4-BIT FULL ADDER DESIGN WITH IP CATALOG IN XILINX VIVADO

by

NAME	ROLL NUMBER	REGISTRATION NO.
1. Ayan Munshi	35000323063	233500120325

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS & COMMUNICATION ENGINEERING

Department of Electronics & Communication Engineering Ramkrishna Mahato Government Engineering College, Purulia

Affiliated to Maulana Abul Kalam Azad University of Technology, West Bengal

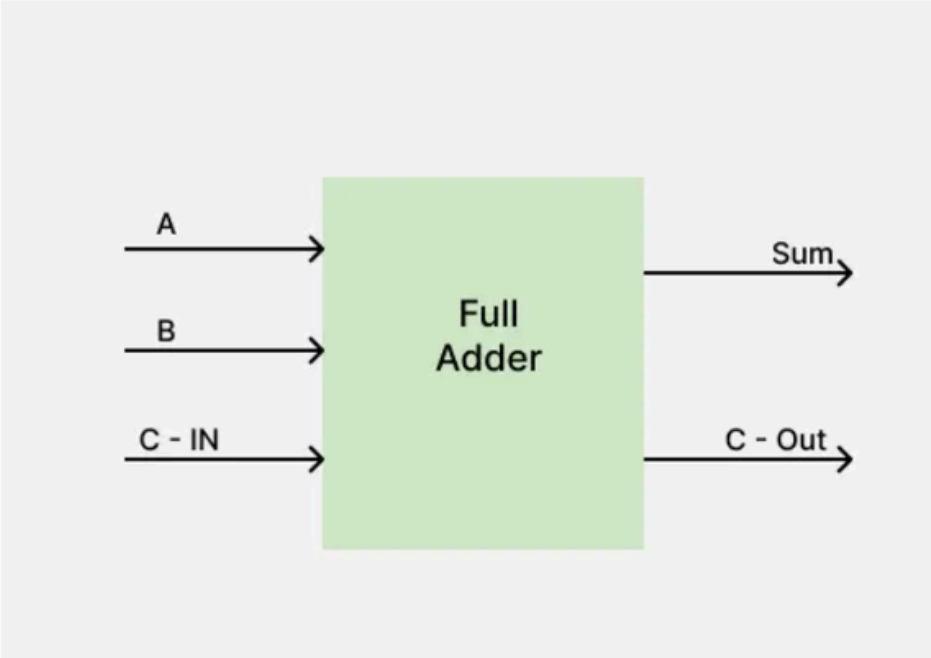
AGHARPUR, JOYPUR, PURULIA – 723 103

Introduction

A 4-bit full adder is a digital circuit that performs binary addition on two 4-bit numbers and an optional carry-in. It consists of four cascaded 1-bit full adders, each responsible for adding corresponding bits from the input numbers along with the carry from the previous stage. The inputs are $A_3A_2A_1A_0$ and $B_3B_2B_1B_0$, with Cin as the initial carry-in. The outputs are the 4-bit sum $S_3S_2S_1S_0$ and a final carry-out Cout.

Each full adder computes the sum and carry using the logic:

- Sum = $A \oplus B \oplus Cin$
- Cout = $(A \cdot B) + (Cin \cdot (A \oplus B))$



Inputs			Outputs	
A	B	C_{in}	S (Sum)	C_{out} (Carry)
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Design 4 BirAdder Using Xilinx Vivado ML Edition

To simulate a 4-bit adder using Xilinx Vivado, you'll go through a few key steps: writing the HDL code, creating a testbench, running simulation, and analyzing the waveform. Using the IP Catalog in Xilinx Vivado is a powerful way to simulate a 4-bit adder without writing the entire HDL code manually.

1. Create a New Vivado Project

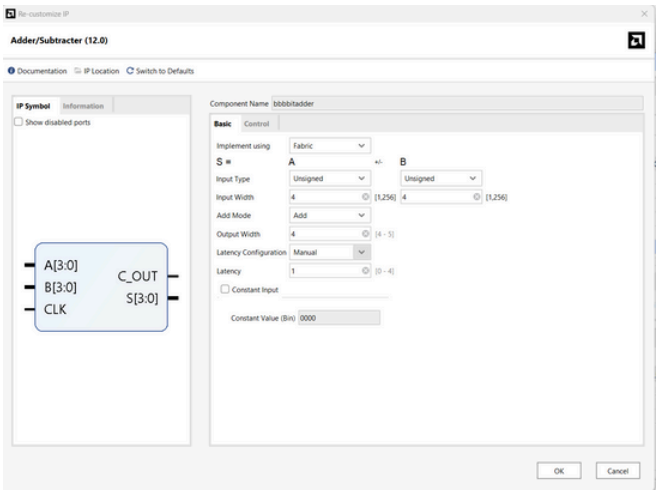
- Launch Vivado and select "Create New Project"
- Name your project (e.g., Adder_IP_Sim)
- Choose RTL Project, and skip adding sources for now
- Select your FPGA part (e.g., xc7a100tcsg324-1 for Artix-7)

