

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. Работа с программной моделью	3
3.1. Запись и чтение регистров	3
3.2. Смена конфигурации	4
3.3. Прием сообщений	4
3.4. Работа с прерываниями	5
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 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>SL0</i>	In	1	b1	Сигнал нулей SL канала
<i>D_in</i>	In	32	-	Данные для записи в регистры
<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 два регистра:

- Служебный (**config_status_r**)
- Данных к отправке (**buffered_data_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	-	IRQM[5:0]						PCE	BC[5:0]						SR
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. 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](#)
4. **PCE** (parity check enable) — включение (**PCE** = 1) или выключение (**PCE** = 0) контроля четности

*Таблица 3. Назначение разрядов части состояния служебного регистра (**config_status_r [31:16]**)*

Bit	31-30	29	28	27	26	25	24	23-18	17	16
Name	-	IRQICC	IRQWCC	IRQLE	IRQWLC	IRQPEM	IRQRM	-	PEF	WRP
Mode	R	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	R	R	R
Initial	0	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) — разряд идущего процесса приема слова по SL-каналу.
2. **PEF** - (parity error flag) — разряд наличия ошибки четности в хранящемся в буфере сообщении.
3. **IRQRM** - (interrupt request of recieved message) — разряд запроса на прерывание успешно принятого сообщения.
4. **IRQPEM** - (interrupt request of parity error message) — разряд запроса на прерывание принятия слова не прошедшего проверку четности.

5. **IRQWLC** - (interrupt request of word length check) — разряд запроса на прерывание принятия слова неверной длины.
6. **IRQLE** - (interrupt request of level error on line) — разряд запроса на 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 приема сообщения.
8. **IRQICC** (interrupt request of incorrect configuration change) — разряд запроса на прерывание попытки установить некорректную конфигурацию.

Разряды **IRQRM**, **IRQPEM**, **IRQWL**, **IRQLE**, **IRQWCC** и **IRQICC** отражают зарегистрированные приемником события. Более подробно события описаны разделе Работа с программной моделью → [Работа с прерываниями](#).

*Таблица 4. Соответствие разрядов **IRQM** [5:0] и маскирования разрядов причин прерываний*

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

2.2. Регистр полученных данных

*Таблица 5. Назначение разрядов регистра полученных данных (**buffered_data_r**)*

Bit	31 - 0
Name	DATA
Mode	R
Initial	0

*Описание разрядов регистра полученных данных (**buffered_data_r**)*

DATA - полученные данные

3. Работа с программной моделью

3.1. Запись и чтение регистров

Управление модулем осуществляется путем записи или чтения регистров.

Для считывания текущего значения одного из регистров блока необходимо подать 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 служебный регистр блока необходимо сформировать:

- на порт *addr* — адрес выбранного регистра,
- на шине *D_in* — записываемую информацию,
- на порт *wr_en* — значение 1.

Запись в регистр полученных (**buffered_data_r**) данных игнорируется.

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.

[image SIReciever read write waveform] | *image_SIReciever_read_write_waveform.png*

Рисунок 1. Временная диаграмма чтения и записи регистров модуля SIReciever

Таблица 6. Адреса регистров

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

3.2. Смена конфигурации

Для изменения конфигурации передатчика необходимо:

1. Считав служебный регистр убедиться, что в данный момент не идет прием сообщения (**WRP** = 0). Если изменить поля **BC** и **FQM** во время приема сообщения, прием сообщения будет прерван.
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.

Некорректной считается конфигурация с нечетными длинами слова или длиной слова лежащей вне промежутка от 8 до 32 разрядов. При попытке записать подобную конфигурацию будет выставлен разряд **IRQICC** = 1, а поля **BC** и **PCE** останутся неизменными.

3.3. Прием сообщений

Для приема 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 приемом сообщения, или, работая по таймеру и периодически опрашивая регистр состояния, убедится, что сообщение было принято (**IRQRM** == 1 или, если контроль четности отключен и принято сообщение неверной четности **IRQPEM** == 1).
4. Считать принятое сообщение из регистра полученных данных (**buffered_data_r**).
5. Сбросить поле причины прерывания **IRQRM** (или, если и/или контроль четности отключен и принято сообщение неверной четности **IRQPEM**). Возможна работа без сбрасывания полей **IRQRM** и **IRQPEM**, но тогда вы не сможете отличить заново принятое сообщение от принятого в прошлый раз.
6. Ожидать следующее сообщение.

