

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего профессионального образования

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ

**ВНУТРИПЛАТНЫЕ ИНТЕРФЕЙСЫ
ВСТРАИВАЕМЫХ СИСТЕМ**

Учебное пособие

Санкт-Петербург
2012

Востриков А.А., Балонин Н.А., Сергеев А.М.

Вхх Внутриплатаые интерфейсы встраиваемых систем: Учеб. пособие/
СПбГУАП. СПб., 2012.

Рецензенты:

кафедра информационно-сетевых технологий ГУАП;
доктор технических наук, профессор З.М. Юлдашев.

В учебном пособии приведены общие сведения об интерфейсах, изложена специфика внутриплатаых интерфейсов встраиваемых систем и описаны интерфейсы, наиболее часто применяемые в настоящее время.

Основное внимание уделено вопросам, связанным с организацией современных внутриплатаых интерфейсов, их использованием для объединения микропроцессоров и контроллеров на платах устройств встраиваемого класса, их взаимодействия с остальными компонентами цифровых микропроцессорных модулей

Учебное пособие предназначено для студентов, обучающихся по специальности 230101 и многим смежным специальностям информационного направления.

Утверждено
редакционно-издательским советом университета
в качестве учебного пособия

© ГОУ ВПО «СПбГУАП», 2012
© А.А.Востриков, Балонин Н.А.
Сергеев А.М., 2012

ОГЛАВЛЕНИЕ

| | |
|---|-----------|
| ВВЕДЕНИЕ | 5 |
| 1. ИНТЕРФЕЙСЫ. КЛАССИФИКАЦИЯ И ТЕРМИНОЛОГИЯ | 7 |
| 1.1. ХАРАКТЕРИСТИКИ ИНТЕРФЕЙСОВ. ИНТЕРФЕЙСЫ ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРОВ | 7 |
| 1.2. ИНТЕРФЕЙСЫ ВО ВСТРАИВАЕМЫХ СИСТЕМАХ | 15 |
| 2. SPI. ПОСЛЕДОВАТЕЛЬНЫЙ ПЕРИФЕРИЙНЫЙ ИНТЕРФЕЙС | 18 |
| 2.1. СИГНАЛЬНЫЕ ЛИНИИ ИНТЕРФЕЙСА И ФОРМАТ ПЕРЕДАЧИ ДАННЫХ | 18 |
| 2.2. ВАРИАНТЫ АРХИТЕКТУРЫ ИНФОРМАЦИОННОГО ВЗАИМОДЕЙСТВИЯ | 20 |
| 2.3. QSPI. ПОСЛЕДОВАТЕЛЬНЫЙ ПЕРИФЕРИЙНЫЙ ИНТЕРФЕЙС С ОЧЕРЕДЬЮ | 25 |
| 2.4. MICROWIRE | 25 |
| 2.5. АЛЬТЕРНАТИВНЫЕ НАИМЕНОВАНИЯ ИНТЕРФЕЙСА SPI | 26 |
| 3. I²C. ШИНА ИНФОРМАЦИОННОГО ОБМЕНА ИНТЕГРАЛЬНЫХ СХЕМ | 28 |
| 3.1. АРХИТЕКТУРА СИСТЕМ С ИНТЕРФЕЙСОМ I ² C | 29 |
| 3.2. АППАРАТНАЯ РЕАЛИЗАЦИЯ ИНТЕРФЕЙСА | 30 |
| 3.3. ФОРМАТЫ ПЕРЕДАЧИ ДАННЫХ | 35 |
| 3.4. НЕКОТОРЫЕ ПРАКТИЧЕСКИЕ ПРИМЕНЕНИЯ ИНТЕРФЕЙСА I ² C | 40 |
| 4. SMBUS. ШИНА СИСТЕМНОГО УПРАВЛЕНИЯ | 44 |
| 4.1. ФИЗИЧЕСКИЙ УРОВЕНЬ СПЕЦИФИКАЦИИ SMBUS | 45 |
| 4.2. УРОВЕНЬ СВЯЗИ ДАННЫХ СПЕЦИФИКАЦИИ SMBUS | 47 |
| 4.3. СЕТЕВОЙ УРОВЕНЬ СПЕЦИФИКАЦИИ SMBUS | 47 |
| 4.3.1. Статическая адресация в SMBus и команды интерфейса | 48 |
| 4.3.2. Автоматическое назначение адресов в SMBus | 52 |
| 4.4. ПОДДЕРЖКА SMBUS В BIOS ПК И ACPI | 56 |
| 4.5. СРАВНЕНИЕ ШИН I ² C И SMBUS | 60 |
| 5. 1-WIRE. ОДНОПРОВОДНЫЙ ИНТЕРФЕЙС | 61 |
| 5.1. ОСНОВНЫЕ ПРИНЦИПЫ ОРГАНИЗАЦИИ | 62 |
| 5.2. ОРГАНИЗАЦИЯ ВЕДУЩИХ УСТРОЙСТВ | 65 |
| 5.3. ВЕДОМЫЕ ОДНОПРОВОДНЫЕ КОМПОНЕНТЫ | 68 |
| 5.4. СХЕМНАЯ РЕАЛИЗАЦИЯ 1-WIRE ИНТЕРФЕЙСА | 70 |
| 5.5. СИНХРОНИЗАЦИЯ И ПОБИТНАЯ ПЕРЕДАЧА ИНФОРМАЦИИ | 72 |
| 5.5.1. Синхронизация при передаче | 72 |
| 5.5.2. Передача информационного бита | 73 |
| 5.5.3. Получение информационного бита | 75 |
| 5.5.4. Параметры временных интервалов | 75 |
| 5.5.5. Побитная группировка последовательности бит | 77 |
| 5.6. СБРОС И ОБНАРУЖЕНИЕ ПРИСУТСТВИЯ НА ЛИНИИ | 78 |
| 5.6.1. Команды шины 1-Wire | 79 |
| 6. I²S. ИНТЕРФЕЙС ПЕРЕДАЧИ ЦИФРОВЫХ АУДИОСИГНАЛОВ | 81 |
| 6.1. ОСНОВНЫЕ ТРЕБОВАНИЯ | 81 |
| 6.2. ПРИНЦИП ПЕРЕДАЧИ ИНФОРМАЦИИ | 83 |
| 6.2.1. Линия последовательной передачи данных | 83 |
| 6.2.2. Линия выбора канала | 84 |
| 6.3. ВРЕМЕННЫЕ ТРЕБОВАНИЯ | 85 |
| 6.4. ТРЕБОВАНИЯ К УРОВНЯМ НАПРЯЖЕНИЯ СИГНАЛОВ | 86 |
| 7. АСИНХРОННЫЙ ИНТЕРФЕЙС ПОСЛЕДОВАТЕЛЬНОГО ПОРТА | 87 |
| 7.1. ТИПЫ ВЗАИМОДЕЙСТВУЮЩЕГО ОБОРУДОВАНИЯ И НАБОР СИГНАЛОВ | 87 |
| 7.2. КОДИРОВАНИЕ ИНФОРМАЦИИ ПРИ ПЕРЕДАЧЕ | 90 |
| 7.3. ОСОБЕННОСТИ ВНУТРИПЛАТНОГО ПРИМЕНЕНИЯ RS-232C | 92 |

| | |
|--|------------|
| 8. LVDS. НИЗКОВОЛЬТНАЯ ДИФФЕРЕНЦИАЛЬНАЯ ПЕРЕДАЧА СИГНАЛОВ | 94 |
| 8.1. ВОПРОСЫ СТАНДАРТИЗАЦИИ..... | 94 |
| 8.2. СПОСОБ ПЕРЕДАЧИ СИГНАЛОВ | 97 |
| 8.3. КОНФИГУРАЦИЯ СИСТЕМ С ИНТЕРФЕЙСОМ LVDS..... | 99 |
| 8.4. СОГЛАСОВАНИЕ СИГНАЛОВ ПРИ ПЕРЕДАЧЕ..... | 101 |
| 8.5. МАКСИМАЛЬНАЯ СКОРОСТЬ ПЕРЕКЛЮЧЕНИЯ В ЛИНИИ | 102 |
| 8.6. ЭНЕРГОСБЕРЕЖЕНИЕ | 102 |
| 8.7. ЭКОНОМИЧНАЯ АППАРАТНАЯ РЕАЛИЗАЦИЯ..... | 103 |
| 8.8. ОБЛАСТИ ПРИМЕНЕНИЯ LVDS | 104 |
| 8.9. ИНТЕГРАЛЬНЫЕ СХЕМЫ С ИНТЕРФЕЙСОМ LVDS..... | 105 |
| 9. СПЕЦИФИЧЕСКИЕ ФИРМЕННЫЕ ИНТЕРФЕЙСЫ..... | 107 |
| 9.1. УНИВЕРСАЛЬНЫЕ ПЕРИФЕРИЙНЫЕ БЛОКИ «НА КРИСТАЛЛЕ»..... | 107 |
| 9.2. ИНТЕРФЕЙС UNI/O | 108 |
| 9.2.1. <i>Схема взаимодействия устройств и адресация</i> | <i>109</i> |
| 9.2.2. <i>Команды интерфейса и формат передачи данных.....</i> | <i>110</i> |
| ЗАКЛЮЧЕНИЕ | 113 |
| БИБЛИОГРАФИЧЕСКИЙ СПИСОК..... | 114 |

ВВЕДЕНИЕ

Стремительное развитие микроэлектроники в последние десятилетия привело к повсеместному распространению малогабаритных и низкопотребляющих контроллеров, реализованных на основе программно управляемых интегральных схем – микропроцессоров и микроконтроллеров – и на основе схем программируемой логики. Такие контроллеры чаще всего являются частью различных программно-аппаратных комплексов, т.е. встраиваются в них, вследствие чего они условно выделяются в отдельный класс «встраиваемых систем» (соответствующий англоязычный термин – embedded systems).

Несмотря на большое разнообразие периферийных блоков и блоков памяти, располагаемых непосредственно на кристалле вычислителя, при проектировании разработчик очень часто сталкивается с необходимостью применения дополнительных интегральных схем. Причём, чем шире функциональные возможности контроллера, тем объемнее спецификация электронных компонентов.

Для обмена информацией между интегральными схемами, расположенными на печатной плате устройства, применяются интерфейсы, обладающие рядом специфических особенностей. В частности, предельно ограничивается количество линий, входящих в состав интерфейса, ограничивается пропускная способность и почти всегда реализуется схема взаимодействия «ведущий-ведомый».

Некоторые из таких интерфейсов применяются во встраиваемых системах настолько давно и широко, что стали для инженера-электроника своеобразной «классикой» (например, SPI и I²C). Другие же являются относительно новыми – разработанными для удовлетворения непрерывно возрастающих функциональных требований к системам данного класса (например, I²S и LVDS).

В данном учебном пособии приведены общие сведения об интерфейсах, изложена специфика внутриплатных интерфейсов встраиваемых систем и описаны интерфейсы, наиболее часто применяемые в настоящее время.

1. ИНТЕРФЕЙСЫ. КЛАССИФИКАЦИЯ И ТЕРМИНОЛОГИЯ

Толковый словарь по вычислительным системам определяет понятие «интерфейс» (interface) как *границу раздела двух систем, устройств или программ; элементы соединения и вспомогательные схемы управления, используемые для соединения устройств*. В ГОСТ Р 50304-92 понятию «интерфейс» дается следующее определение: *совокупность средств и правил, обеспечивающих взаимодействие устройств вычислительной машины или системы обработки информации и (или) программы*. Кроме этого в том же ГОСТ отдельно определяется «стандартный интерфейс»: *интерфейс, взаимодействие радиоэлектронных средств по которому регламентировано нормативным документом* [1].

В описании любого интерфейса содержится перечень правил, соблюдение которых необходимо для обеспечения передачи информации между электронными устройствами.

1.1. Характеристики интерфейсов. Интерфейсы персональных компьютеров

Интерфейсы позволяют подключать к компьютерам или встраиваемым контроллерам разнообразные периферийные устройства, а также соединять отдельные подсистемы компьютера в системном блоке или интегральные микросхемы на печатной плате электронного модуля.

Интерфейсы периферийных устройств персональных компьютеров известны гораздо лучше, чем интерфейсы интегральных схем. Это связано с тем, что с применением первых регулярно сталкиваются обычные пользователи вычислительной техники, а применение вторых – скорее удел инженеров-электронщиков. В связи с этим классификацию и рассмотрение характеристик лучше проводить с примерами компьютерных интерфейсов.

Ключевыми параметрами идентификации конкретного интерфейса являются следующие (см. рис. 1):

- способ передачи информации;
- способ синхронизации;
- режим передачи информации;
- структура шины;
- максимальная пропускная способность.

Кроме этого, также учитываются (см. рис. 2):

- способ кодирования сигналов при передаче;
- наличие и метод контроля целостности информации;
- возможность «горячего» подключения и отключения периферии;
- допустимое удаление соединяемых устройств друг от друга;
- наличие или отсутствие гальванической развязки интерфейсных линий.

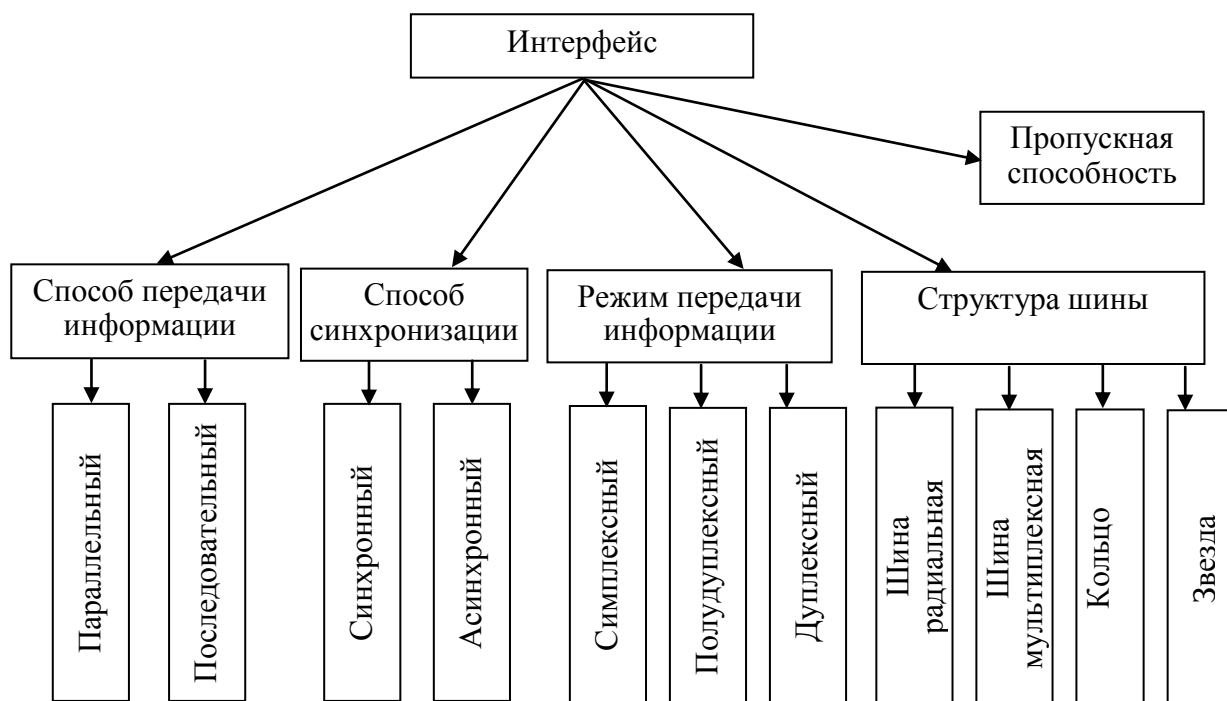


Рисунок 1. Ключевые параметры идентификации интерфейса

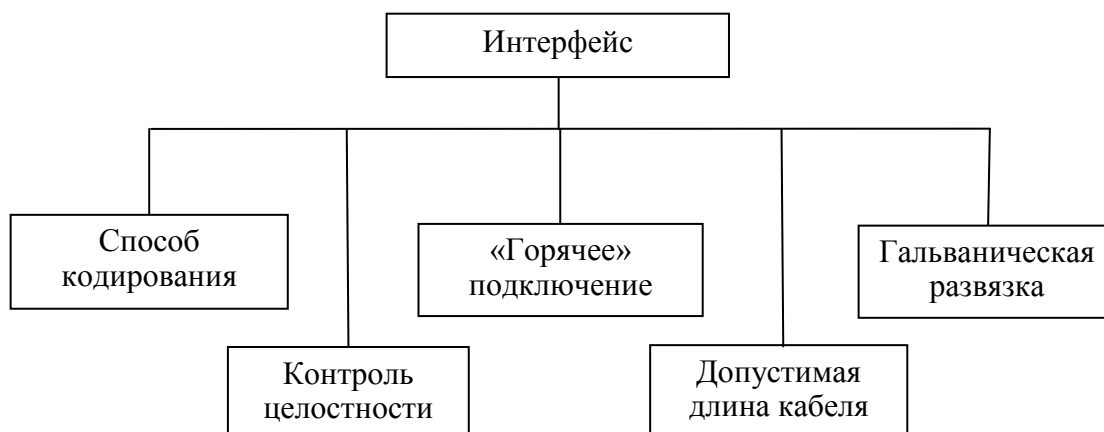


Рисунок 2. Функциональные характеристики интерфейсов

По способу передачи информации интерфейсы подразделяются на параллельные и последовательные. В параллельном интерфейсе все биты передаваемого слова (обычно байта) выставляются и передаются по соответствующим параллельно идущим проводам (проводникам) одновременно. В персональных компьютерах традиционно используется параллельный интерфейс Centronics, реализуемый LPT-портами, шины ATA, SCSI и все шины расширения. В последовательном интерфейсе биты передаются друг за другом, обычно по одной (иногда по более чем одной) линии. Эта линия может быть как однонаправленной (например, в RS-232C, реализуемой COM-портом, шине FireWire, SPI, JTAG), так и двунаправленной (I²C, USB, PC).

При рассмотрении интерфейсов важным параметром является пропускная способность. Технический прогресс приводит к неуклонному росту объемов передаваемой информации. Если раньше матричные принтеры, печатающие в символьном режиме, могли обходиться и COM-портом с невысокой пропускной способностью, то современным лазерным принтерам при высоком разрешении не хватает производительности даже самых быстрых LPT-портов. То же касается и сканеров. А передача «живого»

видео, даже с применением компрессии, требует ранее немислимой пропускной способности.

Вполне очевидно, что при одинаковом быстродействии приемопередающих цепей и пропускной способности соединительных линий по скорости передачи параллельный интерфейс должен превосходить последовательный. Однако повышение производительности за счет увеличения тактовой частоты передачи данных упирается в волновые свойства соединительных проводников. В случае параллельного интерфейса начинают сказываться задержки сигналов при их прохождении по линиям кабеля и, что самое неприятное, задержки в разных линиях интерфейса могут быть различными вследствие неидентичности проводов и контактов разъемов. Для надежной передачи данных временные диаграммы обмена строятся с учетом возможного разброса времени прохождения сигналов, что является одним из факторов, сдерживающих рост пропускной способности параллельных интерфейсов. В последовательных интерфейсах, конечно же, есть свои проблемы повышения производительности, но, поскольку в них используется меньшее число линий (в пределе – одна), повышение пропускной способности линий связи обходится дешевле.

Очевидно, что при передаче информации необходима синхронизация, поскольку устройство, ожидающее приёма очередного информационного слова, должно определить момент, когда это слово (бит) окажется на его входном порте данных. В случае синхронных интерфейсов для этого используется дополнительная линия, изменение уровня сигнала на которой является признаком присутствия на линии данных очередной порции информации. При использовании асинхронных интерфейсов – такая линия отсутствует, и для взаимной синхронизации используется принудительное изменение линии данных, даже если передаваемое слово (бит) повторяет предыдущий (такой метод используется в так называемом «манчестерском коде»). Кроме этого, устройства, соединенные асинхронным интерфейсом могут ориентироваться на внутренние часы – внутренние тактовые сигналы,

которые должны быть синхронизированы непосредственно перед началом передачи.

Для повышения пропускной способности синхронных параллельных интерфейсов с середины 90-х годов стали применять двойную синхронизацию DDK (Dual Data Rate). Её идея заключается в выравнивании частот переключения информационных сигнальных линий и линий стробирования (синхронизации). В «классическом» варианте данные информационных линий воспринимались только по одному перепаду (фронту или спаду) синхросигнала, что удваивает частоту переключения линии синхросигнала относительно линий данных. При двойной синхронизации данные воспринимаются и по фронту, и по спаду, так что частота смены состояний всех линий выравнивается, что при одних и тех же физических параметрах кабеля и интерфейсных схем позволяет удвоить пропускную способность. Волна этих модернизаций началась с интерфейса ATA (режимы UltraDMA) и прошла уже по SCSI (Ultra160 и выше), и по интерфейсам с модулями памяти (DDK SDRAM), а также по системной шине процессоров (Pentium 4).

Для интерфейса, соединяющего (физически или логически) два устройства, различают три возможных режима обмена — дуплексный, полудуплексный и симплексный. Дуплексный режим позволяет по одному каналу связи одновременно передавать информацию в обоих направлениях. Он может быть асимметричным, если значения пропускной способности в разных направлениях существенно различаются, или симметричным. Полудуплексный режим позволяет передавать информацию в разных направлениях поочередно, при этом интерфейс имеет средства переключения направления канала. Симплексный (односторонний) режим предусматривает только одно направление передачи информации (иногда во встречном направлении передаются вспомогательные сигналы интерфейса).

Немаловажен для интерфейса контроль достоверности передачи данных, который, увы, имеется далеко не везде. «Ветераном» контроля является шина SCSI с её битом паритета; контроль паритета применяется и в

последовательных интерфейсах и в шине PCI. Шина ISA в этом плане беззащитна, как и ее «потомок» — интерфейс ATA, в котором до введения режима UltraDMA контроля достоверности не было. В новых интерфейсах контролю достоверности уделяется серьезное внимание, поскольку они, как правило, рассчитываются на экстремальные условия работы (высокие частоты, большие расстояния и помехи). Контроль достоверности может производиться и на более высоких уровнях реализации протокола (контроль целостности пакетов и их полей), но на аппаратном уровне он работает, естественно, быстрее.

Другим немаловажным параметром интерфейса является допустимое удаление соединяемых устройств. Оно ограничивается как частотными свойствами кабелей, так и помехозащищенностью интерфейсов. Часть помех возникает от соседних линий интерфейса — это перекрестные помехи, защитой от которых может быть применение витых пар проводов для каждой линии. Другая часть помех вызывается искажением уровней сигналов.

С появлением шин USB и Fire Wire в качестве характеристики интерфейса стала фигурировать и топология соединения. Для интерфейсов RS-232C и Centronics практически всегда применялась двухточечная топология ПК-устройство (или ПК-ПК). Исключениями из этого правила являются различные устройства безопасности и защиты данных (Security devices), которые подключаются к COM- или LPT-портам, но имеют разъем для подключения внешнего устройства. Однако эти устройства для традиционной периферии прозрачны, поэтому можно считать, что они не нарушают общего правила. Аналогично обстоит дело и с адаптерами локальных сетей (например, Paraport) и внешних дисковых накопителей (Omega Zip), подключаемых к LPT-портам. Хотя разрабатываемые стандарты для параллельного порта (IEEE 1284.3) и предусматривают соединение устройств в цепочку (Daisy Chain) или через мультиплексоры, широкого распространения такие способы подключения пока не получили. К другому классу исключений относится построение моноканала на COM-

портах, которое 10-15 лет назад применялось в «любительских» локальных сетях, но было вытеснено существенно более эффективной и подешевевшей технологией Ethernet. Интерфейсные шины USB и Fire Wire реализуют древовидную топологию, в которой внешние устройства могут быть как оконечными, так и промежуточными (разветвителями). Эта топология позволяет подключать множество устройств к одному порту USB или Fire Wire.

Важным свойством интерфейса, которому часто не уделяют должного внимания, является гальваническая развязка, а точнее – её отсутствие. «Схемные земли» устройств, соединяемых интерфейсом с COM- или LPT-портом ПК, оказываются связанными со «схемной землей» компьютера (а через интерфейсный кабель и между собой). Если между ними до подключения интерфейса была разность потенциалов, то по общему проводу интерфейса потечет уравнивающий ток, что плохо по целому ряду причин. Падение напряжения на общем проводе, вызванное протеканием этого тока, приводит к смещению уровней сигналов, а протекание переменного тока приводит к сложению полезного сигнала с переменной составляющей помехи. К этим помехам особенно чувствительны TTL-интерфейсы; в то же время в RS-232C смещение и помеху в пределах 2В поглотит зона нечувствительности. В случае обрыва общего провода или плохого контакта, а гораздо чаще – при подключении и отключении интерфейсов без выключения питания устройств, разность потенциалов прикладывается к сигнальным цепям, а протекание уравнивающих токов через них часто приводит к искрению, подгоранию контактов и нагреву кабелей. Гальваническую развязку устройств обеспечивают интерфейсы MIDI, «токовая петля», S/PDIF, шина FireWire, сетевые интерфейсы Ethernet.

Существенным свойством является возможность «горячего» подключения/отключения или замены устройств (Hot Swap), причем в двух аспектах. Во-первых, это безопасность переключений «на ходу» как для самих устройств и их интерфейсных схем, так и для целостности хранящихся

и передаваемых данных и, наконец, для человека. Во-вторых, это возможность использования вновь подключенных устройств без перезагрузки системы, а также продолжения устойчивой работы системы при отключении устройств. Далеко не все внешние интерфейсы поддерживают «горячее подключение» в полном объеме, так, например, зачастую сканер с интерфейсом SCSI должен быть подключен к компьютеру и включен до загрузки ОС, иначе он не будет доступен системе. С новыми шинами USB и FireWire проблем «горячего подключения» не возникает. Для внутренних интерфейсов «горячее подключение» несвойственно. Это касается и шин расширения, и модулей памяти, и даже большинства дисков ATA и SCSI. «Горячее подключение» поддерживается для шин расширения промышленных компьютеров, а также в специальных конструкциях массивов устройств хранения.

В ряде интерфейсов заложены возможности PnP (Plug and Play — включай и играй), которые предназначены для снятия с пользователей забот по конфигурированию подключаемых устройств. В современных интерфейсах эти возможности закладывались изначально (PCI, USB, FireWire, Bluetooth), и эти функции в большинстве случаев работают нормально. Однако для интерфейсов-ветеранов (например, ISA, SCSI) технология PnP является поздней искусственной надстройкой, работающей с переменным успехом. Часто побочные эффекты вызваны соседством устройств PnP с традиционными (legacy) устройствами. На закате шины ISA ее система PnP в общем работала, но в SCSI от идей автоконфигурирования со временем отказались.

При разработке встраиваемых систем встает вопрос выбора подходящего интерфейса подключения. Этот вопрос следует решать, исходя из принципа разумной достаточности, по возможности отдавая предпочтение внешним интерфейсам. Следует помнить, что разработка аппаратной части устройства («hardware») тесно связана и с программной поддержкой устройств — как модулями ПО, исполняемыми процессором компьютера

(«software»), так и программами встроенного микроконтроллера («firmware»), на базе которого, как правило, строятся современные устройства. Промышленностью выпускается множество моделей микроконтроллеров, имеющих популярные интерфейсы (USB, RS-232, PC и другие). Однако в ряде случаев приходится использовать и стандартизованные шины расширения ввода-вывода. Эти шины предоставляют более широкие возможности для взаимодействия процессора с аппаратурой, не скованные жесткими ограничениями внешних интерфейсов. Однако за универсальность и производительность внутренних шин расширения приходится расплачиваться более замысловатой реализацией интерфейсных схем и сложностями при обеспечении совместимости с другим установленным в компьютер оборудованием. Здесь ошибки могут приводить к потере работоспособности компьютера. Недаром серьезные производители компьютеров гарантируют работоспособность своих изделий только при установке сертифицированных карт расширения. При использовании внешних интерфейсов неприятности в случае ошибок чаще всего имеют отношение только к подключаемому устройству.

