Expt. No. 6

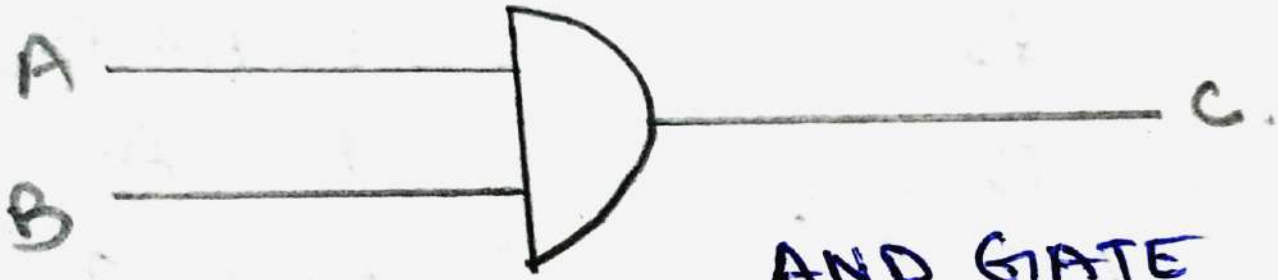
ASSIGNMENT-6

Page No. 20

1. WAP in VHDL to implement AND Gate.

```
-- library declaration.
library ieee;
-- package declaration
use ieee.std_logic_1164.all;
-- entity AND_GATE is
port (A : in std_logic;
B : in std_logic;
c : out std_logic);
end AND_GATE;
-- ARCHITECTURE DEFINITION
architecture behav of AND_GATE is
begin
c <= A AND B;
end;
```

OUTPUT.



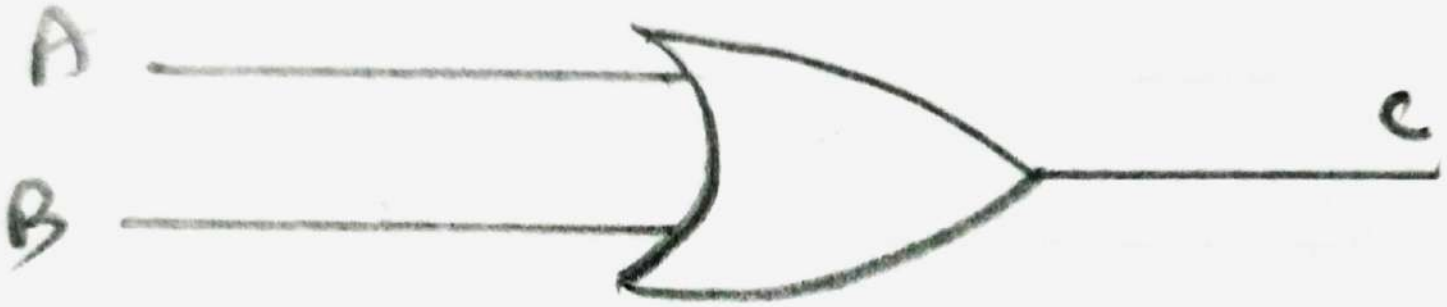
AND GATE.

2. WAP in VHDL to implement OR Gate.

```
-- library declaration
library ieee;
-- package declaration
use ieee.std_logic_1164.all;
-- entity declaration
entity OR_GATE is
port
(
A : in bit;
B : in bit;
C : out bit;
);
end OR_Gate;
-- ARCHITECTURE DEFINITION.
architecture dataflow of OR_GATE is
begin
C <= A OR B;
end dataflow;
```

Teacher

OUTPUT.



OR GATE

2) WAP in VHDL to implement NOT GATE

```
-- library declaration
library ieee;
-- package declaration
use ieee.std_logic_1164.all;
-- entity NOT_GATE is
port
(
A: in bit;
B: out bit;
);
end NOT_GATE;
-- ARCHITECTURE DEFINITION.
architecture dataflow of NOT_Gate GATE is
begin
B <= (NOT A);
end dataflow;
```

OUTPUT

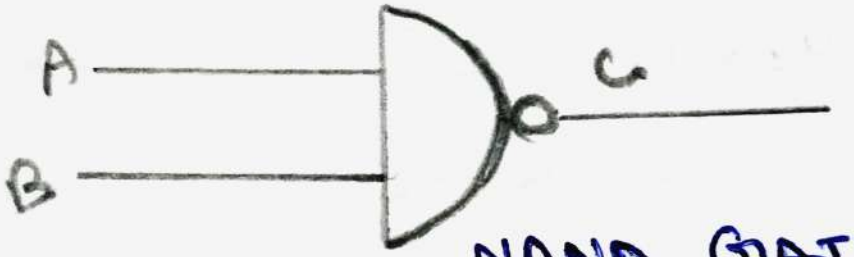
NOT GATE

4) WAP in VHDL to implement NAND GATE

```
--library declaration
library ieee;
--package declaration
use ieee.std_logic_1164.all
--entity declaration
entity NAND_GATE is
port
(
A : in bit;
B : in bit;
C : out bit;
);
end NAND_GATE;
--ARCHITECTURE DEFINITION
architecture dataflow of NAND_GATE is
begin
C <= A NAND B;
end dataflow;
```

Teacher's Signature _____

OUTPUT .

