Міністерство світи і науки України
Дніпровський національний університет імені Олеся Гончара
фізико-технічний факультет
кафедра радіоелектронної автоматики

В.Б.Мазуренко

ПРОГРАМНІ ТА АПАРАТНІ ЗАСОБИ КОМП`ЮТЕРНО-ІНТЕГРОВАНИХ ТЕХНОЛОГІЙ

Конспект лекцій

Дніпро

2018

Наведено конспект лекцій з курсу «Програмні та апаратні засоби комп`ютерно-інтегрованих технологій», який розроблено у відповідності до освітньо-професійної програми другого рівня вищої освіти «Автоматизація та комп’ютерно-інтегровані технології». Для студентів фізико-технічного факультету ДНУ, що навчаються за спеціальністю 151 «Автоматизація та комп’ютерно-інтегровані технології» на другому рівні вищої освіти.

Укладач: доцент кафедри радіоелектронної автоматики фізико-технічного факультету Дніпровського національного університету ім. Олеся Гончара Мазуренко Валерій Борисович.

Лекция № 1

Тема: Компьютерно-интегрированные технологии. Введение.

Оглавление

Определение технологии

Современное общество – это общество технологий. Все предметы, изделия, что нас окружают, средства и системы коммуникации, сервисы и услуги, которыми пользуется современный человек созданы путем применения тех или иных технологий.

Технология в общем случае – это совокупность методов обработки, изготовления, изменения состояния, свойств, формы сырья, материала или полуфабриката в процессе производства.

Все известные технологии можно классифицировать по сфере их приложения, например:

- Сельскохозяйственные технологии
- Производственные технологии
- Технологии научных исследований
- Военные технологии
- Транспортные технологии
- Информационные технологии
- Телекоммуникационные технологии
- Социальные технологии
- Политтехнологии
- Технологии в сфере медицины: лечение, профилактика, оздоровление и др.

Приведенные примеры - это обобщение, которое указывает на совокупность, целый набор отдельных технологий. Так, например, промышленные технологии включают в себя следующие разделы, которые в свою очередь также являются обобщением:

- Технология металлов
- Химическая технология
- Машиностроительные технологии
- Технология строительства и др.

Среди промышленных технологий можно выделить группу под общим названием "Технологии неразрушающего контроля", в которую входят технологии радиоволнового, акустического, оптического и других видов НК.

Если продолжать углубляться, то придём в конечном итоге к конкретной технологии, которая реализуется при производстве конкретного вида продукции, например, "технология контроля сварных соединений трубопроводов радиографическим методом".

Общее число применяемых во всех сферах человеческой деятельности конкретных технологий, по всей видимости, неизвестно.

Зачастую технологии различают по физическим (или другим) явлениям, на основе которых они функционируют, например:

- Технологии, связанные с электричеством
- Акустические технологии

Существует еще одна классификация технологий: по степени вовлечения в технологический процесс труда человека. Роль человека при производстве продукции различна. Наиболее простой классификацией по степени вовлечения труда человека можно считать следующее разделение технологий:

- ручные (используется только труд человека);
- механизированные (механизация тяжелого физического труда)
- автоматизированные (частичное участие человека, чаще всего для манипулирования техникой);
- автоматические (без какого-либо участия человека во время производства).

Средства массовой информации выделяют такое понятие, как "высокие технологии" - это технологии, которые определяют направление научно-технического прогресса современного общества, которые повышают, подымают вверх технический уровень цивилизации, да и сами эти технологии находятся на вершине самых передовых научных знаний. Высокими эти технологии называют еще и потому, что не только создание, но и применение этих технологий требует вовлечения специалистов высочайшей квалификации, требует наличия высокоточного, специализированного оборудования, а также требует высоких финансовых затрат. В силу указанных факторов высокими технологиями обладают только высокоразвитые государства, с высоким экономическим и научным потенциалом. К высоким технологиям, в частности, относят:

- Космические технологии
- Нанотехнологии
- Биотехнологии и др.

Информационные технологии

Среди всех видов технологий отдельно выделяются информационные технологии (ИТ). Что это такое? Напомним, что технология - это совокупность методов обработки, изготовления, изменения состояния, свойств, формы сырья, материала или полуфабриката в процессе производства. Конкретная технология предполагает наличие трех основных составных элементов: материал, средства обработки материала, методы обработки материала. Когда сырьем для обработки и результатом является *информация в форме данных*, а средствами обработки являются программы вычислительной техники, то под

информационной технологией следует понимать совокупность методов (способов) в форме программ вычислительной техники для сбора, накопления, хранения, поиска, обработки и выдачи информации потребителю.

Компьютерно-интегрированные технологии

Применяемые технологии постоянно совершенствуются в направлении уменьшения доли ручного труда, повышения производительности и качества.

Появление компьютеров позволило не только механизировать технологические процессы, но и перейти к автоматизированным и даже автоматическим технологиям, управление механизмами в которых берет на себя компьютер. Компьютер, а точнее компьютерная техника, становится составной частью технологического процесса, иначе говоря, интегрируется в него. Многие технологии уже в принципе не реализуются без компьютерных систем. К таким технологиям относятся многие телекоммутационные технологии, технологии работы с видео и фото изображениями, технологии управления космической и ракетной техникой, энергетическими установками, системами распределения электроэнергии и, конечно же, информационные технологии. Однако, многие технологии по-прежнему не требуют, применения компьютерной техники, а некоторые до сих пор остаются ручными, например, технологии художественных промыслов.

Таким образом, под компьютерно-интегрированной технологией можно было бы понимать любую технологию, которая реализуется с применением компьютерной техники. Однако, если пользоваться данным определением, возникает противоречие. Получается, что если выполняется глажка ткани обычным электрическим утюгом, то это технология глажки ткани. Если в утюге используется микропроцессор для управления нагревом, то это - компьютерно-интегрированная технология глажки?

Для того, чтобы разобраться с данным вопросом, следует в первую очередь выяснить, для решения каких задач применяется компьютерная техника в технологическом процессе? На самом деле, число этих задач совсем невелико:

1) получение данных об объекте, то есть измерение,
2) выработка управляющего воздействия.

В автоматических и автоматизированных технологиях - это все, других задач нет. Таким образом, компьютер имеет дело не с технологией изготовления, лечения или глажки, а только с технологиями измерения и управления, без которых данная конкретная технология изготовления, лечения или глажки не может быть реализована.

Существует, правда, два исключения, представленные ниже.

1. Отдельно необходимо выделить технологии научных исследований, где компьютерная техника используется для проведения обработки результатов исследований, а также для моделирования.

2. Особые задачи компьютерные системы решают в технологиях генерации нематериальных (мультимедийных: изображение, звук, видео) и материальных объектов (например, скульптур). Эти технологии близки к технологиям искусственного интеллекта.

Обобщая вышесказанное, можно дать следующее определение компьютерно-интегрированных технологий.

Компьютерно-интегрированными технологиями (КИТ) называются технологии измерения, управления, научного исследования и генерации объектов, которые реализуются с применением компьютерной техники.

Примечание: Компьютерно-интегрированные технологии для решения большого спектра задач используют информационные технологии, при этом ИТ не являются частью КИТ, также как и КИТ не является частью ИТ.

Возвращаясь к приведенному выше примеру можно говорить не о компьютерно-интегрированной технологии глажки, а о компьютерно-интегрированной технологии управления нагревом утюга. Терминологически и лексически применительно к какой-либо технологии, устройству, агрегату правильно говорить в отношении КИТ только об управлении, то есть: компьютерно-интегрированная технология управления автомобилем, КИТ управления ткацким станком и т.д. При этом подразумевается, что данная КИТ включает и измерение, так как управление без получения данных об объекте, то есть без измерения, невозможно.

В настоящем курсе технологии научного исследования и генерации объектов не рассматриваются, так как КИТ научного исследования изучается в целом ряде других читаемых на кафедре курсах, а КИТ генерации объектов в нестоящее время находится в разработке и не может рассматриваться как установившаяся.

Программно-аппаратные средства компьютерно-интегрированных технологий

Компьютерно-интегрированные технологии реализуются в виде управляющих и измерительных систем. Можно сказать, что системы, в которых реализуются компьютерно-интегрированные технологии, представляют собой следующее поколение измерительных систем, а также АСУ ТП - автоматизированных систем управления технологическими процессами. То есть - это те же системы, но построенные на основе компьютерной техники.

Классическая архитектура управляющей системы воплощена в моносистеме (рис.1) – сосредоточенной системе, которая обеспечивает управление сосредоточенным объектом управления. Сосредоточенными в данном случае можно называть системы, в которых расстояние от датчиков и приводов до компьютера исчисляется метрами или десятками метров.

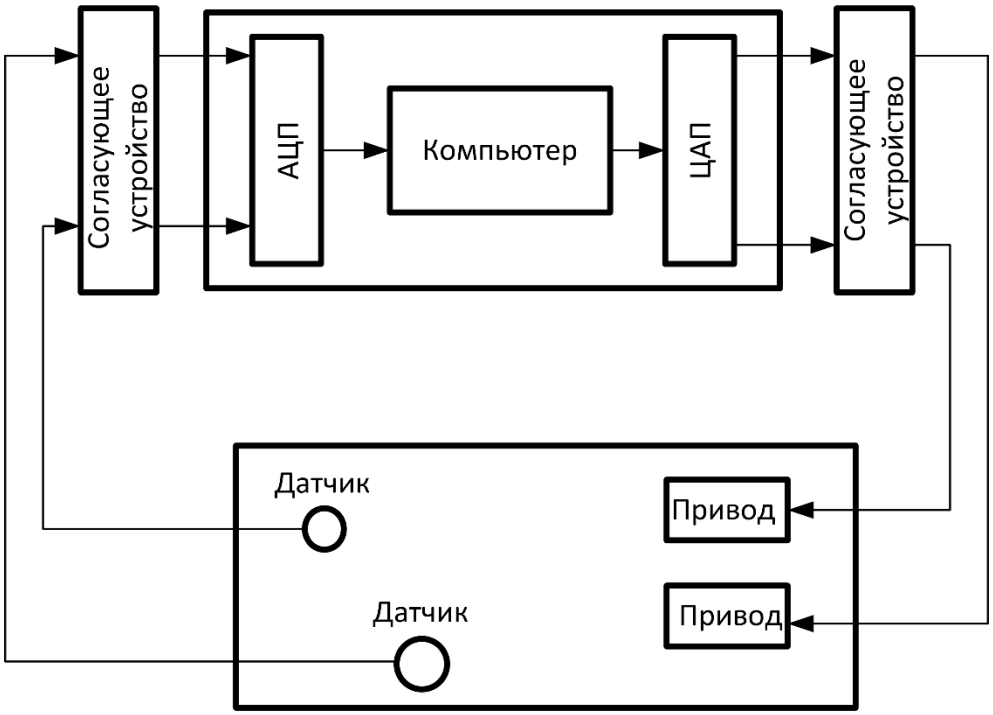


Рис. 1 Сосредоточенная система

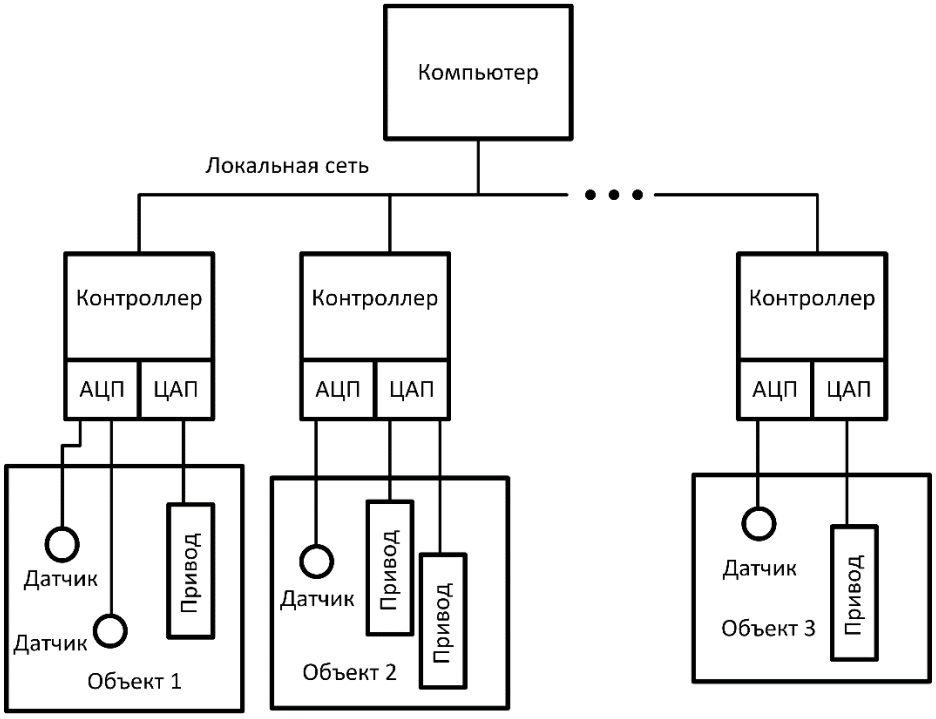


Рис.2 Распределенная система

Со временем стали развиваться распределенные системы (рис. 2), которые обеспечивают измерение и управление на совокупности объектов. В этом случае расстояния обычно составляют сотни метров и километры, а в некоторых случаях - десятки километров.

Класс решаемых при помощи компьютерно-интегрированных технологий постоянно расширяется. В последнее время все более широкое применение находят глобально распределенные системы - системы в которых объекты контроля отстоят на значительные расстояния - расстояния, которые уже не ограничены какими-либо пределами. Такие системы строятся по схеме, представленной на следующем рисунке. Они, в основном, являются измерительными.

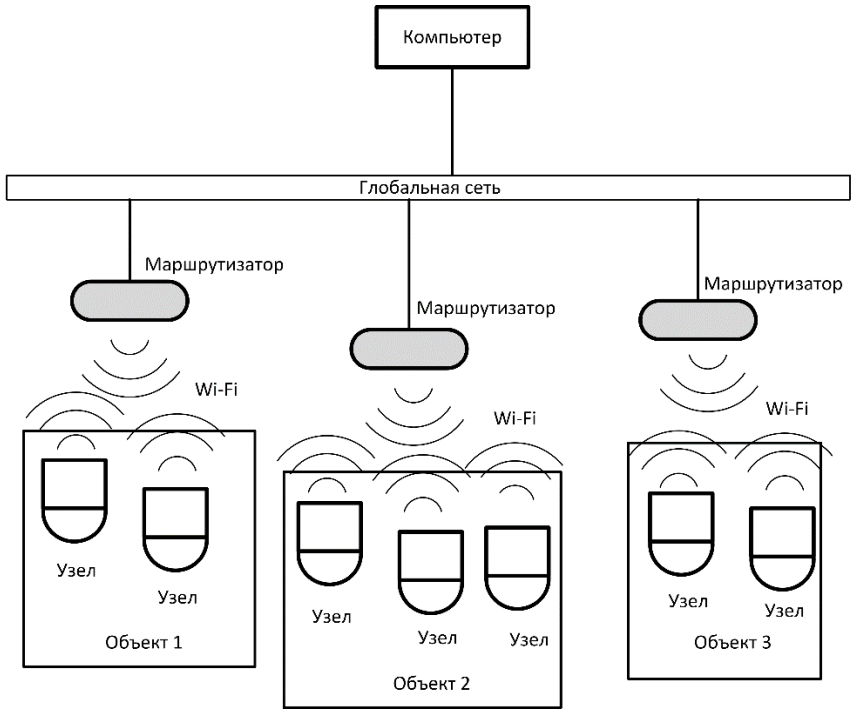


Рис. 3 Глобально распределенная система

Как можно видеть, системы, реализующие компьютерно-интегрированные технологии, в основном строятся с применением таких аппаратных средств:

- компьютер,
- аналогово-цифровой преобразователь (АЦП),
- цифро-аналоговый преобразователь (ЦАП),
- датчики (узлы),
- привода (или исполнительные устройства),
- локальная сеть,
- глобальная сеть.

Все причисленные аппаратные средства будут рассмотрены в настоящем курсе, который носит название "Програмні та апаратні засоби комп`ютерно-інтегрованих технологій".

Также будут рассмотрены программные средства, под управлением которых функционируют аппаратные средства. Для освоения компьютерно-интегрированных технологий необходимо получить знания о следующих программных средствах:

- системное программное обеспечение (ПО),
- прикладное ПО,
- технология "клиент-сервер",
- базы данных,
- системы программирования контроллеров,
- SCADA-системы.

Контрольные вопросы по теме

Уровень модуля

1. Дайте определение понятия "технология".
2. Приведите пример классификации технологий по сфере приложения.
3. Приведите примеры технологий, различающихся по виду физических явлений, которые положены в основу их функционирования?
4. Каким образом классифицируются технологии по степени вовлечения человеческого труда?
5. Какие технологии называют высокими?
6. Почему, по вашему мнению, применяется термин высокие технологии?
7. Какие технологии можно отнести к высоким технологиям?
8. Какие технологии называют информационными?
9. Для решения каких задач применяется компьютерная техника в технологическом процессе?
10. Дайте определение понятию "компьютерно-интегрированные технологии".
11. Информационные технологии и компьютерно-интегрированные технологии – это одно и то же? Почему?
12. Представьте на рисунке функциональную схему сосредоточенной управляющей системы.
13. Представьте на рисунке функциональную схему распределенной управляющей системы.
14. Представьте на рисунке функциональную схему глобально распределенной измерительной системы.
15. Перечислите основные аппаратные средства компьютеризированных систем управления и контроля.
16. Перечислите основные программные средства компьютеризированных систем управления и контроля.

Уровень курса

1. Понятие компьютерно-интегрированных технологий.
2. Архитектура сосредоточенной компьютерно-интегрированной системы.
3. Архитектура распределенной компьютерно-интегрированной системы.

Лекция № 2

Тема: Общее устройство компьютера

Оглавление

Источники:

1. Таненбаум Э., Остин Т. Архитектура компьютера. 6-е изд. — СПб.: Питер, 2013. — 816 с.: ил.

https://rutracker.org/forum/viewtopic.php?t=4956359

Машина фон Неймана

В 1946 году Д. фон Нейман, Г. Голдстайн и А. Беркс в своей совместной статье изложили принципы построения и функционирования электронно-вычислительной машины (ЭВМ). Предложенное построение компьютера получило название "машина фон Неймана". Принципы Неймана реализованы во всех используемых в настоящее время ЭВМ. Современные компьютеры, пр своей сути, - это машины фон Неймана.

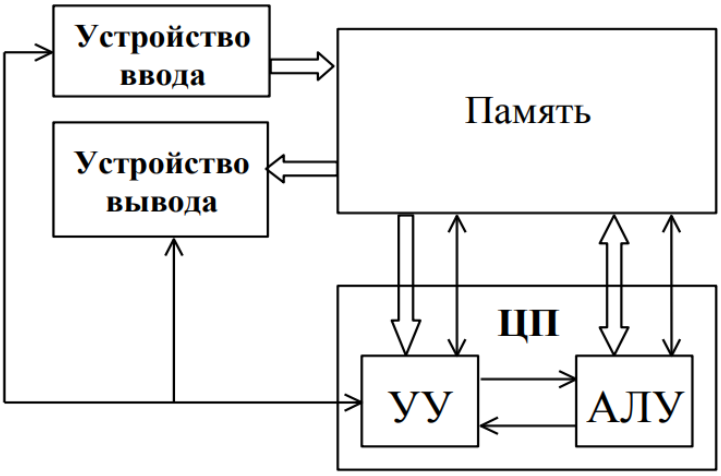


Рис. 2.1 Машина фон Неймана

Принципы фон Неймана:

1. **Использование двоичной системы счисления в вычислительных машинах**. Преимущество перед десятичной системой счисления заключается в том, что устройства можно делать достаточно простыми, арифметические и логические операции в двоичной системе счисления также выполняются достаточно просто.

2. **Программное управление ЭВМ**. Работа ЭВМ контролируется программой, состоящей из набора команд. Команды выполняются последовательно друг за другом. Созданием машины с хранимой в памяти программой было положено начало тому, что мы сегодня называем программированием.

3. **Память компьютера используется не только для хранения данных, но и программ**. При этом и команды программы и данные кодируются в двоичной системе счисления, т.е. их способ записи одинаков. Поэтому в определенных ситуациях над командами можно выполнять те же действия, что и над данными.

4. **Ячейки памяти ЭВМ имеют адреса, которые последовательно пронумерованы**. В любой момент можно обратиться к любой ячейке памяти по ее адресу. Этот принцип открыл возможность использовать переменные в программировании.

5. **Возможность условного перехода в процессе выполнения программы**. Несмотря на то, что команды выполняются последовательно, в программах можно реализовать возможность перехода к любому участку кода.

Двоичная система счисления

Двоичная система счисления — позиционная система счисления с основанием 2. Благодаря непосредственной реализации в цифровых электронных схемах на логических вентилях, двоичная система используется практически во всех современных компьютерах и прочих вычислительных электронных устройствах.

В ЭВМ применяется двоичная система счисления, т.е. все числа в компьютере представляются с помощью нулей и единиц. Компьютер может обрабатывать информацию, представленную только в цифровой форме. Для преобразования числовой, текстовой, графической, звуковой информации в цифровую необходимо применить кодирование. Кодирование – это преобразование данных одного типа через данные другого типа. В ЭВМ применяется система двоичного кодирования, основанная на представлении данных последовательностью двух знаков: 1 и 0, которые называются двоичными цифрами (binary digit – сокращенно bit). Двоичная система счисления – позиционная система счисления с основанием 2. Благодаря непосредственной реализации в цифровых электронных схемах на логических вентилях, двоичная система используется практически во всех современных компьютерах и прочих вычислительных электронных устройствах.

Таким образом, единицей информации в компьютере является один бит, т.е. двоичный разряд, который может принимать значение 0 или 1. Восемь последовательных бит составляют байт. В одном байте можно закодировать значение одного символа из 256 возможных (256 = 2 в степени 8). Более крупной единицей информации является килобайт (Кбайт), равный 1024 байтам (1024 = 2 в степени 10). Еще более крупные единицы измерения данных: мегабайт, гигабайт, терабайт (1 Мбайт = 1024 Кбайт; 1 Гбайт = 1024 Мбайт; 1 Тбайт = 1024 Гбайт).

Целые числа кодируются двоичным кодом довольно просто (путем деления числа на два). Отрицательные целые числа кодируются в так называемом дополнительном коде. Специальный код применяется для кодирования вещественных чисел, дробных чисел, которые представляются в компьютере в виде "чисел с плавающей точкой". Однако и эти числа записываются в виде совокупности нулей и единичек - в двоичном коде. Для кодирования нечисловой информации используется следующий алгоритм: все возможные значения кодируемой информации нумеруются и эти номера

кодируются с помощью двоичного кода. Например, для представления текстовой информации используется таблица нумерации символов или таблица кодировки символов, в которой каждому символу соответствует целое число (порядковый номер). Восемь двоичных разрядов могут закодировать 256 различных символов.

В любом случае, в памяти компьютера, в процессоре, в шинах имеются только сигналы или состояния 0 и 1, других нет. Все данные представлены в двоичном коде. Все команды на управление компьютером также состоят из нулей и единиц, команды - это тоже последовательность двоичных чисел. Невозможно понять, что находится в данной ячейке памяти. Например, записано 10011101. Что это: простое число, часть вещественного числа или команда на управление процессором? Все зависит от того, как компьютер интерпретирует эти данные: как простое число, как часть вещественного числа или как команду. Для того, чтобы правильно интерпретировать, необходимо заранее знать, какой тип информации находится в данной ячейке памяти.

Организация компьютерных систем

Цифровой компьютер состоит из связанных между собой процессоров, модулей памяти и устройств ввода-вывода.

На рис. 2.2 показана структура обычного компьютера с шинной организацией. Центральный процессор — это мозг компьютера. Его задача — выполнять программы, находящиеся в основной памяти. Для этого он вызывает команды из памяти, определяет их тип, а затем выполняет одну за другой. Компоненты соединены шиной, представляющей собой набор параллельно связанных проводов для передачи адресов, данных и управляющих сигналов. Шины могут быть внешними (связывающими процессор с памятью и устройствами ввода-вывода) и внутренними. Современный компьютер использует несколько шин.

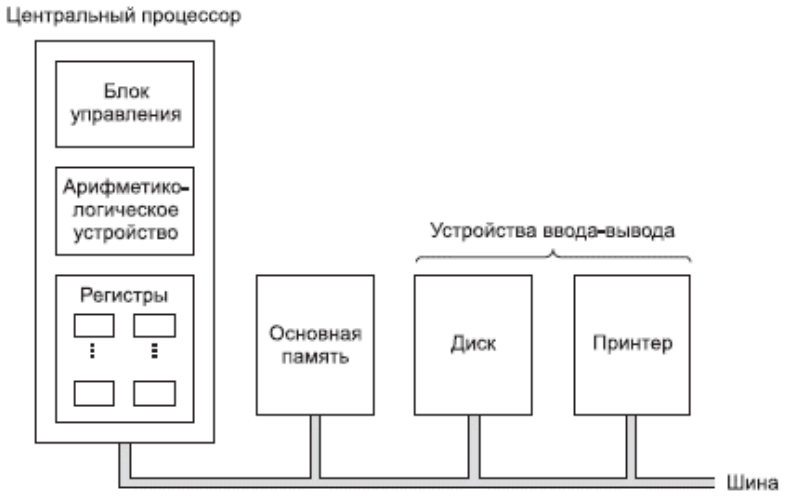


Рис. 2.2 Структура компьютера с шинной организацией

Процессор

Процессор состоит из нескольких частей. Блок управления отвечает за вызов команд из памяти и определение их типа. Арифметико-логическое устройство выполняет арифметические операции (например, сложение) и логические операции (например, логическое И).

Внутри центрального процессора находится быстрая память небольшого объема для хранения промежуточных результатов и некоторых команд управления. Эта намять состоит из нескольких регистров данных, каждый из которых выполняет определенную функцию. Обычно размер всех регистров одинаков. Каждый регистр содержит одно число в диапазоне, верхняя граница которого зависит от размера регистра. Операции чтения и записи с регистрами выполняются очень быстро, поскольку они находятся внутри центрального процессора.

Самый важный регистр – счетчик команд или указатель команд, который указывает, какую команду нужно выполнять следующей. Еще есть регистр команд, в котором находится выполняемая в данный момент команда. У большинства компьютеров имеются и другие регистры, одни из них многофункциональны, другие служат лишь какие-либо конкретным целям. Третьи регистры используются операционной системой для управления компьютером.

Устройство центрального процессора

Внутреннее устройство тракта данных типичного фон-неймановского процессора иллюстрирует рис. 2.3. Тракт данных состоит из регистров данных (обычно от 1 до 32), арифметико-логического устройства (АЛУ) и нескольких соединительных шин. В самом АЛУ имеются входные и выходные регистры. Регистры процессора и регистры АЛУ - это разные регистры. Содержимое регистров данных процессора (для примера на рисунке - это регистры №3 и №4) поступает во входные регистры АЛУ, которые на рис. 2.3 обозначены буквами А и В. АЛУ непосредственно работает только со своими входными и выходными регистрами. Во входных регистрах АЛУ находятся входные данные АЛУ, пока АЛУ производит вычисления.

АЛУ выполняет сложение, вычитание и другие простые операции над входными данными и помещает результат в выходной регистр. Содержимое этого выходного регистра может записываться обратно в один из регистров процессора (на рисунке это регистр №1 процессора) или сохраняться в памяти, если это необходимо. На рис. 2.3 представлена операция сложения, но АЛУ может выполнять и другие операции.

Большинство команд можно разделить на две группы: команды типа регистр – память и типа регистр-регистр. Команды первого типа вызывают

слова из памяти, помещают их в регистры процессора, где они используются в качестве входных данных АЛУ. Слова – это такие элементы данных, порции данных, которые перемещаются между памятью и регистрами. Размер слова обычно соответствует разрядности регистра данных. Так, у 16-разрядных микропроцессоров слово имеет длину 16 бит, а у 32-разрядных микропроцессоров – 32 бита. Другие команды этого типа помещают данные из регистров обратно в память.

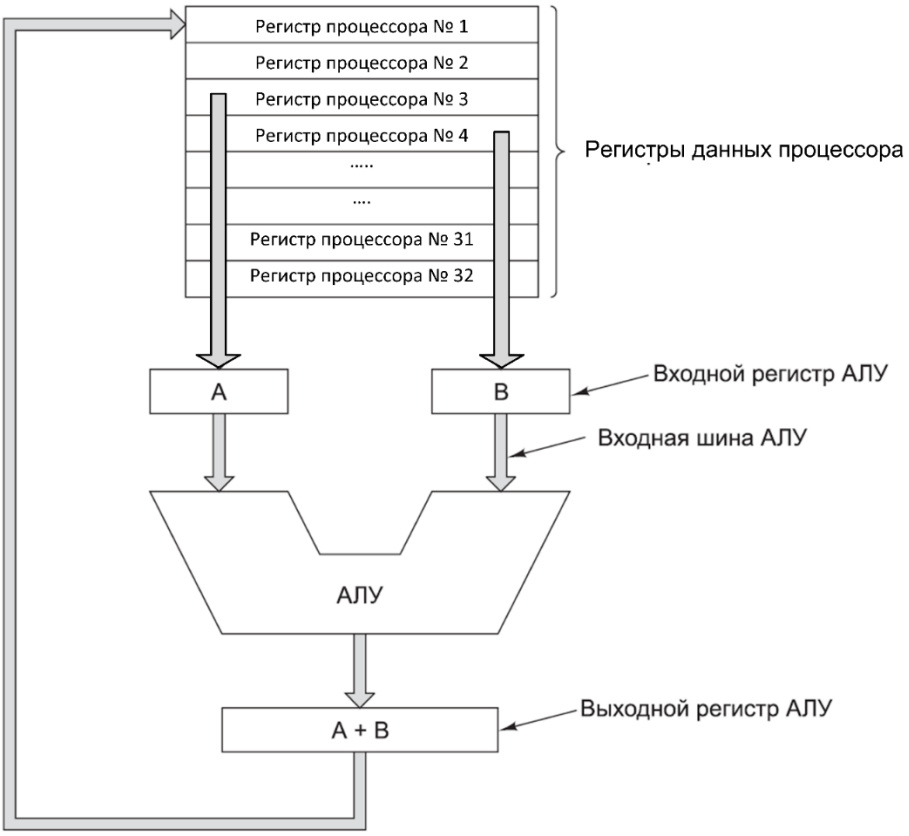


Рис. 2.3 Функциональная схема процессора

Команды второго типа вызывают два операнда из регистров, помещают их во входные регистры АЛУ, выполняют над ними какую-нибудь арифметическую или логическую операцию и переносят результат обратно в один из регистров процессора. Этот процесс называется циклом тракта данных. В какой-то степени он определяет, что может делать машина. Современные компьютеры оснащаются несколькими АЛУ, работающими параллельно и специализирующимися на разных функциях. Чем быстрее происходит цикл тракта данных, тем быстрее компьютер работает.

Выполнение команд

Центральный процессор выполняет каждую команду за несколько шагов. Он делает следующее:

1. Вызывает следующую команду из памяти и переносит ее в регистр команд.

2. Меняет положение счетчика команд, который после этого указывает на следующую команду.

3. Определяет тип вызванной команды.

4. Если команда использует слово из памяти, определяет, где находится это слово.

5. Переносит слово, если это необходимо, в регистр центрального процессора.

6. Выполняет команду.

7. Переходит к шагу 1, чтобы начать выполнение следующей команды.

Такая последовательность шагов (выборка – декодирование – исполнение) является основой работы всех компьютеров.

Контрольные вопросы по теме

Уровень модуля

Уровень курса

1. Машина фон Неймана.
2. Двоичная система счисления.
3. Организация компьютерных систем.
4. Устройство центрального процессора.

Лекция № 3

Тема: Общее устройство компьютера. Часть 2. Память компьютера.

Оглавление

Источники:

1. Таненбаум Э., Остин Т. Архитектура компьютера. 6-е изд. — СПб.: Питер, 2013. — 816 с.: ил.

   https://rutracker.org/forum/viewtopic.php?t=4956359

Основная память

Память – это тот компонент компьютера, в котором хранятся программы и данные. Также часто встречается термин «запоминающее устройство». Без памяти, откуда процессоры считывают и куда записывают информацию, не было бы современных цифровых компьютеров.

Бит

Основной единицей хранения данных в памяти является двоичный разряд, который называется битом. Бит может содержать 0 или 1. Эта самая маленькая единица памяти.

Адреса памяти

Память состоит из ячеек, каждая из которых может хранить некоторую порцию информации. Каждая ячейка имеет номер, который называется адресом. Выполняемая компьютером программа может ссылаться на определенную ячейку по ее адресу. Если память содержит n ячеек, они будут иметь адреса от 0 до n —1. Каждая ячейка памяти состоит из определенного числа разрядов. Каждый разряд хранит один бит информации. Все ячейки памяти одного компьютера содержат одинаковое число разрядов и соответственно могут хранить одинаковое число битов. Если ячейка состоит из $к$ разрядов, она может хранить ровно $к$ бит информации. В такой ячейке может быть записана любая из $2^к$ комбинаций нулей и единиц. На рис. 3.1 показаны различные варианты организации 96-разрядной памяти, то есть памяти, которая должна содержать 96 бит информации. Отметим, что соседние ячейки по определению имеют последовательные адреса.

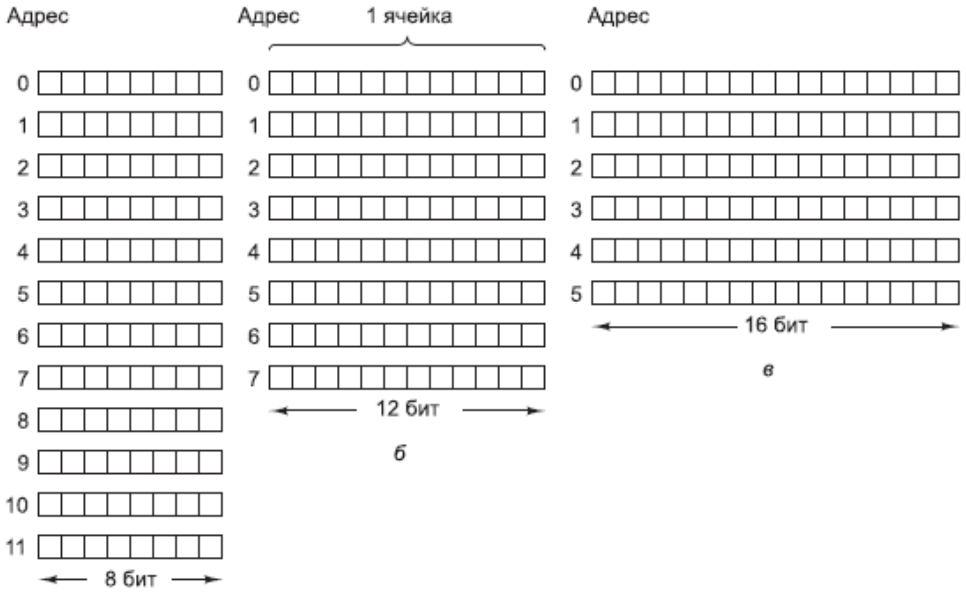


Рис. 3.1. Три варианта организации 96-разрядной памяти

В компьютерах используется двоичная система счисления, адреса памяти также выражаются в двоичных числах. Если адрес состоит из *m* бит, максимальное число адресуемых ячеек составит 2^m. Например, адрес для обращения к памяти, изображенной на рис. 3.1*а* должен состоять по крайней мере из 4 бит, чтобы выражать все числа от 0 до 11. При устройстве памяти, показанном на рис. 3.1*б* и 3.1*в* достаточно 3-разрядного адреса. Число битов в адресе определяет максимальное количество адресуемых ячеек памяти и не зависит от числа битов в ячейке. 12-разрядные адреса нужны и памяти из 2^{12} ячеек по 8 бит каждая, и памяти из 2^{12} ячеек по 64 бит каждая.

Ячейка—минимальная адресуемая единица памяти. В последние годы практически все производители выпускают компьютеры с 8-разрядными ячейками, которые называются байтами. Байты группируются в слова. В компьютере с 32-разряднымн словами на каждое слово приходится 4 байт, а в компьютере с 64-разрядными словами —8 байт. Такая единица, как слово, необходима, поскольку большинство команд производят операции над целыми словами (например, складывают два слова). Таким образом, 32-разрядная машина содержит 32-разрядные регистры и команды для манипуляций с 32-разряднымн словами, тогда как 64-разрядная машина имеет 64-разрядные регистры и команды для перемещения, сложения, вычитания и других операций над 64-разрядными словами.

Кэш-память

Процессоры всегда работали быстрее, чем память. Поскольку процессоры и память совершенствуются параллельно, это несоответствие сохраняется. Поскольку на микросхему можно помещать все больше и больше транзисторов, разработчики процессоров создают конвейерные и суперскалярные архитектуры, что еще больше увеличивает быстродействие процессоров. Разработчики памяти обычно используют новые технологии для увеличения емкости, а не быстродействия, что делает разрыв еще большим. На практике такое несоответствие в скорости работы приводит к тому, что, когда процессор обращается к памяти, проходит несколько машинных циклов, прежде чем он получит запрошенное слово. Чем медленнее работает память, тем дольше процессору приходится ждать, тем больше циклов проходит.

Для того, что память работала так же быстро, как процессор ее необходимо размещать прямо на микросхему процессора (поскольку информация через шину поступает очень медленно). Размещение памяти большого объема на микросхеме процессора увеличивает его размеры, а следовательно, и стоимость. При этом существуют технические ограничения на размеры создаваемых процессоров.

Память небольшого объема с высокой скоростью работы называется кэш-памятью. Основная идея кэш-памяти проста: в ней находятся слова, которые чаще всего используются. Если процессору нужно какое-нибудь слово, сначала он обращается к кэш-памяти. Только в том случае, если слова там нет, он обращается к основной памяти. Если значительная часть слов находится в кэш-памяти, среднее время доступа значительно сокращается.

Таким образом, успех или неудача зависит от того, какая часть слов находится в кэш-памяти. Известно, что программы не обращаются к памяти наугад. Если программе нужен доступ к адресу *A*, то скорее всего после этого ей понадобится доступ к адресу, расположенному поблизости от *A*. Практически все команды обычной программы (за исключением команд перехода и вызова процедур) вызываются из последовательных областей памяти. Кроме того, большую часть времени программа тратит на циклы, когда ограниченный набор команд выполняется снова и снова. Точно так же при работе с матрицами программа, скорее всего, будет многократно обращаться к одной и той же матрице, прежде чем перейдет к чему-либо другому.

Ситуация, когда при последовательных обращениях к памяти в течение некоторого промежутка времени используется только небольшая ее область, называется принципом локальности. Этот принцип составляет основу всех систем кэш-памяти. Идея состоит в том, что, когда определенное слово вызывается из памяти, оно вместе с соседними словами переносится в кэш-память, что позволяет при очередном запросе быстро обращаться к следующим словам. Общее устройство процессора, кэш-памяти и основной памяти иллюстрирует рис. 3.1. Если слово считывается или записывается *к* раз, компьютеру требуется сделать одно обращение к медленной основной памяти и *к*–1 обращений к быстрой кэш-памяти. Чем больше *к*, тем выше общая производительность.

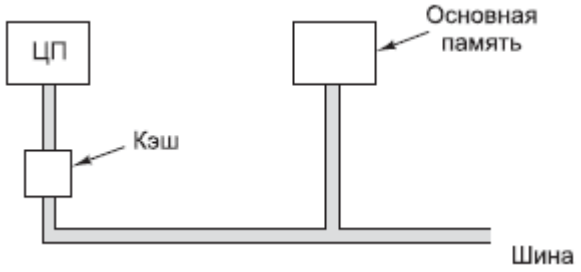


Рис. 2.13. Кэш-память должна находиться между процессором и основной памятью.

Возникает вопрос: должны ли команды и данные находиться вместе в общей кэш-памяти. Проще всего разработать объединенную кэш-память

(unified cache), в которой будут храниться и данные, и команды. В этом случае вызов команд и данных автоматически уравновешивается. Однако в настоящее время существует тенденция к использованию разделенной кэш-памяти (split cache), когда команды хранятся в одной кэш-памяти, а данные — в другой. Такая архитектура также называется гарвардской (Harvard architecture), поскольку идея использования отдельной памяти для команд и отдельной памяти для данных впервые воплотилась в компьютере Marc III, который был создан Говардом Айкеном (Howard Aiken) в Гарварде. Современные разработчики пошли по этому пути, поскольку сейчас широко распространены конвейерные архитектуры, а при конвейерной организации обращения и к командам, и к данным (операндам) должны осуществляться одновременно. Разделенная кэш-память позволяет осуществлять параллельный доступ, а общая – нет. К тому же, поскольку команды обычно не меняются во время выполнения программы, содержание кэша команд не приходится записывать обратно в основную память.

В настоящее время очень часто кэш-память первого уровня располагается прямо на микросхеме процессора, кэш-память второго уровня – не на самой микросхеме, но в корпусе процессора, а кэш-память третьего уровня – еще дальше от процессора.

Сборка модулей памяти и их типы

Со времен появления полупроводниковой памяти и до начала 90-х годов все микросхемы памяти производились, продавались и устанавливались в виде отдельных микросхем. Эти микросхемы вмещали от 1 Кбит до 1 Мбит информации и выше. В первых персональных компьютерах часто оставлялись пустые разъемы, чтобы покупатель в случае необходимости мог вставить дополнительные микросхемы памяти.

В настоящее время распространен другой подход. Группа микросхем (обычно 8 или 16) монтируется на одну крошечную печатную плату и продается как один блок. Он называется SIMM (Single Inline Memory Module – модуль памяти с односторонним расположением выводов) или DIMM (Dual Inline Memory Module —модуль памяти с двухсторонним расположением выводов). На платах SIMM устанавливается один краевой разъем с 72 контактами; при этом скорость передачи данных за один тактовый цикл составляет 32 бит. Модули DIMM, как правило, снабжаются двумя краевыми разъемами (по одному на каждой стороне платы) с 120 контактами; таким образом, общее количество контактов достигает 240, а скорость передачи данных возрастает до 64 бит за цикл. Типичный модуль DIMM изображен на рис. 3.3.

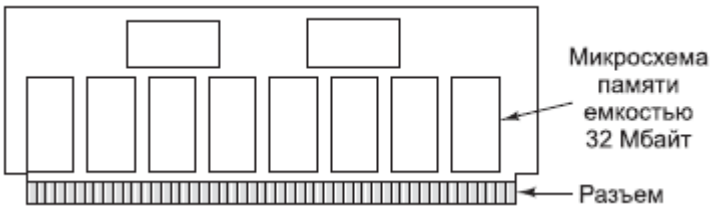


Рис. 3.3. 4-гигабайтный модуль DIMM с восемью 256-мегабайтными микросхемами с каждой стороны. Другая сторона выглядит аналогично.

Обычно модули DIMM содержат 8 микросхем но 256 Мбайт каждая. Таким образом, весь модуль вмещает 2 Гбайт информации. Во многих компьютерах предусматривается возможность установки четырех модулей; следовательно, при использовании модулей по 2 Гбайт общий объем памяти достигает 8 Гбайт (и более при использовании модулей большей емкости).

В настоящее время наиболее распространенными являются DDR4 SDRAM – синхронная динамическая память с произвольным доступом.

Вспомогательная память

Иерархическая структура памяти является традиционным решением проблемы хранения больших объемов данных (рис. 3.4). На самом верху иерархии находятся регистры процессора. Доступ к регистрам осуществляется быстрее всего. Дальше идет кэш-память, объем которой сейчас составляет от 32 Кбайт до нескольких мегабайт. Затем следует основная память, объем которой в настоящее время лежит в диапазоне от 1 Гбайт до сотен гигабайт. Затем идут магнитные диски и твердотельные накопители для долгосрочного хранения данных. Нижний уровень иерархии занимают накопители на магнитной ленте и оптические диски для хранения архивов.

Рис. 3.4. Пятиуровневая организация памяти

По мере продвижения сверху вниз по иерархии меняются три параметра. Во-первых, увеличивается время доступа. Доступ к регистрам занимает несколько наносекунд, доступ к кэш-памяти – немного больше, доступ к основной памяти – несколько десятков наносекунд. Дальше идет большой разрыв: доступ к дискам происходит по крайней мере в 10 раз медленнее для твердотельных дисков и в сотни раз медленнее для магнитных дисков. Время доступа к магнитным лентам и оптическим дискам вообще может измеряться в секундах (поскольку эти накопители информации еще нужно взять и поместить в соответствующее устройство).

Во-вторых, растет объем памяти. Регистры могут содержать в лучшем случае 128 байт, кэш-память – десятки мегабайт, основная память — гигабайты, магнитные диски —терабайты.

В-третьих, увеличивается количество битов за один доллар. Стоимость объема основной памяти измеряется в долларах за мегабайт, твердотельных накопителей —в долларах за гигабайт, магнитных дисков и лент – в центах за гигабайт или еще дешевле.

Магнитные диски

Магнитный диск состоит из одной или нескольких алюминиевых поверхностей, покрытых магнитным слоем. Изначально их диаметр составлял 50 см, сейчас — от 3 до 9 см, у портативных компьютеров —меньше 3 см, причем это значение продолжает уменьшаться. Головка диска, содержащая индукционную катушку, двигается над поверхностью диска, опираясь на воздушную подушку. Когда через головку проходит положительный или отрицательный ток, он намагничивает поверхность под головкой. При этом магнитные частицы намагничиваются направо или налево в зависимости от полярности тока. Когда головка проходит над намагниченной областью, в ней (в головке) возникает положительный или отрицательный ток, что дает возможность считывать записанные ранее биты. Поскольку диск вращается под головкой, поток битов может записываться, а потом считываться. Конфигурация дорожки диска показана на рис. 3.5.

Большинство магнитных дисков состоит из нескольких пластин, расположенных друг под другом, как показано на рис. 3.6. Каждая поверхность снабжена кронштейном и головкой. Кронштейны скреплены таким образом, что одновременно могут перемещаться на разные расстояния от оси. Совокупность дорожек, расположенных на одном расстоянии от центра, называется цилиндром.

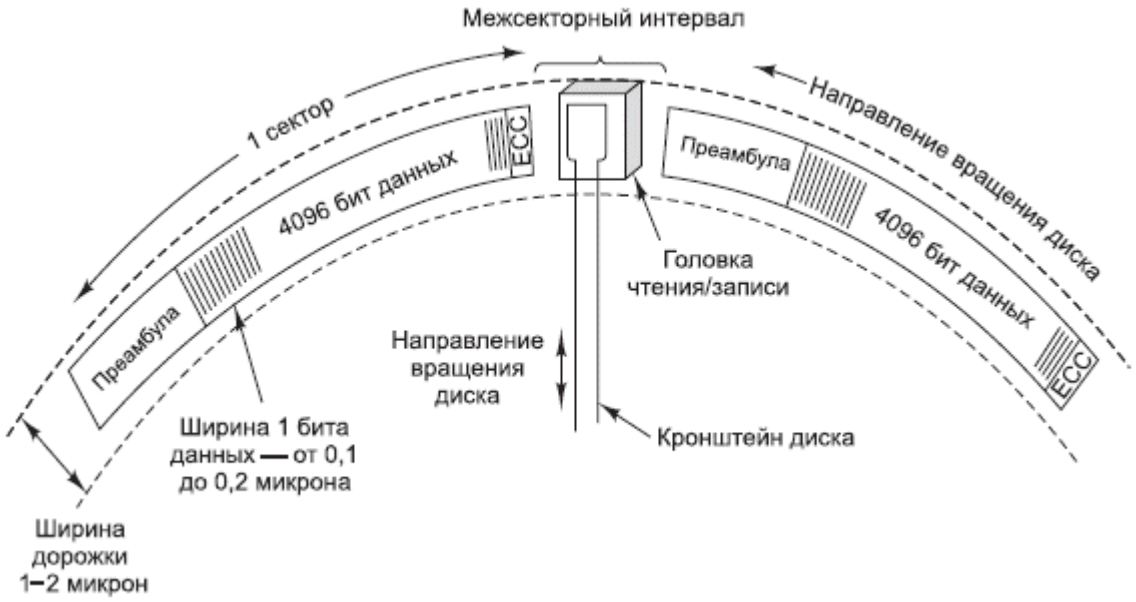


Рис. 3.5. Фрагмент дорожки диска (два сектора)

В современных моделях дисков для ПК устанавливается от 1 до 12 пластин, содержащих от 12 до 24 рабочих поверхностей. На одной пластине современных высокопроизводительных дисков может храниться до 12 Тбайт данных, и со временем это значение будет наверняка превышено.

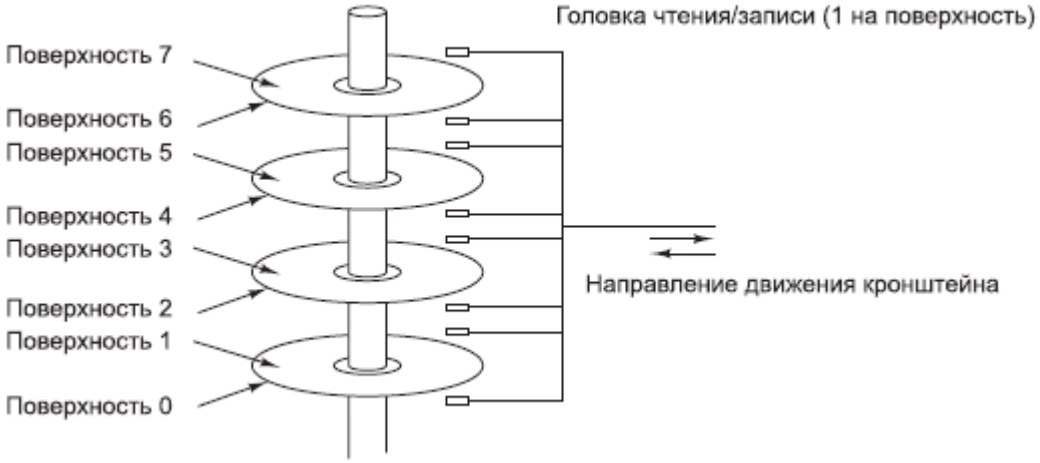


Рис. 3.6. Винчестер с четырьмя дисками

IDE-диски

Прототипом дисков современных персональных компьютеров был диск машины IBM PC XT. Это был диск Seagate на 10 Мбайт, управляемый контроллером Xebec на встроенной карте. Сначала контроллер помещался на отдельной плате, а с выходом в середине 80-х годов устройств **IDE** (Integrated Drive Electronics – устройство со встроенным контроллером) стал встраиваться в материнскую плату.

На смену IDE-дискам пришли устройства **EIDE** (Extended IDE— усовершенствованные устройства со встроенным контроллером), поддерживающие дополнительную схему адресации LBA (Logical Block Addressing — линейная адресация блоков).

Стандарт EIDE совершенствовался вместе с развитием технологического прогресса и его преемником считается **ATA-3** (AT Attachment). Следующая версия стандарта, названная ATAPI-4 (ATA Packet Interface —пакетный интерфейс ATA), отличалась скоростью 33 Мбит/с. В версии ATAP1-5 она достигла 66 Мбит/с.

Настоящий прорыв был совершен в стандарте ATAPI-7. Вместо расширения разъема диска (и, соответственно, скорости передачи данных) появилась спецификация последовательного интерфейса ATA (**Serial ATA, SATA**), позволившего передавать через 7-контактный разъем информацию на скоростях от 150 Мбит/с (со временем скорость увеличилась до 1,5 Гбит/с). Замена 80-проводного плоского кабеля круглым кабелем диаметром в несколько миллиметров улучшила вентиляцию системного блока. Кроме того, при отправке сигналов через интерфейс SATA потребляется всего 0,5 В (в сравнении с 5 В по стандарту ATAPI-6), вследствие чего уменьшается общий уровень энергопотребления.

SCSI-диски

SCSI-диски с точки зрения расположения цилиндров, дорожек и секторов не отличаются от IDE-дисков, но они имеют другой интерфейс и более высокую скорость передачи данных. Поскольку у SCSI -дисков высокая скорость передачи данных, они используются во многих высокопроизводительных рабочих станциях и серверах, особенно в конфигурациях RAID (см. ниже).

SCSI – это не просто интерфейс жесткого диска. Это шина, к которой могут подсоединяться SCSI-контроллер и до семи дополнительных устройств. Ими могут быть один или несколько жестких SCSI-дисков, дисководы CD-ROM, сканеры, накопители на магнитной ленте и другие периферийные устройства.

Твердотельные накопители

Устройства на базе энергонезависимой флэш-памяти, часто называемые твердотельными накопителями или SSD-дисками (Solid State Disk), рассматриваются как высокоскоростная альтернатива традиционным технологиям магнитных дисков.

Так как SSD-диски по сути являются памятью, они обладают более высокой производительностью по сравнению с вращающимися магнитными дисками при нулевом времени поиска. Если типичный магнитный диск может

обращаться к данным со скоростью 100 Мбит/с, то у SSD эта скорость в два-три раза выше. И поскольку устройство не имеет подвижных частей, оно особенно хорошо подходит для ноутбуков (колебания и перемещения не влияют на его способность обращаться к данным). Недостатком SSD-устройств по сравнению с магнитными дисками является их стоимость.

