

Синхронный последовательный интерфейс SPI в микроконтроллерах «от А до Я» и его реализация на примере ADuC70xx фирмы Analog Devices

Александр НОВИЦКИЙ

Поначалу в данной статье предполагалось описать особенности реализации синхронного последовательного интерфейса SPI в семействе однокристалльных микроконтроллеров (далее — МК) ADuC70xx фирмы Analog Devices. Автор намеревался не только рассмотреть особенности реализации SPI, но и обсудить, когда эти особенности оказываются полезными, как и для чего их следует использовать.

В ходе написания статьи оказалось, что доля текста, посвященного вопросам «когда, как и для чего», превысила 50%. Тогда было принято решение изменить характер изложения в направлении, обозначенном в заглавии.

Вследствие выросшего объема материал будет разбит на три отдельные статьи:

1. Принципы интерфейса SPI и организация ведомых устройств.
2. Подсистемы SPI в микроконтроллерах: варианты организации и использование.
3. Подсистема SPI в МК семейства ADuC70xx.

История SPI

Изобретатель интерфейса SPI — фирма Motorola [1]. Однако изобретение было сделано не на «пустом месте». Предшественником послужил интерфейс MicroWire фирмы National Semiconductor [2]. Автор статьи познакомился с интерфейсом SPI, работая с популярным семейством M68HC11 — практически первым, в котором SPI появился в конце 1980-х годов. Основное предназначение этого интерфейса — подключение к МК различных периферийных устройств. Очень быстро SPI стал весьма популярен по следующим причинам:

- а) минимальные требования к аппаратной части подключаемого к МК устройства (сдвиговый регистр);
 - б) SPI-ведущего в минимальной конфигурации легко эмулировать программно, если в МК отсутствует аппаратная реализация SPI.
- В последние два десятилетия многочисленные фирмы производят огромное количество различных периферийных узлов, подключаемых к МК через SPI (см., например, [3]).

Это датчики физических величин (температуры, давления, ускорения и т. п.), устройства аналого-цифрового преобразования (АЦП, ЦАП, цифровые потенциометры), устройства преобразования интерфейсов (CAN-контроллеры, Ethernet-контроллеры), модули энерго-независимой памяти (Flash-карты MMC и SD, микросхемы EEPROM) и многие другие. В [4] (2005 год) утверждается, что «примерно 85% МК оснащены интерфейсом SPI».

Исторически, однако, сложилось так, что какой-либо официальный общепризнанный стандарт на интерфейс SPI отсутствует. Поэтому свойства подсистемы SPI в микросхемах разных производителей могут различаться как набором возможностей, так и их реализацией, что нередко вызывает затруднения при использовании. Разработчики преодолевают эти затруднения по большей части методом проб и ошибок либо обмениваясь опытом. В свою очередь производители МК нередко, в предположении, что SPI — это общеизвестный стандарт, описывают свойства подсистемы излишне кратко. Этим, в частности, грешит техническое описание МК семейства ADuC70xx фирмы Analog Devices [5].

Весьма нелегко найти источники, в которых бы систематически были описаны все (или хотя бы большинство) из возможных особенностей интерфейса SPI. Обычно встречаемая на форумах разработчиков рекомендация — взять техническое описание МК, в котором SPI имеется. Но там, естественно, описывается лишь подмножество опций, реализованных в конкретном МК.

Автору известно лишь несколько публикаций, в которых сделана попытка описать свойства SPI без привязки к конкретному типу МК [3, 4, 6, 14], и все они страдают неполнотой. Некоторые фирмы опубликовали свое «видение» того, что представляет собой SPI, например [1, 7–9], однако упомянутые документы в значительной степени «привязаны» к реализациям МК, выпускаемых этими фирмами.

Минимальная архитектура интерфейса и конфигурации

Минимальная архитектура изображена на рис. 1. Интерфейс SPI содержит четыре линии:

- MOSI (Master Out Slave In) — выход данных ведущего (она же вход данных ведомого);

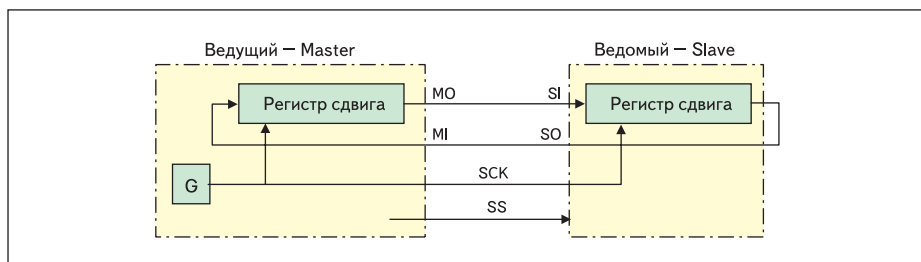


Рис. 1. Структура связей и линии интерфейса SPI

- MISO (Master In Slave Out) — вход данных ведущего (она же выход данных ведомого);
- SCK (Serial Clock) — тактирование (синхронизация);
- SS (Slave Select) — выбор ведомого.

Синхронизация передачи

Частота следования битовых интервалов в линиях данных SPI определяется синхросигналом SCK, который генерирует один из абонентов — ведущий (Master). Прочие абоненты — ведомые (Slave), которых может быть как один, так и несколько, используют синхросигнал для определения моментов изменения битов на линии данных. Ведомые устройства никак не могут влиять на частоту следования битовых интервалов (в отличие, например, от интерфейса I²C).

Ведущий в подавляющем большинстве случаев — это приемопередатчик SPI в составе МК (если аппаратный SPI отсутствует, его можно эмулировать программно). Однако автору известны реализации SPI-ведущего устройства в составе микросхем PLD фирмы Xilinx [10], а также в ИМС преобразователя параллельного интерфейса в интерфейс USB [11]. В обоих этих случаях ведущий ориентирован на подключение определенного типа микросхем: EEPROM с SPI-совместимым интерфейсом.

Ведомое устройство — это обычно периферийная микросхема. Однако SPI можно использовать и для связи двух (или больше) МК, эта возможность будет обсуждаться позже.

Передача битов осуществляется пакетами. Длина пакета чаще всего составляет 1 байт (8 битов), хотя встречаются реализации SPI с иной длиной пакета [9, 12, 15]. Как в ведущем устройстве, так и в ведомом имеется (обычно программно недоступный) счетчик импульсов синхронизации (битов). Счетчик в ведомом устройстве позволяет последнему определить момент окончания передачи пакета. Счетчик битов сбрасывается в ноль при выключении (деактивации) подсистемы SPI, такая возможность всегда имеется в ведущем устройстве. В ведомом устройстве счетчик битов обычно сбрасывается деактивацией интерфейсного сигнала SS. Передача пакета, то есть запуск генератора синхроимпульсов в ведущем, обычно осуществляется автоматически при записи передаваемого байта в регистр сдвига ведущего.

