

4.4.5 Predstavljanje podataka i programa

U komunikaciji sa računarom koriste se različiti oblici informacija: tekst, brojevi, slika, zvuk, video i animacija. Svi oblici se moraju predstaviti u obliku koji je razumljiv računaru.

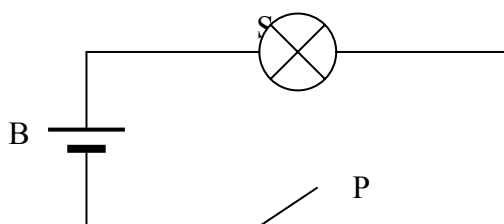
Podaci se u računaru mogu memorisati i obrađivati jedino u binarnom obliku.

Razlog zašto računari rade na binarnim principima je u tome što je većina komponenata računara bazirana na elektronskim elementima koje mogu razlikovati samo dva stanja: da li struje ima ili nema. Na taj način ovi elementi omogućuju definisanje dva stanja svoga rada, koji odgovaraju binarnim ciframa "1" i "0" (latinski "bini" – po dva).

Kao ilustraciju navedimo jedan jednostavni primjer električnog kola koje radi na ovom principu. Kolo se sastoji iz jedne baterije, prekidača i sijalice.

Stanje ovog električnog kola, kada je prekidač zatvoren pa kroz kolo protiče struja i sijalica svijetli, možemo označiti sa "1". Kada se prekidač digne, struja ne teče i sijalica ne svijetli, pa kažemo da je kolo u stanju "0". Dakle, imamo jedno kolo koje u svakom trenutku definiše jednu binarnu cifru.

Stvarne elektronske komponente računara rade takođe na principu dva stanja. Ovaj primjer je ilustracija digitalnog načela. Riječ digitalno nastala je od engleske riječi "digit" koja znači broj.



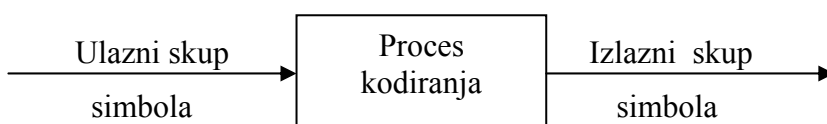
Sl. 4.4 Jednostavno električno kolo kao ilustracija digitalnog načela

Očito je da se binarni brojevi lako mogu zapisati i obrađivati u računaru, jer su njegove komponente prilagođene ovakvom radu.

Međutim, podaci i programi koje računaru dajemo na ulazu obično nisu u ovom obliku.

Čovjeku je lakše sa računarom komunicirati preko standardnih znakova koje upotrebljava u svakodnevnom obavljanju svojih zadataka. To su razna slova, cifre decimalnog brojnog sistema, kao i neki specijalni znaci: (,), ? , ! , = " itd.

Da bi računar mogao da prihvati ove podatke, potrebno ih je kodirati. Kodiranje znači pretvaranje jednog skupa simbola u drugi skup i šematski je predstavljeno na slici 4.5.



Sl. 4.5 Pojednostavljena predstava procesa kodiranja

U našem slučaju ulazni skup simbola, koji se često zove i alfa-numerički skup znakova, prevodi se u niz "0" i "1" tako da ih računar može prihvatiti. Pri tome se kodiranje obavlja tako da se za svaki ulazni simbol odredi tačan niz "0" i "1" koji se zove kod datog simbola i koji se razlikuje od kodova svih drugih simbola.

Postavlja se pitanje koliko je binarnih cifara potrebno da bi se pomoću njihovih različitih kombinacija predstavio dati skup ulaznih znakova: sva slova, sve cifre i određeni specijalni znaci.

Ako se uzme da postoji 10 cifara, oko 30 slova i oko 20 specijalnih znakova, zadatak je da se zakodira oko 60 ulaznih simbola.

S druge strane, sa n binarnih cifara može se postići 2^n različitih stanja, tj. zapisati binarno 2^n različitih ulaznih simbola. Da bi se svi naši simboli mogli zapisati, potrebno je da bude $2^n \geq 60$. Iz ove nejednakosti slijedi da je najmanji broj n , koji je zadovoljava, jednak 6 ($2^6 = 64$).

