Project 22 : Manchester Carry Chain Adder(MCC)

A Comprehensive Study of Advanced Digital Circuits

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

Contents

1 Project Overview

The MCC Adder is a high-speed adder designed to optimize the carry propagation process, making it faster than traditional adders like the Ripple Carry Adder (RCA).

2 Manchester Carry Chain Adder(MCC)

2.1 Description

The Manchester Carry Chain (MCC) Adder is a type of high-speed adder that focuses on minimizing carry propagation delay by using an efficient carry chain mechanism. It is a parallel prefix adder designed to improve the speed of binary addition by allowing carry signals to propagate quickly across multiple stages. The MCC Adder was first introduced by engineers at the University of Manchester and is particularly well-suited for Very Large Scale Integration (VLSI) designs, where reducing delay is crucial.

2.2 Key Features of the Manchester Carry Chain Adder (MCC):

- **Carry Propagation Speed:**

The MCC Adder uses a special chain structure for carry propagation, allowing the carry signals to propagate through multiple stages with minimal delay. The carry chain is implemented in a way that the carry signals can ripple through the chain in constant time, regardless of the adder's size.

- **Parallel Prefix Structure:**

Like other parallel prefix adders, the MCC Adder employs a prefix tree to compute the generate (G) and propagate (P) signals in parallel for each bit position. These signals determine whether a carry is generated or propagated across the bit positions.

- **Generate and Propagate Signals:**

For each bit, the adder computes the generate (G) and propagate (P) signals, which dictate whether a carry is generated or passed forward. Generate (G): If both inputs are 1, the bit generates a carry. Propagate (P): If one of the inputs is 1, the carry is propagated to the next bit.

- **Efficient Carry Chain Design:**

The MCC Adder's defining feature is its carry chain, which allows the carry to ripple through each bit quickly. This chain minimizes the time it takes to compute the carry signals across wide bit-widths, making it highly efficient for high-speed arithmetic operations.

- **Constant Propagation Time:**

The design of the Manchester Carry Chain ensures that the carry propagation time remains nearly constant, even for wide bit-widths like 16-bit, 32-bit, or more. This is achieved by carefully optimizing the wiring and logic within the carry chain.

- **Simple Hardware Design:**

Despite its speed advantages, the MCC Adder maintains a relatively simple hardware design, making it less complex and more power-efficient than other high-speed adders like the Carry Look-Ahead Adder (CLA)

2.3 RTL Code

Listing 1: ManchesterCarry Chain Adder(MCC)

```

MCC Adder Code
systemverilog
```

```

module mcc_adder #(parameter N = 4) (
    input  [N-1:0] A,
    input  [N-1:0] B,
    output [N-1:0] Sum,
    output CarryOut
);

    wire [N-1:0] P, G;
    wire [N:0] C;

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

    // Carry chain using Manchester logic
    assign C[0] = 1'b0; // Initial carry in
    generate
        genvar i;
        for (i = 0; i < N; i = i + 1) begin
            assign C[i+1] = G[i]  (P[i] & C[i]);
        end
    endgenerate

    assign Sum = P ^ C[N-1:0];
    assign CarryOut = C[N];

endmodule
```

2.4 Testbench

Listing 2: ManchesterCarry Chain Adder(MCC)

```
Testbench for MCC Adder
systemverilog
module tb_mcc_adder;

    reg [3:0] A;
    reg [3:0] B;
    wire [3:0] Sum;
    wire CarryOut;

    mcc_adder #(4) uut (
        .A(A),
        .B(B),
        .Sum(Sum),
        .CarryOut(CarryOut)
    );

    initial begin
        // Test case 1
        A = 4'b0110; // 6
        B = 4'b0011; // 3
        #10;
        $display("A = %b, B = %b, Sum = %b, CarryOut = %b", A, B, Sum,
           CarryOut);

        // Test case 2
```

```
        A = 4'b1101; // 13
        B = 4'b0101; // 5
        #10;
        $display("A = %b, B = %b, Sum = %b, CarryOut = %b", A, B, Sum,
           CarryOut);

        $stop;
    end
endmodule
```

3 How it works ?

3.1 Working of the Manchester Carry Chain Adder:

1. Generate/Propagate Computation:

For each bit position in the binary numbers to be added, the MCC Adder calculates the generate and propagate signals based on the inputs.

2. Carry Chain Propagation:

The carry chain efficiently propagates the carry signals across the bit positions. In contrast to Ripple Carry Adders (RCA), where the carry must ripple sequentially through each bit, the MCC Adder's optimized carry chain allows the carry to move in parallel across multiple bits, significantly reducing the carry propagation delay.