1.2. Интерфейсы во встраиваемых системах

Программно-аппаратные системы встраиваемого класса отличаются малыми габаритами и весом, а также относительно малым энергопотреблением. Зачастую они выполняют вспомогательные функции в составе сложных комплексов (панели ввода-вывода информации, всевозможные автоматические регуляторы и т.д.), однако могут представлять собой и самостоятельные законченные устройства.

В процессе функционирования происходит информационное взаимодействие на разных уровнях организации систем:

- вертикально – между различными уровнями организационной иерархии (прием информации для обработки, возврат

обработанной информации, передача сигналов о собственном состоянии или мониторинг других устройств);

- горизонтально – между устройствами, выполняющими сходные функции;
- внутри устройства между электронными компонентами;
- с внешними устройствами и системами.

Любое из таких взаимодействий требует от участников информационного обмена соблюдения правил, т.е. следования технической спецификации используемого интерфейса.

Выбор же конкретного интерфейса делается проектировщиком исходя из следующих основных критериев:

1. необходимая пропускная способность при заданной разрядности и длине физической линии передачи;
2. минимизация дополнительных аппаратных средств и программных разработок для организации интерфейса.

Второй из вышеуказанных критериев оказывает особенно сильное влияние на выбор внутрислатных интерфейсов, ввиду специфических характеристик устройств встраиваемого класса: добавление нового интерфейса способно привести, например, к увеличению габаритов и энергопотребления из-за необходимости установки интегральной схемы, реализующей данный интерфейс на физическом и/или логическом уровне.

В настоящее время производители микроконтроллеров и схем программируемой логики стремятся охватить своей номенклатурой наибольшее число интерфейсов, применяемых для обмена информацией внутри печатной платы, а также вне её (см. рис. 3). В результате, проектировщик избавляется от необходимости расширять электрическую схему устройства и объем встраиваемого программного обеспечения для обмена данными с периферийными для центрального вычислителя аналого-цифровыми преобразователями (АЦП), цифро-аналоговыми преобразователями (ЦАП), аккумуляторными источниками питания со

встроенными контроллерами, расширителями портов, датчиками температуры и т.д.

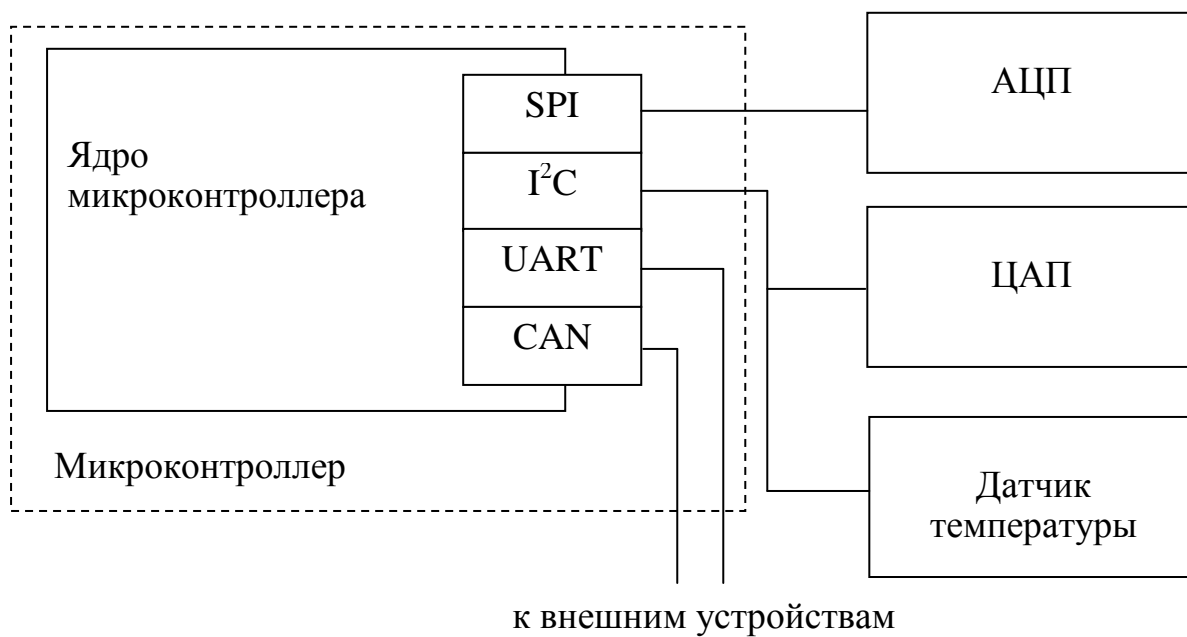


Рисунок 3. Оснащение микроконтроллера периферийными модулями «на кристалле»

2. SPI. ПОСЛЕДОВАТЕЛЬНЫЙ ПЕРИФЕРИЙНЫЙ ИНТЕРФЕЙС

Последовательный периферийный интерфейс SPI разработан, как полнодуплексный синхронный четырехпроводный интерфейс с шинной конфигурацией подключаемых узлов (устройств) для систем с одним главным узлом. Первоначальная базовая версия интерфейса SPI позволяет подключать к одному главному (или ведущему) «Master»-узлу несколько ведомых «Slave»-узлов через общую шину. Отдельный сигнал выбора ведомого устройства NSS (Slave Select signal – сигнал выбора ведомого) используется для выбора ведомого устройства при осуществлении с ним обмена данными.

Специализированные интегральные схемы с интерфейсом SPI в большинстве случаев выступают в качестве ведомых, тогда как периферийный модуль интерфейса SPI в микроконтроллерах обычно настраивается для работы как в качестве ведущего, так и в качестве ведомого узла.

2.1. Сигнальные линии интерфейса и формат передачи данных

Интерфейс SPI имеет четыре сигнальные линии, имеющие следующие обозначения: MOSI, MISO, SCK и NSS.

Линия MOSI (Master-Out, Slave-In) - выходная линия данных ведущего интерфейса и входная линия данных ведомого интерфейса. Из названия следует, что линия предназначена для передачи данных от ведущего интерфейса (или узла шины) к ведомому интерфейсу (или узлу шины).

Линия MISO (Master-In, Slave-Out) - входная линия данных ведущего интерфейса и выходная линия данных ведомого интерфейса. Линия предназначена для передачи данных от ведомого интерфейса к ведущему. Данные передаются байтами, побитно, начиная со старшего бита. Следует помнить, что вывод MISO ведомого интерфейса находится в

высокоимпедансном состоянии, если ведомый интерфейс не выбран по линии NSS.

Линия NSS (Slave Select) – как уже упоминалось выше, это линия выборки ведомого, с которым ведущий намерен обмениваться информацией. Активный логический уровень для данного сигнала – низкий, т.е. для выбора ведомого необходимо на его вход NSS подать логический «0». Иногда применяется также альтернативное наименование линии – CS (Chip Select – выбор чипа).

Линия SCK (Serial Clock) - выходная линия тактовых импульсов ведущего узла и входная линия тактовых импульсов ведомого узла. Линия SCK используется для синхронизации передачи данных между ведущим и ведомым интерфейсами по линиям MOSI и MISO.

Каждый очередной бит информации фиксируется принимающей стороной (см. рис. 4) по переднему фронту на линии SCK (по изменению уровня от низкого к высокому). Следует отметить, что в некоторых реализациях интерфейса программисту предоставляется право выбора фронта (переднего или заднего), по которому осуществляется синхронизация при передаче. В этом случае в фазе инициализации микроконтроллера необходимо настроить соответствующий бит.

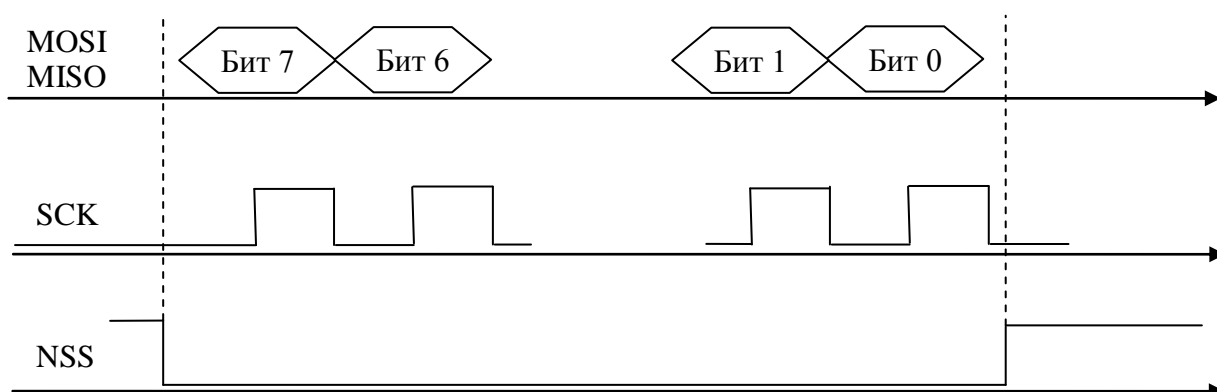


Рисунок 4. Временная диаграмма передачи одного байта информации по интерфейсу SPI

2.2. Варианты архитектуры информационного взаимодействия

В базовой версии шины на базе SPI интерфейсов только одно устройство может быть ведущим. Однако широко применяется режим работы шины со многими ведущими, реализуемый в современных микроконтроллерах, что выразилось в расширении средств диагностики ошибок и возможности управления сигналом NSS.

Классическая архитектура SPI шины для базового варианта интерфейса приведена на рис. 5. Это так называемая классическая четырехпроводная структура. Один ведущий в ней управляет несколькими (N) ведомыми. Все ведомые подключены параллельно на линиях SCLK, MISO и MOSI шины SPI. Выборка одного из ведомых происходит с помощью одной из линий портов ввода/вывода, которая соединяется с входом NSS соответствующего ведомого. Обычно такая архитектура используется для построения микроконтроллерных систем с одним микроконтроллером, выполняющим роль ведущего и рядом периферийных микросхем, выполняющих роль ведомых. В качестве периферийных микросхем может быть использован ряд современных микросхем, оснащенных интерфейсом SPI: таймеров реального времени (RTC – Real Time Clock), аналого-цифровых преобразователей (ADC – Analog to Digital Converter), цифро-аналоговых преобразователей (DAC – Digital to Analog Converter), различных микросхем памяти и т.п.

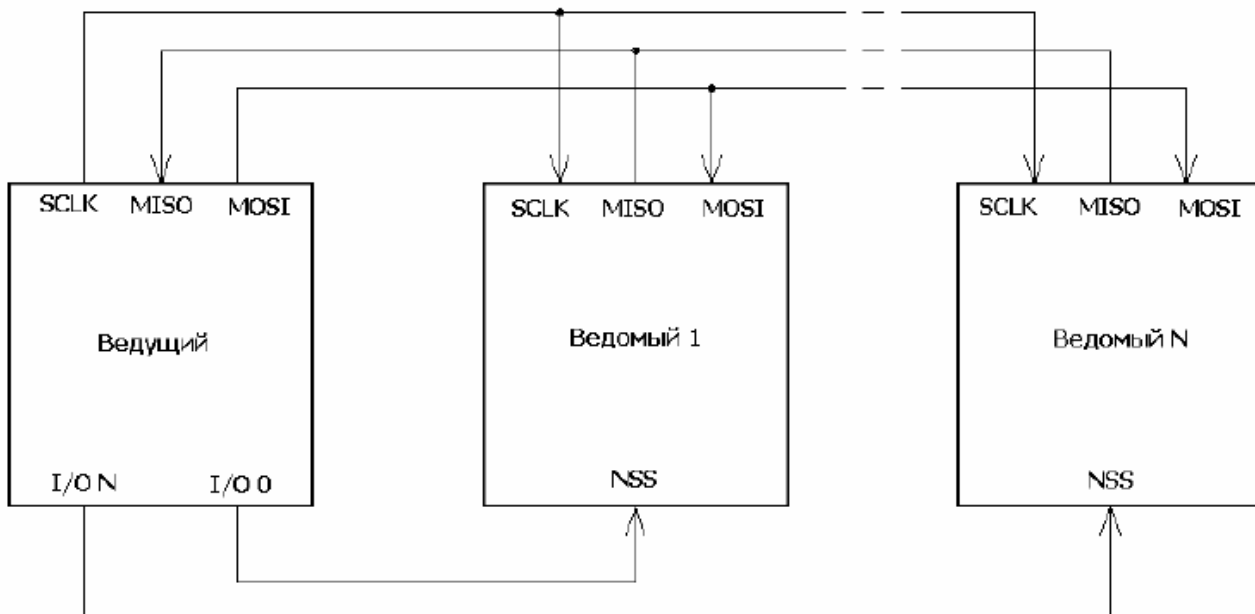


Рисунок 5. Архитектура четырехпроводной шины SPI с одним ведущим и несколькими ведомыми

Частным случаем предыдущей архитектуры является так называемая 3х-проводная структура, приведенная на рис. 6. Ее отличие состоит в том, что ведомая микросхема постоянно выбрана, и, следовательно, необходимость в четвертой линии интерфейса SPI отсутствует.

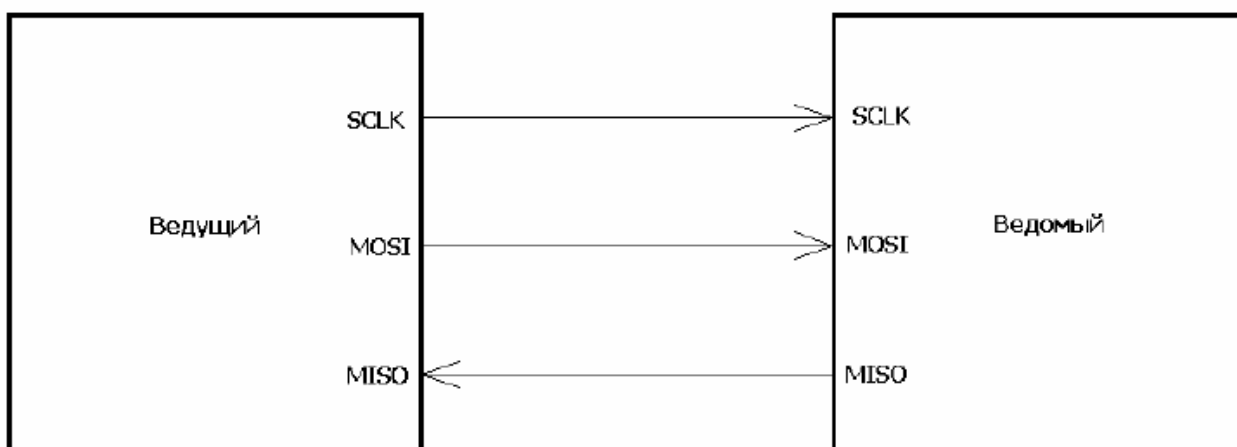


Рисунок 6. Архитектура трехпроводной шины SPI с одним ведущим и ведомым

Расширенный же вариант интерфейса позволяет организовать двухпроцессорную четырехпроводную архитектуру, приведенную на рис. 7.

В этом случае каждый из двух узлов может выполнять роль ведущего. На практике такая архитектура используется достаточно редко, ввиду алгоритмической сложности диагностики состояний коллизий (конфликтов, вызванных одновременной передачей на шину двух ведущих узлов). Более широко распространена более простая в понимании и диагностике конфликтов, так называемая архитектура с четырехпроводной «перекрестной» выборкой, приведенная на рис. 8. В этом случае для выборки каждого из узлов используется линия ввода/вывода другого узла, работающая только на вывод, а не на ввод/вывод, как в предыдущей структуре.

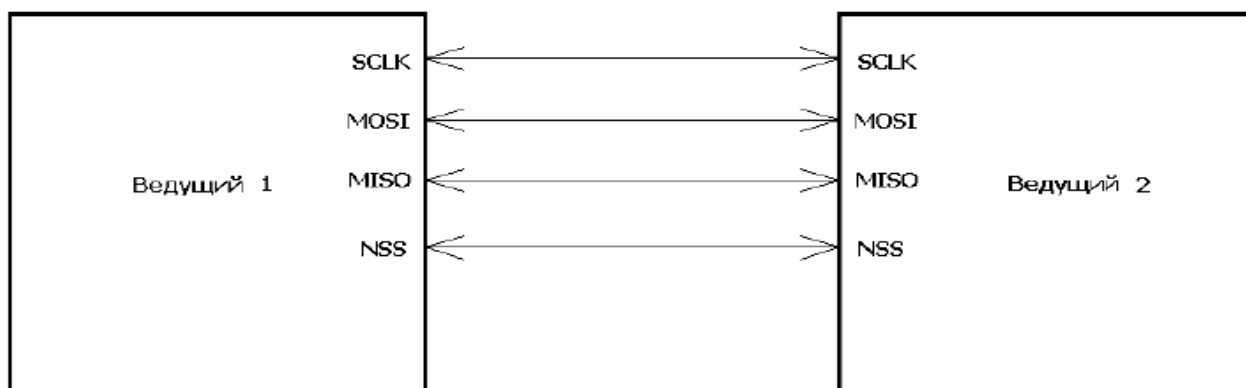


Рисунок 7. Архитектура четырехпроводной шины SPI с двумя ведущим

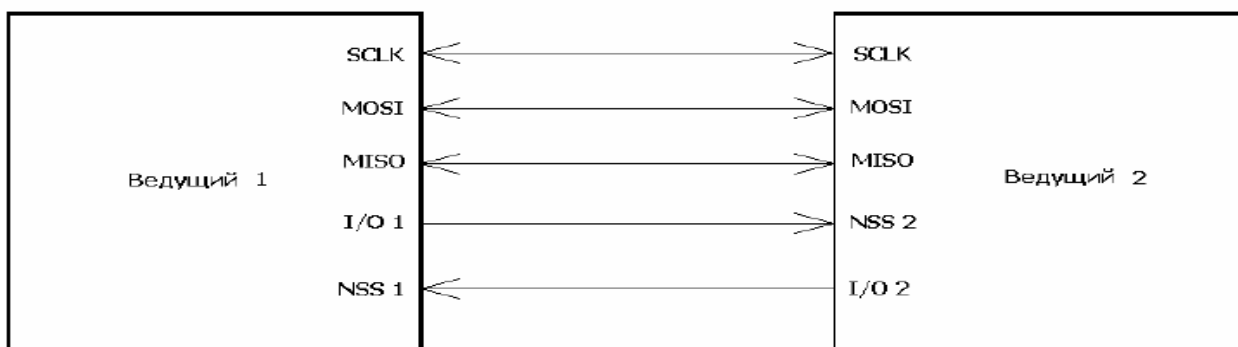


Рисунок 8. Архитектура четырехпроводной шины SPI с двумя ведущим и «перекрестной» выборкой

Как уже упоминалось выше, чисто двухпроцессорные архитектуры используются достаточно редко, например, в случае, если один из микроконтроллеров предназначен для ввода и обработки сигналов, а второй – для организации интерфейса с персональным компьютером или локальной сетью. В этом случае и один, и другой микроконтроллеры должны иметь возможность инициировать процедуру обмена данными, а, следовательно, должны иметь возможность функционирования в режиме ведущего. Более распространенный случай, это использование двухпроцессорной структуры с разделяемым ресурсом, например памятью. Пример такой архитектуры приведен на рис. 9.

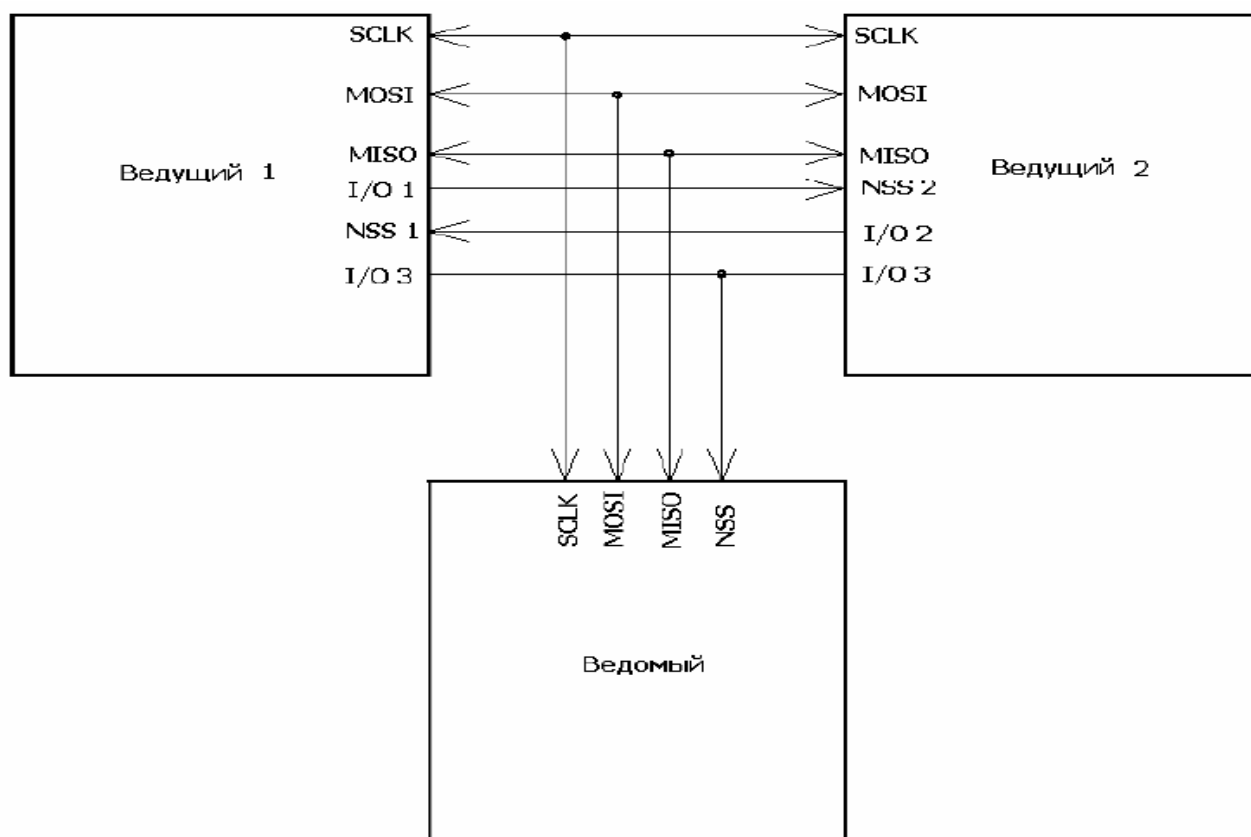


Рисунок 9. Двухпроцессорная архитектура с разделяемым ресурсом

В приведенной на рис. 9 архитектуре присутствуют два ведущих и один ведомый. Связи между ведущими минимизированы и соответствуют

архитектуре четырехпроводной SPI шины с двумя ведущими и «перекрестной» выборкой. Линии основных сигналов SPI шины подключены также к общему ведомому, а для выборки общего ведомого используется по одной дополнительной линии ввода/вывода от каждого из ведущих, объединенных по «монтажному ИЛИ». Архитектура вполне работоспособна, однако анализ занятости SPI шины и диагностики коллизий в ней еще более затруднен. Это объясняется необходимостью последовательного опроса как минимум двух линий выборки (индивидуальной для иницилирующего обмен контроллера и общего ресурса) перед занятием шины, а, следовательно, и увеличением вероятности возникновения коллизии. Для упрощения процедур обмена рекомендуется ввести еще две линии флагов запроса шины ID1 и ID2, как показано на рис. 10.

В этом случае при необходимости занятия шины контроллер выставляет свой флаг запроса шины, и дожидается ответного флага от второго контроллера. Второй контроллер, получив флаг запроса, либо сразу выставляет свой флаг, подтверждая, что шина свободна, либо заканчивает текущие операции обмена и только затем выставляет свой флаг. Такой механизм значительно упрощает диагностику занятости шины. [2]

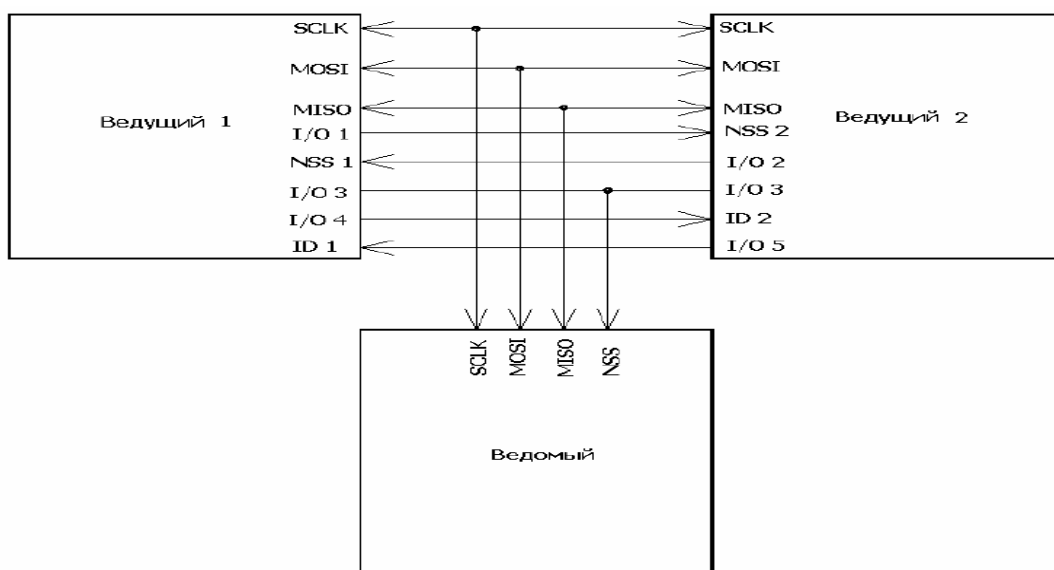


Рисунок 10. Двухпроцессорная архитектура с разделяемым ресурсом и линиями запроса шины

2.3. QSPI. Последовательный периферийный интерфейс с очередью

Последовательный периферийный интерфейс с очередью с точки зрения непосредственно обмена информацией между взаимодействующими интегральными схемами не имеет отличий от SPI. В данном случае лишь иным образом организуется работа центрального вычислителя с периферийным модулем.

QSPI предполагает использование очереди данных с программируемыми указателями, позволяющими осуществлять пересылку данных по интерфейсу без вмешательства вычислительного ядра. В результате, периферийные для ядра устройства выглядят для него, как расположенные в определенных местах ячейки памяти. QSPI предполагает также возможность применения других полезных программируемых опций, таких как автоматическая реализация сигналов «Chip Select» и задание длительности передачи и задержки между соседними словами.

