

# 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. Регистр конфигурации и состояния .....	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
6. Принцип работы .....	8
7. Алгоритм работы .....	8
7.1. Отправка сообщения .....	9
7.2. Формирование запроса на прерывание .....	9
7.3. Счетчик частоты .....	11
7.4. Счетчик количества бит .....	11
7.5. Соответствие значения счетчика количества бит и значений на выходах SL0 и SL1 .....	11
7.6. Подсчет четности .....	11
7.7. Экстренное завершение отправки .....	12

## 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.

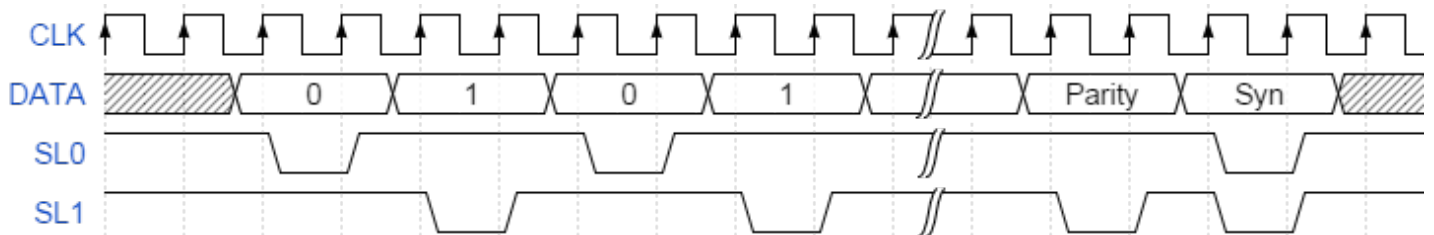


Рисунок 1. Временная диаграмма SL-канала

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>D_in</i>	In	32	-	Данные для записи в регистры

<i>addr</i>	In	1	-	Сигнал выбора регистра
<i>wr_en</i>	In	1	-	Сигнал разрешения записи
<i>D_out</i>	Out	32	h0000_0000	Данные для чтения регистров
<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

## 4. Программная модель

Пользователю для работы доступно несколько регистров:

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

### 4.1. Регистр конфигурации и состояния

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	R/W				R/W			R/W						R/W
Initial	0	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. Задает, 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](#). **FQM** (frequency mode) — 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	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	-	-	-	<b>IRQDWE</b>	<b>IRQIC</b>	<b>IRQCC</b>	<b>IRQSM</b>	-	-	-	-	-	-	-	<b>SIP</b>
Mode	R	R	R	R	R/W0	R/W0	R/W0	R/W0	R	R	R	R	R	R	R	R
Initial	0	0	0	0	0	0	0	0	0	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) — произошла попытка сменить конфигурацию во время отправки сообщения
4. **IRQICC** (interrupt request of incorrect configuration change) — произошла попытка установить неверную конфигурацию
5. **IRQDWE** (interrupt request of data write error) — произошла попытка 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.

