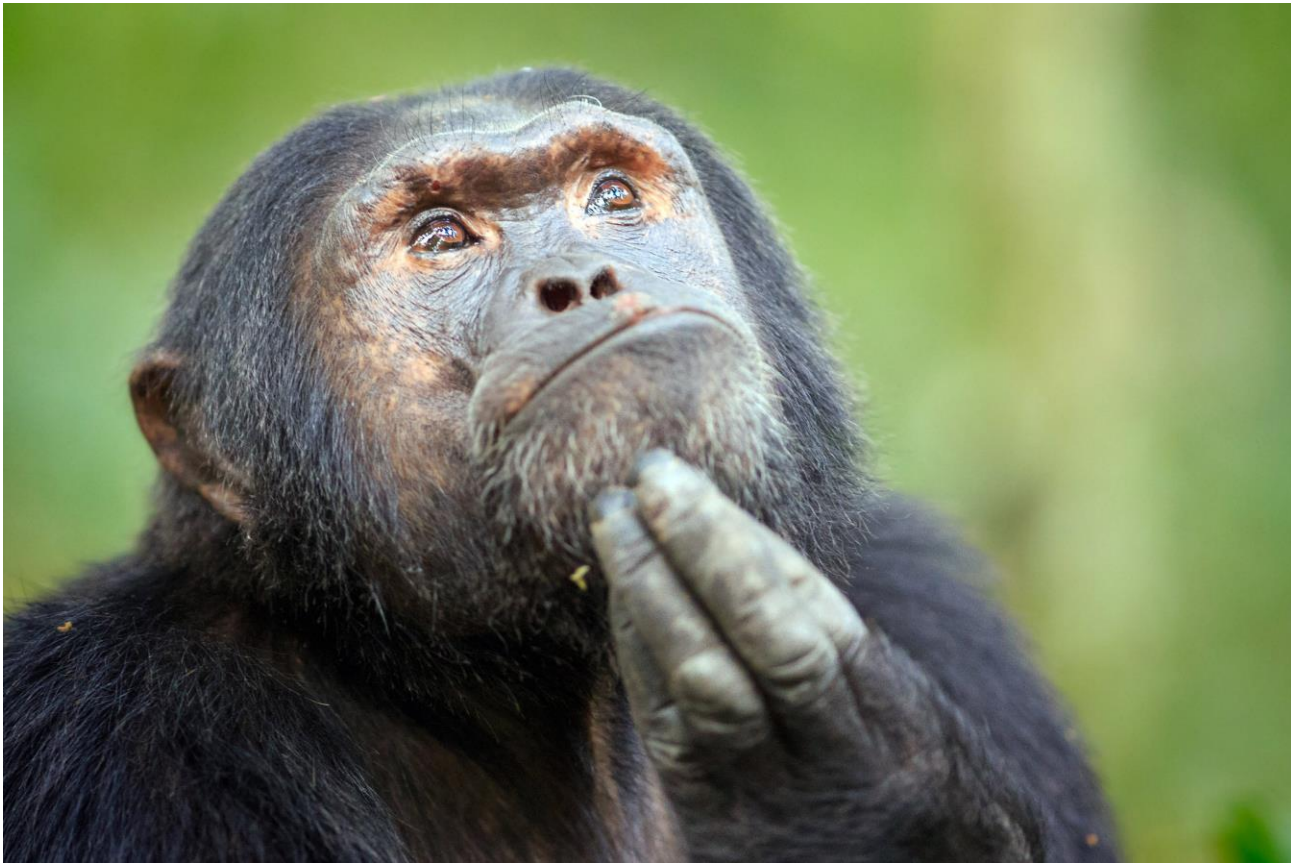


Лекция 2.

Двойная природа и внутреннее устройство компьютера



Для чего нужен компьютер?
С какой целью он создавался?

Замена человека машиной

- **Механизация** – замена физического/умственного труда человека машинным.
- **Автоматизация** – замена управленческого труда человека (машина реализует некоторую программу, алгоритм управления).
- **Кибернетизация** – замена человека в сфере принятия решений.

Механизация вычислений



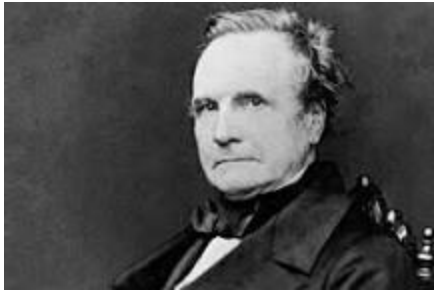
Суммирующая машина Паскаля (1652 год)



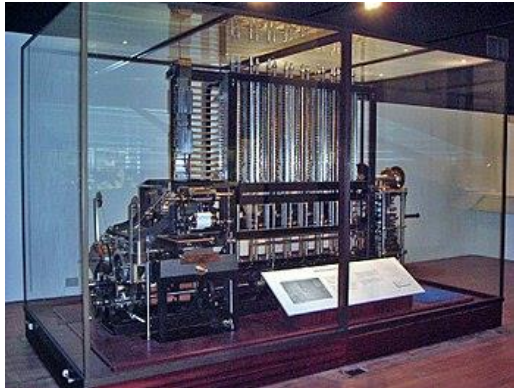
Арифмометр Лейбница (1673 год)

Выполняется только одно действие, промежуточные результаты записывались на бумаге

Автоматизация вычислений

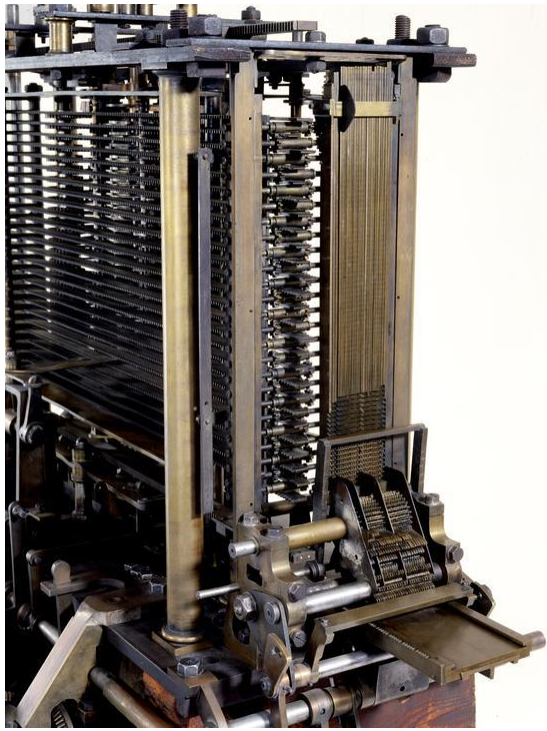


Чарльз Бэббидж (1791-1871), Англия



Разностная машина Бэббиджа (проект 1822 года) для составления таблиц значений функций

Автоматизация вычислений



Аналитическая машина Бэббиджа (идея 1834 года) - прообраз цифровых компьютеров:

- *Склад* для сохранения значений переменных и результатов вычислений (память)
- *Мельница* для выполнения арифметических действий над переменными (арифметическое устройство)
- Устройство управления последовательностью операций
- Механизм ввода исходных данных и инструкций с перфокарт
- Механизм печати результатов

"Сам процесс вычисления осуществляется с помощью алгебраических формул, записанных на перфорированных картах. Вся умственная работа сводится к написанию формул, пригодных для вычислений, производимых машиной, и неких простых указаний, в какой последовательности эти вычисления должны производиться" Луи Менабреа

Первая программа в мире



Ада Лавлейс (Байрон) (1817-1852), Англия

- Перевела и значительно дополнила работу Луи Менабреа.
- В комментариях привела алгоритм вычисления чисел Бернулли на аналитической машине Бэббиджа.
- Использовала термины "рабочая ячейка" (переменная) и "цикл".

"Суть и предназначение машины изменятся от того, какую информацию мы в нее вложим. Машина сможет писать музыку, рисовать картины и покажет науке такие пути, которые мы никогда и нигде не видели"

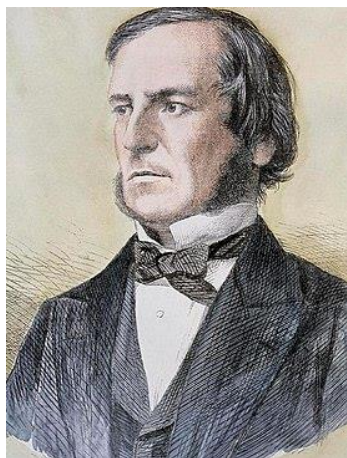
Двоичная система счисления



Готфрид Вильгельм Лейбниц (1646-1716), Германия

- Создал комбинаторику как науку.
- Заложил основы математической логики
- Описал двоичную систему счисления с цифрами 0 и 1.

Логика и математика



Джордж Буль (1815-1864), Англия

- Выполнение символических математических операций не только над числами, но и над множествами.
- Законы алгебры логики позволяют упрощать логические выражения точно так же, как элементарная алгебра упрощает числовые.

Логические переменные: 0 и 1

Логические операторы: AND (И), OR (ИЛИ), NOT (НЕ), XOR (Исключающее ИЛИ)

Логические выражения можно реализовать с помощью телеграфных реле или переключателей, проводов и лампочек.

