

# Информатика

## Введение в вычислительную технику и программирование

Гирик Алексей Валерьевич

Университет ИТМО

2022

# Кто во всем виноват

- Лекции

Гирик Алексей Валерьевич

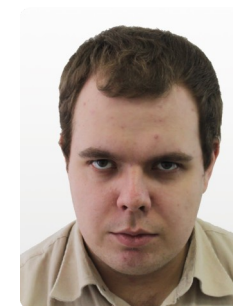
[avg@itmo.ru](mailto:avg@itmo.ru)



- Лабораторные занятия

Грозов Владимир Андреевич

[va\\_groz@mail.ru](mailto:va_groz@mail.ru)



Горлина Анастасия Витальевна

[gorlina.a.v@mail.ru](mailto:gorlina.a.v@mail.ru)

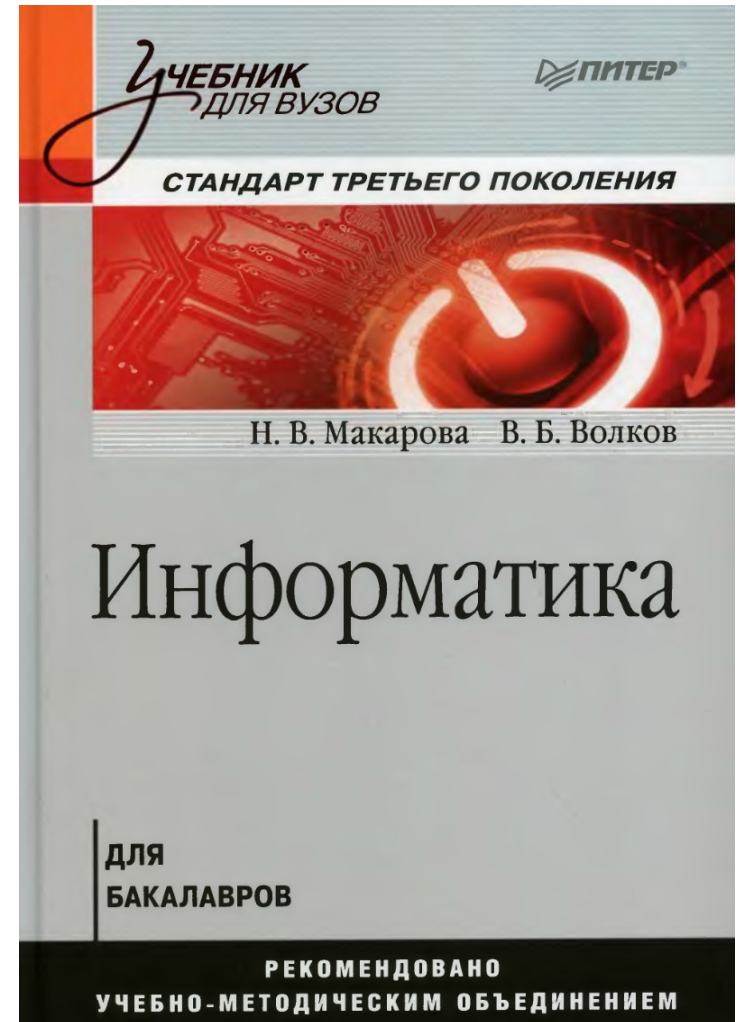


# Что делать

*Макарова Н.В., Волков В.Б.*  
Информатика. Учебник для  
вузов

Главы:

- 1, 2
- 4
- 10 – 12
- 14, 15
- 19



# Цель и задачи курса

## ■ Цель курса

- сформировать у слушателей системные представления об основах вычислительной техники, алгоритмах и языках программирования

## ■ Задачи курса

- сформировать базовые знания об аппаратных средствах вычислительной техники и принципах их работы
- привить теоретические и практические знания и навыки использования методов и средств разработки программ на языках программирования

# Структура курса

- Системы счисления и двоичная арифметика
- Аппаратные средства вычислительной техники
- Ассемблеры
- Основы алгоритмизации
- Введение в алгебру логики
- Основы программирования на языке высокого уровня
- Общие сведения об инструментальных средствах трансляции и отладки программ

# Препятствия

- Лабораторные работы

- 2 шт.

- обязательны к сдаче!

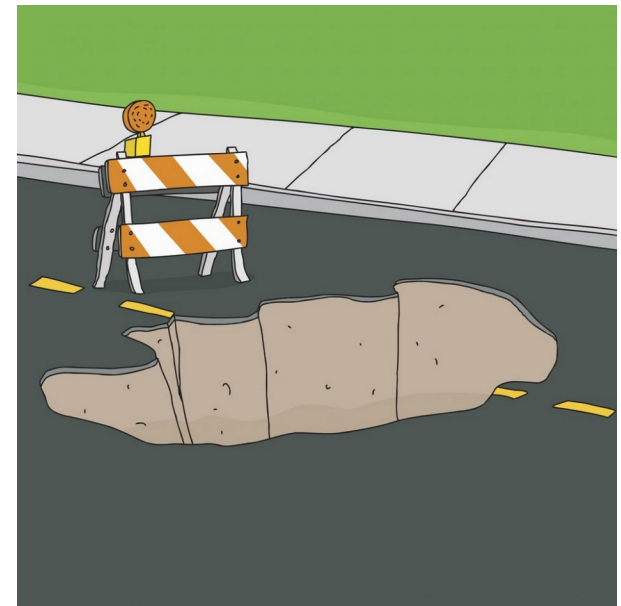
- Тесты

- 7 шт.

- необязательны ;-)

- Экзамен

- ~ 0 шт.



# Препятствия

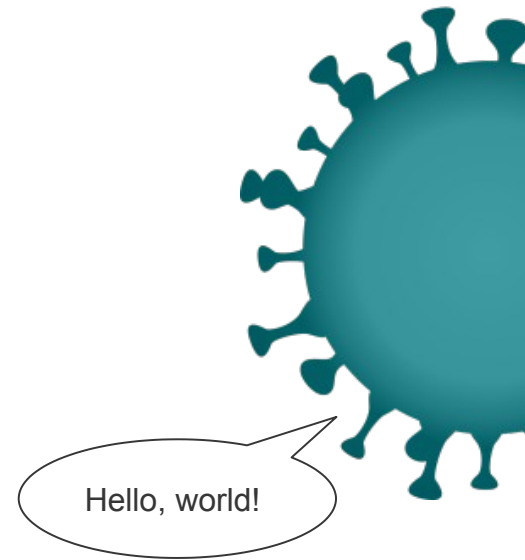
- 2 лабораторных работы
  - обязательны к сдаче
- 4 теста
  - по 5 или 6 вопросов
- экзамен
  - оценка определяется тестами и лабораторными

	ㄟ(ツ)ㄟ	3	4	5
Л/р № 1	-	+	+	+
Л/р № 2	-	+	+	+
Правильных ответов в тестах*	< 8	>= 8	>= 19	>= 31

\* – но это не точно

# План занятий

- **лекции**
  - 8 шт.
    - ауд. 2202/1222
  - теоретический материал
- **лабораторные занятия**
  - 8 шт.
    - ауд. ?? (см. расписание)
  - тесты по теоретическому материалу
    - по лекциям, литературе и прочим источникам
  - выполнение и защита лабораторных работ

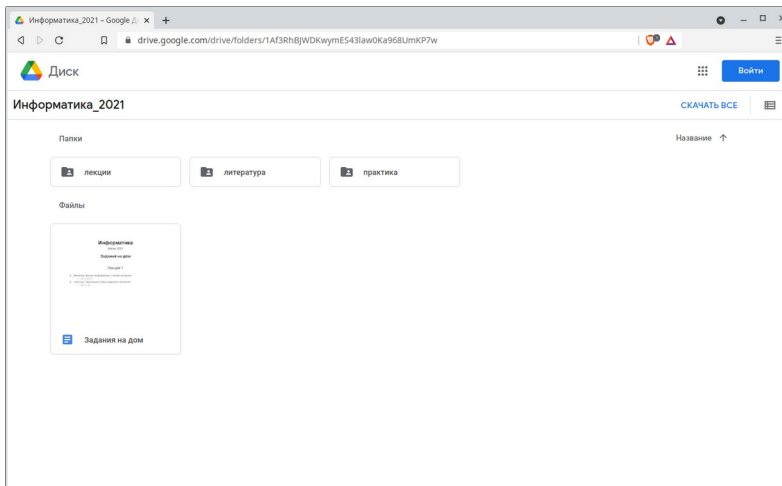




# Материалы курса

- Презентации, материалы к лекциям, литература, задания на лабораторные работы

[shorturl.at/jqRZ6](https://shorturl.at/jqRZ6)



# Computer Science

- Теоретические дисциплины:

- теория информации
- теория алгоритмов и автоматов
- теория вычислимости
- теория компиляторов
- дискретная математика
- математическая логика
- теория графов
- теория множеств
- и т.д. см

[http://en.wikipedia.org/wiki/Computer\\_science](http://en.wikipedia.org/wiki/Computer_science)

