**Feedforward Neural Network Design**

**(HSCD Assignment 1)**

Name  **: Ashutosh Rajendra Karve**

Roll no **: 01021**

Email **:** [**2024ht01021@wilp.bits-pilani.ac.in**](mailto:2024ht01021@wilp.bits-pilani.ac.in)

Contact **: +91 9765541324**

Date **: 10/10/2024**

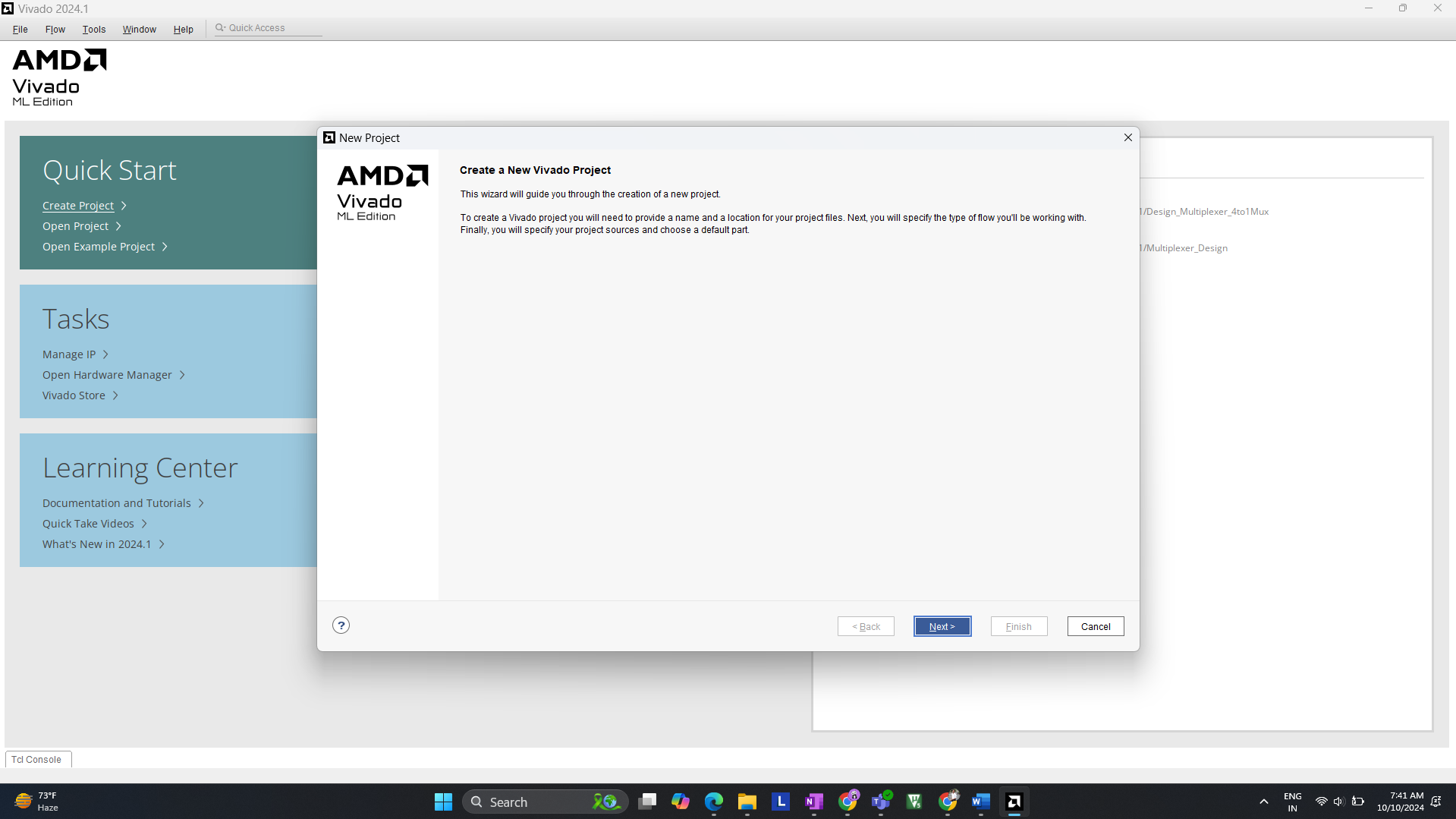
Place **: Pune, Maharashtra**

**1. Introduction:**

In this assignment, a **Feedforward Neural Network** (FNN) with one hidden layer was implemented using Verilog HDL. The network architecture consists of four inputs, a hidden layer with three neurons, and an output layer with two neurons. Each neuron uses a **ReLU (Rectified Linear Unit)** activation function, which ensures that the output of the neurons is non-negative.

