Project 31: Reduced Area Multiplier

A Comprehensive Study of Advanced Digital Circuits

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

Documentation Specialist: Dhruv Patel & Nandini Maheshwari

Contents

1 Project Overview

A Reduced Area Multiplier is a type of digital multiplier circuit designed to optimize hardware resources by minimizing the number of logic gates and interconnections required. This optimization is especially critical in applications where silicon area, power consumption, and cost need to be minimized, such as in embedded systems, mobile devices, and low-power processors. Traditional multipliers, like Wallace tree or Dadda tree multipliers, focus on speed but often at the cost of increased area. In contrast, reduced area multipliers prioritize compactness by employing efficient algorithms and techniques, such as partial product reduction and simplified carry propagation, without sacrificing significant performance. This balance between area and speed makes reduced area multipliers a popular choice in hardware-efficient designs.

2 Reduced Area Multiplier

2.1 Key Concept of Reduced Area Multiplier

- **Partial Product Generation:** In any binary multiplier, the product is formed by multiplying individual bits of the operands, which creates partial products. For an n-bit multiplier, n^2 partial products are generated. These partial products must then be summed to obtain the final result.
- **Partial Product Reduction:** Reduced area multipliers focus on minimizing the hardware required for this summation process. Several approaches have been developed to efficiently reduce the number of logic gates used in combining partial products.
- **Reduced Depth of the Multiplier Tree:** Reduced area multipliers aim to keep the logic depth (or levels of computation) shallow to balance area efficiency and propagation delay. Techniques like Sklansky's algorithm (a divide-and-conquer method for parallel prefix adders) can also be used to reduce the depth and, by extension, the area.
- **Trade-off Between Area, Speed, and Power:** The main objective of reduced area multipliers is to achieve a compact design, but this often results in a trade-off with speed. These designs usually sacrifice some performance to gain area efficiency. However, in low-power applications, this trade-off is acceptable, as reducing the area also reduces the power consumption due to fewer switching gates.

2.2 Working of Reduced Area Multiplier

1. Inputs:

- Two 8-bit inputs, A and B, represent the operands to be multiplied.
- The result of the multiplication will be stored in a 16-bit output product.

2. Partial Product Generation: The partial products are calculated in the loop. For each bit of A (from A[0] to A[7]):

- If A[i] is 1, the code shifts the value of B left by i positions, effectively multiplying B by 2^i. This operation produces a partial product corresponding to that bit position.
- If A[i] is 0, the corresponding partial product is set to 0 (no contribution from this bit).

3. Summing the Partial Products: After generating each partial product, the code immediately adds it to a running sum (sum). This cumulative sum represents the intermediate result of the multiplication.

4. Final Product: After all 8 bits of A are processed, the total sum (which is the final multiplication result) is assigned to the output product.

2.3 RTL Code

Listing 1: Reduced Area Multiplier

```
• \end{itemize}


module reduced_area_multiplier (
    input  logic [7:0] A,  // 8-bit input A
    input  logic [7:0] B,  // 8-bit input B
    output logic [15:0] product  // 16-bit product output
);
    logic [15:0] partial_products[7:0];
    logic [15:0] sum;

    always_comb begin
        sum = 0;
        for (int i = 0; i < 8; i++) begin
            partial_products[i] = A[i] ? (B << i) : 0;
            sum = sum + partial_products[i];
        end
        product = sum;
    end
endmodule
```


2.4 Testbench

Listing 2: Reduced Area Multiplier

```


module tb_reduced_area_multiplier;
    logic [7:0] A, B;
    logic [15:0] product;

    reduced_area_multiplier uut (.A(A), .B(B), .product(product));

    initial begin
        // Test case 1
        A = 8'b00011010; // 26
        B = 8'b00000101; // 5
        #10;
        $display("A = %d, B = %d, Product = %d", A, B, product);

        // Test case 2
        A = 8'b11111111; // 255
        B = 8'b00000010; // 2
        #10;
        $display("A = %d, B = %d, Product = %d", A, B, product);

        $finish;
    end
endmodule
```

