# Lab 5

## Multipliers

CS1050 Computer Organization and Digital Design

RATHNAYAKE R.M.I.B
220526N

Group 41

## Introduction:

In this lab, we are going to implement a 2x2 multiplier. It will be done using pre-built FAs and basic logic gates. We can perform the 2x2 multiplier, which is the multiplication of two 2-bit numbers, as follows:

| | | | | |
|---|---|---|---|---|
| | | $b_1$ | $b_0$ | |
| | | $a_1$ | $a_0$ | $s_2$ – sum of 2nd FA |
| | $c_1$ | $b_1a_0$ | $b_0a_0$ | $c_2$ – carry out of 2nd FA |
| | $b_1a_1$ | $b_0a_1$ | | $s_1$ – sum of 1st FA |
| $c_2$ | $s_2$ | $s_1$ | $b_0a_1$ | $c_2$ – carry out 1st FA |
| Y(3) | Y(2) | Y(1) | Y(0) | Y - output |

After that we have to implement a 4x4 multiplier which is the multiplication of two 4-bit numbers, as follows:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | b3 | b2 | $b_1$ | $b_0$ | |
| | | | | a3 | a2 | $a_1$ | $a_0$ | |
| | | | | $b_3a_0$ | $b_2a_0$ | $b_1a_0$ | $b_0a_0$ | |
| | | | $b_3a_1$ | $b_2a_1$ | $b_1a1$ | $b_0a_1$ | | |
| | | $b_3a_2$ | $b_2a_2$ | $b_1a_2$ | $b_0a_2$ | | | |
| | $b_3a_3$ | $b_2a_3$ | $b_1a_3$ | $b_0a_3$ | | | | |
| Y(7) | Y(6) | Y(5) | Y(4) | Y(3) | Y(2) | Y(1) | Y(0) | Y - output |

