Assignment No. 1

HSCD

Submitted by:

Name: Mukund Vishwas Chavan
Student ID: 01011
Student E-mail: 2024ht01011@wilp.bits-pilani.ac.in

Design and implement a simple feedforward neural network using Verilog HDL on an FPGA. The network will have one hidden layer and will perform a basic classification task.

1. Network Architecture:

- Input Layer:

  Inputs: Four 8-bit signed inputs (in1, in2, in3, in4) represent the features of the system.

- Hidden Layer:
  Neurons: Three neurons calculate weighted sums of the inputs. Each neuron takes four weighted inputs and a bias term, then applies a ReLU activation:
  hidden_out = ReLU((w1× in1) + (w2 × in2) + (w3 × in3)+ (w4×in4) + bias) ReLU Activation Function: Implemented as:
  ReLU(x) = max(0,x)

- Output Layer:
  Neurons: Two neurons calculate weighted sums of the hidden layer outputs and apply ReLU:
  output = ReLU ((w1 × hidden_out1) + (w2 × hidden_out2) + (w3 × hidden_out3) + bias)

2. Weights and Biases:

- For simplicity, the neural network uses 8-bit signed integers for weights and biases. This choice allows easy representation and computation while ensuring that the values fit within the operational range of the model.

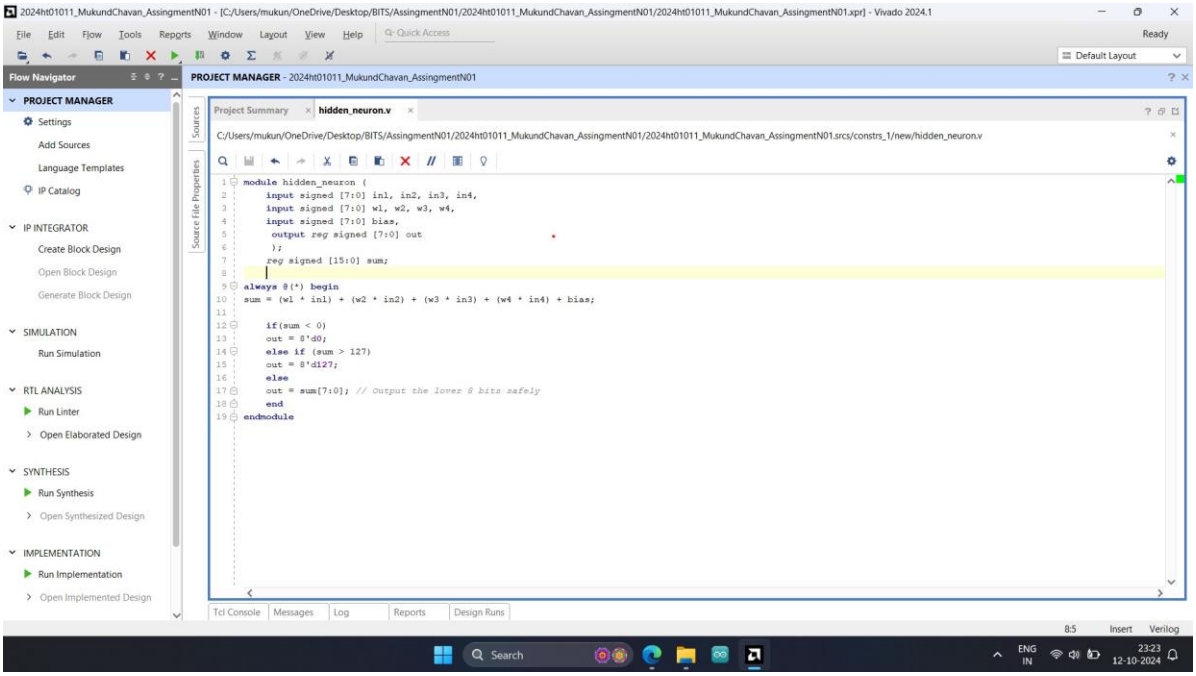
3. Design Implementation and Verilog Code Overview

a) feedforward_nn: The top-level module that integrates the input layer, hidden layer, and output layer.

