

Как SDA, так и SCL являются двунаправленными линиями, подсоединенными к положительному источнику питания через подтягивающий резистор (см. Рис 4). 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 I²C 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).

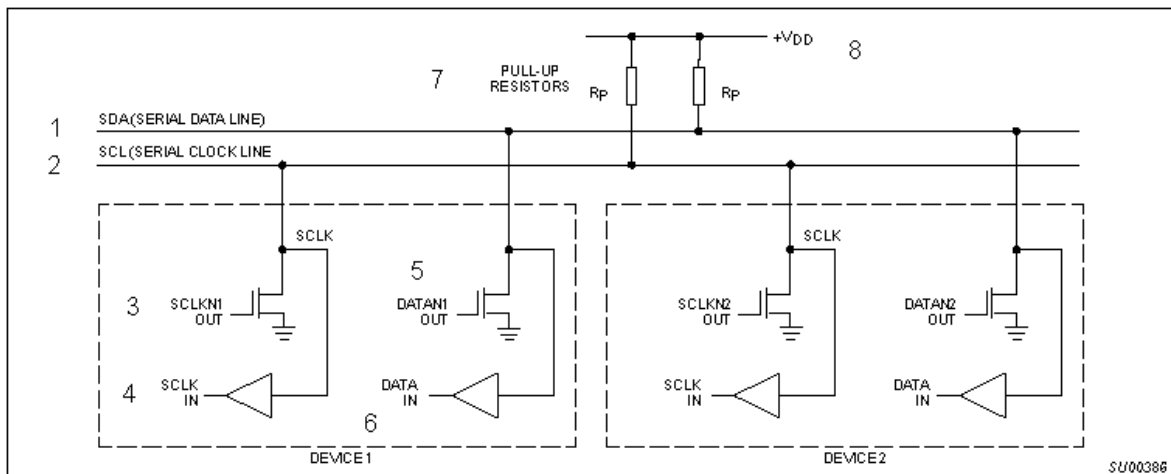


Рисунок 4. Подключение I²C-устройств к шине

1. SDA (линия данных)
2. SCL (линия синхронизации)
3. Выход синхронизации
4. Вход синхронизации
5. Выход данных
6. Вход данных
7. Подтягивающие резисторы
8. Напряжение питания

5.0 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 15.0 для электрических параметров). Один синхроимпульс генерируется на каждый пересылаемый бит.

5.1 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. Рис 5).

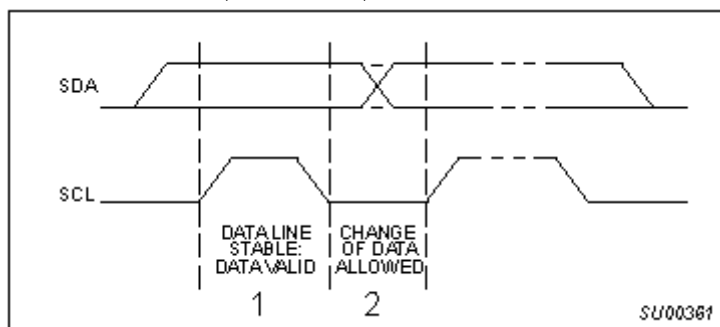


Рисунок 5. Пересылка бита в шине I²C

1. Линия данных находится в стабильном состоянии, данные определены
2. Допускается изменение данных

5.2 Сигналы START и STOP

Специальные ситуации на шине отмечают сигналы START и STOP (см. Рис 6).

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.

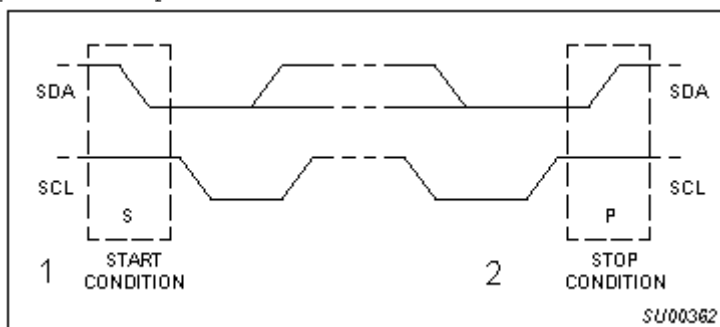


Рисунок 6. Сигналы СТАРТ и СТОП

1. Сигнал СТАРТ
2. Сигнал СТОП

6.1 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. Рис. 7). 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 9.1.3).

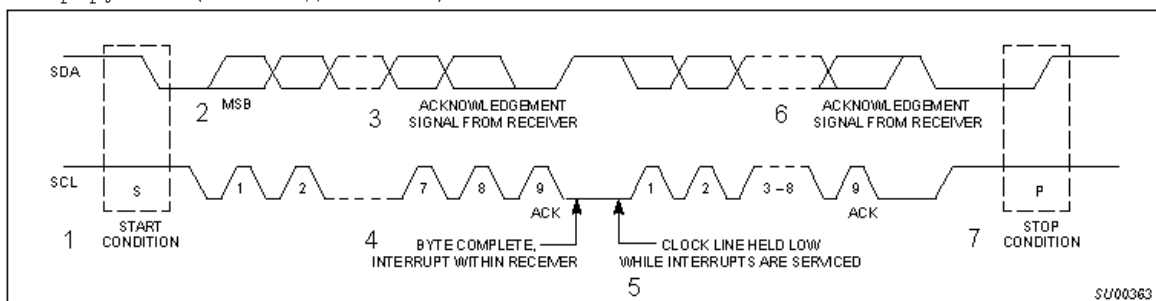


Рисунок 7. Пересылка данных по шине I²C

1. Сигнал СТАРТ
2. Старший разряд байта
3. Сигнал подтверждения от приёмника
4. Прием байта завершен. Прерывание внутри приемника
5. Синхронизация удерживается в низком состоянии, пока обслуживается прерывание
6. Сигнал подтверждения от приемника

7. Сигнал СТОП

6.2 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 (рис. 8). Конечно, время установки и удержания также должны быть приняты во внимание (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. Совместимость с CBUS).

