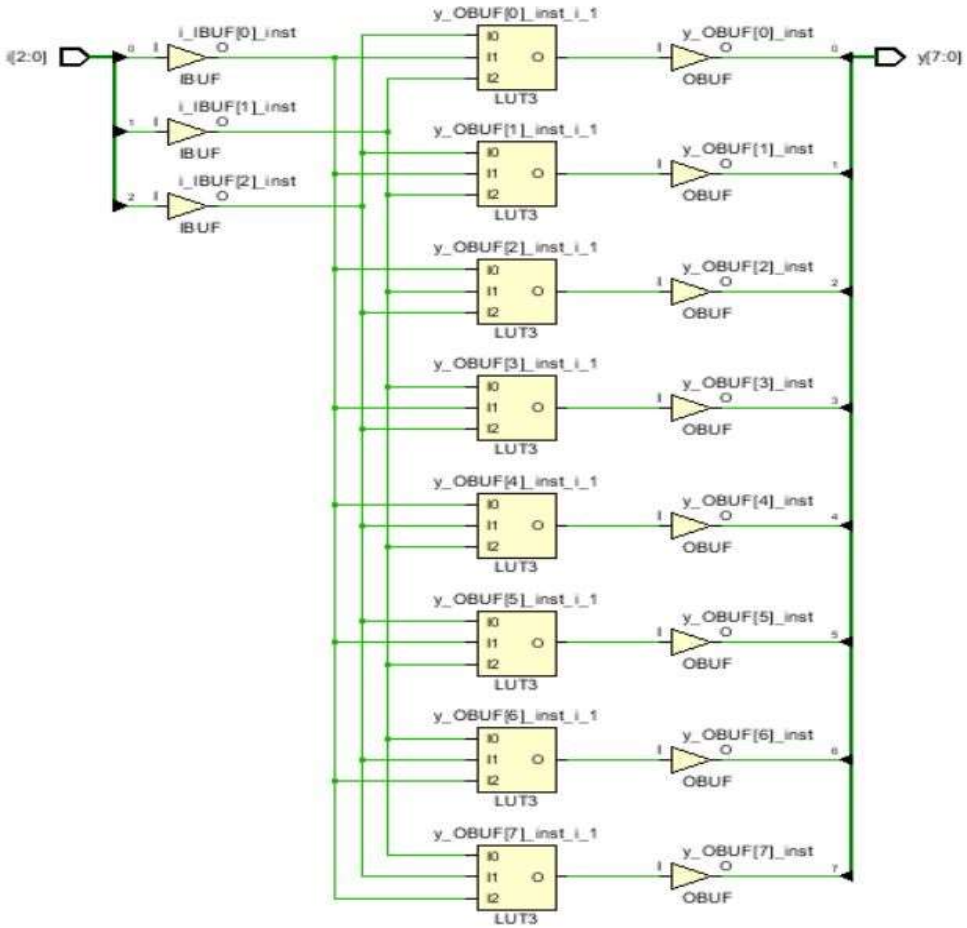
# Practical 2

**Aim**: Write a VHDL code to implement 3x8 decoder using behavioral modelling

**Code:**

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

entity Decoder_3x8 is
    Port ( i : in STD_LOGIC_VECTOR (2 downto 0);
           y : out STD_LOGIC_VECTOR (7 downto 0));
end Decoder_3x8;

