

# 10 Контроллеры SPI-совместимого порта

Процессор имеет порт последовательного периферийного интерфейса (SPI, Serial Peripheral Interface), обеспечивающий интерфейс ввода-вывода с различными SPI-совместимыми периферийными устройствами.

За счёт ряда конфигурируемых опций SPI-порт обеспечивает аппаратный интерфейс, не требующий дополнительных внешних компонентов, с другими SPI-совместимыми устройствами. SPI представляет собой четырёхпроводной интерфейс с двумя выводами данных, выводом выбора устройства и выводом сигнала синхронизации. SPI является полнодуплексным синхронным последовательным интерфейсом, поддерживающим режимы ведущего и ведомого устройств и режим с несколькими ведущими устройствами. Также поддерживается возможность программного задания скорости двоичной передачи данных и полярности/фазы сигнала синхронизации SPI-совместимого порта. Характерной особенностью SPI является использование формирователей с открытым стоком, обеспечивающих работу в системе с несколькими ведущими устройствами и позволяющими избегать конфликтов данных.

Типичными SPI-совместимыми периферийными устройствами, которые могут подключаться к SPI-совместимому интерфейсу, являются:

- другие процессоры или микроконтроллеры;
- кодеки;
- аналого-цифровые преобразователи;
- цифро-аналоговые преобразователи;
- преобразователи частоты дискретизации;
- SP/DIF или AES/EBU цифровые аудио приемники и передатчики;
- жидкокристаллические индикаторы;
- сдвиговые регистры;
- FPGA с эмуляцией SPI.

SPI это синхронный последовательный канал промышленного стандарта, обеспечивающий связь с несколькими SPI-совместимыми устройствами. SPI-порт является синхронным четырёхпроводным интерфейсом с двумя выводами данных (MOSI и MISO), выводом выбора устройства ( $\overline{SPSS}$ ) и выводом стробированного сигнала синхронизации (SCK). Использование двух выводов данных позволяет организовать взаимодействие с другими SPI-совместимыми устройствами в полнодуплексном режиме. Также имеется возможность программного задания скоростей передачи и фазы/полярности сигнала синхронизации.

## Контроллеры SPI-совместимого порта

SPI-порт может работать в среде с несколькими ведущими устройствами, выполняя роль ведущего или ведомого устройства. Для предотвращения конфликтов на шине данных при работе в среде с несколькими ведущими SPI-интерфейс использует выводы с открытым стоком.

На рис. 10-1 приведена блок-схема SPI-интерфейса. Интерфейс представляет собой сдвиговый регистр, который последовательно передает данные в другое SPI-устройство или принимает данные из другого SPI-устройства (по одному биту за такт SCK). Приём и передача данных при помощи сдвигового регистра осуществляются одновременно. При выполнении передач через SPI передаваемые данные последовательно сдвигаются на выход через сдвиговый регистр, а принимаемые данные последовательно сдвигаются на вход на другом конце того же самого сдвигового регистра. Сигнал SCK служит для синхронизации сдвигаемых и выбираемых данных на двух выводах данных.

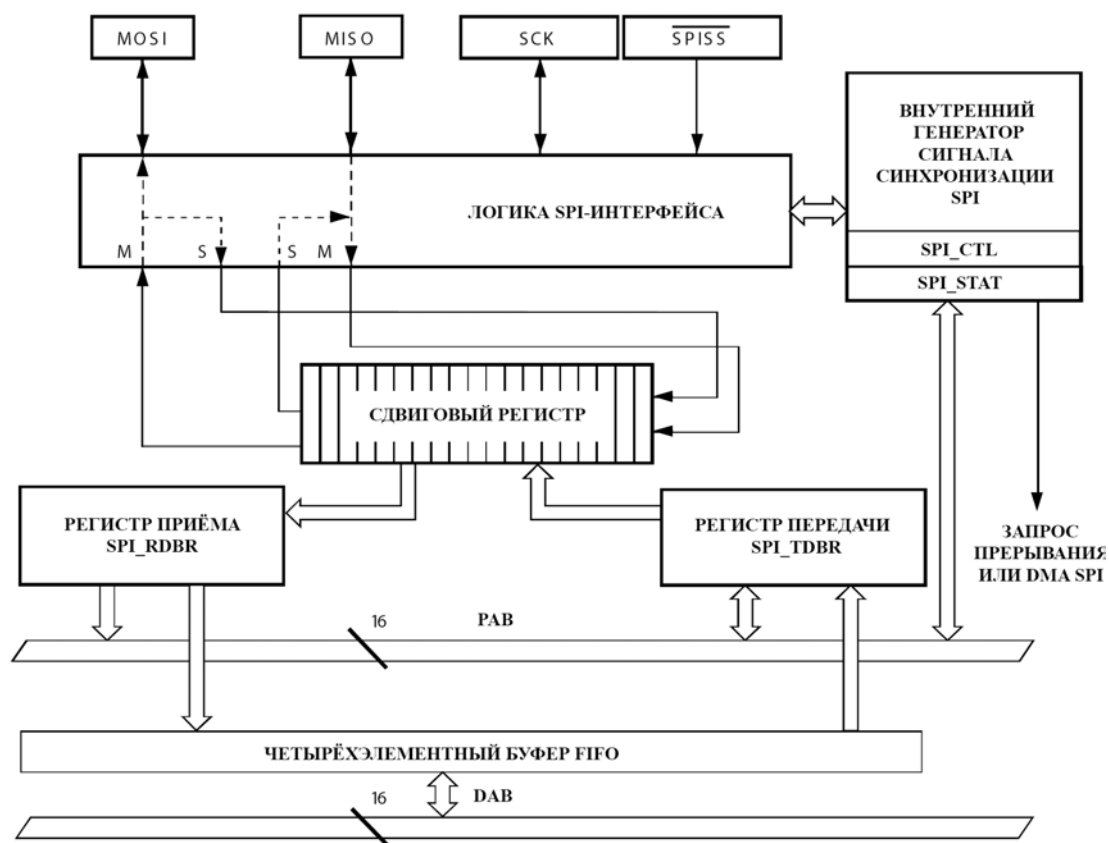


Рис. 10-1. Блок-схема SPI


При передачах данных по SPI-интерфейсу одно из SPI-устройств работает в режиме ведущего устройства SPI-канала. Оно управляет потоком данных, генерируя сигнал синхронизации SPI и выставляя сигнал выбора SPI-устройства (SPISS). Другое SPI-устройство работает в режиме ведомого. Оно принимает данные, поступающие от ведущего устройства, в сдвиговый регистр и передаёт запрашиваемые данные из сдвигового регистра на вывод передачи данных. Несколько процессоров Blackfin, а также других микроконтроллеров и микропроцессоров, могут по очереди становиться ведущими устройствами.

# Контроллеры SPI-совместимого порта

Одно ведущее устройство может одновременно передавать данные нескольким ведомым (широковещательный режим). Однако, в один момент времени только одно из ведомых устройств может передавать данные ведущему устройству. Это свойство должно обязательно выполняться в широковещательном режиме, когда для приёма данных от ведущего устройства может выбираться несколько ведомых, но в один момент времени только одному ведомому устройству разрешено передавать данные ведущему.

В системах с несколькими ведущими устройствами, в которых несколько процессоров взаимодействуют через SPI-порты, соответствующие выводы MOSI, MISO, SCK различных процессоров соединяются друг с другом.

При работе в системе с несколькими ведомыми устройствами процессор использует семь программируемых флагов PF1–PF7 в качестве источников сигналов выбора ведомых устройств SPI.

 После сброса SPI-порт настроен в режим ведомого устройства и его работа запрещена.

## Сигналы интерфейса

В данном разделе описываются сигналы SPI-интерфейса.

### **Сигнал синхронизации последовательного периферийного интерфейса (SCK)**

Сигнал SCK является сигналом синхронизации SPI. Этот управляющий сигнал формируется ведущим устройством и определяет скорость передачи данных. Ведущее устройство может передавать данные с различными скоростями. На каждом такте сигнала SCK передаётся один бит. SCK является выходным сигналом, если устройство работает в режиме ведущего, и входным сигналом, если устройство работает в режиме ведомого.


Сигнал SCK является стробированным тактовым сигналом: он активен только при передаче данных (в течение длины пересылаемого слова). Число активных фронтов импульсов синхронизации равно числу битов, выдаваемых на линии данных. Ведомые устройства игнорируют сигнал синхронизации, если соответствующий сигнал выбора ведомого устройства SPI ( $\overline{SPISS}$ ) неактивен (имеет высокий уровень).

Сигнал SCK используется для сдвига на вход или на выход данных, подаваемых на линии MISO и MOSI. Сдвиг данных на выход всегда выполняется по активному фронту сигнала синхронизации, сдвиг на вход – по неактивному фронту. Полярность сигнала SCK и его фаза относительно данных программируются в регистре управления SPI (SPI\_CTL) и определяют формат передачи данных.

# Контроллеры SPI-совместимого порта

## Входной сигнал выбора ведомого устройства последовательного периферийного устройства

Сигнал  $\overline{SPISS}$  является входным сигналом выбора ведомого SPI-устройства с активным низким уровнем, который используется для разрешения работы процессора, находящегося в режиме ведомого устройства. Функция этого сигнала аналогична сигналу выбора микросхемы; он формируется ведущим устройством для ведомых. При работе в системе с несколькими ведущими устройствами он может использоваться в качестве входа сигнала ошибки ведущего устройства. При работе в такой системе установление активного уровня входного сигнала  $\overline{SPISS}$  ведущего устройства при разрешённом бите PSSE в регистре SPI\_CTL сигнализирует о том, что произошла ошибка. Это означает, что другое устройство пытается стать ведущим.


 В качестве источника сигнала  $\overline{SPISS}$  используется вывод PF0.

## Сигнал MOSI (Выход ведущего вход ведомого)

Вывод MOSI (выход ведущего вход ведомого, Master Out Slave In) является одним из двунаправленных выводов данных. Если процессор находится в режиме ведущего, вывод MOSI является выводом передачи данных (выходным выводом); если процессор находится в режиме ведомого, вывод MOSI является выводом приёма данных (входным выводом). При взаимном соединении SPI-портов данные сдвигаются на выход с вывода MOSI ведущего устройства и сдвигаются на вход (входы) MOSI ведомого устройства (ведомых устройств).

