

# Микроархитектурная спецификация SITransiever

=====

# Оглавление

- Описание ..... 1
  - Назначение ..... 1
  - Параметры конфигурации..... 1
- Функциональное описание..... 1
  - Структурная схема ..... 1
  - Описание работы..... 2
  - Синхронизация и сброс..... 3
  - Подмодули ..... 3
- Описание SL-канала..... 33
  - Описание верхнего уровня ..... 33
  - Программная модель..... 34
  - Конечный автомат ..... 37

# Описание

Данный проект подразумевает реализацию RTL-описания на языке Verilog многоканального приемопередатчика SL-канала.

## Назначение

### Тип блока

Переферийный

### Функция

Сложно функциональный блок SITransiever - это блок, посредством которого осуществляется обмен данными с внешними устройствами в соответствии с протоколом обмена

## Параметры конфигурации

Параметры конфигурации СФ-блока SITransiever приведены в таблице. .Описание параметров блока SITransiever

Имя	Значение по умолчанию	Описание
CHANNEL_COUNT	1	Количество каналов. На каждый канал создается по одному приемнику и передатчику

## Функциональное описание

## Структурная схема

Структурная схема СФ-блока SITransiever приведена на рисунке:

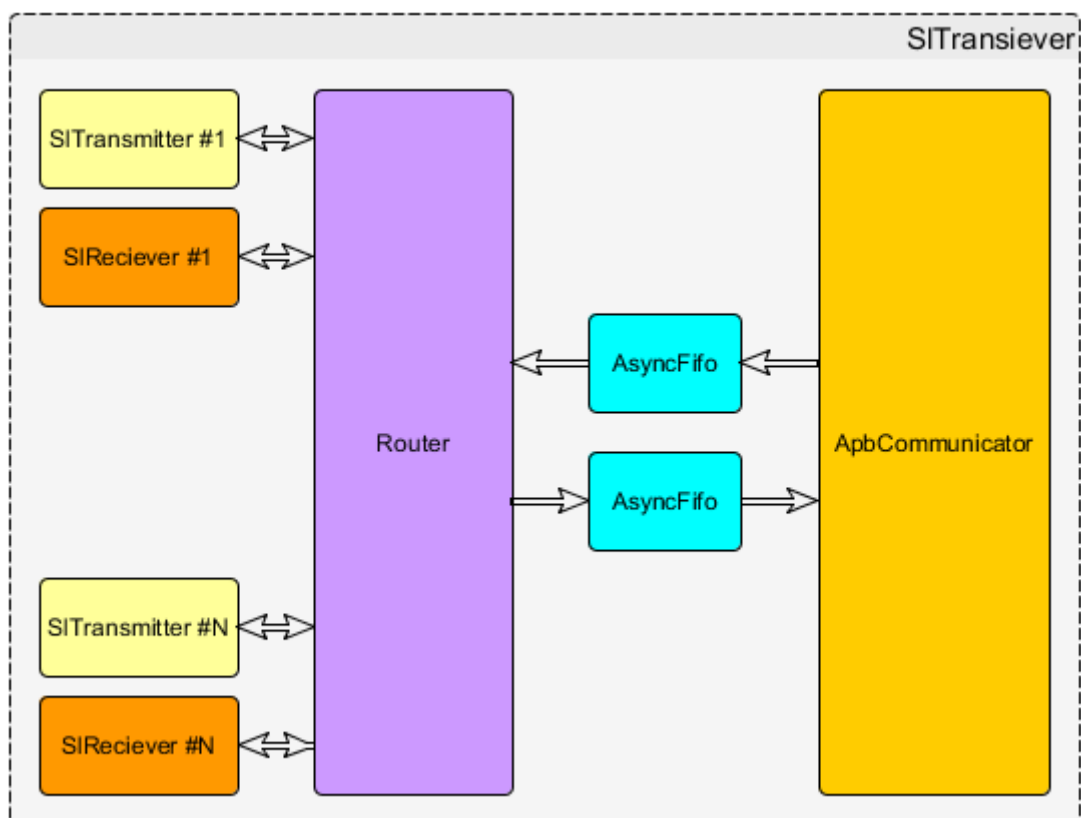


Рисунок 1. Общая схема СФ-блока SITransiever

Элементы, которые входят в СФ-блок SITransiever и представлены на структурной схеме, перечислены в таблице:

Таблица 1. Модули, входящие в СФ-блок SITransiever

Название	Функция
SITransiever	Верхний уровень СФ-блока. Выполняет функции отправки и приема сообщений по нескольким SI каналам
SITransmitter	Передатчик. Отправляет SI сообщения
SIReceiver	Приемник. Принимает SI сообщения
Router	Управляет инстансами приемников и передатчиков
AsyncFifo	Асинхронный буфер, используются для передачи информации из одного клокового домена в другой
ApbCommunicator	Обрабатывает Apb транзакции

## Описание работы

SITransiever предназначен для взаимодействия с периферийными устройствами, подключенными по SI интерфейсам. Для этого инстанцируется блок с указанным количеством каналов. На каждый канал приходится по одному приемнику и передатчику.

Каждый инстанс приемника и передатчика имеет свой адрес. Одновременно пользователь может управлять только одним устройством. Управление устройствами осуществляется записью и чтением регистров по апб шине.

## Транзакции записи

Возможна запись в следующие регистры:

### *Регистры доступные для записи*

- Регистр адреса устройства
- Регистр конфигурации выбранного устройства
- Регистр данных выбранного устройства

При транзакции записи в один из выбранных регистров модуль ArbCommunicator записывает соответствующее сообщение в асинхронный буфер. Модуль Router, в зависимости от содержимого этого сообщения, либо записывает новый адрес управляемого устройства, либо передает содержимое сообщения в управляемое устройство. Управляемое устройство задается содержимым регистра адреса устройства внутри модуля Router.

## Транзакции чтения

Модуль Router считывает значения регистров выбранного устройства при смене адреса управляемого устройства, или же при их изменении. Сообщения с соответствующими данными он помещает в обратный асинхронный буфер.

Модуль ArbCommunicator считывает эти сообщения и помещает их в соответствующие регистры, дублирующие регистры управляемого устройства. Содержимое регистров ArbCommunicator по запросу помещает на шину Arb.

## Синхронизация и сброс

СФ-блок спроектирован для работы в двух доменах синхронизации: системная частота clk и частота Arb шины pclk. Частоты clk и pclk примерно равны.

В каждом домене синхронизации необходимо наличие своего сигнала сброса с активным низким уровнем, выставляемым асинхронно и снимаемым по фронту сигнала синхронизации.

## Подмодули

### ArbCommunicator

#### Описание

Модуль используется в проекте приемопередатчика SL канала для обработки APB транзакций, записи в выходной асинхронный буфер и считывания из входного асинхронного буфера

## Описание верхнего уровня

### Входные сигналы

#### *АРВ-связанные сигналы*

- pclk - сигнал тактовой частоты
- preset\_n - сигнал сброса
- [15:0] paddr - шина адреса
- psel - сигнал выбора устройства
- penable - сигнал разрешения работы
- pwrite - сигнал выбора чтения или записи
- [31:0] pwrdata - шина записи данных

#### *Сигналы входного и выходного буферов*

- fifo\_read\_empty - сигнализирует что входной буфер пуст
- fifo\_write\_full - сигнализирует что выходной буфер полон
- [33:0] fifo\_read\_data - шина данных входного буфера

### Выходные сигналы

#### *АРВ-связанные сигналы*

- [31:0] prdata - шина чтения данных
- pready - сигнал готовности к чтению или записи данных

#### *Сигналы входного и выходного буферов*

- fifo\_read\_inc - сигнал для чтения из входного буфера
- [33:0] fifo\_write\_data - шина данных выходного буфера
- fifo\_write\_inc - сигнал для записи в выходной буфера

### Двунаправленные сигналы

Отсутствуют

### Программная модель

#### *Пользователю для работы доступно несколько регистров:*

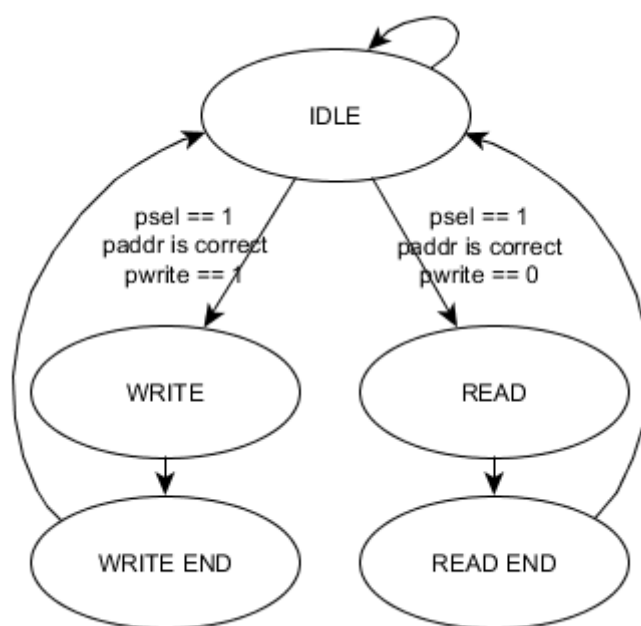
1. Конфигурационный
2. Состояния
3. Данных (при операции чтения это регистр приема, а при операции записи - отправки )
4. Адреса управляемого устройства (приемника или передатчика)

В ходе работы, модуль принимает транзакции APB шины и обрабатывает их следующим образом: При транзакции записи в соответствующий регистр, данные для записи помещаются в выходной буфер, расширенные до 34 разрядов, где 34 и 33 разряд представляют собой модификатор, однозначно задающий регистр, для которого предназначаются эти данные. При транзакции чтения на APB шину подается содержимое одного из четырех соответствующих внутренних регистров модуля, в который предварительно были записаны данные из выходного буфера.

