Встроенные системы

Лекция 1

Список литературы (в библиотеке)

1. Мануковский А.В. Микропроцессорные средства и системы управления Алматы : New book, 2020.
2. Хартов В.Я. Микропроцессорные системы в. - 2-е изд., испр. и доп. - Москва : Академия, 2014.
3. Кравченко, А. В. 10 практических устройств на AVR-микроконтроллерах Киев : МК-Пресс ; Санкт-Петербург : КОРОНА-ВЕК, 2013.
4. Пош М. Программирование встроенных систем на C++17 : создание универсальных и надежных встроенных решений для микроконтроллеров и операционных систем реального времени на современной версии языка программирования C++ / М. Пош; перевод с английского А.В. Снастина. - Москва : ДМК Пресс, 2020. - 393с.
5. Юсупова Г.М. Микропроцессоры и микропроцессорные системы Алматы : Альманахъ, 2019. - 125с.
6. Тулегулов А.Д. Робототехника и программирование на платформе Arduino : учебное пособие для технических специальностей вузов / А.Д. Тулегулов, А.О. Тлеубаева. - Алматы : Лантар Трейд, 2019. - 114с.
7. Зиатдинов С.И. Схемотехника телекоммуникационных устройств : учебник для студентов учреждений высшего профессионального образования, обучающихся по направлению подготовки 21070 «Инфокоммуникационные технологии и системы связи» / С.И. Зиатдинов. - Москва : Академия, 2013. - 365с.
8. Тапалов, Т. LabVIEW ортасында графикалық программалау : оқу құралы / Т. Тапалов ; Қазақстан Республикасының Білім және ғылым министрлігі. – Қарағанды : Medet Group, 2014. – 186 б
9. Bindal A. Electronics for embedded systems / A. Bindal. - Cham : Springer International Publishing, 2017. - XIII, 298 c.
10. Kharate G.K. Digital electronics / G.K. Kharate. - 7th impr. - New Delhi : Oxford University Press, 2013. - XII, 667 c.
11. Hami A.E. Embedded mechatronic systems. Vol. 2 : Analysis of failures, modeling, simulation and optimization / A.E. Hami, P. Pougnet. - London : ISTE Press ; Oxford : Elsevier, 2015. - XVII, 246 c

Список литературы (ссылки в интернете)

1. А.О. Ключев, П.В. Кустарев, Д.Р. Ковязина, Е.В. Петров Программное обеспечение встроенных вычислительных систем - Санкт-Петербург:2009. - 212 с. https://books.ifmo.ru/file/pdf/499.pdf
2. А.О. Ключев, Д.Р. Ковязина, П.В. Кустарев, А.Е. Платунов Аппаратные и программные средства встраиваемых систем - Санкт-Петербург: , 2010. - 290 с. https://books.ifmo.ru/file/pdf/686.pdf
3. Платт Ч. Электроника для начинающих. СПб.: БХВ-Петербург, 2017. – 416 с. : https://fileskachat.com/getfile/50260_725d31f0fea5d547899476ef4bbdd69f
4. Белов, М. П. Технические средства автоматизации и управления [Электронный ресурс] : учебное пособие / Михаил Петрович Белов ; Федеральное агентство по образованию, Государственное образовательное учреждение высшего профессионального образования, Северо-западный государственный заочный технический университет. – Санкт-Петербург : СЗТУ, 2006. – 181 с. https://library.enu.kz/enulib-web/public/portal/book/view/113076
5. Руководство по решениям в автоматизации [Электронный ресурс] : практические аспекты систем управления технологическими процессами.– 320 с. https://library.enu.kz/enulib-web/public/portal/book/view/118963
6. Бурукина И.П. Операционные системы реального времени: Учебное пособие. - Пенза: ПГУ, 2011. - 73 с. http://window.edu.ru/resource/985/74985/files/burukina.pdf
7. Уроки и статьи по Arduino, Raspberry Pi и созданию электроники своими руками: https://arduinoplus.ru

История встраиваемой системы

- В 1960 году встроенная система впервые была использована для разработки системы наведения Аполлона Чарльзом Старком Дрейпером в Массачусетском технологическом институте.
- В 1965 году Autonetics разработала D-17B, компьютер, используемый в системе наведения ракет Minuteman.
- В 1968 году была выпущена первая встроенная система для автомобиля.
- Texas Instruments разработала первый микроконтроллер в 1971 году.
- В 1987 году Wind River выпустила первую встроенную ОС VxWorks.
- Windows, встроенный в Microsoft CE в 1996 году.
- К концу 1990-х годов появилась первая встроенная система Linux.
- Аналитики прогнозируют, что к 2030 году рынок встраиваемых систем превысит 40 миллиардов долларов.

Встроенная система

Система — это система, в которой все ее узлы работают вместе в соответствии с набором правил.

Встраиваемые системы — это специализированные системы управления, которые разработаны таким образом, что такие системы будут работать, будучи встроенными непосредственно в устройство, которым они управляют.

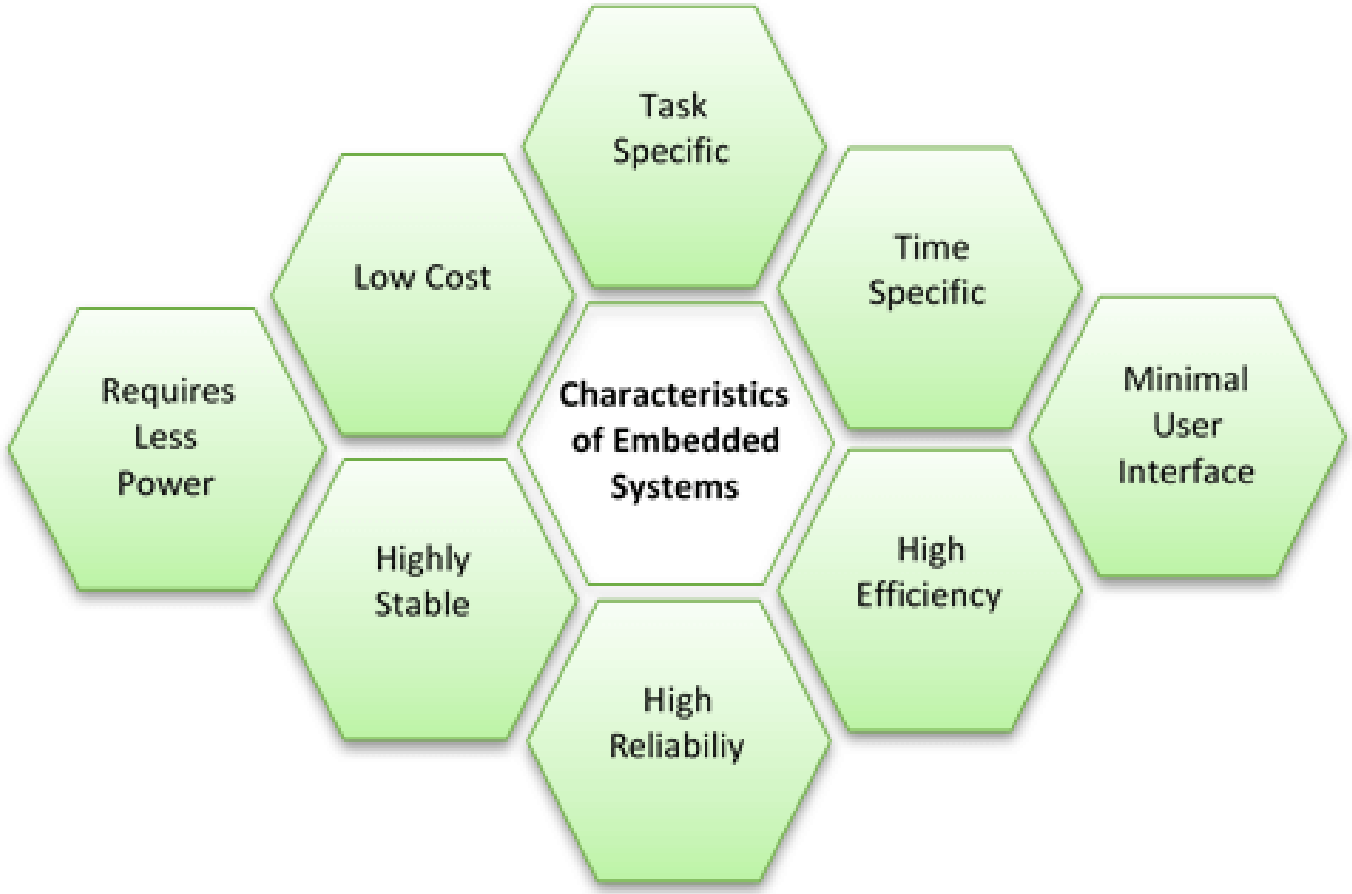
Встроенная система

Встроенная система состоит из трех компонентов −

- **аппаратное обеспечение**.
- **прикладное программное обеспечение**.
- **операционная система реального времени** (RTOS), которая контролирует прикладное программное обеспечение и предоставляет механизм, позволяющий процессору запускать процесс в соответствии с расписанием, следуя плану контроля задержек. RTOS определяет способ работы системы. Он устанавливает правила во время выполнения прикладной программы. Встраиваемая система небольшого масштаба может не иметь ОСРВ.

Таким образом, мы можем определить встраиваемую систему как основанную на микроконтроллере, программно управляемую, надежную систему управления в реальном времени.

Характеристики встроенной системы



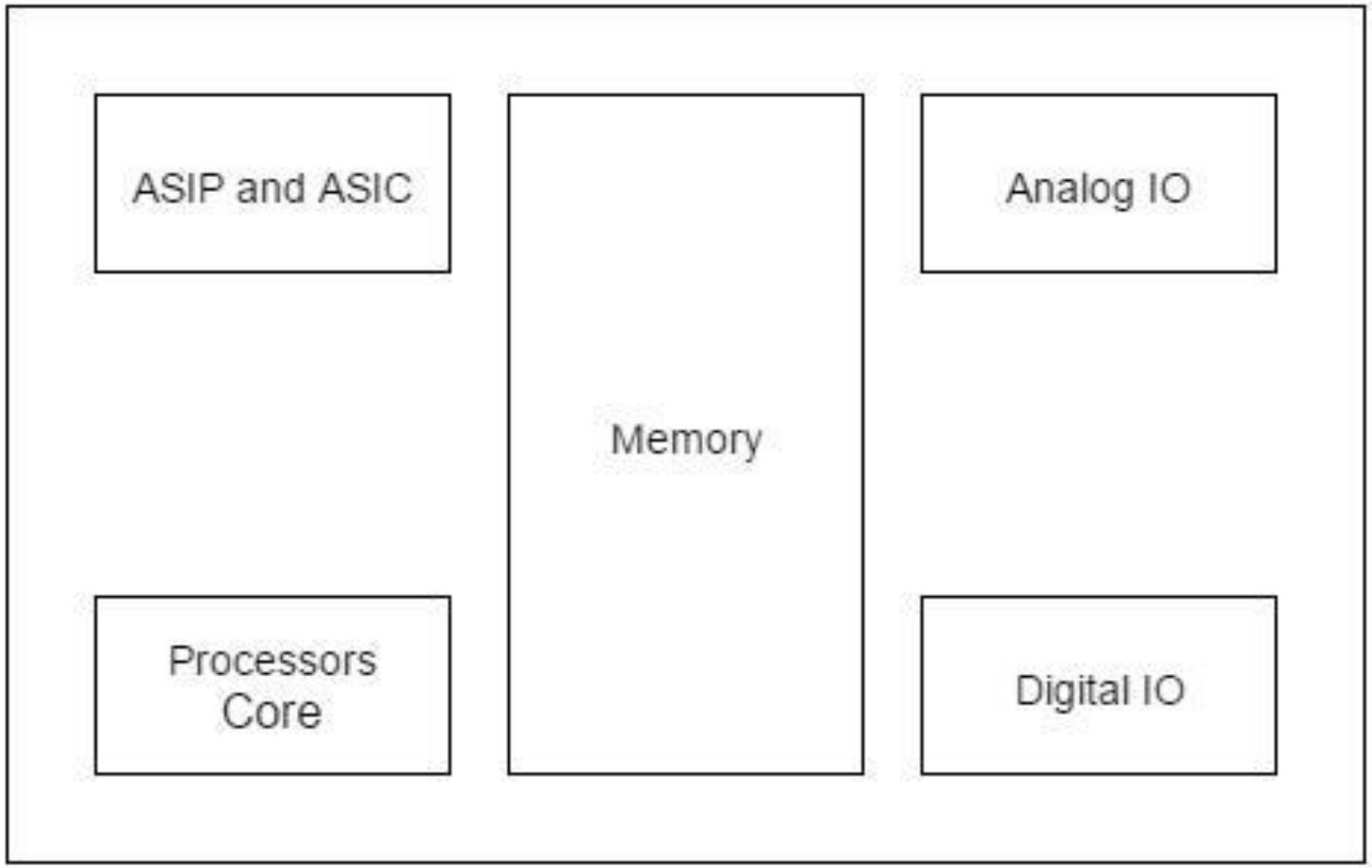
Характеристики встроенной системы

- **Однофункциональная** – встроенная система обычно выполняет специализированную операцию и делает то же самое неоднократно. Например: пейджер всегда функционирует как пейджер.
- **Жесткие ограничения** – все вычислительные системы имеют ограничения на показатели проектирования, но во встроенной системе они могут быть особенно жесткими. Показатели проектирования - это показатель характеристик реализации, таких как ее стоимость, размер, мощность и производительность. Он должен быть такого размера, чтобы поместиться на одном чипе, должен работать достаточно быстро, чтобы обрабатывать данные в режиме реального времени, и потреблять минимум энергии для продления срока службы батареи.
- **Реагирование и в режиме реального времени** – Многие встроенные системы должны постоянно реагировать на изменения в среде системы и должны вычислять определенные результаты в режиме реального времени без каких-либо задержек. Рассмотрим пример автомобильного круиз-контроллера; он постоянно отслеживает и реагирует на датчики скорости и тормоза. Он должен многократно вычислять ускорение или снижение ускорения в течение ограниченного времени; задержка вычисления может привести к потере контроля над автомобилем.

Характеристики встроенной системы

- **На основе микропроцессоров** – Это должен быть микропроцессор или микроконтроллер.
- **Память** – у нее должна быть память, поскольку ее программное обеспечение обычно встраивается в ПЗУ. Для этого не требуется никаких дополнительных запоминающих устройств в компьютере.
- **Подключение** – для подключения устройств ввода и вывода должны быть подключены периферийные устройства.
- **Системы HW-SW** - Программное обеспечение используется для расширения возможностей и гибкости. Аппаратное обеспечение используется для обеспечения производительности и безопасности.

Характеристики встроенной системы



Важные термины, используемые во встроенной системе

Надежность:

Это мера вероятности выживания системы, когда функция является критической во время выполнения.

Отказоустойчивость:

Отказоустойчивость – это способность компьютерной системы выживать при наличии неисправностей.

В режиме реального времени:

Встроенная система должна соответствовать различным временным и другим ограничениям. Они навязаны ему естественным поведением внешнего мира в реальном времени.

Например, военно-воздушный департамент, который отслеживает поступающие ракетные атаки, должен точно рассчитывать и планировать свои контратаки из-за жестких сроков в реальном времени. В противном случае он будет уничтожен.

Важные термины, используемые во встроенной системе

Гибкость:

Это сборка систем со встроенными возможностями отладки, которая позволяет осуществлять удаленное обслуживание.

Например, вы строите космический корабль, который приземлится на другой сеялке для сбора различных типов данных и отправки собранных нам деталей. Если этот космический корабль сошел с ума и потерял управление, мы сможем провести важную диагностику. Таким образом, гибкость жизненно важна при разработке встроенной системы.

Переносимость:

Портативность – это мера простоты использования одного и того же встроенного программного обеспечения в различных средах. Это требует обобщенных абстракций между самой логикой прикладной программы и низкоуровневыми системными интерфейсами.

Важные термины, используемые во встроенной системе

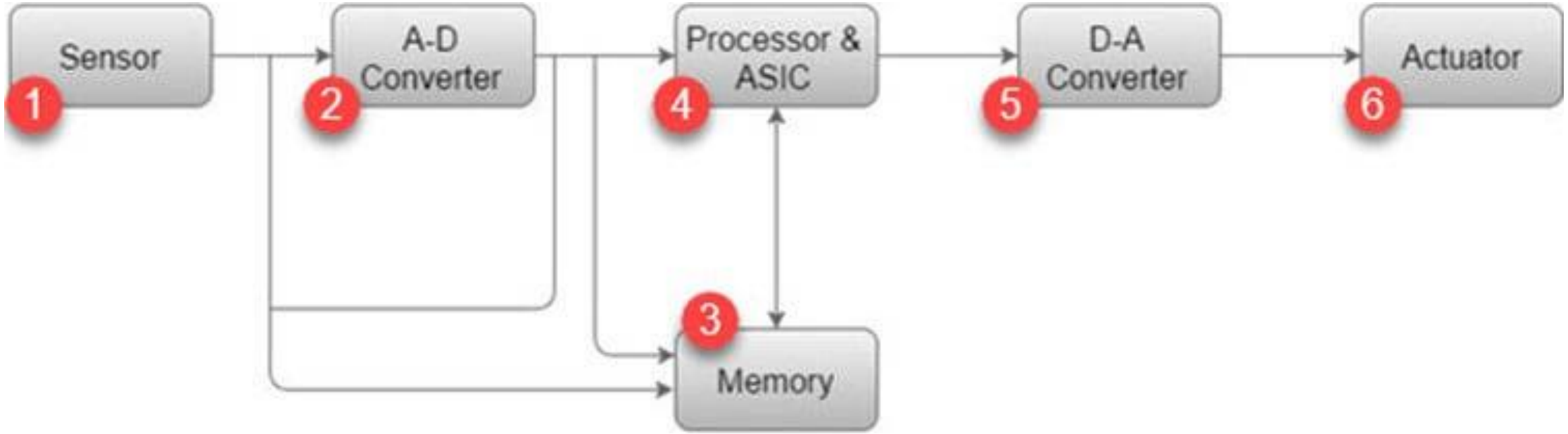
Что такое микроконтроллер?

Микроконтроллер – это одночиповый модуль СБИС, который также называют микрокомпьютером. Он содержит всю необходимую память и интерфейсы ввода / вывода, в то время как микропроцессору общего назначения нужны дополнительные микросхемы, предлагаемые этими необходимыми функциями. Микроконтроллеры широко используются во встроенных системах для приложений управления в реальном времени.

Что такое микропроцессор?

Микропроцессор представляет собой однокристальное полупроводниковое устройство. Его центральный процессор содержит программный счетчик, ALU указатель стека, рабочий регистр, схему синхронизации. Он также включает в себя ПЗУ и ОЗУ, декодер памяти и множество последовательных и параллельных портов.

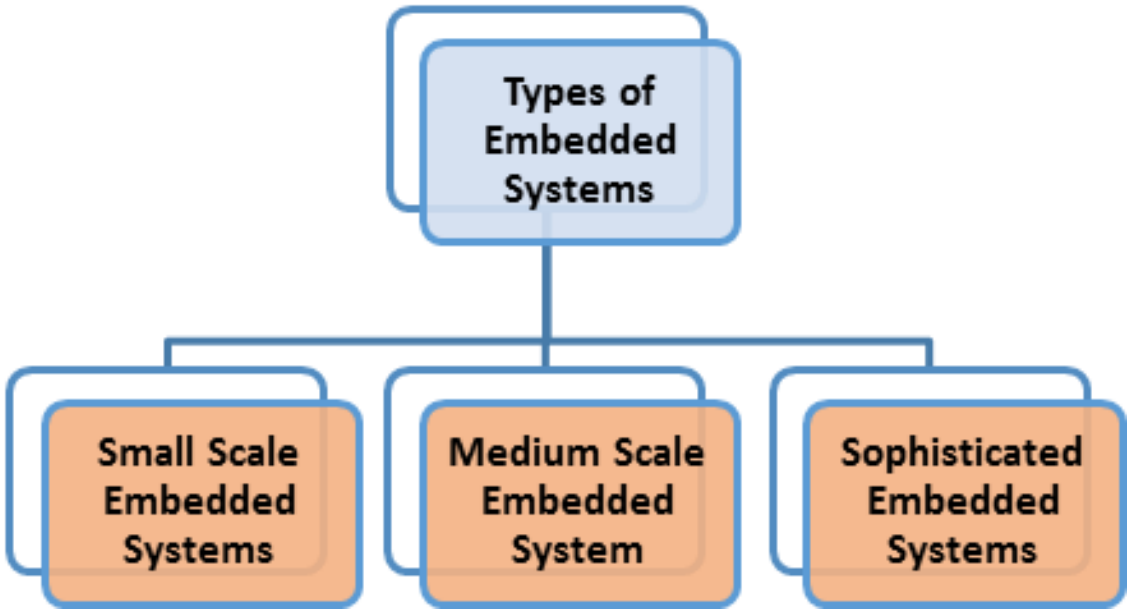
Архитектура встроенной системы



Типы встраиваемых систем

Три типа встраиваемых систем:

- Малый масштаб
- Средний масштаб
- Сложные



Маломасштабные встраиваемые системы

Эта встроенная система может быть разработана с одним 8 или 16-битным микроконтроллером. Им можно управлять с помощью питания (батареи). Для разработки небольшой встроенной системы наиболее важными инструментами программирования являются редактор, ассемблер (IDE) и кросс-ассемблер.

Встраиваемые системы среднего масштаба

Эти типы встроенных систем разработаны с использованием 16- или 32-разрядных микроконтроллеров. Эти системы предлагают как аппаратные, так и программные сложности. C, C ++, Java, инструмент разработки исходного кода и т.д. Используются для разработки такого типа встроенных систем.

Сложные встраиваемые системы

Этот тип встроенных систем имеет много аппаратных и программных сложностей. Вам могут потребоваться IPS, ASIPS, PLA, процессор конфигурации или масштабируемые процессоры. Для разработки этой системы вам необходимо совместное проектирование и компоненты аппаратного и программного обеспечения, которые необходимо объединить в конечную систему.

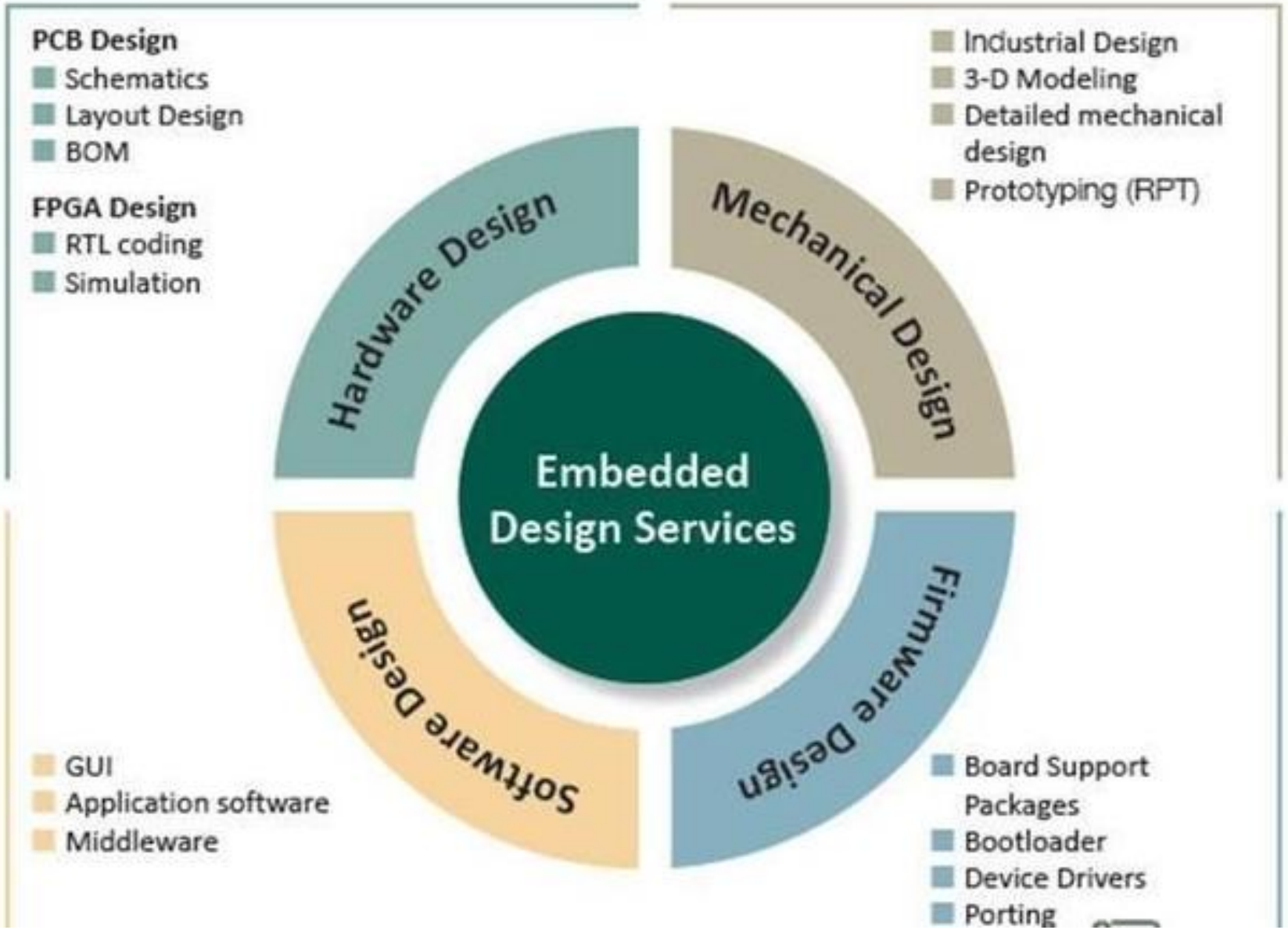
Разница между микропроцессором и микроконтроллером

Микропроцессор	микроконтроллер
Он использует функциональные блоки, такие как регистр, АЛУ, тайминг и блоки управления.	Он использует функциональные блоки микропроцессоров, такие как RAM, таймер, параллельный ввод / вывод, ADC и DAC.
В микропроцессоре инструкция по обработке битов меньше, только один или два типа.	Микроконтроллер предлагает много видов инструкций по обработке битов.
Обеспечивает быстрое перемещение кода и данных между внешней памятью и микропроцессором.	Обеспечивает быстрое перемещение кода и данных в микроконтроллере.
Помогает вам разработать цифровую компьютерную систему общего назначения.	Помогает вам разрабатывать специализированные системы.
Это позволяет вам делать многозадачность одновременно.	Это единая система, ориентированная на задачи.
В микропроцессорной системе вы можете выбрать необходимое количество портов памяти или портов ввода / вывода.	В системе микроконтроллера фиксированное число для памяти или ввода / вывода делает микроконтроллер идеальным для выполнения конкретной задачи.
Обеспечивает поддержку внешней памяти и портов ввода / вывода, что делает ее более тяжелой и дорогой системой.	Этот тип системы является легким и более дешевым по сравнению с микропроцессором.
Внешним устройствам нужно больше места, а их энергопотребление значительно выше.	Этот тип системы занимает меньше места, а энергопотребление также очень низкое.

Области применения встраиваемых систем

- измерительная техника и приборостроение
- медицина
- авиация
- промышленное оборудование
- транспорт
- мобильные и портативные устройства
- торговое оборудование
- робототехника
- системы связи
- бытовая электроника

Принципы проектирования ВС



Преимущества встраиваемой системы

- Способна покрывать самые разные среды
- Менее вероятны повторные ошибки
- Встроенная система упрощает аппаратное обеспечение, что снижает общие расходы.
- Предлагает улучшенную производительность
- Встроенная система полезна для массового производства.
- Встроенная система очень надежна.
- У него очень мало взаимосвязей.
- Встроенная система небольшого размера.
- У нее быстрые операции.
- Предлагает улучшенное качество продукции.
- Оптимизирует использование системных ресурсов.
- Она работает на малой мощности.

Недостатки встраиваемой системы

- Для разработки встроенной системы требуются большие усилия по разработке.
- Требуется много времени для выхода на рынок.
- Встраиваемые системы выполняют очень специфическую задачу, поэтому их нельзя запрограммировать на разные вещи.
- Встроенные системы предлагают очень ограниченные ресурсы для памяти.
- Она не предлагает никаких технологических улучшений.
- Трудно сделать резервную копию встроенных файлов.

Встроенные системы – процессоры, типы архитектуры, инструменты и периферия

Лекция 2

Процессоры в системе

Процессор – это сердце встроенной системы. Это базовая единица, которая принимает входные данные и производит выходные данные после обработки данных. Для разработчика встроенных систем необходимо знание как микропроцессоров, так и микроконтроллеров.

Процессор имеет два основных блока –

- Блок управления потоком программ (CU)
- Модуль исполнения (EC)

Процессоры в системе

CU включает в себя блок выборки для извлечения инструкций из памяти. В EC есть схемы, которые реализуют инструкции, относящиеся к операции передачи данных и преобразования данных из одной формы в другую.

EC включает в себя Арифметико-логический блок (АЛУ), а также схемы, которые выполняют инструкции для задачи управления программой, такой как прерывание, или переход к другому набору инструкций.

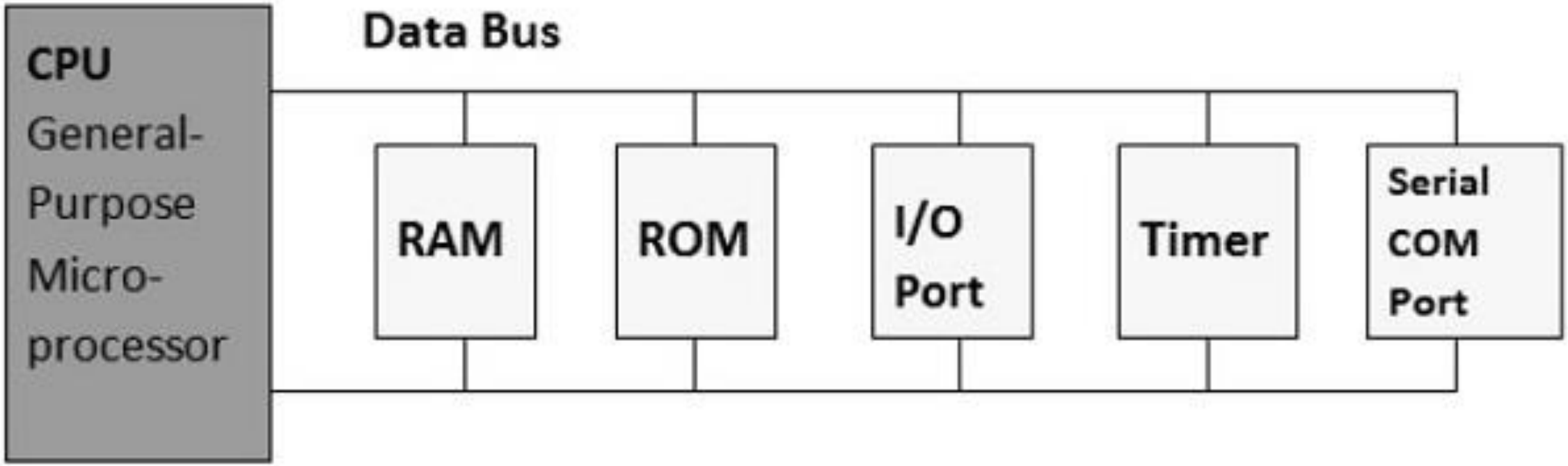
Процессор выполняет циклы выборки и выполняет инструкции в той же последовательности, в которой они извлекаются из памяти.

Типы процессоров

Процессоры могут быть следующих категорий –

- Процессор общего назначения (GPP)
- Микропроцессор
- микроконтроллер
- Встроенный процессор
- Цифровой сигнальный процессор
- Медиапроцессор
- Системный процессор для приложений (ASSP)
- Специфичные для приложений процессорные инструкции (ASIP)
- Ядра (и) GPP или ядро (а) ASIP либо в интегральной микросхеме конкретного приложения (ASIC), либо в схеме очень большой интеграции (VLSI).

Микропроцессор



A SIMPLE BLOCK DIAGRAM OF A MICROPROCESSOR

Микропроцессор

Микропроцессор - это отдельная микросхема СБИС, имеющая центральный процессор. Кроме того, в нем могут быть и другие устройства, такие как тренеры, арифметическое устройство для обработки с плавающей запятой и конвейерные устройства, которые помогают ускорить обработку инструкций.

Цикл выборки и выполнения микропроцессоров предыдущего поколения определялся тактовой частотой порядка ~ 1 МГц. Процессоры теперь работают с тактовой частотой свыше 2 ГГц

Микроконтроллер

Микроконтроллер представляет собой однокристальный модуль СБИС (также называемый микрокомпьютером), который, несмотря на ограниченные вычислительные возможности, обладает расширенными возможностями ввода / вывода и рядом встроенных функциональных блоков.

Процессор	ОЗУ	ПЗУ
Порт ввода-вывода	Таймер	Последовательный COM-порт

Микроконтроллеры, в частности, используются во встраиваемых системах для приложений управления в реальном времени с встроенной программной памятью и устройствами.

Микропроцессор против микроконтроллера

Микропроцессор	Микроконтроллер
Микропроцессоры по своей природе многозадачны. Может выполнять несколько задач одновременно. Например, на компьютере мы можем воспроизводить музыку во время написания текста в текстовом редакторе.	Ориентирован на решение одной задачи. Например, стиральная машина предназначена только для стирки одежды.
ОЗУ, ПЗУ, порты ввода-вывода и таймеры могут быть добавлены извне и могут различаться по количеству.	ОЗУ, ПЗУ, порты ввода-вывода и таймеры не могут быть добавлены извне. Эти компоненты должны быть встроены вместе в микросхему и фиксированы в цифрах.
Проектировщики могут выбрать необходимое количество портов памяти или ввода-вывода.	Фиксированный номер памяти или ввода-вывода делает микроконтроллер идеальным решением для ограниченных, но специфических задач.
Внешняя поддержка внешней памяти и портов ввода-вывода делает систему на базе микропроцессора более тяжелой и дорогостоящей.	Микроконтроллеры имеют малый вес и дешевле, чем микропроцессор.
Для внешних устройств требуется больше места, а их энергопотребление выше.	Система на базе микроконтроллера потребляет меньше энергии и занимает меньше места.

Встроенные системы - Типы архитектуры

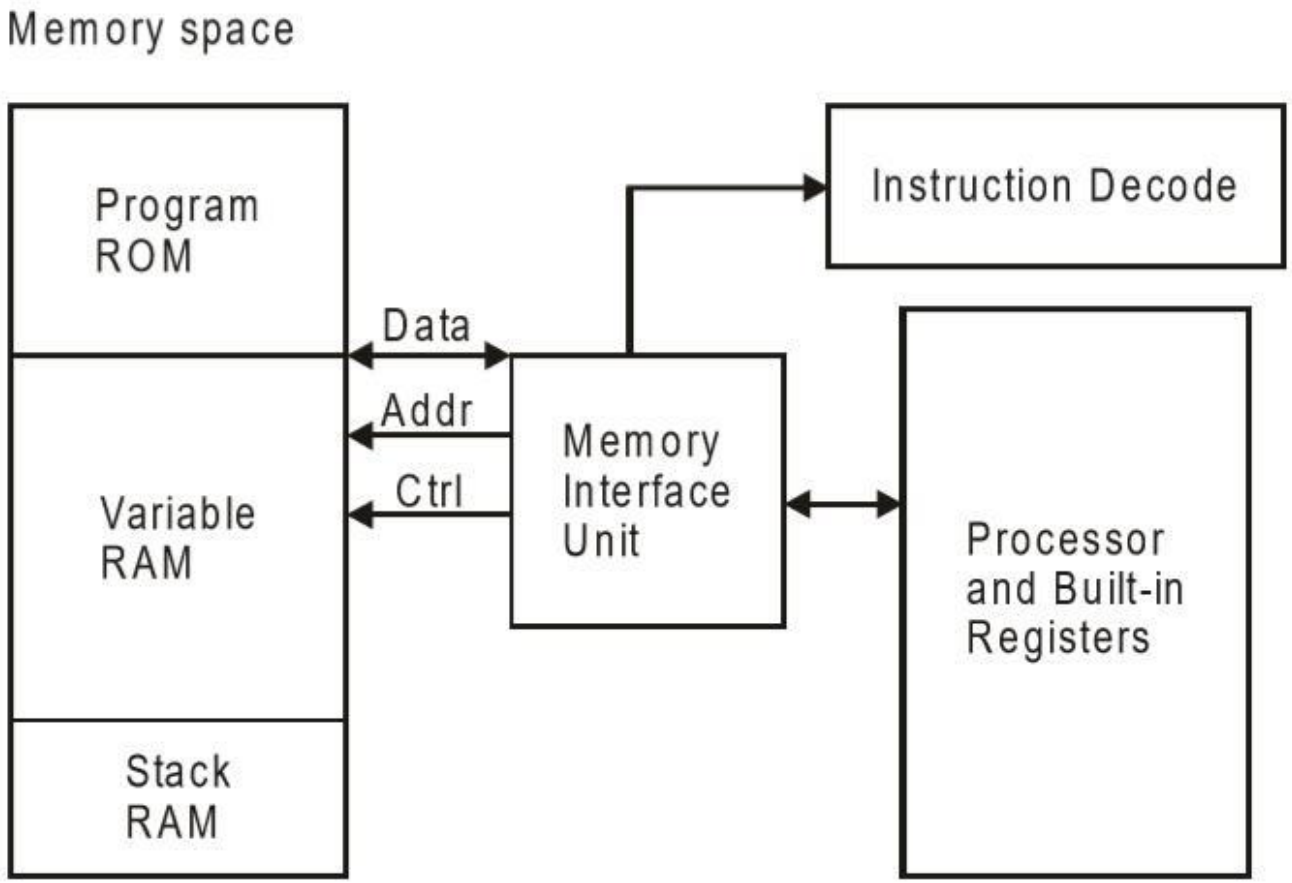
Микроконтроллеры 8051 работают с 8-разрядной шиной данных. Таким образом, они могут поддерживать внешнюю память данных объемом до 64 КБ и внешнюю память программ объемом в лучшем случае 64 КБ. В совокупности микроконтроллеры 8051 могут обрабатывать 128 КБ внешней памяти.

Когда данные и код находятся в разных блоках памяти, тогда архитектура называется **гарвардской архитектурой**. В случае, если данные и код находятся в одном блоке памяти, тогда архитектура называется архитектурой **фон Неймана**.

Архитектура фон Неймана

Архитектура фон Неймана была впервые предложена ученым Джоном фон Нейманом. В этой архитектуре существует один канал передачи данных или шина как для команд, так и для данных. В результате процессор выполняет одну операцию за раз. Он либо извлекает инструкцию из памяти, либо выполняет операцию чтения / записи данных. Таким образом, выборка инструкций и операция с данными не могут выполняться одновременно, используя общую шину.

Архитектура фон Неймана



Архитектура фон Неймана

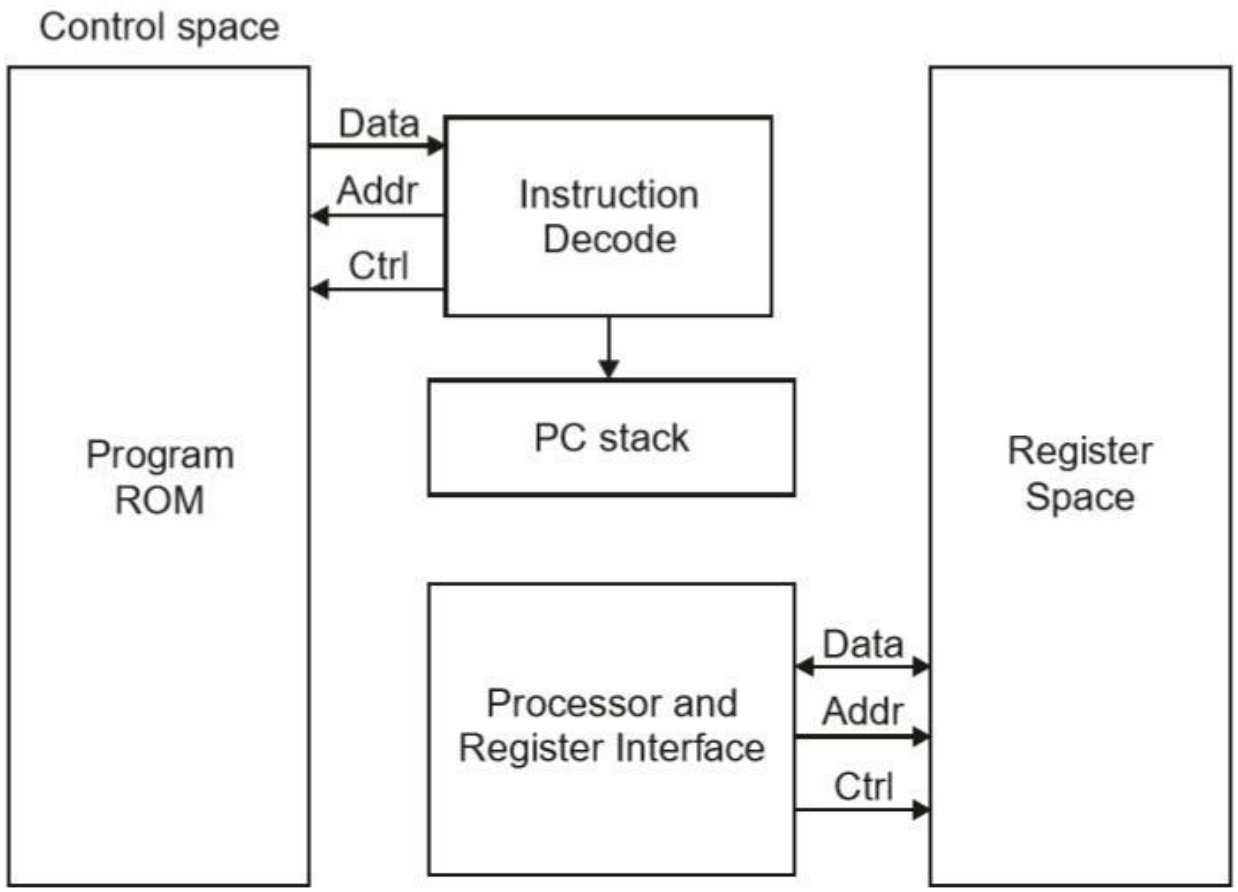
Архитектура фон-Неймана поддерживает простое аппаратное обеспечение. Это позволяет использовать единую последовательную память. Сегодняшние скорости обработки значительно превышают время доступа к памяти, и мы используем очень быстрый, но небольшой объем памяти (кэш), локальной для процессора.

Гарвардская Архитектура

Гарвардская архитектура предлагает отдельные шины хранения и сигналов для инструкций и данных. В этой архитектуре хранилище данных полностью содержится в центральном процессоре, и нет доступа к хранилищу инструкций в виде данных. Компьютеры имеют отдельные области памяти для программных инструкций и данных, используя внутренние шины данных, что обеспечивает одновременный доступ как к инструкциям, так и к данным.

Программы должны были загружаться оператором; процессор не мог загрузиться сам. В гарвардской архитектуре нет необходимости заставлять две памяти совместно использовать свойства.

Гарвардская Архитектура



Архитектура фон-Неймана против архитектуры Гарварда

Основа сравнения	Von Neumann	Архитектура Гарварда
Определение	Архитектура Фон Неймана - это простой стиль компьютерной архитектуры, использующий единственное подключение к памяти.	Архитектура Гарварда является текущим стандартом проектирования, и она имеет ОЗУ и ПЗУ, которые полностью независимы.
Дизайн	Макет прост и использует один и тот же путь как для хранения данных, так и для выполнения инструкций.	По сравнению с архитектурой Фон Неймана эта конструкция более сложная, поскольку в ней используются отдельные подключения для оперативной памяти и ПЗУ.
Аппаратное обеспечение	По сравнению с гарвардской архитектурой требования к оборудованию значительно ниже.	По сравнению с архитектурой Фон Неймана, архитектура Гарварда уделяет большее внимание использованию аппаратных средств.
Скорость	По сравнению с гарвардской архитектурой скорости процессоров значительно ниже.	Архитектура Гарварда быстрее других. Стоимость компьютерного моделирования значительно ниже. ведь архитектура Гарварда требует увеличения доступного пространства.
Физическое пространство	По сравнению с компьютерами с архитектурой Гарварда компьютеры Фон Неймана занимают меньшую площадь с точки зрения требуемого объема физического пространства.	В гарвардской архитектуре требования к фактическому пространству повышены.
Внутренняя память	Поскольку память и программы занимают одно и то же пространство, во внутренней памяти нет неиспользуемого пространства.	Поскольку память команд и память данных не могут занимать одно и то же пространство, часть внутренней памяти Гарварда будет где-то расходоваться впустую.
Инструкции по выполнению	Инструкции для запуска могут быть либо взяты из программы, которая была сохранена, либо они могут быть даны явно. В результате их нельзя рассматривать вместе.	Из-за того факта, что входные данные и программные инструкции, которые хранятся в программе, выполняются одновременно, инструкции по запуску несколько сложны и несколько медленны.

CISC и RISC

CISC - это компьютер со сложным набором команд. Это компьютер, который может обрабатывать большое количество инструкций.

В начале 1980-х годов разработчики компьютеров рекомендовали компьютерам использовать меньше инструкций с простыми конструкциями, чтобы они могли выполняться намного быстрее в процессоре без использования памяти. Такие компьютеры классифицируются как компьютеры с ограниченным набором команд или RISC.

CISC против RISC

CISC	RISC
Более обширный набор инструкций. Простота программирования	Меньший набор инструкций. Сложно программировать.
Упрощенный дизайн компилятора с учетом большего набора инструкций.	Сложный дизайн компилятора.
Множество режимов адресации, вызывающих сложные форматы команд.	Несколько режимов адресации, исправьте формат инструкции.
Длина инструкции может варьироваться.	Длина инструкции варьируется.
Более высокие тактовые частоты в секунду.	Низкий такт в секунду.
Акцент делается на аппаратном обеспечении.	Основное внимание уделяется программному обеспечению.
Блок управления реализует большой набор команд с помощью микропрограммного блока.	Каждая инструкция должна выполняться аппаратным обеспечением.
Более медленное выполнение, поскольку инструкции должны считываться из памяти и декодироваться блоком декодирования.	Более быстрое выполнение, поскольку каждая инструкция должна выполняться аппаратным обеспечением.
Конвейерная обработка невозможна.	Возможна конвейерная обработка инструкций с учетом одного такта.

Встроенные системы - Инструменты и периферийные устройства. Компилятор

Компилятор - это компьютерная программа (или набор программ), которая преобразует исходный код, написанный на языке программирования (исходном языке), в другой компьютерный язык (обычно двоичный формат). Наиболее распространенной причиной преобразования является создание исполняемой программы. Название "компилятор" в основном используется для программ, которые переводят исходный код с языка программирования высокого уровня на язык низкого уровня (например, язык ассемблера или машинный код).

Кросс-компилятор

Если скомпилированная программа может выполняться на компьютере с процессором или операционной системой, отличными от компьютера, на котором компилятор скомпилировал программу, тогда этот компилятор называется кросс-компилятором.

Декомпилятор

Программа, которая может переводить программу с языка низкого уровня на язык высокого уровня, называется декомпилятором.

Конвертер языков

Программа, которая переводит программы, написанные на разных языках высокого уровня, обычно называется переводчиком языков, переводчиком из источника в источник или преобразователем языков.

Компилятор, скорее всего, будет выполнять следующие операции –

- Предварительная обработка
- Синтаксический анализ
- Семантический анализ (синтаксический перевод)
- Генерация кода
- Оптимизация кода

Ассемблеры

Ассемблер - это программа, которая берет базовые компьютерные инструкции (называемые языком ассемблера) и преобразует их в набор битов, которые процессор компьютера может использовать для выполнения своих основных операций. Ассемблер создает объектный код, преобразуя мнемонику инструкции ассемблера в коды операций, преобразуя символьные имена в ячейки памяти. Язык ассемблера использует мнемонику для представления каждой низкоуровневой машинной операции (кода операции).

Средства отладки во встроенной системе

Отладка - это методичный процесс поиска и уменьшения количества ошибок в компьютерной программе или части электронного оборудования, чтобы оно работало должным образом. Отладка затруднена, когда подсистемы тесно связаны, потому что небольшое изменение в одной подсистеме может привести к ошибкам в другой. Средства отладки, используемые во встроенных системах, сильно различаются по времени разработки и возможностям отладки. Здесь мы обсудим следующие инструменты отладки −

- Симуляторы
- Стартовые наборы микроконтроллеров
- Эмулятор

Симуляторы

Код тестируется для MCU / системы путем его моделирования на главном компьютере, используемом для разработки кода. Симуляторы пытаются моделировать поведение всего микроконтроллера в программном обеспечении.

