Project 16: Conditional Sum Adder

A Comprehensive Study of Advanced Digital Circuits

By: Nikunj Agrawal , Abhishek Sharma, Ayush Jain , Gati Goyal

Contents

1 Introduction

The **Conditional Sum Adder** is an advanced digital adder designed to improve the speed of addition operations. Unlike traditional adders, which propagate carry signals sequentially, the COSA performs multiple carry computations in parallel to enhance computational speed. This architecture is particularly beneficial in high-speed arithmetic operations where carry propagation delay can be a limiting factor.

2 Background

Traditional adders like the Ripple Carry Adder (RCA) suffer from significant delays due to sequential carry propagation. To address this, the Conditional Sum Adder (COSA) performs two possible carry computations simultaneously, each corresponding to different carry-in conditions. By using a multiplexer to select the correct result based on the actual carry-in, the COSA reduces the propagation delay and increases the overall speed of addition.

The COSA architecture generally involves: - Multiple carry lookahead units that calculate potential sums based on different carry assumptions. - A multiplexer that selects the final sum and carry-out based on actual input conditions.

3 Structure and Operation

The structure of the Conditional Sum Adder consists of the following key components:

- **Input Ports**: Two primary inputs, A and B, and a carry input Cin.
- **Sum Calculation Units**: Units compute potential sums based on different carry-in assumptions:
  - $S0 = A + B$ (when $Cin = 0$)
  - $S1 = A + B + 1$ (when $Cin = 1$)
- **Carry Generation Logic**: Each sum calculation unit generates carry-out signals:
  - $C0$ for $S0$
  - $C1$ for $S1$
- **Multiplexer (MUX)**: Selects the correct sum output based on the value of Cin.
- **Output Port**: The final output, Sum, is a signal that includes the computed sum and possibly a carry-out signal.

The operation of the COSA can be summarized as follows:

1. Initialize inputs A, B, and Cin.
2. Compute possible sums $S0$ and $S1$ based on Cin.
3. Generate corresponding carry-out signals $C0$ and $C1$.
4. Use the multiplexer to select the final sum based on the actual carry-in.
5. Output the result.

3.1 Design Considerations

When designing the COSA, several factors should be considered to achieve optimal performance:

- *Bit-width Scalability*: The COSA can be extended to larger bit-widths to handle more complex arithmetic operations.
- *Resource Utilization*: The design must balance speed and area, as additional parallel computations can increase hardware complexity.
- *Power Efficiency*: Efficient design practices are important to minimize power consumption while achieving high-speed performance.

3.2 Performance Metrics

The performance of the COSA is evaluated based on several metrics:

- *Propagation Delay*: The time required for the output to reflect changes in the inputs. The COSA reduces this delay compared to traditional adders.
- *Throughput*: The rate at which addition operations are completed. The COSA improves throughput by performing parallel carry computations.
- *Power Consumption*: The amount of power used during operation. Design optimizations can help reduce power consumption while maintaining performance.

4 Implementation in SystemVerilog

The following RTL code implements the Conditional Sum Adder in SystemVerilog:

```
 module conditional_sum_adder (
    input logic [3:0] A,       // 4-bit input A
    input logic [3:0] B,       // 4-bit input B
    input logic Cin,           // Carry-in
    output logic [3:0] Sum,    // 4-bit Sum
    output logic Cout          // Carry-out
);

    logic [4:0] P, G;          // Propagate and Generate signals
    logic [3:0] C;             // Carry signals

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

    // Carry computations
    assign C[0] = Cin;
    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 Cout = G[3] | (P[3] & C[3]);

    // Sum computation
    assign Sum = P ^ {C[2:0], Cin};

endmodule
```