## Сигнал MISO (Вход ведущего выход ведомого)

Вывод MISO (вход ведущего выход ведомого, Master In Slave Out) является одним из двунаправленных выводов данных. Если процессор находится в режиме ведущего, вывод MISO является выводом приёма данных (входным выводом); если процессор находится в режиме ведомого, вывод MISO является выводом передачи данных (выходным выводом). При взаимном соединении SPI-портов данные сдвигаются на выход с вывода MISO ведомого устройства и сдвигаются на вход MISO ведущего устройства.

 Только одно ведомое устройство может передавать данные в определенный момент времени.

На рис. 10-2 приведен пример конфигурации SPI-интерфейса, в которой процессор используется в режиме ведомого устройства. 8-разрядный микроконтроллер (хост) является ведущим SPI-устройством.

## Контроллеры SPI-совместимого порта



Перед началом работы процессор может выполнять загрузку через SPI-интерфейс программы и данных пользователя.

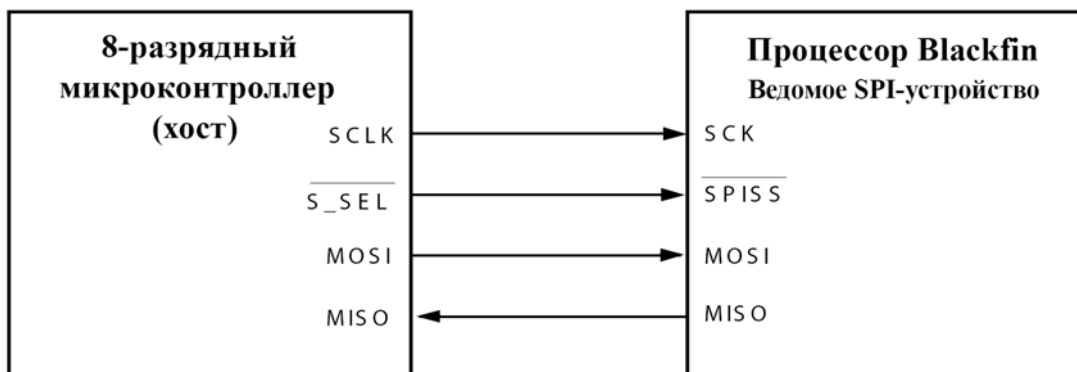


Рис. 10-2. Пример использования ADSP-BF533 в качестве ведомого SPI-устройства

### Выход прерывания

SPI-интерфейс имеет два сигнала прерывания: прерывание данных и прерывание ошибки.

Работа сигнала прерывания данных SPI-интерфейса зависит от значения поля режима инициации передачи (TIMOD) регистра управления SPI. В режиме DMA (TIMOD = 1X) прерывание данных эквивалентно запросу DMA и генерируется, когда буфер FIFO DMA готов для записи (TIMOD = 11) или для чтения (TIMOD = 10). При передаче данных без DMA (TIMOD = 0X) прерывание данных генерируется, когда SPI\_TDBR готов для записи (TIMOD = 01) или когда SPI\_RDBR готов для чтения (TIMOD = 00).

Прерывание ошибки SPI генерируется при работе в режиме ведущего устройства в случае возникновения ошибки сбоя режима. Оно может генерироваться как при передачах данных в режиме DMA, так и при передаче данных без DMA. Прерывание ошибки также может генерироваться в режиме DMA при возникновении условия ошибки отсутствия данных для передачи (TXE при TIMOD = 11) или переполнения буфера приёма (RBSY при TIMOD = 10). При передаче данных без DMA эти условия приводят к установлению битов TXE и RBSY в регистре SPI\_ASTAT, соответственно, но не вызывают генерацию прерывания ошибки.

Дополнительную информацию об этих прерываниях см. в описании битов TIMOD в разделе “Регистр управления SPI (SPI\_CTL)”.

# Контроллеры SPI-совместимого порта

## Регистры SPI

SPI-порт процессора содержит ряд регистров, доступных пользователю. Некоторые из этих регистров также доступны по шине DMA. Четыре регистра содержат информацию управления и состояния – SPI\_BAUD, SPI\_CTL, SPI\_FLG и SPI\_STAT. Два регистра используются для буферизации принимаемых и передаваемых данных – SPI\_RDBR и SPI\_TDBR. Дополнительную информацию о регистрах используемых для обеспечения DMA см. в главе 9 “Прямой доступ к памяти”. Сдвиговый регистр SFDR является внутренним регистром модуля SPI и не доступен напрямую.

Дополнительную информацию об использовании битов этих регистров для сигнализирования об ошибках и других состояниях см. в разделе “Флаги и сигналы ошибок”. Дополнительную информацию о функциях битов регистров SPI см. в разделе “Функции регистров”.

### Регистр скорости двоичной передачи SPI (SPI\_BAUD)

Регистр SPI\_BAUD служит для задания битовой скорости передачи ведущего устройства. Если устройство работает в режиме ведомого, значение, записанное в этот регистр, игнорируется. Частота сигнала синхронизации определяется по следующей формуле:

$$\text{Частота SCK} = (\text{такты частота периферийных устройств}) / (2 * \text{SPI\_BAUD})$$

Запись значения 0 или 1 в регистр запрещает генерацию сигнала синхронизации. Таким образом, максимальное значение частоты SCK составляет одну четвёрть от тактовой частоты системы.

#### Регистр скорости двоичной передачи SPI (SPI\_BAUD)

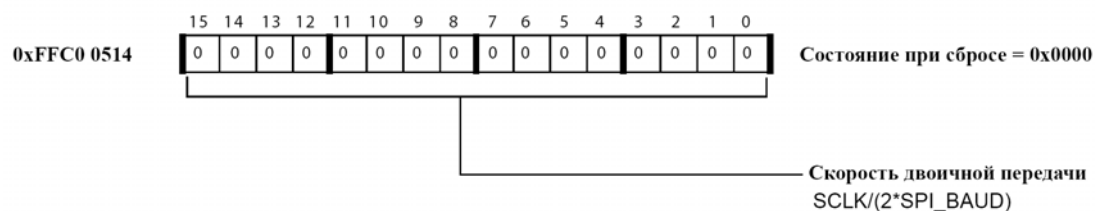


Рис. 10-3. Регистр скорости двоичной передачи SPI

В таблице 10-1 перечислены несколько возможных вариантов задания скорости двоичной передачи при помощи регистра SPI\_BAUD.

# Контроллеры SPI-совместимого порта

Таблица 10-1. Примеры задания скорости двоичной передачи ведущего SPI-устройства.

Десятичное значение SPI_BAUD	Коэффициент деления частоты сигнала SCK	Скорость двоичной передачи при частоте SCLK = 100 МГц
0	сигнал не генерируется	сигнал не генерируется
1	сигнал не генерируется	сигнал не генерируется
2	4	25 МГц
3	6	16.7 МГц
4	8	12.5 МГц
65,535 (0xFFFF)	131,070	763 Гц

## Регистр управления SPI (SPI\_CTL)

Регистр SPI\_CTL используется для разрешения работы и конфигурирования SPI-порта. При помощи этого регистра разрешается работа SPI-интерфейса, выбирается режим работы устройства (в качестве ведущего или ведомого) и определяются формат передачи данных и размер слова.

Термин “слово” относится к одиночной передаче данных, которые могут быть 8- или 16-разрядными в зависимости от значения бита длины слова (SIZE) в регистре SPI\_CTL. В этом регистре имеются два специальных бита, которые могут быть модифицированы аппаратной частью интерфейса: SPE и MSTR.

Поле TIMOD определяет действие, инициирующее выполнение передач в буфер приёма/из буфера передачи. Когда TIMOD = 00 транзакция SPI-порта начинается при чтении из буфера приёма. Данные, полученные при первом чтении, должны игнорироваться, так как выполнение этой операции чтения необходимо для инициации первой транзакции SPI-порта. Когда TIMOD = 01, транзакция инициируется при записи в буфер передачи. При TIMOD = 10 выбирается режим приёма по DMA, первая транзакция инициируется разрешением работы SPI-порта в режиме приёма по DMA. Каждая последующая транзакция инициируется чтением SPI\_RDBR в режиме DMA. При TIMOD = 11 выбирается режим передачи по DMA, транзакция инициируется записью в SPI\_TDBR в режиме DMA.

Бит PSSE используется для разрешения входа  $\overline{SPISS}$  ведущего устройства. Если вход не используется, он может быть запрещён битом PSSE и использован в качестве сигнала ввода/вывода общего назначения.

Бит EMISO активизирует вывод MISO как выход. Это необходимо в системах, в которых может потребоваться одновременная передача данных ведущим устройством в различные ведомые устройства (широковещательный режим). В определённый момент времени только одно ведомое устройство может передавать данные для ведущего. Этот бит должен быть сброшен во всех ведомых устройствах, за исключением того, которое передаёт данные ведущему.

# Контроллеры SPI-совместимого порта

Биты SPE и MSTR могут модифицироваться аппаратной частью интерфейса при установленном бите MODF в регистре состояния. См. раздел «Ошибка сбоя режима».

На рис. 10-4 приведено описание битов регистра SPI\_CTL.

