# Carry look ahed adder

Project Report

Submitted by

**P.Abhilash Roll no:123EC0053**

Department of Electronics and Communication Engineering

IIITDM Kurnool

Under the guidance of

**Dr.Ranga babu**

## Contents

## 1 Introduction

A Carry Look-Ahead Adder (CLA) is a type of digital adder used to perform fast binary addition by reducing the carry propagation delay found in traditional ripple-carry adders. In conventional adders, each bit must wait for the carry output from the previous stage before producing its own result, which slows down the operation as the number of bits increases.

## 2 Design Flow

**3** 1. RTL Design using Verilog HDL 2. Functional Verification using Vivado Simulator 3. FPGA Synthesis and Implementation using Zynq-7000 Board

4. Bitstream Generation and Hardware Testing on FPGA

## RTL Design (Source Code)

Below is the Verilog RTL for a 4-bit Carry Save Adder. Replace or parameterize width as needed.

```
module cla_4bit (
    input  [3:0] A, B,   // 4-bit inputs
input        Cin,    // Carry input
output [3:0] Sum,    // 4-bit Sum
output
    output       Cout    // Carry output
);
    wire [3:0] G, P; // Generate and
Propagate signals
    wire [4:0] C;    // Carry signals

    assign C[0] = Cin;

    // Generate and Propagate
assign G = A & B;     assign P
= A ^ B;

    // Carry Look-Ahead Logic
assign C[1] = G[0] | (P[0] & C[0]);
assign C[2] = G[1] | (P[1] & C[1]);
assign C[3] = G[2] | (P[2] & C[2]);
assign C[4] = G[3] | (P[3] & C[3]);

    // Sum and Carry Out
assign Sum  = P ^ C[3:0];
assign Cout = C[4];

endmodule
```

Listing 1: Verilog RTL — 4 bit carry look adder

## Testbench:

A simple testbench used for functional verification

1

```
tb_cla_4bit.v

`timescale 1ns / 1ps

module tb_cla_4bit;

   reg [3:0] A, B;    reg Cin;    wire [3:0] Sum;
   wire Cout;

   // Instantiate the
DUT (Device Under Test)    cla_4bit uut (
      .A(A),
      .B(B),
      .Cin(Cin),
      .Sum(Sum),
      .Cout(Cout)
   );

   initial begin
      // Display header
      $display("Time | A
| B | Cin | Sum | Cout");
      $monitor("%4t |
%b | %b | %b | %b | %b",
$time, A, B, Cin, Sum, Cout);

      // Test vectors
      A=4'b0000;
B=4'b0000; Cin=0; #10;
```

```
                A=4'b0101;
      B=4'b0011; Cin=0; #10;
           A=4'b1111;
      B=4'b0001; Cin=0; #10;
           A=4'b1010;
      B=4'b0101; Cin=1; #10;
           A=4'b1111;
      B=4'b1111; Cin=1; #10;

                $finish;
             end

           endmodule
```

**Explanation: In a normal ripple-carry adder, each bit position must wait for the carry from its previous bit before producing its own carry and sum. This makes the addition slower as the number of bits increase.**

