

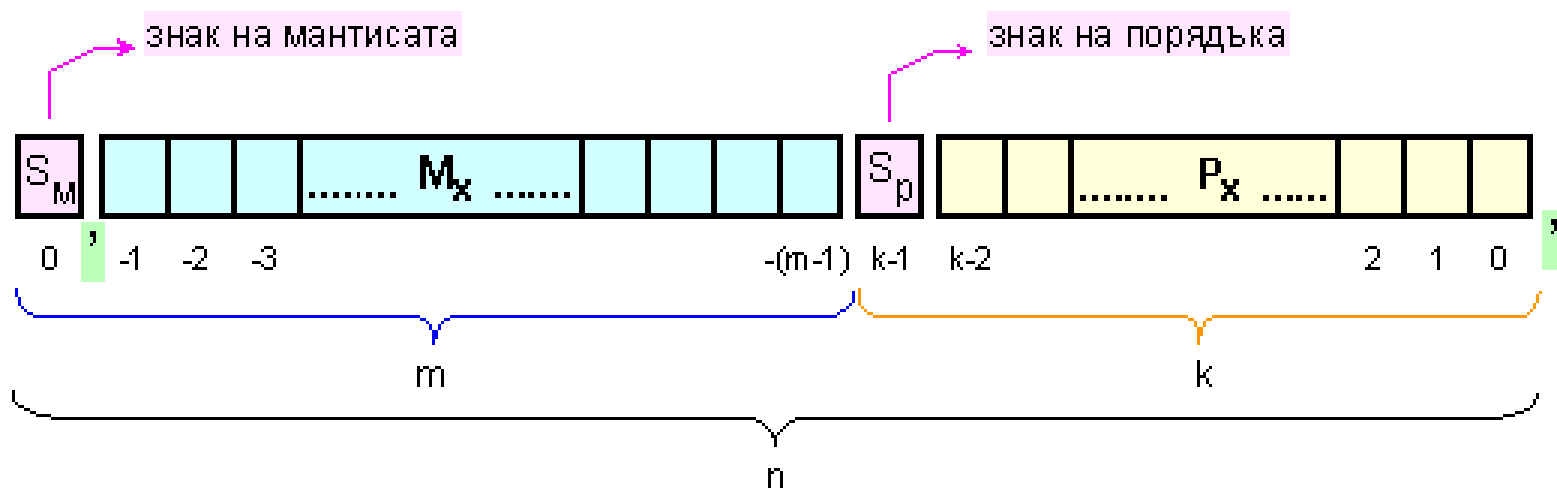
(Приложението не е самостоятелна единица, а онагледява преподаван по предмета урок)

**Кодове за представяне на данни в КС – Числа с плаваща запетая;  
BCD, ASCII и UTF.**

**Дробни числа с плаваща запетая**

Позицията на запетаята в полето, в което се записват числата не е фиксирано. Числата с плаваща запетая се дефинират с т.нар. мантиса (дробна част)  $m$  и порядък (експонента)  $p$ .

$$(\pm M_x, \pm p_x) :$$



Едно

десетично число  $N$  може да се представи като:

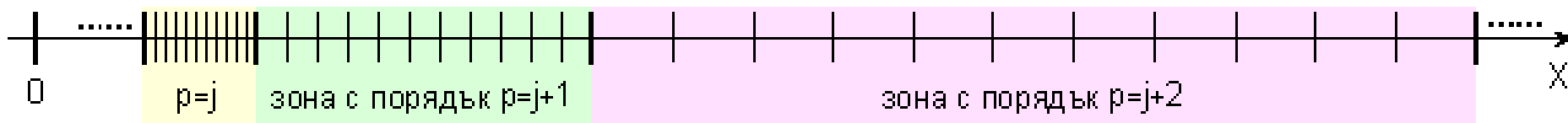
$$N = m 10^p.$$

**Пример:**  $-374.25$  може да се запише във вида:  
 $-3.7425 \times 10^2$ .

В този запис  $m = -3.7425$ , а  $p = 2$ .

**Пример:**  $0.000453$  може да се запише като  
 $4.53 \times 10^{-4}$   
мантисата е  $m = 4.53$ , а порядъкът е  $p = -4$ .

