Experiment 2

CSE460: VLSI Design Lab

Submitted To

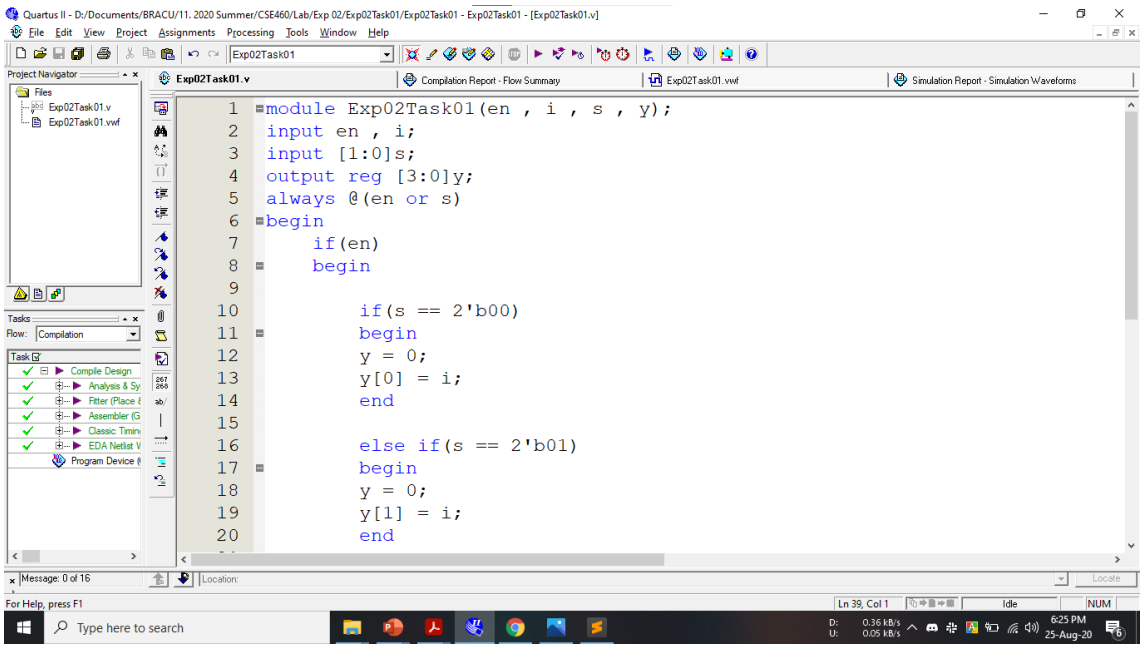
Mehedi Hasan Himel
Lecturer
Department of Computer Science and Engineering
BRAC University

