Design and Implementation of an 8-bit Multiplier using Booth's Algorithm

Bratin Mondal (21CS10016)
Somya Kumar (21CS30050)

August 30, 2023

1 Introduction

In the realm of digital circuit design, multiplication operations play a vital role in various applications, such as signal processing and arithmetic calculations. The objective of this project is to design and implement an 8-bit multiplier using Booth's Multiplication Algorithm. The Booth's Algorithm is renowned for its efficiency in optimizing the multiplication process by reducing the number of partial products required for the final result. In this report, we present a detailed analysis of our design approach, the individual building blocks of the design, the top-level module, and the verification through a comprehensive testbench.

2 The Booth's Multiplication Algorithm

Algorithm 1 Booth's Multiplication Algorithm

```
1:  procedure BoothMultiplication(Multiplier M, Multiplicand Q)
2:      Initialize Accumulator A to 0
3:      Initialize Booth Register BR to 0
4:      for i = 0 to n − 1 do                ▷ Where n is the number of bits
5:          BR ← (BR ≪ 1) + Q[i]
6:          if BR ends in 01 then
7:              A ← A + M                    ▷ Partial product addition
8:          else if BR ends in 10 then
9:              A ← A − M                    ▷ Partial product subtraction
10:         end if
11:     end for
12: end procedure
```

Booth's Multiplication Algorithm is a technique that streamlines the process of binary multiplication by considering patterns of bits in the multiplier. It decreases the number of addition and subtraction operations required in traditional

multiplication methods. Booth's Algorithm accomplishes this by operating on pairs of adjacent bits in the multiplier, which helps in eliminating redundant calculations. This approach significantly enhances the efficiency of the multiplication process, making it particularly suitable for hardware implementations.

3 Module Descriptions

3.1 Adder Module

The adder module is responsible for performing binary addition on two 8-bit inputs. It accepts two 8-bit numbers as inputs and generates an 8-bit sum as the output. The module employs simple combinational logic to compute the sum of corresponding bits in the inputs, along with handling any carry generated during the addition.

3.2 Subtractor Module

The subtractor module calculates the difference between two 8-bit inputs. It accepts two 8-bit numbers as inputs and produces an 8-bit output representing the result of the subtraction. Similar to the adder, the subtractor utilizes combinational logic to perform bit-wise subtraction while managing borrow propagation.

3.3 Shifter Module

The shifter module plays a pivotal role in Booth's Algorithm by facilitating the leftward shift of input values. It accepts an 8-bit input and shifts the bits one position to the left. The module is essential for the algorithm's functioning as it allows the manipulation of the input values based on the algorithm's prescribed shifting steps.

3.4 Booth Multiplier Module

The Booth multiplier module is the centerpiece of our design, orchestrating the operations of the adder, subtractor, and shifter modules. The module accepts two 8-bit inputs - the multiplier and the multiplicand - along with control signals including clock, reset, and load. By adhering to the steps outlined in Booth's Algorithm, the multiplier module coordinates the submodules to efficiently calculate the 16-bit product output.

4 Top-Level Module and Interconnections

The top-level module, `booth_mult`, serves as the integration hub for the submodules and control logic. It encapsulates the adder, subtractor, and shifter modules, orchestrating their activities to implement Booth's Algorithm. The module accepts inputs from the testbench, processes these inputs using the

algorithmic steps, and produces the final 16-bit product output. By interconnecting these modules, we create a cohesive design that accurately implements the multiplication algorithm.

5 Test Bench Usage

The testbench, `tb_bsa`, forms an essential component of our design verification process. Its purpose is to thoroughly test the functionality of the Booth multiplier. It supplies input values (num1 and num2) to the multiplier module and monitors the corresponding output product. Through the manipulation of clock signals, reset signals, and load signals, the testbench simulates various scenarios, thereby verifying the correctness and robustness of our design.

6 Conclusion

In conclusion, the design and implementation of an 8-bit multiplier using Booth's Algorithm exemplify the synergy of algorithmic understanding, digital circuit design, and efficient hardware utilization. Through the careful integration of adder, subtractor, and shifter submodules, we created a Booth multiplier module that executes the algorithm with precision. Our testbench, designed to encompass diverse scenarios, ensures the reliability and functionality of our multiplier design. This project underscores the significance of algorithm-driven hardware design in achieving optimized computation in digital circuits.