## 1. 2x2 multiplier

- **Design source file**

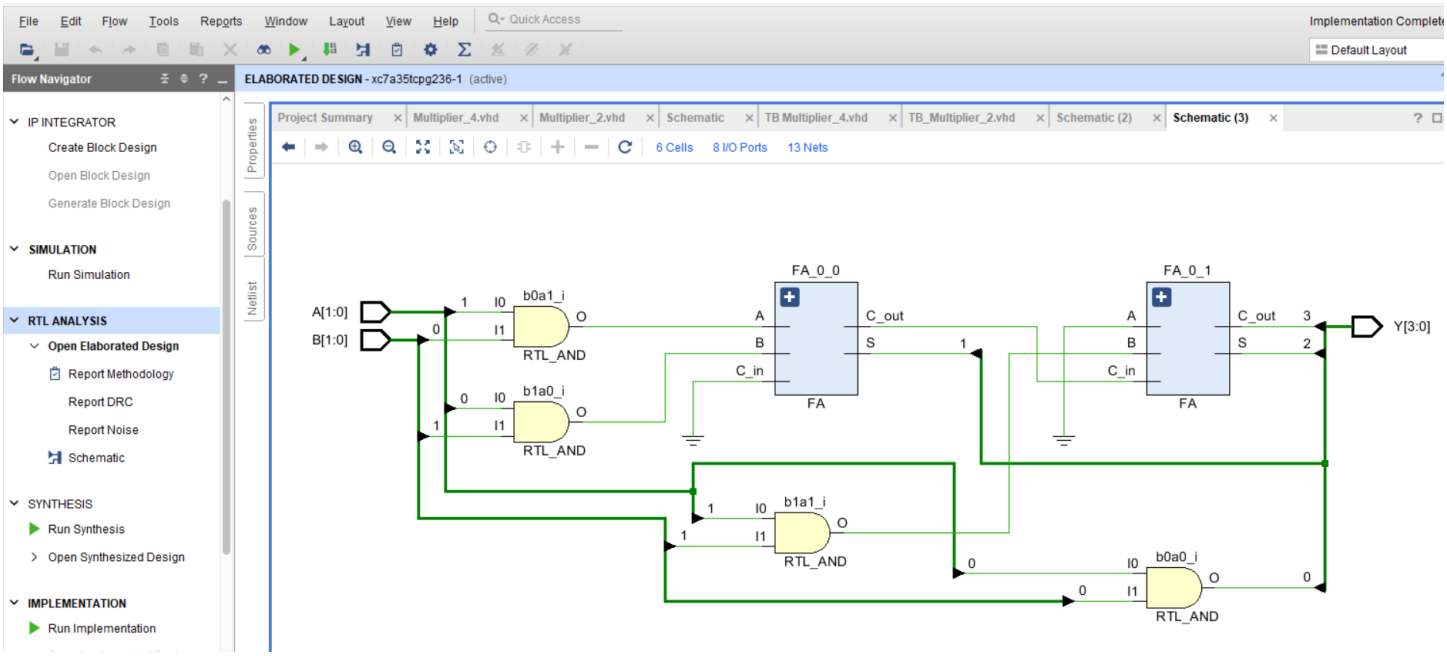
```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 02/27/2024 02:33:22 PM
-- Design Name:
-- Module Name: Multiplier_2 - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- 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 leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity Multiplier_2 is
    Port ( A : in STD_LOGIC_VECTOR (1 downto 0);
```

```
           B : in STD_LOGIC_VECTOR (1 downto 0);
           Y : out STD_LOGIC_VECTOR (3 downto 0));
end Multiplier_2;

architecture Behavioral of Multiplier_2 is

component FA
    port(
        A : in std_logic;
        B : in std_logic;
        C_in : in std_logic;
        S : out std_logic;
        C_out : out std_logic
);
end component;

signal b0a0, b0a1, b1a0, b1a1 : std_logic;
signal s_0_0, s_0_1, c_0_0, c_0_1 : std_logic;

begin

FA_0_0: FA port map(
    A => b0a1,
    B => b1a0,
    C_in => '0',
    S => s_0_0,
    C_out => c_0_0
);

FA_0_1: FA port map(
    A => '0',
    B => b1a1,
    C_in => c_0_0,
    S => s_0_1,
    C_out => c_0_1
);

b0a0 <= A(0) and B(0);
b1a0 <= A(0) and B(1);
b0a1 <= A(1) and B(0);
```

```
b1a1 <= A(1) and B(1);

Y(0) <= b0a0;
Y(1) <= s_0_0;
Y(2) <= s_0_1;
Y(3) <= c_0_1;

end Behavioral;
```