Функции симуляторов

Симулятор выполняет следующие функции −

- Определяет процессор или семейство устройств обработки, а также его различные версии для целевой системы.
- Отслеживает подробную информацию о части исходного кода с метками и символьными аргументами по мере выполнения каждого отдельного шага.
- Обеспечивает состояние оперативной памяти и моделируемых портов целевой системы для выполнения каждого отдельного шага.
- Отслеживает реакцию системы и определяет пропускную способность.
- Обеспечивает трассировку вывода содержимого счетчика программ по сравнению с регистрами процессора.
- Содержит подробное значение данной команды.
- Отслеживает подробную информацию о командах симулятора, вводимых с клавиатуры или выбираемых из меню.
- Поддерживает условия (до 8, 16 или 32 условий) и безусловные точки останова.
- Предоставляет точки останова и трассировку, которые вместе являются важным инструментом тестирования и отладки.
- Облегчает синхронизацию внутренних периферийных устройств и задержек.

Стартовый набор для микроконтроллеров

Стартовый комплект микроконтроллера состоит из:

- Плата аппаратного обеспечения (оценочная плата)
- Внутрисхемный программатор
- Некоторые программные средства, такие как компилятор, ассемблер, компоновщик и т.д.

Иногда пробная версия компилятора IDE и размер кода ограничены.

Большим преимуществом этих наборов перед симуляторами является то, что они работают в режиме реального времени и, таким образом, позволяют легко проверять функциональность ввода/вывода. Однако стартовых наборов вполне достаточно и это самый дешевый вариант для разработки простых проектов микроконтроллеров.

Эмуляторы

Эмулятор - это аппаратный набор или программное обеспечение, или может быть и тем, и другим, которые эмулируют функции одной компьютерной системы (гостевой) в другой компьютерной системе (хосте), отличной от первой, так что эмулируемое поведение очень похоже на поведение реальной системы (гостевой).

Эмуляция относится к способности компьютерной программы в электронном устройстве эмулировать (имитировать) другую программу или устройство. Эмуляция направлена на воссоздание оригинальной компьютерной среды. Эмуляторы имеют возможность поддерживать более тесную связь с подлинностью цифрового объекта. Эмулятор помогает пользователю работать с любым приложением или операционной системой на платформе аналогично тому, как программное обеспечение работает в исходной среде.

Периферийные устройства во встроенных системах

Встроенные системы взаимодействуют с внешним миром через свои периферийные устройства, такие как

- Интерфейсы последовательной связи (SCI), такие как RS-232, RS-422, RS-485 и др.
- Синхронный последовательный интерфейс связи, такой как I^2C, SPI, SSC и ESSI
- Универсальная последовательная шина (USB)
- Мультимедийные карты (SD-карты, Compact Flash и т.д.)
- Сети, такие как Ethernet, LonWorks и др.
- Промышленные шины, такие как CAN-Bus, LIN-Bus, PROFIBUS и т.д.
- такие как PLL (phase lock loop, ФАПЧ), модули захвата/сравнения и обработки времени.
- Дискретный ввод-вывод, он же ввод-вывод общего назначения (GPIO)
- Аналого-цифровой / цифро-аналоговый преобразователи (АЦП / ЦАП)
- Отладка с использованием портов JTAG, ISP, ICSP, BDM, BITP и DP9

Критерии выбора микроконтроллера

При выборе микроконтроллера убедитесь, что он соответствует поставленной задаче и является экономически эффективным. Мы должны выбрать, какой 8-разрядный, 16-разрядный или 32-разрядный микроконтроллер лучше всего справится с вычислительными потребностями задачи. Кроме того, при выборе микроконтроллера следует учитывать следующие моменты −

- Скорость − Какую максимальную скорость может поддерживать микроконтроллер?
- Корпус − это 40-контактный DIP (двойной встроенный корпус) или QFP (квадратный плоский корпус)? Это важно с точки зрения пространства, сборки и прототипирования конечного продукта.
- Энергопотребление − это важный критерий для продуктов с батарейным питанием.
- Объем ОЗУ и ПЗУ на чипе.
- Количество выводов ввода-вывода и таймеров на микросхеме.
- Стоимость за единицу − это важно с точки зрения конечной стоимости продукта, в котором будет использоваться микроконтроллер.

Кроме того, убедитесь, что у вас есть такие инструменты, как компиляторы, отладчики и ассемблеры, доступные с микроконтроллером. Самое главное, вы должны приобрести микроконтроллер из надежного источника.

Технические средства встраиваемых систем

Лекция 3

Программируемые логические интегральные схемы

Программируемая логическая интегральная схема (PLD, Programmable Logic Device) – электронный компонент, состоящий из логических ячеек и конфигурируемой схемы соединений. Основное назначение – построения реконфигурируемых цифровых схем. В отличие от обычных интегральных схем, функциональность не определяется на этапе изготовления раз и навсегда и может создаваться и изменяться конечным пользователем (инженером), исходя из своих нужд.

Программируемые логические интегральные схемы

ПЛИС, в первом приближении, представляет собой множество однотипных логических элементов, соединяемых с помощью специальных коммутационных матриц. Соединение и инициализация элементов осуществляется посредством бинарного образа, загружаемого в конфигурационную память ПЛИС. Файлы конфигурации (бинарные образы) генерируются с помощью САПР производителя конкретной ПЛИС и являются его интеллектуальной собственностью. На аппаратной базе FPGA можно реализовывать системы на кристалле. Описания проектов возможно делать как на языках структурно-функционального описания аппаратуры (Verilog,VHDL), так и при помощи более высокоуровневых языков, таких как SystemC.

Программируемые логические интегральные схемы

Применение ПЛИС:

• Связывающая логика (glue logic). ПЛИС выступает как средство обеспечения совместимости нескольких интерфейсов. Изначально это достигалось с помощью логики 74 и 40 серий, а в более сложных случаях с помощью CPLD и FPGA. Примеры: декодер шины адреса, расширитель портов.

• Для шифрования интеллектуальной собственности электронных схем

• Для прототипирования ИС и в малосерийном производстве

• Цифровая обработка сигналов, изображений

• Криптография

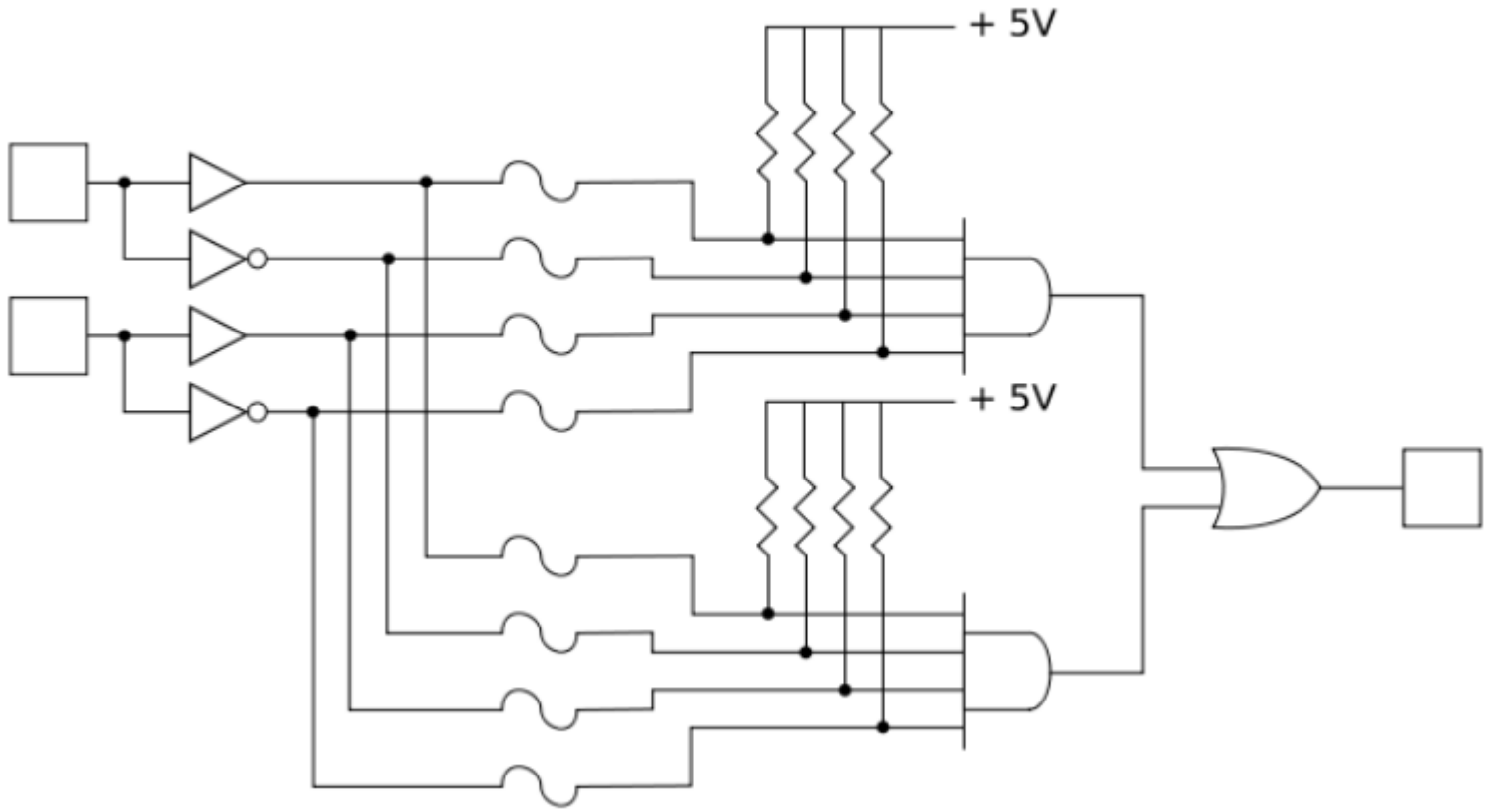
• Высокопроизводительные вычисления

Программируемая логическая матрица

PAL (Programmable Array Logic) – программируемая логическая матрица (ПЛМ). Представляет собой простую программируемую логическую интегральную схему (ПЛИС).

ПЛМ делаются на базе технологии ПЗУ, то есть они программируются однократно. Внутри, ПЛМ представляют собой несколько логических элементов «И» с перемычками на входах, выходы которых подключены фиксированным образом к элементам «ИЛИ». Процесс программирования заключается в пережигании перемычек (на рисунке они показаны волнистой линией) с целью получения нужной булевой функции.

Фрагмент ПЛМ: входной и выходной буфер, матрица И (2 терма), матрица ИЛИ



CPLD

CPLD (complex programmable logic device) – электронное устройство, принадлежащее к классу программируемых электронных схем (ПЛИС) и находящееся по сложности между FPGA и PAL. CPLD состоят из блоков логических вентилей, объединенных программируемой коммутационной матрицей. В отличие от FPGA, схемы CPLD делают обычно на базе энергонезависимой памяти. В последнее время, различия между CPLD и FPGA постепенно стираются.

FPGA

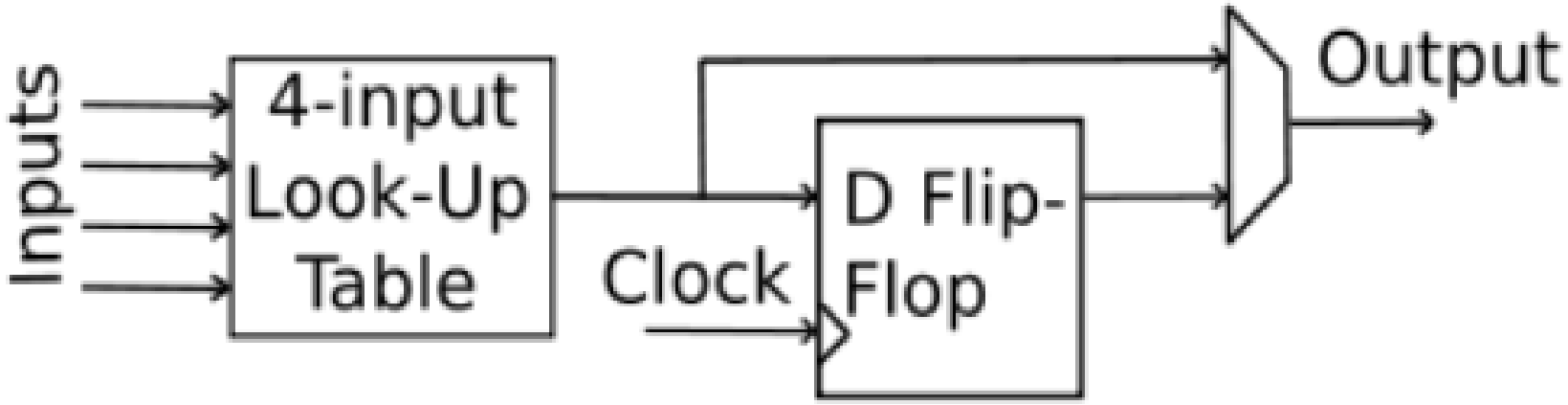
FPGA (Field-Programmable Gate Array) представляет собой множество однотипных логических элементов, соединить которые можно с помощью программируемой коммутационной схемы. Основная цель создания FPGA состоит в том, чтобы позволить проектировщику с помощью относительно простых технологий, в лабораторных условиях получить на кристалле достаточно сложное и при этом ещё и работающее цифровое устройство. Итак, что мы выигрываем при использовании технологии FPGA? Во-первых, мы избавляемся от сложного производства интегральных схем. Во-вторых, упрощается проектирование кристалла из-за отсутствия проблем с топологией микросхемы и взаимным влиянием различных узлов. Что мы теряем? Мы теряем производительность, снижаем надежность, увеличиваем чувствительность к помехам, увеличиваем энергопотребление кристалла и получаем меньшую отдачу от использования площади кристалла из-за роста количества вентилей в 20..30 раз.

FPGA

К основным направлениям применения FPGA можно отнести: прототипирование, симуляцию и изготовление мелко и среднесерийных устройств. При достаточно больших партиях устройств становится экономически выгодным изготовление ASIC.

Схемы строятся на баз логических элементов. Каждый логический элемент FPGA состоит из двух основных частей: программируемого логического элемента (так называемый LUT — Lookup Table) и триггера на выходе.

Логический элемент FPGA



FPGA

Один логический элемент состоит из нескольких десятков логических вентилей сделанных в базисе 2 И-НЕ или 2 ИЛИ-НЕ. Программируемый логический элемент содержит в себе таблицу истинности, позволяющую реализовать любую комбинационную схему. Как правило, такого рода элементы выполняют в виде набора регистров памяти. Так называемая конфигурационная память FPGA, содержит в себе информацию необходимую для соединения логических элементов и таблицы истинности для LUT.

FPGA

Логические элементы, составляющие FPGA, принадлежат уровню RTL. Их соединение хорошо описывается структурными VHDL и Verilog. Используемая модель вычислений — модель дискретных событий.

По мере совершенствования технологий производства интегральных схем появляется тенденция к укрупнению базовых элементов FPGA. В большинство современных моделей добавляют память, элементы арифметико-логических устройств, умножители и целые процессорные ядра.

Системы-на-кристалле

Система-на-кристалле (System-on-Chip, SoC) – в общем случае системы, на едином кристалле которых интегрированы процессор (процессоры, в том числе специализированные), некоторый объем памяти, ряд периферийных устройств и интерфейсов, — то есть максимум того, что необходимо для решения задач, поставленных перед системой. Выражение "система на кристалле" не является, строго говоря, термином. Это понятие отражает общую тенденцию к повышению уровня интеграции за счет интеграции функций.

Системы-на-кристалле

Производительность приборов класса "система-на-кристалле" в значительной мере зависит от эффективности взаимодействия всех встроенных компонентов и от эффективности их взаимодействия с внешним, относительно прибора, миром. В первую очередь это связано с различием в быстродействии встроенных компонентов, в особенности организации интерфейсов.

Системы-на-кристалле

Системы на кристалле обычно состоят из трех основных цифровых системных блоков: процессор, память и логика. Процессорное ядро реализует поток управления, когда каждой управляющей программой однозначно устанавливаются последовательности выполнения операций обработки данных, что позволяет задавать один из возможных алгоритмов работы всей интегральной схемы. Память используется по ее прямому назначению — хранение кода программы процессорного ядра и данных. Наконец, логика используется для реализации специализированных аппаратных устройств обработки и прохождения данных, состав и назначение которых определяются конечным приложением — потока данных.

Системы-на-кристалле

Реальная система на кристалле содержит как минимум все три перечисленных блока, что исключает применение многочисленных отдельных интегральных схем и реализацию интерфейсов связи между ними. Однокристальное конфигурируемое или программируемое решение допускает оперативное изменение своей внутренней аппаратной структуры и конечного предназначения как на этапе производства, так и в полевых условиях, непосредственно в проекте. Такие интегральные схемы были отнесены к группе изделий системного уровня интеграции, но получили другое название — Configurable System on a Chip или CSoC. Поскольку термин CSoC не стандартизован, то существуют и другие названия изделий этого класса — System on Programmable Chip (SoPC), Programmable System on a Chip (PSoC) или просто SoC, что определяется вкусом и желаниями конкретного производителя микросхем.

Системы-на-кристалле

Типовая встраиваемая система, построенная на базе SoC, содержит различные наборы следующих интерфейсов и контроллеров:

- Системная шина и контроллеры шин LPC/ISA, PCI, PCMCIA;
- Контроллеры управления NOR/NAND Flash, SDRAM, SRAM, DDR;
- Контроллер Ethernet;
- Последовательные интерфейсы UART, SPI/SSP/uWire, RS-232, RS- 422/RS-485, CAN;
- Беспроводные интерфейсы WiFi/IEEE802.11, ZigBee, Bluetooth, IrDA;
- Интерфейсы поддержки Flash-карт памяти: SD/MMC, CompactFlash, MemoryStick;
- Контроллер LCD STN/TFT/OLED;
- Контроллер матричной клавиатуры;
- Модули беспроводной передачи данных GSM/GPRS, CDMA;
- Модули приема сигналов спутниковых навигационных систем GPS, Glonass;
- Аппаратные поддержки плавающей точки, шифрования, DRM и т.п.;
- Аудио- и видеоинтерфейсы.

Модульный принцип организации процессора ВВС

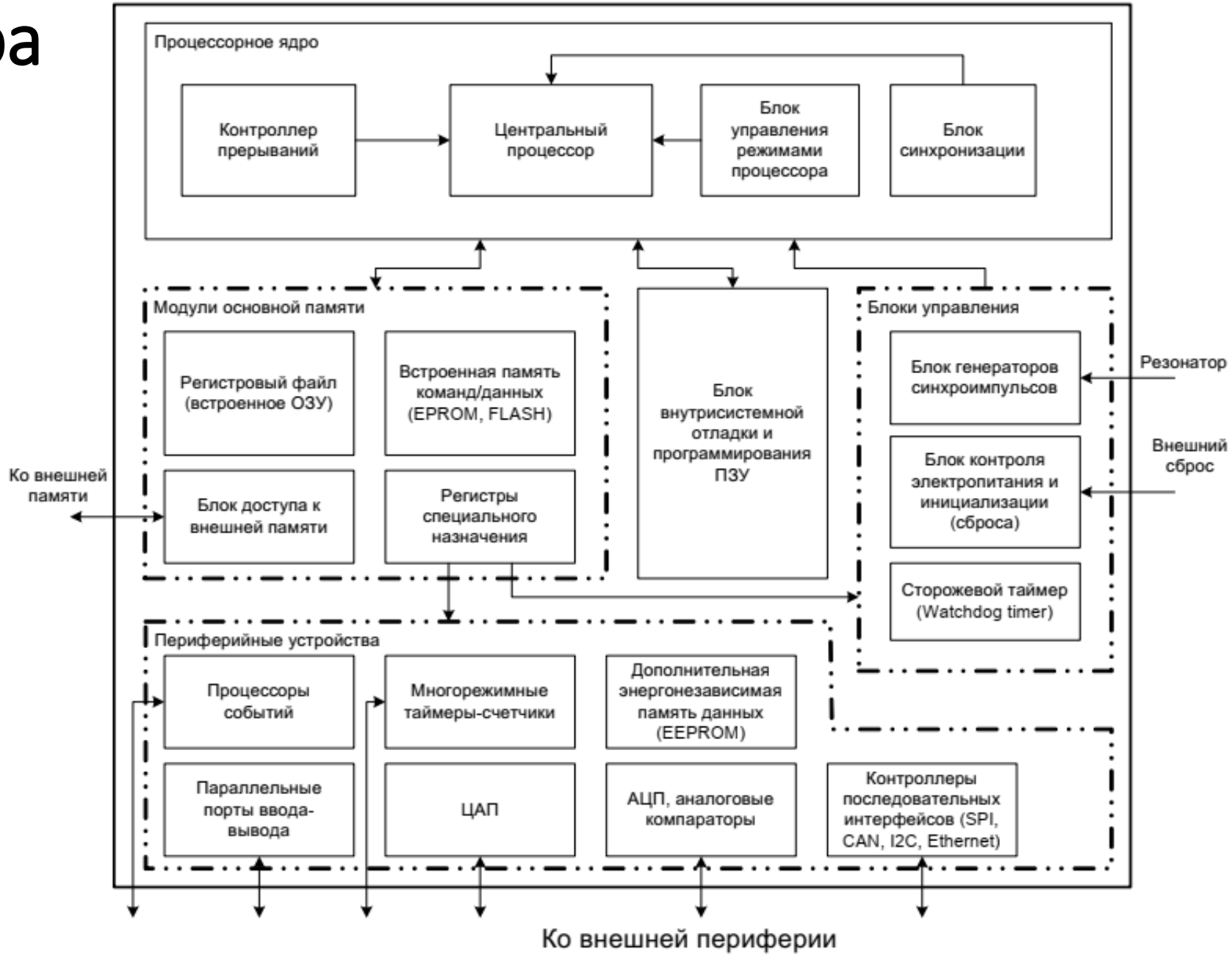
Лекция 4

Типовая структура процессора для встраиваемых систем

В настоящее время выпускается большое количество разнообразных по структуре и функциям процессоров для применения во встроенных системах.

Эта номенклатура постоянно расширяется, чтобы обеспечить решение специфических задач в различных прикладных задачах. Возможность разработки и производства новых моделей в сжатые сроки обеспечивает модульный принцип структурной организации.

Типовая структура процессора для встроенных систем



Типовая структура процессора для встраиваемых систем

При модульном принципе построения все процессоры одного семейства содержат в себе одинаковый базовый функциональный блок – процессорное ядро, и изменяемый функциональный блок.

Базовый блок (процессорное ядро) включает:

- Центральный процессор;
- Внутренние магистрали адреса, данных и управления;
- Блок формирования множества сигналов с различными фазами и частотами для синхронизации центрального процессора и внутренних магистралей.
- Блок управления режимами работы процессора, который может настраивать процессор на активный режим, режим обработки прерываний, несколько режимов пониженного энергопотребления, режим рестарта.

Типовая структура процессора для встраиваемых систем

Процессорное ядро является основным отличительным признаком архитектуры определенного семейства процессоров, поэтому его (ядро) называют по названию семейства. Например, ядро MCS-51 или ядро PIC16.

Изменяемый функциональный блок включает:

- Модули памяти различных типов: оперативную память данных типа SRAM, постоянную память команд (программ) типов ROM, EPROM или FLASH, энергонезависимую память данных типа EEPROM;
- Модули периферийных устройств;
- Модули управления и синхронизации.

Типовая структура процессора для встраиваемых систем

В различных микросхемах семейства может иметься различный набор модулей изменяемого функционального блока. Общую совокупность модулей, реализованных в микросхемах одного семейства, называют библиотекой периферийных модулей данного семейства. В данную библиотеку входят, как говорилось, не только периферийные, но и модули памяти, встроенные генераторы синхронизации, блок контроля электропитания и формирования сигналов рестарта системы в случае сбоев или «внешнего сброса», модули внутрисхемной отладки и программирования. В последнее время активно развивается направление «System-On-Chip», когда конечный пользователь сам может формировать структуру специализированного процессора из предоставленной библиотеки периферийных модулей, а также самостоятельно разрабатывать новые модули.

Процессорное ядро

Техническое решение процессорного ядра определяют следующие параметры:

- Архитектурные – набор регистров, организация памяти, способы адресации операндов в памяти, система команд для обработки этих данных.
- Схемотехнические решения – схемы регистров, АЛУ, схемы управления магистралями и т.п. Схемотехника определяет также внутреннюю диаграмму функционирования – последовательность перемещения данных по магистралям между регистрами, памятью, АЛУ.
- Технология производства – определяет допустимую сложность схемы, максимальную частоту переключений, энергопотребление.

В современных процессорах для встраиваемых систем реализуют как CISC-архитектуру (Motorola HC11, Intel MCS-51, AMD Am186 и др.), так и RISC-архитектуру (MicrochipPIC, Atmel AVR, Triscend E7-ARM)

Процессорное ядро

Производительность процессорного ядра определяется комплексом факторов:

• Частотой тактирования межмодульных магистралей адреса и данных Fbus. Она определяется из частоты генератора синхронизации Fxclk по соотношению, индивидуальному для каждого процессорного ядра. Например, для MCS51 – Fxclk/Fbus = 12 и при частоте генератора 12МГц ядро работает на частоте 1МГц; для Am186ES - Fxclk/Fbus = 1; для Motorola HC08 существует режим умножения входной частоты и Fxclk/Fbus < 1.

• Количеством пересылок регистр-регистр за единицу времени. Для RISC-процессоров это одна пересылка за такт шины, для CISC – 1..3 пересылки (они медленнее).

• Производительностью при выполнении операций наиболее используемым в конкретном алгоритме управления. Например, для ПИД – регуляторов – это операции умножения/деления; для простых конечных автоматов – это логические операции.

• Временем вызова/возврата подпрограммы обработки прерывания. Этот параметр значим для функционирования в режиме жесткого реального времени и определяет максимальную интенсивность обрабатываемых событий

Организация прерываний в управляющих процессорах

Источниками прерываний могут быть:

1. Внешние источники. Запрос передается перепадом напряжения на

входе (из «1» в «0» или из «0» в «1») или определенным уровнем

напряжения («0» или «1») на внешнем входе запроса прерывания.

2. Внутренние источники – встроенные модули памяти (обычно от модуля

EEPROM) или модули периферийных устройств:

a) Таймеры/счетчики. Запрос вырабатывается по переполнению;

b) Блоки захвата/сравнения. Запрос по событию входного захвата или равенства при выходном сравнении.

c) АЦП. Запрос по завершению преобразования.

d) Аналоговые компараторы. Запрос по изменению соотношения

уровней входных сигналов.

e) Приемопередатчики последовательных интерфейсов (RS-232 (SIO),

SPI, I2C, USB, CAN, Ethernet, HDLC и т.п.). Запрос вырабатывается:

♦ По приему байта или пакета и доступности новых принятых данных;

♦ По завершению передачи байта или пакета и освобождению передатчика.

3. Программные прерывания.

Организация прерываний в управляющих процессорах

Организация прерываний в процессорах для управляющих систем ничем принципиально не отличается от универсальных процессоров. В различных семействах управляющих процессоров реализованы различные механизмы обработки прерываний:

1. Векторный с жестким приоритетом (ST7, AVR, Am186).

2. Векторный с программируемым приоритетом (MCS-51, M16C, i386EX).

3. Векторный с динамической таблицей векторов (M16C).

4. С общим вектором (механизм полинга) (PIC).

Укрупненная схема блока обработки запросов прерываний (механизм масок)

Механизм масок основан на использовании специального бита для каждого запроса прерываний, с помощью которого разрешается или запрещается обработка прерываний, связанных с этим запросом. В процессоре семейства Intel функцию маски выполняет бит IF, с помощью которого разрешается (IF=1) или запрещается (IF=0) обработка запросов внешних прерываний (как правило, от ВУ). При сброшенном флаге IF принято говорить, что прерывание замаскировано.

Запросы прерываний от ВУ формируются с помощью PIC (Programmable Interruption Controller), который связан линией запроса с CPU. Запросы от PIC поступают в CPU на внешний вход INTR.

Укрупненная схема блока обработки запросов прерываний (механизм масок)

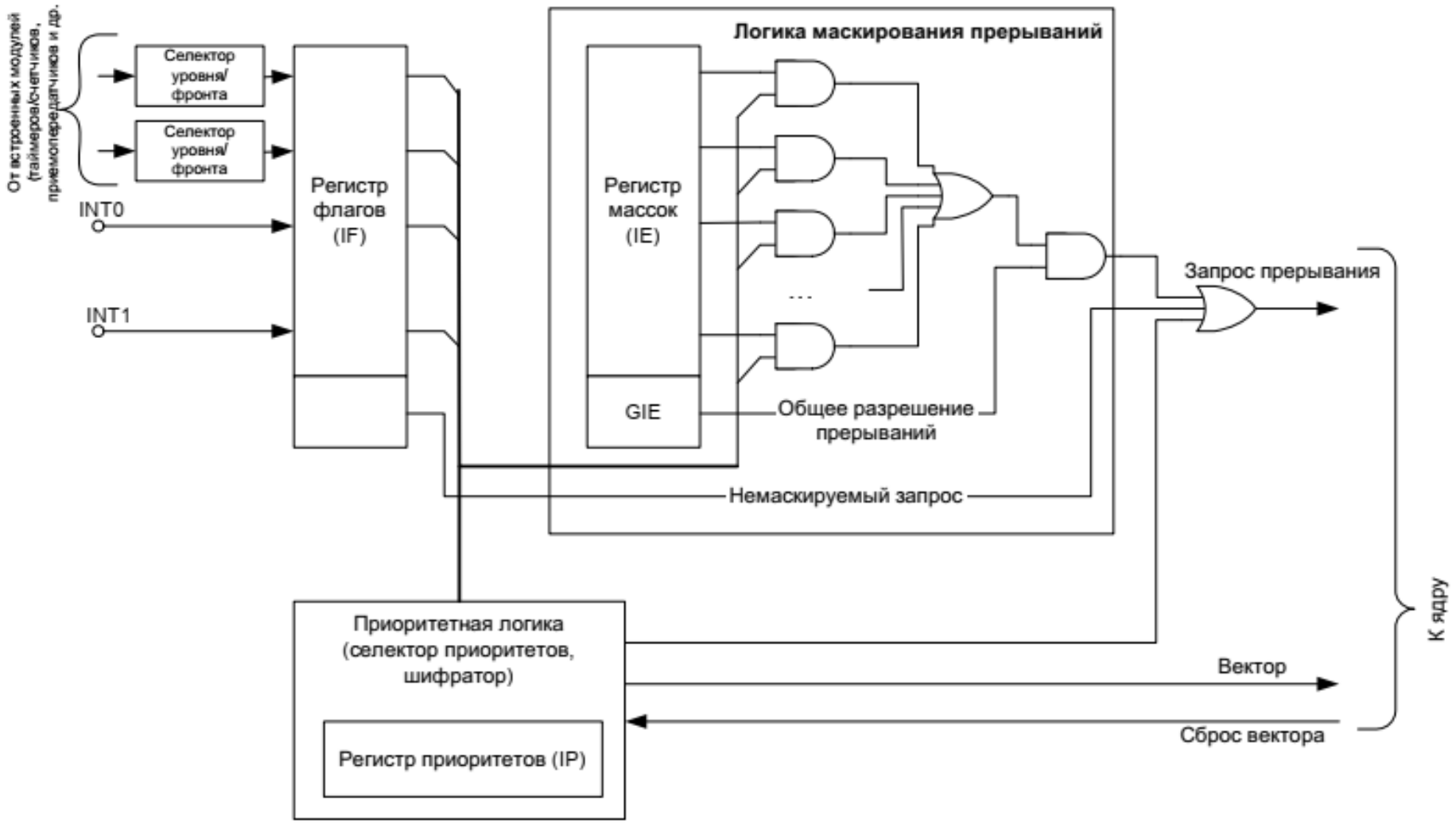
Для управления прерываниями используется маска прерываний, представляющая собой двоичное слово $M = m_1\ m_2 \ldots m_k$ с числом разрядов, равным числу маскируемых причин прерывания. Если разряд маски $m_k = 0$, то прерывание по причине k запрещено (замаскировано), если разряд маски $m_k = 1$, то прерывание по причине k разрешено (не замаскировано). Маска прерываний хранится в процессоре, куда она загружается командой УСТАНОВИТЬ МАСКУ А, где А – адрес. По этой команде слово с адресом А загружается в качестве маски в процессор и определяет отношение процессора к сигналам прерывания. Если все разряды маски равны нулю, процессор не реагирует ни на одну причину прерывания.

Укрупненная схема блока обработки запросов прерываний (механизм масок)

В простейших процессорах используется следующий способ маскирования прерываний. В систему команд компьютера вводятся две системные команды ЗАПРЕТИТЬ ПРЕРЫВАНИЯ и РАЗРЕШИТЬ ПРЕРЫВАНИЯ, выполнение которых приводит к запрещению и разрешению прерываний одновременно по всем причинам.

Команды, маскирующие прерывания, относятся к группе привилегированных команд.

Блок обработки запросов прерываний



Укрупненная схема блока обработки запросов прерываний (механизм масок)

На рисунке селектор уровня/фронта внешнего сигнала запроса прерывания выбирает событие, по которому вырабатывается запрос прерывания от внешнего сигнала («внешнее прерывание»). Возможны следующие настройки: по перепаду (фронт или спад сигнала) или по уровню. При возникновении запроса прерывания в регистре флагов (Interrupt Flag, IF) устанавливается бит, соответствующий источнику. Логика маскирования прерываний разрешает или запрещает выработку запросов от определенных источников или от всех источников сразу. Для разрешения прерывания необходимо установить в «1» соответствующий бит регистра масок (Interrupt Enable, IE) и бит общего разрешения прерываний (Global Interrupt Enable, GIE). Логика маскирования никак не влияет на немаскируемый запрос прерывания (Non-Maskable Interrupt, NMI). Приоритетная логика (Interrupt Priority, IP) вырабатывает вектор для наиболее приоритетного запроса и передает его вычислительному ядру синхронно с сигналом запроса прерывания; отслеживает приоритет запроса, находящегося в обработке и вытесняет (прерывает) данных запрос, если пришел другой запрос с более высоким приоритетом.

Модули памяти

Память – совокупность устройств, предназначенных для хранения программ, обрабатываемой информации (данных), промежуточных или окончательных результатов вычислений.

Важнейшие характеристики памяти – емкость, быстродействие и стоимость. Емкость ЗУ определяется предельным количеством информации, размещаемым в ЗУ, и исчисляется в кило-, мега- и гигабайтах. Быстродействие ЗУ характеризуется затратами времени на чтение запись информации при обращении к ЗУ. Стоимость ЗУ – это затраты средств в денежном выражении на хранение всего объема информации, определяемого емкостью ЗУ. Для сравнения качества ЗУ различных типов используется характеристика, называемая удельной стоимостью и равная стоимости ЗУ, деленной на емкость ЗУ. Удельная стоимость имеет размерность, например, доллар/Мбайт.

Модули памяти

В зависимости от назначения и особенностей реализации устройств памяти, по-разному подходят и к вопросам их классификации.

Критерии классификации:

1. По назначению;

2. По виду физического носителя (технология производства);

3. По организации доступа (адресный: произвольный, прямой

(циклический), последовательный; ассоциативный доступ);

4. По возможности записи и перезаписи;

5. По энергозависимости/энергонезависимости;

6. По типу интерфейса;

7. По типу организации адресного пространства;

8. По удалённости и доступности для центрального процессора

(первичная, вторичная, третичная память).

Модули памяти

Термин «модуль памяти» подразумевает объединение собственно массивов ячеек памяти со специальными аналоговым и цифровыми схемами управления режимами записи–стирания, со схемами (и иногда источниками) электропитания, с регистрами управления режимами.

В качестве примера приведем классификацию модулей памяти по критерию энергозависимости. В этом случае модули памяти делятся на ПЗУ и ОЗУ.

Модули памяти

Модули ПЗУ бывают:

• ПЗУ масочного типа (MaskROM) – записывается на заводеизготовителе и не может быть изменено пользователем. Обладают высоким качеством хранения. Самый дешевый тип ПЗУ. Используются для изделий, выпускаемых большими партиями в несколько десятков тысяч штук.

• ПЗУ, однократно программируемые пользователем (One-Time Programmable ROM (OTPROM)). Программируется пользователем. При выпуске на заводе все ячейки имеют значения FFh. Подачей импульсов напряжения в биты могут быть записаны «0», но обратная запись – в «1» уже не возможна. Имеют высокое качество хранения при строгом соблюдении режимов программирования (уровни напряжения, временная диаграмма, режимы проверки), в противном случае через некоторое время (месяцы-годы) биты могут самопроизвольно «распрограммироваться» - перейти в состояние «1». Дешевое ПЗУ. Используется для небольших партий изделий.

Модули памяти

• ПЗУ программируемое пользователем, со стиранием ультрафиолетовыми или рентгеновскими лучами (EPROM). Допускается многократное перепрограммирование (несколько десятков раз). Технология программирования схожа с OTPROM, но может быть осуществлено стирание всех запрограммированных в «0» ячеек в состояние «1» под УФ лучами. Для того на корпусе есть специальное окно из кварцевого стекла. Нарушение режимов стирания программирования приводит к резкому сокращению числа циклов перепрограммирования и времени хранения. Очень дорогая память (примерно на порядок дороже чем OTPROM). Используется в отладочных образцах.

Модули памяти

ПЗУ программируемое пользователем с электрическим стиранием (Electrically Erasable Programmable ROM – EEPROM или E2PROM). Допускается перезапись произвольной ячейки. При этом стирание выполняется автоматически, прозрачно для пользователя. Число циклов перезаписи до 10000..100000 шт. Значительное время хранения (годы .. 10 лет). Однако, блоки EEPROM имеют ограниченный объем (байты…десятки кБ), в связи с чем их почти всегда используют как память данных.

Модули памяти

ПЗУ с электрическим стиранием типа FLASH является модификацией EEPROM со значительно увеличенным объемом. Для увеличения объема удалены схемы стирания каждого бита по отдельности и стирание выполняется страницами размером от десятков байт до десятков кБ. Также доступно стирание блоками по несколько страниц или всей памяти разом. Такой режим работы (страничное стирание) неудобен для хранения данных, но приемлем для записи программ. Поэтому память FLASH используется в качестве памяти программ. Объем встраиваемой FLASH-памяти – десятки-сотни кБ. Число циклов перепрограммирования – до 100000. Время хранения – до 10 лет. Время стирания –десятки ms на килобайт, время программирования – десятки мкс на байт. Напряжение питания от 1.8 В. ПЗУ типа FLASH в настоящее время выходит на ведущие позиции в секторе встраиваемых и внешних модулей (микросхем) памяти ПЗУ.

Модули памяти

В качестве встроенного ОЗУ в большинстве случаев используются модули статической памяти (Static Random Access Memory, SRAM). Ядро микросхемы статической оперативной памяти представляет собой совокупность триггеров – логических устройств, имеющих два устойчивых состояния, одно из которых условно соответствует логическому нулю, а другое – логической единице.

Другими словами, каждый триггер хранит один бит информации.

Модули памяти

К достоинствам триггера по сравнению с конденсатором в динамических ОЗУ можно отнести:

• состояния триггера устойчивы и при наличии питания могут сохраняться бесконечно долго, в то время как конденсатор требует периодической регенерации;

• триггер, обладая мизерной инертностью, без проблем работает на частотах вплоть до нескольких ГГц, тогда как конденсаторы "сваливаются" уже на 75-100 МГц.

Модули памяти

К недостаткам триггеров следует отнести их высокую стоимость и низкую плотность хранения информации. Если для создания ячейки динамической памяти достаточного всего одного транзистора и одного конденсатора, то ячейка статической памяти состоит как минимум из четырех, а в среднем шести – восьми транзисторов, поэтому статические ЗУ в 4...5 раз дороже динамических и приблизительно во столько же раз меньше по информационной емкости. Их достоинством является высокое быстродействие, а типичной областью применения – схемы кэш-памяти.

Модули памяти

В динамических ОЗУ (DRAM) данные хранятся в виде зарядов конденсаторов, образуемых элементами МОП-структур. Саморазряд конденсаторов ведет к разрушению данных, поэтому они должны периодически (каждые несколько миллисекунд) регенерироваться. В то же время плотность упаковки динамических элементов памяти в несколько раз превышает плотность упаковки, достижимую в статических RAM.

Регенерация данных в динамических ЗУ осуществляется с помощью специальных контроллеров. Разработаны также ЗУ с динамическими запоминающими элементами, имеющие внутреннюю встроенную систему регенерации, у которых внешнее поведение относительно управляющих сигналов становится аналогичным поведению статических ЗУ. Такие ЗУ называют квазистатическими.

Модули памяти

Динамические ЗУ характеризуются наибольшей информационной емкостью и невысокой стоимостью, поэтому именно они используются как основная память компьютеров.

Выбор статического ОЗУ в качестве модуля встроенной основной памяти, а не динамического, определяется возможностью хранения данных при снижении частоты вплоть до полной остановки процессора. Такой режим используется для энергосбережения, например, при питании от батарейки.

Модули памяти

На современном этапе модули встроенного ОЗУ не обладают значительным объемом – единицы иногда десятки килобайт. В случае мощных систем требующих больших объемов ОЗУ подключаются внешние микросхемы памяти. Это могут быть как микросхемы статического ОЗУ, так и динамическое ОЗУ (DRAM, SDRAM). В последнем случае процессор (например, AMD Am186ED, Mitsubishi M16C и др.) имеет модуль интерфейса динамического ОЗУ, который поддерживает интерфейс классических микросхем DRAM, или SDRAM, а так же регенерацию динамической памяти.

Модули памяти

Второй важной особенностью современных модулей ОЗУ – низкое, примерно 1 В, напряжение хранения информации. Это позволяет сохранять данные при провалах питания. С этой же целью иногда в модуль статического ОЗУ включают батарейку электропитания, позволяющую хранить данные до 10 лет (семейство DS5000 Dallas Semiconductor).

Энергонезависимая память E^2PROM: историческая справка

В 1974 году в Intel пришел Джордж Перлегос (George Perlegos), грек по происхождению и будущий основатель компании Atmel. Под его руководством в 1983 была разработана микросхема EEPROM (кодовое название 2816). Основой EEPROM стал транзистор с плавающим затвором, изобретенный в той же Intel Доном Фрохманом (Don Frohman). И в дальнейшем, несмотря на смены технологических эпох, принцип устройства ячейки энергонезависимой памяти остался неизменным – какой бы способ стирания и записи ни использовался.

Энергонезависимая память E^2PROM

Ячейка памяти представляет собой МОП-транзистор с плавающим затвором, который окружен диоксидом кремния. Сток транзистора соединен с «землей», а исток подключен к напряжению питания с помощью резистора. В стертом состоянии (до записи) плавающий затвор не содержит заряда, и МОПтранзистор закрыт. В этом случае на истоке поддерживается высокий потенциал, и при обращении к ячейке считывается логическая единица. Программирование памяти сводится к записи в соответствующую ячейки логических нулей. Программирование осуществляется путем подачи на управляющий затвор высокого напряжения. Этого напряжения должно быть достаточно, чтобы обеспечить пробой между управляющим и плавающим затвором, после чего заряд с управляющего затвора переносится на плавающий. МОП-транзистор переключается в открытое состояние, закорачивается исток с землей. В этом случае при обращении к ячейке считывается логический нуль. Такой метод записи называется «инжекцией горячих электронов», слой окисла между плавающим затвором и подложкой при этом составляет 50 нм. Стирание содержимого ячейки происходит путем электрического соединения плавающего затвора с «землей».

Энергонезависимая память E^2PROM

Таким образом, в EEPROM образца 1980-х запись производилась «горячей инжекцией», а стирание – «квантовым туннелированием». Оттого микросхемы эти были довольно сложны в эксплуатации и требовали два, а то и три питающих напряжения, причем подавать их при записи и стирании требовалось в определенной последовательности.

Позже, в электрически стираемой памяти Джордж Перлегос предложил использовать "квантовый эффект туннелирования Фаулера-Нордхейма". За этим непонятным названием кроется довольно простое по сути (но очень сложное с физической точки зрения) явление: при достаточно тонкой пленке изолятора (в пределах 10 нм) электроны, если их слегка подтолкнуть подачей не слишком высокого напряжения в нужном направлении, могут просачиваться через барьер, не перепрыгивая его.

Энергонезависимая память E^2PROM

Основным преимуществом использования памяти EEPROM заключается в возможности ее многократного перепрограммирования без удаления из платы. Такой способ программирования получил название «In-System Programming» или «ISP». При этом сокращаются затраты на программирование.

В процессе перезаписи, слой окисла постепенно накапливает захваченные электроны, которые со временем могут переходить в плавающий затвор. Тем самым уменьшается различие между пороговыми напряжениями соответствующими лог. «1» и «0». После достаточного количества циклов перезаписи, различие становится слишком маленьким для распознания. Как правило, минимальное число перезаписей доходит до сотен тысяч и миллиона раз.

Энергонезависимая память E^2PROM

Во время записи, электроны, инжектируясь в плавающий затвор, могут дрейфовать через изолятор, особенно при повышении температуры, и при этом возможна потеря заряда, то есть информация ячейки стирается. Производители обычно гарантируют, что данные будут храниться не менее 10 лет.

Основные характеристики EEPROM на примере AT24Cxx (Atmel)

Внутреннее ПЗУ 128x8 (1 K), 256x8 (2 K), 512x8 (4 K), 1024x8 (8 K) или 2048x8 (16 K).

• Двухпроводной последовательный интерфейс (I2C).

• Двунаправленный протокол передачи данных.

• Совместимость по частоте 100 кГц (1.8 V, 2.5 V, 2.7 V) и 400 кГц (5 V).

• Вывод защиты записи, обеспечивающий аппаратную защиту данных.

• Поддержка страничной записи в 8-байтном (1 K, 2 K), и 16-байтном (4 K, 8 K, 16 K) режимах.

• Поддержка неполной страничной записи.

• Самосинхронизирующийся цикл записи (максимум – 10 мс).

• Высокая надежность:

  ♦ Продолжительность работы 1 миллион циклов перезаписи;

  ♦ Сохранение данных в памяти в течение 100 лет.

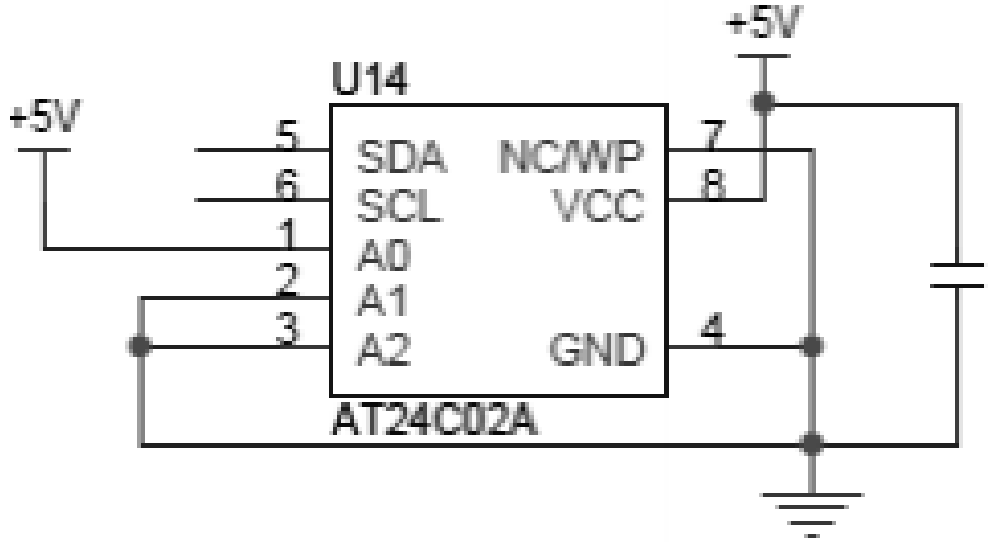
• Автоматическая градуировка и возможность работы в широком диапазоне температур.

Описание микросхемы AT24Cxx

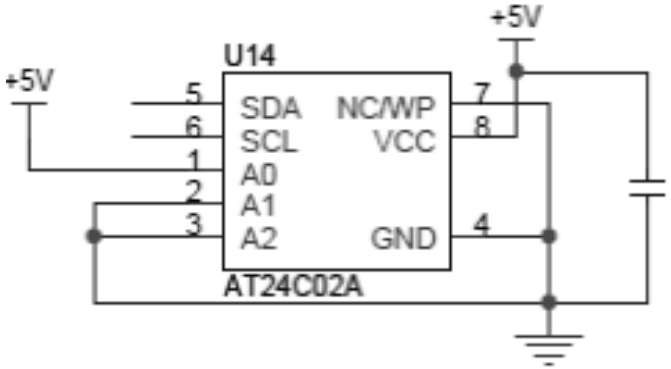
Микросхема AT24C01A / 02 / 04 / 08 / 16 – это 1024 / 2048/ 4096/ 8192/ 16384 бит последовательной памяти E2PROM (128 / 256 / 512 / 1024 / 2048 байт), которая может быть перезаписана с помощью электрических сигналов и считана программным путем. Данное устройство предназначено для применения в промышленных и коммерческих областях, где важным условием является низкое энергопотребление и малое напряжение питания.

Назначение выводов микросхемы

Вывод	Назначение
A0 – A2	Адресные входы.
SDA	Последовательная передача данных.
SCL	Линия синхронизации.
WP	Защита от записи.
NC	Не используется



Назначение выводов микросхемы



SERIAL CLOCK (SCL) – линия синхронизации.

