Verification of Universal Asynchronous Receiver Transmitter by Mentor Graphics Tool

[1]M.Vimala, [2]Ajithkumar, [3]Hariharan, [4]Kannan
[1]Assistant Professor, [2,3,4]UG students
[1,2,3,4] Department Of Electronics And Communication Engineering,
P.S.R Engineering College, Sivakasi (India)

Abstract: Universal Asynchronous Receiver Transmitter (UART) is the serial communication protocol that is used for data exchange between computer & peripherals. UART is a low velocity, short- distance, low-cost protocol. UART includes three modules which are received, the baud rate generator and transmitter. The UART design with Very High Description Language can be integrated into the Field Programmable Gate Array to achieve stable data transmission and to make system reliable and compact. In the result and simulation part, this project will focus on check the receive data with error free & baud rate generation at different frequencies. Before synthesizing of UART a baud rate generator is incorporated into the system. We use the frequency divider which sets itself to required frequency for the functionality at lower frequency. All modules are designed using VERILOG and implemented on Xilinx Suite development board. And Verilog methodology to be created testbench of UART and timing value of given verified operation of data correctly passed or not by mentor graphics tool

KEYWORDS: UART, mentor graphics tool, transmitter, receiver, baud rate, Verilog

I.INTRODUCTION

A universal asynchronous receive/transmit (UART) is an integrated circuit which plays the most important role in serial communication. It handles the conversion between serial and parallel data. Serial communication reduces the distortion of a signal, therefore makes data transfer between two systems separated in great distance possible. It contains a parallel-to serial converter for data transmitted from the computer and a serial to parallel converter for data coming in via the serial line. The baud rate generator is used to produce a local clock signal which is much higher than the baud rate to control the UART receive and transmit; The UART receiver module is used to receive the serial signals at RXD, and convert them into parallel data; The UART transmit module converts the bytes into serial bits according to the basic frame format and transmits those bits through TXD.

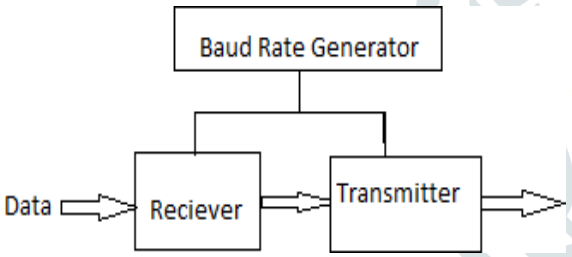


FIG 1. UART Module

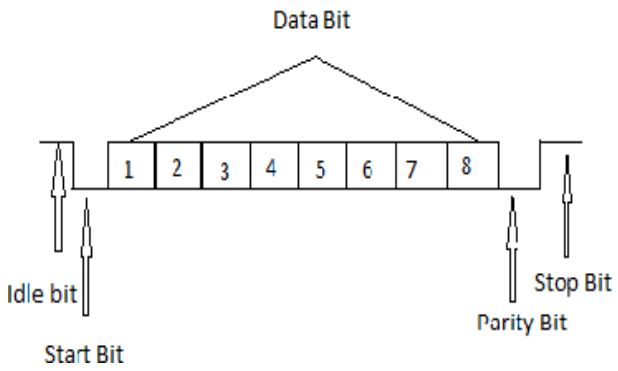
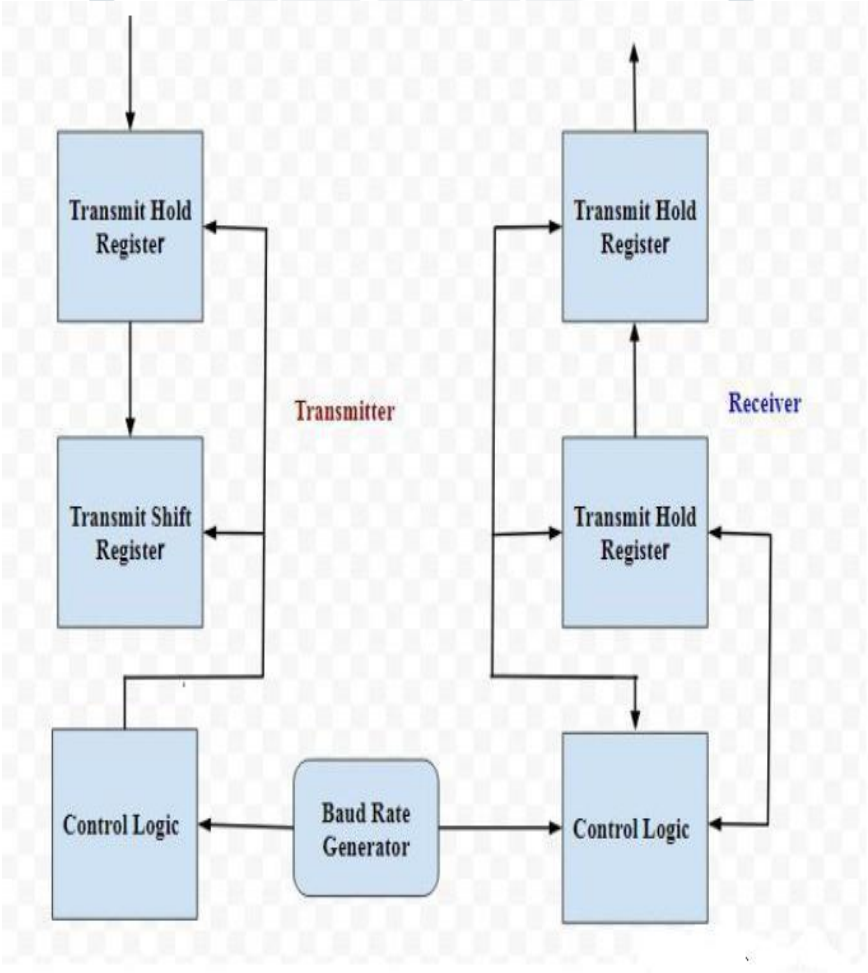


