International Journal Of Engineering Inventions
E-ISSN: 2278-7461, P-ISSN: 2319-6491
Volume 5, Issue 1 [Oct. 2015] PP: 78-81

The Design of Control Enabled UART

Mahesh d gowdhar[1], ch shekar[2], narendara chary[3]
[1,2,3]*ECE Department,SreeDattha Institute of Engineering & Science*

***Abstract**:-* UART is a simple bus communication system that is used as worldwide transmission for the slow receiver and fast sender. The design of UART can be of the different methods we are using the advance design FSM method for the design of the UART .we introduces the new logic for the devices that which works for the different clocks. Whenever a clock source is driving by the circuit elements like gates and flops that cannot able to drive the cascaded circuit elements generally we need sub systems like clock gating or buffer plane. In order to avoid the turbulences that occurred for the transmitting the data for the different user defined frequencies depend on the receiver we are introducing new concept called control enables. The proposed UART is designed using the VHDL and simulated with model-sim and synthesized by Xilinx

I. INTRODUCTION

Simple asynchronous serial bus that used for the receiving and transmit data between the processor and controllers generally we prefer the UART as the protocol. By this we can easily transmit the parallel data transmission serially with user or design specified rates. UART can be well known, cost effective and simple it efficient for the point to point communications.For the transmission and reception in UART they are different design methods, in present VLSI systems complexity is increases the design issues and more over the features are also gains the transmitting based on the receiver speed by using the baud-rate generators as a part of design.

To the processor, the UART appears as an 8-bit read-write parallel port that performs serial-to-parallel conversions for the processor, and vice versa for the peripheral. The UART allows reliable data transfer at high speeds with its 16-byte first in, first out (FIFO) input register. The FIFO feature can buffer up to 16 bytes at a time, which improves serial communications by preventing data overruns in applications. The implementation of UART the serial communication is done with high data rate and no interrupts. The UART 16550 serial communication interface device receives data and converts data from serial to parallel, where as the transmitter performs parallel to serial conversion.

This paper organizes as follows, session I gives the introduction to the paper session II describes about UART description with the FSM's and parity enable and disables third session deals with the proposed control enables and session IV deals with the results and discussion followed by conclusion

II. UART DATA TRANSMISSION

For the serial asynchronous data communication between the remote embedded systems UART can be used efficiently. This makes a channel to communicate processor/controller to asynchronous transmission

A UART communication module as transmitter it converts the parallel form of data into the serial format by making the frames by word length by adding the star bit, stop bit, sometimes parity based on requirement generally the world length can be 5, 6, 7 or 8 bits the Parity can be odd or even

There are different flavors UART in the industry. Some of them may contain FIFOs in the receiver/transmitter modules and some of have different bit modes like 9 Data bits mode (Start bit + 9 Data bits + Parity + Stop bits). Generally the basic UART design consists of a receiver and transmitter modules. Parallel-to- serial conversion by receiving the 8-bit data from CPU (processor/controller) is performed by the transmitter. Serial-to-parallel conversion is done by the receiver module asynchronously received data frame which sent as the serial data format

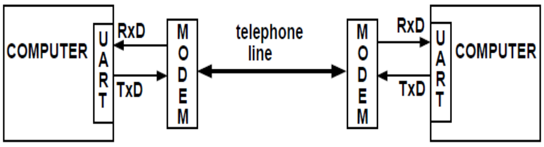

Figure1. Basic Application of UART

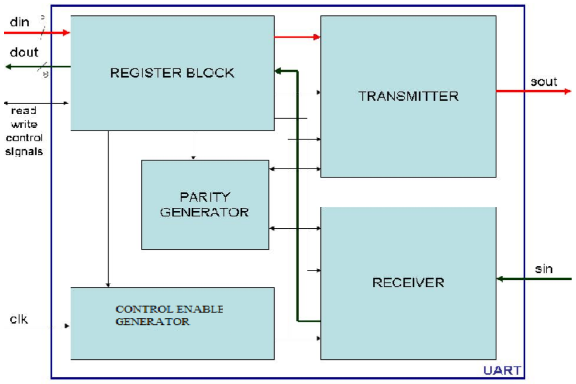

Figure2: Block Diagram of UART

UART module internally consists of register block, parity generator, transmitter and receiver.
Register block is used to store the data temporarily for serial transmission and parallel sending to D_{out}
The transmitter module internally consists of the FSM module which function to transmit the data serially with parity if necessity by parity enable
It internally consists of 12 states