# Системы счисления

# Числа и цифры

- Число — понятие универсальное, однако системы записи чисел могут отличаться
- Цифра — символ, обозначающий число в рамках определенной системы счисления

16

١٦

0x10 10h

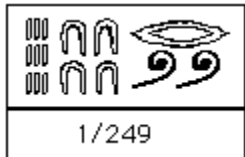
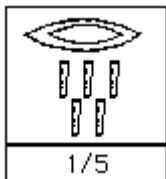
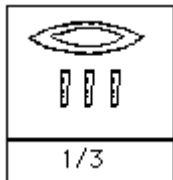
10000

XVI

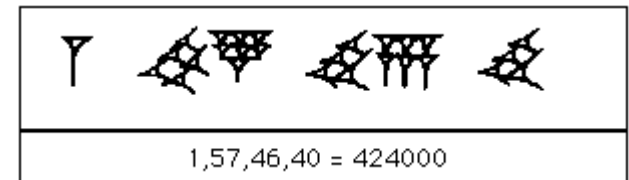
# Системы счисления

## Классификация:

- по основанию
- по позиционности
  - позиционные/непозиционные/смешанные



0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	19

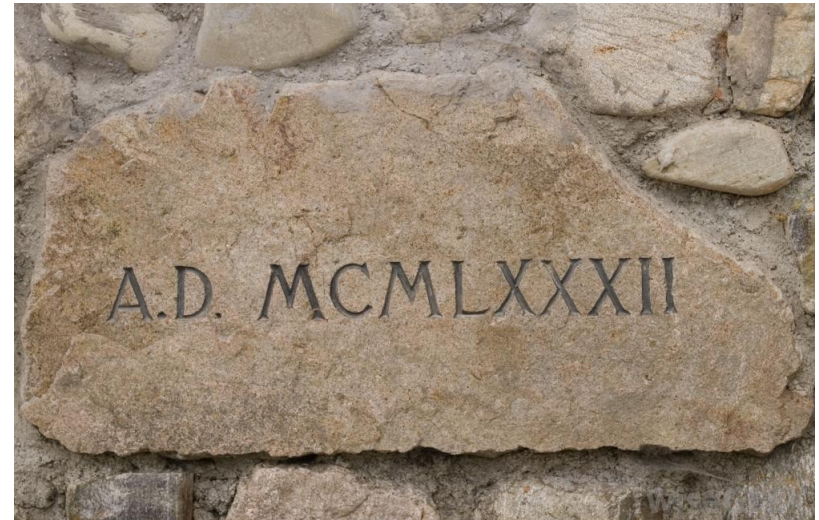


# Непозиционные и смешанные системы счисления

- Египетская система счисления



- Римская система счисления



Мы Даем Советы Лишь Хорошо Воспитанным Индивидуумам

# Позиционные системы счисления

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

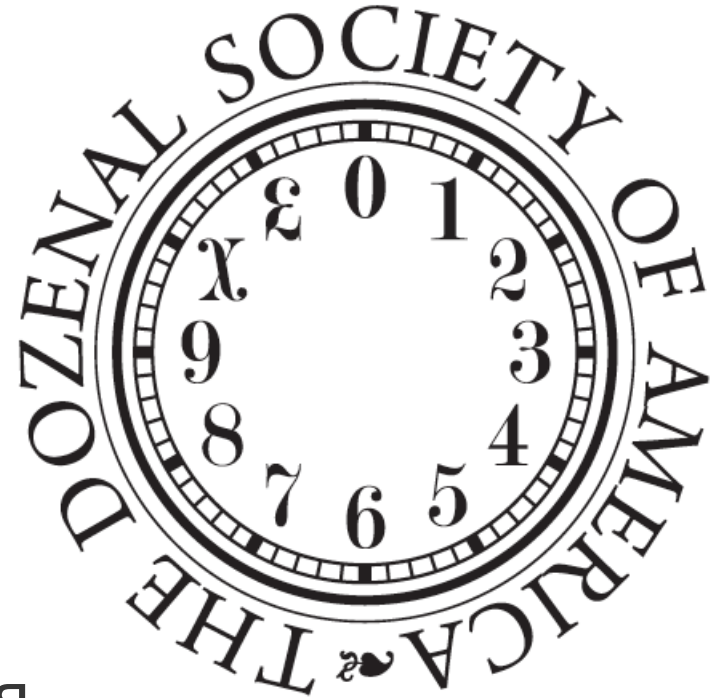
старший  
разряд

младший  
разряд

$$1234 = 1 \cdot 10^3 + 2 \cdot 10^2 + 3 \cdot 10^1 + 4 \cdot 10^0$$

# Системы счисления

- человек
  - десятичная
  - пятеричная, двенадцатеричная, двадцатеричная, шестидесятеричная
- ЭВМ
  - двоичная, шестнадцатеричная
  - троичная, десятичная

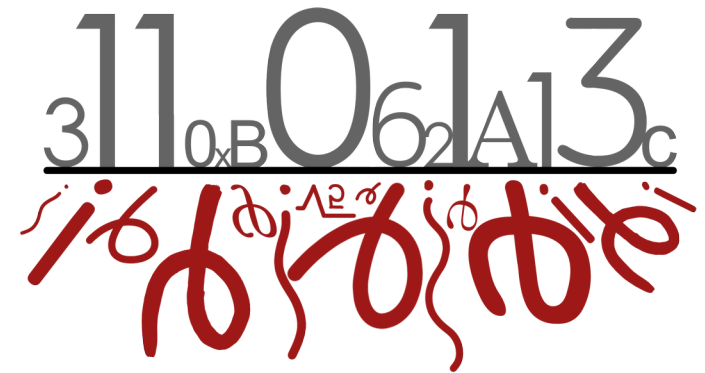


<http://dozenal.org>



# Очень странные системы счисления

- фибоначчиева
- факториальная
- биномиальная
- система остаточных классов
- Штерна-Броко
- ...

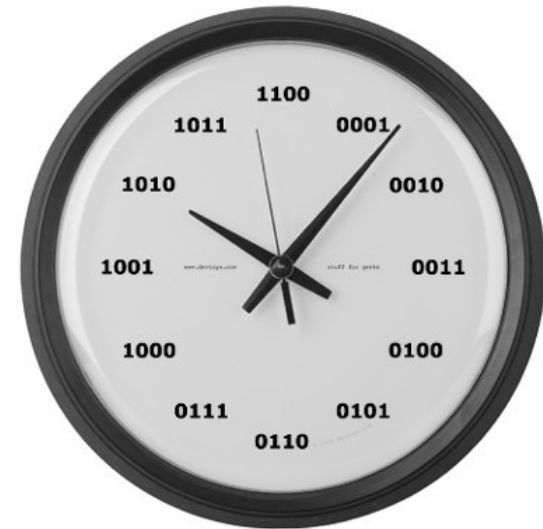


# Повторение

- системы счисления
  - двоичная
  - шестнадцатеричная
- перевод чисел из одной системы счисления в другую
- количество информации
- единицы измерения количества информации

# Двоичная система счисления

1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010
...	...



# Степени двойки

$$2^0 = 1$$

$$2^1 = 2$$

$$2^2 = 4$$

$$2^3 = 8$$

$$2^4 = 16$$

$$2^5 = 32$$

$$2^6 = 64$$

$$2^7 = 128$$

$$2^8 = 256$$

$$2^9 = 512$$

$$2^{10} = 1024$$

$$2^{11} = 2048$$

$$2^{12} = 4096$$

$$2^{13} = 8192$$

$$2^{14} = 16384$$

$$2^{15} = 32768$$

$$2^{16} = 65536$$

$$2^{17} = 131072$$

...

$$2^{20} = 1048576$$

		32	2
	2	256	2
2	4	128	4
8	2	4	16

# Перевод чисел из десятичной системы в двоичную

$$(137)_{10} = ( \quad ? \quad )_2$$

- медленный – деление в столбик
- быстрый – подбор степеней двойки
- ленивый – например, с помощью python:

```
>>> f'{137:b}'
```

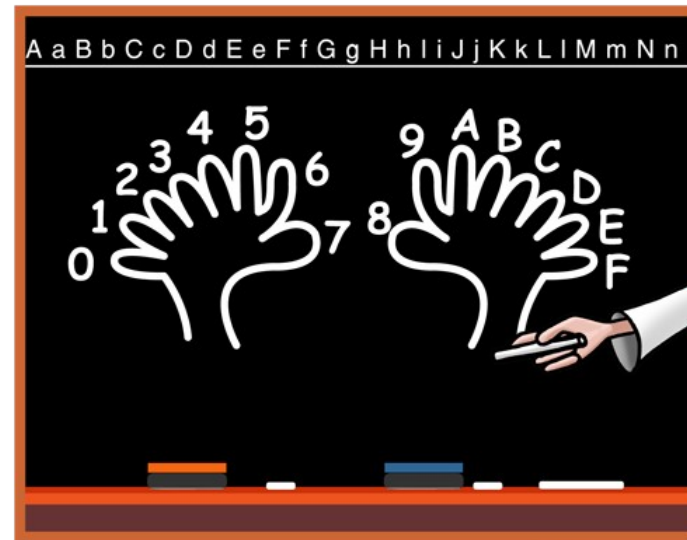
```
10001001
```

```
>>> int(f'{137:b}', 2)
```

```
137
```