Kodovi koji koriste 6 binarnih cifara za predstavljanje skupa alfa-numeričkih znakova danas se ne koriste i od njih su najpoznatiji BCD-kod i FIEL-DATA-kod.

Pri konstrukciji računara pokazalo se da je pogodno da ovaj kod bude nešto duži, da bi se pomoću njega mogao predstaviti znatno veći broj uglavnom specijalnih znakova, ali i da bi se u njega ugradile i određene kontrole.

Zbog toga se danas najčešće koriste kodovi sa 8 binarnih cifara pomoću kojih se može predstaviti $2^8 = 256$ različitih znakova.

Najpoznatiji iz ovog skupa kodova su: EBCDIC-kod (Extended binary-coded decimal) i ASCII-kod (American Standard Code for Information Interchange – američki standardni kod za razmjenu informacija), koji je danas prevladao.

Primjeri iz ASCII koda:

Slovu A pridijeljena je vrijednost 65 (od 256), odnosno slovo A je kodirano sa 0 1 0 0 0 0 1 (65 u binarnom obliku), slovo B je kodirano sa 0 1 0 0 0 1 0 (66 u binarnom obliku). Dakle, u računar se za slovo A pohranjuje njegov ASCII kod i to je jedini način da računar razlikuje različite znakove.

Sa druge strane, cifra 1 kodirana je sa 0 0 1 1 0 0 0 1 (redni broj 49 u ASCII tabeli), dok cifra 5 ima kod 0 0 1 1 0 1 0 1 (redni broj 53).

Tačan raspored svih znakova obično se daje tabelarno.

Tabela 5.2 Dio tabele ASCII kodova

Redni broj u ASCII tabeli od 0 do 255	Karakter	ASCII kod – redni broj u binarnom obliku
32	Space	00100000
48	0	00110000
49	1	00110001
50	2	00110010
51	3	00110011
57	9	00111001
63	?	00111111
65	A	01000001
66	B	01000010
67	C	01000011
97	a	01100001
98	b	01100010

Ovi primjeri mogu poslužiti da bi se shvatila razlika između kodiranja i konverzije decimalnog u binarni broj. Kod slova i specijalnih znakova konverzija uopšte ne postoji, pa ne postoji ni problem razlikovanja.

Prilikom konverzije kompletan decimalni broj posmatra se kao jedinstven (a ne cifra po cifra) i konvertuje se u binarni oblik. U računaru se binarne operacije provode samo nad binarnim brojevima nastalim konverzijom, a ne i nad binarnim kodovima. Razliku sagledajmo i na slijedeća dva primjera:

decimalni broj 5	kodiran ASCII	0 0 1 1 0 1 0 1
	preveden u binarni oblik	1 0 1
decimalni broj 15	kodiran ASCII	0 0 1 1 0 0 0 1 0 0 1 1 0 1 0 1
	preveden u binarni oblik	1 1 1 1

Zbog ovih razlika između koda decimalnih brojeva i njihovog binarnog oblika, obično se usložnjava proces obrade numeričkih podataka u računaru.

Naime, podaci se unose preko tastature u decimalnom obliku, računar vrši kodiranje i ovi podaci stižu zakodirani kao niz "0" i "1" u centralnu memoriju.

Međutim, oni se ne mogu odmah obrađivati, nego se prvo iz koda određenim algoritmima prevedu u odgovarajuću binarnu vrijednost. Tek tada se vrši obrada ovih podataka.

Rezultate, čije su vrijednosti u binarnom obliku, treba ponovo prebaciti u kod i tek onda vratiti korisniku. Iako ovo izgleda dosta složeno, računar to obavlja brzo i efikasno, tako da korisnik to uopće ne primjećuje.

Načini predstavljanja brojeva u računaru

U slučaju obrade numeričkih podataka, u računaru se ovi podaci memorišu u binarnom obliku. Binarni brojevi se memorišu u registrima koji imaju određeni broj bitova (binarnih mjesta). Dužina registra zavisi od potrebne preciznosti izračunavanja. Krajnji lijevi bit predviđen je za znak broja, a ostali bitovi su cifre binarnog broja.

Vidjeli smo da unutar jedne skupine od osam bitova možemo zapisati 256 različitih vrijednosti, odnosno vrijednosti od 0 do 255. U računaru

ćemo najlakše upisati cjelobrojnu vrijednost u slučaju kada je manja od 256.

