******

**JK Lakshmipat University**

[**Computer**](https://canvas.instructure.com/courses/8496533) **Organization and Architecture**

**ASSIGNMENT 1**

**Name – Indrajit Roy**

**Section – A**

**Group- A2**

**Roll No.- 2023BTECH037**

**Date of Submission – 23rd September, 2024**

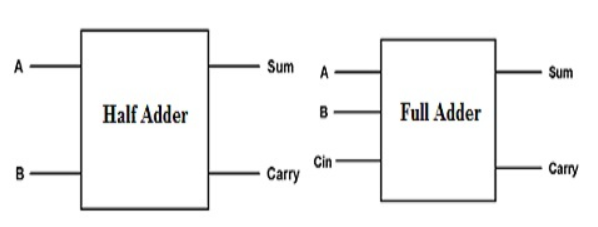
**Submitted To – DR. Pranab Roy Sir**

**Question 1**

**Design a VHDL model using dataflow architecture for half adder and full adder**

**Aim : To design and implement VHDL models using dataflow architecture for a half adder and a full adder, which perform basic binary addition operations. The half adder adds two single-bit inputs, while the full adder extends this functionality by adding three inputs, including a carry-in bit, to generate a sum and a carry-out.**

**Circuit Diagram :**

******

**HALF ADDER:**

***● VHDL Code:***

***----------------------------------------------------------------------------------***

***-- Company:***

***-- Engineer:***

***--***

***-- Create Date: 08:30:07 09/20/2024***

***-- Design Name:***

***-- Module Name: HalfAdder - Dataflow***

***-- Project Name:***

***-- Target Devices:***

***-- Tool versions:***

***-- Description:***

***--***

***-- Dependencies:***

***--***

***-- Revision:***

***-- Revision 0.01 - File Created***

***-- Additional Comments:***

***--***

***----------------------------------------------------------------------------------***

***library IEEE;***

***use IEEE.STD\_LOGIC\_1164.ALL;***

***-- Uncomment the following library declaration if using***

***-- arithmetic functions with Signed or Unsigned values***

***--use IEEE.NUMERIC\_STD.ALL;***

***-- Uncomment the following library declaration if instantiating***

***-- any Xilinx primitives in this code.***

***--library UNISIM;***

***--use UNISIM.VComponents.all;***

***entity HalfAdder is***

***Port ( A : in STD\_LOGIC;***

***B : in STD\_LOGIC;***

***Sum : out STD\_LOGIC;***

***Carry : out STD\_LOGIC);***

***end HalfAdder;***

***architecture Dataflow of HalfAdder is***

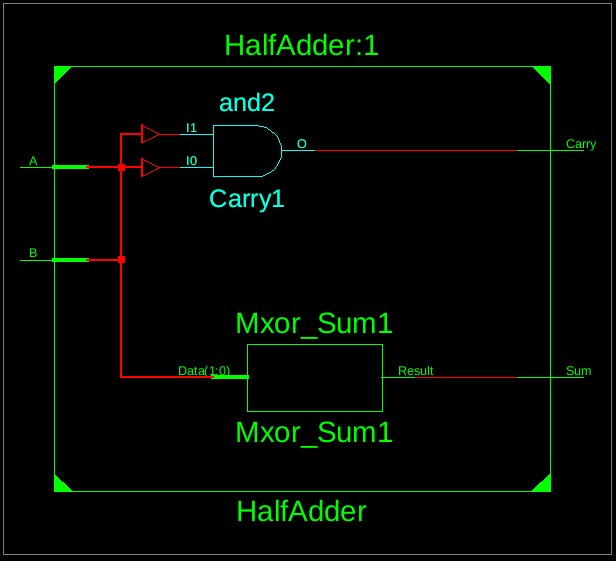
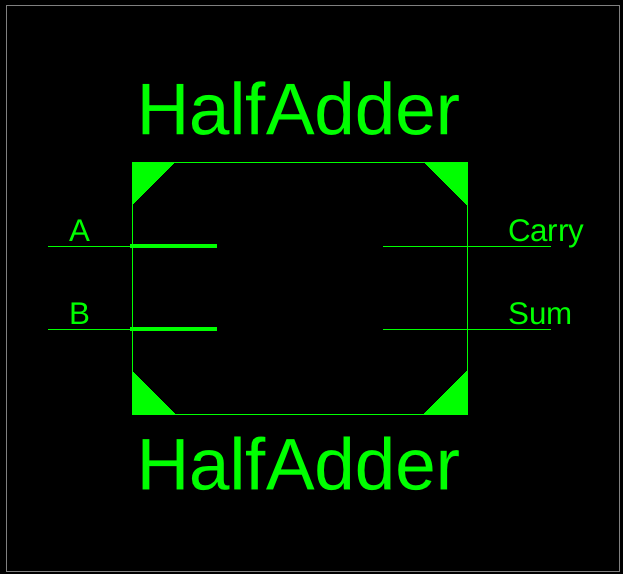
***begin***

***Sum <= A XOR B;***

***Carry <= A AND B;***

***end Dataflow;***

***● RTL Diagram:***



***● Testbench Code:***

***-- Company:***

***-- Engineer:***

***-- Create Date: 08:48:05 09/20/2024***

***-- Design Name:***

***-- Module Name: /home/ise/Assignment1/tb\_HalfAdder.vhd***

***-- Project Name: Assignment1***

***-- Target Device:***

***-- Tool versions:***

***-- Description:***

***-- VHDL Test Bench Created by ISE for module: HalfAdder***

***-- Dependencies:***

***-- Revision:***

***-- Revision 0.01 - File Created***

***-- Additional Comments:***

***-- Notes:***

***-- This testbench has been automatically generated using types std\_logic and***

***-- std\_logic\_vector for the ports of the unit under test. Xilinx recommends***

***-- that these types always be used for the top-level I/O of a design in order***

***-- to guarantee that the testbench will bind correctly to the post-implementation***

***-- simulation model.***

***--------------------------------------------------------------------------------***

***LIBRARY ieee;***

***USE ieee.std\_logic\_1164.ALL;***

***-- Uncomment the following library declaration if using***

***-- arithmetic functions with Signed or Unsigned values***

***--USE ieee.numeric\_std.ALL;***

***ENTITY tb\_HalfAdder IS***

***END tb\_HalfAdder;***

***ARCHITECTURE behavior OF tb\_HalfAdder IS***

***-- Component Declaration for the Unit Under Test (UUT)***

***COMPONENT HalfAdder***

***PORT(***

***A : IN std\_logic;***

***B : IN std\_logic;***

***Sum : OUT std\_logic;***

***Carry : OUT std\_logic***

***);***

***END COMPONENT;***

***--Inputs***

***signal A : std\_logic := '0';***

***signal B : std\_logic := '0';***

***--Outputs***

***signal Sum : std\_logic;***

***signal Carry : std\_logic;***

***-- No clocks detected in port list. Replace <clock> below with***

***-- appropriate port name***

***BEGIN***

***-- Instantiate the Unit Under Test (UUT)***

***uut: HalfAdder PORT MAP (***

***A => A,***

***B => B,***

***Sum => Sum,***

***Carry => Carry***

***);***

***-- Stimulus process***

***stim\_proc: process(A,B)***

***begin***

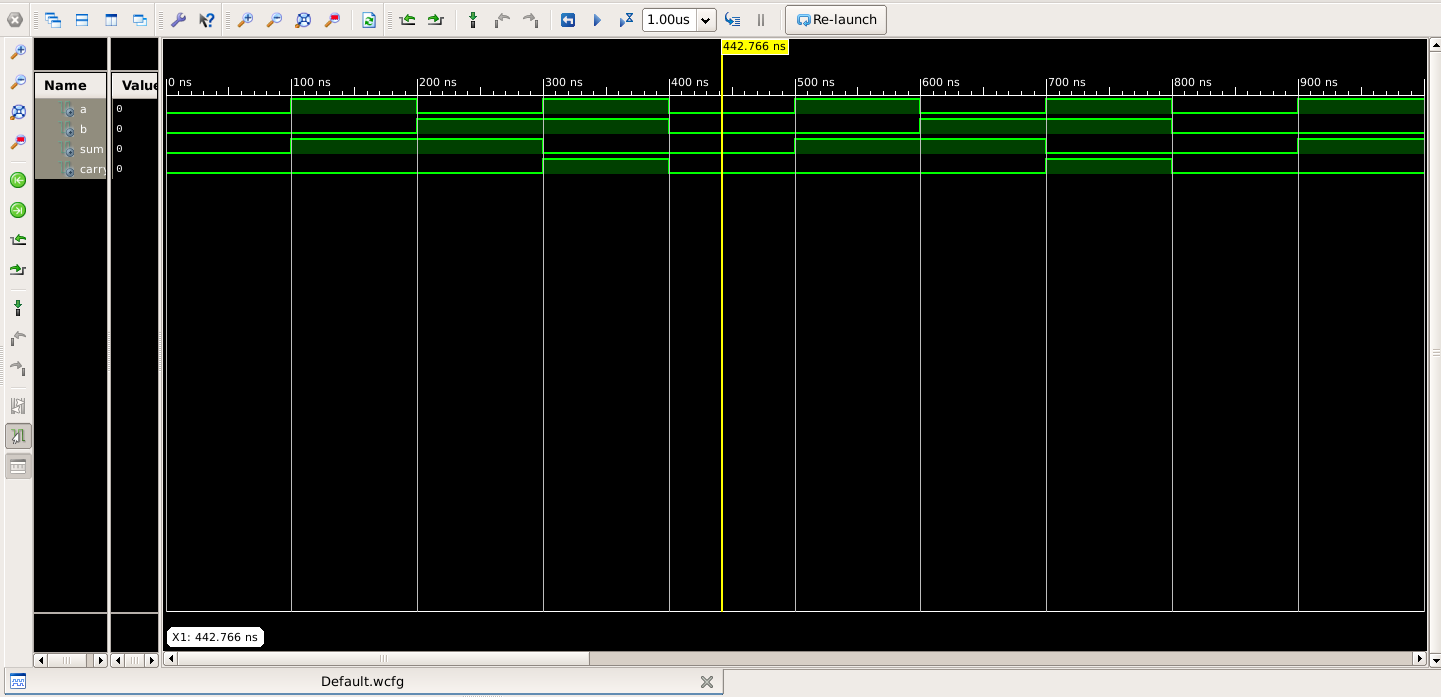
***A <= NOT A AFTER 100 NS;***

***b <= NOT B AFTER 200 NS;***

***end process;***

***END;***

***● Waveform:***



**FULL ADDER:**

***● VHDL CODE:***

***----------------------------------------------------------------------------------***

***-- Company:***

***-- Engineer:***

***--***

***-- Create Date: 14:42:54 09/22/2024***

***-- Design Name:***

***-- Module Name: FullAdder - Dataflow***

***-- Project Name:***

***-- Target Devices:***

***-- Tool versions:***

***-- Description:***

***--***

***-- Dependencies:***

***--***

***-- Revision:***

***-- Revision 0.01 - File Created***

***-- Additional Comments:***

***--***

***----------------------------------------------------------------------------------***

***library IEEE;***

***use IEEE.STD\_LOGIC\_1164.ALL;***

***-- Uncomment the following library declaration if using***

***-- arithmetic functions with Signed or Unsigned values***

***--use IEEE.NUMERIC\_STD.ALL;***

***-- Uncomment the following library declaration if instantiating***

***-- any Xilinx primitives in this code.***

***--library UNISIM;***

***--use UNISIM.VComponents.all;***

***entity FullAdder is***

***Port ( A : in STD\_LOGIC;***

***B : in STD\_LOGIC;***

***Cin : in STD\_LOGIC;***

***Sum : out STD\_LOGIC;***

***Cout : out STD\_LOGIC);***

***end FullAdder;***

***architecture Dataflow of FullAdder is***

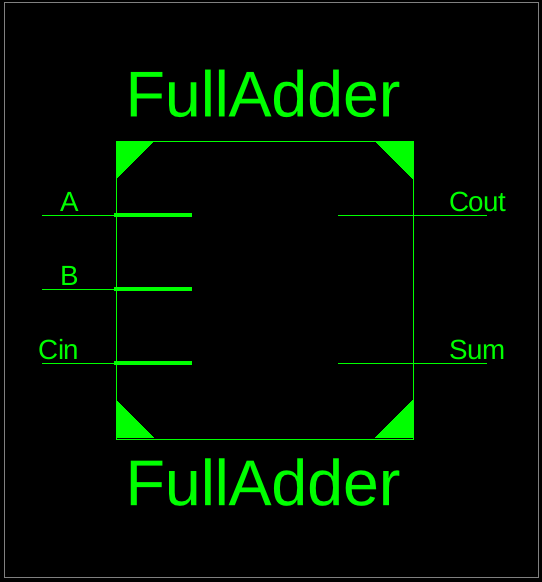
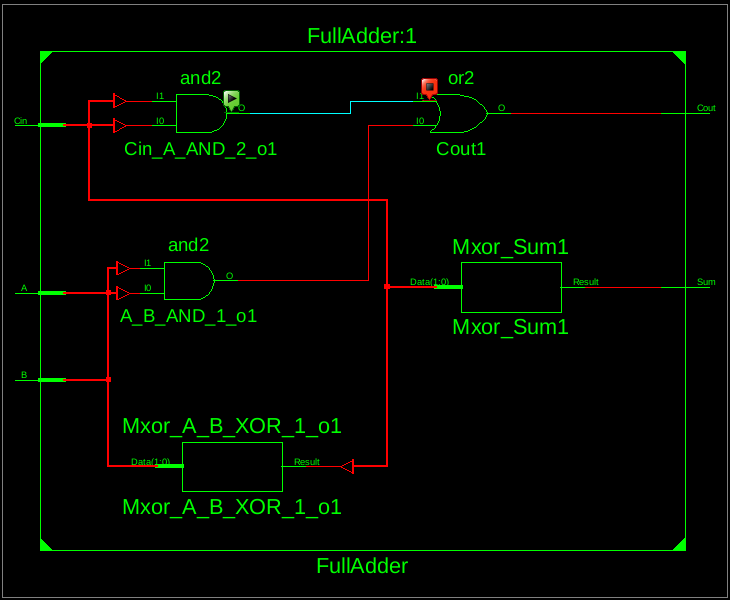
***begin***

