

**JK Lakshmipat University**

*Institute of Engineering & Technology*

**LAB1 (ASSIGNMENT SET -1)**

*CS1134: Computer Organisation and Architecture*

**Submitted by**  
Name: Siddhi Nyati  
Roll No: 2022btech101

**Faculty Supervisor:**

Dr. Pranab Roy

Dr. Ajai Jain

Activity no: 1

1.2: Full Adder using half adder

**Question 2:** Design a VHDL Model of a full adder using half adder components.

**Theory:**

To construct a Full Adder, we first need to understand the basic logic gates involved, which include AND gates, OR gates, and XOR gates.

***AND Gate:***

***A yellow and black electrical circuit

Description automatically generated with medium confidence***

* **Truth Table:**

|  |  |  |
| --- | --- | --- |
| ***A (Input)*** | ***B (Input)*** | ***Q (Output) = A \cdot B*** |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

* **Boolean Expression:** Q = A . B

***OR Gate:***

A yellow and black arrow with black text

Description automatically generated

* **Truth Table:**

|  |  |  |
| --- | --- | --- |
| ***A (Input)*** | ***B (Input)*** | ***Q (Output) = A + B*** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

* **Boolean Expression:** Q = A + B

**Half Adder**

A Half Adder is a basic combinational circuit that adds two single-bit binary digits. It consists of an XOR gate and an AND gate, providing outputs for the sum and carry.

**Boolean Expressions:**

* **Sum:** Sum=A⊕B
* **Carry:** Carry=A⋅B

**Truth Table for Half Adder:**

|  |  |  |  |
| --- | --- | --- | --- |
| ***A (Input)*** | ***B (Input)*** | ***SUM (Output)*** | ***CARRY (Output)*** |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

**Full Adder**

The Full Adder combines two binary inputs and an additional carry input to produce a sum and a new carry output. It consists of two Half Adders and one OR gate.

**Boolean Expressions:**

* **Sum:** Sum = (A ⊕ B ) ⊕ C
* **Carry:** Carry = (A . B) + ((A⊕B) . C)

**Truth Table for Full Adder:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***A (Input)*** | ***B (Input)*** | ***C (Input)*** | ***Sum (Output)*** | ***Carry (Output)*** |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

**Constructing Full Adder with Half Adders**

To implement a Full Adder using Half Adders, we follow these steps:

1. First Half Adder (H1) takes inputs A and B:

* Output: sum1= A ⊕ B and carry1 = A . B

1. Second Half Adder (H2) takes inputs sum1 and C:

* Output: sum2 = sum1 ⊕ C and carry2 = sum1 . C

1. OR Gate combines carry1 and carry2 to produce the final carry output.
2. The final outputs are:

* **Sum:** sum2
* **Carry**: carry1 + carry2

A diagram of a block diagram

Description automatically generated

**VHDL Modeling**

VHDL, or Very High-Speed Integrated Circuit Hardware Description Language, is a crucial tool for designing hardware systems. It enables the modeling of digital circuits using different styles, primarily Dataflow, Behavioral, and Structural modeling.

In this instance, we utilize Structural Modeling to create the Full Adder from Half Adders. This approach promotes code reuse by allowing the integration of smaller components into larger designs.

**VHDL Code for Full Adder Using Half Adders**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

-- Half Adder entity

entity HalfAdder is

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

Sum : out STD\_LOGIC;

Carry : out STD\_LOGIC);

end HalfAdder;

architecture Behavioral of HalfAdder is

begin

Sum <= A XOR B;

Carry <= A AND B;

end Behavioral;

-- Full Adder entity

entity FullAdder is

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

C : in STD\_LOGIC;

Sum : out STD\_LOGIC;

Carry : out STD\_LOGIC);

end FullAdder;

architecture Structural of FullAdder is

signal sum1, carry1, carry2 : STD\_LOGIC;

component HalfAdder is

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

Sum : out STD\_LOGIC;

