More Complex Combinational Circuit Design using VHDL Programming

Presented by

Dr. Sudipta Basu Pal

CST & CSIT, UEM,Kolkata

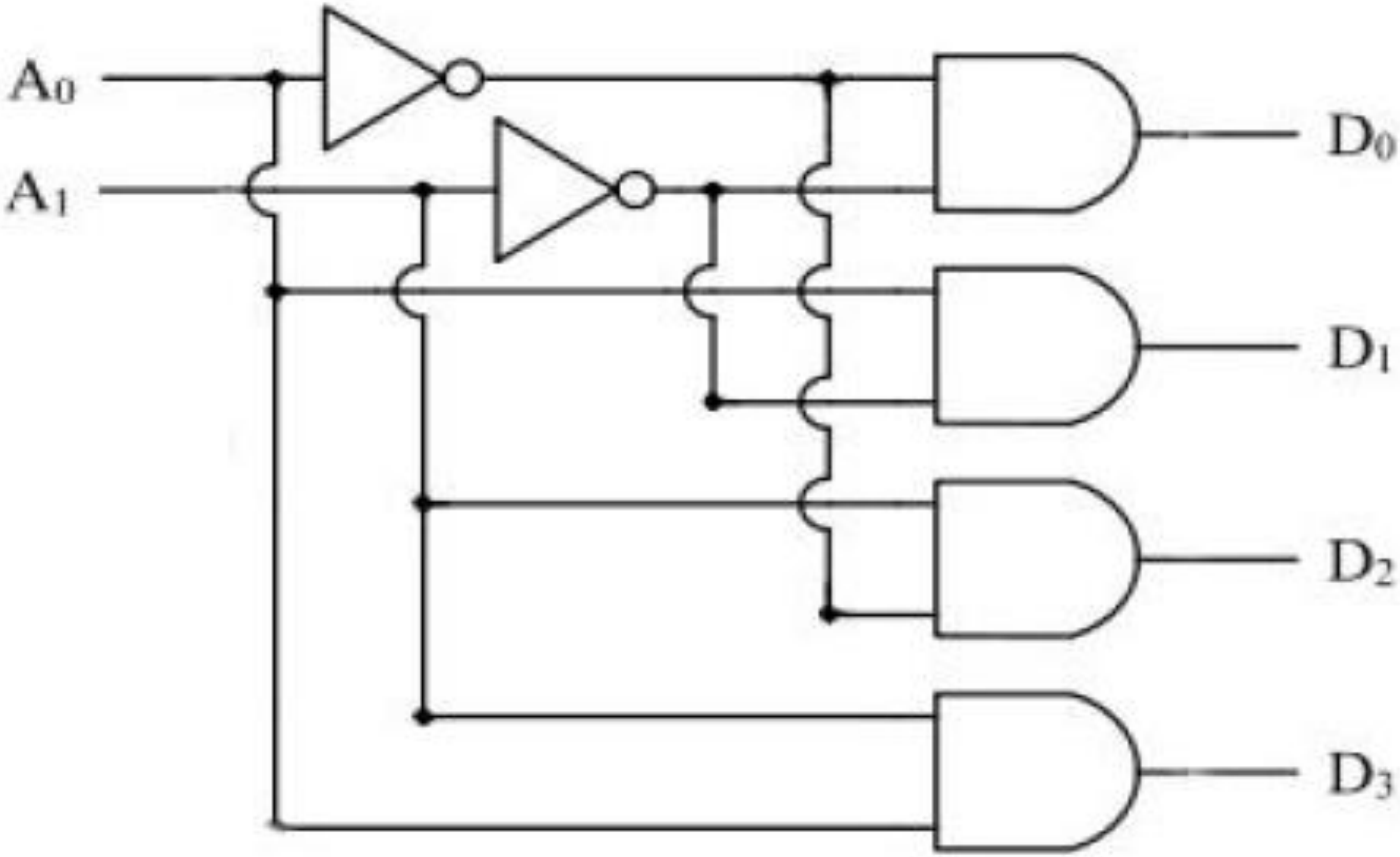
Write a VHDL program to implement 2x4 Decoder using Case statement.

- Title: Implementing 2x4 Decoder using Case statement.

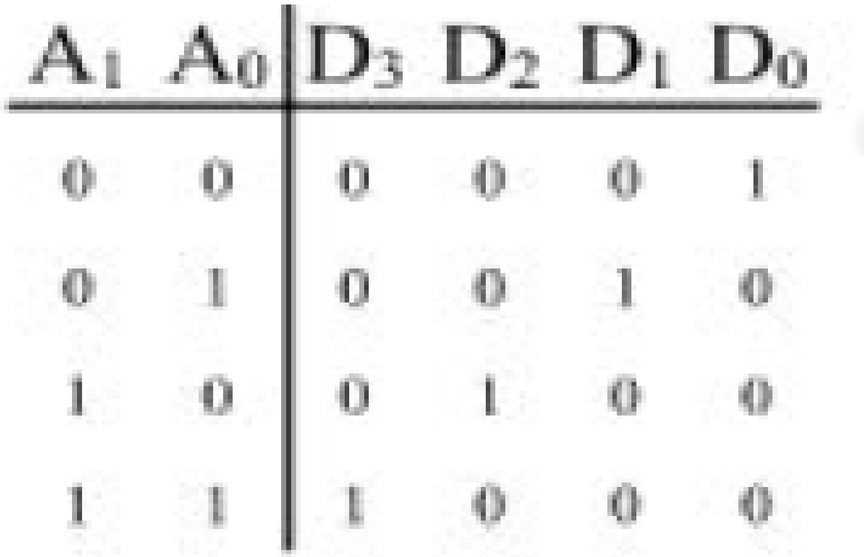
- Expression : D_0 = X'Y' ; D_1 = X'Y ; D_2 = XY' ;