U tom slučaju podatak ćemo zapisati u jedan jedini bajt. Kod cjelobrojnih vrijednosti koje su u rasponu od 0 do 65535 upisat ćemo podatak u dva uzastopna bajta (16 bitova omogućuje 65536 stanja). U četiri uzastopna bajta možemo zapisati cjelobrojne vrijednosti do nešto preko četiri milijarde.

Kod zapisivanja necjelobrojnih (decimalnih) brojeva (realnih brojeva u programiranju), predstavljanje istih se izvodi u formi kliznog zareza, odnosno sa kliznom tačkom (floating point), koje se izvodi u obliku:

$$m b^e$$

gdje je:

- m - mantisa broja,
- b - brojna osnova i
- e - eksponent.

Na primjer broj 234,567 se može predstaviti u kliznom zarezu sa:

$$0,234567 \cdot 10^3 \quad \text{odnosno} \quad 0.234567E3 = 0.234567*10^{**3}$$

Pretpostavimo da nam je zadan broj **29.375**, koji treba prevesti u binarni zapis sa normalizovanom mantisom. Opći postupak koji će se primijeniti, sastoji se u tome da se broj prvo prevede u binarni zapis s nepokretnim zarezom, a zatim naknadno izvrši normalizacija.

Prvo će se prevesti cjelobrojni dio zadanog broja:

$$\begin{array}{r} 29 : 2 = 14.5 \text{ --- } 1 \\ 14 : 2 = 7 \text{ ---- } 0 \\ 7 : 2 = 3 \text{ ---- } 1 \\ 3 : 2 = 1 \text{ ---- } 1 \\ 1 : 2 = 0 \text{ ---- } 1 \end{array} \quad \uparrow$$

Dakle: $29_{10} = 11101_2$

Zatim će se prevesti razlomljeni dio broja:

$$\begin{array}{r} 0.375 \cdot 2 = 0.75 \text{ ---- } 0 \\ 0.75 \cdot 2 = 1.5 \text{ ---- } 1 \\ 0.5 \cdot 2 = 1.0 \text{ ---- } 1 \end{array} \quad \downarrow$$

Dakle: $0.375_{10} = 0.011_2$

Prevedeni cijeli broj ima oblik:

$$29.375_{10} = 11101.011_2$$

Da bi se izvršila normalizacija, zarez moramo pomjeriti za 5 mjesta ulijevo, tako da će konačan zapis imati oblik:

$$29.375_{10} = 0.11101011 \cdot 2^5.$$

Ukupna dužina binarne riječi se dijeli na:

- bit predznaka,
- eksponent i
- mantisu.

Na primjer, ako je dužina binarne riječi 4 bajta, odnosno 32 bita, onda podjela može biti slijedeća:

- 1 bit za predznak (0 za pozitivan i 1 za negativan broj),
- 24 bita za mantisu i
- 7 bita za eksponent (od toga jedan bit za predznak eksponenta).

Mantisa sadrži sve cifre broja. Ako je mantisa broja kraća od raspoloživa 24 bita, ostatak "prostora" do desne granice se jednostavno popunjava nulama.

Kod zapisivanja broja u memoriju računara, nije potrebno prepisati bukvalno sve sa papira. Od mantise se uzimaju samo njene cifre iza decimalne tačke, jer se nula i tačka podrazumijevaju, a od ostatka zapisa treba sačuvati samo predznak broja i eksponent.

Brojna osnova je obično 2, 10 ili 16. Kombinacija brojne osnove i eksponenta određuje stvarnu poziciju decimalnog zareza. Znak mantise isti je kao i znak broja. Eksponent je uvijek cijeli broj. Pogodnim izborom brojne osnove i broja cifara mogu se predstaviti veoma mali i veoma veliki brojevi.

Ovako zapisani brojevi bit će predstavljeni sa određenom pogreškom. Najčešće će biti tačno samo prvih osam znamenki broja. Ova greška potpuno je zanemariva za sve primjene, osim kod nekih posebnih tehničkih proračuna, za koje se onda koristi zapisivanje s većim brojem bajtova rezerviranih za mantisu, što povećava tačnost zapisivanja.

