## Позиционни бройни системи от типа на Арабската

## Кирил Иванов

Текстът е предназначен за упражнения в този смисъл, че е съсредоточен изцяло върху практиката и непосредствените следствия от нея.

## Съдържание

1. Броини системи от	типа на Арабската		2	
2. Някои свойства на Г	IБС от типа на Арабскап	1a	4	
3. Преобразуване на пе	риодична дроб в рациона	лен израз с крайни запис	<i>u</i> 5	
4. Сравняване на памет	пи с близка сложност, ба	зирани на различни ПБС	· 6	
5. Преобразуване на кр	аен непериодичен запис с	от s-ичнав 10-ична П	<i>IEC</i> 8	
6. Преобразуване на кр	аен непериодичен запис с	от 10-ична в s-ична П	<i>IBC</i> 11	
7. Преобразуване на кр	аен непериодичен запис с	от ПБС с основа $\ _{S}^{\ k}$ в ПБ	С с основа <sub>S</sub> <sup>n</sup> 17	
8. Сумиране в ПБС с осн	108a S			
9. Изваждане в ПБС с о	снова s		21	
10. Умножаване в ПБС	с основа S		22	
11. Делене в ПБС с осно	ва s		24	
12. Преобразуване от в	една в друга ПБС на запис	и, съдържащи период	26	
Разположе	ние на примерип	ne		
Стр. 3 : Примери 1, 2.	Стр. 10 : Примери 16-19.	Стр. 15 : Примери 32-35.	Стр. 20 : Примери 48-51.	Стр. 25 : Примери 66-67.
Стр. 5: Пример 3.	Стр. 11 : Примери 20-22.	Стр. 16 : Примери 36-39.	Стр. 21 : Примери 52-54.	Стр. 26 : Примери 68-70.
	Стр. 12 : Примери 23-24.	Стр. 17 : Примери 40-41.	Стр. 22 : Примери 55-58.	Стр. 27: Примери 71-73.
Стр. 8 : Примери 8-11.	Стр. 13 : Примери 25-28.	Стр. 18 : Примери 42-44.	Стр. 23 : Примери 59-62.	Стр. 28: Пример 74, 75.
Стр. 9 : Примери 12-15.	Стр. 14 : Примери 29-31.	Стр. 19 : Примери 45-47.	Стр. 24 : Примери 63-65.	

## 1. Бройни системи от типа на Арабската

**Бройна система** (БС) наричаме система от знакове и правила, чрез които може да се представят числа с две основни цели: първо, и най-важно – да има възможност за извършване на основните аритметични действия – събиране, изваждане, умножение и деление и, второ, да може да се представят всякакви числа. Първите използвани от хората БС са служили само за представяне на относително малки естествени числа. В най-удобните БС, каквато е Арабската, сложността и на изчисленията, и на представянето на числата нараства малко и плавно с увеличение на големината на числата или с приближаването им към нулата.

Говорим именно за представяне на числа, защото знаковете може да имат всякаква природа – визуален образ, звук, електрически ток, магнитно поле, светлинен лъч, наличие или липса на перфорация върху пластина и т. н.

**Цифра** наричаме знак от БС, който означава някаква стойност. Обикновено болшинството знакове на БС са цифри.

**Непозиционна бройна система** наричаме тази, в която стойността на цифрата винаги участвува по един и същ начин във формирането на представеното число, независимо къде, в коя позиция се намира самата цифра в това представяне. Например в някои от най-ранните, известни днес, БС всяка резка върху пръчка добавя по една единица към записаното върху пръчката число.

**Позиционна бройна система** (ПБС) наричаме тази, в която стойността на една и съща цифра може да участвува по различни начини във формирането на представеното число, в зависимост от различните места на цифрата в представянето. Например в записа 101 (сто и едно) първата единица се умножава по сто, а втората – по едно.

По-надолу ще работим само с визуални изображения на числа и ги наричаме записи на числата.

Този конспект разглежда само ПБС със записи от вида

(1) 
$$\overline{a_n \dots a_1 a_0}, \overline{c_1 \dots c_k \dots}_{(s)}$$

където (точно по аналогия с привичната за нас Арабска ПБС):

- може да има още знак плюс или минус, например  $+34.01_{(5)}$  и -0.f  $9_{(16)}$  ;
- може да има период в дробната част, означен с кръгли скоби, например  $210,21(304)_{(8)}$  ;
- хоризонталната линия означава, че под нея поне една цифра е заменена с някакво означение (във формулата 1 всички цифри са означени с индексирани букви);
- запетаята отделя цялата от дробната част;
- с индекс *s* в кръгли скоби в края на записа е означена **основата на ПБС**, т. е. естествено число, по-голямо от единица, което участвува в дефиницията на записаното число, както е показано във формулите *2* и *3* (или еквивалентните им формули *4* и *5*);
- възможните стойности на цифрите са числата 0, 1, 2, ..., s-1 ;
- при основа до 10 се използват цифрите 0, 1, ..., 9;
- при основи от 11 до 36 за цифри се използват латинските букви със стойност 10 на цифрата a или A , 11 на b или B , 12 на c или C , ..., 35 на z или Z .

Числото, записвано чрез горния запис, се дефинира с формулите

**(2)** 
$$\overline{a_n...a_1a_0}_{(s)} \stackrel{\text{def}}{=} a_n.s^n + ... + a_1.s^1 + a_0.s^0$$
 за цялата част и

(3) 
$$\overline{0, c_1 c_2 ... c_k ...}_{(s)} \stackrel{\text{def}}{=} \frac{c_1}{s^1} + \frac{c_2}{s^2} + ... + \frac{c_k}{s^k} + ...$$
 за дробната част .

Понякога, например при изчисления с калкулатор или алгоритъм, е по-удобно да се използват еквивалентни на горните формули

(4) 
$$\overline{a_n...a_1a_0}_{(s)} = ((...(a_n.s+a_{n-1}).s+...+a_2).s+a_1).s+a_0$$
 за цялата част и

$$c_{k-2} + \frac{c_{k-1} + \frac{c_k + \dots}{s}}{s}$$

(5) 
$$\overline{0, c_1 c_2 ... c_{k-2} c_{k-1} c_k ...}_{(s)} = \frac{c_1 + \frac{c_2 + \frac{\cdots}{s}}{s}}{s}$$
 за дробната част .

Когато числото се записва с една цифра, основата на ПБС може да се пропуска, защото смисълът на записа е еднозначен.

## Пример 1

$$2504_{(6)} = 2.6^3 + 5.6^2 + 0.6^1 + 4.6^0 = ((2.6+5).6+0).6+4$$

## Пример 2

$$0,215_{(7)} = \frac{2}{7^1} + \frac{1}{7^2} + \frac{5}{7^3} = \frac{2 + \frac{1 + \frac{5}{7}}{7}}{7}$$

ПБС от описания вид с основа s наричаме s -ична ПБС.

## Арабска ПБС наричаме 10-ичната ПБС.

Прочитането на запис в такава ПБС с основа, различна от 10, става знак по знак. Думите сто, осемдесет, хиляда, петнадесет и т. н. означават (по подразбиране), че записът е в десетична ПБС (Арабската). При четенето на цифри букви в математиката в България е прието да се използват латинските им названия, т. е. "а", "бе", "це", "де" и т. н., вместо английските названия на същите букви "ей", "би", "си", "ди" и т. н.

## Например:

 $-21,2(01)_{(3)}$  прочитаме минус две едно запетайка две и нула едно в период троично или минус две едно цяло и две и нула едно в период троично.

4f ,8  $c(2a)_{(16)}$  прочитаме четири еф запетайка осем це и две а в период шестнадесетично или четири еф цяло осем це и две а в период шестнадесетично. В този запис няма нужда от хоризонтална черта, защото буквите f, c и a са самите цифри, а не буквени означения на цифри.

 $21099_{(10)}$  прочитаме двадесет и една хиляди деветдесет и девет.

 $3451_{(7)}$  прочитаме три четири пет едно седмично.

Параметричния запис  $-\overline{xycx}_{(9)}$  прочитаме: минус хикс игрек це хикс деветично.

## 2. Някои свойства на ПБС от типа на Арабската

(Някои от изброените по-долу свойства са елементарни следствия от други, също изброени, свойства.)

Във всяка s -ична ПБС и за всяко естествено число k са верни следните твърдения:

(6) 
$$s^k = 1 \underbrace{0 \dots 0}_{k} (s)$$
.

