Lab Report

On

Multiplication of Two Unsigned Integer Binary Numbers

Computer Organization and Architecture

By

Spandan Guragain

PUR078BCT086

To

Mr. Sujan Karki

Tribhuwan University

Institute of Engineering

Faculty of Computer and Eletronics Engineering

Purwanchal Campus

Dharan, Nepal

2024-05-26

1 Multiplication of Two Unsigned Integer Binary Numbers

1.1 Introduction

The multiplication of binary numbers is a fundamental operation in digital systems, forming the basis of arithmetic and logical operations in computer architecture. This report details the design and implementation of a hardware module to multiply two 4-bit unsigned binary numbers using the partial-product method in SystemVerilog, along with a testbench to verify its functionality.

1.2 Objective

To design an unsigned binary multiplier using the partial-product method in SystemVerilog and understand its workings.

1.3 Theory

The partial-product method for binary multiplication follows the following process:

1. Multiply each bit of the multiplier by the multiplicand to generate partial products.
2. Align the partial products according to their bit positions and sum all of them.

For 4-bit multiplication, the final product will be an 8-bit number.

1.4 Design

The `UnsignedMultiplier` module takes two 4-bit inputs, A and B, and produces an 8-bit product.

```
module UnsignedMultiplier(
  input [3:0] A, // 4-bit input A (multiplicand)
  input [3:0] B, // 4-bit input B (multiplier)
  output [7:0] Product // 8-bit output for the product
);

  // Internal signal for the product
  reg [7:0] partial_products [3:0]; // Array to store partial products
  integer i;

  // Generate partial products
  always @(*) begin
    for (i = 0; i < 4; i = i + 1) begin
      if (B[i])
        partial_products[i] = A << i; // Shift A left by i positions
      else
        partial_products[i] = 8'b00000000; // Zero if B[i] is 0
    end
  end

  // Sum the partial products
  assign Product = partial_products[0] +
                   partial_products[1] +
                   partial_products[2] +
                   partial_products[3] ;

endmodule
```

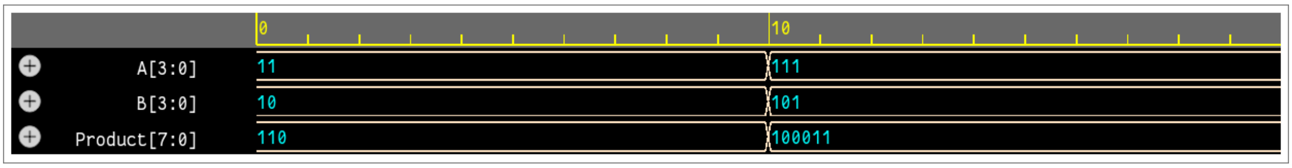
1.5 Testbench

The testbench module is used to apply various test cases to the `UnsignedMultiplier` module and verify the correctness of its outputs.

```
module testbench;
    // Inputs
    reg [3:0] A;
    reg [3:0] B;

    // Outputs
    wire [7:0] Product;

    UnsignedMultiplier dut (A, B, Product);

    initial begin
        $dumpfile("dump.vcd");
        $dumpvars(1);

        // Test case 1: A = 3, B = 2
        A = 4'b0011;
        B = 4'b0010;
        #10; // Wait for 10 time units
        // Display the output in the log
        $display(
            "1: A = %4b, B = %4b, Product = %8b", A, B, Product
        );

        // Test case 2: A = 7, B = 5
        A = 4'b0111;
        B = 4'b0101;
        #10; // Wait for 10 time units
        // Display the output in the log
        $display(
            "2: A = %4b, B = %4b, Product = %8b", A, B, Product
        );
    end
endmodule
```

1.6 Results

The simulation was conducted in EDA Playground. The results were as follows:



Output:

```
1: A = 0011, B = 0010, Product = 00000110
2: A = 0111, B = 0101, Product = 00101111
```

The results are the expected outputs.

1.7 Conclusion

The `UnsignedMultiplier` module correctly multiplies two unsigned 4-bit binary numbers using the partial-product method.