architecture Behavioral of Decoder_3x8 is
begin
process(i)
 begin
 case(i) is

when "000" => y<="00000001";
when "001" => y<="00000010";
when "010" => y<="00000100";
when "011" => y<="00001000";
when "100" => y<="00010000";
when "101" => y<="00100000";
when "110" => y<="01000000";
when "111" => y<="10000000";
when others =>y<="10000000";
end case;
end process;
end Behavioral;
```

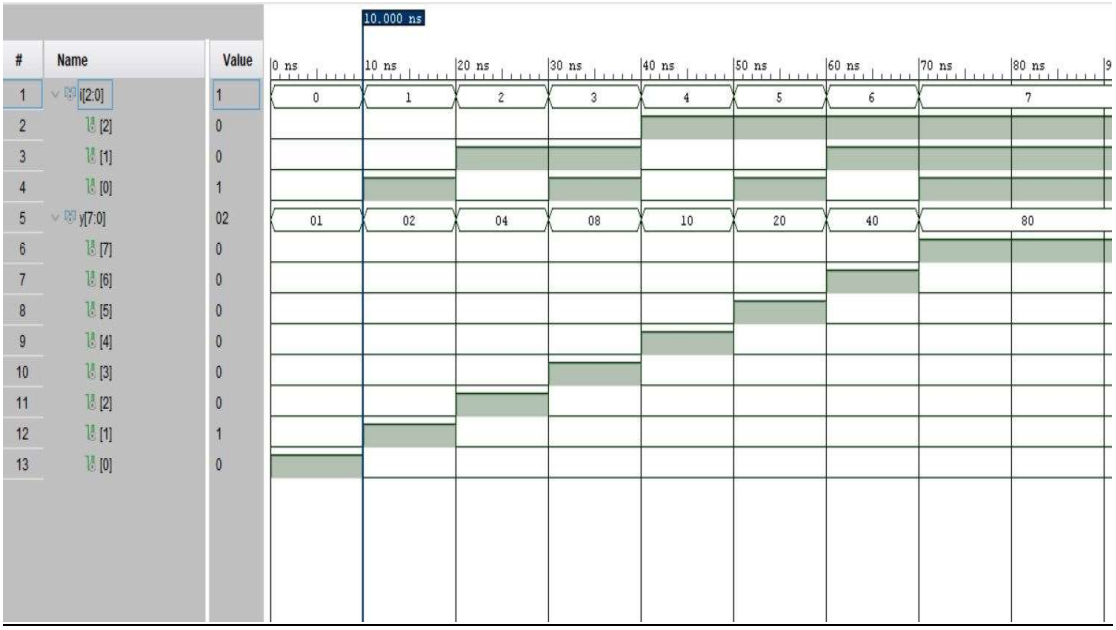
**RTL DIAGRAM:**



**Test bench Code :**

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity Tb_Decoder_3x8 is
--  Port ( );
end Tb_Decoder_3x8;

architecture Behavioral of Tb_Decoder_3x8
is
component Decoder_3x8 is
Port ( i : in STD_LOGIC_VECTOR (2 downto
0);
y : out STD_LOGIC_VECTOR (7 downto 0));
end component Decoder_3x8;
signal i:std_logic_vector(2 downto 0);
signal y:std_logic_vector(7 downto 0);

begin
x1:Decoder_3x8 port map(i,y);
process
begin
i <="000";
wait for 10ns;
i <="001";
wait for 10ns;

i <="010";
wait for 10ns;

i <="011";
wait for 10ns;

i <="100";
wait for 10ns;

i <="101";
wait for 10ns;

i <="110";
wait for 10ns;

i <="111";
wait;
end process;
end Behavioral;
```

**SIMULATION WAVEFORM :**



| Name | Slack ^1 | Levels | Routes | High Fanout | From | To | Total Delay | Logic Delay | Net Delay | Requirement | Source Clock |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Path 1 | ∞ | 3 | 4 | 8 | i[0] | y[1] | 5.377 | 3.778 | 1.599 | ∞ | input port clock |
| Path 2 | ∞ | 3 | 4 | 8 | i[0] | y[3] | 5.377 | 3.778 | 1.599 | ∞ | input port clock |
| Path 3 | ∞ | 3 | 4 | 8 | i[0] | y[5] | 5.377 | 3.778 | 1.599 | ∞ | input port clock |
| Path 4 | ∞ | 3 | 4 | 8 | i[2] | y[7] | 5.377 | 3.778 | 1.599 | ∞ | input port clock |
| Path 5 | ∞ | 3 | 4 | 8 | i[1] | y[0] | 5.351 | 3.752 | 1.599 | ∞ | input port clock |
| Path 6 | ∞ | 3 | 4 | 8 | i[2] | y[2] | 5.351 | 3.752 | 1.599 | ∞ | input port clock |
| Path 7 | ∞ | 3 | 4 | 8 | i[1] | y[4] | 5.351 | 3.752 | 1.599 | ∞ | input port clock |
| Path 8 | ∞ | 3 | 4 | 8 | i[0] | y[6] | 5.351 | 3.752 | 1.599 | ∞ | input port clock |

**SYNTHESIS SUMMARY:**

| Resource | Utilization | Available | Utilization % |
|---|---|---|---|
| LUT | 4 | 17600 | 0.02 |
| IO | 11 | 100 | 11.00 |