3. Sum Computation:

Once the carry signals have been computed for each bit, the sum for each bit is calculated using the propagate signal and the carry-in for that bit.

Explanation

The Manchester carry chain adder is a type of parallel adder that uses a technique called carry-lookahead to reduce the time it takes for carries to propagate through the adder. This makes it faster than traditional ripple-carry adders.

3.2 Key Concepts in the MCC Adder:

Generate and Propagate Logic

The *generate* ($G[i]$) and *propagate* ($P[i]$) signals are essential in controlling the flow of the carry signals across the bit positions.

$$G[i] = A[i] \,\&\, B[i]$$

The generate signal is set when both inputs are 1, causing a carry to be generated.

$$P[i] = A[i] \ B[i]$$

The propagate signal indicates that a carry from the previous bit can propagate through the current bit if there is no generation.

Manchester Carry Chain

The MCC Adder implements an efficient carry chain mechanism, where the carry signals are computed in parallel for all bit positions. This design minimizes the delay in carry propagation, making the addition process faster than in conventional adders, such as Ripple Carry Adders.

Sum Computation

The sum is computed based on the propagate signal and the carry signal for each bit, ensuring accurate addition.

$$\mathrm{Sum}[i] = P[i] \oplus C[i]$$

3.3 Comparison with Other Adders:

- **Manchester Carry Chain (MCC) Adder:**

Speed: Faster than traditional sequential adders due to parallel carry propagation. Complexity: Moderate complexity, balances speed and hardware requirements. Use Case: High-performance applications with moderate bit-widths.

- **Ripple Carry Adder (RCA):**

Speed: Slow; carries propagate sequentially. Complexity: Simple, minimal hardware. Use Case: Suitable for low-power, small bit-width applications. MCC vs. RCA: MCC is much faster but slightly more complex.

- **Carry Look-Ahead Adder (CLA):**

Speed: Very fast; precomputes carries in parallel. Complexity: High hardware complexity and power consumption. Use Case: High-speed, large bit-width operations. MCC vs. CLA: MCC is simpler but slower than CLA for large bit-widths.

- **Brent-Kung Adder:**

Speed: Fast; logarithmic delay with reduced fan-out. Complexity: Moderate, uses a tree structure for efficiency. Use Case: High-performance systems with focus on area optimization. MCC vs. Brent-Kung: Brent-Kung is faster for larger bit-widths but more complex.

- **Kogge-Stone Adder:**

Speed: Extremely fast, logarithmic carry propagation. Complexity: High; large area and power usage due to extensive wiring. Use Case: Ultra-high-speed computing with large bit-widths. MCC vs. Kogge-Stone: MCC is simpler and less power-hungry, but slower for very large bit-widths.

3.4 Advantages of the MCC Adder:

- **Reduced Carry Propagation Delay:**

The primary advantage of the MCC Adder is the significant reduction in carry propagation delay. By calculating the carry signals in parallel using the Manchester carry chain, the adder achieves much faster addition times, especially for wide bit-widths like 16-bit, 32-bit, or more.

- **High-Speed Operation:**

The MCC Adder is particularly useful in high-speed arithmetic units such as Arithmetic Logic Units (ALUs) and Digital Signal Processors (DSPs), where fast addition is critical to system performance.

- **Scalability:**

The Manchester Carry Chain design can be easily scaled to accommodate different bit-widths, making it flexible for use in various digital systems, from small processors to large data processing units.

- **Hardware Simplicity:**

Compared to more complex high-speed adders like Carry Look-Ahead Adders (CLA), the MCC Adder maintains a relatively simple structure, which reduces hardware complexity and power consumption.

3.5 Disadvantages of the MCC Adder:

- **Fan-Out Issues:**

One challenge with the MCC Adder is managing fan-out, where certain stages in the carry chain may need to drive multiple subsequent stages. This can lead to unbalanced delays and complicates the design in larger systems.

- **Wiring Complexity:**

As the carry chain spans more bits, the wiring and interconnects between the stages can become complex, especially in larger adders. This can increase the overall area and complexity of the adder.

3.6 Applications of the MCC Adder:

- **High-Performance Processors:**

The MCC Adder is ideal for use in high-performance processors where fast arithmetic operations are essential for overall system speed. It is often found in ALUs and floating-point units (FPUs).

