Project 33 : Signed Sequential Multilpier

A Comprehensive Study of Advanced Digital Circuits

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

Contents

1 Introduction

The **signed sequential multilpier** The Signed Sequential Multiplier is a fundamental component in digital systems, widely used in various computational tasks such as arithmetic operations, signal processing, and processor designs where signed number multiplication is essential. As digital systems evolve to handle more complex and dynamic operations, efficient signed multiplication techniques become critical for applications requiring both positive and negative number operations.

Signed sequential multiplication operates by processing two signed binary numbers to produce their product, accounting for the sign of each operand. Unlike simple combinational multipliers, which aim for immediate results, sequential multipliers compute the product iteratively over multiple clock cycles. This approach allows for a more resource-efficient design at the cost of a slower computation time. By sequentially generating partial products and summing them, signed sequential multipliers can handle larger bit-widths with minimal hardware resources.

The architecture of the Signed Sequential Multiplier leverages state machines and iterative logic to systematically compute the final product over time. This design approach ensures flexibility and scalability, particularly when dealing with larger signed numbers in resource-constrained environments.

This documentation presents a detailed exploration of the Signed Sequential Multiplier design, including its structure, operation, and performance evaluation through SystemVerilog simulation. By investigating its implementation and providing a thorough testing framework, this project seeks to demonstrate the effectiveness and efficiency of the signed sequential multiplier in handling signed binary multiplication in modern digital systems.

2 Background

Binary multiplication is a fundamental operation in digital arithmetic, essential for various computational applications, from basic arithmetic functions in simple calculators to more complex processors and digital signal processors (DSPs). In most digital systems, numbers are often represented in binary form, and multiplication must account for both positive and negative values, making signed multiplication a crucial aspect for accurate computation in many applications.

The process of multiplying signed binary numbers can be seen as an extension of traditional binary multiplication, with an additional requirement of handling the sign of the operands. This involves first determining the sign of the result based on the signs of the multiplicand and multiplier, followed by performing the multiplication of the absolute values of the two numbers. The resulting product is then adjusted to reflect the correct sign. While combinational multipliers execute this in a single cycle, sequential multipliers, like the signed sequential multiplier, process partial products over multiple clock cycles, allowing for a more resource-efficient implementation.

In digital circuits, signed multiplication is often realized using algorithms that sequentially add and shift partial products, much like the process of long multiplication in decimal systems. The signed sequential multiplier improves upon this by reducing the complexity and hardware required for the multiplication of large signed numbers, spreading the computation over several cycles to minimize the need for extensive combinational logic.

The signed sequential multiplier is versatile, as it can handle both positive and negative operands by using additional control logic to manage sign extensions and ensure the correct alignment of partial products. This makes it suitable for applications where resource constraints and power efficiency are of paramount importance, such as in embedded systems or digital signal processing units, where high throughput may not always be the primary goal.

As digital systems grow in complexity, the demand for efficient signed multiplication techniques increases. Signed sequential multipliers offer a balanced approach by trading off speed for reduced area and power consumption, making them a preferred choice in applications that require efficient handling of signed numbers with constrained hardware resources.

3 Structure and Operation

The Signed Sequential Multiplier efficiently multiplies two signed binary numbers using a step-by-step process. It consists of registers to store the signed multiplicand and multiplier, a 32-bit product register, and a control unit to manage sequential operations. The design may incorporate **Booth's Algorithm** or **Shift-and-Add Logic** to minimize the number of additions required. A **sign extension unit** ensures correct sign handling. During operation, the multiplier shifts through bits, adding partial products sequentially, with the control unit determining the sign of the final product after all cycles are completed.

3.1 Key Components

- **Input Ports**:
  - **Multiplicand**: A 16-bit signed binary input representing the multiplicand.
  - **Multiplier**: A 16-bit signed binary input representing the multiplier.
- **Control Unit**: The control unit governs the sequential operation of the multiplier, handling the shift-and-add process over multiple clock cycles. It tracks the number of cycles, monitors the operation, and determines when the multiplication is complete.
- **Partial Product Generation**: Each bit of the multiplier is used to generate partial products. The multiplicand is either added to or subtracted from the product based on the value of the multiplier's bits.
  - For each bit m_i of the multiplier:
    - If $m_i = 1$, the multiplicand is added to the product, shifted left by the appropriate number of positions.
    - If $m_i = 0$, the partial product is 0 and no addition is performed.
- **Sign Extension Logic**: The sign of the result is handled through sign extension to ensure that signed numbers are treated correctly during the shifting and accumulation process.
- **Addition and Accumulation Unit**: The partial products generated in each cycle are accumulated to form the final result. This accumulation is performed over multiple cycles in a sequential manner, unlike combinational multipliers that produce the result in a single clock cycle.
- **Output Port**:
  - **Product**: A 32-bit signed output representing the final result of the multiplication, capable of holding the result of multiplying two 16-bit signed numbers.

