Assignment -7

1) *Introduction*

In this lab we have built a Receiver in which FPGA board receive 8 bits data written on "gtkterm screen". Gtkterm is a software which is used for manually providing input so that the "Receiver" can read that data.

UART (Universal Asynchronous Receiver Transmitter) is used as an interface for communication between Basay 3 board and PC.

Some important points regarding UART :

a) UART is a serial interface , so transmitter send data bit by bit and at same time receiver has to capture each bit and convert it into parallel data.

b) Baud rate is a standard feature of URAT which tell us number of bits transfer per second.

c) URAT interface allow use of different Baud rate like 300, 600 , 900, 1200, 9600, 115200 *etc.*

d) Similar to Baud rate there are different variants parameter like 'Data bits' , 'Parity' , 'Start bits' and 'Stop bits'.

Some basic feature of Receiver :

a) Serial − in Parallel − out register

b) Shift the bits serially into a register

c) Read out data in parallel

Some important diagram related to URAT :

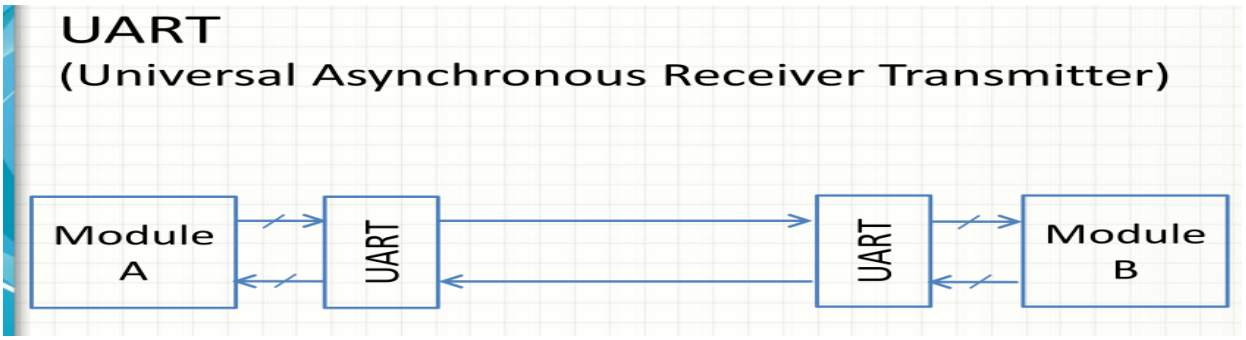
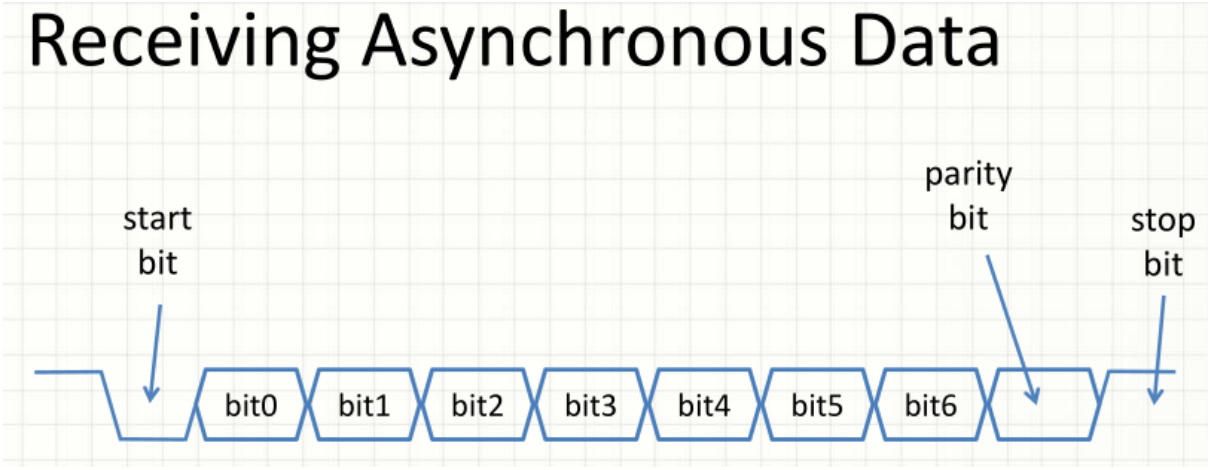


Fig 1 : Basic diagram of URAT



UART interface time diagram

Fig 2: Time diagram along with different parameters

2) *Implementation Design*

In this lab we have implement a Finite State Machine (FSM) /State Machine which is based on bit − data received by Receiver Input pin (rx − in) . We have also used FPGA clock of 100MHz as clock for the receiver (rx − clock), reset pushButton is also used so that 'idle' state can be reached.

