

Assignment - 1

Q.1. Implement Various Gates.

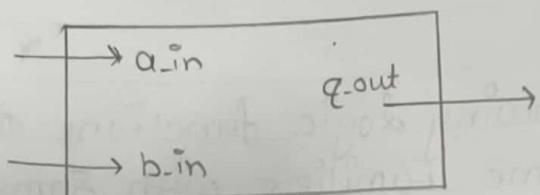
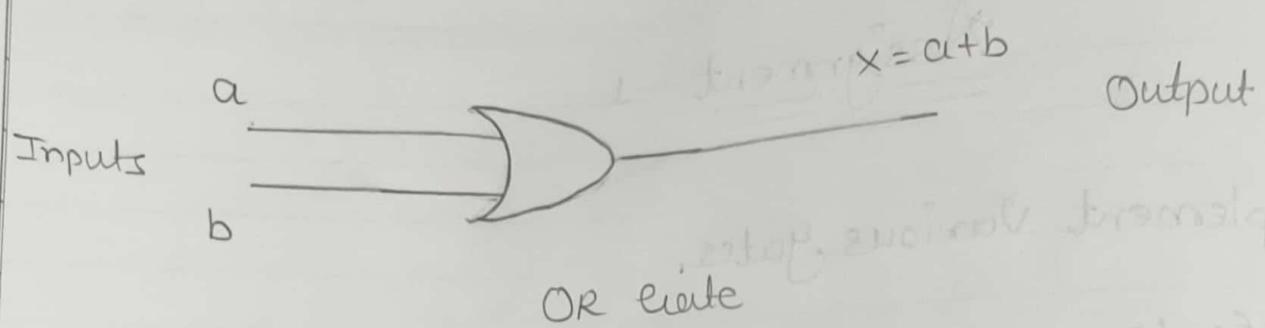
OR Gate

- Objective :- To knowing logic functions OF OR gate and become familiar with some of its application.
- Description:- An OR gate is a logic gate that performs logical OR operation. A logical OR operation has a high output (1) if one or both the inputs to the gate are high (1). If neither input is high, a low output (0) results.

OR gate may have any number of input probes but only one output probe.

- Truth Table :- List the output of a particular digital logic circuit for all the possible Combinations of its inputs.

Teacher's Signature _____



Input		Output
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

- VHDL Code :-

```
library IEEE;
use IEEE.std_logic_1164.all;
entity Orgate is
port (
    a : in std_logic;
    b : in std_logic;
    q : out std_logic );
end orgate;
```

architecture Behavioral of Orgate is

begin

q <= a or b;

end Behavioral;

Teacher's Signature _____

-- Test bench of OR gate

```
library IEEE;  
use IEEE.std_logic_1164.all;
```

```
entity Orbench is  
    -- empty  
end Orbench;
```

architecture tb of Orbench is

component Orgate is
port (

```
    a : in std_logic;  
    b : in std_logic;  
    q : out std_logic);
```

end component;

Signal a_in, b_in, q_out : std_logic;

begin

```
    unit: Orgate port map(a_in, b_in, q_out);
```

process

begin

```
    a_in <='0';
```

```
    b_in <='0';
```

```
    wait for 100 ns;
```

```
    a_in <='0';
```

```
    b_in <='1';
```

```
    wait for 100 ns;
```

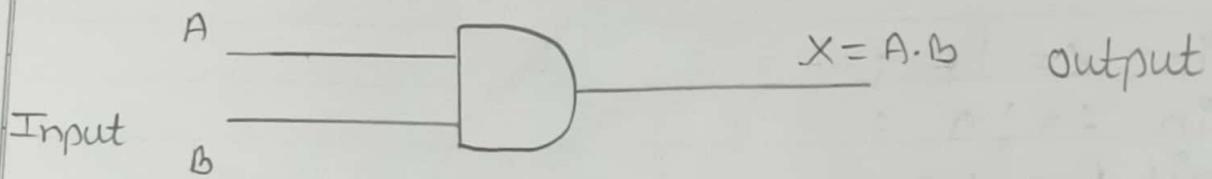
Teacher's Signature _____

a-in <= '1';
b-in <= '0';
Wait for 100 ns;

