

# PREDSTAVITEV INFORMACIJE

## Predstavitev informacij

V računalniku za predstavitev informacij ponavadi uporabljamo **dvojiški številski sistem** (0 in 1). S **šestnajstiškim številskim sistemom** (0 do F) krajše zapišemo (velika) dvojiška števila, **desetiški sistem** pa je najboljši za razumevanje pomena številke človeku.

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

**Računalniki za predstavitev števil uporabljajo samo dvojiški sistem, ne morejo jih neposredno predstaviti v desetiškem ali šestnajstiškem sistemu!**

**Števila** so lahko:

- **znaki za zapis,**
- **št. za računanje:** različni št. sistemi, različen zapis racionalnih št. (fiksna vejica, plavajoča vejica)

128	64	32	16	8	4	2	1
1	0	0	0	0	1	1	0
134 - 128 = 6							
6 - 4 = 2							
2 - 2 = 0							

**Negativna števila** lahko zapišemo:

- s **prvim bitom za predznak**: 0 predstavlja pozitiven predznak, 1 pa negativnega ( $00101001_{(2)} = 41_{(10)}$ ,  $10101001_{(2)} = -41_{(10)}$ ), problem se pojavi pri ničli ( $10000000_{(2)} = 0_{(10)}$ ,  $00000000_{(2)} = -0_{(10)}$ ),
- z **dvojiškim komplementom**:
  - 1) za **pozitivna št.**: ni spremembe – zapis s prvim bitom za predznak (0);
  - 2) za **negativna št.**: št. pretvorimo v dvojiško (brez predznaka), obrnemo številke ( $0 \rightarrow 1$ ,  $1 \rightarrow 0$ ) in obrnjenemu številu prištejemo dvojiško 1;
  - 3) za **pretvarjanje nazaj**: vsa neg. št. se začnejo z 1, vsa poz. pa z 0; uporabimo nasprotni postopek → odštejemo dvojiško 1, obrnemo številke, pretvorimo v desetiško ter dodamo predznak minus.

Če shranjujemo števila z 8 biti, lahko brez dvojiškega komplementa predstavimo št. od 0 do 255 v desetiškem sistemu, z dvojiškim komplementom pa št. od -128 do 127 v desetiškem sistemu.

Kako **seštevamo** dvojiška števila: [\[link\]](#)

- **pozitivna št.** enako kot po stolpcih, pazimo le pri  $1 + 1 = 10$  (0, prenos 1 naprej) in  $1 + 1 + 1 = 11$  (1, prenos 1 naprej)
- **neg. št. z bitom za predznak**: ne deluje!
- **neg. št. z dvojiškim komplementom**: če je potrebno, negativno št. pretvorimo v dvojiški komplement; če dobimo odvečni bit, ga ignoriramo

Pri zapisu realnih št. s **fiksno vejico** se dogovorimo, koliko bitov predstavlja celi del in koliko bitov predstavlja decimalni del števila. Postopek pretvorbe iz desetiškega v dvojiški sistem:

Oglejmo si primer, pri katerem želimo zapisati število 118,45 s 16 biti tako, da bo 8 bitov predstavljal celi del, 8 bitov pa decimalni del števila. Število pretvorimo tako, da ločeno pretvorimo celi del števila, levo od decimalne vejice in ločeno decimalni del, desno od decimalne vejice. Za pretvarjanje lahko uporabimo Hornerjev algoritem.

Pretvorimo celi del.

```
118 = 59 × 2 + 0
59 = 29 × 2 + 1
29 = 14 × 2 + 1
14 = 7 × 2 + 0
7 = 3 × 2 + 1
3 = 1 × 2 + 1
1 = 0 × 2 + 1 ↑
```

Števke preberemo od spodaj navzgor in dobimo  $118_{10} = 1110110_2$ . Ker zahtevamo zapis celega dela z 8 biti, ta del zapišemo kot 01110110<sub>2</sub>.

