Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Сибирский государственный университет телекоммуникаций и

информатики»

(СибГУТИ)

Кафедра «Инфокоммуникационных систем и сетей»

ИКСС

Мелентьев О.Г.

***Универсальный асинхронный приемо-передатчик.***

***Последовательный порт***

Методические указания

к лабораторной работе

Новосибирск, 2023

Краткая теория

UART – Universal Asynchronous Receiver-Transmitter, что в переводе звучит как Универсальный Асинхронный Приемо-Передатчик. Он широко используется в компьютерах, контроллерах, датчиках, средствах коммуникации, и других электронных устройствах. Операционная система распознает его как СОМ порт.

Асинхронный способ передачи данных — такой способ передачи цифровых данных от передатчика к приемнику по последовательному интерфейсу, при котором данные передаются в любой момент времени. Для того, чтобы приёмник инициировал прием данных, вводятся специальные битовые последовательности, обрамляющие данные. Перед началом передачи данных передается стартовый бит (равен нулю), в конце передачи данных передается стоповый бит (равен единице).

Передача данных в UART осуществляется по одному биту в равные промежутки времени. Этот временной промежуток определяется заданной скоростью UART и для конкретного соединения указывается в бодах (что в данном случае соответствует битам в секунду). Существует общепринятый ряд стандартных скоростей: 300; 600; 1200; 2400; 4800; 9600; 19200; 38400; 57600; 115200; 230400; 460800; 921600 бод.

Структурная схема возможного варианта построения приемника приведена на рисунке 1.

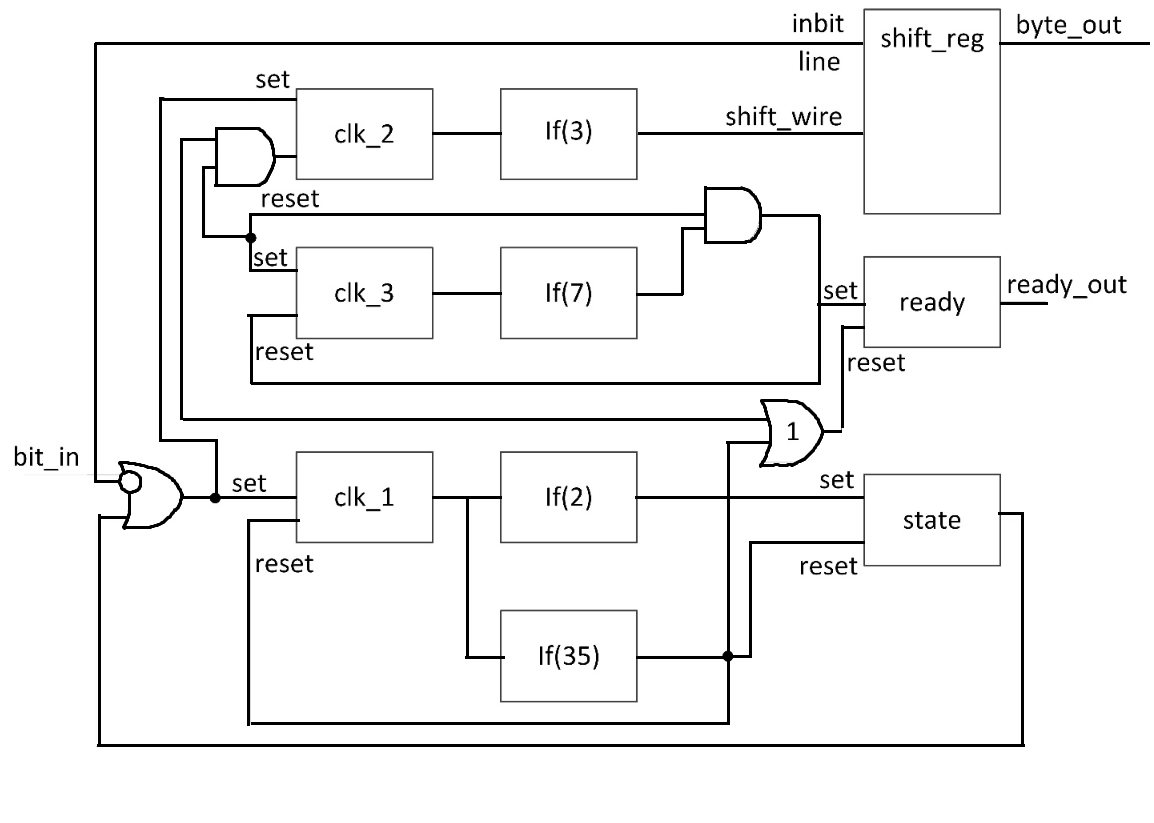


Рисунок 1. Структурная схема приемника UART

Входами приемника являются: сигнал тактовой частоты clk, сигнал сброса reset и информационный однобитный сигнал bit\_in. Выходными – byte\_out (принятый байт) и сигнал готовности данных ready\_out.

