

Продолжение. Начало в № 3 `2009

Синхронный последовательный интерфейс SPI

в микроконтроллерах «от А до Я» и его реализация в ADuC70xx фирмы Analog Devices

Александр НОВИЦКИЙ

Общая характеристика подсистемы SPI в составе МК

В [2] упоминается, что около 85% выпускаемых МК (на 2005 г.) оснащены интерфейсом SPI. Практика показывает, что реализации SPI в МК могут различаться во многих деталях, что не лишает их совместимости, но затрудняет портирование ПО с одного МК на другой. Рассмотреть сколько-нибудь значительную часть этих реализаций в рамках статьи не представляется возможным, поэтому далее эти различия обсуждаются «в общем», но со ссылками на конкретные МК. В качестве «референтных» автор выбрал, в первую очередь, реализации прародителя SPI — фирмы Motorola, фирм Atmel и MicroChip, как крупнейших производителей МК, а кроме того, модели Analog Devices, поскольку их рассмотрение было первоначальной мотивацией написания статьи.

Подсистема SPI в составе микроконтроллера (МК) должна обеспечить диаграммы сигналов, которых могут требовать периферийные микросхемы, а как мы сообщили в предыдущей статье, набор этих требований может быть весьма разнообразным. Поэтому SPI в МК представляет собой достаточно сложную программно управляемую подсистему со многими настройками. Обычно в состав регистровой модели SPI входит регистр управления, и каждой из этих настроек соответствует битовое поле, либо отдельный бит в этом регистре.

Сигналы SPI в МК

Подсистема SPI в составе МК использует все четыре сигнала интерфейса — MOSI, MISO, SCK и SS, соответствующие выходы

в подавляющем большинстве МК многофункциональные и могут быть использованы другими подсистемами МК, в первую очередь и чаще всего портами параллельного ввода/вывода. А при построении симметричной магистрали с несколькими ведущими приходится использовать еще и дополнительные сигналы.

Включение/отключение подсистемы SPI

Отключенное состояние SPI обеспечивает уменьшение энергопотребления. При включении инициализируются внутренние узлы SPI: тактовый генератор, регистр состояний и ряд других. В некоторых МК выходы, используемые SPI, подключаются к ней автоматически, в то время как в других МК программист должен сконфигурировать мультиплексор внешних выводов дополнительными командами. При этом могут понадобиться дополнительные настройки, в частности, направление передачи и режим работы вывода (комплементарный выход либо «открытый сток»), а также состояние встроенных pull-up или pull-down резисторов на выводах.

В зависимости от режима SPI некоторые выходы являются выходами, и эта настройка может конфликтовать с настройкой выводов, относящихся к режиму параллельного ввода/вывода. В некоторых МК активация подсистемы SPI автоматически устанавливает нужное направление передачи для выводов, в то время как в других МК программист в ряде случаев должен выполнить дополнительные действия по настройке направления передачи.

С целью обезопасить выходы от повреждения, если вывод в режиме SPI должен быть входом, это состояние устанавливается при

активации SPK автоматически. Если же вывод должен в режиме SPI быть выходом, то для этого необходимо бывает после активации SPI дополнительно выполнить настройку направления, предназначенную для параллельного ввода/вывода.

Выбор режима ведущего либо ведомого

Интерфейс SPI может использоваться не только для подключения к МК периферийных устройств, но и для коммуникации между двумя (или несколькими) МК. Для этого подсистема SPI в одном из МК должна быть установлена в режим ведомого, что может быть сделано переключением специального бита в регистре управления SPI. При переключении из режима ведущего в режим ведомого спецификация предусматривает изменение функций выводов SPI (см. далее).

В режиме ведомого вывод MISO является выходом, в то время как выходы MOSI, SCK, SS — входы. Если МК является единственным ведомым (связаны между собой два МК), то его вывод MISO связан с одноименным входом второго (ведущего) МК, каких-либо конфликтов не происходит. Если же необходимо связать через SPI несколько МК, то необходимы дополнительные меры во избежание электрических и логических конфликтов, вопрос связи между собой более чем двух МК будет обсуждаться подробно далее.

Выбор режима тактирования

Особенности четырех режимов тактирования в SPI были рассмотрены в [1], они различаются полярностью и «фазой» тактирующих импульсов. На рис. 1 (заимствован из

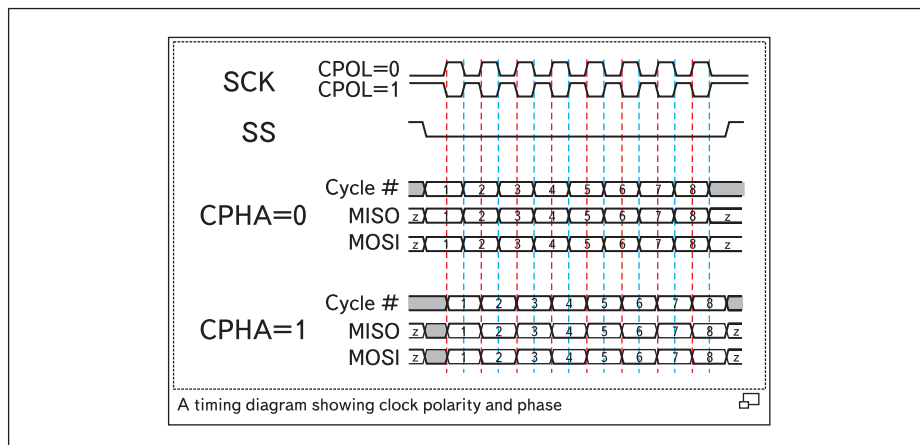


Рис. 1. Временные диаграммы, иллюстрирующие четыре режима тактирования в SPI (значения сигналов в интервале, когда SS активен, а сдвиг не происходит, в описании Википедии считаются неопределенными)

Википедии) изображены диаграммы, иллюстрирующие особенности этих режимов. Их основное различие заключается в порядке действий «сдвиг» (Shift) и «захват» (Latch). Этот порядок для простейших применений может быть безразличным, как, в частности, при использовании в качестве ведомого универсального сдвигового регистра (в [1] был рассмотрен пример подключения регистра 74НС164 к МК через SPI).