В целом периферийные модули SPI в микроконтроллерах различных производителей функционируют по-разному. В частности, подобные очереди с прямым доступом в память распространены не широко, хотя и могут быть совмещены с контроллерами прямого доступа в память других интерфейсов. При этом большинство периферийных модулей SPI поддерживают до 4 сигналов «Chip Select», однако в некоторых микроконтроллерах данные сигналы интерфейса приходится реализовывать программно через порты ввода-вывода общего назначения. [3]

2.4. Microwire

Microwire (часто пишется «μWire») – это фактически ограниченный по функционалу интерфейс SPI. Наименование «Microwire» является зарегистрированной торговой маркой корпорации National Semiconductor. Интерфейс «MicrowirePLUS» также принадлежит National Semiconductor и функционально является эквивалентом QSPI. [4]

Microwire предполагает использование только полудуплексного режима передачи, MicrowirePLUS поддерживает и остальные режимы SPI.

Как правило, интегральные схемы с интерфейсом Microwire требуют меньшей частоты синхронизирующего сигнала, чем версии с SPI (в соотношении примерно 2 МГц к 20 МГц).

2.5. Альтернативные наименования интерфейса SPI

Название SPI является зарегистрированной торговой маркой корпорации Motorola. Поэтому в технических описаниях можно встретить различные названия и самого этого интерфейса и выводов интегральных схем, используемых для обмена информацией. Чаще всего встречается название Three-Wire Serial Interface – «трехпроводной последовательный интерфейс», что формально не совсем верно, т.к. SPI требует до 4 или даже, считая сигнал выбора кристалла, 5 выводов.

Вариантов названий SPI-выводов интегральных схем очень большое множество, потому что они сильно зависят от назначения устройства, и могут различаться в каждом отдельном случае. Так, например, в АЦП фирмы Analog Devices вывод SCK носит название SCLK, MISO называется DOUT (поскольку АЦП всегда – ведомый), а MOSI, соответственно, DIN. Другие обозначения могут встретиться в случае, например, энергонезависимой памяти производства Atmel, где MOSI будет просто SI, а MISO – просто SO. Название вывода /SS характерно лишь для аппаратной реализации SPI в микроконтроллерах AVR от Atmel. В других устройствах чаще всего имеется обычный вход выбора кристалла /CS, а иногда совместно с ним и другие управляющие выходы. Например, в памяти AT25 Atmel есть еще вывод задержки обмена /HOLD. Практически во всех микросхемах энергонезависимой памяти при этом есть вывод запрета записи /WR.

Также АЦП может быть оснащен выводом «готовности данных» с наименованием, например, DRDY. По этим причинам полный алгоритм обмена по интерфейсу SPI у различных устройств может различаться, хотя в

его основе всегда лежит последовательный побитный сдвиг в каждом такте частоты, задаваемой «мастером» на выводе SCK. Существуют также режимы, различающиеся по комбинации логических уровней.

Таким образом, разработчику встраиваемых систем при ознакомлении с технической документацией на ту или иную интегральную схему следует уделять внимание специфике обозначений конкретного производителя для составляющих уже, казалось бы, хорошо знакомого интерфейса.

3. I²C. ШИНА ИНФОРМАЦИОННОГО ОБМЕНА ИНТЕГРАЛЬНЫХ СХЕМ

Название шины I²C (или IC) является сокращением её англоязычного наименования «Inter-Integrated Circuit» – объединение интегральных схем. Среди наиболее распространенных среди разработчиков встраиваемых систем произношений данного названия на русском языке встречаются: «и-квадрат-си» и «и-два-цэ».

Интерфейс I²C разработан в начале 1990-х годов фирмой Philips для внутрисхемного низкоскоростного обмена между интегральными схемами: в основном между центральным вычислителем и внутренней периферией контроллера. Первая спецификация (версия 1.0, 1992 г.) исключала возможность задания адреса Slave-устройства программным способом и наряду со стандартным режимом скорости передачи данных 100 кбит/с (low-speed) был введен режим быстрой передачи (fast-speed) со скоростью до 400 кбит/с.

Версия 2.0, выпущенная в 1998 году, ввела в спецификацию шины быстродействующий режим (HS-mode) со скоростью передачи до 3,4 Мбит/с. Причем требования предписывали вводить обязательную возможность совместимости с режимами low-speed и fast-speed.

Наконец версия 2.1, датируемая 2000 годом, включает в себя незначительные модификации, не нашедшие отражения в версии 2.0.

Официальную документацию об интерфейсе I²C можно получить с Интернет-сервера фирмы Philips (в настоящее время переименована в NXP Semiconductors) и на других ресурсах сети Интернет.

1 октября 2006 года были отменены лицензионные отчисления за использование интерфейса I²C в пользу Philips, однако сохранились отчисления за выделение уникального адреса ведомого устройства на шине.

3.1. Архитектура систем с интерфейсом I²C

Интерфейс представляет собой реализацию двунаправленной полудуплексной мультиплексной шины с синхронной последовательной передачей данных и использует специальные состояния шины, реализующиеся через взаимное изменение состояния линий передачи данных и синхросигнала.

Шина I²C принадлежит к наиболее типичному классу шин с так называемой «Master-организацией». Такая организация подразумевает неравноправность обменивающихся информацией устройств (рис. 11): одно из устройств является ведущим на шине («Master»), а все остальные – ведомыми («Slave»). [5]

Гораздо реже в аппаратуре используется режим multi-master (рис. 12), когда к одной шине подключено несколько master-устройств.

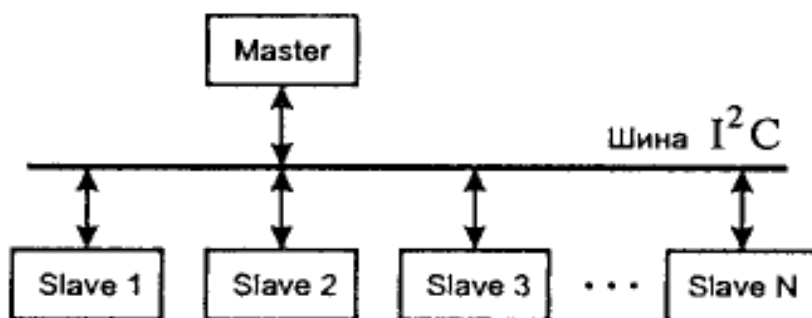


Рисунок 11. Организация шины с одним ведущим

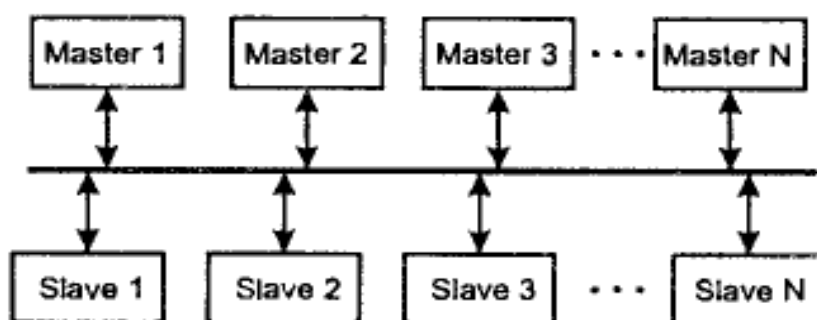


Рисунок 12. Организация шины с многими ведущими

Сложность такой организации шины состоит главным образом в том, что Master-устройства каким-то образом должны определять взаимный порядок работы со Slave-устройствами. В противном случае возникает ситуация, называемая шинным конфликтом, нарушающая работу встраиваемой системы в целом.

Для того чтобы исключить шинные конфликты в режиме multi-master должны содержаться процедуры арбитража и синхронизации, устанавливающие порядок работы Master-устройств. В данном учебном пособии не будем рассматривать этот режим подробнее, поскольку в абсолютном большинстве случаев в практике проектирования встраиваемых систем встречается классический вариант с одним Master-устройством.

3.2. Аппаратная реализация интерфейса

Интерфейс любого устройства, подключенного к шине I²C, состоит из двух транзисторов с открытым стоком (или коллектором для биполярной технологии) и двух буферных элементов с высоким входным сопротивлением (рис. 13). Один из выводов, названный SDA (Serial Data Line – линия последовательных данных), предназначается для связи с линией передачи информации. Второй вывод имеет сокращенное название SCL (Serial Clock Line – линия последовательной синхронизации) и предусмотрен для связи с линией синхронизации устройств.

Главным отличием Master-устройства от Slave-устройства является то, что только Master имеет право генерировать сигнал синхронизации на линии SCL, а Slave данную линию может только «слушать».

Каждая из линий может находиться в одном из двух состояний - «0» («низкий уровень») или «1» («высокий уровень»). При этом состояние шины, при котором обе линии имеют высокий уровень, считается свободным шинным состоянием. Шина в этот момент не занята и готова к трансляции данных (или, другими словами, находится в состоянии ожидания).

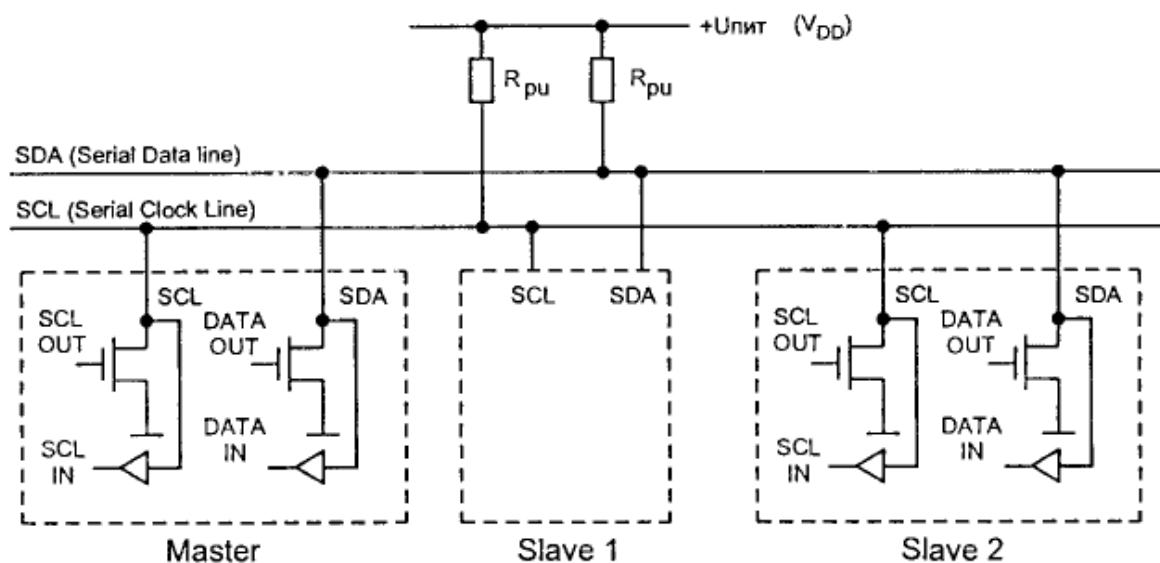


Рисунок 13. Реализация аппаратной части интерфейса

Поскольку транзисторы каждого из выводов интерфейса включены по схеме с открытым стоком (см. рис. 13 выше), то для реализации высокого уровня используется так называемая «подтяжка» к напряжению питания $U_{\text{пит}}$. Для этого используются «подтягивающие» резисторы R_{pu} (pull-up), величина электрического сопротивления которых колеблется в пределах от 1 до 10 кОм. Иногда «подтягивающие» резисторы присутствуют непосредственно на кристалле устройства Master. Следует отметить, что проектировщику при ознакомлении со спецификацией на интегральную схему с интерфейсом I²C следует обратить пристальное внимание на схемотехническую организацию соответствующих портов, поскольку непонимание их внутреннего устройства может приводить к конфликтам сигналов на шине.

Основные технические характеристики шины I²C по самой первой спецификации её разработчика приведены в табл. 1.

Таблица 1. Основные технические характеристики шины

| Наименование параметра | Значение параметра |
|--|---------------------|
| Скорость обмена low-speed | не более 100 кбит/с |
| Скорость обмена fast-speed | не более 400 кбит/с |
| Число адресуемых устройств (7 бит) | до 128 |
| Суммарная длина линий SCL и SDA | не более 4 м |
| Суммарная паразитная емкость относительно общего провода | не более 400 пФ |
| Входная емкость на каждый вывод абонента | не более 10 пФ |

Для протокола обмена по шине характерно то, что значение имеют не только статические состояния линий («0» и «1»), но и перепады уровней (из «0» в «1» и из «1» в «0»).

Для распознавания начала и конца передачи в спецификацию шины введены условия Start и Stop (см. рис. 14 и рис. 15). В фирменной документации условие Start имеет условное сокращение «S», условие Stop – «P». Разумеется, ответственным за создание условий Start и Stop является Master.

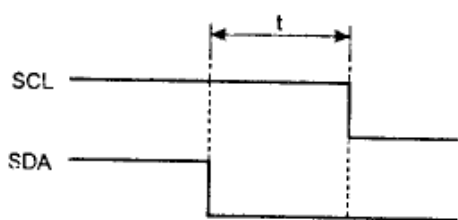


Рисунок 14. Условие Start

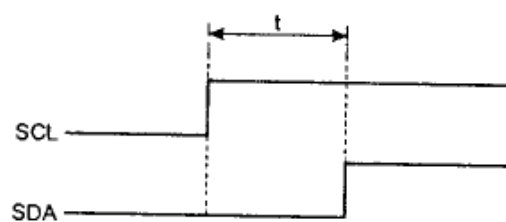


Рисунок 15. Условие Stop

Условие Start образуется при отрицательном перепаде сигнала на линии SDA при высоком уровне на линии SCL. И наоборот, условие Stop возникает при положительном перепаде линии SDA при высоком уровне на линии SCL. Таким образом, укрупненно информационный пакет, передаваемый по шине I²C, выглядит так, как показано на рис. 16.

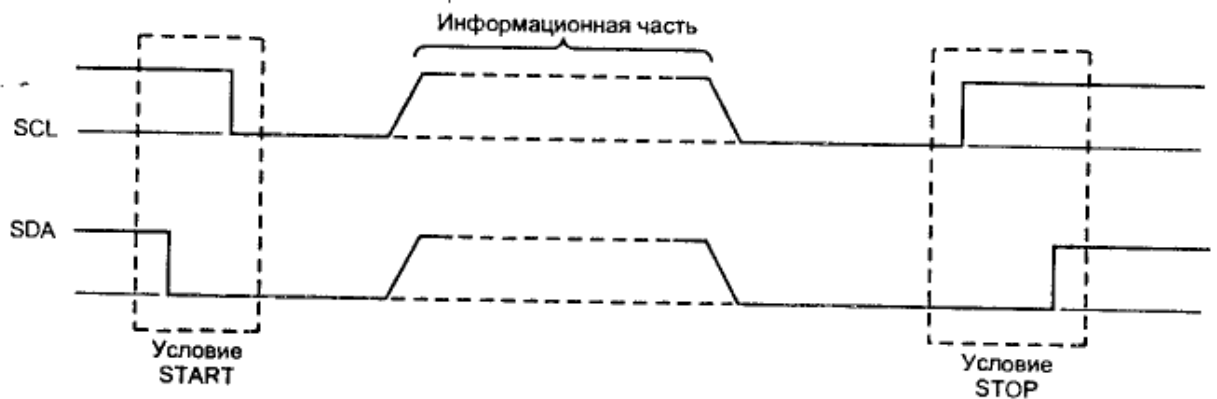


Рисунок 16. Информационный пакет данных на шине

Передача любого бита по шине происходит при условии стробирования данных SDA по линии SCL (рис. 17). Slave-устройство фиксирует бит, значение которого присутствует на SDA, по перепаду сигнала на SCL из низкого уровня в высокий (передний фронт).

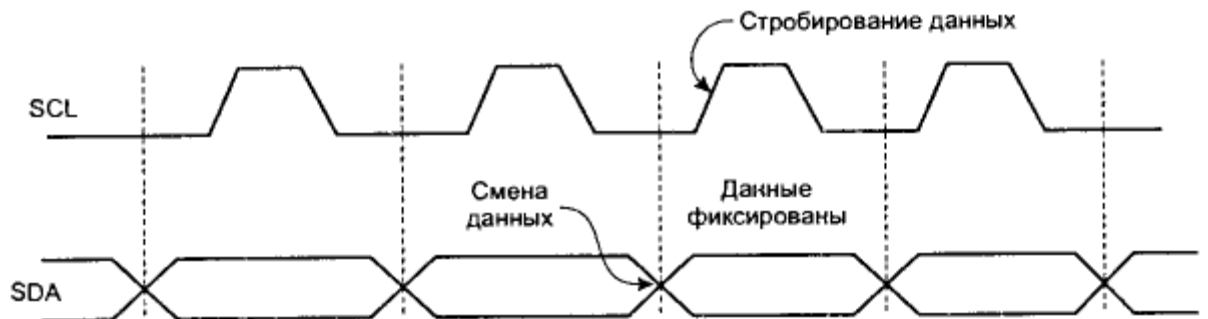


Рисунок 17. Передача бита по шине

Главное правило организации протокола шины I²C состоит в том, что смена информации на линии SDA может быть произведена только при нулевом состоянии линии SCL. В противном случае возникнет условие Start или условие Stop и передаваемый информационный пакет будет разрушен.

Передача данных по шине производится по 8 битов, после чего следует сигнал подтверждения (Acknowledge), генерируемый Slave-устройством. Сигнал подтверждения свидетельствует о том, что данные нормально

приняты и обработаны. При этом если байт, переданный по шине, является последним в пакете, Master-устройство вместо проверки сигнала подтверждения может выставить состояние Stop, и slave-устройство должно освободить линию SDA. В фирменной документации состояние acknowledge условно обозначается буквой «А».

Рис. 18 отражает процесс передачи байта по шине I²C. После отработки состояния Start передатчик последовательно выставляет на линии SDA данные, начиная со старшего бита и заканчивая младшим. Данные стробируются по линии SCL импульсами.

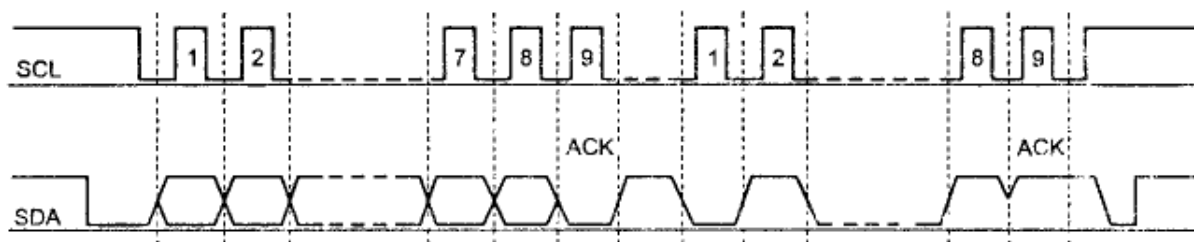


Рисунок 18. Передача байта по шине I²C

Линия SDA приемника (Slave-устройства) в момент приема информационных битов должна быть выставлена в единичное состояние. Физически это означает, что транзистор, подключенный к линии SDA, должен быть закрыт. В момент отрицательного перепада восьмого импульса на линии SCL Slave-устройство должно выставить на линию SDA нулевой уровень – открыть транзистор. Тем самым приемник подтверждает прием байта. На этот период Master выставляет на линию SDA единичное состояние, благодаря чему действительно присутствующий на ней уровень определяется только Slave-устройством. Передатчик должен проверить состояние линии SDA, затем выдать девятый стробирующий импульс, с которым Slave-абонент снимет сигнал подтверждения (выставит высокий уровень на линии SDA), проверит выполнение этой операции и лишь после этого продолжит передачу (рис. 19). В случае отсутствия подтверждения нормального приема (сигнал Acknowledge не появляется, т.е. Slave не

выставляет нулевой уровень на линии SDA) передатчику желательно выполнить условие Stop и повторить передачу.

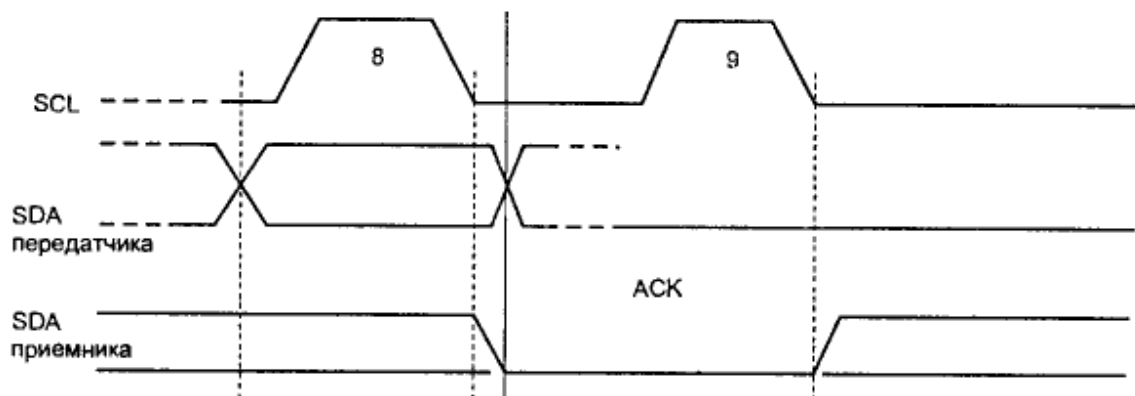


Рисунок 19. Сигнал Acknowledge

Следует отметить, что в случае проведения длительной обработки информации Slave-устройством оно может задержать освобождение линии SDA от сигнала подтверждения. Это не является ошибочным состоянием интерфейса, поэтому Master-устройству следует ожидать снятия сигнала подтверждения и лишь затем продолжать работу.

3.3. Форматы передачи данных

В силу того, что шина I²C предполагает присутствие множества Slave-устройств, то различение их при передаче информации требует применения механизма адресации.

В случае шины I²C адреса ведомых устройств, называемые также «slave-адресами», жестко определяются при проектировании микросхем, описываются в документации и не подлежат переназначению, т.е. модификации. Хотя зачастую у разработчика аппаратной части встраиваемой системы остается возможность переназначения двух-трех младших битов адреса путём установки соответствующих логических состояний на определенных выводах микросхемы. Таким образом решается задача расположения на шине нескольких микросхем одного и того же типа.

В начале передачи со стороны Master-устройства все Slave-абоненты «слушают» линию на предмет опознавания своего Slave-адреса. Опознавший свой адрес абонент продолжает прием данных и выдачу сигналов подтверждения, остальные только следят за моментом выдачи команды Stop.

Структура байта адресации представлена на рис. 20. Под Slave-адрес отводится 7 бит, а оставшийся (младший) бит байта адресации отведен под сигнал, указывающий на тип операции производимой Master-устройством: 0 – запись, 1 – чтение.

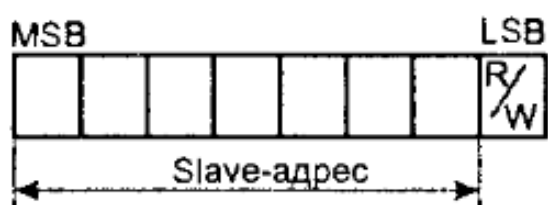


Рисунок 20. Структура байта адресации (управляющего слова)

Некоторые Slave-адреса используются не для прямой адресации устройств на шине I²C, а для служебных сообщений. В табл. 2 приведены краткие описания служебных адресов. Применение некоторых из них приведено далее по тексту.

Таблица 2. Служебные адреса

| Slave-адрес | R/W | Описание служебного адреса |
|-------------|-----|---|
| 0000 000 | 0 | Общий вызов |
| 0000 000 | 1 | Программный Start |
| 0000 001 | x | Адрес шины CBUS |
| 0000 010 | x | Зарезервирован для других форматов шины |
| 0000 011 | x | Зарезервирован на будущее |
| 0000 1xx | x | Код режима Hs-mode |
| 1111 1xx | x | Зарезервирован на будущее |
| 1111 0xx | x | 10-битная адресация slave-абонентов |

При использовании «традиционной» 7-битной адресации на шине определяется три формата передачи:

1. Master транслирует данные на Slave;
2. Master читает данные от Slave;
3. Комбинированный режим трансляции/чтения.

На рис. 21 приведен формат передачи данных от Master-устройства к Slave-абоненту. На данном рисунке и последующих в настоящем разделе, посвященном шине I²C, прямоугольники, отображенные серым тоном, относятся к направлению передачи от Master к Slave, а белым цветом – от Slave к Master.



Рисунок 21. Передача данных от Master к Slave

На рис. 22 приведен формат передачи данных от Slave-устройства к Master-устройству. Следует обратить внимание, что после чтения информационного байта DATA Master-устройство обязано подтвердить получение байта сигналом ACK.



Рисунок 22. Получение мастером данных от Slave

Наконец комбинированный формат передачи, применяющийся в случаях, когда осуществляется запись и чтение в одном цикле, приведен на рис. 23.

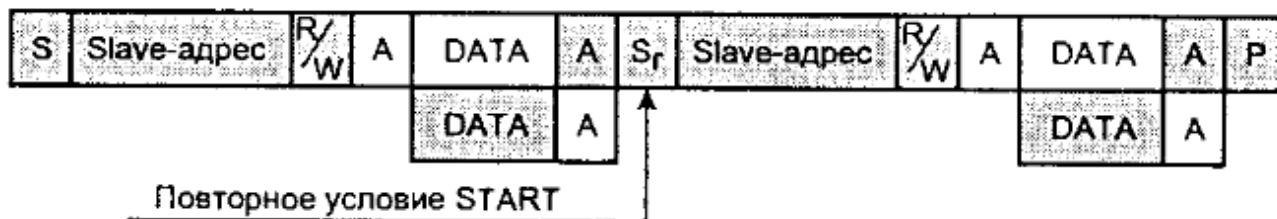


Рисунок 23. Комбинированный формат передачи данных

Обычно комбинированный формат применяется для работы с последовательной памятью (для сокращения времени доступа к данным). На рис. 23 видно, что здесь повторяются форматы трансляции данных к Slave-устройству и чтения данных от Slave-устройства. Единственное отличное обозначение «Sr» соответствует повторному состоянию Start («repeated Start»). Повторный Start по своей реализации на шине не отличается от обычного состояния Start, а обозначение «Sr» используется для лучшей читаемости рисунков с форматами передачи.

По прошествии относительно небольшого времени после начала применения I²C шина завоевала популярность. Следствием этого стало быстрое исчерпание доступных Slave-адресов, а также стремление производителей микросхем увеличить пропускную способность интерфейса. В результате, шину модернизировали, наделив её новыми возможностями:

1. Режим fast-speed с возможностью передачи до 400 Кбит/с;
2. Режим Hs-mode – до 3,4 Мбит/с;
3. Адресация с помощью 10 разрядов, а не 7, позволяющая увеличить количество абонентов до 1024.

Все устройства, доступные для использования в шинах с расширениями, естественно, совместимы и со старыми форматами передачи информации.

Увеличение пропускной способности шины заключается фактически в увеличении частоты синхронизирующих импульсов на линии SCL. Поэтому

для совместимости со старыми интерфейсами достаточно сохранить прежнюю скорость передачи – не более 100 Кбит/с.

Отличие I²C-абонентов, работающих в режиме fast-speed, состоит в наличии входных формирователей фронтов на основе триггеров Шмитта на линиях SDA и SCL. Обмен же информацией в режиме Hs-mode требует внесения изменений, в том числе, в формат передачи данных.

Hs-mode распространен незначительно, поэтому останавливаться на нем отдельно не целесообразно. Подробно с форматами передачи в этом режиме, принципами совместимости его с другими режимами можно ознакомиться в [5]. Значительно более существенный практический интерес представляет 10-разрядная адресация шины I²C.

