

CSE 460

VLSI LAB

ASSIGNMENT 01

**Submitted by**

Md. Nasimuzzaman

Id: 19101051

CSE-460 Summer-2022

Submission Date: 17th August 2022

**Lab Assignment 01**

**Problem statement**

Write a Verilog code to implement the AOI-32 gate and verify it with the timing diagram in Quartus.

**Code**

//verilog code for AND-OR-INVERT(AOI) gate

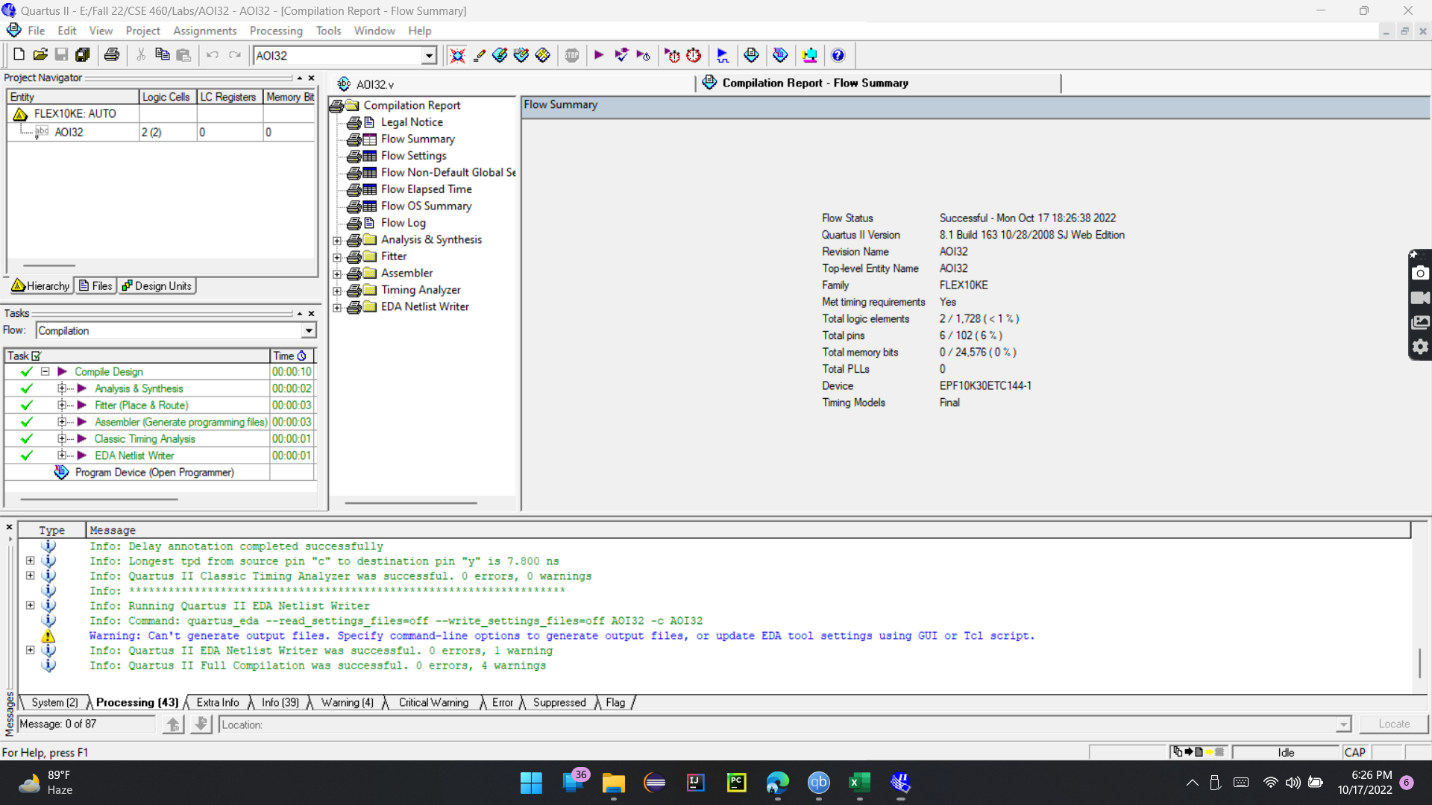
module AOI32 (input a,b,c,d,e, output y);

assign y = ~((a & b & c) | (d & e)) ;