Submitted By

Mehadi Hassan
ID: 17101177
Section: 13
Summer 2020

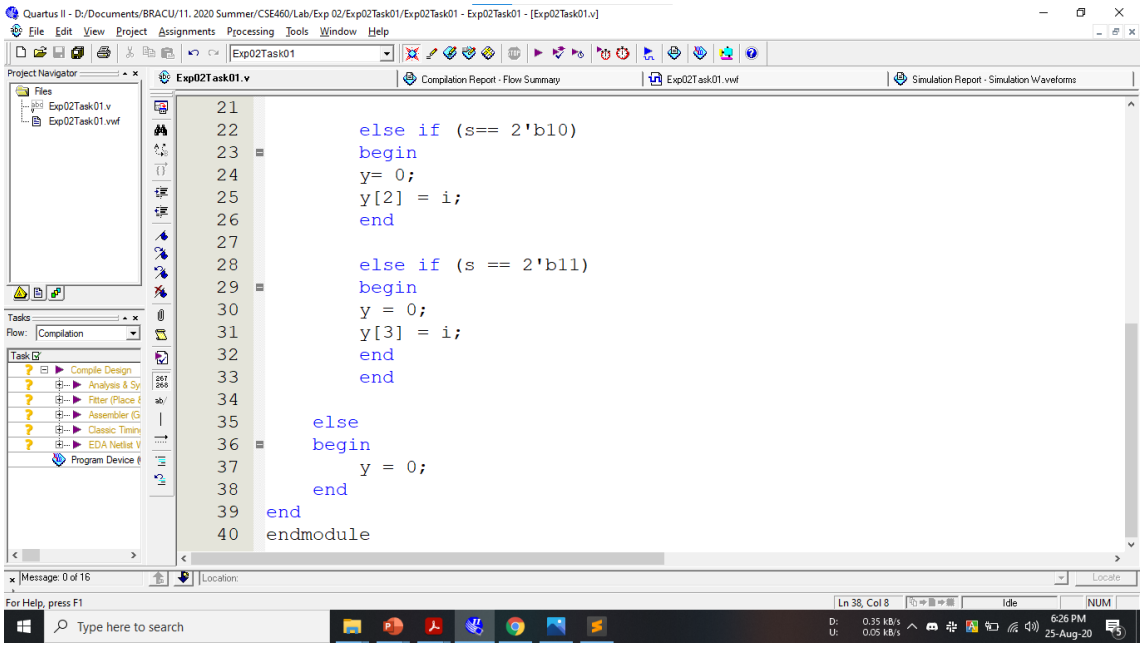
1. 1 to 4 Demultiplexer

Ans:



```
1  module Exp02Task01(en , i , s , y);
2  input en , i;
3  input [1:0]s;
4  output reg [3:0]y;
5  always @(en or s)
6  begin
7      if(en)
8      begin
9
10         if(s == 2'b00)
11         begin
12         y = 0;
13         y[0] = i;
14         end
15
16         else if(s == 2'b01)
17         begin
18         y = 0;
19         y[1] = i;
20         end
```

Figure 1: 1 to 4 Demultiplexer using Verilog HDL [Part 01]



```
21
22         else if (s== 2'b10)
23         begin
24         y= 0;
25         y[2] = i;
26         end
27
28         else if (s == 2'b11)
29         begin
30         y = 0;
31         y[3] = i;
32         end
33         end
34
35     else
36     begin
37         y = 0;
38     end
39 end
40 endmodule
```

Figure 2: 1 to 4 Demultiplexer using Verilog HDL [Part 02]

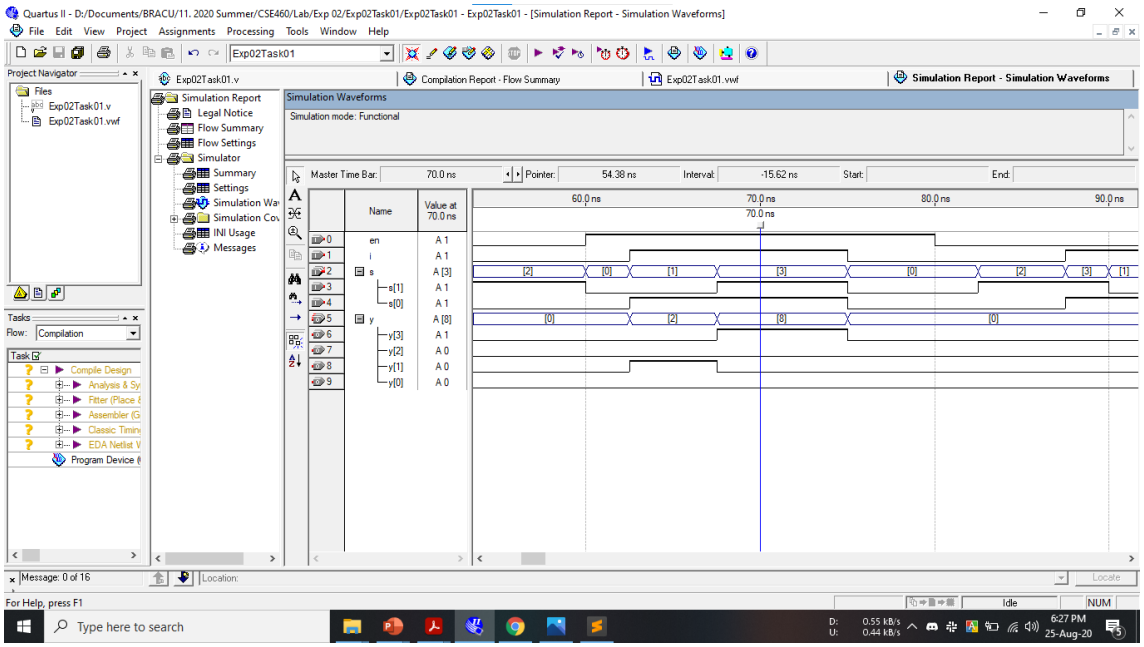


Figure 3: 1 to 4 Demultiplexer Vector Waveform Simulation

Discussion: 1 to 4 Demultiplexer consists has one input, four outputs, and two control lines to make select the output pin. Whichever output pin is selected through the switches the input passes to that line. In the Verilog HDL Code, if the enable is set low the output is set to 0 otherwise we proceed. In the if block we check for the selected switch and pass input to that output line. For reducing error y has been cleared to 0 beforehand.

2. 3 to 8 Decoder

Ans:

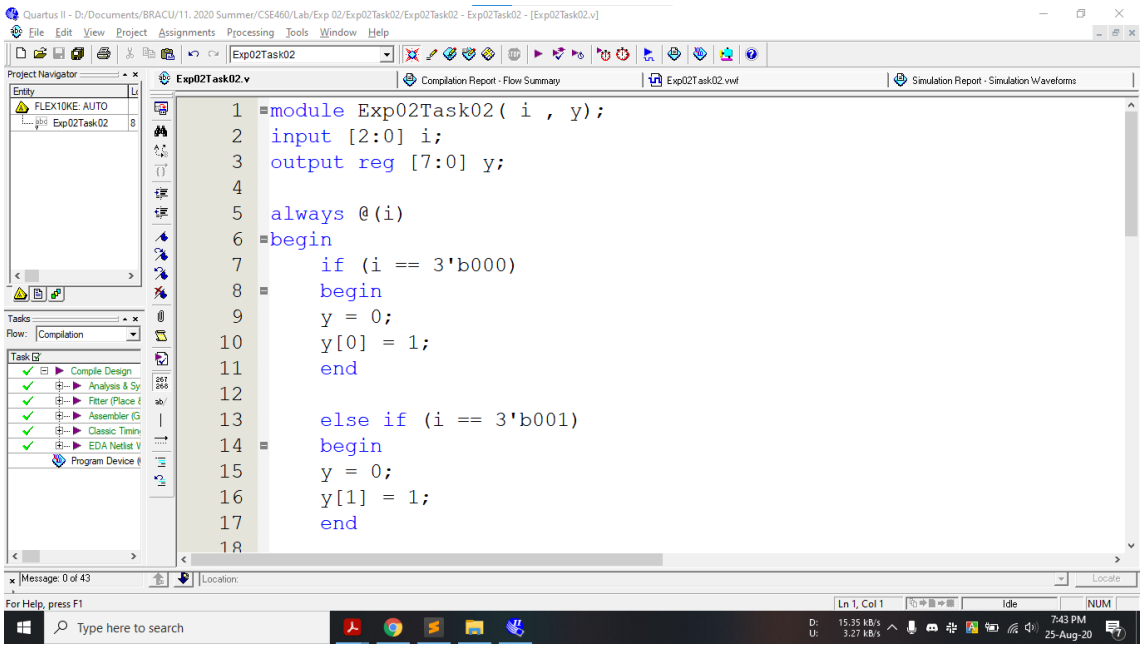


Figure 4: 3 to 8 Decoder using Verilog HDL [Part 01]

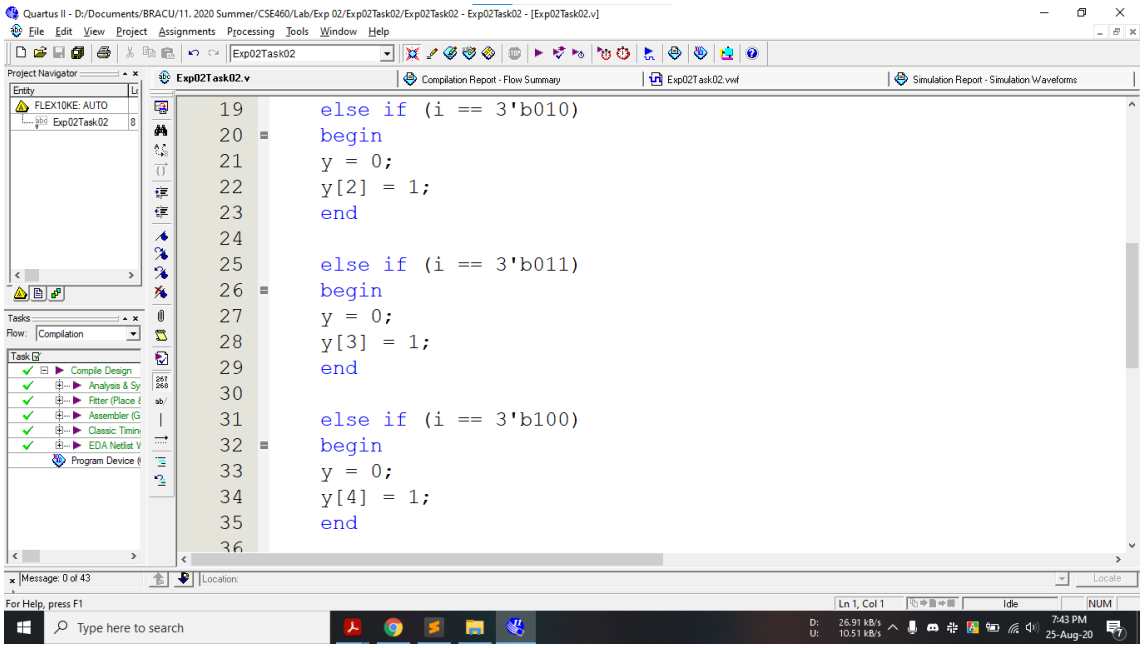


Figure 5: 3 to 8 Decoder using Verilog HDL [Part 02]

