

# Математически и логически основи на компютърните системи

## 1. Математически основи на компютърните системи

### 1.1 Бройни системи.

Едно от най-древните математически открития на човека са числата. Много преди да се научат да четат и пишат, на хората им се е налагало да броят. Първите помощници на човека в тази дейност са били пръстите на ръцете му. По този начин до голяма степен природата е предопределила начина на броене и системата на образуване на числата (бройната система).

**Бройна система** се нарича способът за изобразяване на числата с помощта на ограничен брой символи, имащи определени количествени значения. Символите, с които се изобразяват числата, се наричат цифри. Наборът от символи (цифри) се нарича азбука на бройната система. Бройната система съдържа съвкупност от правила и съглашения за представяне на числата посредством цифри.

Бройните системи биват позиционни и непозиционни. В позиционните системи всяка цифра има определено тегло, зависещо от позицията на цифрата в последователността, изобразяваща числото. Позицията на цифрата се нарича порядък (разряд). Броят на цифрите, които съдържа бройната система, се нарича основа на бройната система. Едно число може да бъде представено в различни бройни системи. Във всяка от тях то трябва да се композира по един единствен начин. Съществуват алгоритми (правила) за преобразуване на числата от една бройна система в друга.

#### Десетична бройна система.

Азбуката на десетичната бройна система съдържа десет знака (цифри), използвани за представяне на числата: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Числото 10 играе изключително важна роля в структурата на бройната система. То се явява основа на бройната система. Изобретяването на графичния запис на числото 10 и системата за представяне на числата с основа 10 е едно от най-големите открития в развитието на математиката.

Едно произволно число може да бъде записано в десетична бройна система по един единствен начин:

$$35187 = 3 \cdot 10^4 + 5 \cdot 10^3 + 1 \cdot 10^2 + 8 \cdot 10^1 + 7 \cdot 10^0.$$

В общия случай, всяко число  $N$  може да се представи във вида:

$$N = a_k \cdot 10^k + a_{k-1} \cdot 10^{k-1} + \dots + a_1 \cdot 10^1 + a_0 \cdot 10^0.$$

където  $a_k, a_{k-1}, \dots, a_1, a_0$  са десетични цифри от азбуката на бройната система.

#### Двоична бройна система

По аналогичен начин, както при десетичната бройна система, би могло да се разработи бройна система с друга основа, различна от 10. Колкото по-малко число е основата на бройната система, толкова по-малко символи съдържа азбуката на бройната система. Най-простата бройна система е двоичната бройна система. Азбуката на двоичната бройна система се състои от два символа. Естествено е за представяне на цифрите в двоичната бройна система да се използват символи от десетичната бройна система - в случая 0 и 1. Тогава едно число в двоична бройна система би се представяло само с цифрите 0 и 1, например 10010011101. Тъй като цифрите 0 и 1 са цифри и от десетичната бройна система, за да е ясно в каква бройна система се записва числото, основата на бройната система се записва като индекс към числото:  $10111001001_{(2)}$ .

Едно двоично число може да се представи по следния начин:

$$10111001001_{(2)} = 1 \cdot 2^{10} + 1 \cdot 2^8 + 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^4 + 1 \cdot 2^0.$$

Ако се извършат действията в горния запис и се сумират отделните събираеми, ще се получи десетичният еквивалент на представеното двоично число. Това всъщност е и алгоритъмът за преобразуване на двоично число в десетично.

Обратното преобразуване, от десетично в двоично число се извършва на основата на следните разсъждения.

Нека  $N$  е десетично число, което може да се представи във вида:

$$N = a_k \cdot 10^k + a_{k-1} \cdot 10^{k-1} + \dots + a_1 \cdot 10^1 + a_0.$$

където  $a_i$  са цифри от десетичната бройна система:

$$a_i \in \{0,1,2,3,4,5,6,7,8,9\}; i = 0,1,2,\dots,k; a_k \neq 0$$

В двоична бройна система същото число би изглеждало по следния начин:

$$N = b_m \cdot 2^m + b_{m-1} \cdot 2^{m-1} + \dots + b_1 \cdot 2^1 + b_0.$$

където  $b_i$  са цифри от двоичната бройна система:

$$b_i \in \{0,1\}; i = 0,1,2,\dots,m; b_m \neq 0$$

Трябва да се уточни, че в случая на десетично и двоично представяне на едно число, броят на двоичните цифри  $m$  е по-голям от броя на десетичните цифри  $k$  (повече от три пъти). Това се дължи на различното съдържание на отделните цифри (при десетичните числа всяка цифра може да има 10 различни стойности, а при двоичните - 2 стойности).

