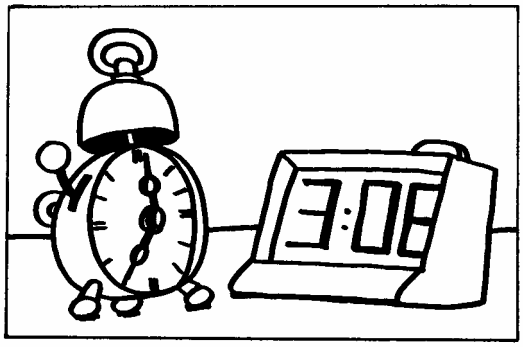
1.1 INFORMACIJE IN NJIHOVA PREDSTAVITEV

Ko želiš s svojimi prijatelji ali prijateljicami izmenjati informacijo, jo moraš seveda predstaviti z določenimi izraznimi sredstvi: s pisnimi (pošlješ običajno ali e-pismo), svetlobnimi (pošlješ SMS, nemi ljudje se pogovorijo s posebnimi dogovorjenimi znaki), z zvočnimi (se pogovoriš preko telefona ali pa kar direktno) itd. Ne glede na to, ali govorimo o predstavitvi podatkov v zvezi s človekom ali strojem, ločimo dva bistveno različna načina predstavitve:

- **analogni** ali zvezni način in
- **digitalni** ali diskretni način.

Za boljše razumevanje si oglejva dva primera (slika 1). Ura s kazalci je lep primer analognega (neprekinjenega) podajanja informacij, saj čas odčitavaš s pripadajočo lego urnih kazalcev. Drugačna je elektronska digitalna ura, pri kateri je čas povezan s štetjem oziroma stalnim spreminjanjem števil na zaslonu in je primer digitalnega (končnega – številnega števila stanj) podajanja informacije.



Slika 1: *Analogni in digitalni zapis časa*

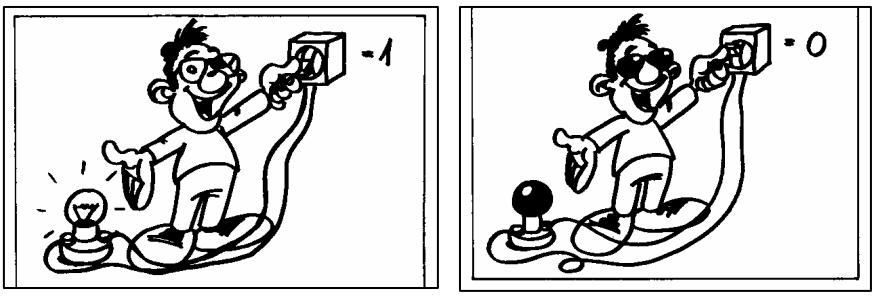
Drugi primer je povezan z avtomobilom. V avtomobilu nam običajno merilna ura hitrosti s kazalcem kaže analogno informacijo o hitrosti vožnje, medtem ko nam števec prevoženih kilometrov daje digitalno informacijo.

Informacije so prav tako kot vse ostale količine (npr. čas, dolžina, masa itd.) merljive; seveda je način njihovega merjenja dokaj zapleten, zato se na tem mestu ne bomo ukvarjali s podrobnostmi postopka. Spoznali bomo le osnovno enoto (in njene izpeljanke), ki se je izkazala za ustrezno, še posebno v informacijski tehnologiji. Enota, ki jo je stroka tega področja sprejela, je **tista informacija, s katero dobimo odgovor na vprašanje, na katero sta mogoča samo dva verjetna odgovora.** Ta enota se imenuje **bit** (ang. **bi**nary digi**t**) in je najmanjša enota za merjenje količine podatkov. Vsebuje lahko torej samo dve stanji, ki ju lahko izrazimo (zakodiramo) na več načinov: električni tok teče – tok ne teče, črno – belo, magnetizirano – nemagnetizirano stanje, svetlobni žarek se odbije ali ne. Ta stanja simbolično označimo z DA ali NE oziroma 0 ali 1.