# Шестнадцатеричная система счисления

- 011101010111010001001100110110101010001110
- 1D5D1336A8E
- 2017849469582



$$(DEADBEEF)_{16} = (3735928559)_{10}$$

$$(CAFEBABE)_{16} = (3405691582)_{10}$$

<http://en.wikipedia.org/wiki/Hexspeak>

# Соответствие двоичных, восьмеричных и шестнадцатеричных чисел

<b>dec</b>	<b>bin</b>	<b>oct</b>	<b>hex</b>
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

# Перевод чисел из шестнадцатеричной системы в двоичную и обратно

011101010111010001001100110110101010001110  
└┘└┘└┘└┘└┘└┘└┘└┘└┘└┘└┘  
1 D 5 D 1 3 3 6 A 8 E

1D5D1336A8E



# Очень краткие сведения о двоичной арифметике

# Отрицательные числа в двоичной системе счисления


- при записи можно просто поставить знак "минус":

$-11$  ( $-3$  в десятичной системе)

- при хранении числа в памяти компьютера возможности поставить "минус" нет, приходится хранить знак в виде дополнительного бита:

00000011 ( $3$  в прямом коде)

10000011 ( $-3$  в ??? коде)



# Обратный код

- обратный (*ones' complement*) код получается путем инвертирования битов числа:

00000011      ( 3 в прямом коде)

11111100      ( −3 в обратном коде)

00000000      ( 0 в прямом коде)

11111111      ( −0 в обратном коде)

# Дополнительный код

- дополнительный (*two's complement*) код получается путем инвертирования и сложения с единицей:

00000011 ( 3 в прямом коде)

11111100 ( −3 в обратном коде)

11111101 ( −3 в дополнительном коде)

- дополнительный код позволяет избежать наличия числа "минус 0"

# Вопрос на засыпку

Дано двоичное 8-разрядное число:

10000000

Это число со знаком или без знака?



# Хранение двоичных чисел в разрядной сетке

- числа без знака (прямой код)

00000000	0	0
00000001	1	
00000010	2	
...	...	
11111110	254	
11111111	255	$2^n - 1 \quad (n = 8)$

# Хранение двоичных чисел в разрядной сетке

- числа со знаком (обратный код)

10000000	-127	$-2^{n-1} + 1$ (n=8)
10000001	-126	
...		
11111110	-1	
11111111	-0	
00000000	0	
00000001	1	
...	...	
01111110	126	
01111111	127	$2^{n-1} - 1$ (n=8)

# Хранение двоичных чисел в разрядной сетке

- числа со знаком (дополнительный код)

10000000	-128	$-2^{n-1}$	(n=8)
10000001	-127		
...			
11111110	-2		
11111111	-1		
00000000	0		
00000001	1		
...	...		
01111110	126		
01111111	127	$2^{n-1} - 1$	(n=8)



# Двоичная арифметика

- сложение выполняется так же, как с десятичными числами (например, в столбик):

$$\begin{array}{r} + \quad 00000011 \\ \quad 00000001 \\ \hline 00000100 \end{array}$$

$$\begin{array}{r} + \quad 3 \\ \quad 1 \\ \hline 4 \end{array}$$

# Двоичная арифметика

- вычитание можно выполнять как сложение с числом в дополнительном коде:

$$\begin{array}{r} + \quad 00000011 \\ \quad 11111111 \\ \hline 00000010 \end{array}$$

$$\begin{array}{r} - \quad 3 \\ \quad 1 \\ \hline 2 \end{array}$$

# Ограничения разрядной сетки

$$\begin{array}{r} + 100000000 \\ 100000000 \\ \hline 1000000000 \end{array}$$

перенос  
за пределы  
разрядной сетки

беззнаковая  
арифметика

$$\begin{array}{r} + 128 \\ 128 \\ \hline 256 \end{array}$$

знаковая  
арифметика

$$\begin{array}{r} - 128 \\ + 128 \\ \hline 0 \end{array}$$

# Ограничения разрядной сетки

$$\begin{array}{r} + 01111111 \\ 00000001 \\ \hline 10000000 \end{array}$$

перенос в  
знаковый разряд

беззнаковая  
арифметика

$$\begin{array}{r} + 127 \\ 1 \\ \hline 128 \end{array}$$

знаковая  
арифметика

$$\begin{array}{r} + 127 \\ 1 \\ \hline -128 \end{array}$$

# Результаты сложения

с точки зрения **знаковой** арифметики результаты...

		... верные	... неверные
с точки зрения <b>беззнаковой</b> арифметики результаты...	... верные	переносов нет	переноса за разрядную сетку нет, перенос в знаковый разряд есть
	... неверные	перенос за разрядную сетку есть, перенос в знаковый разряд есть	перенос за разрядную сетку есть, переноса в знаковый разряд нет

# Перенос и переполнение

- если получен **неправильный** результат с точки зрения
  - **беззнаковой** арифметики – имеет место **перенос** (carry)
    - процессор устанавливает в единицу флаг переноса (carry flag)
  - **знаковой** арифметики – имеет место **переполнение** (overflow)
    - процессор устанавливает в единицу флаг переполнения (overflow flag)

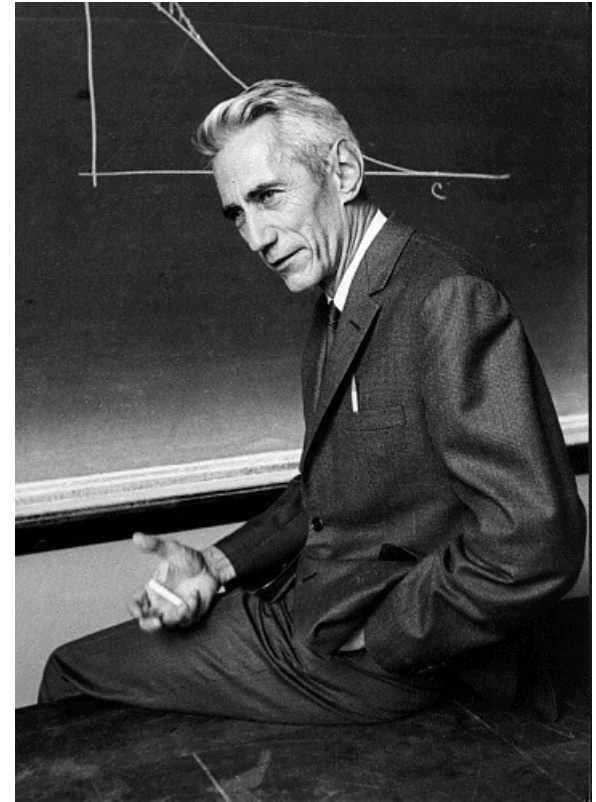


# Единицы измерения информации

# Количество информации

- Информация – любые сведения о чем угодно
- Данные – представление информации в формализованном виде, пригодном для передачи и обработки
- Количество информации как мера уменьшения неопределенности знания

$$I = -\sum_{i=1}^N p_i \cdot \log_2 p_i$$



Клод Элвуд Шеннон



# Единицы измерения информации

- **бит** (bit)
- **байт** (byte, octet)
  - тетрада (nibble)
- **слово** (word)
  - старший и младший байты
- **двойное слово** (double word)
  - старшее и младшее слова
- **учетверенное слово** (quad word)
  - старшее и младшее двойные слова
- **машинное слово**



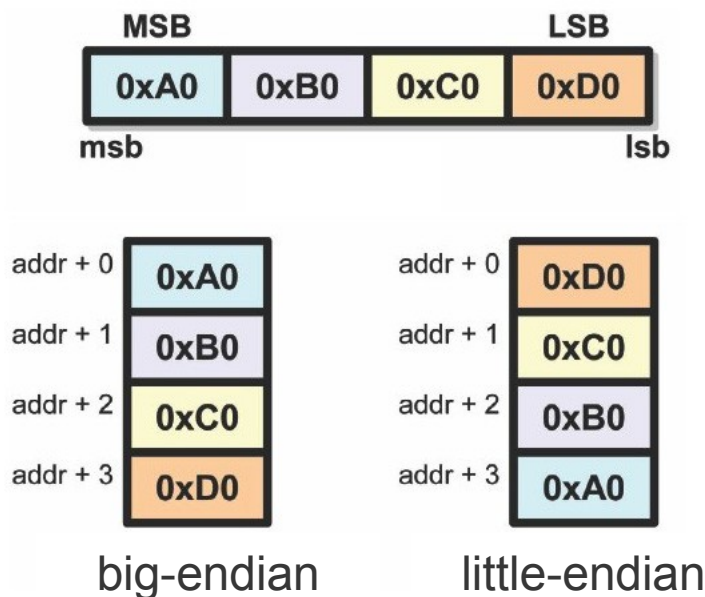
Hobbit



Hobbyte

# Порядок хранения байтов в машинных словах

- **big-endian** (network order)
  - младший байт по большому адресу
- **little-endian** (host order)
  - младший байт по меньшему адресу