Когда транзакций нет, а сообщения во входном буфере есть, то модуль переписывает данные из буфера в регистры, руководствуясь модификатором.

*Таблица 2. Значения модификаторов для разных регистров*

Регистр	Значение модификатора
Конфигурационный	2'd0
Данных	2'd1
Состояния	2'd2
Адреса устройства	2'd3



*Рисунок 2. Конечный автомат модуля SlTransmitter*



# Router

## Описание

Модуль используется в проекте приемопередатчика SL канала для чтения управляющих данных из первого асинхронного буфера и распределения их выбранному приемнику и передатчику, и записи данных о состоянии приемника / передатчика во второй асинхронный буфер .

## Описание верхнего уровня

### Входные сигналы

#### *Общие сигналы*

- clk - системный клок
- rst\_n - системный ресет

#### *Сигналы передатчика*

- [CHANNEL\_COUNT-1:0] rd\_status\_tx - массив выводов регистров состояния передатчиков
- [TX\_CONFIG\_REG\_WIDTH\*CHANNEL\_COUNT-1:0] rd\_config\_tx - массив выводов регистров конфигурационных регистров передатчиков
- [CHANNEL\_COUNT-1:0] status\_changed\_tx - массив выводов сигналов изменения регистра состояния

#### *Сигналы приемника*

- [RX\_STATUS\_REG\_WIDTH\*CHANNEL\_COUNT-1:0] rd\_status\_rx - массив выводов регистров состояния приемников
- [RX\_CONFIG\_REG\_WIDTH\*CHANNEL\_COUNT-1:0] rd\_config\_rx - массив выводов конфигурационных регистров приемников
- [32\*CHANNEL\_COUNT-1:0] rd\_data\_rx - массив выводов регистров данных приемников
- [CHANNEL\_COUNT-1:0] data\_status\_changed\_tx - \* status\_changed\_tx - массив выводов сигналов изменения регистров состояния и данных (регистр данных не может измениться без изменения регистра состояния)

#### *Сигналы асинхронных буферов*

- fifo\_read\_empty - сигнализирует что входной буфер пуст
- fifo\_write\_full - сигнализирует что выходной буфер полон
- [33:0] fifo\_read\_data - шина данных входного буфера

### Выходные сигналы

#### *Сигналы передатчика*

- [TX\_CONFIG\_REG\_WIDTH\*CHANNEL\_COUNT-1:0] wr\_config\_tx - данные, которые должны быть записаны в конфигурационный регистр (массив выходов для всех передатчиков)

- [CHANNEL\_COUNT-1:0] config\_we\_tx - сигналы записи в конфигурационный регистр (массив выходов для всех передатчиков)
- [32\*CHANNEL\_COUNT-1:0] wr\_data\_tx - данные, которые должны быть записаны в регистр для отправляемого сообщения (массив выходов для всех передатчиков)
- [CHANNEL\_COUNT-1:0] data\_we\_tx - сигналы записывающие данные и начинающий отправку сообщения (массив выходов для всех передатчиков)

#### *Сигналы приемника*

- [RX\_CONFIG\_REG\_WIDTH\*CHANNEL\_COUNT-1:0] wr\_config\_rx - данные, которые должны быть записаны в конфигурационный регистр (массив выходов для всех передатчиков)
- [CHANNEL\_COUNT-1:0] config\_we\_rx - сигналы для записи в конфигурационный регистр (массив выходов для всех передатчиков)
- [CHANNEL\_COUNT-1:0] word\_picked\_rx - сигнал для сброса бита принятого слова (массив выходов для всех передатчиков)

#### *Сигналы входного и выходного буферов*

- fifo\_read\_inc - сигнал для чтения из входного буфера (массив выходов для всех передатчиков)
- [33:0] fifo\_write\_data - шина данных выходного буфера (массив выходов для всех передатчиков)
- fifo\_write\_inc - сигнал для записи в выходной буфера (массив выходов для всех передатчиков)

Двунаправленные сигналы

Отсутствуют

Программная модель

#### *Пользователю для работы доступен регистр:*

1. Регистр адреса управляемого устройства

Перезаписать содержимое этого регистра можно, поместив во входной асинхронный буфер сообщение с адресом устройства и соответствующим идентификатором.

#### *Значения первого бита регистра адреса управляемого устройства*

- "0" - модуль сконфигурирован для работы как передатчик
- "1" - модуль сконфигурирован для работы как приемник

Модуль параметризуется значением параметра CHANNEL\_COUNT, для возможности подключить N передатчиков и N приемников. Таким образом адреса приемников имеют четный адрес, а адреса передатчиков имеют нечетный адрес. Все биты адреса кроме первого, задают номер устройства.

Описание работы модуля

Работа модуля делится на две части - обработка команд поступающих из входного буфера, и запись данных регистров управляемого устройства в выходной буфер. Все данные поступающие из

входного буфера и записываемые в выходной снабжаются следующими модификаторами (33 и 34 разряды содержимого сообщения).

*Таблица 3. Значения модификаторов для разных регистров*

Регистр	Значение модификатора
Конфигурационный	2'd0
Данных	2'd1
Состояния	2'd2
Адреса устройства	2'd3

### *Обработка сообщений из входного буфера*

Для обработки сообщений из входного буфера используется машина состояний, работающая по следующему алгоритму: Если выполнены следующие условия, то машина состояний переходит из состояния ожидания в соответствующее состояние записи регистра управляемого устройства. Управляемое устройство задается регистром адреса устройства.

### *Условия перехода*

- Буфер не пуст
- Приемник/передатчик не занят (Для сообщений данных и конфигурации)

При этом, при попытке записать данные в передатчик (у него нет входа для регистра данных), а также при сообщении содержащим данные для регистра состояния (запись в регистр состояния запрещена), сообщение просто уничтожается. В случае смены устройства, содержимое сообщения записывается в регистр адреса устройства, В случае изменения данных передатчика/приемника на соответствующие выходы подается сообщение из буфера и write\_enable для соответствующего входа выставляется в "1".

Следующим тактом машина состояний возвращается в состояние ожидания сообщения, единицы на выходах write\_enable переключаются в 0. При смене устройства и управлении конфигурационными регистрами генерируются внутренние сигналы "channel\_changed" , "rx\_config\_changed", "tx\_config\_changed". Их назначение будет описано далее.

### *Запись сообщений в выходной буфер*

В выходной буфер записываются сообщения следующим образом:

#### *Серия сообщений записываемая при смене адреса устройства (addr\_changed == 1)*

- текущий адрес управляемого устройства
- регистр данных текущего устройства (только для приемников)
- регистр состояния текущего устройства
- конфигурационный регистр текущего устройства

#### *Серия сообщений записываемая при смене регистра состояния модуля, находящегося на текущем устройстве (data\_status\_changed\_rx == 1 , status\_changed\_tx == 1)*

- регистр данных текущего устройства (только для приемников)

- регистр состояния текущего устройства
- конфигурационный регистр текущего устройства

При `config_changed_rx == 1` и `config_changed_tx == 1` в асинхронный буфер записывается сообщение с данными регистра выбранного устройства.

При возникновении конкурирующего импульса, он будет игнорирован. Возникновение таких ситуаций не предусматривается другими модулями.