Алгебра логики

Логические переменные: 0 (ложь) и 1 (истина)

Логические операторы: AND (И), OR (ИЛИ), NOT (НЕ), XOR (Исключающее ИЛИ)

Законы алгебры логики:

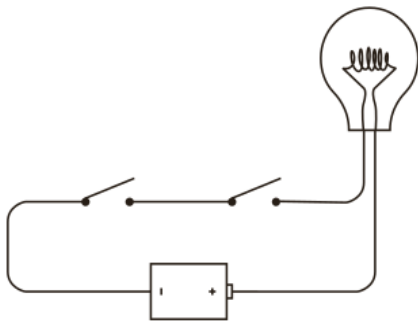
| название | для И | для ИЛИ |
|---------------------|--|--|
| двойного отрицания | $\overline{\overline{A}} = A$ | |
| исключения третьего | $A \cdot \overline{A} = 0$ | $A + \overline{A} = 1$ |
| операции с конст. | $A \cdot 0 = 0, A \cdot 1 = A$ | $A + 0 = A, A + 1 = 1$ |
| повторения | $A \cdot A = A$ | $A + A = A$ |
| поглощения | $A \cdot (A + B) = A$ | $A + A \cdot B = A$ |
| переместительный | $A \cdot B = B \cdot A$ | $A + B = B + A$ |
| сочетательный | $A \cdot (A \cdot B) = (A \cdot B) \cdot C$ | $A + (B + C) = (A + B) + C$ |
| распределительный | $A + B \cdot C = (A + B) \cdot (A + C)$ | $A \cdot (B + C) = A \cdot B + A \cdot C$ |
| законы де Моргана | $\overline{A \cdot B} = \overline{A} + \overline{B}$ | $\overline{A + B} = \overline{A} \cdot \overline{B}$ |

Логика и числа

- Группы логических переменных могут представлять числа в двоичной форме
- Логические операции в случае с двоичными числами могут объединяться для арифметических расчетов

Булевы выражения и окружающий мир

Логические выражения можно реализовать с помощью телеграфных реле или переключателей, проводов и лампочек.



Левый переключатель

Разомкнут
Разомкнут
Замкнут
Замкнут

Правый переключатель

Разомкнут
Замкнут
Разомкнут
Замкнут

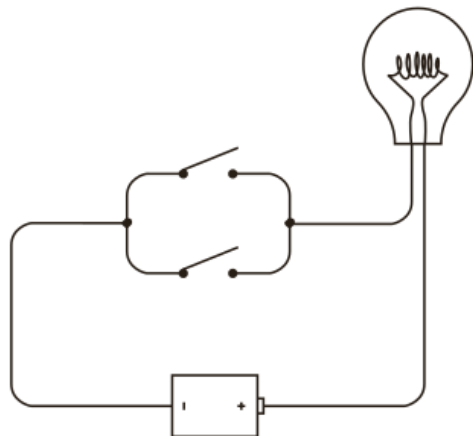
Лампочка

Не горит
Не горит
Не горит
Горит

| Последовательное соединение переключателей | 0 | 1 |
|---|---|---|
| | 0 | 0 |
| 1 | 0 | 1 |

| И | 0 | 1 |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |

Булевы выражения и окружающий мир



Левый переключатель

Разомкнут
Разомкнут
Замкнут
Замкнут

Правый переключатель

Разомкнут
Замкнут
Разомкнут
Замкнут

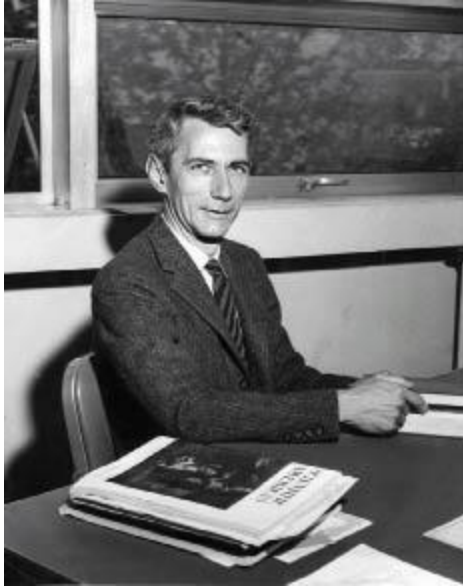
Лампочка

Не горит
Горит
Горит
Горит

| Параллельное соединение переключателей | 0 | 1 |
|--|---|---|
| 0 | 0 | 1 |
| 1 | 1 | 1 |

| ИЛИ | 0 | 1 |
|-----|---|---|
| 0 | 0 | 1 |
| 1 | 1 | 1 |

Булева алгебра и электрические цепи







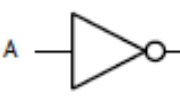

Клод Шеннон (1916-2001), США

- Указал на возможность реализовать логические выражения в виде электрических цепей (реле и переключатели)
- Ввел слово "бит" для обозначения двоичной цифры
- Предложил использовать бит как единицу измерения количества информации

Логические вентили

Вентиль - устройство, которое выдает результат булевой операции после введения в него входных данных.

Вентили реализуют логические функции и являются строительными блоками для построения компьютера.

| Тип | Символ | Таблица истинности | | | | | | | | | | | | | | | |
|-----------------|--|---|---|-------|-------|---|---|---|---|---|---|---|---|---|---|---|---|
| И |  | <table border="1"><thead><tr><th>A</th><th>B</th><th>Выход</th></tr></thead><tbody><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></tbody></table> | A | B | Выход | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| A | B | Выход | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | | | | | | | | | | | | | | | |
| или |  | <table border="1"><thead><tr><th>A</th><th>B</th><th>Выход</th></tr></thead><tbody><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></tbody></table> | A | B | Выход | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| A | B | Выход | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | | | | | | | | | | | | | | | |
| НЕ-И |  | <table border="1"><thead><tr><th>A</th><th>B</th><th>Выход</th></tr></thead><tbody><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></tbody></table> | A | B | Выход | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| A | B | Выход | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | | | | | | | | | | | | | | | |
| НЕ-или |  | <table border="1"><thead><tr><th>A</th><th>B</th><th>Выход</th></tr></thead><tbody><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></tbody></table> | A | B | Выход | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| A | B | Выход | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | | | | | | | | | | | | | | | |
| НЕТ |  | <table border="1"><thead><tr><th>A</th><th>Выход</th></tr></thead><tbody><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></tbody></table> | A | Выход | 0 | 1 | 1 | 0 | | | | | | | | | |
| A | Выход | | | | | | | | | | | | | | | | |
| 0 | 1 | | | | | | | | | | | | | | | | |
| 1 | 0 | | | | | | | | | | | | | | | | |
| Исключающее или |  | <table border="1"><thead><tr><th>A</th><th>B</th><th>Выход</th></tr></thead><tbody><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></tbody></table> | A | B | Выход | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| A | B | Выход | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | | | | | | | | | | | | | | | |

Вентили могут реализовываться в виде зубчатых колес, электромеханических реле, электрических схем, электронных устройств.

Вычисления с помощью вентиляей

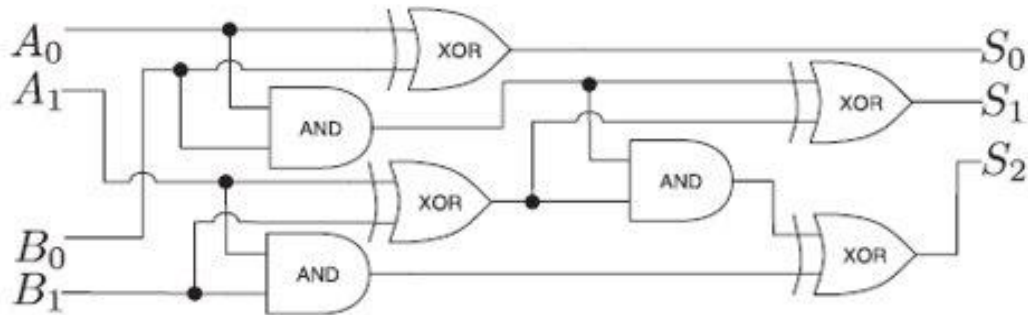


Рис. 1.6. Схема суммирования двухразрядных чисел, передаваемых парами логических переменных (A_1A_0 и B_1B_0) в трехразрядное число ($S_2S_1S_0$)

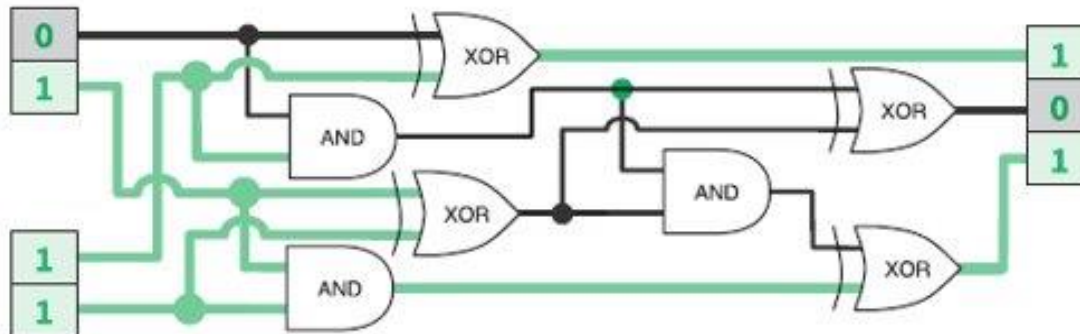


Рис. 1.7. Вычисление $2 + 3 = 5$
(в двоичном формате это $10 + 11 = 101$)

Реализация памяти

Память - компонент, позволяющий хранить и извлекать двоичные данные.