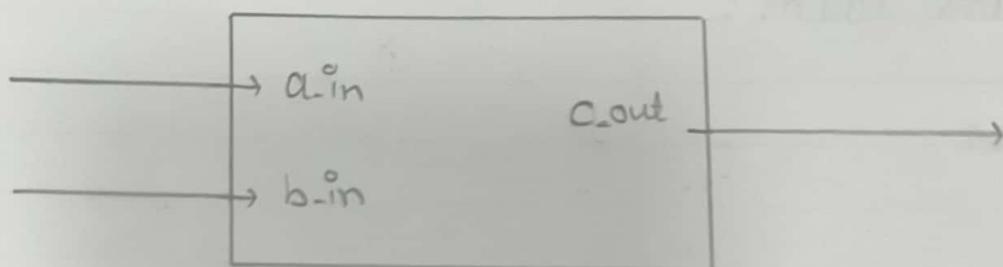
a-in <= '1';
b-in <= '1';
Wait for 100 ns;

end process
end tb;

Teacher's Signature _____



AND gate



And Gate

- **Objective :-** To knowing logic function of AND gate become familiar with some of its application.
- **Description :-** An AND gate is a logic gate having two or more inputs and a single output.

An AND gate operates on logical multiplication rules. In this gate, if either of the input is low(0), then the output is also low. If all of the inputs are high(1), then the output will also have any number of inputs.

AND gate is digital device that produces high output only when all inputs are high.

Logical operation of AND gate can be represented as $X = A \cdot B$

- **Truth Table :-**

Input		Output
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

Teacher's Signature _____

- VHDL Code:-

```
library IEEE;
use IEEE.std_logic_1164.all;
entity andgate is
port (
    a : in std_logic;
    b : in std_logic;
    c : out std_logic);
end andgate;
```

architecture alt of andgate is
begin

```
process(a,b) is
begin
    c <= a and b;
end process;
end alt;
```

- TestBench:-

```
library IEEE;
use IEEE.std_logic_1164.all;
entity andbench is
end andbench;
```

architecture tb of andbench is
Component andgate is

Teacher's Signature _____

```
port (
    a : in std_logic;
    b : in std_logic;
    c : out std_logic);  
end component;
```

Signal a-in, b-in, c-out : std-logic;
begin

unit : andgate port map(a-in, b-in, c-out) :

process

begin

a-in <= '0';

b-in <= '0';

wait for 100ns,

assert (c-out = '0') report "fail:." severity error;

a-in <= '0';

b-in <= '1';

wait for 100ns,

assert (c-out = '1') report "fail:." severity error;

a-in <= '1';

b-in <= '0';

wait for 100ns,

assert (c-out = '0') report "fail:." severity error;

a-in <= '1';

b-in <= '1';

wait for 100ns,

assert (c-out = '1') report "fail:." severity error;

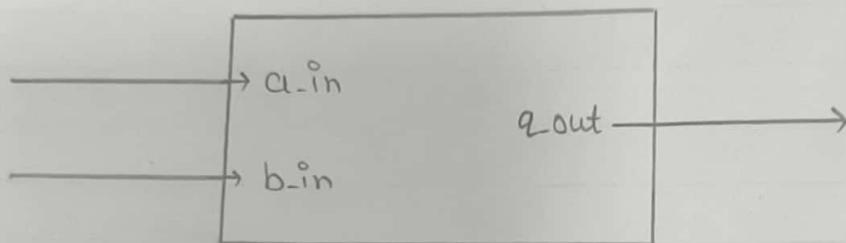
Teacher's Signature _____

```
a.in <='0';
b.in <='0';
assert false report "Test alone" Severity error;
wait;
end process;
end tb;
```

Teacher's Signature _____



XOR gate



ch

XOR Gate

Objective :- knowing logic function of XOR gate become familiar with some of its application.

Description :- XOR gate stands for the exclusive OR gate. OR gate This gate is a special type of gate used in different types of computational circuit.

There are two special gates, i.e., Ex-OR and X-OR. These gates are not basic gates in their own and are constructed by combining with other logic gates.

In the Ex-OR Function, the logic output "1" is obtained only when either $A = 1$ or $B = 1$ but not both together at the same time. Simply, the output of the XOR gate is high (1) only when both the inputs are different from each others.

Plus (+) sign within the circle is used as the Boolean expression of the XOR gate.