Поскольку форматом шины предусмотрена передача информации разделенными объемами по 8 бит, то 10-разрядные адреса приходится транслировать 2 байтами. Первый байт имеет структуру 11110xx(R/W), в котором биты, обозначенные символом «х», являются старшими разрядами 10-разрядного Slave-адреса. При этом аналогично «традиционному» формату в составе первого байта должен быть передан бит R/W.

Принцип адресации в расширенном варианте сохраняется. Устройство, получив служебный код в первом байте и опознав возможность приема 10-разрядного адреса, подтверждает это и принимает второй байт. При совпадении принятого адреса с содержащимся внутри устройства собственным адресом выдается подтверждение и ведется прием и передача данных в обычном режиме до появления состояния Stop.

На рис. 24 показан формат передачи данных от Master-устройства к Slave-устройству при применении 10-битной адресации.



Рисунок 24. Передача данных от Master к Slave при 10-битной адресации

Slave-устройства, присутствующие на шине, сравнивают первый полученный байт со своими данными и, если они совпадают, выдают сигнал АСК (A1 на рис. 24). Очевидно, что в данном случае сразу несколько устройств могут подтвердить совпадение старшей части адреса, что не является конфликтной ситуацией со схемотехнической точки зрения. При получении же второго байта только одно устройство выдаст сигнал АСК (A2). Остальные, которые ранее выдали сигнал АСК (A1), останутся адресованными по первому байту до получения сигнала Stop.

На рис. 25 приведен формат передачи данных от Slave-устройства с 10-битным адресом к Master-устройству.

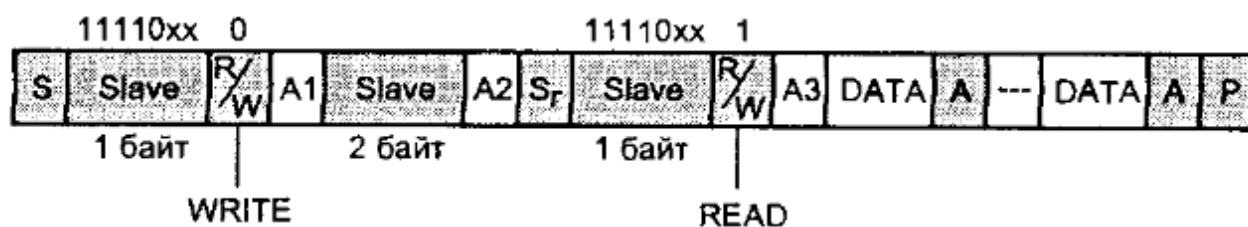


Рисунок 25. Передача данных от Slave к Master при использовании 10-битной адресации

Вначале Master-устройство первым байтом адресует все Slave-устройства соответствующим кодом, и они подтверждают его получение сигналом АСК (A1). Затем вторым байтом адресуется конкретное устройство с выдачей сигнала АСК (A2). После выполнения повторного условия Start (Sr) адресованное Slave-устройство сохраняет возможность обращаться к нему, поэтому достаточно повторить первый байт адреса, но уже с другим значением бита R/W и получить АСК (A3).

3.4. Некоторые практические применения интерфейса I²C

Наибольшее число применений интерфейса I²C приходится на интегральные схемы самого разнообразного назначения, требующие управления со стороны некоторого вычислителя с целью задания и

мониторинга режимов и параметров их работы. В силу своей простоты и ограниченной пропускной способности такое применение шины I²C является наиболее очевидным.

Как правило, данная задача выполняется путём считывания и заполнения определенными данными специализированных регистров, расположенных на кристаллах данных интегральных схем. И, таким образом, с точки зрения информационной модели подобной встраиваемой системы, Master-устройством является центральный микропроцессор или микроконтроллер, а Slave-устройствами – встроенные модули регистровой памяти специализированных интегральных схем. Благодаря возможности уникальной адресации на шине, центральный вычислитель способен выборочно обращаться к тому или иному абоненту для считывания и модификации содержимого его регистров (см. рис. 26).

Так же организуется обмен с широко применяемыми интегральными схемами энергонезависимой памяти относительно небольшого объема (EEPROM), которые чаще всего применяются для хранения параметров настройки встраиваемой системы, конфигурации микросхем программируемой логики, кода программно-управляемых микросхем и т.д.

При обмене информацией в данном случае первыми в последовательности, байтов, следующих за Slave-адресом, является адрес внутреннего регистра интегральной схемы, а затем значение адресуемого регистра, прочитанное со Slave-устройства или записываемое Master-устройством в Slave-устройство (см. рис. 27). В большинстве выпускаемых в настоящее время специализированных интегральных схемах, использующих для хранения настроек регистры, доступные по шине I²C, длина адреса внутреннего регистра и разрядность самих данных составляют 8 бит.

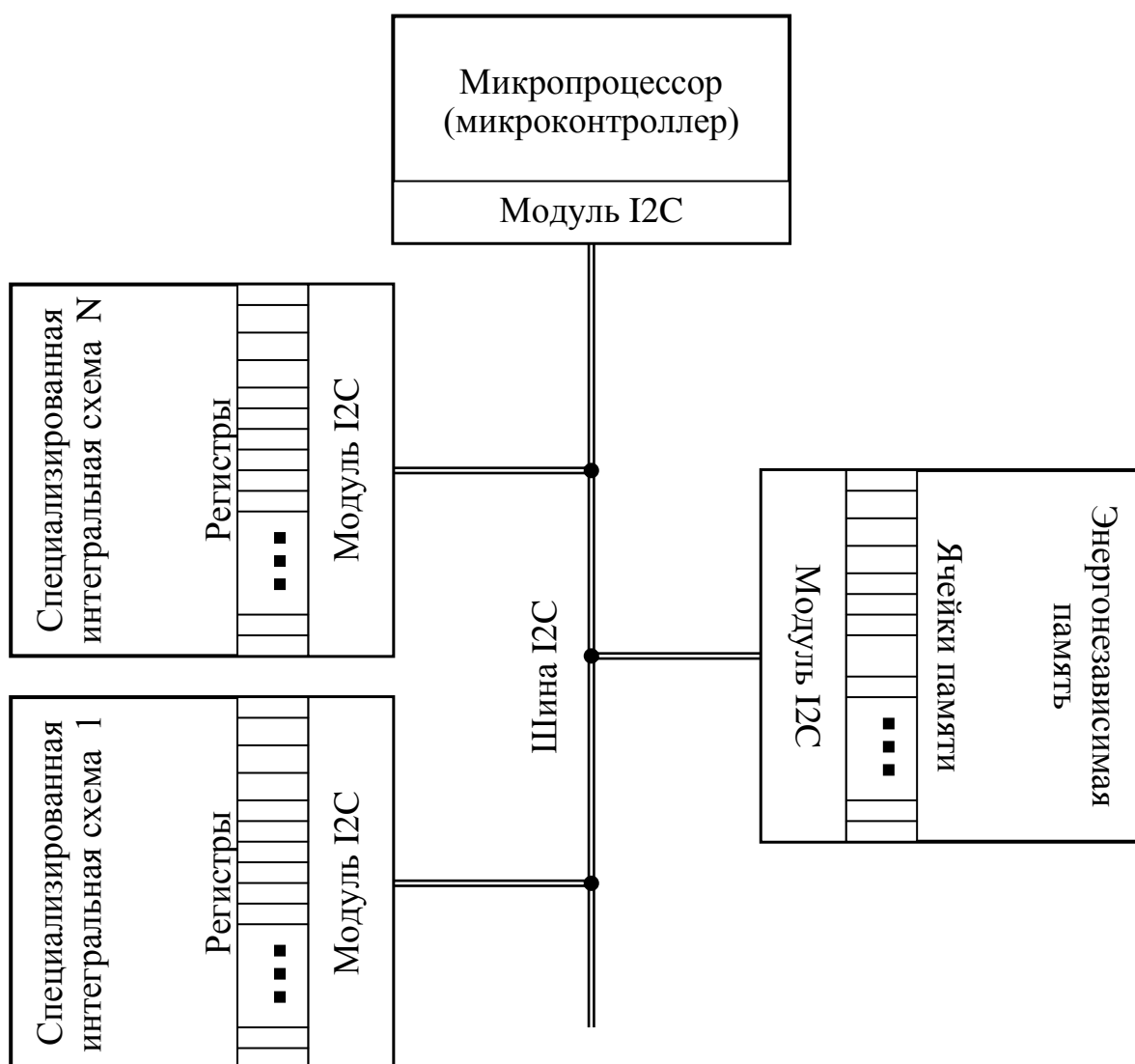


Рисунок 26. Иллюстрация к пояснению информационной модели встраиваемой системы, использующей внутриплатный интерфейс I²C

| | | | | | | | | |
|---|-------|--------|---|-------------------------|---|--------------------------------------|---|---|
| S | Slave | R W | A | Адрес ячейки (регистра) | A | Данные для записи в ячейку (регистр) | A | P |
|---|-------|--------|---|-------------------------|---|--------------------------------------|---|---|

Рисунок 27. Формат информационной посылки по интерфейсу I²C при записи в конфигурационный регистр специализированной интегральной схемы

Разработчику встраиваемой системы достаточно обратиться к спецификации на выбранную интегральную схему, чтобы получить подробное описание каждого из регистров, состояние регистров «по

умолчанию» и получить примеры заполнения регистровой памяти для выполнения микросхемой тех или иных функций. Подобный раздел в спецификации часто носит наименование «Внутренние управляющие регистры» («Internal Control Registers»).

4. SMBus. ШИНА СИСТЕМНОГО УПРАВЛЕНИЯ

Шина SMBus (System Management Bus — шина системного управления) – двухпроводной интерфейс для обмена данными между микросхемами различных системных компонентов персонального компьютера или встраиваемой системы. Основное исходное назначение интерфейса – управление подсистемой питания и сопутствующими подсистемами.

Первоначально шина разрабатывалась для интеллектуальных аккумуляторных батарей и зарядных устройств, и первые версии спецификации SMBus шли под заголовком «Smart Battery System Specifications» (спецификации системы «умных» батарей). Система с «умными» батареями (Smart Battery System) состоит из собственно батарей (аккумуляторов) и зарядных устройств, способных обмениваться управляющей информацией между собой и с центральным вычислителем, которого она и питает. Этот обмен позволяет батареям сообщать свои параметры (текущие значения, ожидаемые величины), менять режимы работы (питания вычислителя) или зарядки. Часть управляющих функций выполняется при участии центрального вычислителя, а часть — автономно.

Первая версия спецификации SMBus вышла в 1995 г., версия 1.1 — в 1998 г. Наконец версия 2.0, вышедшая в 2000 г. уже не имела заголовка, указывающего на специализацию в отношении источников питания, а получила название «Шина системного управления» («System Management Bus (SMBus)»). Тем не менее, организацией, выпустившей спецификацию на версию 2.0, является форум разработчиков систем с интеллектуальными батареями («Smart Battery Systems (SBS)»), в который входит большое число производителей источников питания, а также фирма Intel. Спецификация покрывает три нижних уровня модели взаимодействия открытых систем (OSI), от физического до сетевого.

Шина SMBus основана на интерфейсе I²C, и к ней применимы все описания, приведенные в соответствующем разделе учебного пособия. В шину введена возможность динамического реконфигурирования (подключения и отключения абонентов шины в процессе работы) и автоматического назначения адресов устройств. По сравнению с I²C в шине несколько изменены граничные требования к уровням сигналов и временным диаграммам, но в основном они совпадают и поэтому не требуют детального изложения. SMBus позволяет реализовать все режимы обменов, разрешенные для I²C, но имеет нюансы в правилах генерации подтверждений.

На системных («материнских») платах современных персональных компьютеров для шины SMBus в BIOS имеется стандартизованная поддержка. Отличительными особенностями её применения в данном случае является исполнение главной роли в управлении системой питания и способность автономной работы – соединяемые устройства могут обмениваться информацией, даже когда питание на центральном процессоре (и других подсистемах) отсутствует.

Конкретные задачи, для выполнения которых применяется SMBus в современных персональных компьютерах, следующие:

- определение объема памяти и ее конфигурирование;
- предоставление информации об изготовителе;
- предоставление номера модели и каталожного номера;
- предоставление сообщений о различных ошибках;
- отключение тактовой частоты на свободных разъемах памяти;
- определение пониженного напряжения аккумуляторной батареи и другие.

4.1. Физический уровень спецификации SMBus

На физическом уровне (1-м уровне OSI) спецификация определяет электрические и временные параметры сигналов. По уровням сигналов (и

нагрузочной способности) имеются две различные спецификации. Маломощная (low power) спецификация соответствует «родному» применению SMBus в устройствах с батарейным питанием; здесь характерны малые токи. Мощная (high power) спецификация предназначена для работы абонентов SMBus в окружении источников значительных помех (например, на плате PCI). Маломощные и мощные устройства на одной шине вместе работать в общем случае не смогут, но это и не требуется. При необходимости совместного использования устройств разных классов организуются разные сегменты шины, соединенные мостом.

В спецификациях временных параметров приняты меры по ограничению времени отклика и предотвращению «зависания» шины. Частота тактового сигнала ограничена и снизу (10 кГц), и сверху (100 кГц); введены ограничения на максимальную длительность нахождения синхросигнала в высоком и низком состоянии и максимальную суммарную «растяжку» битовых интервалов на каждый байт. Предусматривается механизм тайм-аутов, по которому устройства, обнаружившие «зависание» шины, должны немедленно прекратить обмен и повторно инициализироваться. В спецификации I²C подобные варианты сбоя не рассматривались.

На шине SMBus могут использоваться дополнительные аппаратные сигналы:

- Сигнал SMBSUS* указывает на приостановку шины и устройств. Этот сигнал вырабатывается устройством управления питанием; во время его активности (низкий уровень) сигналы SDA и SCL штатным образом (как в I²C) не воспринимаются, но могут использоваться для уточнения режима приостановки.
- Сигнал SMBALERT# служит для оповещения ведущего устройства о необходимости обмена с ведомым устройством, не имеющим возможности выступить в роли ведущего устройства. Этот сигнал собирается по схеме «Монтажное И» от всех

устройств. Получив его, хост должен дать команду чтения байта по адресу 0001100, на которое просигналившее устройство должно ответить байтом со своим адресом.

4.2. Уровень связи данных спецификации SMBus

На уровне связи данных (2-м уровне OSI) определяются те же правила передачи данных, что и в I²C: условия S, P, Sr; биты подтверждения; 7-битная адресация и признак R/W в первом байте. Как и в I²C, ведущее устройство может выполнять и запись, и чтение данных ведомого устройства; используются и комбинированные транзакции (через условие Sr). В генерации подтверждений для интерфейса SMBus есть особенности. На собственный адрес устройство всегда должно отвечать битом подтверждения ACK, даже если оно занято внутренними операциями. Это правило обеспечивает работу механизма определения присутствия данного устройства на шине. Ведомое устройство может вводить бит NACK (No Acknowledge) – т.е. отсутствие подтверждения – в ответ на любой неадресный байт, если оно занято или запрашиваемые командой данные недоступны. В этом случае у него есть и альтернатива поведения — задержать синхросигнал на низком уровне (в разрешенных пределах). Ведомое устройство должно вводить бит NACK в ответ на недопустимые коды команд или данных. Ответ NACK вынуждает ведущее устройство прекратить транзакцию (освободить шину). Ведущее устройство, будучи приемником, ответом NACK информирует передатчик о приеме последнего ожидаемого байта.

4.3. Сетевой уровень спецификации SMBus

Сетевой уровень (3-й уровень OSI) наибольшим образом определяет специфику шины SMBus и заслуживает более детального рассмотрения.

В шине SMBus введено понятия «хоста» (host) — абонента шины, выполняющего координирующие и конфигурирующие функции. Хост является ведущим устройством шины, при этом должен выполнять ряд функций ведомого устройства и обрабатывать сообщения уведомления.

4.3.1. Статическая адресация в SMBus и команды интерфейса

Каждое ведомое устройство имеет свой уникальный адрес; в диапазоне 7-битных значений адреса выделяются специальные значения (табл. 3), которых несколько больше, чем в I²C. 10-битная адресация в текущей версии не рассматривается. Адреса устройств разделяются по типам.

Для устройств однозначно понятного назначения SMBus выделяет специальные адреса (Purpose-Assigned Addresses – целевое назначение адреса). Например, батареи Smart Battery имеют адрес 0001011, их зарядные устройства – 0001001. Устройства с этими адресами обязаны соответствовать требованиям SMBus, предъявляемым к устройствам данного класса. Ряд систем с SMBus определяют и используют эти устройства, основываясь на их адресе. Другие системы могут и не доверять одному только адресу, а определять типы присутствующих устройств иным образом.

Для устройств разнообразного назначения, а также устройств, не полностью отвечающих спецификациям SMBus для своего класса, производители назначают иные адреса, с которыми можно ознакомиться на сайте www.smbus.org. Адреса устройств-прототипов задействуются исключительно в экспериментально-отладочных целях и в коммерческих изделиях использоваться не должны. В спецификации SMBus 2.0 появилась возможность автоматического динамического назначения адресов устройств, которая будет рассмотрена ниже.

Упомянутая в табл. 3 шина ACCESS.bus является родственным SMBus интерфейсом, разработанным в 1995 году для подключения к персональным компьютерам типовой периферии. ACCESS.bus предполагал соединение клавиатуры, монитора, «мыши» и т.д. «по цепочке», что позволило бы

сократить количество кабелей и используемых портов компьютера. Однако ACCESS.bus очень быстро был вытеснен универсальным для применения и гораздо более производительным интерфейсом USB.

Таблица 3. Специальные адреса SMBus

| Биты [7:1] | Бит 0 (RW) | Назначение |
|------------|------------|--|
| 0000000 | 0 | General call address – адрес общего вызова |
| 0000000 | 1 | Start – начало активного обмена |
| 0000001 | X | Адрес устройства шины CBUS (для совместимости) |
| 0000010 | X | Адрес для устройств иных шин |
| 0000011 | X | Зарезервировано |
| 00001XX | X | Зарезервировано |
| 0101000 | X | Хост шины ACCESS.bus |
| 0110111 | X | «Дежурный» адрес ACCESS.bus |
| 11110XX | X | Признак 10-битной адресации |
| 11111XX | X | Зарезервировано |
| 0001000 | X | Хост шины ACCESS.bus |
| 0001100 | X | Адрес ответа на сигнальные сообщения SMBus |
| 1100001 | X | «Дежурный» адрес SMbus |
| 11000XX | X | Адрес устройств-прототипов |

Типичное устройство SMBus поддерживает определенный набор команд, с помощью которых выполняется обмен данными. Команда кодируется одним байтом, передаваемым в транзакции вслед за адресным байтом. Команды могут использовать один из 11 протоколов, определенных в SMBus.

В версии 1.1 спецификации SMBus введена возможность контроля ошибок пакета PEC (Packet error checking – проверка ошибок в пакете).

Механизм PEC основан на добавлении в конец каждого передаваемого пакета байта CRC-кода (Cyclic Redundancy Checking – проверка циклическим избыточным кодом), вычисляемого по всем предыдущим байтам пакета, начиная с адресного. Почти все протоколы могут иметь два варианта — без PEC и с PEC; на одной шине могут присутствовать устройства и с поддержкой PEC, и без. На байт PEC приемник отвечает подтверждением, но трактовка ответа неоднозначна. Если передатчик в ответ на PEC получил ответ NACK, это означает, что приемник не подтвердил корректный прием пакета. Однако положительный ответ ACK не является подтверждением достоверности приема: приемник может «не понимать» PEC и отвечать на него как на обычный байт данных; приемник также может не выполнять контроль в реальном времени приема потока данных. Более достоверный контроль могут обеспечить лишь протоколы высших уровней. Так, например, для контроля достоверности записи в устройство можно использовать последующее чтение тех же данных с PEC, и по анализу всего принятого пакета ведущее устройство сделает вывод об успешности или ошибке операции записи.

Как было сказано выше шинные протоколы SMBus основаны на транзакциях I²C с 7-битной адресацией. Передаваемые команды в составе данных пакета I²C состоят из следующего множества:

- Quick Command, короткая команда, – посылка адресного байта; действие команды определяется битом R/W адресного байта;
- Send Byte, посылка байта, – передача ведущим устройством вслед за адресным байтом (при R/W=0) одного байта данных. В варианте с PEC передаются два байта, последний – PEC;
- Receive Byte, прием байта, – прием ведущим устройством вслед за адресным байтом (R/W=1) одного байта данных. В варианте с PEC принимаются два байта, последний – PEC;
- Write Byte, Write Word, запись байта/слова, – передача ведущим устройством вслед за адресным байтом (R/W=0) одного байта

команды, за которым следует 1 или 2 байта (младший, а затем старший) данных. В варианте с PEC в конец добавляется контрольный байт;

- Read Byte, Read Word, чтение байта/слова, – комбинированные транзакции: сначала посылается адресный байт ($R/W=0$), за которым передается код команды. Далее, через команду Start посылается адресный байт с тем же адресом устройства, но $R/W=1$, после которого принимается 1 или 2 байта данных. В варианте с PEC в конце ожидается прием дополнительного (контрольного) байта;
- Block Write, запись блока, – передача ведущим устройством вслед за адресным байтом ($R/W=0$) одного байта команды, за которым следует байт-указатель длины (количество последующих байт) и собственно байты данных. В варианте с PEC в конец добавляется контрольный байт. Указатель длины не учитывает байт PEC; он не может быть нулевым; одной блочной командой можно пересылать до 32 байт данных;
- Block Read, чтение блока, – комбинированная транзакция: сначала посылается адресный байт ($R/W=0$), за которым передается код команды. Далее, через команду Start посылается адресный байт с тем же адресом устройства, но $R/W=1$, после которого принимается байт-указатель длины, а за ним и собственно байты данных. В варианте с PEC в конце ожидается прием дополнительного (контрольного) байта. Правила применения поля указателя длины – такие же, как для записи блока;
- Process Call, вызов процесса, – комбинация команды Write Word с приемом слова (двух байт) от устройства с тем же адресом. Команда называется вызовом процесса, поскольку ожидает данных, зависящих от посланного кода команды и слова данных.

В варианте с PEC контрольный байт ожидается только в самом конце, вслед за последним байтом принятых данных;

- Block Write-Block Read Process Call – комбинация записи блока с последующим чтением блока по тому же адресу устройства. В варианте с PEC контрольный байт ожидается только в самом конце, вслед за последним байтом принятых данных.

В случае, когда ведущим устройством шины собирается выступать обычное устройство (не хост), оно должно использовать протокол уведомления хоста (SMBus host notify protocol): устройство на адрес хоста с R/W=0 (он известен) посылает байт с собственным адресом, за которым следует слово (два байта) собственно уведомления. Хост обязан понимать эти уведомления. Дополнительно может использоваться и необязательный сигнал внимания SMBALERT*.

4.3.2. Автоматическое назначение адресов в SMBus

Динамическое реконфигурирование шины SMBus – возможность «горячего» подключения и отключения – основано на базовых принципах интерфейса I²C. Автоматическое назначение адресов, появившееся в версии 2.0, использует еще и контрольные байты пакетов (PEC). Задача динамического реконфигурирования включает распознавание фактов подключения и отключения устройств и обеспечение бесконфликтного распределения их адресов. Подключение новых устройств может распознаваться двумя способами. Устройство, которое может работать ведущим устройством, при подключении (после своей инициализации по включению питания) может послать хосту уведомление, содержащее его адрес. Другой вариант обнаружения – периодический опрос устройств ведущим устройством для проведения «переучета» всех устройств на шине.

Для динамического бесконфликтного назначения личных адресов устройств используется протокол ARP (Address Resolution Protocol – протокол назначения адресов). Назначение адресов основано на стандартном

механизме арбитража (разрешения конфликтов) шины SMBus (и I²C). Назначенный адрес запоминается устройством на все время, пока подано питание. Существуют также устройства PSA (Persistent Slave Address – постоянный адрес ведомого), хранящие назначенный адрес в энергонезависимой памяти для его восстановления после повторного включения питания. После назначения адреса обмен с устройством выполняется точно так же, как и с устройством с фиксированным адресом. Назначение адресов может выполнять любое ведущее устройство шины SMBus.

Для динамического назначения адреса требуется изоляция устройств — возможность диалога ведущего устройства-нумератора с каждым устройством без помех со стороны других устройств (типичная задача настройки системы по технологии Plug-and-Play). Изоляция основана на уникальном идентификаторе устройства UDID (Unique Device Identifier) – 128-битной структуре, содержащей описание возможностей, версию, идентификаторы производителя, устройства, подсистемы и специфическую информацию. Идентификатор начинается с байта возможностей (Device Capabilities), в котором два старших бита характеризуют способности динамической адресации, а младший бит — поддержку PEC. Чтение идентификатора выполняется ведущим устройством ARP по протоколу блочного чтения по «дежурному» адресу SMBus. На это чтение отзываются все устройства с еще не назначенными адресами, и на арбитраже этой операции работает изоляция устройств. Первый считанный байт (указатель длины) у всех устройств одинаков, поэтому по нему конфликта быть не может. Далее устройства передают идентификаторы, и в арбитраже будут иметь приоритет те устройства, у которых нулевое значение бит данных встретится раньше. С учетом этого принята следующая кодировка классов устройств в старших битах первого байта идентификатора:

- 00 – устройства с фиксированными адресами, идентифицируются в первую очередь;

- 01 – динамические устойчивые (persistent) адреса устройств PSA;
- 10 – динамические изменчивые (volatile) адреса;
- 11 – устройства со случайными номерами, идентифицируются в последнюю очередь.

Младший бит этого байта указывает на поддержку PEC и всех возможностей, основанных на идентификаторе. При нулевом значении этого бита о поддержке PEC ничего определенного сказать нельзя.

Последний элемент UDID – 32-битный идентификатор устройства, играющий важную роль в распознавании похожих устройств. Это может быть либо число, назначаемое изготовителем (отвечающим за его неповторимость), либо случайное число, генерируемое (и запоминаемое) устройством каждый раз при включении или выполнении команды сброса.

Устройство, поддерживающее ARP, должно иметь специальные флаги:

- AR (Address Resolved) – данному устройству процедурой ARP назначен адрес;
- AV (Address Valid) – устройство имеет действительный личный адрес, на который оно отзывается в обычных транзакциях (при AV=0 устройство должно игнорировать все адреса, кроме «дежурного»).

При AV=AR=0 устройство не имеет личного адреса и должно участвовать в процессе ARP (отвечать на общий опрос идентификатора). При AV=1 и AR=0 устройство имеет личный адрес, но должно участвовать в ARP. При AV=AR=1 устройство имеет личный адрес, но не должно отвечать на общий запрос идентификатора. При этом оно должно обрабатывать адресованную ему команду назначения адреса (и впоследствии пользоваться новым назначенным адресом). Комбинация AV=0 и AR=1 недопустима.

Для протокола ARP введены специальные команды:

- Get UDID (general) – общий запрос идентификатора – команда чтения блока данных с адресом 1100001 и кодом команды 3. На нее устройства, поддерживающие ARP, отвечают посылкой

блока с длиной 17 байт, сопровождаемого PEC. Первые 16 байт блока – UDID, 17-й байт – адрес устройства (с единицей в младшем бите). Если у устройства флаг AV=0, оно вместо адреса передает код «111111». Команда не влияет на флаги AV и AR.