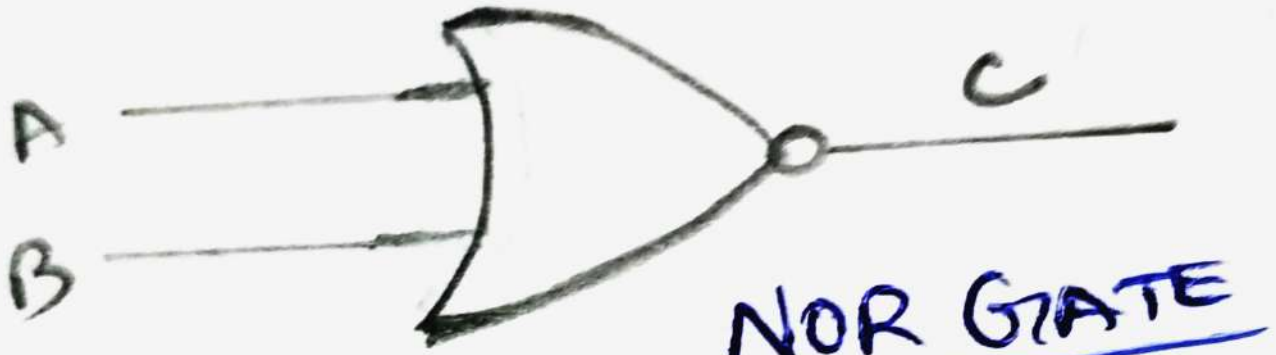


NAND GATE .

5) WAP in VHDL to implement NOR GATE

```
-- library declaration
library ieee;
-- package declaration
use ieee.std_logic_1164.all;
-- entity declaration
entity NOR_GATE is
port
(
A : in bit;
B : in bit;
C : out bit;
);
end NOR_GATE;
-- ARCHITECTURE DEFINITION.
architecture dataflow of NOR_GATE is
begin
C <= A NOR B;
end dataflow;
```

OUTPUT



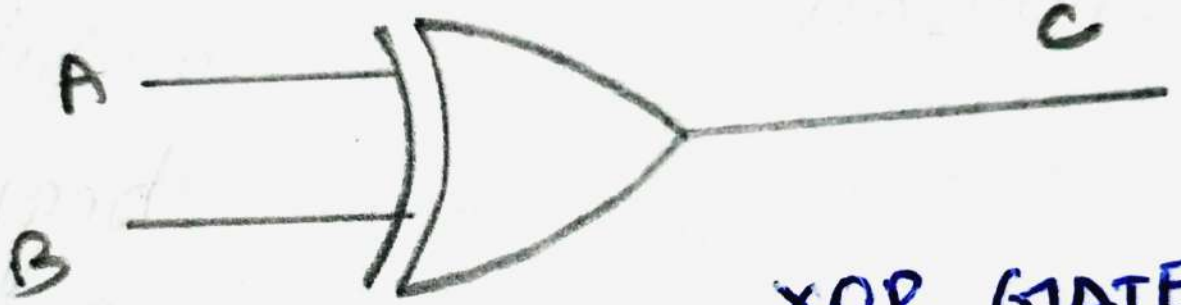
NOR GATE

6) WAP in VHDL to implement XOR GATE

```
-- library declaration
library ieee;
-- package declaration
use ieee.std_logic_1164.all;
-- entity XOR_GATE is
port
(
A : in bit;
B : in bit;
C : out bit;
);
end XOR_GATE;
-- ARCHITECTURE DEFINITION
architecture dataflow of XOR_GATE is
begin
C <= A XOR B;
end dataflow;
```

Teacher's Signature _____

OUTPUT



XOR GIATE

7) WAP in VHDL to implement XNOR GATE

```
-- library declaration
library ieee;
-- package declaration
use ieee.std_logic_1164.all;
--entity XNOR GATE is
port
(
A : in bit;
B : in bit;
C : in bit;
);
end XNOR_GATE ;
-- architecture declaration
architecture dataflow of XNOR_GATE is
begin.
C <= A XNOR B;
end dataflow;
```