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.

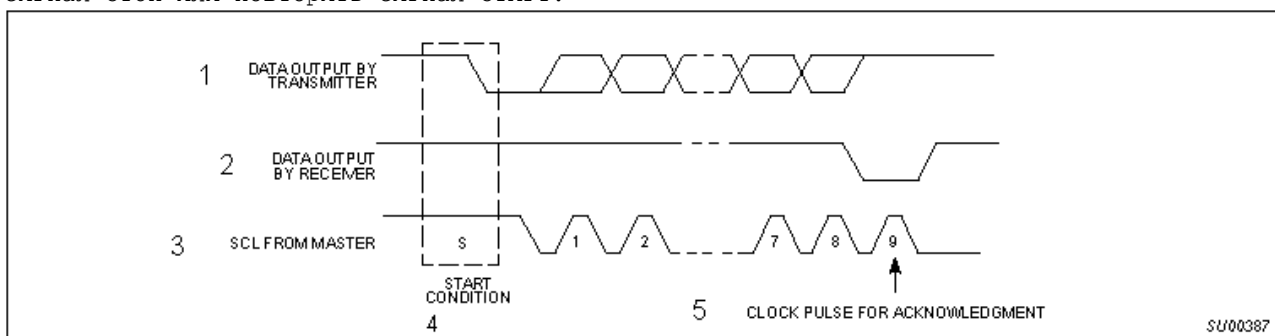


Рисунок 8. Подтверждение

1. Данные, переданные передатчиком
2. Данные, переданные приёмником
3. SCL от ведущего
4. Сигнал СТАРТ
5. Синхроимпульс подтверждения

7.1 Синхронизация

При передаче посылок по шине I²C 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 (рис. 9). 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.

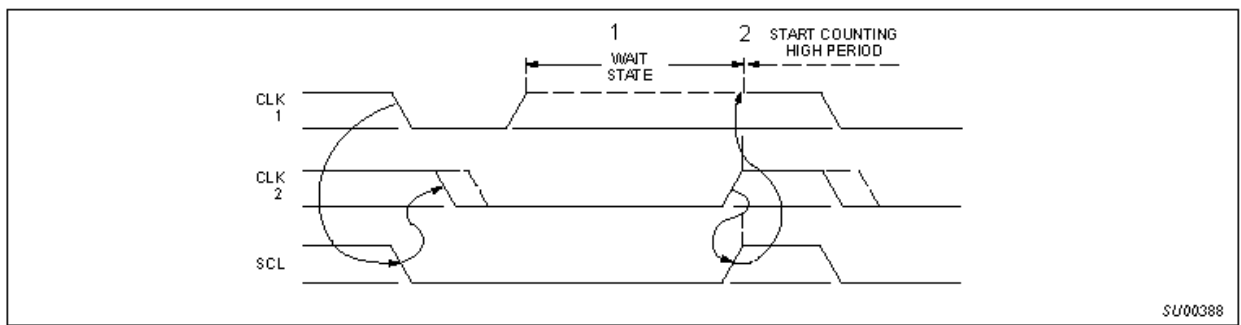


Рисунок 9. Синхронизация во время арбитража

1. Состояние ожидания
2. Начало отсчета ВЫСОКОГО периода синхроимпульса

7.2 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 SDA (НИЗКОЕ) не соответствует ВЫСОКОМУ состоянию его внутренней линии данных.

Арбитраж может продолжаться на протяжении нескольких бит. Так как сначала передается адрес, а потом данные (см. Разделы 7-битная адресация и 10-битная адресация), 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 10 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

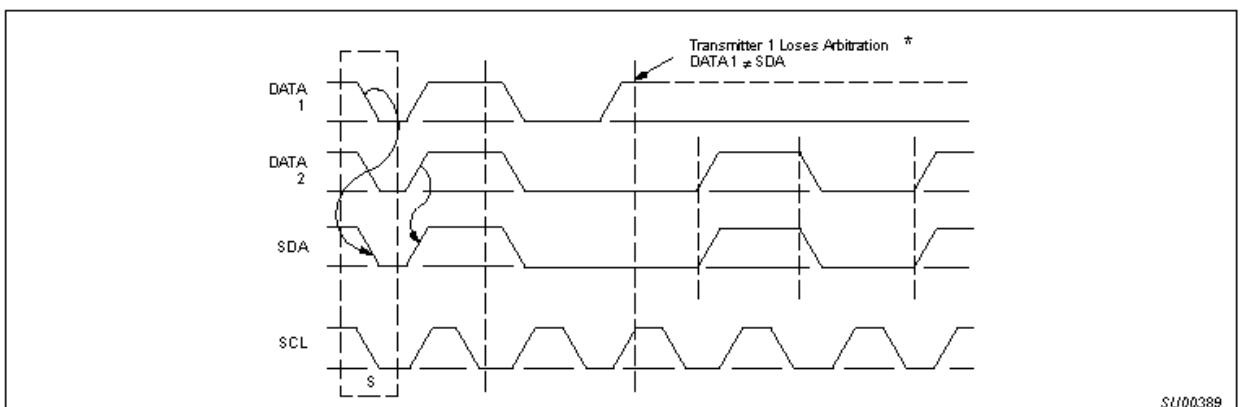


Рисунок 10. Арбитраж между двумя ведущими

* передатчик 1 проигрывает арбитраж – его линия данных не совпадает с SDA

7.3 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 I2C или с ограниченными цепями может замедлить частоту синхроимпульсов путем продления их НИЗКОГО периода. Таким образом скорость передачи любого ведущего адаптируется к скорости медленного устройства.