Следует заметить, что, хотя в большинстве описаний SPI предполагается, что передача данных осуществляется побайтно, тем не менее, существуют реализации ведущих, допускающие передачу пакетов иной длины. Так, например, в микроконтроллере Blackfin от Analog Devices можно выбирать между форматами 8 и 16 бит, а в спецификации [8] предусмотрена возможность задавать длину пакета любой, от 8 до 16 бит включительно.

Задание в МК полярности тактов

Во всех известных автору реализациях оно производится переключением бита (чаще всего именуемого как CPOL) в регистре управления SPI. Значение CPOL соответствует уровню сигнала тактирования в промежутках между пакетами, а каких-либо дополнительных функций кроме сдвига и захвата битов данных на этот сигнал не возложено. Названия этого и следующего управляющих битов являются общепринятыми.

Задание в МК фазы тактирования

Для этого обычно служит бит CPHA в регистре управления SPI. Он определяет порядок следования операций «захвата» и «сдвига». При CPHA = 0 сначала, по первому перепаду сигнала SCK (и всем последующим нечетным, на рис. 1 они отмечены красным) происходит «сдвиг» в регистрах, а по четным перепадам (отмечены синим) — «захват» сигналов на входах сдвиговых регистров. При CPHA = 1 порядок действий обратный, то есть по пер-

вому и нечетным перепадам тактов происходит захват сигналов, а по четным — сдвиг. Весьма важно учитывать (в частности, при программной эмуляции SPI), что в режиме CPHA = 1 сигналы, соответствующие первому передаваемому биту, должны появиться на выходах MISO и MOSI, как только сигнал выбора ведомого перешел в активный (низкий) уровень.

Для большинства ведомых SPI-устройств порядок действий «сдвиг» и «захват» оказывается существенным, поскольку чаще всего эти действия должны быть разнесены во времени. Более того, в ряде случаев ведомый может требовать более сложной диаграммы (в [1] приведен пример с подключением микросхем АЦП MAX1240/1241). Для реализации подобной сложной диаграммы необходимо знать, как ведет себя сигнал MOSI в ведомом устройстве на интервале времени, вне интервала передачи пакета. Подобная информация не всегда содержится в технических описаниях МК в понятном и достоверном виде, и нередко ее можно получить, лишь экспериментируя с микроконтроллером.

Спецификации фирмы Motorola [3] и более поздняя от Freescale [4] регламентируют, что при CPHA = 0 для передачи пакетов от ведомого к ведущему (по линии MISO) сигнал SS в промежутках между пакетами обязательно должен на короткое время возвращаться в неактивное (высокое) состояние (старт-стопная передача), в то время как при CPHA = 1 сигнал SS может непрерывно оставаться активным (низким) в течение последовательной передачи нескольких пакетов (непрерывная передача).

Запуск передачи в ведущем устройстве

Как и во многих других последовательных интерфейсах, он происходит автоматически после того, как программа поместит очередной передаваемый пакет в передатчик SPI. При этом передача данных происходит од-

новременно и синхронно в обоих направлениях: сдвиговые регистры ведущего и ведомого обмениваются содержимым.

Однако в ряде случаев не требуется передачи данных в одном из направлений, например, при управлении микросхемами ЦАП надо только передавать, а при чтении данных из внешней Flash-памяти с SPI-интерфейсом требуется только принимать считанный блок байтов. В последнем случае для приема каждого байта придется сначала произвести «холостую» запись в буфер передатчика, а после окончания обмена пакетов прочитать очередной байт (и затем снова произвести «холостую» запись для запуска очередного обмена).

В некоторых реализациях SPI имеется возможность включения режима, в котором запуск очередного обмена происходит вследствие считывания ранее принятого пакета. Это позволяет избежать лишних затрат времени на «холостую» запись.

Способы и варианты формирования в ведущем МК сигнала выбора ведомого SS

Наиболее простой и часто рекомендуемый способ — формировать сигнал SS программно на отдельном выходе параллельного порта МК. В этом случае поведение сигнала SS во времени целиком определяется программистом, но на переключение SS расходуется процессорное время, и возможно снижение скорости передачи по сравнению с максимально достижимой (см. далее обсуждение вопросов, связанных со скоростью передачи). Такое программное формирование сигнала SS необходимо в ранее рассмотренном примере [1] с подключением через SPI микросхем ЦАП MAX1240 / MAX1241 фирмы Maxim [7].

В ряде МК [9, 11] сигнал SS формируется автоматически (аппаратно) на специально предназначенном выводе. Встречается два режима автоматического формирования SS: старт-стопный и непрерывный.

При старт-стопном режиме после записи пакета в SPI-передатчик сигнал SS переключается в активный (низкий) уровень, и после небольшой задержки, обычно в полпериода битовой частоты, начинается формирование пакета тактовых импульсов. По окончании пакета, также с небольшой задержкой, сигнал SS возвращается в состояние 1.

При непрерывном режиме после окончания передачи пакета возврат сигнала SS в пассивное состояние может отсутствовать, если программа «успешно» поместит в буфер передатчика очередной передаваемый пакет так, что промежутка между пакетами вовсе не будет. Этому может способствовать наличие буферизации передатчика, что мы обсудим позже. Как уже было упомянуто, непрерывный режим работы возможен только для режимов с CPHA = 1.