D_3 = XY

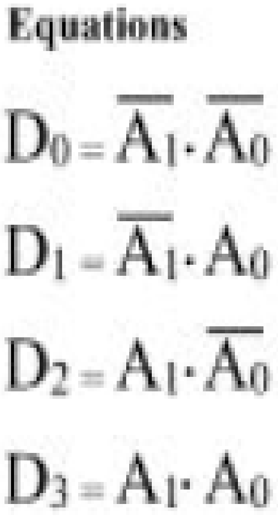
Circuit Diagram :



Truth Table



A_1	A_0	D_3	D_2	D_1	D_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0



Equations

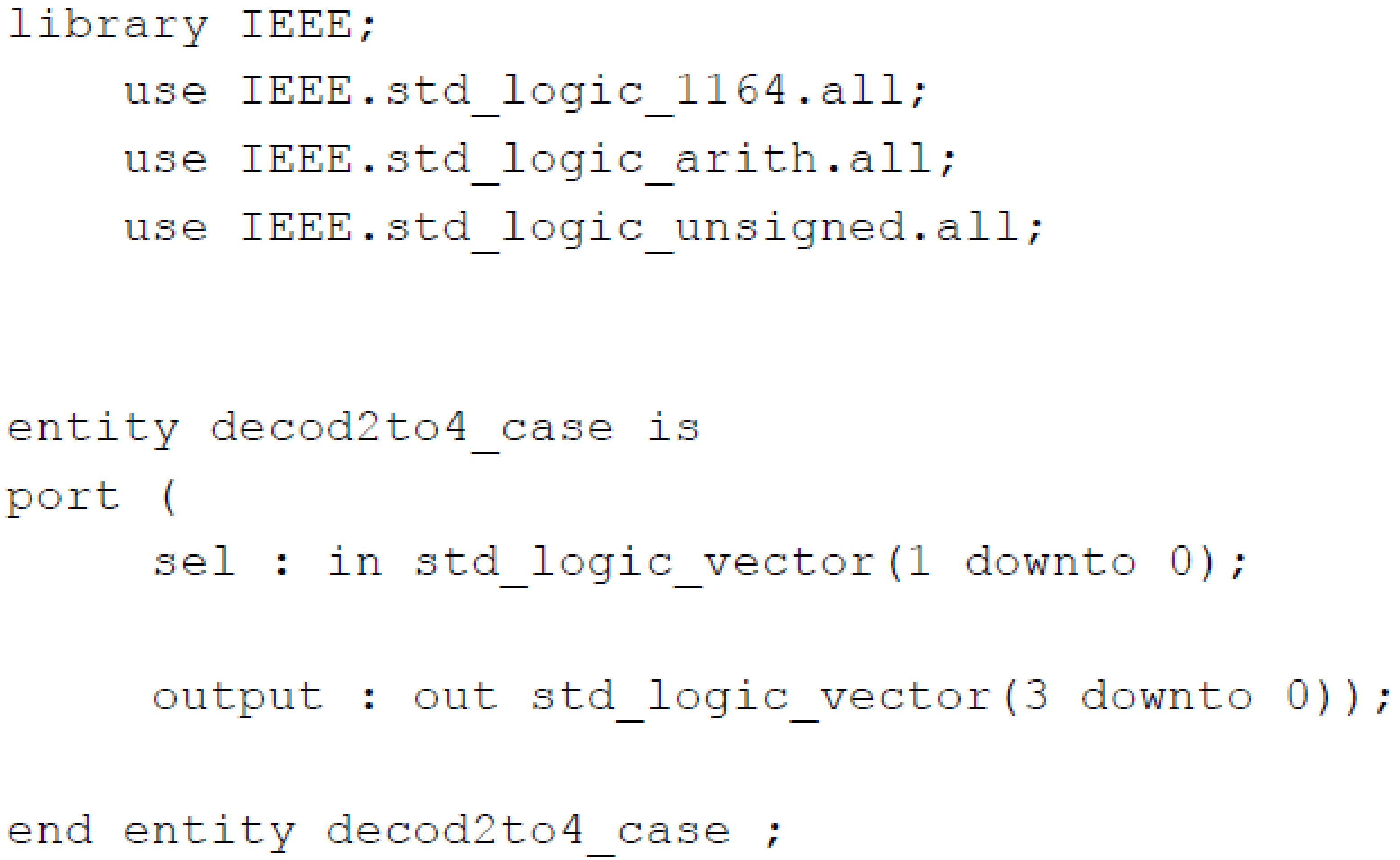
$$D_0 = \overline{A_1} \cdot \overline{A_0}$$

