

## Глава 2

### **ПРЕДСТАВЯНЕ НА ИНФОРМАЦИЯТА В КОМПЮТРИТЕ**

Единният информационен подход, предложен от Фон Нойман, както и преобразуването на информацията в компютрите от една форма в друга, налагат да се разгледат определени въпроси, свързани с начините и способите за нейното представяне и използване. За квалифицирания потребител тези въпроси са необходим минимум, който ще му позволи правилно и компетентно да оцени някои параметри на информацията. Най-същественото тук е да се вникне в характеристиките количество и скорост на пренасяне на информацията на различните нива, където тя се използва. При това, всичко следва да се обвърже с най-общите правила и особености за нейното използване от възможните адресанти – устройства, компютри, хора и други.

Както вече беше дискутирано, компютрите работят с различна по вид информация – числа, текст, графика, звук и т.н. В процеса на работа тази информация непрекъснато се трансформира от един вид в друг като във всички случаи за използване от техническите устройства тя се преобразува в цифров (числов) вид, защото различните устройства в компютрите (хардуера) са в състояние да обработват само цифрова информация. Така например всяка буква, символ, елемент на звук, графика или картина вътре в машината има определена числова стойност. Тази стойност е представена в подходяща (разбираема от устройствата) форма. За да се изясни добре въпросът с представянето на информацията, използвана в компютрите и компютърните системи, е необходимо в най-общ план и съвсем бегло, да се разгледат числовите системи. Те се наричат още бройни системи и числата могат да се представят, преобразуват и изразяват по стойност, чрез тях.



**Бройни системи** (броячни, числови) е онази съвкупност от правила и символи, наречени цифри, с помощта на които се представят различни количествени изрази (числа). В практиката се използват два вида бройни системи – позиционни и непозиционни.

*Непозиционните бройни системи* са онези, при които мястото на цифрата в числовия запис не оказва влияние върху нейната стойност, т.е. тя е постоянна и не зависи по никакъв начин от нейното място в числото. Като примери за такива бройни системи могат да се посочат римската, гръцката, милетската и др. Основният недостатък на непозиционните бройни системи е, че чрез тях големите числа се представят неефективно. Често това би могло да бъде източник на грешка при определяне на стойността на числата. Пример за непозиционна бройна система е римското число *XXXIX*, което има стойност 39 и се записва чрез пет разряда. Цифрите от него имат една и съща стойност независимо от това в кой разряд са записани. Римската цифра *X* е със стойност 10, както в нулевия разряд на числото, така и във всички останали, където тя е използвана – втори, трети, четвърти и пети. Стойностите на числата в непозиционни системи се определят единствено на базата на определени правила. Тези бройни системи не намират приложение за обработка на числови изрази и извършване на аритметични действия в изчислителната техника.

*Позиционните бройни системи*, обратно на непозиционните са онези, при които стойността на цифрата в запис на числата зависи от позицията (разряда), в който тя се намира. Пример за това е числото – *1111*, което се състои от четири разряда. Стойността на цифрите в различните разряди е различна и се получава като същата се умножи по определено число – например 10, повдигнато на степен мястото на цифрата в числовия запис. Числовият еквивалент (стойността) на числото се получава като се съберат получените стойности във всеки отделен разряд на числовия запис.

Позиционните бройни системи са тези, с които се извършват всякакви пресмятания и единствено те намират приложение в компютърната техника, програмирането и информационните тех-



нологии. В различните позиционни бройни системи се използват различен брой цифрите. Прието е броят на цифрите, с които се осъществява записът на едно число в позиционна бройна система да се нарича основа на система и се бележи с  $q$ . В тази връзка може да се подчертае, че са налице безброй много позиционни бройни (числови) системи, като стойността на числата за цялата част, в коя да е от тях се определя от израза:

$$A = a_n q^n + a_{n-1} q^{n-1} + \dots + a_1 q^1 + a_0 q^0$$

При определяне стойностите на числата в позиционни бройни системи, съществено значение има основата на бройната система  $q$ , която е цяло число по-голямо или равно на две. В такъв случай най-ниската позиционна бройна система, в която могат да се записват числата е двоична (основа  $q = 2$ ), следва троична (основа  $q = 3$ ), четвъртична (основа  $q = 4$ ), десетична (основа  $q = 10$ ) и така нататък до безкрай.

Очевидно в позиционна бройна система, стойността на числото се определя както от стойностите на отделните цифри, включени в записа, така и от позицията, където те се намират в него. На практика това означава, че при един и същи запис на числото в различните позиционни бройни системи стойността ще е различна. За потвърждаване на това, имайки предвид изразът по-горе, може да се разгледа числото  $1111$ , което е записано в десетична ( $q = 10$ ) и двоична ( $q = 2$ ) позиционна бройна система:

- за десетичната се получава  $1 \cdot 10^3 + 1 \cdot 10^2 + 1 \cdot 10^1 + 1 \cdot 10^0$ ;
- за двоичната ще е  $1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$ , т.е. стойност  $15$ .

В компютърната техника, програмирането и обработката на информация всички аритметични, логически и други операции над числовите стойности се извършват с числа, представени в позиционна бройна система. В такива бройни системи се преобразува и нечисловата информация с цел обработка от устройствата на компютрите. При това от безброй многото позиционни числови системи приложение в практиката намират само някои от тях и това



е свързано с нивото, на което се представя и разглежда числовата информация. В тази връзка съвсем условно могат да се разграничат и разгледат три основни нива на представяне на информацията, използвана в компютрите – ниво на потребителя, ниво на вътрешно представяне и междинно (сервизно) ниво.

**Потребителско ниво на представяне на информацията** в компютрите е онова, на което всеки потребител (човек) директно оперира с числата. Това е нивото, където информацията се въвежда в традиционни за хората форми и се получават резултатите от обработката отново в такива форми. Хората общуват помежду си с десетична бройна система, тя е исторически обусловена, има основа  $q = 10$  и оперира с десет цифри  $0, 1, 2, 3, \dots, 9$ . Следователно на потребителско ниво се използва единствено десетичната бройна система в различните си форми на представяне на числата.

От гледна точка на компютърните технологии и програмното осигуряване, числата в позиционната десетична бройна система, на потребителското ниво на използване, могат да се записват в две основни форми – фиксирана и плаваща запетая.

**Фиксираната запетая**, наричана още обикновено число, е традиционната форма за записване на числата в десетична бройна система и основно тя се използва в ежедневно общуване между хората. При тази форма на представяне, стойността на числото е видна от самия запис и за нейното определяне не са необходими допълнителни преобразувания или пресмятания. Като пример за фиксирана запетая може да се посочи всяко обикновено, познато в широката практика и ежедневието на потребителите десетично число, например числото  $5$ , дробното число  $3,14$  и т.н.

