Project 32: Truncated 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 truncated multiplier is a type of digital multiplier that reduces hardware complexity and power consumption by selectively omitting less significant partial products during the multiplication process. Unlike a conventional multiplier that computes all partial products, a truncated multiplier discards or approximates some of the least significant bits, achieving faster computation with reduced area and power usage at the cost of minimal accuracy loss. This technique is commonly employed in applications where high precision is not critical, such as digital signal processing (DSP), image processing, and low-power embedded systems. The trade-off between performance and precision can be fine-tuned depending on specific design requirements.

2 Truncated Multiplier

2.1 Basic Concept of Truncated Multiplier

In a conventional binary multiplier, all partial products are computed based on each bit of the multiplier and multiplicand. These partial products are then summed to produce the final product. The size of the product is the sum of the bit widths of the two operands (e.g., an 8x8 multiplication generates a 16-bit product).

In contrast, a truncated multiplier reduces the total number of partial products by omitting or approximating the less significant bits. This leads to a product that is slightly smaller or less precise but can be computed more efficiently. There are two main techniques to achieve this:

- **Omitting partial products:** The least significant bits of the partial product matrix are simply not computed.
- **Approximating partial products:** Instead of ignoring the lower bits completely, a carefully designed approximation can be used to reduce the error introduced by truncation.

2.2 Architecture of Truncated Multiplier

The architecture of a truncated multiplier generally includes the following elements:

- **Partial Product Generation:** Similar to a conventional multiplier, this stage generates partial products for each bit of the multiplier. However, in the truncated multiplier, only a subset of these partial products is selected based on the truncation method.
- **Truncation Logic:** This logic determines which partial products will be included in the final sum and which will be omitted. In some cases, the omitted partial products are replaced with approximations to reduce the truncation error.
- **Reduction Tree:** The selected partial products are summed using adders (e.g., a Wallace tree or Dadda tree) to generate the final product. The size of the reduction tree is reduced compared to a conventional multiplier due to the lower number of partial products.
- **Error Compensation Circuit:** If error compensation is employed, additional logic is added to the higher-order bits to adjust the result, reducing the impact of truncation on the final output.

2.3 Working of Truncated Multiplier

Partial Product Generation:

- In a conventional binary multiplier, partial products are generated by multiplying each bit of the multiplicand by each bit of the multiplier.
- In a truncated multiplier, only a subset of the partial products (typically the most significant ones) is selected for computation.

Truncation of Partial Products:

- Omission of Partial Products: The least significant partial products are ignored entirely, reducing the number of rows in the partial product matrix.
- Approximating Partial Products: Instead of ignoring, some partial products can be approximated to minimize the error while still reducing complexity.

Reduction of Partial Products:

- The selected partial products are summed using a reduction tree (e.g., Wallace tree or Dadda tree).
- Since fewer partial products are considered, the reduction stage is faster and uses fewer adders, reducing hardware complexity and power consumption.

Error Compensation :

- **Constant Bias Addition:** A small constant value is added to the most significant bits to compensate for the omitted partial products.

Statistical Correction: The design uses statistical methods to approximate the effect of the omitted bits.

Adaptive Error Correction: Dynamic correction is applied based on the inputs or application-specific requirements.

Final Product Formation:

- The sum of the remaining partial products gives the final product.
- Any error compensation is applied, if necessary, to reduce the impact of the truncation on the final result.

Error Analysis:

- The truncated result has a slight error compared to the full-precision result.
- The error is a function of the number of truncated partial products and the chosen compensation method.

2.4 RTL Code

Listing 1: Truncated Multiplier

```
1
2
3  module truncated_multiplier #(parameter N = 8, M = 4)(
4      input logic [N-1:0] A, // N-bit input A
5      input logic [N-1:0] B, // N-bit input B
6      output logic [(2*N)-1:0] truncated_prod // Output truncated product
7  );
8
9      logic [(2*N)-1:0] full_prod; // Full product of A * B
10
11     // Perform full multiplication
12     assign full_prod = A * B;
13
14     // Truncate the result by keeping only the higher (N+M)-bits of
           the result
15     assign truncated_prod = full_prod[(2*N)-1:(N+M)];
16
17 endmodule
```