$$D_1 = \overline{A_1} \cdot A_0$$

$$D_2 = A_1 \cdot \overline{A_0}$$

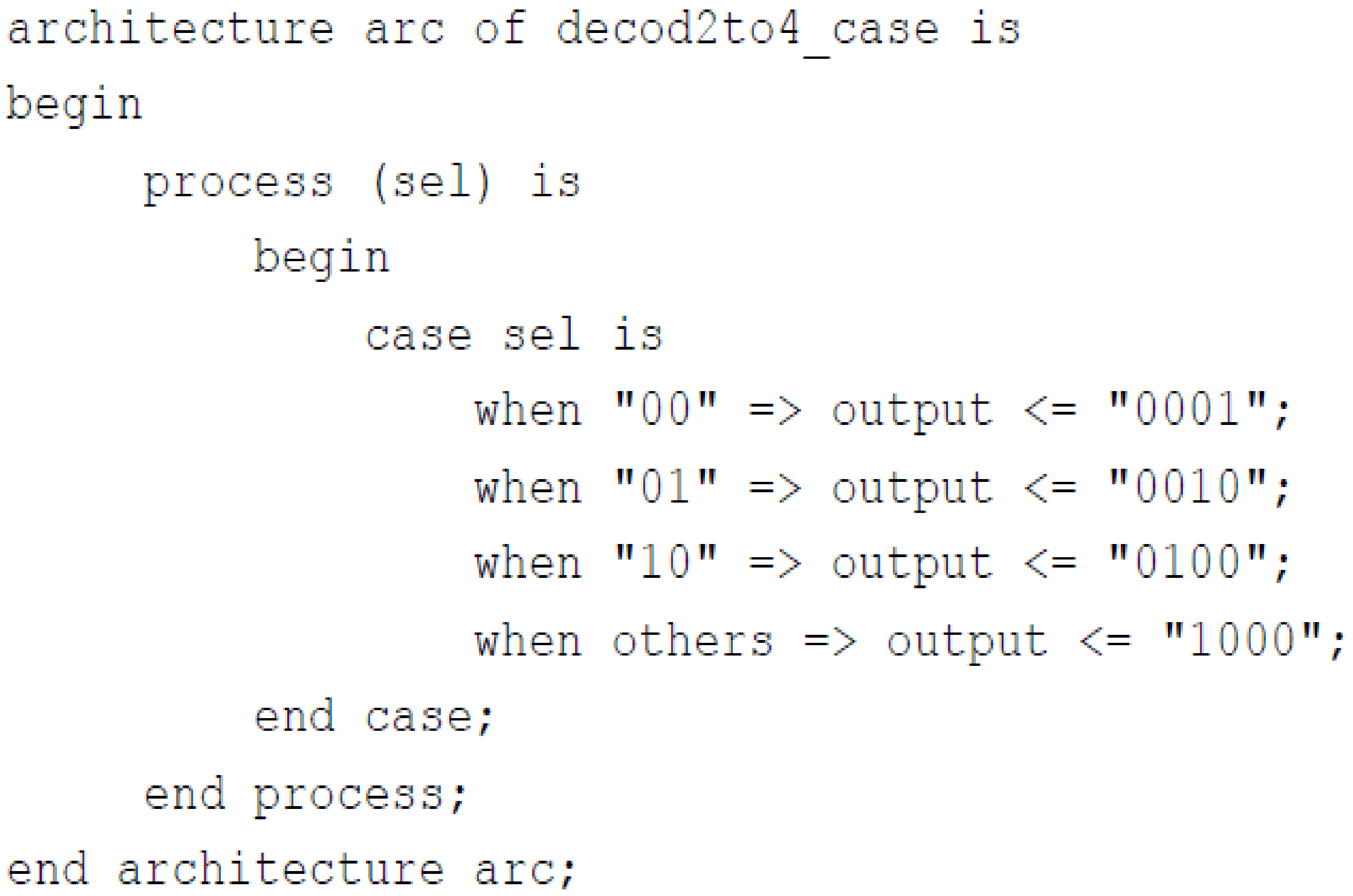
$$D_3 = A_1 \cdot A_0$$

VHDL-CODE



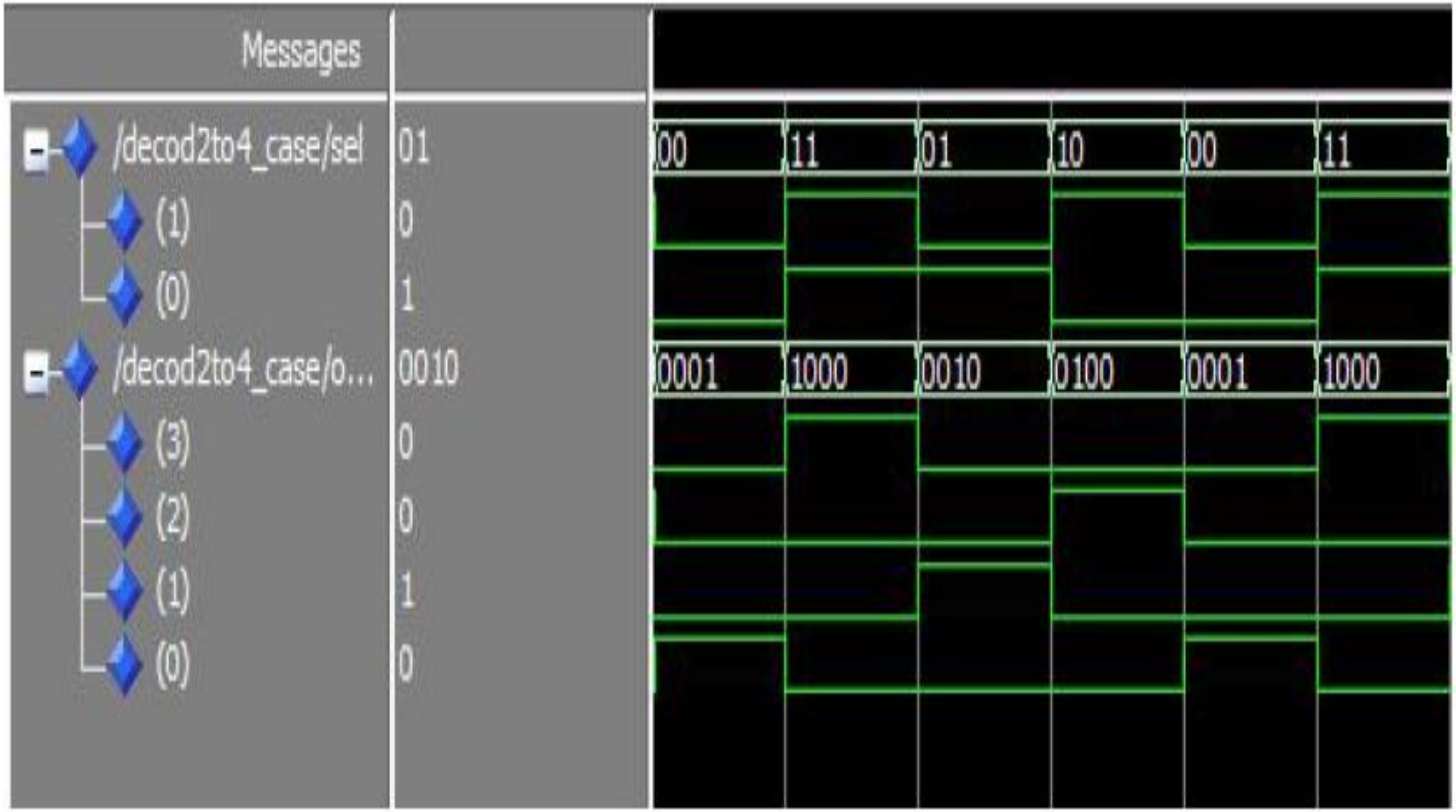
```
library IEEE;
    use IEEE.std_logic_1164.all;
    use IEEE.std_logic_arith.all;
    use IEEE.std_logic_unsigned.all;


entity decod2to4_case is
port (
     sel : in std_logic_vector(1 downto 0);

     output : out std_logic_vector(3 downto 0));

end entity decod2to4_case ;
```

CODE-CONTINUED



```
architecture arc of decod2to4_case is
begin
