

Архитектура компьютера.

Биты и манипулирование ими.

*«Но да будет слово ваше: да, да;
нет, нет; а что сверх того, то от
лукавого.»*

- Евангелие от Матфея 5, 37

Компьютер решает задачи в соответствии с алгоритмом, представленным в виде *машинного кода* – последовательности нулей и единиц.

Бит – двоичный разряд, имеющий два значения – нуль или единицу.

Теоретическим основанием для технических реализаций систем, манипулирующих битами является *булева алгебра* (или изоморфные ей математические структуры – *алгебра высказываний* и *алгебра логики*).

На множестве из двух элементов – 0 и 1 (или «правда» и «ложь», или «да» и «нет») заданы две бинарные операции – конъюнкция *and* и дизъюнкция *or*, и одна унарная – *not*.

**Свойства
логически
х
операций:**

закон двойного отрицания: $\text{not not } a = a$

закон коммутативности:

$$a \text{ or } b = b \text{ or } a$$

$$a \text{ and } b = b \text{ and } a$$

закон ассоциативности:

$$a \text{ or } (b \text{ or } c) = (a \text{ or } b) \text{ or } c$$

$$a \text{ and } (b \text{ and } c) = (a \text{ and } b) \text{ and } c$$

закон дистрибутивности:

$$a \text{ or } (b \text{ and } c) = (a \text{ or } b) \text{ and } (a \text{ or } c)$$

$$a \text{ and } (b \text{ or } c) = (a \text{ and } b) \text{ or } (a \text{ and } c)$$

правила де Моргана:

$$\text{not } (a \text{ or } b) = \text{not } a \text{ and } \text{not } b$$

$$\text{not } (a \text{ and } b) = \text{not } a \text{ or } \text{not } b$$

Дополнительная операция – «исключающее или»

