Multiplication

Contents

Sequential Multiplication Method (for unsigned numbers)

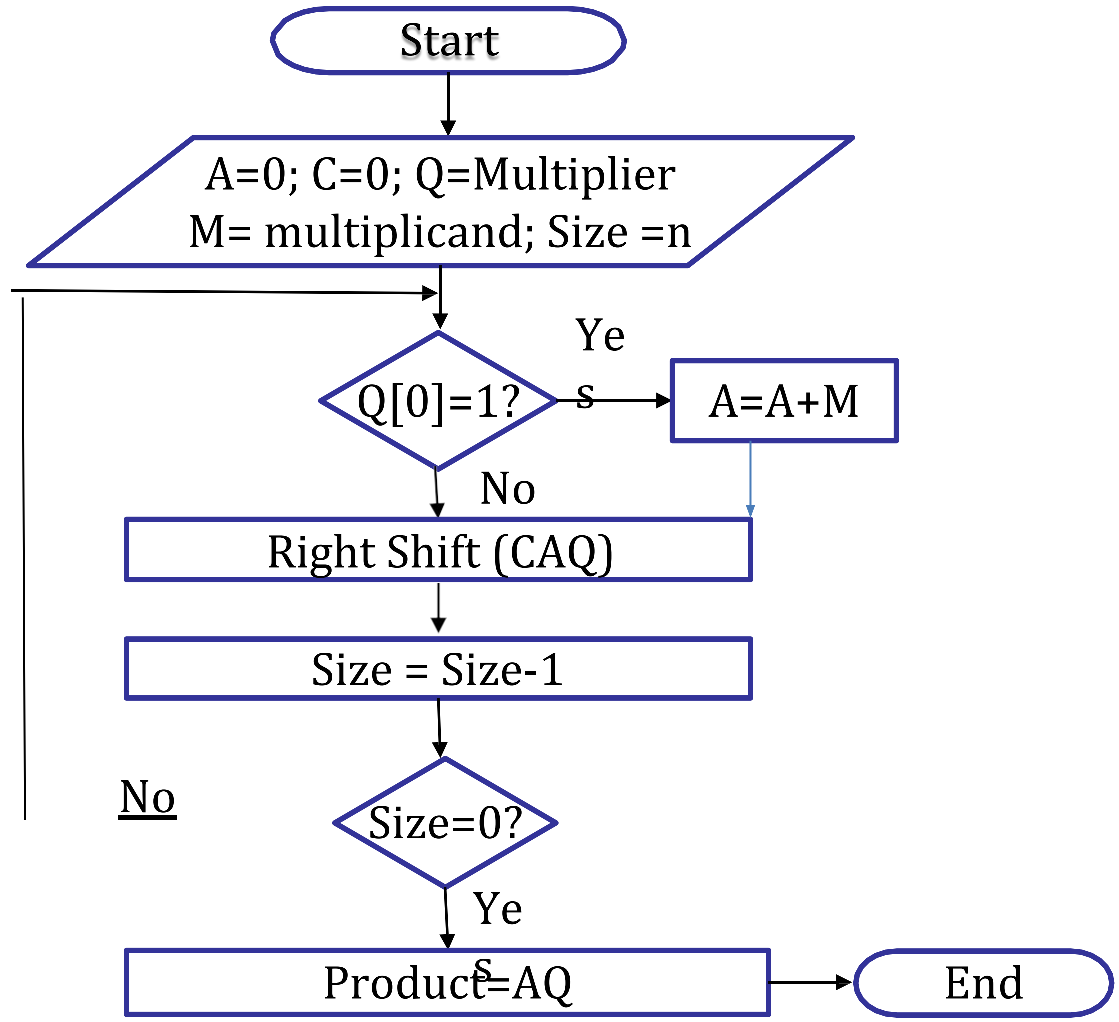
Required registers:

1) Register **A** = Accumulator and initialized to 0
2) Register **Q** = Multiplier value
3) Register **M**= Multiplicand value

The flip-flop **C** holds the end carry generated in the addition

This flip-flop is used as the serial input, when the register pair AQ is shifted right one position.

Flowchart for Sequential Multiplication Method



Booth's Multiplication Method
(for signed numbers)

The multiplication of signed numbers in sequential multiplication method requires extra processing steps besides the main multiplication for the magnitude. This is an overhead when operands are denoted in signed 2's complement form.

The overhead can be eliminated by a specific mapping rule called the **recoded multiplication technique**; in which the multiplier is mapped in accordance with the recoding technique.

