|  |  |  |  |
| --- | --- | --- | --- |
| **Course Name:** | **Hardware Description Language Lab (2UXL401)** | **Semester:** | **IV** |
| **Date of Performance:** | **04/05/2021** | **Batch No:** | **B2** |
| **Faculty Name:** | Prof. Bhargavi Kaslikar | **Roll No:** | **1912052** |
| **Faculty Sign & Date:** |  | **Grade/Marks:** |  |

**Experiment No: 8**

**Title:** Use of Functions and Procedures

|  |
| --- |
| **Aim and Objective of the Experiment:** |
| Write a code for Universal Shift Register with operations as  00: Shift Left , 01: Shift Right, 10: Rotate Right, 11: Load . Write functions for all operations and use them. Test using test bench and Implement the same on CPLD Kit.  To study functions and Procedures in VHDL |

|  |
| --- |
| **COs to be achieved:** |
| **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 done** |
| Upload VHDL codes. Also upload test bench and simulation for the same.  **Entity**  **library ieee;**  **use ieee.std\_logic\_1164.all;**  **use ieee.std\_logic\_arith.all;**  **use ieee.std\_logic\_unsigned.all;**  **entity barrels is**  **port(**  **din: in std\_logic\_vector(7 downto 0);**  **shift: in std\_logic\_vector(2 downto 0);**  **con: in std\_logic\_vector(1 downto 0);**  **dout: out std\_logic\_vector(7 downto 0)**  **);**  **end barrels;**  **architecture barrels\_arch of barrels is**  **--function to convert shift input to integer**  **function conv\_int(shift: std\_logic\_vector(2 downto 0))**  **return integer is**  **variable ret\_var: integer range 0 to 7;**  **begin**  **case shift is**  **when "000" => ret\_var:=0;**  **when "001" => ret\_var:=1;**  **when "010" => ret\_var:=2;**  **when "011" => ret\_var:=3;**  **when "100" => ret\_var:=4;**  **when "101" => ret\_var:=5;**  **when "110" => ret\_var:=6;**  **when "111" => ret\_var:=7;**  **when others => NULL;**  **end case;**  **return ret\_var;**  **end conv\_int;**  **--function to rotate input data to right by n**  **function f\_ror(din: std\_logic\_vector(7 downto 0); n: integer range 0 to 7)**  **return std\_logic\_vector is**  **variable ret\_var: std\_logic\_vector(7 downto 0);**  **begin**  **ret\_var:=din;**  **for i in 0 to 7 loop**  **if(i<(n)) then**  **ret\_var:=ret\_var(0)&ret\_var(7 downto 1);**  **end if;**  **end loop;**  **return(ret\_var);**  **end f\_ror;**  **--function to rotate input data to left by n**  **function f\_rol(din: std\_logic\_vector(7 downto 0); n: integer range 0 to 7)**  **return std\_logic\_vector is**  **variable ret\_var: std\_logic\_vector(7 downto 0);**  **begin**  **ret\_var:=din;**  **for i in 0 to 7 loop**  **if(i<(n)) then**  **ret\_var:=ret\_var(6 downto 0)&ret\_var(7);**  **end if;**  **end loop;**  **return(ret\_var);**  **end f\_rol;**  **--function to shift input data to right by n**  **function f\_shr(din: std\_logic\_vector(7 downto 0); n: integer range 0 to 7)**  **return std\_logic\_vector is**  **variable ret\_var: std\_logic\_vector(7 downto 0);**  **begin**  **ret\_var:=din;**  **for i in 0 to 7 loop**  **if(i<(n)) then**  **ret\_var:='0'&ret\_var(7 downto 1);**  **end if;**  **end loop;**  **return(ret\_var);**  **end f\_shr;**  **--function to shift input data to left by n**  **function f\_shl(din: std\_logic\_vector(7 downto 0); n: integer range 0 to 7)**  **return std\_logic\_vector is**  **variable ret\_var: std\_logic\_vector(7 downto 0);**  **begin**  **ret\_var:=din;**  **for i in 0 to 7 loop**  **if(i<(n)) then**  **ret\_var:=ret\_var(6 downto 0)&'0';**  **end if;**  **end loop;**  **return(ret\_var);**  **end f\_shl;**  **signal n: integer range 0 to 7;**  **--Architecture begins**  **begin**  **n<=conv\_int(shift);**  **process(din,n,con)**  **begin**  **case con is**  **when "00" => dout <= f\_ror(din,n);**  **when "01" => dout <= f\_rol(din,n);**  **when "10" => dout <= f\_shr(din,n);**  **when "11" => dout <= f\_shl(din,n);**  **when others => NULL;**  **end case;**  **end process;**  **end barrels\_arch;**  **TB**  **library ieee;**  **use ieee.std\_logic\_1164.all;**  **use ieee.std\_logic\_arith.all;**  **use ieee.std\_logic\_unsigned.all;**  **entity barrels\_tb is**  **end entity;**  **architecture barrels\_tb\_arch of barrels\_tb is**  **component barrels is**  **port (**  **din: in std\_logic\_vector(7 downto 0);**  **shift: in std\_logic\_vector(2 downto 0);**  **con: in std\_logic\_vector(1 downto 0);**  **dout: out std\_logic\_vector(7 downto 0)**  **);**  **end component;**  **signal din: std\_logic\_vector(7 downto 0):="11111111";**  **signal shift: std\_logic\_vector(2 downto 0):="000";**  **signal con: std\_logic\_vector(1 downto 0):="00";**  **signal dout: std\_logic\_vector(7 downto 0);**  **begin**  **uut: barrels port map(din,shift,con,dout);**  **process**  **begin**  **for b in 0 to 7 loop**  **shift<=shift+1;**  **wait for 20 ns;**  **end loop;**  **wait for 20 ns;**  **for i in 0 to 2 loop**  **con<=con+1;**  **wait for 20 ns;**  **end loop;**  **end process;**  **end;** |

