

## **Модуль MSSP в режиме I2C интерфейса для микроконтроллеров PIC16F87X**

Статья основывается на технической документации DS30292b  
компании Microchip Technology Incorporated, USA.

**© ООО «Микро-Чип»  
Москва - 2001**

Распространяется бесплатно.  
Полное или частичное воспроизведение материала допускается только с письменного разрешения  
ООО «Микро-Чип»  
тел. (095) 737-7545  
[www.microchip.ru](http://www.microchip.ru)

## Модуль MSSP в режиме I2C интерфейса для микроконтроллеров PIC16F87X

Статья основывается на технической документации DS30292b  
компании Microchip Technology Incorporated, USA.

В статье описывается модуль MSSP (режим I2C интерфейса) микроконтроллеров:  
PIC16F873; PIC16F874; PIC16F876; PIC16F877.

Модуль MSSP полностью поддерживает все функции Master и Slave устройств, включая поддержку общего вызова, и обеспечивает аппаратно прерывания по командам START, STOP на шине I2C, для определения занятости шины в режиме Master и при конкуренции на шине. В MSSP модуле реализована поддержка стандартного режима с 7, 10 –битной адресацией.

Для работы с шиной I2C использует два вывода SCL и SDA (RC3/SCK/SCL – сигнал синхронизации; RC4/SDI/SDA – данные). Выводы совмещены с каналами PORTC и автоматически подключаются к MSSP при его включении, установкой бита SSPEN в регистре SSPCON.

Управление модулем MSSP в режиме I2C используются 6 регистров:

SSPSTAT – регистр статуса  
SSPCON – регистр управления 1  
SSPCON2 – регистр управления 2  
SSPBUF – буфер приема/передачи  
SSPADD – регистр адреса  
SSPSR – сдвиговый регистр (пользователю не доступен)

### SSPSTAT (адрес: 94h): регистр статуса

Позволяет контролировать состояние шины I2C, определять тип принятых данных и состояние буфера приема/передачи.

R/W - 0	R/W - 0	R - 0	R - 0	R - 0	R - 0	R - 0	R - 0
<b>SMP</b>	<b>CKE</b>	<b>D/A</b>	<b>P</b>	<b>S</b>	<b>R/W</b>	<b>UA</b>	<b>BF</b>
бит7							бит0

бит 7: **SMP**: управление длительностью фронта  
1 = в стандартном режиме (100КГц и 1МГц)  
0 = в скоростном режиме (400КГц)

бит 6: **CKE**: спецификация входных уровней  
1 = SMBUS спецификация  
0 = I2C спецификация

бит 5: **DA**: тип информации данные/адрес  
1 = последний принятый или переданный байт является информационным  
0 = последний принятый или переданный байт является адресным

бит 4: **P**: флаг выполнения команды STOP на шине I2C  
1 = команда STOP обнаружена  
0 = команда STOP не обнаружена

бит 3: **S**: флаг выполнения команды START на шине I2C  
1 = команда START обнаружена  
0 = команда START не обнаружена

бит 2: **R/W**: бит чтения/записи. Действителен после совпадения адреса и до следующей команды START, STOP или бита ACK  
1 = чтение  
0 = записи

бит 1: **UA**: флаг обновление адреса устройства (только для 10-битного режима адресации)  
1 = требуется обновить адрес в регистре SSPADD  
0 = обновление не требуется

бит 0: **BF**: статус буфера приема/передачи SSPBUF  
*Прием данных*  
1 = прием байта завершен, буфер полон  
0 = данные еще не приняты, буфер пуст

*Передача данных*  
1 = Передача не завершена, буфер полон  
0 = Передача байта завершена, буфер пуст

**SSPCON (адрес: 14h):** регистр управления 1

Управляет режимом работы модуля MSSP, позволяет контролировать ошибки при работе с буфером приема/передачи данных SSPBUF.

R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0
<b>WCOL</b>	<b>SSPOV</b>	<b>SSPEN</b>	<b>CKP</b>	<b>SSPM3</b>	<b>SSPM2</b>	<b>SSPM1</b>	<b>SSPM0</b>
бит7							бит0

- бит 7: **WCOL**: флаг обнаружения коллизий передачи  
1 = регистр SSPBUF обновлен, в момент передачи предыдущего байта или выдачи управляющих битов на шину (должен быть сброшен программно)  
0 = коллизий не обнаружено
- бит 6: **SSPOV**: флаг переполнения приемного буфера  
1 = обнаружено переполнение буфера. Записан новый принятый байт в SSPBUF, при непрочитанном предыдущем байте (должен быть сброшен программно)  
0 = переполнения нет
- бит 5: **SSPEN**: включение модуля MSSP  
1 = модуль MSSP включен, выходы RC4/SDA, RC3/SCL используются синхронным последовательным портом  
0 = модуль MSSP выключен, линии RC4/SDA, RC3/SCL используются как порты ввода/вывода
- бит 4: **CKP**: удержание линии синхронизации  
1 = разрешить синхронизацию  
0 = удерживать сигнал синхронизации в низком уровне
- бит 3-0: **SSPM3-SSPM0**: режим работы модуля MSSP  
0000 – 0101 = используется для работы модуля в SPI режиме  
0110 = Slave режим I2C, 7-битная адресация  
0111 = Slave режим I2C, 10-битная адресация  
1000 = Master режим I2C, частота тактового сигнала =  $F_{osc} / (4 * (SSPAD + 1))$   
1011 = программная реализация «Master» режима  
1110 = Slave режим I2C, 7-битная адресация, разрешены прерывания по командам START, STOP на шине  
1111 = Slave режим I2C, 10-битная адресация, разрешены прерывания по командам START, STOP на шине

**SSPCON2 (адрес: 91h):** регистр управления 2

Управляет режимом работы модуля MSSP в Master режиме I2C

R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0
<b>GCEN</b>	<b>ACKSTAT</b>	<b>ACKDT</b>	<b>ACKEN</b>	<b>RCEN</b>	<b>PEN</b>	<b>RSEN</b>	<b>SEN</b>
бит7							бит0

- бит 7: **GCEN**: разрешение общего вызова в Slave режиме (адрес общего вызова 0000h)  
1 = разрешить прерывания при обнаружении адреса общего вызова  
0 = общий вызов запрещен
- бит 6: **ACKSTAT**: подтверждение последнего переданного байта в Master режиме  
1 = подтверждение не было получено  
0 = подтверждение было получено
- бит 5: **ACKDT**: данные для подтверждения принятого байта в режиме Master  
1 = не выдавать подтверждение  
0 = подтверждение выдавать
- бит 4: **ACKEN**: бит включения подтверждения в режиме Master, автоматически сбрасывается в нуль  
1 = инициировать подтверждение на линиях SCL и SDA, выдать на SDA содержимое бита ACKDT  
0 = подтверждение не формируется
- бит 3: **RCEN**: включение повторного приема в Master режиме  
1 = включить прием  
0 = прием выключен
- бит 2: **PEN**: бит включения команды STOP в режиме Master, автоматически сбрасывается в нуль  
1 = инициировать STOP на линиях SCL и SDA  
0 = STOP не выдается
- бит 1: **RSEN**: бит включения команды повторный START в режиме Master, автоматически сбрасывается в нуль  
1 = инициировать повторный START на линиях SCL и SDA  
0 = повторный START не выдается
- бит 0: **SEN**: бит включения команды START в режиме Master, автоматически сбрасывается в нуль  
1 = инициировать START на линиях SCL и SDA  
0 = START не выдается

В регистре SSPBUF сохраняются принятые данные с шины I2C. Совместное использование регистров SSPBUF и SSPSR позволяет создать двух уровневый буфер FIFO приемника. Когда получен байт, он переписывается из регистра SSPSR в SSPBUF, устанавливается флаг прерывания SSPIF. Если получен новый байт, до того как будет прочитан предыдущий из регистра SSPBUF, устанавливается флаг переполнения приемного буфера SSPOV. Байт из регистра SSPSR будет потерян.

В регистре SSPADD сохраняется Slave адрес устройства. При режиме 10-битной адресации устройств, в первую очередь должен быть загружен старший байт адреса (1111 0 A9 A8 0). После совпадения старшей части адреса, в регистр SSPADD необходимо записать младшую часть адреса (A7:A0).

## Slave режим

Структурная схема модуля MSSP в Slave режиме I2C

В Slave режиме, выводы SCL и SDA, должны быть сконфигурированы как входы (биты 3,4 регистра TRISC должны быть установлены). Модуль MSSP будет сам управлять направлением этих портов ввода/вывода.

