

# Спецификация модуля передатчика SL-канала

## **SITransmitter**

Василий Мочалов  
Версия 1.1, 15.12.2017

# Оглавление

1. Описание .....	1
2. Описание SL-канала .....	1
3. Описание верхнего уровня передатчика .....	1
4. Программная модель .....	3
4.1. Служебный регистр .....	3
4.2. Регистр данных к отправке .....	4
5. Работа с программной моделью .....	5
5.1. Запись и чтение регистров .....	5
5.2. Отправка сообщений .....	6
5.3. Смена конфигурации .....	6
5.4. Работа с прерываниями .....	7
5.5. Выключение модуля .....	7
6. Принцип работы .....	8
7. Алгоритм работы .....	8
7.1. Смена конфигурации и сброс прерываний в режиме ожидания .....	10
7.2. Отправка сообщения .....	11
7.3. Изменение конфигурации и сброс прерываний во время отправки сообщения .....	11
7.4. Формирование запроса на прерывание .....	11
7.5. Режим отправки сообщений .....	13
7.6. Счетчик циклов .....	13
7.7. Счетчик импульсов .....	13
7.8. Соответствие значения счетчика импульса и значений на выходах SL0 и SL1 .....	13
7.9. Подсчет четности .....	13
7.10. Прекращение отправки .....	14

## 1. Описание

Данный проект подразумевает реализацию RTL-описания на языке Verilog одноканального передатчика SL-канала. Передатчик отправляет SL-сообщения. Сообщения могут содержать информацию четной разрядности от 8 до 32 бит. Бит четности формируется автоматически. Частота импульсов может меняться от 500кГц до 16МГц (при частоте тактового сигнала = 16МГц).

## 2. Описание SL-канала

SL - канал - последовательный однонаправленный канал обмена данными, разработанный для внутрислатного и межслатного обмена информацией. Обмен данными типа "точка-с-точкой". Канал состоит из двух линий: линии единиц и линии нулей. Пассивный уровень на линиях - единица. В случае передачи данных каждый разряд кодируется отрицательным импульсом на соответствующей линии. Информация передается словами младшими разрядами вперед. Предпоследний разряд - четность. Передатчик вычисляет четность таким образом, чтобы количество импульсов на линии единиц с учетом разряда четности было нечетным, а на линии нулей - четным. Приемник контролирует четность индивидуально по каждой линии. Последним импульсом является синхроимпульс, представляющий собой отрицательные импульсы по обоим линиям одновременно. Синхроимпульс означает, что передача закончена. Не допускается перекрытия информационных импульсов во время передачи.

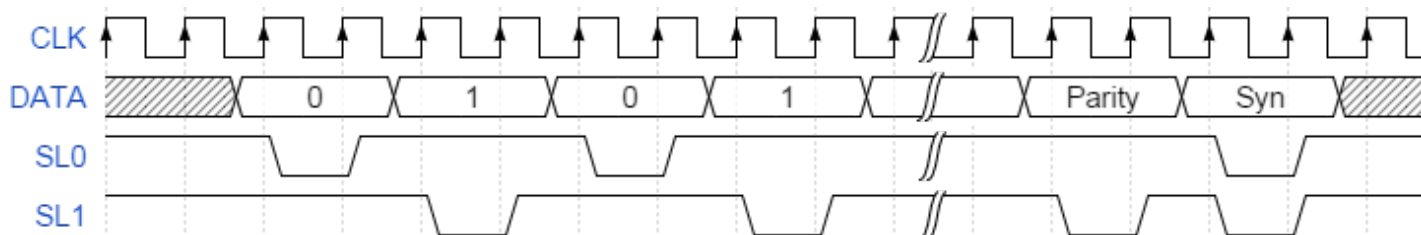


Рисунок 1. Временная диаграмма SL-канала

Типичная рабочая частота передатчика, спроектированного в синхронном стиле, составляет от 500кГц до 1МГц. Пауза между информационными битами равна длительности отрицательного импульса.