Вход SCL используется при передаче в E2PROM (положительный фронт) и отправке данных на любое внешнее устройство (отрицательный фронт).

SERIAL DATA (SDA) – линия последовательной передачи данных.

SDA – вывод для двунаправленной последовательной передачи данных.

Это вывод со свободным стоком, к нему можно подключать любое количество открытых коллекторов или коллекторов со свободным стоком.

Выводы A2, A1 и A0 – это адресные входы устройств, разработанные для микросхем AT24C01A и AT24C02. К одной шине может быть подключено до 8

1К/2К – устройств.

Микросхема AT24C04 использует для фиксированной адресации два вывода A2 и A1, что позволяет подключить к одной шине до четырех таких микросхем. Вывод A0 не используется.

Микросхема AT24C08 использует для фиксированной адресации только вывод A2, что позволяет подключить к одной шине до двух таких микросхем.

Выводы A0 и A1 не используются.

Микросхема AT24C16 не использует адресные выводы. К одной шине можно подключить только одно устройство. Выводы A0, A1 и A2 не используются.

Защита от записи в схемах семейства AT24C01A/02/04/16

Состояние вывода	Защита массива данных				
	24C01A	24C02	24C04	24C08	24C16
Напряжение	Полностью (1К)	Полностью (2К)	Полностью (4К)	Обычные операции чтения/записи	Верхняя половина массива (8К)
Земля	Обычные операции чтения/записи				

Организация памяти

AT24C01A, 1K последовательная E²PROM. Внутренняя память, состоящая из 128 однобайтовых страниц общим объемом в 1 К, для произвольного доступа к которой требуются 7-битные адреса.

AT24C02, 2K последовательная E²PROM. Внутренняя память, состоящая из 256 однобайтовых страниц общим объемом в 2К, для произвольного доступа к которой требуются 8-битные адреса.

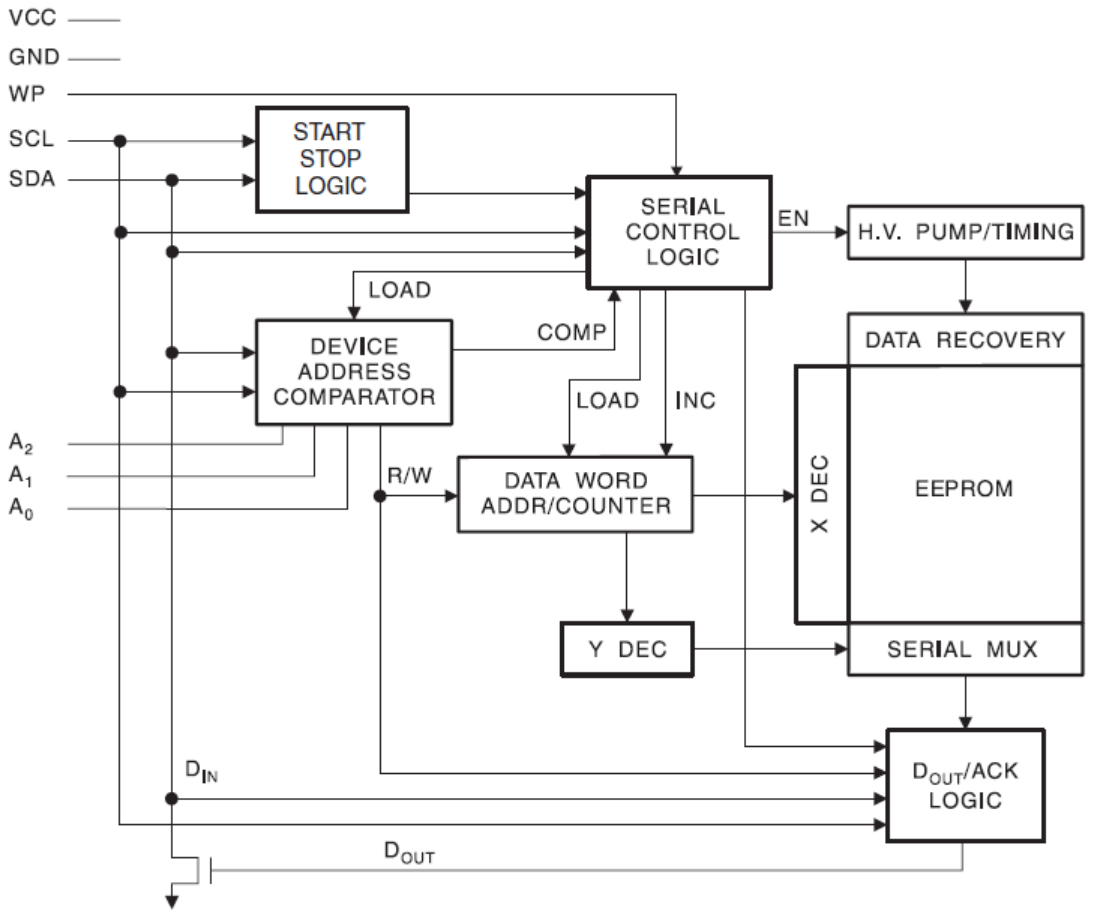
AT24C04, 4K последовательная E²PROM. Внутренняя память объемом в 4К, состоящая из 256 страниц по 2 байта каждая. Для произвольного доступа к данным требуются 9-битные адреса.

AT24C08, 8K последовательная E²PROM. Внутренняя память объемом в 8К, состоящая из 4 блоков. Каждый блок содержит 256 4-байтных страниц. Для произвольного доступа к данным требуется 10-битная адресация.62

AT24C16, 16K последовательная E²PROM. Внутренняя память объемом в 16К, состоящая из 8 блоков. Каждый блок содержит 256 8-байтных страниц. Для произвольного доступа к данным необходима 11-битная адресация.

Структурная схема памяти EEPROM

Обозначения на схеме:
• Start-stop logic – логика «пуск-останов».
• Device Address Comparator – блок сравнения адреса.
• Serial Control Logic – последовательная логика управления.
• Data Word Addr/Counter – адрес/счетчик слова данных.
• H.V. Pump/Timing – генератор подкачки заряда / тактирование.
• Data Recovery – восстановление данных.
• Serial Mux – мультиплексор последовательной передачи.
• Dout/ACK Logic – логика подтверждения передачи сигнала.



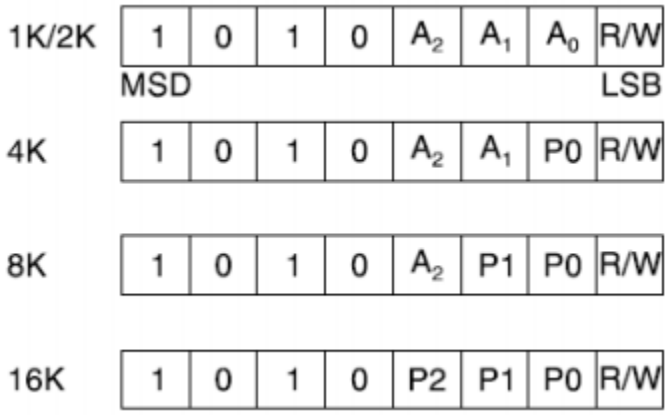
Адресация модулей EEPROM

Микросхемы E^2PROM с объемом памяти 1 К, 2 К, 4 К, 8 К и 16 К после перехода в старт-состояние должны получать слово (8 бит) с адресом устройства. Только тогда микросхема сможет произвести операцию чтения или записи

1K/2K	1	0	1	0	A_2	A_1	A_0	R/W
	MSD							LSB
4K	1	0	1	0	A_2	A_1	P0	R/W
8K	1	0	1	0	A_2	P1	P0	R/W
16K	1	0	1	0	P2	P1	P0	R/W

Адреса устройств

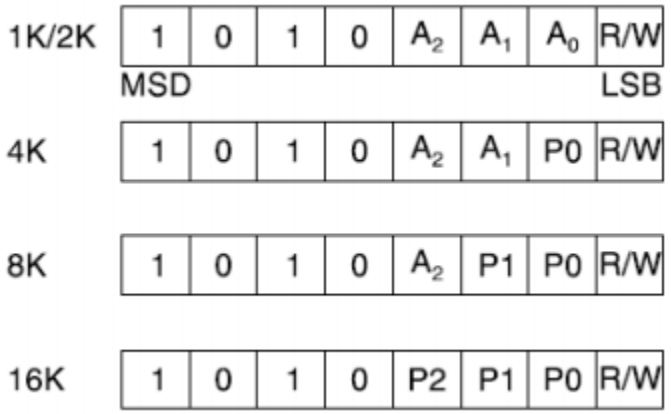
Адресация модулей EEPROM



Первые четыре бита слова адреса представляют собой обязательную последовательность «1010». Данная последовательность одинакова для всех устройств E^2PROM (данной серии, производства Atmel). Следующие 3 бита представляют собой адреса устройств A2, A1 и A0 для 1K/2K E^2PROM. Эти биты соответствуют входам с аналогичными названиями.

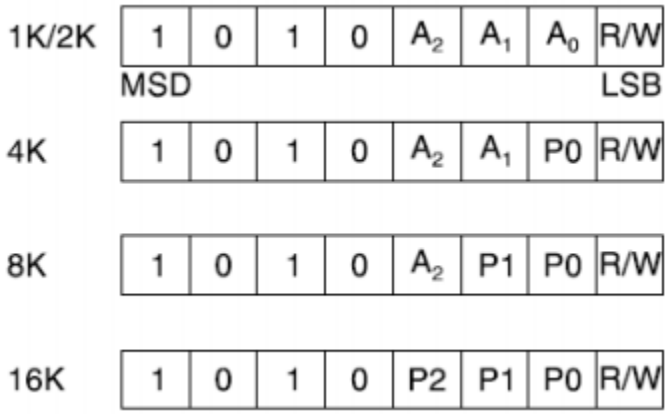
E^2PROM с объемом памяти 4 К использует только биты адресов A2 и A1, а P0 представляет собой адрес страницы памяти. Оба бита адресов устройств соответствуют выходам на микросхеме с аналогичными названиями. Вывод A0 не подключен.

Адресация модулей EEPROM



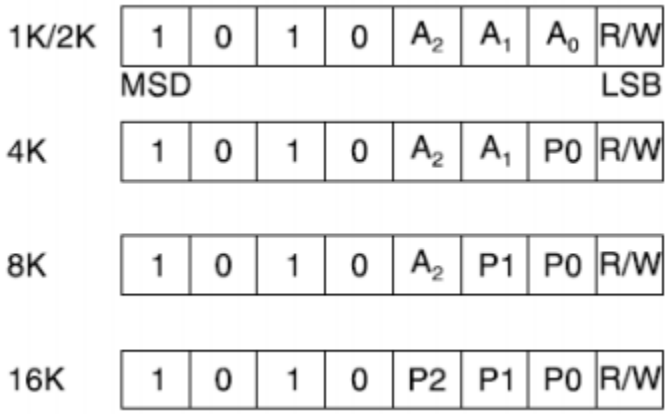
E2PROM с объемом памяти 4 К использует только биты адресов A2 и A1, а P0 представляет собой адрес страницы памяти. Оба бита адресов устройств соответствуют выходам на микросхеме с аналогичными названиями. Вывод A0 не подключен.

Адресация модулей EEPROM



Адресный байт для E2PROM с объемом памяти 8 К содержит только один бит адреса устройства A2, а биты P1 и P0 используются для адресации страницы памяти. Бит A2 соответствует выводу A2 на микросхеме. Выводы A1 и A0 не подключены.

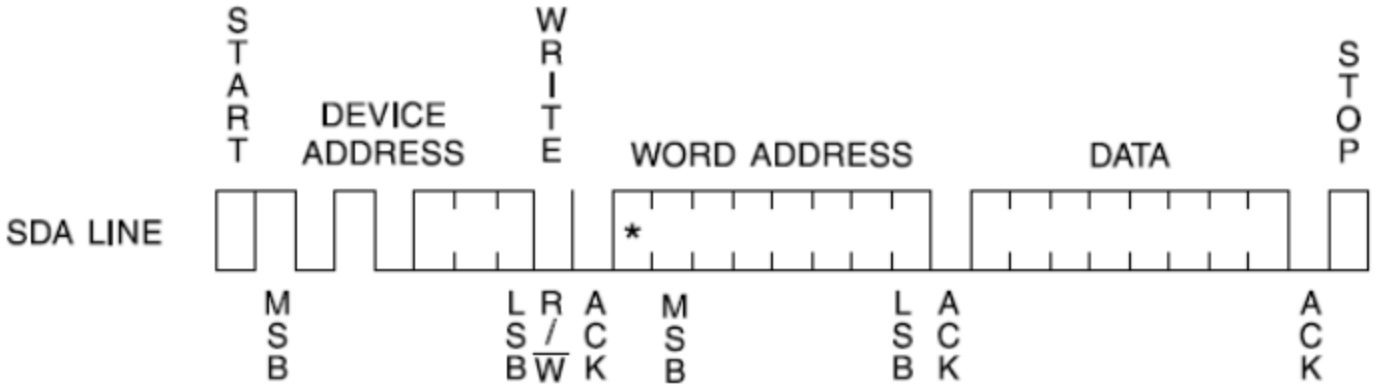
Адресация модулей EEPROM



В E2PROM с 16 К памяти биты P0, P1 и P2 представляют собой адрес страницы памяти в устройствах 4 К, 8 К и 16 К. Выводы A0, A1 и A2 не подключены.

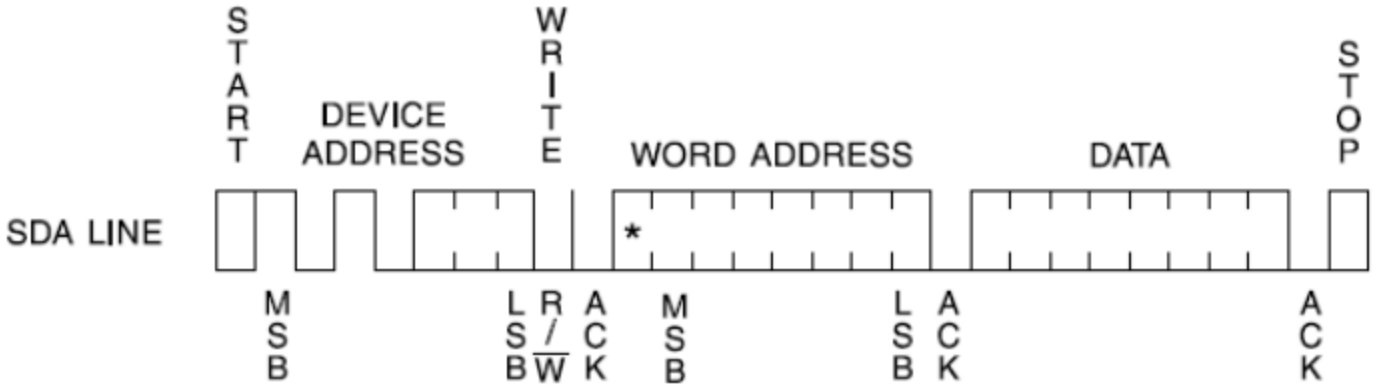
Восьмой бит адреса устройств используется для выбора режима чтения/записи. Если бит равен 1, происходит чтение, иначе – запись. После сравнения адресов устройств E2PROM выдает 0. Если сравнение не было произведено, микросхема возвращается в режим ожидания.

Операция записи. Запись байта



После того, как E2PROM получит адрес ячейки и подтвердит возможность приема, должна происходить операция записи. Получив адрес и ответив выдачей «0», устройство примет первые 8 бит данных. Затем E2PROM выдает «0». Адресующее устройство (например, микроконтроллер) должно остановить процесс записи путем выдачи стоп-сигнала. В этот момент E2PROM начинает цикл записи в постоянную память. До тех пор, пока запись не будет завершена, отключаются все входы и E2PROM не реагирует ни на какие сигналы.

Операция записи. Запись байта



Список обозначений:

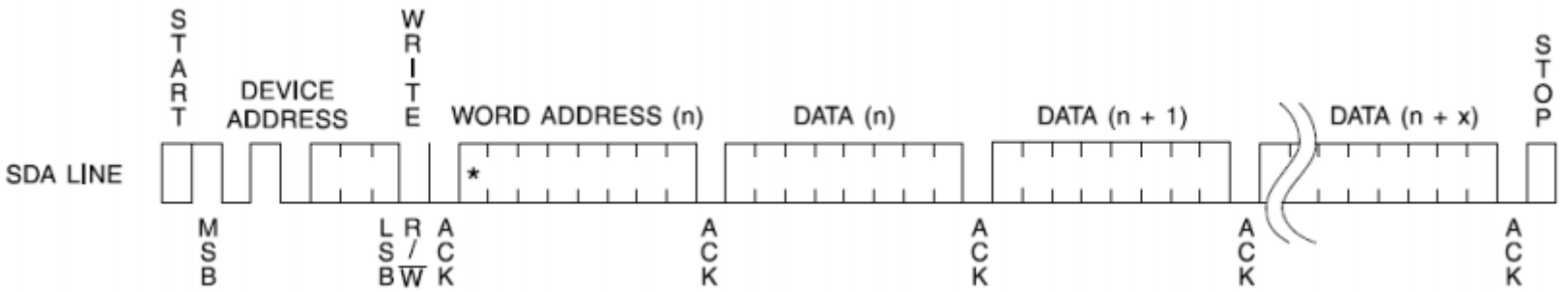
- WRITE – запись;
- ACK – сигнал подтверждения;
- SDA LINE – линия передачи данных SDA;
- DATA – данные;
- DEVICE ADDRESS – адрес устройства;
- WORD ADDRESS – адрес ячейки памяти.

Страничная запись

1K/2K E2PROM может производить страничную запись (по 8 байт), а устройства с объемом памяти в 4 K, 8 K и 16 K производят 16-байтную запись.

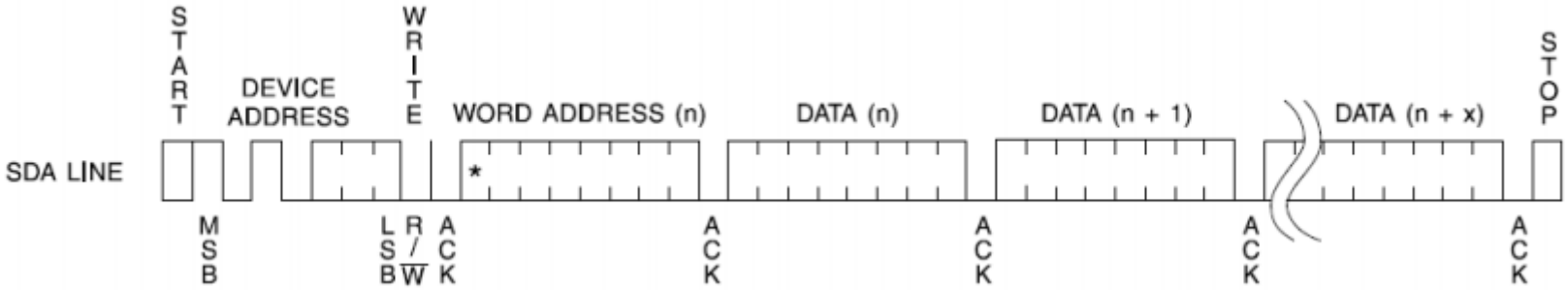
Процесс страничной записи инициируется так же, как запись одного байта, отличие в том, что микроконтроллер после передачи первого слова не выдает стоп-сигнал.

Страничная запись



Вместо этого, как только E2PROM подтвердит получение первого слова данных, микроконтроллер может передать ему еще до 7 (1 К / 2 К) или 15 (4 К, 8 К, 16 К) слов данных. После получения каждого слова E2PROM будет выдавать на линии «0» (подтверджение). Микроконтроллер прекращает страничную запись, выдавая стоп-сигнал.

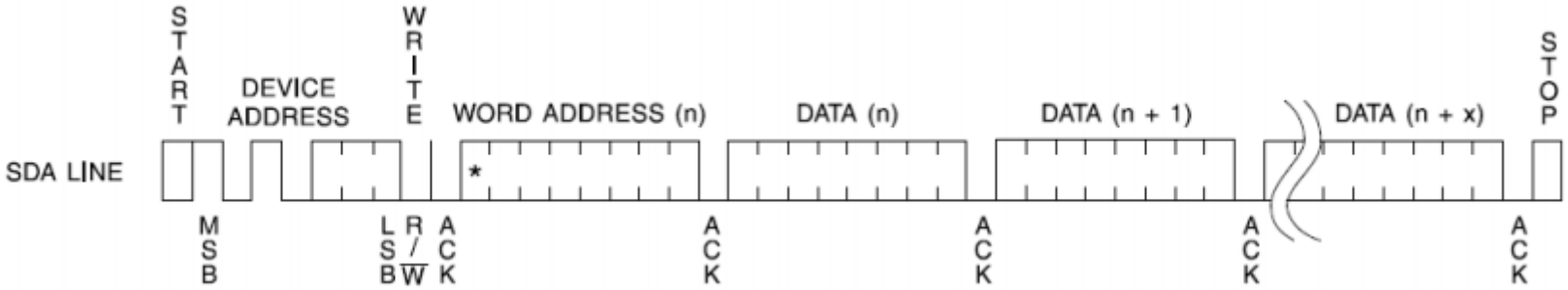
Страничная запись



Список обозначений:

- WRITE – запись;
- ACK – сигнал подтверждения;
- SDA LINE – линия передачи данных SDA;
- DATA – данные;
- DEVICE ADDRESS – адрес устройства;
- WORD ADDRESS – адрес ячейки памяти.

Страничная запись



Каждый раз при получении слова данных E2PROM инкрементирует младшие 3 (1 K / 2 K) или 4 (4 K, 8 K, 16 K) адресных бита. Старшие адресные биты не инкрементируются.

Во время записи последовательности байт счетчик адреса «перепрыгивает» с последнего байта текущей страницы на первый байт той же самой страницы.

Опрос устройства

Как только E2PROM начнет внутренне тактируемый цикл записи и отключит свои входы, можно инициировать запрос на подтверждение получения данных. Этот процесс включает отправку слова с адресом устройства, а затем выдачу стоп-сигнала. Бит чтения/записи устанавливается в зависимости от требуемой операции. E2PROM выставит «0» – это позволит продолжить запись или чтение, только после завершения своего внутреннего цикла.

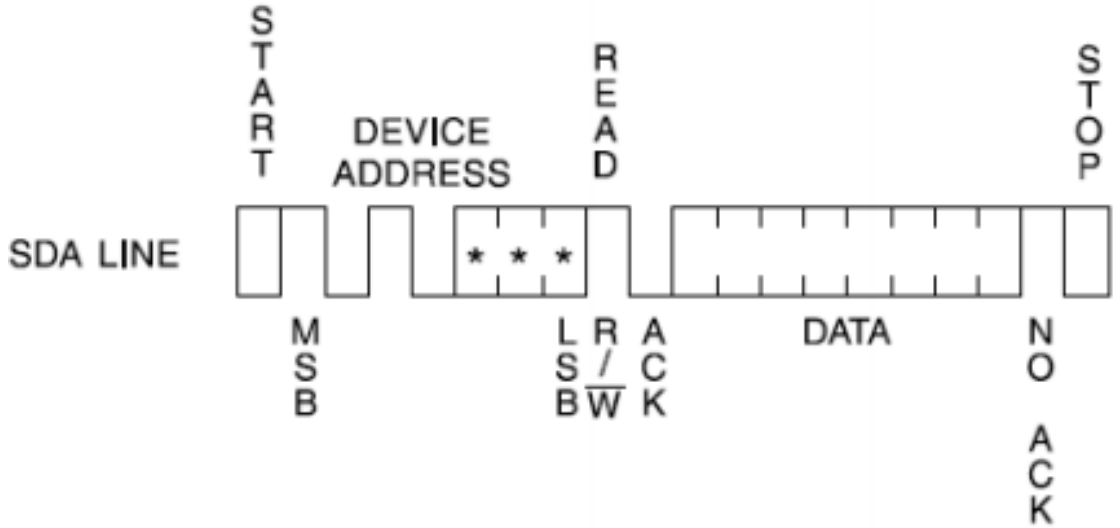
Операция чтения

Операция чтения инициируется точно так же, как и операция записи, за тем исключением, что бит чтения/записи в слове адреса устройства устанавливается равным 1. Существует три операции чтения: чтение с текущего адреса, чтение в режиме произвольного доступа и последовательное чтение.

Чтение с текущего адреса

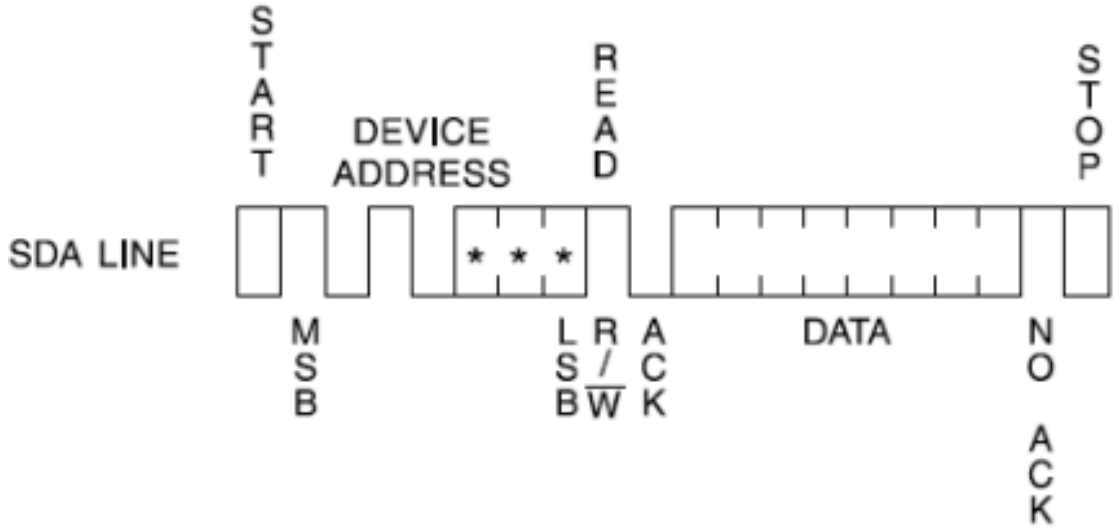
Внутренний счетчик адреса содержит последний адрес, к которому производилось обращение во время операции чтения или записи, увеличенный на единицу. Этот адрес остается корректным в промежутке между операциями до тех пор, пока микросхема работает (питание включено). Во время чтения счетчик адреса «перепрыгивает» с последнего байта последней страницы памяти на первый байт первой страницы.

Чтение с текущего адреса



После передачи микроконтроллером байта адреса устройства с битом чтения/записи, установленным в «1», и подтверждения его модулем E2PROM, следующим микроконтроллеру передается байт данных по текущему адресу памяти. Микроконтроллер в ответ сигнализирует об окончании чтения: посылает NO ACK и стоп-сигнал

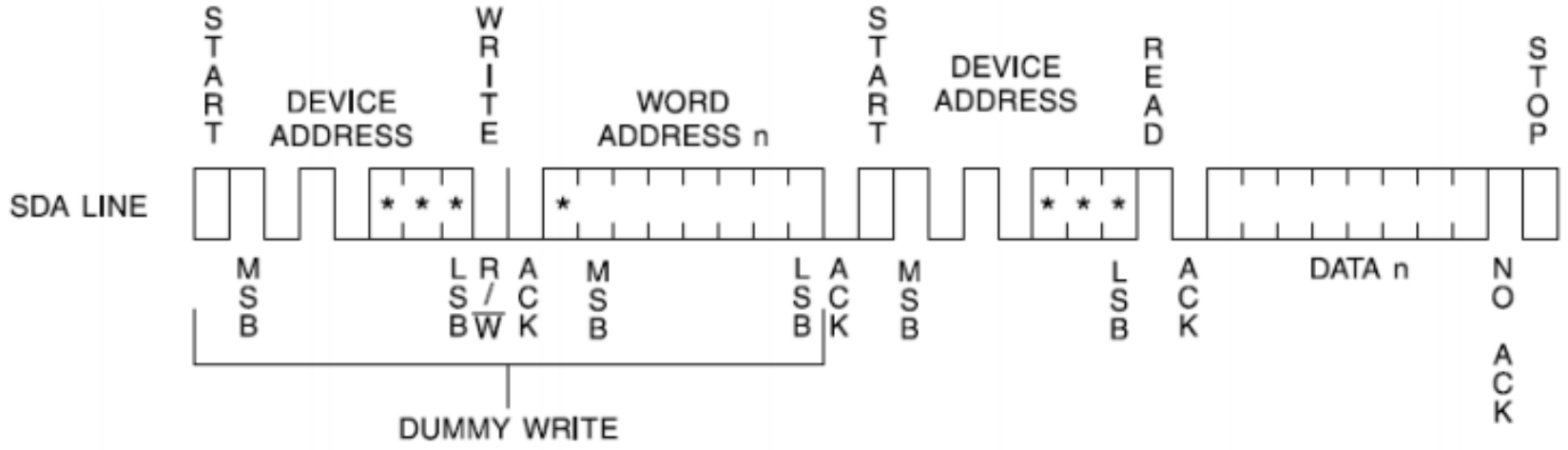
Чтение с текущего адреса



Список обозначений:

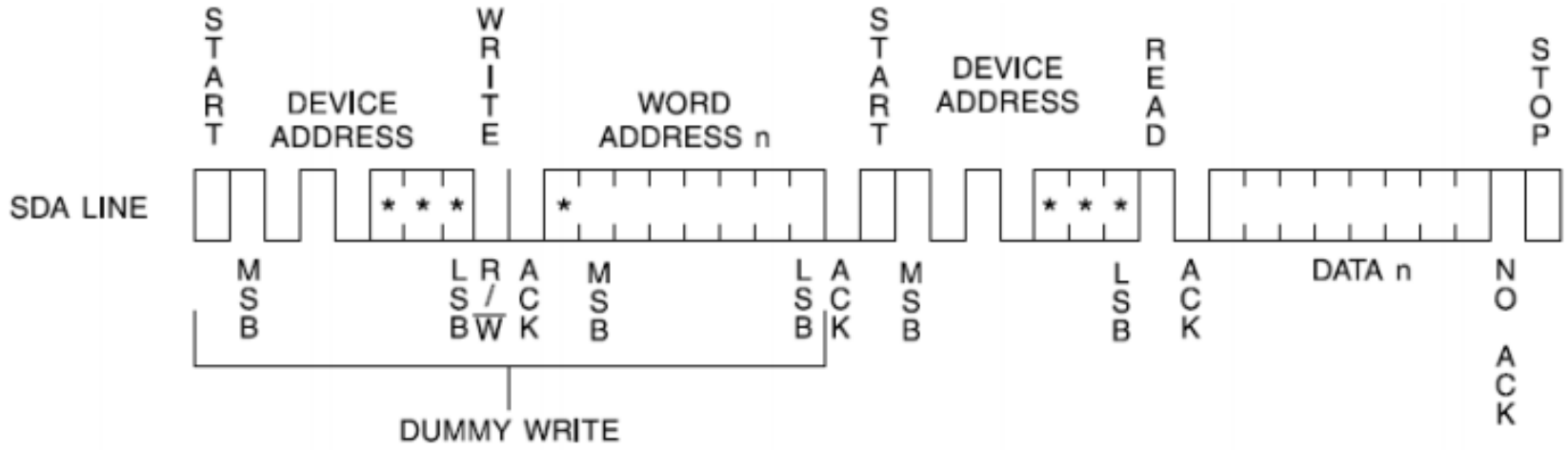
- READ – чтение;
- ACK – сигнал подтверждения;
- NO ACK – отсутствие подтверждения;
- SDA LINE – линия передачи данных SDA;
- DATA – данные;
- DEVICE ADDRESS – адрес устройства.

Чтение в режиме произвольного доступа



*Эти биты для памяти EEPROM емкостью 1 К могут быть любыми.

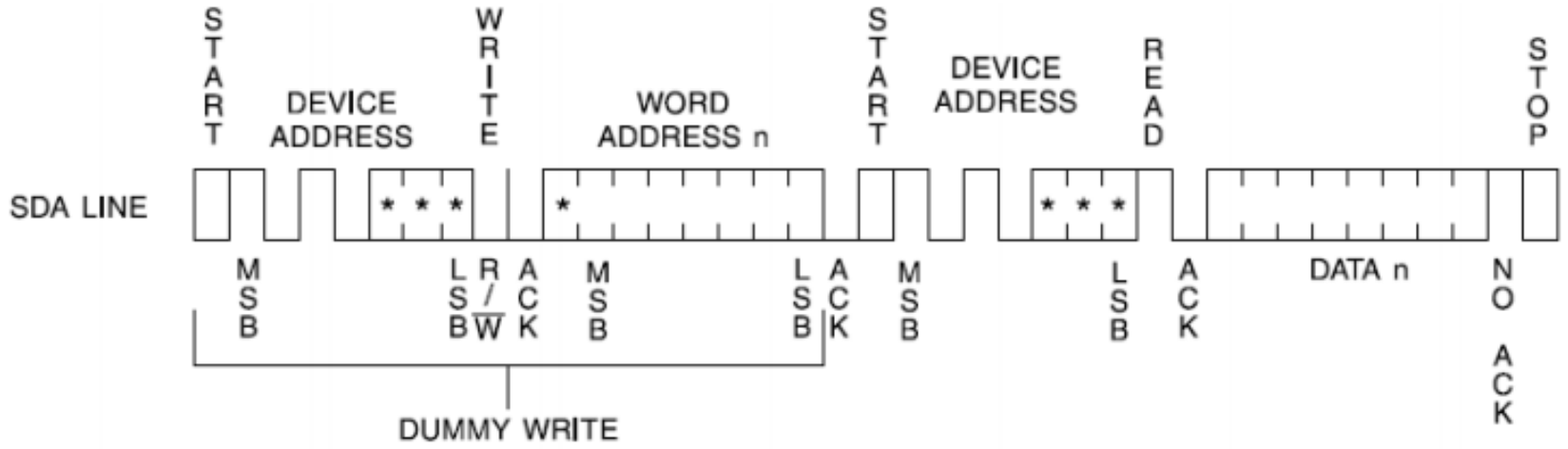
Чтение в режиме произвольного доступа



Список обозначений:

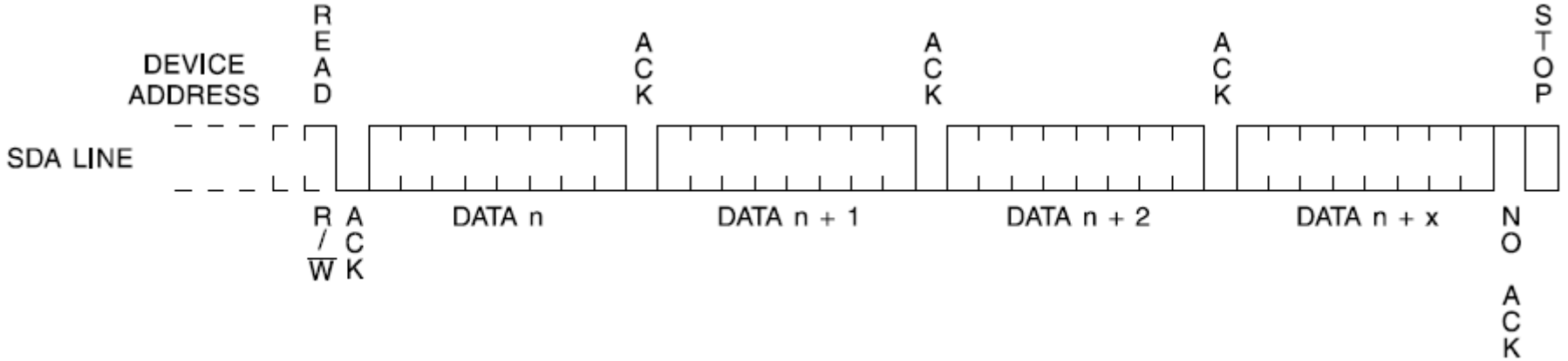
- READ – чтение;
- WRITE – запись;
- ACK – сигнал подтверждения;
- SDA LINE – линия передачи данных SDA;
- DATA – данные;
- DEVICE ADDRESS – адрес устройства;
- WORD ADDRESS – адрес ячейки памяти;
- DUMMY WRITE – холостая запись (установка счетчика адреса).

Чтение в режиме произвольного доступа



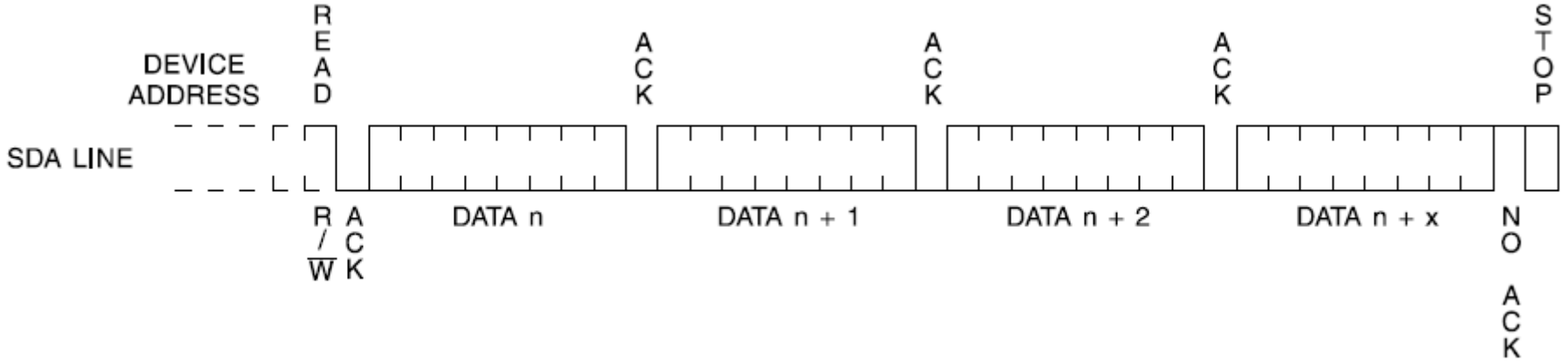
Как только слово адреса устройства и адрес слова данных будут приняты E2PROM, микроконтроллер должен сгенерировать еще один старт-сигнал. После чего начинается чтение с текущего адреса, только что установленного.

Чтение в режиме последовательного доступа



Последовательное чтение данных инициируется в процессе либо чтения с текущего адреса, либо чтения в режиме произвольного доступа – чтением последовательности байтов данных из EEPROM. Когда счетчик адреса достигнет последнего байта последней страницы памяти, он «перепрыгнет» на первый байт первой страницы. Операция последовательного чтения будет остановлена в том случае, если микроконтроллер сигнализирует об окончании чтения: посылает NO ACK и стоп-сигнал.

Чтение в режиме последовательного доступа



Список обозначений:

- READ – чтение;
- ACK – сигнал подтверждения;
- NO ACK – отсутствие подтверждения;
- SDA LINE – линия передачи данных SDA;
- DATA – данные;
- DEVICE ADDRESS – адрес устройства.

Порты ввода-вывода

Порты ввода-вывода

Каждый процессор для встраиваемых применений имеет некоторое количество внешних линий ввода-вывода, подключенных к внешним выводам микросхемы и называемых внешними портами. Одиночные (одноразрядные, состоящие из одной линии) порты ввода-вывода объединяются в группы, обычно, по 4, 8 или 16 линий, которые называются параллельными портами. Разрядность параллельных портов может быть нестандартной, например, 5-разрядный порт у микроконтроллера PIC16F84.

Порты ввода-вывода

Через порты процессорное ядро взаимодействует с различными внешними устройствами – считывает значения входных сигналов и устанавливает значения выходных сигналов.

Во встраиваемых системах в качестве внешних устройств чаще всего рассматриваются датчики, исполнительные устройства, устройства ввода-вывода данных оператором, устройства внешней памяти.

Порты ввода-вывода

По типу сигнала различают порты:

1. Дискретные (цифровые) – используются для ввода-вывода дискретных значений логического «0» или «1».

В большинстве современных процессоров для встраиваемых применений поддерживается как независимое управление каждой линией параллельного порта, так и групповое управление всеми разрядами. Так как схемотехника отдельных линий в рамках одного 4-х, 8-ми или 16-разрядного порта одинакова, то дальше будет рассматриваться устройство и функционирование одиночного разряда.

Порты ввода-вывода

2. Аналоговые – через них вводятся сигналы на вход АЦП или других аналоговых схем и выводятся выходные сигналы ЦАП или других аналоговых схем.

Аналоговые порты (или перестраиваемые порты в аналоговом режиме) – используются подключения внешних сигналов к ЦАП, АЦП или аналоговым компараторам, встроенным приемопередатчикам. В режиме работы с ЦАП, АЦП или компаратором порты обычно позволяют вводить сигнал в диапазоне от 0В- до Uпит+ (индексы + и – означают чуть больше и чуть меньше, примерно на 200..300мВ).

Порты ввода-вывода

В режиме приемопередатчика параметры сигналов определяются конкретным интерфейсом. В большинстве случаев аналоговые или цифровые линии подключения к приемопередатчикам вообще не называют портами, хотя они по схемотехнике и по месту в структуре процессора близки к универсальным портам ввода-вывода. Реализация входных и выходных каскадов зависит от схемы АЦП, компаратора, ЦАП или приемопередатчика.

3. Перестраиваемые – настраиваются на аналоговый или цифровой режим работы.

Порты ввода-вывода

По направлению передачи сигнала различают:

1. Однонаправленные порты, предназначенные только для ввода (входные порты, порты ввода) или только для вывода (выходные порты, порты

вывода).

2. Двунаправленные порты, направление передачи которых определяется в процессе программно-управляемой настройки схемы.

3. Порты с альтернативной функцией. Отдельные линии этих портов связаны со встроенными периферийными устройствами, такими, как таймер, контроллеры последовательных приемопередатчиков. Если соответствующий периферийный модуль не задействован, то линии можно использовать как обычные порты, если модуль активизирован, то связанные с ним линии автоматически или «вручную» (программно) конфигурируются в соответствии с функциональным назначением и не могут быть использованы в качестве универсальных портов ввода-вывода. В некоторых случаях порты могут использоваться только для связи с периферийным модулем (например, входы АЦП в некоторых процессорах)

Порты ввода-вывода

По алгоритму обмена различают порты:

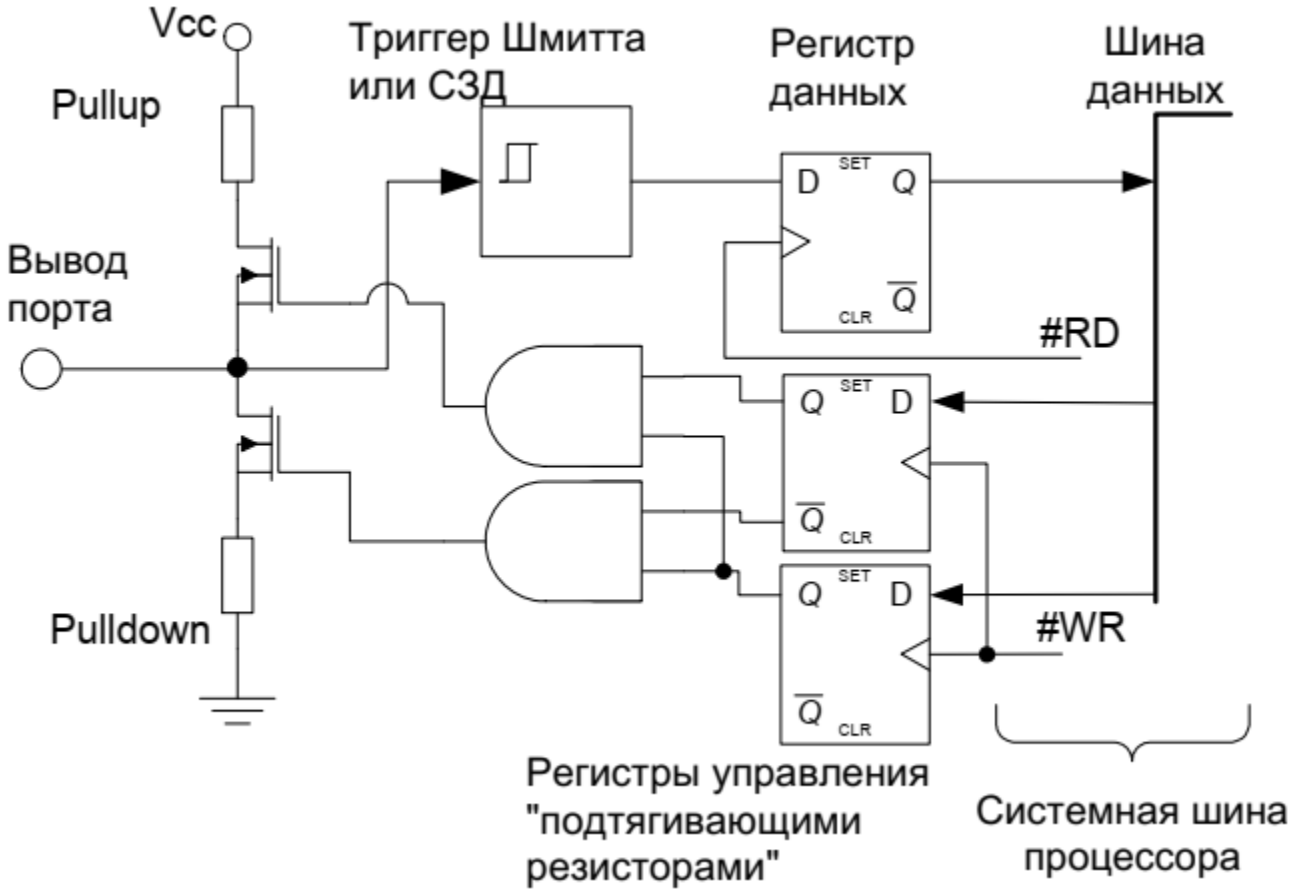
1. С программно управляемым (программным) вводом-выводом – установка и считывание данных определяется только ходом вычислительного процесса. Нет защиты от повторного считывания-записи одного и того же (не изменившегося) значения на выводе и считывания-записи во время переходного процесса на выводе.

2. Со стробированием – каждая операция ввода вывода подтверждается импульсом синхронизации (стробом) со стороны источника сигнала (при выводе – процессор, при вводе – внешнее устройство). Считывание информации приемником происходит только по стробу, что позволяет защититься от приема данных во время переходного процесса входного сигнала. Пример: порт PSP (Parallel slave port) в ОКМЭВМ PICmicro.

Порты ввода-вывода

3. С полным квитированием. Данный режим чаще всего используется для обмена данными с другой вычислительной системой по параллельной шине. Кроме сигналов синхронизации со стороны передатчика используются сигналы подтверждения (готовности к следующему обмену) со стороны приемника. Это позволяет управлять интенсивностью обмена обоим взаимодействующим сторонам и предотвращает потерю данных, когда одна из них перегружена. Пример порта с квитированием – порт LPT персонального компьютера. Во встроенных модулях процессоров данный режим чаще всего реализуется программно-аппаратно.

Однонаправленные порты



Однонаправленный порт ввода

Однонаправленные порты

Внешние данные считываются через вывод порта (ножку микросхемы), проходят через триггер Шмитта (ТШ) или схему защиты от дребезга (СЗД) и по внутреннему сигналу чтения фиксируются в регистре данных, с выхода которого, в свою очередь, данные считываются процессором.

ТШ (используется в большинстве процессоров для встроенного применения) имеет гистерезис по уровню входного напряжения и предотвращает многократное переключение входных схем при пологом фронте сигнала или помехах.

СЗД (например, в семействе Zilog Z8) вводит инерционность переключения и отсекает реакцию на короткие по длительности импульсы. Используется для защиты от помех.

Однонаправленные порты

К входу также могут подключаться так называемые «резисторы поддержки» логической «1» (Pull-up) или логического «0» (Pull-down). Эти резисторы предназначены для переведения входов в устойчивое состояние «0» или «1» и предотвращения произвольных переключений от помех в моменты, когда на них (входы) не подается внешний сигнал, например, неиспользуемых и неподключенных к внешним схемам входов («открытых входов»). Через специальные управляющие регистры «схемы поддержки» могут быть отключены полностью или включены в режим Pull-up или Pull-down.

Все перечисленные блоки – триггер Шмитта, СЗД и «схемы поддержки» используются для защиты от случайных переключений в результате помех и помогают снизить энергопотребление, которое резко возрастает в момент переключений входных схем.

