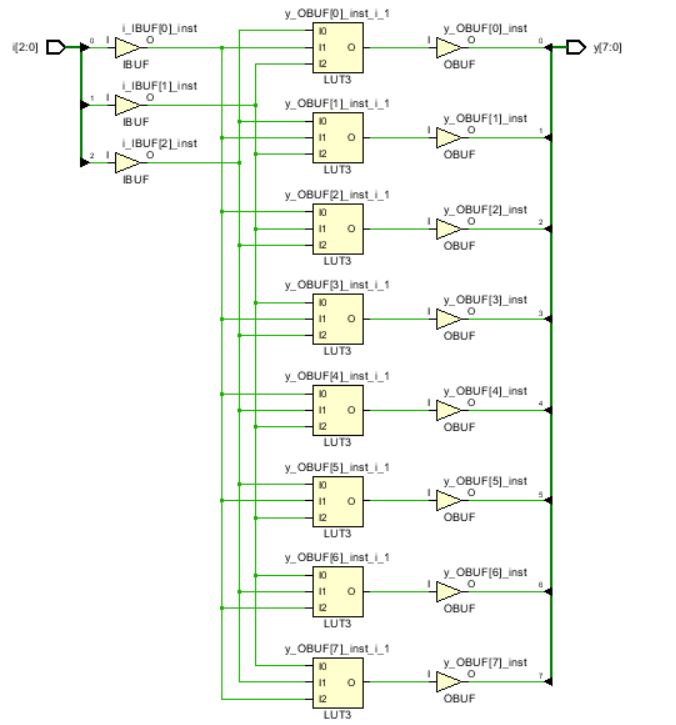
**Practical 2**

|  |
| --- |
| **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;

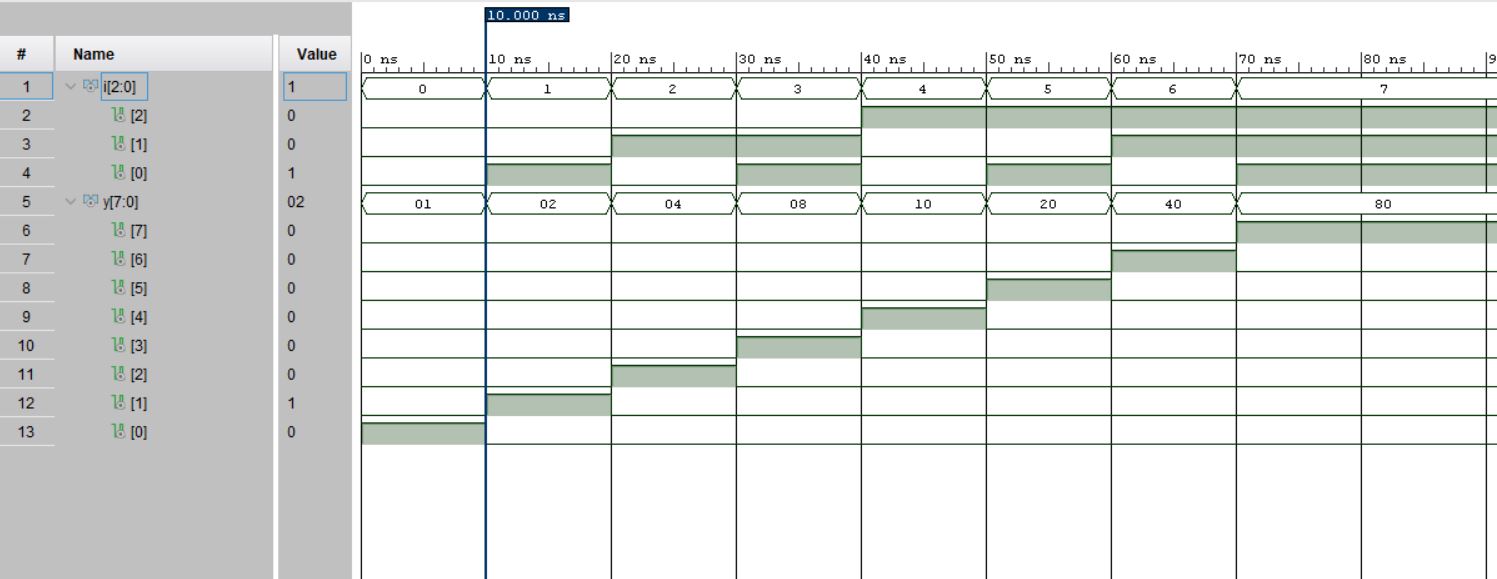
i <="111";