Carry : out STD\_LOGIC);

end component;

begin

-- First Half Adder

HA1: HalfAdder port map (A, B, sum1, carry1);

-- Second Half Adder

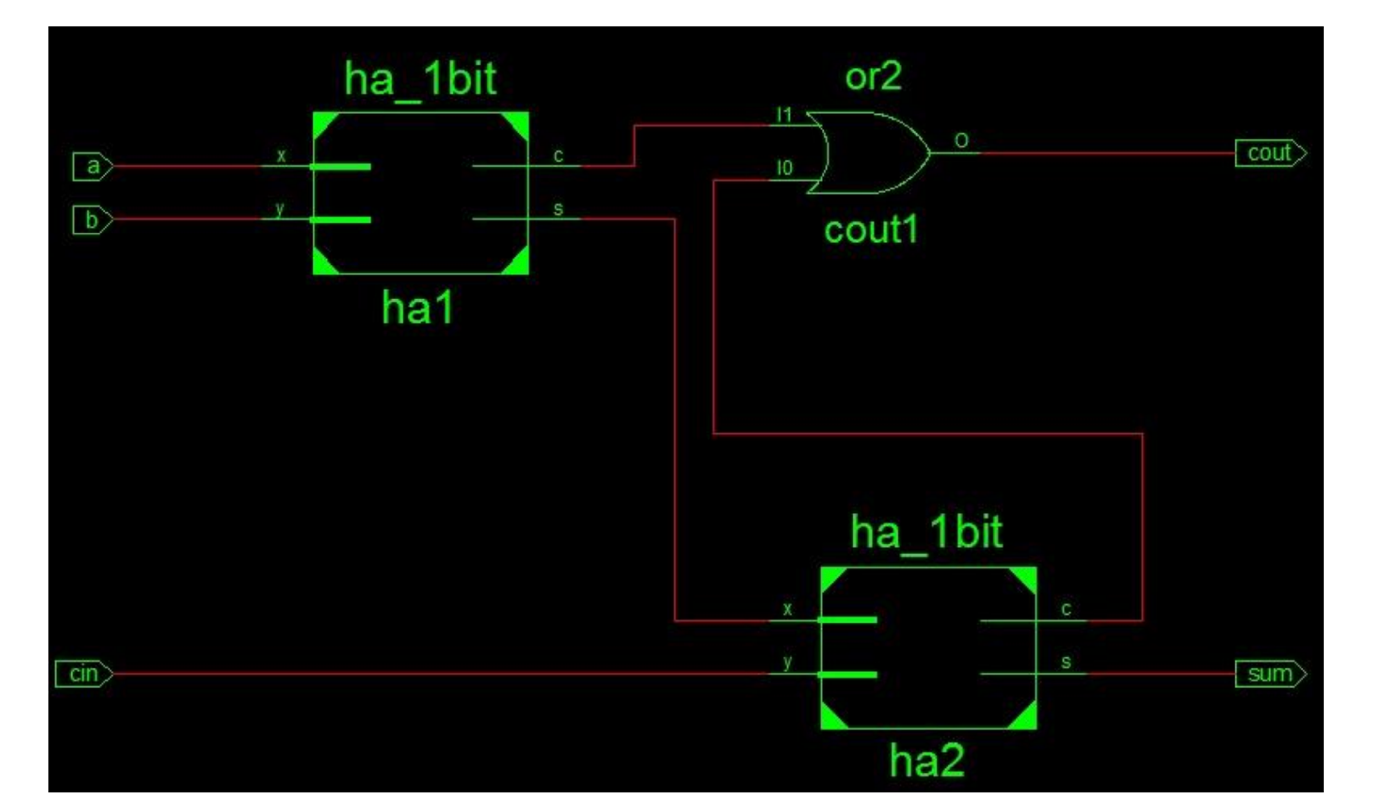
HA2: HalfAdder port map (sum1, C, Sum, carry2);

-- Final Carry

Carry <= carry1 OR carry2;

end Structural;

**SCHEMATIC DIAGRAM:**

****