Project 48: Redundant binary 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

The project focuses on designing a redundant binary multiplier, a digital circuit that efficiently multiplies two binary numbers using a redundant binary representation to reduce carry propagation delays. The objective is to create a WIDTH-bit multiplier that produces a 2*WIDTH-bit output while optimizing for speed and resource utilization in FPGA or ASIC implementations. The design involves generating partial products for each bit of the multiplier and utilizing carry-save addition techniques to sum these products effectively. Verification will be conducted through simulation with various test cases to ensure functionality, followed by potential hardware testing. This multiplier has applications in digital signal processing, graphics processing, high-performance computing, and embedded systems, making it a vital component for enhancing computational efficiency in modern digital systems.

2 Redundant binary multiplier

2.1 Basic Concept of Redundant binary multiplier

A redundant binary multiplier utilizes redundant binary representation to efficiently perform multiplication by reducing carry propagation delays. It generates partial products for each bit of the multiplier and employs carry-save addition to combine these products without waiting for carries, allowing for faster computations. This method is particularly advantageous in applications such as digital signal processing, graphics processing, and high-performance computing, where efficient arithmetic operations are essential. The redundancy in the binary representation enables multiple ways to express the same value, enhancing computational speed and efficiency.

2.2 Architecture of Redundant binary multiplier

Input Stage

1. Two inputs, A and B, represented in redundant binary format.
2. Bit-width defined by the parameter WIDTH.

Partial Product Generation

1. Generates partial products for each bit of the multiplier B.
2. Each bit B[i] multiplies the multiplicand A based on its value (0 or 1).

Carry-Save Adder (CSA) Stage

1. Sum Array: Stores intermediate sums of the partial products.
2. Carry Array: Stores carry values resulting from addition.
3. Uses a carry-save adder to combine partial products without waiting for carry propagation.

Final Adder Stage

1. Combines the results from the CSA stage.
2. Uses a conventional adder (e.g., ripple-carry adder) to produce the final output.

Output Stage

1. Output P has a width of 2*WIDTH bits to accommodate the maximum product.

2.3 Working of Redundant binary multiplier

The redundant binary multiplier operates by efficiently multiplying two binary numbers using redundant binary representation and carry-save addition.

- **Input Representation:** Two operands, A and B, are provided in redundant binary format, allowing multiple representations for the same value.
- **Partial Product Generation:** For each bit of the multiplier B, corresponding partial products are generated by multiplying A with B[i], producing WIDTH partial products.
- **Carry-Save Addition (CSA):** Instead of traditional addition, carry-save addition is used to combine partial products. A sum array and carry array are maintained, allowing for efficient addition without waiting for carry propagation.
- **Final Combination:** After processing all partial products, a conventional adder combines the final sums and carries to produce the output.
- **Output Generation:** The final product P is generated, with a bit-width of 2*WIDTH to accommodate the maximum possible result.

2.4 RTL Code

Listing 1: Redundant binary multiplier

```
1
2  module redundant_binary_multiplier #(parameter WIDTH = 4) (
3      input  logic [WIDTH-1:0] A, B,
4      output logic [2*WIDTH-1:0] P
5  );
6      logic [WIDTH-1:0] sum[WIDTH-1:0]; // Array to store intermediate
           sums
7      logic [WIDTH:0] carry[WIDTH-1:0]; // Array to store carry for each
           stage
8
9      always_comb begin
10         P = 0;
11         for (int i = 0; i < WIDTH; i++) begin
12             {carry[i], sum[i]} = A * B[i]; // Partial product
                   (carry-save)
13             P = P + (sum[i] << i) + (carry[i] << (i+1)); // Accumulate
                   shifted partial products
14         end
15     end
16 endmodule
```

2.5 Testbench

Listing 2: Redundant binary multiplier

```
1
2  module tb_redundant_binary_multiplier;
3      parameter WIDTH = 4;
4      logic [WIDTH-1:0] A, B;
5      logic [2*WIDTH-1:0] P;
6
7      // Instantiate the multiplier
8      redundant_binary_multiplier #(WIDTH) uut (
9          .A(A),
10         .B(B),
11         .P(P)
12     );
13
14     initial begin
15         // Test case 1
16         A = 4'b1011; B = 4'b1101; // A = 11, B = 13
17         #10;
18         $display("A = %0d, B = %0d, P = %0d", A, B, P);
19
20         // Test case 2
21         A = 4'b1110; B = 4'b1010; // A = 14, B = 10
22         #10;
23         $display("A = %0d, B = %0d, P = %0d", A, B, P);
24
25         // Test case 3
26         A = 4'b0101; B = 4'b0110; // A = 5, B = 6
27         #10;
28         $display("A = %0d, B = %0d, P = %0d", A, B, P);
29
30         $finish;
31     end
32 endmodule
```