Память реализуется с помощью триггеров или защелок.

| S | R | Q (выход) | Действие |
|---|---|----------------------------------|------------------------|
| 0 | 0 | Поддерживает предыдущее значение | Удержание |
| 0 | 1 | 0 | Сброс |
| 1 | 0 | 1 | Установка |
| 1 | 1 | X | Недопустимое состояние |

SR-защелка для хранения одного бита

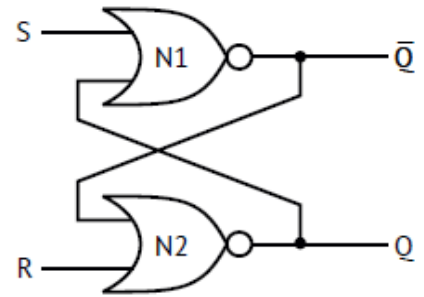
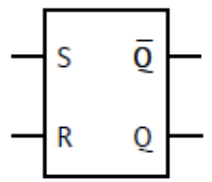


Рис. 6-2. SR-защелка, реализованная с помощью перекрестно соединенных вентилей НЕ-ИЛИ

Тело и душа компьютера

| Нематериальный мир, законы математики | Материальный мир, законы физики |
|--|--|
| Компьютеры работают в двоичной системе из 0 и 1 | Компьютеры - это электронные устройства, построенные на основе электрических цепей и цифровых схем |

Нематериальный мир, законы математики

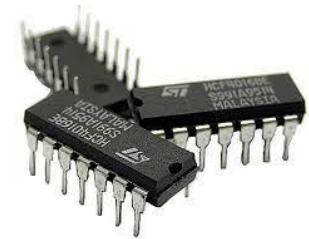
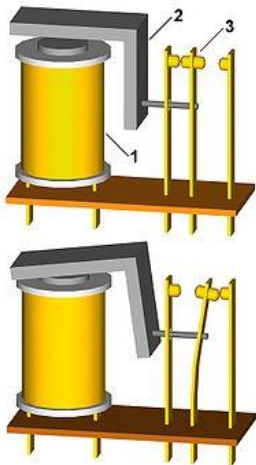
- **Компьютеры работают в двоичной системе из 0 и 1**
 - Все данные и инструкции - последовательности 0 и 1
 - Действуют законы двоичной логики (булева алгебра)
 - Арифметические операции реализуются соединением между собой логических элементов И, ИЛИ, НЕ
 - Все операции процессора основаны на комбинационных логических системах разной сложности
 - Последовательные логические схемы обладают памятью - способностью хранить записи о прошлом

Нематериальный мир, законы математики

- **Компьютеры работают в двоичной системе из 0 и 1**
 - Все данные и инструкции - последовательности 0 и 1
 - Действуют законы двоичной логики (булева алгебра)
 - Арифметические операции реализуются соединением между собой логических элементов И, ИЛИ, НЕ
 - Все операции процессора основаны на комбинационных логических системах разной сложности
 - Последовательные логические схемы обладают памятью - способностью хранить записи о прошлом

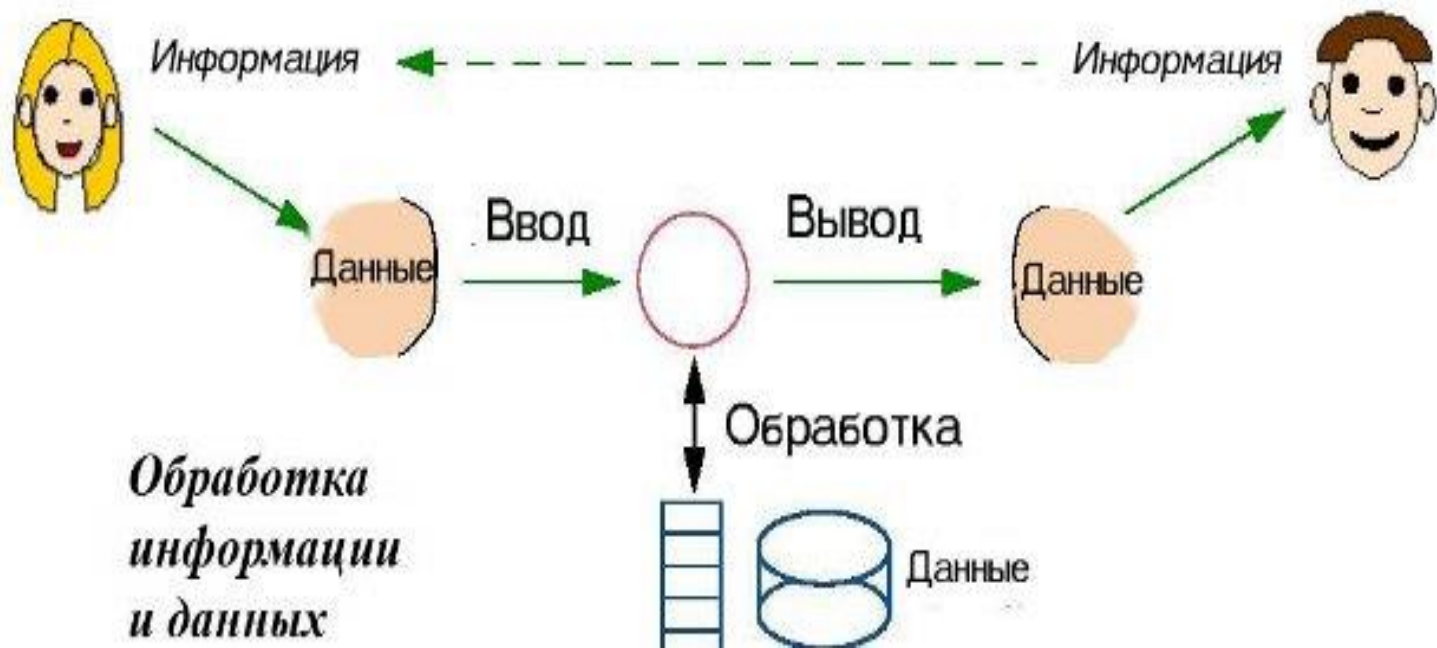
Материальный мир, законы физики

- Компьютеры - это электронные устройства, построенные на основе электрических цепей и цифровых схем
 - Для представления 0 и 1 используются уровни напряжения (низкое - 0, высокое - 1)
 - Логические элементы реализуются как комбинации управляемых ключей на разных элементных базах: электромагнитные реле, электронные лампы, транзисторы, интегральные микросхемы



Компьютеры могут: ...

- Хранить и искать данные в своей памяти.
- Выполнять (очень быстро) базовые операции над данными: сравнение, замена, арифметические операции.
- Обмениваться данными с другими устройствами (компьютерами, мониторами, принтерами, телефонами, датчиками и т.д.).



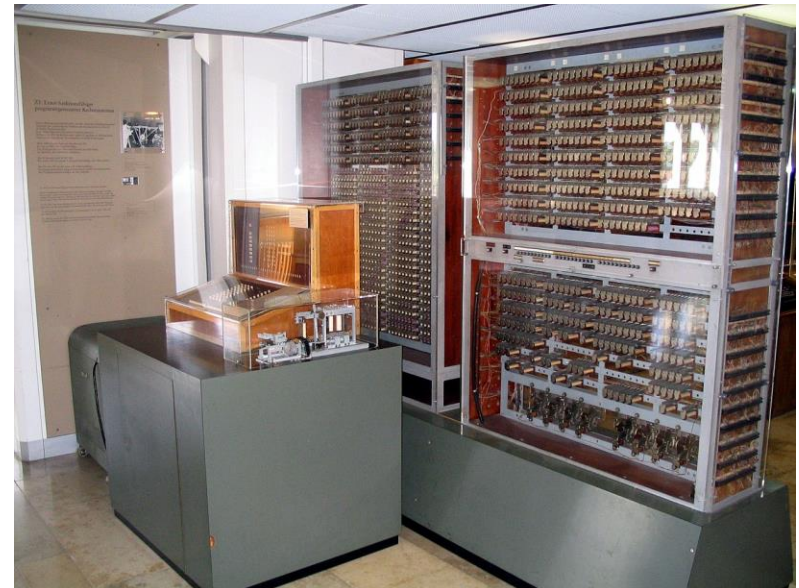
Первый реальный компьютер мире



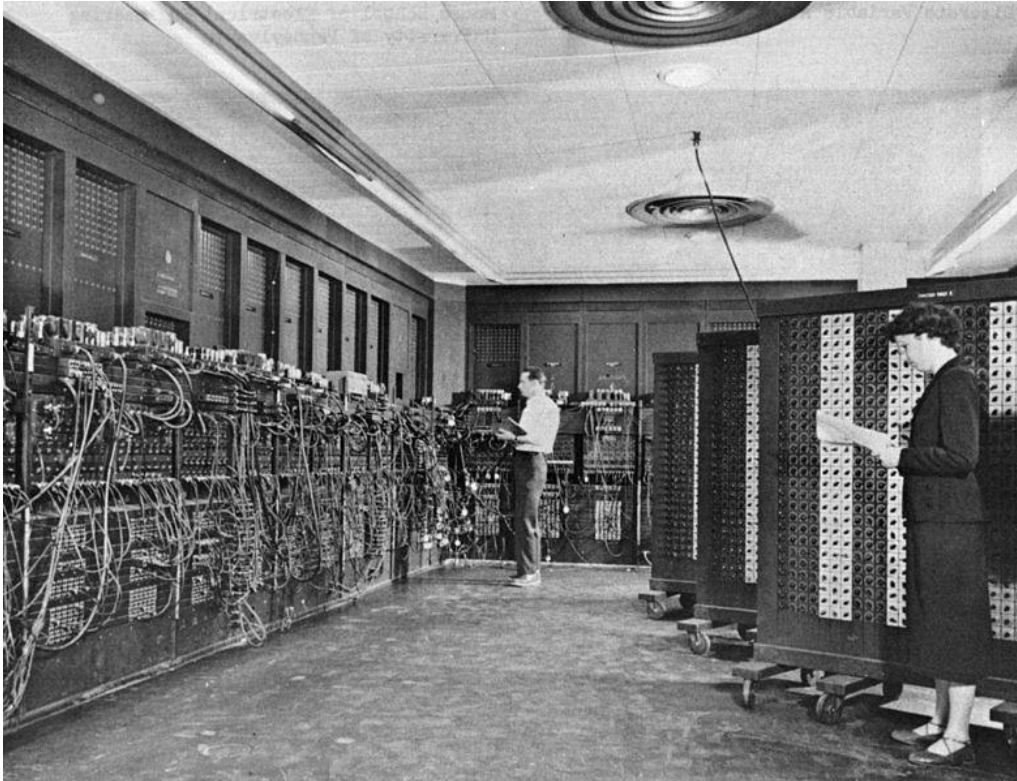
Конрад Цузе (1910-1995), Германия

- Создал первый реально работающий электро-механический программируемый компьютер Z3 (1941 год)
- Создал первый язык программирования высокого уровня Планкалкюль (1948 год).

