LAB 2:
ENCODER, PRIORITY ENCODER AND 8TO3 ENCODER

ARYA AGARWAL
Roll Number - 210070012

16 August 2022

PART A : 4TO2 ENCODER

1. Overview of the Experiment / Assignment:

- Encoder is a combination of gates that encodes a set of inputs into a given set of well defined output signals.
- 4 to 2 encoder means encoder with 4 inputs and 2 outputs.
- If Enable is 0, then the encoder will not work means all output 0.
- In this experiment our motive is to make circuit of encoder using AND and OR gates.

2. Experiment Setup or Approach to the Assignment:

- By observing the Truth table (given below) for 4to2 encoder, we try to formulate the possible circuit which is drawn below.

A	B	C	D	E = 1	Y1	Y0
0	0	0	1	1	0	0
0	0	1	0	1	0	1
0	1	0	0	1	0	0
1	0	0	0	1	1	1

- Therefore, the possible figure of encoder drawn is given in Fig 1

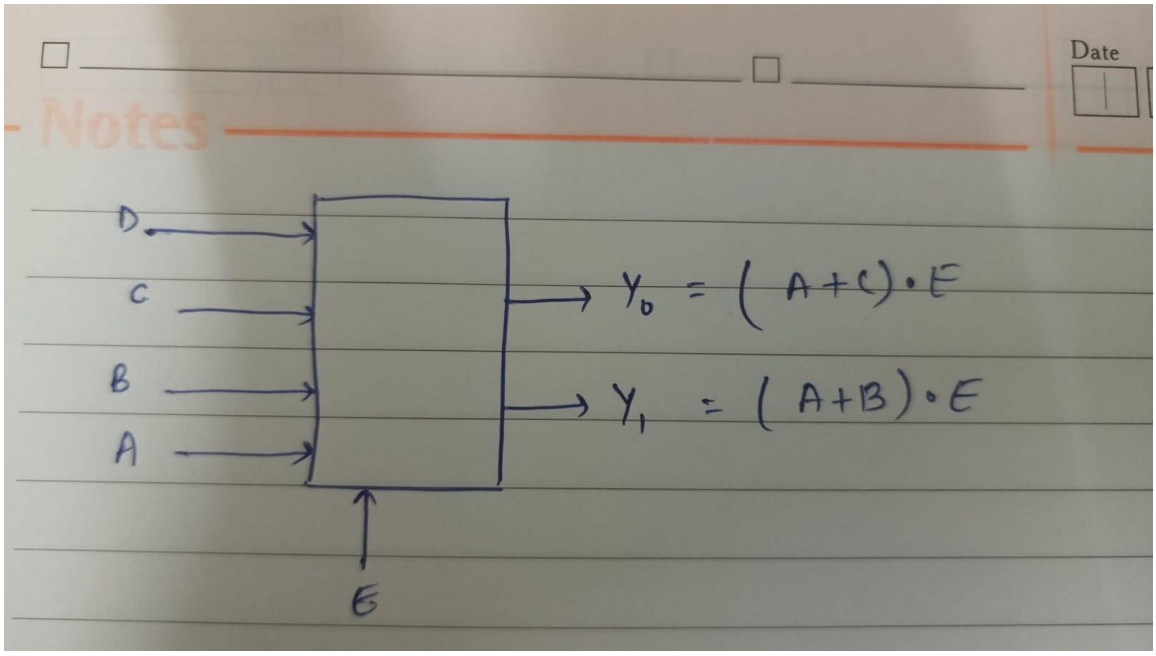