Задачата на преобразуване на десетичното число в двоично се състои в определяне на цифрите  $b_i$ , представящи десетичното число в двоичен формат. Ако се раздели числото  $N$  на 2 се получава:

$$N/2 = N_1 + b_0/2,$$

където  $N_1 = b_{m-1} \cdot 2^{m-1} + b_{m-2} \cdot 2^{m-2} + \dots + b_2 \cdot 2^1 + b_1$  е частното от делението на числото  $N$  на 2. Цифрата  $b_0$  се явява остатък от целочисленото деление на десетичното число  $N$ . Остатъкът от делението  $b_0$  се явява същевременно и последна цифра от двоичното представяне на числото. При по-нататъшното прилагане на правилото към числото  $N_1$ , остатъкът ще представлява предпоследната цифра от двоичното представяне на  $N_1$ . Продължението на процеса води до определяне на следващите цифри в двоичното представяне на числото, докато се получи цялото двоично представяне на числото.

Ето един конкретен пример на преобразуване на десетично число (73) в двоично:

$$\begin{aligned} 73/2 &=> 36 + 1 & b_0 &= 1 \\ 36/2 &=> 18 + 0 & b_1 &= 0 \\ 18/2 &=> 9 + 0 & b_2 &= 0 \\ 9/2 &=> 4 + 1 & b_3 &= 1 \\ 4/2 &=> 2 + 0 & b_4 &= 0 \\ 2/2 &=> 1 + 0 & b_5 &= 0 \\ 1/2 &=> 0 + 1 & b_6 &= 1 \end{aligned}$$

Двоичното число, съответстващо на десетична стойност **73 е 1001001<sub>(2)</sub>**

Проверка за верността на това преобразуване е обратното преобразуване:

$$1001001_{(2)} = 1 \cdot 2^6 + 1 \cdot 2^3 + 1 \cdot 2^0 = 64 + 8 + 1 = 73.$$

Съответствието между десетичните и двоични числа за началото на числовото множество (първите 16 числа) е дадено в таблицата (нулите в началото на двоичните числа са незначещи и могат да не присъстват в запис на числата):

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

От таблицата се вижда механизмът на образуване на двоичните числа. Начинът е същият както при десетичните числа, но тук се комбинират само две цифри вместо 10. Следващото двоично число 16 съдържа 5 цифри - 10000.

Двоичната система е най-простата бройна система. Тя е и най-лесна за използване в изчислителни системи. Устройствата, които представят числата, трябва да имат две устойчиви състояния, отговарящи на двете цифри от системата 0 и 1. Това свойство на двоичната бройна система е причина в съвременните изчислителни системи да се използва само двоична информация (информация представена чрез двоични цифри).

## Други бройни системи - Тройна бройна система

Азбуката на троичната бройна система съдържа цифрите 0, 1 и 2. Преобразуването на числата от троична в десетична бройна система и обратно се извършва по същите общи правила, както при двоичната бройна система. Това е показано посредством пример за преобразуване на десетичното число 135 в троично (извършва се последователно деление на 3 и се разглежда остатъкът от делението):

$$135_{(10)} : 3 \Rightarrow 45 \Rightarrow 15 \Rightarrow 5 \Rightarrow 1 \Rightarrow 0$$

$$\text{остатък} \quad 0 \quad 0 \quad 0 \quad 2 \quad 1$$

Троичното число е  $12000_{(3)}$ .

Обратното преобразуване се извършва по общото правило:

$$12000_{(3)} = 1 \cdot 3^4 + 2 \cdot 3^3 + 0 + 0 + 0 = 81 + 54 = 135_{(10)}$$

Троичната бройна система е най-икономичната бройна система. Тя обаче не е намерила приложение в компютърните системи, тъй като е по-сложна за техническа реализация (по трудно се конструират устройства с три устойчиви състояния).

### - Шестнадесетична бройна система.

Шестнадесетичната бройна система има азбука, която съдържа 16 цифри. Като графични знаци се използват цифрите на десетичната бройна система, допълнени с главни букви от латинската азбука: **0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F**. Шестнадесетичната бройна система се използва в компютърните технологии, главно за компактно записване на двоични числа. Това е така, защото преобразуването на числа от двоична в шестнадесетична бройна система и обратно се извършва много лесно. Причина за това е, че шестнадесетичната бройна система е кратна на двоичната ( $16 = 2^4$ ).

