

## 1. Поколение ЭВМ с точки зрения инженера-электронщика

Поколения ЭВМ. Однако, изобретение в 1948 г. транзистора и замена ламповых компонент ЭВМ полупроводниковыми и их дальнейшая микроминиатюризация позволили более интенсивно развиваться средствам вычислительной техники. Таким образом, развитие вычислительной техники неразрывно связано с развитием электронной техники, являющейся элементной базой ЭВМ. Элементная база определяет периодизацию развития ЭВМ по поколениям.

I поколение (1950-1958 г.г.) – на электронных лампах, навесной монтаж;

II поколение (1959-1967 г.г.) – на транзисторах, печатный монтаж;

III поколение (1968-1978 г.г.) – на микросхемах малой степени интеграции, многослойный печатный монтаж;

IV поколение (1979-1993 г.г.) – на микросхемах большой степени интеграции, кристалл, материнская плата;

V поколение (1994 г.-настоящее время) – на микросхемах сверх большой степени интеграции.

Для ЭВМ V-ого поколения характерно не только совершенствование элементной базы, но и совершенствование архитектуры системы, в первую очередь связанную с использованием методов параллельной, в частности, конвейерной обработки информации. Они также обладают возможностью взаимодействия с ЭВМ при помощи естественного языка: человеческой речи и графических изображений, способностью системы обучаться, производить смысловую обработку информации, делать логические суждения. Следует заметить, что последние функции ЭВМ V-ого поколения связаны с совершенствованием программного обеспечения, а непрерывный рост степени интеграции микросхем имеет предел. Поэтому по элементной базе четкой границы между ЭВМ IV-ого и V-ого нет. Очевидно, это произойдет тогда, когда переключаемый элемент – транзистор заменится на более совершенную логическую обработку на молекулярном уровне и логическая электроника перерастет в функциональную.

## 2. Киев и первые ЭВМ. Киев-«кибернетическая Мекка»

В Киеве 1-ая ЭВМ (Лебедев. 1 на континенте Европа) 1947-1951-МЭСМ. Она имела арифметическое устройство, которое выполняло 50 арифметических или логических операций в секунду. МЭСМ была одной из первых в мире ЭВМ с параллельной обработкой кодов. В 1966 году была сконструирована ЦВМ БЭСМ-6 под руководством Лебедева. До 1973 года она являлась наиболее производительной машиной в СССР. Под руководством Глушкова были разработаны ЦВМ «Мир» и «Мир-2» в 60-х годах, в которых впервые было реализовано ступенчатое микропрограммирование. С момента создания первых ЭВМ их элементная база менялась 5 раз.

Вначале 60-х был создан институт кибернетики (директор – Глушков)

## 3. Система программирования вычислительной системы

Система программирования — совокупность мови програмування і комплексу програм, які забезпечують реалізацію даної мови програмування засобами машинної мови кокретної ЕОМ. У цей програмний комплекс входять: транслятор, бібліотека стандартних програм, компонувальник (редактор зв'язків), наладжувальник та інші засоби.

## 4. Типы вычислительных процессов

Вся совокупность вич.процесов используемых для решения различных задач(мат.,науч.-техн.,и других) на евм может разделена на 3 группы – линейная,ветвящийся,циклическая.

А)Лин . про – вич.процес в котором сопост. Этапы вичисления выпол.в лин.последовательность их записи.на блок схеме представляется в виде блоков размещааемых сверху вниз в порядке их выполняемости. для них характерно то что направление вичисления не зависит от исх.данных .

Б)ветвящ – процес,в котором в зависимости от исходных условий иили промеж.результатов он реализується поодному из нескольких зарание предусмотренных направлений.каждое отдельное направление называют ветвью вичисления.выоор той или иной ветви осущ.проверкой выполнения логич условия определяющего св-ва исх данных или промежуточных результатов.в каждом конкретном случае процес реализується только по одной ветви а выполнения остальных исключаются.

В)циклич – при реализации алгоритмов решение многих задач наблюдается многократное повторение отдельных этапов их вич.процеса.Многократное повторение наз.циклом,а вичисл.процес содержащий эти этапы – циклическим.

## 5. Функциональная схема ЭВМ(Фоннеймановская машина)

Функциональная схема ЭВМ(Фоннеймановская машина)

Архитектура ЭВМ включает в себя как структуру, отражающую состав ПК, так и программно – математическое обеспечение.

Структура ЭВМ - совокупность элементов и связей между ними. Основным принципом построения всех современных ЭВМ является программное управление.

Основы учения об архитектуре вычислительных машин были заложены Джон фон Нейманом. Совокупность этих принципов породила классическую (фон-неймановскую) архитектуру ЭВМ.

Фон Нейман не только выдвинул основополагающие принципы логического устройства ЭВМ, но и предложил ее структуру, представленную на рисунке.



Положения фон Неймана:

- Компьютер состоит из нескольких основных устройств (арифметико-логическое устройство, управляющее устройство, память, внешняя память, устройства ввода и вывода)
- Арифметико-логическое устройство – выполняет логические и арифметические действия, необходимые для переработки информации, хранящейся в памяти
- Управляющее устройство – обеспечивает управление и контроль всех устройств компьютера (управляющие сигналы указаны пунктирными стрелками)
- Данные, которые хранятся в запоминающем устройстве, представлены в двоичной форме

- Программа, которая задает работу компьютера, и данные хранятся в одном и том же запоминающем устройстве
- Для ввода и вывода информации используются устройства ввода и вывода

## 6. Принцип хранения программы (принцип программного управления)

Цифровой электронной вычислительной машиной (ЭВМ) называется комплекс технических средств, имеющих общее управление и предназначенное для преобразования информации по заданному алгоритму.

В ЭВМ автоматизирован процесс преобразования информации за счет использования принципа программного управления, сущность которого состоит в том, что управление процессом обработки данных осуществляется на основе информации, заданной извне. Основные аспекты принципов программного управления сформулированы американским ученым венгерского происхождения Фон-Нейманом в 1945-ом году и заключаются в следующем:

1.Принцип управления извне. Управление работой ЭВМ осуществляется программой, заданной извне. Программа состоит из команд, каждая команда осуществляет единый акт преобразования информации и поступают в ЭВМ в строго

определенной последовательности, определяемой программой. Все разновидности команд конкретной ЭВМ составляет систему команд этой ЭВМ.

2. Принцип условного перехода. Это возможность перехода в процессе вычислений на тот или иной участок программы в зависимости от промежуточных результатов вычислений. Этот принцип обеспечивает механизм ветвления алгоритма программы.

3. Принцип хранения программы. Команды как и данные представляются в единой форме и хранятся в одной памяти. При этом исполняются различные способы интерпретации хранимой информации. Следует заметить, что хранение команд и данных в одной памяти соответствует так называемой Пенсильванской архитектуре ЭВМ, в отличие от Гарвардской архитектуры, где команды и данные хранятся в различных хранилищах.

4. Принцип использования двоичной системы счисления. Для представления информации в ЭВМ использование двоичной системы счисления объясняется простотой и высокой надежностью элементов, имеющих два устойчивых состояния, а также хорошо разработанным аппаратом Булевой алгебры, который, в частности, позволяет проводить синтез двоичных устройств преобразования информации.

5. Принцип иерархии памяти. В соответствии с этим принципом вся память ЭВМ делится на внутреннюю и внешнюю. Непосредственно вычислительный процесс протекает с внутренней памятью, имеющую меньшую емкость, но более высокое быстродействие. Взаимодействие между внешней памятью, имеющей большую емкость, но меньшее быстродействие, и внутренней памятью происходит в процессе программного управления. Принцип иерархии памяти является компромиссом между емкостью и быстрым доступом к памяти. Размещение информации во внутренней и внешней памяти осуществляется по адресам.