# Килобайт, например

## ■ 1024 байта – килобайт

YpqI^hVdwbuBIKrxaaahZgGSpVWgsJImDVs^OY`ddBqqRDsJAx]jQZ]Auvlq`dVNM00IWnfWhOSccvHd  
AibxwWmWWgRpkrIRvKzwngWWHHe|N{|wm^ARtcxAXDzFbPwYZPQV]]h[\_^VKpUXMc]YTyVLE[NmI[FQ  
UiPARZyhZeVMLFuCzOFFXU{sDmvBO[MHRXuXKZGPSq\G`qWhBIbDZpsJJuyuHLBHJD^y[Po^CD[\|C  
EsaaRX\LCMKfHBUJldPqSS]HjlGTvyoFLuIrf|IPHiffll]lMUFYLPFWG\_tW]cRSKSzrvdYkhDEFrMlh  
kNr{w}IcWSa^PfZYjo\_gLFX{zh[UOyBQE^ePVNWqjcOa`bfz^DVK`DJtyL`VvmdSygHb\_iLLj\z\_\_]  
aj{cJrp\_^wOFMt{FRxIlAZtkjqe|PhiIovNH`EoaibpLLWohTfRUj[IrMqyKZu]||TL\_]hJgKGw`cYe  
HIHWAVyEKm[sbdhsc^dDFOutGUqgic{TPpnAk^pezIe\mugRnJMC`mwzAlhZ[yDLObSf\_FqKBD0I^dD  
MZsqMrsqNixnpGtmLtg]ZGgbm\\pBHwglJqKFfqEc^VmdsSyPyLncuWocbK|iJhUAjFqGx|QhMVh[uo  
Wzk^rJh[KMpPNzG[gBID^mExDhhLMqAeGOQ^P\xKGiKOLuln\_jDeM\Yl]{zXVKl^S\KY{xxYSLfJboQ  
eAgZoL\wfbGCzK{ieiaHelloWorldCqu`KweGjff\_TtmPMuXEcDCeq{M}xDz^IhPljhwJR]cv[HcRCE  
pKuqlGbA{`VuPtSqStt\rKvgPJbj^uZdXzvgjW{vDUxIUneh`KyVky`bojLgkDBHD{`GIbY\RsDDlkR  
BNPHR]dYfXpNBkl^IsaUpTYr^aU|rtBkQnIRfgEV\Sml\_nAXNRGdeujyjeliI^DKVmDfVyMUmFnmNrRH  
jSOTrnSaDtiqVrZ\_tQrdXfrWTuAXznhpUDdAeezBCDwW\weCPIxTAlfAcXgso{uuSF^gyO|jEFe

# Единицы измерения информации

- 1024 килобайтов – 1 **мегабайт** ( $2^{20}$  байтов)
  - фотография среднего качества
- 1024 мегабайтов – 1 **гигабайт** ( $2^{30}$  байтов)
  - 5-10 минут видео высокого разрешения
- 1024 гигабайтов – 1 **терабайт** ( $2^{40}$  байтов)
  - HDD объемом 8 терабайтов поступили в продажу в 2015 году
- 1024 терабайтов – 1 **петабайт** ( $2^{50}$  байтов)
  - примерно 13 лет видео высокого разрешения
  - ~ 10 петабайтов – объем данных на почтовых серверах Yandex

# Единицы измерения информации

- 1024 петабайтов – 1 **эксабайт** ( $2^{60}$  байтов)
  - ~ 50 эксабайтов среднемесячный объем глобального IP трафика
- 1024 эксабайтов – 1 **зеттабайт** ( $2^{70}$  байтов)
  - в 2011 году ICANN насчитала в мире порядка 3 зеттабайтов информации
- 1024 зеттабайтов – 1 **йоттабайт** ( $2^{80}$  байтов)
  - ?

# Один ма-а-аленький нюанс

- 1 километр = 1000 метров ( $10^3$  метров)
- 1 килобайт = 1024 байтов ( $2^{10}$  байтов)

Если считать, что 1 килобайт = 1000 байтов, тогда погрешность составляет

$$e = \frac{1024 - 1000}{1024} = 0.0234375 (\approx 2.34\%)$$

1 терабайт =  $2^{40}$  байтов. Или  $10^{12}$  байтов? Погрешность равна

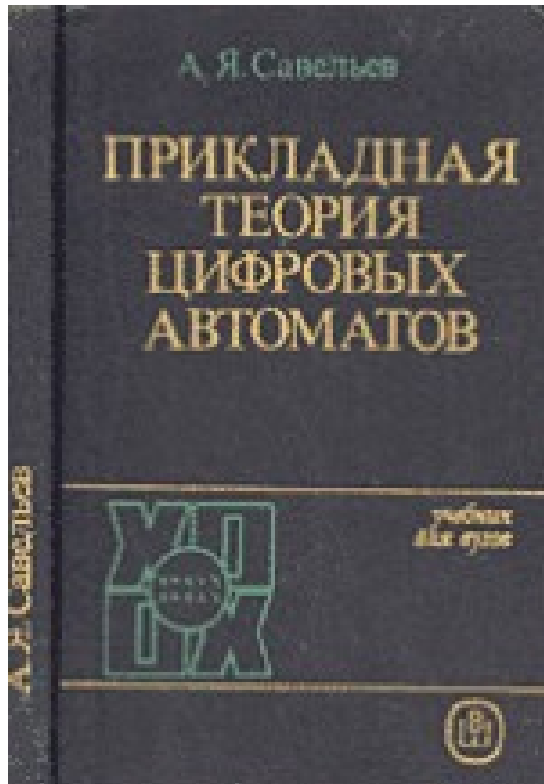
$$e = \frac{1024 * 1024 * 1024 * 1024 - 1000000000000}{1024 * 1024 * 1024 * 1024} = 0.090505298 (\approx 9.05\%)$$

# Обозначения единиц измерения информации

Приставка СИ	ГОСТ 8.417-2002 (для байтов)	Приставка МЭК	Сокращение МЭК	Степень двойки	Степень десятки (СИ)	Относительная ошибка, %
<b>кило</b>	Кбайт	<b>киби</b>	Кибит, КиБ	10	3	2.34
<b>мега</b>	Мбайт	<b>меби</b>	Мибит, МиБ	20	6	4.63
<b>гига</b>	Гбайт	<b>гиби</b>	Гибит, ГиБ	30	9	6.87
<b>тера</b>	Тбайт	<b>теби</b>	Тибит, ТиБ	40	12	9.05
<b>пета</b>	Пбайт	<b>пеби</b>	Пибит, ПиБ	50	15	11.18
<b>экса</b>	Эбайт	<b>эксби</b>	Эибит, ЭиБ	60	18	13.26
<b>зетта</b>	Збайт	<b>зеби</b>	Зибит, ЗиБ	70	21	15.29
<b>йотта</b>	Йбайт	<b>йоби</b>	Йибит, ЙиБ	80	24	17.28

- и в конце-концов, как правильно – «пара йобибайт смешных картинок» или «пара йобибайтов»?

# Учебник по информатике



- *А. Я. Савельев*  
Прикладная теория цифровых автоматов
  - общие сведения о системах счисления и представлении информации
  - общие сведения об аппаратных средствах вычислительной техники
  - операции на двоичных сумматорах
  - логические основы ЦА
  - введение в теорию автоматов