***Sum <= A XOR B XOR Cin;***

***Cout <= (A AND B) OR (Cin AND (A XOR B));***

***end Dataflow;***

***● RTL Diagram:***

***● Testbench Code:***

***-------------------------------------------------------------------------------***

***-- Company:***

***-- Engineer:***

***-- Create Date: 14:47:37 09/22/2024***

***-- Design Name:***

***-- Module Name: /home/ise/Assignment1/tb\_FullAdder.vhd***

***-- Project Name: Assignment1***

***-- Target Device:***

***-- Tool versions:***

***-- Description:***

***-- VHDL Test Bench Created by ISE for module: FullAdder***

***-- Dependencies:***

***-- Revision:***

***-- Revision 0.01 - File Created***

***-- Additional Comments:***

***-- Notes:***

***-- This testbench has been automatically generated using types std\_logic and***

***-- std\_logic\_vector for the ports of the unit under test. Xilinx recommends***

***-- that these types always be used for the top-level I/O of a design in order***

***-- to guarantee that the testbench will bind correctly to the post-implementation***

***-- simulation model.***

***--------------------------------------------------------------------------------***

***LIBRARY ieee;***

***USE ieee.std\_logic\_1164.ALL;***

***-- Uncomment the following library declaration if using***

***-- arithmetic functions with Signed or Unsigned values***

***--USE ieee.numeric\_std.ALL;***

***ENTITY tb\_FullAdder IS***

***END tb\_FullAdder;***

***ARCHITECTURE behavior OF tb\_FullAdder IS***

***-- Component Declaration for the Unit Under Test (UUT)***

***COMPONENT FullAdder***

***PORT(***

***A : IN std\_logic;***

***B : IN std\_logic;***

***Cin : IN std\_logic;***

***Sum : OUT std\_logic;***

***Cout : OUT std\_logic***

***);***

***END COMPONENT;***

***--Inputs***

***signal A : std\_logic := '0';***

***signal B : std\_logic := '0';***

***signal Cin : std\_logic := '0';***

***--Outputs***

***signal Sum : std\_logic;***

***signal Cout : std\_logic;***

***-- No clocks detected in port list. Replace <clock> below with***

***-- appropriate port name***

***BEGIN***

***-- Instantiate the Unit Under Test (UUT)***

***uut: FullAdder PORT MAP (***

***A => A,***

***B => B,***

***Cin => Cin,***

***Sum => Sum,***

***Cout => Cout***

***);***

***-- Stimulus process***

***stim\_proc: process(A,B,Cin)***

***begin***

***A <= NOT A AFTER 100 NS;***

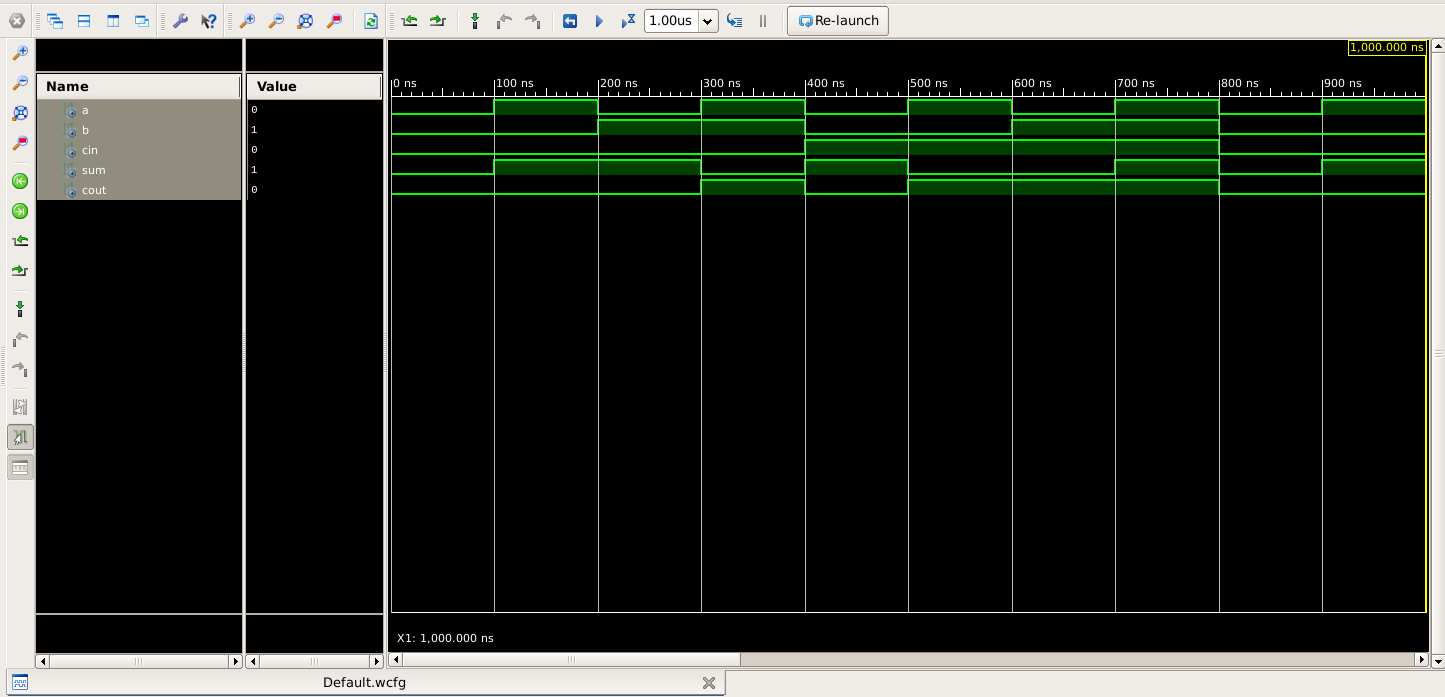
***B <= NOT B AFTER 200 NS;***

***Cin <= NOT Cin AFTER 400 NS;***

***end process;***

***END;***

***● Waveform:***



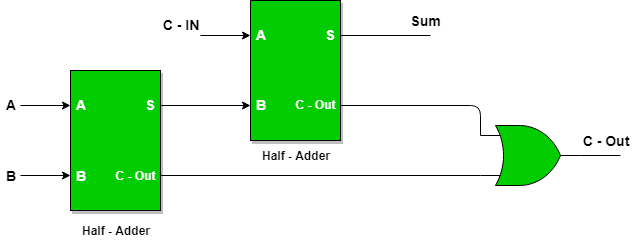
**● CONCLUSION: The Result Obtained by writing VHDL Code for Half Adder and Full Adder are verified with their respective Waveform.**

**Question 2**

**Design a VHDL Model of a full adder using half adder components**

**Aim : The aim of this design is to create a VHDL model of a full adder by utilizing two half adders and an OR gate. A full adder is a digital circuit that computes the sum of three binary bits: two significant bits and a carry bit from a previous stage. By using half adders, the design demonstrates modularity and the ability to construct complex circuits from simpler components.**

**Circuit Diagram :**

******

***● VHDL Code:***

***----------------------------------------------------------------------------------***

***-- Company:***

***-- Engineer:***

***--***

***-- Create Date: 14:57:36 09/22/2024***

***-- Design Name:***

***-- Module Name: FullAdderUsingHalfAdders - Structural***

***-- Project Name:***

***-- Target Devices:***

***-- Tool versions:***

***-- Description:***

***-- Dependencies:***

***-- Revision:***

***-- Revision 0.01 - File Created***

***-- Additional Comments:***

***--***

***----------------------------------------------------------------------------------***

***library IEEE;***

***use IEEE.STD\_LOGIC\_1164.ALL;***

***-- Uncomment the following library declaration if using***

***-- arithmetic functions with Signed or Unsigned values***

***--use IEEE.NUMERIC\_STD.ALL;***

***-- Uncomment the following library declaration if instantiating***

***-- any Xilinx primitives in this code.***

***--library UNISIM;***

***--use UNISIM.VComponents.all;***

***entity FullAdderUsingHalfAdders is***

***Port ( A : in STD\_LOGIC;***

***B : in STD\_LOGIC;***

***Cin : in STD\_LOGIC;***

***Sum : out STD\_LOGIC;***

***Cout : out STD\_LOGIC);***

***end FullAdderUsingHalfAdders;***

***architecture Structural of FullAdderUsingHalfAdders is***

***signal S1, C1, C2 : STD\_LOGIC;***

***component HalfAdder is***

***Port ( A : in STD\_LOGIC;***

***B : in STD\_LOGIC;***

***Sum : out STD\_LOGIC;***

***Carry : out STD\_LOGIC);***

***end component;***

***begin***

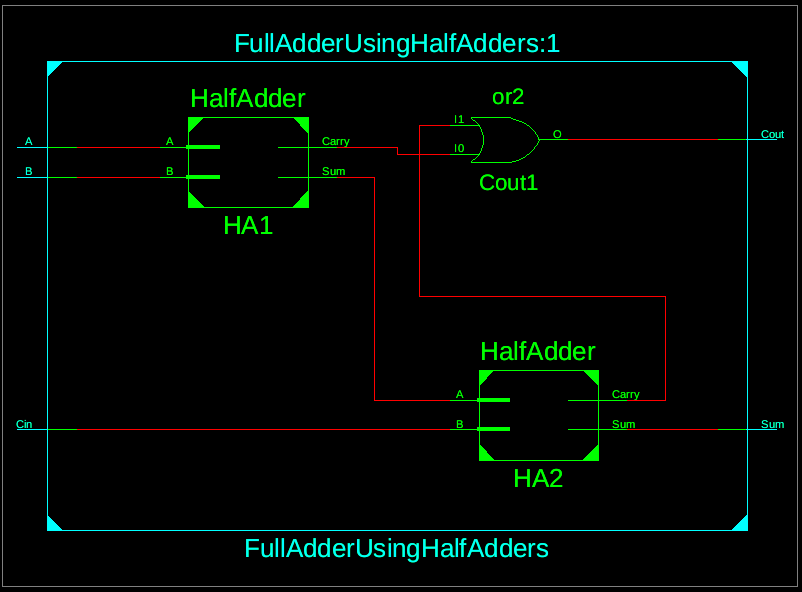
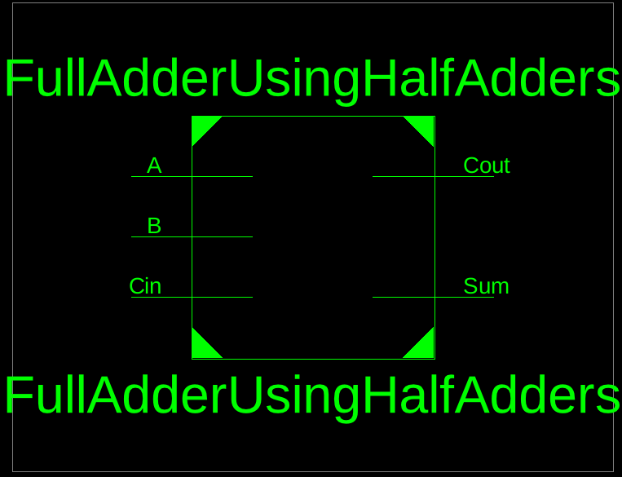
***HA1: HalfAdder port map (A, B, S1, C1);***