## List2: test bech for cla

## 4 Schematic (Cell-Level)

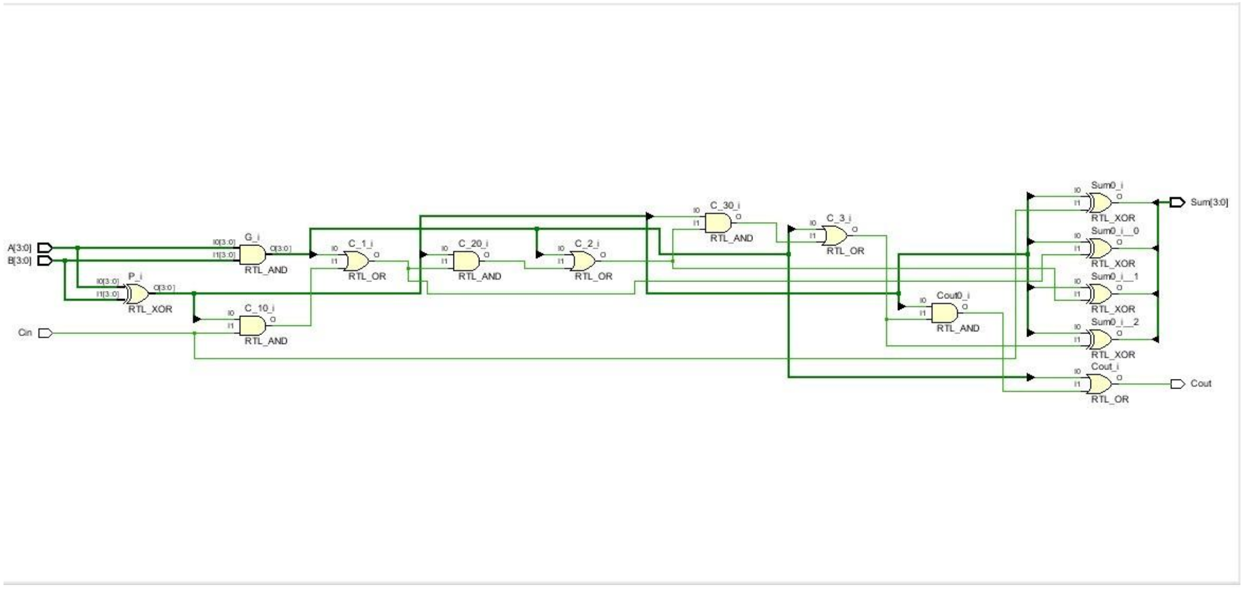
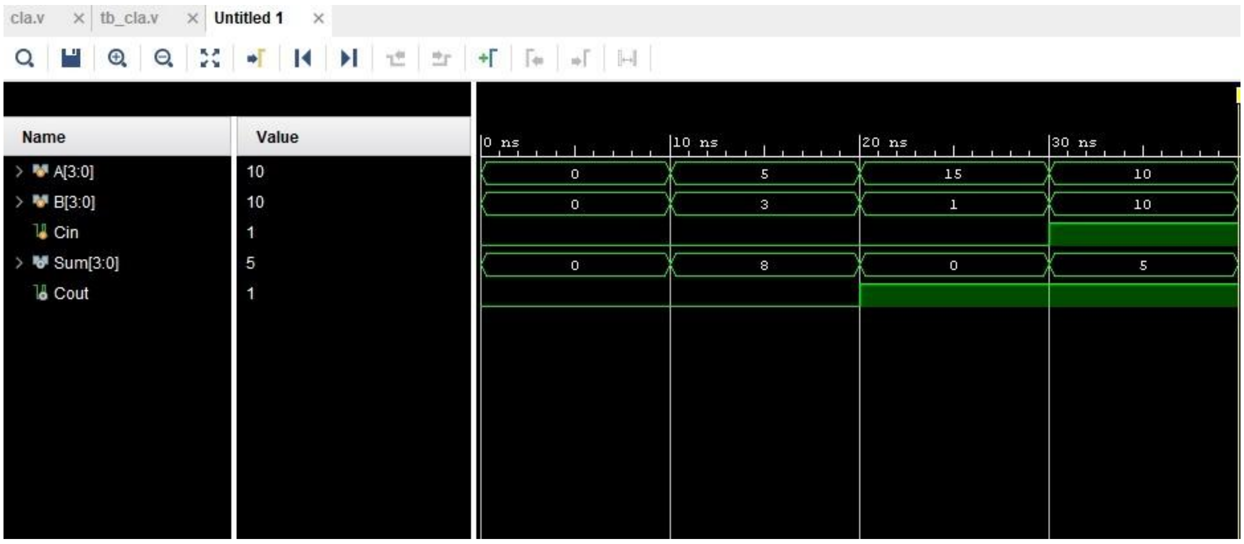


Figure 1: Schematic view (cell-level) of the 4-bit carry Adder after synthesis.

## 5 Functional Simulation Waveforms



| Name | Value | 0 ns | 10 ns | 20 ns | 30 ns |
|---|---|---|---|---|---|
| A[3:0] | 10 | 0 | 5 | 15 | 10 |
| B[3:0] | 10 | 0 | 3 | 1 | 10 |
| Cin | 1 | | | | |
| Sum[3:0] | 5 | 0 | 8 | 0 | 5 |
| Cout | 1 | | | | |

Figure 2: Waveform from SimVision demonstrating correct CLA operation for test vectors.

## 6 Result observation

The 4-bit Carry Look-Ahead Adder was successfully implemented on the Zynq FPGA board. Input switches (SW0–SW8) control the operands and carry input, while the output is displayed on LEDs (LED0–LED4). The design synthesized and implemented without errors, and the output matched the theoretical results

## 7. Conclusion

This project demonstrated the FPGA-based implementation of a 4-bit Carry Look-Ahead Adder on the ZedBoard. The design achieved low delay and correct functionality, showcasing the benefits of carry look-ahead logic in high-speed arithmetic circuits.