Jasno je da prostor rezervisan za mantisu direktno utiče na tačnost sa kojom se broj smješta, jer je time ograničen broj decimala. Ako se želi veća tačnost, mora se povećati "prostor" za zapis broja, kada se ide na format od ukupno 8 bajta, tj. 64 bita, što se označava kao "dvostruka tačnost".

Kao rezime opisa načina predstavljanja brojeva u računaru, može se kazati slijedeće: Ako se raspláže jednim, unaprijed zadanim prostorom za zapisivanje brojeva (bilo na papiru, ili u memoriji računara), onda forma pokretnog zareza omogućava daleko širi opseg brojeva, što je velika prednost.

Tako na primjer, u osam polja se može upisati najveći cijeli broj 255, bez decimala.

Ako se uvede pokretni zarez, u ovih osam polja može se upisati i broj kao što je $0,7 \cdot 10^{71}$.

Šta više, ako se usvoji dogovor da se iz mantise izostave nula i zarez, jer se oni podrazumijevaju, a takođe da se ne piše desetka i znak za množenje, može se na primjer broj:

$$0,735921 \cdot 10^{71}$$

zapisati u obliku

$$73592171,$$

gdje zadnja dva broja predstavljaju eksponent. Dakle, izbacivanjem suvišnih znakova iz zapisa broja, povećao se prostor za cifre mantise i eksponenta. Pri tome treba voditi računa i o tome da se obezbijedi prostor i za predznake broja i eksponenta.

Načini predstavljanja teksta u računaru

Osim numeričkih vrijednosti, u memoriju računara zapisujemo i tekst, odnosno tekstualne podatke. Iste zapisujemo u obliku niza znakova, pri čemu se svakom znaku pridjeljuje odgovarajući jedinstveni kod, odnosno odgovarajuća kombinacija jedinica i nula. ASCII kod kodira 256 različitih simbola, koji imaju svoj redni broj od 0 do 255 (u dekadnom brojnom sistemu), a u računaru se logički pamti kao

kombinacija nula i jedinica kada se taj redni broj konvertuje iz dekadnog u binarni brojni sistem.

Razlika u kodu između velikih i malih slova je konstantna i iznosi 32 (dekadno).

Brzo širenje računara u sve dijelove svijeta natjeralo je proizvođače softvera da pristupe tzv. lokalizaciji, pod kojom se podrazumijeva prilagođavanje programa korisnicima iz raznih država.

Jedan od glavnih segmenata lokalizacije je obezbjeđenje korisnicima u svakoj državi mogućnosti prikazivanja i štampanja "svojih" znakova. To je dovelo do stvaranja kodnih stranica (code page), koje predstavljaju kodove (šifre) znakova koji se registruju u računaru.

U kodnim stranicama prvih 128 znakova su uvijek isti, pri čemu su kodovi sa rednim brojevima < 32 rezervisani za razna upravljanja (novi red, nova strana itd.). Kodovi sa rednim brojevima > 127 zavise od kodne stranice, kod kojih se za razne države ili grupe država u ovaj prostor (od 128 do 255) smještaju odgovarajući znakovi. Kodne stranice se šifriraju trocifrenim brojem (na primjer 437 za SAD, 852 za latinična slova).

Danas postoji veliki broj kodnih stranica. Sve ih je definisao IBM, tako što je znakove iz različitih država grupisao u četiri tabele:

- Latin 1 (jezici Zapadne Evrope: engleski, njemački itd.),
- Latin 2 (istočnoevropski jezici: BiH latinica itd.),
- latin 3 (srednjoistočni nearapski jezici: turski itd.) i
- Latin 4 (baltički jezici: estonski, letonski, latvijski itd.).

Od 1986. godine koristi se kodna stranica 852. U "gornjih" 128 znakova smješteni su svi znakovi za istočnoevropske jezike. Od 1990. godine softverske firme isporučuju i kodnu stranicu 852 kao dio svojih proizvoda. Kod kupovine softvera treba voditi računa da on ima oznaku EE (Istočna Evropa) ili CE (Centralna Evropa), jer takav softver ima podršku za latinična slova.

Zapis slike u memoriji

Ubrzo nakon konstruisanja prvih računara postalo je jasno da nije dovoljan samo prikaz koji se sastoji od slova i znakova. Zbog toga je razvijen prikaz slike kod koje možemo odrediti izgled svake pojedine tačke na slici. Kako se ovakav način prikazivanja slike koristio za grafiku, nazvan je grafički mod.