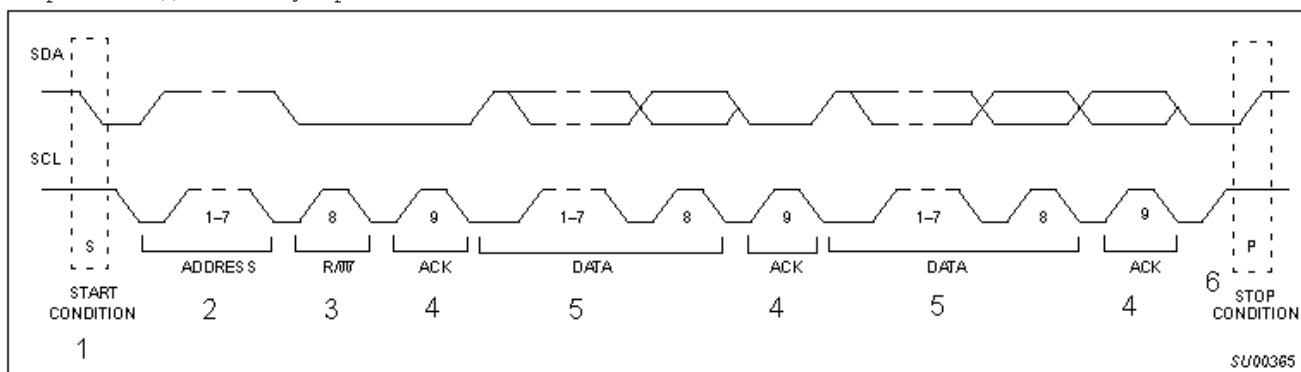


Рисунок 11. Посылка данных

1. Сигнал СТАРТ
2. Адрес
3. Бит направления (R/W[^])
4. Подтверждение
5. Данные
6. Сигнал СТОП

8.0 Форматы с 7-битным адресом

Посылки данных происходят в формате, показанном на рис. 11. 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
- Ведущий читает ведомого немедленно после пересылки первого байта (рис. 13). 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 (рис. 14). 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.

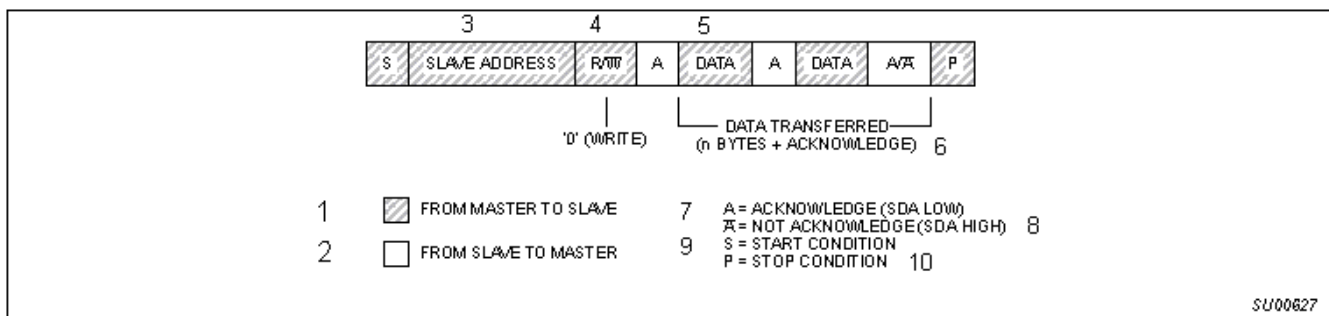


Рисунок 12. Ведущий-передатчик адресует ведомого-приемника 7-битным адресом. Направление пересылки не изменяется

1. От ведущего к ведомому
2. От ведомого к ведущему
3. Адрес ведомого
4. Бит направления
5. Данные
6. Пересылаемые данные (n байт + подтверждение)
7. A - Подтверждение
8. Неподтверждение
9. Сигнал СТАРТ
10. Сигнал СТОП

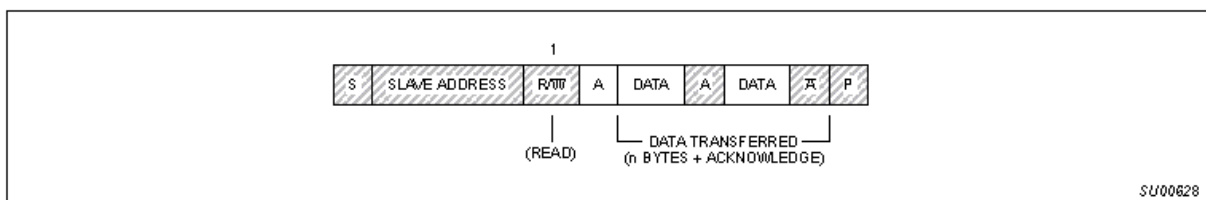


Рисунок 13. Ведущий читает из ведомого непосредственно после первого байта

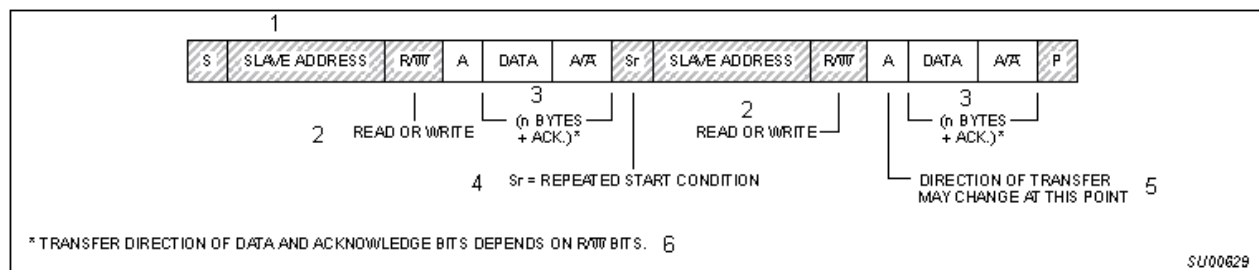


Рисунок 14. Комбинированный формат

1. Адрес ведомого
2. Чтение или запись
3. n байтов + подтверждения
4. Сигнал повторного СТАРТА
5. Направление пересылки может измениться в этой точке
6. Направление пересылки данных и битов подтверждения зависит от битов направления

ПРИМЕЧАНИЯ:

1. 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.
2. Все решения об авто-инкременте или декременте адреса, к которому произошел предыдущий доступ, принимаются конструктором устройства
3. Каждый байт завершается битом подтверждения, обозначенным A или A^ на рисунках

4. I²C-совместимые устройства должны сбрасывать логику шины при получении сигнала СТАРТ или повторный СТАРТ и подготавливаться к приему адреса.

