BRAINWARE UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Lab Assignment

of

Course Name: Computer Organization and Architechture Lab.

Course Code: PCC-CSM391

Name: ARPA KUNDU

Student Code: BWU/BTA/21/020

Program Name: B.Tech CSE (AI & ML)

Semester: III

University Roll No: 21010332020

Registration No: 21013001180 of 2021-2022

Scanned by TapScanner

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Name: Arpa Kundu

Student Code: BWU/BTA/21/020

Course Code: PCC-CSM391

Sl. No	Topic Name	Date of Experiment	Signature with Date	Remarks
01.	Write a VHDL code to implement AND Gate	17/08/22	AKM 17/08/2022	
02.	Write a VHDL code to implement OR Gate	17/08/22		
03.	Write a VHDL code to implement NOT Gate	17/08/22		
04.	Write a VHDL code to implement NAND Gate	17/08/22		
05.	Write a VHDL code to implement NOR Gate	17/08/22		
06.	Write a VHDL code to implement XOR Gate	24/08/22	AKM 24/08/2022	
07.	Write a VHDL code to implement XNOR Gate	24/08/22		
08.	Write a VHDL code to implement AND Gate through dataflow model	24/08/22	AKM 24/08/2022	
09.	Write a VHDL code to implement OR Gate through dataflow model	24/08/22		
10.	Write a VHDL code to implement NOT Gate through data flow model	24/08/22		
11.	Write a VHDL code to implement Half-adder	07/09/22	AKM 07/09/2022	
12.	Write a VHDL code to implement half-adder through data flow model	07/09/22		
13.	Write a VHDL code to implement Full-adder through data flow model	07/09/22	AKM 14/09/2022	
14.	Write a VHDL code to implement 8:1 multiplexer	14/09/2022	AKM 21/09/2022	
15.	Write a VHDL code to implement 3:8 decoder	21/09/2022		
16.	Write a VHDL code to implement carry look ahead adder	28/09/2022	AKM 28/09/2022	
17.	Write a VHDL code to implement De-Multiplexer	12/10/2022	AKM 12/10/2022	

Scanned by TapScanner

Department of Computer Science and Engineering

Name:

Student Code:

Course Code:

Sl. No	Topic Name	Date of Experiment	Signature with Date	Remarks
18.	Write a VHDL code to implement Encoder.	19/10/2022	SKM 19/10/2022	
19.	Write a VHDL code to implement NAND gate through data flow model	28/09/2022	SKM 02/11/2022	
20.	Write a VHDL code to implement NOR gate through data flow model	28/09/2022		
21.	Write a VHDL code to implement XOR gate through data flow model	28/09/2022		
22.	Write a VHDL code to implement XNOR gate through data flow model	28/09/2022		
23.	Write a VHDL code to implement Half-Subtractor	02/11/2022	SKM 02/11/2022	
24.	Write a VHDL code to implement Full-Subtractor	02/11/2022	SKM 02/11/2022	

Scanned by TapScanner

d1

Aim: Write a VHDL code to implement AND Gate.

Description:

Definition of AND Gate: The AND gate is a basic logic gate that implements logical conjunction from mathematical logic. A high output (1) results only if all the inputs to the AND gate are high, if not all inputs are high then low output (0) occurs.

Uses:

- The main AND gate application is in digital electronics and in data transmission. Where AND gate helps for allowing or not allowing the data through a channel.
- Used in digital measuring devices.
- Used in alarm circuits.
- It is also used to inhibit gate, this is the reverse approach of enabling gate.
- It is used in automatic temperature-regulating circuits and in freezer warning buzzer devices.

Truth Table:

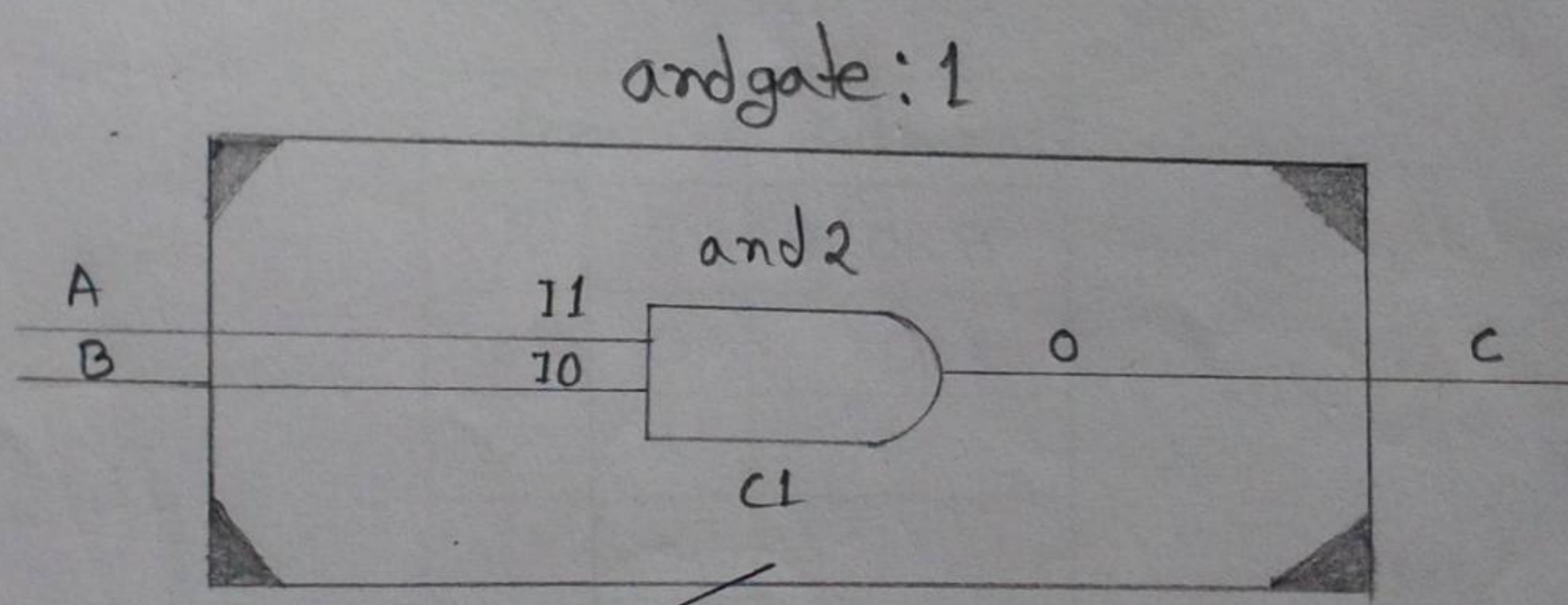
Input		Output
A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

Scanned by TapScanner

VHDL Code:

```
entity andgate is
    port (A : in STD-LOGIC;
          B : in STD_LOGIC;
          C : out STD_LOGIC).

end andgate;

architecture Behavioral of andgate is
begin
C <= A and B;
end Behavioral;
```

Output:



17/08/2022

Scanned by TapScanner

Aim: Write a VHDL code to implement OR Gate.

Description:

Definition of OR Gate: The OR gate is a digital logic gate that implements logical disjunction from mathematical logic. A High output (1) results if one or both the inputs to the gate are High(1), If neither input is high, a low output (0) results.

Uses: OR gates are used in automatic digital circuits. There are many sensors are used to detect faults in an industry. The sensor creates a signal if any fault occurs. Then OR Gates give command signals to the main processor for the shutdown. OR gates are used in Door Bell switches also.

Truth Table:

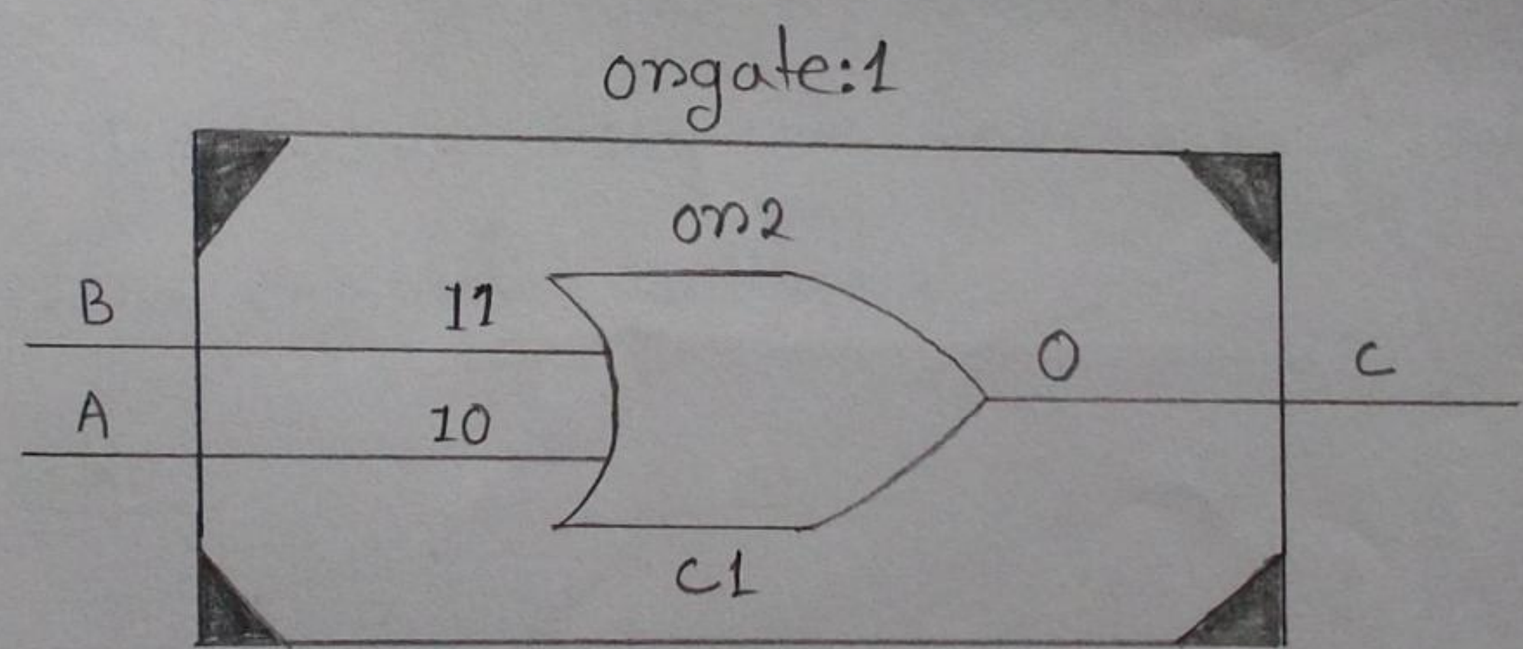
Input		Output
A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

Scanned by TapScanner

VHDL Code:

```
entity orgate is
    port (A: in STD_LOGIC;
          B: in STD_LOGIC;
          C: out STD_LOGIC);

end orgate;

architecture Behavioral of orgate is
begin
C<=A or B;
end Behavioral;
```

Output:



17/08/2022

Scanned by TapScanner

Aim: Write a VHDL code to implement NOT Gate.

Description:

Definition of NOT Gate: A NOT Gate is a logic gate. Each NOT gate has only one input signal. Logically with NOT gates, the input and the output swap. It will always output the opposite signal. Its main function is to interchang logic.

Uses:

- The foremost not gate application is in the CMOS inverter for the generation of waveforms.
- Employed in temperature detection devices.
- Used in daily usage devices like desktops, tablets and mobiles.
- Complex devices such as decoders, state machines, and multiplexers also use inverter circuits.
- Used in a crystal oscillator.

Truth Table:

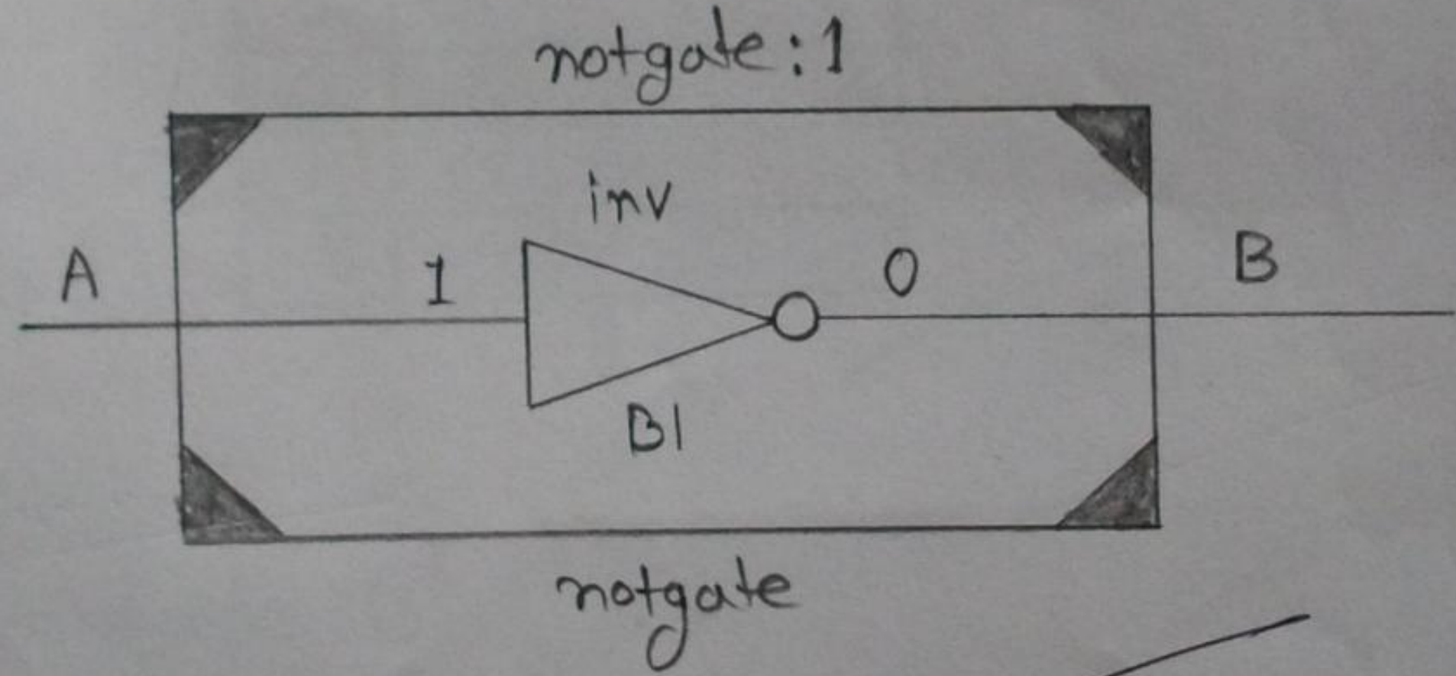
Input	Output
A	NOT A
0	1
1	0

Scanned by TapScanner

VHDL Code:

```
entity notgate is
    Port (A: in STD_LOGIC;
          B: out STD_LOGIC);
end notgate;
architecture Behavioral of notgate is

begin
B <= NOT A;
end Behavioral;
```

Output:



17/08/2022

Scanned by TapScanner

Aim: Write a VHDL Code to implement the NAND Gate.