Fig 2.UART FRAME FORMAT

It usually includes start bit, data bit, parity bit, stop bit and idle state as shown in fig 2.When a word is given to the UART for Asynchronous transmissions, a bit called the "Start Bit" is added to the beginning of each word that is to be transmitted. The Start Bit is used to alert the receiver that a word of data is about to be sent, and to force the clock in the receiver After the Start Bit, the individual bits of the word of data are sent, with the Least Significant Bit (LSB) being sent first into synchronization with the clock in the transmitter. When the entire data word has been sent, the transmitter may add a Parity Bit that the transmitter generates. The Parity Bit may be used by the receiver to perform simple error checking. Then at least one Stop Bit is sent by the transmitter. If incorrectly formatted data is received, the UART may signal a framing error. If another byte is received before the previous one is read, the UART will signal an over run error.

II. METHODOLOGY

- The UART block diagram consists of two components namely the transmitter & receiver that is shown below.
- The transmitter section includes three blocks namely transmit hold register, shift register and also control logic. Likewise, the receiver section includes a receive hold register, shift register, and control logic.
- These two sections are commonly provided by a baud-rate-generator. This generator is used for generating the speed when the transmitter section & receiver section has to transmit or receive the data.
- The hold register in the transmitter comprises the data-byte to be transmitted.
- The shift registers in transmitter and receiver move the bits to the right or left till a byte of data is transmitted or received. A read (or) write control logic is used for telling when to read or write.



- The baud-rate-generator among the transmitter and the receiver generates the speed that ranges from 110 bps to 230400 bps. Typically, the baud rates of microcontrollers are 9600 to 115200.
- For this, simply two cables are required to communicate between two UARTs. The flow of data will be from both the transmitting (Tx) & receiving (Rx) pins of the UARTs.
- The data transmission of a UART can be done by using a data bus in the form of parallel by other devices like a microcontroller, memory, CPU, etc.
- After receiving the parallel data from the bus, it forms a data packet by adding three bits like start, stop and parity.
- It reads the data packet bit by bit and converts the received data into the parallel form to eliminate the three bits of the data packet.
- In conclusion, the data packet received by the UART transfers in parallel toward the data bus at the receiving end.

