UART_baudrate_generator.v

/home/itzzinfinity/Cozy Drive/100daysofRTL/UART_trial/UART_trial.srcs/sources_1/new/UART_baudrate_generator.v

```
1
2   module UART_BaudRate_generator(
3       Clk                     ,
4       Rst_n                   ,
5       Tick                    ,
6       BaudRate
7       );
8
9   input           Clk                 ;
10  input           Rst_n               ;
11  input [15:0]    BaudRate            ;
12  output          Tick                ;
13  reg [15:0]      baudRateReg         ;
14
15
16  always @(posedge Clk or negedge Rst_n)
17      if (!Rst_n) baudRateReg <= 16'b1;
18      else if (Tick) baudRateReg <= 16'b1;
19           else baudRateReg <= baudRateReg + 1'b1;
20  assign Tick = (baudRateReg == BaudRate);
21  endmodule
22
23
```

UART_master.v

/home/itzzinfinity/Cozy Drive/100daysofRTL/UART_trial/UART_trial.srcs/sources_1/new/UART_master.v

```
1   module UART_master(
2         Clk,
3         Rst_n,
4         Rx,
5         Tx,
6         RxData,
7     );
8
9   //////////////////////////////////////////////////////////////////////////////////
10  input           Clk             ;
11  input           Rst_n           ;
12  input           Rx              ;
13  output          Tx              ;
14  output [7:0]    RxData          ;
15  //////////////////////////////////////////////////////////////////////////////////
16  wire [7:0]      TxData      ;
17  wire            RxDone      ;
18  wire            TxDone      ;
19  wire            tick        ;
20  wire            TxEn        ;
21  wire            RxEn        ;
22  wire [3:0]      NBits       ;
23  wire [15:0]     BaudRate    ;
24
25  assign    RxEn = 1'b1 ;
26  assign    TxEn = 1'b1 ;
27  assign    BaudRate = 16'd325;   //baud rate set to 9600
28  assign    NBits = 4'b1000 ;
29
30
31
32
33
34  //////////////////////////////////////////////////////////////////////////////////
35
36  UART_rs232_rx I_RS232RX(
37        .Clk(Clk)               ,
38          .Rst_n(Rst_n)         ,
39        .RxEn(RxEn)             ,
40        .RxData(RxData)         ,
41        .RxDone(RxDone)         ,
42        .Rx(Rx)                 ,
43        .Tick(tick)             ,
44        .NBits(NBits)
45     );
46
47
48
49
50
```

UART_master.v

/home/itzzinfinity/Cozy Drive/100daysofRTL/UART_trial/UART_trial.srcs/sources_1/new/UART_master.v

```
50
51
52
53
54
55
56
57
58
59
60      UART_rs232_tx I_RS232TX(
61          .Clk(Clk)               ,
62          .Rst_n(Rst_n)           ,
63          .TxEn(TxEn)             ,
64          .TxData(TxData)         ,
65          .TxDone(TxDone)         ,
66          .Tx(Tx)                 ,
67          .Tick(tick)             ,
68          .NBits(NBits)
69          );
70
71      UART_BaudRate_generator I_BAUDGEN(
72            .Clk(Clk)                 ,
73            .Rst_n(Rst_n)             ,
74            .Tick(tick)               ,
75            .BaudRate(BaudRate)
76          );
77
78
79
80      endmodule
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
```

UART_rx.v

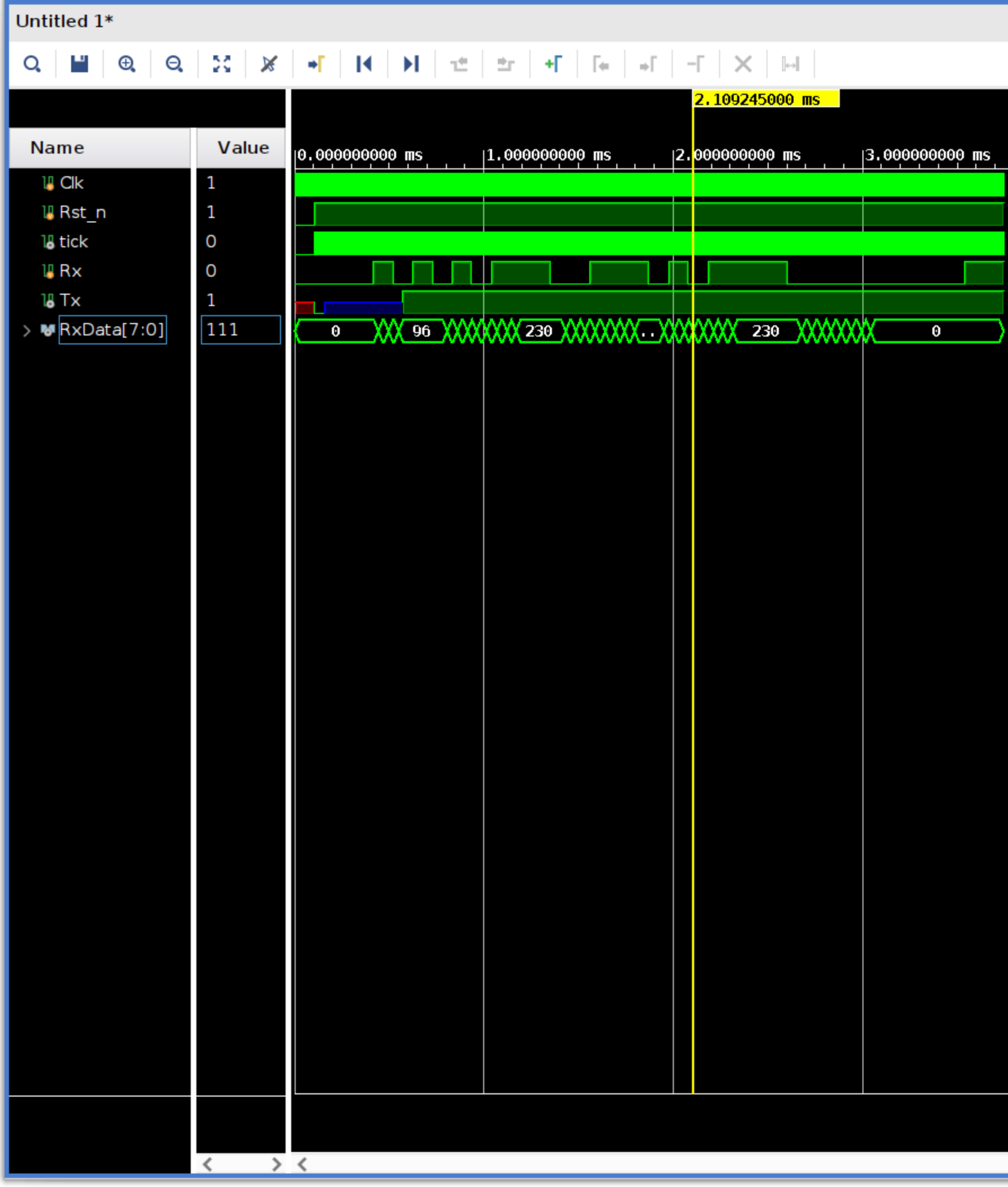
/home/itzzinfinity/Cozy Drive/100daysofRTL/UART_trial/UART_trial.srcs/sources_1/new/UART_rx.v

```
1  module UART_rs232_rx (Clk,Rst_n,RxEn,RxData,RxDone,Rx,Tick,NBits);    //Define my modul
2
3  input Clk, Rst_n, RxEn,Rx,Tick;
4  input [3:0]NBits;
5  output RxDone;
6  output [7:0]RxData;
7
8  parameter  IDLE = 1'b0, READ = 1'b1;
9  reg [1:0] State, Next;
10 reg  read_enable = 1'b0;
11 reg  start_bit = 1'b1;
12 reg  RxDone = 1'b0;
13 reg [4:0]Bit = 5'b00000;
14 reg [3:0] counter = 4'b0000;
15 reg [7:0] Read_data= 8'b00000000;
16 reg [7:0] RxData;
17
18 always @ (posedge Clk or negedge Rst_n)
19 begin
20 if (!Rst_n) State <= IDLE;
21 else    State <= Next;
22 end
23 always @ (State or Rx or RxEn or RxDone)
24 begin
25     case(State)
26   IDLE: if(!Rx & RxEn)    Next = READ;
27     else      Next = IDLE;
28   READ: if(RxDone)    Next = IDLE;
29     else      Next = READ;
30   default       Next = IDLE;
31     endcase
32 end
33
34 always @ (State or RxDone)
35 begin
36     case (State)
37   READ: begin
38     read_enable <= 1'b1;
39         end
40
41   IDLE: begin
42     read_enable <= 1'b0;
43         end
44     endcase
45 end
46
47 always @ (posedge Tick)
48
49   begin
50     if (read_enable)
```

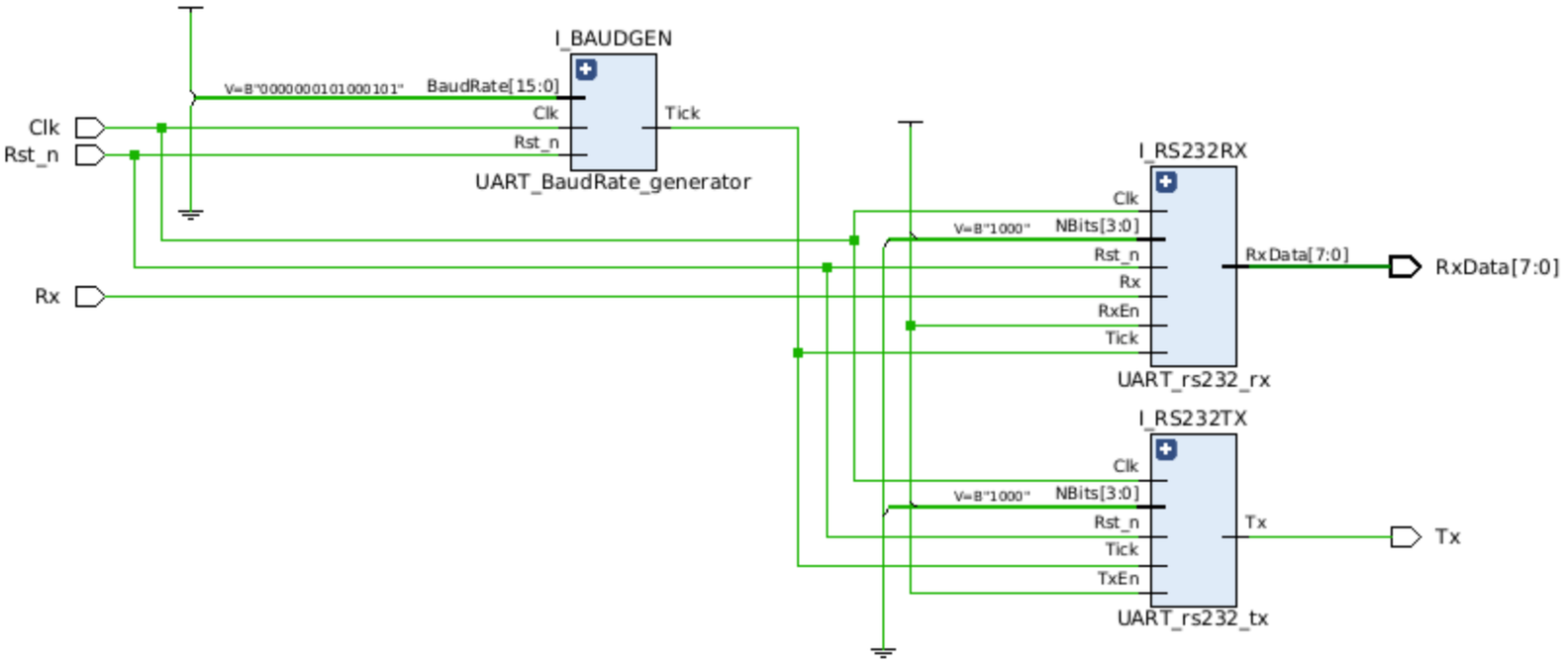
UART_rx.v

/home/itzzinfinity/Cozy Drive/100daysofRTL/UART_trial/UART_trial.srcs/sources_1/new/UART_rx.v

```
51      begin
52        RxDone <= 1'b0;
53        counter <= counter+1;
54
55
56        if ((counter == 4'b1000) & (start_bit))
57        begin
58          start_bit <= 1'b0;
59          counter <= 4'b0000;
60        end
61
62        if ((counter == 4'b1111) & (!start_bit) & (Bit < NBits))
63        begin
64          Bit <= Bit+1;
65          Read_data <= {Rx,Read_data[7:1]};
66          counter <= 4'b0000;
67        end
68
69        if ((counter == 4'b1111) & (Bit == NBits)  & (Rx))
70        begin
71          Bit <= 4'b0000;
72          RxDone <= 1'b1;
73          counter <= 4'b0000;
74          start_bit <= 1'b1;
75        end
76      end
77
78  end
79
80    always @ (posedge Clk)
81      begin
82
83        if (NBits == 4'b1000)
84        begin
85          RxData[7:0] <= Read_data[7:0];
86        end
87
88        if (NBits == 4'b0111)
89        begin
90          RxData[7:0] <= {1'b0,Read_data[7:1]};
91        end
92
93        if (NBits == 4'b0110)
94        begin
95          RxData[7:0] <= {1'b0,1'b0,Read_data[7:2]};
96        end
97      end
98  endmodule
99
100
```

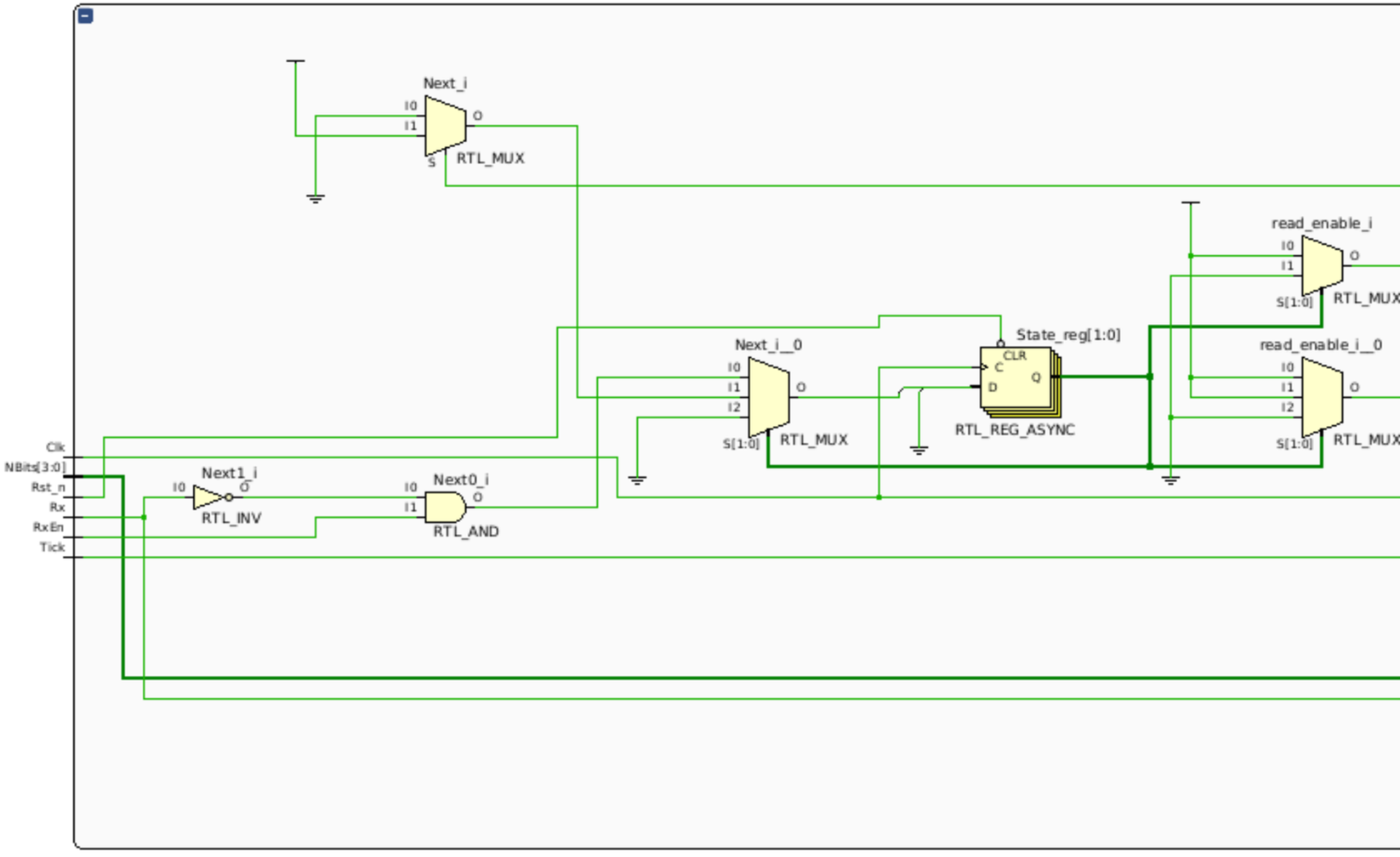

Untitled 1*
Name
Value
Clk
1
Rst_n
1
tick
0
Rx
0
Tx
1
RxData[7:0]
111
2.109245000 ms
0.000000000 ms
1.000000000 ms
2.000000000 ms
3.000000000 ms
0
96
230
..
230
0

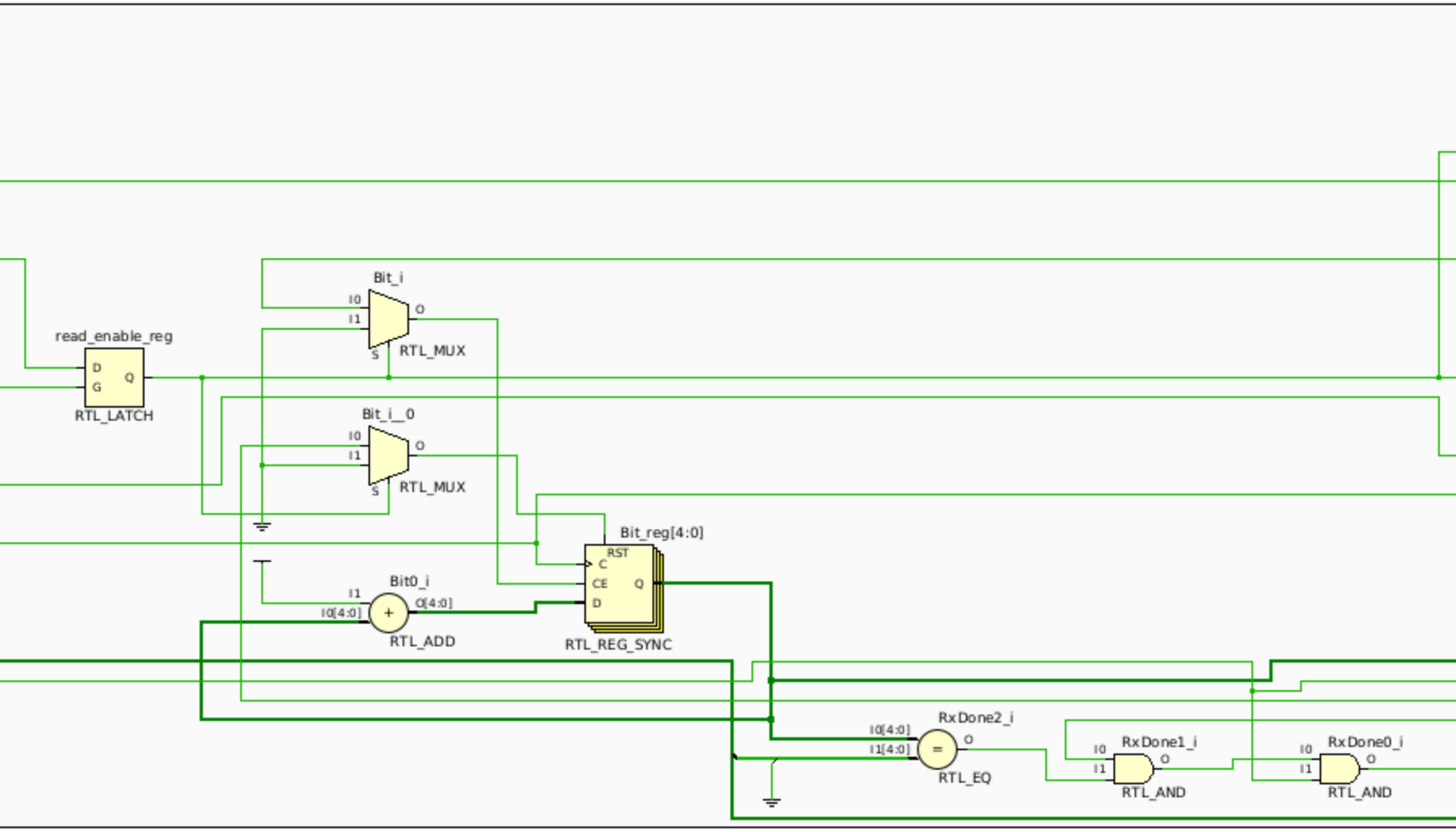
3 Cells 12 I/O Ports 15 Nets



Schematic (2)

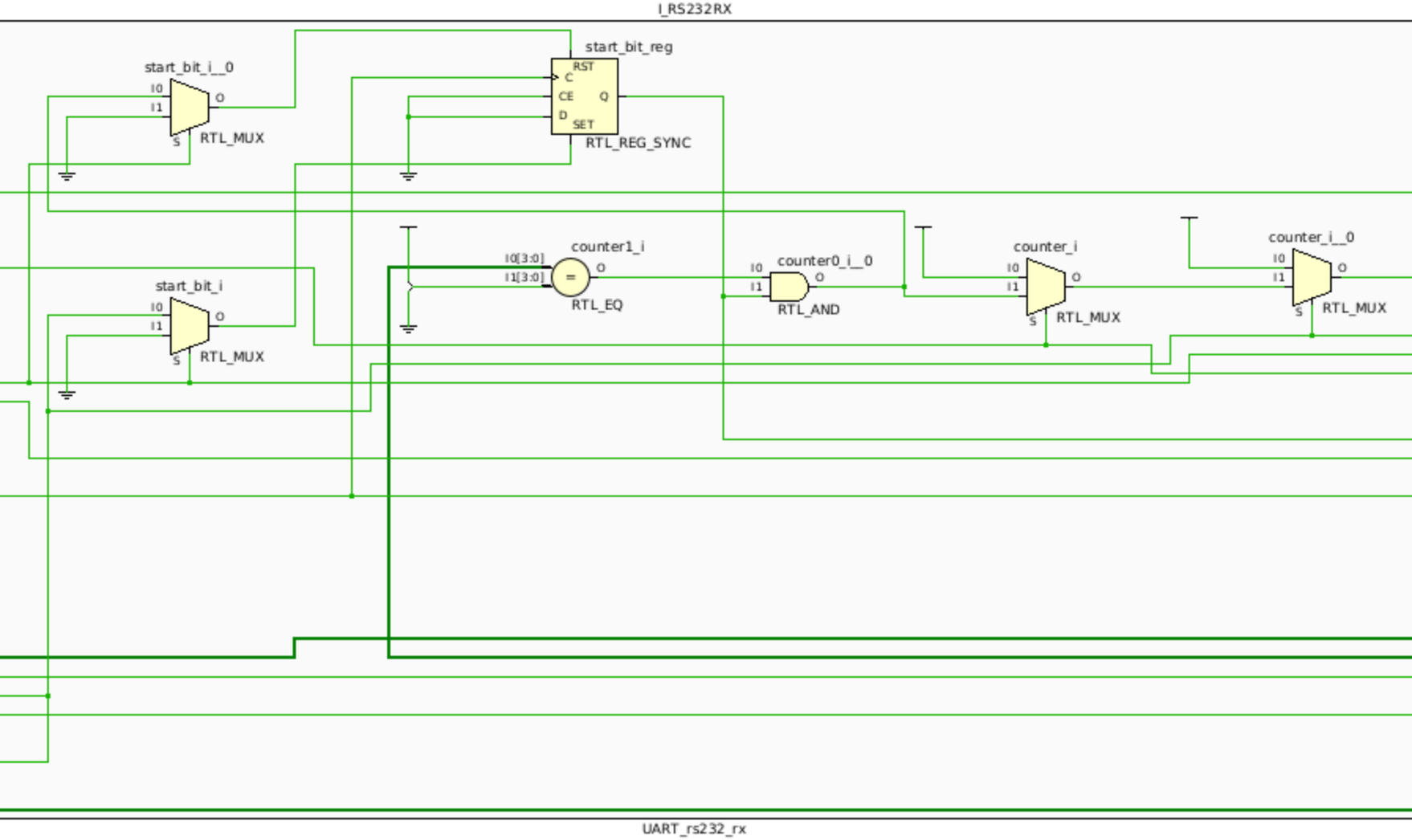
62 Cells 93 Nets

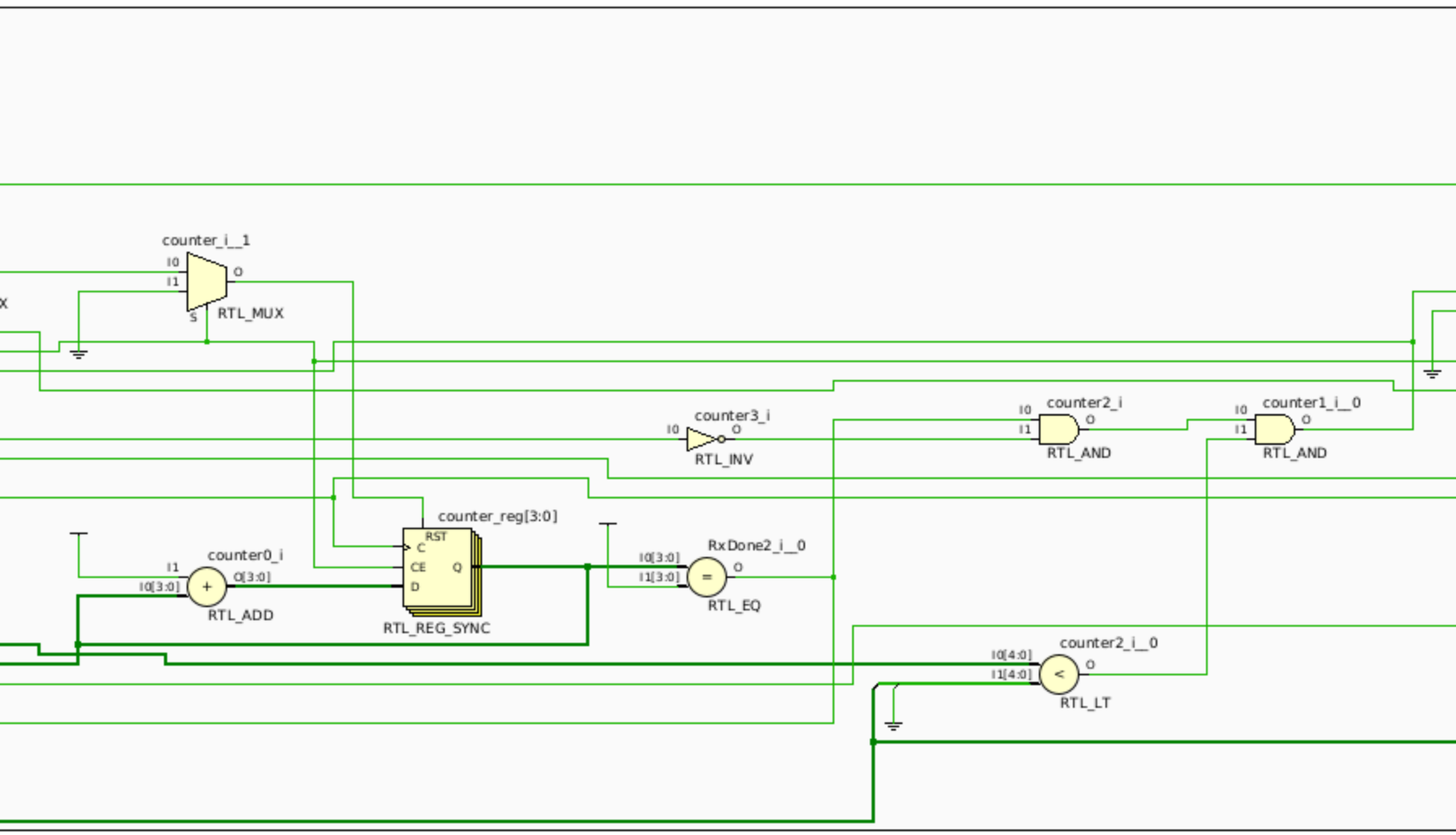


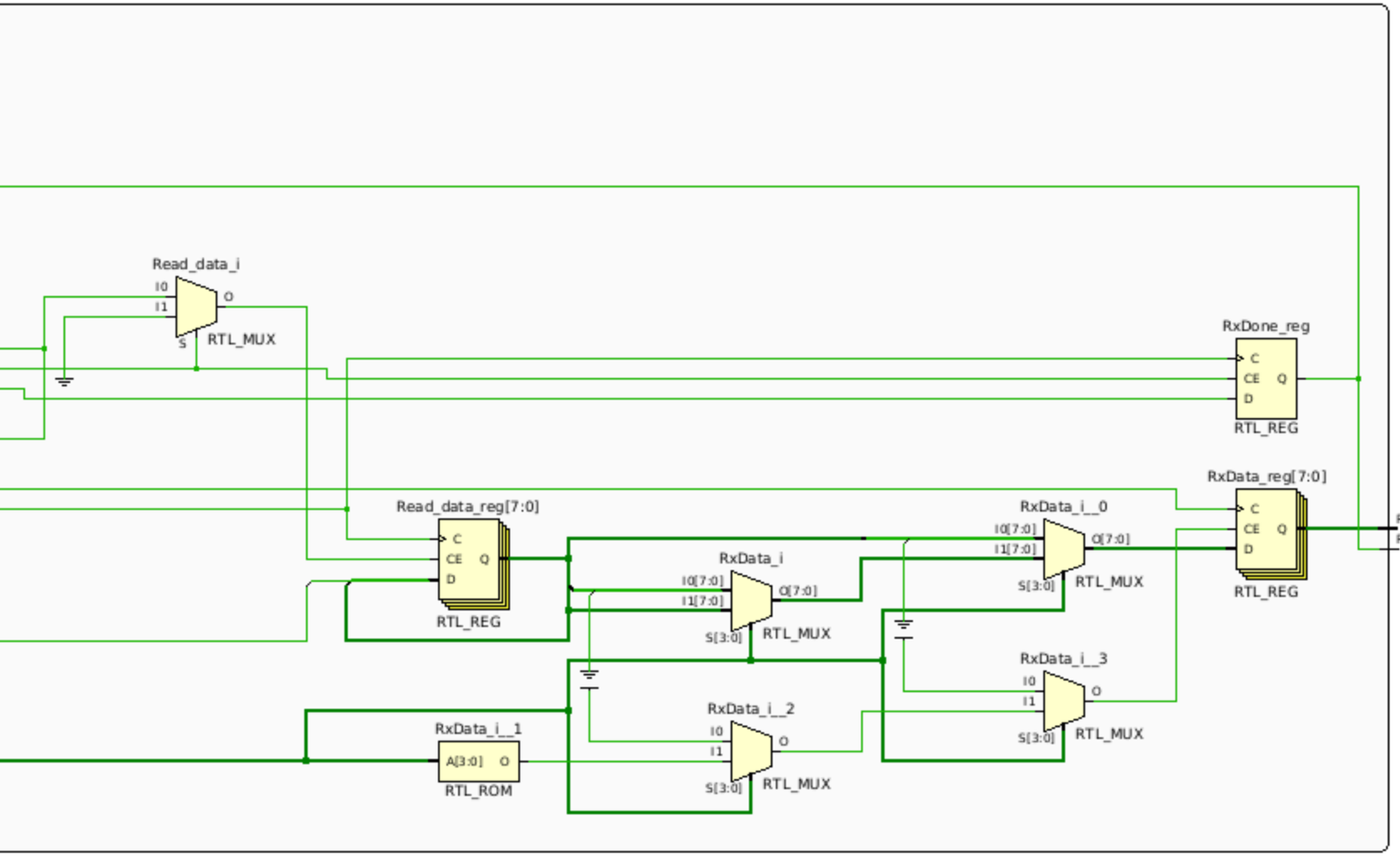

read_enable_reg
D
G
Q
RTL_LATCH
Bit_i
I0
I1
O
S
RTL_MUX
Bit_i__0
I0
I1
O
S
RTL_MUX
Bit_reg[4:0]
RST
C
CE
D
Q
RTL_REG_SYNC
Bit0_i
I1
I0[4:0]
+
O[4:0]
RTL_ADD
RxDone2_i
I0[4:0]
I1[4:0]
=
O
RTL_EQ
RxDone1_i
I0
I1
O
RTL_AND
RxDone0_i
I0
I1
O
RTL_AND

Schematic (2)

62 Cells 93 Nets




counter_i__1
I0
I1
O
S
RTL_MUX
counter3_i
I0
O
RTL_INV
counter2_i
I0
I1
O
RTL_AND
counter1_i__0
I0
I1
O
RTL_AND
counter_reg[3:0]
RST
C
CE
D
Q
RTL_REG_SYNC
counter0_i
I1
I0[3:0]
O[3:0]
+
RTL_ADD
RxDone2_i__0
I0[3:0]
I1[3:0]
=
O
RTL_EQ
counter2_i__0
I0[4:0]
I1[4:0]
<
O
RTL_LT


Read_data_i
I0
I1
O
S
RTL_MUX
RxDone_reg
C
CE
Q
D
RTL_REG
RxData_reg[7:0]
RxData[7:0]
RxDone
Read_data_reg[7:0]
RTL_REG
RxData_i
I0[7:0]
I1[7:0]
O[7:0]
S[3:0]
RTL_MUX
RxData_i__0
RxData_i__3
RxData_i__2
RxData_i__1
A[3:0]
RTL_ROM

tb.v

/home/itzzinfinity/Cozy Drive/100daysofRTL/UART_trial/UART_trial.srcs/sim_1/new/tb.v

```
1   `timescale 1ns / 1ns
2   //////////////////////////////////////////////////////////////////////////////////
3   // Engineer: Anjan Prasad
4   // Create Date: 10/20/2024 09:05:21 PM
5   // Module Name: tb
6   //////////////////////////////////////////////////////////////////////////////////
7
8
9   module tb;
10  reg Clk =0  ;
11  reg Rst_n;
12  reg Rx ;
13  wire Tx;
14  wire [7:0] RxData;
15  UART_master DUT(Clk,Rst_n,Rx,Tx,RxData);
16  always #5 Clk=~Clk;
17  initial begin
18  Rst_n =0;Rx=0;
19
20
21
22
23  /* trying 1st Number */
24  #104000 Rst_n =1;Rx=0;
25  #104000 Rst_n =1;Rx=0;
26  #104000 Rst_n =1;Rx=0;
27  #104000 Rst_n =1;Rx=1;
28  #104000 Rst_n =1;Rx=0;
29  #104000 Rst_n =1;Rx=1;
30  #104000 Rst_n =1;Rx=0;
31  #104000 Rst_n =1;Rx=1;
32  #104000 Rst_n =1;Rx=0;
33  #104000 Rst_n =1;Rx=1;
34  #104000 Rst_n =1;Rx=1;
35  #104000 Rst_n =1;Rx=1;
36
37
38
39  /* trying 2nd Number */
40  #104000 Rst_n =1;Rx=0;
41  #104000 Rst_n =1;Rx=0;
42  #104000 Rst_n =1;Rx=1;
43  #104000 Rst_n =1;Rx=1;
44  #104000 Rst_n =1;Rx=1;
45  #104000 Rst_n =1;Rx=0;
46  #104000 Rst_n =1;Rx=1;
47  #104000 Rst_n =1;Rx=0;
48  #104000 Rst_n =1;Rx=1;
49  #104000 Rst_n =1;Rx=1;
50  #104000 Rst_n =1;Rx=1;
```

tb.v

/home/itzzinfinity/Cozy Drive/100daysofRTL/UART_trial/UART_trial.srcs/sim_1/new/tb.v

```
50      #104000 Rst_n =1;Rx=1;
51      #104000 Rst_n =1;Rx=1;
52
53
54
55
56      /* trying 3rd Number */
57      #104000 Rst_n =1;Rx=0;
58      #104000 Rst_n =1;Rx=0;
59      #104000 Rst_n =1;Rx=0;
60      #104000 Rst_n =1;Rx=0;
61      #104000 Rst_n =1;Rx=0;
62      #104000 Rst_n =1;Rx=0;
63      #104000 Rst_n =1;Rx=0;
64      #104000 Rst_n =1;Rx=0;
65      #104000 Rst_n =1;Rx=0;
66      #104000 Rst_n =1;Rx=1;
67      #104000 Rst_n =1;Rx=1;
68      #104000 Rst_n =1;Rx=1;
69
70
71      #100 $finish;
72
73
74      end
75
76      endmodule
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
```