**Плаващата запетая**, наричана и експоненциална форма, е начин на представяне на числата, при който за определяне на стойността (количествения израз) са необходими допълнителни пресмятания. Числата с плаваща запетая, познати още като „плаващи“, „двойни“, „експоненциални“ или „реални“ могат да бъдат еднозначно определени посредством израза:

$$A = \pm Mq^{\pm p}.$$

Изразът представлява произведение на мантиса ( $M$ ), т.е. значещата част на числото, умножена по основата на бройната система ( $q$ ) повдигната на степен цяло число ( $p$ ). Степенният показател в израза се наричана порядък и той определя размера на числото. Максималната стойност на порядъка зависи от конкретното програмно осигуряване, чрез което се обработват числата. Например в Excel тя е мантиса умножена по  $10$  на степен около  $300$ . Мантисата на числата при плаващата запетая е определена от стойността на числото и може да бъде положителна или отрицателна. Ако числото е отрицателно тя ще бъде отрицателна, а при положителните числа – положителна. За разлика от други предметни области, в компютърните информационни технологии и програмното осигуряване за обработка на числата, мантисата се представя от един десетичен разряд в интервала по-голям или равен на единица и по малък от  $10$  ( $1 \leq m < 10$ ). Тя може да има и дробна част, ако числото съдържа такава. При преобразуването на числата от фиксирана в плаваща запетая, десетичният знак (запетаята) се премества толкова разряда на ляво или дясно, докато се получи един цифров разряд в посочения по-горе интервал. За числата по-малки от единица преместването на запетаята е на дясно и степенния показател ще бъде отрицателно число, а за тези по-големи от единица преместването на запетаята е на ляво и степенния показател е положително число. Основата  $q$  е цяла стойност и зависи от това в коя бройна система е записано числото. В разглеждания случай числата са в десетична бройна система и съответно основата  $q$  е  $10$ , но ако числата са записани в друга бройна система, то тя ще има друга стойност. Стойността на порядъка  $p$  е равна на броя на разрядите, с които запетаята е преместена на ляво или на дясно. Това е цяло десетично число, което показва на коя степен следва да се повдигне основата на бройната система.



фиксирана запетая		плаваща запетая
0,00723	→	$7,23 \cdot 10^{-3}$
0,00000000000000000000000000106	→	$1,06 \cdot 10^{-25}$
423000000000000000000000000000,0	→	$4,23 \cdot 10^{+27}$
1000000000,	→	$1 \cdot 10^{+9}$

**Фиг. 3.** Примери за записване на числа в плаваща запетая.

В подкрепа на направените разяснения, на Фиг. 3 са показани няколко примера за преобразуване на десетични числа от фиксирана в плаваща запетая. Там са спазени посочените правила и условия за преместване на запетаята наляво или надясно в числовия запис. В първият пример числото е по-малко от единица, запетаята се премества три разряда надясно и порядъка е цяло отрицателно число със стойност три. Вторият пример също представя число по-малко от единица, преместването на запетаята е 25 разряда надясно и порядъка ще бъде отрицателно число със стойност 25. В третият пример числото е цяло без дробна част и в този случай преместването на запетаята е наляво с 27 разряда. За четвъртият пример числото е цяло с една значеща цифра, различна от нула и в този случай мантисата за експоненциалната форма също ще бъде една цифра за цяла част без да има дробна.

От изложеното до тук вероятно изниква въпросът защо е необходима плаващата запетая и не усложнява ли тя представянето на числата. Отговорът е лесен и се обяснява с броя на необходимите разряди в конкретно число. В случаите когато стойностите на числата са много големи или много по-малки от единица, т.е. броят на разрядите, с които те се записват е голям е целесъобразно, а често и задължително, да се премине към плаваща запетая.

Особеното при представяне на числата с плаваща запетая в компютърните технологии и по-точно в различните програми за обработка, е наличието на известна разлика в предлагания запис. В практиката за по-лесно изразяване, изобразяване или прегледност



на информацията въвеждана от входните устройства, или пък извеждана върху изходни устройства, втората част на произведението се записва не както е показано в примерите на Фиг. 3, а по друг по-удобен и по-прегледен начин. Примери за различното записване на експоненциалните числа съгласно математиката и програмите за обработка са показани в таблицата от Фиг. 4.

Фиксирана запетая	Експоненциална форма по правилата на математиката	Експоненциална форма по правилата на програмите
0,00723	$7,23 \cdot 10^{-3}$	7,23E-03
0,0523	$5,23 \cdot 10^{-2}$	5,23E-2
4378765	$4378765 \cdot 10^6$	4378765E+06
1000000000	$1 \cdot 10^9$	1E+9

**Фиг. 4. Представяне на десетични числа в различни форми.**

В първата колона на таблицата това са числа във фиксирана запетая (обикновени числа), във втората те са в плаваща, записани по правилата на математиката, а в третата записът е осъществен по правилата на компютърните програми за обработка на числова информация. В третата колона множимото на произведението – основата на бройната система с нейния степенен показател се замества с буква, следвана от знака на степенния показател и числото, представляващо стойността на порядъка (степенния показател). Възприетият начин на записване значително улеснява потребителя и най-вече опростява техническата реализация при извеждане и въвеждане на такива стойности от различните устройства – монитори, печатащи устройства и други. В примера, като символ за кодиране е използвана латинската буква *E*, но при някои програми и особено в програмирането могат да се използват и други символи. Веднага след символа за кодиране се записва знака на степенния показател и неговата десетична стойност.

Очевидно записването на числата по правилата на компютърните програми е значително по-удобно. Записът се представя в



един ред и не е необходимо да се ангажира част от него за разполагане на степенния показател. При записване на степенния показател някои от програмите поставят нула пред значещата цифра, а други не. В посочените примери от дясната колона на таблицата за първото и третото число пред значещия числов разряд на порядъка е поставена нула, а при тези от втори и трети ред нулата отсъства.

При използване на десетичната бойна система в компютрите на потребителско ниво са налице и редица особености. Те трябва да се разбират, да се съобразяват и интерпретират правилно.