9.0 7-битная адресация

(см. 10-битная адресация).

Процедура адресации на шине I²C 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 посылки общего вызова определяет действие, которое должны произвести устройства. Более подробно процедура описана в разделе Назначение битов первого байта.

9.1 Назначение битов первого байта

Первые семь битов первого байта образуют адрес ведомого (см. Рис 15). 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.

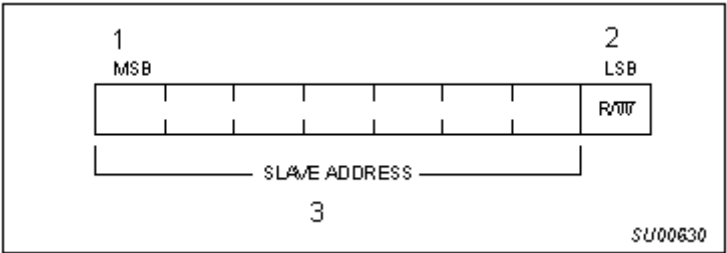


Рисунок 15. Первый байт после сигнала СТАРТ

1. Старший разряд
2. Младший разряд
3. 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 I²C координирует выделение I²C адресов. Дальнейшая информация может быть получена у представителей Philips, указанных на обложке. 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 13.0).

Адрес	R/W бит	Описание
0000000	0	Адрес общего вызова
0000000	1	Байт СТАРТА
0000001	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:

1. Устройствам запрещается подтверждать прием байта СТАРТА
2. Адрес CBUS зарезервирован для того, чтобы можно было использовать CBUS-совместимые и I²C-совместимые устройства в одной системе. I²C-совместимым устройствам запрещается реагировать на прием этого адреса.
3. 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.

9.1.1.1. 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. Есть ведомый не может обработать один из байтов, он не генерирует подтверждение. Значение посылки общего вызова всегда определяется вторым байтом (рис. 16).

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 зашелкивают текущее значение адреса при принятии этой команды. Устройства не сбрасываются
- 00000000. Этот код недопустим для использования в качестве второго байта

Последовательности процедуры программирования опубликованы в соответствующих ТУ.

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 (рис. 17).

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 посылаться данные (рис. 18) После этой процедуры аппаратный ведущий остаётся в режиме ведущий-передатчик.

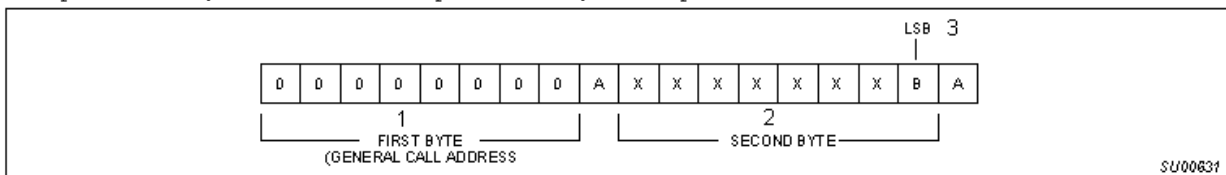


Рисунок 16. Формат адреса общего вызова

1. Первый байт
2. Второй байт
3. Младший разряд

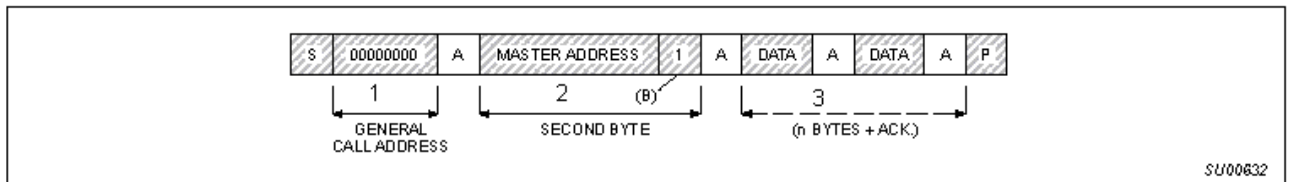


Рисунок 17. Пересылка данных из аппаратного ведущего-передатчика

1. Адрес общего вызова
2. Второй байт
3. N байт + подтверждение

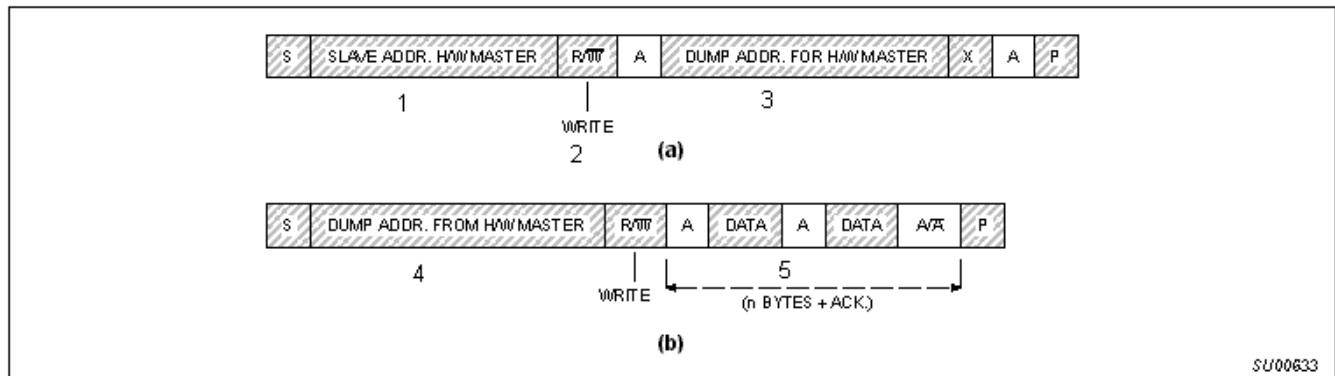


Рисунок 18. Передача данных аппаратным ведущим, способным пересылать данные непосредственно ведомым устройствам

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

1. Адрес аппаратного ведущего
2. Запись
3. Адрес начала данных для аппаратного ведущего
4. Адрес начала данных от аппаратного ведущего
5. n байт + подтверждение

9.1.2 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 (рис. 19). Процедура старта состоит из

- Сигнала СТАРТ
- Байте 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 I²C 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.