## 3. Описание верхнего уровня передатчика

Таблица 1. Порты цифрового модуля *SLTransmitter*

Название	Тип	Разрядность	Значение после сброса	Описание
<i>rst_n</i>	In	1	-	Асинхронный общий сигнал сброса
<i>clk</i>	In	1	-	Сигнал тактовой частоты
<i>addr</i>	In	1	-	Сигнал выбора регистра

<i>wr_en</i>	In	1	-	Сигнал разрешения записи
<i>D_in</i>	In	32	-	Данные для записи в регистры
<i>SL0</i>	Out	1	b1	Сигнал нулей SL канала
<i>SL1</i>	Out	1	b1	Сигнал единиц SL канала
<i>irq</i>	Out	1	b0	Сигнал запроса на прерывание
<i>D_out</i>	Out	32	h0000_0000	Данные для чтения регистров

## 4. Программная модель

Пользователю для работы доступно несколько регистров:

- Служебный (**config\_status\_r**)
- Данных к отправке (**txdata\_r**)

### 4.1. Служебный регистр

Регистр служебный регистр состоит из двух частей - конфигурации и состояния. Части отвечающей за конфигурацию соответствуют младшие 16 разрядов, части состояния старшие.

*Таблица 2. Назначение разрядов конфигурационной части служебного регистра (**config\_status\_r [15:0]**)*

Bit	15-14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	<b>IRQM[3:0]</b>				<b>FQM[2:0]</b>			<b>BC[6:0]</b>						<b>SR</b>
Mode	R	R/W				R/W			R/W						R/W
Initial	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

*Описание разрядов регистра конфигурационной части служебного регистра (**config\_status\_r [15:0]**)*

1. **SR** (soft reset) — включает (SR = 0) и выключает (SR = 1) приемник
2. **BC** (bit count) — количество разрядов данных в отправляемом сообщении
3. **IRQM** (interrupt request mask) — маска разрядов причин прерываний. Задаёт, какие именно разряды причин прерываний вызывают запрос на прерывание. Описание разрядов причин прерываний можно посмотреть в [таблице назначения разрядов части состояния служебного регистра](#). Соответствие разрядов поля IRQM и разрядов причин прерываний можно посмотреть в соответствующей [таблице](#)
4. **FQM** (frequency mode) — Поле настройки частоты. Соответствие значения FQM и делителя частоты описано в [таблице](#) ниже.

*Таблица 3. Соответствие значения поля **FQM [2:0]** и делителя частоты работы передатчика*

FQM2	FQM1	FQM0	Делитель частоты
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
0	0	0	32
1	0	1	32
1	1	0	32
1	1	1	32

*Таблица 4. Назначение разрядов части состояния служебного регистра (**config\_status\_r [31:16]**)*

Bit	31-28	27	26	25	24	23-17	16
Name	-	<b>IRQDWE</b>	<b>IRQIC</b>	<b>IRQCC</b>	<b>IRQSM</b>	-	<b>SIP</b>
Mode	R	R/W0	R/W0	R/W0	R/W0	R	R
Initial	0	0	0	0	0	0	0

*Описание разрядов части состояния служебного регистра (**config\_status\_r** [31:16])*

1. **SIP** (send in process) — разряд идущего процесса отправки сообщения.
2. **IRQSM** (interrupt request of sent message) — разряд успешно отправленного сообщения.
3. **IRQWCC** (interrupt request of wrong configuration change) — разряд попытки сменить конфигурацию во время отправки сообщения.
4. **IRQICC** (interrupt request of incorrect configuration change ) — разряд попытки установить неверную конфигурацию.
5. **IRQDWE** (interrupt request of data write error) — разряд попытки записать сообщение во время отправки предыдущего.

