# LAB 6

**OBJECTIVE:** To design and simulate a **Full Adder Circuit** using VHDL, and verify its functionality using a testbench.

## TOOLS USED:

- VHDL (VHSIC Hardware Description Language)
- https://www.edaplayground.com/ (for simulation)

## THEORY:

A Full Adder adds three one-bit binary numbers (A, B, and Carry-in) and outputs a Sum and a Carry-out. It performs the function:

- Sum (S) = A XOR B XOR Cin
- Carry-out (Cout) = (A XOR B) AND Cin OR (A AND B)

## VHDL CODE

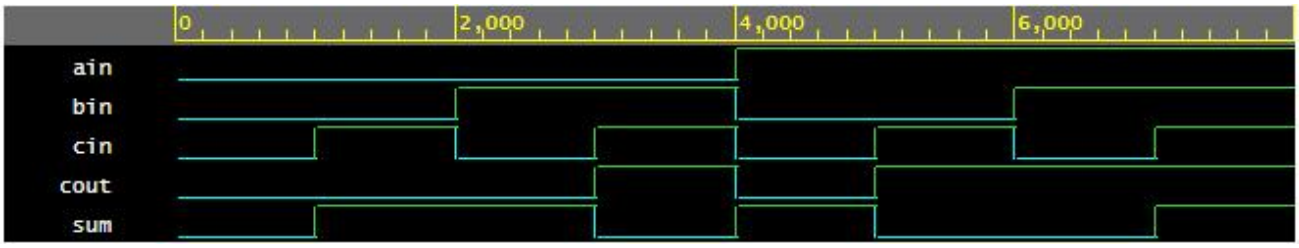
**DESIGN**

```
library ieee;
use ieee.std_logic_1164.all;

entity fa is
	port(
		a: in std_ulogic;
		b: in std_ulogic;
		cin: in std_ulogic;
		cout: out std_ulogic;
		s: out std_ulogic
	      );
end fa;

architecture	behave of fa is
begin
	s <= (a xor b) xor cin;
	cout <= ((a xor b) and cin) or (a and b);
end behave;
```

**TESTBENCH**

```
library ieee;
use ieee.std_logic_1164.all;

entity fa_testbench is
end fa_testbench;

architecture test of fa_testbench is
	component fa
		port(
			a: in std_ulogic;
			b: in std_ulogic;
			cin: in std_ulogic;
			cout: out std_ulogic;
			s: out std_ulogic
		);
	end component;

	signal ain, bin, cin, cout, sum : std_logic;

	begin
		full_adder: fa port map (a => ain, b => bin, cin => cin, cout => cout, s => sum);

		process begin
			ain <= '0';bin <= '0';cin <= '0';wait for 1 ns;
			ain <= '0';bin <= '0';cin <= '1';wait for 1 ns;
			ain <= '0';bin <= '1';cin <= '0';wait for 1 ns;
			ain <= '0';bin <= '1';cin <= '1';wait for 1 ns;
			ain <= '1';bin <= '0';cin <= '0';wait for 1 ns;
			ain <= '1';bin <= '0';cin <= '1';wait for 1 ns;
			ain <= '1';bin <= '1';cin <= '0';wait for 1 ns;
		ain <= '1';bin <= '1';cin <= '1';wait for 1 ns;
			assert false report "Reached end of test";
			wait;
		end process;
end test;
```

Output:




## CONCLUSION:

The Full Adder was successfully implemented and simulated in VHDL. All possible input combinations were tested, and the output matched the expected values as per the truth table. This confirms the correct functionality of the Full Adder logic.