Description: A NAND Gate is a logical gat which is the opposite of an AND logic gate. It is a combination of AND and NOT gates and is commonly used logic gate. Considered as a 'universal' gate.

Uses: It is widely used in applications like buglar alarms, freezer warning buzzers and automated circuits.

Truth Table:

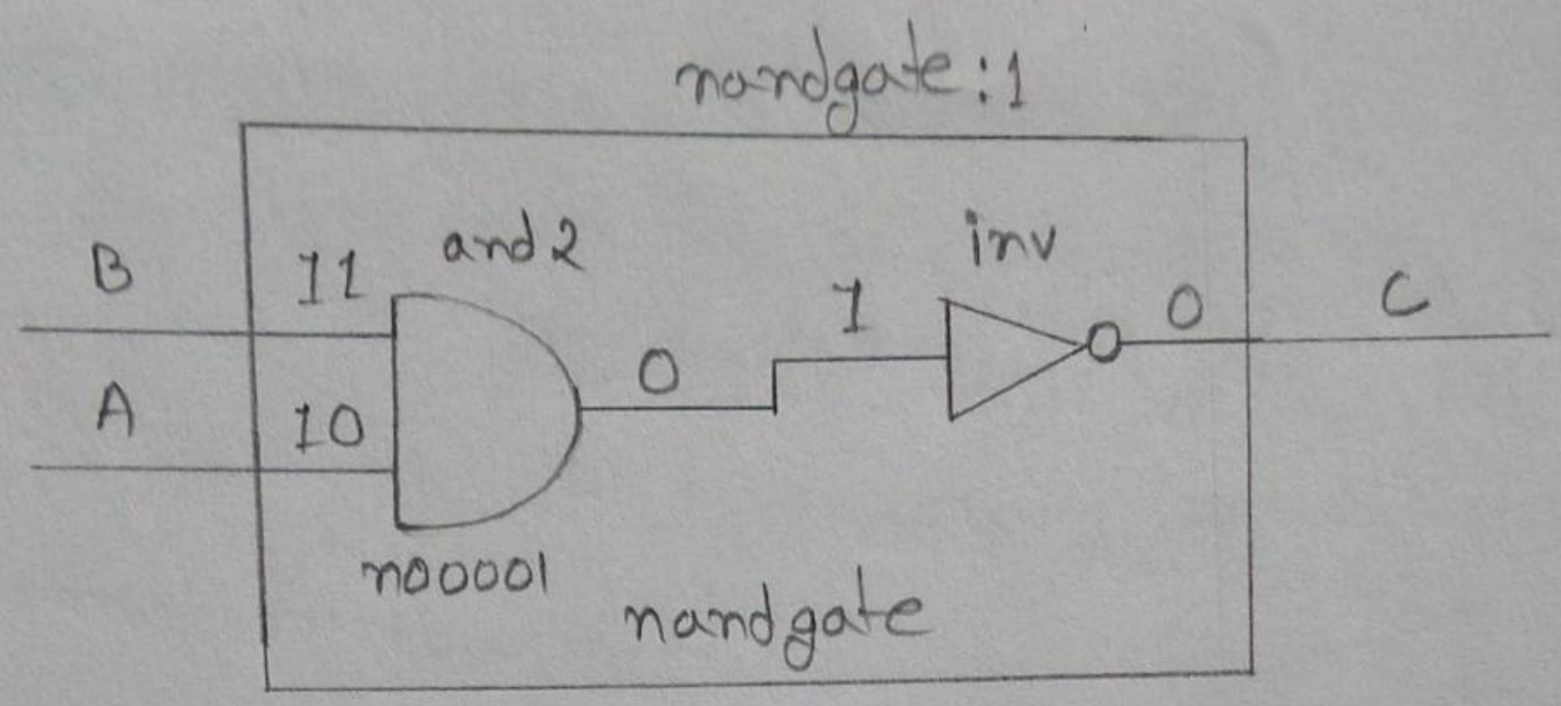
Input		Output
A	B	$C = \overline{A \text{ AND } B}$
0	0	1
0	1	1
1	0	1
1	1	0

VHDL Code:

```
entity nandgate is
    port (A: in STD_LOGIC;
          B: in STD_LOGIC;
          C: out STD_LOGIC);

end nandgate;
architecture Behavioral of nandgate is
begin
C<=B NAND A;
end Behavioral;
```

Scanned by TapScanner

Output:



SKM
17/08/2022

Scanned by TapScanner

Aim: Write a VHDL Code to implement NOR Gate.

Description: A NOR Gate is a type of logic gate that works on the principle of ~~me~~ neither this nor that'. This type of digital logic gate produces high output only if two binary results are satisfied by a zero or low input.

Uses:
- NOR Gates are used in combinational circuits such as multiplexers, half and full adders and in ripple - carry adders.
- Also used in sequential circuits and in shift registers also.

Truth Table:

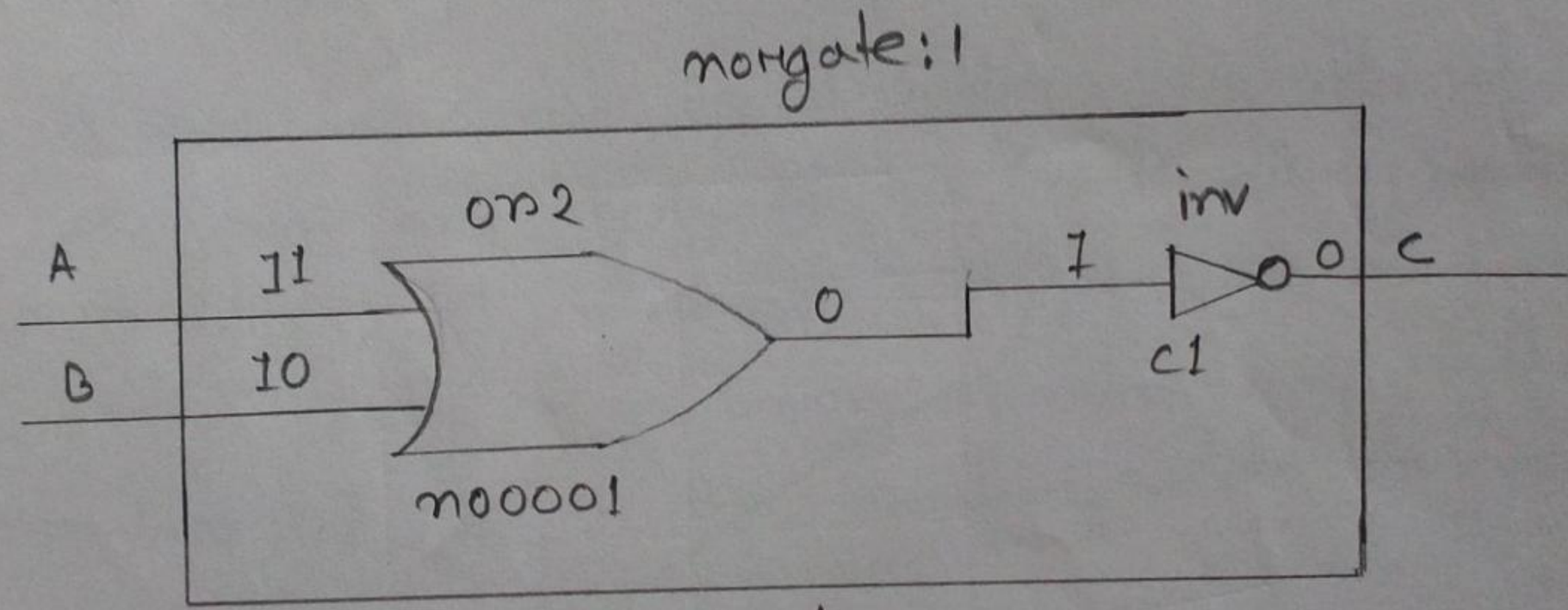
Input		Output
A	B	$C = \overline{A \text{ OR } B}$
0	0	1
0	1	0
1	0	0
1	1	0

Scanned by TapScanner

VHDL code:

```
entity norgate is
    port (A: in STD_LOGIC;
          B: in STD_LOGIC;
          C: out STD_LOGIC);

end norgate;
architecture Behavioral of norgate is

begin
C<= B NOR A;
end Behavioral;
```

Output:



17/08/2022

Scanned by TapScanner

Aim: Write a VHDL Code to implement X-OR Gate.

Description:

Definition: X-OR gate (sometimes EOR, EXOR and pronounced as Exclusise OR) is a digital logic gate that gives a true (1 or HIGH) output when the number of true inputs is odd. An XOR gate implements an exclusive ($\leftrightarrow$) from mathematical logic; that is, a true output results if one, and only one, of the inputs to the gate is true. If both inputs are flase (0/LOW) or both are true, a false output results. XOR represents the inequality function, i.e., the output is true if the inputs are not alike otherwise the output is false. A way to remember XOR is "must have one or the other but not both". The algebric expressions $A.\overline{B} + \overline{A}.B$ or, $(A+B).(\overline{A}+\overline{B})$ or $A \oplus B$ all represent the XOR gate with inputs A and B.

Uses: It is used in simple digital addition circuits which calculate the sum and carry of two (half-adder) or three (full-adder) bit numbers. XOR gates are used to determine the parity of a binary number, i.e., if the total number of 1's in the number is odd or even.

Truth Table:

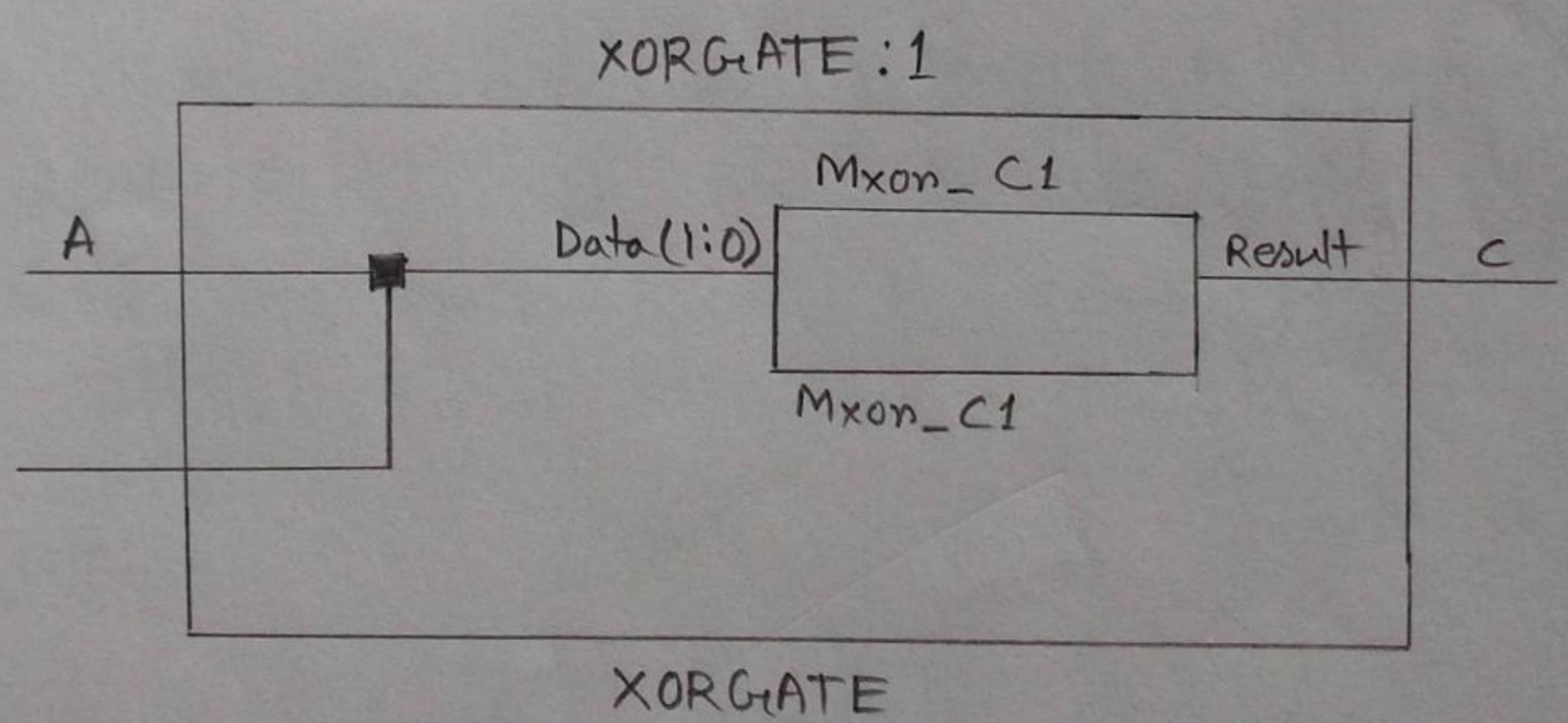
Input		Output
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Scanned by TapScanner

VHDL Code:

```
entity XORGATE is
    port ( A: in STD_LOGIC;
           B: in STD_LOGIC;
           C: out STD_LOGIC);
end XORGATE;

architecture Behavioral of XORGATE is
begin
C<= B XOR A;
end Behavioral;
```

Output:



24/08/2022

Scanned by TapScanner

Aim: Write a VHDL code to implement the XNOR Gate.

Description:

Definition: The XNOR gate (sometimes ENOR, EXNOR or NXOR and pronounced as Exclusive NOR is a digital logic gate whose function is the logical complement of the Exclusive OR gate. It is equivalent to the logical connective (↔) from mathematical logic, also known as the material biconditional. A HIGH output (1) results if both of the inputs to the gate are the same. If one but not both inputs are high (1), a low output (0) results. The algebraic notation used to represent the XNOR operation is $S = A \odot B$. The algebraic expressions $(A+\bar{B}).(\bar{A}+B)$ and $A \cdot B + \bar{A}.\bar{B}$ bothe represent the XNOR gate with inputs A and B.

Uses: The XNOR gates are used in error detecting circuits which are to detect odd parity or even parity bits in digital data transmission circuits. This gates are mainly used in arithmetic and encryption circuits.