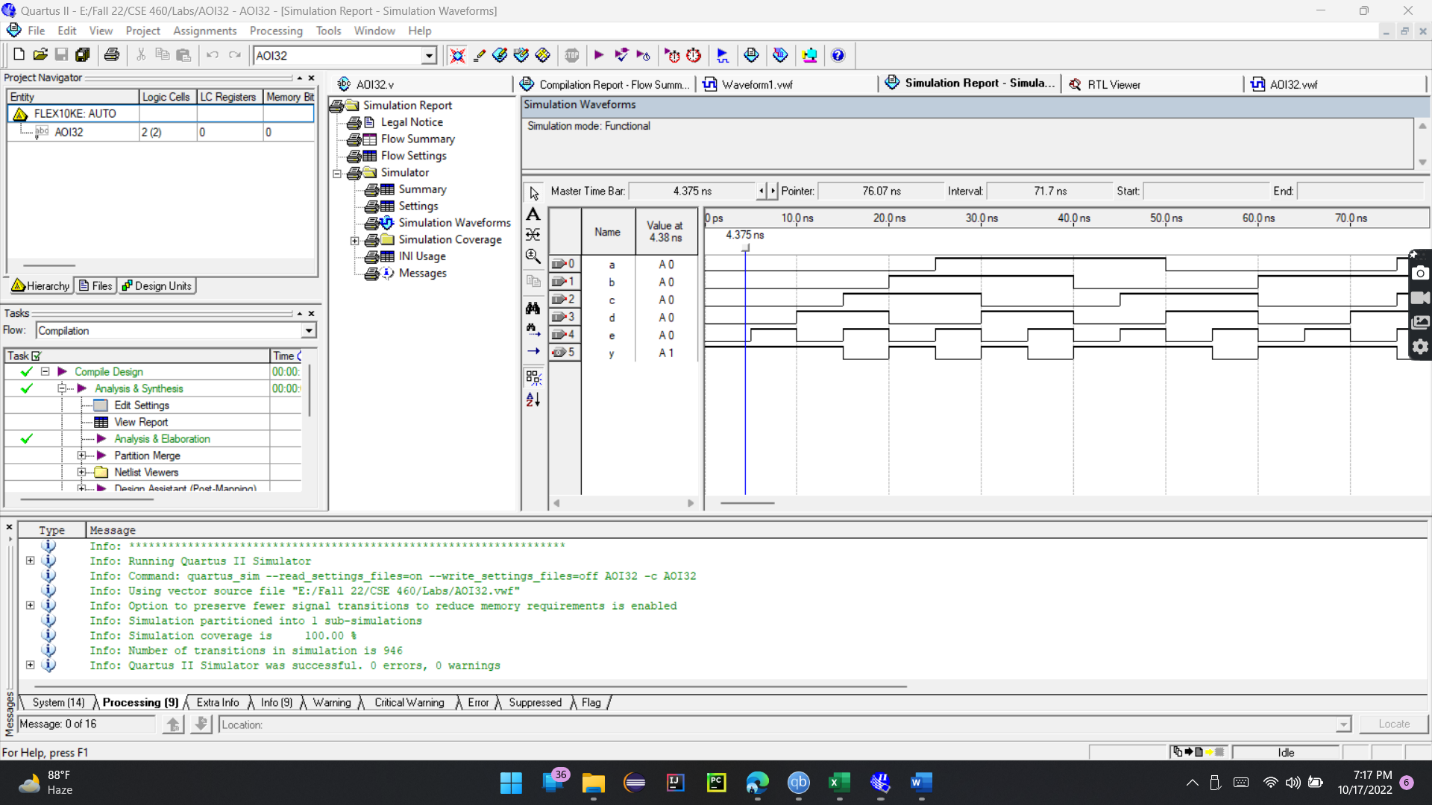
endmodule

//end of verilog code

**Compilation Report**



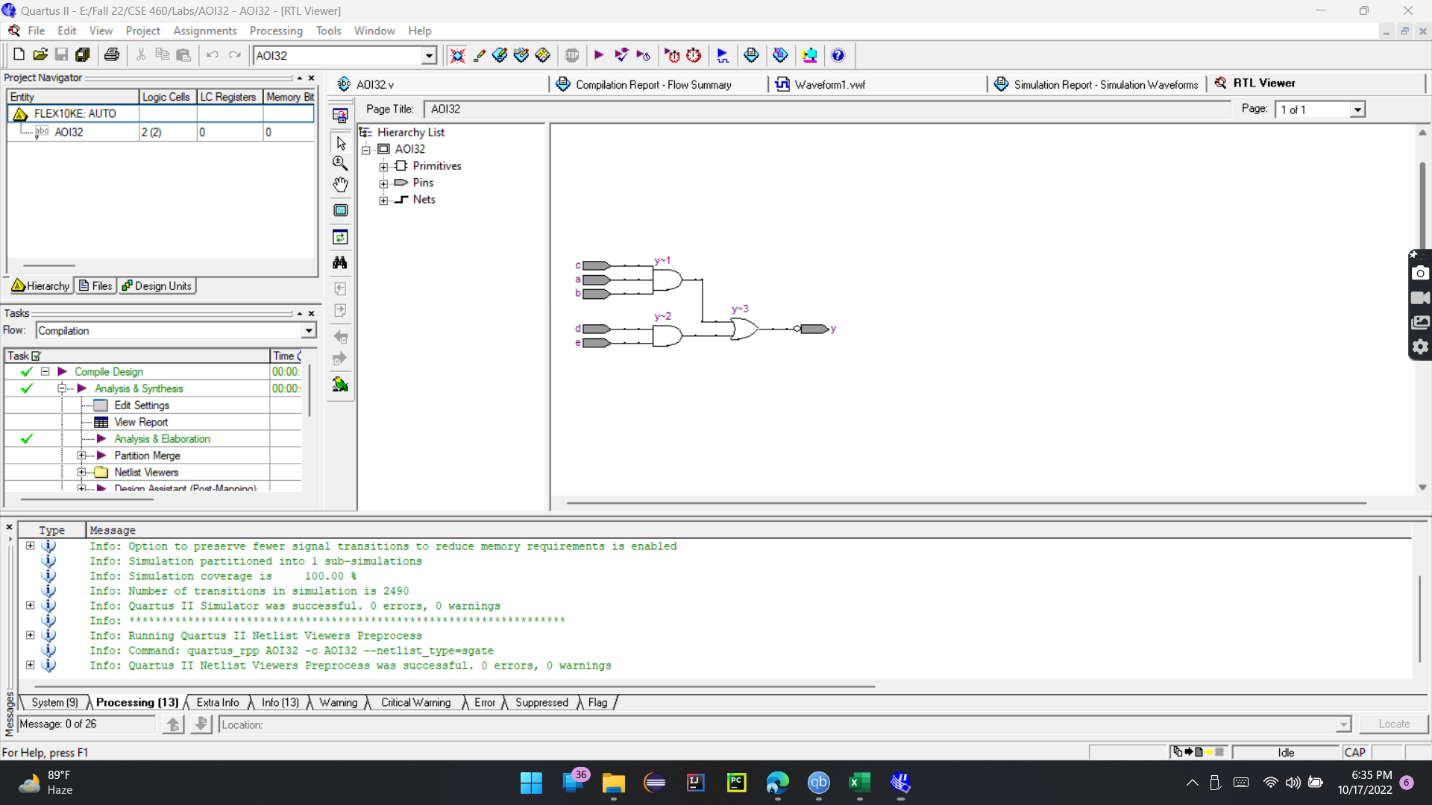
**Simulation Report**



**Truth Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **E** | **Y = ~ ((A.B.C) | (D.E))** |
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 |

**RTL View**

****

**Explanation of how the timing diagram manifests the Truth Table**

In this task, I’ve simulated an AOI32 [And Or Invert] gate, which is based on five inputs and provides one output. For input, A binary value changes faster, so I’ve given the least time to change the value in A. The binary value of input B changes slowly compared to input A. Accordingly, the value of input C changes slower than input B. Input D changes slower than input C. Input E changes slower than input D. In that case; I’ve assigned the values of these inputs depending on their changing rate, which is why I’ve taken 50ns time for input A, 40ns for input B, 30ns for input C, 20ns for input D, 10ns for input E. I’ve taken 10ns difference of time for each input.

Now, if we look at the timing simulation and analyze it, we will be able to see the truth table representation in the timing diagram. In the timing diagram, 0ps to 10ns, if we notice the output value Y = 1 for input A=0, B=0, C=0, D=0 and E=0, also output Y=1 when input A=0, B=0, C=0, D=0 and E=1. Similarly, output Y=0 when input A=0, B=0, C=1, D=1, E=1.

So, according to the simulations and what has been discussed, it represents that the timing diagram manifests the truth table.