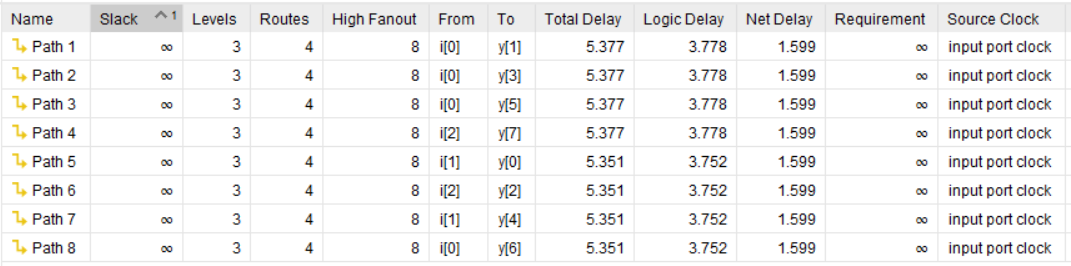
wait;

end process;

end Behavioral;

**SIMULATION WAVEFORM :**

****



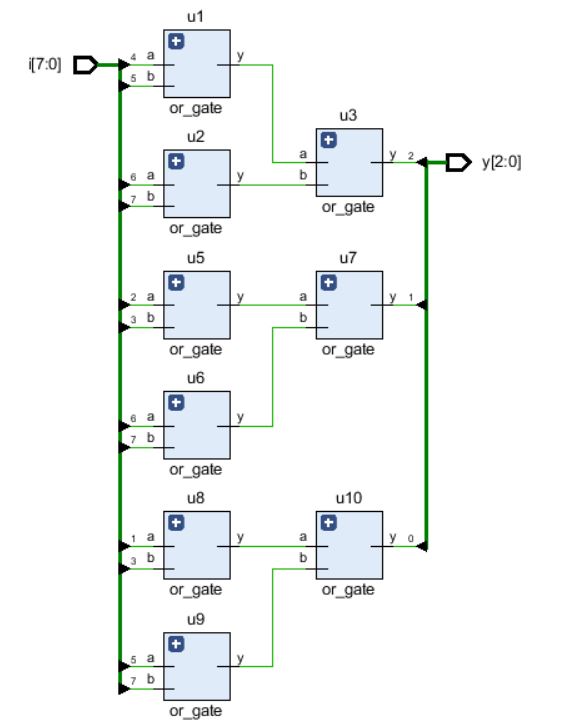
**SYNTHESIS SUMMARY:**

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

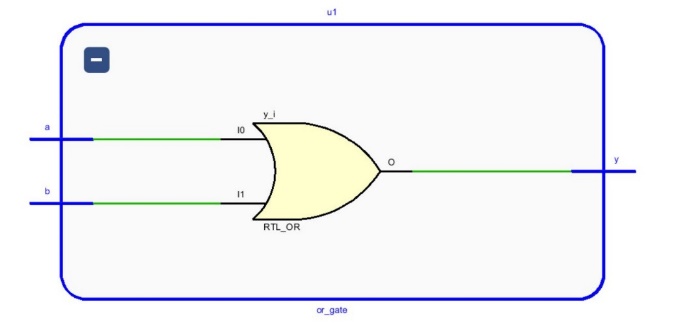
Maximum Combinational Delay: 5.377nSec