Логическият анализ на двете системи показва, че при преобразуване на двоично число в шестнадесетично, всяка четворка двоични цифри се заменя със съответна шестнадесетична цифра. При обратното преобразуване всяка шестнадесетична цифра трябва да се замести със съответна четворка двоични цифри. Съответствието между двоичните и шестнадесетични цифри е дадено в таблицата (същата както по-горе, но с шестнадесетични цифри):

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

Разделянето на двоичното число на четворки започва отзад напред и за последната група (първите цифри на двоичното число) се допълват незначещи нули, ако е необходимо. Ето един пример на преобразуване на число от двоична система в шестнадесетична  $101100100100111110_{(2)}$ :

$$0010 \ 1100 \ 1001 \ 0011 \ 1110_{(2)}$$

$$3 \ C \ 9 \ 3 \ E - \text{шестнадесетичното число е } 3C93E_{(16)}$$

Този лесен начин за преобразуване на числата прави шестнадесетичната бройна система основна за комуникация между компютърната система и системните и приложни програмисти, които създават програмното осигуряване. Вместо да изпраща съобщения, съдържащи двоични числа с голям брой цифри (за адресите в паметта например), компютърната система изпраща компактни шестнадесетични числа, които много по-лесно могат да бъдат анализирани от потребителите. В този смисъл, шестнадесетичната бройна система се явява помощна, използвана предимно при разработване на програмното осигуряване.

### 1.2 Двоична аритметика.

Двоичната бройна система е основна за представяне на информацията в компютърните системи. Тъй като информацията трябва да се обработва, от съществено значение е как се извършват основните аритметични операции, когато числата са представени в двоична бройна система. Тъй като правилата за представяне на числата в двоична и десетична бройна система са еднакви, то и аритметичните операции се извършват по същите правила. Това се вижда от показания пример:

$$\begin{array}{r} 1001101 \\ + \quad 101011 \\ \hline \end{array} \quad \begin{array}{l} \rightarrow \text{десетичен еквивалент } 77 \\ \rightarrow \text{десетичен еквивалент } 43 \end{array}$$

---

1111000                   -> десетичен еквивалент 120

Събирането на две двоични числа се извършва, като се започне от последните разреди, както е при десетичното събиране. В случая последните цифри на двете числа са единици и при събирането  $1 + 1$  резултатът е 2, което в двоична бройна система се записва като 10. По аналогия с десетичното събиране в резултата трябва да се запише 0 и да се отчете пренос 1 към събирането за следващите разреди.

При изваждане, когато изважданата цифра е по-голяма от цифрата, от която се изважда, е необходимо да се вземе стойност от предишния разред, както това се прави при десетичното изваждане.

.	.	
1001101		-> десетичен еквивалент 77
- 101011		-> десетичен еквивалент 43
<hr/>		
100010		-> десетичен еквивалент 34

Умножение и деление на двоични числа се извършва по правила, аналогични на тези, които важат за десетичните операции.

## 2. Представяне на информацията в двоичен код

Информацията може да се дефинира като съвкупност от сведения за околния свят и протичащите в него процеси, възприемани от човека или специализирани устройства за обезпечаване на целенасочена дейност. По своята природа информацията бива: числова, текстова, графическа, звукова, видеоинформация и други.

Оценка за количеството информация може да се направи по различен начин. Когато се отнася за информация в двоичен вид може да се използва формулата на Шенон:

$H = \log_2 N$  , където **H** е количеството информация, а **N** – броят на равновероятни алтернативни състояния на обекта, носител на информацията.

При компютърните системи всяка информация се представя в двоична форма (посредством две алтернативни състояния – 0 и 1). Различните типове информация се кодират (представят) в двоична форма по определени правила.

Най-малката порция информация се явява отговорът на елементарен въпрос, изискващ два отговора – ‘да’ или ‘не’. Това графически може да се представи с една от двете цифри от двоичната бройна система – 0 и 1. Информацията, съдържаща един от двата възможни отговора ‘да’ или ‘не’ или една от двете цифри от двоичната бройна система се нарича ‘бит’ (bit). По аналогия с терминологията от квантовата механика, това количество информация може да се смята за един квант (не съществува по-малко количество информация).

Един бит информация може да бъде представена по различен начин в техническите устройства – в електронните устройства посредством нулево ниво на напрежение и някакво друго ниво; затворена или отворена електрическа верига; намагнитен или ненамагнитен участък от магнитен материал и т.н. По този начин чрез устройства, осигуряващи две устойчиви състояния, може да се представя двоична информация.