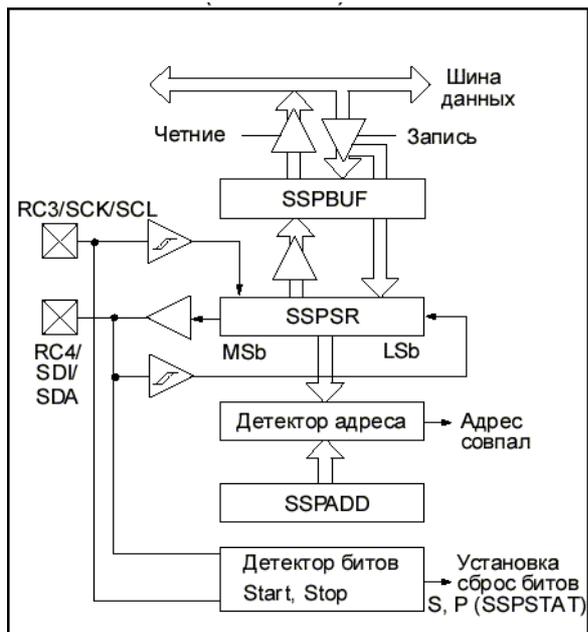
Когда принятый адрес совпадает с адресом устройства, или приняты данные после совпадения адреса, аппаратно формируется бит подтверждения ACK, и принятые данные из регистра SSPSR переписываются в SSPBUF.

Существует два условия, при которых бит ACK не формируется:

- установлен бит BF (буфер полон), если был установлен до того как данные были получены
- установлен бит SSPOV (переполнение буфера), если был установлен до того как данные были получены.

При этом данные в SSPSR будут потеряны, флаг SSPIF установлен.

В таблице показан пример работы прием данных с шины I2C при различных состояниях битов статуса приемника. Бит BF очищается аппаратно при чтении из регистра SSPBUF, а бит SSPOV жен быть очищен программно. В затененных ячейках показан случай, когда во время не был очищен бит SSPOV.



Биты статуса приемника		Запись из SSPSR в SSPBUF	Импульс ACK	Флаг прерывания SSPIF
BF	SSPOV			
0	0	Есть	Есть	Есть
1	0	Нет	Нет	Есть
1	1	Нет	Нет	Есть
0	1	Нет	Нет	Есть

## Адресация

После получения команды START, модуль SSP принимает 8 бит с шины I2C в регистр SSPSR, по переднему фронту сигнала SCL, и сравнивает их с содержимым регистра SSPADD. Сравнение производится на заднем фронте сигнала SCL. Если принятый байт, совпадает с записанным в SSPADD, и биты BF, SSPOV сброшены, выполняются следующие действия:

- значение регистра SSPSR загружается SSPBUF
- устанавливается бит BF
- формируется импульс ACK
- устанавливается флаг прерывания SSPIF по заднему фронту девятого импульса SCL, и формирует прерывание, если оно разрешено

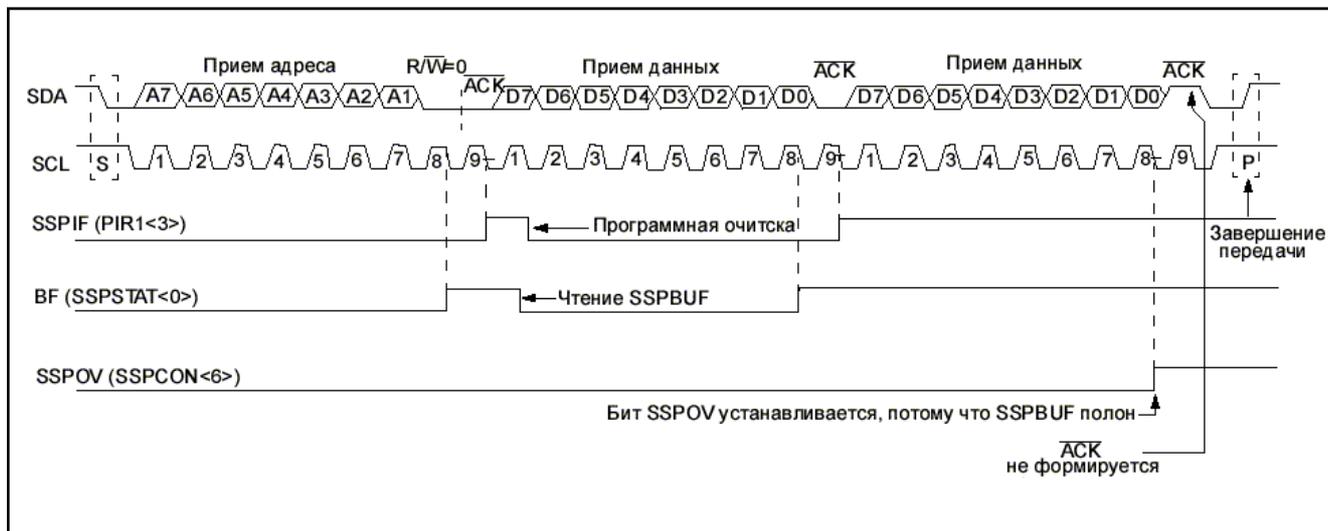
Для устройств с 10-битной адресацией, необходимо выполнить следующую последовательность действий для проверки адреса:

- получить старший байт адреса (устанавливаются биты SSPIF, BF и UA)
- записать младший байт адреса в регистр SSPADD (аппаратно сбрасывается бит UA)
- прочитать регистр SSPBUF (для сброса бита BF), сбросить флаг прерывания SSPIF
- записать старший байт адреса в регистр SSPADD (аппаратно сбрасывается бит UA, отпускает линию SCL)
- прочитать регистр SSPBUF (для сброса бита BF), сбросить флаг прерывания SSPIF
- принять команду повторный START
- получить старший байт адреса (устанавливаются биты SSPIF, BF)
- прочитать регистр SSPBUF (для сброса бита BF), сбросить флаг прерывания SSPIF

### Прием данных

Когда бит  $R/W$  равен нулю, в адресном байте, и принятый адрес совпадает с адресом устройства, тогда бит  $R/W$  в регистре  $SSPSTAT$  очищается. Принятый адрес загружается в регистр  $SSPBUF$ . Если, установлены биты  $BF$  или  $SSPOV$ , то бит подтверждения  $ACK$  не формируется.

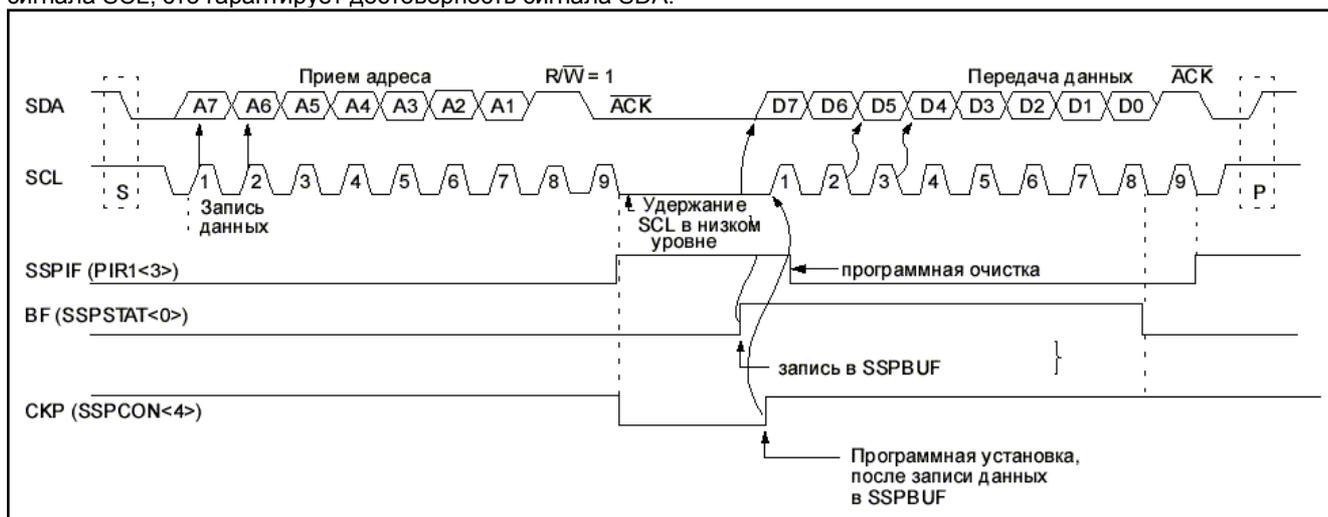
Прерывание от модуля  $SSP$  формируется при каждом принятом байте с шины  $I2C$ , установкой флага  $SSPIF$  в регистре  $PIR1$ . Флаг прерывания необходимо программно сбросить. Регистр  $SSPSTAT$  используется для определения типа принятого бита.



### Передача данных

Когда  $R/W$  бит входящего адресного бита установлен, и адрес совпал с записанным в регистре  $SSPADD$ , устанавливается бит  $R/W$  в регистре  $SSPSTAT$ . Полученный адресный байт записывается в регистр  $SSPBUF$ . Импульс подтверждения  $ACK$  формируется на 9 импульсе тактового сигнала, после чего, линия  $RC3/SCK/SCL$  удерживается в низком уровне.

Данные, которые необходимо передать, записываются в регистр  $SSPBUF$ . Установкой бита  $CKP$  регистра  $SSPCON$  «отпускается» линия  $SCL$ , активизируя передачу данных. Master шины  $I2C$  должен контролировать состояние линии  $SCL$  при приеме данных. Прием данных «Master» устройством осуществляется по заднему фронту сигнала  $SCL$ , это гарантирует достоверность сигнала  $SDA$ .