OUTPUT

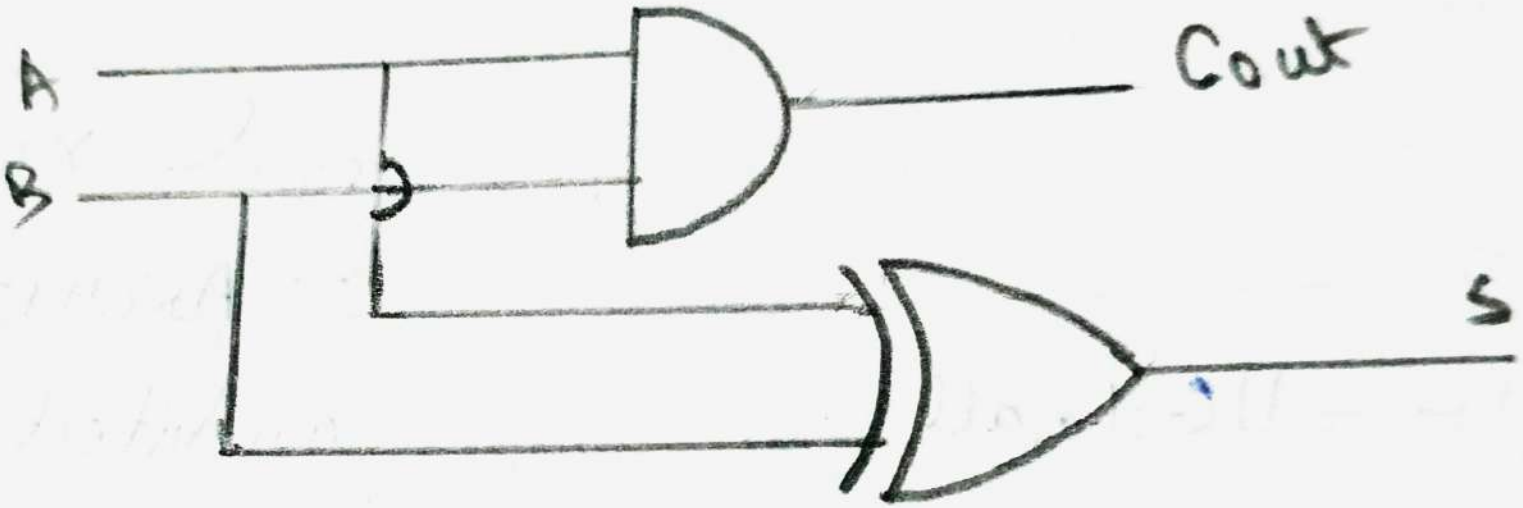
XNOR GATE

8) WAP to design a half adder in VHDL.

```
--library declaration
library IEEE;
-- package declaration
use IEE.std_logic_1164.all;
-- entity design.
entity half-adder is
port
(
A : in std_logic;
B : in std_logic;
S : out std_logic;
Cout : out std_logic;
);
end half-adder;
-- Architecture design.
architecture dataflow of half-adder
begin
S <= A XOR B;
Cout <= (A AND B);
end dataflow;
```