     process (sel) is
         begin
             case sel is
                 when "00" => output <= "0001";
                 when "01" => output <= "0010";
                 when "10" => output <= "0100";
                 when others => output <= "1000";
         end case;
     end process;
end architecture arc;
```

OUTPUT-WAVEFORM



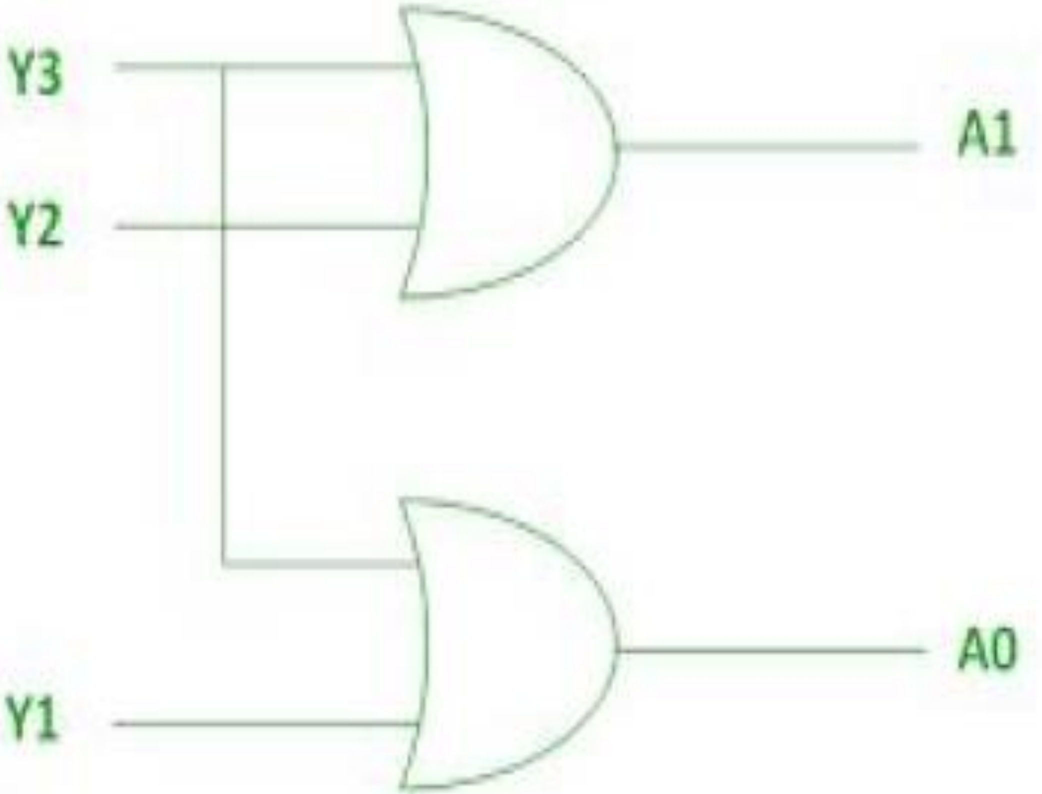
Messages	
/decod2to4_case/sel	01
(1)	0
(0)	1
/decod2to4_case/o...	0010
(3)	0
(2)	0
(1)	1
(0)	0

Write a VHDL program to implement Priority Encoder using If-Else statement.