Pretvorimo decimalni del.

```
0,45 × 2 = 0,90 → 0 ↓
0,90 × 2 = 1,80 → 1
0,80 × 2 = 1,60 → 1
0,60 × 2 = 1,20 → 1
0,20 × 2 = 0,40 → 0
0,40 × 2 = 0,80 → 0
0,80 × 2 = 1,60 → 1
0,60 × 2 = 1,20 → 1
```

Vsakič upoštevamo le celi del, neceli del vzamemo naprej v postopek. Števke preberemo od zgoraj navzdol in dobimo  $0,45_{10} = 0,01110011_2$ .

Realno število ima v dvojiški predstavitvi s stalno vejico, naslednjo obliko

$$118,45_{10} = 1110110,01110011_2$$

Negativna realna števila predstavimo na enak način kot pozitivna, le da jih na koncu pretvorimo še v dvojiški komplement.

Pri zapisu realnih št. s **plavajočo vejico** števila predstavimo z eksponentnim zapisom. Tak zapis omogoča zapis zelo velikih in zelo majhnih št. z enakim št. bitov, saj mesto vejice navidezno pomikamo s spreminjanjem eksponenta. Zapis izgleda tako:

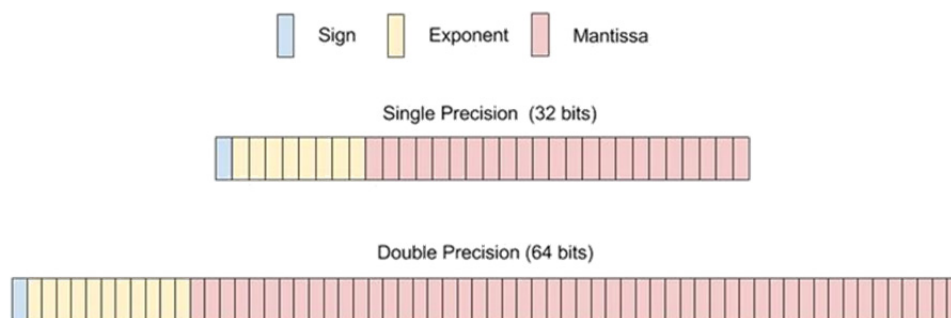
$$m * B^n$$

$m$  = mantisa,  $B$  = osnova,  $n$  = eksponent

Realna števila tako zapišemo s katerokoli osnovo  $B$ , mantisa ima vejico vedno na določenem mestu, spreminja pa se eksponent. V **normalizirani obliki** imamo eno celo mesto, druga pa so namenjena za decimalni del. V vseh drugih primerih govorimo o **denormalizirani obliki**.

### STANDARD IEEE 754

S standardom IEEE 754 najpogosteje zapisujemo decimalna števila z enojno natančnostjo (*single precision*) ali dvojno natančnostjo (*double precision*).



**Enojna natančnost** (32 bitov): 1 bit predznak, 8 bitov eksponent, 23 bitov mantisa

**Dvojna natančnost** (64 bitov): 1 bit predznak, 11 bitov eksponent, 52 bitov mantisa

$p$	$E = n - 127$	$m$
-----	---------------	-----

Eksponent je zapisan z **odmikom 127**. Računalnik eksponent obravnava, ko od njega odšteje 127. Eksponent 0 tako zapišemo kot 127 (dvojiško 01111111). Tako lahko zapišemo eksponente od -127 do +128. Mantiso vedno najprej **normaliziramo**, da dobimo zaporedje bitov **1, mmm...m** – prvi bit je zato vedno enak 1 in ni zajet v zapisani mantisi.

### Zvrsti MIME

Zvrsti datotek na omrežju so določene po **dogovoru MIME** (*Multipurpose Internet Mail Extensions*). Ko strežnik odpošlje datoteko, zapiše na njeno »ovojnico« podatek o njeni zvrsti. Tega običajno določi na osnovi podaljška imena datoteke (*file extension*). Osnovne zvrsti po dogovoru MIME so:

<b>text</b>	znakovna datoteka	<b>audio</b>	datoteka z zvokom
<b>message</b>	datoteka s sporočilom	<b>application</b>	datoteka s podatki za program ali dvojiška datoteka
<b>image</b>	datoteka s sliko	<b>multipart</b>	datoteka z združenimi več raznovrstnimi datotekami
<b>video</b>	datoteka z videom	<b>model</b>	datoteka z opisom modela

**Standardizacija** omogoča usklajenost izdelkov različnih proizvajalcev.

### ZGOŠČEVANJE PODATKOV

Z zgoščevanjem podatkov zmanjšamo prostor, ki ga zavzamejo datoteke in povečamo hitrost prenosa podatkov. Osnovna principa zgoščevanja sta:

- **brezizgubno zgoščevanje**: formati png, flac, zip itd.
- **izgubno zgoščevanje**: formati jpeg, mp3, mp4 itd.

