

МИКРОЭЛЕКТРОНИКА

УДК 621.396.621:621.396.61:654.16

ГРНТИ 47.47.29+47.47.31+49.43.29

DOI: 10.34680/2076-8052.2024.3(137).391-402

Специальность ВАК 2.2.15

Поступила в редакцию / Received 13.06.2024

Принята к публикации / Accepted 19.09.2024

Научная статья

ОПТИМИЗАЦИЯ ФУНКЦИОНАЛЬНОЙ СХЕМЫ ПЕРЕДАТЧИКА/ПРИЕМНИКА ПО ИНТЕРФЕЙСУ RS-232

Румянцев Д. В., Петров М. Н.

Новгородский государственный университет имени Ярослава Мудрого (Великий Новгород, Россия)

Аннотация В статье раскрываются методы по улучшению качества связи информационного взаимодействия между устройствами через интерфейс передачи данных RS-232, путем добавления логики HDL, с использованием языка описания оборудования System Verilog. Подробно описан сам последовательный интерфейс, принцип его работы, структура посылки данных, способы подключения с указанием назначения выводов сигнала, пример передачи сообщения с использованием указанного интерфейса в графическом представлении. Приводится результат тестирования доработанной версии приемника, в среде ModelSim, с использованием метода множественной выборки для захвата данных с линии передачи, проверки ложного сигнала, информирующего о начале передачи посылки, поскольку передача в интерфейсе RS-232 осуществляется без синхронизирующего тактирования. Экспериментально было выявлено в каком промежутке времени установления уровня сигнала лучше всего захватывать данные. Приведена доработка модулей приемника и передатчика, путем добавления переменной сброса, которая играет роль счетчика цикла работы приема/отправления сообщения. Также объясняется почему скорость передачи 115200 бит в секунду пользуется популярностью среди выпускаемого промышленного и лабораторного оборудования. Расписан метод получения значения делителя, чтобы установить специализированную скорость передачи информации по последовательному интерфейсу, и настройка микросхемы UART с использованием конфигурационных регистров.

Ключевые слова: HDL, RS-232, приемник и передатчик, улучшение качества связи, информационное взаимодействие

Для цитирования: Румянцев Д. В., Петров М. Н. Оптимизация функциональной схемы передатчика/приемника по интерфейсу RS-232 // Вестник НовГУ. 2024. 3 (137). 391-402. DOI: 10.34680/2076-8052.2024.3(137).391-402

Research Article

OPTIMIZATION OF THE FUNCTIONAL DIAGRAM OF THE TRANSMITTER/RECEIVER VIA RS-232 INTERFACE

Rumiantsev D. V., Petrov M. N.

Yaroslav-the-Wise Novgorod State University (Veliky Novgorod, Russia)

Abstract The article discusses methods for improving the quality of communication in information interaction between devices through the RS-232 data transmission interface by adding HDL logic using the System Verilog hardware description language. The serial interface itself is described in detail, including its operating principle, data packet structure, connection methods with pinout assignments, and an example of message transmission using the specified interface in graphical representation. The results of testing the modified receiver version in the ModelSim environment are presented, using a multiple sampling method to capture data from the transmission line and check for false signals indicating the start of packet transmission, since transmission in the RS-232 interface is performed without synchronized clocking. Experimentally, the optimal time interval for capturing data during signal level establishment was identified. The article also details the modifications made to the receiver and transmitter modules by adding a reset variable that acts as a cycle

counter for the receive/send message operation. Additionally, it explains why a transmission speed of 115200 bits per second is popular among industrial and laboratory equipment. The method for obtaining the divider value to set a specialized data transmission speed via the serial interface is described, along with the configuration of the UART chip using configuration registers.

Keywords: HDL, RS-232, receiver and transmitter, communication quality improvement, information interaction

For citation: Rumiantsev D. V., Petrov M. N. Optimization of the functional diagram of the transmitter / receiver via RS-232 interface // Vestnik NovSU. 2024. 3 (137). 391-402. DOI: 10.34680/2076-8052.2024.3(137).391-402

Введение

Последовательный интерфейс – один из способов обмена данными, который до сих пор остаётся популярным. Данный интерфейс несложно реализовывать, он обладает высокой надёжностью и доступностью оборудования. Программное обеспечение для последовательного обмена данными можно реализовать с помощью стандартных функций, доступных в операционной системе Windows.