So, the symbol of the XOR gate is \oplus . X-OR symbol also defines the "direct sum of sub-objects" expression.

$$Y = (A \oplus B)$$

$$Y = (A'B + AB')$$

• Truth Table :-

Inputs		Output
A	B	y
0	0	0
0	1	1
1	0	1
1	1	0

• VHDL code :-

library IEEE;

use IEEE.std_logic_1164.all;

entity Xor_gate is

Port (

a : in std_logic;

b : in std_logic;

q : out std_logic);

end Xor_gate;

Architecture atr1 of Xor_gate is

begin

process (a,b) is

begin

q <= a XOR b ;

end process;

end atr1;

• Testbench Code :

```

library IEEE;
use IEEE.std_logic_1164.all;
entity Xorbench is
end Xorbench;
architecture tb of Xorbench is
Component Xor_gate is
port(
    a : in std_logic;
    b : in std_logic;
    q : out std_logic);
end Component;
Signal a_in, b_in, q_out : std_logic;
begin
unit : Xor_gate port map(a_in, b_in, q_out);
process
begin
    a_in <='0';
    b_in <='0';
    wait for 100ns;
    assert(q_out='0') report "fail." severity error;
    a_in <='1';
    b_in <='1';
    wait for 100ns;
    assert(q_out='1') report "fail." severity error;
    a_in <='1';
    b_in <='0';
    wait for 100ns;

```

Teacher's Signature _____

assert(c_out == '1') report "fail"; Severity error.

a_in <= '1';

b_in <= '1';

wait for 100ns

assert(q_out == '0') report "fail"; Severity error.

a_in <= '0';

b_in <= '0';

assert false report "Test Done" Severity error.

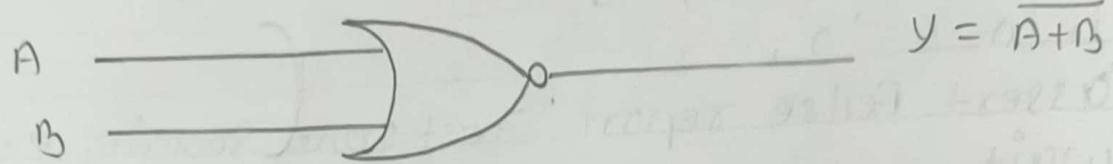
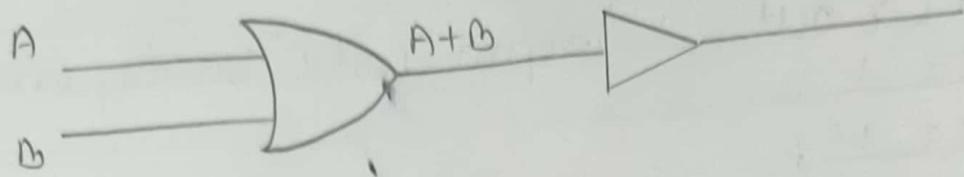
wait;

end process;

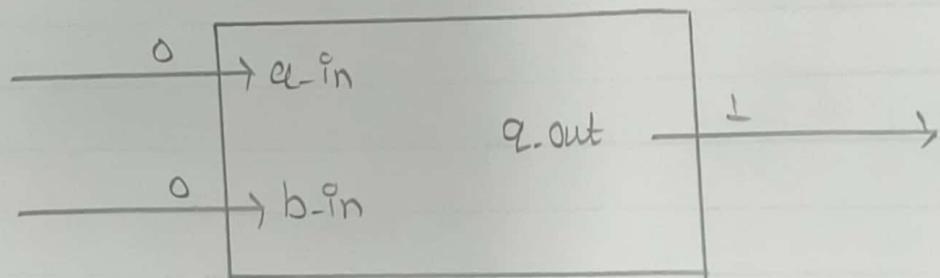
end tb;

Teacher's Signature _____

$$y = \overline{A+B}$$



NOR gate



NOR GATE :-

- **Objective :-** To design and stimulate the NOR gate circuit.
- **Description :-** NOR gate is also a Universal gate. So, we can also form all the basic gates using the NOR gate. The NOR gate is the combination of NOT-OR gate. Output state of NOR gate will be high only where all of the input are low. The logical Boolean expression for the NOR gate is complement of logical multiplication of inputs denoted by the plus sign us.

$$Y = (A+B)'$$

Input		Output
A	B	y
0	0	1
0	1	0
1	0	0
1	1	0

Teacher's Signature _____

- VHDL Code :-

```
library IEEE;
use IEEE.std_logic_1164.all;
entity norgate is
port (
    a : in std_logic;
    b : in std_logic;
    q : out std_logic);
end norgate;
```

architecture atri1 of norgate is
begin

```
process(a,b) is
begin
    q <= a nor b;
end process;
end atri1
```

