1. **История развития вычислительной техники до появления ЭВМ**

1623 г. – В. Шиккард, профессор Тюбингенского университета предложил агрегат, состоящий из суммирующего и множительного устройства.

1642 г. – Блез Паскаль продемонстрировал в Люксембургском дворце машину, которая могла складывать и вычитать.

1673 г. – немецкий математик и философ Г. Лейбниц представил в Парижской академии вычислитель, выполняющий все 4 арифметических действия.

* 1812–1823 гг. – профессор Кембриджского университета Чарльз Беббидж построил разностную машину, а в 1835 г. он же представил проект аналитической машины (прообраз ПК): склад, мельница, управляющий. Фрагмент такого вычислителя построил сын ученого, а программы для него готовила первый программист Ада Лавлейс (Байрон).
* 1880 г. – Г. Холлерит сконструировал электромеханический перфокарточный табулятор, который использовался при переписи населения в США и в России.
* 1911 г. – механик А.Н. Крылов построил уникальный аналоговый решатель дифференциальных уравнений.
* 1918 г. – М.А. Бонч-Бруевич изобрел триггер.
* 1919 г. – академик Н.Н. Павловский создал аналоговую вычислительную машину.
* 1928 г. – основана фирма Motorola для производства электронных узлов вычислителей.
* 1936 г. – английский математик А. Тьюринг опубликовал доказательство того, что любой алгоритм может быть реализован с помощью дискретного автомата.
* 1939 г. – американский инженер Дж. Стибниц создал релейную машину BELL.
* 1939 г. – У. Хьюлетт и Д. Паккард основали компанию для производства компонентов первых вычислителей.
* 1946 г. – Джон фон Нейман и Гольдстейн опубликовали статью «Предварительное обсуждение логической конструкции ЭВМ».
* 1947 г. – Шокли и др. изобрели транзистор.
* 1948 г. – в Массачусетском университете был построен первый компьютер с памятью EDVAC.
* 1949 г. – Морис Уилкс построил компьютер EDSAC в соответствии с принципами фон Неймана.
* 1957 г. – Б. Нойс и Г. Мур открыли первую в мире компанию по производству полупроводниковых приборов, спустя 10 лет ими была создана фирма «Intel Corporation».
* 1971 г. – сотрудник фирмы Intel Д. Хофф создает первый микропроцессор i4004.
* 1975 г. – рождение корпорации «Microsoft Corporation».
* 1977 г. – Стефен Возняк и Стивен Джобс собирают первый настольный компьютер «Apple».

1981 г. – IBM представляет свой первый персональный компьютер IBM PC.

1. **Поколения ЭВМ, описание, краткая характеристика**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Характеристики** | **Первое 1951–1954 гг.** | **Второе 1958–1960 гг.** | **Третье 1965–1966 гг.** | **Четвертое** | **Пятое** |
| **1976–1979 гг.** | **1985 г.** |
| 1. Элементная база ЦП | Электронные лампы | Транзисторы | Интегральные схемы | БИС | СБИС | СБИС + опто – и крио – электроника |
| 2. Элементная база ОЗУ | Электронно-лучевые трубки | Ферритовые сердечники | Ферритовые сердечники | БИС | СБИС | СБИС |
| 3. Maксимальная емкость ОП в байтах | 102 | 103 | 104 | 105 | 107 | 108 |
| 4. Максимальное быстродействие ЦП в ОС | 104 | 106 | 107 | 108 | 109 + многопроцессорность | 1012 + многопроц |
| 5. Языки программиро-вания | Машинный код | + ассемблер | + процедурные языки высокого уровня (ЯВУ) | + новые процедурные ЯВУ | + непроцедурные ЯВУ | + новые непроцедурные ЯВУ |
| 6. Средства связи пользователя с ЭВМ | Пульт управления, перфокарты | Перфокарты, перфоленты | Алфавитно-цифровой терминал | Монохромный графический дисплей, клавиатура | Цветной графический дисплей, клавиатура, «мышь» и т.д. | + устройства голосовой связи с ЭВМ |

1. **Принципы фон Неймана архитектуры построения ЭВМ**
* наличие единого вычислительного устройства, включающего процессор, средства передачи информации и память;
* линейная структура адресации памяти, состоящей из слов фиксированной длины;
* двоичная система исчисления;
* централизованное последовательное управление;
* хранимая программа;
* низкий уровень машинного языка;
* наличие команд условной и безусловной передачи управления;
* АЛУ с представлением чисел в форме с плавающей точкой.
1. **Системы счисления. Функции, разновидности, перевод целых и дробных чисел из одной системы счисления в другую**

**Cистема счисления –** способ представления любого числа с помощью некоторого алфавита символов, называемых цифрами.

*Позиционная* система счисления – количественное значение каждой цифры зависит от ее места (позиции) в числе.

*Непозиционная* система счисления – цифры не меняют своего количественного значения при изменении их расположения в числе.

Целое число с основанием P1 переводится в систему счисления с основанием P2 путем последовательного деления числа Ap1 на основание P2*,* записанного в виде числа с основанием P1*,* до получения остатка.