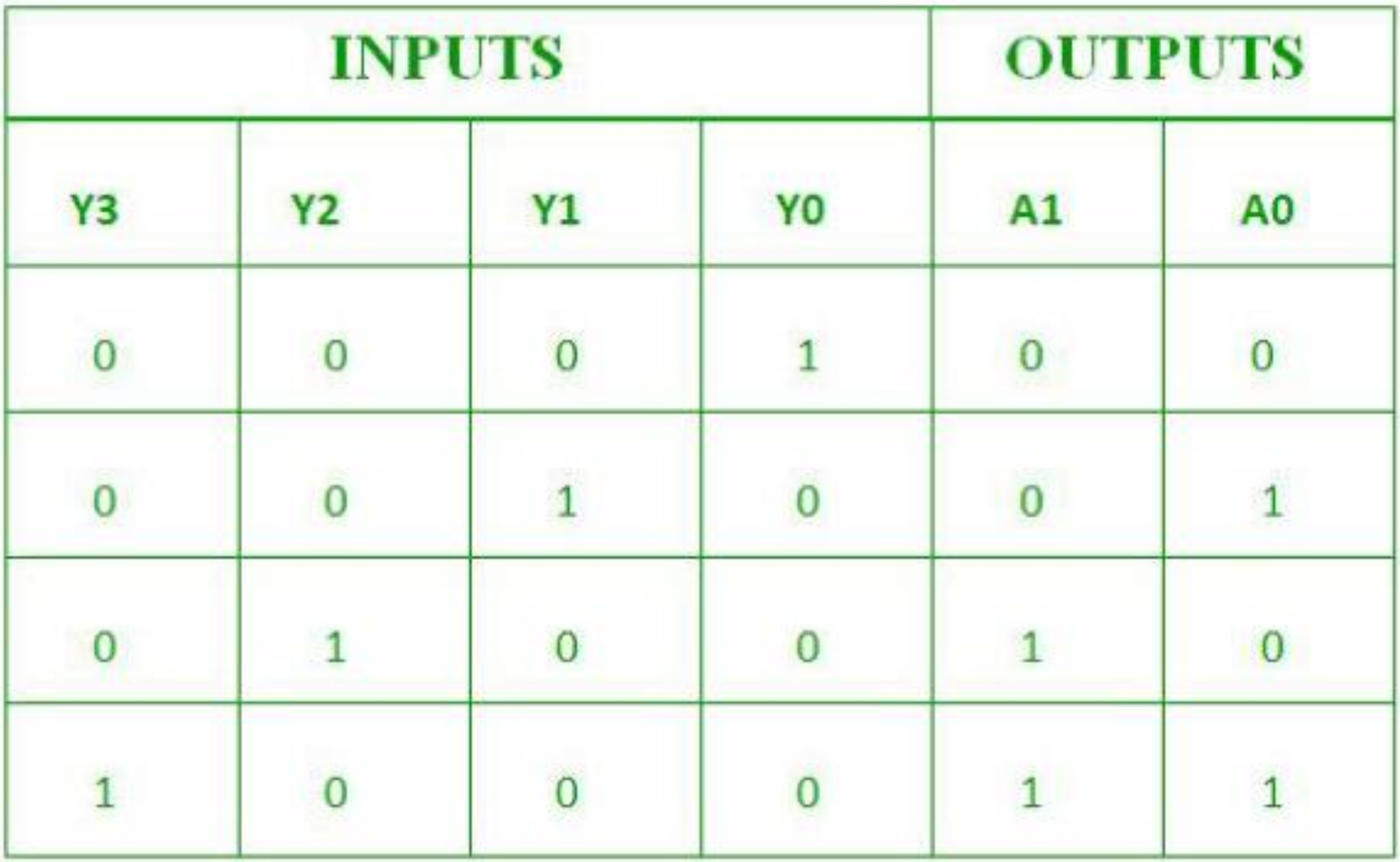
- Title : Implementing a Priority Encoder using If-Else statement.

- Expression : $O_0 = I_1 + I_3$; $O_1 = I_2 + I_3$

Circuit Diagram :

