

**Описание интерфейса Г<sup>2</sup>С.**

## СОДЕРЖАНИЕ

<u>1.0. Концепция шины I<sup>2</sup>C.</u>
<u>2.0. Общие параметры.</u>
<u>2.1. Валидность данных.</u>
<u>2.2. Сигналы START и STOP.</u>
<u>3.0. Пересылка бита.</u>
<u>3.1. Формат байта.</u>
<u>3.2. Подтверждение.</u>
<u>4.0. Синхронизация.</u>
<u>4.1. Арбитраж.</u>
<u>4.2. Использование механизма синхронизации как процедуры управления связью.</u>
<u>5.0. Форматы с 7-битным адресом.</u>
<u>6.0. 7-битная адресация.</u>
<u>6.1. Назначение битов первого байта.</u>
<u>6.1.1. Адрес общего вызова.</u>
<u>6.1.2. Байт СТАРТА.</u>
<u>6.1.3. Совместимость с CBUS.</u>
<u>7.0. Электрические параметры устройств I<sup>2</sup>C.</u>
<u>8.0. Максимальные и минимальные значения для резисторов R<sub>p</sub> и R<sub>s</sub>.</u>
<u>9.0. Дополнения к спецификации шины I<sup>2</sup>C.</u>
<u>10.0. Быстрый режим.</u>
<u>11.0. 10-битная адресация.</u>
<u>11.1. Назначение битов первых двух байтов.</u>
<u>11.2. 10-битные форматы.</u>
<u>12.0. Адрес общего вызова и байт СТАРТА.</u>
<u>13.0. Электрические и временные параметры.</u>
<u>14.0. Практические рекомендации.</u>
<u>14.1. Особенность реализации интерфейса с шиной I<sup>2</sup>C в ЭКЛЗ.</u>
<u>14.2. Рекомендации по созданию "Master"- устройства.</u>

### 1.0. Концепция шины I<sup>2</sup>C.

Шина I<sup>2</sup>C поддерживает любую технологию изготовления микросхем (НМОП, КМОП, биполярную). Две линии, данных (SDA) и синхронизации (SCL), служат для переноса информации. Каждое устройство распознается по уникальному адресу и может работать как передатчик или приёмник, в зависимости от назначения устройства. Кроме того, устройства могут быть классифицированы как ведущие и ведомые при передаче данных (см. [Таблицу 1](#) ). Ведущий - это устройство, которое инициирует передачу данных и вырабатывает сигналы синхронизации. При этом любое адресуемое устройство считается ведомым по отношению к ведущему.

**Таблица 1. Терминология шины I<sup>2</sup>C.**

Термин (англ)	Термин (рус)	Описание
Transmitter	Передатчик	Устройство, посылающее данные в шину
Receiver	Приемник	Устройство, принимающее с шины

Master	Ведущий	Начинает пересылку данных, вырабатывает синхроимпульсы, заканчивает пересылку данных
Slave	Ведомый	Устройство, адресуемое ведущим
Multi-master	-	Несколько ведущих могут пытаться захватить шину одновременно, без нарушения передаваемой информации
Arbitration	Арбитраж	Процедура, обеспечивающая Multi-master
Synchronization	Синхр.	Процедура синхронизации двух устройств

**Рисунок 1. Пример конфигурации шины I<sup>2</sup>C с двумя микроконтроллерами.**

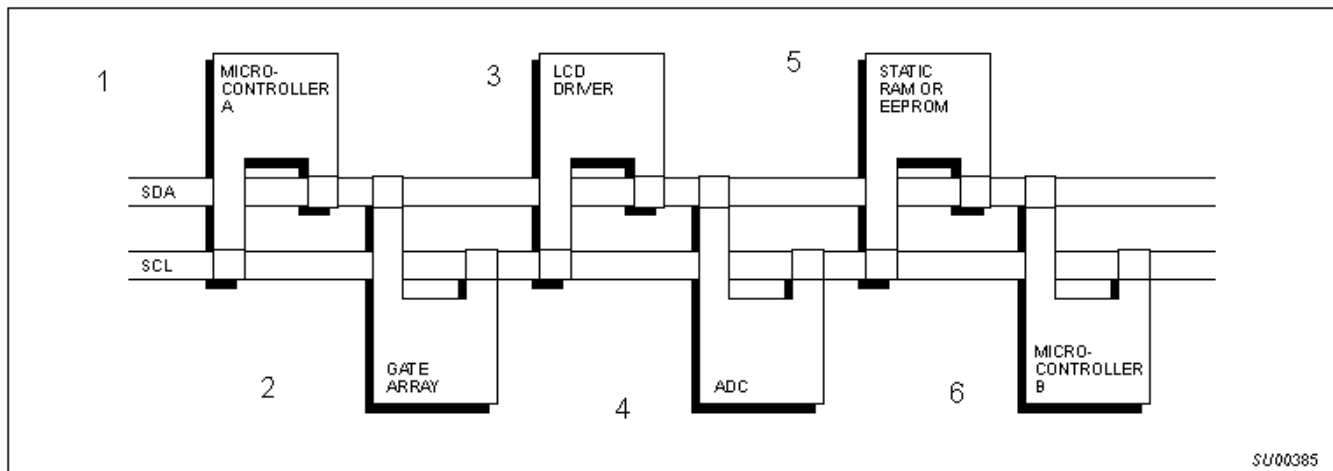


Рисунок 1. Пример конфигурации шины I<sup>2</sup>C с двумя микроконтроллерами:

- Микроконтроллер А
- Набор регистров
- ЖКИ драйвер
- АЦП
- Статическая ОЗУ или ППЗУ
- Микроконтроллер В

Шина I<sup>2</sup>C допускает несколько ведущих. Это означает, что более чем одно устройство, способное управлять шиной, может быть подключено к ней. Поскольку в качестве ведущих обычно выступают микроконтроллеры, давайте рассмотрим пример пересылки данных между двумя микроконтроллерами, подключенными к шине (см. [Рисунок 3](#)). Пример покажет взаимоотношения передатчик-приемник и ведущий-ведомый, существующие в шине I<sup>2</sup>C. Необходимо заметить, что эти отношения не постоянны, а зависят только от направления пересылки данных в данный момент времени. Пересылка данных будет происходить следующим образом:

1. Пусть микроконтроллер А желает послать информацию в микроконтроллер В:

- микроконтроллер А (ведущий) адресует микроконтроллер В (ведомый)
- микроконтроллер А (ведущий-передатчик) посылает данные микроконтроллеру В (ведомый-приёмник)
- микроконтроллер А заканчивает пересылку

2. Пусть микроконтроллер А желает принять информацию от микроконтроллера В:

- микроконтроллер А (ведущий) адресует микроконтроллер В (ведомый)

- микроконтроллер А (ведущий-приемник) принимает данные от микроконтроллера В (ведомый-передатчик)
- микроконтроллер А заканчивает пересылку

В обоих случаях ведущий (микроконтроллер А) генерирует синхроимпульсы и заканчивает пересылку.

Возможность подключения более одного микроконтроллера к шине означает, что более чем один ведущий может попытаться начать пересылку в один и тот же момент времени. Для устранения хаоса, который может возникнуть в данном случае, разработана процедура арбитража. Эта процедура основана на том, что все I<sup>2</sup>C-устройства подключаются к шине по правилу монтажного И. Подробнее об арбитраже см. [Раздел 4.2](#). Генерация синхросигнала - это всегда обязанность ведущего; каждый ведущий генерирует свой собственный сигнал синхронизации при пересылке данных по шине. Сигнал синхронизации может быть изменен только если он "вытягивается" медленным ведомым устройством (путем удержания линии в низком состоянии), или другим ведущим, если происходит столкновение.

## 2.0. Общие параметры.

Как SDA, так и SCL являются двунаправленными линиями, подсоединенными к положительному источнику питания через подтягивающий резистор (см. [Рисунок 2](#)). Когда шина свободна, обе линии находятся в ВЫСОКОМ положении. Выходные каскады устройств, подключенных к шине, должны иметь открытый сток или открытый коллектор для обеспечения функции монтажного И. Данные по шине I<sup>2</sup>C могут передаваться со скоростью до 100 кбит/с в стандартном режиме, и до 400 кбит/с в "быстром" режиме. Количество устройств, подключенных к шине, определяется единственным параметром - емкостью линии (до 400 пф).

**Рисунок 2. Подключение I<sup>2</sup>C-устройств к шине.**

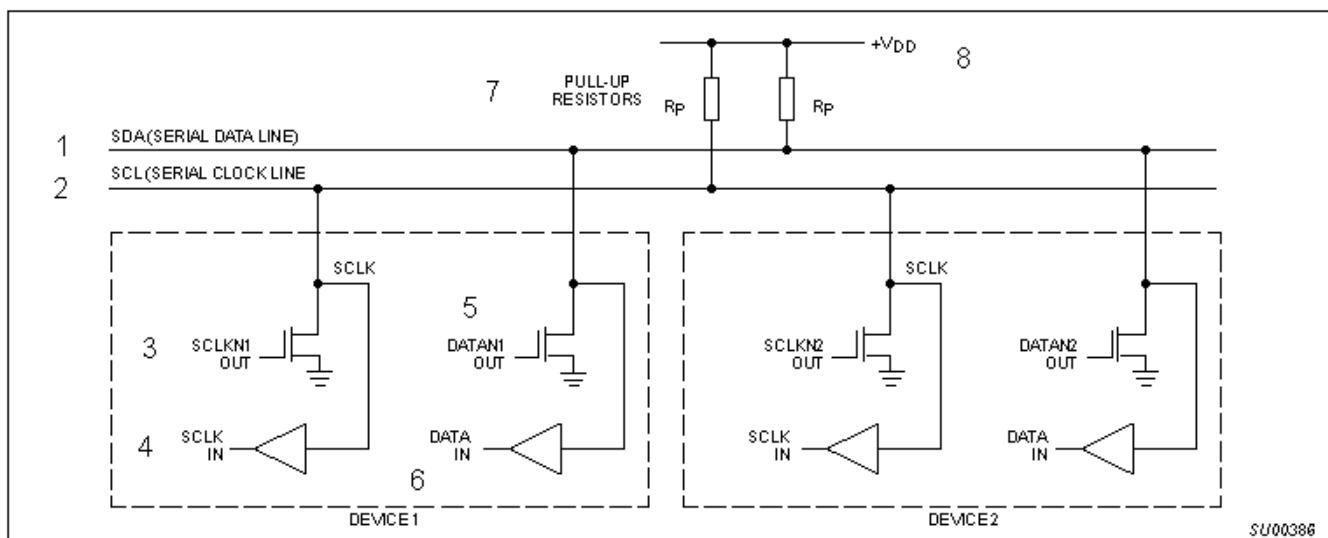


Рисунок 2. Подключение I<sup>2</sup>C-устройств к шине:

- SDA (линия данных)
- SCL (линия синхронизации)
- Выход синхронизации
- Вход синхронизации
- Выход данных

- Вход данных
- Подтягивающие резисторы
- Напряжение питания

## 2.1. Валидность данных.

Данные на линии SDA должны быть стабильными в течение ВЫСОКОГО периода синхроимпульса. ВЫСОКОЕ или НИЗКОЕ состояние линии данных должно меняться, только если линия синхронизации в состоянии НИЗКОЕ (см. [Рисунок 3](#)).

**Рисунок 3. Пересылка бита в шине I<sup>2</sup>C.**

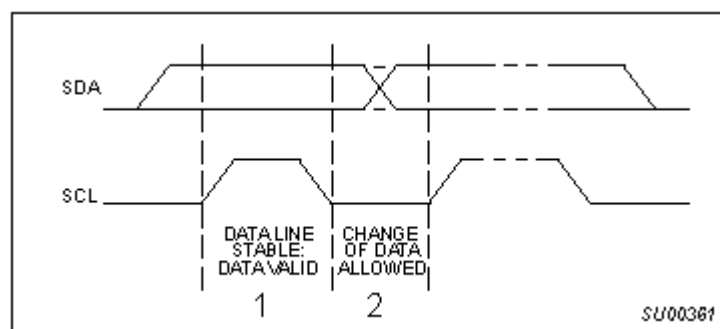


Рисунок 3. Пересылка бита в шине I<sup>2</sup>C:

- Линия данных находится в стабильном состоянии, данные определены
- Допускается изменение данных

## 2.2. Сигналы START и STOP.

Специальные ситуации на шине отмечают сигналы START и STOP (см. [Рисунок 4](#)).

Переход линии SDA из ВЫСОКОГО состояния в НИЗКОЕ, в то время как SCL находится в ВЫСОКОМ состоянии означает START.

Переход линии SDA из НИЗКОГО состояния в ВЫСОКОЕ при SCL в ВЫСОКОМ состоянии означает STOP.

Сигналы СТАРТ и СТОП всегда вырабатываются ведущим. Считается, что шина занята после сигнала СТАРТ. Шина считается освободившейся через определенное время после сигнала СТОП.

Определение сигналов СТАРТ и СТОП устройствами, подключенными к шине достаточно легко, если в них встроены необходимые цепи. Однако микроконтроллеры без таковых цепей должны осуществлять считывание значения линии SDA как минимум дважды за период синхронизации для того, чтобы определить переход состояния.

**Рисунок 4. Сигналы СТАРТ и СТОП.**

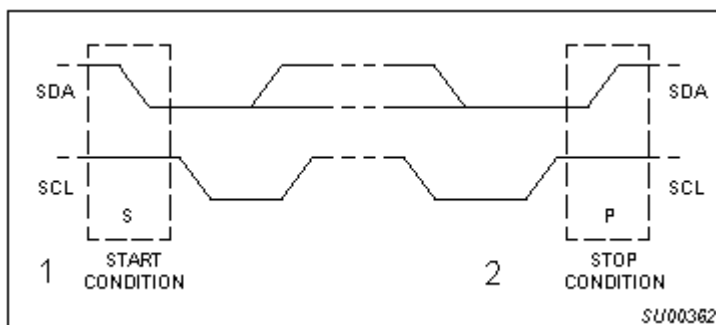


Рисунок 4. Сигналы СТАРТ и СТОП:

- Сигнал СТАРТ
- Сигнал СТОП

### 3.0. Пересылка бита.

Вследствие различных технологий микросхем (КМОП, НМОП, биполярная), которые могут быть подключены к шине, уровни логического нуля ("НИЗКИЙ") и логической единицы ("ВЫСОКИЙ") не фиксированы и зависят от соответствующего уровня Vdd (см. [Раздел 13.0](#) для электрических параметров). Один синхроимпульс генерируется на каждый пересылаемый бит.

#### 3.1. Формат байта.

Каждый байт, передаваемый по линии SDA, должен состоять из 8 бит. Количество байт, передаваемых за один сеанс связи неограничено. Каждый байт должен оканчиваться битом подтверждения. Данные передаются, начиная с наиболее значащего бита (см. [Рисунок 5](#)). Если приёмник не может принять еще один целый байт, пока он не выполнит какую-либо другую функцию (например, обслужит внутреннее прерывание), он может удерживать линию SCL в НИЗКОМ состоянии, переводя передатчик в состояние ожидания. Пересылка данных продолжается, когда приёмник будет готов к следующему байту и отпустит линию SCL.

В некоторых случаях, необходимо использовать другой формат данных (например, CBUS). Посылка, которая передается с таким адресом, может быть закончена выдачей сигнала СТОП, даже если это происходит во время передачи байта. В этом случае подтверждение не генерируется (см. [Раздел 6.1.3](#)).

**Рисунок 5. Пересылка данных по шине I<sup>2</sup>C.**

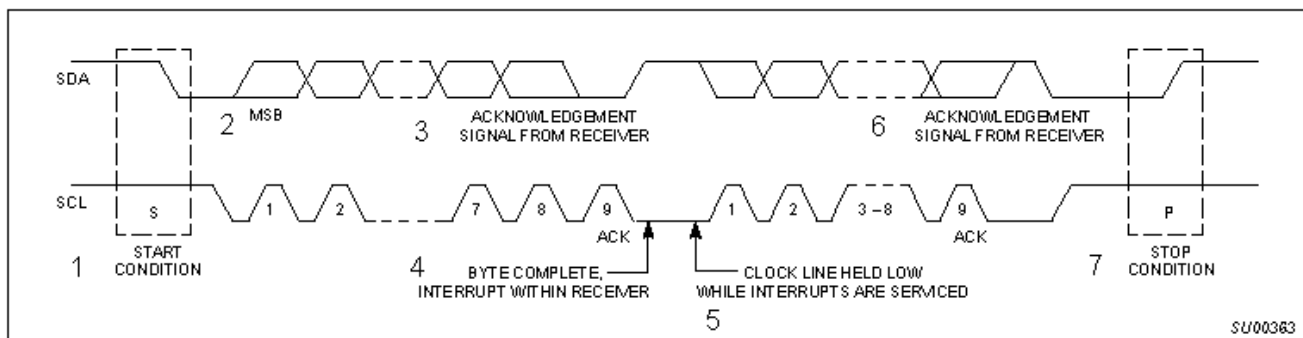


Рисунок 5. Пересылка данных по шине I<sup>2</sup>C:

- Сигнал СТАРТ
- Старший разряд байта
- Сигнал подтверждения от приёмника
- Прием байта завершен. Прерывание внутри приемника
- Синхролиния удерживается в низком состоянии, пока обслуживается прерывание
- Сигнал подтверждения от приемника
- Сигнал СТОП

### 3.2. Подтверждение.

Подтверждение при передаче данных обязательно. Соответствующий импульс синхронизации генерируется ведущим. Передатчик отпускает (ВЫСОКОЕ) линию SDA в течение синхроимпульса подтверждения. Приёмник должен удерживать линию SDA в течение ВЫСОКОГО состояния синхроимпульса подтверждения в стабильно НИЗКОМ состоянии ([Рисунок 6](#)). Конечно, время установки и удержания также должны быть приняты во внимание (см. [Раздел 13.0](#) Электрические и временные параметры).

Обычно, приёмник, который был адресован, обязан генерировать подтверждение после каждого принятого байта, исключая те случаи, когда посылка начинается с адреса CBUS (см. [Раздел 6.1.3](#)).

В том случае, когда ведомый-приёмник не может подтвердить свой адрес (например, когда он выполняет в данный момент какие-либо функции реального времени), линия данных должна быть оставлена в ВЫСОКОМ состоянии. После этого ведущий может выдать сигнал СТОП для прерывания пересылки данных.

Если ведомый-приёмник подтвердил свой адрес, но через некоторое время больше не может принимать данные, ведущий также должен прервать пересылку. Для этого ведомый не подтверждает следующий байт, оставляет линию данных в ВЫСОКОМ состоянии и ведущий генерирует сигнал СТОП.

Если в пересылке участвует ведущий-приёмник, то он должен сообщить об окончании передачи ведомому-передатчику путем не подтверждения последнего байта. Ведомый-передатчик должен освободить линию данных для того, чтобы позволить ведущему выдать сигнал СТОП или повторить сигнал СТАРТ.

**Рисунок 6. Подтверждение.**

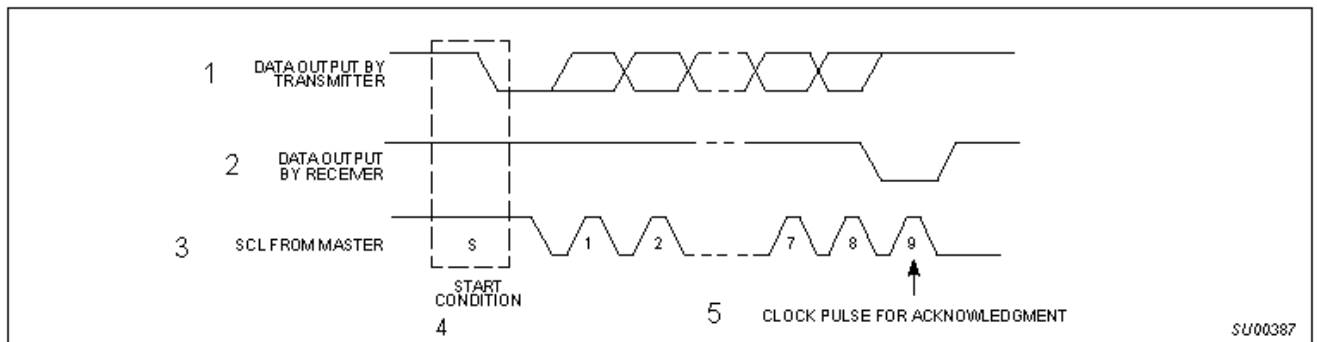


Рисунок 6. Подтверждение:

- Данные, переданные передатчиком
- Данные, переданные приёмником
- SCL от ведущего
- Сигнал СТАРТ
- Синхроимпульс подтверждения

#### 4.0. Синхронизация.

При передаче посылок по шине I<sup>2</sup>C каждый ведущий генерирует свой синхросигнал на линии SCL. Данные действительны только во время ВЫСОКОГО состояния синхроимпульса.

Синхронизация выполняется с использованием подключения к линии SCL по правилу монтажного И. Это означает, что вследствие перехода линии SCL из ВЫСОКОГО состояния в НИЗКОЕ, вызванного переходом синхросигнала одного из устройств в НИЗКОЕ состояние, произойдет также переход синхросигнала другого устройства в НИЗКОЕ состояние.

Это состояние линии SCL удерживается до тех пор, пока не будет достигнуто ВЫСОКОЕ состояние внутреннего синхросигнала одного из устройств ([Рисунок 7](#)). Однако, переход из НИЗКОГО состояния в ВЫСОКОЕ синхросигнала может не вызвать аналогичный переход на линии SCL, если синхросигнал другого устройства все еще находится в НИЗКОМ состоянии. Таким образом, линия SCL будет находиться в НИЗКОМ состоянии на протяжении самого длинного НИЗКОГО периода из двух синхросигналов. Устройства с более коротким НИЗКИМ периодом будут входить в состояние ожидания на время, пока не кончится длинный период.

Когда у всех задействованных устройств кончится НИЗКИЙ период синхросигнала, линия SCL перейдет в ВЫСОКОЕ состояние. Все устройства начнут проходить ВЫСОКИЙ период своих синхросигналов. Первое устройство, у которого кончится этот период, снова установит линию SCL в НИЗКОЕ состояние.

Таким образом, НИЗКИЙ период синхролинии SCL определяется наидлиннейшим периодом синхронизации из всех задействованных устройств, а ВЫСОКИЙ период определяется самым коротким периодом синхронизации устройств.



## Рисунок 7. Синхронизация во время арбитража.

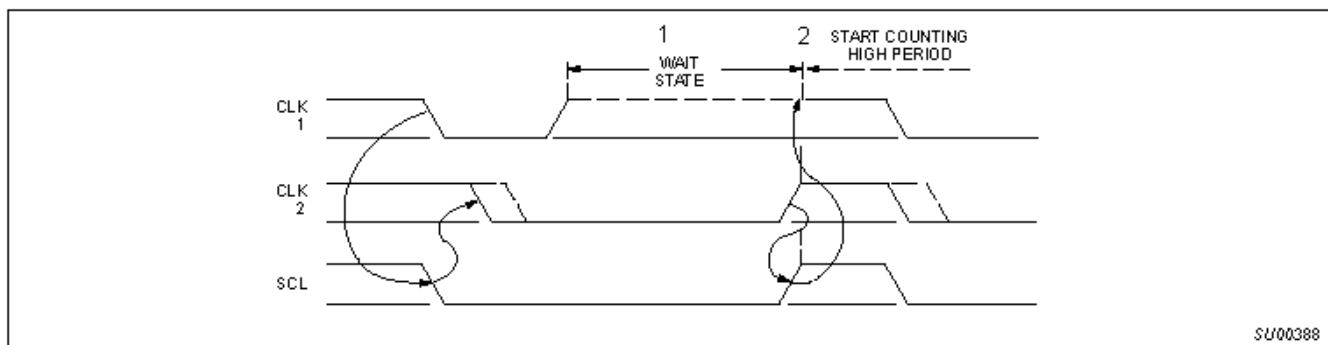


Рисунок 7. Синхронизация во время арбитража:

- Состояние ожидания
- Начало отсчета ВЫСОКОГО периода синхроимпульса

### 4.1. Арбитраж.

Ведущий может начинать пересылку данных только если шина свободна. Два и более ведущих могут сгенерировать сигнал СТАРТ за время минимального удерживания ( $T_{hd;sta}$ ), что ведет к определенному сигналу СТАРТ на шине.

Арбитраж происходит на шине SDA, в периоды, когда шина SCL находится в ВЫСОКОМ состоянии. Если один ведущий передает на линию данных НИЗКИЙ уровень, в то время как другой - ВЫСОКИЙ, то последний отключается от линии, так как состояние SDL (НИЗКОЕ) не соответствует ВЫСОКОМУ состоянию его внутренней линии данных.

Арбитраж может продолжаться на протяжении нескольких бит. Так как сначала передается адрес, а потом, то арбитраж может продолжаться до окончания адреса, а если ведущие адресуют одно и то же устройство, то в арбитраже будут участвовать и данные. Вследствие такой схемы арбитража при столкновении данные не теряются.

Ведущему, проигравшему арбитраж, разрешается выдавать синхроимпульсы на шину SCL до конца байта, в течение которого был потерян доступ.

Если в устройство ведущего также встроены и функции ведомого и он проигрывает арбитраж на стадии передачи адреса, то он немедленно должен переключиться в режим ведомого, так как выигравший арбитраж ведущий мог адресовать его.

[Рисунок 8](#) показывает процедуру арбитража двух ведущих. Конечно, большее количество ведущих может быть вовлечено в процесс. В момент, когда обнаруживается различие между уровнем внутренней линии данных и SDA, выход первого ведущего принимает ВЫСОКОЕ значение, не влияя таким образом на пересылку данных выигравшего ведущего.

Вследствие того, что арбитраж зависит только от адреса и данных, передаваемых соревнующимися ведущими, не существует центрального ведущего, а также приоритетного доступа к шине.

Особое внимание следует обратить на ситуацию, когда во время арбитражной процедуры на шину передается повторный сигнал СТАРТ или сигнал СТОП. Если существует возможность возникновения такой ситуации, то ведущие должны послать повторный сигнал СТАРТ или сигнал СТОП в одних и тех же позициях кадра. Другими словами, арбитраж запрещен между:

- повторным сигналом СТАРТ и битом данных

- сигналом СТОП и битом данных
- повторным сигналом СТАРТ и сигналом СТОП

**Рисунок 8. Арбитраж между двумя ведущими.**

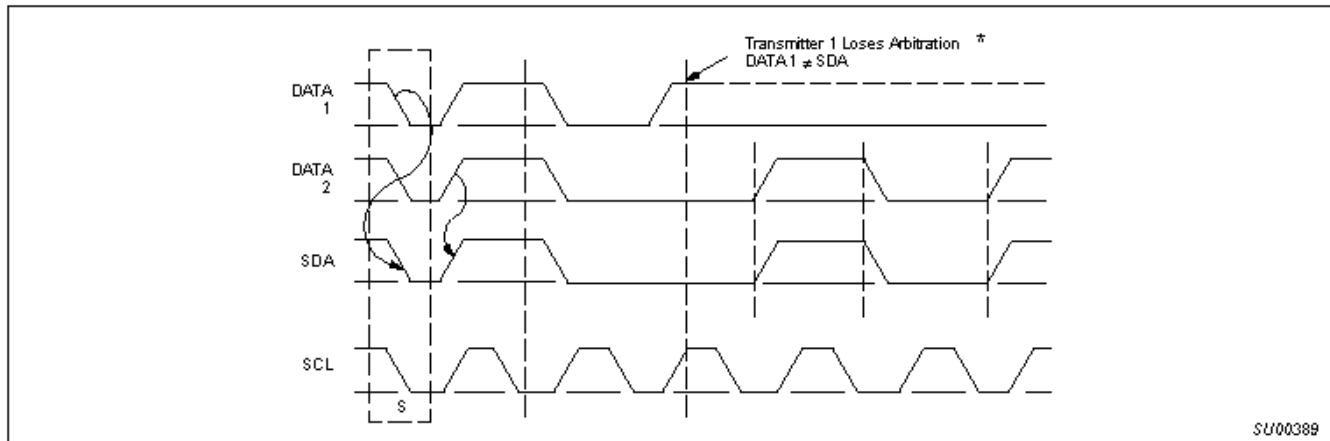


Рисунок 8. Арбитраж между двумя ведущими:

\* передатчик 1 проигрывает арбитраж - его линия данных не совпадает с SDA.

## 4.2. Использование механизма синхронизации как процедуры управления связью.

Кроме использования в процедуре арбитража, механизм синхронизации может быть использован приемниками как средство управления пересылкой данных на байтовом и битовом уровнях.

На уровне байта, если устройство может принимать байты данных с большой скоростью, но требует определенное время для сохранения принятого байта или подготовки к приему следующего, то оно может удерживать линию SCL в НИЗКОМ состоянии после приема и подтверждения байта, переводя таким образом передатчик в состояние ожидания.

На уровне битов, устройство, такое как микроконтроллер без встроенных аппаратных цепей I<sup>2</sup>C или с ограниченными цепями, может замедлить частоту синхроимпульсов путем продления их НИЗКОГО периода. Таким образом скорость передачи любого ведущего адаптируется к скорости медленного устройства.

## 5.0. Форматы с 7-битным адресом.

Посылки данных происходят в формате, показанном на [Рисунок 9](#).

**Рисунок 9. Посылка данных.**

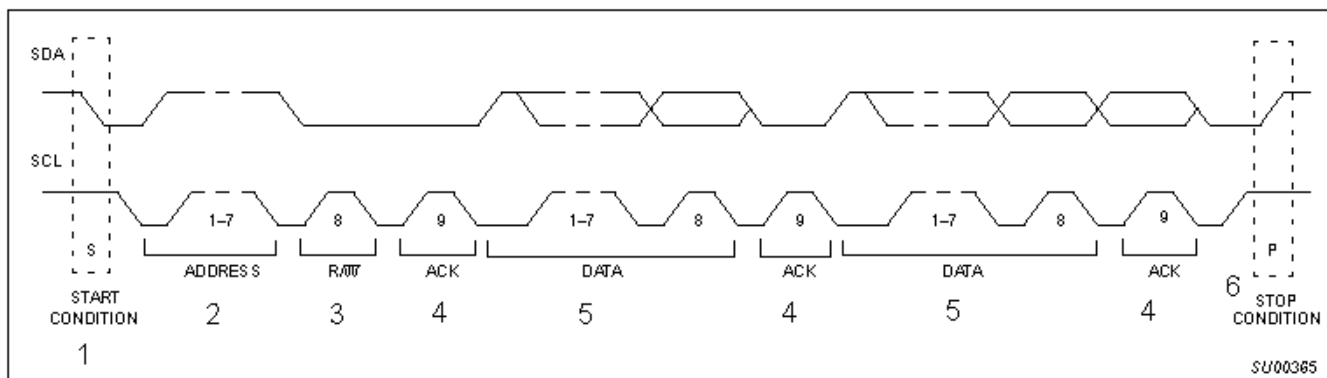


Рисунок 9. Посылка данных:

- Сигнал СТАРТ
- Адрес
- Бит направления (R/W)
- Подтверждение
- Данные
- Сигнал СТОП

После сигнала СТАРТ посылается адрес ведомого. После 7 бит адреса следует бит направления данных (R/W), "ноль" означает передачу (запись), а "единица" - прием (чтение). Пересылка данных всегда заканчивается сигналом СТОП, генерируемым ведущим. Однако, если ведущий желает оставаться на шине дальше, он должен выдать повторный сигнал СТАРТ и затем адрес следующего устройства. При таком формате посылки возможны различные комбинации чтения/записи.

Возможные форматы:

- Ведущий-передатчик передает ведомому-приёмнику. Направление пересылки данных не изменяется (см. [Рисунок 10](#)).
- Ведущий читает ведомого немедленно после пересылки первого байта (см. [Рисунок 11](#)). В момент первого подтверждения ведущий-передатчик становится ведущим-приёмником и ведомый-приёмник становится ведомым-передатчиком.
- Подтверждение тем не менее генерируется ведомым. Сигнал СТОП генерируется ведущим
- Комбинированный формат (см. [Рисунок 12](#)). При изменении направления пересылки данных повторяется сигнал СТАРТ и адрес ведомого, но бит направления данных инвертируется. Если ведущий-приёмник посылает повторный сигнал СТАРТ, он обязан предварительно послать сигнал неподтверждения.

Рисунок 10.

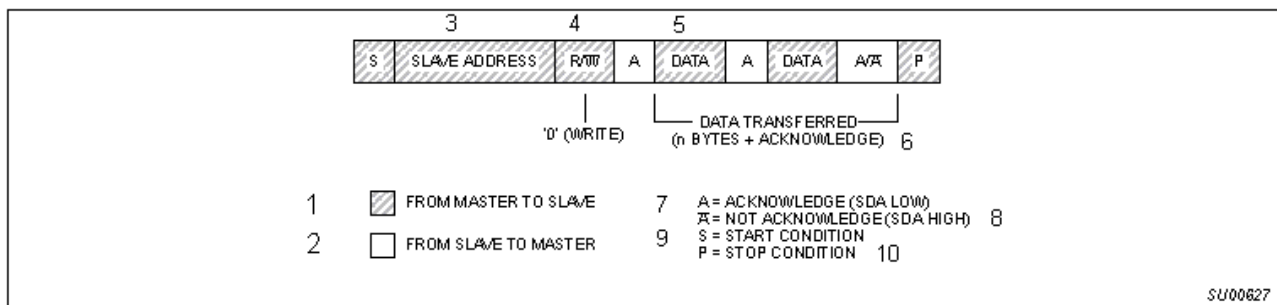


Рисунок 10. Ведущий-передатчик адресует ведомого-приемника 7-битным адресом. Направление пересылки не изменяется:

- От ведущего к ведомому
- От ведомого к ведущему
- Адрес ведомого
- Бит направления
- Данные
- Пересылаемые данные (n байт + подтверждение)
- A - Подтверждение
- Неподтверждение
- Сигнал СТАРТ
- Сигнал СТОП

Рисунок 11.

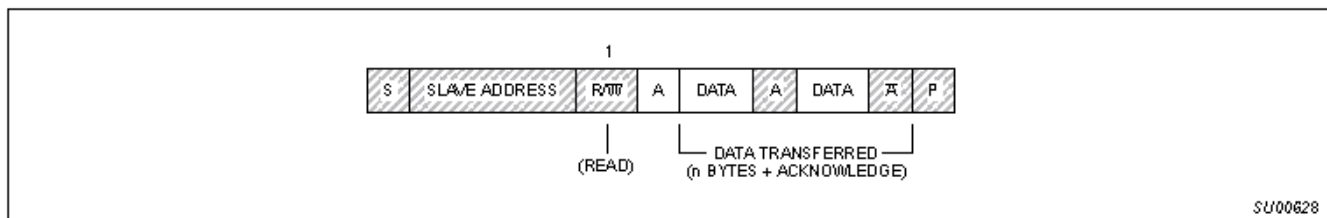


Рисунок 11. Ведущий читает из ведомого непосредственно после первого байта.

Рисунок 12.

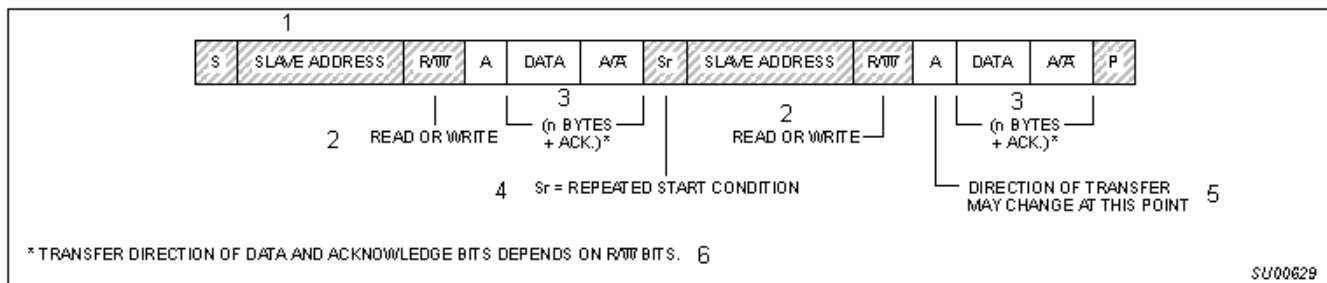


Рисунок 12. Комбинированный формат:

- Адрес ведомого
- Чтение или запись
- n байтов + подтверждения
- Сигнал повторного СТАРТА
- Направление пересылки может измениться в этой точке

- Направление пересылки данных и битов подтверждения зависит от битов направления

## ПРИМЕЧАНИЯ:

- Комбинированные форматы могут быть использованы, например, для управления последовательной памятью. Во время первого байта данных можно передавать адрес в памяти, который записывается во внутреннюю защелку. После повторения сигнала СТАРТа и адреса ведомого выдаются данные из памяти.
- Все решения об авто-инкременте или декременте адреса, к которому произошел предыдущий доступ, принимаются конструктором устройства
- Каждый байт завершается битом подтверждения, обозначенным А или А<sup>^</sup> на рисунках
- I<sup>2</sup>C-совместимые устройства должны сбрасывать логику шины при получении сигнала СТАРТ или повторный СТАРТ и подготавливаться к приему адреса.

## 6.0. 7-битная адресация.

Процедура адресации на шине I<sup>2</sup>C заключается в том, что первый байт после сигнала СТАРТ определяет, какой ведомый выбирается ведущим для работы. Исключение составляет адрес "Общего вызова", который адресует все устройства на шине. Когда используется этот адрес, все устройства в теории должны послать сигнал подтверждения. Однако, устройства могут быть сделаны игнорирующими этот адрес. Второй байт послышки общего вызова определяет действие, которое должны произвести устройства. Более подробно процедура описана в следующем разделе.

### 6.1. Назначение битов первого байта.

Первые семь битов первого байта образуют адрес ведомого (см. [Рисунок 13](#)). Восьмой, младший бит, определяет направление пересылки данных. "Ноль" означает, что ведущий будет записывать информацию в выбранного ведомого. "Единица" означает, что ведущий будет считывать информацию из ведомого.

**Рисунок 13. Первый байт после сигнала СТАРТ.**

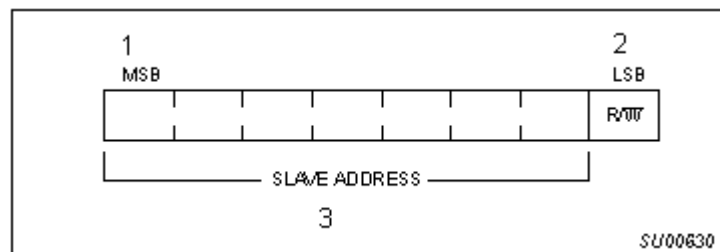


Рисунок 13. Первый байт после сигнала СТАРТ:

- Старший разряд
- Младший разряд
- Адрес ведомого

После того, как адрес послан, каждое устройство в системе сравнивает первые семь бит после сигнала СТАРТ со своим адресом. При совпадении устройство полагает себя выбранным как ведомый-приёмник или как ведомый-передатчик, в зависимости от бита направления.

Адрес ведомого может состоять из фиксированной и программируемой частей. Вероятно, что в системе будет несколько таких одинаковых устройств, поэтому при помощи программируемой части адреса становится возможным подключить к шине максимально возможное количество таких устройств. Количество программируемых бит в адресе зависит от количества свободных выводов микросхемы. Например, если устройство имеет 4 фиксированных и 3 программируемых адресных битов, всего 8 одинаковых устройств может быть подключено к шине.

Комитет I<sup>2</sup>C координирует выделение I<sup>2</sup>C адресов. Две группы по восемь адресов (0000XXX и 1111XXX) зарезервированы для целей, указанных в [таблице 2](#). Комбинация бит 11110XX адреса зарезервирована для 10-битной адресации (см. [Раздел 11.0](#)).

**Таблица 2. Зарезервированные адреса.**

Адрес	R/W бит	Описание
00000000	0	Адрес общего вызова
00000000	1	Байт СТАРТА
00000001	X	Адрес CBUS
0000010	X	Адрес, зарезервированный для шин другого формата
0000011	X	Зарезервирован для дальнейшего использования
00001XX	X	Зарезервирован для дальнейшего использования
11111XX	X	Зарезервирован для дальнейшего использования
11110XX	X	10-битная адресация

#### **ПРИМЕЧАНИЯ:**

- Устройствам запрещается подтверждать прием байта СТАРТА
- Адрес CBUS зарезервирован для того, чтобы можно было использовать CBUS-совместимые и I<sup>2</sup>C-совместимые устройства в одной системе. I<sup>2</sup>C-совместимым устройствам запрещается реагировать на прием этого адреса.
- Адрес, зарезервированный для шин другого формата также предназначен для смешанного использования различных протоколов. Отвечать на прием этого адреса могут только устройства, умеющие работать с другим форматом.

#### **6.1.1. Адрес общего вызова.**

Адрес общего вызова адресует все устройства на шине. Однако, если устройству не нужны какие-либо данные, которые могут быть переданы по общему вызову, оно может игнорировать обращение путем не выдачи подтверждения. Если устройству нужны данные общего вызова, оно генерирует подтверждение и становится ведомым-приёмником. Второй и последующий байты должны подтверждаться каждым ведомым-приёмником, способным обработать эти данные. Если ведомый не может обработать один из байтов, он не генерирует подтверждение. Значение послышки общего вызова всегда определяется вторым байтом (см. [Рисунок 14](#)).

Существуют два варианта:

- когда бит направления второго байта равен нулю
- когда бит направления второго байта равен единице

Когда бит "0", второй байт обладает следующим значением:

- 00000110. Сбросить устройство и записать программируемую часть адреса. При получении этой посылки все устройства сбрасываются и перечитывают программируемую часть их адресов. Перед выдачей команды необходимо убедиться, что устройства после подачи питания не удерживают линии шины в низком состоянии
- 00000100. Записать программируемую часть адреса. Все устройства, имеющие возможность задания программируемой части адреса защелкивают текущее значение адреса при принятии этой команды. Устройства не сбрасываются
- 00000000. Этот код недопустим для использования в качестве второго байта

Остальные коды не установлены и устройства должны игнорировать их.

Когда бит "1", двухбайтовая последовательность называется "аппаратный общий вызов". Это означает, что последовательность передана аппаратным ведущим устройством (таким как сканер клавиатуры), которое не может быть запрограммировано на выдачу конкретного адреса ведомого. Поскольку аппаратный ведущий не знает, какому устройству передается посылка, он может только сгенерировать аппаратный общий вызов и свой собственный адрес - идентифицируя себя для системы (см. [Рисунок 15](#)).

Семь битов, остающихся во втором байте, содержат адрес аппаратного ведущего. Этот адрес распознается умным устройством (микроконтроллером), которое потом будет получать информацию от аппаратного ведущего. Если последний также может работать как ведомый, его адрес совпадает с адресом ведущего.

С другой стороны в некоторых системах аппаратный ведущий-передатчик устанавливается в режим ведомого-приёмника сразу после сброса. Таким способом, ведущий, конфигурирующий систему, может сообщить аппаратному ведущему (который находится в режиме ведомый-приемник), на какой адрес должны посылааться данные (см. [Рисунок 16](#)). После этой процедуры аппаратный ведущий остаётся в режиме ведущий-передатчик.

**Рисунок 14. Формат адреса общего вызова.**

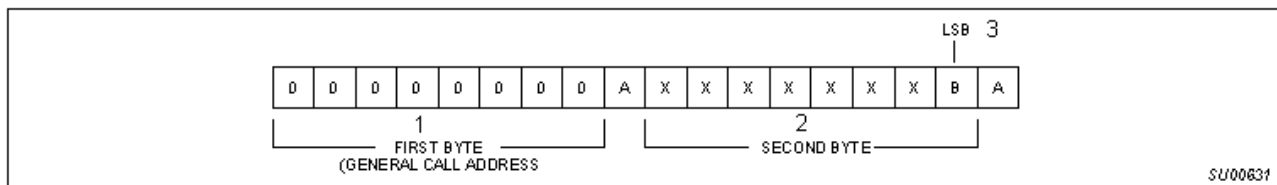


Рисунок 14. Формат адреса общего вызова:

- Первый байт
- Второй байт
- Младший разряд

**Рисунок 15. Пересылка данных из аппаратного ведущего-передатчика.**

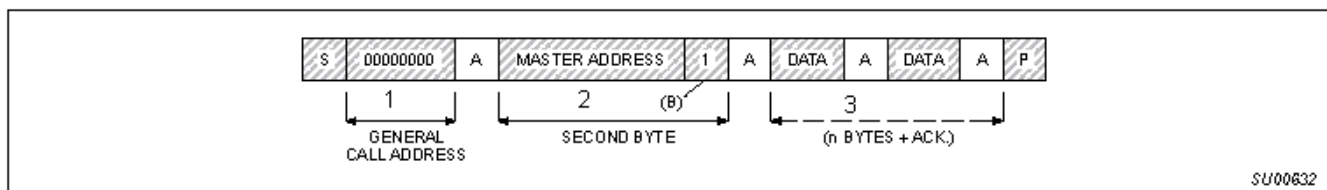


Рисунок 15. Пересылка данных из аппаратного ведущего-передатчика:

- Адрес общего вызова
- Второй байт
- N байт + подтверждение

**Рисунок 16.**

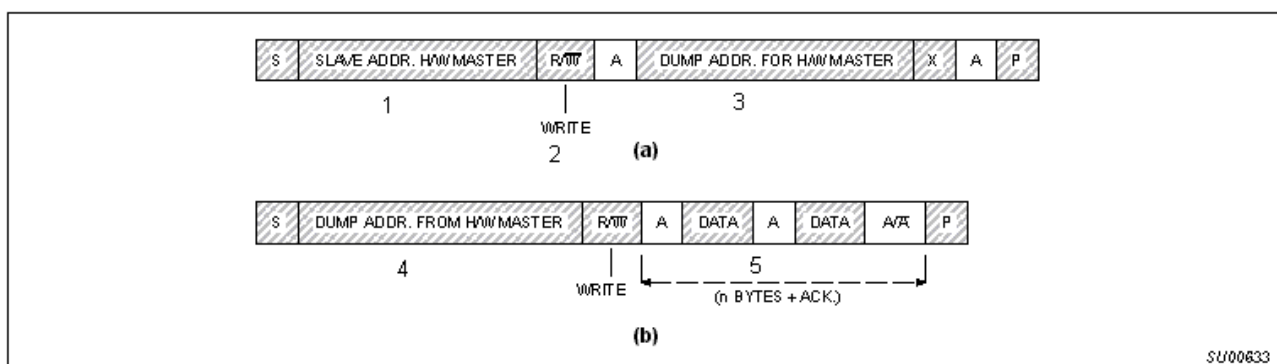


Рисунок 16. Передача данных аппаратным ведущим, способным пересылать данные непосредственно ведомым устройствам

- (a) Конфигурирующий ведущий посылает адрес начала данных к аппаратному ведущему  
 (b) Аппаратный ведущий посылает данные выбранному ведомому

- Адрес аппаратного ведущего
- Запись
- Адрес начала данных для аппаратного ведущего
- Адрес начала данных от аппаратного ведущего
- n байт + подтверждение

### 6.1.2. Байт СТАРТА.

Микроконтроллеры могут быть подключены к шине двумя способами. Микроконтроллер с встроенными шинными цепями разгружен и только реагирует при помощи прерываний на события, происходящие на шине, в то время как микроконтроллер без таких цепей должен постоянно отслеживать состояние шины программным способом. Очевидно, что чем больше времени процессор тратит на обслуживание шины, тем меньше у него остается на основную задачу. Вследствие этого возникает разница между быстрыми аппаратными устройствами и медленными микроконтроллерами, полагающимися на программный опрос.

В этом случае посылка данных может начинаться со стартовой процедуры, которая много дольше, чем обычный сигнал СТАРТ (см. [Рисунок 17](#)). Процедура старта состоит из:

- Сигнала СТАРТ
- Байта СТАРТА



- Импульса подтверждения
- Повторного сигнала СТАРТ

Ведущий-передатчик после обычного сигнала СТАРТ передает байт СТАРТА (00000001). Микроконтроллер поэтому может отслеживать линию SDA с меньшей частотой, пока не обнаружит последовательность из семи нулей, по сути - НИЗКИЙ уровень на SDA на протяжении семи тактовых импульсов. После обнаружения этой последовательности микроконтроллер может переключиться на более высокую частоту опроса шины, для того чтобы обнаружить повторный сигнал СТАРТ.

Аппаратный приёмник I<sup>2</sup>C сбросится при приёме повторного сигнала СТАРТ и поэтому проигнорирует байт СТАРТА.

После байта СТАРТА генерируется тактовый импульс для подтверждения. Он присутствует только для совместимости с форматом байта. Устройствам запрещается подтверждать прием байта СТАРТА.

**Рисунок 17. Процедура байта СТАРТА.**

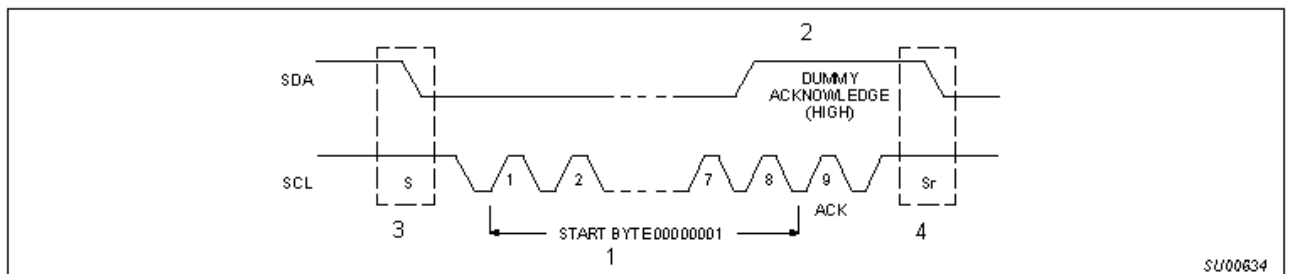


Рисунок 17. Процедура байта СТАРТА:

- Байт СТАРТА
- Фальшивое подтверждение (ВЫСОКОЕ)
- Сигнал СТАРТ (S)
- Сигнал повторный СТАРТ (Sr)

### 6.1.3. Совместимость с CBUS.

Приёмники CBUS могут быть подключены к шине I<sup>2</sup>C. Однако, при этом должна быть введена третья линия DLEN и бит подтверждения должен быть опущен. Обычно послыки I<sup>2</sup>C состоят из 8-битовых байтов, в то время как CBUS-совместимые устройства обладают другим форматом.

В смешанной шине I<sup>2</sup>C-совместимые устройства не должны отвечать на послыки формата CBUS. Для этого зарезервирован специальный адрес CBUS (0000001X). После передачи адрес CBUS линия DLEN может быть сделана активной и посылается посылка формата CBUS ([Рисунок 18](#)). После сигнала СТОП все устройства опять готовы принимать данные. Ведущие-передатчики могут посылать данные в CBUS формате после посылки адреса CBUS. Передача заканчивается сигналом СТОП, распознаваемым всеми устройствами.

#### ПРИМЕЧАНИЕ:

Если конфигурация шины CBUS известна и расширение CBUS-совместимых устройств не предусматривается, конструктору разрешается устанавливать время удержания (DLEN), руководствуясь конкретными требованиями используемых устройств.

**Рисунок 18. Формат данных при посылках с CBUS передатчиком/приемником.**

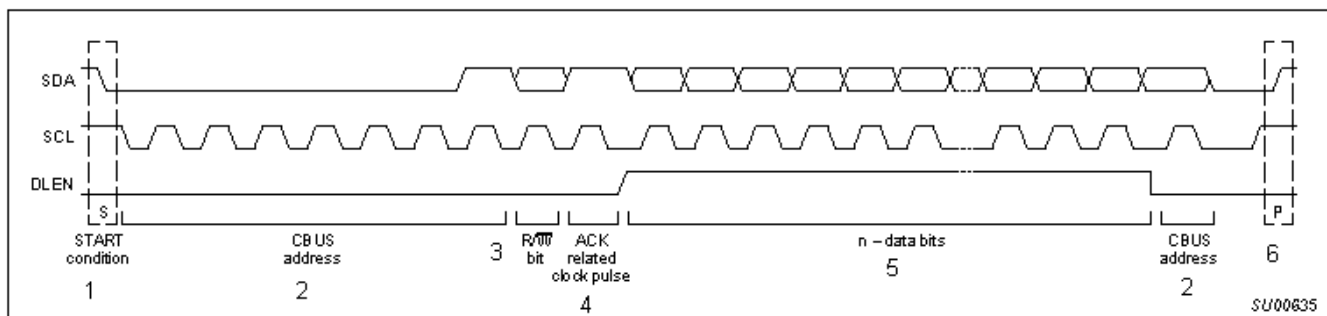


Рисунок 18. Формат данных при посылках с CBUS передатчиком/приемником:

- Сигнал СТАРТ
- Адрес CBUS
- Бит направления передачи
- Синхроимпульс подтверждения
- n бит данных
- Сигнал СТОП

## 7.0. Электрические параметры устройств I<sup>2</sup>C.

Электрические параметры ввода/вывода I<sup>2</sup>C-совместимых устройств и характеристики подключенных к ним линий шины приведены в [таблице 3](#) и [таблице 4 Раздела 13.0](#).

I<sup>2</sup>C-устройства с фиксированными входными уровнями 1.5 В и 3 В могут иметь свои собственные питающие напряжения. Подтягивающие резисторы должны быть подключены к источнику 5 В  $\pm 10\%$  (см. [Рисунок 19](#)). I<sup>2</sup>C-устройства с входными уровнями, зависящими от напряжения питания должны иметь одну общую линию питания, к которой также должен быть подключен подтягивающий резистор (см. [Рисунок 20](#)).

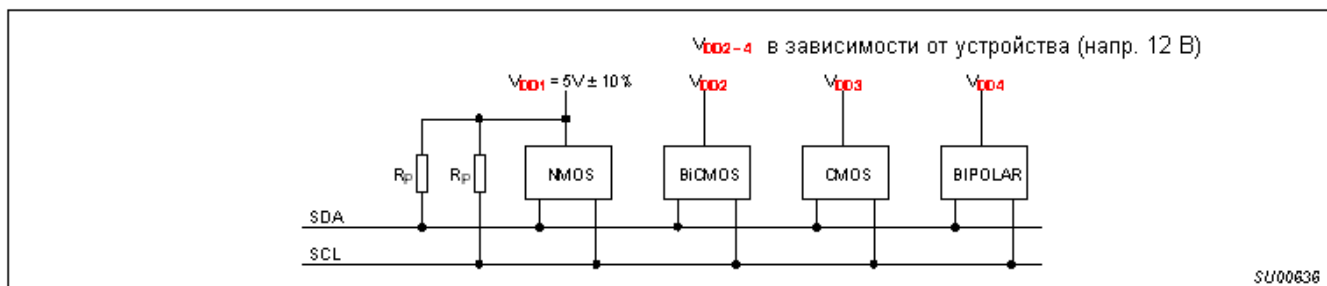
Когда устройства с фиксированным входным уровнем смешаны с устройствами с относительным входным уровнем, последние должны быть подключены к одной общей линии питания 5 В  $\pm 10\%$  и должны иметь подтягивающие резисторы, подключенные к SDA и SCL контактам как показано на [Рисунке 21](#).

Входные уровни определяются следующим образом:

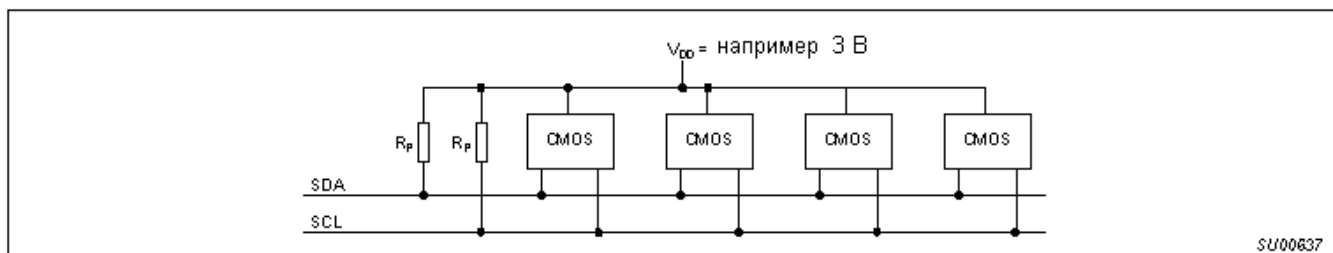
- шумовая граница НИЗКОГО уровня есть 0.1 от напряжения питания
- шумовая граница ВЫСОКОГО уровня есть 0.2 от напряжения питания

Как показано на [Рисунке 22](#), последовательно включенные резисторы  $R_s$  (например, 300 Ом) могут быть использованы для защиты от высоковольтных выбросов напряжения на линиях шины.

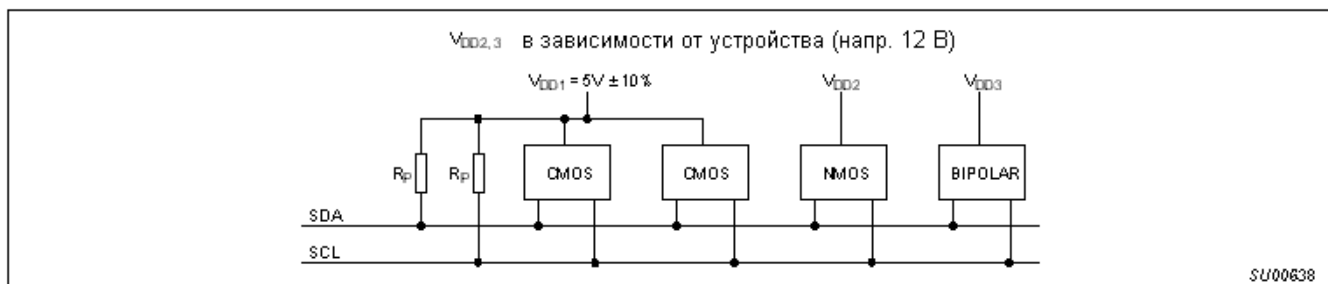
**Рисунок 19. Подключение устройств с фиксированным входным уровнем к шине I<sup>2</sup>C.**



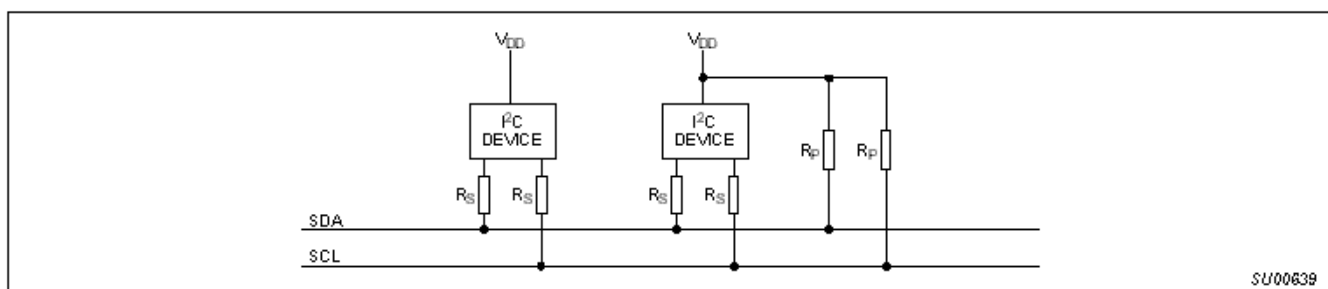
**Рисунок 20. Подключение устройств с широким диапазоном питания к шине I<sup>2</sup>C.**



**Рисунок 21. Подключение устройств с относительным (Vdd1) уровнем входного напряжения и фиксированным входным уровнем (Vdd2-4) к шине I<sup>2</sup>C.**



**Рисунок 22. Последовательные резисторы Rs для защиты от высоковольтных выбросов.**



## 8.0. Максимальные и минимальные значения для резисторов Rp и Rs.

Для I<sup>2</sup>C-устройств в стандартном режиме величины резисторов зависят от следующих параметров:

- Напряжение питания
- Емкость шины
- Количество подключенных устройств (входной ток + ток утечки)

Величина питающего напряжения ограничивает минимальное значение Rp вследствие ограниченного минимального тока выходных каскадов (3 мА) при  $V_{olmax}=0.4$  В. Vdd как функция  $R_{pmin}$  показана на [Рисунке 23](#).

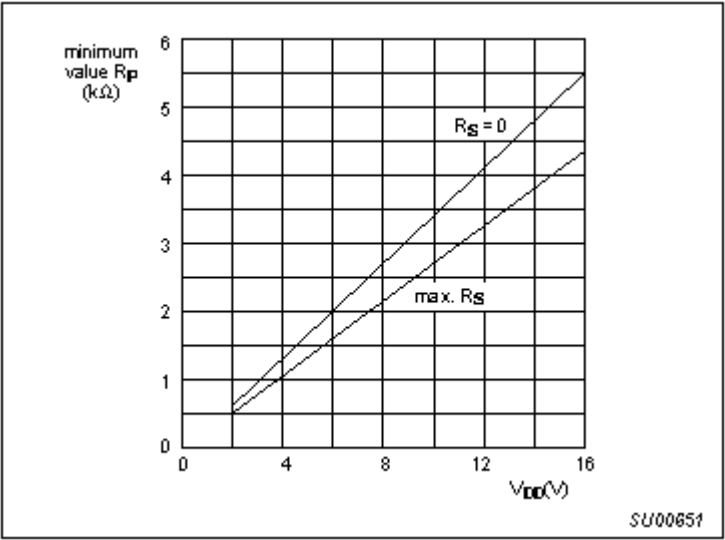
Желаемая шумовая граница  $0.1 V_{dd}$  для НИЗКОГО уровня ограничивает максимальное значение Rs.  $R_{smax}$  как функция от Rp показано на [Рисунке 24](#).

Емкость шины состоит из общей емкости проводов, подключенных портов и контактов. Эта емкость ограничивает максимальное значение Rp вследствие ограничений на время установления (фронта). [Рисунок 25](#) показывает  $R_{pmax}$  как функцию от емкости шины.

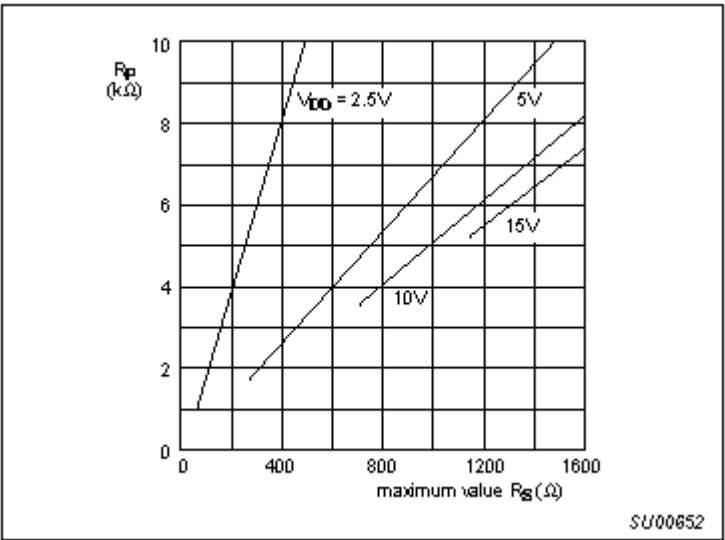
Максимальный ток при ВЫСОКОМ уровне для каждого порта не превышает 10 мкА. Вследствие желаемой шумовой границы  $0.2 V_{dd}$  для ВЫСОКОГО уровня, входной ток

ограничивает максимальное значение  $R_p$ . Этот предел зависит от  $V_{dd}$ . Общий ток при ВЫСОКОМ уровне как функция от  $R_{pmax}$  показан на [Рисунке 26](#).

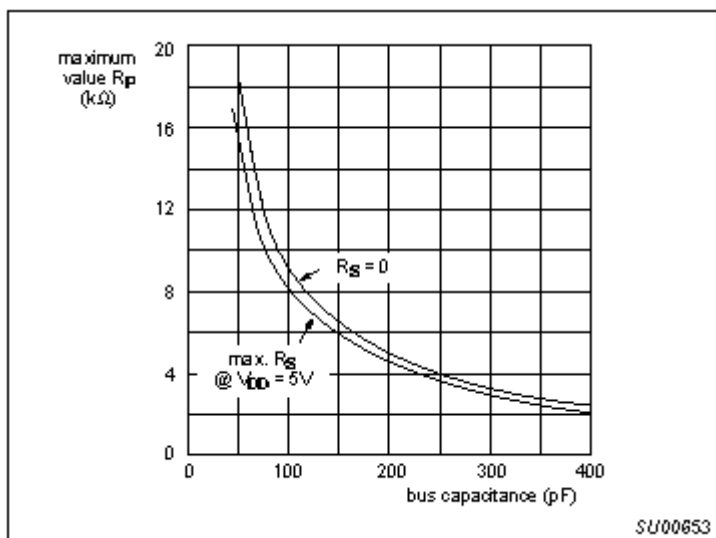
**Рисунок 23. Минимальное значение  $R_p$  как функция от напряжения питания при параметре  $R_s$ .**



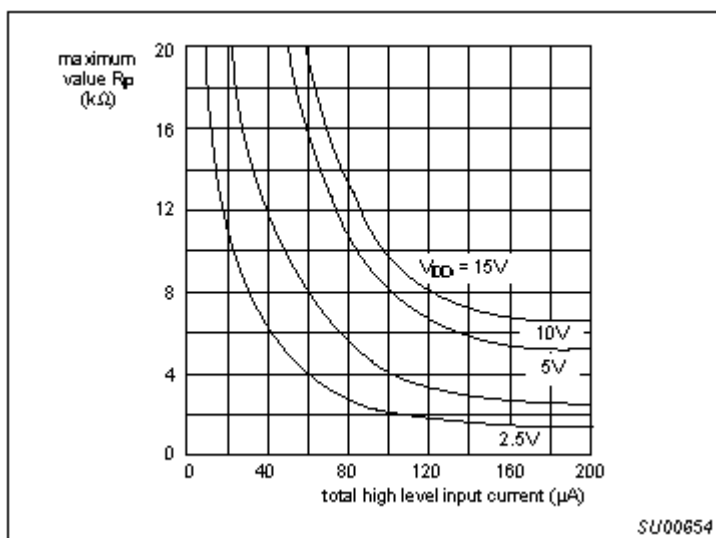
**Рисунок 24. Максимальное значение  $R_s$  как функция от  $R_p$  при параметре -  $V_{dd}$ .**



**Рисунок 25. Максимальное значение  $R_p$  как функция от емкости шины для стандартного режима.**



**Рисунок 26. Общий ток при напряжении ВЫСОКОГО уровня как функция максимального значения  $R_p$  при параметре - Vdd.**



## 9.0. Дополнения к спецификации шины I<sup>2</sup>C.

Шина I<sup>2</sup>C со скоростью передачи данных 100 кбит/с и 7-битным адресом существует уже на протяжении более 10 лет в неизменном виде. Концепция принята повсеместно как стандарт для сотен типов микросхем, выпускаемых фирмой Philips и другими поставщиками. В настоящее время спецификация шины I<sup>2</sup>C дополнена следующими вещами:

- Быстрый режим, позволяющий в четыре раза увеличить скорость передачи данных
- 10-битная адресация, позволяющая использовать 1024 дополнительных адресов

Существуют две причины, ради которых были сделаны эти дополнения:

- Новые приложения нуждаются в пересылке больших объемов информации, следовательно требуется большая пропускная способность шины. Улучшенная

технология производства микросхем позволила в четыре раза увеличить скорость передачи данных без изменения себестоимости изделия.

- Большинство из 112 адресов, допустимых при 7-битной адресации, уже были использованы более чем один раз. Для предотвращения проблем с размещением адресов новых устройств, желательно иметь большее количество адресных комбинаций. Примерно десятикратное увеличение количества доступных адресов получено при использовании новой 10-битной адресации.

Устройства с I<sup>2</sup>C интерфейсом, работающие в быстром режиме, как минимум должны быть способны входить в синхронизацию в быстром режиме, с тем чтобы снизить скорость передачи (путем удлинения НИЗКОГО периода SCL) до допустимой величины. Быстрые устройства должны быть совместимы снизу-вверх, что означает их способность работать со стандартными устройствами по медленной шине.

Очевидно, что стандартные устройства не способны работать в быстрой шине, потому что они не могут синхронизироваться на высокой скорости и их состояние станет непредсказуемым.

Ведомые быстрые устройства могут обладать как 7-битным, так и 10-битным адресом. Однако, 7-битный адрес более предпочтителен, так как его аппаратная реализация более проста и длина посылки меньше.

Устройства с 7-битным и 10-битным адресами могут одновременно использоваться на одной шине, независимо от скорости передачи.

## **10.0. Быстрый режим.**

В быстром режиме протокол, формат, логические уровни и максимальная емкостная нагрузка линий шины остается неизменными. Изменения в спецификации таковы:

- максимальная скорость передачи возросла до 400 кбит/с
- Синхронизация SDA и SCL линий была изменена. Не требуется совместимости с CBUS-устройствами, так как они не могут работать на высоких скоростях
- Входные цепи быстрых устройств должны иметь встроенное подавление выбросов и триггер Шмитта на обеих линиях
- Выходной буфер быстрых устройств должен иметь каскад с управлением временем заднего фронта линий SDA и SCL
- Если источник напряжения питания быстрых устройств выключается, линии должны переходить в третье состояние
- Внешние подтягивающие устройства, подключенные к линиям шины должны быть изменены для обеспечения допустимого времени нарастания переднего фронта. Для нагрузок шины до 200 пФ это подтягивающее устройство может быть простым резистором, а для нагрузок от 200 пФ до 400 пФ это должен быть источник тока (3 мА максимум) или схема на переключаемых резисторах.

## **11.0. 10-битная адресация.**

10-битная адресация не меняет формат шины. Для этого используется зарезервированная адресная комбинация 1111XXX первых семи бит первого байта (см. [Раздел 6.0](#)). 10-битная адресация не влияет на существующую 7-битную адресацию. Устройства с 7-битной и 10-битной адресацией могут быть подключены к одной шине.

Хотя имеются восемь возможных комбинаций последовательности 1111XXX, из них используются только четыре - 11110XX. Комбинации типа 11111XX зарезервированы для дальнейших улучшений шины.

## 11.1. Назначение битов первых двух байтов.

10-битный адрес формируется из первых двух байтов. Первые семь бит первого байта являются комбинацией вида 11110XX, где два младших бита (XX) являются двумя старшими (9 и 8) битами 10-битного адреса; восьмой бит первого байта - бит направления. "Ноль" в этом бите означает, что ведущий собирается записывать информацию в ведомого, а "единица" - что ведущий будет считывать информацию из ведомого.

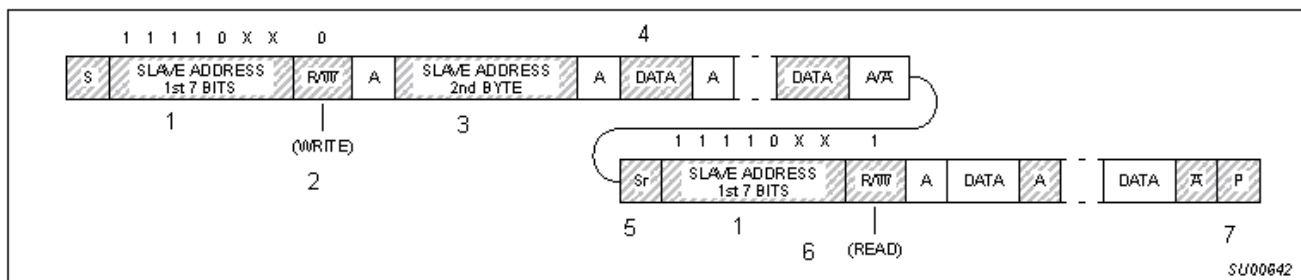
Если бит направления равен "нулю", то второй байт содержит оставшиеся 8 бит 10-битного адреса. Если бит направления равен "единице", то следующий байт содержит данные, переданные с ведомого ведущему.

## 11.2. 10-битные форматы.

Возможны различные комбинации форматов 10-битных посылок:

- Ведущий-передатчик передает информацию ведомому-приемнику. Направление пересылки не меняется. Когда за сигналом СТАРТ появляется начало 10-битного адреса, каждый ведомый на шине сравнивает первые семь бит первого байта со своим собственным адресом и удостоверяется, что бит направления равен "нулю". Допустимо, чтобы более одного устройства обнаружили совпадение и сгенерировали сигнал подтверждения (A1). Все ведомые с совпавшими адресами продолжают сравнивать последующие 8 бит адреса, и только одно устройство обнаруживает совпадение и генерирует подтверждение (A2). Это устройство остается выбранным, пока ведущий не пошлет сигнал СТОП или сигнал повторного СТАРТА с другим адресом.
- Ведущий-приемник принимает данные от ведомого-передатчика. Направление передачи меняется после второго бита направления. Процедура идентична вышеописанной вплоть до момента второго подтверждения (A2). Далее передается сигнал повторного СТАРТА. Выбранный ведомый помнит, что был адресован ранее. Этот ведомый сравнивает первые семь бит адреса со своим адресом, а также удостоверяется, что бит направления равен "единице". При совпадении ведомый полагает, что он адресован как передатчик и генерирует подтверждение (A3). Ведомый-передатчик остается адресованным до прихода сигнала СТОП или сигнала повторного СТАРТА с другим адресом. После сигнала повторного СТАРТА все другие устройства также сравнивают первые семь бит со своим адресом и проверяют бит направления. Однако, ни одно из них не адресуется, так как бит направления равен "единице" (для 10-битных устройств), или такого адреса не существует (для 7-битных устройств).
- Комбинированный формат. Ведущий передает данные ведомому, а потом читает данные с этого же ведомого (см. [Рисунок 27](#)). Один ведущий занимает шину на все время пересылки. Направление пересылки меняется после второго бита направления
- Комбинированный формат. Ведущий передает данные одному ведомому, а потом передает данные другому ведомому (см. [Рисунок 28](#)). Один ведущий занимает шину на все время пересылки
- Комбинированный формат. 10-Битная и 7-битная адресация применяются в одной посылке (см. [Рисунок 29](#)). После каждого сигнала СТАРТ или повторный СТАРТ передается 10-битный или 7-битный адрес. [Рисунок 30](#) показывает, как ведущий передает данные ведомому с 7-битным адресом, а потом передает данные второму ведомому с 10-битным адресом. Один ведущий занимает шину на все время пересылки.

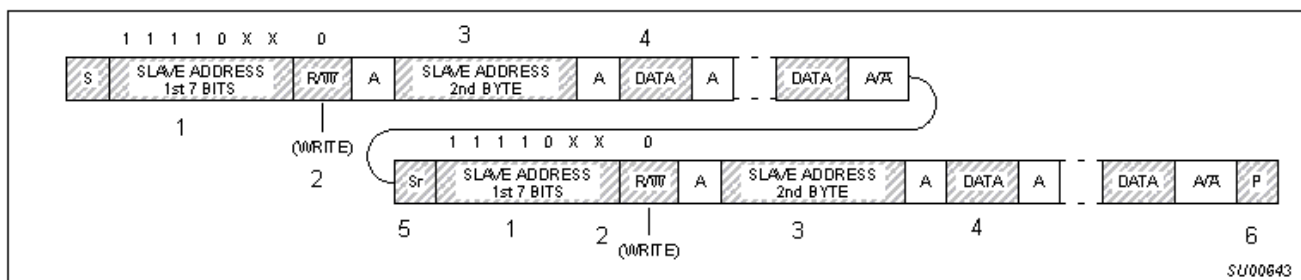
**Рисунок 27. Комбинированный формат. Ведущий адресует ведомого 10-битным адресом, потом передает ему данные и принимает из него данные.**



**Рисунок 27. Комбинированный формат. Ведущий адресует ведомого 10-битным адресом, потом передает ему данные и принимает из него данные.**

- Адрес ведомого (первые 7 бит)
- Запись
- Адрес ведомого (второй байт)
- Данные
- Сигнал повторный СТАРТ
- Чтение
- Сигнал СТОП

**Рисунок 28. Комбинированный формат. Ведущий передает данные двум ведомым, оба 10-битные.**

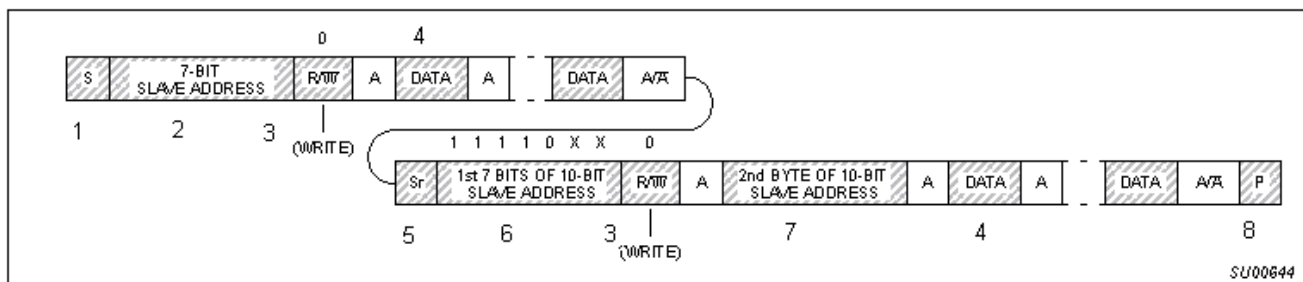


**Рисунок 28. Комбинированный формат. Ведущий передает данные двум ведомым, оба 10-битные**

- Адрес ведомого (первые 7 бит)
- Запись
- Адрес ведомого (второй байт)
- Данные
- Сигнал повторный СТАРТ
- Сигнал СТОП



**Рисунок 29. Комбинированный формат. Ведущий передает данные двум ведомым, один из них 7-битный, второй - 10-битный.**



**Рисунок 29. Комбинированный формат. Ведущий передает данные двум ведомым, один из них 7-битный, второй - 10-битный**

- Сигнал СТАРТ
- 7-битный адрес ведомого
- Запись
- Данные
- Сигнал повторный СТАРТ
- Адрес ведомого (первые 7 бит)
- Адрес ведомого (второй байт)
- Сигнал СТОП

#### ПРИМЕЧАНИЯ:

- Комбинированные форматы могут быть использованы, например, для управления последовательной памятью. Во время первого байта данных можно передавать адрес в памяти, который записывается во внутреннюю защелку. После повторения сигнала СТАРТа и адреса ведомого выдаются данные из памяти.
- Все решения об авто-инкременте или декременте адреса, к которому произошел предыдущий доступ, принимаются конструктором устройства
- Каждый байт завершается битом подтверждения, обозначенным A или A<sup>^</sup> на рисунках
- I<sup>2</sup>C-совместимые устройства должны сбрасывать логику шины при получении сигнала СТАРТ или повторный СТАРТ и подготавливаться к приему адреса.

#### 12.0. Адрес общего вызова и байт СТАРТА.

Процедура 10-битной адресации такова, что первые два байта после сигнала СТАРТ определяют адрес ведомого. Исключение составляет адрес "общего вызова" 00000000. Ведомые устройства с 10-битной адресацией должны реагировать на "общий вызов" также, как и устройства с 7-битной адресацией. Аппаратные ведущие могут передавать свои 10-битные адреса после "общего вызова". В этом случае, с байтом "общего вызова" следуют два байта, содержащих 10-битный адрес ведущего-передатчика. Формат совпадает с показанным на рис. 15, только первый байт ДАННЫХ содержит 8 младших разрядов адреса.

Байт СТАРТА 00000001 может предварять 10-битный адрес точно таким же способом, как и при 7-битной адресации (см. [Раздел 6.0](#)).

### 13.0. Электрические и временные параметры.

Уровни ввода/вывода, ток, подавление выбросов, управление скоростью нарастания и емкость контактов для I<sup>2</sup>C-совместимых устройств даны в [таблице 3](#). Временные параметры (см. [Рисунок 30](#)) даны в [таблице 4](#).

Шумовые границы для ВЫСОКОГО и НИЗКОГО уровней для быстрых устройств совпадает с границами для стандартных устройств (см. [Раздел 7.0](#)).

Минимальное значение НИЗКОГО и ВЫСОКОГО периодов линии SCL, указанные в [таблице 4](#), определяет максимальные скорости передачи 100 кбит/с и 400 кбит/с соответственно. Стандартные и быстрые устройства должны быть способны передавать данные на этих скоростях, либо, как минимум, синхронизироваться и снижать скорости до требуемого значения. Конечно, в последнем случае скорость передачи данных будет определяться возможностями устройства, а не шины.

**Таблица 3. Электрические параметры.**

Параметр	Символ	Стандарт		Быстрый		Ед
		мин	макс	мин	макс	
Входное напряжение НИЗКОГО уровня, фиксированное зависящее от Vdd(V)	Vil	-0.5 - 0.5	1.5 - 0.3	-0.5 - 0.5	1.5 0.3	В
Входное напряжение ВЫСОКОГО уровня, фиксированное зависящее от Vdd(V)	Vih	3.0 0.7	*1) *1)	3.0 0.7	*1) *1)	В
Гистерезис входного триггера Шмитта фиксированные вх уровни зависящие от Vdd(V)	Vhys	- -	- -	0.2 0.05	- -	В
Ширина импульсов выбросов, которые должен подавлять входной фильтр	Tsp	-	-	0	50	нс
Выходное напряжение НИЗКОГО уровня (открытый сток или открытый коллектор) при токе 3 мА при токе 8 мА Vol1	Vol2	0 -	0.4 -	0 0	0.4 0.6	В
Время падения входного напряжения от Vihmin до Vilmax при емкости шины от 10 до 400 пФ при токе 3 мА на Vol1 при токе 8 мА на Vol2	Tof	- -	250* -	20 + 0.1C **	250 250*	нс
Входной ток любого входа при входном напряжении между 0.4V и 0.9V	Ii	10	-10	-10 ***	10 ***	мкА
Емкость любого контакта	Ci	-	10		10	пФ

**ПРИМЕЧАНИЯ:**

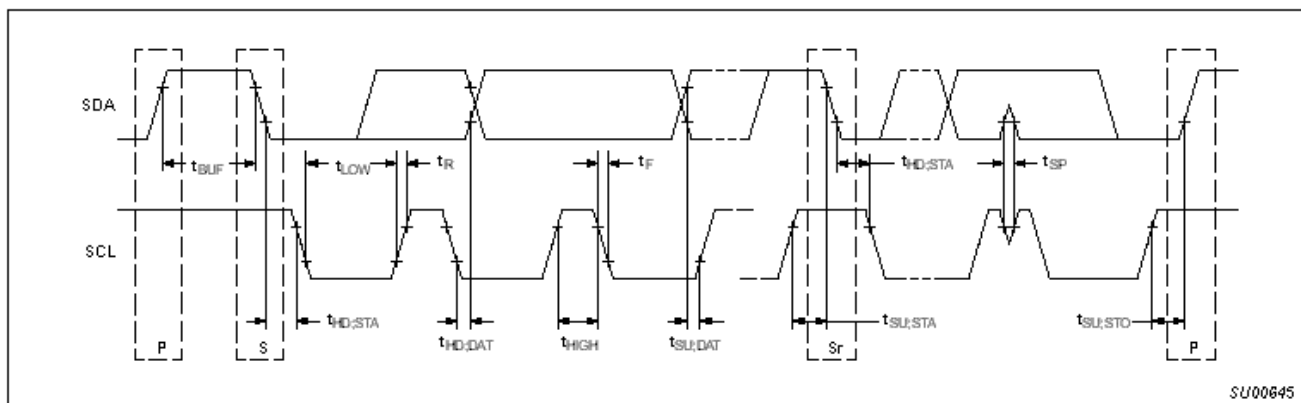
\*1) Максимальное Vih = Vddmax + 0.5 В

\* Максимальное время Tf для линий SDA и SCL, указанное в [таблице 4](#) (300 нс) больше данного максимального Tof. Это позволяет подключить последовательные резисторы Rs (рис 37) без превышения максимального значения Tf.

\*\* C - емкость одной линии шины в пФ.

\*\*\* Порты ввода/вывода быстрых устройств при выключенном питании не должны нарушать работу шины.

**Рисунок 30. Временная диаграмма работы шины I<sup>2</sup>C.**



SU00645

**Таблица 4. Временные параметры.**

Параметр	Символ	Стандарт		Быстрый		Ед
		мин	макс	мин	макс	
Частота SCL	fsc1	0	100	0	400	КГц
Время свободной шины - между сигналами СТОП и СТАРТ	Tbuf	4.7	-	1.3	-	мкс
Длительность сигнала (повторного) СТАРТА. После окончания этого периода может быть сгенерирован первый синхроимпульс	Thd;sta	4.0	-	0.6	-	мкс
НИЗКИЙ период SCL	Tlow	4.7	-	1.3	-	мкс
ВЫСОКИЙ период SCL	Thigh	4.0	-	0.6	-	мкс
Время установления сигнала повторного СТАРТА	Tsu;sta	4.7	-	0.6	-	мкс
Время удержания данных для CBUS совместимых для I <sup>2</sup> C-устройств	Thd;dat	5.0 0*	- -	- 0*	- 0.9**	мкс мкс
Время установления данных	Tsu;dat	250	-	100***	-	нс
Время нарастания SDA, SCL	Tr	-	1000	20+0.1C	300	
Время спада SDA, SCL	Tf	-	300	20+0.1C	300	нс
Время установления сигнала СТОП	Tsu;sto	4.0	-	0.6	-	мкс
Емкостная нагрузка на каждую линию шины	C	-	400	-	400	пФ

**ПРИМЕЧАНИЯ:**

Все значения относительно уровней Vihmin и Vilmax (табл 3)

\* Устройство должно обеспечить время удержания не менее 300 нс для сигнала SDA (относительно Vihmin сигнала SCL), чтобы данные не менялись во время изменения SCL.

\*\* Время удержания должно достигать максимума только если устройство не вытягивает НИЗКИЙ период сигнала SCL.

\*\*\* Быстрые устройства могут быть использованы в медленной шине, но требование Tsu;dat > 250 нс должно выполняться. Это происходит автоматически, если устройство не вытягивает НИЗКИЙ период SCL. Если же устройство вытягивает НИЗКИЙ период, оно должно выдать следующий бит на линию данных SDA за время T до отпускания линии SCL, где T = Trmax + Tsu;dat = 1000 + 250 = 1250 нс (в соответствии со спецификацией стандартного режима шины).

\*\*\*\* С - общая ёмкость шины в пФ.

## 14.0. Практические рекомендации.

Ниже мы рассмотрим основные трудности, связанные с использованием шины I<sup>2</sup>C в целом, типичные ошибки, возникающие на начальных этапах реализации алгоритмов обмена и особенности стыковки с ЭКЛЗ.

### 14.1. Особенность реализации интерфейса с шиной I<sup>2</sup>C в ЭКЛЗ.

Электрический интерфейс шины I<sup>2</sup>C реализован на микросхемах LVC07A с открытым коллектором с нагрузочными резисторами на линиях SCL и SDA номиналом 5.6 кОм. По электрическим характеристикам данная реализация соответствует требованиям стандарта Philips на шину I<sup>2</sup>C.

Анализируемое устройство обрабатывает интерфейс I<sup>2</sup>C в режиме "Slave" и реализует протокол I<sup>2</sup>C режима "Slave" программным способом. Временные параметры реализованного интерфейса укладываются в требования стандарта Philips на шину I<sup>2</sup>C (см. [Таблицу 4](#)) для стандартных устройств со скоростью передачи данных до 100 кбит/с.

"Slave"-адрес ЭКЛЗ на шине I<sup>2</sup>C в данной версии программного обеспечения 7-и битный, фиксирован, не может быть изменен и равен 02h.

ЭКЛЗ не реагирует на команды "общий вызов" и "байт СТАРТА", совместим для совместной работы с 10-и битово адресуемыми устройствами.

Подстройку скорости обмена данными под свои возможности ЭКЛЗ производит путем перевода линии SCL в НИЗКОЕ состояние. ЭКЛЗ не способен работать в "быстром" режиме (400 кГц на шине SCL).

### 14.2. Рекомендации по созданию "Master"- устройства.

I<sup>2</sup>C-абоненты жестко разделяются по классам: "Master"- и "Slave"- устройство. Тот факт, что сигнал SCL всегда генерируется "Master"-устройством означает, что "Master"-абонент может быть достаточно легко реализован чисто программными средствами, так как все изменения на шине будут происходить только по сигналу SCL. (Реализацию "Slave"-устройства мы в данном руководстве не рассматриваем, поскольку сопрягаемое с ЭКЛЗ устройство должно работать в режиме "Master")

Типичная ошибка при реализации программ "Master"-абонента - управление значением порта для установки нулевого и единичного состояний линий SCL и SDA. Если для, например, ОМЭВМ семейства MCS-51 это нормальный режим работы, так как единичное состояние порта у них реализуется встроенным подтягивающим резистором = 50 кОм, то для ОМЭВМ с симметричными портами ([Motorola 68HCxx](#), [Microchip PIC](#), [Atmel AVR](#)), а так же при использовании LPT-порта IBM-совместимого компьютера в однонаправленном режиме вывода (линии LPT-порта D0-D7), это будет порождать электрические конфликты. Например, в руководстве "Microchip. Embedded Control Handbook 1994/1995" приведены практические программы для связи PIC с шиной I<sup>2</sup>C, содержащие подобные грубые ошибки. Положение усугубляется тем, что в случае взаимодействия по шине I<sup>2</sup>C с некоторыми микросхемами (например, EEPROM) такой вариант может сработать, так как они являются 100% аппаратными схемами и не вносят задержек в связной протокол, а паузу ожидания окончания цикла программирования производят переходом в пассивное состояние. Использование таких подпрограмм с устройствами, производящими захват линии SCL (а в случае ЭКЛЗ так оно и есть), приведет к невозможности связи, а возможно, и к выходу одного из сопрягаемых устройств из строя.

Реализовать настоящую имитацию ОК (мы назвали этот режим имитацией ОК, так он не позволяет устанавливать на линии напряжение выше напряжения питания, что было бы

нормально для настоящего ОК, но так как по спецификации I<sup>2</sup>C напряжение на линиях SCL и SDA не должно превышать напряжение питания, его вполне законно можно считать выходом с ОК) на порте с симметричным выходом можно, если установить значение порта постоянно в ноль, а управлять состоянием линии через манипуляции с регистром направления данных. Для ОМЭВМ PIC это будет регистр "TRISx", переводящий порт либо в третье состояние, либо подключающий линии в соответствии с состоянием регистра "PORTx". Практически так же это реализуется в AVR и MC68HC05 (08, 11), где "DDRx" коммутирует порт "PORTx", с той лишь разницей, что у них другая полярность управляющего сигнала - у PIC ноль в "TRISx" соответствует нулю на выходе, а у AVR и MC68HC05 единица в "DDRx" соответствует нулю на выходе.

Еще одна тонкость, не имеющая собственно к I<sup>2</sup>C никакого отношения, заключается в том, что у некоторых ОМЭВМ совмещены регистр чтения состояния линий порта и регистр записи значения порта, в результате чего, если часть линий порта запрограммирована на вывод, а другая находится в третьем состоянии, то при выполнении последовательности операции "чтение из порта" - "модификация отдельных линий порта" - "запись в порт" - "активизация линий, находившихся в третьем состоянии" может произойти изменение ранее предустановленных состояний отдельных линий, находящихся в момент чтения в третьем состоянии. Это может нарушить (и скорее всего нарушит) работу программы обмена (потеряются нули, обеспечивающие имитацию выходов с ОК). Поэтому лучше производить принудительную установку в ноль битов перед каждой операцией обмена.

Для IBM-совместимых ПЭВМ для реализации шины I<sup>2</sup>C на базе LPT-порта рекомендуется использовать не регистр данных DR, а регистр управления CR, выходы которого построены по схеме открытого коллектора. Но и в этом случае следует проявлять осторожность, поскольку регистр управления LPT-порта имеет открытый коллектор только в режиме "Normal", а в режимах EPP и ECP этот регистр может оказаться с симметричным выходом.

Другая важная сторона вопроса - необходимость тщательного соблюдения параметров временной диаграммы процесса обмена. Несмотря на то, что шина I<sup>2</sup>C асинхронная и позволяет "Slave"-абоненту затягивать передачу бита (байта) на сколь угодно длительное время, требования к минимальным значениям длительностей импульсов очень жесткие. Ситуация усугубляется тем, что положительные перепады состояния линии имеют склонность затягиваться, так как несимметричные управляющие выходы не могут создать крутые положительные фронты.

При написании программ очень важно контролировать время между операциями на шине, реализуемыми различными подпрограммами, например выдача "Start" и "Stop"-условия, передача бита, передача байта. При состыковке этих подпрограмм не должны быть нарушены минимальные значения времени, что очень легко происходит при использовании высокоскоростных процессоров. Кроме того, необходимо следить, чтобы время между изменением на линии SDA и стробированием положительным импульсом на линии SCL было не меньше половины минимальной длительности полупериода SCL ( $4.7 \text{ мкс} / 2 = 2.4 \text{ мкс}$ .)

Еще одна распространенная ошибка - игнорирование требования слежения за захватом линии SCL "Slave"-абонентом. Грамотно реализованные программы операций "Master"-абонента должны контролировать возврат линии SCL после того, как переводят ее в единичное состояние, и только дождавшись реальной установки линии SCL в единичное состояние продолжать операции приема-передачи.

Теперь мы опишем минимально необходимый набор операций для реализации "Master"-абонента. Прежде всего, это подпрограммы выдачи "Start"- и "Stop"-условия, подпрограмма передачи байта, подп85

рограмма приема байта, выдающая сигнал ACK и подпрограмма приема байта, не выдающая сигнал ACK. Три последние используют подпрограммы передачи бита и приема бита. Во временных промежутках между подпрограммами линия SCL имеет

нулевое значение. Как уже говорилось ранее, важно правильно согласовывать все временные характеристики подпрограмм. Если, к примеру, вы вызываете повторный "Start" сразу за передачей (приемом) байта, то необходимо, чтобы полностью завершился нулевой полупериод ( $>4.7$  мкс), предшествующий "Start"-условию, а подпрограмма, реализующая "Start"-условие, переводя линии SDA и SCL в единичное состояние, дождавшись их реального возврата, должна удерживать ее в таком состоянии не менее установленного времени ( $>4.7$  мкс), и так во всех случаях.

Оперируя этими пятью подпрограммами можно легко наладить обмен по шине I<sup>2</sup>C. Можно выдать на шину "Start"-условие, за ним "Slave Adress", далее данные, в соответствии с протоколом адресуемого устройства. Можно выдать повторное "Start"-условие, новый "Slave Adress" и так далее, формируя процесс обмена как из элементов конструктора.