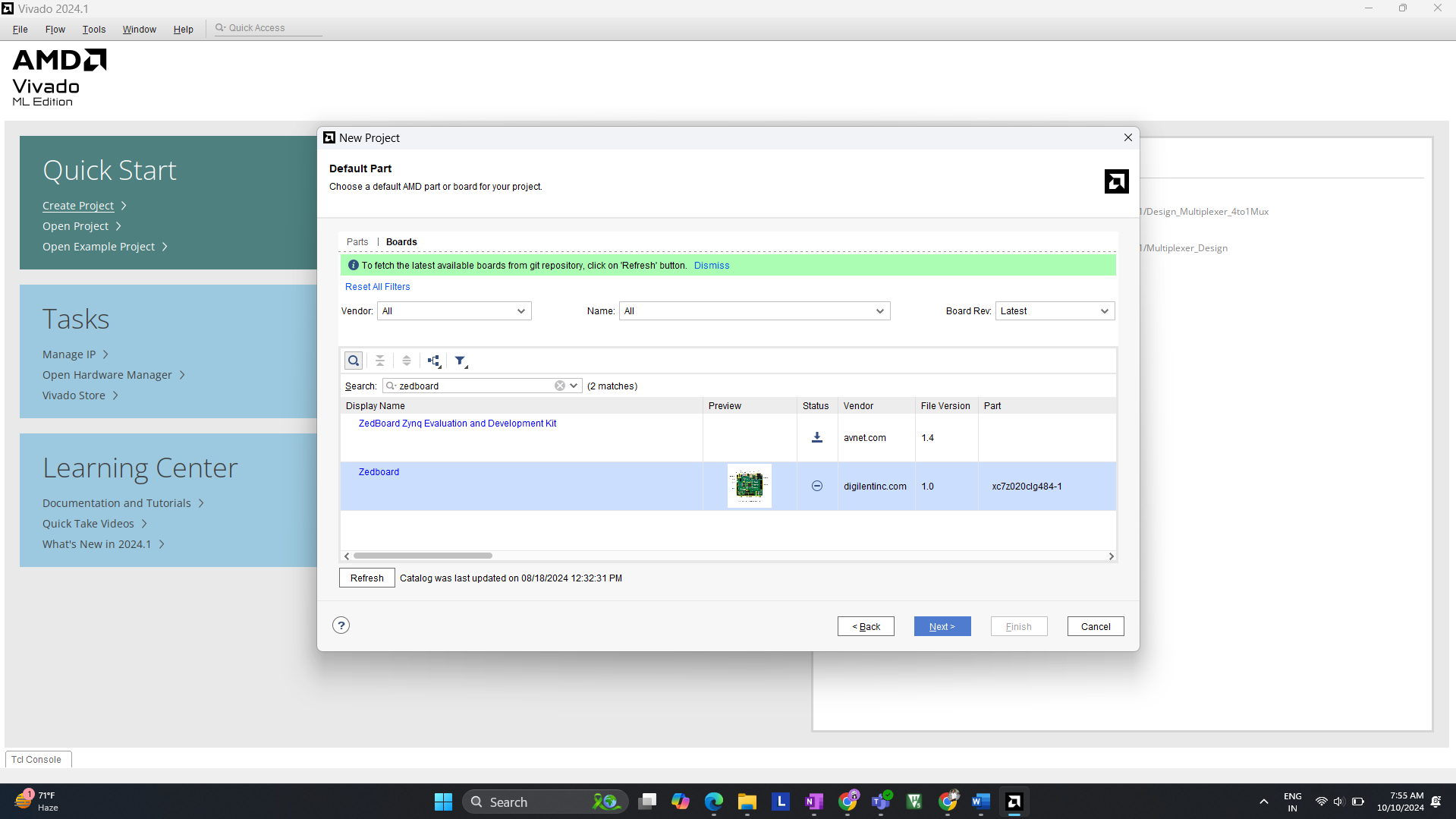
**2. Vivado Project Setup**

**2.1 Project Creation**

The project was created in Vivado, targeting the ZedBoard (Zynq-7000). The RTL project was set up for Verilog coding. *(Create Project >> Next)*

A screenshot of a computer

Description automatically generated  
*(Project Name (FeedForward\_NN) >> Next)*A screenshot of a computer

Description automatically generated  
*(Select >> RTL Project >> Do not specify sources at this time>> Next)* *(Select >> Zedboard >> Next)*

A screenshot of a computer

Description automatically generated

**2.2 Design Sources Overview**

The design consists of the **feedforward\_nn.v** file for the network and the **tb\_feedforward\_nn.v** file for the testbench. Both files were added as design and simulation sources, respectively.

A screenshot of a computer

Description automatically generated

**Example: how to add files >> (feedforward\_nn.v) (design source)**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

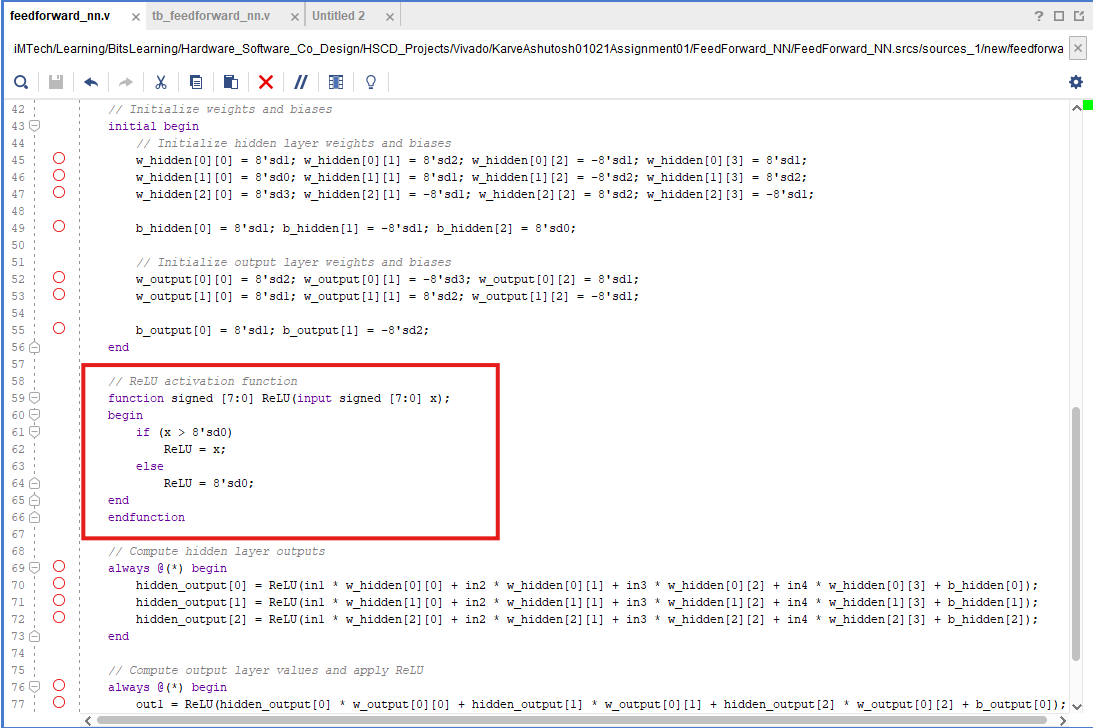
Description automatically generated

**3. Network Architecture**

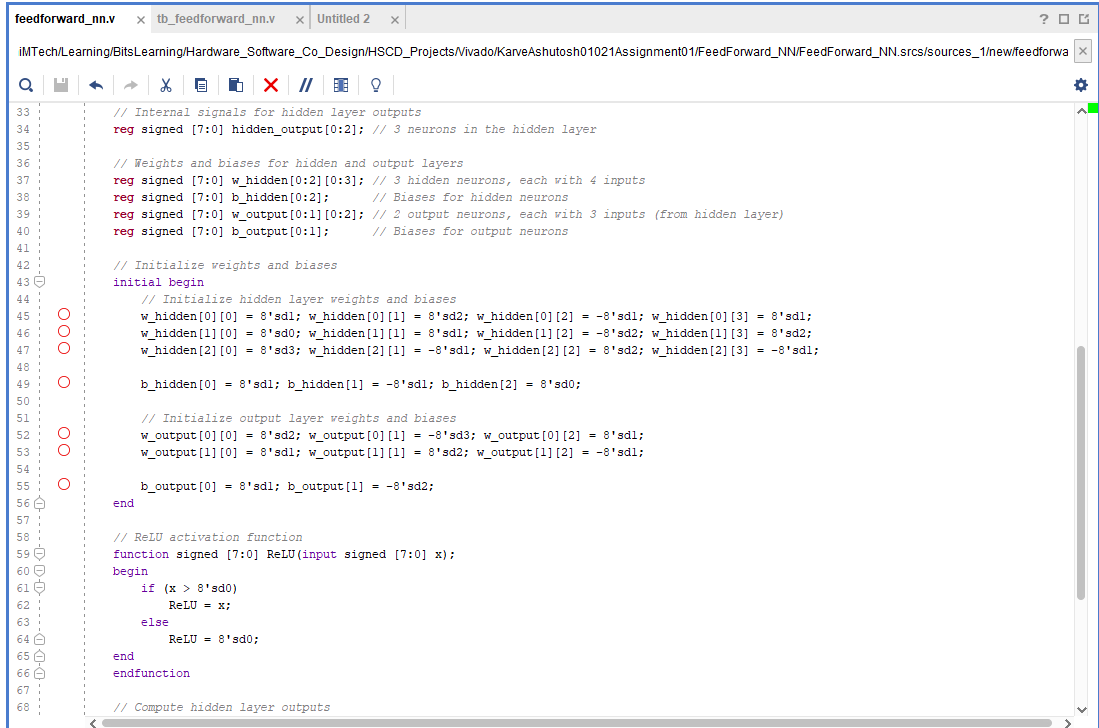
The network was implemented with the following layers:

* **Input Layer**: 4 inputs, each 8-bit signed.
* **Hidden Layer**: 3 neurons with ReLU activation.
* **Output Layer**: 2 outputs, each 8-bit signed.

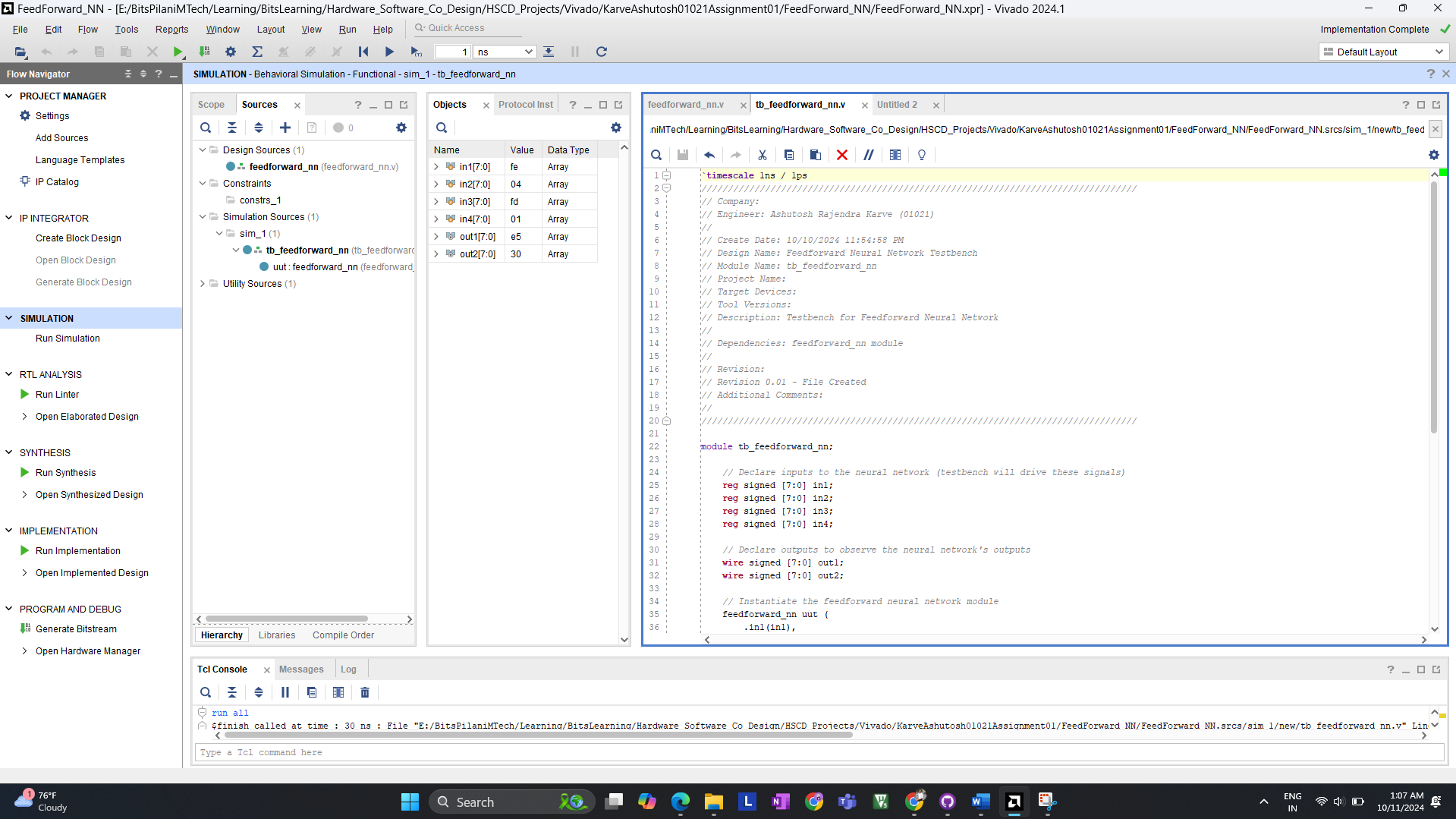
**3.1 ReLU Activation Function**

The ReLU function ensures that the output of a neuron is non-negative, and is defined as follows: 

