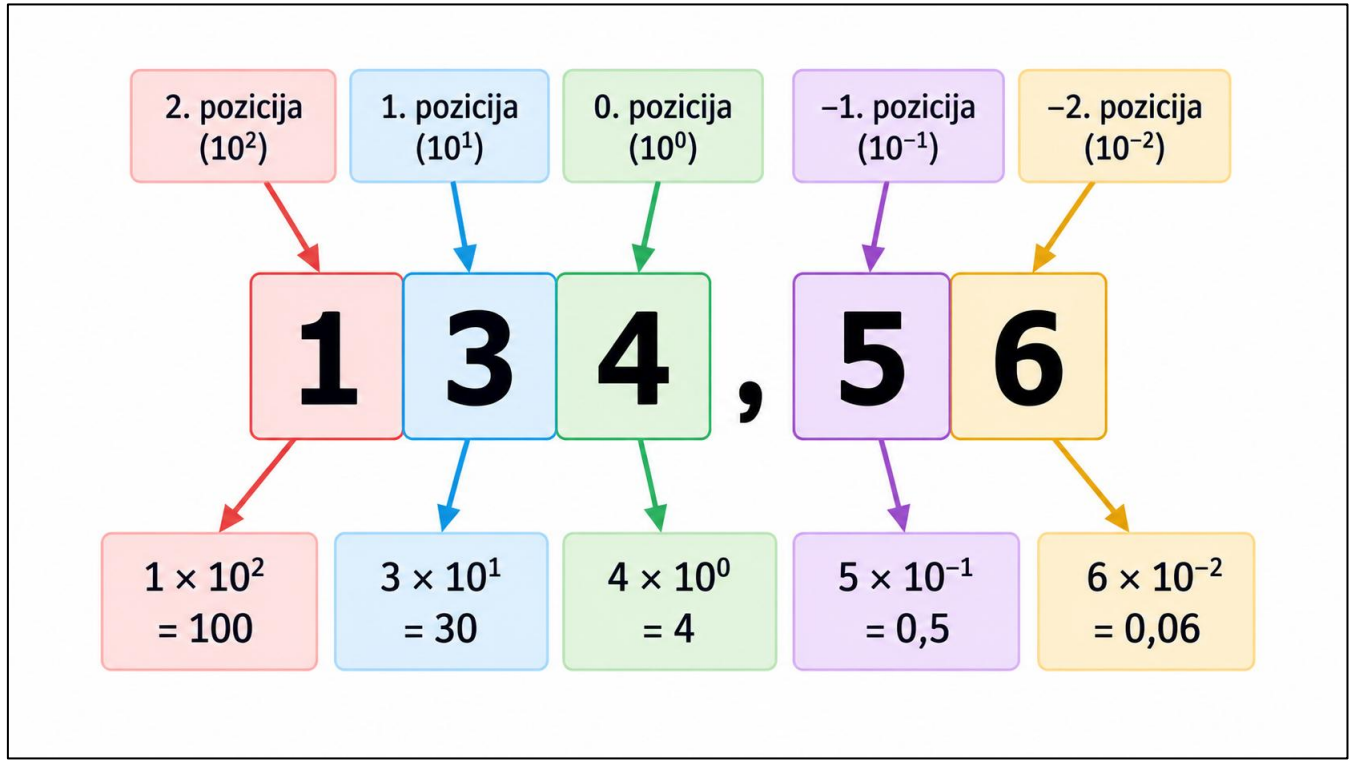
- **Pozicioni brojni sistem** predstavlja sistem u kome vrednost cifre zavisi od njenog položaja u broju
- **Osnova brojnog sistema** određuje broj različitih cifara koje se koriste za zapisivanje brojeva
- U **dekadnom brojnom sistemu** osnova je 10 i koriste se cifre 0, 1, 2, 3, 4, 5, 6, 7, 8 i 9
- Svaka pozicija cifre ima svoju **težinu**, koja odgovara stepenu osnove brojnog sistema
- Na primer, broj **384,27** može se zapisati kao: $3 \times 10^2 + 8 \times 10^1 + 4 \times 10^0 + 2 \times 10^{-1} + 7 \times 10^{-2}$
- Isti princip pozicionog zapisivanja primenjuje se i u drugim brojnim sistemima, uz različite osnove.

Ceo broj – dekadni sistem



- Svaka **pozicija cifre** ima svoju težinu, određenu stepenom osnove
- Kod celog broja pozicije se računaju **s desna nalevo**, počevši od pozicije 0 koja pripada cifri jedinica.
- Broj **7 524** možemo rastaviti na vrednosti pojedinačnih cifara.
- **$7\,524 = 7 \times 10^3 + 5 \times 10^2 + 2 \times 10^1 + 4 \times 10^0$**
- **7 524 = 7 000 + 500 + 20 + 4**

Decimalni broj – dekadni sistem



$134{,}56 = 1 \times 10^2 + 3 \times 10^1 + 4 \times 10^0 + 5 \times 10^{-1} + 6 \times 10^{-2}$

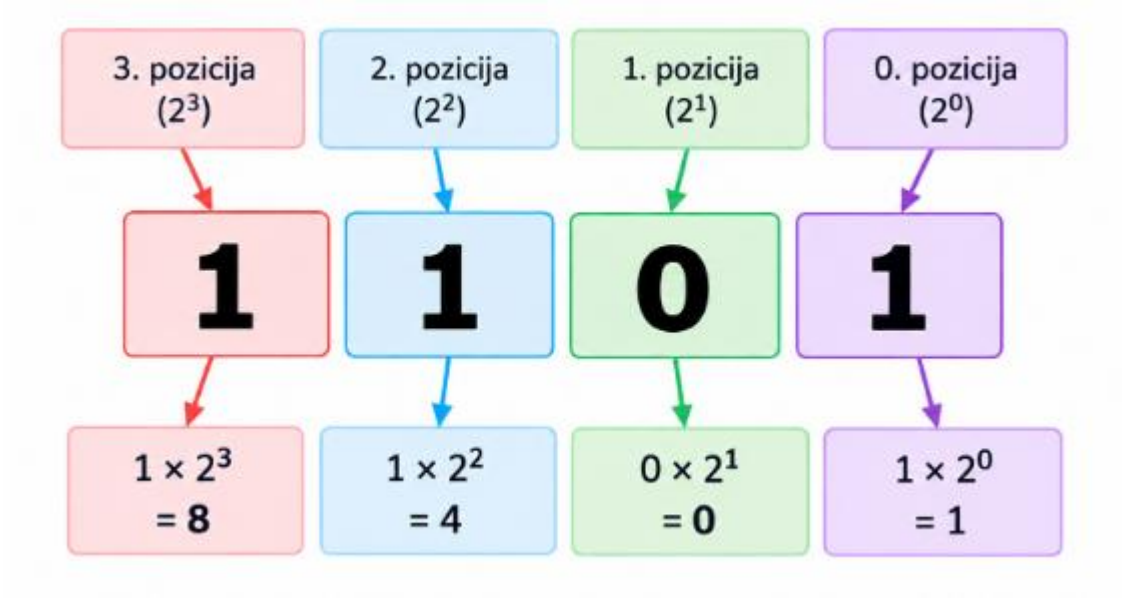
$= 100 + 30 + 4 + 0{,}5 + 0{,}06$

$= \mathbf{134{,}56}$

Binarni brojni sistem

- **Binarni brojni sistem** je pozicioni brojni sistem sa osnovom **2**.
- Za zapisivanje brojeva koristi samo dve cifre: **0 i 1**
- Vrednost svake cifre određena je njenom **pozicijom i težinskim koeficijentom**, koji predstavlja odgovarajući stepen osnove 2
- Binarni sistem je od posebnog značaja za računare zato što elektronske komponente mogu da rade sa **dva jasno razdvojena stanja**
- Ta stanja možemo predstaviti kao: **ima napona / nema napona** ili **ima struje / nema struje**
- Na taj način se fizička stanja elektronskih komponenti mogu povezati sa binarnim vrednostima **1 i 0**, što omogućava pouzdano predstavljanje i obradu podataka
- **Svi podaci** koje računar obrađuje, uključujući brojeve, tekst, slike i zvuk, na nivou računara predstavljaju se kombinacijama binarnih cifara