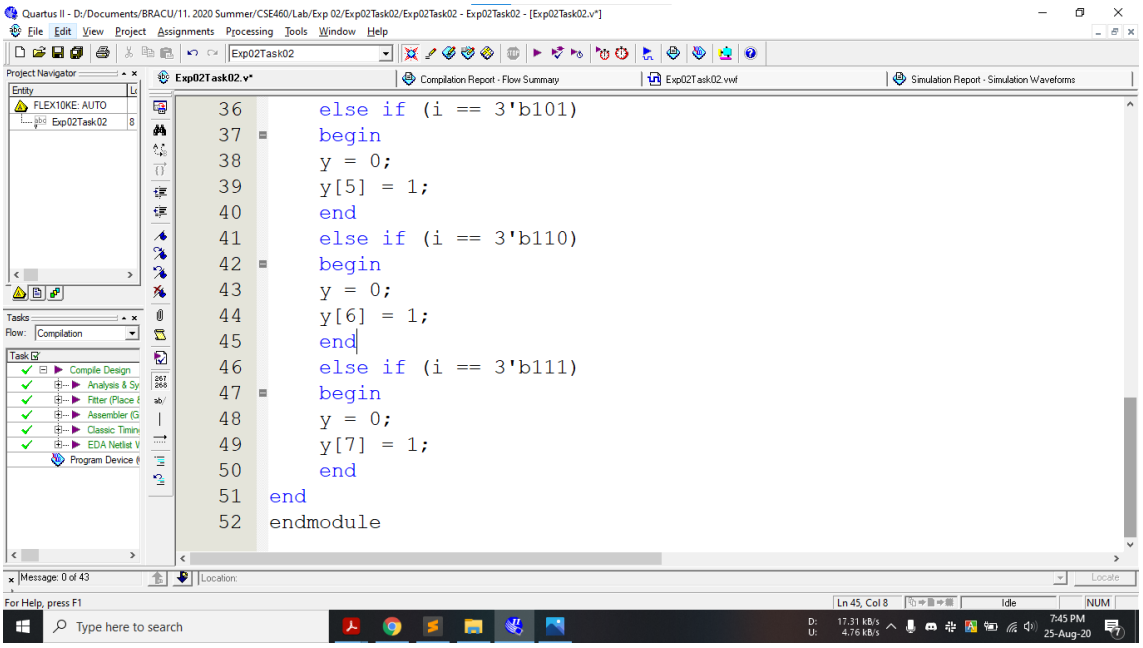


Figure 6: 3 to 8 Decoder using Verilog HDL [Part 03]

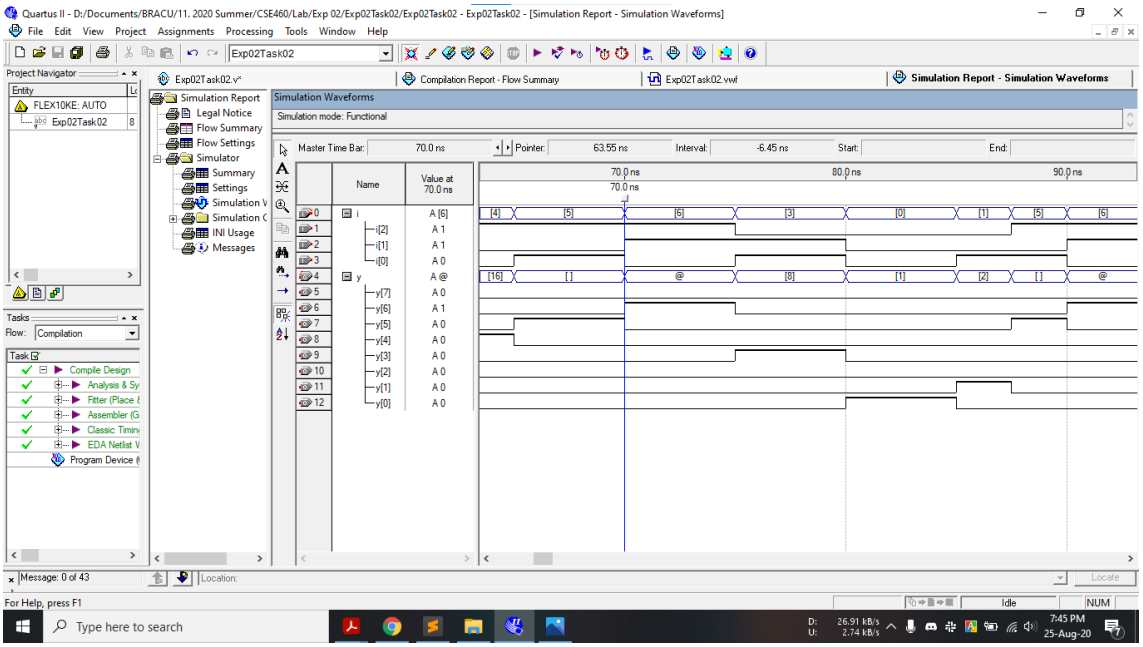
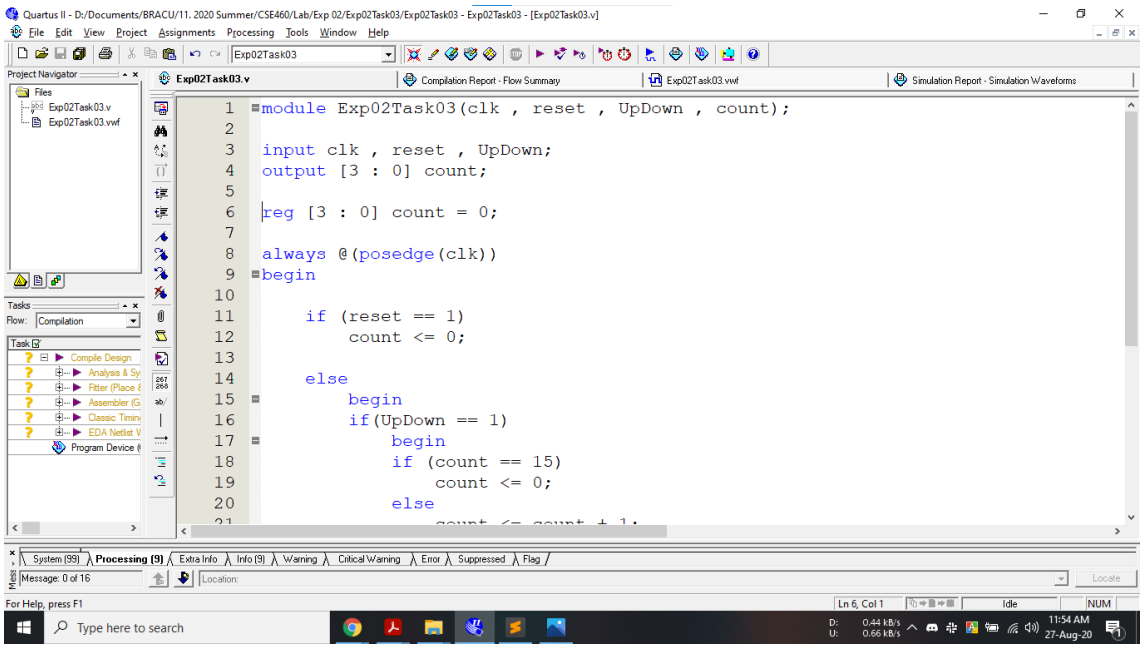


