Implementation of FULL & HALF ADDER using VHDL

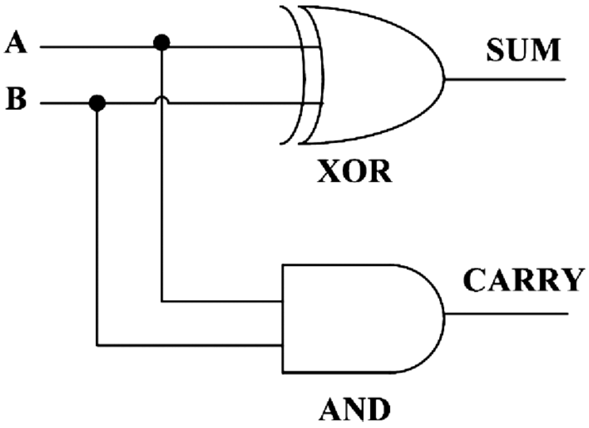
▪ OBJECTIVE:

To implement **Half & Full Adder** using VHDL and verify its waveform in Xilinx.

▪ INTRODUCTION:

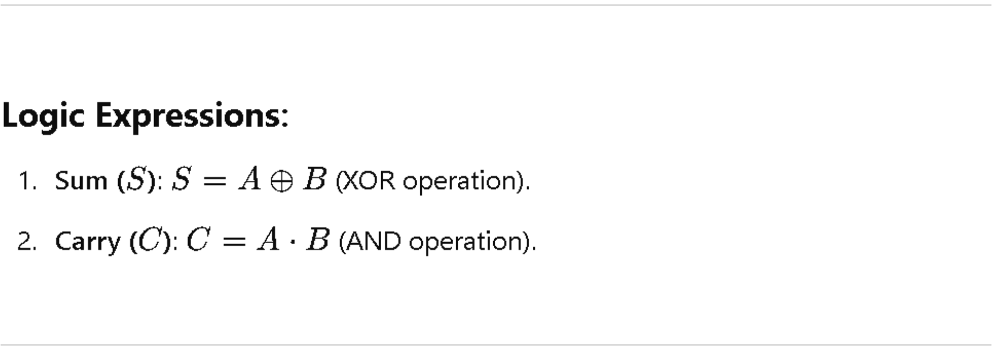
1. Half Adder:

A **Half Adder** is a basic digital circuit used for the addition of two single-bit binary numbers. It is one of the simplest types of combinational logic circuits, with two inputs and two outputs.



Inputs and Outputs:

1. **Inputs**:
   - A: First binary digit.
   - B: Second binary digit.
2. **Outputs**:
   - **Sum (S)**: Represents the sum of A and B.
   - **Carry (C)**: Represents the carry generated during the addition.



Logic Expressions:

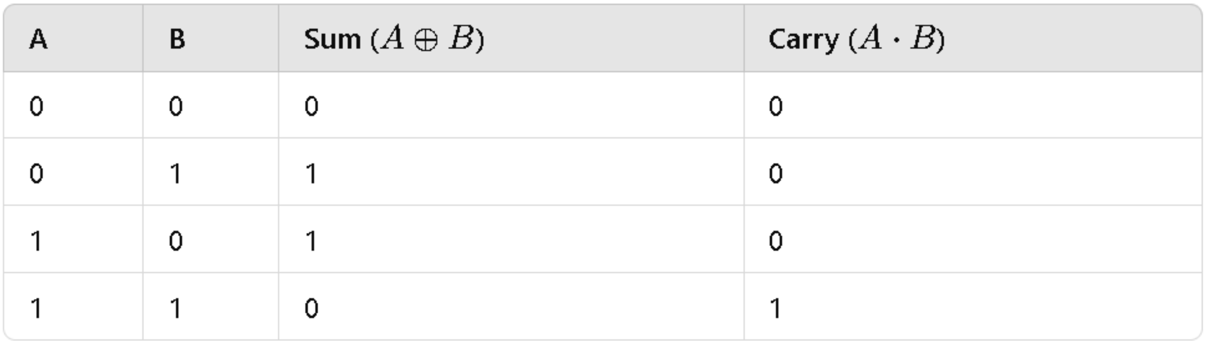
1. **Sum (S)**: $S = A \oplus B$ (XOR operation).
2. **Carry (C)**: $C = A \cdot B$ (AND operation).