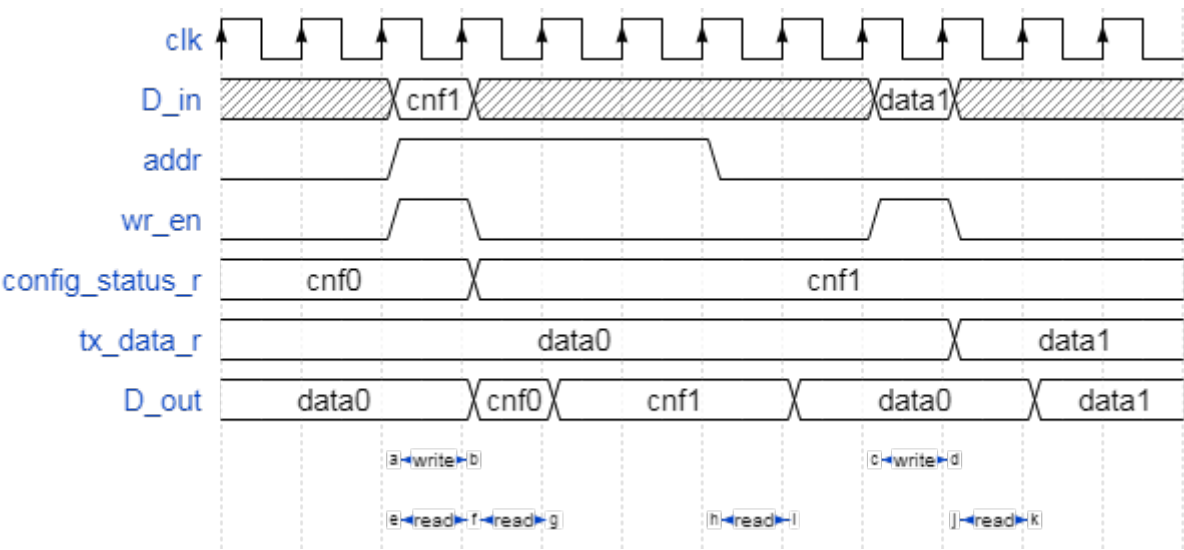


Рисунок 2. Временная диаграмма чтения и записи регистров модуля *SlTransmitter*

Таблица 7. Адреса регистров

Значение сигнала <i>addr</i>	Выбранный регистр
1'b0	регистр данных ( <i>txdata_r</i> )
1'b1	регистр конфигурации и состояния ( <i>config_r</i> и <i>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. Если это необходимо, записать в регистр `config_r` необходимые настройки частоты и длины слова (см. раздел "Смена конфигурации")
2. Записать в регистр данных сообщение на отправку
3. Если вы работаете по прерываниям, дождаться запроса на прерывания вызванного отправкой сообщения (`IRQSM == 1`).
4. Если вы работаете по таймеру, периодически опрашивая регистр состояния, убедиться, что сообщение было отправлено (`IRQSM == 1`).
5. Сбросить поле причины прерывания `IRQSM`.
6. 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` регистра конфигурации не равно 32, отправляемым сообщением являются младшие биты регистра данных. Старшие биты 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.

В конце отправки будет выставлен бит `IRQSM = 1`, и бит `SIP = 0`. Если бит `IRQSM` не замаскирован, возникнет запрос на прерывание.

Для отправки следующего сообщения нужно сбросить поле причины прерывания `IRQSM` и записать новое сообщение для отправки.

## 5.3. Смена конфигурации

Для изменения конфигурации передатчика необходимо:

1. Считав регистр конфигурации и состояния убедиться, что модуль не занят отправкой сообщения (бит `SIP` регистра конфигурации и состояния равен 0). Если он занят отправкой, дождаться, пока отправка будет завершена.
2. Записать новые параметры в регистр конфигурации и состояния.

Для изменения конфигурации передатчика необходимо перезаписать регистр конфигурации и состояния. В конфигурационной части 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 бит.



Нельзя изменять поля ВС и FQM во время отправки сообщения - отправка будет прервана.

## 5.4. Работа с прерываниями

Запрос прерывания происходит, когда произошло одно из событий и бит этого события не замаскирован :

- Отправка сообщения завершена (IRQSM)
- Была предпринята попытка записать некорректные данные в конфигурационный регистр (IRQICC)
- Изменение конфигурации в процессе отправки сообщения (IRQWCC)
- Попытка записать новые данные во время отправки старых (IRQDWE)

Причину возникновения можно посмотреть в соответствующих полях регистра состояния.

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

Более подробно работа прерываний рассмотрена в разделе Алгоритм работы. === Выключение модуля

Чтобы выключить модуль необходимо записать 1 в бит SR регистра конфигурации и состояния.

Если сделать это во время отправки сообщения, отправка прекращается. Регистры конфигурации и состояния возвращаются в начальное состояние. Когда передатчик выключен, запись в регистр данных игнорируется.

## 6. Принцип работы

Отправка сообщения обеспечивается двумя счетчиками: - счетчиком бит и счетчиком циклов.

Счетчик циклов обеспечивает деление частоты. каждый раз, когда счетчик циклов оказывается равным нулю, переключается счетчик битов.


На нечетных значениях счетчика бит на выходы *SL0* и *SL1* подаются значения соответствующие информационным битам, потом биту четности и синхроимпульсу.

