**Batch: C1 Roll No.: 16010122245**

**Experiment / assignment / tutorial No.\_\_2\_\_\_\_\_**

|  |
| --- |
| **TITLE:** To study and implement Booth’s Multiplication Algorithm. |

**AIM:** Booth’s Algorithm for Multiplication

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected OUTCOME of Experiment: (Mention CO/CO’s attained here)**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

1. Carl Hamacher, Zvonko Vranesic and Safwat Zaky, “Computer Organization”, Fifth Edition, TataMcGraw-Hill.
2. William Stallings, “Computer Organization and Architecture: Designing for Performance”, Eighth Edition, Pearson.

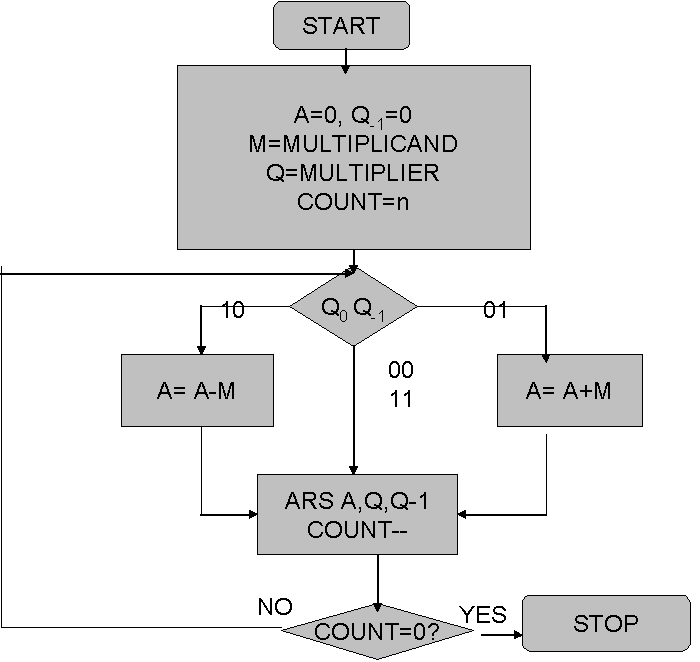
3. Dr. M. Usha, T. S. Srikanth, “Computer System Architecture and Organization”, First Edition, Wiley-India.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Pre Lab/ Prior Concepts:**

It is a powerful algorithm for signed number multiplication which generates a 2n bit product and treats both positive and negative numbers uniformly. Also the efficiency of the algorithm is good due to the fact that, block of 1’s and 0’s are skipped over and subtraction/addition is only done if pair contains 10 or 01

**Flowchart:**

****

**Design Steps**:

1. Start
2. Get the multiplicand (M) and Multiplier (Q) from the user
3. Initialize A= Q-1 =0
4. Convert M and Q into binary
5. Compare Q0 andQ-1 and perform the respective operation.

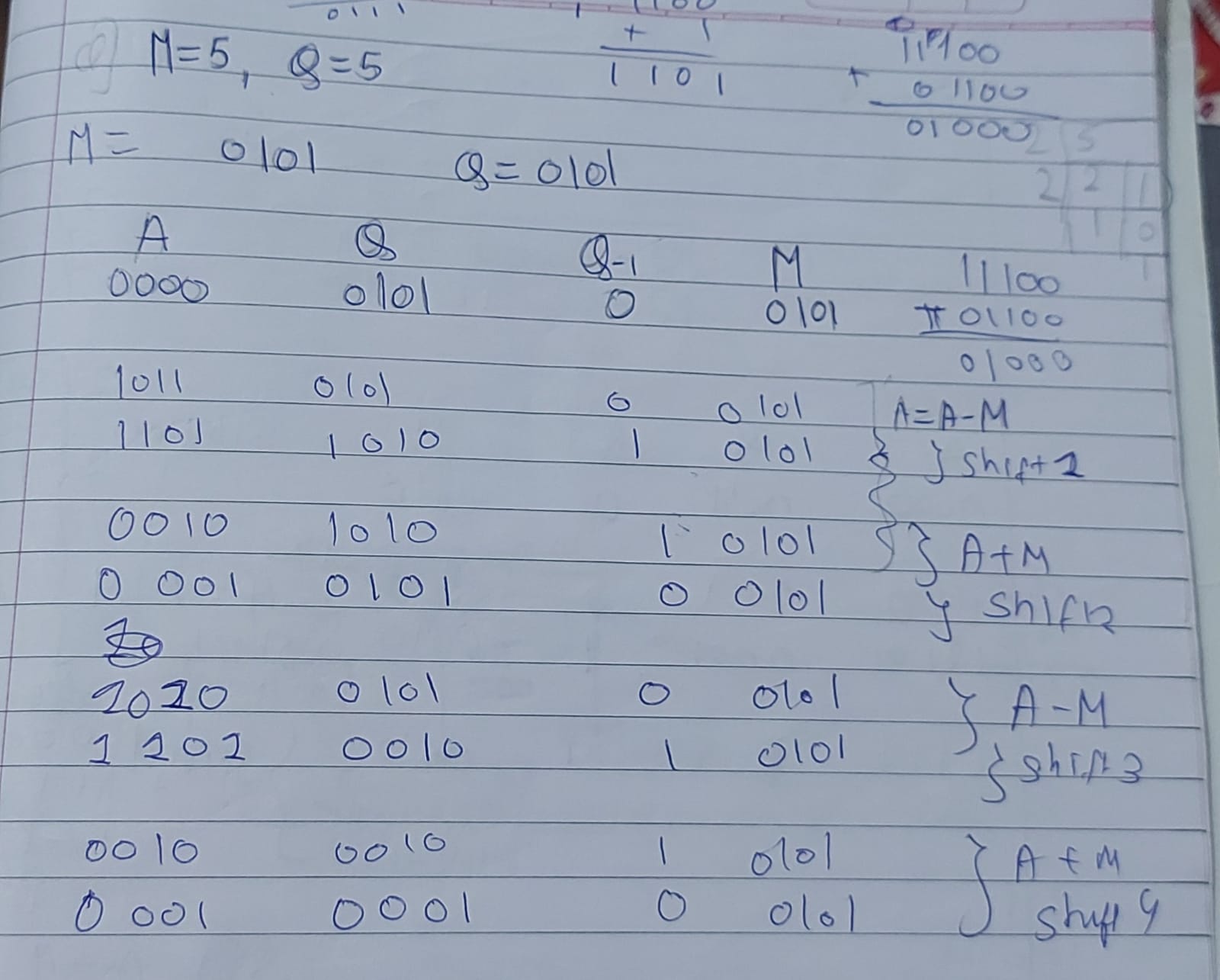
|  |  |
| --- | --- |
| **Q0 Q-1** | **Operation** |
| 00/11 | Arithmetic right shift |
| 01 | A+M and Arithmetic right shift |
| 10 | A-M and Arithmetic right shift |