Двоичната информация се комплектова като съвкупност от битове. Различните типове и количество информация заемат различен брой битове. В компютърните системи битовете с двоична информация се групират в елементи, наречени байтове (B - byte). Един байт се състои от 8 последователно разположени бита. Производни единици на байта са килобайт (1 KB = 1024 B), мегабайт (1 MB = 1024 KB), гигабайт (1 GB = 1024 MB) и т.н. Производните единици KB, MB, GB, TB (терабайт) се различават малко от стандартните измервателни единици - вместо 1000 като множител за производните единици се използва близкото число  $2^{10} = 1024$ . Използва се такъв начин на образуване на производните единици поради специфичната природа на двоичните числа (необходима е кратност към основата на бройната

система - 2). Така най-икономично се управляват апаратните ресурси на компютърните системи, при комплектоване и разпределение на паметта.

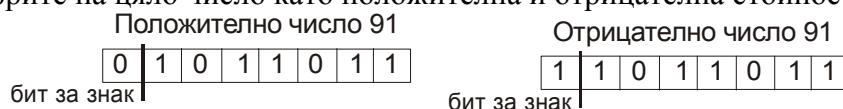
## 2.1 Представяне на числова информация

В компютърните системи се използва два вида числова информация: натурални (цели) числа и реални (веществени) числа. Съществува и една трета форма – двоично-десетични числа, които се използват обикновено за обработка на икономическа информация.

### Цели числа

Положителните цели числа се представят в двоична форма чрез преобразуване по начина, показан по-горе. В зависимост от големината на числото за представянето му са необходими различен брой цифри (битове). В изчислителните системи за представяне на числата обикновено се отделя фиксиран брой битове (байтове). Това означава, че в областта определена за представяне на дадено число може да се изобрази стойност в ограничен диапазон. Максималната стойност (число), която може да се представи в област с  $n$  бита се определя от броя на различните комбинации от '0' и '1', които могат да се съставят с  $n$  двоични цифри. Тъй като броят на комбинациите от две състояния на  $n$  елемента е  $2^n$ , то с  $n$  бита може да се представи цяло число в интервала  $0 \div 2^n - 1$  (числото '0' е една от възможните комбинации и максималната стойност е с 1 по-малка). Така в един байт (8 бита) може да се представи цяло число в интервала  $0 \div 2^8 - 1$  или  $0 \div 255$ .

Представянето на отрицателните цели числа изисква въвеждане на специален бит (разред) за знак на числото. Приема се, че ако в бита за знак има стойност 0, числото е положително, а ако е 1 – отрицателно. Така например, ако за цяло число се отделя 1 байт (8 бита), стойността на числото се записва в последните 7 бита, а първият бит се използва за знак. Тогава максималното цяло число със знак, което може да се запише в 1 байт е  $2^7 = 128$ , а диапазонът на възможните числа е от  $-127$  до  $128$ . На фиг. 1.1 е показано разположението на цифрите на цяло число като положителна и отрицателна стойност.



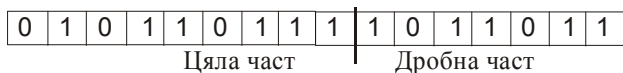
Фиг. 1.1. Представяне на цели числ

### Реални (веществени) числа

За да се представят дробни числа в двоична бройна система се използва позиционна точка (аналогична на десетичната точка в математиката). Цифрите вляво от позиционната точка представят цялата част на числото, а цифрите вдясно – дробната част. По аналогичен начин както при десетичните числа, позицията на цифрата вдясно от позиционната точка (дробните цифри) определя с коя отрицателна степен на числото 2 ( $2^{-k}$ ) трябва да се умножи съответната двоична цифра. Първата цифра след позиционната точка се умножава по  $\frac{1}{2}$ , втората – по  $\frac{1}{4}$ , третата – по  $\frac{1}{8}$  и т.н. Например числото 1011.101 в десетичен вид би изглеждало така:

$$1 \cdot 2^3 + 1 \cdot 2^1 + 1 + 1 \cdot 1/2 + 1 \cdot 1/8 = 8 + 2 + 1 + 1/2 + 1/8 = 11 \frac{5}{8}$$

В компютърните системи реалните числа се представят по два начина – с фиксирана точка и с плаваща точка. При запис на числата с **фиксирана точка** в двоичен формат се предполага точно определено място на позиционната точка в полето за запис на числата. В битовете преди позиционната точка се записва цялата част на числото, а в битовете след нея – дробната (десетична) част (фиг.1.2). Както се вижда, мястото на позиционната точка не изисква специална клетка в паметта (бит) – тя е условно разделяне на полето на две части. Частен случай на този вид запис са целите числа. При тях позиционната точка е в края на полето. В практиката този вид на представяне на реалните числа се използва, като позиционната точка се фиксира в началото на полето за запис на числата (запис на числа по-малки от 1).