Primer binarnog broja



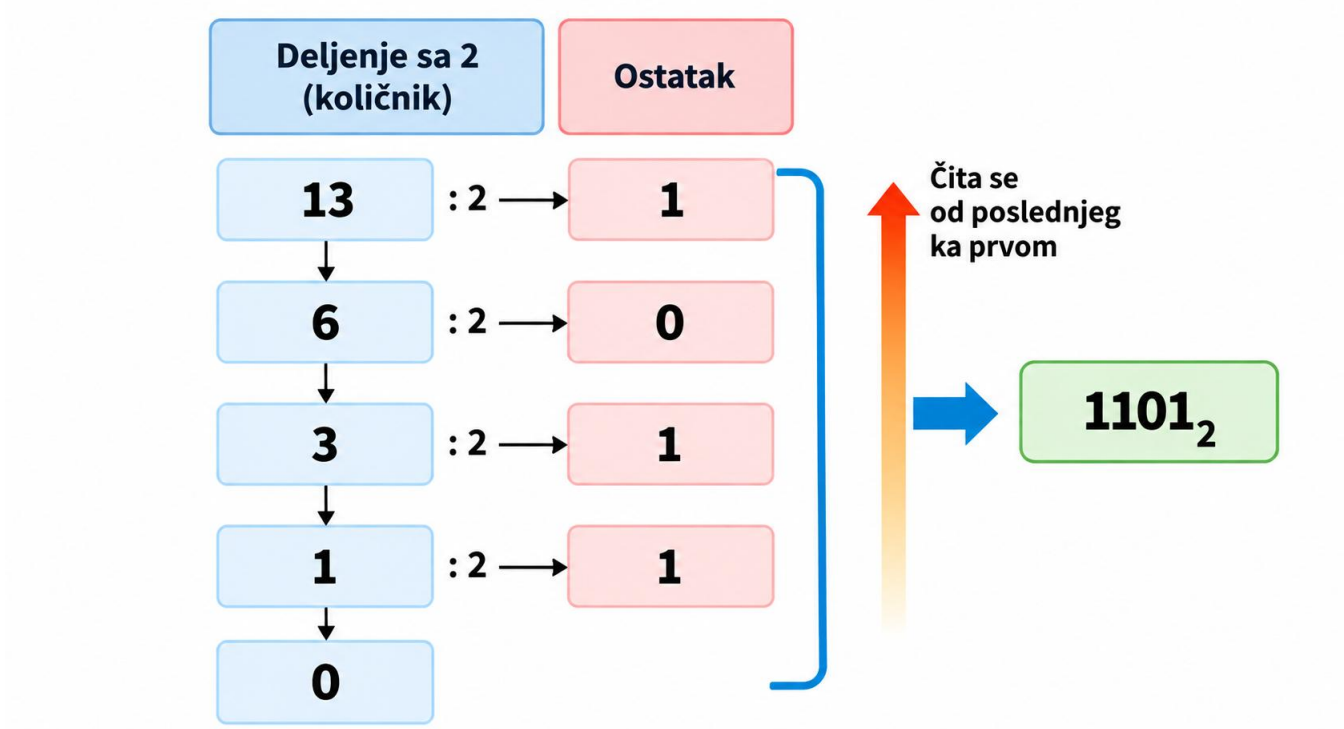
$1101_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

$= 8 + 4 + 0 + 1 = \mathbf{13_{10}}$

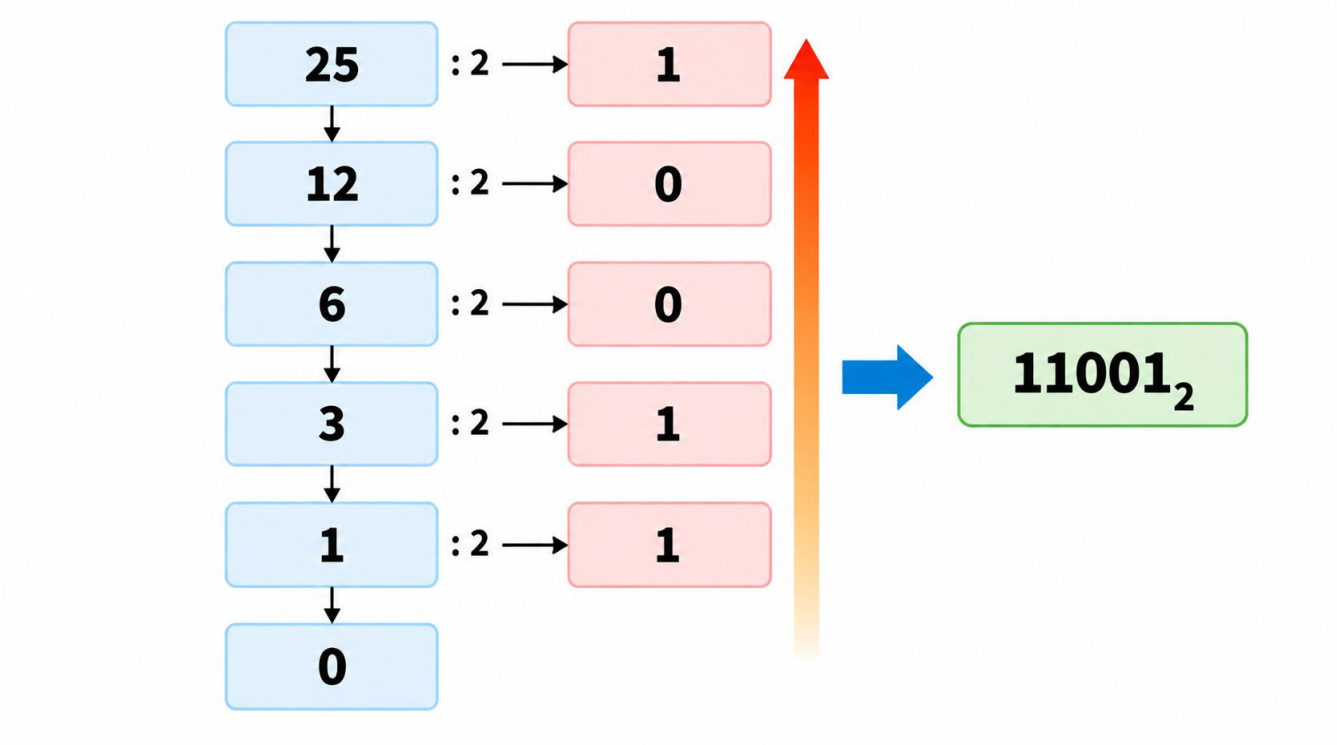
Prevođenje dekadnog broja u binarni

- **Prevođenje dekadnog celog broja u binarni** vrši se uzastopnim deljenjem broja sa 2
- Pri svakom deljenju beleži se **ostatak**, koji može biti samo 0 ili 1
- Dobijeni količnik se ponovo deli sa 2 sve dok količnik ne postane 0
- Binarne cifre dobijaju se **čitanjem ostataka obrnutim redosledom**, od poslednjeg ka prvom
- Na taj način se dekadni broj razlaže na zbir odgovarajućih **težinskih koeficijenata stepena broja 2**
- Isti postupak može se primeniti na svaki dekadni ceo broj, bez obzira na njegovu veličinu