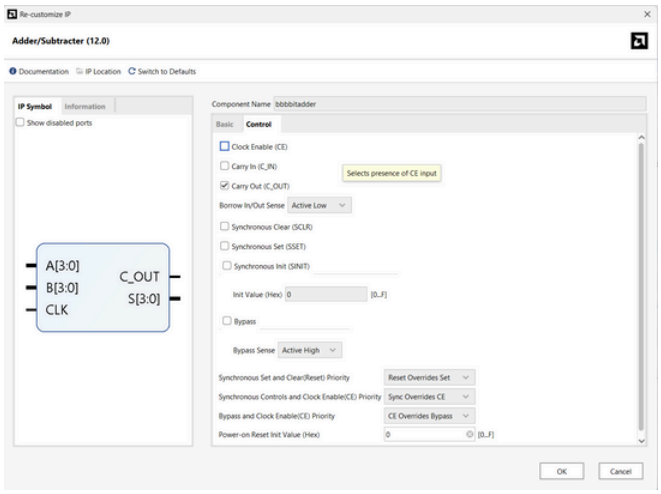
2.Open IP Catalog

- In the Flow Navigator, go to "IP Catalog"
- Search for "Adder/Subtracter" or "Arithmetic Functions"
- Select "Adder/Subtracter" IP and double-click to customize

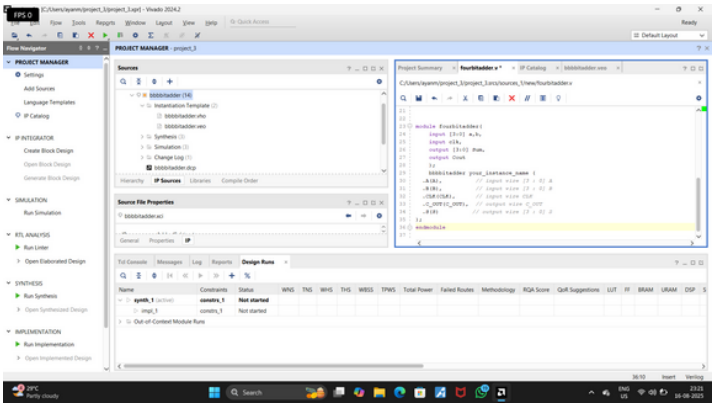
3. Configure the IP

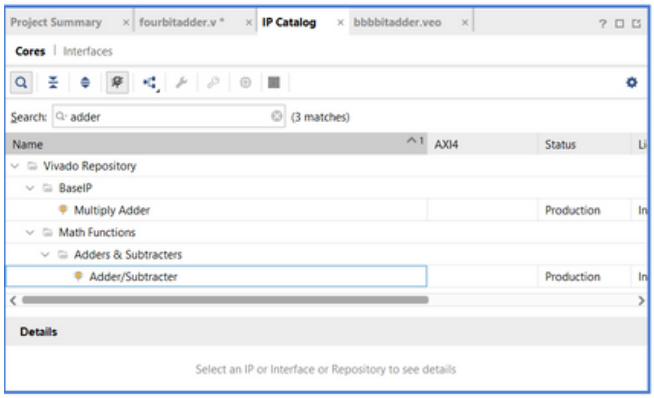
- Set Input Width to 4 bits
- Choose Unsigned or Signed based on your need
- Enable Carry In and Carry Out if required
- Click "OK" to generate the IP