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

## 7. Алгоритм работы

В устройстве используются следующие вспомогательный сигналы:

*Внутренние вспомогательные сигналы:*

- wire end\_of\_msg - сигнал конца сообщения
- wire new\_config\_is\_correct - поле BC шины *D\_in* имеет верное значение
- reg shift\_r - сдвиговый регистр с отправляемым сообщением
- reg par0, reg par1 - регистры подсчета четности
- reg bit\_i - счетчик количества бит
- reg fq\_i - счетчик делителя частоты
- wire no\_error - наличие ошибки в управлении регистром (смена конфигурации или запись в регистр данных во время отправки) 

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

=== Смена конфигурации и сброс прерываний в режиме ожидания

При записи регистра конфигурации и состояния в режиме ожидания происходит проверка битов прерываний:

1. Если значения полей причин прерываний шины *D\_in* равны 0, то соответствующие поля причин прерываний регистра status\_r сбрасываются.
2. В соответствии с битами **IRQM** шины *D\_in* обновляется поле **IRQM** регистра config\_r
3. После этого если поле **BC** шины *D\_in* нечетное или не лежит в интервале от 6'd8 до 6'd32, выставляется **IRQICC** = 1, поля BC и FQM не изменяются. Если бит IRQICC не замаскирован формируется запрос на прерывание.
4. Если конфигурация корректна поля **BC** и **FQM** шины *D\_in* записывается в регистр config\_r. Модуль остается в режиме ожидания.

## 7.1. Отправка сообщения

Сразу после записи данных в режиме ожидания, модуль переходит в режим отправки, устанавливается поле регистра `status_r SIP = 1`. По успешному окончанию отправки сообщения, устанавливаются поля регистра `status_r SIP = 0` и `IRQSM = 1`. Если бит `IRQSM` не замаскирован формируется запрос на прерывание. === Попытка отправить сообщение во время отправки предыдущего

Если записать данные в режиме отправки, выставляется бит `IRQDWE = 1`, модуль экстренно завершает отставку и возвращается в режим ожидания, выставляется биты `SIP = 0`. Если бит `IRQDWE` не замаскирован формируется запрос на прерывание.

### *Изменение конфигурации и сброс прерываний во время отправки сообщения*

Когда модуль находится в режиме отправки, то без отмены приема возможно только изменение полей маскирования прерываний, и сброс битов причин прерываний.

Если в режиме отправки происходит запись регистра конфигурации и состояния, сначала проверяются биты прерываний: если значения полей причин прерываний шины `D_in` равны 0, то соответствующие поля причин прерываний регистра состояния сбрасываются.

После этого, проверяется изменяются ли биты конфигурации (поля `FQM`, `BC`). Если они не изменяются, модуль остается в режиме отправки сообщения. Если они изменяются то отставка завершается, выставляются биты **`SIP = 0`** и **`IRQWCC = 1`**. Если бит **`IRQDWCC`** не замаскирован формируется запрос на прерывание.

Если конфигурация корректна, она записывается в регистр, если же нет, выставляется бит **`IRQICC = 1`**. Модуль переходит в режим ожидания.