3 Results

3.1 Simulation

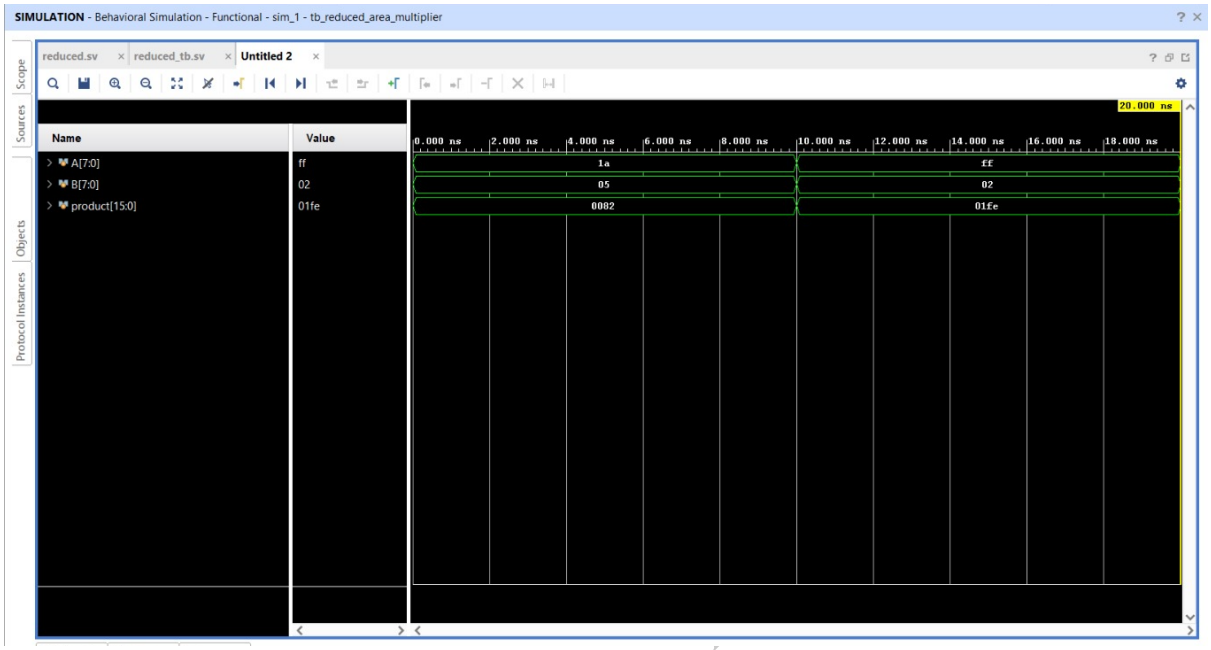

Figure 1: Simulation of Reduced Area Multiplier

3.2 Schematic

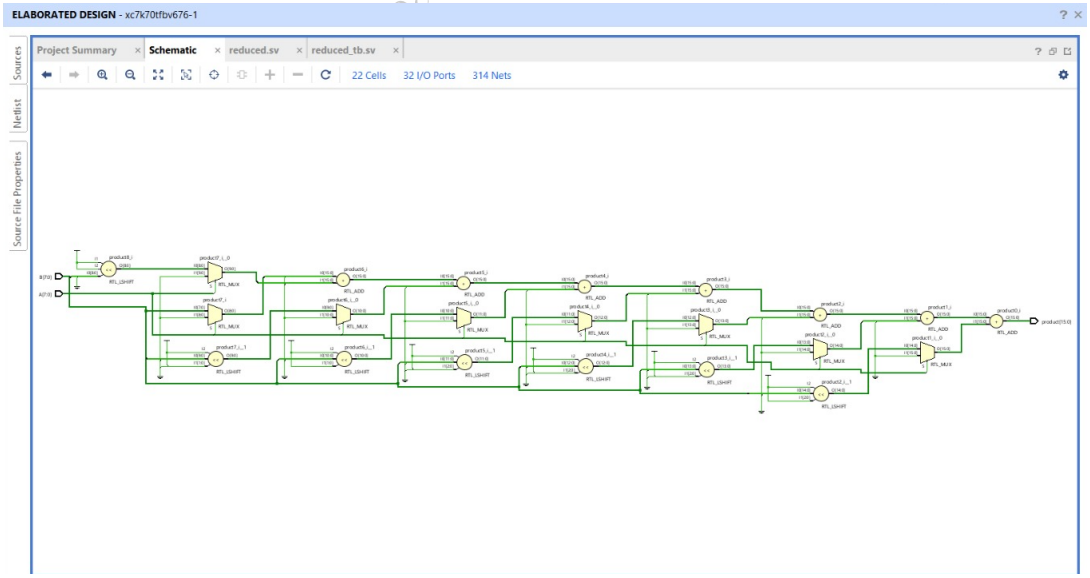

Figure 2: Schematic of Reduced Area Multiplier

3.3 Synthesis Design

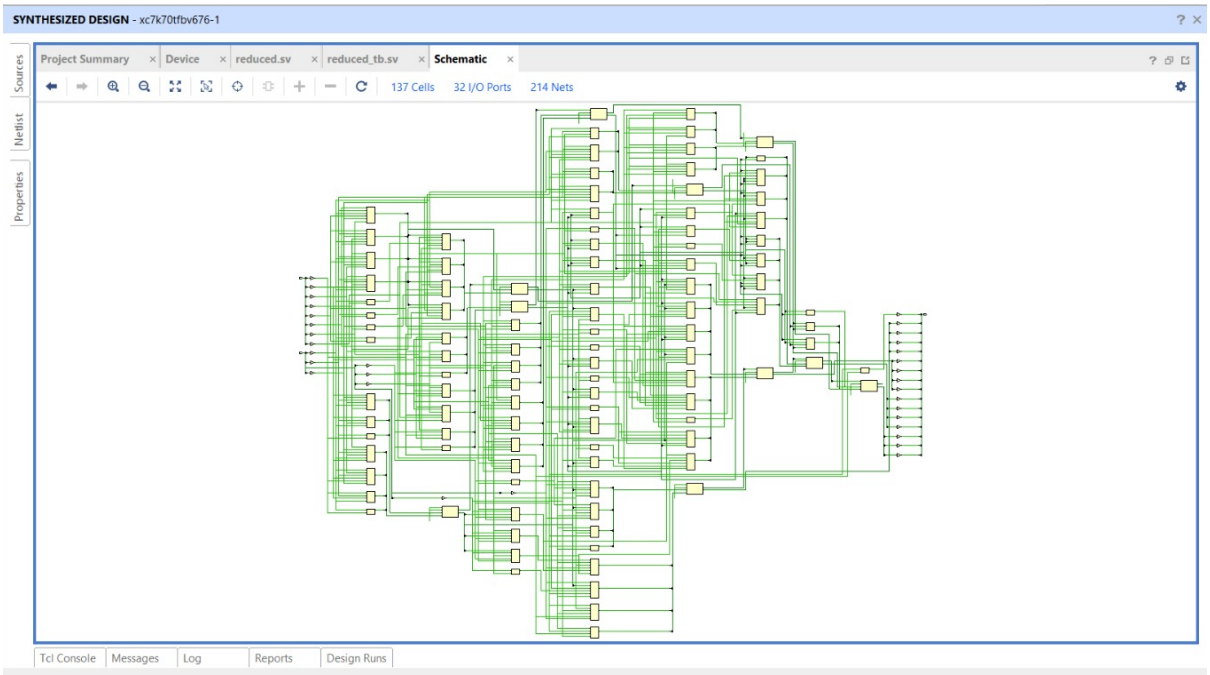

Figure 3: Synthesis Design of Reduced Area Multiplier

4 Advantages of Reduced Area Multiplier

- **Lower Hardware Cost:**By minimizing the number of logic gates and interconnections, reduced area multipliers require fewer hardware resources, making them cost-effective for silicon-based designs.
- **Lower Power Consumption:**Fewer gates and reduced complexity result in lower power consumption, which is essential in battery-operated devices and low-power applications like mobile processors and IoT devices.
- **Smaller Chip Area:**The primary goal is to minimize the circuit area, making it ideal for applications with strict area constraints, such as embedded systems, where available space on the chip is limited.
- **Simplified Design:**Reduced area multipliers rely on basic techniques like shift-and-add operations or simple partial product summation, which are easy to implement and debug.
- **Suitable for Low-Speed Applications:**These multipliers are efficient for applications that do not require high-speed processing, where area and power efficiency are more critical than performance.