Модуль  $MSSP$  формирует прерывания по каждому переданному байту, установкой бита  $SSPIF$  по заднему фронту девятого такта сигнала  $SCL$ . Флаг  $SSPIF$  должен быть сброшен программно. Регистр  $SSPSTAT$  используется для определения статуса передачи данных.

Master устройство, формирует бит подтверждения  $ACK$ , на девятом такте сигнала  $SCL$ , для каждого принятого бита. Если бит подтверждения  $ACK$  не сформирован (высокий уровень сигнала  $SDA$ ), передача данных завершена. Логика Slave устройства настраивается на прием адресного бита.

Если бит подтверждения  $ACK$  был получен (низкий уровень сигнала  $SDA$ ), в регистр  $SSPBUF$  записывается новый байт для передачи. Линию  $SCL$  необходимо «отпустить» установкой бита  $CKP$  в регистре  $SSPCON$ , для начала передачи данных.

**Поддержка адреса общего вызова**

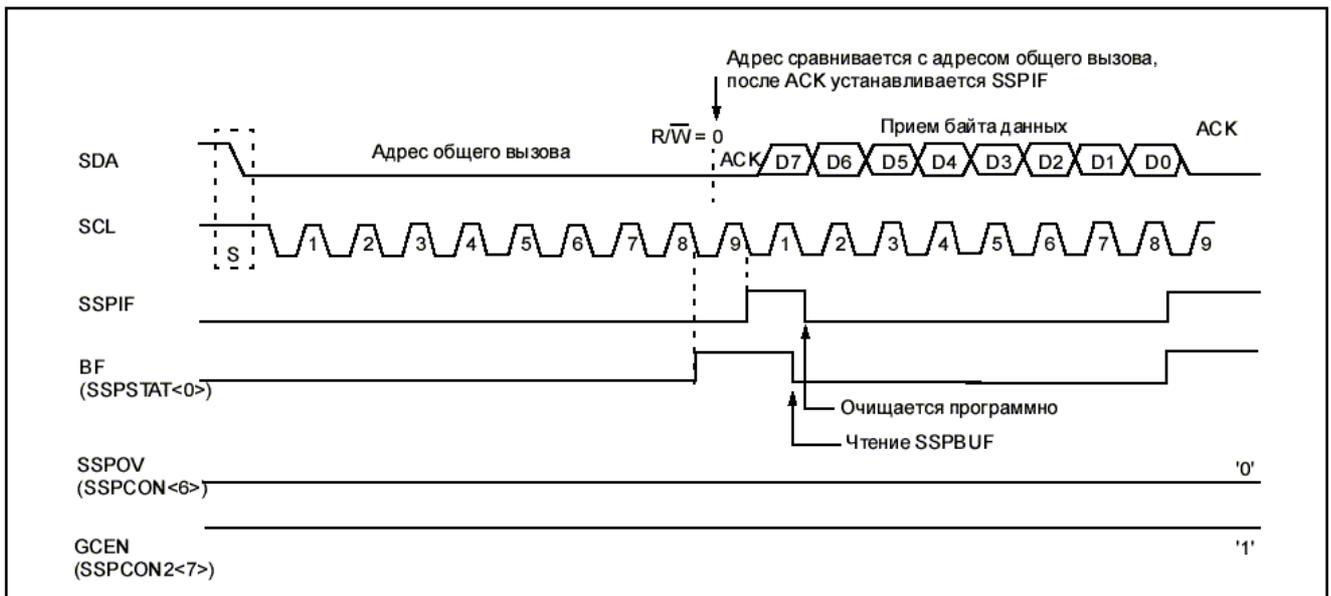
Процедура адресации в I2C такова, что первый после START байт определяет, к которому из Slave устройств обращается Master. Исключением является адрес общего вызова, при использовании которого теоретически должны откликнуться все ведомые.

Адрес общего вызова – один из восьми зарезервированных адресов шины I2C, все биты которого нуль (в том числе и бит R/W).

Распознавание адреса общего вызова включается установкой бита GCEN в регистре SSPCON2. Следующий за START байт помещается в регистр SSPSR и сравнивается с содержимым SSPADD и фиксированным адресом общего вызова.

При совпадении адреса общего вызова, содержимое SSPSR переписывается в SSPBUF, по заднему фронту восьмого такта с установкой бита BF. При обработке прерывания содержимое SSPBUF может быть использовано для адреса обращения.

В режиме 10-битной адресации устройств, требуется обновление регистра SSPADD после совпадения старшего байта адреса, но при общем вызове младшая половина адреса отсутствует, бит UA не устанавливается, и Slave начинает прием данных сразу после подтверждения.





### Работа в режиме Master

Master формирует на шине I2C тактовый сигнал и команды START, STOP. Текущий обмен данными завершается после формирования команд STOP и повторный START. Поскольку команда повторный START инициирует новый обмен данными, шина остается занятой.

Передатчик Master выдает данные на линию SDA, а тактовый сигнал на линию SCL. Первый передаваемый байт содержит 7-битный адрес приемника (при 7-битной адресации устройства) и бит направления данных R/W = 0. После каждого переданного байта принимается бит подтверждения ACK.

Master приемник, выдает на линию SDA байт, содержащий 7-битный адрес передатчика (при 7-битной адресации устройства) и бит направления данных R/W = 1. На линии SCL Master формирует тактовый сигнал. Данные принимаются с линии SDA, после каждого принятого байта формируется бит подтверждения.

Генератор скорости обмена BRG используется для установки требуемой частоты тактового сигнала на линии SCL – 100КГц, 400КГц или 1МГц. Значение для перезагрузки BRG берется из 7 младших бит регистра SSPADD. BRG начинает работу сразу после инициализации какого либо события – START, STOP, передачи записью в SSPBUF, приема и т.д. По завершению события, тактовый сигнал остается в текущем состоянии до инициации следующего.

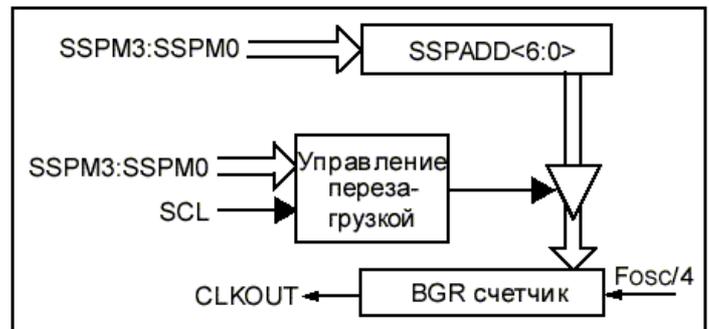
Рекомендованная последовательность действий при передаче данных

1. Инициировать START установкой бита SEN в регистре SSPCON2
2. Ожидать прерывания (если оно разрешено) или установки бита SSPIF после завершения выполнения START
3. Записью в SSPBUF инициируется передача адреса
4. 7 бит адреса (при 7-битной адресации) и бит направления данных выданы на SDA
5. Принять подтверждение ACK от приемника, результат записывается в бит ACKSTAT регистра SSPCON2
6. По заднему фронту девятого такта устанавливается бит SSPIF
7. Записью в SSPBUF инициируется передача данных
8. 8 бит данных выдаются на SDA
9. Принять подтверждение ACK от приемника, результат записывается в бит ACKSTAT регистра SSPCON2
10. По заднему фронту девятого такта устанавливается бит SSPIF
11. Инициировать STOP установкой бита PEN в регистре SSPCON2
12. Ожидать прерывания (если оно разрешено) или установки бита SSPIF после завершения выполнения STOP

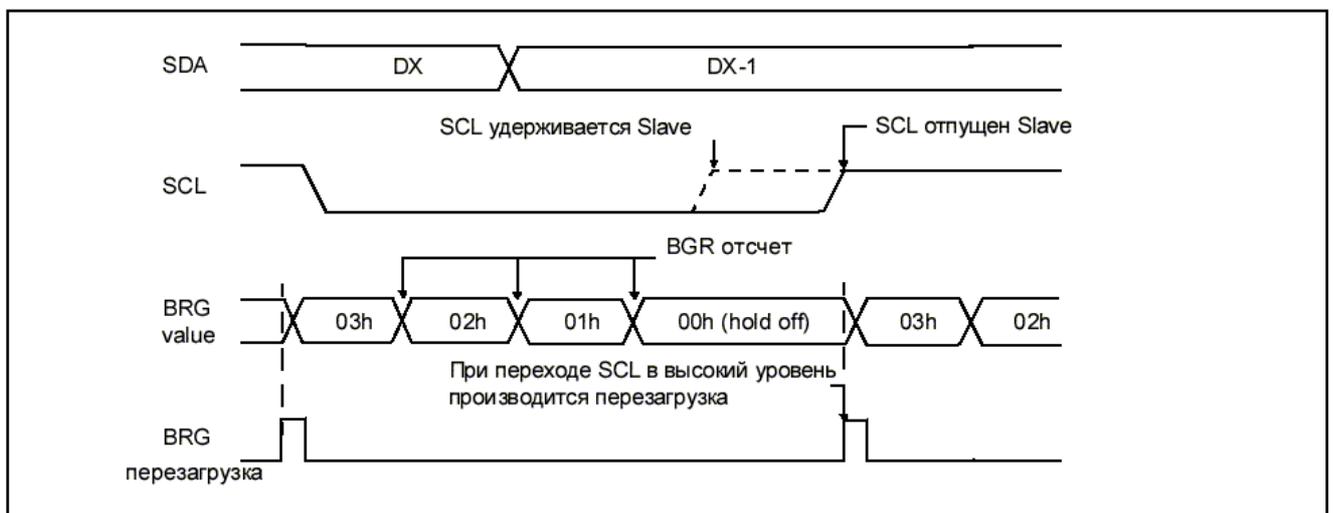
### Генератор скорости обмена BRG

В Master режиме, значение для перезагрузки BRG берется из младших 7 бит регистра SSPADD. После загрузки SSPADD в BRG, счетчик BRG считает декрементируя до нуля и останавливается до следующей перезагрузки, которая не всегда производится автоматически. Если после окончания счета, сигнал на линии SCL должен перейти в высокий уровень, перезагрузка производится только после этого перехода.