Последовательный интерфейс передачи информации применяется в коммуникационных системах, большинстве систем обмена данными и управления. Последовательные порты компьютерных систем поддерживают стандарт информационного взаимодействия RS-232, относящийся к протоколам последовательного обмена данными. Передача и приём данных по RS-232 интерфейсу происходит последовательно, бит за битом, что и дало название «последовательный порт». Структура асинхронной передачи данных по последовательному интерфейсу RS-232 представлена на рисунке 1 [1].

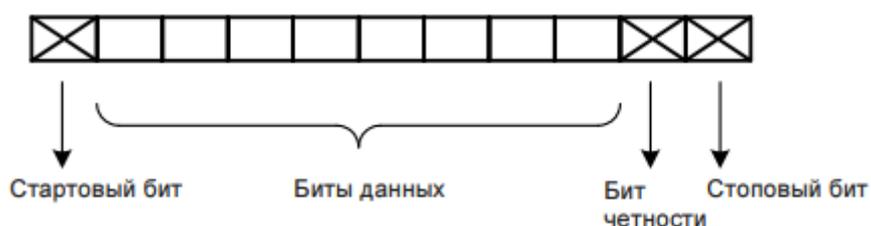


Рисунок 1. Структура посылки данных

Скорость обмена данными по последовательному интерфейсу может достигать 115200 бит в секунду, хотя современные разработки позволяют увеличить скорость до 921 Кбит/с. С увеличением скорости передачи растёт и количество помех, которые появляются на линии передачи и являются проблемой при отладке или настройке оборудования, у которого происходит информационное взаимодействие через RS-232 интерфейс. Также можно отметить, что скорость передачи часто выбирается именно 115200 бит/с.

Подробнее про интерфейс RS-232

В интерфейсе RS-232 логические уровни сигналов находятся в диапазонах от минус 3 до минус 25 вольт, что соответствует логической единице, и от плюс 3 до плюс 25 вольт, логический ноль. Диапазон от минус 3 до плюс 3 вольт соответствует зоне нечувствительности.

Интерфейс RS-232 позволяет передавать данные на расстояние до 15 метров. При этом были случаи, когда удавалось развернуть передачу данных через интерфейс RS-232 на расстоянии до 25 м. Это делает его популярным выбором для лабораторных и учебных систем обмена данными, а также для настройки, тестирования и отладки оборудования, в котором предусмотрено информационное взаимодействие с помощью микросхемы UART.

В передаче данных по интерфейсу RS-232, лежит принцип считывания значений заключенных между стартовым и стоповым битами. В упрощенном виде информационное взаимодействие представлено на рисунке 2.