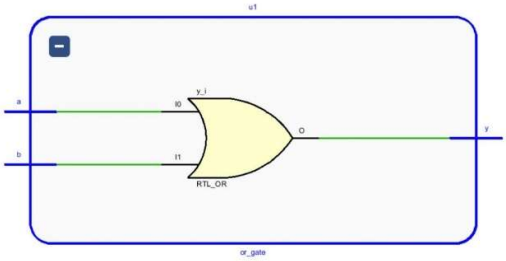
Maximum Combinational Delay: 5.377nSec

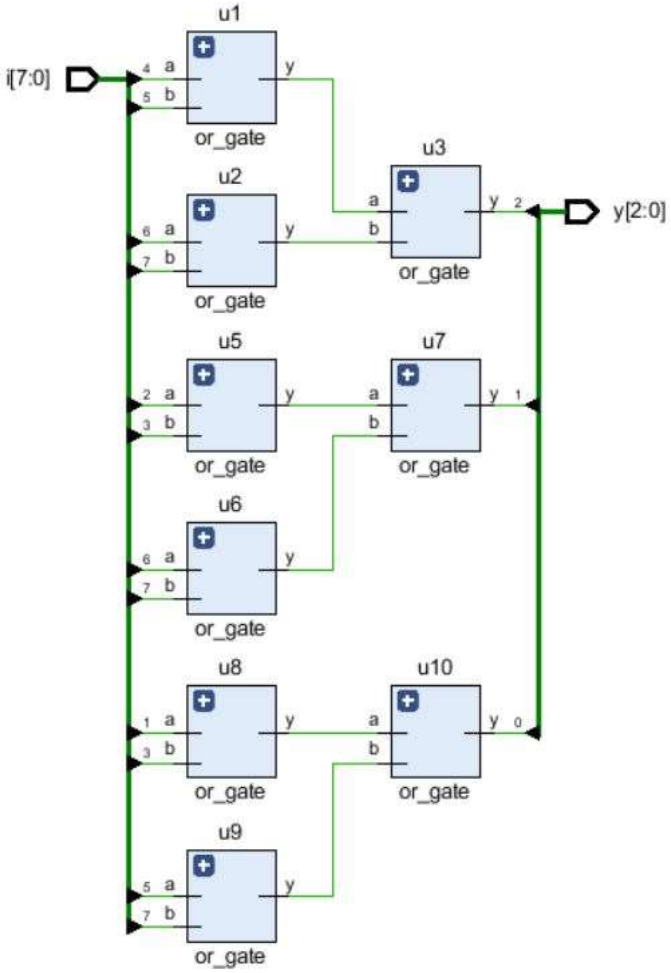
**Aim**: Write a VHDL code to implement 8x3 encoder using structural modelling

**Code:**

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

entity Encoder_8x3 is
   Port ( i : in STD_LOGIC_VECTOR (7 downto 0);
          y : out STD_LOGIC_VECTOR (2 downto 0));
end Encoder_8x3;

architecture Behavioral of Encoder_8x3 is
component or_gate
 Port ( a : in STD_LOGIC;
        b : in STD_LOGIC;
        y : out STD_LOGIC);
end component or_gate;
signal x: std_logic_vector(0 to 5);
begin

u1:or_gate port map(i(4),i(5),x(0));
u2:or_gate port map(i(6),i(7),x(1));
u3:or_gate port map(x(0),x(1),y(2));
u5:or_gate port map(i(2),i(3),x(2));
u6:or_gate port map(i(6),i(7),x(3));
u7:or_gate port map(x(2),x(3),y(1));
u8:or_gate port map(i(1),i(3),x(4));
u9:or_gate port map(i(5),i(7),x(5));
u10:or_gate port map(x(4),x(5),y(0));

end Behavioral;
```

**RTL DIAGRAM:**

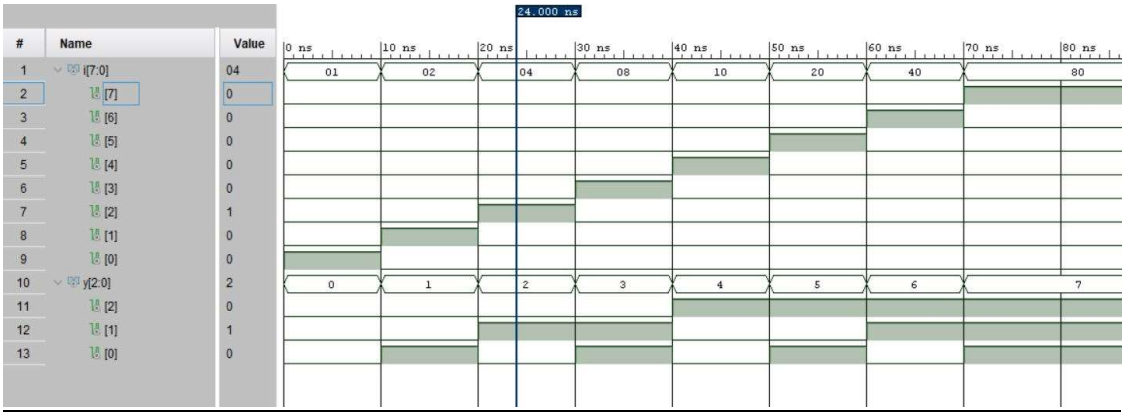


**Test bench Code :**

