

# Carry look ahead adder

Project Report

Submitted by

**P.Abhilash**

**praneesh**

**Roll no:123EC0053 ,123ec0043**

Department of Electronics and Communication Engineering

IIITDM Kurnool

Under the guidance of

**Dr.Ranga babu**

## Contents

<b>1 Introduction</b>	<b>1</b>
<b>2 Design Flow</b>	<b>1</b>
<b>3 RTL Design (Source Code)</b>	<b>1</b>
<b>4 Testbench</b>	<b>3</b>
<b>8 Schematic (Cell-Level)</b>	<b>5</b>
<b>9 Functional Simulation Waveforms</b>	<b>5</b>
<b>14 Result</b>	<b>6</b>

# 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

### Design Of Carry Look Ahead Adder:

Carry look ahead adder: The basic block diagram of carry look ahead adder is discussed in this. The carry look ahead adder using the concept of propagating and generating the carry bit. It calculates one or more carry bits before the sum, which reduces the wait time to calculate the result of the larger value bits.

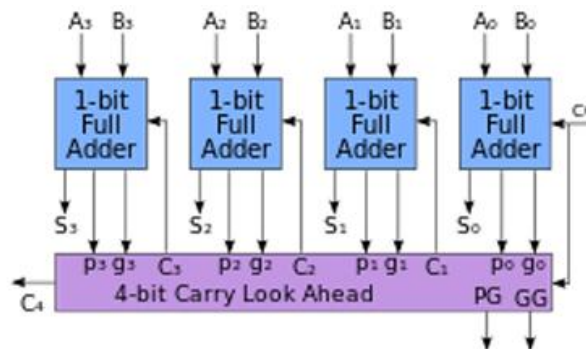


Fig 1: Block diagram of 4 bit carry look ahead adder

## 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
    input      Cin,
    output [3:0] Sum,
    output
        Cout
);
    wire [3:0] G, P;
    Propagate signals
    wire [4:0] C;
    assign C[0] = Cin;

    assign G = A & B;    assign
P = A ^ B;

    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]);

    assign Sum = P ^ C[3:0];
    assign Cout = C[4];

endmodule

```

1  
2

Listing 1: Verilog RTL — 4 bit carry look adder

# Testbench:

A simple testbench used for functional verification

1

```
`timescale 1ns / 1ps

module tb_cla_4bit;

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

    cla_4bit uut (
        .A(A),
        .B(B),
        .Cin(Cin),
        .Sum(Sum),
        .Cout(Cout)
    );

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

    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.

## 4 Schematic (Cell-Level)

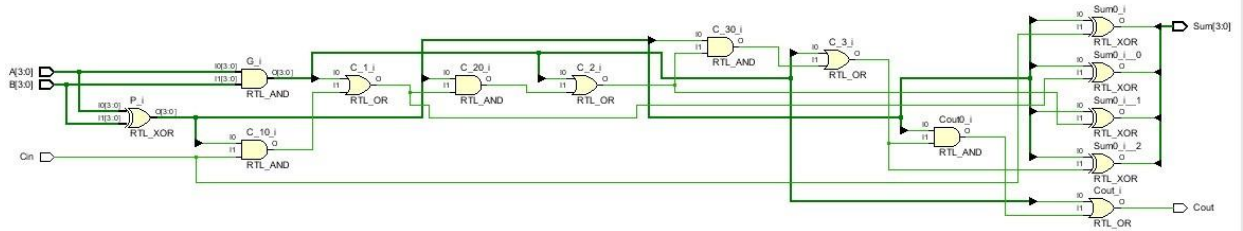


Figure 1: Schematic view (cell-level) of the 4-bit carry Adder after synthesis.

## 5 Functional Simulation Waveforms

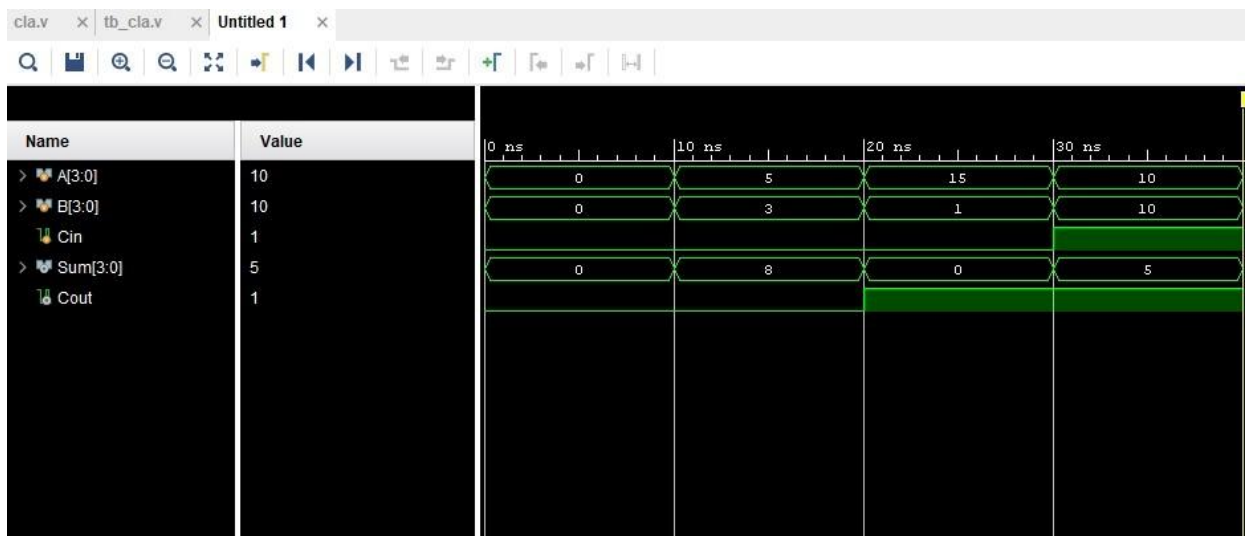


Figure 2: Waveform from SimVision demonstrating correct CLA operation for test vectors.

## **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.