## Регистр управления SPI (SPI\_CTL)

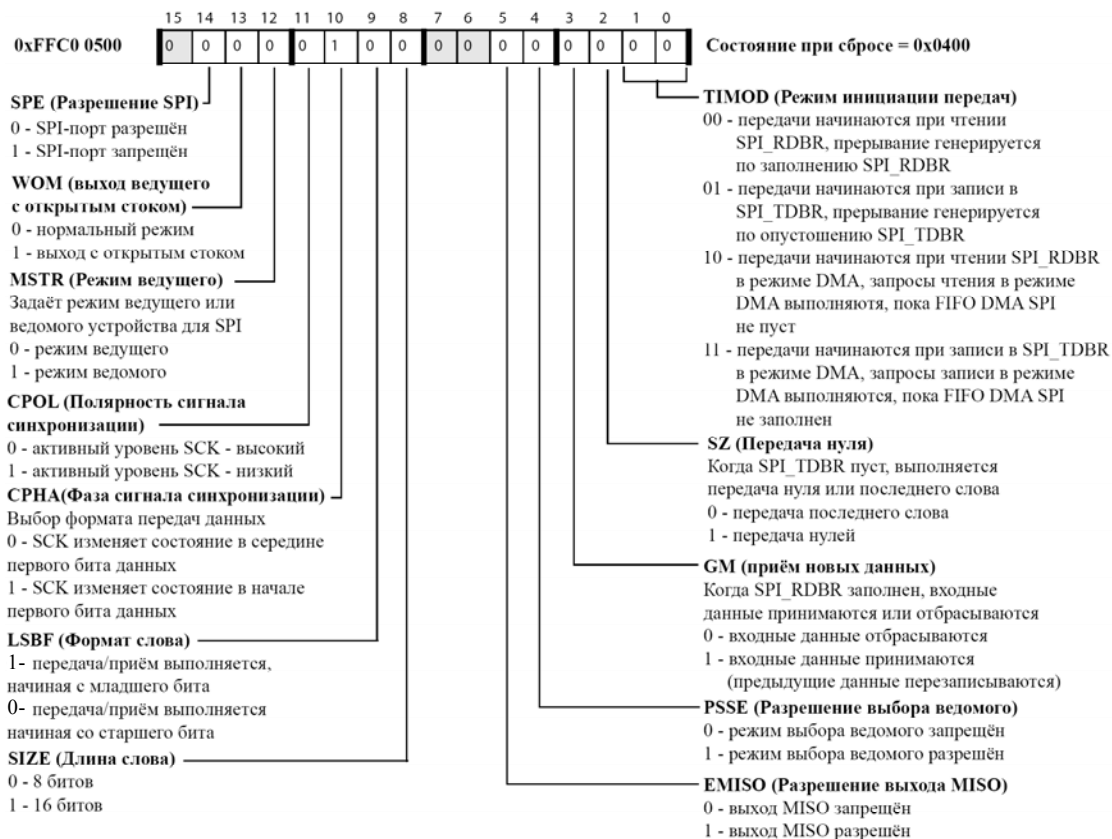


Рис. 10-4. Регистр управления SPI

## Регистр флагов SPI (SPI\_FLG)

Регистр SPI\_FLG применяется SPI-портом для разрешения использования до семи выводов программируемых флагов общего назначения в качестве отдельных линий выбора ведомого устройства при работе SPI-порта в режиме ведущего. В режиме ведомого устройства биты SPI\_FLG ни на что не влияют, и каждый SPI-порт использует в качестве источника входного сигнала выбора ведомого устройства вывод SPISS. Регистр SPI\_FLG показан на рис. 10-5.



# Контроллеры SPI-совместимого порта

## Регистр флагов SPI (SPI\_FLG)

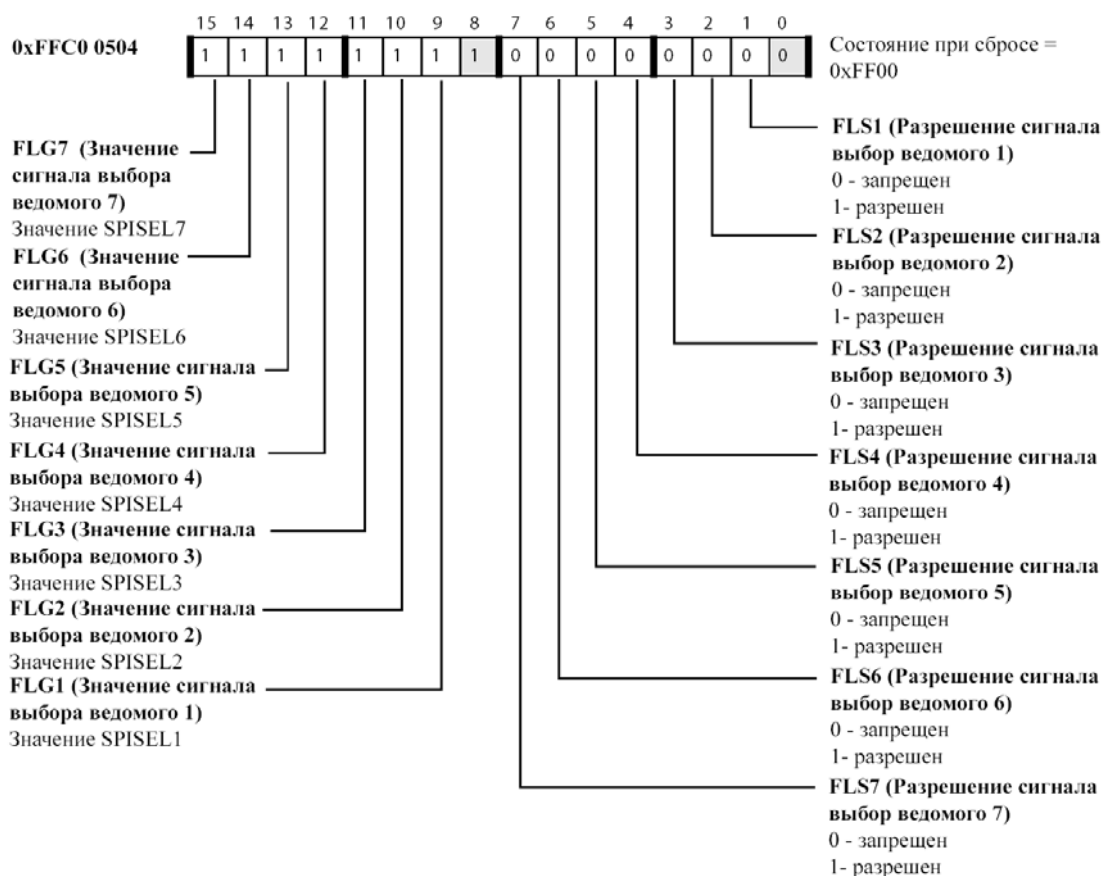


Рис. 10-5. Регистр флагов SPI

Регистр SPI\_FLG состоит из двух наборов битов, функционирующих следующим образом.

- Биты разрешения сигнала выбора ведомого устройства (FLS<sub>x</sub>).

Каждый бит FLS<sub>x</sub> соответствует выводу программируемого флага (PF<sub>x</sub>). Когда бит FLS<sub>x</sub> установлен, соответствующий вывод PF<sub>x</sub> используется для выдачи сигнала выбора ведомого устройства. Например, если в регистре SPI\_FLG установлен бит FLS1, вывод PF1 будет использоваться для выдачи сигнала выбора ведомого устройства (SPISEL1). В таблице 10-2 показано соответствие битов FLS<sub>x</sub> выводам PF<sub>x</sub>.

Если бит FLS<sub>x</sub> не установлен, конфигурирование соответствующего вывода PF<sub>x</sub> и управление им осуществляется при помощи регистров программируемых флагов общего назначения (FIO\_DIR и других).

- Биты значения сигнала выбора ведомого устройства (FLG<sub>x</sub>).

## Контроллеры SPI-совместимого порта

Когда вывод PF<sub>x</sub> используется для выдачи сигнала выбора ведомого устройства, биты FLG<sub>x</sub> могут использоваться для задания его значения. Если установлен бит CPHA в регистре SPI\_CTL, значения выходных сигналов задаются программно с помощью битов FLG<sub>x</sub>. Протокол SPI-порта позволяет либо сбрасывать, либо оставлять активным уровень сигнала на линиях выбора ведомого устройства в интервалах между передачей слов данных. Для этого пользователь должен устанавливать или сбрасывать соответствующие биты FLG<sub>x</sub>. Например, для того чтобы PF3 использовался для выдачи сигнала выбора ведомого устройства, нужно установить бит FLS3 в регистре SPI\_FLG. При сбросе бита FLG3 уровень сигнала на выводе PF3 станет низким, при установке бита FLG3 – высоким. Сигнал на выводе PF3 может периодически принимать высокое и низкое состояние между передачами данных, что достигается периодической установкой и сбросом бита FLG3. Если подобное управление не используется, сигнал на выводе PF3 в интервалах между передачами сохраняет активный (низкий) уровень.

Если CPHA = 0, управление значениями выходных сигналов выбора ведомого устройства осуществляется аппаратно, и биты FLG<sub>x</sub> игнорируются. Протокол SPI требует, чтобы в интервалах между передаваемыми словами данных сигнал на линиях выбора ведомого имел неактивный уровень. Например, для того чтобы PF3 использовался для выдачи сигнала выбора ведомого устройства, необходимо лишь установить бит FLS3 в регистре SPI\_FLG. Установка бита FLG3 не требуется, т.к. аппаратная часть SPI-интерфейса автоматически управляет сигналом на выводе PF3.

Таблица 10-2. Соответствие битов регистра SPI\_FLG и выводов PF<sub>x</sub>

Бит	Название	Функция	Вывод PF <sub>x</sub>	Значение по умолчанию
0		Зарезервирован		0
1	FLS1	Разрешение SPISEL1	PF1	0
2	FLS2	Разрешение SPISEL2	PF2	0
3	FLS3	Разрешение SPISEL3	PF3	0
4	FLS4	Разрешение SPISEL4	PF4	0
5	FLS5	Разрешение SPISEL5	PF5	0
6	FLS6	Разрешение SPISEL6	PF6	0
7	FLS7	Разрешение SPISEL7	PF7	0
8		Зарезервирован		1
9	FLG1	Значение SPISEL1	PF1	1
10	FLG2	Значение SPISEL2	PF2	1
11	FLG3	Значение SPISEL3	PF3	1
12	FLG4	Значение SPISEL4	PF4	1
13	FLG5	Значение SPISEL5	PF5	1
14	FLG6	Значение SPISEL6	PF6	1
15	FLG7	Значение SPISEL7	PF7	1

### Входы выбора ведомого устройства

При работе SPI-порта в режиме ведомого устройства вывод  $\overline{\text{SPISS}}$  служит входом выбора ведомого устройства. При работе SPI-порта в режиме ведущего

# Контроллеры SPI-совместимого порта

устройства вывод  $\overline{\text{SPISS}}$  может служить входом определения ошибки в системе с несколькими ведущими. Это свойство разрешается битом PSSE в регистре SPI\_CTL. Когда PSSE = 1,  $\overline{\text{SPISS}}$  является входом ошибки в режиме ведущего устройства, в противном случае значение сигнала на этом выводе игнорируется.

## Использование битов FLS регистра SPI\_FLG в SPI-системах с несколькими ведомыми устройствами

Биты FLS<sub>x</sub> в регистре SPI\_FLG предназначены для использования в SPI-системах с несколькими ведомыми устройствами. Например, в системе из восьми SPI-устройств с одним ведущим процессором, этот ведущий процессор может поддерживать транзакции через SPI-порт со всеми остальными семью устройствами. В подобной системе с несколькими ведомыми устройствами необходимо, чтобы только один процессор был ведущим, а остальные устройства – ведомыми. Предположим, что SPI-порт работает в режиме ведущего устройства. Тогда семь выводов флагов (PF1–PF7) ведущего процессора могут быть подключены к выводам  $\overline{\text{SPISS}}$  каждого из ведомых устройств. В такой системе биты FLS<sub>x</sub> в регистре SPI\_FLG могут использоваться тремя способами.