Другой недостаток твердотельных дисков по сравнению с магнитными – их ресурс безотказной работы. Типичная флэш-ячейка перестает функционировать примерно через 100 000 операций перезаписи. Процесс инжекции электронов в плавающий затвор медленно повреждает его и окружающие изоляторы вплоть до окончательной потери работоспособности. Для повышения срока жизни SSD была разработана методика нивелирования износа, основанная на распределении записи по всем ячейкам на диске. Каждый раз, когда на устройство записывается новый блок, для записи выбирается блок, относительно давно не использовавшийся. Для этого во флэш-накопителе должна храниться карта блоков – одна из причин, по которой хранение данных на флэш-дисках сопряжено с относительно высокими внутренними затратами. Благодаря нивелированию износа флэш-диск сможет выдержать количество операций записи, равное максимальному количеству операций записи для одной ячейки, умноженному на количество блоков на диске.

Оптические диски

Оптические **CD-ROM** диски, которые изначально использовались для записи телевизионных программ, позже стали одними из основных средств хранения информации в компьютерной индустрии. Благодаря большой емкости и низкой цене оптические диски повсеместно применяются для распространения ПО, книг, фильмов и данных других типов, а также для создания архивных копий жестких дисков.

Со временем появились **DVD-диски** большей, чем CD-ROM емкости. Изначально аббревиатура DVD расшифровывалась как Digital Video Disk (цифровой видеодиск), сейчас она официально превратилась в Digital Versatile Disk (цифровой многоцелевой диск). DVD-диски в целом похожи на компакт-диски. Как и обычные компакт-диски, они имеют 120 мм в диаметре, создаются на основе поликарбоната и содержат лунки и площадки, которые освещаются лазерным диодом и считываются фотодетектором. Однако существует несколько технических отличий, основным из которых является использование при записи-считывании красного лазера.

Имеется четыре формата DVD-дисков:

1. Односторонние однослойные диски (4,7 Гбайт).
2. Односторонние двухслойные диски (8,5 Гбайт).

3. Двухсторонние однослойные диски (9,4 Гбайт).
4. Двухсторонние двухслойные диски (17 Гбайт).

Технология **Blu-Ray** предусматривает применение синего лазера вместо красного. Синий лазер отличается более короткой длиной волны, а значит, повышенной точностью; за счет этого обстоятельства он позволяет уменьшать размеры лунок и площадок. На односторонних дисках Blu-Ray умещается около 25 Гбайт данных; на двухсторонних – 50 Гбайт. Скорость передачи данных составляет 4,5 Мбит/с, что очень неплохо для оптических дисков, хотя по-прежнему несопоставимо с магнитными (напомним, стандарт ATAPI-6 предусматривает передачу данных на скорости 100 Мбит/с, а Ultra4 SCSI позволяет поднять скорость до 320 Мбит/с).

Контрольные вопросы по теме

Уровень модуля

Уровень курса

1. Основная и вспомогательная память. Пятиуровневая организация памяти.
2. Адреса основной памяти компьютера.
3. Кэш-память компьютера.
4. Основные сведения о магнитных дисках.
5. Стандарты подключения дисков.
6. Твердотельные накопители.
7. Оптические диски.

Лекция № 4

Тема: Общее устройство компьютера. Часть 3. Ввод-вывод.

Оглавление

Источники:

1. Таненбаум Э., Остин Т. Архитектура компьютера. 6-е изд. — СПб.: Питер, 2013. — 816 с.: ил.

   https://rutracker.org/forum/viewtopic.php?t=4956359

Шины

Как отмечалось, компьютерная система состоит из трех основных компонентов: центрального процессора, памяти (основной и вспомогательной) и устройств ввода-вывода (принтеров, сканеров, модемов). Обмен данными между этими компонентами производится посредством шин.

Общее представление о компьютерных шинах

Логическую структуру обычного персонального компьютера иллюстрирует рис. 3.8. У данного компьютера имеется одна шина для соединения центрального процессора, памяти и устройств ввода-вывода; однако большинство систем имеют две и более шин. Каждое устройство ввода-вывода состоит из двух частей: одна объединяет большую часть электроники и называется контроллером, а другая представляет собой само устройство ввода-вывода, например, жесткий диск. Контроллер обычно располагается на плате, которая вставляется в свободный разъем системной шины. Исключение представляют собой контроллеры устройств, являющихся неотъемлемыми составными частями компьютера (например, клавиатуры), которые иногда располагаются на материнской плате. Хотя дисплей (монитор) и нельзя назвать дополнительным устройством, соответствующий контроллер иногда располагается на встроенной плате, чтобы пользователь мог по желанию выбирать платы с графическими ускорителями или без них, устанавливать дополнительную намять и т. д. Контроллер связывается с самим устройством кабелем, который соединяется с разъемом на задней стороне корпуса.

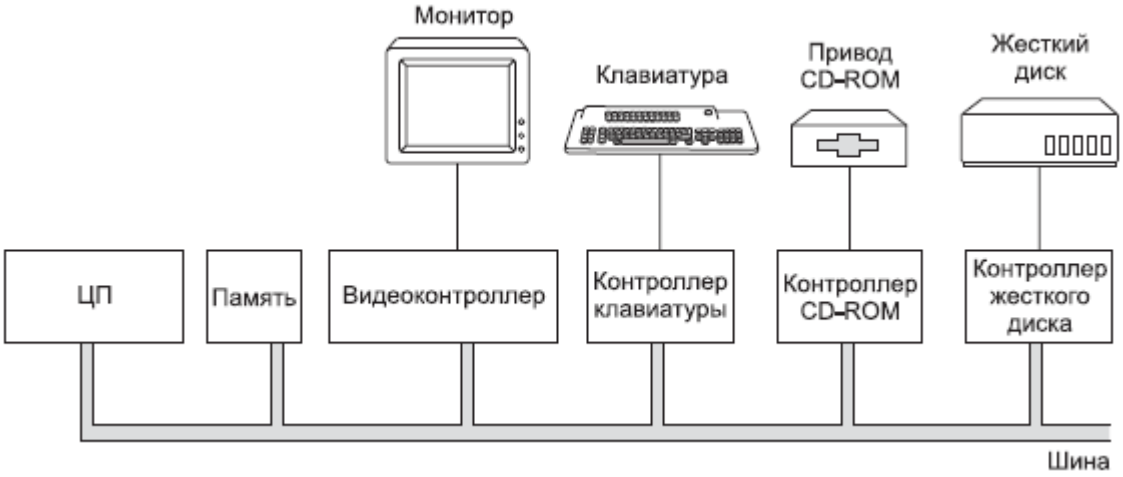


Рис. 3.8. Логическая структура обычного персонального компьютера

Контроллер управляет своим устройством ввода-вывода и для этого регулирует доступ к шине. Например, если программа запрашивает данные с диска, она посылает команду контроллеру диска, который затем отправляет диску команду поиска и другие команды. После нахождения соответствующей

дорожки и сектора диск начинает передавать контроллеру данные в виде потока битов. Задача контроллера состоит в том, чтобы разбить поток битов на фрагменты и записывать каждый такой фрагмент по мере накопления битов для него в память. Отдельный фрагмент обычно представляет собой одно или несколько слов. Если контроллер считывает данные из памяти или записывает их в память без участия центрального процессора, то говорят, что осуществляется прямой доступ к памяти (Direct Memory Access, DMA). Когда передача данных заканчивается, контроллер выдает прерывание, вынуждая центральный процессор приостановить работу текущей программы и начать выполнение особой процедуры. Эта процедура называется программой обработки прерываний и нужна она для того, чтобы проверить, нет ли ошибок, в случае их обнаружения произвести необходимые действия и сообщить операционной системе, что процесс ввода-вывода завершен. Когда программа обработки прерывания завершается, процессор возобновляет работу программы, которая была приостановлена в момент прерывания.

Шина используется не только контроллерами ввода-вывода, но и процессором для передачи команд и данных. Что происходит, если процессор и контроллер ввода-вывода хотят получить доступ к шине одновременно? В этом случае особая микросхема, которая называется арбитром шины, решает, чья очередь первая. Обычно предпочтение отдается устройствам ввода-вывода, поскольку работу дисков и других движущихся устройств нельзя прерывать, так как это может привести к потере данных. Когда ни одного устройства ввода-вывода не функционирует, центральный процессор может полностью распоряжаться шиной для взаимодействия с памятью. Однако если работает какое-нибудь устройство ввода-вывода, оно будет запрашивать доступ к шине и получать его каждый раз, когда ему это необходимо. Этот процесс, который притормаживает работу компьютера, называется захватом цикла памяти (cycle stealing).

Большинство персональных компьютеров и рабочих станций имеют физическую структуру, сходную с показанной на рис. 3.7. Обычно устройство представляет собой металлический корпус с большой интегральной схемой на дне, которая называется материнской (системной) платой. Материнская плата содержит микросхему процессора, несколько разъемов для модулей DIMM и различные вспомогательные микросхемы. Еще на материнской плате располагаются шина (она тянется вдоль платы) и несколько разъемов для подсоединения устройств ввода-вывода.

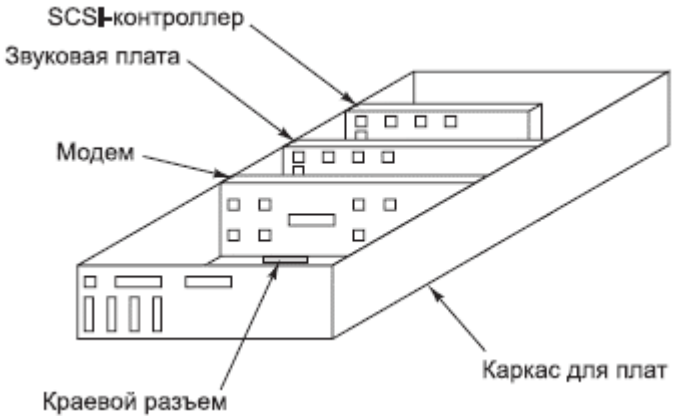


Рис. 3.7. Физическая структура персонального компьютера

Физически шина – это несколько проводников, соединяющих несколько устройств. Шины можно разделить на категории в соответствии с выполняемыми функциями. Они могут быть внутренними по отношению к процессору и служить для передачи данных в АЛУ и из АЛУ, а могут быть внешними по отношению к процессору и связывать процессор с памятью или устройствами ввода-вывода. Каждый тип шины обладает определенными свойствами и к каждому из них предъявляются определенные требования.

Первые персональные компьютеры имели одну внешнюю шину, которая называлась системной. Она состояла из нескольких медных проводов (от 50 до 100), которые встраивались в материнскую плату. На материнской плате на одинаковых расстояниях друг от друга находились разъемы для микросхем памяти и устройств ввода-вывода. Современные персональные компьютеры обычно содержат специальную шину между центральным процессором и памятью и, по крайней мере, еще одну шину для устройств ввода-вывода. На рис. 3.9 изображена система с одной шиной памяти и одной шиной ввода-вывода.

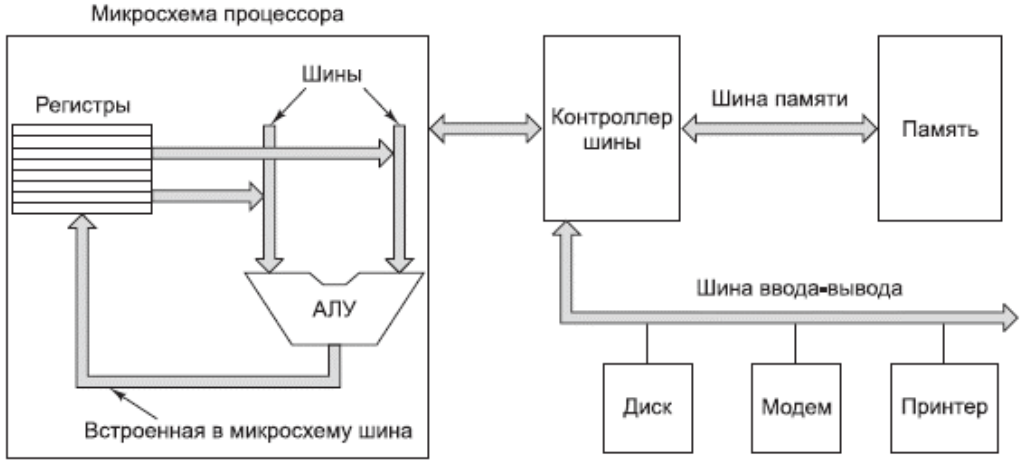


Рис. 3.9. Компьютерная система с несколькими шинами

В литературе шины обычно изображаются в виде жирных стрелок, как показано на этом рисунке. Разница между жирной стрелкой и нежирной стрелкой, через которую проходит короткая диагональная линия с указанием числа битов, небольшая. Когда тип всех битов одинаков, например, все адресные или все информационные, рисуется обычная стрелка. Когда включаются адресные линии, линии данных и управления, используется жирная стрелка.

Хотя разработчики процессоров могут использовать любой тип шины для микросхемы, должны быть введены четкие правила о том, как работает шина; и все устройства, связанные с шиной, должны подчиняться этим правилам, чтобы платы, которые выпускаются сторонними производителями, подходили к системной шине. Эти правила называются **протоколом** шины. Кроме того, должны существовать определенные технические требования, чтобы платы от сторонних производителей подходили к направляющим для печатных плат и имели разъемы, соответствующие материнской плате механически, с точки зрения напряжений, синхронизации и т. д.

Вкратце укажем, как работают шины. Некоторые устройства, соединенные с шиной, являются активными и могут инициировать передачу информации по шине, тогда как другие являются пассивными и ждут запросов. Активное устройство называется задающим, пассивное – подчиненным. Когда центральный процессор требует от контроллера диска считать или записать блок информации, центральный процессор действует как задающее устройство, а контроллер диска —как подчиненное. Контроллер диска может действовать как задающее устройство, когда он командует памяти принять слова, которые считал с диска. Память ни при каких обстоятельствах не может быть задающим устройством.

Как и процессор, шина имеет адресные, информационные линии и управляющие линии. Тем не менее между выводами процессора и сигналами шины может и не быть взаимно однозначного соответствия.

Существует целый ряд шин, широко используемых в компьютерном мире, например: Omnibus (PDP-8), Unibus (PDP-11), Multibus (8086), VME (оборудование для физической лаборатории), IBM PC (PC/XT), ISA (PC/AT), EISA (80386), MicroChannel (PS/2), Nubus (Macintosh), PCI (различные персональные компьютеры), SCSI (различные персональные компьютеры и рабочие станции), Universal Serial Bus (современные персональные компьютеры), FireWire (бытовая электроника).

Ширина шины. Шины ISA и EISA

Ширина (количество адресных линий) шины – самый очевидный параметр. Чем больше адресных линий содержит шина, тем к большему

объему памяти может обращаться процессор. Если шина содержит *n* адресных линий, тогда процессор может использовать ее для обращения к 2^n различным ячейкам памяти. Для памяти большой емкости необходимо много адресных линий. Проблема заключается в том, что для широких шин требуется больше проводов, чем для узких. Они занимают больше физического пространства (например, на материнской плате) и для них нужны разъемы большего размера. Все эти факторы делают шину дорогостоящей.

Многие разработчики систем оказались недальновидными, что привело к неприятным последствиям. Первая модель IBM PC содержала процессор 8088 и 20-разрядную адресную шину **ISA** (рис. 3.10). Шина позволяла обращаться к 1 Мбайт памяти.

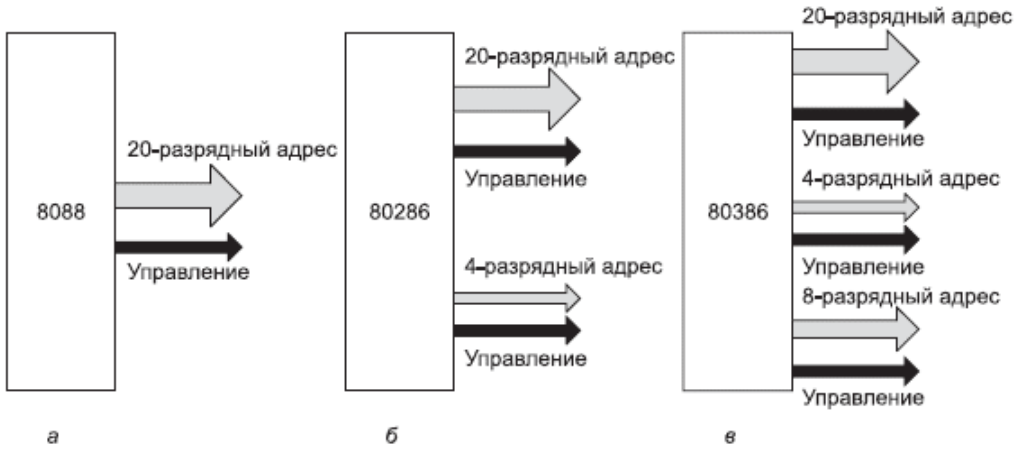


Рис. 3.10. Расширение адресной шины с течением времени

Когда появился следующий процессор (80286). компания Intel решила увеличить адресное пространство до 16 Мбайт, поэтому пришлось добавить еще 4 линии (не нарушая изначальные 20 по причинам совместимости с предыдущими версиями), как показано на рис. 3.10б. К сожалению, пришлось также добавить управляющие линии для новых адресных линий. Когда появился процессор 80386, было добавлено еще 8 адресных линий и, естественно, несколько управляющих линий, как показано на рис. 3.10 в. В результате получилась шина **EISA**. Однако архитектура получилась куда более запутанной, чем если бы с самого начала использовались 32 линии.

С течением времени увеличивается не только число адресных линий, но и число информационных линий, хотя это происходит по другой причине. Можно увеличить пропускную способность шины двумя способами: сократить время цикла шины (сделать большее количество передач в секунду) или увеличить ширину шины данных (то есть увеличить количество битов, передаваемых за цикл). Можно повысить скорость работы шины, но сделать это довольно сложно, поскольку сигналы на разных линиях передаются с

разной скоростью (это явление называется расфазировкой шины). Чем быстрее работает шина, тем больше расфазировка.

При увеличении скорости работы шины возникает еще одна проблема: в этом случае она становится несовместимой с предыдущими версиями. Старые платы, разработанные для более медленной шины, не могут работать с новой. Такая ситуация невыгодна для владельцев и производителей старых плат. Поэтому обычно для увеличения производительности просто добавляются новые линии, как показано на рис. 3.10. В этом тоже есть свои недостатки. Компьютер IBM PC и его преемники, например, начали с 8 информационных линий, затем перешли к 16, потом —к 32 линиям, и все это в одной и той же шине.

Чтобы обойти эту проблему, разработчики иногда отдают предпочтение **мультиплексной** шине. В этой шине нет разделения на адресные и информационные линии. В ней может быть, например, 32 линии и для адресов, и для данных. Сначала эти линии используются для адресов, затем —-для данных. Чтобы записать информацию в память, нужно сначала передавать в память адрес, а потом – данные. В случае с отдельными линиями адреса и данные могут передаваться вместе. Объединение линий сокращает ширину и стоимость шины, но система работает при этом медленнее.

Шина PCI

Peripheral Component Interconnect (PCI) является текущем стандартом для карт расширений персональных компьютеров. Intel разработала эту технологию в 1993 году для процессора Pentium. С помощью этой шины соединяется процессор с памятью и другими периферийными устройствами.

PCI поддерживает передачу 32 и 64 разрядных данных, количество передаваемых данных равно разрядности процессора, 32 битный процессор будет использовать 32 битную шину, а 64 битный — 64 битную. Работает шина на частоте 33 МГц. Большинство PCI карт работают на напряжении 5 вольт.

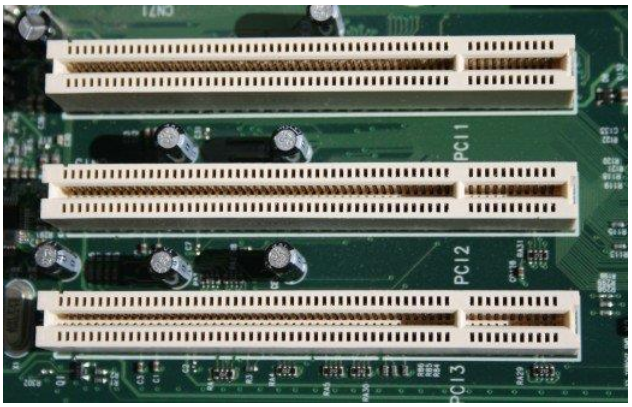


Рис. 3.11. Внешний вид разъемов PCI

Существует много различных конфигураций шины PCI. Наиболее типичная из них показана на рис. 3.12. В такой конфигурации центральный процессор взаимодействует с контроллером памяти по выделенному высокоскоростному соединению. Таким образом, контроллер соединяется с памятью непосредственно, то есть передача данных между центральным процессором и памятью происходит не через шину PCI. Другие периферийные устройства подсоединяются прямо к шине PCI. Машина такого типа обычно содержит 2 или 3 пустых разъема PCI, чтобы покупатели имели возможность подключать карты PCI для новых периферийных устройств).

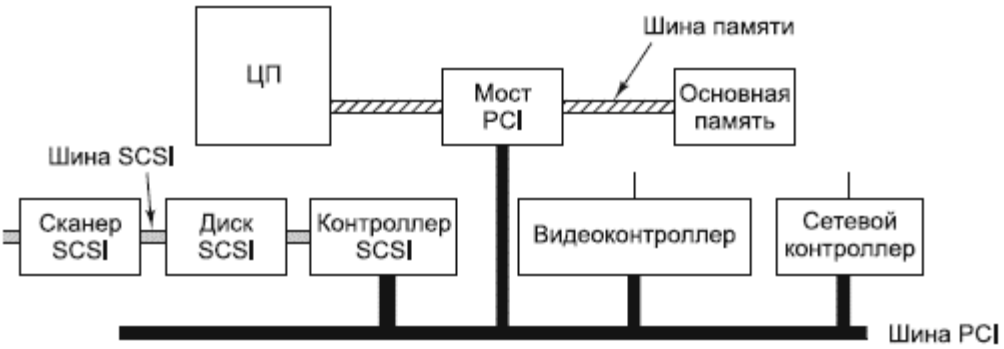


Рис. 3.12. Современный персональный компьютер с шиной PCI. Контроллер SCSI является PCI -устройством

Шина PCI Exspress

Дальнейшее развитие в направлении повышения скорости привело к тому, что PCI была заменена шиной **PCI Exspress** (сокращенно **PCIe**). Многие современные компьютеры поддерживают обе шины, благодаря чему пользователи могут подключать новые, быстрые устройства к шине PCIe, а старые, более медленные – к шине PCI.

Если шина PCI представляла собой обновленную версию старой шины ISA с более высокой скоростью и разрядностью параллельно передаваемых данных, PCIe представляет кардинальное изменение по сравнению с шиной PCI. Собственно, это вообще не шина, а одноранговая сеть, использующая разрядно-последовательные линии и коммутацию пакетов. У нее больше от Интернета, чем от традиционных шин. Архитектура PCIe изображена на рис. 3.13.

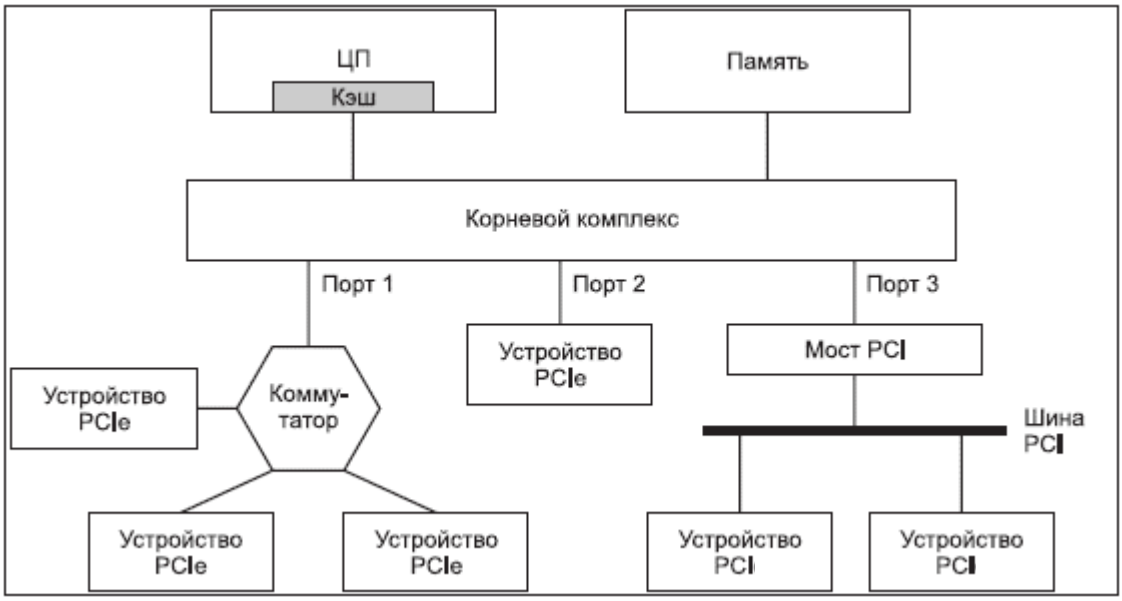


Рис. 2.28. Архитектура системы PCIe с тремя портами PCI

Соединения между устройствами являются последовательными, то есть имеют разрядность в один бит вместо 8, 16, 32 или 64 бит. Хотя, казалось бы, 64-разрядное соединение обладает более высокой пропускной способностью, на практике различия во времени распространения 64-разрядной информации, называемые расфазировкой, заставляют использовать относительно низкие скорости передачи данных. По последовательному соединению данные передаются на значительно более высокой скорости, что более чем компенсирует потерю параллелизма. Шины PCI работают на максимальной тактовой частоте 66 МГц. При передаче 64 бит за такт скорость передачи данных составляет 528 Мбайт/с. При тактовой частоте 8 Гбит/с, даже в случае последовательной передачи, скорость передачи по шине PCIe составляет 1 Гбайт/с. Кроме того, обмен данными между устройством и корневым комплексом или коммутатором не ограничивается одной проводной парой. Устройство может иметь до 32 проводных пар, называемых трактами (lanes) или дорожками. Тракты работают несинхронно, поэтому расфазировка в данном случае несущественна. На большинстве материнских плат имеется 16-трактовый разъем для графической карты, что для PCIe 3.0 обеспечивает пропускную способность в 16 Гбайт/с – примерно в 30 раз больше, чем у графических карт PCI. Такая пропускная способность необходима для приложений, требования которых постоянно растут – например, трехмерной графики.

Видеопамять

Обновление картинки на экранах мониторов производится от 60 до 100 раз в секунду; для этого используется видеопамять, размещенная на плате контроллера дисплея. Видеопамять содержит одну или несколько битовых

карт, представляющих выводимое на экран изображение. Если, скажем, на экране умещается 1920 x 1080 элементов изображения (пикселов), значит, в видеопамяти содержится 1920 x 1080 значений, по одному на каждый пиксел. В целях быстрого переключения с одного изображения на другое в памяти может размещаться несколько таких карт.

В современных дисплеях каждый пиксел представлен 3-байтным значением RGB, которое определяет интенсивность красного (Red), зеленого (Green) и синего (Blue) компонентов изображения (мощные профессиональные мониторы используют 10 и более бит на цвет). Как известно, любой цвет можно представить путем линейной суперпозиции трех упомянутых базовых цветов.

Если в видеопамяти хранится информация о 1920 x 1080 пикселах, причем на каждый из них выделяется по 3 байта, общий объем этих данных составляет около 6,2 Мбайт; поэтому на любые манипуляции таким изображением уходит довольно много процессорного времени. Для вывода растровых (то есть сформированных на основе битовых карт) изображений требуется большая пропускная способность. К примеру, для воспроизведения одного кадра полноцветных мультимедийных данных в полноэкранном формате на дисплее размером 1920 x 1080 необходимо скопировать в видеопамять 6,2 Мбайт. Если учесть, что полноценный видеофильм выводится со скоростью 25 кадров в секунду, общая скорость передачи данных должна составлять 155 Мбайт/с. Такую пропускную способность не способна обеспечить даже первоначальная версия шины PCI (132 Мбайт/с), но шина PCIe легко справляется с ней.

Контрольные вопросы по теме

Уровень модуля

Уровень курса

1. Общее представление о компьютерных шинах.
2. Ширина шины. Шины ISA и EISA.
3. Шина PCI.
4. Шина PCI Express.
5. Видеопамять.

Лекция № 5

Тема: Локальные сети

Оглавление

Источники:

1. Таненбаум Э., Остин Т. Архитектура компьютера. 6-е изд. — СПб.: Питер, 2013. — 816 с.: ил.

   https://rutracker.org/forum/viewtopic.php?t=4956359

Введение в вычислительные сети

Все вычислительные сети можно классифицировать по ряду признаков.

В зависимости от расстояний между ПК различают следующие вычислительные сети:

- локальные вычислительные сети – ЛВС (LAN – Local Area Networks) – компьютерные сети, расположенные в пределах небольшой ограниченной территории (здании или в соседних зданиях) не более 10 – 15 км;
- территориальные вычислительные сети, которые охватывают значительное географическое пространство. К территориальным сетям можно отнести городские (MAN - Metropolitan Area Network), региональные (Regional computer network), национальные (National computer network) и глобальные (WAN - Wide Area Network) сети. Городские и региональные сети связывают абонентов района, города или области. Глобальные сети объединяют абонентов, удаленных между собой на значительное расстояние, находящихся в различных странах или континентах;

В настоящее время нашли широкое применение локальные вычислительные сети, основное назначение которых обеспечить доступ к разделяемым или сетевым (общим, то есть совместно используемым) ресурсам, данным и программам.

Локальные вычислительные сети обеспечивают:

1. Распределение данных (Data Sharing). Данные в ЛВС хранятся на центральном ПК и могут быть доступны на рабочих станциях, поэтому на каждом рабочем месте не надо иметь накопители для хранения одной и той же информации.

2. Распределение информационных и технических ресурсов (Resource Sharing):

- логические диски и другие внешние запоминающие устройства (накопители на CD-ROM, DVD, ZIP и так далее);
- каталоги (папки) и содержащиеся в них файлы;
- подключенные к ПК устройства: принтеры, модемы и другие внешние устройства (позволяет экономно использовать ресурсы, например, печатающие устройства, модемы).

3. Распределение программ (Software Sharing). Все пользователи локальных вычислительных сетей могут совместно иметь доступ к программам (сетевым версиям), которые централизованно устанавливаются в сети.

4. Обмен сообщениями по электронной почте (Electronic Mail). Все пользователи сети могут оперативно обмениваться информацией между собой посредством передачи сообщений.

Локальные вычислительные сети, в зависимости от способов взаимодействия компьютеров в них, можно разделить на централизованные и одноранговые сети. Централизованные локальные сети строятся на основе архитектуры "клиент-сервер", которая предполагает выделение в сети "серверов" и "клиентов". Одноранговые ЛВС основаны на равноправной (peer-to-peer) модели взаимодействия компьютеров, в которой каждый компьютер может быть как сервером, так и клиентом.

Локальные вычислительные сети могут отличаться архитектурой (Ethernet, Token Ring, FDDI и т.д.) и топологией (шинная, кольцевая, “звезда”).

Способы соединения двух компьютеров для совместного использования файлов

Компьютерные сети - это совокупность узлов, объединенных каналами передачи данных. Компьютерные сети также называются информационно - вычислительными сетями, вычислительными сетями или сетями передачи данных.

Рассмотрим способы соединения двух компьютеров, которые используются, как правило, для создания временных сетей компьютер-компьютер с целью совместного использования файлов. В этом случае компьютеры соединяются непосредственно друг с другом, без помощи дополнительного коммуникационного оборудования.

Для обмена данными между двумя компьютерами (например, настольными и переносными ПК) можно использовать различные способы их соединения:

- прямое соединения компьютеров через последовательные и параллельные порты (COM, USB, LPT, IrDA, Bluetooth);
- удаленное соединение двух компьютеров через модемы
- соединение двух компьютеров в локальную сеть, используя сетевые карты и проводные линии связи;
- соединения двух компьютеров в локальную сеть, используя встроенные беспроводные интерфейсы Wi-Fi.

Линии и каналы передачи данных в сетях

Для построения компьютерных сетей применяются линии связи, использующие различную физическую среду. В качестве физической среды в коммуникациях используются: металлы (в основном медь), сверхпрозрачное стекло (кварц) или пластик и эфир. Физическая среда передачи данных может

представлять собой кабель "витая пара", коаксиальные кабель, волоконно-оптический кабель и окружающее пространство.

Линии связи или линии передачи данных – это промежуточная аппаратура и физическая среда, по которой передаются информационные сигналы (данные). В одной линии связи можно образовать несколько каналов связи (виртуальных или логических каналов), например, путем частотного или временного разделения каналов. ***Канал связи*** – это средство односторонней передачи данных. Если линия связи монопольно используется каналом связи, то в этом случае линию связи называют каналом связи.

Канал передачи данных – это средства двухстороннего обмена данными, которые включают в себя линии связи и аппаратуру передачи (приема) данных. Каналы передачи данных связывают между собой источники информации и приемники информации.

В зависимости от физической среды передачи данных линии связи можно разделить на:

- проводные линии связи без изолирующих и экранирующих оплеток;
- кабельные, где для передачи сигналов используются такие линии связи как кабели "витая пара", коаксиальные кабели или оптоволоконные кабели;
- беспроводные (радиоканалы наземной и спутниковой связи), использующие для передачи сигналов электромагнитные волны, которые распространяются по эфиру.

Проводные линии связи

Проводные (воздушные) линии связи используются для передачи телефонных и телеграфных сигналом, а также для передачи компьютерных данных. Эти линии связи применяются в качестве магистральных линий связи.

По проводным линиям связи могут быть организованы аналоговые и цифровые каналы передачи данных. Скорость передачи по проводным линиям "простой старой телефонной линии" (POST - Primitive Old Telephone System) является очень низкой. Кроме того, к недостаткам этих линий относятся помехозащищенность и возможность простого несанкционированного подключения к сети.

Кабельные линии связи

Кабельные линии связи имеют довольно сложную структуру. Кабель состоит из проводников, заключенных в несколько слоев изоляции. В компьютерных сетях используются три типа кабелей.

Витая пара (twisted pair) — кабель связи, который представляет собой витую пару медных проводов (или несколько пар проводов), заключенных в

экранированную оболочку. Пары проводов скручиваются между собой с целью уменьшения наводок. Витая пара является достаточно помехоустойчивой. Существует два типа этого кабеля: неэкранированная витая пара UTP и экранированная витая пара STP.

Характерным для этого кабеля является простота монтажа. Данный кабель является самым дешевым и распространенным видом связи, который нашел широкое применение в самых распространенных локальных сетях с архитектурой Ethernet, построенных по топологии типа “звезда”. Кабель подключается к сетевым устройствам при помощи соединителя RJ45.

Кабель используется для передачи данных на скорости 10 Мбит/с, 100 Мбит/с и 1 Гбит/с. Витая пара обычно используется для связи на расстояние не более нескольких сот метров. К недостаткам кабеля "витая пара" можно отнести возможность простого несанкционированного подключения к сети.

Коаксиальный кабель (coaxial cable) - это кабель с центральным медным проводом, который окружен слоем изолирующего материала для того, чтобы отделить центральный проводник от внешнего проводящего экрана (медной оплетки или слой алюминиевой фольги). Внешний проводящий экран кабеля покрывается изоляцией.

Существует два типа коаксиального кабеля: тонкий коаксиальный кабель диаметром 5 мм и толстый коаксиальный кабель диаметром 10 мм. У толстого коаксиального кабеля затухание меньше, чем у тонкого. Стоимость коаксиального кабеля выше стоимости витой пары и выполнение монтажа сети сложнее, чем витой парой. Коаксиальный кабель применяется, например, в локальных сетях с архитектурой Ethernet, построенных по топологии типа “общая шина”.

Коаксиальный кабель более помехозащищенный, чем витая пара и снижает собственное излучение. Пропускная способность – 50-100 Мбит/с. Допустимая длина линии связи – несколько километров. Несанкционированное подключение к коаксиальному кабелю сложнее, чем к витой паре.

Кабельные оптоволоконные каналы связи. Оптоволоконный кабель (fiber optic) – это оптическое волокно на кремниевой или пластмассовой основе, заключенное в материал с низким коэффициентом преломления света, который закрыт внешней оболочкой.

Оптическое волокно передает сигналы только в одном направлении, поэтому кабель состоит из двух волокон. На передающем конце оптоволоконного кабеля требуется преобразование электрического сигнала в световой, а на приемном конце обратное преобразование.

Основное преимущество этого типа кабеля – чрезвычайно высокий уровень помехозащищенности и отсутствие излучения. Несанкционированное

подключение очень сложно. Скорость передачи данных исчисляется гигабайтами в секунду. Основные недостатки оптоволоконного кабеля – это сложность его монтажа, небольшая механическая прочность и чувствительность к ионизирующим излучениям.

Беспроводные (радиоканалы наземной и спутниковой связи) каналы передачи данных

Радиоканалы наземной (радиорелейной и сотовой) и спутниковой связи образуются с помощью передатчика и приемника радиоволн и относятся к технологии беспроводной передачи данных.

Радиорелейные каналы передачи данных

Радиорелейные каналы связи состоят из последовательности станций, являющихся ретрансляторами. Связь осуществляется в пределах прямой видимости, дальности между соседними станциями - до 50 км. Цифровые радиорелейные линии связи (ЦРРС) применяются в качестве региональных и местных систем связи и передачи данных, а также для связи между базовыми станциями сотовой связи.

Спутниковые каналы передачи данных

В спутниковых системах используются антенны СВЧ-диапазона частот для приема радиосигналов от наземных станций и ретрансляции этих сигналов обратно на наземные станции. В спутниковых сетях используются три основных типа спутников, которые находятся на геостационарных орбитах, средних или низких орбитах. Спутники запускаются, как правило, группами. Разнесенные друг от друга они могут обеспечить охват почти всей поверхности Земли. Работа спутникового канала передачи данных представлена на рисунке.

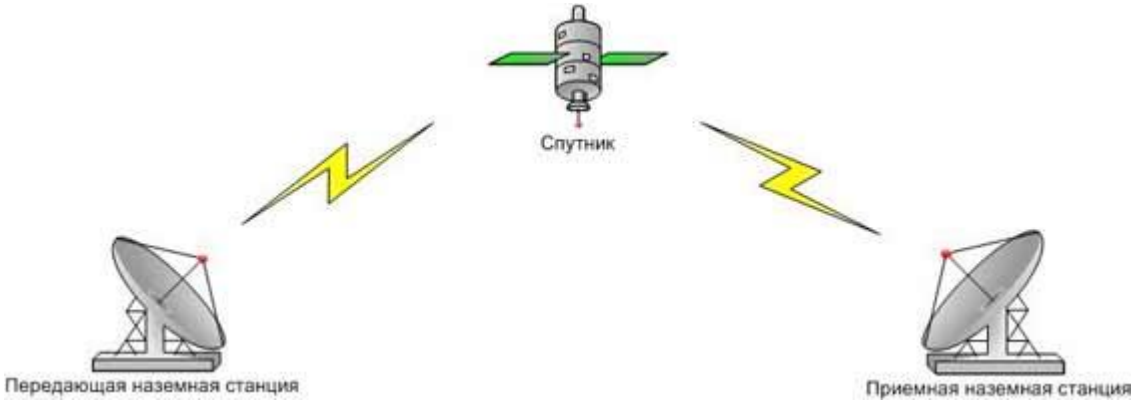


Рис. 1. Спутниковые каналы передачи данных

Целесообразнее использовать спутниковую связь для организации канала связи между станциями, расположенными на очень больших

расстояниях, и возможности обслуживания абонентов в самых труднодоступных точках. Пропускная способность высокая – несколько десятков Мбит/с.

Сотовые каналы передачи данных

Радиоканалы сотовой связи строятся по тем же принципам, что и сотовые телефонные сети. Сотовая связь - это беспроводная телекоммуникационная система, состоящая из сети наземных базовых приемо-передающих станций и сотового коммутатора (или центра коммутации мобильной связи).

Базовые станции подключаются к центру коммутации, который обеспечивает связь, как между базовыми станциями, так и с другими телефонными сетями и с глобальной сетью Интернет. По выполняемым функциям центр коммутации аналогичен обычной АТС проводной связи.

Радиоканалы передачи данных для локальных сетей

Стандартом беспроводной связи для локальных сетей является технология Wi-Fi. Wi-Fi обеспечивает подключение в двух режимах: точка-точка (для подключения двух ПК) и инфраструктурное соединение (для подключения несколько ПК к одной точке доступа). Скорость обмена данными до 300 Мбит/с при инфраструктурном соединении.

Радиоканалы передачи данных **Bluetooth** - это технология передачи данных на короткие расстояния (не более 10 м) и может быть использована для создания домашних сетей. Скорость передачи данных не превышает 2,5 Мбит/с.

Способы передачи данных

Методы передачи на физическом уровне

Пересылка данных в вычислительных сетях от одного компьютера к другому осуществляется последовательно, бит за битом. Физически биты данных передаются по каналам передачи данных в виде аналоговых или цифровых сигналов. Совокупность средств (линий связи, аппаратуры передачи и приема данных), служащая для передачи данных в вычислительных сетях, называется каналом передачи данных.

В зависимости от формы передаваемой информации каналы передачи данных можно разделить на аналоговые (непрерывные) и цифровые (дискретные).

Так как аппаратура передачи и приема данных работает с данными в дискретном виде (т.е. единицам и нулям данных соответствуют дискретные

электрические сигналы), то при их передаче через аналоговый канал требуется преобразование дискретных данных в аналоговые (модуляция).

При приеме таких аналоговых данных необходимо обратное преобразование – демодуляция. Модуляция/демодуляция – процессы преобразования цифровой информации в аналоговые сигналы и наоборот. При модуляции информация представляется синусоидальным сигналом той частоты, которую хорошо передает канал передачи данных.

К способам модуляции относятся:

- амплитудная модуляция;
- частотная модуляция;
- фазовая модуляция.

При передаче дискретных сигналов через цифровой канал передачи данных используется кодирование:

- потенциальное;
- импульсное.

Таким образом, потенциальное или импульсное кодирование применяется на каналах высокого качества, а модуляция на основе синусоидальных сигналов предпочтительнее в тех случаях, когда канал вносит сильные искажения в передаваемые сигналы.

Обычно модуляция используется в глобальных сетях при передаче данных через аналоговые телефонные каналы связи, которые были разработаны для передачи голоса в аналоговой форме и поэтому плохо подходят для непосредственной передачи импульсов.

В зависимости от способов синхронизации каналы передачи данных вычислительных сетей можно разделить на *синхронные* и *асинхронные*. Синхронизация необходима для того, чтобы передающий узел данных мог передать какой-то сигнал принимающему узлу, чтобы принимающий узел знал, когда начать прием поступающих данных.

Синхронная передача данных требует дополнительной линии связи для передачи синхронизирующих импульсов. Передача битов передающей станцией и их прием принимающей станцией осуществляется в моменты появления синхроимпульсов.

При асинхронной передаче данных дополнительной линии связи не требуется. В этом случае передача данных осуществляется блоками фиксированной длины (байтами). Синхронизация осуществляется дополнительными битами (старт-битами и стоп-битами), которые передаются перед передаваемым байтом и после него.

При обмене данными между узлами вычислительных сетей используются три метода передачи данных:

- симплексная (однонаправленная) передача (телевидение, радио);

- полудуплексная (прием/передача информации осуществляется поочередно);
- дуплексная (двунаправленная), каждый узел одновременно передает и принимает данные (например, переговоры по телефону).

Методы передачи на канальном уровне

Прежде чем послать данные в вычислительную сеть, посылающий узел данных разбивает их на небольшие блоки, называемые пакетами данных. На узле–получателе пакеты накапливаются и выстраиваются в должном порядке для восстановления исходного вида.

В составе любого пакета должна присутствовать следующая информация:

1) данные или информация, предназначенная для передачи по сети;
2) адрес, указывающий место назначения пакета. Каждый узел сети имеет адрес. Кроме того, адрес имеет и приложение. Адрес приложения необходим для того, чтобы идентифицировать, какому именно приложению принадлежит пакет;
3) управляющие коды – это информация, которая описывает размер и тип пакета. Управляющие коды включают в себя также коды проверки ошибок и другую информацию.

Существует три принципиально различные схемы коммутации в вычислительных сетях:

- коммутация каналов;
- коммутация пакетов;
- коммутация сообщений.

При коммутации каналов устанавливается соединение между передающей и принимающей стороной в виде непрерывного составного физического канала из последовательно соединенных отдельных канальных участков для прямой передачи данных между узлами. Затем сообщение передается по образованному каналу.

Коммутация сообщений – процесс пересылки данных, включающий прием, хранение, выбор исходного направления и дальнейшую передачу блоков сообщений (без разбивки на пакеты). При коммутации сообщений блоки сообщений передаются последовательно от одного промежуточного узла к другому с временной буферизацией их на дисках каждого узла, пока не достигнут адресата. При этом новая передача может начаться только после того, как весь блок будет принят. Ошибка при передаче повлечет новую повторную передачу всего блока.

Передача пакетов осуществляется аналогично передаче сообщений, но так как размер пакета значительно меньше блока сообщения, то достигается быстрая его обработка промежуточным коммуникационным оборудованием. Поэтому канал передачи данных занят только во время передачи пакета и по ее завершению освобождается для передачи других пакетов. Шлюзы и маршрутизаторы, принимают пакеты от конечных узлов и на основании адресной информации передают их друг другу, а в конечном итоге станции назначения. Данный вид передачи данных является стандартом для сети Интернет.

Открытые системы и модель OSI

Протоколы, интерфейсы, стеки протоколов

Компьютерные сети, как правило, состоят из различного оборудования разных производителей, и без принятия всеми производителями общепринятых правил построения ПК и сетевого оборудования, обеспечить нормальное функционирование сетей было бы невозможно.

То есть для обеспечения нормального взаимодействия этого оборудования в сетях необходим единый унифицированный стандарт, который определял бы алгоритм передачи информации в сетях. В современных вычислительных сетях роль такого стандарта выполняют сетевые протоколы.

В связи с тем, что описать единым протоколом взаимодействия между устройствами в сети не представляется возможным, то необходимо разделить процесс сетевого взаимодействия на ряд концептуальных уровней (модулей) и определить функции для каждого модуля и порядок их взаимодействия, применив метод декомпозиции.

Используется многоуровневый подход метода декомпозиции, в соответствии с которым множество модулей, решающих частные задачи, упорядочивают по уровням, образующим иерархию, процесс сетевого взаимодействия можем представить в виде иерархически организованного множества модулей.

Многоуровневое представление средств сетевого взаимодействия имеет свою специфику, связанную с тем, что в процессе обмена сообщениями участвуют две стороны, то есть необходимо организовать согласованную работу двух иерархий, работающих на разных компьютерах.

Оба участника сетевого обмена должны принять множество соглашений. Соглашения должны быть приняты для всех уровней, начиная от самого низкого – уровня передачи битов – до самого высокого, реализующего сервис для пользователя. Декомпозиция предполагает четкое определение функции каждого уровня и интерфейсов между уровнями.

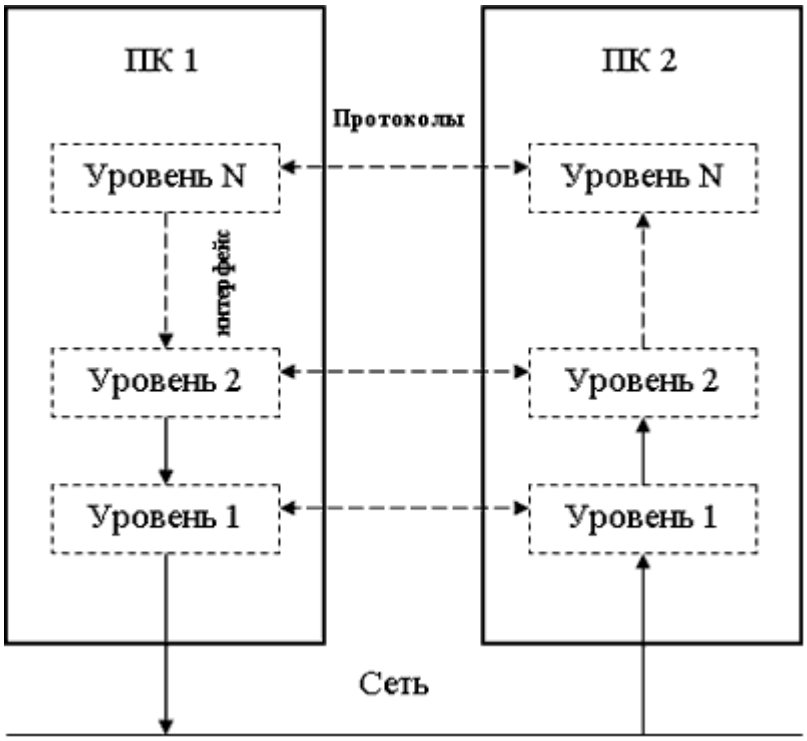


Рис. 2. Сетевые уровни

Взаимодействие одноименных функциональных уровней по горизонтали осуществляется посредством протокола. Протоколом называется набор правил и методов взаимодействия одноименных функциональных уровней объектов сетевого обмена.

Взаимодействия функциональных уровней по вертикали осуществляется через интерфейсы. Интерфейс определяет набор функций, которые нижележащий уровень предоставляет вышележащему уровню.

Таким образом, механизм передачи какого-либо пакета информации через сеть от клиентской программы, работающей на одном компьютере ПК 1, к клиентской программе, работающей на другом компьютере ПК 2, можно условно представить в виде последовательной пересылки этого пакета сверху вниз от верхнего уровня, обеспечивающего взаимодействие с пользовательским приложением, к нижнему уровню, организующему интерфейс с сетью, его трансляции на компьютер ПК 2 и обратной передачи верхнему уровню уже на ПК 2.

Коммуникационные протоколы могут быть реализованы как программно, так и аппаратно. Протоколы нижних уровней часто реализуются комбинацией программных и аппаратных средств, а протоколы верхних уровней – как правило, чисто программными средствами. Протоколы реализуются не только компьютерами, но и другими сетевыми устройствами – концентраторами, мостами, коммутаторами, маршрутизаторами и т.д. В

зависимости от типа устройств в нем должны быть встроенные средства, реализующие тот или иной набор протоколов.

Иерархически организованный набор протоколов, достаточный для организации взаимодействия узлов в сети, называется стеком коммуникационных протоколов. В сети Интернет базовым набором протоколов является стек протоколов TCP/IP.

Семиуровневая эталонная модель OSI. Модель OSI-ISO

Эталонная модель OSI (Open System Interconnection - OSI), разработанная в 1984 году Международной организацией по стандартизации (International Organization of Standardization – ISO), является определяющим документом концепции разработки открытых стандартов для организации соединения систем. Открытая система - система, доступная для взаимодействия с другими системами в соответствии с принятыми стандартами.

Семиуровневая эталонная модель “Взаимосвязь открытых систем” была разработана с целью упрощения взаимодействия устройств в сетях.

Семиуровневая эталонная модель представляет собой рекомендации (разработчикам сетей и протоколов) для построения стандартов совместимых сетевых программных продуктов, и служит базой для производителей при разработке совместимого сетевого оборудования. Рекомендации стандарта должны быть реализованы как в аппаратуре, так и в программных средствах вычислительных сетей.

Семиуровневая эталонная модель OSI определяет семь уровней взаимодействия систем в сетях с коммуникацией пакетов, дает им стандартные имена и указывает, какие функции должен выполнять каждый уровень. Каждый уровень функционирует независимо от выше - и нижележащих уровней. Каждый уровень может общаться с непосредственным соседним уровнем, однако он полностью изолирован от прямого обращения к следующим уровням.

Семиуровневая эталонная модель OSI описывает только системные средства взаимодействия, реализуемые операционной системой, системными утилитами, системными аппаратными средствами.

В соответствии с семиуровневой эталонной моделью сетевая система представляется прикладными процессами и процессами взаимодействия абонентов. Последние разбиваются на семь функциональных уровней:

1) прикладной,
2) представительный (уровень представления данных),
3) сеансовый,
4) транспортный,

5) сетевой,
6) канальный
7) физический.

При запуске на компьютере любого приложения, для функционирования которого требуется диалог с сетью, это приложение вызывает соответствующий протокол прикладного уровня сетевого программного обеспечения. Прикладной уровень формирует сообщение стандартного формата. Обычно сообщение состоит из заголовка и поля данных.

Заголовок содержит служебную информацию, которую необходимо передать через сеть прикладному уровню компьютера-адресата, чтобы сообщить ему, какую работу надо выполнить. Поле данных содержит данные, необходимые для выполнения этой работы.

После формирования сообщения прикладной уровень направляет его вниз к представительному уровню. Представительный уровень, на основании информации, полученной из заголовка прикладного уровня, выполняет требуемые действия и добавляет к сообщению собственную служебную информацию – заголовок представительного уровня, в котором содержатся указания для представительного уровня машины-адресата.

Полученное сообщение отправляется вниз сеансовому уровню, который добавляет свой заголовок и т.д. до физического уровня, который передает сформированное сообщение по линиям связи. К этому моменту сообщение имеет заголовки всех уровней.

Когда сообщение поступает в компьютер - адресат, оно принимается физическим уровнем и последовательно передается вверх с уровня на уровень. Причем каждый уровень анализирует и обрабатывает заголовок своего уровня, выполняя соответствующие данному уровню функции, а затем удаляет этот заголовок и передает сообщение вышележащему уровню.