5 Test Bench

The test bench for the Conditional Sum Adder is used to verify the correctness of the design. It generates various test cases and compares the output with expected results.

```
 module tb_conditional_sum_adder;

    // Testbench signals
    logic [3:0] A;
    logic [3:0] B;
    logic Cin;
    logic [3:0] Sum;
    logic Cout;
```

```
    // Instantiate the Unit Under Test (UUT)
    conditional_sum_adder uut (
        .A(A),
        .B(B),
        .Cin(Cin),
        .Sum(Sum),
        .Cout(Cout)
    );

    // Testbench logic
    initial begin
        // Monitor signals
        $monitor("Time: %0t | A: %b | B: %b | Cin: %b | Sum: %b | Cout: %b",
                 $time, A, B, Cin, Sum, Cout);

        // Test Case 1
        A = 4'b0001; B = 4'b0010; Cin = 1'b0;
        #10;

        // Test Case 2
        A = 4'b0110; B = 4'b1010; Cin = 1'b1;
        #10;

        // Test Case 3
        A = 4'b1111; B = 4'b0001; Cin = 1'b0;
        #10;

        // Test Case 4
        A = 4'b1010; B = 4'b0101; Cin = 1'b1;
        #10;

        // End simulation
        $finish;
    end

endmodule
```


6 Advantages and Disadvantages

The Conditional Sum Adder offers several advantages and disadvantages:

6.1 Advantages

- *Reduced Propagation Delay*: By performing parallel carry computations, the COSA reduces the overall delay.
- *Increased Speed*: The COSA can achieve higher addition speeds compared to traditional adders.

6.2 Disadvantages

- *Increased Hardware Complexity*: The parallel computation units and multiplexers can increase the complexity and area of the design.
- *Higher Power Consumption*: Additional logic for carry computations may lead to increased power consumption.

7 Conclusion

The Conditional Sum Adder (COSA) represents a significant improvement over traditional adder designs by addressing the carry propagation delay issue through parallel computations. The use of multiple carry lookahead units and a multiplexer allows for faster addition operations and improved performance. This design is particularly useful in applications where speed is critical, such as digital signal processing and high-speed computing.

While the COSA offers enhanced performance, it also comes with increased complexity and power consumption. These factors must be carefully considered when implementing the COSA in real-world applications. Overall, the COSA provides a valuable enhancement to digital arithmetic circuits, balancing speed and efficiency in modern computing systems.

