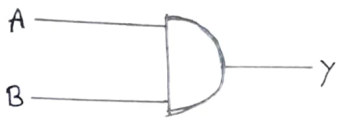
Computer Organisation & Architecture Lab

Assignment - 1

1) Write a VHDL program to implement all the AND gates using data flow.

Title :- Implementation of AND gate using data flow modelling.

Symbolic Diagram :-



Boolean Expression :- [A · B]

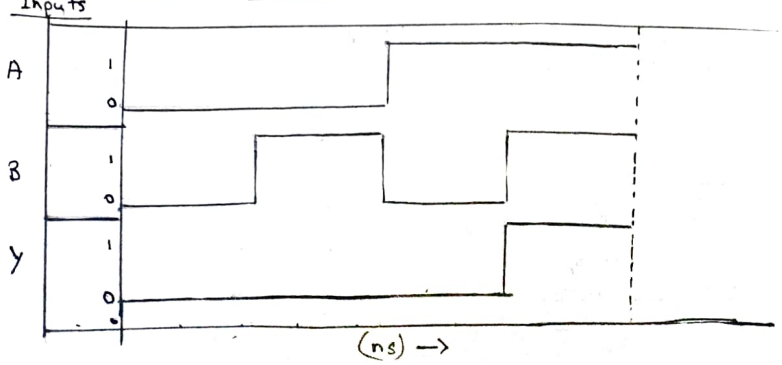
Truth Table :-

A	B	Y=(A · B)
0	0	0
0	1	0
1	0	0
1	1	1

VHDL Code :-

```
Library IEEE;
    use IEEE.std_logic_1164.all;
    entity AND_gate is
        port ( A: in std_logic;
               B: in std_logic;
               Y: out std_logic);
    end AND_gate;
architechture andLogic of AND_gate is
    begin
        Y <= A AND B;
    end andLogic;
```

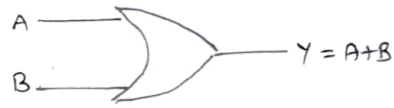
Output Waveforms



2) Write a VHDL program to implement all OR Gate using data flow modelling.

Title :- Implementation of OR Gate using data flow modelling.

Symbollic Diagram :-



Boolean Expressions :- (A+B)

Truth Table :-

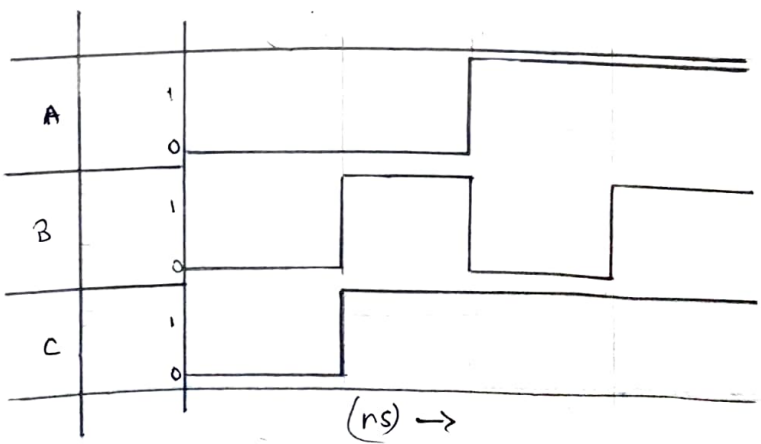
A	B	Y=(A + B)
0	0	0
0	1	1
1	0	1
1	1	1

VHDL Code :-

```
Library IEEE;
    use IEEE.std_logic_1164.all;
    entity OR-gate is
    Port ( A : in std_logic;
           B : in std_logic;
           Y : out std_logic; )
    end OR-gate;
    architechture OR-gateLogic of OR-gate is
    begin
        Y<=A AND B;
    end andLogic;
```

```
Y <= A AND B;
end OR_gateLogic;
```

Output Waveform :-



3) Write a VHDL Programming to implement all NOT Gate using data flow modelling.

Title:- Implementation of NOT Gate using data flow modelling.

Symbolic Diagram :-

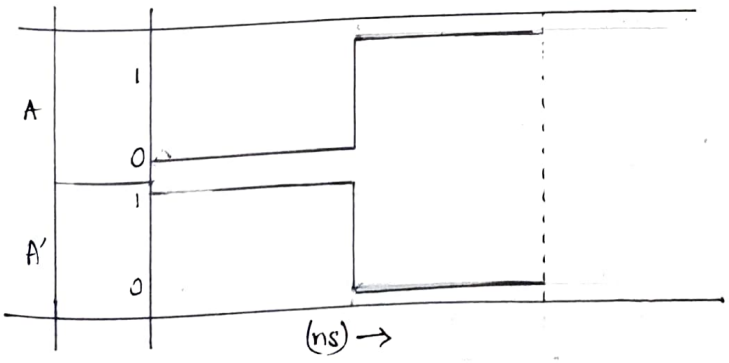
Truth Table:-

A	A'=Y
0	1
1	0

VHDL Code :-

```
Library IEEE;
  use IEEE.std_logic_1164.all;
entity NOT_gate is
    Port ( A: in std_logic;
           Y: out std_logic);
    end NOT_gate;
  architechture NOTLogic of NOT_gate is
    begin
      Y <= NOT (A);
    end NOTLogic;
```

Output Waveform

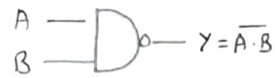


4) Write a VHDL programs to implement all NAND Gate using data flow modelling.

