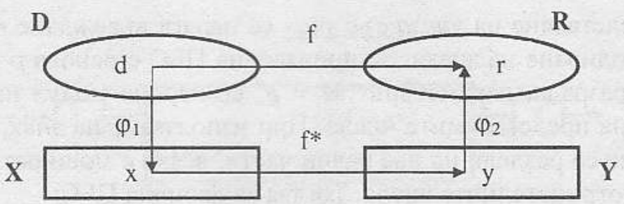
Въведение

Компютърната обработка се основава на реализацията на определени функции за преобразуване на входните данни в изходни решения. За целта обработваната информация се представя (кодира) във вид на съобщения с определена дължина (фиксирана или променлива) чрез система от средства и правила в подходяща компютърна среда. Това е свързано с две нива на функционално преобразуване (фиг. В.1):

- *математическо преобразуване на информацията* – по същество реалната (математическата) обработка на информацията е изпълнение на обобщена функция $f{:}D \rightarrow R$ за преобразуване на елементите от множеството на данните D в елементи от множеството на резултатите R;
- *компютърна реализация на обработката* – форма на конкретна реализация на математическата функция в зависимост от комплекса от апаратна част и необходимото програмно осигуряване за управление.



Фиг. В.1. Информационно поддържане при компютърната обработка

За осъществяване на компютърната обработка е необходимо да се осигури подходящо физическо представяне на реалните информационни обекти – множествата на реалните входни данни D и изходни резултати R. В резултат реалната функция $f{:}D \rightarrow R$ се заменя с нейния еквивалент $f^*\!: X \rightarrow Y$, представящ компютърната обработка във функционален смисъл. За целта се извършват допълнителни функционални преобразувания чрез помощните кодиращи функции: φ_1 – входна (кодираща); φ_2 – изходна (декодираща). Така обобщеният функционален модел на компютърната обработка може да се представи като $r = f(d) = \varphi_2\{f^*[\varphi_1(d)]\} = \varphi_2\{f^*[x]\}$, където $x = \varphi_1(d)$, $y = f^*(x)$ и $r = \varphi_2(y)$.

В компютърните системи (КС) информацията се кодира чрез двоичната азбука, което е свързано с използване на двупозиционни структурни елементи за нейното електронно представяне. Основната организационна единица в компютъра е <u>машинната дума</u>, която представлява подредена последователност от двоични разряди с фиксирана дължина n, а всеки разряд може да бъде представен чрез такъв структурен елемент. Въвеждането на машинна дума с дължина n ограничава разрядността при представяне на информацията и определя не повече от 2^n двоични комбинации. За представяне на различни данни в компютъра са въведени и производни на машинната дума (W) – полудума (HW), двойна дума (DW), четворна дума (FW).

Дължината n на машинната дума определя диапазона на представимите числа – обхват от минималната до максималната стойност, които могат да бъдат записани в n разряда. Очевидно, за цяло число без знак това са стойностите от 0 до $2^n - 1$. В общия случай едно <u>*число без знак*</u> N се кодира в двоична позиционна бройна система (ПБС) като: $N_2 = b_k.2^k + b_{k-1}.2^{k-1} + ... + b_1.2^1 + b_0.2^0$, където $b_i \in \{0,1\}$; за $i = 0 \div k$. При представяне на това число в n-разрядна машинна дума са възможни ситуациите:

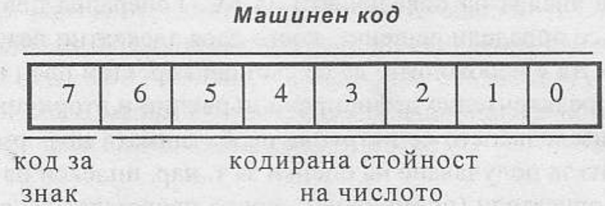
- $k \le n$ – позволява точно представяне на N_2 в разрядната решетка на машинната дума;
- $k > n$ – дължината на машинната дума е недостатъчна и част от двоичния код ще се загуби; например, при ситуацията: $N_2 = b_\kappa.2^\kappa + b_{\kappa-1}.2^{\kappa-1} + ... + b_n.2^n + b_{n-1}.2^{n-1} + ... + b_1.2^1 + b_0.2^0 = N''.2^n + N'$, частта $N''.2^n = (b_\kappa.2^{\kappa-n} + ... + b_n).2^n$ от записа на двоичното число ще се загуби, което ще въведе грешка при представянето.