3.2 Operation Steps

The operation of the **Signed Sequential Multiplier** can be summarized in the following steps:

1. **Initialization**: Load the signed multiplicand and multiplier into their respective registers. The product register is initialized to zero, and the control unit prepares for the sequential multiplication process.
2. **Partial Product Generation and Accumulation**: For each bit of the multiplier:
   - If the bit is set (1), the multiplicand is added to the product, with appropriate shifts applied based on the bit's position in the multiplier.
   - If the bit is not set (0), no addition is performed for that bit.

   This process is repeated over several clock cycles until all bits of the multiplier have been processed.
3. **Handling the Sign**: If the signs of the multiplicand and multiplier differ, the result's sign is adjusted accordingly, as per two's complement rules for signed numbers.
4. **Output**: Once the multiplication is complete, the final product is sent to the output port, and the operation is finished.

3.3 Example

For instance, if the multiplicand is $A = -5$ (binary 1111 1011 in 8-bit signed representation) and the multiplier is $B = 3$ (binary 0000 0011), the multiplication proceeds as follows:

- **Partial Product Generation**:
  - First partial product: $A = -5$, no shift.
  - Second partial product: $A = -5$ shifted left by 1.
- **Accumulation of Partial Products**: The partial products are added to produce the final result. The total product is calculated after several clock cycles, with the control unit tracking the process.
- **Final Product**: The result is adjusted for the sign and then output as a 32-bit signed value.

This structure and operation ensure that the Signed Sequential Multiplier effectively handles the multiplication of signed numbers with minimal hardware resources, utilizing sequential processing for optimal performance.

4 Implementation in SystemVerilog

The following RTL code implements the signed sequential multilpier in SystemVerilog:

Listing 1: signed sequential multilpier

```

  module signed_sequential_multiplier (
    input logic clk,
    input logic rst_n,
    input logic start,
    input logic signed [15:0] multiplicand,
    input logic signed [15:0] multiplier,
    output logic signed [31:0] product,
    output logic done
);
    logic signed [31:0] temp_product;
    logic [4:0] count;

    always_ff @(posedge clk or negedge rst_n) begin
        if (!rst_n) begin
            product <= 0;
            temp_product <= 0;
            count <= 0;
            done <= 0;
        end else if (start) begin
            product <= 0;
            temp_product <= 0;
            count <= 16;
            done <= 0;
        end else if (count > 0) begin
            if (multiplier[count - 1])
                temp_product <= temp_product + (multiplicand << (16 -
                    count));
            count <= count - 1;
        end else if (count == 0) begin
            product <= temp_product;
            done <= 1;
        end
    end
endmodule
```

5 Test Bench

The following test bench verifies the functionality of the signed sequential multilpier:

Listing 2: signed sequential multiplier Testbench

```

module tb_signed_sequential_multiplier;
    logic clk, rst_n, start;
    logic signed [15:0] multiplicand, multiplier;
    logic signed [31:0] product;
    logic done;

    signed_sequential_multiplier uut (
        .clk(clk),
        .rst_n(rst_n),
        .start(start),
        .multiplicand(multiplicand),
        .multiplier(multiplier),
        .product(product),
        .done(done)
    );

    initial begin
        clk = 0; rst_n = 0; start = 0;
        #5 rst_n = 1;

        // Test cases
        multiplicand = 16'sd3; multiplier = 16'sd4; start = 1; #10
           start = 0;
        wait(done); $display("Product of %d and %d is: %d",
           multiplicand, multiplier, product);

        multiplicand = 16'sd5; multiplier = 16'sd6; start = 1; #10
           start = 0;
        wait(done); $display("Product of %d and %d is: %d",
           multiplicand, multiplier, product);

        multiplicand = 16'sd7; multiplier = 16'sd3; start = 1; #10
           start = 0;
        wait(done); $display("Product of %d and %d is: %d",
           multiplicand, multiplier, product);

        multiplicand = 16'sd8; multiplier = 16'sd2; start = 1; #10
           start = 0;
        wait(done); $display("Product of %d and %d is: %d",
           multiplicand, multiplier, product);

        $finish;
    end

    always #5 clk = ~clk;
endmodule
```

