**Feedforward Neural Network Design**

**(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**

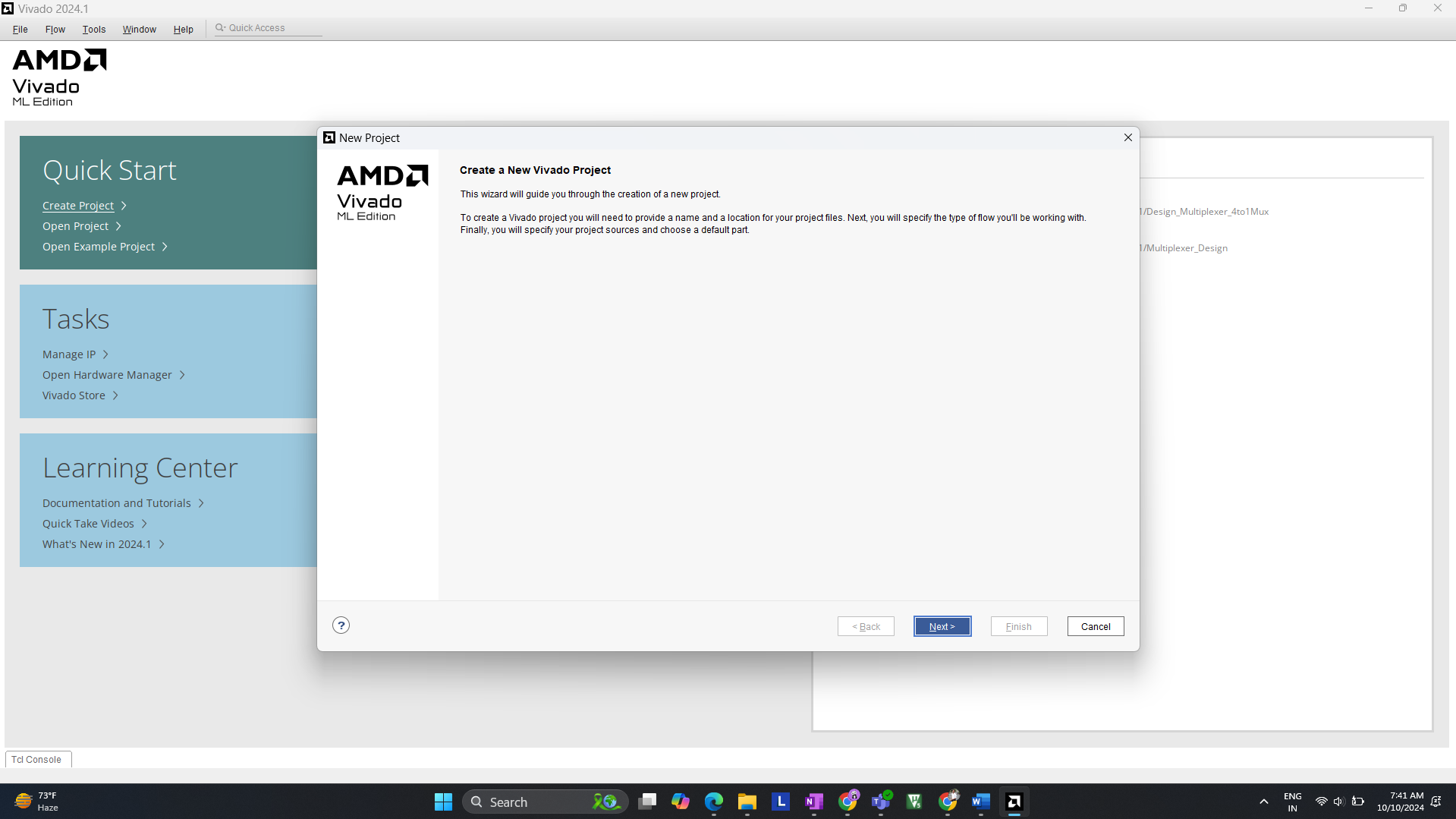
Place **: Pune, Maharashtra**

**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.

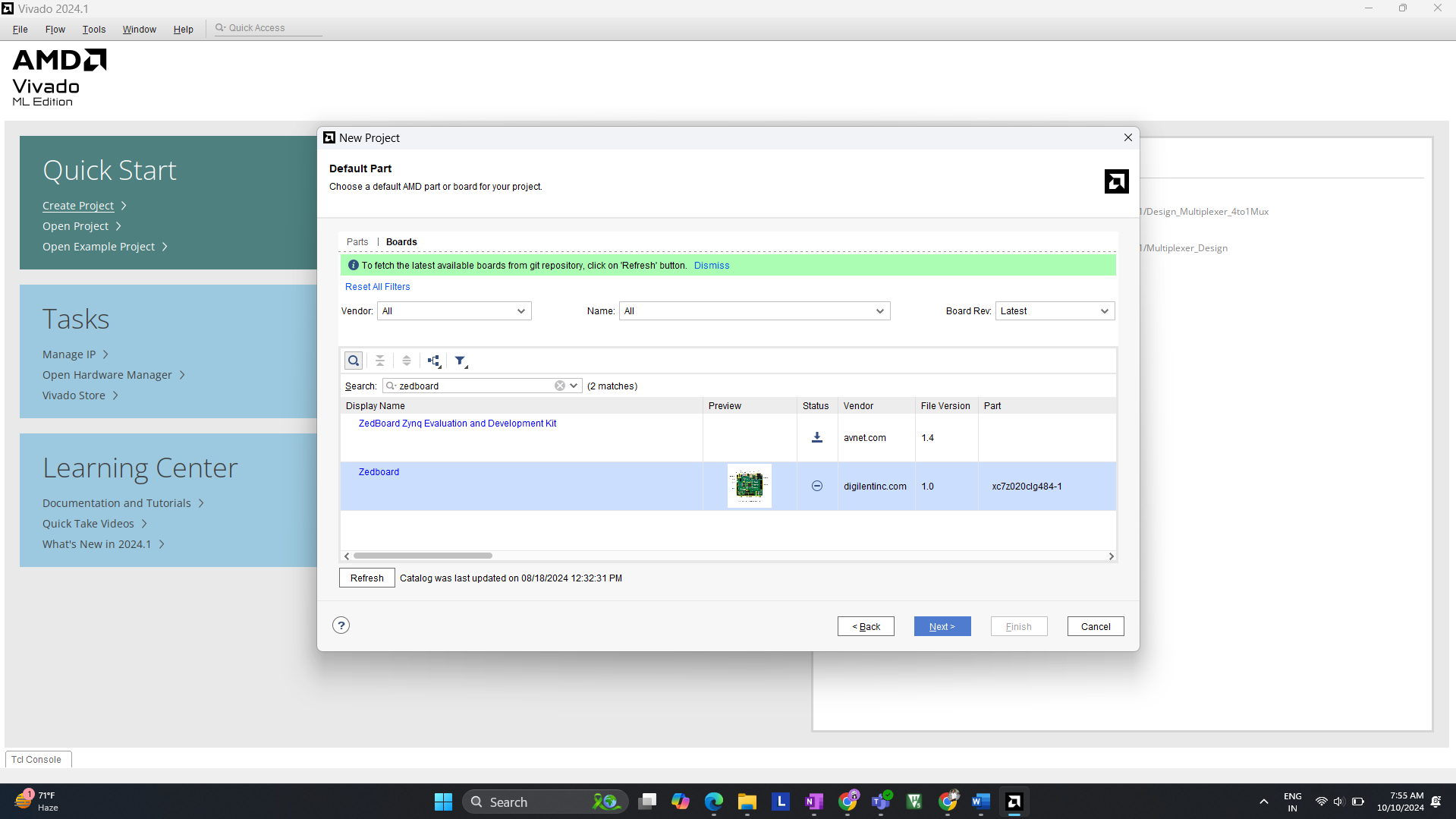
