**NON-RESTORING DIVISION ALGORITHM**

**1. Introduction Non-Restoring Division:**

is a division algorithm used in digital arithmetic to efficiently compute the quotient and remainder of two binary numbers. Unlike restoring division algorithms, non-restoring division does not always restore the remainder to a non-negative value after each subtraction. This report delves into the details of the non-restoring division algorithm, its implementation, and its applications in digital systems.

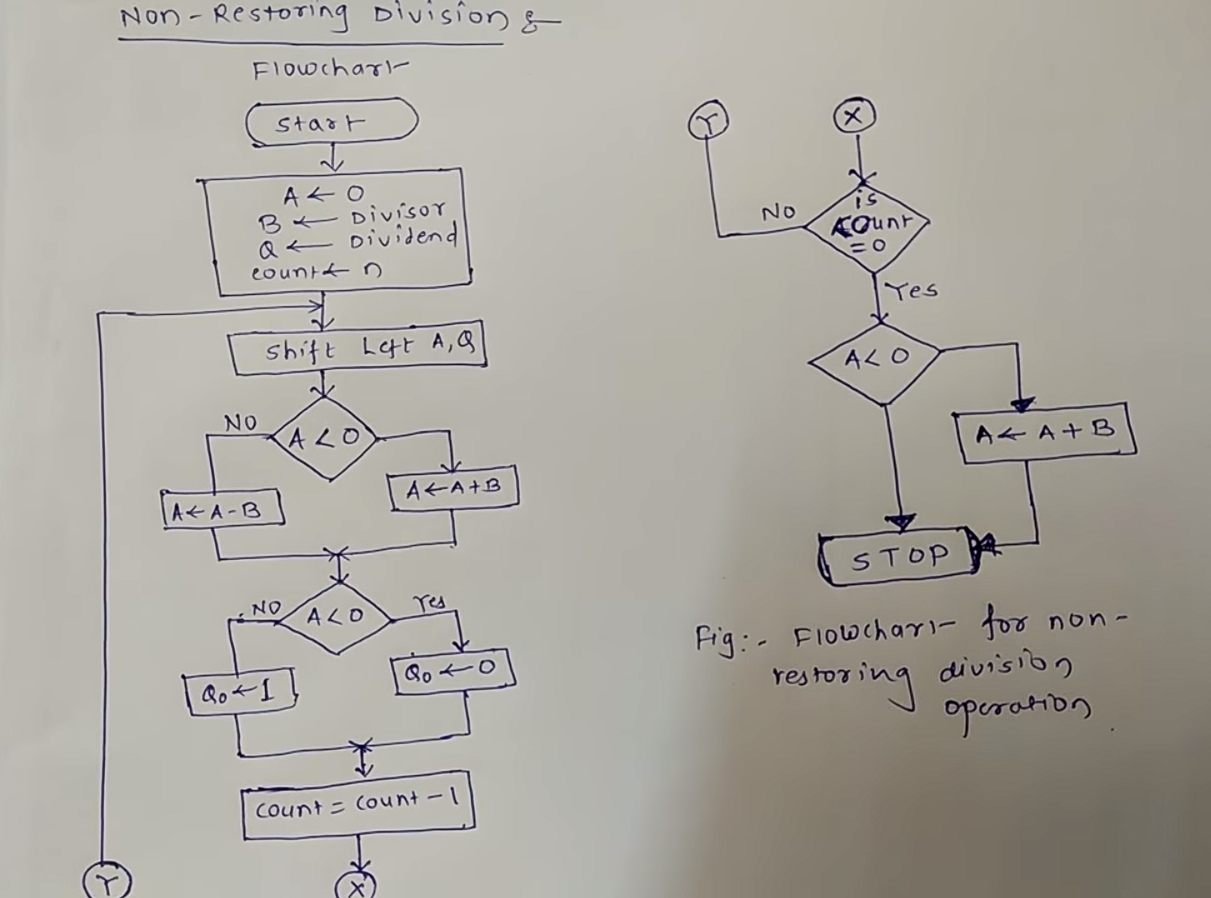
**2. Algorithm Overview:**

Non-Restoring Division operates by iteratively subtracting the divisor from the dividend to obtain the quotient and remainder. The key difference from restoring division is that it does not perform a restoration step when the result of the subtraction is negative. Instead, it adjusts the partial remainder based on the sign of the result.