-- Test bench --

```
library IEEE;
use IEEE.std_logic_1164.all;
entity norbench is
end norbench;
```

Architecture th of norbench is

Teacher's Signature _____

Component norgate is
port (

a : in std_logic ;
b : in std_logic ;
q : out std_logic);

end component;

Signal a_in, b_in, q_out : std_logic;

begin

unit : norgate port map (a_in, b_in, q_out);

process

begin

a_in <= '0' ;

b_in <= '0' ;

wait for 100ns ;

assert (q_out = '0') report "fail." severity error.

a_in <= '0' ;

b_in <= '1' ;

wait for 100ns ;

assert (q_out = '1') report "fail." severity error.

a_in <= '1' ;

b_in <= '0' ;

wait for 100ns ;

assert (q_out = '1') report "fail." severity error.

a_in <= '1' ;

b_in <= '1' ;

wait for 100ns ;

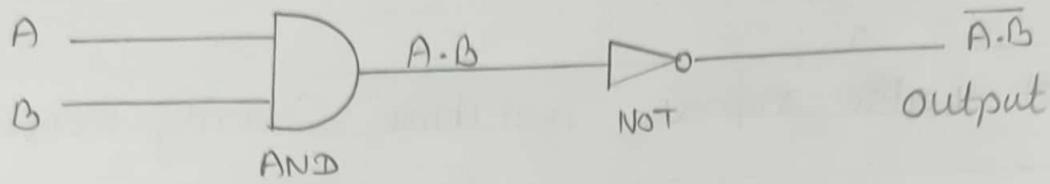
assert (q_out = '0') report "fail." severity error.

Teacher's Signature _____

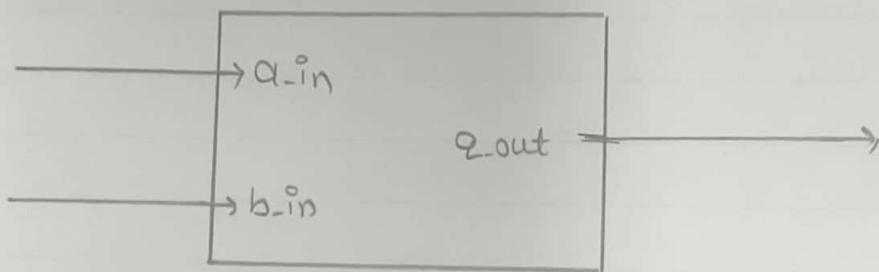
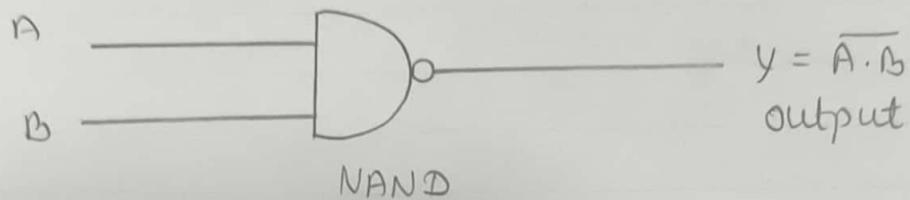
```
a in <= '0';
b in <= '0';
assert False report "Test done" Severity error
Wait,
end process;
end th;
```

Teacher's Signature _____

Inputs



Inputs



NAND GATE

- Objective :- To design and stimulate NAND gate.
- Description :- NAND gate is the combination of NOT-AND gate. Output state of NAND gate will be low only when all the inputs are high.
Simply, this gate returns the complement result of the AND gate.
The logic or Boolean expression for the NAND gate is the complement of logical multiplication of inputs denoted by a full stop on a single dot as

$$Y = (A \cdot B)'$$

- Truth Table :-

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

Teacher's Signature _____

VHDL Code :

```
library IEEE;
use IEEE.std_logic_1164.all;
entity nandgate is
port (
    a : in std_logic;
    b : in std_logic;
    q : out std_logic);
end nandgate;
architecture ar1 of nandgate is
begin
process(a,b) is
begin
    q <= a nand b;
end process;
end ar1;
```