To sample 8 bit data we have used counter(temp − count) and with the help of new − clk we are sampling the middle of each data bit .

Calculation of counter :

i) Rx − clock frequency = 100 MHz.
ii) Baud rate = 9600
iii) Number of clock cycle per bit = 16

As we are using middle data bit sampling method:

$$counter = \frac{(Rx-clock\ Frequency)}{(Baud\ rate)*(No.of\ clock\ cycle\ per\ bit)*2} = \frac{10^8}{(9600*16*2)} = 351$$

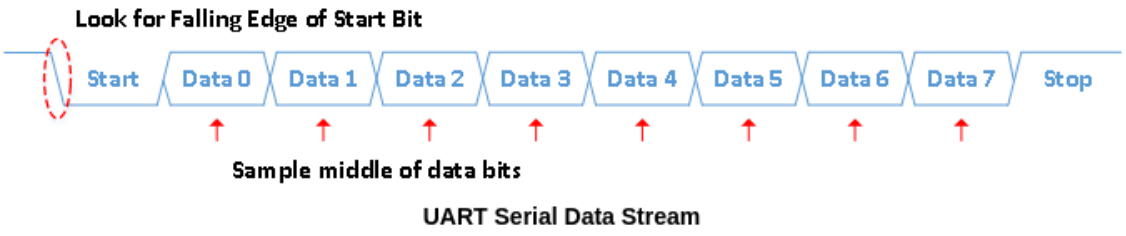


Fig 3: middle method of data bit sampling

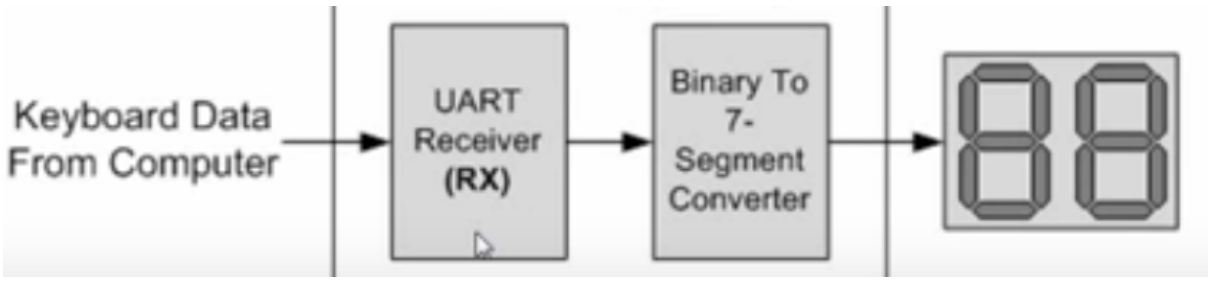


Fig 4 : *Basic Diagram of Circuit of Assignment* − 8

State Diagram (*FSM*) *Description of Receiver* :

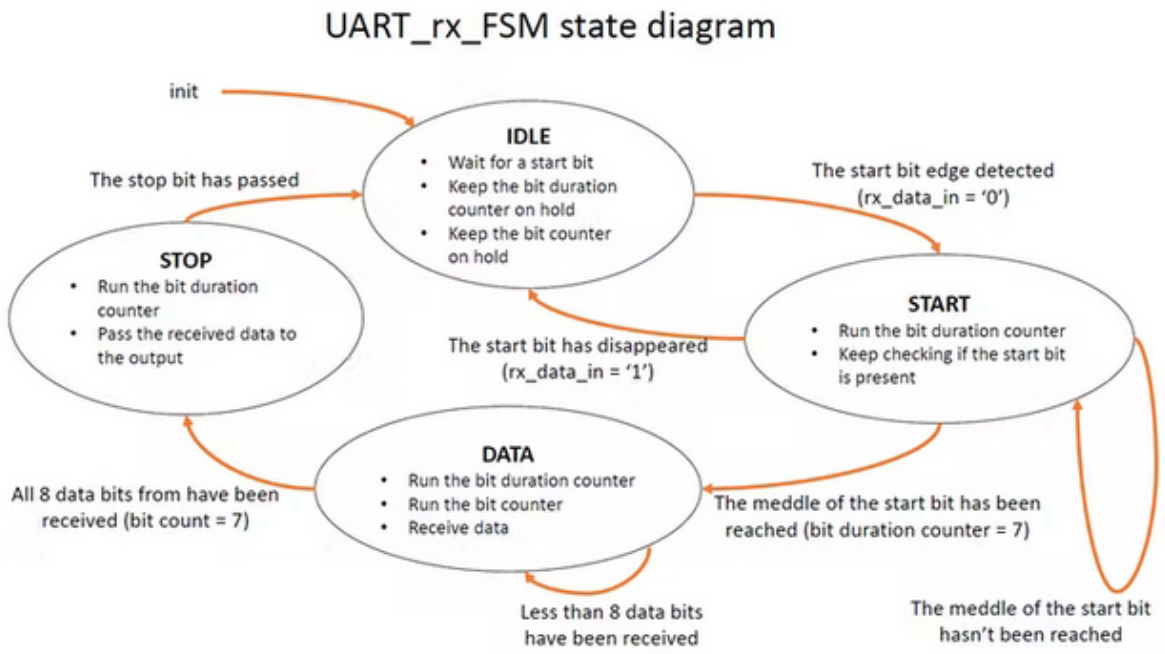
4 *States of FSM* : *Idle*, *Start* , *Read*, *Stop*

a) *Idle* : *If the* $rx - in =$ '1' *then state will be in Idle state and in this state receiver will not be reading any bit. If the "receiver* $-$ *input"* ($rx - in$) *value changes from* 1 *to* 0 *then "Idle" state will transition to the "Start" state.*

b) *Start* : *If the value of "*$rx - in$*" remains* '0' *for* 8 *consecutive cycles of "receiver clock"*($rx - clock$) *then it get identified as "Start state" and now state will change to "Read state". If* 8 *consecutive* '0' *is not received then the state will change to "Idle state" from "Start state".*

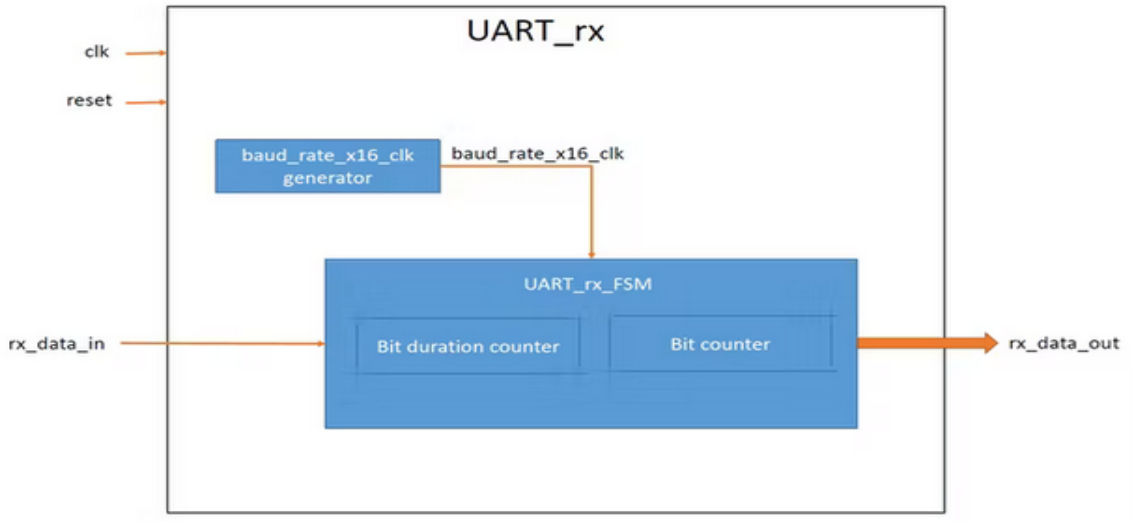
c) *Read* : *In this state* "8 *bit data*" *is read by Receiver and in every* 16 *clock cycle* 1 *bit is read. After* 8 *bit data reading the state will change to "Stop state".*

d) *Stop* : *It is a kind of "end state" which describe that* 8 *bit consecutive data is successfully read by Receiver. After completion of* 16 *clock cycle this state will change to "Idle State".*



The state diagram of the UART_rx_FSM

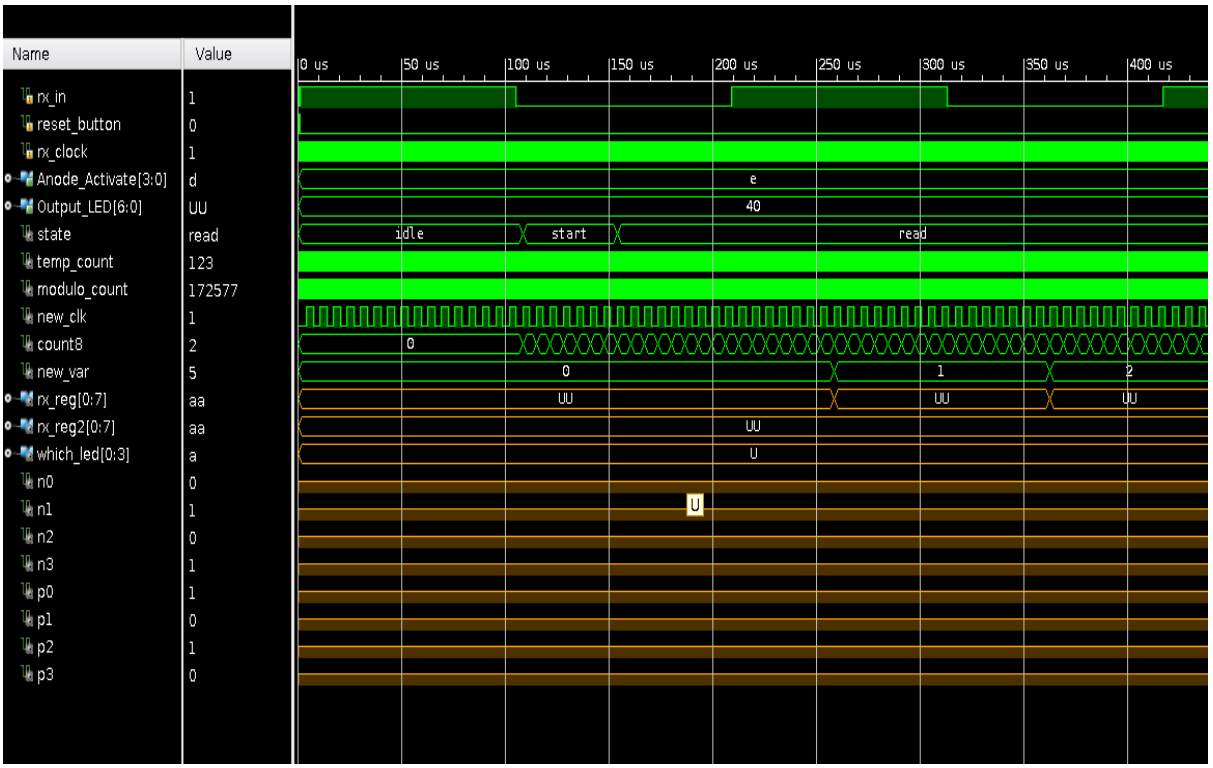
Fig 5 : Receiver FSM diagram



Block diagram of the UART_rx.vhd file

Fig 6 : Block Diagram of Receiver

3) *Simulation Waveform for UART Receiver*



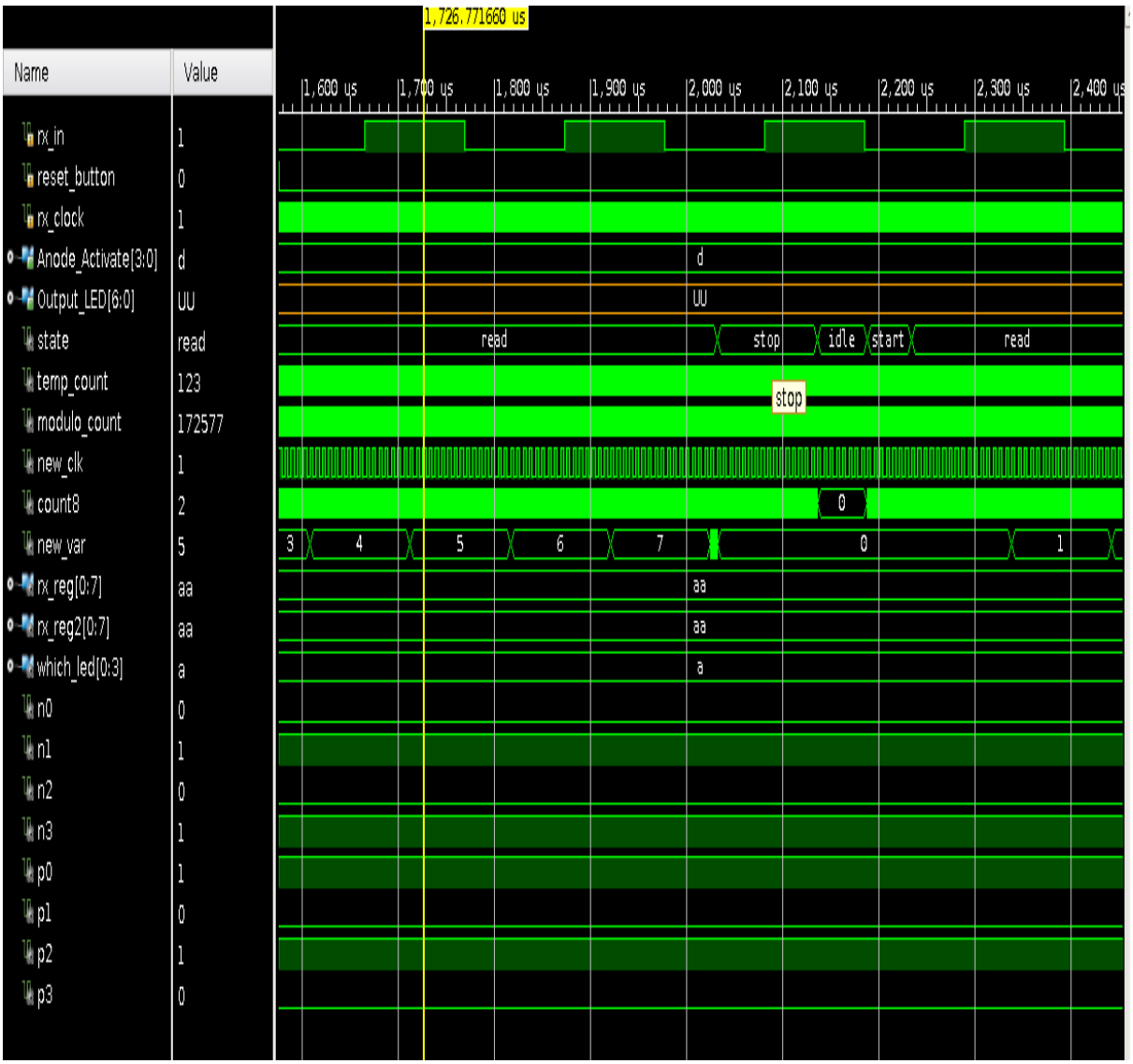


Fig 7 : Waveform for Receiver at different input and parameter

4) *Digital Circuit for URAT Receiver*

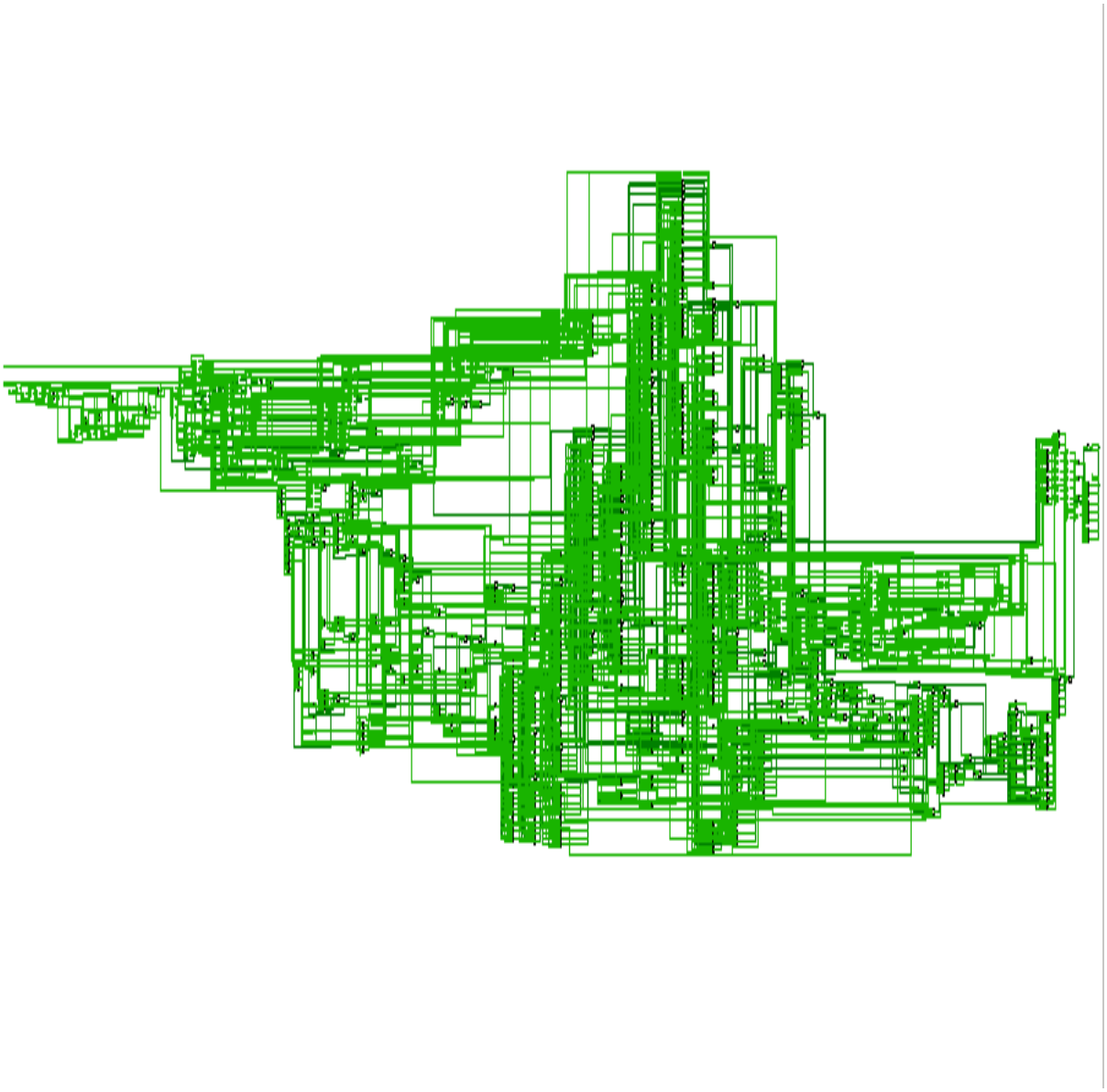


Fig 8: *Complex Digital Circuit for Receiver*