3 Results

3.1 Simulation

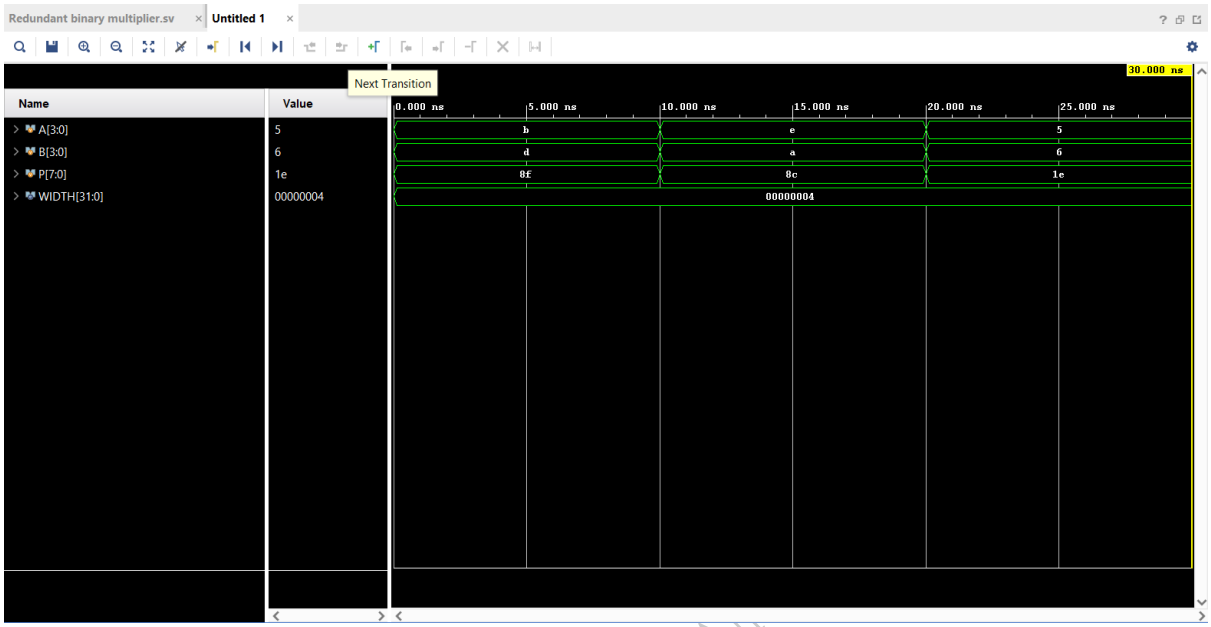

Figure 1: Simulation of Redundant binary multiplier

3.2 Schematic

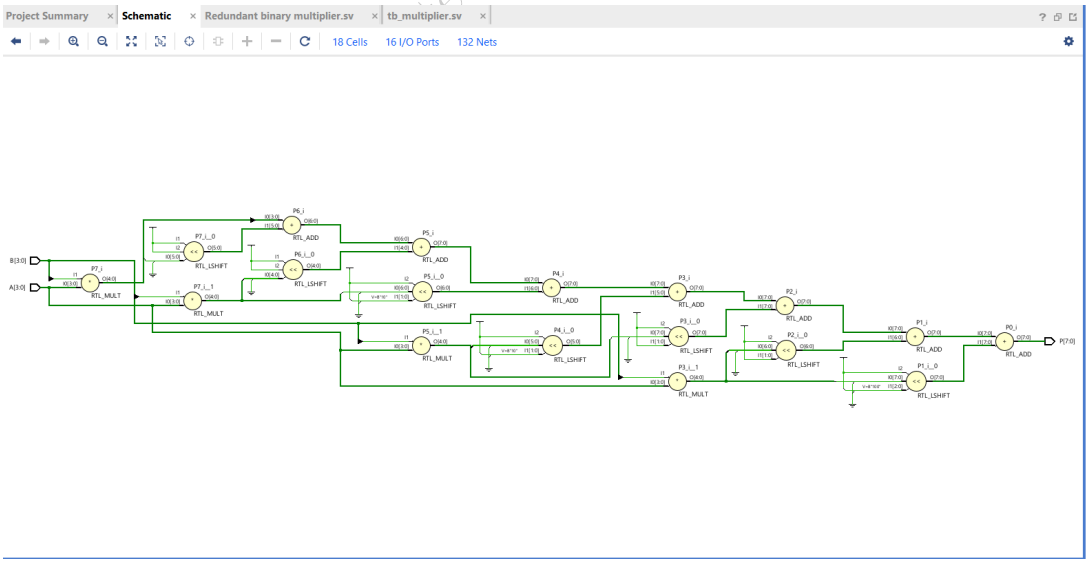

Figure 2: Schematic of Redundant binary multiplier

3.3 Synthesis Design

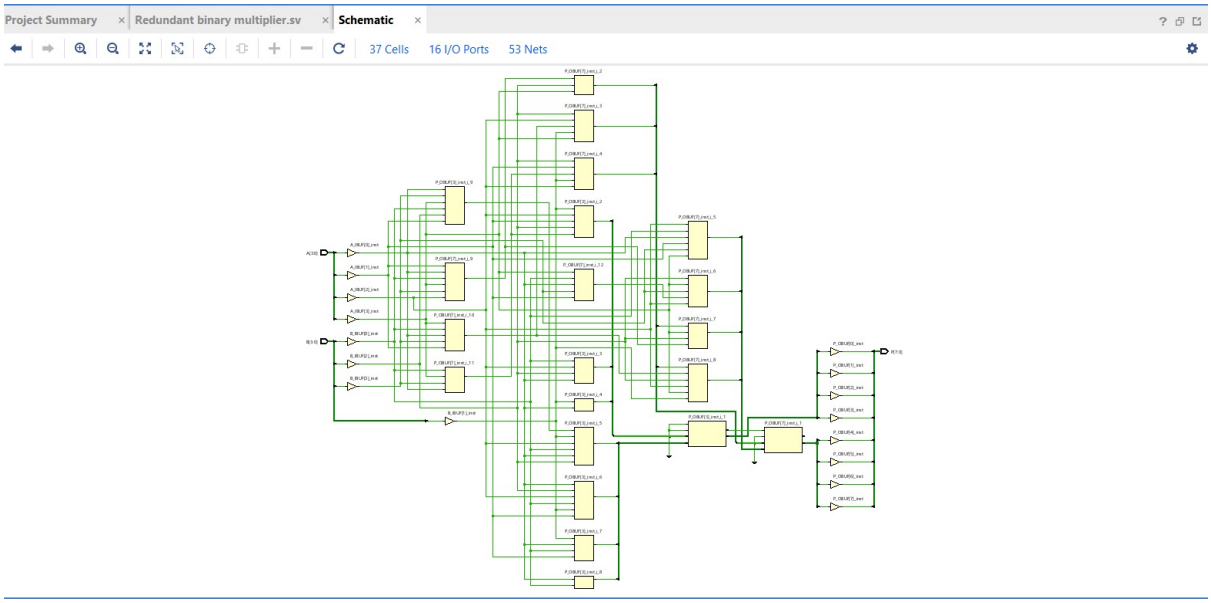

Figure 3: Synthesis Design of Redundant binary multiplier

4 Advantages of Redundant binary multiplier

- **Reduced Carry Propagation:** Minimizes delays during addition, leading to faster multiplication.
- **Increased Speed:** Allows simultaneous addition of partial products, enhancing overall multiplication speed.
- **Efficient Resource Utilization:** Requires fewer logic gates and simpler circuitry compared to traditional multipliers.
- **Scalability:** Easily adaptable for wider bit-widths by adjusting the WIDTH parameter.
- **Flexible Representation:** Multiple representations of the same value optimize arithmetic operations.
- **High Throughput:** Achieves higher throughput in applications with frequent multiplications.
- **Improved Performance in Parallel Processing:** Enhances performance in systems designed for parallel computations.
- **Better Fault Tolerance:** Redundant representations can provide a form of error correction.

5 Disadvantages of Redundant binary multiplier

- **Increased Complexity:**The architecture can be more complex due to the need for carry-save addition and the management of redundant representations, which may complicate design and debugging.
- **Higher Area Requirements:**Although resource utilization can be efficient, the additional circuitry required for handling redundant representations and carry-save addition may lead to a larger chip area compared to simple binary multipliers.