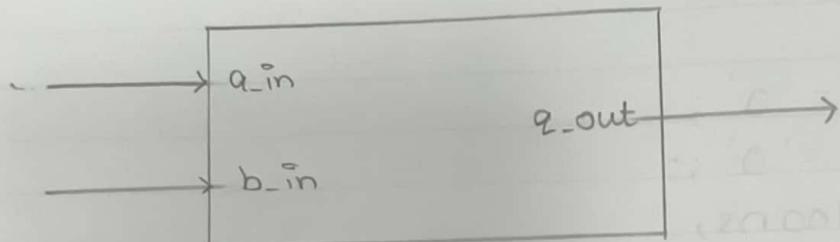
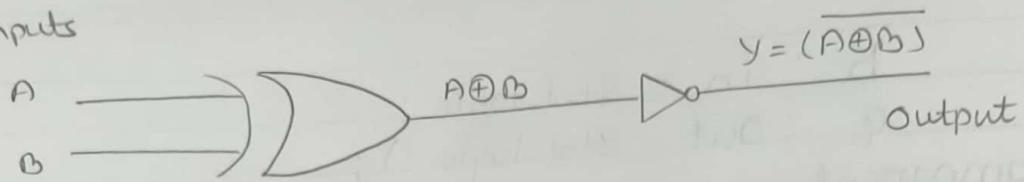
-- Test Bench --

```
library IEEE;
use IEEE.std_logic_1164.all;
entity nandbench is
end nandbench;
architecture tb of nandbench is
Component nandgate is
port (
    a : in std_logic;
```

```
b : in std_logic;  
q : out std_logic );  
end component;  
signal a_in, b_in, q_out : std_logic;  
begin  
unit : handgate port map (a_in, b_in, q_out);  
process  
begin  
a_in <='0';  
b_in <='0';  
wait for 100ns;  
assert (q_out='0') report "fail." severity error;  
a_in <='0';  
b_in <='1';  
wait for 100ns;  
assert (q_out='0') report "fail." severity error;  
a_in <='1';  
b_in <='0';  
wait for 100ns;  
assert (q_out='0') report "fail." severity error;  
a_in <='1';  
b_in <='1';  
wait for 100ns;  
wait;  
end process;  
end tb;
```

Teacher's Signature _____

Inputs



• XNOR Gate

- Objective :- To design and simulate the XNOR gate circuit.
- Description :- The XNOR gate is the complement of the XOR gate. It is a hybrid gate. Simply, It is combination of the XOR gate and NOT gate.

The output level of the XNOR gate is high only when both of its inputs are the same either 0 or 1. Symbol of XNOR gate is the same as XOR, only complement sign is added. XNOR gate is also called the equivalence gate

$$y = (A \oplus B)'$$

• Truth Table :

Input		Output $y = \overline{A \oplus B}$
A	B	
0	0	1
0	1	0
1	0	0
1	1	1

Teacher's Signature _____

• VHDL Code :

```

library IEEE;
use IEEE.std_logic_1164.all;
entity Xnorgate is
port (
    a : in std_logic;
    b : in std_logic;
    q : out std_logic);
end Xnorgate;
architecture ar1 of Xnorgate is
begin
    q <= a xnor b
end ar1;

```

-- Test bench --

```

library IEEE;
use IEEE.std_logic_1164.all;
entity Xnorbench is
end Xnorbench;
architecture tb of Xnorbench is
component Xnorgate is
port(
    a : in std_logic;
    b : in std_logic;
    q : out std_logic);
end component;

```

Signal a in, b in, q out : std_logic ;

begin

unit : Xnorgate port map (a in, b in, q out);

process

begin

a in <='0' ;

b in <='0' ;

wait for 100ns ;

a in <='0' ;

b in <='1' ;

wait for 100ns ;

a in <='1' ;

b in <='0' ;

wait for 100ns ;

a in <='1' ;

b in <='1' ;

wait for 100ns ;

end process;

end tb;

Teacher's Signature _____

INVERTER GATE :

- Objective :- To begin and Simulate the Invertor Circuits.
- Description :- The invertor, often referred to as a NOT gate, is a logic device that has an output opposite of the input. It is sometimes called 'Negator'. The output of an invertor will be complement of the input. Boolean expression for the output of this gate is

$$F = \bar{A}$$

- Truth Table :

Input	Output
A	F
0	1
1	0

- VHDL Code :-

```
library IEEE;
use IEEE.std_logic_1164.all;
entity Invertorgate is
port (
```

```

a : in std_logic;
q : out std_logic);
end invertorgate;

