

ОСОБЕННОСТИ МИКРОКОНТРОЛЛЕРНЫХ АРХИТЕКТУР С ИНТЕРФЕЙСОМ SPI

Автор: **Олег Николайчук**
onic@ch.moldpac.md

Источник: Статья на <http://www.electrosnab.ru>

Целью настоящей статьи является ознакомление читателей с архитектурой микроконтроллерных систем с интерфейсом SPI, особенностями и возможностями таких систем, а также - областями применения.

В течение последних пяти лет в развитии архитектуры микросхем микроконтроллеров ведущих мировых производителей наблюдается устойчивая тенденция роста количества встраиваемых интерфейсов. В отличие от микросхем микроконтроллеров предыдущих поколений, которые зачастую представляли собой «голое» микроконтроллерное ядро, оснащенное минимумом интерфейсов и цифровой периферии (как правило, интерфейс последовательного порта UART, несколько таймеров и подсистема прерываний), современные микроконтроллеры оснащаются достаточно большим набором дополнительных цифровых и аналоговых подсистем. В состав цифровых подсистем входят: достаточно большая Flash память программ (данных), имеющая размер от восьми до нескольких сотен килобайт; оперативная память с объемом до нескольких десятков килобайт; развитая система прерываний, обрабатывающая до тридцати источников прерываний; специализированные охранные таймеры WDT (Watchdog Times); различные типы генераторов и большое количество интерфейсов. Анализ оснащенности современных микроконтроллеров ведущих мировых фирм Silicon Laboratories Corp.[1], Atmel Corp.[2], MAXIM-Dallas Corp.[3] позволяет утверждать, что среди современных микроконтроллеров 100% оснащены одним и примерно 25% - двумя последовательными портами UART. Примерно 85% микроконтроллеров имеют в своем составе последовательный периферийный интерфейс SPI (Serial Peripheral Interface). Примерно 40% оснащаются интерфейсом системной управляющей шины (SMBus, совместимым с I²C). Менее 20% микроконтроллеров оснащаются всеми другими типами интерфейсов, такими как: универсальная последовательная шина - USB (Universal Serial Bus); интерфейс сети микроконтроллеров - CAN (Controller Area Network), однопроводный двунаправленный интерфейс MicroLAN и ряд других.

При этом следует отметить ряд достаточно интересных особенностей, связанных с использованием различных интерфейсов. Если еще несколько лет назад практически каждый из микроконтроллеров оснащался интерфейсом UART, то в настоящее время этот интерфейс уверенно теряет свои лидирующие (в прошлом) позиции. Для связи с персональным компьютером все чаще используется универсальная последовательная шина – USB, поскольку имеет более высокие скорости передачи и обеспечивает целый ряд других дополнительных функциональных возможностей. Для организации микроконтроллерных сетей используются интерфейсы: CAN, SPI, SMBus и MicroLAN. Для внутрисистемных (внутриплатных) связей в современных контроллерах наиболее часто используются интерфейсы SPI и SMBus.

Предметом настоящей статьи являются особенности архитектуры современных микроконтроллерных систем, оснащенных последовательным периферийным интерфейсом SPI. При этом под термином «архитектура» (микроконтроллерных систем) мы будем понимать совокупность основных подсистем (устройств, функциональных узлов и элементов) микроконтроллерной системы, а также структуру основных управляющих и информационных связей между ними, обеспечивающих их функционирование. Иными словами, это взгляд на подсистемы микроконтроллерной системы и связи между ними, как на единое целое, в противоположность рассмотрению этих подсистем по отдельности (как отдельные структуры).

Напомним читателю, что обычно при аппаратной реализации интерфейса SPI, он доступен программисту через четыре SFR регистра. Названия этих регистров могут незначительно отличаться для микроконтроллеров различных производителей, однако их функциональное назначение при этом остается неизменным. Далее, в рамках настоящей статьи, мы будем использовать наименования SFR регистров, применяемых в документации фирмы Silicon Laboratories Corp.[1], которая в настоящее время является лидером в развитии интерфейса SPI.

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

Интерфейс SPI любого микроконтроллера может быть запрограммирован для работы как в качестве ведущего, так и в качестве ведомого узла.

Если интерфейс запрограммирован как ведущий, он может работать на максимальной скорости передачи данных (бит/сек), равной половине тактовой частоты.

Если интерфейс запрограммирован для работы в качестве ведомого, его максимальная скорость в полнодуплексном режиме равняется одной десятой тактовой частоты.