5) *Resource Utilisation*

a) LUT Memory =0
b) LUT logic = 592
c) DSP =0
d) Flip Flops =125
e) BRAM = 0

6) *Some other relevant diagram for resource utilisation*

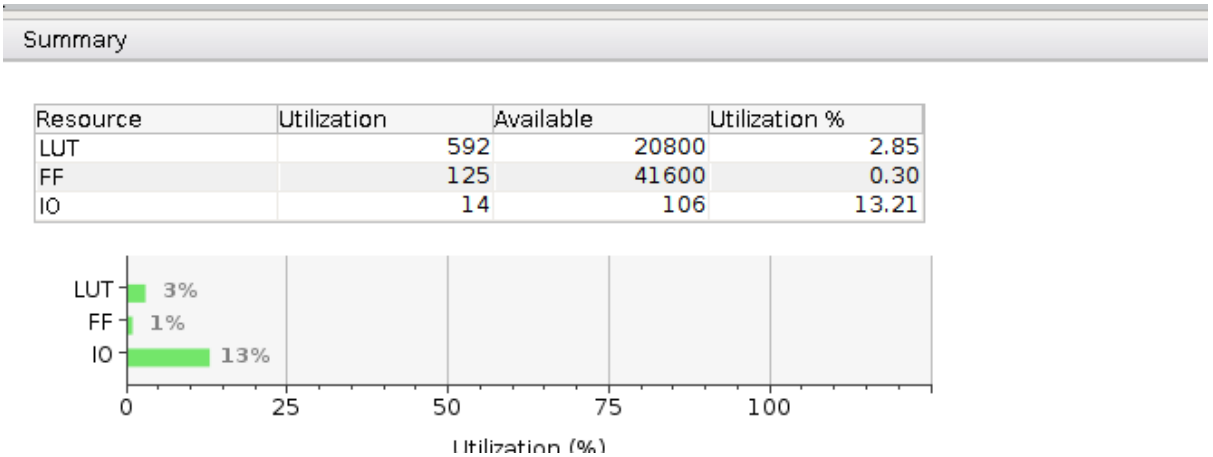


Fig 9 : Summary