Полученное частное следует вновь делить на основание P2 и этот процесс надо повторять до тех пор, пока частное не станет меньше делителя.

Полученные остатки от деления и последнее частное записываются в порядке, обратном полученному при делении.

Сформированное число и будет являться числом с основанием P2

Дробное число с основанием P1 переводится в систему счисления с основанием P2 путем последовательного умножения Ap1 на основание P2 записанное в виде числа с основанием P1.

При каждом умножении целая часть произведения берется в виде очередной цифры соответствующего разряда, а оставшаяся дробная часть принимается за новое множимое.

Число умножений определяет разрядность полученного результата, представляющего число Ap1, в системе счисления P2

1. **Представление информации в ЭВМ. Числовая, текстовая, графическая, видео и звуковая информация**

Представление числовой информации.

В ЭВМ используются три вида чисел:

* с фиксированной точкой,
* с плавающей точкой,
* двоично-десятичное представление.

У чисел с фиксированной точкой – строго определенное место точки – или перед первой значащей цифрой числа (дробное, число по модулю меньше единицы, например 0.101), или после последней значащей цифрой числа (целое число, например 101.0).

Числа с плавающей точкой представляются в виде мантиссы *тa* и порядка *рa,* например число *А10=*373 можно представить в виде 0.373 • 103, при этом *тa=* 0.373, *рa=* 3.

Порядок числа *ра* определяет положение точки в двоичном числе. Например, А2 = (100; 0.101101) – обозначает число *А2=* 1011.01

Двоично-десятичная форма представления двоичных чисел используется при необходимости ввода, вывода и обработки большого количества десятичных данных. Для перевода из десятичной системы в двоичную и наоборот требуется много дополнительных команд.

В двоично-десятичной системе каждая цифра десятичного числа представляется двоичной тетрадой. Например, А10=3759, *А2-10=* 0011 0111 0101 1001.

Значение знака числа отмечается кодом, отличным
от кодов цифр. Например «+» имеет значение тетрады «1100», а «–» – «1101».

Представление символьной информации

При вводе информации с клавиатуры кодирование происходит при нажатии клавиши, на которой изображен требуемый символ, при этом в клавиатуре вырабатывается так называемый scan-код, представляющий собой двоичное число, равное порядковому номеру клавиши.

Опознание символа и присвоение ему внутреннего кода ЭВМ производятся специальной программой по специальным таблицам: КОИ-7, ASCII, Win-1251, ISO, Unicode.

В системе *ASCII* закреплены две таблицы кодирования – *базовая и расширенная.* Базовая таблица закрепляет значения кодов от 0 до 127, а расширенная относится к символам с номерами от 128 до 255.

Первые 32 кода базовой таблицы содержат *управляющие коды.*

Начиная с кода 32 по код 127 размещены коды символов английского алфавита, знаков препинания, цифр, арифметических действий и некоторых вспомогательных символов.

Вторая половина таблицы содержит национальные шрифты, символы псевдографики, из которых могут быть построены таблицы, специальные математические знаки.

В СССР действовала системы кодирования КОИ – 7,8 *(код обмена информацией, семи-, восьмизначный).*

В системах Windows используется кодировка символов русского языка *Windows-1251.*

Система, основанная на 16-разрядном кодировании символов, получила название *универсальной – UNICODE.* Она позволяет обеспечить уникальные коды для 65 536 различных символов.

Представление видеоинформации

Может быть статической или динамической.

Статическая – текст, рисунки, графики, чертежи, таблицы. Рисунки – плоские – двухмерные и объемные – трехмерные.

По способу формирования видеоизображения бывают растровые и векторные.

Растровая графика (а) задается массивом точек, векторная (б) – отрезками линий (с координатами начала, углом наклона и длиной).

1. **Арифметические основы ЭВМ. Машинные коды, операции с ними**

Все современные ЭВМ имеют достаточно развитую систему команд, включающую десятки и сотни машинных операций. Но выполнение любой операции основано на использовании простейших микроопераций типа сложения и сдвиг. Это позволяет иметь единое арифметико-логическое устройство для выполнения любых операций, связанных с обработкой информации.

Машинные коды

Под **знак чисел** отводится специальный знаковый разряд. Знак «+» кодируется двоичным нулем, а знак «–» – единицей.

В ЭВМ все операции выполняются над числами, представленными специальными **машинными кодами**. Их использование позволяет обрабатывать знаковые разряды чисел так же, как и значащие разряды, а также заменять операцию вычитания операцией сложения.

Различают **прямой код** (П), **обратный код** (ОК) и **дополнительный код** (ДК) двоичных чисел.

***Прямой код*** двоичного числа образуется из абсолютного значения этого числа и кода знака (нуль или единица) перед его старшим числовым разрядом.

**Пример** 1. А10=+10 *А2 =*+1010 [A2]п*=* 0|1010

B10=-15 B*2 =* -1111 [B2]п*=* 1|1111

Вертикальной линией здесь отмечена условная граница, отделяющая знаковый разряд от значащих.