В случае когда поле ВС служебного регистра не равно 32, принятым сообщением являются младшие разряды регистра данных (**buffered_data_r [BC-1:0]**).

3.4. Работа с прерываниями

Запрос прерывания происходит, когда произошло одно из событий и разряд причины прерываний соответствующий этому событию не замаскирован. Узнать какое именно событие вызвало запрос на прерывание можно в [полях причин прерываний](#) служебного регистра.

События соответствующие разрядам причин прерываний

- **IRQRM** — Сообщение было успешно принято.
- **IRQPEM** — Принято сообщение с ошибкой четности.
- **IRQWLC** — Принято сообщение неверной длины.
- **IRQLE** — Произошла ошибка уровня на линии
- **IRQWCC** — Прием сообщения [прекращен](#) из-за попытки изменения полей BC и FQM в процессе отправки.
- **IRQICC** — Была предпринята попытка записать некорректную конфигурацию в конфигурационный регистр

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

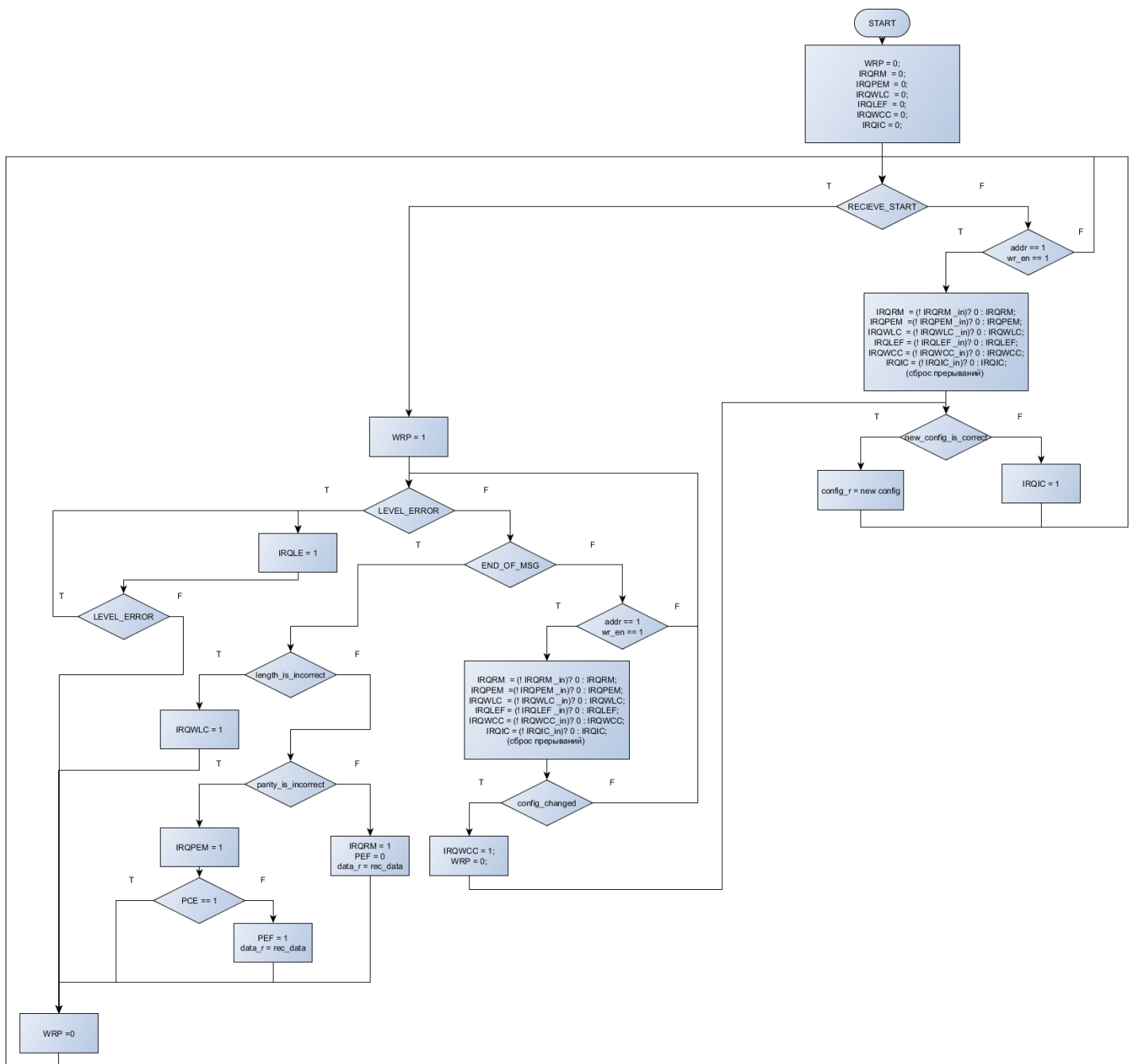


Рисунок 2. Алгоритм работы регистра состояния модуля SIReceiver

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.