При представяне на <u>*числа със знак*</u> се налага въвеждане на допълнителен разряд за кодиране на знака. За произволна ПБС с основа p и дължина на представяне n разряда, параметърът $M = p^n$ се нарича модул на системата и определя броя на представимите числа. При използване на знак, множеството на тези числа ще се раздели на две равни части, всяка с мощност $M/2$, за положителните и за отрицателните числа. Тогава за двоична ПБС:

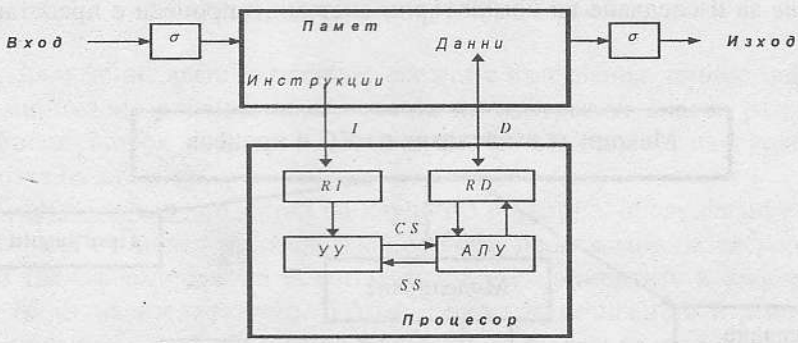
- положителните числа ($N \ge 0$) ще се представят чрез диапазона $[0, 2^{n-1}-1]$, понеже $N < M/2$;
- отрицателните числа ($N < 0$) ще се представят чрез останалите комбинации, т.е. диапазона $[2^{n-1}, 2^n-1]$, което представлява допълнението на числото $N_{доп} = M - N$.

За представяне на числа със знак в КС са въведени <u>*машинни кодове*</u>, които позволяват едновременна обработка на всички разряди от записа (знакови и числови). По този начин алгебричните действия над числата се заместват с алгебрични действия над техните машинни кодове. Машинният код на дадено число N_2 е също двоичен запис, разполаган в разрядната решетка на машинната дума с дължина n. За този случай абсолютният максимум на представимо число е $M/2 = 2^{n-1}$. За кодиране на знака се използва допълнително поле от 1 бит (обикновен код) или 2 бита (модифициран код), като знак "+" се кодира с нулева стойност, а знак "–" с единична стойност. Използват се три основни кода – прав (ПК), обратен (ОК) и допълнителен (ДК). На фиг. В.2 е показана примерна 8-битова машинна дума при представяне на числа чрез обикновен машинен код.

Компютърната система (КС) е интегрирана съвкупност от взаимодействащи си компоненти, организирани в две базови подсистеми – апаратна (hardware) и програмна (software), чрез които се реализира конкретен архитектурен модел за организация на компютърната обработка. Целта е осигуряване на ефективни нива (стойности) за множеството от системни характеристики при използване на системния ресурс. За повишаване на производителността и ефективността на компютърната обработка, както и на бързодействието на изчисленията, могат да бъдат заложени различни принципи при организацията на апаратните и програмните компоненти. Организацията на изчисленията (правилата за компютърна обработка) при традиционните КС може да се представи чрез фиг. В.3.