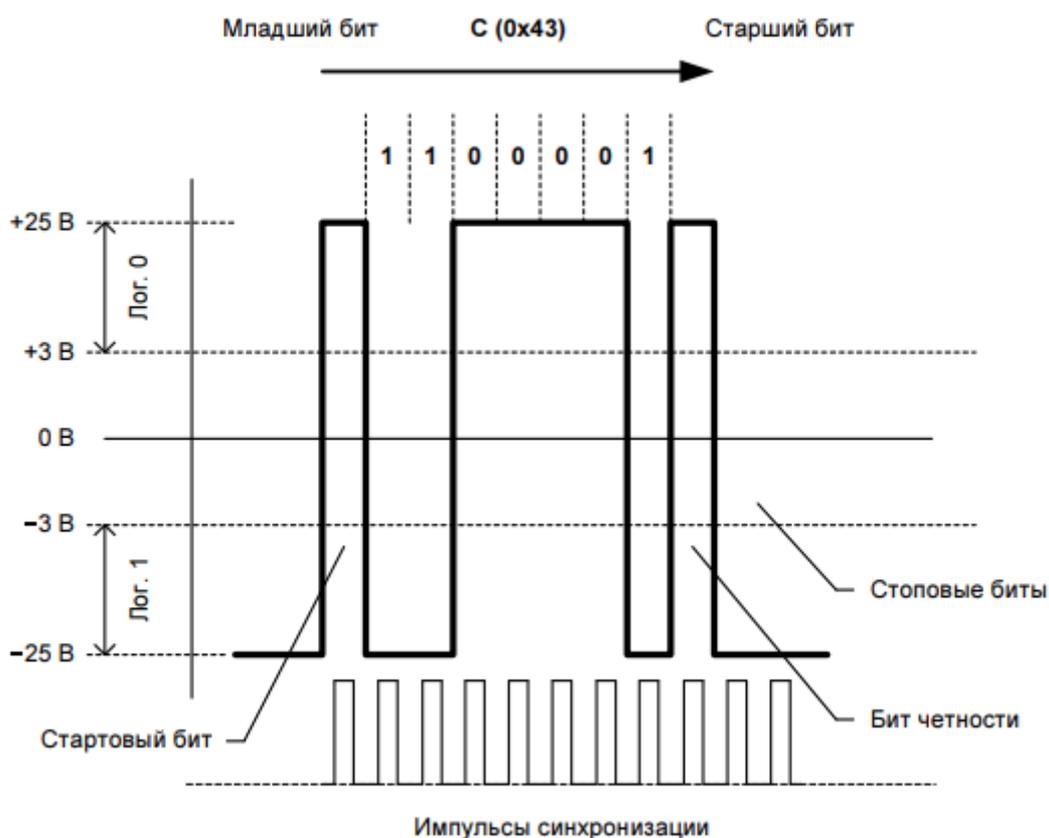


Рисунок 2. Передача байта данных

На рисунке 2 представлена передача значения 7'h43. Весь процесс начинается со стартового бита, затем идут 7 бит данных, бит чётности и в завершении два стоповых бита.

Передача блока сообщения обязательно должна начинаться со стартового бита. Затем выставляются биты передаваемых данных, количество которых зависит

от выбранного протокола информационного взаимодействия. Передача осуществляется младшим битом вперед. После передачи битов данных выставляется бит чётности. И в конечном итоге завершают передачу два стоповых бита, которые указывают о завершении передачи.

Строб начала передачи нового сообщения может быть обнаружен в любой момент времени, строго после завершения передачи предыдущего сообщения. Благодаря этому сигналу в приемнике происходит синхронизация, важное условие для правильной работы системы – это наличие одинаковой тактовой частоты синхронизации.

Когда на линии передачи выставляется стартовый бит от передатчика, приемник, в свою очередь, отсчитывает длительность сигнала, согласно с установленной частоте синхронизации, а также выбранной скорости передаваемого сообщения, и производит запись полученных значений с линии в регистр, где хранится передаваемый байт данных.

Для аппаратной реализации взаимодействия через интерфейс RS-232 обычно используется универсальный асинхронный приемопередатчик, который сокращённо называют UART (Universal Asynchronous Receiver-Transmitter), а физический уровень для преобразования сигналов обеспечивает другая микросхема, например MAX3232 [2, с. 32].

«На практике интерфейс RS-232 реализован в виде сигнальных линий, имеющих следующее назначение:

- GND (Ground) – сигнальная земля (относительно нее отсчитываются уровни сигналов);
- TxD (Transmit Data) – выход передатчика;
- RxD (Receive Data) – вход приемника;
- RTS (Request-To-Send) – запрос обмена данными, который выставляется контроллером последовательного порта;
- CTS (Clear-To-Send) – указывает на готовность модема или терминального устройства к обмену данными;
- DTR (Data Terminal Ready) – указывает модему или другому устройству, что контроллер последовательного порта готов установить соединение;
- DSR (Data Set Ready) – указывает контроллеру последовательного порта на то, что периферийное устройство готово установить соединение;
- DCD (Data Carrier Detect) – входной сигнал от удаленного модема или устройства. Когда модем или другое устройство обнаруживает появление несущей информации на другом конце линии, то линия становится активной.» [3, с. 73]

Для подключения устройств между собой через последовательный порт информационного взаимодействия применяется «нуль-модемное» соединение. В зависимости от требований в аппаратном и программном обеспечении системы обмена данными, может использоваться как полноценное соединение, представленное на рисунке 3а, так и упрощённый вариант подключений, показанный на рисунке 3б [4, с. 17]. Для диагностики и управления системами автоматизации, таких

как микроконтроллеры или платы управления, а также для отладки систем в лабораторных условиях часто применяется упрощенный вид подключения нуля-модемного соединения.