- Assign address – назначить адрес – команда записи блока данных с адресом 1100001 и кодом команды 4 и длиной 17 байт, сопровождаемого PEC. Первые 16 байт блока – UDID, 17-й байт – назначаемый адрес устройства (младший бит игнорируется). По этой команде устройство, обнаружившее полное совпадение UDID со своим собственным, устанавливает флаги AV=AR=1.
- Get UDID (directed) – направленный запрос идентификатора – команда чтения блока данных с адресом 1100001, в поле кода команды находится адрес интересующего устройства с единицей в младшем бите. На нее отвечает только устройство, опознавшее свой адрес в поле команды, и отвечает посылкой блока с длиной 17 байт, сопровождаемого PEC. Первые 16 байт блока – UDID, 17-й байт – адрес устройства (с единицей в младшем бите). Команда не влияет на флаги AV и AR.
- Reset device (general) – общий сброс устройства – посылка по адресу 1100001 байта команды с кодом 2, сопровождаемого байтом PEC. По этой команде все устройства инициализируются и обнуляют флаги AR и AV (PSA-устройства флага AV не изменяют).
- Reset device ARP (directed) – направленный сброс устройства – посылка по адресу 1100001 байта с адресом целевого устройства (с нулем в младшем бите), сопровождаемого байтом PEC. По этой команде выбранное устройство инициализируется и обнуляет флаги AR и AV (PSA-устройства флага AV не изменяют).

- Notify ARP master – уведомление ведущего устройства ARP – посылка на адрес 0001000 байта с «дежурным» адресом 11000010, за которым следуют два байта нулей. Устройство может посылать это уведомление о необходимости выполнения ARP при включении питания, а также при обнаружении коллизии в процессе выполнении чтения данных по индивидуальному адресу устройства.
- На каждый байт команд ARP-устройства, поддерживающие этот протокол, отвечают подтверждением АСК. Отсутствие подтверждений трактуется ведущим ARP-устройством как отсутствие ARP-устройств на шине.

«Переучет» и назначение адресов вкратце выглядит следующим образом: ведущее ARP-устройство выполняет команду общего запроса идентификатора и по ней получает UDID и, возможно, адрес первого «победителя» в арбитраже. Далее этому победителю назначается личный адрес – тот же или по усмотрению ведущего ARP-устройства, после чего он уже не участвует в арбитраже по общему запросу. Сведения об устройстве и его адрес ведущим ARP-устройством заносятся в таблицу устройств. После этого общий запрос и назначение адреса повторяются снова и снова, пока все ARP-устройства не будут удовлетворены. Признаком окончания ARP-процедуры для ведущего ARP-устройства будет отсутствие подтверждений приема команд общего запроса.

4.4. Поддержка SMBus в BIOS ПК и ACPI

Шина SMBus, в отличие от ACCESS.bus, сразу получила спецификацию ее поддержки на уровне BIOS персональных компьютеров. Позже появились спецификации, позволяющие интегрировать устройства SMBus в систему ACPI (Advanced Configuration and Power Interface — усовершенствованный интерфейс конфигурации и управления питанием).

В 1995 г. была опубликована спецификация интерфейса SMBus BIOS – System Management Bus BIOS Interface Specification, основные идеи которой вкратце изложены далее.

Интерфейс SMBus BIOS позволяет верхним уровням ПО абстрагироваться от аппаратной реализации хост-контроллера. Поддержка BIOS обеспечивается для трех типов режимов работы процессора: реального, защищенного 16-разрядного и защищенного 32-разрядного. Вызов функций может выполняться либо через BIOS-прерывание 15h (в реальном режиме), либо через точку входа, полученную при подключении в соответствующем режиме. Для подключения (и отключения) также используется сервис BIOS-прерывания 15h, а после подключения доступ через прерывание 15h блокируется (до отключения). В защищенном режиме вызов интерфейсных функций возможен только через точку входа, полученную при подключении. Поддержка точки входа для реального режима необязательна.

Спецификация SMBus BIOS обеспечивает хост-центрическое обращение к абонентам шины: по инициативе вызывающей программы хост-контроллер посылает устройству команду, которая может предполагать и немедленный ответ устройства. Однако устройства могут посылать хосту сообщения по собственной инициативе, при этом они обязаны использовать протокол Write Word. Хост способен помещать принятые сообщения в небольшую очередь, из которой они могут программно извлекаться путем вызова функции 7 прерывания (программа должна периодически выполнять этот вызов для проверки наличия сообщений в очереди). В очереди каждое сообщение представлено байтом адреса источника и парой байт тела сообщения.

Шина SMBus тесно связана с оборудованием, управляющим питанием и участвующим в генерации запросов системного прерывания SMI (System Management Interrupt – прерывание системного управления, особое аппаратное прерывание процессора) и их обработки. Из-за этого в BIOS введен специальный механизм, позволяющий обнаружить обработку SMI во

время выполнения транзакций и в иное время. Это необходимо, поскольку обработчик SMI, работающий в режиме SMM, совершенно невидим прикладной программе, а результаты его работы могут существенно влиять на работу программы, вызывающей BIOS SMBus.

Функции общих обращений к SMBus через BIOS прерывание 15h:

- SMBus Installation Check (01h) – проверка наличия функций;
- SMBus Real Mode Connect (02h) – подключение в реальном режиме;
- SMBus 16-Bit Connect (03h) – подключение в 16-битном защищенном режиме;
- SMBus 32-Bit Connect (04h) – подключение в 32-битном защищенном режиме;
- SMBus Disconnect (05h) – отключение от сервиса;
- SMBus Device Address (06h) – получение списка адресов устройств SMBus;
- SMBus Critical Messages (07h) – чтение сообщений устройств, переданных хосту.

Для взаимодействия с конкретными устройствами SMBus предназначен набор функций, позволяющих генерировать запросы протокольных команд SMBus и получать результаты их выполнения. Функции запросов и получения ответов разделены, что позволяет на время выполнения (и передачи) довольно длительных команд не занимать время центрального процессора. Большинство протокольных команд вводится за один вызов BIOS. Исключение составляет команда записи блока, данные для которой передаются за один или более последующих запросов продолжения. Результаты большинства команд также принимаются за один вызов, результат же чтения блока получается за несколько вызовов продолжения.

Вызовы протокольных команд устройств SMBus:

- SMBus Request (10h) – запрос команды устройству;
- SMBus Request Continuation (11h) – продолжение запроса для записи блока;

- SMBus Request Abort (12h) – отказ от выполнения ранее посланного запроса;
- SMBus Request Data and Status (13h) – запрос данных и состояния.

Шина SMBus является одним из главных коммуникационных средств в ACPI. Интерфейс SMBus для ACPI определен в спецификации SMBus CMI – документе System Management Bus (SMBus) Control Method Interface Specification Version 1.0, опубликованном в конце 1999 г. Этот интерфейс позволяет легко использовать все возможности SMBus независимо от происхождения или особенностей реализации оборудования. Основные цели составления спецификации следующие:

- Обеспечить эффективный и надежный интерфейс ACPI для аппаратных средств хост-контроллера независимо от его реализации (со встроенным микроконтроллером или без него);
- Обеспечить системную синхронизацию доступа ко всем ресурсам SMBus;
- Гарантировать соответствие интерфейса версиям 1.0 и 1.1 спецификации SMBus и возможности его расширения для поддержки новых свойств в будущих версиях.

Для успешного функционирования системы требуется поддержка соответствующими драйверами операционной системы. Для управления питанием форум разработчиков «интеллектуальных» батарей — Smart Battery System Implementers Forum (SBS-IF) — разработал спецификацию драйверов для ОС MS Windows, с которой можно ознакомиться на сайте <http://www.sbs-forum.org/smbus/>.

Рассмотрение программного интерфейса CMI выходит за рамки данного пособия. Следует лишь отметить, что коды протоколов SMBus, используемые в CMI, отличаются от кодов, используемых в SMBus BIOS. Те же протоколы, но с байтом PEC, кодируются с единицей в старшем бите (значение увеличено на 80h).

4.5. Сравнение шин I²C и SMBus

Как видно из вышеприведенных описаний, интерфейсы I²C и SMBus являются крайне близкими по их спецификации на первом, втором и, частично, третьем уровне семиуровневой модели OSI, хотя и имеют ряд электрических, конструктивных и протокольных отличий.

Электрический интерфейс рассматриваемых шин достаточно близок (табл. 4), и при напряжении питания на уровне 5В проблем совместимости не возникает (в таблице Vdd – напряжение питания). При других напряжениях питания сопрягаемых интерфейсами устройств проектировщик встраиваемой системы должен обратить внимание на взаимное соответствие уровней напряжения для логических сигналов данных устройств. [6]

Таблица 4. Электрические параметры шин I²C и SMBus

| Параметр | I ² C | SMBus мощная | SMBus маломощная |
|---|-------------------------------|-----------------|---------------------|
| Входной уровень лог.0, не более | 1,5В или 0,3Vdd | 0,8В | 0,8В |
| Входной уровень лог.1, не менее | 3,0В или 0,7Vdd | 2,1В | 2,1В |
| Выходной уровень лог.0, не более | 0,4В или 0,2Vdd при Vdd<2В | 0,4В | 0,4В |
| Выходной ток лог.0, мА | 3 | 4 | 0,35 |
| Максимальная емкость линии шины, пФ | 400 | 400 | 400 |
| Частота, КГц | 0-100,0-400,0- 3400 | 10-100 | 10-100 |

5. 1-Wire. ОДНОПРОВОДНЫЙ ИНТЕРФЕЙС

Однопроводной интерфейс 1-Wire, разработанный в конце 90-х годов фирмой Dallas Semiconductor Corp. (в настоящее время приобретена корпорацией Maxim Integrated Products), регламентирован разработчиками для применения в трех основных сферах-приложениях:

- приборы в специальных корпусах MicroCAN для решения задач идентификации, переноса или преобразования информации (технология iButton – хорошо известные так называемые «таблетки», используемые в системах доступа, домофонах и т.п.);
- внутрисхемное программирование энергонезависимой памяти встраиваемых систем или встроенной памяти специализированных интегральных схем;
- системы автоматизации (технология сетей 1-Wire-сетей).

Если первое применение широко известно на мировом рынке, и уже давно пользуется заслуженной популярностью, а второе с успехом обеспечивает возможность легкой перестройки функций полупроводниковых компонентов с малым количеством внешних выводов, производимых фирмой Dallas Semiconductor Corp., то системы автоматизации на базе 1-Wire-шины еще не получили должного признания. Ранее такая ситуация определялась крайне ограниченным набором компонентов для организации применений в области автоматизации. Однако в последнее время появляется все больше сообщений и конкретных примеров использования 1-Wire-интерфейса в самых различных областях, все больше разработчиков проявляют интерес к этой технологии, что связано, прежде всего, со значительным расширением номенклатуры однопроводных компонентов.

Однопроводные линии чаще всего предполагают использование в качестве физической среды передачи обычного телефонного кабеля и, следовательно, скорость обмена в этом случае не велика. Однако если внимательно проанализировать большинство объектов требующих

автоматизации, то более чем для 60% из них предельная скорость обслуживания в 15,4Кбит/с будет более чем удовлетворительной.

1-Wire имеет массу иных преимуществ, привлекающих внимание как разработчиков интегральных схем, так и разработчиков встраиваемых информационных систем. Среди таких преимуществ стоит назвать:

- простое и оригинальное решение адресуемости абонентов;
- несложный протокол;
- простая структура линии связи;
- малое потребление компонентов;
- легкое изменение конфигурации сети;
- значительная протяженность линий связи;
- исключительная дешевизна всей технологии в целом.

5.1. Основные принципы организации

1-Wire представляет собой информационную сеть, использующую для осуществления цифровой связи одну линию данных и один возвратный («общий» или «земляной») провод. Таким образом, для реализации среды обмена в этой сети могут быть применены доступные кабели, содержащие неэкранированную витую пару той или иной категории, и даже обычный телефонный провод. Такие кабели при их прокладке не требуют наличия какого-либо специального оборудования, а ограничение максимальной длины однопроводной линии регламентировано разработчиками на уровне 300м.

Основой архитектуры 1-Wire-сетей, является топология общей шины, когда каждое из устройств подключено непосредственно к единой магистрали, без каких-либо каскадных соединений или ветвлений. При этом в качестве базовой используется структура сети с одним ведущим (Master-устройством) и многочисленными ведомыми (Slave-устройствами). Хотя существует ряд специфических приемов организации работы однопроводных систем в режиме мультимастера, т.е. со многими ведущими.

Конфигурация любой 1-Wire-сети может произвольно меняться в процессе ее работы, не создавая помех дальнейшей эксплуатации и работоспособности всей системы в целом, если при этих изменениях соблюдаются основные принципы организации однопроводной шины. Эта возможность достигается благодаря присутствию в протоколе 1-Wire-интерфейса специальной команды поиска ведомых устройств («Поиск ПЗУ»), которая позволяет быстро определить новых участников информационного обмена. Стандартная скорость отработки такой команды составляет около 75 узлов сети в секунду.

Благодаря наличию в составе любого устройства, снабженного сетевой версией 1-Wire-интерфейса, уникального индивидуального адреса (отсутствие совпадения адресов для приборов, когда-либо выпускаемых Dallas Semiconductor Corp., гарантируется самой фирмой-производителем), такая сеть имеет практически неограниченное адресное пространство. При этом каждый из однопроводных приборов сразу готов к использованию в составе 1-Wire-сети, без каких-либо дополнительных аппаратно-программных модификаций. Однопроводные компоненты являются самотактируемыми полупроводниковыми устройствами, в основе обмена информацией между которыми, лежит управление изменением длительности временных интервалов импульсных сигналов в однопроводной среде и их измерение.

Передача сигналов, для 1-Wire-интерфейса, асинхронная и полудуплексная, а вся информация, циркулирующая в сети, воспринимается абонентами либо как команды, либо как данные. Команды сети генерируются мастером и обеспечивают различные варианты поиска и адресации ведомых устройств, определяют активность на линии даже без непосредственной адресации отдельных компонентов, управляют обменом данными в сети и т.д.

Стандартная скорость работы по интерфейсу 1-Wire-сети – 15,4Кбит/с – была выбрана, во-первых, с учетом обеспечения максимальной надежности

передачи данных на большие расстояния, и, во-вторых, с учетом быстродействия наиболее широко распространенных типов микроконтроллеров, которые в основном должны использоваться при реализации ведущих устройств однопроводной шины. Это значение скорости обмена может быть уменьшено до любого возможного значения благодаря введению принудительной задержки между передачей в линию отдельных битов данных (растягиванию временных слотов протокола). Или увеличено за счет перехода на специальный ускоренный режим обмена (скорость до 125Кбит/с), который допускается для отдельных типов однопроводных компонентов на небольшой по расстоянию, качественной, не перегруженной другими приборами линии связи.

При реализации однопроводного интерфейса используются стандартные КМОП/ТТЛ логические уровни сигналов, а питание большинства однопроводных компонентов может осуществляться от внешнего источника с рабочим напряжением в диапазоне от 2,8В до 6,0В. Альтернативой применению внешнего питания служит, так называемый, механизм «паразитного питания», действие которого заключается в использовании каждым из ведомых компонентов 1-Wire-линии электрической энергии импульсов, передаваемых по шине данных, которая аккумулируется специальной, встроенной в прибор емкостью. Кроме того, отдельные компоненты однопроводных сетей могут использовать режим питания по шине данных, когда энергия к приемнику поступает непосредственно от мастера по линии связи, при этом обмен информацией в сети принудительно прекращается.

Пожалуй, особенно привлекательным качеством технологии 1-Wire является исключительная простота настройки, отладки и обслуживания сети практически любой конфигурации, построенной по этому стандарту. Действительно, для начала работы достаточно любого персонального компьютера, недорогого адаптера 1-Wire-линии, а также свободно распространяемой фирмой Dallas Semiconductor Corp. программы iButton

Viewer. При наличии этого небольшого числа составляющих контроль и управление сетью практически любой сложности, построенной на базе стандартных однопроводных компонентов, организуется буквально в течение нескольких минут. Программа iButton Viewer, в этом случае, позволяет с максимальным комфортом для разработчика идентифицировать любое из ведомых однопроводных устройств на линии и проверить в полном объеме правильность его функционирования в составе конфигурируемой сети.

5.2. Организация ведущих устройств

Отдельные виды адаптеров, которые позволяют наделить любой персональный компьютер возможностью обслуживать в качестве мастера 1-Wire-сеть, выпускаются самой фирмой Dallas Semiconductor Corp. К ним относятся адаптеры для параллельного порта типа DS1410E, для COM-порта (последовательного порта) типа DS9097E и DS9097U, для USB-порта типа DS9490R. Эти приборы имеют различные функциональные возможности и конструктивные особенности, что обеспечивает разработчику максимальную свободу выбора при конструировании. А наличие у пользователя небольших навыков в создании электронной аппаратуры, позволяет легко произвести самостоятельную сборку схемы простейшего адаптера 1-Wire-сети для компьютера из небольшого числа доступных электронных компонентов.

Часто в качестве ведущего однопроводной шины выступает не компьютер или микропроцессорная система, а простейший универсальный микроконтроллер. Для организации его сопряжения с 1-Wire-сетью используются различные программно-аппаратные методы. От простейшего, когда управляющая программа контроллера полностью реализует протокол 1-Wire-интерфейса на одном из своих функциональных двунаправленных выводов, связанных с однопроводной линией, до вариантов, позволяющих высвободить значительные ресурсы контроллера, благодаря использованию

специализированных микросхем сопряжения с 1-Wire-сетью. Такие микросхемы подключаются к процессору, играющему роль ведущего однопроводной шины, через периферийные узлы ввода/вывода, входящие в состав любого универсального микроконтроллера. Например, устройство DS1481 предназначено для подключения непосредственно к функциональным выводам параллельного обмена контроллера.

Для организации Master-устройства однопроводной системы на базе микроконтроллеров с питанием напряжением 3В поставляются пассивные микросхемы DS1482, выполняющие согласование с уровнями сигналов стандартной 1-Wire-магистральной линии. Если же Master-устройство однопроводной линии должен быть организован на базе стандартного интегрированного модуля последовательного интерфейса микроконтроллера, используется микросхема DS2480B, а микросхема DS2490 адаптирует однопроводную линию для работы от встроенного узла UBS-интерфейса. Обе микросхемы реализуют так называемый программируемый механизм активной подтяжки шины данных 1-Wire-магистральной линии, обеспечивающий качественную передачу сигналов в длинных проблемных линиях и увеличение нагрузочной способности ведущего по количеству обслуживаемых им ведомых устройств.

Большинство упомянутых выше адаптеров для персональных компьютеров, также построены на базе подобных микросхем. Более того, учитывая особенности работы современных операционных сред Microsoft Windows, использование именно этих компонентов, которые по своей сути являются управляемыми по последовательному интерфейсу цифровыми автоматами, обеспечивает полномасштабное обслуживание однопроводных линий в реальном масштабе времени.

При построении сложных законченных микропроцессорных систем, имеющих дефицит машинного времени для реализации 1-Wire-протокола, наиболее рациональной является идея о возложении отдельной задачи по обслуживанию однопроводной линии на специальный узел заказной или полузаказной интегральной схемой, для последующего сопряжения такого

цифрового автомата, через системную магистраль, непосредственно с основным процессорным узлом. Фирма Dallas Semiconductor Corp. даже разработала набор рекомендаций по организации подобного узла под названием DS1WM, который был реализован, в том числе, специалистами Xilinx Inc. в виде законченного практического примера для программируемых логических матриц семейств Virtex и Spartan. Более того, и Dallas Semiconductor Corp., которая в том числе известна как поставщик высокоскоростных контроллеров клона MCS51, выпускает специализированный связной микроконтроллер DS80C400, который содержит встроенный в кристалл автомат поддержки 1-Wire-протокола с возможностью реализации механизма активной подтяжки.

Достаточно перспективным представляется также направление, связанное с применением карманных компьютеров (PDA – Personal Digital Assistant) популярных платформ PalmOS, Handspring и WinCE/PocketPC для обслуживания однопроводных компонентов, в том числе работающих в составе 1-Wire-сетей. При этом для подключения PDA к однопроводной шине применяют специализированные адаптеры последовательного порта, которые отличаются малым потреблением и построены на базе схемных решений, использующих перечисленные выше микросхемы сопряжения с 1-Wire-линией. Именно такой подход в настоящее время является наиболее рациональным при организации автономных и мобильных 1-Wire-систем.

Проблема подготовки программного обеспечения для управления мастером линии при обслуживании 1-Wire-сетей, также предельно упрощена компанией-разработчиком. Фирмой Dallas Semiconductor Corp. свободно распространяется профессиональный программный пакет разработчика iButton TMEX SDK, являющийся универсальным средством для профессиональных программистов, который значительно упрощает процесс создания программ для обслуживания однопроводных устройств, подключенных через стандартные типы адаптеров к персональным компьютерам, которые оснащены операционной системой Windows. Он

содержит комплект отлаженных драйверов и утилит для реализации полномасштабного 1-Wire-протокола. В качестве среды взаимодействия с разработчиком пакет iButton TMEX SDK использует специальный стандартизованный программный API-интерфейс. С FTP-сервера компании Dallas Semiconductor Corp. свободно доступен ряд примеров реализации 1-Wire-протокола для некоторых, наиболее популярных видов микропроцессоров, а также готовые библиотеки функциональных программных модулей однопроводного интерфейса для различных программных платформ.

5.3. Ведомые однопроводные компоненты

Ведомые однопроводные компоненты, содержащие 1-Wire-интерфейс, выпускаются в двух различных видах. Либо в корпусах MicroCAN, похожих внешне на дисковый металлический аккумулятор, либо в обычных корпусах для монтажа на печатную плату. Футляр MicroCAN полый внутри (рис. 28). Он выполняет функцию защиты содержащегося в нем полупроводникового кристалла однопроводной микросхемы, который соединен с внешним миром лишь через две, изолированные друг от друга, половинки корпуса, являющиеся по существу контактными площадками для подключения однопроводной линии. В подобных «таблеточных» корпусах поставляются, как правило, приборы iButton.



Рисунок 28. Корпус «таблетка» приборов i-Button

Компоненты, которые предназначены для использования в составе 1-Wire-сетей, упаковываются в пластиковые корпуса интегральных схем. Такой подход объясняется тем, что в отличие от устройств iButton однопроводные приборы для внутриплатажного применения во встраиваемых системах часто имеют более двух выводов. Помимо выводов, которые требуются для обмена данными по однопроводной магистрали, они располагают дополнительными выводами необходимыми, для обеспечения их питания и организации внешних цепей, связывающих такие приборы с объектами автоматизации, например, датчиками или исполнительными устройствами.

К наиболее простым ведомым однопроводным компонентам относится интегральная схема производства корпорации Maxim Integrated Products (разработана Dallas Semiconductors) DS2401 (или модифицированный вариант этой микросхемы с внешним питанием DS2411) и электронный ключ DS2405, управляемый по 1-Wire-интерфейсу.

Первое из этих устройств, также называемое «Кремниевым серийным номером» (Silicon Serial Number), часто используется в качестве электронной метки, которая позволяет идентифицировать состояние, например, механического переключателя, коммутирующего линию данных однопроводного интерфейса. С помощью интегральной схемы DS2405 можно дистанционно осуществить простейшие функции переключения внешнего оборудования, изменяя состояние управляемого ключа относительно возвратного проводника 1-Wire-магистрали.

Однако, наиболее популярными ведомыми компонентами, на базе которых реализовано, пожалуй, наибольшее количество применений, безусловно, являются цифровые термометры типа DS18S20 (более известные под обозначением DS1820, который уже не производится, но успел стать международным брендом). Преимущества этих цифровых термометров с точки зрения организации магистрали, по сравнению с любыми другими интегральными температурными сенсорами, а также неплохие

метрологические характеристики и хорошая помехоустойчивость, выводят их на первое место при построении многоточечных систем температурного контроля в диапазоне от -55°C до $+125^{\circ}\text{C}$. [7]

5.4. Схемная реализация 1-Wire интерфейса

Для начала рассмотрим принципиальную электрическую схему, реализующую 1-Wire интерфейс. Схема соединения ведущего и ведомого устройств посредством однопроводной шины приведена на рис. 29. На том же рисунке показаны особенности схемной реализации выходных каскадов ведущего и ведомого устройств. Как видно из рисунка, в схеме 1-Wire-интерфейса, так же как и в интерфейсе для I²C шины, используются выходные каскады с открытым коллектором (стоком) и общей нагрузкой для всех элементов сети. В спецификации для 1-Wire интерфейса специально оговаривается, что резистор R_H должен находиться в непосредственной близости от ведущего устройства.

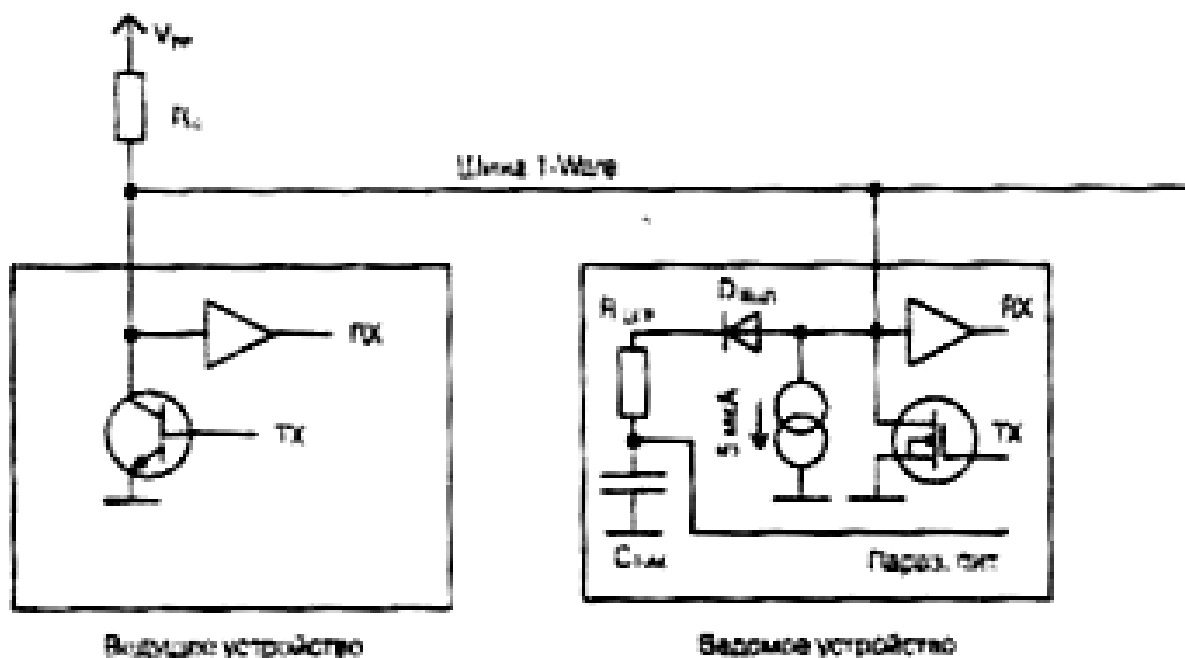


Рисунок 29. Схемотехническая реализация портов устройств 1-Wire

Биполярный транзистор на рис. 29 в выходном каскаде ведущего устройства показан условно, сегодня чаще применяются выходные каскады, построенные по КМОП-технологии. Ведомые устройства обычно целиком построены на КМОП-транзисторах.

В режиме ожидания все выходные транзисторы закрыты. На шине присутствует напряжение логической единицы. Информация по шине передается при помощи импульсов отрицательной полярности. Любое устройство, подключенное к шине, как ведущее, так и ведомое, может создавать отрицательные импульсы и тем самым передавать информацию. Однако ведомое устройство начинает процесс передачи только под управлением ведущего.