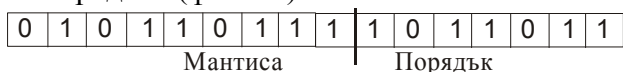
Фиг. 1.2 Представяне на реални числа с фиксирана точка

Числата с **плаваща точка** се записват във вид на мантиса  $m_a$  и порядък  $p$ .

$$N = m_a 10^p.$$

Например числото 374.25 може да се запише във вида:  $0.37425 10^3$ . В този запис  $m_a = 0.37425$ , а  $p = 3$ . Числото 0.000453 може да се запише като  $0.453 10^{-3}$  и мантисата е  $m_a = 0.453$ , а порядъкът е  $p = -3$ . Записано във вид на мантиса и порядък числото е в нормализиран формат.

Числата с плаваща точка се представят в паметта на компютърните системи, като в полето за запис на числото се фиксира част от паметта за запис на мантисата и друга част – за запис на порядъка (фиг.1.3).



Фиг. 1.3. Представяне на реални числа с плаваща точка

Мантисата и порядъкът се записват в паметта както се записват целите числа. Те трябва да имат по един бит за знак, а останалите битове от полето са за запис на стойността.

**Двоично-десетичното** представяне на числата се използва за обработка на големи масиви от данни. Когато трябва да се показват числови данни от такива масиви, се губи много време за преобразуване на числата от двоична в десетична форма. Посредством специални функционални блокове (специална аритметика) всяка десетична цифра се представя с група двоични цифри. Най-малкият брой двоични цифри, с които могат да се представят десетичните цифри, е четири. Затова при този начин на представяне, всяка десетична цифра се преобразува в двоична форма с използване на 4 двоични цифри. Ето един пример:

$$3795_{(10)} = 0011\ 0111\ 1001\ 0101_{(2)}$$

Тук преобразуването не изисква специална процедура, а просто заменяне на четворки двоични цифри с десетични цифри.

Трябва да се отбележи, че представянето на целите числа може да се разглежда като частен случай на представянето на числата с плаваща точка. Това, че в компютърните системи се използват и двата начина на представяне се дължи на следните причини:

- За икономия на памет. Не е необходимо при целите числа да се резервира памет за порядък. Освен това, устройствата за обработка на целите числа са по-прости и операциите се извършват с по-голяма скорост.

- За по-добър контрол в процеса на програмиране. Когато предварително е определен видът на различните променливи в програмата, по-лесно се откриват грешките в програмите.

## 2.2 Машинна аритметика.

Микропроцесорът в компютърните системи изпълнява прости команди при аритметичната обработка на числови данни. За целта, числовите данни се представят в подходяща форма. Въведени са няколко вида машинни кодове за представяне на числова информация.

### 1. Прав код.

Правият код на двоично число се образува от абсолютната стойност на числото в двоична бройна система и код за знака (нула или единица). Пример:

$$N_{(10)} = 11 ; N_{(2)} = 1011 ; N_{(2),п.к} = 0|1011$$

$$N_{(10)} = -15 ; N_{(2)} = -1111 ; N_{(2),п.к} = 1|1111$$

### 2. Обратен код

Обратният код на двоичните числа се образува по следното правило: обратният код на положителните двоични числа съпада с техният прав код; обратният код на отрицателните

числа съдържа '1' в бита за знак, а цифрите на числото се заменят с техните инвертирани стойности (нулите се заменят с единици и обратно). Пример:

$$N_{(10)} = 6 ; N_{(2)} = 0110 ; N_{(2),o.k} = 0|0110$$

$$N_{(10)} = -13 ; N_{(2)} = -1101 ; N_{(2),o.k} = 1|0010$$

Едно от свойствата на обратния код е че сумата на положително с обратния код на отрицателното му число дава така наречената машинна единица 1111111111. Нулата в обратен код има двойко значение: 0|0000 и 1|1111. Това е причина да не се използва обратен код в машинната аритметика, а той да се модифицира в допълнителен код.

### 3. Допълнителен код.

В допълнителен код положителните двоични числа съвпадат с техния прав код. Допълнителният код на отрицателните числа се образува, като към обратния код на числото се прибави '1' в най-младшия разред (бит). Пример:

$$N_{(10)} = 19 ; N_{(2)} = 10011 ; N_{(2),d.k} = 0|10011$$

$$N_{(10)} = -13 ; N_{(2)} = -1101 ; N_{(2),d.k} = 1|0011$$

Основни свойства на допълнителния код.

-Сумата на положително число с допълнителния код на неговото отрицателно значение дава така наречената машинна единица в допълнителен код  $ME_{дк} = 10|00000$ . Характерното в този запис е, че в битовете за знак има 10.