В случаях 1 и 2 процессор является ведущим, а семь микроконтроллеров/периферийных устройств с SPI-интерфейсом – ведомыми устройствами. При этом процессор может выполнять следующие действия.

1. Передача во все семь SPI-устройств одновременно в широкопередаточном режиме. В данном случае все биты FLS<sub>x</sub> должны быть установлены.
2. Прием или передача из одного ведомого SPI-устройства в один конкретный момент времени.

В случае 3 все восемь устройств, подключенных друг к другу через SPI-порты, могут быть процессорами.

3. Если все ведомые устройства также являются процессорами, устройство, формирующее запрос, может принимать данные только от одного процессора в один момент времени (разрешается посредством сброса бита EMISO в шести других ведомых процессорах) и передавать данные в широкопередаточном режиме в остальные семь процессоров в тот же момент времени. Аналогичная функция также имеется в некоторых микроконтроллерах. Таким образом, свойство разрешения MISO может использоваться совместно с другими SPI-устройствами, в которых оно поддерживается.

На рис. 10-6 показана система с одним ведущим процессором и тремя ведомыми процессорами (или другими SPI-совместимыми устройствами).

## Контроллеры SPI-совместимого порта

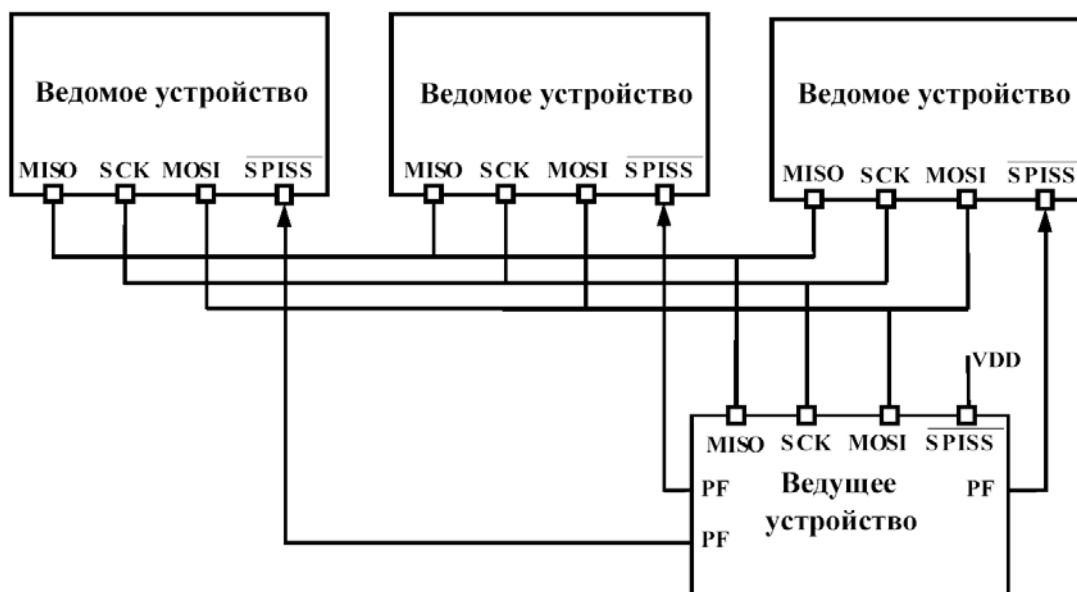


Рис. 10-6. Схема с одним ведущим и несколькими ведомыми устройствами

### Регистр состояния SPI (SPI\_STAT)

Регистр состояния SPI (SPI\_STAT) используется для обнаружения завершения передач данных по SPI-интерфейсу или возникновения ошибки передачи/приёма. Чтение регистра SPI\_STAT возможно в любой момент времени.

Часть битов регистра SPI\_STAT предназначены только для чтения, остальные являются “залипающими”. Биты, доступные только для чтения, отражают информацию о состоянии SPI-порта. Залипающие биты устанавливаются при возникновении условия ошибки. Эти биты устанавливаются аппаратно, а сброс их должен осуществляться программно. Для сброса залипающего бита пользователь должен выполнить запись значения 1 в требуемую позицию регистра SPI\_STAT. Например, если установлен бит TXE, то для сброса условия ошибки TXE пользователь должен записать единицу в бит 2 регистра SPI\_STAT. Таким образом, возможно чтение пользователем регистра SPI\_STAT без изменения значения залипающих битов.

# Контроллеры SPI-совместимого порта

**i** Залипающие биты обнуляются при сбросе, они не обнуляются при запрещении работы SPI-порта.

## Регистр состояния SPI (SPI\_STATUS)

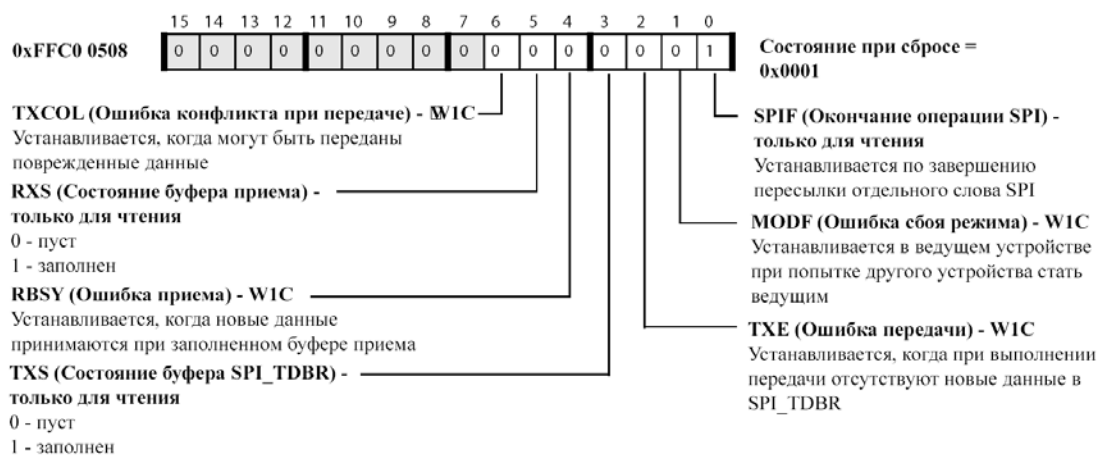


Рис. 10-7. Регистр состояния SPI-порта

Буфер передачи становится заполненным после записи в него данных и становится пустым, когда начинается передача и передаваемое значение загружается в сдвиговый регистр. Приемный буфер становится полным в конце передачи, когда в него загружается значение из сдвигового регистра, и становится пустым при чтении из него.

- i** При запрещении работы SPI-порта устанавливается бит SPIF.
- i** При входе в режим DMA, передающий и приёмный буферы опустошаются. При этом сбрасываются биты TXS и RXS.
- i** При использовании DMA для передачи по SPI-интерфейсу, прерывание DMA\_DONE указывает на то, что буфер FIFO DMA пуст. В действительности, на момент возникновения прерывания в FIFO DMA SPI могут оставаться данные, ожидающие передачи. В связи с этим, необходимо выполнять программный опрос бита TXS в регистре SPI\_STAT, до тех пор, пока при двух операциях чтения подряд не будет возвращено значение 0 в данном бите. Это условие указывает, что буфер FIFO DMA SPI действительно пуст. Последующее обнуление бита SPIF означает, что последнее слово было передано, и можно запретить работу SPI-порта или задать другой режим его работы.

## Регистр буфера передачи SPI (SPI\_TDBR)

Регистр буфера передачи SPI (SPI\_TDBR) является 16-разрядным регистром, доступным для чтения и записи. Загрузка данных в этот регистр выполняется перед началом передачи. Непосредственно перед началом передачи данные, содержащиеся в регистре SPI\_TDBR, загружаются в сдвиговый регистр

## Контроллеры SPI-совместимого порта

данных (SFDR). Чтение регистра SPI\_TDBR может выполняться в любой момент времени, оно не является помехой для операции передачи данных SPI-портом и не инициирует ее.

Если DMA используется для передачи данных, контроллер DMA выполняет загрузку данных в этот регистр непосредственно перед началом передачи. Запись в регистр SPI\_TDBR в этом режиме запрещена, т.к. это может привести к перезаписи передаваемых по DMA данных.

Если DMA используется для приема данных, то SPI-порт циклически выполняет передачу содержимого регистра SPI\_TDBR. В этом режиме разрешена запись в регистр SPI\_TDBR, и записываемые данные передаются на выход.

Если установлен бит передачи нуля (бит SZ в регистре SPI\_CTL), то при определенных обстоятельствах содержимое регистра SPI\_TDBR может быть обнулено.

Если во время текущей передачи выполняется многократная запись в регистр SPI\_TDBR, то будут переданы только те данные, которые были записаны последними. Промежуточные значения, записанные в SPI\_TDBR, передаваться не будут. Многократная запись в регистр SPI\_TDBR возможна, но не рекомендуется.

### Регистр буфера передачи SPI (SPI\_TDBR)

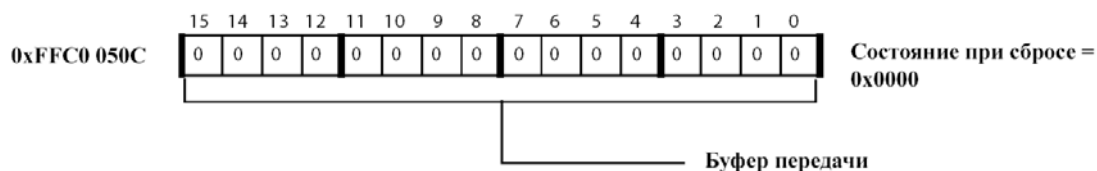


Рис. 10-8. Регистр буфера передачи SPI

### Регистр буфера приёма SPI (SPI\_RDBR)

Регистр буфера приёма SPI (SPI\_RDBR) является 16-разрядным регистром, доступным только для чтения. В момент окончания передачи в регистр SPI\_RDBR загружаются данные из сдвигового регистра. При выполнении операции приема в режиме DMA данные из регистра SPI\_RDBR автоматически считываются по DMA. При программном чтении регистра SPI\_RDBR сбрасывается бит RXS и может быть начата передача данных SPI-портом (если TIMOD=00).