Структурная схема генератора скорости обмена BRG



Временная диаграмма работы BRG с арбитражем SCL



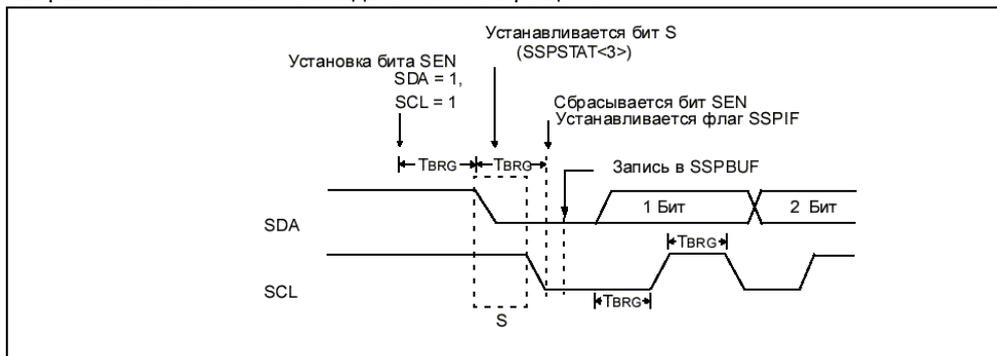
### Временная диаграмма команды START

Чтобы инициировать команду START на шине I2C, необходимо установить бит SEN в регистре SSPCON2. Если на линиях SCL и SDA высокий уровень, BRG загружается значением из регистра SSPADD <6:0>. Если по окончании отсчета времени TBRG сохраняется высокий уровень на SCL и SDA, сигнал SDA переводится в низкий уровень. Перевод SDA в низкий уровень, в то время когда на линии SCL высокий, и есть команда START на шине I2C. После выполнения команды START устанавливается бит S и флаг прерывания SSPIF, BRG загружается новым значением и начинает счет. По окончании счета бит SEN автоматически сбрасывается, генератор останавливается, на SDA остается низкий уровень. Выполнение команды START завершено.

**Примечание:** Если в начале выполнения команды START на SDA или SCL присутствует низкий уровень или во время выполнения START низкий уровень на SCL появляется раньше, чем на SDA, устанавливается флаг прерывания BCLIF (конфликт шины), выполнение START прекращается, MSSP переходит в состояние ожидания.

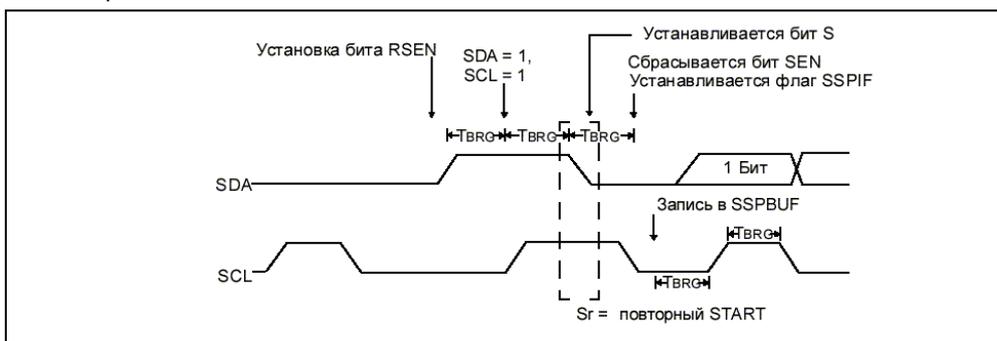
Если во время выполнения команды START производится попытка записи в SSPBUF, устанавливается бит WCOL, а запись не происходит.

**Примечание:** поскольку MSSP не имеет стека событий, установка любого из младших 5 битов регистра SSPCON2 до завершения выполнения команды START запрещено.



### Временная диаграмма команды повторный START

Чтобы инициировать команду повторный START, необходимо установить бит RSEN в регистре SSPCON2 в то время, когда MSSP находится в режиме ожидания. При включении команды повторный START линия SCL переводится в низкий уровень. Когда на SCL устанавливается низкий уровень сигнала, BRG перезагружается содержимым регистра SSPADD<6:0> и начинает отсчет, при этом SDA отпускается в высокий уровень. Если по окончании счета BRG, на линии SDA сохраняется высокий уровень, SCL так же отпускается. BRG вновь перезагружается и начинает отсчет. Если по окончании отсчета времени TBRG сохраняется высокий уровень на SCL и SDA, сигнал SDA переводится в низкий уровень. После выполнения команды повторный START устанавливается бит S и флаг прерывания SSPIF, BRG загружается новым значением и начинает счет. По окончании счета бит RSEN автоматически сбрасывается, генератор останавливается, на SDA остается низкий уровень. Выполнение команды повторный START завершено.



**Примечание:** Установка бита RSEN не приведет к каким-либо действиям

**Примечание:** Если на SDA низкий уровень при переходе SCL из низкого уровня в высокий или низкий уровень на SCL появляется раньше, чем на SDA, устанавливается флаг прерывания BCLIF (конфликт шины), выполнение команды повторный START прекращается, MSSP переходит в состояние ожидания.

Сразу после установки бита SSPIF пользователь может заполнить SSPBUF 7-битным адресом или старшим байтом 10-битного адреса. По завершению передачи 8 бит и получения подтверждения ACK, можно передать байт данных или младший байт адреса.

После записи в SSPBUF каждый бит данных будет выдаваться на SDA по заднему фронту сигнала SCL. По заднему фронту 8-го такта Master отпустит SDA, чтобы Slave мог передать бит подтверждения ACK. По заднему фронту 9-го такта Master опрашивает SDA, чтобы проверить принял ли Slave байт. Значение бита ACK записывается в ACKSTAT регистра SSPCON2. В этот же момент устанавливается флаг SSPIF, а BGR отключается до следующей операции на шине, оставляя низкий уровень на SCL и отпуская SDA.

Если во время выполнения команды повторный START производится попытка записи в SSPBUF, устанавливается бит WCOL, а запись не происходит.