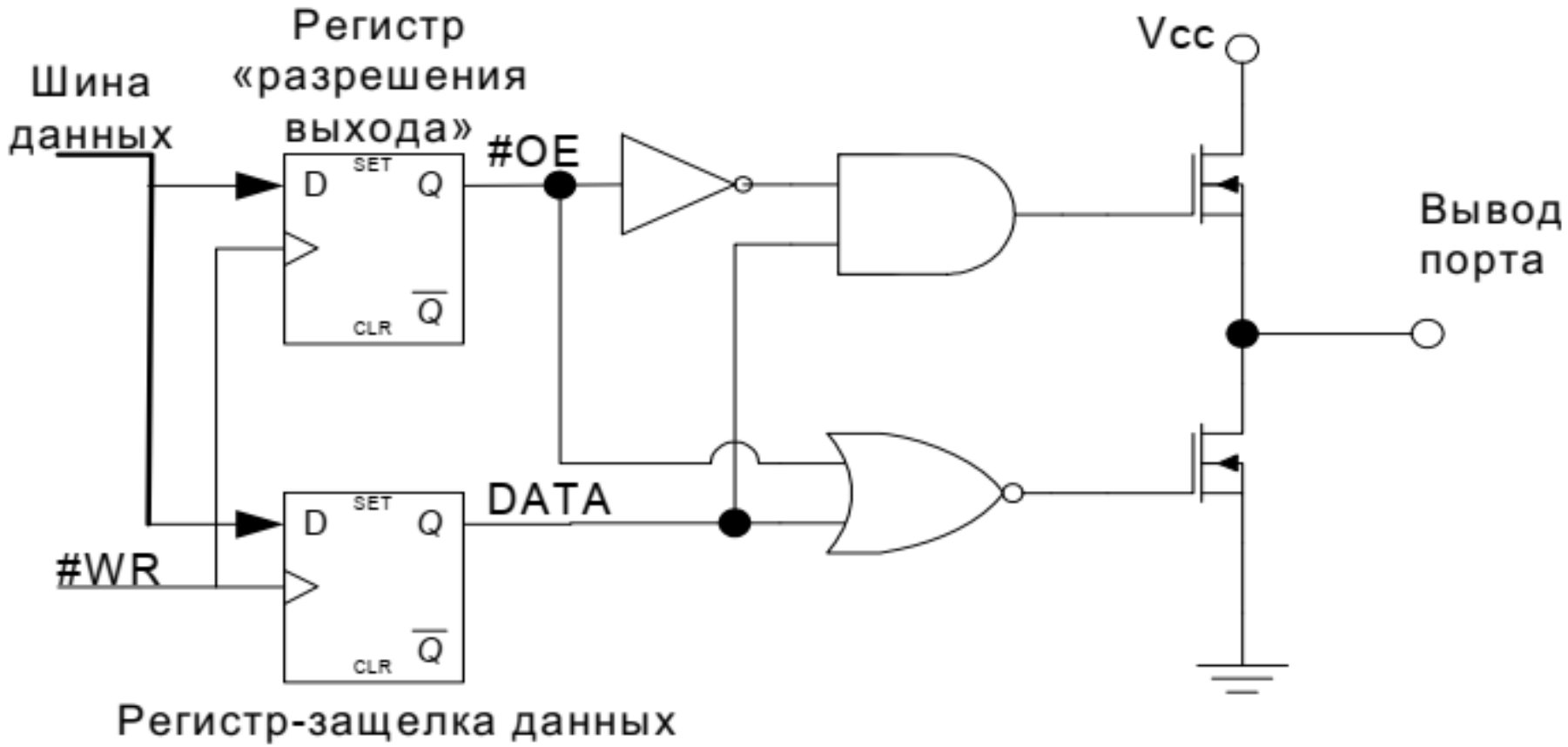
Однонаправленные порты

Порты вывода бывают:

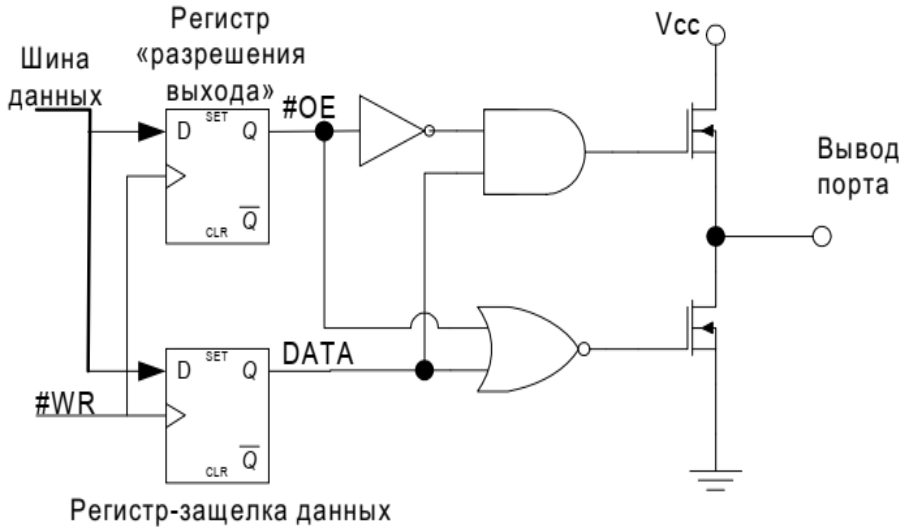
- с двухтактной выходной схемой (комплементарные);
- с однотактной выходной схемой и внутренней нагрузкой;
- с открытым выходом (открытым коллектором или стоком).

Порты вывода с двухтактной выходной схемой являются самыми распространенными и реализованы, например, в семействах Atmel AVR, Microchip PICmicro, AMD AM186, Motorola HC08, HC11 и многих других.

Порт вывода с двухтактной схемой



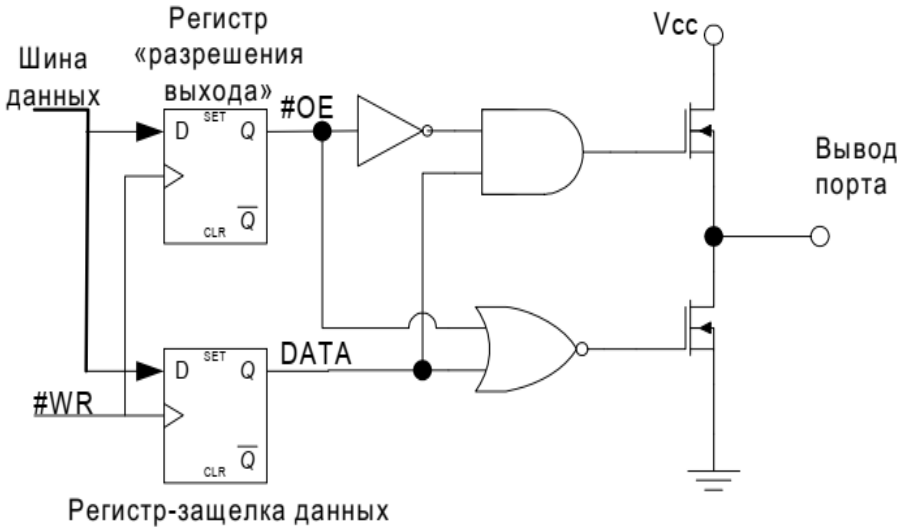
Порт вывода с двухтактной схемой



Рассмотрим функционирование данной схемы.

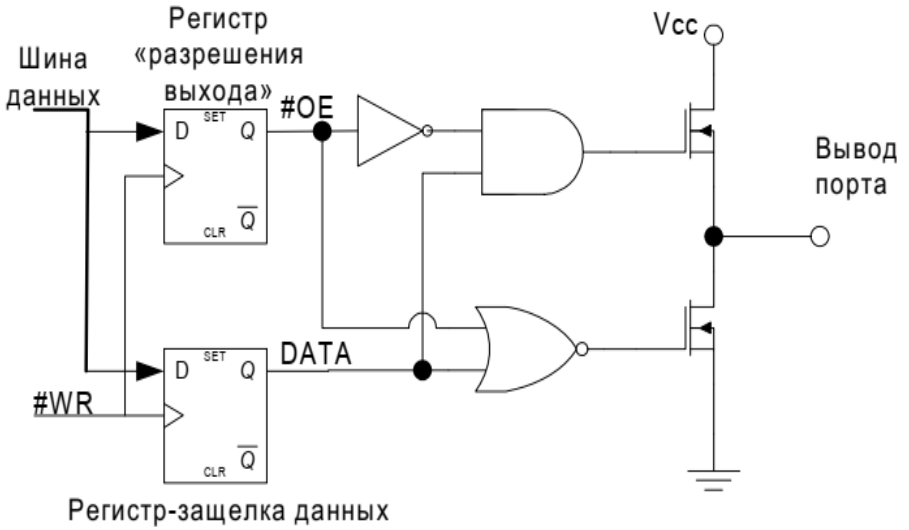
Выходные данные записываются в регистр-защелку данных по внутреннему сигналу записи #WR и через простейшую логическую схему управляют выходными транзисторами. Если в регистр записано значение «1», то открыт верхний по схеме транзистор, а нижний закрыт: на выходе Vcc, то есть «1». Если в регистр записано значение «0», то открыт нижний по схеме транзистор, а верхний закрыт: выход соединен с минусовой шиной питания, то есть там установлен «0».

Порт вывода с двухтактной схемой



Верхний по схеме регистр управляет сигналом #OE - «разрешение выходов». Если в регистр записан «0», то схема работает, как было описано выше. Если записана «1», то оба транзистора закрываются и схема переводится в «высокоомное» состояние (Z-состояние). В этом состоянии выходное сопротивление порта очень высокое и он фактически «оторван» от микропроцессора.

Порт вывода с двухтактной схемой



Это необходимо:

• Если к выходному порту подключены выходы других схем и необходимо разделять линии передачи данных с этими устройствами.

Например: наш процессор используется как периферийный контроллер и его выходной порт подключен к периферийной шине другого процессора (мастера), и к шине также подсоединены еще несколько периферийных контроллеров;

• В схемах двунаправленных портов

Порт вывода с двухтактной схемой

Достоинства:

Значительный максимальный втекающий (в состоянии «0») и вытекающий (в состоянии «1») ток выхода: 2..6mA для каскадов с нормальной нагрузочной способностью (например, Fujitsu MB90) и 5..30mA для каскадов с повышенной нагрузочной способностью (например, PICmicro, AVR). Встречаются отдельные микросхемы со сверхвысокой нагрузочной способностью – до 60..90mA (например, PIC17). Большой выходной ток позволяет непосредственно с ножки, без схем усиления и согласования сигнала, управлять достаточно мощной нагрузкой: светодиодами, реле, мощным электронным ключем (транзистор, тиристор). Это значительно упрощает схему устройств.

Порт вывода с двухтактной схемой

Недостатки:

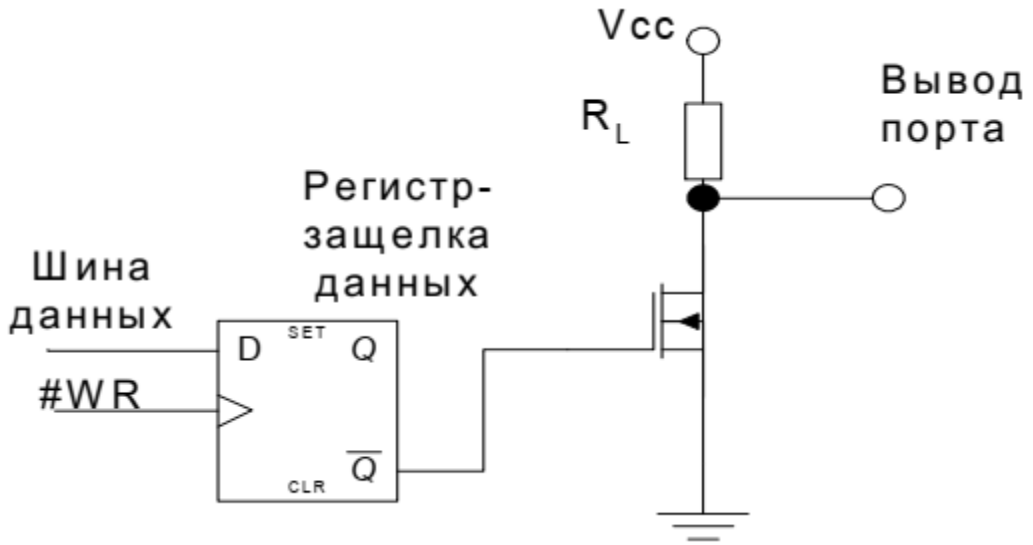
• При программировании необходимо управлять дополнительным битовым регистром «разрешение выхода»;

• Значительное энергопотребление и уровень помех при переключении. Последний особо зависит от скорости переключения. Для ограничения токов в момент переключений иногда используют специальные демпфирующие схемы. Однако они снижают быстродействие портов. Наибольшее применение демпфирующие схемы находят в портах ПЛИС в силу их особо высокого быстродействия;

• Относительно сложная внутренняя схема повышающая сложность и стоимость микросхемы в целом. Однако на нынешнем этапе, в связи с успехами технологии производства микросхем, это уже не является проблемой.

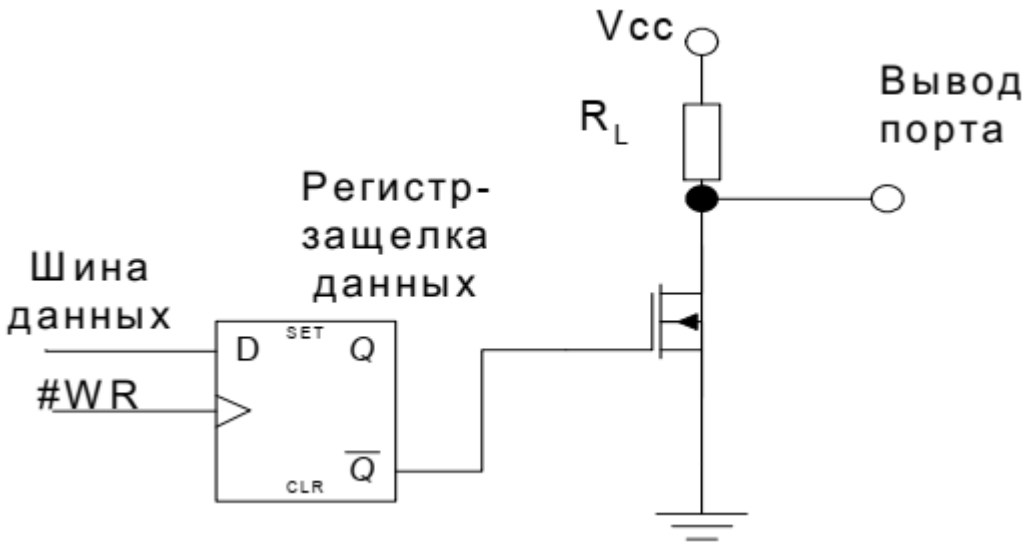
Порты вывода с однотактной выходной схемой и внутренней нагрузкой

Порты вывода с однотактной выходной схемой и внутренней нагрузкой применяются, например, в семействе MCS-51. Они имеют более простую внутреннюю схему.



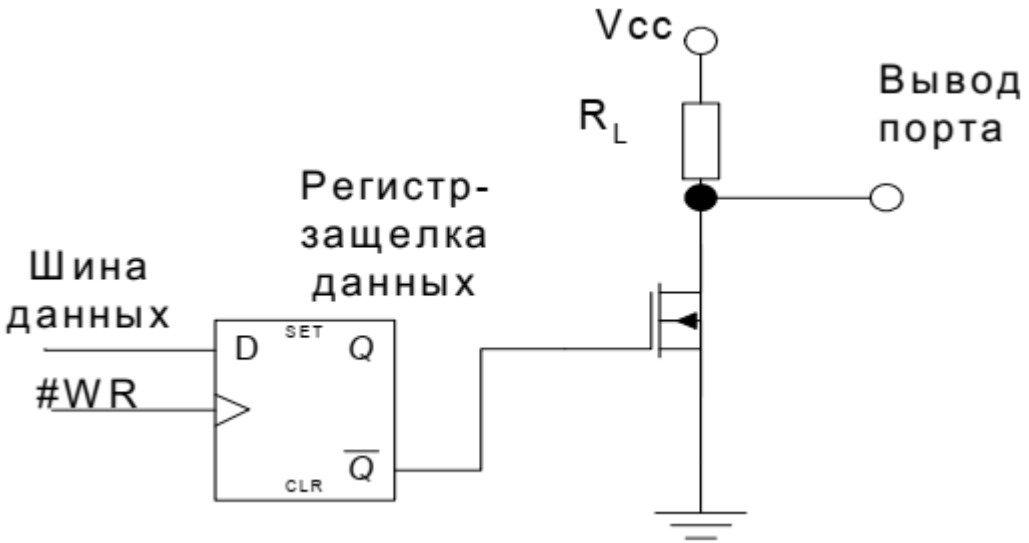
Порт вывода с однотактной схемой

Порты вывода с однотактной выходной схемой и внутренней нагрузкой



Когда в регистр-защелку записано значение «1», транзистор закрыт и на выходе через резистор R_L устанавливается Vcc – логическая «1». Когда же в регистр-защелку записан «0», открывается транзистор и соединяет выход с минусовой шиной питания, то есть там устанавливается «0». При этом резистор R_L оказывается подключенным между шинами питания. Во избежания высокого тока через резистор и его перегрева, сопротивление делают достаточно высоким – 10..100кОм. Высокое сопротивление резистора позволяет непосредственно соединять несколько выходов, не опасаясь их встречного включения, так как если «0» на одном из выходов «подсадит» «1» на другом, то мощность, выделяемая на «подсаженном» резисторе будет мала, он не перегреется и каскад н6е выйдет из строя.

Порты вывода с однотактной выходной схемой и внутренней нагрузкой



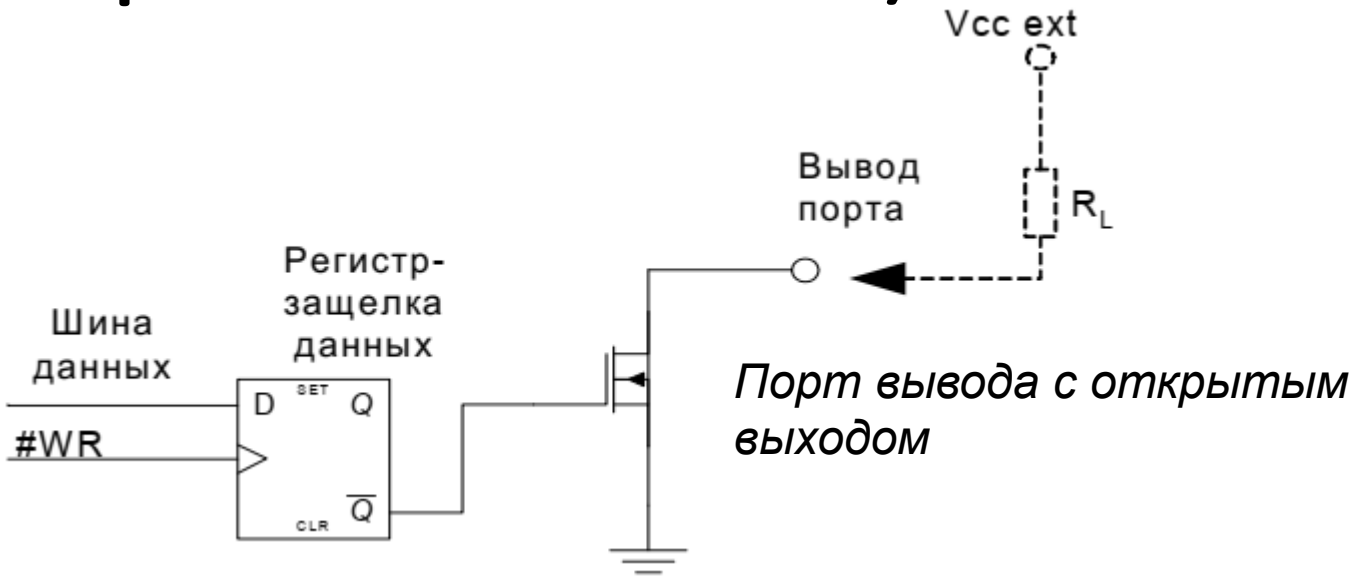
Достоинства:

- Необходимо управлять только одним регистром;
- Простая схема;
- Возможность без дополнительных схем организовать подключение на одну внешнюю шину несколько таких выходов. Легко построить квазидвунаправленный порт ввода-вывода (см. ниже).

Недостатки:

- Малый вытекающий ток (в состоянии «1»), ограниченный резистором RL – сотни uA. Это не дает управлять относительно мощными нагрузками без дополнительных каскадов усиления либо требует обеспечивать, чтобы активным был сигнал со значением «0» («управление нулем»).

Порты вывода с открытым выходом (открытым коллектором или стоком)



Порт вывода с открытым выходом

Применяются во многих семействах микропроцессоров, например, AMD Am186 (там это один из режимов порта), PICmicro. Выходной каскад построен по однотактной схеме с внешней нагрузкой. Принцип функционирования аналогичен описанному для однотактного выходного каскада.

Порты вывода с открытым выходом (открытым коллектором или стоком)

Достоинства:

• Внешнее напряжение питания нагрузки Vcc ext может быть иным – большим или меньшим, чем питание микропроцессора. Это может быть удобным для сопряжения схем с различными уровнями логической «1», например, 3.3В и 5В. Если внешнее напряжение достаточно высокое, то можно непосредственно управлять высоковольтной нагрузкой.

Например, анонсирован микроконтроллер семейства PICmicro допускающий подключение внешнего напряжения Vcc ext до 15В при питании ядра 2..6В.

• Необходимо управлять только одним регистром;

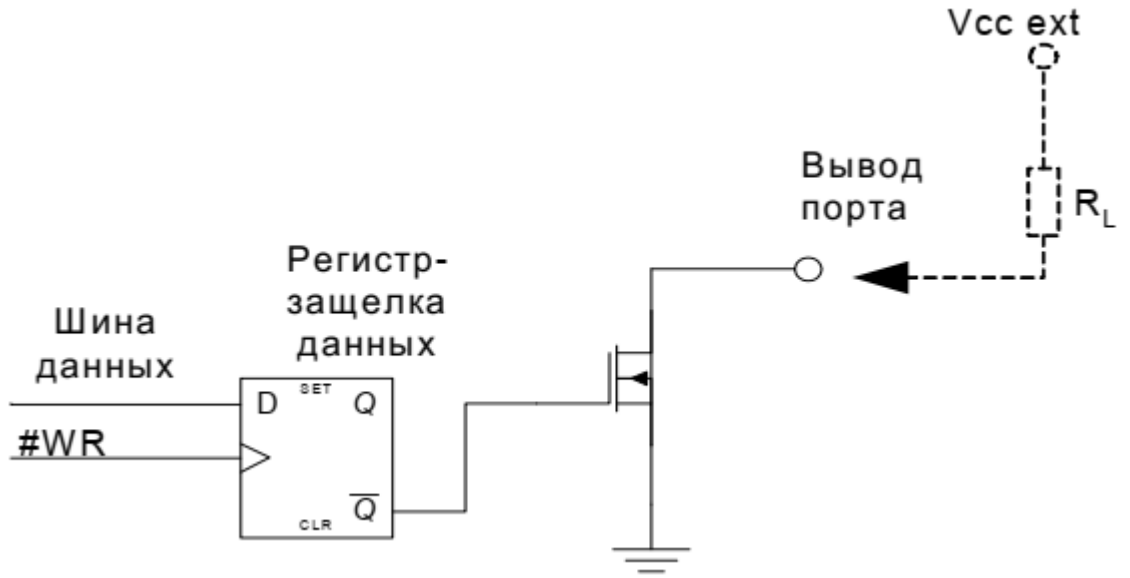
• Простая схема;

Порты вывода с открытым выходом (открытым коллектором или стоком)

- Возможность без дополнительных схем организовать подключение на одну внешнюю шину несколько таких выходов. При этом можно подбирать требуемое сопротивление RL, например, стандарт I2C требует чтобы сопротивление было 2.2кОм. Легко построить квазидвунаправленный порт ввода-вывода.

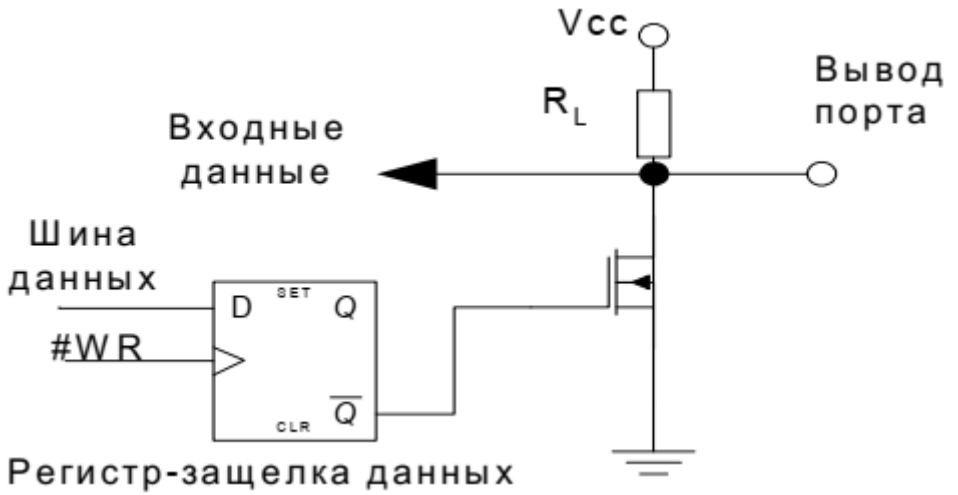
Недостатки:

- Требуется внешняя нагрузка;
- Малый вытекающий ток (в состоянии «1»), ограниченный внешним нагрузочным резистором.



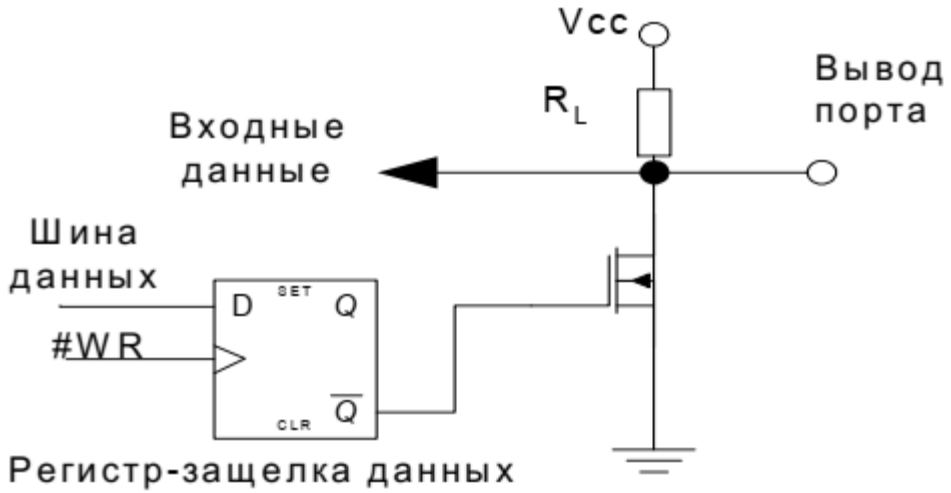
Двунаправленные порты и порты с альтернативной функцией

Самой простой схемой двунаправленного порта является квазидвунаправленный порт со схемой, аналогичной схеме порта вывода с однотактным выходным каскадом.



Квазидвунаправленный порт

Двунаправленные порты и порты с альтернативной функцией



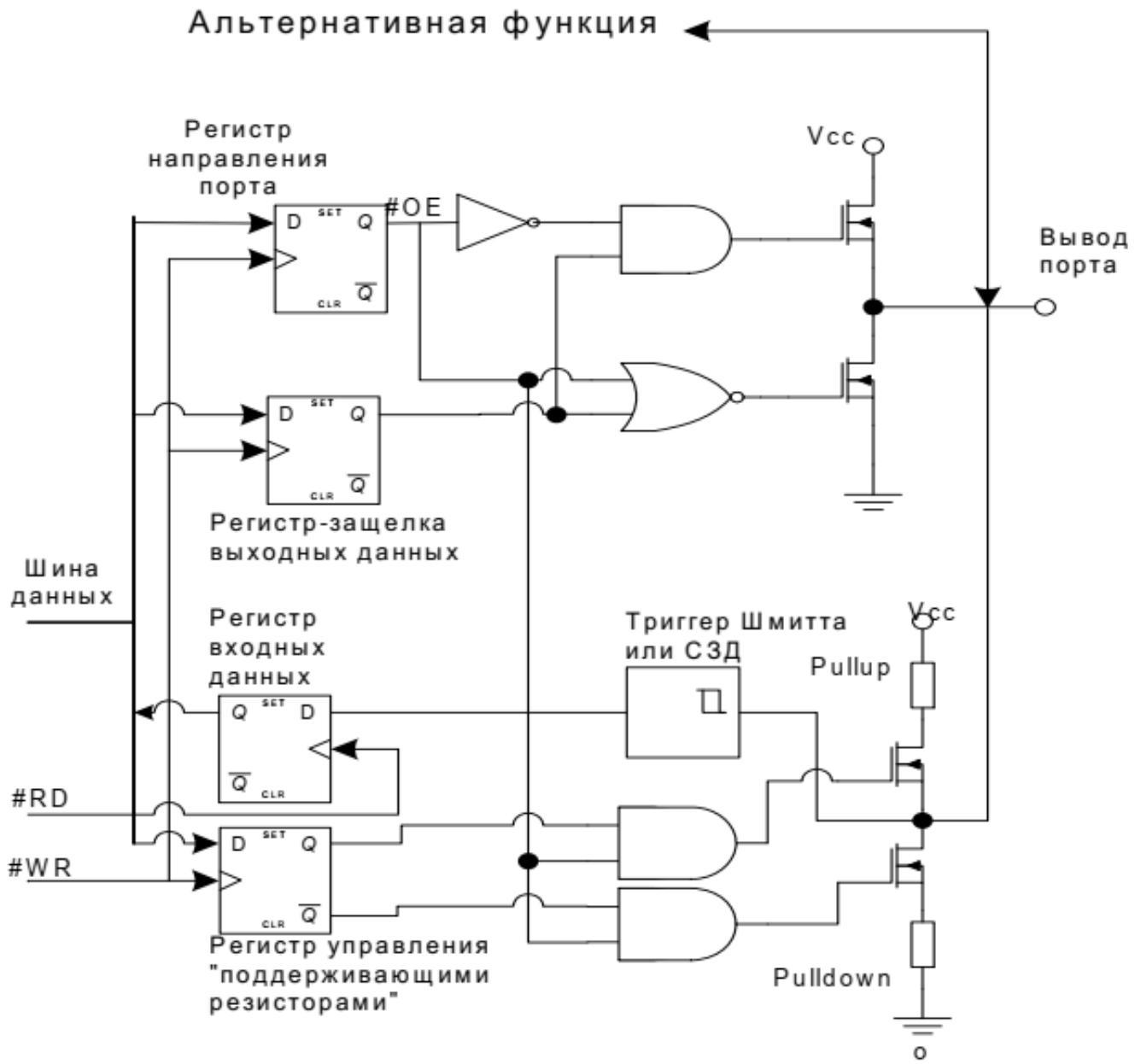
Регистр входных данных (на схеме не показан) подключен к внешнему выводу порта. Перед считыванием входных данных необходимо предварительно записать «1» в регистр-защелку выходных данных. Это закроет транзистор и исключит влияние порта вывода на входной сигнал. Резистор RL останется подключенным к входному сигналу и будет являться для него дополнительной нагрузкой, однако, так как сопротивление резистора велико (10..100 кОм), то даже на маломощный входной сигнал данная нагрузка не окажет заметного влияния. Схема квазидвунаправленного порта используется в семействе MCS-51.

Двунаправленные порты и порты с альтернативной функцией

Более часто используется схема переключаемого двунаправленного порта с комплементарным выходным каскадом.

Она объединяет схемы порта ввода и порта вывода с двухтактной выходной схемой, описанные выше. Переключение порта в режим ввода осуществляется записью «1» в регистр «вход/выход». В этом случае (как было указано при описании порта вывода) оба транзистора переводятся в закрытое состояние и порт вывода не влияет на входной сигнал. В двунаправленных портах резисторы pull-up и pull-down подключаются только в режиме ввода, для чего на вход соответствующей схемы управления подключается выход регистра «вход/выход» («1» - ввод).

Переключаемый двунаправленный порт комплементарным выходным каскадом



Двунаправленные порты и порты с альтернативной функцией

Кроме исполнения функции порта ввода-вывода, внешние выводы микросхемы могут быть задействованы для связи с внутренними периферийными модулями микропроцессора, а так же с подсистемами процессорного ядра, схем памяти и управления (с контроллером прерываний, блоком интерфейса внешней памяти и т.п.). Данные функции называются альтернативными. Обычно, когда вывод порта используется для выполнения альтернативной функции основные схемы переводятся в состояние ввода или вообще отключаются.

Таймеры-счетчики

Таймеры-счетчики

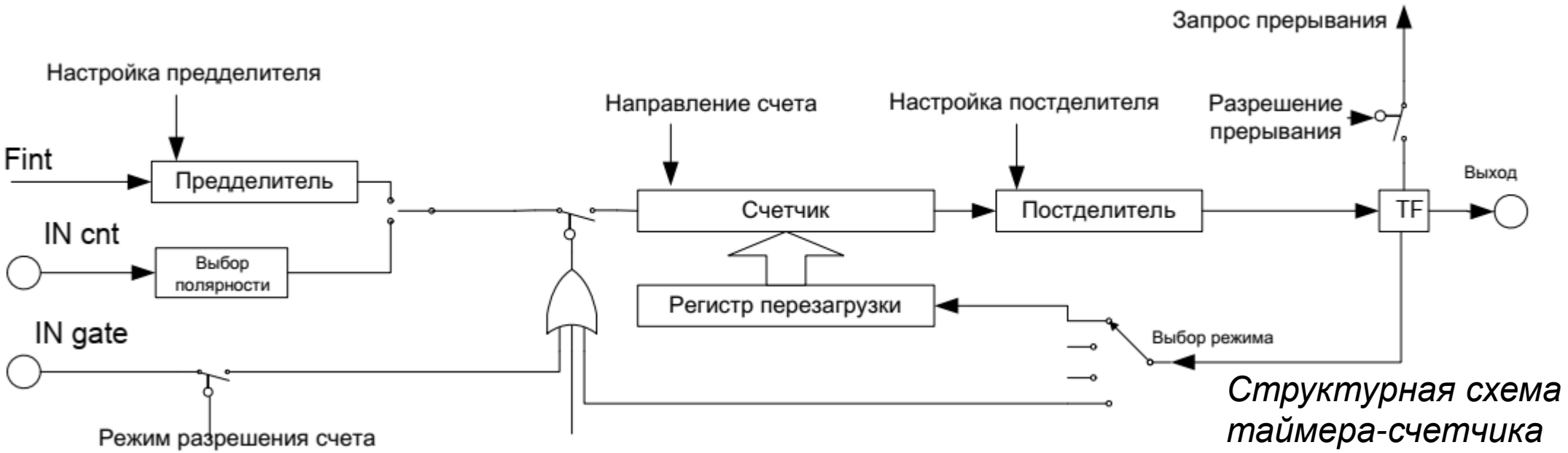
Таймеры-счетчики предназначены для:

- Подсчета временных интервалов (режим таймера);
- Подсчета числа импульсов («внешних событий») на специальном внешнем входе (режим счетчика).

Режим таймера

Тактирование счетчика выполняется от сигнала внутренней синхронизации процессора Fint. Обычно это частота процессорных циклов формируемая от основного генератора. Подсчет временных интервалов выполняется в периодах сигнала Fint.

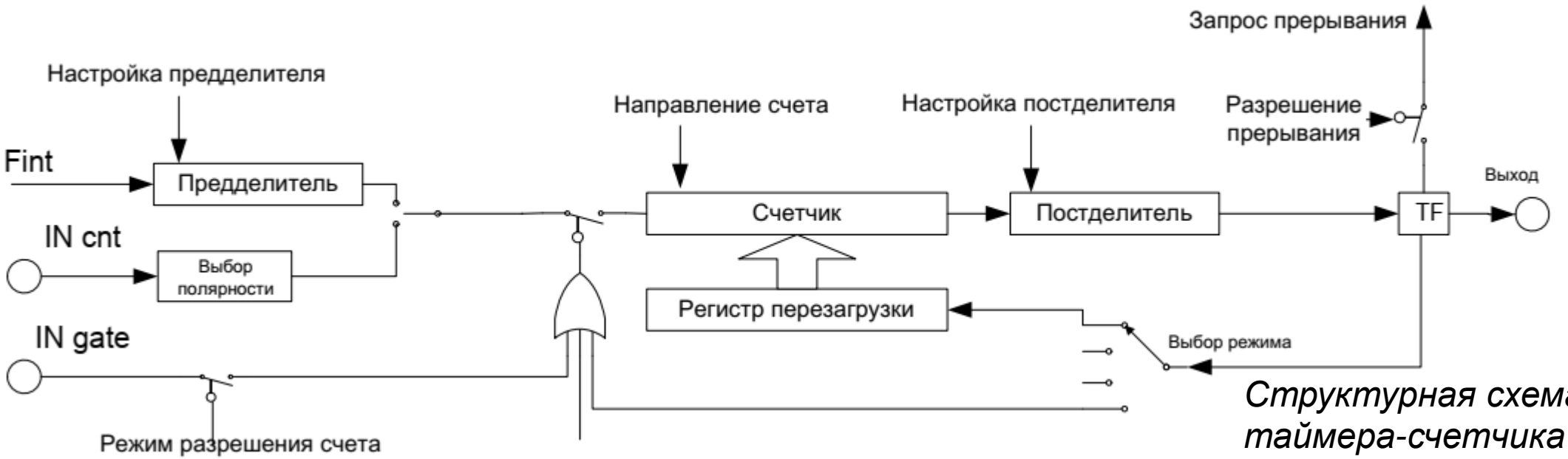
Режим таймера



Структурная схема таймера-счетчика

Предделитель используется для снижения тактовой частоты, подаваемой на регистр-счетчик. Это позволяет подсчитывать в более длительные интервалы, но увеличивает шаг дискретизации, а соответственно уменьшает точность. Предделитель может быть с фиксированным или программируемым коэффициентом деления. У программируемых предделителей обычно выбирается коэффициент деления из ряда 1, 2, 4, 8, …

Режим таймера

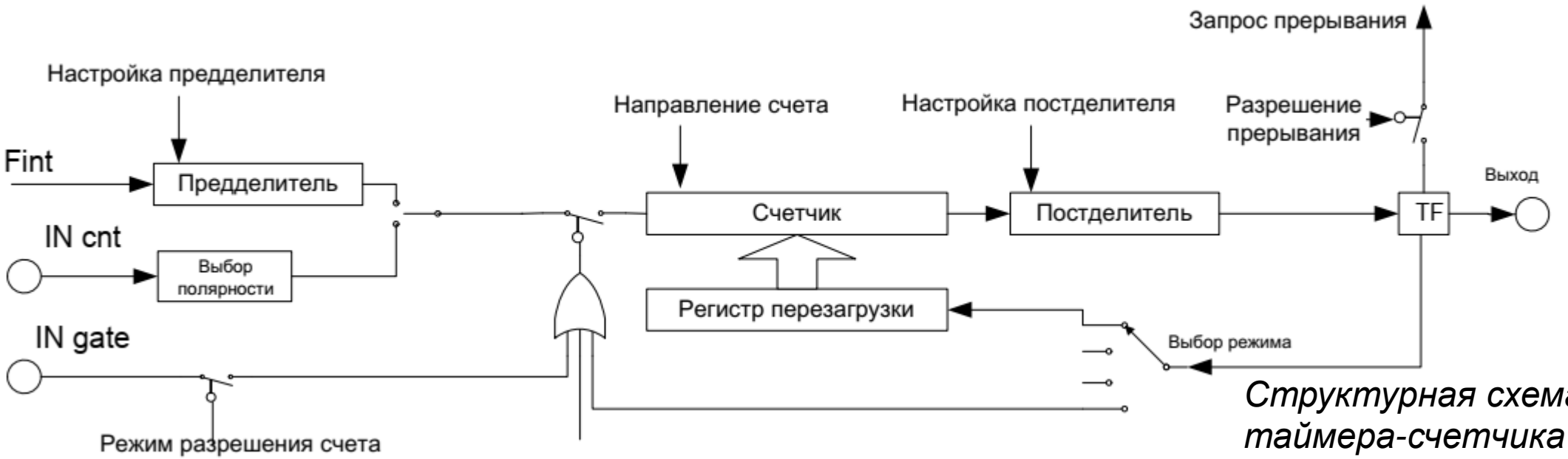


Структурная схема таймера-счетчика

Регистр-счетчик накапливает (считает) значение временного интервала в единицах входных тактов счетчика (после предделителя). Разрядность регистра-счетчика определяет разрядность всего таймера-счетчика.

Постделитель встречается достаточно редко (PICmicro) и служит для увеличения периода установки флага переполнения TF. Обычно постделитель – это дополнительные разряды регистра-счетчика недоступные по чтению-записи. Постделитель обычно программируемый на разные коэффициенты деления как и предделитель.

Режим таймера



Структурная схема таймера-счетчика

TF – флаг переполнения таймера. Устанавливается при переходе всех разрядов регистра счетчика-постделителя из 1 в 0. Обычно используется для указания окончания временного интервала. По нему может вырабатываться запрос прерывания.

Режим таймера

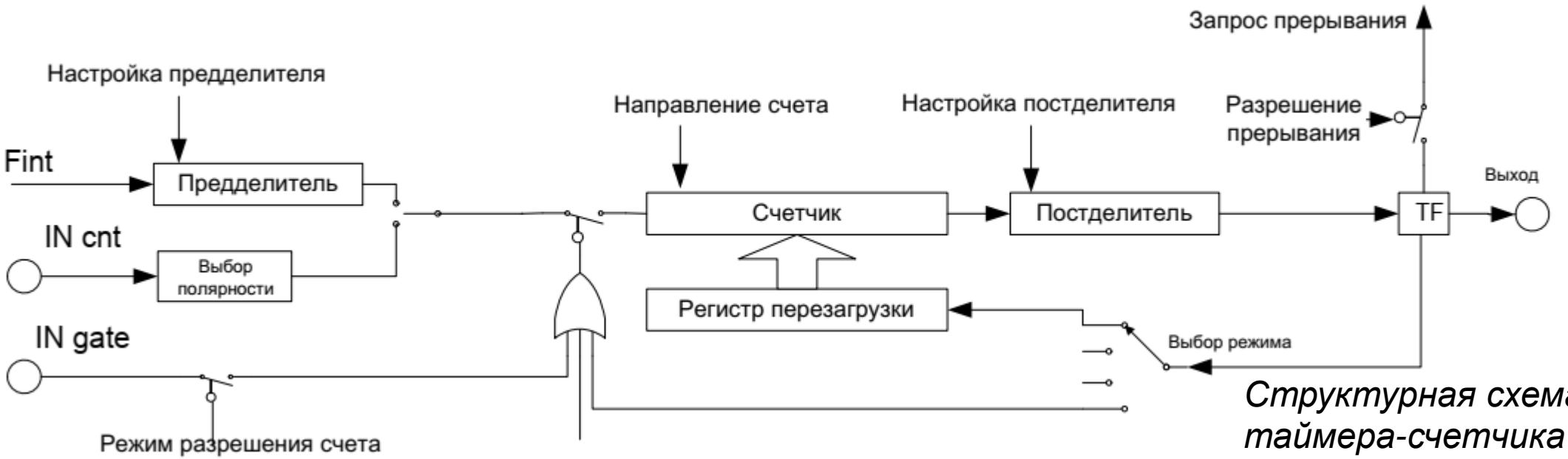
От флага TF идет цепь обратной связи, задающая режим работы таймера:

1. Однократный счет: после переполнения в регистр-счетчик загружается значение 0 и счет останавливается. Запуск следующего цикла – специальной командой из программы;

2. Циклический счет с полным циклом: после переполнения в регистр-счетчик загружается значение 0 и счет начинается снова. Полный цикл счета таймера будет 2k тактов, где k – разрядность счетчик + постделитель.

3. Циклический счет с автоперезагрузкой: после переполнения в регистр счетчик загружается значение из регистра перезагрузки. Таким образом счет можно начинать не с 0 и уменьшается (программируется) длительность цикла таймера.

Режим таймера



Структурная схема таймера-счетчика

Во многих процессорах имеется специальный вывод INgate, который выполняет функцию разрешения счета внешним сигналом. С помощью этого механизма легко подсчитывать длительность временного интервала, определяемого длительностью импульса на входе INgate.

Режим счетчика

В отличие от режима таймера, в режиме счетчика выбирается тактирование от внешнего импульсного сигнала, подаваемого на вход INcnt.

При этом подсчитываются импульсы внешнего сигнала. Инкрементация или декрементация счетчика происходит по перепаду (фронту) сигнала. Фронт сигнала в данном случае называют «внешним событием». Полярность фронтов можно программировать.

В остальном функционирование в режимах счетчика и таймера аналогично.

Программируемые таймеры в микроконтроллере с ядром Intel MCS-51

Микроконтроллер ADuC812 имеет три программируемых 16-битных таймера/счетчика: Таймер 0, Таймер 1, Таймер 2. Каждый таймер состоит из двух 8-битных регистров THX и TLX. Все три таймера могут быть настроены на работу в режимах “таймер” или "счетчик".

В режиме "таймер" регистр инкрементируется каждый машинный цикл, т.е. можно рассматривать это как подсчет машинных циклов. Так как машинный цикл состоит из 12 перепадов напряжения на тактовом входе микроконтроллера, частота инкрементирования таймера в 12 раз меньше тактовой частоты микроконтроллера (соответствующего кварцевого резонатора).

Программируемые таймеры в микроконтроллере с ядром Intel MCS-51

В режиме "счетчик" регистр инкрементируется по перепаду из "1" в "0“ внешнего входного сигнала, подаваемого на вывод микроконтроллера T0, T1 или T2. Когда опрос показывает высокий уровень в одном машинном цикле и низкий уровень в следующем, счетчик увеличивается на 1. Таким образом, на распознавание периода требуются два машинных цикла, максимальная частота подсчета входных сигналов равна 1/24 частоты кварцевого резонатора. На длительность периода входных сигналов ограничений сверху нет. Для гарантированного прочтения входной сигнал должен удерживать значение 1, как минимум, в течение одного машинного цикла микроконтроллера.

Программируемые таймеры в микроконтроллере с ядром Intel MCS-51

Для конфигурации и контроля Таймеров используются 3 регистра специального назначения: TMOD и TCON для управления Таймерами 0 и 1, T2CON для управления Таймером 2.

Схемы управления Таймерами 0 и 1 идентичны (оба Таймера входят в ядро Intel MCS-51, Таймер 2 – нет). Далее рассмотрим принцип работы этих таймеров.

Программируемые таймеры в микроконтроллере с ядром Intel MCS-51

Таймер 0 и Таймер 1 могут работать в четырех режимах работы:

- режим 0: 13-битный таймер
- режим 1: 16-битный таймер
- режим 2: 8-битный автоперезагружаемый таймер
- режим 3: Таймер 0 как 2 раздельных 8-битных таймера.

Кроме того, Таймер 1 можно использовать для задания скорости передачи (baud rate) последовательного порта

Формат регистра управления режимами работы таймеров TMOD

TMOD Адрес = 89H Значение после сброса = 0000 0000B.

Не имеет побитовой адресации

	ТАЙМЕР 1				ТАЙМЕР 0			
	GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0
Бит	7	6	5	4	3	2	1	0

Программируемые таймеры в микроконтроллере с ядром Intel MCS-51

Так как управление таймерами 0 и 1 полностью идентично, то приведём назначение битов по именам:

Название	Позиция бита	Назначение
GATE	TMOD.7 для таймера 1 и TMOD.3 для таймера 0	Управление блокировкой таймера от сигнала INTx. Если бит установлен в 1, то таймер/счетчик "x" разрешен до тех пор, пока на входе "INTx" высокий уровень и бит управления "TRx" установлен. Если бит сброшен в 0, то T/C разрешается, как только бит управления "TRx" устанавливается в 1.
C/T	TMOD.6 для таймера 1 и TMOD.2 для таймера 0	Бит выбора режима таймера или счетчика событий. Если бит сброшен в 0, то таймер работает от внутреннего генератора, если установлен в 1, то работает от внешних сигналов на входе "Tx".

