Course Name:	**Digital Design Laboratory**	**Semester:**	**III**
Date of Performance:	**9 /10 /2023**	**Batch No:**	**C-2**
Faculty Name:		**Roll No:**	**16010122267**
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

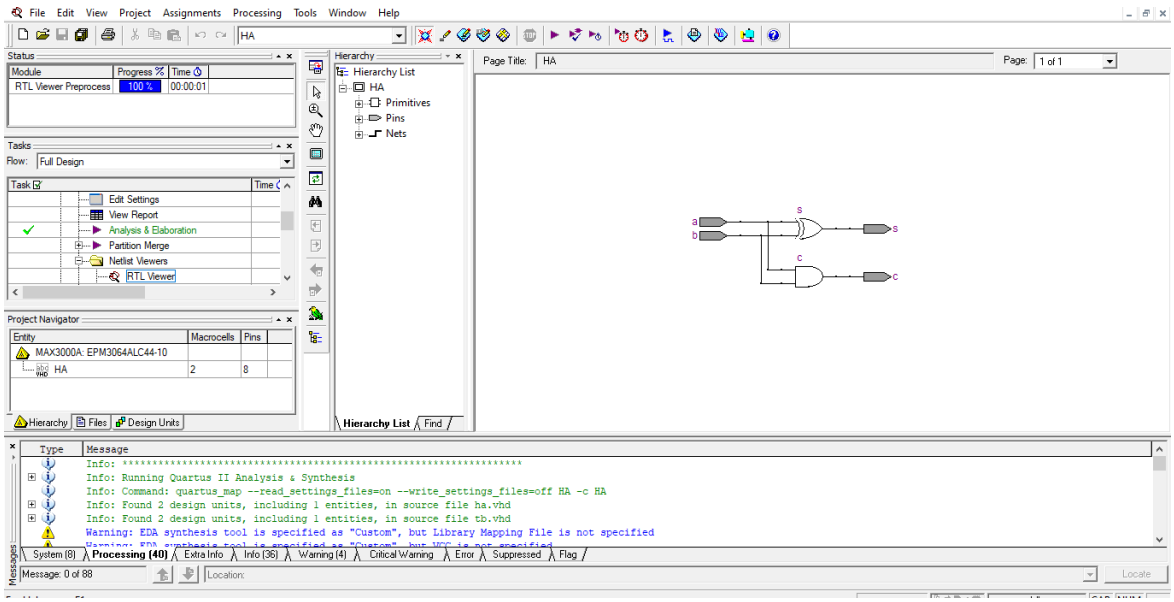
Tools used:

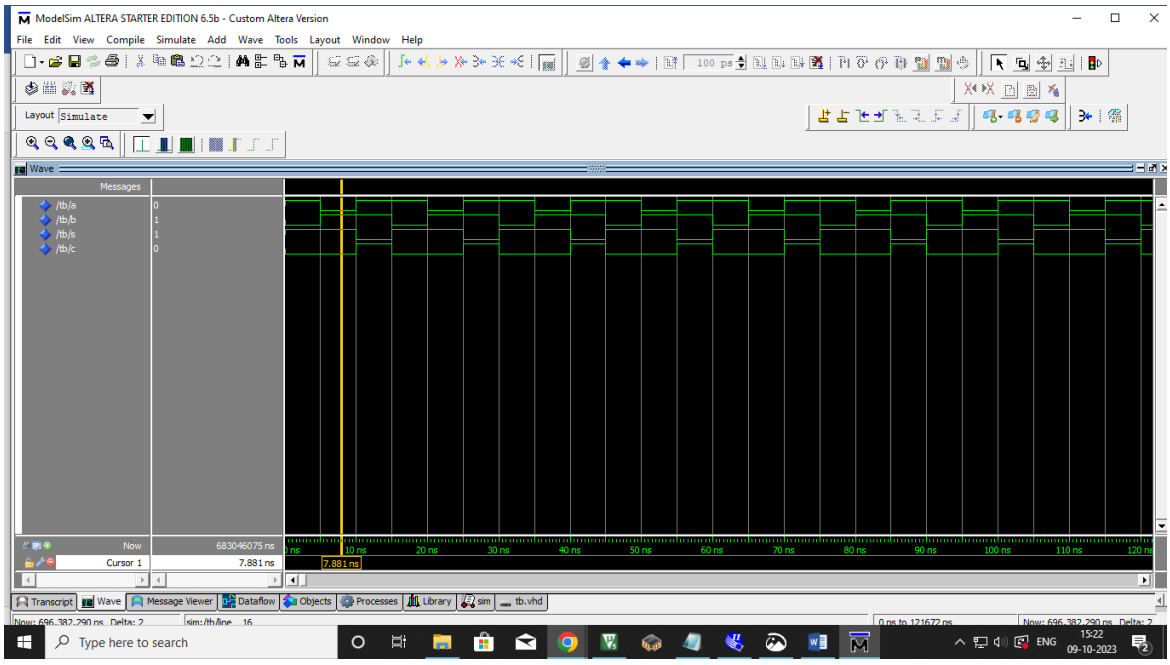
Quartus, ModelSim

Theory:

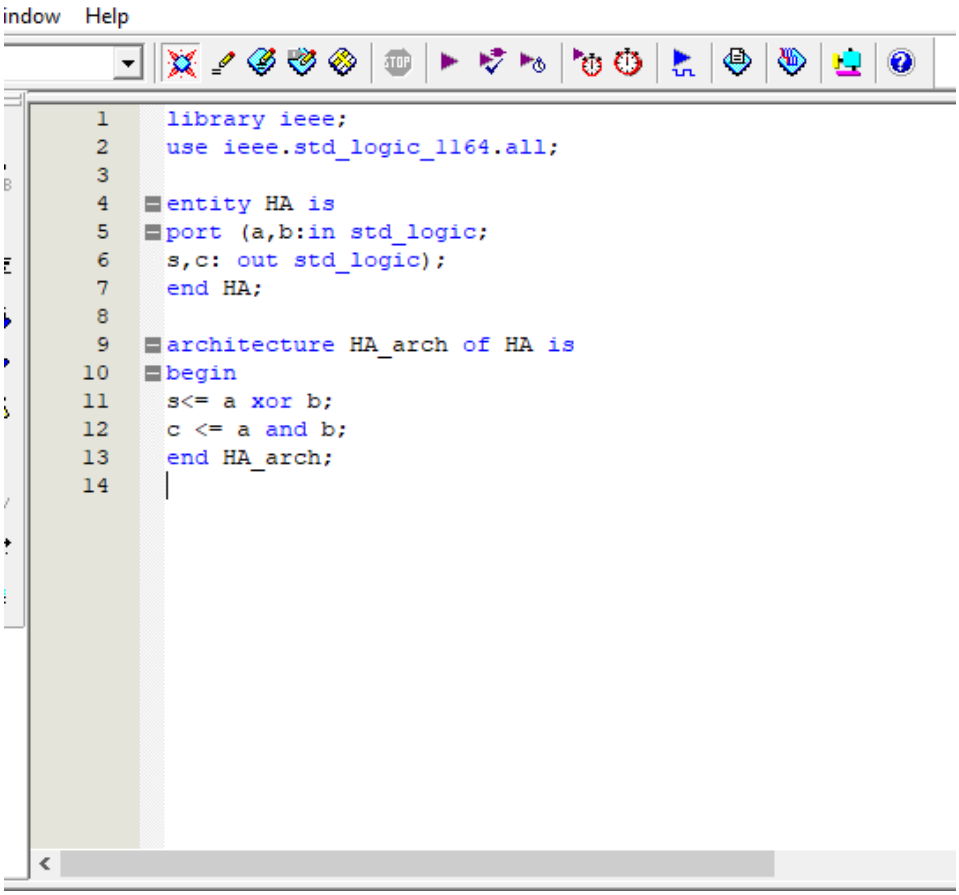
A 1-bit adder, a fundamental component of digital circuits, performs binary addition of two 1-bit numbers. It utilizes logic gates to generate the sum and carry-out outputs. A half-adder adds two bits without considering the carry from the previous stage, while a full-adder accounts for the carry input. 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