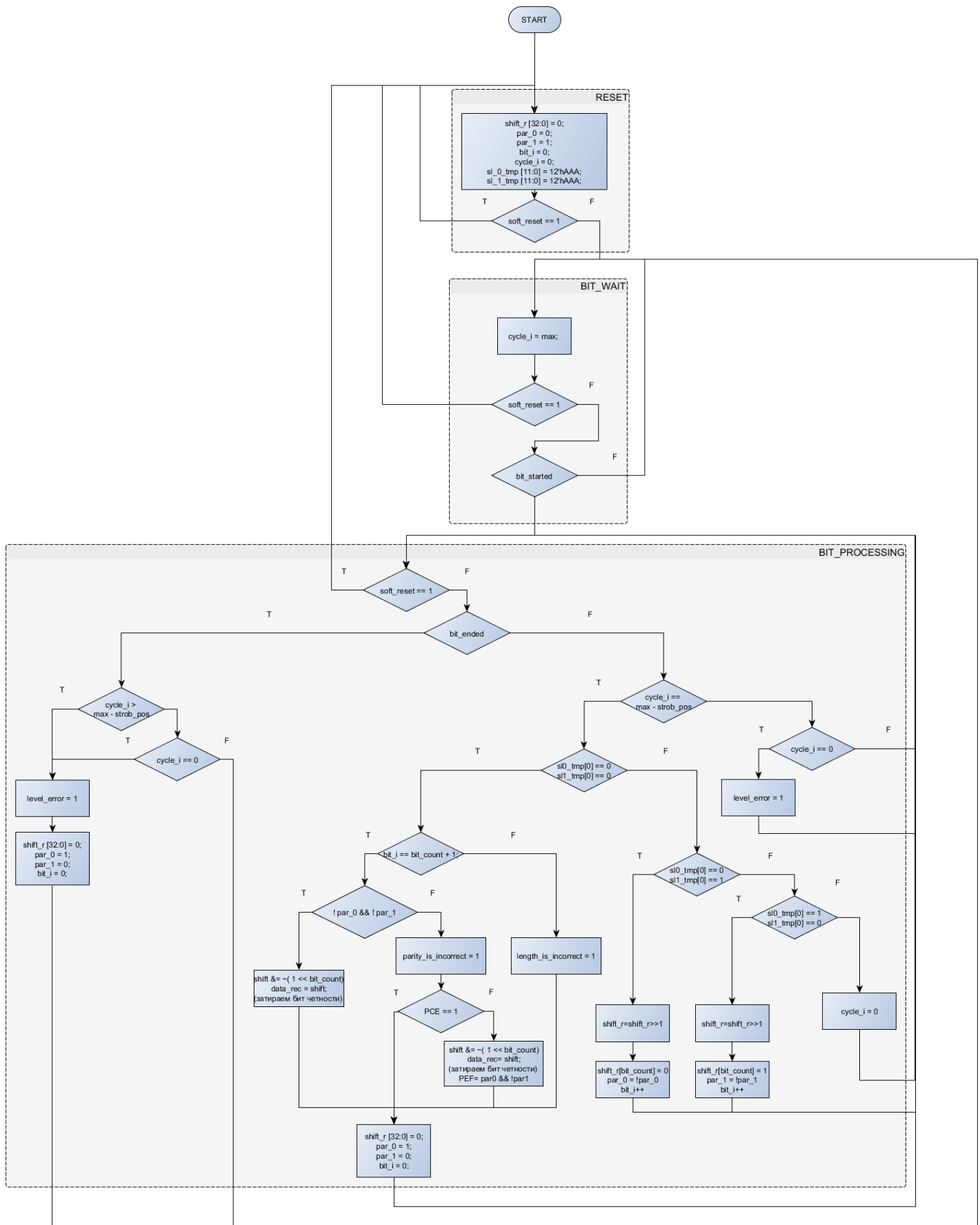


Рисунок 3. Алгоритм работы приема сообщения модуля SIReceiver

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.