Assignment 1

VLSI Design - VHDL Introduction

VHDL stands for very high-speed integrated circuit hardware description language. It is a programming language used to model a digital system by dataflow, behavioral and structural style of modeling. This language was first introduced in 1981 for the department of Defense (DoD) under the VHSIC program.

Describing a Design

In VHDL an entity is used to describe a hardware module. An entity can be described using,

- Entity declaration
- Architecture
- Configuration
- Package declaration
- Package body

Let's see what are these?

Entity Declaration

It defines the names, input output signals and modes of a hardware module.

Syntax –

```
entity entity_name is
   Port declaration;
end entity_name;
```

An entity declaration should start with 'entity' and end with 'end' keywords. The direction will be input, output or inout.

In	Port can be read
Out	Port can be written
Inout	Port can be read and written
Buffer	Port can be read and written, it can have only one source.

Architecture –

Architecture can be described using structural, dataflow, behavioral or mixed style.

Syntax –

```
architecture architecture_name of entity_name
architecture_declarative_part;

begin

   Statements;
end architecture_name;
```

Here, we should specify the entity name for which we are writing the architecture body. The architecture statements should be inside the 'begin' and 'énd' keyword. Architecture declarative part may contain variables, constants, or component declaration.

Data Flow Modeling

In this modeling style, the flow of data through the entity is expressed using concurrent (parallel) signal. The concurrent statements in VHDL are WHEN and GENERATE.

Besides them, assignments using only operators (AND, NOT, +, *, sll, etc.) can also be used to construct code.

Finally, a special kind of assignment, called BLOCK, can also be employed in this kind of code.

In concurrent code, the following can be used –

- Operators
- The WHEN statement (WHEN/ELSE or WITH/SELECT/WHEN);
- The GENERATE statement;
- The BLOCK statement

Behavioral Modeling

In this modeling style, the behavior of an entity as set of statements is executed sequentially in the specified order. Only statements placed inside a PROCESS, FUNCTION, or PROCEDURE are sequential.

PROCESSES, FUNCTIONS, and PROCEDURES are the only sections of code that are executed sequentially.

However, as a whole, any of these blocks is still concurrent with any other statements placed outside it.

One important aspect of behavior code is that it is not limited to sequential logic. Indeed, with it, we can build sequential circuits as well as combinational circuits.

The behavior statements are IF, WAIT, CASE, and LOOP. VARIABLES are also restricted and they are supposed to be used in sequential code only. VARIABLE can never be global, so its value cannot be passed out directly.

Structural Modeling

In this modeling, an entity is described as a set of interconnected components. A component instantiation statement is a concurrent statement. Therefore, the order of these statements is not important. The structural style of modeling describes only an interconnection of components (viewed as black boxes), without implying any behavior of the components themselves nor of the entity that they collectively represent.

In Structural modeling, architecture body is composed of two parts – the declarative part (before the keyword begin) and the statement part (after the keyword begin).

logic gates

Experiment -1: Write a VHDL code for all the logic gates

Objective:

The objective of this experiment is to:

i. To revise the working of various logic gates

ii. To learn the VHDL coding

iii. To simulate for functional verification

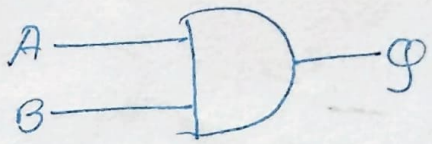
iv. To implement on CPLD / FPGA

1. TITLE: AND gate

TRUTH TABLE:

x	y	z
0	0	0
0	1	0
1	0	0
1	1	1

Symbol:



Logic Equation:

Z = X.Y

This equation can be used for describing the dataflow model of AND architecture

We observe that the output is high when both X and Y are high '1', otherwise the output is low '0'. The property can be used in modeling the sequential behaviour