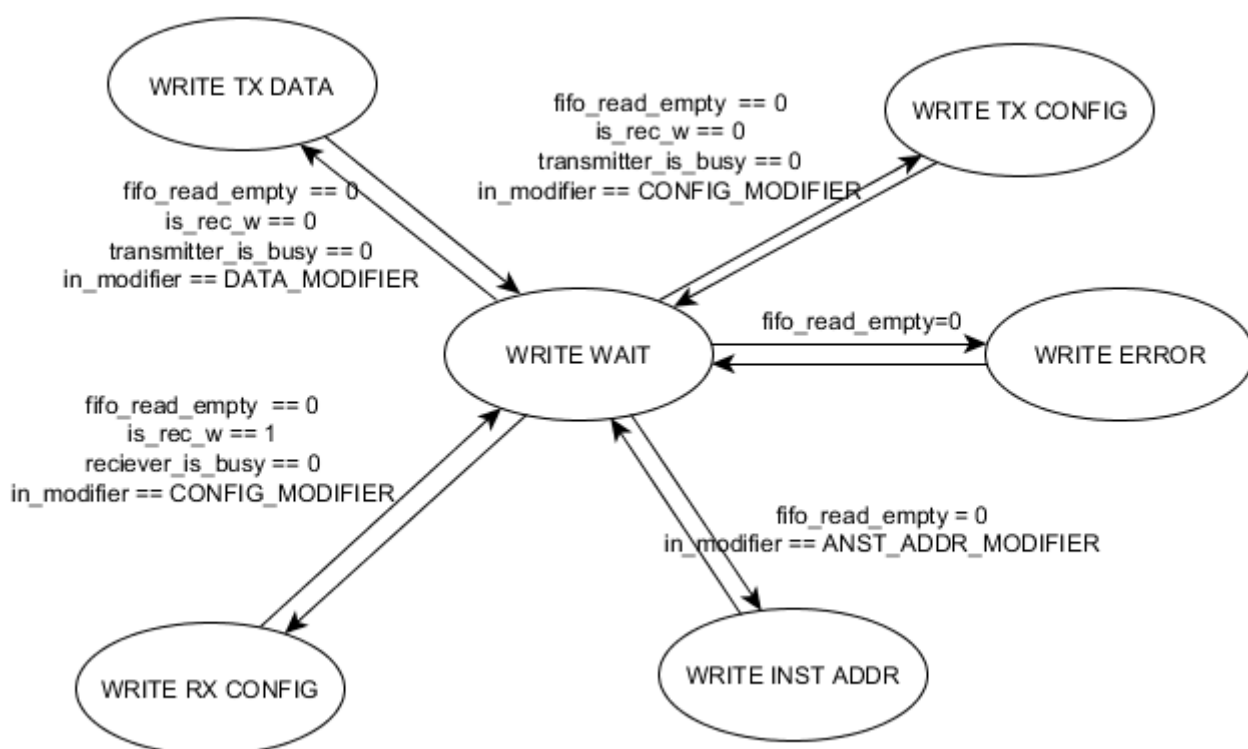


Рисунок 3. Конечный автомат транзакций записи в регистры

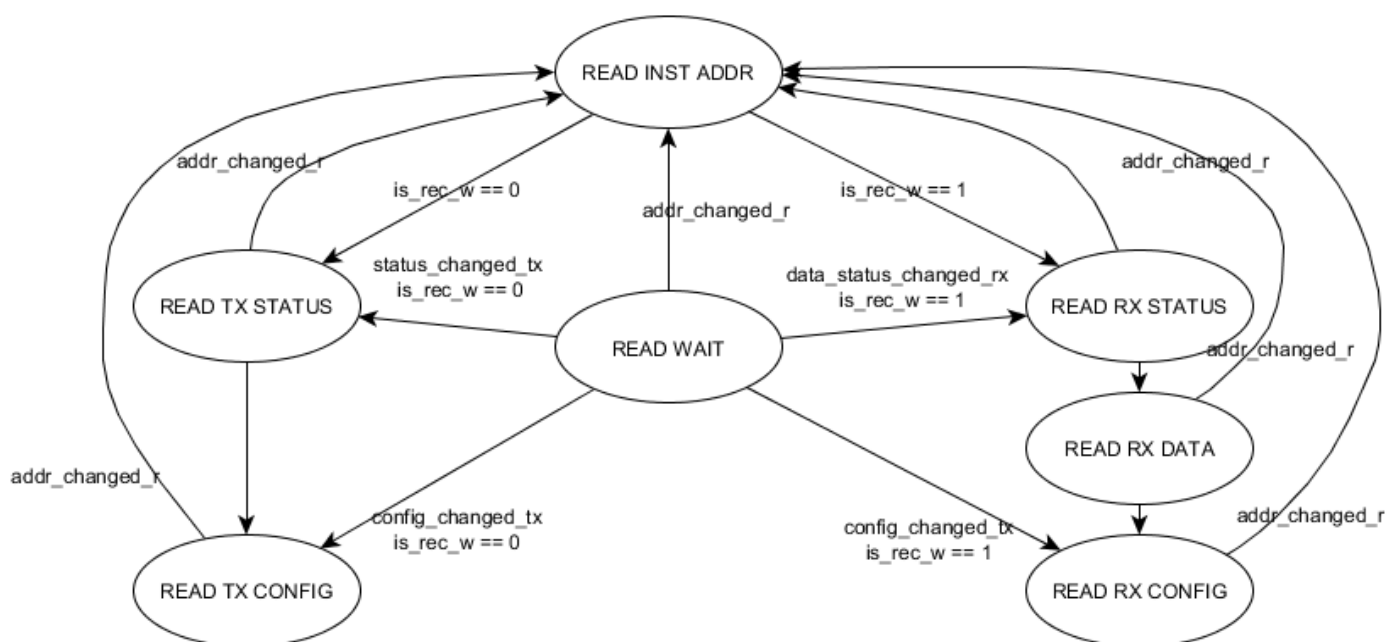


Рисунок 4. Конечный автомат транзакций чтения регистров

# SLTransmitter

## Описание

Данный проект подразумевает реализацию RTL-описания на языке Verilog одноканального передатчика SL-канала

## 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.

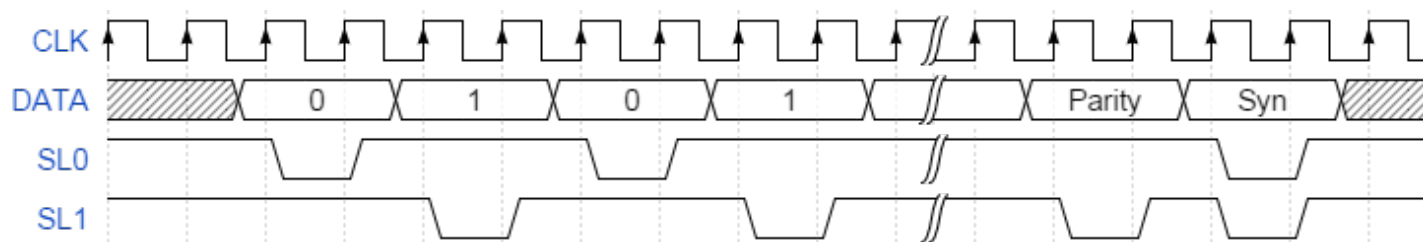


Рисунок 5. Временная диаграмма SL-канала

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

## Описание верхнего уровня передатчика

### Входные сигналы

- `rst_n` - асинхронный общий сигнал сброса
- `clk` - сигнал тактовой частоты
- `[31:0] D_in` - порт для записи данных в регистры
- `addr` - порт адреса регистра. "0" - регистр данных, "1" - регистр конфигурации и состояния.
- `wr_en` - После установки в 1 в выбранный портом `addr` регистр записывается необходимое число

## Выходные сигналы

- `SL0` - сигнал нулей SL канала

- SL1 - сигнал единиц SL канала
- [31:0] D\_out - порт для чтения регистров
- irq - вывод прерывания

Двунаправленные сигналы

Отсутствуют.

## Программная модель

Пользователю для работы доступно несколько регистров:

- Регистр конфигурации и состояния (config\_r и status\_r)
- Данных к отправке (txdata\_r)

Регистр конфигурации и состояния

Регистр конфигурации и состояния состоит из двух объединенных регистров - конфигурации и состояния. Регистру конфигурации соответствуют младшие 16 разрядов, регистру состояния - старшие.

*Таблица 4. Назначение разрядов регистра конфигурации (config\_r)*

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
SR	BC[6:0]						FQM[9:7]			IRQM				Res*	Res*

*Описание разрядов регистра конфигурации (config\_r)*

1. SR - soft reset, включает (SR=1) и выключает (SR=0) приемник
2. BC - bit count, количество бит данных в отправляемом сообщении
3. IRQM - interrupt request mask, задает, какие именно биты причин прерываний вызывают запрос прерывания
4. FQM - frequency mode, соответствие значения FQM и делителя частоты описано в таблице ниже.

