

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

**by**

**Ishraq Ahmed Esha  
19301261**

**A Lab Report 1 submitted to the CSE460 VLSI Design  
Course of Sec: 5**

**Brac University  
October 17, 2022**

## Problem Statement

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

## Code

```
module assignment_1(a,b,c,d,e,y);  
    input a,b,c,d,e;  
    output y;  
  
    assign h = a&b&c;  
    assign k = d&e;  
    assign r = h|k;  
    assign y = ~r;  
  
endmodule
```

## Compilation Report

Quartus II - F:\Study\VL5\CSE460-Lab\Assignment 1\assignment\_1 - assignment\_1.v - [Compilation Report - Flow Summary]

Project Navigator

| Entity        | Logic Cells | LC Registers | Memory Bits |
|---------------|-------------|--------------|-------------|
| FLEX10KE-AUTO |             |              |             |
| assignment_1  | 2 (0)       | 0            | 0           |

Tasks

| Task   | Time     |
|--|----------|
| ✓ [x] Compile Design                             | 00:00:04 |
| ✓ [x] [x] Analysis & Synthesis                   | 00:00:01 |
| ✓ [x] [x] Fitter (Place & Route)                 | 00:00:01 |
| ✓ [x] [x] Assembler (Generate programming files) | 00:00:01 |
| ✓ [x] [x] Classic Timing Analysis                | 00:00:00 |
| ✓ [x] [x] EDA Netlist Writer                     | 00:00:01 |
| Program Device (Open Programmer)                 |          |

Flow Summary

|                         |   |
|-------------------------|---|
| Flow Status             | Successful - Mon Oct 17 21:42:14 2022   |
| Quartus II Version      | 8.1 Build 163 10/20/2008 SJ Web Edition |
| Revision Name           | assignment_1                            |
| Top-level Entity Name   | assignment_1                            |
| Family                  | FLEX10KE                                |
| Met timing requirements | Yes                                     |
| Total logic elements    | 2 / 1,728 (< 1 %)                       |
| Total pins              | 6 / 102 (< 6 %)                         |
| Total memory bits       | 0 / 24,576 (0 %)                        |
| Total PLLs              | 0                                       |
| Device                  | EPF10K30ETC144-1                        |
| Timing Models           | Final                                   |

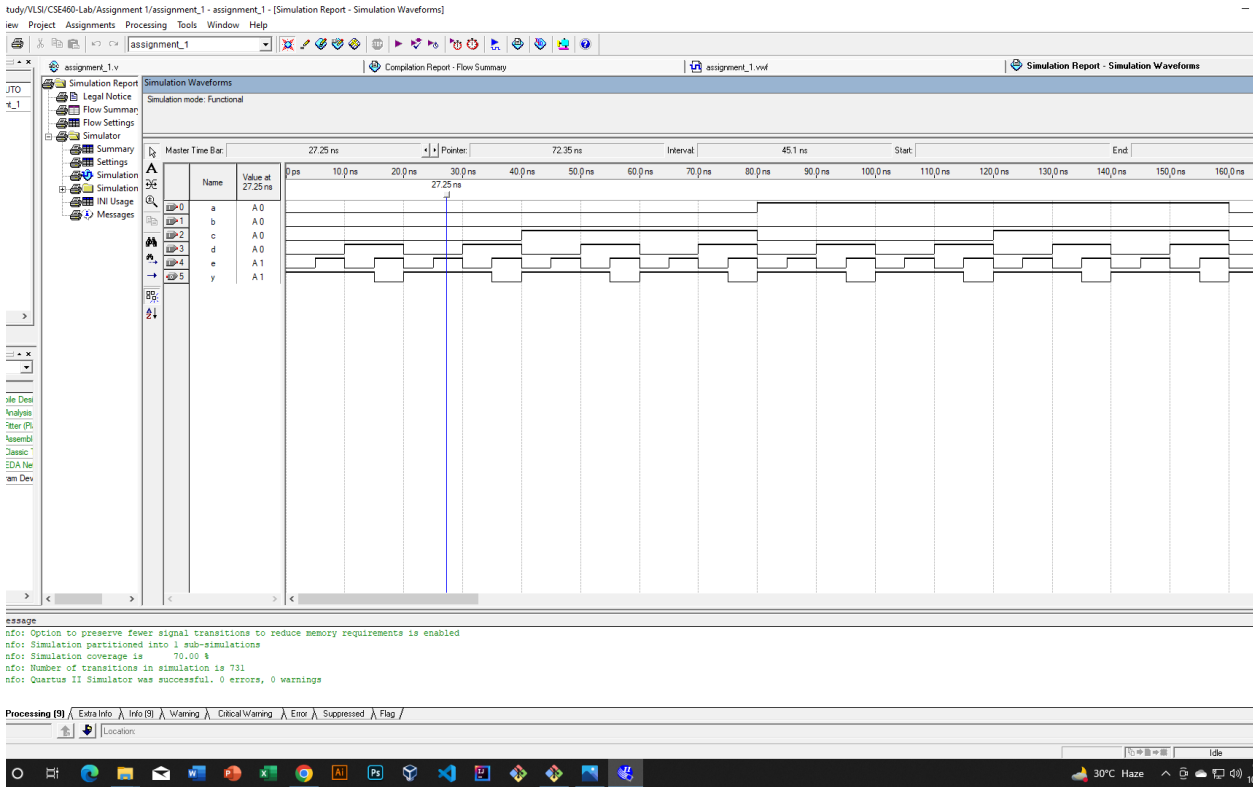
Message: 0 of 90

System (2) | Processing (46) | Info (39) | Warning (7) | Error | Critical Warning | Flag |