# Контроллеры SPI-совместимого порта

## Регистр буфера приема SPI (SPI\_RDBR)

Только для чтения

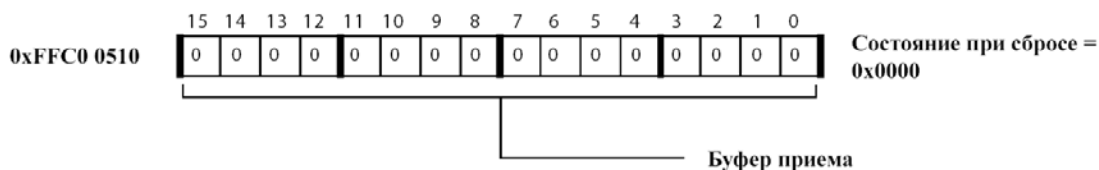


Рис. 10-9. Регистр буфера приёма SPI

## Теневой регистр RDBR (SPI\_SHADOW)

Теневой регистр RDBR (SPI\_SHADOW) предназначен для использования при отладке программы. Его адрес отличается от адреса регистра SPI\_RDBR, а содержимое идентично содержимому SPI\_RDBR. При программном чтении регистра SPI\_RDBR сбрасывается бит RXS в регистре SPI\_STAT и может быть начата передача данных SPI-портом (если поле TIMOD в регистре SPI\_CTL равно 00). При чтении регистра SPI\_SHADOW этого не происходит. Регистр SPI\_SHADOW доступен только для чтения.

## Теневой регистр RDBR\_SPI (SPI\_SHADOW)

Только для чтения

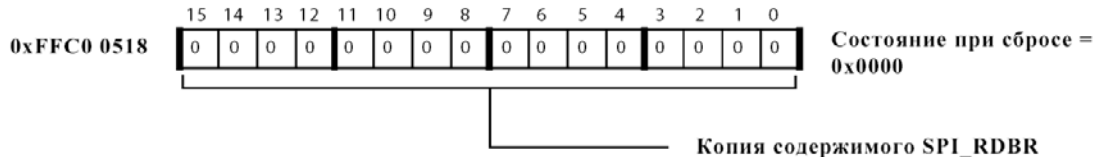


Рис. 10-10. Теневой регистр RDBR

## Функции регистров

В таблице 10-3 перечислены функции регистров SPI-порта.

# Контроллеры SPI-совместимого порта

Таблица 10-3. Карта регистров SPI

Название регистра	Функция	Примечания
SPI_CTL	Управление SPI-портом	Биты SPE и MSTR также могут модифицироваться аппаратно (если установлен бит MODF)
SPI_FLG	Флаги SPI-порта	Биты 0 и 8 зарезервированы
SPI_STAT	Состояние SPI-порта	Бит SPIF может быть установлен путём сброса бита SPE в регистре SPI_CTL
SPI_TDBR	Буфер передачи SPI-порта	Содержимое регистра также может модифицироваться аппаратно (при выполнении DMA и/или когда бит SZ в регистре SPI_CTL равен 1)
SPI_RDBR	Буфер приёма SPI-порта	Чтение этого регистра инициирует аппаратные события
SPI_BAUD	Управление двоичной скоростью передачи SPI-порта	Значение 0 или 1 запрещает генерацию сигнала синхронизации
SPI_SHADOW	Данные SPI-порта	Содержимое регистра совпадает с содержимым регистра SPI_RDBR, однако его чтение не инициирует аппаратные события

## Форматы передачи данных SPI

SPI-порт поддерживает четыре различных варианта при задании фазы и полярности сигнала синхронизации. Они выбираются настройками битов CPOL и CPHA в регистре SPI\_CTL.

Два основных формата передачи, определяемые битом CPHA, показаны на рис. 10-21 “Протокол передачи SPI при CPHA = 0” и рис. 10-22 “Протокол передачи SPI при CPHA = 1”. На каждом из рисунков показаны две временные диаграммы сигнала SCK: для случая CPOL = 0 и для случая CPOL = 1. Эти диаграммы подходят для описания и режима ведущего устройства и режима ведомого устройства, так как выводы SCK, MISO и MOSI ведущего и ведомого устройств соединены друг с другом напрямую. Сигнал MISO является выходным сигналом ведомого устройства (передача ведомого устройства), а MOSI – выходным сигналом ведущего устройства (передача ведущего устройства). Ведущее устройство генерирует сигнал синхронизации (SCK) и сигнал выбора ведомого устройства ( $\overline{SPISS}$ ), который является входным для ведомого. Временные диаграммы приведены для случая 8-разрядной передачи данных (SIZE = 0) начиная со старшего бита (LSBF = 0). Разрешены любые комбинации битов SIZE и LSBF в регистре SPI\_CTL (например, передача 16-разрядных данных, начиная с младшего бита).

Фаза и полярность сигналов синхронизации ведущего и ведомого устройств, включённых в канал связи, должны совпадать. Формат передачи данных со стороны ведущего устройства может изменяться от передачи к передаче для согласования с требованиями к формату различных ведомых устройств.

Если CPHA = 0, сигнал на линии выбора ведомого устройства  $\overline{SPISS}$  должен находиться в неактивном состоянии (высокий уровень) между выполнением передач. В данном случае состояние сигнала  $\overline{SPISS}$  управляется автоматически аппаратной логикой SPI-порта. Если CPHA = 1, сигнал  $\overline{SPISS}$  может сохранять



# Контроллеры SPI-совместимого порта

активное состояние (низкий уровень) или становится неактивным (высокий уровень) между смежными передачами. В данном случае состояние сигнала  $\overline{SPISS}$  должно управляться программно при помощи регистра  $SPI\_FLG$ .

На рис. 10-11 показан протокол передачи SPI-порта при  $CPHA = 0$ . Следует отметить, что сигнал  $SCK$  начинает изменять состояние в середине передачи данных,  $SIZE = 0$  и  $LSBF = 0$ .

На рис. 10-12 показан протокол передачи SPI-порта при  $CPHA = 1$ . Следует отметить, что сигнал  $SCK$  начинает изменять состояние в начале передачи данных,  $SIZE = 0$  и  $LSBF = 0$ .

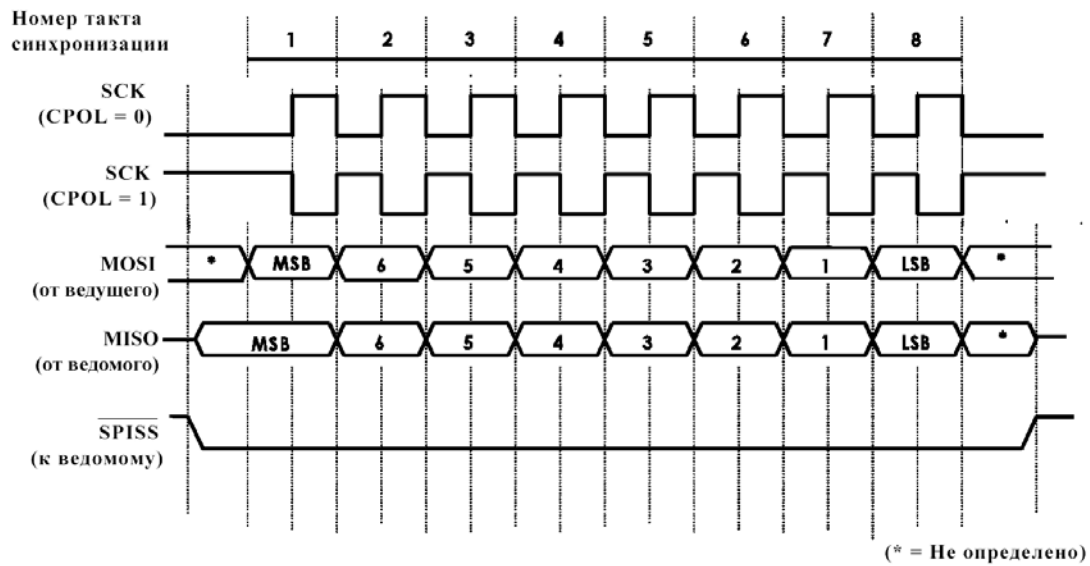


Рис. 10-11. Протокол передачи SPI-порта при  $CPHA = 0$

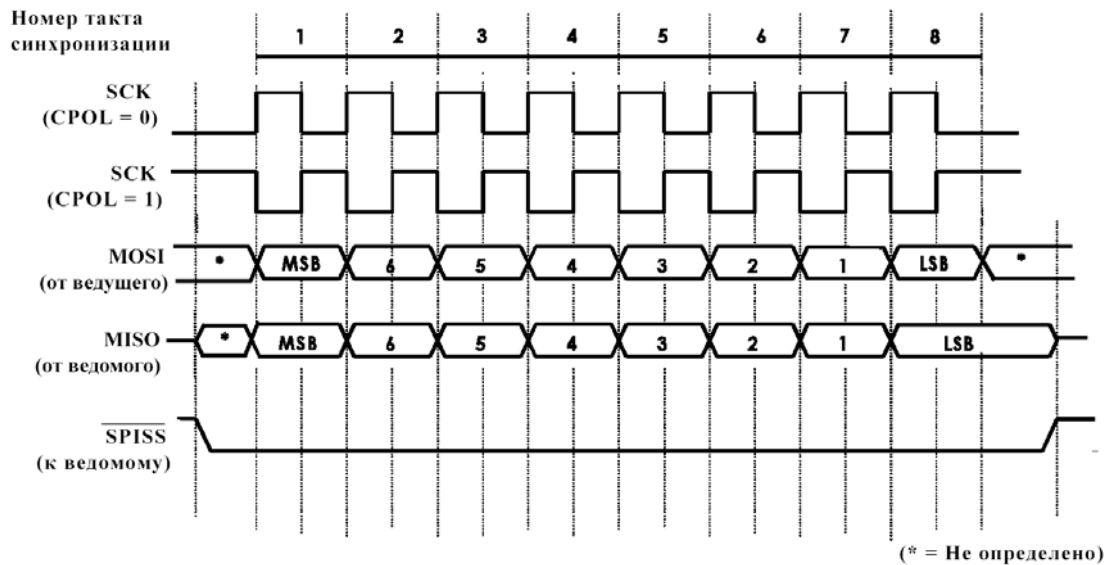


Рис. 10-12. Протокол передачи SPI-порта при  $CPHA = 1$

# Контроллеры SPI-совместимого порта

## Общие принципы работы SPI

SPI-порт может использоваться в системе с одним или несколькими ведущими устройствами. В обеих конфигурациях выводы MOSI, MISO и SCK ведущего и ведомого устройств соединяются друг с другом. Прием и передача SPI-порта всегда разрешаются одновременно, за исключением случая широкопередаточного режима. В широкопередаточном режиме прием данных разрешен нескольким ведомым устройствам, а передача по линии MISO разрешена только одному. В этом разделе описываются сигналы синхронизации, работа SPI-порта в режимах ведущего и ведомого устройств и генерация ошибок.