Truth Table:

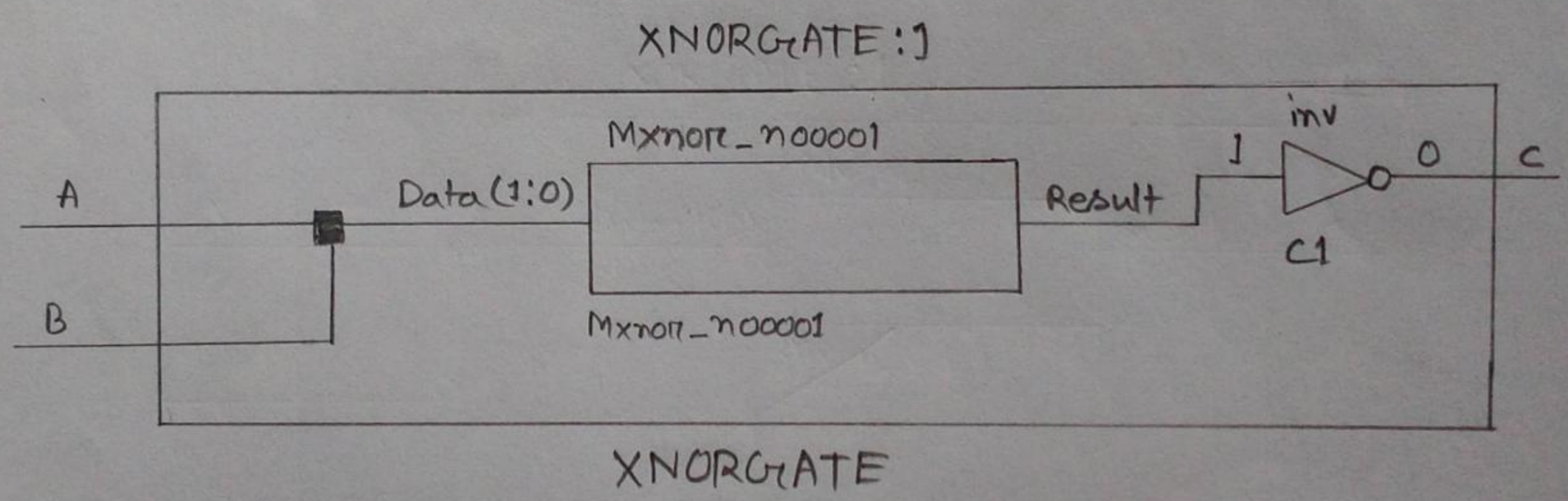
Input		Output
A	B	A XNOR B
0	0	1
0	1	0
1	0	0
1	1	1

Scanned by TapScanner

VHDL Code:

```
entity XNORGATE is
    port ( A: in STD_LOGIC;
           B: in STD_LOGIC;
           C: out STD_LOGIC);

end XNORGATE;

architecture Behavioral of XNORGATE is
begin
C<= B XNOR A;
end Behavioral;
```

Output:



24/08/2022

Q8

Aim: Write a VHDL Code to implement the AND Gate through Data Flow model.

Description:

The AND Gate is basic logic gate that implements logical conjunction from mathematical logic. A high output (1) results only if all the inputs to the AND gate are high, if not then low output occurs.

uses:

It is used in digital electronics and in data transmission. Where AND gate helps for allowing or not allowing the data through a channel. It is also used to inhibit gate, this is the reverse approch of enabling gate.

Truth Table:

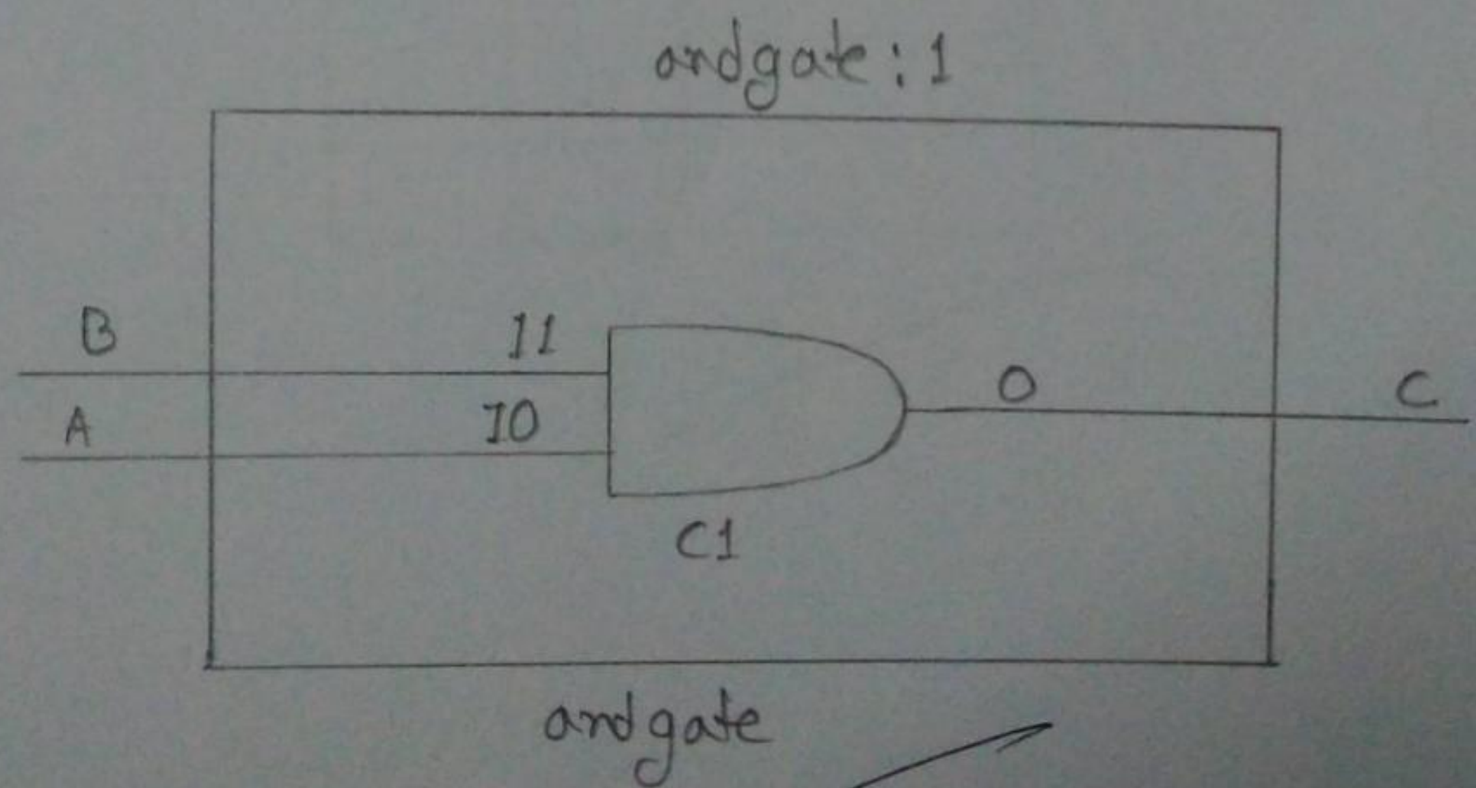
Input		Output
A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

Scanned by TapScanner

VHDL Code:

```
entity andgate is
    Port (A: in STD_LOGIC;
          B: in STD_LOGIC;
          C: out STD_LOGIC);

end andgate;

architecture Behavioral of andgate is
begin
process (A,B)
begin
if((A='1') and (B='1')) then
 c <= '1';
else
c <= '0';
end if;
end process;
end Behavioral;
```

Output:



24/08/2022

Scanned by TapScanner

Aim: Write a VHDL Code to implement OR Gate through dataflow model.

Description:

The OR gate is a digital logic gate that implements logical disjunction from mathematical logic. A High output (1) results if one or both the inputs to the gate are High (1), If neither input is high a low output (0) results.

Uses: OR gates are used in automatic digital circuits. There are many sensors are used to defect faults in an industry. The sensor creates a signal if any fault occurs. Ther OR Gates give command signals to the main processor for the shutdown. OR gates are used in DOBH Bell switches also.

Truth Table:

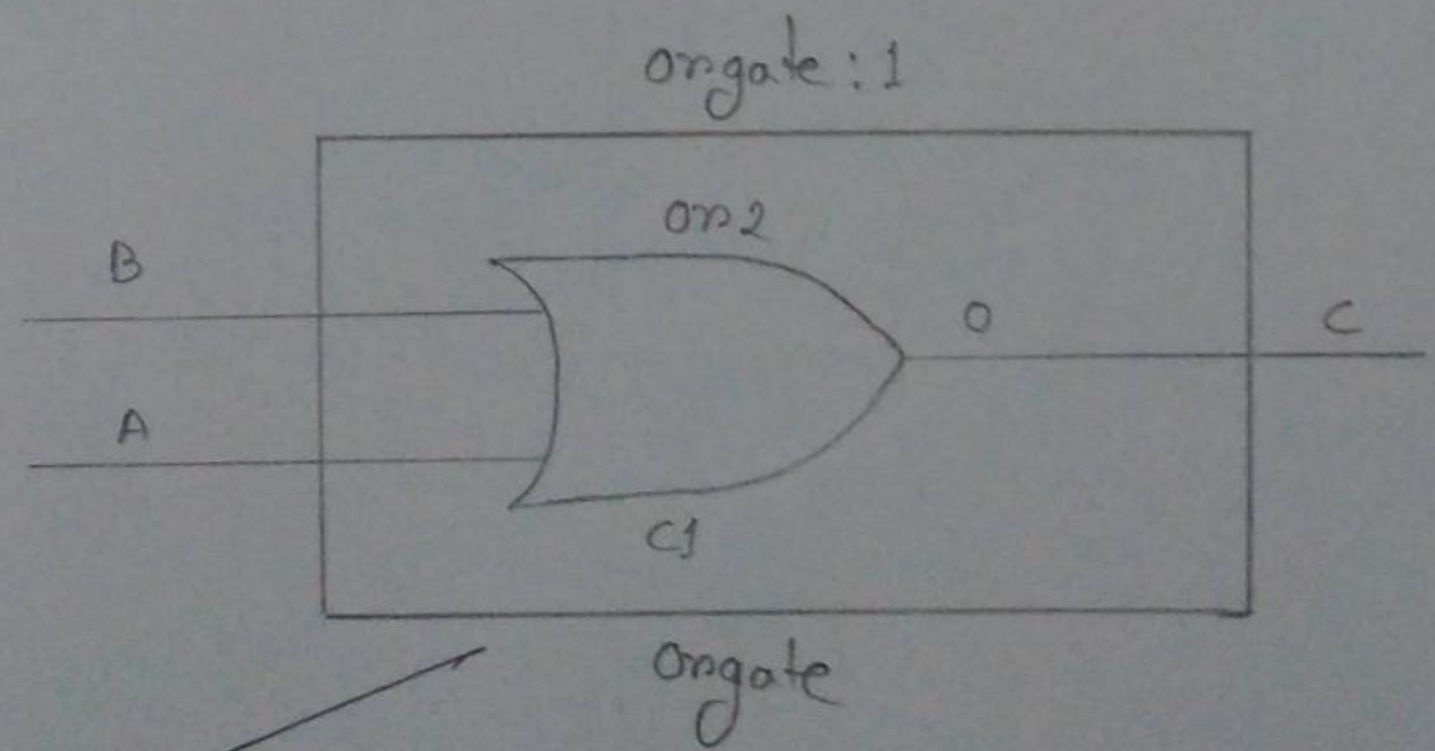
Input		Output
A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

Scanned by TapScanner

VHDL Code:

```
entity orgate is
   Port (A: in STD_LOGIC;
         B: in STD_LOGIC;
         C: out STD_LOGIC);
end orgate;

architecture Behavioral of orgate is
begin
process (A,B)
begin
if ((A='1') or (B='1')) then
 C<='1';
else
 C<='0';
end if;
end process;
end Behavioral;
```

Output:



24/08/2022

Aim: Write a VHDL Code to implement NOT gate through data flow model.

Description:

A NOT gate is a logic gate. Each NOT gate has only one input signal. Logically with NOT gates, the input and the output swap. It will always output the opposite signal. Its main function is to interchange logic.

Uses:

The foremost not gate application is in the CMOS inverter for the generation of waveforms. It is also used in daily usage devices like desktops, tablets and mobiles. Complex devices such as decoders, state machines, and multiplexers also use inverter circuits.

Truth Table:

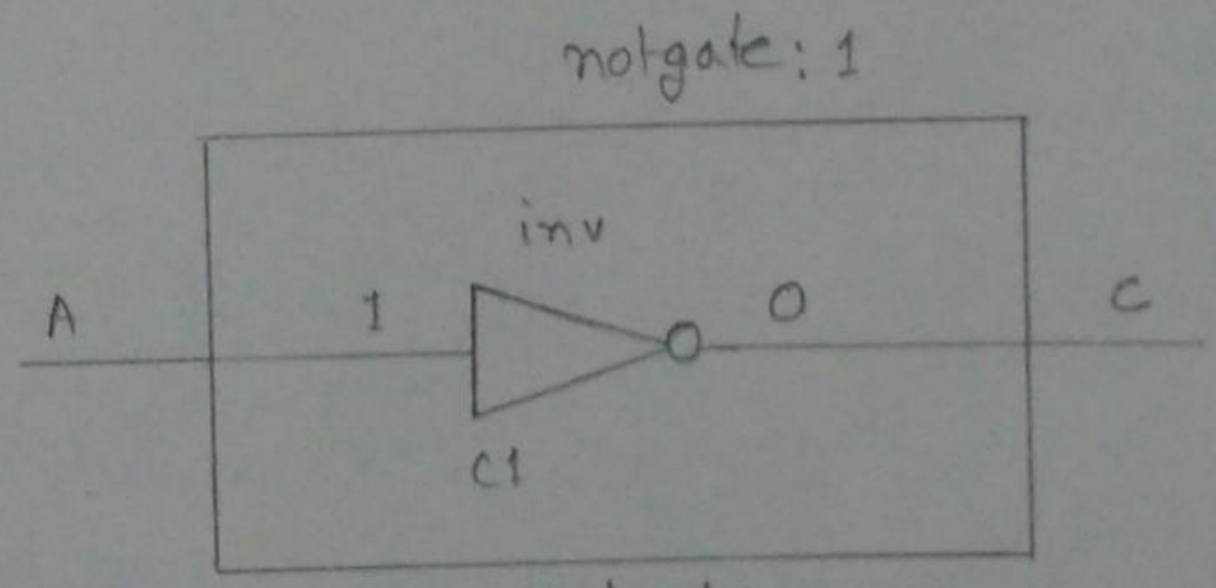
Input	Output
A	NOT A
0	1
1	0

Scanned by TapScanner

VHDL Code:

```
entity notgate is
   Port (A: in STD_LOGIC;
         C: out STD_LOGIC);
end notgate;

architecture Behavioral of notgate is
begin
process(A)
begin
if (A='1') then
C<='0';
else
C<='1';
end if;
end process;
end Behavioral;
```

Output:



notgate

24/08/2022

Scanned by TapScanner

Write a VHDL code to implement Half-Adder.