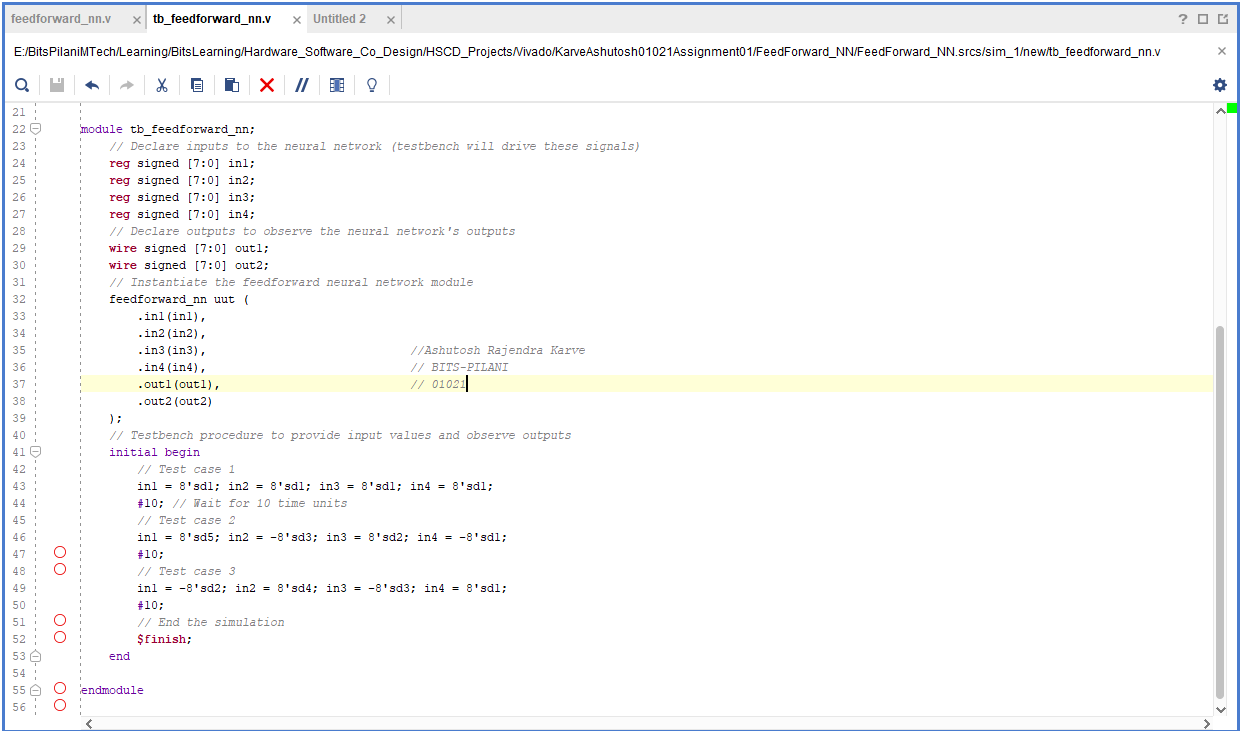
**3.2 Weights and Bias Initialization**

The weights and biases for both the hidden layer and the output layer were initialized as signed 8-bit values.  


**4. Testbench Design**

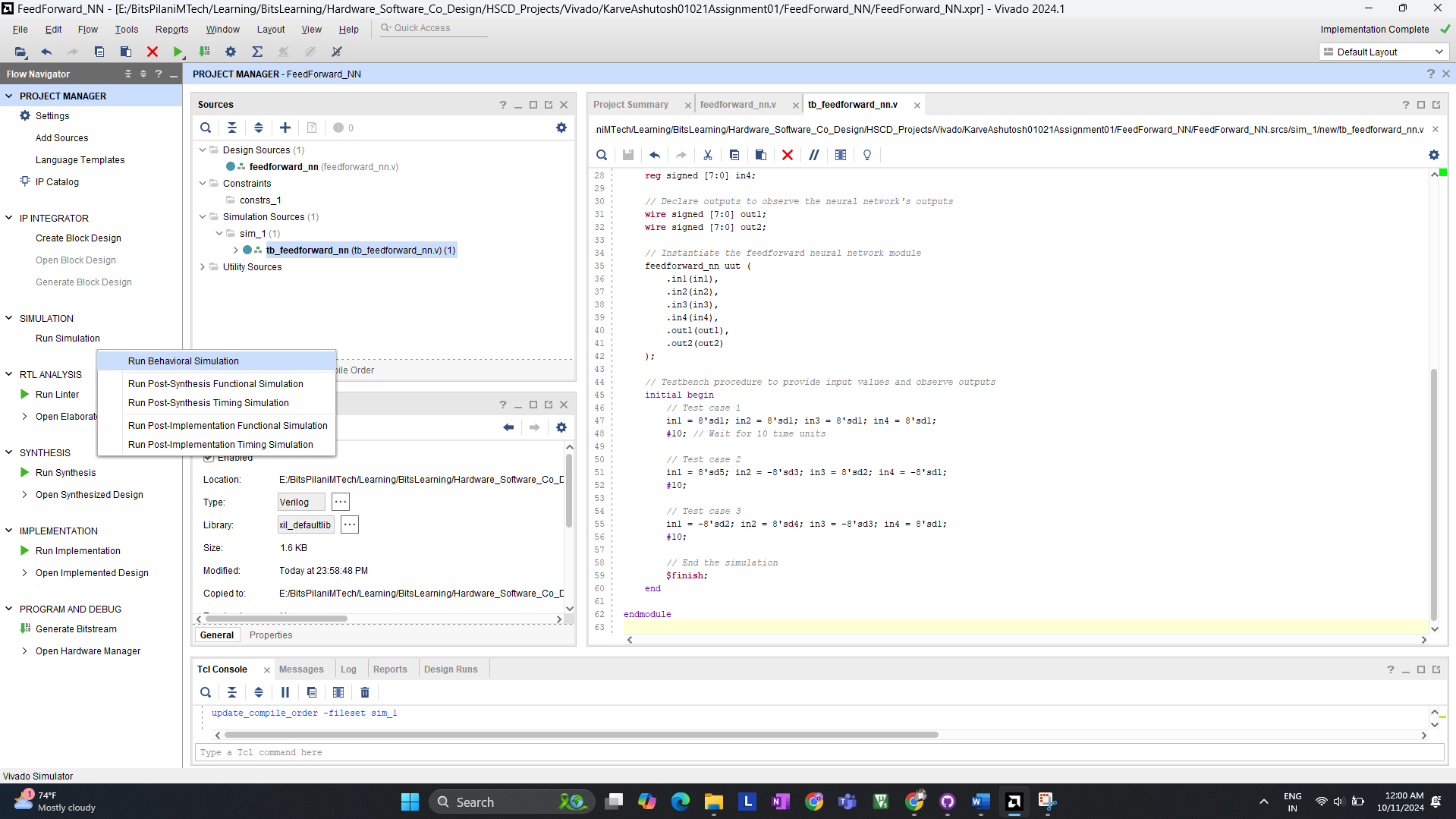
The testbench was created to validate the functionality of the feedforward neural network. The testbench provides input values to the network and captures the outputs. **Three test cases** were used to assess the behavior of the network.  


**4.1 Testbench Code**



**5. Simulation Results**

**5.1 Running the Simulation**

The testbench was run using Vivado's behavioral simulation. The waveform viewer shows the changes in inputs and corresponding outputs of the network.  