Сетевые топологии

Все компьютеры в локальной сети соединены линиями связи. Геометрическое расположение линий связи относительно узлов сети и физическое подключение узлов к сети называется физической топологией. В зависимости от топологии различают сети: шинной, кольцевой, звездной, иерархической и произвольной структуры.

Различают физическую и логическую топологию. Логическая и физическая топологии сети независимы друг от друга. Физическая топология - это геометрия построения сети, а логическая топология определяет направления потоков данных между узлами сети и способы передачи данных.

В настоящее время в локальных сетях используются следующие физические топологии:

- физическая "шина" (bus);
- физическая “звезда” (star);
- физическое “кольцо” (ring);
- физическая "звезда" и логическое "кольцо" (Token Ring).

Шинная топология

Сети с шинной топологией используют линейный моноканал (коаксиальный кабель) передачи данных, на концах которого устанавливаются оконечные сопротивления (терминаторы). Каждый компьютер подключается к коаксиальному кабелю с помощью Т-разъема (Т - коннектор). Данные от передающего узла сети передаются по шине в обе стороны, отражаясь от оконечных терминаторов. Терминаторы предотвращают отражение сигналов, т.е. используются для гашения сигналов, которые достигают концов канала передачи данных.

Таким образом, информация поступает на все узлы, но принимается только тем узлом, которому она предназначается. В топологии логическая шина среда передачи данных используются совместно и одновременно всеми ПК сети, а сигналы от ПК распространяются одновременно во все направления по среде передачи. Так как передача сигналов в топологии физическая шина является широковещательной, т.е. сигналы распространяются одновременно во все направления, то логическая топология данной локальной сети является логической шиной.

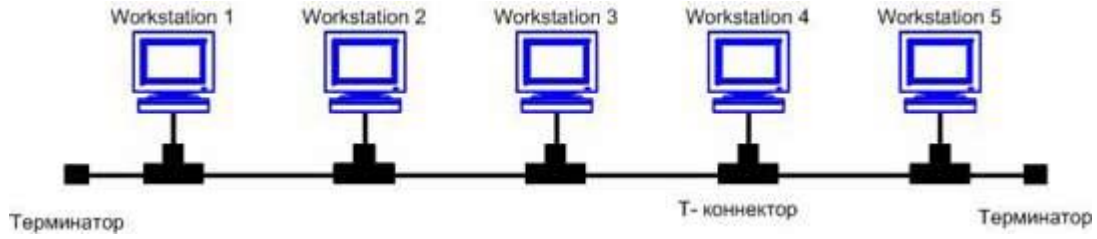


Рис. 3. Шинная топология

Данная топология применяется в локальных сетях с архитектурой Ethernet (классы 10Base-5 и 10Base-2 для толстого и тонкого коаксиального кабеля соответственно).

Преимущества сетей шинной топологии:

- отказ одного из узлов не влияет на работу сети в целом;
- сеть легко настраивать и конфигурировать;
- сеть устойчива к неисправностям отдельных узлов.

Недостатки сетей шинной топологии:

- разрыв кабеля может повлиять на работу всей сети;
- ограниченная длина кабеля и количество рабочих станций;
- трудно определить дефекты соединений.

Топология типа “звезда”

В сети построенной по топологии типа “звезда” каждая рабочая станция подсоединяется кабелем (витой парой) к концентратору или хабу (hub). Концентратор обеспечивает параллельное соединение ПК и, таким образом, все компьютеры, подключенные к сети, могут общаться друг с другом.

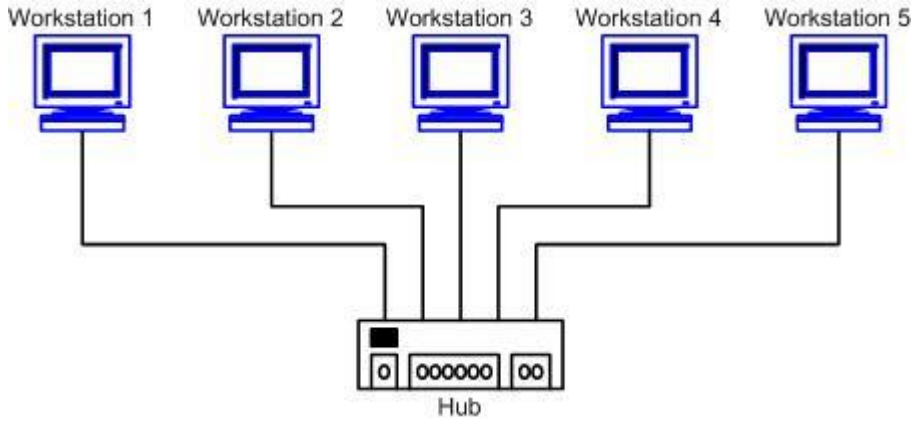


Рис. 4. Топология "звезда"

Данные от передающей станции сети передаются через хаб по всем линиям связи всем ПК. Информация поступает на все рабочие станции, но принимается только теми станциями, которым она предназначается. Так как передача сигналов в топологии физическая звезда является широковещательной, т.е. сигналы от ПК распространяются одновременно во все направления, то логическая топология данной локальной сети является логической шиной.

Данная топология применяется в локальных сетях с архитектурой 10Base-T Ethernet.

Преимущества сетей топологии звезда:

- легко подключить новый ПК;
- имеется возможность централизованного управления;
- сеть устойчива к неисправностям отдельных ПК и к разрывам соединения отдельных ПК.

Недостатки сетей топологии звезда:

- отказ хаба влияет на работу всей сети;
- большой расход кабеля.

Топология “кольцо”

В сети с топологией кольцо все узлы соединены каналами связи в неразрывное кольцо (необязательно окружность), по которому передаются данные. Выход одного ПК соединяется со входом другого ПК. Начав движение из одной точки, данные, в конечном счете, попадают на его начало. Данные в кольце всегда движутся в одном и том же направлении.

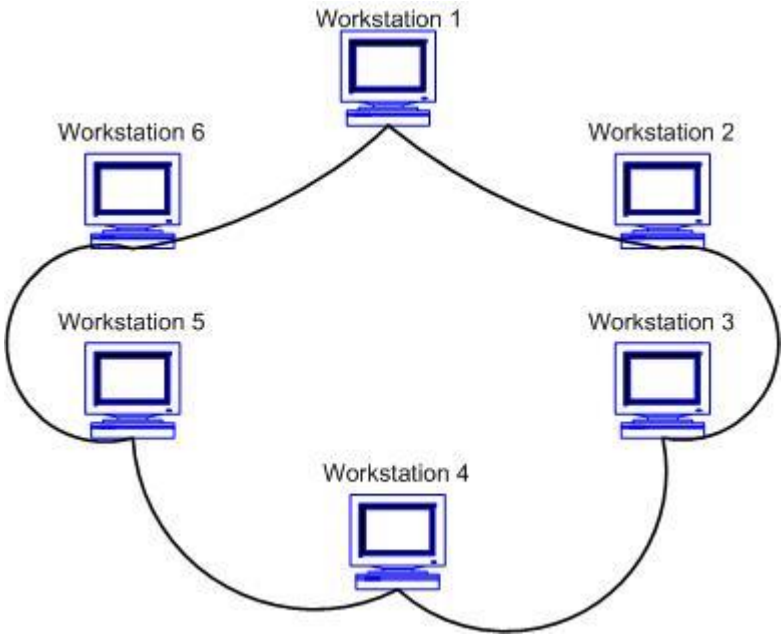


Рис. 5. Топология "кольцо"

Принимающая рабочая станция распознает и получает только адресованное ей сообщение. В сети с топологией типа физическое кольцо используется маркерный доступ, который предоставляет станции право на использование кольца в определенном порядке. Логическая топология данной сети - логическое кольцо. Данную сеть очень легко создавать и настраивать.

К основному недостатку сетей топологии кольцо является то, что повреждение линии связи в одном месте или отказ ПК приводит к неработоспособности всей сети.

Как правило, в чистом виде топология “кольцо” не применяется из-за своей ненадёжности, поэтому на практике применяются различные модификации кольцевой топологии.

Топология Token Ring

Эта топология основана на топологии "физическое кольцо с подключением типа звезда". В данной топологии все рабочие станции подключаются к центральному концентратору (Token Ring) как в топологии физическая звезда. Центральный концентратор - это интеллектуальное устройство, которое с помощью перемычек обеспечивает последовательное соединение выхода одной станции со входом другой станции.

Другими словами, с помощью концентратора каждая станция соединяется только с двумя другими станциями (предыдущей и последующей станциями). Таким образом, рабочие станции связаны петлей кабеля, по которой пакеты данных передаются от одной станции к другой, и каждая станция ретранслирует эти посланные пакеты. В каждой рабочей станции имеется для этого приемо-передающее устройство, которое позволяет

управлять прохождением данных в сети. Физически такая сеть построена по типу топологии “звезда”.

Концентратор создаёт первичное (основное) и резервное кольца. Если в основном кольце произойдёт обрыв, то его можно обойти, воспользовавшись резервным кольцом, так как используется четырёхжильный кабель. Отказ станции или обрыв линии связи рабочей станции не вличет за собой отказ сети как в топологии кольцо, потому что концентратор отключет неисправную станцию и замкнет кольцо передачи данных.

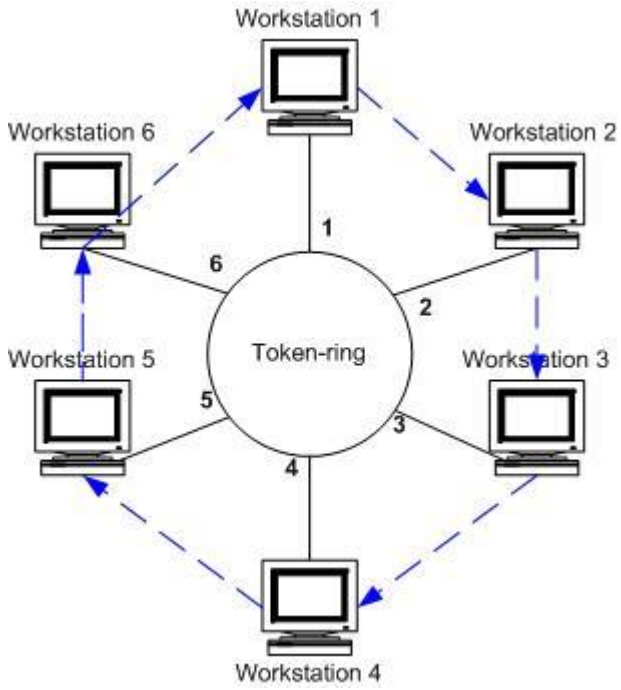


Рис. 6. Топология Token Ring

В архитектуре Token Ring маркер передаётся от узла к узлу по логическому кольцу, созданному центральным концентратором. Такая маркерная передача осуществляется в фиксированном направлении (направление движения маркера и пакетов данных представлено на рисунке стрелками синего цвета). Станция, обладающая маркером, может отправить данные другой станции.

Для передачи данных рабочие станции должны сначала дождаться прихода свободного маркера. В маркере содержится адрес станции, пославшей этот маркер, а также адрес той станции, которой он предназначается. После этого отправитель передает маркер следующей в сети станции для того, чтобы и та могла отправить свои данные.

Один из узлов сети (обычно для этого используется файл-сервер) создаёт маркер, который отправляется в кольцо сети. Такой узел выступает в качестве

активного монитора, который следит за тем, чтобы маркер не был утерян или разрушен.

Преимущества сетей топологии Token Ring:

- топология обеспечивает равный доступ ко всем рабочим станциям;
- высокая надежность, так как сеть устойчива к неисправностям отдельных станций и к разрывам соединения отдельных станций.

Недостатки сетей топологии Token Ring: большой расход кабеля и соответственно дорогостоящая разводка линий связи.

Методы доступа и протоколы передачи данных в локальных сетях

В различных сетях применяются различные сетевые протоколы (протоколы передачи данных) для обмена данными между рабочими станциями.

В 1980 году в Международном институте инженеров по электротехнике и радиоэлектронике (Institute of Electronics Engineers–IEEE) был организован комитет 802 по стандартизации локальных сетей. Комитет 802 разработал семейство стандартов IEEE802. x, которые содержат рекомендации по проектированию нижних уровней локальных сетей.

Стандарты семейства IEEE802.x охватывают только два нижних уровня семиуровневой модели OSI – физический и канальный, так как именно эти уровни в наибольшей степени отражают специфику локальных сетей. Старшие же уровни, начиная с сетевого, в значительной степени имеют общие черты, как для локальных, так и глобальных сетей.

К наиболее распространенным методам доступа относятся: Ethernet, ArcNet и Token Ring, которые реализованы соответственно в стандартах IEEE802.3, IEEE802.4 и IEEE802.5. Кроме того, для локальных сетей, работающих на оптическом волокне, американским институтом по стандартизации ASNI был разработан стандарт FDDI, обеспечивающий скорость передачи данных 100 Мбит/с.

В этих стандартах канальный уровень разделяется на два подуровня, которые называются уровнями:

- управление логическим каналом (LCC - Logical Link Control);
- управление доступом к среде (MAC - Media Access Control).

Уровень управления доступом к среде передачи данных (MAC) появился, так как в локальных сетях используется разделяемая среда передачи данных. В современных локальных сетях получили распространение несколько протоколов уровня MAC, реализующих разные алгоритмы доступа к разделяемой среде. Эти протоколы полностью определяют специфику таких

технологий локальных сетей, как Ethernet, Fast Ethernet, Gigabit Ethernet, Token Ring, FDDI.

После того, как доступ к среде получен, ею может воспользоваться более высокий канальный уровень – уровень LCC, организующий передачу логических единиц данных, кадров информации, с различным уровнем качества транспортных услуг.

Методы доступа к среде передачи данных

В локальных сетях, использующих разделяемую среду передачи данных (например, локальные сети с топологией шина и физическая звезда), актуальным является доступ рабочих станций к этой среде, так как если два ПК начинают одновременно передавать данные, то в сети происходит столкновение.

Для того чтобы избежать этих столкновений необходим специальный механизм, способный решить эту проблему. Шинный арбитраж - это механизм призванный решить проблему столкновений. Он устанавливает правила, по которым рабочие станции определяют, когда среда свободна, и можно передавать данные.

Существуют два метода шинного арбитража в локальных сетях:

- обнаружение столкновений;
- передача маркера.

Обнаружение столкновений

Когда в локальных сетях работает метод обнаружения столкновений, компьютер сначала слушает, а потом передает. Если компьютер слышит, что передачу ведет кто-то другой, он должен подождать окончания передачи данных и затем предпринять повторную попытку.

В этой ситуации (два компьютера, передающие в одно и то же время) система обнаружения столкновений требует, чтобы передающий компьютер продолжал прослушивать канал и, обнаружив на нем чужие данные, прекращал передачу, пытаясь возобновить ее через небольшой (случайный) промежуток времени.

Прослушивание канала до передачи называется “прослушивание несущей” (carrier sense), а прослушивание во время передачи — обнаружение столкновений (collision detection). Компьютер, поступающий таким образом, использует метод, называющийся “обнаружение столкновений с прослушиванием несущей”, сокращенно CSMA/CD.

Передача маркера в локальных сетях

Системы с передачей маркера работают иначе. Для того чтобы передать данные, компьютер сначала должен получить разрешение. Это значит, он

должен “поймать” циркулирующий в сети пакет данных специального вида, называемый маркером. Маркер перемещается по замкнутому кругу, минуя поочередно каждый сетевой компьютер.

Каждый раз, когда компьютер должен послать сообщение, он ловит и держит маркер у себя. Как только передача закончилась, он посылает новый маркер в путешествие дальше по сети. Такой подход дает гарантию, что любой компьютер рано или поздно получит право поймать и удерживать маркер до тех пор, пока его собственная передача не закончится.

Контрольные вопросы по теме

Уровень модуля

Уровень курса

1. Линии и каналы передачи данных в сетях.
2. Кабельные линии связи.
3. Беспроводные каналы передачи данных.
4. Методы передачи данных в локальной сети на физическом уровне.
5. Методы передачи данных в локальной сети на канальном уровне.
6. Протоколы, интерфейсы, стеки протоколов в сетевых технологиях.
7. Семиуровневая эталонная модель взаимодействия в сети.
8. Сетевые топологии.
9. Шинная топология сети.
10. Топология сети типа “звезда”.
11. Топология сети “кольцо”.
12. Топология сети Token Ring.
13. Методы доступа к среде передачи данных. Шинный арбитраж.

Лекция № 6
Тема: Глобальные сети

Оглавление

Источники:

1. Таненбаум Э., Остин Т. Архитектура компьютера. 6-е изд. — СПб.: Питер, 2013. — 816 с.: ил.

   https://rutracker.org/forum/viewtopic.php?t=4956359

Введение в глобальные сети

Глобальные вычислительные сети Wide Area Networks (WAN), которые относятся к территориальным компьютерным сетям, предназначены, как и ЛВС для предоставления услуг, но значительно большему количеству пользователей, находящихся на большой территории.

Глобальные вычислительные сети — это компьютерные сети, объединяющие локальные сети и отдельные компьютеры, удаленные друг от друга на большие расстояния. В настоящее время всеобъемлющей глобальной сетью стала сеть Интернет. Попытки создания других сетей остались в прошлом. Поэтому, вместо глобальных вычислительных сетей фактически можно говорить только об одной - сети Интернет.

Глобальная сеть Интернет

Интернет - это множество компьютеров (хостов) и различных сетей, объединенных сетью на базе протоколов связи TCP/IP. Компьютеры, подключенные к сети Интернет, могут иметь любые аппаратные и программные платформы, но при этом они должны поддерживать стек протоколов (семейство протоколов) связи TCP/IP. Единого владельца и центра управления сети Интернет не существует.

Интернет начал свое существование с сети ARPANet в 1969 году. Эта компьютерная сеть с применением технологии коммутации пакетов была создана в США по заданию военного ведомства США как высоконадежная сеть передачи данных. В 1983 году ARPANet разделилась на две сети, одна - MILNET стала частью оборонной сети передачи данных США, другая - была использована для соединения академических и исследовательских центров, которая постепенно развивалась и в 1990 году трансформировалась в Интернет.

Компоненты структуры сети Интернет объединяются в общую иерархию.

В настоящее время в интернете существует достаточно большое количество служб, обеспечивающих работу со всем спектром ресурсов. Наиболее известными среди них являются:

- служба DNS, или система доменных имён, обеспечивающая возможность использования для адресации узлов сети мнемонических имён вместо числовых адресов;
- электронная почта (E-mail), обеспечивающая возможность обмена сообщениями одного человека с одним или несколькими абонентами;
- служба IRC, предназначенная для поддержки текстового общения в реальном времени (chat);

- телеконференции, или группы новостей (Usenet), обеспечивающие возможность коллективного обмена сообщениями;
- служба FTP — система файловых архивов, обеспечивающая хранение и пересылку файлов различных типов;
- служба Telnet, предназначенная для управления удалёнными компьютерами в терминальном режиме;
- World Wide Web (WWW, W3, «Всемирная паутина») — гипертекстовая (гипермедиа) система, предназначенная для интеграции различных сетевых ресурсов в единое информационное пространство;
- потоковое мультимедиа.

Список предоставляемых услуг непрерывно растет.

Структура и основные принципы построения сети Интернет

Internet – всемирная информационная компьютерная сеть, представляющая собой объединение множества региональных компьютерных сетей и компьютеров, обменивающих друг с другом информацией по каналам общественных телекоммуникаций (выделенным телефонным аналоговым и цифровым линиям, оптическим каналам связи и радиоканалам, в том числе спутниковым линиям связи).

Информация в Internet хранится на серверах. Серверы имеют свои адреса и управляются специализированными программами. Они позволяют пересылать почту и файлы, производить поиск в базах данных и выполнять другие задачи.

Обмен информацией между серверами сети выполняется по высокоскоростным каналам связи (выделенным телефонным линиям, оптоволоконным и спутниковым каналам связи). Доступ отдельных пользователей к информационным ресурсам Internet обычно осуществляется через провайдера или корпоративную сеть.

Провайдер - поставщик сетевых услуг – лицо или организация предоставляющие услуги по подключению к компьютерным сетям. В качестве провайдера выступает некоторая организация, имеющая модемный пул для соединения с клиентами и выхода во всемирную сеть.

Основными ячейками глобальной сети являются локальные вычислительные сети. Если некоторая локальная сеть непосредственно подключена к глобальной, то и каждая рабочая станция этой сети может быть подключена к ней.

Существуют также компьютеры, которые непосредственно подключены к глобальной сети. Они называются хост - компьютерами (host - хозяин). Хост

– это любой компьютер, являющийся постоянной частью Internet, т.е. соединенный по Internet- протоколу с другим хостом, который в свою очередь, соединен с другим, и так далее.

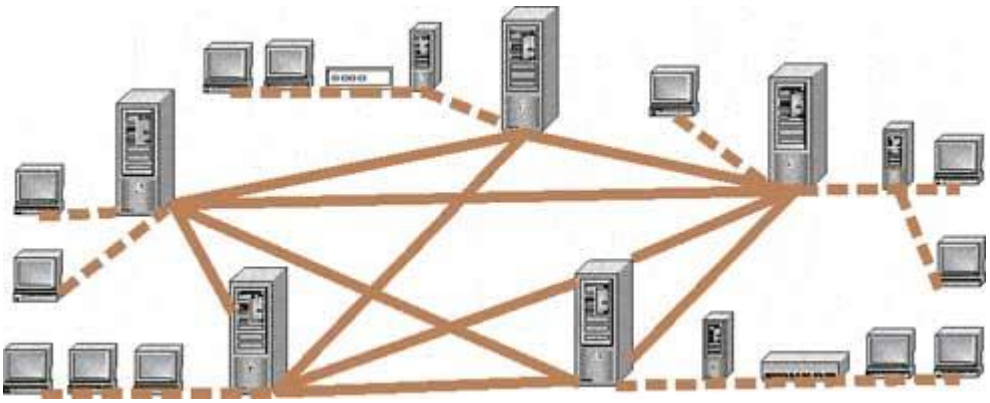


Рис. 1. Структура глобальной сети Internet

Для подсоединения линий связи к компьютерам используются специальные электронные устройства, которые называются сетевыми платами, сетевыми адаптерами, модемами и т.д.

Практически все услуги Internet построены на принципе клиент-сервер. Вся информация в Интернет хранится на серверах. Обмен информацией между серверами осуществляется по высокоскоростным каналам связи или магистралям. Серверы, объединенные высокоскоростными магистралями, составляют базовую часть сети Интернет.

Отдельные пользователи подключаются к сети через компьютеры местных поставщиков услуг Интернета, Internet - провайдеров (Internet Service Provider - ISP), которые имеют постоянное подключение к Интернет. Региональный провайдер, подключается к более крупному провайдеру национального масштаба, имеющего узлы в различных городах страны. Сети национальных провайдеров объединяются в сети транснациональных провайдеров или провайдеров первого уровня. Объединенные сети провайдеров первого уровня составляют глобальную сеть Internet.

Передача информации в Интернет обеспечивается благодаря тому, что каждый компьютер в сети имеет уникальный адрес (IP-адрес), а сетевые протоколы обеспечивают взаимодействие разнотипных компьютеров, работающих под управлением различных операционных систем.

В основном в Интернет используется семейство сетевых протоколов (стек) TCP/IP. На канальном и физическом уровне стек TCP/IP поддерживает технологию Ethernet, FDDI и другие технологии. Основой семейства протоколов TCP/IP является сетевой уровень, представленный протоколом IP, а также различными протоколами маршрутизации. Этот уровень обеспечивает перемещение пакетов в сети и управляет их машрутизацией. Размер пакета,

параметры передачи, контроль целостности осуществляется на транспортном уровне TCP.

Прикладной уровень объединяет все службы, которые система предоставляет пользователю. К основным прикладным протоколам относятся: протокол удаленного доступа telnet, протокол передачи файлов FTP, протокол передачи гипертекста HTTP, протоколы электронной почты: SMTP, POP, IMAP, MIME.

Наиболее распространённые в интернете протоколы (в алфавитном порядке, сгруппированные в примерном соответствии модели OSI):

Уровень OSI	Протоколы, примерно соответствующие уровню OSI
Прикладной	BGP, DNS, FTP, HTTP, HTTPS, IMAP, LDAP, POP3, SNMP, SMTP, SSH, Telnet, XMPP (Jabber)
Сеансовый/Представления	SSL, TLS
Транспортный	TCP, UDP
Сетевой	EIGRP, ICMP, IGMP, IP, IS-IS, OSPF, RIP
Канальный	Arcnet, ATM, Ethernet, Frame relay, HDLC, PPP, L2TP, SLIP, Token ring

Способы доступа к глобальной сети Интернет

Известны следующие способы доступа в Интернет:

1. Dial-Up (когда компьютер пользователя подключается к серверу провайдера, используя телефон) – коммутируемый доступ по аналоговой телефонной сети скорость передачи данных до 56 Кбит/с.

2. ISDN - коммутируемый доступ по цифровой телефонной сети. Главная особенность использования ISDN - это более высокая скорость передачи информации, по сравнению с Dial-Up доступом. Скорость передачи данных составляет 64 Кбит/с при использовании одного и 128 Кбит/с, при использовании двух каналов связи.

3. DSL (Digital Subscriber Line) - семейство цифровых абонентских линий, предназначенных для организации доступа по аналоговой телефонной сети, используя кабельный модем. Эта технология (ADSL, VDSL, HDSL, ISDL, SDSL, SHDSL, RADSL под общим названием xDSL) обеспечивает высокоскоростное соединение до 50 Мбит/с (фактическая скорость до 2 Мбит/с). Основным преимуществом технологий xDSL является возможность значительно увеличить скорость передачи данных по телефонным проводам

без модернизации абонентской телефонной линии. Пользователь получает доступ в сеть Интернет с сохранением обычной работы телефонной связи.

4. Доступ в Интернет по выделенным линиям (аналоговым и цифровым). Доступ по выделенной линии - это такой способ подключения к Интернет, когда компьютер пользователя соединен с сервером провайдера с помощью кабеля (витой пары) и это соединение является постоянным, т.е. некоммутируемым, и в этом главное отличие от обычной телефонной связи. Скорость передачи данных до 100 Мбит/с.

5. Доступ в Интернет по локальной сети (Fast Ethernet). Подключение осуществляется с помощью сетевой карты со скоростью передачи данных до 1 Гбит/с.

6. Доступ в Интернет по локальной сети с подключением через оптический кабель со скоростью передачи данных 1 Гбит/с для конечного пользователя.

7. Спутниковый доступ в Интернет или спутниковый Интернет (DirecPC, Europe Online). Спутниковый доступ в Интернет бывает двух видов - ассиметричный и симметричный:

- обмен данными компьютера пользователя со спутником двухсторонний;
- запросы от пользователя передаются на сервер спутникового оператора через любое доступное наземное подключение, а сервер передает данные пользователю со спутника. Максимальная скорость приема данных до 52,5 Мбит/с (реальная средняя скорость до 3 Мбит/с).

8. Доступ в Интернет с использованием каналов кабельной телевизионной сети (“coax at a home”), скорость приема данных от 2 до 56 Мб/сек. В настоящее время известны две архитектуры передачи данных – это симметричная и асимметричная архитектуры.

9. Беспроводные технологии:

- WiFi;
- WiMax;
- RadioEthernet;
- MMDS;
- LMDS
- мобильный GPRS – Интернет;
- мобильный CDMA – Internet.

Адресация в глобальной сети Интернет

Адреса в Интернете могут быть представлены как последовательностью цифр, так и именем, построенным по определенным правилам. Компьютеры при пересылке информации используют цифровые адреса, а пользователи в работе с Интернетом используют в основном имена. Таким образом, адреса компьютеров имеют двойную кодировку:

- цифровой IP-адрес.
- DNS-адрес (доменный адрес).

Основным протоколом сети Интернет является сетевой протокол TCP/IP. Каждый компьютер в сети TCP/IP (подключенный к сети Интернет) имеет свой уникальный IP-адрес или IP – номер. IP-адреса, представленные в цифровом виде (IP-номера), состоят из четырех байтов, т.е. из 32-разрядного двоичного числа, которое разделяется на четыре блока по 8 бит. Цифровой IP-адрес можно записать в виде четырех десятичных чисел, разделенных точками. Например, 195.82.54.17. Каждое число не должно превышать двухсот пятидесяти шести. IP-адрес включает две части: номер сети и номер узла (компьютера) в сети. Такой способ нумерации позволяет иметь в сети более четырех миллиардов компьютеров. Если отдельный компьютер (хост-компьютер) или сеть являются составной частью сети Интернет, то IP-адрес присваивается специальным подразделением Интернета. Для отдельного компьютера или локальной сети, которые впервые подключаются к сети Интернет, специальная организация, занимающейся администрированием доменных имен, присваивает IP – номера. Распределением IP адресов занимается организация ICANN (Internet Corporation for Assigned Names and Numbers), а в Европе распределением IP адресов между региональными провайдерами занимается RIPE. Адреса компьютеров в сети определяют администраторы сети.

Первоначально в сети Internet применялись IP – номера, но когда количество компьютеров в сети стало больше чем 1000, то был принят метод связи имен и IP – номеров, который называется сервер имени домена (Domain Name Server, DNS).

Преобразование DNS-адреса в цифровой IP-адрес осуществляет сервер имени домена DNS (Domain Name Server). Сервер DNS поддерживает список имен локальных сетей и компьютеров и соответствующих им IP – номеров.

Доменный адрес (DNS-адрес) состоит из нескольких доменов (буквенно-числовых обозначений), которые отделяются друг от друга точкой. Доменный адрес построен на основе иерархической классификации, т.е. доменный адрес включает в себя несколько уровней доменов, например: gorod.dp.ua.

Домен верхнего уровня располагается в имени правее, а домен нижнего уровня – левее. Пользователь сети Интернет работает не с IP-адресами, а только с доменными адресами.

В Интернете применяется так называемая доменная система имен. Каждый уровень в такой системе называется доменом. Типичное имя домена состоит из нескольких частей, расположенных в определенном порядке и разделенных точками. Домены отделяются друг от друга точками, например: www.lessons.info или yyy.zzz.ua.

В Интернете доменная система имен использует принцип последовательных уточнений, также как и в обычных почтовых адресах - страна, город, улица и дом, в который следует доставить письмо.

Домен верхнего уровня располагается в имени правее, а домен нижнего уровня - левее. В нашем примере домены верхнего уровня info и ua указывают на то, что речь идет о принадлежности сайта www.lessons.info к тематическому домену верхнего уровня info, а сайта yyy.zzz.ua к украинской (ua) части Интернета. Но в Украине множество пользователей Интернета, и следующий уровень определяет организацию, которой принадлежит данный адрес. В нашем случае это компания zzz.

Интернет-адрес этой компании - zzz.ua. Все компьютеры, подключенные к Интернету в этой компании, объединяются в группу, имеющую такой адрес. Имя отдельного компьютера или сети каждая компания выбирает для себя самостоятельно, а затем регистрирует его в той организации Интернет, которая обеспечивает подключение.

Это имя в пределах домена верхнего уровня должно быть уникальным. Далее следует имя хоста yyy, таким образом, полное имя домена третьего уровня: yyy.zzz.ua. В имени может быть любое число доменов, но чаще всего используются имена с количеством доменов от трех до пяти.

Доменная система образования адресов гарантирует, что во всем Интернете больше не найдется другого компьютера с таким же адресом. Для доменов нижних уровней можно использовать любые адреса, но для доменов самого верхнего уровня существует соглашение.

В системе адресов Интернета приняты домены, представленные географическими регионами. Они имеют имя, состоящее из двух букв, например:

- Украина - ua;
- Франция - fr;
- Канада - ca;
- США - us;
- Россия - ru.

Существуют и домены, разделенные по тематическим признакам, например:

- Учебные заведения - edu.
- Правительственные учреждения - gov.
- Коммерческие организации - com.

В последнее время добавлены новые зоны, например: biz, info, in, .cn и так далее.

Регистрация домена осуществляется в выбранной пользователем зоне ua, ru, com, net, info и так далее. В зависимости от назначения сайта выбирается его зона регистрации. Для регистрации сайта желательно выбрать домен второго уровня, например, lessons.info, хотя можно работать и с доменом третьего уровня. Домен второго уровня регистрируется у регистратора – организации занимающейся администрированием доменных имен. Домен третьего уровня приобретается, как правило, вместе с хостингом у хостинговой компании. Имя сайта выбирают исходя из вида деятельности, названия компании или фамилии владельца сайта.

Универсальные указатели ресурсов

При работе в Internet используются не доменные имена, а универсальные указатели ресурсов, называемые URL (Universal Resource Locator). URL - это адрес любого ресурса (документа, файла) в Internet, он указывает, с помощью какого протокола следует к нему обращаться, какую программу следует запустить на сервере и к какому конкретному файлу следует обратиться на сервере. Общий вид URL: протокол://хост-компьютер/имя файла.

World Wide Web

В настоящее время большая часть трафика в Интернет приходится на службу World Wide Web (всемирная паутина). Данная служба использует протокол HTTP.

Принцип работы сервиса WWW был разработан физиками Тимом Бернес-Ли и Робертом Кайо в европейском исследовательском центре CERN (Женева) в 1989 году. В настоящее время Web – служба Интернет содержит миллионы страниц информации с различными видами документов.

Суть системы World Wide Web (WWW) состоит в применении гипертекстовой модели к информационным ресурсам, распределенным в глобальной сети. WWW - один из видов сервиса Интернет. WWW предоставляет возможность работы с документами, в которых объединены текст, графические изображения, звуки, анимация, что значительно облегчает восприятие информации. Гипертекстовые документы (Web-страницы)

создаются с помощью специального языка разметки гипертекста HTML (Hyper Text Markup Language).

Система WWW работает по принципу клиент-сервер. Клиент – это интерпретатор HTML, специальная программу просмотра, называемая WWW-браузер (WWW-browser). WWW-браузер - это прикладная программа, которая взаимодействует с системой WWW, получает затребованные документы, интерпретирует данные и отображает содержание документов на экране. Программа клиент обеспечивает доступ практически ко всем информационным ресурсам Интернет, которые хранятся на серверах.

HTTP (англ. HyperText Transfer Protocol — «протокол передачи гипертекста») — протокол прикладного уровня передачи данных (изначально – в виде гипертекстовых документов в формате «HTML», в настоящий момент используется для передачи произвольных данных). Основой HTTP является технология «клиент-сервер», то есть предполагается существование:

- Потребителей (клиентов), которые инициируют соединение и посылают запрос;
- Поставщиков (серверов), которые ожидают соединения для получения запроса, производят необходимые действия и возвращают обратно сообщение с результатом.

Основным объектом манипуляции в HTTP является ресурс, на который указывает URI (Uniform Resource Identifier) в запросе клиента. Обычно такими ресурсами являются хранящиеся на сервере файлы, но ими могут быть логические объекты или что-то абстрактное. Особенностью протокола HTTP является возможность указать в запросе и ответе способ представления одного и того же ресурса по различным параметрам: формату, кодировке, языку и т. д. (в частности, для этого используется HTTP-заголовок).

Для работы с системой WWW необходимо установить на своем компьютере одну из программ просмотра Web-страниц, например, Internet Explorer, Mozilla Firefox, MyIE Web Browser, Opera и т.д. Большинство браузеров предоставляют доступ к другим серверам Интернета: к FTP-серверам, Gopher-серверам и серверам телеконференций UseNet.

Контрольные вопросы по теме

Уровень модуля

Уровень курса

1. Структура и основные принципы построения сети Интернет.
2. Способы доступа к глобальной сети Интернет.
3. Адресация в глобальной сети Интернет.

Лекция № 7

Тема: Аналогово-цифровые и цифро-аналоговые преобразователи

Оглавление

Источники:

1. Таненбаум Э., Остин Т. Архитектура компьютера. 6-е изд. — СПб.: Питер, 2013. — 816 с.: ил.

   https://rutracker.org/forum/viewtopic.php?t=4956359

Классы сигналов

Сигналы могут быть **непрерывными** и **дискретными**. В связи с этим сигналы можно разделить на следующие классы:

а) произвольные по величине и непрерывные по времени (рис. 7.1, а);

б) произвольные по величине и дискретные по времени (рис. 7.1, б);

в) квантованные по величине и непрерывные по времени (рис. 7.1, в);

г) квантованные по величине и дискретные по времени (рис. 7.1, г).

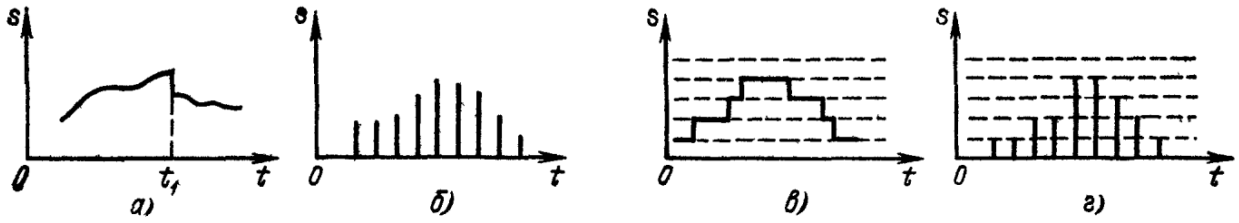


Рис. 7.1. Сигналы произвольные по величине и по времени (а), произвольные по величине и дискретные по времени (б), квантованные по величине и непрерывные по времени (в), квантованные по величине и дискретные по времени (г)

Сигналы первого класса (рис. 7.1а) называют **аналоговыми** или непрерывными, так как они задаются по оси времени на несчетном множестве точек. Такие множества называют континуальными. При этом по оси ординат сигналы могут принимать любое значение в определенном интервале. Эти сигналы могут иметь разрывы, как, например, на рис. 7.1а.

На рис. 7.1б представлен сигнал, заданный при дискретных значениях времени *t* (на счетном множестве точек); величина же сигнала в этих точках может принимать любое значение в определенном интервале по оси ординат (как и на рис. 7.1а). Таким образом, термин **дискретный** характеризует не сам сигнал, а способ задания его на временной оси. Дискретные сигналы могут создаваться непосредственно источником информации (например, дискретными датчиками в системах управления или телеметрии) или образовываться в результате дискретизации континуальных сигналов.

Сигнал на рис. 7.1в задан на всей временной оси, однако его величина может принимать лишь дискретные значения. В подобных случаях говорят о сигнале, квантованном по уровню.

В дальнейшем термин дискретный будет применяться только по отношению к дискретизации по времени; дискретность же по уровню будет обозначаться термином **квантование**.

Квантование используют при представлении сигналов в цифровой форме с помощью цифрового кодирования, поскольку уровни можно

пронумеровать числами с конечным числом разрядов. Поэтому дискретный по времени и квантованный по уровню сигнал (рис. 7.1г) называется цифровым.

Таким образом, можно различать: аналоговые (рис. 7.1а), дискретные (рис. 7.1б), квантованные (рис. 7.1в) и цифровые (рис. 7.1г) сигналы. Каждому из этих классов сигналов можно поставить в соответствие аналоговую, дискретную или цифровую цепи.

Аналого-цифровое преобразование сигналов

Для преобразования любого аналогового сигнала (звука, изображения) в цифровую форму необходимо выполнить три основные операции: дискретизацию, квантование и кодирование.

Дискретизация - представление непрерывного аналогового сигнала последовательностью его значений (отсчетов). Эти отсчеты берутся в моменты времени, отделенные друг от друга интервалом, который называется интервалом или шагом дискретизации. Величину, обратную интервалу между отсчетами, называют частотой дискретизации.

Квантование представляет собой замену величины отсчета сигнала ближайшим значением из набора фиксированных величин - уровней квантования. Другими словами, квантование - это округление величины отсчета. Уровни квантования делят весь диапазон возможного изменения значений сигнала на конечное число интервалов - шагов квантования. Расположение уровней квантования обусловлено шкалой квантования. Используются как равномерные, так и неравномерные шкалы.

Цифровое кодирование. Квантованный сигнал, в отличие от исходного аналогового, может принимать только конечное число значений. Это позволяет представить его в пределах каждого интервала дискретизации числом, равным порядковому номеру уровня квантования. В свою очередь это число можно выразить комбинацией некоторых знаков или символов. Совокупность знаков (символов) и система правил, при помощи которых данные представляются в виде набора символов, называют кодом. Конечная последовательность кодовых символов называется кодовым словом. Квантованный сигнал можно преобразовать в последовательность кодовых слов. Эта операция и называется кодированием. Каждое кодовое слово передается в пределах одного интервала дискретизации. Для кодирования сигналов звука и изображения широко применяют двоичный код. Если квантованный сигнал может принимать N значений, то число двоичных символов в каждом кодовом слове $n >= \log_2 N$. Один разряд, или символ слова, представленного в двоичном коде, называют битом. Обычно число уровней квантования равно целой степени числа 2, т.е. $N = 2^n$.

Кодовые слова можно передавать в параллельной или последовательной формах. Для передачи в параллельной форме надо использовать n линий связи. Символы кодового слова одновременно передаются по линиям в пределах интервала дискретизации. Для передачи в последовательной форме интервал дискретизации надо разделить на n подинтервалов - тактов. В этом случае символы слова передаются последовательно по одной линии, причем на передачу одного символа слова отводится один такт. Каждый символ слова передается с помощью одного или нескольких дискретных сигналов - импульсов. Преобразование аналогового сигнала в последовательность кодовых слов поэтому часто называют импульсно-кодовой модуляцией. Форма представления слов определенными сигналами определяется форматом кода. Можно, например, устанавливать в пределах такта высокий уровень сигнала, если в данном такте передается двоичный символ 1, и низкий - если передается двоичный символ 0 (такой способ представления называют форматом БВН - Без Возвращения к Нулю). Если используются 4-разрядные двоичные слова, то это позволяет иметь 16 уровней квантования. В параллельном цифровом потоке по каждой линии в пределах интервала дискретизации передается 1 бит 4-разрядного слова. В последовательном потоке интервал дискретизации делится на 4 такта, в которых передаются (начиная со старшего) биты 4-разрядного слова.

Операции, связанные с преобразованием аналогового сигнала в цифровую форму (дискретизация, квантование и кодирование), выполняются одним устройством - *аналого-цифровым преобразователем* (АЦП). Современная АЦП выполняется на одной интегральной микросхеме.

Обратная процедура, т.е. восстановление аналогового сигнала из последовательности кодовых слов, производится в *цифро-аналоговом преобразователе* (ЦАП).

Аналогово-цифровые преобразователи (АЦП)

Аналого-цифровое преобразование – это процесс преобразования входной физической величины в ее числовое представление. Аналого-цифровой преобразователь – устройство, выполняющее такое преобразование. Формально, входной величиной АЦП может быть любая физическая величина – напряжение, ток, сопротивление, емкость, частота следования импульсов, угол поворота вала и т.п. Однако, для определенности, в дальнейшем под АЦП мы будем понимать исключительно преобразователи напряжение-код.

Аналогово-цифровой преобразователь (АЦП) — один из самых важных электронных компонентов в измерительном и тестовом оборудовании. АЦП преобразует напряжение (аналоговый сигнал) в код, над которым микропроцессор и программное обеспечение выполняют определенные

действия. Существует несколько основных типов архитектуры АЦП, хотя в пределах каждого типа существует также множество вариаций. Различные типы измерительного оборудования используют различные типы АЦП. Например, в цифровом осциллографе используется высокая частота дискретизации, но не требуется высокое разрешение. В цифровых мультиметрах нужно большее разрешение, но можно пожертвовать скоростью измерения. Системы сбора данных общего назначения по скорости дискретизации и разрешающей способности обычно занимают место между осциллографами и цифровыми мультиметрами. В оборудовании такого типа используются АЦП последовательного приближения, либо сигма-дельта АЦП.

Существуют также параллельные АЦП для приложений, требующих скоростной обработки аналоговых сигналов, и интегрирующие АЦП с высокими разрешением и помехоподавлением.

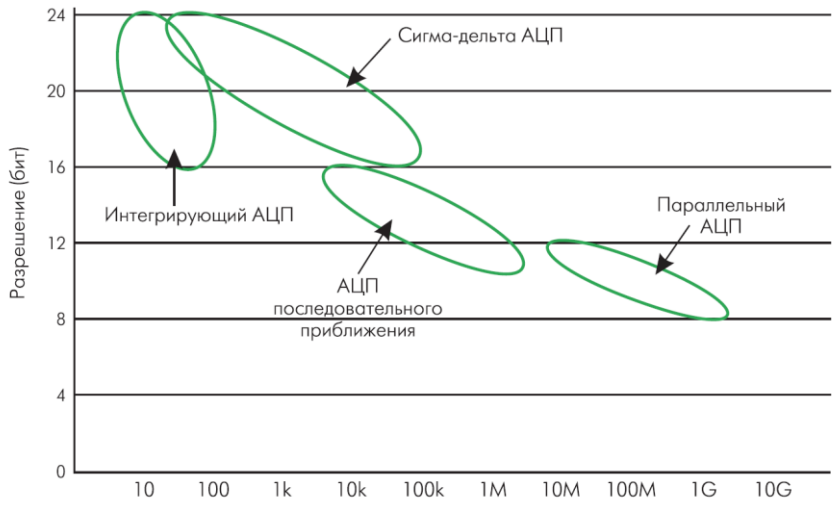


Рис. 7.2 Разрешение в зависимости от частоты дискретизации для различных типов АЦП

На рис. 7.2 показаны возможности основных архитектур АЦП в зависимости от разрешения и частоты дискретизации.

Аналогово-цифровые преобразователи можно классифицировать по нескольким признаком (рис. 7.3). Прежде всего АЦП классифицируют по способу подключения к компьютеру. Возможно непосредственное подключение АЦП к внутренней шине компьютера. Тогда плата АЦП вставляется в разъем, соединенный с системной шиной на материнской плате. Плата АЦП в этом случае должна реализовывать соответствующий шинный интерфейс: ISA, PCI, PCIe, CompactPCI, VMEbus. Возможно также подключение АЦП к компьютеру через порт. Чаще всего используется USB-порт, до распространения USB широко применялось подключение через COM-порт.

АЦП может подключаться по сетевому интерфейсу, тогда данные с выхода АЦП передаются через локальную сеть. Соответственно, АЦП должен в этом случае реализовывать соответствующий интерфейс на физическом уровне и поддерживать необходимые протоколы на канальном, сетевом, транспортном и, если необходимо, на других уровнях. АЦП в промышленности при работе через сеть обычно поддерживают один из таких интерфейсов: Ethernet, Industrial Ethernet, RS-485, CAN, PROFINET.

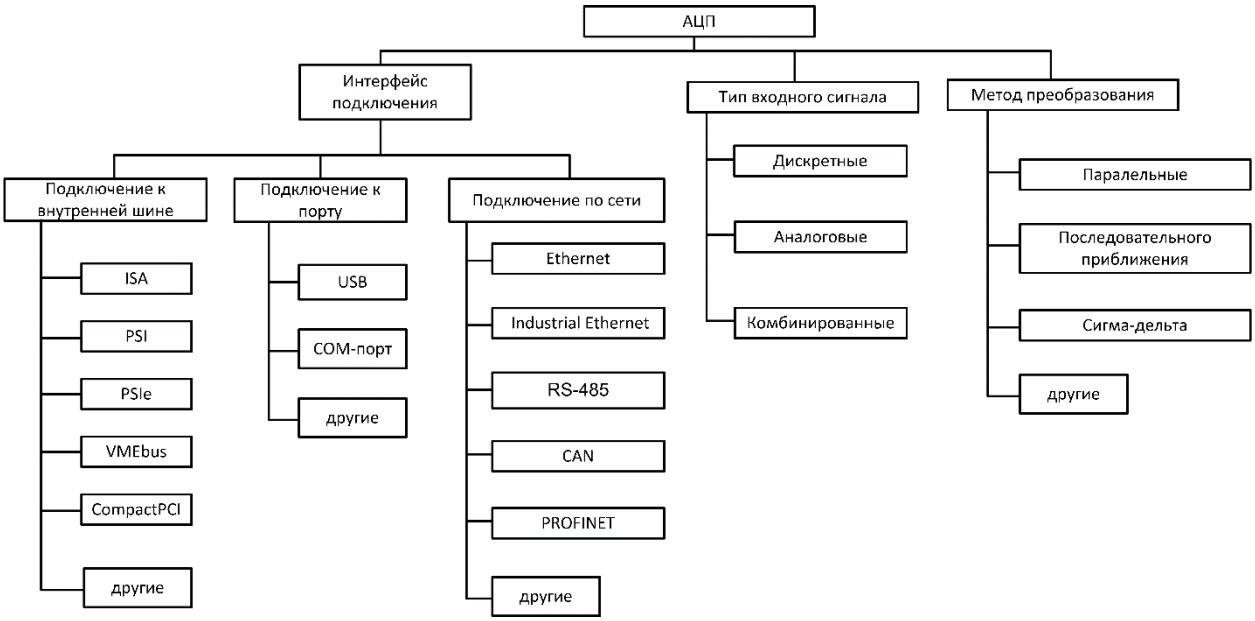


Рис. 7.3. Классификация АЦП

Сигналы, которые необходимо оцифровать и ввести компьютер, на практике разделяют на два вида: релейные (или дискретные) сигналы и аналоговые сигналы. Релейные сигналы - это сигналы, которые могут принимать только два состояния, например: 0 вольт или 5 вольт, 0 вольт или 30 вольт, 0 вольт или 220 вольт и т.д. Однако, необходимо указать, что АЦП обычно работают с входными сигналами постоянного тока порядка 5 Вольт подача высоких напряжений приведет к выходу его из строя. Поэтому релейные сигналы с другими параметрами должны быть предварительно преобразованы в вид, определяемый технической спецификацией АЦП. Другим типом релейного сигнала является сигнал "замкнуто-разомкнуто" или иначе "есть цепь - нет цепи". В данном случае напряжение на входах от объекта контроля не появляется, от объекта контроля сигнал не приходит. АЦП должно само опросить этот вход (подать напряжение на этот вход) и определить: существует ли электрическая цепь между данными двумя контактами (то есть на этом входе, который представляет собой ничто иное как просто два контакта, к которым подводятся сигнальные провода от датчиков объекта контроля) или цепь разомкнута. Такой тип сигнала называют "сухой контакт".

АЦП, работающие с релейными сигналами, называют релейными АЦП, дискретными АЦП, АЦП релейного ввода, АЦП дискретного ввода; встречаются и другие названия. АЦП, которые производят оцифровку аналоговых сигналов называют аналоговыми АЦП, АЦП аналогового ввода, а также применяются аналогичные названия, отражающие назначение этих АЦП.

На практике получили большое распространение АЦП, одновременно имеющие как релейные, так и аналоговые входы. Такие АЦП обычно называют комбинированными.

В компьютерно-интегрированных технологиях широкое применение находят многофункциональные модули, которые объединяют в себе и функции АЦП, и функции цифро-аналоговых преобразователей (ЦАП). Эти модули имеют по несколько релейных и аналоговых входов, а также по несколько релейных и аналоговых выходов.

Параллельные АЦП (АЦП прямого преобразования)

Большинство высокоскоростных осциллографов и некоторые высокочастотные измерительные приборы используют параллельные АЦП из-за их высокой скорости преобразования, которая может достигать 5Г отсчетов/с ($5x10^9$) для стандартных устройств и 20Г отсчетов/с для оригинальных разработок. Обычно параллельные АЦП имеют разрешение до 8 разрядов, но встречаются также 10-разрядные версии.

Архитектура АЦП прямого преобразования изображена на рис. 7.4

Принцип действия АЦП предельно прост: входной сигнал поступает одновременно на все «плюсовые» входы компараторов, а на «минусовые» подается ряд напряжений, получаемых из опорного путем деления резисторами R. Для схемы на рис. 7.4 этот ряд будет таким: (1/16, 3/16, 5/16, 7/16, 9/16, 11/16, 13/16) Uref, где Uref – опорное напряжение АЦП.

Пусть на вход АЦП подается напряжение, равное 1/2 Uref. Тогда сработают первые 4 компаратора (если считать снизу), и на их выходах появятся логические единицы. Приоритетный шифратор (priority encoder) сформирует из «столбца» единиц двоичный код, который фиксируется выходным регистром.

Теперь становятся понятны достоинства и недостатки такого преобразователя. Все компараторы работают параллельно, время задержки схемы равно времени задержки в одном компараторе плюс время задержки в шифраторе. Компаратор и шифратор можно сделать очень быстрыми, в итоге вся схема имеет очень высокое быстродействие.

Но для получения N разрядов нужно 2^N компараторов (и сложность шифратора тоже растет как 2^N). Схема на рис. 7.4. содержит 8 компараторов и

имеет 3 разряда, для получения 8 разрядов нужно уже 256 компараторов, для 10 разрядов – 1024 компаратора, для 24-битного АЦП их понадобилось бы свыше 16 млн.