Режим непрерывной передачи пакетов необходим для управления микросхемами (мо-

дулями) флэш-памяти [5] либо ЦАП [6], соответствующие примеры рассмотрены в [1]. В некоторых реализациях МК программист имеет возможность выбора одного из описанных способов формирования сигнала SS.

Задание скорости следования битовых интервалов

Основное назначение SPI — обеспечить быстрый интерфейс с периферийными модулями. Максимальные значения скорости передачи (битовой частоты) практически во всех реализациях превышают 1 МГц, а в некоторых достигают 50 МГц и даже выше. Вместе с тем, для конкретных периферийных ИМС требуется вполне определенная скорость обмена (хотя в ряде случаев она может понижаться без ущерба для функционирования периферийного устройства). Поэтому во всех МК, имеющих SPI, его битовая частота сделана перестраиваемой. Принцип перестройки везде один и тот же: $F_{SCK} = F_G / DIV$, где F_G — постоянная частота задающего генератора SPI, а DIV — делитель, который может быть задан программно. Значение делителя в одних МК можно выбрать лишь из ограниченного ряда предопределенных значений, в других МК делитель может быть любым в пределах ограниченной разрядности.

Частота F_G практически во всех МК близка к частоте тактирования процессора, таким образом, и максимальная частота битовых интервалов может быть достаточно высокой. В таблице 1 приведены соотношения частоты тактирования процессора и битовой частоты SPI для нескольких типов МК.

Судя по данным таблицы 1, при максимальной скорости SPI, за время передачи бита процессор МК выполнит всего лишь несколько команд (от 1 до 10). В ряде случаев при выбранной высокой битовой частоте SPI может оказаться, что по окончании передачи пакета программа «не успеет» вовремя загрузить в буфер передатчика следующий пакет. Поэтому при разработке системы приходится достаточно внимательно относиться к выбору скорости на входе.

Скорость приема данных в МК в режиме ведомого может превышать скорость в режиме ведущего

Тактирование SPI ведомого устройства происходит внешней частотой, поступающей по линии SCK от ведущего. Если в качестве ведомого используется также МК, то следует учитывать еще два обстоятельства: во-первых, частота (внешнего) тактирования SPI может быть никак не синхронизирована с внутренними процессами в ведомом МК, и, во-вторых, она может превышать значение, которое допустимо при работе в режиме ведущего. Например, для МК семейства ADuC70xx в режиме ведомого обмен можно производить на

Таблица 1. Соотношения частоты тактирования процессора и битовой частоты SPI

Микроконтроллер	Производитель	Макс. частота CPU, МГц	Время выполнения команды	Макс. битовая частота SPI, МГц	Источник
M68HC11	Motorola	2	≥ 1 мкс	1	[3]
M68HC12	Motorola	8	≥ 125 нс	2	[10]
ADuC70xx	Analog Devices	41,78	≥ 24 нс	3,482	[11]
ATMmega64	Atmel	8–12	1–3 такта	$F_{CPU}/2$	[12]
PIC18Fxxx	MicroChip	10	≥ 100 нс	2,5	[14]

частоте до 10,4 МГц, а в режиме ведущего — лишь до 3,482 МГц [11].

Сопряжение процесса передачи в SPI с работой программы в МК

Программа в МК может «узнать» о состоянии передатчика (об окончании передачи пакета), либо опрашивая флаг в регистре состояния SPI, либо по аппаратному прерыванию. В обоих случаях время реакции от момента установки флага (момента возникновения запроса прерывания) до записи следующего пакета в регистр данных может (при максимальной частоте следования битовых интервалов) превысить длительность одного бита. В ведущем устройстве такое запаздывание приведет к появлению промежутка между пакетами, а вот в ведомом МК программа может просто «не успеть» поместить следующий пакет.

Обнаружив установку флага, свидетельствующую об окончании обмена пакетами, МК должен, во-первых, прочитать принятый пакет из регистра данных и, во-вторых, поместить в регистр данных следующий передаваемый пакет — на ведущем МК: это тут же запустит его передачу. При высокой частоте битовых интервалов эти два действия могут потребовать времени, которое превысит длительность одного битового интервала. На ведущем МК это приведет лишь к появлению промежутка между передачами соседних пакетов, а вот ведомый МК может просто не успеть выполнить и считывание, и запись. Исправить положение может как понижение битовой частоты, так и намеренное введение в программу ведущего промежутка между пакетами.

Структура с двойной буферизацией (double-buffering)

Для решения этой проблемы в SPI-приемопередатчик вводят дополнительные буферные регистры (структура «с двойной буферизацией»). По окончании приема пакета принятый байт немедленно (аппаратно) копируется в дополнительный регистр (буфер приема), сдвиговый регистр может без промежутка вести прием следующего пакета, а у программы есть время для считывания принятого пакета из буфера, вплоть до окончания приема следующего пакета (на порядок больше длины битового интервала). Это

позволяет, в частности, менять порядок действий: по окончании приема пакета сначала поместить в передатчик следующий передаваемый пакет, а только затем считать принятый пакет из буфера.

В ранних реализациях SPI в МК фирмы Motorola (M68HC11) двойная буферизация была сделана на прием и отсутствовала на передачу [3]. В более поздних реализациях, как от Motorola, так и от других производителей, двойная буферизация имеется как на передачу, так и на прием. Это дает программе время для записи нового пакета в буфер передачи и последующего чтения принятого пакета из буфера приема, равное времени передачи пакета (а не длительности одного битового интервала).

Двойная буферизация: увы, иногда придется экспериментировать...