9.1.3 Совместимость с CBUS

Приёмники CBUS могут быть подключены к шине I2C. Однако, при этом должна быть введена третья линия DLEN и бит подтверждения должен быть опущен. Обычно послыки I2C состоят из 8-битовых байтов, в то время как CBUS-совместимые устройства обладают другим форматом.

В смешанной шине I²C-совместимые устройства не должны отвечать на послыки формата CBUS. Для этого зарезервирован специальный адрес CBUS (0000001X). После передачи адреса CBUS линия DLEN может быть сделана активной и посылается посылка формата CBUS (рис. 20). 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.

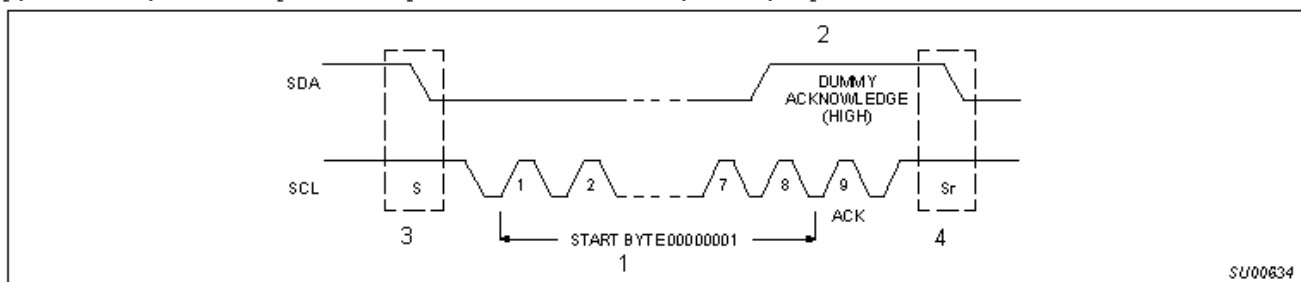


Рисунок 19. Процедура байта СТАРТА

1. Байт СТАРТА
2. Фальшивое подтверждение (ВЫСОКОЕ)
3. Сигнал СТАРТ (S)
4. Сигнал повторный СТАРТ (Sr)

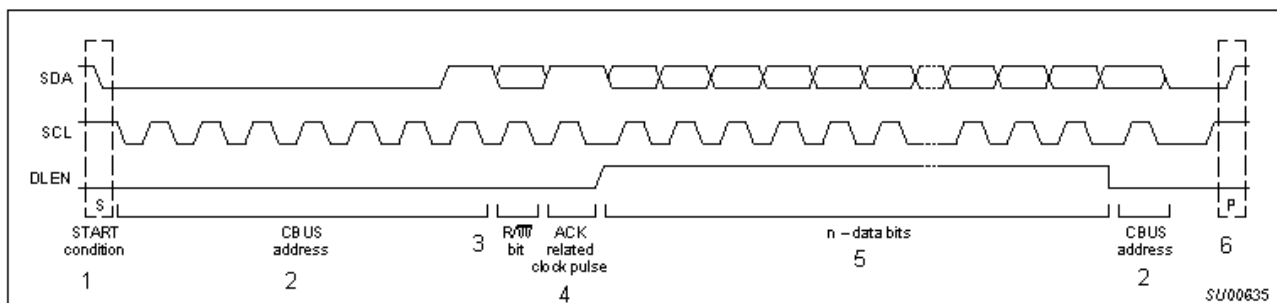


Рисунок 20. Формат данных при посылках с CBUS передатчиком/приемником

1. Сигнал СТАРТ
2. Адрес CBUS
3. Бит направления передачи
4. Синхроимпульс подтверждения
5. n бит данных
6. Сигнал СТОП

10.0 Электрические параметры устройств I2C

Электрические параметры ввода/вывода I2C-совместимых устройств и характеристики подключенных к ним линий шины приведены в таблицах 3 и 4 раздела 15.0

I²C-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% (рис. 21). I2C-устройства с входными уровнями, зависящими от напряжения питания должны иметь одну общую линию питания, к которой также должен быть подключен подтягивающий резистор (рис. 22).

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 рис. 23.

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 рис. 24, последовательно включенные резисторы R_S (например, 300 Ом) могут быть использованы для защиты от высоковольтных выбросов напряжения на линиях шины.