---

- **Elaborated design schematic**



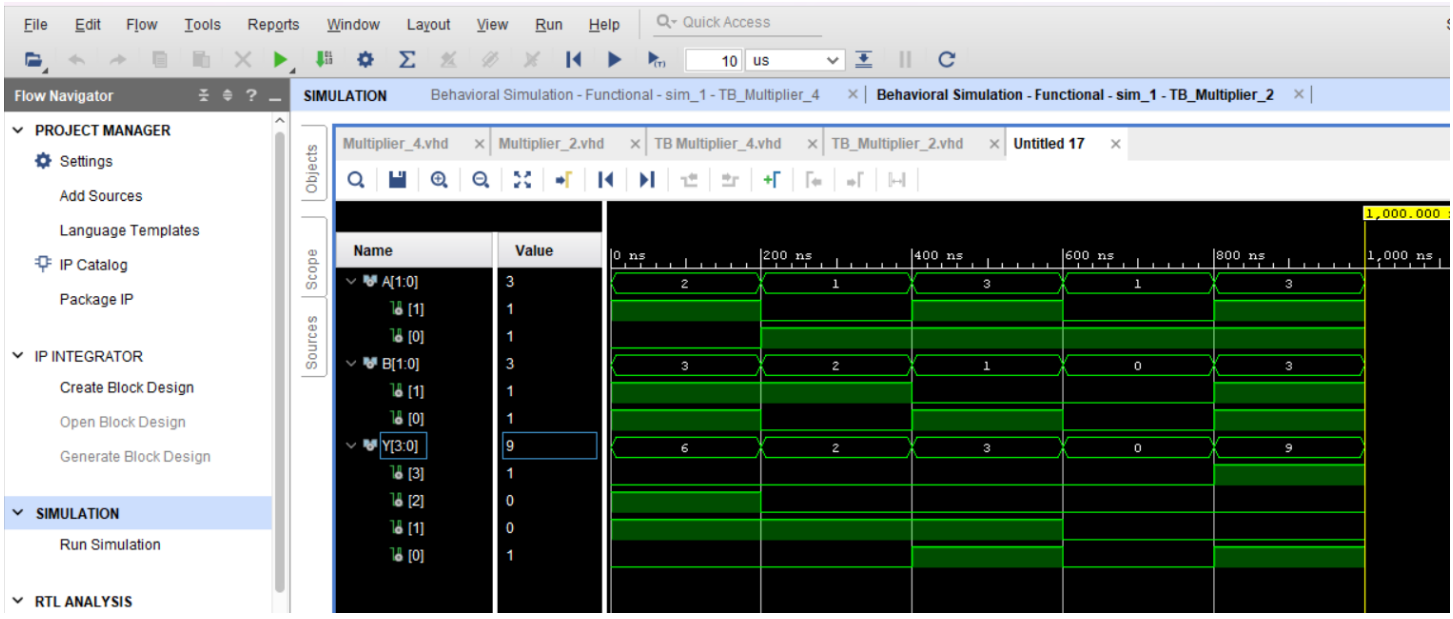

- **Simulation source file**

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 02/27/2024 03:25:58 PM
-- Design Name:
-- Module Name: TB_Multiplier_2 - 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 leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity TB_Multiplier_2 is
--  Port ( );
end TB_Multiplier_2;

architecture Behavioral of TB_Multiplier_2 is
component Multiplier_2 is
    Port ( A : in STD_LOGIC_VECTOR (1 downto 0);
           B : in STD_LOGIC_VECTOR (1 downto 0);
           Y : out STD_LOGIC_VECTOR (3 downto 0));
end component;

signal A,B : STD_LOGIC_VECTOR (1 downto 0);
signal Y :  STD_LOGIC_VECTOR (3 downto 0);

begin
```

```
UUT : Multiplier_2 port map(A,B,Y);
process begin
    A <= "10";
    B <= "11";
    wait for 200 ns;

    A <= "01";
    B <= "10";
    wait for 200 ns;

    A <= "11";
    B <= "01";
    wait for 200 ns;

    A <= "01";
    B <= "00";
    wait for 200 ns;

    A <= "11";
    B <= "11";
   wait ;
end process;
end Behavioral;
```

- **Time diagram**



## 2. 4x4 multiplier

- **Design source file**

```
----------------------------------------------------------------------------
--------
-- Company:
-- Engineer:
--
-- Create Date: 02/27/2024 03:52:00 PM
-- Design Name:
-- Module Name: Multiplier_4 - 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 leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity Multiplier_4 is
```

```
    Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
           B : in STD_LOGIC_VECTOR (3 downto 0);
           Y : out STD_LOGIC_VECTOR (7 downto 0));
end Multiplier_4;

architecture Behavioral of Multiplier_4 is
component FA
    port(
        A : in std_logic;
        B : in std_logic;
        C_in : in std_logic;
        S : out std_logic;
        C_out : out std_logic
);
end component;

signal b3a0, b2a0, b1a0, b0a0 : std_logic;
signal b3a1, b2a1, b1a1, b0a1 : std_logic;
signal b3a2, b2a2, b1a2, b0a2 : std_logic;
signal b3a3, b2a3, b1a3, b0a3 : std_logic;