**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;

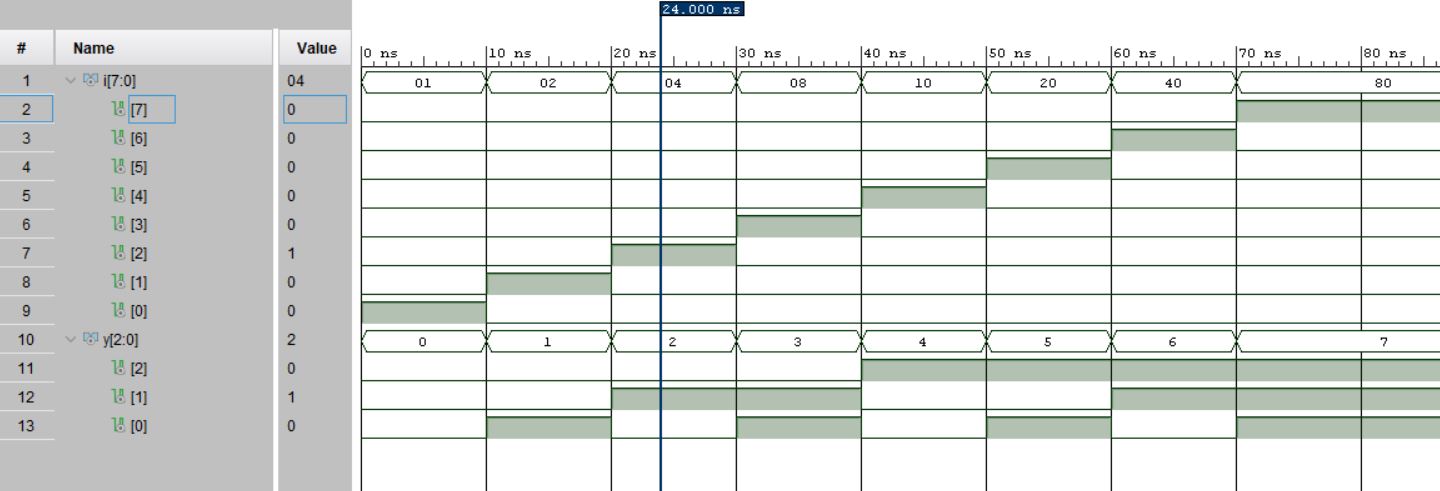
i<="10000000";

wait;

end process;

end Behavioral;

**SIMULATION WAVEFORM :**

****

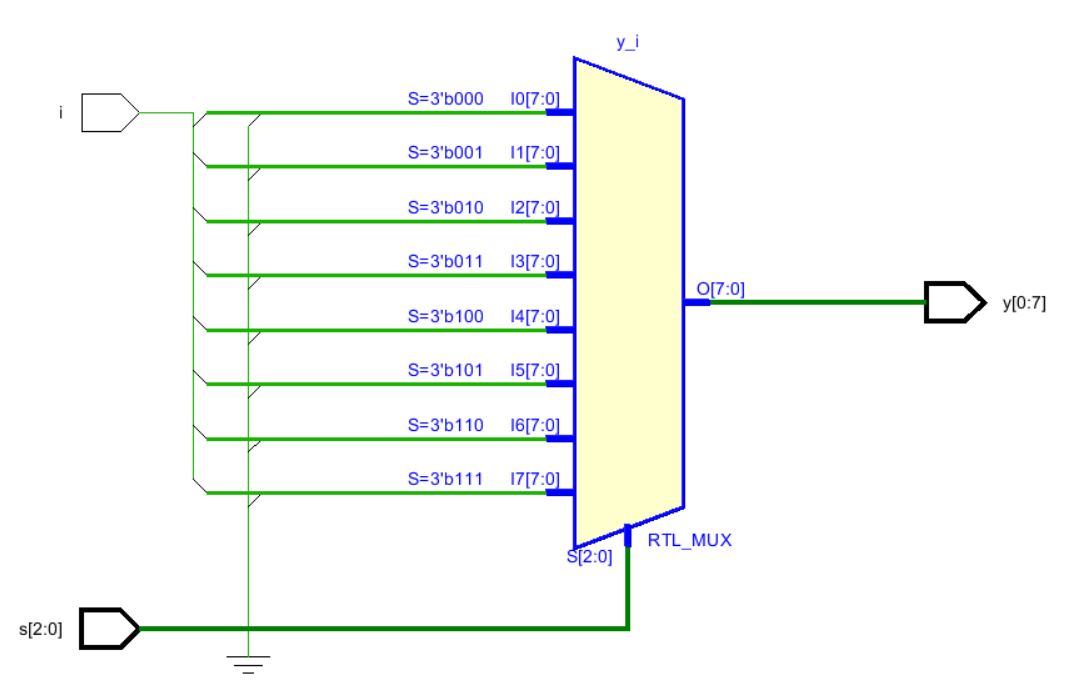


**SYNTHESIS SUMMARY:**

|  |  |  |  |
| --- | --- | --- | --- |
| **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;