STATE	OPERATION
IDEL	no operation state (system synchronization and wait for request will be done)
START	sends the start bit
DATA0	sends the 0th position bit as output to sout
DATA1	sends the 1th position bit as output to sout
DATA2	sends the 2th position bit as output to sout
DATA3	sends the 3th position bit as output to sout
DATA4	sends the 4th position bit as output to sout
DATA5	sends the 5th position bit as output to sout
DATA6	sends the 6th position bit as output to sout
DATA7	sends the 7th position bit as output to sout
PARITY	provides the parity bit (odd parity of the input 8-bit) based on request of user by enabling the parity enable
STOP	sends the stop bit

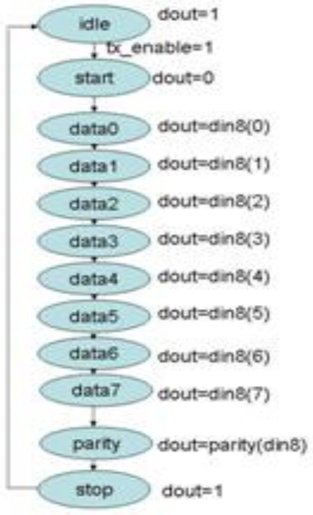

Figure3:Transmitter FSM

The receiver also internally consists of the FSM this receives the data serially and sent as parallel to $_{Dout.}$The receiver operation also similar to transmitter FSM but in receiver FSM receives the each bit in per stateIn receiver parity calculation and comparison can be carried out if the parity enable is high only then it performs the parity otherwise it shifts the particular state and sends the frame to processor/controller.

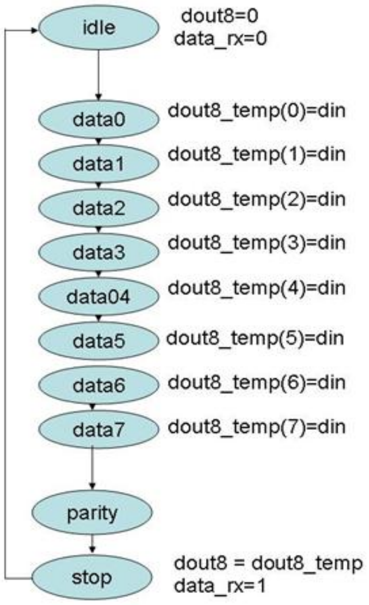

Figure3:Reciver FSM

III. PROPOSED CONTROL ENABLES

In control enable module internally consist of the default counter circuit in it. It generally counts the numbers from '0' to the 'n' N- can be depend on the rate of transmission.The counter table and control enable generation table is shown below

TABLE IControl enable generation table

count	CE1	CE2	CE3
000	0	0	0
001	0	0	1
010	0	1	0
011	0	1	1
100	1	0	0
101	1	0	1
110	1	1	0
111	1	1	1

This control enables are positive edge detected by the edge detection circuit. Only single operation can be performed even though CE value is high for more than one clock cycle.
The edge detection can be performed by the following circuit shown in fig