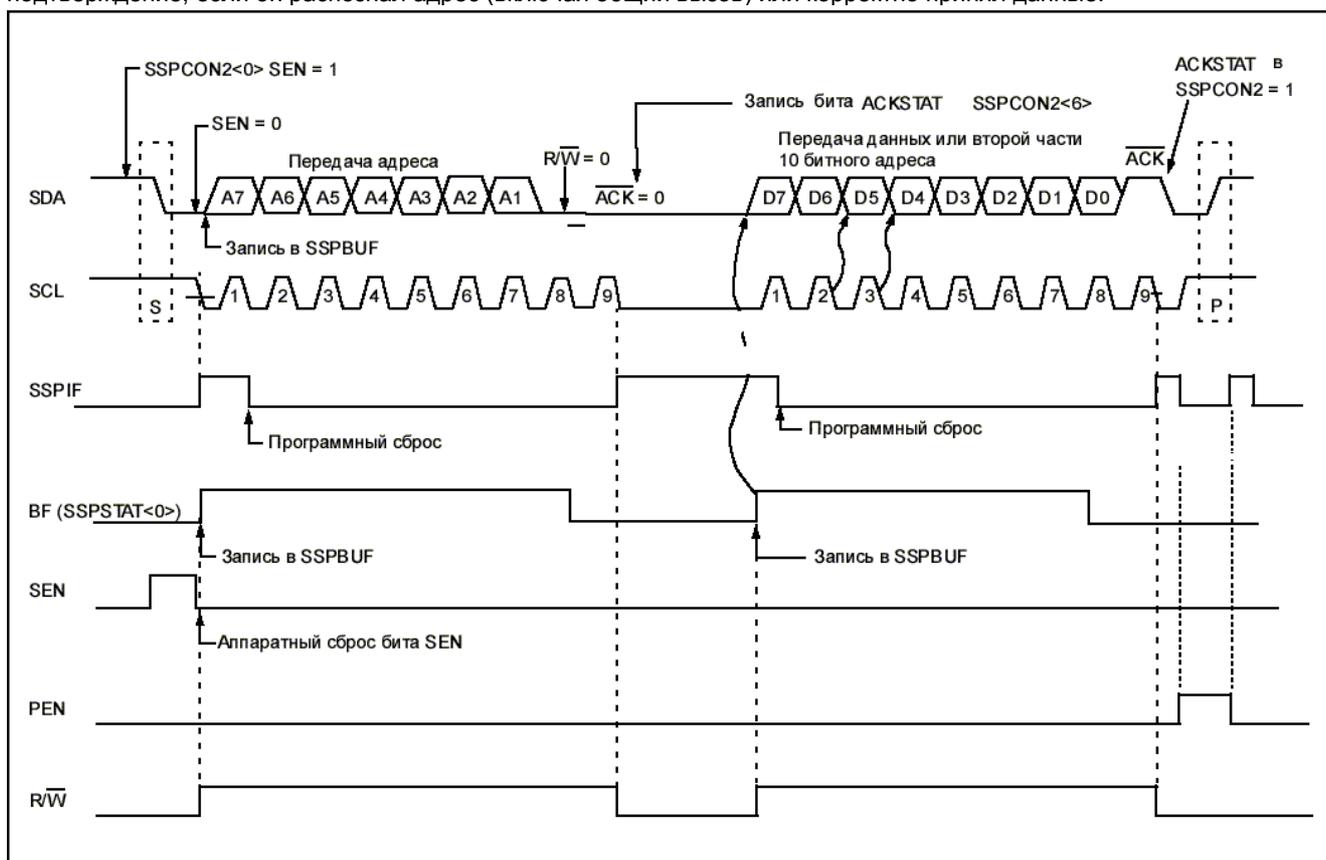
**Примечание:** поскольку MSSP не имеет стека событий, установка любого из младших 5 битов регистра SSPCON2 до завершения выполнения команды повторный START запрещено.

## Передатчик Master

Чтобы инициировать передачу байта данных, 7-битного адреса или любой части 10-битного адреса, нужно просто записать байт в регистр SSPBUF, в результате чего установится бит BF, а BRG начнет формировать сигнал для передачи данных. Каждый передаваемый бит будет выдаваться на SDA по заднему фронту SCL. Низкий уровень на SCL удерживается в течении одно периода BRG. Данные должны поступать на SDA до прихода переднего фронта на SCL. После отпускания SCL в высокий уровень, на время TBRG, данные должны удерживаться на SDA в том же состоянии. По окончании передачи 8-го бита сбрасывается флаг BF, а Master отпускает SDA с тем, чтобы принять бит подтверждения. По заднему фронту 9-го такта значение ACK записывается в бит ACKSTAT регистра SSPCON2. В этот же момент устанавливается флаг SSPIF, а BRG отключается до следующей операции на шине, оставляя низкий уровень на SCL и отпуская SDA.

Если во время передачи данных производится попытка записи в SSPBUF, устанавливается бит WCOL, а запись не происходит.

При передаче данных, бит ACKSTAT равен нулю, если Slave выдал подтверждение. Водомый посылает подтверждение, если он распознал адрес (включая общий вызов) или корректно принял данные.



### Приемник Master

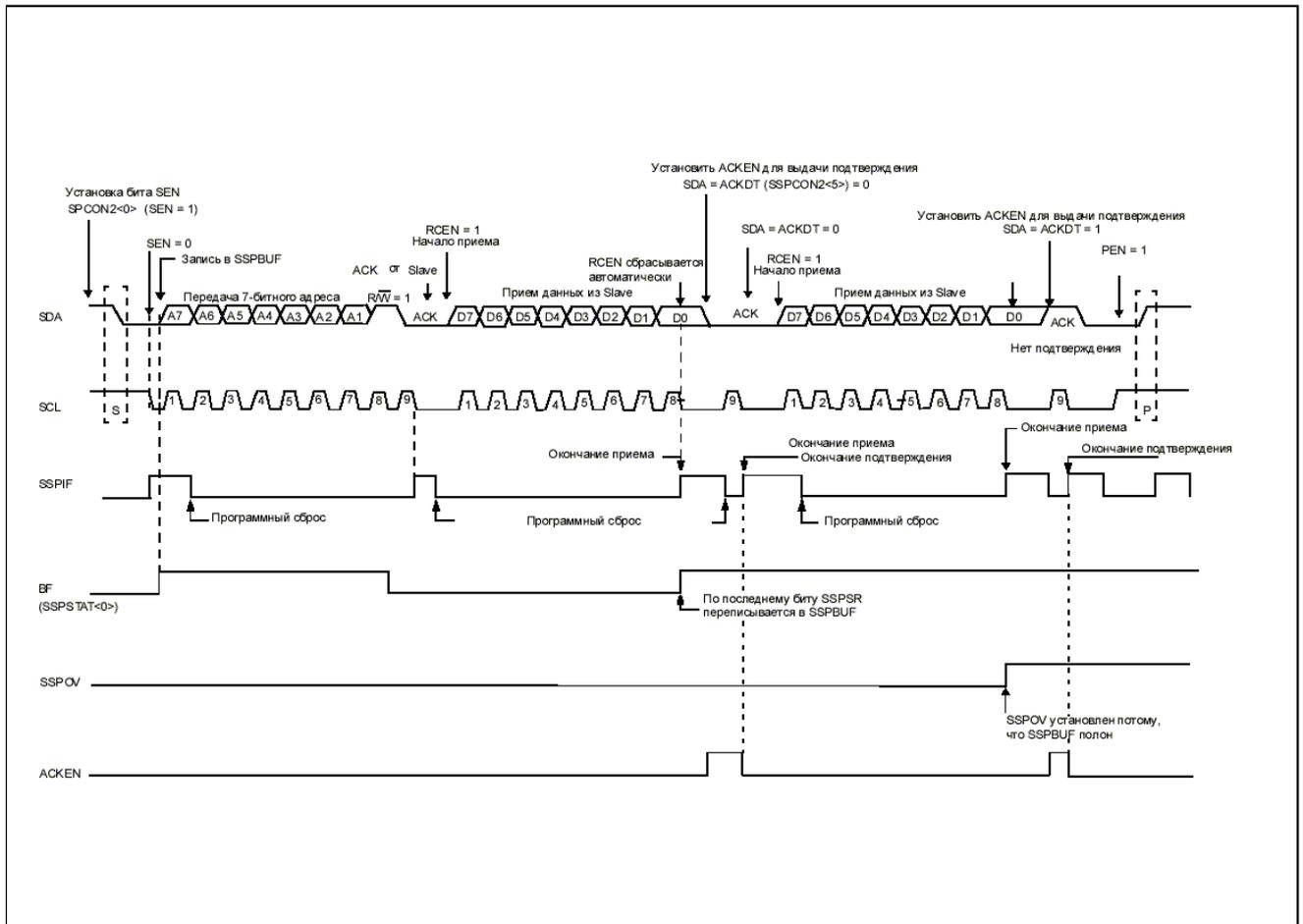
Master включает прием данных установкой бита RCEN в регистре SSPCON2.

**Примечание:** При установки бита RCEN модуль MSSP должен находиться в режиме ожидания

BRG начинает формировать тактовый сигнал, для последующего приема данных. Каждый бит данных будет приниматься с SDA по заднему фронту SCL. По заднему фронту 8-го такта, значение из SSPSR переписывается в SSPBUF, устанавливается бит BF и SSPIF, сбрасывается RCEN, BGR останавливается, удерживая SCL в низком уровне, а модуль MSSP переходит в режим ожидания. После чтения SSPBUF сбрасывается бит BF. По окончании приема, Master может сформировать бит подтверждения установкой бита ACKEN в регистре SSPCON2.

При приеме данных, бит SSPOV устанавливается, если в момент приема 8-го бита, бит BF еще установлен после предыдущего приема.

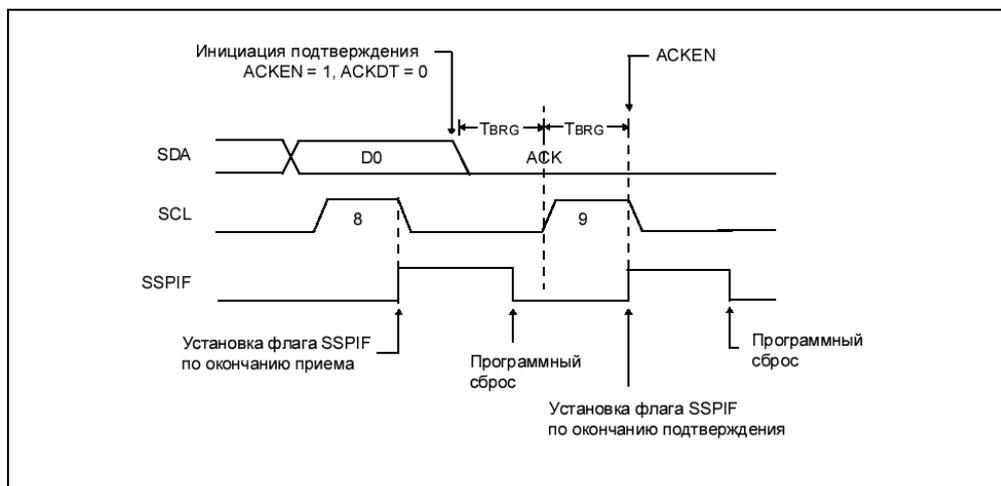
Если во время приема данных производится попытка записи в SSPBUF, устанавливается бит WCOL, а запись не происходит.