Задание частоты следования битов F_{SCK} осуществляется стандартным для последовательных интерфейсов способом: $F_{SCK} = F_G / DIV$, где F_G — постоянная частота задающего генератора SPI, значение которой приведено в техническом описании на микросхему, DIV — делитель, который может быть задан программно. В ряде МК делитель DIV может быть выбран только из набора предопределенных значений. Например, в МК семейства M68HC11 это одно из четырех значений: 2, 4, 16, 32 [13]. При номинальном значении

F_G , равном 2 МГц, частота сдвига в этих МК может составлять 1000, 500, 125 или 62,5 кГц. В других МК значение DIV разрешается выбирать произвольно в пределах ограниченной разрядности. В семействе микроконтроллеров ADuC70xx [5] эта разрядность составляет один байт, величина делителя может задаваться в пределах от 5 до 255, что позволяет выбрать одно из 251 значения частоты сдвигов в пределах от 3482 до 81,2 кГц.

Поскольку действия на ведущем и ведомом устройствах тактируются одним и тем же сигналом SCK, к стабильности частоты этого сигнала не предъявляется никаких требований (за исключением ограничения сверху на длительности полупериодов, определяемого максимальной рабочей частотой более медленного абонента). Это позволяет использовать SPI в системах с низкостабильной частотой тактирования (тактовый генератор без кварцевой стабилизации, с временнозадающей RC-цепью), а также существенно облегчает программную эмуляцию ведущего устройства. Более того, при программной эмуляции вполне допустимо приостанавливать передачу «в середине» байта, если это допускает ведомое устройство.

Передача данных

SPI предполагает передачу бит за битом: от ведущего по линии MOSI и от ведомого (в обратную сторону) по линии MISO. Таким образом, возможен полнодуплексный режим, когда данные передаются в обоих направлениях одновременно и синхронно. Порядок следования битов в исходной версии SPI фирмы Motorola [13] можно определить как «начиная со старшего», однако в дальнейшем некоторые производители микроконтроллеров позволяли его задавать программно. Возможны реализации, в которых для передачи данных используется единственная двунаправленная линия; подобная конфигурация описана, в частности, в документе фирмы Freescale, полупроводниковой «дочки» фирмы Motorola [1].

Выбор ведомого — сигнал SS

Четвертый сигнал интерфейса — SS (Slave Select, выбор ведомого) — имеет основным на-

значением включение или выключение ведомого устройства сигналом от ведущего (от МК).

При активном (низком) уровне сигнала SS:

- Схемотехника ведомого устройства находится в активном состоянии.
 - Вывод MISO — в режиме «выход».
 - Тактовый сигнал SCK от ведущего воспринимается ведомым, вызывает считывание на входе MOSI значений передаваемых от ведущего битов и сдвиг регистра ведомого.
- При пассивном (высоком) уровне сигнала SS тактовый сигнал SCK и сигнал MOSI не оказывают никакого воздействия на ведомое устройство.

В простейших вариантах использования SPI сигнал SS может не употребляться. В других, более сложных периферийных устройствах сигнал SS не только используется, но выполняет еще и дополнительные функции, и этот вопрос будет обсуждаться далее.

Временная диаграмма сигналов

Возможно несколько вариантов диаграммы. Один из них изображен на рис. 2, а другие будут рассмотрены далее. Опишем диаграмму подробно.

До начала передачи пакета, левее момента времени А сигнал SS на выходе ведущего в высоком состоянии — рис. 2д), ведомый неактивен, его выход MISO в отключенном состоянии, а входы ведомого MOSI и SCK не воспринимают сигналов, формируемых ведущим. Тактовый сигнал на выходе ведущего в этот период имеет низкий уровень.

Для активации ведомого ведущий переводит в активное (низкое) состояние сигнал SS выбора ведомого (момент времени А). Ведомый включается, его вход тактирования способен воспринимать тактовый сигнал, а вывод MISO переходит в режим выхода. Тактовые импульсы пока еще не начались, выходные сигналы на линиях данных MOSI и MISO теперь отражают значения бита, который будет первым передан другому абоненту (интервал времени от А до 1).

Через некоторое время ведущий начинает формирование пакета сдвиговых импульсов — рис. 2а), сигнал SCK. Перепады тактового сигнала вызывают сдвиг содержимого сдвиговых регистров ведущего и ведомого,

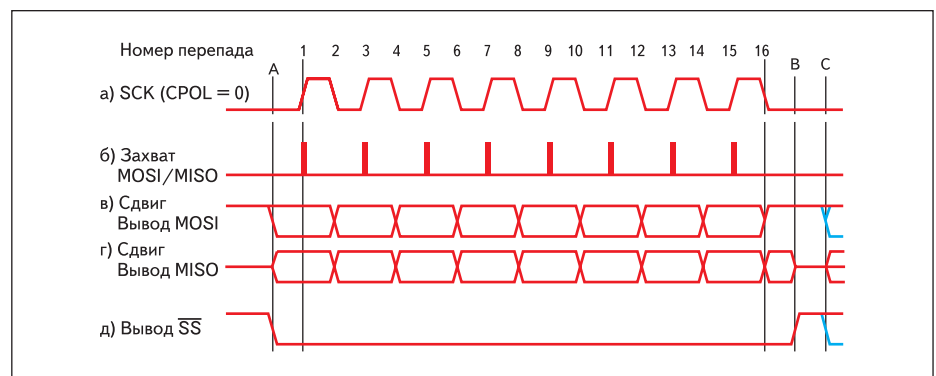


Рис. 2. Диаграмма SPI, режим 0

а также захват новых значений битов после сдвига. Моменты сдвига и захвата должны быть разнесены во времени, поэтому эти действия выполняются в интерфейсе SPI по разным перепадам тактовых импульсов. Диаграмма изображена в предположении, что захват производится по нарастанию тактового сигнала (по нечетным перепадам на рисунке), соответствующие моменты времени отмечены также на рис. 2б), а сдвиг в регистрах — по спаду (четные перепады). Таким образом, порядок событий при действии каждого тактового импульса можно описать как «захват, затем сдвиг» (Latch, then Shift).

После окончания обмена пакетами (правее перепада 16-тактной последовательности) выходной сигнал на каждой из линий данных MOSI и MISO отражает значение бита, который был передан первым. Передача пакета завершается снятием сигнала SS на выходе ведущего (момент В), что деактивирует ведомого абонента. Через некоторое время возможно начало передачи следующего пакета (момент С).

Сигнал SS может формироваться на выходе ведущего разными способами. В некоторых МК он формируется аппаратно: после того как будет программно выполнена запись передаваемого байта в регистр данных передатчика, выходной сигнал SS автоматически переключается в низкий уровень, а по окончании пакета — возвращается в исходное неактивное значение (высокий). Если в ведущем устройстве отсутствует аппаратно реализованное формирование сигнала SS, его формируют программно, используя для этого выход параллельного порта общего назначения. Однако при работе с некоторыми периферийными устройствами приходится формировать сигнал SS программно даже при наличии аппаратного формирования. Примеры будут приведены позже.