Для того чтобы избежать повреждения данных при изменении конфигурации модуля SPI необходимо соблюдать следующие правила. Конфигурация не должна изменяться в процессе передачи данных. Полярность сигнала тактовой синхронизации должна изменяться только в тот момент, когда не выбрано ни одно ведомое устройство. Исключением является случай, когда канал связи состоит из одного ведущего и одного ведомого устройства,  $CPHA = 1$ , а вывод выбора ведомого устройства привязан к низкому уровню. В данном случае ведомое устройство выбрано всегда, и повреждения данных можно избежать, разрешая работу ведомого устройства только после завершения конфигурирования обоих устройств.

В системах SPI с несколькими ведущими или несколькими ведомыми устройствами выходные выводы данных (MOSI и MISO) могут быть сконфигурированы как выходы с открытым стоком. Это предотвратит возникновение конфликтных ситуаций и возможное повреждение формирователей на этих выводах. В данном режиме необходимо использовать внешние подтягивающие к питанию (pullup) резисторы на выводах MOSI и MISO.

Режим выходов с открытым стоком управляется битом WOM. Если бит WOM установлен, и SPI-порт работает в режиме ведущего устройства, вывод MOSI переводится в третье состояние, когда выдаваемые на MOSI данные имеют высокий логический уровень. Когда выдаваемые на MOSI данные имеют низкий логический уровень, вывод MOSI не переводится в третье состояние. Аналогично, если бит WOM установлен, и SPI-порт работает в режиме ведомого устройства, вывод MISO переводится в третье состояние, когда выдаваемые на MISO данные имеют высокий логический уровень.

## Сигналы синхронизации

Сигнал SCK является стробированным тактовым сигналом – он активен только в процессе передачи данных (в течение пересылаемого слова). Число активных фронтов импульсов синхронизации равно числу битов, выдаваемых на линии данных. Частота сигнала синхронизации может достигать четверти частоты

# Контроллеры SPI-совместимого порта

сигнала SCLK. Для ведущих устройств частота сигнала синхронизации определяется 16-разрядным значением в регистре SPI\_BAUD. Для ведомых устройств значение этого регистра игнорируется. Когда SPI-устройство является ведущим, сигнал SCK является выходным, когда SPI-устройство является ведомым – входным. Ведомые устройства игнорируют сигнал синхронизации, если сигнал на выводе выбора ведомого устройства имеет неактивный (высокий) уровень.

Сигнал SCK используется для сдвига на вход или на выход данных, выдаваемых на линии MISO и MOSI. Сдвиг данных на выход выполняется по одному фронту сигнала синхронизации (активный фронт), сдвиг на вход – по противоположному фронту (фронт выборки). Полярность сигнала синхронизации и его фаза относительно данных программируются в регистре SPI\_CTL и определяют формат передачи данных.

## Работа в режиме ведущего устройства

Когда SPI-интерфейс сконфигурирован для работы в режиме ведущего устройства (и режим DMA не выбран), он работает следующим образом:

1. Ядро выполняет запись в регистр SPI\_FLG, устанавливая один или более битов выбора флагов (FLSx). Это необходимо для того, чтобы во время конфигурирования ведущего устройства ведомые не выбирались.
2. Ядро выполняет запись в регистры SPI\_CTL и SPI\_BAUD, разрешая работу устройства в режиме ведущего и конфигурируя SPI-систему (задается длина слова, формат передачи данных, скорость двоичной передачи и т.п.).
3. Если CPHA = 1, ядро активизирует нужные ведомые устройства, сбрасывая один или более битов флагов FLGx в регистре SPI\_FLG.
4. Режим инициации передач данных SPI-портом определяется битами TIMOD в регистре SPI\_CTL. Передачи по каналу SPI могут начаться либо после записи данных ядром процессора в буфер передачи (SPI\_TDBR), либо после чтения данных из буфера приема (SPI\_RDBR).
5. Затем SPI-порт генерирует импульсы сигнала синхронизации SCK и одновременно сдвигает данные на выход на вывод MOSI или на вход с вывода MISO. Перед началом сдвига в сдвиговый регистр записывается содержимое регистра SPI\_TDBR. После окончания передачи содержимое сдвигового регистра загружается в регистр SPI\_RDBR.
6. При выполнении каждой новой команды, иницирующей передачи, SPI-порт продолжает принимать и передавать слова в соответствии с режимом инициации передач.

Если буфер передачи остается пустым или буфер приема остается полным, порт работает в соответствии с состоянием битов SZ и GM в регистре SPI\_CTL. Если SZ = 1, и буфер передачи пуст, то на вывод MOSI повторно выдаются нули. Если SZ = 0, и буфер передачи пуст, то повторно передается слово, которое было передано последним до опустошения буфера. Если GM = 1 и буфер приема заполнен, то устройство продолжает принимать новые данные с вывода MISO. При этом старые данные в буфере SPI\_RDBR перезаписываются новыми.

## Контроллеры SPI-совместимого порта

Если  $GM = 0$  и буфер приёма заполнен, то принимаемые данные отбрасываются, а содержимое регистра SPI\_RDBR не обновляется.

### Инициация передач ведущим устройством (режимы передач)

Когда устройство работает в режиме ведущего, инициация передач данных определяется двумя битами TIMOD в регистре SPI\_CTL. В зависимости от значения этих битов и состояния интерфейса новые передачи начинаются либо после записи данных в буфер передачи SPI\_TDBR, либо после чтения данных из приемного буфера SPI\_RDBR (см. табл. 10-4).

Таблица 10-4. Инициация передач

TIMOD	Функция	Условие инициации передач	Выполняемое действие и генерируемое прерывание
00	Передача и приём	Передача одного нового слова инициируется после чтения регистра SPI_RDBR и завершения предыдущей передачи	Прерывание активно, когда буфер приёма заполнен. Прерывание сбрасывается чтением регистра SPI_RDBR.
01	Передача и приём	Передача одного нового слова инициируется после записи в регистр SPI_TDBR и завершения предыдущей передачи	Прерывание активно, когда буфер передачи пуст. Прерывание сбрасывается записью в регистр SPI_TDBR.
10	Приём в режиме DMA	Передача нового блока слов данных инициируется разрешением работы SPI-устройства в режиме DMA. Передачи отдельных слов данных начинаются после чтения SPI_RDBR в режиме DMA и завершения последней передачи.	Запросы чтения в режиме DMA продолжают выполняться до опустошения FIFO DMA SPI
11	Передача в режиме DMA	Передача нового блока слов данных инициируется разрешением работы SPI-устройства в режиме DMA. Передачи отдельных слов данных начинаются после записи в SPI_TDBR в режиме DMA и завершения последней передачи.	Запросы записи в режиме DMA продолжают выполняться до заполнения FIFO DMA SPI

### Работа в режиме ведомого устройства

Когда устройство работает в режиме ведомого (и режим DMA не выбран), передача данных начинается при переходе сигнала  $SPISS$  в активное состояние (низкий уровень) или по первому активному фронту сигнала тактовой синхронизации (SCK) в зависимости от состояния бита SNPA.

При работе в режиме ведомого устройства выполняются следующие действия:

1. Ядро выполняет запись в регистр SPI\_CTL для задания режима канала связи, соответствующего аналогичным настройкам ведущего устройства.

## Контроллеры SPI-совместимого порта

2. Для подготовки к передаче данных ядро записывает передаваемые данные в регистр SPI\_TDBR.
3. При обнаружении заднего фронта сигнала  $\overline{SPISS}$  ведомое устройство начинает принимать данные по неактивным фронтам SCK и посылать данные по активным фронтам SCK.
4. Процесс приёма/передачи продолжается до тех пор, пока сигнал  $\overline{SPISS}$  не будет переведен в неактивное состояние или пока на ведомое устройство не поступит требуемое число импульсов сигнала тактовой синхронизации.
5. Ведомое устройство продолжает принимать/посылать данные по каждому заднему фронту сигнала  $\overline{SPISS}$  и/или по активным фронтам SCK.

Если буфер передачи остается пустым или буфер приёма остается полным, устройство работает в соответствии с состоянием битов SZ и GM в регистре SPI\_CTL. Если  $SZ = 1$ , и буфер передачи пуст, то на вывод MISO повторно выдаются нули. Если  $SZ = 0$ , и буфер передачи пуст, то повторно передается слово, которое было передано последним до опустошения буфера. Если  $GM = 1$ , и буфер приёма заполнен, то устройство продолжает принимать данные с вывода MOSI. При этом старые данные в буфере SPI\_RDBR перезаписываются новыми. Если  $GM = 0$  и приемный буфер заполнен, то принимаемые данные отбрасываются, а содержимое SPI\_RDBR не обновляется.

### Готовность ведомого устройства к передаче данных

Когда устройство работает в режиме ведомого, для подготовки к новой передаче данных необходимо выполнение условий, перечисленных в таблице 10-5.

Таблица 10-5. Подготовка к передаче данных

TIMOD	Функция	Выполняемое действие и генерируемое прерывание
00	Передача и приём	Прерывание активно, когда буфер приёма заполнен. Прерывание сбрасывается чтением регистра SPI_RDBR.
01	Передача и приём	Прерывание активно, когда буфер передачи пуст. Прерывание сбрасывается записью в регистр SPI_TDBR.
10	Приём в режиме DMA	Запросы чтения в режиме DMA продолжают выполняться до опустошения FIFO DMA SPI
11	Передача в режиме DMA	Запросы записи в режиме DMA продолжают выполняться до заполнения FIFO DMA SPI

### Флаги и сигналы ошибок

Состояние устройства отображается в регистре SPI\_STAT. Дополнительную информацию см. в разделе “Регистр состояния SPI (SPI\_STAT)”.

