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.

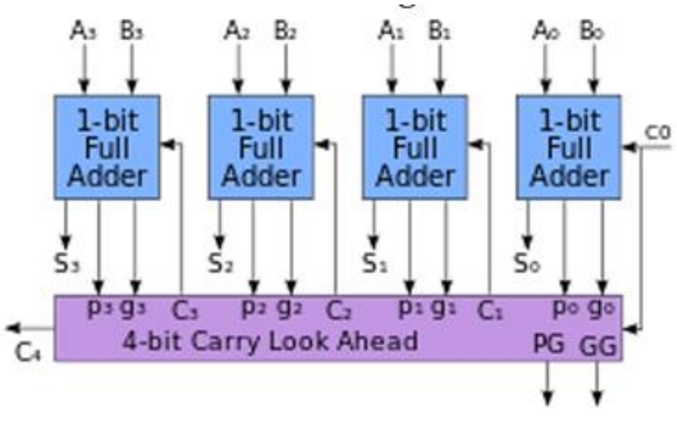
2 Design Flow

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

Design Of Carry Look Ahead Adder and Ripple Carry 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.



Ripple carry adder : A ripple carry adder is a logic circuit in which the carry-out of each full adder is the carry in of the succeeding next most significant full adder. It is called a ripple carry adder because each carry bit gets rippled into the next stage. In a ripple carry adder the sum and carry out bits of any half adder stage is not valid until the carry in of that stage occurs.

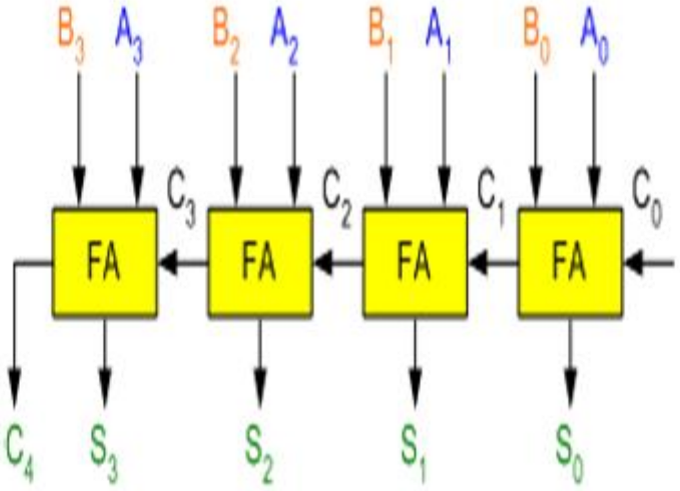


Fig 2 Block diagram of ripple carry adder

RTL Design (Source Code)

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)

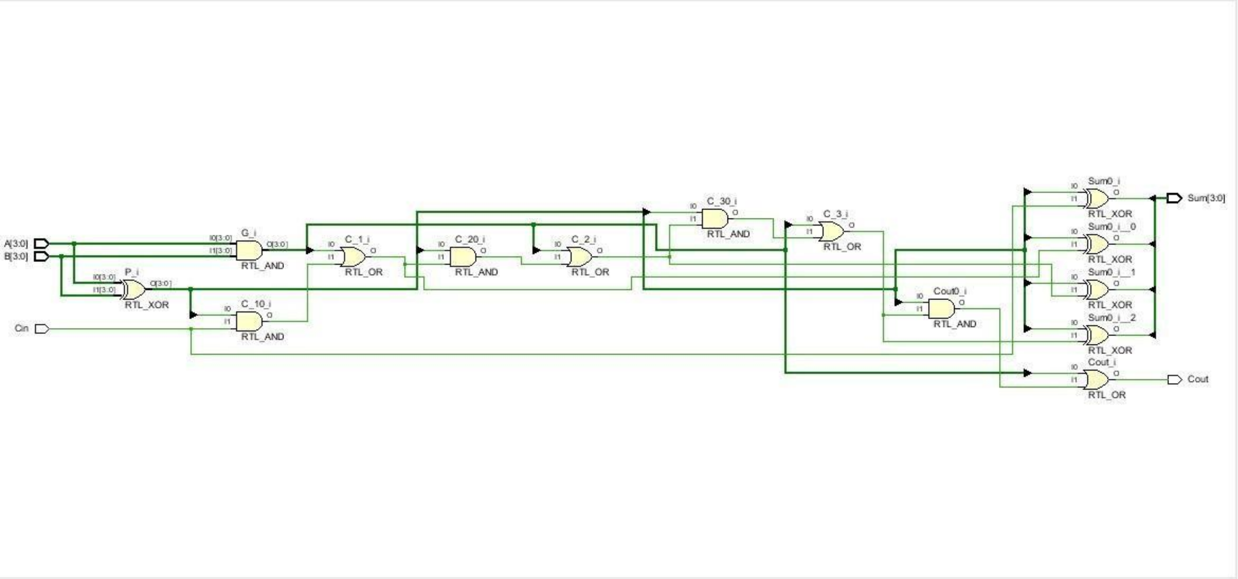
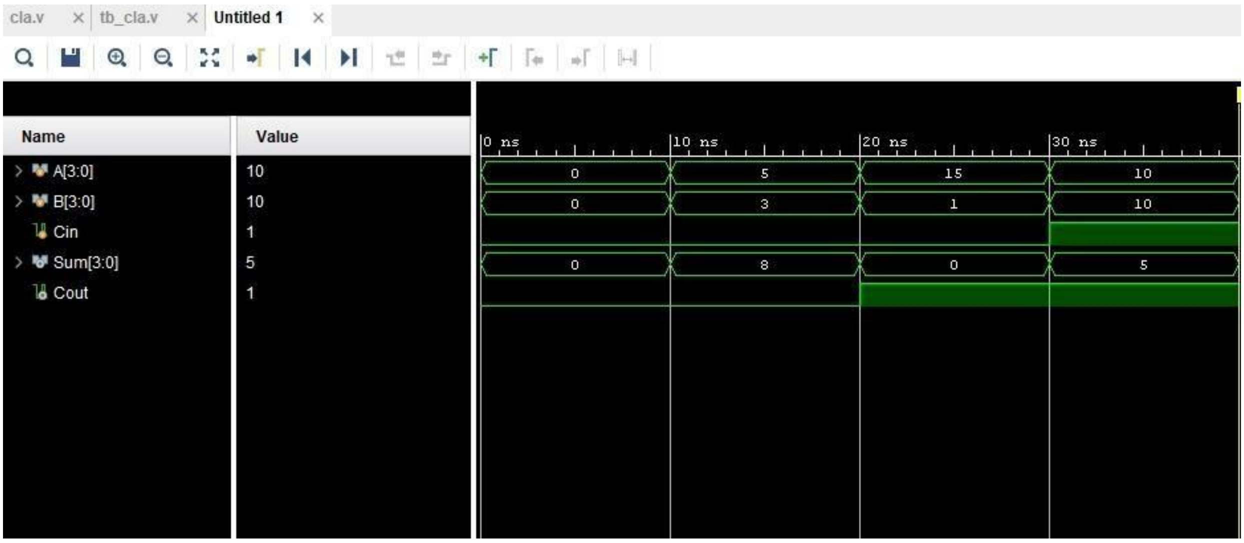


Figure 1: Schematic view (cell-level) of the 4-bit carry Adder after synthesis.

5 Functional Simulation Waveforms



Name	Value
A[3:0]	10
B[3:0]	10
Cin	1
Sum[3:0]	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.