Fig 1

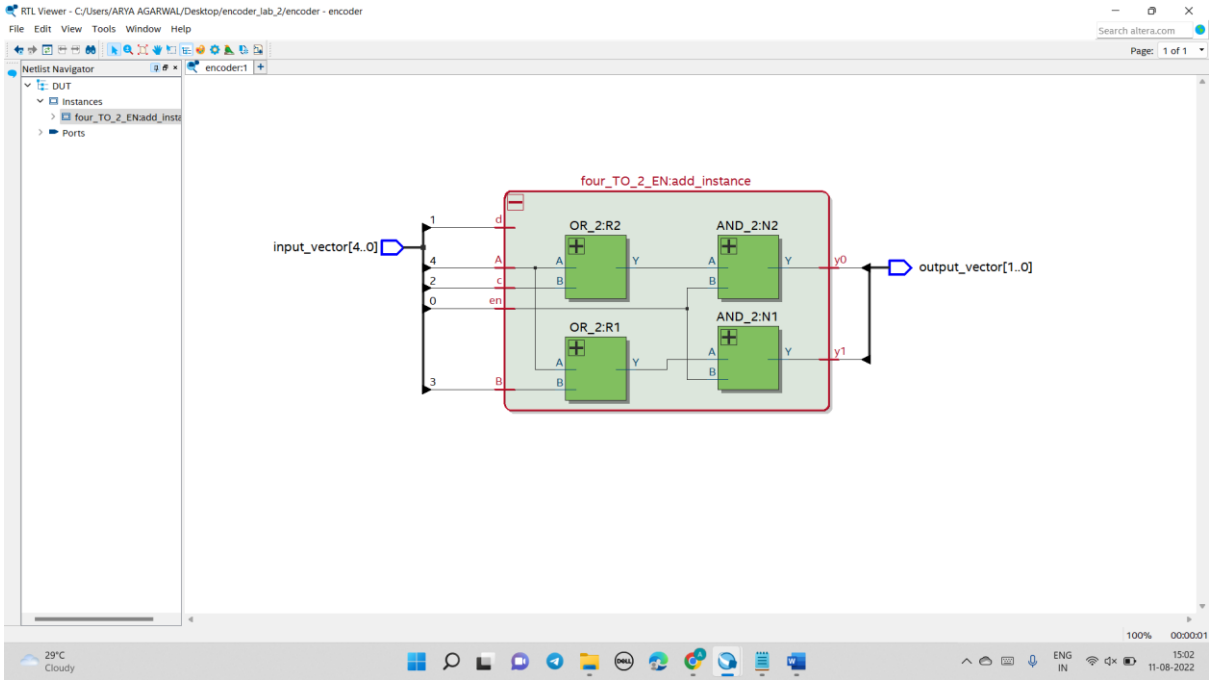
2.1 Design Code and Documentation:

Now once we have drawn the circuits on paper, its time to describe the circuit using code in vhdl language in Quartus. The dut, gates files will almost be same except some minor changes in both case.

The code for the main file is written below which is used to describe the circuit.

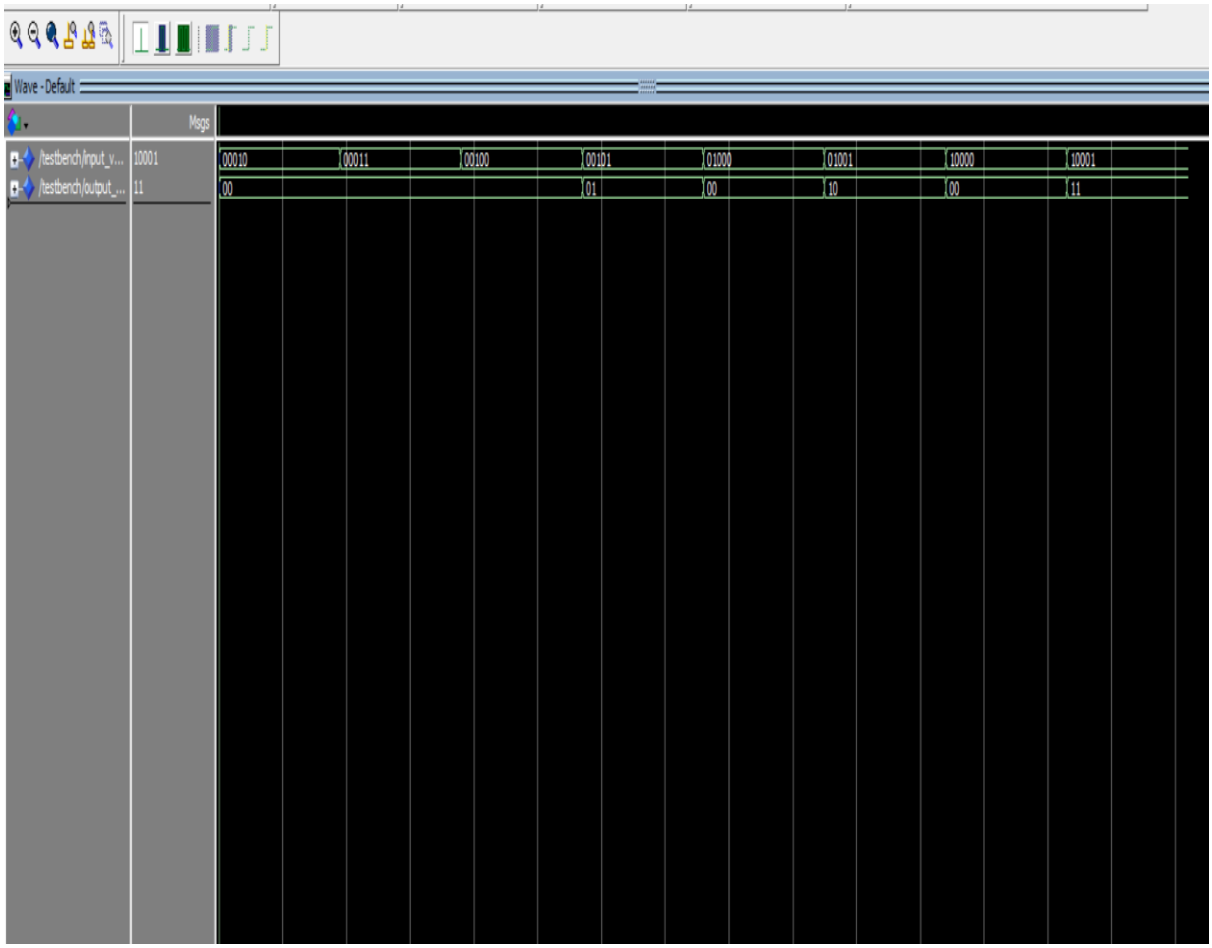
```
library ieee;
use ieee.std_logic_1164.all;

library work;
use work.Gates.all;

entity four_TO_2_EN is
 port (A,B,c,d,en: in std_logic;
       y1,y0: out std_logic);
end entity four_TO_2_EN;

architecture struct of four_TO_2_EN is
signal s1,s2:std_logic;
begin
R1:OR_2
 port map(A=>A, B=>B, Y=>s1);
R2:OR_2
port map(A=>A, B=>C , Y=>s2);
N1: AND_2
port map(A=>s1, B=>en, Y=>Y1);
N2: AND_2
port map(A=>s2, B=>en, Y=>Y0);
end struct;
```