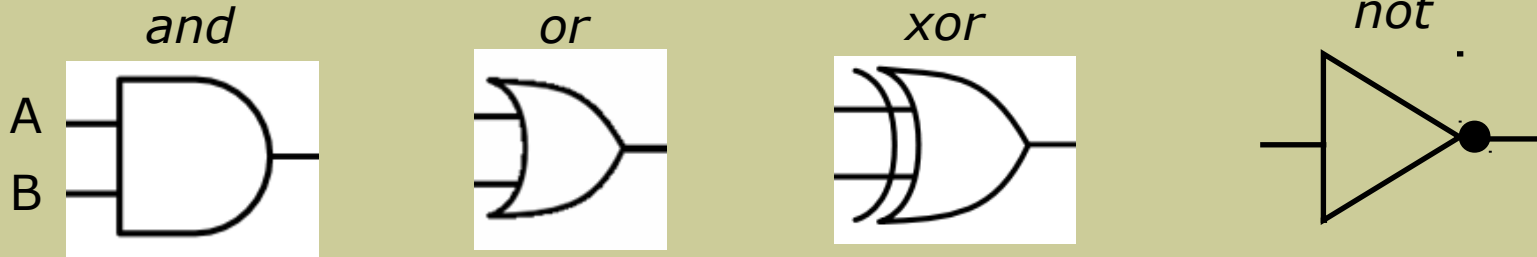
xor

Таблица

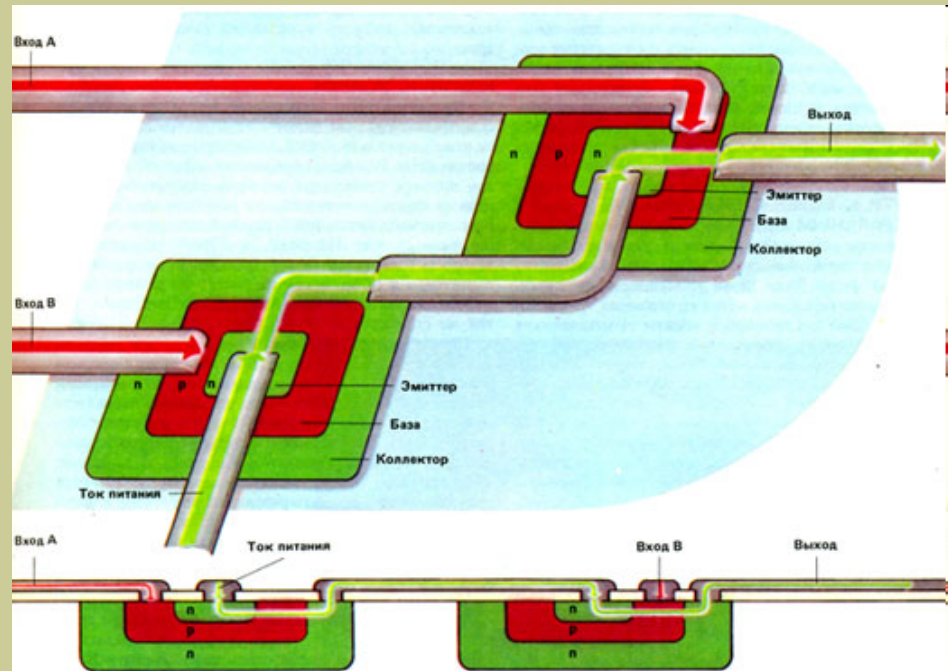
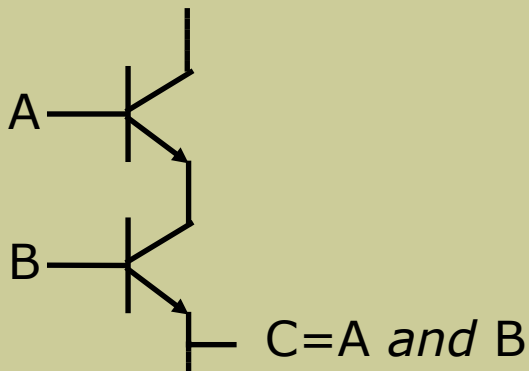
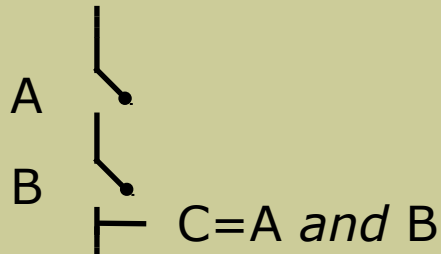
истинности:

<i>a</i>	<i>b</i>	<i>and</i>	<i>or</i>	<i>xor</i>	<i>a</i>	<i>not</i>
0	0	0	0	0	0	1
0	1	0	1	1	1	0
1	0	0	1	1		
1	1	1	1	0		

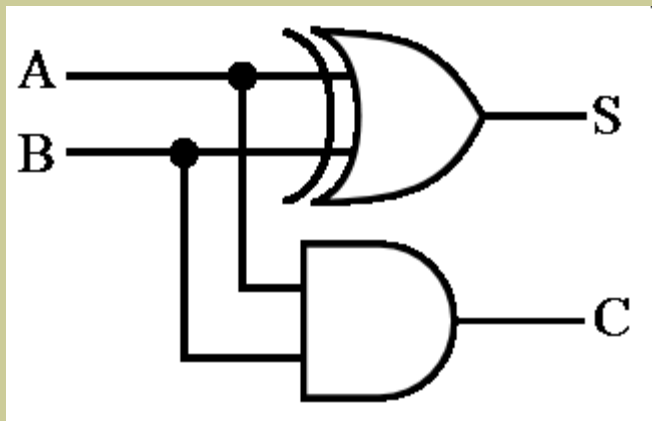
Абстрактные устройства, реализующие логические операции (*вентили*):



Техническая реализация вентилей:

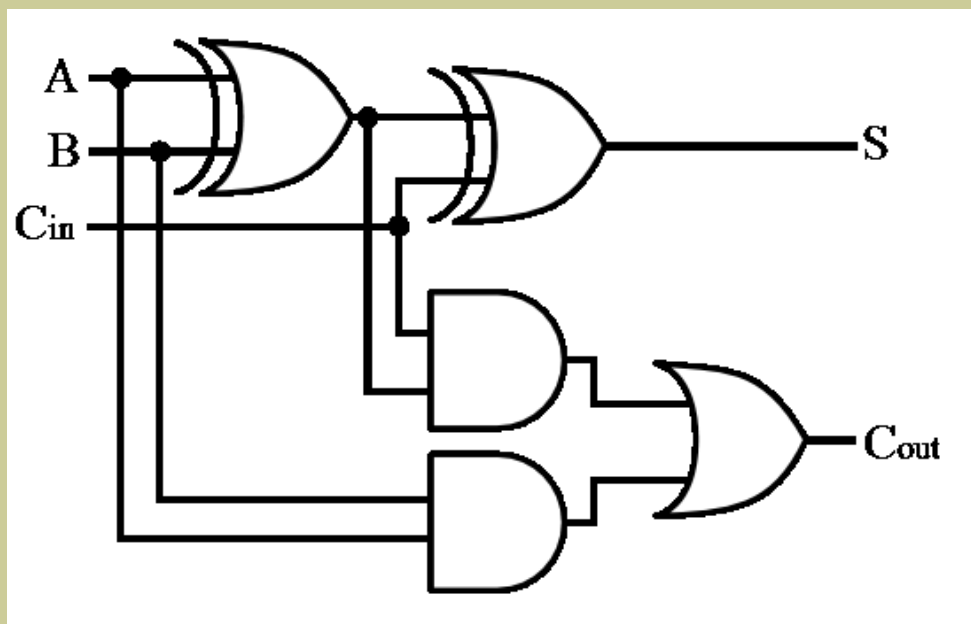


Двоичный полусумматор:



A	B	S	C
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

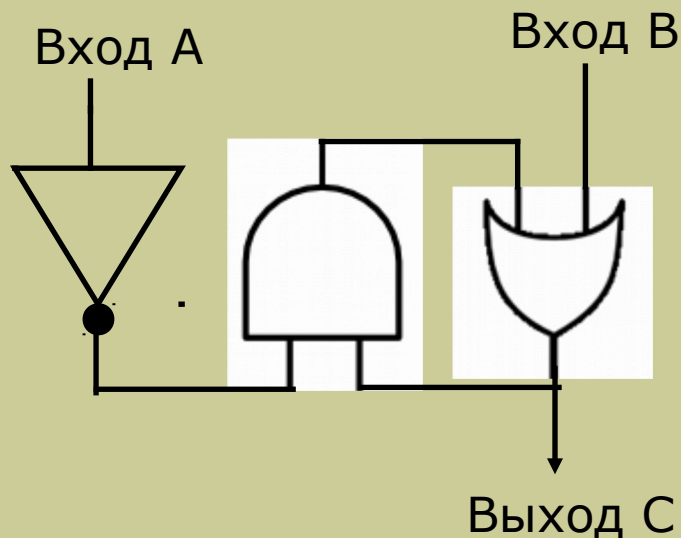
Полный двоичный сумматор:



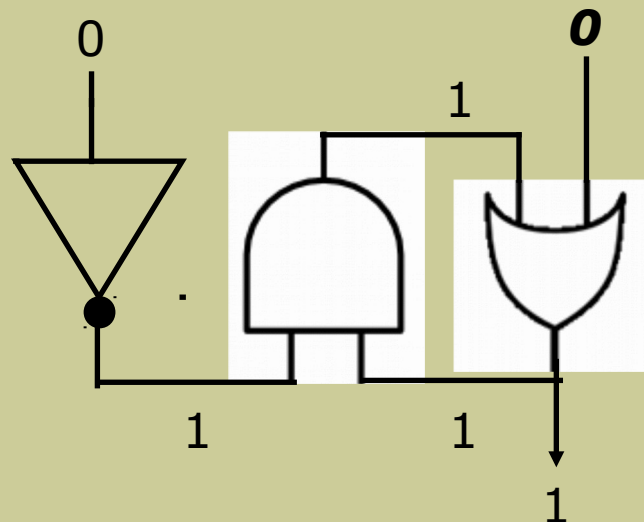
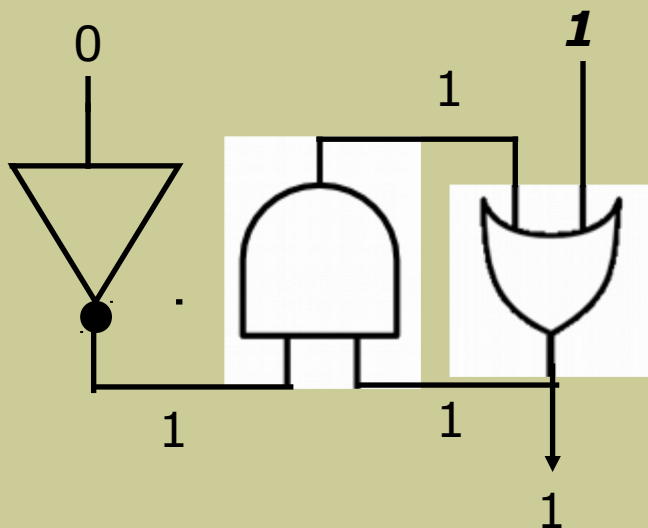
A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
1	0	0	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

Триггер:

Биты и манипулирование ими.



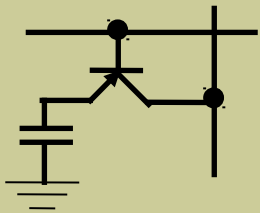
Подача сигнала на вход В устанавливает триггер в состояние 1. После снятия напряжения с этого входа триггер остается в этом состоянии. Для перехода триггера в состояние 0 необходимо подать сигнал на вход А.



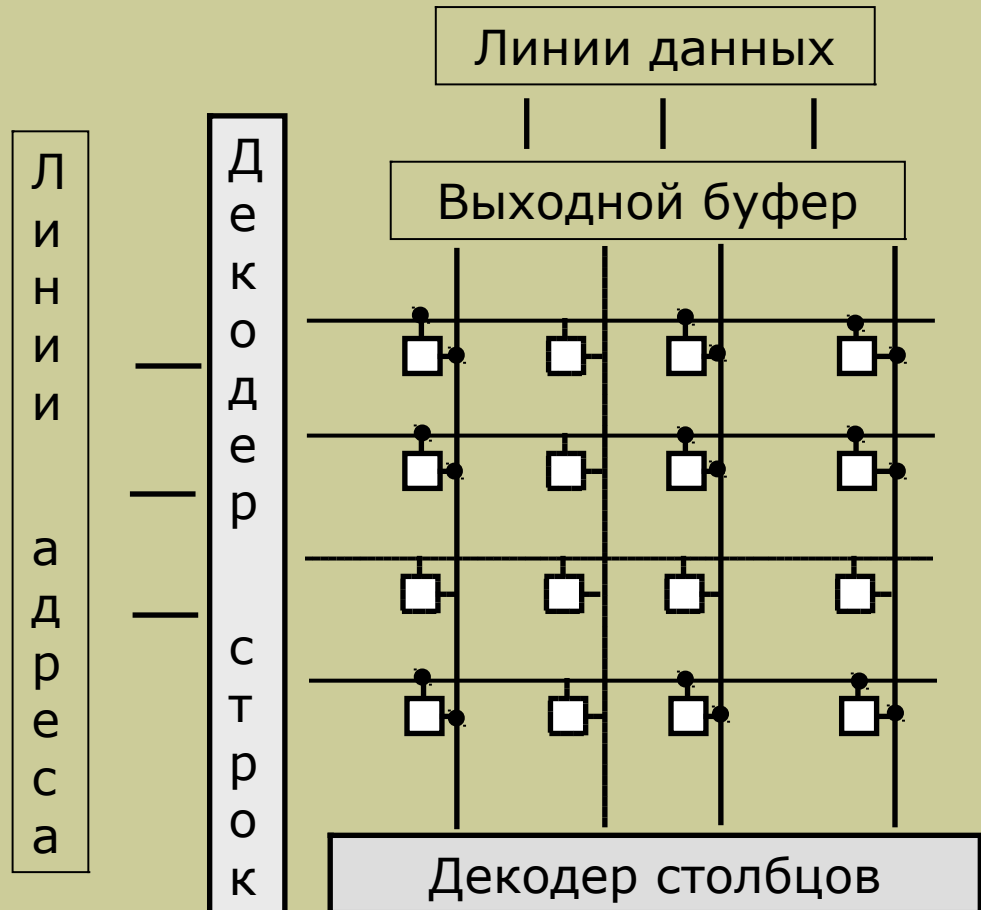
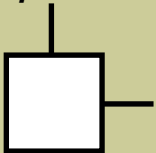
Техническая реализация триггера дорогостоящая, поэтому биты с помощью триггеров хранят в небольшой по объему памяти, но с большим быстродействием. Эта статическая память – *SRAM*, используется в персональных компьютерах для *регистров* и *кэшей*.