Подразумевается, что источником синхронизации в обоих случаях является системный генератор тактовой частоты. Если ведущий интерфейс вырабатывает SCK, NSS и последовательные входные данные асинхронно, максимальная скорость передачи должна быть меньше одной десятой тактовой частоты.

Существует еще один особенный режим, когда ведущий должен только передавать данные ведомому (полудуплексный режим), и не должен принимать данные от ведомого. В этом случае максимальная скорость передачи составляет одну четвертую от системной тактовой частоты, при синхронном режиме работы.

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

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

Линия MISO (Master-In, Slave-Out) - входная линия данных ведущего интерфейса и выходная линия данных ведомого интерфейса. Линия предназначена для передачи данных от ведомого интерфейса к ведущему. Данные передаются байтами, побитно, начиная со старшего бита. Следует помнить, что вывод MISO ведомого интерфейса находится в высокоимпедансном состоянии, если ведомый интерфейс не выбран по линии NSS

Линия NSS (Slave Select) - линия выборки ведомого, предназначена для выборки (низким логическим потенциалом) ведомого интерфейса ведущим.

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

В базовой версии сети на базе SPI интерфейсов только один интерфейс может быть ведущим. Интерфейс устанавливается в режим ведущего установкой флага MSTEN (Master Enable Flag) - бита SPI0CN.1. Если интерфейс установлен в режим ведущего, то запись байта данных в регистр данных SPI0DAT приводит к началу передачи. Ведущий интерфейс немедленно побитно сдвигает данные и выдает их на линию MOSI в сопровождении тактовых импульсов на линии SCK. После завершения передачи устанавливается флаг SPIF (SPI0CN.7). Если разрешены прерывания, выдается соответствующее прерывание. Кроме того, интерфейс может быть запрограммирован на выдачу от одного до восьми битов для осуществления связи с SPI приборами, имеющими различную длину слова. Длина передачи (количество передаваемых битов) может быть задана битами SPIFRS в регистре конфигурации SPI0CFG.[2:0] (SPI Configuration Register).

Выше уже отмечалось, что интерфейс может работать в полнодуплексном режиме, это означает, что возможна одновременная передача данных по линиям MOSI от ведущего к ведомому, и MISO от ведомого к ведущему. Данные, полученные от ведомого интерфейса, заменяют данные в регистре данных ведущего интерфейса. Этот регистр дважды буферизирован на ввод, но не на вывод. Т.е. если в регистр данных SPI0DAT производится попытка записи данных во время передачи предыдущего байта, устанавливается флаг WCOL (SPI0CN.6) и попытка записи игнорируется. Таким образом, текущая передача данных продолжается непрерывно. Чтение из регистра данных SPI0DAT приводит к чтению приемного буфера. Если прием не закончен, устанавливается флаг RXOVRN (SPI0CN.4). Новые данные не передаются в регистр чтения, пока предыдущий принятый байт не будет прочитан. Очевидно, что при задержке чтения принятых байтов может произойти потеря данных. Если SPI интерфейс настроен, как ведущий, он будет работать в режиме ведомого.

Не смотря на то, что базовая версия интерфейса SPI разработана для микроконтроллерных систем с одним ведущим, возможен все же режим сети со многими ведущими. Флаг MODF (SPIOCN.5 - Mode Fault flag) устанавливается в логическую единицу, если интерфейс определен как ведущий (MSTEN=1) и

вывод NSS переведен в низкий логический уровень, т.е. SPI интерфейс пытаются использовать в качестве ведомого. Если при этом установлен флаг MODF, биты MSTEN и SPIEN в регистре управления SPI стираются автоматически аппаратно, переводя интерфейс в автономное состояние. Таким образом, в системе с многими ведущими ядро может определить свободна ли шина путем опроса флага SLVSEL (SPIOCN.2) перед тем, как установить MSTEN флаг (т.е. определить интерфейсу режим ведущего) и инициализировать обмен.

Итак, интерфейс SPI содержит всего четыре SFR регистра:

1. SPIODAT - регистр данных;
2. SPIOCKR - регистр управления скоростью;
3. SPIOCFG - регистр конфигурации;
4. SPIOCN - регистр управления шиной SPI.

Первые два регистра, назначение которых очевидно из названия, не представляют особого интереса при рассмотрении архитектурных особенностей интерфейса. Единственное, что следует отметить, так это то обстоятельство, что код, записываемый в регистр SPIOCKR - управления скоростью, позволяет определить скорость работы интерфейса SPI (частоту тактовых импульсов F_{SCLK}) по следующей формуле:

$$F_{SCLK} = 0.5 * SYSCLK / (SPIOCKR + 1)$$

Из приведенной формулы следует, что максимальная тактовая частота SPI интерфейса может быть равной половине системной тактовой частоты в режиме ведущего (мастера). В режиме ведомого скорость передачи интерфейса SPI определяется тактовой частотой ведущего интерфейса SPI, т.е. мастера.

Третий регистр - регистра конфигурации интерфейса - SPIOCFG. Этот регистр содержит биты СКРНА (SPI Clock Phase) - управления фазой тактирования, и СКПОЛ (SPI Clock Polarity) - управления полярностью тактирующих импульсов. Эти биты позволяют выбрать фазу и полярность импульсов тактирования.

Теперь уместно заметить, что кроме первоначального базового интерфейса PSI во многих современных микроконтроллерах используется также и расширенный интерфейс PSI.

В базовом варианте интерфейса SPI регистр конфигурации SPIOCFG содержит также биты: BC2- BC0 (SPI Bit Count) – индицирующие номер текущего передаваемого бита, и биты SPIFRS2- SPIFRS0 (SPI Frame Size), определяющие размер фрейма (длины передаваемого слова). В расширенном интерфейсе эти биты отсутствуют, т.к. при современных высоких скоростях передачи определение текущего передаваемого бита и изменение формата фрейма теряет всякий смысл. Вместо этих битов расширенный интерфейс SPI имеет следующие конфигурирующие биты, позволяющие оптимизировать работу систем со многими ведущими интерфейсами: MSTEN (Master Mode Enable) – бит разрешения режима ведущего; NSSIN (NSS Instantaneous Pin Input) – бит, индицирующий состояние входа NSS в момент чтения; RXBMT (Receive Buffer Empty) – бит, индицирующий, что регистр чтения пуст; SLVSEL (Slave Selected Flag) – бит состояния ведомого; SPIBSY (SPI Busy) – бит занятости интерфейса SPI, в некоторой степени заменяет биты SPI Bit Count базового интерфейса; SRMT (Shift Register Empty) - бит, индицирующий освобождение сдвигового регистра данных. Приведенные биты позволяют диагностировать интерфейс при шинной организации системы со многими ведущими (см. ниже).

Четвертый регистр SPIOCN - регистр управления интерфейса SPI. Он имеет ряд битов управления, общих как для базовой, так и для расширенной версий: MODF (Mode Fault Flag) - флаг ошибки режима, индицирующий несоответствие назначенного режима (бита ведущего MSTEN=1) и состояния входа NSS=0; RXOVRN (Receive Overrun Flag) – бит, индицирующий случай, если начат прием следующего байта при непрочитанном предыдущем байте, т.е. бит индицирует рассинхронизацию чтения; WCOL (Write Collision Flag) - флаг рассинхронизации записи, устанавливаемый в случае, если совершена попытка записи байта в регистр во время незавершенной передачи предыдущего байта; SPIF (SPI Interrupt Flag) - флаг прерывания интерфейса SPI, устанавливаемый аппаратно после завершения передачи; SPIEN (SPI Enable) - бит разрешения работы интерфейса SPI. Кроме этого, базовый интерфейс имеет ряд битов, которые

в расширенном интерфейсе были переведены в регистр конфигурации SPI0CFG: TXBSY (Transmit Busy Flag) - бит занятости, - эквивалентен новому биту SPIBSY (SPI Busy); а также биты SLVSEL и MSTEN (смотри выше описание регистра конфигурации SPI0CFG). Расширенный интерфейс SPI имеет следующие конфигурирующие биты: NSSMD1- NSSMD 0 (Slave Mode Select) – биты выбора режима ведомого интерфейса: 3-хпроводный режим ведущего или ведомого; 4-хпроводный режим ведомого или режим с многими ведущими (NSS – всегда в режиме входа); 4-хпроводный режим одного ведущего.

Таким образом, очевидно основное отличие расширенного интерфейса SPI от базового. В расширенном варианте предусмотрена работа шины SPI в многоконтроллерном режиме, что выразилось в расширении средств диагностики ошибок и возможности управления сигналом NSS.

Классическая архитектура SPI шины для базового варианта интерфейса приведена на рис.1. Это так называемая классическая 4х – проводная структура. Один ведущий в ней управляет несколькими (N) ведомыми. Все ведомые подключены параллельно на линиях SCLK, MISO и MOSI шины SPI. Выборка одного из ведомых происходит с помощью одной из линий портов ввода/вывода, которая соединяется с входом NSS соответствующего ведомого. Обычно такая архитектура используется для построения микроконтроллерных систем с одним микроконтроллером, выполняющим роль ведущего и рядом периферийных микросхем, выполняющих роль ведомых. В качестве периферийных микросхем может быть использован ряд современных микросхем, оснащенных интерфейсом SPI: таймеров реального времени RTC, аналого-цифровых преобразователей ADC, цифро-аналоговых преобразователей DAC, различных микросхем памяти и т.п.

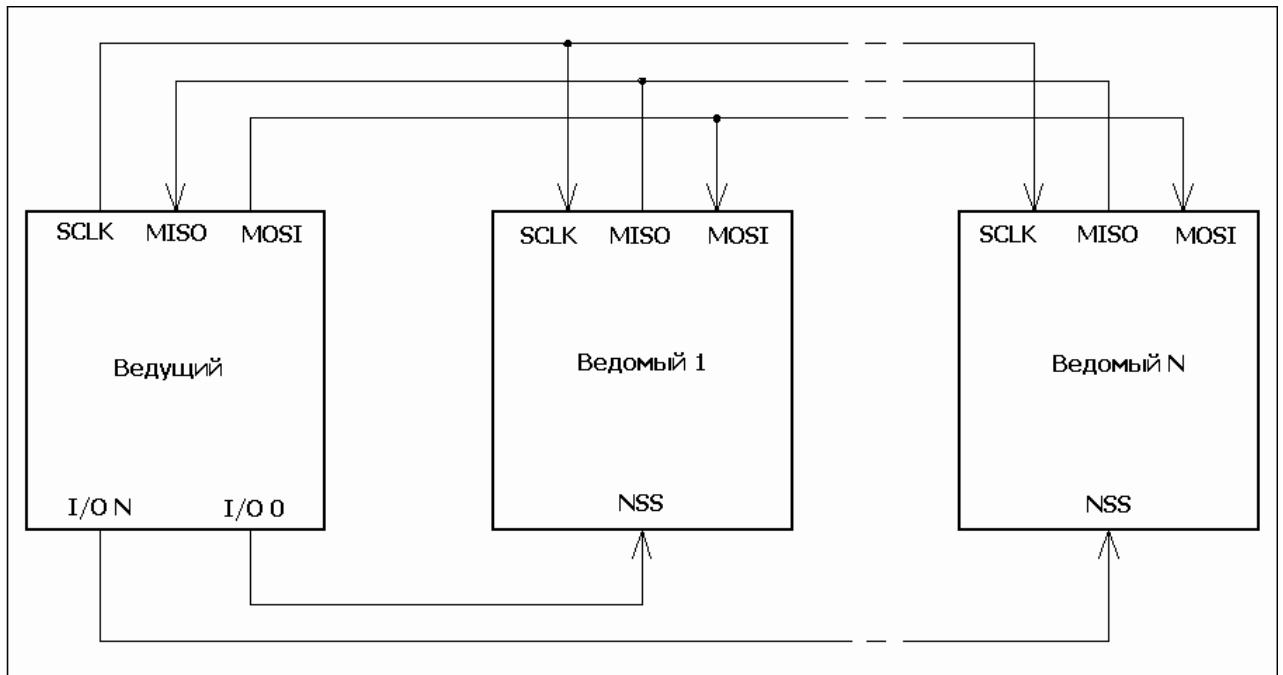


Рис.1. Архитектура 4х–проводной SPI шины с одним ведущим и несколькими ведомыми

Частным случаем предыдущей архитектуры является так называемая 3х-проводная структура, приведенная на рис.2. Ее отличие состоит в том, что ведомая микросхема постоянно выбрана, и следовательно, необходимость в четвертой линии интерфейса SPI отсутствует.

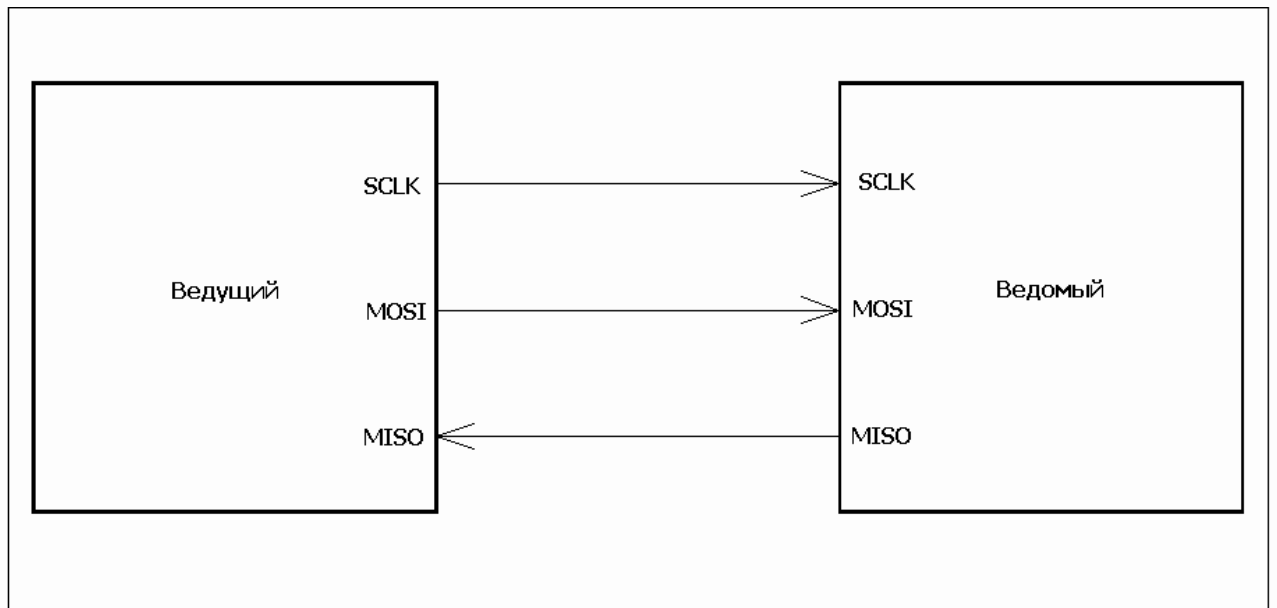


Рис.2. Архитектура 3х-проводной SPI шины с одним ведущим и ведомым.

Расширенный интерфейс позволяет организовать двухпроцессорную 4х-проводную архитектуру, приведенную на рис.3. В этом случае каждый из двух узлов может выполнять роль ведущего. На практике такая архитектура используется достаточно редко, ввиду алгоритмической сложности диагностики состояний коллизий (конфликтов, вызванных одновременной передачей на шину двух ведущих узлов). Более широко распространена более простая в понимании и диагностике конфликтов, так называемая архитектура с 4х-проводной «перекрестной» выборкой, приведенная на рис.4. В этом случае для выборки каждого из узлов используется линия ввода/вывода другого узла, работающая только на вывод, а не на ввод/вывод, как в предыдущей структуре.

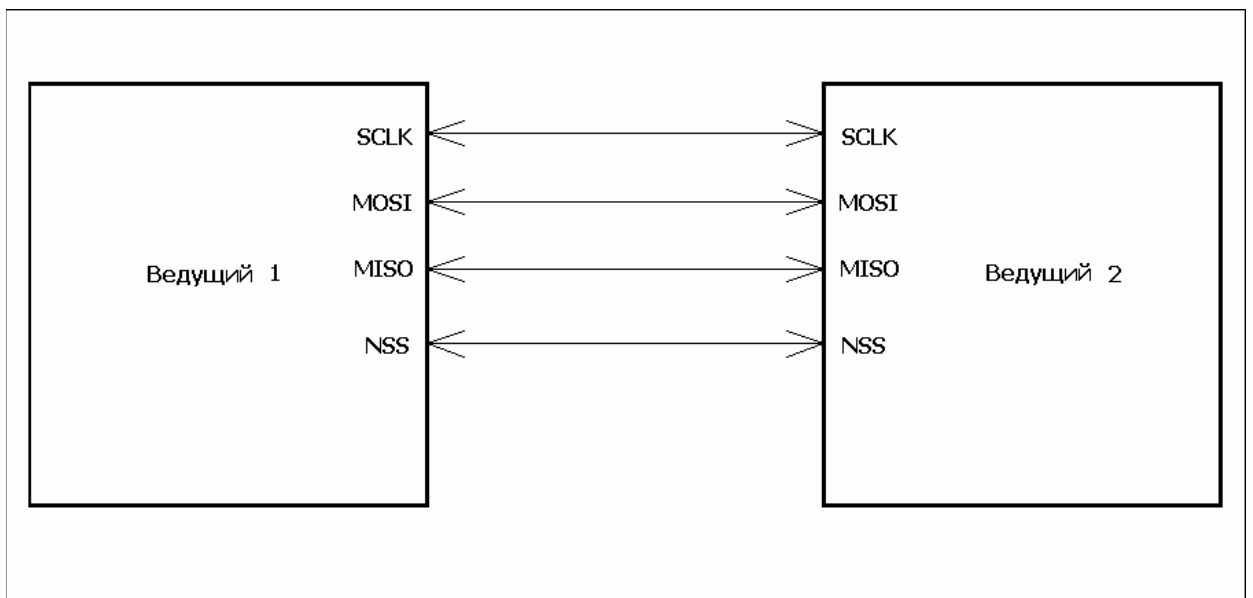


Рис.3. Архитектура 4х-проводной SPI шины с двумя ведущим

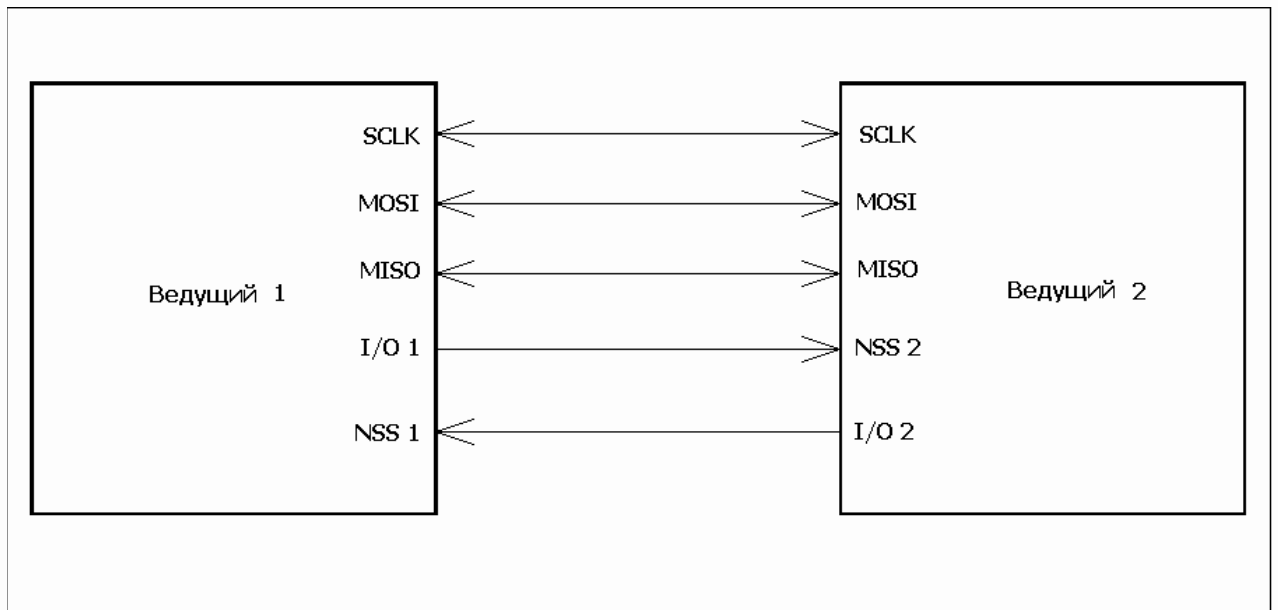


Рис.4. Архитектура 4х-проводной SPI шины с двумя ведущим и «перекрестной» выборкой

Как уже упоминалось выше, чисто двухпроцессорные архитектуры используются достаточно редко, например, в случае, если один из микроконтроллеров предназначен для ввода и обработки сигналов, а второй – для организации интерфейса с персональным компьютером или локальной сетью. В этом случае и один, и другой микроконтроллеры должны иметь возможность инициировать процедуру обмена данными, а следовательно, должны иметь возможность функционирования в режиме ведущего. Более распространенный случай, это использование двухпроцессорной структуры с разделяемым ресурсом, например памятью. Пример такой архитектуры приведен на рис.5.