**Особености в използване на десетичните числа** в компютрите и правилата, по които те се отличават от традиционно приетите в математиката са онези, които са свързани основно с разнообразното програмно осигуряване. В различните програми и системи за обработка тези особености и правила са различни. Съществените от тях могат да се обобщят в следните няколко пункта:

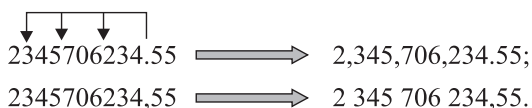
- 1) Ако програмното осигуряване допуска, числата могат да се въвеждат, както във фиксирана, така и в плаваща запетая.
- 2) При много големи или много малки стойности на числата програмното осигуряване, по принцип, автоматично ги преобразува и извежда върху устройствата в плаваща запетая.
- 3) Ако програмите предвиждат това, то цялата част на числата може да бъде отделена от дробната както с точка, така и със запетая. В единия случай те се определят като числа записани в европейски (EU) стандарт, а в другия в американски (US) стандарт. В разпространените на Майкрософт операционни системи, избора на формата се осъществява автоматично, чрез избор на съответния регион. Настройва се от контролния панел (Control panel) на операционна система Windows и иконата от там езикови и регионални настройки (Region and language). Нужно е да се подчертае, че много програми изискват настройка на конкретен формат за представяне на числата – разделител между цялата





и дробната част запетая или точка. Това означава, че за да работи коректно определена програма е необходимо да се избере изисквания от нея формат за представяне на числата – България, Англия, САЩ и т.н.

4) При визуализиране на числата или извеждане на печат, с цел тяхното по-лесно разчитане, в програмите за обработка са въведени допълнителни спомагателни формати. Наричат се още търговски формат на числата или числа с отделени порядъци. При тези формати порядъците в цялата част на числото са отделени с някакъв символ за разделител. Разделителят между порядъците се назначава автоматично от операционната система и зависи от настройката за избор на регион (държава).



**Фиг. 5.** Търговски формат на десетичните числа.

На Фиг. 5 са показани два примера за използване на търговски формат за записване на числата. В първият числото е записано в американски (US) стандарт и като разделител в търговския формат за всеки един от порядъците в цялата му част е използван символа запетая. Във втория пример числото е в европейски (EU) стандарт и като разделител на порядъците за търговския формат е използван символа интервал. Програмното осигуряване предоставя възможност за преобразуване в търговски формат, а потребителят сам преценява дали да го използва или не. Много специализирани програми за обработка на числова информация предлагат извеждането на числата да се извършва основно в търговски формат.

В изнесеното по-горе бе употребено понятието „порядък“. Порядъците са стандарти за записване на числата и тяхното именуване по международни правила. Съгласно тези стандарти за малките числа има еднозначни наименования – едно, две, ....де-



вет, десет, единадесет и т.н. За много големите или много малките числа, чиито разряди са три или повече, наименованията също са стандартизирани по международната система „СИ“ (SI). За големите те са десет, сто, хиляда, милион, милиард и т.н. За дробните се използват десета част, стотна част, хилядна част, милиардна част и т.н. В различните страни по света има и разлики в наименованията на стойностите на числата. Например наименованието милиард и билион се използва за означаване на една и съща стойност – едното е по френската система, а другото – по швейцарската.

В практиката за именуване на числата в десетична бройна система с голям брой на разрядите в тях се използва групиране от по три цифри. За всяка група от по три цифри от дясно на ляво в цялата част на числото и от ляво дясно в дробната се използва име. В този смисъл, порядъците са кратни и дробни и те означават определени стойности. Тяхното основно предназначение е да се стандартизира наименованието на определените числени значения.

*Кратните порядъци*, за представяне на десетичните числа са цяла степен на основата на десетичната бройна система ( $q = 10$ ). Те са  $10^3$  (хиляда) – нарича се *Кило* и се означава с *K*;  $10^6$  (милион) – нарича се *Мега* и се означава с *M*;  $10^9$  (милиард) – нарича се *Гига* и се означава с *G*;  $10^{12}$  (трилион) – нарича се *Тера* и се означава с *T*;  $10^{15}$  (квадрилион) – нарича се *Пета* и се означава с *P*;  $10^{18}$  нарича се *Екса* и се означава с *E*;  $10^{21}$  – нарича се *Сета* и се означава със *Z*;  $10^{24}$  – нарича се *Йота* и се означава с *Y*. Има и други по-големи от тях, но те рядко се използват в практиката, тъй като представят много големи стойности. Кратните порядъци се използват за означаване на големи стойности като за числата, чиито разряди са повече от 24, за сега няма точно фиксирани имена и те се означават като цяло число с положителен степенен показател (плаваща запетая). Като примери за използване на кратни порядъци в десетична бройна система могат да се посочат, измерването на високи честоти, например 20 KHz (20 000 Hz), 3,2 GHz (3,2.10<sup>9</sup> Hz), измерването на мощност – *ват* (W), *киловат* (KW) и други. Към настоящия момент измерването на количеството информация в компютрите е от

□ порядъка на *Мега* и *Гига*, като вече бързо навлиза и се използва и порядъкът *Тера*. Вероятно съвсем скоро ще се премине и към квадрилион (*P – Пета*). Предсказването на по-дълъг период от време е рисковано, тъй като може да се окаже неточно и невярно, поради лавинообразното нарастване на информацията.

*Дробните порядъци* са също точна степен на основата на десетичната бройна система, но тук тя е с отрицателен знак и редицата са числа по-малки от единица (част от цялото). Тези порядъци се използват за представяне на числови стойности по-малки от единица. Те са стандартизирани като  $10^{-3}$  (хилядна част) – нарича се *Мили* и се означава с *m*;  $10^{-6}$  (милионна част) – нарича се *Микро* и се означава с *μ*;  $10^{-9}$  (милиардна част) – нарича се *Нано* и се означава с *n*;  $10^{-12}$  (трилионна част) – нарича се *Пико* и се означава с *p*;  $10^{-15}$  (квадрилионна част от цялото) – нарича се *Фемто* и се означава с *f*;  $10^{-18}$  – нарича се *Ато* и се означава с *a*;  $10^{-21}$  – нарича се *Сенто* и се означава със *z*;  $10^{-24}$  – нарича се *Йокто* и се означава с *y*. Дробните порядъци намират голямо приложение при измервания в областта на медицинската информация. В компютърната техника тези порядъци се използват за измерване на малки размери и малки интервали от време. С тях например се измерва характеристиката скорост на пренасяне на информацията, време за превключване на различни елементи от хардуера на компютъра, размер на електронните компоненти и други. Към настоящия момент технологиите са достигнали равнище, при което като временни и размерни характеристики се използва дробния порядък *Нано* –  $10^{-9}$ . За време това е порядъкът *Наносекунда* (*ns*), а за размер на електронни елементи – *Нанометър* (*nm*). В следващите години бариера за скорост вероятно ще е премината и ще се отиде към порядък за време *Пикосекунда* –  $10^{-12}$ . Няма яснота как ще се премине към следващият порядък за размер, но вероятно и това ще стане.

На потребителско ниво числовите стойности се въвеждат в компютърните програми в десетична бройна система, фиксирана или плаваща запетая и се извеждат отново в такива форми. Ако има изключение от това правило, то вероятно ще се касае за някакъв



специфичен, строго специализиран случай, който не е типичен и не засяга широката практика.