Z3 - ограниченно программируемый (не было условных переходов) двоичный вычислитель на основе телефонных реле. Устройство хранения данных также на реле.



Первые компьютерные системы: ограниченный набор исполняемых команд и программ



1945 год. Компьютер ENIAC (Electronic Numerical Integrator and Computer)

Площадь — 167 кв.м.

Вес — 27 тонн

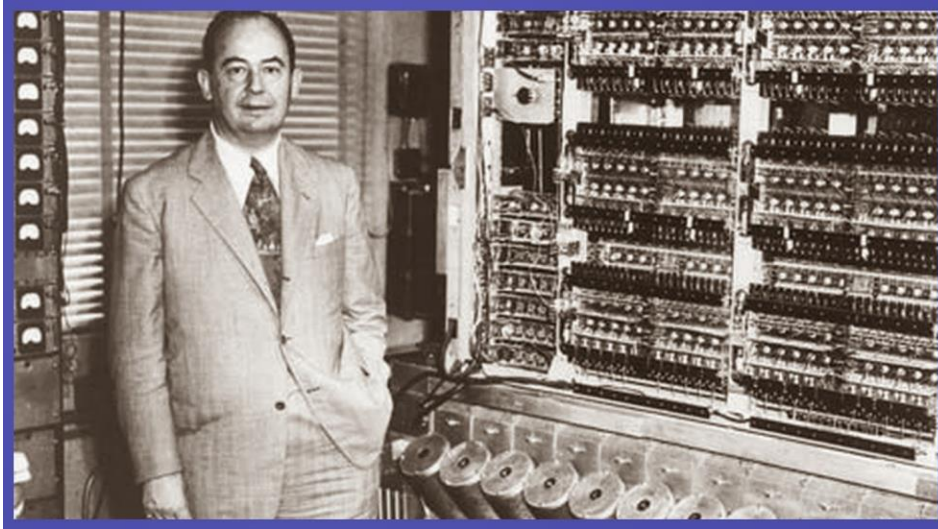
18 000 электронных ламп

Десятичная арифметика

Ввод/вывод данных — перфокарты

Программирование — ручная установка переключателей

Идея хранения программ в общей памяти



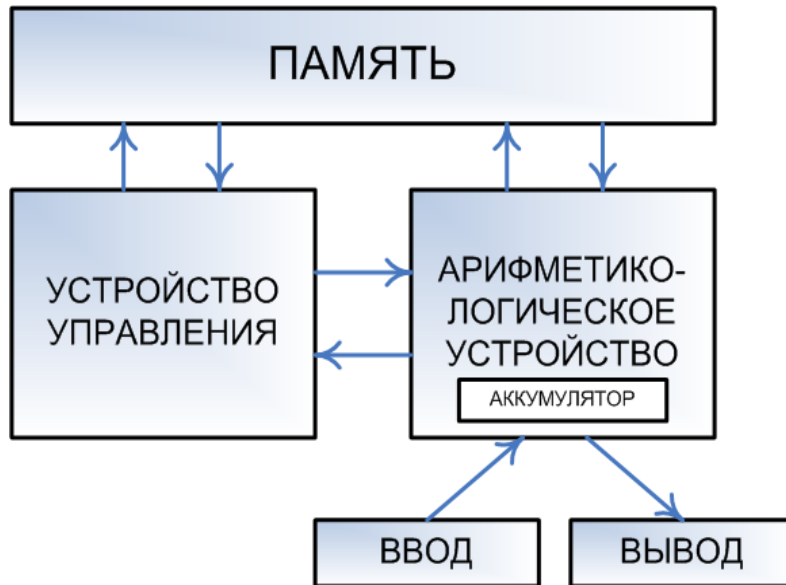
Джон фон Нейман
(1903-1957)

1946 год, статья Дж. фон Нейман, Г. Голдстайна и А. Беркса «Предварительное рассмотрение логической конструкции электронного вычислительного устройства» в журнале Nature

Принципы фон Неймана

- **Двоичное кодирование.** Вся информация кодируется с помощью двоичных сигналов (двоичных цифр, битов) и разделяется на единицы, называемые словами.
- **Однородность памяти.** Программы и данные хранятся в одной и той же памяти. Компьютер не различает, что хранится в данной ячейке памяти — число, текст или команда.
- **Адресуемость памяти.** Память состоит из пронумерованных ячеек; процессору в произвольный момент времени доступна любая ячейка. Можно давать имена областям памяти.
- **Последовательного программного управления.** Предполагает, что программа состоит из набора команд, которые выполняются процессором автоматически друг за другом в определенной последовательности.

Архитектура фон Неймана



Машина работает в двоичном коде.

Программы и данные хранятся совместно в одном и том же пространстве памяти.

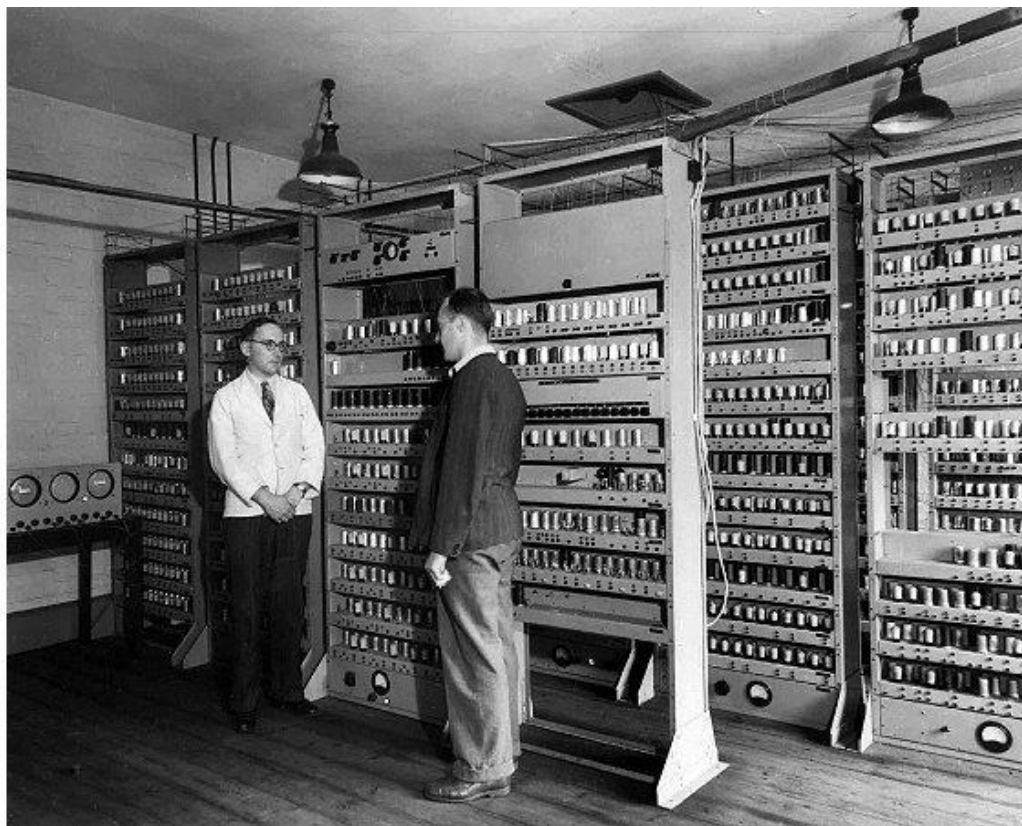
УУ - извлекает инструкции, расшифровывает их, координирует выполнение

АЛУ - выполняет арифметические и логические операции

Программа для фон-неймановской машины – последовательность нулей и единиц, одни участки которой кодируют команды, а другие – данные.

Подобная последовательность называется **машинным кодом**.

Первый компьютер с архитектурой фон Неймана



1949 год. Компьютер EDSAC
(Electronic Delay Storage
Automatic Computer)
Кембриджский университет

Площадь — 20 кв.м.

3 000 электронных ламп

Поддерживал язык ассемблера и подпрограммы.

В 1952 году на нем была написана первая игра «Крестики-нолики».