A screenshot of a computer

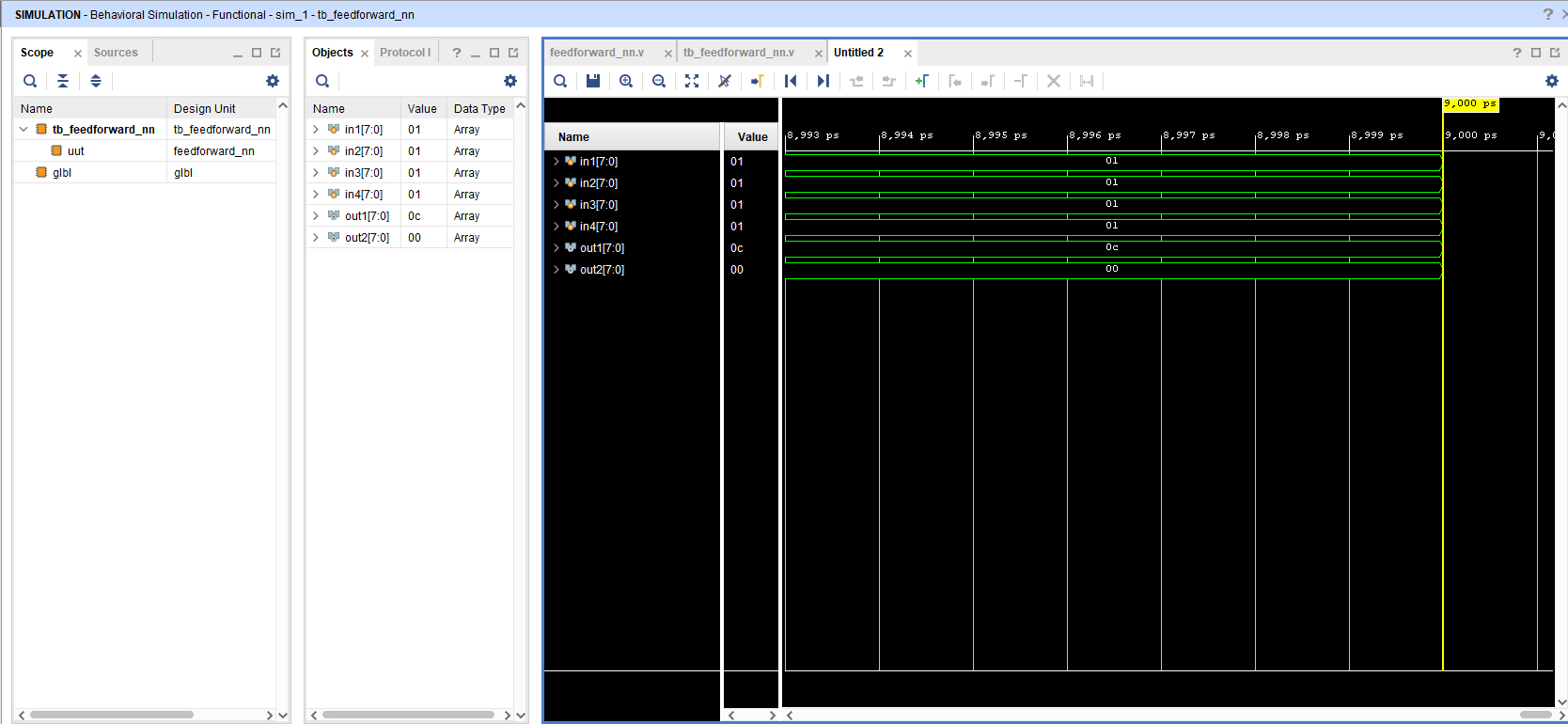
Description automatically generated

**5.2 Test Case 1:**

* Inputs:
  + in1 = 01, in2 = 01, in3 = 01, in4 = 01
* Outputs:
  + out1 = 0c (12 in decimal)
  + out2 = 00 (0 in decimal)

**Explanation:**

* out1 = 0c (12) indicates that the neural network computed a positive result for the first output neuron.
* out2 = 00 is expected, as the network output likely produced a negative result before ReLU was applied, which ReLU correctly converted to 0.

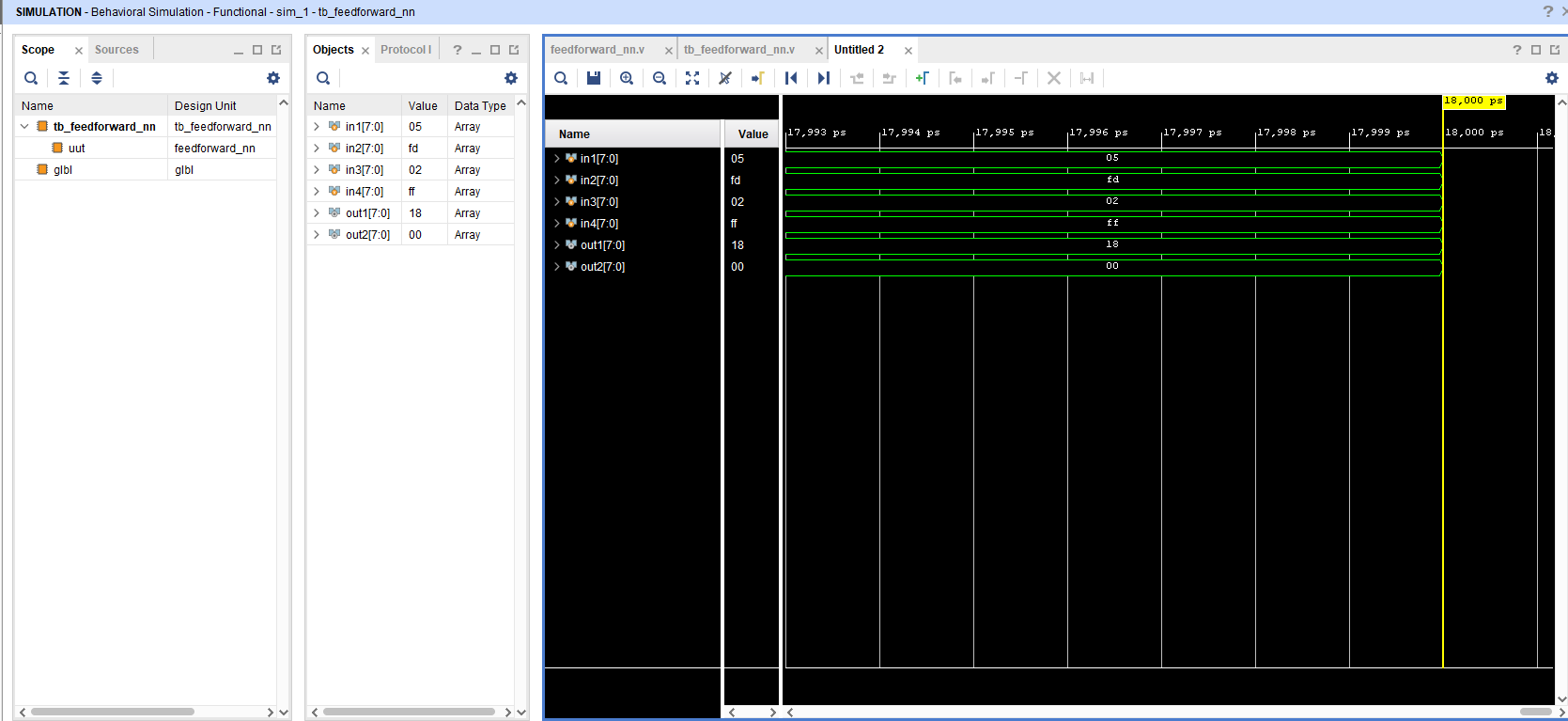
****

**5.3 Test Case 2:**

* Inputs:
  + in1 = 05, in2 = fd (-3 in decimal), in3 = 02, in4 = ff (-1 in decimal)
* Outputs:
  + out1 = 18 (24 in decimal)
  + out2 = 00 (0 in decimal)

**Explanation:**

* out1 = 18 (24) is a positive output, indicating that the input combination resulted in a valid, non-negative result after ReLU.
* out2 = 00 means that the network’s second output neuron computed a negative value before ReLU was applied, and ReLU converted it to 0.