Glavna ideja brezizgubnega zgoščevanja je, da so **določeni znaki bolj verjetni od drugih**. Za bolj verjetne uporabimo **krajšo** kodno zamenjavo, za manj verjetne pa **daljšo** kodno zamenjavo. Nobena kodna zamenjava pa ne sme biti predpona druge kodne zamenjave. V povprečju je tako celoten zapis krajši. Dva glavna algoritma/postopka:

- **Huffmanov algoritem**: večkrat ponavljajoče se nize zapišemo z manj biti kot manjkrat ponavljajoče

HUFFMANOV ALGORITEM   Niz: <i>Gori na gori gori.</i>									
Zapišemo št. ponovitev vsakega znaka, pogostejšim določimo krajšo kodo.									
št.:	3	3	3	3	2	1	1	1	1
znak:	_	o	r	i	g	a	G	n	.
zapis:	0	1	00	01	10	11	000	001	010
Zgoščen zapis: <b>33 bitov</b>									
Nezgoščen zapis: <b>54 bitov</b>									
(za nezgoščen 18 znakov x 3 biti)									

ZIP je izboljšan Huffmanov algoritem: poleg znakov upošteva tudi ponavljanje znakov v njihovem zapisu. Drugi algoritem je:

➤ **Shannon-Fanojev algoritem:** vsakemu znaku določi različno dolg niz sestavljen iz 0 in 1; postopek je tak (dan imamo niz znakov):

1. vsakemu znaku določimo frekvenco (kolikokrat se pojavi),
2. rezultate razporedimo v seznam tako, da so najpogostejši znaki na vrhu,
3. seznam razdelimo v dva dela tako, da je seštevek ponavljanj znakov v zgornjem delu čim bližje seštevku v spodnjem delu,
4. zgoščenemu zapisu znakov v zgornjem delu pripišemo 0, spodnjim pa 1,
5. razdeljevanje seznama rekurzivno ponavljamo, dokler ne pridemo do posameznih znakov.

SHANNON-FANOJEV ALGORITEM   Niz: CABADBA	
A: 3x, B: 2x, C: 1x, D: 1x	
A 3x → 0	A = 0
B 2x → 1 → 0	B = 10
C 1x → 1 → 1 → 0	C = 110
D 1x → 1 → 1 → 1	D = 111
Nezgoščen zapis po ASCII: 7 znakov * 7 bitov = 56 bitov	
Zgoščen zapis: 3 * 1b + 2 * 2b + 1 * 3b + 1 * 3b = 13 bitov	
Faktor zgoščevanja: 56 / 13 = 4,3	

**Faktor zgoščevanja** izračunamo tako, da nezgoščeno velikost delimo z zgoščeno.

## Pisna predstavitev informacije

**Kodne tabele** določajo kodo posameznega znaka – vsak znak ima svojo kodo. Primeri:

- **ASCII:** v originalu 7-bitna, zdaj razširjena v 8 bitov;
- **UNICODE:** več različic, npr. UTF-16 je 16-bitna, UTF-8 je 8-bitna;
- **Code page:** Microsoftova razširitev ASCII, 8-bitna, npr. Windows-1250;
- **LATIN:** mednarodni standardi ISO 8859, Latin-2 vsebuje tudi slovenske znake.

