

УНИВЕРСИТЕТ ИТМО

Системы ввода/вывода и
периферийные устройства

Лекция 3. Аппаратные интерфейсы вычислительных систем

Быковский С.В., Кожоголова А.К., Поторокин О.И.

e-mail: sergei_bykovskii@corp.ifmo.ru

Санкт-Петербург, 2016



Аппаратные интерфейсы | уровни описания

На физическом уровне определяются:

- Набор сигнальных линий
- Типы соединительных элементов
- Характеристики сигналов
- Нагрузочная способность линий связи
- Физическая топология
- Максимальная длина линий связи
- Требования к помехоустойчивости и заземлению

На логическом уровне определяются:

- Формат пакетов данных
- Логическая топология
- Протокол передачи данных
- Система адресации



Основные характеристики интерфейсов

1. Скорость передачи данных (пропускная способность)
2. Способ передачи слова данных (параллельный, последовательный)
3. Протяженность линий связи
4. Количество подключаемых устройств
5. Топология
6. Наличие синхронизации (синхронный, асинхронный)
7. Возможные направления передачи данных (симплексный, полудуплексный, дуплексный)



Типы передаваемых данных

По функциональному назначению различают:

- Прикладные данные
- Адрес
- Данные управления

Различные типы данных могут передаваться как на физическом, так и на логическом уровне.

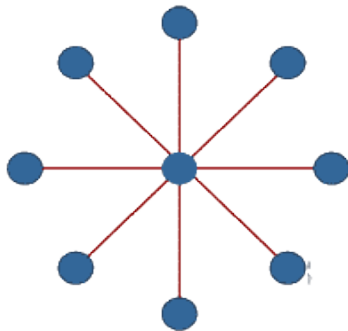
На физическом уровне для разных типов данных выделяют отдельные сигнальные линии.

На логическом уровне для разных типов данных выделяют специальные поля на уровне пакетов передачи данных.

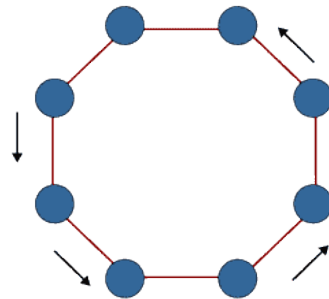


Схемы подключения устройств (топология)

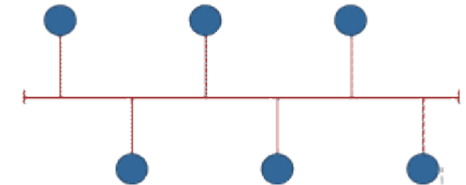
Звезда (радиальная топология)



Кольцо

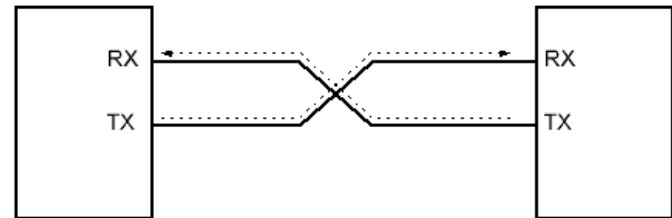
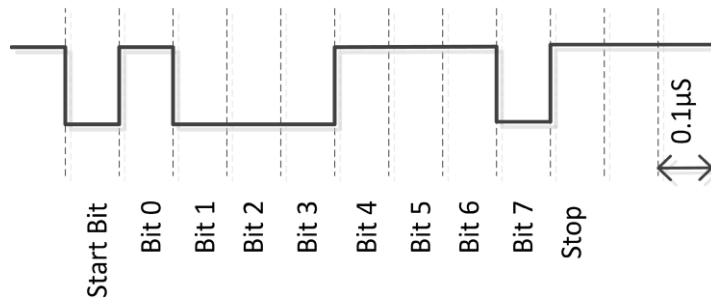


Общая шина

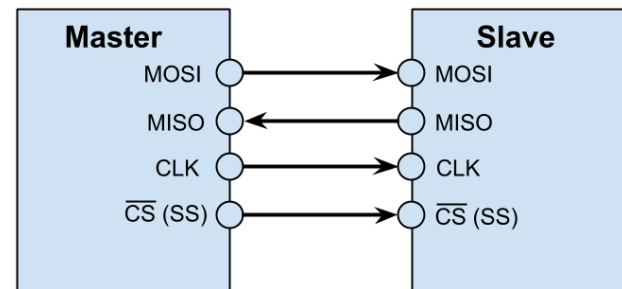
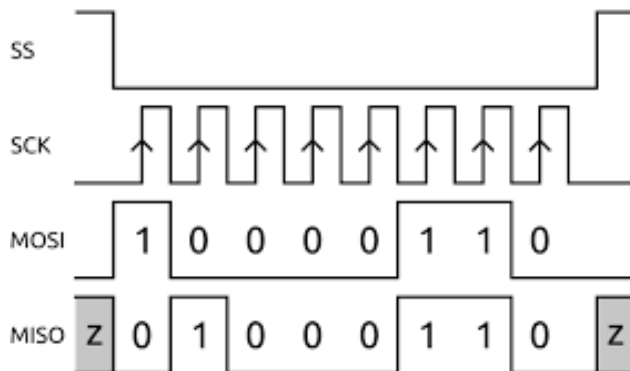


Виды синхронизации

Асинхронный интерфейс (RS-232, USB, SATA)

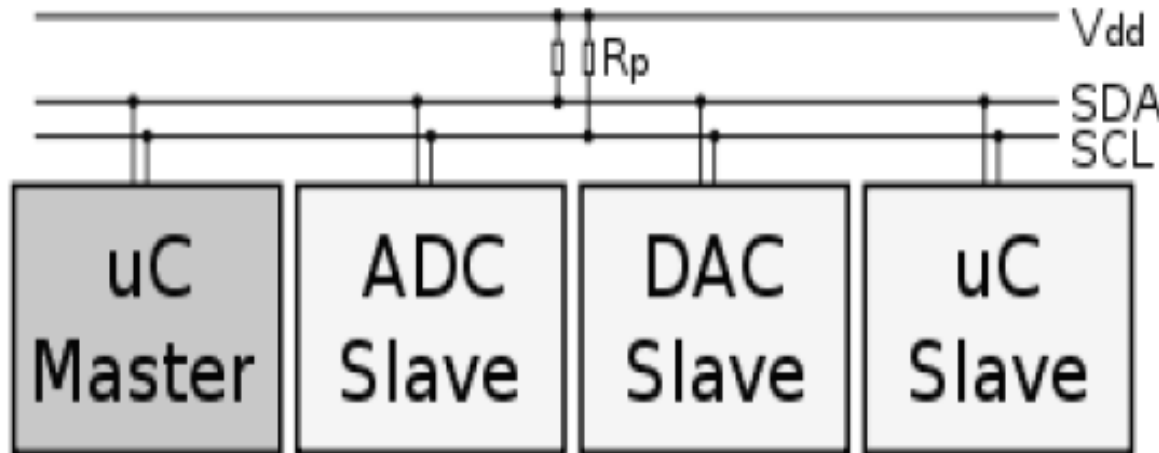


Синхронный интерфейс (SPI, I2C)



Интерфейс I2C

- ✓ **I²C (Inter-Integrated Circuit)** — последовательная шина данных для связи интегральных схем, использующая две двунаправленные линии связи (SDA и SCL).



Пример соединения устройств по шине I2C:

один ведущий – микроконтроллер, три ведомых устройства – АЦП, ЦАП, МК

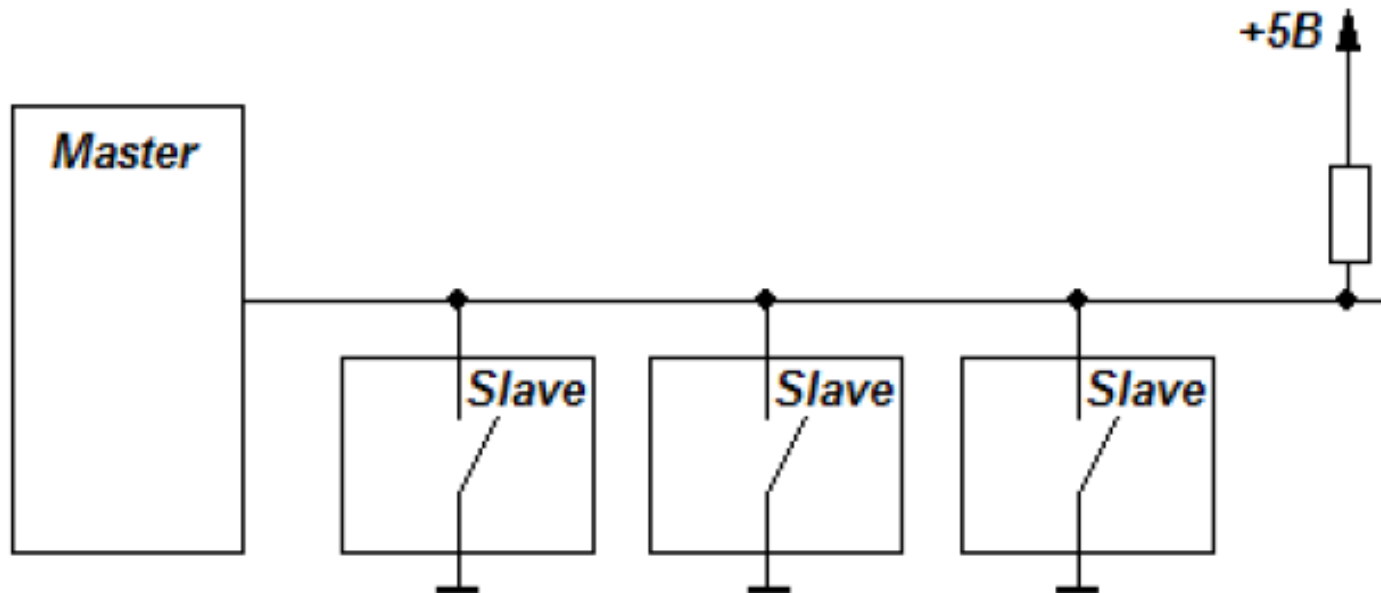


Характеристики интерфейса I2C

- ✓ Полудуплексный канал передачи данных
- ✓ Количество сигнальных линий: 2
- ✓ Роли устройств: ведущий, ведомый
- ✓ Поддерживается режим работы с множеством ведущих
- ✓ Синхронный интерфейс



I2C | подключение ведомых по схеме монтажного И



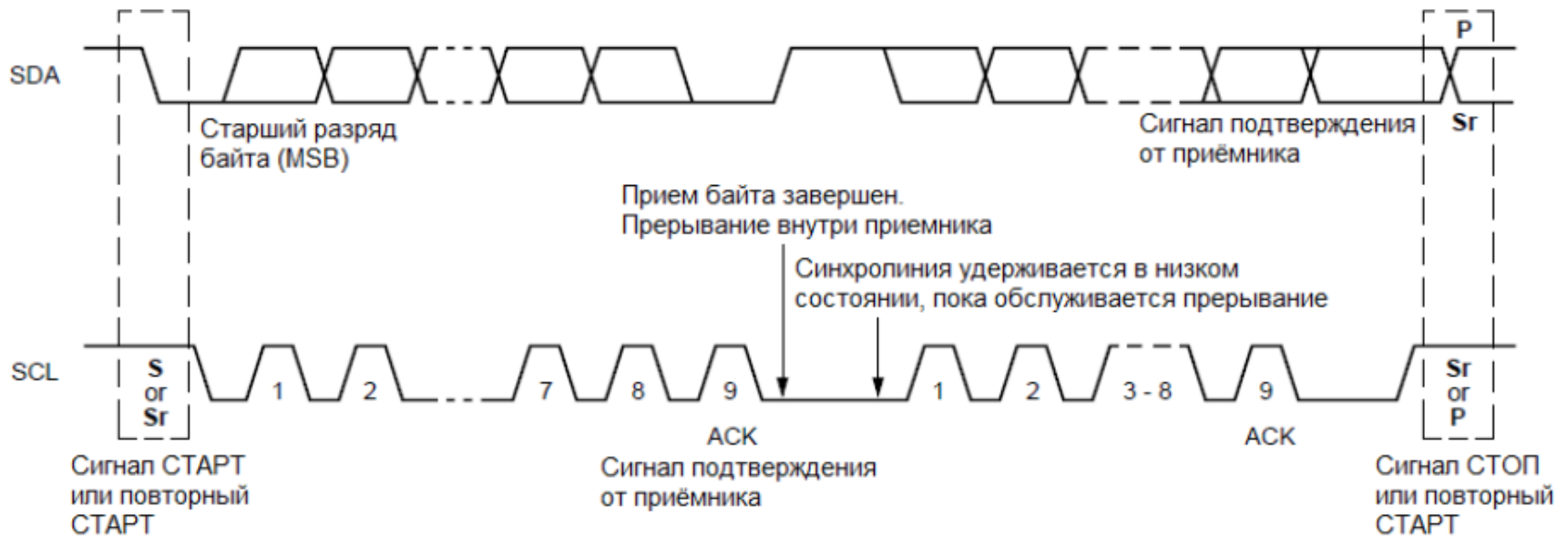
I2C | протокол передачи данных (1)

Пересылка бита по шине I2C.



I2C | протокол передачи данных (2)

Пересылка данных по шине I2C

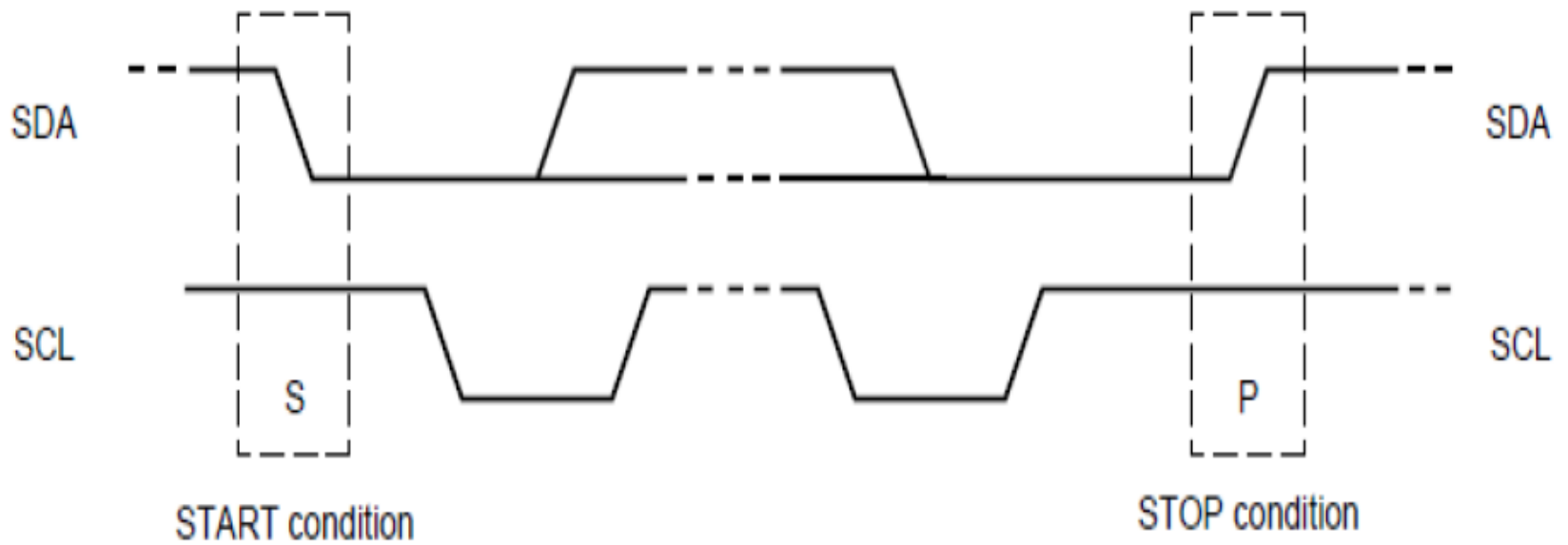




Состояния СТАРТ и СТОП (1)

- ✓ состояние СТАРТ: переход сигнала линии SDA из ВЫСОКОГО состояния в НИЗКОЕ при ВЫСОКОМ уровне на линии SCL.
- ✓ состояние СТОП : переход состояния линии SDA из низкого состояния в ВЫСОКОЕ при ВЫСОКОМ состоянии линии SCL

Состояния СТАРТ и СТОП (2)



Адресация в шине I2C

Первый байт после СТАРТ-состояния - адресный байт

Служебный пакет - запрос к Slave



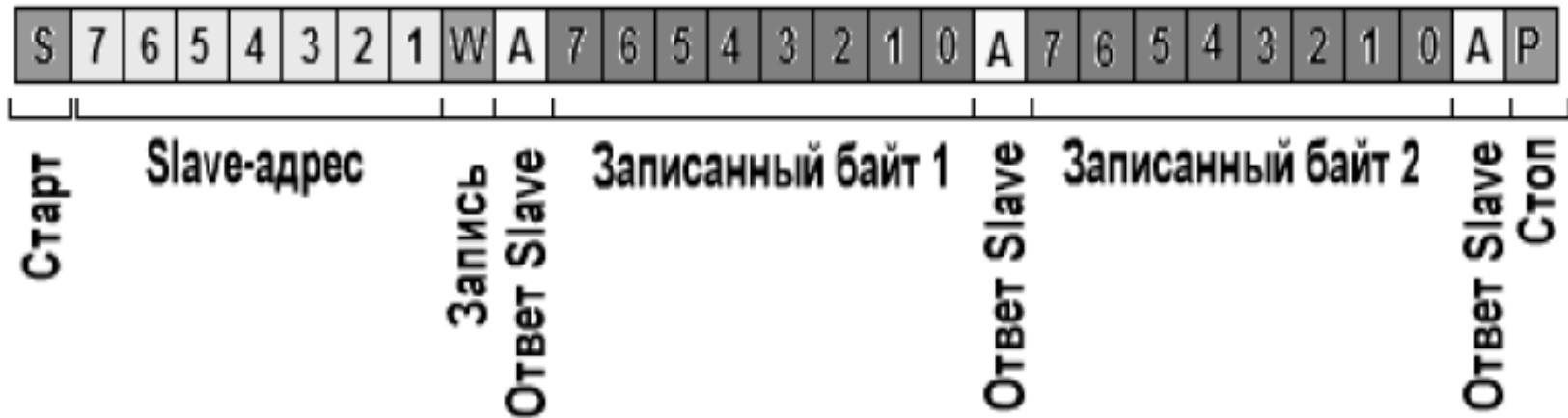
Адрес Slave устройства

Что Slave должен сделать

Ответ ведомого (ACK). Или отсутствие ответа (NACK)



I2C | операция записи, формат данных





I2C | операция чтения, формат данных



I2C | арбитраж

Арбитраж между двумя ведущими (случай одновременной передачи данных)





I²C | сферы использования

I²C находит применение в устройствах, предусматривающих простоту разработки и низкую себестоимость изготовления при относительно неплохой скорости работы.

Список возможных применений:

- ✓ доступ к модулям памяти;
- ✓ доступ к ЦАП/АЦП;
- ✓ чтение информации с датчиков мониторинга и диагностики оборудования, например, термостат центрального процессора или скорость вращения вентилятора охлаждения;
- ✓ чтение информации с часов реального времени;
- ✓ управление включением/выключением питания системных компонент;
- ✓ информационный обмен между микроконтроллерами.



I²C | преимущества и недостатки

Преимущества

- ✓ используется всего два проводника для подключения множества устройств;
- ✓ возможна одновременная работа нескольких ведущих (master) устройств, подключенных к одной шине I²C.

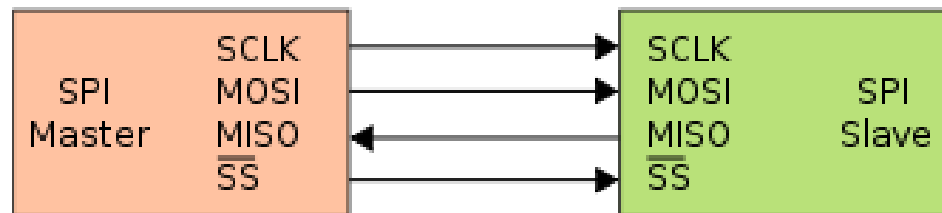
Недостатки

- ✓ ограничение на ёмкость линии — 400 пФ;
- ✓ программирование контроллера I²C затруднено из-за возможных нештатных ситуаций на шине.

Интерфейс SPI

Характеристики интерфейса

- ✓ Направление передачи данных: полный дуплекс
- ✓ Роли устройств: ведущий, ведомый
- ✓ Количество сигнальных линий: 4
- ✓ Синхронный
- ✓ Последовательная передача данных

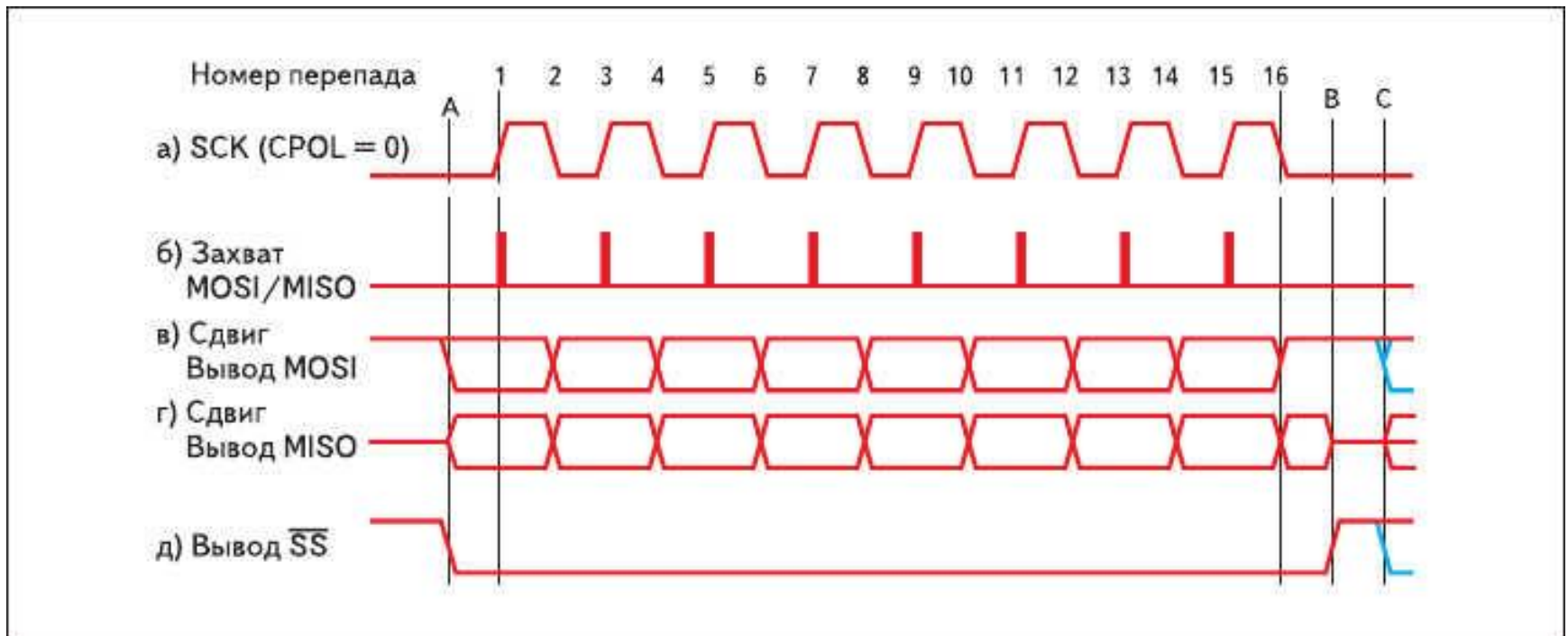




Сигнальные линии интерфейса SPI

- ✓ **MOSI** — Master Output / Slave Input. Служит для передачи данных от ведущего устройства к ведомому.
- ✓ **MISO** — Master Input / Slave Output. Служит для передачи данных от ведомого устройства к ведущему.
- ✓ **SCK** — Serial Clock. Сигнал синхронизации. Служит для передачи тактового сигнала всем ведомым устройствам.
- ✓ **SS** — Slave Select. Выбор ведомого. Служит для выбора ведомого устройства, а также для контроля старта и окончания передачи.

SPI | протокол передачи данных



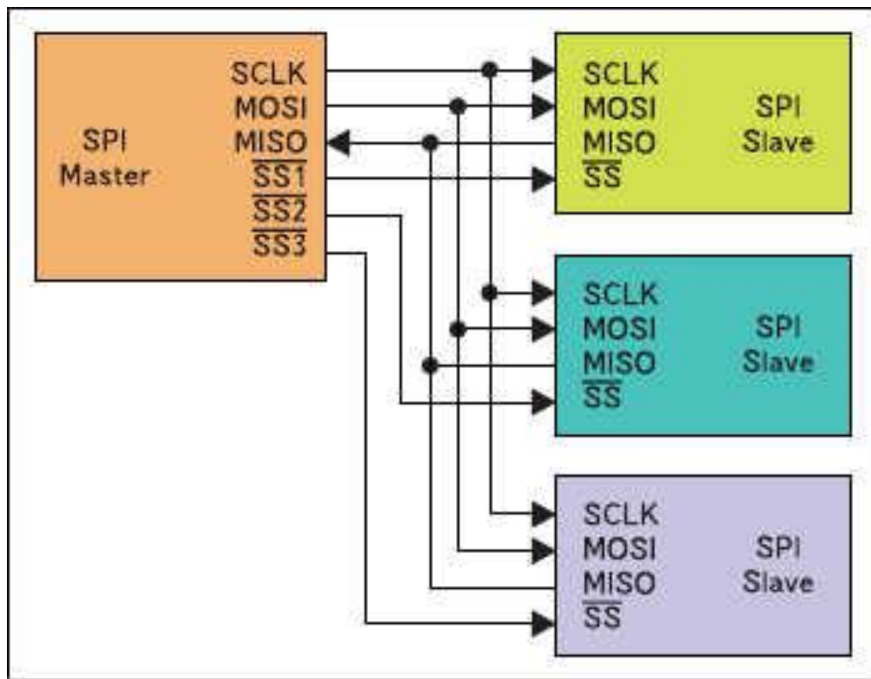


SPI | режимы работы

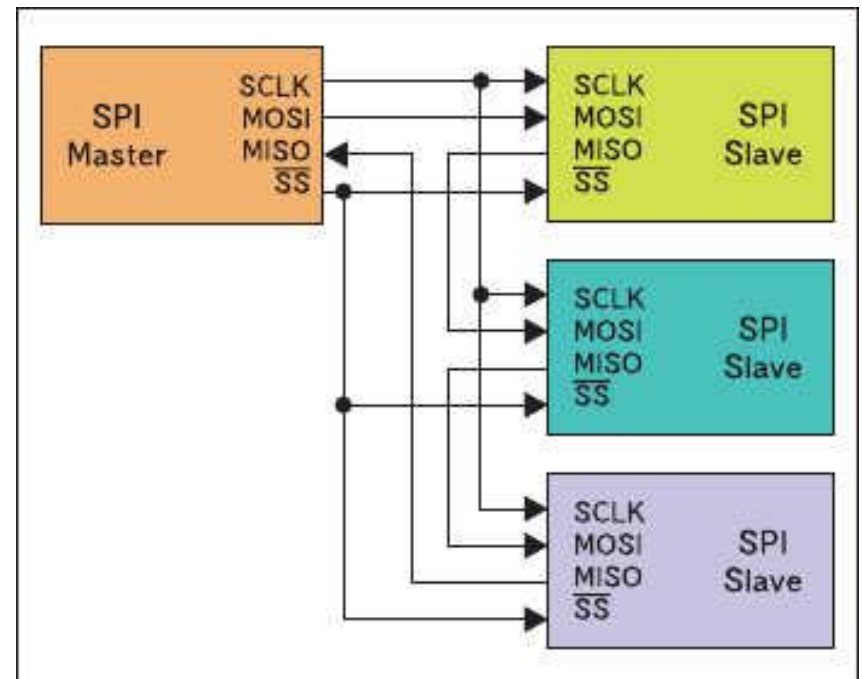
- ✓ CPOL (Clock Polarity) — определяет начальный уровень (полярность) сигнала синхронизации.
- ✓ CPHA (Clock Phase) — фаза синхронизации, определяет по какому из фронтов синхронизирующего сигнала производить выборку данных.