*Таблица 5. Соответствие разрядов **IRQM** [3:0] и маскирования разрядов причин прерываний*

Разряд поля <b>IRQM</b>	Маскируемый разряд
<b>IRQM0</b>	<b>IRQSM</b>
<b>IRQM1</b>	<b>IRQWCC</b>
<b>IRQM2</b>	<b>IRQICC</b>
<b>IRQM3</b>	<b>IRQDWE</b>

## 4.2. Регистр данных к отправке

**txdata\_r**[31:0]

*Таблица 6. Назначение разрядов регистра данных к отправке (**txdata\_r**)*

Bit	31 - 0
Name	<b>DATA</b>
Mode	R/W
Initial	0

*Описание разрядов регистра данных к отправке (**txdata\_r**)*

**DATA** - данные к отправке.

# 5. Работа с программной моделью

## 5.1. Запись и чтение регистров

Управление модулем осуществляется путем записи или чтения регистров.

Для считывания текущего значения одного из регистров блока необходимо последовательный на порт *addr* адрес регистра, указанный в [таблице](#), длительностью не меньше такта опорной тактовой частоты *clk*. Значение регистра будет сформировано на шине *D\_out* через такт опорной частоты после фронта сигнала на шине *addr*.

Для записи значения в один из регистров блока необходимо сформировать:

- на порт *addr* — адрес регистра
- на шине *D\_in* — записываемую информацию,
- на порт *wr\_en* — значение 1.

Также на на шине *d\_out* через такт опорной частоты *clk* после фронта сигнала на шине *addr* будет сформировано значение записанного регистра. Значение шины *d\_out* будет соответствовать значению последнего опрошенного или записанного регистра до формирования следующего запроса.

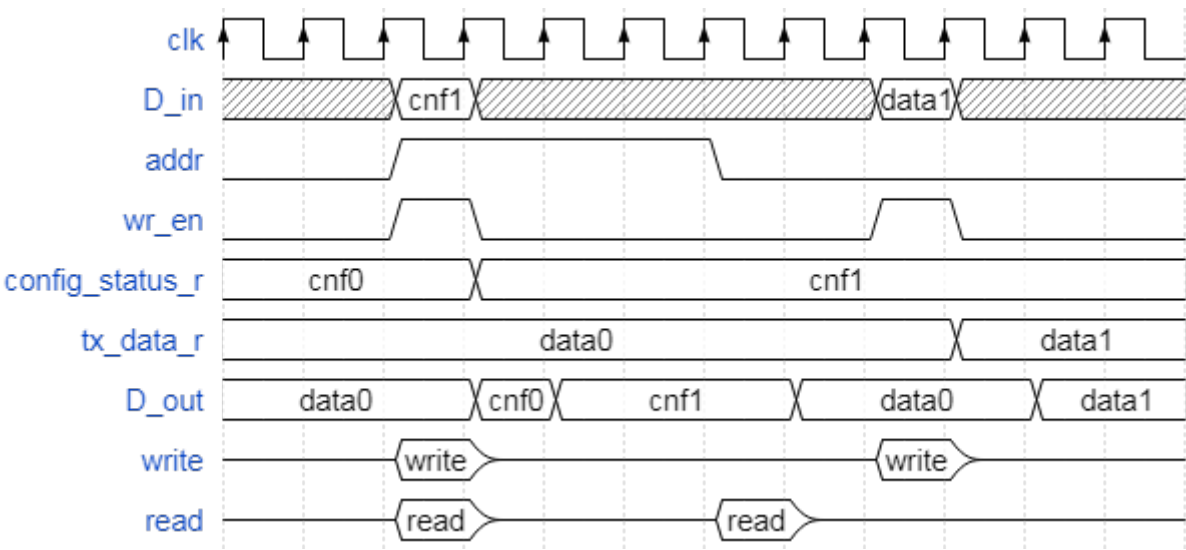


Рисунок 2. Временная диаграмма чтения и записи регистров модуля SlTransmitter

Таблица 7. Адреса регистров

Значение сигнала <i>addr</i>	Выбранный регистр
b0	регистр данных ( <i>txdata_r</i> )
b1	служебный регистр ( <i>config_status_r</i> )

Запись в регистр данных во время отправки сообщения, приведет к прекращению отправки и формированию прерывания.

