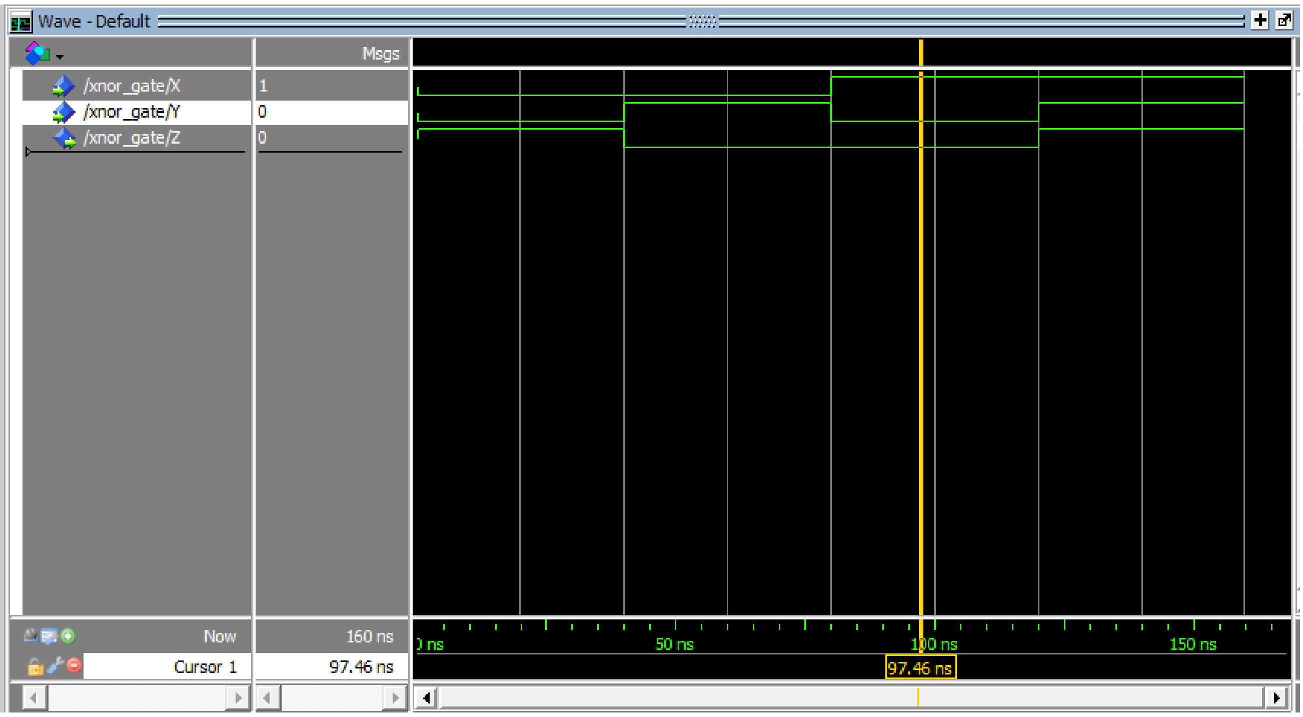
PACO, ARMIN R.
BSCpE – 3A

LOGIC GATES

AIM: Write a VHDL code for all the logic gates.