*Таблица 5. Связь значения FQM и частоты работы передатчика*

Значение FQM в десятичной системе	Делитель частоты
1	2
2	4
3	8
4	16
5	32
>5	32

*Таблица 6. Связь разрядов IRQM и маскирования причин прерываний*

Разряд поля IRQM	Маскируемый бит
IRQM[0]	IRQSM
IRQM[1]	IRQWCC
IRQM[2]	IRQICC
IRQM[3]	IRQDWE

*Таблица 7. Назначение разрядов регистра состояния (status\_r)*

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----



SIP	Res*	Res*	Res*	Res*	Res*	Res*	Res*	IRQS M	IRQC C	IRQIC	IRQD WE	Res*	Res*	Res*	Res*
-----	------	------	------	------	------	------	------	-----------	-----------	-------	------------	------	------	------	------

Описание разрядов регистра состояния (status\_r)

1. SIP - send in process, сообщение отправляется, при попытке перезаписи этого бита ничего не происходит
2. IRQSM - interrupt request of sent message, запрос прерывания успешно отправленного сообщения
3. IRQWCC - interrupt request of wrong configuration change, запрос прерывания попытки сменить конфигурацию во время отправки сообщения
4. IRQICC - interrupt request of incorrect configuration change, запрос прерывания попытки установить неверную конфигурацию
5. IRQDWE - interrupt request of data write error, запрос прерывания попытки записать сообщение во время отправки предыдущего

Регистр данных к отправке

txdata\_r[31:0]

Таблица 8. Назначение разрядов регистра данных к отправке (txdata\_r)

0 - 31
Data

Data - данные к отправке.

## Описание работы

Модуль 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).

### Запись и чтение регистров

Управление модулем осуществляется путем записи/чтения регистров.

Для считывания текущего значения одного из регистров блока необходимо сформировать на шине addr соответствующее ему значение, указанное в таблице, длительностью не меньше такта опорной тактовой частоты. Значение регистра будет сформировано на шине d\_out через такт опорной после фронта сигнала на шине addr.

Для записи значения в один из регистров блока необходимо сформировать:

- на шине addr значение соответствующее регистру
- на шине d\_in записываемую информацию,
- на порт wr\_en - значение "1".

Также на на шине d\_out через такт опорной 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.

*Таблица 9. Адреса регистров*

Значение шины addr	Выбранный регистр
1'b0	регистр данных
1'b1	регистр конфигурации и состояния

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

### Смена конфигурации

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.

### Отправка сообщений

Для отправки сообщения необходимо записать отправляемое сообщение в регистр данных. Сразу после записи модуль переходит в режим отправки сообщения. При этом поле SIP регистра состояния устанавливается в "1".

В случае когда поле ВС регистра конфигурации не равно 32, отправляемым сообщением являются младшие биты регистра данных. Старшие биты регистра, которые не входят в длину сообщения заданную конфигурацией ( txdata[31:32-BC] ) будут записаны в регистр, но игнорированы при отправке.

В конце отправки будет выставлен бит IRQSM = 1, и бит SIP = 0.

## **Прерывания**

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 сбрасывания прерываний, вам необходимо считать регистр конфигурации и состояния и записать считанное снова, занулив биты прерываний. Более подробно работа прерываний рассмотрена в разделе Алгоритм работы.

## **Выключение модуля**

Чтобы выключить модуль необходимо выставить поле регистра конфигурации SR = "1". 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.

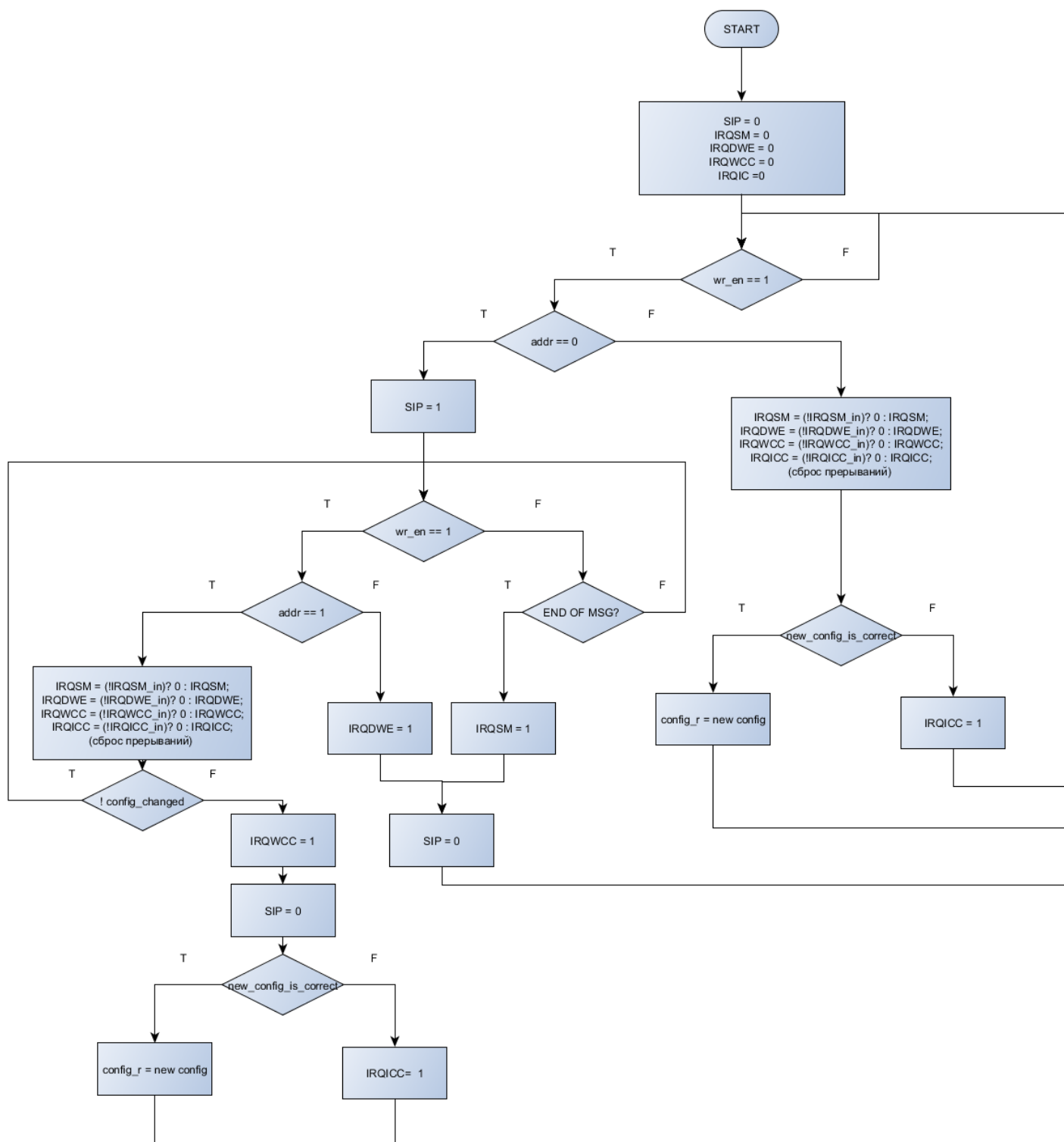


Рисунок 6. Алгоритм работы регистра состояния модуля SITransmitter

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

Чтобы сменить конфигурацию, необходимо записать новую конфигурацию в регистр

конфигурации и состояния. При записи регистра конфигурации и состояния в режиме ожидания происходит проверка битов прерываний: если значения полей причин прерываний шины D\_in равны 0, то соответствующие поля причин прерываний регистра состояния сбрасываются. После этого если конфигурация некорректна, выставляется  $IRQICC = 1$ , конфигурация не изменяется. Если бит  $IRQICC$  не замаскирован формируется запрос на прерывание. Если конфигурация корректна она записывается в регистр. Модуль остается в режиме ожидания.

### **Отправка сообщения**

Чтобы отправить сообщение, необходимо записать информацию для отправки в регистр данных, когда модуль находится в режиме ожидания. Сразу после записи данных в режиме ожидания, модуль переходит в режим отправки, устанавливается поле регистра состояний  $SIP = 1$ . По успешному окончанию отправки сообщения, устанавливаются поля регистра состояний  $SIP = 0$ , 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, выставляются 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.

