ECE 520 - Spring 2022

California State University, Northridge Department of Electrical and Computer Engineering

**Final Project-Traffic Generator with AXI-4 Stream Master**

Report Created By

Venkatesh Uppugunduru

Report Completed on 5/12/22

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **Contents** | **Page number** |
| Cover Page | 1 |
| Objective | 3 |
| Introduction/AXI | 4 |
| AXI Channels | 5 |
| Traffic Generator | 6 |
| Creating Ip in vivado | 7 |
| Waveform | 12 |
| Conclusion | 12 |

**Objective:**

This report walks you through the steps of creating a simple traffic generator peripheral with a slave AXI4-Lite interface and a master AXI-4 streaming interface using Vivado IP Integrator.

**Hardware Components**

* **Digilent Zybo-20**

**Software**

* **Vivado v2021.2**

**Introduction:**

**Advanced eXtensible Interface (AXI)**

ARM's AMBA specifications include the Advanced eXtensible Interface, or AXI. The AXI is a high-performance, high-speed point-to-point interface designed for microcontroller systems. The AXI protocol uses a point-to-point link to minimize bus sharing, resulting in increased bandwidth and lower latency. AXI is the most widely used of the AMBA interface interconnects.

The AXI protocol's main purpose is to offer a foundation for how distinct blocks inside each chip communicate with one another. It includes a method that must be followed before any data is sent, ensuring that communication is clear and uninterrupted.

The AXI protocol features are:

• It is suitable for high-bandwidth and low-latency designs.

• High-frequency operation is provided, without using complex bridges.

• The protocol meets the interface requirements of a wide range of components.

• It is suitable for memory controllers with high initial access latency.

• Flexibility in the implementation of interconnect architectures is provided.

• It is backward-compatible with AHB and APB interfaces.

**AXI Architecture**

The AXI protocol is burst-based and defines five independent transaction channels:

• Read address, which has signal names beginning with AR.

• Read data, which has signal names beginning with R.

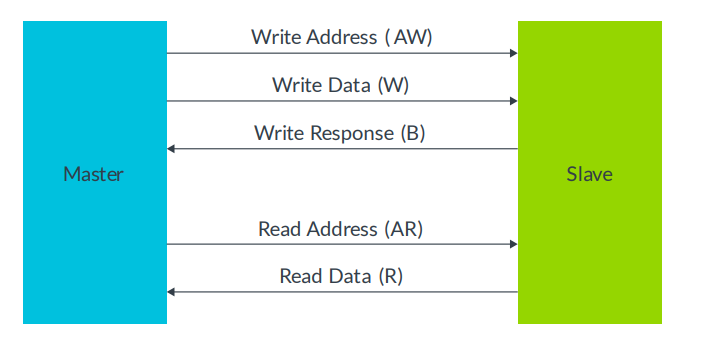
• Write address, which has signal names beginning with AW.

• Write data, which has signal names beginning with W.

• Write response, which has signal names beginning with B.

**AXI Channels**

The AXI specification describes a point-to-point protocol between two interfaces: a master and a slave. The following diagram shows the five main channels that each AXI interface uses for communication:

****

Write operations use the following channels:

• The master sends an address on the Write Address (AW) channel and transfers data on the Write Data (W) channel to the slave.

• The slave writes the received data to the specified address. Once the slave has completed the write operation, it responds with a message to the master on the Write Response (B) channel.

Read operations use the following channels:

• The master sends the address it wants to read on the Read Address (AR) channel.

• The slave sends the data from the requested address to the master on the Read Data (R) channel