На этом рассмотрение базовых свойств интерфейса SPI заканчиваем, а более подробное их обсуждение продолжим на примерах использования.

Простейший пример использования SPI

Рассмотрим простой пример использования интерфейса SPI для увеличения количества линий дискретного вывода (параллельных портов) в МК, например, для управления группой светодиодов либо семисегментным ЖК-индикатором. Для этого в качестве ведомого можно взять обычный регистр сдвига, например 74НС164. Такой восьмиразрядный регистр имеет последовательный вход данных, вход тактирования (сдвига) и восемь параллельных выходов. Отдельный последовательный выход данных в этом регистре отсутствует, но для этого можно использовать выход старшего разряда D7. Схема подключения показана на рис. 3. В ней исполь-

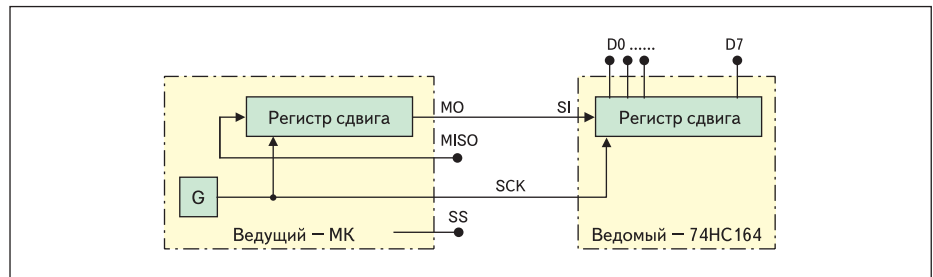


Рис. 3. Подключение к МК сдвигового регистра через SPI

зованы только две из четырех линий интерфейса SPI.

Схемотехника микросхемы 74НС164 такова, что как восприятие нового значения входного сигнала MOSI, так и сдвиг (в результате которого изменяются состояния параллельных выходов D7, ..., D0) происходит по нарастанию тактового сигнала SCK. Таким образом, используя два вывода (MOSI и SCK) можно получить дополнительно восемь цифровых выходов в системе.

Почти во всех микроконтроллерах с интерфейсом SPI линии этого интерфейса мультиплексированы с выводами параллельных портов общего назначения. Данный факт позволяет использовать незадействованные выходы микроконтроллера MISO и SS для других целей. Но этот же факт требует от программиста не забыть переконфигурировать соответствующие выходы параллельных портов для их использования в SPI.

Последовательное включение ведомых

При необходимости увеличения количества выходов можно соединить n регистров 74НС164 последовательно в цепочку, которая образует $8 \times n$ -разрядный сдвиговый регистр и предоставит $8 \times n$ выходов. На рис. 4 показано, как это выглядит для $n = 2$.

Для загрузки этого составного сдвигового регистра новым содержимым придется передать через SPI последовательно n байтов. Недостаток такого решения состоит в том, что во время передачи цепочки битов через многоразрядный сдвиговый регистр состояния его выходов будут переключаться, однако для управления набором светодиодов или многоразрядным семисегментным индикатором это вполне допустимо.

Аналогично можно подключить к МК через интерфейс SPI одну или несколько ИМС

регистров сдвига типа 74НС299, имеющих двунаправленные параллельные выходы, что даст дополнительное количество двунаправленных линий ввода/вывода.

Периферийные микросхемы с интерфейсом SPI могут быть устроены сложнее, нежели рассмотренные сдвиговые регистры, и требовать определенной последовательности сигналов SCK и MOSI на своих входах. Это относится:

- к полярности тактовых импульсов (или иначе, к уровню сигнала, которого требует на входе SCK периферийная микросхема в промежутках между передачами пакетов),
- а также к порядку событий «захват и сдвиг».

По этой причине в спецификациях интерфейса SPI различают четыре режима работы [14], диаграммы для этих режимов изображены на рис. 5.

Буквы S и L над графиками обозначают соответственно Shift (сдвиг) и Latch (захват). Рассмотренная ранее диаграмма соответствует Mode 0. Считается, что наиболее употребительны Mode 0 и Mode 3.

Для выбора в ведущем устройстве (в МК) одного из режимов, различающихся полярностью тактовых импульсов и их «фазой» (порядком действий), в одном из периферийных регистров подсистемы SPI обычно есть два управляющих бита: CPOL (полярность) и CPHA (фаза) — приведенные обозначения битов и обозначения режимов являются широко употребительными. Периферийные устройства (ведомые) чаще всего поддерживают единственный режим, и перед началом обмена ведущий (МК) должен быть настроен на этот режим.

Обратим внимание читателя на то, что в рассмотренном выше примере работы с обычным регистром сдвига одинаково пригоден любой из этих четырех режимов.

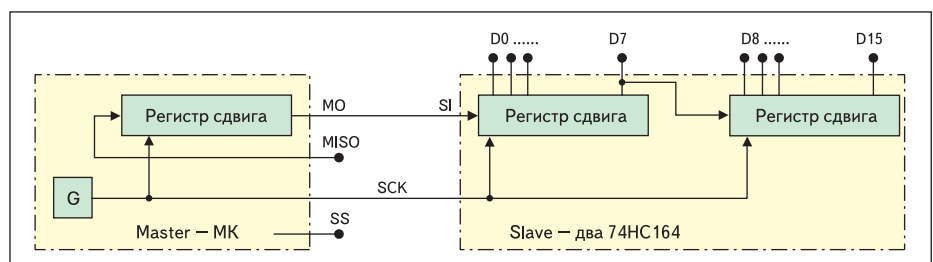


Рис. 4. Последовательное подключение к МК простых сдвиговых регистров как ведомых