6. Repeat steps 5 till all bits are compared

7. Convert the result to decimal form and display

8. End

Example: (Handwritten solved problem needs to be uploaded)



**Conclusion:**

**Post Lab Descriptive Questions**

1. **Explain advantages and disadvantages of Booth’s algorithm.**:

Advantages of Booth's Algorithm:

1. \*\*Reduction in Partial Products\*\*: One of the main advantages of Booth's algorithm is its ability to reduce the number of partial products that need to be generated and added during the multiplication process. This reduction leads to fewer arithmetic operations, which can significantly improve the efficiency of multiplication, especially for larger numbers.

2. \*\*Faster Execution\*\*: Due to the reduction in partial products and the resulting fewer arithmetic operations, Booth's algorithm generally performs multiplication faster than conventional methods, such as the straightforward shift-and-add method.

3. \*\*Efficient for Multiplier Implementation\*\*: Booth's algorithm is particularly well-suited for hardware implementation in digital circuits. It can be efficiently designed using logic gates and flip-flops, making it suitable for use in hardware multipliers.

4. \*\*Saves Hardware Resources\*\*: The reduction in the number of partial products not only speeds up the multiplication process but also reduces the amount of hardware resources required for the multiplier implementation.

Disadvantages of Booth's Algorithm:

1. \*\*Complexity\*\*: Booth's algorithm involves a complex sequence of steps, including the generation of recoded values and multiple shifts and adds. This complexity can make the algorithm harder to understand, implement, and debug, compared to simpler multiplication methods.

2. \*\*Limited Applicability\*\*: Booth's algorithm is most beneficial for larger numbers and scenarios where efficient hardware implementation is a priority. For smaller operands, the overhead of the algorithm's complexity might outweigh the benefits gained.

3. \*\*Not Optimized for Modern Processors\*\*: Modern processors use various optimizations and instruction-level parallelism techniques to accelerate arithmetic operations. Booth's algorithm doesn't align well with these optimizations, which can make it less suitable for modern high-performance processors.

4. \*\*Increased Latency\*\*: While Booth's algorithm can lead to fewer overall operations, it may introduce additional latency due to the need for recoding and multiple shift-and-add operations.

5. \*\*Non-Trivial Negative Number Handling\*\*: Handling negative numbers in Booth's algorithm requires extra care and considerations, as the recoding process can be more complex for signed numbers.

In summary, Booth's algorithm offers advantages in terms of reducing partial products and speeding up multiplication for larger numbers in hardware implementations. However, its complexity, limited applicability, and lack of alignment with modern processor optimizations can be considered disadvantages. The choice of whether to use Booth's algorithm depends on the specific requirements and constraints of the application or hardware platform.

1. **Is Booth’s recoding better than Booth’s algorithm? Justify**

Booth's recoding and Booth's algorithm are related concepts used in binary multiplication, but they serve different purposes and are not directly comparable in terms of being "better" or "worse." Let's clarify their roles and differences:

1. \*\*Booth's Algorithm\*\*: Booth's algorithm is a method for performing binary multiplication by reducing the number of partial products and optimizing the process for hardware implementation. It achieves this by using a recoding technique to efficiently add or subtract the multiplicand based on the multiplier's adjacent bits. This results in fewer arithmetic operations compared to traditional methods like shift-and-add.

2. \*\*Booth's Recoding\*\*: Booth's recoding is a specific aspect of Booth's algorithm. It involves converting the multiplier from its original binary representation to a recoded binary representation, where the goal is to replace sequences of repeated bits with shorter representations. This recoding step is integral to Booth's algorithm's efficiency, as it determines when and how much to add or subtract the multiplicand based on the recoded multiplier's bits.

In other words, Booth's recoding is a technique used within Booth's algorithm to streamline the multiplication process. It's not a standalone method that can be directly compared to Booth's algorithm.

**Date: \_\_\_\_\_\_\_\_\_\_\_\_\_**

\