Lab 1-3
And Gate, Half Adder, and Full Gate

Junseok Oh, Shiddharath Patel

October 24, 2022

Lab 1 AND Gate:

Purpose:

The purpose of this experiment is to get familiar with Vivado Software, and. With the goal being to understand the characteristics of AND gates. AND gates are hardware implementations of Boolean operations, they are represented on a diagram by gate symbols, and their characteristics can be described by truth tables, Boolean equations, and timing waveforms. AND gates can have two or more inputs, but only one output.

Lab 2 Half Adder:

Purpose:

The goal of this project is to use VHDL to design a half-adder using Vivado. One AND gate and one XOR gate make up the half-adder. It computes the total of the two inputs by passing them through the XOR gate, and the carryout number by passing the same two inputs through the AND gate. It is impossible to have both the total and the carry out at high in a half adder.

Lab 3 Full Adder:

Purpose:

The purpose of this lab was to design a full-adder circuit using Vivado. The full-adder circuit consists of two XOR gates, two AND gates, and one OR gate. There are three inputs and two outputs. With the three inputs, the first two inputs are the addition, and the third is the carry. The two outputs are the sum and the carry.

Data Collected:

All the data collected was displayed using the VHDL code in Vivado.

Calculations:

Lab 1 AND Gate:
Truth Table:

A_in	B_in	X_out
0	0	0
0	1	0
1	0	0
1	1	1

Lab 2 Half Adder:
Truth Table:

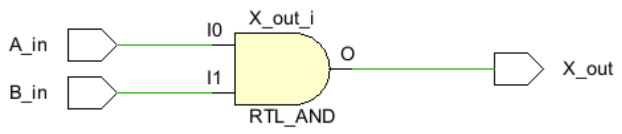
A_in	B_in	S_out	C_out
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

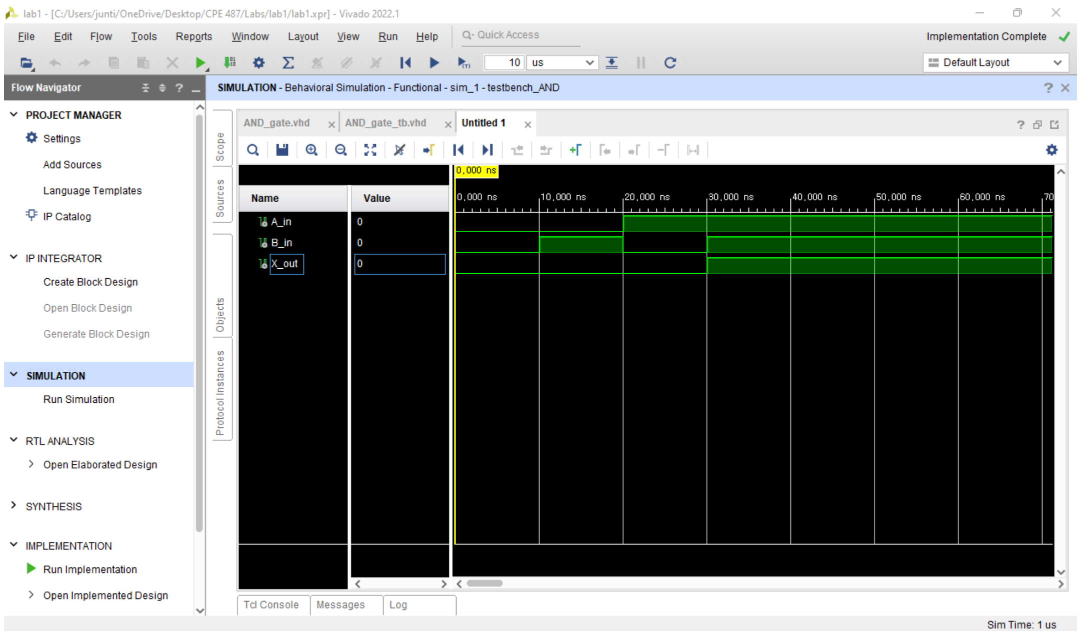
Lab 3 Full Adder:
Truth Table:

A	B	C_in	S	Cout
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

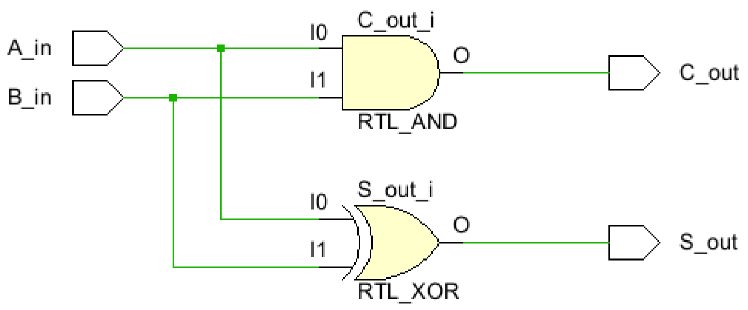
Results Schematics and Waveforms:

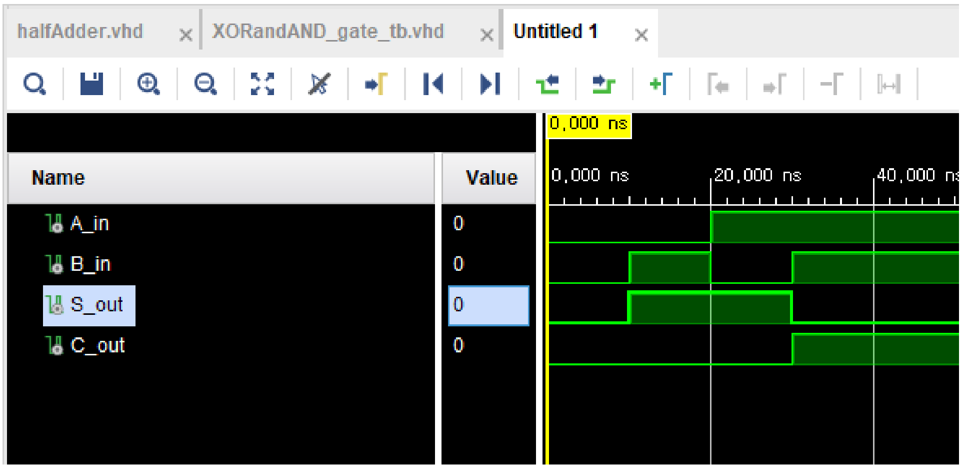
LAB 1 AND Gate:



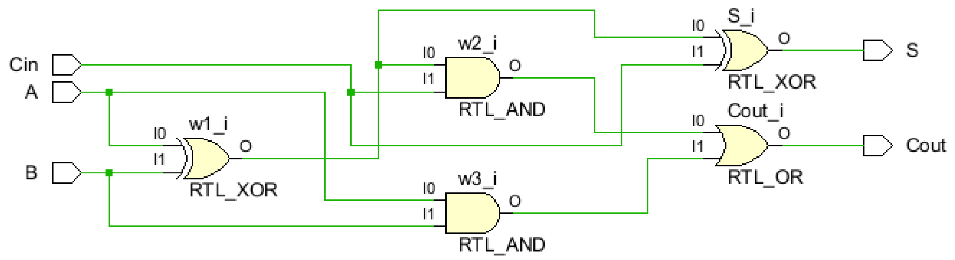


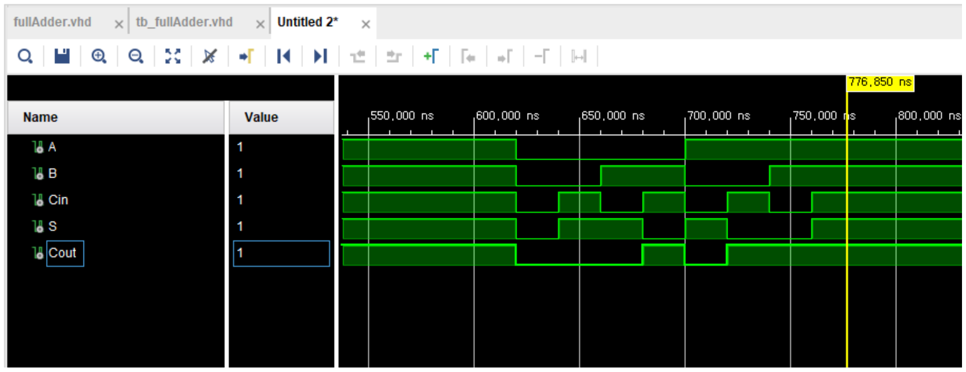
Lab 2 Half Adder:





Lab 3 Full Adder:





Name	Value
A	1
B	1
Cin	1
S	1
Cout	1

Conclusion:

The group was able to gain familiarity with Vivado, by building circuits and displaying their outputs at the end of these labs. The team was also able to compute the circuit output and check the outcomes using manual calculations.

Source Code:

< Lab 1 VHDL code >

```
---------------------------------------------------------------------
-- Name:          Lab 1
-- Author:        Junseok Oh & Shiddharath Patel
-- Date:          09/22/2022
-- Description:   Lab 1 - AND gate

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

```
entity AND_gate is
   Port ( A_in : in STD_LOGIC;
          B_in : in STD_LOGIC;
          X_out : out STD_LOGIC);
end AND_gate;

architecture Behavioral of AND_gate is

begin
   X_out <= (A_in and B_in);

end Behavioral;
```

< Lab 1 TB code >

```
----------------------------------------------------------------------
-- Name:          Lab 1
-- Author:        Junseok Oh & Shiddharath Patel
-- Date:          09/22/2022
-- Description:   Lab 1 - AND gate testbench code