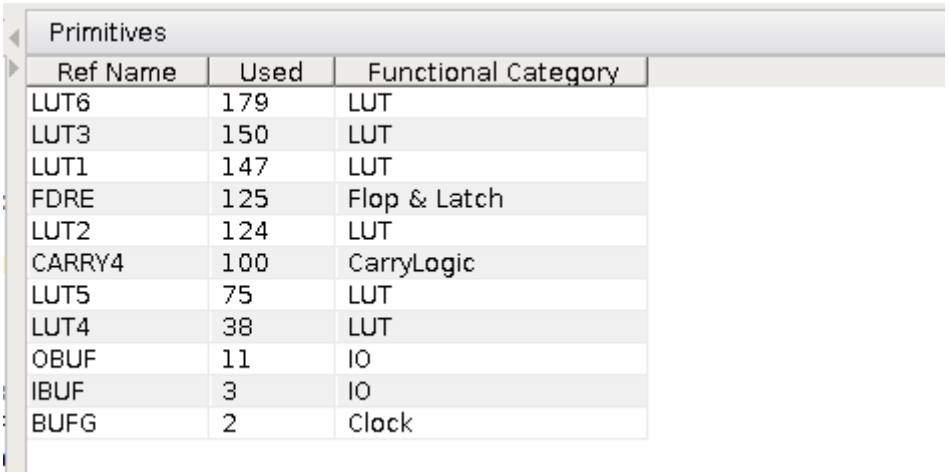


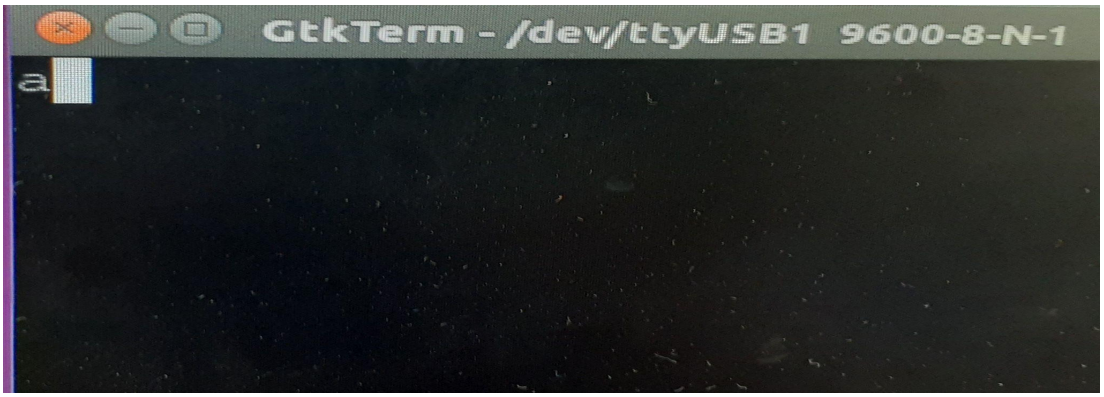
Fig 10 : Primitives

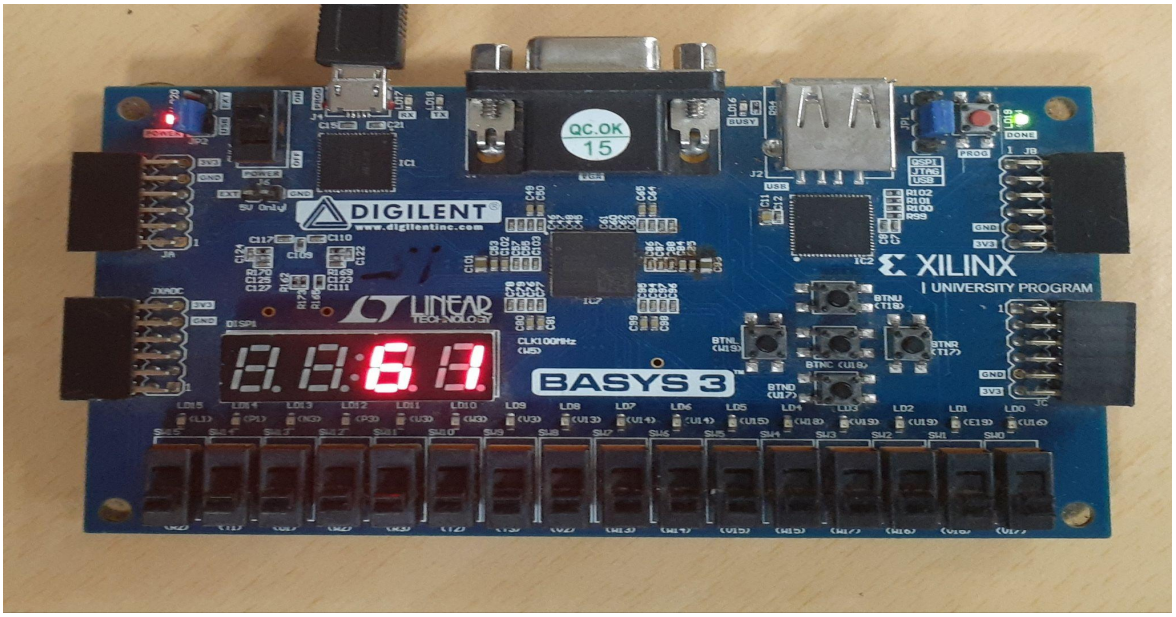
Hierarchy

Name	Slice LUTs (20800)	Slice Registers (41600)	Bonded IOB (106)	BUFGCTRL (32)
Receiver	592	125	14	2