Оперативная память основана на технологии динамической памяти – *DRAM*, использующей конденсаторы для хранения бит.

Элемент *DRAM*:



И его
схематическое
изображение:



В ОЗУ персональных компьютеров принята линейная побайтовая адресация памяти. Каждая ячейка памяти – байт, имеет свой адрес от нуля до N (емкость памяти).

Адресация памяти предоставляет возможность доступа к произвольной ячейке – *память с произвольным доступом (RAM)* (см. Лекцию 3).

Основные характеристики памяти - *емкость (размер) и пропускная способность.*

Пропускная способность определяется

- тактовой частотой памяти;
- шириной шины памяти;
- количеством бит на линию за такт (технология DDR – 2 бита за такт)

Частоте в 1МГц
соответствует время
1000 нс.

Например (DDR SDRAM): частота памяти 200 МГц, значит эффективная частота – 400 МГц, разрядность шины 64 бит.

Пропускная способность = $400 \cdot 64 = 25600 \text{ Мбит/с} = 3.2 \text{ Гб/с}$

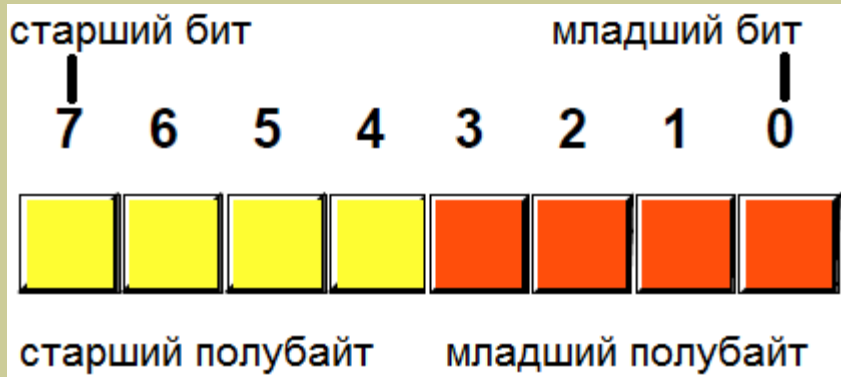
Архитектура компьютера.

Организация памяти.