Эти принципы дополняются для современных ЭВМ следующими положениями:

1. принципом иерархии построения ЭВМ в целом;
2. принципом пространственно-временного преобразования обрабатываемой информации;
3. принципом не нарушения причинно-следственных связей, установленных в ЭВМ.

-Первое положение будет рассмотрено ниже.

-Второе положение заключается в том, что, чем больше тратится аппаратных затрат, размещаемых в пространстве, тем меньше требуется времени для обработки данных, и наоборот, чем меньше оборудования, тем больше требуется времени для обработки данных. Это положение включает в себя принцип временного разделения, характерного для построения многих устройств ЭВМ.

-Третье положение заключается в том, что в ЭВМ устанавливается определенная цепочка причинно-следственных связей, нарушение которых приводит к неработоспособности ЭВМ.

#### 7. Алгоритм. Его свойства. Способы изображения

Алгоритм — чітко визначена послідовність дій, виконуючи які, Обробник інформації (людина або автомат) реалізує процес перетворення даних, з метою одержання кінцевого результату з початкових даних (аргументів).

Властивості алгоритму:

- детермінованість — кожна дія алгоритму повинна сприйматися Обробником інформації однозначно;
- дискретність (покроковість) — алгоритм складається з дій, які у змозі виконати Обробник інформації;
- масовість — алгоритм конструється для розв'язання не однієї конкретної задачі, а цілого класу задач даного типу;
- результативність (цілеспрямованість, скінченність) — алгоритм повинен подати результат за скінчену кількість кроків.

Способи записи:-Словесний (на естественном языке)-Форменно-словесный (например с использованием формул)-

Графический-С помощью языка программирования.

#### 8. Прикладное ПО вычислительных систем

Прикладне програмне забезпечення — сукупність програм, призначених для розв'язування певних цільових задач або класу таких задач. Нині ринок пропонує широке розмаїття програмних продуктів, що автоматизують різноманітні види (сфери) людської діяльності. Серед них:

- текстові редактори;
- бази даних;
- інформаційно-пошукові системи;
- електронні таблиці;
- системи програмованого навчання;
- системи штучного інтелекту;
- математичні програми;
- програми для моделювання;
- системи автоматизованого проектування.

#### 9. Эволюция программных модулей в процессе обработки(по Олефиру схема е)

Програма, записана будь-якою мовою програмування, називається початковим модулем (ПМ). Початковий модуль перетворюється відповідним **транслятором** в об'єктний модуль (ОМ) — модуль в проміжному форматі, спільному для всіх трансляторів. Об'єктний модуль ще не призначений для безпосереднього виконання, оскільки крім тексту програми (ще не зовсім готового для виконання на ЕОМ) містить додаткову інформацію про організацію зв'язків між програмними модулями та про налагодження програми на конкретне місце в оперативній пам'яті під час її завантаження. ОМ повинен пройти ще один етап обробки — редагування. На цьому етапі він обробляється системою програмою «Редактор зв'язків». У результаті з'являється програма, готова до завантаження її в ОП з налаштуваннями на конкретні адреси. Така програма називається завантажувальним модулем (ЗМ). ПМ, ОМ та ЗМ є переміщуваними модулями, не пов'язаними з конкретними адресами пам'яті ЕОМ. Після завантаження в ОП спеціальною програмою-завантажником і налаштування на певні адреси модуль стає нереміщуваним (абсолютним).

Об'єктний модуль складається з тексту програми в машинному коді та двох словників — словника зовнішніх імен та словника



## 10. Основные технические характеристики ЭВМ. Ресурсы ЭВМ

1. Быстродействие. Быстродействие определяет способность ЭВМ выполнять определенные типы операций, как правило, регистровые пересылки, за единицу времени. Быстродействие ЭВМ зависит от скорости протекания переходных процессов в элементной базе и задается тактовой частотой внешнего генератора.
2. Производительность. Производительность определяет способность ЭВМ выполнять любые типы операций, так называемые «смеси Гибсона», за единицу времени. Производительность зависит не только от используемой элементной базы, но и от применяемых архитектурных решений ЭВМ.
3. Разрядность машинного слова. Эта характеристика влияет на точность вычислений и диапазон представимых в ЭВМ чисел.
4. Максимально возможная адресация адресного пространства ЭВМ. Эта характеристика определяет максимальный объем оперативной памяти и возможности по ее виртуальному расширению.
5. Количество групп команд и команд в группах, составляющих систему команд ЭВМ. Эта характеристика определяет возможности ЭВМ по применению соответствующих операционных систем и по программированию.
6. Количество способов адресации команд и данных. Эта характеристика определяет возможности ЭВМ по использованию гибких алгоритмов обработки данных.
7. Тип используемого интерфейса (сопряжения) ядра ЭВМ с внешними устройствами.
8. Надежность.
9. Стоимость.
10. Потребляемая мощность.

**Ресурсы ЭВМ** — средства вычислительной системы, которые могут быть выделены процессу обработки данных на определенном интервале времени. К ресурсам ЭВМ традиционно относят объем доступной памяти, процессорное время и др. При разработке принципов архитектуры компьютеров широко используется идея о закреплении разных операций процесса решения задачи (процесса вычислений) за разными специализированными устройствами. К **центральному** (системному) устройствам ПК относятся прежде всего центральный процессор и оперативная память. **Периферийными** устройствами компьютера являются: дисплей, клавиатура, манипуляторы — мышь, джойстик, световое перо и т. п., винчестер, дисководы — для гибких дисков, компакт-дисков и т. п., принтер, плоттер, сканер, модем и пр. Архитектура современных персональных компьютеров основана на **магистрально-модульном** принципе. В соответствии с этим принципом ПК состоит из отдельных заменяемых устройств и эти устройства взаимодействуют между собой (обмениваются информацией) через системную (Информационную) магистраль. **Системная магистраль** предназначена для передачи данных, адресов, команд управления. Центральные устройства подсоединены к шине непосредственно, а периферийные — через устройства сопряжения (контроллеры или адаптеры). Устройства компьютера целесообразно характеризовать по следующим признакам: класс устройств (название), назначение, принципы работы, основные технические характеристики (пользовательские характеристики, потребительские свойства), особенности, программная поддержка. Для того чтобы устройства работали в комплексе, нужны специальные программы управления устройством (для каждого устройства — своя). Такие программы называются **драйверами**. Производительность процессора, емкость памяти, возможности системного обмена, программы общего пользования

ЦП+ОП ↔ | ОШ | ↔ ПУ

ОШ: ШД (данных), ША (адреса), ШУ (управления)

## 11. Способы адресации к данным

(в. Способы адресации к данным)

- Подразумеваемый операнд (операнд фактически задается кодом операции команд)
- **-/-** адрес (в команде не содержится указаний об адресе, но он подразумевается)
- Непосредственная адресация (в команде содержится не адрес, а сам операнд)
- Прямая **-/-** (исполнительный адрес совпадает с адресной частью команды)
- Относительная **-/-** (исполнительный адрес определяется суммой адресного кода команды и базового адреса)
- Укороченная **-/-** (в команде задаются только младшие разряды адресов, старшие подразумеваются нулевыми)
- Регистровая **-/-** (в качестве фиксированных ячеек с короткими адресами используются регистры процессора)
- Косвенная **-/-** (адресный код команды указывает адрес ячейки памяти, в которой находится адрес операнда или команды)
- Автоинкрементная **-/-** (сначала содержимое регистра используется как адрес операнда, а затем получает приращение, равное числу байт в элементе массива)
- Автодекрементная **-/-** (сначала содержимое указанного в команде регистра уменьшается на число, равное числу байт в элементе массива, а затем используется как адрес операнда)
- Адресация слов переменной длины

## 12. План вычислительного эксперимента

1. Постановка задачи (определяются цели исследования; моменты, которыми можно пренебречь)
2. Формализация задачи (построение математической модели)
3. Выбор численного метода решения математической модели (доработка существующего или разработка нового метода).
4. Алгоритмизация численного метода (представление процесса решения данной мат. модели выбранным методом).
5. Запись алгоритма на языке, который воспринимает компьютер - программирование.
6. Отладка программы (поиск и устранение ошибок (дебаг) в программе) Ошибки : синтакс, смысл.
7. Тестирование программы.
8. Решение основной задачи (модели)
9. Анализ полученных результатов.