Следует обратить внимание на дополнительные элементы в составе ведомого устройства (см. рис. 29). Во-первых, это встроенный источник тока (обозначен двумя пересекающимися окружностями). Этот источник создает внутреннюю утечку на входе 1-Wire интерфейса. Назначение утечки – создать нулевой уровень сигнала на внутренних элементах ведомого устройства при его отключении от шины 1-Wire. Когда соединение будет восстановлено, внутренняя логика ведомой микросхемы обнаруживает перепад напряжения с нуля на единицу. Сразу после получения такого сигнала ведомая микросхема должна выдать на шину 1-Wire сигнал присутствия (о его реализации – см. далее). Обнаружив сигнал присутствия на линии связи, ведущее устройство может провести процедуру обнаружения новых устройств. Таким образом, шина 1-Wire позволяет легко подключать и отключать различные устройства, не нарушая при этом их работу.

Еще один дополнительный элемент, который присутствует только в схеме интерфейса ведомого устройства – это цепь паразитного питания. Эта цепь состоит из накопительного конденсатора ($C_{\text{нлк}}$), токоограничивающего резистора ($R_{\text{огр}}$) и выпрямительного диода ($D_{\text{вып}}$). Все эти элементы используются в режиме паразитного питания. Каждая микросхема, рассчитанная на работу с 1-Wire интерфейсом, имеет два режима питания.

Первый режим – обычный. В этом режиме питание подается на специальный вывод микросхемы. Такой вывод обычно имеется у всех микросхем, кроме приборов серии iButton.

Независимо от наличия вывода питания любая микросхема может питаться непосредственно от информационной шины. Этот способ получения электроэнергии, как указывалось выше, называется «паразитным питанием». В процессе работы на шине всегда присутствует импульсный сигнал. В те моменты, когда напряжение на шине равно единице, выпрямительный диод открывается и накопительный конденсатор заряжается через токоограничивающий резистор. Емкость конденсатора невелика (примерно 800 нФ). Однако и ток потребления КМОП микросхемы тоже очень мал, поэтому заряда конденсатора хватает для обеспечения питания микросхемы в промежутках между импульсами.

Однако режим паразитного питания применим далеко не во всех случаях. Некоторые операции невозможно выполнить при сверхмалом потреблении энергии. Например, процесс преобразования температуры в код в соответствующих интегральных схемах требует полноценного, а не паразитного питания. На этот случай 1-Wire протокол предусматривает третий, специальный режим питания. Такой режим питания используется только совместно с режимом паразитного питания. Основная идея – подавать полноценное питание на ту же самую шину, по которой передается информация в те моменты времени, когда ведомая микросхема выполняет особо энергоемкие операции. При этом пока на шине присутствует полноценное питание, передача информации невозможна.

5.5. Синхронизация и побитная передача информации

5.5.1. Синхронизация при передаче

Как и прочие интерфейсы, интерфейс 1-Wire имеет несколько разных уровней описания. Самый низкий уровень описывает, каким образом передаются отдельные биты. Как уже говорилось выше, 1-Wire протокол

предусматривает двухсторонний обмен информацией. При этом все операции на шине производятся исключительно под управлением Master-устройства. Master может выполнять операции двух видов: записывать информацию в Slave устройство или считывать информацию из него. Информация передается побайтно, в последовательном виде, бит за битом, начиная с младшего бита.

В любом из этих двух случаев, для передачи информации Master-устройство вырабатывает на шине тактовые импульсы. Для этого оно периодически «подсаживает» шину при помощи выходного транзистора своего I-Wire интерфейса (см. рис. 29). Полезная информация передается путем изменения длительности этих импульсов. Причем при записи информации длительностью импульсов управляет исключительно Master-устройство.

В режиме чтения начинается формирование импульса Master-устройство, но Slave-устройство может продлевать длительность любого импульса, «подсаживая» в свою очередь сигнал на линии в нужный момент. На рис. 30 изображены две временные диаграммы. Верхняя диаграмма иллюстрирует режим записи двух разных битов информации, а нижняя — режим чтения. Участки диаграммы, где линия «отпущена» и уровень сигнала на линии определяется лишь резистором R_n , изображены на диаграмме при помощи тонких линий. Участки, где один из элементов сети «подсаживает» линию, изображены при помощи широких линий. Рассмотрим подробнее два режима работы сети.

5.5.2. Передача информационного бита

В исходном состоянии все Slave-устройства, подключенные к шине, находятся в режиме ожидания – линия «отпущена». То есть выходные транзисторы всех элементов шины закрыты, и напряжение на шине определяется резистором нагрузки (R_n на рис. 29). Для того чтобы записать данные в одно из Slave-устройств, Master начинает формировать

отрицательные синхроимпульсы (верхняя диаграмма на рис. 30). На каждый передаваемый бит формируется один импульс. Импульсы передаются путем «подсаживания» линии до нуля. Для передачи каждого бита выделяется промежуток времени стандартной длительности. Этот промежуток получил название «слот» (Slot). Если значение передаваемого бита равно 0, то Master вырабатывает шинный импульс, длина которого равна длительности слота. Для передачи единичного бита Master вырабатывает короткий импульс, который, по сути, является чистым синхроимпульсом. Оставшаяся часть слота должна быть заполнена единичным сигналом. Между двумя слотами обязательно должен быть небольшой промежуток, во время которого уровень сигнала на шине тоже равен единице.

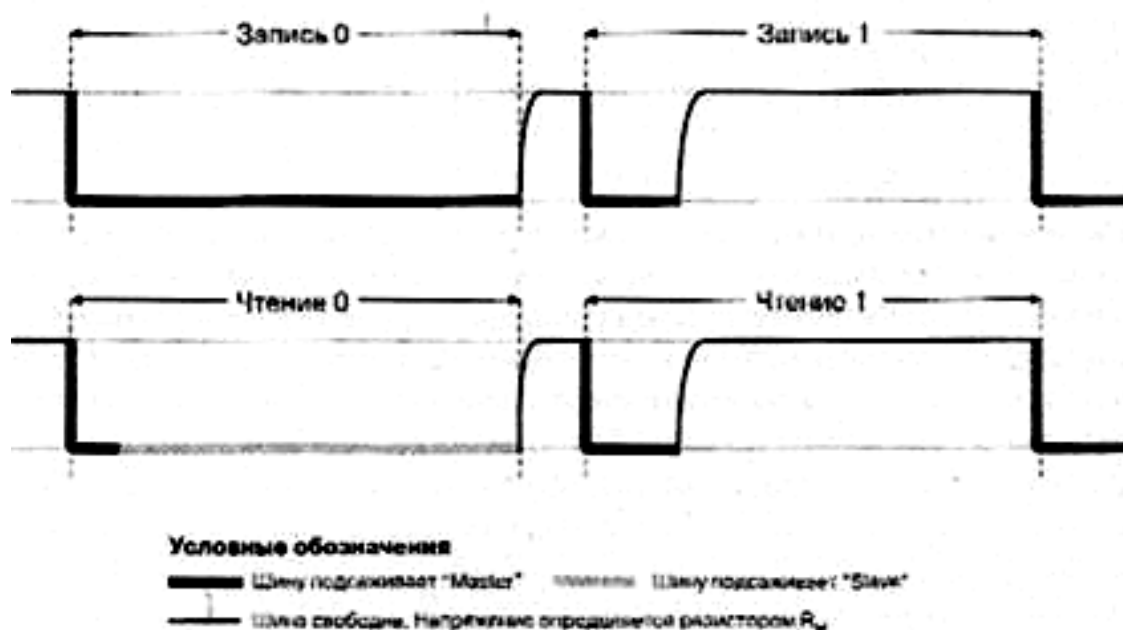


Рисунок 30. Временные диаграммы записи и чтения информации на однопроводной линии 1-Wire

Slave-устройство в этом режиме только принимает сигнал. Для этого оно постоянно находится в режиме ожидания. Обнаружив начало синхроимпульса, Slave-устройство начинает процесс приема бита информации. Передний фронт этого импульса служит Slave-устройству началом отсчета. Выдержав паузу, равную длительности синхроимпульса,

Slave-устройство считывает уровень сигнала на линии. Если в этот момент времени уровень сигнала на линии равен нулю, значит и передаваемый бит равен нулю, если единице, то передаётся единичный бит.

Протокол шины 1-Wire жестко определяет только длительность слота. Интервал между слотами имеет ограничение только на минимальное свое значение. Максимальное значение интервала между слотами не ограничено. Таким образом, скорость передачи данных может регулироваться от своего максимального значения (15.3 Кбит/с) практически до нуля.

5.5.3. Получение информационного бита

Процесс чтения бита информации (нижняя диаграмма на рис. 30) сходен с процессом записи. Отличие его в том, что при чтении Master вырабатывает только синхроимпульсы (короткой длительности). Обнаружив синхроимпульс, Slave устройство должно удлинить или не удлинять этот синхроимпульс в пределах слота. Если очередной считываемый бит равен нулю, то синхроимпульс удлиняется, если единице – удлинения не происходит.

Для иллюстрации этого процесса на рис. 30 участки временной диаграммы, где линию «подсаживает» Master-устройство, изображены черным цветом. Участки, которые «подсаживает» Slave устройство изображены серым цветом. Master-устройство считывает эту информацию. Для этого оно контролирует уровень сигнала внутри слота сразу после синхроимпульса.

5.5.4. Параметры временных интервалов

Для надежной работы однопроводного интерфейса необходимо, чтобы в процессе передачи информации всеми элементами сети строго соблюдались временные параметры. Каждая микросхема, подключенная к сети, самостоятельно вырабатывает все необходимые для ее работы интервалы времени. Для ведущего устройства сети эти требования более

жесткие, чем для ведомых. Это связано с тем, что в качестве ведущего устройства обычно выступает микроконтроллер. Любой микроконтроллер способен с высокой точностью обрабатывать любые временные интервалы, благодаря использованию кварцевого резонатора. Ведомые же микросхемы обычно выполнены в микроминиатюрном корпусе и временные параметры таких микросхем формируются обычно параметрическими методами.

На рис. 31 приведены временные параметры протокола 1-Wire в различных режимах работы. Как видно из рисунка, величина слота для передачи одного бита информации (T_x) должна лежать в пределах от 60 до 120 мкс. Длительность синхроимпульса равна 1 мкс. Ведомое устройство, обнаружив на шине передний фронт синхроимпульса, должно сформировать задержку минимум в 15 мкс, и затем произвести проверку сигнала на шине. Допустимый разброс времени задержки для разных экземпляров микросхем лежит в пределах от 15 до 60 мкс. Этот диапазон показан на рисунке в виде области, обозначенной как «Зона проверки уровня Slave».

Минимальная величина интервала между слотами равна 1 мкс. Максимальная, как уже говорилось, не ограничена.

В режиме записи нулевого бита Master вырабатывает только синхроимпульсы, длительность которых равна 1 мкс. Если читаемый бит равен нулю, Slave устройство продлевает длительность синхроимпульса. Минимальная длительность продленного импульса составляет 15 мкс. Для этого временного интервала тоже допускается довольно значительный разброс. В пределах этого разброса длительность удлиненного импульса можем вырасти еще на 45 мкс. Если читаемый бит равен единице, удлинения синхроимпульса не происходит. Для того чтобы правильно оценить значение читаемого байта, Master-устройство должно прочитать уровень сигнала на шине сразу после окончания синхроимпульса, но не позднее, чем через 15 мкс. Зона проверки уровня для Master-устройства в режиме чтения значительно меньше аналогичной зоны для Slave-устройства в режиме записи.

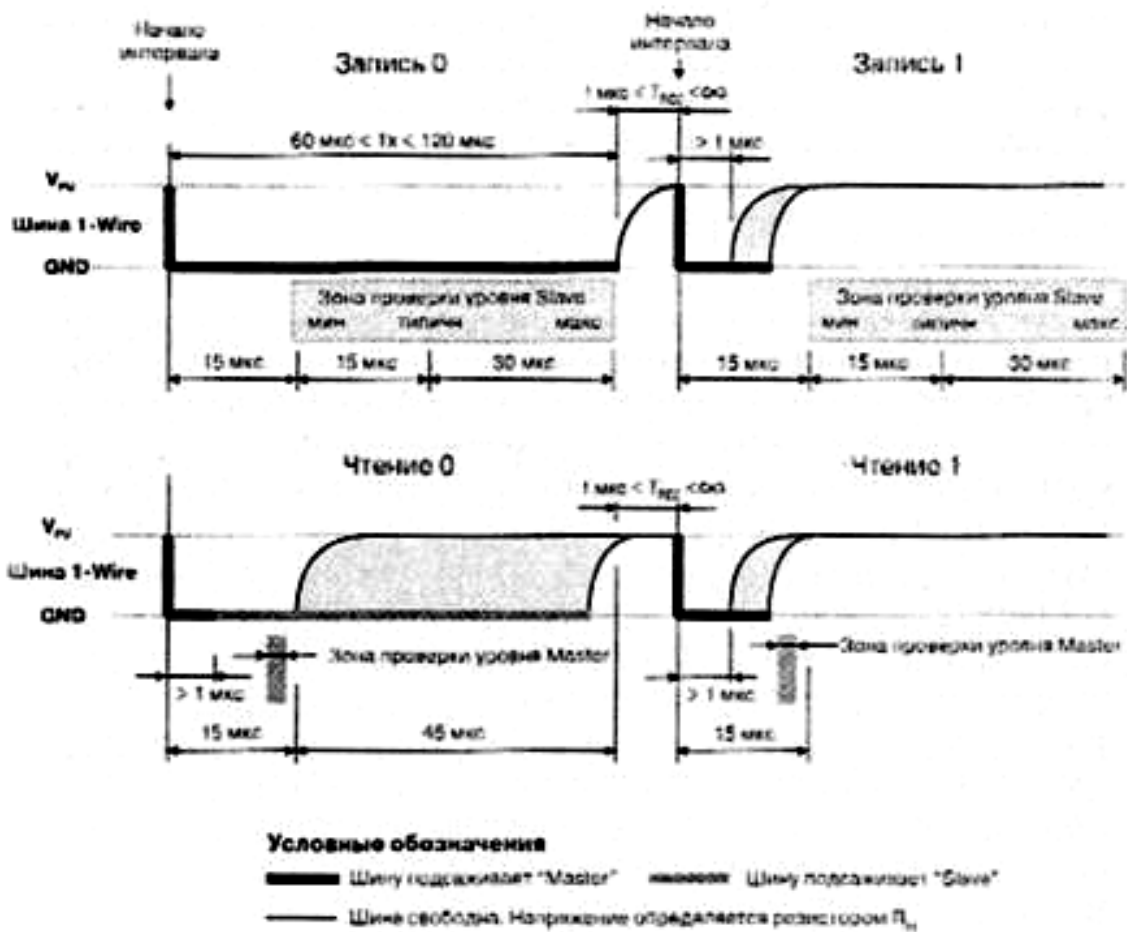


Рисунок 31. Временные параметры протокола 1-Wire в различных режимах работы

5.5.5. Побайтная группировка последовательности бит

Байты при передаче формируются путем подсчета битов, начиная с самого младшего. Первые восемь битов – это первый байт. Следующие восемь битов – второй байт и так далее. Начало же всей этой цепочки определяется сигналом сброса. Любой цикл обмена данными в сети 1-Wire начинается с импульса сброса. Импульс сброса – это сверхдлинный отрицательный импульс на шине 1-Wire, вырабатываемый ведущим устройством.

5.6. Сброс и обнаружение присутствия на линии

Временная диаграмма, иллюстрирующая процесс формирования импульса сброса, приведена на рис. 32. С импульсом сброса тесно связан еще один служебный сигнал – сигнал присутствия на шине. Сигнал присутствия вырабатывает каждое Slave-устройство сразу после окончания действия импульса сброса. Master-устройство должно проконтролировать наличие этого импульса. Если импульса присутствия нет, то это значит, что на линии нет ни одного Slave устройства.

Кроме инициации импульсов присутствия импульс сброса переводит в исходное состояние всю систему. Любые незаконченные процессы на линии моментально завершаются, и отсчет битов начинается сначала. Временные характеристики сигнала сброса и сигнала присутствия на линии приведены на рис. 32. Длительность импульса сброса должна быть не менее 480 мкс. Процесс передачи информации по линии может начинаться также не раньше, чем через 480 мкс после окончания действия импульса сброса. В этом временном интервале и ожидается появление сигнала присутствия. Для этого после окончания импульса сброса Master «отпускает» линию и ждет сигнала от Slave-устройств.



Рисунок 32. Временная диаграмма процесса начального сброса

Каждое Slave-устройство, обнаружив импульс сброса, выдерживает паузу в 15...60 мкс, а затем «подсаживает» линию. Длительность импульса присутствия составляет 60...240 мкс. Сигналы присутствия от всех Slave-устройств, если на шине более устройства, сливаются в один общий импульс. Ведущее устройство проверяет наличие нулевого уровня на линии в середине этого интервала. Если сигнал обнаружен, то это значит, что на линии имеется хотя бы одно нормально работающее Slave-устройство и Master может продолжать работу в сети. Если сигнал не обнаружится, то микропроцессор перейдет к обработке этой ситуации. Обычно в этом случае он выдает сигнал ошибки (зуммер, светодиод, надпись на дисплее и т.д.).

5.6.1. Команды шины 1-Wire

Для управления направлением передачи в сети 1-Wire существует понятие команды. Master отправляет команды, все Slave-устройства их выполняют. Любая операция начинается с определенной команды, которая представляет собой один байт информации.

Для описания механизма передачи команд, рассмотрим передачу и выполнение команды «Чтение ПЗУ» или «Read ROM» (ROM – Read Only Memory – Постоянное Запоминающее Устройство).

В ПЗУ устройства с интерфейсом 1-Wire записан ее индивидуальный код (ID-код). Команда «Чтение ПЗУ» позволяет микроконтроллеру (Master-устройству) прочитать этот код. На рис. 33 приведена временная диаграмма сигнала, возникающего на линии 1-Wire в процессе выполнения команды «Чтение ПЗУ» (для удобства восприятия все импульсы внутри пакетов условно показаны одной длительности). Следует заметить, что команда «Чтение ПЗУ», в отличие от всех остальных команд, может корректно выполняться только в том случае, когда на шине имеется один Master и одно Slave-устройство.

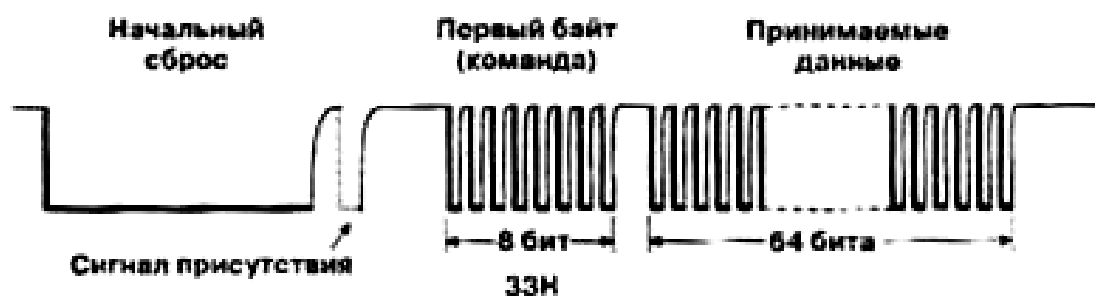


Рисунок 33. Временная диаграмма чтения данных на линии 1-Wire

Выполнение команды начинается с импульса сброса. Затем Slave-устройство вырабатывает, а Master проверяет сигнал присутствия на линии. При обнаружении сигнала присутствия Master выдаст на линию код команды «Чтение ПЗУ» (33h). Код выдается в режиме записи. Получив этот код, Slave микросхема переключается в режим выдачи информации. В этом режиме Master читает из Slave-микросхемы 64 бита (8 байт), составляющие её ID-код. Последние восемь битов кода – это контрольная сумма, вычисляемая по специальному алгоритму из его первых 56 битов. Микропроцессор проверяет контрольную сумму и в случае несоответствия повторяет процедуру «Чтение ПЗУ».

Разумеется команда «Чтение ПЗУ» не единственная в протоколе 1-Wire-шины. Существует целая система команд, позволяющая гибко управлять процессами обмена. Для ознакомления с данной системой команд можно обратиться к спецификации интерфейса на Интернет-сайте компании-разработчика или с другой литературой, содержащей подробное описание интерфейса (например, [8]).

6. I²S. Интерфейс передачи цифровых аудиосигналов

В настоящее время на потребительском рынке представлены многочисленные цифровые аудиосистемы: проигрыватели компакт-дисков со звуковыми записями, цифровые процессоры звука, системы передачи звукового сопровождения цифрового телевизионного сигнала и т.д.

Звуковой сигнал в цифровой форме обрабатывается в таких системах интегральными схемами различного назначения:

- аналого-цифровые и цифро-аналоговые процессоры (АЦП и ЦАП);
- цифровые сигнальные процессоры;
- микросхемы декодирования и коррекции ошибок;
- цифровые фильтры;
- микросхемы цифровых интерфейсов ввода/вывода.

Вследствие этого стандартизированные коммуникационные структуры, как и в случае с другими интерфейсами, крайне необходимы как разработчикам и производителям звукового оборудования, так и фирмам, выпускающим интегральные схемы, т.к. это увеличивает гибкость систем и минимизирует затраты на их создание. Для достижения данных преимуществ и был разработан интерфейс I²S (Inter-IC Sound – звук, передаваемый между интегральными схемами) – специализированный однонаправленный последовательный интерфейс для цифрового аудио. [9]

6.1. Основные требования

Прежде всего, в спецификации отмечается, что интерфейс предназначен для передачи только звуковых данных, тогда как другие сигналы, например, сигналы управления, должны передаваться отдельно. Для минимизации количества используемых выводов интегральных схем и для упрощения интерфейса используются три линии последовательной передачи

данных: одна линия для передачи двухканального цифрового звука с временным разделением, линия выбора звукового канала и линия синхронизации.

Передатчик и приемник объединены одним синхронизирующим (тактовым) сигналом для передачи данных. При этом ведущим (Master-устройством) в данном тандеме может выступать как передатчик, так и приемник. Ведущий обязан генерировать для ведомого (Slave-устройства) тактовый сигнал для последовательной передачи битов информации и сигнал выбора канала. Передача самих звуковых данных, соответственно, происходит от передатчика к приемнику.

Спецификация на интерфейс, однако, допускает в сложных системах, когда одни и те же линии используются для передачи информации между многими передатчиками и приемниками, вводить так называемого «системного ведущего» («system master»), который должен управлять передачей между различными интегральными схемами. Передатчики в данном случае фактически функционируют как ведомые и должны передавать данные по внешнему тактовому сигналу системного ведущего.

На рис. 34. некоторые типичные варианты организации коммуникаций с использованием I²S, а также временная диаграмма для иллюстрации процесса передачи. Следует отметить, что системный ведущий может совмещать в себе также функции передатчика или приемника, при этом его активация может производиться программно или путём управления каким-либо входным портом (определяется и реализуется проектировщиком встраиваемой системы и программистом микропроцессора или микроконтроллера).

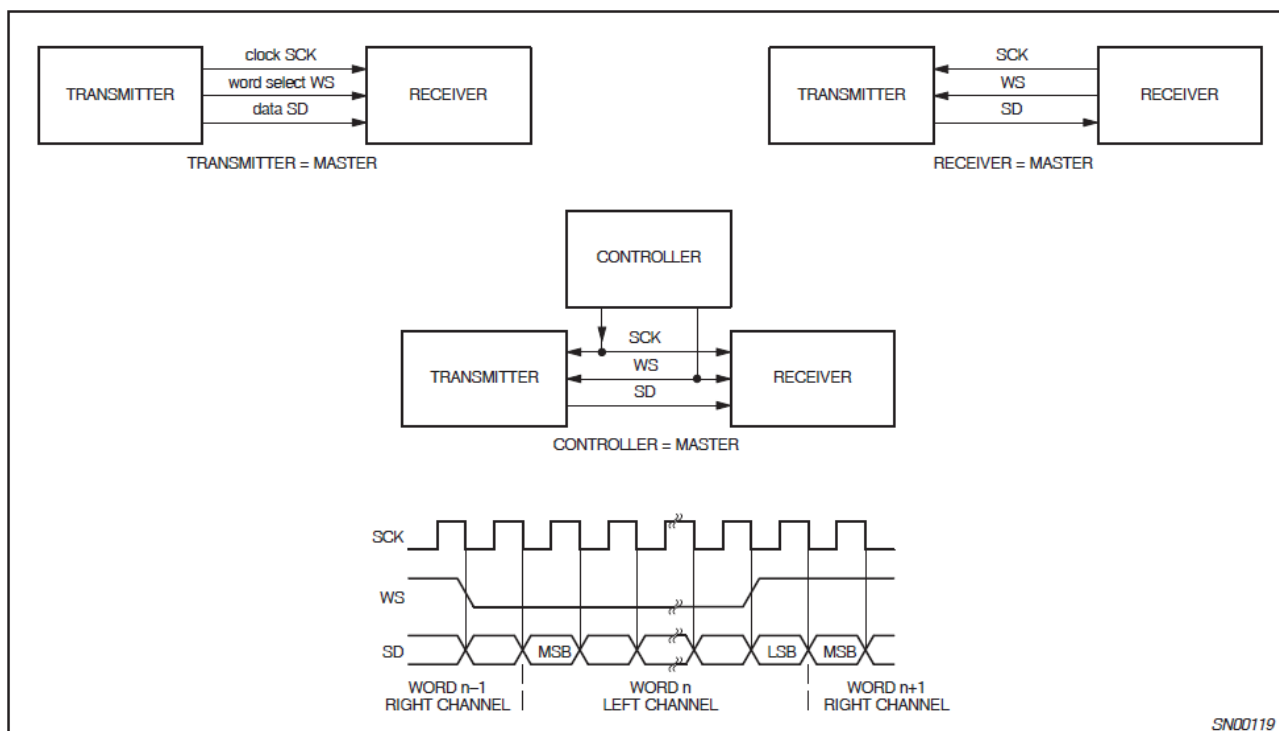


Рисунок 34. Варианты реализации обмена информацией по интерфейсу I²S

6.2. Принцип передачи информации

Как было указано выше и показано на рис. 34, интерфейс состоит из трёх линий:

1. непрерывный тактовый сигнал (SCK);
2. выбор канала (WS);
3. последовательные данные (SD).

Устройство, генерирующее SCK и WS, является ведущим.

6.2.1. Линия последовательной передачи данных

Последовательные данные передаются по I²S в двоичном виде, старшим битом вперед. Такой порядок передачи бит выбран не случайно – это позволяет передатчику и приемнику использовать различную разрядность звуковых данных и при этом успешно передавать информацию

без каких-либо дополнительных настроек. Принцип реализации данной возможности следующий:

- если приемник получает больше бит, чем его собственная разрядность данных, то все выходящие за разрядную сетку биты, начиная с самого младшего, игнорируются;
- если приемник получает меньше бит, чем его собственная разрядность, то все недостающие младшие биты в составе принятого слова принимаются за '0'.

Таким образом, самый старший бит передаваемых слов всегда имеет фиксированную позицию, а положение самого младшего бита зависит от принятой в каждом из абонентов разрядности звуковых данных.

Передачик всегда отправляет старший бит очередного слова со звуковыми данными в первом тактовом периоде, следующим за изменением WS. То есть время, в течение которого, WS остаётся относительно тактовых импульсов неизменной определяет длину слов со звуковыми данными, отправляемыми передатчиком.

