**Assignment 4 - FPGA Implementation**

Submission – R. Mewan Chandira

Functionality of the mvm\_uart\_system RTL and the python script

Mvm\_uart\_system

This system is a wrapper for serial communication modules and computational modules. For serial communication two modules uart\_rx and uart\_tx are designed where uart\_rx convert input serial data to parallel data while uart\_tx convert output parallel data to serial data for sending. The main computation block is a wrapper named axis\_matvec\_mul with two modules named matvec\_mul and skid\_buffer with a shifter. skid\_buffer handles axi stream signals and matvec\_mul handles the matrix multiplication, the intended operation of the design.

Python script

Text

Description automatically generatedSince the communication is serial between computer and Zybo, the following code initatie the serial connection.

The script generate two matrices (k,x) with random integers of range ±128 such that they can undergo vector multiplication. Each element in these arrays are written in 1 byte. The result of the vector multiplication is stored in another array (y\_exp) with 4 bytes elements.

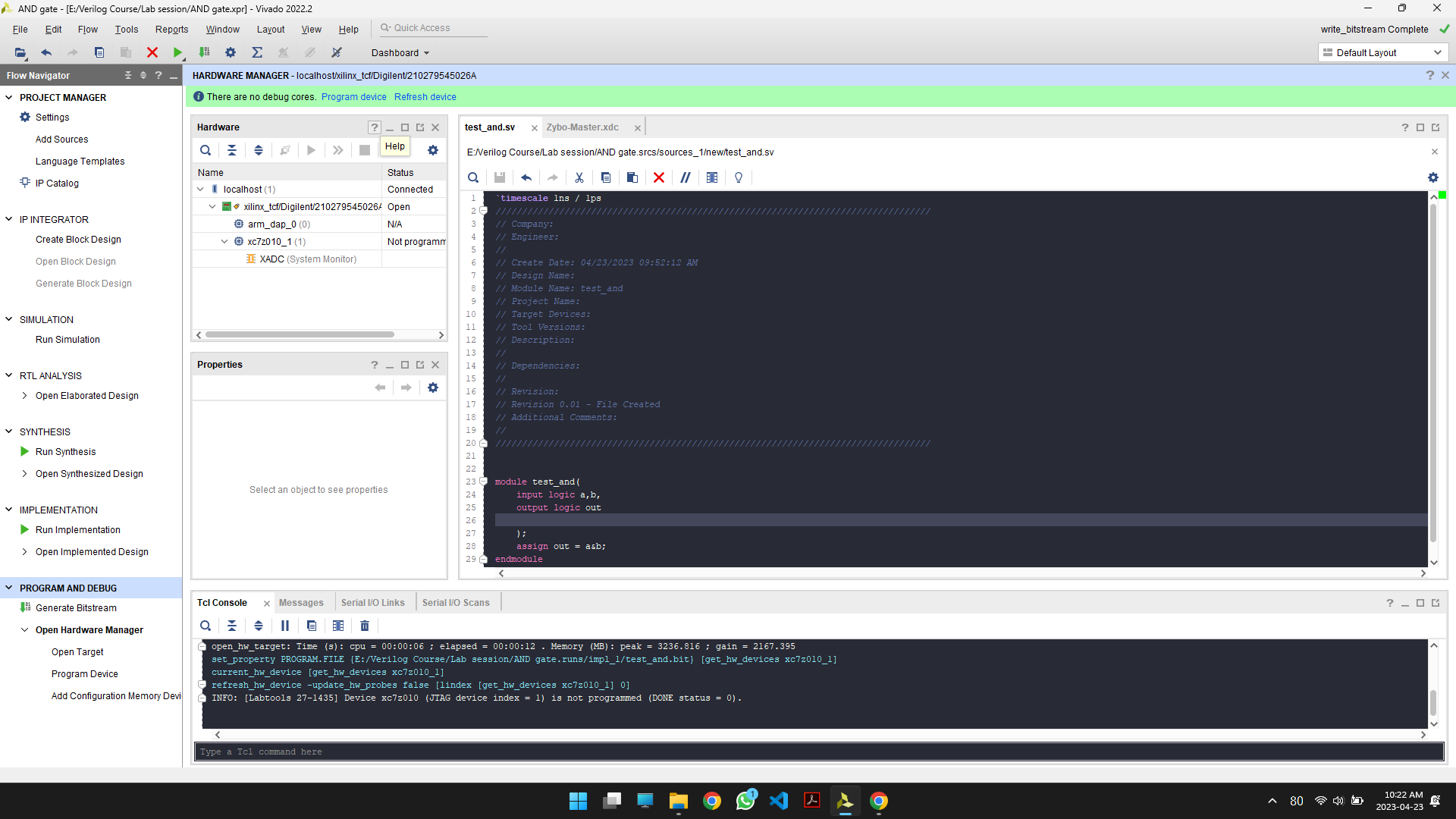
A screenshot of a computer

Description automatically generated with low confidenceThe arrays k and x are then prepared to sent through a serial connection by flattening and concatenated to a single 1 dimensional array.

Next the script monitors the receiving connection and extract the resultant matrix as an array (y). Then the script compares the received array and the computed result and display an error message if any mismatches present. 

Process of programming the FPGA (using Vivado)

1. In the creation of a new RTL project after adding the required source files select the FPGA board for implementation (Zybo in the session conducted) as the default part and create the project.