***HA2: HalfAdder port map (S1, Cin, Sum, C2);***

***Cout <= C1 OR C2;***

***end Structural;***

***● RTL Diagram:***



***● Testbench Code:   
--------------------------------------------------------------------------------***

***-- Company:***

***-- Engineer:***

***-- Create Date: 15:04:19 09/22/2024***

***-- Design Name:***

***-- Module Name: /home/ise/Assignment1/tb\_FullAdderUsingHalfAdders.vhd***

***-- Project Name: Assignment1***

***-- Target Device:***

***-- Tool versions:***

***-- Description:***

***-- VHDL Test Bench Created by ISE for module: FullAdderUsingHalfAdders***

***-- Dependencies:***

***-- Revision:***

***-- Revision 0.01 - File Created***

***-- Additional Comments:***

***-- Notes:***

***-- This testbench has been automatically generated using types std\_logic and***

***-- std\_logic\_vector for the ports of the unit under test. Xilinx recommends***

***-- that these types always be used for the top-level I/O of a design in order***

***-- to guarantee that the testbench will bind correctly to the post-implementation***

***-- simulation model.***

***--------------------------------------------------------------------------------***

***LIBRARY ieee;***

***USE ieee.std\_logic\_1164.ALL;***

***-- Uncomment the following library declaration if using***

***-- arithmetic functions with Signed or Unsigned values***

***--USE ieee.numeric\_std.ALL;***

***ENTITY tb\_FullAdderUsingHalfAdders IS***

***END tb\_FullAdderUsingHalfAdders;***

***ARCHITECTURE behavior OF tb\_FullAdderUsingHalfAdders IS***

***-- Component Declaration for the Unit Under Test (UUT)***

***COMPONENT FullAdderUsingHalfAdders***

***PORT(***

***A : IN std\_logic;***

***B : IN std\_logic;***

***Cin : IN std\_logic;***

***Sum : OUT std\_logic;***

***Cout : OUT std\_logic***

***);***

***END COMPONENT;***

***--Inputs***

***signal A : std\_logic := '0';***

***signal B : std\_logic := '0';***

***signal Cin : std\_logic := '0';***

***--Outputs***

***signal Sum : std\_logic;***

***signal Cout : std\_logic;***

***-- No clocks detected in port list. Replace <clock> below with***

***-- appropriate port name***

***BEGIN***

***-- Instantiate the Unit Under Test (UUT)***

***uut: FullAdderUsingHalfAdders PORT MAP (***

***A => A,***

***B => B,***

***Cin => Cin,***

***Sum => Sum,***

***Cout => Cout***

***);***

***-- Stimulus process***

***stim\_proc: process***

***begin***

***A <= '0'; B <= '0'; Cin <= '0';***

***wait for 10 ns;***

***A <= '0'; B <= '0'; Cin <= '1';***

***wait for 10 ns;***

***A <= '0'; B <= '1'; Cin <= '0';***

***wait for 10 ns;***

***A <= '0'; B <= '1'; Cin <= '1';***

***wait for 10 ns;***

***A <= '1'; B <= '0'; Cin <= '0';***

***wait for 10 ns;***

***A <= '1'; B <= '0'; Cin <= '1';***

***wait for 10 ns;***

***A <= '1'; B <= '1'; Cin <= '0';***

***wait for 10 ns;***

***A <= '1'; B <= '1'; Cin <= '1';***

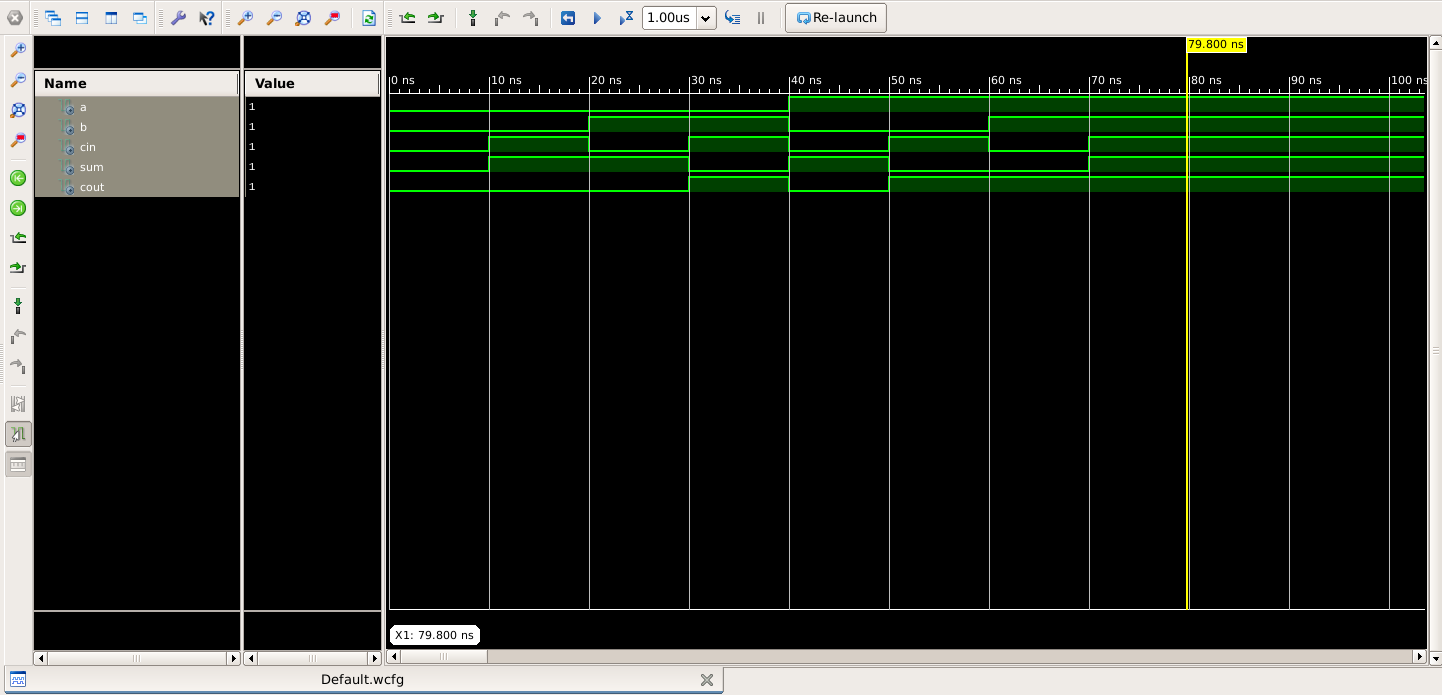
***wait for 10 ns;***

***wait;***

***end process;***

***END;***

***● Waveform:***

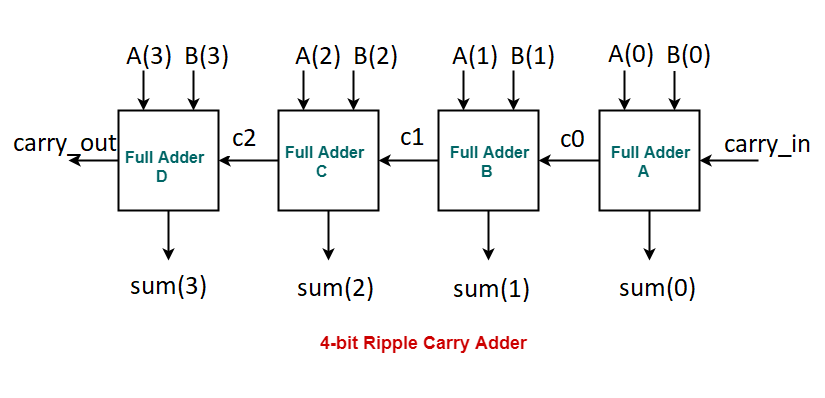


**Question 3**

**Design a VHDL model for a four bit full adder using full adder components .**

**Aim : The aim of this design is to create a VHDL model of a 4-bit full adder using multiple full adder components. This model will allow for the addition of two 4-bit binary numbers along with a carry-in bit, producing a 4-bit sum and a carry-out bit.**