Figure 7: 3 to 8 Decoder Vector Waveform Simulation

Discussion: 3 to 8 Decoder has 3 input pins and 8 output pins. It outputs high on the data line based on the given input binary values. For example if $(111)_b$ is given then the 7 number output channel will be given high as output and rest of them will be low. In the Verilog HDL Code, if. else-if block checks the given input and then outputs 1 to that data line. At first we make the whole output line to zero and then assign 1 as output.

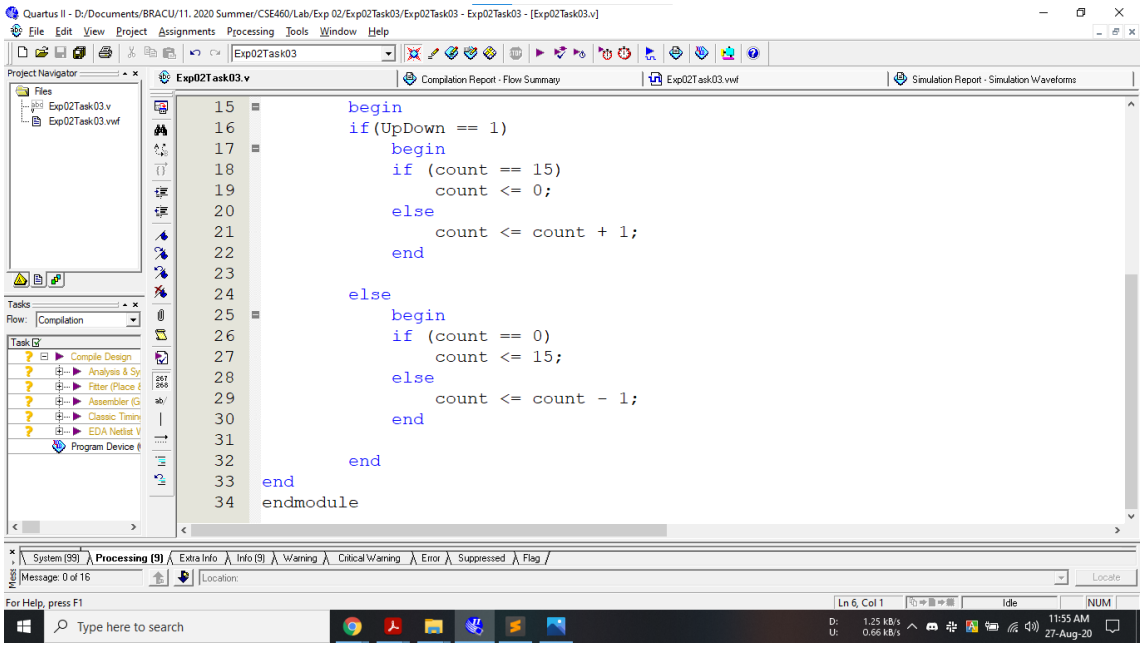
3. Up-Down Counter

Ans:



```
1  module Exp02Task03(clk , reset , UpDown , count);
2
3  input clk , reset , UpDown;
4  output [3 : 0] count;
5
6  reg [3 : 0] count = 0;
7
8  always @(posedge(clk))
9  begin
10
11     if (reset == 1)
12         count <= 0;
13
14     else
15         begin
16         if(UpDown == 1)
17             begin
18             if (count == 15)
19                 count <= 0;
20             else
```