Модуль приемника включает следующие блоки.

Сдвиговый регистр (shift\_reg) с последовательной записью в старший разряд и параллельным 8-разрядным выходом.

Триггер готовности данных (read). Триггер состояния приема (state).

Счетчик тактов в цикле приема cnt\_1.

Счетчик тактов между сдвигами содержимого регистра cnt\_2.

Счетчик принятых бит cnt\_3.

Схемы проверки содержимого счетчиков на выполнение условий.

И комбинационные логические схемы для формирования внутренних управляющих сигналов.

Блок – схема приемника представлена на рисунке 2, эпюры сигналов приведены на рисунке 3.

Исходное состояние: все счетчики и регистры обнулены, на входе сигнал стопа равный логической единице.

При переходе входного сигнала в ноль (старт), счетчик тактов цикла cnt\_1 получает сигнал разрешения счета и начинает считать.

При подтверждении старта на следующем такте (if(cnt\_1 = 2)), выход схемы сравнения переходит в единицу, которая обнуляет триггер валидности выходных данных read, счетчик cnt\_2 и переводит триггер состояния state в единицу. Единица с выхода триггера через схему логического сложения поступит на входы set счетчиков cnt\_1 и cnt\_2, что разрешит их работу до конца цикла приема.

Через три такта значение счетчика cnt\_2 станет равным 3, что приведет к выполнению условия и появлению логической единицы на входе shift\_reg, входе set счетчика cnt\_3 и входе reset счетчика cnt\_2. Значит на следующем фронте тактового импульса произойдет запись первого бита в старший разряд регистра со сдвигом его содержимого в направлении младшего разряда, увеличение значения счетчика принятых бит cnt\_3 на единицу и сброс счетчика тактов между сдвигами. Аналогичные процедуры будут повторяться каждые четыре такта, пока значение счетчика принятых бит не достигнет 7. При выполнении двух условий if(cnt\_2 =3) & if(cnt\_3 = 7), триггер готовности данных устанавливается в единицу и обнуляются счетчики cnt\_2 и cnt\_3.

По окончанию 35-го такта обнуляются счетчик cnt\_1, триггеры готовности данных и состояния. На входе сигнал СТОП равный логической единице, на выходе схемы ИЛИ ноль, разрешения счета нет. Цикл завершен.