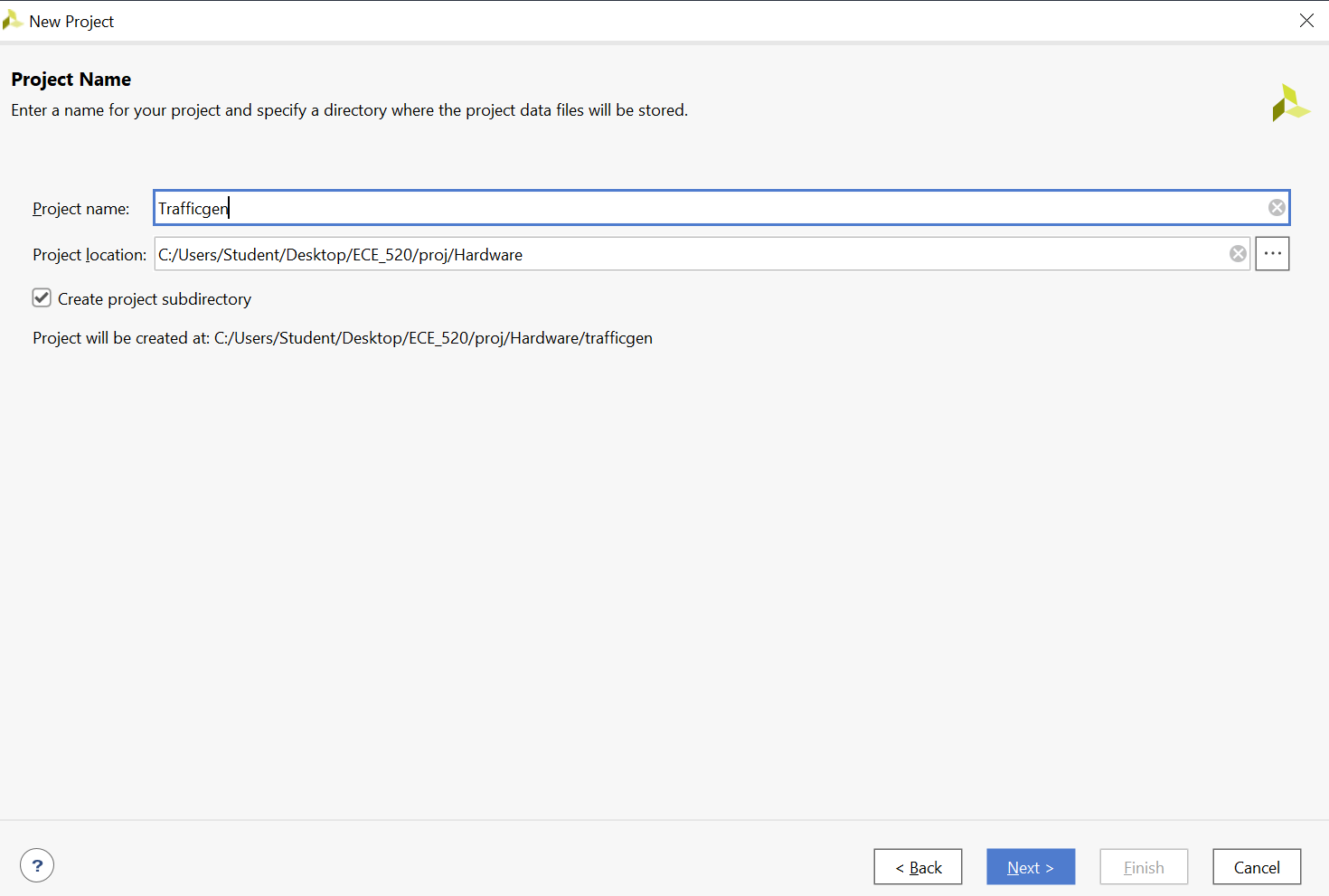
### Traffic Generator IP

### The traffic generator IP we are going to create consists of a top-level module that instantiates a register module and a stream master module. An AXI-Lite slave interface connected to the register module is used to configure the core. We have two parameters to control the functionality of the traffic generator; (i) enable, and (ii) the number of words. The enable signal is used to turn on/off the master module. The output of the master module is a traffic pattern determined by the number of words.