Рисунок 3. Нуль-модемное соединение: а – полнофункциональное соединение, б – упрощенный вариант

В процессе передачи данных используется определённый формат, который включает в себя несколько ключевых элементов:

- стартовый бит: первый бит в последовательности, который всегда равен нулю. Он служит сигналом начала передачи данных;
- 8 бит данных: основная часть сообщения, которая содержит информацию, подлежащую передаче;
- стоповый бит: последний бит в последовательности, равный единице. Он сигнализирует о завершении передачи данных.

Скорость 115200 бит/с

Асинхронный приёмопередатчик является ключевым элементом последовательного интерфейса и представляет собой сложный аппаратно-программный модуль для обмена данными. В этом разделе статьи мы сосредоточимся на работе аппаратной части этого устройства. Микросхема UART широко используются в одноплатных системах, основанных на микропроцессорах и микроконтроллерах. Модули последовательной передачи данных во всех современных микроконтроллерах работают с помощью UART микросхемы.

Настройка устройства UART осуществляются через внутренние регистры. Через конфигурирование определяют режимы функционирования микросхемы. Настройки записываются в начале цикла работы микросхемы и корректируются управляющей системой по мере необходимости внутри цикла работы устройства. Большая часть этих регистров доступна для использования в пользовательских

приложениях, в то время как некоторые из них являются внутренними служебными регистрами и предназначены для выполнения специализированных операций.

Скорость обмена данными по последовательному интерфейсу задается путем деления тактовой частоты кристалла микросхемы UART. Стандартизированная тактовая частота для микросхемы UART обычно равняется 1,84 МГц. Внутренняя частота синхронизации микросхемы в 16 раз меньше частоты тактирования кристалла, находится по формуле (1). С помощью данной формулы также можно найти значение делителя «Divider», чтобы установить специализированную скорость передачи по последовательному интерфейсу.

$$Baud = \frac{f_{clk}}{Divider} \quad (1)$$

где *Baud* – скорость передачи, бит/с;

f_{clk} – тактовая частота синхронизации, Гц;

Divider – константа.

В ходе расчета внутренней частоты синхронизации получаем значение равное 115200 Гц. Для получения специализированной скорости обмена данными в оба байта конфигурационного регистра делителя частоты нужно записать значение, равное частоте синхронизации в герцах, деленное на скорость обмена данными в бодах. Например, для получения стандартизированной скорости обмена информацией между устройствами, а именно 9600 бод [5, с. 68], в регистр делителя частоты нужно передать значение 8`h0C в шестнадцатеричной системе исчисления. В младшем байте управляющего регистра делителя частоты будет значение 8`h0C, а в старшем – 8`h00. Для получения максимальной скорости обмена 115200 бод, для указанной микросхемы UART с тактовой частотой равной 1,8432 МГц, в конфигурационный регистр делителя частоты нужно передать значение 8`h01.

Захват данных в начале при смене уровня сигнала

При захвате данных в начале, сбор данных в регистр происходит сразу же после смены уровня сигнала без задержек, при данном подходе помехи появляются уже на первом бите посылке. В ходе эксперимента отправлялось сообщение 8`h01 и из 100 посылок три значения пришли с ошибочными данными - 8`h03. При проверке седьмого бита данных наблюдается 21 верное принятое значение сообщения из 100. Восьмой бит данных принимается без помех поскольку стоповый бит имеет состояние высокого уровня.

Захват данных в конце при смене уровня сигнала

При записи данных в конце смены уровня сигнала стоповый бит, равный логической единице, создает помеху для последующего использования принятой информации. Искажения принятого сообщения, в отличии от исходного отправленного, происходят настолько сильно, что из 100 попыток отправить сообщение, ни одно не было получено в исходном представлении. Проверялось нахождение бита в каждом

разряде байта, результат не изменился. Отсюда следует невозможным считывание сообщения в конце смены уровня сигнала, при использовании интерфейса RS-232.

