Project 7: Carry Select Adders

A Comprehensive Study of Advanced Digital Circuits

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

Contents

1 Introduction

A Carry Select Adder (CSA) is a high-speed adder used in digital circuits to improve the speed of binary addition. It achieves this by precomputing the sum and carry outputs for both possible values of the carry-in, allowing the correct result to be quickly selected once the actual carry-in is known.

2 Structure and Operation

The Carry Select Adder divides the addition process into several smaller sub-adders, each responsible for a portion of the total bit-width. Each sub-adder performs two parallel additions for each possible carry-in (0 and 1), and the final result is selected based on the actual carry-in.

1. **Division into Blocks:** The adder is divided into multiple blocks, each handling a portion of the bits.
2. **Dual Addition Paths:** Each block computes two possible sums and carries:
   - One assuming the carry-in is 0.
   - One assuming the carry-in is 1.
3. **Multiplexers for Selection:** Once the actual carry-in for a block is known, a multiplexer selects the correct sum and carry-out based on this carry-in.

3 Example: 16-bit Carry Select Adder

Consider a 16-bit Carry Select Adder divided into four blocks, each containing 4 bits.

1. **Block 1 (Bits 0-3):**
   - Computes the sum and carry for bits 0-3 assuming carry-in is 0.
   - Computes the sum and carry for bits 0-3 assuming carry-in is 1.
2. **Block 2 (Bits 4-7):**
   - Computes the sum and carry for bits 4-7 assuming carry-in is 0.
   - Computes the sum and carry for bits 4-7 assuming carry-in is 1.
3. **Block 3 (Bits 8-11):**
   - Computes the sum and carry for bits 8-11 assuming carry-in is 0.
   - Computes the sum and carry for bits 8-11 assuming carry-in is 1.
4. **Block 4 (Bits 12-15):**
   - Computes the sum and carry for bits 12-15 assuming carry-in is 0.
   - Computes the sum and carry for bits 12-15 assuming carry-in is 1.

Each block has two outputs for sums and carries: one for carry-in 0 and one for carry-in 1. The actual carry-in for each block is determined by the carry-out of the previous block, selected using a multiplexer.

4 Synthesis Design

5 Circuit Diagram

The circuit diagram for a Carry Select Adder includes:

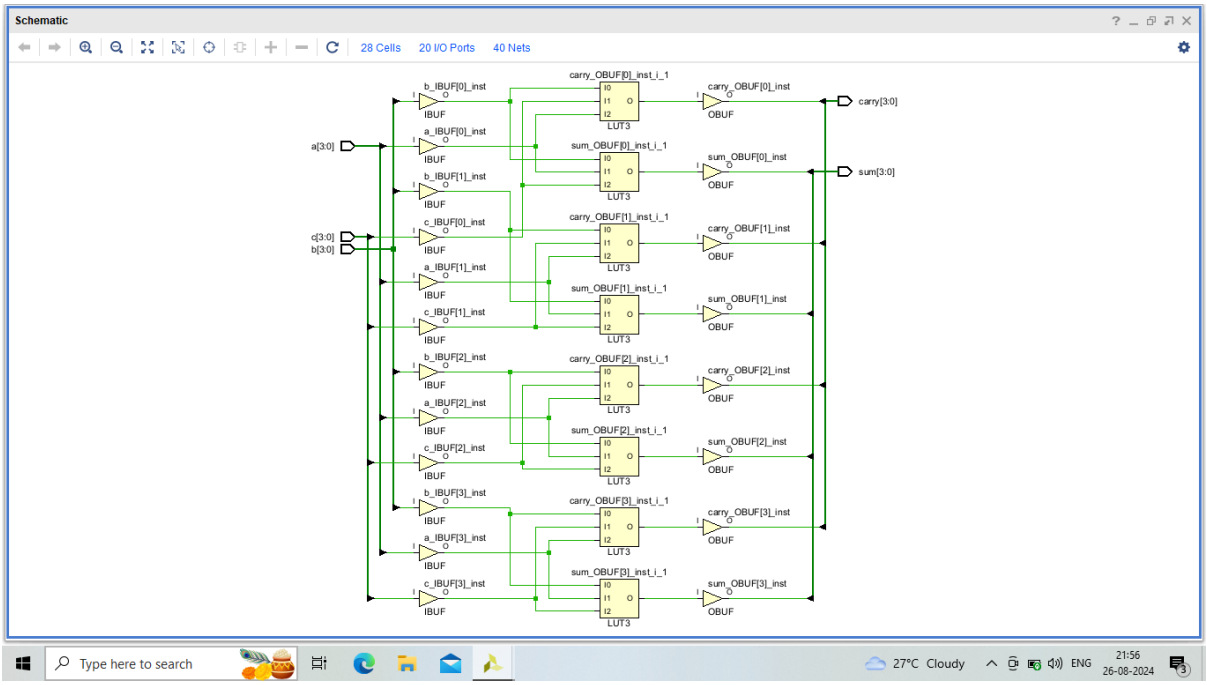

Figure 1: Synthesis of Carry Select Adder

Block 1:	Block 2:	Block 3:	Block 4:
RCA0	RCA0	RCA0	RCA0
RCA1	RCA1	RCA1	RCA1
Mux	Mux	Mux	Mux

6 RTL Code

Listing 1: Carry Save Adder RTL Code

```
module carry_save_adder #(parameter N = 4) (
    input logic [N-1:0] a,
    input logic [N-1:0] b,
    input logic [N-1:0] c,
    output logic [N-1:0] sum,
    output logic [N-1:0] carry
);

    assign sum = a ^ b ^ c;
    assign carry = (a & b)  (b & c)  (c & a);

endmodule
```

6.1 Testbench

Listing 2: 32 bit Adder Testbench

```
module tb_carry_save_adder;

    parameter N = 4;
    logic [N-1:0] a, b, c;
    logic [N-1:0] sum, carry;
```

```
6
7       carry_save_adder #(N) csa (.a(a), .b(b), .c(c), .sum(sum),
            .carry(carry));
8
9       initial begin
10          // Test cases
11          a = 4'b0011; b = 4'b0101; c = 4'b0110; #10;
12          $display("A = %b, B = %b, C = %b, SUM = %b, CARRY = %b", a, b,
                c, sum, carry);
13
14          a = 4'b1111; b = 4'b0001; c = 4'b0001; #10;
15          $display("A = %b, B = %b, C = %b, SUM = %b, CARRY = %b", a, b,
                c, sum, carry);
16
17          a = 4'b1001; b = 4'b1001; c = 4'b1001; #10;
18          $display("A = %b, B = %b, C = %b, SUM = %b, CARRY = %b", a, b,
                c, sum, carry);
19
20          $finish;
21      end
22
23  endmodule
```