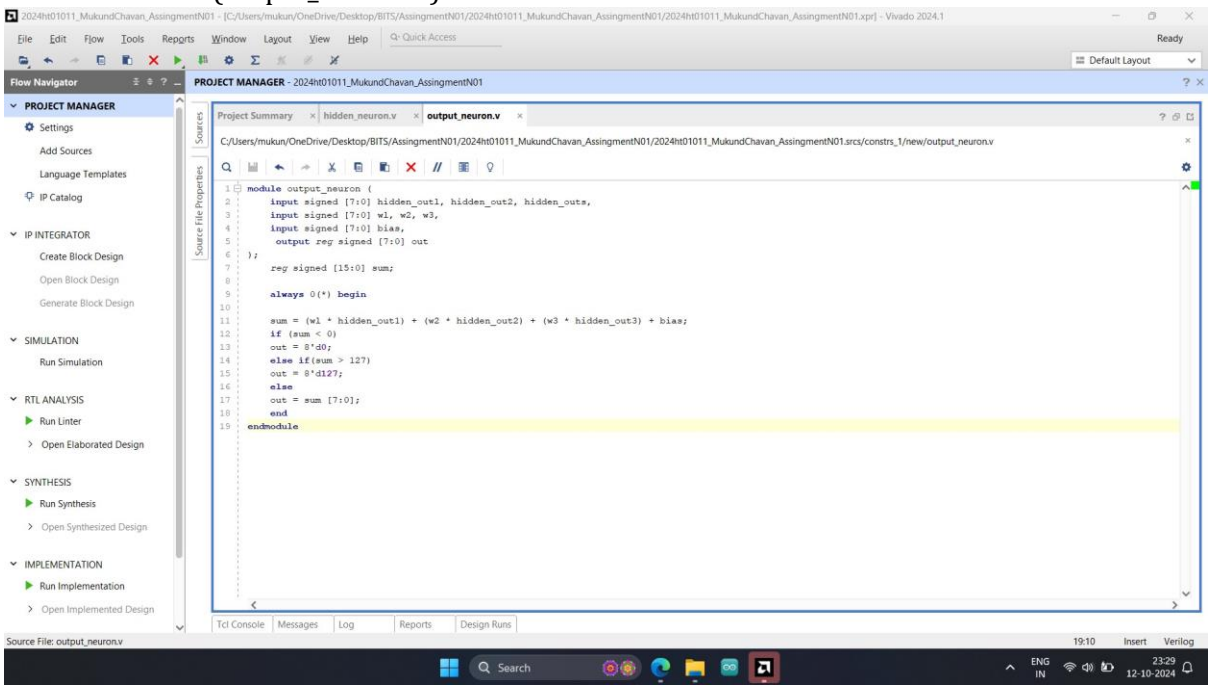
b) hidden_neuron: A module representing each neuron in the hidden layer, implementing the weighted sum and ReLU activation.

c) output_neuron: A module for output neurons, similar to hidden neurons but specifically for combining hidden layer outputs.