Glede na ta dva načina predstavitve in obdelave informacij lahko v osnovi delimo računalnike na **digitalne** in **analogne**. Najbolj se uporabljajo digitalni računalniki. Sodobni računalniki te vrste

temeljijo na t.i. digitalni elektroniki dveh stanj 0 in 1; v matematiki imenujemo takšen številski sistem **dvojiški ali binarni**. Ko je angleški matematik George Boole (ustanovitelj matematične logike -

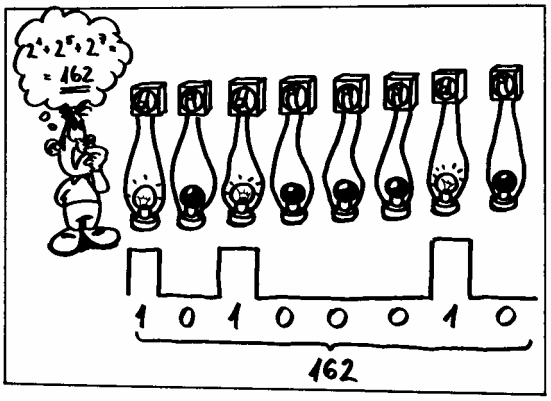
Boolove algebre), razvil algebro, s katero se razmerje med velikim številom elementov popiše s ponavljanjem operacij med vrsto parov elementov, je dal osnovo za razvoj mikroelektronike, saj simbola 0 in 1 nista omejena samo na števila. Če se uporabljata za "vključen" in "izključen", predstavljata dva položaja stikala (slika 2). Lahko pa predstavljata tudi druge alternative: DA, NE ali IN, ALI. Prav te so za delovanje računalnikov še posebej pomembne.



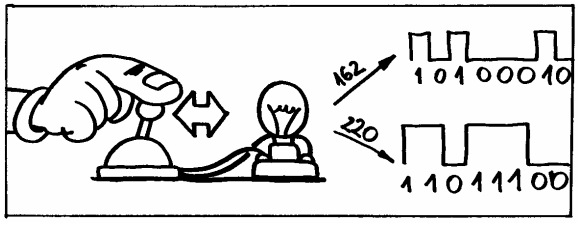
Slika 2: *Predstavitev osnovne informacije (0 in 1) s stikalom in žarnico*

Narava elektronskih komponent narekuje skromnost v zapisovanju simbolov, hkrati pa je tudi delovanje natančnejše in zanesljivejše; laže se odločamo le med dvema simboloma (ali stanjema) kot med več kot 50 simboli (ali stanji), ki jih uporabljamo pri zapisovanju črk, števil, ločil. Laže tudi ugotavljamo, ali po kakem vodniku teče električni tok oz. obstaja napetostno stanje (znak – stanje 1) ali ne (znak – stanje 0), kot pa, kolikšen je ta tok (npr. 0.883 A) oz. napetost (6.6 V). Znak 1 v dvojiškem sistemu računalnik razume kot prisotnost signala (pozitivne električne napetosti, npr. 5 V), znak 0 pa z njegovo odsotnostjo (napetost je 0 V).

Zapis npr. večjega števila si lahko predstavljamo kot zaporedje vklopljenih oziroma izklopljenih stikal in glede na to tudi ustreznih napetostnih impulzov (slika 3). Tako bi lahko z osmimi stikali zapisali desetiško število 162 (imamo osem stikal, od katerih so prvo, tretje in sedmo vključeni, ostala pa ne - zato naslednji zapis: 10100010). V resnici »stikal« ne vključujemo hkrati, ampak zaporedno, kot kaže slika 3. Električni signal, ki ima tako obliko, predstavlja število 162 oziroma 220. *Glede na to lahko ugotovimo,* ***da lahko vsak podatek kodiramo*** *z ustreznim zaporedjem napetosti. Tako lahko zapišemo številke, po posebnem dogovoru (kodi) pa tudi črke in znake, kar bomo pozneje še podrobneje spoznali.*



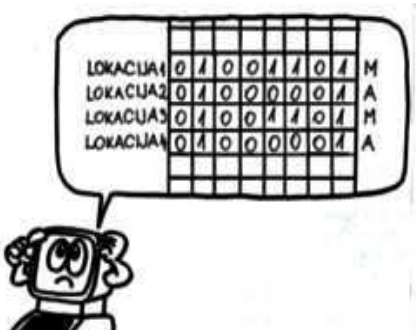
Slika 3: *Predstavitev števila 162 s stikali in žarnicami*



Slika 4: *Predstavitev dvojiško kodiranega zapisa števil 162 in 220*

Računalnik prepoznava in si zapomni le tako zapisane številke in znake. Osnovna pomnilna celica pri računalniku je sestavljena iz določenega števila dvojiških pomnilnih elementov, od katerih lahko vsak hrani le ničlo ali enico, število teh elementov pa določa velikost podatka, ki ga lahko spravimo v eno pomnilno celico. Pri današnjih računalnikih lahko ima ena pomnilna celica 8, 16, 32, 64 itd. dvojiških pomnilnih elementov.