```
library IEEE;
use
IEEE.STD_LOGIC_1164.ALL;
entity Tb_Encoder_8x3 is
--  Port ( );
end Tb_Encoder_8x3;

architecture Behavioral of
Tb_Encoder_8x3 is
component Encoder_8x3
is
Port ( i : in
STD_LOGIC_VECTOR (7
downto 0);
y : out
STD_LOGIC_VECTOR (2
downto 0));
end component
Encoder_8x3;
signal i :
STD_LOGIC_VECTOR (7
downto 0);
signal y :
STD_LOGIC_VECTOR (2
downto 0);
begin
x:Encoder_8x3 port
map(i,y);
process
begin

i<="00000001";
wait for 10ns;
i<="00000010";
wait for 10ns;
i<="00000100";
wait for 10ns;
i<="00001000";
wait for 10ns;
i<="00010000";
wait for 10ns;
i<="00100000";
wait for 10ns;
i<="01000000";
wait for 10ns;
i<="10000000";
wait;
end process;

end Behavioral;
```

**SIMULATION WAVEFORM :**




| Name | Slack | Levels | Routes | High Fanout | From | To | Total Delay | Logic Delay | Net Delay | Requirement | Source Clock |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Path 1 | ∞ | 3 | 4 | 2 | i[5] | y[0] | 5.351 | 3.752 | 1.599 | ∞ | input port clock |
| Path 2 | ∞ | 3 | 4 | 2 | i[6] | y[1] | 5.351 | 3.752 | 1.599 | ∞ | input port clock |
| Path 3 | ∞ | 3 | 4 | 2 | i[6] | y[2] | 5.351 | 3.752 | 1.599 | ∞ | input port clock |

**SYNTHESIS SUMMARY:**

| Resource | Utilization | Available | Utilization % |
|---|---|---|---|
| LUT | 3 | 17600 | 0.02 |
| IO | 10 | 100 | 10.00 |

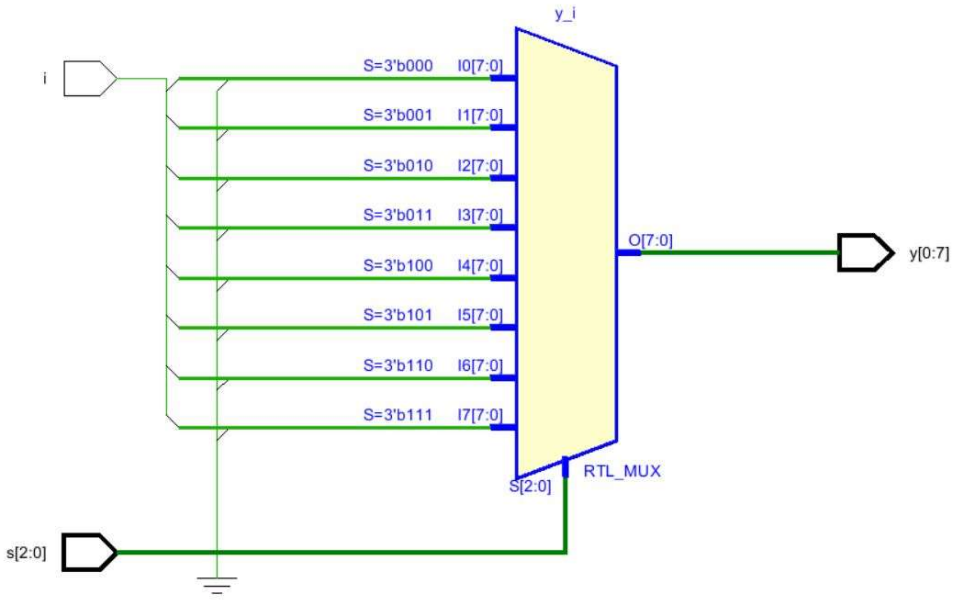
Maximum Combinational Delay: 5.351nSec

**Aim**: Write a VHDL code to implement 1x8 Demultiplexer using behavioral modelling

**Code:**

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

entity Demux1x8 is
Port ( i : in STD_LOGIC;
y : out STD_LOGIC_VECTOR (0 to 7);
s : in STD_LOGIC_VECTOR (2 downto 0));
end Demux1x8;

architecture Behavioral of Demux1x8 is

begin
process(i,s)
begin
case(s) is
when "000" => y(0)<=i;
y(1 to 7)<="0000000";
when "001" => y(1)<=i;
y(0)<='0';
y(2 to 7)<="000000";
when "010" => y(2)<=i;
y(0 to 1)<="00";
y(3 to 7)<="00000";
when "011" => y(3)<=i;
y(0 to 2)<="000";
y(4 to 7)<="0000";
when "100" => y(4)<=i;
y(0 to 3)<="0000";
y(5 to 7)<="000";
when "101" => y(5)<=i;
y(0 to 4)<="00000";
y(6 to 7)<="00";
when "110" => y(6)<=i;
y(0 to 5)<="000000";
y(7)<='0';
when "111" => y(7)<=i;
y(0 to 6)<="0000000";
when others => y<="00000000";
end case;
end process;
end Behavioral;
```