III. CALCULATIONS OF UART:

Calculation of frequency:

- The UART transmitting frequency=10MHZ
- The UART TX TO RX FREQUENCY= 1.4MHZ
- The UART data bit transmission = 8 bits
- The UART 1 cycle transmission = 100 ns
- The UART 7 cycles transmission = 7*100=700 ns

1. **Find the frequency of transmitter to receiver: FREQUENCY=1/T = 1/7*10~7**
   = 0.14*10~7
   = 1.410 MHZ
   The transmitting frequency of one-bit data transmitter to receiver is **1.410mhz**

Calculation of frequency:

- **CALCULATION OF BAUD RATE**:

- Baud rate is defined as no of samples as per second in tx data
- In the serial port context, "9600 **baud**" means that the serial port is capable of transferring a maximum of 9600 bits per second.
- The transmitting a pass the data 1 sec=9600 samples
- The transmitting 11 bit of data pass through per seconds

2. FIND THE PASSING TIME OF TRANSMITTING DATA TO RECEIVER

- The 1 bit of data to passing per sec

  =**1/9600 seconds**

  =1.04*10~4 seconds

- The 11 bits of data to transmitting per sec

  = 1.04*10~4*11

  = 114.5*10~6 seconds

- the 11bits of data how much frame pass in 1 sec

  = 1/11 bit of data pass per sec

  = 1/114.5*10~6

  = 872.72 seconds

- the baud rate is calculated to transmitted how many frames

  =**baud rate/11 bits of data frame per sec**

  = 9600/827.72

  = 11 frame

Application design of uart

From Careful study of reported work, it is observed that researchers have proposed various techniques to design the UART chip and to improve its characteristics. Considering all the advantages suggested by different researchers we are defining an application of UART where there is serial communication of key board characters from PC hyper terminal through RS232 serial port using UART controller will be display on the HD44780 based 16x2 LCD display using LCD controller. The hardware implementation of UART Controller and LCD driver would be based on VHDL language. The proposed block diagram is shown below:

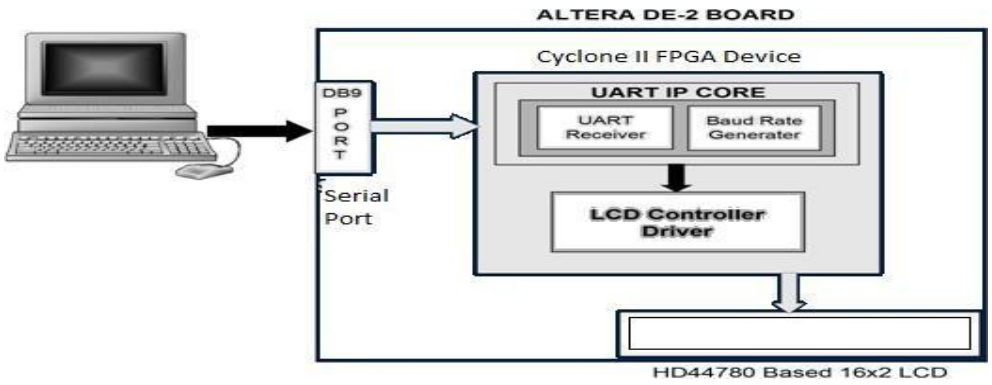
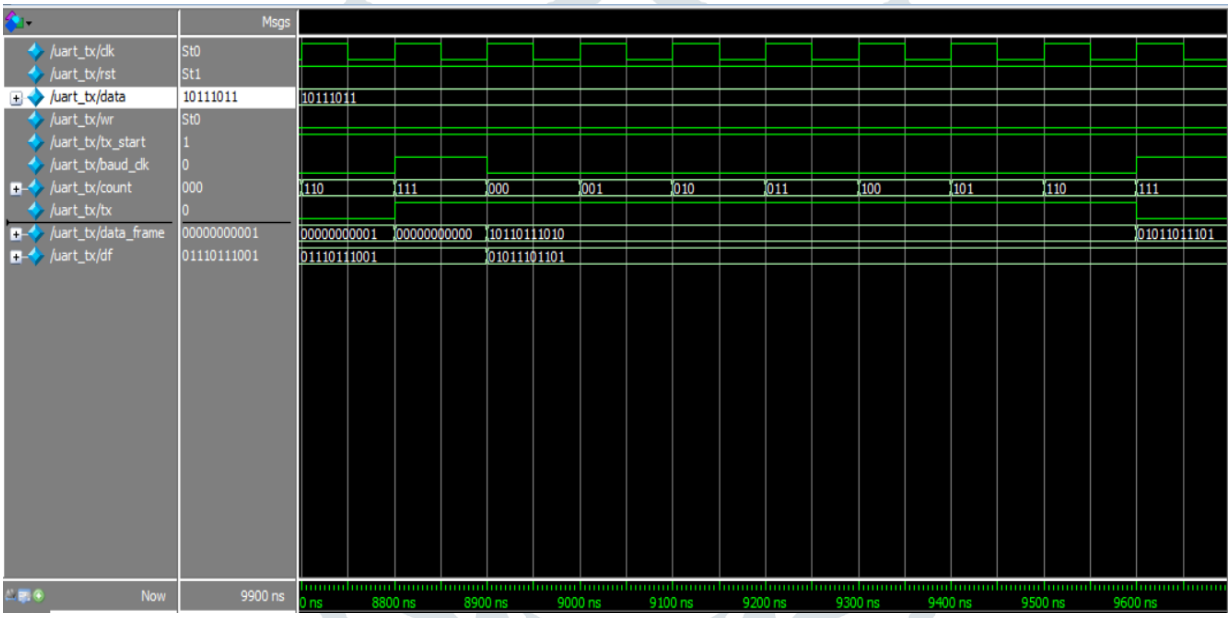