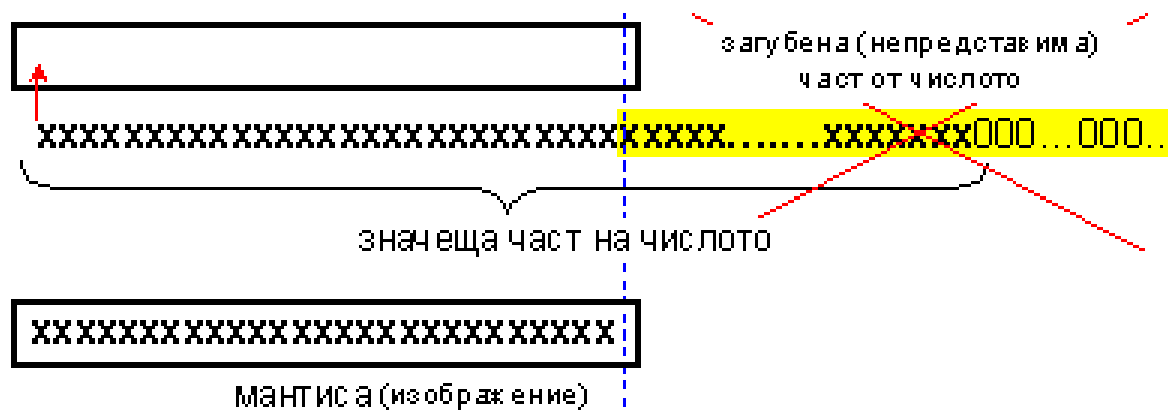
При експоненциалния запис на числата, с нарастването на порядъка се получава грешка в стойността, тъй като при порядък 1 единиците в мантисата имат теглови



коэффициент „10”, при порядък 2 - "100", т.е. най-малката стъпка на запис е през 100 и т.н.

Двоичните числа с плаваща запетая се представят в паметта на компютърните

системи, като двоичното число се разделя на две части - едната за запис на мантисата, а другата за порядъка. Мантисата и порядъкът се записват като **числа със знак**.



**Пример:** 01100000  
 1011<sub>2</sub> мантисата е 1  
 байт,  
 а порядъка 4 бита  
 или **m =**  
**01100000,**

$$a \quad p = 1011$$

$$\text{и тогава } N = 96 \times 2^{-5} = 3$$

**IEEE 754** е стандартът регламентиращ формата на числата с плаваща запетая.

Този стандарт се използва за представяне на дробни числа в съвременните МПС-ми, използващи Intel процесори, в Apple Mac и Unix платформите. Ако знаковият бит е 0 – числото е положително, ако е 1-ца е отрицателно.

**За да се избегне загуба на точност при математически операции, се препоръчва първо да се извършват изчисленията с числа от близки порядъци.**

Проверка на равенство се извършва за диапазон +/- делта.

**Проверка на равенство между числа с плаваща запетая**

**AND(A12>(D12-E10);A12<(D12+E10))**

задаване на

точност - 0,0001

0,3579**9** = 0,3579**8** TRUE

Ако в същият пример за прегледност се се търси сравнение до втория знак след десетичната запетая, коефициента на точност ще е 0,01.

Двете числа с плаваща запетая се различават в петия знак, което би довело до отговор за неравенство при опит за директна проверка със знак равно.

Ако към първото число се добави коефициента на точност и резултатът е по-голям от второто  $0,3579**9** + 0,01 = 0,3**6**799 > 0,3**5**79**8**$

И ако от първото число се извади коефициента на точност и резултатът е по-малък от второто  $0,3579**9** - 0,01 = 0,3**4**799 < 0,35**7**9**8**$

Тогава с операцията „логическо И” се обединяват двете условия и се отчита дали са равни да знакът след десетичната запетая, които се указва с коефициента на точност.

### Изчисляване на степенуване и коренуване чрез логаритмуване

a	b	c	$(C1/E1) * (Ln(A1))$	EXP(F1)
9	^	1 / 2	1,098612289	3
3	^	2 / 1	2,197224577	9

В групата на числовите данни самостоятелно значение имат двоично кодирани десетични числа - BCD-числа (BCD – *Binary Code Decimal*), наричани също **двоично-десетични**. Друго често срещано название е 8421-BCD код.