signal s_0_0,c_0_0 : std_logic;
signal s_0_1,c_0_1 : std_logic;
signal s_0_2,c_0_2 : std_logic;
signal s_0_3,c_0_3 : std_logic;
signal s_1_0,c_1_0 : std_logic;
signal s_1_1,c_1_1 : std_logic;
signal s_1_2,c_1_2 : std_logic;
signal s_1_3,c_1_3 : std_logic;
signal s_2_0,c_2_0 : std_logic;
signal s_2_1,c_2_1 : std_logic;
signal s_2_2,c_2_2 : std_logic;
signal s_2_3,c_2_3 : std_logic;

begin
```

```
FA_0_0: FA port map(
    A => b1a0,
    B => b0a1,
    C_in => '0',
    S => s_0_0,
    C_out => c_0_0
);

FA_0_1: FA port map(
    A => b2a0,
    B => b1a1,
    C_in => c_0_0,
    S => s_0_1,
    C_out => c_0_1
);

FA_0_2: FA port map(
    A => b3a0,
    B => b2a1,
    C_in => c_0_1,
    S => s_0_2,
    C_out => c_0_2
);

FA_0_3: FA port map(
    A => '0',
    B => b3a1,
    C_in => c_0_2,
    S => s_0_3,
    C_out => c_0_3
);

FA_1_0: FA port map(
    A => s_0_1,
    B => b0a2,
    C_in => '0',
```

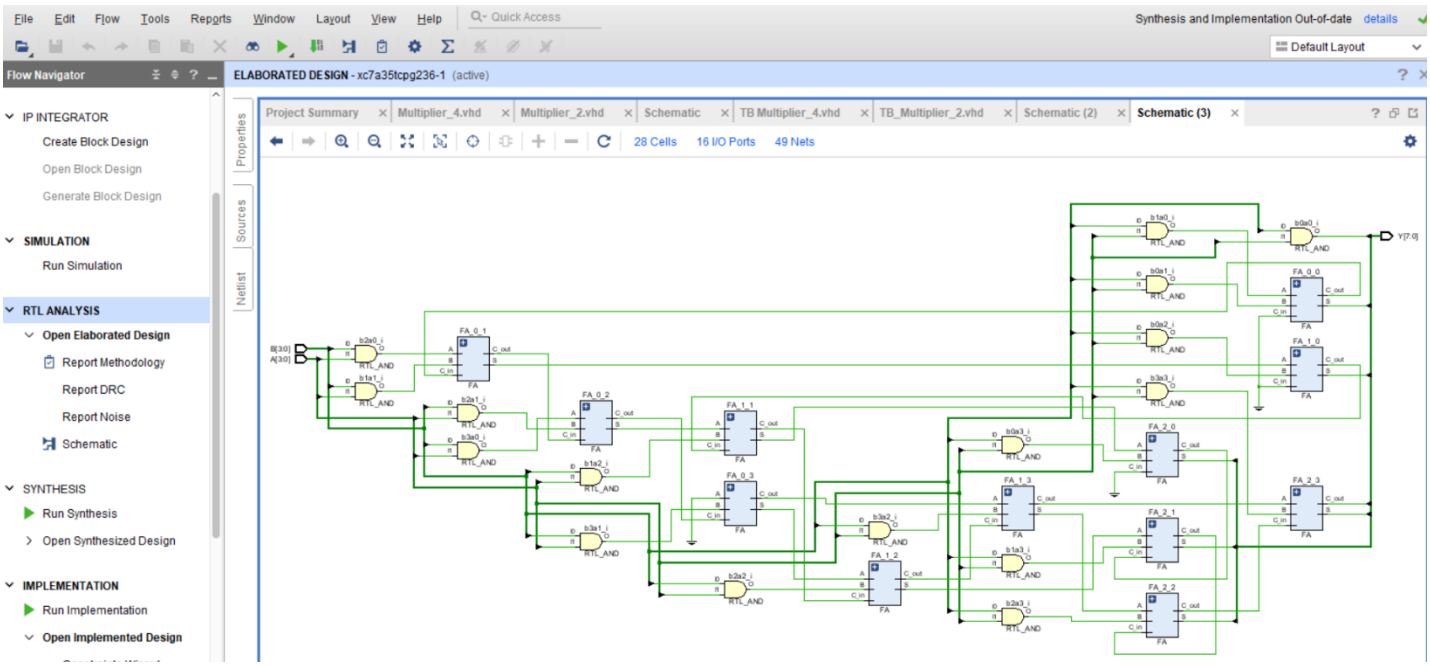
```
    S => s_1_0,
    C_out => c_1_0
);