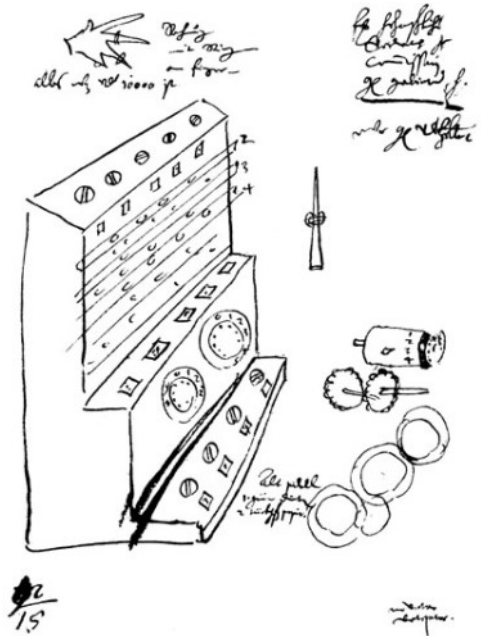


# Вычислительная техника

# Краткая история вычислительной техники

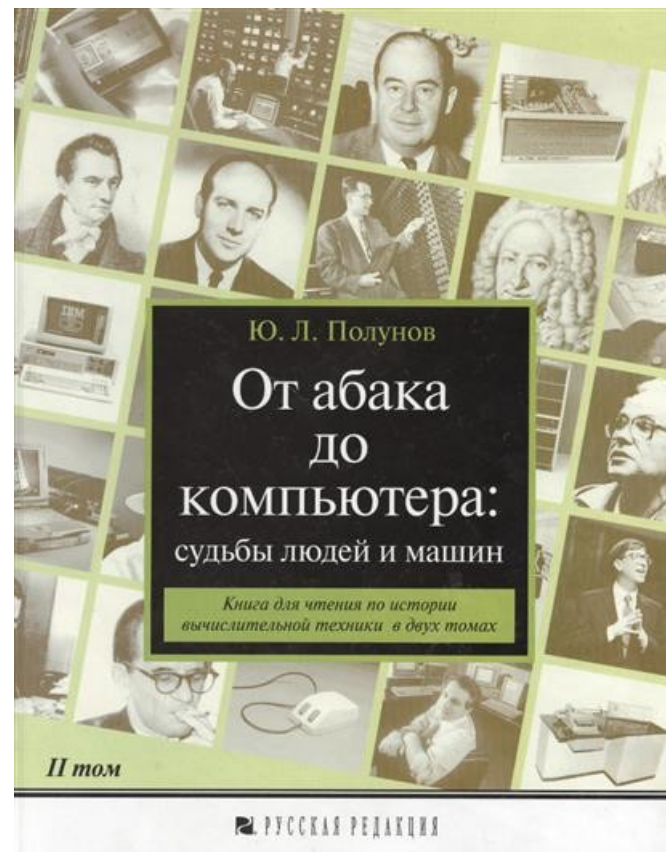
- ...
- около V в. до н.э. – абак
- ...
- 1623 – «часы для счета» Шиккарда
- 1645 – арифмометр Паскаля
- 1670 – счетная машина Лейбница
- ...
- 1801 – ткацкий станок Жаккара
- 1823 – «разностная машина» Бэббиджа
- 1833 – «аналитическая машина» Бэббиджа
- 1885 – счетная машина Берроуза
- 1888 – табулятор Холлерита
- ...
- 1938 – компьютер V-1 Цузе
- ...
- история создания вычислительных машин

□ <http://edu.mccme.ru/School/INet/sch1685/history.htm>

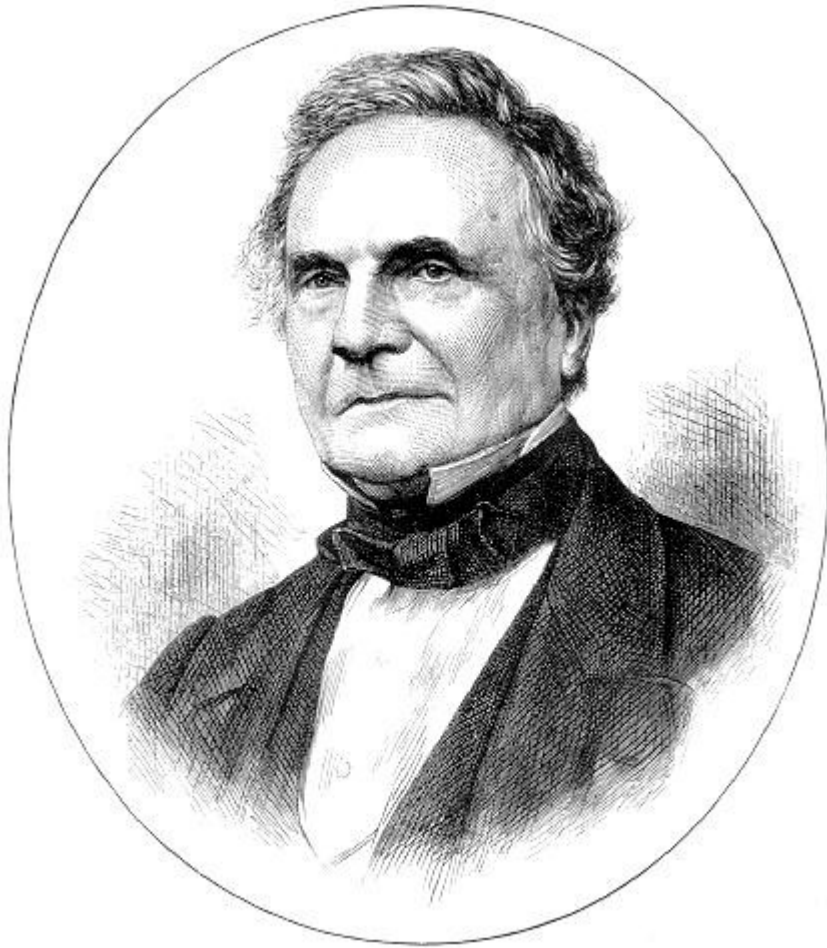


# Подробная история вычислительной техники

- *Ю.Л. Полунов*  
От абака до  
компьютера: судьбы  
людей и машин

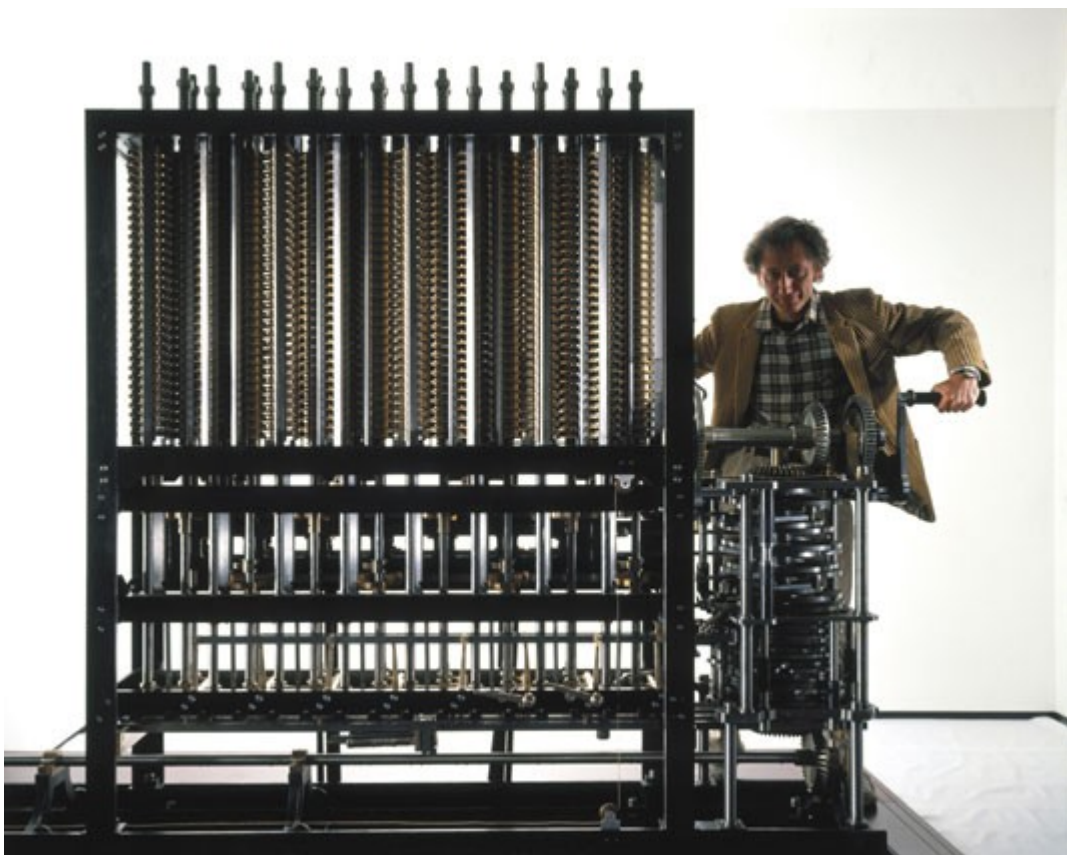


# Аналитическая машина Бэббиджа



- Чарльз Бэббидж (Charles Babbage)
  - differential engine (1823)
  - analytical engine (1833)
  - потратил на исследования 17 000 фунтов
  - оба проекта «провалились»

# Разностная машина Бэббиджа



- 1991, Британский музей науки
- рабочий прототип разностной машины Бэббиджа (Difference Engine No. 2)

# Первый в мире программист

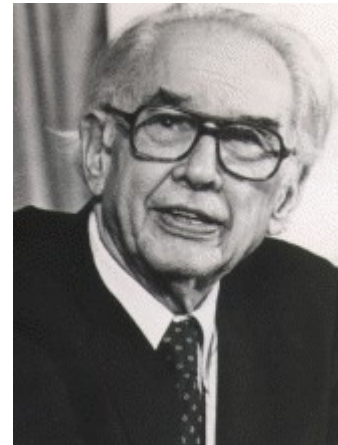
- Ада Августа Лавлейс  
(Ada Augusta Lovelace)
  - Sketch of the analytical engine  
invented by Charles Babbage  
(L. Menabrea), 1842
  - <http://www.fourmilab.ch/babbage/sketch.html>
  - использовала термины:
    - рабочая ячейка
    - цикл
    - индекс
    - подпрограмма
  - её именем назван язык  
программирования Ada