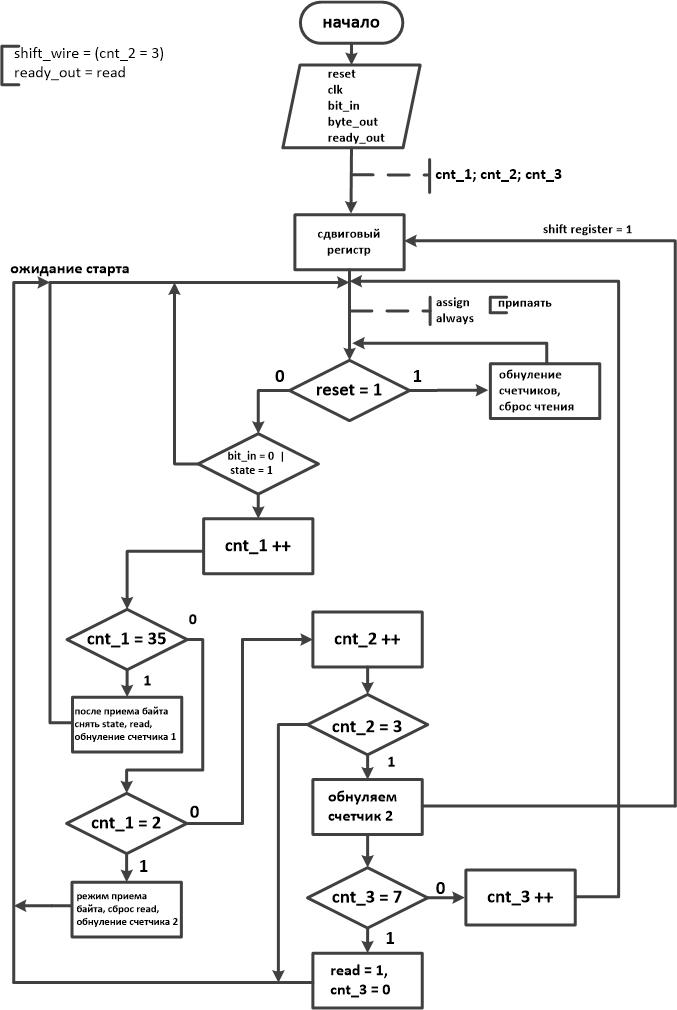


Рисунок 2. Блок – схема приемник

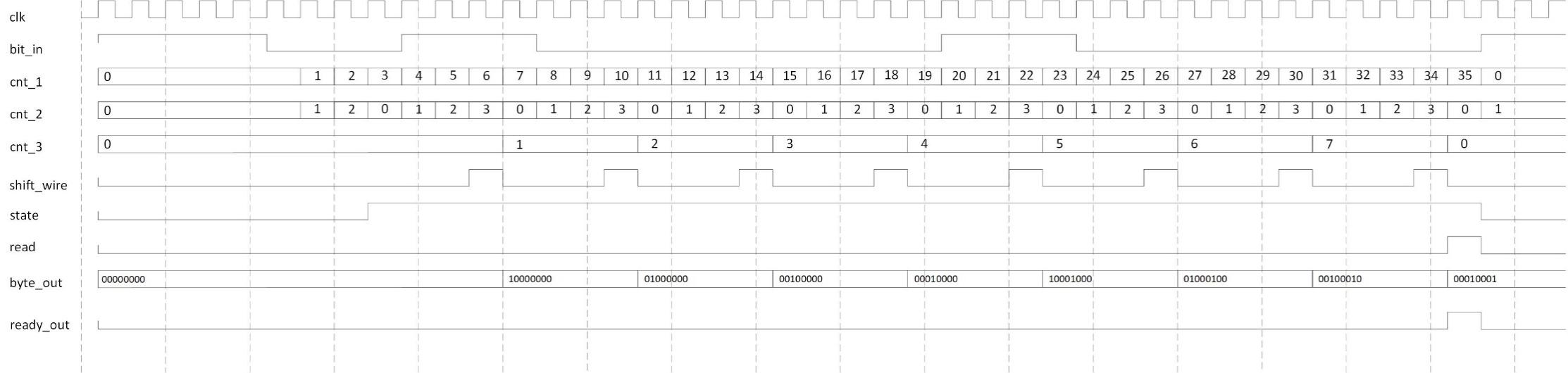


Рисунок 3. Эпюры сигналов одного цикла приема

Структурная схема передатчика UART приведена на рисунке 4, а его блок схема на рисунке 5.

