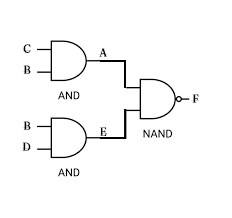
**Object Oriented Electronic Modeling – CA#1 Report**

Amir Abbas Moumeni Zadeh – Mahdis Mirzaei

In this project, we simulated a digital circuit using an event-driven model in C++. We reused the *Wire* and *Gate* classes from our previous project, where basic gate behavior and signal management were already implemented. The first new component we added was the *IO* class, which handled parsing input files, testbenches, and managing user interaction. Then, we implemented a *Circuit* class to organize gates and wires into a complete circuit structure, connecting everything based on the input netlist. The core of the simulation lies in the *simulate* function, which processes time-ordered input transitions and propagates changes through the circuit while maintaining correct timing behavior. The simulation also supports unknown values ('X') when gate delays exceed testbench timing, ensuring realistic modeling.

Some Simple Logic Circuit

We simulated the Above Circuit just for test. In this Example Delay values are like this , AND : 11 , NAND :12

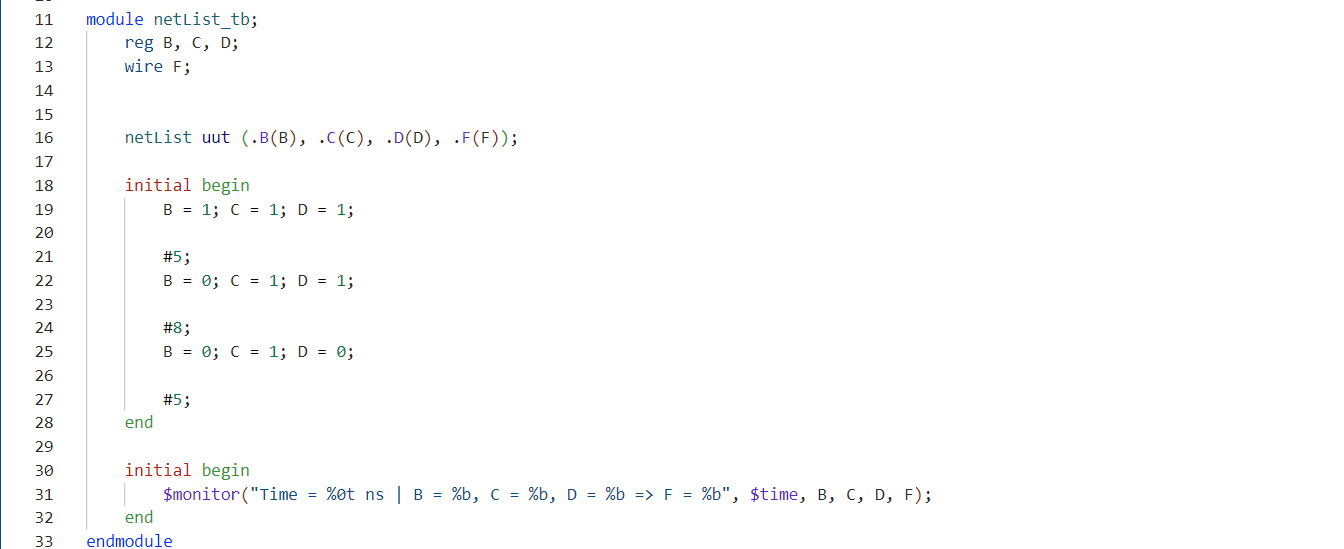
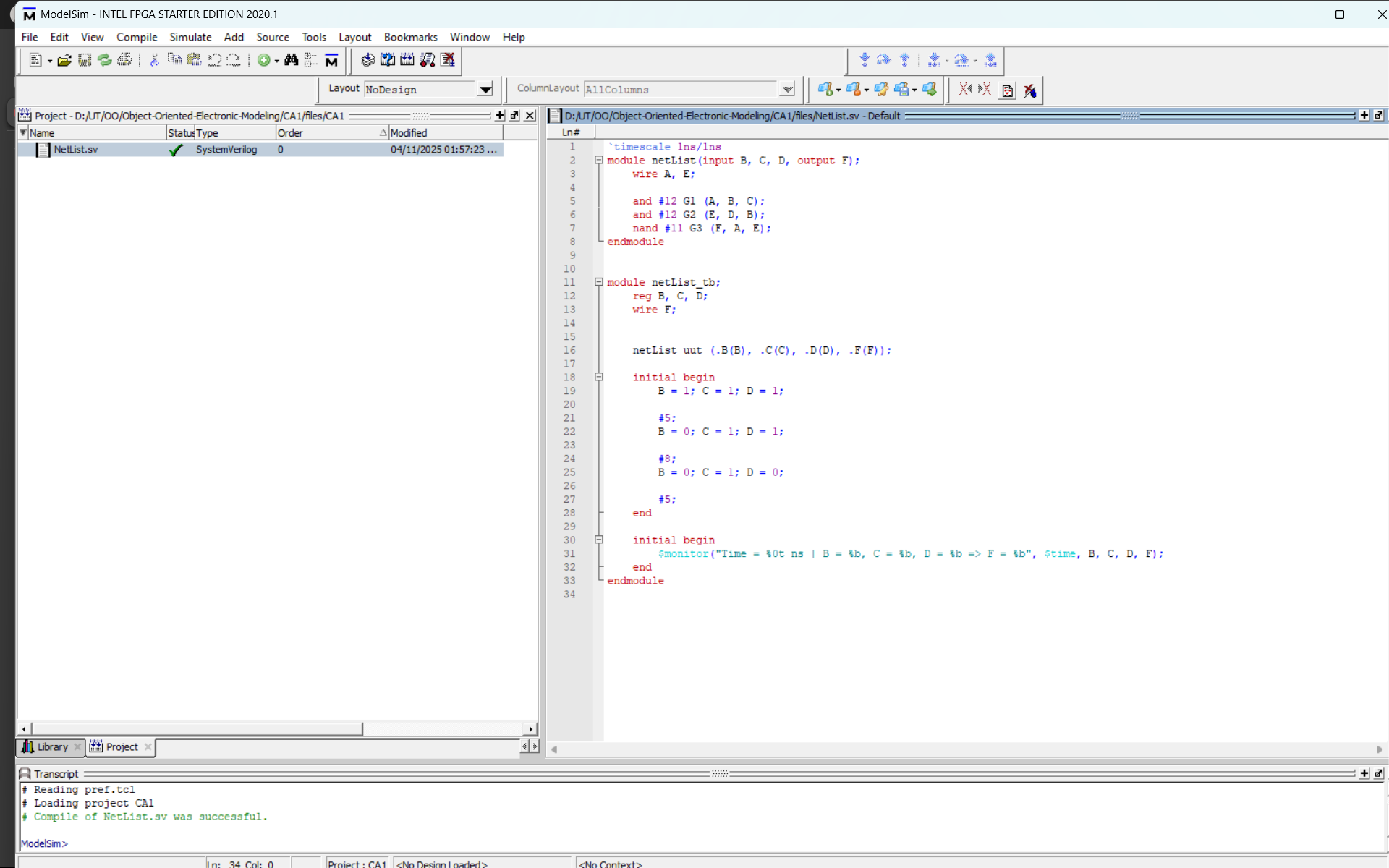
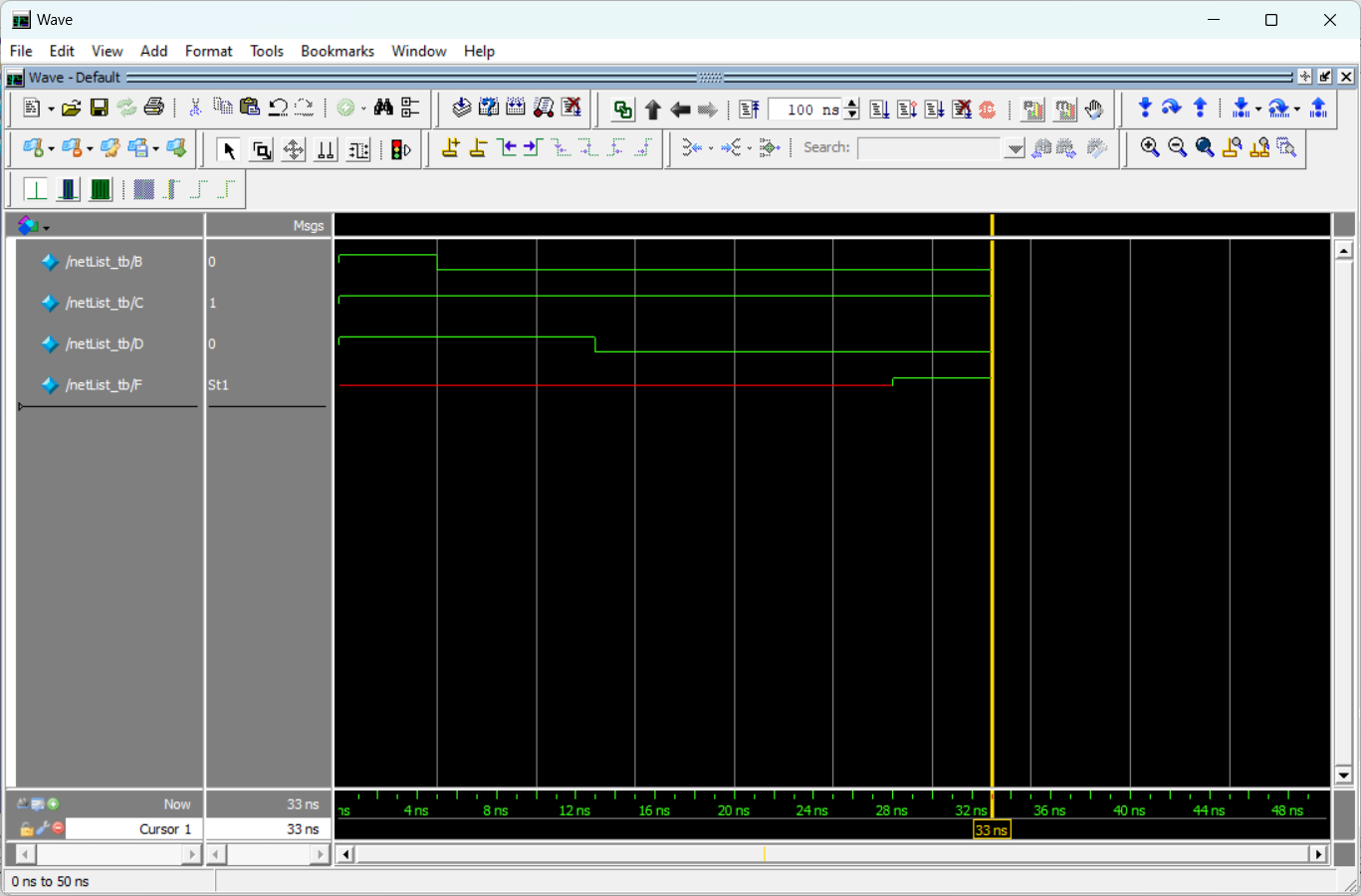
Hand Simulation :

A is Ready at : 12 , E is Ready at : 12

So F will be Ready at : 11 + 12 = 23ns

So the worth-case delay is 23ns.

The SystemVerilog Description of mentioned Circuit.



Simulation Result

ModelSim Project

Also the testbench