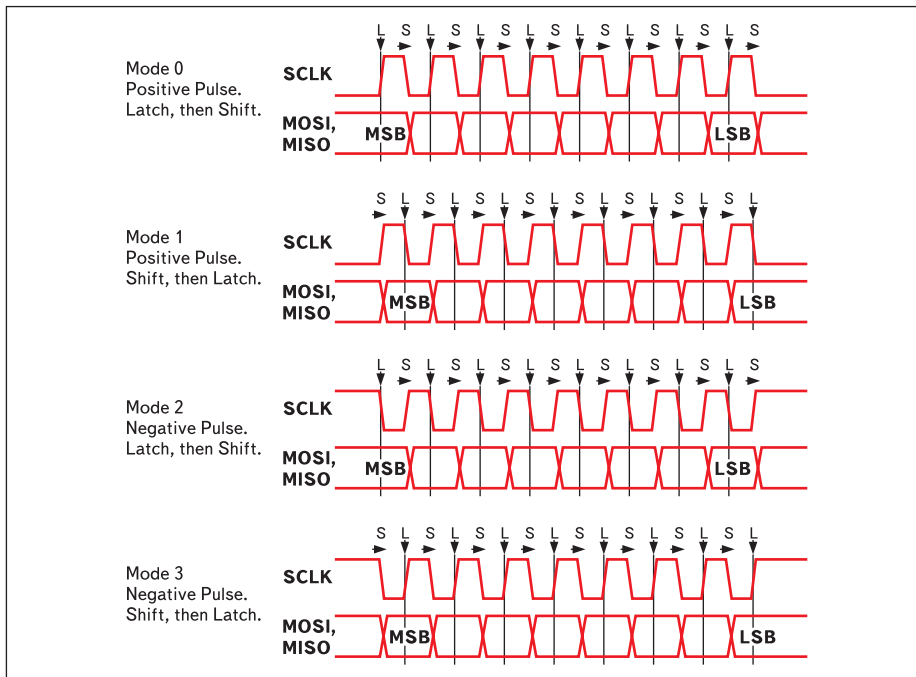


Рис. 5. Четыре режима работы интерфейса SPI

Возможные топологии систем связи на базе интерфейса SPI

Простейший случай: к ведущему подключен единственный ведомый, и необходим двусторонний обмен. В этом случае структура межсоединений может использовать трехпроводную схему (рис. 6).

На вход выбора SS на ведомом может быть постоянно подан активный (низкий) уровень, в то время как выход SS ведущего не используется вовсе и может быть задействован как вывод параллельного порта.

Интерфейс SPI позволяет подключить к одному ведущему несколько ведомых устройств, причем это может быть сделано различными способами. Первый из них позволяет реализовать радиальную структуру связи («звезда»), его принято считать основным вариантом использования интерфейса SPI. Соответствующая конфигурация соединений показана на рис. 7 (заимствован из [6]).

Как видно на рисунке, для обмена более чем с одним ведомым ведущий должен формировать соответствующее количество сигналов выбора ведомого SS. Это как раз тот

случай, когда сигналы SS приходится формировать программно, поскольку аппаратно в ведущем формируется не более одного сигнала SS. Если требуется обеспечивать обмен всегда только между ведущим и лишь одним из ведомых, программа на ведущем должна в промежутках между передачами данных держать все сигналы SS в неактивном (высоком) состоянии, а для обмена с одним из ведомых переводить в активный (низкий) уровень соответствующий сигнал SS (отсюда и название «выбор ведомого» — Slave Select). Выводы данных ведомых MISO соединены параллельно, но не «мешают» друг другу, поскольку находятся в отключенном состоянии, а перед началом обмена лишь один из этих выходов (на выбранном ведомом) переходит в активный режим. Обеспечение такого положения дел требует от программиста внимательности.

Однако может возникнуть желание в некоторых приложениях использовать «широковещательную» передачу, при которой данные, передаваемые ведущим, воспринимаются (может быть, всеми) ведомыми. Подобный

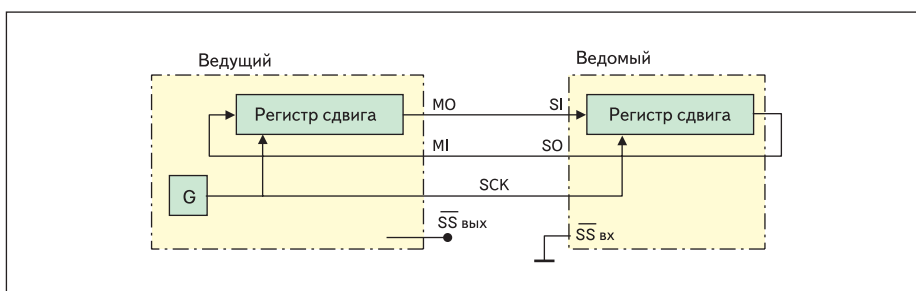


Рис. 6. Двухнаправленный обмен с одним ведомым без использования сигнала SS

режим возможен без опасности возникновения электрического конфликта на линии MISO в случае, если не требуется передавать данные из ведомых в ведущее устройство, тогда обратные линии данных из ведомых просто отсутствуют.

Если в качестве ведомых в рассматриваемой структуре используются тоже МК, в которых имеется возможность программно управлять режимом выходов MISO, то можно организовать широковещательную передачу и при наличии обратного канала (линии MISO). Для этого программы, работающие на микроконтроллерах — ведомых, должны постоянно держать выводы MISO в отключенном состоянии, при этом возможен режим широковещательной передачи от ведущего. Если же требуется получение данных от одного из ведомых, ведущий может сообщить ему об этом в широковещательном режиме, пошлав пакет с командным кодом-идентификатором, распознав который, один из ведомых должен перевести свой вывод MISO в активное состояние, после чего возможен двусторонний обмен данными между ведущим и выбранным ведомым. По окончании двустороннего обмена ведущий должен вернуть выбранного ведомого в исходное состояние. Иными словами, для получения такой возможности разработчику придется сконструировать некий протокол обмена, настроенный над программами SPI, в котором некоторые значения байтов (или их последовательностей) играют роль команд для ведомых МК. Этот вопрос будет рассмотрен более подробно в следующей статье цикла при обсуждении организации симметричной магистрали с несколькими ведущими на базе SPI.

Второй способ, позволяющий соединить одного ведущего абонента с несколькими ведомыми, — это структура «кольцо» (иногда эту структуру называют «цепочкой» или каскадным включением, в английской терминологии используется название Daisy-Chain) [6]. Эта структура изображена на рис. 8 (заимствован из [6]).

В ней для активации одновременно нескольких ведомых используется единственный сигнал SS, а выводы данных всех абонентов соединены последовательно и образуют замк-

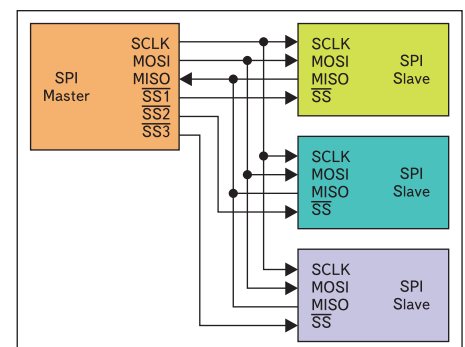


Рис. 7. Радиальная структура связи с несколькими ведомыми через SPI

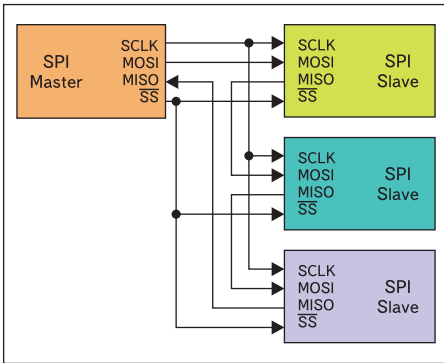


Рис. 8. Кольцевая топология связи через SPI

нутую цепь (кольцо). При передаче пакета от ведущего этот пакет получает первый ведомый, который, в свою очередь, транслирует свой пакет следующему ведомому, и т. д., наконец, последний в цепи ведомый передает свой пакет ведущему. Для того чтобы пакет от ведущего достиг определенного ведомого, ведущий должен выдать еще несколько пакетов. Вариант этой структуры (с незамкнутым кольцом) уже обсуждался (рис. 4).