## 5.2. Отправка сообщений

Для отправки сообщений необходимо:

1. Записать в регистр **config\_r** необходимые настройки частоты и длины слова (см. раздел "[Смена конфигурации](#)")
2. Записать в регистр данных сообщение на отправку
3. Дождавшись запроса на прерывания вызванного отправкой сообщения, или, работая по таймеру и периодически опрашивая регистр состояния, убедиться, что сообщение было отправлено (**IRQSM** == 1).
4. Сбросить поле причины прерывания **IRQSM**. Это необходимо для того, чтобы узнать, когда закончится следующая отправка сообщения.
5. Записать в регистр данных следующее сообщение.

Сразу после записи в регистр данных модуль переходит в режим отправки сообщения. При этом поле **SIP** регистра состояния устанавливается в "1".

В случае когда поле **BC** служебного регистра не равно 32, отправляемым сообщением являются младшие разряды регистра данных. Старшие разряды регистра, которые не входят в длину сообщения, заданную полем **BC** регистра **config\_r** (**txdata\_r** [**31:32-BC**]), будут записаны в регистр, но игнорированы при отправке.

Во время отправки сообщения нельзя записывать новое сообщение - отправка будет прервана, модуль вернется в режим ожидания нового сообщения.

Если отправка не была прервана попыткой изменения конфигурации или записью следующего сообщения, в конце отправки будет выставлен разряд **IRQSM** = 1, и разряд **SIP** = 0. Если разряд **IRQSM** не замаскирован (см. таблицу описания поля [IRQM](#)), возникнет запрос на прерывание.

## 5.3. Смена конфигурации

Для изменения конфигурации передатчика необходимо:

1. Считав служебный регистр убедиться, что разряд **SIP** равен 0. Нельзя изменять поля **BC** и **FQM** во время отправки сообщения - отправка будет прервана.
2. Записать новые параметры в служебный регистр.

В конфигурационной части служебного регистра может быть установлена необходимая частота, длина слова, маскировка причин запроса прерывания или осуществлен сброс модуля к исходным настройкам.

Неверной считается конфигурация с нечетными длинами слова или длиной слова лежащей вне промежутка от 8 до 32 разрядов. При попытке записать подобную конфигурацию



## 5.4. Работа с прерываниями

Запрос прерывания происходит, когда произошло одно из событий и разряд причины прерываний соответствующий этому событию не замаскирован:

- **IRQSM** — Отправка сообщения завершена
- **IRQICC** — Была предпринята попытка записать неверную конфигурацию в конфигурационный регистр
- **IRQWCC** — Изменение конфигурации в процессе отправки сообщения
- **IRQDWE** — Попытка записать новые данные во время отправки сообщения

Узнать какое именно событие вызвало запрос на прерывание можно в [полях причин прерываний](#) служебного регистра.

Для сброса прерывания необходимо записать 0 в разряды причин прерываний, которые необходимо сбросить.

Более подробно работа прерываний рассмотрена в разделе [Алгоритм работы](#).

## 5.5. Выключение модуля

Чтобы выключить модуль необходимо записать 1 в разряд **SR** служебного регистра.

Если сделать это во время отправки сообщения, отправка прекращается. Служебный регистр возвращается в начальное состояние, регистр данных сбрасывается. На выходах *SL0* и *SL1* устанавливается 1. Когда передатчик выключен, запись в регистр данных игнорируется.

## 6. Принцип работы

Отправка сообщения обеспечивается двумя счетчиками: счетчиком импульсов и счетчиком циклов.

Счетчик циклов обеспечивает деление частоты. Каждый раз, когда счетчик циклов оказывается равным нулю, переключается счетчик импульсов.

На нечетных значениях счетчика импульсов на выходы *SL0* и *SL1* подаются значения соответствующие информационным импульсам, потом импульсу четности и синхроимпульсу.

На четных значениях счетчика импульсов на выходы *SL0* и *SL1* подаются единицы.