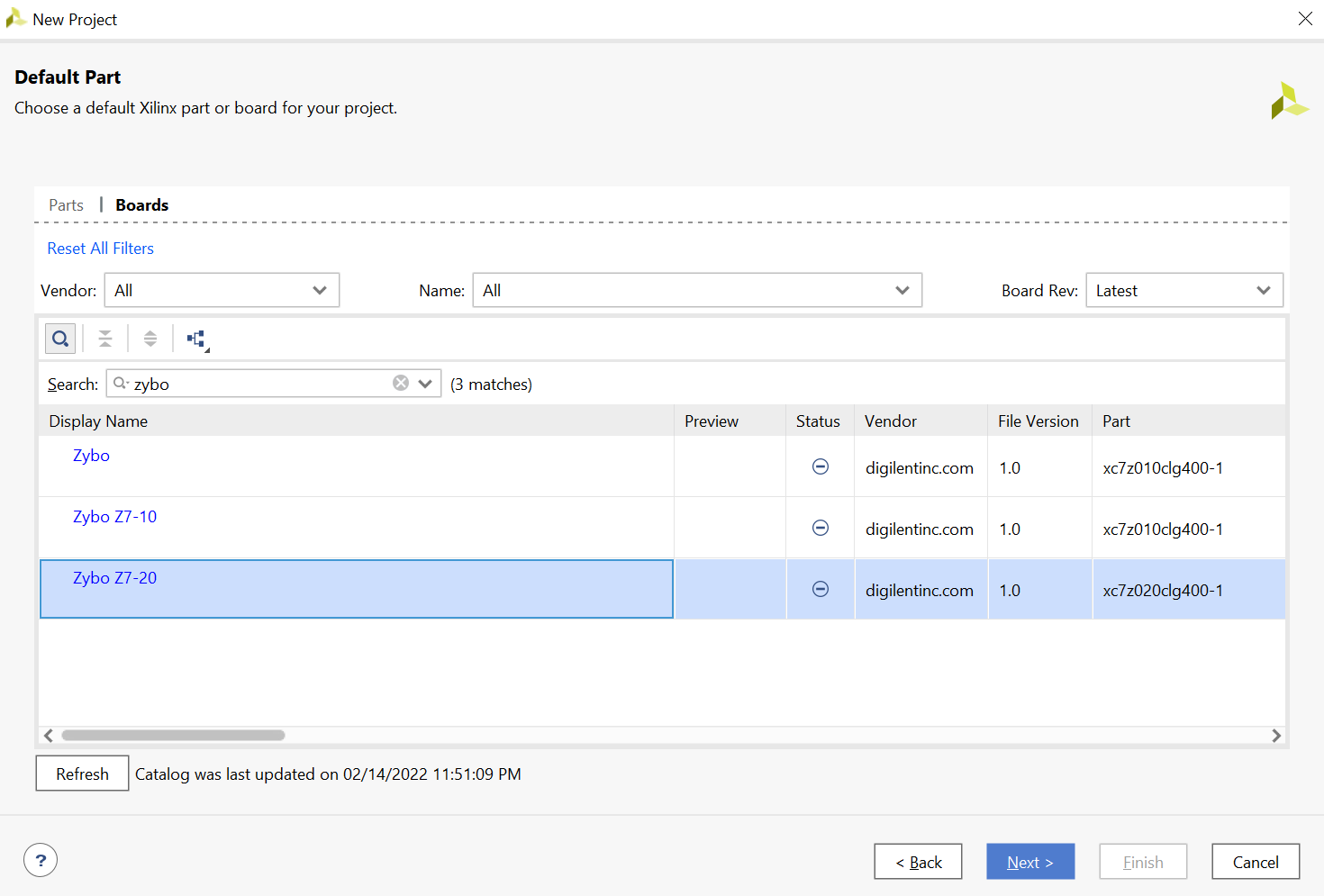
### 

**Creating IP in Vivado**

Step 1:Open up Vivado and create a new RTL project.