Teacher's Signature _____

OUTPUT

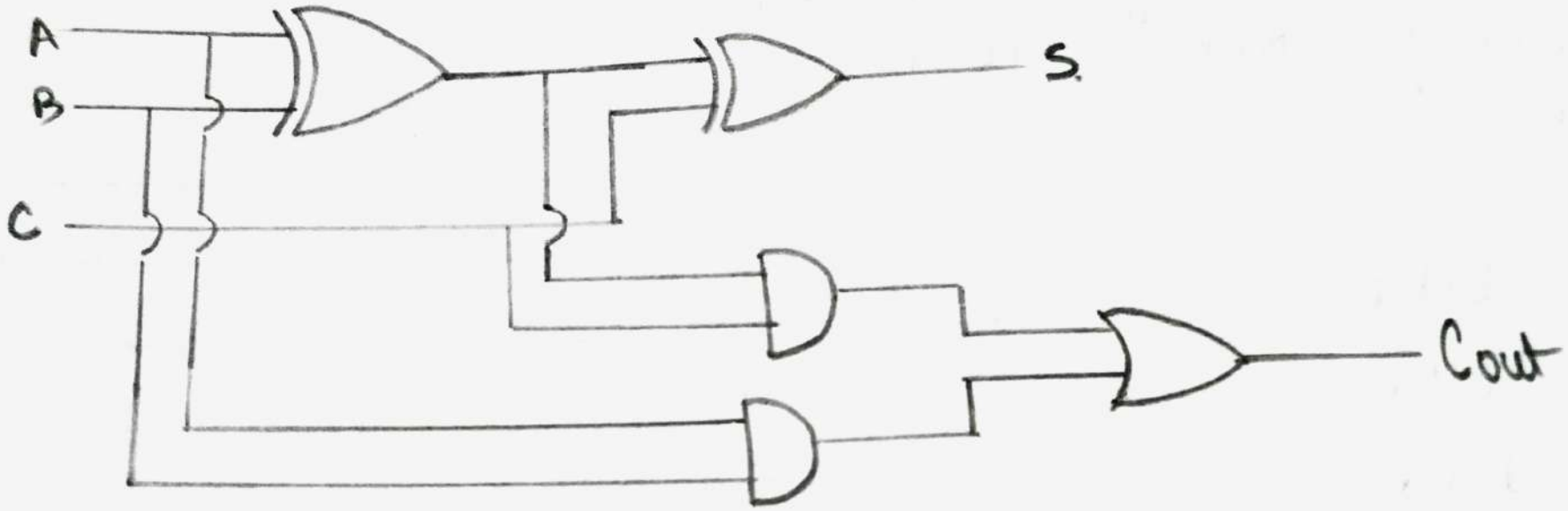


Half Adder

9) WAP in VHDL to design full adder.

```
-- library declaration
library IEEE;
-- package declaration
use IEEE.STD_LOGIC_1164.ALL;
-- entity declaration
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 declaration
architecture dataflow of full_adder is
begin
S <= A XOR B XOR Cin;
Cout <= (A AND B) OR (Cin AND A) OR (Cin AND B)
end dataflow;
```

OUTPUT



Full Adder

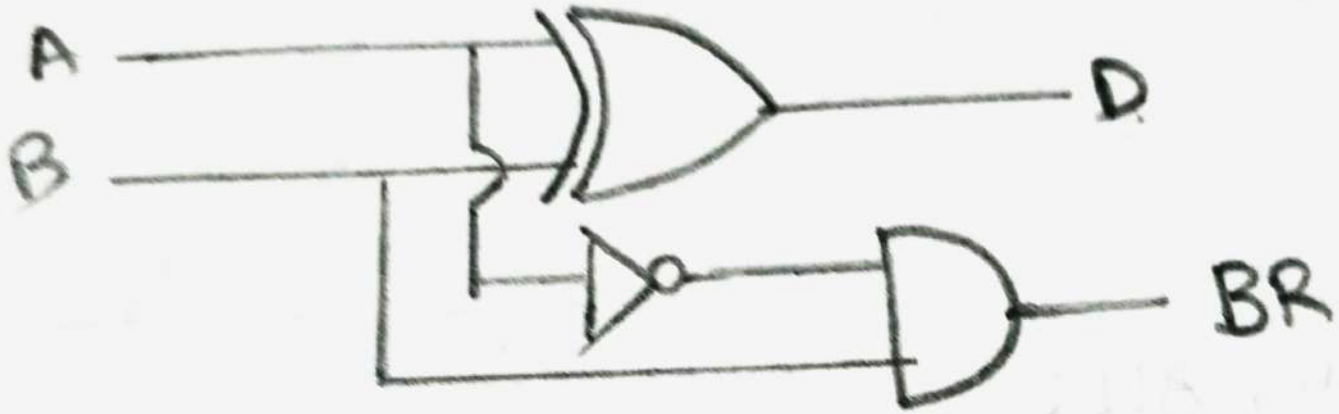
10) WAP in VHDL to design half subtractor.

```
--library declaration
Library IEEE;
--package declaration
use IEEE.STD_LOGIC_1164.ALL;
--entity declaration
entity half_sub is
Port
(
```

Teacher's Signature

```
A : in STD_LOGIC;
B : in STD_LOGIC;
D : out STD_LOGIC;
BR : Out STD_LOGIC;
);
end half_sub
--ARC Architecture declaration
architecture dataflow half_sub is
begin
D <= A XOR B;
BR <= (NOT A) AND B;
end dataflow;
```