При този вид представяне на цели числа се задава пряка връзка между цифрите на десетичното число и техния двоичен код и всяка десетична цифра се записва като 4-битов

двоичен код. А забранени (неизползвани) комбинации са:

10 – 1010; 11 – 1011; 12 – 1100; 13 – 1101; 14 – 1110 и 15 - 1111

Например числото 268,47 се кодира така:

268,47 ≡ 0010 0110 1000 , 0100 0111

Може да се каже, че представянето на десетичните цифри чрез техните двоични кодови комбинации, е своеобразна имитация на десетичните числа с машинни средства и на преобразуването в бройни системи с основа кратна на 2. Например поразрядните преобразувания между осмична и двоична и между шестнадесетична и двоична бройни системи.

Двоично-десетичните числа се представят най-често във форма с фиксирана запетая, т.е. като последователност (низ) от двоични кодове на десетичните цифри на числото. Естественият произход на този низ обикновено е процес на въвеждане на данни от клавиатурата.

Форматите за съхраняване на двоично-десетични числа са два: *непакетиран* и *пакетиран*.

*Непакетираният формат* представя десетичното число като последователност от кодовете на отделните цифри, знак, десетичен разделител (,) и др., според кодовата таблица на използвания стандарт, т.е. съществуват като символен низ. При това положение младшите 4 бита на символния код представляват кода на десетичната цифра. Така например число 7095, според ASCII таблицата ще има следния непакетиран формат:

0011 0111	0011 0000	0011 1001	0011 0101
K3	K3	K3	K3
7	0	9	5

където K3 означава ASCII-код на зоната. В този формат обикновено числата съществуват непосредствено след тяхното въвеждане от клавиатурата или преди непосредственото им визуализиране на екрана на монитора.

*Пакетираният формат* на едно двоично-десетично число представлява последователност от 4-битови двоични кодове на неговите цифри. Например горното число 7095 може да се представи в пакетирани формат чрез последователността: <cd7><cd0><cd9><cd5>, която се побира в поле с дължина 2[B], т.е. всеки 4 бита представляват код на десетична цифра (cd). Когато двоично-десетичното

число се представя като число със знак, последният се кодира с 4-битова кодова комбинация, несъответстваща на десетична цифра. Пакетираният формат е удобен и като вътрешно-машинен при непосредственото обработване на двоично-десетичните числа. От гледна точка на обработването обаче знакът на числото може да се представя от десетична цифра.

Форматите на двоично-десетичните числа не са пряко свързани с форматите на разрядната мрежа на процесора, тъй като десетичната бройна система не е основна (не е оптимална) бройна система.

## Представяне на символна информация в МП С-ми

В МП системи информацията се съхранява само в двоичен вид. Използваните символи също се представят като комбинация от двоични числа.

При натискане на клавиш от клавиатурата се формира определен код наречен **scan cod.** код който не е свързан със символа на клавиша, а с поредния номер на клавиша.

Връзката между 'scan cod'-а и символа се определя от специална таблица. В нея на всеки символ, използван от компютърната система, се присвоява двоично число (байт или 2 байта).



Така текстът, записан в паметта на компютъра, представлява последователност от **байтове**, съответстващи на символите от текста.

Най-често в КС се използват ASCII и Unicode таблиците. Те са се превърнали в стандарт, чрез който може да се обменя информация между различни КС.