Ali lahko informacije merimo? Odgovor nam je že znan: DA, toda ker nam ta odgovor ni dovolj, si oglejmo še nekaj podrobnosti. Z različno kombinacijo bitov lahko tvorimo večje enote. V praksi se pogosto srečujemo z enoto ***zlog* (ang. byte)**, ki pomeni 8 bitov. Večje enote se imenujejo ***besede***, ki lahko vsebujejo 16 bitov (2 zloga), 32 bitov (4 zloge) itd.; združujemo jih v še večje enote, ki jih imenujemo ***bloki***. Z »biti« torej lahko opišemo črke, znake, besede, zvoke, slike. Vzemimo za primer računalnik, ki potrebuje za ponazoritev enega znaka 8 bitov, kar predstavlja enoto zlog (byte). Ni težko ugotoviti, da 8 bitov lahko opiše 2 na osmo potenco (2^8), kar je 256 različnih možnosti oziroma stanj. To pa je seveda dovolj za kodiranje latinske abecede, številk od 0 do 9, ločil, posebnih znakov itd. Vsak tak zlog je spravljen v ločeni pomnilniški celici, ki si jo lahko za lažje razumevanje predstavljamo združeno v veliko »omaro« (slika 5) s policami. Police so sistematično razporejene in označene (t.i. pomnilniška lokacija), tako da računalnik lahko vedno najde zlog, ki ga išče.



Slika 5: Primer zapisa besede MAMA v štiri »police« oz. pomnilniške lokacije - celice 1, 2, 3 in 4

Vsa števila od 0 do 255 lahko torej predstavimo s kombinacijo 8 bitov. Glede na to je v praksi uporabljen **ameriški standard** za zapis črkovnih in posebnih nadzornih znakov, imenovan **ASCII** (ang. American Standard Code for Information Interchange), v katerem je vsaka tipka na tipkovnici računalnika predstavljena z dvojiškim zapisom, ki predstavlja en zlog.

Za primer vzemimo zelo enostaven proces, ko med uporabo programa za pisanje oz. urejanje besedil pritisnemo tipko »A«; tipkovnica sporoči računalniku kodo 65 (01000001), nato računalnik pogleda v svoj pomnilnik (ASCII koda), kateri znak ustreza kodi 65, in oblikuje na zaslon znak »A«.

ASCII tabela

Dvojiško	Desetiško	Znak	Dvojiško	Desetiško	Znak	Dvojiško	Desetiško	Znak
00100000	**32**	**(presledek)**	01000000	64	**@**	01100000	96	**`**
00100001	33	**!**	01000001	65	**A**	01100001	97	**a**
00100010	34	**"**	01000010	66	**B**	01100010	98	**b**
00100011	35	**#**	01000011	67	**C**	01100011	99	**c**
00100100	36	**$**	01000100	68	**D**	01100100	100	**d**
00100101	37	**%**	01000101	69	**E**	01100101	101	**e**
00100110	38	**&**	01000110	70	**F**	01100110	102	**f**
00100111	39	**'**	01000111	71	**G**	01100111	103	**g**
00101000	40	**(**	01001000	72	**H**	01101000	104	**h**
00101001	41	**)**	01001001	73	**I**	01101001	105	**i**
00101010	42	*****	01001010	74	**J**	01101010	106	**j**
00101011	43	**+**	01001011	75	**K**	01101011	107	**k**
00101100	44	**,**	01001100	76	**L**	01101100	108	**l**
00101101	45	**-**	01001101	77	**M**	01101101	109	**m**
00101110	46	**.**	01001110	78	**N**	01101110	110	**n**
00101111	47	**/**	01001111	79	**O**	01101111	111	**o**
00110000	48	**0**	01010000	80	**P**	01110000	112	**p**
00110001	49	**1**	01010001	81	**Q**	01110001	113	**q**
00110010	50	**2**	01010010	82	**R**	01110010	114	**r**
00110011	51	**3**	01010011	83	**S**	01110011	115	**s**
00110100	52	**4**	01010100	84	**T**	01110100	116	**t**
00110101	53	**5**	01010101	85	**U**	01110101	117	**u**
00110110	54	**6**	01010110	86	**V**	01110110	118	**v**
00110111	55	**7**	01010111	87	**W**	01110111	119	**w**
00111000	56	**8**	01011000	88	**X**	01111000	120	**x**
00111001	57	**9**	01011001	89	**Y**	01111001	121	**y**
00111010	58	**:**	01011010	90	**Z**	01111010	122	**z**
00111011	59	**;**	01011011	91	**[**	01111011	123	**{**
00111100	60	**<**	01011100	92	****	01111100	124	**|**
00111101	61	**=**	01011101	93	**]**	01111101	125	**}**
00111110	62	**>**	01011110	94	**^**	01111110	126	**~**
00111111	63	**?**	01011111	95	**_**			