File Edit View Project Assignments Processing Tools Window Help
Status
Module
Progress
Time
RTL Viewer Preprocess
100 %
00:00:01
Tasks
Flow: Full Design
Task
Edit Settings
View Report
Analysis & Elaboration
Partition Merge
Netlist Viewers
RTL Viewer
Project Navigator
Entity
Macrocells
Pins
MAX3000A: EPM3064ALC44-10
HA
2
8
Hierarchy
Hierarchy List
HA
Primitives
Pins
Nets
Page Title: HA
Page: 1 of 1
a
b
s
c
Type
Message
Info: Running Quartus II Analysis & Synthesis
Info: Command: quartus_map --read_settings_files=on --write_settings_files=off HA -c HA
Info: Found 2 design units, including 1 entities, in source file ha.vhd
Info: Found 2 design units, including 1 entities, in source file tb.vhd
Warning: EDA synthesis tool is specified as "Custom", but Library Mapping File is not specified
System (8)
Processing (40)
Extra Info
Info (36)
Warning (4)
Critical Warning
Error
Suppressed
Flag
Message: 0 of 88
Location:
Locate


ModelSim ALTERA STARTER EDITION 6.5b - Custom Altera Version
File Edit View Compile Simulate Add Wave Tools Layout Window Help
Layout Simulate
Wave
Messages
/tb/a
/tb/b
/tb/s
/tb/c
Now
683046075 ns
Cursor 1
7.881 ns
10 ns
20 ns
30 ns
40 ns
50 ns
60 ns
70 ns
80 ns
90 ns
100 ns
110 ns
120 ns
Transcript
Wave
Message Viewer
Dataflow
Objects
Processes
Library
sim
tb.vhd
Type here to search
ENG
15:22
09-10-2023

Code:


indow Help
1 library ieee;
2 use ieee.std_logic_1164.all;
3
4 entity HA is
5 port (a,b:in std_logic;
6 s,c: out std_logic);
7 end HA;
8
9 architecture HA_arch of HA is
10 begin
11 s<= a xor b;
12 c <= a and b;
13 end HA_arch;
14

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  entity tb is
5  end tb;
6
7  architecture tb_arch of tb is
8  component HA is
9      port(a,b:in std_logic;
10         s,c: out std_logic);
11 end component;
12 signal a,b,s,c:std_logic;
13 begin
14     tbm:HA port map(a,b,s,c);
15 process
16     begin
17     a<='1';
18     b<='0';
19     wait for 5ns;
20
21     a<='0';
22     b<='1';
23     wait for 5ns;
24
25     a<='1';
26     b<='1';
27     wait for 5ns;
28 end process;
29 end tb_arch;
```

Post Lab Subjective/Objective type Questions:

1. How can 1-bit adder be used to implement a 4-bit adder?

Ans:

A 1-bit adder, also known as a half-adder, can be used as a building block to create a 4-bit adder. To build a 4-bit adder, you would cascade four 1-bit adders together, with each 1-bit adder handling one bit of the input and output

1. **Define the Inputs and Outputs:** A 4-bit adder takes two 4-bit inputs, A and B, and produces a 4-bit sum (S) and a carry-out (Cout).

2**. Cascading Half-Adders:** To add two 4-bit numbers, you start by adding the least significant bits (LSBs) first. So, you connect the LSBs of A and B to the inputs of the first half-adder, resulting in a 1-bit sum (S0) and a carry-out (Cout0). The sum S0 is the LSB of the output, and Cout0 is then carried to the next stage.

3. **Repeat the Process:** Connect the next bit (bit 1) of A and B to the inputs of the second half-adder. Additionally, connect Cout0 from the first half-adder to one of the inputs of the second half-adder. This simulates adding bit 1 along with the carry-out from the previous stage. You get another 1-bit sum (S1) and carry-out (Cout1).

4. **Continue Cascading:** Continue this process for bits 2 and 3, connecting them to the inputs of the third and fourth half-adders and taking carry-outs from the previous stages.

This will give you S2, S3, Cout2, and Cout3.
5. **Final Output:** The final 4-bit sum (S) is composed of S3, S2, S1, and S0 from the individual half-adders, while the carry-out Cout3 represents the carry-out of the entire 4-bit addition.

2. What is VHDL used for?
Ans:
VHDL serves a dual purpose in the realm of electronic design, as it can be employed for both hardware design and the generation of test entities to assess the hardware's functionality. It is utilized as a preferred input format by numerous Electronic Design Automation (EDA) tools, encompassing synthesis solutions like Quartus® Prime Integrated Synthesis, simulation software, and formal verification tools.

Conclusion:

Through this experiment, we successfully implemented 1 bit adder on VHDL.

Signature of faculty in-charge with Date: