

ФЕДЕРАЛЬНОЕ БЮДЖЕТНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»



МИКРОПРОЦЕССОРНЫЕ СИСТЕМЫ
ЛЕКЦИЯ №14

«Последовательные интерфейсы» (продолжение)

Лектор:
доцент каф. ЭАФУ ФТИ
Горюнов А.Г.

Томск 2012 г.

План лекции

- 14.1 Последовательная синхронная и асинхронная передача данных;
- 14.2 Микроконтроллерные интерфейсы;
- 14.3 Организация физического уровня промышленных интерфейсов;
- 14.4 Способы обмена информацией в микропроцессорной системе.

14.1 Последовательная синхронная и асинхронная передача данных

В последовательных интерфейсах применяют различные методы кодирования последовательной информации. Наиболее часто используются следующие коды:

- Код без возвращения к нулю (БВН) (англоязычная аббревиатура – NRZ).
- Код без возвращения к нулю с инверсией (БВНИ).
- Код Манчестер 2.

Код **без возвращения к нулю** отображает последовательность двоичных битов последовательностью уровней напряжения, постоянных на интервале каждого передаваемого двоичного разряда.

В коде **без возвращения к нулю с инверсией** логическая единица передается отсутствием изменения уровня напряжения предшествующего бита, а логический нуль – инверсией этого уровня.

Код **Манчестер 2** отображает каждый бит двоичной последовательности переходом уровней: если в середине битового интервала низкий уровень сменяется высоким, то передается логический нуль, если в середине битового интервала высокий уровень сменяется низким, то передается логическая единица.

Примеры представления последовательности двоичных битов 10111000011 в кодах БВН, БВНИ и Манчестер 2 приведены на рисунке 14.1.

Физический уровень реализации последовательных интерфейсов определяет число линий связи, электрические уровни сигналов на линиях, скорость передачи, максимально допустимую длину линии.

Последова-
тельность бит

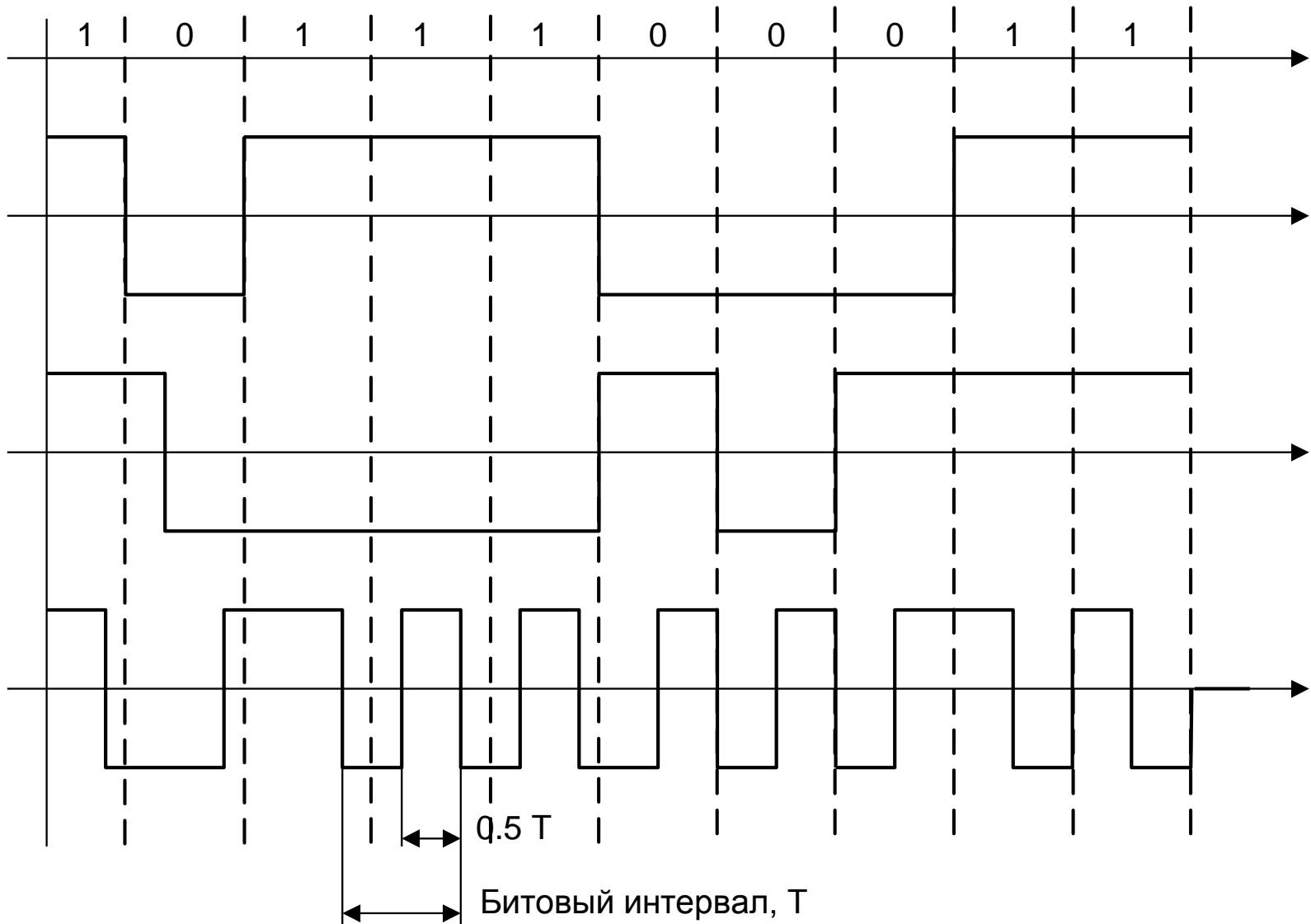
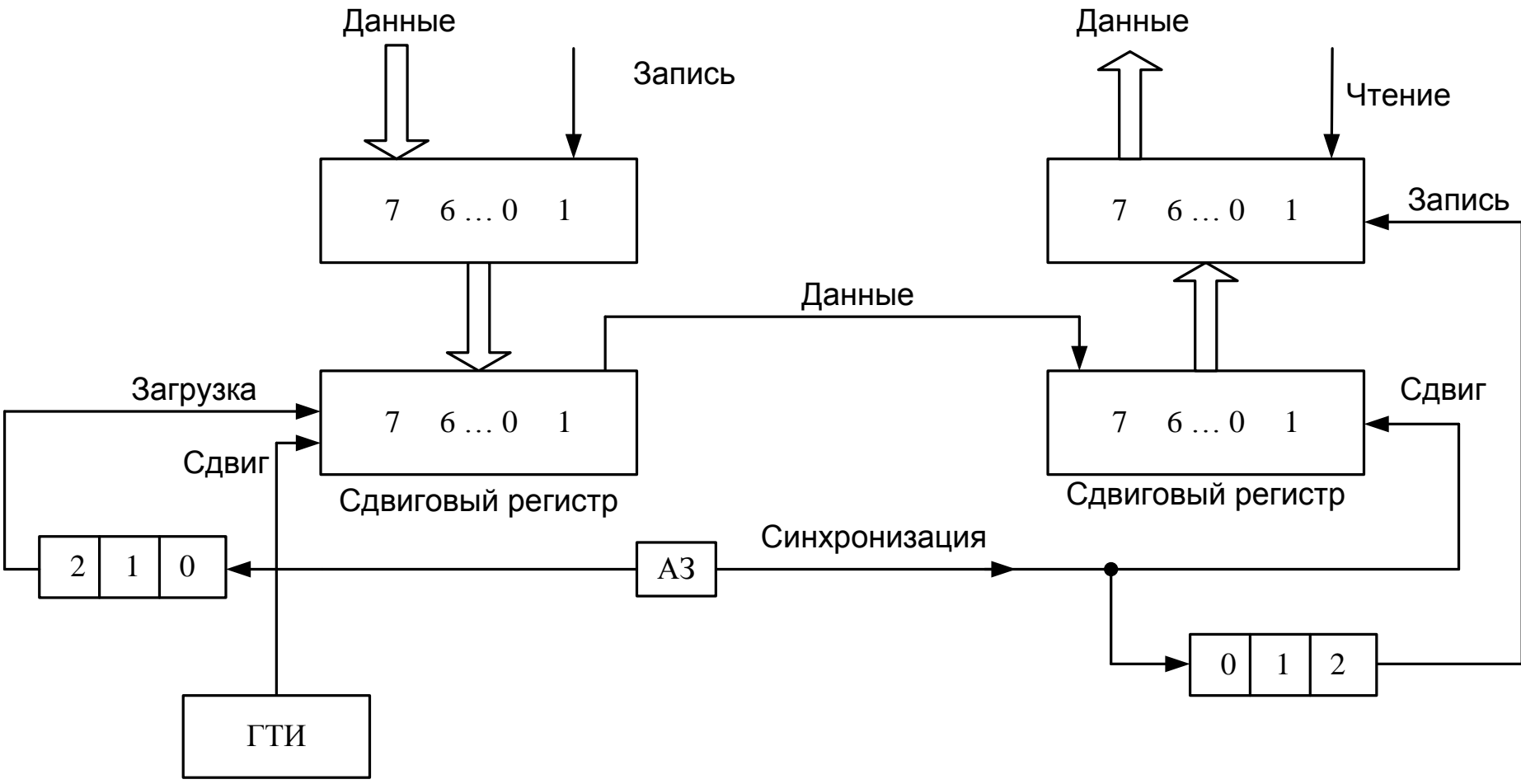
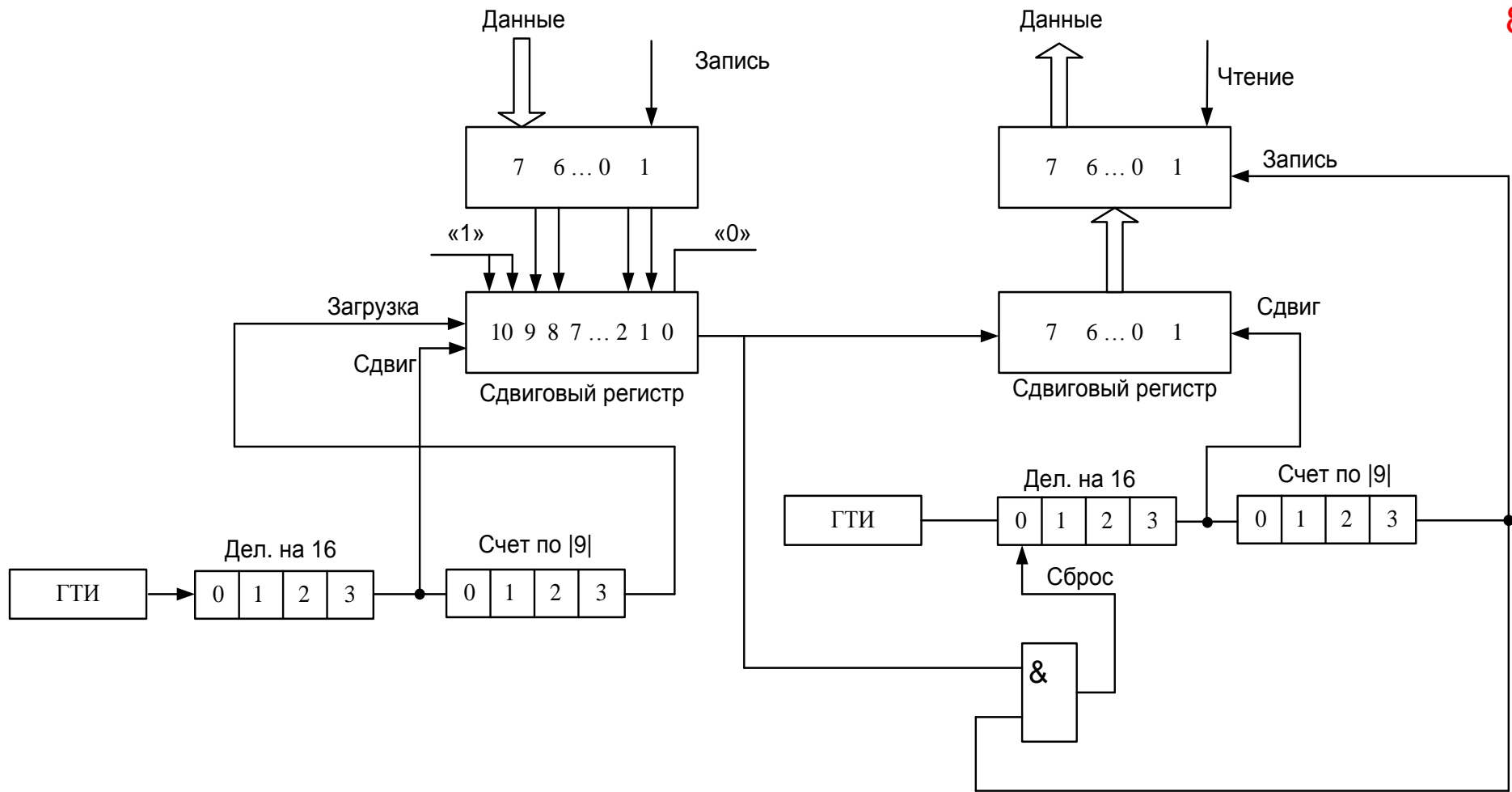


Рисунок 14.1 – Примеры информационных кадров для синхронного режима



Примеры: SPI, I2C

Рисунок 14.2 – Последовательный синхронный контроллер



Примеры: UART, USB, CAN, ETHERNET

Рисунок 14.3 – Последовательный асинхронный контроллер

14.2 Микроконтроллерные интерфейсы

Шина SPI

Последовательный периферийный интерфейс SPI (Serial Peripheral Interface) предназначен для связи МК с периферийными устройствами МП-системы.

Наиболее часто эти устройства расположены на одной плате с МК, реже – это вынесенные пульты управления, индикаторные панели и т.п.

В качестве периферийных устройств могут использоваться как простейшие сдвиговые регистры, так и сложные периферийные ИС со встроенными контроллерами управления, такие, как ЦАП, сигма-дельта АЦП с цифровой фильтрацией, последовательные запоминающие устройства типа FLASH или EEPROM, энергонезависимые ОЗУ и т.д.

Рынок периферийных компонентов с интерфейсом, поддерживающим один из протоколов обмена SPI, чрезвычайно широк.

Знание этого рынка элементной базы – путь к проектированию встраиваемых МП-систем на современном уровне.

На рис 14.4 представлена структурная схема сопряжения МК и двух периферийных ИС с использованием интерфейса SPI.

В рассматриваемом примере МК является ведущим устройством, он инициирует обмен при передаче информации между МК и одной из периферийных ИС.

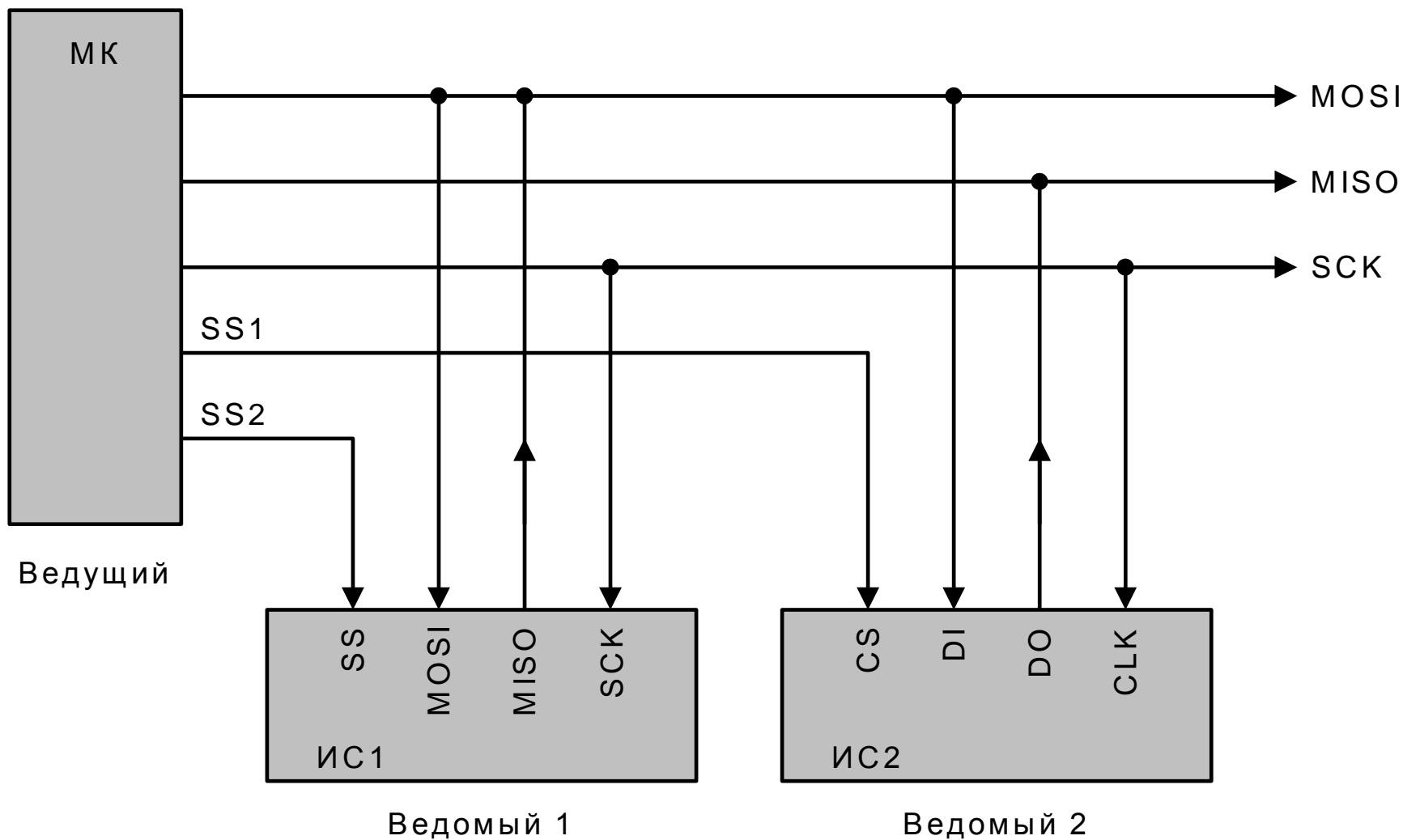


Рисунок 14.4 – Сопряжение МК с периферийными ИС посредством интерфейса SPI

Каждая из периферийных ИС является ведомым устройством. SPI-шина представлена тремя общими линиями связи (MISO, MOSI, SCK) и двумя линиями выбора ведомого устройства (SS1, SS2), которые индивидуальны для каждой периферийной ИС:

- **MOSI** – линий передачи данных от ведущего к ведомому (Master Output Slave Input).
- **MISO** – линия передачи данных от ведомого к ведущему (Master Input Slave Output).
- **SCK** – линия сигнала стробирования данных.
- **SS1** и **SS2** – линии сигналов выбора ведомого устройства.

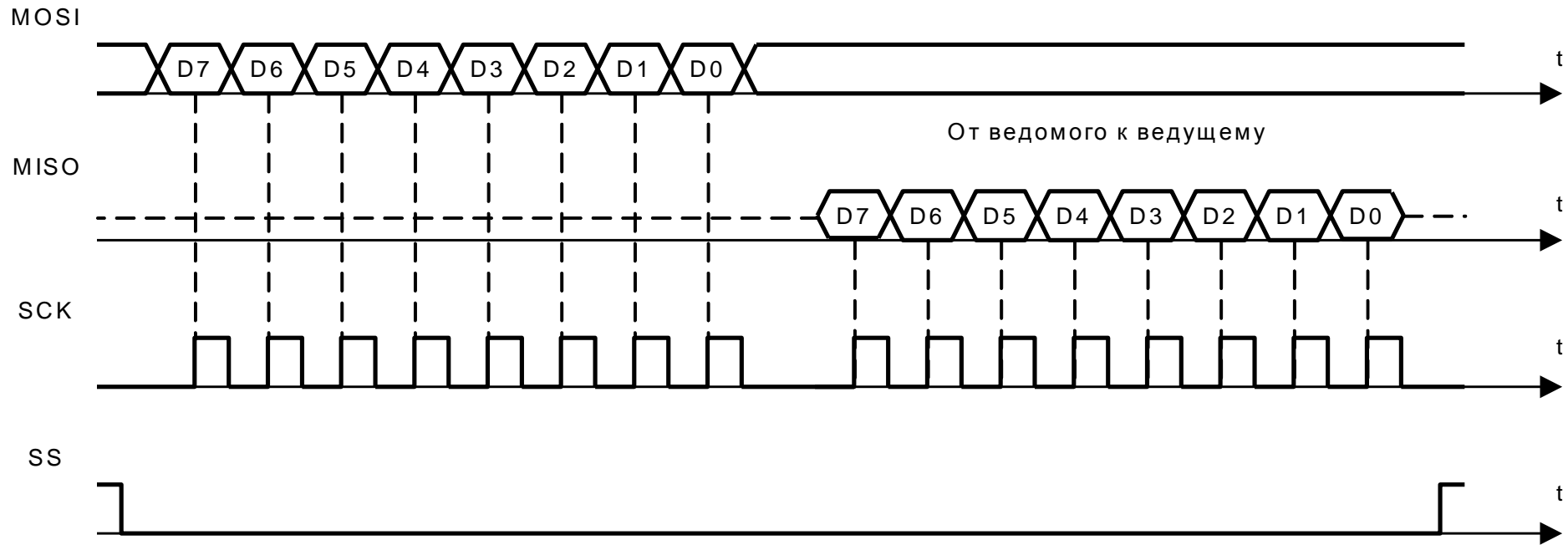


Рисунок 14.5 – Временные диаграммы,
поясняющие принцип действия SPI-интерфейса

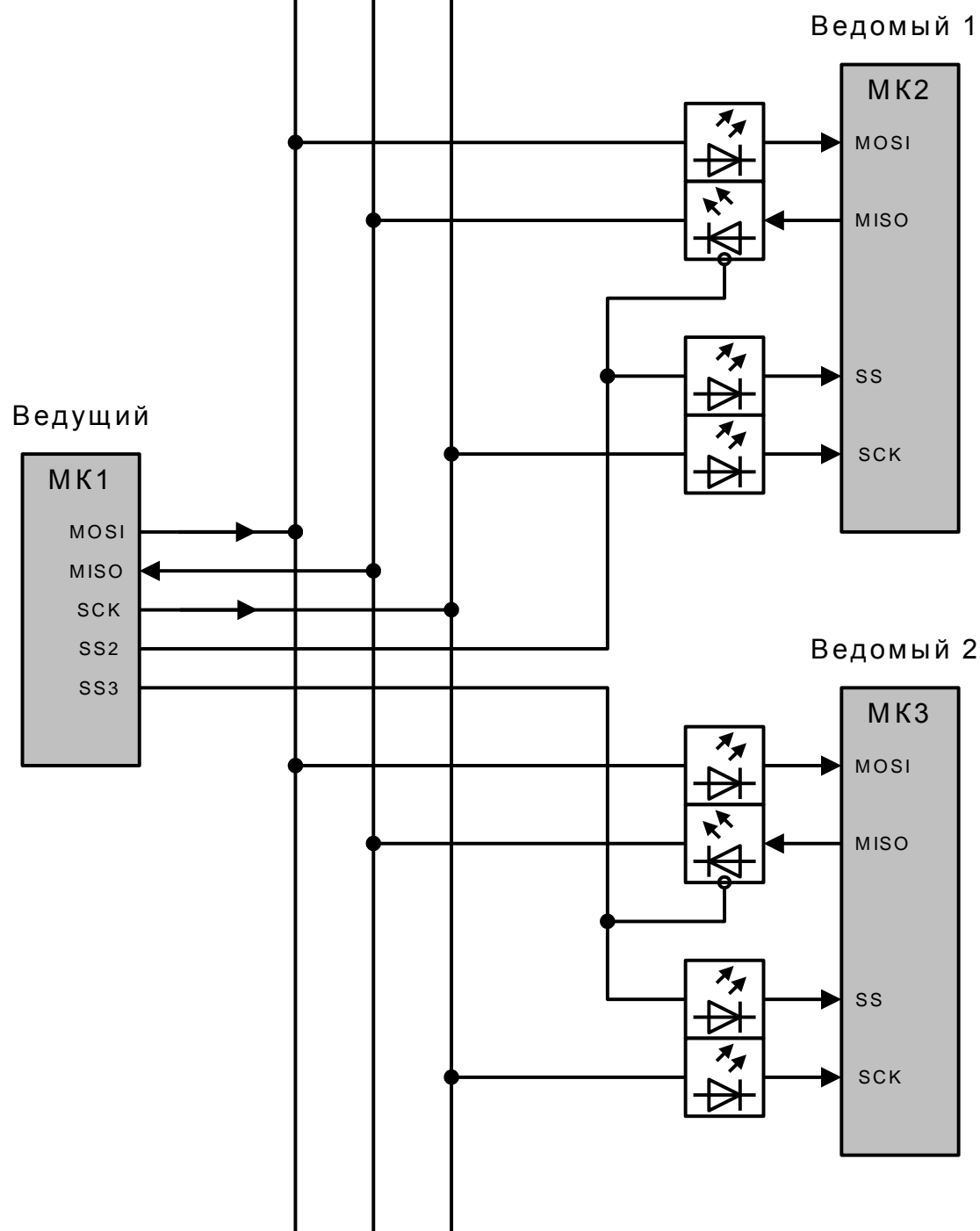


Рисунок 14.6 – Связь МК посредством интерфейса SPI с потенциальной развязкой

Шина I2C

Интерфейс I2C (синхронный последовательный интерфейс I2C) является двухпроводным последовательным интерфейсом, разработанным фирмой Philips.

Изначально в стандартном режиме шина была предназначена для скорости передачи данных до 100 кбит/с.

В усовершенствованном быстром режиме поддерживается передача данных со скоростью до 400 кбит/с.

На одной шине одновременно могут работать устройства и стандартного, и быстрого режима.

Протокол связи I²C. Интерфейс I²C использует протокол с битом подтверждения ASK для обеспечения надежной передачи и приема.

При передаче одно устройство – ведущее (выдает тактовый сигнал), а другое – ведомое.

Все составляющие протокола ведомого, включая поддержку общего вызова, а также протокол ведущего реализованы аппаратно в модуле I²C.

Адресация на шине I²C. В протоколе интерфейса I²C каждое устройство имеет адрес.

Когда ведущий хочет инициировать передачу данных, он сначала передает адрес устройства SLA, к которому хочет обратиться.

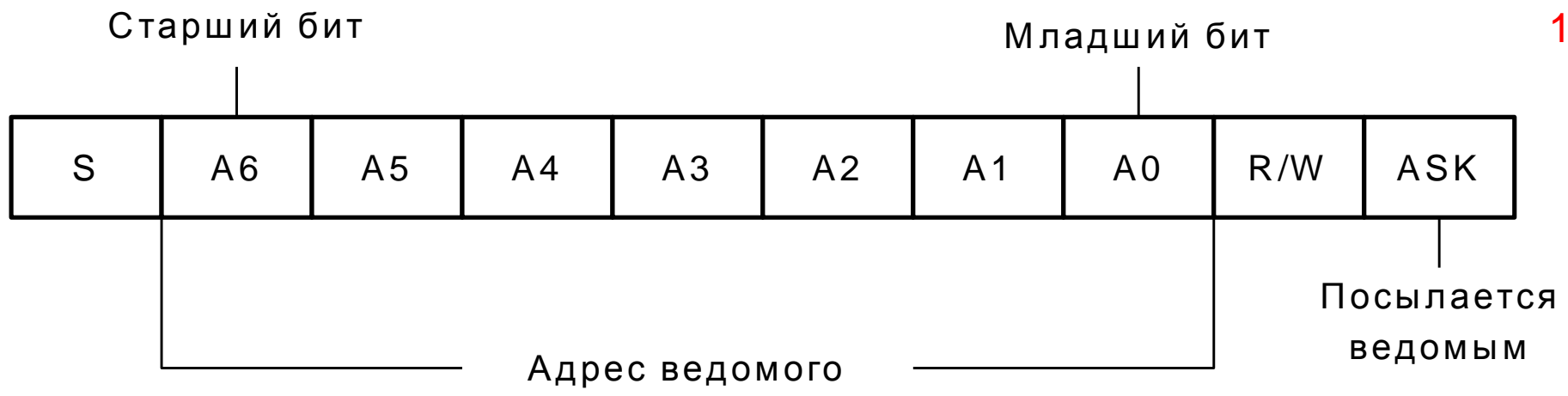
Все устройства на шине следят за выставленным на шину адресом и сравнивают его с собственным.

Вместе с адресом передается бит направления передачи R/W, который определяет, будет ли ведущий читать из ведомого или писать в него.

Существуют два формата адреса. Простейший – 7-битный формат с битом R/W (рис. 14.7).

Адрес – это 7 старших битов байта. Более сложным является 10-битный формат с битом R/W (рис. 14.8).

В 10-битном адресе первые пять битов определяют, что это 10-битный адрес.

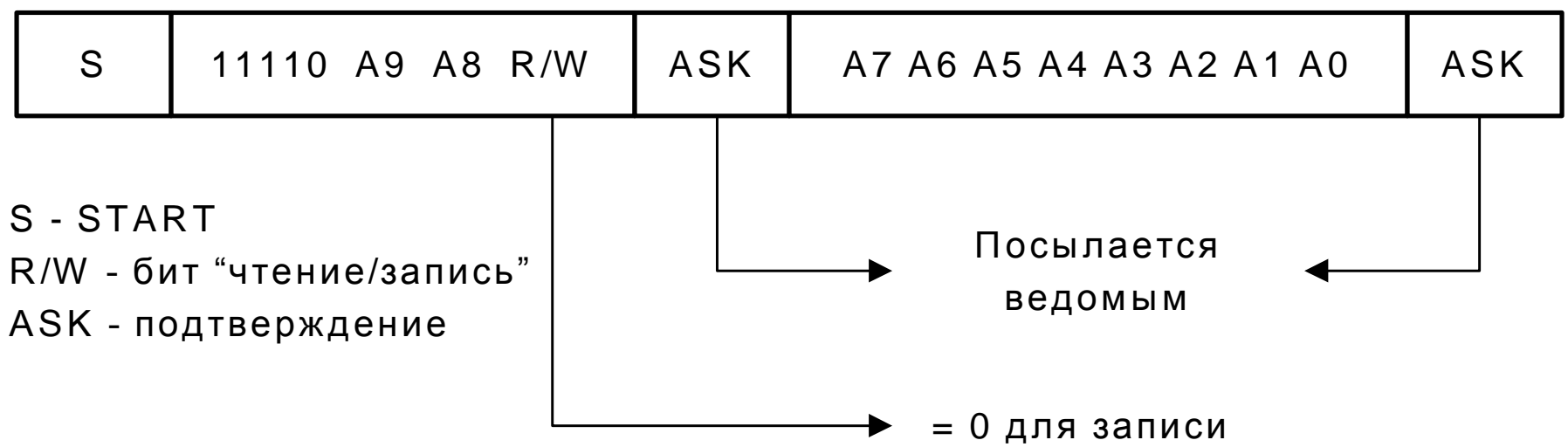


S - START

R/W - бит "чтение/запись"

ASK - подтверждение

Рисунок 14.7 – Формат 7-битного адреса



S - START

R/W - бит "чтение/запись"

ASK - подтверждение

= 0 для записи

Рисунок 14.8 – Формат 10-битного адреса

Основные типы передачи данных.

В зависимости от направления передачи битов (R/W) для I²C-шины возможны два типа передачи данных:

- Передача данных от главного передатчика к подчиненному приемнику.

Первый байт, передаваемый передатчиком, является адресом подчиненного приемника.

Затем следует несколько байтов данных. Подчиненный приемник возвращает бит подтверждения после каждого принятого байта.

- Передача данных от подчиненного передатчика к главному приемнику.

Первый байт (адрес подчиненного передатчика) передается главным устройством.

Затем подчиненный передатчик возвращает бит подтверждения.

Следующие несколько байтов данных передаются подчиненным устройством к главному.

Главное устройство возвращает бит подтверждения после всех принятых байтов, кроме последнего байта.

В конце последнего принятого байта возвращается «нет подтверждения»

Инициализация и прекращение передачи данных.

Когда нет передачи данных (режим ожидания), линии тактирования SCL и данных SDA приведены подтягивающим резистором к высокому уровню.

Главное устройство генерирует все последовательные синхроимпульсы и условия START и STOP, определяющие начало и конец передачи данных.

Условие START определяется как переход SDA из высокого уровня в низкий при высоком уровне SCL, а условие STOP – как переход SDA из низкого уровня в высокий при высоком уровне SCL.

Ввиду такого способа определения условий START и STOP при передаче данных, линия SDA может изменять свое состояние только при низком уровне SCL.

Когда ведущий не желает освободить шину (выставив STOP), он должен выставить повторное условие START, которое идентично START (переход SDA из высокого уровня в низкий при высоком уровне на SCL), но выдается вслед за подтверждением, являясь началом следующей последовательной передачи, таким образом шина не освобождается.

Режимы работы и арбитраж самостоятельно

14.3 Организация физического уровня промышленных интерфейсов

На рисунке 14.9. приведены структурные схемы организации физического уровня последовательных интерфейсов, наиболее часто встречающихся в системах с МК.

Использованы следующие обозначения:

- T (передатчик – **Transmitter**);
- R (приемник – **Receiver**);
- T/R (двунаправленный приемопередатчик) – специальные ИС приемопередатчиков и трансиверов.

Обратите внимание, что физический уровень внутриплатных интерфейсов SPI и I2C реализуется логическими сигналами, поэтому специальные схемы формирователей уровней не требуются.

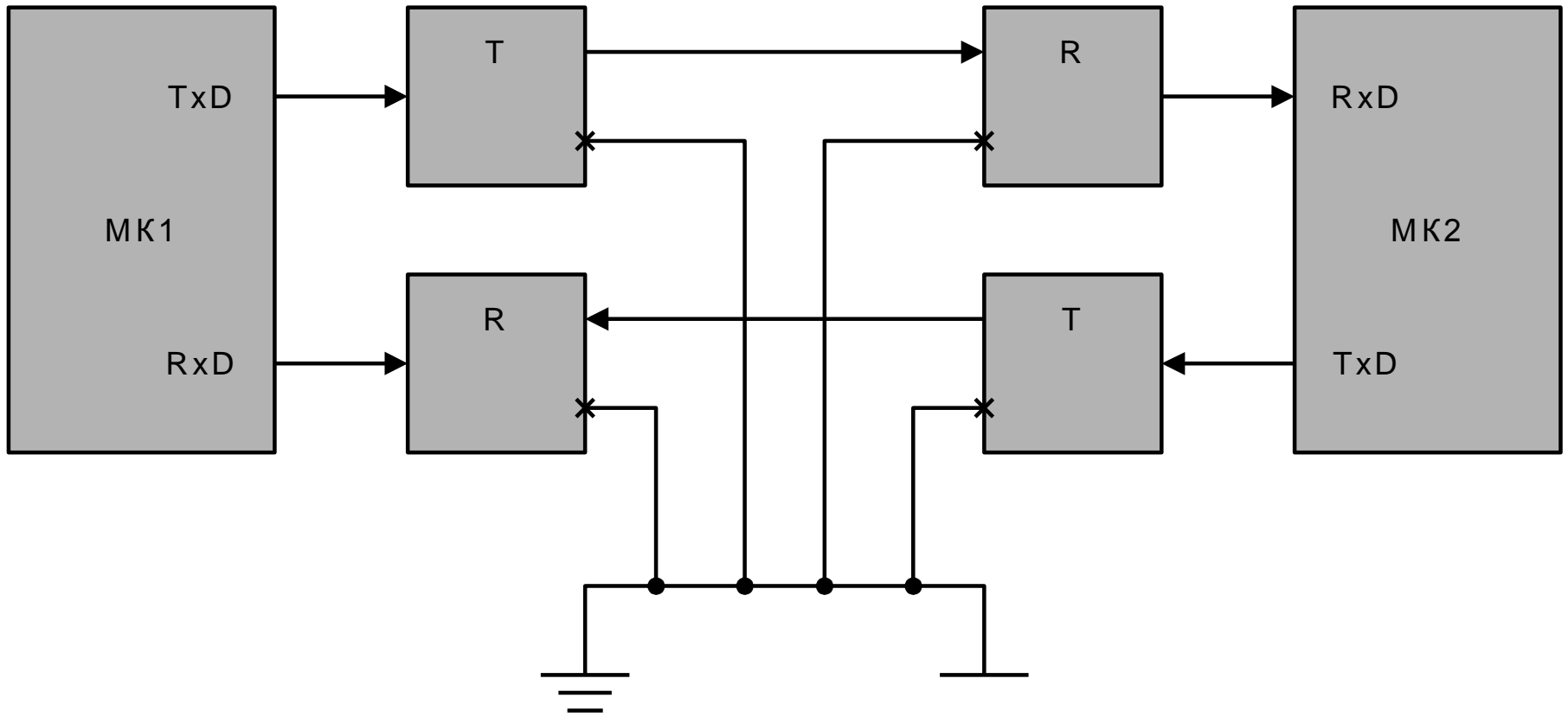


Рисунок 14.9 – Организация физического уровня интерфейса RS-232C

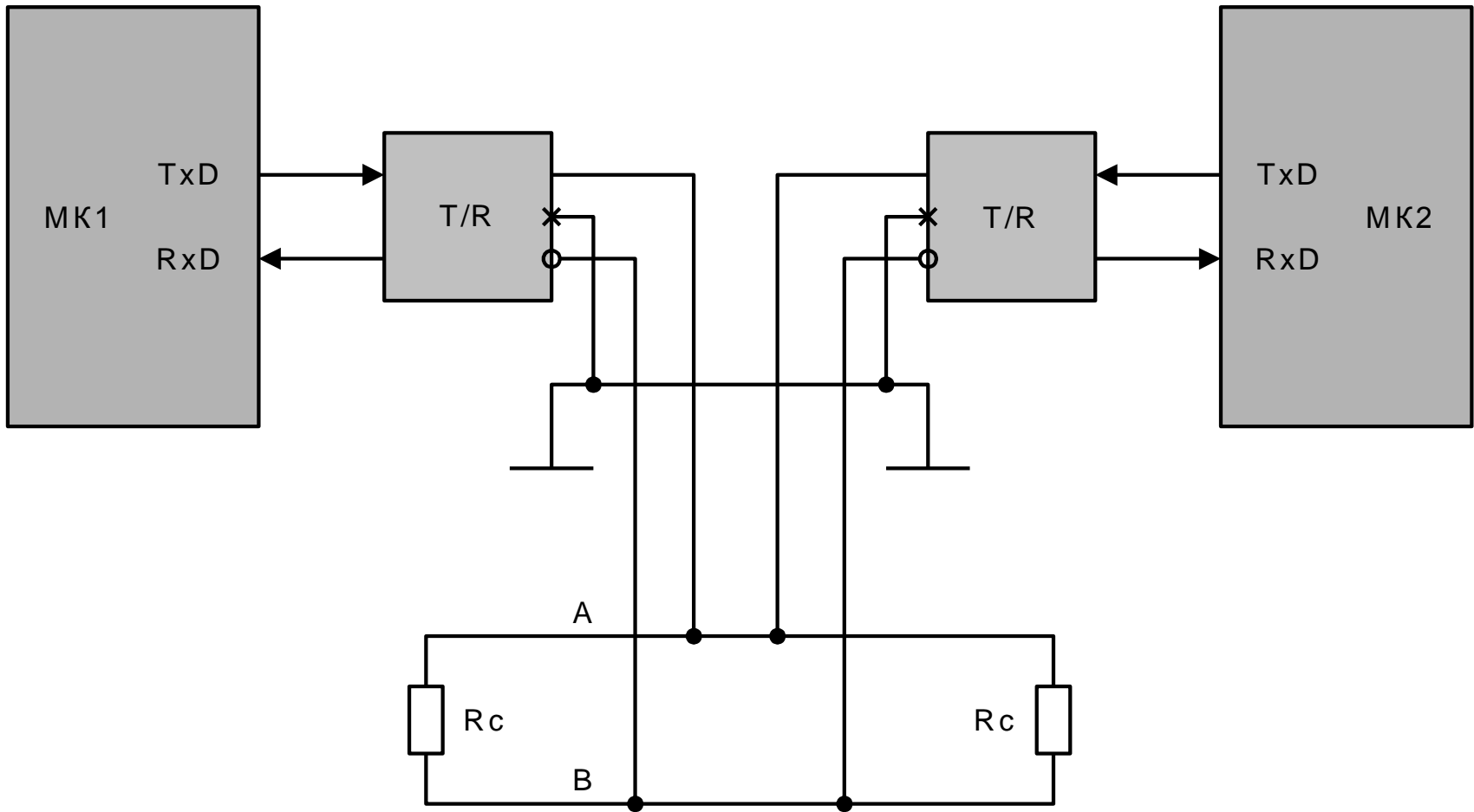


Рисунок 14.10 – Организация физического уровня интерфейса RS-485

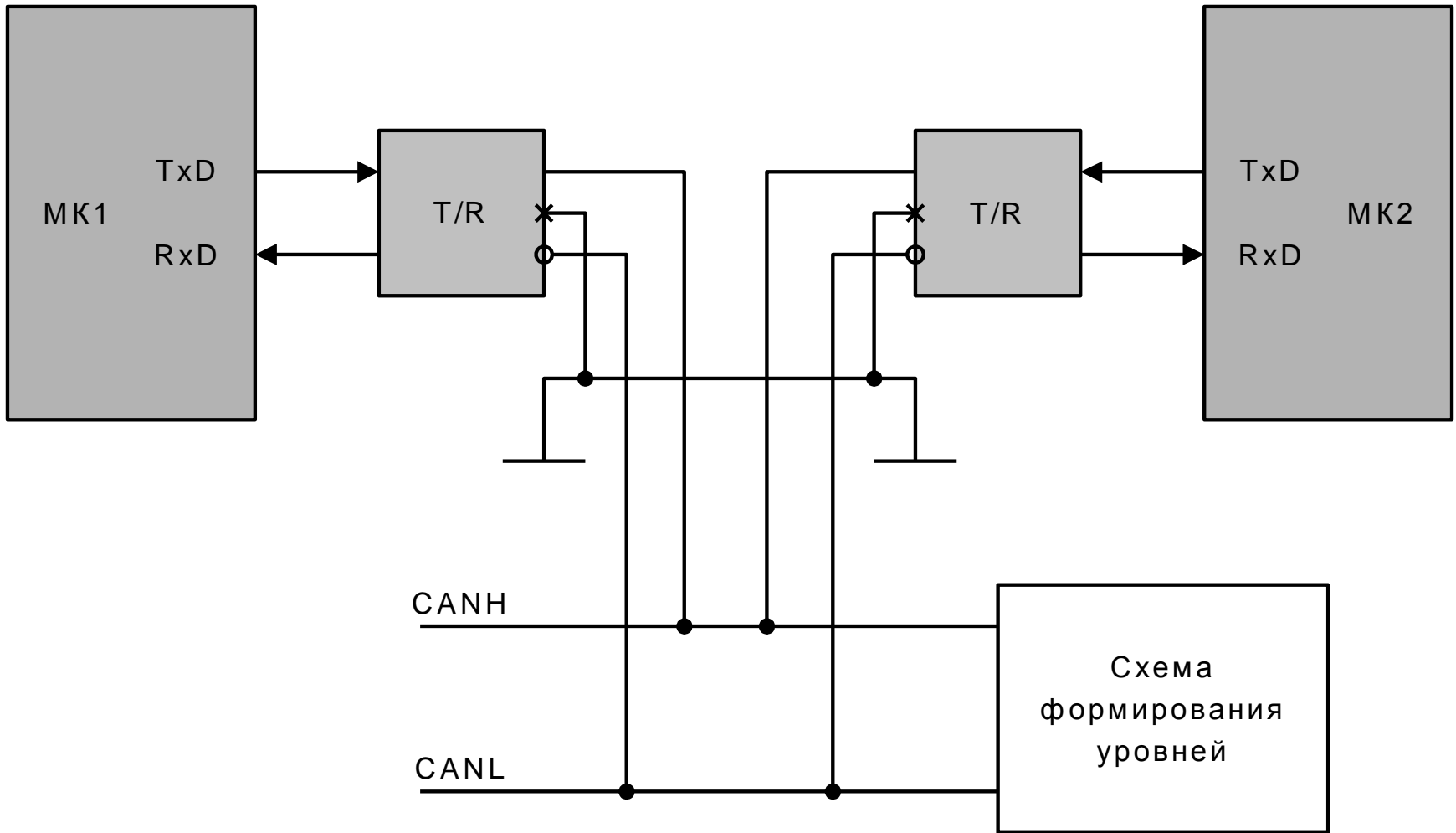


Рисунок 14.11 – Организация физического уровня CAN-сети

Таблица 14.1 – Основные параметры последовательных интерфейсов

Параметры	SPI	I2C	RS-485	RS-422	CAN
Максимальная скорость передачи	10 Мбод	400 кбод	115 кбод	10 Мбод	1 Мбод
Макс. протяженность линий связи [м]	10	500	15	1200 при 100 кбод 12 при 10 Мбод	5000
Число линий связи	3 + n*	2	2	2	2
Наименование линий связи	MISO, MOSI, SCK	SDA, SCL	TxD, RxD	TxD, RxD	CANL, CANH
Число двунаправленных линий связи	Нет	2	Нет	2	2
«1». передача данных	Соответствует уровню «1» логического сигнала	Соответствует уровню «1» логического сигнала	$-2.5V < U < -3.0V$	$A < B$ $-7V < U_{AB} < -0.3V$	$> 2 U_{сети} / 3$

Таблица 14.1 (продолжение)

«0» передача данных	Соответствует уровню «0» логического сигнала	Соответствует уровню «0» логического сигнала	$+3.0V < U < +25 V$	$A > B$ $+0.3V < U_{AB} < +7V$	$< 2 U_{сети} / 3$
Число информационных бит в кадре сообщения	8		8	8	До 64
Число служебных бит в кадре сообщения	Нет	3...4	3...4	3...4	До 64
Наименование служебных бит	Нет	Условия СТАРТ, СТОП, флаг АСК, бит направления R/W	Старт-бит, бит паритета, 1 или 2 стоп-бита	Старт-бит, бит паритета, 1 или 2 стоп-бита	SOF, 11-18 бит ID, RTR, IDE, r0, r1, 4бит DLC, 16бит CRC, ASK, EOF
Требования к передатчику	Нет	Нет	$I_{вых. макс.} = 500mA$ $t_{фр} < 0.04 t_{БИТА}$	$I_{вых. макс.} = 150mA$ $t_{фр} < 0.3 t_{БИТА}$	
Требования к приемнику	Нет	Нет	$R_{вх.} = 3...7k\Omega$, $C_{вх.} < 2500пФ$	$R_{вх.} > 12 k\Omega$, $U_{вх.диф.} > 200 мВ$ $ U_{вх.син.} < 12В$	
Элементная база приемопередатчика	Специальные ИС не требуются	Специальные ИС не требуются	Maxim MAX 1488/ 9E MAX 200/235 MAX562 MAX202	Maxim MAX253 MAX3480A MAX481/489 MAX1480 MAX485	Motorola MC33388 MC33389 Siemens TLE6252 TLE6260

14.4 Способы обмена информацией в микропроцессорной системе

Существует несколько способов обмена информацией в микропроцессорной системе:

- **программно-управляемый ввод-вывод;**
- **ввод-вывод по прерыванию;**
- **прямой доступ в память.**

В ЭВМ применяются три режима ввода/вывода: **программно-управляемый** ВВ (называемый также программным или нефорсированным ВВ), ВВ **по прерываниям** (форсированный ВВ) и **прямой доступ к памяти** (ПДП (DMA)).

Первый из них характеризуется тем, что инициирование и управление ВВ осуществляется программой, выполняемой процессором, а внешние устройства играют сравнительно пассивную роль и сигнализируют только о своем состоянии, в частности, о готовности к операциям ввода/вывода.

Во втором режиме ВВ инициируется не процессором, а внешним устройством, генерирующим специальный сигнал **прерывания**.

Реагируя на этот сигнал готовности устройства к передаче данных, процессор передает управление подпрограмме обслуживания устройства, вызвавшего прерывание.

Действия, выполняемые этой подпрограммой, определяются пользователем, а непосредственными операциями ВВ управляет процессор.

Наконец, в режиме **прямого доступа к памяти**, который используется, когда пропускной способности процессора недостаточно, действия процессора приостанавливаются, он отключается от системной шины и не участвует в передачах данных между основной памятью и быстродействующим ВУ.

Во всех вышеуказанных случаях основные действия, выполняемые на системной магистрали ЭВМ, подчиняются двум основным принципам:

1. В процессе взаимодействия любых двух устройств ЭВМ одно из них обязательно выполняет активную, управляющую роль и является задатчиком, второе оказывается управляемым, исполнителем. Чаще всего задатчиком является процессор.
2. Другим важным принципом, заложенным в структуру интерфейса, является принцип квитирования (запроса-ответа): каждый управляющий сигнал, посланный задатчиком, подтверждается сигналом исполнителя.

При отсутствии ответного сигнала исполнителя в течение заданного интервала времени формируется так называемый тайм-аут, задатчик фиксирует ошибку обмена и прекращает данную операцию.

Подсистема прерываний и прямой доступ в память
самостоятельно по учебному пособию