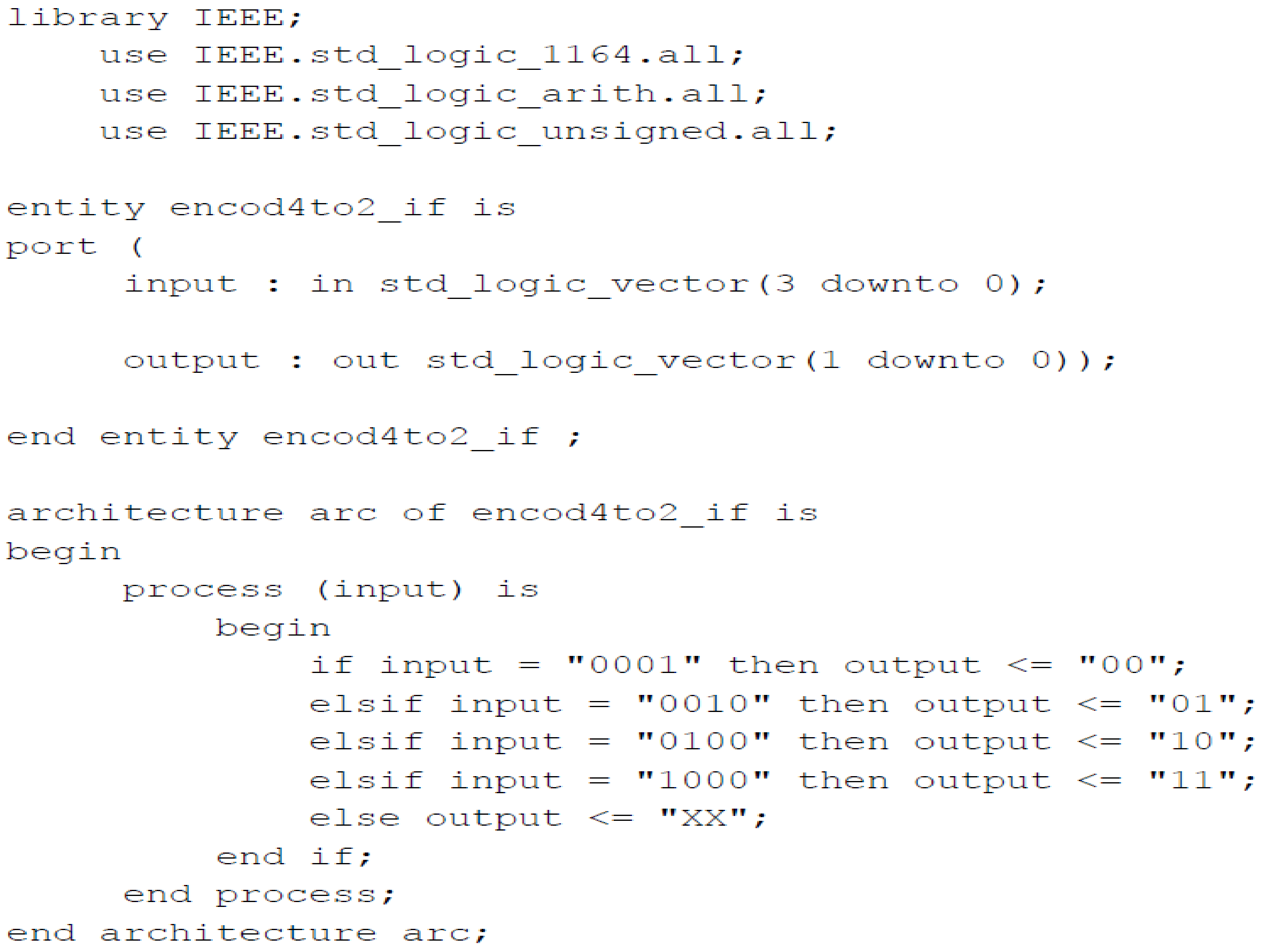


Truth Table :



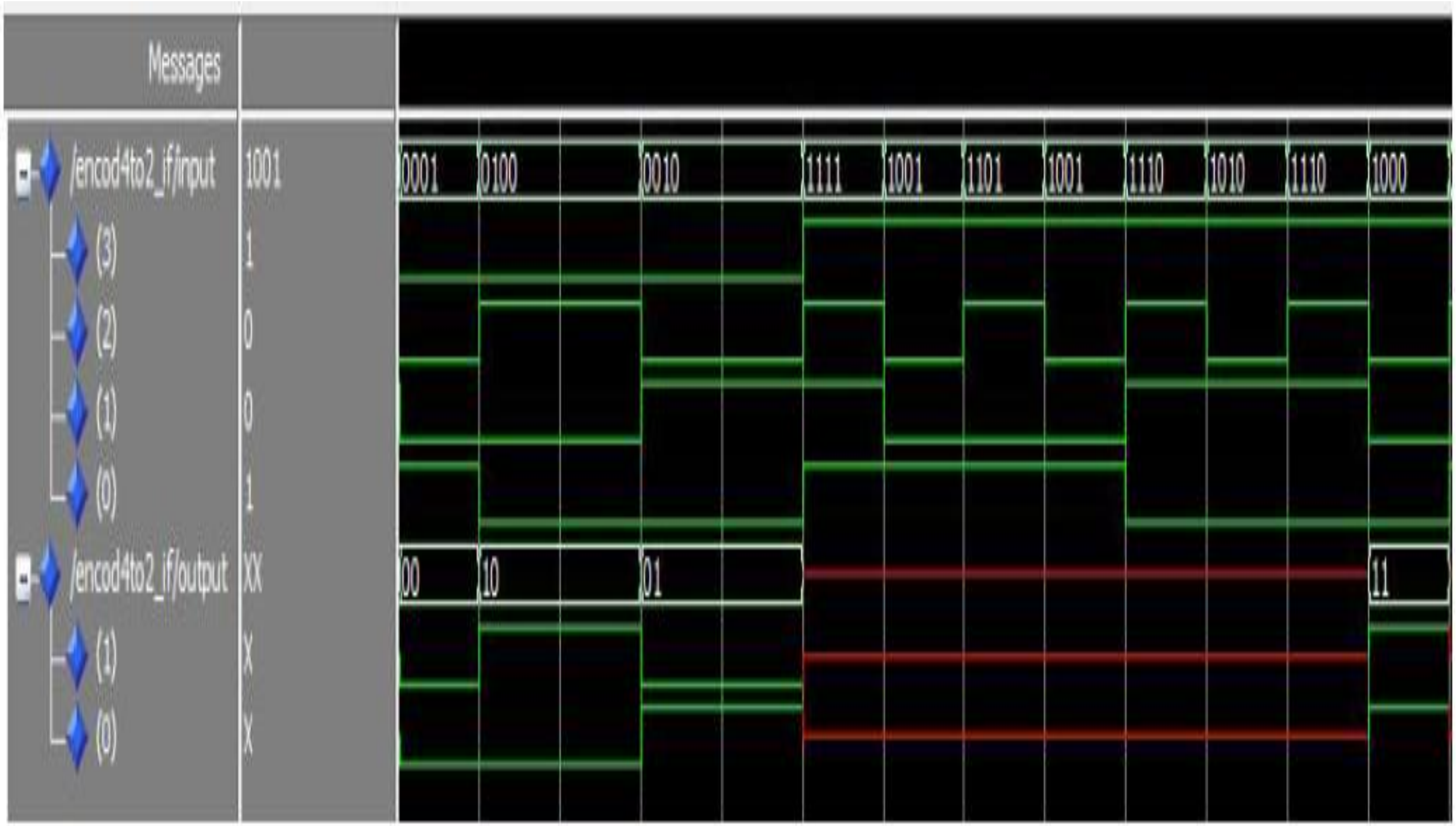
INPUTS				OUTPUTS	
Y3	Y2	Y1	Y0	A1	A0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