Title :- Implementation of NAND Gate using data flow modeling.

Boolean Expression :- $Y = \overline{A \cdot B}$

Symbolic Diagram :-



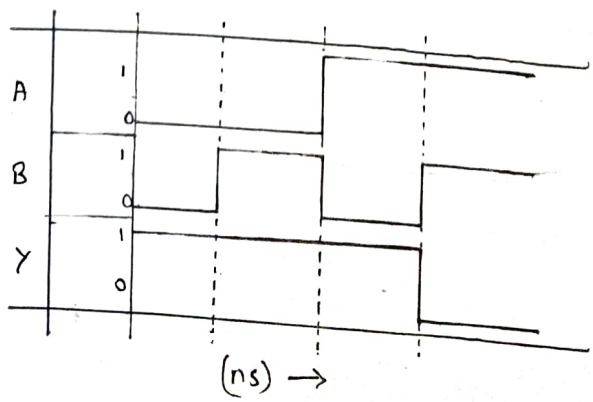
Truth Table :-

A	B	$Y = \overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

VHDL Code :-

```
Library IEEE;
  use IEEE_std_logic_1164.all;
entity NAND_gate is
    Port ( A : in std_logic;
           B : in std_logic;
           Y : out std_logic);
  end NAND_gate;
architechture NANDLogic of NAND_gate is
    begin
      Y <= A NAND B;
    end NANDLogic;
```

Output Waveform



5) Write a VHDL Program to implement all NOR GATE using data flow modeling.

Title :- Implementation of NOR gate using data flow modeling

Boolean Expression :- $Y = \overline{A + B}$

Symbolic Diagram :-

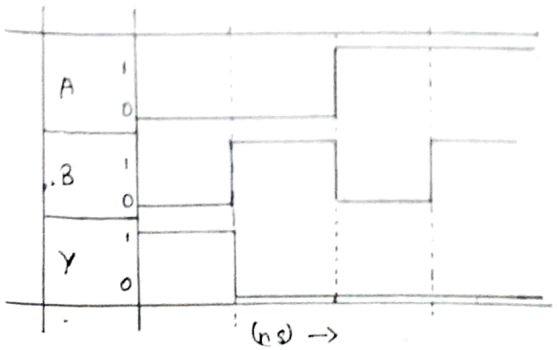
Truth Table :-

A	B	$Y = \overline{A + B}$
0	0	1
0	1	0
1	0	0
1	1	0

VHDL Code :-

```
Library IEEE;
  use IEEE_std_logic_1164.all;
  entity NORgate is
    Port ( A: in std_logic;
           B: in std_logic;
           Y: out std_logic);
  end NORgate;
architecture NORLogic of NORgate is
    begin
      Y <= A NOR B;
    end NORLogic;
```

Output Waveform :-

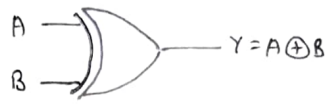


6) Write a VHDL program to implement all XOR Gate using data flow modeling.

Title :- Implementation of XOR gate using data flow modeling

Boolean Expression :- $Y = A \oplus B$

Symbolic Diagram :-



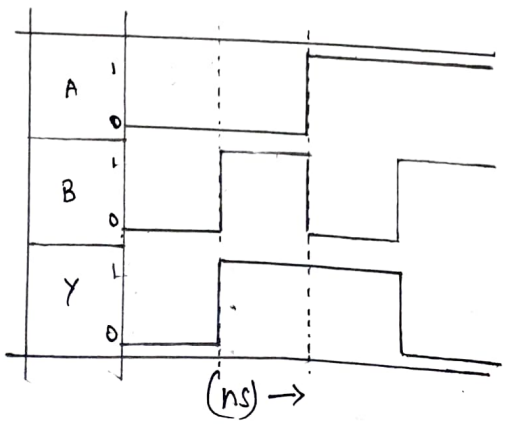
Truth Table :-

A	B	Y = A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0

VHDL CODE :-

```
Library IEEE;
  use IEEE_std_logic_1164.all;
  entity XORgate is
    port ( A: in std_logic;
           B: in std_logic;
           Y: out std_logic);
    end XORg;
  architechture XORLogic of XORg is.
    begin
      Y <= A XOR B;
    end XORLogic;
```

Output Waveform :-



7) Write a VHDL program to implement all XNOR Gate using data flow modeling.

Title :- Implementation of XNOR gate using data flow modeling.

Boolean Expression :- $Y \Rightarrow Y = \overline{A \oplus B}$

Symbolic Diagram :-

Truth Table :-

A	B	$Y = \overline{A \oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1

VHDL CODE :-

```
Library IEEE;
    use IEEE_std_logic_1164.all;
    entity XNORg is
        Port ( A: in std_logic;
               B: in std_logic;
               Y: out std_logic);
    end XNORg;
    architechture XNORLogic of XNORg is
    begin
        Y <= A XNOR B;
    end XNORLogic;
```

Output Waveform :-