It is clear from the code that I have used 2 OR and 2 AND gates to fulfil my circuit. The RTL view of the code written above is shown below.

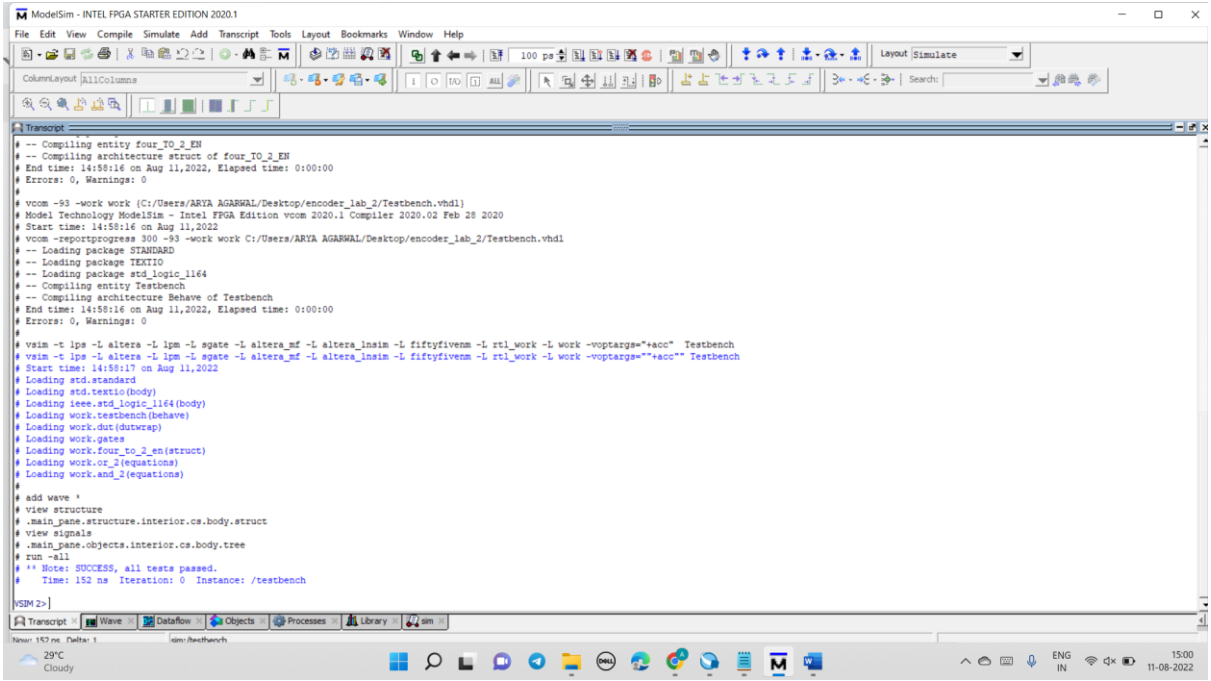


3. Observations:

After you run the analysis of code with no error. The next step is to run the simulation. For this we again use the Modelsim - Altera Software. We also need the Testbench and the tracefile (already provided) to run the simulation. The simulation is shown below.