0	<NUL>	32	<SPC>	64	@	96	`	128	À	160	†	192	¿	224	+
1	<SOH>	33	!	65	A	97	a	129	Á	161	°	193	¡	225	·
2	<STX>	34	"	66	B	98	b	130	Â	162	¢	194	ª	226	¸
3	<ETX>	35	#	67	C	99	c	131	Ã	163	£	195	»	227	ˆ
4	<EOT>	36	\$	68	D	100	d	132	Ä	164	§	196	¼	228	‰
5	<ENQ>	37	%	69	E	101	e	133	Å	165	•	197	½	229	ˆ
6	<ACK>	38	&	70	F	102	f	134	Ä	166	¶	198	¾	230	È
7	<BEL>	39	'	71	G	103	g	135	Å	167	ß	199	»	231	Å
8	<BS>	40	(	72	H	104	h	136	À	168	©	200	™	232	Ê
9	<TAB>	41	)	73	I	105	i	137	Á	169	®	201	™	233	Ë
10	<LF>	42	*	74	J	106	j	138	Â	170	™	202	™	234	Ì
11	<VT>	43	+	75	K	107	k	139	Ã	171	™	203	À	235	Í
12	<FF>	44	,	76	L	108	l	140	Ä	172	™	204	Á	236	Î
13	<CR>	45	-	77	M	109	m	141	Å	173	™	205	Â	237	Ï
14	<SO>	46	.	78	N	110	n	142	Ä	174	Æ	206	Ã	238	Ï
15	<SI>	47	/	79	O	111	o	143	È	175	Ø	207	Ä	239	Ö
16	<DL>	48	0	80	P	112	p	144	É	176	œ	208	Å	240	×
17	<DC1>	49	1	81	Q	113	q	145	Ê	177	±	209	–	241	Ö
18	<DC2>	50	2	82	R	114	r	146	Ë	178	≤	210	™	242	Ù
19	<DC3>	51	3	83	S	115	s	147	Ì	179	≥	211	™	243	Ú
20	<DC4>	52	4	84	T	116	t	148	Í	180	¥	212	™	244	Û
21	<NAK>	53	5	85	U	117	u	149	Î	181	µ	213	™	245	Ü
22	<SYN>	54	6	86	V	118	v	150	Ï	182	®	214	™	246	Ý
23	<ETB>	55	7	87	W	119	w	151	Ö	183	Σ	215	™	247	~
24	<CAN>	56	8	88	X	120	x	152	×	184	Π	216	™	248	™
25	<EM>	57	9	89	Y	121	y	153	×	185	π	217	™	249	™
26	<SUB>	58	:	90	Z	122	z	154	×	186	ƒ	218	™	250	™
27	<ESC>	59	;	91	[	123	{	155	×	187	ª	219	™	251	™
28	<FS>	60	<	92	\	124		156	×	188	º	220	™	252	™
29	<GS>	61	=	93	]	125	}	157	×	189	Ω	221	™	253	™
30	<RS>	62	>	94	^	126	~	158	×	190	æ	222	™	254	™
31	<US>	63	?	95	_	127	<DEL>	159	×	191	ø	223	™	255	™

Zapis znakov v kodnih tabelah je **standardiziran, urejen**, vsi znaki pa nimajo nujno enake dolžine – pogostejši znaki imajo krajšo dolžino (npr. znaki v kodni tabeli UTF-8 lahko obsegajo od 8 do 32 bitov).

← celotna kodna tabela ASCII

## Slikovna predstavitev informacije

V računalniku je grafika shranjena na dva bistveno različna načina. Kot **rastrska grafika (posnetek)**, kjer je bitna slika sestavljena iz milijonov pikslov, ali kot **vektorska grafika (opis)**, ki je sestavljena iz osnovnih geometrijskih elementov.

### Lastnosti rastrske grafike:

- **ločljivost:** 1920 x 1080, 6000 x 4000 ...
- **barvna globina:** s koliko biti zapišemo barvo enega piksla; 1-bitna (monokromatske slike), 8-bitna (256 barv), 24-bitna (RGB – 16,7 milijona barv).

**Barva** lahko nastane na dva načina:

- s **snovnim (substrahivnim)** mešanjem: odboj bele svetlobe od predmeta (CMYK),
- z **optičnim (aditivnim)** mešanjem: seštevanje različnih EMV frekvenc (RGB).