#7-TITLE: EX-NOR gate

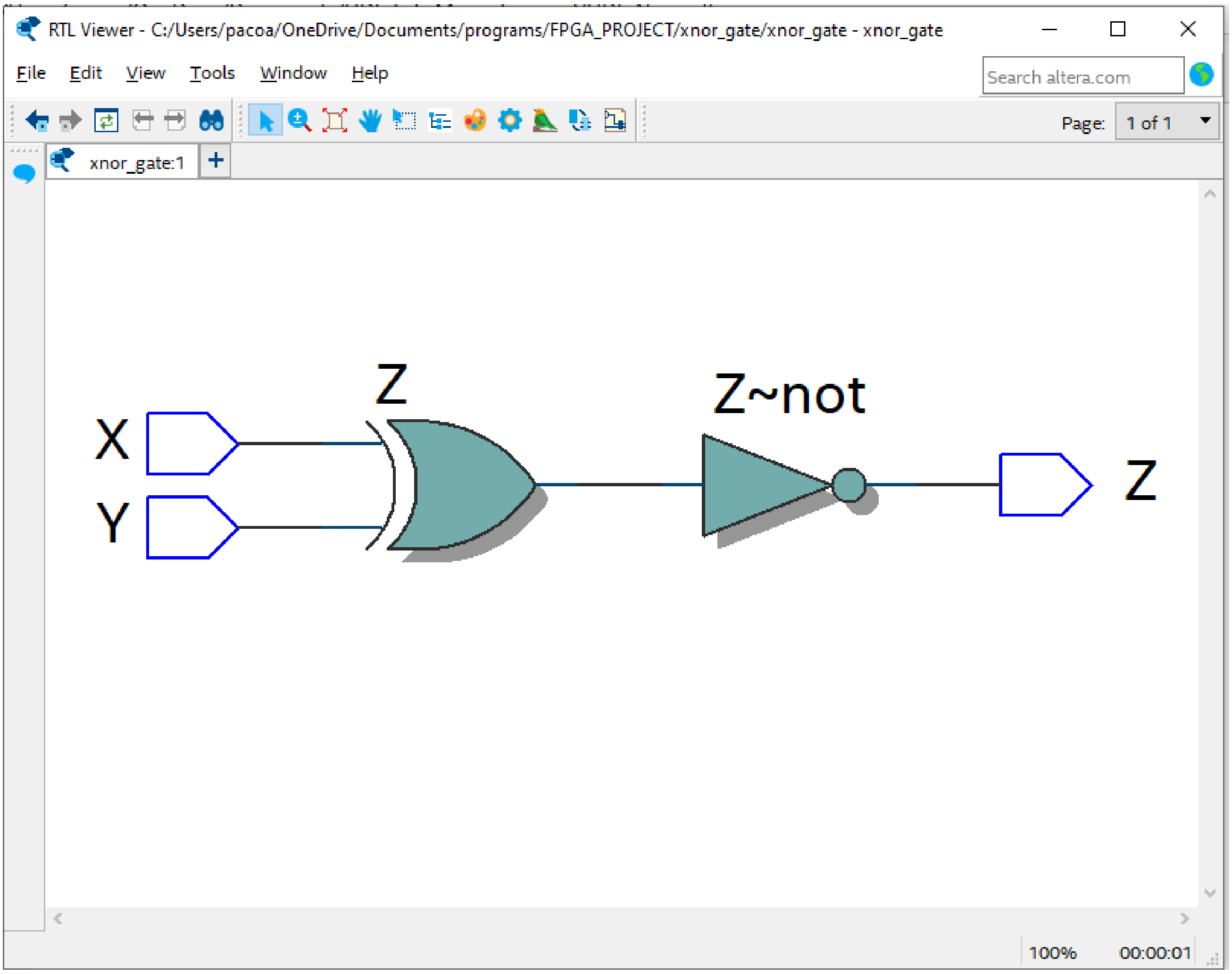
OUTPUT WAVE FORM:



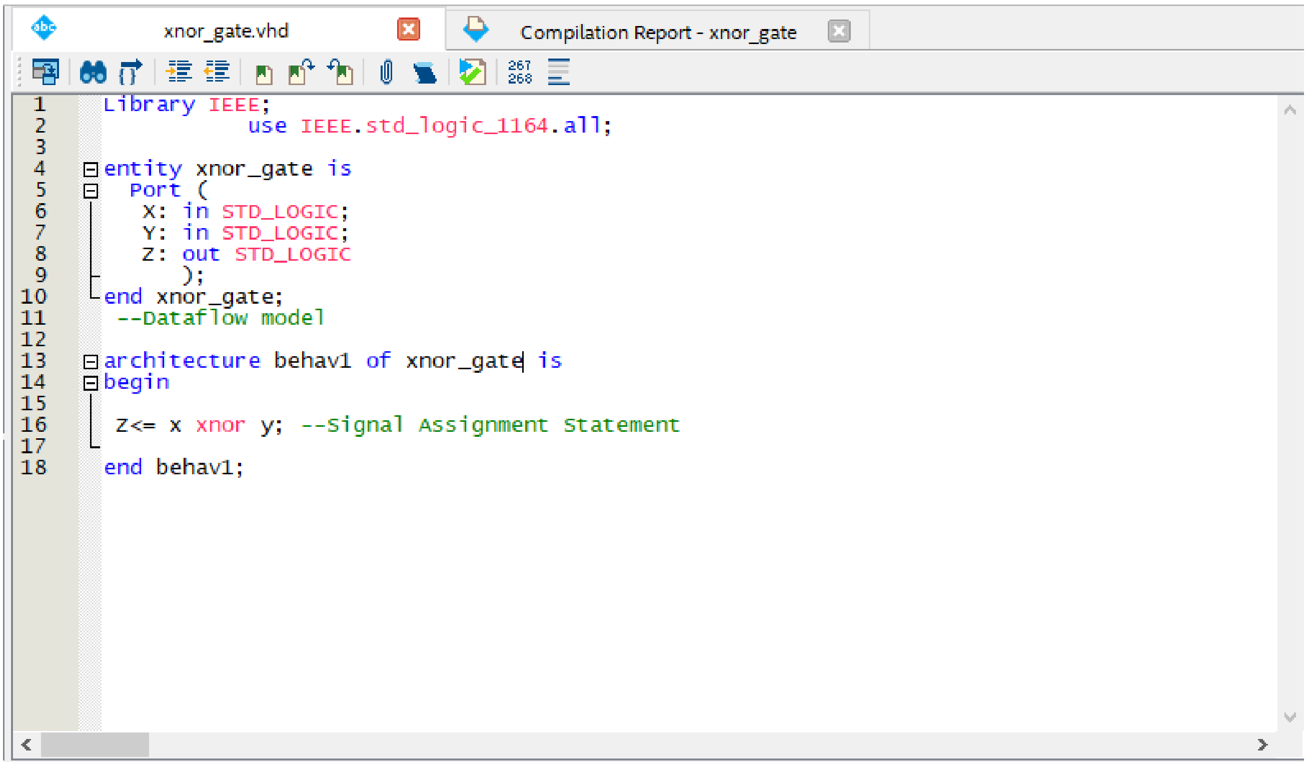
Wave - Default

	Msgs
/xnor_gate/X	1
/xnor_gate/Y	0
/xnor_gate/Z	0
Now	160 ns
Cursor 1	97.46 ns

RTL VIEW:


RTL Viewer - C:/Users/pacoa/OneDrive/Documents/programs/FPGA_PROJECT/xnor_gate/xnor_gate - xnor_gate
File Edit View Tools Window Help
Search altera.com
Page: 1 of 1
xnor_gate:1
Z
Z~not
X
Y
Z
100%
00:00:01

CODE:



```
Library IEEE;
           use IEEE.std_logic_1164.all;

entity xnor_gate is
  Port (
    X: in STD_LOGIC;
    Y: in STD_LOGIC;
    Z: out STD_LOGIC
       );
end xnor_gate;
 --Dataflow model

architecture behav1 of xnor_gate is
begin

 Z<= x xnor y; --Signal Assignment Statement

end behav1;
```

VIVA QUESTIONS:

1. Implement the following function using VHDL coding. (Try to minimize if you can).
F(A,B,C,D)=(A′+B+C) . (A+B′+D′). (B+C′+D′) . (A+B+C+D)

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY logic_function IS
    PORT (
        A, B, C, D : IN STD_LOGIC;
        F : OUT STD_LOGIC
    );
END logic_function;