FA_1_1: FA port map(
    A => s_0_2,
    B => b1a2,
    C_in => c_1_0,
    S => s_1_1,
    C_out => c_1_1
);

FA_1_2: FA port map(
    A => s_0_3,
    B => b2a2,
    C_in => c_1_1,
    S => s_1_2,
    C_out => c_1_2
);
FA_1_3: FA port map(
    A => c_0_3,
    B => b3a2,
    C_in => c_1_2,
    S => s_1_3,
    C_out => c_1_3
);

FA_2_0: FA port map(
    A => s_1_1,
    B => b0a3,
    C_in => '0',
    S => s_2_0,
    C_out => c_2_0
);

FA_2_1: FA port map(
```

```
    A => s_1_2,
    B => b1a3,
    C_in => c_2_0,
    S => s_2_1,  --Y(4)
    C_out => c_2_1
);

FA_2_2: FA port map(
    A => s_1_3,
    B => b2a3,
    C_in => c_2_1,
    S => s_2_2,
    C_out => c_2_2
);

FA_2_3: FA port map(
    A => c_1_3,
    B => b3a3,
    C_in => c_2_2,
    S => s_2_3,
    C_out => c_2_3
);

b0a0 <= B(0) and A(0);
b0a1 <= B(0) and A(1);
b0a2 <= B(0) and A(2);
b0a3 <= B(0) and A(3);

b1a0 <= B(1) and A(0);
b1a1 <= B(1) and A(1);
b1a2 <= B(1) and A(2);
b1a3 <= B(1) and A(3);

b2a0 <= B(2) and A(0);
b2a1 <= B(2) and A(1);
b2a2 <= B(2) and A(2);
```

```
b2a3 <= B(2) and A(3);

b3a0 <= B(3) and A(0);
b3a1 <= B(3) and A(1);
b3a2 <= B(3) and A(2);
b3a3 <= B(3) and A(3);

Y(0) <= b0a0;
Y(1) <= s_0_0;
Y(2) <= s_1_0;
Y(3) <= s_2_0;
Y(4) <= s_2_1;
Y(5) <= s_2_2;
Y(6) <= s_2_3;
Y(7) <= c_2_3;

end Behavioral;
```

---

- **Elaborated design schematic**



- **Simulation source file**

```
----------------------------------------------------------------------------
--------
-- Company:
-- Engineer:
--
-- Create Date: 02/27/2024 04:54:52 PM
-- Design Name:
-- Module Name: TB Multiplier_4 - 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 leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity TB_Multiplier_4 is
--  Port ( );
end TB_Multiplier_4;
```

```
architecture Behavioral of TB_Multiplier_4 is
component Multiplier_4 is
    Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
           B : in STD_LOGIC_VECTOR (3 downto 0);
           Y : out STD_LOGIC_VECTOR (7 downto 0));
end component;

signal A,B : STD_LOGIC_VECTOR (3 downto 0);
signal Y :  STD_LOGIC_VECTOR (7 downto 0);

begin
UUT : Multiplier_4 port map(A,B,Y);

process begin
    A <= "1110";
    B <= "0110";

    wait for 200 ns;

    A <= "1101";
    B <= "0101";

    wait for 200 ns;

    A <= "1111";
    B <= "1001";

    wait for 200 ns;

    A <= "1010";
    B <= "0101";

    wait for 200 ns;

    A <= "1111";
    B <= "1111";

    wait;
end process;
```

```
end Behavioral;
```

- **Time diagram**