# Контроллеры SPI-совместимого порта

## Ошибка сбоя режима (MODF)

Бит MODF регистра SPI\_STAT устанавливается в том случае, когда входной сигнал на выводе SPISS ведущего устройства переводится в неактивный уровень другим устройством системы. Такая ситуация возникает в системе с несколькими ведущими устройствами, когда другое устройство пытается стать ведущим. Для разрешения изменения ведущего устройства необходимо установить бит PSSE в регистре SPI\_CTL. Конкурирование между двумя устройствами потенциально может повредить выходные формирователи. При обнаружении данной ошибки выполняются следующие действия:

- Сбрасывается бит управления MSTR в регистре SPI\_CTL, что переводит SPI-порт в режим ведомого устройства.
- Сбрасывается бит управления SPE в регистре SPI\_CTL, что запрещает работу SPI-системы.
- Устанавливается бит состояния MODF в регистре SPI\_STAT.
- Генерируется прерывание ошибки SPI.

Эти четыре состояния сохраняются до тех пор, пока бит MODF не будет сброшен программно. Если бит MODF установлен, то работа SPI-порта не может быть разрешена даже в режиме ведомого устройства. Аппаратные средства не позволяют пользователю программно устанавливать биты SPE или MSTR, пока установлен бит MODF.

При сбросе бита MODF прерывание сбрасывается. Перед попыткой снова задать для SPI-порта режим ведущего устройства, следует выполнить опрос состояния входа SPISS. Сигнал на этом входе должен иметь высокий уровень. В противном случае, после установки битов SPE и MSTR снова возникает состояние ошибки сбоя режима.

В результате сброса битов SPE и MSTR запрещается формирование сигналов на выводах данных и синхронизации SPI-интерфейса (MOSI, MISO и SCK). Выходные сигналы на выводах выбора ведомого устройства при этом возвращаются в состояние, когда они управляются регистрами программируемых флагов. Это изменение может привести к конкурированию на линиях выбора ведомого устройства, если эти линии все еще управляются процессором. Для того чтобы запретить формирование выходных сигналов на выводах выбора ведомого устройства при возникновении ошибки MODF, нужно соответствующим образом настроить регистры программируемых флагов.

При разрешении механизма MODF все используемые для выбора ведомого устройства выводы PFx должны быть настроены как входы. Это может быть выполнено программно путём конфигурирования направления выводов PFx перед конфигурированием SPI-интерфейса. В этом случае, когда при возникновении ошибки MODF выводы выбора ведомого устройства автоматически переводятся в режим работы в качестве выводов PFx, формирование выходных сигналов на этих линиях будет запрещено.

# Контроллеры SPI-совместимого порта

## Ошибка передачи (TXE)

Бит TXE устанавливается в регистре SPI\_STAT, когда при полностью выполненных условиях передачи отсутствуют новые данные в буфере SPI\_TDBR (буфер SPI\_TDBR пуст). В этом случае передаваемые данные зависят от состояния бита SZ в регистре SPI\_CTL. Бит TXE является “залипающим” (имеет тип W1C).

## Ошибка приёма (RBSY)

Флаг RBSY устанавливается в регистре SPI\_STAT, если новая операция приёма данных завершается до того, как предыдущие данные считываются из регистра SPI\_RDBR. Необходимость обновления содержимого регистра SPI\_RDBR новыми данными определяется состоянием бита GM в регистре SPI\_CTL. Бит RBSY является “залипающим” (имеет тип W1C).

## Ошибка конфликта при передаче (TXCOL)

Флаг TXCOL устанавливается в регистре SPI\_STAT, когда запись в регистр SPI\_TDBR совпадает с загрузкой регистра сдвига. Запись в регистр SPI\_TDBR может выполняться программно или в режиме DMA. Этот бит указывает на то, что поврежденные данные могли быть загружены в регистр сдвига и затем переданы. В этом случае данные, которые находятся в регистре SPI\_TDBR, могут не соответствовать переданным. Этой ошибки легко можно избежать при соответствующем программном управлении. Бит TXCOL является “залипающим” (имеет тип W1C).

## Начало и окончание передач по SPI

Начало и окончание передачи по SPI-интерфейсу определяется режимом работы SPI-устройства (ведущее или ведомое) и состояниями битов CPHA и TIMOD. Для ведущего SPI устройства при CPHA = 0 передачи начинаются с записи в регистр SPI\_TDBR или с чтения регистра SPI\_RDBR, в зависимости от состояния поля TIMOD. В начале передачи данных на разрешенные выходы выбора ведомого устройства подаётся сигнал активного (низкого) уровня. Однако, сигнал SCK остается неактивным в течение первой половины первого такта SCK. Для ведомого устройства при CPHA = 0 передача начинается сразу после того, как сигнал SPISS принимает активный уровень.

При CPHA = 1 передача начинается по первому активному фронту сигнала SCK и для ведущего и для ведомого устройства. Для ведущего устройства передача считается законченной после того, как он отправит последние данные и одновременно примет последний бит данных. Для ведомого устройства передача заканчивается после последнего фронта выборки SCK.

## Контроллеры SPI-совместимого порта

Момент, когда буфер приёма может быть считан, определяется битом RXS, а момент, когда буфер передачи может быть заполнен – битом TXS. Окончание приема одного слова происходит, когда устанавливается бит RXS. Установление этого бита означает, что новое слово было только что принято и зафиксировано в буфере приёма, SPI\_RDBR. В ведущем устройстве бит RXS устанавливается вскоре после последнего фронта выборки SCK. В ведомом устройстве бит RXS устанавливается вскоре после последнего фронта SCK, независимо от состояния битов CPHA или CPOL. Задержка обычно составляет несколько тактов SCLK и не зависит от CPHA, TIMOD и скорости двоичной передачи. Если разрешена генерация прерывания по заполнению буфера SPI\_RDBR (TIMOD = 00), то сигнал прерывания выставляется на один такт SCLK после установки бита RXS. Окончание приема можно отследить также, не используя прерывание, при помощи опроса состояния бита RXS.

Для обеспечения программной совместимости с другими SPI-устройствами разрешен опрос бита SPIF. Поведение этого бита в процессоре Blackfin может слегка отличаться от его поведения в других серийно выпускаемых SPI-устройствах. Для ведомого устройства бит SPIF сбрасывается, спустя небольшое время после начала передачи (перехода SPISS в активное состояние при CPHA = 0 или первого активного фронта SCK при CPHA = 1), и устанавливается одновременно с битом RXS. Для ведущего устройства бит SPIF сбрасывается, спустя небольшое время после начала передачи (при записи в регистр SPI\_TDBR или при чтении регистра SPI\_RDBR, в зависимости от значения TIMOD), и устанавливается через половину периода сигнала SCK после последнего его фронта, независимо от состояния битов CPHA или CPOL.

Время, спустя которое устанавливается бит SPIF, зависит от скорости двоичной передачи. В общем случае, бит SPIF будет установлен после установки бита RXS. Однако, при низких скоростях передачи (SPI\_BAUD < 4) бит SPIF будет установлен до установки бита RXS и, следовательно, до того, как новые данные будут зафиксированы в SPI\_RDBR, так как существует задержка. Поэтому при SPI\_BAUD = 2 или 3 перед чтением регистра SPI\_RDBR необходимо ожидать установки бита RXS после установки бита SPIF. При более высоких скоростях двоичной передачи данных гарантируется, что RXS будет установлен раньше SPIF.

## DMA

SPI-порт также может использовать прямой доступ к памяти (DMA) Дополнительную информацию о DMA см. в главе 9 “Прямой доступ к памяти”.

### Возможности DMA SPI-порта

SPI-порт имеет один контроллер DMA, который может использоваться для поддержки канала приёма или канала передачи DMA SPI. Одновременная



# Контроллеры SPI-совместимого порта

работа обоих каналов DMA невозможна. Вследствие этого, при разрешении канала передачи DMA, принимаемые данные игнорируются.

При разрешении канала приёма, передаваемые данные не имеют значения. Для увеличения пропускной способности при передачах по шине доступа DMA (DAB) имеется четырёхэлементный 16-разрядный буфер FIFO (без поддержки пакетных передач).

**i** При передаче по SPI-интерфейсу в режиме DMA, прерывание DMA\_DONE указывает на то, что буфер FIFO DMA пуст. В действительности, на момент возникновения прерывания в FIFO DMA SPI могут оставаться данные, ожидающие передачи. В связи с этим, необходимо выполнять программный опрос бита TXS в регистре SPI\_STAT, до тех пор, пока при двух операциях чтения подряд не будет возвращено значение 0 в данном бите. Это условие указывает, что буфер FIFO DMA SPI действительно пуст. Последующее обнуление бита SPIF означает, что последнее слово было передано, и можно запретить работу SPI-порта или задать другой режим его работы.

**i** При запрещении работы SPI-порта четырёхэлементный буфер FIFO очищается.

## Передачи DMA в режиме ведущего устройства

Когда SPI-интерфейс работает в режиме ведущего устройства и использует DMA для приёма или передачи данных, выполняются следующие операции:

1. Ядро процессора выполняет записи в соответствующие регистры DMA для разрешения канала DMA SPI и для настройки параметров DMA (адресов передаваемого блока данных, направления доступа, количества слов и т.д.). Дополнительную информацию см. в главе 9 “Прямой доступ к памяти”.
2. Ядро процессора выполняет запись в регистр SPI\_FLG, устанавливая один или несколько битов выбора флагов SPI (FLSx).
3. Ядро процессора выполняет запись в регистры SPI\_CTL и SPI\_BAUD, разрешая работу устройства в режиме ведущего и конфигурируя SPI-систему (задается длина слова, формат передачи данных, скорость двоичной передачи и т.п.). В поле TIMOD задается режим приема (TIMOD = 10) или передачи (TIMOD = 11) с использованием DMA.
4. В режиме приёма – первая операция приёма инициируется разрешением работы SPI-порта. Последующие операции инициируются при чтении SPI-портом регистра SPI\_RDBR и записи в FIFO DMA SPI. Затем SPI-порт запрашивает запись в память в режиме DMA. При предоставлении доступа, контроллер DMA выполняет чтение слова из FIFO DMA SPI и записывает его в память.

В режиме передачи – SPI-порт запрашивает чтение из памяти в режиме DMA. При предоставлении доступа, контроллер DMA выполняет чтение слова из памяти и записывает его в FIFO DMA SPI. При записи SPI-портом

## Контроллеры SPI-совместимого порта

данных из FIFO DMA SPI в регистр SPI\_TDBR инициируется передача по каналу SPI.