**2. Vivado Project Setup**

**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:

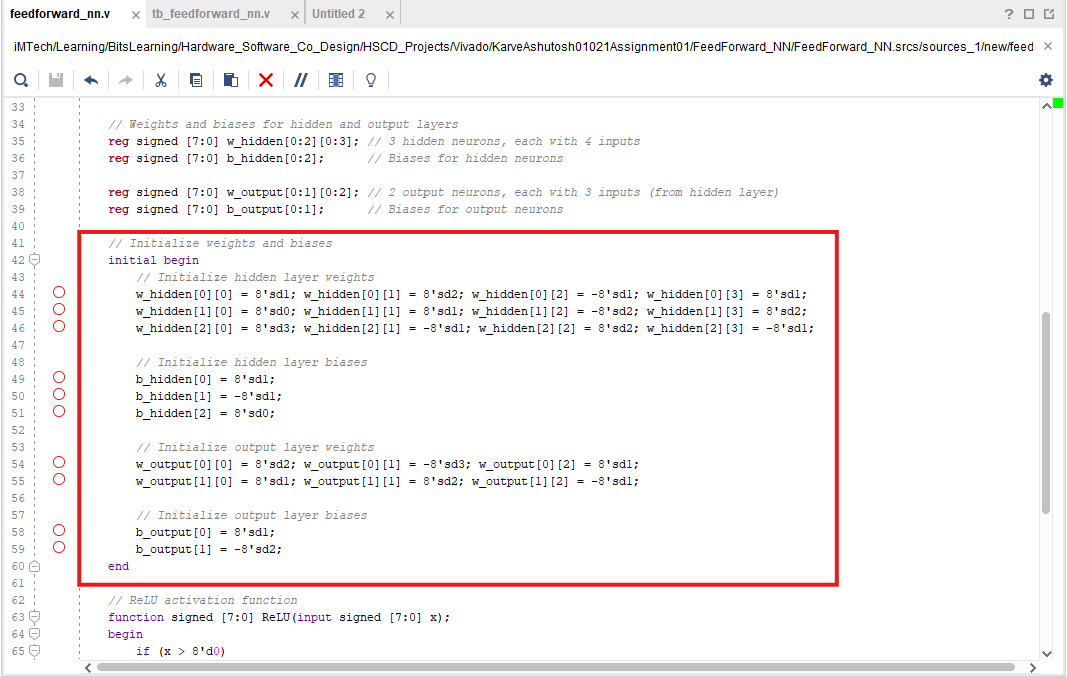
* **Input Layer**: 4 inputs, each 8-bit signed.
* **Hidden Layer**: 3 neurons with ReLU activation.
* **Output Layer**: 2 outputs, each 8-bit signed.