**Monitorji** za prikaz barv uporabljajo optično (aditivno) mešanje, torej uporabljajo barvni model RGB. Druge lastnosti prikaza slik na zaslonu so:

- **velikost zaslona:** v pikslih (npr. 1920 x 1080) in diagonalni (npr. 24"),
- **frekvenca osveževanja:** ponavadi 60 Hz,
- **ppi (pixels per inch):** koliko pikslov lahko monitor naslika v dolžini enega inča, npr. 72 dpi, 96 dpi, 120 dpi itd.; pri tiskanju se uporablja izraz **dpi (dots per inch)**.

**Barvni model RGB** pozna več standardov za barvno globino: VGA (4 biti), True Color (24 bitov), 32 bitov (24 bitov za barvo + 8 bitov za alfa kanal) itd. Slike lahko med različnimi barvnimi globinami pretvarjamo s tehniko **raztrosa (dithering)**: vzorci s spremenljivo gostoto pik različnih barv ustvarijo učinek ustrezne barve.

**Tiskalniki** za prikaz/nanašanje barv na papir uporabljajo snovno (substrahivno) mešanje, torej uporabljajo barvni model CMYK (cyan, magenta, yellow, black oz. key).

### STANDARDNI SLIKOVNI FORMATI

- **rastrski formati:** PNG, JPEG, GIF, BMP, TIFF ...
- **vektorski formati:** SVG, CDR (CorelDraw), DWG (AutoCad) ...

Prednosti in slabosti obeh grafičnih načinov:

BITNI	VEKTORSKI
rišemo podobno kot ročno	sliko sestavljamo iz geometrijskih objektov
barvo posameznih pikslov lahko spreminjamo	barvo spreminjamo celemu objektu
zbrišemo le del slike	zbrišemo le cel objekt
izguba kvalitete pri spreminjanju velikosti	velikost spreminjamo brez izgube kvalitete

### Glavne lastnosti formata GIF

- 8-bitna barvna globina → 256 barv
- vsaka slika ima svojo **barvno palet**: če v sliki ni zelene, v paleti ni zelene
- **brezizgubno zgoščevanje** in **hitri predprikaz** (sliko naloži po plasteh)
- specifikacija GIF 89a omogoča **prozornost ene barve** in **animacije**

### Glavne lastnosti formata PNG

- nadomešča GIF, ima **brezizgubno zgoščevanje**
- omogoča **večbarvno prozornost** in **animacije** v formatu APNG

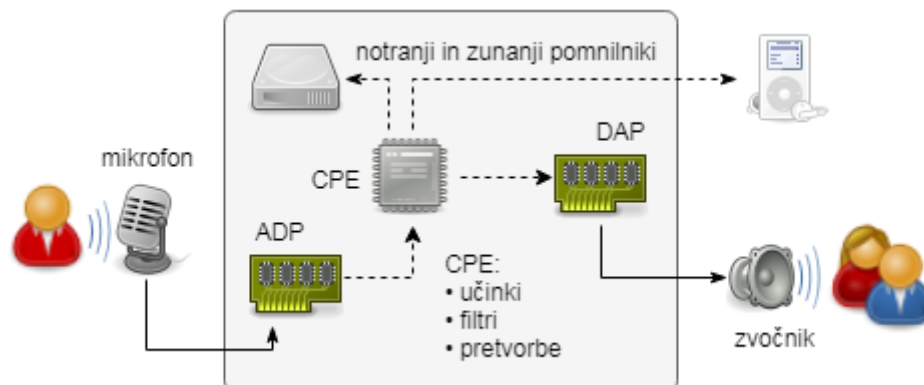
### Glavne lastnosti formata JPEG

- slike so razdeljene na kvadratke 8px \* 8px; v kvadratu je mreža RGB števil
- **izgubno zgoščevanje**: zapomni si samo nekaj št. iz mreže → izgube

## Zvočna predstavitev informacije

V računalniku je zvok shranjen na dva bistveno različna načina. Kot **posnetek zvočnega valovanja**, ki ga dobimo z vzorčenjem, ali kot **opis zvoka**, ki ga predvajamo s pomočjo vnaprej posnetih zvokov različnih glasbil (standard MIDI).