### **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.

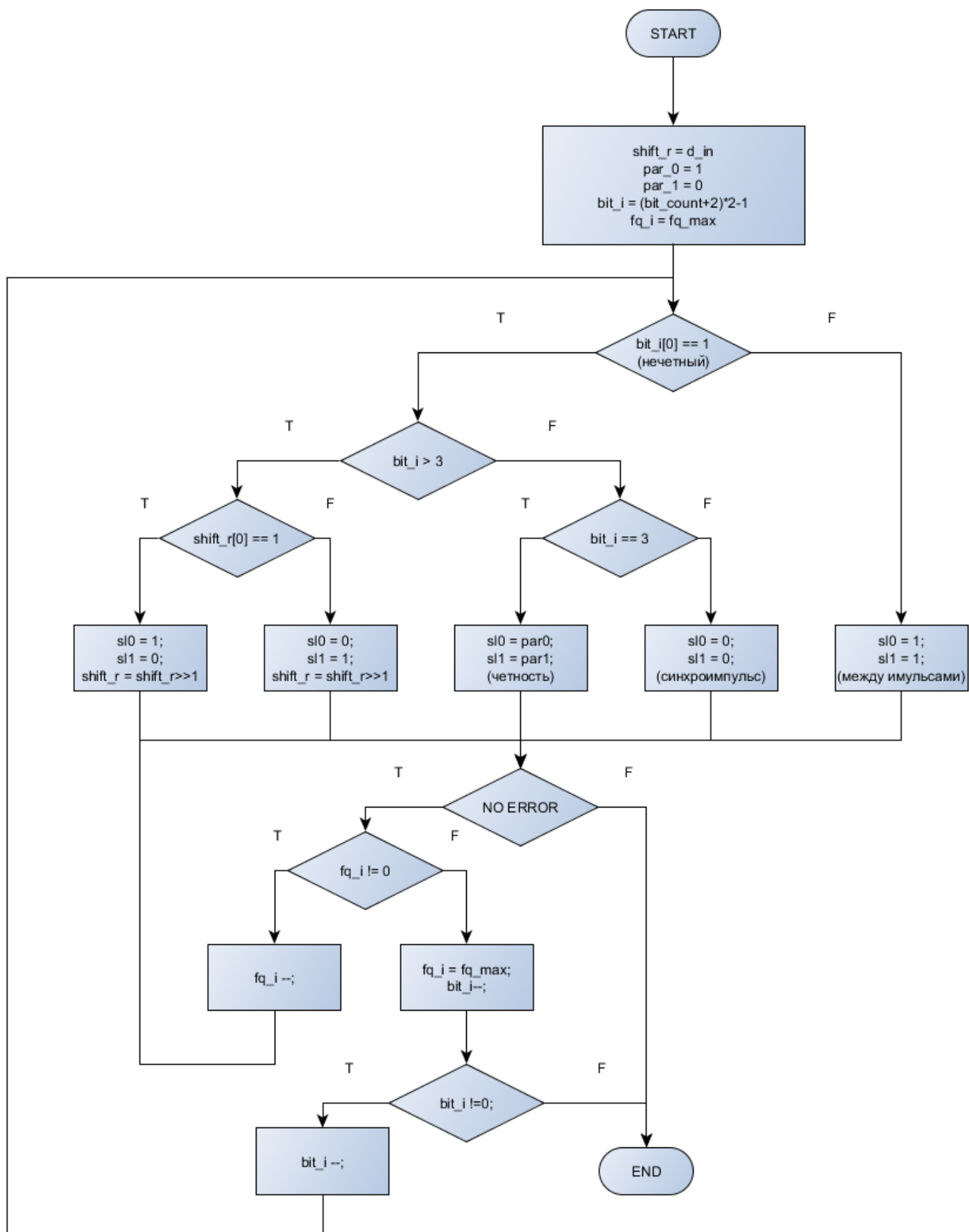


Рисунок 7. Алгоритм работы модуля SlTransmitter в режиме отправки

В начале режима отправки в сдвиговый регистр загружается отправляемое сообщение. Для

организации отправки используются два счетчика:

### **Счетчик частоты**

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

### **Счетчик количества бит**

Счетчик импульсов считает от числа, равного  $(BC + 2) * 2 - 1$  до 0, где BC (bit count) - поле регистра конфигурации. Число  $(BC + 2) * 2 - 1$  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  $(BC+2) * 2$  итераций счетчика и максимальное значение равно  $(BC+2) * 2 - 1$ .

### **Соответствие значения счетчика количества бит и значения на выходах sl0 и sl1**

На значениях счетчика битов  $(BC+2) * 2 - 1$ ,  $(BC+2) * 2 - 3$  .. 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  $(BC+1) * 2$ ,  $(BC+1) * 2 - 2$ , .. 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.

### **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.

# SIReceiver

## Описание

Данный проект подразумевает реализацию RTL-описания на языке Verilog одноканального приемника SL-канала.

## Описание верхнего уровня

### Входные сигналы

- `rst_n` - асинхронный общий сигнал сброса
- `clk` - сигнал тактовой частоты
- `[31:0] D_in` - порт для записи данных в регистры
- `wr_en` - После установки в 1 в выбранный портом `addr` регистр записывается необходимое число
- `addr` - адрес регистра "0" - регистр данных, "1" - регистр конфигурации и состояния
- `serial_line_zeroes_a` - асинхронный вход линии нулей SL-канала
- `serial_line_ones_a` - асинхронный вход линии единиц SL-канала

### Выходные сигналы

- `[31:0] D_out` - порт для чтения регистров
- `irq` - вывод прерывания

### Двунаправленные сигналы

Отсутствуют.

## Программная модель

Пользователю для работы доступно несколько регистров:

- Регистр конфигурации и состояния (`config_r` и `status_r`)
- Регистр последнего успешно принятого сообщения (`buffered_data_r`)

### Регистр конфигурации и состояния

Регистр конфигурации и состояния состоит из двух объединенных регистров - регистра конфигурации и регистра состояния. Регистру конфигурации соответствуют младшие 16 разрядов, регистру состояния - старшие.

*Таблица 10. Назначение разрядов регистра конфигурации (`config_r`)*

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
SR	BC[6:0]						PCE	IRQM						Res*	Res*



### Описание разрядов регистра конфигурации (config\_r)

1. SR - soft reset, включает (SR=1) и выключает (SR=0) приемник
2. BC - bit count, количество бит в слове
3. IRQM - interrupt request mask, маскирование прерываний модуля
4. PCE - parity check enable, разрешение контроля четности(PCE = 1), или запрещение(PCE = 0)

Таблица 11. Связь разрядов IRQM и маскирования причин прерываний

Разряд поля IRQM	Маскируемый бит
IRQM[0]	IRQRM
IRQM[1]	IRQPEM
IRQM[2]	IRQWLC
IRQM[3]	IRQLE
IRQM[4]	IRQWCC
IRQM[5]	IRQICC

Таблица 12. Назначение разрядов регистра состояния (status\_r)

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
WRP	PEF	Res*	Res*	Res*	Res*	Res*	Res*	Res*	IRQRM	IRQPEM	IRQWLC	IRQLE	IRQWCC	IRQICC	Res*

### Описание разрядов регистра состояния (status\_r)

1. WRP - word receiving process, флаг идущего процесса приема слова по SL-каналу
2. PEF - parity error flag, присутствует ли ошибка четности в хранящемся в буфере сообщении
3. IRQRM - interrupt request of recieved message - прерывание успешно принятого сообщения
4. IRQPEM - interrupt request of parity error message, принято слово не прошедшее проверку четности
5. IRQWLC - interrupt request of word length check, принято слово не прошедшее проверку длины полученного слова на равенство значению BC регистра config\_r
6. IRQLE - interrupt request of level error on line, прерывание ошибки уровня напряжения на линии SL-канала
7. IRQWCC - interrupt request of wrong configuration changed - прерывание смены конфигурации во время приема сообщения
8. IRQICC - interrupt request of incorrect configuration прерывание попытки смены конфигурации на неверную
9. Res\* - Зарезервированно

Регистр полученных данных

buffered\_data\_r[31:0]

Таблица 13. Назначение разрядов регистра полученных данных (buffered\_data\_r)

0 - 31
Data

Data - данные к отправке.

Описание работы

Модуль принимает SL-сообщения. Сообщения могут иметь четную длину от 8 до 32 бит. Бит четности проверяется автоматически. Частота импульсов принимаемых сообщений может меняться от 500кГц до 2МГц (при частоте тактового сигнала = 16МГц).

### Запись и чтение регистров

Управление модулем осуществляется путем записи/чтения регистров.

Для считывания текущего значения одного из регистров блока необходимо сформировать на шине addr соответствующее ему значение, указанное в таблице, длительностью не меньше такта опорной тактовой частоты. Значение регистра будет сформировано на шине d\_out через такт опорной после фронта сигнала на шине addr.

Для записи значения в один из регистров блока необходимо сформировать:

- на шине addr значение соответствующее регистру
- на шине d\_in записываемую информацию,
- на порт wr\_en - значение "1".

Также на на шине d\_out через такт опорной 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.

*Таблица 14. Адреса регистров*

Значение шины addr	Выбранный регистр
1'b0	регистр данных
1'b1	регистр конфигурации и состояния

### Смена конфигурации

Для изменения конфигурации приемника необходимо перезаписать регистр конфигурации и состояния. В конфигурационной части может быть установлена 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.

### Прием сообщений

Если на вход модуля начинают поступать импульсы, модуль переходит в режим приема

сообщения, выставляется бит  $WRP = 1$ .

Модуль переходит в режим ожидания нового сообщения в ситуациях:

- Успешного приема сообщения
- Приема сообщения с ошибкой
- Завершившейся ошибки уровня на линии
- Попытки изменить конфигурацию модуля во время приема сообщения

Успешным приемом сообщения называется прием сообщения с совпадающим со значением поля  $BC$  количеством информационных бит и, если включен контроль четности, верной четностью.