Программируемые таймеры в микроконтроллере с ядром Intel MCS-51

M1	TMOD.5 для таймера 1 и TMOD.1 для таймера 0
M0	TMOD.4 дляT/C1 и TMOD.0 дляT/C0

Выбор режима работы таймера

M1	M0	
0	0	13 битный таймер/счетчик "TLx" работает как 5-битный предварительный делитель
0	1	16 битный таймер/счетчик. "THx" и "TLx" включены последовательно
1	0	8-битный автоперезагружаемый таймер/счетчик. "THx" хранит значение, которое должно быть перезагружено в "TLx"каждый раз по переполнению
1	1	Таймер/счетчик 1 останавливается. Таймер/счетчик 0: TL0 работает как 8-битныйтаймер/счетчик, и его режим определяется управляющими битами таймера 0. TH0 работает только как 8 битный таймер, и его режим определяется управляющими битами таймера 1

Схема управления Таймерами 0 или 1

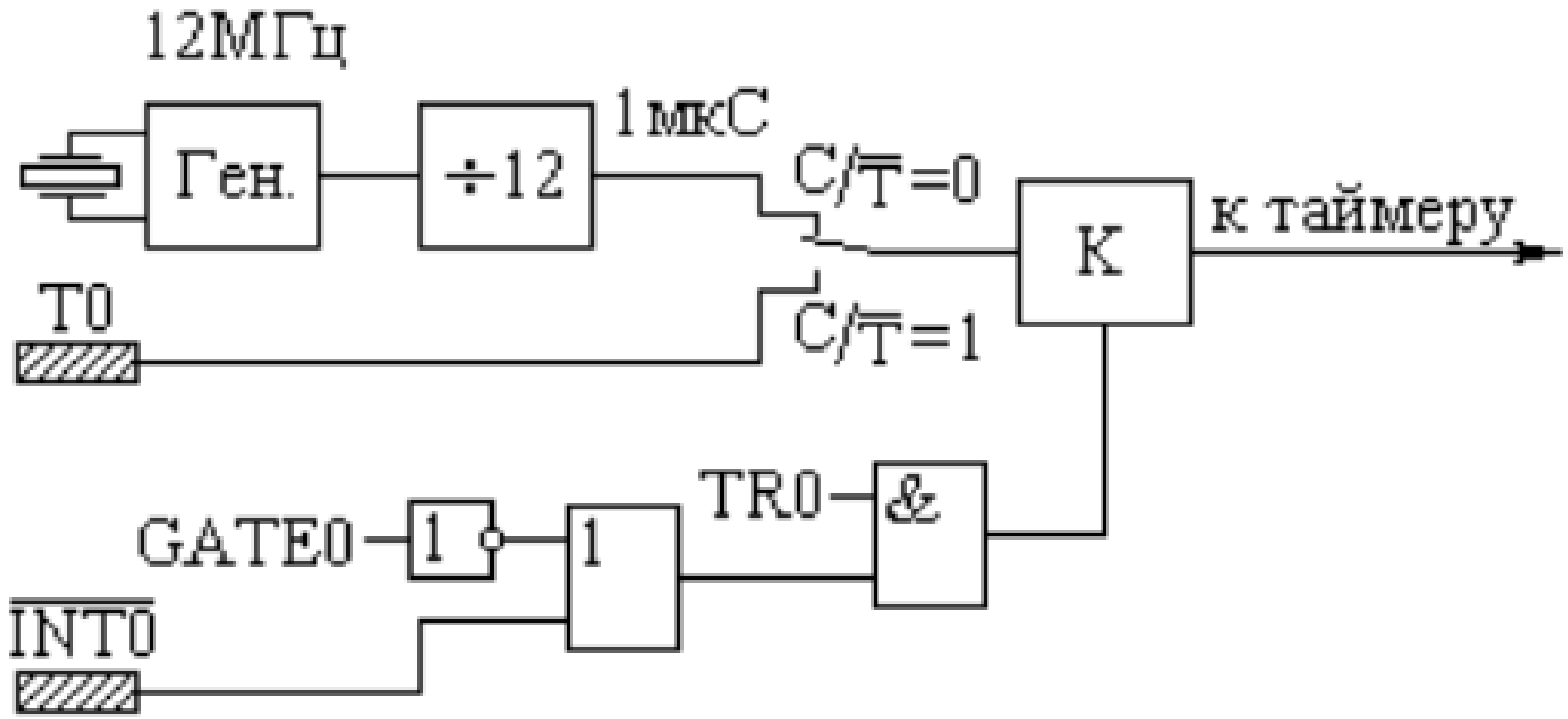
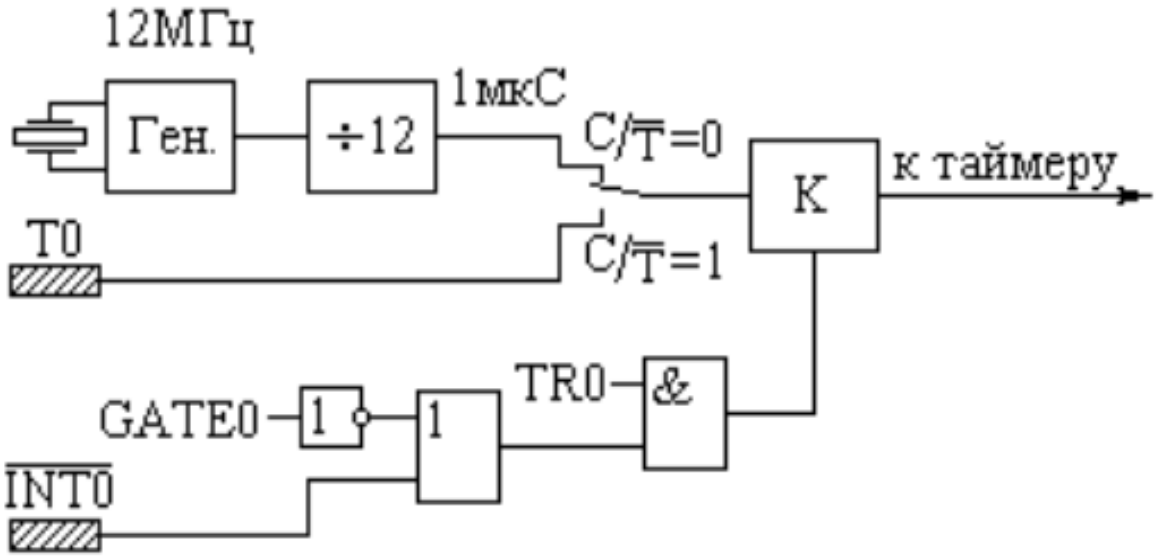


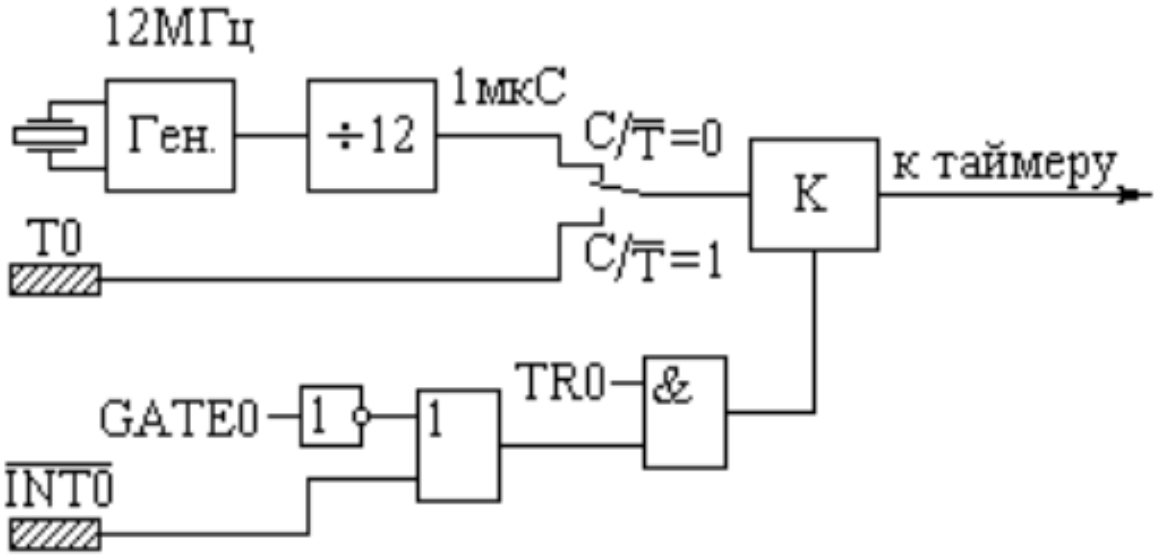
Схема управления Таймерами 0 или 1



В приведенной схеме заштрихованным прямоугольником обозначены внешние выводы микросхемы микроконтроллера.

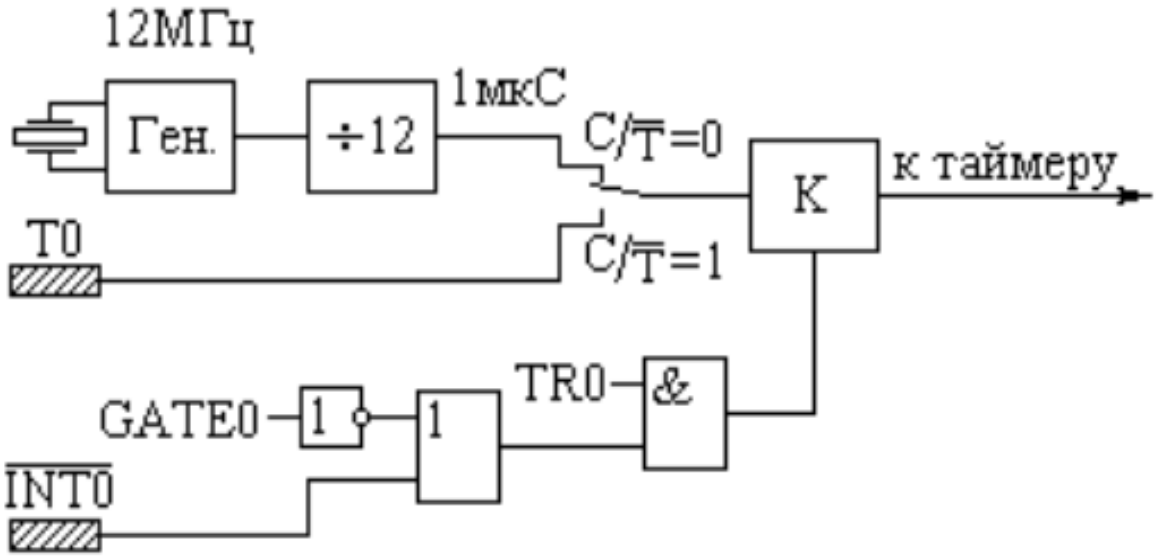
Из схемы видно, что таймер может включаться и выключаться битами TRx. Таким образом, можно уменьшать потребление микросхемы и уровень помех, создаваемый ею. Учитывая, что счетчики таймеров переключаются на высокой частоте, то они могут потреблять до половины тока потребления микроконтроллера. Следует отметить, что при включении и после сброса микроконтроллера работа таймеров запрещена.

Схема управления Таймерами 0 или 1



Есть возможность управлять работой таймера извне при помощи внешнего вывода Т0 для таймера Т0 или Т1 для таймера Т1. Для этого необходимо записать в бит GATEx логическую единицу (не забыв при этом разрешить работу таймера при помощи бита TRx). Установка GATE=1приводит к тому, что работа таймера контролируется внешним входом INTx, что позволяет измерять длительность импульса..

Схема управления Таймерами 0 или 1



Биты включения таймеров TR0 и TR1 размещены в регистре TCON (control - управлять), а биты GATE и C/T в регистре TMOD.

Формат регистра управления режимами работы таймеров TCON

TCON Адрес = 88H Значение после сброса = 0000 0000B

Побитовая адресация

	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
Бит	7	6	5	4	3	2	1	0

Формат регистра управления режимами работы таймеров TCON

TCON Адрес = 88H Значение после сброса = 0000 0000B

Побитовая адресация

	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
Бит	7	6	5	4	3	2	1	0

Символ	Позиция	Назначение
TF1	TCON.7	Флаг переполнения таймера 1. Устанавливается аппаратно при переполнении таймера/счетчика. Сбрасывается при обслуживании прерывания аппаратно.
TR1	TCON.6	Бит управления таймера 1. Устанавливается/сбрасывается программой для пуска/останова.
TF0	TCON.5	Флаг переполнения таймера 0. Устанавливается аппаратурно. Сбрасывается при обслуживании прерывания
TR0	TCON.4	Бит управления таймера 0. Устанавливается / сбрасывается программой для пуска/останова таймера/счетчика.

Формат регистра управления режимами работы таймеров TCON

TCON Адрес = 88H Значение после сброса = 0000 0000B

Побитовая адресация

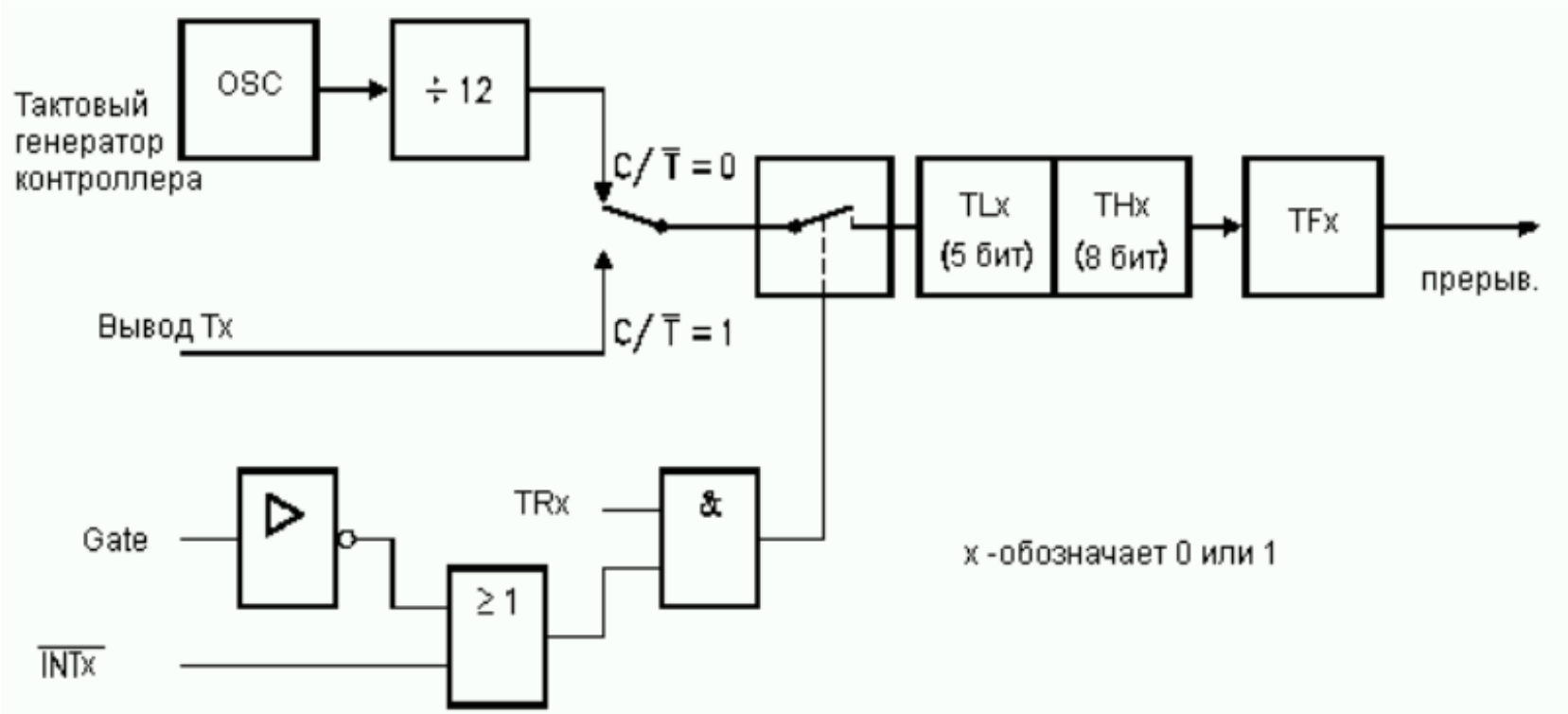
	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
Бит	7	6	5	4	3	2	1	0

IE1	TCON.3	Флаг внешнего прерывания 1. Устанавливается аппаратно, когда детектируется срез внешнего сигнала INT1. Сбрасывается при обслуживании прерывания.
IT1	TCON.2	Бит управления типом прерывания 1. Устанавливается / сбрасывается программно для определения типа запроса прерывания INT1 (по спаду/по низкому уровню).
IE0	TCON.1	Флаг внешнего прерывания 0. Устанавливается по срезу сигнала INT0. Сбрасывается при обслуживании прерывания.
IT1	TCON.0	Бит управления типом прерывания 0. Устанавливается / сбрасывается программно для определения типа запроса прерывания INT0 (по спаду/по низкому уровню).

Режим 0 (13-битный Таймер/Счетчик)

В данном режиме Таймер X работает как 13-битный суммирующий таймер/счётчик. Этот таймер/счётчик состоит из 8 бит регистра THx и младших 5 бит регистра TLx, где x в обозначении регистра заменяется на 0 или 1 в зависимости от того таймера, которым мы управляем. Старшие 3 бита регистров TLx не определены и игнорируются. Таким образом, этот режим можно трактовать как 8-битный таймер/счетчик с предварительным делителем (на 32) на входе. Установка запускающего таймер флага TR0 или TR1 не очищает эти регистры.

Работа таймера в режиме 0.

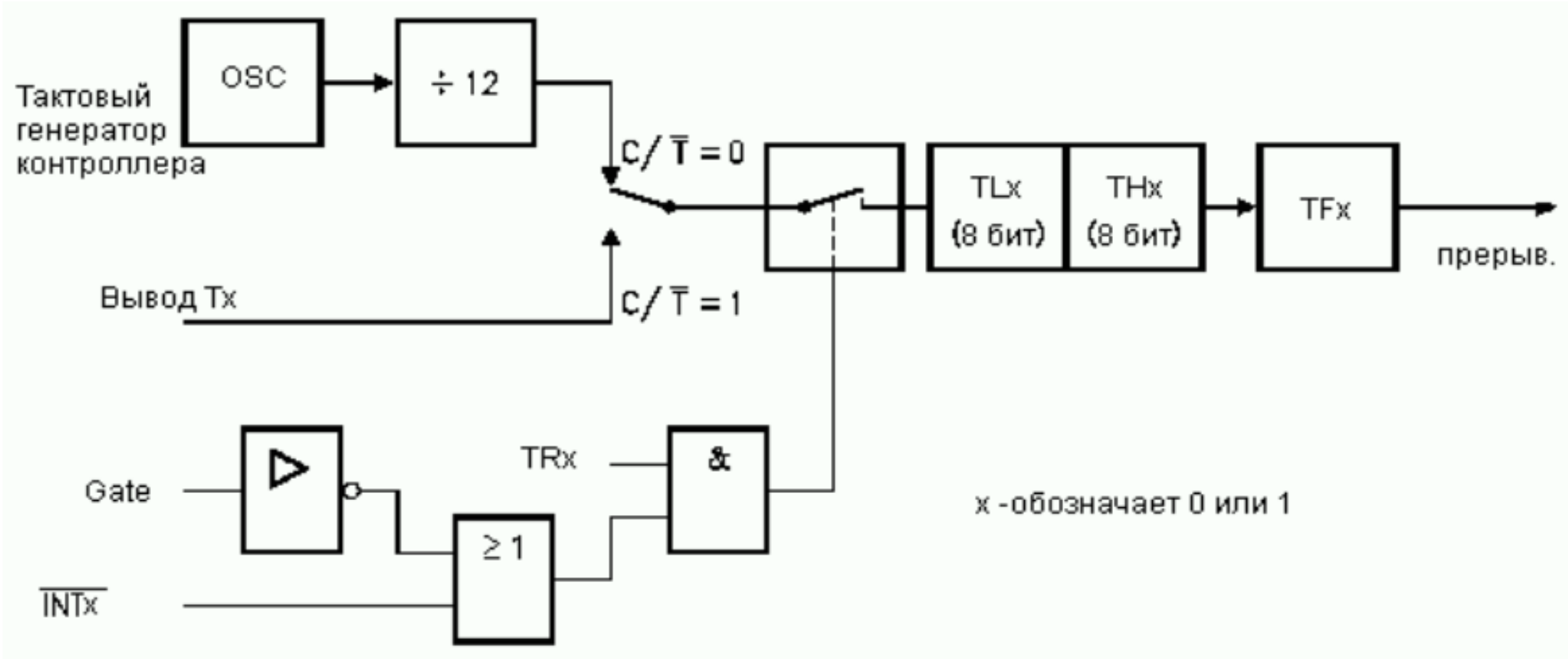


Когда содержимое счетчика изменяется из состояния все "1" в состояние все "0" , то устанавливается (принимает значение "1") флаг прерывания таймера TF0 или TF1.

Режим 1 (16-битный Таймер/Счетчик)

В первом режиме работы Таймер X работает как шестнадцатиразрядный таймер/счётчик. Режим 1 похож на режим 0, за исключением того, что в регистрах таймера использует все 16 бит. В этом режиме регистры THx и TLx также включены друг за другом.

Работа таймера в режиме 1



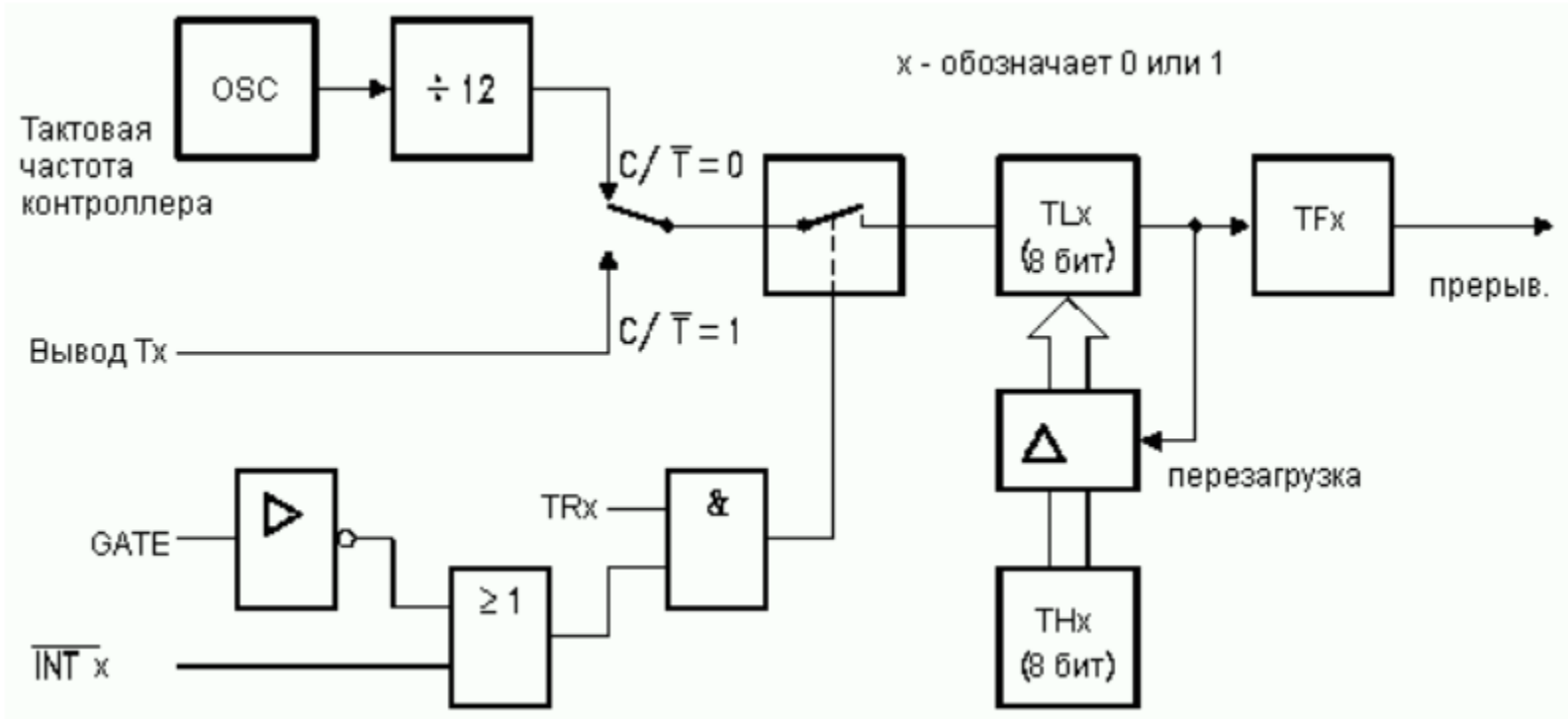
Программируемые таймеры в микроконтроллере с ядром Intel MCS-51

Нулевой и первый режимы работы Таймеров 0 и 1 предназначены для формирования одиночного интервала времени. Если возникает необходимость формировать последовательность интервалов времени для периодических процессов, то загрузка регистров TH0 и TL0 для задания нужного интервала времени производится программно, что для коротких интервалов времени может привести к значительным затратам процессорного времени. Для формирования последовательности одинаковых интервалов времени используется режим работы таймера с автоперезагрузкой – режим 2.

Режим 2 (8-битный Таймер/Счетчик с автоперезагрузкой)

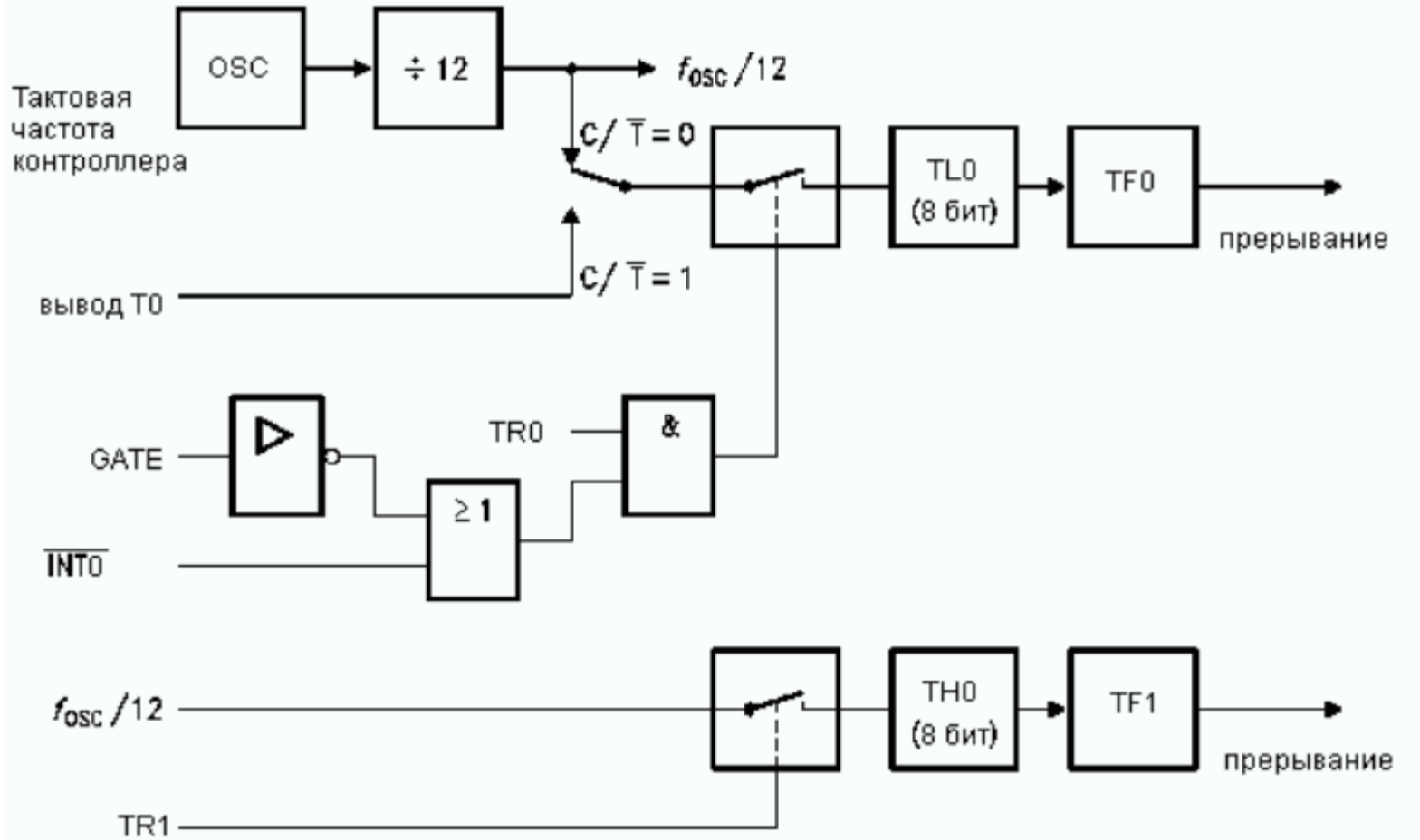
В режиме 2 регистр таймера TLx работает как 8-битный счетчик с автоматической перезагрузкой начального значения из регистра THx в регистр TLx. Переполнение регистра TLx не только устанавливает флаг TFx, но и загружает регистр TLx содержимым регистра THx, который предварительно инициализируется программно. Перезагрузка не изменяет содержимое регистра THx.

Работа таймера в режиме 2



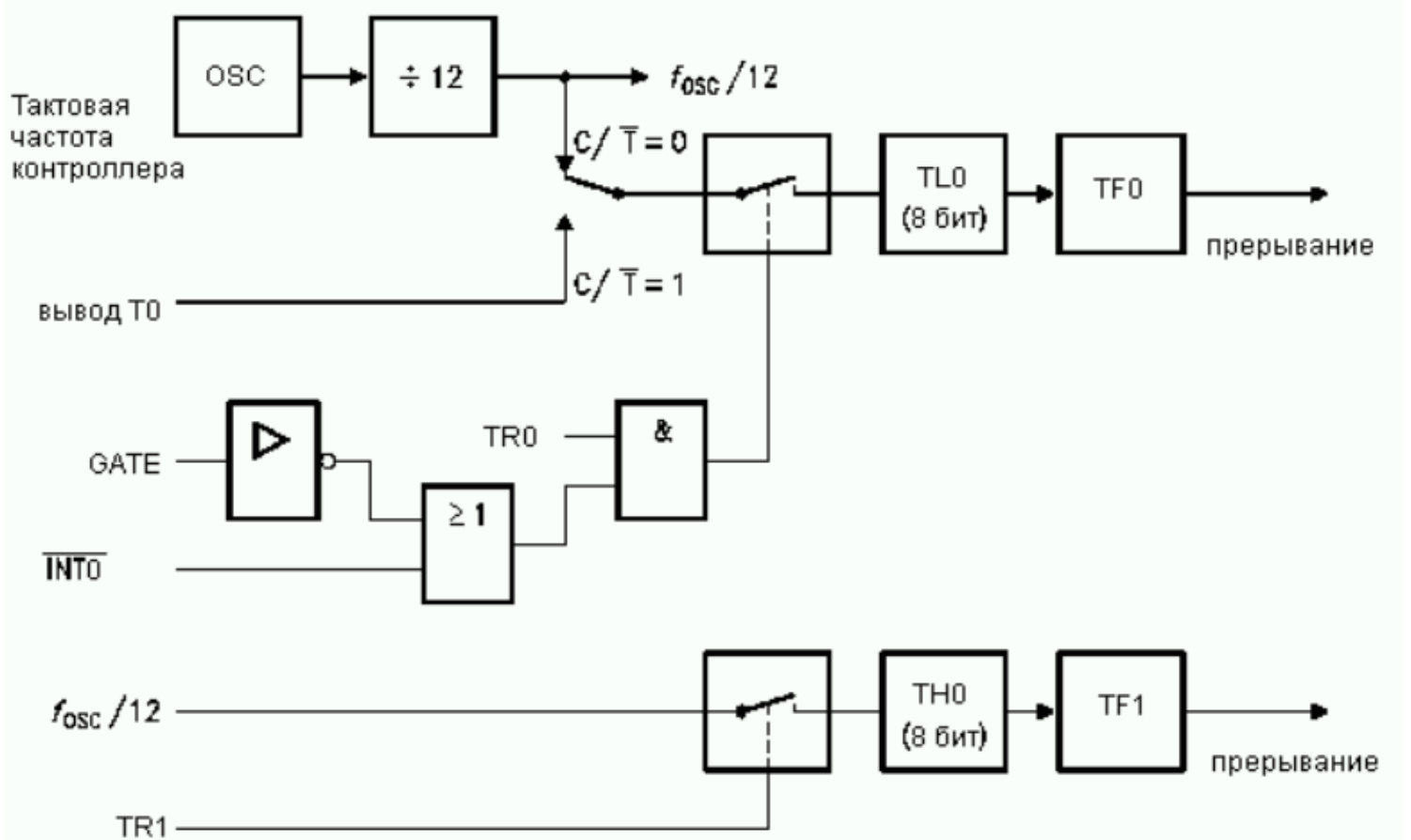
Режим 3 (Два 8-битных Таймера/Счетчика)

Работа в режиме 3 имеет отличия для таймеров 0 и 1. Таймер 1 в этом режиме просто останавливает свой счет. Тот же эффект даст установка TR1=0. Логика работы таймера 0 в режиме 3 показана на схеме:



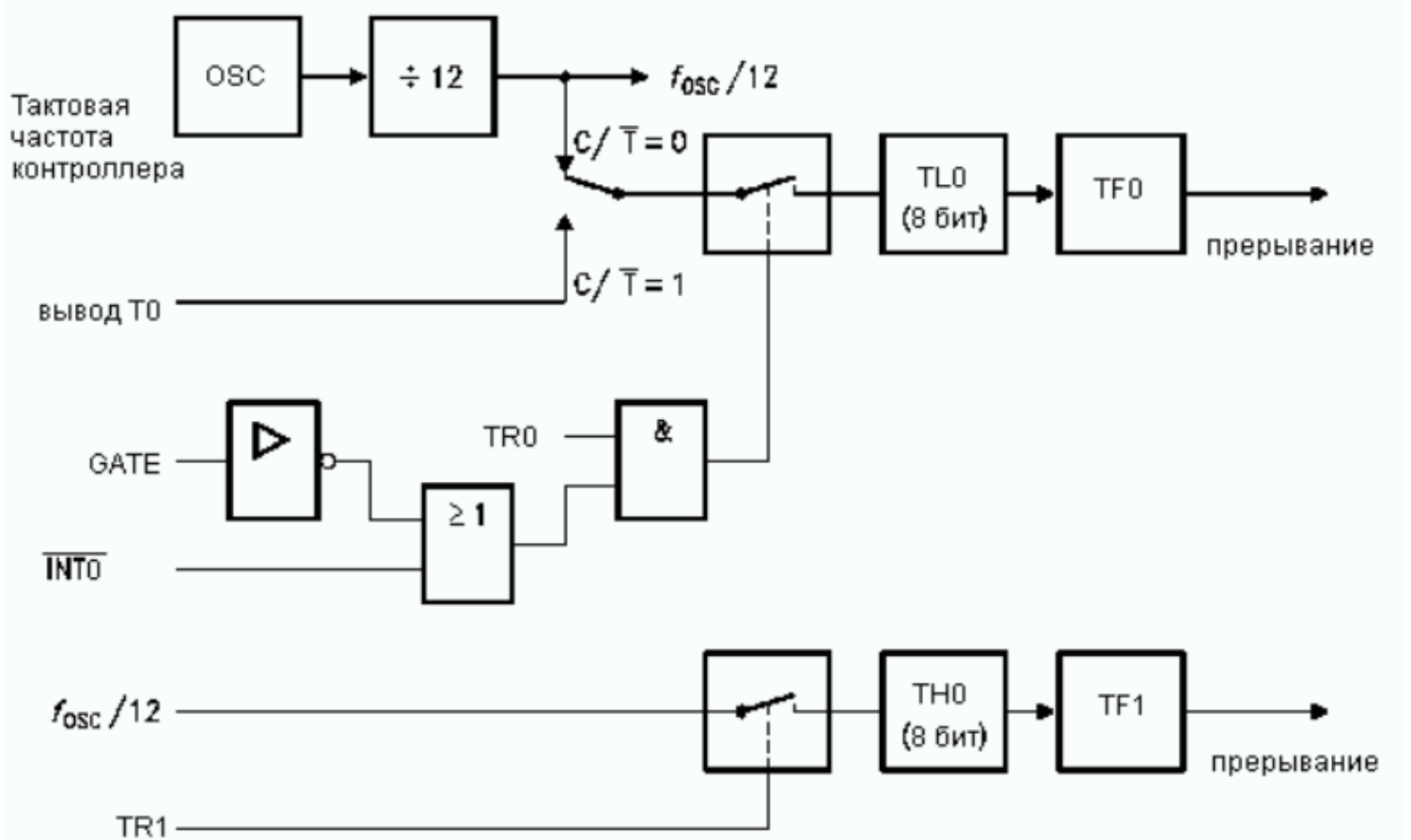
Режим 3 (Два 8-битных Таймера/Счетчика)

Таймер 0 в режиме 3 устанавливает TL0 и TH0 как два разных счетчика. Счетчик на базе TL0 использует биты управления таймера 0: C/T, GATE, TR0, INT0, TF0. TH0 зафиксирован в режиме таймера (считающего машинные циклы), и использует для управления биты TR1 и TF1 таймера 1. Таким образом, прерывание от переполнения регистра TH0, будет обозначено флагом TF1.



Режим 3 (Два 8-битных Таймера/Счетчика)

Режим 3 предназначен для приложений, которым нужен дополнительный 8-битный таймер/счетчик. Когда таймер 0 работает в режиме 3, таймер 1 может быть выключен установкой его в режим 3, или может быть оставлен включенным для использования в качестве генератора тактовых импульсов для последовательного интерфейса, или для любого приложения, которому не требуется прерывание именно от таймера 1.



Настройка таймера на заданную частоту

Задача настройки таймера на заданную частоту во встраиваемых системах обычно связана с организацией системного времени.

В нашем случае под этим подразумевается настройка таймера в режиме «таймер» таким образом, чтобы его переполнения происходили через одинаковые интервалы времени (1 мс, 5 мс, 10 мс и т.д.), так называемые кванты времени.

Настройка таймера на заданную частоту

Расчет необходимой частоты работы таймера может быть произведен по следующей формуле:

$$F = \frac{f_{osc}}{12 \cdot counts},$$

где F – необходимая частота, fosc – частота микроконтроллера, Counts – количество тиков (счетов) таймера для достижения частоты F.

Настройка таймера на заданную частоту

Так как таймеры у нас суммирующие, то регистры таймера (THx, TLx) нужно инициализировать следующих кодом:

$$T_{timer} = Counts_{max} - Counts$$

где $Counts_{max}$ – максимальное количество тиков в таймере, которое определяется по разрядности таймера, его режиму работы. Например, если таймер 16-битный (режим 1), то $Counts_{max}$ = 65536.

Настройка таймера на заданную частоту

Если в стенде SDK-1.1 необходимо настроить Таймер 0 на частоту

1000 Гц, то для этого нужно использовать 16-битный таймер (режим 1). По формулам получится, что Counts = 921 (чуть больше) и Ttimer = 64615 (FC67h).

Таким образом, регистры Таймера 0 должны быть инициализированы так: TH0 = FCh, TL0 = 67h.

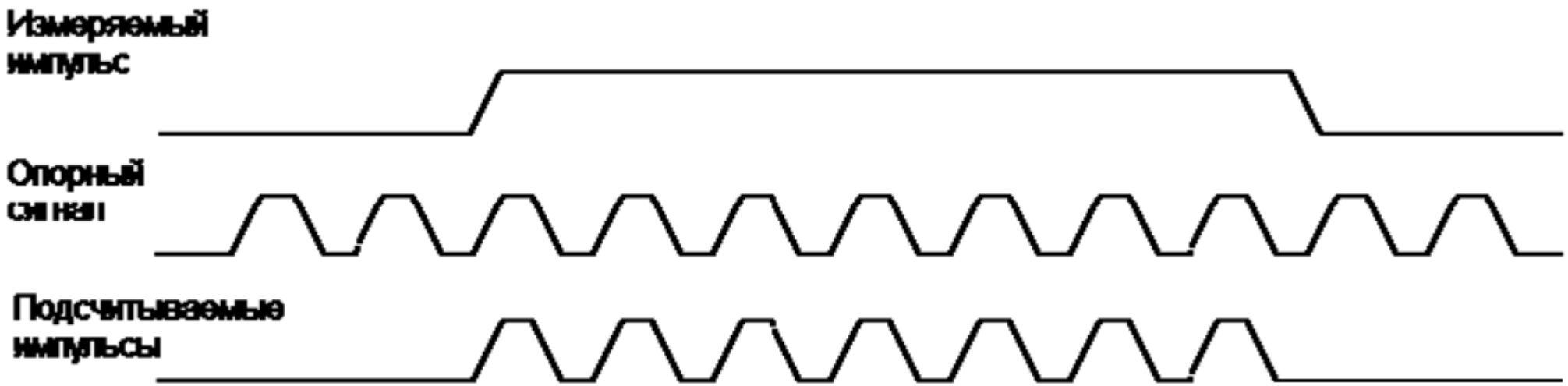
Использование таймера в качестве измерителя ширины импульсов

Известно, что измерение длительности импульса можно произвести, подсчитав импульсы эталонной частоты.

Для измерения длительности импульса измеряемый сигнал подаётся на вывод микроконтроллера INTx и в бит управления GATE записывается разрешающий сигнал логической единицы. Таймер/счётчик настраивается в режим таймера записью в бит C/Tx логического нуля. Содержимое таймера обнуляется.

Если теперь на вход микроконтроллера INT0 подать импульс с неизвестной длительностью, то в регистрах TH0 и TL0 будет записана его длительность в микросекундах.

Принцип измерения длительности импульсов



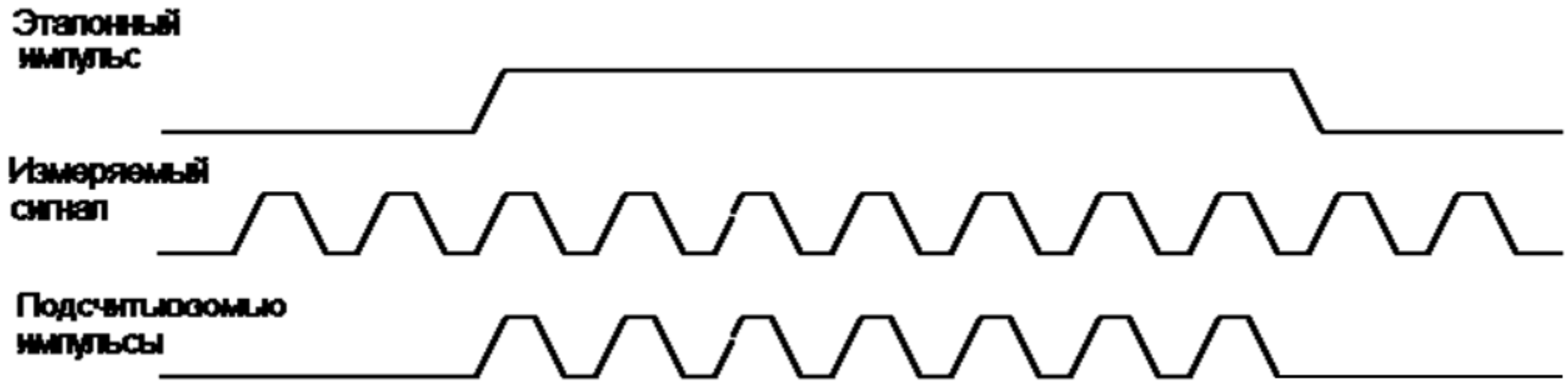
Использование таймера в качестве частотомера

Известно, что измерение частоты можно произвести, подсчитав количество периодов неизвестной частоты за единицу времени.

Для измерения частоты измеряемый сигнал подаётся на вывод микроконтроллера Tx. Таймер/счётчик настраивается в режим счётчика записью в бит C/Tx логической единицы. Содержимое таймера обнуляется. Таймер включается на строго определённый интервал времени. Этот интервал задаётся оставшимся таймером.

Если теперь на вход микроконтроллера T0 подать сигнал с неизвестной частотой, то в регистрах TH0 и TL0 будет записана его частота в килогерцах.

Принцип измерения частоты



Модули таймеров-счетчиков со схемами входного захвата, выходного сравнения и выработки сигналов с ШИМ

Модули Capture/Compare/PWM (CCP) являются развитием структуры таймеров-счетчиков и выполняют схожие функции, однако требуют меньшей программной поддержки, более гибки в настройке на различные задачи, позволяют достигнуть более высокого быстродействия. Наибольшую эффективность они обеспечивают при работе с внешними периодическими или непериодическими сигналами при решении следующих задач:

- Фиксация времени (момента) внешнего события (фронта);
- Определение частоты и длительности импульсов внешнего сигнала, фазового сдвига нескольких сигналов;
- Формирование одиночных импульсов с программируемой длительностью.

Модули таймеров-счетчиков со схемами входного захвата, выходного сравнения и выработки сигналов с ШИМ

• Формирование на одном или нескольких выводах периодических последовательностей импульсов и программируемой частотой, длительностью, фазовым сдвигом (в случае нескольких выходных сигналов);

• Формирование сигналов с широтно-импульсной модуляцией (ШИМ, PWM). При ШИМ частота сигнала остается постоянной, а длительность положительного и отрицательного импульсов программируется. Основная характеристика сигнала с ШИМ является скважность: отношение периода к длительности положительного импульса. Для меандра скважность равна 2. Модуль ШИМ с подключенной к его выходу интегрирующей цепочкой образует простейший ЦАП. Такое использование модулей ШИМ является основным во встраиваемых системах.

Все перечисленные функции выполняются модулями CCP автономно, а вмешательство программиста требуется только на этапе настройки режимов модуля.

Схема выходного сравнения (Output Compare)

Многоразрядный цифровой компаратор непрерывно сравнивает изменяющийся во времени код таймера-счетчика с кодом, который записан в регистре сравнения. В момент равенства этих кодов устанавливается флаг OCF (Output Compare Flag) и изменяется сигнал на выводе OCO (Output Compare Output). Возможны три варианта изменения сигнала, которые могут быть настроены программно: установка «1», установка «0», инвертирование сигнала на выводе OCO (OCO <= #OCO). По установке флага OCF может быть сброшен (обнулен) или перезагружен определенным значением регистр-счетчик. Кроме того, по установке флага OCF может быть выработан запрос прерывания, если данное прерывание разрешено. Запрос прерывания может вырабатываться и при переполнении таймера-счетчика.

Модуль выходного сравнения (Output Compare)

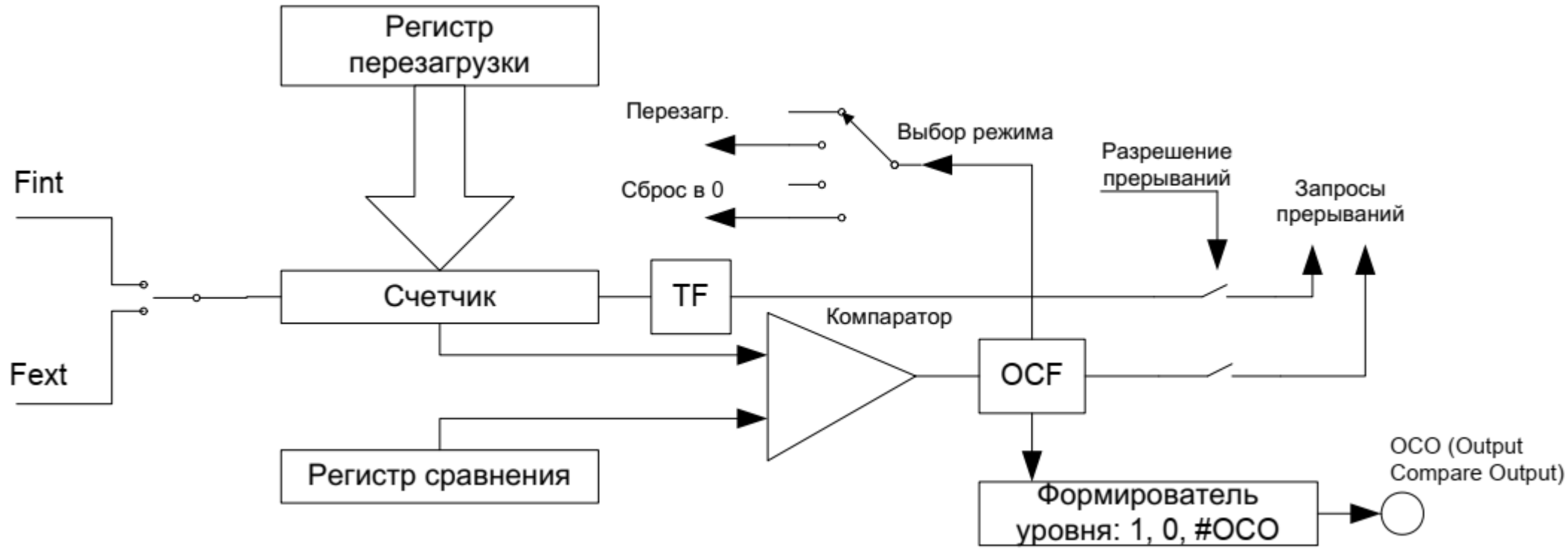


Схема выходного сравнения (Output Compare)

Рассмотрим примеры типовых применений модуля CCP в режиме выходного сравнения:

1. Формирование сигнала с определенной частотой: формирователь уровня настраивают на режим инверсии OCO, управление таймеромсчетчиком в режим сброса по флагу OCF, в регистр сравнения – значение, равное полупериоду формируемой частоты. По каждому событию сравнения раз в полупериод порт OCO инвертируется и формируется передний или задний фронт сигнала.

2. Формирование одиночного импульса определенной длительности: формирователь уровня настраивают на режим установки OCO в «0», в регистр сравнения – длительность импульса, таймер обнуляем и одновременно устанавливаем порт OCO в «1» (передний фронт). По событию сравнения порт обнуляется (задний фронт).