**Zvok** je valovanje, ki se prenaša skozi snovi; harmonično nihanje zvoka ustvari ton. Zvok opredelimo s **frekvenco [Hz]** (višina tona) in **amplitudo [dB]** (glasnost tona).



Če želimo zvok shraniti v digitalno obliko (datoteko), moramo neprekinjene analogne zvočne valove pretvoriti v digitalne dvojiške vrednosti.

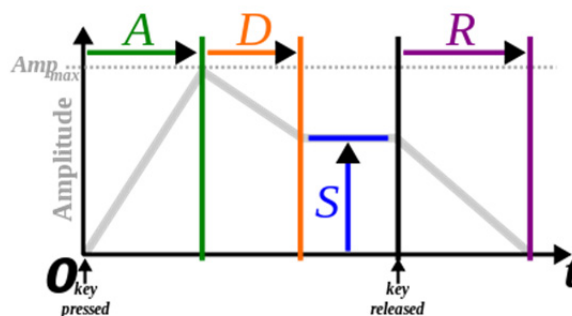
- **ADP (analogno-digitalni pretvornik)**: mikrofon zajame zvočni val, ki ga pretvori v električni signal, ADP pa ga pretvori v dvojiške vrednosti.
- **DAP (digitalno-analogni pretvornik)**: dvojiške vrednosti iz zvočne datoteke pretvori v električne signale za zvočnik, ki nato vibrira.

Pri **vzorčenju zvoka** v rednih in enakomernih časovnih presledkih odčitamo vrednost zvočnega vala. **Frekvenca vzorčenja** (*sampling rate*) je število prebranih vzorcev na sekundo in jo merimo v hertzih (Hz). Ponavadi je to 44,1 kHz. **Ločljivost vzorčenja** (*sampling resolution*) je število bitov, ki jih uporabimo za zapis vsakega vzorca. Ponavadi je 16-bitna.

**Nyquistov teorem** pravi, naj bo frekvenca vzorčenja dvakratnik najvišje frekvence, ki jo zajamemo. Ljudje slišimo do ~22 kHz, zato je frekvenca vzorčenja najpogostejše 44,1 kHz.

## ZAPISOVANJE ZVOKA V DATOTEKE

**Ovojnica** je krivulja, ki sledi spreminjanju amplitudi oz. jakosti zvoka (ADSR = *attack*, *decay*, *sustain*, *release* → vzpon, spust, trajanje, sprostitve). **Dinamika** je razmerje med najglasnejšim in najtišjim zvokom.



Nestisnjena velikost zvočne datoteke je

odvisna od **bitne hitrosti**, ki je odvisna od frekvence vzorčenja in ločljivosti vzorčenja.

$$\begin{aligned}\text{bitna hitrost} &= \text{frekvenca} * \text{ločljivost} \\ \text{nestisnjena velikost} &= \text{trajanje} * \text{bitna hitrost} * \text{št. kanalov}\end{aligned}$$

## STANDARDNI ZVOČNI FORMATI

- **WAV**: nezgoščen format, zapišemo lahko mono ali stereo zvok
- **FLAC**: brezizgubno zgoščevanje (išče ponavljajoče se vzorce, linearno napove...)
- **MP3**: izgubno zgoščevanje (zavrže človeku neslišne podatke + dodatna obdelava), prilagojen je lastnostim človeškega ušesa
- **WMA**: Microsoftov format, manjše datoteke pri enaki kvaliteti kot MP3
- **OGG Vorbis**: brezplačen format podoben WMA

## OPIS ZVOKA – MIDI *Musical Instrument Digital Interface*

**MIDI** je zapis, ki zvočni kartici pove, katere **zvočne vzorce** naj sproži in kako **dolgo** ter **glasno** naj se izvajajo. Standard General MIDI določa **128 tipk klavirja** in **128 drugih glasbil** oz. zvočil. Vokalov zaradi tega pri formatu MIDI ni! MIDI ukaz je sestavljen iz kontrolnega bajta in 2 ali 3 podatkovnih bajtov.

## Predstavitev informacije z gibljivo sliko

Predstavitev z gibljivo sliko lahko izvedemo z **videom**, **animacijo** ali **navidezno resničnostjo**.