Некоторые периферийные микросхемы с интерфейсом SPI проектируются специально для использования в цепочечных конфигурациях. Пример таких микросхем — цифро-аналоговые преобразователи AD5444 и AD5446 фирмы Analog Devices [15]. Их структура изображена на рис. 9.

Интерфейс этих приборов, совместимый с SPI, содержит 16-разрядный сдвиговый регистр с последовательным входом данных SDIN (MOSI), последовательным выходом данных SDO (MISO), входом тактирования сдвига SCLK (SCK) и входом управления SYNC. Три первых вывода по своим функциям полностью аналогичны выводам SPI (они указаны в скобках). Наличие выхода сдвигового регистра SDO позволяет подключить к МК (ведущему) несколько микросхем ЦАП по цепочечной структуре. Захват значения на входе регистра ведомого AD5444/6 происходит по спаду сигнала SCLK, изменение сигнала

ла на этом входе (сдвиг выходного регистра ведущего) должен происходить по нарастающей SCLK. Такая последовательность действий соответствует Mode 1 (рис. 5). Длина передаваемого в микросхему ЦАП пакета составляет 16 битов, из них два старших выполняют управляющие функции, а последующие 14 битов (в AD5446) или 12 битов (в AD5444) представляют значение отсчета, преобразуемое в напряжение. У программиста есть возможность переключить режим работы SPI в ведомом AD544x, задав значение старших двух битов пакета равным 11, в результате чего ведомый AD544x переключится в Mode 2, захват входного сигнала будет происходить по нарастанию сигнала SCLK.

А вот вход SYNC (SS) выполняет одновременно несколько функций. Во-первых, он низким уровнем активирует в микросхеме ЦАП функцию восприятия сигналов входной линии данных (MOSI) и линии тактирования SCLK (SCK). Кроме того, по окончании передачи пакета в микросхеме ЦАП переход в высокий уровень сигнала SYNC инициирует передачу полученного 16-битового пакета во входной буфер схемотехники ЦАП (что приведет к появлению нового значения напряжения на аналоговом выходе ЦАП синхронно во всех ведомых), а кроме того, отключает питание от интерфейсных цепей для уменьшения мощности, потребляемой микросхемой ЦАП.

Таким образом, штатный режим работы с цепочкой микросхем ЦАП предполагает установку активного уровня сигнала SYNC (SS), по которому включается питание интерфейсной схемотехники в ЦАП, затем последовательную передачу нужного количества 16-разрядных посылок с учетом порядка включенных в цепочку ведомых (при этом изменения выходных напряжений ЦАП не происходит) и, наконец, выключение активного сигнала SYNC, по которому изменятся одновременно выходные напряжения всех ЦАП в цепочке. Сигнал SYNC должен сохраняться активным (низким) без переключений до окончания передачи всех посылок.

Еще ряд примеров использования SPI

И в случае радиальной схемы с широкополосным подключением ЦАП дополнительные возможности были получены в результате наличия и использования высокоуровневого протокола, предполагающего передачу многобайтовых посылок. В примере с широкополосным разработкой такого протокола должна выполняться прикладным программистом, в то время как при использовании ИМС AD5444/AD5446 этот протокол специфицирован в их техническом описании.

Широко распространенный отладочный интерфейс JTAG также представляет собой пример реализации высокоуровневого протокола поверх SPI. JTAG предполагает соединение нескольких контролируемых устройств (ведомых) в кольцо и наличие в этом кольце единственного ведущего (JTAG-хоста) [6, 17, 18]. Длины посылок, используемые в JTAG, могут быть достаточно большими, однако их размер кратен 8 битам.

Рассмотрим еще один пример, в котором передача данных осуществляется только в обратном направлении — от ведомого к ведущему, а сигнал SS выполняет дополнительную функцию — запуск операции в периферийном устройстве.

Фирма Maxim¹ выпускает периферийные микросхемы MAX1240/1241 — 12-битовые АЦП с интерфейсом SPI [12]. Структурная схема устройства изображена на рис. 10.

Для подключения к МК через SPI в микросхеме имеются три вывода: вход тактирования SCLK (SCK), вход выбора кристалла CS (SS) и выход последовательных данных DOUT (MISO). Таким образом, в этом примере линия данных MOSI не используется, и соответствующий вывод в ведомом устройстве отсутствует. Временная диаграмма сигналов и внутренних состояний периферийной микросхемы АЦП изображена на рис. 11 (заимствован из [12]).

