

# Спецификация модуля приемника SL-канала :SlReciever

[[rec-main-description]]

Версия 1.1, 15.12.2017

# Оглавление

- 1. Описание верхнего уровня ..... 1
- 2. Программная модель ..... 2
  - 2.1. Служебный регистр. .... 2
  - 2.2. Регистр полученных данных. .... 3
- 3. Описание работы ..... 3
- 4. Алгоритмы работы..... 6

Данный проект подразумевает реализацию RTL-описания на языке Verilog одноканального приемника SL-канала. Приемник принимает SL-сообщения. Сообщения могут содержать информацию четной разрядности от 8 до 32 бит. Бит четности проверится автоматически. Приемник способен принимать сообщения с частотой импульсов может меняться от 500кГц до 2МГц (при частоте тактового сигнала = 16МГц).

## 1. Описание верхнего уровня

*Таблица 1. Порты цифрового модуля SLReciever*

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>	In	1	b1	Сигнал нулей SL канала
<i>SL1</i>	in	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

## 2. Программная модель

Пользователю для работы доступно несколько регистров:

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

### 2.1. Служебный 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[4:0]</b>				<b>PCE</b>	<b>BC[6:0]</b>						<b>SR</b>	Mode	R
	R/W			R/W	R/W						R/W	Initial	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. **PCE** (parity check enable) — разрешение контроля четности (**PCE** = 1), или запрещение (**PCE** = 0)

**Таблица 3. Назначение разрядов части состояния служебного регистра (config\_status\_r [31:16])**

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*	IRQR M	IRQP EM	IRQW LC	IRQL EF	IRQW CC	IRQIC C	Res*

Bit	31-30	29	28	27	26	25	24
23-17	16	Name	-	<b>IRQICC</b>	<b>IRQWCC</b>	<b>IRQLE</b>	<b>IRQWLC</b>
<b>IRQPEM</b>	<b>IRQRM</b>	-	<b>SIP</b>	Mode	R	R/W0	R/W0
R/W0	R/W0	R/W0	R/W0	R	R	Initial	0
0	0	0	0	0	0	0	0

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

1. **WRP** - (word receiving process) — флаг идущего процесса приема слова по SL-каналу
2. **PEF** - (parity error flag) — флаг присутствия ошибки 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 ВС 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 change) — разряд запроса на прерывание попытки сменить конфигурацию во время отправки сообщения.
8. **IRQICC** (interrupt request of incorrect configuration change ) — разряд запроса на прерывание попытки установить некорректную конфигурацию.

*Таблица 4. Соответствие разрядов **IRQM [4:0]** и маскирования разрядов причин прерываний*

Разряд поля <b>IRQM</b>	Маскируемый разряд
<b>IRQM0</b>	<b>IRQRM</b>
<b>IRQM1</b>	<b>IRQPEM</b>
<b>IRQM2</b>	<b>IRQWLC</b>
<b>IRQM3</b>	<b>IRQLE</b>
<b>IRQM4</b>	<b>IRQWCC</b>
<b>IRQM5</b>	<b>IRQICC</b>

## 2.2. Регистр полученных данных

buffered\_data\_r[31:0]

*Таблица 5. Назначение разрядов регистра полученных данных (buffered\_data\_r)*

0 - 31
Data

Data - данные к отправке.

## 3. 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 6. 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 ВС 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  $SL1$ .