Като цифрови машини, компютрите са в състояние да обработват единствено числа и то представени в подходящ формат. При хората въпросът е предрешен от историята и се свързва с десетте пръста на ръцете. Как да стане при компютрите и колко „пръста“ имат те? Това са въпросите, които са вълнували учени и конструктори в началото на миналия век. Отговорът е намерен в разработване и предлагане на нивото за вътрешно представяне на информацията в компютрите.

**Вътрешно представяне на информацията** в компютрите е нивото, на което числата и символите се преобразуват в двоични стойности. То е най-ниското ниво, защото информацията в него се използва и е „разбираема“ само от техническите устройства на компютъра и не намира приложение в практика на хората.

За вътрешно представяне на информацията, от възникване на компютрите и до днес се използва единствено позиционната двоична бройна система. По данни на различни източници тя е предложена за първи път от Джон Атанасов и представлява една от неговите основни постановки за възникването и развитието на компютърната техника. Използването на двоична бройна система за вътрешно представяне на информацията в компютрите е свързано с редица предпоставки и конструкторски виждания.

*Първата основна предпоставка* за въвеждане на двоична бройна система се свързва с наличието само на две цифри за представяне на числата. Това на практика е и първият съществен отговор, според който хората имат десет пръста и работят с десетична бройна система, а компютрите, условно казано са с „два пръста“ и работят само с две цифри.

От техническа и конструктивна гледна точка най-лесно могат да се определят две състояния – включено или изключено; наличие или отсъствие на напрежение; наличие или отсъствие на електрически ток; намагнитено или размагнитено; светло или тъмно и така нататък, т.е. налице е един бит информация.



Използването само на две състояния за изразяване на цифрите на практика разрешава сложни технически и технологични проблеми и позволява да се построят прости устройства за еднозначното представяне на тези състояния. Например наличието на определено електрическо напрежение в едно техническо устройство регистрира двоичната цифра *единица*, а отсъствието на напрежение цифрата *нула*; включената електрическа верига, по която тече ток определя състоянието *единица*, а изключената – безтокова верига *нула*; зареденият електрически кондензатор съхранява двоичната цифра *единица*, а разреденият *нула* и така нататък.

*Втората основна предпоставка* за въвеждане на двоичната система е свързана с елементарната реализация на основните аритметични операции събиране, изваждане, умножение и деление. Тук таблиците за умножение и събиране са много лесни, свързани са с наличието само на две цифри и могат лесно да се реализират с технически устройства. Още повече, че съгласно теоретичните постановки на математиката всички сложни пресмятания могат да се представят и реализират само с помощта на операциите събиране и изваждане. Точно тази особеност и постановка е използвана при построяване на компютрите.

*Третата основна предпоставка* за въвеждане на двоичната бройна система е възможността в компютърната техника и програмирането широко да се въведе апарата и инструментариума на алгебрата на логиката. Основите на тази алгебра са разработени от Джордж Бул и поради това тя се нарича още Булева алгебра. Като наука Булевата алгебра се явява раздел от математическата логика и оперира с класове, съждения или елементи с две състояния. Предвид на това, методите и средствата на Булевата алгебра масово се използват и прилагат в теорията на изчислителните машини в дискретните автомати, в разнообразни задачи за анализ и синтез на устройства с превключващи схеми и други. Апарата на алгебрата на логиката намира изключително голямо приложение в компютърната техника и програмното осигуряване за нея. На база на нея се реализират разнообразни езикови средства от областта



на програмирането и обработката на информацията и са разработени конкретни технически устройства за компютърната техника. Характерна особеност в инструментариума на Булевата алгебра е, че тя оперира с логически променливи, които се наричат още двоични или логически функции.

**Логическа променлива и логически функции** са онази, които могат да приемат само две стойности. Това се елементи от алгебрата на логиката, които намират много голямо приложение в компютърните технологии.

*Логическата променлива* е в основата на логическите функции. Тя може да приема само две стойности – „Истина“ и „Неистина“. Нарича се също „Булева променлива“ или „Двоична променлива“. Основанието да се нарича така се заключава във възможността тя да приема само две противоположни състояния. На едното се присвоява числова стойност единица, а на другото нула.

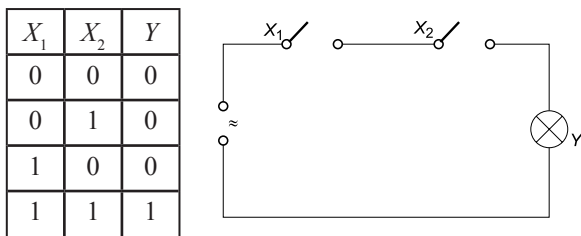
В компютърните информационни технологии логическата променлива „Истина“ се представя с единица, а „Неистина“ с нула. Като пример за логическа (двоична) променлива може да се посочи всяко твърдение или физическо явление с две противоположни състояния – включено и изключено, тъмно и светло, топло и студено и т. н. Едното състояние се означава (кодира) с единица, а другото с нула. Например, ако в компютъра следва да се представи състояние „Истина“, то просто ще се въведе или запише (установи) единица. В техническата литература логическите променливи са известни още като „Логическа единица“ и „Логическа нула“ и на всяка една от тях съответства някаква физическа величина. Тази величина може да бъде електрическо напрежение с определена прагова стойност – стойност, която се променя (превключва) скокообразно, електрически ток, светлинен поток и т.н.

*Логическите функции* са онези, които зависят само от логически променливи и могат да приемат само две стойности – *Истина* и *Неистина*. Те намират изключително голямо приложение в компютърната техника и са в основата на построяването и принципа на

действие на редица технически устройства. Логическите функции намират приложение и на потребителското ниво, например в програмирането, в създаването и работата с бази от данни в електронните таблици и т. н. Чрез логическите функции могат да се синтезират логически съждения, логически изводи и да се построят логически анализи.

Известен е набор от логически функции, три от които са основни – „И“, „ИЛИ“ и „НЕ“. Тези функции се наричат още класически или основен базис. Посредством тях е възможно да се изразят всички останали. Логическите функции могат да се изразяват по различни начини, като най-често използваните в практиката са *таблица на истинност*, *физическа интерпретация* и *аналитичен израз*. В следващите примери, за всяка една от основните логически функции, са показани схематично таблиците на истинност и елементарен пример за тяхната физическата интерпретация, последвани от кратък коментар, относно смисъла и приложението им.

Функцията „И“ наричана още „Конюнкция“, „Логическо умножение“ или на английски „AND“ е истина, ако всички променливи, от които зависи са със стойност единица (*Истина*).