TRANSCRIPT(all test cases passed successfully)



PART B: PRIORITY ENCODER

1. Overview of the Experiment / Assignment:

- The job of a priority encoder is to produce a binary output address for the input with the highest priority.
- It is also a 4to2 encoder with same number of inputs and outputs but with different characteristics.
- Here in this experiment we will draw the circuit for Priority Encoder using the truth table.

2. Experiment Setup or Approach to the Assignment:

- Below is the truth table for the priority encoder.

Note - X represents Don't Care meaning the output does not depend on the particular input.

A	B	C	D	V	Y1	Y0
0	0	0	0	0	0	0
0	0	0	1	1	0	0
0	0	1	X	1	0	1
0	1	X	X	1	1	0
1	X	X	X	1	1	1

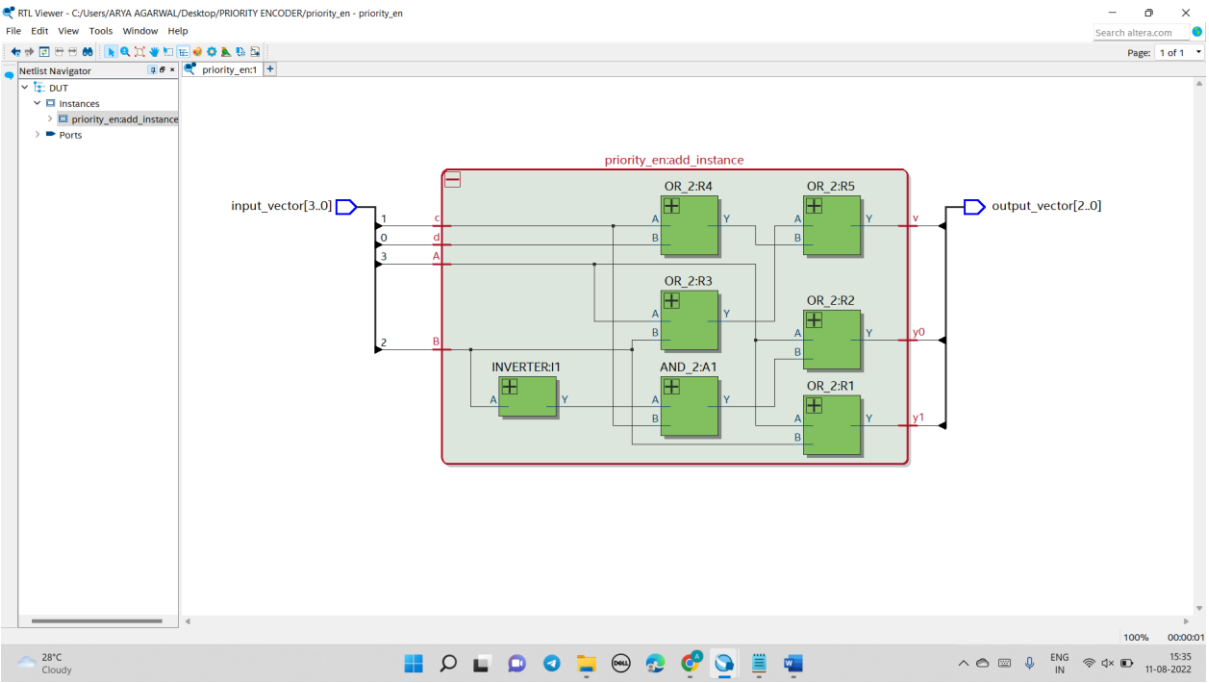
2.1 Design Code and Documentation:

Now once we have drawn the circuits on paper, its time to describe the circuit using code in vhdl language in Quartus. The dut, gates files will almost be same except some minor changes in both case.

Now its time to write the code for the circuit drawn above in order to run its simulation. We have already made another folder. Now we will make a file in which I would write the code to describe the circuit. The same code is also written below.

```
library ieee;
use ieee.std_logic_1164.all;

library work;
use work.Gates.all;

entity priority_en is
 port (A,B,c,d: in std_logic;
       y1,y0,v: out std_logic);
end entity priority_en;

architecture struct of priority_en is
signal s1,s2,s3,s4:std_logic;
begin
R1:OR_2
 port map(A=>A, B=>B, Y=>y1);
I1: INVERTER
 port map(A=>b, y=>s1);

A1: AND_2
port map(A=>s1, B=>c, Y=>s2);
R2:OR_2
 port map(A=>A, B=>s2, Y=>y0);
R3:OR_2
 port map(A=>A, B=>B, Y=>s3);
R4:OR_2
 port map(A=>c, B=>d, Y=>s4);
 R5:OR_2
 port map(A=>s3, B=>s4, Y=>v);

end struct;
```