¹ www.maxim-ic.com

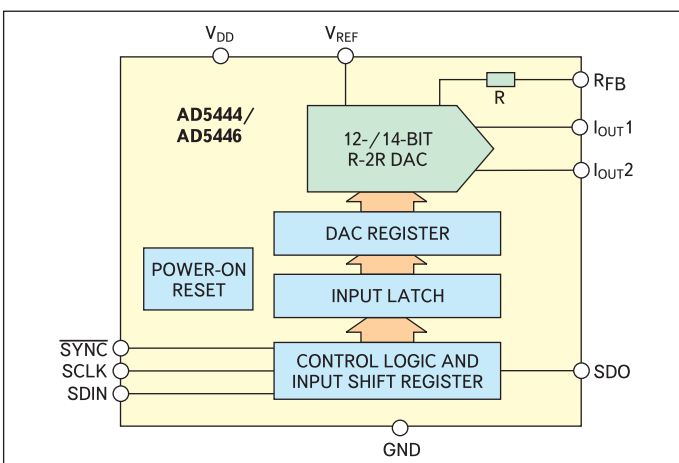


Рис. 9. Структура ЦАП AD5444/AD5446, приспособленная для SPI-цепочки

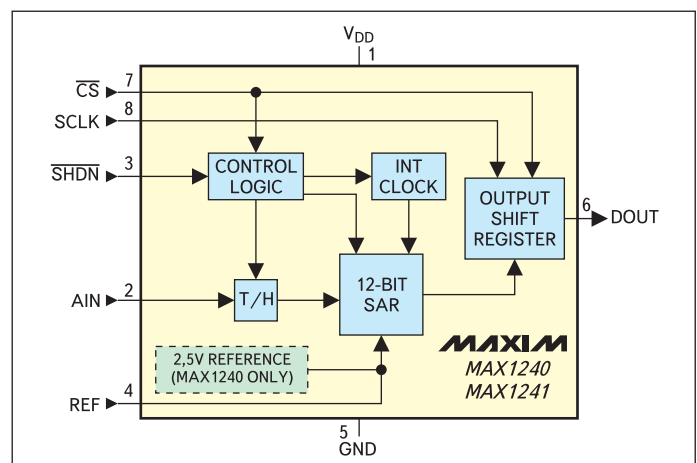


Рис. 10. Структура АЦП MAX1240/MAX1241 с запуском по перепаду сигнала CS

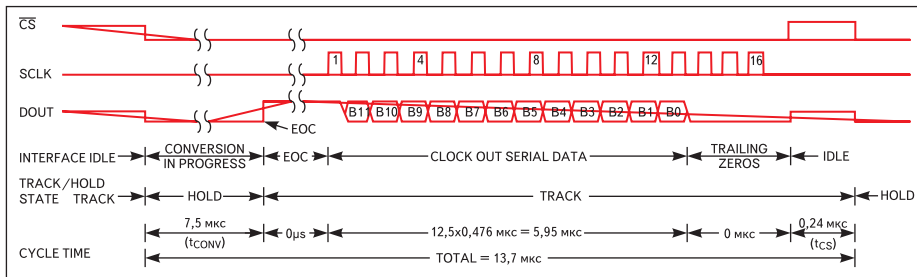


Рис. 11. Диаграмма работы АЦП MAX1240/1241 с интерфейсом SPI и с запуском от сигнала SS

Исходно при неактивном (высоком) уровне сигнала CS выход данных DOUT находится в отключенном (высокоимпедансном) состоянии.

По спаду сигнала CS из ведущего МК в ведомом MAX1240/1241 происходит:

- захват входного аналогового сигнала на встроенном устройстве выборки/хранения УВХ;
- запуск процесса аналого-цифрового преобразования с одновременным переходом сигнала DOUT в низкий уровень.

По окончании (через 7,5 мкс) процесса АЦ преобразования УВХ переходит из состояния «хранение» (HOLD) в состояние «выборка» (TRACK), а сигнал DOUT — в высокий уровень: последнее сигнализирует микроконтроллеру о готовом результате.

После этого ведущий МК может начать (немедленно или через некоторое время) считывание результата АЦ преобразования путем подачи последовательности тактовых импульсов на линию SCLK. Для его получения ведущий должен инициировать передачу двух байтов из ведомого. В них будет передано 16 битов, из которых первые 12 представляют результат, а последние четыре (нуля) должны быть просто отброшены. На диаграмме показано, что для повторения измерения ведущий должен снять активный уровень сигнала SS на предопределенное время (не менее 0,24 мкс), после чего процесс вновь можно запустить переключением сигнала SS в активный (низкий) уровень.

В этом примере сигнал CS (SS) выполняет в ведомом MAX1240/1 несколько функций:

- активирует внутреннюю схемотехнику SPI и выводит из отключенного состояния выход сигнала DOUT (MISO);

- запускает процесс аналого-цифрового преобразования в АЦП. Следует обратить внимание на то, что в течение времени, пока происходит АЦ-преобразование, сигнал SS должен иметь активное значение (низкий уровень), но сдвиговые такты должны отсутствовать. Это возможно лишь при программном управлении сигналом SS в ведущем устройстве.

Отдельно нужно сказать о сигнале SHDN в микросхемах MAX1240/MAX1241. Он предназначен для перевода периферийной микросхемы в режим энергосбережения. Для этой цели можно было бы использовать SS, однако внутренней схемотехнике преобразовате-

лей напряжения (для АЦП) требуется некоторое время для выхода на режим (около 4 мкс). Разработчики микросхемы могли бы реализовать внутреннюю задержку, но они предпочли использовать дополнительный управляющий сигнал. Это оправдано тем, что при необходимости непрерывного преобразования SHDN нужно включить лишь единожды, а запуск АЦП производить многократно. Для управления сигналом SHDN можно использовать выход порта общего назначения.

Что должна делать программа в ведущем (в МК)?

Вот какой может быть последовательность действий, выполняемая микроконтроллером в последнем примере, если его подсистема SPI имеет стандартную длину пакета 8 битов:

- Установить высокий уровень сигнала SHDN (программно).
- Подождать не менее 4 мкс (выход на режим преобразователей напряжения для АЦП).
- Установить низкий уровень сигнала SS (программно).
- Убедиться, что сигнал DOUT принял значение лог. 0 (программно или по прерыванию).
- Проверить состояние сигнала DOUT (признак окончания АЦ-преобразования) до тех пор, пока этот сигнал не переключится в лог. 1 (программно или по прерыванию).
- Инициировать передачу байта из периферийной микросхемы. Частота следования тактовых импульсов не должна превышать 2,1 МГц.

- Дождаться окончания приема первого байта (программно или по прерыванию).
- Прочитать и сохранить принятый первый байт.
- Инициировать передачу второго байта.
- Дождаться окончания приема второго байта (программно или по прерыванию).
- Прочитать и сохранить второй байт.
- Установить высокий уровень сигнала SS для того, чтобы в будущем можно было без задержек запустить следующее измерение.
- При необходимости немедленно выполнить еще одно измерение, установить низкий уровень сигнала SS (время нахождения сигнала SS в высоком уровне между двумя циклами измерения должно быть не меньше 0,24 мкс).
- Из полученных двух байтов взять первые 12 битов и записать их в переменную, предназначенную для результата.

Как видим, последовательность достаточно длинная, и нагрузка на МК оказывается значительной, если хочется выполнять измерения с частотой, близкой к максимальной.

И под конец еще более сложный пример ведомого устройства. Фирма Atmel производит микросхемы Flash-памяти с интерфейсом SPI [19]. Рассмотрим особенности подключения микросхемы AT45DB161B с организацией 2Mx8 бит. На рис. 12 изображена структурная схема устройства с указанием всех внешних выходов.

Сигналы SI (MOSI), SO (MISO), SCK (SCK), CS (SS) образуют интерфейс, совместимый с SPI. Обмен с внутренним массивом Flash осуществляется через два буфера размером (512+16) байт. Внутреннее состояние «микросхема занята выполнением операции» либо «операция закончена, микросхема свободна» отражается состоянием сигнала RDY/BUSY. При выполнении длительных операций состояние этого сигнала можно анализировать программно, либо использовать механизм аппаратных прерываний.