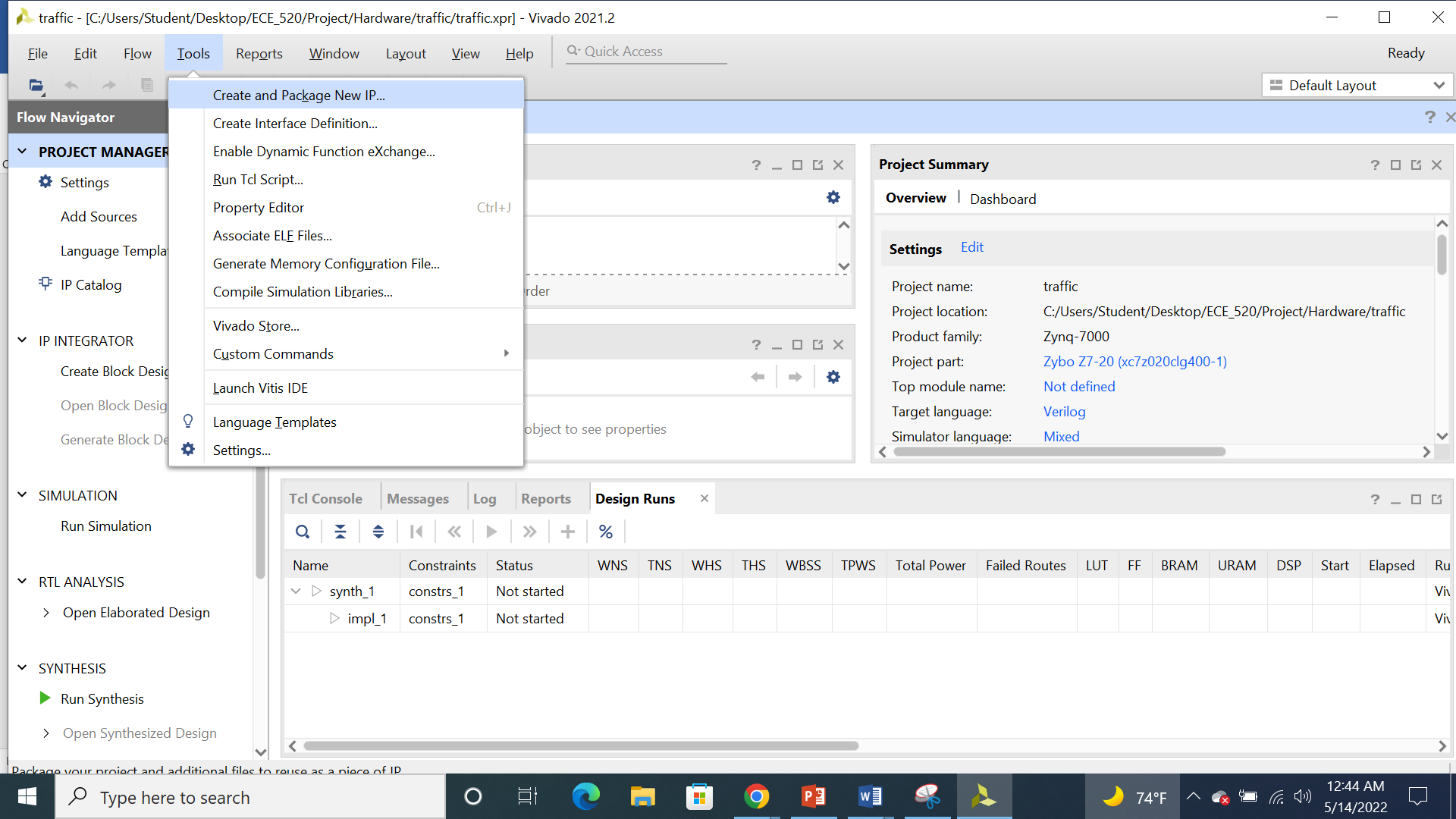


Step 2: Select the board as zybo-20 and click next and finish



Step 3: Create a new package by going to tools🡪 create and package new ip, then

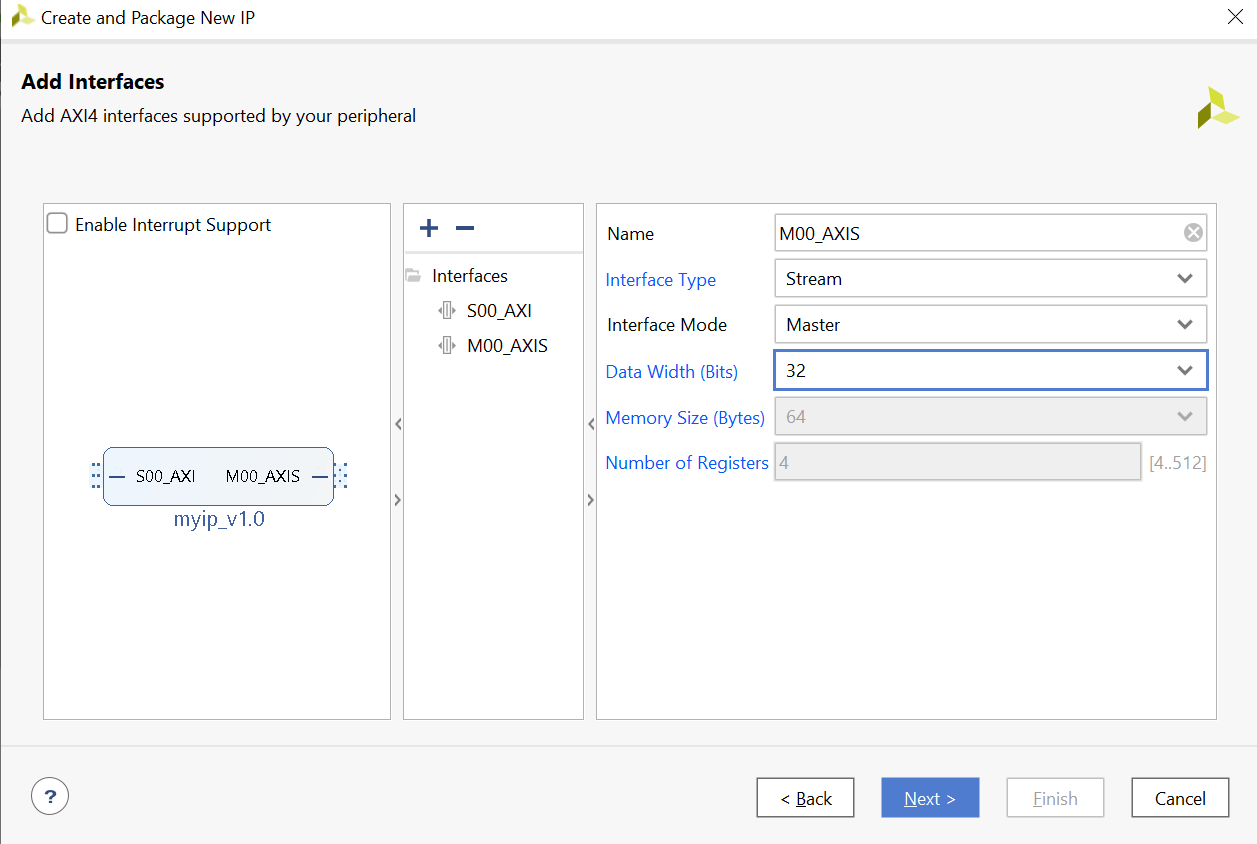
select AXI-4 peripheral.