В технических описаниях некоторых МК об особенностях двойной буферизации говорится весьма невнятно. Так, например, в описании МК PIC16F87x от MicroChip [14] из описания можно понять, что в SPI имеется лишь один буфер, который используется как для приема, так и для передачи. Так заставляет думать и тот факт, что для доступа к буферу по записи и по чтению используется один и тот же адрес¹. Такая архитектура возможна, если схемотехника по окончании передачи пакета выполняет обмен содержимого буфера и сдвигового регистра, однако из описания это никак не следует. В этом случае программа после окончания приема очередного пакета сначала должна считать принятый (предыдущий) пакет, а затем поместить в буфер пакет, который будет передаваться следующим.

В реальности же во многих известных автору реализациях физически имеется два отдельных буферных регистра — на передачу и на прием. При такой архитектуре порядок обращения к буферам может быть любым: сначала чтение принятого, потом запись следующего передаваемого, или наоборот. В варианте же с единственным буфером обязательно следует сначала выполнить чтение принятого, а затем запись следующего передаваемого. Узнать о том, как сделана буферизация, зачастую можно лишь экспериментально.

¹ Так сделано и в COM-порте стандартного ПК: приемник и передатчик имели двойную буферизацию, а с некоторых пор (NS16550) имеют и буферы FIFO по 16 байт, хотя обе очереди данных «видны» программисту через один и тот же адрес: передатчик — по записи, приемник — по чтению.

Развитие идеи буферизации

В некоторых МК вместо одиночных буферных регистров имеются аппаратные «очереди» FIFO. Это позволяет программе максимально быстро записать в «очередь» подряд несколько пакетов, которые затем будут передаваться в непрерывном режиме, с одновременным аппаратным помещением принимаемых пакетов в «очередь» приемника. В такой архитектуре программе приходится «отвлекаться» для чтения пачки принятых пакетов лишь незадолго до заполнения очереди.

Фирма FreeScale в 1996 г. выпустила документ [8], в котором специфицирована структура интерфейса SPI с 16-байтовым буфером-очередью. Буфер организован в участке памяти, доступ к нему использует идеологию ПДП.

В микропроцессорах семейства Blackfin фирмы Analog Devices [9], в подсистеме SPI реализованы буферы-очереди как на передачу, так и на прием, а кроме того, имеется механизм заполнения либо опустошения «очереди» с использованием механизма прямого доступа к памяти (только в одном направлении, но не в обоих сразу!). Таким образом, программа имеет возможность, заранее запрограммировав ПДП-контроллер, не отвлекаться на обслуживание подсистемы SPI: заданное количество пакетов будет передано из указанной области памяти (либо принято) полностью автоматически, без оперативного участия программы.

Организация связи между несколькими МК через SPI

Подсистема SPI в микроконтроллере (далее МК) предназначена главным образом для работы в качестве ведущего устройства при подключении периферийных микросхем в качестве ведомых. Однако в практике нередки случаи, когда желательно связать между собой два (или даже несколько) МК, либо иметь возможность управлять группой периферийных микросхем от нескольких различных ведущих. Например, система может включать МК общего назначения, а кроме того, цифровой сигнальный процессор DSP и оба процессора должны иметь доступ к периферийным узлам.

Именно в расчете на такие случаи спецификация SPI дает возможность организовать симметричную информационную магистраль с несколькими ведущими устройствами.

К магистральной может быть подключено два (или больше) МК, каждый из которых может быть либо ведущим, либо ведомым, когда с ним связывается другой МК, работающий в режиме ведущего. Одноименные выходы всех МК в такой магистральной связаны между собой четырьмя линиями, как показано на рис. 2. Обычно один из битов в регистре управления SPI позволяет программно включить в МК режим ведущего либо ведомого.