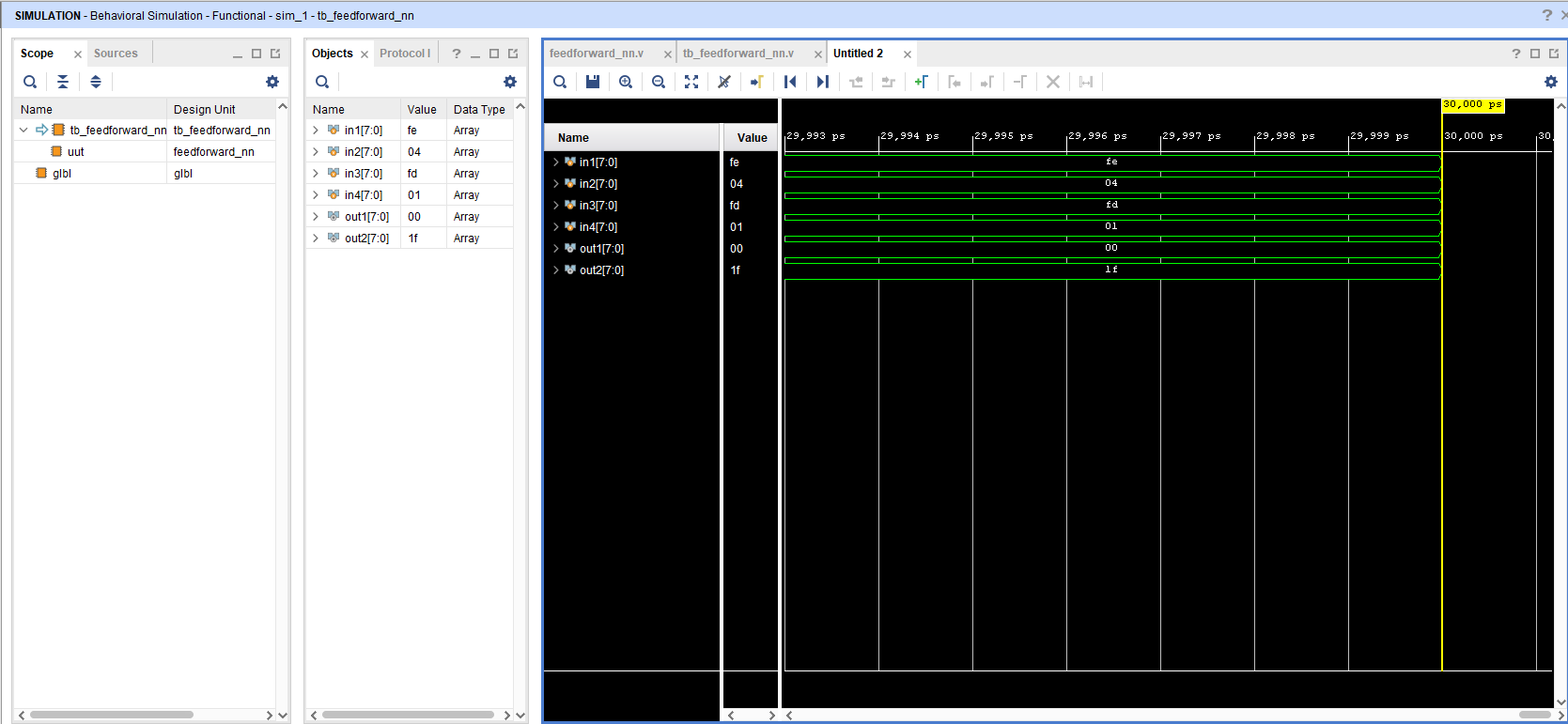
****

**5.4 Test Case 3:**

* Inputs:
  + in1 = fe (-2 in decimal), in2 = 04, in3 = fd (-3 in decimal), in4 = 01
* Outputs:
  + out1 = 00 (0 in decimal)
  + out2 = 1f (31 in decimal)