Key Applications:

1. **Binary Addition**: Adds two binary digits (A and B).
2. **Building Complex Adders**: Used in full adders and multi-bit adders.
3. **Basic Arithmetic**: Used in simple calculators and basic digital systems.
4. **ALU Design**: Forms part of Arithmetic Logic Units in processors.
5. **Error Detection**: Used in parity checkers and error detection systems.
6. **Sequential Logic**: Used in circuits like memory and shift registers.

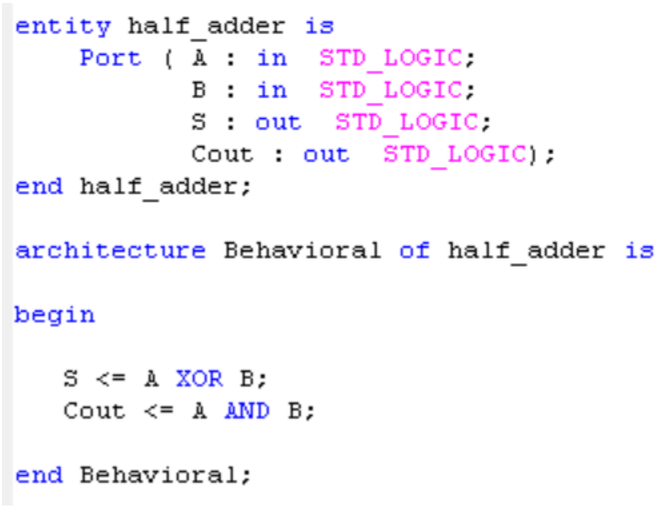
It is a fundamental building block for more advanced digital arithmetic circuits.

Truth Table:



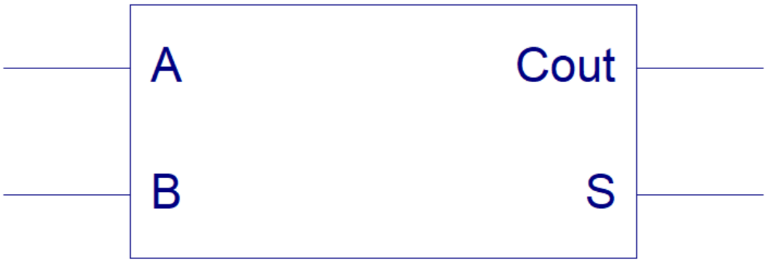
A	B	Sum ($A \oplus B$)	Carry ($A \cdot B$)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