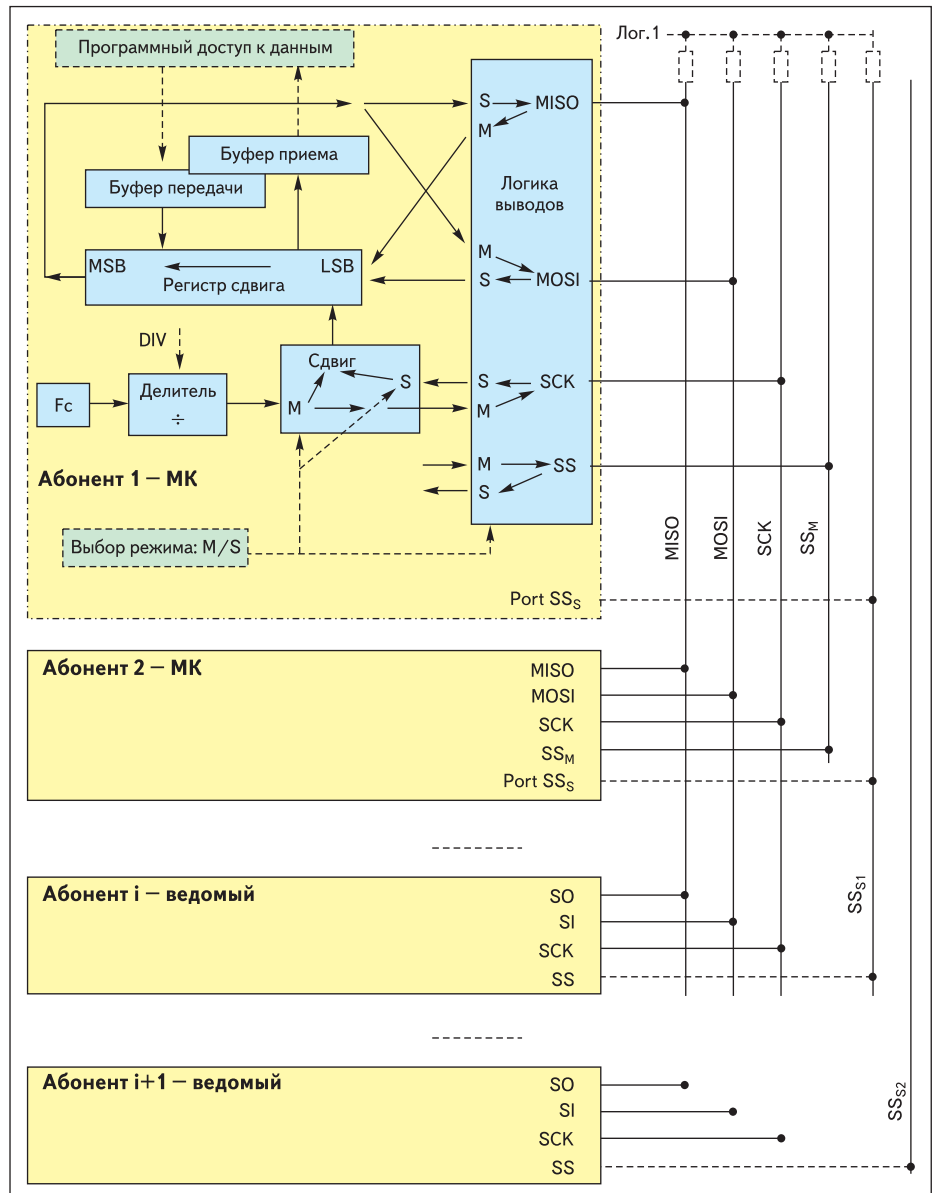


Рис. 2. Структура магистрали со многими ведущими на базе SPI

Основное свойство линий SPI, позволяющее осуществить такое соединение, состоит в том, что при переключении SPI в МК из режима ведущего в режим ведомого автоматически изменяется направление передачи выводов. В режиме ведущего (Master) выходы MOSI, SCK, SS являются выходами, в то время как MISO — вход. Если же абонент находится в режиме ведомого (Slave), то выходы MOSI, SCK, SS_M работают в режиме входов, а MISO — в режиме входа. Функции выводов отражаются в их названиях: например, MOSI означает if Master then Output, if Slave then Input. На рис. 2 для одного из МК отображена внутренняя структура приемопередатчика SPI и там же (в блоках «Логика выводов» и «Сдвиг») буквами М и S отмечены связи, возникающие в режимах Master и Slave. Изменение функций выводов показано в таблице 2.

Особо отметим, что если к SPI-магистральной, кроме МК, подключены еще и периферийные

ИМС, которые всегда являются ведомыми, то для управления каждым из них придется организовать отдельную линию SS_S . Два таких устройства и соответствующие линии SS_S показаны на рис. 2. Если этого не сделать, то периферийный модуль будет активироваться при обмене между МК и создавать конфликт на магистральной. Для управления линиями SS_S придется в МК использовать выходы портов общего назначения.

Таблица 2. Изменение функций выводов SPI при переключении режима «ведущий—ведомый»

Название сигнала	Режим	
	Ведущего	Ведомого
MOSI	МО — выход	СИ — вход
MISO	МИ — вход	СО — выход
SCK	выход	вход
SS	выход/вход	вход

Проблема коллизий на магистрали SPI

На любой из линий магистрали возможно возникновение электрических коллизий, заключающихся в том, что более чем один вывод (с комплементарной схмотехникой) из подключенных к одной линии находится в режиме «выход», и разные выходы пытаются передавать в линию различные значения логического сигнала. Два таких выхода образуют «почти» короткое замыкание (latchup) источника питания, это может быть чревато физическим повреждением выходных транзисторов комплементарных каскадов. Даже если не произойдет физического повреждения, будет иметь место логическая коллизия, при которой значения принимаемых байтов могут отличаться от передаваемых. Вопрос разрешения коллизий уже рассматривался в первой части данной статьи [1] при обсуждении широкополосного режима обмена в радиальной структуре связи на базе SPI.

Для исключения электрических и логических коллизий необходимо, чтобы в любой момент времени лишь один из выводов, подключенных к линии магистрали, находился в режиме выхода, либо одновременно активные выходы имели бы схмотехнику «открытый сток», гарантирующую отсутствие электрических коллизий, и алгоритмы программного управления абонентами исключали бы возникновение логических коллизий. В магистрали на базе SPI чаще используется схмотехника выходов с открытым стоком, и некоторые МК предоставляют программисту простую возможность перевода выводов SPI в режим «с открытым стоком». Например, в микроконтроллерах семейств M68HC11 [3], разд. 8.6.2) единственный бит (DWOM) в регистре управления SPI позволяет включить режим «открытого стока» одновременно для всех выводов SPI и тем самым исключить возможность возникновения электрических коллизий. Однако при этом потребуются подключение внешних pull-up резисторов на каждую из линий магистрали (на рис. 2 показаны пунктиром). Надо заметить, что возможность переключения выводов SPI в режим «открытого стока» имеется не во всех МК.

В правильно сконфигурированной магистрали только один из МК-абонентов на время обмена переходит в режим ведущего. Чтобы не пропустить его вызов, все МК в отсутствии обмена должны находиться в режиме ведомого. Если один из абонентов перешел в режим ведущего, а затем начал обмен и перевел свой вывод SS_M в низкий (активный) уровень, то все прочие ведомые активируются, и возможны логические коллизии на линии MISO (все остальные выводы абонентов-ведомых являются входами). Появление активного (низкого) уровня на линии SS_M является сигналом всем прочим абонентам, что кто-то перешел в режим ведущего и намерен начать обмен по магистрали.

Для исключения коллизий все ведомые МК (может быть, за исключением одного) должны держать свои выводы MISO в отключенном состоянии, как это уже обсуждалось в первой части при описании радиальной структуры связи. То есть разработчику придется самому спроектировать протокол-надстройку, предусматривающий адресацию и передачу от ведущего МК к одному из ведомых МК команды, в результате которой тот включит свой вывод MISO и будет способен передавать данные ведущему, а все прочие ведомые могут принимать данные от ведущего, но не должны активировать свои выводы MISO.