**Circuit Diagram :**

******

***● VHDL Code:***

***----------------------------------------------------------------------------------***

***-- Company:***

***-- Engineer:***

***--***

***-- Create Date: 15:19:53 09/22/2024***

***-- Design Name:***

***-- Module Name: FourBitFullAdder - Structural***

***-- Project Name:***

***-- Target Devices:***

***-- Tool versions:***

***-- Description:***

***--***

***-- Dependencies:***

***--***

***-- Revision:***

***-- Revision 0.01 - File Created***

***-- Additional Comments:***

***--***

***----------------------------------------------------------------------------------***

***library IEEE;***

***use IEEE.STD\_LOGIC\_1164.ALL;***

***-- Uncomment the following library declaration if using***

***-- arithmetic functions with Signed or Unsigned values***

***--use IEEE.NUMERIC\_STD.ALL;***

***-- Uncomment the following library declaration if instantiating***

***-- any Xilinx primitives in this code.***

***--library UNISIM;***

***--use UNISIM.VComponents.all;***

***entity FourBitFullAdder is***

***Port (A : in STD\_LOGIC\_VECTOR(3 downto 0);***

***B : in STD\_LOGIC\_VECTOR(3 downto 0);***

***Cin : in STD\_LOGIC;***

***Sum : out STD\_LOGIC\_VECTOR(3 downto 0);***

***Cout : out STD\_LOGIC);***

***end FourBitFullAdder;***

***architecture Structural of FourBitFullAdder is***

***signal C : STD\_LOGIC\_VECTOR(3 downto 0);***

***component FullAdder is***

***Port ( A : in STD\_LOGIC;***

***B : in STD\_LOGIC;***

***Cin : in STD\_LOGIC;***

***Sum : out STD\_LOGIC;***

***Cout : out STD\_LOGIC);***

***end component;***

***begin***

***FA0: FullAdder port map (A(0), B(0), Cin, Sum(0) , C(0));***

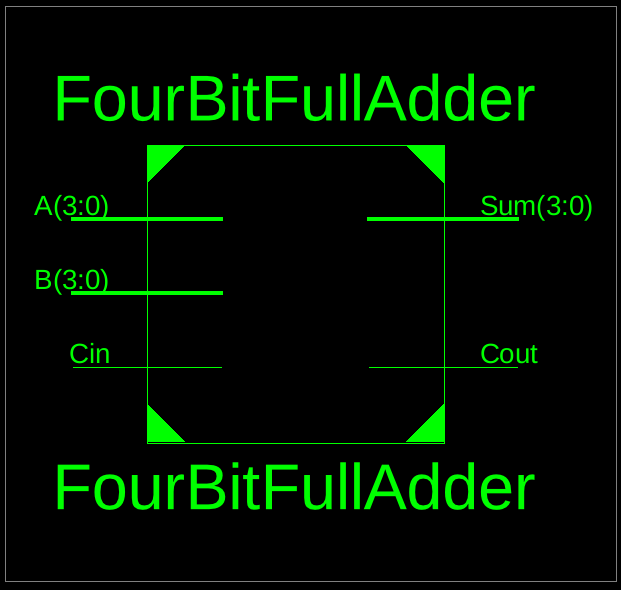
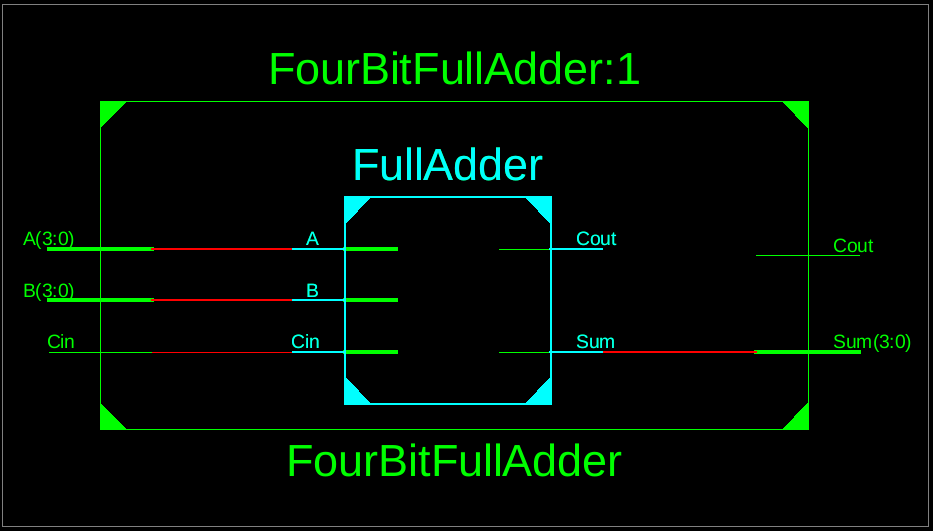
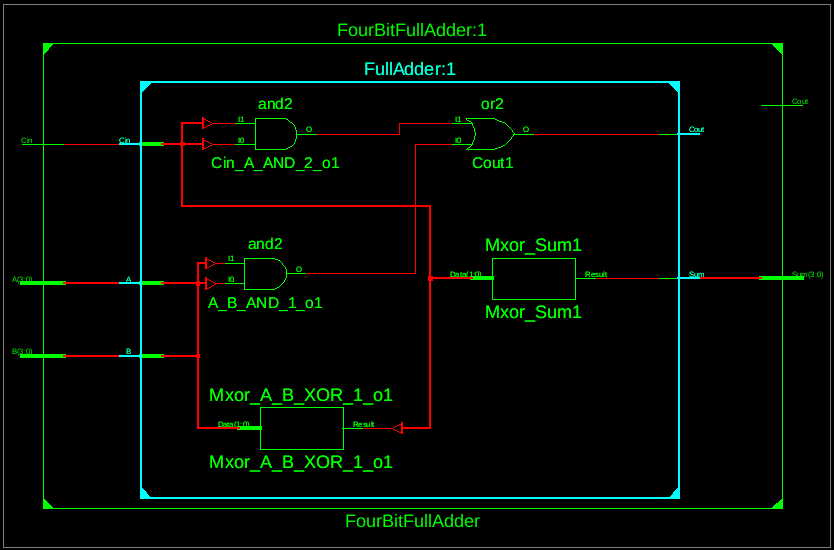
***FA1: FullAdder port map (A(1), B(1), C(0), Sum(1), C(1));***