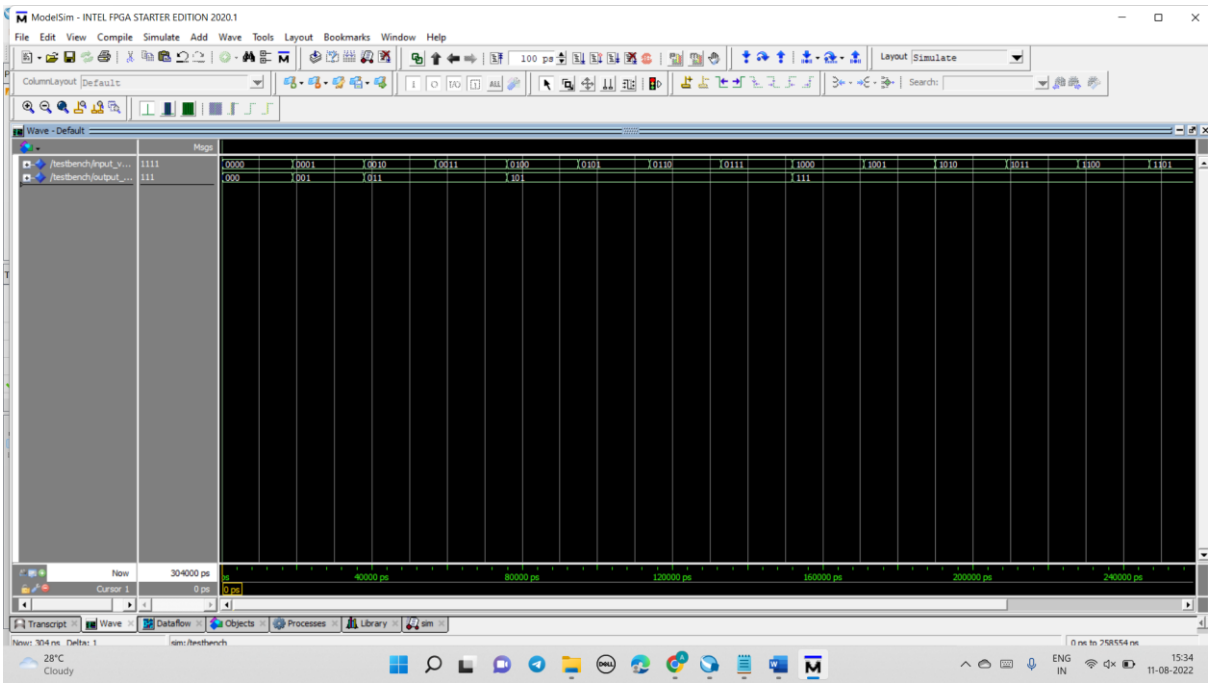
RTL VIEW



3. Observations:

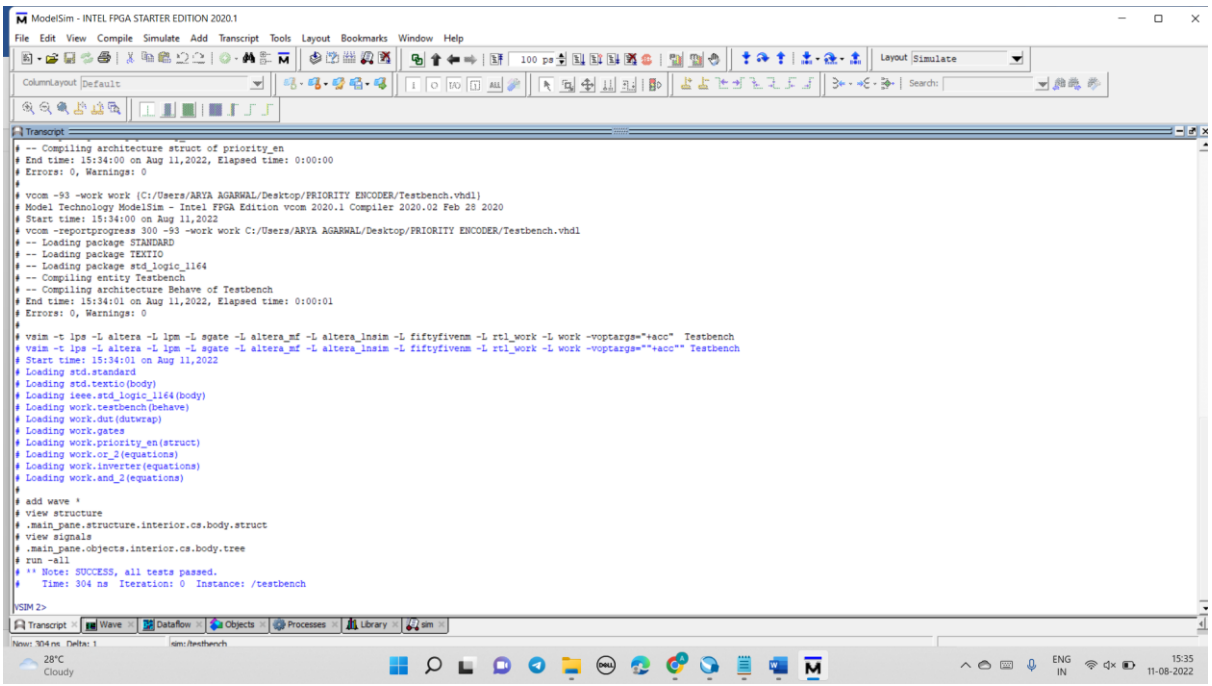
After you run the analysis of code with no error. The next step is to run the simulation. For this we again use the Modelsim - Altera Software. We also need the Testbench and the tracefile (already provided) to run the simulation.