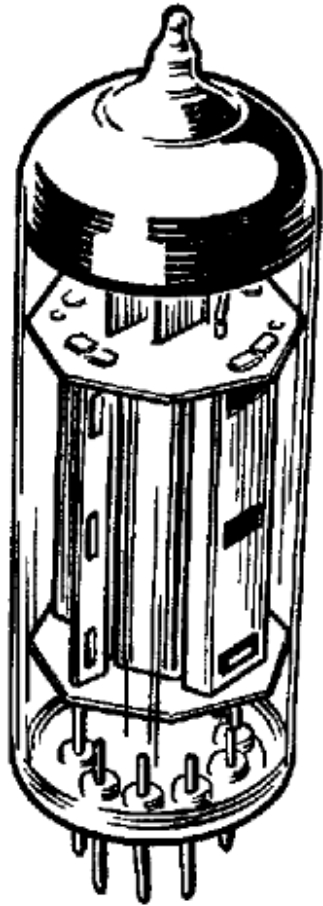


# Первый компьютер

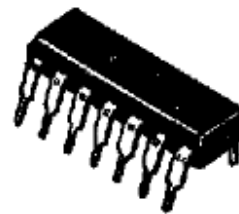
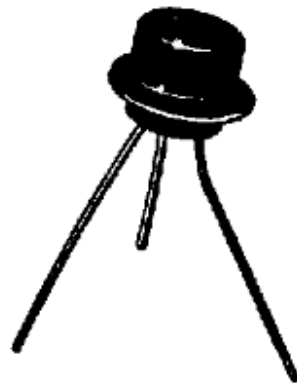
- Конрад Цузе, Германия
  - Z1, 1938
  - Z3, 1939 – 1941
- Джон Атанасов, США
  - ABC, 1939 – 1942
- Томас Флауэрс, Англия
  - Colossus, 1943 – 1944
- Говард Айкен, США
  - Mark I, 1944
- Джон Моучли, Джон Эккерт, США
  - ENIAC, 1945



# Поколения компьютеров



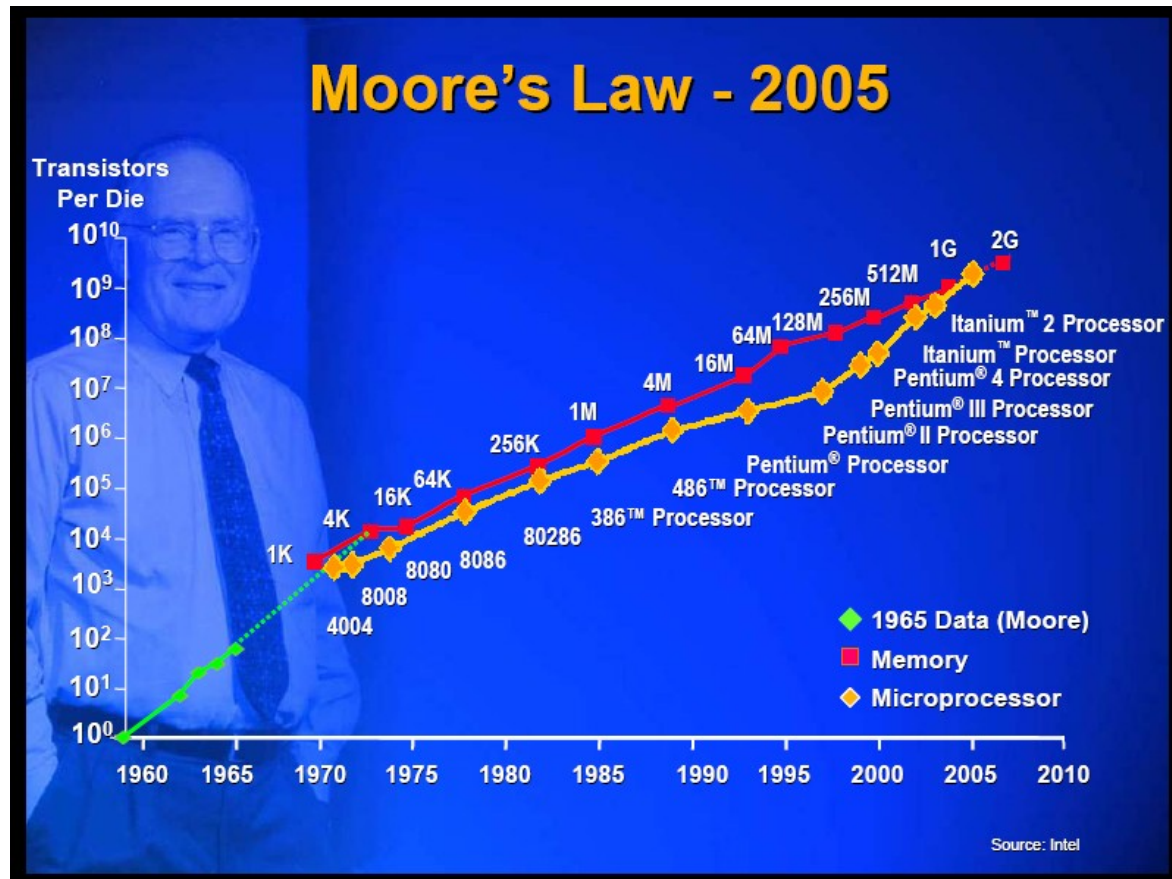
- 1940-е – 1950-е – 1-е поколение
- 1960-е – 2-е поколение
- 1970-е – 3-е поколение
- 1980-е – 4-е поколение
- 1980-е - ... – 5-е поколение





# "Закон" Мура

- Гордон Мур, 1965
  - количество транзисторов в интегральных схемах увеличивается в 2 раза примерно каждые 18 месяцев



# Две формы представления информации

- аналоговая (непрерывная)
  - в виде одного сигнала, сравнимого с заданной величиной:

$$x = 1024 \quad \Leftrightarrow \quad V_x = 1,024 \text{ В}$$

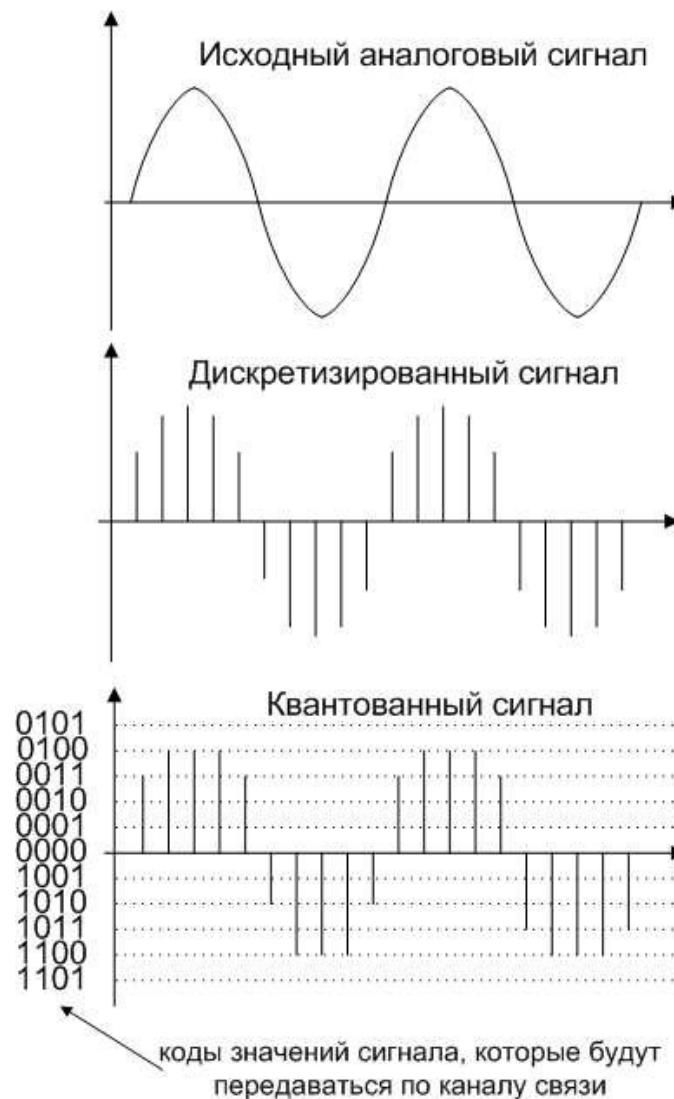
- бесконечное множество значений
- дискретная (цифровая)
  - в виде нескольких сигналов, представляющих отдельные разряды числа:

$$x = 1024 \quad \Leftrightarrow \quad \begin{array}{l} V_4 = 1 \text{ В} \\ V_3 = 0 \text{ В} \\ V_2 = 2 \text{ В} \\ V_1 = 4 \text{ В} \end{array} \quad (V = 0 - 9 \text{ В})$$