Fig. 3 Application design of UART

IV. RESULTS AND DISCUSSION:



Transmitter module:

The operation of transmit module is conversion of the sending 8-bit parallel data into serial data, adds start bit at the MSB of the data as well as the parity and stop bits at the LSB of the data. When the UART transmit module is reset, the transmit module immediately gets activated to send the data. The output appears as 1 start bit, 8 data bits, 1 parity bit and 1 stop bit. The parity bit is specified as the output. Finally, stop bit displays logic 1.

Depending on the manufacturer, specifies the different representations for UARTs Intel called their 8251 device as a "Programmable Communication Interface". The technique cannot send or receive data at high speed, but provides level of compatibility.

- When the transmitter fn block should given RST=0; UART module is off state condition
- When RST=1, the UART module is on state suppose data to be transmitted is 8 bit of data **10101011**
- UART transmitter is depends on the read and write console operation pins
- When WR=0 UART going to transmitting operation

- When WR=0 and RST=1, both the conditions satisfied UART transmitting and count ready to transmit

- When WR=0 & RST=1, the data 10101011 to store inside the temp reg data frame
- Inside the data frame UART set in frame
- Data frame is 11 bits to looks at the function set

- When the start bit always set at 0. Because the receiving shift operation is done
- When the 2 to 9 data bit to added the frame
- When 10th bit is parity bit

Parity bit=(^data in[xor operation]) Parity bit=1^0^1^0^1^0^1^1

- Data having read NO. OF ODD DATA=SET P=1
- Data having read NO OF EVEN DATA =SET P=0
- Last stop bit=1
- Alternating setting of data frame