Фиг. В.2. Представяне на машинен код в един байт

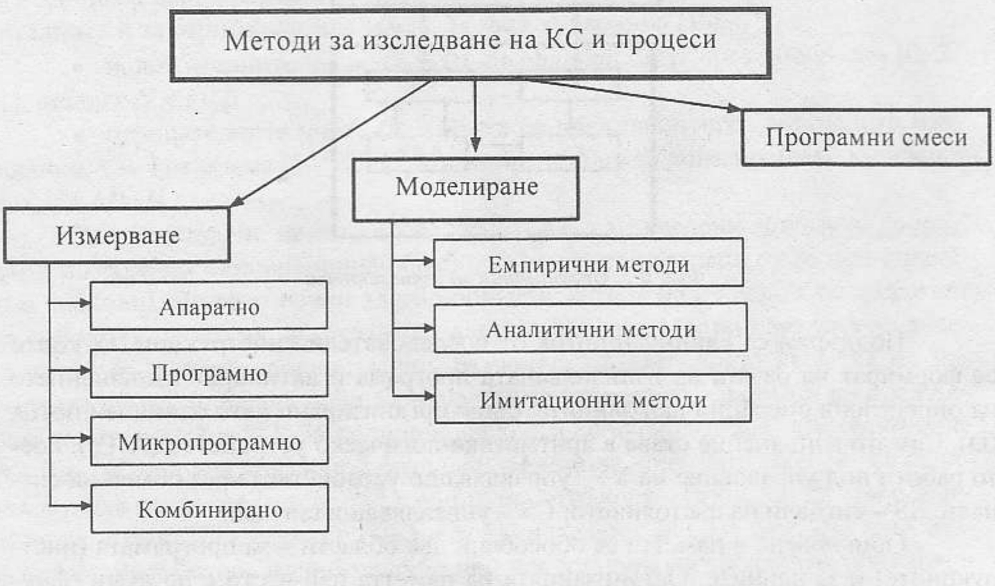


Фиг. В.3. Организация на изчисленията

Поддържа се единичен поток от последователни инструкции (I), които се формират на базата на изпълняваната програма и активират изпълнението на определени операции над данните, също организирани като единичен поток (D). Самото изпълнение става в аритметико-логическо устройство (АЛУ), което работи под управление на УУ (управляващо устройство) чрез обмен на сигнали: SS – сигнали на състоянието; CS – управляващи сигнали.

Обикновено в паметта са обособени две области – за програмата (инструкциите) и за данните. Организацията на паметта най-често е по думи (фиксиран брой байтове). За прочитането на всяка дума е необходимо определено време за достъп, свързано с адресиране и извличане на информацията. Обменът на информация с външните устройства, вкл. външните памети, се реализира чрез средствата за управление на системата за вход-изход (σ). При решаване на дадена задача (приложен процес; програма) централният процесор осъществява последователен достъп до инструкции и данни. Реалното изпълнение става на базата на машинна програма – последователност от машинни инструкции, съхранявани в паметта. Всяка една машинна инструкция се прочита и изпраща в регистъра за инструкции (RI). Нейното изпълнение води до преобразуване на входните данни в изходен резултат, съхраняван в регистъра на данните (RD).

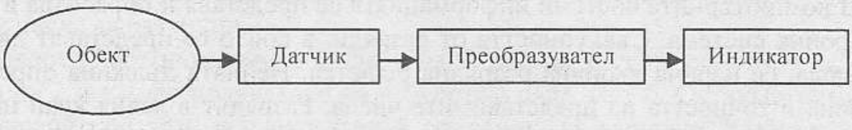
Изследването на компютърната обработка си поставя за цел определяне на количествени оценки за системни параметри на базата на проведени експерименти за анализ на поведението на КС. Генерална цел на всяко едно изследване е да се определи решение, което дава адекватни резултати при разумна цена. За целта е необходимо да се състави коректен план на експеримента на базата на предварително дефинирани първични и вторични фактори. При провеждане на изследването се натрупва необходимата информация, която се подлага на анализ за получаване на оценки за т. нар. индекси на производителността. Това са описатели (променливи), които представят аспекти на системната производителност, като например: време за изпълнение на инструкция, време за отговор, време за достъп до информационен ресурс, коефициент на използване, пропускателна способност и пр. Една примерна класификация на *методите* за изследване на компютърни системи и процеси е представена на фиг. В.4.



Фиг. В.4. Класификация на методи за изследване на КС и процеси

По същество *програмните смеси* са аналитични зависимости, които дават ориентировъчна оценка за производителността на процесора на базата на коефициенти (тегла) на използваемост за отделни класове от операции при решаване на даден клас от задачи. Програмното натоварване на процесора се дефинира априорно въз основа на статистически анализ на изпълнявани типови програми за различни приложения.