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity testbench_AND is
end testbench_AND;

architecture Behavioral of testbench_AND is

component AND_gate
PORT(  A_in: in STD_LOGIC;
       B_in: in STD_LOGIC;
       X_out: out STD_LOGIC
       );
end component;

signal A_in, B_in, X_out : STD_LOGIC;

begin
uut: AND_gate
   PORT MAP(   A_in => A_in,
            B_in => B_in,
            X_out => X_out
            );
process
begin

   A_in <= '0';
```

```
    B_in <= '0';
    wait for 10 ns;
    A_in <= '0';
    B_in <= '1';
    wait for 10 ns;
    A_in <= '1';
    B_in <= '0';
    wait for 10 ns;
    A_in <= '1';
    B_in <= '1';
    wait for 10 ns;

    assert false report "end of test";

    wait;
  end process;
end Behavioral;
```

< Lab 2 VHDL code >

```
----------------------------------------------------------------------
-- Name:          Lab 2
-- Author:        Junseok Oh & Shiddharath Patel
-- Date:          09/22/2022
-- Description:   Lab 2 - half Adder

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity halfAdder is
    Port ( A_in : in STD_LOGIC;
           B_in : in STD_LOGIC;
           S_out : out STD_LOGIC;
           C_out : out STD_LOGIC);
end halfAdder;

architecture XORandAND of halfAdder is

begin
    C_out <= (A_in and B_in);
    S_out <= (A_in xor B_in);

end XORandAND;
```

< Lab 2 TB code >

```
----------------------------------------------------------------------
-- Name:          Lab 2
-- Author:        Junseok Oh & Shiddharath Patel
-- Date:          09/22/2022
-- Description:     Lab 2 - half Adder testbench code

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity testbench_XORandAND is
end testbench_XORandAND;

architecture Behavioral of testbench_XORandAND is

component halfAdder
PORT(  A_in: in STD_LOGIC;
       B_in: in STD_LOGIC;
       S_out: out STD_LOGIC;
       C_out: out STD_LOGIC
       );
end component;

signal A_in, B_in, S_out, C_out : STD_LOGIC;

begin
uut: halfAdder
   PORT MAP(   A_in => A_in,
               B_in => B_in,
               S_out => S_out,
               C_out => C_out
               );
process
begin

   A_in <= '0';
   B_in <= '0';
   wait for 10 ns;
   A_in <= '0';
   B_in <= '1';
   wait for 10 ns;
   A_in <= '1';
   B_in <= '0';
   wait for 10 ns;
   A_in <= '1';
   B_in <= '1';
   wait for 10 ns;

   assert false report "end of test";

   wait;
  end process;
end Behavioral;
```

< Lab 3 VHDL code >

```
----------------------------------------------------------------------
-- Name:          Lab 3
-- Author:        Junseok Oh & Shiddharath Patel
-- Date:          09/29/2022
-- Description:   Lab 3 - full Adder

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity fullAdder is
   Port ( A : in STD_LOGIC;
          B : in STD_LOGIC;
          Cin : in STD_LOGIC;
          S : out STD_LOGIC;
          Cout : out STD_LOGIC);
end fullAdder;

architecture rtl of fullAdder is

   signal w1 : std_logic;
   signal w2 : std_logic;
   signal w3 : std_logic;

begin
   w1 <= A xor B;
   w2 <= w1 and Cin;
   w3 <= A and B;

   Cout <= w2 or w3;
   S <= w1 xor Cin;

end rtl;
```

< Lab 3 TB code >

```
----------------------------------------------------------------------
-- Name:          Lab 3
-- Author:        Junseok Oh & Shiddharath Patel
-- Date:          09/29/2022
-- Description:   Lab 3 - full Adder testbench

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity tb_fullAdder is
end tb_fullAdder;
```

```
architecture Behavioral of tb_fullAdder is
    component fullAdder
        port(A,B,Cin : in std_logic;
            S, Cout : out std_logic);
    end component;
    signal A, B, Cin : std_logic;
    signal Cout, S : std_logic;
begin
    uut : fullAdder port map(A => A, B => B, Cin => Cin,
                    S => S, Cout => Cout);

    stim_proc : process
    begin
    wait for 100ns;
     A <= '0';
    B <= '0';
    Cin <= '0';
    wait for 20ns;
    A <= '0';
    B <= '0';
    Cin <= '1';
    wait for 20ns;
    A <= '0';
    B <= '1';
    Cin <= '0';
    wait for 20ns;
    A <= '0';
    B <= '1';
    Cin <= '1';
    wait for 20ns;
    A <= '1';
    B <= '0';
    Cin <= '0';
    wait for 20ns;
    A <= '1';
    B <= '0';
    Cin <= '1';
    wait for 20ns;
    A <= '1';
    B <= '1';
    Cin <= '0';
    wait for 20ns;
    A <= '1';
    B <= '1';
    Cin <= '1';
    wait for 20ns;
     end process;
    end Behavioral;
```