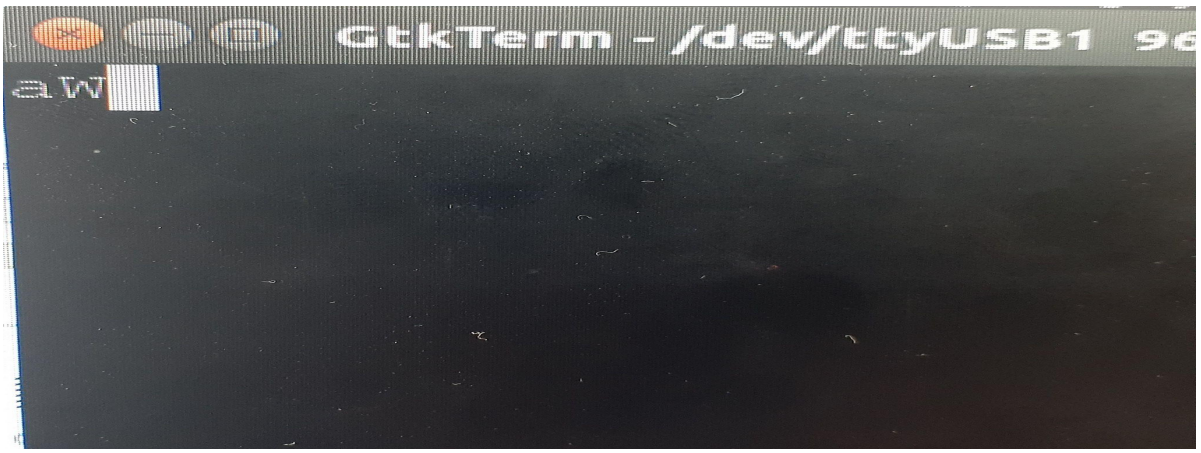
Fig 11 : *Hierarchy*

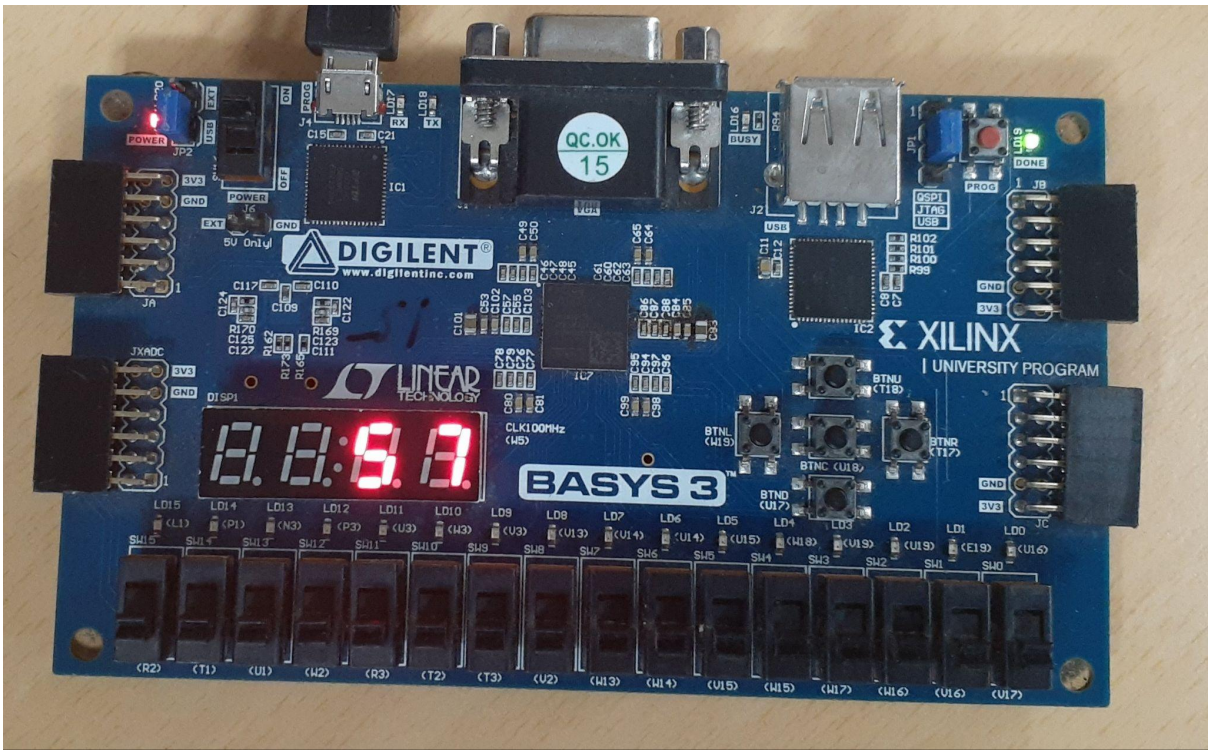
7) *Some photographs of FPGA Board*





(a)


GtkTerm - /dev/ttyUSB1


DIGILENT
www.digilentinc.com
XILINX
UNIVERSITY PROGRAM
LINEAR TECHNOLOGY
BASYS 3
QC.OK
15

(b)