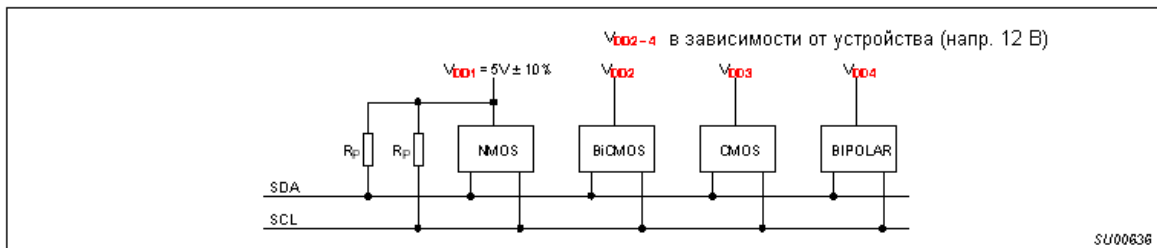


Рисунок 21. Подключение устройств с фиксированным входным уровнем к шине I²C

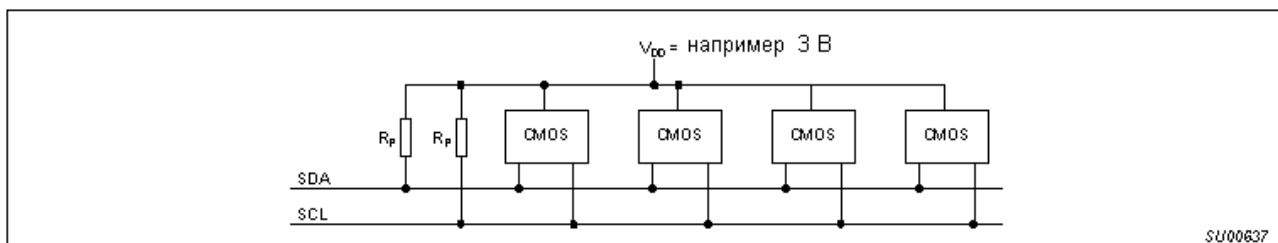


Рисунок 22. Подключение устройств с широким диапазоном питания к шине I²C

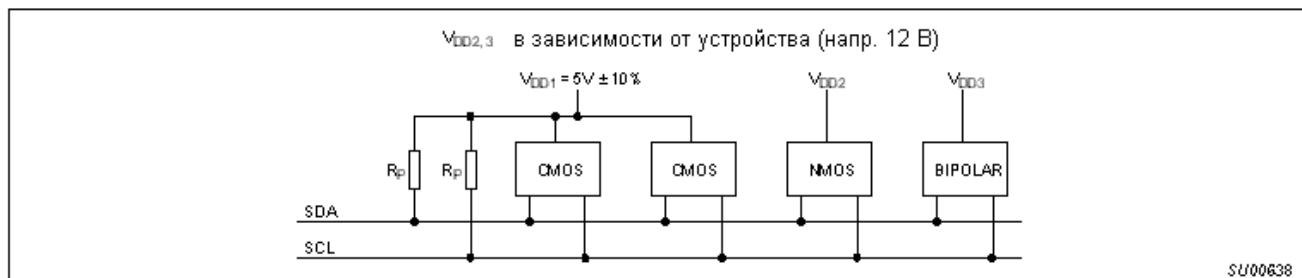


Рисунок 23. Подключение устройств с относительным (V_{DD1}) уровнем входного напряжения и фиксированным входным уровнем (V_{DD2-4}) к шине I²C

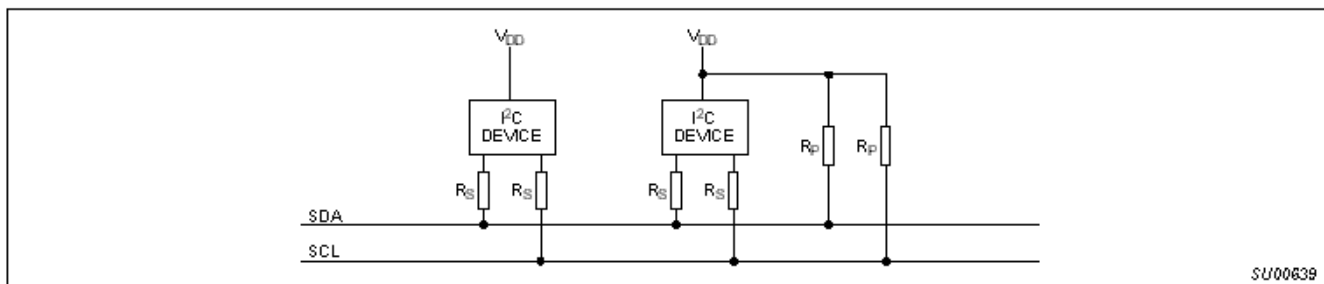


Рисунок 24. Последовательные резисторы R_S для защиты от высоковольтных выбросов

10.1 Максимальные и минимальные значения для резисторов Rp и Rs

Для I²C-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 Volmax=0.4 В. Vdd как функция Rmin показана на рис. 25.

Желаемая шумовая граница 0.1Vdd для НИЗКОГО уровня ограничивает максимальное значение Rs. Rmax как функция от Rp показано на рис. 25. Емкость шины состоит из общей емкости проводов, подключенных портов и контактов. Эта емкость ограничивает максимальное значение Rp вследствие ограничений на время установления (фронта). Рис. 27 показывает Rmax как функцию от емкости шины. Максимальный ток при ВЫСОКОМ уровне для каждого порта не превышает 10 мкА. Вследствие желаемой шумовой границы 0.2Vdd для ВЫСОКОГО уровня, входной ток ограничивает максимальное значение Rp. Этот предел зависит от Vdd. Общий ток при ВЫСОКОМ уровне как функция от Rmax показан на рис. 28.

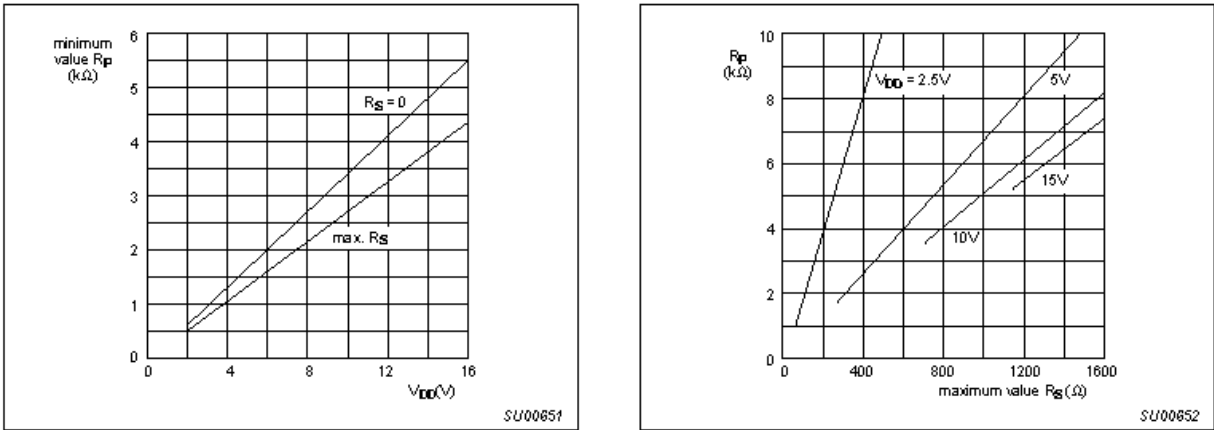


Рисунок 25. Минимальное значение Rp как функция от напряжения питания при параметре Rs