0	10101011	1	1

- All the bits set reversed position for state of ready to transmit the data

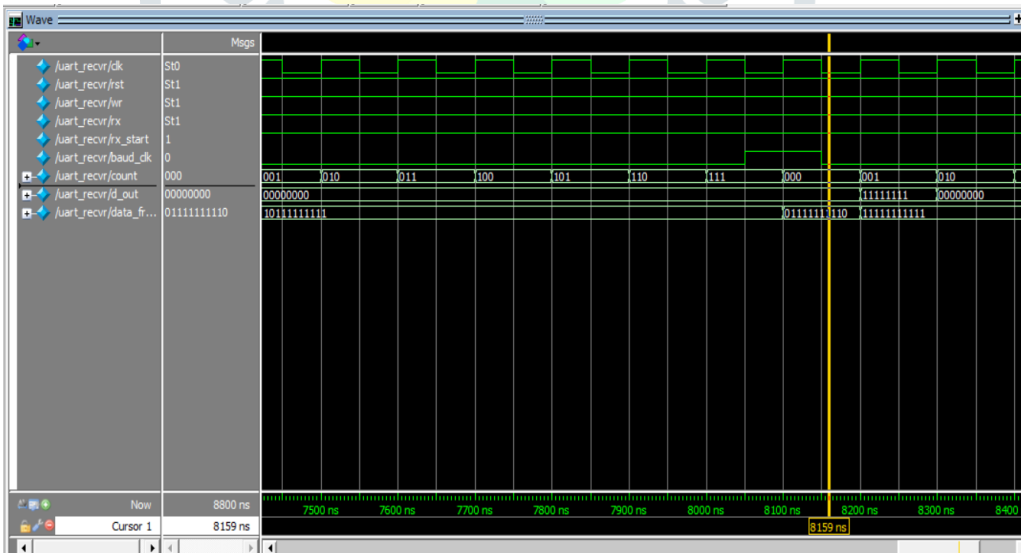
1	110101010	1	0

- All the data transmitter completed data frame is shifted and all are 0

0	00000000	0	0

- Next the data frame is added the load the data and transmit the value

RECEIVER



During the UART reception, the serial data and the receiving clock are asynchronous, so it is very important to correctly determine the start bit of a frame data. RXD transfers into logic 0from logic 1which resembles as the beginning of a data frame. Once the start bit gets identified, next bit begin to count the rising edge of the baud clock, and when counting considers the sample RXD. Each value of the logic level gets placed in the register r buffer [7, 0] in order. When the count equals 8, we can ensure that all the data bits are received, and are converted into a byte parallel data. UARTs resynchronize the internal clocks on change of the data line. Due to re synchronisation they reliably receive when the transmitter is sending at a slightly different speed than it should. The other UARTs do not support this, and they resynchronize on the falling edge of the start bit only, and then read the center of each expected data bit, and this works if the broadcast data rate is accurate for allowance of the stop bits for reliability. Three error detection signals are commonly used in UART: 1. Parity Error used to justify the whether there exists even or parity by focussing on 1's. 2. Overrun Error specifies whether data is overwritten than expected. 3. A framing error occurs when the designated start and stop bits are not valid.

- When the receiver fn block should given RST=0; UART module is off state condition
- When RST=1, the UART module is on state suppose data to be transmitted is 11 bit of data
  11010101110
- UART transmitter is depends on the read and write console operation pins
- When WR=1 UART going to receiving operation
- When WR=1 and RST=1, both the conditions satisfied UART receiving and count ready to receive
- When WR=1 & RST=1, the data 1101010101 to store inside the temp reg data frame
- Inside the data frame UART set in frame
- Data frame is 11 bits to looks at the function set

Start bit	8 bit of data	Priority bit	Stop bit

- When the start bit always set at 0. Because the receiving shift operation is done
- When the 2 to 9 data bit to added the frame
- When 10th bit is parity bit
  Parity bit=(^data in[xor operation]) Parity bit=1^0^1^0^1^0^1^1
- Data having read NO. OF ODD DATA=SET P=1
- Data having read NO OF EVEN DATA =SET P=0
- Last stop bit=1
- Alternating setting of data frame