Spremanje, odnosno obrada slika u računaru zahtijeva da se slika na neki način pretvori u računarski zapis. Jednostavnije rečeno, to znači da sliku moramo pretvoriti u brojčane vrijednosti.

Rješavanjem ovog problema, razvijene su dvije tehnike računarskog zapisivanja slika:

- slika može biti zapisana kao niz koordinata koje se povezuju linijama, ili krivuljama (vektorska slika), pa računar na osnovu tih podataka svaki put iznova iscertava sliku;
- slika se pretvara u veliki broj tačaka. Računar čuva podatke o svakoj pojedinoj tački i na osnovu toga prikazuje i obrađuje sliku.

Predmet našeg interesovanja je drugi način zapisa slike. U ovom slučaju slika je definisana kao mreža tačaka i ovakav oblik tačkaste slike naziva se **bitmapa**. Riječ je o dvodimenzionalnom modelu, jer je svaka tačka određena brojem retka i položajem u njemu. U memoriji računara kreira se jednodimezionalni model.

Koristeći se računarom, uvijek se zapravo promatraju slike prikazane na opisani način, jer je na isti način prikazana i slika na monitoru. Ova slika, gledana na monitoru, ili je ispisana na štampaču, nastaje prividnim spajanjem većeg broja pomenutih tačaka.

Za prikazivanje najjednostavnije crno-bijele slike koristi se jedan bit za jednu tačku. Tako će prvi bit prvog bajta prikaza slike određivati izgled krajnje gornje lijeve tačke na ekranu. Drugi bit istog bajta određivat će izgled druge tačke u prvom redu itd., sve do osmog bita. Tako unutar jednog bajta imamo prikaz osam tačaka. Slijedećih osam tačaka bit će prikazano u slijedećem bajtu memorije.

Ako radimo sa slikom koja ima 640 X 480 tačaka, bit će nam potrebno 307.200 bitova, odnosno 38.400 bajtova za zapisivanje ovakve slike. Pri

tome se postavljanjem bita u memoriji na vrijednost 1 osvjetljava tačka na ekranu, a postavljanjem bita na 0 zatamni tačka na ekranu. Tako će se postavljanjem odgovarajućih bitova u memoriji osvjetliti željene tačke i formirati slika, kod koje će se na svaku tačku potrošiti samo jedan bit.

Kod ovakvog prikaza slike imamo samo dvije boje. Za realan prikaz slike za svaku pojedinu tačku potrebne su nam i nijanse, odnosno boje. Zbog toga se za jednu tačku na ekranu upotrebljava više bitova.

Na primjer, ako za svaku tačku na ekranu rezerviramo po dva bita, moći ćemo toj tački pridijeliti četiri različite boje (dva bita pružaju četiri različita stanja).

Upotrebom četiri bita možemo pojedinoj tački pridružiti 16 različitih boja, a rezerviramo li cijeli bajt, tačka može biti u jednoj od 256 različitih boja.

Slike kod kojih je svaka tačka zapisana s 8 bitova (1 bajt) koriste se na Internetu za prikaz jednostavnijih slika, jer je ponuđena paleta boja dovoljna, a zbog manjeg broja dostupnih boja, zapis slike je kraći i lakše se prenosi preko Mreže.

Ljudsko oko u stanju je razlikovati približno 7.000 nijansi, a za stvarno uvjerljivu sliku potrebno je i više od toga. Zbog toga se za zapisivanje slike u boji najčešće koriste po 3 bajta za jednu tačku. Tri bajta izabrana su zbog toga što se tačkica u računarskoj slici najčešće sastoji od 3 komponente (crvena, zelena i plava). Tada svaku od komponenti zapisujemo u posebni bajt, a kako 3 bajta imaju ukupno 24 bita, kaže se da je u tom slučaju slika zapisana s 24-bitnom bojom. Ovakav zapis omogućava da se na slici koristi paleta od približno 16 miliona boja.