```

architecture ar1 of invertorgate is
begin
process(a) is
begin
q <= not a;
end process;
end ar1;

-- Test bench --

```

library IEEE;
use IEEE.std_logic_1164.all;
entity not2 is
end not2;

```

Architecture tb of not2 is
Component invertorgate is
port (
a : in std_logic;
q : out std_logic);
end Component;

Signal a_{in}, q_{out} : Std-logic;
begin

Unit : invertorgate port map (a_{in}, q_{out});

Process

begin

a_{in} <= '0';

wait for 100ns;

a_{in} <= '1';

wait for 100ns;

a_{in} <= '0';

wait;

end process;

end th;

Teacher's Signature _____

Q.2 Tm

- Adder
- object

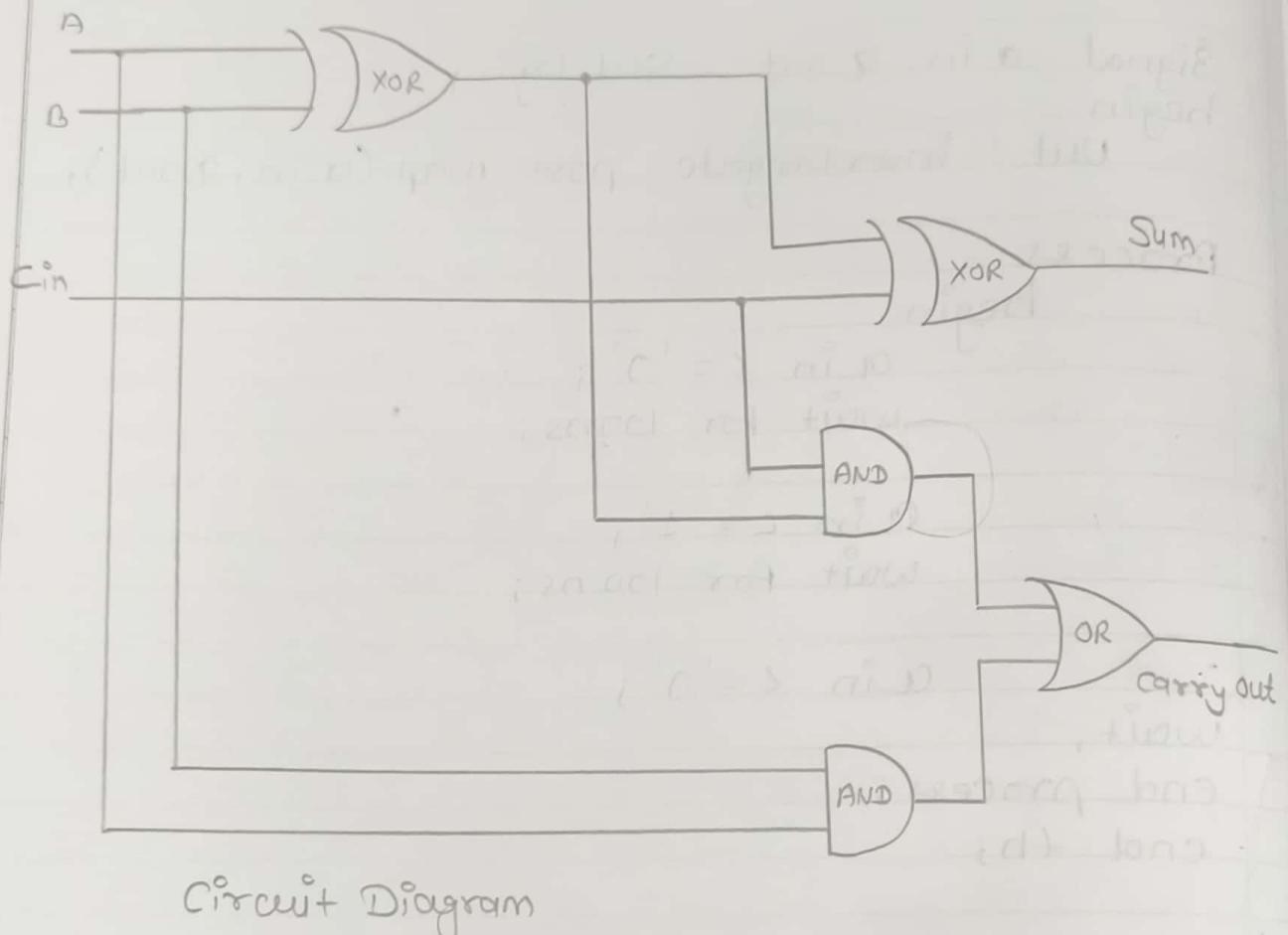
• Descr

two
and
eyThe
the
whiWe
Car

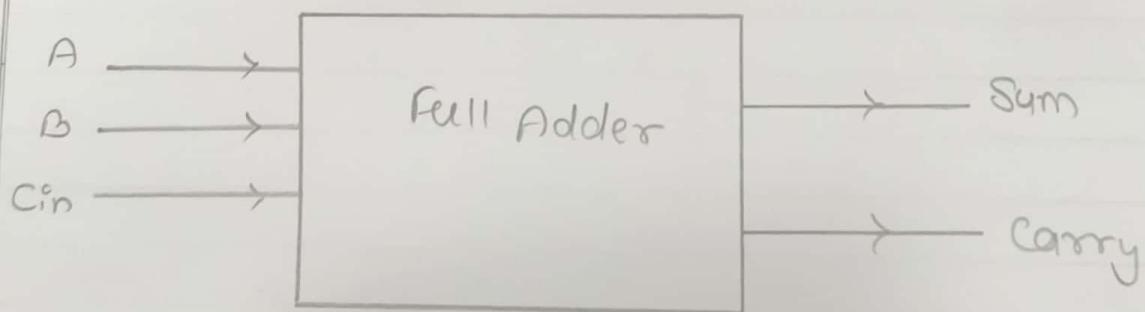
Add

A

Cir



Circuit Diagram



Q.2 Implement -

- Adder :-
- Objective :- To design and simulate the adder circuit and get knowledge of working.
- Description :- Full adder is the adder that adds three inputs and produces two outputs. The first two inputs are 'A' and 'B' and the third input is an input carry by 'c-in'.

The output Carry is designated by 'c-out' and the normal output is designated by 'S' which is sum.

We can use a full adder because when a carry in bit is available, another 1-bit adder must be used since a 1-bit half adder does not take a carry-in bit.

A 1-bit full adder adds three operands and generate 2-bit result.

- Truth Table :-

Inputs			Outputs	
A	B	Cin	Sum	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

• VHDL code :-

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;