-Представянето на отрицателните числа в допълнителен код се явява допълнение до на правия код до машинната единица. Значението на допълнителния код може да се види от записа на последователните числа при прехода от положителни към отрицателни стойности:

Дес. число	5	4	3	2	1	0	-1	-2	-3	-4
Двоично число	0 0101	0 0100	0 0011	0 0010	0 0001	0 0000	1 0001	1 0010	1 0011	1 0100
Доп. код	0 0101	0 0100	0 0011	0 0010	0 0001	0 0000	1 1111	1 1110	1 1101	1 1100

От таблицата може да се види, че ако към някое число, представено в допълнителен код, прибавим 1 към най-младшия бит се получава предхождащото го число, независимо дали е положително или отрицателно. Това свойство на допълнителния код на двоичните числа улеснява извършването на аритметичните действия с тях.

### Сумиране и изваждане на числа в допълнителен код.

Използва се същият алгоритъм както в обикновената двоична аритметика. За целта числата трябва да са с еднаква дължина, като резултатът е със същата дължина. В този случай всеки бит на пренос в най-лявата част (където е бита за знак) се игнорира. Когато числото е с по-малък размер то се допълва отляво с незначещи нули. Действието изваждане се представя като събиране, но второто число се представя като отрицателно число в допълнителен код. Примери:

$$\begin{array}{r}
 3+2 \quad 0011 \\
 \quad \quad 0010 \\
 \hline
 \quad \quad 0101 \Rightarrow 5
 \end{array}
 \qquad
 \begin{array}{r}
 -3 \quad 1|101 \\
 \quad \quad -2 \quad 1|110 \\
 \hline
 \quad \quad -5 \quad 1|011 \Rightarrow -5
 \end{array}
 \qquad
 \begin{array}{r}
 7 \quad 0|111 \\
 \quad \quad -5 \quad 1|011 \\
 \hline
 \quad \quad 2 \quad 0|010 \Rightarrow 2
 \end{array}$$

Основното преимущество на двоичния допълнителен код е, че сумирането на всички числа (положителни и отрицателни) се извършва по един и същи алгоритъм.

В компютърните системи се използва и **модифициран допълнителен код**, който е подобен на допълнителния код, но при него се използват два бита за знак: 00| - за положителни числа и 11| - за отрицателни числа. Ако се получи препълване в полето за стойност на числата, се модифицира и комбинацията за знак – получават се комбинации 01| или 10|.

### 2.3 Представяне на символна и текстова информация.

Тъй като в компютърните системи информацията се съхранява само в двоичен код, използваните символи при съставяне на даден текст трябва да се представят като комбинация от двоични цифри (1 и 0). При натискане на клавиш от клавиатурата, който съответства на даден символ се формира определен код (scan cod) – двоично число, определящо се от поредния номер на клавиша. Номерът на натиснатия клавиш обикновено не е свързан със съответния символ. Връзката между формирания 'scan cod' и символа се

определя от специална таблица. В нея на всеки символ, използван от компютърната система, се присвоява двоично число. Когато се записва текст в паметта на компютъра, всъщност се записва последователност от двоични числа, съответстващи на символите от текста. Те не се различават по начина на записване от обикновените цели числа. Разглеждането на тази информация като текст е условно, тъй като трябва да има предварително уточнение, че това е текст, а не обикновени числа.

В началния период на развитие на компютърните системи са използвани различни принципи на кодиране на символна и текстова информация. Това е създавало големи затруднения, тъй като трудно е можело да се обменя информация между различните изчислителни машини. Поради тази причина Американският национален институт по стандартизация (ANSI) разработва специална кодова таблица ASCII (American Standard Code for Information Interchange). Тази кодова таблица намери много широко разпространение, като продължава да се използва и сега (произнася се 'ас-кий').

Когато се е разработвал този стандарт, се е смятало, че всички използвани в изчислителните машини символи не надвишават 120 броя и могат да се кодират с не повече от 7 двоични цифри:  $2^7 = 128$ . Все пак, като допълнително подсигуриране е прието символите да записват с 8 двоични цифри, което е позволявало да се кодират до 256 символа. Това всъщност е и причината битовете в паметта да се групират по 8 и да образуват байтове. Всъщност 1 байт е поле в паметта, където може да бъде записан един символ. В таблица 1.1 е показана част от кодовата таблица ASCII с кодовете на основните символи.