Prevođenje dekadnog broja u binarni - primer



Prevođenje dekadnog broja u binarni - primer



Prevođenje čistog decimalnog broja u binarni

- Čisti decimalni broj prevodi se u binarni sistem uzastopnim množenjem sa 2
- Pri svakom množenju zapisuje se ceo deo rezultata – 0 ili 1
- Za sledeći korak uzima se razlomljeni deo rezultata (broj posle decimalnog zareza)
- Postupak se ponavlja sve dok razlomljeni deo ne bude 0 ili dok se ne dobije dovoljan broj cifara
- Dobijeni celi delovi redom čine binarne cifre posle zareza
- Ako se postupak ne završava, binarni zapis se prekida na potrebnom broju cifara

Prevođenje čistog decimalnog broja u binarni - primer

Množenje sa 2	Ceo deo (binarna cifra)	Razlomljeni deo (za sledeći korak)
0,387 × 2 = **0,774**	**0**	0,774
0,774 × 2 = **1,548**	**1**	0,548
0,548 × 2 = **1,096**	**1**	0,096
0,096 × 2 = **0,192**	**0**	0,192
0,192 × 2 = **0,384**	**0**	0,384
0,384 × 2 = **0,768**	**0**	0,768
0,768 × 2 = **1,536**	**1**	0,536
0,536 × 2 = **1,072**	**1**	0,072

Binarne cifre čitaju se redosledom kako su dobijene (od prve ka poslednjoj).

$0{,}387_{10} = 0{,}01100011_2$

Prevođenje mešovitog decimalnog broja u binarni

45,6875

- **45** – ceo deo
- **0,6875** – razlomljeni deo

Prevođenje celog dela broja u binarni

45_{10}

Deljenje sa 2 (količnik)	Ostatak (binarna cifra)	Čitanje
45 : 2 = 22	1	Čita se od poslednjeg ka prvom.
22 : 2 = 11	0	
11 : 2 = 5	1	
5 : 2 = 2	1	
2 : 2 = 1	0	
1 : 2 = 0	1	

$45_{10} = 101101_2$

Prevođenje razlomljenog dela broja u binarni

$0{,}6875_{10}$

Množenje sa 2	Ceo deo (binarna cifra)	Razlomljeni deo (za sledeći korak)
0,6875 × 2 = 1,375	1	0,375
0,3750 × 2 = 0,750	0	0,750
0,7500 × 2 = 1,500	1	0,500
0,5000 × 2 = 1,000	1	0,000

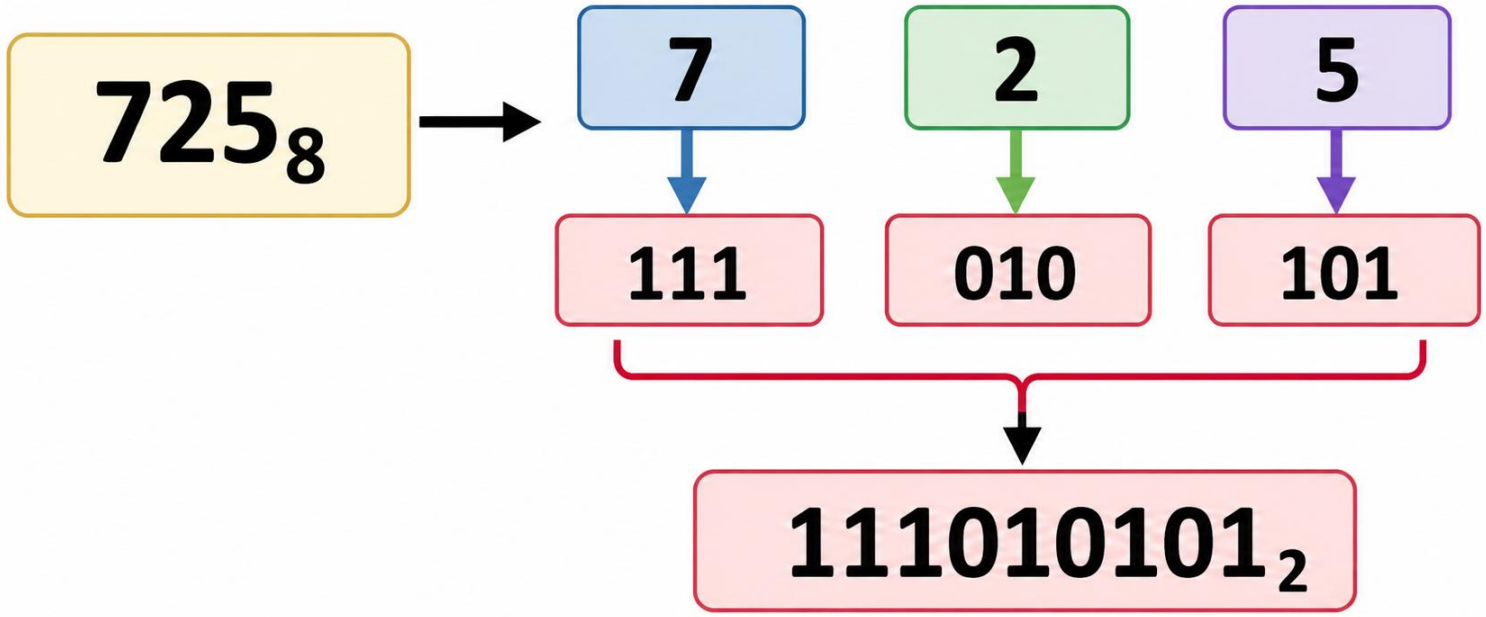
$0{,}6875_{10} = 0{,}1011_2$

$45{,}6875_{10} = 101101{,}1011_2$

Oktalni brojni sistem

- **Oktalni brojni sistem** je pozicioni brojni sistem sa osnovom **8**
- Za zapisivanje brojeva koristi **osam cifara: 0, 1, 2, 3, 4, 5, 6 i 7**
- Vrednost cifre određena je njenom **pozicijom** i odgovarajućim stepenom osnove 8
- Ispred razdvojnog zareza težine pozicija rastu **s desna na levo**: $\mathbf{8^0, 8^1, 8^2, 8^3...}$
- Iza razdvojnog zareza težine pozicija su $\mathbf{8^{-1}, 8^{-2}, 8^{-3}...}$
- Oktalni sistem je povezan sa binarnim sistemom jer **jedna oktalna cifra odgovara tačno trima binarnim ciframa**
- Koristi se za **kraći i pregledniji zapis binarnih podataka**, posebno u računarstvu i pri predstavljanju **dozvola pristupa u Unix/Linux operativnim sistemima**