Description:

A half adder is a type of adder, an electronic circuit that performs the addition of numbers. The half adder is able to add two single binary digits and provide the output plus a carry value. It has two inputs, called A and B, and two outputs S (sum) and C (carry)

Uses:

It is used in calculators, computers, digital measuring devices etc.

Truth Table:

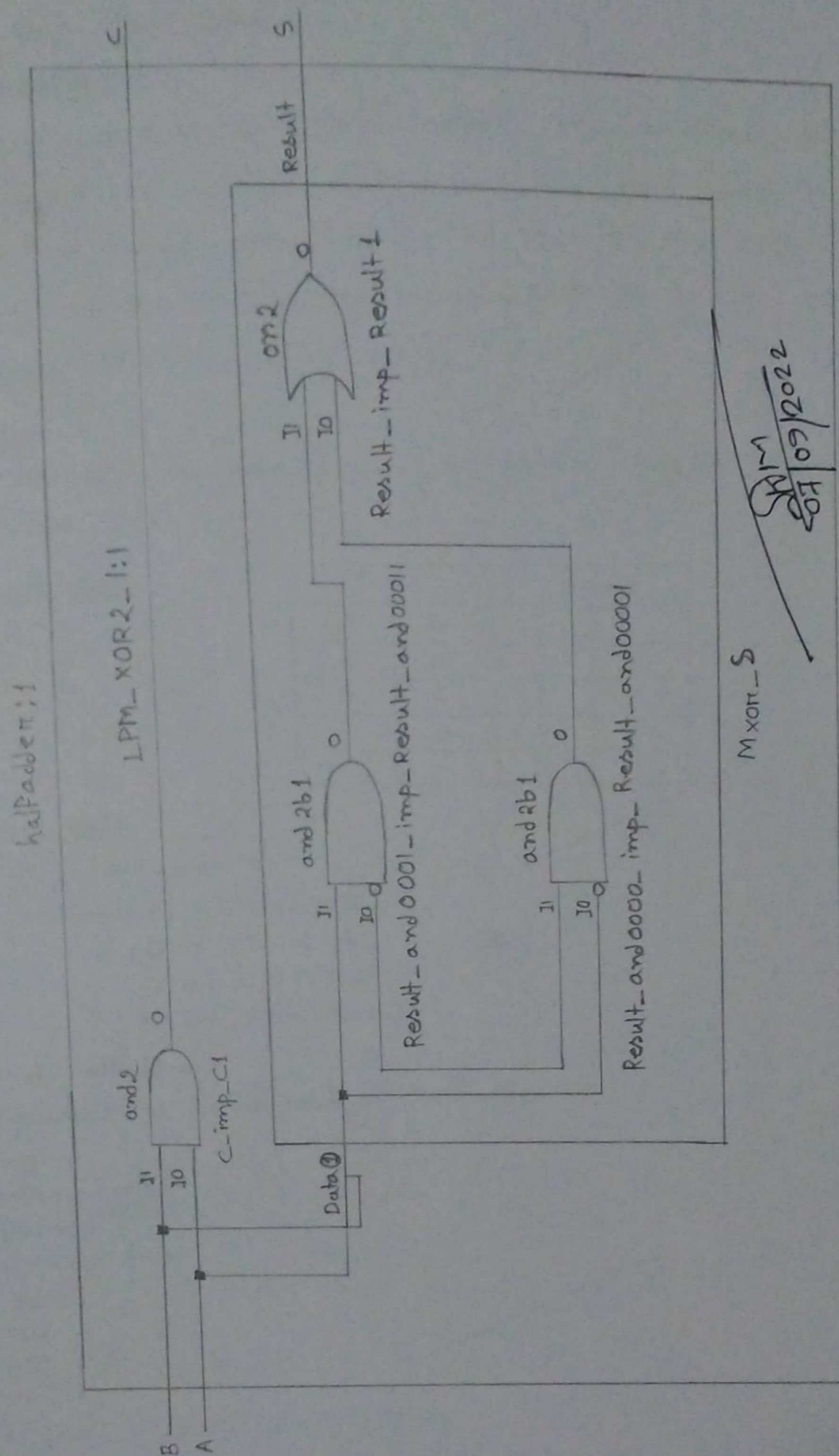
Input		Output	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

VHDL Code:

```
entity halfadder is
    port (A: in STD_LOGIC;
          B: in STD_LOGIC;
          S: out STD_LOGIC;
          C: out STD_LOGIC);

end halfadder;
architecture Behavioral of halfadder is
begin
S<= A XOR B;
C<= A AND B;
end Behavioral;
```

Scanned by TapScanner

output:



Scanned by TapScanner

Aim: Write a VHDL code to implement Half-Adder with the help of data flow model.

Discription:

A half adder is a type of adder, an electronic circuit that performs the addition of numbers. The half adder is able to add two single binary digits and provide the output plus a carry value. It has two inputs, called A and B, and two outputs S(sum) and C(carry).

Uses:

It is used in calculators, computers, digital measuring devices etc.

Truth Table:

Input		Output	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

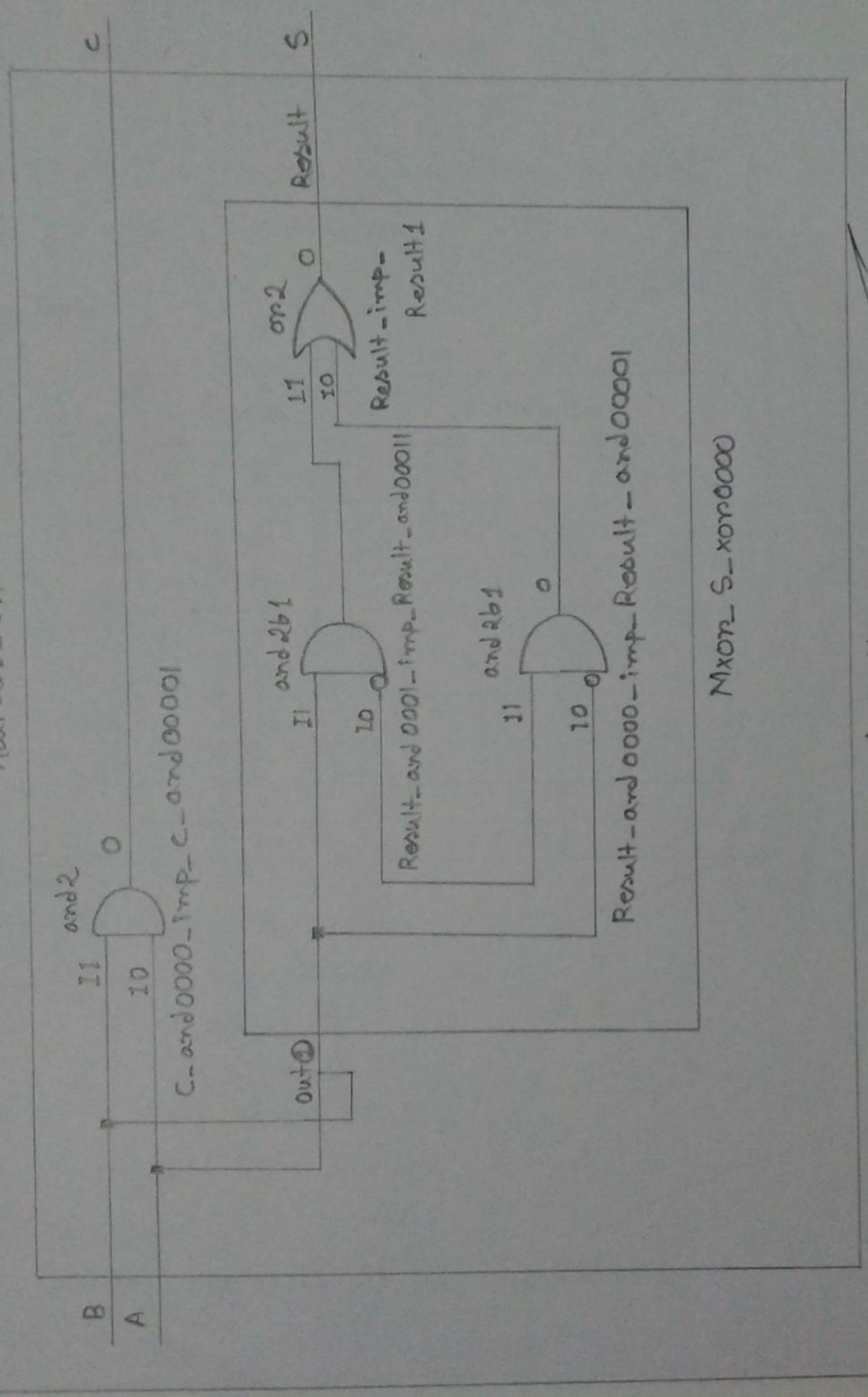
VHDL Code:

```
entity halfadder is
    Port (A: in STD_LOGIC;
          B: in STD_LOGIC;
          S: out STD_LOGIC;
          C: out STD_LOGIC);

end halfadder;
Architecture Behavioral of halfadder is

begin
process (A,B)
begin
if(A /=B) then
 s<='1';
else
s<='0';
end if;
if((A='1' AND (B='1')) then
c<='1';
```

Scanned by TapScanner

```
else
c<='0';
end if;
end process;
end Behavioral;
```

<u>output:</u>



halfadder

07/09/2022

Scanned by TapScanner

3 Write a VHDL code to implement Full-adder through data flow model.

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 as Cin. The output carry is recognised as C-out and the normal output is known as S which is sum.

Uses:

Full adders are mainly used for multiple bit addition, in digital processing devices etc.

Truth Table:

Inputs			Outputs	
A	B	Cin	S	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

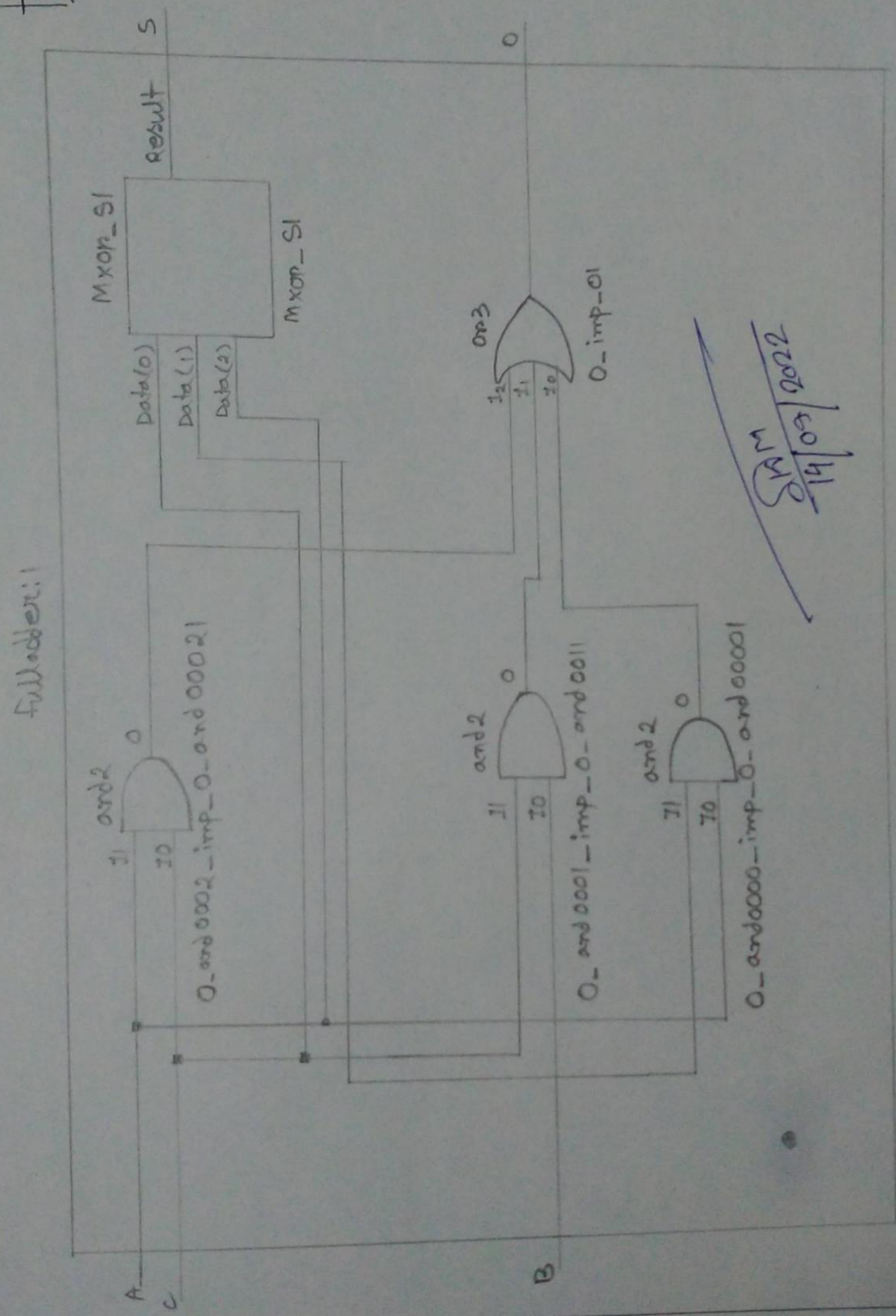
VHDL Code:

```
entity fulladder is
    port ( A: in STD_LOGIC;
           B: in STD_LOGIC;
           C: in STD_LOGIC;
           S: out STD_LOGIC;
           D: out STD_LOGIC);
end fulladder;
architecture Behavioral of fulladder is
begin
```

Scanned by TapScanner

```
S<= A XOR B XOR C;
D<= ((A AND B) OR (B AND C) OR (C AND A));
end Behavioral;
```

<u>Output:</u>



14/09/2022

14 Write a VHDL code to implement 8:1 Multiplexer.

Description:

A multiplexer is a combinational circuit. In this circuit, depending on the select lines one of the input line will be selected as a output line. The relationship between input line and select line is 2^n, where n is the number of select line.