VHDL-CODE



```vhdl
library IEEE;
    use IEEE.std_logic_1164.all;
    use IEEE.std_logic_arith.all;
    use IEEE.std_logic_unsigned.all;

entity encod4to2_if is
port (
     input : in std_logic_vector(3 downto 0);

     output : out std_logic_vector(1 downto 0));

end entity encod4to2_if ;

architecture arc of encod4to2_if is
begin
     process (input) is
         begin
             if input = "0001" then output <= "00";
             elsif input = "0010" then output <= "01";
             elsif input = "0100" then output <= "10";
             elsif input = "1000" then output <= "11";
             else output <= "XX";
         end if;
     end process;
end architecture arc;
```

OUTPUT-WAVEFORM

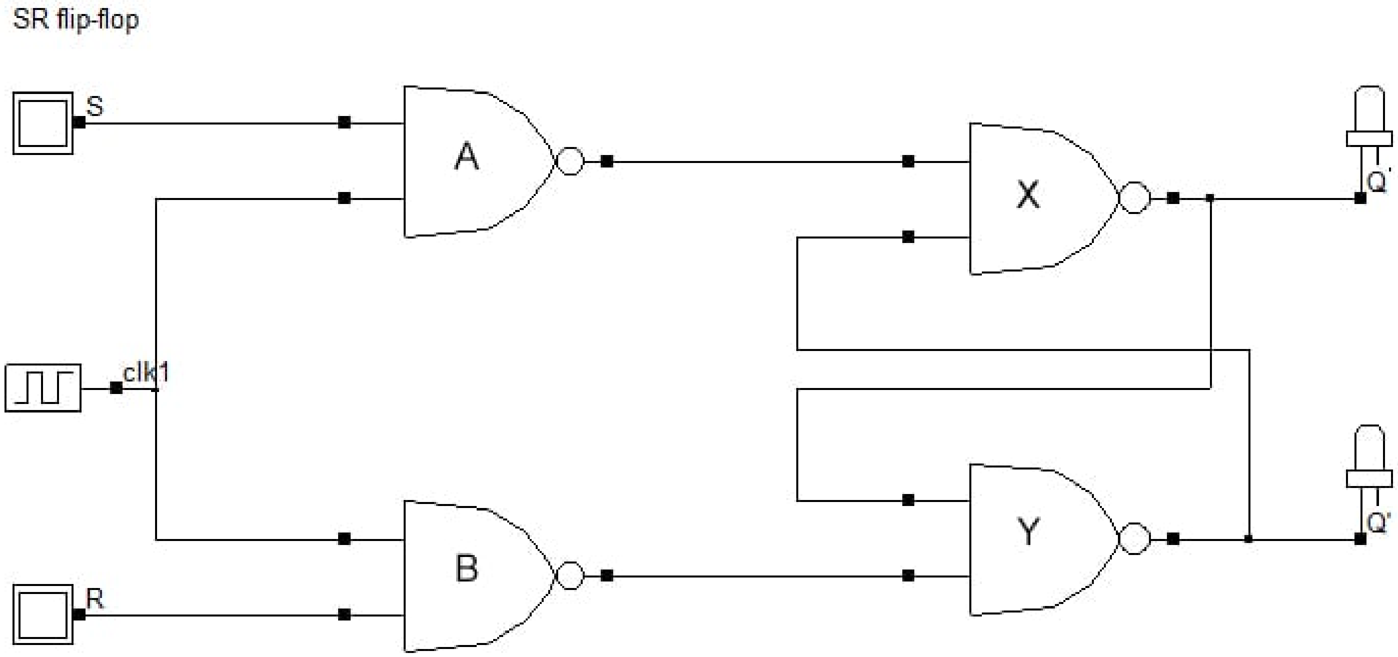


Sequential Circuits

- FLIP FLOP
- REGISTER
- COUNTER

Design a SR flip-flop using VHDL

Circuit-Diagram



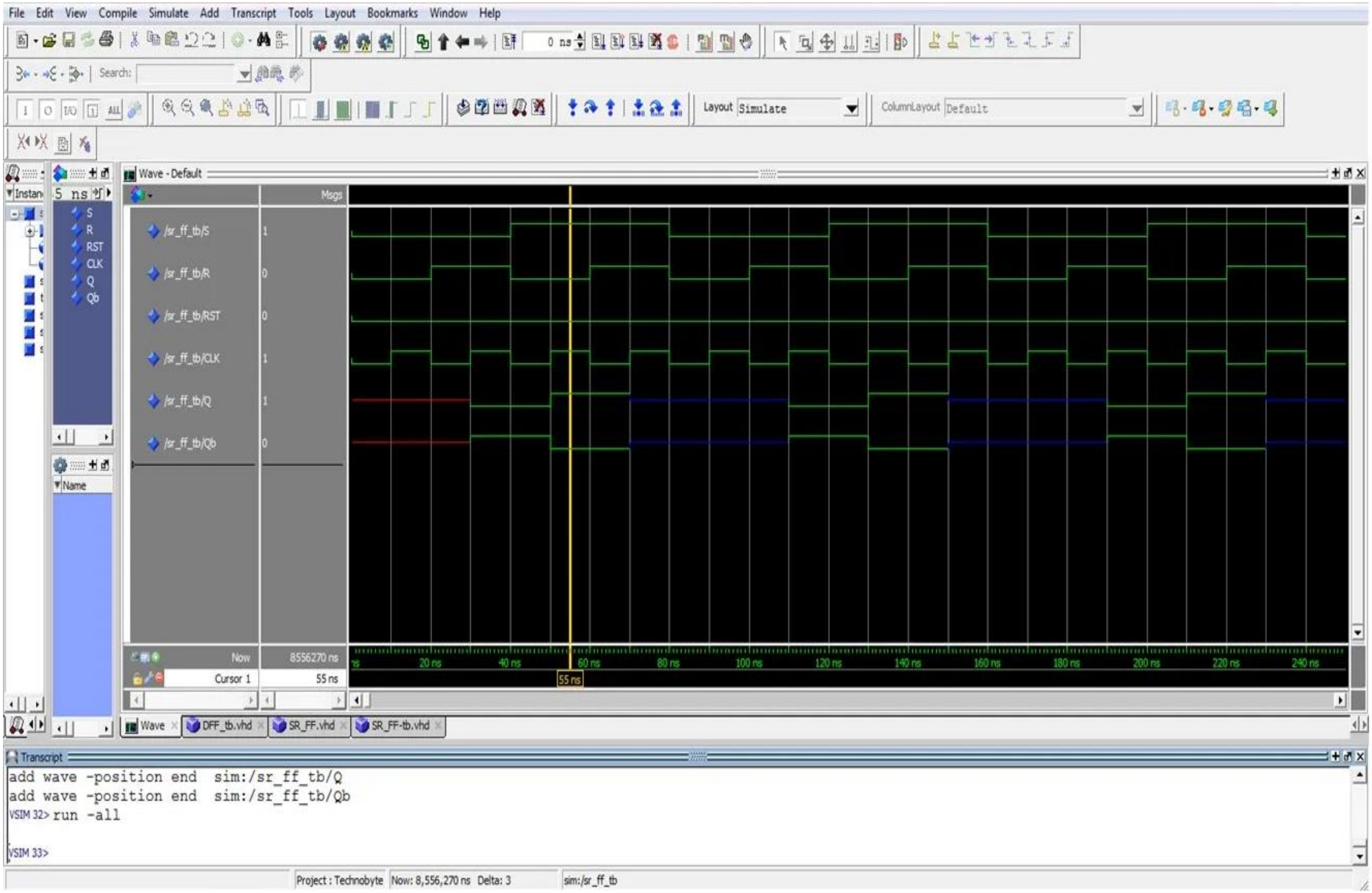
Truth table

CLK	S	R	Q	Q’
0	x	x	Qprv	Q’prv
1	0	0	Qprv	Q’prv
1	0	1	0	1
1	1	0	1	0
1	1	1	–	–