При *измерване на компютърни характеристики* необходимата за определяне на оценките информация се получава непосредствено от изследваната КС чрез различни измерващи средства – апаратни и програмни монитори (monitoring), контролни задачи (benchmark), синтетично натоварване (synthetic workload) и др. При измерването е необходимо обектът (КС) да е достъпен, а достъпът до него се осъществява чрез сонди, включени към контролни точки. *Измерващото средство (монитор)* регистрира събдването на определени събития в обекта на измерване и натрупва информацията по подходящ начин. Общият вид на монитор е представен на фиг. В.5.



Фиг. В.5. Принципна структура на монитор

Към мониторите се поставят следните изисквания: точност на измерването; минимално влияние върху обекта на измерване; висока разрешителна способност; широк спектър на приложимост; възможност за предварителна обработка на данните.

Моделирането е метод на научното познание, обединяващ етапите на построяване на модел на изследвания обект, провеждане на експерименти с него и трансформиране на резултатите от експериментите в информация за самия обект на изследването. В този смисъл моделирането е заместване на обект-оригинал с друг обект-модел с цел да се изследват свойствата или поведението в определени ситуации на оригинала чрез експериментиране с модела. Обектът-оригинал може да бъде произволна система или процес, който може и да не съществува реално. Независимо от това, особеностите на оригинала могат да бъдат описани с достатъчна точност, определяна от целта на изследването. Създаваният модел представлява изображение на оригинала във форма, различна от реалното му съществуване, като отразява или възпроизвежда обекта на изследването в достатъчна степен, позволяваща получаването на достоверна информация за неговото изучаване.

Лабораторно упражнение № 1

МАШИННИ КОДОВЕ И МАШИННА АРИТМЕТИКА НАД ДВОИЧНИ ЧИСЛА С ФИКСИРАНА ЗАПЕТАЯ

1.1. Теоретична постановка

1.1.1. Предварителни сведения

В компютърните системи информацията се представя и обработва в двоична бройна система. Съвкупността от разряди, в които се представят двоичните числа, се нарича двоична разрядна решетка. Нейната дължина определя диапазона и точността на представимите числа. Разрядът в левия край на решетката се използва за представяне на знака на числото и се нарича знаков. Прието е знакът "+" да се представя с 0, а знакът "–" – с 1. Числовият разряд, разположен непосредствено след знаковия разряд, се нарича старши, а разрядът в десния край на решетката – младши.

Записът на едно реално число в общия случай се състои от цяла и дробна част, разделени със запетая. При представяне и обработване на числата в компютрите мястото на запетаята се указва по някакъв начин в неявен вид. В най-простия случай се приема, че запетаята е разположена на определено постоянно място в разрядната решетка. Това представяне се нарича представяне на числата с фиксирана запетая. Използват се два варианта, показани на фиг. 1.1.

Запетаята се разполага или след най-младшия числов разряд (фиг. 1.1а), при което могат да се представят само цели числа, или непосредствено след знаковия разряд (фиг. 1.1б), при което могат да се представят само дробни двоични числа (правилни дроби). Ако разрядната решетка съдържа n числови разряда, диапазонът на представимите цели числа е $0 \le |x| \le 2^n - 1$, а на дробните числа е $0 \le |x| \le 1 - 2^{-n}$.

±	a_{n-1}	a_{n-2}	…	a_i	…	a_1	a_0

а)

±	a_1	a_2	…	a_i	…	a_{n-1}	a_n

б)

Фиг. 1.1

В текста и примерите по-нататък запетаята не се отбелязва, а знаковият разряд се отделя от числовите разряди с апостроф (запетая горе).

В компютрите двоичните числа с фиксирана запетая се представят и обработват само като цели числа съгласно фиг. 1.1а. Вариантът от фиг. 1.1б се използва като съставна част (за представяне на мантисата) при представяне на числата с така наречената плаваща запетая. Тук числата се представят във вида $X = m_x.2^{P_x}$, където m_x се нарича мантиса на числото и в нормализиран вид отговаря на условието $2^{-1} \leq |m_x| < 1$, а P_x е порядъкът на числото и показва мястото на запетаята в двоичния му запис. Знакът на числото се представя чрез знака на мантисата му, а порядъкът на числото има отделен знак. Мантисите на числата с плаваща запетая се представят и обработват като дробни числа с фиксирана запетая. Порядъците на числата с плаваща запетая се представят и обработват като цели числа с фиксирана запетая. Събирането и изваждането на цели и на дробни числа с фиксирана запетая се извършват по един и същи начин и по-нататък тези случаи се разглеждат заедно.