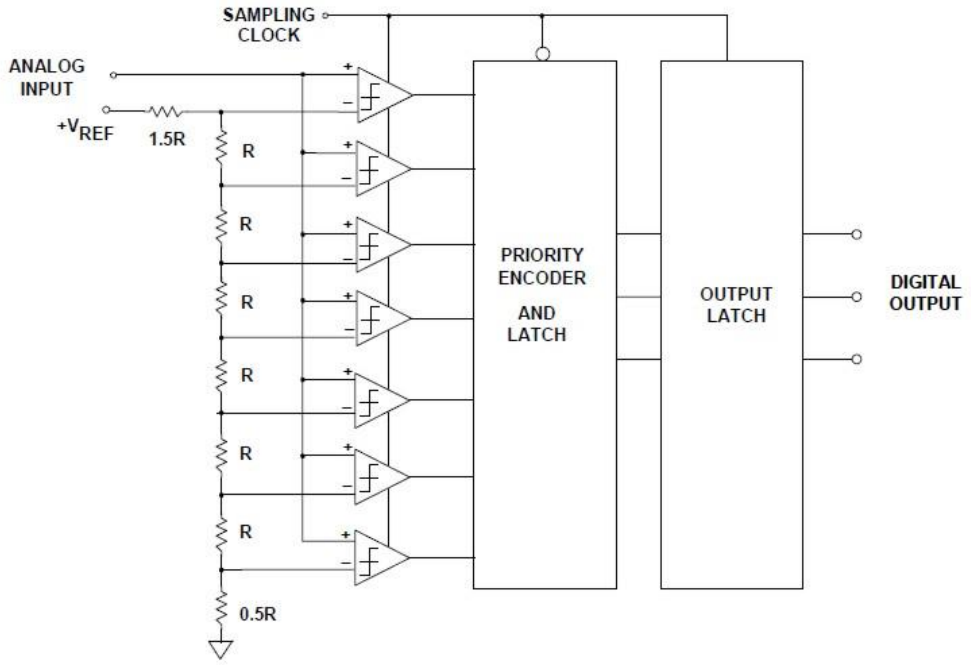


Рис. 7.4. Структурная схема АЦП прямого преобразования

Параллельные АЦП — достаточно быстрые устройства, но они имеют свои недостатки. Из-за необходимости использовать большое количество компараторов параллельные АЦП потребляют значительную мощность, и их нецелесообразно использовать в приложениях с батарейным питанием.

АЦП последовательного приближения

Когда необходимо разрешение 12,14 или 16 разрядов и не требуется высокая скорость преобразования, а определяющими факторами являются невысокая цена и низкое энергопотребление, то обычно применяют АЦП последовательного приближения. Этот тип АЦП чаще всего используется в разнообразных измерительных приборах и в системах сбора данных. В настоящий момент АЦП последовательного приближения позволяют измерять напряжение с точностью до 16 разрядов с частотой дискретизации от 100К ($100x10^3$) до 1М ($1x10^6$) отсчетов/с.

АЦП последовательного приближения реализует алгоритм «взвешивания». Аналого-цифровой преобразователь последовательного приближения (SAR, Successive Approximation Register) измеряет величину входного сигнала, осуществляя ряд последовательных «взвешиваний», то есть сравнений величины входного напряжения с рядом величин, генерируемых следующим образом:

1. На первом шаге на выходе встроенного цифро-аналогового преобразователя устанавливается величина, равная 1/2Uref (здесь и далее мы предполагаем, что сигнал находится в интервале (0 – Uref).

2. Если сигнал больше этой величины, то он сравнивается с напряжением, лежащим посередине оставшегося интервала, т.е., в данном случае, 3/4Uref. Если сигнал меньше установленного уровня, то следующее сравнение будет производиться с меньшей половиной оставшегося интервала (т.е. с уровнем 1/4Uref).

3. Шаг 2 повторяется N раз. Таким образом, N сравнений («взвешиваний») порождает N бит результата.

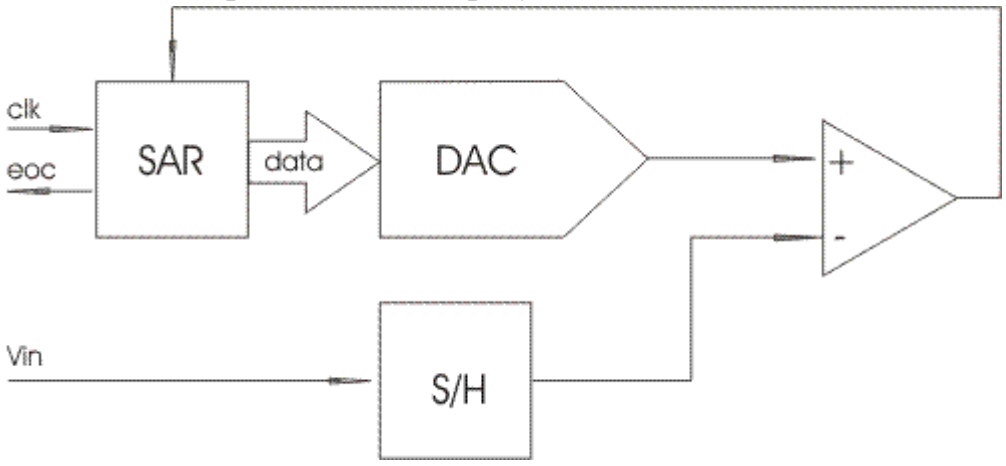


Рис. 7.5. Структурная схема АЦП последовательного приближения.

Таким образом, АЦП последовательного приближения состоит из следующих узлов:

1. Компаратор. Он сравнивает входную величину и текущее значение «весового» напряжения (на рис. 7.5. обозначен треугольником).

2. Цифро-аналоговый преобразователь (Digital to Analog Converter, DAC). Он генерирует «весовое» значение напряжения на основе поступающего на вход цифрового кода.

3. Регистр последовательного приближения (Successive Approximation Register, SAR). Он осуществляет алгоритм последовательного приближения, генерируя текущее значение кода, подающегося на вход ЦАП. По его названию названа вся данная архитектура АЦП.

4. Схема выборки-хранения (Sample/Hold, S/H). Для работы данного АЦП принципиально важно, чтобы входное напряжение сохраняло неизменную величину в течение всего цикла преобразования. Однако «реальные» сигналы имеют свойство изменяться во времени. Схема выборки-хранения «запоминает» текущее значение аналогового сигнала, и сохраняет его неизменным на протяжении всего цикла работы устройства.

Достоинством устройства является относительно высокая скорость преобразования: время преобразования N-битного АЦП составляет N тактов.

Точность преобразования ограничена точностью внутреннего ЦАП и может составлять 16-18 бит (сейчас стали появляться и 24-битные SAR ADC, например, AD7766 и AD7767).

Сигма-дельта АЦП

Для проведения большинства измерений часто не требуется АЦП со скоростью преобразования, которую дает АЦП последовательного приближения, зато необходима большая разрешающая способность. Сигма-дельта АЦП могут обеспечивать разрешающую способность до 24 разрядов, но при этом уступают в скорости преобразования. Так, в сигма-дельта АЦП при 16 разрядах можно получить частоту дискретизации до 100К отсчетов/с, а при 24 разрядах эта частота падает до 1К отсчетов/с и менее, в зависимости от устройства.

Обычно сигма-дельта АЦП применяются в разнообразных системах сбора данных и в измерительном оборудовании (измерение давления, температуры, веса и т. п.), когда не требуется высокая частота дискретизации и необходимо разрешение более 16 разрядов.

Структурная схема сигма-дельта АЦП приведена на рис. 7.6.

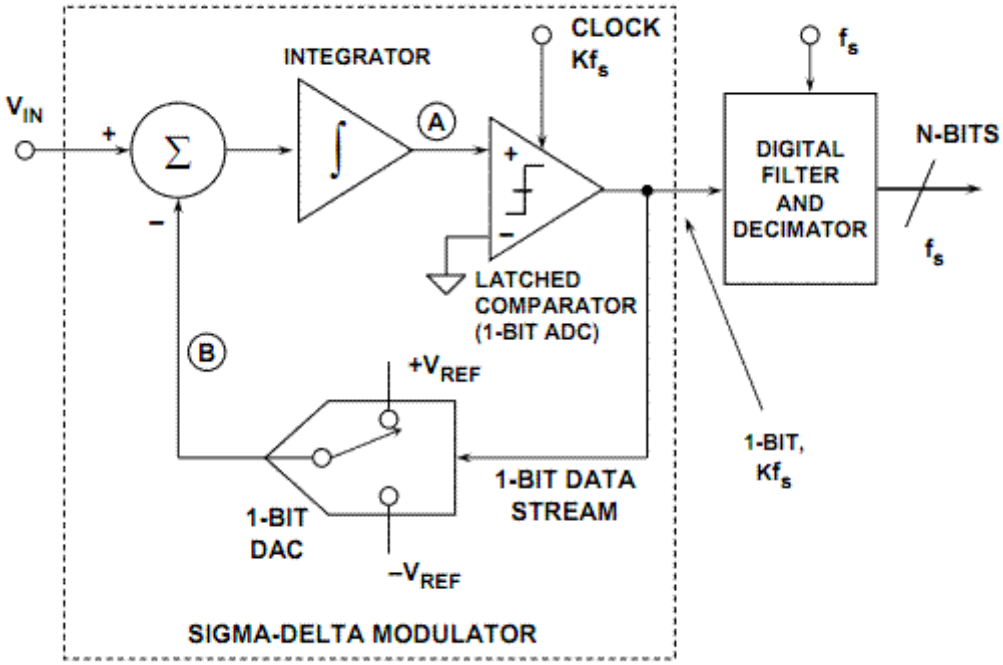


Рис.7.6. Структурная схема сигма-дельта АЦП.

Принцип действия данного АЦП несколько более сложен, чем у других типов АЦП. Его суть в том, что входное напряжение сравнивается со значением напряжения, накопленным интегратором. На вход интегратора подаются импульсы положительной или отрицательной полярности, в зависимости от результата сравнения. Таким образом, данный АЦП представляет собой простую следящую систему: напряжение на выходе

интегратора «отслеживает» входное напряжение (рис. 7.7). Результатом работы данной схемы является поток нулей и единиц на выходе компаратора, который затем пропускается через цифровой ФНЧ, в результате получается N-битный результат. ФНЧ на рис. 7.6. Объединен с «дециматором», устройством, снижающим частоту следования отсчетов путем их «прореживания».

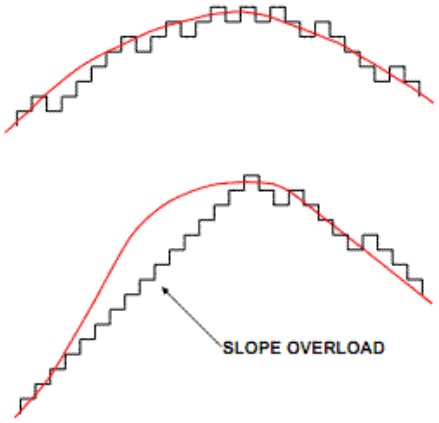


Рис. 7.7. Сигма-дельта АЦП как следящая система

На рис. 7.7 изображена структурная схема сигма-дельта АЦП первого порядка. Сигма-дельта АЦП второго порядка имеет два интегратора и две петли обратной связи.

Дополнительным и очень важным достоинством сигма-дельта АЦП является то, что все его внутренние узлы могут быть выполнены интегральным способом на площади одного кремниевого кристалла. Это заметно снижает стоимость конечных устройств и повышает стабильность характеристик АЦП.

Контрольные вопросы по теме

Уровень модуля

Уровень курса

1. Классы сигналов.
2. Аналого-цифровое преобразование сигналов.
3. Аналогово-цифровые преобразователи и их классификация.
4. Аналогово-цифровые преобразователи прямого преобразования.
5. Аналогово-цифровые преобразователи последовательного приближения.
6. Сигма-дельта аналогово-цифровые преобразователи.

Лекция № 8

Тема: Аналогово-цифровые и цифро-аналоговые преобразователи. Часть вторая.

Оглавление

Источники:

1. Таненбаум Э., Остин Т. Архитектура компьютера. 6-е изд. — СПб.: Питер, 2013. — 816 с.: ил.

https://rutracker.org/forum/viewtopic.php?t=4956359

Спецификация АЦП

Существуют общие определения, которые принято использовать в отношении аналого-цифровых преобразователей. Тем не менее характеристики, приводимые в технической документации производителей АЦП, могут показаться довольно путаными. Правильный же выбор оптимального по сочетанию своих характеристик АЦП для конкретного приложения требует точной интерпретации данных, приводимых в технической документации.

Наиболее часто путаемыми параметрами являются разрешающая способность и точность, хотя эти две характеристики реального АЦП крайне слабо связаны между собой. Разрешение не идентично точности, 12-разрядный АЦП может иметь меньшую точность, чем 8-разрядный. Для АЦП разрешение представляет собой меру того, на какое количество сегментов может быть поделен входной диапазон измеряемого аналогового сигнала (например, для 8-разрядного АЦП это $2^8 = 256$ сегментов). Точность же характеризует суммарное отклонение результата преобразования от своего идеального значения для данного входного напряжения. То есть разрешающая способность характеризует потенциальные возможности АЦП, а совокупность точностных параметров определяет реализуемость такой потенциальной возможности.

АЦП преобразует входной аналоговый сигнал в выходной цифровой код. Для реальных преобразователей, изготавливаемых в виде интегральных микросхем, процесс преобразования не является идеальным: на него оказывают влияние как технологический разброс параметров при производстве, так и различные внешние помехи. Поэтому цифровой код на выходе АЦП определяется с погрешностью. В спецификации на АЦП указываются погрешности, которые дает сам преобразователь. Их обычно делят на статические и динамические. При этом именно конечное приложение определяет, какие характеристики АЦП будут считаться определяющими, самыми важными в каждом конкретном случае.

Статическая погрешность

В большинстве применений АЦП используют для измерения медленно изменяющегося низкочастотного сигнала (например, от датчика температуры, давления, от тензодатчика и т. п.), когда входное напряжение пропорционально относительно постоянной физической величине. Здесь основную роль играет статическая погрешность измерения. В спецификации АЦП этот тип погрешности определяют аддитивная погрешность (Offset), мультипликативная погрешность (Full-Scale), дифференциальная нелинейность (DNL),

интегральная нелинейность (INL) и погрешность квантования. Эти пять характеристик позволяют полностью описать статическую погрешность АЦП.

Идеальная передаточная характеристика АЦП

Передаточная характеристика АЦП — это функция зависимости кода на выходе АЦП от напряжения на его входе. Такой график представляет собой кусочно-линейную функцию из 2^N «ступеней», где N — разрядность АЦП. Каждый горизонтальный отрезок этой функции соответствует одному из значений выходного кода АЦП (рис. 8.1). Если соединить линиями начала этих горизонтальных отрезков (на границах перехода от одного значения кода к другому), то идеальная передаточная характеристика будет представлять собой прямую линию, проходящую через начало координат.

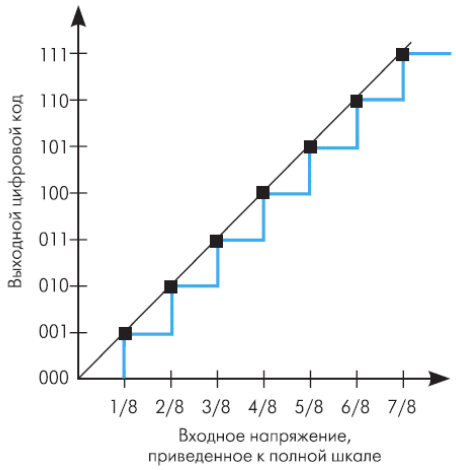


Рис. 8.1 Идеальная передаточная характеристика 3-разрядного АЦП

Рис. 8.1 иллюстрирует идеальную передаточную характеристику для 3-разрядного АЦП с контрольными точками на границах перехода кода. Выходной код принимает наименьшее значение (000b) при значении входного сигнала от 0 до 1/8 полной шкалы (максимального значения кода этого АЦП). Также следует отметить, что АЦП достигнет значения кода полной шкалы (111b) при 7/8 полной шкалы, а не при значении полной шкалы. Таким образом, переход в максимальное значение на выходе происходит не при напряжении полной шкалы, а при значении, меньшем на наименьший значащий разряд (LSB), чем входное напряжение полной шкалы. Передаточная характеристика может быть реализована со смещением -1/2 LSB. Это достигается смещением передаточной характеристики влево, что смещает погрешность квантования из диапазона -1...0 LSB в диапазон -1/2...+1/2 LSB.

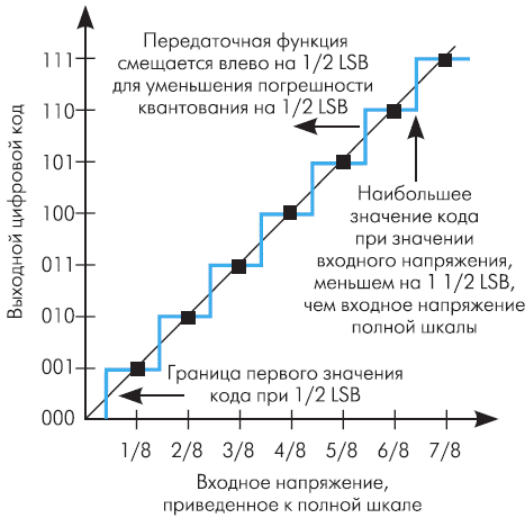


Рис. 8.2 Передаточная характеристика 3-разрядного АЦП со смещением на -1/2 LSB

Из-за технологического разброса параметров при изготовлении интегральных микросхем реальные АЦП не имеют идеальной передаточной характеристики. Отклонения от идеальной передаточной характеристики определяют статическую погрешность АЦП и приводятся в технической документации.

Аддитивная погрешность

Идеальная передаточная характеристика АЦП пересекает начало координат, а первый переход кода происходит при достижении значения 1 LSB. Аддитивная погрешность (погрешность смещения) может быть определена как смещение всей передаточной характеристики влево или вправо относительно оси входного напряжения, как показано на рис. 8.3. Таким образом, в определение аддитивной погрешности АЦП намеренно включено смещение 1/2 LSB.

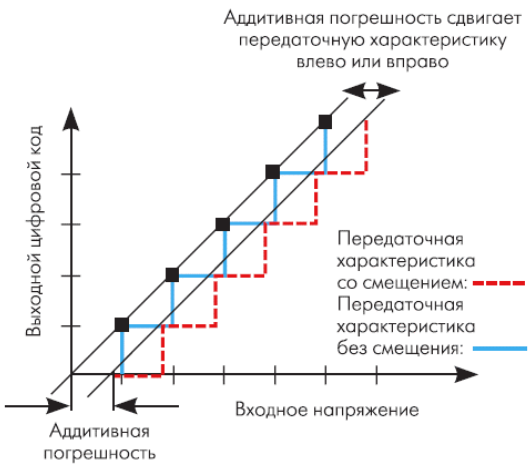


Рис. 8.3 Аддитивная погрешность

Мультипликативная погрешность

Мультипликативная погрешность (погрешность полной шкалы) представляет собой разность между идеальной и реальной передаточными характеристиками в точке максимального выходного значения при условии нулевой аддитивной погрешности (смещение отсутствует). Это проявляется как изменение наклона передаточной функции, что иллюстрирует рис. 8.4.

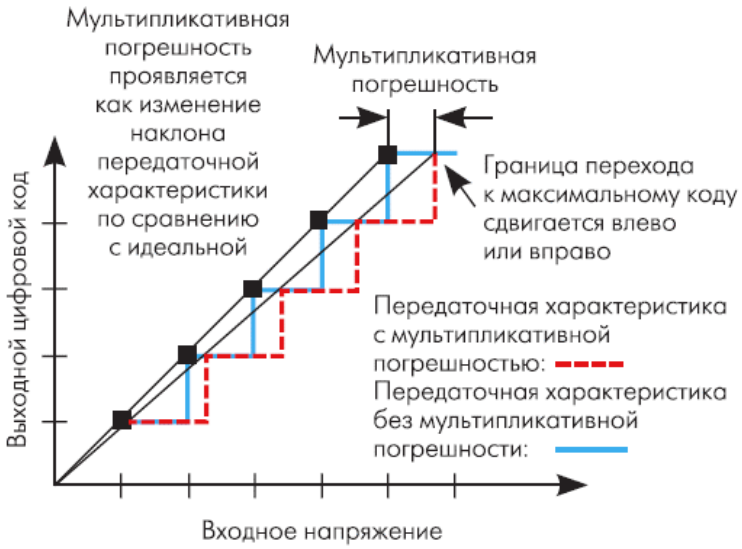


Рис. 8.4 Мультипликативная погрешность

Дифференциальная нелинейность

У идеальной передаточной характеристики АЦП ширина каждой «ступеньки» должна быть одинакова. Разница в длине горизонтальных отрезков этой кусочно-линейной функции из 2^N «ступеней» представляет собой дифференциальную нелинейность (DNL).

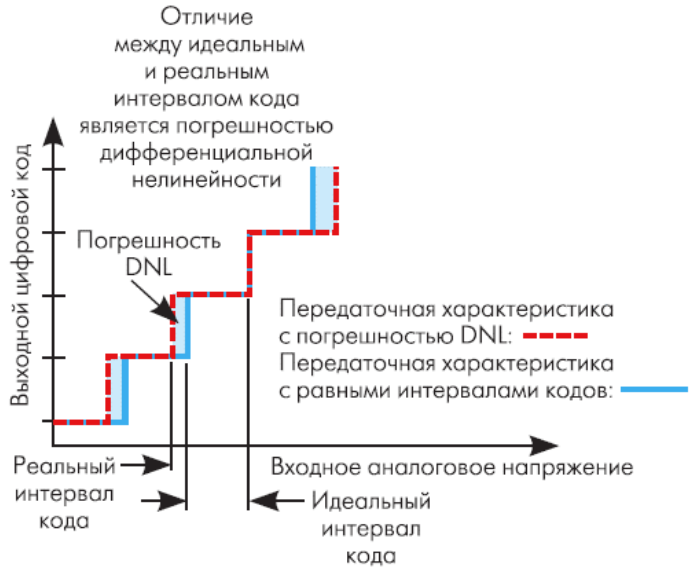


Рис. 8.5 Дифференциальная нелинейность

Величина наименьшего значащего разряда у АЦП составляет Uref/2, где Uref — опорное напряжение, N — разрешение АЦП. Разность напряжений между каждым кодовым переходом должна быть равна величине LSB. Отклонение этой разности от LSB определяется как дифференциальная нелинейность. На рисунке это показано как неравные промежутки между «шагами» кода или как «размытость» границ переходов на передаточной характеристике А ЦП.

Интегральная нелинейность

Интегральная нелинейность (INL) — это погрешность, которая вызывается отклонением линейной функции передаточной характеристики АЦП от прямой линии, как показано на рис. 8.6. Обычно передаточная функция с интегральной нелинейностью аппроксимируется прямой линией по методу наименьших квадратов. Часто аппроксимирующей прямой просто соединяют наименьшее и наибольшее значения. Интегральную нелинейность определяют путем сравнения напряжений, при которых происходят кодовые переходы. Для идеального АЦП эти переходы будут происходить при значениях входного напряжения, точно кратных LSB. А для реального преобразователя такое условие может выполняться с погрешностью. Разность между «идеальными» уровнями напряжения, при которых происходит кодовый переход, и их реальными значениями выражается в единицах LSB и называется интегральной нелинейностью.

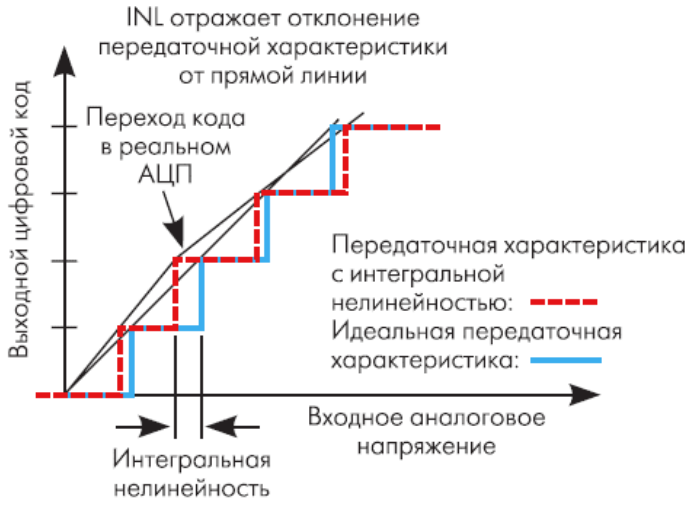


Рис. 8.6 Интегральная нелинейность

Погрешность квантования

Одна из наиболее существенных составляющих ошибки при измерениях с помощью АЦП — погрешность квантования — является результатом самого процесса преобразования. Погрешность квантования—это погрешность,

вызванная значением шага квантования и определяемая как 1/2 величины наименьшего значащего разряда (LSB). Она не может быть исключена в аналого-цифровых преобразованиях, так как является неотъемлемой частью процесса преобразования, определяется разрешающей способностью АЦП и не меняется от АЦП к АЦП с равным разрешением.

Динамические характеристики

Динамические характеристики АЦП обычно определяют с помощью спектрального анализа, по результатам выполнения быстрого преобразования Фурье (БПФ) над массивом выходных значений АЦП, соответствующих некоторому тестовому входному сигналу.

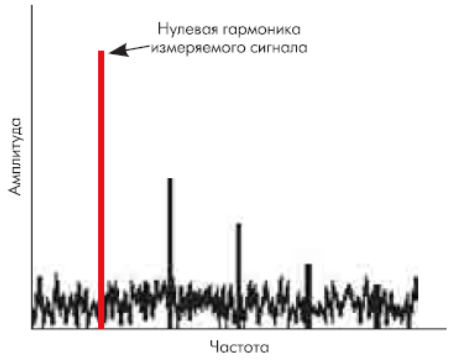


Рис. 8.7. Результат выполнения БПФ над выходными данными АЦП

Отношение «сигнал/шум»

Отношение «сигнал/шум» (SNR) — это отношение среднеквадратического значения величины входного сигнала к среднеквадратическому значению величины шума (за исключением гармонических искажений), выраженное в децибелах. Это значение позволяет определить долю шума в измеряемом сигнале по отношению к полезному сигналу.

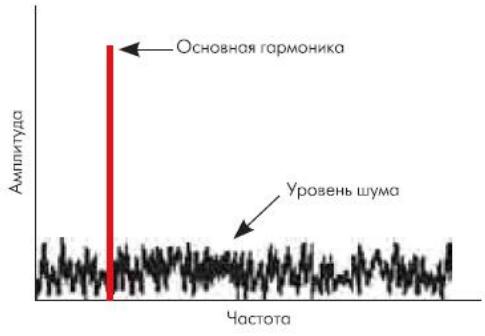


Рис. 8.8. Отношение «сигнал/шум»

Общие гармонические искажения

Нелинейность в результатах преобразования данных приводит к появлению гармонических искажений. Такие искажения наблюдаются как «выбросы» в спектре частот на четных и нечетных гармониках измеряемого сигнала. Эти искажения определяют как общие гармонические искажения (THD).

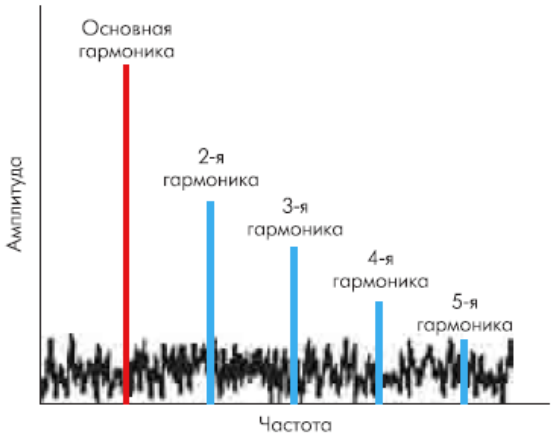


Рис. 8.9 Появление гармоник в выходных данных АЦП вследствие нелинейности преобразования

Отношение «сигнал/шум и искажения»

Отношение «сигнал/шум и искажения» (SiNAD) более полно описывает шумовые характеристики АЦП. SiNAD учитывает величину как шума, так и гармонических искажений по отношению к полезному сигналу.

Динамический диапазон, свободный от гармоник

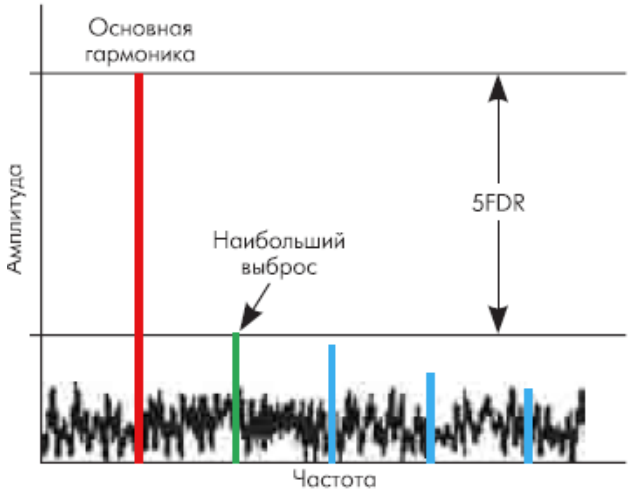


Рис. 8.10. Динамический диапазон, свободный от гармоник

Динамический диапазон, свободный от гармоник, представляет собой разницу между величиной измеряемого сигнала и наибольшим пиком

искажений (см. рис. 8.10). Этот динамический диапазон обозначается как SFDR. Он ограничен снизу амплитудой максимальной гармоники паразитных выбросов на выходе АЦП в диапазоне его рабочих частот.

Цифро-аналоговый преобразователь

Цифро-аналоговый преобразователь — устройство для перевода цифровых данных в аналоговый сигнал. Это своеобразный мост между аналоговой и цифровой частями схемы. Сфера применения ЦАП очень широка. Это — усилители звука, аудиокодеки, обработка видео, устройства отображения, системы распознавания данных, калибровка датчиков и других измерительных устройств, схемы управления двигателями, системы распределения данных, цифровые потенциометры, программируемое радио (SDR) и т.д.

Принцип работы

Принцип работы ЦАП заключается в суммировании аналоговых сигналов (ток или напряжение). Суммирование производится с коэффициентами, равными нулю или единице в зависимости от значения соответствующего разряда кода. Выходной сигнал ЦАП может иметь форму тока, напряжения или заряда. Табл. 1. Сигналы четырехразрядного ЦАП (опорное напряжение 5 В)

Входной код	Выходное напряжение, В
0000	0,0000
0001	0,3125
0010	0,6250
0011	0,9375
0100	1,2500
0101	1,5625
0110	1,8750
0111	2,1875
1000	2,5000
1001	2,8125
1010	3,1250
1011	3,4375
1100	3,7500
1101	4,0625
1110	4,3750
1111	4,6875

Преобразователи с токовым выходом используются в основном в прецизионных и высокочастотных схемах. Для определенности мы будем рассматривать ЦАП с выходным напряжением, как наиболее распространенные. Из таблицы 1 видно, что максимальное выходное напряжение на 1 МЗР (младший значащий разряд входного кода) ниже

напряжения полной шкалы (ПШ). Некоторые ЦАП позволяют использовать всю шкалу.

Характеристики ЦАП

Наиболее важные характеристики ЦАП — это разрядность, шаг квантования (разрешающая способность) и точность преобразования.

- Передаточная характеристика (ПХ) — зависимость выходного сигнала ЦАП от входных данных.
- Разрядность (N) — количество бит во входном коде.
- Разрешение — это выходное напряжение, соответствующее 1 МЗР. Оно зависит от количества разрядов и определяет точность преобразования сигнала.
- Частота дискретизации (частота Найквиста) — максимальная частота, на которой ЦАП может работать, выдавая на выходе корректный результат. В соответствии с теоремой Котельникова, для корректного воспроизведения аналогового сигнала из цифровой формы необходимо, чтобы частота дискретизации была не меньше удвоенной максимальной частоты в спектре сигнала.
- Полная шкала — диапазон значений выходного сигнала.
- Монотонность — участок на ПХ, где наклон постоянен. На этом участке ЦАП линеен.
- Время установления — интервал времени от момента изменения входного кода до окончательного вхождения выходного сигнала в заданный диапазон отклонения.
- Выходной выброс — это переходный процесс, возникающий во время смены входных данных. Величина выброса зависит от количества переключаемых разрядов.
- Погрешность смещения нуля — разность между фактическим и идеальным выходным сигналом, когда на входе ноль.
- Погрешность ПШ — разница между фактическим выходным напряжением и напряжением ПШ.
- Погрешность усиления — отклонение наклона ПХ от идеального.
- Дифференциальная нелинейность — разность приращений выходных сигналов, соответствующих смежным соседним кодам.
- Интегральная нелинейность — максимальное отклонение реальной ПХ от прямой линии.

Классификация

Цифро-аналоговые преобразователи делятся по типу входных данных на последовательные и параллельные. По разрядности выделяют ЦАП с повышенной точностью (большая разрядность, $N \geq 14$) или с высоким

быстродействием (6—8 разрядов). Выходной сигнал может иметь форму напряжения, тока или заряда.

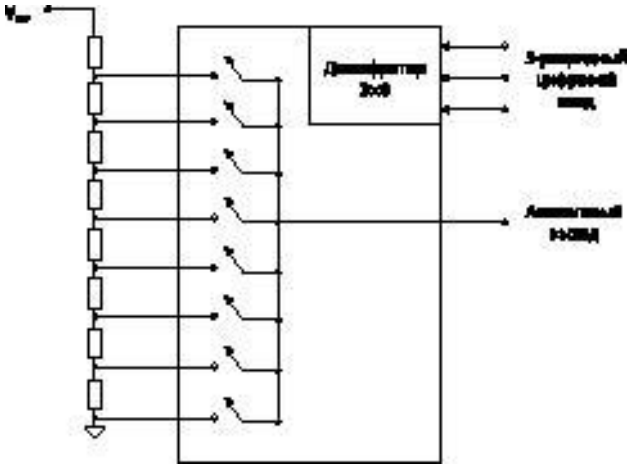


Рис. 8.11. Структура взвешивающего ЦАП

Рассмотрим некоторые структуры ЦАП. Простейшим ЦАП является взвешивающий (делитель Кельвина), структура которого показана на рисунке 1. Каждому биту преобразуемого двоичного кода соответствует резистор или источник тока, подключенный на общую точку суммирования. Сила тока источника (или проводимость резистора) пропорциональна весу бита, которому он соответствует. N-разрядный ЦАП содержит 2N одинаковых последовательно соединенных резистора и 2N ключа (обычно КМОП), по одному между каждым узлом цепи и выходом.

Взвешивающий метод — один из самых быстрых, однако характеризуется наименьшей точностью. Обычно такой ЦАП имеет выход по напряжению и отличается хорошей монотонностью. Если все резисторы одинаковы, ЦАП линеен. Недостаток данной модели — относительно высокий выходной импеданс и большое количество резисторов и ключей.

ЦАП на матрице R–2R. Это одна из наиболее распространенных структур (см. рис. 8.12). Здесь используются только две величины сопротивлений, находящихся в отношении 2:1. Количество резисторов равно 2N. Резистивный делитель можно использовать в качестве ЦАП двумя способами, в режиме напряжения и режиме тока (они также известны как нормальный и инверсный режимы). Главное преимущество ЦАП с выходом по напряжению заключается в том, что выходной импеданс постоянен. Второе достоинство — отсутствие емкостных токов в нагрузке. Недостатки данной структуры: во-первых, опорный источник должен иметь очень низкий импеданс; во-вторых, для регулирования усиления нельзя использовать резистор, включенный последовательно с опорным источником. В токовом режиме это допустимо, однако выбросы в токовой схеме больше. С другой

стороны, ключи находятся под потенциалом земли, поэтому защита от большого перепада напряжений не требуется.

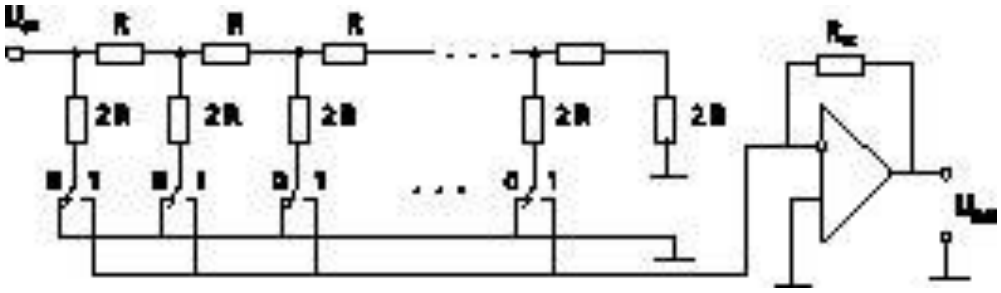


Рис. 8.12. ЦАП на R–2R матрице с выходом в форме напряжения

В сигма-дельта ЦАП преобразование осуществляется с помощью сигма-дельта модуляции, когда квантование осуществляется всего одним разрядом, но с частотой, в десятки и сотни раз превышающей частоту Найквиста. Сигма-дельта модулятор преобразует входной сигнал в последовательный непрерывный поток нулей и единиц. Если входной сигнал близок к положительному краю полной шкалы, в битовом потоке на выходе больше единиц, чем нулей, и наоборот, если сигнал ближе к отрицательному краю, то больше нулей. Для сигнала, близкого к середине шкалы, количество нулей и единиц примерно одинаково.

Интерполяционный фильтр представляет собой цифровую схему, которая принимает данные, поступающие с низкой частотой дискретизации, вставляет нули в поток данных, увеличивая тем самым частоту дискретизации, затем применяет алгоритм интерполяции и выдает данные с высокой частотой дискретизации. Выходное напряжение одноразрядного ЦАП переключается между равными по значению положительным и отрицательным опорными напряжениями. Выход фильтруется аналоговым ФНЧ.

Подбор ЦАП

Для выбора подходящего ЦАП необходимо определить требования, которым должны соответствовать его параметры. В первую очередь это — разрядность, разрешение, время установления выходного сигнала (быстродействие), интерфейс подключения, напряжение питания и т.д.

Разрядность ЦАП и величина опорного напряжения определяют шаг изменения выходного сигнала. Время установления определяет быстродействие ЦАП. При работе с постоянными или низкочастотными сигналами этот параметр не имеет большого значения. Однако им нельзя пренебрегать при работе на ВЧ.

Нелинейности бывают двух типов: интегральная и дифференциальная. Линейный ЦАП работает как зеркало, точно отражая входные данные. Влияние нелинейностей проиллюстрировано рисунком 5. Как правило, эти

искажения следует учитывать в прецизионных схемах, таких как системы калибровки или измерительное оборудование.

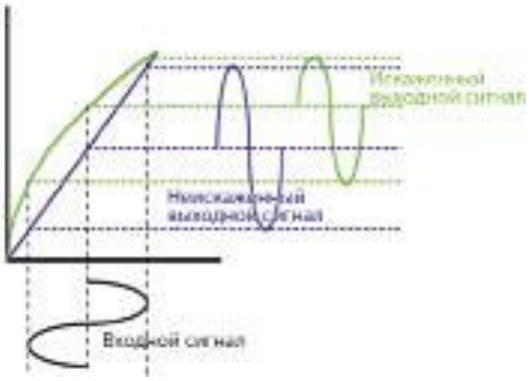


Рис. 8.13. Нелинейные искажения выходного сигнала

Для работы ЦАП нужно два источника напряжения (питания и опорное). В некоторых схемах для них используется один вывод, однако в этом случае точность ИП должна быть очень высокой, не хуже 1%. Преобразователи с раздельными выводами имеют более сложную схему, однако они не так требовательны к выбору ИП.

Контрольные вопросы по теме

Уровень модуля

Уровень курса

1. Спецификация аналогово-цифрового преобразователя.
2. Идеальная передаточная характеристика АЦП.
3. Аддитивная и мультипликативная погрешность АЦП.
4. Дифференциальная и интегральная нелинейность АЦП.
5. Принцип работы цифро-аналогового преобразователя.
6. Основные характеристики цифро-аналоговых преобразователей.
7. Классификация цифро-аналоговых преобразователей.

Лекция № 9
Тема: Датчики

Оглавление

Источники:

1. Таненбаум Э., Остин Т. Архитектура компьютера. 6-е изд. — СПб.: Питер, 2013. — 816 с.: ил.

   https://rutracker.org/forum/viewtopic.php?t=4956359

2. Бабак В.П. Теоретические основы информационно-измерительных систем: Учебник / В.П. Бабак, С.В. Бабак, В.С. Ерёменко и др. – К., 2014. – 832 с.

Классификация датчиков, основные требования к ним

Автоматизация различных технологических процессов, эффективное управление различными агрегатами, машинами, механизмами требуют многочисленных измерений разнообразных физических величин.

Датчики (в литературе часто называемые также измерительными преобразователями), или по-другому, **сенсоры** являются элементами многих систем автоматики - с их помощью получают информацию о параметрах контролируемой системы или устройства.

Датчик – это элемент измерительного, сигнального, регулирующего или управляющего устройства, преобразующий контролируемую величину (температуру, давление, частоту, силу света, электрическое напряжение, ток и т.д.) в сигнал, удобный для измерения, передачи, хранения, обработки, регистрации, а иногда и для воздействия им на управляемые процессы. Или проще, датчик – это устройство, преобразующее входное воздействие любой физической величины в сигнал, удобный для дальнейшего использования.

Используемые датчики весьма разнообразны и могут быть **классифицированы по различным признакам**:

В зависимости от вида входной (измеряемой) величины различают: датчики механических перемещений (линейных и угловых), пневматические, электрические, расходомеры, датчики скорости, ускорения, усилия, температуры, давления и др.

В настоящее время существует приблизительно следующее распределение доли измерений различных физических величин в промышленности: температура – 50%, расход (массовый и объемный) – 15%, давление – 10%, уровень – 5%, количество (масса, объем) – 5%, время – 4%, электрические и магнитные величины – менее 4%.

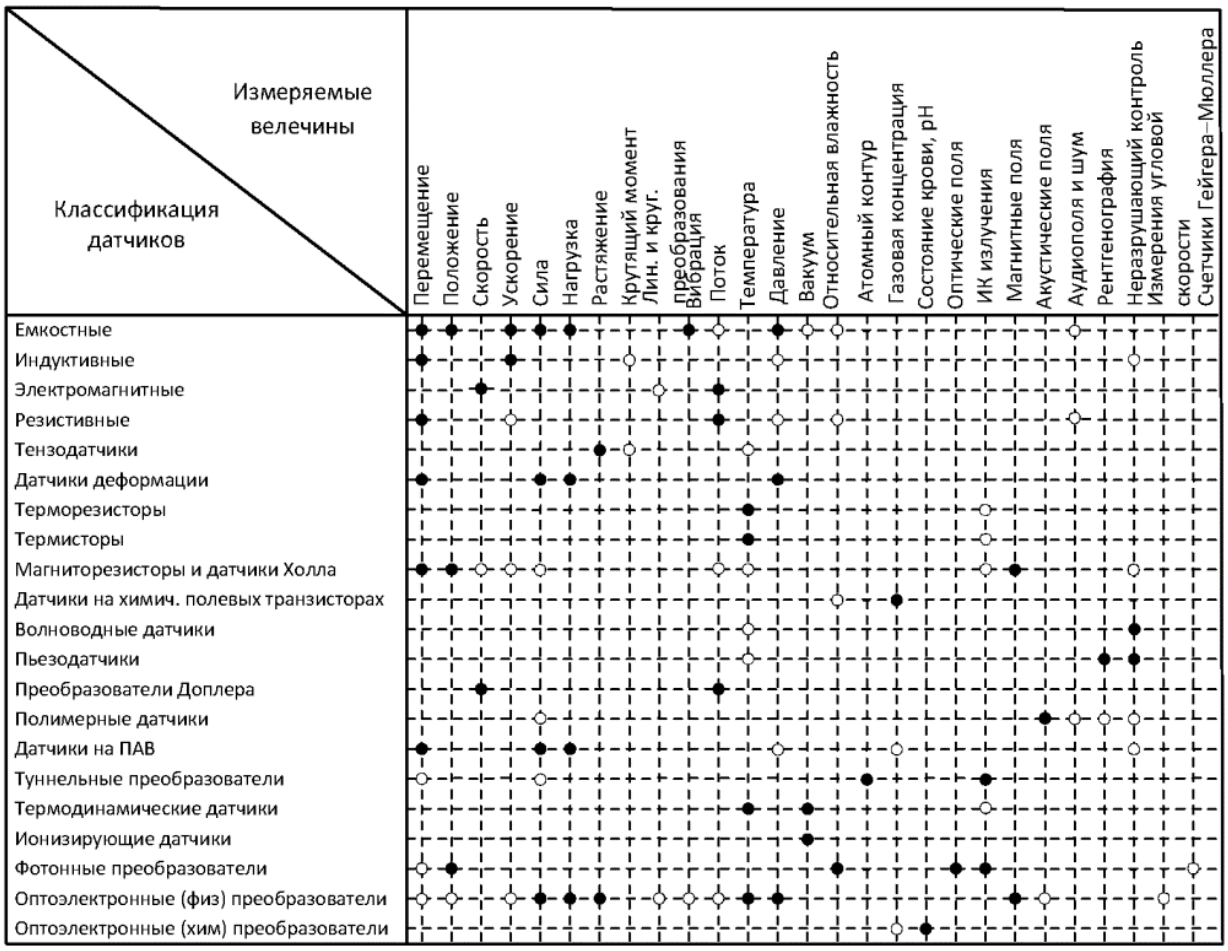
По виду выходной величины, в которую преобразуется входная величина, различают *неэлектрические* и *электрические*: датчики постоянного тока (ЭДС или напряжения), датчики амплитуды переменного тока (ЭДС или напряжения), датчики частоты переменного тока (ЭДС или напряжения), датчики сопротивления (активного, индуктивного или емкостного) и др.

Большинство датчиков являются электрическими. Это обусловлено следующими достоинствами электрических измерений:

- электрические величины удобно передавать на расстояние, причем передача осуществляется с высокой скоростью;

- электрические величины универсальны в том смысле, что любые другие величины могут быть преобразованы в электрические и наоборот;

- они точно преобразуются в цифровой код и позволяют достигнуть высокой точности, чувствительности и быстродействия средств измерений.



По принципу действия датчики можно разделить на два класса: *генераторные* и *параметрические* (датчики-модуляторы). Генераторные датчики осуществляют непосредственное преобразование входной величины в электрический сигнал.

Параметрические датчики входную величину преобразуют в изменение какого-либо электрического параметра (R, L или C) датчика.

По принципу действия датчики также можно разделить на омические, реостатные, фотоэлектрические (оптико-электронные), индуктивные, емкостные и д.р.

Различают три класса датчиков:

- аналоговые датчики, т. е. датчики, вырабатывающие аналоговый сигнал, пропорционально изменению входной величины;

- цифровые датчики, генерирующие последовательность импульсов или двоичное слово;

- бинарные (двоичные) датчики, которые вырабатывают сигнал только двух уровней: "включено/выключено" (иначе говоря, 0 или 1); получили широкое распространение благодаря своей простоте.

Требования, предъявляемые к датчикам:

- однозначная зависимость выходной величины от входной;

- стабильность характеристик во времени;

- высокая чувствительность;
- малые размеры и масса;
- отсутствие обратного воздействия на контролируемый процесс и на контролируемый параметр;
- работа при различных условиях эксплуатации;
- различные варианты монтажа.

Параметрические датчики

Параметрические датчики (датчики-модуляторы) входную величину X преобразуют в изменение какого-либо электрического параметра (R, L или C) датчика. Передать на расстояние изменение перечисленных параметров датчика без энергонесущего сигнала (напряжения или тока) невозможно. Выявить изменение соответствующего параметра датчика только и можно по реакции датчика на ток или напряжение, поскольку перечисленные параметры и характеризуют эту реакцию. Поэтому параметрические датчики требуют применения специальных измерительных цепей с питанием постоянным или переменным током.

Омические (резистивные) датчики – принцип действия основан на изменении их активного сопротивления при изменении длины *l*, площади сечения *S* или удельного сопротивления *p*:

$$R= pl/S$$

Кроме того, используется зависимость величины активного сопротивления от контактного давления и освещённости фотоэлементов. В соответствии с этим омические датчики делят на: *контактные, потенциометрические (реостатные), тензорезисторные, терморезисторные, фоторезисторные*.

Контактные датчики — это простейший вид резисторных датчиков, которые преобразуют перемещение первичного элемента в скачкообразное изменение сопротивления электрической цепи. С помощью контактных датчиков измеряют и контролируют усилия, перемещения, температуру, размеры объектов, контролируют их форму и т. д. К контактным датчикам относятся *путевые* и *концевые выключатели*, *контактные термометры* и так называемые *электродные датчики*, используемые в основном для измерения предельных уровней электропроводных жидкостей.

Контактные датчики могут работать как на постоянном, так и на переменном токе. В зависимости от пределов измерения контактные датчики могут быть одно предельными и многопредельными. Последние используют для измерения величин, изменяющихся в значительных пределах, при этом части резистора R, включенного в электрическую цепь, последовательно

закорачиваются. Недостаток контактных датчиков — сложность осуществления непрерывного контроля и ограниченный срок службы контактной системы. Но благодаря предельной простоте этих датчиков их широко применяют в системах автоматики.

Реостатные датчики представляют собой резистор с изменяющимся активным сопротивлением. Входной величиной датчика является перемещение контакта, а выходной – изменение его сопротивления. Подвижный контакт механически связан с объектом, перемещение (угловое или линейное) которого необходимо преобразовать.

Наибольшее распространение получила потенциометрическая схема включения реостатного датчика, в которой реостат включают по схеме делителя напряжения. Напомним, что делителем напряжения называют электротехническое устройство для деления постоянного или переменного напряжения на части; делитель напряжения позволяет снимать (использовать) только часть имеющегося напряжения посредством элементов электрической цепи, состоящей из резисторов, конденсаторов или катушек индуктивности. Переменный резистор, включаемый по схеме делителя напряжения, называют потенциометром.

Обычно реостатные датчики применяют в механических измерительных приборах для преобразования их показаний в электрические величины (ток или напряжение), например, в поплавковых измерителях уровня жидкостей, различных манометрах и т. п.

Датчик в виде простого реостата почти не используется вследствие значительной нелинейности его статической характеристики $I_н = f(x)$, где $I_н$ - ток в нагрузке.

Выходной величиной такого датчика является падение напряжения $U_{вых}$ между подвижным и одним из неподвижных контактов. Зависимость выходного напряжения от перемещения x контакта $U_{вых} = f(x)$ соответствует закону изменения сопротивления вдоль потенциометра. Закон распределения сопротивления по длине потенциометра, определяемый его конструкцией, может быть линейным или нелинейным.

Потенциометрические датчики, конструктивно представляющие собой переменные резисторы, выполняют из различных материлов — обмоточного провода, металлических пленок, полупроводников и т. д.

Тензорезисторы (*тензометрические датчики*) служат для измерения механических напряжений, небольших деформаций, вибрации. Действие тензорезисторов основано на тензоэффекте, заключающемся в изменении активного сопротивления проводниковых и полупроводниковых материалов под воздействием приложенных к ним усилий.

Термометрические датчики (*терморезисторы*) - сопротивление зависит от температуры. Терморезисторы в качестве датчиков используют двумя способами:

1) Температура терморезистора определяется окружающей средой; ток, проходящий через терморезистор, настолько мал, что не вызывает нагрева терморезистора. При этом условии терморезистор используется как датчик температуры и часто называется «термометром сопротивления».

2) Температура терморезистора определяется степенью нагрева постоянным по величине током и условиями охлаждения. В этом случае установившаяся температура определяется условиями теплоотдачи поверхности терморезистора (скоростью движения окружающей среды – газа или жидкости – относительно терморезистора, ее плотностью, вязкостью и температурой), поэтому терморезистор может быть использован как датчик скорости потока, теплопроводности окружающей среды, плотности газов и т. п. В датчиках такого рода происходит как бы двухступенчатое преобразование: измеряемая величина сначала преобразуется в изменение температуры терморезистора, которое затем преобразуется в изменение сопротивления.

Терморезисторы изготовляют как из чистых металлов, так и из полупроводников.Материал, из которого изготавливается такие датчики, должен обладать высоким температурным коэффициентом сопротивления, по возможности линейной зависимостью сопротивления от температуры, хорошей воспроизводимостью свойств и инертностью к воздействиям окружающей среды. В наибольшей степени всем указанным свойствам удовлетворяет платина; в чуть меньшей – медь и никель.

По сравнению с металлическими терморезисторами более высокой чувствительностью обладают полупроводниковые терморезисторы (термисторы).

Индуктивные датчики служат для бесконтактного получения информации о перемещениях рабочих органов машин, механизмов, роботов и т.п. и преобразования этой информации в электрический сигнал.

Принцип действия индуктивного датчика основан на изменении индуктивности обмотки на магнитопроводе в зависимости от положения отдельных элементов магнитопровода (якоря, сердечника и др.). В таких датчиках линейное или угловое перемещение X (входная величина) преобразуется в изменение индуктивности (L) датчика. Применяются для измерения угловых и линейных перемещений, деформаций, контроля размеров и т.д.

В простейшем случае индуктивный датчик представляет собой катушку индуктивности с магнитопроводом, подвижный элемент которого (якорь) перемещается под действием измеряемой величины.

Индуктивный датчик распознает и соответственно реагирует на все токопроводящие предметы. Индуктивный датчик является бесконтактным, не требует механичесого воздействия, работает бесконтактно за счет изменения электромагнитного поля.

Преимущества

- нет механического износа, отсутствуют отказы, связанные с состоянием контактов
- отсутствует дребезг контактов и ложные срабатывания
- высокая частота переключений до 3000 Hz
- устойчив к механическим воздействиям

Недостатки - сравнительно малая чувствительность, зависимость индуктивного сопротивления от частоты питающего напряжения, значительное обратное воздействие датчика на измеряемую величину (за счет притяжения якоря к сердечнику).

Емкостные датчики - принцип действия основан на зависимости электрической емкости конденсатора от размеров, взаимного расположения его обкладок и от диэлектрической проницаемости среды между ними.

