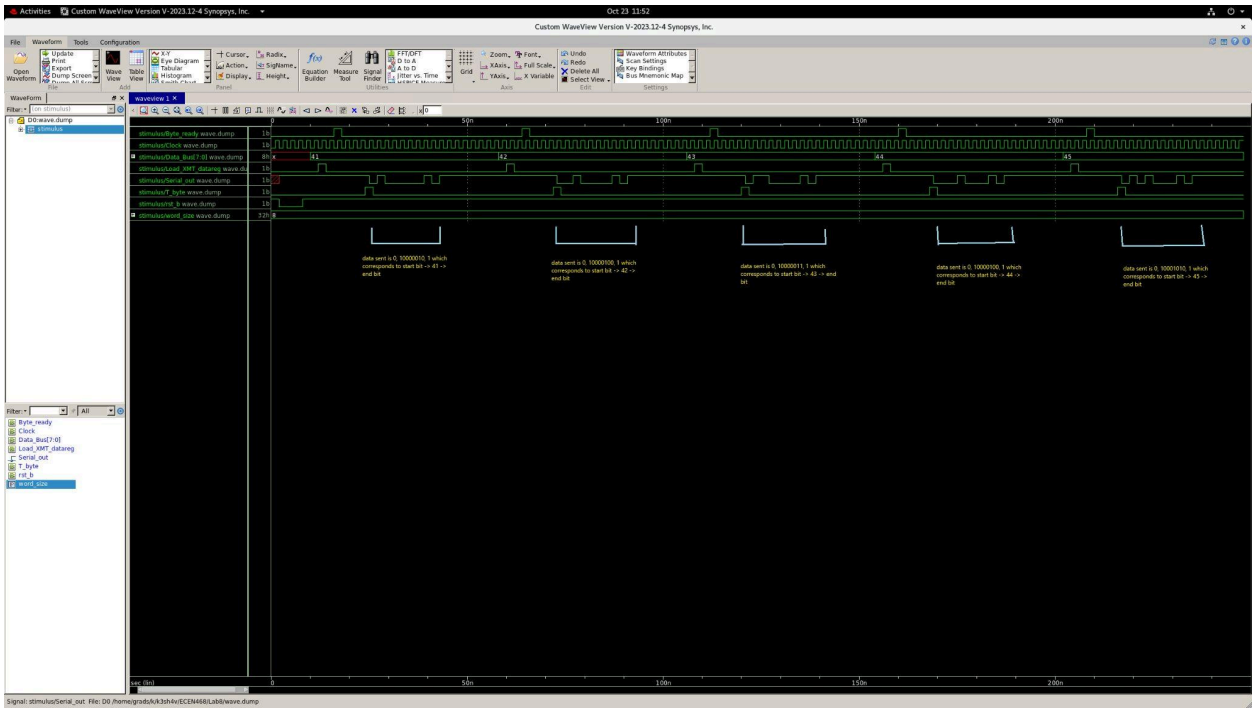
a. Text result of the functional simulation (3 points)

Attached in the email as the output is too large to add in the report. The attached file name is FunctionalSimulation_out.txt

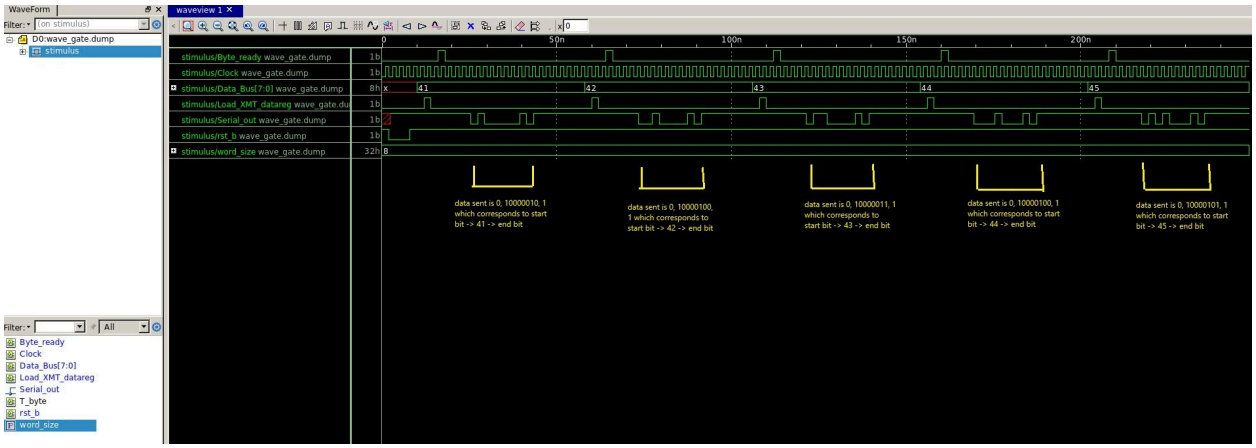
b. Waveform result(screen captured) of the function simulation with analysis. (5 points)



c. Text result of the gate simulation (3 points)

Attached in the email as the output is too large to add in the report. The attached file name is GateSimulation_out.txt

d. Waveform result(screen captured) of the gate simulation with analysis. (5 points)



e. UART_XMTR.v with detailed comments (19 points)

Attached in the email along with other files.

Additionally, as Synthesis do not support UDP blocks, UART_XMTR_UDP.v contains the UDP block for the signal BC_lt_BCmax.

UDP definition screenshot:

```
124     // Connect your UDP (User Defined Primitive)
125     // Insert your code here.
126     cmp cmp1(BC_lt_BCmax, bit_count[3], bit_count[2], bit_count[1], bit_count[0]);
127
128
129     // Data Path for UART Transmitter
130     always @(posedge Clock or negedge rst_b)
131     begin
132         // ------------------------------------
133         // Insert your code here.
134         // ------------------------------------
135         if(!rst_b) begin
136             XMT_datareg <= 8'b0;
137             XMT_shftreg <= 9'h1ff;
138             bit_count   <= 4'b0;
139         end
140         else begin
141             if(Load_XMT_DR) XMT_datareg      <= Data_Bus;
142             XMT_shftreg <= XMT_shftreg_d;
143             bit_count  <= bit_count_d;
144         end
145     end
146 endmodule
147
148 // UDP (User Defined Primitive)
149 // ------------------------------------
150 // Insert your UDP here.
151 // ------------------------------------
152 primitive cmp(out, a3, a2, a1, a0);
153     output out;
154     input a3, a2, a1, a0; // Define individual bits of a
155
156     table
157         // a3 a2 a1 a0 : out
158         0 0 0 0 : 1; // 0
159         0 0 0 1 : 1; // 1
160         0 0 1 0 : 1; // 2
161         0 0 1 1 : 1; // 3
162         0 1 0 0 : 1; // 4
163         0 1 0 1 : 1; // 5
164         0 1 1 0 : 1; // 6
165         0 1 1 1 : 1; // 7
166         1 0 0 0 : 1; // 8
167         1 0 0 1 : 0; // 9
168         1 0 1 0 : 0; // 10
169         1 0 1 1 : 0; // 11
170         1 1 0 0 : 0; // 12
171         1 1 0 1 : 0; // 13
172         1 1 1 0 : 0; // 14
173         1 1 1 1 : 0; // 15
174
175     endtable
176 endprimitive
```