Figure 8: Up-Down Counter using Verilog HDL [Part 01]



```
15         begin
16         if(UpDown == 1)
17             begin
18             if (count == 15)
19                 count <= 0;
20             else
21                 count <= count + 1;
22             end
23
24         else
25             begin
26             if (count == 0)
27                 count <= 15;
28             else
29                 count <= count - 1;
30             end
31
32         end
33 end
34 endmodule
```

Figure 9: Up-Down Counter using Verilog HDL [Part 02]

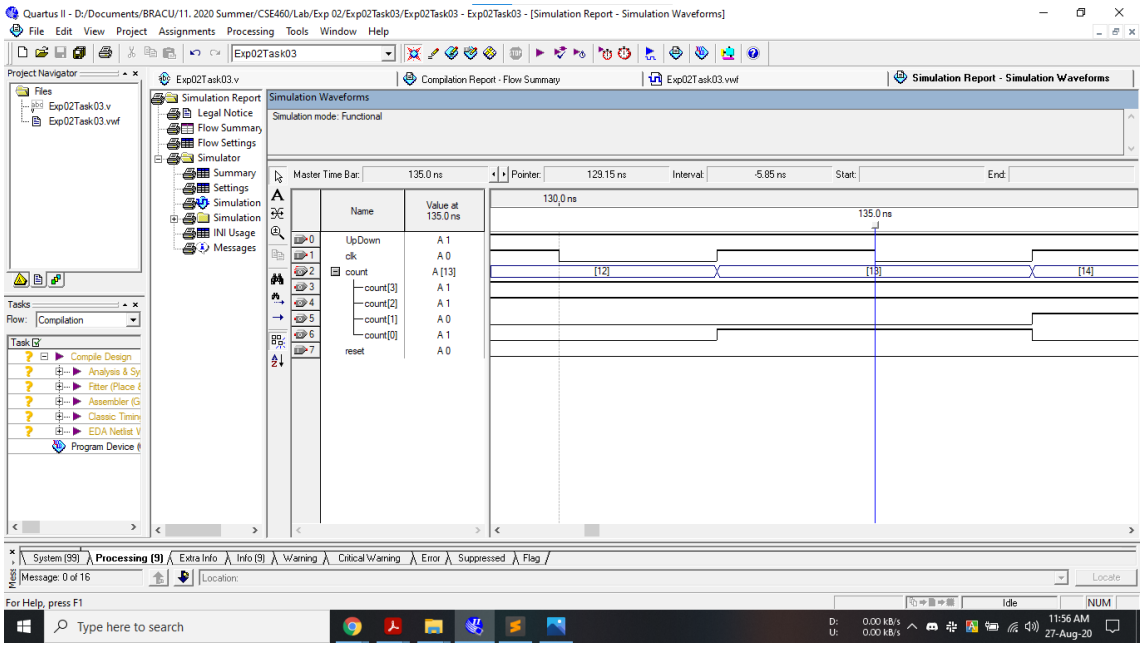


Figure 10: Up-Down Counter Vector Waveform Simulation

Discussion: Up-Down Counter is a bidirectional counter, which can count upwards and downwards at any given time.

In the Verilog HDL Code, first assign clk, reset and UpDown as input and count as output. The code works on positive clock cycle and then if reset is high then the count is set to 0. We have set UpDown 1 to up counter and 0 to Down counter. If the up counter hits 15 then we reset to 0 or the Down counter hits 0 we reset to 15. We increase or decrease the count by one value based on the UpDown value.

4. Ring Counter

Ans:

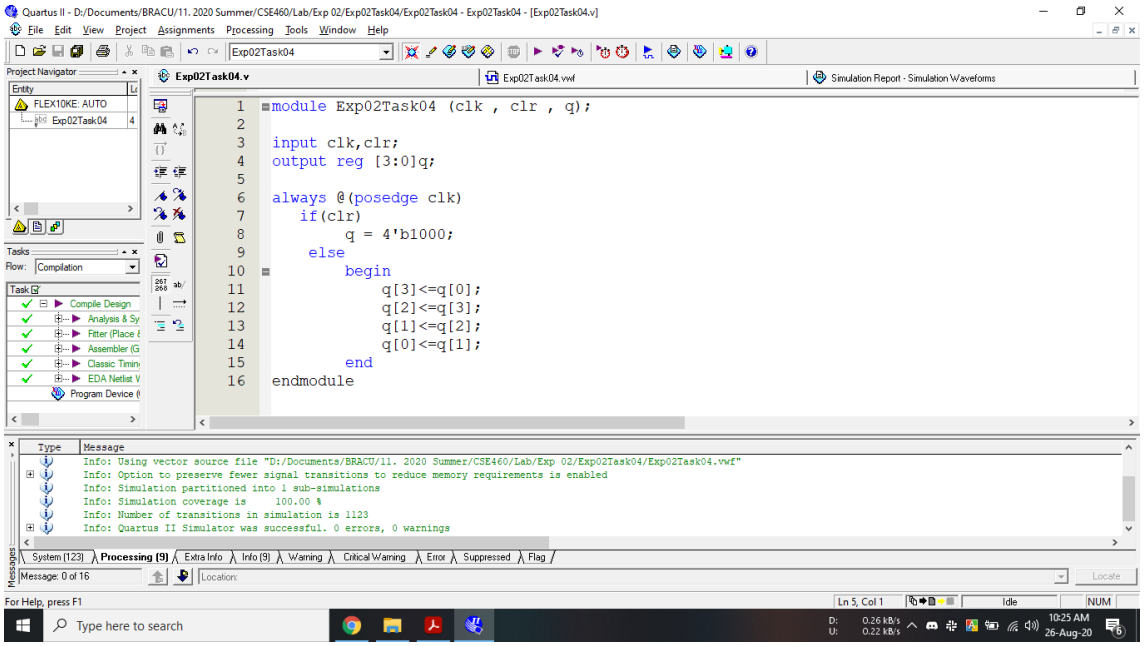


Figure 11: Ring Counter using Verilog HDL

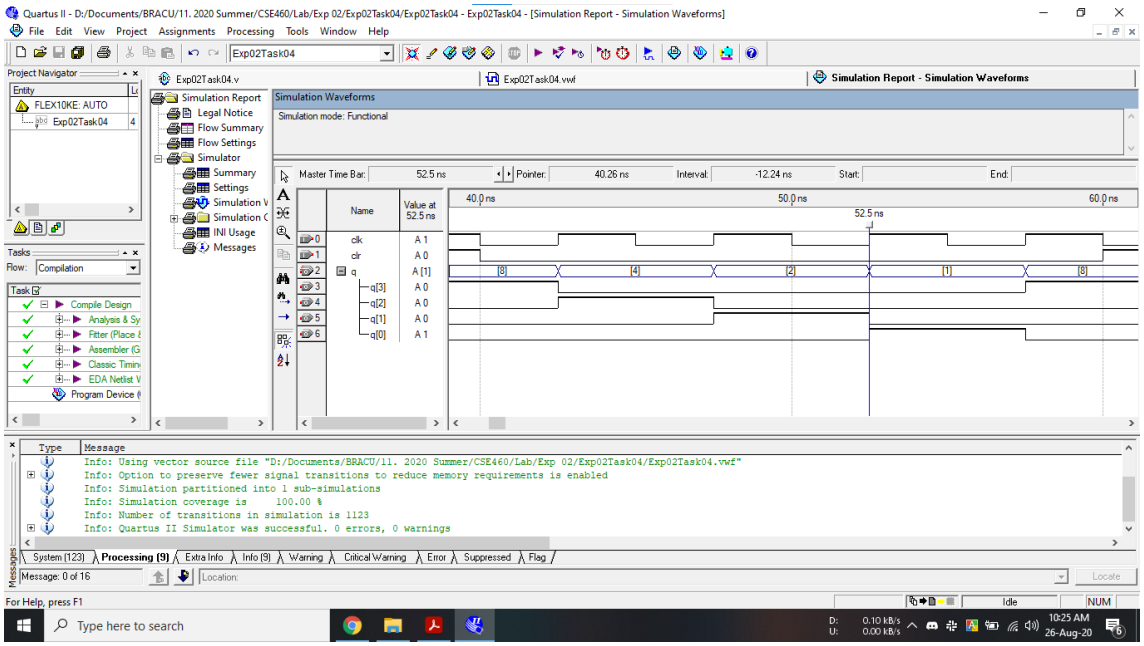


Figure 12: Ring Counter Vector Waveform Simulation

Discussion: Ring counter is a type of counter that shifts the counter by one bit hence it makes a ring shape. If the clr is high then we reset the count to 1000 as it is the initial start. then in every clock by using non blocking notation we update the previous value to the next value which is another word shifting the value and hence makes a ring shape.