Step 4: Enter the details for the peripherals and click next.

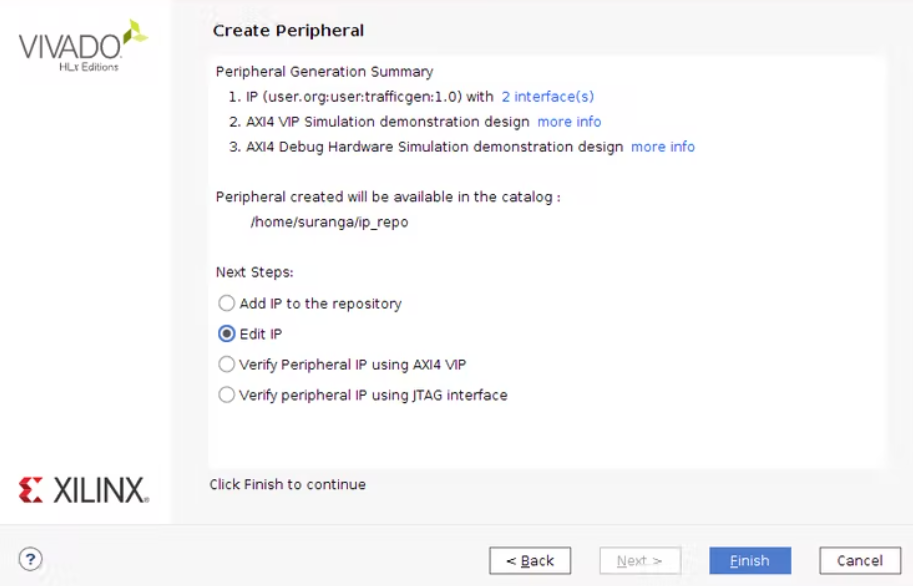
Step 5: Now we will add a stream master interface. This is the interface on which

traffic generated in the IP move toward other peripherals on the FPGA design.