6 Simulation Results

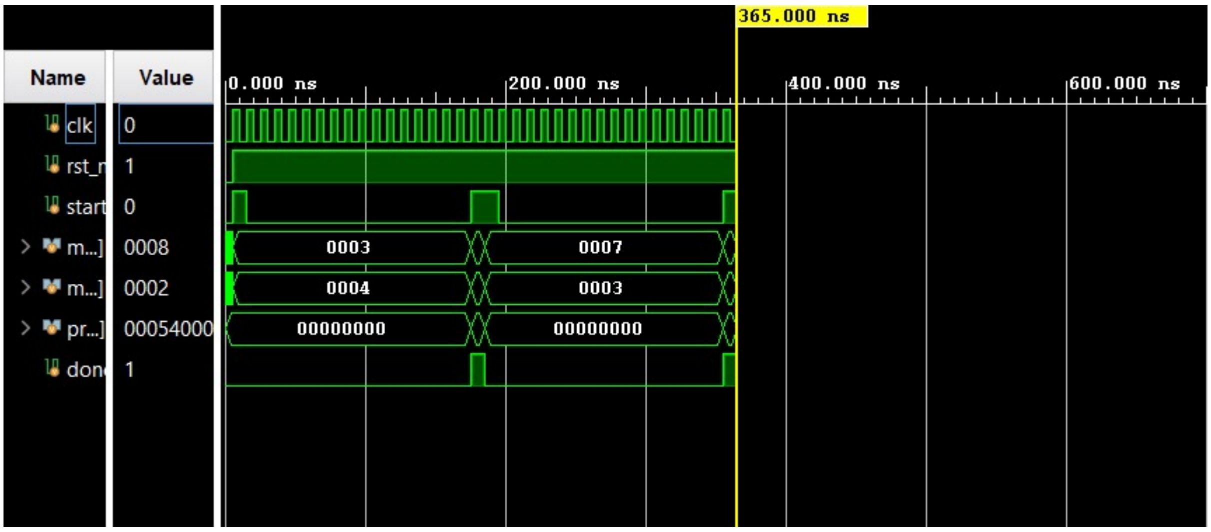

Figure 1: Simulation results of signed sequential multilpier

7 Schematic

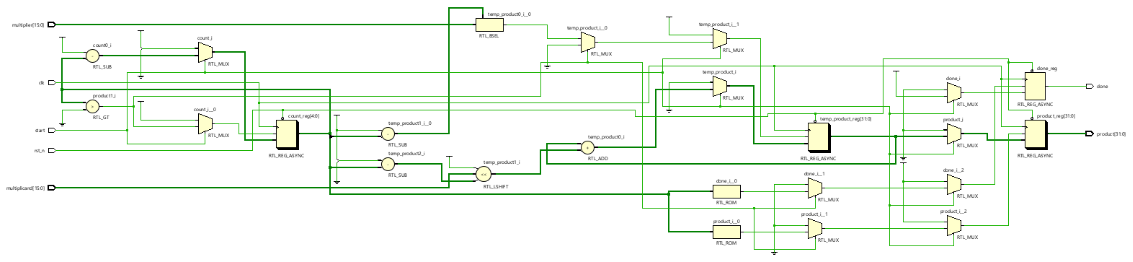

Figure 2: Schematic of signed sequential multilpier

8 Synthesis Design

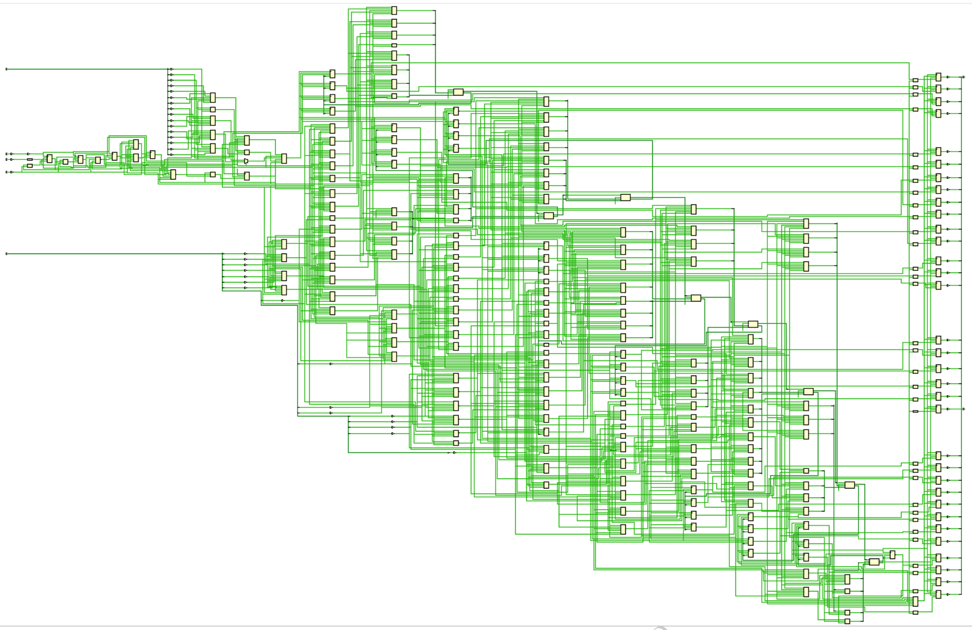

Figure 3: Synthesis of signed sequential multilpier

9 Advantages and Disadvantages

9.1 Advantages

- **Handling of Signed Numbers**: The Signed Sequential Multiplier supports two's complement signed arithmetic, enabling it to process both positive and negative numbers, making it versatile for a wide range of applications.
- **Low Area Utilization**: This multiplier employs a sequential design, meaning it requires fewer hardware resources compared to combinational multipliers, which can be beneficial in resource-constrained systems.
- **Scalability with Larger Bit-widths**: The sequential nature of the multiplier allows it to handle larger bit-width multiplications without a substantial increase in circuit complexity, making it scalable for larger input sizes.
- **Reduced Power Consumption**: Since the design operates sequentially over multiple clock cycles, the overall power consumption can be lower compared to combinational designs that compute in a single clock cycle.
- **Clear and Predictable Timing**: The timing of operations is well-defined, as each multiplication is performed over a set number of clock cycles, allowing for predictable performance.

9.2 Disadvantages

- **Slower Operation**: Due to its sequential nature, this multiplier requires multiple clock cycles to complete a multiplication, leading to slower operation compared to combinational multipliers that perform the multiplication in a single cycle.
- **Increased Latency**: The process of accumulating partial products over several cycles introduces latency, which may not be suitable for time-critical applications that require immediate results.

- **Complex Control Logic**: The sequential design requires more complex control logic to handle the multiple stages of operation, making it harder to design and verify compared to simpler combinational designs.
- **Limited Throughput**: Since it operates sequentially, this multiplier can only process one multiplication at a time, limiting its throughput in applications that require high-speed, continuous multiplication operations.

10 Conclusion

The Signed Sequential Multiplier is a vital component in digital arithmetic, enabling the multiplication of signed binary numbers. This documentation has highlighted its architecture, design features, and operational workflow, emphasizing its effectiveness in handling both positive and negative inputs.