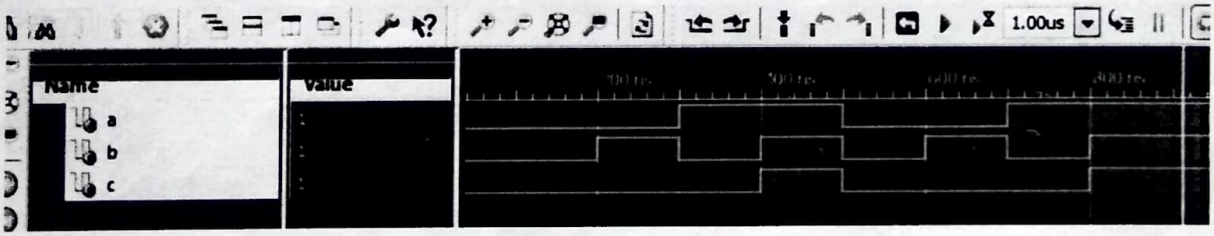
VHDL CODE:

```
Library IEEE;
use IEEE.std_logic_1164.all;
entity AND2 is
        port(
                A : in STD_LOGIC;
                B : in STD_LOGIC;
                C : out STD_LOGIC
```

```
        );
    end AND2;

-- The three architectural models are given below:
--1.  Dataflow model
        architecture behav1 of AND2 is
        begin
          C <= A and B;        --Signal Assignment Statement
        end behav1;
-- 2.  Behavioral model
        architecture behav2 of AND2 is
        begin
          process (A, B)
          begin
              if (A='1' and B='1') then
               C<= '1';
              else
               C<= '0';
              end if;
          end process; end behav2;
```

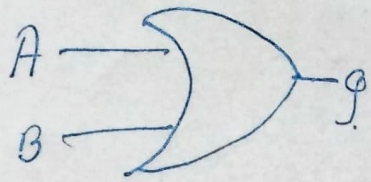
OUT PUT WAVE FORM:



2. TITLE: OR gate

TRUTH TABLE:

x	y	z
0	0	0
0	1	1
1	0	1
1	1	1



Logic Equation:

i. **Z = X + Y This can be used for describinbg the dataflow model of AND architecture**

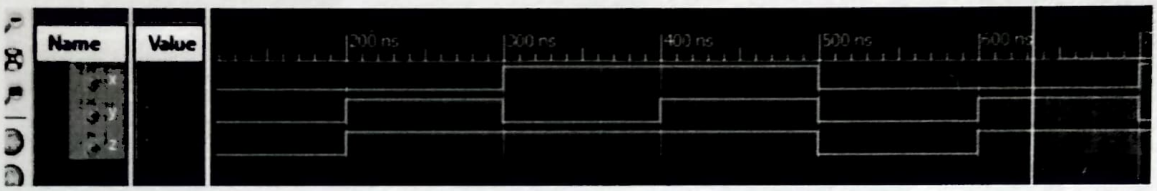
ii. **We observe that the output is high when any or all the inputs X and Y are high '1', otherwise the output is low '0'. The property can be used in modeling the sequential behaviour**

VHDL CODE

```
Library IEEE;
use IEEE.std_logic_1164.all;
entity OR2 is
        port(
                x : in STD_LOGIC;
                y : in STD_LOGIC;
                z : out STD_LOGIC
            );
end OR2;
--Dataflow model
architecture behav1 of OR2 is
begin
   Z <= x or y;          --Signal Assignment Statement
end behav1;
-- Behavioral model
        architecture behav2 of OR2 is
```

```
begin
    process (x, y)
    begin
        if (x='0' and y='0') then
          Z <= '0';
        else
          Z<= '1';
        end if;
     end process; end behav2;
```

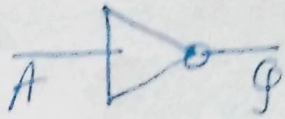
OUTPUT WAVEFORM:



3. TITLE: NOT gate

TRUTH TABLE:

x	z
0	1
1	0



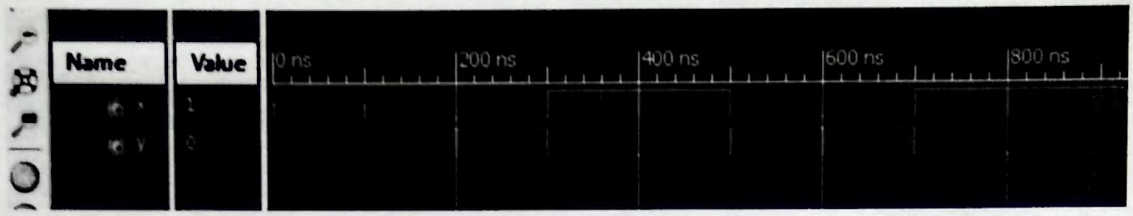
1. **Y = X' This can be used for describing the dataflow model of AND architecture**
2. **We observe that the output is high when inputs X low '0', and the output is low when input X is high '1'. The property can be used in modeling the sequential behavior**