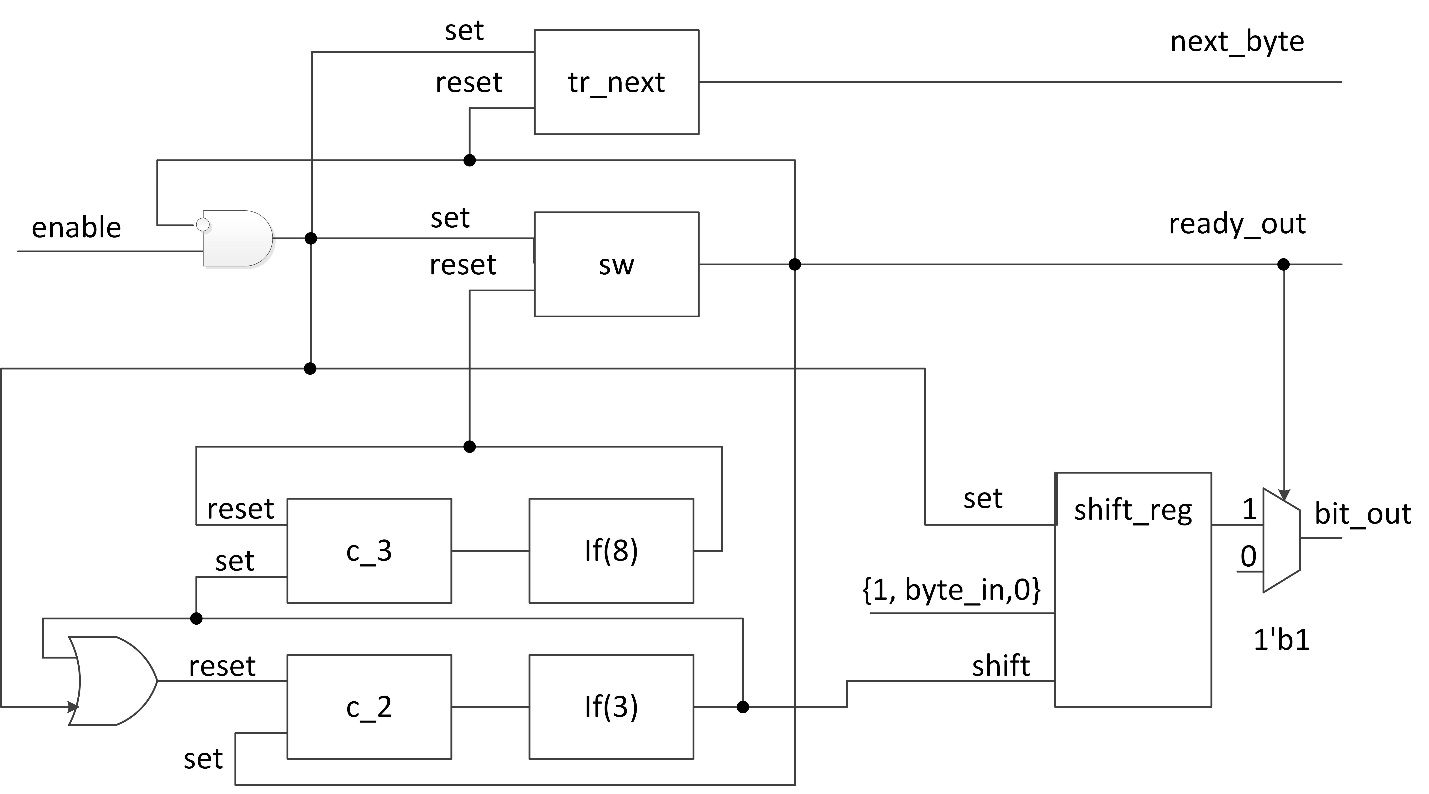


Рисунок 4. Структурная схема передатчика UART

Входами приемника являются: сигнал тактовой частоты clk, сигнал сброса reset, byte\_in – входной байт, enable – разрешение чтения.

Выходы: bit\_out – последовательный выходной сигнал uart;

ready\_out – устройство занято передачей последовательных бит;

next\_byte – разрешение смены следующего байта на входе;

Модуль передатчика содержит 10-ти разрядный сдвиговый регистр с последовательной загрузкой, триггер управления выходным мультиплексором sw он же формирует сигнал ready\_out, триггер разрешения смены данных на входе tr\_next, cnt\_2 – счетчик тактов между сменой выходных битов, cnt\_3 – счетчик переданных бит, и схемы проверки условий. Выходной мультиплексор в исходном состоянии коммутирует на выход СТОП, т.е. логическую единицу. В рабочем режиме (sw = 1) на выход модуля коммутируется младший разряд сдвигового регистра.

При наличии сигнала reset обнуляем счетчики и регистры.

В исходном состоянии sw = 0, на вход транслируется стоп = 1.

При первом получении сигнала разрешения (in\_enable == 1 & sw == 0)

входной байт шинируется с битами старта и стопа и записывается в регистр. Формируется сигнал tr\_next разрешения смены входного байта на один такт и триггер sw устанавливается в единицу до передачи последнего бита байта. Обнуляется счетчик тактов между сдвигами cnt\_2.