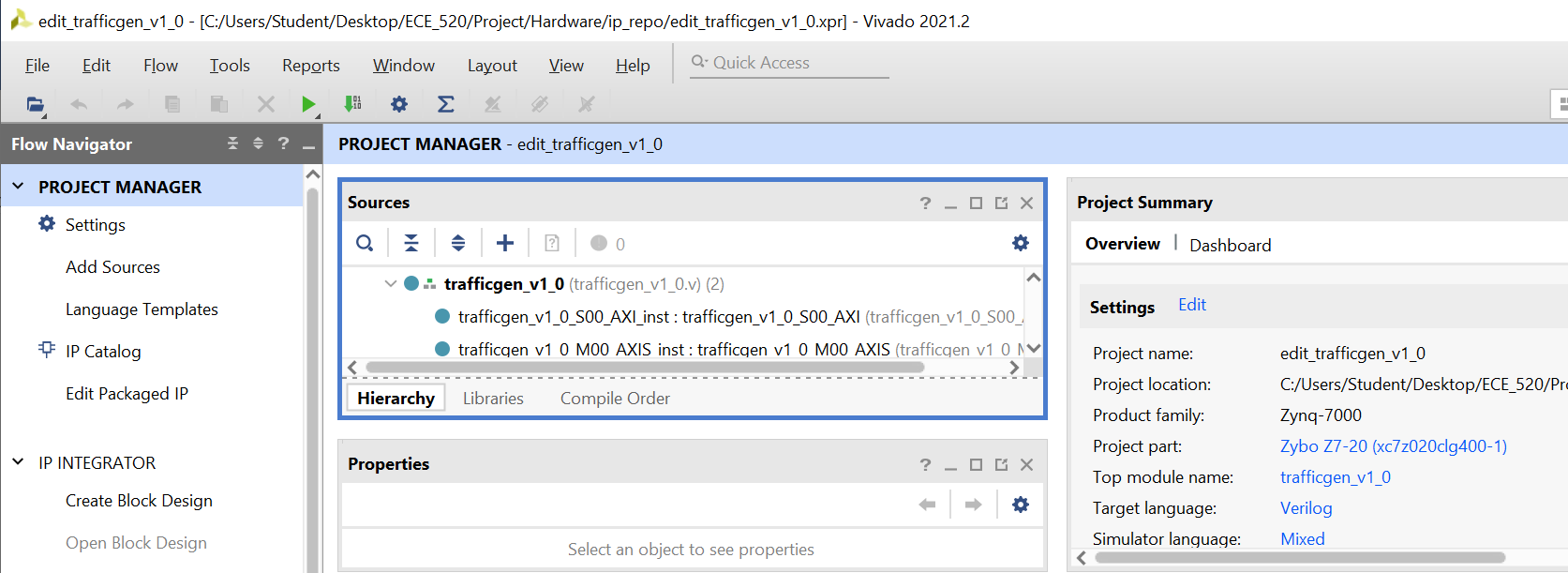


Step 6: Following the creation of interfaces, we must personalize the modules that

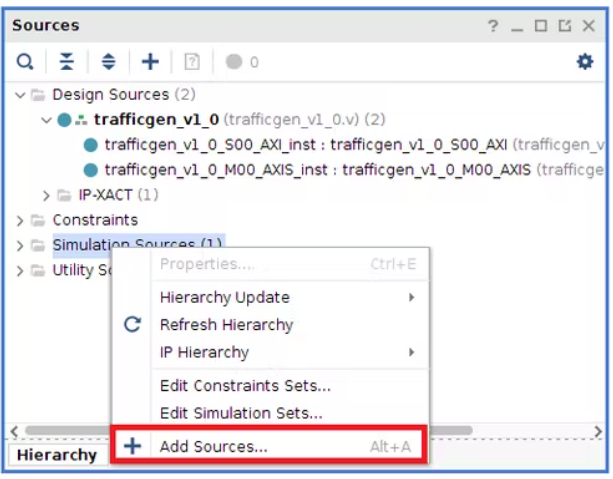
make up our IP. To do so, select Edit IP and then click Finish.



Step 7: Once the project has been created, we would be able to see top module, register module, master module, now make the changes as required.