Схема выходного сравнения (Output Compare)

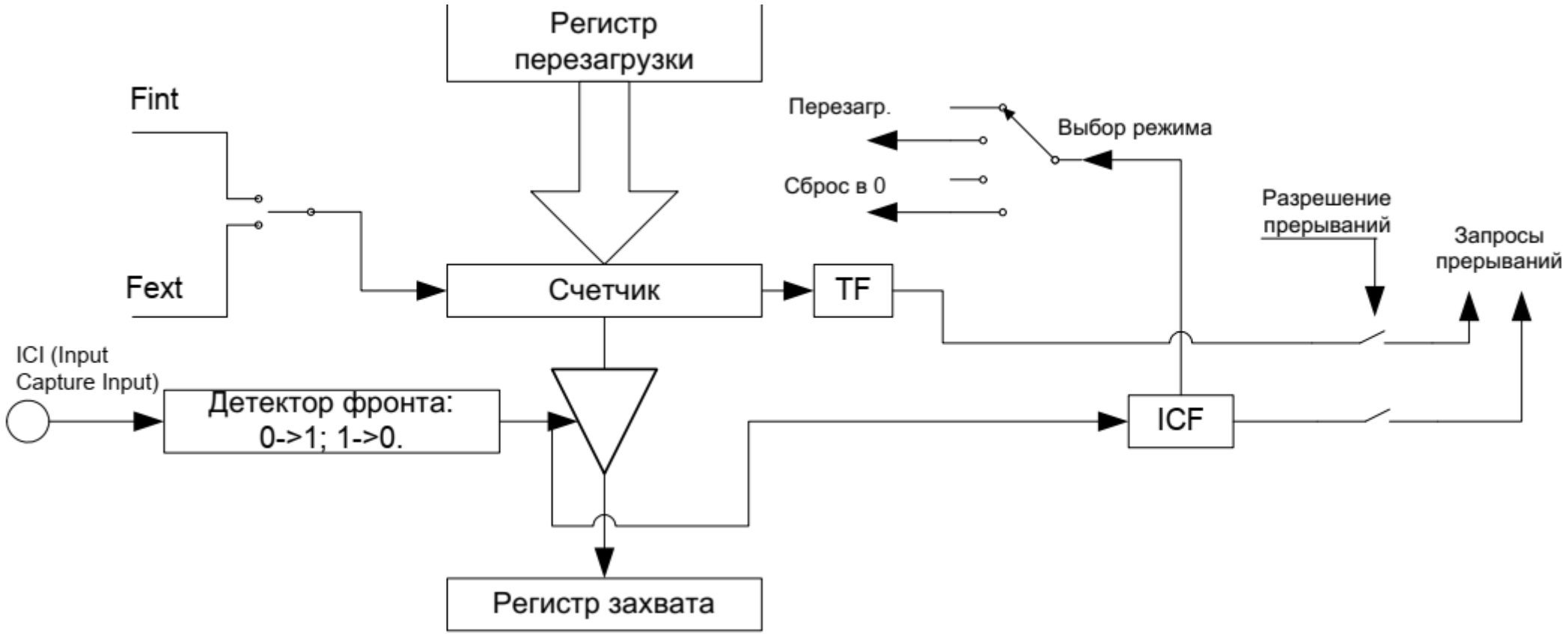
3. Ожидание определенного числа импульсов на счетном входе (сигнал Fext) таймера-счетчика: таймер настраиваем в режим счетчика, обнуляем, в регистр сравнения записываем требуемое число импульсов, разрешаем прерывание по событию сравнения (по флагу OCF). После прохождения заданного числа импульсов будет выработан запрос прерывания.

4. Делитель входной частоты на заданное число N, кратное двум: таймерсчетчик переключаем в режим счетчика, устанавливаем обнуление счетчика по флагу OCF, формирователь уровня настраивают на режим инверсии OCO, в регистр сравнения записываем значение N/2.

Схема входного захвата (Input Capture)

Функцию входного захвата поддерживают микроконтроллеры семейств (Atmel), 8051GB(Intel), AVR(Atmel), PIC16(Microchip), ST7, ST9 (SGS-T), HC08, HC11 (Motorola) и многие другие.

Модуль входного захвата (Input Capture)



Модуль входного захвата (Input Capture)

Данная схема предназначена для фиксации времени возникновения внешнего события: когда на внешнем выводе ICI происходит событие (перепад), определяемый настройкой схемы «детектора фронта», то текущее значение регистра-счетчика переписывается в регистр захвата, откуда может быть прочитано программно. Во многих реализациях захват может быть программно-управляемым – по команде обращения к специальному регистру.

Модуль входного захвата (Input Capture)

Тактирование регистра-счетчика чаще выбирается от сигнала внутренней синхронизации процессора Fint, то есть счетная часть модуля Input Capture настроена на режим подсчета времени – таймера. Но так же можно использовать и внешнее тактирование. По событию захвата устанавливается флаг ICF, может вырабатываться запрос прерывания. Кроме этого может быть перезагружен «0» или определенным значением регистр-счетчик.

С помощью схемы входного захвата удобно:

1. Определять период/частоту сигнала на входе ICI;
2. Фиксация относительного времени возникновения различных событий.

Схема выработки сигнала с ШИМ

Данная схема является модифицированным вариантом схемы выходного сравнения (Output Compare). Разница в том, что выходом управляет как компаратор, так и схема фиксации переполнения регистра-счетчика. Передний фронт сигнала с ШИМ (0→1) формируется по событию сравнения (когда регистр-счетчик равен регистру сравнения). Задний фронт (1 → 0) – по переполнению регистра-счетчика.

Модуль генератора сигнала ШИМ

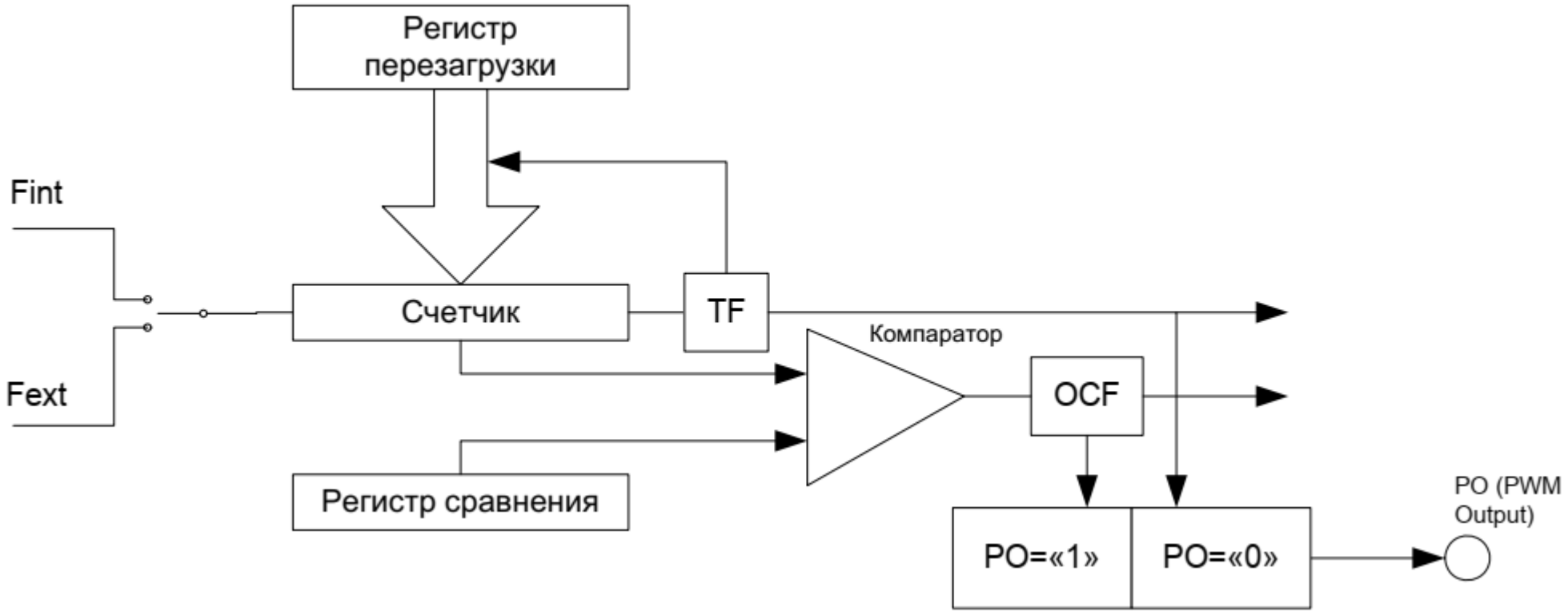


Схема выработки сигнала с ШИМ

Период сигнала с ШИМ равен частоте переполнения таймера и задается содержимым регистра перезагрузки. Длительность положительного импульса в периоде определяется как разница (максимального значения регистра-счетчика +1) и содержимого регистра сравнения.

В различных процессорах могут использоваться схемы генераторов ШИМ немного отличающиеся от данной.

Процессоры событий

Под управлением единого счетчика могут быть объединены несколько каналов входного захвата/ выходного сравнения/ формирования сигналов ШИМ. Каждый из каналов может быть индивидуально настроен на один из перечисленных режимов. Такие сложные блоки называют процессорами событий, а также: массивом программируемых счетчиков – PCA (Programmable Counter Array) (Intel), блоком CAPCOM (Infineon), блоком TIM8 (Motorola).

Процессоры событий позволяют формировать взаимно синхронизированные выходные сигналы: с фиксированным сдвигом фаз или считывать временные сдвиги между событиями, как частный случай – сдвиг фазы.

Блок процессора событий

Аналого-цифровой преобразователь. Цифро-аналоговый преобразователь

Аналого-цифровой преобразователь

Модуль аналого-цифрового преобразования (АЦП, Analog-to-digital converter, ADC) предназначен для ввода в процессор аналоговых сигналов с датчиков физических величин и преобразования значения напряжения этих сигналов в двоичный код с целью дальнейшей программной обработки.

Простейшим одноразрядным двоичным АЦП является компаратор.

Аналого-цифровой преобразователь

Характеристики:

- Разрешение АЦП – минимальное изменение величины аналогового сигнала, которое может быть преобразовано данным АЦП. Обычно измеряется в вольтах, поскольку для большинства АЦП входным сигналом является электрическое напряжение.
- Разрядность АЦП характеризует количество дискретных значений, которые преобразователь может выдать на выходе.
- Частота дискретизации.
- Точность.
- Скорость преобразования.

Аналого-цифровой преобразователь

Аналоговый сигнал является непрерывной функцией времени, в АЦП он преобразуется в последовательность цифровых значений. Следовательно, необходимо определить частоту выборки цифровых значений из аналогового сигнала. Частота, с которой производятся цифровые значения, получила название частоты дискретизации АЦП.

Аналого-цифровой преобразователь

Непрерывно меняющийся сигнал с ограниченной спектральной полосой подвергается оцифровке (то есть значения сигнала измеряются через интервал времени T — период дискретизации) и исходный сигнал может быть точно восстановлен из дискретных во времени значений путём интерполяции.

Точность восстановления ограничена ошибкой квантования. Однако в соответствии с теоремой Котельникова-Шеннона (https://www.graphicon.ru/oldgr/courses/cg_el00/kotelnikov.pdf) точное восстановление возможно только, если частота дискретизации выше, чем удвоенная максимальная частота в спектре сигнала.

Аналого-цифровой преобразователь

Поскольку реальные АЦП не могут произвести аналого-цифровое преобразование мгновенно, входное аналоговое значение должно удерживаться постоянным, по крайней мере, от начала до конца процесса преобразования (этот интервал времени называют время преобразования). Эта задача решается путём использования специальной схемы на входе АЦП – устройства выборки хранения – УВХ. УВХ, как правило, хранит входное напряжение в конденсаторе, который соединён со входом через аналоговый ключ: при замыкании ключа происходит выборка входного сигнала (конденсатор заряжается до входного напряжения), при размыкании – хранение. Многие АЦП, выполненные в виде интегральных микросхем, содержат встроенное УВХ.

Аналого-цифровой преобразователь

Полученное в результате преобразования значение записывается в регистр данных (РД). АЦП, интегрированные на кристалл процессора, обычно строят по схеме последовательного приближения. Время преобразования обычно составляет несколько десятков микросекунд, в зависимости от частоты тактирования АЦП. Завершение процесса преобразования отмечается установкой флага Fацп и (если разрешено) вырабатывается запрос прерывания. В современных управляющих процессорах и микроконтроллерах наиболее распространены АЦП с разрядностью 8, 10, реже 12 и совсем редко 14 и 16 бит.

Аналого-цифровой преобразователь

Аналоговый коммутатор выбирает один из возможных аналоговых входов (выводов) и подключает его к входу внутреннего АЦП для преобразования. При последовательной выборке каналов создается имитация многоканального АЦП. Применение действительно многоканальных АЦП резко повышает энергопотребление и стоимость процессора и обычно не используется (Если требуется несколько каналов и высокая скорость преобразования, то используют микросхему внешнего АЦП).

Аналого-цифровой преобразователь

Код выбора канала может формироваться программно, то есть программист «вручную» переключается между каналами, или аппаратно (автоматически), последовательно перебирая каналы (режим сканирования).

Для большего удобства использования модуля АЦП в режиме сканирования могут быть реализованы несколько регистров данных (Fujitsu MB90, Intel 8051GB), по одному на канал. Программисту будет достаточно считывать данные из регистра, соответствующего требуемому каналу. При этом код выбора канала параллельно подается на адресные входы блока регистров данных.

Источник опорного напряжения Vref и коммутатор Vref

Опорное напряжение Vref определяет диапазон значений напряжения на аналоговых входах и разрешающую способность АЦП, равную Vref/2n , где n – разрядность АЦП. Если значение напряжение на входе не велико, то точность преобразования может быть увеличена путем уменьшения Vref. Диапазон допустимых значений Vref обычно находится в рамках значения напряжения питания процессора.

Источник опорного напряжения Vref и коммутатор Vref

Могут быть использованы опорные источники следующего типа:

1. Внешний, подключаемые через специальные выводы микросхемы;

2. Внутренний фиксированный или программируемый (с помощью встроенного ЦАП).

Подключение к АЦП внешнего или внутреннего источников выполняется с помощью коммутатора Vref.

Источник опорного напряжения Vref и коммутатор Vref

Коммутатор сигнала запуска АЦП позволяет выбрать способ запуска процесса преобразования, а также определяет один из возможных режимов

работы АЦП:

1. Периодического преобразования. В этом режиме АЦП запускается периодическим сигналом от основного тактового генератора или встроенного таймера.

2. Если сигнал запуска подать на двоичный счетчик, выходами подключенный к управляющим входам аналогового коммутатора и адресным линиям блока регистров данных, то таким образом легко реализовать режим последовательного сканирования каналов.

3. Внешнего запуска. Запуск осуществляется внешним сигналом, что позволяет четко определить момент считывания значения аналогового напряжения со входа.

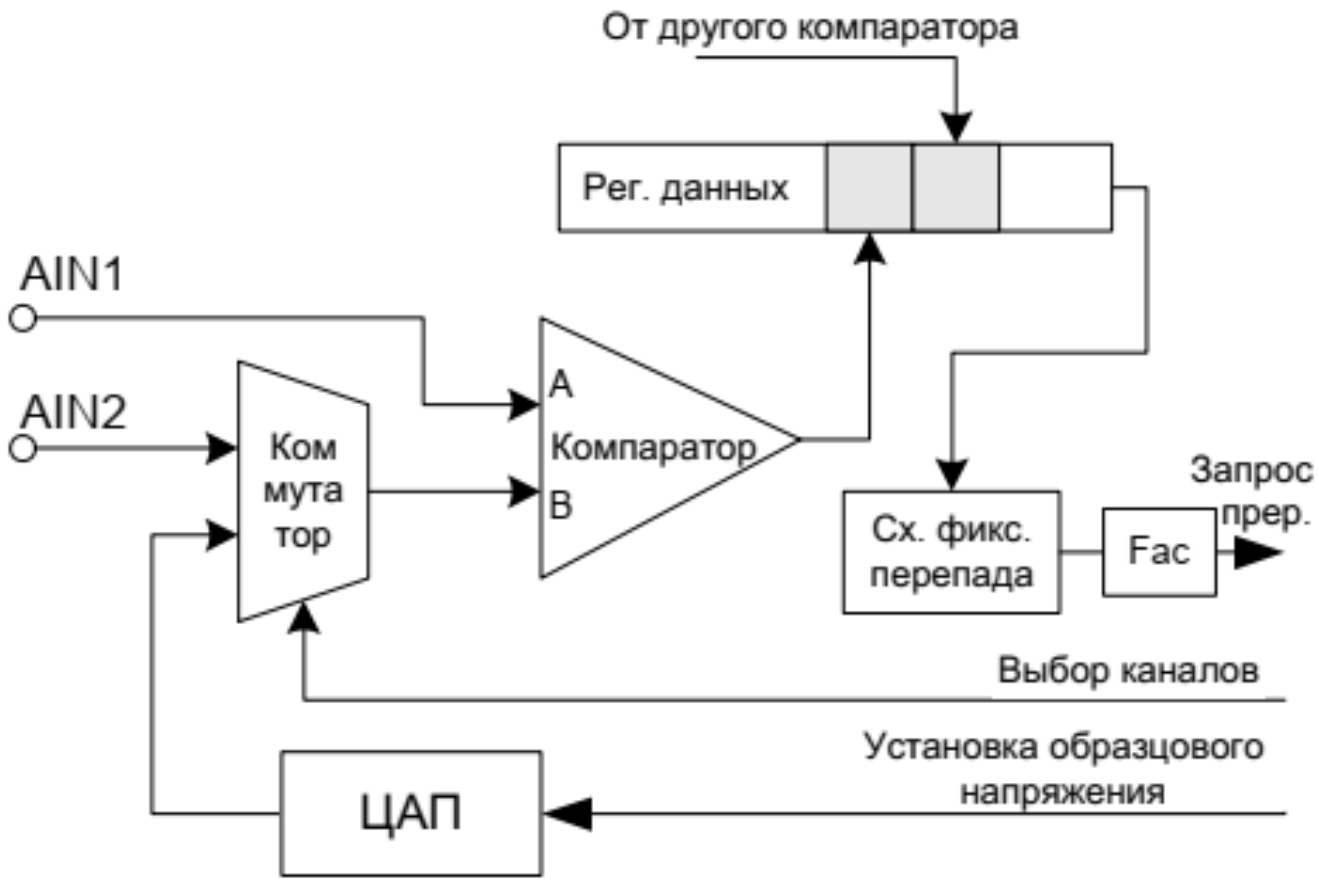
4. Программно управляемого запуска, по установке специального бита.

Блок управления модулем АЦП конфигурирует и синхронизирует функционирование других (вышеперечисленных) блоков, управляется программно, через регистры специального назначения.

Аналоговый компаратор

Аналоговый компаратор используется для сравнения напряжения двух внешних аналоговых сигналов или для сравнения напряжения внешнего аналогового сигнала с образцовым напряжением, вырабатываемым внутри процессора. Могут быть запрограммированы различные уровни образцового напряжения. Результат сравнения кодируется битом в регистре специального назначения, например, “1” – вход А больше или равно чем В, “0” – вход А меньше чем В. В случае изменения соотношения изменяется значение бита, а также может быть установлен флаг и выработан запрос прерывания.

Модуль аналогового компаратора



Аналоговый компаратор

Аналоговый коммутатор входов выбирает аналоговые сигналы для сравнения. Один сигнал берется с внешнего входа AIN1, в качестве второго берется или сигнал с внешнего входа AIN2, или образцовое внутреннее напряжение, которое вырабатывается с помощью ЦАП.
ЦАП – программируемый генератор образцового напряжения.
Регистр данных – программно доступный регистр, в битах которого сохраняются результаты сравнения одного или нескольких компараторов.
Схема фиксации перепада определяет изменение одного из бит в регистре данных (выхода одного из компараторов) и вырабатывает по этому событию запрос прерывания.

Аналоговый компаратор

Пример использования аналогового компаратора:

• Контроль превышения допустимых значений температуры, давления, тока, напряжения и других физических величин. Физическая величина преобразуется в напряжение с помощью датчика и контролируется с помощью аналогового компаратора. Порог сравнения устанавливается встроенным генератором образцового напряжения.

• Обнаружение (формирование) фронтов внешних сигналов.

• Встроенные схемы контроля напряжения питания системы.

Классификация АЦП

В настоящее время известно большое число методов преобразования напряжение-код. Эти методы существенно отличаются друг от друга потенциальной точностью, скоростью преобразования и сложностью аппаратной реализации.

В основу классификации АЦП положен признак, указывающий на то, как во времени разворачивается процесс преобразования аналоговой величины в цифровую. В основе преобразования выборочных значений сигнала в цифровые эквиваленты лежат операции квантования и кодирования. Они могут осуществляться с помощью либо последовательной, либо параллельной, либо последовательно-параллельной процедур приближения цифрового эквивалента к преобразуемой величине.

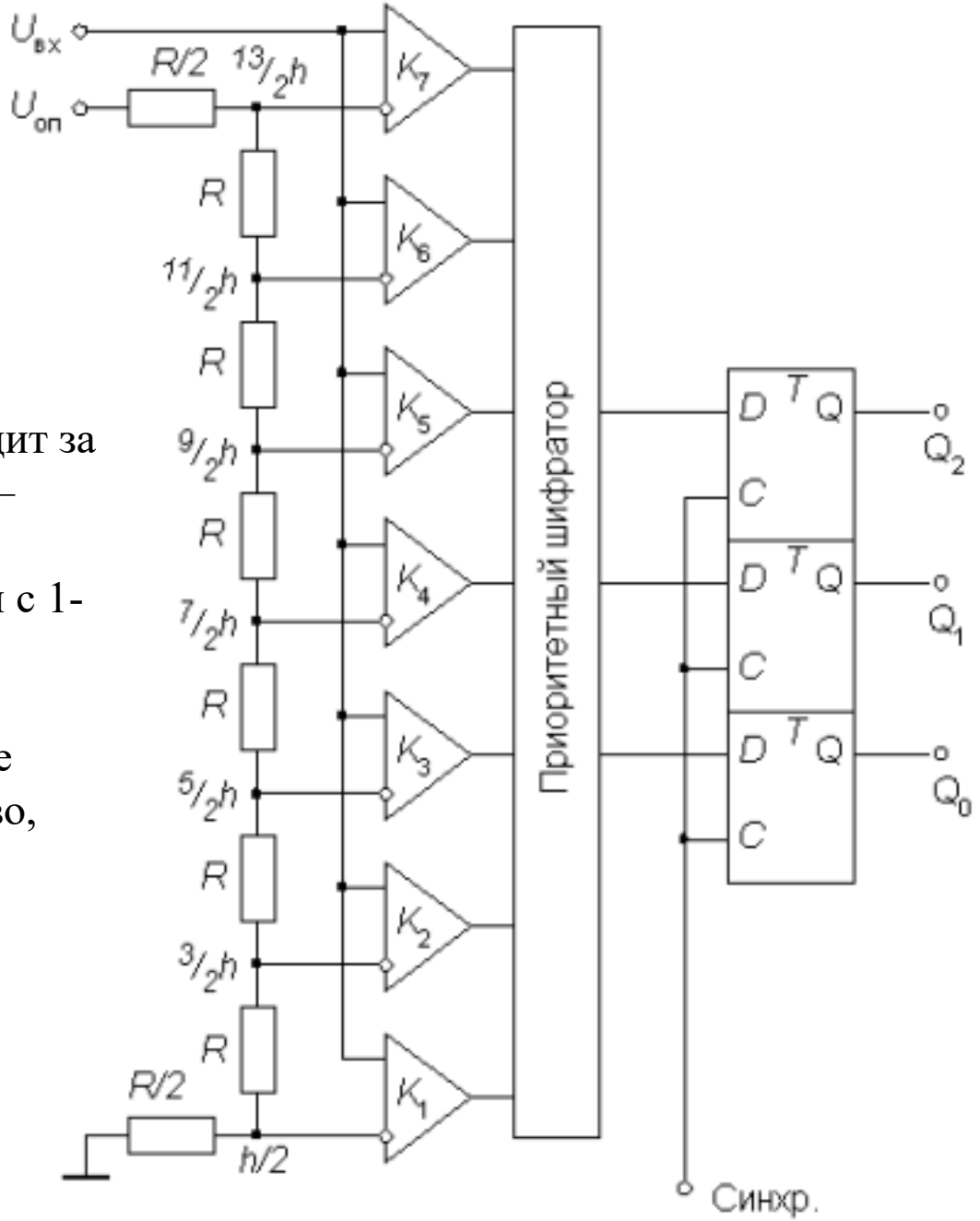
Классификация АЦП по методам преобразования

Параллельные АЦП

АЦП этого типа осуществляют квантование сигнала одновременно с помощью набора компараторов, включенных параллельно источнику входного сигнала. На следующем рисунке показана реализация параллельного метода АЦ преобразования для 3-разрядного числа. С помощью трех двоичных разрядов можно представить восемь различных чисел, включая нуль. Необходимо, следовательно, семь компараторов. Семь соответствующих эквидистантных опорных напряжений образуются с помощью резистивного делителя.

Схема параллельного АЦП

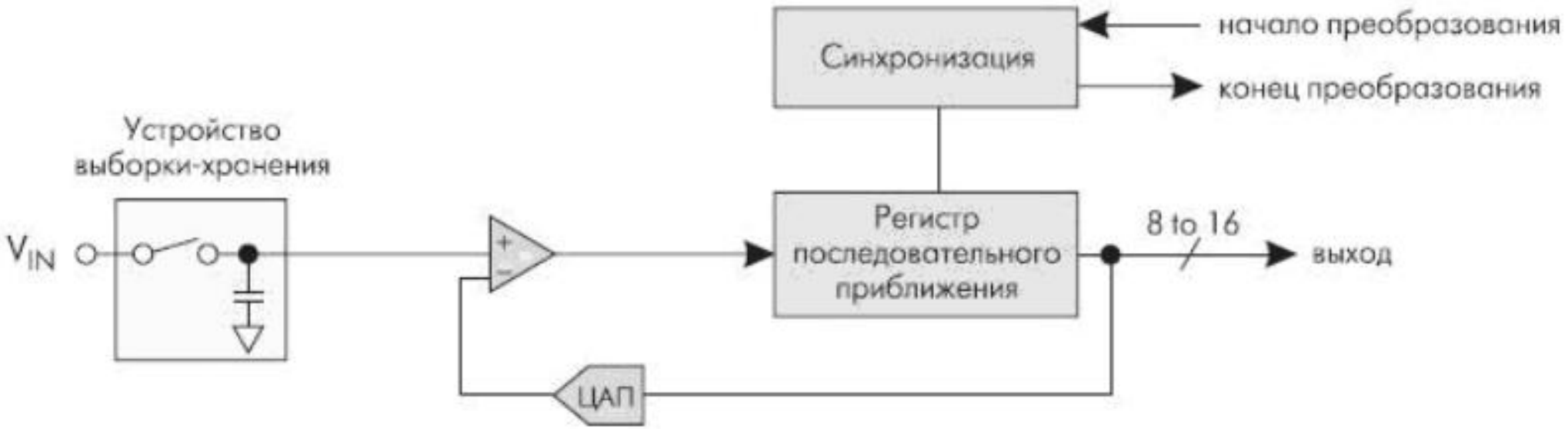
Если приложенное входное напряжение не выходит за пределы диапазона от 5/2h, до 7/2h, где h=Uоп/7 – квант входного напряжения, соответствующий единице младшего разряда АЦП, то компараторы с 1-го по 3-й устанавливаются в состояние 1, а компараторы с 4-го по 7-й – в состояние 0. Преобразование этой группы кодов в трехзначное двоичное число выполняет логическое устройство, называемое приоритетным шифратором.



АЦП последовательного приближения

АЦП последовательного приближения (successive approximation architecture, SAR) или АЦП с поразрядным уравновешиванием содержит компаратор, вспомогательный ЦАП и регистр последовательного приближения. АЦП преобразует аналоговый сигнал в цифровой за N шагов, где N — разрядность АЦП.

АЦП последовательного приближения



АЦП последовательного приближения

На каждом шаге определяется по одному биту искомого цифрового значения, начиная от старшего значащего разряда (СЗР) и заканчивая младшим значащим разрядом (МЗР). Последовательность действий по определению очередного бита заключается в следующем. На вспомогательном ЦАП выставляется аналоговое значение, образованное из битов, уже определённых на предыдущих шагах; бит, который должен быть определён на этом шаге, выставляется в 1, более младшие биты установлены в 0. Полученное на вспомогательном ЦАП значение сравнивается с входным аналоговым значением. Если значение входного сигнала больше значения на вспомогательном ЦАП, то определяемый бит получает значение 1, в противном случае 0. Таким образом, определение итогового цифрового значения напоминает двоичный поиск. АЦП этого типа обладают одновременно высокой скоростью и хорошим разрешением. Однако при отсутствии устройства выборки хранения погрешность будет значительно больше (представьте, что после оцифровки самого большого разряда сигнал начал меняться).

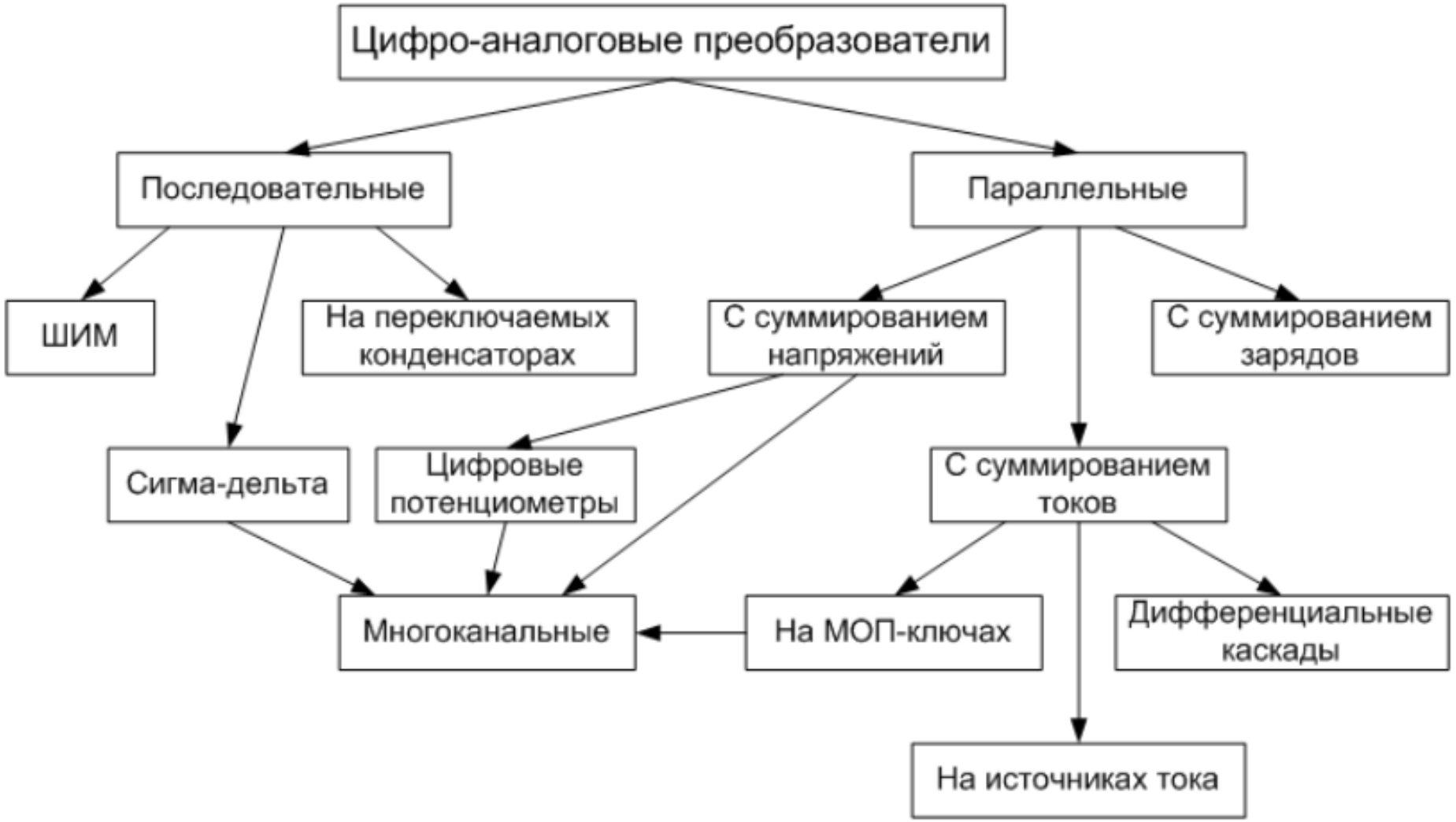
Характеристика АЦП последовательного приближения: невысокая скорость преобразования, невысокая цена и низкое энергопотребление.

Цифро-аналоговый преобразователь

Цифро-аналоговый преобразователь (digital-analog converter, DAC) предназначен для преобразования числа, представленного, как правило, в виде двоичного кода, в напряжение или ток, пропорциональные этому числу.

Схемотехника цифро-аналоговых преобразователей весьма разнообразна. На следующем рисунке ниже представлена общая классификация ЦАП по способам преобразования входного кода и схемам формирования выходного сигнала.

Обобщенная классификация ЦАП



Цифро-аналоговый преобразователь

Дальнейшую классификацию цифро-аналоговых преобразователей можно провести по ряду специфических признаков, например:

- по роду выходного сигнала: преобразователи с токовым выходом или с выходом по напряжению;
- по типу цифрового интерфейса: с последовательным вводом или с параллельным вводом;
- по числу ЦАП на кристалле: одноканальные и многоканальные;
- по быстродействию: низкого, среднего и высокого быстродействия;
- по разрядности.

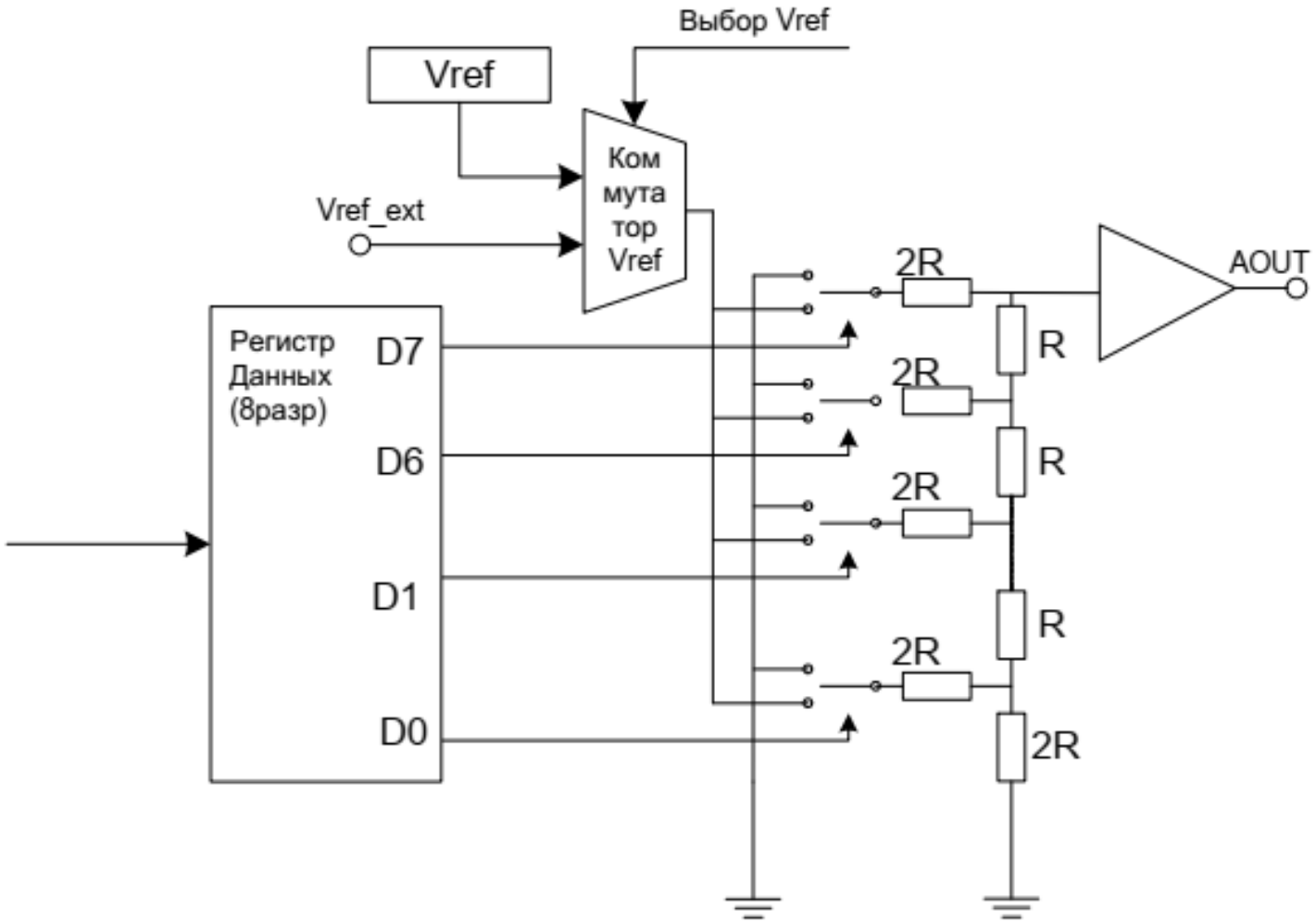
Цифро-аналоговый преобразователь

Матрица R-2R – самый распространенный метод цифро-аналогового преобразования. Матрица работает по принципу деления входного напряжения на входах. Матрица имеет число входов по числу разрядов регистра данных. На каждый вход через ключ может быть подано опорное напряжение Vref или 0В. Ключи управляются разрядами регистра данных: “1” – на матрицу подается Vref, “0” – подается 0В.

Коммутатор опорного напряжения Vref позволяет выбрать внешний или встроенный источник опорного напряжения.

В регистр данных записывается цифровой код. Регистр данных определяет разрядность ЦАП.

Модуль ЦАП с типом преобразования «Матрица R-2R»



Цифро-аналоговый преобразователь

На практике ЦАП применяется для управления различными исполнительными устройствами (приводами) и системами: электродвигателями постоянного тока с переменной скоростью вращения, источниками питания с управляемым напряжением, различными индикаторами и т.п. С помощью ЦАП можно синтезировать аналоговые сигналы различной формы, например, синусоидальной.

Контроллеры последовательных интерфейсов

Контроллеры последовательных интерфейсов

Контроллеры последовательных интерфейсов ориентированы на решение следующих задач:

• Связь встраиваемой микропроцессорной системы с системой управления верхнего уровня: промышленным или офисным компьютером, программируемым контроллером. Наиболее часто для этих целей используют интерфейсы RS-232C, RS-422, USB, IrDA.

• Связь с внешними по отношению к микропроцессору периферийными микросхемами (памяти EEPROM, часов реального времени (RTC) и т.д.), а также с различными датчиками с последовательным цифровым выходом. Для этих целей наиболее часто применяются интерфейсы SPI, I2C, MicroWire, uLAN и другие.

• Интерфейс связи с локальной сетью в распределенных информационно-управляющих системах. В этой сфере находят применение интерфейсы RS-232C, RS-485, I2C, uLAN, CAN, Ethernet.

• Внутрисистемное программирование резидентной памяти программ (OTPROM, EPROM, FLASH) или данных (EEPROM) у процессоров для встраиваемых применений. Обычно для этого используется интерфейс RS-232C (ADuC (Analog Devices), MB90Fxxx (Fujitsu), MSP430 (Texas Instruments)) или SPI (AVR(Atmel)).

Контроллеры последовательных интерфейсов

В настоящее время встроенные контроллеры последовательных интерфейсов имеются почти у всех встраиваемых процессоров, исключая простейшие 8-16 выводные микросхемы. У большинства процессоров имеются несколько таких модулей одного или различных типов.
Среди контроллеров последовательного обмена стандартом «де-факто» стал модуль универсального синхронно-асинхронного приемопередатчика (Universal Synchronous/Asynchronous Receiver and Transmitter, USART). В названии часто опускают слово «синхронный» и модуль не совсем корректно именуется UART (чисто асинхронные приемопередатчики сейчас встречаются достаточно редко). Характеристики последовательного порта UART не позволяют производить приём и передачу данных за пределы печатной платы.

Контроллеры последовательных интерфейсов

Для связи с другими устройствами, сигнал от UART необходимо пропустить через приёмопередатчик, работающий в одном из стандартов:

- RS-232;
- RS-485;
- RS-422.

Обычно модули UART в асинхронном режиме поддерживают протокол обмена для интерфейса RS-232 (8N1 или 9N1); в синхронном режиме – нестандартные синхронные протоколы, в некоторых случаях – протокол SPI. Приемопередатчик – преобразователь уровня, как правило, выполненный в интегральном исполнении. Предназначен для преобразования электрических сигналов из уровня ТТЛ в уровень, соответствующий физическому уровню определенного стандарта.

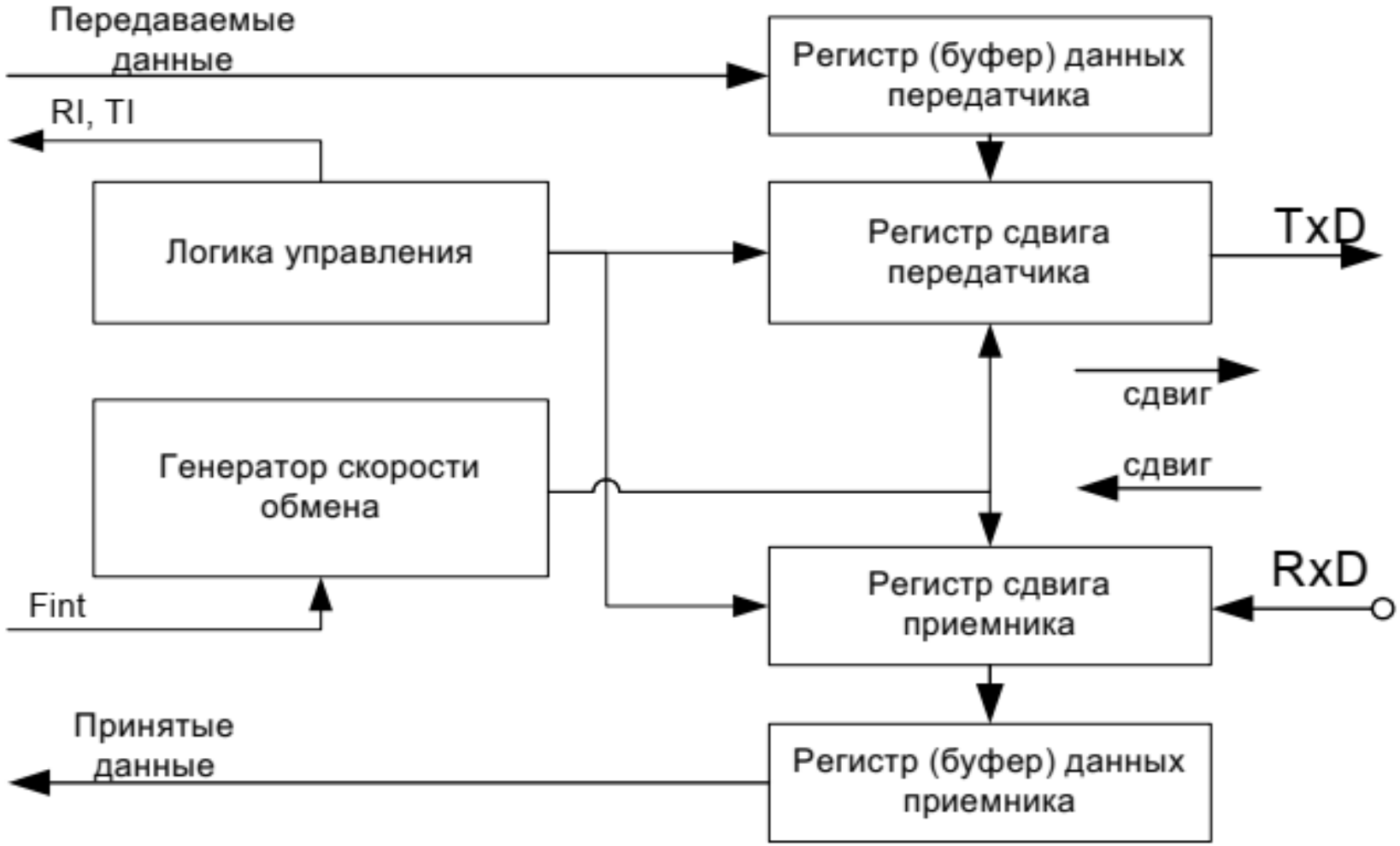
Контроллеры последовательных интерфейсов

Контроллер UART обычно содержит:

- Источник тактирования (обычно с увеличенной частотой тактирования по сравнению со скоростью обмена, чтобы иметь возможность отслеживать состояние лини передачи данных в середине передачи бита).
- Входные и выходные сдвиговые регистры.
- Регистры управления приемом/передачей данных; чтением/записью.
- Буферы приема/передачи.
- Параллельная шина данных для буферов приема/передачи.
- FIFO буферы памяти (опционально).

Упрощенная структура приемопередатчика типа UART представлена на следующем рисунке.

Модуль UART


Передаваемые данные
RI, TI
Регистр (буфер) данных передатчика
Логика управления
Регистр сдвига передатчика
TxD
сдвиг
сдвиг
Генератор скорости обмена
Fint
Регистр сдвига приемника
RxD
Принятые данные
Регистр (буфер) данных приемника

Контроллеры последовательных интерфейсов

Генератор скорости обмена представляет собой делитель внутренней тактовой частоты процессора Fint с плавно или пошагово (дискретно) программируемым коэффициентом деления. При «плавном» программировании можно настраивать требуемую скорость вне зависимости (в определенных пределах) от частоты Fint. Для этого используется стандартный или специально выделенный таймер-счетчик в режиме автоперезагрузки. В случае «фиксированных» коэффициентов деления для поддержания стандартного ряда скоростей необходимо выбирать определенную частоту тактирования процессора.

Контроллеры последовательных интерфейсов

С выхода генератора скорости сигнал синхронизации поступает на вход тактирования приемного и передающего сдвиговых регистров, которые осуществляют последовательную выдачу/прием бит данных с заданной скоростью. Полностью принятый байт попадает в регистр – буфер данных приемника. Байт для передачи помещается в сдвиговый регистр из буфера передатчика.

Процессы приема и передачи в асинхронном режиме UART происходят независимо. Таким образом, поддерживается дуплексный режим обмена. Однако требуется, чтобы приемник и передатчик были настроены на одну скорость.

Более простым является функционирование в синхронном режиме. Здесь каждый принимаемый/передаваемый бит стробируется специальным сигналом и нет необходимости точно согласовывать скорость приемника и передатчика.

Контроллеры последовательных интерфейсов

Ошибки UART:

• **Overrun Error** (ошибка из-за повышенной скорости передачи, переполнение буфера приема). Эта ошибка случается, когда приемник UART не успевает обрабатывать приходящие из канала символы, т. е. буфер переполняется.

• **Framing Error** (ошибка кадрирования). Эта ошибка случается, когда фиксируется некорректное состояние линии данных в момент передачи старт- или стоп-бита. Например, после передачи 8 бит данных приемник ожидает перехода линии в стоп-состояние, но этого не происходит.

• **Break Condition** (сигнал прерывания передачи, разрыва связи). Этот сигнал информирует о том, что входная линия данных находилась в неизменном нулевом состоянии в течение времени, больше передачи одного символа. В буфере приема нулевой байт. Некоторые устройства используют такую последовательность, чтобы сообщить передатчику, например, о переходе на другую скорость обмена данными.

Контроллеры последовательных интерфейсов

Еще более упрощается функционирование в режиме SPI: приемник и передатчик работают синхронно: приему одного бита соответствует передача одного бита, начало передачи байта совпадает с началом приема, за сеанс обмена происходит прием одного байта и передача одного байта.

В большинстве случаев приемопередатчики работают с входными и выходными сигналами уровней TTL. Формирование физических сигналов с уровнями напряжения и тока, соответствующими реализуемому интерфейсу, выполняется с помощью специальных микросхем – трансиверов или адаптеров физического интерфейса. Например: MAX232 (MAXIM) – RS-232C, MAX485 (MAXIM) – RS-422/485, PCA82C251 (Philips) – CAN.