Таблица 1.1. Основни символни кодове в ASCII кодова таблица

Символ	ASCII код- двоичен	Десетичен код	Символ	ASCII код- двоичен	Десетичен Код	Символ	ASCII код- двоичен	Десетичен Код
(Пробел)	00100000	32	A	01000001	65	c	01100010	98
!	00100001	33	B	01000010	66	d	01100011	99
“	00100010	34	C	01000011	67	e	01100100	100
#	00100011	35	D	01000100	68	f	01100101	101
\$	00100100	36	E	01000101	69	g	01100110	102
%	00100101	37	F	01000110	70	h	01100111	103
&	00100110	38	G	01000111	71	i	01101000	104
‘	00100111	39	H	01001000	72	j	01101001	105
(	00101000	40	I	01001001	73	l	01101010	106
)	00101001	41	J	01001010	74	m	01101011	107
*	00101010	42	K	01001011	75	n	01101100	108
+	00101011	43	L	01001100	76	o	01101101	109
,	00101100	44	M	01001101	77	p	01101110	110
-	00101101	45	N	01001110	78	q	01101111	111
.	00101110	46	O	01001111	79	r	01110000	112
/	00101111	47	P	01010000	80	t	01110001	113
0	00110000	48	Q	01010001	81	u	01110010	114
1	00110001	49	R	01010010	82	v	01110011	115
2	00110010	50	S	01010011	83	w	01110100	116
3	00110011	51	T	01010100	84	x	01110101	117
4	00110100	52	U	01010101	85	y	01110110	118
5	00110101	53	V	01010110	86	z	01110111	119
6	00110110	54	W	01010111	87	{	01111000	120
7	00110111	55	X	01011000	88	}	01111001	121
8	00111000	56	Y	01011001	89			
9	00111001	57	Z	01011010	90			
:	00111010	58	[	01011011	91			
;	00111011	59	\	01011100	92			
<	00111100	60	]	01011101	93			
=	00111101	61	^	01011110	94			
>	00111110	62	_	01011111	95			
?	00111111	63	a	01100000	96			
@	01000000	64	b	01100001	97			



Както се вижда, основните символи, използвани в компютърните системи, са кодирани в първата половина на кодовата таблица (номера до 128). Това са всички възможни двоични комбинации, за които първият бит е 0. С номера по-големи от 128 (старшият бит 1) са кодирани някои специални символи като символи от немската и френската азбука и редица графични символи за получаване на прости графични обекти (правоъгълници, таблици и други).

Символите с номера над 128 се използваха дълго време за вграждане на други символни вериги (азбуки) в компютърните системи. По този начин се използваше и кирилицата при създаване на документи и програмно осигуряване на български език. За целта част от символите от стандартната ASCII кодова таблица с номера над 128 се заменяха с графични знаци от кирилицата. С появата на операционната система Windows, стана възможно използването на различни текстови шрифтове, което изискваше замяна на графичните символи в съответния шрифт с такива на кирилица. И сега съществуват шрифтове с възможност за писане на кирилица и такива, на които не е извършена замяна на съответните графични символи. Това е причината някои текстове писани на кирилица, да изглеждат напълно неразбираеми, когато се разглеждат на друг компютър.

### **Unicode.**

Ако на планетата се използваше само един език (английски), тогава кодовата таблица ASCII щеше да бъде напълно достатъчна и вероятно щеше да се използва много дълго време. Съществуват обаче множество езици, които при съвременното развитие и използване на компютърните системи изискват много повече от старата ASCII таблица и пригаждането ѝ към различни азбуки. Особено големи са изискванията на литературата – тя използва много характерни особености на различните азбуки като ударения, транскрипции и други.

Решението на този въпрос (може би също временно) е въвеждането на системата за кодиране Unicode. Тя се използва вече няколко години и с нея работят новите операционни системи Windows 98, Windows 2000, Windows XP, Linux и други. Същественото при тази система за кодиране на текстови символи е, че за представянето на един символ се използва не един байт (8 бита), а 2 байта (16 бита). Това означава, че в Unicode могат да се кодират до 65536 различни символа ( $2^{16}$ ). Кодът на всеки символ съдържа старши байт (8 бита), който определя веригата (азбуката) и младши байт, определящ конкретния символ в дадената азбука. За всяка азбука се отделят по 265 символа, където се кодират както основните символи, така и специални символи (ударения, графични означения и т.н.).

Кирилицата в тази система е номерирана с номер 204 като код на азбуката (този номер е в първия байт на всеки символ от кирилицата). Разбира се, различните програмни системи използват специфични методи за икономии на памет (постоянното повторение на кода на азбуката ако текстът е само на един език може да се избегне).

Над 20000 символа в стандартните шрифтове в Unicode системата са отделени за йероглифи от китайското и японско писмо. Съвременните операционни системи се разпространяват с голям набор от шрифтове в Unicode кодова система, така че проблемът с различните езикови особености може да се смята до голяма степен решен. Това не се отнася за текстове написани в старата кодова система, които са все още твърде много и дълго време ще поддържат ASCII кодовата система.