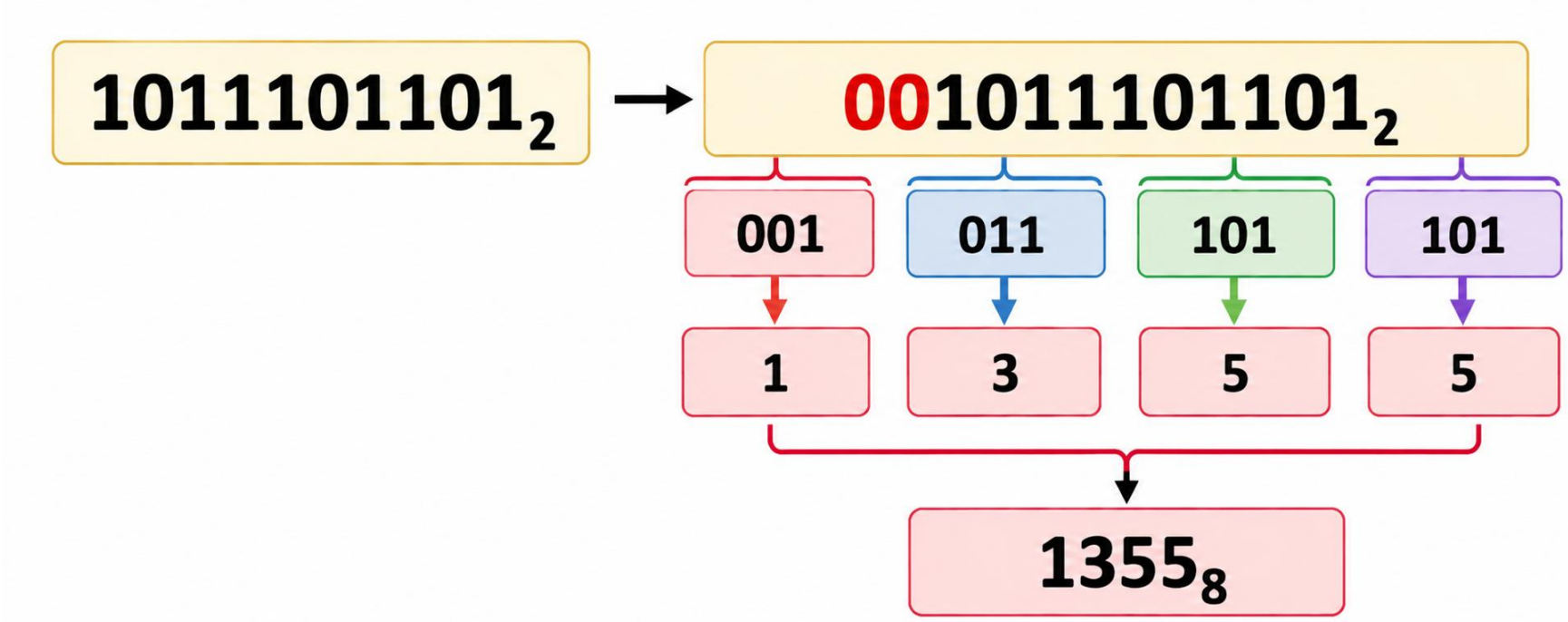
Pretvaranje oktalnog broja u binarni



Oktalna cifra	Binarni zapis (3 cifre)
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

- **Svaka oktalna cifra odgovara trima binarnim ciframa**
- Svaku oktalnu cifru zamenjujemo odgovarajućom binarnom grupom
- Dobijene binarne grupe spajamo u jedan broj

Pretvaranje binarnog broja u oktalni



- Ako broj cifara nije deljiv sa 3, **dodajemo vodeće nule**
- Binarne cifre zatim **grupišemo po tri**
- Svaku grupu zamenjujemo odgovarajućom **oktalnom cifrom**

Heksadecimalni brojni sistem

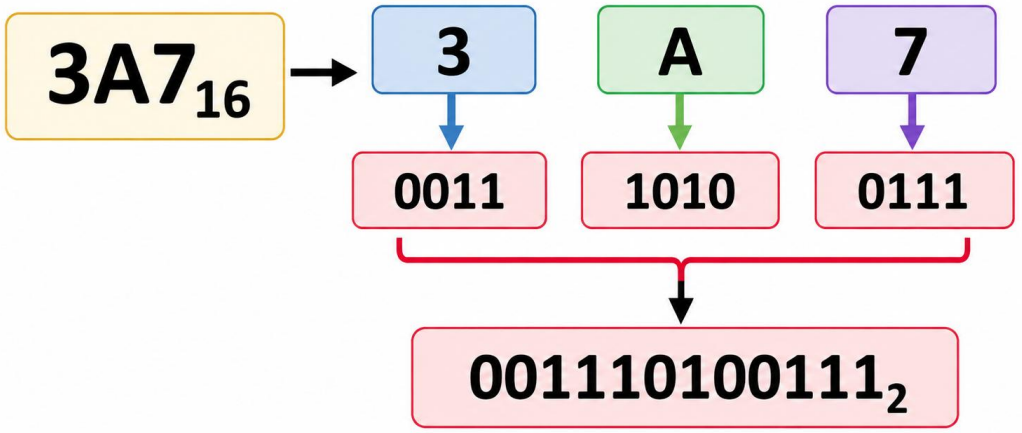
- **Heksadecimalni brojni sistem** je pozicioni brojni sistem sa osnovom **16**
- Za zapisivanje brojeva koristi **16 cifara: 0–9 i A–F**
- Slova **A, B, C, D, E i F** predstavljaju vrednosti **10, 11, 12, 13, 14 i 15**
- Vrednost cifre određena je njenom **pozicijom** i odgovarajućim stepenom osnove 16
- Heksadekadni sistem je posebno pogodan za **kraći i pregledniji zapis binarnih podataka**
- Heksadecimalni sistem koristi se i za pregledan zapis **kodnih vrednosti** znakova i drugih podataka u računaru
- **Jedna heksadecimalna cifra odgovara četiri binarne cifre**

Binarni zapis heksadecimalnih cifara

Heksadekadna cifra	Binarni zapis (4 cifre)
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