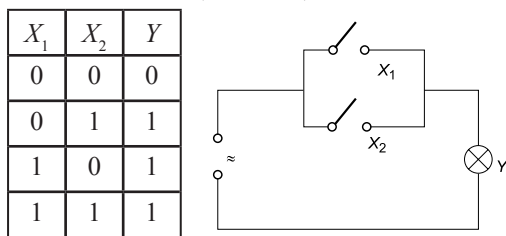


Фиг. 6. Таблица на истинност и физически модел на функция „И“.

В таблицата на истинност (Фиг. 6) е показан възможния набор на две логически променливи  $X_1$  и  $X_2$ , от които функцията  $Y$  зависи. За всеки един от тях са посочени и стойностите, които тя е в състояние да приема. Пълният набор на комбинациите от възможни стойности на определен брой от двоични променливи е  $2n$ , къде-

то  $n$  е броят на променливите. В примера променливите, от които функцията зависи са две и следователно възможните комбинации са четири. При три променливи  $2^3 = 8$ , комбинациите ще са осем, при четири те ще са  $2^4 = 16$ , ... при осем  $2^8 = 256$  и така нататък. От таблицата се вижда, че стойността на функцията  $Y$  е истина, само ако са истини (единици) и двете променливи. Физическият модел, показан в дясно на фигурата, илюстрира наличието и смисълът на логическата функция „AND“. Като пример от практиката за функция „И“ може да се посочи превключването на клавиатурата от латиница на кирилица. В този случай най-често стандартно е необходима функция „И“ на два клавиша (Alt + Shift).

Функцията „ИЛИ“ известна още като „Дизюнкция“, „Логическо събиране“ или „OR“, обратно на функцията „И“, е в състояние истина, ако поне една от включените в нея логически променливи са в състояние единица (Истина).



**Фиг. 7.** Таблица на истинност и физически модел на функция „ИЛИ“.

Таблицата на истинност и физическата интерпретация от Фиг. 7 достатъчно добре илюстрират смисъла и действието на функцията „ИЛИ“. За разлика от функцията „И“ тук е налице истина, ако коя да е от променливите, от които тя зависи са в състояние Истина (единица). Физическата интерпретация е видна от показаната на фигурата електрическа верига, при която лампата светва (функцията е истина), ако е затворен кой да е от двата превключвателя. Като често срещан пример в практика на компютрите може да се посочи отново клавиатурата и дублираните в нея клавиши по функция „ИЛИ“ – клавишите *Shift*, *Ctrl*, *Alt* и други.



□ Функцията „НЕ“, наричана още „Отрицание“, „Превключвател“, „Инвертор“, „Алтернативна функция“ или „NOT“ се използва в състава на логически изрази и намира голямо приложение в програмирането, в базите от данни в електронните таблици и други. Функцията „NOT“ има механизъм на действие, съгласно който тя алтернативно се превключва в двете си възможни противоположни състояния – активно и неактивно, включено и изключено и т.н. Пример за реализиране на функция „НЕ“ е механизмът на действието на клавиша от клавиатурата „Caps Lock“. При всяко еднократно натискане на този клавиш се преминава от големи в малки букви и отново, т. е. налице е логическа функция „НЕ“ (отрицание) или казано по друг начин *Алтернативно превключване*. Ето защо много често клавиши, команди или опции, които работят по подобен начин се наричат още *алтернативни*.

Разгледаните до тук предпоставки за въвеждане на двоичната бройна система при вътрешното представяне на информацията в компютрите не са единствени. Те са основните, които са предизвикали учените и конструкторите да прибягнат към нея и тя вече е факт, с който днес всички се съобразяват. Понастоящем основни характеристики в компютрите, свързани с техните изчислителни възможности и измерването на информацията в тях се свързва с двоичната бройна система и в близко бъдеще не се очертава нещо по-различно. Ето защо разглеждането на начините и единиците за измерване на информацията от вътрешното представяне в компютрите и върху техническите носители има много съществено значение при изучаване основите на компютърната техника.

**Единици за измерване на информацията в компютрите** е въпрос от първостепенно значение при разглеждане на различните процеси в техническите устройства и програмното осигуряване. Само при неговото правилно разбиране и интерпретиране е възможно да се вникне в редица основни и допълнителни характеристики, както на компютрите, така и на свързаното с тях програмно осигуряване. Изучаването и разбирането на единиците за измер-



ване на информацията е важно за потребителите, тъй като при работата си с компютър те непрекъснато ще се сблъскват с тях и ще ги използват в практиката си. В тази връзка може да се отбележи, че като единици за измерване на информация се срещат различни измерителни величини, но не всички намират широко приложение сред потребителите. Някои от тях имат по-скоро чисто техническо значение, ориентирани са към конкретни характеристики на отделни устройства или програми и не представляват голям интерес. От всички използвани измерителни единици на информацията най-голямо приложение в компютърната практиката понастоящем имат единиците бит – *bit* и байт – *Byte*.

*Информационната единица бит (bit)*, както вече беше коментирано, е най-малката информационна единица за измерване на количеството информация от нейното вътрешно представяне в компютрите. Наименованието бит произлиза от английски и има много значения, едно от които е малко количество.

*Информационната единица бит за секунда (bit per second)* е най-малката информационна единица за скорост на пренасяне на информацията от нейното вътрешно представяне в компютрите. Като една от съществените характеристики на информацията, мярката скорост практически показва колко бита за една секунда могат да се предадат (приемат) от различните устройства на компютъра между компютрите и в компютърните мрежи. За разлика от най-малката информационна единица за количество *bit*, то единицата за скорост *bps* (бит за секунда) понастоящем намира много голямо приложение при измерване на информацията.

При преобразуване и пренасяне на аудио и видео информация като оценка на скоростта често се въвежда и понятието побитова скорост (*bit rate*). Тази скорост показва какво е качеството на преобразуване на аудио и видео информацията при нейното компресиране с цел пренасяне и се оценява в битрейт. Характеристиката Битрейт започва да се използва с навлизане на Интернет и предаването на мултимедийна информация – аудио и видео. Мерната единица за нея е също бит за секунда. За разлика от чисто техническата

□ характеристика бит за секунда, то *bitrate* отчита само ефективната скорост за предаване (приемане) на полезната информация. Не се отчита спомагателната, която така или иначе се предава и приема заедно с полезната. Например, за да има пълно и качествено предаване на аудио е необходима скорост от порядъка на няколко десетки хиляди бита за секунда, т.е. да е налице *bitrate* от порядъка на около 10 000 и повече *bps*. Високият битрейт предполага високо качество на звука в аудио или видео информацията, но за сметка на това значително увеличава нейното количество.