Захват данных по центру установления сигнала

При добавлении счетчика отслеживания длительности сигнала появляется возможность захватывать данные в любом промежутке времени. Прописываем условие для считывания значений в регистр сообщения, таким образом, чтобы оно происходило приблизительно по середине:

```

    «if (tickCount[14:0] == Divider [15:1])
    begin
    rData[bitCount] = rx; // сбор данных в регистр
    end».

```

В данном случае, при достижении счетчиком «tickCount» половины регистра «Divider», условие выполняется и происходит запись значения с линии передачи данных. Интересным образом происходит деление переменной «Divider» путем сдвига регистра константы вправо на 1 бит.

В ходе проверки такого варианта захвата данных выяснилось, что сообщения отправляются без ошибок, все биты находятся на своих местах и точно передают несущую информацию. Было отправлено 900 сообщений, каждое из них было верно обработано и принято.

Множественная выборка для захвата данных

Захват значений по центру установленного сигнала значительно улучшает качество связи, но проверка происходит всего один раз. Что будет когда наступит момент помех, находящихся в центре передаваемого сигнала. В регистр байта запишется то значение, которое было во время захвата. Чтобы избежать этого можно использовать метод множественной выборки. Если три раза снять значение на линии передачи и значение большинства записать в регистр, то мы уберем вероятность одиночного сбоя по центру. Как уже ранее выяснилось, самые достоверные данные находятся примерно в середине установленного уровня сигнала передачи, где и следует применять множественную выборку, поскольку влияние времени установления сигнала отсутствует.

Метод множественной выборки был реализован в RS-232 приемнике на HDL языке System Verilog. Выборка значений производилась три раза по центру формирования сигнала на линии передачи. Небольшая часть логики работы приемника, а именно запись значений в регистр:

```

    «if (tickCount[14:0] == Divider [15:1]-Divider [15:3])
    v1 = rx;

```

```

if (tickCount[14:0] == Divider [15:1])
v2 = rx;
if (tickCount[14:0] == Divider [15:1] + Divider [15:3])
begin
v3 = rx;
if (v1&v2||v1&v3||v3&v2)
rData[bitCount] = 1; // сбор данных в регистр
else if (!(v1&v2||v1&v3||v3&v2))
rData[bitCount] = 0; // сбор данных в регистр
end».

```

Производится запись трех значений с линии передачи в переменные «v1», «v2», «v3» в небольшом промежутке, чтобы не затрагивать время нарастания сигнала на высоких скоростях передачи. А далее уже идет передача значения на основе выбора большинства, в регистр «rData» записывается либо единица, либо ноль. Моделирование передачи с использованием множественной выборкой приведено на рисунке 4, с использованием программного обеспечения ModelSim средства Waveform.