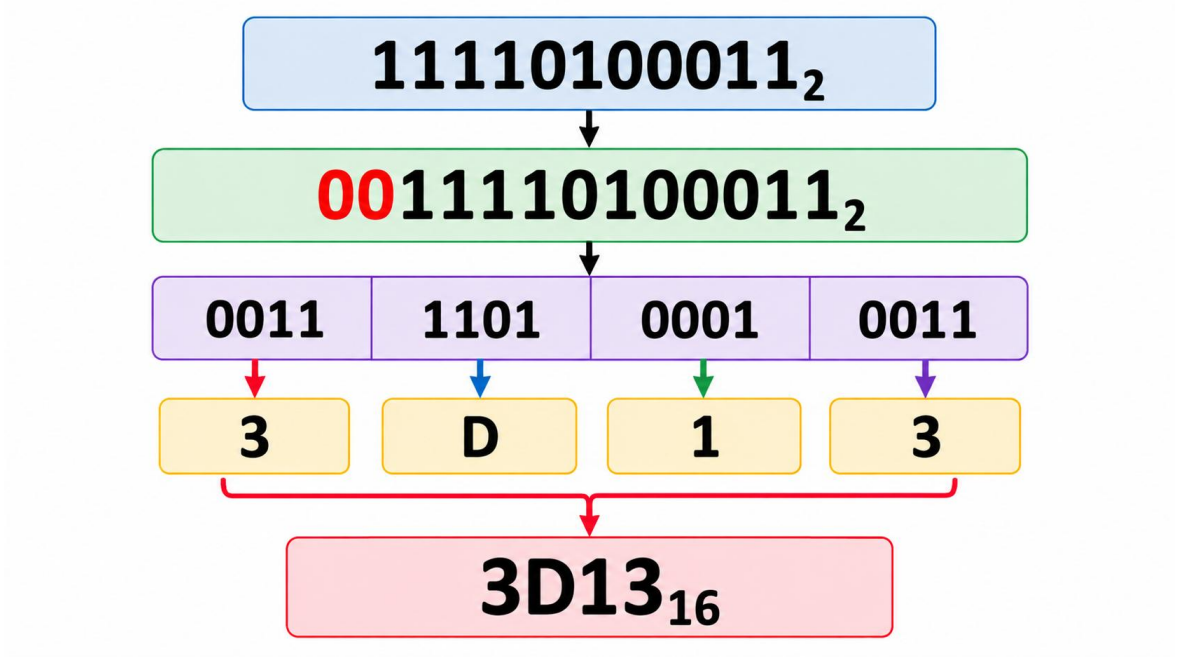
Prevođenje heksadecimalnog broja u binarni

- **Svaka heksadecimalna cifra odgovara četiri binarne cifre**
- Svaku heksadecimalnu cifru zamenjujemo odgovarajućom **četvorocifrenom binarnom grupom**
- Dobijene binarne grupe spajamo u **jedan broj**



Prevođenje binarnog broja u heksadecimalni

- **Binarni broj grupišemo po četiri cifre**, počevši od desne strane
- Ako broj cifara nije deljiv sa 4, **dodajemo vodeće nule**
- Svaku grupu od četiri binarne cifre zamenjujemo odgovarajućom **heksadecimalnom cifrom**
- Dobijene heksadecimalne cifre spajamo u **jedan broj**



Povezanost binarnog, oktalnog i heksadecimalnog sistema

- **Binarni sistem** koristi osnovu **2** i cifre 0 i 1
- **Oktalni sistem** koristi osnovu **8** i cifre 0–7
- **Heksadecimalni sistem** koristi osnovu **16** i cifre 0–9 i A–F
- **Tri binarne cifre** odgovaraju jednoj oktalnoj cifri
- **Četiri binarne cifre** odgovaraju jednoj heksadecimalnoj cifri
- Zbog ove povezanosti binarni brojevi mogu se **kraće zapisivati** u oktalnom ili heksadecimalnom obliku

Binarno predstavljanje podataka u računaru

- Računar sve podatke predstavlja pomoću **binarnih vrednosti 0 i 1**
- Binarne vrednosti odgovaraju različitim **fizičkim stanjima elektronskih komponenti**
- **Brojevi** se predstavljaju binarnim zapisom
- **Tekst** se predstavlja pomoću binarnih kodova koji odgovaraju znakovima
- **Slike, zvuk i video** takođe se predstavljaju skupovima binarnih podataka
- Zbog toga se svi podaci u računaru mogu posmatrati kao **nizovi binarnih cifara**

Binarne cifre

- **Binarna cifra** može imati samo dve vrednosti: **0 ili 1**
- Kombinovanjem više binarnih cifara dobijamo različite **binarne kombinacije**
- Sa **2 binarne cifre** možemo predstaviti **4 različite kombinacije**
- Sa **3 binarne cifre** možemo predstaviti **8 različitih kombinacija**
- Sa **4 binarne cifre** možemo predstaviti **16 različitih kombinacija**
- Sa **8 binarnih cifara** možemo predstaviti **256 različitih kombinacija**, jer je $2^8 = 256$
- Broj mogućih kombinacija raste sa brojem binarnih cifara prema izrazu $\mathbf{2^n}$, gde je n broj cifara
- Svaka kombinacija može predstavljati određenu **vrednost, znak ili drugi podatak**, u zavisnosti od primenjenog načina kodiranja

Kombinovanje binarnih cifara

2 binarne cifre	
Redni broj	Binarna kombinacija
0	00
1	01
2	10
3	11

3 binarne cifre	
Redni broj	Binarna kombinacija
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

4 binarne cifre	
Redni broj	Binarna kombinacija
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Bit i bajt

- **Bit** je osnovna jedinica za predstavljanje podataka u računaru i može imati vrednost **0 ili 1**
- Naziv **bit** potiče od engleskog izraza *binary digit*
- **Bajt** predstavlja grupu od **8 bitova**
- Sa 8 bitova može se formirati **256 različitih kombinacija**, od 00000000 do 11111111
- Jedna od osnovnih uloga bajta je predstavljanje **znakova**, kao što su slova, cifre i znakovi interpunkcije
- Veće količine podataka izražavaju se većim jedinicama, kao što su **kilobajt, megabajt, gigabajt i terabajt**
- Veličina podatka u računaru određuje koliko je **bitova i bajtova** potrebno za njegovo predstavljanje

Jedinice za količinu podataka

- **1 bajt (B) = 8 bitova (b)**
- **1 kilobajt (KB) = 1 024 bajta = 2^{10} bajta**
- **1 megabajt (MB) = 1 024 KB = 2^{10} KB**
- **1 gigabajt (GB) = 1 024 MB = 2^{10} MB**
- **1 terabajt (TB) = 1 024 GB = 2^{10} GB**
- Svaka veća jedinica predstavlja **2^{10} = 1 024 jedinice prethodnog nivoa**
- Ove jedinice koriste se za izražavanje **veličine podataka i kapaciteta memorije i uređaja za skladištenje**

ASCII kodiranje znakova

- **ASCII** je skraćenica od **American Standard Code for Information Interchange**
- Predstavlja standard za **numeričko kodiranje znakova**, kojim se svakom znaku dodeljuje određena brojčana vrednost
- Osnovni ASCII skup sadrži **128 znakova**, sa vrednostima od **0 do 127**
- Za predstavljanje 128 različitih vrednosti potrebno je **7 bitova**, jer je $\mathbf{2^7 = 128}$
- Skup obuhvata **slova engleske abecede, cifre, znakove interpunkcije i kontrolne znakove**
- Na primer, znak **A** ima ASCII vrednost **65**, odnosno binarni zapis **1000001**
- ASCII je omogućio **jednoznačno predstavljanje osnovnih znakova** u različitim računarskim sistemima i programima

Grupe ASCII znakova

- **Kontrolni znakovi (0–31)** služe za upravljanje komunikacijom, unosom i radom uređaja, a ne za prikazivanje znakova na ekranu: TAB – horizontalni tabulator, VT – vertikalni tabulator, ESC – izlaz iz trenutnog režima ili sekvence
- **Razmak, cifre i interpunkcija (32–63)** obuhvataju razmak, cifre i osnovne znakove interpunkcije i specijalne znakove: razmak, !, ", #, %, &, (,), 0–9, ?
- **Velika slova i simboli (64–95)** obuhvataju velika slova engleske abecede i određene specijalne znakove: @, A–Z, [, \,], ^, _
- **Mala slova i simboli (96–127)** obuhvataju mala slova engleske abecede i preostale specijalne odnosno kontrolne znakove: `, a–z, {, |, }, ~, DEL