2.5 Testbench

Listing 2: Truncated Multiplier

```
1
2
3  module tb_truncated_multiplier;
4
5      parameter N = 8;
6      parameter M = 4;
7
8      logic [N-1:0] A;
9      logic [N-1:0] B;
10     logic [(2*N)-1:0] truncated_prod;
11
12     // Instantiate the truncated multiplier
13     truncated_multiplier #(N, M) uut (
14         .A(A),
15         .B(B),
16         .truncated_prod(truncated_prod)
17     );
18
19     initial begin
20         // Test case 1
21         A = 8'd15; // 15 * 10
22         B = 8'd10;
23         #10;
24
25         // Test case 2
26         A = 8'd20; // 20 * 20
27         B = 8'd20;
28         #10;
29
30         // Test case 3
31         A = 8'd25; // 25 * 30
32         B = 8'd30;
33         #10;
34
35         // Test case 4
36         A = 8'd100; // 100 * 50
37         B = 8'd50;
38         #10;
39
40         $finish;
41     end
42
43 endmodule
```

3 Results

3.1 Simulation

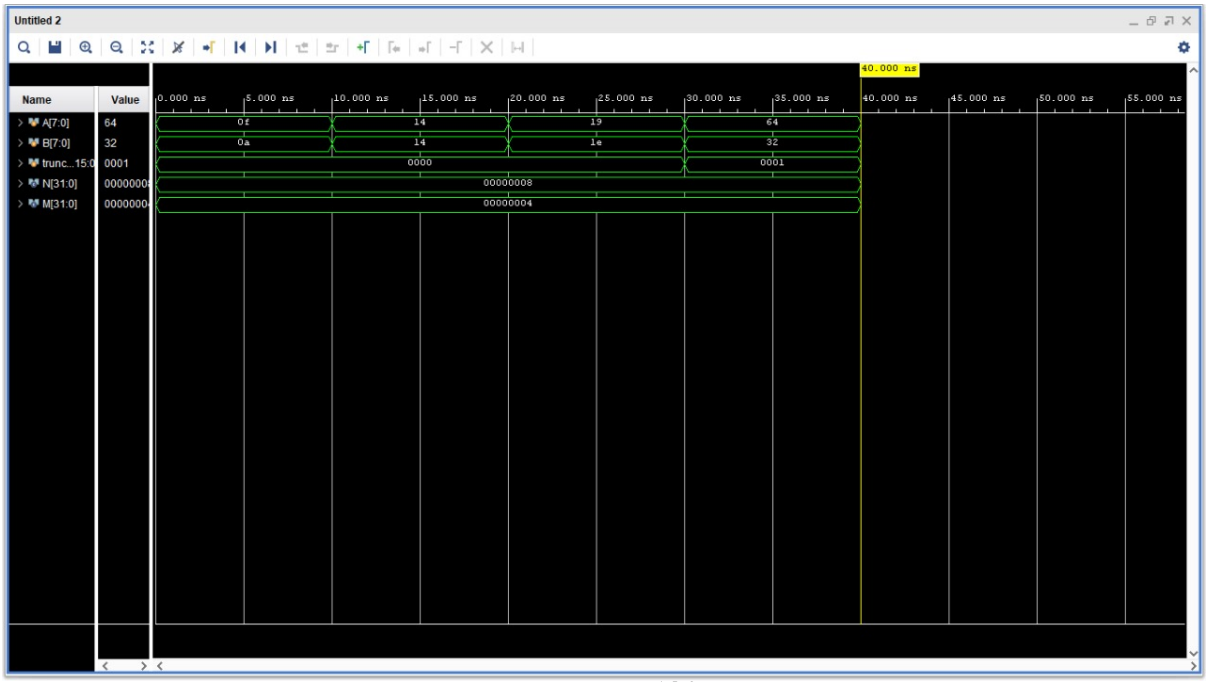

Figure 1: Simulation of Truncated Multiplier

3.2 Schematic

