```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity ripple_adder_16bit is
    Port ( X : in  STD_LOGIC_VECTOR (15 downto 0);
           Y : in  STD_LOGIC_VECTOR (15 downto 0);
                       Cin : STD_LOGIC;
           S : out  STD_LOGIC_VECTOR (15 downto 0);
           Cout : out  STD_LOGIC);
end ripple_adder_16bit;

architecture Behavioral of ripple_adder_16bit is
--components
        --full adder component
        COMPONENT full_adder
        PORT(
                        X : in  STD_LOGIC;
                        Y : in  STD_LOGIC;
                        cin : in  STD_LOGIC;
                        cout : out  STD_LOGIC;
                        S : out  STD_LOGIC
                        );
        END COMPONENT;

--signals
signal carry_0, carry_1, carry_2, carry_3, carry_4, carry_5, carry_6, carry_7, carry_8, carry_9, carry_10,
carry_11, carry_12, carry_13, carry_14 : STD_LOGIC;

begin
--port maps
        --full adder 0
        adder0: full_adder PORT MAP(
                                X => X(0),
                                Y => Y(0),
                                cin => Cin,
                                cout => carry_0,
                                S => S(0)
        );

        --full adder 1
        adder1: full_adder PORT MAP(
                                X => X(1),
                                Y => Y(1),
                                cin => carry_0,
                                cout => carry_1,
                                S => S(1)
        );

        --full adder 2
        adder2: full_adder PORT MAP(
                                X => X(2),
                                Y => Y(2),
                                cin => carry_1,
                                cout => carry_2,
                                S => S(2)
        );

        --full adder 3
        adder3: full_adder PORT MAP(
                                X => X(3),
                                Y => Y(3),
                                cin => carry_2,
                                cout => carry_3,
                                S => S(3)
        );

        --full adder 4
        adder4: full_adder PORT MAP(
                                X => X(4),
                                Y => Y(4),
                                cin => carry_3,
                                cout => carry_4,
                                S => S(4)
        );

        --full adder 5
        adder5: full_adder PORT MAP(
```

```
                        X => X(5),
                        Y => Y(5),
                        cin => carry_4,
                        cout => carry_5,
                        S => S(5)
);

--full adder 6
adder6: full_adder PORT MAP(
                        X => X(6),
                        Y => Y(6),
                        cin => carry_5,
                        cout => carry_6,
                        S => S(6)
);

--full adder 7
adder7: full_adder PORT MAP(
                        X => X(7),
                        Y => Y(7),
                        cin => carry_6,
                        cout => carry_7,
                        S => S(7)
);

--full adder 8
adder8: full_adder PORT MAP(
                        X => X(8),
                        Y => Y(8),
                        cin => carry_7,
                        cout => carry_8,
                        S => S(8)
);

--full adder 9
adder9: full_adder PORT MAP(
                        X => X(9),
                        Y => Y(9),
                        cin => carry_8,
                        cout => carry_9,
                        S => S(9)
);

--full adder 10
adder10: full_adder PORT MAP(
                        X => X(10),
                        Y => Y(10),
                        cin => carry_9,
                        cout => carry_10,
                        S => S(10)
);

--full adder 11
adder11: full_adder PORT MAP(
                        X => X(11),
                        Y => Y(11),
                        cin => carry_10,
                        cout => carry_11,
                        S => S(11)
);

--full adder 12
adder12: full_adder PORT MAP(
                        X => X(12),
                        Y => Y(12),
                        cin => carry_11,
                        cout => carry_12,
                        S => S(12)
);

--full adder 13
adder13: full_adder PORT MAP(
                        X => X(13),
                        Y => Y(13),
                        cin => carry_12,
                        cout => carry_13,
                        S => S(13)
```

```
        );

        --full adder 14
        adder14: full_adder PORT MAP(
                                X => X(14),
                                Y => Y(14),
                                cin => carry_13,
                                cout => carry_14,
                                S => S(14)
        );

        --full adder 15LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY ripple_adder_tb IS
END ripple_adder_tb;

ARCHITECTURE behavior OF ripple_adder_tb IS

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

    COMPONENT ripple_adder_16bit
    PORT(
         X : IN  std_logic_vector(15 downto 0);
         Y : IN  std_logic_vector(15 downto 0);
         Cin : IN  std_logic;
         S : OUT  std_logic_vector(15 downto 0);
         Cout : OUT  std_logic
        );
    END COMPONENT;


   --Inputs
   signal X : std_logic_vector(15 downto 0) := (others => '0');
   signal Y : std_logic_vector(15 downto 0) := (others => '0');
   signal Cin : std_logic := '0';

        --Outputs
   signal S : std_logic_vector(15 downto 0);
   signal Cout : std_logic;
BEGIN

        -- Instantiate the Unit Under Test (UUT)
   uut: ripple_adder_16bit PORT MAP (
          X => X,
          Y => Y,
          Cin => Cin,
          S => S,
          Cout => Cout
        );

   -- Stimulus process
   stim_proc: process
   begin
      -- hold reset state for 100 ns.
      wait for 100 ns;

                X <= "0000000000000001";
                Y <= "0000000000000001";
                Cin <='0';

                wait for 20 ns;
                X <="1111111111111111";

                wait for 20 ns;
                Cin <= '1';


      wait;
   end process;

END;
        adder15: full_adder PORT MAP(
                                X => X(15),
                                Y => Y(15),
                                cin => carry_14,
                                cout => Cout,
```

```
                                S => S(15)
        );

end Behavioral;
```