В случае, если успешно принято слово с правильной четностью выставляются биты  $IRQRM = 1$  и  $WRP = 0$ . Если контроль четности отключен и принято слово с неправильной четностью, выставляются биты  $IRQPEM = 1$ ,  $PEF = 1$  и  $WRP = 0$ .

В случае приема сообщения с ошибкой выставляются биты:

- Контроль четности включен и принято сообщение с ошибкой четности —  $IRQPEM = 1$  и  $WRP = 0$
- Принято сообщение с несовпадающим с конфигурацией количеством бит —  $IRQWLC = 1$  и  $WRP = 0$

В случае, если во время приема произошла ошибка уровня, выставляется флаг  $IRQLEF = 1$ . Модуль вернется в режим ожидания сообщения только когда уровень на линиях будет восстановлен. До этого момента будет флаг  $WRP = 1$ , а бит причины прерывания  $IRQLEF$  будет невозможно сбросить.

В случае, если во время приема произошла попытка изменить поля  $PCE$  и  $BC$  регистра конфигурации, выставляются биты  $IRQWCC = 1$  и  $WRP = 0$ . Если новая конфигурация верна, она записывается в регистр. Если новая конфигурация неверна, выставляется бит  $IRQICC$ , поля  $PCE$  и  $BC$  остаются неизменными.

В регистре данных всегда хранится последнее успешно принятое сообщение. А в поле  $PEF$  регистра состояния - наличие ошибки четности последнего успешно принятого сообщения.

После считывания сообщения необходимо сбросить возникшие биты причин прерываний, и ожидать приема следующего сообщения.

## Прерывания

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

- Успешно принято сообщение ( $IRQRM$ )
- Принято сообщение с ошибкой четности ( $IRQPEM$ )
- Принято сообщение неверной длины ( $IRQWLC$ )
- Произошла ошибка уровня на линии ( $IRQLE$ )
- Была предпринята попытка записать некорректные данные в конфигурационный регистр

(IRQICC)

- Изменение конфигурации в процессе отправки сообщения (IRQWCC)

Причину возникновения можно посмотреть в соответствующих полях регистра состояния. Для сбрасывания прерываний, вам необходимо считать регистр конфигурации и состояния и записать считанное снова, занулив биты прерываний. Более подробно работа прерываний рассмотрена в разделе Алгоритм работы.

### **Выключение модуля**

Чтобы выключить модуль необходимо выставить поле регистра конфигурации SR = "1". Если сделать это во время отправки сообщения, прием сообщения прекращается. Регистры конфигурации и состояния возвращаются в начальное состояние. Когда приемник выключен, он не реагирует на сигналы на входах SL0 и SL1.

Алгоритмы работы

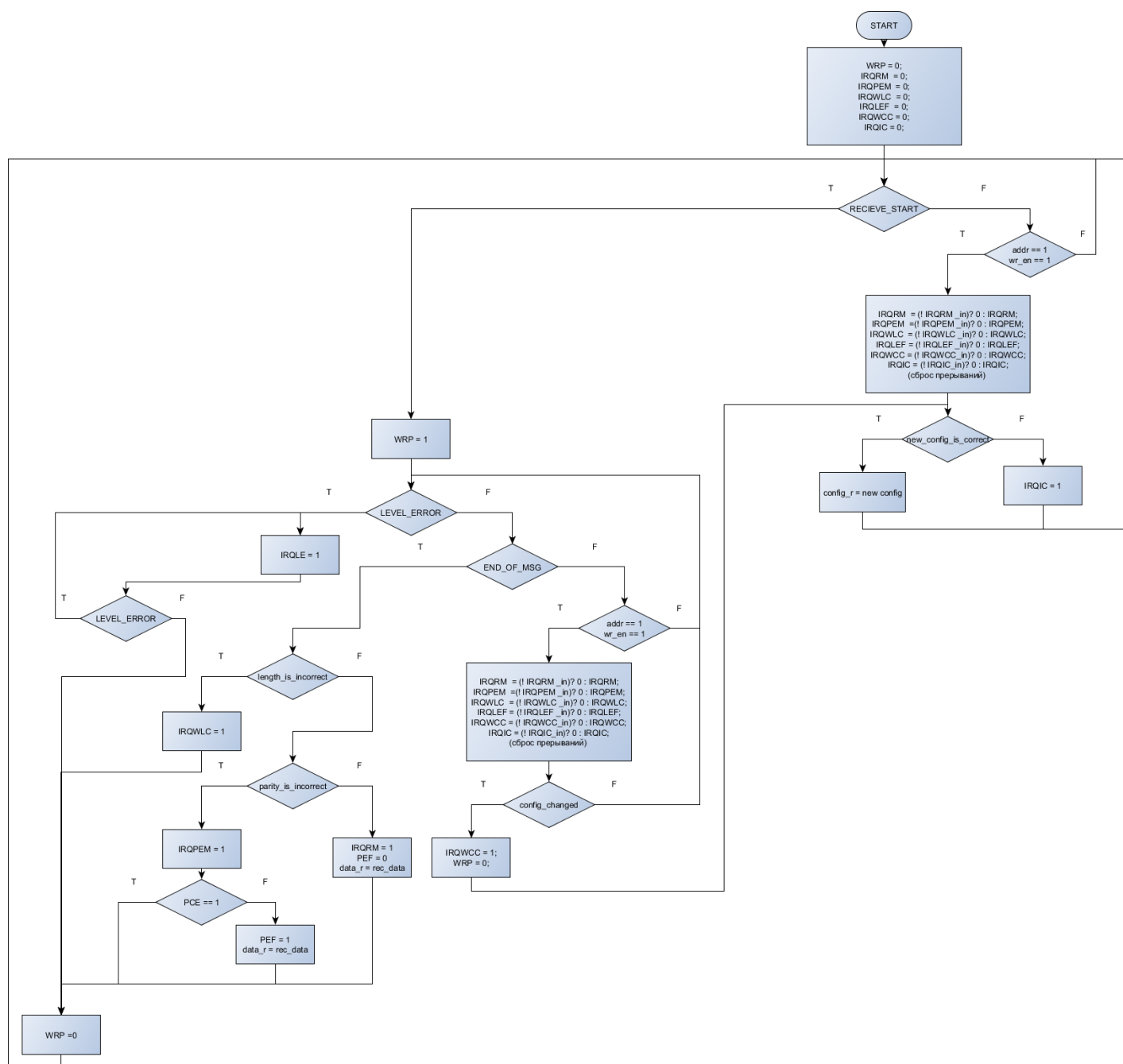


Рисунок 8. Алгоритм работы регистра состояния модуля SIReceiver

Модуль может находиться в двух режимах: режим приема 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

Чтобы сменить конфигурацию, необходимо записать новую конфигурацию в регистр конфигурации и состояния. При записи регистра конфигурации и состояния в режиме ожидания происходит проверка битов причин прерываний: если значения соответствующих записываемых битов прерываний равны 0, то они сбрасываются.

Если конфигурация некорректна, выставляется IRQICC = 1, конфигурация не изменяется. Если бит IRQICC не замаскирован формируется запрос на прерывание.

Если конфигурация корректна она записывается в регистр. Модуль остается в режиме ожидания.

### **Прием сообщения**

Если на одной из линий возникает импульс, модуль переходит в режим приема, устанавливается поле регистра состояний  $WRP = 1$ . Если импульс слишком короткий или слишком длинный, возникает ошибка уровня, выставляется бит  $IRQLE = 1$ . Бит выставляется каждый такт, пока уровень на линии не будет восстановлен. После этого модуль возвращается в режим отправки сообщения, выставляется бит  $WRP = 0$ .

Если импульс является синхроимпульсом модуль выставляет соответствующие принятому сообщению биты статусного регистра и, если сообщение принято успешно, запоминает сообщение в регистр данных, выставляется бит  $WRP = 0$ .

### **Анализ принятого сообщения**

При приеме синхроимпульса (условие  $END\_OF\_MSG$  на рис. 1) сначала проверяется длина принятого сообщения, если длина не совпадает с конфигурацией (значение поля  $BC + 1$  за счет бита четности) выставляется  $IRQWLC = 1$ . Если бит  $IRQWLC$  не замаскирован, формируется запрос на прерывание.

Если длина совпадает с выставленной в конфигурации, проверяется четность полученного сообщения. Если четность верна, сообщение считается успешно принятым, выставляется  $IRQRM = 1$ , содержимое сдвигового регистра с удаленным битом четности записывается в регистр данных. Если бит  $IRQRM$  не замаскирован, формируется запрос на прерывание.