Автоматическая детекция логических коллизий на магистрали SPI

Однако в такой организации магистрали возможно возникновение логической коллизии, когда два (или более) абонента пытаются переключиться в режим ведущего почти одновременно. Все абоненты должны как-то непрерывно «слушать» линию SS_M и делать такую попытку лишь в случае, если сигнал SS_M — высокий. При (почти) одновременном анализе сигнала SS два абонента могут получить результат «свободно», и оба переключиться в режим ведущего. После этого возникновение коллизии возможно уже на любой из трех линий SCK, MOSI, SS_M , которые для ведущих являются выходами. Распознать такую ситуацию программно оказывается весьма сложно, если не невозможно.

Для выявления описанной ситуации уже в ранних реализациях SPI фирмы Motorola [3] имелась аппаратная поддержка разрешения коллизий между ведущими. Она заключается в том, что вывод SS у МК, который перешел в режим ведущего, может находиться в особом состоянии, когда он является входом. Если второй МК одновременно с данным перешел в режим ведущего и успел перевести линию SS_M в активное состояние (низкий уровень), то есть пытается начать обмен, а вывод SS в первом МК-ведущем еще находится в упомянутом особом состоянии входа, то этот факт распознается в первом абоненте аппаратно, и далее в нем генерируется «ошибка режима» (Mode Fault): немедленно автоматически выполняется следующая последовательность действий.

- Все выводы, связанные с портом SPI, переводятся в режим входа.
- Схмотехника SPI переводится в режим ведомого.
- Подсистема SPI отключается (disabled).
- Устанавливается флаг, сигнализирующий о возникшей коллизии.
- Генерируется (и происходит, если оно разрешено) прерывание от подсистемы SPI².

Теоретически возможна ситуация, когда переход двух абонентов в состояние ведущего укладывается в очень узкое временное окно, соизмеримое со временем срабатывания логического элемента (для современных МК оно составляет единицы наносекунд). В этом случае два абонента могут выполнить и переход в режим ведущего, и активацию сигнала SS строго синхронно, а уже затем при передаче данных все-таки возникнет (логическая) коллизия, но вероятность этого очень мала.

О формировании сигнала SS_M на симметричной магистрали SPI

Скажем отдельно о формировании сигнала SS_M на такой симметричной магистрали. Если МК находится в режиме ведомого, его вывод SS является входом и управляется внешним сигналом, который должен формироваться другим — ведущим — МК. Это означает, что вывод SS последнего должен быть в состоянии играть роль формирователя сигнала SS_M . Спецификации SPI не предусматривают явно такой возможности для вывода SS в устройстве, находящемся в режиме ведущего. Однако во многих МК выводы SPI (и вывод SS , в частности) мультиплексированы с выводами параллельных портов. Таким образом, переключив вывод SS в режим дискретного выхода общего назначения, можно затем программно формировать на этом выходе ведущего требуемую диаграмму сигнала SS_M .

Порядок обмена между МК на магистрали SPI

Какой может быть последовательность действий МК на такой SPI-магистрали? Исходно все абоненты находятся в режиме ведомого. При этом вывод MISO всех ведомых должен находиться в состоянии «вход». Если какой-либо МК желает обменяться данными с другим МК, он должен:

- Проанализировать состояние сигнала SS_M : если он низкий, это означает, что другой МК ведет передачу, если же высокий, то можно продолжить. Для анализа можно использовать механизм аппаратной детекции коллизий.
- Перейти в режим ведущего, одновременно переключив вывод SS в описанный режим детекции коллизий.
- Перевести свой вывод SS в режим «выход» и одновременно установить SS_M в низкий уровень. Эти два действия можно в некоторых МК выполнить одной командой, просто переключив вход SS в режим выхода порта общего назначения (в регистре данных порта заранее должен быть запи-

² Следует отметить, что в поведении МК при распознавании «ошибки режима» нет унификации даже в реализациях фирмы Motorola. В семействе M68HC11 [3] при детекции коллизии подсистема SPI переводится в режим ведомого, а кроме того, и отключается, а в МК семейства M68HC08 [13] выполняется только отключение, а перевода в режим ведомого не происходит. Кроме того, в семействе M68HC08 «ошибка режима» генерируется и в режиме ведомого том случае, если ведущий деактивировал сигнал SS до окончания передачи пакета.

- сан 0, обеспечивающий после переключения режима низкий уровень на линии SS_M). Если другой абонент попытается сделать то же самое, сработает аппаратная детекция коллизии, и один из двух абонентов будет принудительно (аппаратно) переведен в режим ведомого.
- Теперь все остальные МК находятся в режиме ведомого, активированы сигналом SS_M и могут принять пакеты, содержащие адресную информацию. После чего один из них должен активировать свой выход MISO и участвовать в двустороннем обмене, а остальные должны держать свои выводы MISO неактивными, но могут принимать данные, передаваемые ведущим.

Управление ведомыми на магистрали с несколькими ведущими

Микроконтроллеры, выполняя роль ведомых при межконтроллерной связи на магистрали, способны программно управлять своими выводами MISO, а значит, все МК, не участвующие в данный период в обмене, могут перевести свои выводы MISO в состояние «вход» или в высокоимпедансное состояние и не мешать обмену. В то же время значительная часть периферийных устройств с интерфейсом SPI, являющихся ведомыми, не имеет других способов управления выводом MISO кроме сигнала SS. Поэтому при магистральном соединении многих абонентов по схеме рис. 2 для управления ведомыми придется для каждого из них использовать отдельную линию SS_S (показана на рис. 2). Заметим, что на такую линию SS_S можно «нанизать» несколько ведомых, включенных по каскадной схеме (daisy chain). Управлять сигналом SS_S придется от свободного выхода параллельного порта.

Для реализации описанной магистральной архитектуры потребуется разработка протокола с программной адресацией для «урегулирования отношений» между микроконтроллерами, в то время как управление каждым ведомым осуществляется по отдельной линии SS_S, на которой «нулевые» сигналы от отдельных МК-ведущих объединены «монтажным ИЛИ». Однако, скорее всего, такие структуры «монстры» могут понадобиться в практическом проектировании исключительно редко.

Типовая регистровая модель SPI в микроконтроллере

Она включает, как правило, следующий набор регистров:

- Регистр управления SPI. В нем обычно содержатся биты и битовые поля, позволяющие включать/выключать подсистему SPI и выбирать режимы ее работы, в том числе разрешать прерывания по событиям, происходящим в SPI.

- Регистр состояния SPI. В нем содержатся флаги, отражающие состояние интерфейса, наличие свободного места в буфере передачи, наличие принятого пакета, а также сигнализирующие об ошибочных ситуациях.
- Регистр данных SPI. Чаще всего за этим названием скрывается пара буферных регистров, а может быть, и очереди FIFO, хотя доступ к ним может осуществляться через единственный адрес (к передатчику по записи, к приемнику — по чтению).
- Регистр задания скорости передачи. Запись в него позволяет задать делитель, определяющий частоту следования битов, как это было уже описано. Иногда отдельный регистр делителя отсутствует, а в регистре управления имеется короткое битовое поле, позволяющее выбрать скорость из небольшого набора predetermined значений. Какой-либо унификации набора регистров и функций битовых полей этих регистров в МК разных производителей почти не наблюдается, что затрудняет перенос программ на другое семейство МК.

Перечень основных функций регистра управления SPI:

- Разрешить работу SPI. При запрещенном состоянии обычно останавливается подсистема тактирования, и отключается выходная интерфейсная схемотехника, что приводит к уменьшению энергопотребления.
- Выбрать режим ведущего либо ведомого, при этом переключаются функции некоторых выводов. В частности, SS в режиме ведущего — выход, а в режиме ведомого — вход.
- Задать режим: полярность и фазу тактовых импульсов.
- Разрешить/запретить прерывания от подсистемы SPI.

В отдельных моделях МК программисту могут быть предоставлены и дополнительные возможности, которые в других МК отсутствуют (и, таким образом, являются опциональными).

На рис. 3 приведена структура регистра управления SPI в одной из самых ранних реализаций — в семействе M68HC11 фирмы Motorola.

Этот набор функций управления минимален для SPI. Надо заметить, что фирма Atmel

Бит 7	6	5	4	3	2	1	Бит 0
SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0

Рис. 3. Структура регистра управления в простой реализации SPI в M68HC11 [3]: бит 7 (SPIE) — разрешение прерывания по событиям в SPI; бит 6 (SPE) — включение подсистемы SPI; бит 5 (DWOM) — выбор режима открытого стока (Wired OR Mode) для выводов SPI; бит 4 (MSTR) — выбор режима ведущий либо ведомый; бит 3 (CPOL) — задание полярности тактирующих импульсов; бит 2 (CPHA) — задание фазы тактирующих импульсов; биты 1 и 0 (SPR1, SPR0) — выбор одного из четырех значений битовой частоты

в своих МК семейств AVR воспроизвела регистровую модель МК M68HC11 практически без изменений [12]. Это достойно подражания, так как существенно облегчает портирование управляющей программы с одного МК на другой.

В более поздних реализациях от Motorola и в МК других производителей могут присутствовать дополнительные возможности, которые перечислены далее со ссылками на описание известных автору моделей МК, в которых эти возможности реализованы:

- Возможность связать действия «захват» и «сдвиг» с перепадами тактового сигнала независимо один от другого [14].
- Изменение порядка следования битов в пакете — начиная с младшего бита [10, 11, 12].
- Выбор длины пакета [4, 9].
- Запуск передачи не записью в буфер передатчика, а чтением буфера приемника [11].
- Включение/выключение аппаратного формирования сигнала ведомого SS (а в некоторых МК — даже нескольких таких сигналов) [9, 11].
- Переключение режима аппаратного формирования сигнала SS «непрерывный/старт-стопный» (хотя было бы логично связать это с переключением фазы тактирования СРНА) [11].
- Выбор «потерянного» пакета «старый-новый» для случая, когда программа не успела прочитать из буфера приемника принятый пакет до прихода следующего [11].
- При запуске передачи чтением буфера приемника — выбор содержимого пакета, который будет передаваться: «нули» либо содержимое буфера передатчика. Это позволяет занести это содержимое единожды, а затем принять несколько пакетов от ведомого [11].

Автор уверен в том, что приведенный список не полон, поскольку ему пока не удалось познакомиться со всеми существующими в мире реализациями SPI.

Особенности реализации регистра состояний в SPI

Его отдельные биты обычно переключаются автоматически, отражая изменение состояния интерфейса, о котором необходимо поставить в известность программу. Набор и поведение флагов в различных реализациях SPI фактически никак не стандартизовано, что затрудняет перенос программ с одного МК на другой. Кроме того, в зависимости от наличия двойной буферизации и количество флагов в разных МК также может быть различным.

В структуре без буферных регистров достаточно было единственного флага состояния (либо «происходит сдвиг пакета, передача не завершена, следующий байт записывать нельзя, и новый байт еще не принят», либо «сдвиг пакета завершен, следует сначала прочитать принятый байт, а затем записать в сдвиговый

регистр следующий байт для передачи»). При наличии буферов передачи и приема приемопередатчик может находиться в большем количестве состояний:

- Сдвиги не происходят, буфер передачи и буфер приема пусты, можно записать байт для передачи, но читать нечего.
- Байт был записан в буфер передачи, скопирован в сдвиговый регистр, идет обмен, буфер передатчика пуст, можно записать еще один, буфер приемника пуст, читать нечего.
- Сдвиги завершены, в буфере приемника есть новый байт для чтения, регистр сдвига готов принять новый байт и начать передачу.
- Идет процесс передачи, в буфере приемника есть новый, еще не прочитанный байт. В некоторых сериях МК биты управления и флаги состояния распределены между регистром управления и регистром состояния «вперемешку», так что для настройки SPI программисту придется поработать с обоими типами регистров. В частности, это характерно для МК PicMicro от MicroChip [14].

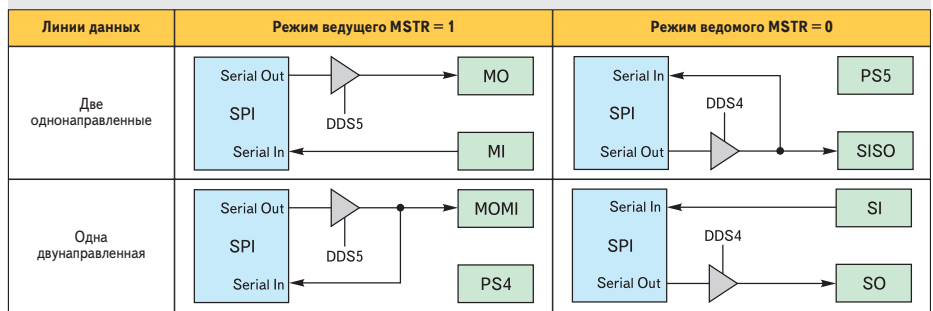
Трехпроводная версия интерфейса SPI с двунаправленной линией данных

Версия интерфейса SPI с двунаправленной линией данных была реализована в семействе МК M68HC12 [10] и в МК от MicroChip [14]. Она также описана в спецификации фирмы Freescale [4]. Режим двунаправленной линии данных обычно доступен только для связи двух МК, один из которых играет роль ведущего, а другой — ведомого.

Структура двунаправленной связи представлена в таблице 3.

В этом режиме SPI использует только одну линию для передачи данных: MOSI на ведущем МК и MISO — на ведомом. На обоих устройствах выход передатчика и вход приемника объединены внутри микросхемы, а для выбора направления передачи следует активировать или деактивировать выходной драйвер, для этого в МК M68HC12 используются дополнительные биты DDS4 и DDS5 в регистре направления передачи порта S, вы-

Таблица 3. Режим двунаправленной линии данных в МК семейства M68HC12



воды которого использует подсистема SPI [10]. Очевидно, что по однопроводной двунаправленной линии возможен лишь полудуплексный обмен, а для исключения электрических и логических конфликтов на линии данных следует использовать те же приемы, что уже обсуждались ранее.

В заключение хочется еще раз подчеркнуть, что, несмотря на бытующее мнение о том, что SPI является промышленным СТАНДАРТОМ, приведенное (далеко не полное) рассмотрение существующего многообразия реализаций заставляет все-таки думать, что стандартизована разве что временная диаграмма сигналов на линиях. И, кроме того, вопреки убеждению, что интерфейс SPI очень прост, автор решается утверждать, что его возможности в полной мере мало кто использует.

В последней статье цикла предполагается достаточно полно описать свойства реализации SPI в аналоговых микроконтроллерах семейства ADuC70xx фирмы Analog Devices. ■

Литература

1. SPI в микроконтроллерах «от А до Я». Часть 1 // Компоненты и технологии. 2009. № 3.
2. http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus — англоязычное описание интерфейса SPI.
3. M68HC11RM/D Rev. 6 (M68HC11 Reference Manual, Motorola Inc., 1989,... 2002): <http://www.pdf-search-engine.com/reference-manual-pdf.html> или <http://www.tcc.edu/faculty/webpages/PGordy/EGR270/M68HC11RM.pdf>.
4. Описание SPI от Freescale: SPI Block Guide V03.06, Freescale Semiconductor: http://www.freescale.com/files/microcontrollers/doc/ref_manual/S12SPIV3.pdf
5. Микросхема Flash-памяти AT45DB161B: www.atmel.com/dyn/resources/prod_documents/DOC2224.pdf
6. AD5444: 12-Bit High Bandwidth Multiplying DAC with Serial Interface: <http://www.analog.com/en/digital-to-analog-converters/da-converters/AD5444/products/product.html>
7. Микросхемы 12-бит АЦП с последовательным интерфейсом MAX1240/ MAX1241: http://www.maxim-ic.com/quick_view2.cfm/qv_pk/1620
8. QSM Queued Serial Module Reference Manual. Freescale Semiconductor, Inc. 1991, 1996. www.freescale.com/files/microcontrollers/doc/ref_manual/QSMRM.pdf
9. ADSP-BF533 Blackfin Processor Hardware Reference, rev. 3.3, 2008, Analog Devices Inc. www.analog.com/static/imported-files/processor_manuals/bf533hwrRev3.3.pdf
10. MC68HC911B32 Technical Data, Motorola Semiconductor, Order No. MC68HC912B32TS/D.
11. ADuC70xx Series: Precision Analog Microcontroller, 12-Bit Analog I/O, ARM7TDMI MCU Data Sheet (Rev. B, 04/2007).
12. AVR ATmega64 Technical Reference. 2008 Atmel Corp.: www.atmel.com/dyn/resources/prod_documents/doc2490.pdf
13. MC68HC908AB32/D Technical Data: www.freescale.com/files/microcontrollers/doc/data_sheet/MC68HC908AB32.pdf
14. PIC16F87X Data Sheet, Microchip Technology Inc., 2001: www.microchip.com/downloads/en/DeviceDoc/30292c.pdf