VHDL CODE

```
•     library IEEE;
•     use IEEE.STD_LOGIC_1164.ALL;
•     use IEEE.STD_LOGIC_ARITH.ALL;
•     use IEEE.STD_LOGIC_UNSIGNED.ALL;
•     entity SR_FLIPFLOP_SOURCE is
•         Port ( S,R,RST,CLK : in  STD_LOGIC;
•                Q,Qb : out STD_LOGIC);
•     end SR_FLIPFLOP_SOURCE;

•     architecture Behavioral of SR_FLIPFLOP_SOURCE is

•     begin

•     process (S,R,RST,CLK)
•     Begin
•     if (RST = '1') then
•     Q <= '0';
•     elsif (RISING_EDGE(CLK))then
•     if (S /= R) then
•     Q <= S;
•     Qb <= R;
•     elsif (S = '1' AND R = '1') then
•     Q <= 'Z';
•     Qb <= 'Z';
•     end if;
•     end if;
•     end process;
•     end Behavioral;
```

Waveform



/sr_ff_tb/S 1
/sr_ff_tb/R 0
/sr_ff_tb/RST 0
/sr_ff_tb/CLK 1
/sr_ff_tb/Q 1
/sr_ff_tb/Qb 0

Now 8556270 ns
Cursor 1 55 ns

```
add wave -position end  sim:/sr_ff_tb/Q
add wave -position end  sim:/sr_ff_tb/Qb
VSIM 32> run -all

VSIM 33>
```

Project : Technobyte Now: 8,556,270 ns Delta: 3 sim:/sr_ff_tb