Для двухобкладочного плоского конденсатора электрическая емкость определяется выражением:

$$C = \varepsilon_0 \varepsilon S/h$$

где ε_0 - диэлектрическая постоянная; ε - относительная диэлектрическая проницаемость среды между обкладками; S - активная площадь обкладок; h - расстояние между обкладками конденсатора.

Зависимости $C(S)$ и $C(h)$ используют для преобразования механических перемещений в изменение емкости.

Емкостные датчики, также как и индуктивные, питаются переменным напряжением (обычно повышенной частоты - до десятков мегагерц). В качестве измерительных схем обычно применяют мостовые схемы и схемы с использованием резонансных контуров. В последнем случае, как правило, используют зависимость частоты колебаний генератора от емкости резонансного контура, т.е. датчик имеет частотный выход.

Достоинства емкостных датчиков - простота, высокая чувствительность и малая инерционность. Недостатки - влияние внешних электрических полей, относительная сложность измерительных устройств.

Емкостные датчики применяют для измерения угловых перемещений, очень малых линейных перемещений, вибраций, скорости движения и т. д., а

также для воспроизведения заданных функций (гармонических, пилообразных, прямоугольных и т. п.).

Емкостные преобразователи, диэлектрическая проницаемость *е* которых изменяется за счет перемещения, деформации или изменения состава диэлектрика, применяют в качестве датчиков уровня непроводящих жидкостей, сыпучих и порошкообразных материалов, толщины слоя непроводящих материалов (толщиномеры), а также контроля влажности и состава вещества.

Датчики – генераторы

Генераторные датчики осуществляют непосредственное преобразование входной величины *X* в электрический сигнал. Такие датчики преобразуют энергию источника входной (измеряемой) величины сразу в электрический сигнал, т.е. они являются как бы генераторами электроэнергии (откуда и название таких датчиков - они генерируют электрический сигнал).

Дополнительные источники электроэнергии для работы таких датчиков принципиально не требуются (тем не менее дополнительная электроэнергия может потребоваться для усиления выходного сигнала датчика, его преобразования в другие виды сигналов и других целей). Генераторными являются термоэлектрические, пьезоэлектрические, индукционные, фотоэлектрические и многие другие типы датчиков.

Индукционные датчики преобразуют измеряемую неэлектрическую величину в ЭДС индукции. Принцип действия датчиков основан на законе электромагнитной индукции. К этим датчикам относятся тахогенераторы постоянного и переменного тока, представляющие собой небольшие электромашинные генераторы, у которых выходное напряжение пропорционально угловой скорости вращения вала генератора. Тахогенераторы используются как датчики угловой скорости.

Тахогенератор представляет собой электрическую машину, работающую в генераторном режиме. При этом вырабатываемая ЭДС пропорциональна скорости вращения и величине магнитного потока. Кроме того, с изменением скорости вращения изменяется частота ЭДС. Применяются как датчики скорости (частоты вращения).

Температурные датчики.

В современном промышленном производстве наиболее распространенными являются измерения температуры (так, на атомной электростанции среднего размера имеется около 1500 точек, в которых производится такое измерение, а на крупном предприятии химической промышленности подобных точек присутствует свыше 20 тыс.). Широкий диапазон измеряемых температур, разнообразие условий использования

средств измерений и требований к ним определяют многообразие применяемых средств измерения температуры.

Если рассматривать датчики температуры для промышленного применения, то можно выделить их основные классы: кремниевые датчики температуры, биметаллические датчики, жидкостные и газовые термометры, термоиндикаторы, термисторы, термопары, термопреобразователи сопротивления, инфракрасные датчики.

Кремниевые датчики температуры используют зависимость сопротивления полупроводникового кремния от температуры. Диапазон измеряемых температур -50…+150 ^{0}C. Применяются в основном для измерения температуры внутри электронных приборов.

Биметаллический датчик сделан из двух разнородных металлических пластин, скрепленных между собой. Разные металлы имеют различный температурный коэффициент расширения. Если соединенные в пластину металлы нагреть или охладить, то она изогнется, при этом замкнет (разомкнет) электрические контакты или переведет стрелку индикатора. Диапазон работы биметаллических датчиков -40…+550 ^{0}C. Используются для измерения поверхности твердых тел и температуры жидкостей. Основные области применения – автомобильная промышленность, системы отопления и нагрева воды.

Термоиндикаторы – это особые вещества, изменяющие свой цвет под воздействием температуры. Изменение цвета может быть обратимым и необратимым. Производятся в виде пленок.

Термопреобразователи сопротивления. Принцип действия термопреобразователей сопротивления (терморезисторов) основан на изменении электрического сопротивления проводников и полупроводников в зависимости от температуры (рассмотрен ранее).

Платиновые терморезисторы предназначены для измерения температур в пределах от –260 до 1100 ^{0}C. Широкое распространение на практике получили более дешевые медные терморезисторы, имеющие линейную зависимость сопротивления от температуры.

Недостатком меди является небольшое ее удельное сопротивление и легкая окисляемость при высоких температурах, вследствие чего конечный предел применения медных термометров сопротивления ограничивается температурой 180 ^{0}C. По стабильности и воспроизводимости характеристик медные терморезисторы уступают платиновым. Никель используется в недорогих датчиках для измерения в диапазоне комнатных температур.

Полупроводниковые терморезисторы (термисторы) имеют отрицательный или положительный температурный коэффициент сопротивления, значение которого при 20 ^{0}C составляет $(2…8)*10^{-2}$ $(^0C)^{-1}$, т.е.

на порядок больше, чем у меди и платины. Полупроводниковые терморезисторы при весьма малых размерах имеют высокие значения сопротивления (до 1 МОм). В качестве полупров. материала используются оксиды металлов: полупроводниковые терморезисторы типов КМТ - смесь окислов кобальта и марганца и ММТ - меди и марганца.

Полупроводниковые датчики температуры обладают высокой стабильностью характеристик во времени и применяются для изменения температур в диапазоне от –100 до 200 ^{0}C.

Термоэлектрические преобразователи (термопары) - принцип действия термопар основан на термоэлектрическом эффекте, который состоит в том, что при наличии разности температур мест соединений (спаев) двух разнородных металлов или полупроводников в контуре возникает электродвижущая сила, называемая термоэлектродвижущей (сокращенно термо-ЭДС). В определенном интервале температур можно считать, что термо-ЭДС прямо пропорциональна разности температур $\Delta T = T_1 – T_0$ между спаем и концами термопары.

Соединенные между собой концы термопары, погружаемые в среду, температура которой измеряется, называют рабочим концом термопары. Концы, которые находятся в окружающей среде, и которые обычно присоединяют проводами к измерительной схеме, называют свободными концами. Температуру этих концов необходимо поддерживать постоянной. При этом условии термо-ЭДС $E_т$ будет зависеть только от температуры T_1 рабочего конца.

$U_{вых} = E_т = C(T_1 – T_0)$, где C – коэффициент, зависящий от материала проводников термопары.

Создаваемая термопарами ЭДС сравнительно невелика: она не превышает 8 мВ на каждые 100 ^{0}C и обычно не превышает по абсолютной величине 70 мВ. Термопары позволяют измерять температуру в диапазоне от –200 до 2200 ^{0}C.

Наибольшее распространение для изготовления термоэлектрических преобразователей получили платина, платинородий, хромель, алюмель.

Термопары имеют следующие преимущества: простота изготовления и надёжность в эксплуатации, дешевизна, отсутствие источников питания и возможность измерений в большом диапазоне температур.

Наряду с этим термопарам свойственны и некоторые недостатки - меньшая, чем у терморезисторов, точность измерения, наличие значительной тепловой инерционности, необходимость введения поправки на температуру свободных концов и необходимость в применении специальных соединительных проводов.

Инфракрасные датчики (пирометры) - используют энергию излучения нагретых тел, что позволяет измерять температуру поверхности на расстоянии. Пирометры делятся на радиационные, яркостные и цветовые.

Радиационные пирометры используются для измерения температуры от 20 до 2500 ^{0}C, причем прибор измеряет интегральную интенсивность излучения реального объекта.

Яркостные (оптические) пирометры используются для измерения температур от 500 до 4000 ^{0}C. Они основаны на сравнении в узком участке спектра яркости исследуемого объекта с яркостью образцового излучателя (фотометрической лампы).

Цветовые пирометры основаны на измерении отношения интенсивностей излучения на двух длинах волн, выбираемых обычно в красной или синей части спектра; они используются для измерения температуры в диапазоне от 800 ^{0}C.

Пирометры позволяют измерять температуру в труднодоступных местах и температуру движущихся объектов, высокие температуры, где другие датчики уже не работают.

Кварцевые термопреобразователи

Для измерения температур от – 80 до 250 ^{0}C часто используются так называемые кварцевые термопреобразователи, использующие зависимость собственной частоты кварцевого элемента от температуры. Работа данных датчиков основана на том, что зависимость частоты преобразователя от температуры и линейность функции преобразования изменяются в зависимости от ориентации среза относительно осей кристалла кварца. Данные датчики широко используются в цифровых термометрах.

Пьезоэлектрические датчики

Действие пьезоэлектрических датчиков основано на использовании пьезоэлектрического эффекта (пьезоэффекта), заключающегося в том, что при сжатии или растяжении некоторых кристаллов на их гранях появляется электрический заряд, величина которого пропорциональна действующей силе.

Пьезоэффект обратим, т. е. приложенное электрическое напряжение вызывает деформацию пьезоэлектрического образца - сжатие или растяжение его соответственно знаку приложенного напряжения. Это явление, называемое обратным пьезоэффектом, используется для возбуждения и приема акустических колебаний звуковой и ультразвуковой частоты.

Используются для измерения сил, давления, вибрации и т.д.

Оптические (фотоэлектрические) датчики

Различают *аналоговые* и *дискретные* оптические датчики. У аналоговых датчиков выходной сигнал изменяется пропорционально внешней

освещенности. Основная область применения – автоматизированные системы управления освещением.

Датчики дискретного типа изменяют выходное состояние на противоположное при достижении заданного значения освещенности.

Фотоэлектрические датчики могут быть применены практически во всех отраслях промышленности. Датчики дискретного действия используются как своеобразные бесконтактные выключатели для подсчета, обнаружения, позиционирования и других задач на любой технологической линии.

Оптический бесконтактный датчик, регистрирует изменение светового потока в контролируемой области, связанное с изменением положения в пространстве каких-либо движущихся частей механизмов и машин, отсутствия или присутствия объектов. Благодаря большим расстояниям срабатывания оптические бесконтактные датчикинашли широкое применение в промышленности и не только.

Оптический бесконтактный датчик состоит из двух функциональных узлов, приемника и излучателя. Данные узлы могут быть выполнены как в одном корпусе, так и в различных корпусах.

По методу обнаружения объекта фотоэлектрические датчики подразделяются на 4 группы:

1) пересечение луча - в этом методе передатчик и приемник разделены по разным корпусам, что позволяет устанавливать их напротив друг друга на рабочем расстоянии. Принцип работы основан на том, что передатчик постоянно посылает световой луч, который принимает приемник. Если световой сигнал датчика прекращается, в следствии перекрытия сторонним объектом, приемник немедленно реагирует меняя состояние выхода.

2) отражение от рефлектора - в этом методе приемник и передатчик датчика находятся в одном корпусе. Напротив датчика устанавливается рефлектор (отражатель). Датчики с рефлектором устроены так, что благодаря поляризационному фильтру они воспринимают отражение только от рефлектора. Это рефлекторы, которые работают по принципу двойного отражения. Выбор подходящего рефлектора определяется требуемым расстоянием и монтажными возможностями.

Посылаемый передатчиком световой сигнал отражаясь от рефлектора попадает в приемник датчика. Если световой сигнал прекращается, приемник немедленно реагирует, меняя состояние выхода.

3) отражение от объекта - в этом методе приемник и передатчик датчика находятся в одном корпусе. Во время рабочего состояния датчика все объекты, попадающие в его рабочую зону, становятся своеобразными рефлекторами. Как только световой луч отразившись от объекта попадает на приемник датчика, тот немедленно реагирует, меняя состояние выхода.

4) фиксированное отражение от объекта -принцип действия датчика такой же как и у "отражение от объекта" но более чутко реагирующий на отклонение от настройки на объект. Например, возможно детектирование вздутой пробки на бутылке с кефиром, неполное наполнение вакуумной упаковки с продуктами и т.д.

По своему назначению фотодатчики делятся на две основные группы: датчики общего применения и специальные датчики. К специальным, относятся типы датчиков, предназначенные для решения более узкого круга задач. К примеру, обнаружение цветной метки на объекте, обнаружение контрастной границы, наличие этикетки на прозрачной упаковке и т.д.

Задача датчика обнаружить объект на расстоянии. Это расстояние варьируется в пределах 0,3мм-50м, в зависимости от выбранного типа датчика и метода обнаружения.

Контрольные вопросы по теме

Уровень модуля

Уровень курса

1. Классификация датчиков, основные требования к датчикам.
2. Резистивные и контактные датчики.
3. Реостатные датчики и тензорезисторы.
4. Индуктивные и емкостные датчики.
5. Датчики генераторного типа.
6. Температурные датчики.
7. Оптические датчики.

Лекция № 10

Тема: Датчики. Аналоговые интерфейсы.

Оглавление

Источники:

1. Таненбаум Э., Остин Т. Архитектура компьютера. 6-е изд. — СПб.: Питер, 2013. — 816 с.: ил.

   https://rutracker.org/forum/viewtopic.php?t=4956359

2. Бабак В.П. Теоретические основы информационно-измерительных систем: Учебник / В.П. Бабак, С.В. Бабак, В.С. Ерёменко и др. – К., 2014. – 832 с.

Стандартный сигнал

Первичные измерительные приборы - датчики (температуры, давления, веса, сопротивления, влажности, расхода и т.д.) реализуют тот или иной физический эффект. Поэтому, первичный выходной сигнал от них может быть, в принципе, любым. Не подвергнутые обработке сигналы от датчиков весьма разнообразны и диапазон их изменения простирается от нескольких милливольт (для термопары) до более чем сотни вольт для тахогенератора. Кроме того, они могут быть вызваны изменениями напряжения постоянного тока, переменного тока или даже сопротивления. Поэтому совершенно очевидно, что если аналоговые входные платы работают лишь в определенном диапазоне сигналов, то необходимо использовать некоторую стандартизацию. Поэтому в технике используют так называемые "датчики со стандартным сигналом выхода". Важно отметить, что "датчик со стандартным сигналом выхода" на самом деле состоит из первичного измерительного прибора и преобразователя первичного сигнала в стандартный токовый сигнал или стандартный сигнал напряжения (аналоговый интерфейс).

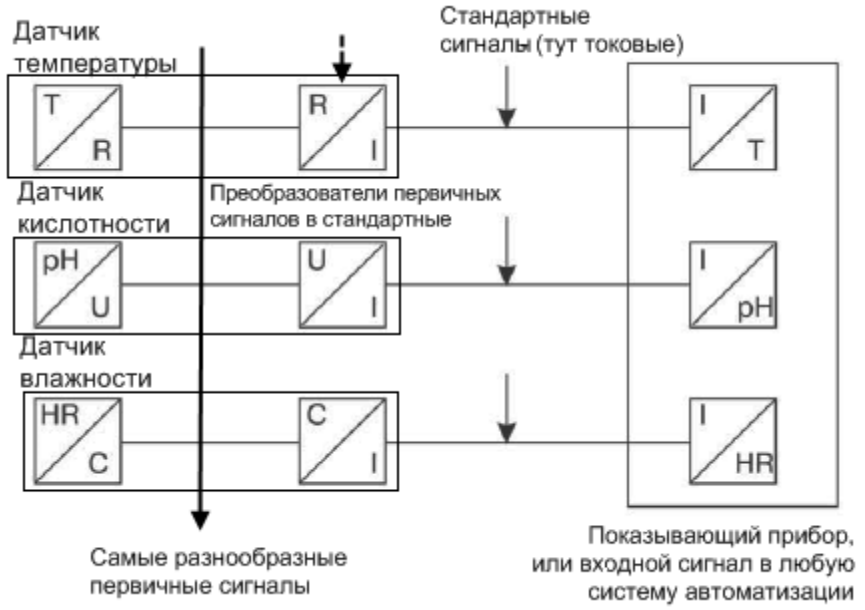


Рис. 1

Самый распространенный стандарт представляет аналоговый сигнал в виде тока с диапазоном изменения 4—20 мА, где 4мА соответствует минимальному уровню сигнала, а 20 мА — максимальному. Если, например, преобразователь давления дает сигнал 4—20 мА, представляющий давление в диапазоне 0—10 бар, то давлению 8 бар будет соответствовать величина тока 8 х (20 — 4)/10 + 4 = 16.8 мА. Сигнал 4—20 мА часто с помощью балластного резистора величиной 250 Ом преобразуется в сигнал 1 —5 В.

«Нулевой» сигнал 4 мА (называемый смещением) предназначен для двух целей. Во-первых, он используется как защита от повреждений преобразователя или кабельного шнура. Если происходит отказ преобразователя или обрыв шнура, или же в линии связи возникает короткое замыкание, то ток через балластный резистор будет равен нулю, что соответствует «отрицательному» сигналу 0 В на приемной стороне. Это может быть очень легко обнаружено и использовано как аварийный сигнал «неисправность преобразователя».

Ток смещения 4 мА также упрощает компоновку системы.

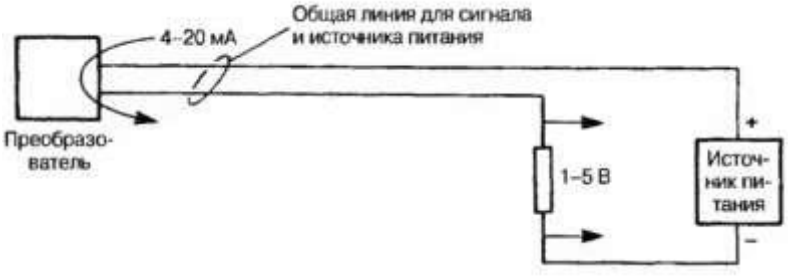


Рис. 2. Двухпроводным преобразователь 4-20 мА

Наиболее распространенной (и наиболее простой) является схема, изображенная на рис. 2. Здесь источник питания (обычно 24—30В постоянного тока) помещается на стороне приемного устройства, а сигнальные линии служат как для питания преобразователя, так и для передачи тока. Преобразователь отбирает от источника питания ток в диапазоне 4—20 мА в соответствии с измеряемым сигналом. Этот ток, как и раньше, преобразуется в напряжение с помощью балластного резистора.

Смещение в 4 мА обеспечивает ток, необходимый преобразователю для его нормальной работы. Очевидно, этого нельзя добиться, если диапазон сигнала будет составлять 0—20 мА. Преобразователи, включаемые по схеме рис. 2, обычно называются двухпроводными.

Требования к выходному сигналу приведены в стандарте ГОСТ 26.011-80. Средства измерений и автоматизации. Сигналы тока и напряжения электрические непрерывные входные и выходные.

Таблица 1. Требования к цепям аналоговых сигналов напряжения постоянного тока. Аналоговые сигналы напряжения.

Сигнал напряжения, В	**Нагрузочное сопротивление, Ом, не более**	**Входное сопротивление приемника Ом, не менее**
От 0 до 0, 01 включит	–	10000
От 0 до 1 включит.	–	10000
От 0 до 10 включит.	2000	–

Таблица 2. Требования к цепям аналоговых сигналов постоянного тока. Аналоговые токовые сигналы. Аналоговые токовые петли. Постоянный ток.

Сигнал тока, мА	**Выходное сопротивление источника, Ом, не менее**	**Входное сопротивление приемника Ом, не более**
От 0 до 5 включит	2500 (2000)	500
От 0 до 20 включит.	1000 (500)	250
От 4 до 20 включит.	1000 (500)	250

Аналоговый интерфейс

Аналоговый интерфейс (АИ) - это совокупность средств измерительной техники, являющихся составной частью измерительного канала между датчиком и АЦП, находящимися в информационно-измерительной (ИИС) системе сбора данных.

АИ являются одной из наиболее важных составных частей ИИС с точки зрения определения их метрологических характеристик.

АИ в ИИС выполняют следующие функции.

Масштабное преобразование - обеспечивает повышение отношения сигнал/шум на входе АЦП при наличии линии связи между передающей и приемной составляющими АИ. Усиление сигнала происходит в передающей части АИ в непосредственной близости от первичного преобразователя – собственно датчика.

Фильтрация необходима для ограничения полосы частот помех и подавления нежелательных частотных составляющих в полосе измерительного сигнала, в первую очередь сетевых наводок, а также для уменьшения влияния эффекта наложения спектров во время дискретизации аналоговых сигналов.

Изолирование достигается с помощью трансформаторных, емкостных или оптических гальванически развязывающих устройств и обеспечивает разрыв паразитных контуров заземления, возможность работы при значительных по величине синфазных составляющих сигнала.

Компенсация температуры холодных спаев термопар необходима для уменьшения методической погрешности термопары.

Линеаризация характеристик датчиков. В большинстве случаев функция преобразования датчика является нелинейной. Для уменьшения влияния аддитивных погрешностей последующих звеньев измерительного канала се линеаризуют, что достигается включением в передающую часть АИ программно-управляемых усилителей или функциональных преобразователей.

Инициализация пассивных датчиков. Большинство первичных преобразователей (например, термометры сопротивления, тензорезистивные, индуктивные, емкостные преобразователи) требуют дополнительной энергии в виде напряжения или тока для получения активного выходного электрического сигнала.

Разновидности аналоговых интерфейсов

Классификация АИ по типу электрических сигналов (типу информативного параметра или модуляции) представлена на рис. 8.5. Постоянный ток, благодаря высокой помехозащищенности и отсутствию в значительной степени влияния паразитных параметров линии связи, используется в качестве сигнала измерительной информации для распределенных в пространстве систем в случаях, когда расстояние между системой сбора данных и объектом измерений находится в пределах от нескольких метров до сотен метров. В случаях непосредственной близости датчика и системы сбора данных (менее нескольких метров) в качестве сигнала измерительной информации используют напряжение. Частота, как информативный параметр измерительного сигнала, пока еще не нашла широкого применения несмотря на известные преимущества из-за ограничений помехозащищенности и быстродействия, присущих используемым сегодня широкополосным частотным демодуляторам.

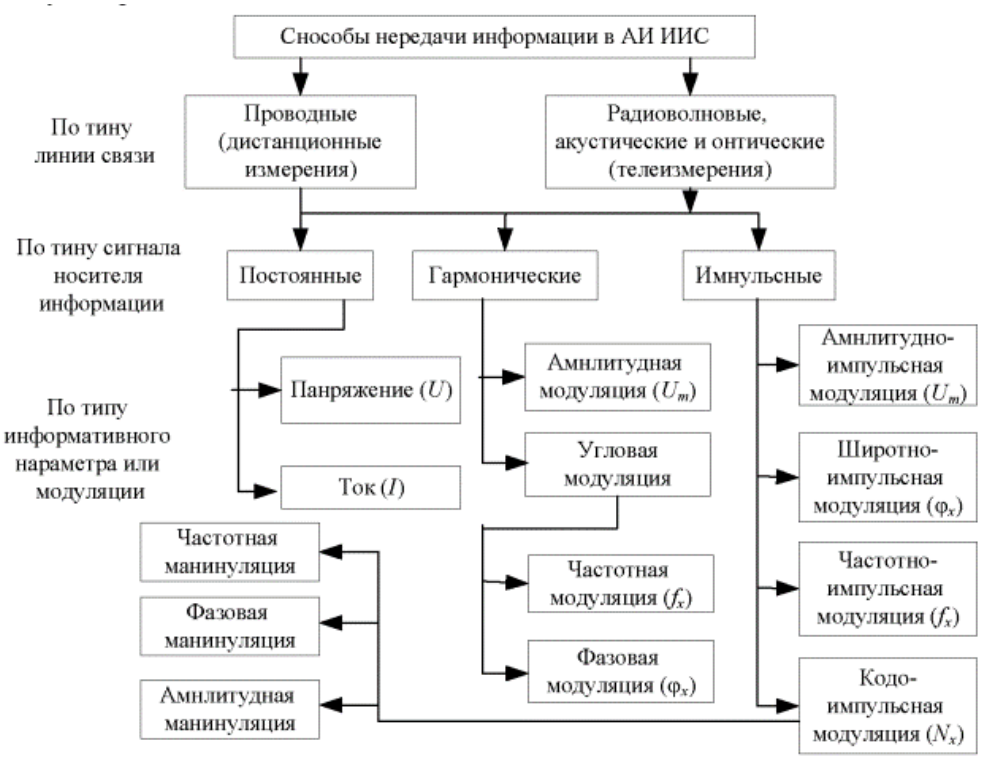


Рис. 3

Классификация АИ по типу средств связи. По этому признаку различают следующие АИ:

АИ с проводными линиями связи (используются при дистанционных измерениях) подразделяются на:

двухпроводные, в которых по линиям связи передаются как измерительные, так и сигналы инициализации резидентной части АИ; наиболее распространен токовый унифицированный сигнал в диапазоне 4-20 мА;

трехпроводные, в которых ток питания резидентной части и выходной ток датчика передаются в линиях связи по отдельным проводам;

четырехпроводные, в которых контуры питания и выходного сигнала изолированы друг от друга;

оптические;

радиоволновые (используют электромагнитные колебания f>30 кГц);

акустические.

Последние три способа используются с одним из методов модуляции и применяются в телеизмерительных системах.

Наиболее распространенными в ИИС являются АИ с проводными линиями связи. Оптические АИ более помехозащищенные, но дороже. Радиоволны в АИ используют, как правило, совместно с кодоимпульсной модуляцией. АИ на акустических волнах применяются в специальных условиях, например на подводных лодках.

Реализация двухпроводных и трехпроводных АИ существенно упрощается благодаря тому, что ряд универсальных приемопередающих измерительных преобразователей для унифицированных измерительных сигналов постоянного тока выпускаются в виде интегральных микросхем.

АИ с четырехпроводными линиями связи используются, как правило, совместно с датчиками общего применения и потенциальными выходными сигналами (0-5 В, 0-10 В).

Мультиплексирование

Мультиплексирование заключается в поочередном считывании всех сигналов в ИИС путем их поочередного подключения. Применяется для повышения эффективности использования цифровых процессоров и других компонент ИИС, сокращения времени их простаивания и уменьшения аппаратных затрат и стоимости ИИС. Реализуется введением в структуру АИ мультиплексоров или коммутаторов. Многоканальные ИИС разрабатывают, как правило, на 10... 100 каналов.

Мультиплексоры строятся на основе электронных ключей. В простейшем случае мультиплексор содержит *к* ключей, подключающих один из к входов системы к одному выходу. Для предотвращения короткого

замыкания между двумя входами эти ключи действуют по принципу переключения: сначала размыкается одна цепь, потом замыкается другая. В простом мультиплексоре переключение происходит только тогда, когда он получает команду от системного блока синхронизации. Такой мультиплексор называют последовательным: в нем *к* каналов подключаются к выходу в порядке, в котором они присоединены ко входам. В мультиплексорах с произвольным доступом центральный процессор определяет, какой из каналов должен быть подключен, указывая его адрес. С помощью таких мультиплексоров реализуют адаптивное мультиплексирование, что позволяет обращаться к узкополосным входным сигналам реже и получать выборки сигнала с меньшей частотой, чем для широкополосных сигналов. Этот метод позволяет экономить ресурсы ИИС, если спектры входных сигналов существенно различны.

Контрольные вопросы по теме

Уровень модуля

Уровень курса

1. Стандартный сигнал от датчика.
2. Назначение и функции аналогового интерфейса.
3. Классификация аналоговых интерфейсов.
4. Мультиплексирование сигналов от датчиков.

Лекция № 11

Тема: Исполнительные устройства

Оглавление

Источники:

1. Таненбаум Э., Остин Т. Архитектура компьютера. 6-е изд. — СПб.: Питер, 2013. — 816 с.: ил.

   https://rutracker.org/forum/viewtopic.php?t=4956359

2. Бабак В.П. Теоретические основы информационно-измерительных систем: Учебник / В.П. Бабак, С.В. Бабак, В.С. Ерёменко и др. – К., 2014. – 832 с.

Исполнительные устройства и механизмы

Исполнительные устройства — это устройства, воздействующие на физические процессы в соответствии с поступающим командными сигналами. Также, как и датчики, исполнительные устройства подбираются для каждой конкретной задачи с учетом необходимой выходной мощности и быстродействия.

Исполнительные механизмы — это устройства, которые механически воздействуют на объект контроля путем преобразования электрических сигналов в требуемое управляющее механическое воздействие.

Существуют устройства, которые также оказывают физическое воздействие на объект, но это воздействие не является механическим. К таким воздействиям относятся тепловое воздействие, различные виды облучений: оптическое, радиоактивное, акустическое и другие. Эти воздействия на объект оказывают путем применения соответствующих элементов. Например, нагрев производят нагревательные элементы, акустическое воздействие выполняют пьезоэлементы, а световое облучение получаем за счет включения осветителей. Весь класс устройств, оказывающих воздействие на объект путем исполнения команд от компьютерно-интегрированных систем, называют исполнительными элементами, исполнительными органами или исполнительными устройствами. Отдельным подклассом этих устройств являются исполнительные механизмы.

Классификация исполнительных механизмов

Классификация исполнительных механизмов (рис.1.1) производится в первую очередь по виду энергии, создающей усилие (момент) перемещения регулирующего органа. Соответственно, ИМ бывают пневматические, гидравлические и электрические.

В **пневматических** ИМ усилие перемещения создается за счет давления сжатого воздуха на мембрану, поршень или сильфон.

В **гидравлических** ИМ усилие перемещения создается за счет давления жидкости на мембрану, поршень или лопасть; давление жидкости в них обычно находится в пределах (2,5-20)103 кПа.

Пневматические и гидравлические мембранные и поршневые ИМ подразделяются на *пружинные* и *беспружинные*. В пружинных ИМ усилие перемещения в одном направлении создается давлением в рабочей полости ИМ, а в обратном направлении – силой упругости сжатой пружины. В беспружинных ИМ усилие перемещения в обоих направлениях создается перепадом давления на рабочем органе механизма.

Электрические ИМ по принципу действия подразделяются на электродвигательные (электромашинные) и электромагнитные.

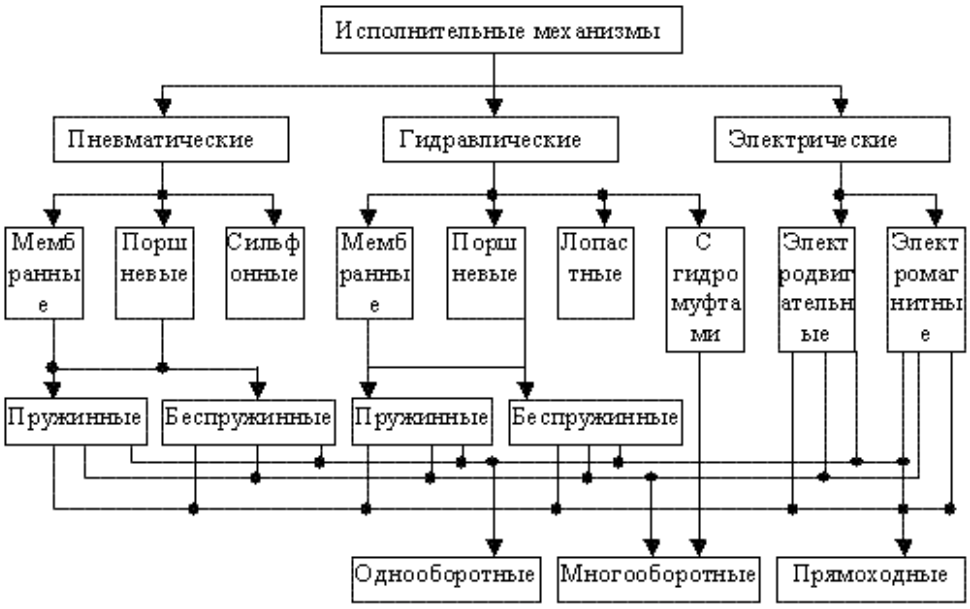


Рисунок11.1 Классификация исполнительных механизмов

По характеру движения выходного элемента большинство ИМ подразделяются на: *прямоходные* с поступательным движением выходного элемента, *поворотные* с вращательным движением до 360° (*однооборотные*) и *с вращательным движением* на угол более 360° (*многооборотные*).

Существуют ИМ, в которых используются одновременно два вида энергии: электропневматические, электрогидравлические и пневмогидравлические. Вид энергии управляющего сигнала может отличаться от вида энергии, создающей усилие перемещения.

В электрических системах автоматизации и управления наиболее широко применяются электромашинные и электромагнитные исполнительные механизмы.

Множество регулирующих органов также многообразно, как многообразны объекты управления. В качестве примера можно привести основные типы РО, применяемых в системах подачи и перемещения жидких, газообразных и сыпучих материалов. По виду воздействия на объект их можно подразделить на два основных типа: *дросселирующие* и *дозирующие*.

Дросселирующие РО изменяют сопротивление (гидравлическое, аэродинамическое) в системе путем изменения своего проходного сечения, воздействуя на расход вещества. Примерами таких РО являются заслонки, диафрагмы, задвижки, краны, клапаны.

Дозирующие РО выполняют заданное дозирование поступающего вещества или энергии за счет изменения производительности определенных агрегатов: дозаторов, насосов, компрессоров, питателей, электрических усилителей мощности.

Более полная классификация исполнительных механизмов может быть представлена следующим образом:

Классификационный признак	**Краткие характеристики классификационного признака**
1. По назначению работы	- Запорные (отсечные, предохранительные) - Регулирующие (задвижки) - Запорно-регулирующие клапаны (КЗР)
2. По типу рабочего регулирующего органа	- Седельные (односедельные, двухседельные, трехходовые) - Заслоночные (с диафрагменной заслонкой) - Клеточные - Шаровые - Пробковые - Дисковые поворотные затворы и заслонки - Пилотные
3. По виду перемещения регулирующего органа	- Однооборотные (МЭО рычажные), неполноповоротные (0,25 - 0,63 об.) фланцевые (МЭОФ) - Многооборотные регулирующие задвижки с постоянной скоростью (МЭМ) - Прямоходные поступательного действия для прямолинейного перемещения регулирующих органов с постоянной скоростью (МЭП, МЭПК, КЗР, краны)
4. По управлению направлениями потоков	- Прямые - Угловые
5. По типу управления потоками	- Двухходовые (запорные) - Трехходовые (распределительные, смесительные) - Четырехходовые (распределительные, смесительные)
6. По типу управляющего сигнала	- Пневматические - Электрические, электропневматические - Гидравлические
7. По типу привода	- Механический - Электромеханический - Электропневматический - Пневматический (мембранный, поршневой, лопастный) - Пневмомеханический - Пневмогидравлический - Гидравлический
8. По исполнению вида исполнительного устройства	- НО - нормально открытые - НЗ - нормально закрытые - произвольного положения
9. По пропускной характеристике Kv (сигнал - положение ИМ)	- Линейные. Обеспечивается пропорциональная зависимость между пропускной способностью клапана и ходом плунжера (затвора)

	- Равнопроцентные (экспоненциальные, логарифмические). Обеспечивается приращение пропускной способности клапана пропорционально текущему значению пропускной способности клапана, т.е. чем больше ход клапана, тем больше увеличивается Kv на единицу хода
10. По виду управляющего сигнала	- Аналоговые - Дискретные 2-х позиционные (статические и динамические) - Дискретные 3-х позиционные (статические и динамические)
11. По оснащению дополнительным оборудованием	- МПУ (HART, интерфейс), команда (управление, конфигурация) - состояние (оборудования) - Позиционер (пропорциональный ИМ) - Пилотный механизм - Ручной дублер (для ручного управления регулирующим органом) - Указатель положения регулирующего органа индуктивный, реостатный, токовый - Датчики крайних положений, блоки концевых выключателей

Классификация электрических исполнительных механизмов

Электрическим исполнительным механизмом в системах управления обычно называют устройство, предназначенное для перемещения рабочего органа в соответствии с сигналами, поступающими от управляющего устройства.

Рабочими органами могут быть различного рода дроссельные заслонки, клапаны, задвижки, шиберы, направляющие аппараты и другие регулирующие и запорные органы, способные производить изменение количества энергии или рабочего вещества, поступающего в объект управления. При этом перемещение рабочих органов может быть как поступательным, так и вращательным в пределах одного или нескольких оборотов. Следовательно, исполнительный механизм с помощью рабочего органа осуществляет непосредственное воздействие на управляемый объект.

В общем случае электрический исполнительный механизм состоит из электропривода, редуктора, узла обратной связи, датчика указателя положения выходного элемента и конечных выключателей.

В качестве электропривода в исполнительных механизмах используются либо электромагниты, либо электродвигатели с понижающим редуктором для снижения скорости перемещения выходного элемента до величины, обеспечивающей возможность непосредственного соединения этого элемента (вала или штока) с рабочим органом.

Узлы обратной связи предназначены для введения в контур регулирования воздействия, пропорционального величине перемещения выходного элемента исполнительного механизма, а, следовательно, и сочлененного с ним рабочего органа. С помощью конечных выключателей производится отключение электропривода исполнительного механизма при достижении рабочим органом своих конечных положений во избежание возможных повреждений механических звеньев, а также для ограничения перемещения рабочего органа.

Рисунок 11.2 Классификация электрических исполнительных механизмов

Как правило, мощность сигнала, вырабатываемого регулирующим устройством, бывает недостаточной для непосредственного перемещения рабочего органа, поэтому исполнительный механизм можно рассматривать как усилитель мощности, в котором слабый входной сигнал, усиливаясь во много раз, передается на рабочий орган.

Все электрические исполнительные механизмы, нашедшие широкое применение в самых различных отраслях современной техники автоматизации производственных процессов, можно разделить на две основные группы:

1) электромагнитные

2) электродвигательные.

К первой группе относятся прежде всего ***соленоидные электроприводы***, предназначенные для управления различного рода регулирующими и запорными клапанами, вентилями, золотниками и т. п. Сюда же можно отнести исполнительные механизмы с различными видами электромагнитных муфт. Характерная особенность электрических исполнительных механизмов этой группы состоит в том, что необходимое для перестановки рабочего органа усилие создается за счет электромагнита, являющегося неотъемлемой частью исполнительного механизма.

Для целей регулирования соленоидные механизмы обычно применяются только в системах двухпозиционного регулирования. В системах автоматического управления в качестве исполнительных элементов часто используются электромагнитные муфты, которые подразделяются на муфты трения и муфты скольжения.

Ко *второй,* наиболее распространенной в настоящее время группе относятся электрические исполнительные механизмы ***с электродвигателями*** различных типов и конструкций.

Электродвигательные исполнительные механизмы обычно состоят из двигателя, редуктора и тормоза (последнего иногда может и не быть). Сигнал управления поступает одновременно к двигателю и тормозу, механизм растормаживается и двигатель приводит в движение выходной орган. При исчезновении сигнала двигатель выключается, а тормоз останавливает механизм. Простота схемы, малое число элементов, участвующих в формировании регулирующего воздействия, и высокие эксплуатационные свойства сделали исполнительные механизмы с управляемыми двигателями

основой для создания исполнительных устройств современных промышленных систем автоматического регулирования.

Существуют, хотя и не получили широкого распространения, исполнительные механизмы с неуправляемыми двигателями, которые содержат управляемую электрическим сигналом механическую, электрическую либо гидравлическую муфту. Характерной их особенностью является то, что двигатель в них работает непрерывно все время работы системы регулирования, а сигнал управления от регулирующего прибора передается рабочему органу через управляемую муфту

Исполнительные механизмы с управляемыми двигателями в свою очередь можно разделить по способу построения системы управления на механизмы с контактным и бесконтактным управлением.

Включение, отключение и реверсирование электродвигателей исполнительных механизмов **с контактным управлением** производится с помощью различной релейной или контактной аппаратуры. Это определяет основную отличительную особенность исполнительных механизмов с контактным управлением: у таких механизмов скорость выходного органа не зависит от величины управляющего сигнала, подаваемого на вход исполнительного устройства, а направление перемещения определяется знаком (или фазой) этого сигнала. Поэтому исполнительные механизмы с контактным управлением относят обычно к исполнительным устройствам с постоянной скоростью перемещения рабочего органа.

Для получения средней переменной скорости перемещения выходного органа исполнительного механизма при контактном управлении широко используется импульсный режим работы его электродвигателя.

В большинстве исполнительных механизмов, предназначенных для работы в схемах с контактным управлением, используются реверсивные электродвигатели. Применение электродвигателей, вращающихся только в одну сторону, весьма ограничено, но все же имеет место.

Бесконтактные электрические исполнительные механизмы отличаются повышенной надежностью и позволяющие относительно просто получать как постоянную, так и переменную скорость перемещения выходного органа. Для бесконтактного управления исполнительными механизмами используются электронные, магнитные или полупроводниковые усилители, а также их сочетание. При работе управляющих усилителей в релейном режиме скорость перемещения выходного органа исполнительных механизмов постоянна.

Как электрические исполнительные механизмы с контактным управлением, так и бесконтактные можно подразделять также по следующим признакам.

- По характеру движения: однооборотные, многооборотпые, прямоходные.
- По характеру действия: позиционного действия; пропорционального действия.
- По исполнению: в нормальном исполнении, в специальном исполнении (пылеводозащищенном, взрывозащищением, тропическом, морском и т. п.).

Выходной вал **однооборотных** исполнительных механизмов может вращаться в пределах одного полного оборота. Такие механизмы характеризуются величиной крутящего момента на выходном валу и временем его полного оборота.

В отличие от однооборотных **многооборотных механизмов**, выходной вал которых может осуществлять перемещение в пределах нескольких, иногда значительного количества, оборотов, характеризуются также полным числом оборотов выходного вала.

Прямоходные механизмы имеют поступательное движение выходного штока и оцениваются усилием на штоке, величиной полного хода штока, временем его перемещения на участке полного хода и по скорости движения выходного органа в оборотах в минуту для однооборотных и многооборотных и в миллиметрах в секунду для прямоходных механизмов.

Конструкция исполнительных механизмов **позиционного** действия такова, что с их помощью рабочие органы можно устанавливать только в определенные фиксированные положения. Чаще всего таких положений бывает два: «открыто» и «закрыто». В общем случае возможно существование и многопозиционных механизмов. Исполнительные механизмы позиционного действия обычно не имеют устройств для получения сигнала обратной связи по положению выходного органа.

Исполнительные механизмы **пропорционального** действия конструктивно таковы, что обеспечивают в заданных пределах установку рабочего органа в любое промежуточное положение в зависимости от величины и длительности управляющего сигнала. Подобные исполнительные механизмы широко используются в системах автоматического регулирования с пропорциональным, пропорционально-интегральным и пропорционально-интегрально-дифференциальным законом управления.

Виды управляющих сигналов

Простейшим управляющим сигналом является релейный сигнал, которой имеет самые разнообразные названия: "**открыт-закрыт**", "вкл-выкл", "on-off", "0-1", двухточечный, двухпозиционный сигнал и т.д. Этот сигнал применяется для управления пружинными механизмами, которые могут находится только в двух состояниях, например, "открыто" (1) или "закрыто" (0). Для того чтобы установить сигнал в положение "открыто" (1) необходимо подать заданное напряжение (это может быть 5В или 10В, или 30В или 220В – зависит от самого устройства, которым управляют) достаточной мощности. Если напряжение снять (снять сигнал), то пружина вернет исполнительный механизм в закрытое состояние (0).

Другим видом сигнала является так называемый **трехточечный** сигнал (иначе: трехпозиционный, плавающий сигнал и др.), который тоже является релейным, так как имеет только два уровня напряжения: 0 и высокое (это может быть 5В или 10В, или 30В или 220В – зависит от самого устройства, которым управляют). Двигатель 3-точечного исполнительного механизма вращается в оба направления (по часовой и против часовой стрелки) под воздействием электрических сигналов, определяющих открытие или закрытие. Как только выходное напряжение реле пропадает, исполнительный механизм останавливает свою работу и сохраняет позицию на момент остановки. Стабильность в данной позиции обеспечивается самоблокирующимся устройством.

Пропорциональный сигнал – сигнал переменной величины, который несет в себе информацию о требуемом угле поворота ИМ или скорости его вращения. Сигнал обычно имеет стандартную форму (см. предыдущую лекцию), например, 0-10В или 4-20мА. Исполнительный механизм переходит в позицию, заданную управляющим сигналом или же начинает вращаться с заданной скоростью.

Сигналы **ШИМ** – сигналы с широтно-импульсной модуляцией. Широтно-импульсная модуляция – процесс управления мощностью, подводимой к нагрузке, путём изменения скважности импульсов, при постоянной частоте.

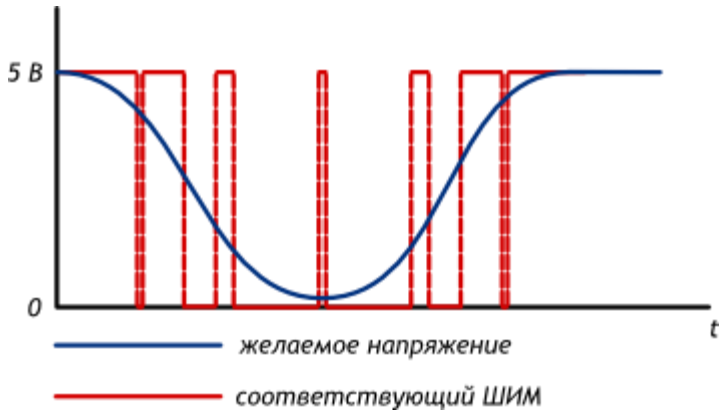


Рисунок 11.3 Широтно-импульсная модуляция.

При помощи ШИМ легко управлять количеством тепла, отдаваемого нагревателем. ШИМ часто используется для управления скоростью вращения вала двигателя. В этом случае частоту вращения двигателя регулируют не величиной постоянно подводимого напряжения, а длительностью питания двигателя номинальным напряжением. Одна из возможных схем импульсного управления приведена на рис. 11.4, *а*. Там же (рис. 11.4, *б*) показаны графики скорости при различных *t*.

В период, когда электронный ключ открыт, питающее напряжение полностью подается на двигатель, ток якоря увеличивается, двигатель развивает положительный момент и частота вращения возрастает; когда электронный ключ закрыт, ток под действием запаса электромагнитной энергии продолжает протекать в том же направлении, но через обратный диод. При этом ток уменьшается, момент двигателя уменьшается, угловая скорость вращения падает.

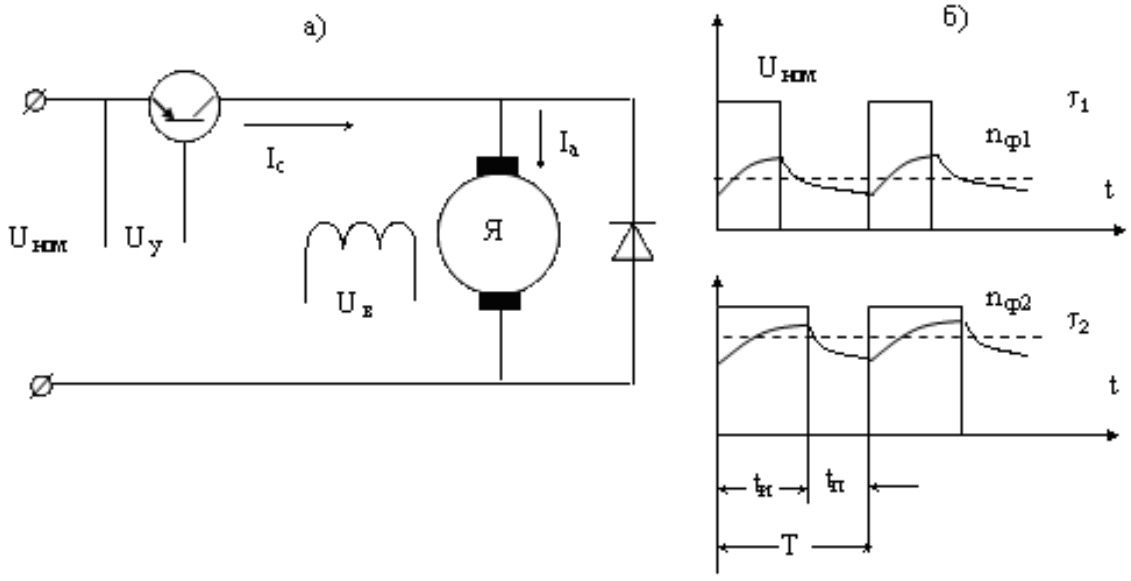


Рис. 11.4. Схема импульсного управления (а), графики скорости вращения (б) при разных τ ($\tau_2 > \tau_1$).

Работа двигателя состоит из чередующихся периодов разгона и торможения. И, если эти периоды малы по сравнению с электромагнитной постоянной времени якорной цепи, устанавливается некая средняя скорость, однозначно определяемая относительной продолжительностью включения (скважностью) $t = t_и/T$, где $t_и$ - длительность импульса напряжения; Т - период.

Контрольные вопросы по теме

Уровень модуля

Уровень курса

1. Исполнительные устройства и механизмы в компьютерно-интегрированных системах.
2. Классификация исполнительных механизмов.
3. Классификация электрических исполнительных механизмов.
4. Виды управляющих сигналов.

Лекции № 12 Операционные системы. (Часть первая)

Оглавление

Источники:

1. Таненбаум Э., Остин Т. Архитектура компьютера. 6-е изд. — СПб.: Питер, 2013. — 816 с.: ил.

   https://rutracker.org/forum/viewtopic.php?t=4956359

2. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. — СПб.: Питер, 2015. — 1120 с.: ил.
3. Бабак В.П. Теоретические основы информационно-измерительных систем: Учебник / В.П. Бабак, С.В. Бабак, В.С. Ерёменко и др. – К., 2014. – 832 с.

Структура программного обеспечения

Совокупность программ, предназначенная для решения задач на компьютере, называется программным обеспечением. Программное обеспечение (ПО), можно условно разделить на три категории:

1. **Системное** (программы общего пользования), выполняющие различные вспомогательные функции, например создание копий используемой информации, выдачу справочной информации о компьютере, проверку работоспособности устройств компьютера и т.д.

2. **Прикладное**, обеспечивающее выполнение задач, необходимых пользователю: обработка информационных массивов, математическое моделирование, редактирование текстовых документов, создание и редактирование изображений, и т.д.

3. **Инструментальное** (системы программирования), обеспечивающее разработку новых программ для компьютера на языке программирования.

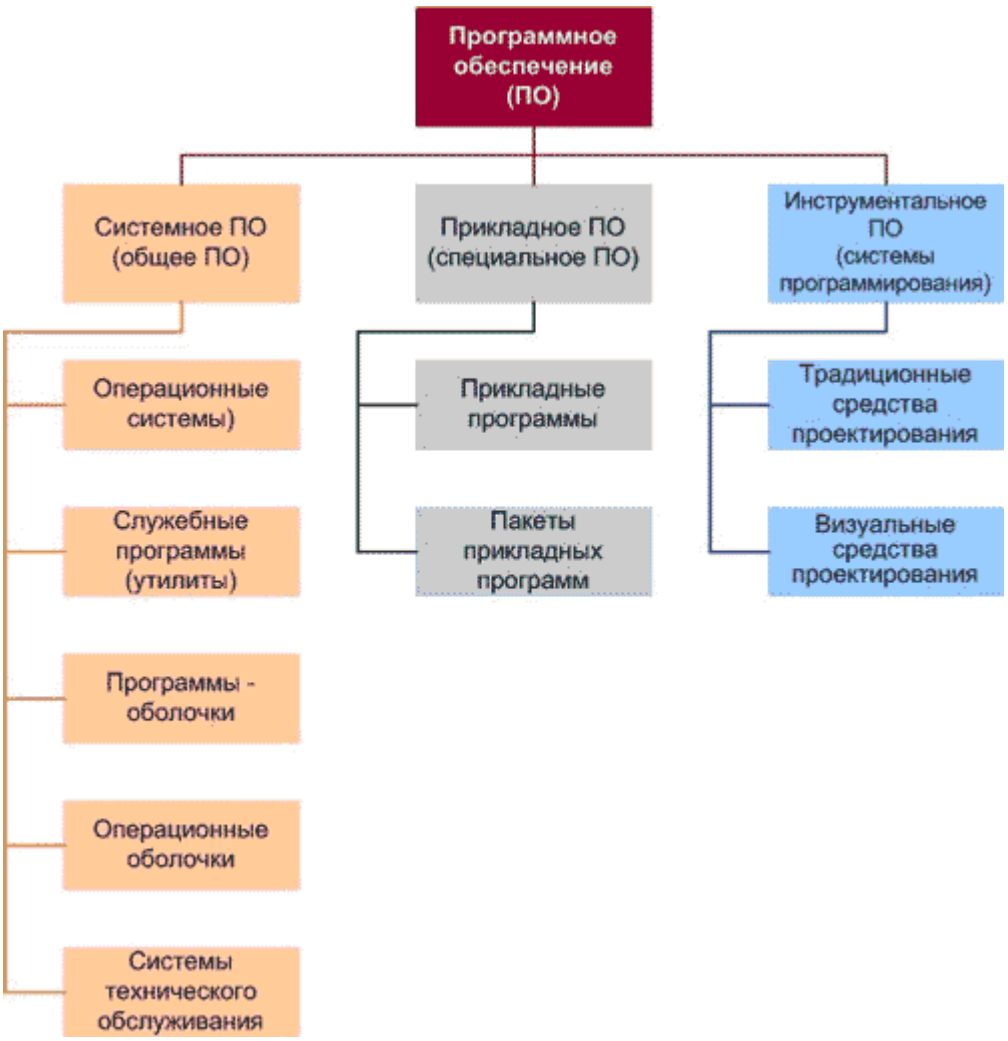


Рис. 1. Классификация программного обеспечения

Системное ПО

Системное программное обеспечение — комплекс программ, которые обеспечивают управление компонентами компьютерной системы, такими как процессор, оперативная память, устройства ввода-вывода, сетевое оборудование, выступая как «межслойный интерфейс», с одной стороны которого аппаратура, а с другой — приложения пользователя. В отличие от прикладного программного обеспечения, системное не решает конкретные практические задачи, а лишь обеспечивает работу других программ, предоставляя им сервисные функции, абстрагирующие детали аппаратной и микропрограммной реализации вычислительной системы, управляет аппаратными ресурсами вычислительной системы.