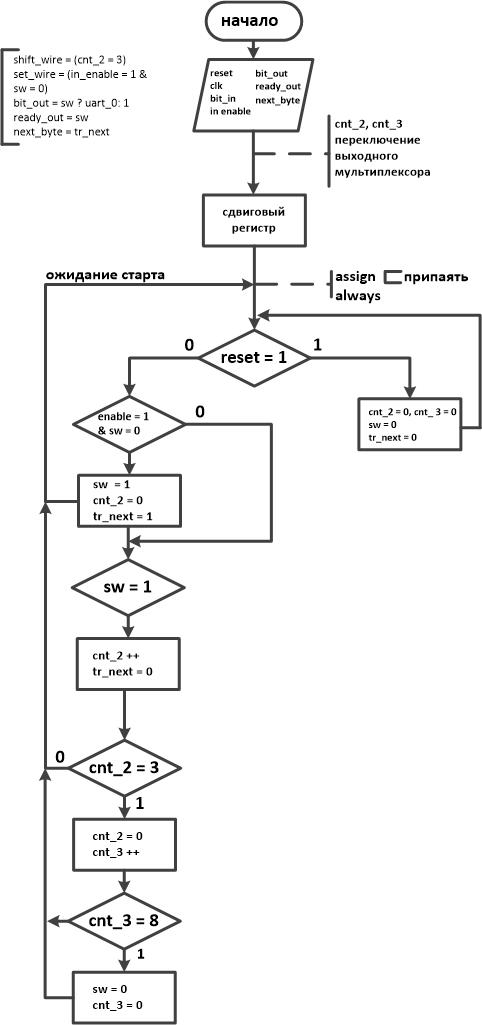


Рисунок 5

Логическая единица с выхода sw переключает младший разряд регистра к выходу мультиплексора и на выходе передатчика появляется сигнал СТАРТ, равный нулю.

На следующем такте снимаем сигнал разрешения смены данных (tr\_next) и инкрементируем счетчик cnt\_2.

При значении счетчика cnt\_2 = 3, формируем логическую единицу на входе shift и на входе reset счетчика cnt\_2. Таким образом, по фронту следующего тактового импульса содержимое регистра сдвинется в сторону младшего разряда и на выходе передатчика появится младший разряд входного байта. Инкрементируем счетчик переданных бит.

Каждые четыре такта сдвигаем регистр, обнуляем cnt\_2 и инкрементируем счетчик переданных битов cnt\_3.

При достижении cnt\_3 значения 8, Обнуляем счетчик переданных бит и обнуляем триггер управления мультиплексором. На выход модуля коммутируется сигнал СТОП.

Эпюры сигналов приведены на рисунке 6.

Адаптируйте тест бенч из ЛПЗ «Сдвиговые регистры» и проведите симуляцию передатчика и приемника UART вместе.

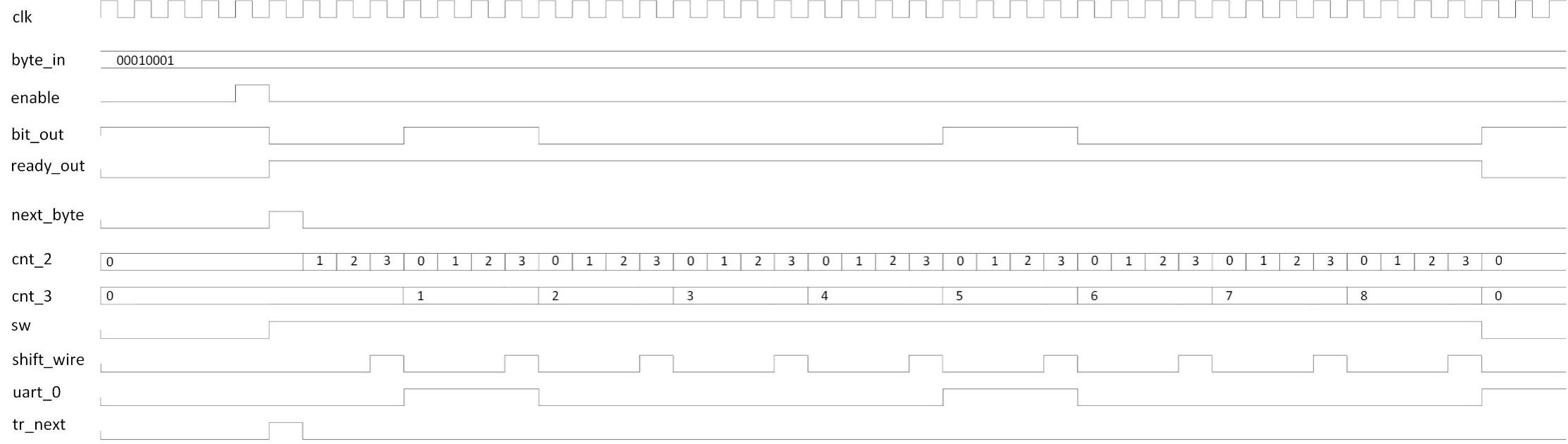
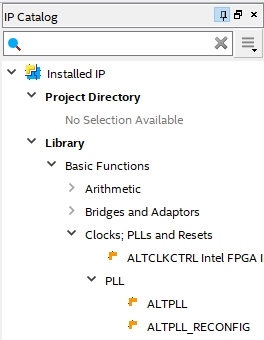