## 13. Режимы работы ЭВМ. Эволюция ОС

Еволюція ОС тісно зв'язана з режимами доступу та режимами роботи ЕОМ.

1. Режим безпосереднього доступу до ЕОМ, коли користувач сам запускає свою програму і веде з нею діалог у процесі роботи. Використовується в ЕОМ першого та другого покоління.
2. Пакетний режим. Завдання користувачів формуються в пакети, які пропускає оператор ЕОМ.
3. Мультипрограмування (мультизадачність). У пам'яті ЕОМ одночасно знаходяться кілька задач користувачів, готових до виконання. Мультипрограмування забезпечує найвищу продуктивність обладнання ЕОМ.
4. Режим розподілу часу. Є два типи систем розподілу часу: мультидоступ та реальний час.

Покоління ОС:

0) Операційні системи відсутні. Процес програмування трудомісткий. Використання режиму безпосереднього доступу до ресурсів ЕОМ. (1945-1955)

1) Використання алгоритмічних мов програмування, трансляторів, бібліотек, стандартних програм, створення моніторних систем для керування проходженням завдань. Впровадження пакетного режиму ЕОМ. (1955-1965)

2)Родина програмно-сумісних ЕОМ різної потужності з розвинутою системою переривань та розвиненими операційними системами. Впровадження режимів роботи ЕОМ:1)мультипрограмного; 2)розподілу часу; 3)реального часу.(1965-1980)  
3)ЕОМ, які забезпечують персональне та комунальне використання обчислювальних потужностей, об'єднаних в єдину мережу.(1980-2008)

#### 14. Классификация ЕВМ по принципу действия. Сравнение различных типов ЕВМ

1)АВМ 2)ЦВМ 3)ГВМ (АЦВМ)

В АВМ перерабатываемая информация представляется в непрерывной форме в виде измененных во времени физических величин – электрических напряжений и токов.

В ЦВМ – обработке подвергаются цифровые (дискретные) коды математических величин.

В ГВМ – присутствие информации в аналоговом и цифровом виде.

##### Сравнительные характеристики АВМ и ЦВМ:

Достоинство АВМ:

- 1)высокое быстродействие;
- 2)простота конструкции;
- 3)наглядность и простота ввода и вывода данных;
- 4)простота в изменении параметров исследуемой задачи;
- 5)возможность работы в различных масштабах времени.

Недостатки АВМ:

- 1)Невысокая точность получения результата;
- 2)Сравнительно узкая область применения.

Достоинство ЦВМ:

- 1)универсальность;
- 2)высокое быстродействие;
- 3)точность вычислений.

Недостаток ЦВМ:

довольно трудоемкий и длительный процесс подготовки задачи для решения.

#### 15. ОС. Архитектура ОС. Функции ОС

ОС — це основа всієї системи програмного забезпечення сучасної ЕОМ, яка визначає операційне середовище, де працюють оператори, програмісти, інженери, адміністратори та власники обчислювальних засобів.

Операційні системи виконують дві головні функції: розширення можливостей ЕОМ та управління її ресурсами.

Основні функції операційних систем:

- діалог із користувачами;
- ініціація та завершення виконання задач користувачів;
- керування ходом їх виконання;
- обробка різних виняткових ситуацій, що виникають в процесі роботи;
- розподіл ресурсів ЕОМ між задачами;
- забезпечення можливості використання наявних програмних та інформаційних засобів загального користування;
- взаємний захист програм та інформації, які належать різним користувачам;
- оптимізація паралельної роботи пристроїв ЕОМ з метою досягнення найвищої продуктивності;
- реєстрація та облік усієї виконуваної роботи

#### 16. Системное ПО вычислительной системы. Структура ПО вычислительной системы

Системне програмне забезпечення (СПЗ) — комплекс програмних засобів, які призначені для підвищення ефективності використання потужностей ЕОМ, полегшення її експлуатації, зниження трудомісткості роботи з проектування та виконання програмних продуктів, надання користувачу ЕОМ різноманітних послуг.

До складу СПЗ ЕОМ входять:

- операційні системи — організує виконання програм та взаємодію користувача з комп'ютером;
  - сервісні системи розширюють можливості ОС, надаючи користувачу та програмам, що виконуються, набори додаткових послуг. З цієї причини іноді сервісні системи (особливо інтерфейсні системи) відносять до складу ОС;
  - інструментальні системи — об'єднання різноманітних системних програмних засобів, які використовуються для розробки програмних продуктів, хоча частина з них може застосовуватися для розв'язування прикладних задач.
  - системи технологічного обслуговування призначені для полегшення тестування обладнання та пошуку несправностей.
- Програмне забезпечення (у широкому значенні) — сукупність програм, стандартних програм (підпрограм), мов програмування, правил та документації, яка необхідна для використання та експлуатації програмних продуктів.

#### 17. Программа для ЕВМ. Команды их разновидности. Структура мкомманд

##### 1.1. программа для ЦВМ. Команды, их разновидности. Структура команды

Программа – описание алгоритма в форме, воспринимаемой ЭВМ. Программа представляет собой совокупность команд, записанных в определенной последовательности, обеспечивающей решение задачи на ЭВМ. Команды выполняются в порядке, соответствующем их расположению в последовательных ячейках памяти. При естественном порядке после выполнения очередной команды выбирается команда, расположенная в следующей по порядку ячейке памяти. Естественный порядок может быть нарушен:

- Командами перехода (безусловного и условного - адрес следующей команды зависит от выполнения условия)
- -//- замещения
- Сменой состояния программы
- Запросами прерывания программы

Структура команды определяется составом, назначением и расположением полей в команде.



# 18. Поколения ЕВМ с т.з. математика-программиста 1945 1965 1985 200000000

Операційні системи відсутні. Процес програмування трудомісткий. Використання режиму безпосереднього доступу до ресурсів ЕОМ.

Використання алгоритмічних мов програмування, трансляторів, бібліотек стандартних програм, створення моніторних систем для керування проходженням завдань. FMS – Fortran Monitor System. IBSYS – операційна система для IBM 7094. «ДУБНА» - ОС для ЕОМ БЕСМ-6. Диспетчер ЕОМ М-222. Впровадження пакетного режиму роботи ЕОМ.

Родини програмно-сумісних ЕОМ різної потужності з розвинутою системою переривань та розвиненими операційними системами. IBM/360, IBM/370, ЕС ЕОМ, СМ ЕОМ, міні-ЕОМ «Електроніка». Впровадження режимів роботи ЕОМ:  
- мультипрограми  
- розподілу часу  
- реального часу  
Операційні системи: SABRE, MCP, CTSS, OS/360, OS/370, MULTICS, UNIX, MINIX, Linux

ЕОМ, які забезпечують персональне та комунальне використання обчислювальних потужностей, об'єднаних в єдину мережу.