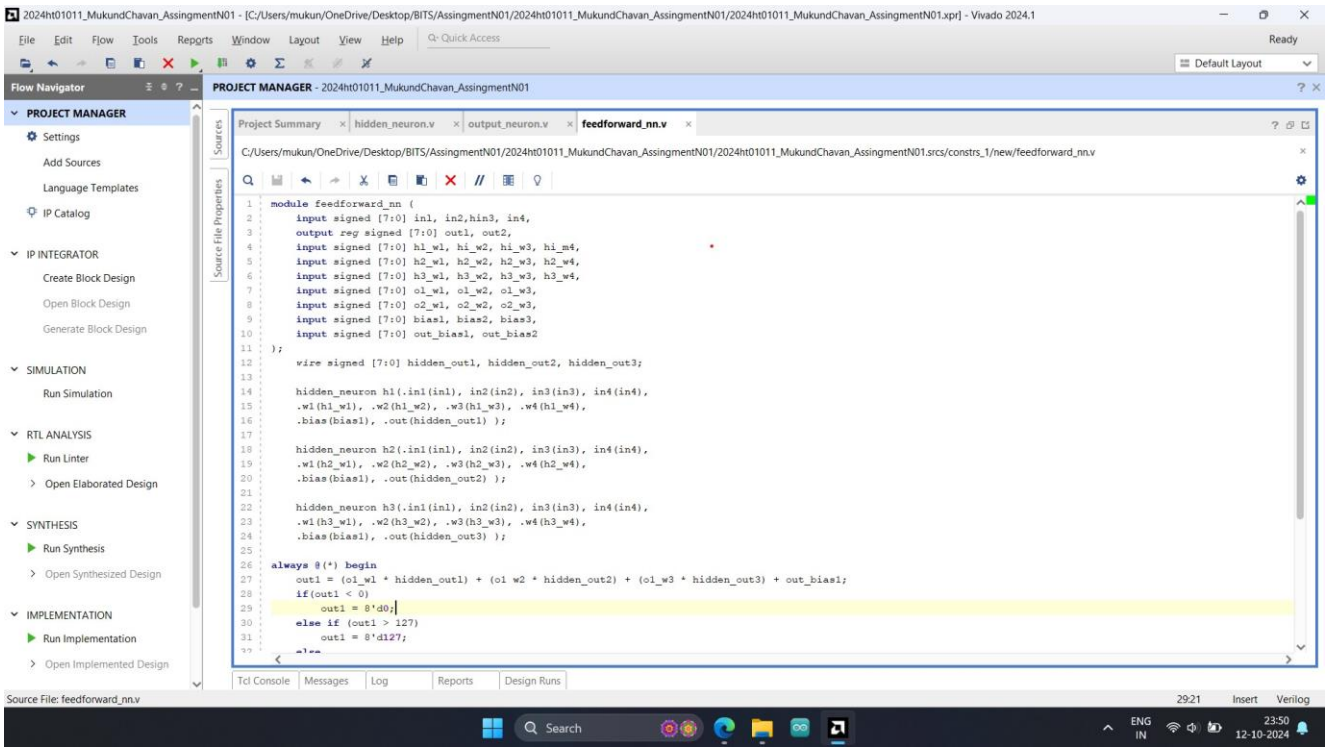
- Hidden Neuron Module (hidden_neuron.v)

