Question 4: Using the IP Core for FFT

a.Using Xilinx FFT IP Core

In this part of the Assignment, our goal is to use the **Xilinx FFT IP Core** instead of implementing the FFT algorithm manually. This IP Core can perform FFT computations efficiently and quickly. The following explains how to configure the input and output connections to this IP Core and how to set it up in Vivado.

Xilinx FFT IP Core Configuration:

To use the Xilinx FFT IP Core, we first need to configure it in the **Vivado** environment. The key parameters in this configuration are as follows:

- **FFT Type**: Typically, for complex FFTs, you should configure the number of FFT points.
- **Inputs and Outputs**: Specify the number of bits for the FFT's input and output. In this case, inputs will be divided into real and imaginary parts.
- **Scaling**: We need to adjust scaling to avoid overflow and to ensure accurate FFT computation results.

Input and Output Connections:

Inputs and outputs are directly connected to the appropriate ports of the IP Core. For correct connectivity, the inputs and outputs must be defined in the hardware code and correctly connected to the relevant ports of the IP Core.

1. **Inputs**:
   The inputs consist of real and imaginary signals that are fed into the FFT. These inputs need to be sent to the IP Core in the appropriate bit format. Typically, these inputs are paired as real and imaginary signals that connect to their respective ports. In this case, the inputs are configured for 16 FFT points (for example, 16 points).
2. **Outputs**:
   Similar to the inputs, the outputs are divided into real and imaginary parts. After the FFT computation, the outputs are taken from the IP Core’s output ports. At this stage, the results can be saved or used for further processing in the system.

Configuration and Connection in Vivado:

1. **Create a Project in Vivado**:
   - First, create a new project in Vivado.
   - Then, search for the **Xilinx FFT IP Core** in the **IP Catalog** and add it to the project.
2. **Configure FFT IP Core**:
   - In the IP Core settings, select the number of FFT points (16 points for this example).
   - Set the input and output format to use the correct values for complex signals.
   - Configure the scaling to prevent overflow.
3. **Connect Inputs and Outputs**:
   - The inputs and outputs should be connected to the relevant ports of the IP Core.
   - The complex inputs (comprising real and imaginary data) should be connected to the **real** and **imag** ports of each input data point.
   - Similarly, the outputs will be connected to the **real** and **imag** ports of the output signals.

Output Configuration:

While configuring the **Xilinx FFT IP Core**, outputs must be connected to the ports that store and display the FFT results. These output ports are typically of the **real** and **imag** type and need to be correctly mapped in the Verilog or VHDL code.

In this part, we used the Xilinx FFT IP Core to perform FFT computations instead of implementing the FFT algorithm manually. The configuration of this IP Core includes setting the number of FFT points, configuring scaling, and connecting the inputs and outputs to the IP Core. This IP Core performs efficiently, providing high-speed and accurate FFT computations.