**3.1 ReLU Activation Function**

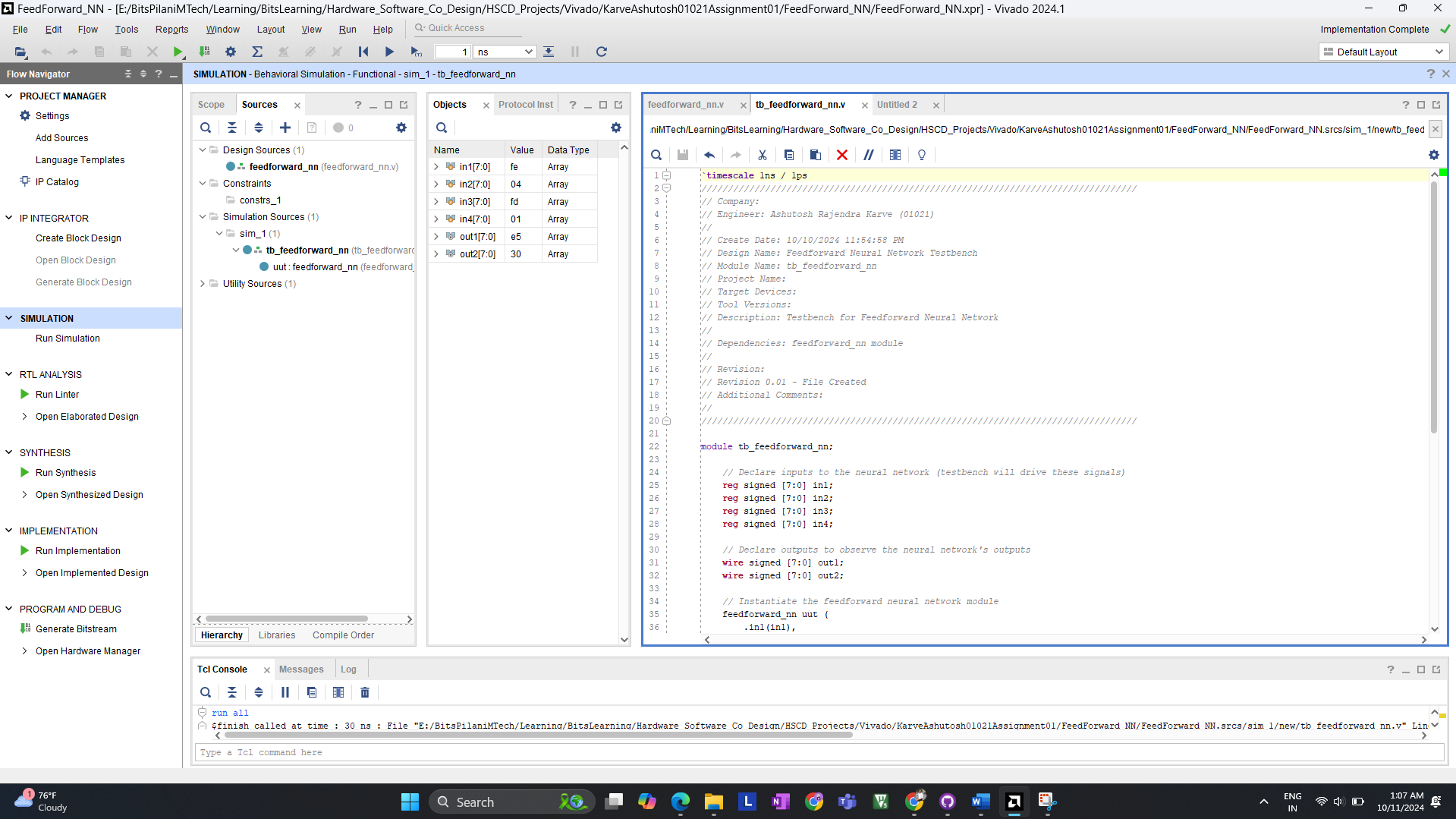
The ReLU function ensures that the output of a neuron is non-negative, and is defined as follows:**