ASCII tabela (0 – 127)

0 – 31 Kontrolni znakovi			
Dec	Hex	Bin	Znak / Naziv
0	00	0000000	NUL (null)
1	01	0000001	SOH (start of heading)
2	02	0000010	STX (start of text)
3	03	0000011	ETX (end of text)
4	04	0000100	EOT (end of transmission)
5	05	0000101	ENQ (enquiry)
6	06	0000110	ACK (acknowledge)
7	07	0000111	BEL (bell)
8	08	0001000	BS (backspace)
9	09	0001001	TAB (horizontal tab)
10	0A	0001010	LF (line feed)
11	0B	0001011	VT (vertical tab)
12	0C	0001100	FF (form feed)
13	0D	0001101	CR (carriage return)
14	0E	0001110	SO (shift out)
15	0F	0001111	SI (shift in)
16	10	0010000	DLE (data link escape)
17	11	0010001	DC1 (device control 1)
18	12	0010010	DC2 (device control 2)
19	13	0010011	DC3 (device control 3)
20	14	0010100	DC4 (device control 4)
21	15	0010101	NAK (negative ack.)
22	16	0010110	SYN (synchronous idle)
23	17	0010111	ETB (end of trans. block)
24	18	0011000	CAN (cancel)
25	19	0011001	EM (end of medium)
26	1A	0011010	SUB (substitute)
27	1B	0011011	ESC (escape)
28	1C	0011100	FS (file separator)
29	1D	0011101	GS (group separator)
30	1E	0011110	RS (record separator)
31	1F	0011111	US (unit separator)

32 – 63 Razmak i interpunkcija			
Dec	Hex	Bin	Znak
32	20	0100000	(space)
33	21	0100001	!
34	22	0100010	"
35	23	0100011	#
36	24	0100100	$
37	25	0100101	%
38	26	0100110	&
39	27	0100111	'
40	28	0101000	(
41	29	0101001	)
42	2A	0101010	*
43	2B	0101011	+
44	2C	0101100	,
45	2D	0101101	-
46	2E	0101110	.
47	2F	0101111	/
48	30	0110000	0
49	31	0110001	1
50	32	0110010	2
51	33	0110011	3
52	34	0110100	4
53	35	0110101	5
54	36	0110110	6
55	37	0110111	7
56	38	0111000	8
57	39	0111001	9
58	3A	0111010	:
59	3B	0111011	;
60	3C	0111100	<
61	3D	0111101	=
62	3E	0111110	>
63	3F	0111111	?

64 – 95 Velika slova i simboli			
Dec	Hex	Bin	Znak
64	40	1000000	@
65	41	1000001	A
66	42	1000010	B
67	43	1000011	C
68	44	1000100	D
69	45	1000101	E
70	46	1000110	F
71	47	1000111	G
72	48	1001000	H
73	49	1001001	I
74	4A	1001010	J
75	4B	1001011	K
76	4C	1001100	L
77	4D	1001101	M
78	4E	1001110	N
79	4F	1001111	O
80	50	1010000	P
81	51	1010001	Q
82	52	1010010	R
83	53	1010011	S
84	54	1010100	T
85	55	1010101	U
86	56	1010110	V
87	57	1010111	W
88	58	1011000	X
89	59	1011001	Y
90	5A	1011010	Z
91	5B	1011011	[
92	5C	1011100	\
93	5D	1011101	]
94	5E	1011110	^
95	5F	1011111	_

96 – 127 Mala slova i simboli			
Dec	Hex	Bin	Znak
96	60	1100000	`
97	61	1100001	a
98	62	1100010	b
99	63	1100011	c
100	64	1100100	d
101	65	1100101	e
102	66	1100110	f
103	67	1100111	g
104	68	1101000	h
105	69	1101001	i
106	6A	1101010	j
107	6B	1101011	k
108	6C	1101100	l
109	6D	1101101	m
110	6E	1101110	n
111	6F	1101111	o
112	70	1110000	p
113	71	1110001	q
114	72	1110010	r
115	73	1110011	s
116	74	1110100	t
117	75	1110101	u
118	76	1110110	v
119	77	1110111	w
120	78	1111000	x
121	79	1111001	y
122	7A	1111010	z
123	7B	1111011	{
124	7C	1111100	|
125	7D	1111101	}
126	7E	1111110	~
127	7F	1111111	DEL (delete)

Unicode kodiranje znakova

- **Unicode** je standard za **jedinstveno predstavljanje znakova različitih jezika i pisama**
- Razvijen je da omogući predstavljanje znakova koji nisu obuhvaćeni **osnovnim ASCII skupom**
- Omogućava predstavljanje znakova različitih pisama, kao što su **latinica, ćirilica, grčko, arapsko, kinesko i druga pisma**
- Svakom znaku dodeljuje se **jedinstvena kodna vrednost**
- Na taj način isti znak ima **jednoznačno definisano predstavljanje** u različitim računarskim sistemima i programima
- Unicode predstavlja osnovu za **savremeno predstavljanje teksta u računarima**

Načini kodiranja Unicode znakova

- Unicode definiše **skup znakova i njihove kodne vrednosti**, dok način kodiranja određuje **kako se te vrednosti zapisuju u memoriji**
- Za zapis Unicode znakova koriste se različiti načini kodiranja, među kojima su **UTF-8, UTF-16 i UTF-32**
- **UTF-8** koristi promenljiv broj bajtova za jedan znak: **1–4 bajta**
- **UTF-16** koristi **2 ili 4 bajta** za jedan znak
- **UTF-32** koristi **4 bajta** za svaki znak
- Različiti načini kodiranja omogućavaju različit odnos između **zauzeća memorije i kompatibilnosti sa postojećim sistemima**
- **UTF-8** je danas posebno široko korišćen za razmenu i čuvanje tekstualnih podataka

Unicode znak u različitim načinima kodiranja

- U ASCII tabeli znak **A** ima dekadnu vrednost **65**
- Isti znak u Unicode standardu ima kodnu vrednost **U+0041**
- Vrednosti $\mathbf{65_{10}}$ i **U+0041** predstavljaju **isti znak – A**, ali su zapisane u različitim kontekstima
- Unicode kodna vrednost **U+0041** odgovara heksadecimalnoj vrednosti $\mathbf{41_{16}}$
- Kodiranje znaka zatim određuje kako će se njegova Unicode vrednost zapisati u **bajtovima**
- Unicode kodna vrednost ne određuje broj bajtova koji će znak zauzeti u memoriji

