Expt.No.

Page No.
Date.

Objective :- i.) XNOR GATE design using VHDL
ii.) NAND GATE design using VHDL

Theory :- The XNOR GATE and NAND GATE is a digital Logic GATE whose functions are the logical complement of the exclusive OR (XOR) gate and AND gate respectively. Two input version of XNOR implements logical equality, behaving according to the truth table, hence XNOR gate is sometimes called as "equivalence gate". In case of XNOR gate a high output (1) results if both of the inputs to the gate are the same. In case of NAND gate (3-input) a high output (1) results if all among the inputs to the gate are at low level (0).

Boolean expression of XNOR Gate :- $Q = \overline{A \oplus B} = A \odot B$
Boolean expression of NAND Gate :- $Q = \overline{A \cdot B \cdot C}$

Truth table :-

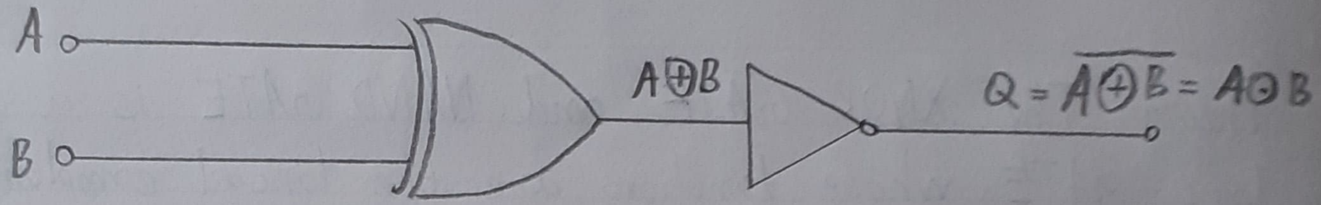
2 input XNOR gate :-

A	B	$Y = A \odot B$
0	0	1
0	1	0
1	0	0
1	1	1

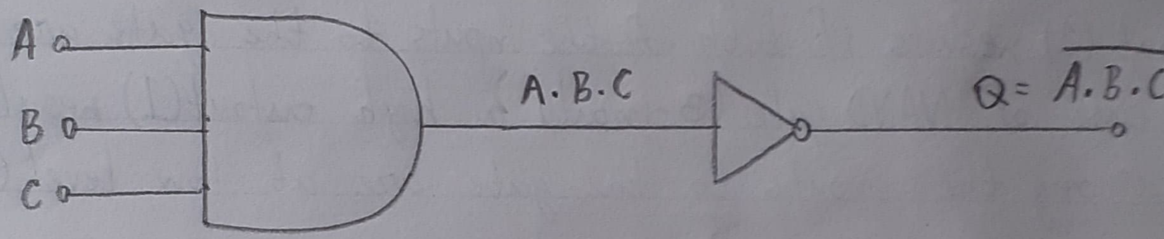
3 input NAND Gate :-

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

Teacher's Signature ....



2 input XNOR GATE



3 input NAND GATE

Expt.No.

Page No./
Date./

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

Code:-

2-input XNOR:-

Main code:-

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

entity xnor_gate is
        Port (a,b : in std_logic;
              q: out std_logic);

end xnor_gate;

architecture rtl of xnor_gate is
begin
    q<= (a and b) or (not(a) ~~not~~ and not(b));
end rtl;
```

Testbench code:-

```
-- Testbench for XNOR gate
library IEEE;
use IEEE.std_logic_1164.all
```

OXFORD©

Teacher's Signature ....

Expt.No.

Page No.
Date.

```
entity testbench is
-- empty
end testbench;

architecture tb ~~for~~ of testbench is

--DUT component
component xnor_gate is
port(
   a: in std_logic;
   b: in std_logic
   q: in std_logic);
   ~~q: in s~~
end component;

 signal a_in, b_in, q_out: std_logic;

 begin

    --Connect DUT
    DUT: xnor_gate port map(a_in, b_in, q_out);

    process
    begin
       a_in <= '0';
       b_in <= '0';
```

OXFORD

Teacher's Signature ....

Expt.No.

Page No.

Date.

```
    wait fon 1ns;

    a_in <= '0';
    b_in <= '1';
    wait fon 1 ns;

    a_in <= '1';
    b-in <= '0';
    wait for 1ns;

    a_in <= '1';
    b-in <= '1';
    wait fon 1 ns;
    wait;
  end pnocess;
end tb;
```

3-input NAND gate - Simple NAND gate design

Main code:-

```
Libnany IEEE;
use IEEE.std_logic_1164.all;

entity nand_gate is
port (
    a: in std_logic;
    b: in std_logic;
    C: in std_logic;
    q: out std_logic);
 end nand_gate;
```

OXFORD®

Teacher's Signature ....

Expt.No. Page No. Date.

```
    architecture rtl of nand_gate is
        begin
           q <= not(a and b and c)
    end rtl;
```

Testbench code:-

```
--Testbench for NAND gate
library IEEE;
use IEEE.std_logic_1164.all;

entity testbench is
-- empty
end testbench;

architecture tb of testbench is

--DUT component
component nand_gate is
port (
   a: in std_logic;
   b: in std_logic;
   c: in std_logic;
   q: out std_logic);
end component;
```

Teacher's Signature ....

Expt.No.

Page No.	
Date.	

```
signal  a_in, b_in, c_in, q_out : std_logic;

begin
    --Connect DUT
    DUT: nand_gate  port map(a_in, b_in, c_in, q_out);

    process
    begin
        a_in <= '0';
        b_in <= '0';
        c_in <= '0';
        wait for 1 ns;

        a_in <= '0';
        b_in <= '0';
        c_in<= '1';
        wait for 1 ns;

        a_in <= '0';
        b_in <= '1';
        c_in <= '0';
        wait for 1 ns;

        a_in <= '0';
        b_in<= '1';
        c_in <= '1';
        wait for 1 ns;
```

OXFORD

Teacher's Signature ....

Expt.No.

Page No.

Date.

```
        a_in <= '1';
        b_in <= '0';
        c_in <= '0';
        wait for 1 ns;

        a_in <= '1';
        b_in <= '0';
        c_in <= '1';
        wait for 1 ns;

        a_in <= '1';
        b_in <= '1';
        c_in <= '0';
        wait for 1 ns;

        a_in <= '1';
        b_in <= '1';
        c_in <= '1';
        wait for 1 ns;

        wait;
    end process;
end tb;
```

OXFORD

Conclusion:- Here in this experement we have designed logic gates using VHDL programming in EDA playground.

Teacher's Signature ....