| **Course Name:** | **Digital Design Laboratory** | **Semester:** | **III** |
| --- | --- | --- | --- |
| **Date of Performance:** | **26 / 10/2023** | **Batch No:** | **C1** |
| **Faculty Name:** |  | **Roll No:** | **16010122257** |
| **Faculty Sign & Date:** |  | **Grade/Marks:** | **\_\_\_/25** |

**Experiment No: 8**

**Title: 1-bit adder on VHDL**

| **Aim and Objective of the Experiment:** |
| --- |
| To implement 1-bit adder on VHDL |

| **COs to be achieved:** |
| --- |
| **CO4**: Implement digital networks using VHDL |

| **Tools used:** |
| --- |
| Quartus, ModelSim |

| **Theory:** |
| --- |
| A 1-bit adder, a fundamental component of digital circuits, performs binary addition of two 1-bit numbers. It utilizes logic gates to generate the sum and carry-out outputs. A half-adder adds two bits without considering the carry from the previous stage, while a full-adder accounts for the carry input.  Using VHDL, a hardware description language, the 1-bit adder can be designed as a combinational circuit. VHDL facilitates the creation of a structural and behavioral description of the adder. In practice, this simple unit serves as a building block for constructing larger multi-bit adders, enabling arithmetic operations in microprocessors and digital systems. |

| **Implementation Details** |
| --- |
| Half Adder Code:  library ieee;  use ieee.std\_logic\_1164.all;  entity HA is  port (a,b:in std\_logic;  s,c: out std\_logic);  end HA;  architecture HA\_arch of HA is  begin  s<= a xor b;  c<= a and b;  end HA\_arch;  Full Adder Code:  library ieee;  use ieee.std\_logic\_1164.all;  entity tb is  end tb;  architecture tb\_arch of tb is  component HA is  port (a,b: in std\_logic;  s,c: out std\_logic);  end component;  signal a,b,s,c:std\_logic;  begin  tbm:HA port map (a,b,s,c);  process  begin  a<='1';  b<='0';  wait for 5ns;  a<='0';  b<='1';  wait for 5ns;  a<='1';  b<='1';  wait for 5ns;  end process;  end tb\_arch; |
|  |
| **Post Lab Subjective/Objective type Questions:** |
| 1. How can 1-bit adder be used to implement a 4-bit adder?   A 1-bit adder can be used as a building block to create a 4-bit adder through a technique called cascading. A 1-bit adder, also known as a half adder, can add two 1-bit numbers and produce a sum and a carry output.  To build a 4-bit adder, you need to cascade four 1-bit adders together. Here's how you can do it:   * First Bit (Least Significant Bit): Add the least significant bits of the two 4-bit numbers using the first 1-bit adder. This gives you the first bit of the sum (S0) and the first carry out (C1). * Second Bit: Add the second bits of the two 4-bit numbers along with the carry out (C1) from the previous step using the second 1-bit adder. This gives you the second bit of the sum (S1) and the second carry out (C2). * Third Bit: Add the third bits of the two 4-bit numbers along with the carry out (C2) from the previous step using the third 1-bit adder. This gives you the third bit of the sum (S2) and the third carry out (C3). * Fourth Bit (Most Significant Bit): Add the most significant bits of the two 4-bit numbers along with the carry out (C3) from the previous step using the fourth 1-bit adder. This gives you the fourth bit of the sum (S3) and the final carry out (C4).   In this way, you've effectively added two 4-bit numbers using four 1-bit adders, where each adder handles one bit of the numbers and the carry from the previous addition. The output of the 4-bit adder will be the four bits of the sum (S3, S2, S1, S0), and the final carry out (C4) can be an overflow indicator if you're working with signed numbers.  This cascading approach can be extended to create n-bit adders by using n 1-bit adders together.   1. What is VHDL used for? |

| VHDL, which stands for VHSIC Hardware Description Language (VHSIC: Very High-Speed Integrated Circuit), is a programming language used for describing the behavior and structure of electronic systems. It is commonly used for the design and simulation of digital circuits and systems before they are physically implemented on hardware.  Here are some key uses of VHDL:  Digital Circuit Design: VHDL is used to design digital circuits at various levels of abstraction, including high-level behavioral descriptions, data flow descriptions, and structural descriptions. Designers can specify the functionality of electronic systems using VHDL constructs.  Simulation: VHDL allows designers to create simulation models of digital systems. These models can be simulated on a computer to verify the functionality of the design before it is manufactured. Simulation is crucial for finding and fixing design errors without the cost of physical prototypes.  Synthesis: VHDL code can be synthesized into actual hardware components. Synthesis tools transform high-level VHDL descriptions into netlists that represent the physical implementation of the design, such as gates and flip-flops. These netlists can then be used by electronic design automation (EDA) tools to create printed circuit boards (PCBs) or application-specific integrated circuits (ASICs). |
| --- |
| **Conclusion:** |
| Thus,in this experiment,we successfully realized a 1-bit adder using VHDL and demonstrated the creation of digital networks with VHDL. This practical exercise enhanced understanding of digital logic design and VHDL's role in designing complex digital circuits. |

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