### Временная диаграмма подтверждения

Чтобы инициировать подтверждение на шине I2C, необходимо установить бит ACKEN в регистре SSPCON2. При установке этого бита на SCL выдается низкий уровень, а на SDA содержимое бита ACKDT. Если нужно подтвердить прием бит ACKDT должен быть равен нулю. По окончании счета BRG, SCL отпускается в высокий уровень. Как только SCL перейдет из низкого уровня в высокий, BRG опять начнет счет. После окончания счета, SCL переводиться в низкий уровень, бит ACKEN автоматически сбрасывается, устанавливается флаг прерывания SSPIF, BGR останавливается, а модуль MSSP переходит в режим ожидания.

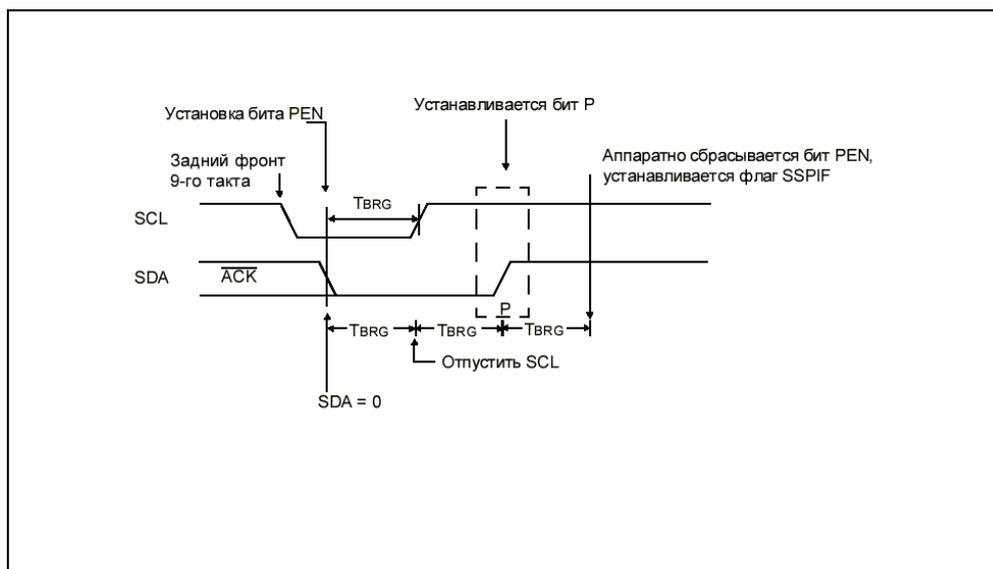
Если во время подтверждения производится попытка записи в SSPBUF, устанавливается бит WCOL, а запись не происходит.



### Временная диаграмма команды STOP

Чтобы инициировать команду STOP, необходимо установить бит PEN в регистре SSPCON2. По окончании приема/передачи данных, после прохождения заднего фронта тактового сигнала SCL удерживается в низком уровне. При установке бита PEN Master выдает низкий уровень на линии SDA, перезагружает BRG и начинает счет. По окончании счета отпускается линия SCL в высокий уровень, через время TBRG после установки высокого уровня на SCL отпускается SDA. Когда на SDA появляется высокий уровень, устанавливаются биты P и SSPIF, бит PEN автоматически сбрасывается, а генератор BRG останавливается.

Если во время выполнения команды STOP производится попытка записи в SSPBUF, устанавливается бит WCOL, а запись не происходит.





### Режим конкуренции (Multi-Master), арбитраж и конфликты шины

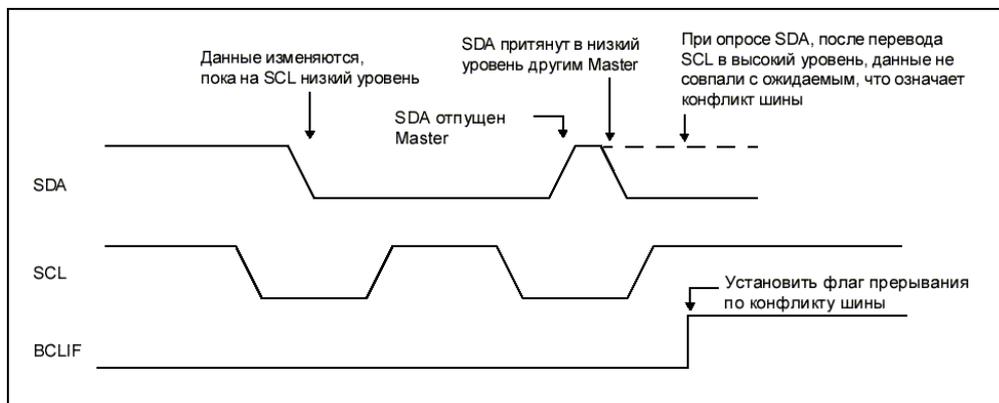
Режим конкуренции работает с использованием арбитража на шине I2C. Когда Master выдает адрес/данные на SDA, он теряет арбитраж, если он формирует высокий уровень, а другой Master низкий. При переходе SCL в высокий уровень, сигнал на SDA изменяться не может. Если на SDA ожидается высокий уровень, а в действительности низкий, значит, возник конфликт. Обнаруживший это Master устанавливает флаг прерывания BCLIF, прекращает текущую операцию на шине и переводит порт I2C в режим ожидания.

Если при возникновении конфликта шины выполнялась передача данных, она обрывается, устанавливается бит BF, а линии SCL и SDA отпускаются в высокое состояние. В регистр SSPBUF может быть произведена запись, причем запись в SSPBUF инициирует передачу независимо от того, в какой момент передатчик отключился при возникновении конфликта шины. Если пользователь обрабатывает прерывания по конфликту шины, после освобождения шины он может продолжить обмен, выполнив команды START.

Если при возникновении конфликта шины выполнялись команды START, повторный START, STOP или ACK, выполняемая операция обрывается, SCL и SDA отпускаются, а соответствующий бит управления в SSPCON2 сбрасывается. Если пользователь обрабатывает прерывания по конфликту шины, после освобождения шины он может продолжить обмен, выполнив команды START.

Master продолжает следить за состоянием шины, и при появлении команды STOP устанавливается флаг прерывания SSPIF.

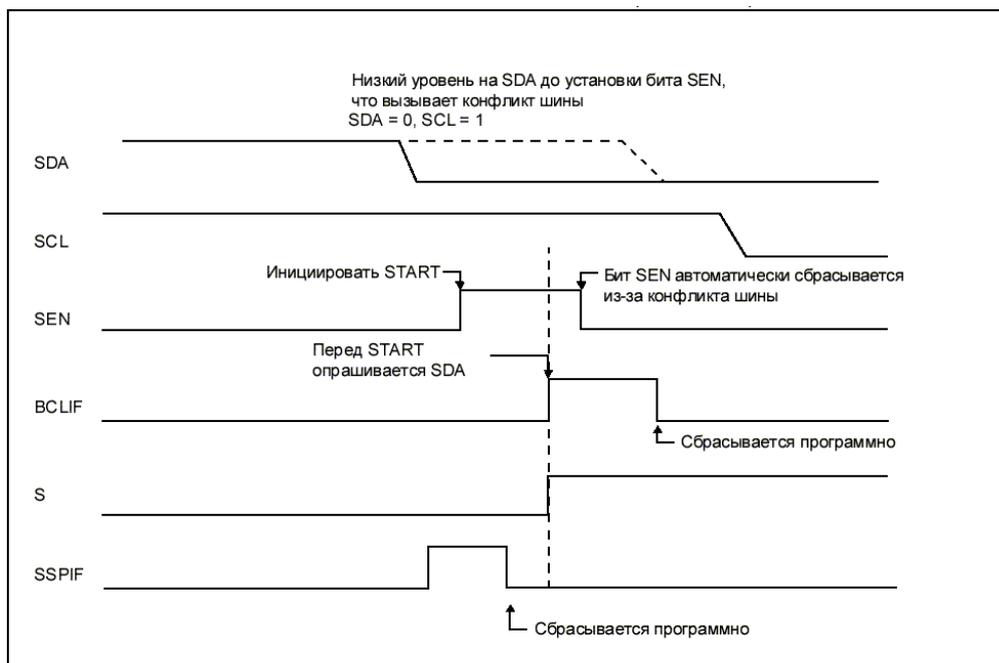
В режиме конкуренции использование прерывания при обнаружении команд START и STOP позволяет определить занятость шины. Управление шиной может быть перехвачено при установленном бите P, или сброшенных битах S и P.