Рисунок 6. Эпюры сигналов одного цикла передачи

Для обеспечения максимальной скорости работы нашего COM порта 921 600 Бод, нам понадобится тактовая частот в 4 раза больше, т.е. 3.6864 МГц. Для получения данной частоты воспользуемся встроенной макрофункцией ALTPLL.

Выбираем данный компонент из IP каталога, расположенного справа.



Запускается визард связанный с компонентой ALTPLL. Задаем нужные нам параметры. В большинстве случаев многие из параметров можно не устанавливать. Например, использовать Select the PLL type automatically и еще In normal mode. Из важных параметров - это тип микросхемы, согласно проекта (Cyclone IV), значение входной частоты - это частота генератора, установленного на плате. У нас на плате **Марсоход3**- это 100Мгц. И еще указать быстродействие (speed grade) микросхемы - это 8.  
Нажимаем кнопку Next.

На следующем экране визарда отмечаете, какие сигналы PLL вам нужны. Нужен ли вам в проекте сброс PLL areset или разрешение работы PLL pfdena. Нужен ли сигнал сигнализирующий о том, что нужная частота установлена locked? Принимаете решение об этих сигналах и нажимаете Next.

Пропускаем несколько окон до с0.

Устанавливаем желаемую частоту и видим какую частоту может выдать PLL. В нашем случае 3.686397 МГц, что нас устраивает.

Пропускаем следующие окна и сохраняем файл на языке Verilog с именем mypll и добавляем к проекту.

Пройдя визард до конца мы получаем некоторые сгенерированные файлы. Файл с расширениями \*.BSF - это компонент, который можно вставить в схему, если вы делаете проект в схематическом виде. Файл вида mypll.v это модуль на Verilog, экземпляры которого можно вставлять в свои модули Verilog

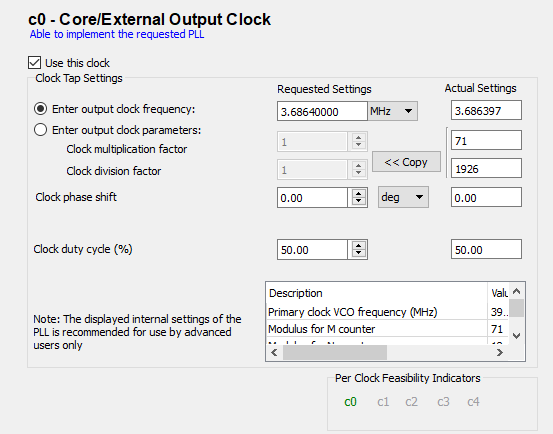


Рисунок 7. Задание выходной частоты

Соберите модуль верхнего уровня проекта uart\_trasit.

Начало модуля

module uart\_transit

    (

    input sb0, sb1,

    input clk\_100,  //тактовые импульсы 100 МГц для PLL

    input uart\_in,  // последовательный вход от FTDI\_BD0

    output uart\_out, //последовательный вывод на FTDI\_BD1

    output [7:0] hl );

    logic areset;   //кнопка дребезжит

    assign areset = ~ sb0;  //сделал асинхронный ресет

    reg reset;  //и синхронный триггер ресет, который работает по Фронту

    always @ (posedge clk\_100)

    reset <= areset;

    wire [7:0] Urd\_buf, buf\_Uwr;

    wire Urd\_wr;

    wire Uwr\_rd;