## **2.4 Представяне на графична информация.**

Съвременните компютърни системи могат да обработват графични изображения, картини, анимация, видео. Тези възможности на компютрите всъщност ги направиха толкова популярни и широко разпространени. В тази връзка е от съществено значение как се представя едно графично изображение в паметта на компютърните системи и как се обработва графична информация.

Съществуват две категории графични изображения: растерна графика и векторна графика.

- **Растрерна графика.** При растрерните методи изображението се представя като съвкупност от отделни елементи от картината, наречени пиксели (pixel – съкращение от picture element). Казано по-просто, графическото изображение се кодира във вид на последователност от битове, които описват редовете от пиксели. Ако изображението е черно-бяло, за описанието на един пиксел бихме могли да използваме 1 бит (стойност 0 ако пикселът е бял и 1 – ако е черен). Такава картина би била много проста и неточна, тъй като не се отчита интензивността на черното в отделните пиксели. Още по-сложно става представянето на цветни изображения. Представянето на графичните изображения чрез съвкупност от пиксели се нарича битова карта на изображението (bitmap).

Повечето графични устройства (дисплей, скенер, цветен печат и други) разделят цвета на даден пиксел на три основни цвята – червен, зелен и син (RGB – red, green, blue). При стандартно графично изображение интензивността на всеки цвят се задава с помощта на 256 нива (стойности). Следователно, за всеки основен цвят на пиксела е необходим 1 байт (8 бита) памет. Ако изображението се съхранява в растрер (разрешаваща способност) от 1280 реда x 1024 пиксела на ред (1280 x 1024) ще са необходими няколко мегабайта (MB) памет за съхраняване на изображението.

Съществуват различни методи за съгъстяване (компресиране) на информацията за графичните изображения. Типични формати за представяне на графика са GIF (Graphic Interface Format), JPEG () и други. Изображенията могат да се представят и като определена комбинация от цветове (16, 32, 256 цвята). В тези случаи от всички възможни комбинации на интензивност на трите основни цвята се подбира набор от определен брой основни цвята, с които се представя дадена картина (всеки пиксел се идентифицира с някой от тези основни цветове).

Основен недостатък на растрерната графика е невъзможността за пропорционално изменение на картината. При увеличаване на размера на картината се получава зърнитост, тъй като размерите на отделните пиксели се увеличават и това се отразява на общия вид на изображението.

Типични програмни продукти, които обработват растрерни графични изображения, са Photoshop, Paint, Corel Draw и други.

- **Векторна графика.** Друг способ за представяне на графични изображения е посредством векторна графика. Изображението се представя като съвкупност от геометрични елементи (линии, окръжности, правоъгълници и други). В паметта се записват данни, които определят тези елементи (координати, радиуси, ъгли), а приложенията, които обработват графиката, построяват визуалното изображение по тези данни. Този начин на представяне на графични изображения може да се използва само за схеми, диаграми, графики. Предимството му е, че изображението заема по-малка памет и мащабирането на изображенията не представлява проблем. Така са построени и голяма част от текстовите шрифтове, което позволява лесното им мащабиране. Най-често използваните графични приложения за работа с векторна графика са AutoCAD, Corel Draw, Microsoft Visio и други.

## 2.5 Представяне на звукова информация

Най-широко приложение за кодиране на звукова информация (аудиоинформация) е получил методът на измерване на амплитудата на звуковата вълна през определени интервали и запис на получените стойности като двоични числа. За да се получи необходимото качество на звука, трябва да се използва достатъчно малък интервал на разделяне на времетраенето на звука. Основното правило е да се подбере такава честота на запис на амплитудата на звуковата вълна, която да е най-малко два пъти по-голяма от максималната звукова честота, която може да се възприеме от човешкото ухо. В съвременните цифрови системи за запис на звук (музика) се използва звуков формат с отчитане на 44100 стойности на амплитудата на звука за секунда. Всяка амплитуда се записва в 16 - битов формат (65536 нива на звукова интензивност), а ако е стереозапис, се

прилага 32 - формат. Следователно за запис на 1 секунда текст са необходими 88200 байта (172400 байта за стереозапис).

Използването на такъв формат за запис на звукова информация изисква огромна памет. Използват се различни начини за компресиране на звуковата информация, така че разходът на памет да стигне поносими нива. Един от най-популярните разработчици на формати за компресиране на графична и звукова информация е асоциацията Motion Picture Experts Group (MPEG). Напоследък голямо разпространение получи един от продуктите на тази група - MP3, при който се реализира степен на компресия на информацията 12:1.