Особенностью микросхемы Flash-памяти является то, что при обмене используются многобайтовые пакеты переменной длины. Первый байт каждого пакета является коман-

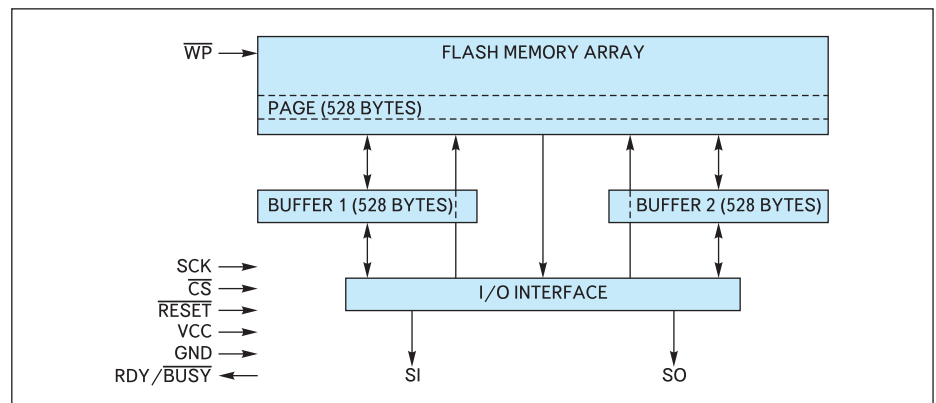


Рис. 12. Структура и выходы ИМС Flash-памяти AT45DB161B с интерфейсом SPI

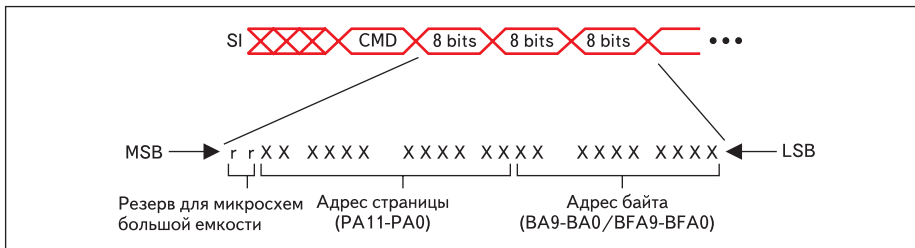


Рис. 13. Структура заголовка пакета, содержащего команду и адресную информацию

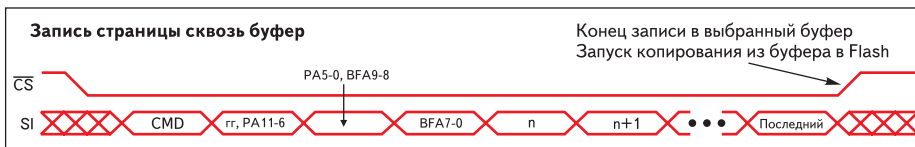


Рис. 14. Структура пакета при записи данных в массив Flash в ИМС AT45DB161B

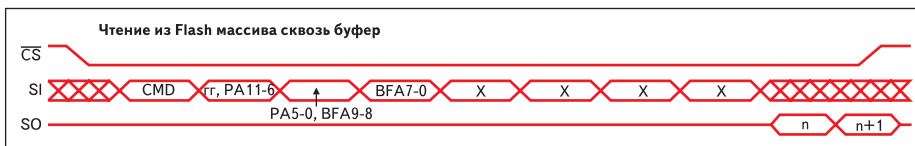


Рис. 15. Структура пакета, выполняющего чтение нескольких байтов из массива Flash AT45DB161B

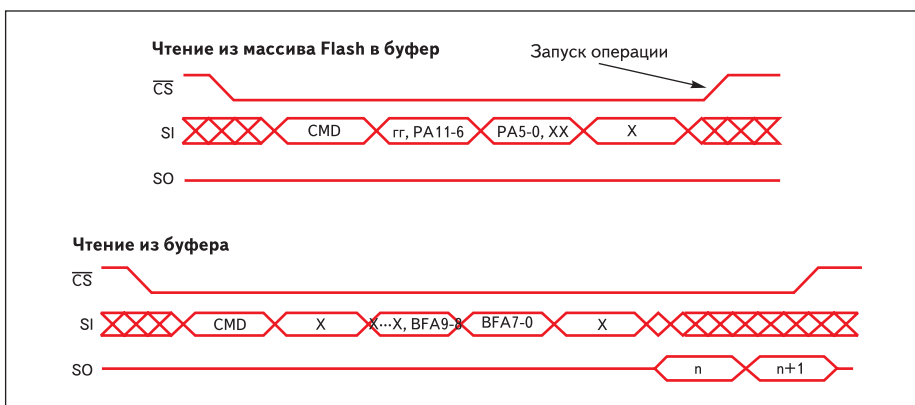


Рис. 16. Диаграммы двухэтапной операции чтения многих байтов из AT45DB161B

дой, а последующие байты содержат параметры или записываемые данные. Переход сигнала CS в активный (низкий) уровень отмечает начало пакета, а возврат сигнала CS в высокий уровень — конец пакета. Часть пакетов, предназначенных для чтения данных из Flash-памяти, имеют переменную длину. Структура заголовка команд чтения/записи изображена на рис. 13.

Как видно на рис. 13, первый байт содержит команду (код операции), а следующие три байта — два параметра: адрес страницы в Flash-массиве и адрес байта в странице, с которого требуется осуществить запись либо чтение. Параметры не выровнены по границам байтов, следовательно, заголовок должен быть сформирован программистом заранее, до начала передачи пакета.

Последующие байты пакета в командах записи содержат записываемую информацию, а в командах чтения не используются. При-

мер диаграммы для команды записи приведен на рис. 14.

Как видно на рис. 14, записываемые байты следуют в пакете за заголовком, их количество может быть различным. По окончании пакета ведущий должен снять активный уровень сигнала CS, и это запустит внутренний процесс копирования содержимого буфера в массив Flash.

Диаграмма для операции чтения из Flash-памяти приведена на рис. 15.

После окончания заголовка на линии SI микросхема выполняет (достаточно долго) операцию чтения, а ведущий должен продолжать сдвиги до тех пор, пока микросхема не закончит чтение и не начнет возвращать прочитанное по линии SO. Между сдвигами отдельных байтов допускаются промежутки. Это означает, что, запустив «длинную» операцию, ведущий (микроконтроллер) может некоторое время не производить обменов через SPI, а выпол-

нить другие действия, единственное условие состоит в том, что до окончания операции должен оставаться активным (низким) сигнал CS.

Поскольку длительность доступа к массиву Flash может быть значительной, микросхема допускает разделение операции на два этапа. Временные диаграммы процессов на линиях SPI для этого варианта действий приведены на рис. 16.

На первом выдается команда чтения из массива Flash в буфер, которая может выполняться в ведомом весьма долго, но командный пакет содержит всего четыре байта (верхние диаграммы), после чего сигнал CS может быть снят и ведущий получит возможность производить обмены с другим ведомым. После того как выполнение первой команды в микросхеме Flash будет закончено, вторая команда (нижние диаграммы) может произвести чтение из буфера, которое выполнится без ожидания.