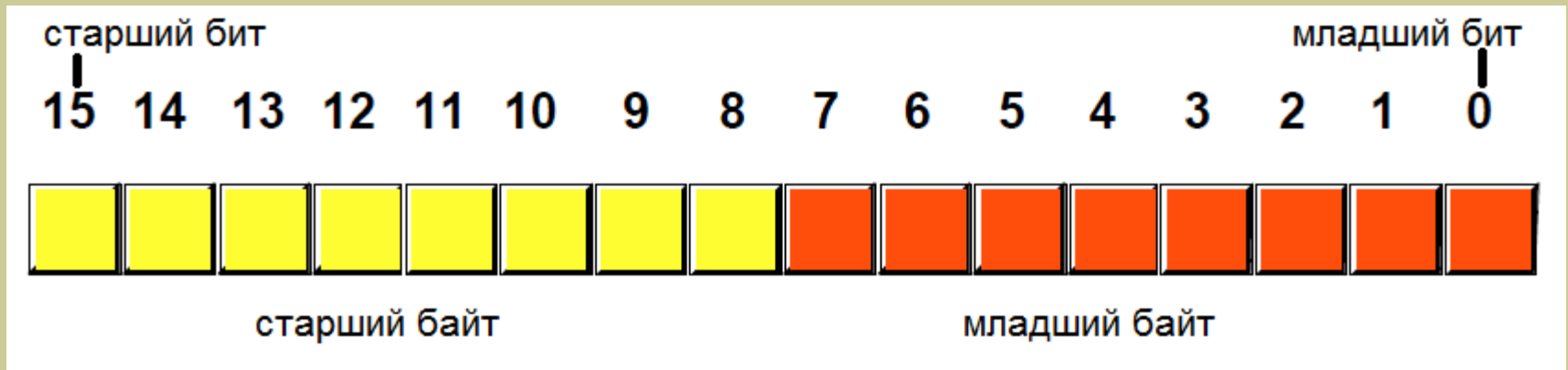
«**В начале было слово,**
но это не было
фиксированное число
битов»

- Р.С. Бартон

Б
а
й
т

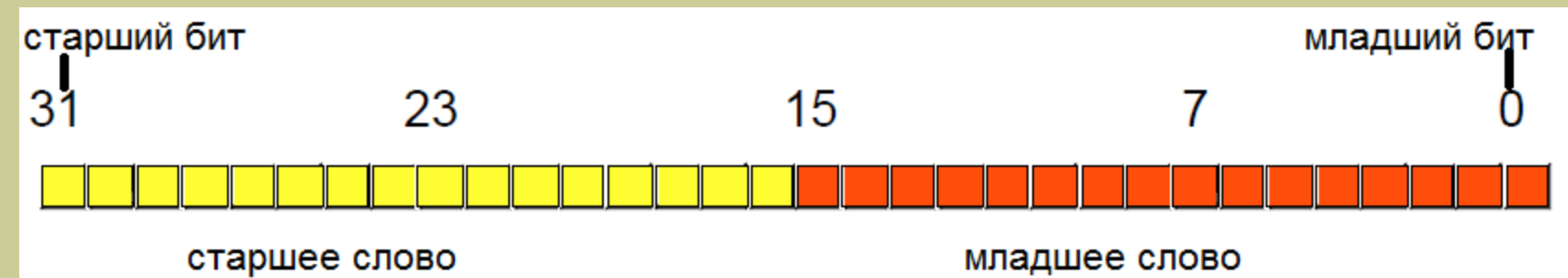


С
л
о
в
о



Д
в
о
й
н
о
е

С
л
о
в
о



Представление информации в виде двоичного кода.

Двоичный код.

Двоичная система счисления.

Представление целых числовых значений.

$$b = b_0 + b_1 * 2^1 + b_2 * 2^2 + \dots + b_{n-1} * 2^{n-1}$$

Программа

Ввод a

i:=0

Выполнять

b_i:=остаток(a, 2)

a:=частное(a,2)

i:=i+1

До a=0

Вывод {b_i}

Конец

Программа

Ввод {b_i}, n

a:=b₀

d:=2

i:=1

Цикл-пока i<n

a:=a+b_i*d

d:=d*2

i:=i+1

Конец-цикл

Вывод a

Конец

Представление дробной части.

$$a = b_1 * 2^{-1} + b_2 * 2^{-2} + \dots + b_n * 2^{-n}$$

Двоичный дополнительный код.