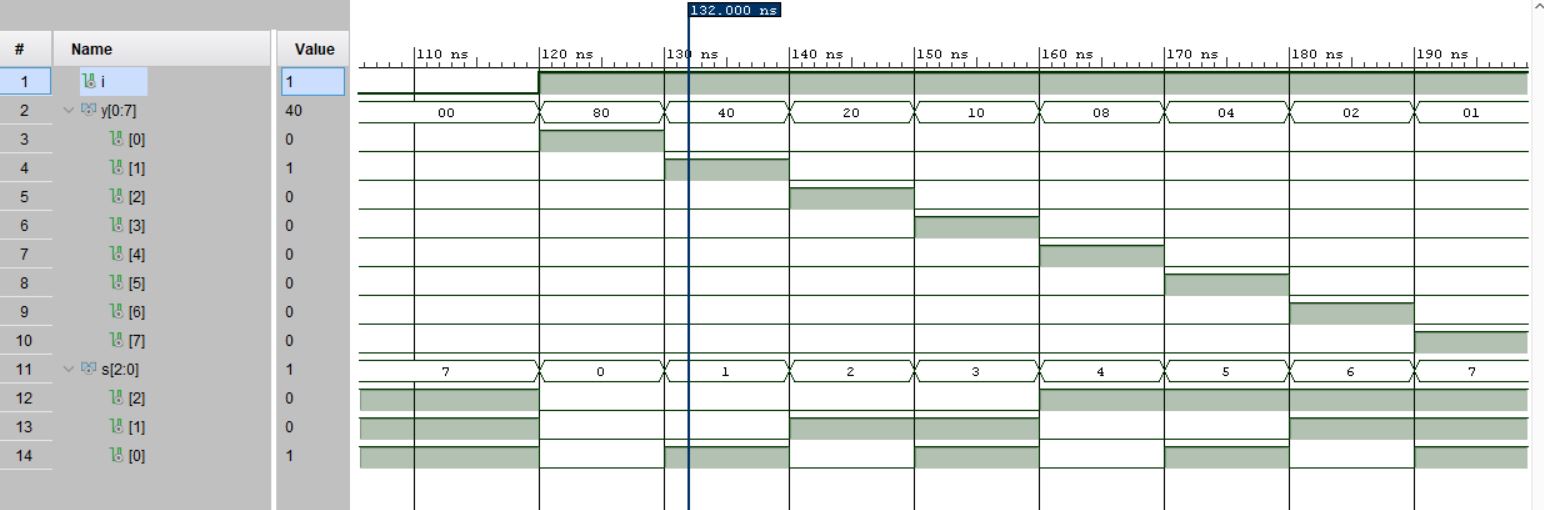
s<="111";

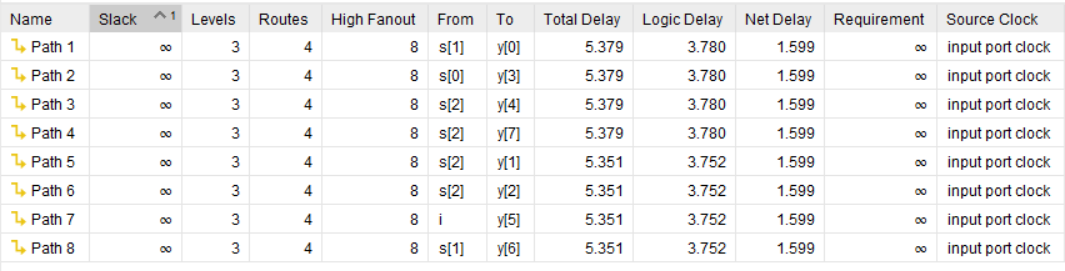
wait;

end process;

end Behavioral;