Последовательные данные от передатчика могут быть синхронизированы как задним фронтом (изменение от высокого к низкому уровню), так и передним фронтом (изменение от низкого к высокому уровню) такового сигнала. Однако биты данных должны защёлкиваться приемником только по переднему фронту тактового импульса. При проектировании встраиваемой системы следует обратить на данное требование внимание во избежание битового смещения данных при передаче.

6.2.2. Линия выбора канала

Линия выбора канала своим состоянием определяет передачу «левого» или «правого» звукового канала в каждый момент времени:

- $WS = 0$ – передается слово канала №1 («левый»);
- $WS = 1$ – передаётся слово канала №2 («правый»).

WS может изменяться и по переднему, и по заднему фронту тактового импульса. Изменение должно происходить за один тактовый импульс до начала передачи самого старшего бита слова звуковых данных. Ведомый, при этом, защёлкивает состояние линии WS только по переднему фронту такта.

6.3. Временные требования

В соответствии с требованиями к структуре систем, объединенных интерфейсом I²S, любое из устройств может выполнять функции ведущего, обеспечивая генерацию тактового сигнала. А ведомое устройство обычно использует внешний тактовый сигнал, поступающий от ведущего, для внутренней синхронизации. Это означает, что необходимо принимать в расчет вероятные задержки распространения сигналов синхронизации, выбора канала и собственно последовательных данных. Общая задержка является суммой следующих слагаемых:

1. Задержка между внешним тактовым сигналом (от ведущего) и внутреннего тактового сигнала ведомого;
2. Задержка между внутренним тактовым сигналом последовательными данными и/или сигналом выбора канала.

Для входа данных и входа выбора канала задержка между внешними и внутренними тактовыми импульсами не имеет значения, т.к. она только увеличивает время установки действительных уровней сигнала. Основное значение имеет разница между временем распространения сигнала от передатчика и временем восстановления приемника для начала приема данных.

Все временные требования специфицируются относительно периода тактовых импульсов или минимально допустимой длительности периода для интегральной схемы. Это означает, что в будущем возможно получение более высокой скорости передачи данных через интерфейс.

6.4. Требования к уровням напряжения сигналов

Спецификация интерфейса I²S определяет соответствие логических сигналов '0' и '1' уровням напряжения традиционной технологии производства интегральных схем ТТЛ (Транзисторно-Транзисторная Логика):

- низкий уровень (логический '0') на входе: менее 0.8В;
- высокий уровень (логическая '1') на входе: более 2.0В;
- низкий уровень (логический '0') на выходе: менее 0.4В;
- высокий уровень (логическая '1') на выходе: более 2.4В.

(Нагрузочная способность выходов также соответствует ТТЛ-технологии: при низком уровне не менее -1.6мА, при высоком уровне не более 0.04мА.)

В настоящее время, однако, в результате гораздо более широкого распространения интегральных схем, выполненных по технологии КМОП (Комплементарная логика на транзисторах Металл-Оксид-Полупроводник), имеющих различное напряжение питания, уровни напряжения для логических сигналов интерфейса обычно соответствуют уровням напряжения всех остальных цифровых сигналов, которыми оперирует интегральная схема.

7. АСИНХРОННЫЙ ИНТЕРФЕЙС ПОСЛЕДОВАТЕЛЬНОГО ПОРТА

Один из наиболее давно известных и распространенных стандартов последовательной дуплексной асинхронной передачи данных – RS-232C (Reference Standard №232 Revision C – эталонный стандарт №232, версия C). Достаточно длительное время портом с RS-232C оснащались персональные компьютеры (так называемый COM-порт). Чаще всего порт использовался для соединения с манипуляторами типа «мышь» и с модемами, хотя его применение специально никак не ограничивалось – в ряде случаев, например, с помощью него организовывался обмен информацией между двумя компьютерами.

В настоящее время, интерфейс RS-232C (или его вариации с отличными от специфицированных уровнями напряжения при сигнальном кодировании) применяется в промышленных сетях, контроллерах и компьютерах, в качестве интерфейса сервисных портов коммуникационного и в качестве внутрислатного интерфейса встраиваемых систем. [10]

7.1. Типы взаимодействующего оборудования и набор сигналов

Данный интерфейс, определенный стандартом Ассоциации электронной промышленности (EIA), подразумевает наличие оборудования двух видов: терминального DTE и связного DCE. Чтобы не составить неправильного представления об интерфейсе RS-232C, необходимо отчетливо понимать различие между этими видами оборудования. Терминальное оборудование, например компьютер, может посылать и (или) принимать данные по последовательному интерфейсу. Оно как бы оканчивает (terminate) последовательную линию. Связное оборудование – устройства, которые могут упростить передачу данных совместно с терминальным оборудованием. Наглядным примером связного оборудования служит модем (модулятор–демодулятор). Он оказывается соединительным

звеном в последовательной цепочке между компьютером и телефонной линией.

Различие между терминальными и связными устройствами довольно расплывчато, поэтому возникают некоторые сложности в понимании того, к какому типу оборудования относится то или иное устройство. К тому же нередко встречаются применения интерфейса для связи идентичных устройств, например, собственно двух персональных компьютеров. В конечном счете, если проанализировать функции сигналов интерфейса RS–232C (см. табл. 5), то достаточно просто определить соединяемые линии одного и другого абонента вне зависимости от типа устройства.

Таблица 5. Функции всех сигнальных линий интерфейса RS–232C

| Номер контакта | Сокращение | Направление | Полное название |
|----------------|-----------------|-------------|--|
| 1 | <i>FG</i> | — | Основная или защитная земля |
| 2 | <i>TD (TXD)</i> | K DCE | Передаваемые данные |
| 3 | <i>RD (RXD)</i> | K DTE | Принимаемые данные |
| 4 | <i>RTS</i> | K DCE | Запрос передачи |
| 5 | <i>CTS</i> | K DTE | Сброс передачи |
| 6 | <i>DSR</i> | K DTE | Готовность модема |
| 7 | <i>SG</i> | — | Сигнальная земля |
| 8 | <i>DCD</i> | K DTE | Обнаружение несущей |
| 9 | — | K DTE | (Положительное контрольное напряжение) |
| 10 | — | K DTE | (Отрицательное контрольное напряжение) |
| 11 | <i>QM</i> | K DTE | Режим выравнивания |
| 12 | <i>SDCD</i> | K DTE | Обнаружение несущей вторичных данных |
| 13 | <i>SCTS</i> | K DTE | Вторичный сброс передачи |
| 14 | <i>STD</i> | K DCE | Вторичные передаваемые данные |
| 15 | <i>TC</i> | K DTE | Синхронизация передатчика |
| 16 | <i>SRD</i> | K DTE | Вторичные принимаемые данные |
| 17 | <i>RC</i> | K DTE | Синхронизация приемника |
| 18 | <i>DCR</i> | K DCE | Разделенная синхронизация приемника |
| 19 | <i>SRTS</i> | K DCE | Вторичный запрос передачи |
| 20 | <i>DTR</i> | K DCE | Готовность терминала |
| 21 | <i>SQ</i> | K DTE | Качество сигнала |
| 22 | <i>RI</i> | K DTE | Индикатор звонка |
| 23 | — | K DCE | (Селектор скорости данных) |
| 24 | <i>TC</i> | K DCE | Внешняя синхронизация передатчика |
| 25 | — | K DCE | (Занятость) |

В табл. 5 нумерация контактов приведена для наиболее часто используемого для RS-232C 25-контактного разъема типа «D» и дана здесь для информации, поскольку внутрислатное применение интерфейса, разумеется, не подразумевает использование такого громоздкого разъема.

Сигналы интерфейса RS–232C подразделяются на следующие классы:

1. Управляющие сигналы квитирования (например, RTS, CTS).

Сигналы квитирования – средство, с помощью которого обмен сигналами позволяет DTE начать диалог с DCE до фактической передачи или приема данных по последовательной линии связи.

2. Сигналы синхронизации (ТС, RC).

Синхронный режим интерфейса применяется крайне редко, однако в исходной спецификации стандарта они предусмотрены, и в этом случае передаваемая в обоих направлениях информация синхронизируется фронтами сигналов на линиях ТС и RC.

3. Последовательные данные (например, TXD, RXD).

Интерфейс RS–232C обеспечивает два независимых последовательных канала данных: первичный (главный) и вторичный (вспомогательный).

Оба канала могут работать в дуплексном режиме, т.е. одновременно осуществляют передачу и прием информации.

На практике вспомогательный канал RS–232C применяется редко, и в асинхронном режиме вместо 25 линий используются 9 линий и соответственно 9-контактный разъем типа «D» (табл. 6).

Таблица 6. Основные линии интерфейса RS–232C.

| Номер контакта | Сигнал | Выполняемая функция |
|----------------|--------|--|
| 1 | FG | Подключение земли к стойке или шасси оборудования |
| 2 | TXD | Последовательные данные, передаваемые от DTE к DCE |
| 3 | RXD | Последовательные данные, принимаемые DTE от DCE |
| 4 | RTS | Требование DTE послать данные к DCE |
| 5 | CTS | Готовность DCE принимать данные от DTE |
| 6 | DSR | Сообщение DCE о том, что связь установлена |
| 7 | SG | Возвратный тракт общего сигнала (земли) |
| 8 | DCD | DTE работает и DCE может подключиться к каналу связи |

7.2. Кодирование информации при передаче

Как было упомянуто выше в большинстве схем, содержащих интерфейс RS-232C, данные передаются асинхронно, т.е. в виде последовательности пакета данных. Каждый пакет содержит один информационный символ, а синхронизация осуществляется внутренними тактовыми сигналами абонентов, что требует согласования скорости передачи или, другими словами, согласования длин битовых интервалов.

RS-232C предполагает передачу информации двумя уровнями напряжения «без возврата к нулю». При этом перед началом информационной последовательности (количество информационных бит от 5 до 8) следует так называемый «старт-бит» определенной длительности (высокий уровень), а по окончании – «стоп-бит» одинарной, полуторной или двойной длины (низкий уровень). Кроме того, перед стоп-битом иногда добавляется так называемый бит «четности» или «нечетности», которые позволяют обнаруживать нечетные количества ошибок в информационной части пакета данных. Следует отметить, что поскольку передача асинхронная, то ключевую роль для безошибочной передачи играет правильная настройка скорости передачи для приемной и передающей аппаратуры.

Для примера рассмотрим формат передачи по RS-232C одного из кодов символов ASCII. Символы кода ASCII представляются семью битами, например буква 'А' имеет код 1000001. Чтобы передать букву 'А' по интерфейсу RS-232C, необходимо ввести дополнительные биты, обозначающие начало и конец пакета. Кроме того, как указывалось выше, желательно добавить лишний бит для простого контроля ошибок по паритету (допустим, бит четности).

Наиболее широко распространен формат, включающий в себя один стартовый бит, один бит паритета и два стоповых бита. Начало пакета данных всегда отмечает низкий уровень стартового бита. После него следует

7 бит данных. Бит четности содержит 1 или 0 так, чтобы в нашем случае общее число единиц в 8-битной группе было четным. Последним передаются два стоповых бита, представленных высоким уровнем напряжения. Эквивалентный ТТЛ-сигнал при передаче буквы 'А' показан на рис. 35.

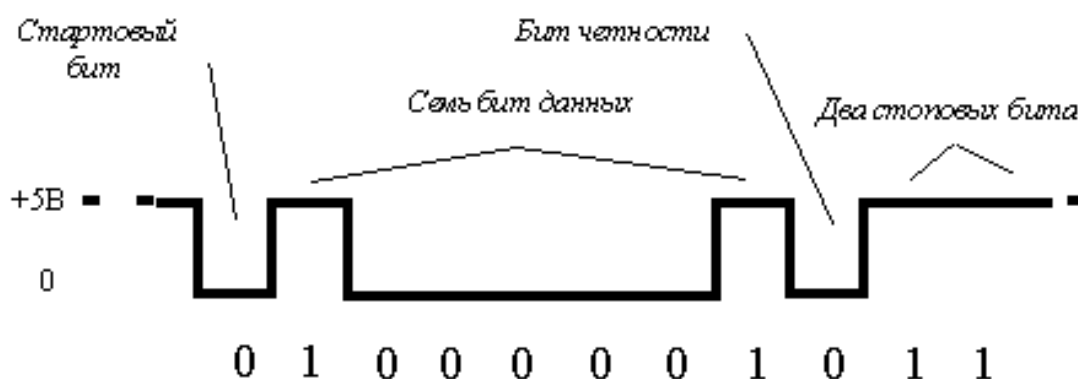


Рисунок 35. Представление кода буквы А сигнальными уровнями ТТЛ.

Таким образом, полное асинхронно передаваемое слово состоит из 11 бит (фактически данные содержат только 7 бит) и записывается в виде 01000001011.

Используемые в интерфейсе RS-232C уровни сигналов отличаются от уровней сигналов, применяемых для внутриплатного обмена. Логический 0 (SPACE) представляется положительным напряжением в диапазоне от +3 до +25 В, логическая 1 (MARK) — отрицательным напряжением в диапазоне от –3 до –25 В (за счет такой большой разности потенциалов между уровнями нуля и единицы данный стандарт достаточно устойчив к внешним помехам.). На рис. 36 показан сигнал в том виде, в каком он существует на линиях TXD и RXD интерфейса RS–232C.

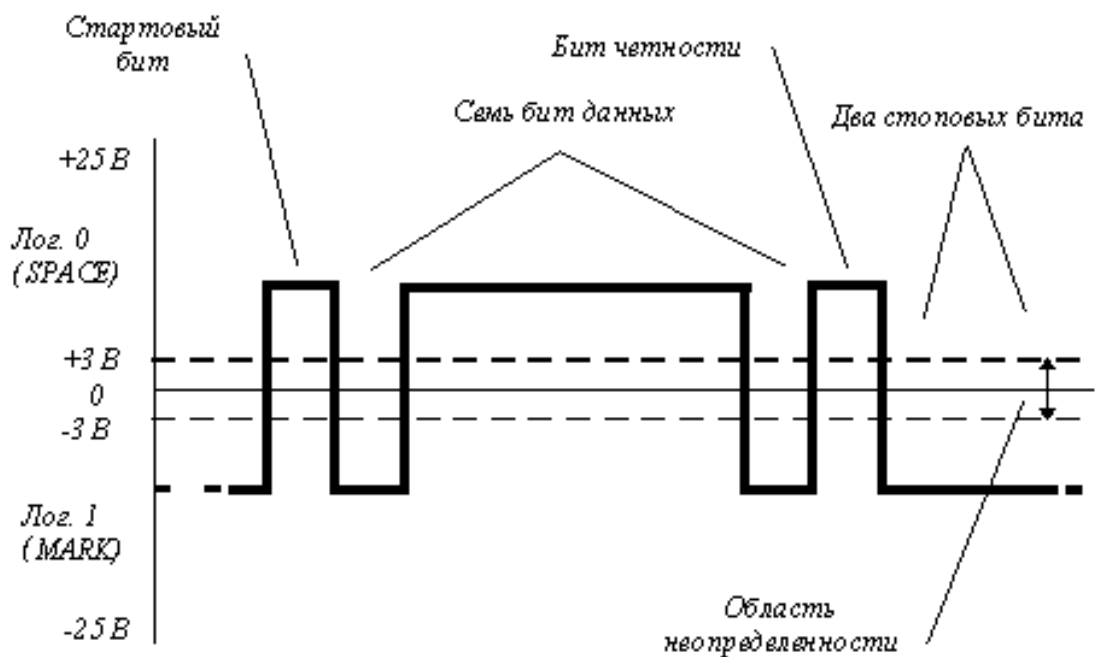


Рисунок 36. Вид кода буквы А на сигнальных линиях *TXD* и *RXD*

Сдвиг уровня, т.е. преобразование ТТЛ-уровней в уровни интерфейса RS-232C и наоборот производится специальными микросхемами драйвера линии и приемника линии (например, интегральная схема MAX232 от Maxim Integrated Products или ADM232 от Analog Devices).

7.3. Особенности внутриплатного применения RS-232C

В силу того, что внутриплатное применение интерфейса не требует передачи на такие значительные расстояния, для которых разработан стандарт, то сигнальное кодирование разнополярным напряжением с большим размахом не целесообразно. Вследствие этого вариант асинхронной последовательной передачи данных с использованием формата интерфейса RS-232C, но с ТТЛ-уровнями сигналов (как на рис. 35 выше) в настоящее время достаточно широко распространен.

В частности, микроминиатюрные модули GSM-модемов, GPS/ГЛОНАСС приемопередатчиков, монтируемых непосредственно на печатную плату обычно оснащены TTL-версией RS-232C.

Микроконтроллеры и DSP-процессоры сегодня, как правило, оснащены интегрированным на кристалл периферийным модулем, который позволяет реализовать требуемый интерфейсом формат передачи данных. В большинстве случаев такие периферийные модули имеют наименование USART (Universal Synchronous Asynchronous Receiver Transmitter – универсальный синхронный асинхронный приемо-передатчик) или просто UART, если синхронный режим не предусматривается.

8. LVDS. Низковольтная дифференциальная передача сигналов

Современные системы обработки информации требуют применения высокоскоростных интерфейсов, обеспечивающих достоверную передачу данных, малое потребление и низкую стоимость реализации. Все эти достоинства сочетает в себе интерфейс LVDS (Low-Voltage Differential Signaling – низковольтная дифференциальная передача сигналов), и на данный момент этот интерфейс принят как стандарт де-факто. Этому способствует большой ассортимент микросхем и различных датчиков (в основном по захвату изображений и сигналов с высокой скоростью), оснащенных данным интерфейсом. [11]

Метод низковольтной дифференциальной передачи сигналов использует очень малые перепады дифференциального напряжения (до 350 мВ) на двух проводниках печатной платы встраиваемой системы или линиях сбалансированного кабеля. Ключевыми преимуществами интерфейса LVDS являются [12]:

- высокая производительность (от сотен до нескольких тысяч мегабит в секунду);
- низкая мощность потребления при передаче;
- минимальный уровень излучаемых шумов;
- низкая чувствительность к внешним шумам;
- относительная дешевизна применения.

8.1. Вопросы стандартизации

Спецификация на интерфейс LVDS разработана фирмой National Semiconductor, которая председательствует в подкомитете ответственном за сигнальные электрические требования, и в настоящее время LVDS отражен в двух стандартах:

- TIA/EIA (Telecommunications Industry Association/Electronic Industries Association) - ANSI/TIA/EIA-644 (LVDS);

- IEEE (Institute for Electrical and Electronics Engineering) - IEEE 1596.3.

Общий мультисистемный LVDS стандарт ANSI/TIA/EIA-644 разработан комитетом TIA Data Transmission Interface TR30.2 и утвержден в ноябре 1995 г. Данный стандарт определяет выходные характеристики передатчиков и входные характеристики приёмников (см. табл. 7), т.е. он определяет только электрические характеристики. Он не ограничивает функциональные спецификации, протоколы, характеристики кабелей, соединений, т.е. оставляет открытыми конкретные варианты применений.

Стандарт ANSI/TIA/EIA рекомендует максимальную производительность в 655Mbps и оговаривает теоретический максимум в 1.923 Gbps, ограниченный потерями в среде распространения. Это позволяет по стандарту специфицировать требуемую максимальную производительность, зависящую от качества сигнала, длины и типа среды распространения.

Таблица 7. Электрические характеристики интерфейса в соответствии со стандартом ANSI/TIA/EIA-644

| ANSI/TIA/EIA-644 (LVDS) Standard | | | | |
|---|---|------------|-------------------|--------------|
| Note: Actual datasheet specifications may be significantly better. | | | | |
| Parameter | Description | Min | Max | Units |
| V_{OD} | Differential Output Voltage | 247 | 454 | mV |
| V_{OS} | Offset Voltage | 1.125 | 1.375 | V |
| ΔV_{OD} | Change to V_{OD} | | 50 | 1mV |
| ΔV_{OS} | Change to V_{OS} | | 50 | 1mV |
| I_{SA}, I_{SB} | Short Circuit Current | | 24 | 1mA |
| t_r/t_f | Output Rise/Fall Times (≥ 200 Mbps) | 0.26 | 1.5 | ns |
| | Output Rise/Fall Times (< 200 Mbps) | 0.26 | 30% of t_{ui} † | ns |
| I_{IN} | Input Current | | 20 | 1μA |
| V_{TH} | Threshold Voltage | | ±100 | mV |
| V_{IN} | Input Voltage Range | 0 | 2.4 | V |

† t_{ui} is unit interval (i.e. bit width).

Стандарт также оговаривает минимальные требования к линии связи, безопасные условия работы приёмника в случаях отказов аппаратуры и другие конфигурационные ограничения, такие как одновременная работа

множества приёмников. Настоящее издание стандарта 644 версии пересмотрено и дополнено информацией о работе на множество приёмников. Пересмотренный стандарт известный как TIA-644-A утверждён в 2000 г.

Другой LVDS стандарт относится к проектам IEEE. Этот стандарт является попыткой развития технологии для целей объединения процессоров в мультипроцессорных системах или объединения рабочих станций в группу. Это программа интерфейса SCI (Scalable Coherent Interface – масштабируемый согласованный интерфейс), оригинально описывающая дифференциальный интерфейс, обеспечивающий высокую скорость передачи информации, но не накладывающий ограничений на потребляемую мощность и степень интеграции.

Стандарт SCI-LVDS малой мощности был позже определён как часть SCI и описан в IEEE1596.3 стандарте. Стандарт SCI-LVDS также описывает уровни сигналов (электрические спецификации), характеризующиеся по отношению к ANSI/TIA/EIA стандарту как высокоскоростной малой мощности SCI интерфейс физического уровня. Стандарт принят в марте 1996 г. и определяет и методы кодирования пакетов информации используемых в SCI передаче данных.

В настоящее время интерес представляет развитие широкого стандарта, не определяющего однозначно технологию процесса производства комплектующих, среду распространения, напряжение питания, определённых в перечисленных двух стандартах. Это означает, что LVDS сможет применяться в КМОП, арсенид-галлиевых или других первичных микросхемных технологиях, преодолеет барьер напряжения питания в 5В до 3.3В и даже ниже, сможет применяться не только для передачи информации на печатных платах и через кабель, и тем самым обеспечит чрезвычайно широкий круг применений во многих отраслях индустрии.

8.2. Способ передачи сигналов

LVDS выход содержит источник тока (номиналом 3.5 мА), нагруженный на дифференциальную пару линии передачи (см. рис. 37).

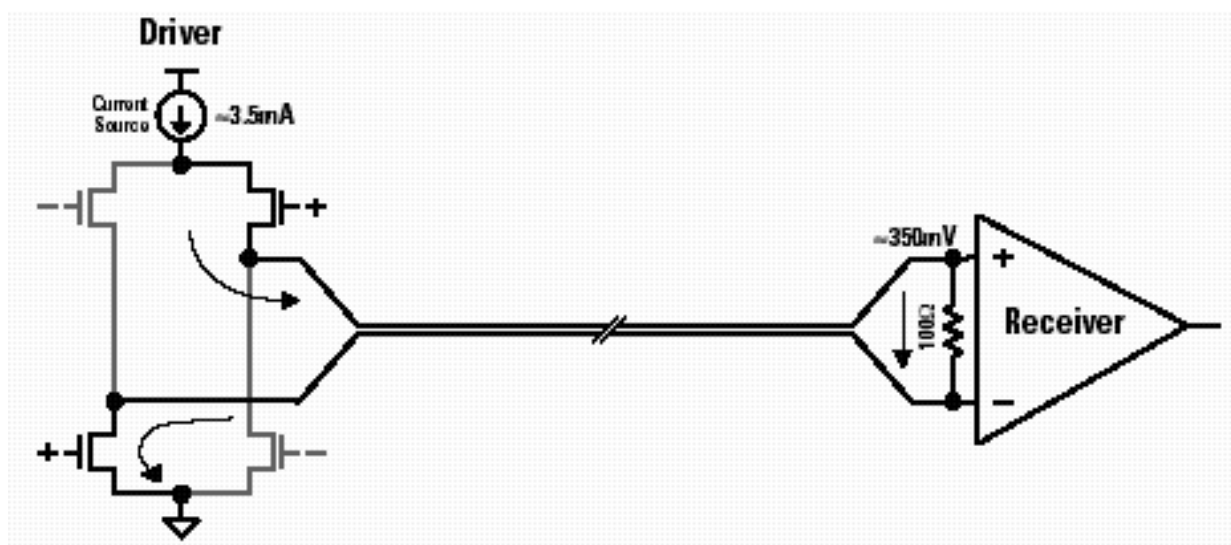


Рисунок 37. Упрощенная схема соединения LVDS передатчика с приёмником через 100 Ом линию

Основной приёмник имеет высокий входной импеданс, поэтому основная часть выходного тока передатчика протекает через 100 Ом резистор терминатора линии, создавая на нём падение напряжения до 350 мВ, приложенное к входу приёмника. При переключении выхода передатчика направление протекания тока через терминатор меняется на противоположное, обеспечивая достоверные логические состояния «0» или «1».

Дифференциальный метод передачи используется в LVDS поскольку обладает меньшей чувствительностью к общим помехам, чем простая однопроводная схема. Данный метод подразумевает двухпроводную схему соединения с формированием перепадов инверсией тока или напряжения в отличие от однопроводной простой схемы передачи информации. Достоинством дифференциального метода является то, что шумы

наводящиеся на двухпроводной линии симметричны и не нарушают дифференциального сигнала, к которому чувствителен приёмник. Кроме этого, токовый выход сигнального передатчика LVDS не склонен к так называемому «звону» в линии и выбросам фронтов, что в целом снижает уровень шума в линии передачи.

Поскольку дифференциальные технологии, в том числе и LVDS, менее чувствительны к шумам, то в них возможно использование меньших перепадов напряжения. Это достоинство является решающим, т.к. невозможно достичь высокой производительности и минимума потребляемой мощности одновременно без снижения размаха напряжения на входе. Формирование малых перепадов напряжения на выходе передатчика достижимо при более высоких скоростях. Токовый режим передатчика обеспечивает очень низкий, всегда постоянный уровень потребления во всём диапазоне частот. Выбросы фронтов передатчика очень незначительны, поэтому ток потребления не увеличивается экспоненциально при увеличении скорости передачи.

Табл. 8 позволяет быстро сравнить основные параметры LVDS метода с параметрами других наиболее часто используемых методов. Из таблицы видно, что LVDS имеет в два раза меньший уровень перепада напряжения по сравнению с PECL методом и одну десятую от перепада RS-422 и традиционных уровней TTL/КМОП. Важным достоинством LVDS является то, что характеристики приёмников и передатчиков не зависят напрямую от напряжения питания схемы, например от +5 В. Поэтому LVDS легко преодолевает барьеры по снижению напряжения питания до 3.3 В и даже 2.5 В без изменения электрических уровней сигналов передачи и производительности. И наоборот технологии ECL и PECL имеют большую зависимость от напряжения питания, что делает весьма затруднительным переход к более низким напряжениям питания в системах использующих данные технологии.