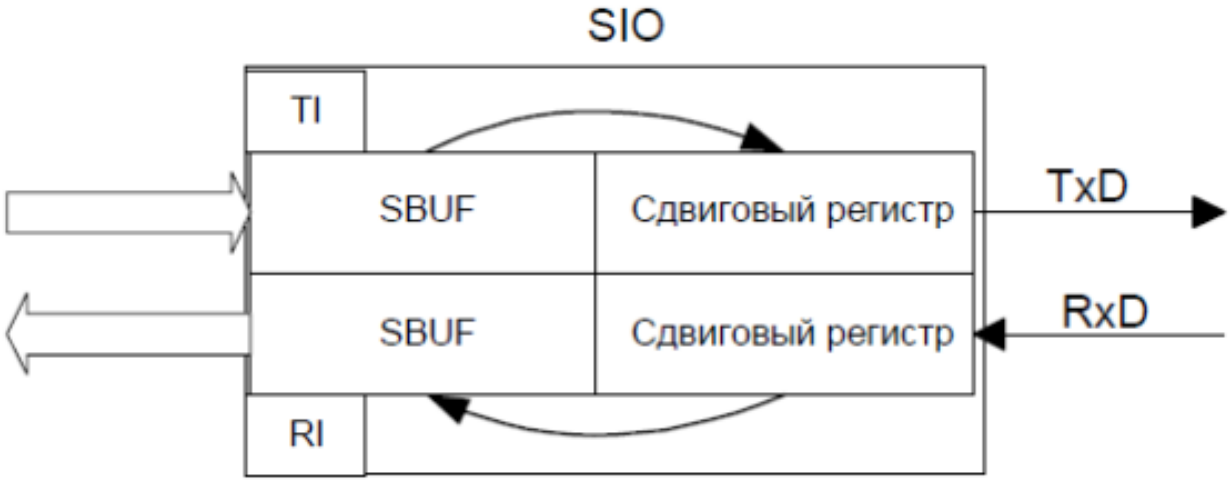
Кроме рассмотренных приемопередатчиков USART во встраиваемых процессорах широко используются другие интерфейсы, например, USB, CAN.

Контроллер последовательного интерфейса в микроконтроллере с ядром Intel MCS-51

Последовательный порт в микроконтроллерах MCS-51 позволяет осуществлять последовательный дуплексный ввод-вывод в синхронном и асинхронном режимах с разными скоростями обмена. Помимо обычного вводавывода, в нем предусмотрена аппаратная поддержка взаимодействия нескольких микроконтроллеров.

Контроллер последовательного интерфейса в микроконтроллере с ядром Intel MCS-51

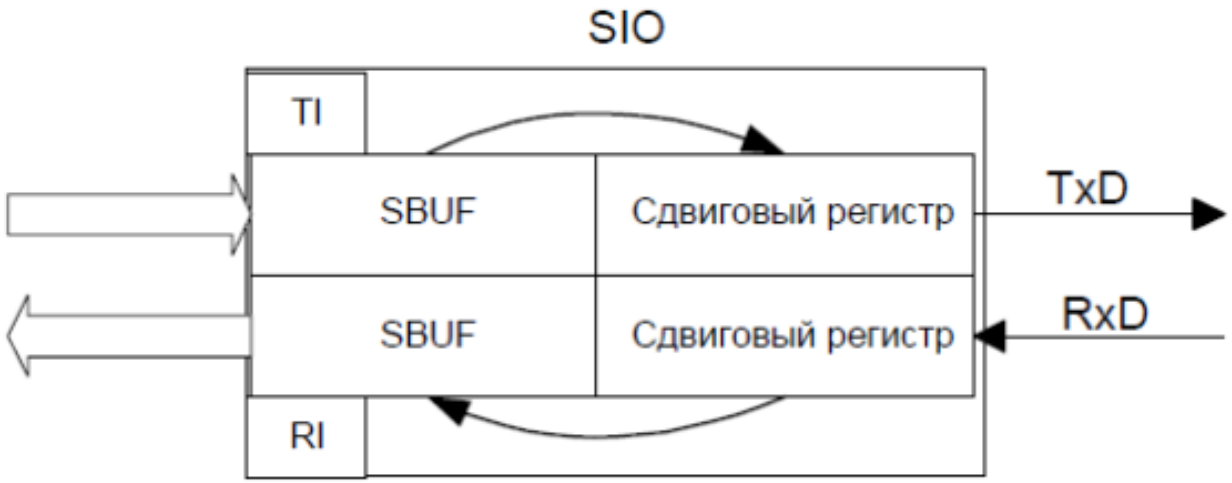
Схематически контроллер последовательного порта представлен на рисунке. Как видно из рисунка, регистр SBUF, доступный пользователю для чтения и записи, есть на самом деле два регистра, в один из которых можно только записывать, а из другого – читать. Контроллер позволяет дуплексный обмен данными, т.е. одновременно может передавать и принимать информацию по линиям TxD (P3.1) и RxD (P3.0) соответственно.



Передача инициируется записью в SBUF байта данных. Этот байт переписывается в сдвиговый регистр, из которого пересылается бит за битом. Как только байт будет переслан целиком, устанавливается флаг TI, сигнализирующий о том, что контроллер готов к передаче очередного байта.

Контроллер последовательного интерфейса в микроконтроллере с ядром Intel MCS-51

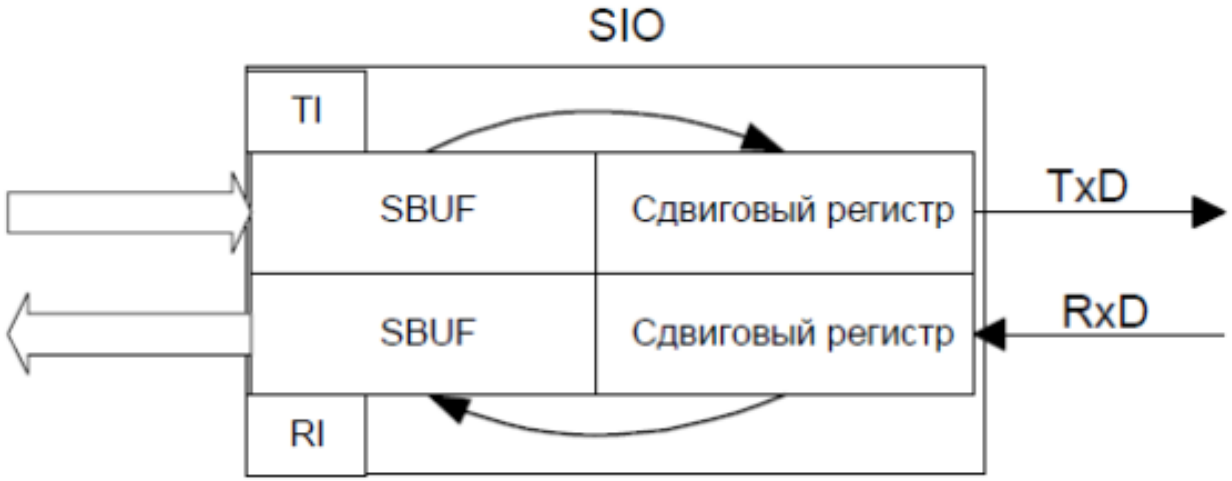
Прием осуществляется в обратном порядке: после начала процесса приема принятые биты данных последовательно сдвигаются в сдвиговом регистре, пока не будет принято их установленное количество, затем содержимое сдвигового регистра переписывается в SBUF и устанавливается флаг RI. После этого возможен прием следующей последовательности бит. Необходимо отметить, что запись принятого байта из сдвигового регистра в SBUF происходит только при условии, что RI=0, поэтому при заборе пользовательской программой очередного байта из SBUF ей необходимо сбрасывать этот флаг, иначе принятый в сдвиговый регистр, но не записанный в SBUF, следующий байт будет безвозвратно утерян.



Контроллер последовательного интерфейса в микроконтроллере с ядром Intel MCS-51

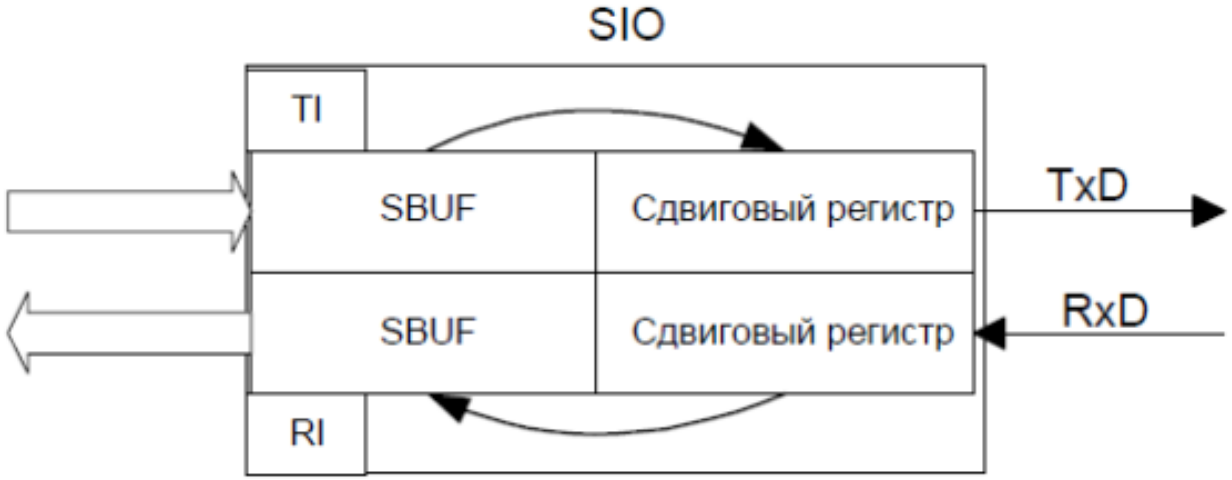
Контроллер последовательного порта может работать в одном из четырех режимов (один синхронный и три асинхронных), различающихся скоростями обмена (бод) и количеством передаваемых/принимаемых бит:

- Режим 0 (синхронный): данные передаются и принимаются через RxD, TxD является синхронизирующим (выдает импульсы сдвига). Скорость в этом режиме фиксирована (1/12 частоты тактового генератора).
- Режим 1 (8 бит данных, асинхронный, переменная скорость).Передаются 10 бит: старт-бит, 8 бит данных (SBUF) и стоп-бит.



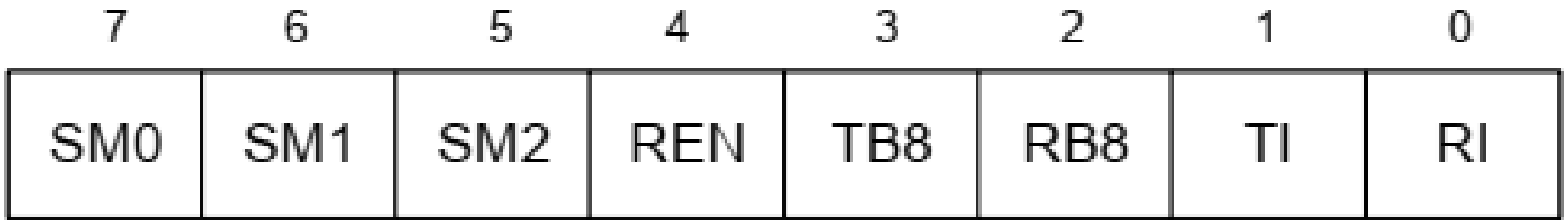
Контроллер последовательного интерфейса в микроконтроллере с ядром Intel MCS-51

- Режим 2 (9 бит данных, асинхронный, фиксированная скорость). Передаются 11 бит: старт-бит, 8 бит (SBUF), бит TB8/RB8 (посылка/прием соответственно) и 1 стоп-бит. Биты TB8 и RB8 содержатся в регистре SCON (биты 3 и 2 соответственно), первый устанавливается программно, а второй содержит 9-й бит принятой комбинации. С помощью них можно организовать, например, контроль четности или обмен с двумя стоп-битами.
- Режим 3 (9 бит данных, асинхронный, переменная скорость). То же, что и режим 2, только скорость обмена переменная. В режиме 1 и 3 используются Таймер 0 и/или Таймер 1 для настройки скорости обмена.

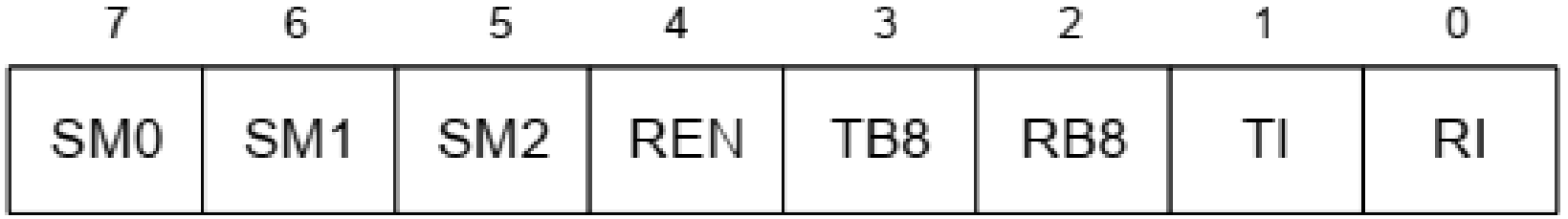


Контроллер последовательного интерфейса в микроконтроллере с ядром Intel MCS-51

Режим контроллера UART, а также другие параметры его работы задаются в регистре специального назначения SCON (98h).

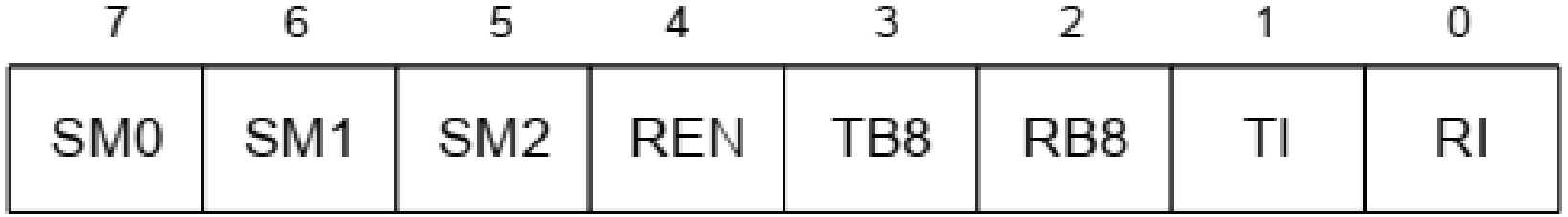


Описание регистра специального назначения SCON



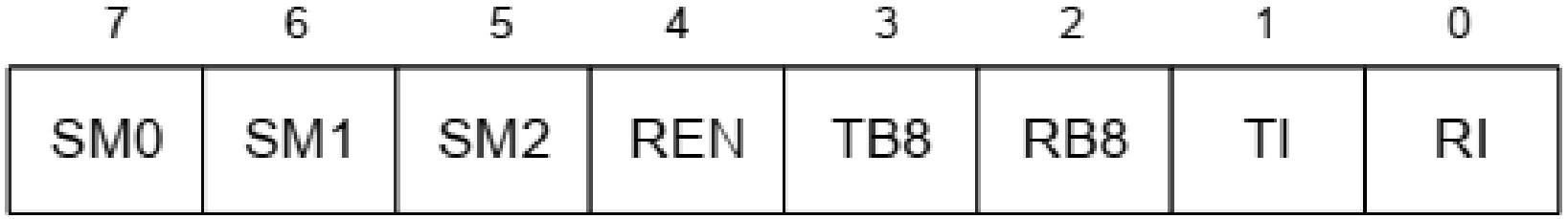
Бит	Описание
SM0 **SM1**	Режим работы UART: SM0 SM1 Выбранный режим работы 0 0 Режим 0 0 1 Режим 1 1 0 Режим 2 1 1 Режим 3
SM2	Включает поддержку взаимодействия нескольких микроконтроллеров в режимах 2 и 3. В Режиме 0 бит SM2 должен быть сброшен (SM2=0). При приеме в Режиме 1 при SM2=1 RI не устанавливается, если не был принят правильный стоп-бит; если же SM2=0 RI установится как только будет принят байт данных. Если SM2=1, то при приеме в режимах 2 и 3 RI не устанавливается в том случае, если принятый 9-й бит (RB8) равен 0.

Описание регистра специального назначения SCON



REN	(Receiver Enable) Если установлен, то разрешен прием данных. Устанавливается и сбрасывается программно.
TB8	(Transmitter Bit 8) 9-й бит посылаемых данных в режимах 2 и 3. Устанавливается и сбрасывается программно.
RB8	(Receiver Bit 8) 9-й бит принимаемых данных в режимах 2 и 3. В режиме 1, если SM2=0, в RB8 записывается принятый стоп-бит (точнее то, что было принято в момент приема стоп-бита). В режиме 0 не используется.
TI	(Transmitter Interrupt) Флаг завершения посылки. Устанавливается аппаратно по завершении посылки 8-го бита данных в режиме 0 или стоп-бита в других режимах. При ES = 1 (бит 4 регистра IEN0 (0A8h)) происходит прерывание №4 (вектор 023h), когда TI устанавливается контроллером в 1. Сбрасывается программно.

Описание регистра специального назначения SCON



RI	(Receiver Interrupt) Флаг завершения приема. Устанавливается аппаратно по завершении приема 8-го бита данных в режиме 0 или стоп-бита в других режимах. При ES = 1 (бит 4 регистра IEN0 (0A8h)) происходит прерывание №4 (вектор 023h), когда RI устанавливается контроллером в 1. Сбрасывается программно.

Контроллер последовательного интерфейса в микроконтроллере с ядром Intel MCS-51

Как было сказано выше, скорости обмена в разных режимах различаются. В двух из них скорости фиксированы, а в двух других переменные. Рассмотрим подробнее способы задания скорости обмена в каждом из режимов.

- Режим 0. Фиксированная скорость обмена. В этом режиме скорость вычисляется следующим образом $baudrate = \frac{Core\_Clk}{12}$, где Core_Clk есть внутренняя тактовая частота МК.

Контроллер последовательного интерфейса в микроконтроллере с ядром Intel MCS-51

• Режим 2. Скорость обмена зависит от значения бита SMOD (старший бит в регистре PCON (087h)). Если SMOD=0 (по умолчанию), то скорость определяется как 1/64 тактовой частоты. SMOD=1 удваивает это значение. В общем случае: $baudrate = \frac{2^{SMOD} \times f_{osc}}{64}$

• Режимы 1 и 3. В этих режимах порт можно синхронизировать от двух источников: от Таймера 1 или Таймера 2, или обоих (один для передачи, другой для приема байта данных). Источник синхронизации определяется значением бита BD (старший бит регистра ADCON(0D8h)): если BD=1, то используется встроенный генератор импульсов, в противном случае используется таймер 1.

Контроллер последовательного интерфейса в микроконтроллере с ядром Intel MCS-51

При использовании Таймера 1 скорость обмена определяется временем переполнения таймера и значением бита SMOD: $baudrate = \frac{2^{SMOD} \times TM1OVrate}{64}$

где TM1OVrate – время переполнения Таймера 1, зависящее от режима его работы. Таймер 1 может быть установлен как «таймер» или как «счетчик», в любом режиме работы. Прерывание от таймера при этом обычно запрещается.

Контроллер последовательного интерфейса в микроконтроллере с ядром Intel MCS-51

Чаще всего таймер 1 устанавливают как «таймер» в режиме 2 («auto-reload»): для этого старшая тетрада регистра TMOD (089h) должна равняться 0010b. При таком использовании после каждого переполнения регистра таймера TL1 в него загружается значение из регистра TH1. Скорость обмена в этом случае вычисляется по формуле:

$$baudrate = \frac{2^{SMOD}}{32} \times \frac{f_{osc}}{12 \times (256 - TH)}$$

Контроллер последовательного интерфейса в микроконтроллере с ядром Intel MCS-51

В следующей таблице приведены способы задания некоторых стандартных скоростей обмена:

Скорость	fosc, МГц	SMOD	Таймер 1 (TH1)
19200 бод	11.059	1	FD
9600 бод	11.059	0	FD
4800 бод	11.059	0	FA
2400 бод	11.059	0	F4

Подсистема синхронизации. Механизмы начальной инициализации встроенной памяти

Подсистема синхронизации

Подсистема синхронизации отвечает за формирование устойчивых сигналов синхронизации внутренних блоков процессора и внешних цепей (блоков) управляющих вычислительных систем, построенных на данном процессоре.

К внутренним блокам относятся:

- Вычислительное ядро;
- Периферийные устройства (таймеры/счетчики, блоки CCP, АЦП, ЦАП,приемопередатчики и др.);
- Схемы рестарта («сброса»).

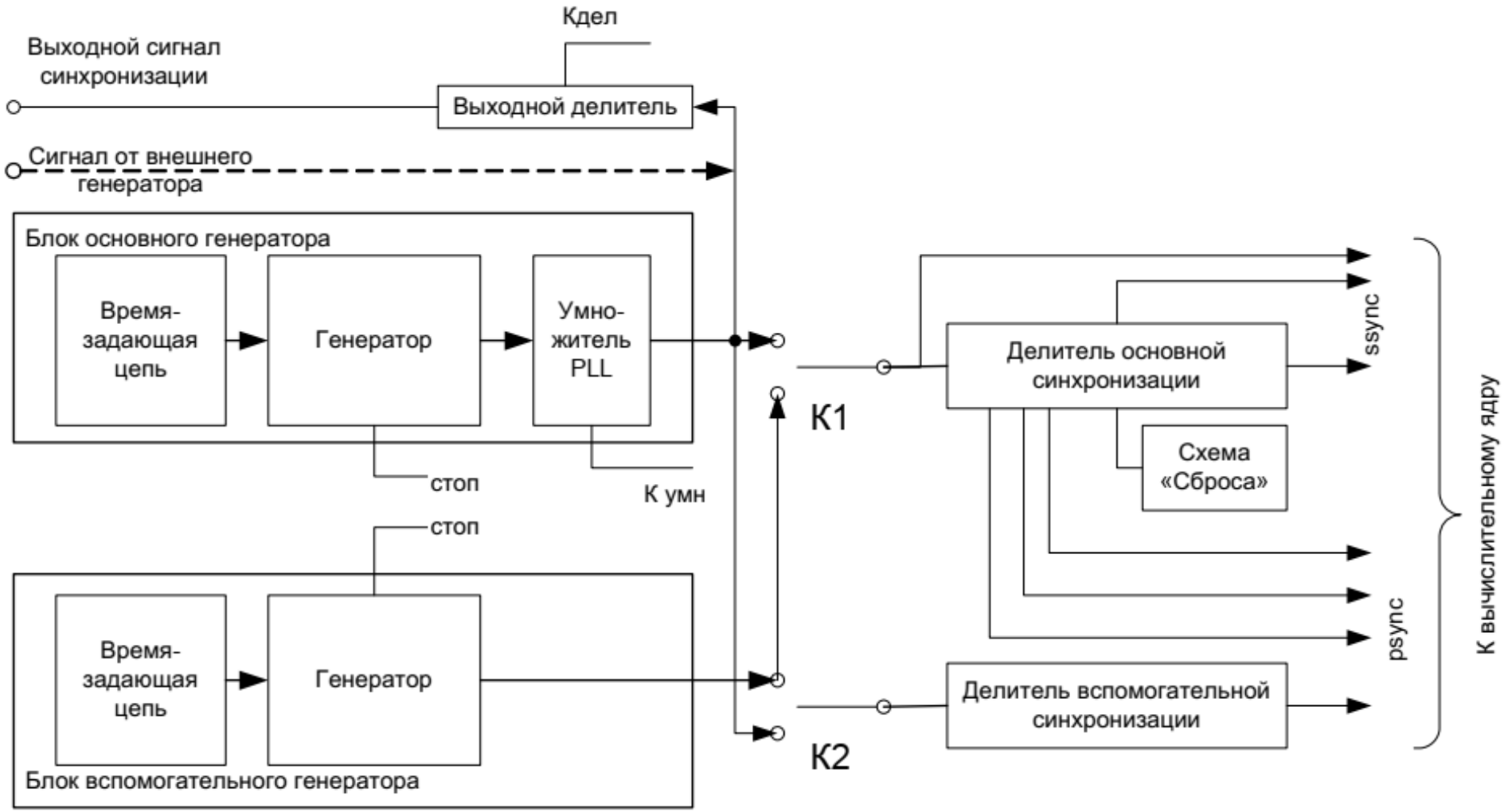
Подсистема синхронизации

Для синхронизации внешних схем (периферийных контроллеров, интерфейсных микросхем, блоков программируемой логики и др.) из процессора на специальную ножку выводится сигнал синхронизации.

Обобщенная структура подсистемы синхронизации встраиваемых процессоров приведена на рисунке.

Подсистема синхронизации включает блок генераторов синхроимпульсов (основной и вспомогательный генераторы, схему формирователя выходного сигнала синхронизации) и формирователь внутренних синхросигналов процессора (коммутаторы К1 и К2, делители, схему «сброса»).

Подсистема синхронизации



Подсистема синхронизации

Блок основного генератора вырабатывает сигналы синхронизации для вычислительного ядра, большинства периферийных устройств, схемы рестарта.

Блок включает собственно схему генератора, внешнюю или встроенную времязадающую цепочку для генератора и (не всегда) схему умножителя частоты.

Основная частота синхронизации может изменяться в очень широких пределах – от десятков килогерц до десятков и сотен мегагерц. Невысокая частота (до 1 МГц) используется в системах с пониженным энергопотреблением.

Подсистема синхронизации

В зависимости от выбранной рабочей частоты, требований точности и стабильности параметров сигнала синхронизации необходимо могут быть использованы различные типы времязадающих цепочек, которые в свою очередь требуют перенастройки режимов работы генераторов. Выбор режима работы генератора и типа времязадающей цепочки осуществляется программированием специальных конфигурационных бит во встроенной памяти программ микропроцессора.

Подсистема синхронизации

Времязадающие цепочки подключаются к специальным выводам микропроцессора. В некоторых моделях имеются встроенные цепочки. Наиболее часто используются следующие типы времязадающих цепочек:

1. Кварцевый резонатор: частоты от десятков килогерц до десятков мегагерц, высокая стабильность частоты (погрешность – сотые/тысячные доли процента), относительно высокая цена;

2. Пьезокерамический резонатор: частоты от десятков килогерц до единиц мегагерц, средняя стабильность частоты (погрешность – десятые доли процента), невысокая цена;

3. LC-цепь: частоты единицы-сотни килогерц, средняя стабильность частоты (погрешность – десятые доли/единицы процента), невысокая цена;

4. RC-цепь: частоты единицы-сотни килогерц, низкая стабильность частоты (погрешность – единицы процента), низкая цена, часто реализуется как встроенная времязадающая цепочка;

Подсистема синхронизации

В случаях высокой частоты (свыше 30 МГц) рекомендуется встроенный генератор отключать полностью и подключать внешний. Так же можно поступать, если от одного внешнего генератора синхронизируется несколько схем, включая процессор.

Работа генератора на высокой частоте ведет за собой следующие трудности: сложность «запуска» встроенного генератора (на частотах свыше 30 МГц), специальные требования к трассировке и качеству печатных плат, высокий уровень помех от внешних высокочастотных цепей (например, цепей подключения кварцевого резонатора), невозможность оперативной перестройки частоты. Для избежания этих проблем почти во всех 16/32-разрядных процессорах используются встроенные цифровые умножители частоты с программируемым коэффициентом умножения. Наиболее распространенным умножителем на сегодняшний день является схема синтезатора частоты с фазовой автоподстройкой (PLL).

Подсистема синхронизации

От основного генератора синхросигнал выводится на ножку микросхемы и может использоваться для тактирования внешних схем. Для снижения частоты выходного сигнала в этой цепи может использоваться управляемый или фиксированный делитель.

Блок вспомогательного генератора обеспечивает тактирование части периферийных устройств, обычно таймеров-счетчиков (PICmicro, ATmega, Fujitsu MB90), а в некоторых режимах может принимать на себя функции основного генератора (синхронизацию ядра, периферии, схем «сброса»). Вспомогательный генератор обычно работает на частотах до 1 МГц. В случае использования его как базы часов реального времени – на частоте 32768 Гц.

По структуре вспомогательный генератор аналогичен основному генератору, но почти никогда не используется умножитель частоты.

Подсистема синхронизации

Ядром формирователя внутренних сигналов являются делители частоты основного и вспомогательного генераторов. С их выходов берутся сигналы тактирования ядра и схемы сброса ssync и сигналы синхронизации периферийных модулей psync. Обычно все внутренние сигналы (ssync и psync) получаются делением частоты генераторов на фиксированный коэффициент, но в некоторых процессорах коэффициенты деления могут программироваться, например, если необходимо снизить частоту тактирования ядра в режимах пониженного энергопотребления.

Подсистема синхронизации

Коммутаторы синхросигналов К1 и К2 используются для выбора источника тактирования внутренних схем процессора: основного или вспомогательного генератора. В обычном режиме большинство подсистем синхронизируется от основного генератора, а вспомогательный используется как временная база таймеров, часов реального времени или сторожевого таймера. В случае нестабильности работы основного генератора или при необходимости перейти на более низкие частоты функционирования, например, в режимах энергосбережения, можно подключить вход делителя основного генератора на выход вспомогательного генератора.

Механизмы начальной инициализации встроенной памяти

Механизмы начальной инициализации (начальной загрузки) обеспечивают запись программного кода, данных или конфигурационных параметров во встроенную энергонезависимую память процессора или однокристальной микроЭВМ. Процесс начальной инициализации предполагает работу с «голой» аппаратурой, т.е. без помощи какой-либо инструментальной программы (загрузчика), исполняющейся в рабочем режиме процессора.

В качестве записываемого программного кода выступает или более высокоуровневый загрузчик или непосредственно прикладная программа.

Данные – обычно начальные значения рабочих параметров (уставок).

Механизмы начальной инициализации встроенной памяти

Конфигурационные параметры настраивают режимы работы аппаратуры процессора. К ним могут относиться:

- тип генератора (кварцевый, на пьезокерамическом резонаторе, LC или RC);
- используемые подсистемы сброса при сбое электропитания, автоматического сброса при включении питания (Power On Reset);
- использование сторожевого таймера (Watch Dog Timer);
- флаги защиты внутренней памяти от несанкционированного копирования;
- использование и разрядность шины внешней памяти;
- адрес старта программы (вектор сброса);
- и т.д.

Классификация механизмов начальной инициализации представлена на рисунке ниже.

Механизмы инициализации встроенной памяти

Механизмы начальной инициализации встроенной памяти

Встроенная программа загрузчика (Bootsrap loader) – специальная программа, записанная при производстве процессора в специальный блок встроенной памяти программ ПЗУ. При выполнении bootstrap loader принимает записываемые программы или данные через последовательный порт (обычно порт UART) и записывает их в память процессора.

Механизмы начальной инициализации встроенной памяти

Возможны несколько вариантов сохранения загруженной программы:

1. Программа загружается в ОЗУ и сразу после этого ей передается управление. Это должен быть загрузчик, в свою очередь принимающий и записывающий во встроенное или внешнее ПЗУ (обычно это – FLASH-память) прикладную программу. После рестарта управление передается прикладной программе.

2. Программа записывается непосредственно во встроенное ПЗУ и начинает исполняться после перезапуска в нормальном режиме. В качестве загруженной программы может выступать загрузчик или целевой код. Такой режим используется, например, в семействах MB90F (Fujitsu), MSP430 (Texas Instruments).

Переход в режим bootstrap loader обычно выполняется подачей специального кода на конфигурационные выводы с одновременным рестартом процессора.

Внешнее программирование встроенного ПЗУ

В данном режиме процессор со встроенным ПЗУ (OTP, EEPROM, FLASH) рассматривается как обычная микросхема ПЗУ с последовательным или параллельным интерфейсом. Ядро процессора при этом отключено; некоторые семейства, например, AVR (Atmel), требуют, чтобы функционировал генератор синхроимпульсов. Переключение процессора в режим внешнего программирования обычно осуществляется подачей на специальный вывод напряжения программирования с уровнем около 12В или подачей специального кода на конфигурационные выводы с одновременным рестартом процессора (аналогично переходу в режим bootsrap loader).

Внешнее программирование встроенного ПЗУ

В зависимости от типа интерфейса различают два варианта внешнего программирования:

1. Программирование по параллельному интерфейсу (поддерживается семействами AVR, MCS-51, Z8). Процессор или микроЭВМ программируется вне целевой системы, в специальных программаторах.

Порты ввода-вывода и/или сигналы внешней шины используются в качестве линий адреса/данных/управления ПЗУ.

Внешнее программирование встроенного ПЗУ

Достоинства:

• Простые алгоритмы программирования;

• Высокая степень защиты от случайного перепрограммирования в системе;

Недостатки:

• Необходимо использовать панельку для микропроцессора или программировать однократно перед монтажом печатной платы системы. Последнее не позволяе модифицировать (обновлять) программное обеспечение системы.

Внешнее программирование встроенного ПЗУ

2. Программирование по последовательному интерфейсу (поддерживается семействами AVR, PICmicro). В этом режиме адреса, данные и команды доступа к ПЗУ (записи, чтения, проверки и другие) передаются по специальному или стандартному последовательному интерфейсу.

Внешнее программирование встроенного ПЗУ

Достоинства:

- Небольшое количество сигналов (2-5 шт.) позволяет подключать программатор к микросхеме, установленной на плате, и программировать ее, не отключая от схемы. В связи с этим режим последовательного программирования часто называют режимом внутрисистемного программирования (In System Programming, ISP).
- Неограниченность числа различных команд, которые можно передавать в последовательном коде, позволяет значительно увеличить функциональные возможности программатора.

Недостатки:

- Относительно сложный протокол (алгоритм) программирования.

Внешнее программирование встроенного ПЗУ

В настоящее время в режиме внутрисистемного программирования микропроцессорных систем с внешним или внутренним ПЗУ начинает широко использоваться последовательный интерфейс JTAG (IEEE-1049). JTAG – интерфейс граничного сканирования, позволяющий устанавливать на ножках микросхемы сигналы с определенным значением и считывать значения сигналов, установленные внешними схемами или внутренними подсистемами процессора. С помощью интерфейса JTAG имитируется диаграмма записи во внешнее ПЗУ или во внутреннее ПЗУ (процессор должен находиться в режиме внешнего параллельного программирования).

Сетевые интерфейсы встраиваемых систем

Лекция 10

Сетевые интерфейсы встраиваемых систем

Данный раздел посвящен обзору сетевых промышленных проводных и беспроводных, локальных и глобальных интерфейсов, которые применяются в современных встраиваемых системах для организации канала связи между компонентами самой ВВС или с другими информационными системами верхних уровней управления.

Последовательный интерфейс I^2C

В бытовой технике, телекоммуникационном оборудовании и промышленной электронике часто встречаются похожие решения, в, казалось бы, никак не связанных изделиях. Например, практически каждая система включает в себя:

- Некоторый «умный» узел управления, обычно однокристалльная микроЭВМ.
- Узлы общего назначения, такие как буферы ЖКИ, порты ввода-вывода, RAM, EEPROM или преобразователи данных.
- Специфические узлы, такие как схемы цифровой настройки и обработки сигнала для радио- и видео- систем, или генераторы тонального набора для телефонии.

Последовательный интерфейс I²C

Для того, чтобы использовать эти общие решения к выгоде конструкторов и производителей (технологов), а также для увеличения эффективности аппаратуры и упрощения схемотехнических решений, Philips в 1980 году разработала простую двунаправленную двухпроводную шину для эффективного «межмикросхемного» (inter-IC) управления. Шина так и называется – Inter-Integrated Circuit, или IIC (I²C) шина. В настоящее время ассортимент продукции Philips включает более 150 КМОП и биполярных I²C-совместимых устройств, функционально предназначенных для работы во всех трех вышеперечисленных категориях электронного оборудования. Все I²Ссовместимые устройства имеют встроенный интерфейс, который позволяет им связываться друг с другом по шине I²C. Это конструкторское решение разрешает множество проблем сопряжения различных устройств, которые обычно возникают при разработке цифровых систем.

Последовательный интерфейс I²C

Основной режим работы шины I²C – 100 кбит/с; 10 кбит/с в режиме работы с пониженной скоростью. Заметим, что стандарт допускает тактирование с частотой вплоть до нулевой. Для адресации I²C-устройств используется 7 бит.

После пересмотра стандарта в 1992 году становится возможным подключение ещё большего количества устройств на одну шину (за счёт возможности 10-битной адресации), а также большую скорость до 400 кбит/с в скоростном режиме. Соответственно, доступное количество свободных узлов выросло до 1008. Максимальное допустимое количество микросхем, подсоединенных к одной шине, ограничивается максимальной емкостью шины в 400 пФ.

Последовательный интерфейс I²C

Версия стандарта 2.0, выпущенная в 1998 году представила высокоскоростной режим работы со скоростью до 3.4 Мбит/с с пониженным энергопотреблением. Последняя версия 2.1 2000 года включила лишь незначительные доработки.

1 октября 2006 года были отменены лицензионные отчисления за использование протокола I²C. Однако, отчисления сохраняются для выделения эксклюзивного подчинённого адреса на шине I²C.

Последовательный интерфейс I^2C

Список возможных применений I^2C:

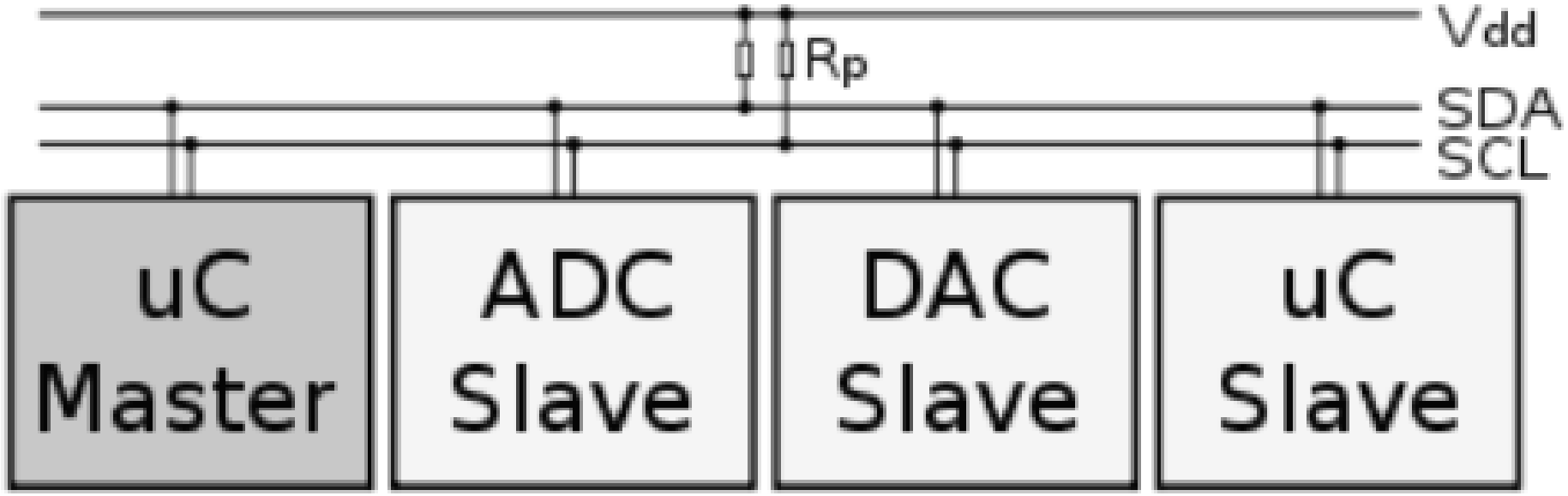
- доступ к модулям памяти (RAM, EEPROM, Flash и др.);
- доступ к низкоскоростным ЦАП/АЦП;
- работа с часами реального времени (RTC);
- регулировка контрастности, насыщенности и цветового баланса мониторов;
- управление интеллектуальными звукоизлучателем (динамиками);
- управление ЖКИ, в том числе в мобильных телефонах;
- чтение информации с датчиков мониторинга и диагностики оборудования, например, термостат центрального процессора или датчик скорости вращения вентилятора охлаждения процессора;
- информационный обмен между микроконтроллерами.

Концепция шины I²C

I²C использует две двунаправленных линии с открытым стоком – последовательная линия данных (SDA, англ. Serial DAta) и последовательная линия тактирования (SCL, англ. Serial CLock), обе нагруженные резисторами.

Максимальное напряжение +5В, часто используется +3,3В, однако допускаются и другие напряжения (не менее +2В). Шина I²C поддерживает любую технологию изготовления микросхем (НМОП, КМОП, биполярную).

Пример соединения устройств по шине I^2C: один ведущий – микроконтроллер, три ведомых устройства – АЦП, ЦАП, МК



Концепция шины I²C

Каждое устройство распознается по уникальному адресу – будь то микроконтроллер, ЖКИ-буфер, память или интерфейс клавиатуры – и может работать как передатчик или приёмник, в зависимости от назначения устройства. Обычно ЖКИ-буфер – только приёмник, а память может как принимать, так и передавать данные. Кроме того, устройства могут быть классифицированы как ведущие и ведомые при передаче данных. Ведущий – это устройство, которое инициирует передачу данных и вырабатывает сигналы синхронизации. При этом любое адресуемое устройство считается ведомым по отношению к ведущему. Классическая адресация включает 7-битное адресное пространство с 16 зарезервированными адресами. Это означает до 112 свободных адресов для подключения периферии на одну шину.

Термины, используемые в спецификации I²C

Термин (англ.)	Термин (рус.)	Описание
Transmitter	Передатчик	Устройство, посылающее данные в шину
Receiver	Приемник	Устройство, принимающее с шины
Master	Ведущий	Начинает пересылку данных, вырабатывает синхроимпульсы, заканчивает пересылку данных
Slave	Ведомый	Устройство, адресуемое ведущим
Multi-master	–	Несколько ведущих могут пытаться захватить шину одновременно, без нарушения передаваемой информации
Arbitration	Арбитраж	Процедура, обеспечивающая Multi-master
Synchronization	Синхр.	Процедура синхронизации двух устройств

Концепция шины I^2C

Возможность подключения более одного микроконтроллера к шине означает, что более чем один ведущий может попытаться начать пересылку в один и тот же момент времени. Для устранения хаоса, который может возникнуть в данном случае, разработана процедура арбитража. Эта процедура основана на том, что все I^2C-устройства подключаются к шине по правилу монтажного И.

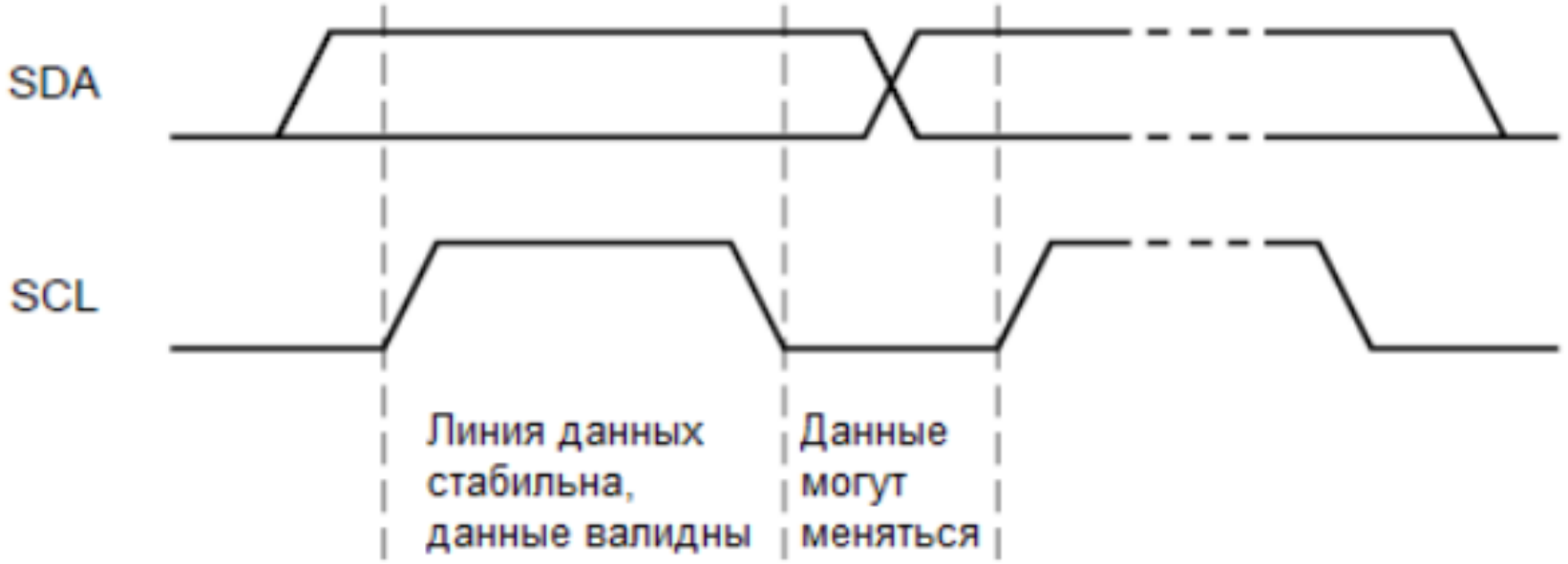
Генерация синхросигнала – это всегда обязанность ведущего; каждый ведущий генерирует свой собственный сигнал синхронизации при пересылке данных по шине. Сигнал синхронизации может быть изменен, только если он “вытягивается” медленным ведомым устройством (путем удержания линии в низком состоянии), или другим ведущим, если происходит столкновение.

Принцип работы шины I^2C

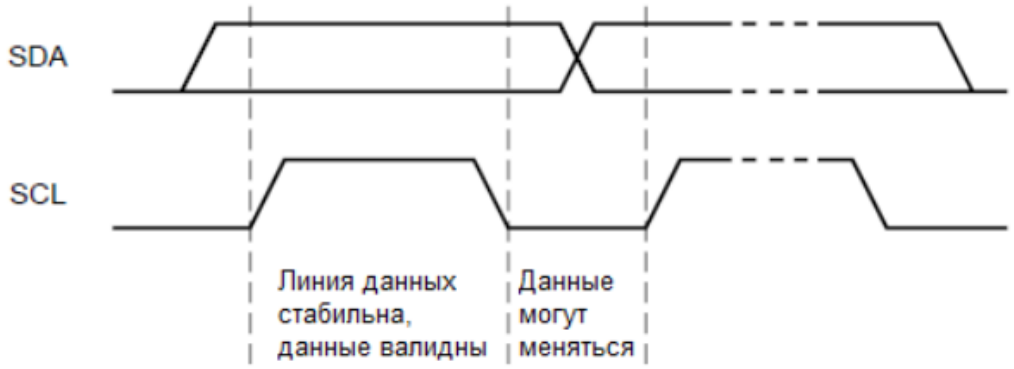
Вследствие различных технологий микросхем (КМОП, НМОП, биполярная), которые могут быть подключены к шине, уровни логического нуля (НИЗКИЙ) и логической единицы (ВЫСОКИЙ) не фиксированы и зависят от соответствующего уровня Vdd. Один синхроимпульс генерируется на каждый пересылаемый бит.

Данные на линии SDA должны быть стабильными в течение ВЫСОКОГО периода синхроимпульса. ВЫСОКОЕ или НИЗКОЕ состояние линии данных должно меняться, только если линия синхронизации в состоянии НИЗКОЕ.

Пересылка бита по шине I²C

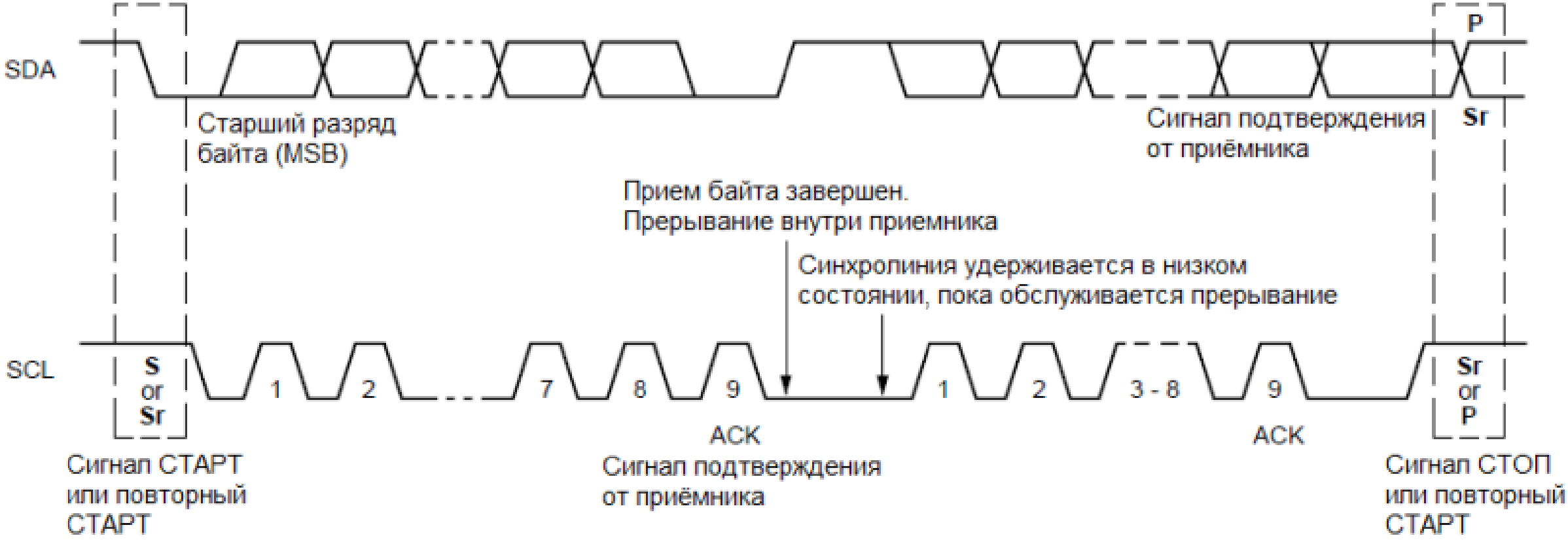


Принцип работы шины I²C



Данные по линии SDA передаются байтами, при этом каждый байт должен оканчиваться битом подтверждения. Количество байт, передаваемых за один сеанс связи, не ограничено. Данные передаются, начиная со старшего бита. Если приёмник не может принять еще один целый байт, пока он не выполнит какую-либо другую функцию (например, обслужит внутреннее прерывание), он может удерживать линию SCL в НИЗКОМ состоянии, переводя передатчик в состояние ожидания. Пересылка данных продолжается, когда приёмник будет готов к следующему байту и отпустит линию SCL (опять срабатывает правило монтажного И).