OUTPUT



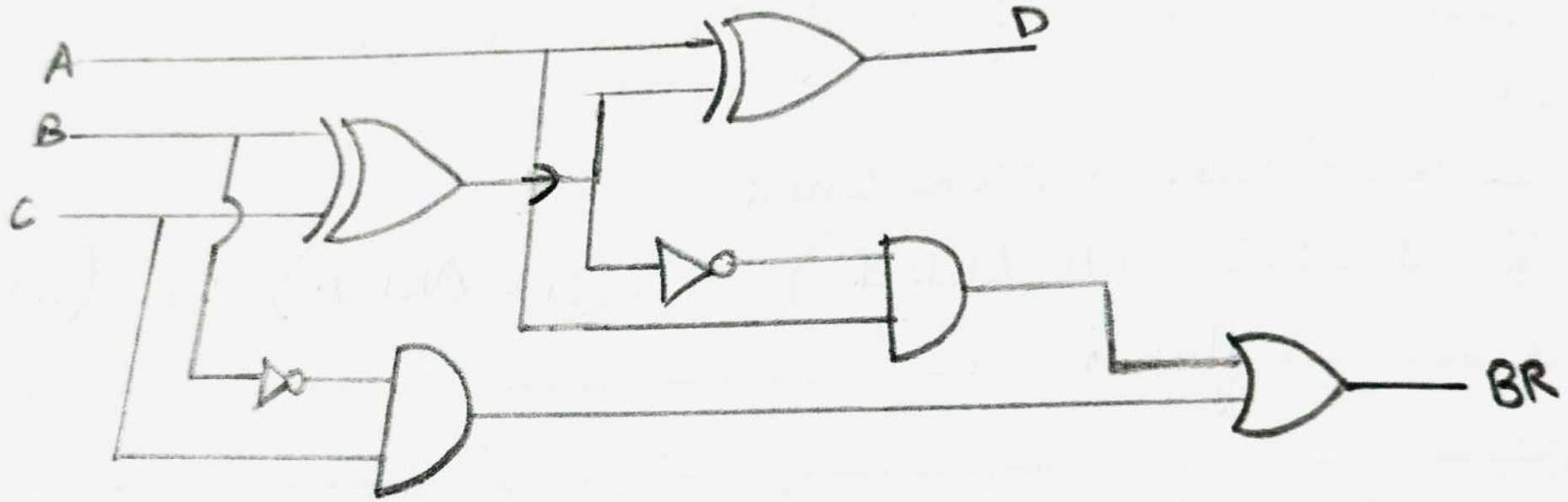
Half Subtractor

11.) WAP in VHDL to design full subtractor.

```
--library IEEE;
use IEEE STD_LOGIC_1164.ALL;
use IEEE STD_LOGIC_ARICH.ALL;
use IEEE STD_LOGIC_UNSIGNED.ALL;
--entity FULL_SUBTRACTOR is
port
(
A,B, ~~Cin~~ C: in STD_LOGIC;
DIFFERENCE, BORROW : out STD_LOGIC_VECTOR
);
end FULL_SUBTRACTOR;
architecture dataflow of FULL_SUBTRACTOR is
begin
DIFFERENCE (D) <= A XOR B XOR C;
BORROW (O) <= ((NOT A) AND (B OR C)) OR (B AND C);
end dataflow;
```

Teacher's Signature _____

OUTPUT

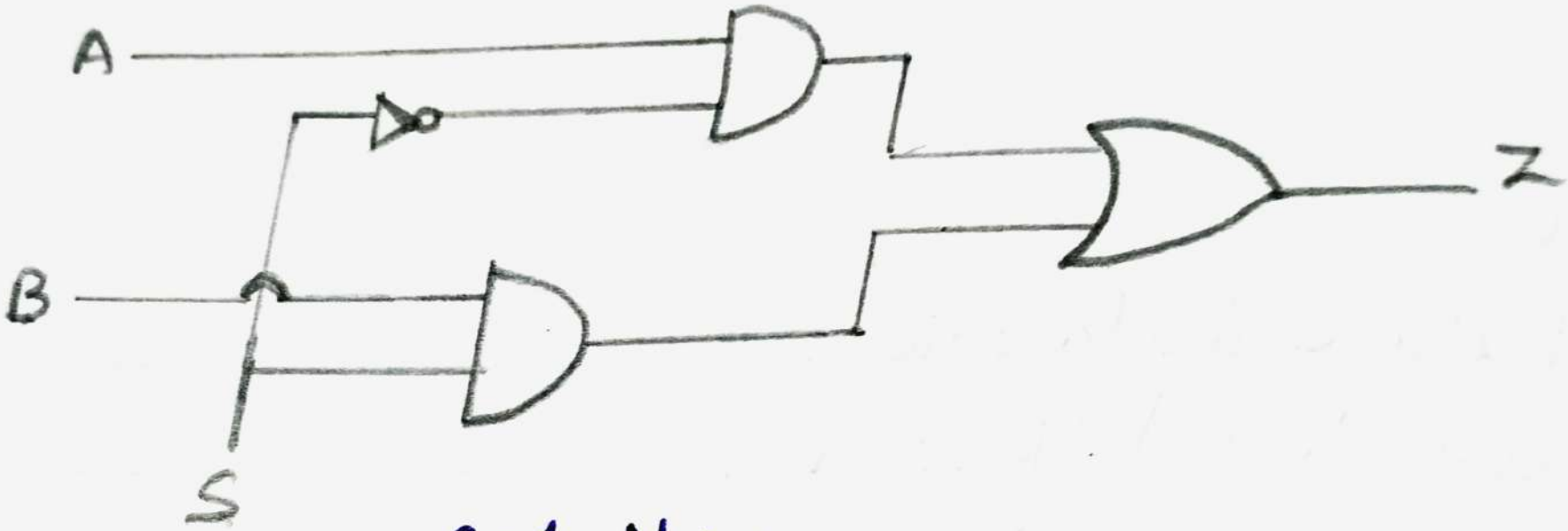


Full Subtractor

(2) WAP in VHDL to design 2x1 MUX

```
-- library ieee;
use ieee.std_logic_1164.all;
entity mux_2to1 is
port (
a, b : in std_logic;
s : in std_logic;
z : out std_logic;
);
end mux_2to1;
architecture behave of mux_2to1 is
begin
    process (a, b, s)
        begin
            if (s = '0') then
            z <= a;
            else if (s = '1') then
              z <= b;
            end if
        end process;
    end behave;
```