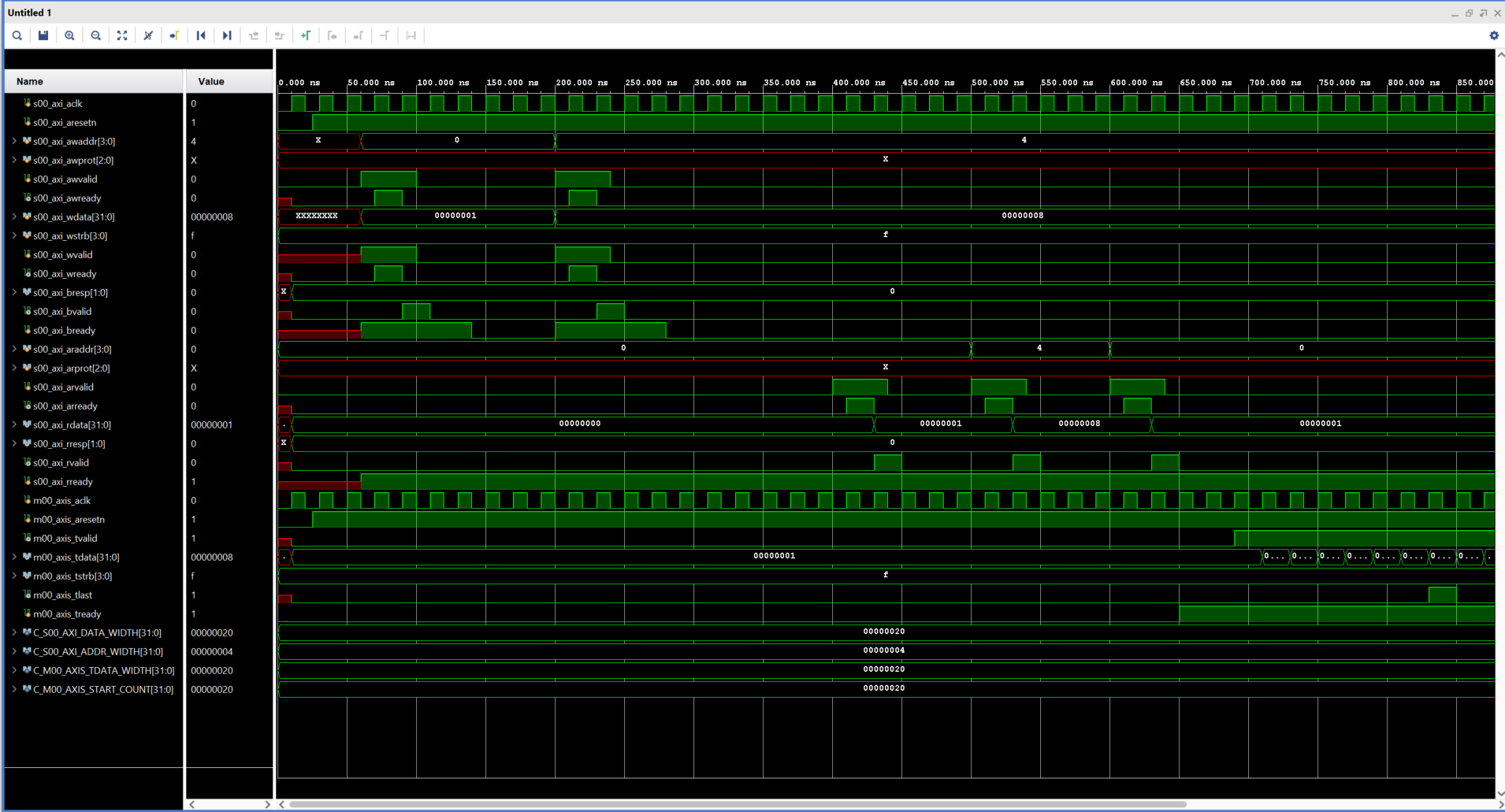


Step 8: Add the testbench file by going to simulation sources🡪add sources.



Step 9: Now add the library for all source files and run the simulation.

Waveform:



**Packaging Ip**

After verifying, we need to re-package the IP. The IP is now ready to be used in another Vivado project.

**Conclusion**

# We have successfully created a Traffic Generator with AXI-4 Stream Master ip which can be implemented in another vivado projects.