Mode	CPOL	CPHA
SPI Mode 0	0	0
SPI Mode 1	0	1
SPI Mode 2	1	0
SPI Mode 3	1	1

SPI | схемы подключения ведомых

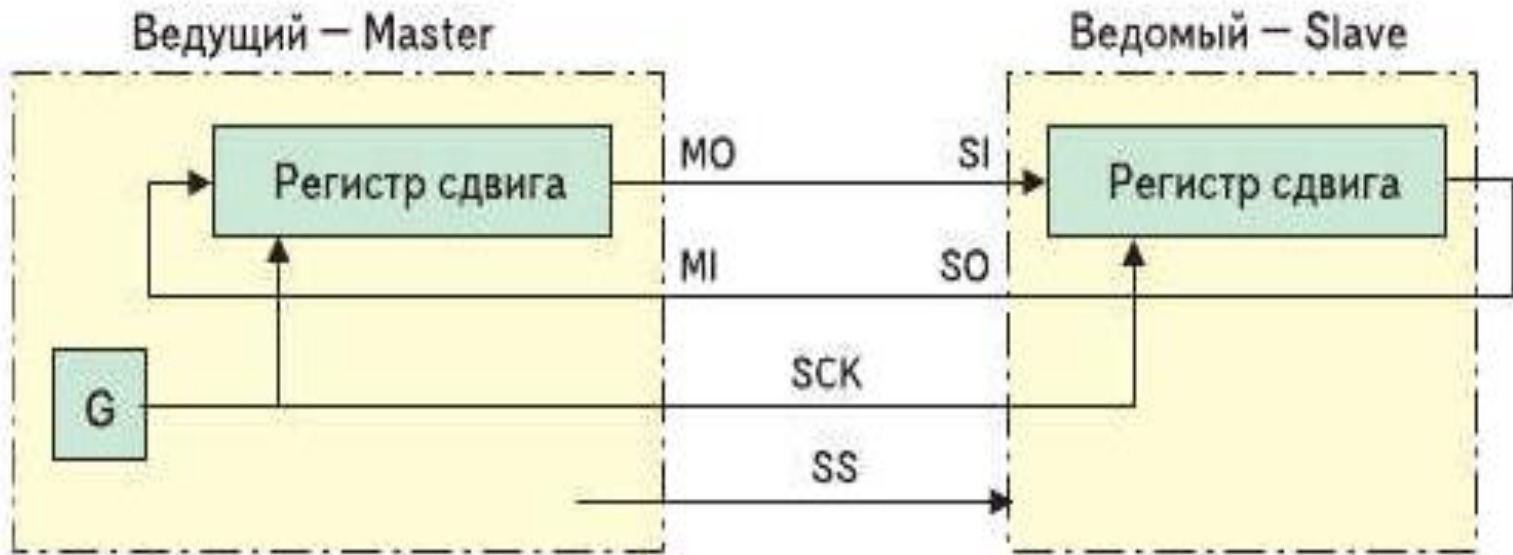


Радиальная структура связи



Кольцевая структура связи

SPI | схема контроллера





SPI | регистры контроллера

С программной точки зрения работы с контроллером SPI представляет собою взаимодействие с регистрами:

- ✓ Управления контроллером (SPCR)
- ✓ Состояния контроллера (spi_done)
- ✓ Управления шиной SS
- ✓ Данных

Все перечисленные регистры должны быть отображены в памяти.

SPI | регистр управления контроллером

Bit	7	6	5	4	3	2	1	0
	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- ✔ **SPIE** – разрешает /запрещает прерывания от модуля SPI. Если бит установлен в 1, прерывания от SPI разрешены.
- ✔ **SPE** – включает/выключает модуль SPI. Если бит установлен в 1, модуль SPI включен.
- ✔ **DORD** – определяет порядок передачи данных. Когда бит установлен в 1, содержимое регистра данных передается младшим битом вперед. Когда бит сброшен, то старшим битом вперед.
- ✔ **MSTR** – определяет режим работы микроконтроллера. Если бит установлен в 1, микроконтроллер работает в режиме Master (ведущий). Если бит сброшен – в режиме Slave (ведомый).
- ✔ **SPR1** и **SPR0** – определяют частоту тактового сигнала SPI модуля, то есть скорость обмена. Максимально возможная скорость обмена всегда указывается в спецификации периферийного устройства.



SPI | алгоритм передачи данных

Алгоритм передачи данных выглядит следующим образом:

- 1) Инициализация сеанса передачи ($SS = 0$)
- 2) Запись байта данных в сдвиговый регистр
- 3) Ожидание бита готовности (`spi_done`)
- 4) Считывание пришедших данных из сдвигового регистра (если необходимо)
- 5) Повторить шаг 2, 3 и 4 для последующих байтов данных
- 6) Окончание сеанса передачи ($SS \leftarrow 1$)



Реализация SPI в различных устройствах

Нижеперечисленные свойства интерфейса SPI зависят от конкретного периферийного устройства.

- ✓ Частота тактового сигнала SCK
- ✓ Режим работы (0-3)
- ✓ Порядок передачи данных
- ✓ Структура пакета данных



SPI | преимущества

- ✓ Полнодуплексная передача данных.
- ✓ Более высокая пропускная способность по сравнению с I2C.
- ✓ Возможность произвольного выбора длины пакета, длина пакета не ограничена восемью битами.
- ✓ Простота аппаратной реализации.
- ✓ Используется только четыре вывода, что гораздо меньше, чем для параллельных интерфейсов.



SPI | недостатки

- ✓ Необходимо больше выводов, чем для интерфейса I2C.
- ✓ Ведомое устройство не может управлять потоком данных.
- ✓ Нет подтверждения приема данных со стороны ведомого устройства (ведущее устройство может передавать данные «в никуда»).
- ✓ Нет определенного стандартом протокола обнаружения ошибок.
- ✓ По дальности передачи данных интерфейс SPI уступает таким стандартам, как UART.



УНИВЕРСИТЕТ ИТМО

Спасибо за внимание!

<http://embedded.ifmo.ru>

sergei_bykovskii@corp.ifmo.ru