**Explanation:**

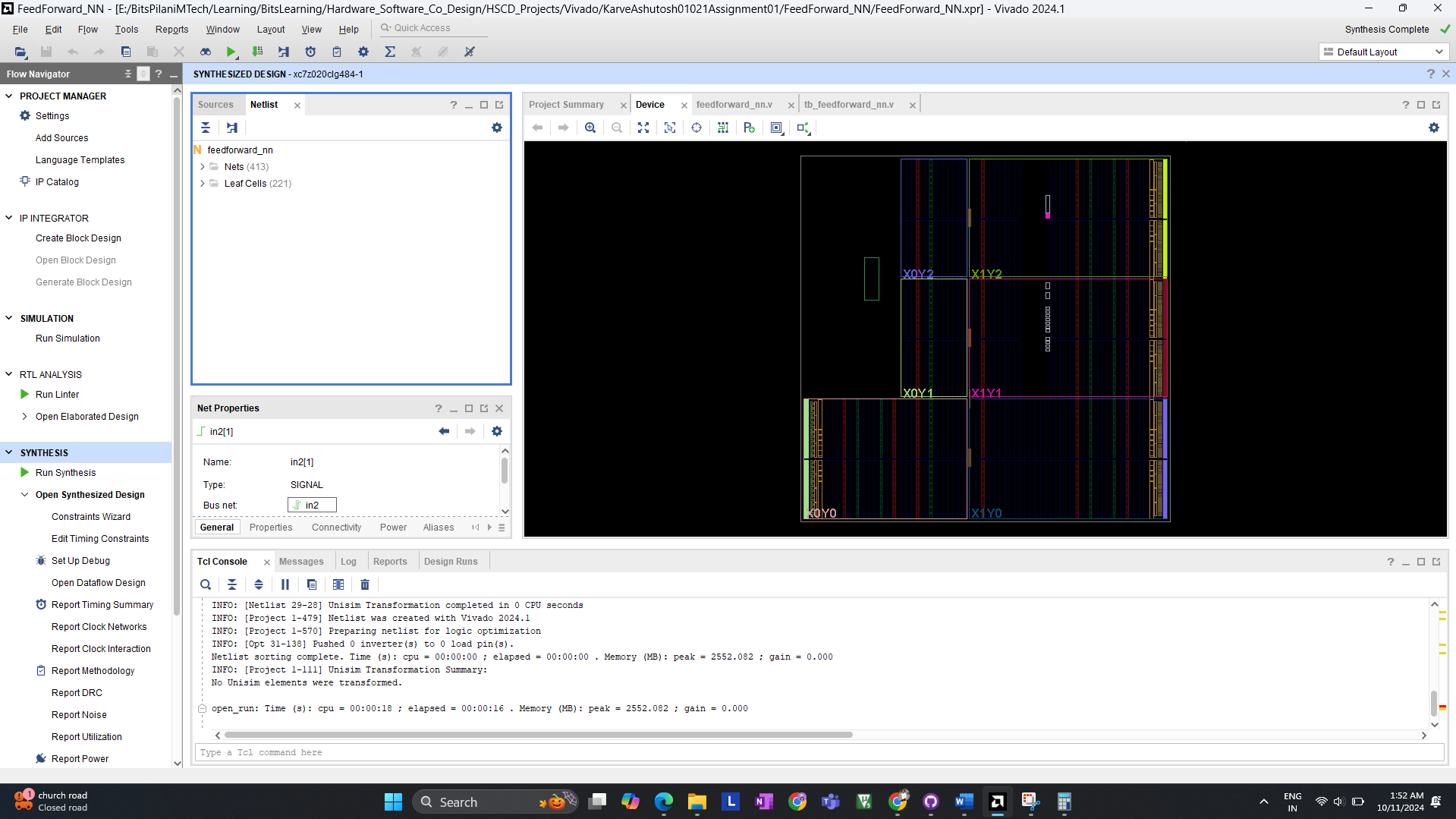
* out1 = 00 shows that the first output neuron computed a negative result before ReLU, which was correctly zeroed out.
* out2 = 1f (31) is a valid positive result from the second output neuron, which was not affected by ReLU since it was non-negative.