For Help, press F1

30°C Haze 9:42 PM 10/17/2022

Simulation Report

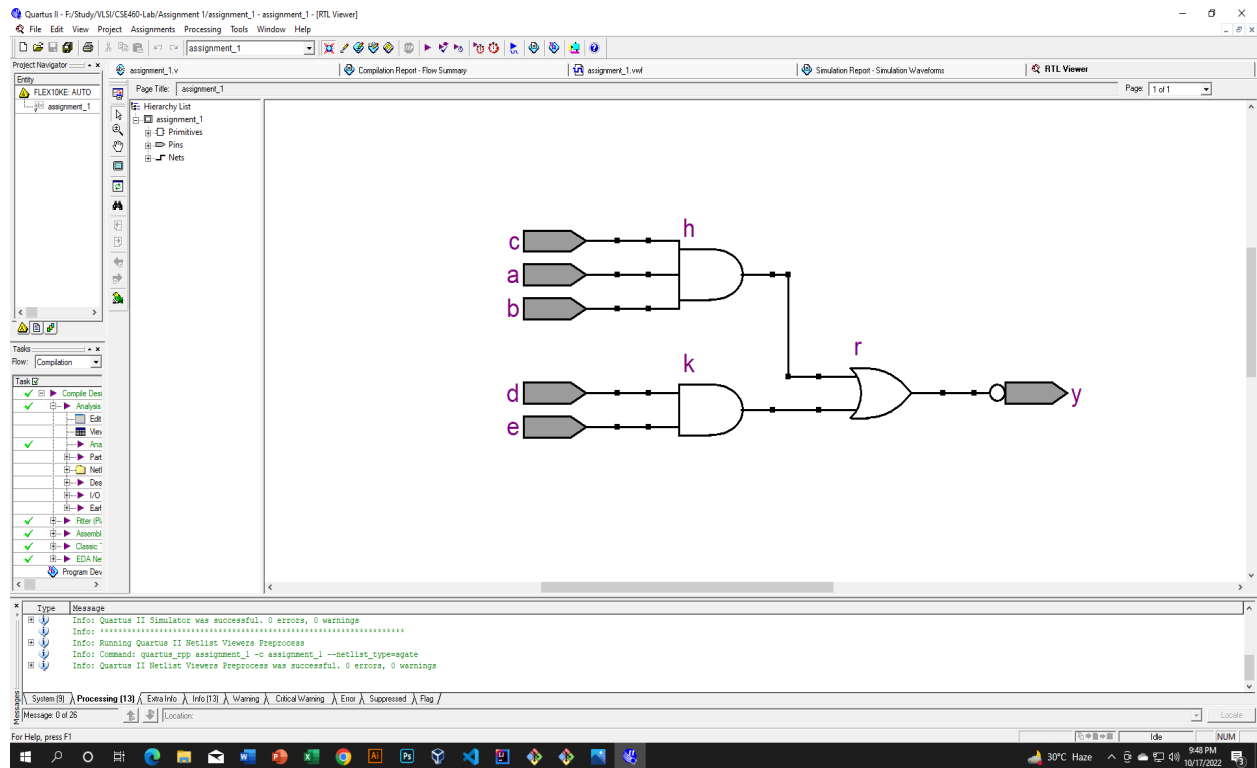


Truth Table

| a | b | c | d | e | y |
|---|---|---|---|---|---|
| 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 timing diagram manifests the Truth Table

In the AOI-32 gate, we have 5 inputs. We know that the total number of combinations equals 2 to the power of the number of inputs ( $2^5 = 32$ ). So, for this gate there can be 32 logic combinations. Here, the timing diagram manifests the truth table. For example, if we consider the first combination, the truth table shows us all the inputs are 0, and the output will be 1. Now, if we look at the timing diagram when all the inputs are low or logic 0, the output y is 1. In addition, if we move to the second combination there, we have a,b,c,d equals 0 and e equals 1, and the output is 1. Observing the timing diagram for the second combination shows us the output is still 1, which matches the truth table.

Now, if we consider any random combination from the truth table, such as let's consider input a is 1 and b,c,d,e inputs are 0 from the timing diagram, we can observe that the output shows 1, which matches with our truth table. To achieve the proper combinations in the timing diagram, we adjust the clock time period. For the AOI-32 gate, we have 5 inputs which are a,b,c,d, and e, respectively. To achieve the combination, we set the different clock periods for the inputs. To elaborate, we set 10ms, 20ms, 40ms, 80ms, and 160ms for the inputs e,d,c,b, and a, respectively. This is clearly visible in the timing diagram, and like this, we achieved all 32 combinations for the AOI-32 gate. So, from the above discussion, we can clearly say that the timing diagram manifests the truth table.