***Обратный код*** двоичного числа образуется по следующему правилу.

Обратный код положительных чисел совпадает с их прямым кодом.

Обратный код отрицательного числа содержит единицу в знаковом разряде числа, а значащие разряды числа заменяются на инверсные, т.е. нули заменяются единицами, а единицы – нулями.

***Дополнительный код*** положительных чисел совпадает с их прямым кодом. Дополнительный код отрицательного числа представляет собой результат суммирования обратного кода числа с единицей младшего разряда (2° – для целых чисел, 2-к – для дробных).

***Сложение (вычитание).*** Операция вычитания приводится к операции сложения путем преобразования чисел в обратный или дополнительный код. Пусть числа А>=О и В>=О, тогда операция алгебраического сложения выполняется в соответствии с табл.

***Умножение.*** Умножение двоичных чисел наиболее просто реализуется в прямом коде. Рассмотрим, каким образом оно приводится к операциям сложения и сдвигам.

Операция деления, как и в десятичной арифметике, является обратной операции умножения. Покажем, что и эта операция приводится к последовательности операций сложения и сдвига.

1. **Алгебра логики. Булевы функции, способы задания**

Алгебра логики – устанавливает основные законы формирования и преобразования логических функций. Она позволяет представить любую сложную функцию в виде композиции простейших функций.

Существует несколько синонимов по отношению к *функциям алгебры* логики:

* *функции алгебры* логики (ФАЛ);
* переключательные *функции*;
* булевские *функции*;
* двоичные *функции*.

Вся информация в ЭВМ представляется в двоичной системе счисления. Поставим в соответствие входным сигналам отдельных устройств ЭВМ соответствующие значения *хi,* (i=1, n), а выходным сигналам – значения функций *уj (j=1, m)*

Зависимости

yj =f(x1, x1,…..xn),

где *хi* - i-й вход; n *–* число входов; *уj – j* – й выход; m – число выходов в устройстве, описывают алгоритм работы любого устройства ЭВМ.

Каждая такая зависимость y*j,* является «**булевой функцией» (функцией алгебры логики)** – число возможных состояний её и каждой её независимой переменной равно двум, а её аргументы определены на множестве {0,1}.

Способы представления ФАЛ

**Словесный**

При этом способе словесное описание однозначно определяет все случаи, при которых функция принимает значения 0 или 1. Например, многовходовая функция ИЛИ может иметь такое словесное описание: функция принимает значение 1, если хотя бы один из аргументов принимает значение 1, иначе – 0.

**Числовой**

Функция задается в виде десятичных (или восьмеричных, или шестнадцатиричных) эквивалентов номеров тех наборов аргументов, на которых функция принимает значение 1.

Условие, что функция f (x1, x2, x3) = 1 на наборах 1,3,5,6,7 записывается f (1, 3, 5, 6, 7) = 1.

Аналогичным образом булева функция может быть задана по нулевым значениям.

При нумерации наборов переменным x1, x2, x3 ставится в соответствие веса 22, 21, 20, т.е. 6 набору соответствует двоичный эквивалент 110, а 1 набору – 001.

**Табличный**

Функция задается в виде *таблицы истинности* (соответствия), которая содержит 2n строк (по числу наборов аргументов), n столбцов по числу переменных и один столбец значений функции. В такой таблице каждому набору аргументов соответствует значение функции.

**Аналитический**

Функция задается в виде алгебраического выражения, получаемого путем применения каких-либо логических операций к переменным алгебры логики. применяя операции конъюнкции и дизъюнкции можно задать функцию выражением f (x1, x2, x3) = x1x2 v x3.

**Координатный**

При этом способе задания таблица истинности функции представляется в виде координатной карты состояний, которая часто называется *картой Карно*. Такая карта содержит 2n клеток по числу наборов всевозможных значений n переменных функции. Переменные функции разбиваются на две группы так, что одна группа определяет координаты столбца, а другая – координаты строки.

При такoм способе построения клетка определяется координатами переменных, соответствующих определенному двоичному набору.

Внутри клетки карты Карно ставится значение функции на данном наборе.

Переменные в строках и столбцах располагаются так, чтобы соседние клетки карты Карно различались только в одном разряде переменных, т.е. были соседними.

Такой способ представления очень удобен для наглядности при минимизации булевых функций.

**Диаграмный**

Является способом представления функционирования схемы, реализующей булеву функцию, во времени. Изображается в виде системы графиков, у которых ось Х соответствует автоматному времени (моментам времени), а ось Y соответствует напряжению дискретных уровней сигналов «логический 0» (0,4 в) и «логическая 1» (2,4 в).

**Графический**

Функция задается в виде n-мерного единичного куба, *вершинам* которого соответствуют наборы значений аргументов и приписаны значения функции на этих наборах. Куб назван единичным, так как каждое ребро соединяет вершины, наборы которых различаются только по одной переменной, т.е. являются *соседними*.

Такой способ задания булевых функций иногда называют геометрическим, но чаще всего *кубическим*. Кубическое представление наиболее пригодно для машинных методов анализа булевых функций, так как позволяет компактно представлять булевы функции от большого количества переменных.