При въвеждане в компютърните системи десетичните числа се преобразуват в двоична бройна система. Алгоритмите за преобразуване на цели и дробни числа са различни.

Цялото десетично число се дели на 2, цялата част на резултата се дели отново на 2 и т.н. до получаване на резултат с цяла част 0. Остатъците (0 или 1), отделени при тези последователни операции, се записват в обратен ред. Полученият двоичен запис на цялото число се разполага от десния край на разрядната решетка. Ако останат незаети старши (леви) разряди в решетката, те се запълват с нули, които не променят стойността на числото.

Пример:

$$\begin{array}{cccccccccc} 29:2 & = & 14:2 & = & 7:2 & = & 3:2 & = & 1:2 & = 0 \\ & 1 & & 0 & & 1 & & 1 & & 1 \longleftarrow \text{остатъци} \end{array}$$

т.е. $29_{(10)} = 11101_{(2)}$, респективно $00011101_{(2)}$ в 8 разрядна решетка.

Една правилна десетична дроб се преобразува в двоична система като се умножава по 2, дробната част на полученото произведение се умножава отново по 2 и т.н. до получаване на произведение с дробна част 0 или до запълване на всички разряди в решетката. Отделните цели части (0 или 1) на поредните произведения се записват в реда, по който са намерени. Полученият двоичен запис на дробното число се разполага непосредствено след знаковия разряд в решетката (в нейната лява част). Ако останат незапълнени младши (десни) разряди, в тях се записват нули, които не променят стойността на числото.

Пример:

$$0{,}6875 \times 2 = \boxed{1}{,}3750 \times 2 = \boxed{0}{,}7500 \times 2 = \boxed{1}{,}5000 \times 2 = \boxed{1}{,}0000$$

т.е. $0{,}6875_{(10)} = 0{,}1011_{(2)} = 0{,}10110000_{(2)}$ в 8 разрядна решетка.

ПРИЛОЖЕНИЕ

Бройни системи

Таблица на някои числа в различни бройни системи

БРОЙНИ СИСТЕМИ			
16 (HEX)	10 (DEC)	8 (OCT)	2 (BIN)
0	0	0	0000
1	1	1	0001
2	2	2	0010
3	3	3	0011
4	4	4	0100
5	5	5	0101
6	6	6	0110
7	7	7	0111
8	8	10	1000
9	9	11	1001
A	10	12	1010
B	11	13	1011
C	12	14	1100
D	13	15	1101
E	14	16	1110
F	15	17	1111

ДЕСЕТИЧНА БРОЙНАСИСТЕМА

Това е общоприетата бройна система - тази, в която работим (съкращението й е DEC - Decimal /десетичен/). За нейното образуване се използват числата от 0 до 9. Преминаването от шестнайсетична, осмична и двоична бройни системи в десетична става посредством следната формула:

$$A_q = a_n q^n + a_{n-1} q^{n-1} + \ldots + a_0 q^0 + a_{-1} q^{-1} + \ldots + a_m q^{-m}$$

където: q е основата (бройната система - 2, 8, 10 или 16); n - брой позиции (започват от 0, 1, 2, ...) преди десетичната запетая (ако числото е дробно); m - брой позиции (започват от 1, 2, ...) след десетичната запетая (ако числото е дробно); a - тегловен коефициент.

