Experiment 1

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. **Design a 4 to 1 MUX using Verilog HDL and verify using timing diagram.**

Ans:

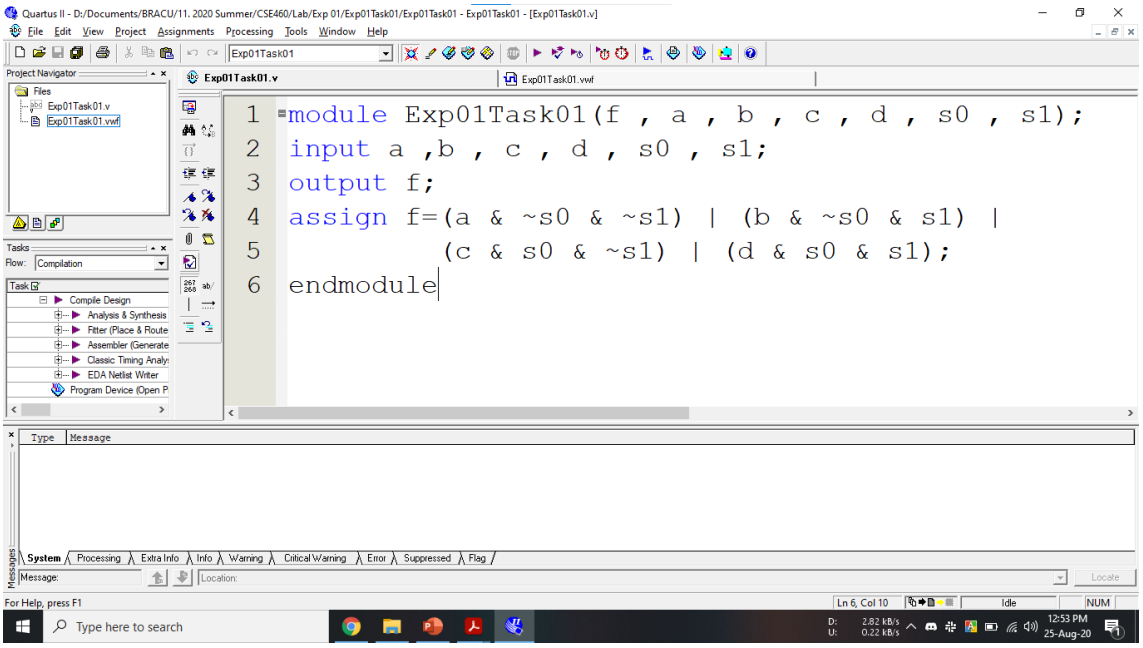

Figure 1: 4 to 1 MUX using Verilog HDL

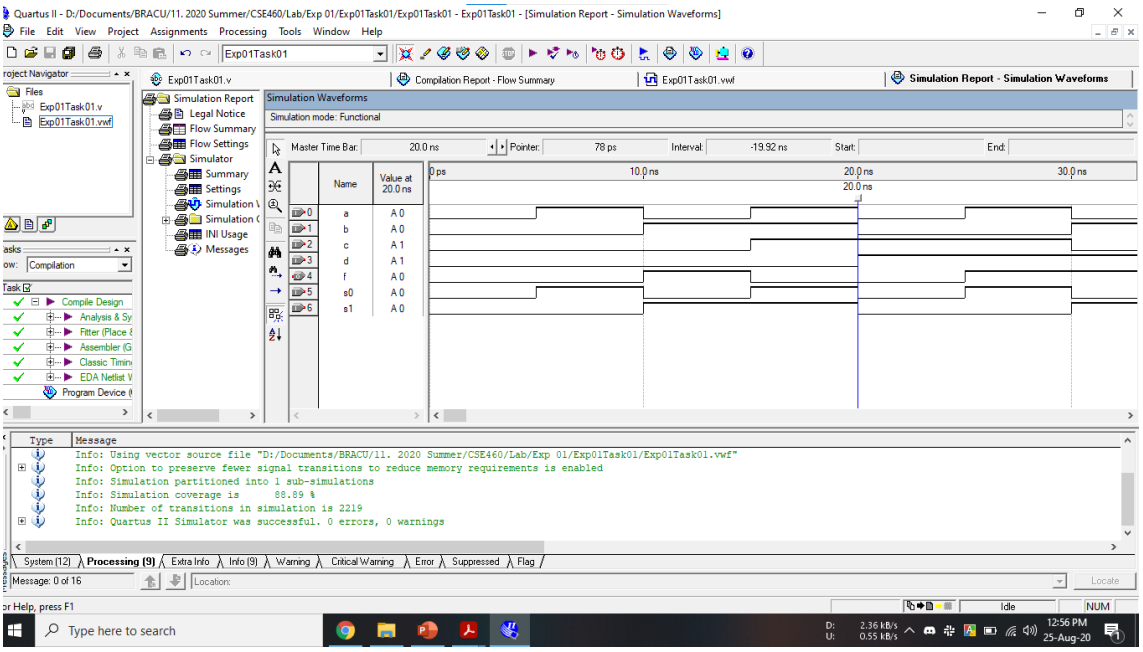

Figure 2: 4 to 1 MUX Vector Waveform Simulation

Discussion: MUX is a digital switch which passes the value of selected pins. 4 to 1 MUX has two input lines, two switches and an output. The particular input combination on select lines selects one of input to the output. By using K-Map and truth table we get 4 to 1 MUX expression as $f = a\bar{s_0}\bar{s_1} + b\bar{s_0}s_1 + cs_0\bar{s_1} + ds_0s_1$. Here, a, b, c, d are data lines and s_0 and s_1 are switches. This expression has been assigned as output f in the Verilog HDL Code.

2. **Design a priority encoder (3>1>0>2) using Verilog HDL and verify using timing diagram.**

Ans:

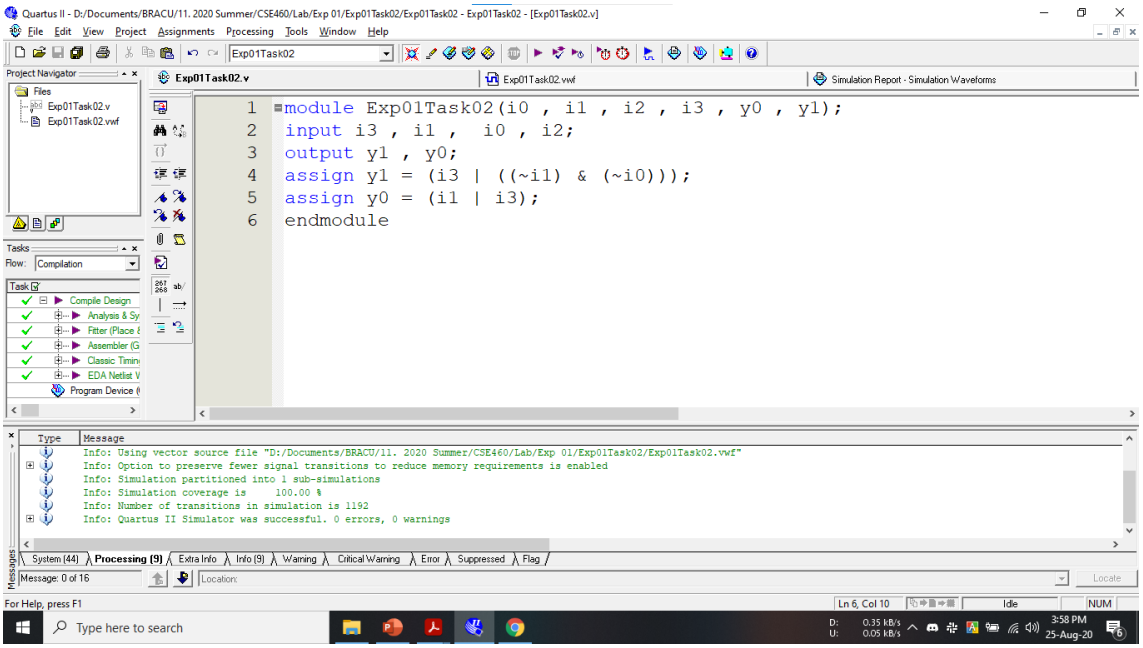

Figure 3: Priority Encoder (3>1>0>2) using Verilog HDL

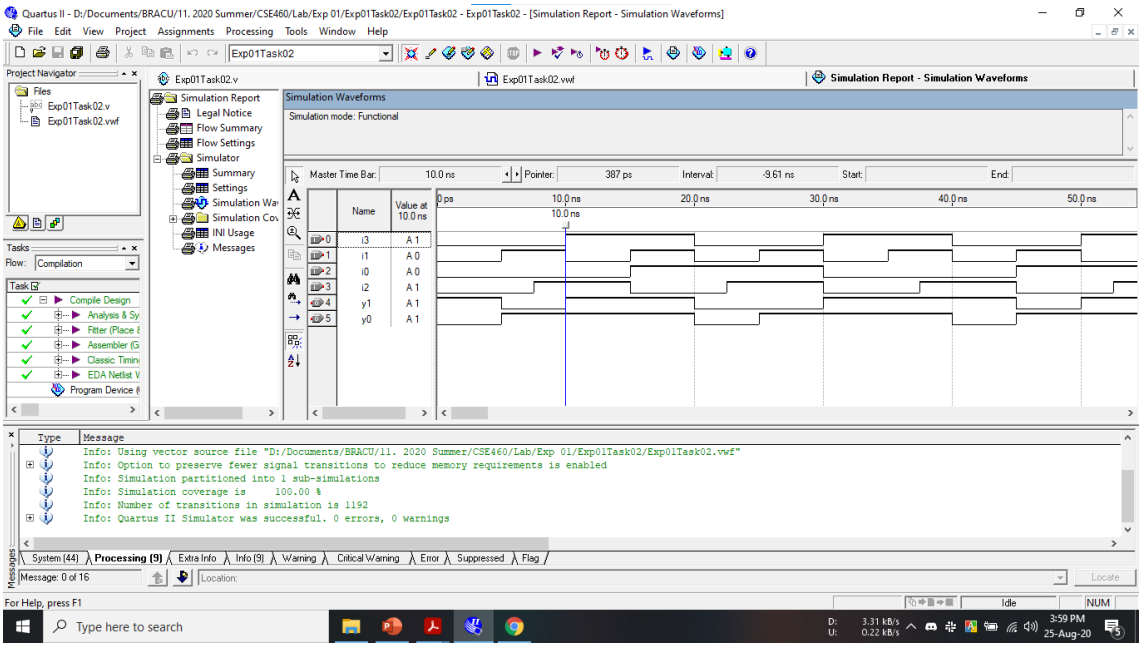

Figure 4: Priority Encoder (3>1>0>2) Vector Waveform Simulation

Discussion: Priority encoder outputs the binary representation of the original number based on priority. If a higher priority input is received than lower priority inputs are omitted. In the task was asked to design a priority encoder which priority hierarchy is $3 > 1 > 0 > 2$. By using the truth table and K-Map the expression generates as $y_1 = i_3 + \bar{i_1}\bar{i_0}$ & $y_0 = i_1 + i_3$. y_1 and y_0 was assigned as output on the Verilog HDL Code

3. **Design a 2 to 4 Decoder using Verilog HDL and verify using timing diagram**

Ans:

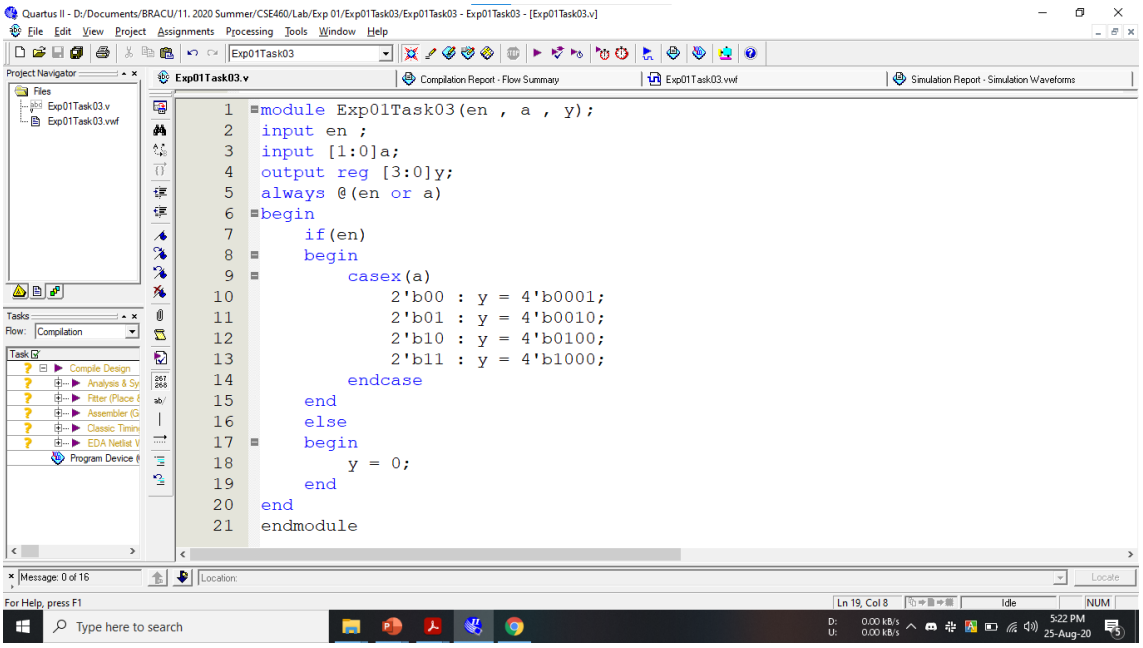

Figure 5: 2 to 4 Decoder using Verilog HDL

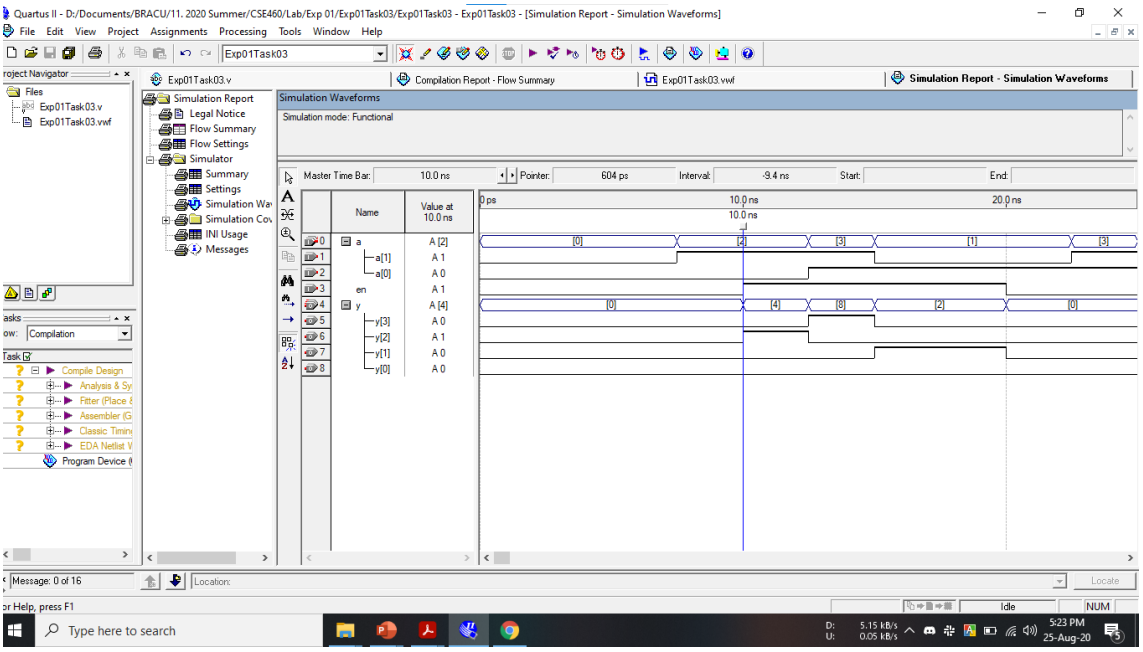

Figure 6: 2 to 4 Decoder Vector Waveform Simulation

Discussion: Decoder one of the outputs will be active High based on the combination of inputs present, when the decoder is enabled. For example, if the input is 11 the output will be 1000. In the Verilog HDL code this combinational logic was used using Case. Here, a is a 2-bit binary output and based on a we output a 4-bit binary value. If the enable is set to 0 then we output 0 hence the decoder will not give any output based on the input.

4. **Design a 4 bit adder-subtractor using Verilog HDL and verify using timing diagram.**

Ans:

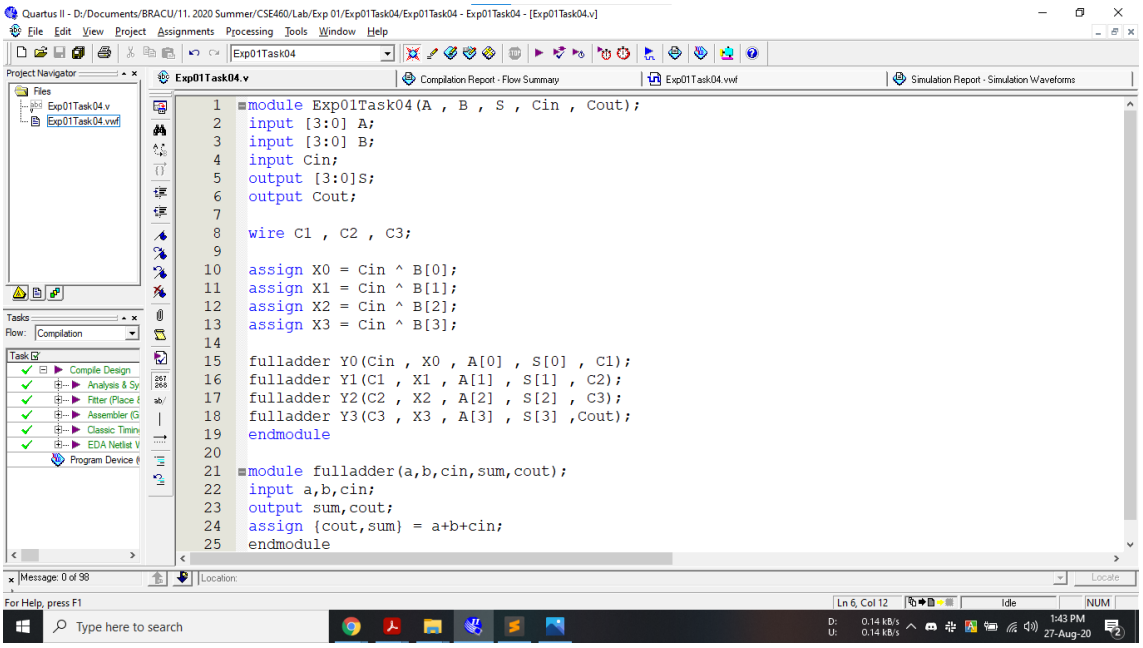

Figure 7: 4 bit adder-subtractor using Verilog HDL

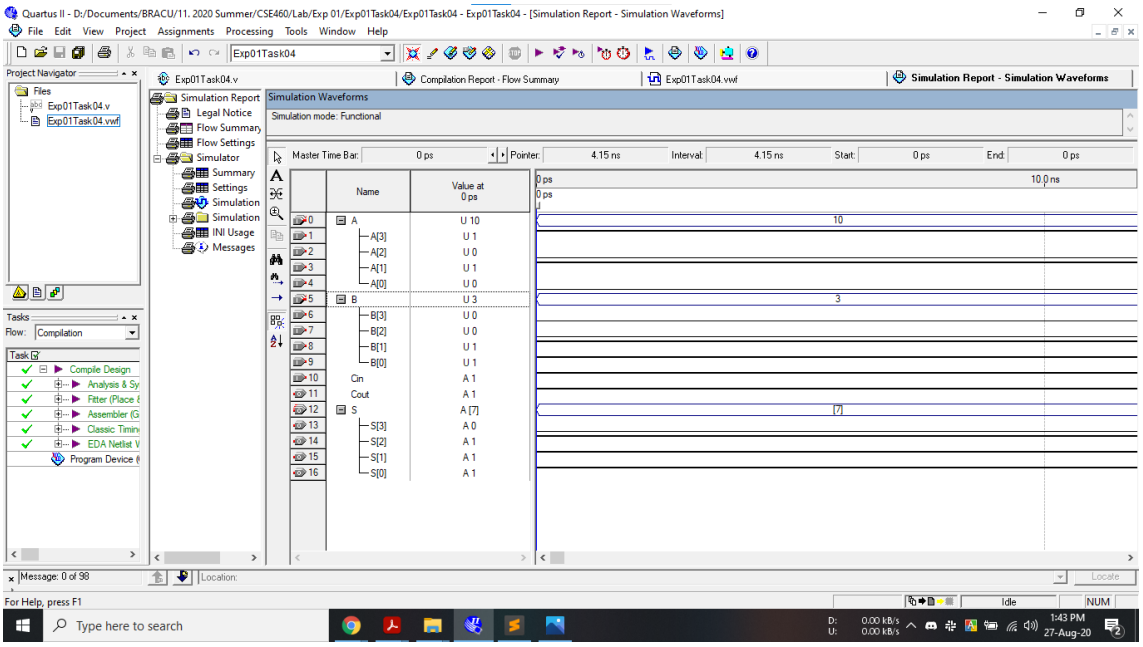

Figure 8: 4 bit adder-subtractor Vector Waveform Simulation

Discussion: Adder–Subtractor is a circuit that can add and subtract in a single combinational circuit. A 4-bit Adder–Subtractor will require 4 Full adder to perform the task. Before passing the values to full adder B is X-OR with the C_{in} and then the full adder does the bit wise addition of A, B and C_{in}. Here, C_{in} controls the circuit. If $C_{in} = 0$ it performs addition and when it is 1 it does the subtraction.

In the Verilog HDL file values of B was first X-OR with C_{in} and then passed in full adder for bit wise addition. Then after performing addition carry out was again passed in the second full adder as carry in. After four iteration we get the addition or subtracted values.