Uses:

Multiplexers are part of computer systems to select data from a specific source, be it a memory chip on a hardware peripheral. A computer uses multiplexers to control the data and address buses, allowing the processor to select data from multiple data sources.

Truth Table:

Input			output
S_0	S_1	S_2	O
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7

VHDL Code:

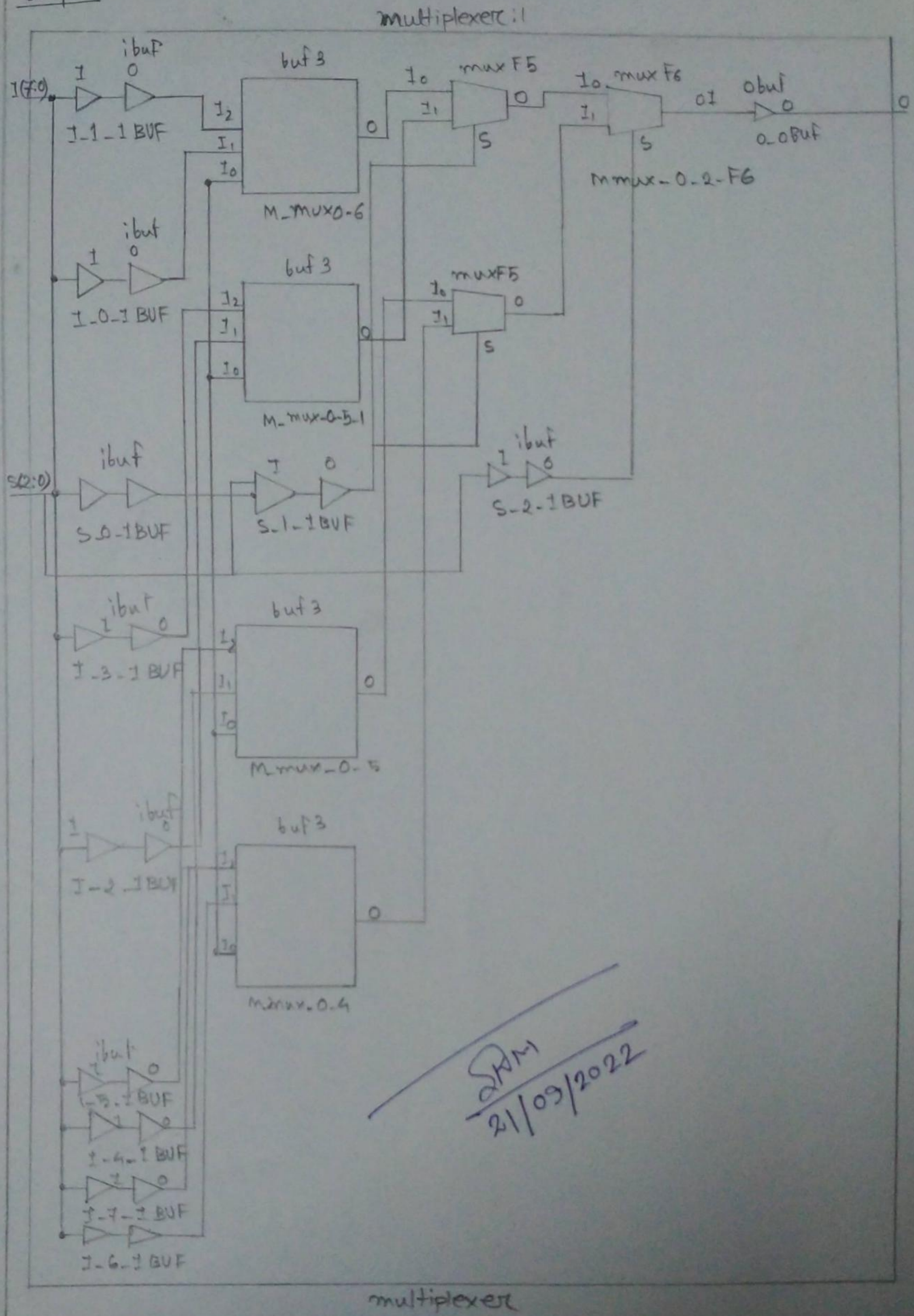
```
entity multiplexer is
    port (S: in STD_LOGIC_VECTOR(2 downto 0);
          I: in STD_LOGIC_VECTOR(7 downto 0);
          O: out STD_LOGIC);
```

Scanned by TapScanner

```
end multiplexer;
architecture Behavioral of multiplexer is
begin
process (I,S)
begin
if (S="000") then O<=I(0);
end if;
if(S="001") then O<=I(1);
end if;
if (S="010") then O<=I(2);
end if;
if(S="011") then O<=I(3);
end if;
if (S="100") then O<=I(4);
end if;
if (S="101") then O<=I(5);
end if;
if( S="110") then O<=I(6);
end if;
if(S="111") then O<=I(7);
end if;
end process;
end Behavioral;
```

Scanned by TapScanner

Output:



21/09/2022

15 Write a VHDL code to implement 3:8 Decoder.

Description:

Decoder is a combinational circuit in which depending on the input lines one of the output line will be selected. The relationship between the input line & output line is 2^n, where n is the number of input line. These type of circuit will be activated on the control signal's reference.

Uses:

Decoders are used to input data to a specified output line as is done in addressing core memory where input data is to be stored in a specified memory location. It is also used in code conversions.

Truth Table:

A_0	A_1	A_2	O
0	0	0	O_0
0	0	1	O_1
0	1	0	O_2
0	1	1	O_3
1	0	0	O_4
1	0	1	O_5
1	1	0	O_6
1	1	1	O_7

VHDL Code:

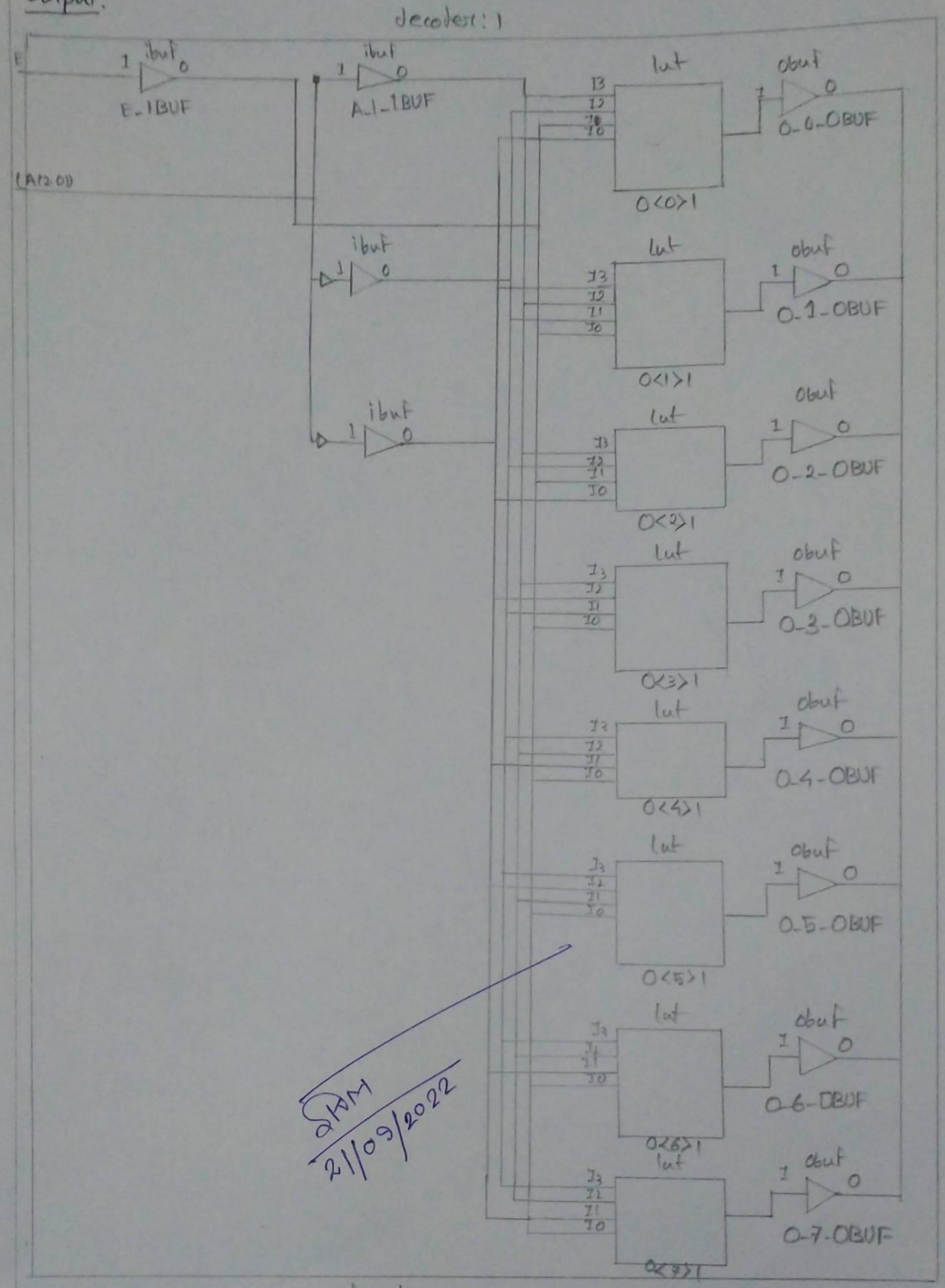
```
entity decoder is
    port (A: in STD_LOGIC_VECTOR(2 downto 0);
          E: in STD_LOGIC;
          O: out STD_LOGIC_VECTOR(7 downto 0));

end decoder;

architecture Behavioral of decoder is

begin
process (A,E)
```

Scanned by TapScanner

```
begin
if(E='0')then
if (A="000") then O<="00000001";
elsif (A="001") then O<="00000010";
elsif (A="010") then O<="00000100";
elsif (A="011") then O<="00001000";
elsif (A="100") then O<="00010000";
elsif (A="101") then O<="00100000";
elsif (A="110") then O<="01000000";
else
O<="10000000";
end if;
else
O<="00000000";
end if;
end process;
end Behavioral;
```

Scanned by TapScanner

Output:



SAM
21/09/2022

Scanned by TapScanner

16 Write a VHDL Code to implement Carry Look Ahead Adder.

Description:

The adder produce carry propagation delay while performing other arithmetic operations like multiplication and divisions as it uses several additions or subtraction steps. One widely used approach is to employ a carry look ahead which solves the problem by calculating the carry signals in advance, based on the input signals. This type of adder circuit is called a carry look-ahead adder.

Truth Table:

A	B	C	C+1	Condition
0	0	0	0	No Carry Generate
0	0	1	0	
0	1	0	0	
0	1	1	1	No Carry Propagate
1	0	0	0	
1	0	1	1	
1	1	0	1	carry Generate
1	1	1	1	

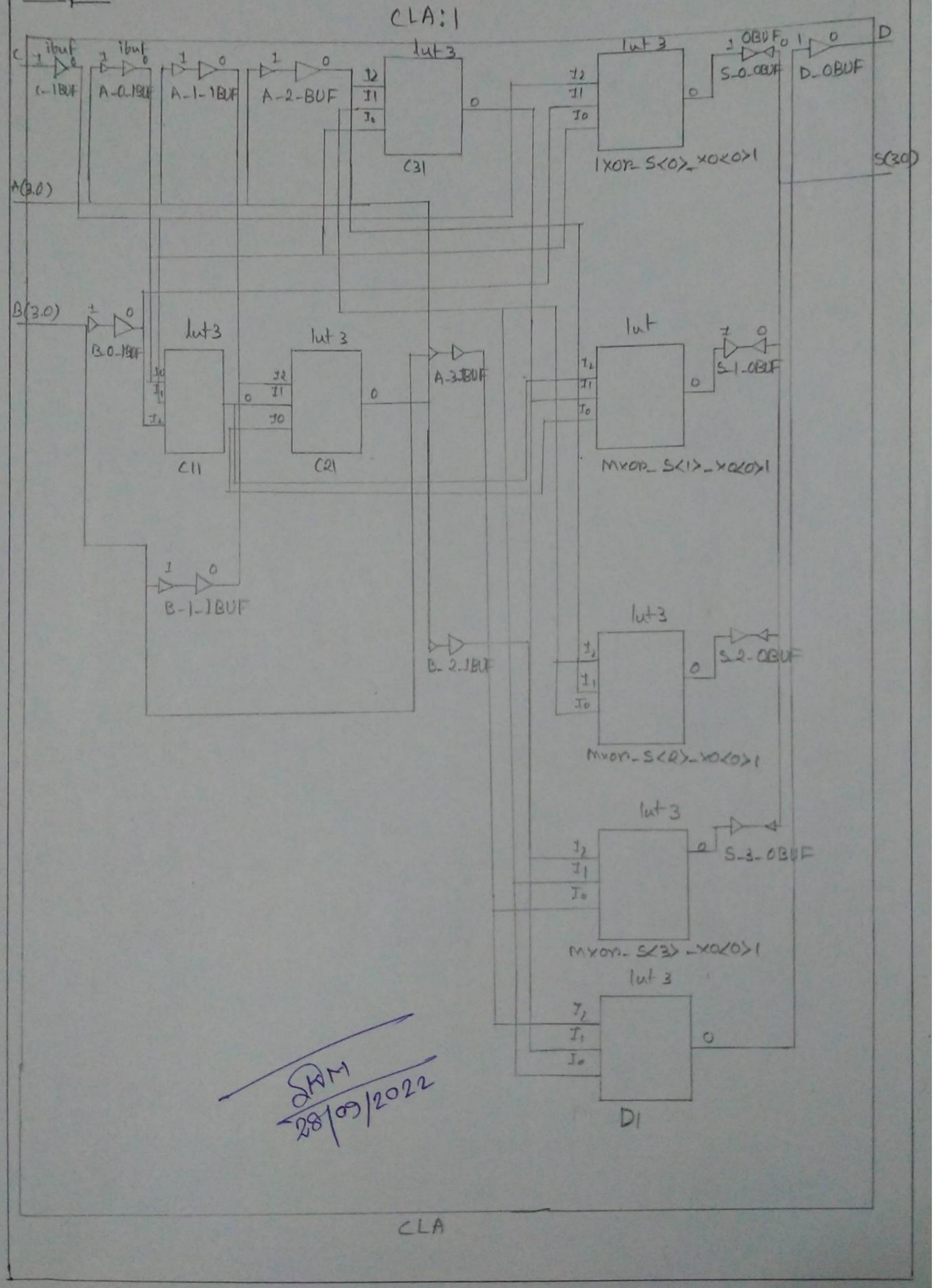
Scanned by TapScanner

VHDL Code:

```
entity CLA is
   Port (A: in STD_LOGIC_VECTOR (3 downto 0);
         B: in STD_LOGIC_VECTOR (3 downto 0);
         C: in STD_LOGIC;
         D: out STD_LOGIC;
         S: out STD_LOGIC_VECTOR (3 downto 0));
end CLA;

architecture Behavioral of CLA is
signal C1, C2, C3, P0, P1, P2, P3, G0, G1, G2, G3: STD_LOGIC;
begin
S(0) <= A(0) XOR B(0) XOR C;
S(1) <= A(1) XOR B(1) XOR C1;
S(2) <= A(2) XOR B(2) XOR C2;
S(3) <= A(3) XOR B(3) XOR C3;
G0 <= A(0) AND B(0);
G1 <= A(1) AND (B(1);
G2 <= A(2) AND B(2);
G3 <= A(3) AND B(3);
P0 <= A(0) OR B(0);
P1 <= A(1) OR B(1);
P2 <= A(2) OR B(2);
P3 = A(3) OR B(3);
C1 <= G0 OR (P0 AND C);
C2 <= G1 OR (P1 AND C1);
C3 <= G2 OR (P2 AND C2);
D <= G3 OR (P3 AND C3);
end Behavioral;
```

Scanned by TapScanner

Output:



SKM
28/09/2022

Scanned by TapScanner

17 Write a VHDL code to implement De-Multiplexer.

Description:

Demultiplexer is a type of electronic circuit which receives a single input signal and selects one of multiple possible output routes to which to transmit the signal. The relation between outputput lines and select lines ~~are~~ is 2^n

Uses:

Demultiplexer is used to connect a single source to multiple destinations. The main application area of demultiplexer is communication system, where multiplexers are used.