ARCHITECTURE Dataflow OF logic_function IS
BEGIN
    F <= (NOT A OR B OR C) AND
         (A OR NOT B OR NOT D) AND
         (B OR NOT C OR NOT D) AND
         (A OR B OR C OR D);
END Dataflow;
```

2. What will be the no. of rows in the truth table of N variables?
Each variable has two possible values: **0 or 1**. If you have NNN independent variables, the number of different ways you can assign 0s and 1s to them is **2×2×...×22 \times 2 \times ... \times 22×2×...×2 (N times), which is 2N2^N2N**.

3. What are the advantages of VHDL?
VHDL has several key benefits: it's portable across different platforms, encourages modular and reusable design, and provides strong simulation and verification tools. It allows for precise hardware modeling and supports concurrent processes, making it a great choice for creating efficient and reliable digital systems.

4. Design Ex-OR gate using behavioral model?

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

entity XOR_Gate is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           Y : out STD_LOGIC);
end XOR_Gate;

architecture Behavioral of XOR_Gate is
```

```
begin
    -- Behavioral description of XOR gate
    process(A, B)
    begin
        if (A = '1' and B = '0') or (A = '0' and B = '1') then
            Y <= '1';
        else
            Y <= '0';
        end if;
    end process;
end Behavioral;
```

5. Implement the following function using VHDL code
f=AB+CD.

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

entity logic_function is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           C : in STD_LOGIC;
           D : in STD_LOGIC;
           f : out STD_LOGIC);
end logic_function;

architecture Behavioral of logic_function is
begin
    -- Behavioral description of the function f = AB + CD
    process(A, B, C, D)
    begin
        f <= (A and B) or (C and D);
    end process;
end Behavioral;
```

6. What are the differences between half adder and full adder?
- A full adder is a more complex version of a half adder as it can handle the carry input from previous stages, making it essential for multi-bit addition.

7. What are the advantages of minimizing the logical expressions?
- Minimizing logical expressions enhances efficiency, performance, and cost-effectiveness in digital circuit design.

8. What does a combinational circuit mean?
- Combinational circuits are digital circuits where the output depends only on the current inputs, with no memory of past inputs. They perform specific logical functions, and the output is determined instantly based on the present input values. Examples include logic gates, adders, and multiplexers.

9. Implement the half adder using VHDL code?

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

entity Half_Adder is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           Sum : out STD_LOGIC;
           Carry : out STD_LOGIC);
end Half_Adder;

architecture Behavioral of Half_Adder is
begin
    -- Sum is the XOR of A and B
    Sum <= A xor B;

    -- Carry is the AND of A and B
    Carry <= A and B;
end Behavioral;
```

10. Implement the full adder using two half adders and write VHDL program in structural model?

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

entity Full_Adder is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           Cin : in STD_LOGIC;   -- Carry input
           Sum : out STD_LOGIC;
           Cout : out STD_LOGIC); -- Carry output
end Full_Adder;

architecture Structural of Full_Adder is

    -- Component declaration of Half_Adder
    component Half_Adder
        Port ( A : in STD_LOGIC;
               B : in STD_LOGIC;
               Sum : out STD_LOGIC;
               Carry : out STD_LOGIC);
    end component;

    -- Internal signals for the half adders
    signal Sum1, Sum2 : STD_LOGIC;
    signal Carry1, Carry2 : STD_LOGIC;
```

```
begin

    -- First half adder adds A and B
    HA1: Half_Adder port map (A => A, B => B, Sum => Sum1, Carry =>
Carry1);

    -- Second half adder adds Sum1 and Cin
    HA2: Half_Adder port map (A => Sum1, B => Cin, Sum => Sum2,
Carry => Carry2);

    -- Sum is the output of the second half adder
    Sum <= Sum2;

    -- Carry out is the OR of the two carry outputs
    Cout <= Carry1 or Carry2;

end Structural;
```