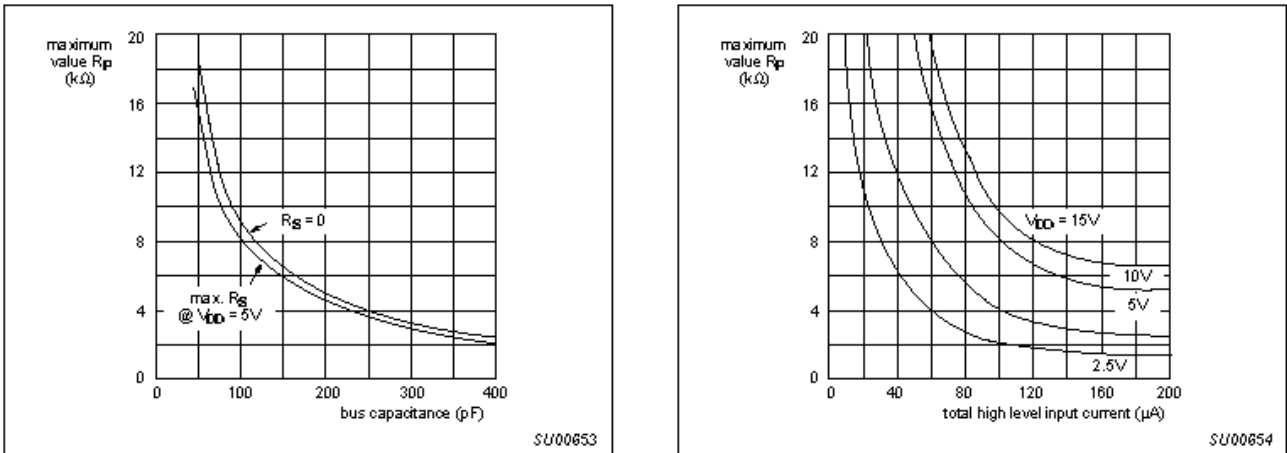


Рисунок 26. Максимальное значение Rs как функция от Rp при параметре - Vdd

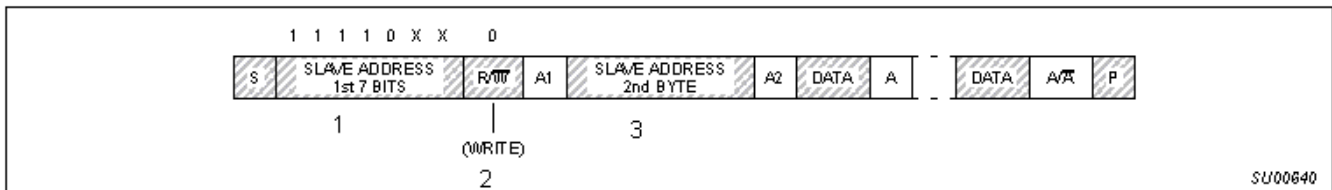


Рисунок 27. Максимальное значение Rp как функция от емкости шины для стандартного режима

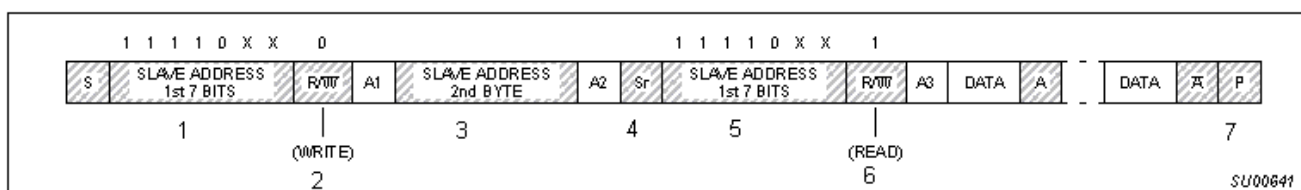


Рисунок 28. Общий ток при напряжении ВЫСОКОГО уровня как функция максимального значения Rp при параметре - Vdd

11.0 Дополнения к спецификации шины I²C

Шина I²C 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 I²C 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.

Все новые устройства с I²C интерфейсом работают в быстром режиме. Предпочтительно, они должны уметь принимать и/или передавать данные на скорости 400 кбит/с. Как минимум они 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. Как существующие, так и будущие ведущие смогут генерировать и 7-битные, и 10-битные адреса.

12.0 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, показанная на рис. 37

13.0 10-битная адресация

10-битная адресация не меняет формат шины. Для этого используется зарезервированная адресная комбинация 1111XXX первых семи бит первого байта (см. **Раздел 9.1**). 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.

13.1 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.

13.2 10-битные форматы

Возможны различные комбинации форматов 10-битных посылок:

- Ведущий-передатчик передает информацию ведомому-приемнику (рис 29). 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 pošлет сигнал СТОП или сигнал повторного СТАРТА с другим адресом.
- Ведущий-приемник принимает данные от ведомого-передатчика. Направление передачи меняется после второго бита направления (рис. 30). 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 (рис 31). 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 (рис. 32). Один ведущий занимает шину на все время пересылки
- Комбинированный формат. 10-Битная и 7-битная адресация применяются в одной посылке (рис. 33). После каждого сигнала СТАРТ или повторный СТАРТ передается

10-битный или 7-битный адрес. Рисунок 33 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.