Примери:

- Числото 56_{10} (чете се числото 56 в десетична бройна система) = $5.10^1+6.10^0$ = 56, където (за този пример) $q=10$; $n=1$ (5 заема нулева позиция, 6 - първа).
- Числото $7F,13_{16}$ (чете се числото 7F,13 в шестнайсетична бройна система) = $7.16^1+F.16^0+1.16^{-1}+3.16^{-2} = (127+19/256)_{10}$, където $q=16$; $n=1$; и $m=2$.
- Числото $10110,1_2$ (чете се числото 10110,1 в двоична бройна система) = $1.2^4+0.2^3+1.2^2+1.2^1+0.2^0+1.2^{-1} = 22,5_{10}$ ($q=2$; $n=4$; $m=1$).
- Числото 76_8 (чете се числото 76 в осмична бройна система) = $7.8^1+6.8^0 = 62_{10}$ ($q=8$; $n=1$; $m=0$).

ДВОИЧНА БРОЙНАСИСТЕМА

При тази бройна система числата се получават като поредица от нули и единици - 10010. Всяка една цифра от това число е един бит, като най-старшият бит стои най-вляво на числото. Обикновено този бит (най-старшия) се взема за знаков бит при извършване на числови операции. За да стане по-ясно това погледнете следния пример:

знаков бит	бит 9	бит 8	бит 7	бит 6	бит 5	бит 4	бит 3	бит 2	бит 1	бит 0
1	0	0	1	1	1	0	1	0	1	0

Двоичната бройна система се означава с BIN (Binary - двоен). Превръщането на десетично число в двоично става по следния начин:

- Извършва се **целочислено** делене на две (основата) и се получава цяла част и остатък до получаване на нулева цяла част;
- Получените остатъци от деленията се записват в обратен ред на получаването им.

Пример:

Да се превърне числото 57_{10} в двоична бройна система.

	ц.	ос.
57 :2=	28	1

28	:2=	14	0
14	:2=	7	0
7	:2=	3	1
3	:2=	1	1
1	:2=	0	1

Резултата се записва в ред обратен на получаването му: $57_{10}=111001_2$

Дробнатачаст на числото се изчислява по следнияалгоритъм:

* Умножава се дробната част с основата (2) и се получава нова дробна част и цяла част.

Преносът към цялата част (1 или 0) са цифри от резултата, **в реда на тяхното получаване**. Умножението продължава **до получаване на нужните цифри след десетичната запетая** или до дробна част равна на нула.

Пример:

	x2	x2
0,25	0,5	0
пренос към цялата част	0	1

Резултатът се записва в реда на получаването му: $0{,}25_{10}=0{,}01_2$

	x2	x2	x2	x2	x2	x2	
0,551	0,102	0,204	0,408	0,816	0,622	0,244	...
пренос към цялата част	1	0	0	0	1	1	...

Резултатът се записва в реда на получаването му: $0{,}551_{10}=0{,}100011\ldots_2$

ОСМИЧНА БРОЙНАСИСТЕМА

В тази бройна система се използват цифрите от 0 до 7. Съкращението ѝ е OCT (от Octal - осмичен). Превръщането на число от десетична в осмична бройна система става аналогично на превръщането от десетична в двоична, но за основа се взима числото 8 и вместо да се дели (умножава) на две - се дели (умножава) на 8.

Пример:

Да се превърне числото 57_{10} в осмична бройна система.

		ц.	ос.
57	:8=	7	1
7	:8=	0	7

Записваме резултата обратно на получаването му: $57_{10}=71_8$

Да се превърне числото $57{,}25_{10}$ в осмична бройна система.

	др.	ц.
0,25*8=	0	2

Резултатът се записва в реда на получаването му: $57{,}25_{10}=71{,}2_8$

Превръщанеотосмичнав двоичнабройнасистема

Превръщането става като всяко едно от числата в осмична бройна система се превърне в триади (триадата е поредица от три бита) в двоично число, като се използва таблицата в началото на документа.

Пример:

Да се превърне числото 72 от осмична в двоична бройна система.

Осмичните цифри се преобразуват в **двоични** триади и всяка осмична цифра заема три двоични бита - 7 ->111; 2 -> 010.
Слепват се двете части на числото и се получава резултатът: $52_8=111010_2$.

Превръщанеотдвоичнав осмичнабройнасистема

Това превръщане става по обратния начин - двоичното число се разделя на триади като се **започва от единиците** и от таблицата се записват съответните осмични числа. Ако в началото на двоичното число битовете не стигат, за да образуват триада, то се попълват с нули.