**3.2 Weights and Bias Initialization**

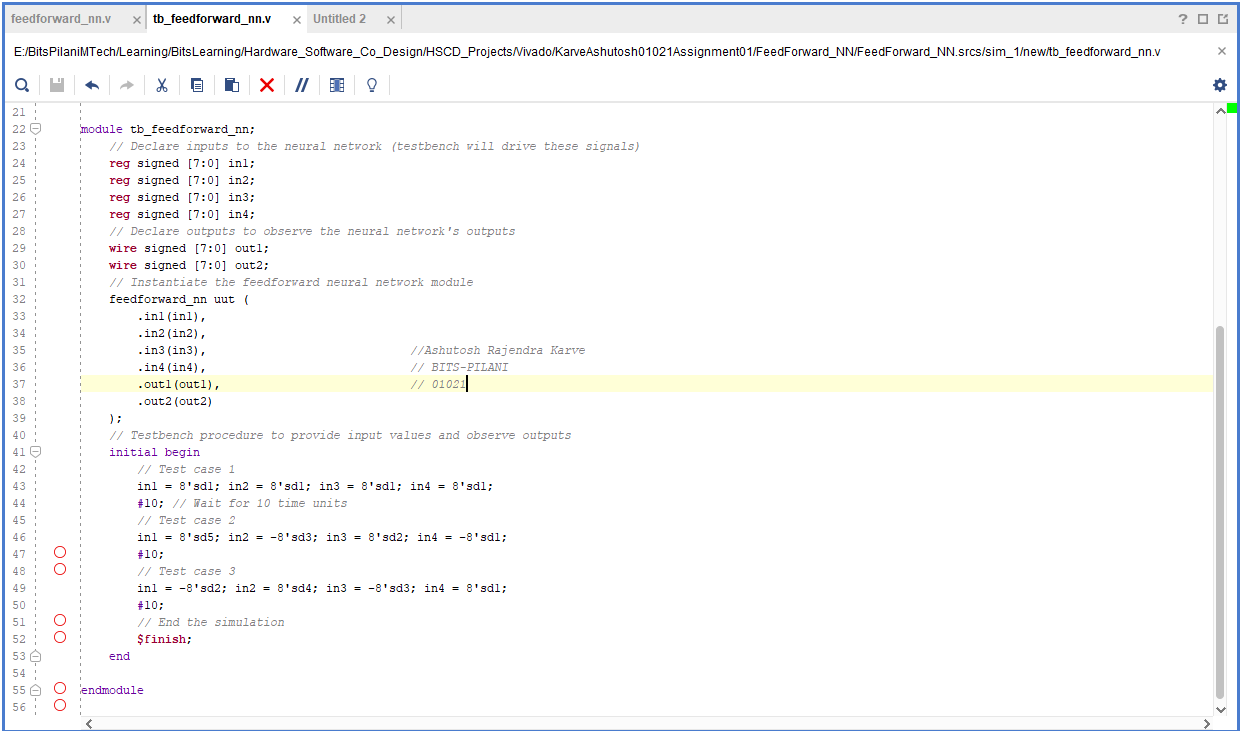
The weights and biases for both the hidden layer and the output layer were initialized as signed 8-bit values.