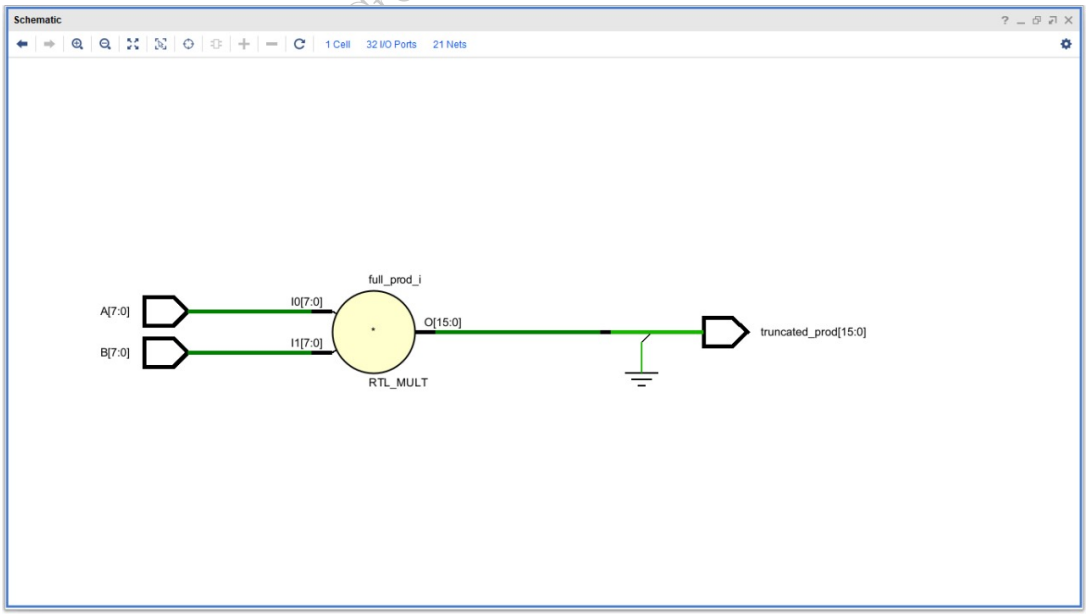

Figure 2: Schematic of Truncated Multiplier

3.3 Synthesis Design

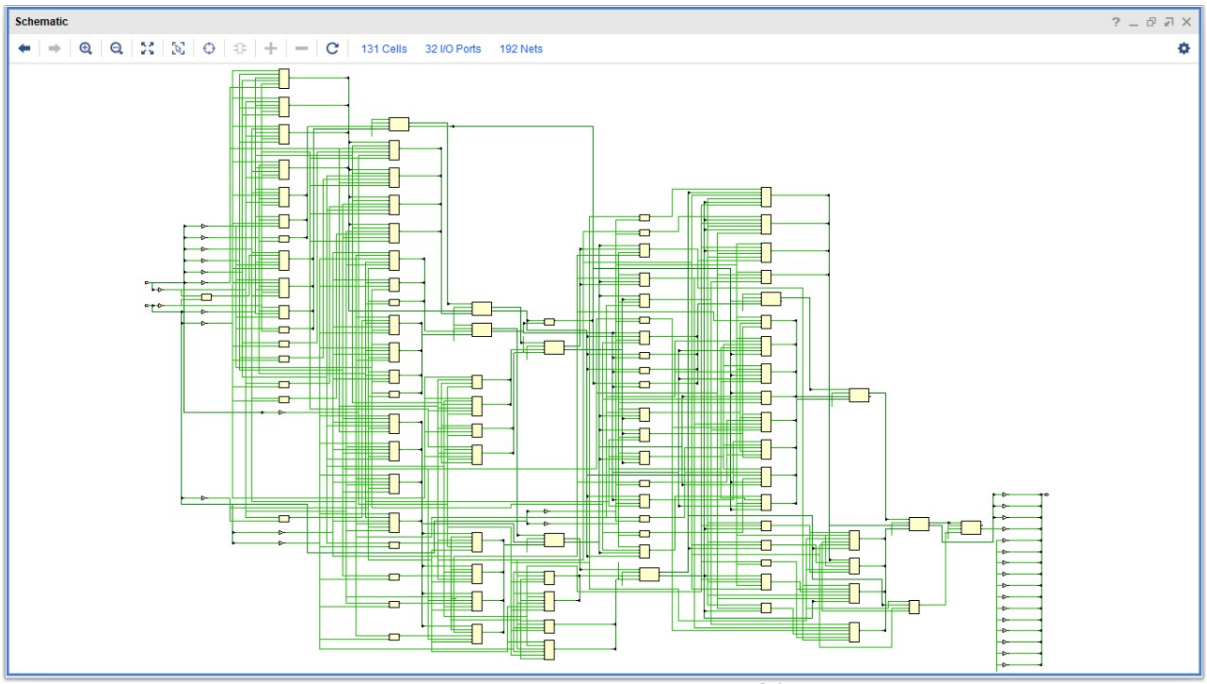

Figure 3: Synthesis Design of Truncated Multiplier

4 Advantages of Truncated Multiplier

- **Reduced Power Consumption:** By eliminating some of the partial products and reducing the number of adders, the overall power consumption is significantly lower.
- **Smaller Area:** Fewer partial products and adders result in less silicon area used, which is crucial for applications requiring compact designs.
- **Faster Computation:** Since fewer operations are needed, the delay in the critical path is reduced, leading to faster multiplication.
- **Design Flexibility:**Truncation allows designers to fine-tune the balance between precision and hardware efficiency, enabling custom designs that optimize for specific application requirements.
- **Reduced Delay:**Since fewer partial products need to be summed, the propagation delay is reduced, making the design more time-efficient.