Zapis nastao kombiniranjem crvene, zelene i plave boje pogodan je za prikaz slike na monitoru. No, želi li se slika otisnuti, koristit će se drugačiji način slaganja boja. U ovom slučaju boja se dobiva miješanjem plavo-zelene (*cyan*), ljubičaste (*magenta*), žute (*yellow*) i crne (*black*). Kako se ponovno za svaku pojedinu komponentu koristi 1

bajt, dolazi se do podatka da se kod ovakvog zapisivanja za jednu tačku utroše 4 bajta, ili ukupno 32 bita.

Navedeni standardi zapisivanja slike najčešće se susreću, odnosno vrlo rijetko se koristi drugačiji prikaz. Iznimka može biti kod prikaza slike na ekranu monitora. Radi smanjenja utroška memorije i povećanja brzine računara, na ekranu se često slika prikazuje tako da se za jednu tačkicu koristi 16 bita (po 5 bita za svaku komponentu i 1 bit za intenzitet).

Načini predstavljanja ostalih zapisa u računaru

Slično brojevima i tekstu i ostali oblici zapisa na računaru (grafika, video, zvuk, animacije, fotografije i njihove kombinacije) predstavljaju se kao odgovarajuće kombinacije nula i jedinica u određenim formatima.

Kod zapisa programa u računarsku memoriju, ovisno o izabranom višem programskom jeziku, postoji nekoliko različitih načina upisivanja programa. Neki viši programski jezici upisuju program u memoriju u obliku teksta, pri čemu su upisana sva slova svih naredbi. Drugi viši programski jezici skraćuju zapis pa ponekad čitave naredbe zamjenjuju samo jednim bajtom. Mašinski jezici zapisuju programe u obliku koji je najrazumljiviji računaru. U tom slučaju, programska komanda zapisana je odgovarajućim kodom koji najčešće ima dužinu od jednog do dva bajta.

Rezime

Kao što smo vidjeli, sadržaj nekog bajta može predstavljati više različitih informacija. U jednom bajtu vrijednost 65 može označavati numeričku cjelobrojnu vrijednost 65 ili veliko slovo A, ili komandu nekog višeg programskog jezika, ili komandu strojnog jezika.

Postavlja se pitanje kako računar razlikuje značenje podatka u memoriji? **Odgovor je: nikako!** Ako u programu naredimo računaru da uzme sadržaj određenog bajta i promatra ga kao tekst, on će to i učiniti. Ako u istom tom programu računaru naredimo da sadržaj istog tog bajta

promatra kao numeričku vrijednost, računar će to i učiniti. Prema tome, značenje pojedinog bajta zavisi od njegove primjene u datom trenutku.

4.4.6 Osnovne jedinice predstavljanja podataka (elementi računarskih podataka)

Koristeći se principom promjene stanja u električnom strujnom krugu, prema slici 4.4, možemo predstaviti svaki broj, slovo ili neki specijalni znak. U osnovi ovog zapisivanja je binarna cifra.

Jedno mjesto u računaru gdje se može zapisati ova cifra naziva se bit (od engleskih riječi: BInary digiT, što znači binarna cifra). Bit je najmanja jedinica u okviru digitalnog uređaja. Ona u računaru može da bude realizovana pomoću različitih elektronskih komponenti. Na primjer, jedna žica po kojoj teče ili ne teče struja je osnovna informacijska jedinica, BIT. Bit je dakle najmanji element svakog podatka. On poprima samo dvije vrijednosti: "0" i "1".

Da bi smo zapisali veću količinu podataka, kako brojeva tako i slova, bitove udružujemo u veću jedinicu – BAJT (engl. byte), sastavljenu od 8 bitova. Dužina bajta određena je kodom za alfa-numeričke podatke (8-bitni ASCII kod).

Znači da u jedan bajt možemo da zapišemo jedno slovo, neki specijalni znak, određenu decimalnu cifru, ali takođe i binarni broj. Bajtom možemo prikazati brojeve od 0 do 255, odnosno definisati 256 različitih stanja.

Riječ bajt nema neko određeno značenje i skovana je u Americi za potrebe računarske tehnologije. Bajt je osnovna jedinica podataka kod savremenih računarskih sistema. Svaka lokacija memorije u računarima koji koriste ASCII ili EBCDIC kod sastoji se od elemenata elektronskog kola (primarna memorija) ili pozicija magnetskog medija (hard disk) koje se mogu predstaviti sa 8 binarnih cifara.