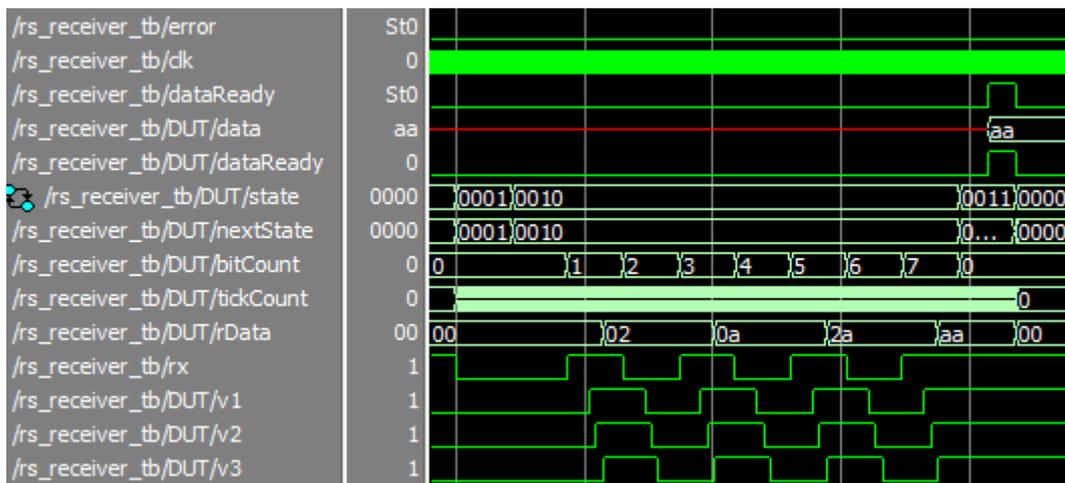


Рисунок 4. Моделирование работы RS-232 приемника

Как видно из рисунка 4, на линии передачи «rx» выставляются данные байта 8'hAA, в шестнадцатеричной системе исчисления младшим битом вперед. Запись данных производится три раза, а затем значение большинства передается в регистр «rData» сразу после третьей проверки. И когда все 8 бит данных будут успешно приняты, тогда приемник выставит выходной сигнал о готовности данных «dataReady» для дальнейшей обработки.

Улучшение передатчика и приемника

Как выяснилось для корректного приёма каждый бит данных должен быть захвачен точно либо же примерно в середине временного интервала, когда он является действительным. При этом одиночного захвата значения не всегда достаточно и этот момент сможет поправить метод множественной выборки значений. По мере увеличения скорости передачи данных синхронизация передатчика и приёмника становится более сложной задачей. Это связано с тем, что паразитные ёмкости электронных компонентов начинают влиять на фронты синхроимпульсов, а также индуктивность физической линии передачи.

Логика работы приёмника, описанная на языке HDL System Verilog с использованием конечных автоматов переключения состояний, представлена в виде схемы на рисунке 5 [6, с. 70].

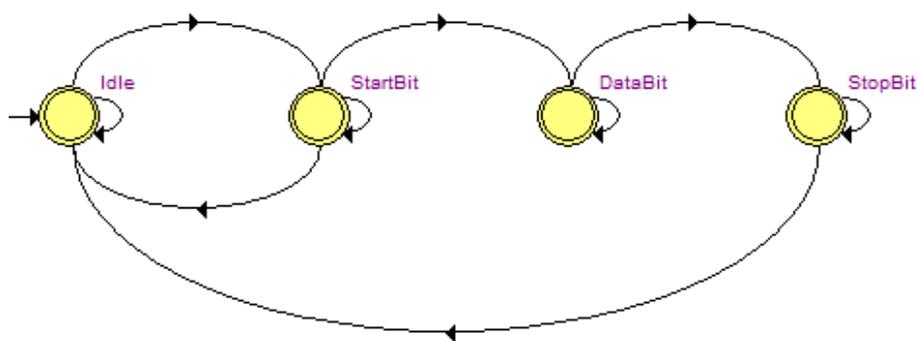


Рисунок 5. Схема работы приемника RS232

Работа передатчика и приёмника происходит в следующих состояниях:

- Idle: режим ожидания, когда передатчик или приёмник не активны;
- StartBit: приём или отправка стартового бита;
- DataBit: формирование восьмибитного сообщения данных;
- StopBit: формирование стопового бита и завершение передачи или приёма сообщения.

Однако при работе с данными могут возникать проблемы, связанные с помехами на линии передачи. В таких случаях возможны ложные срабатывания прерывания от линии передачи, что может привести к обнаружению ложного стартового сигнала. В этом случае можно получить сообщение, заполненное значением: «8`hFF».

Чтобы избежать этой проблемы, была доработана логика в проверке стартового и стопового значений битов в середине формирования сигнала. Это позволяет исключить ложные данные из дальнейшей обработки. Если на месте стартового бита обнаруживается логическая единица, выставляется сигнал ошибки, и передатчик возвращается в режим ожидания. Это можно увидеть на рисунке 5. По такому же принципу происходит обработка стопового бита, если на его месте в середине формирования сигнала будет обнаружен «логический ноль».

Моделирование ложного сигнала начала передачи приведено на рисунке 6, с использованием программного обеспечения ModelSim средства Waveform.