***FA2: FullAdder port map (A(2), B(2), C(1), Sum(2), C(2));***

***FA3: FullAdder port map (A(3), B(3), C(2), Sum(3), Cout);***

***end Structural;***

***● RTL Diagram:***

***● Testbench Code:   
--------------------------------------------------------------------------------***

***-- Company:***

***-- Engineer:***

***-- Create Date: 15:32:13 09/22/2024***

***-- Design Name:***

***-- Module Name: /home/ise/Assignment1/tb\_FourBitFullAdder.vhd***

***-- Project Name: Assignment1***

***-- Target Device:***

***-- Tool versions:***

***-- Description:***

***-- VHDL Test Bench Created by ISE for module: FourBitFullAdder***

***-- Dependencies:***

***-- Revision:***

***-- Revision 0.01 - File Created***

***-- Additional Comments:***

***-- Notes:***

***-- This testbench has been automatically generated using types std\_logic and***

***-- std\_logic\_vector for the ports of the unit under test. Xilinx recommends***

***-- that these types always be used for the top-level I/O of a design in order***

***-- to guarantee that the testbench will bind correctly to the post-implementation***

***-- simulation model.***

***--------------------------------------------------------------------------------***

***LIBRARY ieee;***

***USE ieee.std\_logic\_1164.ALL;***

***-- Uncomment the following library declaration if using***

***-- arithmetic functions with Signed or Unsigned values***

***--USE ieee.numeric\_std.ALL;***

***ENTITY tb\_FourBitFullAdder IS***

***END tb\_FourBitFullAdder;***

***ARCHITECTURE behavior OF tb\_FourBitFullAdder IS***

***-- Component Declaration for the Unit Under Test (UUT)***

***COMPONENT FourBitFullAdder***

***PORT(***

***A : IN std\_logic\_vector(3 downto 0);***

***B : IN std\_logic\_vector(3 downto 0);***

***Cin : IN std\_logic;***

***Sum : OUT std\_logic\_vector(3 downto 0);***

***Cout : OUT std\_logic***

***);***

***END COMPONENT;***

***--Inputs***

***signal A : std\_logic\_vector(3 downto 0) := (others => '0');***

***signal B : std\_logic\_vector(3 downto 0) := (others => '0');***

***signal Cin : std\_logic := '0';***

***--Outputs***

***signal Sum : std\_logic\_vector(3 downto 0);***

***signal Cout : std\_logic;***

***-- No clocks detected in port list. Replace <clock> below with***

***-- appropriate port name***

***BEGIN***

***-- Instantiate the Unit Under Test (UUT)***

***uut: FourBitFullAdder PORT MAP (***

***A => A,***

***B => B,***

***Cin => Cin,***

***Sum => Sum,***

***Cout => Cout***

***);***

***-- Stimulus process***

***stim\_proc: process***

***begin***

***A <= "0000"; B <= "0000"; Cin <= '0';***

***wait for 10 ns;***

***A <= "0001"; B <= "0001"; Cin <= '0';***

***wait for 10 ns;***

***A <= "0010"; B <= "0011"; Cin <= '1';***

***wait for 10 ns;***

***A <= "1111"; B <= "0001"; Cin <= '1';***

***wait for 10 ns;***

***A <= "1001"; B <= "0110"; Cin <= '0';***

***wait for 10 ns;***

***A <= "1010"; B <= "1011"; Cin <= '1';***

***wait for 10 ns;***

***A <= "1111"; B <= "1111"; Cin <= '1';***

***wait for 10 ns;***

***A <= "0000"; B <= "1111"; Cin <= '0';***

***wait for 10 ns;***

***A <= "1111"; B <= "0000"; Cin <= '1';***

***wait for 10 ns;***

***A <= "0101"; B <= "1010"; Cin <= '0';***

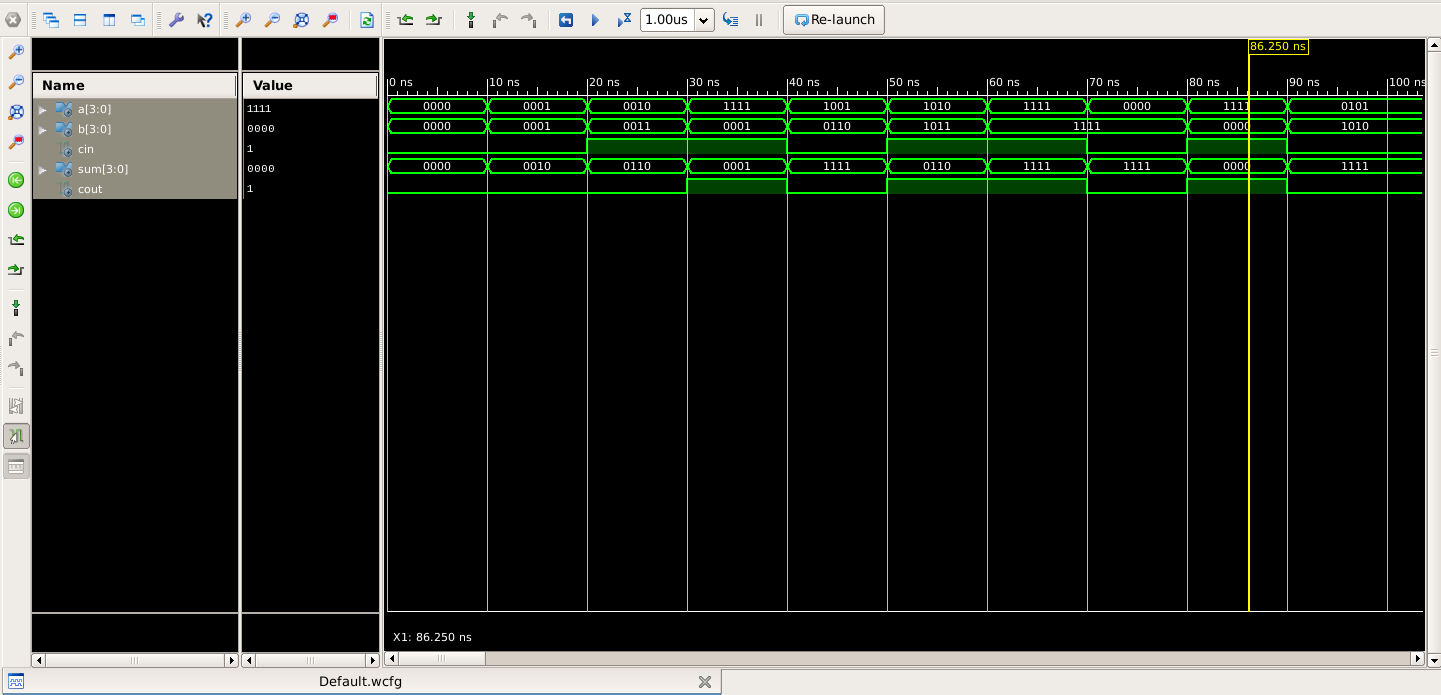
***wait for 10 ns;***

***wait;***

***end process;***

***END;***

***● Waveform:***



**Question 4**

**Design a VHDL Model for a four bit adder subtractor using :-**

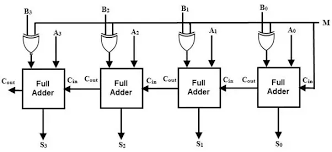
**• Use 2’s compliment subtraction.**

**• Use a control bit to decide addition/Subtraction.**

**• Make sure to take care of the carry out in case of addition/subtraction.**

**Aim : The aim of this design is to create a VHDL model for a 4-bit adder-subtractor that can perform both addition and subtraction using 2's complement. A control bit will determine whether to add or subtract, and the design will ensure proper handling of carry-out in both operations.**

**Circuit Diagram :**

******

***● VHDL Code:***

***----------------------------------------------------------------------------------***

***-- Company:***

***-- Engineer:***

***--***

***-- Create Date: 17:53:40 09/22/2024***

***-- Design Name:***

***-- Module Name: FourBitAdderSubtractor - Behavioral***

***-- Project Name:***

***-- Target Devices:***

***-- Tool versions:***

***-- Description:***

***--***

***-- Dependencies:***

***--***

***-- Revision:***

***-- Revision 0.01 - File Created***

***-- Additional Comments:***

***--***

***----------------------------------------------------------------------------------***

***library IEEE;***

***use IEEE.STD\_LOGIC\_1164.ALL;***

***-- Uncomment the following library declaration if using***

***-- arithmetic functions with Signed or Unsigned values***

***--use IEEE.NUMERIC\_STD.ALL;***

***-- Uncomment the following library declaration if instantiating***

***-- any Xilinx primitives in this code.***

***--library UNISIM;***

***--use UNISIM.VComponents.all;***

***entity FourBitAdderSubtractor is***

***Port ( A : in STD\_LOGIC\_VECTOR(3 downto 0);***

***B : in STD\_LOGIC\_VECTOR(3 downto 0);***

***Control : in STD\_LOGIC;***

***Sum : out STD\_LOGIC\_VECTOR(3 downto 0);***

***Cout : out STD\_LOGIC);***

***end FourBitAdderSubtractor;***

***architecture Behavioral of FourBitAdderSubtractor is***

***signal B\_temp : STD\_LOGIC\_VECTOR(3 downto 0);***

***signal Cin : STD\_LOGIC;***

***signal C : STD\_LOGIC\_VECTOR(3 downto 0);***

***component FullAdder is***

***Port ( A : in STD\_LOGIC;***

***B : in STD\_LOGIC;***

***Cin : in STD\_LOGIC;***

***Sum : out STD\_LOGIC;***

***Cout : out STD\_LOGIC);***

***end component;***

***begin***

***B\_temp <= B XOR (Control, Control, Control, Control); -- 2's complement if Control is 1***

***Cin <= Control; -- Carry-in for subtraction***

***FA0: FullAdder port map (A(0), B\_temp(0), Cin, Sum(0), C(0));***

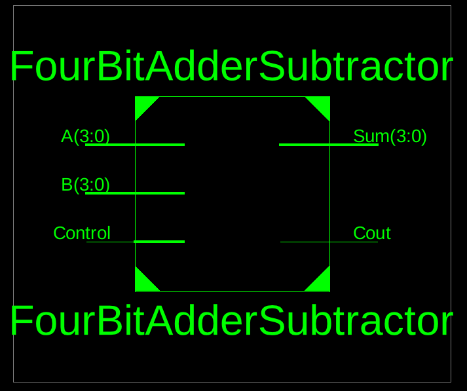
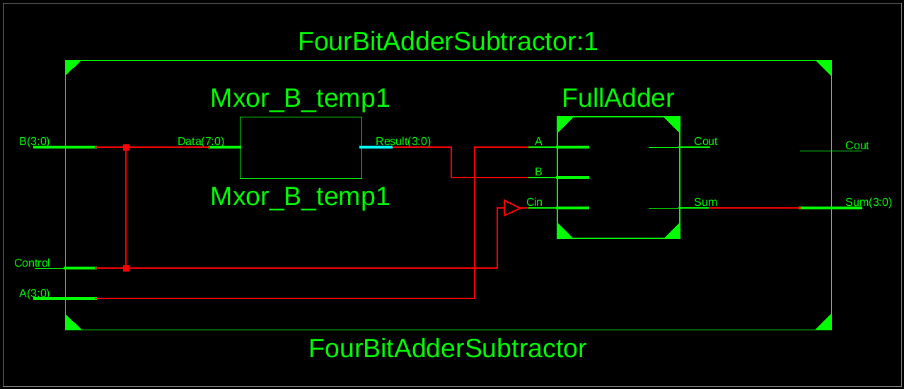
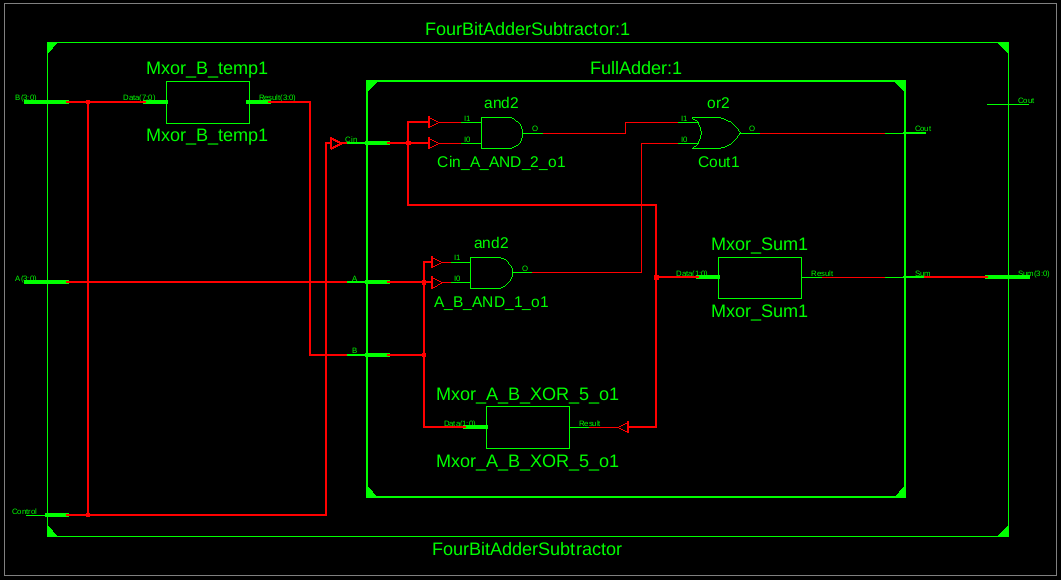
***FA1: FullAdder port map (A(1), B\_temp(1), C(0), Sum(1), C(1));***

***FA2: FullAdder port map (A(2), B\_temp(2), C(1), Sum(2), C(2));***

***FA3: FullAdder port map (A(3), B\_temp(3), C(2), Sum(3), Cout);***

***end Behavioral;***

***● RTL Diagram:***

***● Testbench Code:   
--------------------------------------------------------------------------------***

***-- Company:***

***-- Engineer:***

***--***

***-- Create Date: 18:10:01 09/22/2024***

***-- Design Name:***

***-- Module Name: /home/ise/Assignment1/tb\_FourBitAdderSubtractor.vhd***

***-- Project Name: Assignment1***

***-- Target Device:***

***-- Tool versions:***

***-- Description:***

***--***

***-- VHDL Test Bench Created by ISE for module: FourBitAdderSubtractor***

***--***

***-- Dependencies:***

***--***

***-- Revision:***

***-- Revision 0.01 - File Created***

***-- Additional Comments:***

***--***

***-- Notes:***

***-- This testbench has been automatically generated using types std\_logic and***

***-- std\_logic\_vector for the ports of the unit under test. Xilinx recommends***

***-- that these types always be used for the top-level I/O of a design in order***

***-- to guarantee that the testbench will bind correctly to the post-implementation***

***-- simulation model.***

***--------------------------------------------------------------------------------***

***LIBRARY ieee;***

***USE ieee.std\_logic\_1164.ALL;***

***-- Uncomment the following library declaration if using***

***-- arithmetic functions with Signed or Unsigned values***

***--USE ieee.numeric\_std.ALL;***

***ENTITY tb\_FourBitAdderSubtractor IS***

***END tb\_FourBitAdderSubtractor;***

***ARCHITECTURE behavior OF tb\_FourBitAdderSubtractor IS***

***-- Component Declaration for the Unit Under Test (UUT)***

***COMPONENT FourBitAdderSubtractor***

***PORT(***

***A : IN std\_logic\_vector(3 downto 0);***

***B : IN std\_logic\_vector(3 downto 0);***

***Control : IN std\_logic;***

***Sum : OUT std\_logic\_vector(3 downto 0);***

***Cout : OUT std\_logic***

***);***

***END COMPONENT;***

***--Inputs***

***signal A : std\_logic\_vector(3 downto 0) := (others => '0');***

***signal B : std\_logic\_vector(3 downto 0) := (others => '0');***

***signal Control : std\_logic := '0';***

***--Outputs***

***signal Sum : std\_logic\_vector(3 downto 0);***

***signal Cout : std\_logic;***

***-- No clocks detected in port list. Replace <clock> below with***

***-- appropriate port name***

***BEGIN***

***-- Instantiate the Unit Under Test (UUT)***

***uut: FourBitAdderSubtractor PORT MAP (***

***A => A,***

***B => B,***

***Control => Control,***

***Sum => Sum,***

***Cout => Cout***

***);***

***-- Stimulus process***

***stim\_proc: process***

***begin***

***-- addition: A = 5, B = 3***

***A <= "0101";***

***B <= "0011";***

***Control <= '0'; -- Addition***

***wait for 10 ns;***

***-- subtraction: A = 5 , B = 3***

***A <= "0101";***

***B <= "0011";***

***Control <= '1';***

***wait for 10 ns;***

***-- subtraction with borrow: A = 3 , B = 5***

***A <= "0011";***

***B <= "0101";***

***Control <= '1';***

***wait for 10 ns;***

***-- addition of two maximum values: A = 15 , B = 15***

***A <= "1111";***

***B <= "1111";***

***Control <= '0';***

***wait for 10 ns;***

***-- subtraction resulting in zero: A = 7 , B = 7***

***A <= "0111";***

***B <= "0111";***

***Control <= '1';***

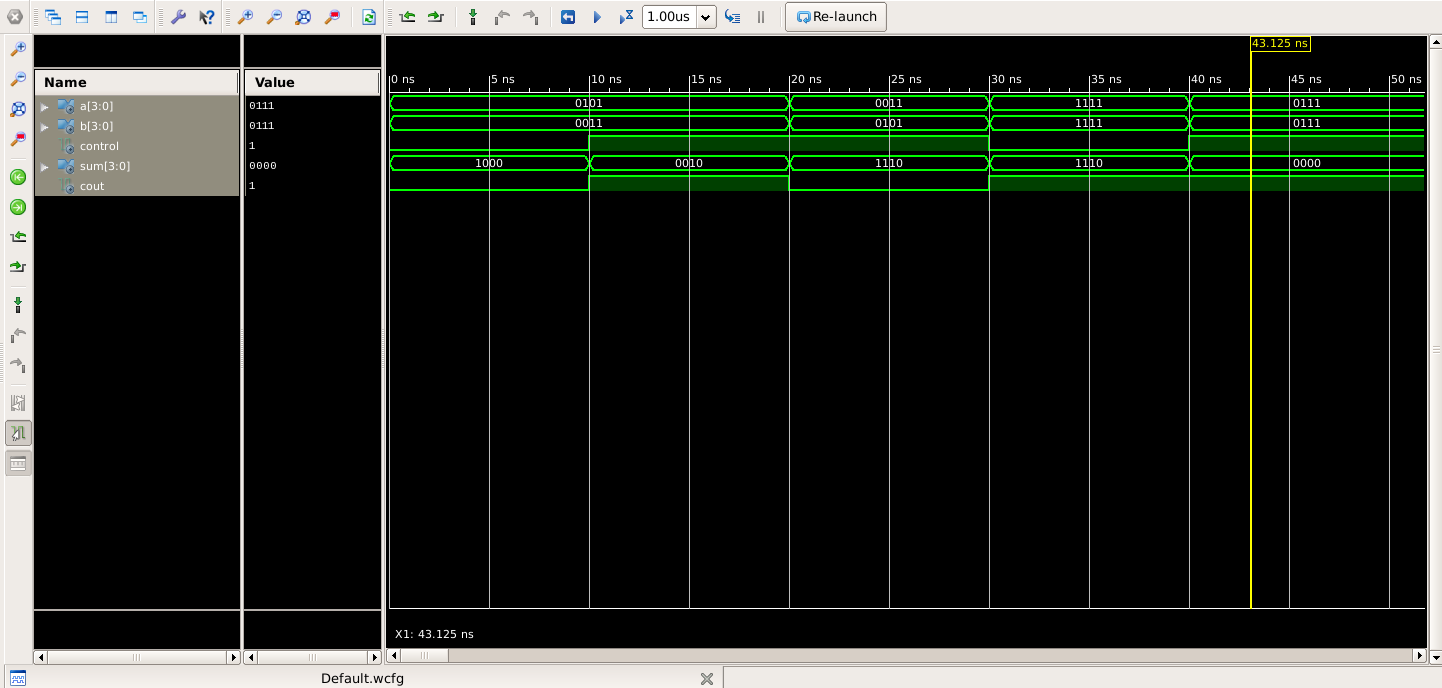
***wait for 10 ns;***

***wait;***

***end process;***

***END;***

***● Waveform:***

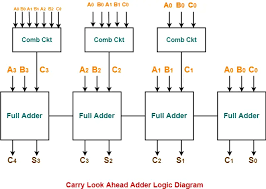


**Question 5**

**Design a VHDL model for a 4 bit Carry Lookahead Adder using Full adder components. Generate all the carries in parallel except the final one.**

**Aim : The aim is to design a 4-bit Carry Lookahead Adder (CLA) using full adder components in VHDL. This design should efficiently compute the sum of two 4-bit binary numbers (A and B) along with a carry-in (Cin) by generating all intermediate carry signals in parallel, except for the final carry output (Cout). The design leverages the carry lookahead principle to speed up the addition process compared to traditional ripple-carry adders.**

**Circuit Diagram :**

******

***● VHDL Code:***

***----------------------------------------------------------------------------------***

***-- Company:***

***-- Engineer:***

***--***

***-- Create Date: 18:27:06 09/22/2024***

***-- Design Name:***

***-- Module Name: CarryLookaheadAdder - Structural***

***-- Project Name:***

***-- Target Devices:***

***-- Tool versions:***

***-- Description:***

***--***

***-- Dependencies:***

***--***

***-- Revision:***

***-- Revision 0.01 - File Created***

***-- Additional Comments:***

***--***

***----------------------------------------------------------------------------------***

***library IEEE;***

***use IEEE.STD\_LOGIC\_1164.ALL;***

***-- Uncomment the following library declaration if using***

***-- arithmetic functions with Signed or Unsigned values***

***--use IEEE.NUMERIC\_STD.ALL;***

***-- Uncomment the following library declaration if instantiating***

***-- any Xilinx primitives in this code.***

***--library UNISIM;***

***--use UNISIM.VComponents.all;***

***entity CarryLookaheadAdder is***

***Port ( A : in STD\_LOGIC\_VECTOR(3 downto 0);***

***B : in STD\_LOGIC\_VECTOR(3 downto 0);***

***Cin : in STD\_LOGIC;***

***Sum : out STD\_LOGIC\_VECTOR(3 downto 0);***

***Cout : out STD\_LOGIC);***

***end CarryLookaheadAdder;***

***architecture Structural of CarryLookaheadAdder is***

***signal P, G, C : STD\_LOGIC\_VECTOR(3 downto 0);***

***component FullAdder is***

***Port ( A : in STD\_LOGIC;***

***B : in STD\_LOGIC;***

***Cin : in STD\_LOGIC;***

***Sum : out STD\_LOGIC;***

***Cout : out STD\_LOGIC);***

***end component;***

***begin***

***P <= A XOR B;***

***G <= A AND B;***

***C(0) <= G(0) OR (P(0) AND Cin);***

***C(1) <= G(1) OR (P(1) AND G(0)) OR (P(1) AND P(0) AND Cin);***

***C(2) <= G(2) OR (P(2) AND G(1)) OR (P(2) AND P(1) AND G(0)) OR (P(2) AND P(1) AND P(0) AND Cin);***

***C(3) <= G(3) OR (P(3) AND G(2)) OR (P(3) AND P(2) AND G(1)) OR (P(3) AND P(2) AND P(1) AND G(0)) OR (P(3) AND P(2) AND P(1) AND P(0) AND Cin);***