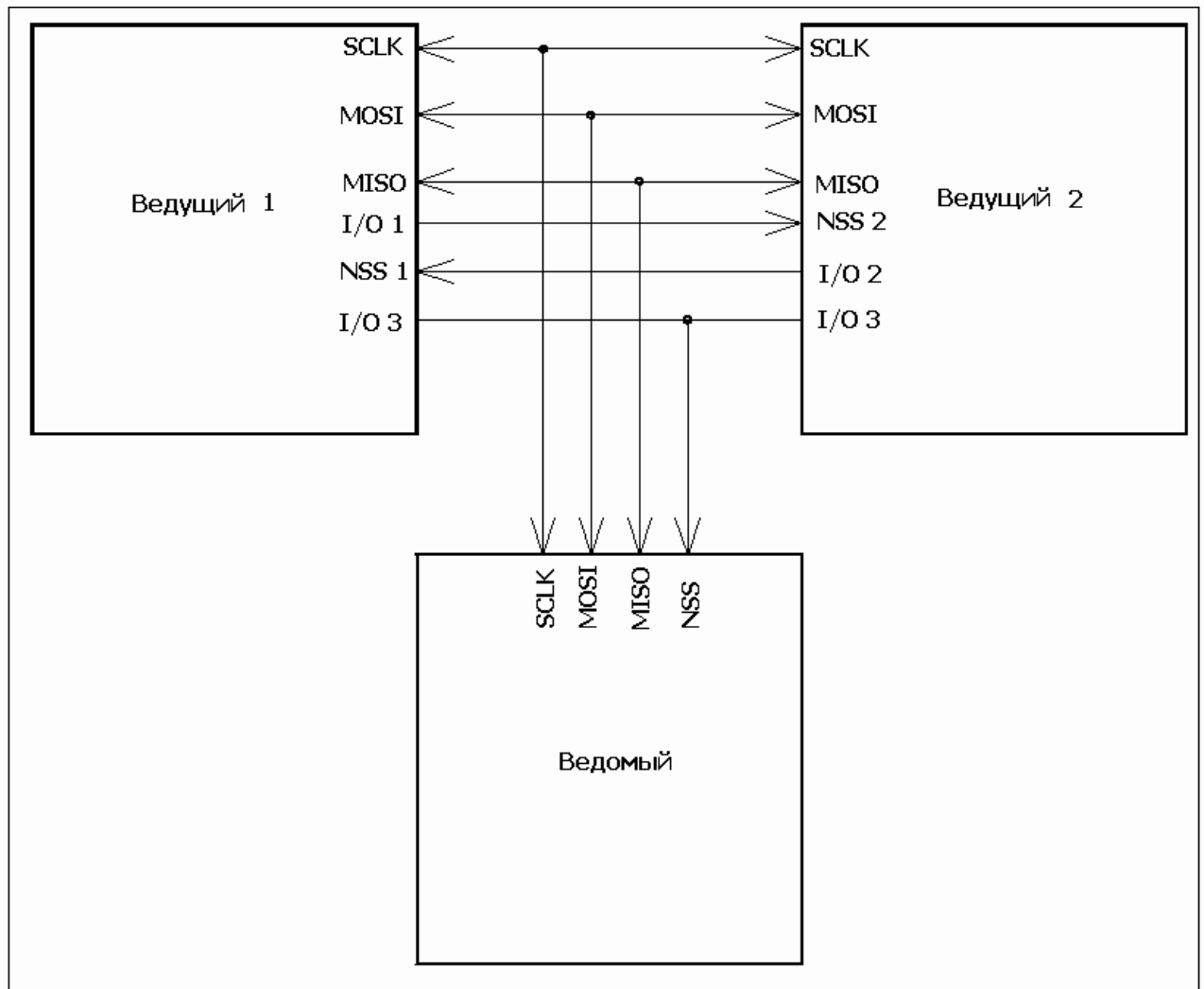


Рис.5. Двухпроцессорная архитектура с разделяемым ресурсом

В приведенной на рис.5. архитектуре имеется два ведущих и один ведомый. Связи между ведущими минимизированы и соответствуют архитектуре 4х-проводной SPI шины с двумя ведущим и «перекрестной» выборкой. Линии основных сигналов SPI шины подключены также к общему ведомому, а вот для выборки общего ведомого используется по одной дополнительной линии ввода/вывода от каждого из ведущих, объединенных по «монтажному ИЛИ». Архитектура в принципе работоспособна, однако анализ занятости SPI шины и диагностики коллизий в ней еще более затруднен. Это объясняется необходимостью последовательного опроса как минимум двух линий выборки (индивидуальной иницирующего обмен контроллера и общего ресурса) перед занятием шины, а следовательно и увеличением вероятности возникновения коллизии. Для упрощения процедур обмена рекомендуется ввести еще две линии флагов запроса шины ID1 и ID2, как показано на рис.6.

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

В заключении отметим как минимум два варианта применения последних двух архитектур.