**4. Testbench Design**

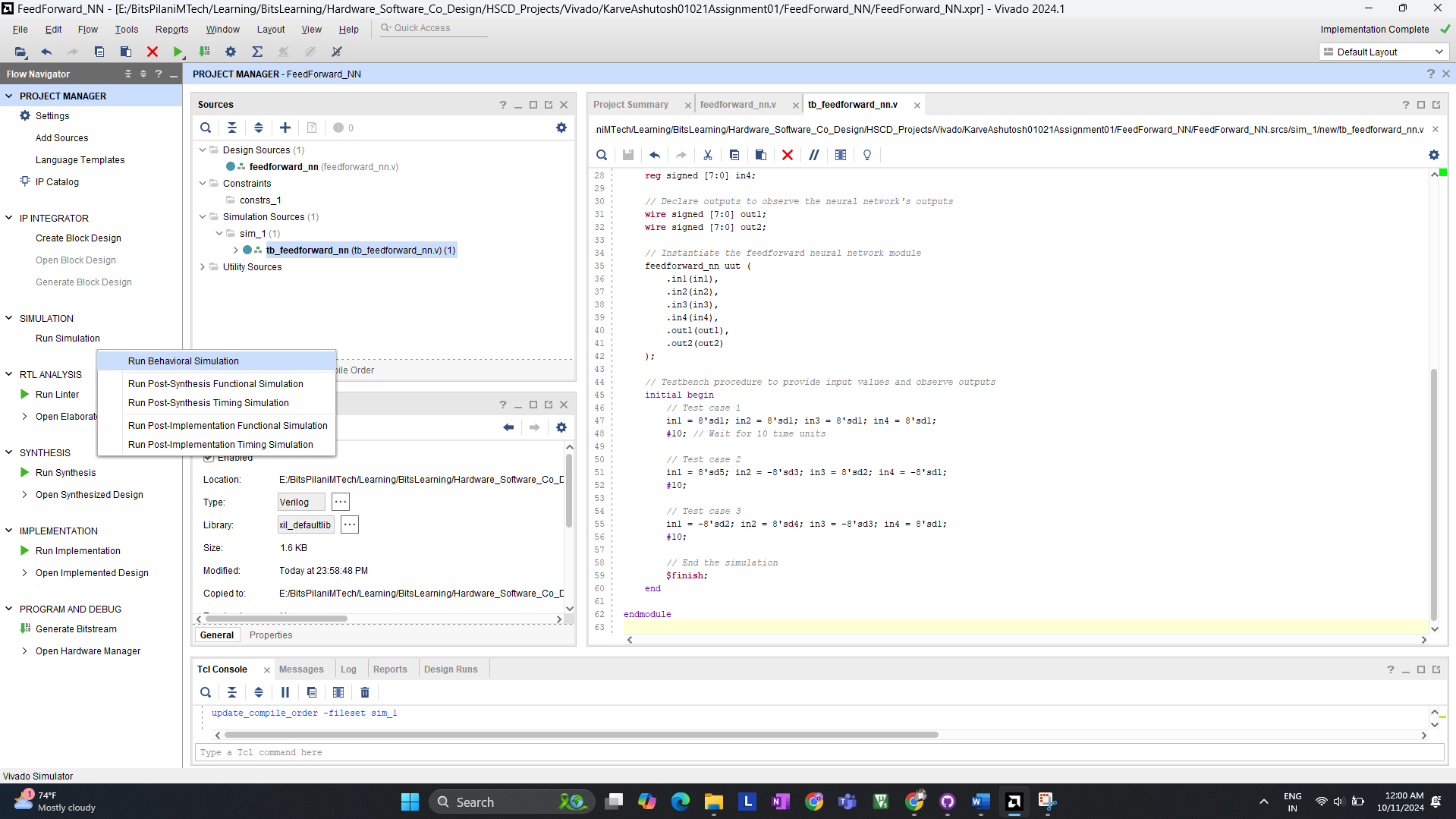
The testbench was created to validate the functionality of the feedforward neural network. The testbench provides input values to the network and captures the outputs. Three test cases were used to assess the behavior of the network.  