Unicode znakovi srpske abecede

Br.	Slovo	Veliko slovo (Unicode)	Malo slovo (Unicode)	Br.	Slovo	Veliko slovo (Unicode)	Malo slovo (Unicode)
1	**A a**	U+0041	U+0061	16	**N n**	U+004E	U+006E
2	**B b**	U+0042	U+0062	17	**Nj nj**	U+004E + U+006A	U+006E + U+006A
3	**V v**	U+0056	U+0076	18	**O o**	U+004F	U+006F
4	**G g**	U+0047	U+0067	19	**P p**	U+0050	U+0070
5	**D d**	U+0044	U+0064	20	**R r**	U+0052	U+0072
6	**Đ đ**	U+0110	U+0111	21	**S s**	U+0053	U+0073
7	**E e**	U+0045	U+0065	22	**T t**	U+0054	U+0074
8	**Ž ž**	U+017D	U+017E	23	**Ć ć**	U+0106	U+0107
9	**Z z**	U+005A	U+007A	24	**U u**	U+0055	U+0075
10	**I i**	U+0049	U+0069	25	**F f**	U+0046	U+0066
11	**J j**	U+004A	U+006A	26	**H h**	U+0048	U+0068
12	**K k**	U+004B	U+006B	27	**C c**	U+0043	U+0063
13	**L l**	U+004C	U+006C	28	**Č č**	U+010C	U+010D
14	**Lj lj**	U+004C + U+006A	U+006C + U+006A	29	**Š š**	U+0160	U+0161
15	**M m**	U+004D	U+006D	30	**Ž ž**	U+017D	U+017E

Izbor načina kodiranja

- **Unicode** određuje skup znakova i njihove **kodne vrednosti**, dok kodiranje **UTF-8, UTF-16 i UTF-32** određuju način njihovog zapisivanja u bajtovima
- Način kodiranja određuje **program, format datoteke ili komunikacioni protokol**, a ne sam Unicode znak
- Informacija o primenjenom kodiranju može biti navedena u **zaglavlju, metapodacima ili specifikaciji formata**
- Na primer, kada je dokument definisan kao **UTF-8**, program koji ga otvara zna kako treba da protumači zapisane bajtove
- Ako program koristi **pogrešno kodiranje**, isti bajtovi mogu biti protumačeni kao drugi znakovi i tekst će biti pogrešno prikazan
- Zato učesnici u razmeni podataka moraju znati **koje je kodiranje primenjeno**
- UTF-8 je danas najčešće korišćeni način kodiranja Unicode teksta, posebno na webu i pri razmeni tekstualnih podataka

Primer: uticaj kodiranja na ramenu podataka

- **Pošiljalac** šalje znak **A**
- Znak **A** ima Unicode kodnu vrednost **U+0041**
- Izabrano kodiranje pretvara ovu vrednost u niz **bajtova**
- **UTF-8:** 41 → **1 bajt**
- **UTF-16:** 00 41 → **2 bajta**
- **UTF-32:** 00 00 00 41 → **4 bajta**
- **Primalac** mora znati koje je kodiranje primenjeno da bi primljene bajtove pravilno protumačio

Napomena: Dve heksadecimalne cifre predstavljaju jedan bajt, jer svaka heksadecimalna cifra predstavlja 4 bita, pa dve cifre daju 8 bitova = 1 bajt

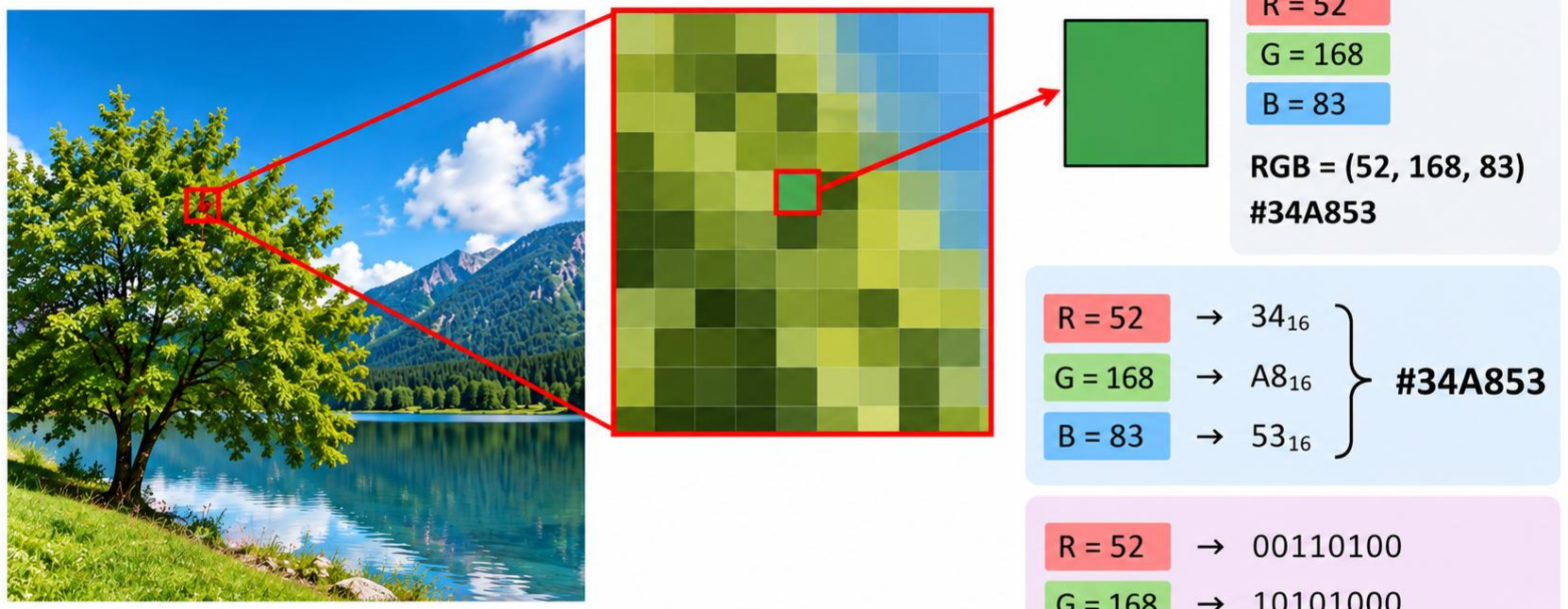
Predstavljanje različitih vrsta podataka

- **Svi podaci u računaru** na najnižem nivou predstavljaju se kao nizovi **binarnih cifara 0 i 1**
- Način tumačenja tih binarnih podataka zavisi od **vrste podatka i pravila njegovog predstavljanja**
- **Brojevi** se predstavljaju binarnim zapisom odgovarajuće vrednosti
- **Tekst i znakovi** predstavljaju se pomoću kodnih sistema kao što su **ASCII i Unicode**
- **Slike** se predstavljaju pomoću podataka koji opisuju njihove **piksele, boju i druge osobine**
- **Zvuk** se predstavlja nizom numeričkih vrednosti dobijenih **digitalizacijom zvučnog signala**
- Iako se način predstavljanja razlikuje, računar sve ove podatke na svom nivou obrađuje kao **binarne podatke**

Predstavljanje slike u računaru

- **Digitalna slika** sastoji se od velikog broja malih elemenata koji se nazivaju **pikseli**
- Svaki piksel ima određenu **boju**, koja se predstavlja numeričkim vrednostima
- Kod slike u boji, boja piksela najčešće se određuje vrednostima za **crvenu, zelenu i plavu komponentu (RGB)**
- Svaka od tih vrednosti predstavlja se određenim brojem **bitova**
- Kombinovanjem vrednosti RGB komponenti dobija se konačna **boja svakog piksela**
- Skup svih piksela i njihovih vrednosti čini **digitalni zapis slike**
- Što je veći broj piksela i što je veći broj bitova kojim se opisuju njihove boje, to se može predstaviti **veći nivo detalja i veći broj različitih boja**