## 4. Алгоритмы работы

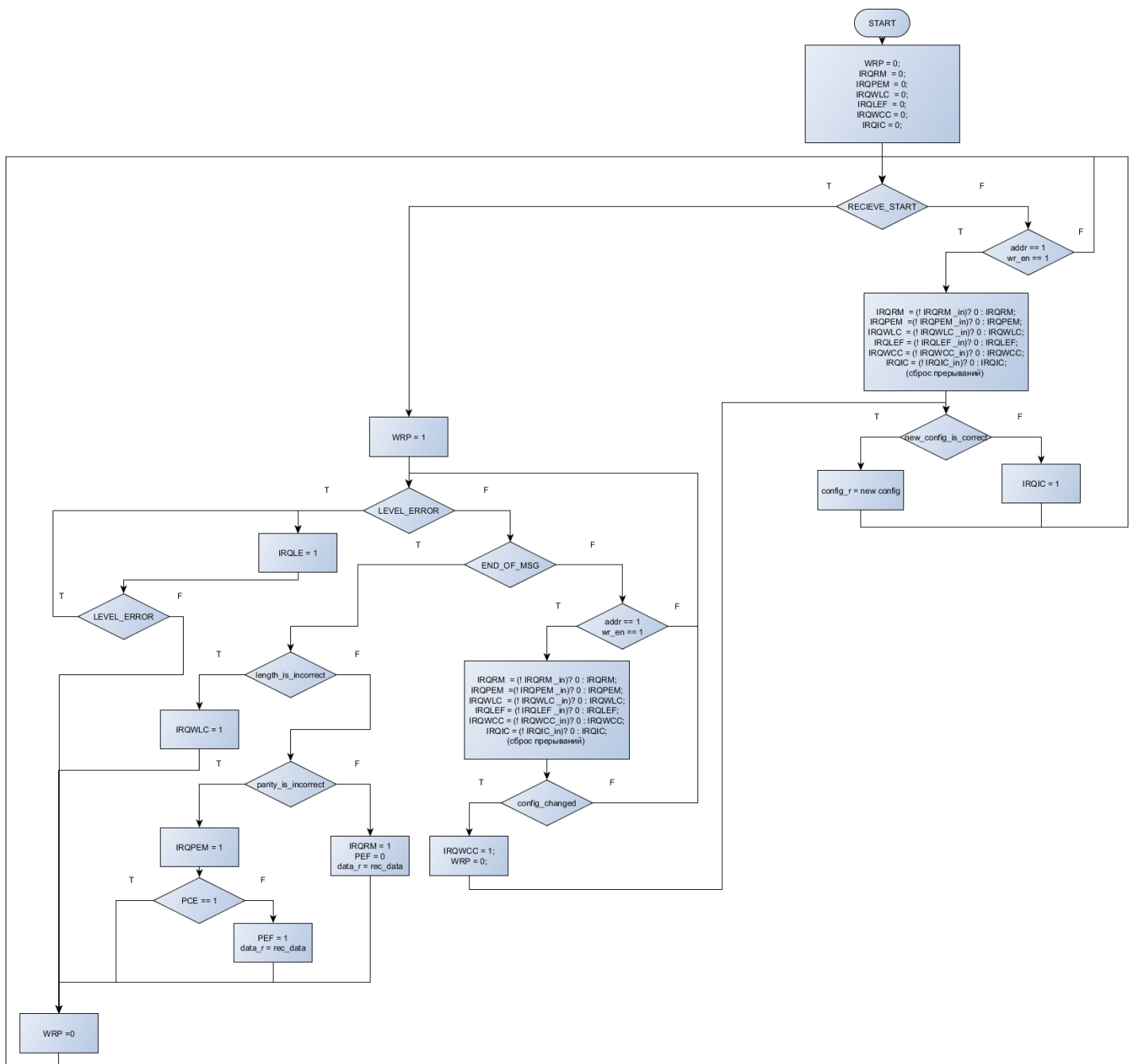


Рисунок 1. Алгоритм работы регистра состояния модуля SIReceiver

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 длинна принятого сообщения, если длинна 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 длинна 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 длинны.

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.