## 7. Алгоритм работы

В устройстве используются следующие вспомогательные сигналы и регистры:

Название	Тип	Разрядность	Значение после сброса	Описание
<i>end_of_msg</i>	сигнал	1	b0	Сигнал конца сообщения
<i>new_conf_is_corr</i>	сигнал	1	b0	Сигнал корректности разрядов шины <i>D_in</i> соответствующих полю ВС имеют верное значение
<i>config_changed</i>	сигнал	1	b0	Сигнал отличия разрядов шины <i>D_in</i> соответствующих полям ВС и FQM и полей ВС и FQM служебного регистра
<i>no_error</i>	сигнал	1	b0	Сигнал отсутствия ошибки в управлении регистром (смена конфигурации или запись в регистр данных во время отправки)

<i>fq_max</i>	сигнал	5	b0	Сигнал равный значению делителя частоты уменьшенному на 1
<b>shift_r</b>	регистр	1	b0	Сдвиговый регистр с отправляемым сообщением
<b>par0</b>	регистр	1	b0	Регистр подсчета четности на линии нулей
<b>par1</b>	регистр	1	b0	Регистр подсчета четности на линии единиц
<b>fq_counter_r</b>	регистр	5	b0	Регистр счетчика циклов
<b>bit_counter_r</b>	регистр	7	b0	Регистр счетчика количества импульсов