**3. Implementation Steps**

* Initialization: Load the divisor and dividend into their respective registers.
* Iterative Subtraction:
  + Subtract the divisor from the partial remainder.
  + If the result is negative, add the divisor to the partial remainder.
  + Shift the partial remainder and quotient registers.
* Quotient Bit Adjustment: Set or clear the quotient bit based on the result of the subtraction.
* Repeat: Repeat the iterative subtraction and shifting process until the desired precision or number of iterations is reached.
* Final Adjustment: If needed, adjust the final remainder to ensure it is positive or zero.

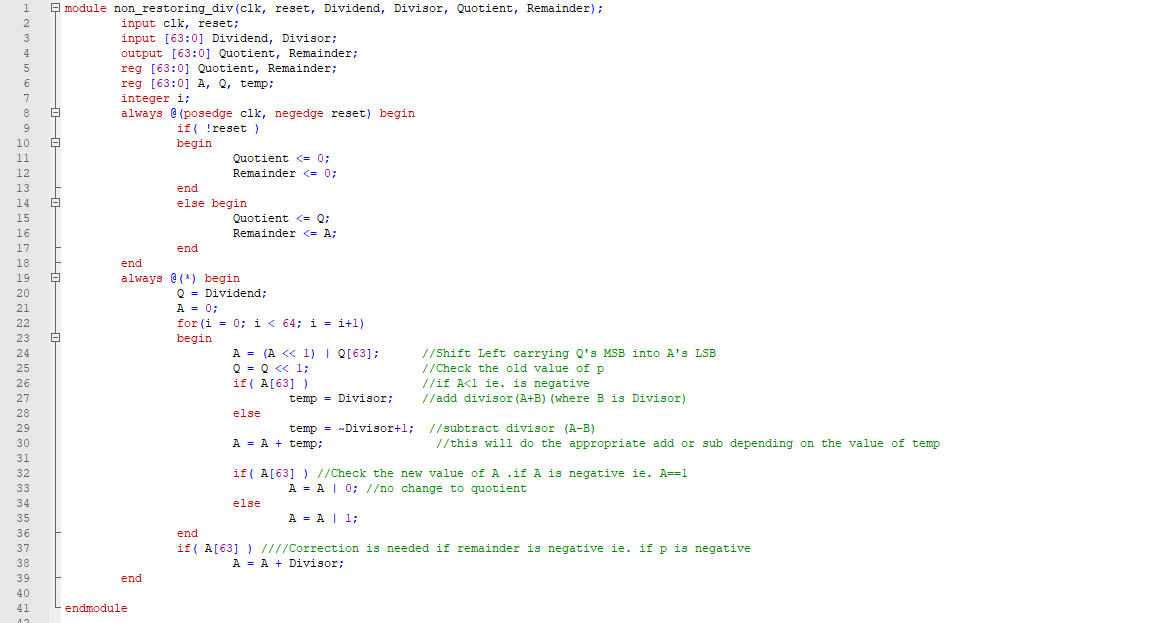
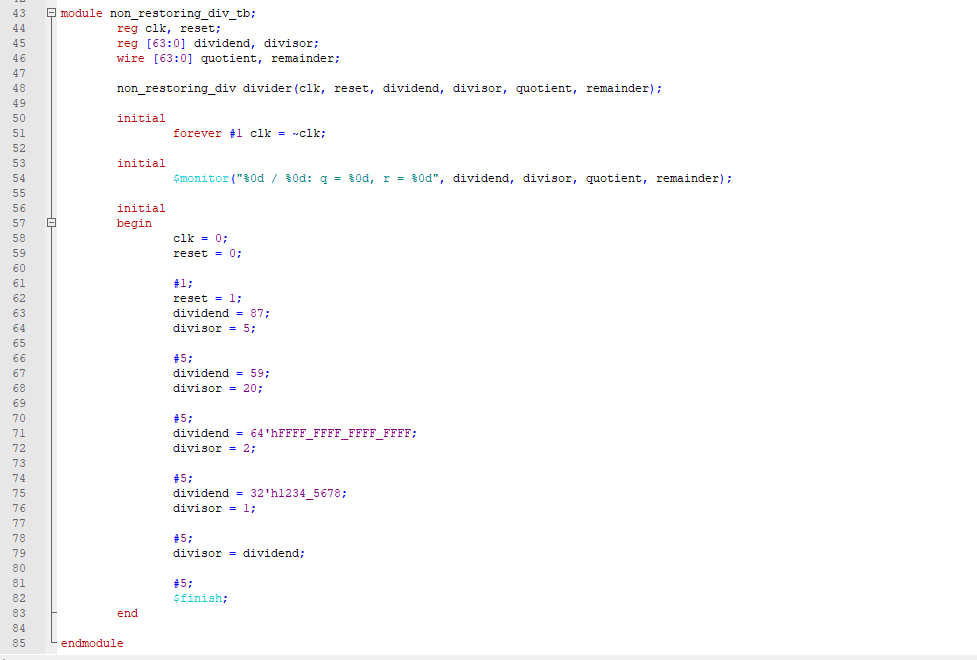
**FLOW CHART:**