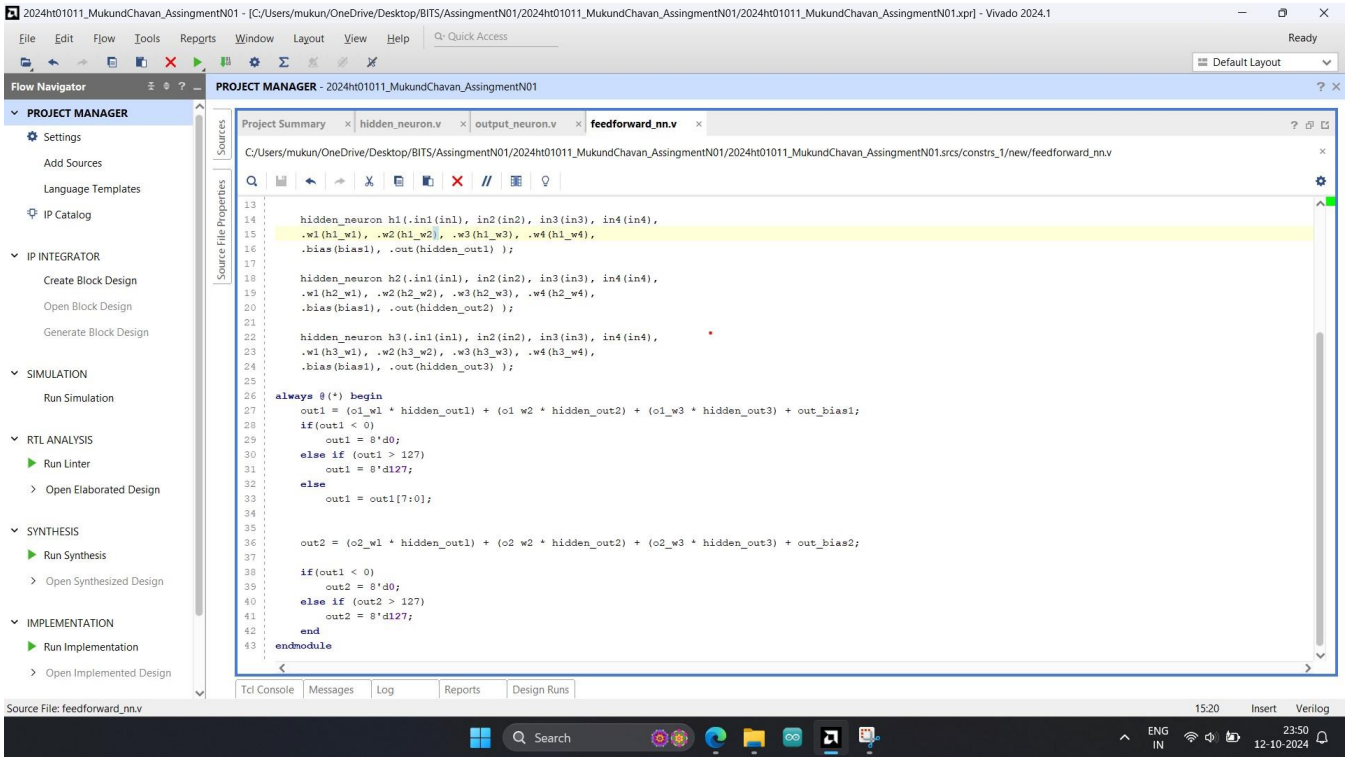


- Output Neuron Module (output_neuron.v)



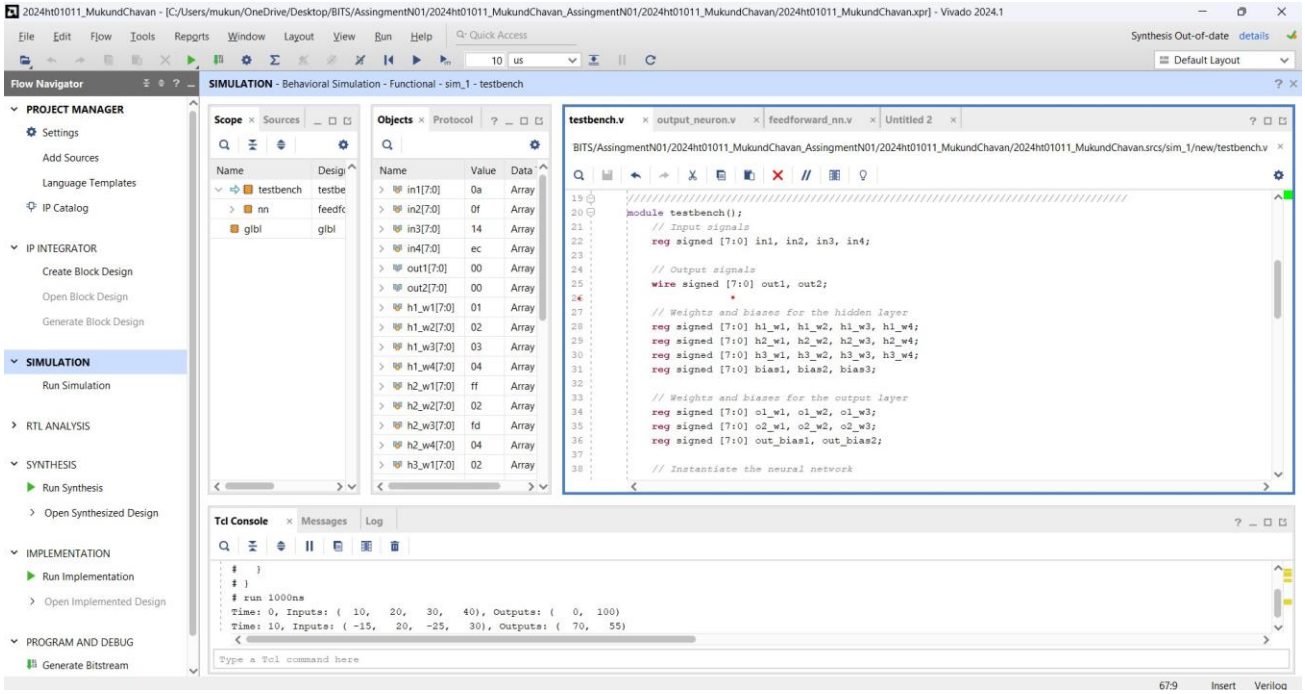
- Top module (feedforward_nm.V)