Для того чтобы МК мог «узнать» об окончании длинной операции, микросхема AT45DB161 имеет специальный выход RDY/BUSY#, который можно либо опрашивать программно, либо использовать как источник запроса аппаратного прерывания.

Таким образом, мы видим, что в данном случае над интерфейсом SPI настроен достаточно сложный протокол, для его реализации ведущий должен иметь возможность формировать пакеты произвольной длины, в которых сигнал CS (SS) сохраняет свое активное значение.

Микросхемы AT45DB161B имеют (предоставляют пользователю) еще одну интересную возможность, а именно: выбирать режим работы интерфейса (сочетание полярности и фазы тактовых импульсов), не используя дополнительных линий. В каждом пакете схемотехника микросхемы анализирует состояние сигнала CSK в момент перехода сигнала CS в активный (низкий) уровень. Если уровень сигнала CSK низкий, микросхема настраивается на работу в режиме 0, если же высокий — в режиме 3 (рис. 17), моменты настройки отмечены жирными стрелками.

Рассмотренные примеры ведомых позволяют сформулировать определенные требования к структуре и функциям устройства, выполняющего роль ведущего. Чаще всего это подсистема SPI в составе МК.

О том, какой может быть организация ведущего абонента SPI, пойдет речь в следующей статье цикла. ■

Литература

1. Описание SPI от FreeScale: SPI Block Guide V03.06, FreeScale Semiconductor. Доступна здесь: http://www.freescale.com/files/microcontrollers/doc/ref_manual/S12SPIV3.pdf
2. MICROWIRE™ Serial Interface. National Semiconductor Application Note 452, Abdul Aleaf, Jan. 1992. www.national.com/an/AN/AN-452.pdf

3. SPI — Serial Peripheral Interface. Статья с описанием основ интерфейса и большой таблицей периферийных устройств с SPI.
<http://www.mct.net/faq/spi.html>
4. Николайчук О. Особенности микроконтроллерных архитектур с интерфейсом SPI // Схемотехника. 2005. № 12.
<http://www.silabs.ru/publication.php>
5. ADuC70xx Series: Precision Analog Microcontroller, 12-Bit Analog I/O, ARM7TDMI MCU Data Sheet (Rev B, 04/2007) (pdf, 1130 kB).
http://www.analog.com/static/imported-files/data_sheets/ADUC7019_7020_7021_7022_7024_7025_7026_7027_7028.pdf
6. http://en.wikipedia.org/wiki/SPI_bus
7. Описание QSPI от Freescale: QSM QUEUED SERIAL MODULE Reference Manual.
http://www.freescale.com/files/microcontrollers/doc/ref_manual/QSMRM.pdf
8. Описание SPI от Microchip: www.microchip.com/downloads/en/devicedoc/spi.pdf
9. Руководство пользователя по цифровым сигнальным процессорам семейства BlackFin. Разд. 10. Контроллеры SPI-совместимого порта.
http://www.analog.com.ru/pub_dsp.htm
10. Configuring Xilinx FPGAs with SPI Flash Memories Using CoolRunner-II CPLDs. App. Note XAPP800, XILINX, 2008.
www.xilinx.com/support/documentation/application_notes/xapp800.pdf
11. FT245BM — Преобразователь параллельного байтового интерфейса в интерфейс USB от фирмы Future Technology.
<http://www.ftdichip.com/Products/FT245BM.htm>
12. Микросхемы 12-бит АЦП с последовательным интерфейсом MAX1240/ MAX1241, техническое описание: MAX1240-MAX1241.pdf,
http://www.maxim-ic.com/quick_view2.cfm/qv_pk/1620

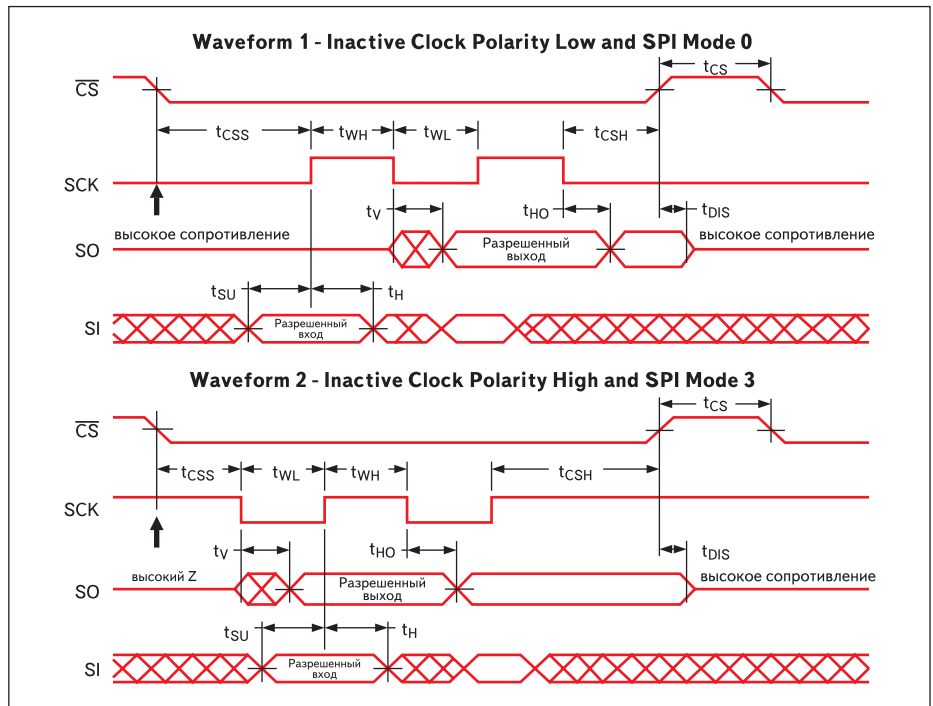


Рис. 17. Диаграммы настройки работы AT45DB161B в разные режимы тактирования SPI

13. M68HC11RM/D Rev.6 (M68HC11 Reference Manual, Motorola inc., 1989, ... 2002).
<http://www.pdf-search-engine.com/reference-manual-pdf.html> или здесь:
<http://www.tcc.edu/faculty/webpages/PGordy/EGR270/M68HC11RM.pdf>
14. About SPI. http://elm-chan.org/docs/spi_e.html
15. 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>
16. ADSP-BF533 Blackfin Processor Hardware Reference (Revision 3.3, September 2008), Документ bf533hwrRev3.3.pdf. <http://search.analog.com/search/default.aspx?local=en&query=bf531+hrm&contentType=Manuals>
17. <http://www.jtag-technologies.ru/Learn/Standards>
18. http://www.asset-intertech.com/pdfs/Boundary-Scan_Tutorial_2007.pdf
19. Микросхема Flash-памяти AT45DB161B — техническое описание. www.atmel.com/dyn/resources/prod_documents/DOC2224.pdf