Slijedeća veća jedinica, fiksne dužine, u koju zapisujemo veće brojeve, naziva se riječ (engl. word), a dobiva se povezivanjem više bajta i to cijelog broja bajta.

Riječ je osnovna grupa binarnih cifara ili bajta koja se prenosi tokovima podataka između primarne memorije i registara aritmetičko-logičke i kontrolne jedinice.

Povećavanjem broja bitova u jednoj riječi, kao osnovnoj memorijskoj jedinici, ubrzava se rad računara, zbog toga što računar u jednom trenutku koristi više podataka.

Do sredine osamdesetih standard za računare bio je 8 bita (riječ od jednog bajta) i kod ovakvih računara bilo je moguće prenositi podatke i instrukcije unutar CPU-a u grupama od po 8 bita. Nakon toga konstruirani su računari koji rade sa 16 bita (riječ od dva bajta), a krajem osamdesetih pojavljuju se računari sa dužinom riječi od 32 bita (riječ od 3 bajta). Danas su najrašireniji računari koji rade sa 32 bita, a pojavili su se i mikroprocesori koji rade sa 64 bita (riječ od 8 bajta) Trend je u povećanju broja bitova koje računar odjednom obrađuje.

Očito je da se u jednu riječ može zapisati više slova ili ostalih alfa-numeričkih znakova. Pri tome, može se zapisati i samo jedan broj, ali koji sada može biti znatno veći od onog koji se zapisivao u bajt. tako sa riječi od dva bajta možemo prikazati brojeve od 0 do 65535, ili brojeve između -32768 i $+32767$. Prvi od 16 bitova predstavlja u ovom posljednjem slučaju predznak broja. Još veće brojeve prikazujemo pomoću 4, 8, 16, 32 ili 64 bajta.

Već je pomenuto da je veličina riječi značajna za brzinu rada računara. Ovaj podatak se povezuje i sa preciznošću aritmetičkih izračunavanja, jer se sa povećavanjem dužine riječi omogućava registrovanje decimalnih brojeva sa više decimala. Dužina riječi je povezana i sa kapacitetom primarne memorije. Tako 8-bitni mikroprocesori mogu direktno pristupati samo oko 64 kB memorije, dok 16-bitni mogu imati 640 kB direktno adresabilne memorije.

Kod savremenih računarskih sistema značajan element računarskih podataka je stranica (engl. page).

Stranica je element računarskih podataka koja se kreira zbog razvoja virtualne memorije, gdje se sekundarna memorija tretira kao proširenje

primarne memorije računara. Stranice se prenose između primarne i sekundarne memorije računara u procesu koji se naziva straničenje (engl. paging).

Osim navedenih, koriste se i slijedeće jedinice:

- 1 kB = 1024 bajta = 2^{10} bajta,
- 1 MB = 1024 kB \cong 1 milion bajta,
- 1 GB = 1024 MB \cong 1 bilion bajta.

Kapacitet primarne memorije računara danas se izražava u MB, a kapacitet tvrdih diskova kao perifernih memorija u GB.

Kod mikroračunara baziranih na 8-bitnim mikroprocesorima kapacitet primarne memorije obično je iznosio 64 kB (2^{16} = 65.536 bajta). Kod današnjih računara ovaj kapacitet iznosi od nekoliko stotina MB (256, 512 MB), do nekoliko GB, pri čemu je prisutna stalna tendencija rasta ovih vrijednosti.

Sa korisničke tačke gledišta, podaci su u logičkom i fizičkom smislu organizovani u slijedeće entitete:

- baza podataka – računarski organizovan sistem podataka koji se koristi za organizovanje velikih količina podataka, kao što su na primjer podaci o finansijskom poslovanju firme. Sistem rezervacije avionskih karata je primjer interaktivne baze podataka;
- datoteka – skup srodnih podataka o određenoj pojavi ili problemu, kao što je na primjer stanje nekog skladišta, kao što su podaci o platama ili zaposlenim radnicima u firmi;
- slog – element datoteke koji sadrži podatke o jednom entitetu, kao što je na primjer jedan radnik iz datoteke zaposlenih;
- polje – dio sloga, unutar datoteke, koji sadrži specifičnu informaciju kao što je na primjer šifra proizvoda, ili ime čovjeka;
- karakter – slovo ili neki simbol. Polje sadrži jedan ili više karaktera.