**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 = 1, in2 = 1, in3 = 1, in4 = 1

**Outputs**:

* out1 = 0c, out2 = ff

**Explanation**: In Test Case 1, all input values are set to 1. The ReLU activation ensures that the outputs remain non-negative.

**A screenshot of a computer

Description automatically generated**

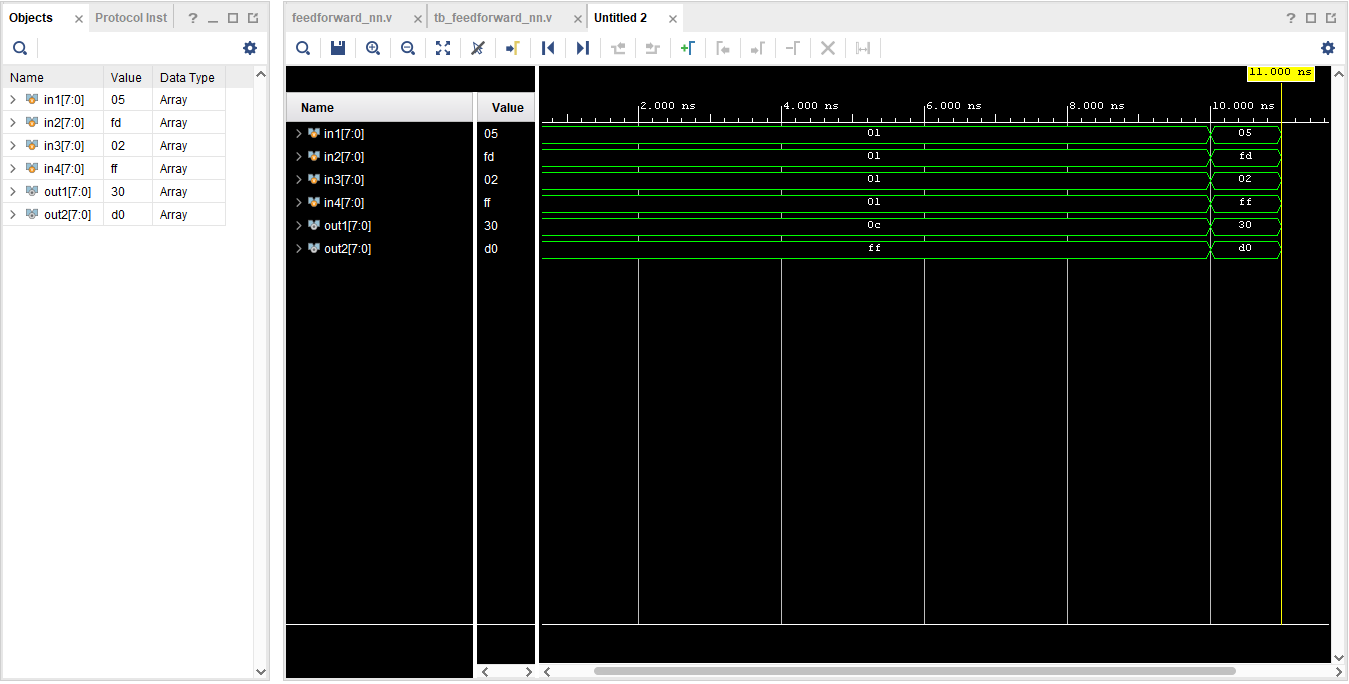
**5.3 Test Case 2**

**Inputs**:

* in1 = 5, in2 = -3, in3 = 2, in4 = -1

**Outputs**:

* out1 = 30, out2 = d0

**Explanation**: This test case includes both positive and negative inputs. Negative weighted sums will be reduced to zero by the ReLU function, resulting in the corresponding outputs.  
****