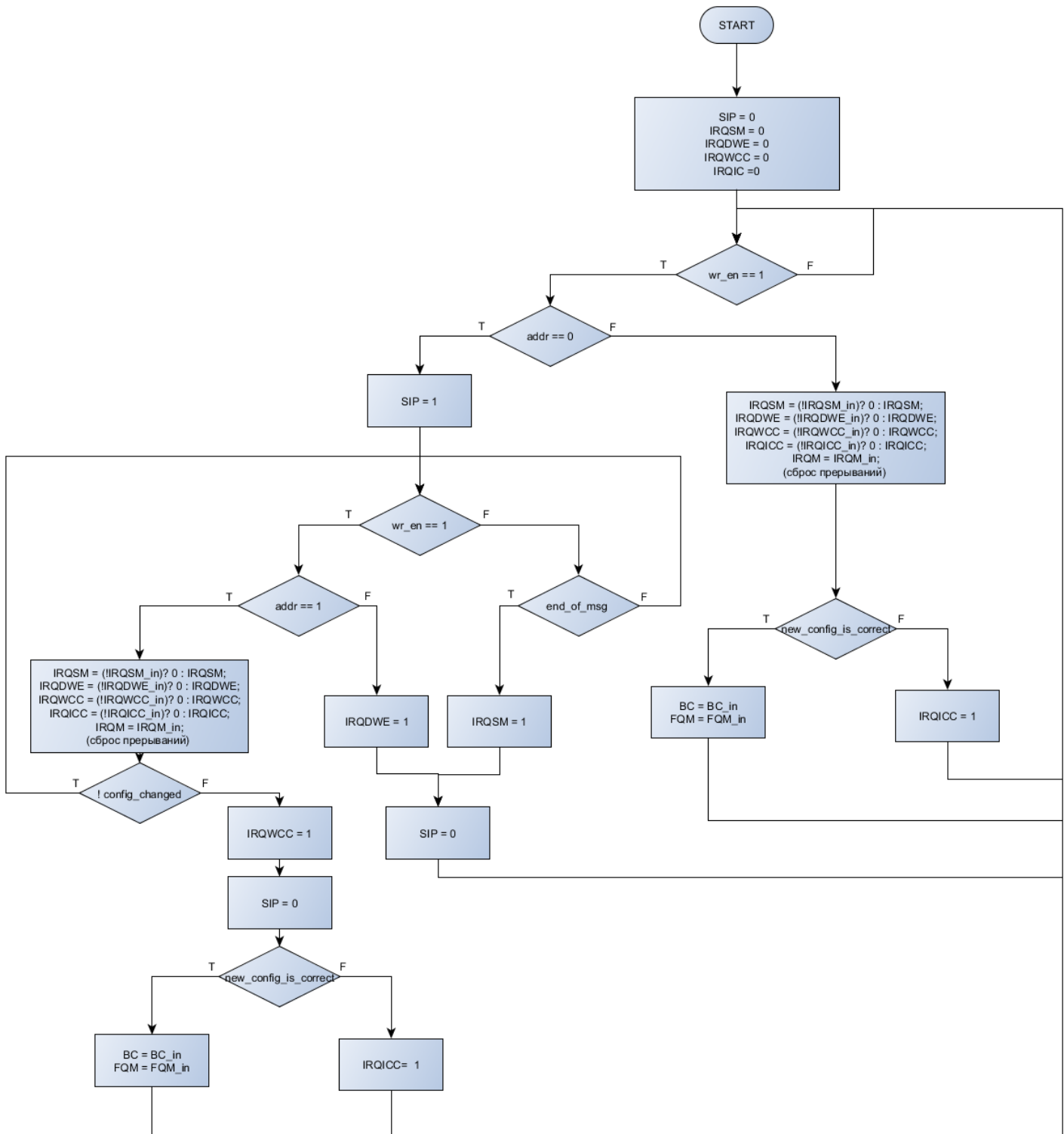


Рисунок 3. Алгоритм работы модуля SITransmitter

Модуль может находиться в двух режимах: режим отправки и режим ожидания. После включения модуля, все разряды части состояния служебного регистра устанавливаются в 0, модуль находится в режиме ожидания.

### 7.1. Смена конфигурации и сброс прерываний в режиме ожидания

При записи служебного регистра в режиме ожидания происходит проверка разрядов прерываний:

1. Если значения разрядов шины *D\_in* соответствующие разрядам полей причин прерываний служебного регистра, равны 0, то соответствующие поля причин прерываний служебного регистра сбрасываются.
2. Обновляется поле **IRQM** служебного регистра
3. Если разряды шины *D\_in* соответствующие полю **BC** являются нечетным числом или числом не из интервала от 6'd8 до 6'd32, выставляется **IRQICC** = 1, поля **BC** и **FQM** не изменяются. Если разряд **IRQICC** не замаскирован формируется запрос на прерывание. Если же конфигурация корректна, поля **BC** и **FQM** шины *D\_in* записывается в служебный регистр. Модуль остается в режиме ожидания.

## 7.2. Отправка сообщения

Сразу после записи данных в режиме ожидания, модуль переходит в режим отправки, устанавливается поле служебного регистра **SIP** = 1. По успешному окончанию отправки сообщения, устанавливаются поля **SIP** = 0 и **IRQSM** = 1. Если же отправка сообщения была прервана попыткой смены конфигурации или записи в регистр данных, выставляются устанавливаются поля **SIP** = 0 и **IRQWCC** = 1 или **IRQDWE** = 1. Поле **IRQSM** в случае прерванной отправки не устанавливается в 1.

## 7.3. Изменение конфигурации и сброс прерываний во время отправки сообщения

При записи служебного регистра в режиме отправки сообщения происходит проверка разрядов причин прерываний:

1. Если значения разрядов шины *D\_in* соответствующие разрядам полей причин прерываний служебного регистра, равны 0, то соответствующие поля причин прерываний служебного регистра сбрасываются.
2. Обновляется поле **IRQM** служебного регистра.
3. Если разряды шины *D\_in* соответствующие полям **BC** и **FQM** имеют значения отличные от полей **BC** и **FQM** служебного регистра выставляется выставляется **IRQWCC** = 1 и **SIP** = 0, отправка сообщения прекращается.
4. Если разряды шины *D\_in* соответствующие полю **BC** являются нечетным числом или числом не из интервала от 6'd8 до 6'd32, выставляется **IRQICC** = 1 и **SIP** = 0, поля **BC** и **FQM** не изменяются. Если же конфигурация корректна, поля **BC** и **FQM** служебного регистра перезаписываются.

## 7.4. Формирование запроса на прерывание

Запрос на прерывание формируется на выходе *irq*, через один такт после возникновения причины прерывания, если причина этого прерывания не замаскирована в поле **IRQM** [3:0].