Output



2:1 Mux

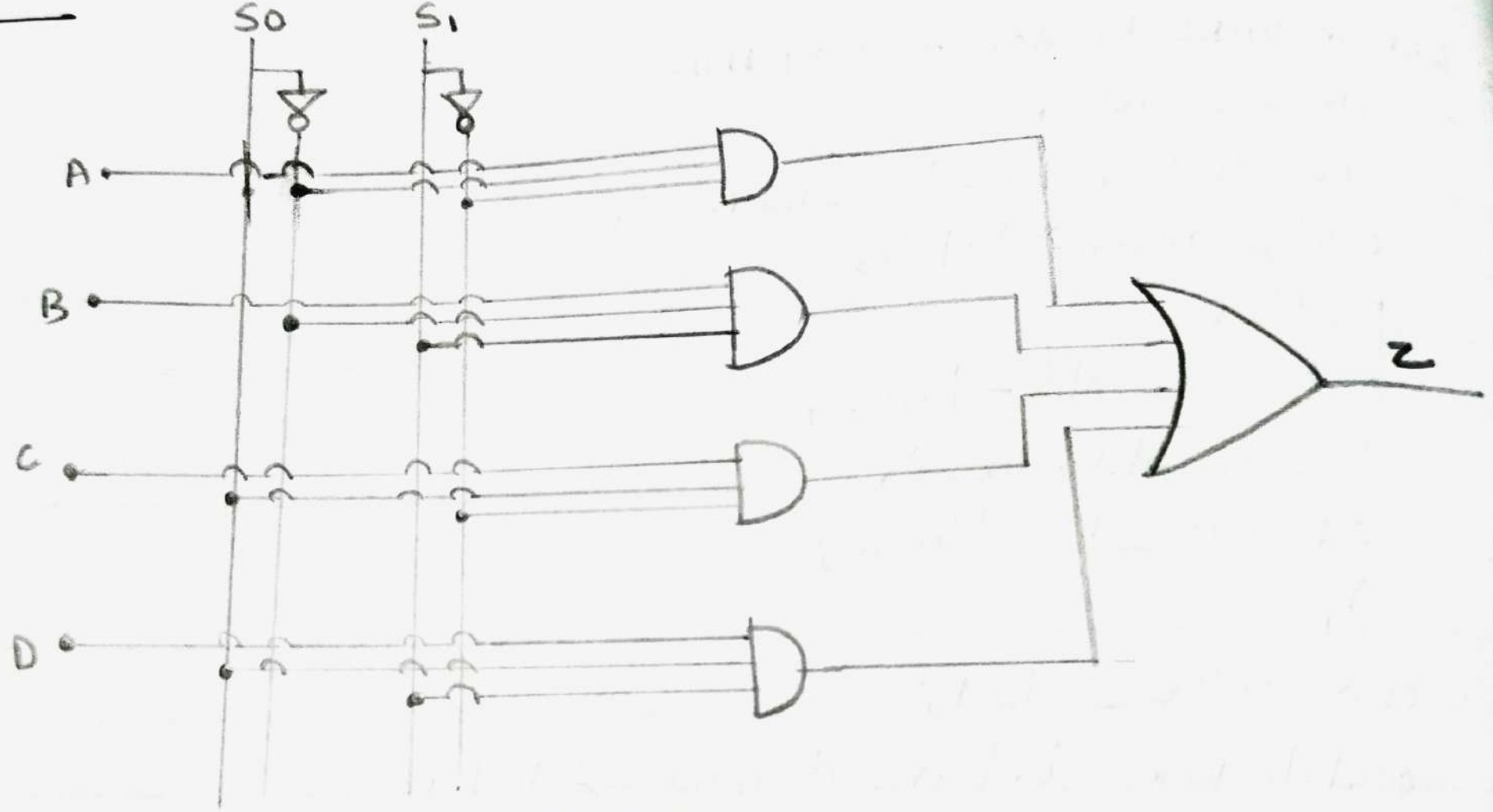
13) WAP in VHDL to design 4x1 mux.

```
library ieee;
use ieee.std_logic_1164.all;
entity mux 4 to 1 is
port (
a, b, c, d : in std_logic;
s0, s1 : in std_logic;
z : out std_logic
);
```

