

COL215 LAB Assignment 7 : Asynchronous Serial Transmitter

Mohit Sharma(2019CS10372), Arka Mandal(2019CS50617)

1 Aim

Design asynchronous serial transmitter to create a loop with the receiver developed in the previous assignment

2 Design

There is one receiver module which have almost same logic as the one in previous assignment. The only addition is that it gives a signal received as output which is set '1' when data is completely received and in next cycle it is again set to '0'.

The output of the receiver are the 8 bits sent from the pc which are shown in the seven segment display and also sent to a transmitter module which.

Transmitter rate is same as receiver rate for consistency. The transmitter sends data when it receives a transmit signal as '1'. When this happens transmitter sets the RsTx to corresponding bits to send one by one at approximately the baud rate.

There are 4 states in the receiver.

State : 0

This is idle state and checks if there is any data to transmit, if there is any data to transmit then it sets RsTx to '0' for start bit and transits to state 1 else remains in its own state.

State : 1

This state just wait for the start bit to be completely sent and then transition to state 2.

State : 2

This state is used to transmit all 8 bits of the data one by one at baud rate and then transitions to state 3.

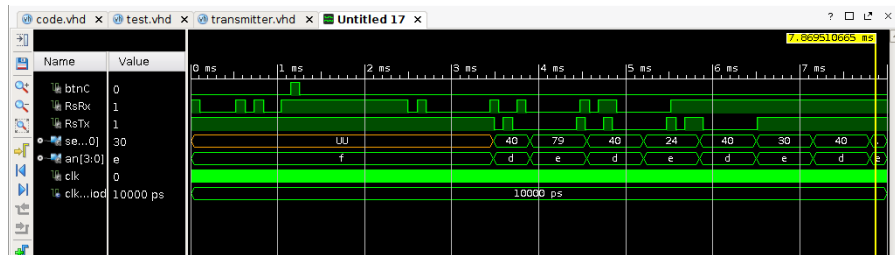
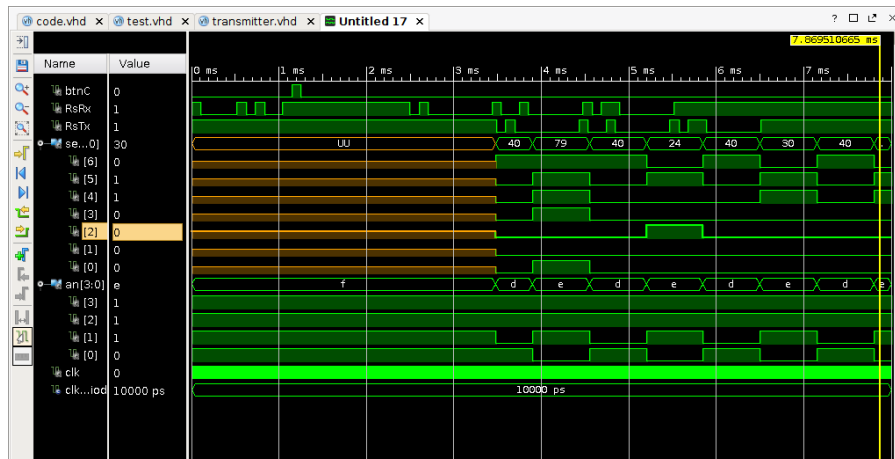
State : 3

This state is to send stop bit and transitions to idle state.

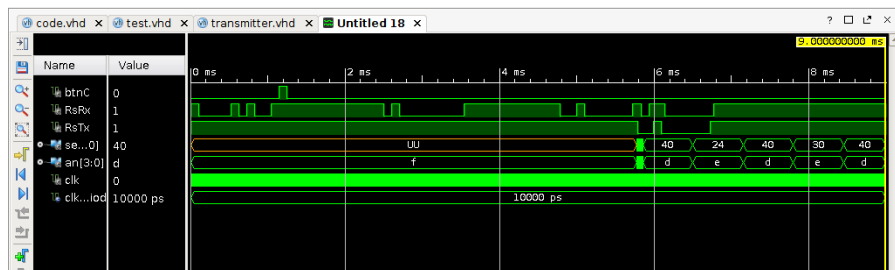
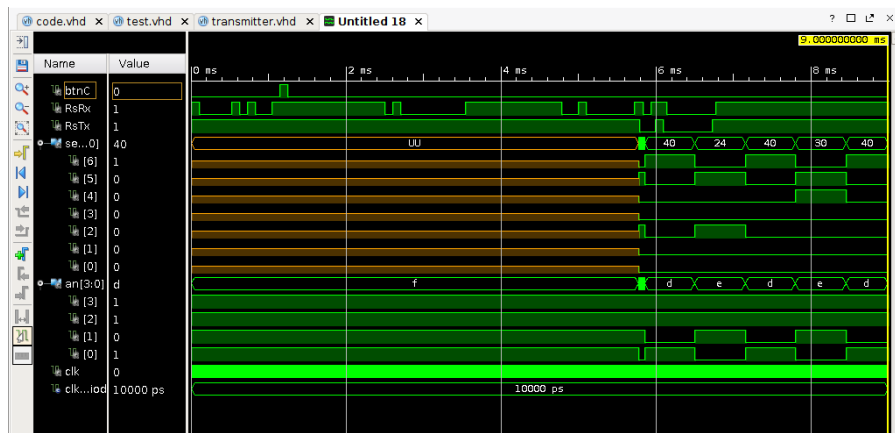
3 Simulation

The code is simulated on test-bench created by the name test.vhd.

The below images show the normal case of data being transmitted from gtkterm and how it is ignored until push button is pressed for reset and then later it continuously receive data and display it in segment. And also the same data ones received is then transferred to transmitter which transmit the bits serially while the receiver could be reading next set of 8 bit data.



The below two images shows how the receiver goes in synchronization state to auto correct things in case there is some problem with the gtkterm software and it does not send the stop bit properly. In this case the receiver also does not echo back the data.



4 Utilisation Report

128 LUT's are used (as logic)

84 registers are used as flip-flops.

15 Input Output Blocks are used (1 RsRx input(data to be received) and 1 clock input(clk) and 7 segment outputs(seg) and 4 display outputs(an) and 1 push button inputs and 1 RsTx output(transmitted data))

2 BUFGCTRL is used.

Ref Name	Used	Functional Category
FDRE	83	Flop & Latch
LUT6	57	LUT
LUT5	22	LUT
LUT4	19	LUT
LUT3	18	LUT
LUT1	17	LUT
LUT2	16	LUT
obuf	12	IO
CARRY4	4	CarryLogic
MUXF7	3	MuxFx
IBUF	3	IO
BUFG	2	Clock
FDSE	1	Flop & Latch

(Utilisation report is also given in the submission.)