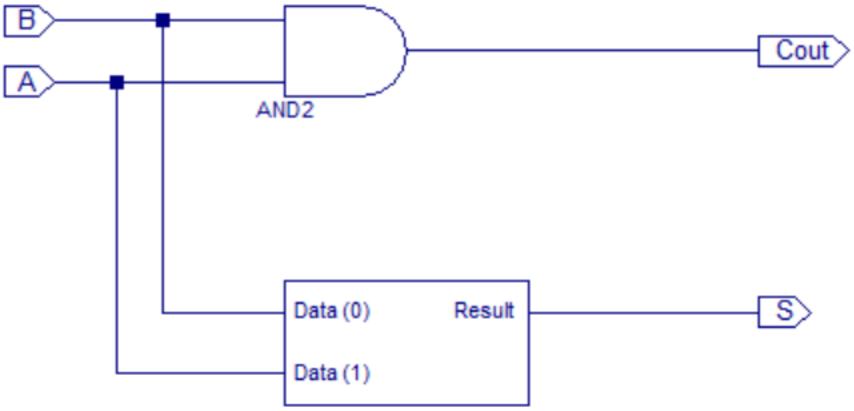
- **Source Code:**



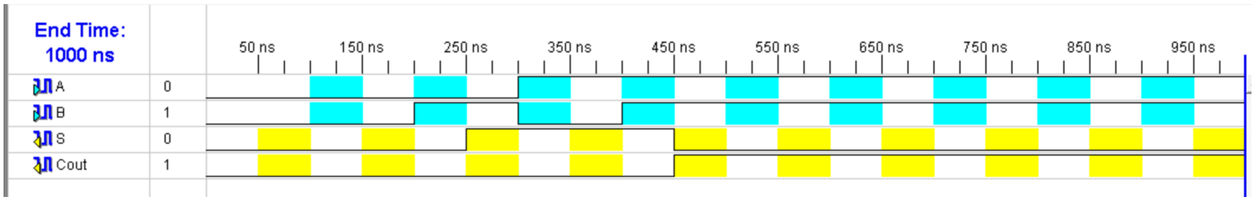
```
entity half_adder is
    Port ( A : in  STD_LOGIC;
           B : in  STD_LOGIC;
           S : out  STD_LOGIC;
           Cout : out  STD_LOGIC);
end half_adder;

architecture Behavioral of half_adder is

begin

   S <= A XOR B;
   Cout <= A AND B;

end Behavioral;
```

- **RTL (Register Transfer Level) Schematic**:



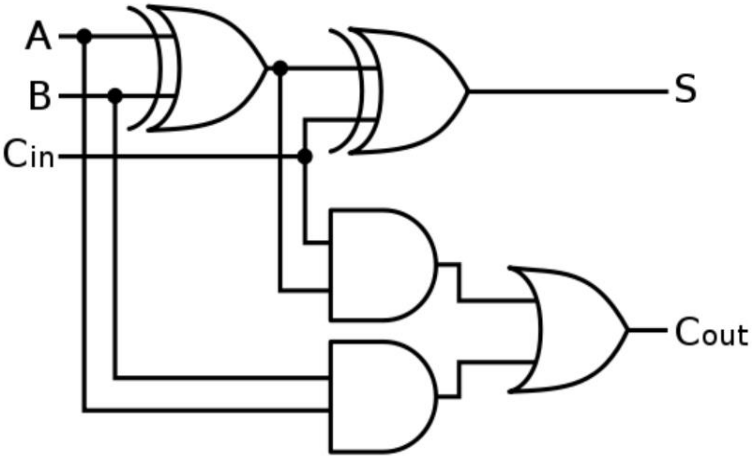


- **Test Bench Waveform:**



2. Full Adder:

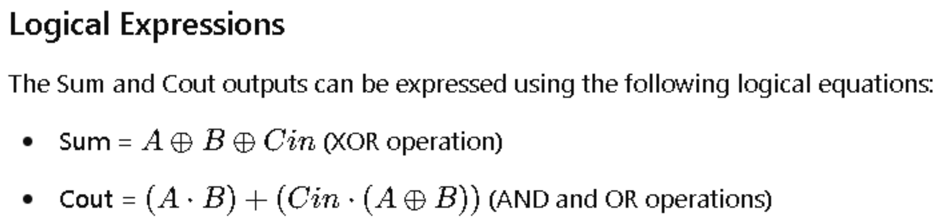
A **Full Adder** is a fundamental component in digital circuit design used to perform the addition of three binary bits. It calculates the sum and carry outputs, making it essential for arithmetic operations like binary number addition.



Structure of a Full Adder

A full adder has three inputs and two outputs:

1. **Inputs**:
   - **A**: First input bit
   - **B**: Second input bit
   - **Cin**: Carry-in bit from the previous stage
2. **Outputs**:
   - **Sum**: Result of the binary addition
   - **Cout**: Carry-out bit to the next stage



Logical Expressions

The Sum and Cout outputs can be expressed using the following logical equations:

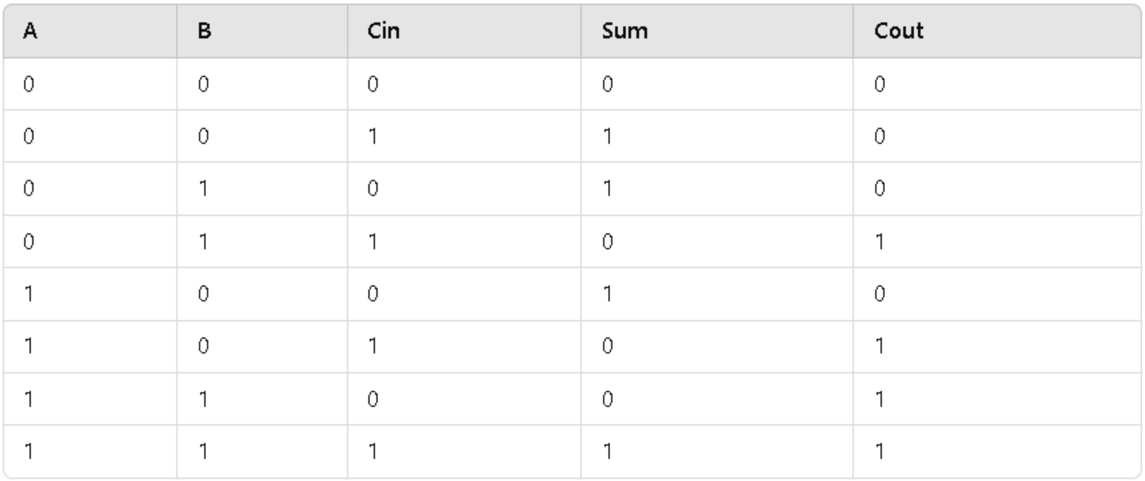
- Sum = $A \oplus B \oplus Cin$ (XOR operation)
- Cout = $(A \cdot B) + (Cin \cdot (A \oplus B))$ (AND and OR operations)

Applications

Full adders are widely used in:

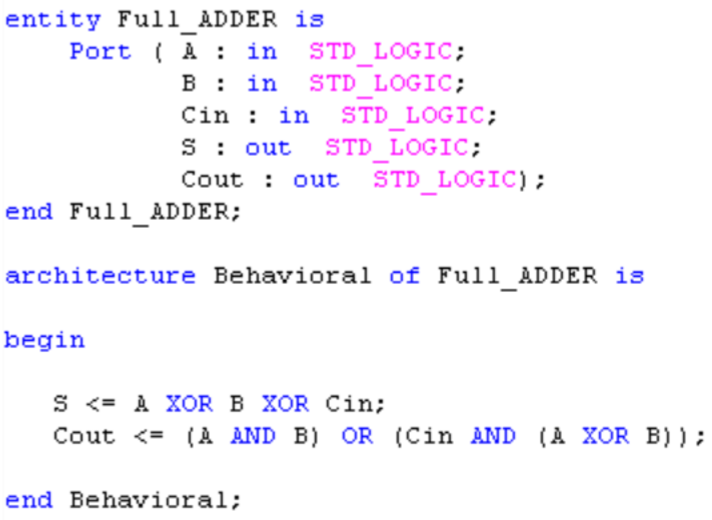
- Arithmetic and logic units (ALUs)
- Digital adders for processors
- Cascading to create multi-bit binary adders like ripple-carry adders.

Truth Table:



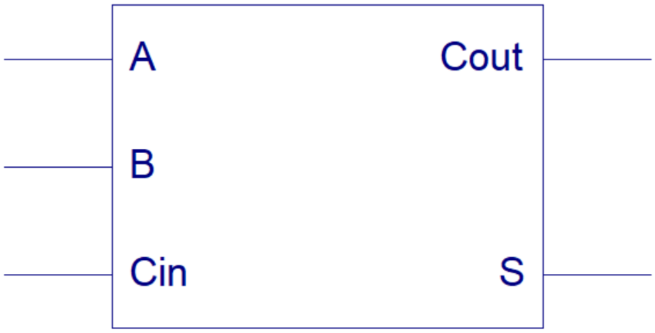
A	B	Cin	Sum	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