*Информационната единица Байт (Byte)* е по-голяма от единицата бит, представя се от осем двоични бита и се явява основна информационна единица, използвана като мярка за количеството на информацията от нейното вътрешното представяне в компютрите и техните устройства. Байтът е извънсистемна единица за измерване на информацията и се използва само при компютрите. Приемането му като основна информационна единица за количество информация е свързано с появата на персоналните компютри още в началото на осемдесетте години на миналия век, когато за представянето на един символ (буква, цифра или специален знак) се е използвал точно един байт. И до момента той си остава основна информационна единица, макар че тя вече е твърде малка. Може да се счита, че използването на байта, като основна информационна единица в известна степен е исторически обусловено.

Информацията, която може да се запише в един байт е числова стойност в интервала 0 до 255. Нулата са осем двоични цифри със стойност нула – 00000000, а 255 са осем единици – 11111111. Тази стойност може да бъде определена като се съберат теглата на всеки един от разрядите в двоичното число. Калкулаторите, включително и предлаганият в Windows, също имат вградени функции за преобразуване на числата от една бройна система в друга, така че при нужда те могат да бъдат използвани.

*Основната информационна единица за скоростта* на пренасяне на двоичната информация се измерва в *Байт за секунда (Bytes per second)*. Тя показва колко байта се пренасят за една секунда



между различните източници и приемници. Както при информационната единица *bit*, така и тук за означаване на тази основна характеристика за скорост на пренасяната информация се използват различни означения. От тях най-голямо разпространение в практиката има означението *Bps*, но се използват и другите. Като интересен факт при означаване на скоростните характеристики може да се посочи твърде голямата прилика между тях. Например двете означения *bps* и *Bps* са идентични по значение, звучене и изговаряне. Разликата е в изписването на първата буква. Ако буквата е малка, то се има предвид *бит за секунда*, а ако е голяма, то тогава значението е за скорост *байт за секунда*.

**Кодиране на двоичната информация** в устройства от хардуера на компютрите е тема, която касае повече специалистите. Нейното дискутиране от квалифицираните потребители също е полезно, защото то отговоря на много въпроси, свързани с количеството информация, скоростта за предаване, въвеждането и извеждането от устройствата. Във всички случаи информацията в устройствата на компютрите е под формата на двоични стойности и може да се разглежда в два аспекта – представяне на числа и представяне на символи (букви, цифри и т.н.)

*Числата се представят в двоичен формат* в двете основни форми – фиксирана и плаваща запетая. Всяка числова стойност заема един или няколко последователни байта от устройствата.

Вътрешното представяне на числата не представлява особен интерес. Просто следва да се знае, че числата, в зависимост от техните стойности и изразена точност, са еднозначно определена последователност от един или няколко байта. Тази последователност се разглежда от софтуера като обща стойност и точно програмите са тези, които я преобразуват, записват и определят.

*Символите се представят също с двоични числа*, като на всеки отделен символ от азбуката на някои говорим език отговаря точно определена двоична стойност. Тази стойност се нарича още код на символа. За представяне на символите е налице строго определена

□ стандартизация, като за всеки символ обикновено се ангажира един или два последователни байта със стойност съответстваща на кода на символа. В практиката на компютърните технологии и особено след появата на персоналните компютри се преминава към използването на няколко основни системи за кодиране на символите. Те са приети и като световен стандарт.

*Международният ASCII код (American Standard Code for Information Interchange)* е първият, който е възприет за представяне на символите. Той е предложен още в средата на шестдесетте години на миналия век и първоначално е бил предвиден за кодиране на 128 символа от двете азбуки – латиница и кирилица и някои специални символи. В този код се използват само седем бита от байта, като основната му цел е да се въведе единен стандарт за връзка и обмен на информация между различните устройства. За всеки символ от азбуките – буква, цифра или някакъв друг знак е присвоена точно определена стойност. Тя се записва в устройствата под формата на двоично число. Пълният набор на символите и съответстващите им стойности се нарича кодова таблица и в конкретния пример за *ASCII* кода тя е седем битова. По-късно, с навлизането на персоналните компютри в началото на осемдесетте години, фирма *IBM (International Business Machines)* разширява кодовата таблица до осем бита (един байт). По този начин става възможно да се разграничат най-много до 256 символа, т.е. максималната стойност от един байт. В този случай към двете азбуки се включват и някои други, които обхващат вече определен регион. Разширената *ASCII* таблица включва всичките 128 знака от стандартната *ASCII* кодова таблица както и още 128. Въпреки тези допълнителни знаци, в някои езици има символи, за които дори тези 256 знака не достигат. Поради тази причина *ASCII* таблицата има няколко разновидности, основната цел на които е да обхване повече регионални знаци и символи по света.

*ANSI кода (American National Standards Institute)* представя стандартните кодови таблици на *Windows*. Символите в *ANSI* кода включват стандартните символи от *ASCII* таблицата плюс няколко



разширения. Това е също осембитов код, който е свързан с появата на първите операционни системи с графичен потребителски интерфейс. Неговата основна цел е била да отговори на големият по него време бум на развитие на персоналните компютри и крещящата нуждата от обмен на информация. Между ASCII и ANSI кода няма пряка връзка и идентичност, т.е. едни и същи символи в различните кодови таблици имат различен код. Това означава, че кодът на определен символ от едната таблица може да съответства на съвсем друг символ от другата таблица. Ако например информацията е кодирана с ANSI код, а се обработва с ASCII таблица, то резултатът ще бъде несъответствие на символи, което пък ще доведе до загуба на смисъл на текста или пък загуба информация. Това е често срещан случай, когато с различни програми се чете един и същ текст.

Осембитовите *ASCII* и *ANSI* кодове до преди около едно десетилетие бяха двата основни кода, които се използваша в практиката. Те бяха в основата, както за представяне на символите вътре в устройствата на компютъра, така и за обмен на информация между различните устройства и между компютрите. Сега също се запазил принципът обмена на информацията (включително в Интернет) да се осъществява под формата на символи байт по байт. Това може би е и една от основните причини, поради които байтът е приет и за сега си остава основна информационна единица от вътрешното представяне на информацията, измерването на количество и скоростта за предаване.