Truth Table:

Input			Output
S_0	S_1	S_2	I
0	0	0	O_0
0	0	1	O_1
0	1	0	O_2
0	1	1	O_3
1	0	0	O_4
1	0	1	O_5
1	1	0	O_6
1	1	1	O_7

VHDL code:

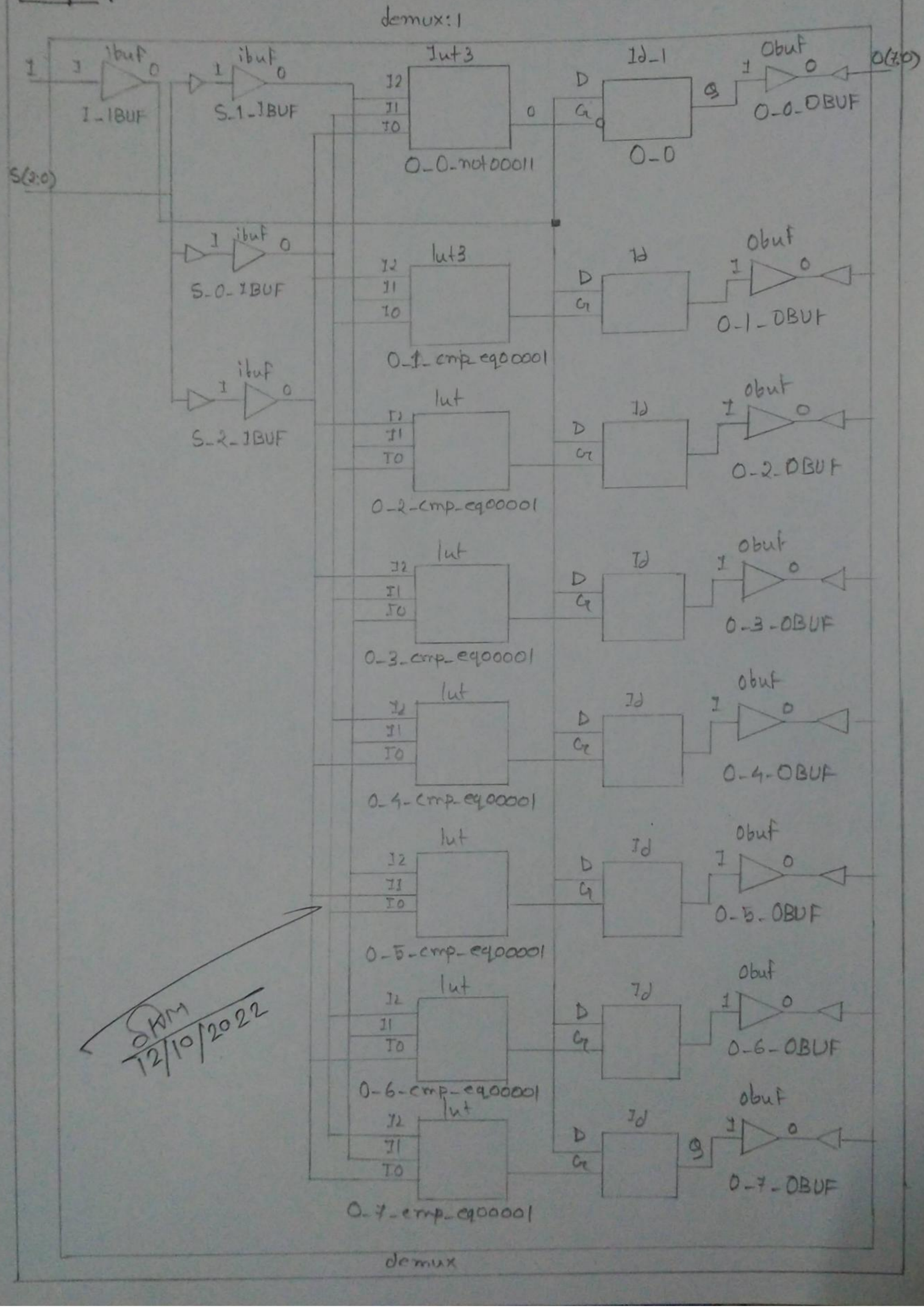
```
entity demux is
    Port (S: in STD_LOGIC_VECTOR(2 downto 0);
          I: in STD_LOGIC;
          O: out STD_LOGIC_VECTOR(7 downto 0));

end demux;
architecture Behavioral of demux is
```

Scanned by TapScanner

```
begin
process(S,I)
begin
if(S="000") then O(0)<=I;
end if;
if(S="001") then O(1)<=I;
end if;
if(S="010") then O(2)<=I;
end if;
if(S="011") then O(3)<=I;
end if;
if(S="100") then O(4)<=I;
end if;
if(S="101") then O(5)<=I;
end if;
if(S="110") then O(6)<=I;
end if;
if(S="111") then O(7)<=I;
end if;
end process;
end Behavioral;
```

Scanned by TapScanner

Output:



SKM
12/10/2022

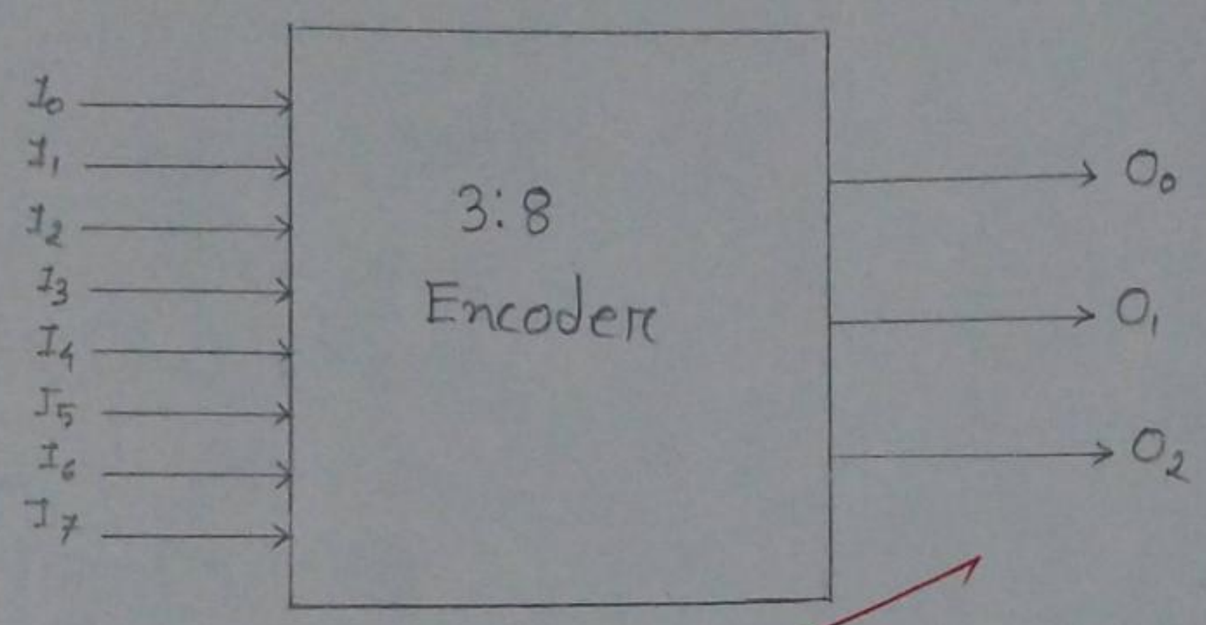
Scanned by TapScanner

18 Write a VHDL code to implement Encoder.

Description:

An encoder is a combinational circuit that performs the reverse operation of decoder. It has maximum 2^n input lines and 'n' output lines, hence it encodes the information form 2^n inputs into an n-bit code. It will produce a binary code equivalent to the input, which is active high.

Block Diagram:



Truth Table:

I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	O_0	O_1	O_2
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

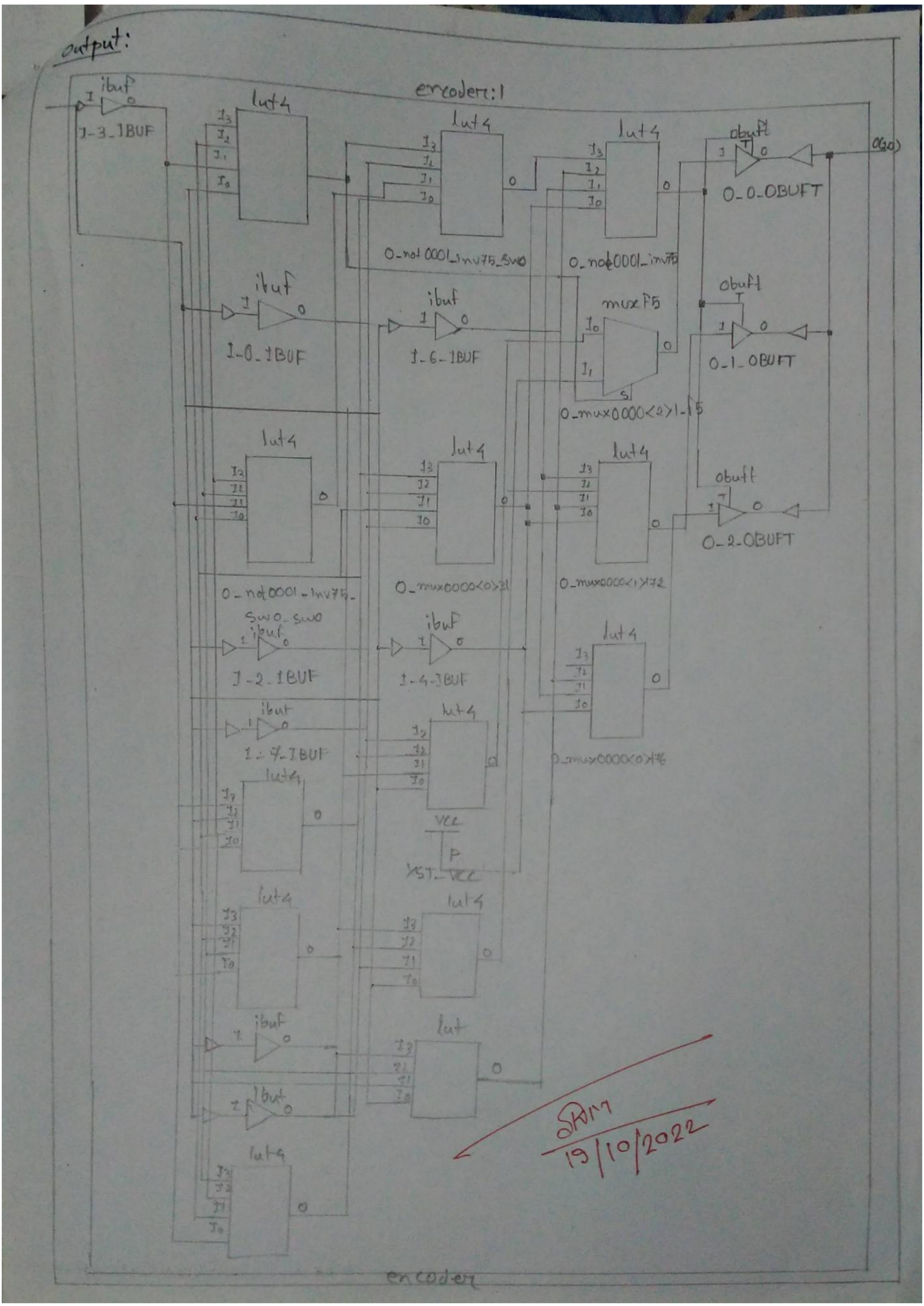
Scanned by TapScanner

VHDL Code:

```
entity encoder is
    Port (I: in STD_LOGIC_VECTOR(7 downto 0);
          O: out STD_LOGIC_VECTOR (2 downto 0));
end encoder;
architecture Behavioral of encoder is
begin
process (I)
begin
case I is
when "10000000" => O<= "000";
when "01000000" => O<= "001";
when "00100000" => O<= "010";
when "00010000" => O<= "011";
when "00001000" => O<= "100";
when "00000100" => O<= "101";
when "00000010" => O<= "110";
when "00000001" => O<= "111";
when others => O<= "ZZZ";
end case;
end process;
end Behavioral;
```

Scanned by TapScanner

output:



Scanned by TapScanner

Aim: Write a VHDL code to implement the NAND gate through data flow model.

Description: NAND Gate is a logical gate which is opposite of AND Gate. It is a combination of AND and NOT gates and is commonly used logic gate considered as a 'Universal Gate'.

Uses: It is widely used in application like burgler alarms, freezer warming burglers and automated circuits.