**RTL DIAGRAM:**

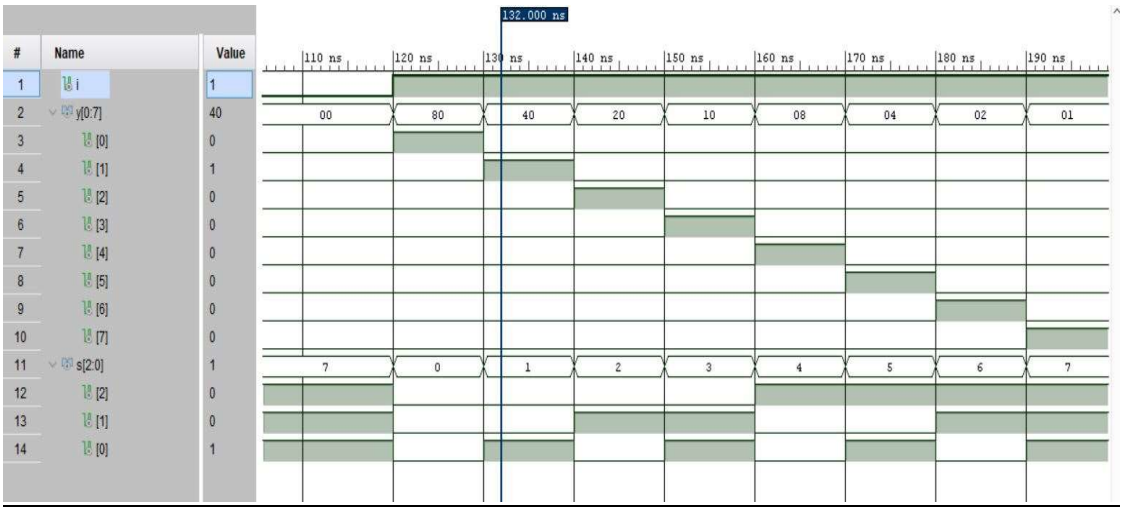



**Test bench Code :**

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity Tb_Demux1x8 is
--  Port ( );
end Tb_Demux1x8;
architecture Behavioral of Tb_Demux1x8 is
component Demux1x8 is
   Port ( i : in STD_LOGIC;
          y : out STD_LOGIC_VECTOR (0 to 7);
          s : in STD_LOGIC_VECTOR (2 downto
0));
end component Demux1x8;
signal i : STD_LOGIC;
signal y : STD_LOGIC_VECTOR (0 to 7);
signal s : STD_LOGIC_VECTOR (2 downto 0);
begin
x1:Demux1x8 port map(i,y,s);
process
begin
i<='0';
s<="000";
wait for 10 ns;
s<="001";
wait for 10 ns;
s<="010";
wait for 10 ns;
s<="011";
wait for 10 ns;
s<="100";
wait for 10 ns;
s<="101";
wait for 10 ns;
s<="110";
wait for 10 ns;
s<="111";
wait for 50 ns;

i<='1';
s<="000";
wait for 10 ns;
s<="001";
wait for 10 ns;
s<="010";
wait for 10 ns;
s<="011";
wait for 10 ns;
s<="100";
```