Користувачські	Серверні
MS DOS, PC DOS, MS Windows 3.1, MS Windows 95, MS Windows 98, MS Windows Me, MS Windows XP, Linux (Ubuntu, ASP, Lindows), Sun OS, MS Windows Vista	Novell NetWare, Windows NT (3.0, 4.0), Windows 2000 Server, Windows 2003 Server, FreeBSD, NetBSD, OpenBSD, Sun OS Server, Linux (SuSe, Red Hat)

## 19. Концепція ВОС(OSD). Взаємодіяєтьх систем СПД

Базовая эталонная модель взаимодействия открытых систем – абстрактная сетевая модель для коммуникации и разработки сетевых протоколов представляет уровневый подход к сети. Каждый уровень обслуживает свою часть процесса взаимодействия. Благодаря такой структуре совместная работа сетевого оборудования и программного обеспечения становится гораздо проще и прозрачней. В настоящее время основным используемым семейством протоколов является TCP-IP разработка которого не была связана с моделью OSI.

Уровни моделей:

- прикладной (предоставление услуг на уровне конечного пользователя: почта)
  - уровень представления данных (интерпретация и сжатие данных)
  - сеансовый аутентификация и проверка полномочий
  - транспортный (обеспечение корректной сквозной пересылки данных)
  - сетевой (маршрутизация и ведение учета)
  - канальный (передача и прием пакетов, определение аппаратных адресов)
  - Физический(кодирование информации в электрический сигнал)
- 1, 2 – Ethernet, FDDI, Token Ring, PPP, SLIP...
  - 3 – IP, ICMP, IGMP, RIP, ARP, RARP, OSPF...
  - 4 – TCP, UDP...
  - 5, 6, 7 – HTTP, FTP, SMTP...

## 20. Способы повышения производительности ВС

### б. Способы повышения производительности вычислительных систем

Коэффициент эффективности машины представляет собой отношение ее производительности к сумме стоимости самой машины и затрат на ее эксплуатацию за определенный промежуток времени. Производительность ЭВМ общего назначения оценивается по скорости выполнения некоторых смесей команд, формируемых путем анализа частоты исполнения разного вида команд.

К более частным характеристикам ЭВМ относятся: число разрядов в машинном слове, скорость выполнения основных видов команд, емкость оперативной памяти, максимальная скорость передачи информации между ядром ЭВМ и внешним оборудованием.

- Увеличение тактовой частоты
- -//- количества инструкций программного кода выполняемых за один такт процесса
- Увеличение кол-ва процессоров
- Создание вычислительных классов
- Увеличение количества ядер в одном процессоре
- Оптимизация алгоритмов вычисления

## 21. Кодирование символов в вычислительной технике и систем передачи данных

Современные ЭВМ обрабатывают алфавитно-цифровую информацию. Совокупность всех символов, используемых в вычислительной системе, представляет собой ее алфавит. Символу соответствует машинная единица информации – слог. Наибольшее распространение получило представление алфавитно-цифровой информации с помощью 8-разрядных слогов – байтов. С помощью байта можно кодировать 256 различных символов.

В качестве внутреннего кода для представления а-ц символов в памяти машины ЕС ЭВМ применяется двоичный код для обработки информации.

Для упрощения автоматизации обработки данных применяют весовой принцип при кодировании символов. Двоичное число, соответствующее коду символа, называется его весом. При весовом кодировании веса

Кодов цифр последовательно возрастают, а веса кодов букв увеличиваются в алф. порядке ( вес кода Б на 1 больше А)

## 22. Інформація. Форми представлення. Теорема Котельникова

Джерело інформації та створювані ним повідомлення та сигнали розділяють на неперервну та дискретну.

Якщо сигнал (повідомлення) в кінцевому інтервалі амплітуд приймає довільну кількість значень, то він (вона) називається неперервним; якщо кількість значень обмежена, то сигнал (повідомлення) – дискретне.

Неперервні повідомлення відображаються сигналами, відображаючими собою якісь фізичні величини, змінні неперервно та приймаючи нескінченне число значень в деяких діапазонах. Така форма уявлення інформації використовується в аналогових і обчислювальних пристроях.

Дискретні повідомлення складаються з скінченної множ. елементів (скінченного числа значень), які формуються джерелом інформації та називаються літерами, або символами. Літери відображаються сигналами, приймаючими кінцеве число значень. В інформаційній та обчислювальній техніці поняття літери включає в себе символи різних систем письма, цифри, пунктуаційні знаки, математичні символи та інші. Якщо елементам дискретного повідомлення поставленні у відповідність цифри або деякі їх сукупності, та така перестановка інформації називається цифровим (числовим). Кінцевий набір букв складає алфавіт.

Дискретні сигнали як засіб передачі інформації знайшли більш широке застосування, ніж неперервні, так як вони менш схильні впливу перешкод в каналах зв'язку. Можливість передачі неперервних повідомлень за допомогою дискретних була доведена В.А. Котельниковим в 1993:

Якщо функція неперервна та її частковий спектр не містить складаючих частин, перебільшуючих  $F$  гц, то вона цілком визначається сукупністю ординат, відстаючих у часі друг від друга на  $\frac{1}{2} F$  сек.

Одним з найбільших досягнень інженерної думки є те, що може бути дана кількісна оцінка: інформацію можна виміряти.

### 1. Інформація. Форми представлення інформації. Теорема Котельникова

Св-во объектов и явлений (процессов) порождают многообразные состояния, которые посредством отражения передаются от одного объекта к другому и запечатлеваются в его структуре. При этом не сами объекты и явления являются информацией, а их отражения в виде чисел, формул, символов и др. абстрактных хар-ках. Информация может быть отнесена к области абстрактных категорий. Если сигнал (сообщение) в конечном интервале амплитуд принимает произвольное кол-во значений, то он называется непрерывным (сообщения отображаются сигналами, представляющими собой какие-либо физические величины, изменяющиеся непрерывно и принимающие бесконечное число значений в некотором диапазоне); если же кол-во значений ограничено, то сигнал – дискретный (сообщения состоят из конечного множества элементов, формируемых источником информации и называемых буквами, или символами).

Т. Любая несинусоидная непериодическая прямая может быть представлена в виде суммы синусоид, которые имеют различную амплитуду, фазу и частоту.

### 23. Измерение количества информации. Единицы измерения

Кол-во информации зависит от степени неопределенности событий, которая хар-ся их вероятностями.

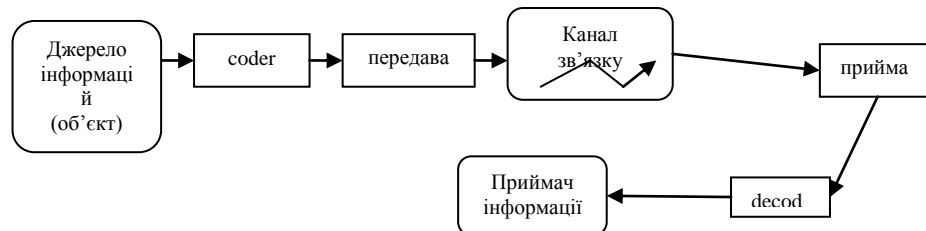
Клод Шеннон:  $I = -(p_1 \cdot \log_2 p_1 + p_2 \cdot \log_2 p_2 + \dots + p_n \cdot \log_2 p_n)$ , где  $I$  – кол-во информации,  $n$  – состояние объекта,  $p$  – вероятность состояния.

Частный случай – Хартли: если объект может находиться в одном из  $n$  состояний и каждое из этих состояний одинаково возможно, то кол-во информации  $I = \log_2 n$ .

Единица информации называется бит – кол-во информации, посредством которого выделяется одно из двух равновероятных состояний объекта. (Если объект может находиться в одном из 8 равновероятных состояний, то  $I = \log_2 8 = 3$  бита; в одном из 16 – 4 бита). Какое кол-во информации содержит один символ в тексте точно ответить трудно (необходимо знать частоту использования и сочетания символов). Если же все символы равновероятны, то кол-во информации в одном символе  $I = \log_2 n$ . Аналогично, если предположить, что цифры 0,1...9 используются одинаково часто, то одна цифра содержит  $I = \log_2 10 = 3.32$  бита информации.