Если четность неверна, выставляется бит  $IRQPEM = 1$ . Однако, если контроль четности отключен, сообщение все равно считается успешно принятым, данные сдвигового регистра с удаленным битом четности переписываются в регистр данных. Выставляется бит  $PEF = 1$ . Если бит  $IRQPEM$  не замаскирован, формируется запрос на прерывание.

### **Дублирование битов проверки честности**

Бит причины прерывания  $IRQPEM$  и бит четности  $PEF$  дублируют функции друг друга. Тем не менее эта система необходима для разрешения следующего конфликта: Допустим с приемником с отключенным контролем четности, и за время, прошедшее с последнего опроса пришло 2 сообщения, одно с верной четностью, а другое с ошибкой. Тогда флаг  $PEF$  - единственный способ определить, верна ли честность сообщения, лежащего в регистре данных.

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

Когда модуль находится в режиме приема сообщения, то без отмены приема возможно только изменение полей маскирования прерываний, и сброс битов причин прерываний. Если изменить длину сообщения в середине приема сообщения, прием сообщения будет отменен, а остаток сообщения будет воспринят как новое сообщение неправильной длины.

Если в режиме отправки происходит запись регистра конфигурации и состояния, сначала проверяются биты прерываний: если значения соответствующих записываемых битов прерываний равны 0, то они сбрасываются. После этого, проверяется изменяются ли биты

конфигурации (поля PCE, BC). Если они не изменяются, модуль остается в режиме приема сообщения. Если они изменяются то прием завершается, выставляются биты WRP = 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 сообщения.

### **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.

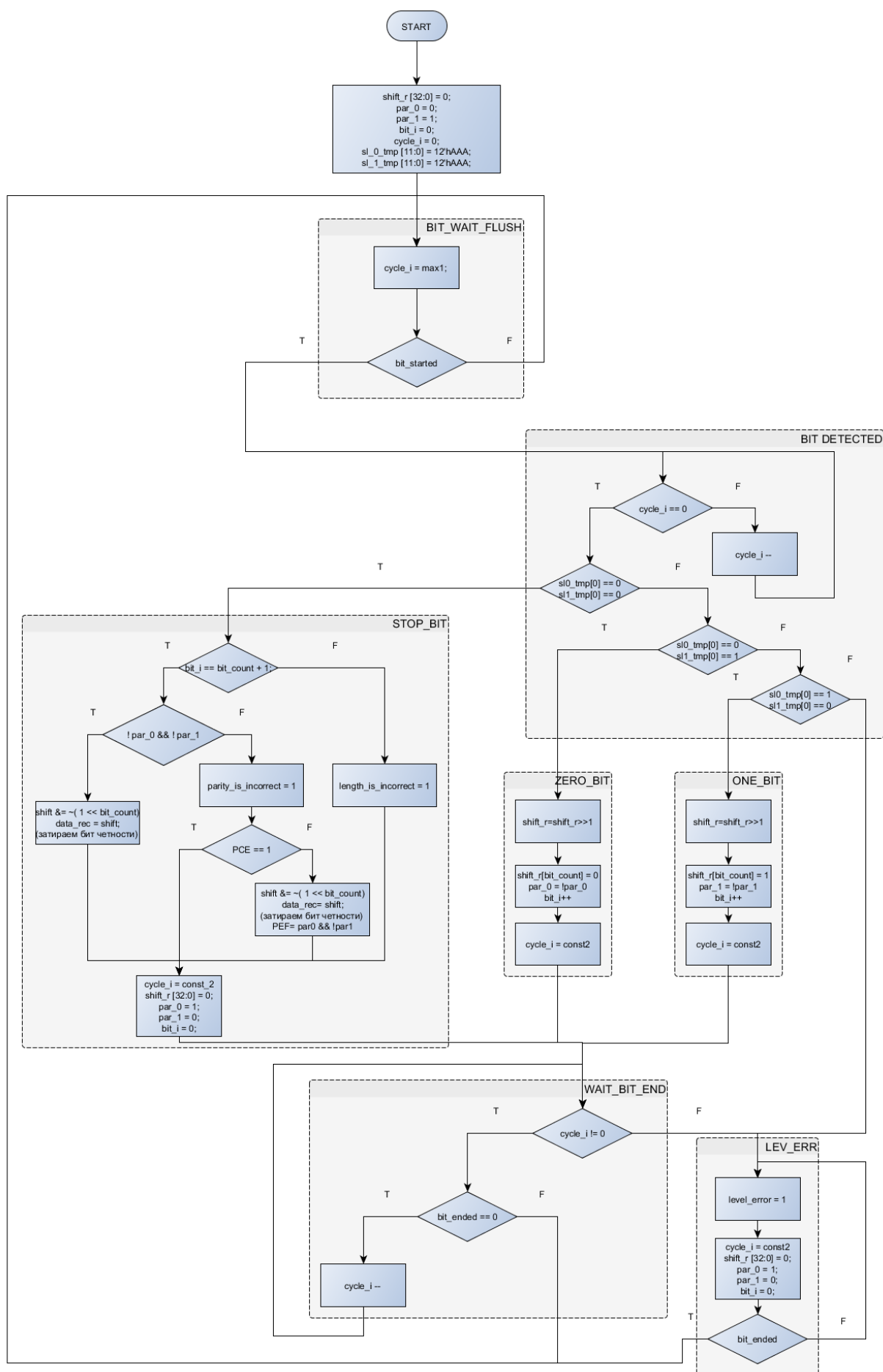


Рисунок 9. Алгоритм работы приема сообщения модуля SIReceiver



После включения приемника сдвиговый регистр приема сообщения shift\_r заполняется нулями, Сдвиговых регистры sl\_0\_tmp и sl\_1\_tmp - единицами, счетчик количества бит bit\_i устанавливается в 0, счетчик циклов cycle\_i устанавливается в 0, регистры контроля четности par\_0 и par\_1 устанавливаются в 0 и 1 соответственно.

Каждый такт значение с асинхронных входов serial\_line\_zeroes\_a и serial\_line\_ones\_a помещаются в нулевые разряды сдвиговых регистров sl\_0\_tmp и sl\_1\_tmp. Остальные разряды при этом сдвигаются. Условия bit\_started, и bit\_ended получаются при сравнении содержимого sl\_0\_tmp и sl\_1\_tmp с масками.

*Таблица 15. Условия переходов*

Обозначение	Выражение
bit_started	(sl0_tmp_r == 12'hF??0)    (sl1_tmp_r == 12'hF??0)
bit_ended	(sl0_tmp_r == 12'h0??F)    (sl1_tmp_r == 12'h0??F)

*Таблица 16. Значения констант счетчика cycle\_i*

Обозначение	Значение
const_1	3
const_2	32

### **Состояние BIT\_WAIT\_FLUSH**

В начале приема машина состояний находится в состоянии BIT\_WAIT\_FLUSH. В этом состоянии счетчик циклов приравнивается к константе const1. Если выполняется условие bit\_started, происходит переход в состояние BIT\_DETECTED.

### **Состояние BIT\_DETECTED**

В состоянии BIT\_DETECTED работает счетчик циклов cycle\_i. Как только этот счетчик становится равным до 0, производится анализ, какой именно бит принят, и в зависимости от значения первых разрядов сдвиговых регистров sl\_0\_tmp и sl\_1\_tmp определяется, на какую из линий поступил импульс и происходит переход в состояние обработки определенного бита: ONE\_BIT, ZERO\_BIT или STOP\_BIT. При этом, если импульс отсутствует, происходит переход в состояние LEV\_ERR.

### **Состояния ONE\_BIT и ZERO\_BIT**

В состояниях ONE\_BIT и ZERO\_BIT соответствующее значение 0 или 1 загружаются в бит с номером BC сдвигового регистра shift\_r. Сам сдвиговый регистр при этом сдвигается вправо. В состоянии ONE\_BIT инвертируется значение бита четности единиц par\_1, в состоянии ZERO\_BIT - значение par\_0. В регистр cycle\_i помещается значение const2. Счетчик принятых bit\_i инкрементируется.

### **Состояние STOP\_BIT**

В состоянии STOP\_BIT происходит анализ полученного сообщения. Сравняется количество принятых бит с установленным в конфигурации, Проверяется четность. Если количество бит и четность верны, или, если верно количество бит, контроль четности отключен и не верна четность, значение из сдвигового регистра shift\_r переписывается в регистр данных с обнулением

бита четности shift\_r[BC].

Для приема следующего сообщения регистры shift\_r и bit\_i устанавливаются в нулевые значения. В регистры подсчета четности загружаются значения par\_0 = 1, par\_1 = 0. В регистр cycle\_i помещается значение const2.

### **Особенности контроля честности**

Считая бит честности, количество импульсов на линии единиц с учетом разряда четности должно быть нечетным, а на линии нулей - четным.