**A screenshot of a computer

Description automatically generated**

**5.4 Test Case 3**

**Inputs**:

* in1 = -2, in2 = 4, in3 = -3, in4 = 1

**Outputs**:

* out1 = e5, out2 = 30

**Explanation**: This test case tests the network's ability to handle a wider range of positive and negative inputs.

A screenshot of a computer

Description automatically generated

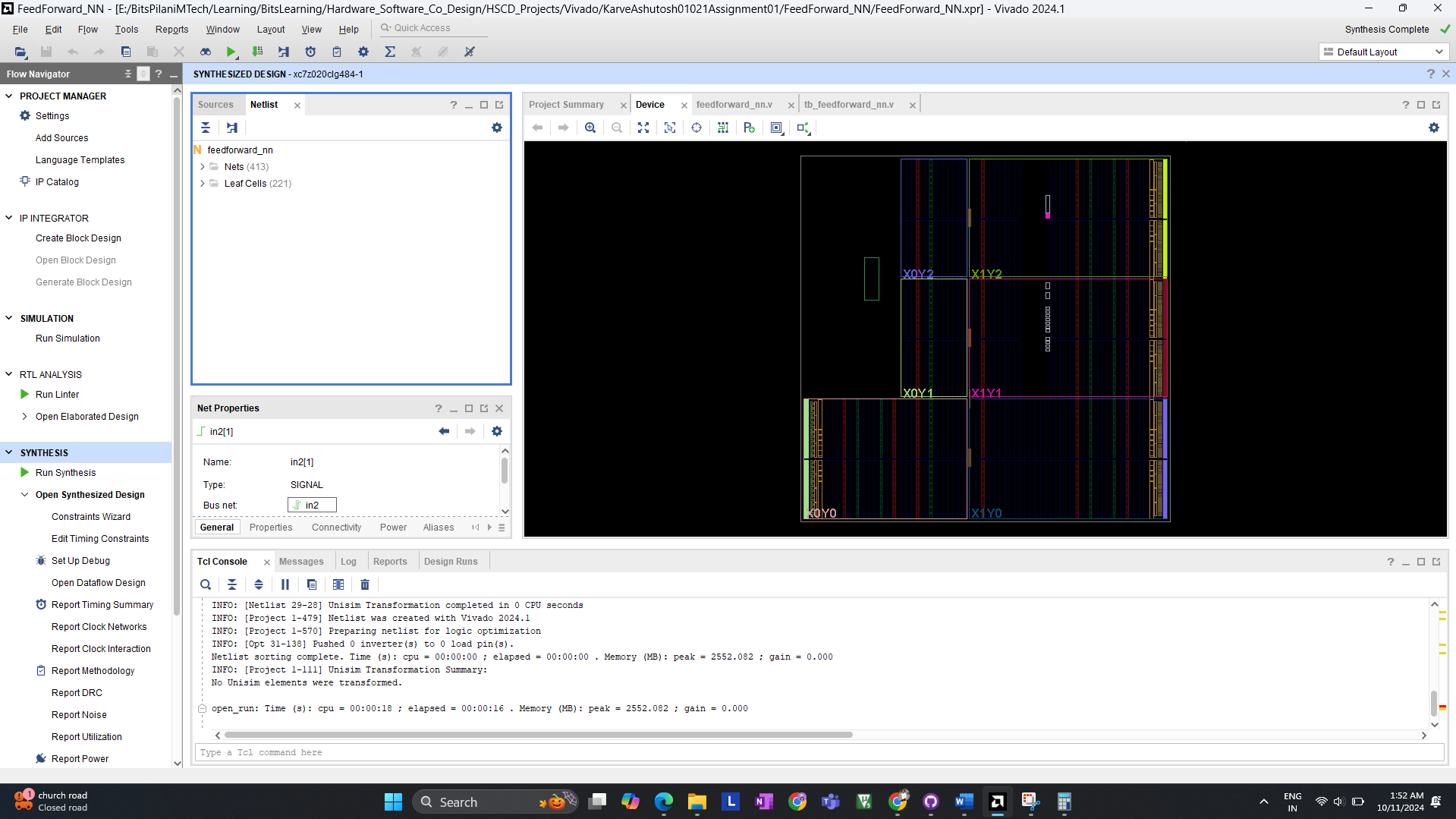