*UNICODE* – универсален код е третият стандарт за кодиране на символите в компютърните технологии. Името му е образувано от думите UNIFORM и CODE и това кодиране започва масово да се въвежда след средата на деветдесетте години. Това е уникален код за всеки знак, независимо от компютърната платформа (Операционната система), независимо от програмата и независимо от говоримия език. При това кодиране за всеки отделен символ се ангажират по два последователни байта, т.е. шестнадесет бита. Основната цел на шестнадесет битовото кодиране е да се обхванат всички възможни азбуки и символи в света в един общ международен

□ стандарт за обмен на информация между компютрите. Докато при първите две кодираня броят на различните символи е ограничен до 256, то при този код вече е възможно да се представят 65536 различни символа, т.е.  $2^{16}$ . Това практически разрешава всички въпроси, свързани с многоезичната поддръжка и поради това кодът се нарича още универсален или интернационален. Unicode осигурява уникален номер (стойност) за всеки знак, независимо от платформата, независимо от програмата, независимо от говоримия език. Той е приет като стандарт от редица лидери в информационните технологии, като IBM, Microsoft, Apple и други. Unicode е задължителен за новите световни стандарти за представяне на текстова информация като XML, JAVA, WML и т.н. Поддържа се в много операционни системи, всички съвременни браузъри (програми за работа в Интернет) и редица други програмни продукти. Въвеждането на стандарта Unicode, както и наличието на редица програмни похвати за неговата поддръжката е едно от най-значимите технологични постижения в последните години.

В различните компютърни програми и операционните системи понастоящем се работи с различни системи за кодиране на символите. Тези системи са известни като кодови таблици, те поддържат различните стандарти за кодиране на символите и разработчиците на софтуер и уеб страници са длъжни да се съобразяват с тях. Съвременни кодови таблици, широко използвани сега са тези, които поддържат Unicode стандарта. Това са таблиците Unicode (UTF-8) (Unicode Transformation Format) и Unicode (UTF-16). И в двете таблици се използва кодиране на символите с променлива дължина на двоичното число. Например при UTF-8 символите от английската азбука се кодира в един байт, кирилицата в два байта, а другите по-рядко използвани знаци от разпространени по света азбуки в 3 или 4 байта. При UTF-16 всеки символ се кодира в минимум 2 байта, а по-рядко използваните символи в 3 или 4 байта. От това следва, че UTF-8 е по-икономичен стандарт за кодиране на символите. Освен тези две таблици за кодиране на символите се използват, макар и по-рядко, Western (Windows-1252), Cyrillic (Windows-1251) и други.



Неправилно подбраната, указаната или настроена таблица за кодиране на символите води до извеждане на неразбираеми символи – това може да бъде в получено по електронна поща съобщение, отворена уеб страница или пък текст в някои текстови редактор. Ако в изведената информация има неразбираеми символи от вида  $\text{Ð}\frac{1}{2}\text{Ð}^\circ$   $\text{Ð}^\circ\text{Ð}\frac{1}{2}\text{Ð}^3\text{Ð}$ ;  $\text{п}\blacksquare\text{п}\frac{\text{||}}{\text{||}}\text{я}$   $\text{┆}$ .  $\text{п}$   $\text{┆}$   $\text{┆}$   $\text{┆}$   $\text{п}\frac{\text{||}}{\text{||}}\text{я}$ ; ???????;  $\text{íèúò ùâ$   $\text{íðâ}$ ;  $\square\square\text{Э}\square\square?\square\square$  и други подобни, то това означава, че кодовата таблица е неправилно подбрана или автоматично установена от софтуера. По принцип програмите автоматично определят коя таблица да използват, но ако това не се случи, то потребителят може да опита да прекодира информацията. Например при отваряне на Интернет страници смяната на таблиците може да се осъществи посредством командата Encoding. За Internet Explorer тя е достъпна от меню View, а за Google Home от Tools. За прекодиране на текстова информация са разпространени и редица други средства. В някои от програмите за текстообработка те съществуват и като вградени.

Като обобщение на разглежданите въпроси, свързани с вътрешното представяне на информацията може да се каже, че макар и различна по вид, информацията записана в компютрите се интерпретира от устройствата по един и същи начин. Явна разлика между символите и числата вътре в устройствата на компютъра няма. Там те просто са единици и нули, обединени в отделни байтове, които програмното осигуряване разграничава, като числа, букви, цифри, управляващи команди и така нататък. Разграничаването се осъществява от самата програма, то е във функция от логиката на изпълняваната задача и в това именно се заключава единния информационен подход предложен от Джон Фон Нойман и коментиран в първа глава на това издание. В този аспект количеството информация представена или поместена върху дадено устройство от компютъра зависи от много фактори, но се определя и оценява единствено от броя на байтове, които тя заема. По същият начин стои и въпросът с оценка на скоростта за обмен на информацията (предаване/приемане). Тук



съществено значение има не вида на самата информация, а броя на байтовете предадени или приети за една секунда от различните устройства.

**Кратни единици за измерване на двоична информация** има и при вътрешното представяне в компютрите. Тук те се отличават по стойности от тези в десетичната и също представляват порядък. Порядъкът в двоичната система е числото две на степен  $10$ ,  $20$ ,  $30$  ( $2^{10}$ ,  $2^{20}$ ,  $2^{30}$ ) и т.н. за порядъци един след друг.

Първият порядък за измерване на двоичната информация е *Кбит* (*КБайт*). Често се изговаря *Килобит* (*Килобайт*), но се има предвид не 1000, а 1024. Килобит (Килобайт) е информация  $2^{10}$  или това са 1024 бита (байта). Само за сравнение може да се припомни, че в десетичната бройна система кратната единица *Кило* е  $10^3$ , което е точно хиляда. Правилното е да се изговори Кбит, а не Килобит, макар, че това не се спазва и изговарянето по-често е Килобит. Същото се отнася и за порядъка Кбайт. Независимо от начина на изговаряне, следва винаги да се има предвид, че това не е 10 на степен 3, 6, 9 и т.н, а 2 на степен 10, 20, 30 и т.н.

Вторият порядък *Мбит* (*МБайт*), често изговарян Мегабит (Мегабайт) е количеството информация  $2^{20}$  или това са 1024 умножени по 1024 бита (байта), т.е. малко над един милион. В десетична бройна система този порядък е  $10^3$  или точно един милион.

Следващият по големина порядък е *Гбит* (*ГБайт*). Тук количеството информация вече е  $2^{30}$  или  $1024 \cdot 1024 \cdot 1024 = 1\ 073\ 741\ 824$  бита (байта), което е над един милиард. В десетична бройна система това е  $10^9$  или точно един милиард;

Порядъкът, който вече се използва е *Тбит* (*ТБайт*) – в практиката се изговаря Терабит (Терабайт). Това е доста голяма информационна единица и все още ограничено се прилага. Счита се, че в близко бъдеще тя ще стане нещо естествено и ще измести по-малките. Количеството информация в 1 *Тбит* (*ТБайт*) е  $2^{40}$  или  $1024 \cdot 1024 \cdot 1024 \cdot 1024$  бита (байта). В десетична бройна система това е точно един трилион.



За измерване на големи информационни масиви вече започна да се използва и следващият порядък, а именно Петабит (Петабайт). Това е двоична стойност  $2^{50}$  и в близко бъдеще тя все по-често ще се среща при оценка на количеството информация.