Пересылка данных по шине I²C

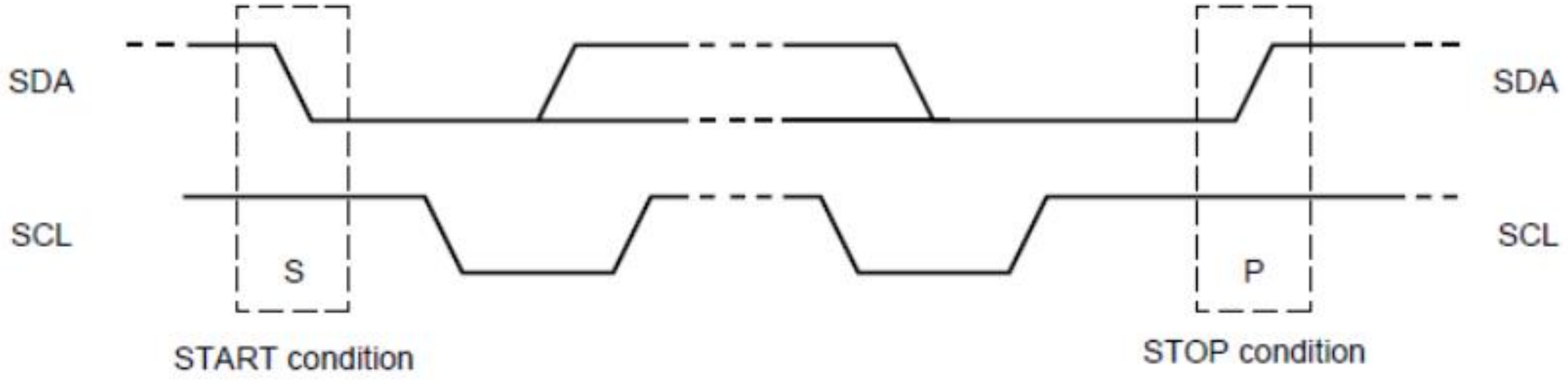


Сигналы СТАРТ и СТОП

Процедура обмена данными по шине I^2C начинается с того, что ведущий формирует состояние СТАРТ – ведущий генерирует переход сигнала линии SDA из ВЫСОКОГО состояния в НИЗКОЕ при ВЫСОКОМ уровне на линии SCL. Этот переход воспринимается всеми устройствами, подключенными к шине как признак начала процедуры обмена. Процедура обмена завершается тем, что ведущий формирует состояние СТОП – переход состояния линии SDA из НИЗКОГО состояния в ВЫСОКОЕ при ВЫСОКОМ состоянии линии SCL. Состояния СТАРТ и СТОП всегда вырабатываются ведущим. Считается, что шина занята после фиксации состояния СТАРТ. Шина считается освободившейся через некоторое время после фиксации состояния СТОП.

Определение сигналов СТАРТ и СТОП устройствами, подключенными к шине достаточно легко, если в них встроены необходимые цепи. Однако микроконтроллеры без таковых цепей должны осуществлять считывание значения линии SDA как минимум дважды за период синхронизации для того, чтобы определить переход состояния.

СТАРТ и СТОП состояния



Подтверждение

Подтверждение при передаче данных обязательно, кроме случаев окончания передачи ведомой стороной. Соответствующий импульс синхронизации генерируется ведущим. Передатчик отпускает (ВЫСОКОЕ) линию SDA в течение синхроимпульса подтверждения. Приёмник должен удерживать линию SDA в течение ВЫСОКОГО состояния синхроимпульса подтверждения в стабильно НИЗКОМ состоянии. Конечно, время установки и удержания также должны быть приняты во внимание (электрические и временные параметры).

Таким образом передача 8 бит данных от передатчика к приемнику завершаются дополнительным циклом (формированием 9-го тактового импульса линии SCL), при котором приемник выставляет НИЗКИЙ уровень сигнала на линии SDA, как признак успешного приема байта.

Подтверждение

В том случае, когда ведомый-приёмник не может подтвердить свой адрес (например, когда он выполняет в данный момент какие-либо функции реального времени), линия данных должна быть оставлена в ВЫСОКОМ состоянии. После этого ведущий может выдать сигнал СТОП для прерывания пересылки данных. Если в пересылке участвует ведущий-приёмник, то он должен сообщить об окончании передачи ведомому-передатчику путем не подтверждения последнего байта. Ведомый-передатчик должен освободить линию данных для того, чтобы позволить ведущему выдать сигнал СТОП или повторить сигнал СТАРТ.

Синхронизация

При передаче посылок по шине I^2C каждый ведущий генерирует свой синхросигнал на линии SCL. Данные действительны только во время ВЫСОКОГО состояния синхроимпульса.

Синхронизация выполняется с использованием подключения к линии SCL по правилу монтажного И. Это означает, что ведущий не имеет монопольного права на управление переходом линии SCL из НИЗКОГО состояния в ВЫСОКОЕ. В том случае, когда ведомому необходимо дополнительное время на обработку принятого бита, он имеет возможность удерживать линию SCL в низком состоянии до момента готовности к приему следующего бита. Таким образом, линия SCL будет находиться в НИЗКОМ состоянии на протяжении самого длинного НИЗКОГО периода синхросигналов.

Синхронизация

Устройства с более коротким НИЗКИМ периодом будут входить в состояние ожидания на время, пока не кончится длинный период. Когда у всех задействованных устройств кончится НИЗКИЙ период синхросигнала, линия SCL перейдет в ВЫСОКОЕ состояние. Все устройства начнут проходить ВЫСОКИЙ период своих синхросигналов. Первое устройство, у которого кончится этот период, снова установит линию SCL в НИЗКОЕ состояние. Таким образом, НИЗКИЙ период синхролинии SCL определяется наидлиннейшим периодом синхронизации из всех задействованных устройств, а ВЫСОКИЙ период определяется самым коротким периодом синхронизации устройств.

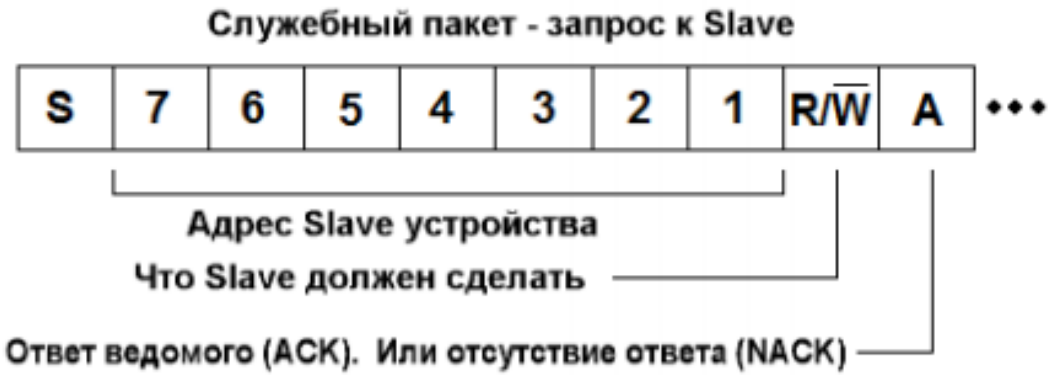
Механизм синхронизации может быть использован приемниками как средство управления пересылкой данных на байтовом и битовом уровнях.

Синхронизация

На уровне байта, если устройство может принимать байты данных с большой скоростью, но требует определенное время для сохранения принятого байта или подготовки к приему следующего, то оно может удерживать линию SCL в НИЗКОМ состоянии после приема и подтверждения байта, переводя таким образом передатчик в состояние ожидания.

На уровне битов, устройство, такое как микроконтроллер без встроенных аппаратных цепей I^2C или с ограниченными цепями, может замедлить частоту синхроимпульсов путем продления их НИЗКОГО периода. Таким образом скорость передачи любого ведущего адаптируется к скорости медленного устройства.

Форматы обмена данными по шине I²C (7-битный адрес)



После сигнала СТАРТ посылается адрес ведомого. После 7 бит адреса следует бит направления данных (R/W), «ноль» означает передачу (запись), а «единица» – прием (чтение). Пересылка данных всегда заканчивается сигналом СТОП, генерируемым ведущим. Однако, если ведущий желает оставаться на шине дальше, он должен выдать повторный сигнал СТАРТ и затем адрес следующего устройства.

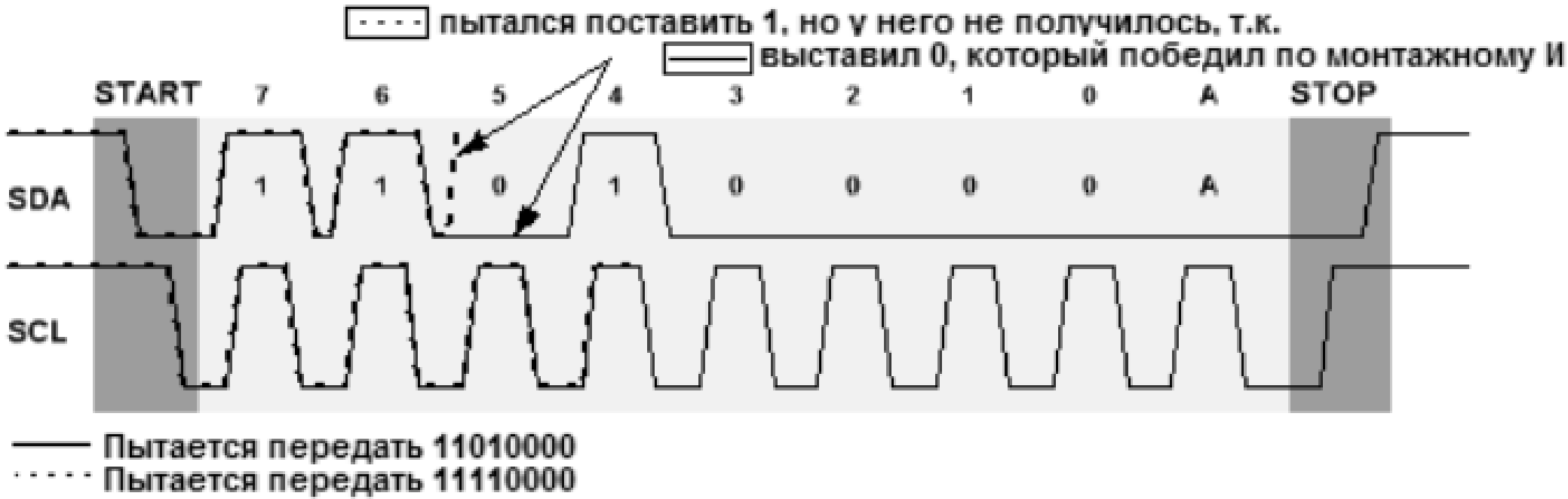
При таком формате посылки возможны различные комбинации чтения/записи.

Арбитраж

Арбитраж помогает решать конфликтные ситуации во время передачи данных по I^2C, когда присутствует несколько ведущих (режим мультимастера). Ведущий может начинать пересылку данных только, если шина свободна. Если один ведущий передает на линию данных НИЗКИЙ уровень, в то время как другой – ВЫСОКИЙ, то последний отключается от линии, так как состояние SDA (НИЗКОЕ) не соответствует ВЫСОКОМУ состоянию его внутренней линии данных.

Вследствие того, что арбитраж зависит только от адреса и данных, передаваемых соревнующимися ведущими, не существует центрального ведущего, а также приоритетного доступа к шине.

Арбитраж между двумя ведущими (случай одновременной передачи данных)



Арбитраж

Что же будет, когда два ведущих начнут передачу одновременно? Тут опять помогает свойство монтажного И: оба мастера бит за битом передают адрес ведомого, потом данные, кто первый выставит на линию «0», тот и побеждает в этой конфликтной ситуации. Так что очевидно, что самый важный адрес должен начинаться с нулей, чтобы тот, кто к нему пытался обращаться, всегда выигрывал арбитраж. Проигравшая же сторона вынуждена ждать, пока шина не освободится.

Таким образом, арбитраж может продолжаться до окончания адреса, а если ведущие адресуют одно и то же устройство, то в арбитраже будут участвовать и данные. Вследствие такой схемы арбитража при столкновении данные не теряются.

Арбитраж

Ведущему, проигравшему арбитраж, разрешается выдавать синхроимпульсы на шину SCL до конца байта, в течение которого был потерян доступ.

Если в устройство ведущего также встроены и функции ведомого и он проигрывает арбитраж на стадии передачи адреса, то он немедленно должен переключиться в режим ведомого, так как выигравший арбитраж ведущий мог адресовать его.

Достоинства шины I²C

• Требуется только две линии – линия данных (SDA) и линия синхронизации (SCL) Каждое устройство, подключённое к шине, может быть программно адресовано по уникальному адресу. В каждый момент времени существует простое отношение ведущий/ведомый: ведущие могут работать как ведущий-передатчик и ведущий-приёмник.

• Шина позволяет иметь несколько ведущих, предоставляя средства для определения коллизий и арбитраж для предотвращения повреждения данных в ситуации, когда два или более ведущих одновременно начинают передачу данных. В стандартном режиме обеспечивается передача последовательных 8-битных данных со скоростью до 100 кбит/с, и до 400 кбит/с в “быстром” режиме.

• Встроенный в микросхемы фильтр подавляет всплески, обеспечивая целостность данных.

• Максимальное допустимое количество микросхем, подсоединённых к одной шине, ограничивается максимальной емкостью шины 400 пФ.

Достоинства шины I²C

Это лишь некоторые преимущества. Кроме того, I²C-совместимые микросхемы увеличивают гибкость системы, позволяя простое конструирование вариантов оборудования и легкую модернизацию для того, чтобы поддерживать разработки на современном уровне. Таким образом, целое семейство оборудования может быть разработано, основываясь на базовой модели. Модернизация оборудования или расширение его функций (например, дополнительная память, дистанционное управление и т.п.) может быть произведена путем простого подключения соответствующей микросхемы к шине. Если требуется большая ПЗУ, то дело лишь в выборе микроконтроллера с большим объемом ПЗУ. Поскольку новые микросхемы могут замещать старые, легко добавлять новые свойства в оборудование или увеличивать его производительность путем простого отсоединения устаревшей микросхемы и подключения к шине новой.

Интерфейс RS-485

RS-485 (Recommended Standard 485, Electronics Industries Association 485, EIA-485) – стандарт передачи данных по двухпроводному полудуплексному многоточечному последовательному каналу связи.

Стандарт RS-485 совместно разработан двумя ассоциациями: Ассоциацией электронной промышленности (EIA — Electronics Industries Association) и Ассоциацией промышленности средств связи (TIA — Telecommunications Industry Association). Ранее EIA маркировала все свои стандарты префиксом «RS» (Recommended Standard — Рекомендованный стандарт). Многие инженеры продолжают использовать это обозначение, однако EIA/TIA официально заменил «RS» на «EIA/TIA» с целью облегчить идентификацию происхождения своих стандартов. На сегодняшний день, различные расширения стандарта RS-485 охватывают широкое разнообразие приложений, этот стандарт стал основой для создания целого семейства промышленных сетей широко используемых в промышленной автоматизации.

Интерфейс RS-485

В стандарте RS-485 для передачи и приёма данных часто используется единственная витая пара проводов. Передача данных осуществляется с помощью дифференциальных сигналов. Разница напряжений между проводниками одной полярности означает логическую единицу, разница другой полярности — ноль.

Интерфейс RS-485

RS-485 имеет следующие особенности:

- возможность объединения несимметричных и симметричных цепей,
- параметры качества сигнала, уровень искажений (%),
- методы доступа к линии связи,
- протокол обмена,
- аппаратную конфигурацию (среда обмена, кабель),
- типы соединителей, разъёмов, колодок, нумерацию контактов,
- качество источника питания (стабилизация, пульсация, допуск),
- отражения в длинных линиях.

Интерфейс RS-485

Электрические и временные характеристики интерфейса RS-485:

• 32 приёмопередатчика при многоточечной конфигурации сети (на одном сегменте, максимальная длина линии в пределах одного сегмента сети: 1200 метров).

• Только один передатчик активный.

• Максимальное количество узлов в сети — 250 с учётом магистральных усилителей.

Интерфейс RS-485

Характеристика скорость обмена/длина линии связи (зависимость экспоненциальная):

- 62,5 кбит/с 1200 м (одна витая пара)
- 375 кбит/с 300 м (одна витая пара)
- 500 кбит/с
- 1000 кбит/с
- 2400 кбит/с 100 м (две витых пары)
- 10000 кбит/с 10 м

Примечание: Скорости обмена 62,5 кбит/с, 375 кбит/с, 2400 кбит/с оговорены стандартом RS-485. На скоростях обмена свыше 500 кбит/с рекомендуется использовать экранированные витые пары.

Тип приёмопередатчиков — дифференциальный, потенциальный. Изменение входных и выходных напряжений на линиях A и B: Ua (Ub) от −7В до +12В (+7В).

Согласование и конфигурация линии связи

При больших расстояниях между устройствами, связанными по витой паре и высоких скоростях передачи начинают проявляться так называемые эффекты длинных линий. Причина этому – конечность скорости распространения электромагнитных волн в проводниках. Скорость эта существенно меньше скорости света в вакууме и составляет немногим больше 200 мм/нс. Электрический сигнал имеет также свойство отражаться от открытых концов линии передачи и ее ответвлений. Грубая аналогия - желоб, наполненный водой. Волна, созданная в одном конце, идет по желобу и, отразившись от стенки в конце, идет обратно, отражается опять и так далее, пока не затухнет. Для коротких линий и малых скоростей передачи этот процесс происходит так быстро, что остается незамеченным. Однако, время реакции приемников - десятки/сотни нс. В таком масштабе времени несколько десятков метров электрический сигнал проходит отнюдь не мгновенно. И если расстояние достаточно большое, фронт сигнала, отразившийся в конце линии и вернувшийся обратно, может исказить текущий или следующий сигнал. В таких случаях нужно каким-то образом подавлять эффект отражения.

Согласование и конфигурация линии связи

У любой линии связи есть такой параметр, как волновое сопротивление Zв. Оно зависит от характеристик используемого кабеля, но не от длины. Для обычно применяемых в линиях связи витых пар Zв=120 Ом. Оказывается, что если на удаленном конце линии, между проводниками витой пары включить резистор с номиналом равным волновому сопротивлению линии, то электромагнитная волна дошедшая до "тупика" поглощается на таком резисторе. Отсюда его названия – согласующий резистор или "терминатор".

Большой минус согласования на резисторах – повышенное потребление тока от передатчика, ведь в линию включается низкоомная нагрузка. Поэтому рекомендуется включать передатчик только на время отправки посылки. Есть способы уменьшить потребление тока, включая последовательно с согласующим резистором конденсатор для развязки по постоянному току. Однако, такой способ имеет свои недостатки. Для коротких линий (несколько десятков метров) и низких скоростей (меньше 38400 бод) согласование можно вообще не делать.

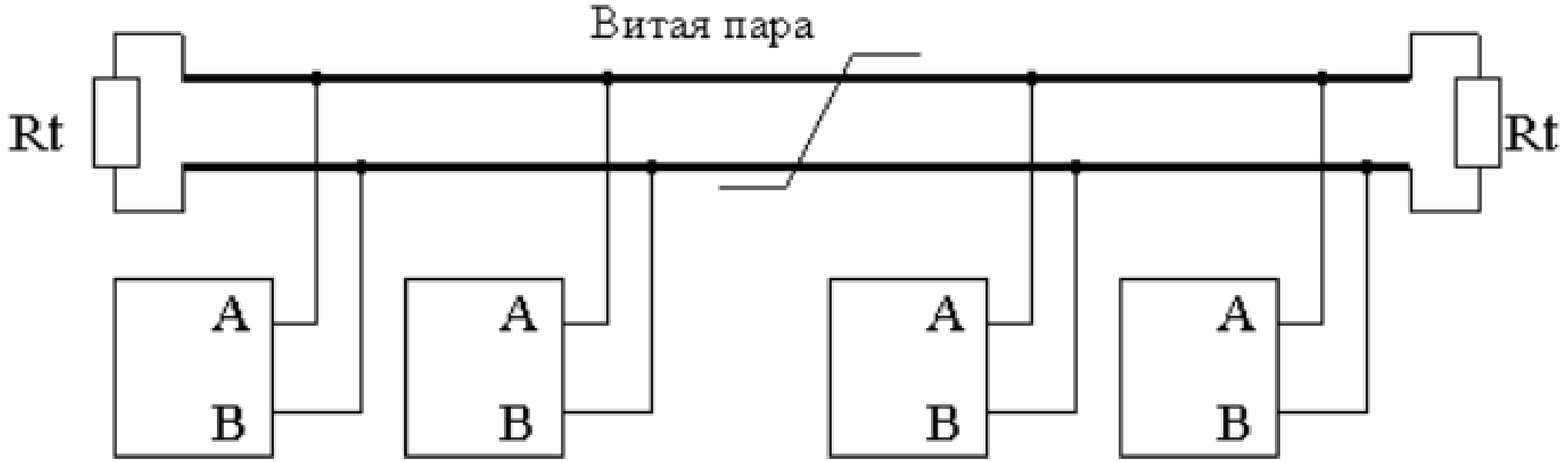
Согласование и конфигурация линии связи

Эффект отражения и необходимость правильного согласования накладывают ограничения на конфигурацию линии связи.

Линия связи должна представлять собой один кабель витой пары. К этому кабелю присоединяются все приемники и передатчики. Расстояние от линии до микросхем интерфейса RS-485 должно быть как можно короче, так как длинные ответвления вносят рассогласование и вызывают отражения.

В оба наиболее удаленных конца кабеля ($Z_в$=120 Ом) включают согласующие резисторы Rt по 120 Ом (0.25 Вт). Если в системе только один передатчик и он находится в конце линии, то достаточно одного согласующего резистора на противоположном конце линии.

Согласование линии связи



Защитное смещение

Как уже упоминалось, приемники большинства микросхем RS-485 имеют пороговый диапазон распознавания сигнала на входах A-B - ±200мВ. Если |Uab| меньше порогового (около 0), то на выходе приемника RO могут быть произвольные логические уровни из-за несинфазной помехи. Такое может случиться либо при отсоединении приемника от линии, либо при отсутствии в линии активных передатчиков, когда никто не задает уровень. Чтобы в этих ситуациях избежать выдачи ошибочных сигналов на приемник UART, необходимо на входах A-B гарантировать разность потенциалов Uab > +200мВ. Это смещение при отсутствии входных сигналов обеспечивает на выходе приемника логическую "1", поддерживая, таким образом, уровень стопового бита.

Добиться этого просто - прямой вход (A) следует подтянуть к питанию, а инверсный (B) - к "земле".

Защитное смещение

Величины сопротивлений для резисторов защитного смещения (Rзс) нетрудно рассчитать по делителю. Необходимо обеспечить Uab > 200мВ. Напряжение питания – 5. Сопротивление среднего плеча – 120Ом//120Ом//12КОм на каждый приемник – примерно 57 Ом (для 10 приемников). Таким образом, выходит примерно по 650 Ом на каждый из двух Rзс. Для смещения с запасом - сопротивление Rзс должно быть меньше 650 Ом. Традиционно ставят 560 Ом.

Обратите внимание: в расчете номинала Rзс учитывается нагрузка. Если на линии висит много приемников, то номинал Rзс дожен быть меньше. В длинных линиях передачи необходимо так же учитывать сопротивление витой пары, которое может "съедать" часть смещающей разности потенциалов для удаленных от места подтяжки устройств. Для длинной линии лучше ставить два комплекта подтягивающих резисторов в оба удаленных конца рядом с терминаторами.

Функция безотказности

Многие производители приемопередатчиков заявляют о функции безотказности (failsafe) своих изделий, заключающейся во встроенном смещении. Следует различать два вида такой защиты:

• Безотказности в открытых цепях (Open circuit failsafe). В таких приемопередатчиках применяются встроенные подтягивающие резисторы. Эти резисторы, как правило, высокоомные, чтобы уменьшить потребление тока. Из-за этого необходимое смещение обеспечивается только для открытых (ненагруженных) дифференциальных входов. В самом деле, если приемник отключен от линии или она не нагружена, тогда в среднем плече делителя остается только большое входное сопротивление, на котором и падает необходимая разность потенциалов. Однако, если приемопередатчик нагрузить на линию с двумя согласующими резисторами по 120 Ом, то в среднем плече делителя оказывается меньше 60 Ом, на которых, по сравнению с высокоомными подтяжками, ничего существенного не падает. Поэтому, если в нагруженной линии нет активных передатчиков, то встроенные резисторы не обеспечивают достаточное смещение. В этом случае, остается необходимость устанавливать внешние резисторы защитного смещения, как это было описано выше.

Функция безотказности

• Истинная безотказность (True failsafe). В этих устройствах смещены сами пороги распознавания сигнала. Например: -50 / -200 мВ вместо стандартных порогов ±200 мВ. То есть при Uab>-50мВ на выходе приемника RO будет логическая "1", а при Uab<-200 - на RO будет "0". Таким образом, и в разомкнутой и в пассивной линии при разности потенциалов Uab близкой к нулю, приемник выдаст "1". Для таких приемопередатчиков внешнее защитное смещение не требуется. Тем не менее, для лучшей помехозащищенности все-таки стоит дополнительно немного подтягивать линию.

Функция безотказности

Сразу виден минус внешнего защитного смещения – через делитель постоянно будет протекать ток, что может быть недопустимо в системах малого потребления. В таком случае можно сделать следующее:

1. Уменьшить потребление тока, увеличив сопротивления Rзс. Хотя производители приемопередатчиков и пишут о пороге распознавания в 200мВ, на практике вполне хватает 100мВ и даже меньше. Таким образом, можно сразу увеличить сопротивления Rзс раза в два-три. Помехозащищенность при этом несколько снижается, но во многих случаях это не критично.

2. Использовать true failsafe приемопередатчики со смещенными порогами распознавания. Например, у микросхем MAX3080 и MAX3471 пороги: - 50мВ / -200мВ, что гарантирует единичный уровень на выходе приемника при отсутствии смещения (Uab=0). Тогда внешние резисторы защитного смещения можно убрать или значительно увеличить их сопротивление.

3. Не применять без необходимости согласование на резисторах. Если линия не будет нагружена на 2 по 120 Ом, то для обеспечения защитного смещения хватит подтяжек в несколько килоом в зависимости от числа приемников на линии.

Функция безотказности

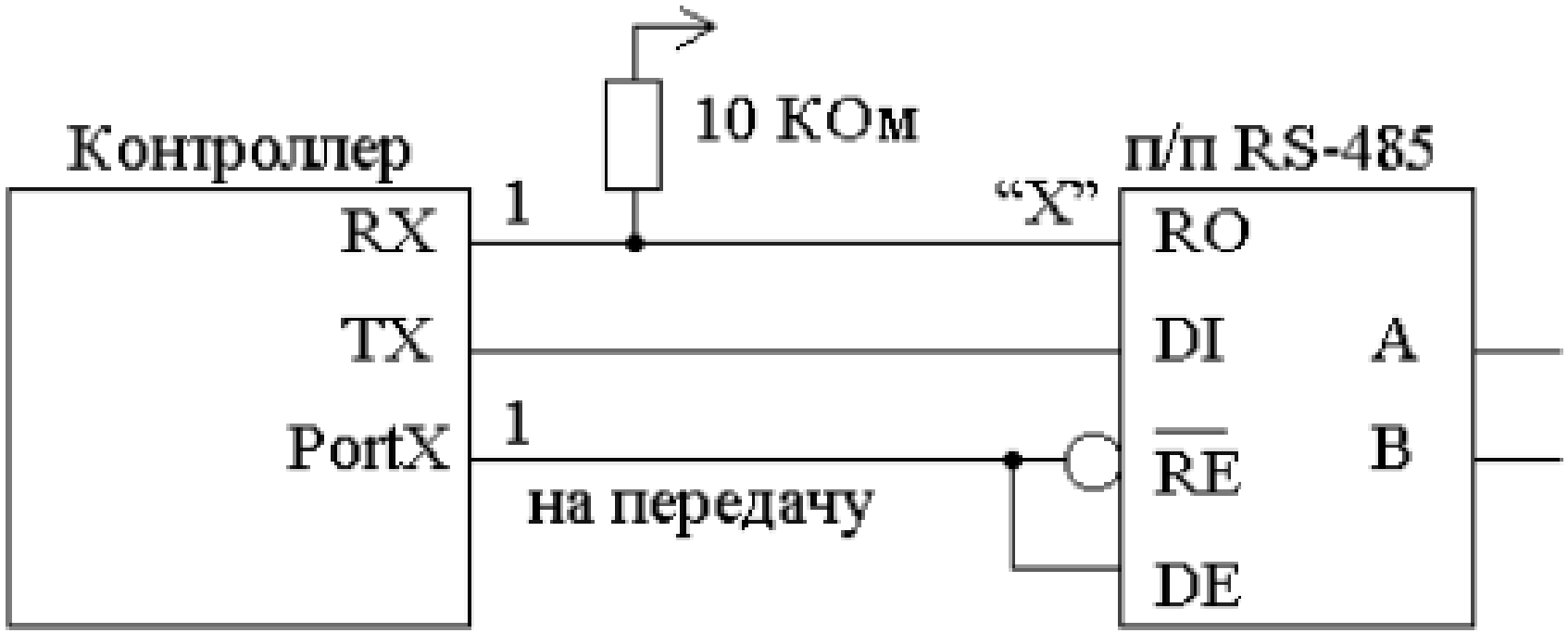
Для опторазвязанной линии подтягивать следует к питанию и "земле« изолированной линии. Если не применяется опторазвязка, подтягивать можно к любому питанию, так как делитель создаст лишь небольшую разность потенциалов между линиями А и В. Нужно только помнить о возможной разности потенциалов между "землями" устройств, расположенных далеко друг от друга.

Исключение приема при передаче в полудуплексном режиме

При работе с полудуплексным интерфейсом RS-485 (прием и передача по одной паре проводов с разделением по времени) можно забыть, что UART контроллера полнодуплексный, то есть принимает и передает независимо и одновременно.

Обычно во время работы приемопередатчика RS-485 на передачу, выход приемника RO переводится в третье состояние и ножка RX контроллера (приемник UART) "повисает в воздухе". В результате, во время передачи на приемнике UART вместо уровня стопового бита ("1") окажется неизвестно что, и любая помеха будет принята за входной сигнал. Поэтому нужно либо на время передачи отключать приемник UART (через управляющий регистр), либо подтягивать RX к единице. У некоторых микроконтроллеров это можно сделать программно – активировать встроенные подтяжки портов.

Схема подключения приемопередатчика RS-485 к микроконтроллеру



Сетевые интерфейсы встраиваемых систем. Продолжение

Лекция 11

Интерфейс CAN

CAN (Controller Area Network) – последовательный протокол связи, который эффективно поддерживает распределенное управление в реальном масштабе времени с высоким уровнем безопасности. Разработан фирмой BOSCH. Режим передачи – последовательный, широковещательный, пакетный. Стандарт не описывает физический уровень, но чаще всего используется сеть с топологией шина на базе дифференциальной пары, стандарта ISO 11898. Передача ведётся кадрами, которые принимаются всеми узлами сети. Метод доступа к среде передачи данных с разрешением конфликтов приоритетно обеспечивает доступ на передачу сообщения. Полезная информация в кадре состоит из идентификатора длиной 11 бит (стандартный формат) или 29 бит (расширенный формат) и поля данных длиной от 0 до 8 байт. Идентификатор говорит о содержимом пакета и служит для определения приоритета при попытке одновременной передачи несколькими узлами.

Интерфейс CAN

Область применения – от высокоскоростных сетей до дешевых мультиплексных шин. В автоматике, устройствах управления, датчиках используется CAN со скоростью до 1 Mbit/s.

Для достижения прозрачности проекта и гибкости реализации, CAN был подразделен на различные уровни согласно модели ISO/OSI:

- Уровень передачи данных (Data Link Layer)
- Подуровень логического управления линией (LLC)
- Подуровень управления доступом к среде передачи (MAC)
- Физический Уровень (Physical Layer)

Интерфейс CAN

Область LLC подуровня:

• обеспечение сервиса для передачи данных и для удалённого запроса данных.

• решение, какие сообщения, полученные LLC подуровнем, должны быть фактически приняты.

• обеспечение средствами для управления восстановлением и уведомления о перегрузке.

Интерфейс CAN

Область MAC подуровня главным образом - протокол передачи, то есть: арбитраж, проверка на ошибки, сигнализация и типизация ошибок. Внутри MAC подуровня решается, является ли шина свободной для начала новой передачи или возможен только приём данных. В MAC подуровень также включены некоторые элементы битовой синхронизации. Всё это находится внутри MAC подуровня и не имеет никакой возможности к модификации.
Область физического уровня - фактическая передача битов между различными узлами с соблюдением всех электрических правил.
Внутри одной сети, физический уровень одинаков для всех узлов. Однако существует свобода в выборе физического уровня.

Интерфейс CAN

Основные характеристики:

• приоритетность сообщений;

• гарантированное время отклика;

• гибкость конфигурации;

• групповой прием с синхронизацией времени;

• система непротиворечивости данных;

• multimaster;

• обнаружение ошибок и их сигнализация;

• автоматическая ретрансляция испорченных сообщений, как только шина снова станет свободной;

• различие между нерегулярными ошибками и постоянными отказами узлов и автономное выключения дефектных узлов.

• Послойная архитектура CAN-сети согласно модели OSI.

• Физический уровень определяет, как сигналы фактически передаются и, следовательно, имеет дело с описанием битовой синхронизации и кодирования битов. Внутри этой спецификации характеристики передатчика / приемника физического уровня не определены, чтобы позволить среде передачи и реализации уровня сигнала быть оптимизированными для конкретных систем.

• MAC подуровень представляет собой ядро протокола CAN. Он передает сообщения, полученные от LLC подуровня, и принимает сообщения, которые будут переданы к LLC подуровню. MAC подуровень ответственен за арбитраж, подтверждение, обнаружение ошибок и их сигнализацию.

• LLC подуровень имеет отношение к фильтрации сообщений, уведомлению о перегрузке и управлению восстановлением.

Промышленный Ethernet

Industrial Ethernet (промышленный Ethernet) – стандартизованный (IEEE 802.3 и 802.11) вариант Ethernet для применения в промышленности. Сеть с процедурой доступа CSMA/CD. Industrial Ethernet обычно используется для обмена данными между программируемыми контроллерами и системами человеко-машинного интерфейса, реже для обмена данных между контроллерами и, незначительно, для подключения к контроллерам удаленного оборудования (датчиков и исполнительных устройств). Широкому применению Ethernet в последних задачах препятствует суть метода CSMA/CD, делающая невозможным гарантию обмена небольшим количеством информации (единицы байт) с высокой частотой (миллисекундные циклы обмена).

В последнее время является одной из самых распространённых промышленных сетей. Широко применяется при автоматизации зданий и в областях, не требующих высокой надёжности.

Протоколы реального времени

Для обеспечения гарантированного времени реакции используют протоколы реального времени:

- Profinet
- EtherCAT
- Ethernet Powerlink
- Ether/IP

Протоколы реального времени

Эти протоколы в различной степени модифицируют стандартный стек TCP/IP, добавляя в него:

- функции синхронизации
- новые алгоритмы сетевого обмена
- диагностические функции
- методы самокорректировки

Канальный и физический уровни Ethernet при этом остаются неизменными. Что позволяет использовать протоколы реального времени в существующих сетях Ethernet с использованием стандартного сетевого оборудования.

Резервирование каналов и кольцевая топология

Для обеспечения защиты каналов связи от единичного отказа необходимо их резервировать. Резервирование неизбежно ведет к возникновению кольцевых участков сети – замкнутых маршрутов. Стандарт Ethernet, предусматривает только древовидную топологию и не допускает кольцевых, так как это приводит к зацикливанию пакетов.

Резервирование каналов и кольцевая топология

Современные коммутаторы, как правило, поддерживают дополнительный прокол Spanning Tree Protocol (STP, IEEE 802.1d), который позволяет создавать кольцевые маршруты в сетях Ethernet. Постоянно анализируя конфигурацию сети, STP автоматически выстраивает древовидную топологию, переводя избыточные коммуникационные линии в резерв. В случае нарушения целостности построенной таким образом сети (обрыв связи, например), STP в считанные секунды включает в работу необходимые резервные линии, восстанавливая древовидную структуры сети. Этот протокол не требует первичной настройки и работает автоматически.

Резервирование каналов и кольцевая топология

Более мощная разновидность данного протокола - Rapid Spanning Tree Protocol (RSTP, IEEE 802.1w), позволяющая снизить время перестройки сети до нескольких миллисекунд. Протоколы STP и RSTP позволяют создавать произвольное количество избыточных линий связи и являются обязательным функционалом для промышленных коммутаторов, применяемых в резервированных сетях.

Отличия от обычного Ethernet

- Стандарты на кабели и разъемы, удовлетворяющие специфическим требованиям промышленности: усиленное экранирование, стойкость к агрессивным средам и т. п.
- Специальные стандарты и устройства для связи с подвижными объектами: гибкие кабели, устройства беспроводной связи
- Дополнение стека протоколов TCP/IP протоколом RFC 1006 обеспечивает регулярную и частую передачу по сети небольших объемов информации, что характерно для обмена данными между промышленными контроллерами
- С помощью специальных коммутаторов можно организовать кольцевую топологию, которая при обрыве восстанавливает связь, то есть находит новый путь для передачи данных значительно быстрее, чем применяемый в обычных сетях "алгоритм избыточного дерева"
- Частое использование наряду со стеком протоколов TCP/IP Специфического стека протоколов ISO Transport Protocol

Интерфейс LIN

LIN (Local Interconnect Network) – стандарт промышленной сети, разработаны консорциумом европейских автопроизводителей и других известных компаний, включая Audi AG, BMW AG, Daimler Chrysler AG, Motorola Inc., Volcano Communications Technologies AB, Volkswagen AG и VolvoCar Corporation. Протокол LIN предназначен для создания дешёвых локальных сетей обмена данными на коротких расстояниях. Он служит для передачи входных воздействий, состояний переключателей на панелях управления и так далее, а также ответных действий различных устройств, соединённых в одну систему через LIN, происходящих в так называемом «человеческом» временном диапазоне (порядка сотен миллисекунд).

Основные задачи, возлагаемые на LIN консорциумом европейских автомобильных производителей - объединение автомобильных подсистем и узлов (таких как дверные замки, стеклоочистители, стеклоподъёмники, управление магнитолой и климат-контролем, электролюк и так далее) в единую электронную систему. LIN-протокол утверждён Европейским Автомобильным Консорциумом как дешёвое дополнение к сверхнадёжному протоколу CAN.

Интерфейс LIN

LIN и CAN дополняют друг друга и позволяют объединить все электронные автомобильные приборы в единую многофункциональную бортовую сеть. Причём область применения CAN - участки, где требуется сверхнадёжность и скорость; область же применения LIN – объединение дешёвых узлов, работающих с малыми скоростями передачи информации на коротких дистанциях и сохраняющих при этом универсальность, многофункциональность, а также простоту разработки и отладки. Стандарт LIN включает технические требования на протокол и на среду передачи данных. Как последовательный протокол связи, LIN эффективно поддерживает управление электронными узлами в автомобильных системах с шиной класса «А» (двунаправленный полудуплексный), что подразумевает наличие в системе одного главного (master) и нескольких подчинённых (slave) узлов.

Технология PLC

PLC (Power Line Communication/Carrier) – относительно новая телекоммуникационная технология категории «последняя миля». Так называемый «Интернет из розетки», базируется на использовании внутридомовых и внутриквартирных электросетей для высокоскоростного информационного обмена. В этой технологии, основанной на частотном разделении сигнала, высокоскоростной поток данных разбивается на несколько низкоскоростных, каждый из которых передается на отдельной частоте с последующим их объединением в один сигнал. При этом PLC-устройства могут «видеть» и декодировать информацию, хотя обычные электрические устройства – лампы накаливания, двигатели и т. п. – даже «не догадываются» о присутствии сигналов сетевого трафика и работают в обычном режиме.

Технология PLC

Основой технологии Power Line является использование частотного разделения сигнала, при котором высокоскоростной поток данных разбирается на несколько относительно низкоскоростных потоков, каждый из которых передается на отдельной поднесущей частоте с последующим их объединением в один сигнал. Реально в технологии Power Line используются 84 поднесущие частоты в диапазоне 4-21 Мгц.
PLC включает BPL (Broadband over Power Lines – широкополосная передача через линии электропередачи), обеспечивающий передачу данных со скоростью более 1 Мбит в секунду, и NPL (Narrowband over Power Lines – узкополосная передача через линии электропередач) с намного меньшими скоростями передачи данных.

Технология PLC

При передаче сигналов по бытовой электросети могут возникать большие затухания в передающей функции на определенных частотах, что может привести к потере данных. В технологии PowerLine предусмотрен специальный метод решения этой проблемы – динамическое включение и выключение передачи сигнала (dynamically turning off and on data-carrying signals). Суть данного метода заключается в том, что устройство осуществляет постоянный мониторинг канала передачи с целью выявления участка спектра с превышением определенного порогового значения затухания. В случае обнаружения данного факта, использование этих частот на время прекращается до восстановления нормального значения затухания.

Технология PLC

Существует также проблема возникновения импульсных помех (до 1 микросекунды), источниками которых могут быть галогенные лампы, а также включение и выключение мощных бытовых электроприборов, оборудованных электрическими двигателями.

PDSL – технология семейства xDSL, обеспечивающая симметричную передачу данных со скоростью до 2Мбит/с по силовым кабелям (4-20 кВ), параллельно с транспортируемым электричеством. Подключение оборудования PDSL к высоковольтным линиям осуществляется посредством устройств сопряжения, которые устанавливаются в трансформаторных шкафах.

Преимущества

- Не требуется прокладка кабеля, заключение его в короба, сверление; стен и опорных конструкций;
- Простота использования;
- Скорость монтажа

Преимущества PLC по сравнению с Wi-Fi

- Не требует настроек;
- Более стабильная связь;
- Большая безопасность информации;
- Подходит для передачи Multicast-трафика, например, IPTV;
- На качество связи не влияет материал и толщина стен в квартире;

Недостатки

- Нарушение радиоприема в помещениях, где работают PLC-модемы, особенно на средних и коротких волнах, но на очень небольшом расстоянии порядка 3-5 метров от модема.
- Пропускная способность сети по электропроводке делится между всеми её участниками. Например, если в одной Powerline-сети две пары адаптеров активно обмениваются информацией, то скорость обмена для каждой пары будет составлять примерно по 50% от общей пропускной способности.
- На стабильность и скорость работы PLC влияет качество выполнения электропроводки, наличие стыков из разных материалов (например, медного и алюминиевого проводника), а также просто количество соединений проводника.

Недостатки

- Не работает через сетевые фильтры и ИБП, не оборудованные специальными розетками "PLC READY".
- На качество связи могут оказывать отрицательное влияние дешевые энергосберегающие лампы, тиристорные диммеры, импульсные блоки питания и зарядные устройства. Максимальное влияние на скорость в сети перечисленные устройства оказывают при подключении в непосредственной близости от PLC-модема.
- Неясные правовые аспекты использования данной технологии в Российской Федерации.

Технология M2M

M2M (Machine-to-Machine) – межмашинная коммуникация. Обычно системы M2M строятся на базе сотовых GPRS модемов. Связь осуществляется через встроенный в модуль модема стек TCP/IP. В связи с стем, что скорость обмена по сотовым сетям не очень велика, есть существенные задержки сигнала и возможны обрывы связи, основным применением таких систем является:

- Управление медленными процессами на больших территориях (системы управления наружным освещением, АСКУЭ, системы «умный дом», системы автоматизации ЖКХ);
- Сбор телеметрической информации с территориально-распределенных систем;
- Системы сигнализации для стационарных и подвижных объектов;
- Сбор информации о местонахождении подвижных объектов (учет пробега транспорта и так далее).

Стандарт ARINC 429

ARINC 429 - стандарт на компьютерную шину для применения в авионике. Разработан фирмой ARINC. Стандарт описывает основные функции и необходимые физические и электрические интерфейсы для цифровой информационной системы самолёта. Сегодня, ARINC 429 является доминирующей авиационной шиной для большинства хорошо экипированных самолётов.

ARINC 429 является двухпроводной шиной данных. Соединительные проводники – витые пары. Размер слова составляет 32 бита, а большинство сообщений состоит из единственного слова данных. Спецификация определяет электрические характеристики, характеристики обмена данными и протоколы. ARINC 429 использует однонаправленный стандарт шины данных (линии передачи и приёма физически разделены). Сообщения передаются на одной из трёх скоростей: 12,5, 50 или 100 Кбит/сек. Передатчик всегда активен, он либо передаёт 32-битовые слова данных или выдаёт «пустой» уровень. На шине допускается не более 20 приёмников, и не более одного передатчика.

Стандарт MIL-STD-1553

MIL-STD-1553 (MIL-STD-1553B) – стандарт Министерства обороны США, распространяется на магистральный последовательный интерфейс (МПИ) с централизованным управлением, применяемый в системе электронных модулей. ГОСТ 26765.52-87, ГОСТ Р 52070-2003, МКИО - российский аналог американского военного стандарта MIL-STD-1553 (MIL-STD-1553B).

Изначально разрабатывался по заказу МО США для использования в военной бортовой авионике, однако позднее спектр его применения существенно расширился, стандарт стал применяться и в гражданских системах. Особенностью интерфейса является двойная избыточная линия передачи информации, полудуплексный протокол <команда-ответ> и до 31 удалённого абонента (оконечного устройства). Каждая линия управляется своим контроллером канала.

Стандарт MIL-STD-1553

Стандарт устанавливает требования к:

- составу технических средств интерфейса;
- организации контроля передачи информации;
- характеристикам линии передачи информации (ЛПИ);
- характеристикам устройств интерфейса;
- интерфейсу с резервированием.

Впервые опубликован в США как стандарт ВВС в 1973 году, применён на истребителе F-16. Принят в качестве стандарта НАТО - STANAG 3838 AVS. В новейших самолетах заменяется стандартом IEEE 1394b.

Физический уровень

Одна шина состоит из пары проводов с волновым сопротивлением 70-85 Ом при частоте 1 МГц. Для соединения используется круглый разъём, по центральной ножке которого передаётся сигнал, закодированный Манчестерским кодом. Принимающее и передающее оконечные устройства подключаются к шине с использованием трансформаторной развязки, а не задействованные подключения отделяются с использованием пары изолирующих резисторов, развязанных через трансформатор. Это уменьшает влияние короткого замыкания и добавляет уверенности, что через шину ток не течёт по корпусу самолёта. Манчестерский код используется для того, чтобы передавать сигнал данных и сигнал синхронизации по одной паре проводников, а также для исключения любых постоянных составляющих, задерживаемых трансформаторной развязкой. Пропускная способность канала составляет 1 Мбит/с. Допуск на погрешность и долговременный дрейф пропускной способности составляет 0,1 %; краткосрочная стабильность тактовых импульсов должна быть в пределе 0,01 %. Амплитуда входного напряжения передатчика должна составлять 18-27 В.

Физический уровень

Избыточность сообщений в системе передачи информации передачи может быть достигнута за счёт использования двух или трёх независимых каналов (проводников), к которым подключены все устройства на шине. Эти меры предосторожности приняты для того, чтобы можно было задействовать дублирующий контроллер шины, в случае отказа используемого в текущий момент.

Также существует вторая версия стандарта, известная как MIL-STD-1773, в которой в качестве канала передачи информации используется оптоволокно, имеющее меньший вес и лучшие показатели по электромагнитной совместимости.

Физический уровень

Типичная шина MIL-STD-1553B может состоять из:

- Двух каналов (основного и резервного);
- Контроллера шины;
- Оконечных устройств;
- Монитора канала.

Контроллер шины

На одной шине может быть всего один контроллер в любой момент времени. Он является инициатором всех сообщений по этой шине.

Контроллер:

- Оперирует командами из списка в своей внутренней памяти;
- Командует оконечным устройствам послать или принять сообщения;
- Обслуживает запросы, получаемые от оконечных устройств;
- Фиксирует и восстанавливает ошибки;
- Поддерживает историю ошибок.

Оконечные устройства

Оконечные устройства служат для:

- Организации взаимодействия шины и подключаемой подсистемы;
- Организации моста между двумя шинами.

Монитор канала

Монитор канала отличается от оконечного устройства тем, что не может передавать сообщения по шине. Его роль заключается в мониторинге и записи транзакций по шине, без вмешательства во взаимодействие контроллера и оконечных устройств. Эта запись может быть использована для последующего анализа.