Сравнительная таблица основных параметров ЭВМ

| Параметры сравнения | Поколения ЭВМ | | | |
|-------------------------------|---|---|--|--|
| | Первое | Второе | Третье | Четвертое |
| Период времени | 1946 - 1959 | 1960 - 1969 | 1970 - 1979 | С 1980 г. |
| Элементная база (для УУ, АЛУ) | Электронные (или электрические) лампы | Полупроводники (транзисторы) | Интегральные схемы | Большие интегральные схемы (БИС) |
| Основной тип ЭВМ | Большие | | Малые (мини) | Микро ЭВМ |
| Основные устройства ввода | Пульт, перфоленточный, перфокарточный ввод | Добавился алфавитно-цифровой дисплей, клавиатура | Алфавитно-цифровой дисплей, клавиатура | Цветной графический дисплей, сканер, клавиатура |
| Основные устройства вывода | Алфавитно-цифровое печатающее устройство (АЦПУ), перфоленточный вывод | | Графопостроитель, принтер | |
| Внешняя память | Магнитные ленты, барабаны, перфоленты, перфокарты | Добавился магнитный диск | Перфоленты, магнитный диск | Магнитные и оптические диски |
| Ключевые решения в ПО | Универсальные языки программирования, трансляторы | Пакетные операционные системы, оптимизирующие трансляторы | Интерактивные операционные системы, структурированные языки программирования | Дружественность ПО, сетевые операционные системы |
| Режим работы ЭВМ | Однопрограммный | Пакетный | Разделения времени | Персональная работа и сетевая обработка данных |
| Цель использования ЭВМ | Научно-технические расчеты | Технические и экономические расчеты | Управление и экономические расчеты | Телекоммуникации, информационное обслуживание |

Хранение в компьютере числовых и символьных данных

Основы двоичной системы

1 бит = 0, 1

1 байт = 8 бит

1 слово = 8 байт = 64 бит (раньше 4 байт = 32 бит). Равно разрядности регистров процессора и/или шины данных.

Основы двоичной системы

1 бит = 0, 1

1 байт = 8 бит

1 слово = 8 байт = 64 бит (раньше 4 байт = 32 бит). Равно разрядности регистров процессора и/или шины данных.

Сколько разных чисел можно закодировать 1 байтом?

Основы двоичной системы

1 бит = 0, 1

1 байт = 8 бит

1 слово = 8 байт = 64 бит (раньше 4 байт = 32 бит). Равно разрядности регистров процессора и/или шины данных.

Сколько разных чисел можно закодировать 1 байтом?

$$00000000_2 = 0_{10}$$

$$00000001_2 = 1_{10}$$

$$00000010_2 = 2_{10}$$

$$00000011_2 = 3_{10}$$

...

$$11111111_2 = 255_{10}$$

Целые положительные числа

Перевод из десятичной в двоичную:

1. Если текущее число меньше 2, то записать его в младший разряд результата, выполнение прекратить. Иначе, разделить текущее число с остатком на 2.
2. Остаток записать в младший разряд результата.
3. Применить пункт 1 к частному.

$$\begin{array}{r} 567 \mid \frac{2}{566} \mid \frac{2}{283} \mid \frac{2}{141} \mid \frac{2}{70} \mid \frac{2}{35} \mid \frac{2}{17} \mid \frac{2}{8} \mid \frac{2}{4} \mid \frac{2}{2} \mid \frac{2}{1} \\ \underline{-566} \quad \underline{-282} \quad \underline{-140} \quad \underline{-70} \quad \underline{-34} \quad \underline{-16} \quad \underline{-8} \quad \underline{-4} \quad \underline{-2} \\ \mathbf{1} \quad \mathbf{1} \quad \mathbf{1} \quad \mathbf{0} \quad \mathbf{1} \quad \mathbf{1} \quad \mathbf{1} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{1} \end{array}$$

$$567_{10} = 1000110111_2$$

Перевод из двоичной в десятичную:

$$1 \cdot 2^9 + 0 \cdot 2^8 + 0 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 512 + 0 + 0 + 0 + 32 + 16 + 0 + 4 + 2 + 1 = 567$$

Целые числа со знаком

Как учесть знак числа, как хранить отрицательные числа?

Целые числа со знаком

Как учесть знак числа, как хранить отрицательные числа?

Под знак числа отводят один бит (двоичный разряд). Остальные разряды определяют модуль числа.

Целые числа со знаком

Как учесть знак числа, как хранить отрицательные числа?

Под знак числа отводят один бит (двоичный разряд). Остальные разряды определяют модуль числа.

Прямой код

Старший бит – знак числа (0, если «+» и 1, если «-»). Остальные биты не меняются.

Два недостатка:

$$+0 = 0.0000000_2 \text{ и } -0 = 1.0000000_2$$

$$a = 65_{10} = 0.1000001_2$$

$$-a = -65_{10} = 1.1000001_2$$

$$a+(-a) = 0.0110010 + 1.0110010 = 1.1100100 \neq 0$$

Целые числа со знаком

Как учесть знак числа, как хранить отрицательные числа?

Под знак числа отводят один бит (двоичный разряд). Остальные разряды определяют модуль числа.

Прямой код

Старший бит – знак числа (0, если «+» и 1, если «-»). Остальные биты не меняются.

Два недостатка:

$$+0 = 0.0000000_2 \text{ и } -0 = 1.0000000_2$$

$$a = 65_{10} = 0.1000001_2$$

$$-a = -65_{10} = 1.1000001_2$$

$$a+(-a) = 0.0110010 + 1.0110010 = 1.1100100 \neq 0$$

Обратный код

Старший бит – знак числа. Остальные биты инвертируются (0 -> 1, 1 -> 0).

$$+0 = 0.0000000_2 \text{ и } -0 = 1.1111111_2$$

$$a = 65_{10} = 0.1000001_2$$

$$-a = -65_{10} = 1.0111110_2$$

В сложении участвуют все биты, единица переноса из знакового разряда прибавляется к младшему разряду суммы.

$$a+(-a) = 0.1000001_2 + 1.0111110_2 = 1.1111111_2 = -0$$

Целые числа со знаком

Дополнительный код

Старший бит – знак числа. Остальные биты инвертируются, к результату прибавляется 1.

Единственная запись нуля: $0 = 0.0000000$

В сложении участвуют все биты, единица переноса из знакового разряда отбрасывается.

$$a = 65_{10} = 0.1000001_2$$

$$-a = -65_{10} = 1.0111111_2$$

$$a + (-a) = 0.1000001_2 + 1.0111111_2 = 0.0000000_2 = 0 \text{ (1 в 9 разряде отбросили)}$$

Вещественные числа

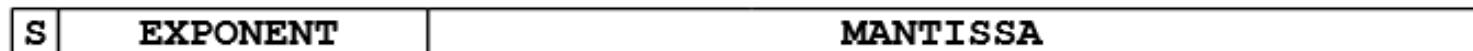
1985 год, стандарт IEEE754 «Двоичная арифметика с плавающей точкой»

Число одинарной точности от $-3,4 \cdot 10^{38}$ до $3,4 \cdot 10^{38}$ хранится в 32-битовом контейнере, разделенном на три части:

- S (1 бит) для хранения знака
- E (8 бит) для хранения экспоненты (порядка)
- M (23 бита) для хранения мантиссы



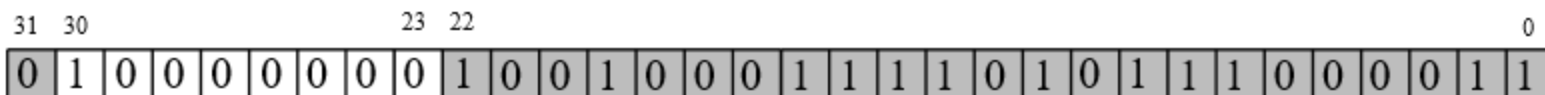
Внутренности числа с плавающей запятой.



Три части числа с плавающей запятой.

Числа представляются с помощью формулы $(-1)^S * 1, M * 2^{(E-127)}$

$$3,14 = (-1)^0 * 1,57 * 2^{(128-127)}$$



Двоичное представление с плавающей точкой числа 3,14.

Вещественные числа

How to draw an owl

1.



1. Draw some circles

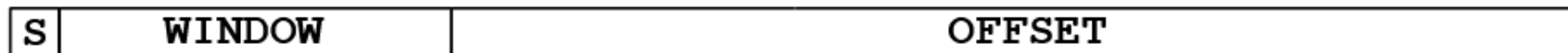
2.



2. Draw the rest of the fucking owl

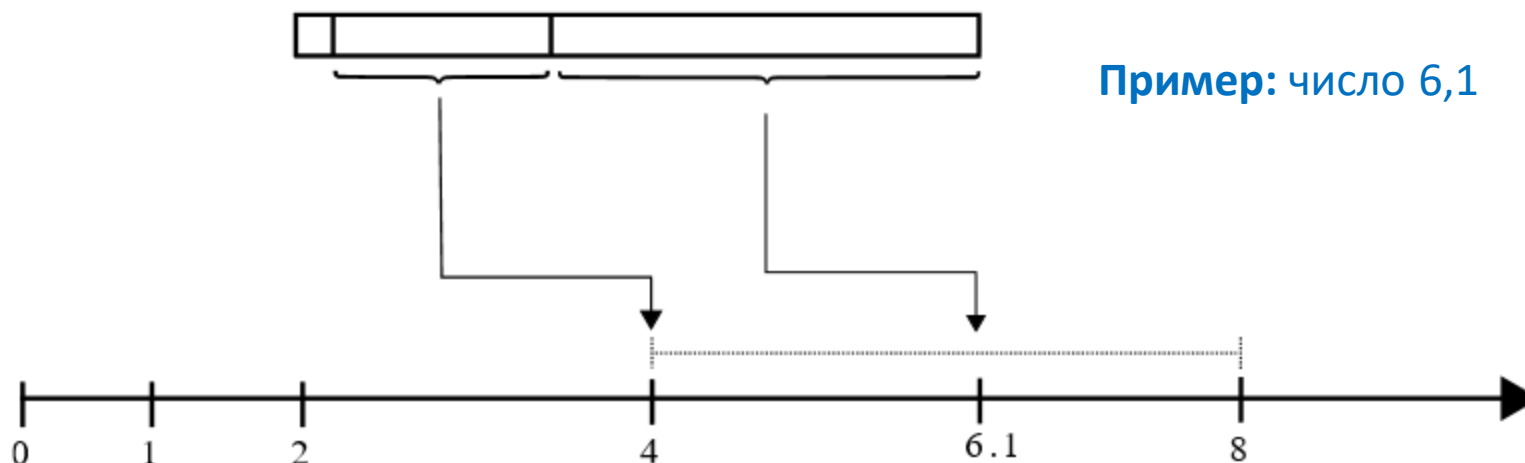
Вещественные числа

«Наглядное объяснение чисел с плавающей запятой» <https://habrahabr.ru/post/337260/>



Три части числа с плавающей запятой.