**SIMULATION WAVEFORM :**

****



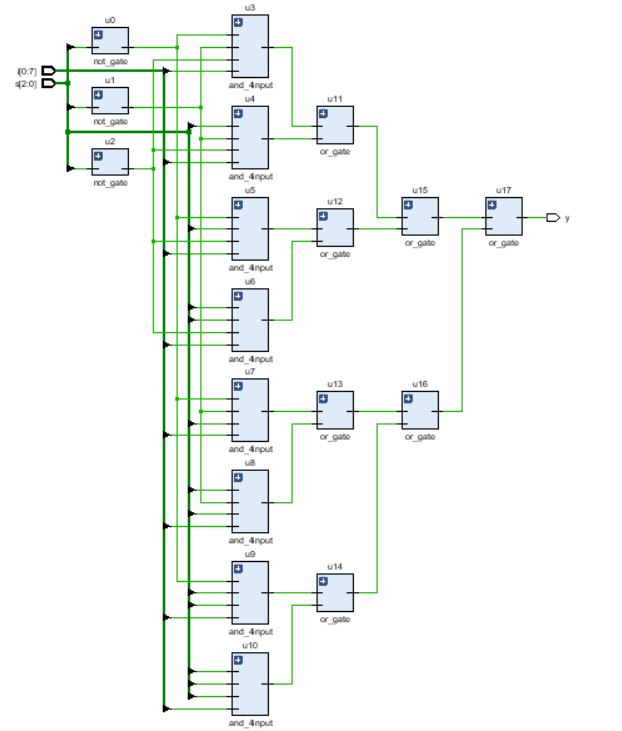
**SYNTHESIS SUMMARY:**

|  |  |  |  |
| --- | --- | --- | --- |
| **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;

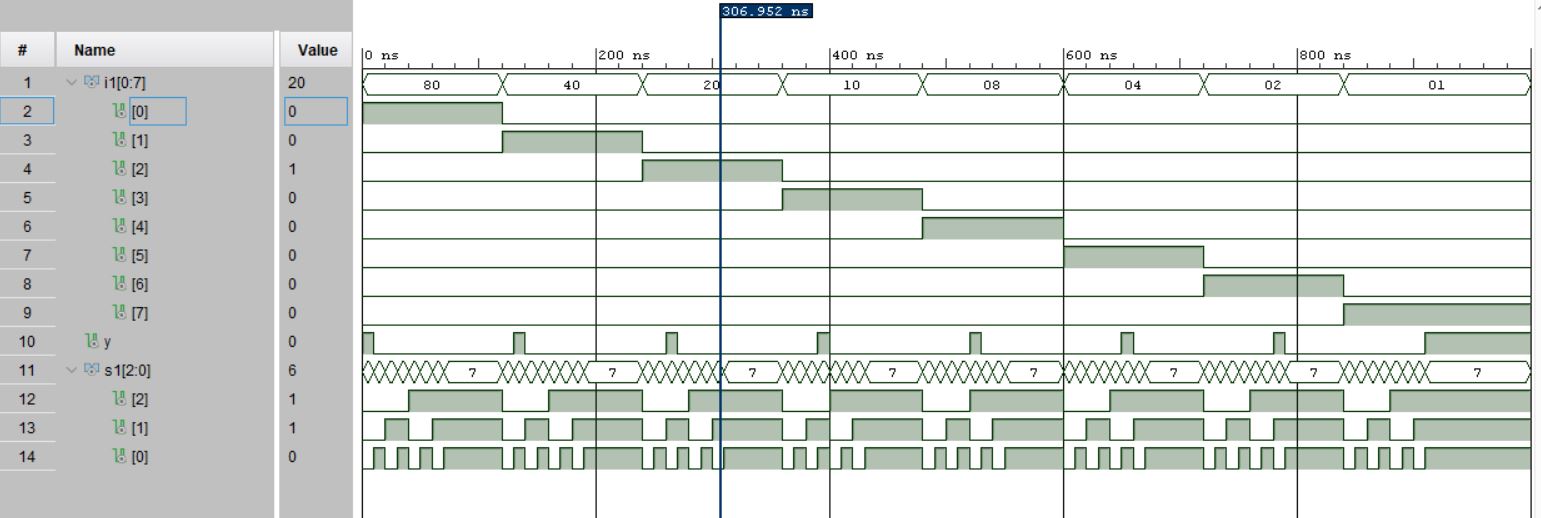
s1<="111";

wait;

end process;

end Behavioral;

**SIMULATION WAVEFORM :**

****



**SYNTHESIS SUMMARY:**

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

Maximum Combinational Delay: 5.77nSec