The simulation thus obtained for the above code/circuit is shown below.



It can be easily observed that the above simulation correctly matches with the trace file as well as the Table which we wrote in Experiment Setup section. Thus we can conclude that our Experiment is successful.

TRANSCRIPT(all test cases passed successfully)



PART C: 8TO3 ENCODER

1. Overview of the Experiment.

- 8 to 3 encoder is an extended version of 4 to 2 encoder that we have already made.
- We know that number of output n is given as log to the base 2 and argument number of inputs. Thus for 8 input we have 3 output.
- The purpose of this experiment is to draw the circuit of 8to3 encoder using only 4to2 encoder and OR gates.
- Then as usual we are required to code the circuit made and run its simulation based on the trace file given.

2. Experiment Setup or Approach to the Assignment:

- Firstly we need to make the truth table based on the characteristics of the 8to3 encoder.
- The truth table for 8to3 encoder is drawn below. Do not forget the Enable (E) input that we used in the first experiment.

Note - X represents Don’t Care meaning the output does not depend on the particular input.

I7	I6	I5	I4	I3	I2	I1	I0	E	A2	A1	A0
X	X	X	X	X	X	X	X	0	0	0	0
0	0	0	0	0	0	0	1	1	0	0	0
0	0	0	0	0	0	1	0	1	0	0	1
0	0	0	0	0	1	0	0	1	0	1	0
0	0	0	0	1	0	0	0	1	0	1	1
0	0	0	1	0	0	0	0	1	1	0	0
0	0	1	0	0	0	0	0	1	1	0	1
0	1	0	0	0	0	0	0	1	1	1	0
1	0	0	0	0	0	0	0	1	1	1	1

Based on the truth table drawn above it is easy to observe the following equations of the output of the encoder. A0 = E and (I1 + I3 + I5 + I7)

A1 = E and (I2 + I3 + I6 + I7)

A2 = E and (I4 + I5 + I6 + I7)

Now as we are directed to only use 4to2 encoder whose output we have already understood from the first experiment.

2.1 Design Code and Documentation:

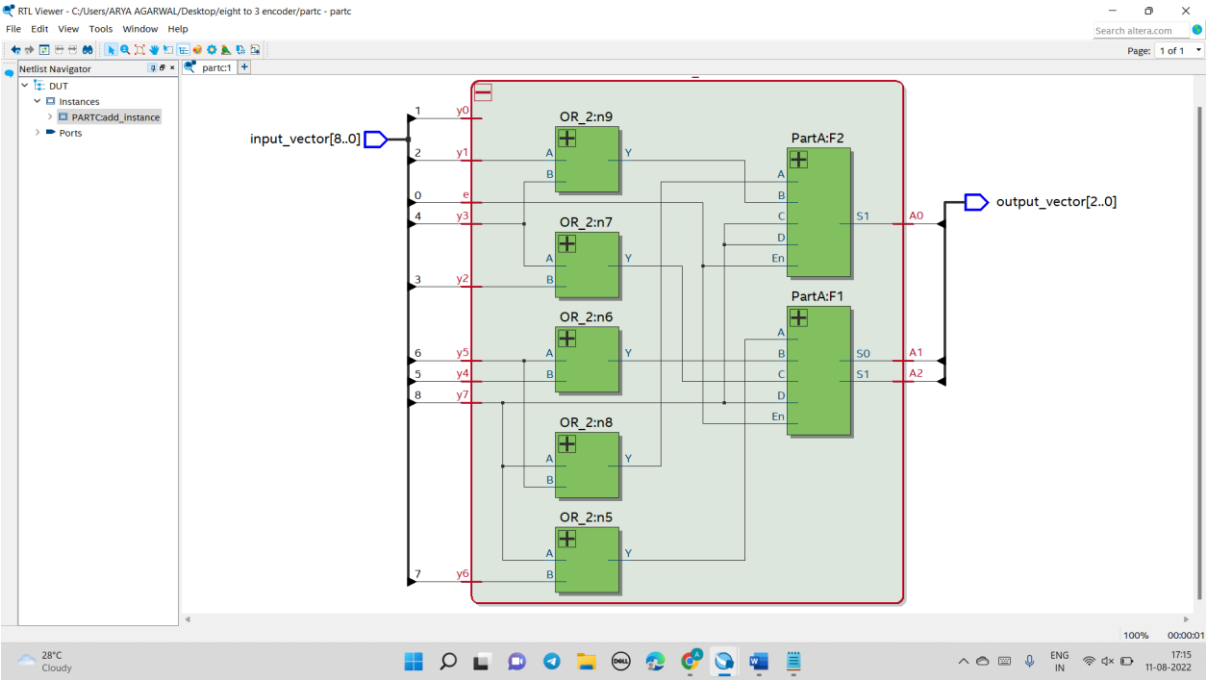
Now once we have drawn the circuits on paper, its time to describe the circuit using code in vhdl language in Quartus. The dut, gates files will almost be same except some minor changes in both case.