# ASCII (American Standard Code for Information Interchange)

Когато е разработван този стандарт, се е смятало, че всички, използвани в

ASCII control characters		ASCII printable characters						
00	NULL	(Null character)	32	space	64	@	96	`
01	SOH	(Start of Header)	33	!	65	A	97	a
02	STX	(Start of Text)	34	"	66	B	98	b
03	ETX	(End of Text)	35	#	67	C	99	c
04	EOT	(End of Trans.)	36	\$	68	D	100	d
05	ENQ	(Enquiry)	37	%	69	E	101	e
06	ACK	(Acknowledgement)	38	&	70	F	102	f
07	BEL	(Bell)	39	'	71	G	103	g
08	BS	(Backspace)	40	(	72	H	104	h
09	HT	(Horizontal Tab)	41	)	73	I	105	i
10	LF	(Line feed)	42	*	74	J	106	j
11	VT	(Vertical Tab)	43	+	75	K	107	k
12	FF	(Form feed)	44	,	76	L	108	l
13	CR	(Carriage return)	45	-	77	M	109	m
14	SO	(Shift Out)	46	.	78	N	110	n
15	SI	(Shift In)	47	/	79	O	111	o
16	DLE	(Data link escape)	48	0	80	P	112	p
17	DC1	(Device control 1)	49	1	81	Q	113	q
18	DC2	(Device control 2)	50	2	82	R	114	r
19	DC3	(Device control 3)	51	3	83	S	115	s
20	DC4	(Device control 4)	52	4	84	T	116	t
21	NAK	(Negative acknowl.)	53	5	85	U	117	u
22	SYN	(Synchronous idle)	54	6	86	V	118	v
23	ETB	(End of trans. block)	55	7	87	W	119	w
24	CAN	(Cancel)	56	8	88	X	120	x
25	EM	(End of medium)	57	9	89	Y	121	y
26	SUB	(Substitute)	58	:	90	Z	122	z
27	ESC	(Escape)	59	;	91	[	123	{
28	FS	(File separator)	60	<	92	\	124	
29	GS	(Group separator)	61	=	93	]	125	}
30	RS	(Record separator)	62	>	94	^	126	~
31	US	(Unit separator)	63	?	95	_		
127	DEL	(Delete)						

Extended ASCII characters							
128	Ç	160	á	192	Ł	224	Ó
129	ü	161	í	193	ł	225	ó
130	é	162	ó	194	Ł	226	Ô
131	â	163	ú	195	ł	227	ô
132	ä	164	ñ	196	—	228	ö
133	à	165	Ñ	197	†	229	Õ
134	á	166	ª	198	ä	230	μ
135	ç	167	º	199	Ä	231	ρ
136	ê	168	¿	200	Ł	232	ρ
137	ë	169	®	201	ł	233	Ů
138	è	170	™	202	Ł	234	Ů
139	ï	171	½	203	ł	235	Ů
140	ì	172	¼	204	Ł	236	Ÿ
141	í	173	¾	205	ł	237	Ÿ
142	Ä	174	«	206	†	238	—
143	Å	175	»	207	‡	239	·
144	É	176	⋮	208	ø	240	≡
145	æ	177	⋮	209	Ð	241	±
146	Æ	178	⋮	210	È	242	—
147	ô	179		211	É	243	¼
148	ö	180	┆	212	È	244	¶
149	ò	181	À	213	Í	245	§
150	û	182	Á	214	İ	246	÷
151	ù	183	Â	215	ı	247	°
152	ÿ	184	©	216	İ	248	°
153	Ö	185	⌈	217	Ј	249	·
154	Ü	186	⌋	218	Г	250	·
155	ø	187	┆	219	█	251	·
156	£	188	┆	220	█	252	·
157	∅	189	¢	221	ı	253	·
158	x	190	¥	222	ı	254	·
159	f	191	₯	223	█	255	nbsp

изчислителни  
те машини  
символи  
Символите  
с номера над  
**128** се  
използват за  
**други**  
**символи и**  
**азбуки,**  
например за  
символите на  
кирилицата.  
Това е  
причината  
някои

текстове писани на кирилица, да са нечетими или да са с променени букви, когато се смени шрифтът.

Пример със стандартни функции за работа с кодовата таблица.

**function** chr(kod: **integer**): **char** - преобразува число - код на символ в самия символ.

**function** ord(ch: **char**): **integer** - преобразува символ в съответното му число - код на символа.

{ Преобразуване на големи букви кирилица в малки кирилица }

if (ch >= 'А') **and** (ch <= 'Я')

**then** ch := chr(ord('а') + ord(ch) - ord('А'));

Кодове на символни  
множества

CODE(J2)

CHAR(N2)

simb	code	simb	code
a	97	0	48
A	65	1	49
z	122	9	57
Z	90	A	65
0	48	Z	90
1	49	a	97
9	57	z	122
a	224	A	192
я	255	Я	223
A	192	а	224
Я	223	я	255

Поради нарастналата необходимост от повече символи се въвежда системата за кодиране **Unicode**. С нея работят *съвременните операционни системи*. При тази

система, 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  $2^{16} = 65536$  различни символа.

Например в тази система първият байт на всеки символ от кирилицата съдържа числото 204 като код на азбуката.