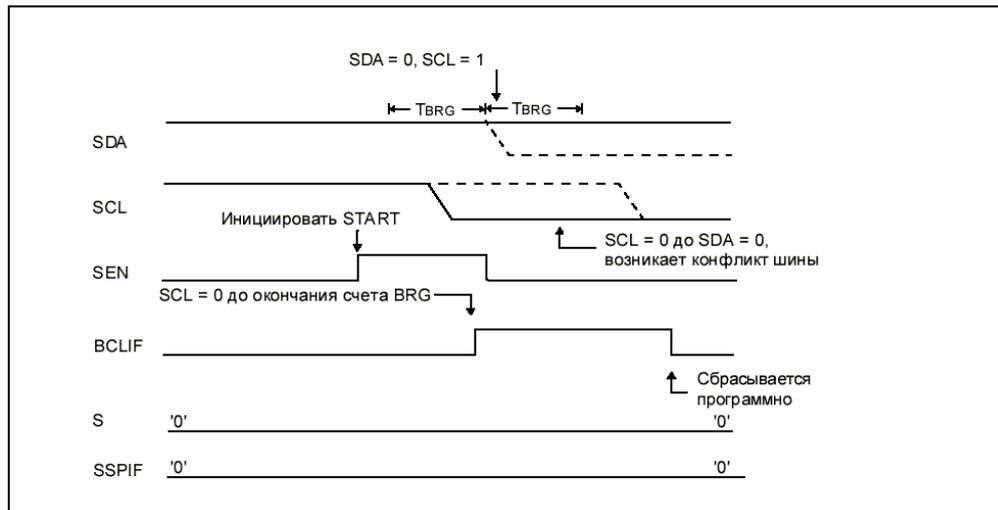
### Конфликт шины во время выполнения команды START

Во время выполнения команды START конфликт шины возникает если:

- в начале START на SDA или SCL низкий уровень сигнала



- на SCL низкий уровень появляется раньше чем на линии SDA



Во время выполнения START сигналы SCL и SDA продолжают отслеживаться.

Если

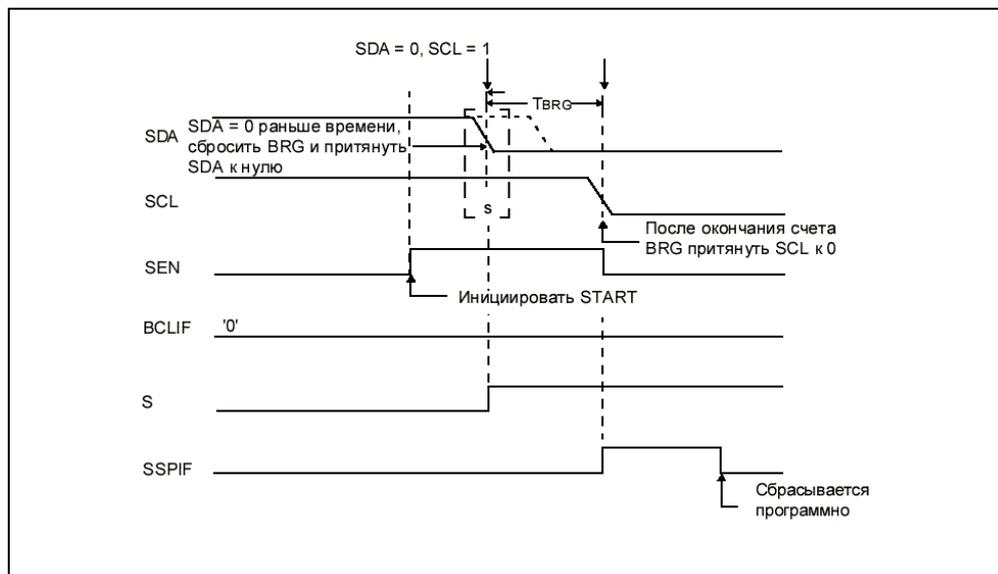
SCL или SDA в низком уровне

То

Выполнение команды START прекращается  
Устанавливается флаг BCLIF  
Модуль MSSP переводиться в режим ожидания

Команда START начинается, когда сигналы на линиях SCL и SDA высокого уровня. Если на SCL появляется низкий уровень раньше чем на SDA, возникает конфликт шины, поскольку это подразумевает, что другой Master пытается в это время передать данные.

Если во время счета BRG, на SDA появляется низкий уровень, BRG сбрасывается, а на SDA формируется низкий уровень раньше времени.



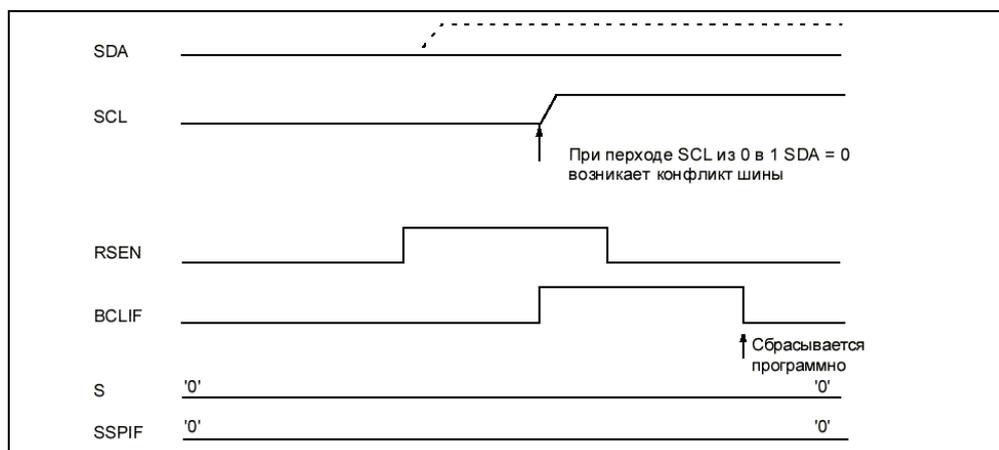
Если же на SDA высокий уровень, низкий уровень формируется в конце счета BRG, после чего тот перезагружается, и если в это время на SCL появиться низкий уровень, конфликт шины не возникает. В конце счета BRG SCL переводиться в низкий уровень.

Примечание: Конфликт шины во время START не возникает потому, что два или более Master могут сформировать START одновременно, но при этом один из них первым переведет SDA в низкий уровень. Конфликт шины не возникает, поскольку Master могут продолжить арбитраж во время передачи адреса, данных, выполнения команд повторный START и STOP.

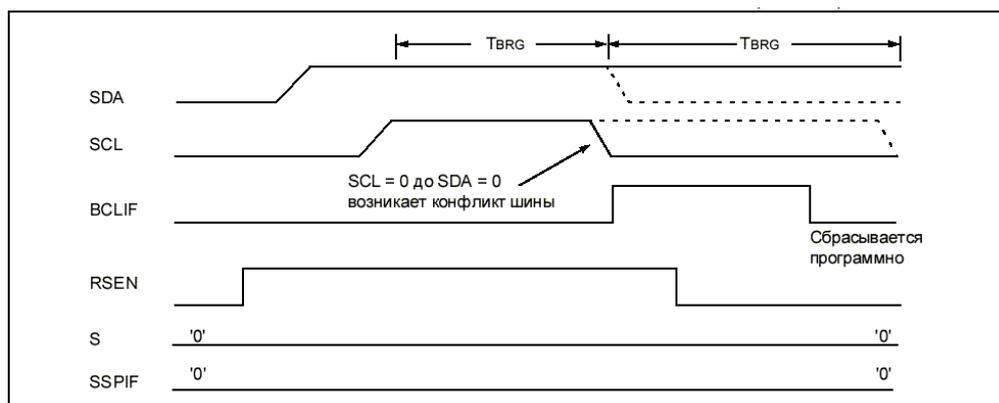
**Конфликт шины во время выполнения команды повторный START**

Во время выполнения команды повторный START конфликт шины возникает если:

- на SDA низкий уровень при переходе SCL из низкого уровня в высокий



- SCL переходит в низкий уровень раньше SDA, что указывает на то, что другой Master пытается передать данные



После отпущения SDA сигнал на выводе должен перейти в высокий уровень, после чего BRG перезагружается и начинает счет. Затем отпускается SCL и при появлении на нем высокого уровня опрашивается SDA. Если на SDA низкий уровень, значит произошел конфликт шины, т.е. другой Master пытается передать данные. Если на SDA высокий уровень, то BRG снова перезагружается и начинается счет. Если SDA переходит в низкий уровень до окончания счета, конфликт шины не происходит, поскольку два или более Master могут пытаться получить доступ к шине одновременно.