Процессоры семейства 80x86 трактуют отрицательные числа, как двоичные дополнения (которые содержат единичный бит в старшем разряде). Чтобы получить отрицательное число надо инвертировать все биты и добавить единицу.

Например (пусть единица хранения – 1 байт):

Проверяем:

00000101	пять
11111010	инвертируем биты
<u> +1</u>	добавляем единицу
11111011	минус пять

00000101
+11111011
1 00000000

(в прямом двоичном коде минус пять записывается 10000101)

Вопрос: как перевести отрицательное число в двоичном дополнительном коде в прямую десятичную запись?

Вопрос: 10000000 - какое это число?

Представление со смещением (с избытком).

Пример: 3-х битовое представление
(со смещением 4)

000	001	010	011	100	101	110	111
-4	-3	-2	-1	0	1	2	3

Вопрос: каково смещение в 1-байтовом представлении с избытком?

Представление с плавающей точкой.

Нормализованная запись числа: мантисса всегда меньше единицы и её первый разряд содержит отличную от нуля цифру (в двоичной системе счисления - единицу).

В общем случае запись числа A имеет вид:

$$A = (\pm M) \times Q^{\pm P},$$

где M – мантисса, Q – основание системы счисления, P – порядок числа.

Пример: расшифруем число 01101011, записанное в формате с плавающей точкой (единица хранения – 1 байт, старший бит – знаковый, младшие 4 бита – мантисса, остальные три бита – порядок (записан в формате со смещением)).

0	110	1011
Знак	Порядок	Мантисса

.1011 – мантисса. В 3-х битовом представлении со смещением 110 – это два. Переносим точку вправо на два разряда: 10.11. Целая часть числа – 10 равна двум. Дробная часть числа – 11 равна $1/2 + 1/4 = 3/4$. Итак, 01101011 – это запись числа 2.75.

Стандарт IEEE записи чисел с плавающей точкой одинарной точности: старший бит – знак, 8 младших бит – порядок, остальные 23 бита – мантисса. Кроме того, не записывается первый бит мантиссы («скрытый»).

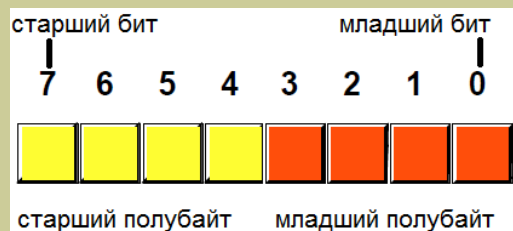
Вопрос: каков диапазон значений в такой записи?

Представление текста.

ASCII код

	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.A	.B	.C	.D	.E	.F
0.	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	TAB	LF	VT	FF	CR	SO	SI
1.	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2.		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3.	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4.	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5.	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6.	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7.	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

Примечание (16-ричное представление):



В полубайте можно кодировать числа со значениями от 0 до 15. Для записи содержимого байта удобно использовать систему счисления с основанием 16.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Кодовые страницы.