Predstavljanje slike u računaru - primer



R = 52
G = 168
B = 83

RGB = (52, 168, 83)
#34A853

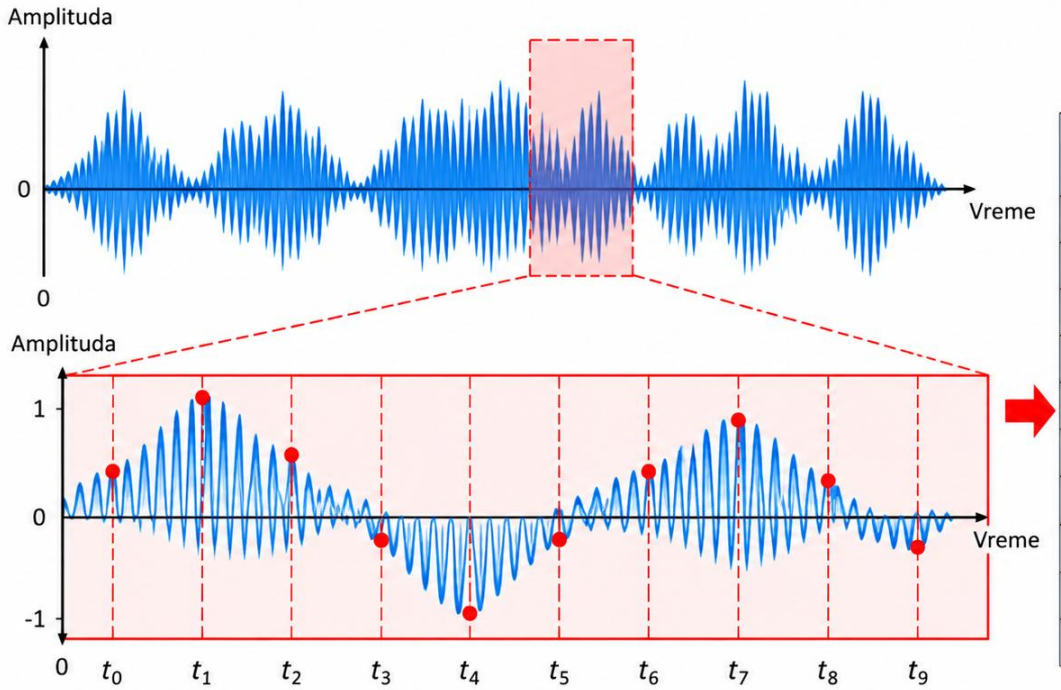
R = 52 → 34_{16}
G = 168 → $A8_{16}$
B = 83 → 53_{16}
} **#34A853**

R = 52 → 00110100
G = 168 → 10101000
B = 83 → 01010011

Predstavljanje zvuka u računaru

- **Zvuk** se u računaru predstavlja kao niz **numeričkih vrednosti**
- Te vrednosti predstavljaju karakteristike zvučnog signala izmerene u određenim vremenskim trenucima
- Svaka vrednost se zapisuje pomoću određenog broja **bitova**
- Tako se kontinualni zvučni signal pretvara u **digitalni zapis**
- Digitalni zapis zvuka računar čuva i obrađuje kao **binarne podatke**

Predstavljanje zvuka u računaru



Uzorkovane vrednosti	
t_0	34
t_1	98
t_2	120
t_3	76
t_4	-20
t_5	-96
t_6	-118
t_7	-72
t_8	10
t_9	56

Binarni zapis (8 bitova)	
t_0	00100010
t_1	01100010
t_2	01111000
t_3	01001100
t_4	11101100
t_5	10010000
t_6	10001010
t_7	10111000
t_8	00001010
t_9	00111000

Pitanje 1

Dekadni brojni sistem za osnovu ima broj:

a) 2
b) 8
c) 10
d) 16

Odgovor: c

Pitanje 2

Koliko različitih vrednosti može da se predstavi pomoću 8 bitova?

a) 64
b) 128
c) 256
d) 512

Odgovor: c

Pitanje 3

Koji standard omogućava predstavljanje velikog broja znakova različitih jezika i pisama?

a) ASCII
b) Unicode
c) EBCDIC
d) Binarni brojni sistem

Odgovor: b

Pitanje 4

Koji brojni sistem se koristi za predstavljanje podataka u računaru?

a) Dekadni
b) Binarni
c) Oktalni
d) Heksadecimalni

Odgovor: b

Pitanje 4

Koji je ispravan binarni zapis dekadnog broja 53?

a) 101101_2
b) 110011_2
c) 110101_2
d) 111001_2

Odgovor: c

5 4 3 2 1 0

$$1\ 1\ 0\ 1\ 0\ 1_2 = 1\cdot 2^5 + 1\cdot 2^4 + 0\cdot 2^3 + 1\cdot 2^2 + 0\cdot 2^1 + 1\cdot 2^0$$
$$= 1\cdot 32 + 1\cdot 16 + 0\cdot 8 + 1\cdot 4 + 0\cdot 2 + 1\cdot 1$$
$$= 32 + 16 + 0 + 4 + 0 + 1 = 53_{10}$$

Deljenje sa 2	Količnik	Ostatak
53 ÷ 2	26	**1**
26 ÷ 2	13	**0**
13 ÷ 2	6	**1**
6 ÷ 2	3	**0**
3 ÷ 2	1	**1**
1 ÷ 2	0	**1**

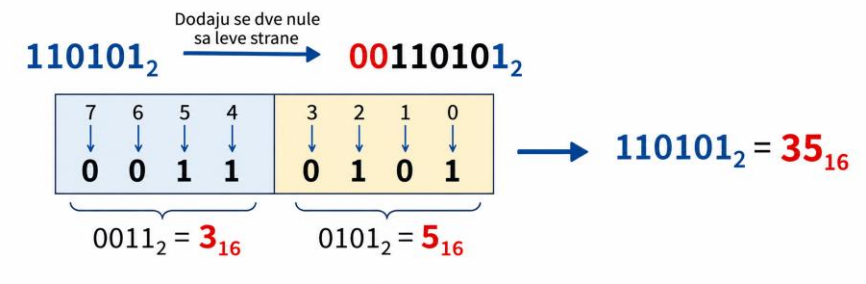
Ostatke čitamo od dole nagore

Pitanje 5

Koji heksadecimalni zapis odgovara binarnom broju 110101?

a) 35_{16}
b) $2D_{16}$
c) $3B_{16}$

Odgovor: a



Pitanje 6

Koliko bajtova može UTF-8 da koristi za zapis jednog Unicode znaka?

a) 1 do 4 bajta
b) Uvek 2 bajta
c) Uvek 4 bajta

Odgovor: a

Pitanje 7

Koja je razlika između ASCII i Unicode standarda za kodiranje znakova?

a) **ASCII** je standard za kodiranje 128 znakova, dok **Unicode** omogućava kodiranje znakova velikog broja jezika i pisama
b) **ASCII** i **Unicode** kodiraju isti skup znakova, ali koriste različite brojne sisteme
c) **Unicode** je ograničen na 128 znakova, dok **ASCII** omogućava kodiranje znakova svih jezika

Odgovor: a

Pitanje 8

Koji binarni zapis odgovara heksadecimalnom broju $3A_{16}$?

a) $0011\ 1001_2$
b) $0011\ 1010_2$
c) $0010\ 1010_2$

Odgovor: b $3A_{16} \rightarrow 0011 \mid 1010 \rightarrow 0011\ 1010_2$