2. Add the constraints file to map the properties of the board according to the RTL design. Set the top module appropriately from the sources tab, in case of the design containing several source files. Next from generate bit stream from Program and Debug section.
3. Upon successful completion, a separate tab named Open Hardware manager will appear. Now connect the FPGA board to the computer. A red colour bulb will indicate the power on state of the board. Once the device is recognized the following message will appear.



1. Click on the program device button to program the device. Once it is successfully completed a green colour bulb will light up as follows.

A picture containing text, electronics, circuit

Description automatically generated

Power

Utilization reports of synthesis and implementation

Resources

Synthesis

Table

Description automatically generated

Implementation

Table

Description automatically generated

Timing

Synthesis

Graphical user interface, application

Description automatically generated

Implementation

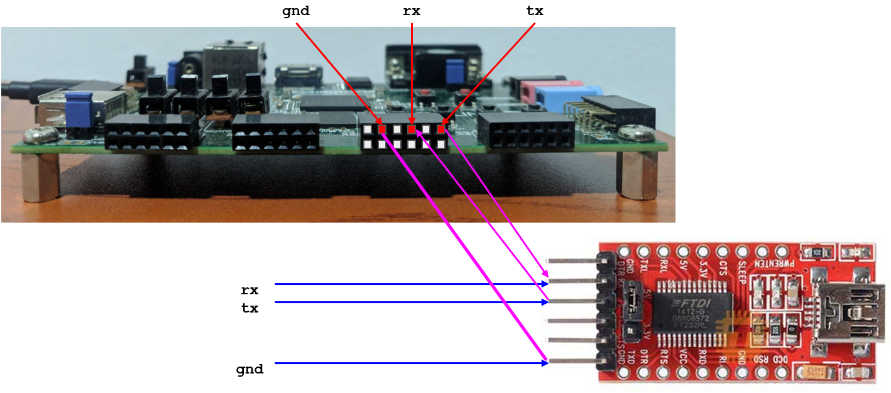
Graphical user interface

Description automatically generated with medium confidence

Matrix vector multiplication

Video of performing a matrix multiplication could not be captured due to several issues in troubleshooting. The following is the complete process in performing the matrix multiplication.

USB-tty is used to communicate between FPGA and the computer using serial communication. First connect USB-tty to the FPGA board using JC Pmod Ports as shown below.



Now connect the USB-tty to computer and get the USB serial port of connection by device manager. Rename the python script with the port identification as follows.

Text

Description automatically generated

Now program the FPGA and reset the circuit with sw[0] on the board. Connect the USB-tty and run the python script. Upon successful completion the results will be as follows. Text

Description automatically generatedText

Description automatically generatedText

Description automatically generated with medium confidence

Text

Description automatically generated