**Сервизно ниво на представяне на информацията** е междинното или спомагателно стъпало, което се използва при записване на двоичните стойности. На това ниво действия с двоичните числа от устройствата на компютъра не се извършват. То се използва основно от специалисти, програмисти и инженери или напреднали потребители на компютърни технологии. За представяне на информацията на това ниво се прилагат позиционни бройни системи, които са кратни на двоичната. Наричат се още позиционни системи на междинно или сервизно ниво и често се цитират в инструкциите по експлоатация на компютрите или тяхното програмно осигуряване. Основната цел при използването на подобни бройни системи е възможността една двоична стойност от вътрешното представяне на информацията да се запише в съкратена форма с по-малък брой на разрядите. За реализиране на това в практиката се използват основно две позиционни бройни системи. Това е осмичната  $2^3 = 8$  и шестнадесетичната  $2^4 = 16$ . Двете бройни системи са кратни на двоичната.

*Шестнадесетичната бройна система* към момента има най-голямо разпространение в практиката на компютърните технологии. Тя се представя от 16 цифри –  $0, 1, 2, \dots, 9, A, B, C, D, E$  и  $F$ , и има основа  $q = 16$ . Стойностите на числата в тази система се определят в съответствие с правилата за образуване на стойностите в коя да е позиционна числова система. Първите десет цифри от посочените съвпадат с тези от десетичната система, а останалите се означават със символите  $A, B$  и така нататък до  $F$ . Означаването със символи е въведено, за да няма двусмислие с числата  $10, 11$ , и т.н.  $15$ . Стойностите на символите до цифрата  $9$  са идентични с тези в десетичната бройна система, а за останалите те са съответно  $A = 10, B = 11, C = 12, D = 13, E = 14$  и  $F = 15$ .



Шестнайсетичната бройна система има редица особености, като една от най-съществените е възможността четирибитово двоично число да се запише с една шестнайсетична цифра. Тази особеност лесно се доказва и е характерна не само за тази бройна система, а и за всички останали, които са кратни на двоичната. Така например в осмичната числова система с една осмична цифра (0, 1, 2, ..., 7) се представя трибитово двоично число. Тази система е използвана преди години и в момента не намира приложение в компютрите и програмирането. Тя е била актуална при големите ЕИМ или в навечерието на съвременните компютри.

Замяната на четири битово двоично число с един шестнайсетичен разряд е една от най-съществените особености, които са и предпоставката за използването на шестнайсетичната бройна система в съвременната компютърна практика. Например двоичното число 0101, чиято стойност е пет се замества с шестнайсетичната цифра 5; числото 1010, чиито десетична стойност е 10 се замества с шестнайсетичната цифрата А; дванайсет битовото двоично число 00111111000 се замества от триразрядното шестнайсетично число 3F8 и т.н. Това преобразуване на двоичните числа често се нарича и двоично-шестнайсетично кодиране. При записване на двоичните стойности със шестнайсетични цифри първата стъпка е поредицата от двоични цифри да се раздели на групи от по четири разряда от дясно на ляво, като най-лявата, ако е с по-малко от четири разряда се допълва от дясно на наляво с нули. След това всяка четворка от двоични битове са замества с шестнайсетична цифра, чиито стойност е стойността на цифрата в четирите бита.

Очевидно е, че използването на шестнайсетични записи на двоичните стойности дава възможност да се получи значителна икономия в броя на използваните разряди и по-голяма прегледност при тяхното записване и възприемане от потребителите. Икономията, обаче е само при записа. Вътрешното представяне си е отново двоичната форма, като всяка шестнайсетична цифра ще ангажира по четири бита от устройствата на компютъра.

Като конкретни примери за използване на шестнайсетичната система в практиката може да се посочат начините за задаване на



адресите на устройствата в Интернет и компютърните мрежи. Тези адреси, наричани MAC адреси, представляват 48 битово двоично число, което се изразява, записва и представя на потребителя посредством 12 шестнайсетични цифри, обединени по двойки, с разделител помежду им тире или двоеточие, както следва:

**0100 0100 1010 1110 0001 1101 0100 1111 0110 1110 0011 1101**

**88-AE-1D-8F-6E-3D**

В първият ред от примера е показан записа на адреса в двоична бройна система, като за по-лесно възприемане между всяка четворка битове от дясно на ляво е оставен по един интервал. На вторият ред под него е шестнайсетичния запис. Всяка група от по четири двоични разряда се замества с еквивалентната стойност на шестнайсетичната цифра – от ляво на дясно съответно 0100 с 8, отново 0100 с 8, 1010 с А, 1110 с Е и т.н. 1101 най-вдясно с D.

Освен приложението на шестнайсетичната система, на междинно ниво се използват и други спомагателни записи за изразяване на различни двоични стойности. Такива са например двоично-десетичното кодиране, десетично-точковата нотация и други.

Двоично-десетичното кодиране BCD-кодиране (Binary Code Decimal) позволява всяка десетична цифра да се запише посредством нейният двоичен еквивалент. Например десетичното число 678,73 в двоично-десетичен запис ще бъде записано като 0110 01111000,0111011. Може да се каже, че представянето на десетичните цифри чрез техните двоични кодови комбинации, е начин на изразяване на десетичните числа посредством машинни средства.

Десетично-точковата нотация (някъде цитирана и като двоично-точкова) се използва при записване на Интернет (IP) адреси, които представляват 32 разрядно двоично число (версия на адресите в Интернет – IPv4). Адресите ще бъдат разгледани при изясняване на мрежите в това пособие. При десетично-точковата нотация двоичното число (в случая адрес) се разделя на четири байта, на-



речени още октети. Отделните октети в записа се разделят един от друг с точка. След това всеки байт се замества с неговата стойност записана в двоична бройна система – например, ако байтът е 11111111, то той се замества с неговата десетична стойност, която в случая е 255. По този начин четирите десетични числа, разделени помежду си с точка представляват посочения Интернет адрес (десетично-точковата нотация), както следва:

**1100000000.10101000.00000001.00010100**

**192.168.1.20**

В първият ред от примера е показан 32 битовият Интернет (IP) адрес представен, посредством четири байта разделени помежду си с точка. На вторият ред са показани четири десетични числа, също разделени с точка. Всяко от десетичните числа е еквивалентната стойност, която се съдържа в съответният байт – от ляво на дясно 11000000 е със стойност 192; 10101000 е със стойност 168; 00000001 е със стойност 1 и 00010100 е със стойност 20.

В заключение може да се отбележи, че има и други примери от практиката, в които се използва междинно представяне на двоичната бройна система. Това са таблици за кодиране, характерни адреси на интерфейси от устройствата в компютрите и други.