***FA0: FullAdder port map (A(0), B(0), Cin, Sum(0), C(0));***

***FA1: FullAdder port map (A(1), B(1), C(0), Sum(1), C(1));***

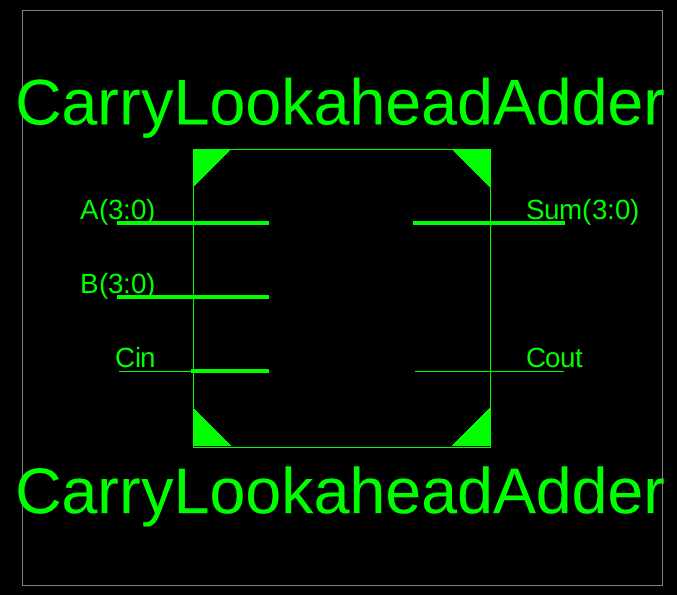
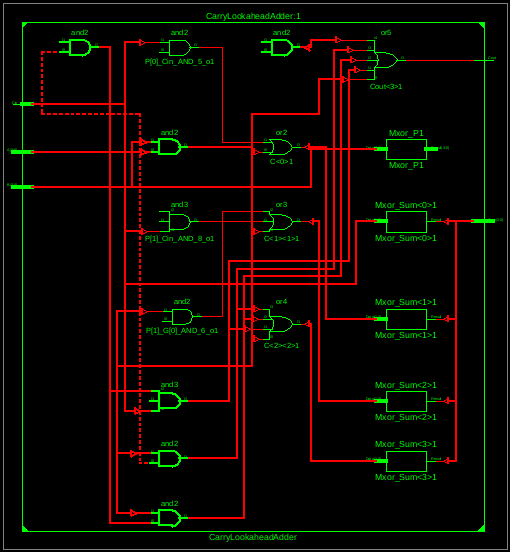
***FA2: FullAdder port map (A(2), B(2), C(1), Sum(2), C(2));***

***FA3: FullAdder port map (A(3), B(3), C(2), Sum(3), C(3));***

***Cout <= C(3);***

***end Structural;***

***● RTL Diagram:***

***● Testbench Code:   
--------------------------------------------------------------------------------***

***-- Company:***

***-- Engineer:***

***-- Create Date: 18:44:17 09/22/2024***

***-- Design Name:***

***-- Module Name: /home/ise/Assignment1/tb\_CarryLookaheadAdder.vhd***

***-- Project Name: Assignment1***

***-- Target Device:***

***-- Tool versions:***

***-- Description:***

***-- VHDL Test Bench Created by ISE for module: CarryLookaheadAdder***

***-- Dependencies:***

***-- Revision:***

***-- Revision 0.01 - File Created***

***-- Additional Comments:***

***-- Notes:***

***-- This testbench has been automatically generated using types std\_logic and***

***-- std\_logic\_vector for the ports of the unit under test. Xilinx recommends***

***-- that these types always be used for the top-level I/O of a design in order***

***-- to guarantee that the testbench will bind correctly to the post-implementation***

***-- simulation model.***

***--------------------------------------------------------------------------------***

***LIBRARY ieee;***

***USE ieee.std\_logic\_1164.ALL;***

***-- Uncomment the following library declaration if using***

***-- arithmetic functions with Signed or Unsigned values***

***--USE ieee.numeric\_std.ALL;***

***ENTITY tb\_CarryLookaheadAdder IS***

***END tb\_CarryLookaheadAdder;***

***ARCHITECTURE behavior OF tb\_CarryLookaheadAdder IS***

***-- Component Declaration for the Unit Under Test (UUT)***

***COMPONENT CarryLookaheadAdder***

***PORT(***

***A : IN std\_logic\_vector(3 downto 0);***

***B : IN std\_logic\_vector(3 downto 0);***

***Cin : IN std\_logic;***

***Sum : OUT std\_logic\_vector(3 downto 0);***

***Cout : OUT std\_logic***

***);***

***END COMPONENT;***

***--Inputs***

***signal A : std\_logic\_vector(3 downto 0) := (others => '0');***

***signal B : std\_logic\_vector(3 downto 0) := (others => '0');***

***signal Cin : std\_logic := '0';***

***--Outputs***

***signal Sum : std\_logic\_vector(3 downto 0);***

***signal Cout : std\_logic;***

***-- No clocks detected in port list. Replace <clock> below with***

***-- appropriate port name***

***BEGIN***

***-- Instantiate the Unit Under Test (UUT)***

***uut: CarryLookaheadAdder PORT MAP (***

***A => A,***

***B => B,***

***Cin => Cin,***

***Sum => Sum,***

***Cout => Cout***

***);***

***-- Stimulus process***

***stim\_proc: process***

***begin***

***A <= "0001"; B <= "0010"; Cin <= '0';***

***wait for 10 ns;***

***A <= "0101"; B <= "0011"; Cin <= '1';***

***wait for 10 ns;***

***A <= "1111"; B <= "1111"; Cin <= '1';***

***wait for 10 ns;***

***A <= "0000"; B <= "0000"; Cin <= '0';***

***wait for 10 ns;***

***A <= "0110"; B <= "1001"; Cin <= '0';***

***wait for 10 ns;***

***A <= "1000"; B <= "1000"; Cin <= '1';***

***wait for 10 ns;***

***A <= "1101"; B <= "0101"; Cin <= '0';***

***wait for 10 ns;***

***A <= "0011"; B <= "0111"; Cin <= '1';***

***wait for 10 ns;***

***A <= "1110"; B <= "0001"; Cin <= '0';***

***wait for 10 ns;***

***A <= "0001"; B <= "0001"; Cin <= '1';***

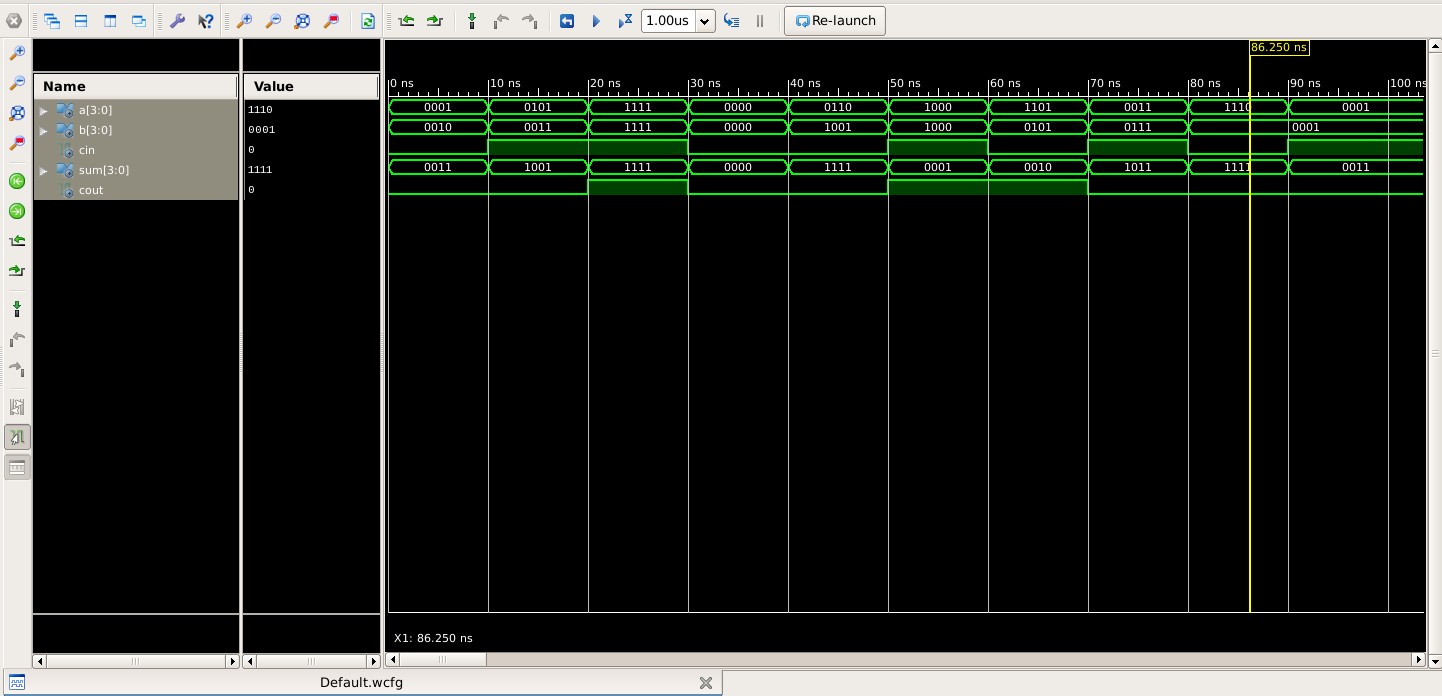
***wait for 10 ns;***

***wait;***

***end process;***

***END;***

***● Waveform:***



**Note For The Teacher**

***Dear Sir,***

***The Above File is Purely Made by Me (IJ Roy) I had taken some help from geeks for geeks for writing the test bench code hence it may differ a little from the one you taught in class but I found this method more easy hence I am using it***

**Proof That it Was Not Copied**

***You can check My Github Account***

https://github.com/ij-roy/Semester-3/tree/main/COA/Assignment%201

***I had uploaded Every picture and code used above and you can also check the Date of Upload***