5 Disadvantages of Reduced Area Multiplier

- **Slower Speed:**Reduced area multipliers often use simpler adders (e.g., ripple-carry adders) and sequential addition processes, which increase the overall latency. They are generally slower compared to high-performance multipliers like Wallace trees or Kogge-Stone adders.
- **Increased Propagation Delay:**The sequential summation of partial products can introduce significant propagation delay, especially for larger operands. This can be a limiting factor in time-sensitive applications like real-time processing.
- **Limited Performance:**For high-performance applications, such as digital signal processing (DSP) or high-frequency operations, reduced area multipliers may not be suitable due to their lower throughput.
- **Trade-off with Accuracy:**Some techniques used in reduced area multipliers, such as truncated multiplication or approximate multiplication, may introduce small errors or reduce precision to save hardware. This may not be acceptable in applications requiring exact results.
- **Less Optimization for Carry Handling:**Reduced area multipliers typically do not include sophisticated techniques for efficient carry propagation (like in carry-lookahead adders), which leads to increased carry delays and slower addition stages.

6 Applications of Reduced Area Multiplier:

- **Embedded Systems:**Used in microcontrollers and embedded processors where area and power efficiency are crucial due to limited hardware resources.
- **Mobile Devices:**Utilized in smartphones, tablets, and wearable devices to optimize battery life by minimizing power consumption and chip area.
- **Internet of Things (IoT):**Deployed in IoT devices where low-cost, low-power, and small form factor designs are necessary for mass deployment.
- **Low-Power Signal Processing:**Employed in digital signal processing (DSP) systems for applications like audio processing, image processing, and sensor data analysis, where moderate speed and area efficiency are prioritized.
- **Battery-Operated Devices:**Ideal for devices running on batteries (e.g., hearing aids, fitness trackers) where power consumption needs to be minimized.

7 Summary

A Reduced Area Multiplier is an optimized digital multiplier that minimizes the circuit area by focusing on efficient partial product generation and reduction techniques, often at the cost of speed. Through various algorithms and architectural modifications like Booth encoding, carry-save adders, and compressor trees, reduced area multipliers strike a balance between area efficiency, power consumption, and performance. They are especially useful in systems with tight constraints on area and power, making them a key component in modern low-power hardware design.

8 FAQs

1. What is a reduced area multiplier?

A reduced area multiplier is a digital multiplier that optimizes for minimal hardware area by reducing the number of logic gates and interconnections, often at the cost of speed.

2. How does a reduced area multiplier differ from a traditional multiplier?

It focuses on minimizing circuit area and power consumption, while traditional multipliers (like Wallace tree) prioritize speed, often using more complex architectures and occupying more area.

3. What techniques are used to reduce the area in multipliers?

Techniques include using simple adders like ripple-carry adders, partial product reduction methods like modified Booth encoding, and truncating less significant bits to save on logic gates.

4. Why is area reduction important in multiplier design?

Reduced area minimizes silicon usage, lowers manufacturing costs, and decreases power consumption, making the design more efficient for embedded, mobile, and IoT applications.

5. What are partial products in a multiplier?

Partial products are intermediate results generated by multiplying individual bits of the multiplier with the multiplicand. These partial products are then summed to produce the final product.

6. How does a carry-save adder (CSA) help in reduced area multipliers?

CSAs delay carry propagation until the final stage, allowing for simpler hardware implementation and reducing the number of logic gates required for intermediate addition.

7. What is Booth encoding, and how does it help in area reduction?

Booth encoding reduces the number of partial products by grouping bits of the multiplier and encoding them in a way that fewer multiplication operations are needed, leading to area savings.

8. What are the trade-offs involved in using a reduced area multiplier?

The main trade-off is speed; reduced area multipliers are slower compared to high-performance multipliers because they use simpler adders and delay carry propagation.

9. In which applications are reduced area multipliers most commonly used?

They are commonly used in embedded systems, IoT devices, mobile devices, and other low-power applications where space and power efficiency are more important than high speed.

10. How do truncated multipliers reduce area?

Truncated multipliers ignore the least significant bits during partial product summation, reducing the number of required additions and saving hardware at the cost of slight accuracy loss.