- **Окно** – число, определяющее интервал $[0, 1], [1, 2], [2, 4], [4, 8], \dots, [2^{127}, 2^{128}]$, в который попадает целая часть числа.
- Отрезок Window делится на $2^{23} = 8\,388\,608$ сегментов.
- **Смещение** определяет номер сегмента, в которые попадает число.

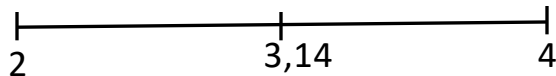


Значение 6,1 аппроксимированное с помощью числа с плавающей запятой.

Вещественные числа

Число 3,14

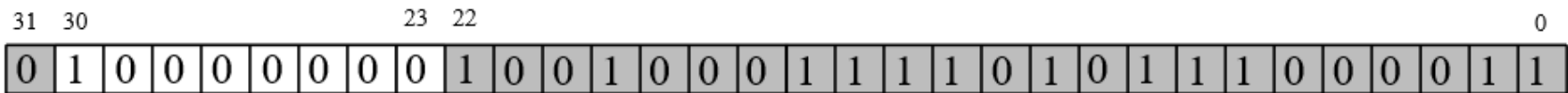
- $3,14 > 0$, поэтому $S=0$.
- $2 < 3,14 < 4$, поэтому $E=128$ (так как окно – это 2^{E-127}).
- $(3,14 - 2)/(4-2) = 0,57$ – местоположение точки 3,14 на интервале $[2, 4]$.



- Вычисляем номер смещения: $M = 2^{23} * 0,57 = 4781507$.

В двоичном виде:

- $S = 0 = 0b$
- $E = 128 = 10000000b$
- $M = 4781507 = 10010001111010111000011b$



Двоичное представление с плавающей точкой числа 3,14.

Шестнадцатеричная система

Для представления двоичных данных в удобном виде.

Двоичная тетрада = один 16-ичный знак:

$$0000_2 = 0_{16}$$

$$0001_2 = 1_{16}$$

$$0010_2 = 2_{16}$$

$$0011_2 = 3_{16}$$

$$0100_2 = 4_{16}$$

$$0101_2 = 5_{16}$$

$$0110_2 = 6_{16}$$

$$0111_2 = 7_{16}$$

$$1000_2 = 8_{16}$$

$$1001_2 = 9_{16}$$

$$1010_2 = A_{16}$$

$$1011_2 = B_{16}$$

$$1100_2 = C_{16}$$

$$1101_2 = D_{16}$$

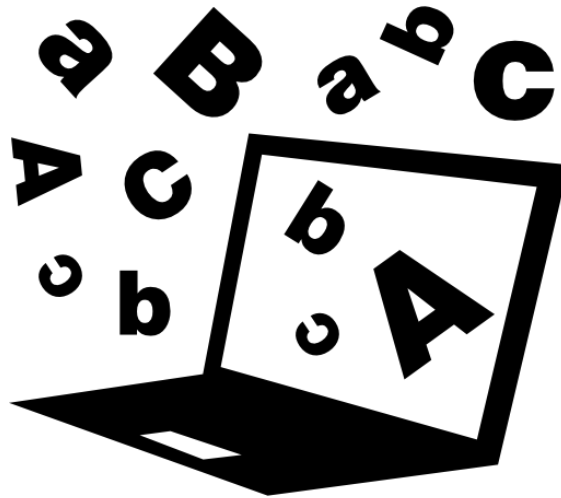
$$1110_2 = E_{16}$$

$$1111_2 = F_{16}$$

Символы, буквы, знаки

Для работы с символами в компьютере нужно:

1. Определить множество символов, которые должны отображаться.
2. Сопоставить каждому символу число (код).
3. Определить способ хранения этих чисел в памяти компьютера (кодирование).



Первые кодировки

Вначале требовалось кодировать минимум символов:

- 10 цифр
- 26 заглавных букв английского алфавита
- 26 строчных букв
- математические знаки $+ - / * = > < %$
- знаки препинания $. , ! ? : ; ' "$
- различные скобки $() \{ \} []$
- служебные символы $_ ^ \% \$ @ |$
- 32 непечатных управляющих символов

Первые кодировки

Вначале требовалось кодировать минимум символов:

- 10 цифр
- 26 заглавных букв английского алфавита
- 26 строчных букв
- математические знаки $+ - / * = > < %$
- знаки препинания $, ! ? : ; ' "$
- различные скобки $() \{ \} []$
- служебные символы $_ ^ \% \$ @ |$
- 32 непечатных управляющих символов

Для этого хватало **128 символов**.

Для хранения информации использовались **7 бит** в байте (старший бит – для контроля или других целей).

Кодовая таблица ASCII (American Standard Code for Information Interchange)

| символ | 10- Б код | 2-Б код | символ | 10- Б код | 2-Б код | символ | 10-Б код | 2-Б код | символ | 10-Б код | 2-Б код |
|--------|-----------------|----------|--------|-----------------|----------|--------|-------------|----------|--------|-------------|----------|
| | 32 | 00100000 | 8 | 56 | 00111000 | P | 80 | 01010000 | h | 104 | 01101000 |
| ! | 33 | 00100001 | 9 | 57 | 00111001 | Q | 81 | 01010001 | i | 105 | 01101001 |
| " | 34 | 00100010 | : | 58 | 00111010 | R | 82 | 01010010 | j | 106 | 01101010 |
| # | 35 | 00100011 | ; | 59 | 00111011 | S | 83 | 01010011 | k | 107 | 01101011 |
| \$ | 36 | 00100100 | < | 60 | 00111100 | T | 84 | 01010100 | l | 108 | 01101100 |
| % | 37 | 00100101 | = | 61 | 00111101 | U | 85 | 01010101 | m | 109 | 01101101 |
| & | 38 | 00100110 | > | 62 | 00111110 | V | 86 | 01010110 | n | 110 | 01101110 |
| ' | 39 | 00100111 | ? | 63 | 00111111 | W | 87 | 01010111 | o | 111 | 01101111 |
| (| 40 | 00101000 | @ | 64 | 01000000 | X | 88 | 01011000 | p | 112 | 01110000 |
|) | 41 | 00101001 | A | 65 | 01000001 | Y | 89 | 01011001 | q | 113 | 01110001 |
| * | 42 | 00101010 | B | 66 | 01000010 | Z | 90 | 01011010 | r | 114 | 01110010 |
| + | 43 | 00101011 | C | 67 | 01000011 | [| 91 | 01011011 | s | 115 | 01110011 |
| , | 44 | 00101100 | D | 68 | 01000100 | \ | 92 | 01011100 | t | 116 | 01110100 |
| - | 45 | 00101101 | E | 69 | 01000101 |] | 93 | 01011101 | u | 117 | 01110101 |
| . | 46 | 00101110 | F | 70 | 01000110 | ^ | 94 | 01011110 | v | 118 | 01110110 |
| / | 47 | 00101111 | G | 71 | 01000111 | _ | 95 | 01011111 | w | 119 | 01110111 |
| 0 | 48 | 00110000 | H | 72 | 01001000 | ` | 96 | 01100000 | x | 120 | 01111000 |
| 1 | 49 | 00110001 | I | 73 | 01001001 | a | 97 | 01100001 | y | 121 | 01111001 |
| 2 | 50 | 00110010 | J | 74 | 01001010 | b | 98 | 01100010 | z | 122 | 01111010 |
| 3 | 51 | 00110011 | K | 75 | 01001011 | c | 99 | 01100011 | { | 123 | 01111011 |
| 4 | 52 | 00110100 | L | 76 | 01001100 | d | 100 | 01100100 | | 124 | 01111100 |
| 5 | 53 | 00110101 | M | 77 | 01001101 | e | 101 | 01100101 | } | 125 | 01111101 |
| 6 | 54 | 00110110 | N | 78 | 01001110 | f | 102 | 01100110 | ~ | 126 | 01111110 |
| 7 | 55 | 00110111 | O | 79 | 01001111 | g | 103 | 01100111 | □ | 127 | 01111111 |