- **Digital Signal Processing (DSP):**

In DSP systems, where real-time data processing is crucial, the MCC Adder's ability to quickly handle large bit-width additions makes it a good choice for applications such as filtering, convolution, and fast Fourier transforms.

- **Application-Specific Integrated Circuits (ASICs):**

The MCC Adder is commonly used in ASIC designs where performance, area, and power consumption are key considerations. Its high-speed operation and simplicity make it well-suited for custom integrated circuits used in communication and multimedia processing.

4 Results

4.1 Simulation Results

4.2 Schematic

4.3 Synthesis Design

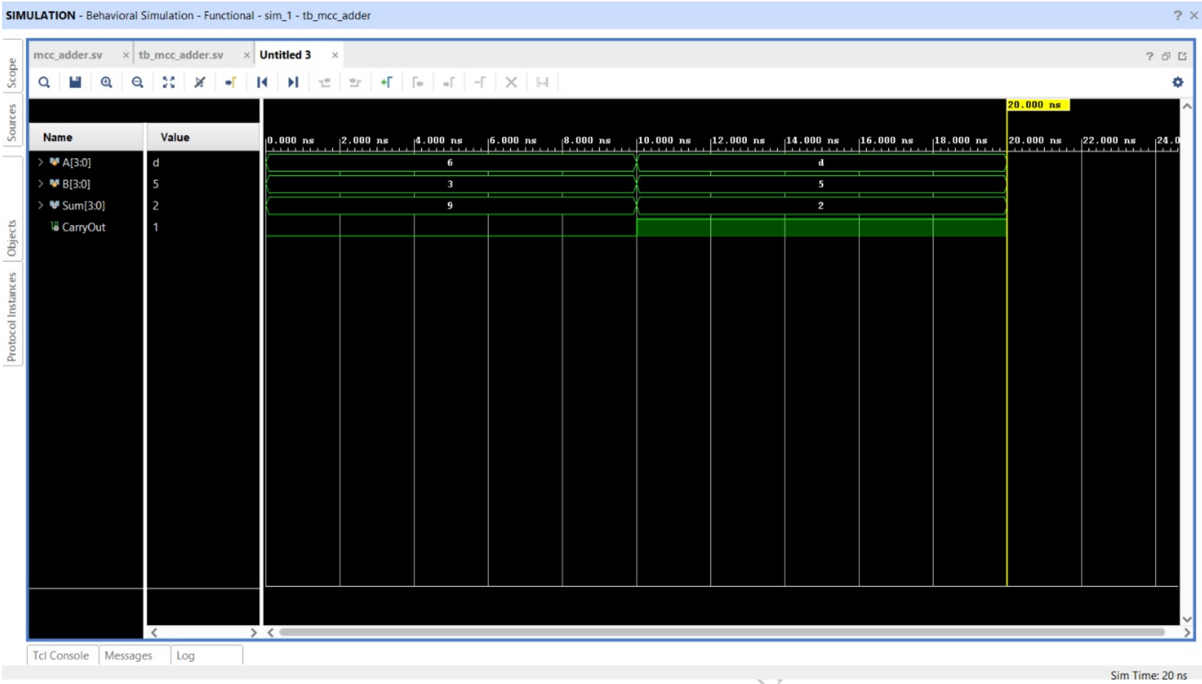

Figure 1: Simulation results of Manchester Carry Chain Adder

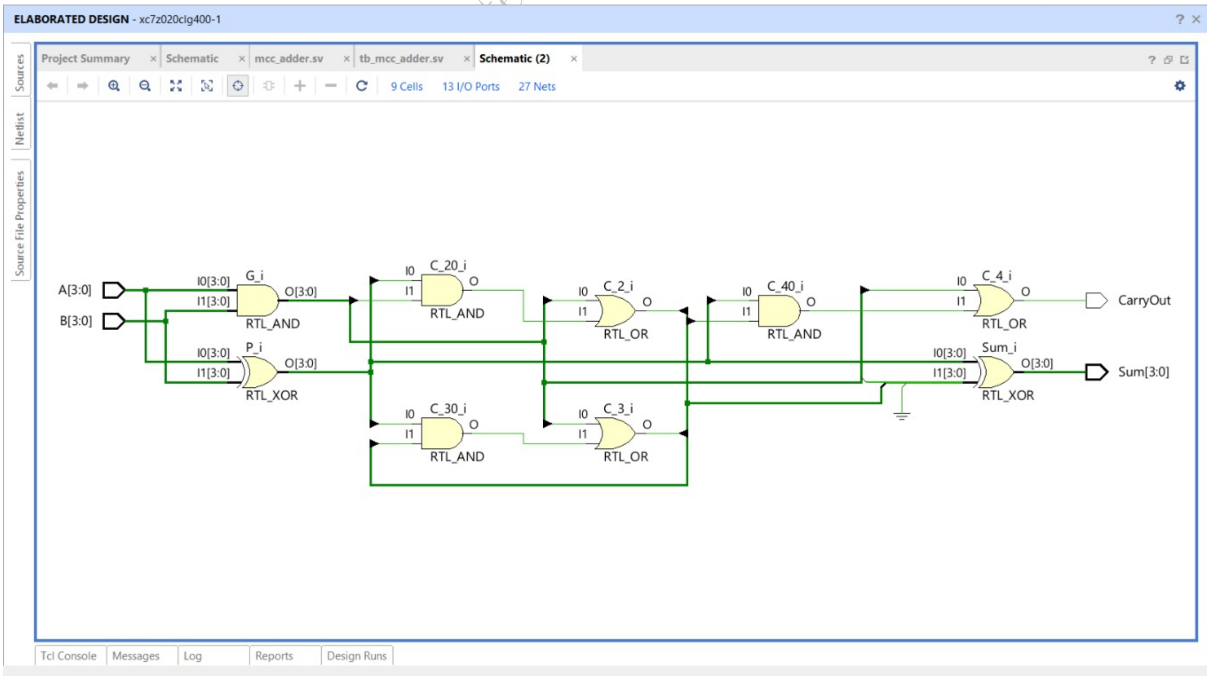

Figure 2: Schematic of Manchester Carry Chain Adder

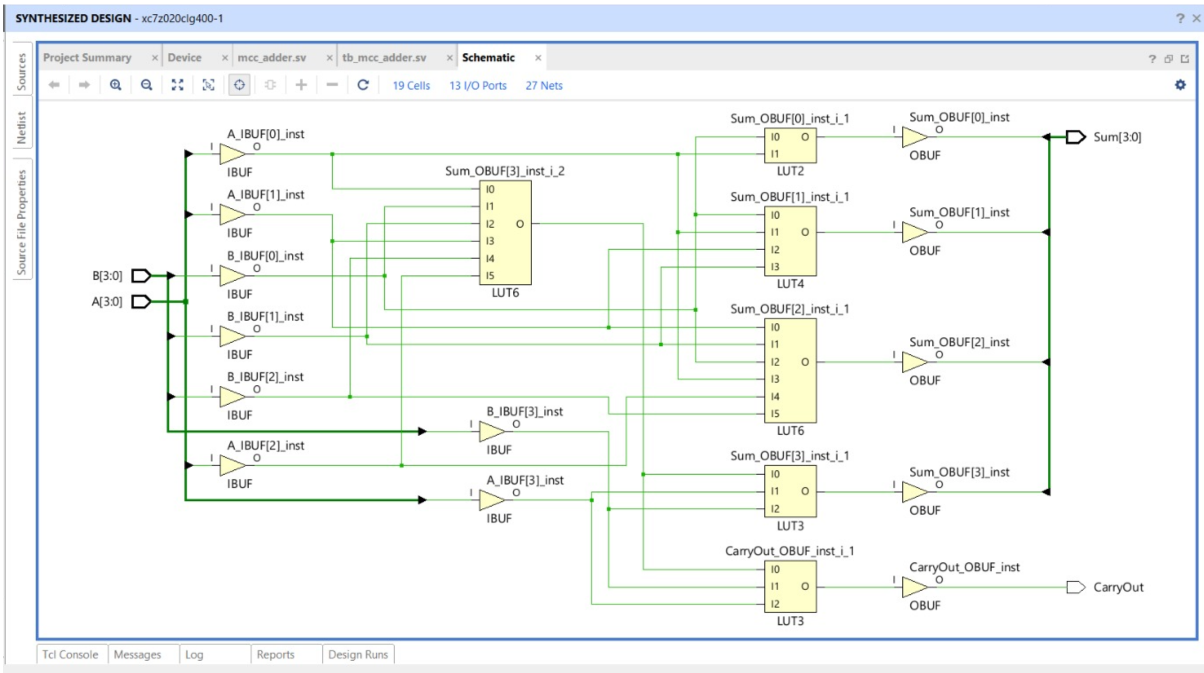

Figure 3: Synthesis Design of Manchester Carry Chain Adder