Таблица 8. Сравнение некоторых характеристик LVDS с аналогичными характеристиками других стандартов ANSI/TIA/EIA-644

| Parameter | RS-422 | PECL | LVDS |
|------------------------------------|----------------------|-----------------------|----------------------|
| Differential Driver Output Voltage | ± 2 to ± 5 V | ± 600 - 1000 mV | ± 250 - 450 mV |
| Receiver Input Threshold | ± 200 mV | ± 200 - 300 mV | ± 100 mV |
| Data Rate | < 30 Mbps | > 400 Mbps | > 400 Mbps |

| Parameter | RS-422 | PECL | LVDS* |
|--|------------|---------------|-------------|
| Supply Current Quad Driver (no load, static) | 60mA (max) | 32-65mA (max) | 8.0mA |
| Supply Current Quad Receiver (no load, static) | 23mA (max) | 40mA (max) | 15mA (max) |
| Propagation Delay of Driver | 11ns (max) | 4.5ns (max) | 1.7ns (max) |
| Propagation Delay of Receiver | 30ns (max) | 7.0ns (max) | 2.7ns (max) |
| Pulse Skew (Driver or Receiver) | N/A | 500ps (max) | 400ps (max) |

*LVDS devices noted are DS90LV047A/048A

8.3. Конфигурация систем с интерфейсом LVDS

Наиболее часто LVDS передатчик и приёмник используются в конфигурации «точка-точка», как показано на рис. 38. Однако возможны и другие топологии-конфигурации.

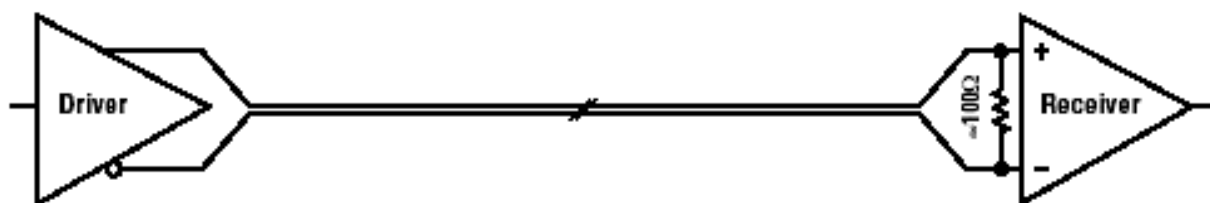


Рисунок 38. Конфигурация «точка-точка»

На рис. 39 приведена топология двунаправленной передачи сигнала через витую пару.

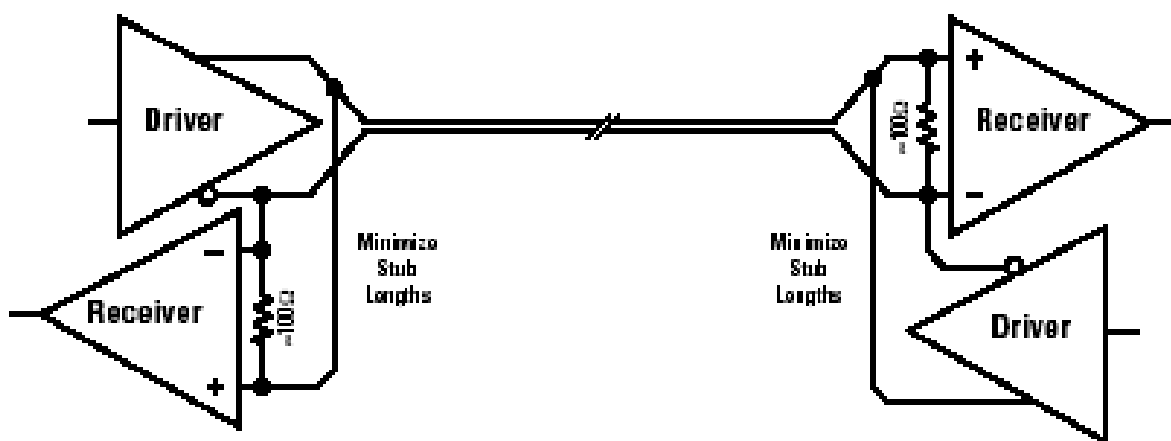


Рисунок 39. Топология для двунаправленной передачи данных

Одновременно данные могут передаваться только в одном направлении. Необходимость в двух терминаторах ослабляет сигналы (и запас по дифференциальным шумам), поэтому данная конфигурация может применяться в случае малых шумов и дальность передачи не превышает 10 метров.

Многоточечная конфигурация объединяет множество приёмников с одним передатчиком (рис. 40). Данная конфигурация встречается в системах распределения информации, а так же в системах с множеством близко расположенных приёмников.

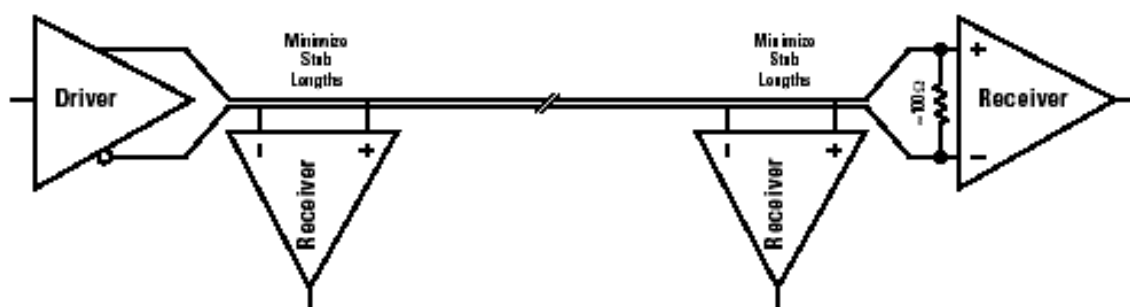


Рисунок 40. Многоточечная конфигурация

Необходимо отметить, что LVDS технология обеспечивает наивысшее качество сигналов в конфигурации «точка-точка», ради которой и

создавалась. Но в целом LVDS имеет множество достоинств и может стать очередным важным стандартом передачи данных со скоростями до сотен мегабит в секунду на небольшие расстояния – до десятков метров.

В этой роли LVDS значительно превышает возможности распространённых кабельных интерфейсов RS-422, RS-232 и RS-485, обеспечивающих пропускную способность ориентировочно от 20 Кбит/с до 30 Мбит/с.

8.4. Согласование сигналов при передаче

Поскольку среда распространения LVDS сигналов состоит из кабеля или двухпроводной линии на печатной плате с легко контролируемым дифференциальным импедансом, то такая линия должна заканчиваться терминатором с импедансом данной линии для завершения токовой петли и подавления искажений коротких импульсов. При отсутствии согласования, сигналы отражаются от несогласованного конца линии и могут интерферировать с другими сигналами. Правильное согласование так же подавляет нежелательные электромагнитные наводки, обеспечивая оптимальное качество сигналов.

Для предотвращения отражений сигнала, LVDS требует применения терминатора в виде простого резистора с расчётным значением сопротивления равным дифференциальному сопротивлению линии распространения. Наиболее часто используется среда с импедансом 100 Ом и соответствующий терминатор. Этот резистор должен располагаться на конце линии передачи, по возможности, на минимальном расстоянии от входа приёмника.

Простота схемы согласования LVDS обеспечивает лёгкость применения интерфейса в большинстве случаев. Похожие по параметрам кабельные интерфейсы ECL и PECL могут потребовать более сложного устройства согласования, чем один резистор. PECL-передатчик обычно

требует подтягивающего к земле резистора 220 Ом у каждого выхода передатчика и резистора 100 Ом на входе приёмника.

8.5. Максимальная скорость переключения в линии

Вопрос максимальной скорости переключения LVDS интерфейса достаточно сложен и ответ на него зависит от нескольких факторов. Этими факторами являются:

- производительность передатчика и приёмника (скорость преобразования поступающей информации в последовательный поток сигналов для передачи информации и обратно – потока битов в N-разрядные слова);
- полоса пропускания среды распространения (кабеля или печатных проводников) в конкретных условиях эксплуатации;
- требуемое качество сигнала в применении.

В общем случае, разработчику встраиваемой системы следует руководствоваться спецификациями на конкретные LVDS-приемопередатчики, доступные на рынке полупроводников, и на определяемые их производителем требования к линии передачи.

8.6. Энергосбережение

LVDS технология обеспечивает сбережение энергии по нескольким направлениям. Мощность, рассеиваемая нагрузкой (терминатор 100 Ом) составляет менее 1.2 мВт. Для сравнения, передатчик RS-422 обеспечивает напряжение 3В на нагрузке 100 Ом, что составляет 90 мВт потребления, что в 75 раз больше чем LVDS. Микросхемы LVDS изготавливаются по КМОП-технологии, благодаря чему имеют малое статическое потребление. Также для примера, схемотехника передатчиков и приёмников LVDS требует не более десятой части потребляемого тока устройств с упомянутым выше интерфейсом PECL/ECL.

Помимо малой рассеиваемой мощности на нагрузке и статического потребляемого тока, LVDS имеет меньшее потребление и благодаря токовому режиму работы схемы передатчика. Эта схема сильно подавляет составляющие тока потребления, зависящие от частоты переключения передатчика. Ток потребления LVDS-передатчика практически не зависит от частоты переключения в диапазоне частот от 10 МГц до 100 МГц, и, например, для счетверённого передатчика составляет не более 50 мА. Для сравнения, передатчик, основанный на технологиях TTL/КМОП, потребляет мощность, возрастающую по экспоненциальному закону от частоты.

8.7. Экономичная аппаратная реализация

Применение LVDS способствует снижению стоимости аппаратной части встраиваемых систем по следующим причинам:

1. Интегральные схемы, как правило, с LVDS изготавливаются по недорогой КМОП технологии.
2. Высокая производительность достижима при использовании дешёвых кабелей марки CAT3 (витая пара, третья категория), соединителей и материалов печатных плат FR4.
3. LVDS требует очень малой мощности источника питания, что снижает количество источников и охлаждающих устройств.
4. LVDS является источником весьма малых шумов и слабо подвержен искажающему воздействию внешних шумов, в том числе, и электромагнитного характера.
5. LVDS приёмники относительно дешёвы и могут быть легко интегрированы на кристалл микросхемы, обеспечивая высочайший уровень интеграции.
6. Поскольку интерфейс LVDS способен передавать информацию значительно быстрее интерфейса на основе технологий TTL/КМОП, то множество TTL/КМОП сигналов могут быть мультиплексированы в один LVDS канал, что исключает

необходимость затрат на дополнительные печатные платы, кабели и соединители (для примера см. рис. 41).

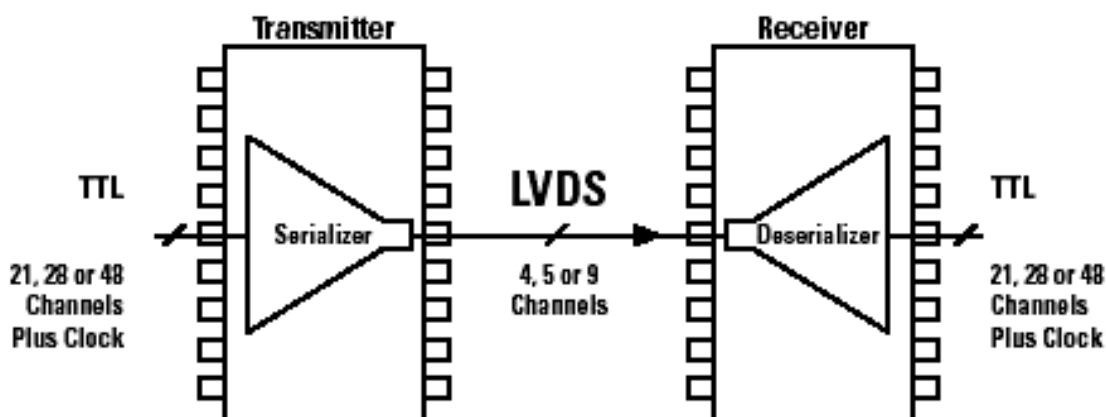


Рисунок 41. Сопряжение интегральных схем передатчика и приемника

Известно, что во многих применениях стоимость дополнительных интегральных схем LVDS значительно ниже стоимости заменяемых ими печатный плат, кабелей и соединителей. Кроме того, отсутствие дополнительных механических деталей упрощает и удешевляет изделие в целом, увеличивает его надёжность.

8.8. Области применения LVDS

Высокая производительность и малые мощность, электромагнитный шум и стоимость LVDS расширяют границы применения интерфейса взамен традиционных технологий.

Ниже приведены такие примеры:

- Периферийные устройства персональных компьютеров;
- Обмен информацией между модулями персональных компьютеров и внутри материнской платы;
- Специализированные сети масштаба CAN (Controller Area Network – сеть масштаба взаимодействия контроллеров);

- Потребительские системы: видеовоспроизводящие устройства, телевизоры, игровые приставки и дисплеи и т.д.;
- Коммуникации внутри специализированных программно-аппаратных комплексов.

8.9. Интегральные схемы с интерфейсом LVDS

В настоящее время многие производители интегральных полупроводниковых схем оснащают свои изделия интерфейсом LVDS, а также выпускают микросхемы, осуществляющие преобразование интерфейсов, с LVDS.

В частности, компания National Semiconductor предлагает LVDS-технологии в нескольких формах. Например, интегральные схемы с напряжением питания 5В DS90C032 и с напряжением питания 3В DS90LV047A/048A – это четверённые линейные приёмопередатчики, позволяющие встроить LVDS-интерфейс в изделия дискретной техники общего назначения. Данные семейства приёмо-передатчиков содержат также одиночные и сдвоенные устройства.

Для соединения переносных компьютеров с LCD панелями высокого разрешения National Semiconductor предлагает микросхемы шины FPD-Link (Flat Panel Display Link – соединение с дисплеями с плоскими панелями) и LDI интерфейса (LVDS Display Interface – LVDS-интерфейс для дисплеев). Эти изделия обеспечивают широкую полосу пропускания, малую потребляемую мощность, малые геометрические размеры и предназначены для мониторов XGA/SXGA/UXGA как для ноутбуков, так и для стационарных компьютеров.

Другое семейство микросхем с LVDS – Channel Link – предназначено для преобразования 21, 28 или 48 бит TTL-данных в 3, 4 или 8 LVDS-каналов с тактовым сигналом. В результате обеспечивается формирование высокоскоростного потока данных (до 5.4Гбит/с) для сверхбыстродействующих сетевых серверов или маршрутизаторов, или везде

где требуются дешёвые, скоростные шины данных. Эти формирователи потоков LVDS везде позволяют экономить затраты на систему за счёт экономии кабелей, соединителей, площади печатных плат.

Шины LVDS являются развитием семейства дискретных линейных LVDS приёмопередатчиков. Они специально разработаны для многоточечных применений, и согласованы с обоими концами линий передачи. Такие формирователи могут использоваться в мощных связных панелях где эффективный импеданс линий может отличаться от 100 Ом в сторону меньших сопротивлений. По этой причине передатчики могут нагружаться на сопротивление до 30-50 Ом. Выходной ток передатчика составляет величину 10 мА для обеспечения необходимого перепада напряжения на такой нагрузке. В данном семействе микросхем доступны также повторители и 10-битные формирователи цифровых LVDS-потоков, которые поддерживают функцию добавления и извлечения тактового сигнала из цифрового потока. Некоторые демультиплексоры цифрового потока впервые в промышленности реализуют функцию случайного ключа данных и функционируют непосредственно от потока, не требуя фазовой автоподстройки частоты.

Такие микросхемы обеспечивают дополнительную функциональность по сравнению с обычными изделиями. Например, изготавливается специальный тактируемый трансивер с шестью КМОП-выходами DS92CK16, анонсирован линейный многоточечный переключатель.

Для получения последних новостей и дополнительной информации по технологии LVDS от компании National Semiconductor можно посетить специальный сайт компании: www.national.com/appinfo/lvds.

9. СПЕЦИФИЧЕСКИЕ ФИРМЕННЫЕ ИНТЕРФЕЙСЫ

При отсутствии ограничений, связанных с правовой защитой разработок конкурирующих на рынке организаций, любой производитель интегральных схем, электронных модулей и систем способен создавать уникальные интерфейсы, наилучшим образом удовлетворяющие его спецификациям.

В подобных случаях интерфейс подробно описывается в технической документации с демонстрацией нюансов подключения и применения, поэтому появление даже незнакомого интерфейса для опытного разработчика не должно существенным образом отразиться на скорости и качестве выполнения проекта.

Гораздо чаще, однако, производители микропроцессоров и микроконтроллеров создают просто универсальные встроенные в кристалл периферийные блоки, которые позволяют выбирать режим их работы и, таким образом, реализовывать различные, хотя и схожие, интерфейсы.

9.1. Универсальные периферийные блоки «на кристалле»

Например, так называемый блок USART (Universal Synchronous Asynchronous Receiver Transmitter – Универсальный Синхронный Асинхронный Приемник Передатчик) в микроконтроллерах фирмы Microchip путем конфигурирования бит управляющего регистра способен исполнять роль полнодуплексного асинхронного приемо-передатчика или «мастера» полудуплексного синхронного приемо-передатчика.

Другой пример – блок USI (Universal Serial Interface – Универсальный Последовательный Интерфейс), которым снабжены некоторые микроконтроллеры «tinyAVR» фирмы Atmel, представляет собой, по сути, сдвиговый регистр (USIDR), доступный в области памяти данных, с регистром управления (USICR) и статуса (USISR) без какого-либо буферирования. С

USI связано аппаратное прерывание. В состав USI входит 4-битовый счетчик тактовых импульсов, управляющийся от регистра USISR [13].

Блок USI можно использовать и в качестве трехпроводного SPI, и для имитации асинхронной последовательной передачи данных, и еще для ряда применений (например, тактовый счетчик совместно с таймером №0 может образовывать 12-разрядный таймер; с помощью USI можно организовать дополнительное внешнее прерывание и т. п.). Управляется сдвиг в регистре USIDR либо пользовательской программой, либо от прерывания переполнения таймера №0, либо от внешнего источника (что допускает функционирование устройства с USI в качестве ведомого). С USI связаны три внешних вывода микроконтроллера – вход и выход данных (DI, DO) и ввод-вывод тактовых импульсов (USCK).

К числу недостатков USI относится то, что он присутствует лишь в небольшом числе моделей, поэтому программы с его использованием плохо переносятся. В большинстве случаев функции USI можно заменить программной имитацией, что в данном случае делает программиста независимым от конкретных моделей микроконтроллеров AVR.

9.2. Интерфейс UNI/O

Показательным примером специфического фирменного внутриплатного интерфейса для встраиваемых систем является шина UNI/O, разработанная и запатентованная фирмой Microchip. [14]

UNI/O – это асинхронный однопроводной интерфейс, использующий манчестерский код для сигнальной передачи. В манчестерском коде передаваемые нули и единицы кодируются перепадами логического уровня от высокого к низкому («0») и от низкого к высокому («1») в середине битового интервала («bit period»). Благодаря данному методу кодирования обеспечивается взаимная синхронизация устройств, передающих информацию, т.к. приемник способен извлечь синхронизирующий сигнал с целью корректного декодирования битовой последовательности. Отсутствие

перепада внутри битового интервала приводит к прекращению передачи и переходу принимающего устройства в режим ожидания.

9.2.1. Схема взаимодействия устройств и адресация

Так же, как во многих других интерфейсах, взаимодействие в UNI/O происходит по схеме «ведущий-ведомый» («master-slave») с одним master-устройством и одним или многими slave-устройствами. И master-устройство, и slave-устройство могут выступать в качестве приемника и передатчика. Однако текущее направление передачи при обмене информацией определяет только master-устройство.

Для обеспечения возможности работы со многими slave-устройствами передачу любой команды предваряет адрес, состоящий из восьми или двенадцати бит.

В восьмибитном адресе старшие четыре бита указывают на семейство адресуемых устройств (см. табл. 7), а четыре младших – на адрес устройства в этом семействе.

В случае двенадцатибитной адресации в действительности поле адреса занимает шестнадцать бит – два байта, однако старшие четыре – устанавливаются в единичное состояние, что указывает приемнику на применение расширенного варианта адреса. Далее младшие четыре бита первого байта соответствуют семейству, и, наконец, весь второй байт адреса позволяет обратиться к конкретному устройству.

Таблица 7. Описание семейств устройств для интерфейса UNI/O

| Код семейства | Описание семейства |
|---------------|---|
| 0000 | Зарезервировано |
| 0011 | Контроллеры дисплеев |
| 0100 | Расширители портов |
| 1000 | Частотные, квадратурные контроллеры, контроллеры широтно-импульсной модуляции, часы реального времени |
| 1001 | Температурные датчики |

| | |
|------|---|
| 1010 | Электрически-перепрограммируемая память |
| 1011 | Устройства шифрования и идентификации |
| 1100 | Преобразователи напряжения |
| 1101 | Аналого-цифровые преобразователи |
| 1111 | Указание на применение двенадцатибитного адреса |

9.2.2. Команды интерфейса и формат передачи данных

В спецификации интерфейса UNI/O определены несколько команд или, иными словами, специальных сигналов:

1. Импульс перехода в режим ожидания («Standby pulse»).

Реализуется в виде длинного импульса, чтобы перед началом передачи или после сбоя быть уверенным, что устройства находятся в режиме ожидания и могут принимать информацию. Обязателен при первом обмене информацией после включения питания. Может не использоваться между сеансами обмена с одним и тем же устройством.

2. Начальный заголовок («Start header»).

С помощью начального заголовка master-устройство устанавливает скорость передачи (и, соответственно, длительность битового интервала). Определенная длительность битового интервала запоминается и не изменяется до следующего начального заголовка. При этом частота извлеченного из потока сигнала синхронизации не должна выходить из диапазона от 10КГц до 100КГц.

Реализация начального заголовка осуществляется выполнением следующих действий в линии:

- выставляется низкий уровень, как минимум на 5мкс;
- передаётся последовательность из восьми бит: 01010101;
- генерируется сигнал подтверждения от master-устройства («МАК», см. ниже);

- реализуется сигнал «SAK» (см. ниже) во избежание коллизии на линии по причине одновременного ответа со стороны нескольких slave-устройств.

3. Подтверждение («Acknowledge»).

В интерфейсе определено несколько сигналов, подтверждающих или не подтверждающих выполненное действие. Среди них:

«MAK» (Master Acknowledge) – реализуется после каждого переданного байта информации, как единичный бит, отправленный в специально отведенном для этого сигнала битовом интервале;

«NoMAK» (No MAK) – реализуется, как нулевой бит и используется для прерывания текущей операции, а также для инициации цикла записи в соответствующих командах;

«SAK» (Slave Acknowledge) – единичный бит в специально отведенном битовом интервале, сигнализирующий master-устройству от slave-устройства об успешном приёме байта информации;

«NoSAK» (No SAK) – реализуется, как отсутствие изменения линии каким-либо из устройств в течение битового интервала после начального заголовка и для сигнализации об ошибочном состоянии. В последнем случае после получения «NoSAK» master-устройство обязано сгенерировать импульс перехода в режим ожидания.

Прохождение сигналов «MAK» и «SAK» указывает на наличие взаимной синхронизации master-устройства и slave-устройства.

После генерации импульса перехода в режим ожидания, начального заголовка и адреса slave-устройства (см. рис. 42) master-устройство должно передать восьмибитное значение, содержащее код команды, которую slave-устройство должно исполнить. Множество доступных команд определяет разработчик slave-устройства в зависимости от семейства, к которому данное устройство принадлежит. Разработчик также определяет необходимые дополнительные данные (или поток данных), поступающие сразу после кода команды от master-устройства к slave-устройству или наоборот.

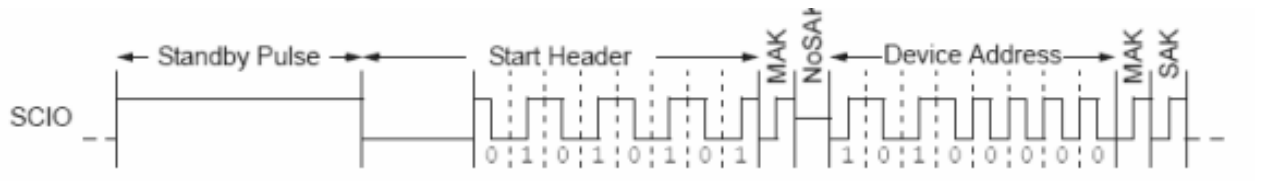


Рисунок 43. Пример начальной части временной диаграммы и формат передачи по интерфейсу UNI/O

ЗАКЛЮЧЕНИЕ

Современный рынок электронных компонентов с каждым днём открывает всё больше и больше возможностей для разработчика встраиваемых систем различного применения. Сохраняя определенный круг задач, при решении которых достаточно лишь несложного контроллера с ограниченным числом входов и выходов, инженеры и программисты, тем не менее, сталкиваются с необходимостью решать всё более сложные задачи. В техническом смысле это выражается в увеличивающихся пропускных способностях внешних и внутренних для системы каналов передачи информации, усложняющихся алгоритмах обработки и расширяющейся разнородности интерфейсов.

Разрабатываемые производителями новые интерфейсы, как правило, призваны удовлетворить появляющиеся функциональные требования, которые существующие интерфейсы решить уже не в состоянии. Реже таким образом решаются также вопросы соблюдения прав интеллектуальной собственности.

И хотя к данному моменту уже сформировался достаточно устойчивый набор внутрисплатных интерфейсов, применяемый в любой более или менее сложной системе встраиваемого управления (как например, I²C и SPI) проектировщик встраиваемых систем сегодня непрерывно возвращается к спецификациям на новые интерфейсы, выпускаемым рабочими группами производителей электронных компонентов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. ГОСТ Р 50304-92. Системы для сопряжения радиоэлектронных средств интерфейсные. Термины и определения. – Москва: Госстандарт России, 1992.
2. Николайчук О. Особенности микроконтроллерных архитектур с интерфейсом SPI // Схемотехника. 2005. №12
3. URL: http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus (дата обращения: 30.08.2011).
4. URL: <http://en.wikipedia.org/wiki/Microwire> (дата обращения: 30.08.2011).
5. Семенов Б.Ю. Шина I²C в радиотехнических конструкциях. - М.: Солон-Р, 2002. - 190 с.
6. Гук М. Аппаратные интерфейсы ПК. Энциклопедия. - СПб.: Питер, 2002. - 528 с.
7. URL: <http://схем.net/comp/comp53.php> (дата обращения: 31.08.2011).
8. Ан П. Сопряжение ПК с внешними устройствами: Пер. с англ. - М.: ДМК Пресс, 2001. - 320 с.
9. I²S bus specification / Philips Semiconductors / February 1986, Revised: June 5, 1996.
10. Анисимов А.Л., Астапкович А.М., Востриков А.А., Сергеев М.Б. Введение в смарт-технологии. // Методические указания к выполнению лабораторных работ. – СПб.: ГУАП, 2000. – 58 с.
11. Сотни мегабит на сотни метров: увеличение дальности передачи для LVDS / КОМПОНЕНТЫ И ТЕХНОЛОГИИ • № 3 '2007
12. URL: www.national.com/appinfo/lvds/ (дата обращения: 30.08.2011).
13. URL: <http://controllersystems.com/static/soderjanie.html> (дата обращения: 31.08.2011).
14. URL: <http://en.wikipedia.org/wiki/UNI/O> (дата обращения: 31.08.2011).

Учебное издание

Востриков Антон Александрович
Балонин Николай Алексеевич
Сергеев Александр Михайлович

.....

ВНУТРИПЛАТНЫЕ ИНТЕРФЕЙСЫ ВСТРАИВАЕМЫХ СИСТЕМ

Учебное пособие

Редактор.....

Компьютерная верстка.....

| | | |
|-----------------------------------|------------------------|--------------------|
| Сдано в набор | Подписано к печати | Формат |
| Бумага офсетная. Печать офсетная. | Усл.печ.л. Уч.-изд. л. | Тираж экз. Заказ № |

Отдел электронных публикаций и библиографии библиотеки
Отдел оперативной полиграфии
ГУАП

190000, Санкт-Петербург, ул.Б.Морская, 67