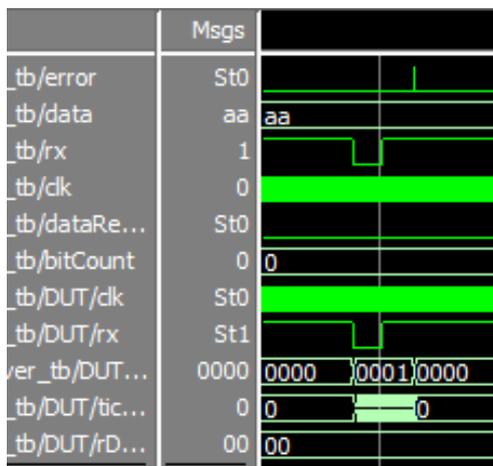


Рисунок 6. Моделирование работы приемника RS232

Можно заметить, что модуль сообщил об ошибке сигналом «error», поскольку произошло ложное начало передачи данных.

Также модуль приемника RS-232 был доработан и исправлен от зависания работы с помощью добавления переменной «reset», которая играет роль счетчика цикла работы приема сообщения. Модуль тестировался на частоте синхронизации 25 МГц, при скорости передачи данных 115200 бит/с. Если переменная достигает значения 2200, то происходил сброс состояний в «IDLE», в том случае, когда зависание происходило на этапе записи данных в регистр, весь процесс принудительно переходил в состояние конечного автомата «Idle» через состояние «StopBit». Общее значение цикла работы вычисляется путем умножения значения «Divider», которое находится из формулы 1, на 10. Следует брать значение, которое отвечает за сброс состояний в статус «Idle», с запасом.

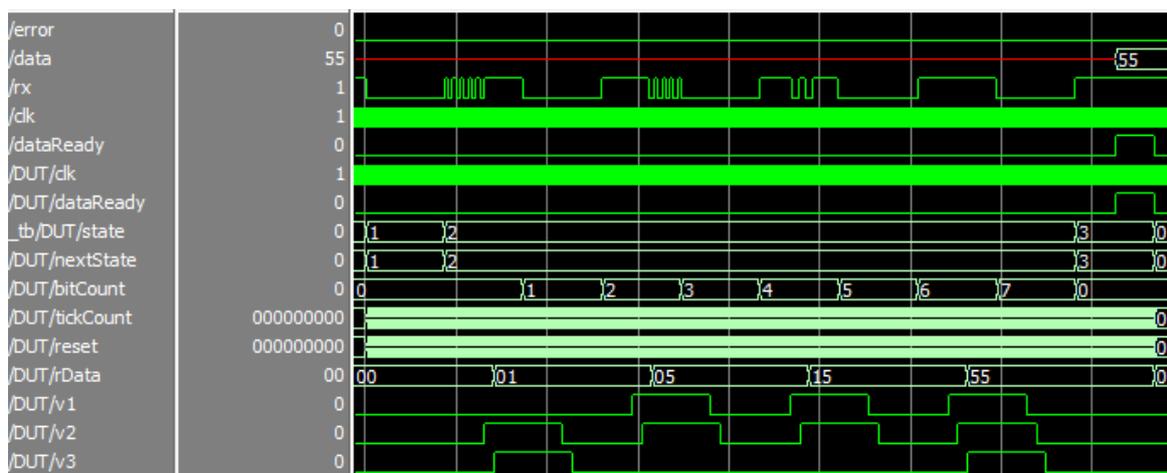


Рисунок 7. Моделирование помех на линии передачи

Моделирование передачи данных 8`h55, в шестнадцатеричной системе исчисления, с моделированием помех в начале смены уровня сигнала и в конце, и по центру установления сигнала, приведено на рисунке 7, с использованием программного обеспечения ModelSim средства Waveform. HDL описание модуля приемника производилось на языке System Verilog.

Как видно из рисунка 7, у модуля приемника повысилась помехоустойчивость на линии последовательной передачи данных. Метод множественной выборки, добавленный в логику работы приемника, вполне справляется со своей задачей, при наличии даже грубых помех.