    wire yyy; // в буфере есть данные

    wire enbl;  //разрешение читать следующий байт

    wire xxx;   //uart занят передачей

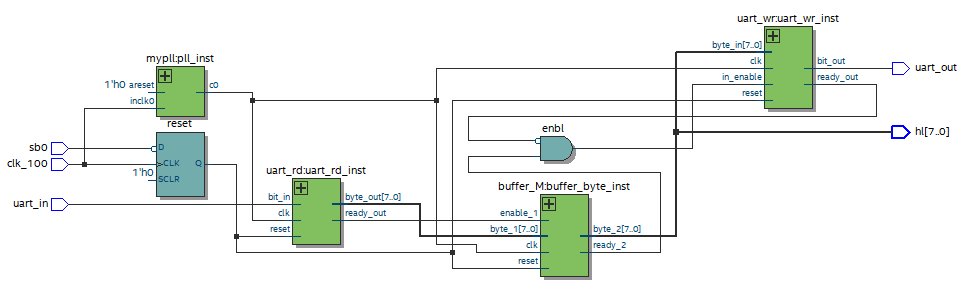
    assign enbl = yyy & ~ xxx;  //в буфере есть данные и uart свободен

    assign hl = buf\_Uwr;   //выходы буфера цепляем ко всем светодиодам

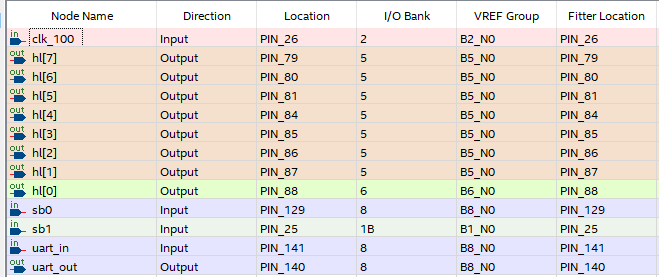
    wire clk\_4; //частота в 4 раза выше скорости СОМ порта

Далее создаете экземпляры модулей, инициированные проводами, входами и выходами.

Создайте и откомпилируйте проект в Quartus. Посмотрите схему.



Назначьте входам и выходам модуля верхнего уровня выводы микросхемы ПЛИС. Для микросхемы 10M08SAE144C8G – Марсоход 3 bis, распиновка приведена ниже.



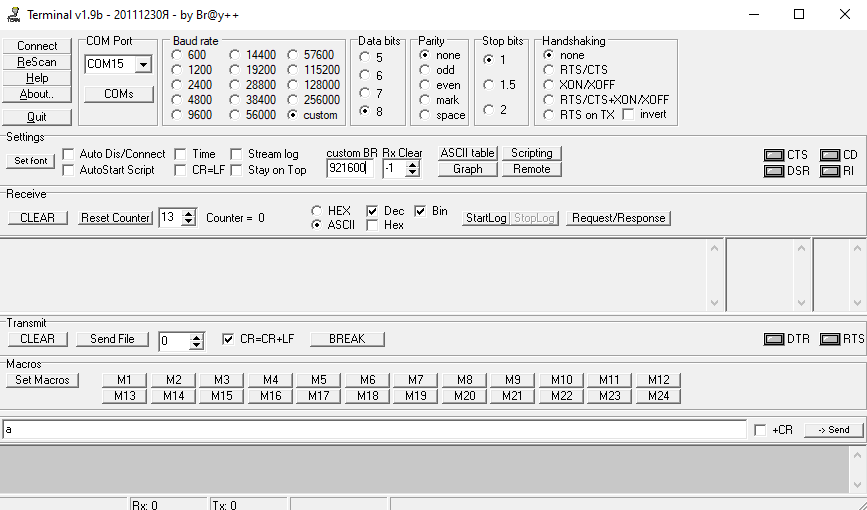
Проведите повторную компиляцию.

Прошейте ПЛИС

Откройте терминальную программу, убедитесь что в ней появились два новых COM порта и выберите последний. Укажите скорость порта custom BR 921600 Бод. Нажмите кнопку connect.

Установите галочки для отображения принятого байта в десятичном и бинарном форматах.

В нижней строке терминала введите латинскую букву a, и нажмите кнопку Send.



UART с кодом буквы прошел через ПЛИС, запомнился в буфере, отразился на светодиодах и вернулся в терминалку, появился в верхнем окне приема.

В окнах справа принятый байт показан в десятичном и бинарном форматах.

Сравните со светодиодами на отладочной плате.