(7) 
$$\frac{1}{s^k} = 0, \underbrace{0...0}_{k-1} 1_{(s)}$$
.

(8) 
$$\overline{a_{k-1}...a_0}_{(s)} < 1 \underbrace{0...0}_{k}_{(s)} \le \overline{1d_{k-1}...d_0}_{(s)}$$
 (каквито и да са цифрите  $a_{k-1},...,a_0,d_{k-1},...,d_0$  ).

**(9)** 
$$0, \underbrace{0...0}_{k} c... c_{(s)} < 0, \underbrace{0...0}_{k-1} 1_{(s)} \le 0, c_1...c_{k-1} 1 c_{k+1}... c_{(s)}$$
 (за всякакви цифри  $c, c_0, ..., c_{k-1}, c_{k+1}$  ).

(10) 
$$0 \le \overline{a_{k-1} ... a_0}_{(s)} < s^k$$
 (каквито и да са цифрите  $a_{k-1}, ..., a_0$ ).

(11) 
$$(n>k \land d_n \neq 0) \Rightarrow \overline{d_n...d_0}_{(s)} > \overline{c_k...c_0}_{(s)}$$
 (каквито и да са цифрите  $d_n,...,d_0,a_{n-1},...,a_0$ ).

(12) 
$$(n>k \land c_{k+1}\neq 0) \Rightarrow \overline{0, 0...0} c_{k+1}... (s) > \overline{0, 0...0} d_{n+1}... (s)$$
 (за всякакви цифри  $c_{k+1}, d_{n+1}$  ).

Едно и също цяло число може да се запише с толкова по-малко цифри, колкото по-голяма е основата на ПБС, т. е.

(13) 
$$\left( \overline{a_k \dots a_0}_{(s)} = \overline{d_n \dots d_0}_{(t)} \wedge a_k \neq 0 \wedge d_n \neq 0 \wedge s < t \right) \Rightarrow k \geq n .$$

Ако две различни положителни числа са записани в s-ичната ПБС с еднакъв брой цифри, евентуално, с добавяне на незначещи нули, и дробната запетайка е на една и съща позиция спрямо първата цифра, тогава по-голямото от числата може да бъде определено чрез лексикографско сравняване (точно както се сравняват низове). Т. е., сравняват се последователно отляво надясно цифрите от едни и същи позиции и първата намерена двойка различни цифри определя кое число е по-голямо. Това може да се изрази с формули по следния начин:

(14) 
$$X = \overline{a_n \dots a_k d_x \dots , \dots }_{(s)} \land Y = \overline{a_n \dots a_k d_y \dots , \dots }_{(s)} \land d_x > d_y \Rightarrow X > Y$$
 (когато има разлика в цялата част) и

(15) 
$$X = a_n \dots a_0, c_1 \dots c_k d_x \dots_{(s)} \land Y = \overline{a_n \dots a_0, c_1 \dots c_k d_y \dots_{(s)}} \land d_x > d_y \Rightarrow X > Y$$
 (когато разликата е само в дробната част).

Лексикографското сравняване често се използва в хардуера, защото е по-лесно от числовото. Например, на машинно ниво, в аритметичния блок на процесора, така се сравняват абсолютните стойности на числа от типа double на езика C++.

Всяко число от интервала  $[0\,;s^k-1]$  може да се представи във вида  $\overline{a_{k-1}...a_0}_{(s)}$  и всяко число от вида  $\overline{a_{k-1}...a_0}_{(s)}$  принадлежи на интервала  $[0\,;s^k-1]$  , т. е.

(16) 
$$\left\{\overline{a_{k-1}...a_{0(s)}}:0\leq a_{k-1}\leq s-1,...,0\leq a_0\leq s-1\right\}\equiv \left[0;s^{k-1}\right]$$
.

Броят на целите неотрицателни k -цифрени s -ични числа е точно  $s^k$  . Т. е.

(17) 
$$|\overline{a_{k-1}...a_0}(s)| \cdot 0 \le a_{k-1} \le s-1,..., 0 \le a_0 \le s-1| = s^k$$
.

(18) 
$$\overline{a_{n-1}...a_0}_{(s)}.s^k = \overline{a_{n-1}...a_0}\underbrace{0...0}_{k}_{(s)}$$
.

(19) 
$$\overline{a_n...a_k a_{k-1}...a_0}_{(s)} = \overline{a_n...a_k}_{(s)}.s^k + \overline{a_{k-1}...a_0}_{(s)}$$
.

(20) 
$$\frac{\overline{a_{n-1}...a_k a_{k-1}...a_0}_{(s)}}{s^k} = \overline{a_{n-1}...a_k, a_{k-1}...a_0}_{(s)}.$$

(21) 
$$\overline{0, a_1 \dots a_k c_1 \dots c_{k+n} \dots c_{s}} \cdot s^k = \overline{a_1 \dots a_k, c_1 \dots c_{k+n} \dots c_{s}}$$
.

(22) 
$$\overline{0}, c_1...c_k c_{k+1}...c_n |_{(s)} = \overline{0}, c_1...c_k |_{(s)} + \frac{\overline{0}, c_{k+1}...c_n |_{(s)}}{s^k} .$$

Ако  $\,\xi\,\,$  е най-голямата цифра в  $\,s\,$  -ичната ПБС, т. е.  $\,\xi_{(s)} \! = \! s \! - \! 1\,\,$  , тогава

(23)  $\underbrace{\xi \dots \xi}_{k} {}^{(s)} = s^{k} - 1$  (следствие от това, че  $\underbrace{\xi \dots \xi}_{k} {}^{(s)}$  е най-голямото k -цифрено s -ично число и че  $\underbrace{1 \dots \xi}_{k} {}^{(s)} = s^{k}$  е най-малкото (k+1) -цифрено s -ично число) и

(24) 
$$0, \underbrace{\xi \dots \xi}_{k} (s) = \frac{s^{k} - 1}{s^{k}}$$
 (следствие от формули 20 и 23).

## 3. Преобразуване на периодична дроб в рационален израз с крайни записи

Във всяка s -ична ПБС и за всяко естествено число k е вярно равенството

(25) 
$$\overline{0}, (c_1...c_k)_{(s)} = \frac{\overline{c_1...c_k}_{(s)}}{s^k-1}$$
.

Доказателство

$$\overline{0,(c_{1}...c_{k})}_{(s)} = \overline{0,c_{1}...c_{k}}_{(s)} + \overline{0,\underbrace{0...0}_{k}(c_{1}...c_{k})}_{(s)} = \overline{0,c_{1}...c_{k}}_{(s)} + \frac{\overline{0,(c_{1}...c_{k})}_{(s)}}{s^{k}} .$$

$$\Rightarrow \overline{0,(c_{1}...c_{k})}_{(s)} = \overline{0,c_{1}...c_{k}}_{(s)} .$$

$$\Rightarrow \overline{0,(c_{1}...c_{k})}_{(s)} = \frac{s^{k}.\overline{0,c_{1}...c_{k}}_{(s)}}{s^{k}-1} = \frac{\overline{c_{1}...c_{k}}_{(s)}}{s^{k}-1} .$$

#### Пример 3

$$0,(613)_{(9)} = \frac{613_{(9)}}{9^3 - 1} = \frac{6 \cdot 9^2 + 1 \cdot 9 + 3}{9^3 - 1} = \frac{486 + 9 + 3}{729 - 1} = \frac{498}{728} = \frac{249}{364}$$

$$0,(002)_{(7)} = \frac{2}{7^3 - 1} = \frac{2}{342} = \frac{1}{171}$$

## Пример 5

$$0,(0001)_{(2)} = \frac{1}{2^4 - 1} = \frac{1}{15} = 0,00(6)_{(10)}$$

От формули 21, 22 и 25 следва

(26) 
$$\overline{0, f_1...f_j(c_1...c_k)}_{(s)} = \frac{\overline{f_1...f_j(s).(s^k-1)} + \overline{c_1...c_k(s)}}{s^j.(s^k-1)}$$
.