Заключение

Изученные методы по улучшению приемника и передатчика по RS-232 интерфейсу, на уровне описания логической схемы, исключают помехи, один из них – это добавление обработки ложных сигналов срабатывания начала передачи. Это необходимо, поскольку передача осуществляется без синхронизирующего тактирования. Также при наличии времени нарастания сигнала, на аппаратном уровне, существует вероятность того, что если сохранять значения сразу же при первой смене уровня на линии передачи, то возможна ситуация, когда передаваемые данные будут находиться в промежутке времени нарастания сигнала, и в этом случае пакет данных будет не корректным. Решение считывать значения на пике установленного уровня, повышает вероятность достоверности несущей информации, но одной проверки недостаточно. Поэтому приемник был улучшен, с помощью добавления метода множественной выборки, которая в свою очередь исключала одиночные сбои на линии передачи. Добавление логики обработки помехи, во время установления сигнала на линии, и обнаружение ложного сигнала о начале передачи сообщения, повышает качество связи по интерфейсу передачи RS-232.

Скорость передачи RS-232 115200 бит/с универсальна для переноса на более старые версии контроллеров, без изменений логики приемника и передатчика. Эта скорость поддерживается как в старых контроллерах, так и в новых. Изменение значения делителя в прошивке для микроконтроллера, и в программном обеспечении не требуется.

Список литературы

1. Ронжин И. В., Макарова А. В. Реализация универсального асинхронного приемопередатчика на кристалле программируемой логической интегральной схемы // Политехнический молодежный журнал. 2018. 4. 1-7. DOI: 10.18698/2541-8009-2018-04-295
2. Иванов Ю. И., Югай В. Я. Интерфейсы средств автоматизации: учебное пособие. Таганрог: Изд-во ТРТУ, 2005. 252 с.
3. Магда Ю. С. Компьютер в домашней лаборатории: практическое руководство. Москва: ДМК Пресс, 2008. 200 с.
4. Магда Ю. С. Программирование последовательных интерфейсов. Санкт-Петербург: БХВ-Петербург, 2009. 304 с.

5. Барабанов А. В. Проектирование цифровых устройств на языках VHDL и Verilog: учебное пособие. Воронеж: Воронежский государственный технический университет, 2015. 167 с.
6. Гребенников А. HDL-реализация асинхронного приёмопередатчика // Современная электроника. 2011. 4. 70-73.

References

1. Ronzhin I. V., Makarova A. V. Realizatsiya universal'nogo asinkhronnogo priyemoperedatchika na kristalle programmiruyemoy logicheskoy integral'noy skhemy [Implementation of a universal asynchronous receiver-transmitter on a programmable logic integrated circuit chip] // Politechnical student journal. 2018. 4. 1-7.
2. Ivanov Yu. I., Yugay V. Ya. Interfeysy sredstv avtomatizatsii: uchebnoye posobiye [Interfaces of automation equipment: tutorial]. Taganrog: Publishing house of TRTU, 2005. 252 p.
3. Magda Yu. S. Komp'yuter v domashney laboratorii: prakticheskoye rukovodstvo [Computer in the home laboratory: practical guide]. Moscow: DMK Press, 2008. 200 p.
4. Magda Yu. S. Programirovaniye posledovatel'nykh interfeysov [Programming of serial interfaces]. Saint-Petersburg: BHV-Petersburg, 2009. 304 p.
5. Barabanov A. V. Proyektirovaniye tsifrovyykh ustroystv na yazykakh VHDL i Verilog: uchebnoye posobiye [Design of digital devices in VHDL and Verilog languages: tutorial]. Voronezh: Voronezh State Technical University, 2015. 167 p.
6. Grebennikov A. HDL-realizatsiya asinkhronnogo priyemoperedatchika [HDL implementation of an asynchronous receiver-transmitter] // Sovremennaya elektronika. 2011. 4. 70-73.

Информация об авторах

Румянцев Денис Вячеславович – студент, Новгородский государственный университет имени Ярослава Мудрого (Великий Новгород, Россия), ORCID: 0009-0004-0470-2146, dan.rumiantsev@yandex.ru

Петров Михаил Николаевич – кандидат технических наук, профессор, профессор, Новгородский государственный университет имени Ярослава Мудрого (Великий Новгород, Россия), ORCID: 0000-0002-5122-2450, Mikhail.Petrov@novsu.ru