Для измерения больших объемов информации:  $1K=2^{10}$ байт,  $1M=2^{10}K$ байт,  $1Г=2^{10}K$ ,  $1T=2^{10}Г$

## 24. Обобщенная модель системы передачи данных



Приклад: скрипаль в студії → слухач

Первинний перетворювач → мікрофон; через посилювач та модулятор (coder; код-ноти) на передавач. По каналу зв'язку інформація поступає на приймач, після чого за допомогою елементів слуху та емоційного сприйняття потрапляє до адресата.

## 25. Системы счисления применяемые в ЭВМ. Код BSD(бі-сі-ді)

Десятичные цифры	0	1	2	3	4	5	6	7	8	9
Двоично-десятичный код (код BCD) Код 8421	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001
Код 8421+3 Код с избытком 3	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100

## 26. Форма представления чисел в ЦВМ. Естественная форма представления чисел

Разряд двоичного числа представляется в ЦВМ неким техническим устройством (например триггером), двумя различным состоянием которого приспущаются значения 0 и 1.

Существует две формы представления чисел в ЦВМ: с плавающей и фиксированной запятой.  $A = \pm a_n \cdot 10^{m-n}$ . Максимальное число  $2^k - 1$ .

**Фиксированная запятая:** строго зафиксировано кол-во разрядов для целой и дробной части числа; при выборе кол-ва разрядов учитывают нужную точность.

1. запятая фиксирована справа:  $A_{\min}=1$ ,  $A_{\max}=11\dots 1$  (k единиц)  $=2^k-1$

2. запятая фиксирована перед старшим разрядом числа:  $A_{\min}=2^{-k}$ ,  $A_{\max}=0,11\dots 1=1-2^{-k}$ .

**Плавающая запятая:** в разрядную сетку записывают не только само число, но и масштаб этого числа  $A=\pm Mq^{\pm p}$  (знак числа, мантисса, основание с.с., знак порядка, порядок). Порядок указывает расположение запятой в числе.

$A_{\max}=0,11\dots 1$  (m единиц)  $\cdot 2^{+11\dots 1}$  (p единиц)

$A_{\min}=0,1 \cdot 2^{-11\dots 1}$

При представлении числа  $x$  в форме с фиксированной точкой указывается знак числа (sign  $x$ ) и модуль числа (mod  $x$ ). в  $q$ -ичном коде ( $q$  - основание с.с.) вместо точки постоянно для всех чисел и в процессе решения задач не меняется. Знак положительного числа кодируется 0, знак отрицательного – 1. код числа с фиксированной точкой состоящий из кода знака и  $q$ -ичного кода его модуля называется прямым кодом  $q$ -ичного числа. Разряд прямого кода числа в котором располагается код знака называется знаковым разрядом кода. Разряды прямого кода числа в которых располагается  $q$ -ичный код модуля числа называется числовыми разрядами кода. При записи прямого кода одинаковый разряд располагается левее старшего цифрового разряда и обычно отделяется от цифровых разрядов точкой.

## 27. Позиционные системы счисления. Основные характеристики ПСС

Совокупность приемов записи (кодирование чисел) с помощью символов называется системой счисления. Условные знаки используемые для обозначения чисел – цифры. Существует два класса СС:

- позиционные (в которых значения цифр зависят от их места(позиции) в последовательности цифр, изображающих число)
- непозиционные (при изменении места положения цифры в изображении числа, значение цифры не изменяется)

Основные характеристики позиционных систем счисления:

1. основание с.с. ( $q$ ) – это величина показывающая во сколько раз единица соседнего старшего разряда больше единицы младшего разряда. (123: 1 – сто единиц, 2 – двадцать единиц, 3 – три единицы).
2. использование цифр в данной с.с. (в качестве символов могут быть выбраны любые знаки (\*, [, ], (, ), ^...)). На практике нашли применения такие с.с. в которых основание равно кол-ву цифр используемых в данной с.с.. Значение цифр на практике: натуральный ряд цифр и ноль. Значение максимальной цифры –  $q-1$
3. для определения значения числа необходимо цифру каждого разряда умножить на вес этого разряда и суммировать полученное произведение. Таким образом позиционный ряд можно представить:

$$A = a_{n-1} q^{n-1} + a_{n-2} q^{n-2} + \dots + a_0 q^0 + a_1 q^{-1} + \dots + a_m q^{-m}$$

## 28. Формы представления чисел в ЦВМ. Нормальная ф-ма представления

При представлении числа  $x$  с плавающей точкой (в нормальной форме) требуется задать знаки мантиссы и порядка, их модули в  $q$ -ичном коде а также основание системы счисления  $x=\pm m q^p$  (мантисса, основание с.с., порядок). Для задания числа в нормальной форме требуется задать знаки мантиссы и порядка и их модули

## 29. Свойства ПСС

Некоторые св-ва позиционных систем счисления

максимальное число, которое может быть записано в с.с. с основанием  $q$  при использовании  $n$  в целой части числа и  $m$  разрядов в дробной части числа  $A=(q-1)(q^{n-1}+q^{n-2}+\dots+q^{-m})$ . числа такой позиционной с.с. с некоторым дискретным интервалом  $\eta=q^{-m}$ .

$$N=A_{\max}/\eta + 1$$

Число разрядов необходимых для представления  $N$  разрядных чисел в с.с. с основанием  $q$ .

## 30. Переключательные ф-ции 2-х аргументов

1. Логическое сложение двух или нескольких простых высказываний - это функциональная зависимость, в результате которой сложное высказывание  $P$  будет истинно, если хотя бы одно из составляющих его простых высказываний истинно, и ложно, когда одновременно ложны все составляющие его простые высказывания.

Формула логической связи :  $P = x \vee y$  ( $P$  есть  $x$  ИЛИ  $y$ ), где  $\vee$  - знак дизъюнкции– логическое сложение ( $P = x+y$ ).

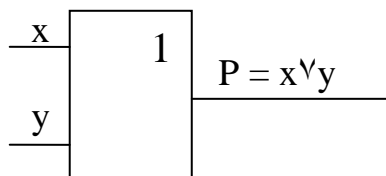
Логика работы элемента дизъюнкции задается таблицей истинности:

x	y	$P = x \vee y$
0	0	0
0	1	1
1	0	1
1	1	1

$F_7(x,y)$

№ функции 7

Схема элемента логического сложения позволяет схемой ИЛИ, дизъюнктом или собирательной схемой. Её функциональное обозначение:



2. Логическое умножение двух или более высказываний заключается в том, что сложное высказывание  $P$  будет истинно в том и только в том случае, когда составляющие его простые высказывания будут одновременно истинны.

Логическое умножение, конъюнкция обозначается знаком конъюнкции  $\wedge$ , или **знаком (and)**, или буквой И. Формула:  $P = x \wedge y$  ( $P$  есть  $x$  И  $y$ ).

Логический элемент И работает в соответствии с таблицей истинности:

x	0	0	1	1
y	0	1	0	1
$P = x \wedge y$	0	0	0	1

№ функции 1  $F_1(x, y)$

и называется конъюнктом, или схемой И, или схемой совпадения.

3. Константа ноль :

x	y	$F_0(x, y)$
0	0	0
0	1	0
1	0	0
1	1	0

$F_0(x, y)$  № функции 0

Логический элемент, реализующий функцию “Константа ноль”, называется “генератор нуля”

4. Константа единица:

x	y	$F_{15}(x, y)$
0	0	1
0	1	1
1	0	1
1	1	1

$F_{15}(x, y)$  № функции 15

Логический элемент реализующий функцию, - “генератор единиц”.

5. Отрицание дизъюнкции ( стрелка Пирса, или функция Вебба) – логическая функция истинна только тогда, когда составляющие её простые высказывания одновременно ложные.