К системному ПО относятся:

- операционные системы;
- драйверы (программы, предназначенные для управления портами периферийных устройств);
- утилиты.

Утилита – вспомогательная компьютерная программа в составе общего программного обеспечения для выполнения специализированных типовых задач, связанных с работой оборудования и операционной системы. Утилиты предоставляют доступ к возможностям (параметрам, настройкам, установкам), недоступным без их применения, либо делают процесс изменения некоторых параметров проще (автоматизируют его).

Утилиты могут входить в состав операционных систем, идти в комплекте со специализированным оборудованием или распространяться отдельно. К утилитам, в частности, относятся:

- утилиты управления процессами
- диспетчеры файлов или файловые менеджеры;
- архиваторы;
- средства динамического сжатия данных;
- средства просмотра и воспроизведения;
- средства контроля и диагностики – позволяют проверить конфигурацию компьютера и проверить работоспособность устройств компьютера, прежде всего жестких дисков;
- утилиты восстановления после сбоев;
- Оптимизатор диска – вид утилиты для оптимизации размещения файлов на дисковом накопителе, например, путём дефрагментации диска;
- деинсталлятор;

- средства обеспечения компьютерной безопасности (резервное копирование, антивирусное ПО).

Необходимо отметить, что часть утилит входит в состав операционной системы, а другая часть функционирует автономно. Большая часть общего (системного) ПО входит в состав ОС. Часть общего ПО входит в состав самого компьютера – часть программ ОС и контролирующих тестов записана в постоянном запоминающем устройстве системной платы. Часть общего ПО относится к автономными программам и поставляется отдельно.

Прикладное ПО

Прикладные программы могут использоваться либо автономно, либо в составе программных комплексов и пакетов.

Пакеты прикладных программ – это система программ, которые по сфере применения делятся на проблемно – ориентированные, пакеты общего назначения и интегрированные пакеты. К прикладному ПО, например, относятся:

- комплект офисных приложений MS OFFICE,
- бухгалтерские системы и системы управления предприятиями,
- CAD – системы (системы автоматизированного проектирования),
- браузеры – средства просмотра Web - страниц,
- графические редакторы,
- редакторы HTML или Web – редакторы,
- системы выполнения математических расчетов и моделирования,
- экспертные системы.

Инструментальное ПО

Инструментальное ПО или системы программирования – это системы для автоматизации разработки новых программ на языках программирования.

В самом общем случае для создания программы на выбранном языке нужно иметь следующие компоненты:

1. Текстовый редактор для создания файла с исходным текстом программы.
2. Компилятор или интерпретатор. Исходный текст с помощью программы-компилятора переводится в промежуточный объектный код. Исходный текст большой программы состоит из нескольких *модулей* (файлов с исходными текстами). Каждый модуль компилируется в отдельный файл с объектным кодом, которые затем надо объединить в одно целое.
3. Редактор связей или сборщик, который выполняет связывание объектных модулей и формирует на выходе работоспособное

приложение – исполнимый код. Исполнимый код – это законченная программа, которую можно запустить на любом компьютере, где установлена операционная система, для которой эта программа создавалась. Как правило, итоговый файл имеет расширение .EXE или .COM.

Операционная система

Современный компьютер представляет собой сложную систему. Компьютер состоит из одного или нескольких процессоров, оперативной памяти, дисков, принтера, клавиатуры, мыши, дисплея, сетевых интерфейсов и других разнообразных устройств ввода-вывода. Если каждому программисту, создающему прикладную программу, нужно будет разбираться во всех тонкостях работы всех этих устройств, то он не напишет ни строчки кода. Более того, управление всеми этими компонентами и их оптимальное использование представляет собой очень непростую задачу. По этой причине компьютеры оснащены специальным уровнем программного обеспечения, который называется операционной системой, в чью задачу входит управление пользовательскими программами, а также всеми упомянутыми ресурсами.

Программы, с которыми непосредственно взаимодействуют пользователи, обычно называемые оболочкой, когда они основаны на применении текста, и графическим пользовательским интерфейсом (Graphical User Interface (GUI)), когда в них используются значки, фактически не являются частью операционной системы, хотя задействуют эту систему в своей работе. Схематично основные рассматриваемые здесь компоненты представлены на рис. 2.

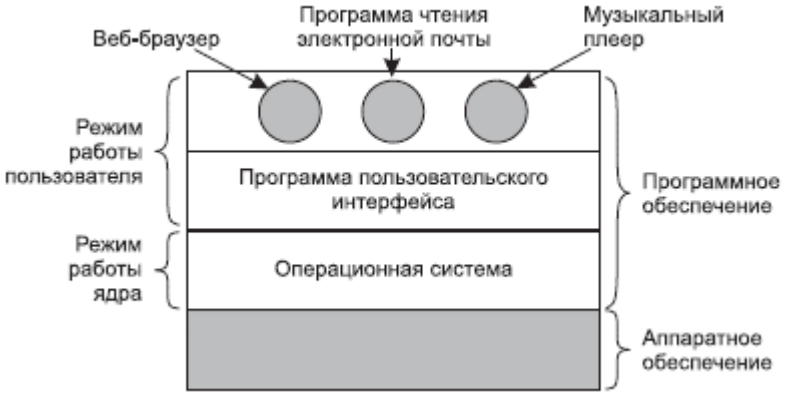


Рис. 2. Место операционной системы в структуре программного обеспечения

В нижней части рисунка показано аппаратное обеспечение. Оно состоит из микросхем, плат, дисков, клавиатуры, монитора и других физических объектов. Над аппаратным обеспечением находится программное обеспечение. Большинство компьютеров имеют два режима работы: режим

ядра и режим пользователя. Операционная система — наиболее фундаментальная часть программного обеспечения, работающая в режиме ядра (этот режим называют еще режимом супервизора). В этом режиме она имеет полный доступ ко всему аппаратному обеспечению и может задействовать любую инструкцию, которую машина в состоянии выполнить. Вся остальная часть программного обеспечения работает в режиме пользователя, в котором доступно лишь подмножество инструкций машины. В частности, программам, работающим в режиме пользователя, запрещено использование инструкций, управляющих машиной или осуществляющих операции ввода-вывода (Input/Output — I/O).

Программы пользовательского интерфейса — оболочка или GUI — находятся на самом низком уровне программного обеспечения, работающего в режиме пользователя, и позволяют пользователю запускать другие программы, такие как веб-браузер, программа чтения электронной почты или музыкальный плеер. Эти программы также активно пользуются операционной системой.

Местонахождение операционной системы показано на рис. 2. Она работает непосредственно с аппаратным обеспечением и является основой остального программного обеспечения. Важное отличие операционной системы от обычного (работающего в режиме пользователя) программного обеспечения состоит в следующем: если пользователь недоволен конкретной программой чтения электронной почты, то он может выбрать другую программу или, если захочет, написать собственную программу, но не может написать собственный обработчик прерываний системных часов, являющийся частью операционной системы и защищенный на аппаратном уровне от любых попыток внесения изменений со стороны пользователя.

Во многих системах также есть программы, работающие в режиме пользователя, но помогающие работе операционной системы или выполняющие особые функции. К примеру, довольно часто встречаются программы, позволяющие пользователям изменять их пароли. Они не являются частью операционной системы и не работают в режиме ядра, но они выполняют важную функцию и должны быть особым образом защищены. В некоторых системах эта идея доведена до крайней формы, и те области, которые традиционно относились к операционной системе (например, файловая система), работают в пространстве пользователя. В таких системах трудно провести четкую границу. Все программы, работающие в режиме ядра, безусловно, являются частью операционной системы, но некоторые программы, работающие вне этого режима, возможно, также являются ее частью или, по крайней мере, имеют с ней тесную связь.

Операционные системы отличаются от пользовательских программ (то есть приложений) не только местоположением. Их особенности – довольно большой объем, сложная структура и длительные сроки использования. Исходный код основы операционной системы типа Linux или Windows занимает порядка 5 млн строк. И это касается только той части, которая работает в режиме ядра. При включении необходимых общих библиотек объем Windows превышает 70 млн строк кода, не считая основных прикладных программ (таких, как Windows Explorer, Windows Media Player и т. д.). Поэтому операционные системы живут достаточно долго, – их очень трудно создавать. После создания одной такой системы нет смысла ее выбрасывать и приступать к созданию новой. Поэтому операционные системы развиваются в течение долгого периода времени. Семейство Windows 95/98/Me по своей сути представляло одну операционную систему, а семейство Windows NT/2000/XP/Vista/Windows 7 – другую. Для пользователя они были похожи друг на друга, поскольку Microsoft позаботилась о том, чтобы пользовательский интерфейс Windows 2000/XP/ Vista/Windows 7 был очень похож на ту систему, которой он шел на замену, а чаще всего это была Windows 98. Другим примером является операционная система UNIX, ее варианты и клоны. Она также развивалась в течение многих лет, существуя в таких базирующихся на исходной системе версиях, как System V, Solaris и FreeBSD. А вот Linux имеет новую программную основу, хотя ее модель весьма близка к UNIX и она обладает высокой степенью совместимости с этой системой.

Контрольные вопросы по теме

Уровень модуля

Уровень курса

1. Структура программного обеспечения компьютерно-интегрированных систем.
2. Системное программное обеспечение, его назначение и состав.
3. Прикладное и инструментальное программное обеспечение.
4. Операционная система, ее назначение и состав.

Лекции № 13 ***Тема:*** Операционные системы.

Оглавление

Источники:

1. Таненбаум Э., Остин Т. Архитектура компьютера. 6-е изд. — СПб.: Питер, 2013. — 816 с.: ил.

https://rutracker.org/forum/viewtopic.php?t=4956359

2. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. — СПб.: Питер, 2015. — 1120 с.: ил.
3. Бабак В.П. Теоретические основы информационно-измерительных систем: Учебник / В.П. Бабак, С.В. Бабак, В.С. Ерёменко и др. – К., 2014. – 832 с.

Функции операционной системы

Дать точное определение операционной системы довольно трудно. Можно сказать, что это программное обеспечение, которое работает в режиме ядра. Операционные системы осуществляют две значительно отличающиеся друг от друга функции: предоставляют прикладным программистам (и прикладным программам, естественно) вполне понятный абстрактный набор ресурсов взамен неупорядоченного набора аппаратного обеспечения, а также управляют этими ресурсами. Рассмотрим эти функции.

Абстрактные ресурсы

Архитектура большинства компьютеров (система команд, организация памяти, ввод-вывод данных и структура шин) на уровне машинного языка слишком примитивна и неудобна для использования в программах, особенно это касается систем ввода-вывода. Поэтому оборудованием, например жестким диском, занимается не пользовательская программа (для чего в каждую пользовательскую программу необходимо было бы вводить специальные сложные блоки программных кодов), а та часть системного программного обеспечения, которая называется драйвером диска и предоставляет, не вдаваясь в детали, интерфейс для чтения и записи дисковых блоков. Операционные системы содержат множество драйверов для управления устройствами ввода-вывода.

Но для большинства приложений слишком низким является даже этот уровень. Поэтому все операционные системы предоставляют еще один уровень абстракции для использования дисков — файлы. Используя эту абстракцию, программы могут создавать, записывать и читать файлы, не вникая в подробности реальной работы оборудования.

Эта абстракция является ключом к управлению сложностью. Хорошая абстракция превращает практически неподъемную задачу в две, решить которые вполне по силам. Первая из этих задач состоит в определении и реализации абстракций, а вторая — в использовании этих абстракций для решения текущей проблемы. Одна из абстракций, понятная практически любому пользователю компьютера, — это уже упомянутый ранее файл. Он представляет собой полезный объем информации, скажем, цифровую фотографию, сохраненное сообщение электронной почты или веб-страницу. Работать с фотографиями, сообщениями электронной почты и веб-страницами намного легче, чем с особенностями SATA-дисков (или других дисковых устройств). Задача операционной системы заключается в создании хорошей абстракции, а затем в реализации абстрактных объектов, создаваемых в рамках этой абстракции, и управлении ими.

Существующие процессоры, блоки памяти, диски и другие компоненты представляют собой слишком сложные устройства, предоставляющие трудные, неудобные, не похожие друг на друга и не обладающие постоянством интерфейсы. Одна из главных задач операционной системы – скрыть аппаратное обеспечение и существующие программы под создаваемыми взамен них и приспособленными для нормальной работы неизменными абстракциями. Следует отметить, что в действительности «заказчиками» операционных систем являются прикладные программы (разумеется, не без помощи прикладных программистов). Именно они непосредственно работают с операционной системой и ее абстракциями. А конечные пользователи работают с абстракциями, предоставленными пользовательским интерфейсом, – это или командная строка оболочки, или графический интерфейс. Абстракции пользовательского интерфейса могут быть похожими на абстракции, предоставляемые операционной системой, но так бывает не всегда. Чтобы пояснить это положение, рассмотрим обычный рабочий стол Windows и командную строку. И то и другое — программы, работающие под управлением операционной системы Windows и использующие предоставленные этой системой абстракции, но они предлагают существенно отличающиеся друг от друга пользовательские интерфейсы.

Операционная система в качестве менеджера ресурсов

Представление о том, что операционная система главным образом предоставляет абстракции для прикладных программ, — это взгляд сверху вниз. Сторонники альтернативного взгляда, снизу вверх, придерживаются того мнения, что операционная система существует для управления всеми частями сложной системы. Современные компьютеры состоят из процессоров, памяти, таймеров, дисков, манипуляторов, сетевых интерфейсов, принтеров и широкого спектра других устройств. Сторонники взгляда снизу вверх считают, что задача операционной системы заключается в обеспечении упорядоченного и управляемого распределения процессоров, памяти и устройств ввода-вывода между различными программами, претендующими на их использование.

Современные операционные системы допускают одновременную работу нескольких программ. Представьте себе, что будет, если все три программы, работающие на одном и том же компьютере, попытаются распечатать свои выходные данные одновременно на одном и том же принтере. Первые несколько строчек распечатки могут быть от программы № 1, следующие несколько строчек — от программы № 2, затем несколько строчек от программы № 3 и т. д. В результате получится полный хаос. Операционная система призвана навести порядок в потенциально возможном

хаосе за счет буферизации на диске всех выходных данных, предназначенных для принтера. После того как одна программа закончит свою работу, операционная система сможет скопировать ее выходные данные с файла на диске, где они были сохранены, на принтер, а в то же самое время другая программа может продолжить генерацию данных, не замечая того, что выходные данные фактически (до поры до времени) не попадают на принтер.

Когда с компьютером (или с сетью) работают несколько пользователей, потребности в управлении и защите памяти, устройств ввода-вывода и других ресурсов значительно возрастают, поскольку иначе пользователи будут мешать друг другу работать. Кроме этого, пользователям часто требуется совместно использовать не только аппаратное обеспечение, но и информацию (файлы, базы данных и т. п.). Первичной задачей операционной системы является отслеживание того, какой программой какой ресурс используется, чтобы удовлетворять запросы на использование ресурсов, нести ответственность за их использование и принимать решения по конфликтующим запросам от различных программ и пользователей.

Управление ресурсами включает в себя **мультиплексирование** (распределение) ресурсов двумя различными способами: во времени и в пространстве. Когда ресурс разделяется во времени, различные программы или пользователи используют его по очереди: сначала ресурс получают в пользование одни, потом другие и т. д. К примеру, располагая лишь одним центральным процессором и несколькими программами, стремящимися на нем выполняться, операционная система сначала выделяет центральный процессор одной программе, затем, после того как она уже достаточно поработала, центральный процессор получает в свое распоряжение другая программа, затем еще одна программа, и, наконец, его опять получает в свое распоряжение первая программа. Определение того, как именно ресурс будет разделяться во времени — кто будет следующим потребителем и как долго, — это задача операционной системы. Другим примером мультиплексирования во времени может послужить совместное использование принтера. Когда в очереди для распечатки на одном принтере находятся несколько заданий на печать, нужно принять решение, какое из них будет выполнено следующим.

Другим видом разделения ресурсов является пространственное разделение. Вместо поочередной работы каждый клиент получает какую-то часть разделяемого ресурса. Например, оперативная память обычно делится среди нескольких работающих программ, так что все они одновременно могут постоянно находиться в памяти (например, используя центральный процессор по очереди). При условии, что памяти достаточно для хранения более чем одной программы, эффективнее разместить в памяти сразу несколько программ, чем выделять всю память одной программе, особенно если ей

нужна лишь небольшая часть от общего пространства. Разумеется, при этом возникают проблемы равной доступности, обеспечения безопасности и т. д., и их должна решать операционная система. Другим ресурсом с разделяемым пространством является жесткий диск. На многих системах на одном и том же диске могут одновременно храниться файлы, принадлежащие многим пользователям. Распределение дискового пространства и отслеживание того, кто какие дисковые блоки использует, — это типичная задача операционной системы по управлению ресурсами.

Основные понятия и абстракции операционной системы

Большинство операционных систем используют определенные основные понятия и абстракции, такие как процессы, адресные пространства и файлы, которые играют главную роль в осмыслении самих систем.

Процессы

Ключевым понятием во всех операционных системах является процесс. Процессом, по существу, является программа во время ее выполнения. С каждым процессом связано его адресное пространство – список адресов ячеек памяти от нуля до некоторого максимума, откуда процесс может считывать данные и куда может записывать их. Адресное пространство содержит выполняемую программу, данные этой программы и ее стек. Кроме этого, с каждым процессом связан набор ресурсов, который обычно включает регистры(в том числе счетчик команд и указатель стека), список открытых файлов, необработанные предупреждения, список связанных процессов и всю остальную информацию, необходимую в процессе работы программы. Таким образом, процесс – это контейнер, в котором содержится вся информация, необходимая для работы программы.

Для того чтобы выработать интуитивное представление о процессе, рассмотрим систему, работающую в мультипрограммном режиме. Пользователь может запустить программу редактирования видео и указать конвертирование одночасового видеофайла в какой-нибудь определенный формат (процесс займет несколько часов), а затем переключиться на блуждания по Интернету. При этом может заработать фоновый процесс, который периодически «просыпается» для проверки входящей электронной почты. И у нас уже будет (как минимум) три активных процесса: видеоредактор, веб-браузер и программа получения (клиент) электронной почты. Периодически операционная система будет принимать решения остановить работу одного процесса и запустить выполнение другого, возможно, из-за того, что первый исчерпал свою долю процессорного временив предыдущую секунду или две.

Если процесс приостанавливается таким образом, позже он должен возобновиться именно с того состояния, в котором был остановлен. Это означает, что на период приостановки вся информация о процессе должна быть явным образом где-то сохранена. Например, у процесса могут быть одновременно открыты для чтения несколько файлов. С каждым из этих файлов связан указатель текущей позиции (то есть номер байта или записи, которая должна быть считана следующей). Когда процесс приостанавливается, все эти указатели должны быть сохранены, чтобы вызов read, выполняемый после возобновления процесса, приводил к чтению нужных данных. Во многих операционных системах вся информация о каждом процессе, за исключением содержимого его собственного адресного пространства, хранится в таблице операционной системы, которая называется **таблицей процессов** и представляет собой массив (или связанный список) структур, по одной на каждый из существующих на данный момент процессов.

Таким образом, процесс (в том числе приостановленный) состоит из собственного адресного пространства, которое обычно называют образом памяти, и записи в таблице процессов с содержимым его регистров, а также другой информацией, необходимой для последующего возобновления процесса.

Системный вызов – это обращение прикладной программы (процесса) к операционной системе для выполнения какой-либо операции.

Главными системными вызовами, используемыми при управлении процессами, являются вызовы, связанные с созданием и завершением процессов. Рассмотрим простой пример. Процесс, называемый интерпретатором команд, или оболочкой, считывает команды с терминала. Пользователь только что набрал команду, требующую компиляции программы. Теперь оболочка должна создать новый процесс, запускающий компилятор. Когда этот процесс завершит компиляцию, он произведет системный вызов для завершения собственного существования.

Если процесс способен создавать несколько других процессов (называющихся дочерними процессами), а эти процессы в свою очередь могут создавать собственные дочерние процессы, то перед нами предстает дерево процессов, подобное изображенному на рис. 3.

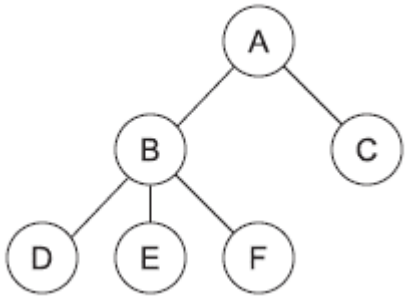


Рис. 1.3. Дерево процессов. Процесс A создал два дочерних процесса, B и C. Процесс B создал три дочерних процесса, D, E и F

Связанные процессы, совместно работающие над выполнением какой-нибудь задачи, зачастую нуждаются в обмене данными друг с другом и синхронизации своих действий. Такая связь называется межпроцессным взаимодействием. Другие системные вызовы, предназначенные для управления процессом, позволяют запросить выделение дополнительной памяти (или освобождение незадействованной), организовать ожидание завершения дочернего процесса или загрузку какой-нибудь другой программы поверх своей.

Временами возникает потребность в передаче информации запущенному процессу, который не находится в состоянии ожидания этой информации. Можно привести в пример процесс, который обменивается информацией с другим процессом, запущенным на другом компьютере, и посылает удаленному процессу сообщение по сети. Чтобы застраховаться от возможной утраты сообщения или ответа на него, отправитель может запросить собственную операционную систему уведомить его по истечении определенного интервала времени, чтобы он мог повторно отправить сообщение, если не получит подтверждения его получения раньше. После установки такого таймера программа может продолжить выполнение другой работы.

Когда истечет заданный интервал времени, операционная система посылает процессу сигнал тревоги. Этот сигнал заставляет процесс приостановить выполняемую работу, сохранить в стеке состояние своих регистров и запустить специальную процедуру обработки сигнала тревоги, для того чтобы, к примеру, заново передать предположительно утраченное сообщение. Когда обработчик сигнала завершит свою работу, запущенный процесс возобновится в том самом состоянии, которое было до поступления сигнала. Сигналы являются программными аналогами аппаратных прерываний. Они могут генерироваться в различных ситуациях, а не только по истечении времени, установленного в таймере. Многие аппаратные прерывания (например, выполнение недопустимой команды или обращение

по неверному адресу) также транслируются процессу, при выполнении которого произошла ошибка.

Напомним (лекция 4), что *прерывание* (англ. interrupt) – это сигнал от программного или аппаратного обеспечения, сообщающий процессору о наступлении какого-либо события, требующего немедленного внимания. Прерывание извещает процессор о наступлении высокоприоритетного события, требующего прерывания текущего кода, выполняемого процессором. Процессор отвечает приостановкой своей текущей активности, сохраняя свое состояние, и выполняя функцию, называемую обработчиком прерывания (или программой обработки прерывания), который реагирует на событие и обслуживает его, после чего возвращает управление в прерванный код.

Адресные пространства

Каждый компьютер обладает определенным объемом оперативной памяти, используемой для хранения исполняемых программ. В самых простых операционных системах в памяти присутствует только одна программа. Для запуска второй программы сначала нужно удалить первую, а затем на ее место загрузить в память вторую.

Более сложные операционные системы позволяют одновременно находиться в памяти нескольким программам. Чтобы исключить взаимное негативное влияние программ друг на друга из-за неправильного обращения к памяти (одна программа обращается к участку оперативной памяти, где в это время хранятся данные или коды совершенно другой программы или даже самой операционной системы), что приведет к общему сбою, необходим защитный механизм. Этот механизм управляется операционной системой.

Кроме необходимости защиты памяти управление адресным пространством процессов необходимо и по другой причине. Обычно каждому процессу отводится для использования некоторый непрерывный набор адресов, как правило, от нуля и до некоторого максимума. В самом простом случае максимальный объем адресного пространства, выделяемого процессу, меньше объема оперативной памяти. Таким образом, процесс может заполнить свое адресное пространство и для его размещения в оперативной памяти будет достаточно места. При этом на многих компьютерах используется 32- или 64-разрядная адресация, позволяющая иметь адресное пространство размером 2^{32} или 2^{64} байт соответственно. Что произойдет, если адресное пространство процесса превышает объем оперативной памяти, установленной на компьютере, а процессу требуется использовать все свое пространство целиком? На первых компьютерах такой процесс неизменно терпел крах. В наше время, на современных компьютерах, применяется

технология виртуальной памяти, которая состоит в том, что операционная система хранит часть адресного пространства в оперативной памяти, а часть – на диске, по необходимости меняя их фрагменты местами. По сути, операционная система создает абстракцию адресного пространства в виде набора адресов, на которые может ссылаться процесс. Адресное пространство отделено от физической памяти машины и может быть как больше, так и меньше нее. Управление адресными пространствами и физической памятью является важной частью работы операционной системы.

Файлы

Другим ключевым понятием, поддерживаемым практически всеми операционными системами, является файловая система. Основная функция операционной системы – скрыть специфику дисков и других устройств ввода-вывода и предоставить программисту удобную и понятную абстрактную модель, состоящую из независимых от устройств файлов. Вполне очевидно, что для создания, удаления, чтения и записи файлов понадобятся системные вызовы. Перед тем, как файл будет готов к чтению, он должен быть найден на диске и открыт, а после считывания – закрыт. Для проведения этих операций предусмотрены системные вызовы.

Чтобы предоставить место для хранения файлов, многие операционные системы используют каталог как способ объединения файлов в группы. Для создания и удаления каталогов нужны системные вызовы. Они также нужны для помещения в каталог существующего файла и удаления его оттуда. Элементами каталога могут быть либо файлы, либо другие каталоги.

Каждый файл, принадлежащий иерархии каталогов, может быть обозначен своим полным именем с указанием пути к файлу, начиная с вершины иерархии – корневого каталога. Этот абсолютный путь состоит из списка каталогов, которые нужно пройти от корневого каталога, чтобы добраться до файла, где в качестве разделителей компонентов служат символы косой черты (слеша).

В любой момент времени у каждого процесса есть текущий рабочий каталог, относительно которого рассматриваются пути файлов, не начинающиеся с косой черты. Процесс может изменить свой рабочий каталог, воспользовавшись системным вызовом, определяющим новый рабочий каталог. Перед тем как с файлом можно будет работать в режиме записи или чтения, он должен быть открыт. На этом этапе происходит также проверка прав доступа. Если доступ разрешен, система возвращает целое число, называемое дескриптором файла, который используется в последующих операциях. Если доступ запрещен, то возвращается код ошибки.

Ввод-вывод данных

У всех компьютеров имеются физические устройства для получения входной и вывода выходной информации. Существует масса разнообразных устройств ввода-вывода: клавиатуры, мониторы, принтеры и т.д., а также АЦП и ЦАП. Управление всеми этими устройствами возлагается на операционную систему. Поэтому у каждой операционной системы для управления такими устройствами существует своя подсистема ввода-вывода. Некоторые программы ввода-вывода не зависят от конкретного устройства, то есть в равной мере подходят для применения со многими или со всеми устройствами ввода-вывода. Другая часть программ, например драйверы устройств, предназначена для определенных устройств ввода-вывода.

Безопасность

Компьютеры содержат большой объем информации, и часто пользователям нужно защитить ее и сохранить ее конфиденциальность. Возможно, это электронная почта, бизнес-планы, налоговые декларации и многое другое. Управление безопасностью системы также возлагается на операционную систему: например, она должна обеспечить доступ к файлам только пользователям, имеющим на это право.

Чтобы понять сам замысел возможной организации работы системы безопасности, обратимся в качестве примера к системе UNIX. Файлам в UNIX присваивается 9-разрядный двоичный код защиты. Этот код состоит из трехбитных полей. Одно поле – для владельца, второе – для представителей группы, в которую он входит (разделяет пользователей на группы системный администратор), третье – для всех остальных. В каждом поле есть бит, определяющий доступ для чтения, бит, определяющий доступ для записи, и бит, определяющий доступ для выполнения. Эти три бита называются rwx-битами (read, write, execute). Например, код защиты rwxr-x--x означает, что владельцу доступны чтение, запись или выполнение файла, остальным представителям его группы разрешается чтение или выполнение файла (но не запись), а всем остальным разрешено выполнение файла (но не чтение или запись). Дефис (минус) означает, что соответствующее разрешение отсутствует.

Кроме защиты файлов существует множество других аспектов безопасности. Один из них – это защита системы от нежелательных вторжений как с участием, так и без участия людей (например, путем вирусных атак).

Оболочка

Операционная система представляет собой программу, выполняющую системные вызовы. Не являясь частью операционной системы, оболочка нашла широкое применение как средство доступа ко многим ее функциям.

Когда не применяется графический пользовательский интерфейс, она также является основным интерфейсом между пользователем, сидящим за своим терминалом, и операционной системой. Существует множество оболочек, Оболочка запускается после входа в систему любого пользователя. В качестве стандартного устройства ввода и вывода оболочка использует терминал1. Свою работу она начинает с вывода приглашения — знака доллара, сообщающего пользователю, что оболочка ожидает приема команды. Например, если теперь пользователь наберет на клавиатуре В наши дни на большинстве персональных компьютеров используется графический пользовательский интерфейс. По сути, графический пользовательский интерфейс — это просто программа (или совокупность программ), работающая поверх операционной системы наподобие оболочки. В системах Linux этот факт проявляется явным образом, поскольку у пользователя есть выбор по крайней мере из двух сред, реализующих графический пользовательский интерфейс: Gnome и KDE. Или он может вообще не выбрать ни одну из них, воспользовавшись окном терминала из X11. В Windows также есть возможность заменить стандартный менеджер рабочего стола (Windows Explorer) какой-нибудь другой программой путем внесения изменений в некоторые значения реестра, хотя этой возможностью практически никто не пользуется.

Редакторы, компиляторы, ассемблеры, компоновщики, утилиты и интерпретаторы команд по определению не являются частью операционной системы. Они относятся к инструментальному ПО.

Классы операционных систем

История операционных систем насчитывает уже более полувека. За это время было разработано огромное количество разнообразных операционных систем, но не все они получили широкую известность. В данном разделе вкратце, для сведения, рассмотрим классы операционных систем. Для компьютерно-интегрированных технологий наиболее важное значение имеют операционные системы жесткого реального времени.

Операционные системы мейнфреймов

К высшей категории относятся операционные системы мейнфреймов (больших универсальных машин) — компьютеров, занимающих целые залы и до сих пор еще встречающихся в крупных центрах обработки корпоративных данных. Такие компьютеры отличаются от персональных компьютеров объемами ввода-вывода данных. Мейнфреймы, имеющие тысячи дисков и петабайты данных, – весьма обычное явление. Мейнфреймы также находят применение в качестве мощных веб-серверов, серверов крупных интернет-магазинов и серверов, занимающихся межкорпоративными транзакциями.

Операционные системы мейнфреймов ориентированы преимущественно на одновременную обработку множества заданий, большинство из которых требует колоссальных объемов ввода-вывода данных. Работа в режиме разделения времени дает возможность множеству удаленных пользователей одновременно запускать на компьютере свои задания, например запросы к большой базе данных. Все эти функции тесно связаны друг с другом, и зачастую операционные системы универсальных машин выполняют их в комплексе. Примером операционной системы универсальных машин может послужить OS/390, наследница OS/360. Однако эти операционные системы постепенно вытесняются вариантами операционной системы UNIX, например Linux.

Серверные операционные системы

Чуть ниже по уровню стоят серверные операционные системы. Они работают на серверах, которые представлены очень мощными персональными компьютерами, рабочими станциями или даже универсальными машинами. Они одновременно обслуживают по сети множество пользователей, обеспечивая им общий доступ к аппаратным и программным ресурсам. Серверы могут предоставлять услуги печати, хранения файлов или веб-служб. Интернет-провайдеры для обслуживания своих клиентов обычно задействуют сразу несколько серверных машин. При обслуживании веб-сайтов серверы хранят веб-страницы и обрабатывают поступающие запросы. Типичными представителями серверных операционных систем являются Solaris, FreeBSD, Linux и Windows Server 201x.

Многопроцессорные операционные системы

Сейчас все шире используется объединение множества центральных процессоров в единую систему, что позволяет добиться большой вычислительной мощности. В зависимости от того, как именно происходит это объединение, а также каковы ресурсы общего пользования, эти системы называются параллельными компьютерами, мультикомпьютерами или многопроцессорными системами. Им требуются специальные операционные системы, в качестве которых часто применяются особые версии серверных операционных систем, оснащенные специальными функциями связи, сопряжения и синхронизации. С появлением многоядерных процессоров для персональных компьютеров операционные системы даже обычных настольных компьютеров и ноутбуков стали работать по меньшей мере с небольшой многопроцессорной системой. Со временем число ядер будет только расти. На многопроцессорных системах могут работать многие популярные операционные системы, включая Windows и Linux.

Операционные системы персональных компьютеров

К следующей категории относятся операционные системы персональных компьютеров. Все их современные представители поддерживают многозадачный режим. При этом довольно часто уже в процессе загрузки на одновременное выполнение запускаются десятки программ. Задачей операционных систем персональных компьютеров является качественная поддержка работы отдельного пользователя. Они широко используются для обработки текстов, создания электронных таблиц, игр и доступа к Интернету. Типичными примерами могут служить операционные системы Linux, FreeBSD, Windows 7, Windows 8, Windows 10 и OS X (macOS) компании Apple..

Операционные системы карманных персональных компьютеров, планшетов и смартфонов

Продолжая двигаться по нисходящей ко все более простым системам, мы дошли до планшетов, смартфонов и других карманных компьютеров. Эти компьютеры, изначально известные как КПК, или PDA (Personal Digital Assistant — персональный цифровой секретарь), представляют собой небольшие компьютеры, которые во время работы держат в руке. Самыми известными их представителями являются смартфоны и планшеты. На этом рынке доминируют операционные системы Android от Google и iOS от Apple, но у них имеется множество конкурентов. Большинство таких устройств обладают многоядерными процессорами, GPS, камерами и другими датчиками, достаточным объемом памяти и сложными операционными системами.

Встроенные операционные системы

Встроенные системы работают на компьютерах, которые управляют различными устройствами. Поскольку на этих системах установка пользовательских программ не предусматривается, их обычно компьютерами не считают. Примерами устройств, где устанавливаются встроенные компьютеры, могут послужить микроволновые печи, телевизоры, автомобили, пишущие DVD, обычные телефоны и MP3-плееры. В основном встроенные системы отличаются тем, что на них ни при каких условиях не будет работать стороннее программное обеспечение. В микроволновую печь невозможно загрузить новое приложение, поскольку все ее программы записаны в ПЗУ. Следовательно, отпадает необходимость в защите приложений друг от друга и операционную систему можно упростить. Наиболее популярными в этой области считаются операционные системы Embedded Linux, QNX и VxWorks.

Операционные системы сенсорных узлов

Сети, составленные из миниатюрных сенсорных узлов, связанных друг с другом и с базовой станцией по беспроводным каналам, развертываются для различных целей. Такие сенсорные сети используются для защиты периметров зданий, охраны государственной границы, обнаружения возгораний в лесу, измерения температуры и уровня осадков в целях составления прогнозов погоды, сбора информации о перемещениях противника на поле боя и многого другого. Узлы такой сети представляют собой миниатюрные компьютеры, питающиеся от батареи и имеющие встроенную радиосистему. Они ограничены по мощности и должны работать длительный период времени в необслуживаемом режиме на открытом воздухе, часто в сложных климатических условиях. Сеть должна быть достаточно надежной и допускать отказы отдельных узлов, что по мере потери емкости батарей питания будет случаться все чаще. Каждый сенсорный узел является настоящим компьютером, оснащенным процессором, оперативной памятью и постоянным запоминающим устройством, а также одним или несколькими датчиками. На нем работает небольшая, но настоящая операционная система, обычно управляемая событиями и откликающаяся на внешние события или периодически производящая измерения по сигналам встроенных часов. Операционная система должна быть небольшой по объему и несложной, поскольку основными проблемами этих узлов являются малая емкость оперативной памяти и ограниченное время работы батарей. Так же как и у встроенных систем, все программы являются предварительно загруженными, и пользователи не могут запустить программу, загруженную из Интернета, что значительно упрощает всю конструкцию. Примером широко известной операционной системы для сенсорных узлов может послужить TinyOS.

Операционные системы реального времени

Еще одна разновидность операционных систем — это системы реального времени. Эти системы характеризуются тем, что время для них является ключевым параметром. Например, в системах управления производственными процессами компьютеры, работающие в режиме реального времени, должны собирать сведения о процессе и использовать их для управления станками на предприятии. Довольно часто они должны отвечать очень жестким временным требованиям. Например, когда автомобиль перемещается по сборочному конвейеру, то в определенные моменты времени должны осуществляться вполне конкретные операции. Если, к примеру, сварочный робот приступит к сварке с опережением или опозданием, машина придет в негодность. Если операция должна быть проведена точно в срок (или в определенный период времени), то мы имеем

дело с системой **жесткого реального времени**. Множество подобных систем встречается при управлении производственными процессами, в авиационно-космическом электронном оборудовании, в военной и других подобных областях применения. Эти системы должны давать абсолютные гарантии того, что определенные действия будут осуществляться в конкретный момент времени. Системами жесткого реального времени являются такие системы как QNX, RTOS, VxWorks.

Другой разновидностью подобных систем является система мягкого реального времени, в которой хотя и нежелательно, но вполне допустимо несоблюдение срока какого-нибудь действия, что не наносит непоправимого вреда. К этой категории относятся цифровые аудио- или мультимедийные системы. Смартфоны также являются системами мягкого реального времени.

Поскольку к системам реального времени предъявляются очень жесткие требования, иногда операционные системы представляют собой простую библиотеку, сопряженную с прикладными программами, где все тесно взаимосвязано и между частями системы не существует никакой защиты. Примером такой системы может послужить eCos. Категории операционных систем для КПК, встроенных систем и систем реального времени в значительной степени перекрываются друг с другом по свойственным им признакам. Практически все они имеют по крайней мере некоторые аспекты систем мягкого реального времени. Встроенные системы и системы реального времени работают только с тем программным обеспечением, которое вложили в них разработчики этих систем; пользователи не могут добавить в этот арсенал собственное программное обеспечение, что существенно облегчает решение задач защиты. КПК и встроенные системы предназначены для индивидуальных потребителей, а системы реального времени чаще используются в промышленном производстве. Тем не менее, несмотря на все это, у них есть определенное количество общих черт.

Операционные системы смарт-карт

Самые маленькие операционные системы работают на смарт-картах. Смарт-карта представляет собой устройство размером с кредитную карту, имеющее собственный процессор. На операционные системы для них накладываются очень жесткие ограничения по требуемой вычислительной мощности процессора и объему памяти. Некоторые из смарт-карт получают питание через контакты считывающего устройства, в которое вставляются, другие — бесконтактные смарт-карты — получают питание за счет эффекта индукции, что существенно ограничивает их возможности. Некоторые из них способны справиться с одной-единственной функцией, например с

электронными платежами, но существуют и многофункциональные смарт-карты.

Контрольные вопросы по теме

Уровень модуля

Уровень курса

1. Понятие абстрактных ресурсов.
2. Операционная система в качестве менеджера ресурсов.
3. Понятие "процесс" в операционных системах.
4. Понятие "адресное пространство" в операционных системах.
5. Понятие "файл" в операционных системах.
6. Оболочка операционной системы.
7. Классы операционных систем.
8. Операционные системы реального времени.

Лекции № 14 ***Тема:*** Базы данных

Оглавление

Источники:

1. Таненбаум Э., Остин Т. Архитектура компьютера. 6-е изд. — СПб.: Питер, 2013. — 816 с.: ил.

   https://rutracker.org/forum/viewtopic.php?t=4956359

2. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. — СПб.: Питер, 2015. — 1120 с.: ил.
3. Бабак В.П. Теоретические основы информационно-измерительных систем: Учебник / В.П. Бабак, С.В. Бабак, В.С. Ерёменко и др. – К., 2014. – 832 с.

Понятие баз данных

Возможно, вы еще не знаете, что входит в понятие базы данных, но то, что вы ими постоянно пользуетесь абсолютно точно. Каждый раз, когда вы что-то ищете в поисковике, вы используете базу данных. Когда вы вводите свои логин и пароль для входа на какой-нибудь сервис, они сравниваются со значениями, которые хранятся в базе данных этого сервиса.

Несмотря на то, что мы постоянно используем базы данных, для многих остается непонятным, что же это такое на самом деле. И связано это отчасти с тем, что одни и те же термины, относящиеся к базам данных, используются людьми для определения совершенно разных вещей.

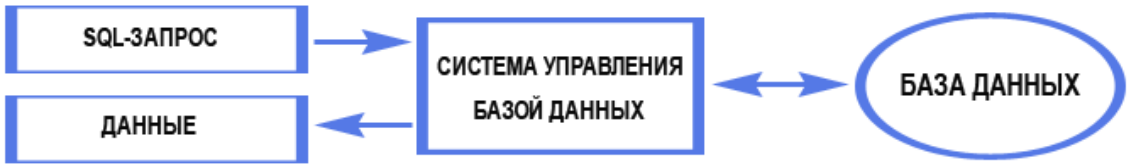
Давайте разберемся с терминами и понятиями баз данных:

База данных - набор сведений, хранящихся некоторым упорядоченным способом. Можно сравнить базу данных со шкафом, в котором хранятся документы. Иными словами, база данных - это хранилище данных. Сами по себе базы данных не представляли бы интереса, если бы не было систем управления базами данных (СУБД).

Система управления базами данных - это совокупность языковых и программных средств, которая осуществляет доступ к данным, позволяет их создавать, менять и удалять, обеспечивает безопасность данных и т.д. В общем СУБД - это система, позволяющая создавать базы данных и манипулировать сведениями из них. А осуществляет этот доступ к данным СУБД посредством специального языка - SQL.

SQL - язык структурированных запросов, основной задачей которого является предоставление простого способа считывания и записи информации в базу данных.

Итак, простейшая схема работы с базой данных выглядит примерно так:



По характеру использования СУБД делят на однопользовательские (предназначенные для создания и использования БД на персональном компьютере) и многопользовательские (предназначенные для работы с единой БД нескольких компьютеров, объединенных в локальные сети). Вообще деление по характеру использования можно представить следующей схемой:

Не вдаваясь далее в подробности, отметим, что на сегодняшний день число используемых СУБД исчисляется десятками. Наиболее известные однопользовательские СУБД - Microsoft Visual FoxPro и Access, многопользовательские - MS SQL Server, Oracle и MySQL.

В определении базы данных говорится, что это сведения, которые упорядочены некоторым образом. А как собственно они упорядочены? Об этом и пойдет речь в следующем разделе.

Структура базы данных

Создавая базу данных, мы стремимся упорядочить информацию по различным признакам для того, чтобы потом извлекать из нее необходимые нам данные в любом сочетании. Сделать это возможно, только если данные структурированы. Структурирование - это набор соглашений о способах представления данных. Понятно, что структурировать информацию можно по-разному. В зависимости от структуры различают иерархическую, сетевую, реляционную, объектно-ориентированную и гибридную модели баз данных. Самой популярной на сегодняшний день является реляционная структура, поэтому об остальных упомянем лишь вскользь.

Иерархическая структура базы данных

Это древовидная структура представления информации. Ее особенность в том, что каждый узел на более низком уровне имеет связь только с одним узлом на более высоком уровне. Посмотрим, например, на фрагмент иерархической структуры базы данных "Университет":

Из структуры понятно, что на одной кафедре может работать несколько преподавателей. Такая связь называется "один ко многим" (одна кафедра - много преподавателей). Но если мы попытаемся добавить в эту структуру группы студентов, то нам понадобится связь "многие ко многим":

(один преподаватель может работать со многими группами, а одна группа может учиться у многих преподавателей), а такой связи в иерархической структуре быть не может (т.к. связь может быть только с одним узлом на более высоком уровне). Это основной недостаток подобной структуры базы данных.

Сетевая структура базы данных

По сути, это расширение иерархической структуры. Все то же самое, но существует связь "многие ко многим". Сетевая структура базы данных позволяет нам добавить группы в наш пример. Недостатком сетевой модели является сложность разработки серьезных приложений.

Реляционная структура базы данных

Все данные представлены в виде простых таблиц, разбитых на строки и столбцы, на пересечении которых расположены данные. Подробно об этом мы будем говорить в следующем разделе.

Объектно-ориентированные и гибридные базы данных

В объектно-ориентированных базах данных данные хранятся в виде объектов, что очень удобно. Но на сегодняшний день такие БД еще не распространены, т.к. уступают в производительности реляционным.

Гибридные БД совмещают в себе возможности реляционных и объектно-ориентированных, поэтому их часто называют объектно-

реляционными. Примером такой СУБД является Oracle, начиная с восьмой версии.

Несомненно, такие БД будут развиваться в будущем, но пока первенство остается за реляционными структурами..

Реляционные базы данных

Реляционные базы данныхсостоят из таблиц. Каждая таблица состоит из столбцов (их называют полями или атрибутами) и строк (их называют записями или кортежами). Таблицы в реляционных базах данных обладают рядом свойств. Основными являются следующие:

- В таблице не может быть двух одинаковых строк.
- Столбцы располагаются в определенном порядке, который создается при создании таблицы. В таблице может не быть ни одной строки, но обязательно должен быть хотя бы один столбец.
- У каждого столбца есть уникальное имя (в пределах таблицы), и все значения в одном столбце имеют один тип (число, текст, дата...).
- На пересечении каждого столбца и строки может находиться только *атомарное* значение (одно значение, не состоящее из группы значений). Таблицы, удовлетворяющие этому условию, называют нормализованными.

В теории баз данных атомарные данные (значения) – это атрибуты, которые хранят единственное значение и не являются ни списком, ни множеством значений. Иными словами, это такие данные, разделение которых на составляющие приводит к потере их смысла с точки зрения решаемой задачи. Например, если атрибут «Цена» содержит значение 15, то попытка разделить его на 1 и 5 приведет к полной бессмыслице. Данные, не являющиеся атомарными, называются составными.

Все будет понятнее на примере. Предположим, мы захотели создать базу данных для форума. У форума есть зарегистрированные пользователи, которые создают темы и оставляют сообщения в этих темах. Эта информация и должна храниться в базе данных.

Теоретически (на бумаге) мы можем все это расположить в одной таблице, например, так:

Имя	E-mail	Пароль	Созданные темы	Созданные сообщения

Но это противоречит свойству атомарности (одно значение в одной ячейке), а в столбцах Темы и Сообщения у нас предполагается неограниченное количество значений. Значит, нашу таблицу надо разбить на три: Пользователи, Темы и Сообщения.

Пользователи

Имя	E-mail	Пароль

Темы

Наименование	Автор

Сообщения

Текст	Автор

Наша таблица Пользователи удовлетворяет всем условиям. А вот таблицы Темы и Сообщения - нет. Ведь в таблице не может быть двух одинаковых строк, а где гарантия, что один пользователь не оставит два одинаковых сообщения, например:

Сообщения

Текст	Автор
Думаю надо сделать так…	Кирилл
Согласен	Вася
А еще можно сделать так…	Семен
Согласен	Вася

Кроме того, мы знаем, что каждое сообщение обязательно относится к какой-либо теме. А как это можно узнать из наших таблиц? Никак. Для решения этих проблем, в реляционных базах данных существуют ключи.

Первичный ключ (сокращенно PK - primary key) - столбец, значения которого во всех строках различны. Первичные ключи могут быть логическими (естественными) и суррогатными (искусственными). Так, для нашей таблицы Пользователи первичным ключом может стать столбец e-mail (ведь теоретически не может быть двух пользователей с одинаковым e-mail). На практике лучше использовать суррогатные ключи, т.к. их применение позволяет абстрагировать ключи от реальных данных. Кроме того, первичные ключи менять нельзя, а что если у пользователя сменится e-mail?

Суррогатный ключ представляет собой дополнительное поле в базе данных. Как правило, это порядковый номер записи (хотя вы можете задавать их на свое усмотрение, контролируя, чтобы они были уникальны). Давайте внесем поля первичных ключей в наши таблицы:

Пользователи

id пользователя	Имя	E-mail	Пароль
1	Кирилл	kirill@mail.ru	Gh345fgh
2	Вася	vasy@rambler.ru	As3bh7
3	Семен	semen@yandex.ru	gk4bb6

Темы

id темы	Наименование	Автор
1	О рыбалке	Кирилл
2	Велосипеды	Вася
3	Ночные клубы	Семен
4	О рыбалке	Вася

Сообщения

id сообщения	Текст	Автор
1	Думаю надо сделать так…	Кирилл
2	Согласен	Вася
3	А еще можно сделать так…	Семен
4	Согласен	Вася

Теперь каждая запись в наших таблицах уникальна. Нам осталось установить соответствие между темами и сообщениями в них. Делается это также при помощи первичных ключей. В таблицу сообщения мы добавим еще одно поле:

Сообщения

id сообщения	Текст	Автор	id темы
1	Думаю надо сделать так…	Кирилл	1
2	Согласен	Вася	4
3	А еще можно сделать так…	Семен	1
4	Согласен	Вася	1

Теперь понятно, что сообщение с id=2 принадлежит теме "О рыбалке" (id темы = 4), созданной Васей, а остальные сообщения принадлежат теме "О рыбалке" (id темы = 1), созданной Кириллом. Такое поле называется внешний ключ (сокращенно FK - foreign key). Каждое значение этого поля соответствует какому-либо первичному ключу из таблицы "Темы". Так устанавливается однозначное соответствие между сообщениями и темами, к которым они относятся.

Последний нюанс. Предположим, у нас добавился новый пользователь, и зовут его тоже Вася:

Пользователи

id пользователя	Имя	E-mail	Пароль
1	Кирилл	kirill@mail.ru	*******
2	Вася	vasy@rambler.ru	*******
3	Семен	semen@yandex.ru	*******
4	Вася	vasy@mail.ru	*******

Как мы узнаем, какой именно Вася оставил сообщения? Для этого поля автор в таблицах "Темы" и "Сообщения" мы сделаем также внешними ключами:

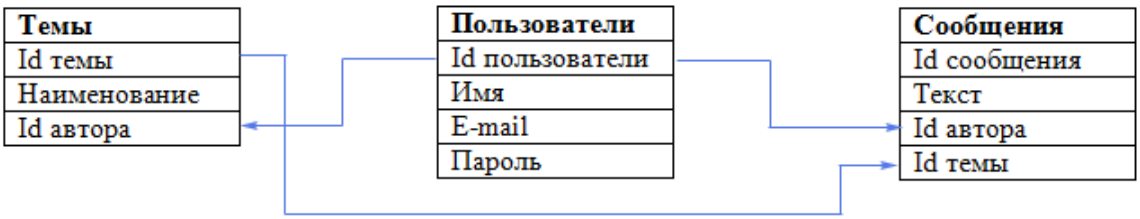
Темы

id темы	Наименование	id автора
1	О рыбалке	1
2	Велосипеды	2
3	Ночные клубы	3
4	О рыбалке	1
5	К кому обратиться	4

Сообщения

id сообщения	Текст	id автора	id темы
1	Думаю надо сделать так…	1	1
2	Согласен	2	4
3	А еще можно сделать так…	3	1
4	Согласен	2	1

Наша база данных готова. Схематично ее можно представить так:



В нашей маленькой базе данных всего три таблички, а если бы их было 10 или 100? Понятно, что сразу невозможно представить все таблицы, поля и связи, которые нам могут понадобиться. Именно поэтому проектирование базы данных всегда начинается с ее концептуальной модели.

Язык программирования SQL

Structured Query Language (структурированный язык запросов) или SQL – язык управления базами данных для реляционных баз данных. SQL изначально создан IBM. Он была принят в качестве стандарта американским Национальным институтом стандартов (ANSI) в 1986 и ISO в 1987.

Как следует из названия, язык программирования SQL предназначен для конкретных, ограниченных целей – запросов данных, содержащихся в реляционной базе данных. Как таковой, он представляет собой набор инструкций языка программирования для создания выборок данных, а не процедурный язык, такой как C или BASIC, которые предназначены для решения гораздо более широкого круга проблем. Основной подход заключается в том, что позволяется в запросы SQL встраивать команды процедурного языка программирования и взаимодействовать с базой данных.

SQL состоит из четырех отдельных частей (групп):

1. Первая - это Data Manipulation Language или DML (язык управления данными). DML является подмножеством языка, используемого для запроса к базам данных, добавления, обновления и удаления данных.

- SELECT является одной из наиболее часто используемых команд; она выбирает данные, удовлетворяющие заданным условиям
- INSERT используется для добавления строк для существующей таблицы.
- UPDATE используется для изменения значений данных в существующей строке таблицы.
- DELETE определяет строки, которые будут удалены из таблицы.

2. Вторая группа Data Definition Language или DDL (язык определения данных). DDL позволяет пользователю определять новые таблицы и связанные с ними элементы. Основными командами DDL являются команды "создавать" и "удалять" объекты.