- **Limited Adoption:**The redundancy in binary representation is not commonly used in mainstream digital systems, which can limit interoperability with other components and make integration more challenging.
- **Potential for Redundant Representation Confusion:**The multiple representations of values can lead to confusion in interpreting results, especially if the system is not designed to handle redundant binary formats explicitly.
- **Slower Final Addition:**The final combination of sums and carries can be slower than in traditional binary systems, especially if not implemented with efficient adder circuits.
- **Higher Power Consumption:**The complexity of the architecture may lead to increased power consumption, which can be a concern in low-power applications.
- **Difficulty in Overflow Handling:**Managing overflow in redundant binary can be more complex than in standard binary representation, as multiple representations can obscure the detection of overflow conditions.

6 Applications of Redundant binary multiplier

- **Digital Signal Processing (DSP):** Used for filtering, convolution, and Fast Fourier Transforms (FFT) where fast multiplications are crucial.
- **Graphics Processing:** Employed in GPUs for real-time rendering and complex calculations.
- **High-Performance Computing (HPC):** Utilized in supercomputers for intensive arithmetic operations.
- **Embedded Systems:** Applied in image and audio processing, and control systems requiring efficient arithmetic.
- **Artificial Intelligence and Machine Learning:** Used in hardware accelerators for neural network computations and matrix multiplications.
- **Cryptography:** Enhances performance in cryptographic algorithms requiring fast arithmetic.
- **Data Compression:** Utilized in encoding and decoding processes for efficient compression techniques.
- **Telecommunications:** Applied in modulation and demodulation processes for signal processing.

7 Conclusion

The redundant binary multiplier is a powerful tool in digital arithmetic that leverages redundant binary representation and carry-save addition to achieve efficient and rapid multiplication of binary numbers. Its architecture reduces carry propagation delays, leading to enhanced speed and improved performance in applications that require frequent arithmetic operations, such as digital signal processing, graphics rendering, and high-performance computing.Despite its advantages, such as increased speed and efficient resource utilization, the redundant binary multiplier also presents challenges, including increased complexity and potential difficulties in integration with standard binary systems. However, its unique benefits make it a suitable choice for specialized applications where performance is critical.

8 FAQs

1. What is a redundant binary multiplier?

A redundant binary multiplier is a digital circuit that performs multiplication using redundant binary representation, which allows for multiple representations of the same value. It minimizes carry propagation delays through carry-save addition.

2. How does a redundant binary multiplier differ from a traditional binary multiplier?
Unlike traditional binary multipliers that use a fixed binary representation and suffer from carry propagation delays, redundant binary multipliers utilize redundant representations and carry-save addition, allowing for faster computations.

3. What are the main advantages of using a redundant binary multiplier?
Advantages include reduced carry propagation delays, increased speed, efficient resource utilization, scalability for wider bit-widths, and higher throughput in applications requiring frequent multiplications.

4.What are the main disadvantages of redundant binary multipliers?
Disadvantages include increased complexity, higher area requirements, potential confusion from multiple representations, and challenges in overflow handling and integration with standard binary systems.

5.In what applications are redundant binary multipliers commonly used?
They are commonly used in digital signal processing, graphics processing, high-performance computing, embedded systems, artificial intelligence, cryptography, data compression, and telecommunications.

6.How does carry-save addition work in a redundant binary multiplier?
Carry-save addition allows for the simultaneous addition of multiple partial products without waiting for carries to propagate. It maintains two outputs (sum and carry) at each stage, which are processed in subsequent stages.

7. Can redundant binary multipliers be implemented in hardware?
Yes, redundant binary multipliers can be implemented in hardware using HDLs like Verilog or VHDL, and they can be synthesized for FPGA or ASIC designs.

8. What is the significance of redundant binary representation?
Redundant binary representation allows for multiple ways to express the same value, which can enhance the efficiency of arithmetic operations and reduce the complexity of arithmetic circuits.

9. How does the final addition work in a redundant binary multiplier?
After processing all partial products through carry-save addition, a conventional adder combines the final sums and carries to produce the final output.

10.Is it possible to combine redundant binary multipliers with other arithmetic units?
Yes, redundant binary multipliers can be integrated with other arithmetic units, such as adders and subtractors, in complex digital systems to improve overall computational performance.