Teacher's Signature _____

```
end mux_4 to 1;
architecture behave of mux-4 to 1 is
begin
   process (a, b, c, d, s0, s1)
    begin.
       if (s0 = '0' and s1 = '0') then
          z <= a;
       elseif (s0 = '0' and s1 = '1') then
          z <= b;
       elseif (s0 = '1' and s1 = '0') then.
          z <= c;
       elseif (s0 = '1' and s1 = '1') then.
          z <= d;
       endif;
    end process;
end behave;
```

OUTPUT



4:1 MUX

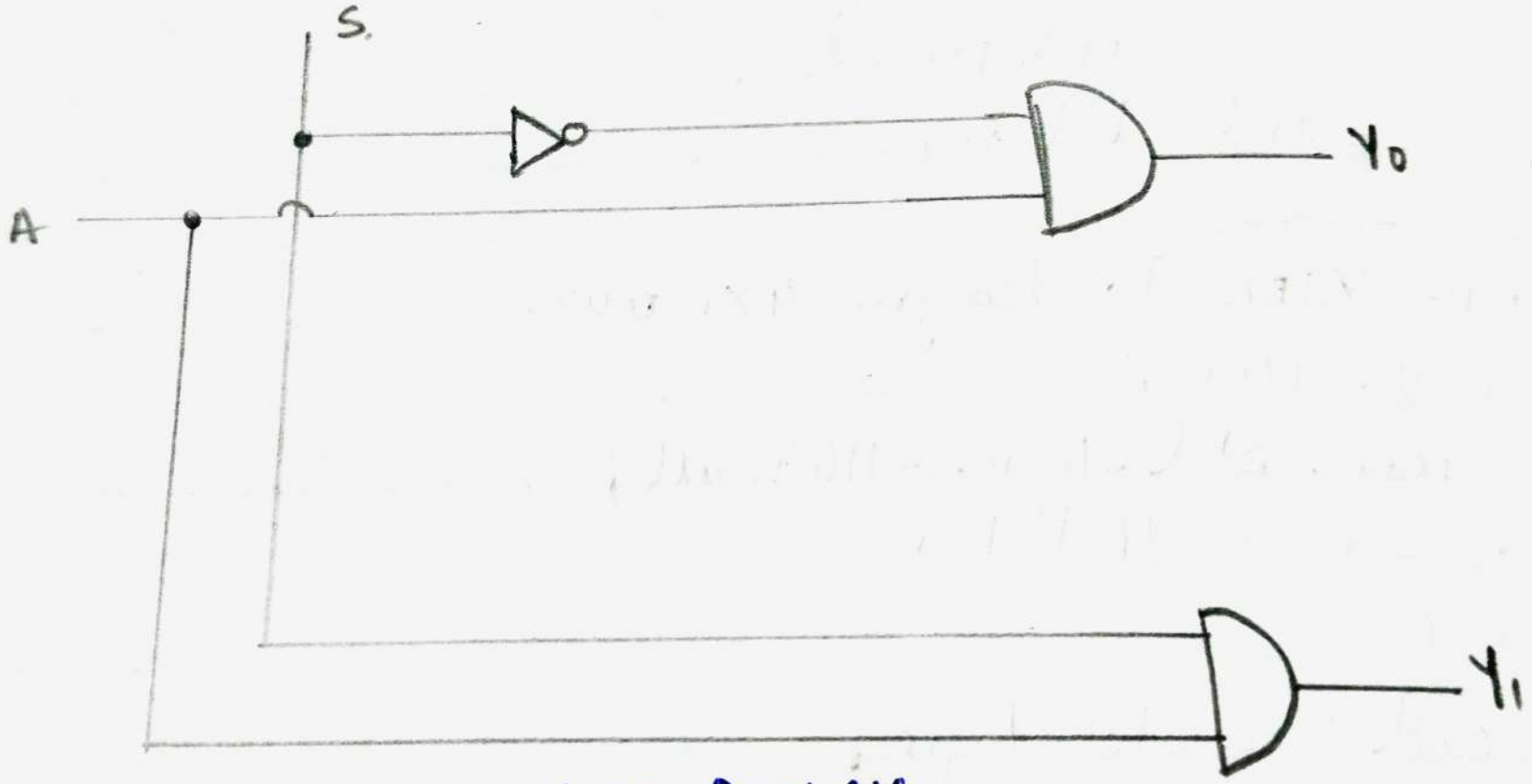
14) WAP in VHDL to design 1X2 D-MUX

```
library ieee;
entity demux-1X2 is
  port (
  input : in std_logic;
  select : in std_logic;
  output_0 : out std_logic;
  output_1 : out std_logic;
  );
  end demux_1X2;
  architectural Behavioral of demux_1X2 is
  begin
```

Teacher's Signature

```
process (input, select)
begin
    if select = '0' then
        output - 0 <= input;
        output - 1 <= '0';
    else
        output - 0 <= '0';
        output - 1 <= input;
    endif;
end process;
end Behavioral;
```