```
wait for 10 ns;
s<="101";
wait for 10 ns;
s<="110";
wait for 10 ns;
s<="111";
wait;
end process;
end Behavioral;
```

**SIMULATION WAVEFORM :**




| Name | Slack | Levels | Routes | High Fanout | From | To | Total Delay | Logic Delay | Net Delay | Requirement | Source Clock |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Path 1 | ∞ | 3 | 4 | 8 | s[1] | y[0] | 5.379 | 3.780 | 1.599 | ∞ | input port clock |
| Path 2 | ∞ | 3 | 4 | 8 | s[0] | y[3] | 5.379 | 3.780 | 1.599 | ∞ | input port clock |
| Path 3 | ∞ | 3 | 4 | 8 | s[2] | y[4] | 5.379 | 3.780 | 1.599 | ∞ | input port clock |
| Path 4 | ∞ | 3 | 4 | 8 | s[2] | y[7] | 5.379 | 3.780 | 1.599 | ∞ | input port clock |
| Path 5 | ∞ | 3 | 4 | 8 | s[2] | y[1] | 5.351 | 3.752 | 1.599 | ∞ | input port clock |
| Path 6 | ∞ | 3 | 4 | 8 | s[2] | y[2] | 5.351 | 3.752 | 1.599 | ∞ | input port clock |
| Path 7 | ∞ | 3 | 4 | 8 | i | y[5] | 5.351 | 3.752 | 1.599 | ∞ | input port clock |
| Path 8 | ∞ | 3 | 4 | 8 | s[1] | y[6] | 5.351 | 3.752 | 1.599 | ∞ | input port clock |

**SYNTHESIS SUMMARY:**

| Resource | Utilization | Available | Utilization % |
|---|---|---|---|
| LUT | 4 | 17600 | 0.02 |
| IO | 12 | 100 | 12.00 |

Maximum Combinational Delay: 5.379nSec

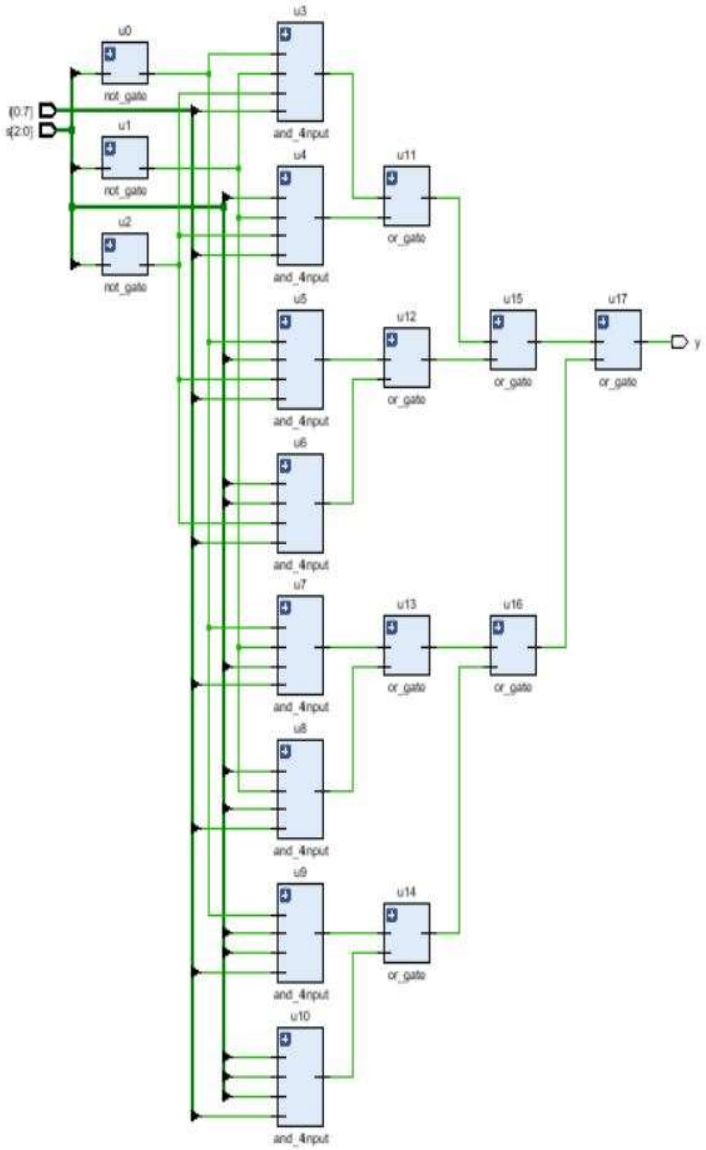
**Aim**: Write a VHDL code to implement 8x1 multiplexer using structural modelling

**Code:**

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

entity Mux_8x1 is
    Port ( i : in STD_LOGIC_VECTOR (0 to 7);