- конечное множество значений

# ЦВМ и АВМ

- в ЦВМ информация представлена в виде целых чисел
  - ЦАП
  - АЦП
- в АВМ информация представлена в виде значений аналоговых физических величин (давления газа или жидкости, напряжения электрического тока)



# Принципы фон Неймана

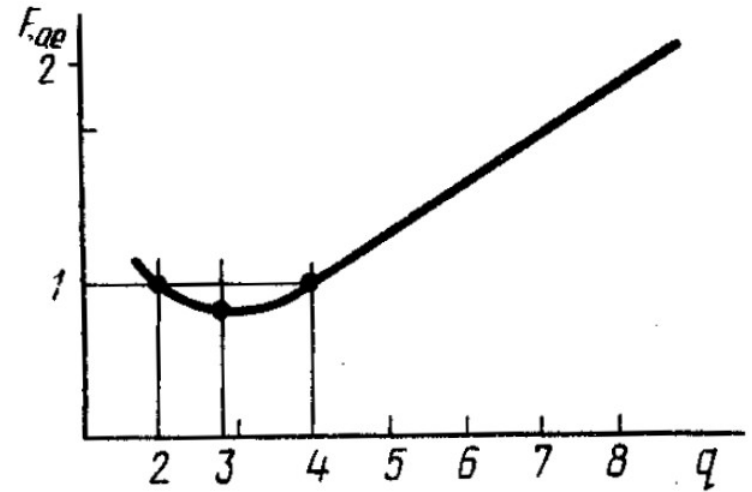
# Архитектура фон Неймана



- Джон фон Нейман (John von Neumann)
  - в отчете «First Draft of a Report on the EDVAC» описал архитектуру ЭВМ, позже названную его именем

# Принцип двоичной системы счисления

- с точки зрения затрат оборудования наиболее экономичной является система счисления с основанием  $e$
- схемотехнически бистабильные элементы дешевле тристабильных



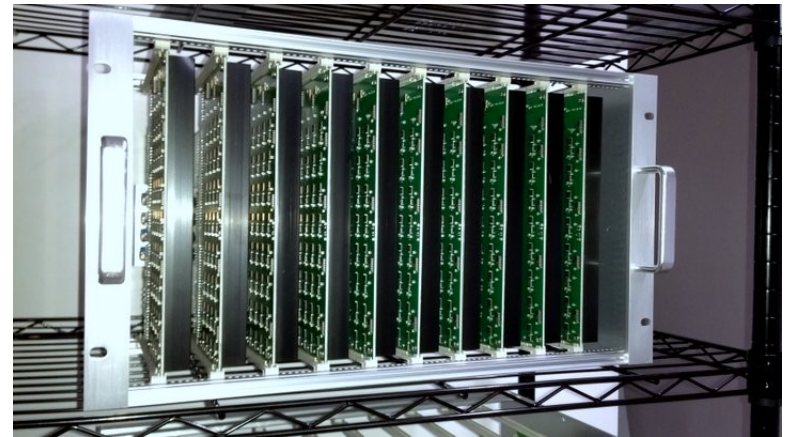
ЭВМ «Сетунь», МГУ, 1959

# Принцип программного управления

- универсальность достигается за счет разделения оборудования и программы, то есть прикладные алгоритмы реализуются не в «железе», а с помощью «софта»
- АЛУ(ЦПУ) выполняет операции, которые определяются набором команд (instruction set)



табулятор Холлерита, 1890

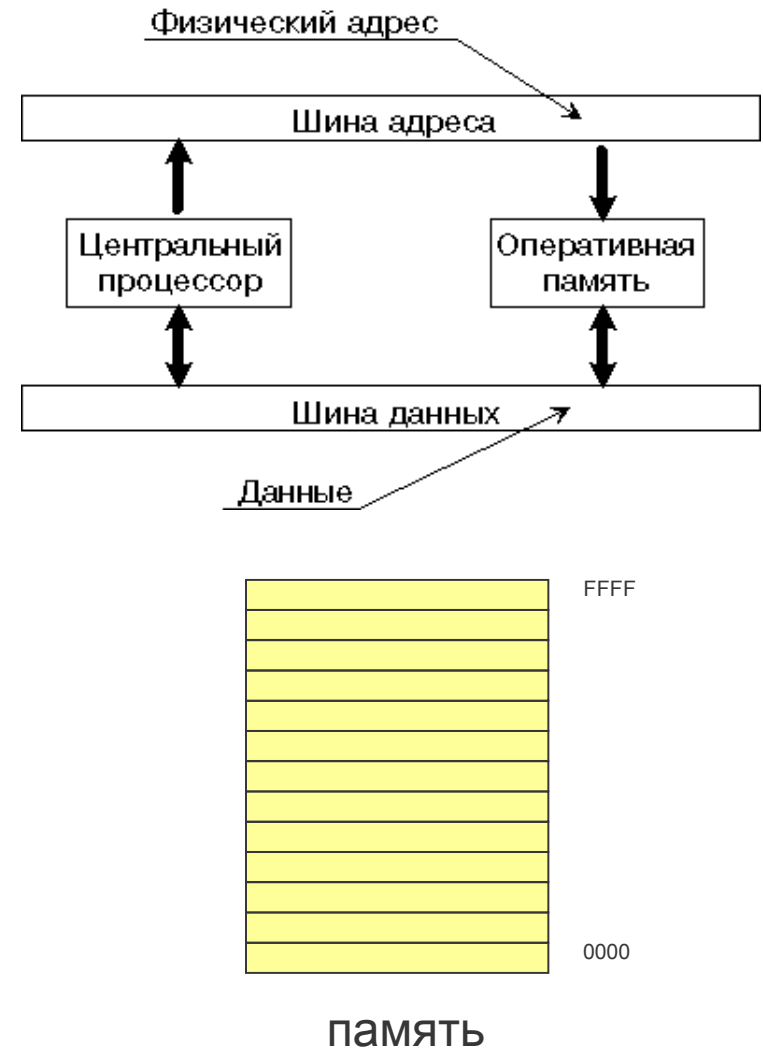


ASICMINER bitcoin miner



# Принцип хранимой программы

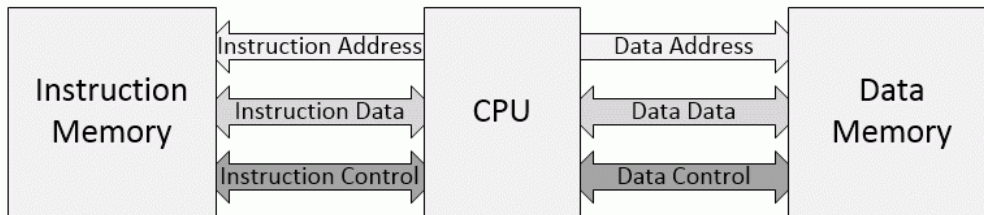
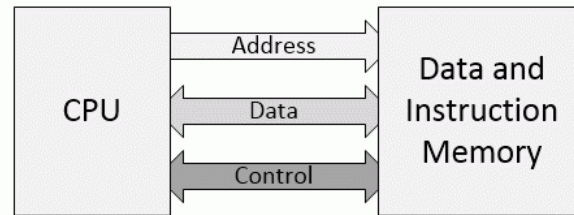
- в начале работы программа загружается в оперативную память
- оперативная память – массив двоичных разрядов
- команды программы и данные хранятся в одном и том же адресном пространстве
  - как отличить команды от данных?





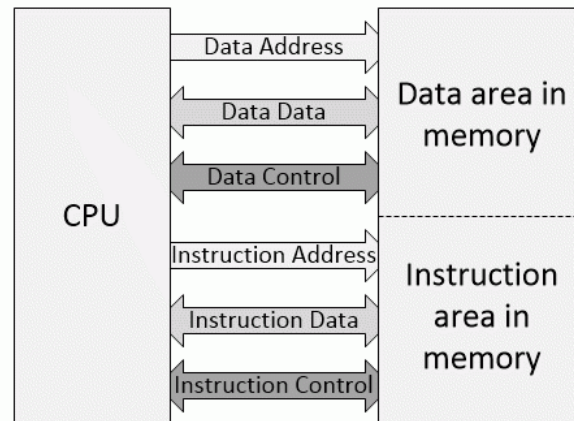
# Принстонская и гарвардская архитектуры процессоров

(a) von Neumann (Princeton) architecture (instructions and data share memory devices and busses)



(b) Harvard architecture (physically separate memory devices and busses)

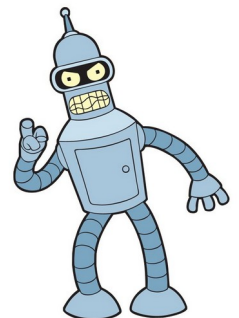
(c) Modified Harvard architecture (physically shared memory, separate busses for instructions and data)