0	10101011	1	1

- All the bits set reversed position for state of ready to receive the data

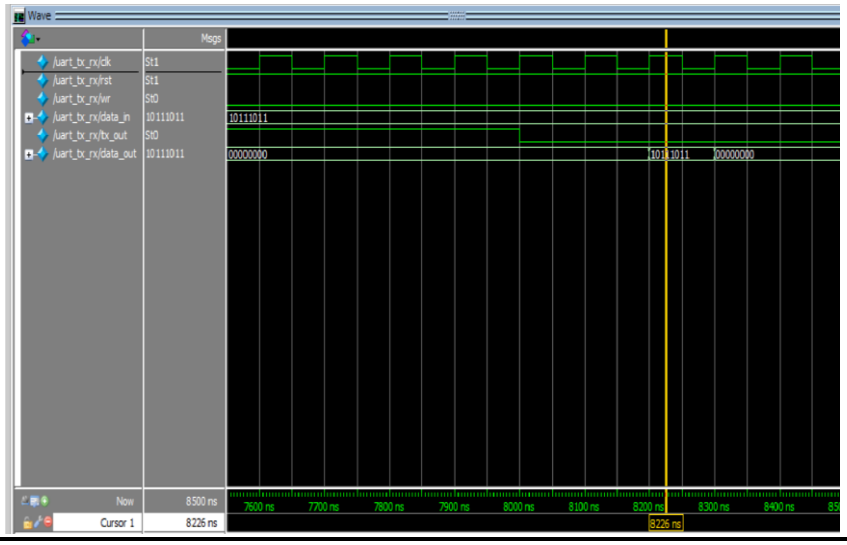
1	110101010	1	0

- All the data receiving completed data frame is shifted and all are 0

0	00000000	0	0

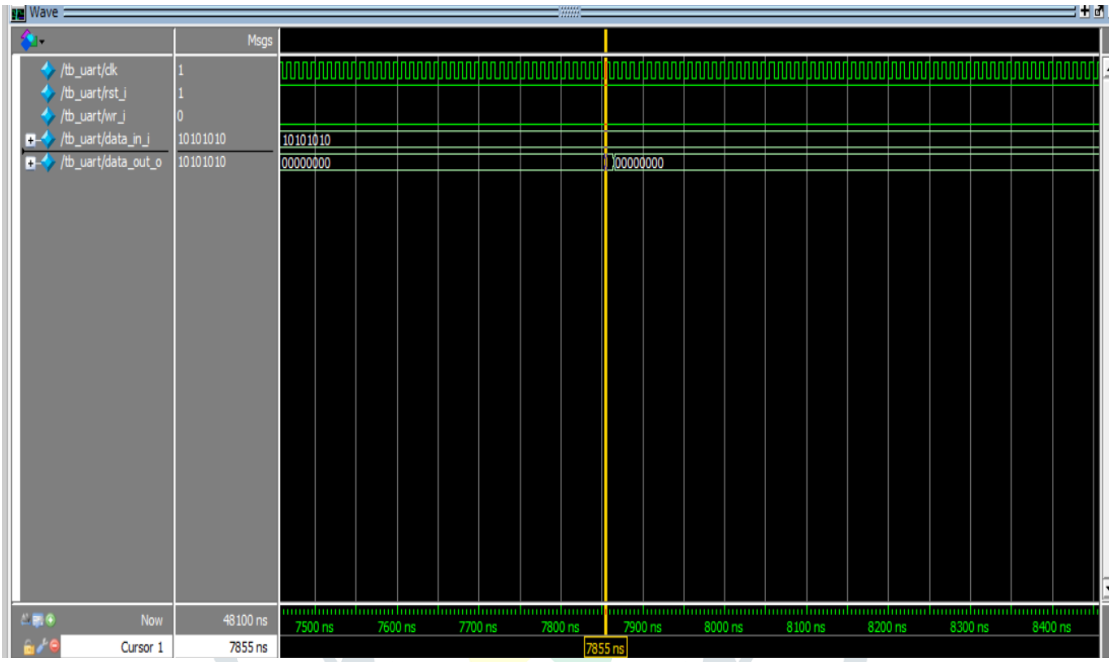
- Next the data frame is added the load the data and receive the value

INTERFACING:



- The following figure shows UART interfacing with a microcontroller. The UART communication can be done using three signals like TXD, RXD, and GND.
- By using this, we can exhibit a text in personal computer from 8051 microcontroller board as well as the UART module.
- In 8051 board, there are two serial interfaces such as UART0 and UART1. Here, UART0 interfacing is used.
- The Tx pin transmits the information to PC & Rx pin receives the information from PC.
- Baud rate can be used to denote the speeds of both the microcontroller and PC.
- The data transmission and reception can be done properly when the baud rates of both microcontroller & PC are similar

VERIFICATION OF UART:



OUTPUT OF UART_PROTOCOL:

 # Loading work.tb_uart

 # Loading work.uart_tx_rx # Loading work.uart_recvr # Loading work.uart_tx

run
 # 1695,matched ,d_exp=11100100,data_out_o=11100100

run
 # 2575,matched ,d_exp=11110000,data_out_o=11110000

run
 # 4335,matched ,d_exp=10101010,data_out_o=10101010

V. CONCLUSION:

A survey of various method for implementation of serial communication through UART is done in this paper. From the reported work we are designing an application that explore usage of UART to achieve benifits like great flexibility, low cost, high performance logic solutions and also meet communication demands quickly and efficiently.This design shows great significance especially in the field of electronic design, where SOC technology has recently become increasinglymature.

REFERENCE:

[1] Design implementation of and SPI in single FGPA ,M. Poorani ; R. Kurunjimalar2016 10th International Conference on Intelligent Systems and Control (ISCO)

[2] Optimizationof communication based on LEON2 hardware debug support unitMintao Liu ; Jianyang Zhou2008 2nd International Conference on Anti-counterfeiting, Security and Identification

[3] Fast for Scalable IoT PlatformRizka Reza Pahlevi ; Aji Gautama Putrada S ; Maman Abdurohman2018 6th International Conference on Information and Communication Technology (ICoICT)

[4]A new approach to realize Yongcheng Wang ; Kefei Song Proceedings of 2011 International Conference on Electronic & Mechanical Engineering and Information Technology ,2011

[5] Design and implementation of a BIST embedded high speed RS-422 utilized over FPGAShumit Saha ; Md. Ashikur Rahman ; Amit Thakuro