           s : in STD_LOGIC_VECTOR (2 downto 0);
           y : out STD_LOGIC);
end Mux_8x1;

architecture Behavioral of Mux_8x1 is
component or_gate is
    Port ( a : in STD_LOGIC;
           b : in STD_LOGIC;
           y : out STD_LOGIC);
end component or_gate;

component and_4input is
   Port ( a1 : in STD_LOGIC;
          a2 : in STD_LOGIC;
          a3 : in STD_LOGIC;
          a4 : in STD_LOGIC;
          y : out STD_LOGIC);
end component and_4input;
component not_gate is
    Port ( a : in STD_LOGIC;
           y : out STD_LOGIC);
end component not_gate;

signal sb:std_logic_vector(0 to 2);

signal x:std_logic_vector(0 to 13);
begin
u0:not_gate port map(s(0),sb(0));
u1:not_gate port map(s(1),sb(1));
u2:not_gate port map(s(2),sb(2));

u3:and_4input port map(sb(0),sb(1),sb(2),i(0),x(0));
u4:and_4input port map(s(0),sb(1),sb(2),i(1),x(1));
u5:and_4input port map(sb(0),s(1),sb(2),i(2),x(2));
u6:and_4input port map(s(0),s(1),sb(2),i(3),x(3));
u7:and_4input port map(sb(0),sb(1),s(2),i(4),x(4));
u8:and_4input port map(s(0),sb(1),s(2),i(5),x(5));
u9:and_4input port map(sb(0),s(1),s(2),i(6),x(6));
u10:and_4input port map(s(0),s(1),s(2),i(7),x(7));

u11:or_gate port map(x(0),x(1),x(8));
u12:or_gate port map(x(2),x(3),x(9));
```

```
u13:or_gate port map(x(4),x(5),x(10));
u14:or_gate port map(x(6),x(7),x(11));
u15:or_gate port map(x(8),x(9),x(12));
u16:or_gate port map(x(10),x(11),x(13));
u17:or_gate port map(x(12),x(13),y);

end Behavioral;
```