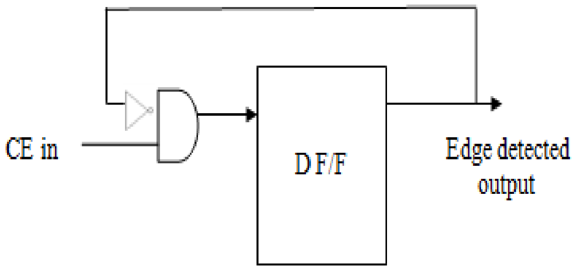

Figure4: edge detection circuit

IV. RESULTS AND DISCUSSION

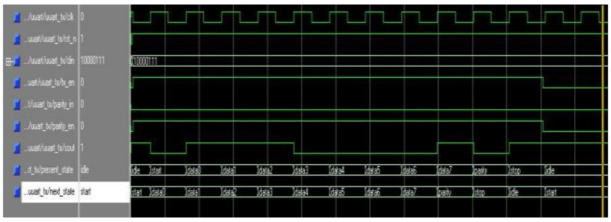
../uuart/uuart_tx/clk 0
..uuart/uuart_tx/rst_n 1
../uuart/uuart_tx/din 10000111
..uart/uuart_tx/tx_en 0
..t/uuart_tx/parity_in 0
../uuart_tx/parity_en 0
..uuart/uuart_tx/sout 1
..rt_tx/present_state idle
..uuart_tx/next_state start
10000111
idle start data0 data1 data2 data3 data4 data5 data6 data7 parity stop idle
start data0 data1 data2 data3 data4 data5 data6 data7 parity stop idle start

Figure 5 UART transmitter waveforms

Figure5 shows the waveforms of the UART transmitter. The present state indicates the current state of the state machine. It traverses from IDLE to STOP state. The data input can be seen on din and corresponding serial output is given on sout. Since parity_en = 1, parity bit is appended to data**.**

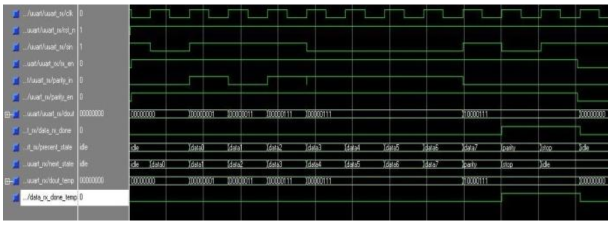
.../uuart/uuart_rx/clk 0
..uuart/uuart_rx/rst_n 1
.../uuart/uuart_rx/sin 1
..uart/uuart_rx/rx_en 0
..t/uuart_rx/parity_in 0
.../uuart_rx/parity_en 0
..uuart/uuart_rx/dout 00000000
..t_rx/data_rx_done 0
..rt_rx/present_state idle
..uuart_rx/next_state idle
..uuart_rx/dout_temp 00000000
.../data_rx_done_temp 0
00000000 00000001 00000011 00000111 00000111 10000111 00000000
idle data0 data1 data2 data3 data4 data5 data6 data7 parity stop idle
idle data0 data1 data2 data3 data4 data5 data6 data7 parity stop idle

Figure 6 UART receiver waveforms

Fig. 6 shows the receiver waveforms. The state transitions are similar to transmitter. The serial data input comes on sin and output data is dout. The data is sampled when data_rx_done = ‘1’.

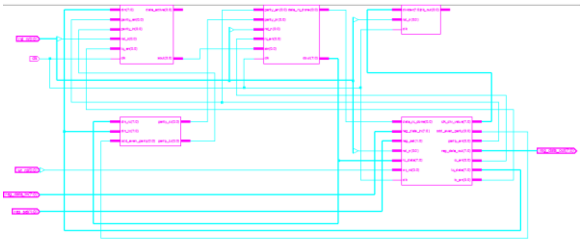

Figure 7 Blocks inside the Developed Top Level UART Design

The internal blocks available inside the design includes parity generator, control enable, register block, transmitter and Receiver which were clearly shown in the above schematic level diagram. Inside each block the gate level circuit will be generated with respect to the modelled HDL code.

V. CONCLUSION

UART can be designed by the FSM’S and the operation can be performed with synchronized control enables which can makes transmission easy with the different frequencies and baud rates efficiently without clock gating and internal buffers. This will be appropriate for the ASIC and FPGAs, more efficient for the FPGAs.

REFERENCES

[1]. Ritesh Kumar Agrawal and Vivek Ranjan Mishra “The Design of High Speed UART” 2013 IEEE Conference on Information and Communication Technologies

[2]. Charles H. Roth, Jr, Digital System Design by using VHDL, PWS Publishing Company, 1998

[3]. Samir Palintkar, Verilog HDL A Guide to Digital Design and Synthesis, Second Edition, Pearson Publication, 2011

[4]. Tomasi, Wayne, Advanced electronic communication systems, Third Edition, Prentice-Hall, United States of America, 1994

[5]. Michael D. Ciletti, Advanced Digital Design with the Verilog HDL, Second Edition, Prentice Hall, Second edition