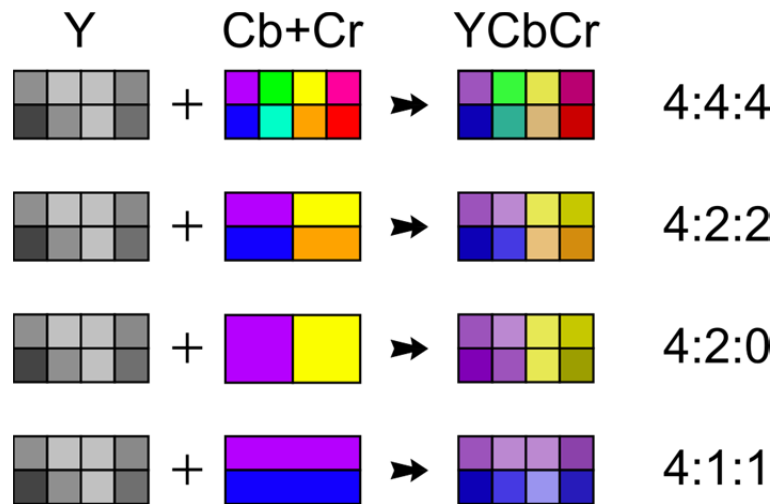
Pri **analogni televiziji** sta najpogostejše uporabljena standarda evropski **PAL** (Phase Alternation Line, 25 FPS) in ameriški **NTSC** (National TV System Committee, 30 FPS). Danes seveda uporabljamo **digitalno televizijo**, zato prevladujejo standardi kot so HD, Full HD, 4K UHD, ki delujejo na standardu prenosa DVB (Digital Video Broadcasting).

**Glavne lastnosti videa**, ki med drugim vplivajo na kvaliteto so:

- **št. slik na sekundo** (oz. **FPS**),
- **ločljivost**: FHD 1920 x 1080, UHD 3840 x 2160 itd.,
- **prepletanje** (oz. *interlacing*): izmenično prikazovanje sodih in lihih vrstic tako, da dobimo 50 polslik, s tem prihranimo prostor; najprej se prikažejo lihe, nato sode,
- **kompresijski algoritem**.



Za **zapis barv v videu** namesto modela RGB uporabljamo modela **YCbCr** in YUV. YUV se je uporabljal pri analognem videu, novejši YCbCr pa se uporablja pri digitalnem videu. Komponenta **Y** predstavlja **svetilnost**, komponenti **Cb** in **Cr** pa določata **barvo**.



Z barvnim **podvzorčenjem** (*subsampling*) zmanjšamo količino podatkov za zapis videa. Pri 4:4:4 ne gre za podvzorčenje, ker vse tri komponente vzorčimo enako pogosto. Najpogosteje uporabljeni sistemi podvzorčenja so:

- **4:2:2** → vzorec v 2 točki: Y v vsaki, Cb in Cr za vsako drugo
- **4:1:1** → vzorec v 4 točkah: Y v vsaki, Cb in Cr za vsako četrto (NTSC)
- **4:2:0** → vzorec v 2 točki in 2 vrsti: Y v vsaki, Cb za vsako drugo točko v lihih vrsticah, Cr pa za vsako drugo v sodih vrsticah (PAL), najpogostejši