## 7.2. 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`**.

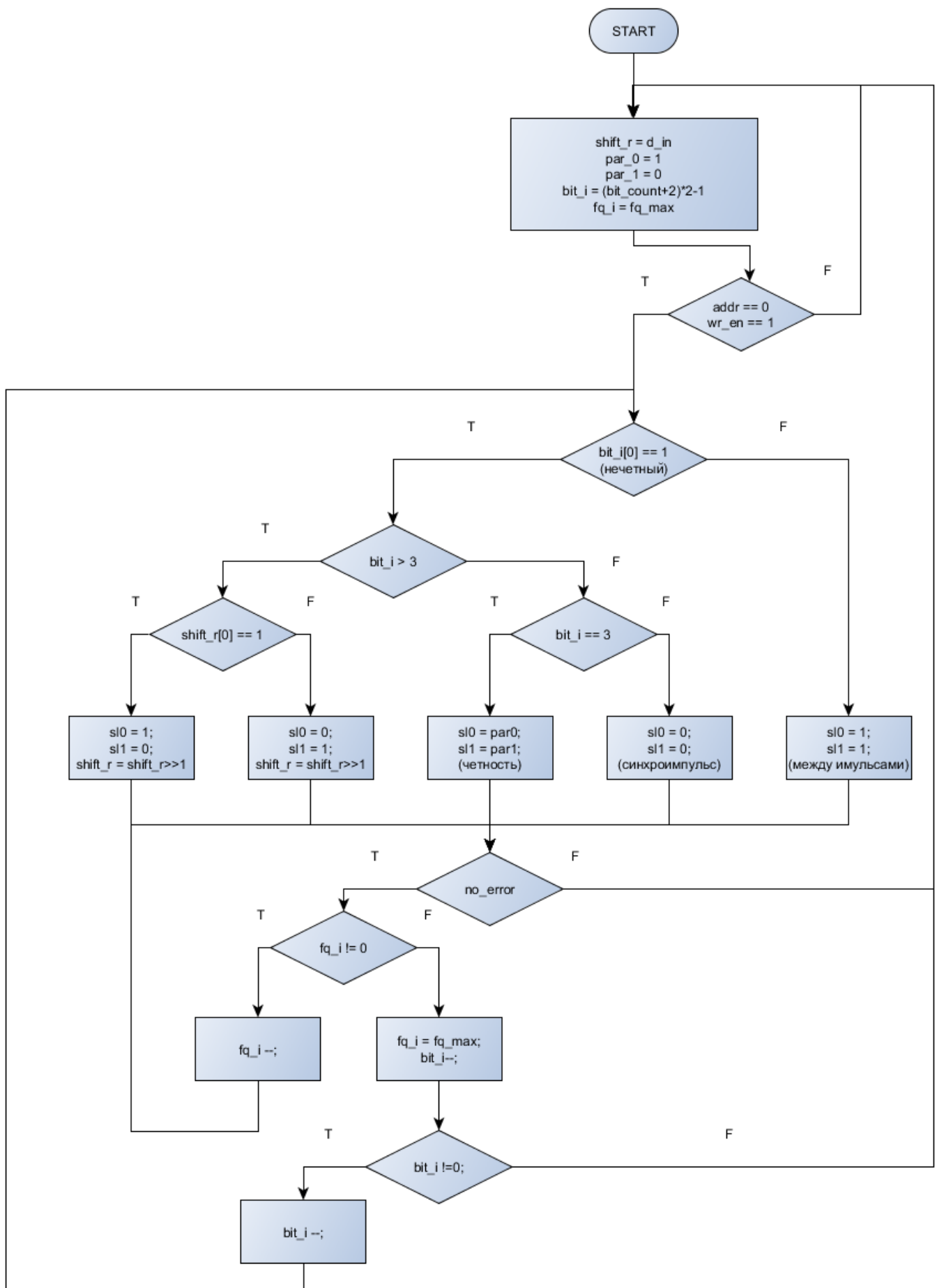


Рисунок 3. Алгоритм работы модуля SITransmitter в режиме отправки

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

### 7.3. Счетчик частоты

Счетчик частоты считает от значения (делитель частоты - 1) до значения 0. Когда счетчик частоты достигает максимального значения, инкрементируется счетчик количества бит.

### 7.4. Счетчик количества бит

Счетчик импульсов считает от числа, равного  $(BC + 2) \cdot 2 - 1$  до 0, где BC (bit count) - поле регистра конфигурации. Число  $(BC + 2) \cdot 2 - 1$  получено следующим образом: необходимо отправить число бит информации заданное полем BC, бит четности и бит синхроимпульса. В сумме BC+2 бит. Между битами нужно выставить на выход промежуточную комбинацию, таким образом значение удваивается и получается  $(BC+2) \cdot 2$  итераций счетчика и максимальное значение равно  $(BC+2) \cdot 2 - 1$ .

### 7.5. Соответствие значения счетчика количества бит и значений на выходах SL0 и SL1

На значениях счетчика битов  $(BC+2) \cdot 2 - 1$ ,  $(BC+2) \cdot 2 - 3$  .. 7, 5 на выход выставляется комбинация соответствующая первому биту сдвигового регистра. Также происходит подсчет четности на основе первого бита сдвигового регистра и сдвиг регистра. Таким образом формируются информационные биты.

На всех четных значениях  $(BC+1) \cdot 2$ ,  $(BC+1) \cdot 2 - 2$ , .. 2, 0 на выход выставляется комбинация соответствующая промежутку между значащими битами (единица на линии нулей и единица на линии единиц)

На значении счетчика битов 3 на выход выставляется комбинация соответствующая подсчитанный четности, а на значении 1 - комбинация стоп бита.

### 7.6. Подсчет четности

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

Таким образом, если единиц в отправляемом сообщении будет четное число, то par1 и par0 изменятся четное количество раз и после отправки всех информационных бит 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 единиц в отправляемом сообщении будет нечетное число, то par1 и par0 изменятся нечетное количество раз и после отправки всех информационных бит 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.

## 7.7. Экстренное завершение отправки

Также 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.