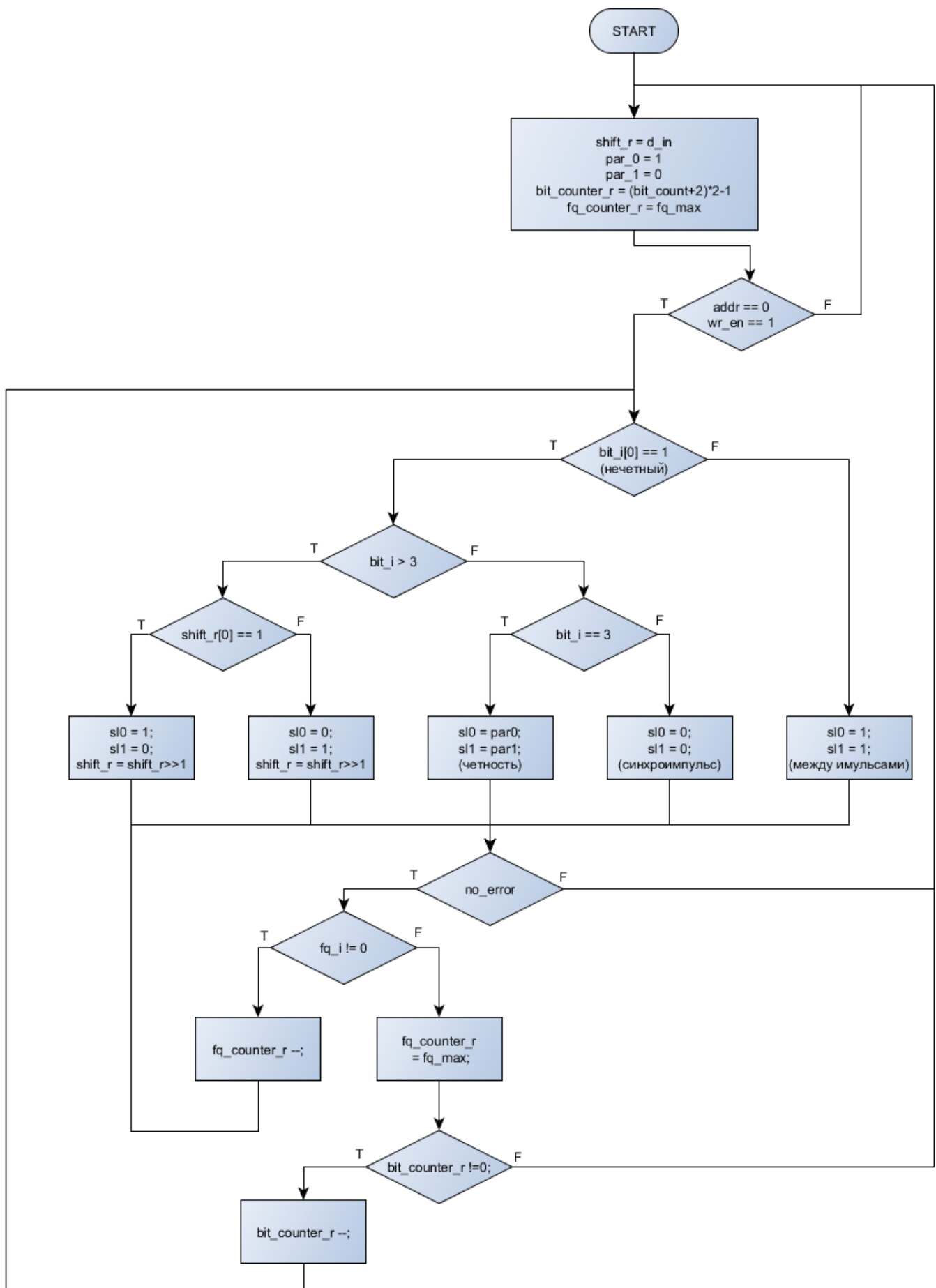


Рисунок 4. Алгоритм работы модуля SlTransmitter в режиме отправки

## 7.5. Режим отправки сообщений

В начале режима отправки в сдвиговый регистр загружается отправляемое сообщение. Для организации отправки используются два счетчика:

- Счетчик циклов (**bit\_counter\_r**)
- Счетчик импульсов (**fq\_counter\_r**)

## 7.6. Счетчик циклов

В режиме ожидания в счетчик циклов загружается максимальное значение  $fq\_max$  равное делителю частоты уменьшенному на 1. В режиме отправки счетчик частоты считает от значения  $fq\_max$  до значения 0. Когда счетчик частоты достигает нуля, счетчик импульсов увеличивается на 1, а в счетчик частоты снова загружается максимальное значение.

## 7.7. Счетчик импульсов

В режиме ожидания в счетчик импульсов загружается максимальное значение равное  $(BC + 2) * 2 - 1$ . В режиме отправки счетчик импульсов считает от числа, равного  $(BC + 2) * 2 - 1$  до 0, где BC (bit count) - поле служебного регистра. Число  $(BC + 2) * 2 - 1$  получено следующим образом: необходимо отправить число разрядов информации заданное полем BC, разряд четности и синхрои́мпульс. В сумме BC+2 импульсов. Между импульсами нужно выставить на выход промежуточную комбинацию, таким образом, значение удваивается и получается  $(BC + 2) * 2$  итераций счетчика и максимальное значение равное  $(BC + 2) * 2 - 1$ .

## 7.8. Соответствие значения счетчика импульса и значений на выходах SL0 и SL1

На значениях счетчика импульсов  $(BC + 2) * 2 - 1$ ,  $(BC + 2) * 2 - 3$ , .. 7, 5 на выходы *SL0* и *SL1* устанавливается комбинация соответствующая первому разряду сдвигового регистра. Также происходит подсчет четности на основе первого разряда сдвигового регистра и сдвиг регистра. Таким образом формируются информационные импульсы.

На всех четных значениях —  $(BC + 1) * 2$ ,  $(BC + 1) * 2 - 2$ , .. 2, 0 на выход устанавливается комбинация, соответствующая промежутку между отрицательными импульсами (*SL0* = 1 и *SL1* = 1)

На значении счетчика импульсов **bit\_counter\_r** = 3 на выход выставляется комбинация соответствующая подсчитанный четности, а на значении **bit\_counter\_r** = 1 - комбинация синхрои́мпульса (*SL0* = 0 и *SL1* = 0).

## 7.9. Подсчет четности

Подсчет четности осуществляется при помощи регистров **par0** и **par1**. Регистр **par0** имеет начальное значение 1, и инвертируется каждый раз при отправке информационного разряда со значением 0. Регистр **par1** имеет начальное значение 0, и инвертируется каждый раз при отправке

информационного разряда со значением 1.

Таким образом, если в отправляемом сообщении будет четное число единиц, то  $par1$  и  $par0$  изменятся четное количество раз и после отправки всех информационных разрядов получаем  $par0 = 1$ ,  $par1 = 0$ . подав эти значения на выходы  $SL0$  и  $SL1$  получим отрицательный импульс на линии единиц, общее же количество импульсов на линии единиц окажется нечетным, а на линии нулей - четным.

Если же в отправляемом сообщении будет четное число единиц, то  $par1$  и  $par0$  изменятся нечетное количество раз и после отправки всех информационных разрядов получаем  $par0 = 0$ ,  $par1 = 1$ . подав эти значения на выходы  $SL0$  и  $SL1$  получим отрицательный импульс на линии нулей, общее же количество импульсов на линии единиц окажется нечетным, а на линии нулей - четным.

## 7.10. Прекращение отправки

На каждом значении счетчика частоты происходит проверка наличия ошибок - попытки записать данные во время отправки или изменения конфигурации. В случае, если ошибка произошла, отправка прекращается: разряд причины прерывания соответствующий событию, вызвавшему прекращение отправки, устанавливается в единицу, на выходы  $SL0$  и  $SL1$  устанавливаются единицы, модуль\*ль готов к отправке следующего сообщения.