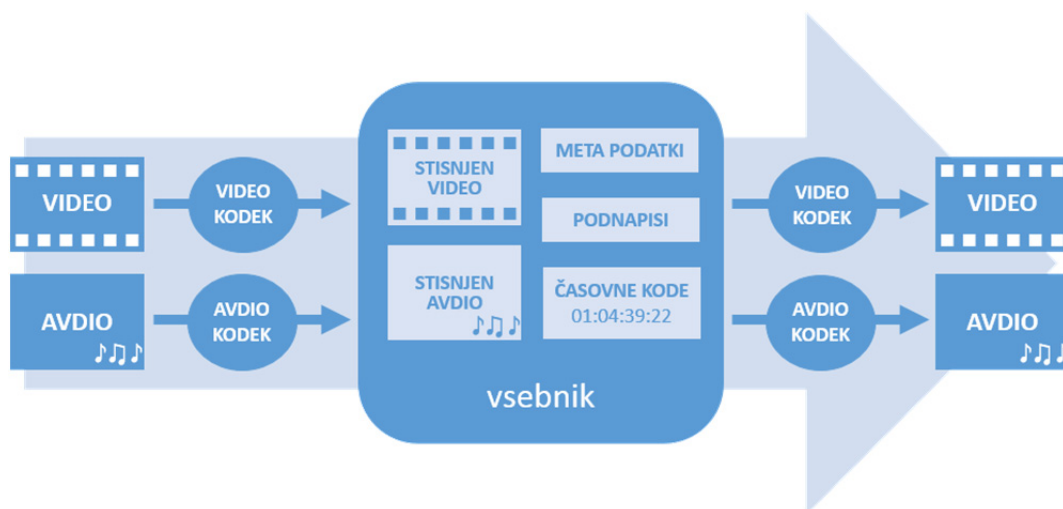
### **KOMPRESIJSKI (ZGOŠČEVALNI) ALGORITMI**

- **diferencialna modulacija**: primerja barvo piksla čez več framov in zapiše samo prvo in zadnjo, vmes se barva interpolira
- **diskretno preoblikovanje**: podobno kot pri JPEG, bloki 8 px \* 8 px
- **izravnava gibanja**: največkrat uporabljen algoritem; ne zapisujemo celotnih framov, ampak zapišemo celoten keyframe, med posameznimi keyframei pa samo spremembe glede na prejšnji frame videa

### **POSTOPKI ZGOŠČEVANJA**

- **MPEG1**: za kasete VHS ločljivosti 352 x 288
- **MPEG2**: za zgoščenke DVD
- **MPEG4**: za multimedijske datoteke za internet, prednosti sta ločevanje na predmete in izbira selektivnih metod zgoščevanja
- **DivX**: je brezplačen, predelan MPEG4





Urejanje in pakiranje podatkov opravimo s **kodeki** – to so postopki zgoščevanja oz. stiskanja (in razširjanja) video ali zvočnih podatkov. **Vsebnik** je datoteka, zasnovana kot vmesnik med OS in temi podatki. Vsebniki so npr. AVI, MKV (Matroska), MP4, OGV, kodeki pa npr. H.264/MPEG-4 AVC, H.265/HEVC, VP9, MPEG-2, MPEG-1 itd.

Učinek **animacije** dosežemo s hitrim spreminjanjem zaporednih slik, ki se med seboj minimalno razlikujejo. Pri tradicionalni animaciji so posamezne slike narisane ročno in nato fotografirane/skenirane (npr. Tom in Jerry), pri računalniški animaciji uporabljamo razne programe za animacijo, npr. Cinema 4D, Blender itd.

Pri **navidezni resničnosti** (*Virtual Reality* = VR) ima uporabnik občutek, da se nahaja v umetnem okolju. Deluje s pomočjo senzorjev in drugih naprav, za VR pa ponavadi rabimo zmogljivo (in drago) strojno opremo.

## Računalniške prosojnice

Z **računalniškimi prosojnicami** ponavadi ustvarjamo digitalne predstavitve za šolo, službo, predavanja ipd. Uporabljamo programe kot so PowerPoint, LibreOffice Impress itd. Z raznolikimi prosojnicami, dobrim kontrastom, uporabo slik, tabel, besedila in drugih stvari privabimo poslušalca/gledalca.

## Predstavitev informacije na svetovnem spletu

**Spletni naslov (URL)** je sestavljen iz treh glavnih delov: imena **protokola**, spletnega **naslova računalnika/strežnika** in **opisa vira** na strežniku (poti do vira na strežniku). Dodamo lahko tudi **vrata (port)**, če jih je potrebno specificirati (ponavadi so 80).

`http://www.zrss.si:8080/rai/ucni_nacrti/informatika.html`

Če hočemo, da vir (stran, slika, css datoteka) deluje neodvisno od tega, na katerem strežniku je, moramo uporabljati **relativno naslavljanje**. **Spletni sestavek** označuje fizično zgradbo spletne strani z mapami, podmapami, stranmi, slikami itd. (torej mapa z vsemi viri).