Now its time to write the code for the circuit drawn above in order to run its simulation. We have already made another folder. Now we will make a file in which I would write the code to describe the circuit. The same code is also written below.

We have declared an entity named PARTA in order to reduce complexity.

```
library ieee;
use ieee.std_logic_1164.all;
library work;
use work.Gates.all;
entity PartA  is
  port (A, B, C, D, En: in std_logic; S0, S1: out std_logic);
end entity PartA;
architecture Struct of PartA is
 signal U,V: Std_logic;
begin
    -- component instances
     n1 : OR_2   port map (A => A, B => C, Y => U); -- complete this port map
   n2 : OR_2   port map (A => A, B => B, Y => V);
   n3 : AND_2   port map (A => U, B => En, Y => S0);
   n4 : AND_2   port map (A => V, B => En, Y => S1);



 end Struct;

--------------PARTC------------
library ieee;
use ieee.std_logic_1164.all;
library work;
use work.Gates.all;

entity PARTC  is
port (y0, y1, y2, y3, y4, y5, y6, y7, e:in std_logic; A2, A1, A0: out std_logic);
end entity PARTC;
architecture struct of PARTC is
component PARTA is
 port (A, B, C, D, En: in std_logic; S0, S1: out std_logic);
 end component;
 signal s1, s2, s3, s4, s5, s6: Std_logic;
 begin
 n5 :  OR_2   port map (A => y7, B => y6, Y => s1);
 n6 :  OR_2   port map (A => y5, B => y4, Y => s2);
 n7 :  OR_2   port map (A => y3, B => y2, Y => s3);
 n8 :  OR_2   port map (A => y7, B => y5, Y => s4);
 n9 :  OR_2   port map (A => y1, B => y3, Y => s5);
 F1 :  PARTA  port map (A => s1, B => s2, C => s3, D => y7, En => e, S0 => A1, S1 => A2);
 F2 :  PARTA  port map (A => s4, B => s5, C => y7, D => y7, En => e, S0 => s6, S1 => A0);


 end Struct;
```