Для проверки этого, до приема сообщения в регистры подсчета четности загружаются значения par\_0 = 0, par\_1 = 1. При принятии единицы меняет значение на противоположное регистр par\_1, а при принятии нуля — par\_0.

Таким образом, после принятия всех бит корректного сообщения (считая бит четности), регистр par\_0 должен поменять свое значение четное количество раз, т.е. сохранить значение par\_0 = 0, а регистр par\_1 свое значение нечетное количество раз, т.е. приобрести значение par\_1 = 0.

При обработке стоп бита считается, что четность нарушена, если хотя бы один из регистров par\_0 и par\_1 не равен нулю.

### **Состояние WAIT\_BIT\_END**

После обработки импульса в состояниях ONE\_BIT, ZERO\_BIT или STOP\_BIT, схема переходит в состояние WAIT\_BIT\_END. Модуль находится в этом состоянии, пока счетчик cycle\_i не достиг нулевого значения, или не выполниться условие bit\_ended.

Если выполнилось условие bit\_ended, модуль возвращается в состояние BIT\_WAIT\_FLUSH. Если же счетчик досчитал до нулевого значения, это значит, что импульс не закончился вовремя, и произошла ошибка уровня на линии - модуль переходит в состояние LEV\_ERR.

### **Состояние LEV\_ERR**

Модуль оказывается в состоянии LEV\_ERR в случаях, когда длинна импульса оказалось слишком большой или слишком маленькой, т.е. произошла ошибка уровня на линии. Когда уровень на обеих линиях восстановлен, модуль переходит в состояние BIT\_WAIT\_FLUSH. Для приема следующего сообщения регистры shift\_r и bit\_i устанавливаются в нулевые значения. В регистры подсчета четности загружаются значения par\_0 = 0 и par\_1 = 1.

## Описание SL-канала

SL - канал - последовательный однонаправленный канал обмена данными, разработанный для внутриплатного и межплатного 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 произвольной разрядности 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.

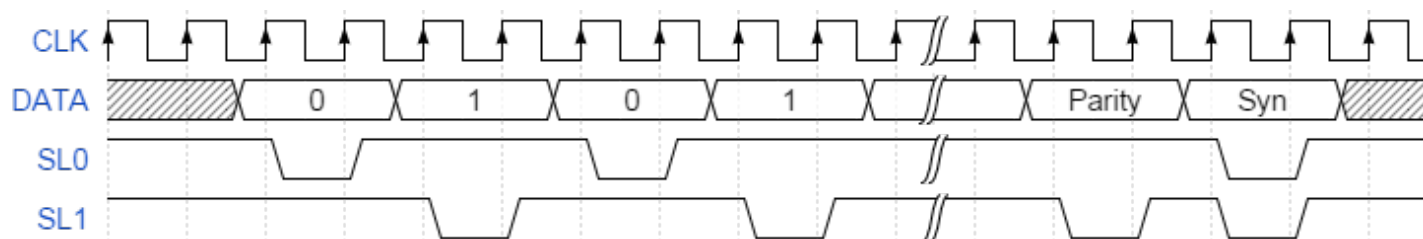


Рисунок 10. Временная диаграмма SL-канала

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

## Описание верхнего уровня

### Входные сигналы

#### Общие сигналы

- rst\_n - асинхронный общий сигнал сброса
- clk - сигнал тактовой частоты

#### АРВ-связанные сигналы

- pclk - сигнал тактовой частоты
- preset\_n - асинхронный сигнал сброса
- [15:0] paddr - асинхронная шина адреса
- psel - асинхронный сигнал выбора устройства
- penable - асинхронный сигнал разрешения работы
- pwrite - асинхронный сигнал выбора чтения или записи
- [31:0] pwdata - асинхронная шина данных

### SL-связанные сигналы

- SL0\_in - асинхронный сигнал линии нулей
- SL1\_in - асинхронный сигнал линии единиц

### Выходные сигналы

### APB-связанные сигналы

- [31:0] prdata - шина данных
- pready - асинхронный сигнал готовности к приему или передаче данных

### SL-связанные сигналы

- SL0\_out - синхронный сигнал линии нулей
- SL1\_out - синхронный сигнал линии единиц

### Двунаправленные сигналы

Отсутствуют.

### Тактирование сигналов

Таблица 17. Указание на источник тактирования входных и выходных сигналов

Сигнал	Направленность	Клоковый домен
rst_n	in	async
SL0_in	in	async
SL1_in	in	async
preset_n	in	async
[15:0]paddr	in	pclk
psel	in	pclk
penable	in	pclk
pwrite	in	pclk
[31:0]pwwdata	in	pclk
[31:0]prdata	out	pclk
pready	out	pclk
SL0_out	out	clk
SL1_out	out	clk

### Программная модель

Пользователю для работы доступно несколько регистров:

1. Конфигурационный
2. Состояния

3. Данных

4. Номера управляемого устройства (приемника или передатчика)

## Регистр конфигурации

Назначение битов регистра конфигурации различается в зависимости от того является ли управляемое устройство приемником или передатчиком (значение первого бита регистра адреса устройства). Назначение разрядов регистра `config_r` в режиме приемника

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PCE	BC[5:0]						MODE	IRQM	Res*	Res*	Res*	Res*	Res*	Res*	Res*
							E								

### Описание разрядов регистра `config_r` в режиме приемника

1. PCE - parity check enable, разрешение контроля четности(PCE = 1), или запрещение(PCE = 0)
2. BC - bit count, количество бит в слове
3. MODE - выбор режима работы модуля в качестве приемника(MODE = 0), или передатчика(MODE = 1)
4. IRQM - interrupt request mode, разрешение(IRQM = 1) или запрещение(IRQM = 0) работы прерываний модуля

Таблица 18. Назначение разрядов регистра `config_r` в режиме передатчика

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
BC[6:0]						IRQM	FQM[9:7]			Res*	Res*	Res*	Res*	Res*	Res*

### Описание разрядов регистра `config_r` в режиме передатчика

1. BC - bit count, количество бит в слове
2. IRQM - interrupt request mode, разрешение(IRQM = 1) или запрещение(IRQM = 0) работы прерываний модуля
3. FQM - frequency mode, соответствие частот описано в таблице ниже

При ошибке указания количества бит в слове (нечетное или меньше восьми) попытка смены конфигурации будет игнорирована.

Таблица 19. Связь значения `FQM` и частоты работы передатчика

Значение FQM в десятичной системе	Частота, МГц
1	8
2	4
3	2
4	1
5	0.5
>5	0.5

## Регистр состояния

Назначение разрядов первого байта разрядов регистра состояния зависят от режима работы приемопередатчика. Второй байт регистров состояния содержит биты отвечающие за состояние приемопередатчика в целом

*Таблица 20. Назначение разрядов регистра status\_r в режиме приемника*

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
WLC	WRP	Res*	WRF	PEF	LEF	Res*	Res*	CBF	CBE	Res*	Res*	Res*	Res*	Res*	Res*

*Описание разрядов регистра status\_r в режиме приемника*

1. WLC - word length check, результат проверки длины полученного слова на равенство значению BC регистра config\_r, WLC = 1, если значения не равны
2. WRP - word receiving process, флаг идущего процесса приема слова по SL-каналу
3. Res\* - Зарезервированно
4. WRF - word received flag, флаг успешно завершеного приема слова
5. PEF - parity error flag, флаг наличия (PEF = 1) ошибки четности принятого слова
6. LEF - level error on line flag, флаг наличия ошибки уровня напряжения на линии SL-канала
7. CBF - control buffer is full буфер, куда записываются управляющие команды полон
8. CBE - control buffer is empty буфер, куда записываются управляющие команды пуст

*Таблица 21. Назначение разрядов регистра status\_r в режиме передатчика*

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
SIP	Res*	Res*	Res*	Res*	Res*	Res*	Res*	CBF	SBE	Res*	Res*	Res*	Res*	Res*	Res*

*Описание разрядов регистра status\_r в режиме передатчика*

1. SIP - send in process - передатчик в данный момент занят отправкой сообщения
2. CBF - control buffer is full буфер, куда записываются управляющие команды полон
3. CBE - control buffer is empty, куда записываются управляющие команды пуст

Описание особенностей работы управляющего буфера будет объяснено далее

## Регистр данных

data\_r[31:0] - регистр данных в котором находится последнее успешно принятое сообщение. В режиме передатчика при транзакции записи в регистр записываемое слово принимается к отправке, значение регистра не меняется.

В режиме приемника при транзакции чтения вы получите последнее принятое приемником сообщение. Транзакция записи в режиме приемника будет игнорирована.

## Регистр адреса устройства

Регистр адреса устройства. Количество устройств определяется количеством каналов (на каждый канал приходится один приемник и один передатчик). Приемники имеют нечетные адреса, передатчики четные. При попытке записать некорректное значение, попытка будет игнорирована.

## Конечный автомат



Рисунок 11. Конечный автомат транзакций записи в регистры