Пример:

Да се превърне числото 1011110 от двоична в осмична бройна система.

Разделяме числото на триади:

1 | 011 | 110

Тъй като от пред не стигат числата за образуване на триада, то там се попълват нули:

001|011|110

Отделната триада се превръща в съответното число в осмичен код, посредством таблицата за бройните системи в началото на документа. Ето и съответните стойности: 001 -> 1; 011 -> 3 и 110 -> 6. Сглобяват се получените цифри и се получава числото в осмична бройна система: $1011110_2=136_8$.

ШЕСТНАДЕСЕТИЧНАБРОЙНАСИСТЕМА

За цифри на тази бройна система се използват цифрите от 0 до 9 и буквите A, B, C, D, E и F. Съкращението й е HEX (от Hexadecimal - шестнадесетичен). Превръщането на число от десетична в шестнадесетична бройна система става аналогично на превръщането от десетична в двоична, но за основа се взима числото 16 и вместо да се дели (умножава) на две - се дели (умножава) на 16.

Пример:

Да се превърне числото 57_{10} в шестнадесетична бройна система.

	ц.	ос.
57 :16=	3	9
3 :16=	0	3

Записват се цифрите в ред обратен на получаването му: $57_{10}=39_{16}$

Да се превърне числото $57{,}25_{10}$ в шестнадесетична бройна система.

	др.	ц.
0,25*16=	0	4

Резултатът се записва в реда на получаването му: $57{,}25_{10}=39{,}4_{16}$

Превръщанеот шестнадесетична в двоичнабройнасистема

Това превръщане става аналогично на осмично-двоичното преобразуване, само, че тук се използват тетради вместо триади (тетрадата е поредица от четири бита в двоичен код). Всяка цифра от шестнадесетичното число се превръщат в тетради в двоично число, като се използва таблицата в началото на документа. Получените тетради се сглобяват и се получава двоичното число.

Пример:

Да се превърне шестнадесетичното число F31 в двоично.

Значи следваме алгоритъма: всяко едно от числата /буквите/ го превръщаме в тетради - F -> 1111; 3 -> 0011; 1 -> 0001.

$F31_{16}=111100110001_2$.

Превръщанеотдвоичнавшестнадесетичнабройнасистема

Превръщането от двоична в шестнадесетична бройна система става обратно на шестнадесетично-двоичното преобразуване: числото в двоичен код се разделя на тетради (от единиците), ако отпред не стигат цифри се допълват нули, и съответните тетради се превръщат в шестнадесетични символи, пак с помощта на горната таблица. Сглобяват се получените шестнадесетични символи и се получава съответното шестнадесетично число.

Пример:

Да се превърне числото 10110111101 от двоично в шестнадесетично.

Разделяме числото на тетради:

101 | 1011 | 1101

От пред се допълва една нула, за да се образува тетрада:

0101 | 1011 | 1101

Съответните тетради се превръщат в шестнадесетични цифри: 0101 -> 5; 1011 -> B; 1101 -> D.

$10110111101_2=5BD_{16}$.

преобразуване чрез съответствия спрямо степените на 2

740_{10}= 1011100100_{2}.

	1	2	3	4	5	6	7	8	9	10	11
	740	740	228	228	100	36	4	4	4	0	0
n	10	9	8	7	6	5	4	3	2	1	0
2^n	1024	512	256	128	64	32	16	8	4	2	1
	0	**1**	**0**	**1**	**1**	**1**	**0**	**0**	**1**	**0**	**0**

- 1 тежестта на разряда (1024) е по-голяма от числото което се преобразува
- 2 тежестта на разряда (512) е по-малка от числото което се преобразува 740-512=228
- 3 256>228 преминава се към колон
- 4 228>128 228-128=100 записва се в колона 5
- 5 100-64=36 записва се в колона 6

. . . . . .

	1	0	1	1	0	0	1	1	0	0	1
n	10	9	8	7	6	5	4	3	2	1	0
2^n	1024	512	256	128	64	32	16	8	4	2	1
	1024	0	256	128	0	0	16	8	0	0	1
SUM=	**1433**										

10110011001_{2}= 1433_{10}.