**RTL DIAGRAM:**

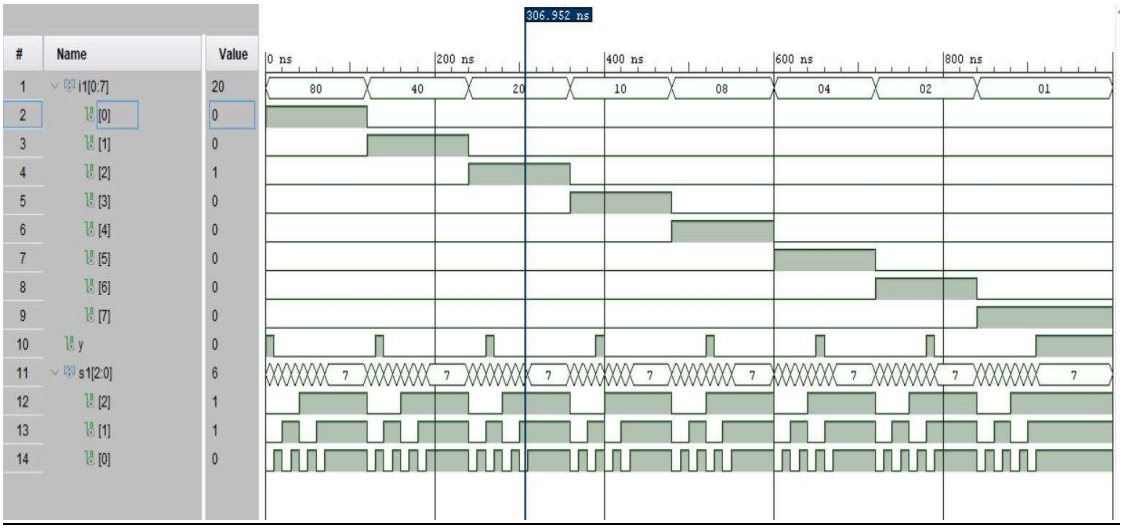


**Test bench Code :**

```
library IEEE;
use
IEEE.STD_LOGIC_1164.ALL;
entity Tb_Mux_8x1 is
--  Port ( );
end Tb_Mux_8x1;
architecture Behavioral of
Tb_Mux_8x1 is
component Mux_8x1 is
   Port ( i : in
STD_LOGIC_VECTOR (0 to
7);
        s : in
STD_LOGIC_VECTOR (2
downto 0);
        y : out STD_LOGIC);
end component Mux_8x1;
signal
i1:STD_LOGIC_VECTOR (0
to 7);
signal y: std_logic;
signal s1:
std_logic_vector(2 downto
0);
begin
u1:Mux_8x1 port
map(i1,s1,y);
process
begin
i1<="10000000";
s1<="000";
wait for 10ns;
s1<="001";
wait for 10ns;
s1<="010";
wait for 10ns;
s1<="011";
wait for 10ns;
s1<="100";
wait for 10ns;
s1<="101";
wait for 10ns;
s1<="110";
wait for 10ns;
s1<="111";
wait for 50ns;

i1<="01000000";
s1<="000";
wait for 10ns;
s1<="001";
wait for 10ns;
s1<="010";
wait for 10ns;
s1<="011";
wait for 10ns;
s1<="100";
wait for 10ns;
s1<="101";
wait for 10ns;
s1<="110";
wait for 10ns;
s1<="111";
wait for 50ns;

i1<="00100000";
s1<="000";
wait for 10ns;
s1<="001";
wait for 10ns;
s1<="010";
wait for 10ns;
s1<="011";
wait for 10ns;
s1<="100";
wait for 10ns;
s1<="101";
wait for 10ns;
s1<="110";
wait for 10ns;
s1<="111";
wait for 50ns;

i1<="00010000";
s1<="000";
wait for 10ns;
s1<="001";
wait for 10ns;
s1<="010";
wait for 10ns;
s1<="011";
wait for 10ns;
s1<="100";
wait for 10ns;
s1<="101";
wait for 10ns;
s1<="110";
wait for 10ns;
s1<="111";
wait for 50ns;
i1<="00001000";
s1<="000";
wait for 10ns;
s1<="001";
wait for 10ns;
s1<="010";
wait for 10ns;
s1<="011";
wait for 10ns;
s1<="100";
wait for 10ns;
s1<="101";
wait for 10ns;
s1<="110";
wait for 10ns;
s1<="111";
wait for 50ns;
i1<="00000100";
s1<="000";
wait for 10ns;
s1<="001";
wait for 10ns;
s1<="010";
wait for 10ns;
s1<="011";
wait for 10ns;
s1<="100";
wait for 10ns;
s1<="101";
```

```
wait for 10ns;
s1<="110";
wait for 10ns;
s1<="111";
wait for 50ns;

i1<="00000010";
s1<="000";
wait for 10ns;
s1<="001";
wait for 10ns;
s1<="010";
wait for 10ns;
s1<="011";
wait for 10ns;
s1<="100";
wait for 10ns;
s1<="101";
wait for 10ns;
s1<="110";
wait for 10ns;
s1<="111";
wait for 50ns;

i1<="00000001";
s1<="000";
wait for 10ns;
s1<="001";
wait for 10ns;
s1<="010";
wait for 10ns;
s1<="011";
wait for 10ns;
s1<="100";
wait for 10ns;
s1<="101";
wait for 10ns;
s1<="110";
wait for 10ns;
s1<="111";
wait;
end process;
end Behavioral;
```

**SIMULATION WAVEFORM :**




| Name | Slack | Levels | Routes | High Fanout | From | To | Total Delay | Logic Delay | Net Delay | Requirement | Source Clock |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Path 1 | ∞ | 4 | 5 | 1 | i[4] | y | 5.770 | 4.171 | 1.599 | ∞ | input port clock |

**SYNTHESIS SUMMARY:**

| Resource | Utilization | Available | Utilization % |
|---|---|---|---|
| LUT | 2 | 17600 | 0.01 |
| IO | 12 | 100 | 12.00 |

Maximum Combinational Delay: 5.77nSec