- CREATE определяет объекты (например, таблицы), которые будут созданы в базе данных.
- DROP определяет, какие существующие объекты в базе данных будут удалены, как правило, безвозвратно.

Некоторые системы баз данных также поддерживают команду ALTER, которая позволяет пользователю изменять существующий объект по-разному. Например, так можно произвести добавление столбцов в существующую таблицу.

3. Третьей группой ключевых слов SQL является Data Control Language или DCL(язык контроля данных). DCL отвечает за права доступа к данным и позволяет пользователю контролировать, кто имеет доступ, чтобы просматривать или манипулировать данными в базе данных. Имеются два основных ключевых слова:

- GRANT - разрешает пользователю выполнять операции

- REVOKE - удаляет или ограничивает возможность пользователю выполнять операции.

4. Язык управления транзакциями (TCL) используется для контроля обработки транзакций в БД. Обычно операторы TCL включают commit для подтверждения изменений, сделанных в ходе транзакции, rollback для их отмены и savepoint для разбиения транзакции на несколько меньших частей.

Простой пример применения оператора SQL SELECT

Синтаксис оператора SELECT

```
SELECT column_list
FROM table_name
[WHERE условие]
[GROUP BY условие]
[HAVING условие]
[ORDER BY условие]
```

SELECT Ключевое слово, которое сообщает базе данных о том, что оператор является запросом. Все запросы начинаются с этого слова, за ним следует пробел.

`Column_list` Список столбцов таблицы, которые выбираются запросом. Столбцы, не указанные в операторе, не будут включены в результат. Если необходимо вывести данные всех столбцов, можно использовать сокращенную запись. Звездочка (*) означает полный список столбцов.

FROM `table_name` Ключевое слово, которое должно присутствовать в каждом запросе. После него через пробел указывается имя таблицы, являющейся источником данных.

Код в скобках является не обязательным в операторе SELECT. Он необходим для более точного определения запроса.

Также необходимо сказать, что SQL код является регистронезависимым. Это означает, что запись SELECT можно написать как select. СУБД не отличит эти две записи, однако советуют все операторы SQL писать прописными буквами, чтобы его легко можно было отличить от другого кода.

Примеры рассмотрим на таблице Salespeople (продавцы). Таблица выглядит так:

snum	sname	city	comm
1001	Peel	London	0,12
1002	Serres	San Jose	0,13
1003	Axelrod	New York	0,1
1004	Motika	London	0,11
1007	Rifkin	Barcelona	0,15

Столбцы таблицы Salespeople:

snum	Номер продавца
sname	Имя продавца
city	Город
comm	Коммисионные продавца, в десятичной форме

Пример использования оператора SELECT

1. Необходимо вывести список продавцов, и отобразить их имена (sname)

```
SELECT sname
FROM Salespeople
```

Результат:

sname
Peel
Serres
Axelrod
Motika
Rifkin

В данном запросе, после оператора SELECT идет имя столбца, которое необходимо отобразить. После ключевого слова FROM указывается имя таблицы.

2. Необходимо вывести список продавцов, и отобразить их имена и город (sname и city)

```
SELECT sname , city
FROM Salespeople
```

Результат:

sname	city
Peel	London
Serres	San Jose
Axelrod	New York
Motika	London
Rifkin	Barcelona

Здесь после оператора SELECT перечисляются столбцы, которые необходимо вывести. Имена столбцов пишутся через запятую.

3. Необходимо вывести всю таблицу

Для этого можно использовать разный синтаксис написания запросов. Перечисляем каждый столбец после оператора SELECT:

```
SELECT snum , sname , city , comm
FROM Salespeople
```

Или можно добиться того же результата, используя сокращенную запись:

```
SELECT * FROM Salespeople
```

Результат:

snum	sname	city	comm
1001	Peel	London	0,12
1002	Serres	San Jose	0,13
1003	Axelrod	New York	0,1
1004	Motika	London	0,11
1007	Rifkin	Barcelona	0,15

Контрольные вопросы по теме

Уровень модуля

Уровень курса

1. Понятие баз данных.
2. Структура базы данных.
3. Реляционные базы данных.
4. Язык программирования SQL.

Лекции № 15 ***Тема:*** Технология "Клиент-сервер"

Оглавление

Источники:

1. Таненбаум Э., Остин Т. Архитектура компьютера. 6-е изд. — СПб.: Питер, 2013. — 816 с.: ил.

   https://rutracker.org/forum/viewtopic.php?t=4956359

2. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. — СПб.: Питер, 2015. — 1120 с.: ил.
3. Бабак В.П. Теоретические основы информационно-измерительных систем: Учебник / В.П. Бабак, С.В. Бабак, В.С. Ерёменко и др. – К., 2014. – 832 с.

Введение в клиент-серверные технологии

Как правило компьютеры и программы, входящие в состав информационной системы, не являются равноправными. Некоторые из них владеют ресурсами (файловая система, процессор, принтер, база данных и т.д.), другие имеют возможность обращаться к этим ресурсам. Компьютер (или программу), управляющий ресурсом, называют сервером этого ресурса (файл-сервер, сервер базы данных, вычислительный сервер...). Клиент – это аппаратный или программный компонент вычислительной системы, посылающий запросы серверу и получающий от сервера запрошенную информацию. Сервер – аппаратный или программный компонент вычислительной системы, выполняющий сервисные (обслуживающие) функции по запросу клиента, предоставляя ему доступ к определённым ресурсам или услугам. Клиент и сервер какого-либо ресурса могут находится как на одном компьютере, так и на различных компьютерах, связанных сетью.

Архитектура «клиент-сервер» определяет общие принципы организации взаимодействия в системе, где имеются *серверы*, узлы-поставщики некоторых специфичных функций (сервисов) и *клиенты*, потребители этих функций.

Основная идея архитектуры «клиент-сервер» состоит в разделении сетевого приложения на несколько компонентов, каждый из которых реализует специфический набор сервисов. Компоненты такого приложения могут выполняться на разных компьютерах, выполняя серверные и/или клиентские функции. Это позволяет повысить надежность, безопасность и производительность сетевых приложений и сети в целом.

Практические реализации такой архитектуры называются **клиент-серверными технологиями**. Каждая технология определяет собственные или использует имеющиеся правила взаимодействия между клиентом и сервером, которые называются *протоколом обмена (протоколом взаимодействия)*.

Архитектура «клиент-сервер» широко применяется в измерительных и управляющих системах. Можно говорить, что технология «клиент-сервер» является частью компьютерно-интегрированных технологий. Датчики компьютерно-интегрированных систем являются источником измерительной информации, и они отправляют измеренные данные в программы управления и сбора данных. То есть, датчики предоставляют сервис по предоставлению измерительных данных. Они, по существу, являются серверами. А программа управления и сбора данных в данном случае является клиентом. Об этом взаимодействии применительно к компьютерно-интегрированным системам речь пойдет в разделе, посвященном OPC серверам. Первоначально будут рассмотрены основные понятия и принципы клиент-серверного взаимодействия.

Отметим, что программы диспетчерского управления и сбора данных в компьютерно-интегрированных технологиях называют SCADA-системами (**S**upervisory **C**ontrol **A**nd **D**ata **A**cquisition – диспетчерское управление и сбор данных) или SCADA-пакетами, поскольку они на самом деле представляют собой целый комплекс программ. SCADA-системам будет посвящена одна из лекций.

Виды клиент-серверных технологий

Архитектура клиент-сервер применяется в большом числе сетевых технологий, используемых для доступа к различным сетевым сервисам. Кратко рассмотрим некоторые типы таких сервисов (и серверов). Это лишь несколько типов из всего многообразия клиент-серверных технологий, используемых как в локальных, так и в глобальных сетях.

Web-серверы

Изначально представляли доступ к гипертекстовым документам по протоколу HTTP (Huper Text Transfer Protocol). Сейчас поддерживают расширенные возможности, в частности работу с бинарными файлами (изображения, мультимедиа и т.п.).

Серверы приложений

Предназначены для централизованного решения прикладных задач в некоторой предметной области. Для этого пользователи имеют право запускать серверные программы на исполнение. Использование серверов приложений позволяет снизить требования к конфигурации клиентов и упрощает общее управление сетью.

Серверы баз данных

Серверы баз данных используются для обработки пользовательских запросов на языке SQL. При этом СУБД находится на сервере, к которому и подключаются клиентские приложения.

Файл-серверы

Файл-сервер хранит информацию в виде файлов и представляет пользователям доступ к ней. Как правило файл-сервер обеспечивает и определенный уровень защиты от несакционированного доступа.

Почтовые серверы

Представляют услуги по отправке и получению электронных почтовых сообщений.

OPC-серверы

OPC-серверы предназначены для управления объектами автоматизации и технологическими процессами. Они реализуют интерфейс OPC — набор спецификаций стандартов, разработанный специально для целей автоматизации.

Клиентское приложение

Для доступа к тем или иным сетевым сервисам используются клиенты, возможности которых характеризуются понятием «толщины». Оно определяет конфигурацию оборудования и программное обеспечение, имеющиеся у клиента. Рассмотрим возможные граничные значения:

«Тонкий» клиент

Этот термин определяет клиента, вычислительных ресурсов которого достаточно лишь для запуска необходимого сетевого приложения через web-интерфейс. Пользовательский интерфейс такого приложения формируется средствами статического HTML (выполнение JavaScript не предусматривается), вся прикладная логика выполняется на сервере. Для работы тонкого клиента достаточно лишь обеспечить возможность запуска web-браузера, в окне которого и осуществляются все действия. По этой причине web-браузер часто называют "универсальным клиентом".

«Толстый» клиент

Таковым является рабочая станция или персональный компьютер, работающие под управлением собственной дисковой операционной системы и имеющие необходимый набор программного обеспечения. К сетевым серверам «толстые» клиенты обращаются в основном за дополнительными услугами (например, доступ к web-серверу или корпоративной базе данных). Так же под «толстым» клиентом подразумевается и клиентское сетевое приложение, запущенное под управлением локальной ОС. Такое приложение совмещает компонент представления данных (графический пользовательский интерфейс ОС) и прикладной компонент (вычислительные мощности клиентского компьютера).

Как уже отмечалось выше, отдельным видом клиента является SCADA-система.

Клиент-серверная архитектура

Двухзвенная архитектура

В любой сети, построенной на современных сетевых технологиях, присутствуют элементы клиент-серверного взаимодействия, чаще всего на основе **двухзвенной архитектуры**. Двухзвенной она называется из-за необходимости распределения трех базовых компонентов: представление

данных, прикладной компонент, управление ресурсами – между двумя узлами (клиентом и сервером).

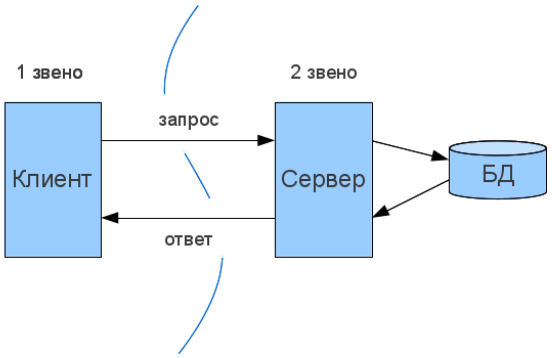


Рис.1. Двухзвенная клиент-серверная архитектура

Двухзвенная архитектура используется в клиент-серверных системах, где сервер отвечает на клиентские запросы напрямую и в полном объеме, при этом используя только собственные ресурсы. Т.е. сервер не вызывает сторонние сетевые приложения и не обращается к сторонним ресурсам для выполнения какой-либо части запроса (рис. 1)

Трехзвенная архитектура

В сетевых технологиях все больше и больше начинают использовать распределенные вычисления. Они реализуются на основе модели сервера приложений, где сетевое приложение разделено на две и более частей, каждая из которых может выполняться на отдельном компьютере. Выделенные части приложения взаимодействуют друг с другом, обмениваясь сообщениями в заранее согласованном формате. В этом случае двухзвенная клиент-серверная архитектура становится **трехзвенной**.

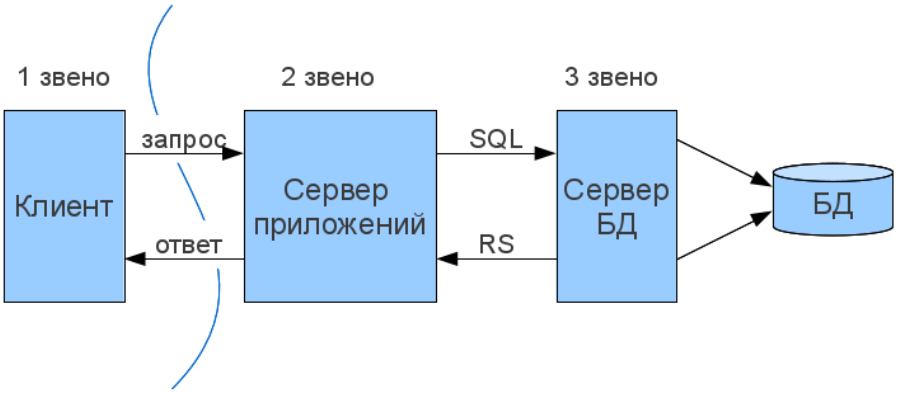


Рис.2. Трехзвенная клиент-серверная архитектура

Как правило, третьим звеном в трехзвенной архитектуре становится сервер приложений, т.е. компоненты распределяются следующим образом (рис. 2):

1. Представление данных – на стороне клиента.
2. Прикладной компонент – на выделенном сервере приложений (как вариант, выполняющем функции промежуточного ПО).
3. Управление ресурсами – на сервере БД, который и представляет запрашиваемые данные.

Трехзвенная архитектура может быть расширена до многозвенной путем выделения дополнительных серверов, каждый из которых будет представлять собственные сервисы и пользоваться услугами прочих серверов разного уровня. Абстрактный пример многозвенной модели приведен на 3.

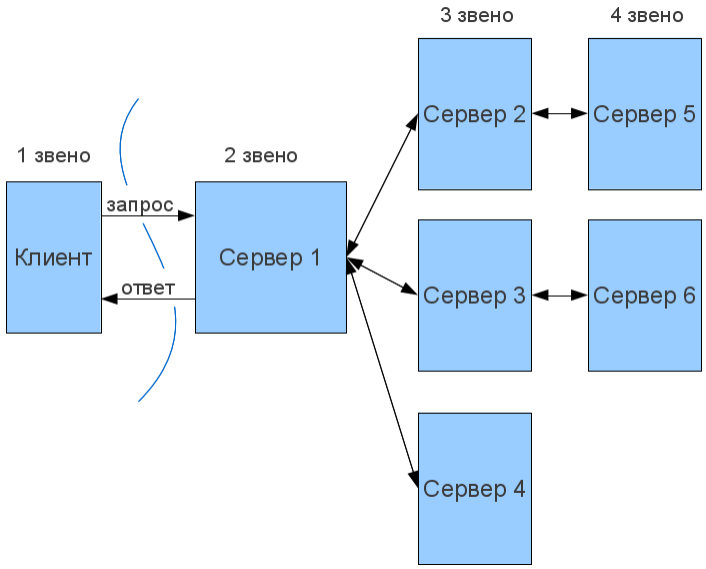


Рис.3. Многозвенная клиент-серверная архитектура

OPC сервер в компьютерно-интегрированных технологиях

Как уже было указано, OPC-серверы специально предназначены для управления объектами автоматизации и технологическими процессами. Они реализуют так называемый интерфейс OPC, представленный в соответствующих стандартах. Стандарт OPC разработан международной организацией OPC Foundation, членами которой являются более 400 фирм, работающих в области средств автоматизации и измерительной техники. Первая версия OPC стандарта была выпущена в 1998 г

Обзор стандарта OPC

Главной целью стандарта OPC явилось обеспечение возможности совместной работы (интероперабельности) средств автоматизации, функционирующих на разных аппаратных платформах, в разных промышленных сетях и производимых разными фирмами. До разработки OPC стандарта SCADA пакет нужно было адаптировать к каждому новому оборудованию индивидуально. Существовали длинные списки

"поддерживаемого оборудования", очень сложной была техническая поддержка. При модификации оборудования нужно было вносить изменения во все драйверы, каждый из которых поддерживал протокол обмена только с одной клиентской программой. Число таких драйверов доходило до сотен.

После появления стандарта OPC практически все SCADA-пакеты были перепроектированы как OPC-клиенты, а каждый производитель аппаратного обеспечения стал снабжать свои контроллеры, модули ввода-вывода, интеллектуальные датчики и исполнительные устройства стандартным OPC сервером. Благодаря появлению стандартизации интерфейса стало возможным подключение любого физического устройства к любой SCADA, если они оба соответствовали стандарту OPC. Разработчики получили возможность проектировать только один драйвер для всех SCADA-пакетов, а пользователи получили возможность выбора оборудования и программ без прежних ограничений на их совместимость.

Стандарт OPC относится только к интерфейсам, которые OPC сервер предоставляет клиентским программам. Метод же взаимодействия сервера с аппаратурой (например, с модулями ввода-вывода), стандартом не предусмотрен и его реализация возлагается полностью на разработчика аппаратуры. Поэтому стандарт OPC может быть использован не только для взаимодействия SCADA с "железом", но и для обмена данными с любым источником данных, например, с базой данных или с GPS приемником.

OPC сервер как средство взаимодействия с техническим устройством может быть использован при разработке заказных программ на C++, Visual Basic, VBA и т. п.

Стандарт OPC состоит из нескольких частей:

- OPC DA (OPC Data Access) - спецификация для обмена данными между клиентом (например SCADA) и аппаратурой (контроллерами, модулями ввода-ввода и др.) в реальном времени;
- OPC Alarms & Events (A&E) - спецификация для уведомления клиента о событиях и сигналах тревоги, которые посылаются клиенту по мере их возникновения. Этот сервер пересылает аварийные сигналы, действия оператора, информационные сообщения, результаты контроля состояния системы;
- OPC HDA (Historical Data Access) - спецификация для доступа к предыстории процесса (к сохраненным в архиве данным). Сервер обеспечивает унифицированный способ доступа с помощью DCOM технологии. Обеспечивает чтение, запись и изменение данных;
- Batch - спецификация для особых физико-химических технологических процессов обработки материалов, которые не

являются непрерывными. В таких процессах выполняется загрузка нескольких видов сырья в определенных пропорциях согласно рецепту, устанавливаются режимы обработки, а после выполнения цикла обработки и выгрузки готового материала загружается новая партия сырья. OPC сервер выполняет обмен между клиентом и сервером рецептами, характеристиками технологического оборудования, условиями и результатами обработки;

- OPC Data eXchange - спецификация для обмена данными между двумя OPC DA серверами через сеть Ethernet;
- OPC Security - спецификация, которая определяет методы доступа клиентов к серверу, которые обеспечивают защиту важной информации от несанкционированной модификации;
- OPC XML-DA - набор гибких, согласующихся друг с другом правил и форматов для представления первичных данных с помощью языка XML, веб технологий и сообщений SOAP (см. раздел "Архитектура автоматизированной системы".);
- OPC Complex Data - дополнительные спецификации к OPC DA и XML-DA, которые позволяют серверам работать со сложными типами данных, такими как бинарные структуры и XML-документы;
- OPC Commands - набор программных интерфейсов, который позволяет OPC клиентам и серверам идентифицировать, посылать и контролировать команды, исполняемые в техническом устройстве (в контроллере, модуле ввода-вывода);
- OPC Unified Architecture - принципиально новый набор спецификаций, который уже не базируется на DCOM технологии.

Наиболее широко используется спецификации OPC DA и реже - OPC HDA.

Тег

При использовании сервер OPC используется понятие тег. Тег – это метка, адрес, идентификатор данных (точка, источник или канал поступления данных), переменная, которая приписана этим данным. В простейшем случае один датчик и есть один тег - переменная в которой находится текущее значение. Например, к системе подключен датчик температуры. Показаниям датчика приписан уникальный тег. Эти показания можно найти по этому уникальному тегу.

В более сложном случае одному устройству могут соответствовать несколько тегов. Например, к сети подключен программируемый логический контроллер (ПЛК), а к этому ПЛК подключено несколько датчиков, каждый

из которых измеряет какой-то один параметр. Тогда каждому датчику (точнее - показаниям каждого датчика) будет соответствовать свой уникальный тег.

Бывает, что одно и то же устройство измеряет несколько параметров, например: влажность и температуру. Тогда данным измерения влажности будет соответствовать один тег, а данным измерения температуры – другой тег.

OPC DA сервер

Сервер OPC DA является наиболее широко используемым в промышленной автоматизации. Он обеспечивает обмен данными (запись и чтение) между клиентской программой и физическими устройствами. Данные состоят из трех полей: значение, качество и временная метка. Параметр качества данных позволяет передать от устройства клиентской программе информацию о выходе измеряемой величины за границы динамического диапазона, об отсутствии данных, ошибке связи и другие.

Существует четыре стандартных режима чтения данных из OPC сервера:

- *синхронный режим*: клиент посылает запрос серверу и ждет от него ответ;
- *асинхронный режим*: клиент отправляет запрос и сразу же переходит к выполнению других задач. Сервер после выполнения функции запроса посылает клиенту уведомление и тот забирает предоставленные данные;
- *режим подписки*: клиент сообщает серверу список тегов, значения которых сервер должен отправлять клиенту только в случае их изменения. Для того, чтобы шум данных не был принят за их изменение, вводится понятие "мертвой зоны", которая слегка превышает максимально возможный размах помехи;
- *режим обновления данных*: клиент вызывает одновременное чтение всех активных тегов. Активными называются все теги, кроме обозначенных как "пассивные". Такое деление тегов уменьшает загрузку процессора обновлением данных, принимаемых из физического устройства.

В каждом из этих режимов данные могут читаться либо из кэша OPC сервера, либо непосредственно из физического устройства. Чтение из кэша выполняется гораздо быстрее, но данные к моменту чтения могут устареть. Поэтому сервер должен периодически освежать данные с максимально возможной частотой. Для уменьшения загрузки процессора используют параметр частоты обновления, которая может быть установлена для каждой группы тегов индивидуально. Кроме того, некоторые теги можно сделать

пассивными, тогда их значения не будут обновляться данными из физического устройства.

Запись данных в физическое устройство может быть выполнена только двумя методами: синхронным и асинхронным и выполняется сразу в устройство, без промежуточной буферизации. В синхронном режиме функция записи выполняется до тех пор, пока из физического устройства не поступит подтверждение, что запись выполнена. Этот процесс может занимать много времени, в течение которого клиент находится в состоянии ожидания завершения функции и не может продолжать выполнение своей работы. При асинхронной записи клиент отправляет данные серверу и сразу продолжает свою работу. После окончания записи сервер отправляет клиенту соответствующее уведомление.

В соответствии со стандартом, OPC сервер во время инсталляции автоматически регистрируется в реестре Windows. Запуск сервера осуществляется так же, как любой другой программы или автоматически из клиентской программы.

На рис. 4 показано диалоговое окно OPC сервера. Сервер позволяет выполнить поиск физических устройств, подключенных к COM-порту компьютера. На рис. 4 окно сервера слева показывает, что к компьютеру подключены три модуля ввода: NL16HV, NL8TI и NL8AI. Для удобства представления измеряемых величин (тегов) на объекте автоматизации имена тегов могут быть составными и путь к тегу может быть представлен в виде дерева, как показано на рис. 4. Имя выделенного на рисунке тега выглядит как "NL8TI.Laboratory32.Top.Vin4". Все имена и их структура задаются с помощью средств окна OPC сервера.

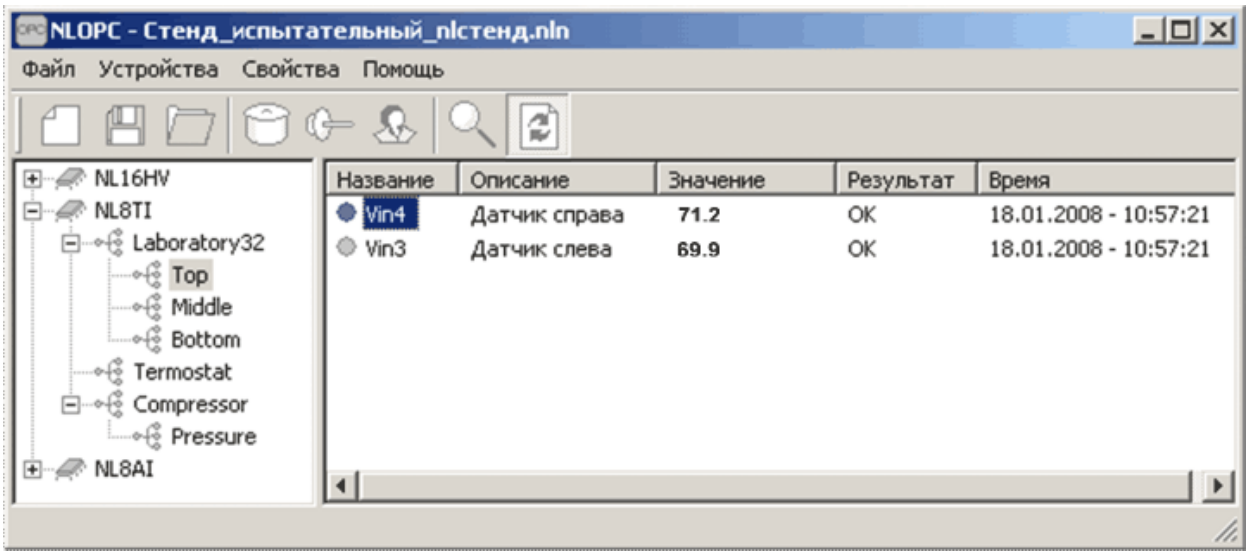


Рис. 4. Пример диалогового окна OPC сервера

При использовании OPC клиента (например, SCADA), имена тегов, доступные через OPC сервер, представляются в аналогичной форме в окне навигатора тегов (рис. 5). Клиент показывает все OPC серверы, установленные на компьютерах, доступных по сети Ethernet, и позволяет использовать все теги этих серверов.

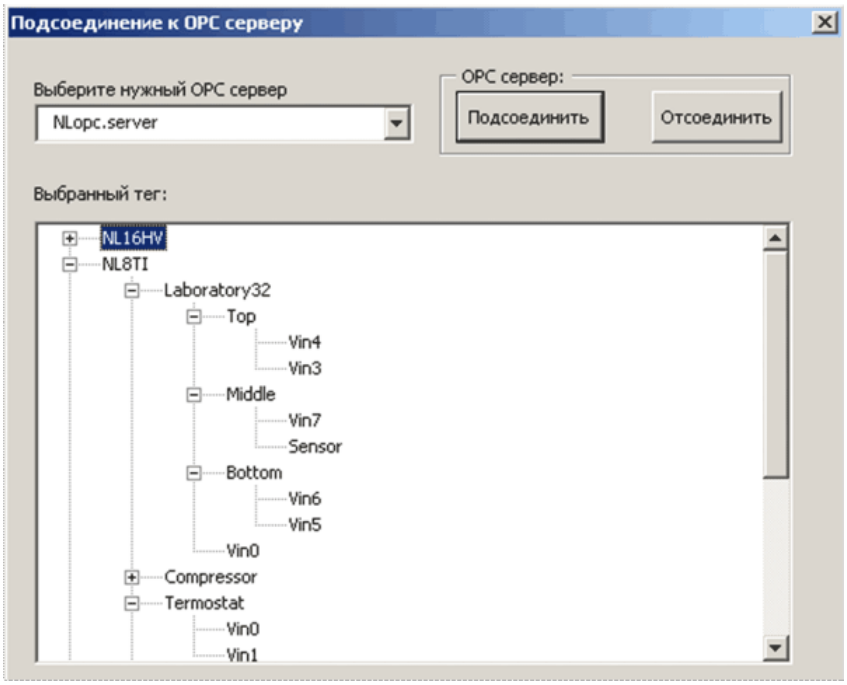


Рис. 5. Пример диалогового окна навигатора тегов OPC клиента

Пример архитектуры систем, включающих OPC серверы и OPC клиенты, показаны на рис. 6 и рис. 7. В качестве OPC клиента может выступать программа на языке C++ (например, SCADA-пакет) или программа на языке Visual Basic, VBA, Delphi или любая другая программа, поддерживающая внедрение COM-объектов (рис. 6). Программа на языке C++ взаимодействует с OPC сервером через интерфейс OPC Custom, а программа на Visual Basic, VBA, Delphi - через интерфейс автоматизации OPC Automation. OPC сервер и OPC клиенты могут работать только на компьютерах и контроллерах с операционными системами, поддерживающими технологию DCOM (например, Windows XP и Windows CE).

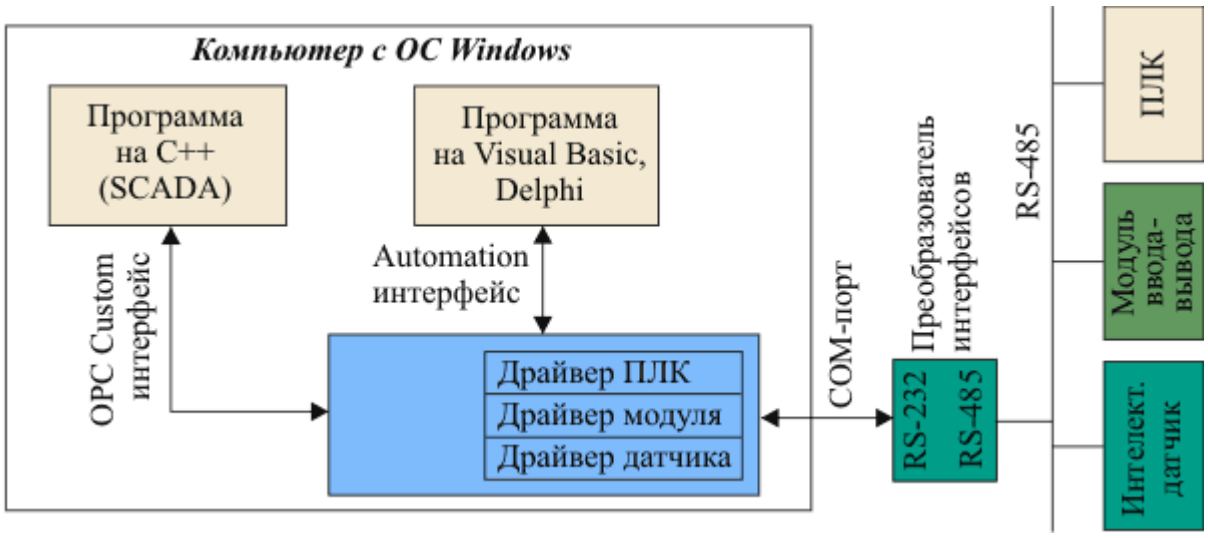


Рис. 6. Простой пример взаимодействия прикладных программ и физических устройств через OPC сервер на одном компьютере.

OPC сервер подключается к физическим устройствам любым способом; эти способы стандартом не предусмотрены. Например, сервер NLopc фирмы НИЛ АП использует для каждого физического устройства свой драйвер (рис. 6).

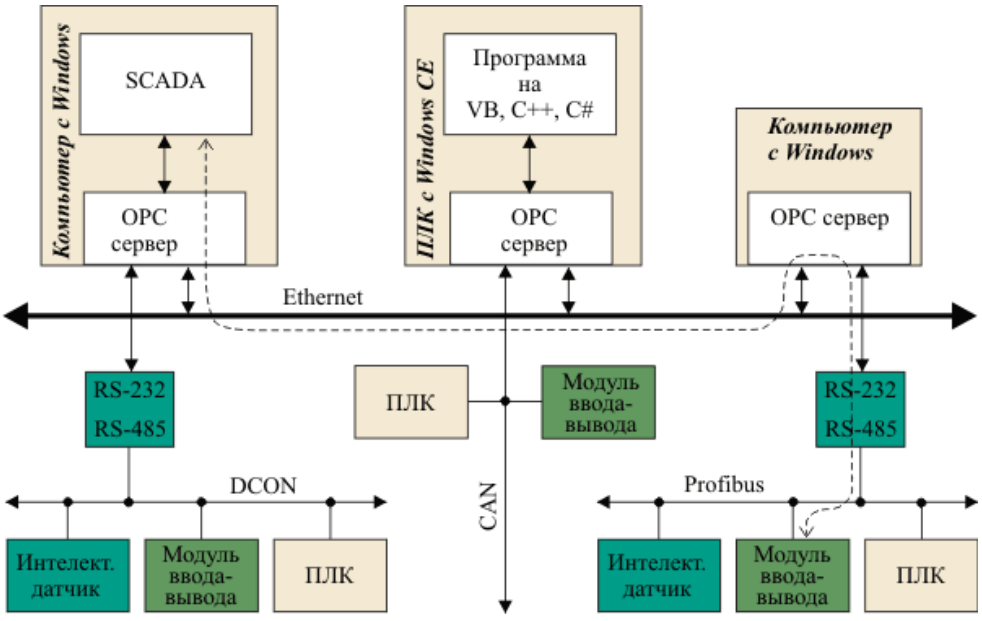


Рис. 7. Пример применения OPC технологии для сетевого доступа к данным в системах автоматизации

Клиентская программа и OPC сервер могут быть установлены на одном и том же компьютере, как показано на рис. 6, или на разных компьютерах сети Ethernet (рис. 7). При наличии нескольких компьютеров каждый из них может содержать OPC сервер и подключенные к нему физические устройства. В такой системе любой OPC клиент с любого компьютера может обращаться к любому OPC серверу, в том числе к расположенному на другом компьютере

сети. Это достигается благодаря технологии DCOM, использующей удаленный вызов процедур (RPC - Remote Procedure Call). Например, SCADA на рис. 7 может обратиться за данными к модулю ввода-вывода по пути, указанному на рис. 7 штриховой линией. Обратим внимание, что компьютеры и контроллеры в такой архитектуре могут работать с разными промышленными сетями. Обмен данными с программируемыми логическими контроллерами (ПЛК), работающими с ОС Windows CE, выполняется точно так, как с компьютерами.

При использовании оборудования разных производителей на компьютере (контроллере) может быть установлено несколько OPC серверов разных производителей, однако OPC сервер монопольно занимает COM-порт компьютера (поскольку непрерывно выполняет обновление данных), поэтому количество портов должно быть равно количеству OPC серверов. Для наращивания количества COM портов можно использовать преобразователи интерфейса USB в RS-232. К разным портам компьютера могут быть подключены разные промышленные сети. В этом случае OPC серверы используются в качестве межсетевых шлюзов.

Контрольные вопросы по теме

Уровень модуля

Уровень курса

1. Введение в клиент-серверные технологии.
2. Виды клиент-серверных технологий.
3. Клиент-серверная архитектура.
4. OPC сервер в компьютерно-интегрированных технологиях. Стандарт OPC.
5. Понятие "тег" в компьютерно-интегрированных технологиях.
6. Режимы чтения данных из OPC сервера.

Лекции № 16 ***Тема:*** Программирование контроллеров

Оглавление

Источники:

1. Таненбаум Э., Остин Т. Архитектура компьютера. 6-е изд. — СПб.: Питер, 2013. — 816 с.: ил.

   https://rutracker.org/forum/viewtopic.php?t=4956359

2. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. — СПб.: Питер, 2015. — 1120 с.: ил.
3. Бабак В.П. Теоретические основы информационно-измерительных систем: Учебник / В.П. Бабак, С.В. Бабак, В.С. Ерёменко и др. – К., 2014. – 832 с.

Программное обеспечение в компьютерно-интегрированных технологиях

Современные системы промышленной и лабораторной автоматизации позволяют решать широкий круг задач, которые можно разделить на несколько групп, имеющих свои особенности:

- автоматизация управления технологическими процессами (АСУ ТП);
- взаимодействие системы с диспетчером (оператором);
- автоматизированный контроль и измерения (мониторинг);
- обеспечение безопасности;
- дистанционное управление, измерение, сигнализация (задачи телемеханики).

История развития программных средств автоматизации показала, что все особенности отдельных применений можно учесть путем настройки нескольких универсальных программ на выполнение конкретной задачи. К таким универсальным программам относятся:

- OPC сервер (рассмотрен в предыдущей лекции);
- средства МЭК-программирования контроллеров;
- SCADA-пакеты.

Для систем автоматизации, не связанных с АСУ ТП, используются программы LabVIEW, MatLab, HP-VEE и др., ориентированные на автоматизацию эксперимента, измерений или математическую обработку их результатов. Для простых задач или широко тиражируемых приложений бывает экономически эффективно использовать заказное программирование на C++ или Visual Basic с применением покупных ActiveX элементов, снижающих трудоемкость разработки.

Развитие программных средств автоматизации

Для решения перечисленных выше задач первоначально использовались универсальные языки программирования высокого уровня и команда профессиональных программистов. Однако практика показала крайне низкую эффективность такой разработки. Оказалось, что разработка системы должна выполняться не программистами, а специалистами той предметной области, которая нуждается в автоматизации, т. е. технологами, а также системными интеграторами, которые осуществляют комплексное внедрение системы.

Необходимость в разработке средств программирования, предназначенных специально для систем автоматизации и ориентированных на технологов, была вызвана следующими причинами:

- требованием надежности программного обеспечения. Система, написанная целиком на алгоритмическом языке для конкретного заказа,

содержала слишком много программного кода, на тщательную разработку и тестирование которого не хватало времени;

- сжатыми сроками внедрения системы и ограниченной стоимостью работ. Для создания системы в короткий срок при ограниченном бюджете требовалось большое количество готовых универсальных программных компонентов, уже написанных и тщательно оттестированных;
- необходимостью модификации системы в процессе ее эксплуатации. Внести изменения в специализированную программу мог только написавший ее программист, который к этому времени обычно работал уже на другом предприятии. Поэтому вместо того, чтобы модифицировать программное обеспечение, его приходилось переписывать заново;
- требованиями совместимости с другими системами автоматизации, работающими на том же предприятии. Были необходимы стандартные интерфейсы между программами, созданными разными производителями на разных аппаратно-программных платформах;
- высокими требованиями к качеству пользовательского интерфейса. Ограниченный бюджет времени и финансовых ресурсов не позволял разработать достаточно хороший программный интерфейс на универсальных алгоритмических языках.

Разделение труда по созданию программных средств автоматизации

Перечисленные причины привели к следующему разделению труда по созданию программных средств для систем автоматизации: фирмы, специализирующиеся на программном обеспечении, создают универсальные системы программирования задач автоматизации (SCADA-пакеты и средства МЭК-программирования), а инжиниринговые фирмы (системные интеграторы) адаптируют эти средства к нуждам конкретного заказчика. В результате достигается решение всех перечисленных выше проблем. Более того, благодаря существенному упрощению процесса адаптации по сравнению с классическим программированием изменения в алгоритмы управления могут быть внесены, например, технологом эксплуатирующей организации без привлечения системных интеграторов или программистов.

В настоящее время заказные программы естественным путем вытеснены с рынка промышленной автоматизации SCADA-пакетами и аналогичными универсальными средствами автоматизации, а также средствами программирования контроллеров на языках стандарта МЭК 61131-3.

Графическое программирование

Языки визуального программирования появились в начале 90-х годов и содержат большое число стандартных функций и библиотек, а также готовых средств визуализации. Они позволяют создавать очень удобные и эффектные программы, однако достигается это за счет резкого увеличения объема программного кода. Поэтому языки визуального программирования, как и текстовые, по-прежнему не позволяют модифицировать алгоритмы силами технологов без участия профессиональных программистов.

Настоящую революцию в программировании систем автоматизации сделали языки графического программирования. Одним из первых в этом классе был графический язык среды Simulink, входящей в состав Matlab (MathWorks Inc), а также языки LabVIEW (National Instruments) и HP-VEE (Hewlett Packard). Они были предназначены и успешно использовались для сбора данных, моделирования систем автоматизации, автоматического управления, обработки собранных данных и их визуального представления в виде графиков, таблиц, звука, с помощью компьютерной анимации. Графические языки были настолько простыми и естественными, что для их освоения зачастую было достаточно метода проб и ошибок без использования учебников и консультаций. Человек, не знакомый с программированием на алгоритмических языках, пользуясь только логикой и понимая постановку прикладной задачи, мог собрать работающее приложение из готовых компонентов, набрасывая их мышкой на экран монитора и проводя графические связи для указания потоков информации.

Первые языки программирования алгоритмов работы систем автоматизации были нестандартными. Каждая фирма, создававшая контроллер или SCADA-пакет, предлагала свой язык. Это требовало от системных интеграторов дополнительных усилий и затрудняло освоение новых SCADA пакетов и средств программирования контроллеров.

Поэтому появление в 1993 году стандарта на языки программирования контроллеров МЭК 61131-3 стало большим шагом в направлении создания открытых систем автоматизации и обеспечило снижение стоимости разработки, сокращение сроков, повышение качества реализации алгоритмов автоматизации и возможность детального изучения языков программирования, пригодных для любого контроллера. МЭК 61131-3 устанавливал стандарты для пяти языков программирования, рассчитанных на специалистов разных профессий, не связанных с программированием.

Графический интерфейс

Создание графических интерфейсов пользователя на компьютере явилось большим достижением в направлении развития средств

диспетчерского управления. Главным эффектом от применения графического интерфейса является существенное снижение количества ошибок, допускаемых оператором (диспетчером) в стрессовых ситуациях при управлении производственными процессами. Проектирование пользовательского интерфейса основано на следующих принципах:

- *узнаваемость*: назначение элементов экрана должно быть понятно без предварительного обучения, допустимые манипуляции с этими элементами также должны быть понятны интуитивно. Пользовательский интерфейс не должен содержать излишней детализации;
- *логичность*: пользователь, имеющий опыт работы с одной программой, должен быть способен быстро, практически без обучения, адаптироваться к любой аналогичной программе;
- *отсутствие "сюрпризов"*: знакомые из прошлого опыта операции с элементами на экране должны вызывать знакомые реакции системы;
- *восстанавливаемость*: система не должна быть чувствительна к ошибкам оператора. Оператор должен иметь возможность отменить любое свое неправильное действие. Для этого используются многократные подтверждения, отмены, возврат на несколько шагов назад, установка контрольных точек и т. п.;
- *наличие удобной справки*, подсказок, встроенных в пользовательский интерфейс, средств контекстного поиска и замены;
- *адаптация к опыту пользователя*: начинающий пользователь должен иметь более простой интерфейс с большим количеством подсказок. Для опытного пользователя количество подсказок должно быть уменьшено, поскольку они мешают в работе.

Связь с физическими устройствами

Связь программного обеспечения с физическими устройствами в системах автоматизации осуществляется с помощью методов DDE, OLE, COM, DCOM и OPC.

Технология обмена данными между приложениями Windows с аббревиатурой DDE (Dynamical Data Exchange - "динамический обмен данными") - появилась в 1987 г. вместе с Windows 2.0. В промышленной автоматизации DDE использовалась для обмена данными между SCADA в качестве DDE-клиента и физическим устройством, которое поставлялось с DDE сервером.

После появления OLE (Object Linking and Embedding - "связывание и внедрение объектов") фирмы Microsoft, а позже COM (Component Object Model - "модель многокомпонентных объектов") и DCOM(Distributed COM - "COM для распределенных систем") технология DDE была полностью вытеснена этими новыми средствами, которые оказались гораздо более эффективными.

Технология COM предоставляет средства для взаимодействия между разрозненными программными модулями, написанными на разных языках программирования, которые собираются в единую систему во время исполнения. Взаимодействие COM объекта с другими программами или программными модулями выполняется через программные интерфейсы с использованием метода "клиент-сервер".

Одной из составляющих COM является Automation - средства взаимодействия программ, написанных на C++ с программами на языке VBA (Visual Basic for Application) или Delphi, а также с программами на языках сценариев (VBScript, JScript). Благодаря автоматизации COM-объект может быть также размещен и исполняться на веб-странице.

Расширение COM в виде DCOM позволяет программам взаимодействовать между собой, даже если они исполняются на разных компьютерах локальной сети. Поэтому DCOM явилась универсальной программной технологией, которая как нельзя лучше позволяет осуществить взаимодействие между SCADA в качестве клиента и сервером, обеспечивающим интерфейс к аппаратным средствам промышленной автоматизации. Именно благодаря этому свойству DCOM была использована в качестве базы для разработки стандарта OPC - "OLE for Process Control" - "OLE для управления процессами", который лежит в основе всех современных SCADA пакетов, взаимодействующих с аппаратурой через OPC сервер.

Базы данных

Системы автоматизации работают с большими объемами данных, которые необходимо хранить, сортировать, группировать, извлекать и представлять в виде, удобном для пользователя. Данные извлекаются с помощью языка запросов SQL (Structured Query Language - "структурированный язык запросов"), который стал стандартом в системах автоматизации. Наиболее распространенными системами управления базами данных (СУБД) являются Microsoft SQL Server, Wonderware Industrial SQL Server, Microsoft Access и Excel. Основными свойствами СУБД являются:

- наличие пользовательского интерфейса на базе языка запросов SQL;

- возможность одновременного обслуживания нескольких пользователей;
- корректность работы с данными.

Открытые системы используют обращение к СУБД через драйвер ODBC (Open Database Connectivity - "подключение к открытой базе данных"). ODBC используется, когда необходимо обеспечить независимость прикладной программы от типа СУБД или типа операционной системы и требуется подключиться к нескольким различным СУБД (например, одновременно к MS SQL Server, MS Excel, MS Access, Paradox и др.). При использовании нескольких ODBC драйверов ими управляет менеджер драйверов. ODBC драйвер транслирует стандартный SQL запрос в формат запроса для конкретной СУБД. Таким образом, для работы с новой базой данных пользователю достаточно добавить в систему новый ODBC драйвер, не изменяя прикладную программу.

Операционные системы реального времени

Быстродействие *программируемого логического контроллера* (ПЛК) или компьютера влияет на величину динамической погрешности системы автоматизации и запас ее устойчивости при наличии обратной связи. Для улучшения этих характеристик используют быстродействующие модули ввода-вывода и компьютер (ПЛК) с высокопроизводительным процессором. Это позволяет улучшить динамические характеристики системы, однако большинство операционных систем (ОС) не могут обеспечить одно и то же время выполнения задачи при повторных ее запусках, т. е. время выполнения является случайной величиной. В некоторых случаях непредсказуемость времени исполнения задачи приводит к отказу системы.

Пусть, к примеру, автомат подсчитывает количество бутылок на конвейере. Если бутылки появляются напротив датчика с периодом 1 с, а время реакции системы на появление бутылки составляет 0,7 с, то система кажется работоспособной. Однако, если задержка является случайной величиной, то в некоторый момент времени она может оказаться больше 1 с, что приведет к появлению случайной ошибки в количестве бутылок, т.е. к отказу системы. Величина ошибки определяется статистической функцией распределения случайных задержек.

Для устранения этой проблемы был разработан класс операционных систем, которые обеспечивают детерминированное (т.е. не случайное) время выполнения задач и время реакции на аппаратные прерывания. Такие ОС получили название операционных систем реального времени ОС РВ) и были поделены на ОС жесткого и мягкого реального времени. Отличительным

признаком ОС РВ является не время выполнение задач, а гарантированность постоянства величины этого времени для одной и той же задачи.

ОС жесткого реального времени гарантирует выполнение задачи за заранее известное время. В ОС мягкого реального времени приняты особые меры для устранения неопределенности времени выполнения, однако полностью неопределенность не устраняется.

Стандарт IEEE 1003.1 даёт следующее определение РВ: "Реальное время в операционных системах — это способность операционной системы обеспечить требуемый уровень сервиса в определённый промежуток времени". Следовательно, ОС РВ отличаются своим поведением, а не внутренним принципом построения. Поэтому если вероятность появления недопустимо больших задержек достаточно низка для достижения требуемого уровня сервиса (например, если она меньше допустимой вероятности отказа системы), то такая ОС в конкретном применении может рассматриваться как ОС РВ. В частности, в соответствии с определением стандарта POSIX операционная система Windows XP при управлении медленными (тепловыми) процессами может рассматриваться как ОС РВ.

В системах РВ обычно отсутствует виртуальная память, поскольку этот метод использует подкачку страниц с диска, время выполнения которой является непредсказуемым.

Наиболее распространенными в ПЛК и компьютерах для решения задач автоматизации являются операционные системы Windows CE, QNX Neutrino и OS-9.

Windows CE.NET

Многозадачная операционная система жесткого реального времени Windows CE.NET корпорации Microsoft поддерживает микропроцессоры с архитектурой ARM, StrongARM и xScale, MIPS, SH, X86-совместимые и имеет следующие свойства:

- допускает одновременное выполнение до 32 процессов;
- имеет 256 уровней приоритетов;
- поддерживает вытесняющую многозадачность;
- обеспечивает карусельное исполнение цепочек с одинаковым приоритетом;
- поддерживает вложенные прерывания;
- имеет среднее время обработки прерывания 2,8 мкс (на Pentium 166 МГц), поддерживает вложенные прерывания;
- обеспечивает время обработки потока прерываний (Interrupt Service Thread, IST), равное 17,9 мкс (на Pentium 166 МГц);

– в минимальной конфигурации может быть установлена при объеме ОЗУ 200 Кб.

Ядро этой ОС принципиально отличается от ядра ОС для настольных компьютеров. В Windows CE.NET объединены все возможности систем реального времени и последние технологии Windows. Планирование выполняется на основе приоритетов, для устранения инверсии используется наследование приоритетов. Несмотря на наличие возможности работы с виртуальной памятью, для обеспечения режима жесткого реального времени ее отключают.

Windows CE .NET поддерживает Microsoft Visual Studio .NET и Microsoft eMbedded Visual C++ с языками программирования Visual C++, Visual C#, and Visual Basic .NET.

QNX Neutrino

QNX Neutrino корпорации QNX Software Systems является операционной системой реального времени и обеспечивает многозадачный режим с приоритетами. Поддерживает микропроцессоры семейств ARM, StrongARM, xScale, x86, MIPS, PowerPC, SH-4.

QNX относится к микроядерным ОС (т.е. реализует только базовые функции ядра - управление адресным пространством ОЗУ и виртуальной памяти, процессами и потоками, обеспечивает межпроцессорную коммуникацию). Состоит из ядра, планировщика процессов и сервисов. Построена на основе сервисов - небольших задач, выполняющих основные функции ОС. Такая структура позволяет отключить любую ненужную функциональность, не изменяя ядро. Каждый драйвер, приложение, протокол или файловая система выполняются вне ядра, в защищенном адресном пространстве.

OS-9

Операционная система OS-9 фирмы Microware System является многозадачной и многопользовательской, работает в режиме мягкого реального времени. Используется во встраиваемых приложениях на платформах ARM, StrongARM, MIPS, PowerPC, Hitachi SuperH, x86, Pentium, XScale, Motorola 68K.

Системы программирования на языках МЭК 61131-3

Стандарт МЭК 61131-3 устанавливает пять языков программирования ПЛК, три графических и два текстовых. Первоначально стандарт назывался IEC 1131-3 и был опубликован в 1993 г. но в 1997 г. МЭК (IEC) перешел на новую систему обозначений и в названии стандарта добавилась цифра "6". Продвижением стандарта занимается организация PLCopen.

Основной целью стандарта было повышение скорости и качества разработки программ для ПЛК, а также создание языков программирования, ориентированных на технологов, обеспечение соответствия ПЛК идеологии открытых систем, исключение этапа дополнительного обучения при смене типа ПЛК.

Системы программирования, основанные на МЭК 61131-3, характеризуются следующими показателями:

- надежностью создаваемого программного обеспечения. Надежность обеспечивается тем, что программы для ПЛК создаются с помощью специально предназначенной для этого среды разработки, которая содержит все необходимые средства для написания, тестирования и отладки программ с помощью эмуляторов и реальных ПЛК, а также множество готовых фрагментов программного кода;
- возможностью простой модификации программы и наращивания ее функциональности;
- переносимостью проекта с одного ПЛК на другой;
- возможностью повторного использования отработанных фрагментов программы;
- простотой языка и ограничением количества его элементов.

Языки МЭК 61131-3 появились не как теоретическая разработка, а как результат анализа множества языков, уже используемых на практике и предлагаемых рынку производителями ПЛК. Стандарт устанавливает пять языков программирования со следующими названиями:

- структурированный текст (ST - Structured Text);
- последовательные функциональные схемы (SFC - "Sequential Function Chart");
- диаграммы функциональных блоков (FBD - Function Block Diagram);
- релейно-контактные схемы, или релейные диаграммы (LD - Ladder Diagram);
- список инструкций (IL - Instruction List).

Графическими языками являются SFC, FBD, LD. Языки IL и ST являются текстовыми.

В стандарт были введены несколько языков (а не один) для того, чтобы каждый пользователь мог применить наиболее понятный ему язык. Программисты чаще выбирают язык IL (похожий на ассемблер) или ST, похожий на язык высокого уровня Паскаль; специалисты, имеющие опыт работы с релейной логикой, выбирают язык LD, специалисты по системам

автоматического управления (САУ) и схемотехники выбирают привычный для них язык FBD.

Выбор одного из пяти языков определятся не только предпочтениями пользователя, но и смыслом решаемой задачи. Если исходная задача формулируется в терминах последовательной обработки и передачи сигналов, то для нее проще и нагляднее использовать язык FBD. Если задача описывается как последовательность срабатываний некоторых ключей и реле, то для нее нагляднее всего будет язык LD. Для задач, которые изначально формулируются в виде сложного разветвленного алгоритма, удобнее будет язык ST.