Кодовая таблица ASCII

| символ | 10- B код | 2-B код | символ | 10- B код | 2-B код | символ | 10- B код | 2-B код | символ | 10- B код | 2-B код |
|--------|-----------------|----------|--------|-----------------|----------|--------|-----------------|----------|--------|-----------------|----------|
| | 32 | 00100000 | 8 | 56 | 00111000 | P | 80 | 01010000 | h | 104 | 01101000 |
| ! | 33 | 00100001 | 9 | 57 | 00111001 | Q | 81 | 01010001 | i | 105 | 01101001 |
| " | 34 | 00100010 | : | 58 | 00111010 | R | 82 | 01010010 | j | 106 | 01101010 |
| # | 35 | 00100011 | ; | 59 | 00111011 | S | 83 | 01010011 | k | 107 | 01101011 |
| \$ | 36 | 00100100 | < | 60 | 00111100 | T | 84 | 01010100 | l | 108 | 01101100 |
| % | 37 | 00100101 | = | 61 | 00111101 | U | 85 | 01010101 | m | 109 | 01101101 |
| & | 38 | 00100110 | > | 62 | 00111110 | V | 86 | 01010110 | n | 110 | 01101110 |
| ' | 39 | 00100111 | ? | 63 | 00111111 | W | 87 | 01010111 | o | 111 | 01101111 |
| (| 40 | 00101000 | @ | 64 | 01000000 | X | 88 | 01011000 | p | 112 | 01110000 |
|) | 41 | 00101001 | A | 65 | 01000001 | Y | 89 | 01011001 | q | 113 | 01110001 |
| * | 42 | 00101010 | B | 66 | 01000010 | Z | 90 | 01011010 | r | 114 | 01110010 |
| + | 43 | 00101011 | C | 67 | 01000011 | [| 91 | 01011011 | s | 115 | 01110011 |
| , | 44 | 00101100 | D | 68 | 01000100 | \ | 92 | 01011100 | t | 116 | 01110100 |
| - | 45 | 00101101 | E | 69 | 01000101 |] | 93 | 01011101 | u | 117 | 01110101 |
| . | 46 | 00101110 | F | 70 | 01000110 | ^ | 94 | 01011110 | v | 118 | 01110110 |
| / | 47 | 00101111 | G | 71 | 01000111 | _ | 95 | 01011111 | w | 119 | 01110111 |
| 0 | 48 | 00110000 | H | 72 | 01001000 | ` | 96 | 01100000 | x | 120 | 01111000 |
| 1 | 49 | 00110001 | I | 73 | 01001001 | a | 97 | 01100001 | y | 121 | 01111001 |
| 2 | 50 | 00110010 | J | 74 | 01001010 | b | 98 | 01100010 | z | 122 | 01111010 |
| 3 | 51 | 00110011 | K | 75 | 01001011 | c | 99 | 01100011 | { | 123 | 01111011 |
| 4 | 52 | 00110100 | L | 76 | 01001100 | d | 100 | 01100100 | | 124 | 01111100 |
| 5 | 53 | 00110101 | M | 77 | 01001101 | e | 101 | 01100101 | } | 125 | 01111101 |
| 6 | 54 | 00110110 | N | 78 | 01001110 | f | 102 | 01100110 | ~ | 126 | 01111110 |
| 7 | 55 | 00110111 | O | 79 | 01001111 | g | 103 | 01100111 | □ | 127 | 01111111 |

Почему большие и маленькие буквы идут не сразу друг за другом?

Кодовые страницы

Как кодировать символы других алфавитов?

Кодовые страницы

Используем все 8 бит в байте. Числам от 0 до 255 сопоставляются графические образы символов – появляются разные кодовые страницы.

| | | | | | | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П |
| 128 | 129 | 130 | 131 | 132 | 133 | 134 | 135 | 136 | 137 | 138 | 139 | 140 | 141 | 142 | 143 |
| Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я |
| 144 | 145 | 146 | 147 | 148 | 149 | 150 | 151 | 152 | 153 | 154 | 155 | 156 | 157 | 158 | 159 |
| а | б | в | г | д | е | ж | з | и | й | к | л | м | н | о | п |
| 160 | 161 | 162 | 163 | 164 | 165 | 166 | 167 | 168 | 169 | 170 | 171 | 172 | 173 | 174 | 175 |
| ▒ | ▒ | ▒ | | ┌ | ┐ | ┌ | ┐ | ┌ | ┐ | ┌ | ┐ | ┌ | ┐ | ┌ | ┐ |
| 176 | 177 | 178 | 179 | 180 | 181 | 182 | 183 | 184 | 185 | 186 | 187 | 188 | 189 | 190 | 191 |
| └ | └ | └ | └ | — | ┌ | ┐ | ┐ | ┐ | ┐ | ┐ | ┐ | ┐ | ┐ | ┐ | ┐ |
| 192 | 193 | 194 | 195 | 196 | 197 | 198 | 199 | 200 | 201 | 202 | 203 | 204 | 205 | 206 | 207 |
| Ш | Т | П | Щ | Е | Р | П | ┌ | ┐ | ┐ | ┐ | ■ | ■ | ■ | ■ | ■ |
| 208 | 209 | 210 | 211 | 212 | 213 | 214 | 215 | 216 | 217 | 218 | 219 | 220 | 221 | 222 | 223 |
| р | с | т | у | ф | х | ц | ч | ш | щ | ъ | ы | ь | э | ю | я |
| 224 | 225 | 226 | 227 | 228 | 229 | 230 | 231 | 232 | 233 | 234 | 235 | 236 | 237 | 238 | 239 |
| Ё | ё | Є | є | Ї | ї | Ў | ў | ° | • | • | | | | | |
| 240 | 241 | 242 | 243 | 244 | 245 | 246 | 247 | 248 | 249 | 250 | | | | | |

Шрифты-кодировки

CP866 (DOS)

CP1251 (Windows)

| | | | | | | | | | | | | | | | |
|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Á | à | , | è | „ | ... | † | ‡ | € | % | É | < | й | Й | Ó | Ú |
| 128 | 129 | 130 | 131 | 132 | 133 | 134 | 135 | 136 | 137 | 138 | 139 | 140 | 141 | 142 | 143 |
| á | ‘ | ’ | “ | ” | • | — | — | è | ™ | é | > | ò | í | ó | ú |
| 144 | 145 | 146 | 147 | 148 | 149 | 150 | 151 | 152 | 153 | 154 | 155 | 156 | 157 | 158 | 159 |
| nbsp | ÿ | Ы | Э | ж | Ы | ı | § | Ë | © | Ю | « | ¬ | shy | ® | Я |
| 160 | 161 | 162 | 163 | 164 | 165 | 166 | 167 | 168 | 169 | 170 | 171 | 172 | 173 | 174 | 175 |
| ° | ± | Ы | Э | ’ | μ | ¶ | • | ë | № | ю | » | э | ю | я | я |
| 176 | 177 | 178 | 179 | 180 | 181 | 182 | 183 | 184 | 185 | 186 | 187 | 188 | 189 | 190 | 191 |
| А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П |
| 192 | 193 | 194 | 195 | 196 | 197 | 198 | 199 | 200 | 201 | 202 | 203 | 204 | 205 | 206 | 207 |
| Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я |
| 208 | 209 | 210 | 211 | 212 | 213 | 214 | 215 | 216 | 217 | 218 | 219 | 220 | 221 | 222 | 223 |
| а | б | в | г | д | е | ж | з | и | й | к | л | м | н | о | п |
| 224 | 225 | 226 | 227 | 228 | 229 | 230 | 231 | 232 | 233 | 234 | 235 | 236 | 237 | 238 | 239 |
| р | с | т | у | ф | х | ц | ч | ш | щ | ъ | ы | ь | э | ю | я |
| 240 | 241 | 242 | 243 | 244 | 245 | 246 | 247 | 248 | 249 | 250 | 251 | 252 | 253 | 254 | 255 |

Кодовые страницы

КОИ-8 (код обмена информацией) – восьмибитовая кодовая страница (исторически первая для кириллических символов).

Пришла из Unix, широко применялась в электронной почте.

| | | | | | | | | | | | | | | | |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| — | | Г | Г | Л | Л | Т | Т | Т | Т | Т | ■ | ■ | ■ | ■ | ■ |
| 128 | 129 | 130 | 131 | 132 | 133 | 134 | 135 | 136 | 137 | 138 | 139 | 140 | 141 | 142 | 143 |
| ▒ | ▒ | ▒ | Г | ■ | ● | √ | ≈ | ≤ | ≥ | nbsp | Ј | ◦ | ² | • | ÷ |
| 144 | 145 | 146 | 147 | 148 | 149 | 150 | 151 | 152 | 153 | 154 | 155 | 156 | 157 | 158 | 159 |
| = | | F | ё | П | Г | Г | П | Г | Е | Ц | Ц | Ј | Л | Л | Т |
| 160 | 161 | 162 | 163 | 164 | 165 | 166 | 167 | 168 | 169 | 170 | 171 | 172 | 173 | 174 | 175 |
| | | Г | Ё | | | Т | П | Т | Ц | Ц | Т | | | | © |
| 176 | 177 | 178 | 179 | 180 | 181 | 182 | 183 | 184 | 185 | 186 | 187 | 188 | 189 | 190 | 191 |
| ю | а | б | ц | д | е | ф | г | х | и | й | к | л | м | н | о |
| 192 | 193 | 194 | 195 | 196 | 197 | 198 | 199 | 200 | 201 | 202 | 203 | 204 | 205 | 206 | 207 |
| п | я | р | с | т | у | ж | в | ь | ы | з | ш | э | щ | ч | ъ |
| 208 | 209 | 210 | 211 | 212 | 213 | 214 | 215 | 216 | 217 | 218 | 219 | 220 | 221 | 222 | 223 |
| Ю | А | Б | Ц | Д | Е | Ф | Г | Х | И | Й | К | Л | М | Н | О |
| 224 | 225 | 226 | 227 | 228 | 229 | 230 | 231 | 232 | 233 | 234 | 235 | 236 | 237 | 238 | 239 |
| П | Я | Р | С | Т | У | Ж | В | Ь | Ы | З | Ш | Э | Щ | Ч | Ъ |
| 240 | 241 | 242 | 243 | 244 | 245 | 246 | 247 | 248 | 249 | 250 | 251 | 252 | 253 | 254 | 255 |

Кодовые страницы

КОИ-8 (код обмена информацией) – восьмибитовая кодовая страница (исторически первая для кириллических символов).