5 Disadvantages of Truncated Multiplier

- **Loss of Precision:**Truncation introduces a small error in the final result, as the least significant partial products are ignored or approximated. This may be unacceptable for applications requiring high precision.
- **Error Management:**The error introduced by truncation needs to be managed carefully, especially in sensitive applications. This may require additional error compensation techniques, which could increase the design complexity.
- **Limited Use in Precision-Critical Applications:**Truncated multipliers are not suitable for applications that demand exact or high-precision results, such as scientific computing or cryptography.

- **Design Complexity in Compensation:**Implementing error compensation methods (such as constant bias addition or adaptive correction) adds complexity to the design, potentially offsetting some of the benefits in terms of simplicity and power savings.

- **Non-Uniform Error Distribution:**The error introduced by truncation is often non-uniform across different input combinations, leading to variable levels of accuracy depending on the specific values being multiplied.

6 Conclusion

Truncated multipliers are a powerful design technique that balances hardware efficiency with acceptable levels of precision loss. By selectively eliminating or approximating the least significant bits, they offer a way to improve performance and reduce area and power consumption in systems where full precision is not required. These trade-offs make them ideal for specific high-performance, low-power applications, but careful design and error management are essential to ensure that the reduced precision does not significantly affect the overall system performance.

7 FAQs

1. What is a truncated multiplier?
A truncated multiplier is a digital multiplier that reduces hardware complexity, power consumption, and delay by omitting or approximating the least significant partial products, trading off some precision for improved efficiency.

2. How does truncation reduce hardware complexity in multipliers?
Truncation reduces the number of partial products generated and summed, which decreases the number of adders and logic gates required, leading to a simpler and more efficient design.

3. What is the main advantage of using a truncated multiplier?
The main advantage is reduced power consumption and area due to the smaller number of partial products and adders, making the design faster and more resource-efficient.

4. What is the trade-off involved in using a truncated multiplier?
The trade-off is between precision and hardware efficiency. Truncated multipliers sacrifice some accuracy in exchange for reduced power, area, and faster operation.

5. In which applications are truncated multipliers commonly used?
Truncated multipliers are commonly used in applications where a small error is tolerable, such as digital signal processing (DSP), image processing, and low-power embedded systems.

6. What is the effect of truncation on the final result?
Truncation introduces a small error in the final result because the least significant partial products are omitted or approximated.

7. What are the techniques used to compensate for the error in truncated multipliers?
Common techniques include constant bias addition, statistical correction, and adaptive error compensation, which help reduce the truncation error.

8. Why is power consumption lower in a truncated multiplier?
Power consumption is lower because fewer partial products are generated and summed, reducing the number of active logic gates and adders, which leads to reduced dynamic power usage.

9. What is the difference between a truncated multiplier and a conventional multiplier?
A conventional multiplier computes all partial products for full precision, whereas a truncated multiplier skips or approximates the less significant partial products to save hardware and improve efficiency.

10. Can truncated multipliers be used in high-precision applications?

No, truncated multipliers are typically not suitable for high-precision applications like scientific computing or cryptography, where exact results are crucial.

11. How does truncation affect the speed of the multiplier?

Truncation increases the speed of the multiplier by reducing the number of partial products that need to be computed and summed, shortening the critical path.

12. What is partial product omission in a truncated multiplier?

Partial product omission refers to the process of ignoring or discarding the least significant partial products during the multiplication process to reduce the overall complexity.

13. How is the final product obtained in a truncated multiplier?

The final product is obtained by summing the selected partial products and, optionally, applying an error compensation technique to reduce the error caused by truncation.

14. What factors determine the level of truncation in a multiplier design?

The level of truncation depends on the precision requirements, power constraints, and performance targets of the application.

15. What are the limitations of truncated multipliers?

Truncated multipliers are limited by the accuracy loss introduced by omitting partial products, making them unsuitable for applications that require exact results.