- Языки МЭК 61131-3 базируются на следующих принципах:
- вся программа разбивается на множество функциональных элементов - Program Organization Units (POU), каждый из которых может состоять из функций, функциональных блоков и программ. Любой элемент МЭК-программы может быть сконструирован иерархически из более простых элементов;
- стандарт требует строгой типизации данных. Указание типов данных позволяет легко обнаруживать большинство ошибок в программе до ее исполнения;
- имеются средства для исполнения разных фрагментов программы в разное время, с разной скоростью, а также параллельно. Например, один фрагмент программы может сканировать концевой датчик с частотой 100 раз в секунду, в то время как второй фрагмент будет сканировать датчик температуры с частотой один раз в 10 сек;
- для выполнение операций в определенной последовательности, которая задается моментами времени или событиями, используется специальный язык последовательных функциональных схем (SFC);
- стандарт поддерживает структуры для описания разнородных данных. Например, температуру подшипников насоса, давление и состояние "включено-выключено" можно описать с помощью единой структуры "Pomp" и передавать ее внутри программы как единый элемент данных;
- стандарт обеспечивает совместное использование всех пяти языков, поэтому для каждого фрагмента задачи может быть выбран любой, наиболее удобный, язык;
- программа, написанная для одного контроллера, может быть перенесена на любой контроллер, совместимый со стандартом МЭК 61131-3.

Любой ПЛК работает в циклическом режиме. Цикл начинается со сбора данных с модулей ввода, затем исполняется программа ПЛК и оканчивается цикл выводом данных в устройства вывода. Поэтому величина контроллерного цикла зависит от времени исполнения программы и быстродействия процессорного модуля.

Язык релейно-контактных схем, LD

Графический язык релейной логики впервые появился в виде электрических схем, которые состояли из контактов и обмоток электромагнитных реле. Такие схемы использовались в автоматике конвейеров для сборки автомобилей до эры микропроцессоров. Язык релейной логики был интуитивно понятен людям, слегка знакомым с электротехникой и поэтому оказался наиболее распространенным в промышленной автоматике. Обслуживающий персонал легко находил отказ в оборудовании, прослеживая путь сигнала по релейной диаграмме.

Однако язык LD проблематично использовать для реализации сложных алгоритмов, поскольку он не поддерживает подпрограммы, функции, инкапсуляцию и другие средства структурирования программ с целью повышения качества программирования. Эти недостатки затрудняют многократное использование программных компонентов, что делает программу длинной и сложной для обслуживания.

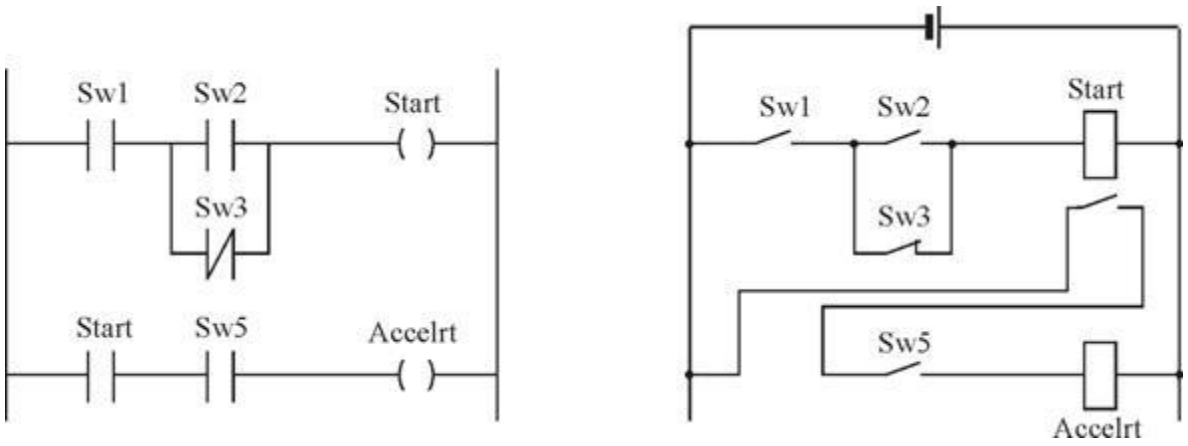


Рис. 1. Пример программы на языке LD (слева) и ее эквивалент в виде электрической цепи с реле и выключателями (справа)

Для выполнения арифметических функций в язык LD были добавлены функциональные блоки, которые выполняли операции сложения, умножения, вычисления среднего и т.д. Сложные вычисления в этом языке невозможны. Недостатком является также то, что только маленькая часть программы умещается на мониторе компьютера или панели оператора при программировании.

Несмотря на указанные недостатки, язык LD относится к наиболее распространенным в мире, хотя используется для программирования только простых задач.

Список инструкций, IL

Язык IL напоминает ассемблер и используется для реализации функций, функциональных блоков и программ, а также шагов и переходов в языке SFC. Основным достоинством языка является простота его изучения. Наиболее часто язык IL используется в случаях, когда требуется получить оптимизированный код для реализации критических секций программы, а также для решения небольших задач с малым количеством разветвлений алгоритма.

Листинг 4. Пример программы на языке IL

Метки	Операторы	Операнды	Комментарии
	LD	Voltage	(*Загрузить Voltage в аккумулятор*)
	GT	220	(*Если >220*)
	JMPCN	M1	(*Перейти к метке, если ">220" не верно*)
	LD	Current	(*Загрузить Current в аккумулятор*)
	SUB	10	(*Вычесть из аккумулятора 10 *)
	ST	Current	(*Присвоить Current значен. аккумулятора*)
M1:	LD	0	(*Загрузить в аккумулятор значение "0"*)
	ST	Out	(*Присвоить Out значение аккумулятора*)

В основе языка лежит понятие аккумулятора и переходов по меткам. Пример программы на зыке IL с комментариями приведен в листинге 4. Начинается программа с загрузки в аккумулятор значения переменной. Дальнейшие шаги программы состоят в извлечении содержимого аккумулятора и выполнении над ним ограниченного числа допустимых действий (их в языке всего 24).

Структурированный текст, ST

Язык ST является текстовым языком высокого уровня и очень сильно напоминает Паскаль:

Листинг 5. Пример программы на языке ST

```
IF Voltage>220 THEN
   Current:=Current - 10;  (*Если V>220 В, то уменьшить ток
на 10*)
ELSE
   Current:=50; Speed:= ON;(*Установить ток 50А и включить
мотор*)
END_IF;
```

Язык ST имеет много отличий от языка Паскаль и разработан специально для программирования ПЛК. Он содержит множество конструкций для присвоения значений переменным, для вызова функций и функциональных блоков, для написания выражений условных переходов, выбора операторов, для построения итерационных процессов. Этот язык предназначен в основном для выполнения сложных математических вычислений, описания сложных функций, функциональных блоков и программ.

Диаграммы функциональных блоков, FBD

FBD является графическим языком и наиболее удобен для программирования процессов прохождения сигналов через функциональные блоки. Язык FBD удобен для схемотехников, которые легко могут составить электрическую схему системы управления на "жесткой логике", но не имеют опыта программирования.

Функциональные блоки представляют собой фрагменты программ, написанных на IL, SFC или других языках, которые могут быть многократно использованы в разных частях программы и которым соответствует графическое изображение, принятое при разработке функциональных схем электронных устройств, см. рис. 2.

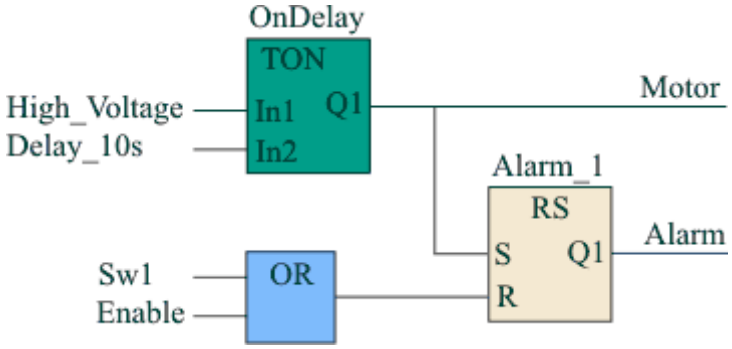


Рис. 2. Пример программы на языке FBD

Язык FBD может быть использован для программирования функций, функциональных блоков и программ, а также для описания шагов и переходов в языке SFC. Функциональные блоки инкапсулируют данные и методы, чем

напоминают объектно-ориентированные языки программирования, но не поддерживают наследование и полиморфизм.

Типичным применением языка FBD является описание "жесткой логики" и замкнутых контуров систем управления. Язык функциональных блоков является удобным также для создания и пополнения библиотеки типовых функциональных блоков, которую можно многократно использовать при программирования задач промышленной автоматизации. К типовым блокам относятся блок таймера, ПИД-регулятора, блок секвенсора, триггера, генератора импульсов, фильтра, и т. п.

Функциональные блоки стандартов МЭК 61499 и МЭК 61804

Функциональные блоки являются не просто частью языка FBD, они применяются также для моделирования и проектирования систем автоматизации. Функциональные блоки могут быть использованы также для поддержания всего жизненного цикла системы, включая проектирование, изготовление, функционирование, валидацию и обслуживание. Описанию и применению функциональных блоков посвящены, помимо МЭК 61131-3, еще и стандарты МЭК 61499 и МЭК 61804.

Стандарт МЭК 61499, состоящий из четырех частей, был опубликован в 2005 г. Он устанавливает обобщенную архитектуру функциональных блоков и предоставляет руководство для их применения в распределенных системах промышленной автоматизации. В таких системах программное обеспечение распределено между несколькими физическими устройствами (ПЛК) и несколькими функциональными блоками (ФБ), а промышленная сеть рассматривается как составная часть системы.

Особенностью ФБ в МЭК 61499 является возможность управления событиями и большая степень обобщения функциональных блоков. Стандарт МЭК 61499 может использоваться совместно с МЭК 61131-3 как средство описания базовых типов функциональных блоков для программирования ПЛК, а внутренне описание ФБ выполняется с помощью языков МЭК 61131-3.

Одной из существенных особенностей МЭК 61499 является ориентация на системы, в которых ФБ *управляются событиями*, в то время как традиционные системы автоматизации строятся обычно на базе тактирования или управления по временному расписанию. Событийное управление использовано потому, что в распределенных системах оно является более общим. Любая система с тактированием может быть представлена в виде системы с событийным управлением, но обратное не всегда верно.

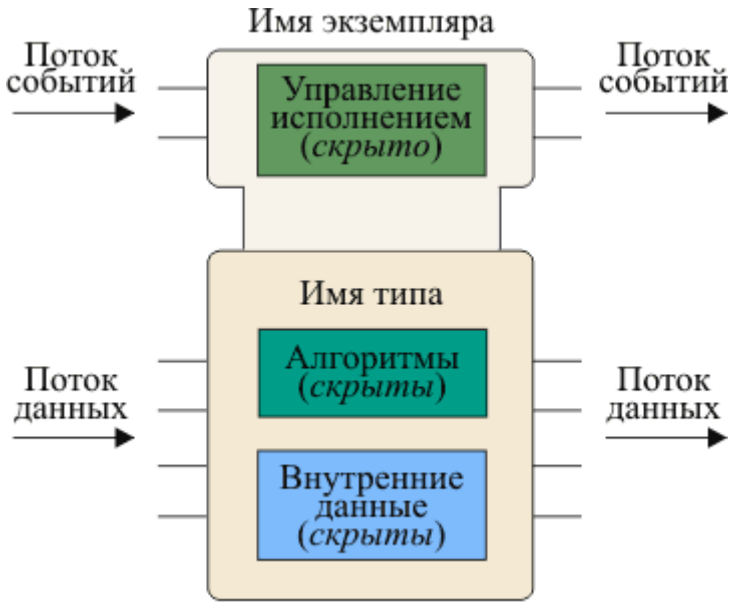


Рис. 3. Функциональный блок стандарта МЭК 61499

Архитектура функциональных блоков представляется с помощью текстового описания и графически (рис. 3). Функциональный блок характеризуется именем типа и именем экземпляра блока. Например, именем типа может быть "PID", а имен экземпляров может быть много: "PID1, PID 2, PID3, ...", по количеству ФБ, использованных в конкретной системе.

Каждый ФБ имеет множество входов и выходов для приема и передачи потока событий. Принятые события могут инициировать исполнение некоторых алгоритмов внутри блока, в результате чего могут вырабатываться события, которые передаются другим блокам системы.

ФБ имеет также множество входов, через которые поступает поток данных. Входящие данные отображаются во входные переменные, которые обрабатываются алгоритмами блока, после чего могут передаваться другим ФВ в виде выходящего потока данных. Блок может содержать также внутренние данные и соответствующие им внутренние переменные.

Каждый ФБ имеет свои функциональные характеристики, которые определяются комбинацией внутренних данных, состояний и алгоритмов, а также функциональными возможностями ресурсов устройства. Ресурс - это функциональный элемент, содержащийся в физическом устройстве и независимо управляющий его операциями, а также обеспечивающий различные сервисы для приложений, включая планирование и выполнение алгоритмов. Ресурс может быть создан, сконфигурирован, стартован, удален и т. д. без воздействия на другие ресурсы в устройстве. Функциями ресурса являются прием данных и событий через входные интерфейсы, обработка и выдача их через выходные интерфейсы.

Третьим стандартом, развивающим представление о функциональных блоках, является МЭК 61804. Он содержит спецификацию (детализацию)

требований к распределенным системам управления, построенным на основе функциональных блоков. МЭК 61804 конкретизирует абстрактные определения, данные в МЭК 61499. Он добавляет в МЭК 61499 описания параметров и функций, выполняемых функциональными блоками, которые могут быть реализованы в физических устройствах.

Стандарт определяет минимальный набор ФБ, который может быть необходим для индустриальных приложений. Набор состоит из двух частей: сложные ФБ (ПИД-регулятор, селектор для схем голосования, инкрементный сумматор, таймер, интегратор) и простые (вычисление тригонометрических функций, модуля, суммирования, усреднения, блоки арифметических операций, блоки Булевых функций и т. п.).

Одним из наиболее широко применяемых спецификаций стандарта МЭК 61804 является описание языка EDDL (Electronic Device Description Language), который является дальнейшим развитием методов генерации GSD файла в сетях Profibus и разрабатывался с поддержкой организации Fieldbus Foundation.

Описанию функциональных блоков для систем автоматизации зданий посвящен стандарт ISO 16484-3 .

Последовательные функциональные схемы, SFC

SFC называют языком программирования, хотя по сути это не язык, а вспомогательное средство для структурирования программ. Он предназначен специально для программирования последовательности выполнения действий системой управления, когда эти действия должны быть выполнены в заданные моменты времени или при наступлении некоторых событий. В его основе лежит представление системы управления с помощью понятий состояний и переходов между ними.

Язык SFC предназначен для описания системы управления на самом верхнем уровне абстракции, например, в терминах "Старт", "Наполнение автоклава", "Выполнение этапа №1", "Выполнение этапа №2", "Выгрузка из автоклава". Язык SFC может быть использован также для программирования отдельных функциональных блоков, если алгоритм их работы естественным образом описывается с помощью понятий состояний и переходов. Например, алгоритм автоматического соединения модема с коммутируемой линией описывается состояниями "Включение", "Обнаружение тона", "Набор номер", "Идентификация сигнала" и переходами "Если длинный - то ждать 20 сек", "Если короткий - перейти в состояние "Набор Номера"" и т.д.

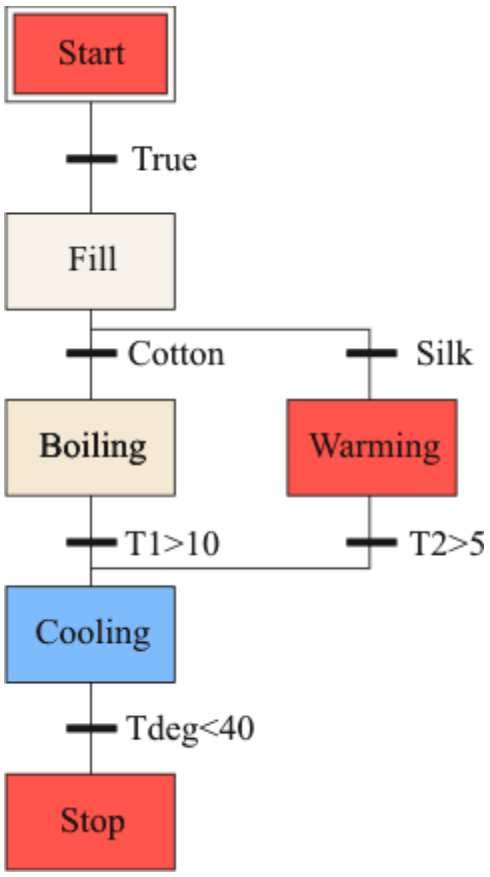


Рис. 4. Пример программы на языке SFC

На рис. 4 показан фрагмент программы на языке SFC. Программа состоит из шагов и условий переходов. Шаги показываются на схеме прямоугольниками, условия переходов - жирной перечеркивающей линией. Программа выполняется сверху вниз. Начальный шаг на схеме показывается в виде двойного прямоугольника. Условия переходов записываются рядом с их обозначениями. Каждый шаг программы может представлять собой реализацию сложного алгоритма, написанного на одном из МЭК-языков.

Программное обеспечение

Программирование ПЛК на описанных выше языках МЭК 61131-3 осуществляется с помощью специализированного программного обеспечения, которое разрабатывается производителями ПЛК или фирмами, специализирующимися на создании ПО для систем автоматизации. Наиболее известными в мире являются системы CoDeSys фирмы 3S и ISaGRAF фирмы ICS Triplex.

CoDeSys

CoDeSys (Co ntroller Development System) представляет собой комплекс программ для проектирования прикладного программного обеспечения, отладки в режиме эмуляции и загрузки программы в ПЛК. Основными

частями системы являются среда разработки программы и среда ее исполнения (CoDeSys SP), которая находится в ПЛК.

В CoDeSys входят графические и текстовые редакторы для всех пяти языков МЭК 61131-3. Этот комплекс полностью реализует требования стандарта и дополнительно вводит ряд оригинальных расширений, самым удобным из которых является объектно-ориентированные программирование. Однако расширениями языка можно не пользоваться, чтобы сохранить требования к совместимости языков, предъявляемое к открытым системам.

В одном проекте может быть использовано несколько контроллеров разных производителей. Каждый из них может программироваться как независимое устройство или с учетом их взаимодействия в промышленной сети. Проект состоит из нескольких приложений, распределенных по нескольким контроллерам. В одном ПЛК может существовать несколько независимых приложений.

Программа, написанная на языках МЭК, компилируется системой CoDeSys в машинный код, оптимизированный для заданной аппаратной платформы. Компилятор выдает диагностические сообщения как на этапе компиляции, так и на этапе ввода операторов языка.

Машинный код, сгенерированный компилятором CoDeSys, загружается в ПЛК, после чего разработчик имеет возможность использовать широкий набор функций для быстрой и эффективной отладки приложения. Текущие значения переменных видны непосредственно в редакторах программ. Программу можно выполнять по шагам или по контроллерным циклам. Можно задавать точки останова программы, просматривать стек вызовов, подготавливать связные наборы значений переменных и загружать их одной командой.

При отсутствии реального контроллера отладку программы можно выполнять с помощью встроенного программного эмулятора.

Система имеет также встроенный многоканальный программный трассировщик (графический самописец) значений переменных. Он позволяет наглядно представить динамически изменяющиеся данные проекта. Данные аккумулируются в памяти ПЛК и могут синхронизироваться с определенными событиями. Трассировщик полезен не только при отладке, но и при анализе нештатных ситуаций в процессе эксплуатации оборудования.

После изменения программы во время отладки перекомпилируются только измененные части программы. Их можно подгружать в контроллер без остановки выполнения прикладной программы. Эта возможность системы называется "горячим обновлением" кода.

Программируемое устройство соединяется с CoDeSys через вспомогательный программный компонент – шлюз связи, который использует

протокол TCP/IP. Шлюз работает на компьютере программиста или удаленно, например, через интернет или сеть Ethernet. Контроллер подключается компьютеру через любой последовательный канал или сеть. Добавив драйвер, изготовитель ПК может поддержать свой оригинальный протокол связи.

Общение ПЛК со SCADA осуществляется с помощью стандартного OPC сервера.

Для того, чтобы ПЛК можно было программировать с помощью CoDeSys, в контроллере должна быть установлена система исполнения. Установку системы выполняет изготовитель контроллера. Изготовитель обеспечивает также поддержку всех модулей ПЛК, поэтому конечный пользователь может сосредоточиться на разработке только прикладной программы.

Среда исполнения CoDeSys может функционировать в ПЛК под управлением различных операционных систем или вообще без них, в том числе на обычном персональном компьютере. Собственное ядро реального времени может устанавливать контроллерный цикл с точностью до нескольких микросекунд. Прикладная программа остается работоспособной даже при зависании ОС.

Помимо средств программирования, CoDeSys имеет встроенную систему визуализации, которая применяется для операторского управления, а также моделирования на этапе разработки. Визуализацию можно запустить на компьютере, графической панели ПЛК или встроенном в контроллер web-сервере.

Пользователь может самостоятельно расширять возможность CoDeSys путем создания библиотек программных модулей. Например, он может реализовать поддержку нестандартных интерфейсов.

Комплекс программирования CoDeSys построен по компонентной технологии Microsoft на базе автоматизации. Поэтому изготовитель ПЛК может включить в комплекс свои собственные компоненты, от конфигуратора оригинальной сети до собственного языка программирования ПЛК.

Для систем, связанных с безопасностью, CoDeSys имеет библиотеку функциональных блоков PLCopen Safety, систему исполнения для оборудования с дублированием и специализированное расширение среды программирования. При внезапном отключении питания CoDeSys автоматически сохраняет значения переменных во флеш-памяти или в ОЗУ с батарейным питанием.

ISaGRAF

Система ISaGRAF фирмы ICS Triplex также состоит из среды разработки и среды исполнения. Среда исполнения может функционировать практически

на любой операционной системе и любой аппаратной платформе, включая персональный компьютер. Среда разработки поддерживает все пять зыков МЭК 61131-3 и функциональные блоки МЭК 61499, имеет средства для редактирования, компиляции, документирования, управления библиотеками, архивирования, моделирования системы при отсутствии реального ПЛК и отладки с подключенным ПЛК.

Комплекс программ ISaGRAF первый на рынке использовал новый стандарт МЭК 61499 для программирования распределенных систем управления.

Связь между SCADA пакетом и контроллером, запрограммированным с помощью ISaGRAF, осуществляется с помощью стандартного OPC сервера.

Среда исполнения создается и загружается в контроллер производителем ПЛК и является независимой от исполняемой в ней программы пользователя.

Среда разработки имеет знакомый по Windows-приложениям интерфейс с подсказками, панелями инструментов, окнами, с функциями вставки и замены и т. п. Код, полученный на выходе среды разработки, может исполняться на любой аппаратно-программной платформе без изменений, если на ней предварительно установлена среда исполнения. Среда разработки может также транслировать пользовательскую программу, написанную на МЭК-языках, в текст на языке Си.

Контрольные вопросы по теме

Уровень курса

1. Системы программирования на языках МЭК 61131-3.
2. Языки МЭК 61131-3: язык релейно-контактных схем, LD.
3. Языки МЭК 61131-3: список инструкций IL и структурированный текст ST.
4. Языки МЭК 61131-3: диаграммы функциональных блоков FBD.
5. Языки МЭК 61131-3: последовательные функциональные схемы SFC.

Лекции № 17 ***Тема:*** SCADA-системы

Оглавление

Источники:

1. Таненбаум Э., Остин Т. Архитектура компьютера. 6-е изд. — СПб.: Питер, 2013. — 816 с.: ил.

   https://rutracker.org/forum/viewtopic.php?t=4956359

2. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. — СПб.: Питер, 2015. — 1120 с.: ил.
3. Бабак В.П. Теоретические основы информационно-измерительных систем: Учебник / В.П. Бабак, С.В. Бабак, В.С. Ерёменко и др. – К., 2014. – 832 с.

Пользовательский интерфейс, SCADA-пакеты

Большинство систем автоматизации функционирует с участием человека (оператора, диспетчера). Интерфейс между человеком и системой называют человеко-машинным интерфейсом (ЧМИ), в зарубежной литературе - HMI (Human-Machinery Interface) или MMI (Man-Machinery Interface). В частном случае, когда ЧМИ предназначен для взаимодействия человека с автоматизированным технологическим процессом, его называют SCADA-системой (Supervisory Control And Data Acquisition). Этот термин переводится буквально как "диспетчерское управление и сбор данных", но на практике его трактуют гораздо шире, а современные SCADA-пакеты включают в себя широчайший набор функциональных возможностей, далеко выходящий за рамки сбора данных и диспетчерского управления.

Рис. 1 Пульт оператора технологического процесса

Функции SCADA

Существующие в настоящее время SCADA-пакеты выполняют множество функций, которые можно разделить на несколько групп:

- настройка SCADA на конкретную задачу (т. е. разработка программной части системы автоматизации);
- диспетчерское управление;
- автоматическое управление;
- хранение истории процессов;
- выполнение функций безопасности;
- выполнение общесистемных функций.

Несмотря на множество функций, выполняемых SCADA, основным ее отличительным признаком является наличие интерфейса с пользователем. При отсутствии такого интерфейса перечисленные выше функции совпадают с функциями средств программирования контроллеров, а управление является автоматическим, в противоположность диспетчерскому.

Качество решений, принятых оператором (диспетчером), часто влияет не только на качество производимой продукции, но и на жизнь людей. Поэтому комфорт рабочего места, понятность интерфейса, наличие подсказок и блокировка явных ошибок оператора являются наиболее важными свойствами SCADA, а дальнейшее их развитие осуществляется в направлении улучшения эргономики и создания экспертных подсистем.

Иногда SCADA комплектуются средствами для программирования контроллеров, однако эта функция вызвана коммерческими соображениями и слабо связана с основным назначением SCADA.

В SCADA-пакетах используют понятие *аларма* и *события*. Событие - это изменение некоторых состояний в системе. Примерами событий могут быть включение перевалки зерна в элеваторе, завершение цикла периодического процесса обработки детали, окончание загрузки бункера, регистрация нового оператора и т. п. События не требуют срочного вмешательства оператора, а просто информируют его о состоянии системы.

В отличие от события, аларм (от английского "alarm" - "сигнал тревоги") представляет собой предупреждение о важном событии, в ответ на которое нужно срочно предпринять некоторые действия. У английского слова "аларм" имеется точный русский перевод - "сигнал тревоги" или "аварийный сигнал", однако термин "аларм" уже прочно вошел в лексикон промышленной автоматизации.

Примерами алармов может быть достижение критической температуры хранения зерна в элеваторе, после которого начинается его возгорание, достижение критического значение давления в автоклаве, после которого возможен разрыв оболочки, срабатывание датчика открытия охраняемой двери, превышение допустимого уровня загазованности в котельной и т.п.

В связи с тем, что алармы требует принятия решения, их делят на подтвержденные и неподтвержденные. Подтвержденным называется аларм, в ответ на который оператор ввел команду подтверждения. До этого момента аларм считается неподтвержденным.

Алармы делятся на дискретные и аналоговые. Дискретные сигнализируют об изменении дискретной переменной, аналоговые алармы появляются, когда непрерывная переменная $y(t)$ входит в заранее заданный интервал своих значений. В качестве примера на рис. 5 показано деление всего интервала изменения переменной $y(t)$ на интервалы "Норма", "Внимание" (предаварийное состояние) и "Авария":

- аларм "Внимание" возникает при $y(a) < y(t) < y(b)$ во время нарастания наблюдаемой переменной и при $y(d) < y(t) < y(c)$ во время ее уменьшения;

- аларм "Авария" возникает при $y(b) < y(t)$.

Каждая критическая граница на рис. 5 имеет зону нечувствительности (мертвую зону), которая нужна для того, чтобы после снятия состояния аларма переменная не могла вернуться в него вследствие случайных выбросов в системе (шумов). Границы зон на рис. 5 могут изменяться с течением времени.

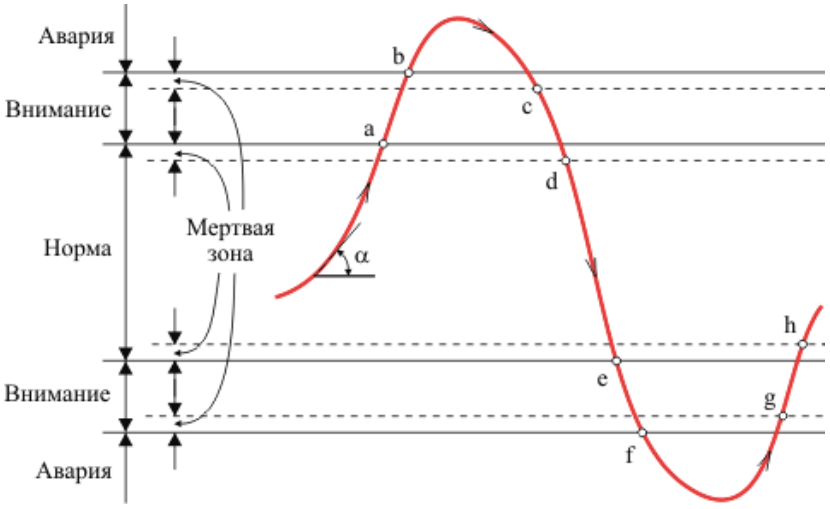


Рис. 2. Пример назначения интервалов аналоговым алармам

Аналогичные границы могут быть назначены для скорости изменения переменной (для производной функции $y(t)$), которая определяется как угол α наклона касательной к кривой $y(t)$.

Методика выдачи алармов должна быть надежной. В частности, всплывающие окна с сообщениями алармов должны быть всегда поверх остальных окон, алармы могут дублироваться звуком и светом. Поскольку алармов в системе может быть много, им назначают разные приоритеты, разные громкости и тоны звукового сигнала и т. п.

Разработка человеко-машинного интерфейса

Одной из основных функций SCADA является разработка человеко-машинного интерфейса, т.е. SCADA одновременно является и ЧМИ, и инструментом для его создания. Быстрота разработки существенно влияет на рентабельность фирмы, выполняющей работу по внедрению системы автоматизации, поэтому скорость разработки является основным показателем качества SCADA с точки зрения системного интегратора. В процесс разработки входят следующие операции:

- создание графического интерфейса (мнемосхем, графиков, таблиц, всплывающих окон, элементов для ввода команд оператора и т д.);

- программирование и отладка алгоритмов работы системы автоматизации. Многие SCADA позволяют выполнять отладку системы как в режиме эмуляции оборудования, так и с подключенным оборудованием;
- настройка системы коммуникации (сетей, модемов, коммуникационные контроллеров и т п.);
- создание баз данных и подключение к ним SCADA.

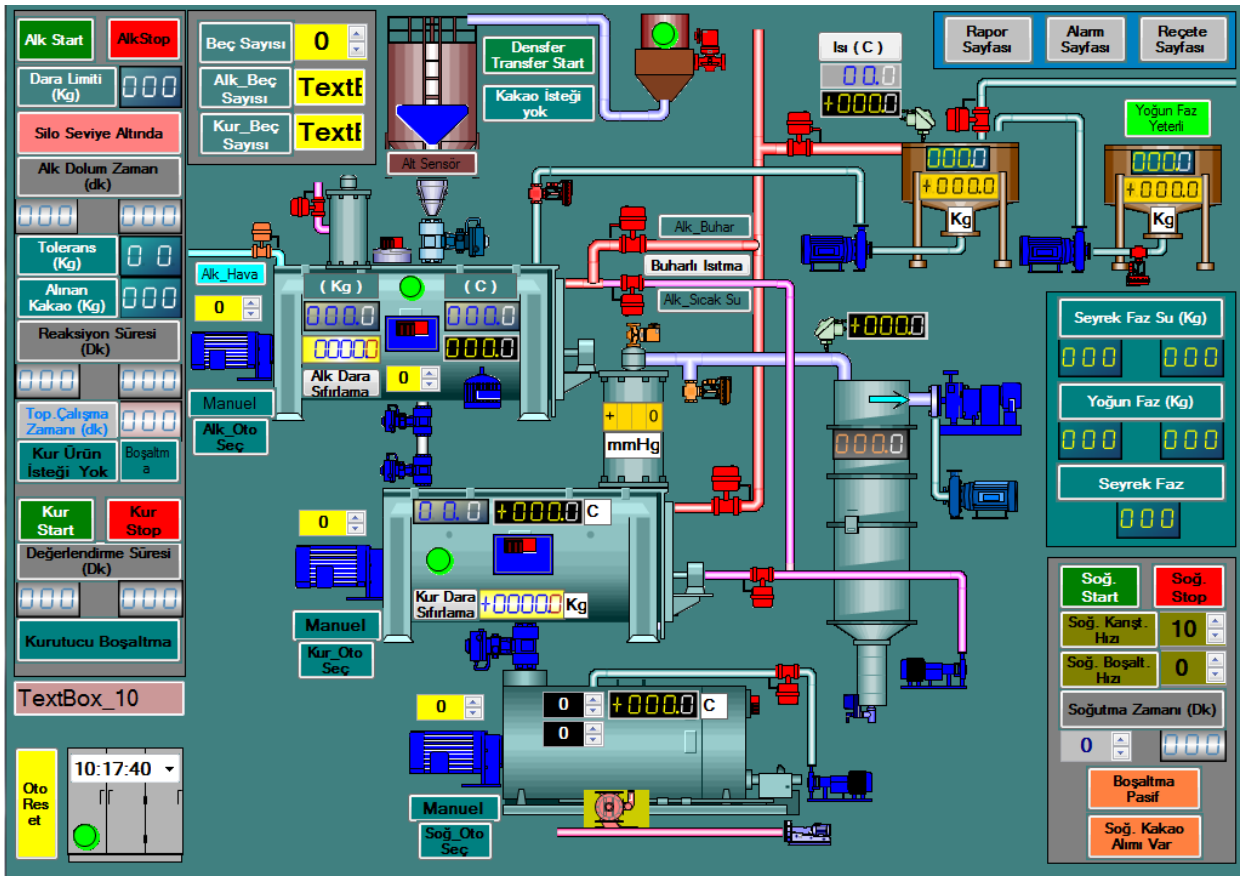


Рис. 3 Пример экрана пульта оператора в SCADA-системе

SCADA как система диспетчерского управления

Как система диспетчерского управления SCADA может выполнять следующие задачи:

- взаимодействие с оператором (выдача визуальной и слуховой информации, передача в систему команд оператора);
- помощь оператору в принятии решений (функции экспертной системы);
- автоматическая сигнализация об авариях и критических ситуациях;
- выдача информационных сообщений на пульт оператора;
- ведение журнала событий в системе;

– извлечение информации из архива и представление ее оператору в удобном для восприятия виде;
– подготовка отчетов (например, распечатка таблицы температур, графиков смены операторов, перечня действий оператора);
– учет наработки технологического оборудования.

SCADA как часть системы автоматического управления

Основная часть задач автоматического управления выполняется, как правило, с помощью ПЛК, однако часть задач может возлагаться на SCADA. Кроме того, во многих небольших системах управления ПЛК могут вообще отсутствовать и тогда компьютер с установленной SCADA является единственным средством управления. SCADA обычно выполняет следующие задачи автоматического управления:

- автоматическое регулирование;
- управление последовательностью операций в системе автоматизации;
- адаптация к изменению условий протекания технологического процесса;
- автоматическая блокировка исполнительных устройств при выполнении заранее заданных условий.

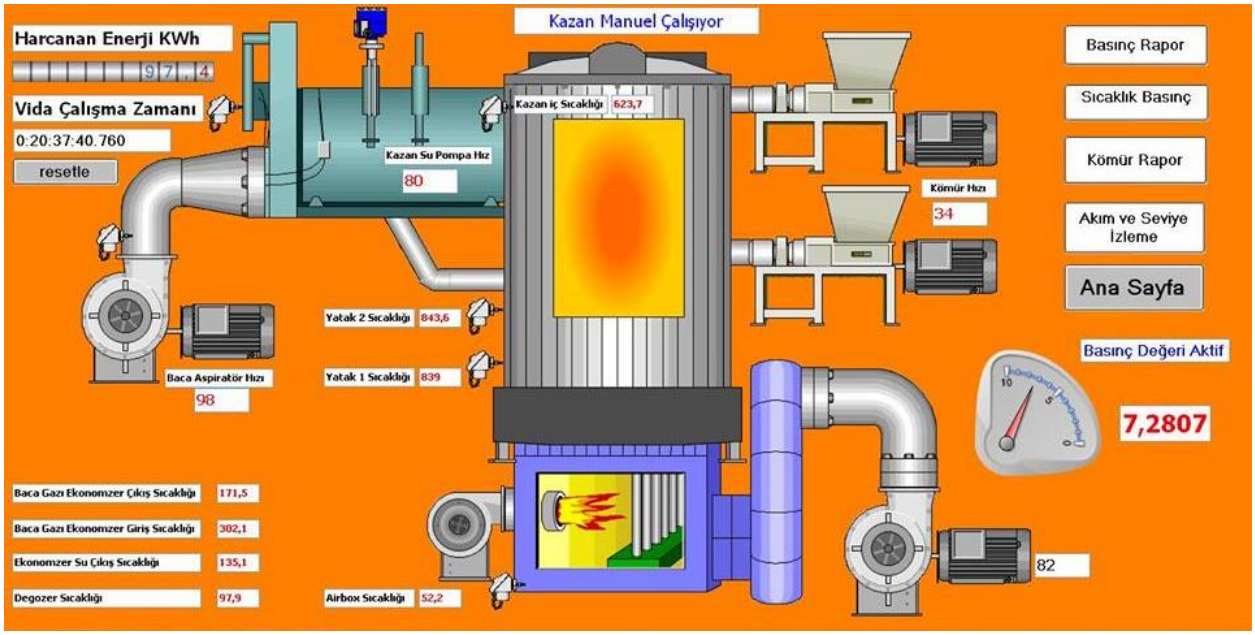


Рис. 4 Пример экрана пульта оператора в SCADA-системе

Хранение истории процесса

Знание предыстории управляемого процесса позволяет улучшить будущее поведение системы, проанализировать причины возникновения опасных ситуаций или брака продукции, выявить ошибки оператора. Для создания истории система выполняет следующие операции:

- сбор данных и их обработка (цифровая фильтрация, интерполяция, сжатие, нормализация, масштабирование и т. д.);
- архивирование данных (действий оператора, собранных и обработанных данных, событий, алармов, графиков, экранных форм, файлов конфигурации, отчетов и т. п.);
- управление базами данных (реального времени и архивных).

Безопасность SCADA

Применение SCADA в системах удаленного доступа через интернет резко повысило уязвимость SCADA к действиям враждебных лиц. Пренебрежение этой проблемой может приводить, например, к отказу в работе сетей электроснабжения, жизнеобеспечения, связи, отказу морских маяков, дорожных светофоров, к заражению воды неочищенными стоками и т.п. Возможны и более тяжелые последствия с человеческими жертвами или большим экономическим ущербом. Для повышения безопасности SCADA используют следующие методы:

- разграничение доступа к системе между разными категориями пользователей (у сменного оператора, технолога, программиста и директора должны быть разные права доступа к информации и к модификации настроек системы);
- защиту информации (путем шифрования информации и обеспечения секретности протоколов связи);
- обеспечение безопасности оператора благодаря его отдалению от опасного управляемого процесса (дистанционное управление). Дистанционный контроль и дистанционное управление являются типовыми требованиями и выполняются по проводной сети, радиоканалу (через GSM- или радиомодем), через интернет и т.д.;
- специальные методы защиты от кибератак;
- применение межсетевых экранов.

Общесистемные функции

Поскольку SCADA обычно является единственной программой для управления системой автоматизации, на нее могут возлагаться также некоторые общесистемные функции:

- осуществление взаимодействий между несколькими SCADA, между SCADA и другими программами (MS Office, базой данных, MATLAB и т.п.);
- диагностика аппаратуры, каналов связи и программного обеспечения.

Свойства SCADA

Анализ свойств различных SCADA позволяет выбирать систему, оптимальную для решения поставленной задачи. Все многообразие свойств SCADA-пакетов можно разбить на следующие группы:

- инструментальные свойства;
- эксплуатационные свойства;
- свойства открытости;
- экономическая эффективность.

Инструментальные свойства

К инструментальным относятся свойства SCADA, влияющие на эффективность работы системных интеграторов:

- быстрота разработки проекта;
- легкость освоения;
- поддерживаемые средства коммуникации;
- наличие функций для сложной обработки данных;
- наличие языков МЭК 61131-3 и универсального алгоритмического языка типа Visual Basic;
- степень открытости для разработчика (поддержка COM и ActiveX для подключения программных модулей пользователя, а также OPC, ODBC, OLE DB;
- качество технической документации (полнота, ясность изложения, количество ошибок);
- наличие режима эмуляции оборудования для отладки;
- наличие внутренних графических редакторов, позволяющих отказаться от применения внешних редакторов типа CorelDraw или Photoshop; поддержка типовых графических форматов файлов;
- качество технической поддержки (время реакции на вопросы пользователей, наличие "горячей линии" технической поддержки.

SCADA используют языки программирования МЭК 61131-3, ориентированные на технологов, которые дополняются функциями, специфическими для SCADA. Большинство SCADA имеют встроенный редактор и интерпретатор языка Visual Basic фирмы Microsoft.

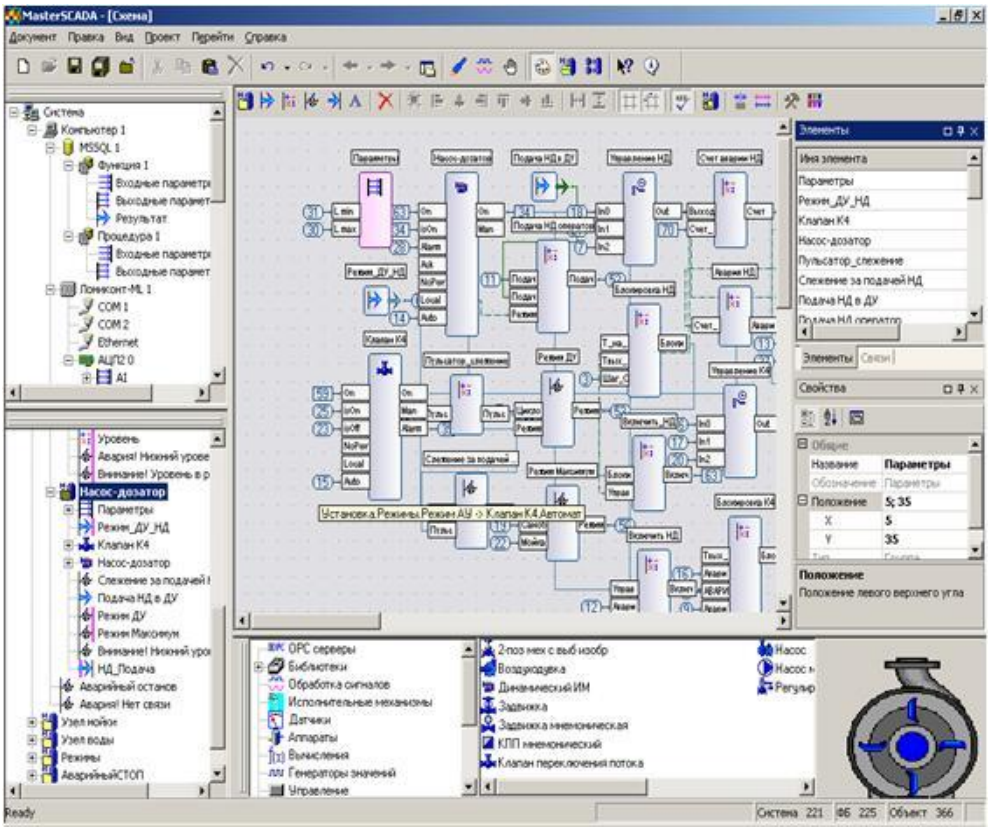


Рис. 5 Среда разработки в SCADA-системе

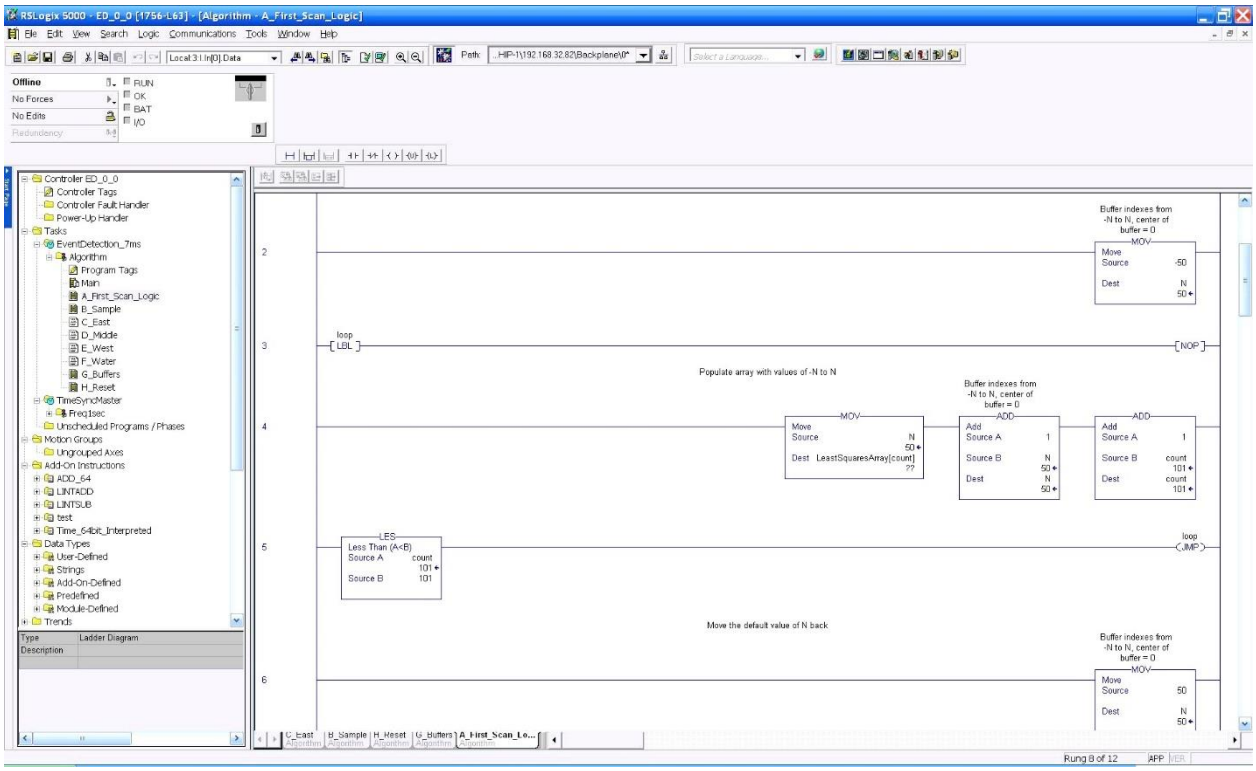


Рис. 6 Программирование ПЛК в SCSDA-системе

Эксплуатационные свойства

Качество SCADA в процессе эксплуатации оценивается конечными пользователями и характеризуется следующим набором свойств:

- робастность (нечувствительность к ошибкам пользователя, защищенность от вандалов и враждебных элементов, устойчивость к ошибкам в исходных данных);
- надежность;
- информационная защищенность;
- наличие средств сохранения данных при нештатных ситуациях, отключениях питания и сбоях;
- наличие автомата перезапуска системы при ее зависании или после прерывания питания;
- поддержка резервирования SCADA (операторской станции, сетевых серверов, клиентских рабочих станций, резервное копирование данных);
- поддержка переключения экранов с разной детализацией изображений; поддержка нескольких мониторов.

Степень открытости

Степень открытости очень сильно влияет на экономическую эффективность системы, однако это влияние носит случайный характер, поскольку зависит от степени использования свойств открытости в конкретном проекте.

Открытость для программирования пользователем SCADA обеспечивается возможностью подключения программных модулей, написанных пользователем или другими производителями. Это обычно достигается тем, что SCADA разрабатывается как контейнер для COM-объектов и ActiveX элементов. Совместимость с аппаратурой и базами данных других производителей достигается с помощью стандарта OPC, применением интерфейса ODBC или OLE DB. Открытость системы программирования достигается поддержкой языков МЭК 61131-3.

Особенно интересно с точки зрения открытости применение веб-интерфейса, поскольку он обеспечивает доступ к SCADA с любого компьютера из любой точки мира, независимо от аппаратной платформы, типа канала связи, операционной системы и используемого веб-навигатора.

Экономическая эффективность

Экономическую эффективность SCADA можно определить как отношение экономического эффекта от ее внедрения к общей сумме затрат на внедрение и поддержание системы в работоспособном состоянии. На экономическую эффективность в конечном счете влияют практически все свойства SCADA, однако в первую очередь можно выделить следующие:

- масштабируемость (возможность применения как для больших, так и для малых систем);
- модульность. Модульность позволяет сделать заказную комплектацию системы в зависимости от поставленной задачи. Типовыми модулями могут быть, например, модуль ввода-вывода, модуль визуализации, модуль алармов, модуль трендов, модуль отчетов, модуль коммерческого учета энергоресурсов и др.;
- стоимость обслуживания;
- условия обновления версий;
- надежность поставщика, наличие опыта практического применения;
- стоимость обучения;
- стоимость технической поддержки;
- методы ценообразования.

Общим недостатком универсальных SCADA является их низкая экономическая эффективность при использовании для решения простых задач. Несмотря на то, что цена SCADA-пакетов существенно снижается при уменьшении количества доступных пользователю тегов и набора модулей, остается высокой цена технической поддержки. Также дорогой (трудоемкой) остается адаптация универсальной SCADA к конкретной задаче. Поэтому ряд фирм предлагают более узкоспециализированные, но достаточно простые в настройке микро-SCADA с сокращенной функциональностью, например, пакет RLDataView.

Программное обеспечение

Ниже мы рассмотрим отличительные особенности двух известных пакетов: MasterSCADA и Trace Mode.

MasterSCADA

Система MasterSCADA предназначена для создания полномасштабных систем автоматизации в различных отраслях промышленности. Основной ее особенностью является объектный подход, использованный на уровне описания системы при ее настройке на конкретный объект автоматизации. Например, цех, участок, технологический блок и физическое устройство при

создании проекта с помощью MasterSCADA рассматриваются как отдельные объекты. Для каждого объекта создается свое описание на технологическом языке программирования. Описание включает в себя свойства объекта и документы объекта. Свойствами могут быть период опроса, способ линеаризации датчика, диапазон входных сигналов. Документами объекта являются его изображение, мнемосхема, график изменения переменных и т. п. Любой документ в системе относится к некоторому объекту. Такой подход позволяет легко размножать один раз созданные объекты, что повышает скорость настройки SCADA на задачу пользователя.

К признакам объектного подхода относится также возможность наследования всех настроек от "родительских" объектов. Это означает, что в MasterSCADA нет необходимости вводить настройки для каждого типа объектов "с нуля". Можно использовать наследование этих настроек от родительского объекта, изменив в них только те параметры, которые отличают родителя от потомка.

Созданные объекты можно копировать с целью многократного использования. При копировании объекта сохраняются все связанные с ним документы и свойства. Связи с внешними источниками и приемниками данных восстанавливаются после копирования, если в системе имеются такие источники или свободные приемники данных (физические устройства). Это позволяет пополнять библиотеку объектов вновь созданными экземплярами и использовать объекты, созданные другими разработчиками.

Trace Mode

SCADA-система Trace Mode 6 фирмы AdAstrA состоит из инструментальной системы и набора исполнительных модулей. В состав Trace Mode 6 входят также средства управления бизнес-процессами производственного предприятия.

Для увеличения скорости разработки проекта пользователя применяется оригинальная технология автопостроения. Автоматически в SCADA могут быть построены:

- ✓ источники данных ПЛК и модулей ввода-вывода по известной конфигурации;
- ✓ каналы по источникам данных;
- ✓ связи каналов из редактора аргументов;
- ✓ связи контроллер-сервер и сервер-сервер;
- ✓ SQL-запросы;
- ✓ связи с OPC-сервером;
- ✓ связь с ODBC.

Автопостроение позволяет снизить количество ошибок, допускаемых пользователем при ручном создании проекта.

В пятой версии Trace Mode инструментальная система представлена в виде отдельных компонентов, в 6-ой использована интегрированная среда разработки.

В систему Trace M ode 6 включены пять языков программирования – Techno SFC, Techno LD, Techno FBD, Techno ST, и Techno IL, которые являются расширениями соответствующих языков стандарта МЭК 61131-3.

Заключение к главе "Программное обеспечение"

Основными тенденциями развития программного обеспечения для средств автоматизации являются максимальное упрощение процесса программирования и обеспечение открытости инструментальных средств. Конечной целью является предоставление потребителю возможности построения качественной системы автоматизации в максимально короткий срок.

Долгий период неопределенности в средствах программирования ПЛК и SCADA пакетов завершился принятием общепризнанного стандарта МЭК 61131-3 и созданием на его основе инструментальных средств программирования, которые поддерживаются фирмами, специализирующимися на программном обеспечении.

Существенный вклад в открытость систем автоматизации внес стандарт OPC, обеспечивший системным интеграторам широчайший выбор аппаратного обеспечения, совместимого с любыми стандартными SCADA пакетами, а разработчикам контроллерного оборудования - расширение рынков сбыта.

Контрольные вопросы по теме

Уровень курса

1. Функции SCADA.
2. Разработка человеко-машинного интерфейса.
3. SCADA как система диспетчерского управления.
4. SCADA как часть системы автоматического управления.
5. Свойства SCADA.