5. Johnson Counter

Ans:

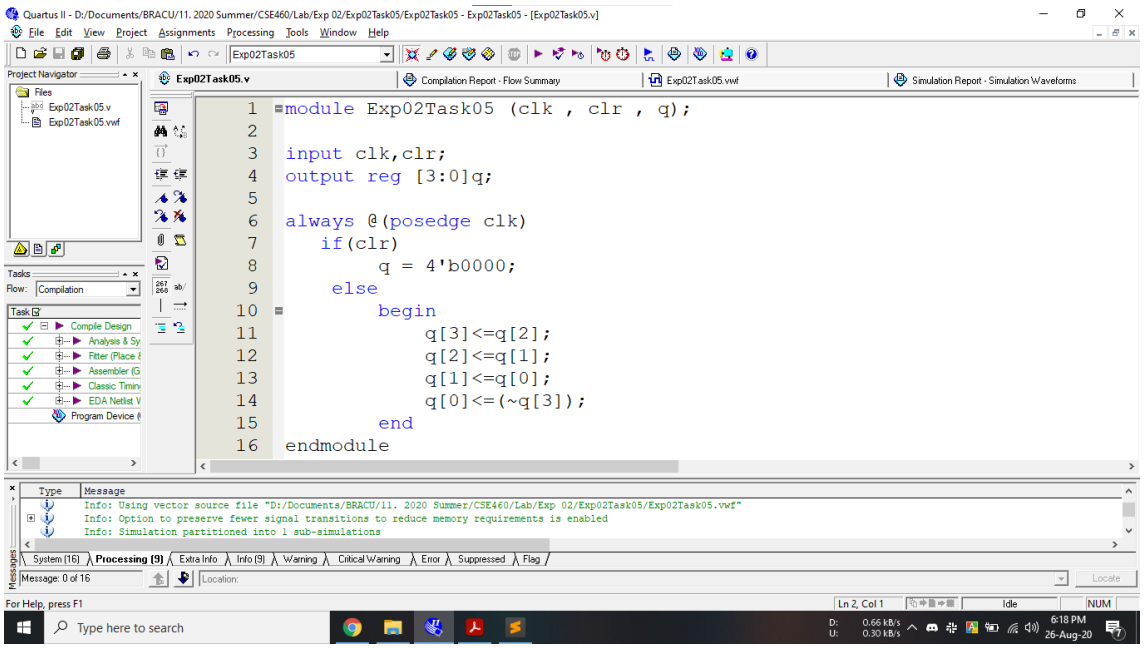


Figure 13: Johnson Counter using Verilog HDL

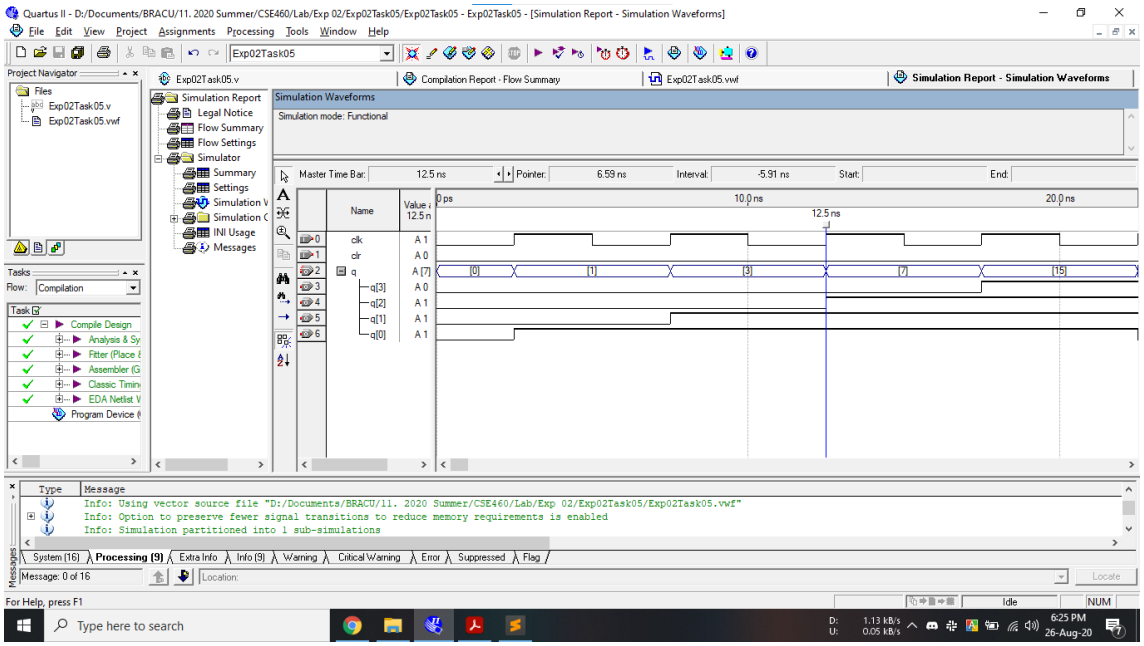


Figure 14: Johnson Counter Vector Waveform Simulation

Discussion: Johnson Counter can count twice the number of Ring counter by having the same number of flip flops. It is used to count the data in a continuous loop.

The counter works on positive clock cycle. If the clear is high then we make the input as 0000. On the other hand, We shift the values but on Q_0 we pass the inverse of Q_3 which makes a continuous loop. For example if the counter starts at 0000 on next iteration it will be 0001 and then 0011 and so on. It will make a continuous loop and this is how a Johnson Counter works