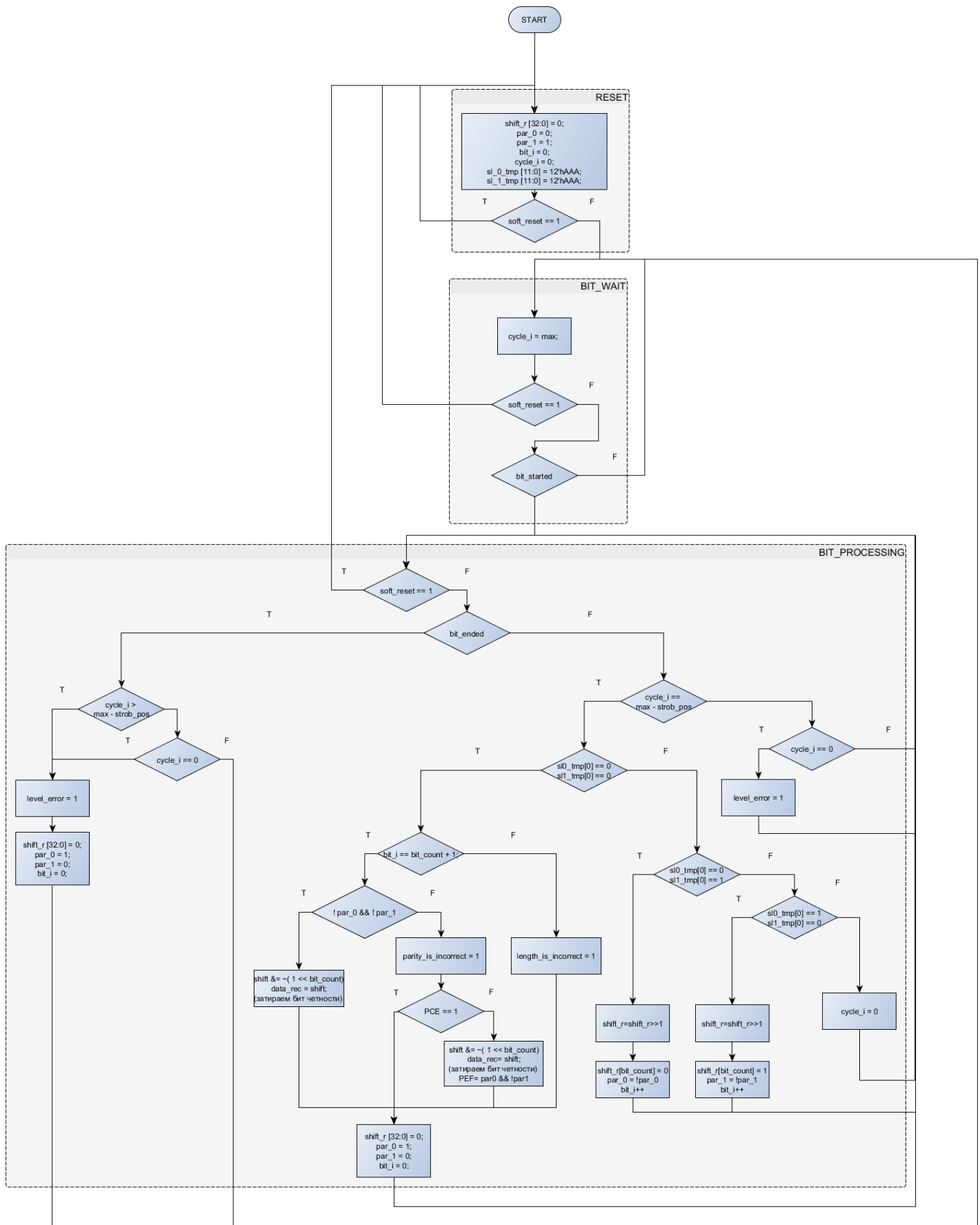


Рисунок 2. Алгоритм работы приема сообщения модуля SIReceiver

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 7. 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 8. 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 бита BIT\_PROCESSING. При этом, если импульс отсутствует, происходит переход в состояние LEV\_ERR.

#### **Состояние BIT\_PROCESSING**

В состоянии BIT\_PROCESSING проверяется, на какую из линий пришел импульс.

Если импульс на линии нулей или на линии единиц соответствующее значение 0 или 1 загружаются в бит с номером ВС сдвигового регистра shift\_r. Сам сдвиговый регистр при этом сдвигается вправо. В состоянии если импульс на линии единиц инвертируется значение бита четности единиц par\_1, а если на линии нулей - значение par\_0. В регистр cycle\_i помещается значение const2. Счетчик принятых bit\_i инкрементируется.

Если же импульсы на обеих линиях, то модуль считает, что это синхроимпульс. 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]`.

В случае синхро импульса 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 `const_2`.

### **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`.