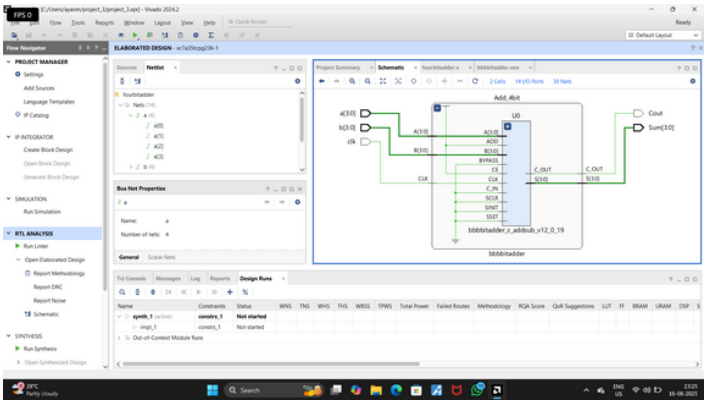


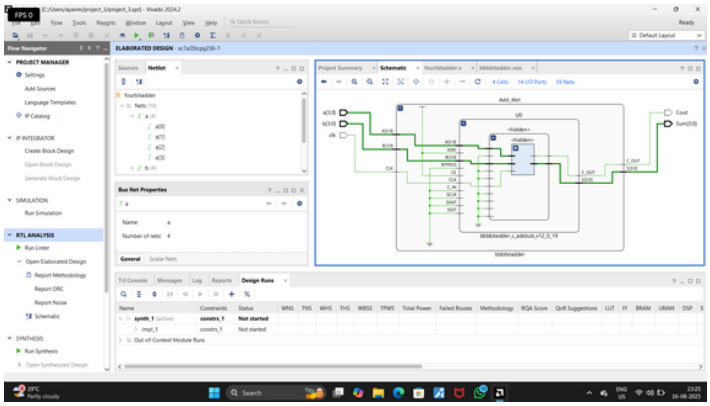
4.Source code with IP catalog





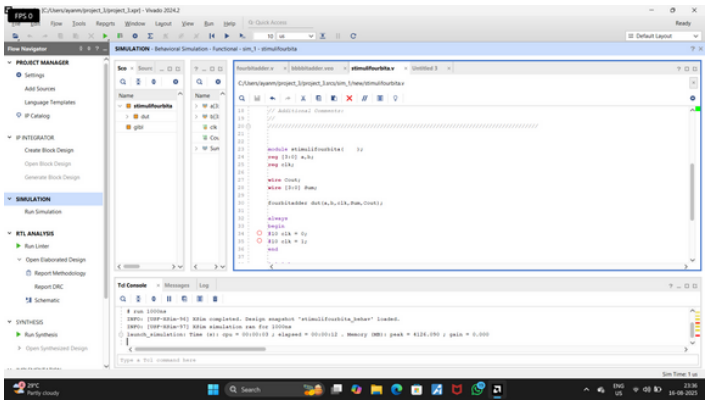
5.cheak the Schematic inner view

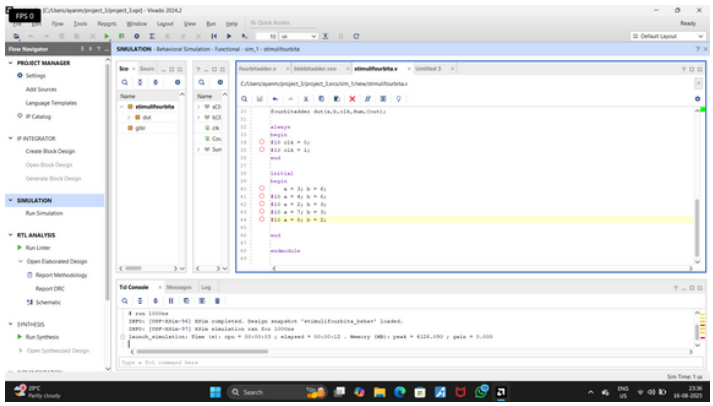




6. Write the Testbench

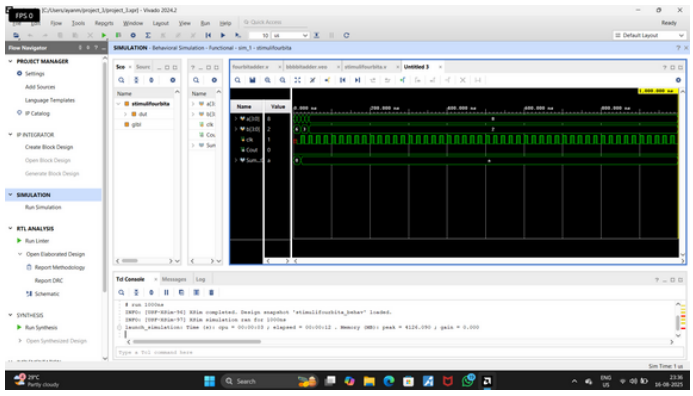
- Create a new Verilog file for the testbench.
- Instantiate the wrapper module and apply various input combinations to test the adder.

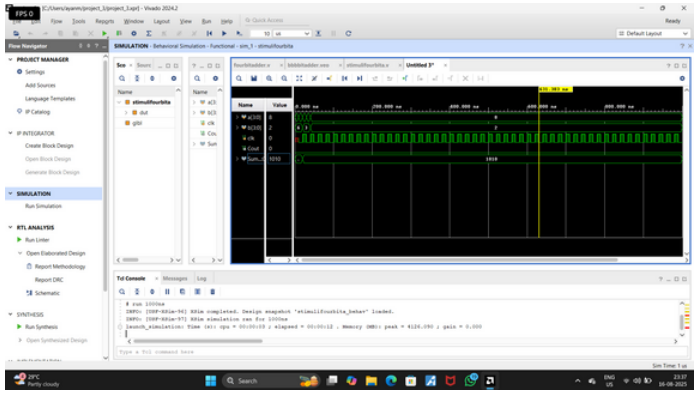




7. Run Behavioral Simulation

- Go to "Run Simulation" → "Run Behavioral Simulation".
- Use the waveform viewer to verify the sum and carry outputs.





Run Synthesis