8 Simulation Results

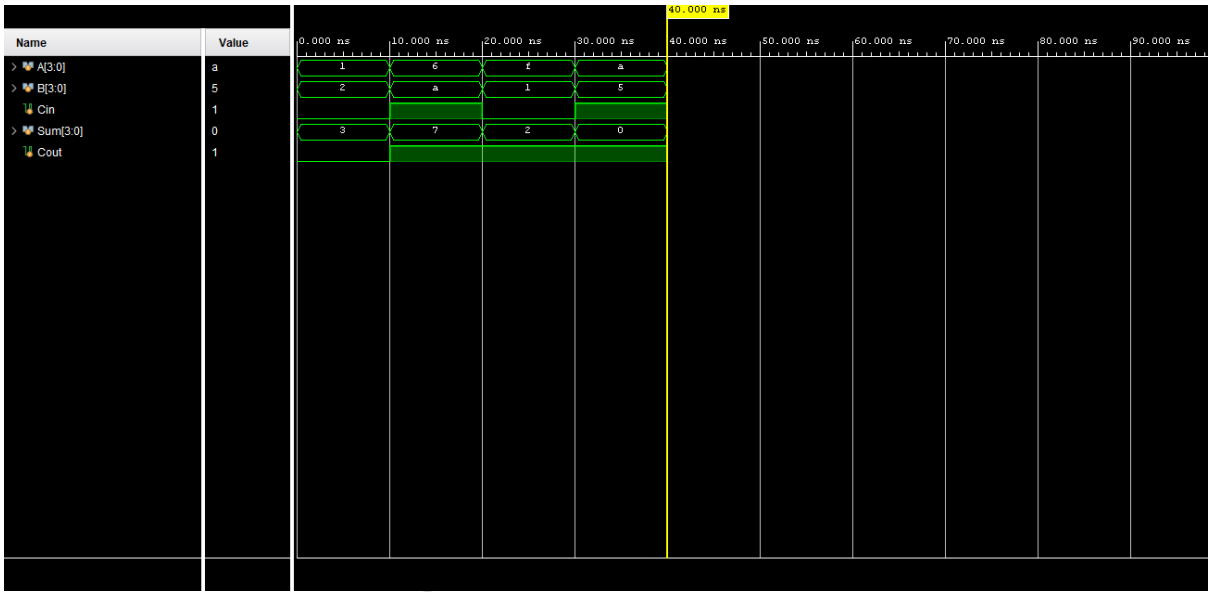

Figure 1: Simulation results of Conditional Sum Adder

9 Schematic

10 Synthesis Design

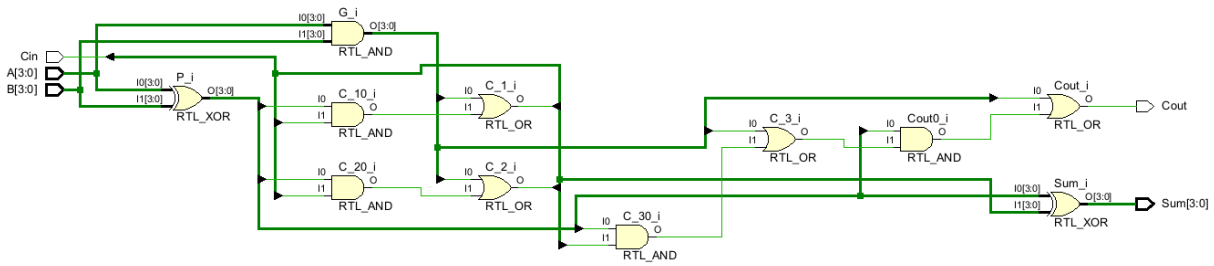

Figure 2: Schematic of Conditional Sum Adder

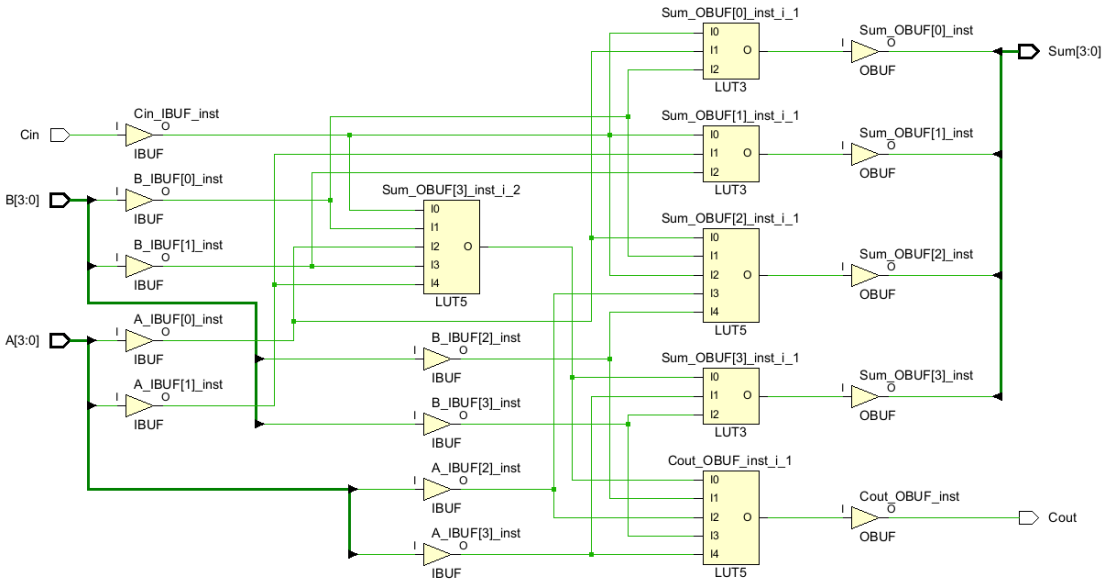

Figure 3: Synthesis of Conditional Sum Adder