Запись м. б. в одном из видов:

$$F = x \downarrow y; \quad F = x \vee y; \quad F = x + y; \quad F = x \text{ О } y;$$

Таблица истинности:

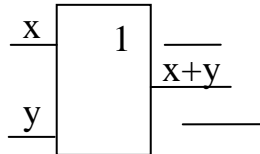


x	y	$F_8(x,y)$
0	0	1
0	1	0
1	0	0
1	1	0

№ функции 8  $F_8(x,y)$

Для реализации операции используется элемент Пирса (ИЛИ-НЕ).

Функциональное изображение элемента ИЛИ-НЕ



6. Отрицание конъюнкции (Штрих Шеффера) – функция истинна, если хотя бы одно из составляющих её высказываний ложно.

Алгебраическая запись операции Шеффера:

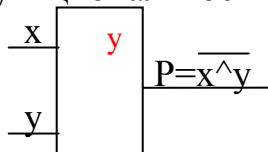
$$F(x,y)=x/y; \quad F=\overline{x \wedge y}; \quad F=\overline{xy}$$

Таблица истинности отображает логику работы:

x	y	$F(x,y)$
0	0	1
0	1	1
1	0	1
1	1	0

№ функции 14  $F_{14}(x,y)$

Для реализации операции применяется элемент Шеффера (элемент И-НЕ) функциональное изображение которого:



7. Эквивалентность (равнозначность) – логическая, операция, в результате которой функция будет истинной, если составляющие её аргументы равноценны или равнозначны.

Обозначение:  $F(x,y)= x \equiv y;$

$$F(x,y)= x \sim y;$$

$$F(x,y)=x \nabla y;$$

Логика работы:

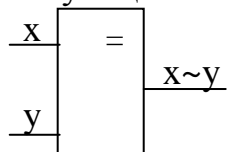
x	y	$F(x,y)$
---	---	----------

0	0	1
0	1	0
1	0	0
1	1	1

№ функции 9  $F_9(x,y)$

Элемент реализующий функцию, называется равнозначность.

Функциональное обозначение:



8. Сумма по модулю 2- операция, в результате которой логическая функция истинна, если составляющие ее аргументы неравнозначны, неравноценны.

Алгебраическая запись:  $F(x,y)=x \oplus y$ ;

Сумма по mod 2 – логическая неравнозначность, т.е.  $F(x,y)=\overline{x \equiv y}$ ;

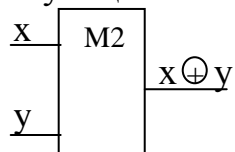
Логика роботы:

x	y	$F_6(x,y)$
0	0	0
0	1	1
1	0	1
1	1	0

№ функции 6  $F_6(x,y)$

Элемент, реализующий операцию, называется “Сложение по mod 2”, или M2.

Функциональное обозначение:



9. Импликацией от  $y$  к  $x$  считается такая логическая операция, в результате которой функция ложна только в том случае, когда величина  $x$  ложна, а  $y$  - истинна.

Функция обозначается:  $F(x,y)=y \rightarrow x$  (ЕСЛИ  $y$ , то  $x$ )

$$F(x,y)=x \vee \bar{y}$$

Таблица истинности:

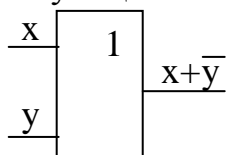
x	y	$F_{11}(x,y)$
0	0	1
0	1	0

1	0	1
1	1	1

№ функции 11  $F_{11}(x,y)$

Операция реализуется элементом импликации.

Функциональное обозначение :



Логическая связь “ЕСЛИ-ТО”.

10. Импликация от x к y – логическая операция, в результате которой функция ложна только в том случае, когда x истинна, а y – ложна. X называется **посылкой**, а Y – следствием.

“Если  $2*2=5$ , то снег черный”. (ЕСЛИ X, то Y)

Функция обозначается:  $F(x,y)=x \rightarrow y$   
(если X, то Y)  
 $F(x,y)= \bar{x} \vee y$

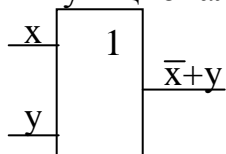
Таблица истинности:

x	y	$F_{13}(x,y)$
0	0	1
0	1	1
1	0	0
1	1	1

№ функции 13  $F_{13}(x,y)$

Операция реализуется элементом импликации.

Функциональное обозначение:



Остальные 6 функций являются либо повторителями входной переменной, либо запретом по одному из входов (отрицание импликации), либо инверсий одного из входов.

$F_2(x,y)$	$x \Delta y$ ;	Запрет по x	Схема запрета
	$\bar{x} \wedge y$ ;	отрицание импликации	
$F_3(x,y)$	x	Переменная x	Повторитель x
$F_4(x,y)$	$y \Delta x$	Запрет по y	Схема запрета
	$x \wedge \bar{y}$ ;	отрицание импликации	
$F_5(x,y)$	y	Переменная y	Повторитель y
$F_{12}(x,y)$	$\bar{x}$	Отрицание x, инверсия x	Инвертор <b>x</b>
$F_{10}(x,y)$	$\bar{y}$	Отрицание y	Инвертор <b>y</b>

## инверсия у

### 31. Булева функция. Набор. Свойства переключательных функций

Логические двоичные функции получили название Булевых по имени английского математика 19-го века Дж. Буля.

Совокупность значений входных переменных (аргументов) называется **набором** и обозначается  $x_1, x_2, \dots, x_n$ , где  $x_i$  равно 0 или 1.

Функция  $f(x_1, x_2, \dots, x_n)$ , определяемая на наборах двоичных аргументов  $x_1, x_2, \dots, x_n$  и принимающая в качестве своих возможных значений 0 или 1, называется **логической**, или **булевой переключательной** функцией.

Для задания булевой функции достаточно построить таблицу её значений, отвечающей всевозможным различным наборам аргументов. Таблица получила название **таблицы истинности**.

Некоторые свойства переключательных функций:

1. Любая переключательная функция  $n$  аргументов определяется на  $2^n$  наборах.

Набор  $n$  аргументов - это двоичное  $n$  разрядное число. Но количество различных  $n$  разрядных чисел равно  $2^n$ .

Пр.:  
 0,0,0,0,0 - нулевой набор;  
 0,0,0,0,1 - первый набор;  
 0,0,0,1,0 - второй набор;  
 0,0,1,0,1 - десятый набор;  
 1,1,1,1,1 - 63-й набор;

Набор аргументов, содержащий все единицы (1,1,1,1,1), называют **единичным** набором.

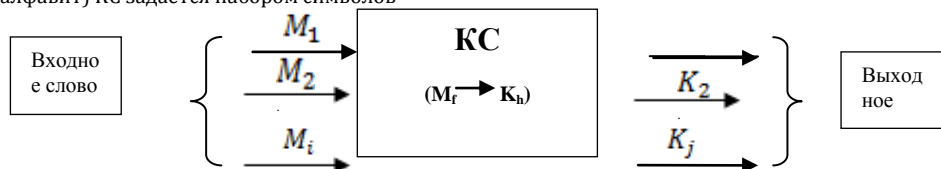
2. Число различных переключательных функций  $n$  аргументов конечно и равно  $2^{2^n}$ .

Переключательная функция  $n$  аргументов определена на  $2^n$  наборах, на которых они принимают значение 0 или 1. Поэтому каждой переключательной функции можно поставить в соответствие  $2^n$  -разрядное двоичное число, а поскольку этих чисел  $2^{2^n}$ , то такое же количество и различных переключательных функций.

### 32. Комбинационные «логические» схемы

**Комбинационная схема** - это устройство, в котором совокупность выходных сигналов в дискретный момент времени  $t$  однозначно определяется набором входных сигналов, поступивших на вход устройства в тот же момент времени  $t$ .

Комбинационную схему можно представить в виде  $m$ -к пелюсника, имеющего  $i$  входов и  $j$  выходов. Входное слово  $M$  (входной алфавит) КС задается набором символов  $M_1, M_2, \dots, M_i$



а выходное слово КС  $K$  (выходной алфавит) задается набором символов  $k_1, k_2, \dots, k_j$ .

КС характеризуется:

- числом входных сигналов;
- числом выходных сигналов;
- логической формулой или таблице истинности.

При изменении набора входных сигналов  $M$  меняется набор выходных сигналов  $K$ . Т.о. выходные сигналы КС полностью определяются входными сигналами и не зависят от внутреннего состояния КС. Поэтому КС называется также **автоматами без памяти**, или **примитивными автоматами**.

### 33. Переключательная функция 1 аргумента

Существует 4 различные переключательные функции одного аргумента.

$x$	0	1	Усл.обозн.	Название функции
$f_0(x)$	0	0	0	const 0
$f_1(x)$	0	1	$x$	Переменная $x$
$f_2(x)$	1	0	$\bar{x}$	Инверсия $x$
$f_3(x)$	1	1	1	const 1

Функция  $f_0(x) \equiv 0$ . Эту функцию называют **константный ноль** и обозначают  $f_0(x)=0$ .

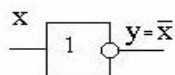
Функция  $f_3(x) \equiv 1$ . Эту функцию называют **константой единицы** и обозначают  $f_3(x)=1$ .

Функция  $f_1(x)$  повторяет значение аргумента  $x$ , поэтому  $f_1(x) \equiv x$ .

Функция  $f_2(x)$  принимает значение, противоположное значению аргумента  $x$ : если  $x=0$ ,  $f_2(x)=1$ ; при  $x=1$ , то  $f_2(x)=0$ . Эта функция названа **инверсией** переменной  $x$ , или **отрицанием**  $x$  ( $x$  с отрицанием).

**Х-черта** называется знаком отрицания.

Математически это выражается формулой:  $P = \bar{x}$  ( $P$  есть не  $x$ ). Элемент, реализующий функцию НЕ, или операцию отрицания, функцию инверсии, называется элементом НЕ (инвертором). На функциональных схемах обозначается так:





### 34. Основные законы и соотношения алгебры логики

С помощью этих законов преобразуют и упрощают исходные логические функции.

1. **Переместительный закон:**
  - для логического сложения:  $(x+y)=(y+x)$
  - для логического умножения:  $xy=yx$
2. **Сочетательный закон:**
  - для логического сложения:  $x+(y+z)=(x+y)+z$
  - для логического умножения:  $(xy)z=x(yz)$
3. **Распределительный закон:**
  - для логического сложения:  $(x+y)z=xz+yz$
  - для логического умножения:  $xy+z=(x+z)(y+z)$
4. **Закон инверсии:**
  - для логического сложения:  $\overline{(x+y)}=\overline{x}\cdot\overline{y}$
  - для логического умножения:  $\overline{(xy)}=\overline{x}+\overline{y}$
5. Отсюда следует закон двойного отрицания:  $\overline{\overline{x}}=x$

} Формулы  
де Моргана

### 35. Функционально полные системы переключательных функций и элементов

универсальными логическими функциями. Одна из задач синтеза схем заключается в выборе типов элементов, из которых должны собираться логические схемы. Основное требование к набору логических элементов - построить с помощью имеющегося этого набора любую сколь угодно сложную схему. Но ввиду того, что законы функционирования элементов описываются переключательными функциями, то сформулированное требование сводится к определению набора таких переключательных функций, с помощью которых можно получить любую сколь угодно сложную функцию, что соответствует операции суперпозиции, заключающейся в замене одних аргументов функции другими.

[ Например, если аргументы функций  $Z(x,y)$  являются, в свою очередь, функциями других аргументов  $X=X(a,b)$  и  $Y=Y(c,d)$ , то можно образовать функцию вида  $Z=Z(a,b,c,d)$ ].

Упрощение логических выражений можно достичь выражением сложных логических функций через другие функции.

Система переключательных функций называется функционально полной, если с помощью функций, входящих в эту систему, применяя операции суперпозиции, можно получить любую сколь угодно сложную переключательную функцию.

Примерами функционально полных систем переключательных функций могут служить:

1) инверсия, дизъюнкция и конъюнкция;

2) т.к. закон инверсии позволяет преобразовывать дизъюнкцию в конъюнкцию и наоборот, то функционально полными будут:

а) дизъюнкция и инверсия;

б) конъюнкция и инверсия.

3) Т.к. через отрицание конъюнкции или отрицание дизъюнкции можно представить любую логическую функцию от  $n$  переменных, и эти функции являются также функционально полными.

### 36. Алгоритм синтеза схем цифровых устройств

При синтезе логических устройств ЭВМ, выполняющих необходимое преобразование цифровой информации, можно выделить следующие этапы:

1. На основании анализа функций, которые должны выполняться данным устройством, формируются логические условия его функционирования в виде соответствующей таблицы.
2. По этой таблице составляется СДНФ логической функции.
3. Производится минимизация логической функции.
4. По упрощенной логической формуле строится функциональная схема устройства, причем минимальному числу и однородности логических элементов **отдается предпочтение.**

### 37. Конечные «полные» автоматы

#### 4. Конечные (полные) автоматы

Устройство, выходное слово в котором в дискретный момент времени  $t_i$  определяется не только по входному слову в тот же момент времени  $t_i$ , а и по его внутреннему состоянию, обусловленному его предшествующими этапами работы

Автомат с памятью задается 3 наборами переменных:

$M(m_1, m_2, \dots, m_l)$  – входной набор (алфавит)

$K(k_1, k_2, \dots, k_j)$  – выходной набор

$Q(q_1, q_2, \dots, q_k)$  – набор внутренних состояний автомата

Функционирование конечного автомата однозначно определено, если установлены связи во времени между 3 его алфавитами. Обычно значения алфавитов сохраняются неизменными на протяжении временных интервалов, называемых тактами.

Закон функционирования конечного автомата возможно задать 3 способами

1. табличным
2. аналитическим
3. графическим.

Табличный способ. Таблица содержит графы:

- Набор входных переменных  $M_t$  в момент дискретного времени  $t$
- Внутренние состояния автомата  $Q_t$  в дискретный момент времени  $t$
- Внутренние состояния автомата  $Q_{t+1}$  в дискретный момент времени  $t+1$
- Значение выходных переменных  $K_{t+1}$  в дискретный момент времени  $t+1$

Аналитический способ реализуется заданием формул для функций переходов и выходов, получаемых из табличного представления образованием СДНФ и СКНФ.

Графическое представление законов функционирования автоматов осуществляется с помощью графов. При этом вершинами графа отображают внутренние состояния автомата, а переход из одного состояния в другое – дугами графа, на которых указывают входные переменные в момент изменения его внутреннего состояния. Выходные переменные для автоматов Мура наносятся внутри вершин графа, а для автомата Мили – на дугах.

#### 41 Классификация элементов ЦВМ. Физическое представление логических 0 и 1

Классификация:

По назначению: логические (реализуют одну булевскую функцию: И, ИЛИ, НЕ), функциональные (реализуют несколько булевых ф-ций: И-ИЛИ-НЕ, И-НЕ, ИЛИ-НЕ, ИЛИ-И-НЕ), вспомогательные (осуществляют усиление, формирование, генерацию сигналов, а так же их преобразования по мощности, амплитуде и длительности, не изменяя их логических значений).

По характеру сигналов, что используются для физического представления 0 и 1, все элементы делятся на потенциальные, импульсные и динамические.

Потенциальный способ реализации двоичных чисел заключается в том, что 0 и 1 передаются различными уровнями потенциалов или определенными значениями тока.

Сигналы 0 и 1 могут быть заданы двояко: в СВП 1 задается более низким уровнем потенциала, чем 0, а в СВП 1 задается более высоким уровнем, чем 0.

Импульсный способ состоит в том, что 1 задается импульсом положительной полярности, а 0 – импульсом отрицательной полярности. Применяется также импульсный способ, когда 1 воспроизводится наличием импульса, а 0 – его отсутствием.

Также используется метод задания численной информации потенциально-импульсным способом, когда в одной схеме отображаются 2 вида сигналов.

По параметрам элементы можно классифицировать в зависимости от:

- Реализации логических функций
- Нагрузочной способности
- Быстродействия
- Предельной рабочей частоты
- Потребляемой мощности
- К общим техническим параметрам относят
- Надежность
- Стоимость
- Конструктивно-механические особенности и др.

#### 43 Регистры

Регистры – устройство предназначенное для приема кратковременного хранения и считывания

двоичной информации а также для выполнения над словом некоторых логических преобразований. регистр представляет собой совокупность триггеров и вспомогательных схем, обеспечивающих выполнение некоторых операций: «сброс» регистра, прием слова из другого устройства, передача слова в другой регистр, сдвиг слова вправо или влево, преобразование последовательного кода слова в параллельный и наоборот, поразрядные логические операции.

По назначению регистры делятся на накопительные (для ввода, хранения и вывода двоичной информации), сдвигающие (для ввода, хранения, сдвига и вывода двоичных чисел), преобразующие.

Разновидности: сегментные регистры – сохраняют адрес начала сегмента: CS – кодового сегмента, DS – сегмента данных, SS – сегмента стека, ES – дополнительного сегмента

Регистры общего назначения: AH – левый байт, AL – правый байт, AX – сумматор в операциях вв/в, некоторых операциях над строками и арифметических операциях, BX – базовый регистр, CX – регистр-счетчик, DX – регистр данных

Регистровые указатели обеспечивают доступ к данным в стеке (SP – указывает на смещение относительно вершины стека, BP – указывает на смещение относительно данных внутри стека)

Индексные регистры используются для расширенной адресации и в операциях сложения/вычитания (SI – является индексом источника и используется совместно с DS, DI – является индексом назначения и используется совместно с ES)

Указатель команд IP содержит смещение, относительно CS, на команду которая должна быть выполнена.

Флаговый регистр хранит в виде битовых полей результаты выполнения операций

Регистр – это узел ЭВМ, предназначенный для хранения, записи и чтения двоичных слов и выполнения над ними некоторых логических операций. Структурно регистры представляют собой совокупность триггеров, число которых соответствует числу разрядов в слове, а также вспомогательных схем, обеспечивающих выполнение некоторых элементарных операций, среди которых могут быть:

1. установка регистра в ноль (сброс) ;
2. прием двоичного слова ;
3. выдача двоичного слова ;
4. сдвиг слова на требуемое количество разрядов ;
5. преобразование последовательного кода слова в параллельный и наоборот ;
6. поразрядные логические операции .

По способу приема и выдачи информации регистры делятся на:

1. параллельные регистры ;
2. регистры сдвига ;
3. комбинационные регистры .

##### Параллельные регистры

Параллельные регистры – используются для выполнения операций приема, хранения, выдачи информации, и поразрядных логических операций над словами. Параллельные регистры представляют собой совокупность RS-, D- или T-триггеров, имеющих связанные входные и выходные цепи.

##### Регистры сдвига

Операция сдвига заключается в перемещении всех цифр слова в направлении от старших к младшим разрядам (правый сдвиг) или от младших к старшим разрядам (левый сдвиг). В освободившиеся разряды всегда заносится «0».

Регистры сдвига используются для выполнения следующих операций:

умножения на  $2^k$ , сдвиг в сторону старших разрядов,

умножение на  $2^{-k}$ , сдвиг в сторону младших разрядов,

преобразование кодов из параллельного в последовательный и наоборот.

Регистры сдвига могут быть построены на RS- или D-триггерах, но сами триггеры должны быть двухступенчатыми, т.е. иметь основные и вспомогательные триггера.

##### Регистры реверсивного сдвига

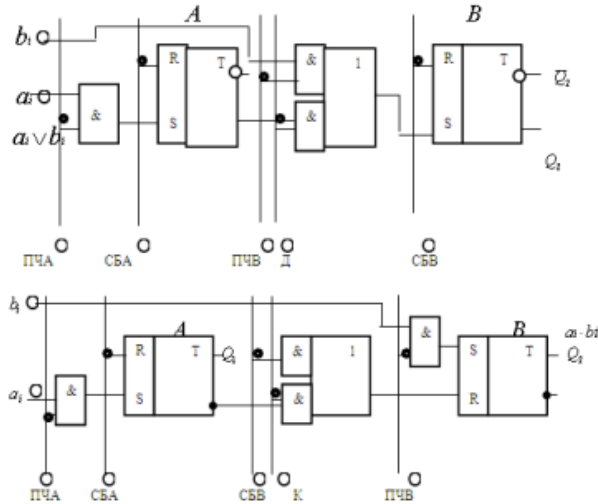
Реверсивный сдвиговый регистр обеспечивает сдвиг информации как влево, так и вправо. Между разрядами реверсивного регистра сдвига встраиваются логические схемы, которые в зависимости от значения сигнала реверса  $N$  пропускают выходы левого или правого соседних разрядов.

Логические операции с использованием регистров.

Операция поразрядной дизъюнкции выполняется за три такта: в первом такте происходит сброс регистров  $A$  и  $B$  сигналами  $СБА$  и  $СБВ$ , во втором такте принимаются

числа в эти регистры с помощью сигналов  $ПЧА$  и  $ПЧВ$ , в третьем такте выполняется операция дизъюнкции с помощью сигнала  $Д$ .

Аналогично выполняется операция поразрядной конъюнкции



$ПЧА$  – прием числа  $A$ .  $СБА$  – сброс регистра  $A$ .  $ПЧВ$  – прием числа  $B$ .  $Д$  – сигнал поразрядной дизъюнкции.  $СБВ$  – сброс регистра  $B$ .  $К$  – сигнал поразрядной конъюнкции

#### 44 Триггеры

##### 14. Триггеры, их назначение, принцип работы. Типы триггеров, способы описания их работы.

Триггер является элементом, который может находиться в одном из двух устойчивых состояний. Одному из этих состояний приписывается значение 1, а другому – 0. состояние триггера распознается по его выходному сигналу. Под влиянием входного сигнала триггер скачкообразно переходит из одного устойчивого состояния в другое, при этом скачкообразно изменяется уровень напряжения его выходного сигнала.

Для удобства использования в схемах вычислительных устройств, триггеры обычно имеют два входа: прямой  $Q$  («выход 1») и инверсный  $\neg Q$  («выход 0»). В единичном состоянии триггера на входе  $Q$  высокий уровень сигнала, а в нулевом – низкий. На выходе  $\neg Q$  наоборот.

Схемы триггеров можно разделить на несколько типов:

с установочными входами – *RS-триггер* – реализует ф-цию вида  $Q(t+1) = S(t) \vee Q(t) \neg R(t)$ ,  $S(t)R(t) = 0$ ,  $t$  – момент времени, предшествующий смене состояния

со счетным входом – *T-триггер* (в простейшем случае может быть построен с использованием двухтактного синхронизируемого RS-триггера) –  $Q(t+1) = Q(t) \neg T(t) \vee \neg Q(t) T(t)$

а также *D-триггер* (реализует ф-цию временной задержки) –  $Q(t+1) = D(t)$

*JK-триггер* – ф-цию переходов можно представить в виде булевой ф-ции:  $Q(t+1) = \neg K(t)Q(t) \vee J(t)\neg Q(t)$  и др.