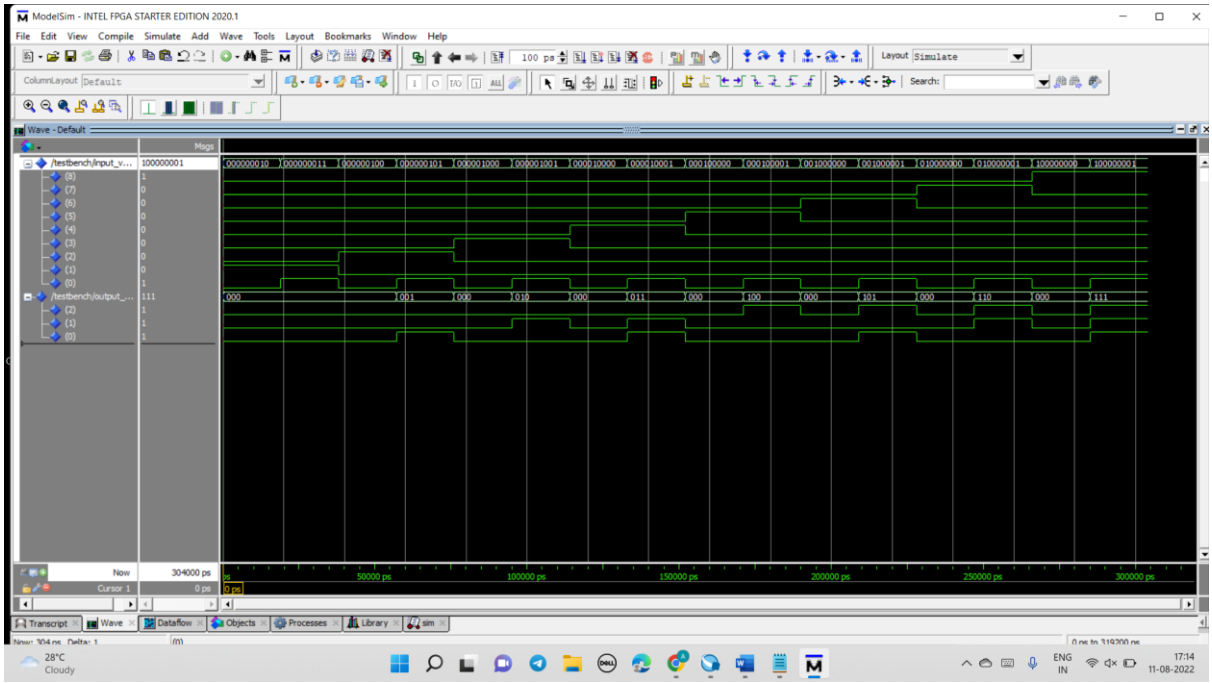
RTL VIEW



3. Observations:

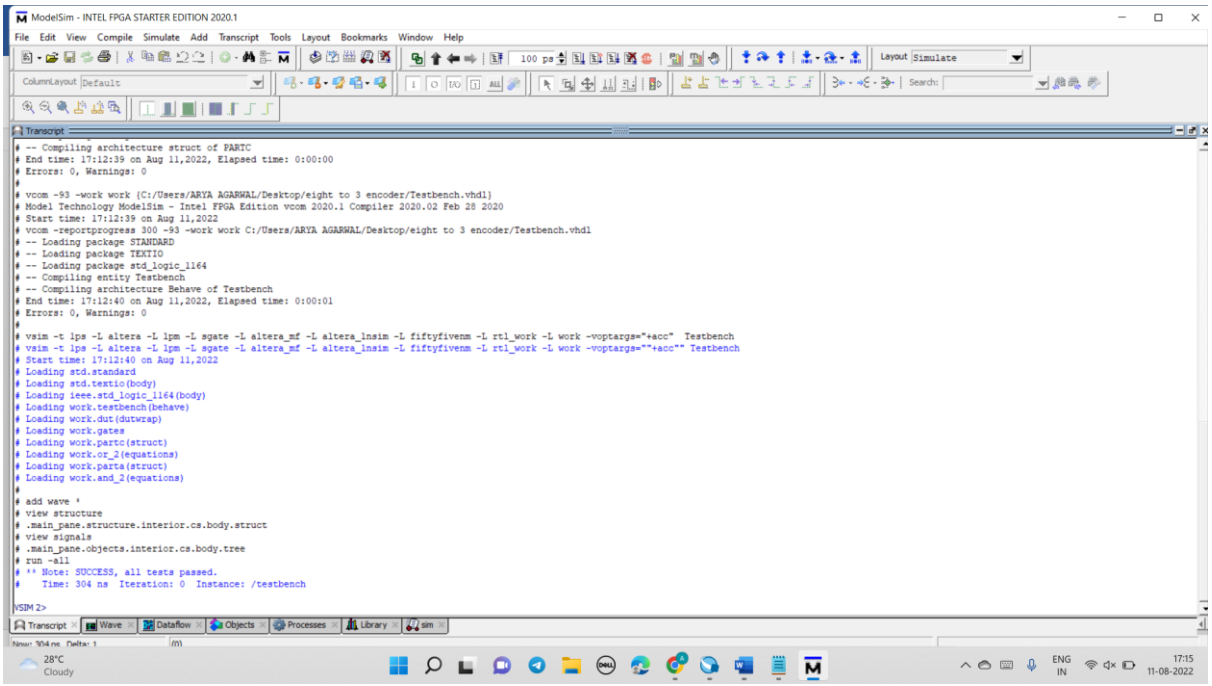
After you run the analysis of code with no error. The next step is to run the simulation. For this we again use the Modelsim - Altera Software. We also need the Testbench and the tracefile (already provided) to run the simulation.

The simulation thus obtained for the above code/circuit is shown below.



It can be easily observed that the above simulation correctly matches with the trace file as well as the Table which we wrote in Experiment Setup section. Thus we can conclude that our Experiment is successful.

TRANSCRIPT(all test cases passed successfully)



4. Reference for all above experiment.

All the images have been taken from the experiments done in lab on Quartus software. Special mention to the Modelsim software which is used to run the simulation.