|  |  |  |  |
| --- | --- | --- | --- |
| **Course Name:** | **Hardware Description Language Lab (2UXL401)** | **Semester:** | **IV** |
| **Date of Performance:** | **2-2-2021** | **Batch No:** | **B2** |
| **Faculty Name:** | **Bhargavi maam** | **Roll No:** | **1912052** |
| **Faculty Sign & Date:** |  | **Grade/Marks:** |  |

**Experiment No: 2**

**Title:** Study of Multiplexer (Dataflow, Behavioral, Structural)

|  |
| --- |
| **Aim and Objective of the Experiment:** |
| Write a VHDL code for implementing a 4:1 multiplexer   1. Using when else construct ( Data Flow) 2. Using structural architecture   Write a testbench to verify your results.  Write a VHDL code for 16: 1 mux using 4:1 mux and gates. ( Structural)  To study basic types of architectures and to understand the use of concurrent statements. |

|  |
| --- |
| **COs to be achieved:** |
| **CO 1**: Use basic Concurrent and Sequential statements in VHDL and write codes for simple applications  **CO 2**: Test a VHDL code and verify the circuit model.  **CO 3**: Synthesize and Implement the designed circuits on CPLD/ FPGA. |

|  |
| --- |
| **Work to be uploaded** |
| Code for 4:1 Mux , Code for 16:1 mux  Test bench for 4:1 mux and 16:1 Mux and simulation waveform of the same.  Scanned copy of post lab questions  **DataFlow**  library ieee;  use ieee.std\_logic\_1164.all;  use ieee.std\_logic\_unsigned.all;  use ieee.std\_logic\_arith.all;    entity Mux\_Dataflow\_Vedant is  port(  i0,i1,i2,i3,s0,s1: In std\_logic;  y: Out std\_logic  );  end Mux\_Dataflow\_Vedant;    architecture Mux\_Dataflow\_Vedant\_arch of Mux\_Dataflow\_Vedant is  signal a,b:std\_logic;  begin  a<= not s1;  b<=not s0;  y<= (a and b and i0) or (a and s0 and i1) or (s1 and b and i2) or (s1 and s0 and i3) ;  end Mux\_Dataflow\_Vedant\_arch;  **Test Bench**  library ieee;  use ieee.std\_logic\_1164.all;  use ieee.std\_logic\_unsigned.all;  use ieee.std\_logic\_arith.all;    entity Mux\_Dataflow\_Vedant\_tb is  end Mux\_Dataflow\_Vedant\_tb;    architecture Mux\_Dataflow\_Vedant\_tb\_arch of Mux\_Dataflow\_Vedant\_tb is  component Mux\_Dataflow\_Vedant is  port(  i0,i1,i2,i3,s0,s1: In std\_logic;  y: Out std\_logic  );  end component;  signal i0,i1,i2,i3,s0,s1,y: std\_logic;    begin  uut: Mux\_Dataflow\_Vedant port map( i0,i1,i2,i3,s0,s1,y);    process begin  i0<='0';  i1<='1';  i2<='1';  i3<='0';    s0<='0';  s1<='0';  wait for 10ns;    s0<='0';  s1<='1';  wait for 10ns;    s0<='1';  s1<='0';  wait for 10ns;    s0<='1';  s1<='1';  wait for 10ns;  end process;  end;    Behavioral  **Entity**  library ieee;  use ieee.std\_logic\_1164.all;  use ieee.std\_logic\_unsigned.all;  use ieee.std\_logic\_arith.all;    entity Mux\_Behavioural\_Vedant is  port(  i: In std\_logic\_vector(3 downto 0);  s: In std\_logic\_vector(1 downto 0);  y: Out std\_logic  );  end Mux\_Behavioural\_Vedant;    architecture Mux\_Behavioural\_Vedant\_arch of Mux\_Behavioural\_Vedant 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 Mux\_Behavioural\_Vedant\_arch;  **Test Bench**  library ieee;  use ieee.std\_logic\_1164.all;  use ieee.std\_logic\_unsigned.all;  use ieee.std\_logic\_arith.all;  entity Mux\_Behavioural\_Vedant\_tb is  end Mux\_Behavioural\_Vedant\_tb;  architecture Mux\_Behavioural\_Vedant\_tb\_arch of Mux\_Behavioural\_Vedant\_tb is  component Mux\_Behavioural\_Vedant is  port(  i: In std\_logic\_vector(3 downto 0);  s: In std\_logic\_vector(1 downto 0);  y: Out std\_logic  );  end component;  signal i: std\_logic\_vector(3 downto 0);  signal s: std\_logic\_vector(1 downto 0);  signal y: std\_logic;    begin  uut: Mux\_Behavioural\_Vedant port map(i,s,y);    process begin  i <= "1010";    s <= "00";  wait for 20ns;    s <= "01";  wait for 20ns;    s <= "10";  wait for 20ns;    s <= "11";  wait for 20ns;    end process;  end; |

|  |
| --- |
| **Post Lab Subjective/Objective type Questions:** |
| Upload Answer of following question before coming to next laboratory.  Q1. **Analyse the following code and explain its output.**  library ieee;  use ieee.std\_logic\_1164.all;  entity xyz is  port(g1,g2,g3:in std\_logic;  sel: in std\_logic\_vector(2 downto 0);  q:out std\_logic\_vector(7 downto 0)  );  end xyz;  architecture xyz\_a of xyz is  signal q\_s: std\_logic\_vector(7 downto 0);  Begin  with sel select q\_s<=  "01111111" when "000",  "10111111" when "001",  "11011111" when "010",  "11101111" when "011",  "11110111" when "100",  "11111011" when "101",  "11111101" when "110",  "11111110" when "111",  "11111111" when others;  q<=q\_s when (g1 and not g2 and not g3)= '1'    else "11111111";    end xyz\_a\_a;      **Q.2 Write a test bench for the above code.** |

|  |
| --- |
| **Conclusion:**  We wrote a VHDL code for implementing a 4:1 multiplexer using structural and dataflow architecture  Then we wrote a VHDL code for 16: 1 mux using 4:1 mux and gates |

|  |
| --- |
| **Signature of faculty in-charge with Date:** |