Windows - 1251

	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.A	.B	.C	.D	.E	.F
8.	Ђ 402	Ѓ 403	, 201A	Ѕ 453	” 201E	… 2026	† 2020	‡ 2021	€ 20AC	‰ 2030	Љ 409	‘ 2039	Њ 40A	Ќ 40C	Ѡ 40B	Ѣ 40F
9.	Ђ 452	‘ 2018	, 2019	“ 201C	” 201D	• 2022	— 2013	— 2014		™ 2122	Љ 459	’ 203A	Њ 45A	Ќ 45C	ђ 45B	ѣ 45F
A.		Ў 40E	ў 45E	Ј 408	□ A4	Г 490	І A6	§ A7	Ё 401	© A9	Є 404	« AB	¬ AC		® AE	Ї 407
B.	° B0	± B1	І 406	і 456	Г 491	μ B5	¶ B6	· B7	ё 451	№ 2116	є 454	» BB	Ј 458	Ѕ 405	ѕ 455	Ї 457
C.	А 410	Б 411	В 412	Г 413	Д 414	Е 415	Ж 416	З 417	И 418	Й 419	К 41A	Л 41B	М 41C	Н 41D	О 41E	П 41F
D.	Р 420	С 421	Т 422	У 423	Ф 424	Х 425	Ц 426	Ч 427	Ш 428	Щ 429	Ъ 42A	Ы 42B	Ь 42C	Э 42D	Ю 42E	Я 42F
E.	а 430	б 431	в 432	г 433	д 434	е 435	ж 436	з 437	и 438	й 439	к 43A	л 43B	м 43C	н 43D	о 43E	п 43F
F.	р 440	с 441	т 442	у 443	ф 444	х 445	ц 446	ч 447	ш 448	щ 449	ъ 44A	ы 44B	ь 44C	э 44D	ю 44E	я 44F

Примечание (Unicode): под символами записаны кодировки Unicode.

KOI8 - R

	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.A	.B	.C	.D	.E	.F
8.	— 2500	 2502	┐ 250C	┐ 2510	└ 2514	└ 2518	└ 251C	└ 2524	└ 252C	└ 2534	└ 253C	■ 2580	■ 2584	■ 2588	■ 258C	■ 2590
9.	▒ 2591	▒ 2592	▒ 2593	┌ 2320	■ 25A0	· 2219	√ 221A	≈ 2248	≤ 2264	≥ 2265	Α0	└ 2321	◦ B0	2 B2	· B7	÷ F7
A.	= 2550	 2551	┐ 2552	ё 451	┐ 2553	┐ 2554	┐ 2555	┐ 2556	┐ 2557	┐ 2558	┐ 2559	┐ 255A	┐ 255B	┐ 255C	┐ 255D	┐ 255E
B.	└ 255F	└ 2560	└ 2561	Ё 401	└ 2562	└ 2563	└ 2564	└ 2565	└ 2566	└ 2567	└ 2568	└ 2569	└ 256A	└ 256B	└ 256C	© A9
C.	Ю 44E	а 430	б 431	ц 446	д 434	е 435	ф 444	г 433	х 445	и 438	й 439	к 43A	л 43B	м 43C	н 43D	о 43E
D.	п 43F	я 44F	р 440	с 441	т 442	у 443	ж 436	в 432	ь 44C	ы 44B	з 437	ш 448	э 44D	щ 449	ч 447	ъ 44A
E.	Ю 42E	А 410	Б 411	Ц 426	Д 414	Е 415	Ф 424	Г 413	Х 425	И 418	Й 419	К 41A	Л 41B	М 41C	Н 41D	О 41E
F.	П 41F	Я 42F	Р 420	С 421	Т 422	У 423	Ж 416	В 412	Ь 42C	Ы 42B	З 417	Ш 428	Э 42D	Щ 429	Ч 427	Ъ 42A

Фрагмент таблицы UNICODE (область ASCII)

00000000000000001111111111111111
0123456789ABCDEF0123456789ABCDEF

0 ! " # \$ % & ' () * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ?

020

0 @ A B C D E F G H I J K L M N O P Q R S T U V V X Y Z [\] ^ _

040

0 ` a b c d e f g h i j k l n n o p q r s t u v w x y z { | } ~ 

060

Фрагмент таблицы UNICODE (область кириллицы)

[illegible]

Фрагмент таблицы UNICODE (окончание)

[illegible]

Представления UNICODE: **UTF-8**, UTF-16, UTF-32 (от 2-х до 6 байт)