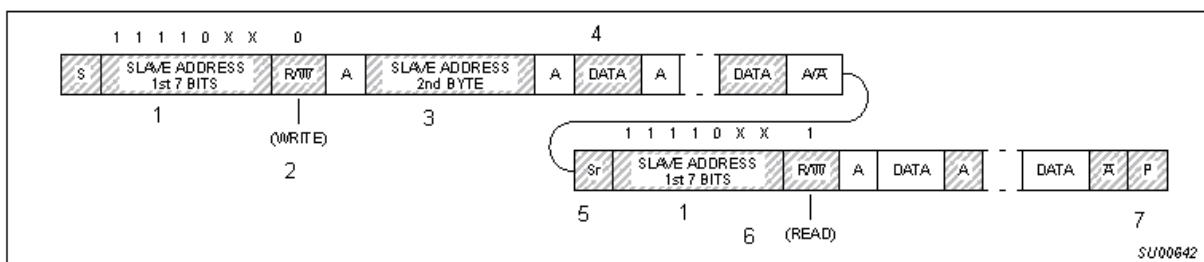


Рисунок 29. Ведущий-передатчик адресует ведомого-приемника 10-битным адресом

1. Адрес ведомого (первые 7 бит)
2. Запись
3. Адрес ведомого (второй байт)

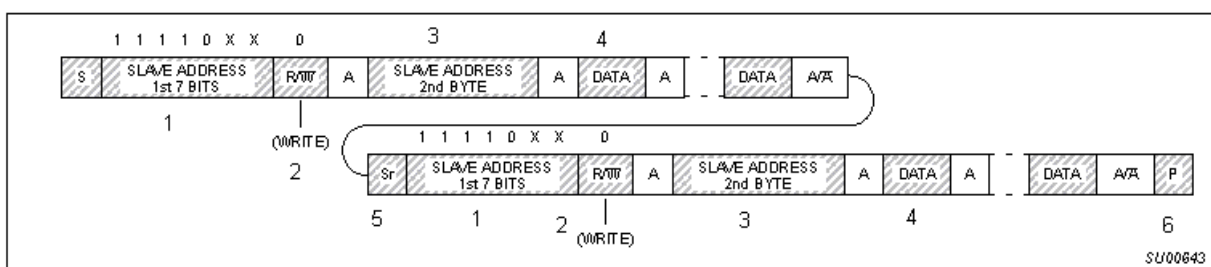


Рисунок 30. Ведущий-приемник адресует ведомого-передатчика 10-битным адресом

1. Адрес ведомого (первые 7 бит)
2. Запись
3. Адрес ведомого (второй байт)
4. Сигнал повторный СТАРТ
5. Адрес ведомого (первые 7 бит)
6. Чтение
7. Сигнал СТОП

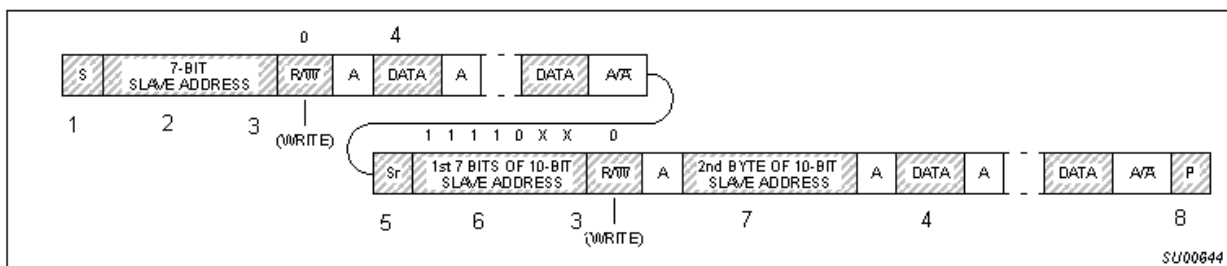


Рисунок 31. Комбинированный формат. Ведущий адресует ведомого 10-битным адресом, потом передает ему данные и принимает из него данные.

1. Адрес ведомого (первые 7 бит)
2. Запись
3. Адрес ведомого (второй байт)
4. Данные
5. Сигнал повторный СТАРТ
6. Чтение
7. Сигнал СТОП

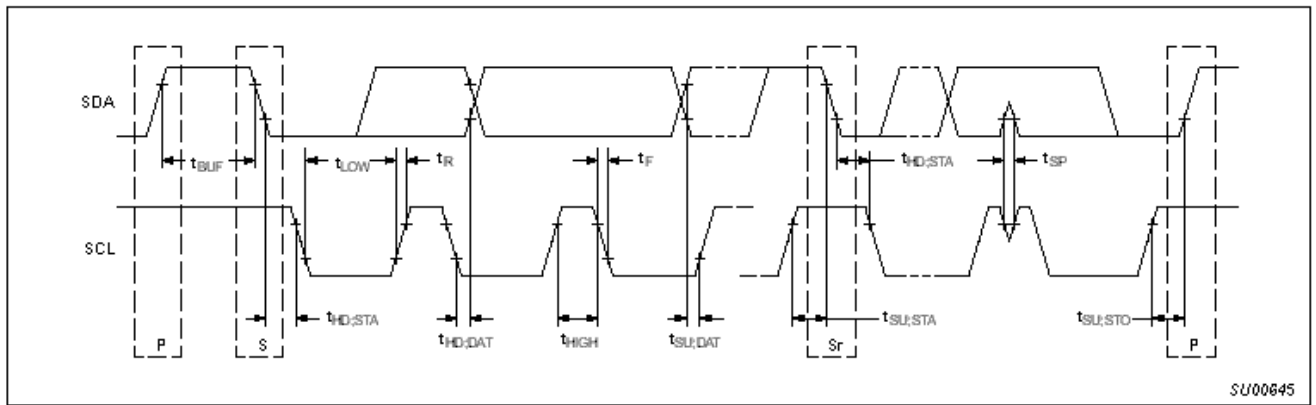


Рисунок 32. Комбинированный формат. Ведущий передает данные двум ведомым, оба 10-битные

1. Адрес ведомого (первые 7 бит)
2. Запись
3. Адрес ведомого (второй байт)
4. Данные
5. Сигнал повторный СТАРТ
6. Сигнал СТОП

