**Experiment / Assignment / Tutorial No. \_\_\_8\_\_\_**

**Grade: AA / AB / BB / BC / CC / CD /DD**

**Signature of the Staff In-charge with date**

|  |
| --- |
| **Batch: B3 Roll No.: 121 Experiment / assignment / tutorial No.: 8** |

|  |
| --- |
| **Title:** 4:1 Mux in VHDL |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Objective:** Design of 3 bit asynchronous counter using JK flip flop in VHDL

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected Outcome of Experiment:**

**CO2:** Use different minimization technique and solve combinational circuits, synchronous & asynchronous sequential circuits.

**CO4:** Implement digital networks using VHDL

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

* VLab Links: <http://vlabs.iitb.ac.in/vlabs-dev/labs/dldesignlab/experimentlist.html>
* R. P. Jain, “Modern Digital Electronics”, Tata McGraw Hill
* M .Morris Mano, “Digital Logic & computer Design”, PHI
* [https://wiki.engr.illinois.edu/download/attachments/84770821/08](https://wiki.engr.illinois.edu/download/attachments/84770821/08-Multiplexers.pdf?version=2&modificationDate=1285128827000)- [Multiplexers.pdf?version=2&modificationDate=128512882700](https://wiki.engr.illinois.edu/download/attachments/84770821/08-Multiplexers.pdf?version=2&modificationDate=1285128827000)0

**Pre Lab/ Prior Concepts:**

**Multiplexer:** Multiplexer is a special type of combinational circuit. It is a digital circuitwhich selects one of the n data inputs and routes it to the output. The selection of one of the n inputs is done by the select lines. To select n inputs we require m select lines, such that 2m=n. Depending on the digital code applied at the select inputs, one out of the n data sources is selected and transmitted to a single output . E is called as the strobe or enable input which is useful for cascading. It is generally on active low terminal that means it will perform the required operation when it is low. The multiplexer act like a digitally controlled single pole, multiple way switches. The output gets connected to only one input at a time. In most of the electronic system the digital data is available on more than one line. It is necessary to route the data over a single line, under such circumstances input at a time

**Types of Multiplexer:**

1. 2:1 Multiplexer
2. 4:1 Multiplexer
3. 8:1 Multiplexer
4. 16:1 Multiplexer
5. 32:1 Multiplexer

**Implementation Details of 4:1 MUX**

Code for entity:

Library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

use ieee.std\_logic\_arith.all;

entity my4to1mux is

port(

a, b, c, d : in std\_logic;

s0, s1: in std\_logic;

y: out std\_logic

);

end my4to1mux;

architecture my4to1mux\_a of my4to1mux is

begin

process (a, b, c, d, s0, s1) is

begin

if (s0 ='0' and s1 = '0') then

y <= a;

elsif (s0 ='1' and s1 = '0') then

y <= b;

elsif (s0 ='0' and s1 = '1') then

y <= c;

else

y <= d;

end if;

end process;

end my4to1mux\_a;

Code for test bench:

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

use ieee.std\_logic\_arith.all;

entity my4to1mux\_tb is

end my4to1mux\_tb;

architecture my4to1mux\_tb\_a of my4to1mux\_tb is

component my4to1mux

port(

a : in std\_logic;

b : in std\_logic;

c : in std\_logic;

d : in std\_logic;

s0 : in std\_logic;

s1 : in std\_logic;

y : out std\_logic

);

end component;

signal a : std\_logic := '0';

signal b : std\_logic := '0';

signal c : std\_logic := '0';

signal d : std\_logic := '0';

signal s0 : std\_logic := '0';

signal s1 : std\_logic := '0';

signal y : std\_logic;

begin

uut: my4to1mux port map (a => a, b => b, c => c, d => d, s0 => s0, s1 => s1, y => y );

stim\_proc: process

begin

wait for 100 ns;

a <= '1';

b <= '0';

c <= '1';

d <= '0';

s0 <= '0';

s1 <= '0';

wait for 100 ns;

s0 <= '1'; s1 <= '0';

wait for 100 ns;

s0 <= '0'; s1 <= '1';

wait for 100 ns;

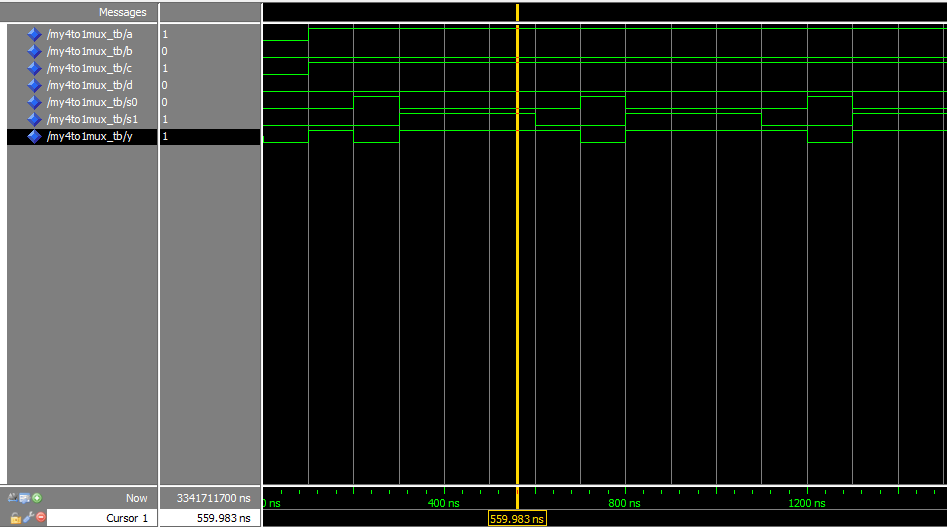
s0 <= '0'; s1 <= '1';

wait for 100 ns;

end process;

end;

Output:



**4:1 MUX using Behavioral Method:**

Code:

Entity:

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity MUX\_SOURCE is

Port ( S : in STD\_LOGIC\_VECTOR (1 downto 0);

I : in STD\_LOGIC\_VECTOR (3 downto 0);

Y : out STD\_LOGIC);

end MUX\_SOURCE;

architecture Behavioral of MUX\_SOURCE is

begin

process (S,I)

begin

if (S <= "00") then

Y <= I(0);

elsif (S <= "01") then

Y <= I(1);

elsif (S <= "10") then

Y <= I(2);

else

Y <= I(3);

end if;

end process;

end Behavioral;

Test bench:

library ieee;

use ieee.std\_logic\_1164.all;

entity mux\_tb is

end entity;

architecture tb of mux\_tb is

component MUX\_SOURCE is

Port ( S : in STD\_LOGIC\_VECTOR (1 downto 0);

I : in STD\_LOGIC\_VECTOR (3 downto 0);

Y : out STD\_LOGIC);

end component;

signal S : STD\_LOGIC\_VECTOR(1 downto 0);

signal I : STD\_LOGIC\_VECTOR(3 downto 0);

signal Y : STD\_LOGIC;

begin

uut : MUX\_SOURCE port map(

S => S,

I => I,

Y => Y);

stim : process

begin

I(0) <= '0';

I(1) <= '1';

I(2) <= '0';

I(3) <= '1';

S <= "00";wait for 10 ns;

S <= "01";wait for 10 ns;

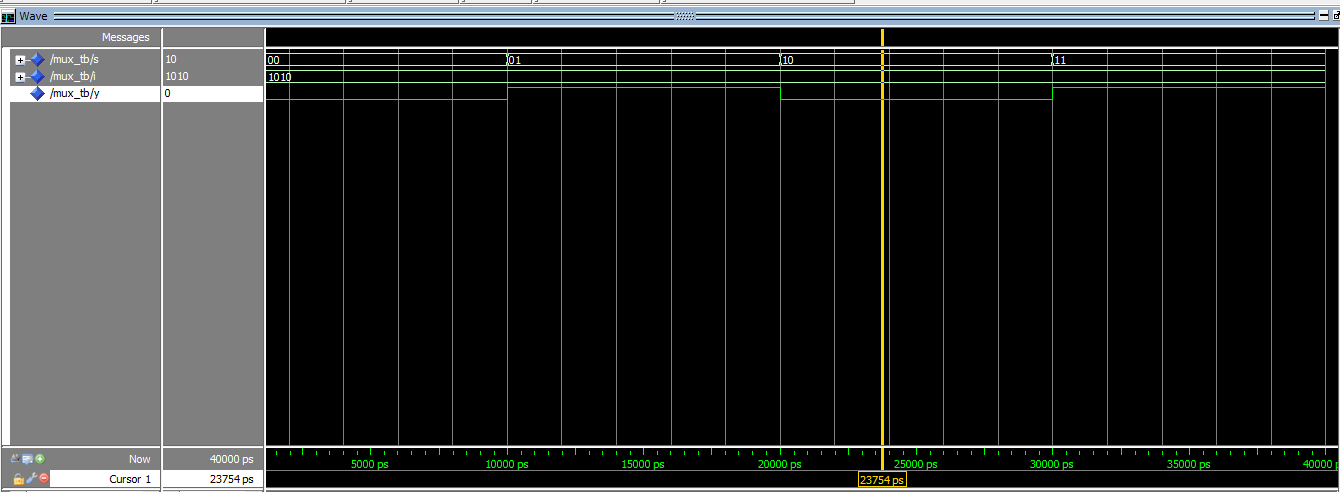
S <= "10";wait for 10 ns;

S <= "11";wait for 10 ns;

wait;

end process;

end tb;

Output:

**Conclusion:**

Thus, in this experiment, the 4:1 MUX has been implemented in VHDL using the Quartus II and Model-Sim Altera simulator softwares.

**Post Lab Descriptive Questions**

1. Application Mux?

Ans.

* + 1. Multiplexers are used in various applications where multiple data needs to be transmitted over a single line.
    2. Multiplexers are used in computer memory to maintain a huge amount of memory in computers and to reduce the number of copper lines required.
    3. In telephone networks, multiple audio signals are integrated to a single line of transmission with the help of a multiplexer.
    4. Multiplexer is used in the implementation of combinational circuits.