Truth Table:

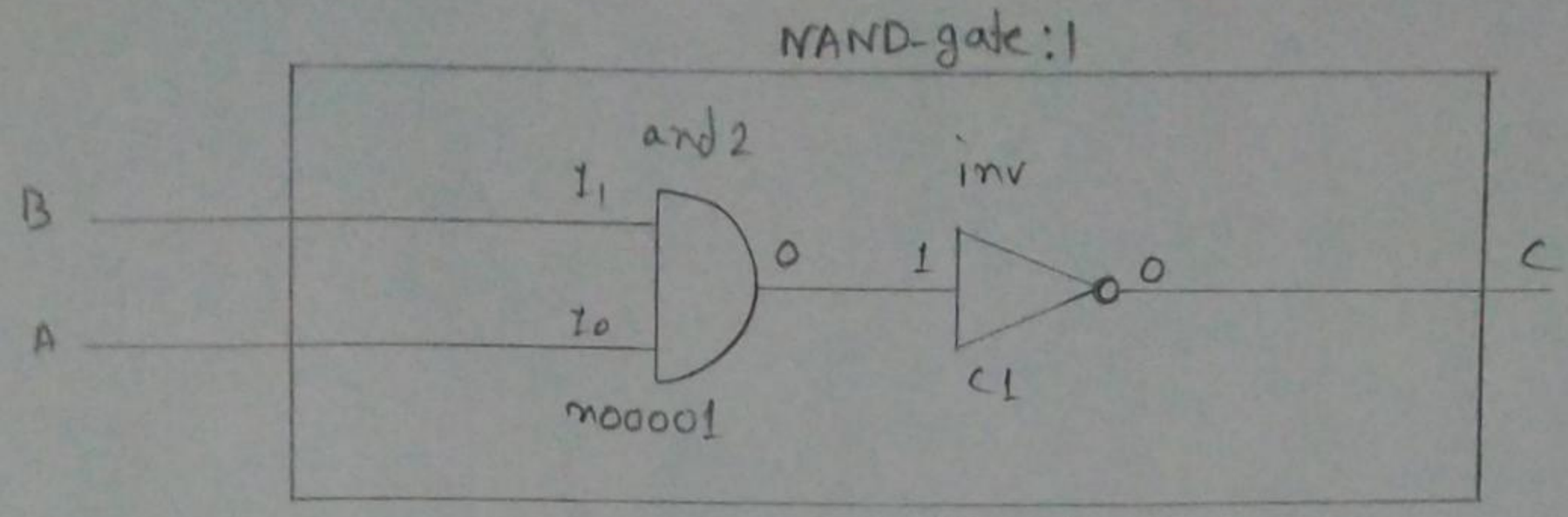
Input		Output
A	B	$C = \overline{AB}$
0	0	1
0	1	1
1	0	1
1	1	0

VHDL Code:

```
entity NAND-gate is
        port (A: in STD_LOGIC);
              B: in STD_LOGIC;
              C: out STD_LOGIC);

end NAND-gate;
architecture Behavioral of NAND-gate is
begin
process(A,B)
begin
if ((A='1') AND (B='1')) then
   C<='0';
else
   C<='1';
end if;
end process;
end behavioral;
```

Scanned by TapScanner

Output:



NAND-gate

SKM
02/11/2022

Scanned by TapScanner

Aim: Write a VHDL code to implement the NOR Gate through Data Flow model.

Description: NOR gate is a type of logic gate that works on the principle of "Neither this nor that". This type of digital logic gate produces a high output only if the binary results are satisfied by the zero or low input.

Uses: It performs as a complimentary of OR gate. They are used in combinational circuits such as multiplexers, multipliers, half and full adders. and in ripple-carry adders. Also used in sequential circuits and in shift registers also.

Truth Table:

Input		Output
A	B	$C = \overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

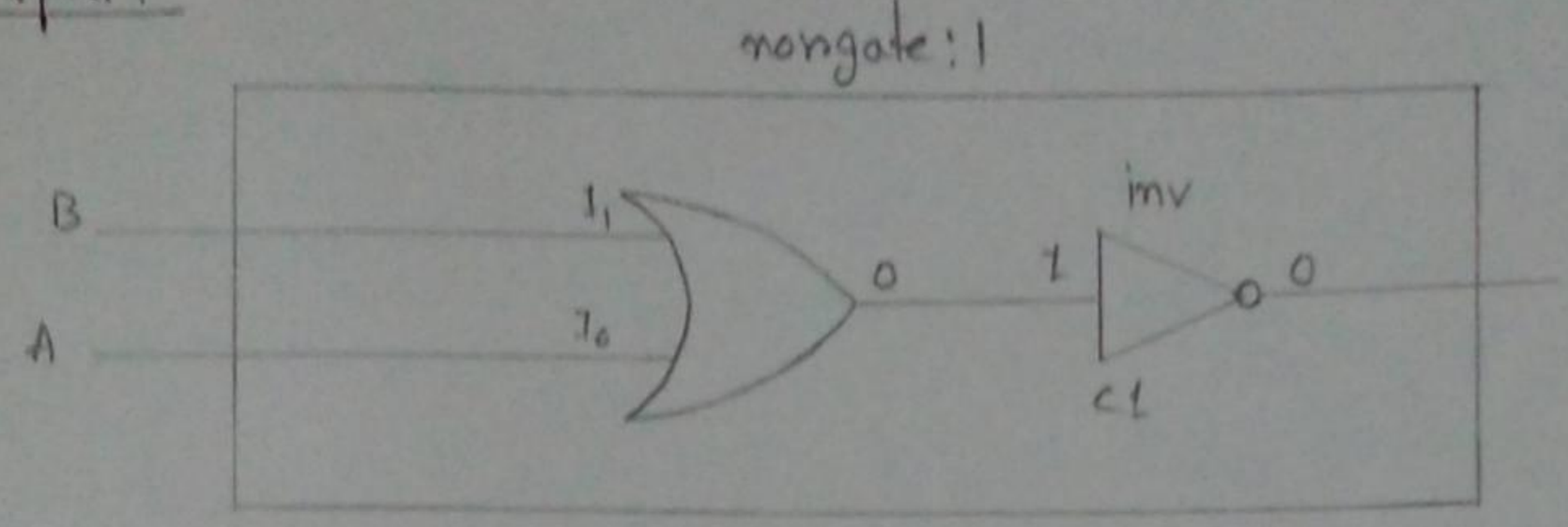
VHDL code:

```
Entity norgate is
        port (A: in STD_LOGIC;
              B: in STD_LOGIC;
              C: out STD_LOGIC);

end norgate ;
architecture Behavioral of norgate is

begin
process (A,B)
begin
if ((A='0') nor (B='0')) then
C<='1';
else
C<='0';
```

Scanned by TapScanner

```
end if;
end process;
end Behavioral;
```

Output:



nongate

ঐম
02/11/2022

Aim: Write a VHDL code to implement the XOR gate through data flow model.

Description: XOR is the abbreviation for "exclusively-OR". The simplest XOR gate is a two-input digital circuit that outputs a logical '1' if the two input values differ.

It is used to pseudorandom number generator.

Truth Table:

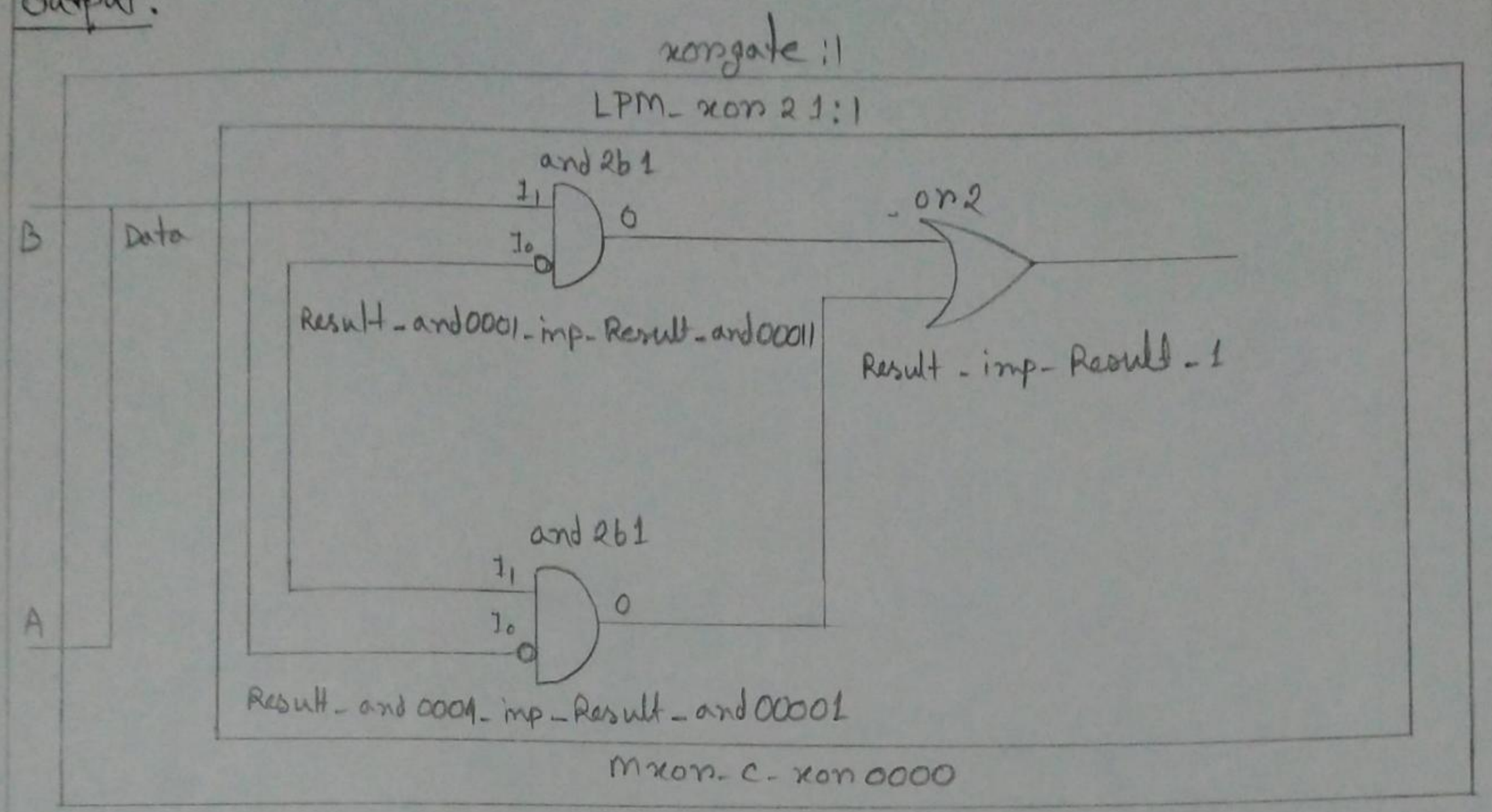
Input		Output
A	B	$C = \bar{A}B + A\bar{B}$
0	0	0
0	1	1
1	0	1
1	1	0

VHDL Code:

```
entity xorgate is
        Port (A: in STD_LOGIC;
              B: in STD_LOGIC;
              C: out STD_LOGIC);

end xorgate ;
architecture Behavioral of xorgate is
 begin
 process(A,B);
 begin
 if (A/=B) then
 C<='1';
 else
 C<='0';
 end if;
 end process;
 end Behavioral;
```

Scanned by TapScanner

Output:



xorgate

SKM
02/11/2022

Scanned by TapScanner

Aim: Write a VHDL code to implement the XNOR gate through data flow model.

Description: The XNOR (exclusively-NOR) gate is a combination of XOR followed by an inverter. Its output is "true" if the inputs are the same and "false" if the inputs are different.

It is used in

i) encryption and arithmetic circuits.

ii) error detecting circuits to find odd parity and even parity in digital data transmission.

Truth Table:

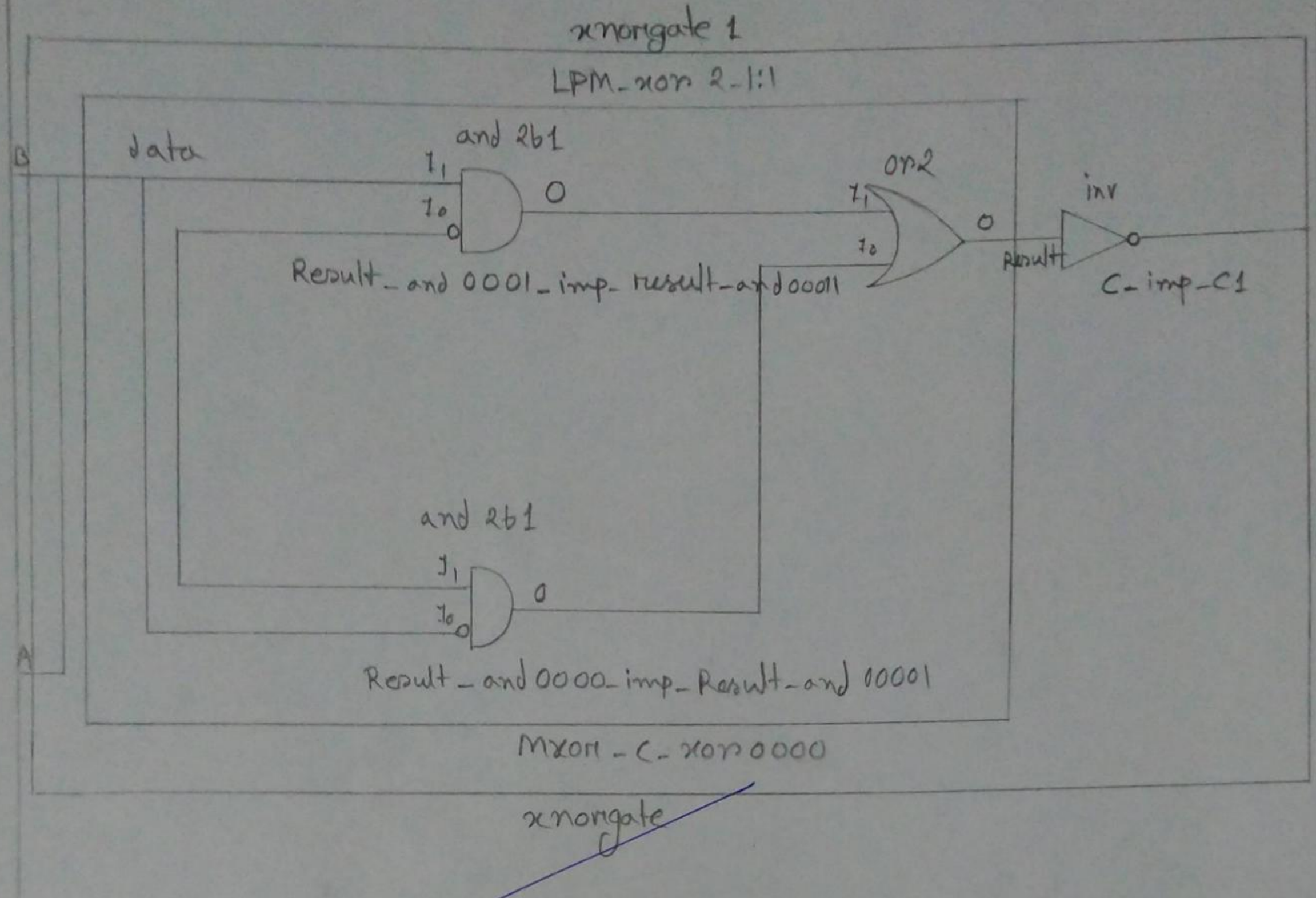
Input		Output
A	B	$C = \overline{A \oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1