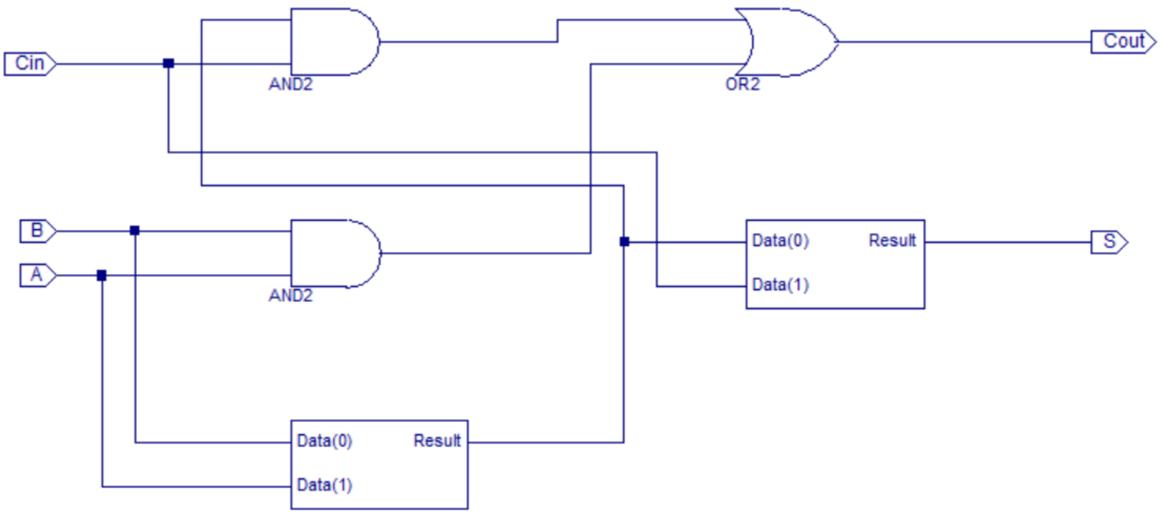
- ## Source Code:



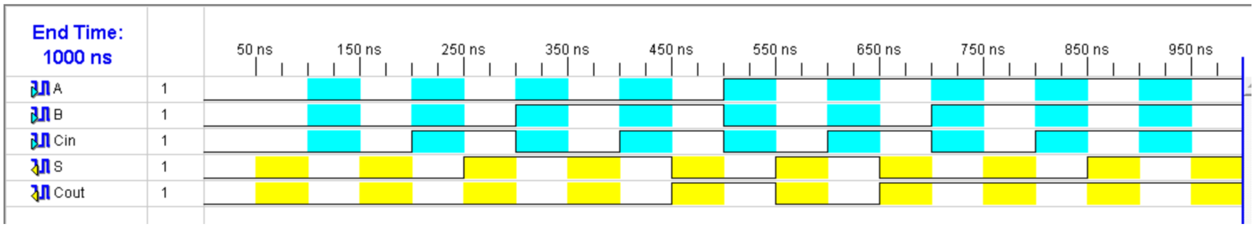
```
entity Full_ADDER is
    Port ( A : in  STD_LOGIC;
           B : in  STD_LOGIC;
           Cin : in  STD_LOGIC;
           S : out  STD_LOGIC;
           Cout : out  STD_LOGIC);
end Full_ADDER;

architecture Behavioral of Full_ADDER is

begin

   S <= A XOR B XOR Cin;
   Cout <= (A AND B) OR (Cin AND (A XOR B));

end Behavioral;
```

- **RTL (Register Transfer Level) Schematic**:





- **Test Bench Waveform:**



▪ Discussion:

In this lab, we implemented in **VHDL** using **Xilinx FPGA** tools, the **half** and **full adders**. Two bits, the half adder adds, providing Carry and Sum outputs. Three bits are added by the full adder (A, B, and Cin), and Sum and Carry-out outputs are given. Basic logic gates like (AND, XOR, OR) were used to describe both circuits in VHDL, and in **Vivado** or **ISE** design tools, we simulated them.

▪ Conclusion:

The correct functionality of both adders was confirmed after implementation and simulation. Key concepts in digital logic design, FPGA programming, and the practical use of VHDL for creating simple but important components in embedded systems, this lab helped reinforce. Insights into binary arithmetic circuits and FPGA implementation, this hands-on experience provided valuable into embedded systems design.