Доказателство

$$\overline{0, f_{1}...f_{j}(c_{1}...c_{k})}_{(s)} = \frac{\overline{f_{1}...f_{j}}_{(s)}}{s^{j}} + \frac{\overline{0, (c_{1}...c_{k})}_{(s)}}{s^{j}}$$

$$= \frac{\overline{f_{1}...f_{j}}_{(s)}}{s^{j}} + \frac{\overline{c_{1}...c_{k}}_{(s)}}{s^{j}(s^{k}-1)} = \frac{\overline{f_{1}...f_{j}}_{(s)}.(s^{k}-1) + \overline{c_{1}...c_{k}}_{(s)}}{s^{j}.(s^{k}-1)}$$

## Пример 6

$$0,0001(4)_{(5)} = \frac{1.(5^{1}-1)+4}{5^{4}.(5^{1}-1)} = \frac{4+4}{5^{4}.4} = \frac{2}{5^{4}} = \frac{2^{5}}{5^{4}.2^{4}} = \frac{32}{10^{4}} = 0,0032$$

От формули 18 и 26 следва

(27) 
$$\overline{a_n...a_0, f_1...f_j(c_1...c_k)}_{(s)} = \frac{\overline{a_n...a_0 f_1...f_j(s).(s^k-1) + \overline{c_1...c_k(s)}}}{s^j.(s^k-1)}.$$

#### Пример 7

$$1,0(12)_{(3)} = \frac{10_{(3)}.(3^2-1)+12_{(3)}}{3.(3^2-1)} = \frac{3.8+5}{3.8} = \frac{29}{24} = 1\frac{5}{24}$$

## 4. Сравняване на памети с близка сложност, базирани на различни ПБС

За съхраняване и разпознаване на стойност в един разряд на компютърната памет трябва да се поддържат и разпознават различни устойчиви състояния за всички различни цифри, които може да съдържа разрядът, т. е. толкова на брой състояния, колкото е основата на ПБС, върху която е базирана паметта. Следователно, с нарастването на основата на ПБС расте и сложността на паметта.

Също така, очевидно, компютърната памет се усложнява и с увеличаването на броя на разрядите.

Затова, като *относителна мярка за сложност* на компютърната памет, може да се приеме произведението на броя на разрядите и основата на ПБС, върху която е базирана паметта. Тогава за практиката има значение въпросът: Измежду паметите с приблизително еднаква сложност, при коя ПБС в паметта може да се съхранят най-много различни стойности?

Ако означим с s основата на ПБС и с n броя на разрядите в паметта, то от формула 17 следва, че в тази памет може да се съхраняват  $s^n$  различни числа. Тогава горният въпрос може да се формулира така: При фиксирано произведение s.n, при кое s е най-голяма степента  $s^n$ ?

	_					U
Следващата таб	ARTITIA MATANANA	CHADITATITIONA II	1 TIGITO 1110	TOMOTH TO	A matem to tentame	miiii
L./IP/IBallia la lat	тлина пиказва	спавнението на	4 няки/ки	памети п	1 148 68 8110 15	чии

s.n=30							
S	2	3	5	6	10	15	30
$\Rightarrow n$	15	10	6	5	3	2	1
$\Rightarrow S^n$	32768	59049	15625	7776	1000	225	30

Може да се докаже, че при фиксирано s.n степента  $s^n$  нараства с приближаването на s към Неперовата единица e=2,7... . Следователно, при основа 3 на ПБС се получава най-голям капацитет на компютърната памет *по горния критерий*. От горната таблица се вижда, че дори при s.n=30 разликата в капацитета на 3-ичната и 2-ичната памети е близо до 1 към 2.

Въпреки това, масово разпространените днес компютри са базирани на 2-ична ПБС, защото получаваната при това функционалност все още е задоволителна, а при основа 2 се опростява тяхното устройство и технологията на производството им и се подобряват някои от експлоатационните им характеристики. Например, значителното опростяване, което може да бъде постигнато в сложните схеми дори при преминаване от основа 3 към основа 2, намалява топлинното отделяне при функционирането им и закономерно ускорява, макар и малко, общата скорост на реакция на цялата схема. Много съществено се оказва и това, че съвременните 2-ични памети, въпреки че имат капацитет, много по-малък от този на 3-ичните памети, все пак са достатъчни за обичайните потребности на отделните потребители и на стопанските приложенията.

Обективно погледнато, при голям брой n на разрядите, произведението s.n, би било точна мярка за сложност на паметта, само когато схемата на паметта се усложнява еднакво и при добавянето на един разряд, и при реализирането на още една възможна стойност на цифра. В практиката има различия между степените на усложнение в двата случая. Освен това, основата на ПБС твърде много влияе и върху сложността на схемите, обработващи числата. Важно значение имат и броят, и разнообразието на схемите. В крайна сметка, сложността на цялата компютърна конфигурация трябва да бъде оценявана *комплексно* и оценката *е относителна*.

За разлика от универсалните компютри, калкулаторите са базирани на 10-ична ПБС по две причини. Първо, защото схемите им са сравнително прости, поради ограниченото си предназначение (само за изчисления), и е икономически приемливо усложняването им заради поддръжката на десет възможни стойности на цифри, вместо само на две. Второ, дробните числа трябва да може да бъдат въвеждани в калкулаторите в десетичен запис, защото той е привичен за нас. Обаче, повечето крайни 10-ични дроби се записват в 2-ична ПБС като безкрайни периодични (както е показано в точка 6). Ако такива дроби бъдат преобразувани до 2-ичен запис, за да бъдат съхранени в 2-ична памет, ще се наложи да бъдат закръглени (традиционно в паметта се съхраняват само краен брой цифри без период). След такова закръгляне окончателният резултат от изчисленията също ще съдържа някаква грешка, макар и малка. За такива крайни 10-ични дроби, които в 2-ичен запис са периодични, 10-ичната памет и 10-ичните изчислителни схеми позволяват да се получават точни резултати, а именно точността е първостепенна цел в предназначението на калкулаторите.

По съвсем аналогични съображения — за да се избегне загубата на точност в много голям брой изчисления с дроби — всички съвременни аритметични блокове на процесори за универсалните компютри, а също и някои (и съвременни, и вече излезли от употреба) езици за програмиране, поддържат формати за представяне на числа в паметта във вид на редици от стойности на десетични цифри. Аритметиката с такива формати изцяло следва правилата на 10-ичната ПБС и избягва закръглянията, които биха били предизвикани при преминаването

от 10-ичен към 2-ичен запис. Например, такъв е форматът на типа cardinal в езика С#, а също и десетбайтовият формат с 2-ично представяне на 10-ичните цифри, разпознаван от аритметичния блок на масово разпространените днес 32-разрядни и 64-разрядни процесори.

## 5. Преобразуване на краен непериодичен запис от s - ична в 10 - ична ПБС

Универсалният алгоритъм за това е да се заместят стойностите на цифрите и на основата съответно във формулите 2, 3, 4 или 5, в тези от тях, с които е най-удобно да се работи, и да се извършат действията.

## Пример 8

$$T$$
ърсим  $1001_{(2)} = ?_{(10)}$  :

$$1001_{(2)} = 1.2^3 + 0.2^2 + 0.2^1 + 1.2^0 = 9$$
 или 
$$1001_{(2)} = \left((1.2 + 0).2 + 0\right).2 + 1 = 9$$
 .

## Пример 9

Търсим 
$$1001_{(3)} = ?_{(10)}$$
 :

$$1001_{(3)} = 1.3^3 + 0.3^2 + 0.3^1 + 1.3^0 = 28_{(10)}$$
 или 
$$1001_{(3)} = \left( (1.3 + 0).3 + 0 \right).3 + 1 = 28_{(10)} .$$

Вижда се, че при представянето на записа на числото в израз със степени на основата на ПБС всяка стойност на цифра се умножава по степен с такъв степенен показател, колкото е броя на цифрите в записа след умножаваната. Например, първата цифра на цялата част винаги се умножава по степен на основата на ПБС със степенен показател, с единица по-малък от броя на цифрите в цялата част. Това може да се изрази с формула по следния начин:

(28) 
$$\overline{\underline{d}_{n \ \mu\nu\phi\rho\mu}}_{(s)} (s) = d \cdot s^{n-1} + \dots .$$

#### Пример 10

$$T_{\text{ърсим}} 1001_{(5)} = ?_{(10)}$$
 :

$$1001_{(5)} = 1.5^3 + 0.5^2 + 0.5^1 + 1.5^0 = 126_{(10)}$$
 или 
$$1001_{(5)} = ((1.5 + 0).5 + 0).5 + 1 = 126_{(10)}.$$

## Пример 11

Търсим 
$$1322_{(4)} = ?_{(10)}$$
 :

$$1322_{(4)} = 1.4^3 + 3.4^2 + 2.4^1 + 2.4^0 = 64 + 48 + 8 + 2 = 122_{(10)}$$
 или 
$$1322_{(4)} = ((1.4+3).4+2).4 + 2 = (7.4+2).4 + 2 = 30.4 + 2 = 122_{(10)}$$
 .

Търсим  $211202_{(3)} = ?_{(10)}$  :

$$211202_{(3)} = 2.3^5 + 1.3^4 + 1.3^3 + 2.3^2 + 0.3^1 + 2.3^0 = 486 + 81 + 27 + 18 + 0 + 2 = 614_{(10)}$$
 или 
$$211202_{(3)} = \left(\left(\left(2.3 + 1\right).3 + 1\right).3 + 2\right).3 + 0\right).3 + 2$$
 
$$= \left(\left(\left(7.3 + 1\right).3 + 2\right).3 + 0\right).3 + 2 = \left(\left(22.3 + 2\right).3 + 0\right).3 + 2 = \left(68.3 + 0\right).3 + 2 = 204.3 + 2 = 614_{(10)}\right)$$

## Пример 13

Търсим  $f9ac_{(16)}=?_{(10)}$  :

$$f9ac_{(16)} = 15.16^3 + 9.16^2 + 10.16^1 + 12.16^0$$
  $= 15.4096 + 9.256 + 10.16 + 12.1 = 61440 + 2304 + 160 + 12 = 63916_{(10)}$  гли  $f9ac_{(3)} = ((15.16 + 9).16 + 10).16 + 12 = (249.16 + 10).16 + 12 = 3994.16 + 12 = 63916$  .

При преобразуване на дроби от 2-ична или от 5-ична към 10-ична система е удобно, вместо делене, да се използват зависимостите

(29) 
$$\frac{A}{2^k} = \frac{A.5^k}{2^k 5^k} = \frac{A.5^k}{10^k} = \overline{0, c_1...c_k}_{(10)}$$
, където  $A.5^k = \overline{c_1...c_k}_{(10)}$ , и

(30) 
$$\frac{A}{5^k} = \frac{A \cdot 2^k}{5^k \cdot 2^k} = \frac{A \cdot 2^k}{10^k} = \overline{0, c_1 \dots c_k}_{(10)}$$
, където  $A \cdot 2^k = \overline{c_1 \dots c_k}_{(10)}$ , като в някои случаи  $c_1 = c_2 = \dots = c_{k-j} = 0$  за някакво  $j$ .

## Пример 14

Търсим  $0,101_{(2)}=?_{(10)}$  :

$$0,101_{(2)} = \frac{1}{2^1} + \frac{0}{2^2} + \frac{1}{2^3} = \frac{5}{8} = 0,625_{(10)}$$

или

$$0,101_{(2)} = \frac{1 + \frac{0 + \frac{1}{2}}{2}}{2} = \frac{5}{8} = 0,625_{(10)}$$

#### Пример 15

Tърсим  $0,101_{(5)} = ?_{(10)}$  :

$$0,101_{(5)} = \frac{1}{5^1} + \frac{0}{5^2} + \frac{1}{5^3} = \frac{26}{125} = 0,184_{(10)}$$

или

$$0,101_{(5)} = \frac{0 + \frac{1}{5}}{5} = \frac{26}{125} = 0,184_{(10)}$$

Tърсим  $0,1_{(9)}=?_{(10)}$  :

$$0,1_{(9)} = \frac{1}{9} = 0,(1)_{(10)}$$
.

## Пример 17

Търсим  $0.72_{(8)} = ?_{(10)}$  :

$$0,72_{(8)} = \frac{7}{8^1} + \frac{2}{8^2} = \frac{58}{64} = 0,90625_{(10)}$$

или

$$0,72_{(8)} = \frac{7 + \frac{2}{8}}{8} = \frac{\frac{56 + 2}{8}}{8} = \frac{58}{64} = 0,90625_{(10)}$$

## Пример 18

Tърсим  $-101,11_{(2)}=?_{(10)}$  :

$$-101,11_{(2)} = -\left(1.2^{2} + 0.2^{1} + 1.2^{0} + \frac{1}{2^{1}} + \frac{1}{2^{2}}\right) = -\left(4 + 0 + 1 + \frac{1}{2} + \frac{1}{4}\right) = -\left(5 + \frac{3}{4}\right) = -5,75_{(10)}$$

или

$$-101,11_{(2)} = -\left((1.2+0).2+1+\frac{1+\frac{1}{2}}{2}\right) = -\left(4+1+\frac{\frac{3}{2}}{2}\right) = -\left(5+\frac{3}{4}\right) = -5,75_{(10)}$$

## Пример 19

Търсим  $0, fa_{(16)} = ?_{(10)}$  :

$$0, fa_{(16)} = \frac{15}{16^{1}} + \frac{10}{16^{2}} = \frac{240 + 10}{256} = \frac{250}{256} = 0,9765625_{(10)}$$

или

$$0, fa_{(16)} = \frac{15 + \frac{10}{16}}{16} = \frac{\frac{240 + 10}{16}}{16} = \frac{250}{256} = 0,9765625_{(10)}$$

Търсим  $-14323,001_{(5)} = ?_{(10)}$  :

$$-14323,001_{(5)} = -\left(1.5^4 + 4.5^3 + 3.5^2 + 2.5^1 + 3.5^0 + \frac{1}{5^3}\right) = -\left(625 + 500 + 75 + 10 + 3 + \frac{1}{125}\right) = -1213,008_{(10)}$$

или

$$-14323,001_{(5)} = -\left(\left(\left(1.5+4\right).5+3\right).5+2\right).5+3+\frac{0+\frac{1}{5}}{5}\right) = -\left(\left(9.5+3\right).5+2\right).5+3+\frac{0+\frac{1}{25}}{5}$$

$$= -\left(\left(48.5+2\right).5+3+\frac{1}{125}\right) = -\left(242.5+3+\frac{1}{125}\right) = -\left(1213+\frac{1}{125}\right) = -1213,008_{(10)}.$$

## Пример 21

Търсим  $-1,22_{(3)}=?_{(10)}$  :

$$-1,22_{(3)} = -\left(1.3^{0} + \frac{2}{3^{1}} + \frac{2}{3^{2}}\right) = -\left(1 + \frac{8}{9}\right) = -1,(8)_{(10)}$$

или

$$-1,22_{(3)} = -\left(1 + \frac{2 + \frac{2}{3}}{3}\right) = -\left(1 + \frac{8}{9}\right) = -1,(8)_{(10)}$$

#### Пример 22

Търсим  $243,15_{(6)}=?_{(10)}$  :

$$243,15_{(6)} = 2.6^2 + 4.6^1 + 3.6^0 + \frac{1}{6^1} + \frac{5}{6^2} = 99 + \frac{11}{36} = 99,30(5)_{(10)}$$

или

$$243,15_{(6)} = (2.6+4).6+3+\frac{1+\frac{5}{6}}{6} = 99+\frac{11}{36} = 99,30(5)_{(10)}$$

## 6. Преобразуване на краен непериодичен запис от $\,10$ - ична в $\,s$ - ична ПБС

Начините за тези преобразувания следват от желанието да правим всички изчисления само в 10-ична ПБС.

Цифрите на търсения запис се получават една след друга, като първа се получава тази до дробната запетая, а последна — цифрата, която е най-далеч от дробната запетая.

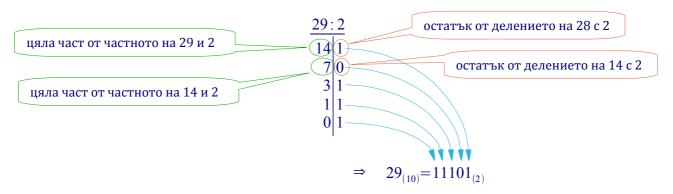
Универсалният алгоритъм за преобразуване на **цяло число** от 10-ична към друга ПБС е следствие от формула 20 при k=1 , т. е.

(31) 
$$\frac{\overline{a_{n-1}...a_1a_0}_{(s)}}{s} = \overline{a_{n-1}...a_1, a_0}_{(s)}.$$

Това равенство е вярно в каквато и ПБС да са записани числата. Следователно, може в 10-ична ПБС да се раздели числото с основата, към която се преминава, и остатъкът от делението е стойността на последната цифра в записа при новата основа, а от цялата част на частното може да се получат останалите търсени цифри.



Такива изчисления се подреждат много по-удобно в таблица от следния вид:



## Пример 24

Търсим 
$$29_{(10)} = ?_{(3)} = ?_{(5)} = ?_{(7)}$$
 :

$$\mathsf{Tърсим} \quad -735_{(10)} = ?_{(2)} = ?_{(3)} = ?_{(7)} :$$

$$\begin{array}{c|c} 735:7 \\ \hline 105 & 0 \\ 15 & 0 \\ 2 & 1 \\ 0 & 2 \end{array} \Rightarrow -735_{(10)} = -2100_{(7)} .$$

## Пример 26

$$\mathsf{Tърсим} \ 347_{(10)} = ?_{(8)} :$$

$$\begin{array}{c|c}
347:8 \\
\hline
43 & 3 \\
5 & 3 \\
0 & 5
\end{array} \Rightarrow 347_{(10)} = 533_{(8)} .$$

## Пример 27

$$T_{\text{ърсим}}$$
 347<sub>(10)</sub> = ?<sub>(16)</sub> :

$$\frac{347:16}{ 21 \begin{vmatrix} 11 \\ 1 \end{vmatrix} 5} \Rightarrow 347_{(10)} = 15b_{(16)}$$
 (16-ичната цифра  $b$  има стойност 11).

## Пример 28

Търсим 
$$-249_{(10)} = ?_{(16)}$$
 :

Универсалният алгоритъм за преобразуване на **правилна дроб** от 10-ична към друга ПБС е следствие от формула 21 при k=1 , т. е.

(32) 
$$\overline{0, c_1 c_2 ... c_n ...}_{(s)}.s = \overline{c_1, c_2 ... c_n ...}_{(s)}$$
.

Това равенство е вярно в каквато и ПБС да са записани числата. Следователно, може в 10-ична ПБС да се умножи числото с основата, към която се преминава, и цялата част от произведението е стойността на първата цифра след дробната запетая в записа при новата основа, а от дробната част на произведението може да се получат останалите търсени цифри.

## Пример 29

Търсим  $0.3125_{(10)} = ?_{(2)}$  :

Тук целите части са стойности на съответните цифри, а от дробните части се получават следващите стойности на цифри.

$$2.0,3125 = 0,625$$
  
 $2.0,625 = 1,25$   
 $2.0,25 = 0,5$   
 $2.0,5 = 1,0$ 
 $\Rightarrow 0,3125_{(10)} = 0,0101_{(2)}$ 

Такива изчисления се подреждат много по-удобно в таблица от следния вид:

цяла част от произведението на 0,3125 и 2

дробна част от произведението на 0,3125 и 2

цяла част от произведението на 0,25 и 2

цяла част от произведението на 0,25 и 2 0 625дробна част от произведението на 0,25 и 2 0 625дробна част от произведението на 0,25 и 2 0 625 0 625дробна част от произведението на 0,25 и 2 0 625

## Пример 30

Tърсим  $0,1875_{(10)} = ?_{(2)}$  :

$$\begin{array}{c|c} \underline{2.0,1875} \\ \hline 0 & 375 \\ 0 & 75 \\ 1 & 5 \\ 1 & 0 \end{array} \Rightarrow 0,1875_{(10)} = 0,0011_{(2)} .$$

### Пример 31

Tърсим  $0,1376_{(10)} = ?_{(5)}$  :

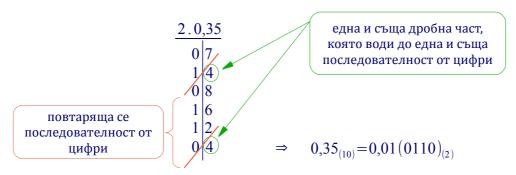
$$\begin{array}{c|c}
5.0,1376 \\
\hline
0 | 688 \\
3 | 44 \\
2 | 2 \\
1 | 0
\end{array} \Rightarrow 0,1376_{(10)} = 0,0321_{(5)} .$$

Търсим  $0.0390625_{(10)} = ?_{(16)}$  :

Когато се преобразува крайна дроб от една в друга ПБС, е възможно да се получи безкрайна периодична дроб. Достигането на края на период се разпознава по това, че два пъти се получава една и съща дробна част, от която се пораждат една и съща последователност от цифри.

## Пример 33

Tърсим  $0,35_{(10)} = ?_{(2)}$  :



## Пример 34

Търсим  $0,1_{(10)}=?_{(3)}$  :

$$\begin{array}{ccc}
3.01 \\
0 & 3 \\
0 & 9 \\
2 & 7 \\
2 & 1
\end{array} \Rightarrow 0,1_{(10)} = 0,(0022)_{(3)} .$$

## Пример 35

Tърсим  $-0.14_{(10)} = ?_{(5)}$  :

$$\frac{5.0,14}{0|7}$$

$$3|5|$$

$$2|5|$$

$$\Rightarrow -0,14_{(10)} = -0,03(2)_{(5)}$$

При преобразуването на крайна 10-ична дроб към 2-ична основа също може да се получи период. В такъв случай, както беше казано на страница 7, съхраняването на краен брой цифри в компютърната памет ще изисква закръгляне, т. е. губи се точност. Обаче при въвеждането от клавиатурата е необходимо да се работи с 10-ични записи, защото те са привични за нас. Затова, ако паметта е 2-ична, когато трябва да се избегнат закръглянията, се използва представяне на числата в паметта като редица от стойности на десетични цифри, което се нарича формат с двоично кодиране на десетичните цифри (често се използва съкращението 2-10-формат). За целта обикновено се съхраняват по две цифри в един байт, което се нарича пакетиран формат, но може и по една цифра в байт, което е непакетиран формат. Съответно изчисленията с такива представяния се правят само по правилата на 10-ичната ПБС.

$$T$$
ърсим  $-0.8_{(10)} = ?_{(7)}$  :

$$\begin{array}{c|cccc}
7.0,8 \\
\hline
5 & 6 \\
4 & 2 \\
1 & 4 \\
2 & 8
\end{array}$$

$$\begin{array}{c}
-0,8_{(10)} = -0,(5412)_{(7)} \\
1 & 2 & 8
\end{array}$$

Тъй като дробната част на числото се записва като дробна част във всяка ПБС и цялата част също се записва като цяла част във всяка ПБС, то двете части се преобразуват поотделно.

## Пример 37

Търсим 
$$-517,325_{(10)} = ?_{(4)}$$
 :

## Пример 38

Търсим 
$$-517,325_{(10)} = ?_{(2)}$$
 :

## Пример 39

Търсим 
$$-325,325_{(10)} = ?_{(5)}$$
 :

# 7. Преобразуване на краен непериодичен запис от ПБС с основа $s^k$ в ПБС с основа $s^n$

Формула 10,  $0 \le \overline{a_{k-1} \dots a_0}_{(s)} < s^k$  , показва, че всяка цифра при основа  $s^k$  на ПБС може да се запише във вида  $\overline{a_{k-1} \dots a_0}_{(s)}$  и всяко число от вида  $\overline{a_{k-1} \dots a_0}_{(s)}$  е стойност на цифра при основа  $s^k$  на ПБС. От тази зависимост следват прости начини за преобразуване на запис от ПБС с основа s в ПБС с основа  $s^k$  и обратно.

Универсалният алгоритъм за преобразуване от основа s към основа  $s^k$  на ПБС е следния: Записът при основа s се разделя на групи по k цифри, започвайки от дробната запетая наляво и надясно, и всяка от получените групи се замества с точно тази цифра при основа  $s^k$ , чиято стойност е записана с групата цифри в s-ична ПБС. При това, за да се получат пълни групи в началото и в края на записа може да се дописват нули (те са незначещи, т. е. записаното число е едно и също и с тях, и без тях).

## Пример 40

Търсим  $11110110000110110,0010011111_{(2)} = ?_{(4)} = ?_{(8)} = ?_{(16)}$  :

 $\Rightarrow$  11110110000110110,001001111<sub>(2)</sub>=132300312,02132<sub>(4)</sub>

 $\Rightarrow$  11110110000110110,001001111<sub>(2)</sub>=366066,117<sub>(8)</sub>.

 $\Rightarrow 11110110000110110,001001111_{(2)} = 1ec36,278_{(16)} \quad (e_{(16)} = 15_{(10)} \quad \text{M} \quad c_{(16)} = 12_{(10)} ).$ 

## Пример 41

Търсим  $-21102012010,012210212_{(3)} = ?_{(9)}$  :

 $\Rightarrow$  -21102012010,012210212<sub>(3)</sub>=-242163,18376<sub>(9)</sub>

Съвсем аналогично, универсалният алгоритъм за преобразуване от основа  $s^k$  към основа s на ПБС е следния: Всяка цифра при основа  $s^k$  се замества k -цифрения s -ичен запис на нейната стойност. Тук е важно да бъде заместена с точно s на брой s -ични, иначе получавания запис би бил за друго число, различно от даденото.

В такива случаи е особено очевидно, колко много изчисления се спестяват, когато се избягва преобразуването от основа 16 на ПБС към основа 10 и от основа 10 към основа 2.

## Пример 43

Търсим 718043,65226<sub>(9)</sub>= $?_{(3)}$ :

 $\Rightarrow -718043,65226_{(9)} = -210122001110,201202022_{(3)}$ 

Универсалният алгоритъм за преобразуване от основа  $s^n$  към основа  $s^k$  на ПБС предвижда преминаване през запис в ПБС с основа s .

## Пример 44

Търсим  $b4fa,7c08_{(16)}=?_{(8)}$ :

$$\Rightarrow b4fa,7c08_{(16)}=132372,37004_{(8)}$$

Строгото математическо доказателство на описаните алгоритми съществено се опира на факта, че всеки две позиции в записа в ПБС с основа  $s^k$  съответствуват на непресичащи се редици от по k позиции в записа в ПБС с основа s , а именно:

Нека  $\overline{x}_{(s^k)} = \overline{x_{k-1} \dots x_0}_{(s)} = x_{k-1} \dots s^{k-1} + \dots + x_0 \dots s^0$  и  $\overline{y}_{(s^k)} = \overline{y_{k-1} \dots y_0}_{(s)} = y_{k-1} \dots s^{k-1} + \dots + y_0 \dots s^0$  . Тогава за всяко число от вида  $\overline{\dots x_0}_{(s^k)} = \overline{y_k}_{(s^k)} = \overline{y_{k-1} \dots y_0}_{(s^k)} = y_{k-1} \dots y_0 \dots s^0$  .

$$\overline{\dots x \dots y \dots} (s^{k}) = \dots + x \cdot (s^{k})^{u} + \dots + y \cdot (s^{k})^{w} + \dots \\
= \dots + (x_{k-1} \cdot s^{k-1} + \dots + x_{0} \cdot s^{0}) \cdot (s^{k})^{u} + \dots + (y_{k-1} \cdot s^{k-1} + \dots + y_{0} \cdot s^{0}) \cdot (s^{k})^{w} + \dots \\
= \dots + (x_{k-1} \cdot s^{k-1} + \dots + x_{0} \cdot s^{0}) \cdot (s^{k \cdot u}) + \dots + (y_{k-1} \cdot s^{k-1} + \dots + y_{0} \cdot s^{0}) \cdot (s^{k \cdot w}) + \dots \\
= \dots + (x_{k-1} \cdot s^{u \cdot k + (k-1)} + \dots + x_{0} \cdot s^{u \cdot k + 0}) + \dots + (y_{k-1} \cdot s^{w \cdot k + (k-1)} + \dots + y_{0} \cdot s^{w \cdot k + 0}) + \dots \\
= \overline{\dots x_{k-1} \dots x_{0} \dots y_{k-1} \dots y_{0} \dots} (s) .$$

Последното преобразование ще бъде коректно, защото от  $u \neq w$  следва, че интервалите [u.k+0;u.k+(k-1)] и [w.k+0;w.k+(k-1)] нямат обща точка .

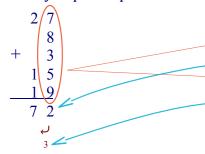
## **8.** Сумиране в ПБС с основа s

Основните аритметични действия се извършват при която и да било основа *s* на ПБС съвсем аналогично на алгоритмите в 10-ичната ПБС.

При сумиране, точно както в 10-ичната ПБС, събираемите се подравняват по дробната запетая и цифрите на сумата се получават една по една отдясно наляво. Единствената разлика между 10-ична ПБС и s-ична ПБС се проявява при възникването на пренос, но има и точна аналогия, която илюстрират следващите примери.

## Пример 45

Сумиране при основа 10 на ПБС:



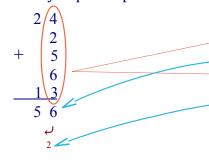
Сумата на цифрите от тази колонка е 32, което е повече от основата 10 на ПБС. Затова 32 се дели с основата 10.

**32=3.10+2**. Затова **2**, т. е. остатъкът от делението на 32 с основата на ПБС, е стойността на **цифрата**, която се записва под колонката в крайната сума. Съответно, **3**, т. е. цялата част от делението на 32 с основата на ПБС, е **преноса** към следващата наляво колонка от цифри на събираемите.

В този и следващите примери всяка двойка оцветени в червено *стрелка* под сумата и *число под стрелката* показват преноса, който възниква от колонката над началото на стрелката към следващата отляво колонка.

## Пример 46

Сумиране при основа 7 на ПБС:

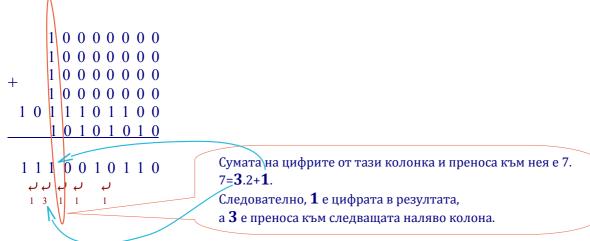


Сумата на цифрите от тази колонка е 20, което е повече от основата 7 на ПБС. Затова 20 се дели с основата 7.

**20=2.7+6**. Затова **6**, т. е. остатъкът от делението на 20 с основата на ПБС е стойността на **цифрата**, която се записва под колонката в крайната сума. Съответно, **2**, т. е. цялата част от делението на 20 с основата на ПБС, е **преноса** към следващата наляво колонка от цифри на събираемите.

## Пример 47

Сумиране при основа 2 на ПБС:

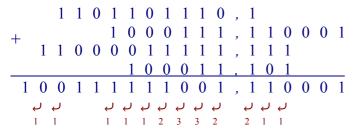


Сумиране при основа 3 на ПБС:

```
\begin{array}{c} 2 & 1 & 2 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & , & 2 \\ + & & 1 & 0 & 2 & 2 & 1 & 0 & 2 & 2 & , & 0 & 1 \\ \hline & 1 & 2 & 1 & 2 & 1 & 1 & 1 & 2 & 2 & , & 0 & 1 \\ \hline 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 2 & 2 & , & 2 & 2 \\ & & \downarrow \\ 1 & 1 & 2 & 1 & 2 & 1 & 1 & 1 & 1 & 1 \end{array}
```

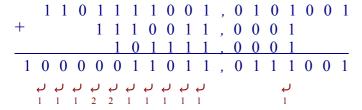
## Пример 49

Сумиране при основа 2 на ПБС:



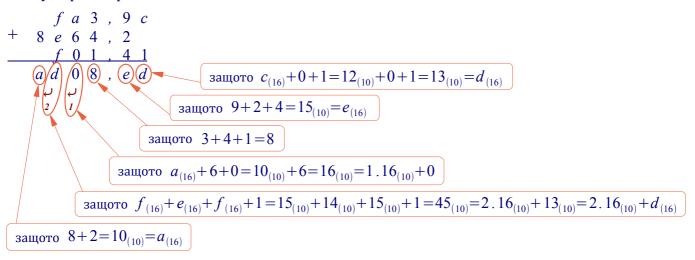
## Пример 50

Сумиране при основа 2 на ПБС:



## Пример 51

Сумиране при основа 16 на ПБС:



## 9. Изваждане в ПБС с основа з

При изваждане, точно както в 10-ичната ПБС, умаляемото и умалителя се подравняват по дробната запетая и цифрите на разликата се получават една по една отдясно наляво. Единственото различие между 10-ична ПБС и s-ична ПБС се появява, когато е необходим заем, но при всички ПБС има и точна аналогия, която илюстрират следващите примери.

Главното правило при изваждането е, че единицата, вземана в заем, преминавайки към съседната позиция надясно, се умножава по основата на ПБС, т. е. във всяка ПБС се взема заем единица, но при основа s на ПБС винаги се получава заем s.

## Пример 52

Изваждане при основа 10 на ПБС:



В този и следващите примери червената точка над цифра показва, че от тази цифра се взема заем една единица и към цифрата отдясно на тази с точката се прибавя като получаван заем толкова, колкото е основата на ПБС.

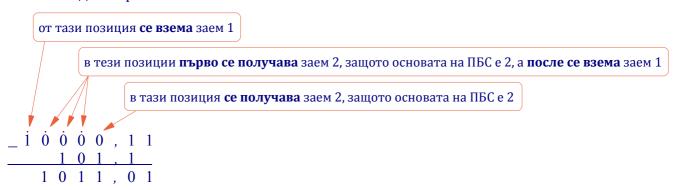
#### Пример 53

Изваждане при основа 4 на ПБС:



## Пример 54

Изваждане при основа 2 на ПБС:



Изваждане при основа 2 на ПБС:



## Пример 56

Изваждане при основа 4 на ПБС:

## Пример 57

Изваждане при основа 16 на ПБС:

$$\begin{array}{c} -3 \quad \dot{f} \quad 1 \quad \dot{a} \quad \dot{9} \quad c \quad , \quad \dot{a} \quad \dot{8} \quad 0 \quad 1 \quad \dot{7} \\ 1 \quad e \quad 8 \quad b \quad f \quad , \quad 9 \quad a \quad 3 \quad 0 \quad 0 \quad 1 \\ \hline 3 \quad \dot{d} \quad 3 \quad 1 \quad \dot{d} \quad \dot{d} \quad , \quad 0 \quad \dot{d} \quad \dot{d} \quad 1 \quad 6 \quad f \\ \hline \qquad \qquad & (0+16_{(10)})-3=13_{(10)}=d_{(16)} \\ \hline \qquad \qquad & (7+16_{(10)})-a_{(16)}=23_{(10)}-10_{(10)}=13_{(10)}=d_{(16)} \\ \hline \qquad \qquad & (c_{(16)}+16_{(10)})-f_{(16)}=(12_{(10)}+16_{(10)})-15_{(10)}=28_{(10)}-15_{(10)}=13_{(10)}=d_{(16)} \\ \hline \qquad \qquad & (8+16_{(10)})-b_{(16)}=24_{(10)}-11_{(10)}=13_{(10)}=d_{(16)} \\ \hline \qquad \qquad & 9-8=1 \\ \hline \qquad \qquad & (1+16_{(10)})-e_{(16)}=17_{(10)}-14_{(10)}=3 \\ \hline \qquad \qquad & 14_{(10)}-1=13_{(10)}=d_{(16)} \\ \hline \end{array}$$

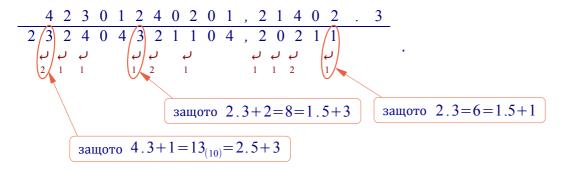
## **10. Умножаване в ПБС с основа** *s*

Точно както при 10-ичната ПБС, умножението на числа се свежда до умножение на число с цифра и събиране.

## Пример 58

Умножение с цифра при основа 10 на ПБС:

Умножение с цифра при основа 5 на ПБС:



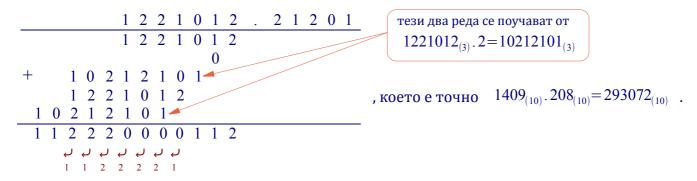
В 2-ична ПБС произведението на число с цифра или е самото число, или е нула.

#### Пример 60

Умножение с цифра при основа 2 на ПБС:

## Пример 61

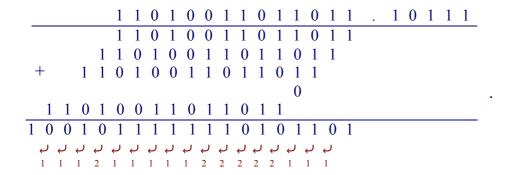
Умножение на числа при основа 3 на ПБС:



В 2-ична ПБС произведението на числа се свежда до преписване с подравняване и сумиране.

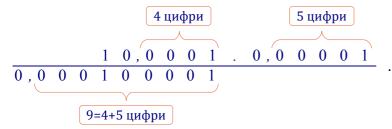
## Пример 62

Умножение на числа при основа 2 на ПБС:



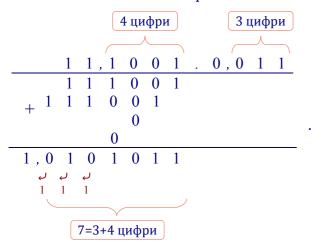
При умножение на числа с дробни части, точно както в 10-ична ПБС, с цифрите се работи както при цели числа, а дробната запетая се поставя в произведението така, че след нея да има такъв брой цифри, колкото е сумата от броевете цифри след дробните запетаи в множителите.

Умножение на числа при основа 2 на ПБС:



## Пример 64

Умножение на числа при основа 2 на ПБС:

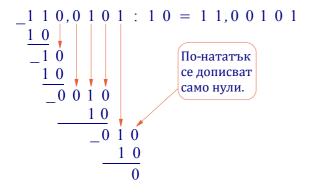


## **11.** Делене в ПБС с основа s

Точно както при 10-ичната ПБС, деленето на числа се свежда до сравняване, умножение на число с цифра и изваждане.

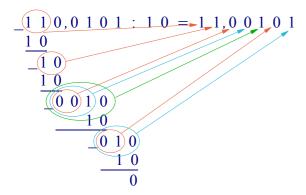
## Пример 65

Делене при основа 2 на ПБС:



Тук стрелките показват участието на цифрите на делимото.

Ето същия пример, като овалите и стрелките показват точно от кое число се получава всяка цифра на частното:



Точно както при 10-ичната ПБС, при делене на крайни записи може да се получи период.

## Пример 66

Делене при основа 2 на ПБС:

$$-1\ 0\ 0\ 1\ 1:\ 1\ 1=1\ 1\ 0, (0\ 1)$$
 $-1\ 1$ 
 $-1\ 1$ 
 $-1\ 1$ 
 $-1\ 1$ 
 $-0\ 1\ 0\ 0$ 
 $1\ 1$ 
 $-1\ 0\ 0$ 
 $-1\ 1$ 
 $-1\ 1$ 
 $-1\ 1$ 
 $-1\ 1$ 
 $-1\ 1$ 
 $-1\ 1$ 
 $-1\ 1$ 
 $-1\ 1$ 
 $-1\ 1$ 
 $-1\ 1$ 

Това е точно 19:3=6,(3) . Верността на горното делене се проверява и от пример 67.

## Пример 67

Делене при основа 2 на ПБС:

Това е точно 15,125:5=3,025 . Верността на горното делене се проверява и от пример 67.

## 12. Преобразуване от една в друга ПБС на записи, съдържащи период

При наличие на период в дадения запис, получаването на записа на същото число при друга основа на ПБС може да стане по поне два относително лесни начина:

Първо, даденият запис, съдържащ период, който трябва да се преобразува в s-ичен вид, може да се преобразува в рационален израз, после да се заместят всички числа от израза с техните s-ични записи и да се извършат действията в s-ична ПБС. Този подход е особено удобен за преминаване от друга към 10-ична ПБС, защото изчисленията ще се провеждат в привичната 10-ична ПБС, но може да се прилага и в посока от всяка към всяка основа.

Второ, към дадения запис, съдържащ период, който трябва да се преобразува в s-ичен вид, може да се приложат описаните по-горе алгоритми за преобразуване на записа, като изчисленията с период се опират на съответните математически правила (или на подходящи разсъждения). Този подход е съществено по-удобен при преминаване от основа 10 към друга основа, защото за десетичните изчисления по-лесно се възприема аритметиката с периодични дроби, но по аналогия може да се използва и за преминаване от друга основа към основа 10.

Следващите три примера илюстрират първия подход, т. е. — чрез получаване на рационален израз, в който участвуват само s -ични числа, и извършване на изчисленията в s -ична ПБС.