5. Затем SPI-порт генерирует импульсы сигнала синхронизации на выводе SCK и одновременно сдвигает данные на выход, используя вывод MOSI, или на вход, используя MISO. В режиме приема значение регистра сдвига загружается в регистр SPI\_RDBR конце операции приема. В режиме передачи значение регистра SPI\_TDBR загружается в регистр сдвига в начале операции передачи.
6. В режиме приёма – SPI продолжает запросы записи в память в режиме DMA до тех пор, пока в FIFO DMA SPI есть данные (буфер FIFO не пуст). Контроллер DMA продолжает чтение слов из FIFO DMA SPI и запись в память, пока содержимое регистра счетчика слов DMA SPI не изменится из 1 в 0. SPI-порт продолжает приём слов до запрещения работы SPI-порта в режиме DMA.  
В режиме передачи – SPI продолжает запросы чтения из памяти в режиме DMA до тех пор, пока в FIFO DMA SPI есть место (буфер FIFO не заполнен). Контроллер DMA продолжает чтение слов из памяти и запись в FIFO DMA SPI, пока содержимое регистра счетчика слов DMA SPI не изменится из 1 в 0. SPI-порт продолжает передачу слов до опустошения FIFO DMA SPI.

Если при приёме в режиме DMA контроллер DMA не справляется с потоком принимаемых данных, буфер приёма работает в соответствии с состоянием бита GM. Если  $GM = 1$ , и буфер DMA заполнен, SPI-порт продолжает принимать данные с вывода MISO, перезаписывая старые данные в регистре SPI\_RDBR новыми. Если  $GM = 0$ , и буфер DMA заполнен, принимаемые данные отбрасываются, а содержимое регистра SPI\_RDBR не обновляется. При приёме в режиме DMA установлен бит TXE, и предполагается, что буфер передачи пуст. Если  $SZ = 1$ , на вывод MOSI повторно выдаются нули. Если  $SZ = 0$ , устройство повторно передаёт содержимое регистра SPI\_TDBR. В данном режиме генерация прерывания ошибки не может быть вызвана состоянием отсутствия данных в буфере передачи (TXE).

При передаче в режиме DMA ведущее SPI-устройство инициирует передачу слова только при наличии данных в FIFO DMA. Если FIFO DMA пуст, SPI-порт ожидает записи в него контроллером DMA для инициации передачи. Если DMA используется при передаче, все аспекты, касающиеся приёма SPI-порта, игнорируются, включая данные в регистре SPI\_RDBR и состояние битов RXS и RBSY. При работе в данном режиме генерация прерывания ошибки не может быть вызвана состоянием переполнения буфера приёма (RBSY). Возникновения ошибки отсутствия данных в буфере передачи (TXE) в данном режиме (передачи в режиме DMA ведущим устройством) невозможно, так как при отсутствии данных в FIFO DMA ведущее SPI-устройство не инициирует передачи данных.

Запись в регистр SPI\_TDBR при выполнении SPI-портом передачи в режиме DMA нежелательна, так как при этом данные DMA будут перезаписаны. Запись в регистр SPI\_TDBR при выполнении SPI-портом приёма в режиме DMA допускается. Чтение регистра SPI\_RDBR возможно в любой момент времени.

# Контроллеры SPI-совместимого порта

Запросы DMA генерируются, когда FIFO DMA не пуст (при  $TIMOD = 10$ ) или не заполнен (при  $TIMOD = 10$ ).

Прерывания ошибки генерируются при возникновении условия переполнения,  $RBSY$  (при  $TIMOD = 10$ ).

Последовательность передач в режиме DMA, выполняемая ведущим SPI-устройством, может представлять собой передачи с подтверждением приёма и/или приём нескольких блоков данных. При выполнении таких последовательностей передач контроллером SPI требуется минимальное вмешательство ядра.

## Передачи с использованием DMA в режиме ведомого устройства

При приёме или передаче с использованием DMA в режиме ведущего устройства начало передач вызывается, в зависимости от состояния бита  $SNRA$ , переходом сигнала  $\overline{SPISS}$  в активное состояние (низкий уровень) или первым активным фронтом  $SCK$ .

При выполнении ведомым SPI-устройством последовательности передачи или приёма в режиме DMA происходит следующее:

1. Ядро процессора выполняет запись в соответствующие регистры DMA для разрешения канала DMA SPI и настройки параметров DMA (адресов передаваемых блоков, количества слов и т.д.). Дополнительную информацию см. в главе 9, “Прямой доступ к памяти”.
2. Ядро процессора выполняет запись в регистр  $SPI\_CTL$ , задавая режим работы канала последовательной связи, совпадающий с настройками ведущего устройства SPI. Полем  $TIMOD$  задаётся режим приёма ( $TIMOD = 10$ ) или передачи ( $TIMOD = 11$ ) с использованием DMA.
3. При приёме в режиме DMA – когда сигнал на выводе выбора ведомого устройства становится активным, ведомое устройство начинает передавать и принимать данные по активным фронтам  $SCK$ . В конце каждой передачи в регистр  $SPI\_RDBR$  загружается значение из регистра сдвига. Запрос записи в память в режиме DMA вызывается при выполнении SPI-портом чтения данных из регистра  $SPI\_RDBR$  или записи в FIFO DMA SPI. При предоставлении доступа контроллер DMA выполняет чтение слова из FIFO DMA SPI и запись в память.  
При передаче в режиме DMA – SPI-порт выполняет запрос чтения из памяти в режиме DMA. При предоставлении доступа контроллер DMA выполняет чтение слова из FIFO DMA SPI и запись в регистр  $SPI\_TDBR$ , после чего ожидает начала следующей передачи. Когда сигнал на выводе выбора ведомого устройства становится активным, ведомое устройство начинает передавать и принимать данные по активным фронтам  $SCK$ . В начале передачи в регистр сдвига загружается содержимое регистра  $SPI\_TDBR$ .
4. В режиме приёма ведомое SPI-устройство продолжает выполнять запросы записи в память в режиме DMA, пока в FIFO DMA SPI имеются данные (буфер FIFO не пуст). Контроллер DMA продолжает чтение слов из FIFO

## Контроллеры SPI-совместимого порта

DMA SPI и запись в память, пока регистр счётчика слов DMA SPI не изменяет значение из 1 в 0. Ведомое SPI-устройство продолжает приём слов по активным фронтам SCK, пока сигнал на входе выбора ведомого устройства сохраняет активный уровень.

В режиме передачи ведомое SPI-устройство продолжает выполнять запросы записи в память в режиме DMA, пока в FIFO DMA SPI есть место (буфер FIFO не заполнен). Контроллер DMA продолжает чтение слов из памяти и запись в FIFO DMA SPI, пока регистр счётчика слов DMA SPI не изменяет значение из 1 в 0. Ведомое SPI-устройство продолжает передачу слов по активным фронтам SCK, пока сигнал на входе выбора ведомого устройства сохраняет активный уровень.

Если при приёме в режиме DMA контроллер DMA не справляется с потоком принимаемых данных, буфер приёма работает в соответствии с состоянием бита GM. Если  $GM = 1$ , и FIFO DMA заполнен, устройство продолжает принимать новые данные на выводе MOSI, перезаписывая старые данные регистра SPI\_RDBR. Если  $GM = 0$ , и FIFO DMA заполнен, поступающие данные отбрасываются, и содержимое регистра SPI\_RDBR не обновляется. При приёме в режиме DMA установлен бит TXE, и предполагается, что данные в буфере передачи отсутствуют. Если  $SZ = 1$ , SPI-порт повторно передаёт нули на вывод MISO. Если  $SZ = 0$ , на вывод MISO повторно передаётся содержимое регистра SPI\_TDBR. При работе в данном режиме генерация прерывания ошибки не может быть вызвана условием отсутствия данных в буфере передачи (TXE).

Если при передаче в режиме DMA контроллер DMA не справляется с потоком передаваемых данных, буфер передачи работает в соответствии с состоянием бита SZ. Если  $SZ = 1$ , и FIFO DMA пуст, SPI-порт повторно передаёт нули на вывод MISO. Если  $SZ = 0$ , и FIFO DMA пуст, повторно передаётся слово, которое было передано последним перед опустошением буфера DMA. Если DMA используется при передаче, все аспекты, касающиеся режима приёма SPI-порта, игнорируются, включая данные в регистре SPI\_RDBR и состояние битов RXS и RBSY. При работе в данном режиме генерация прерывания ошибки не может быть вызвана состоянием переполнения буфера приёма (RBSY).

Запись в регистр SPI\_TDBR при выполнении SPI-портом передачи в режиме DMA нежелательна, так как при этом данные DMA будут перезаписаны. Запись в регистр SPI\_TDBR при выполнении SPI-портом приёма в режиме DMA допускается. Чтение регистра SPI\_RDBR возможно в любой момент времени.

Запросы DMA генерируются, когда FIFO DMA не пуст (при  $TIMOD = 10$ ) или не заполнен (при  $TIMOD = 10$ ).

Прерывания ошибки генерируются при возникновении условия переполнения буфера приёма, RBSY (при  $TIMOD = 10$ ), или отсутствия данных в буфере передачи, TXE (при  $TIMOD = 11$ ).

## Временные характеристики

Времена опережения разрешения ( $T_1$ ), запаздывания разрешения ( $T_2$ ) и задержки последующей передачи ( $T_3$ ) должны быть больше либо равны половине периода  $SCK$ . См. рис. 10-13. Минимальный интервал между двумя смежными передачами данных ( $T_4$ ) составляет два периода  $SCK$ . Этот интервал измеряется от последнего активного фронта  $SCK$  одного слова до первого активного фронта  $SCK$  следующего слова. Он не зависит от конфигурации SPI-порта (значения битов  $CPHA$ ,  $MSTR$  и т.д.).

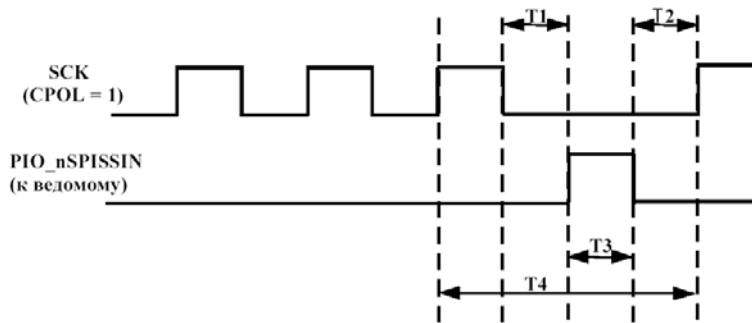


Рис. 10-13. Временные характеристики SPI

Для ведущего устройства с  $CPHA = 0$ , сигнал на выходе выбора ведомого будет иметь неактивный (высокий) уровень, по меньшей мере, в течение половины периода  $SCK$ . В данном случае, времена  $T_1$  и  $T_2$  всегда равны половине периода  $SCK$ .