OUTPUT



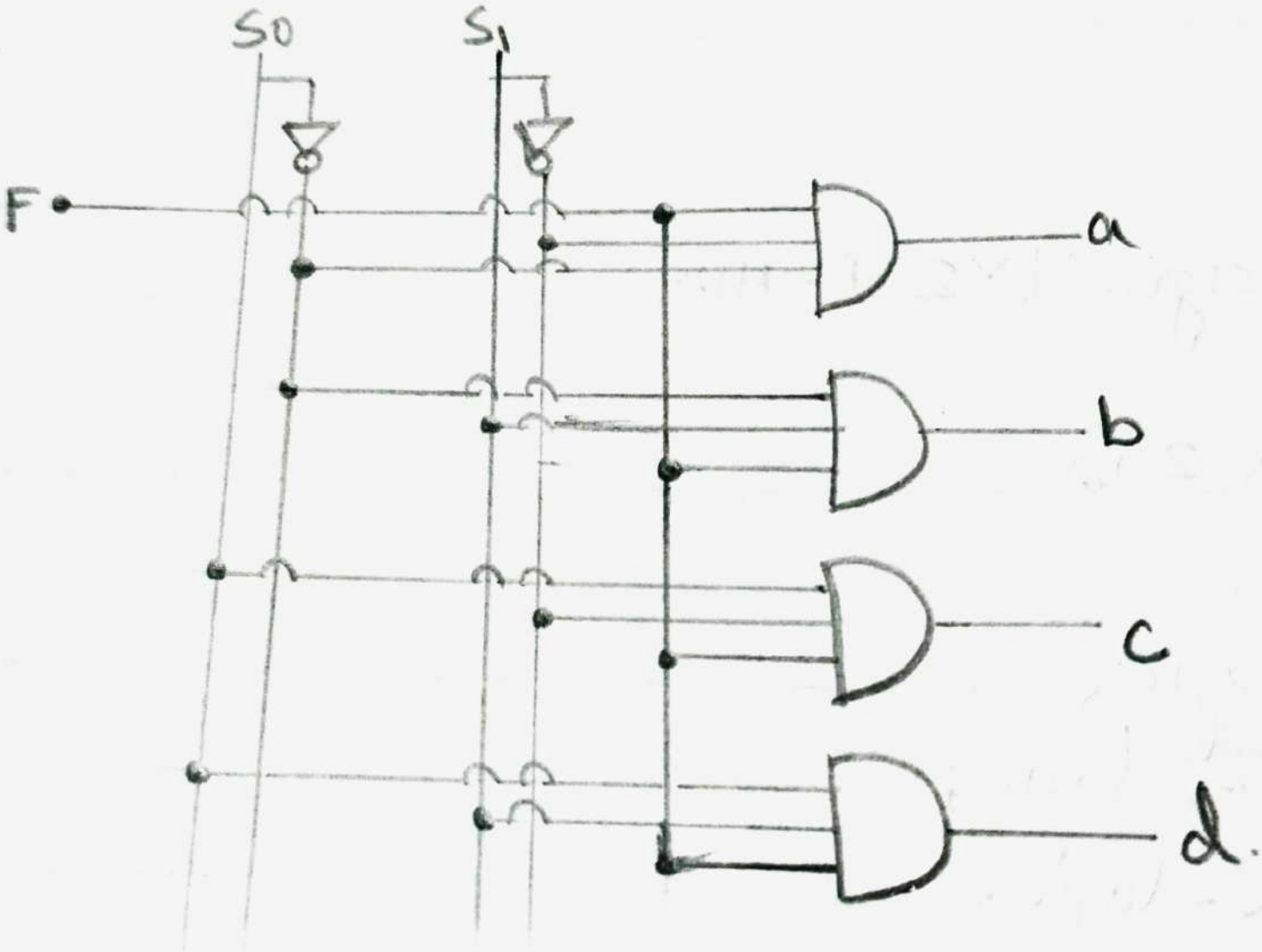
1:2 DEMUX

15) WAP in VHDL to design 1x4 D-Mux.

```
library ieee;
use ieee.std_logic_1164.all;
entity demux_1to4 is
port (
    f : in std_logic;
    s0, s1 : in std_logic;
    a, b, c, d : out std_logic
    );
end demux_1to4;
architecture behave of demux_1to4 is
    begin
        process (f, s0, s1) is
        begin
            if (s0='0' and s1='0') then
            a <= f;
            elsif (s0='0' and s1='1') then
            b <= f;
            elsif (s0='1' and s1='0') then
            c <= f;
            elsif (s0='1' and s1='1') then
            d <= f;
            endif;
        end process;
    end behave;
```

Teacher's Signature

OUTPUT



1:4 DEMUX