|  |
| --- |
| **Post Lab Subjective/Objective type Questions:** |
| Upload Answer of following question before coming to next laboratory.  **Q1 How** **a** **procedure** **is** **written** **in** **VHDL?** **Explain** **by** **writing** **a** **procedure**  f**or** **shifting** **the** **8** **bit** **data** **by** **given** **number** **of** **digits.**    **Q2** **What** **are** **the** **differences** **in** **Functions** **and** **Procedures.**   | **Functions** | **Procedures** | | --- | --- | | A function has a return type and returns a value. | A procedure does not have a return type. But it returns values using the OUT parameters. | | You cannot use a function with Data Manipulation queries. Only Select queries are allowed in functions. | You can use DML queries such as insert, update, select etc… with procedures. | | A function does not allow output parameters | A procedure allows both input and output parameters. | | You cannot manage transactions inside a function. | You can manage transactions inside a function. | | You cannot call stored procedures from a function | You can call a function from a stored procedure. | | You can call a function using a select statement. | You cannot call a procedure using select statements. |   **Q3** **Write a function in VHDL for converting integer number into binary**  **(use b bit). Show how the function can be used in architecture.**  library ieee;  use ieee.std\_logic\_unsigned.all;  use ieee.numeric\_std.all;  use ieee.std\_logic\_1164.all;  entity DistanceCal is  port( timeIn : in integer;  Distance : out std\_logic\_vector(15 downto 0));  end DistanceCal;  architecture behav of DistanceCal is  signal conv\_int : std\_logic\_vector(to\_unsigned(timeIn, conv\_int'length));  begin  process(timeIn)  begin  conv\_int <= std\_logic\_vector(to\_unsigned((timeIn\*340/2), conv\_int'length));  end process;  Distance <= conv\_int;  end behav;  **Q4 Which is correct expression for shift right of Din ( 4 bit data)?**  **a. Din <= Din( 2 downto 0) &‟0‟**  **b. Din <= Din( 3 downto 1) &‟0‟**  **c. Din <=‟0‟ &Din( 2 downto 0)**  **d. Din <= ‟0‟ & Din( 3 downto 1)**  **Ans:**  **a. Din <= Din( 2 downto 0) &‟0”** |

|  |
| --- |
| **Conclusion:**  Thus, in this experiment we have implemented a Barrel Shift register by using functions and procedures in VHDL with various operations: Shift Right, Shift Left, Rotate Right, Rotate Left and Load. |

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