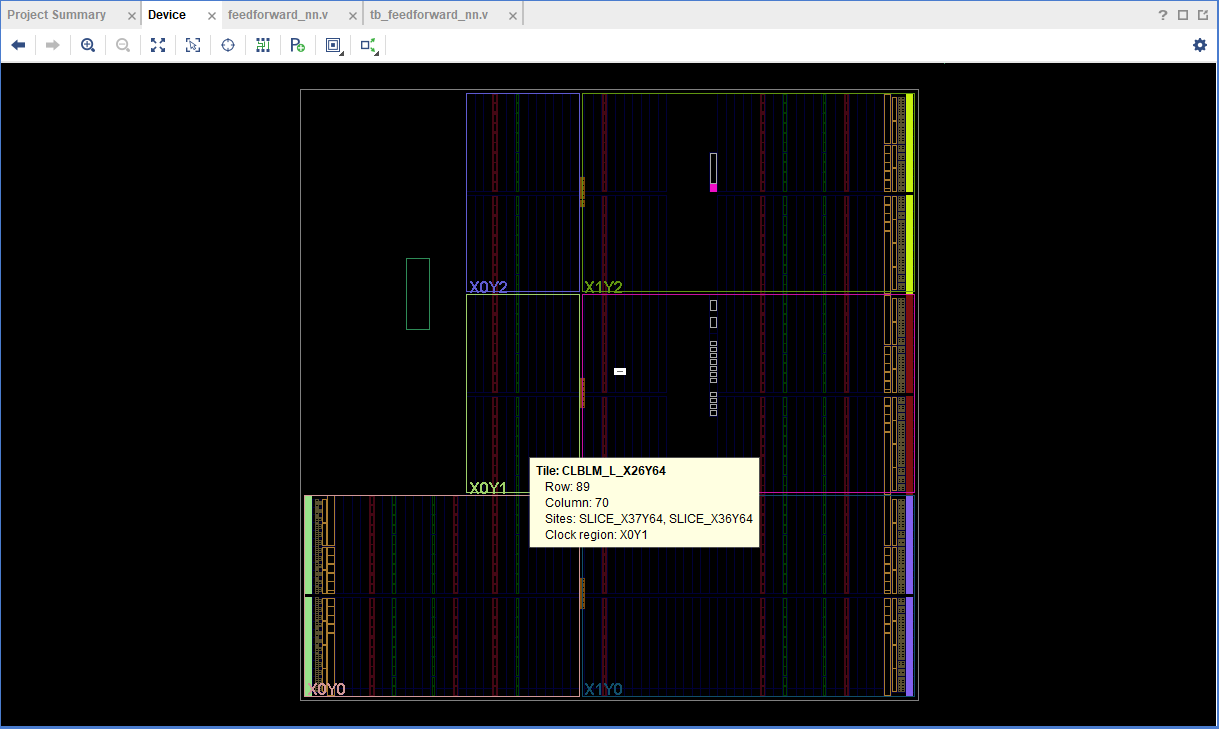


**6. Synthesis and Elaboration**

**6.1 Synthesis Results**

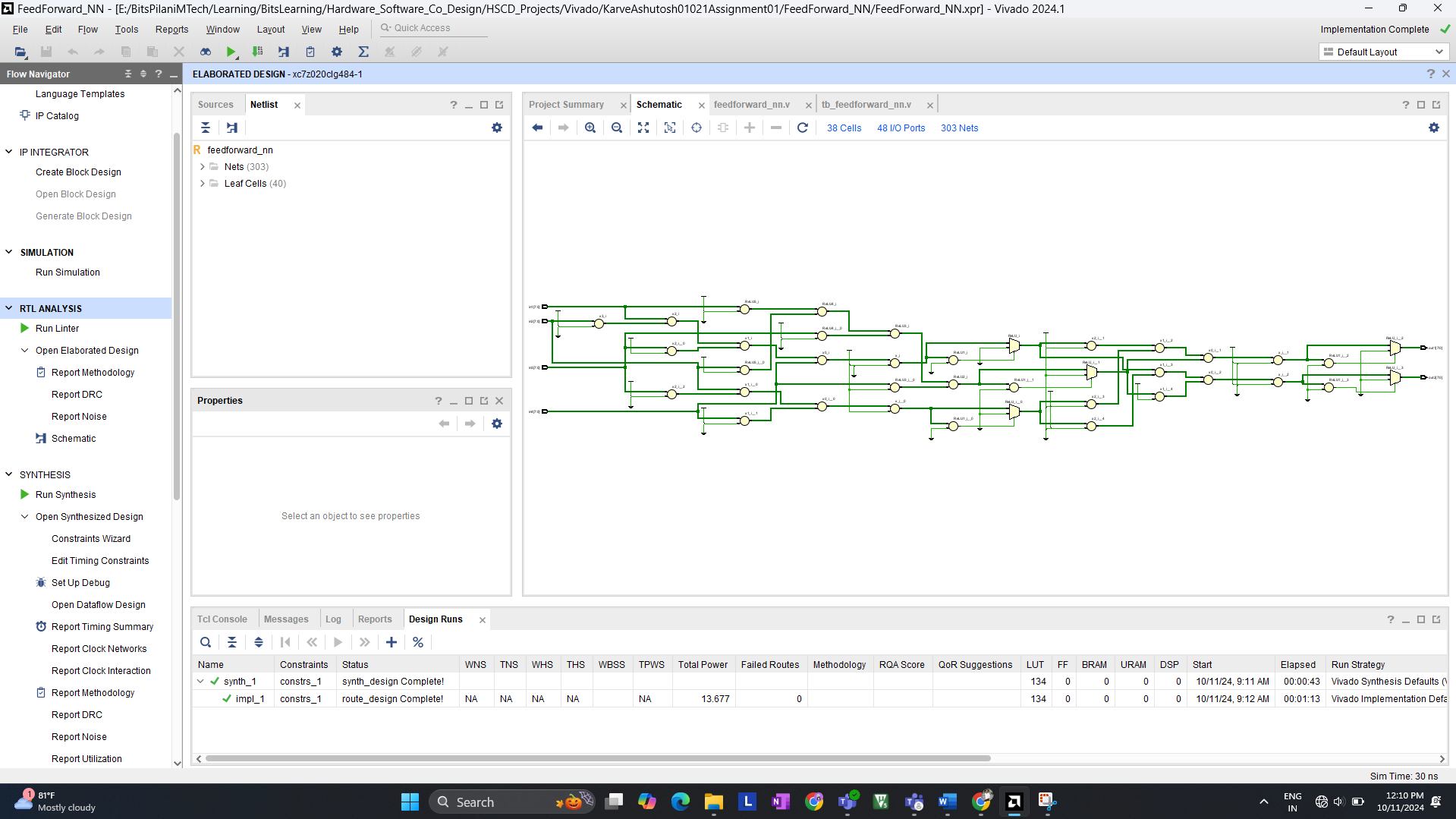
The design was successfully synthesized, and the resulting netlist was generated.

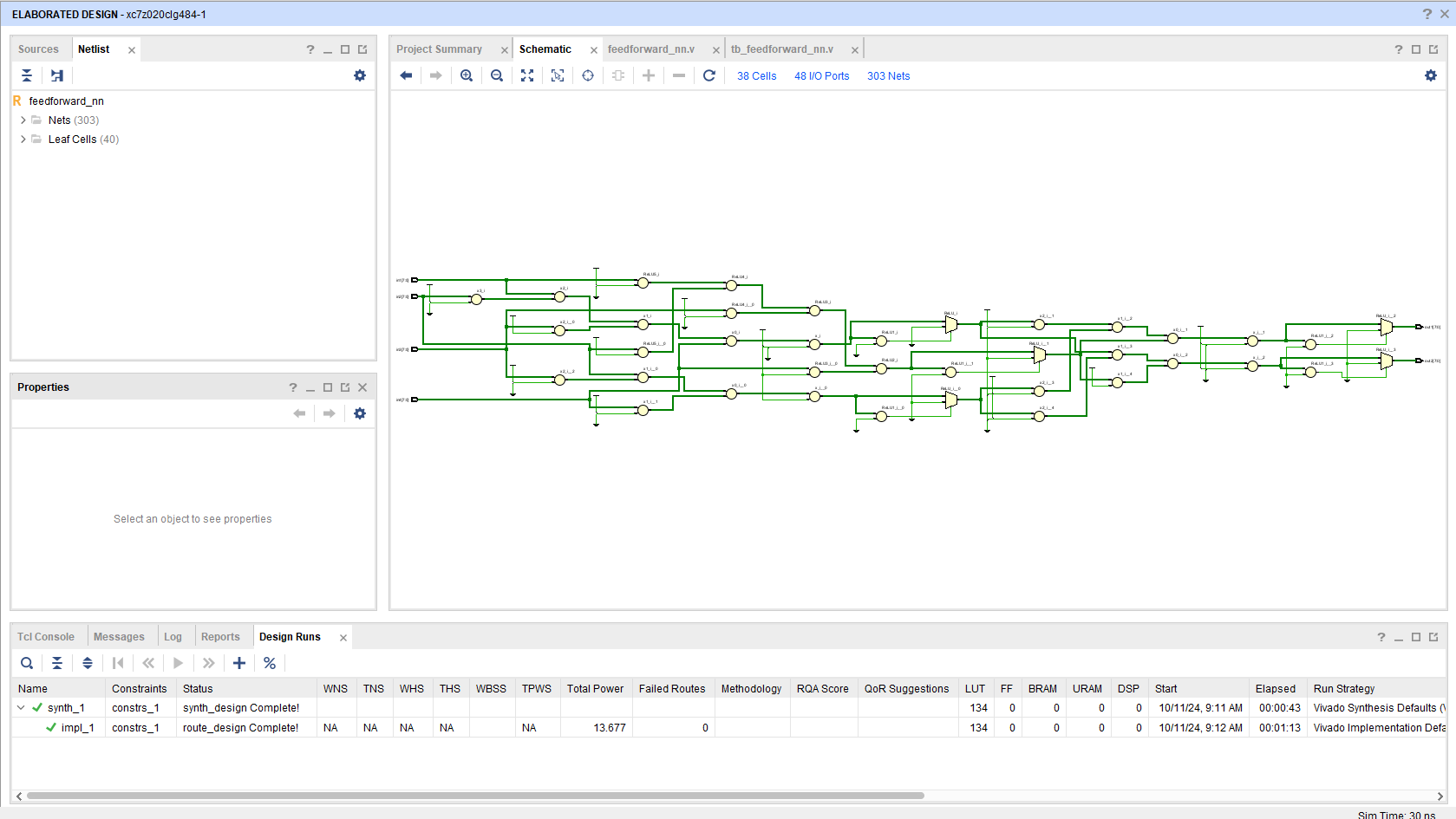
****

****

**6.2 Elaboration Results**

The elaboration process verified that the design was correctly structured.





**7. Conclusion**

This assignment involved the successful implementation and testing of a simple feedforward neural network with ReLU activation using Verilog HDL. The simulation results verified the functionality of the network, confirming that the design behaved as expected for each test case.