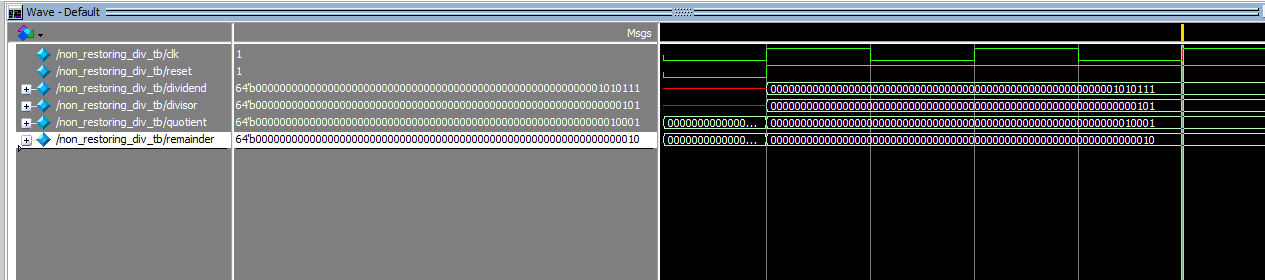
**4. Advantages and Disadvantages**

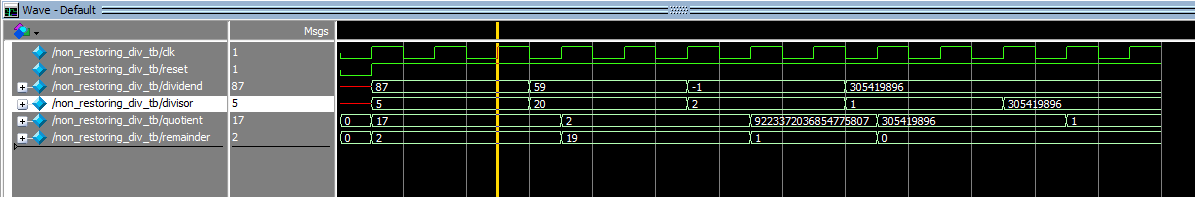
* **Advantages:**
  + Efficient for hardware implementation due to simplified control logic compared to restoring division.
  + Suitable for high-speed arithmetic units in digital systems.
* **Disadvantages:**
  + Requires additional steps to handle negative results, potentially affecting overall performance.
  + May have longer execution times compared to restoring division for certain inputs.

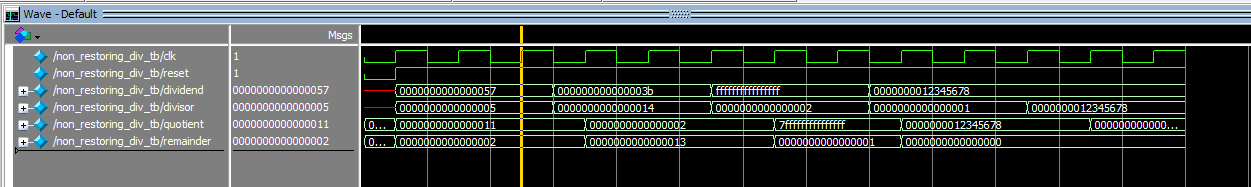
**PROGRAM:**

****

**OUTPUT:**

****

****

****

**5. Applications:**

Non-Restoring Division finds applications in digital signal processing, microprocessor design, and arithmetic units in computer architecture. Its efficiency and suitability for hardware implementation make it a preferred choice in scenarios where high-speed division operations are crucial.

**6. Conclusion:**

In conclusion, the Non-Restoring Division algorithm offers an efficient approach to division in digital systems. While it requires careful handling of negative results, its simplicity and suitability for hardware make it a valuable tool for designing high-performance arithmetic units. Understanding and implementing this algorithm can significantly contribute to the optimization of digital systems' computational capabilities.