S : out Std_logic;

Cout : out Std_logic);

end full adder;

Architecture attr of full adder is

begin

process (A, B, Cin) is

begin

Teacher's Signature _____

```
S <= A XOR B XOR Cin ;  
Cout <= (A AND B) OR (Cin AND A) OR (Cin AND B)  
end process;  
end architecture;
```

-- Test bench --

```
library IEEE;  
use IEEE.Std_logic_1164.all;  
entity adderbench is  
end adderbench;
```

```
architecture tb of adderbench is  
Component Full_adder is  
port (  
    A : in Std_logic;  
    B : in Std_logic;  
    Cin : in Std_logic;  
    S : out Std_logic;  
    Cout : out Std_logic);  
end Component;
```

```
Signal a_in : Std_logic := '0';  
Signal b_in : Std_logic := '0';  
Signal c_in : Std_logic := '0';  
Signal q_out : Std_logic;  
Signal c_out : Std_logic;
```

Teacher's Signature _____

begin

 Vut : full-adder port map (a_{in}, b_{in}, c_{in}, a_{out}, c_{out});

process

begin

 a_{in} <='0';

 b_{in} <='0';

 c_{in} <='0';

 wait for 100ns;

 a_{in} <='0';

 b_{in} <='1';

 c_{in} <='0';

 wait for 100ns;

 a_{in} <='1';

 b_{in} <='0';

 c_{in} <='0';

 wait for 100ns;

 a_{in} <='0';

 b_{in} <='0';

 c_{in} <='1';

 wait for 100ns;

 a_{in} <='1';

 b_{in} <='1';

 c_{in} <='0';

 wait for 100ns;

 a_{in} <='1';

 b_{in} <='0';

 c_{in} <='1';

Teacher's Signature _____

Wait for 100ns;

a_in c = '0' ;

b_in c = '1' ;

c_in l = '1' ;

Wait for 100 ns

a_in l = '0' ;

b_in l = '0' ;

c_in l = '0' ;

Wait ;

end process;

end tb;

Teacher's Signature _____