## Пример 68

Търсим  $110,(01)_{(2)}=?_{(10)}$  (така се проверява пример 65):

От формула 27 , т. е. 
$$\overline{a_n \dots a_0, f_1 \dots f_j(c_1 \dots c_k)}_{(s)} = \frac{\overline{a_n \dots a_0 f_1 \dots f_j(s)}.(s^k-1) + \overline{c_1 \dots c_k}_{(s)}}{s^j.(s^k-1)} \text{ , ce}$$
 получава 
$$110, (01)_{(2)} = \frac{110_{(2)}.(2^2-1) + 01_{(2)}}{2^0.(2^2-1)} = \frac{6.3+1}{1.3} = \frac{19_{(10)}}{3} = 6, (3)_{(10)} \text{ .}$$

#### Ппимер 69

Търсим  $11,000(0011)_{(2)} = ?_{(10)}$  (така се проверява пример 66):

От формула 27, т. е. 
$$\overline{a_n \dots a_0, f_1 \dots f_j(c_1 \dots c_k)}_{(s)} = \frac{\overline{a_n \dots a_0 f_1 \dots f_j}_{(s)}.(s^k-1) + \overline{c_1 \dots c_k}_{(s)}}{s^j.(s^k-1)} \text{ , ce}$$
 получава 
$$11,000(0011)_{(2)} = \frac{11000_{(2)}.(2^4-1) + 0011_{(2)}}{2^3.(2^4-1)} = \frac{24_{(10)}.15_{(10)} + 3}{8.15_{(10)}} = \frac{363_{(10)}}{120_{(10)}} = 3,025_{(10)} \text{ .}$$

### Пример 70

Търсим  $0,(6)_{(10)}=?_{(7)}$  :

От формула 27 , т. е. 
$$\overline{a_n...a_0,f_1...f_j(c_1...c_k)}_{(s)} = \frac{\overline{a_n...a_0f_1...f_j(s).(s^k-1)} + \overline{c_1...c_k(s)}}{s^j.(s^k-1)} \text{ , ce}$$
 получава 
$$0,(6)_{(10)} = \frac{0_{(10)}.(10^1-1) + 6_{(10)}}{10^0.(10^1-1)} = \frac{6_{(10)}}{9_{(10)}} = \frac{6_{(7)}}{12_{(7)}} \text{ . От деленето в 7-ична ПБС се получава}$$
 
$$-\frac{6,0_{(7)}}{5} : 1 \ 2_{(7)} = 0, (4)_{(7)}$$
 
$$. \Rightarrow 0,(6)_{(10)} = 0,(4)_{(7)} \text{ .}$$

Този резултат може да се провери чрез обратното преобразование:

От формула 27, т.е.  $\overline{a_n...a_0, f_1...f_j(c_1...c_k)}_{(s)} = \overline{\frac{a_n...a_0f_1...f_{j(s)}.(s^k-1) + \overline{c_1...c_k}_{(s)}}{s^j(s^k-1)}}$ , се получава:

$$0, (4)_{(7)} = \frac{0_{(7)} \cdot (7^{1} - 1) + 4_{(7)}}{7^{0} \cdot (7^{1} - 1)} = \frac{4_{(7)}}{6_{(7)}} = \frac{4_{(10)}}{6_{(10)}} = \frac{2_{(10)}}{3_{(10)}} = 0, (6)_{(10)}.$$

Примери 72 и 73 илюстрират втория подход, т. е. — чрез аритметика с периодични дроби.

В тези примери съществено се използва правилото, че при умножаване на период възникващият пренос наляво от периода се прибавя и към самия период.

## Пример 71

$$3.0,0.0(8.5)_{(10)}=0,0.2(5.7)_{(10)}$$

ези два преноса са предизвикани от 3.8+1=25=2.10+5

Също така се използва, че:

(33) 
$$\overline{a_n...a_0, f_1...f_j(c_1...c_kc_{k+1}...c_{k+h})}|_{(s)} = \overline{a_n...a_0, f_1...f_jc_1...c_k(c_{k+1}...c_{k+h}c_1...c_k)}|_{(s)}$$
.

Пример 72

Търсим 
$$0,(4)_{(10)} = ?_{(2)}$$
:

$$\frac{2.0,(4)}{0(8)}$$
 $1(7)$ 
 $1(5)$   $\Rightarrow$   $0,(4)_{(10)} = 0,(011100)_{(2)}$ .

1(1)
0(2)
0(4)

Проверка чрез първия подход:

От формула 27 , т. е. 
$$\overline{a_n...a_0,f_1...f_j(c_1...c_k)}_{(s)} = \frac{\overline{a_n...a_0f_1...f_j(s)}.(s^k-1) + \overline{c_1...c_k}_{(s)}}{s^j.(s^k-1)} \text{ , ce}$$
 получава 
$$0,(011100)_{(2)} = \frac{0.(2^6-1) + 011100_{(2)}}{2^0.(2^6-1)} = \frac{28_{(10)}}{63_{(10)}} = 0,(4)_{(10)} \text{ .}$$

$$T_{\text{ърсим}}$$
 -11201,002 (30)<sub>(4)</sub>=?<sub>(10)</sub> :

$$(((1.4+1).4+2).4+0).4+1=353_{(10)}$$
.

$$10_{(10)} = 22_{(4)} \hspace*{0.2cm} ; \hspace*{0.2cm} \frac{0,002(30)_{(4)}.22_{(4)}}{+ 11(21) \atop 112(12) \atop 0,123(33)_{(4)}} \hspace*{0.2cm} ; \hspace*{0.2cm} \frac{0,123(33)_{(4)}.22_{(4)}}{+ 313(33) \atop 3133(33) \atop 10,113(33)_{(4)}} \hspace*{0.2cm} ; \hspace*{0.2cm} \frac{0,113(33)_{(4)}.22_{(4)}}{+ 233(33) \atop 2333(33) \atop 3,233(33)_{(4)}} \hspace*{0.2cm} ;$$

$$\frac{0,233 \left(33\right)_{(4)}.22_{(4)}}{+ \frac{1 \ 133 \left(33\right)}{13,133 \left(33\right)_{(4)}}} \; ; \; \frac{0,133 \left(33\right)_{(4)}.22_{(4)}}{+ \frac{333 \left(33\right)}{10,333 \left(33\right)_{(4)}}} \; ; \; \frac{0,333 \left(33\right)_{(4)}.22_{(4)}}{+ \frac{1 \ 333 \left(33\right)}{13 \ 333 \left(33\right)}} ( тук се използва формула 33 ).$$

$$\Rightarrow \begin{array}{c|c} 22_{(4)} \cdot 0,002(30)_{(4)} \\ \hline 0_{(4)} & 123(33)_{(4)} \\ 10_{(4)} & 113(33)_{(4)} \\ 3_{(4)} & 233(33)_{(4)} \\ 13_{(4)} & 133(33)_{(4)} \\ 10_{(4)} & 333(33)_{(4)} \\ 21_{(4)} & 333(33)_{(4)} \end{array}$$

$$10_{(4)} = 4_{(10)} ; 13_{(4)} = 7_{(10)} ; 21_{(4)} = 9_{(10)} ; \Rightarrow -11201,002 (30)_{(4)} = -353,04374 (9)_{(10)} = -353,04375_{(10)} .$$

Проверка чрез първия подход:

От формула 27 , т.е. 
$$\overline{a_n...a_0}, f_1...f_j(c_1...c_k)_{(s)} = \frac{\overline{a_n...a_0}f_1...f_j_{(s)}.(s^k-1) + \overline{c_1...c_k}_{(s)}}{s^j.(s^k-1)}$$
 , се получава:

$$-11201,002\left(30\right)_{\!(4)} = -\frac{11201002_{(4)}.\left(4^2-1\right)+30_{(4)}}{4^3.\left(4^2-1\right)} = -\frac{22594_{(10)}.\,15_{(10)}+12_{(10)}}{64_{(10)}.\,15_{(10)}} = -\frac{338922_{(10)}}{960_{(10)}} = -353,04375_{(10)}$$

При преобразуване от основа s всяка цифра от периода се заменя, както и извън периода. След това, понякога, е възможно да се намалят цифрите пред периода.

## Пример 74

Търсим  $6,354(172)_{(8)} = ?_{(2)}$  :

$$\Rightarrow 6,354(172)_{(8)} = 110,011101100(001111010)_{(2)} = 110,01110110(000111101)_{(2)}.$$

При преобразуване от основа s към основа  $s^k$  първо трябва числото да се представи по еквивалентен начин така, че периодът да се разделя точно на цяло число групи от по k цифри.

$$T_{\mathbf{ърсим}} \quad 0.102(221)_{(3)} = ?_{(9)} :$$

$$0,102(221)_{(3)} = 0,1022(212)_{(3)} = 0,1022(212212)_{(3)}$$

$$\begin{array}{c|ccccc}
0 & 1 & 0 & 2 & 2 & 2 & 2 & 1 & 2 & 2 & 2 & 3 \\
0 & 3 & 8 & 7 & 8 & 5 & 9 & 9 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0$$