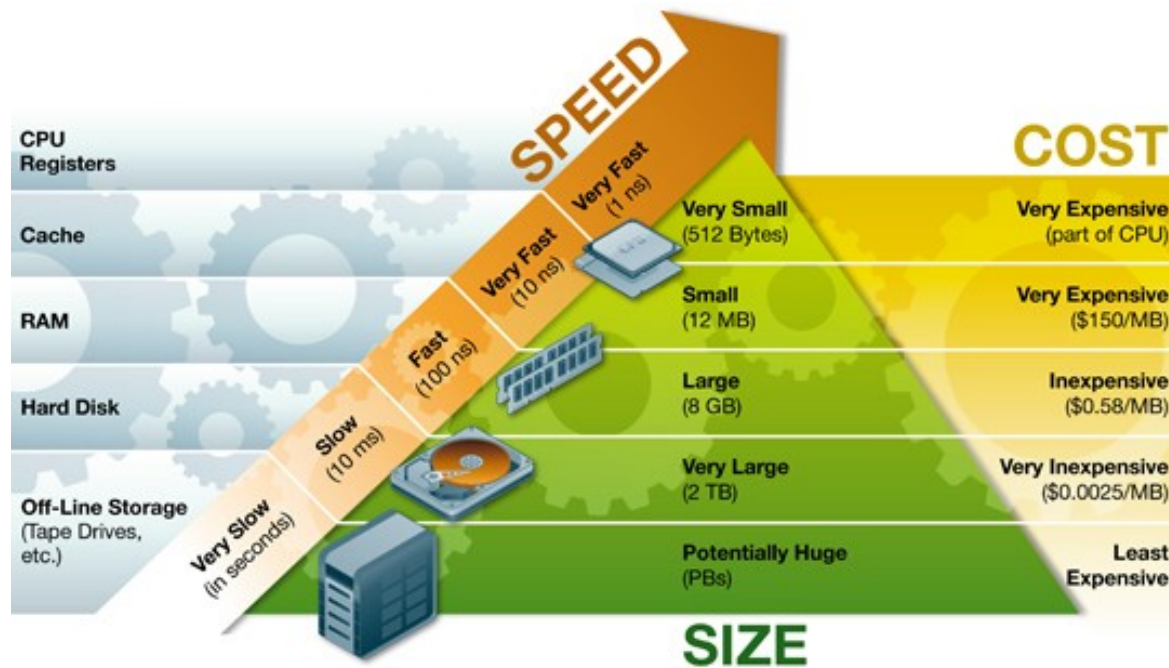
# Принцип условного перехода

- команды выбираются процессором из памяти последовательно
- в системе команд присутствуют специальные команды, позволяющие изменить порядок выборки

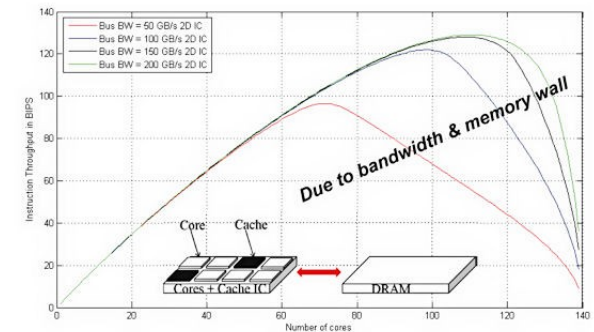
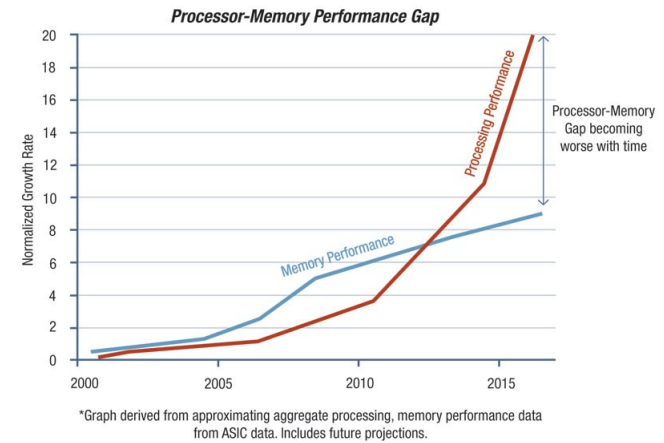
байт 000:	загрузить число a в регистр 1
байт 001:	загрузить число b в регистр 2
байт 002:	вычесть из регистра 1 регистр 2
байт 003:	если в регистре 1 отрицательное число, перейти к адресу <b>140</b>
байт 004:	пока все хорошо
байт 005:	перейти к байту 000
...	
байт 140:	слава роботам! убить всех людей



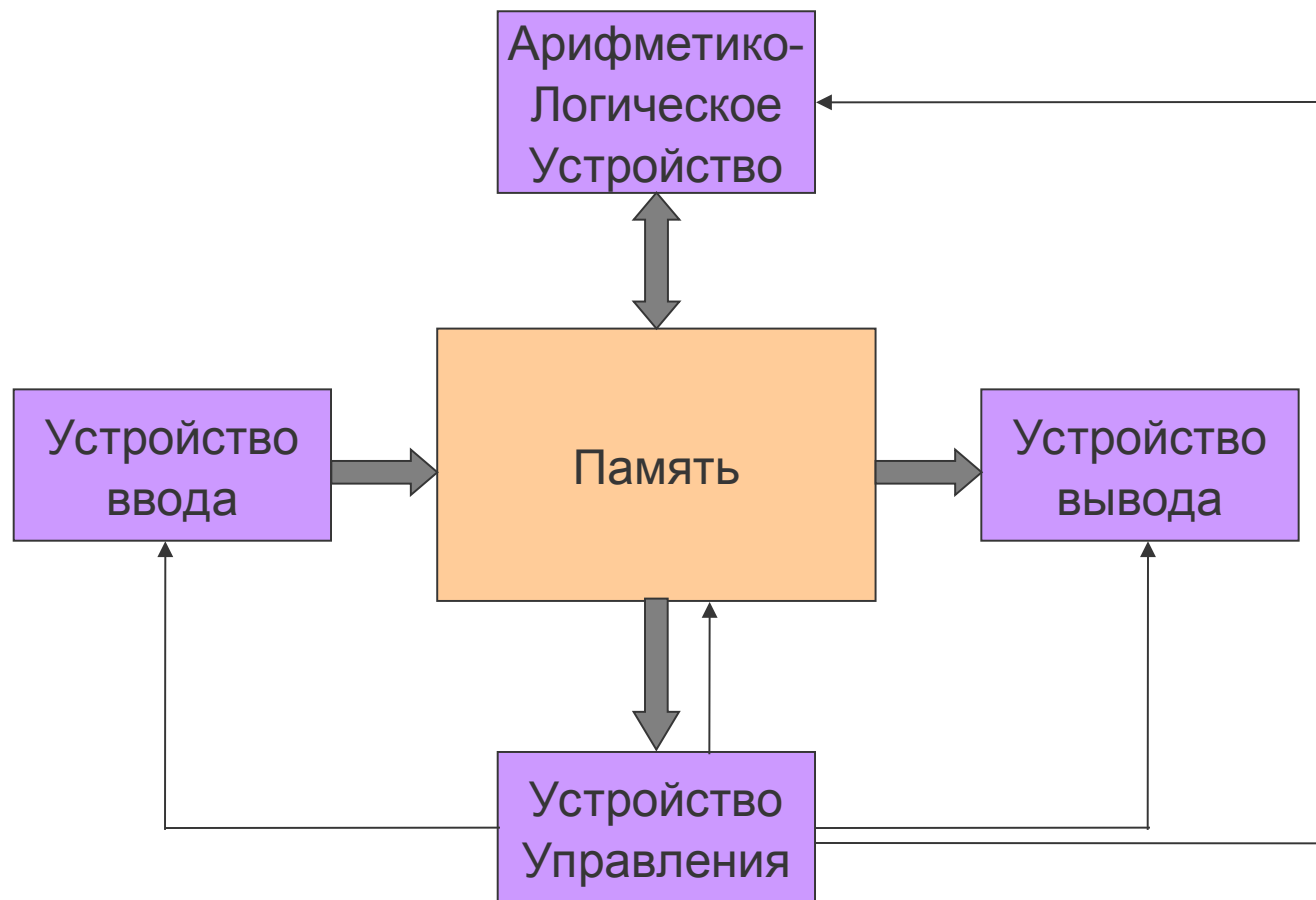
# Принцип иерархической организации памяти



- смысл иерархической организации – удешевление стоимости системы



# Архитектура фон Неймана



## ■ Принципы фон Неймана

- программного управления
- условного перехода
- хранимой программы
- иерархической организации памяти
- двоичной системы счисления

# Резюме

- В вычислительной технике используются в основном двоичная и шестнадцатеричная системы счисления
- Для измерения количества информации существуют различные подходы и специальные единицы
- В современной вычислительной технике информация представляется в дискретной форме



# Задание к следующей лекции

- Макарова, Волков. Информатика. Учебник для вузов.
  - гл. 1, 2, 11
- Савельев. Прикладная теория цифровых автоматов
  - гл. 1 – 4