VHDL Code:

```
entity xnorgate is
    Port( A: in STD_LOGIC;
          B: in STD_LOGIC;
          C: out STD_LOGIC);

end xnorgate;
architecture Behavioral of xnorgate is
begin
process(A,B)
begin
if (A/=B) then
 C<= '0';
else
 C<= '1';
end if;
end process;
end Behavioral;
```

Scanned by TapScanner

Output:



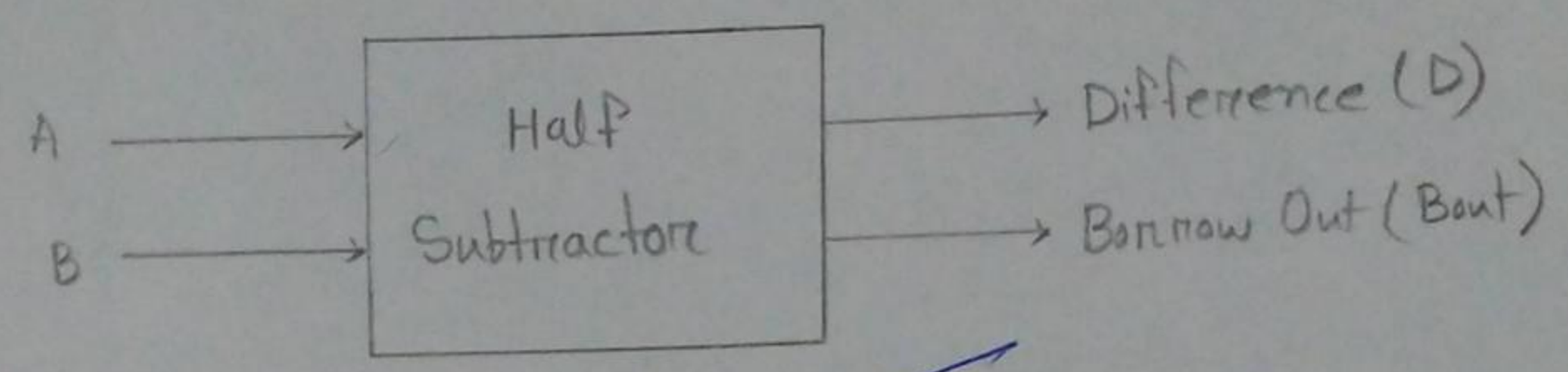
02/11/2022

Scanned by TapScanner

Aim: Write a VHDL code to implement Half Subtractor.

Description: Subtractor is a digital circuit that performs subtraction of numbers. The half subtractor is a combinational circuit which is used to perform subtraction of two bits. It has two inputs, the minuend (A) and subtrahend (B) and two outputs – the difference (D) and borrow out (Bout). An important point worth mentioning is that when the half subtractor generates any borrow out, it is always set to the '1'.

Block Diagram:



Truth Table:

Input		Output	
A	B	D	Bout
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

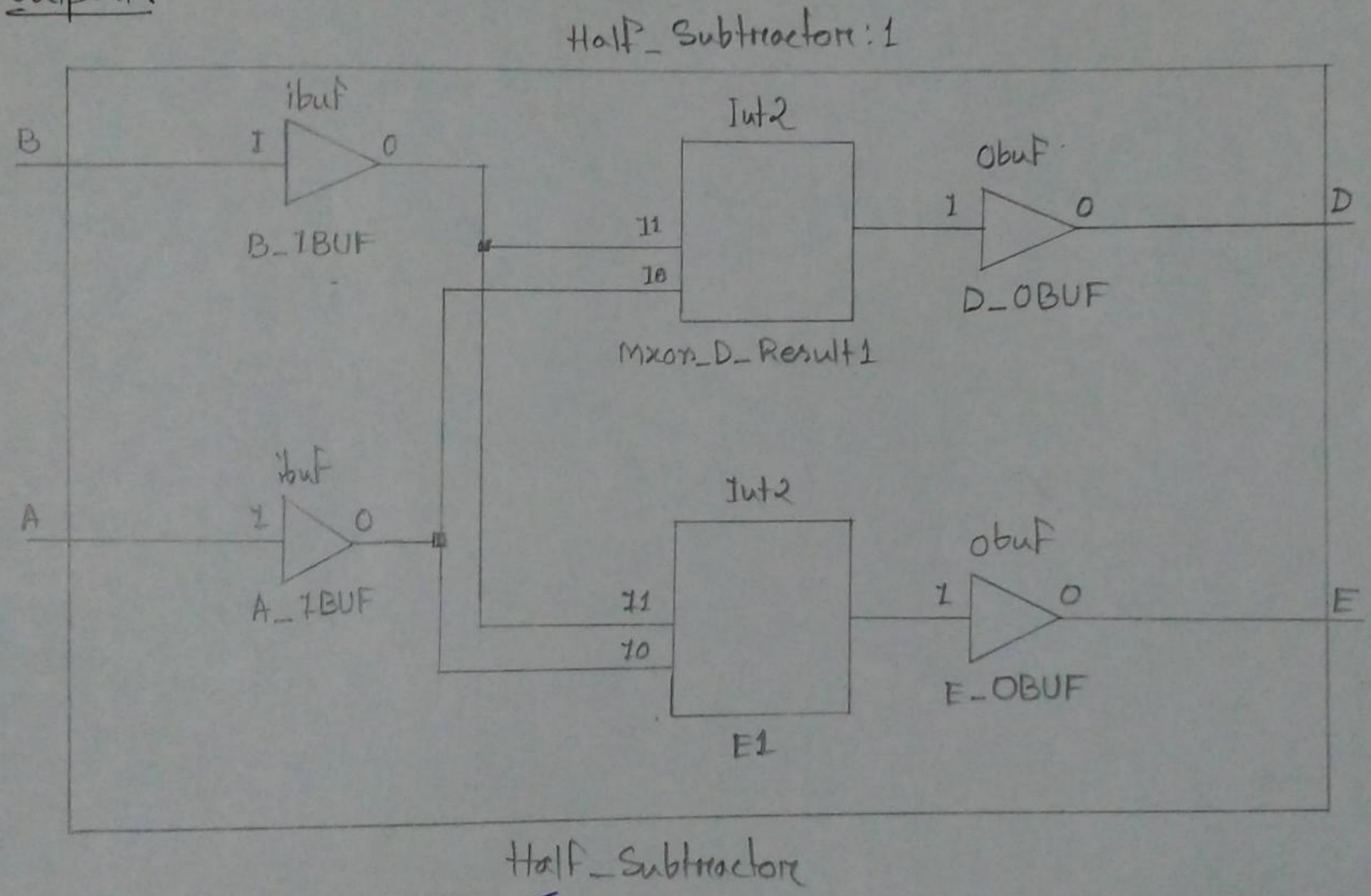
VHDL Code:

```
entity Half_Subtractor is
    Port (A: in STD_LOGIC;
          B: in STD_LOGIC;
          D: out STD_LOGIC;
          E: out STD_LOGIC);
end Half_Subtractor;
```

Scanned by TapScanner

```
architecture Behavioral of Half_Subtractor is

begin
process(A,B)
begin
D<= A XOR B;
E<= (NOT A) AND B;
end process;
end Behavioral;
```

Output:



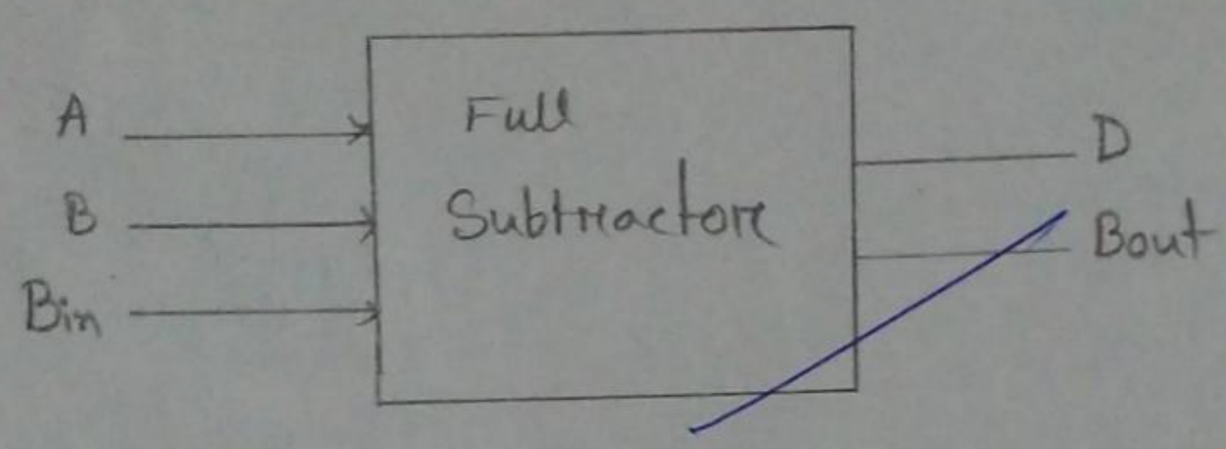
SKM
02/11/2022

Scanned by TapScanner

Aim: Write a VHDL code to implement Full_subtractor.

Description: The full subtractor is a combinational circuit which is used to perform subtraction of three input bits: the minuend (A), subtrahend (B) and Borrow in (Bin). The full subtractor generates two outputs bits: the difference (D) and borrow out (Bout).

Block Diagram:



Truth Table:

Input			Output	
A	B	Bin	D	Bout
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

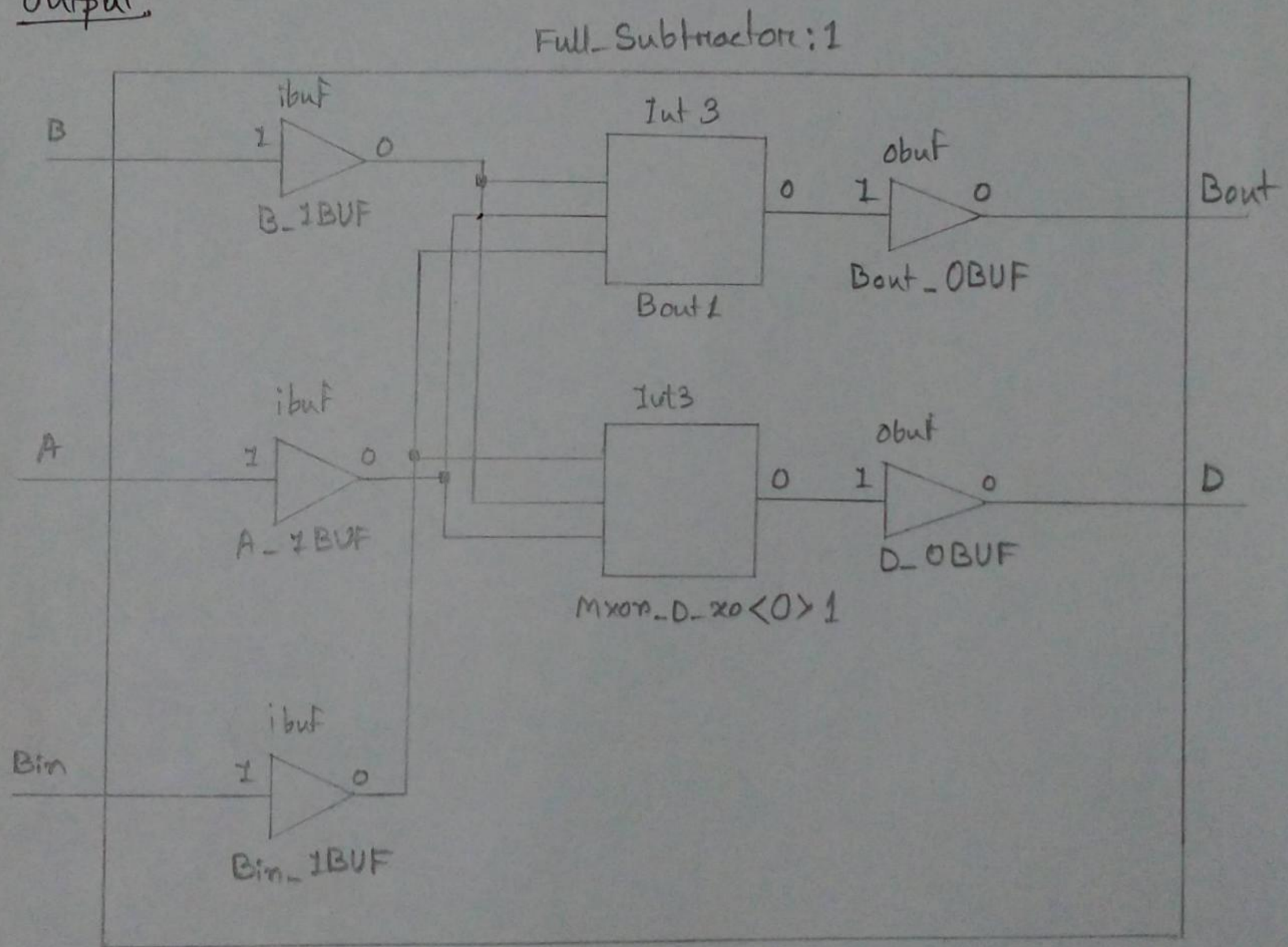
VHDL code:

```
entity Full_Subtractor is
    Port (A: in STD_LOGIC;
          B: in STD_LOGIC;
          Bin: in STD_LOGIC;
          D: out STD_LOGIC;
          Bout: out STD_LOGIC);
end Full_Subtractor;
```

Scanned by TapScanner

```
architecture Behavioral of Full_Subtractor is
begin
process (A, B, Bin)
begin
D <= A XOR B XOR Bin;
Bout <= ((NOT A) AND (B OR Bin)) OR (B AND Bin);
end process;
end Behavioral;
```

Output:



Full_Subtractor

SKM
02/11/2022

Scanned by TapScanner