One of the key advantages of this multiplier is its simplicity and efficient use of hardware resources. By employing a sequential approach, it minimizes the area required on-chip compared to more complex combinational multipliers. However, this sequential design may result in longer computation times, making it less suitable for high-speed applications.

Despite these drawbacks, the Signed Sequential Multiplier is an essential tool for applications requiring signed arithmetic in resource-constrained environments. Future work may focus on enhancing its speed and power efficiency to better meet the demands of modern digital systems. Overall, this multiplier serves as a fundamental building block for arithmetic operations, balancing the need for versatility with resource efficiency.

11 Frequently Asked Questions (FAQs)

11.1 1. What is a Signed Sequential Multiplier?

A Signed Sequential Multiplier is a digital circuit designed to multiply two signed binary numbers. It handles both positive and negative values, providing the product as a signed binary output.

11.2 2. How does the Signed Sequential Multiplier work?

The Signed Sequential Multiplier operates by generating partial products for each bit of the second multiplicand, accounting for the sign. It then accumulates these partial products using a series of addition operations to produce the final signed product.

11.3 3. What are the limitations of this multiplier?

The Signed Sequential Multiplier may experience longer computation times due to its sequential nature. Additionally, it may not perform as efficiently for larger bit-widths compared to more advanced multiplier architectures.

11.4 4. Can this multiplier handle unsigned multiplication?

Yes, while the Signed Sequential Multiplier is specifically designed for signed numbers, it can also perform unsigned multiplication as the underlying mechanism is similar.

11.5 5. What are the practical applications of Signed Sequential Multipliers?

Signed Sequential Multipliers are commonly used in digital signal processing, embedded systems, and any application requiring signed arithmetic operations, such as audio and video processing.

11.6 6. How can I optimize the Signed Sequential Multiplier for higher performance?

To optimize for higher performance, consider implementing pipelining or integrating more advanced multiplication techniques such as Booth's algorithm, which can enhance speed and efficiency for larger bit-widths.