7 Simulation Results

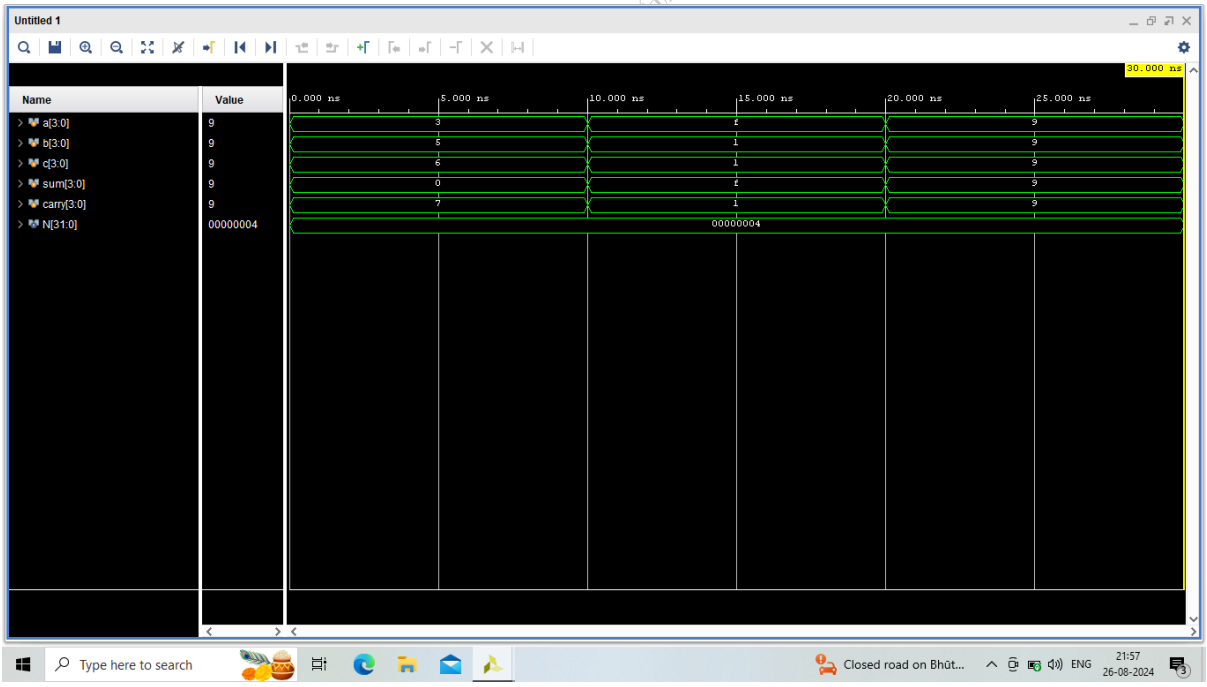

Figure 2: Simulation results of Carry Select Adder

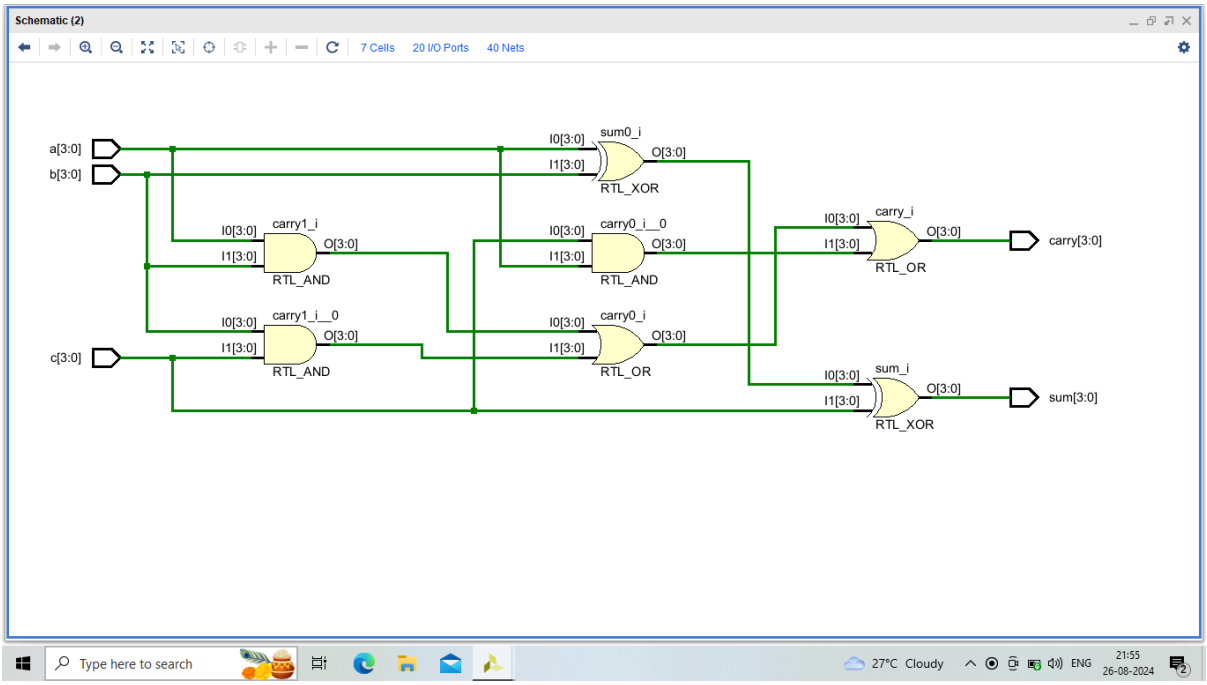

Figure 3: Schematic of Carry Select Adder

8 Schematic

9 Advantages

- **Speed:** By precomputing results for both possible carry-in values, the Carry Select Adder significantly reduces the critical path delay compared to a simple Ripple Carry Adder.
- **Scalability:** The design can be easily scaled to larger bit-widths by adding more blocks.

10 Disadvantages

- **Area and Power:** The duplication of hardware to compute results for both carry-in values increases the area and power consumption.

11 Applications

- **High-Speed Arithmetic Units:** Widely used in arithmetic logic units (ALUs) and processors where fast addition is crucial.
- **Digital Signal Processing (DSP):** Used in DSP applications requiring rapid arithmetic operations.

12 Conclusion

The Carry Select Adder balances the trade-off between speed and complexity, providing a faster alternative to Ripple Carry Adders by leveraging parallel computation and selection logic. Its ability to quickly select the correct sum and carry-out makes it an essential component in high-speed arithmetic operations.