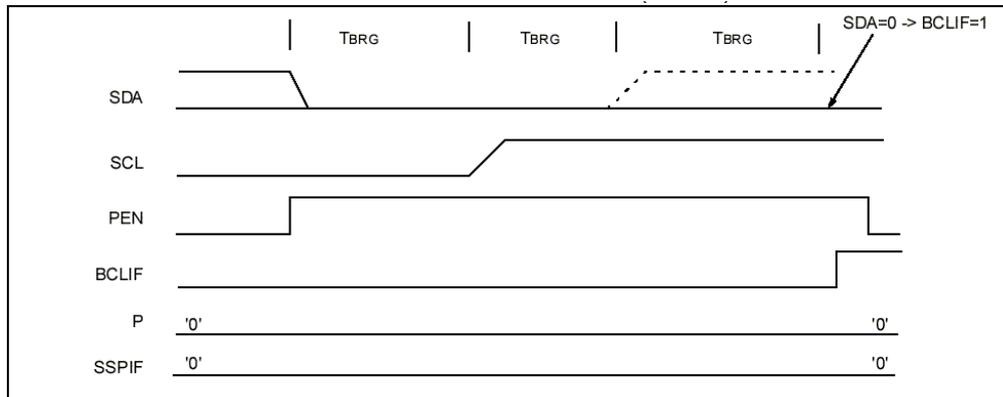
Если на SCL переходит в низкий уровень до окончания счета, а на SDA сохраняется высокий уровень, значит, произошел конфликт шины, т.е. другой Master пытается передать данные.

Если по окончании счета BGR на SCL и SDA высокий уровень, то SDA переводиться в низкий уровень, а BRG перезагружается и начинает счет. По окончании счета, независимо от уровня сигнала на SCL он переводиться в низкий уровень.

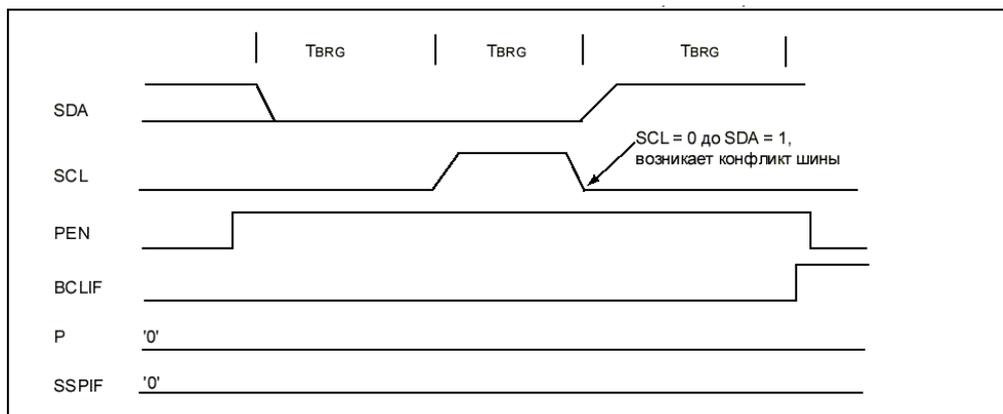
### Конфликт шины во время выполнения команды STOP

Во время выполнения команды STOP конфликт шины возникает если:

- после отпущания SDA в высокий уровень и окончания счета BRG на SDA по-прежнему низкий уровень



- после отпущания SDA в высокий уровень, SCL переходим в низкий уровень до того, как SDA перейти в высокий



Выполнение команды STOP начинается с перевода SDA в низкий уровень, после чего SCL отпускается в высокий. После появления на SCL высокого уровня BRG перезагружается и начинает счет. По окончании счета, после SDA, BRG перезагружается и снова начинает счет, затем опрашивает SDA. Если на нем низкий уровень, или на SCL появился низкий уровень до перехода SDA в высокий, значит, произошел конфликт шины, т.е. другой Master пытается передать данные.

### Соединение с шиной I2C

Для устройств в стандартном режиме значение резисторов  $R_p$  и  $R_s$  зависит от:

- напряжения питания
- емкость шины
- количества устройств на шине (входной ток + ток утечки)

Напряжение питания ограничивает минимальное значение  $R_p$  из-за ограничения минимального тока стока 3mA и  $V_{ol\ max} = 0.4V$ .

Например:

$$V_{dd} = 5V \pm 10\%$$

$$V_{ol\ max} = 0.4\ V \text{ при } 3\text{mA}$$

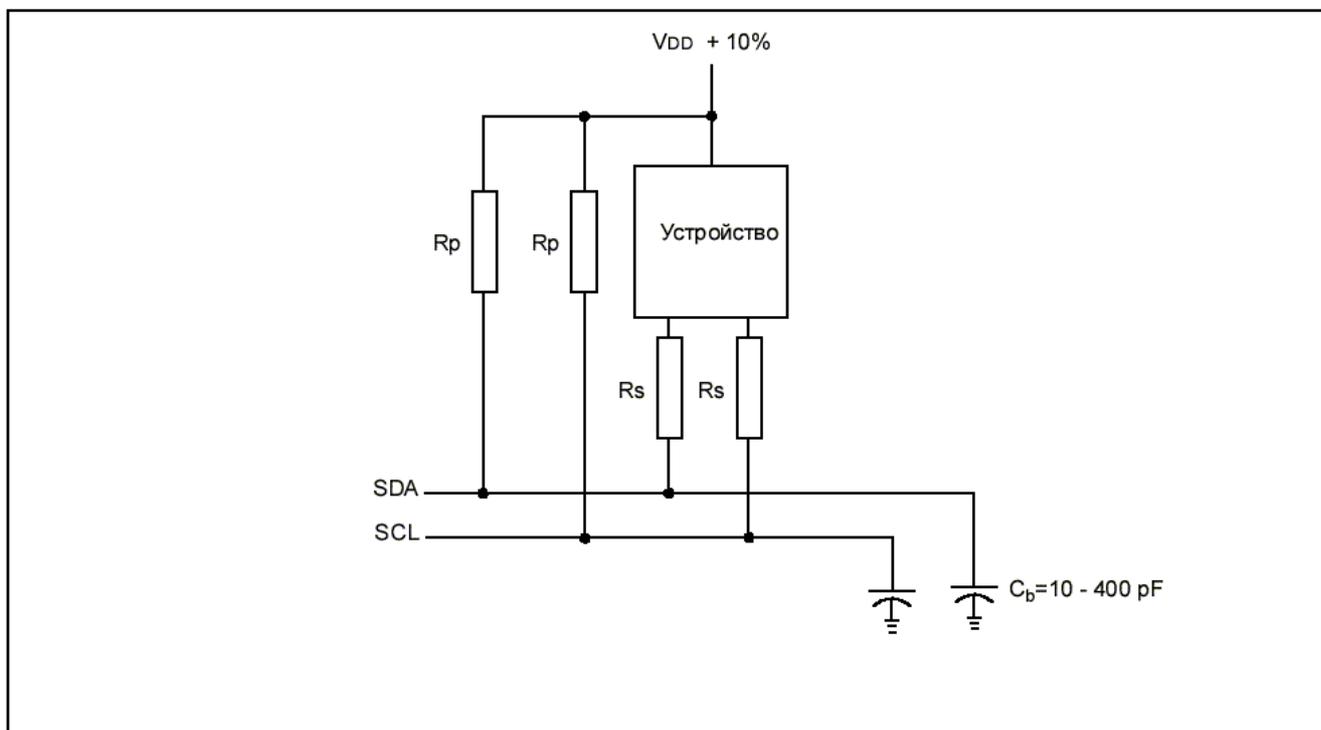
$$R_p\ min = (5.5 - 0.4) / 0.003 = 1.7\ \text{k}\Omega$$

Максимальное значение  $R_s$  определяется допустимым уровнем шума.

Емкость шины определяется суммарная емкостью проводов и выводов. Емкость определяет максимальное значение  $R_p$  из-за допустимой длительности фронта.

Бит SMP в регистре SSPSTAT включает управление длительностью фронта SDA и SCL для того, чтобы фронты этих сигналов удовлетворяли спецификации при работе в скоростном режиме с частотой 400 КГц.

Устройства на шине I2C должны иметь один источник питания, к которому подключаются подтягивающие резисторы.



## Связанные с модулем MSSP в режиме I2C интерфейса регистры и биты

Адрес	Имя	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0	POR, BOR	Reset
0Bh, 8Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTIF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIF	CP1IF	TM2IF	TM1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CP1IE	TM2IE	TM1IE	0000 0000	0000 0000
0Dh	PIR2	-	-	-	EEIF	BCLIF	-	-	CCP2IF	---0 0--0	---0 0--0
8Dh	PIE2	-	-	-	EEIE	BCLIE	-	-	CCP2IE	---0 0--0	---0 0--0
13h	SSPBUF	Буфер приема/передачи данных								xxxx xxxx	uuuu uuuu
93h	SSPADD	Регистр адреса								0000 0000	0000 0000
14h	SSPCON	WCOL	SSPOV	SSPEN	СКР	SSPM3:SSPM0				0000 0000	0000 0000
94h	SSPSTAT	SMP	CKE	D/A	P	S	R/W	UA	BF	0000 0000	0000 0000
91h	SSPCON2	GCEN	ACKST	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	0000 0000
87h	TRISC	Регистр направления PORTC								1111 1111	1111 1111

Примечание: x = неизвестно, u = без изменений, - = не используются, читаются как '0'. Затененные клетки не используются.

Статья основывается на технической документации DS30292b  
компании Microchip Technology Incorporated, USA.