Пришла из Unix, широко применялась в электронной почте.

| | | | | | | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|-----|-----|-----|-----|-----|
| — | | Г | Г | Г | Г | Г | Г | Г | Г | Г | ■ | ■ | ■ | ■ | ■ |
| 128 | 129 | 130 | 131 | 132 | 133 | 134 | 135 | 136 | 137 | 138 | 139 | 140 | 141 | 142 | 143 |
| ▒ | ▒ | ▒ | Г | ■ | ● | √ | ≈ | ≤ | ≥ | nbsp | Ј | ◦ | ² | • | ÷ |
| 144 | 145 | 146 | 147 | 148 | 149 | 150 | 151 | 152 | 153 | 154 | 155 | 156 | 157 | 158 | 159 |
| = | | ƒ | ё | п | Г | Г | п | Г | Г | Г | Г | Г | Г | Г | Г |
| 160 | 161 | 162 | 163 | 164 | 165 | 166 | 167 | 168 | 169 | 170 | 171 | 172 | 173 | 174 | 175 |
| Г | Г | Г | Ё | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | © |
| 176 | 177 | 178 | 179 | 180 | 181 | 182 | 183 | 184 | 185 | 186 | 187 | 188 | 189 | 190 | 191 |
| ю | а | б | ц | д | е | ф | г | х | и | й | к | л | м | н | о |
| 192 | 193 | 194 | 195 | 196 | 197 | 198 | 199 | 200 | 201 | 202 | 203 | 204 | 205 | 206 | 207 |
| п | я | р | с | т | у | ж | в | ь | ы | з | ш | э | щ | ч | ъ |
| 208 | 209 | 210 | 211 | 212 | 213 | 214 | 215 | 216 | 217 | 218 | 219 | 220 | 221 | 222 | 223 |
| Ю | А | Б | Ц | Д | Е | Ф | Г | Х | И | Й | К | Л | М | Н | О |
| 224 | 225 | 226 | 227 | 228 | 229 | 230 | 231 | 232 | 233 | 234 | 235 | 236 | 237 | 238 | 239 |
| П | Я | Р | С | Т | У | Ж | В | Ь | Ы | З | Ш | Э | Щ | Ч | Ъ |
| 240 | 241 | 242 | 243 | 244 | 245 | 246 | 247 | 248 | 249 | 250 | 251 | 252 | 253 | 254 | 255 |

Задача:

1. Записать в КОИ-8 свое имя и фамилию. Прочитать строку с помощью 7-битного “приложения”.

Символьные строки

С-строки (ASCIIZ-строки)

Строка = массив символов, концом строки считается символ с ASCII-кодом 0.

“abc” = [97 98 99 0]

Pascal-строки

Строка = массив символов, длина определяется счетчиком в первом элементе этого массива (1 байт, максимальная длина строки – 255 символов).

“abc” = [3 97 98 99]

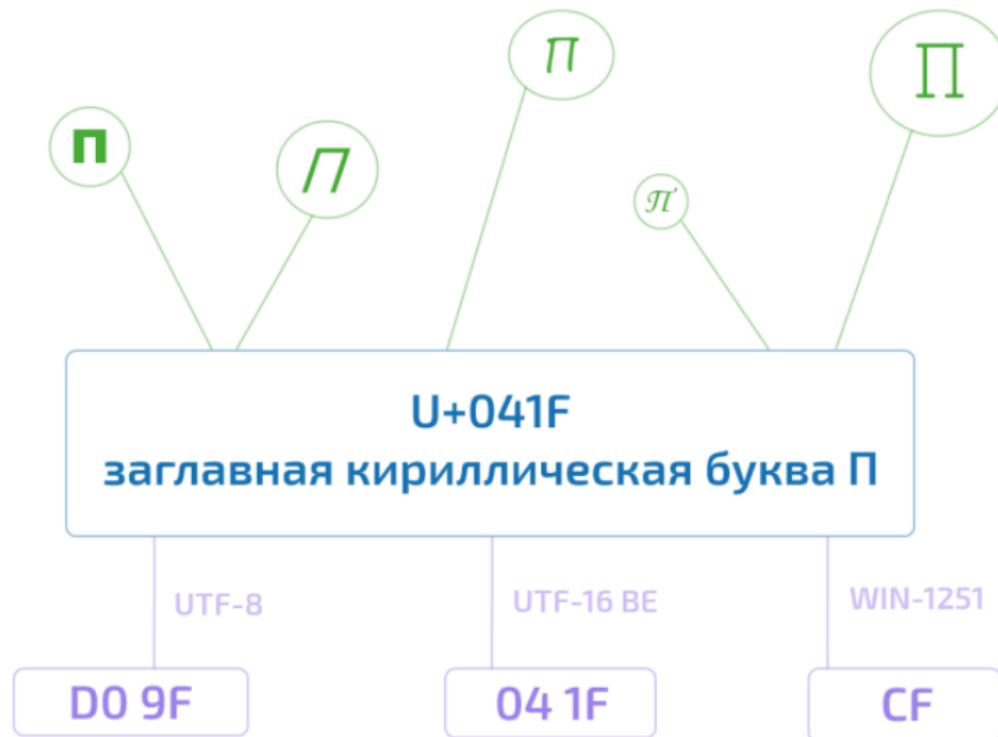
Недостатки однобайтовых кодировок

- Одновременно работать можно лишь с 256 символами (первые 128 зарезервированы).
- Шрифты привязаны к конкретной кодировке.
- Каждая кодировка представляет свой набор символов и конвертация из одной в другую возможна только с частичными потерями (отсутствующие символы заменяются на графически похожие).
- Перенос файлов между устройствами под управлением разных операционных систем затруднителен. Нужно либо иметь программу-конвертер, либо таскать вместе с файлом дополнительные шрифты.
- В мире существуют неалфавитные системы письма (иероглифическая письменность), которые в однобайтной кодировке непредставимы в принципе.

Unicode

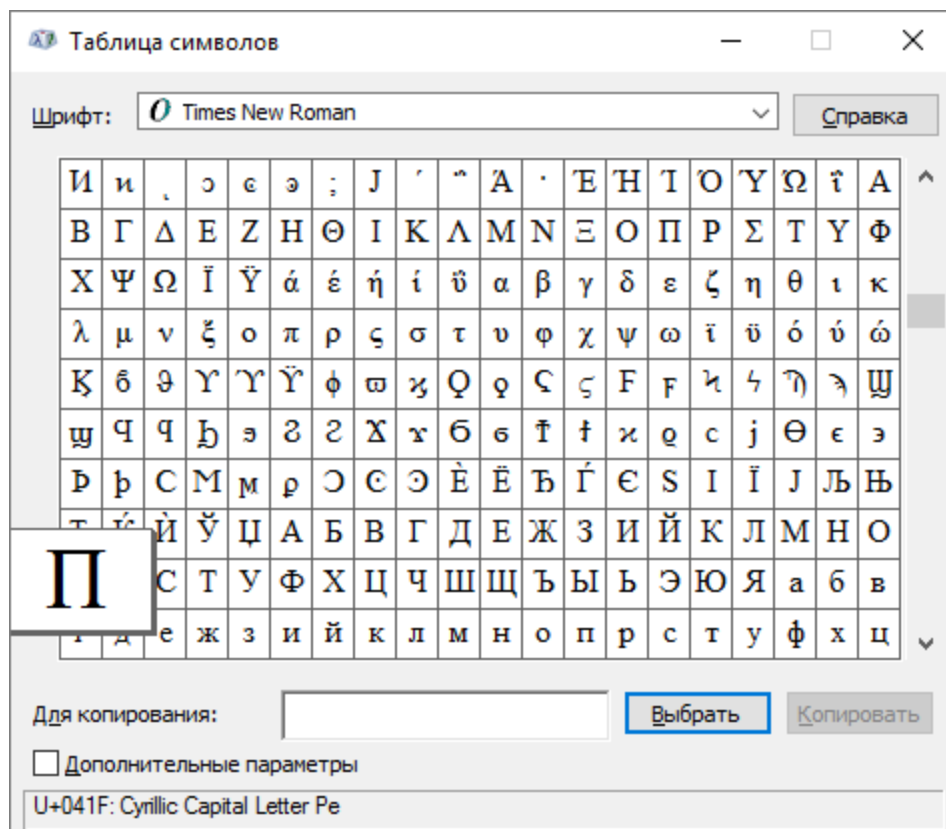
Чёткое разграничение между символами, их представлением в компьютере и их отображением на устройстве вывода.

1. В стандарте определяются абстрактные Unicode-символы.
 - Каждому символу сопоставляется целое число (кодовая позиция, code point). Зарезервировано 1 114 112 кодовых позиций.



Unicode

- Сайт <http://www.unicode.org/charts/>
- Утилита charmap



Unicode

Hello -> U+0048 U+0065 U+006C U+006C U+006F

Как хранить в памяти компьютера?

Unicode

Hello -> U+0048 U+0065 U+006C U+006C U+006F

Как хранить в памяти компьютера?

1. Два байта на каждый символ (65535 символов).

UCS-2 (Universal Character Set, два байта на символ).

Hello -> 00 48 00 65 00 6C 00 6C 00 6F

Hello -> 48 00 65 00 6C 00 6C 00 6F 00

Unicode

Hello -> U+0048 U+0065 U+006C U+006C U+006F

Как хранить в памяти компьютера?

1. Два байта на каждый символ (65535 символов).

UCS-2 (Universal Character Set, два байта на символ).

Hello -> 00 48 00 65 00 6C 00 6C 00 6F

Hello -> 48 00 65 00 6C 00 6C 00 6F 00

2. Мультибайтовая кодировка UTF-8

Unicode Transformation Format, 8 бит

1 байт – 0xxxxxxx

2 байта – 110xxxxx 10xxxxxx

3 байта – 1110xxxx 10xxxxxx 10xxxxxx

4 байта – 11110xxx 10xxxxxx 10xxxxxx 10xxxxxx

Представление первых 127 символов в Unicode совпадает с представлением символов ASCII.

Hello -> 48 65 6C 6C 6F