VHDL CODE:

```
Library IEEE;
use IEEE.std_logic_1164.all;
entity not1 is
        port(
              X:   in  STD_LOGIC;
             Y:    out  STD_LOGIC
             );
        end not1;
--Dataflow model
        architecture behav1 of not1 is
        begin
          Y<= not X;  --Signal  Assignment  Statement
        end behav1;
-- Behavioral model
        architecture behav2 of not1 is
        begin
           process (X)
```

```
begin
    if (x='0') then  -- Compare with truth table
    Y<= '1';
    else
    Y<= '0';
    end if;
    end process;
end behav2;
```

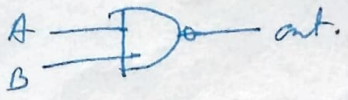
OUTPUT WAVEFORM for NOT Gate:



4. TITLE: NAND gate

TRUTH TABLE:

x	y	z
0	0	1
0	1	1
1	0	1
1	1	0



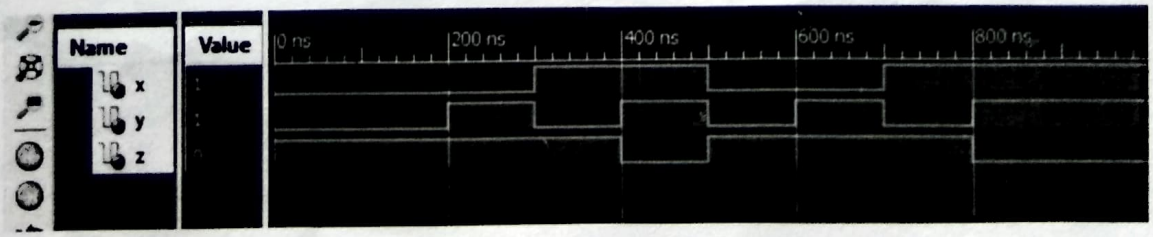
i. **Y = (X.Y)' This can be used for describing the dataflow model of AND architecture**

ii. **We observe that the output is high when any or all inputs X and Y are low '0', and the output is low when all input are high '1'. The property can be used in modeling the sequential behavior**

VHDL CODE:

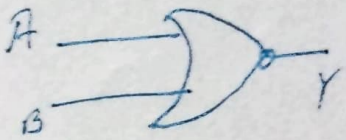
```
Library IEEE;
use IEEE.std_logic_1164.all;
entity nand2 is
        port(
                x : in STD_LOGIC;
                y : in STD_LOGIC;
                z : out STD_LOGIC
            );
end nand2;
--Dataflow model
architecture behav1 of nand2 is
begin
        z<= x nand y;          --Signal Assignment Statement
end behav1;
-- Behavioral model
architecture behav2 of nand2 is
begin
 Process (x, y)
 Begin
    If (x='1' and y='1') then
        Z <= '0';
    else
        Z <= '1';
    end if;
end process; end behav2;
```

OUTPUT WAVEFORM



5. **TITLE**: NOR gate

TRUTH TABLE:



x	y	z
0	0	1
0	1	0
1	0	0
1	1	0

i. **Y = (X + Y)' This can be used for describing the dataflow model of AND architecture**

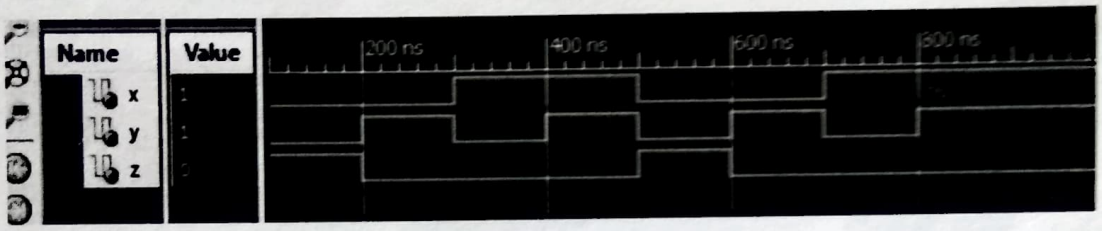
ii. **We observe that the output is high when all inputs X and Y are low '0', and the output is low when any or all input are high '1'. The property can be used in modeling the sequential behavior**

VHDL CODE:

```
Library IEEE;
use IEEE.std_logic_1164.all;
entity nor2 is
        Port (
                X: in STD_LOGIC;
                Y: in STD_LOGIC;
                Z: out STD_LOGIC
            );
end nor2;
--Dataflow model
architecture behav1 of nor2 is
begin
Z<= x nor y;
end behav1;
-- Behavioral model
architecture behav2 of nor2 is begin
```

```
process (x, y)
 begin
--Signal Assignment Statement
        If (x='0' and y='0') then
          Z <= '1';
        else
          Z <= '0';
        end if;
      end process;
    end behav2;
```

OUTPUT WAVEFORM:



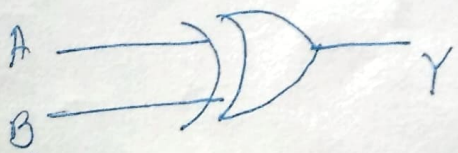
6. TITLE:EX-OR gate

TRUTH TABLE:

a. Y = (X '.Y +X .Y') This can be used for describing the dataflow model of AND architecture

b. We observe that the output is low when all inputs X and Y are low '0' or when all the inputs are high. The output is high when any input is high '1'. The property can be used in modeling the sequential behavior

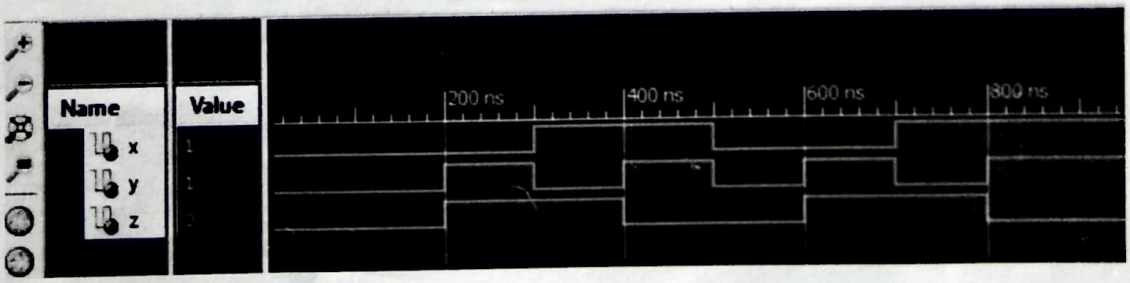
x	y	z
0	0	0
0	1	1
1	0	1
1	1	0



VHDL CODE:

```
Library IEEE;
use IEEE.std_logic_1164.all;
   entity xor2 is
         Port ( X: in STD_LOGIC;
                Y: in STD_LOGIC;
                Z: out STD_LOGIC
         );
   end xor2;
 --Dataflow model
   architecturedataflow  of  xor2  is
   begin
         Z<= x xor y;          --Signal Assignment Statement
   enddataflow;
--behaviour modelling
      architecture behav2 of xor2 is
      begin
        process  (x,  y)
        begin
           If (x/=y) then
             Z <= '1';
           else
             Z<= '0';
           end if;
        end process;
end behav2;
```

OUTPUT WAVEFORM



Implementation of XNOR

Symbol:

X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	1

Data flow model

```
VHDL Code:

Library ieee;
use ieee.std_logic_1164.all;

entity xnor1 is
   port(a,b:in bit ; c:out bit);
end xnor1;

architecture virat of xnor1 is
begin
   c<=not(a xor b);
end virat;
```

Behavioural model

```
Library ieee;
use ieee.std_logic_1164.all;

---------------------------------------

Entity XNOR_ent is
port(       x: in std_logic;
            y: in std_logic;
            F: out std_logic
);
End XNOR_ent;
---------------------------------------
architecture behv1 of  XNOR_ent is
begin

process(x, y)
begin
                                    -- compare to truth table
if (x/=y) then
              F <= '0';
            else
              F <= '1';
            end if;
end process;

end behv1;
```

VHDL Code:

```
Library ieee;
use ieee.std_logic_1164.all;

entity xor1 is
   port(a,b:in bit ; c:out bit);
end xor1;

architecture virat of xor1 is
begin
   c<=a xor b;
end virat;
```

Waveforms