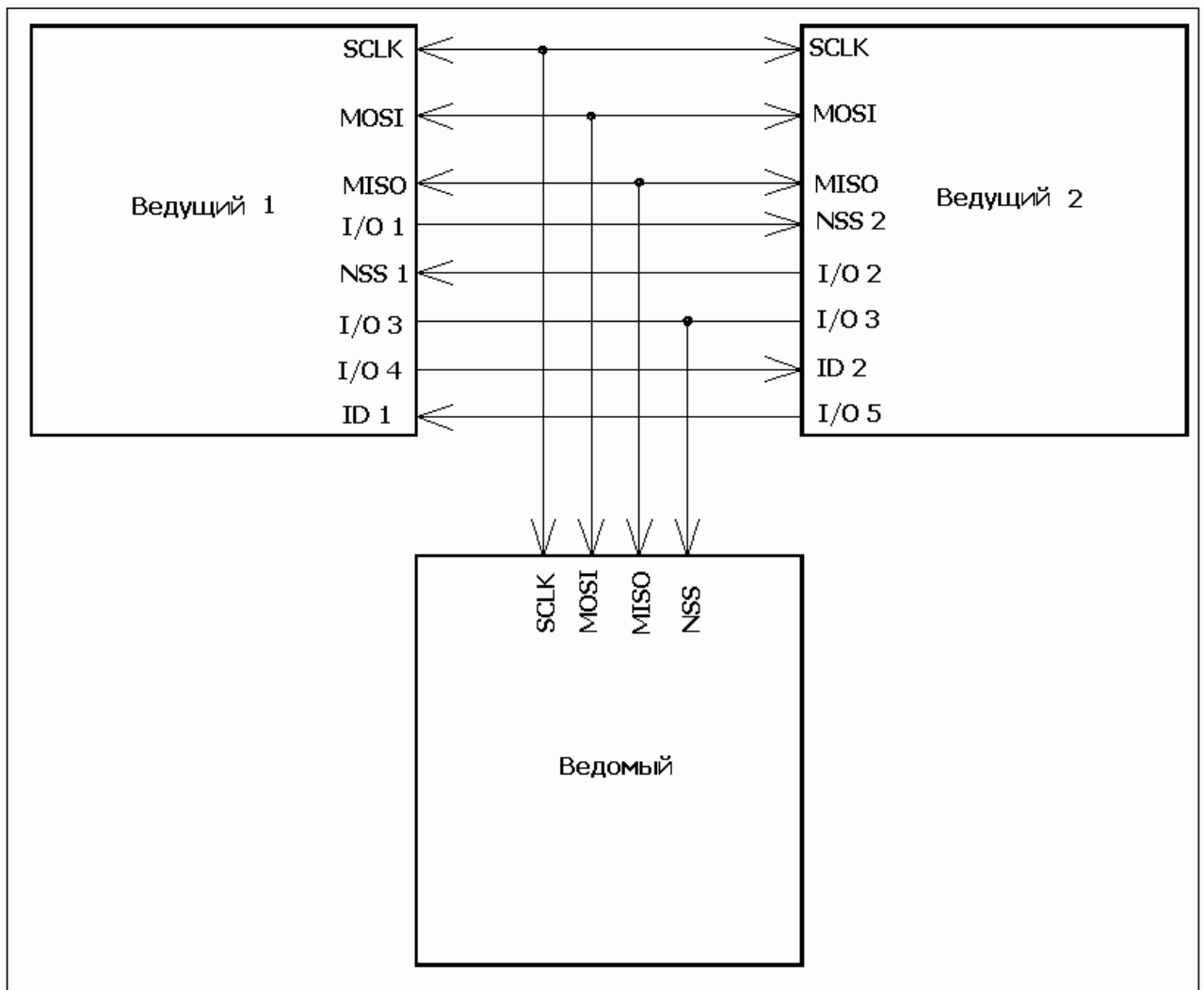


Рис.6. Двухпроцессорная архитектура с разделяемым ресурсом и линиями запроса шины

Вариант первый - двухпроцессорная архитектура, содержащая первый быстродействующий микроконтроллер C8051F121 и второй микроконтроллер C8051F320 с интерфейсом USB для связи с персональным компьютером. В качестве общего разделяемого ресурса используется DataFlash память большого объема для накопления данных, например AT45DB642. Первый микроконтроллер измеряет с помощью встроенных аналого-цифровых преобразователей ADC данные с датчиков объекта наблюдения или управления, обрабатывает их и записывает в DataFlash память, а второй контроллер при определенных условиях передает накопленные данные в персональный компьютер по интерфейсу USB. Второй вариант - аналогичен первому, но вместо быстродействующего первого микроконтроллера используется микроконтроллер C8051F350 со встроенными 24-разрядным аналого-цифровым преобразователем.

Конечно, читатель может возразить, что приведенные в качестве примера архитектуры необходимы только до тех пор, пока первый микроконтроллер (сверхбыстродействующий или с высокоточным ADC) не будет оснащен собственным USB интерфейсом. Конечно, это утверждение соответствует истине. Но, во-первых, пока таких микроконтроллеров в природе не существует, а во-вторых, мы рассмотрели всего лишь два примера из множества возможных.

Литература:

1. <http://www.silabs.com>
2. <http://www.atmel.com>
3. <http://www.maxim-ic.com>