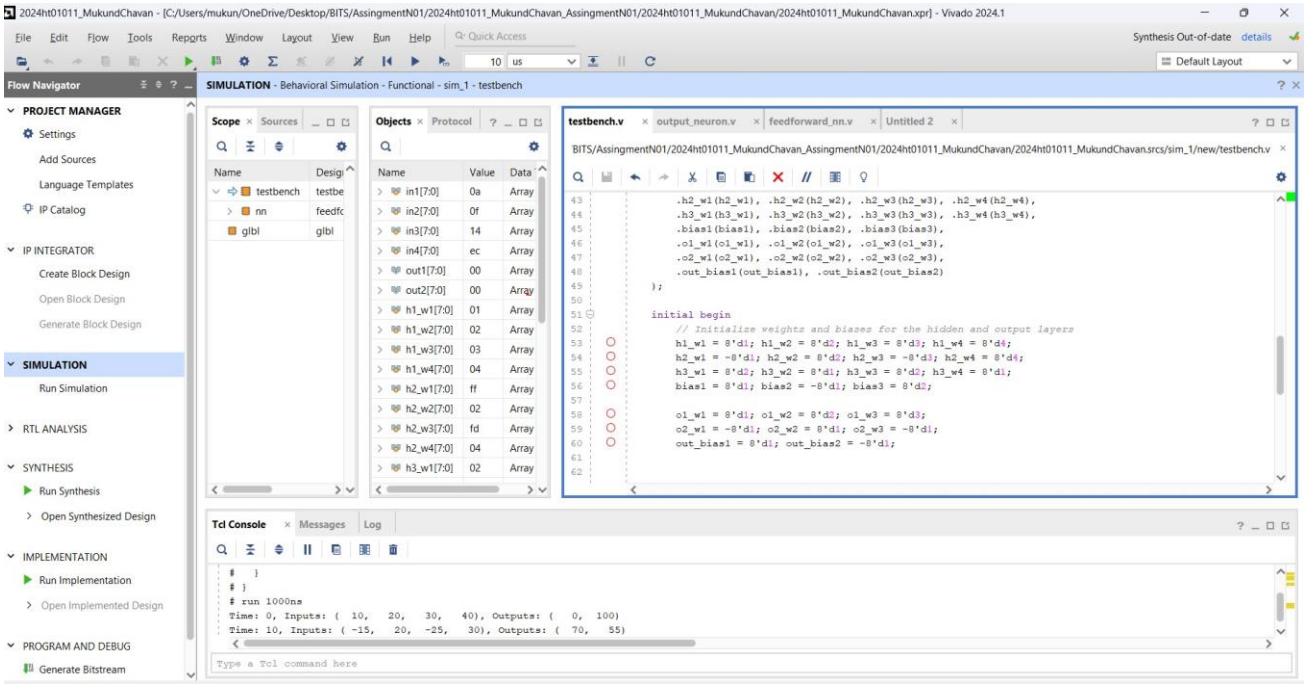


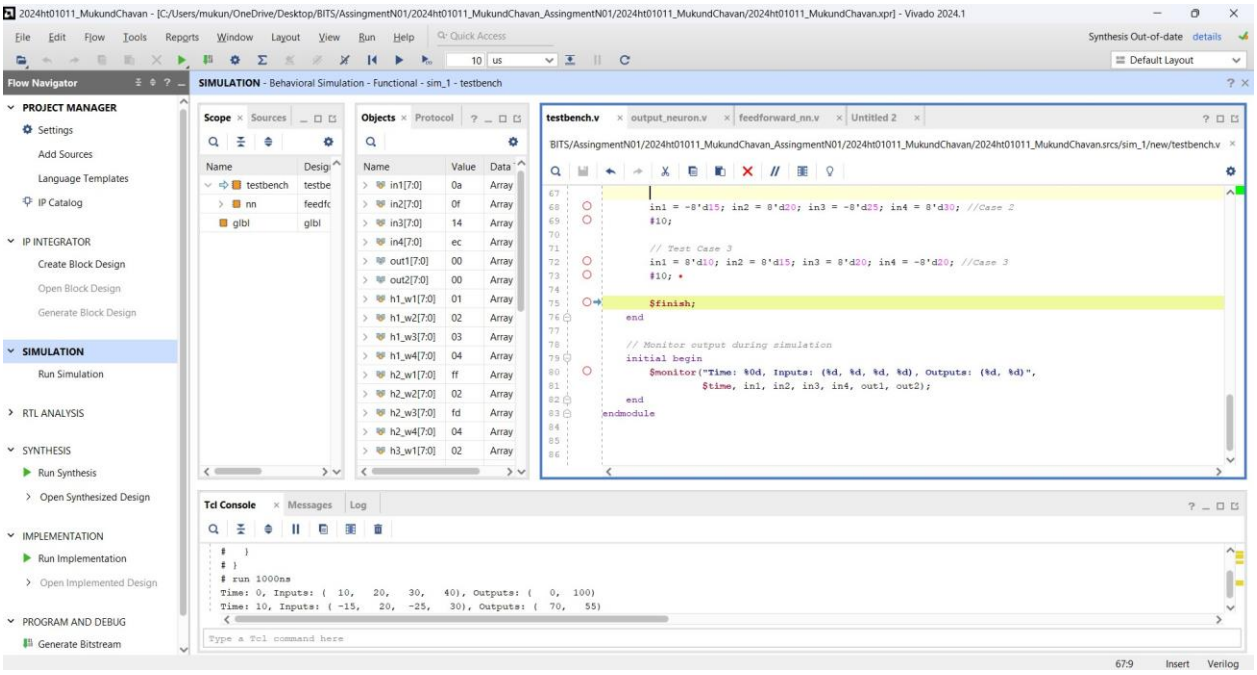


4) Testbench and Verification.

- Testbench






2024ht01011_MukundChavan - [C:/Users/mukun/OneDrive/Desktop/BITS/AssingmentN01/2024ht01011_MukundChavan_AssingmentN01/2024ht01011_MukundChavan/2024ht01011_MukundChavan.xpr] - Vivado 2024.1
File Edit Flow Tools Reports Window Layout View Run Help
Quick Access
Synthesis Out-of-date details
Default Layout
Flow Navigator
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Open Implemented Design
PROGRAM AND DEBUG
Generate Bitstream
SIMULATION - Behavioral Simulation - Functional - sim_1 - testbench
Scope Sources
Objects Protocol
testbench.v output_neuron.v feedforward_nn.v Untitled 2
BITS/AssingmentN01/2024ht01011_MukundChavan_AssingmentN01/2024ht01011_MukundChavan/2024ht01011_MukundChavan.srcs/sim_1/new/testbench.v
in1 = -8'd15; in2 = 8'd20; in3 = -8'd25; in4 = 8'd30; //Case 2
#10;
// Test Case 3
in1 = 8'd10; in2 = 8'd15; in3 = 8'd20; in4 = -8'd20; //Case 3
#10;
$finish;
end
// Monitor output during simulation
initial begin
$monitor("Time: %0d, Inputs: (%d, %d, %d, %d), Outputs: (%d, %d)",
$time, in1, in2, in3, in4, out1, out2);
end
endmodule
Tcl Console Messages Log
}
}
run 1000ns
Time: 0, Inputs: (10, 20, 30, 40), Outputs: (0, 100)
Time: 10, Inputs: (-15, 20, -25, 30), Outputs: (70, 55)
Type a Tcl command here
67:9 Insert Verilog

- Simulation Results

 All Three cases