A screenshot of a computer

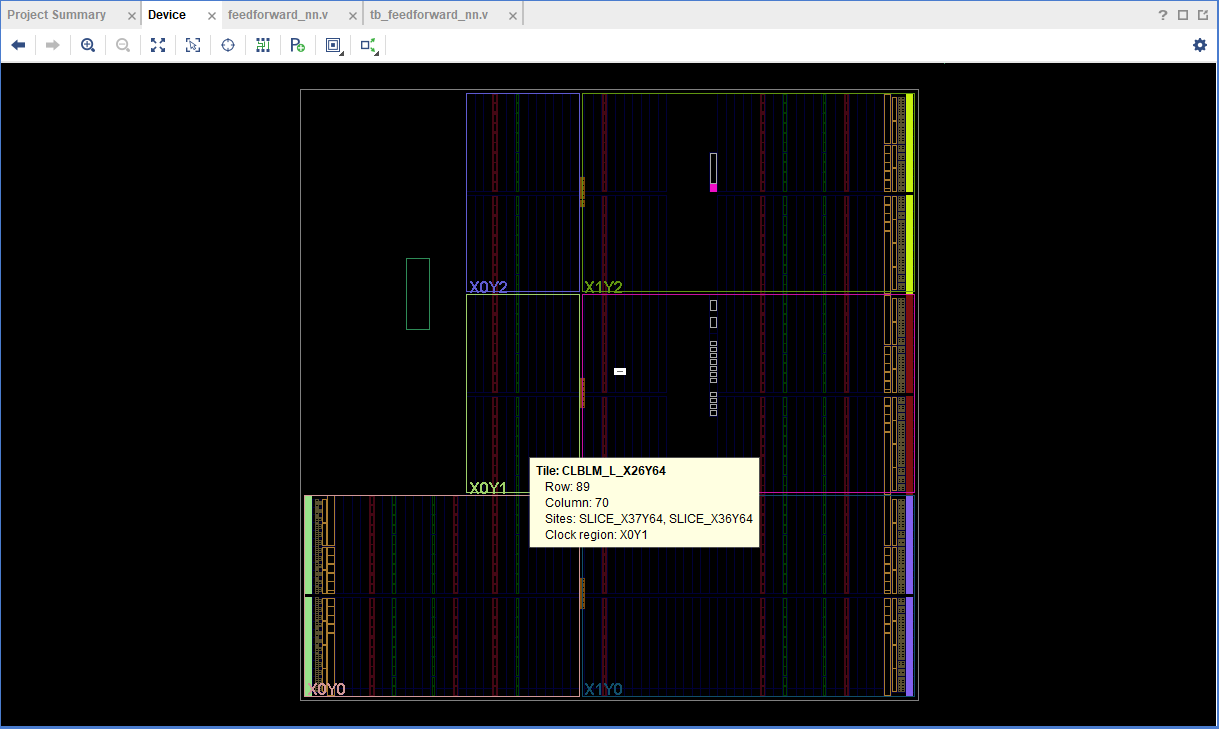
Description automatically generated

**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.

A computer screen shot of a computer

Description automatically generated

A computer screen shot of a computer

Description automatically generated

**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.