Booth's Multiplication Method

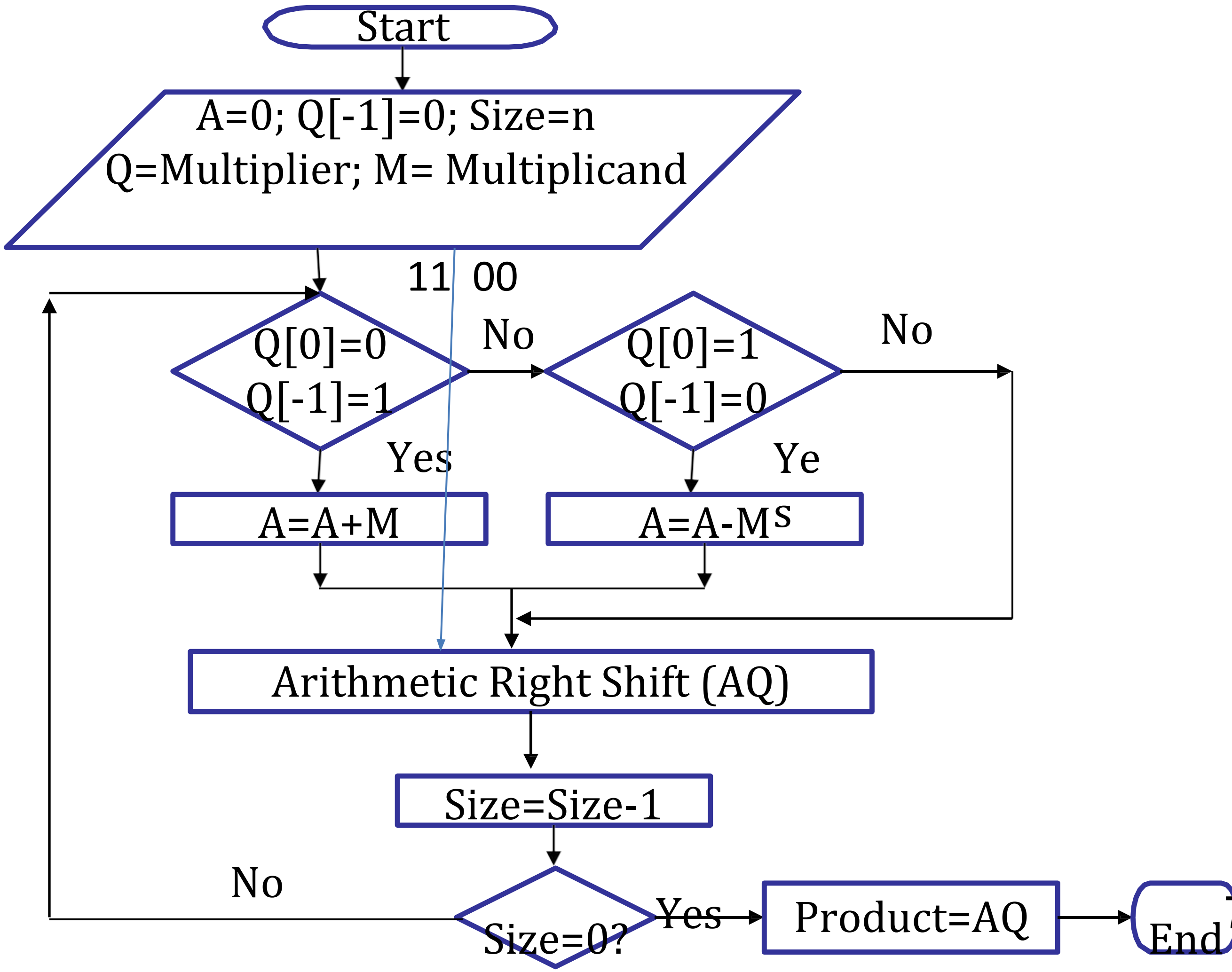
The basis for recoding property is known as **String property.**

" A block of consecutive k 1s in a binary sequence of multiplier may be replaced with a block of (k-1) consecutive 0s surrounded by the digits 1 & $\overline{1}$.

Advantages of Booth's Multiplication:

1. Pre processing steps are unnecessary. So, the Booth's algorithm treats signed numbers in a uniform way with unsigned numbers.
2. Less numbers of addition & subtraction are needed compared to the sequential multiplication method.

Booth's Multiplication Method



Example

Question: M= - 6 & Q= 7; Find out M × Q.

Given, M= - 6; We know, 6 = 110 & 7= 0111

Therefore, 1's complement of 6(110) = 001

& 2's complement of 6(110) = 010

So, - 6= **1**010 (For –ve 6, we have put MSB as 1)

Initial Configuration:	**M**	**A**	**Q**	**F**	**Size**
	1010	0000	0111	0	4
Step 1:					
Q[0]=1					
Q[-1]=0					
So, A=A-M	1010	0110	0111	0	-
&	1010	0011	0011	1	3
ARS(AQ)					

Example contd.

Initial Configuration:	**M**	**A**	**Q**	**F**	**Size**
	1010	0011	0011	1	3
Step 2:					
Q[0]=1					
Q[-1]=1					
So, ARS(AQ)	1010	0001	1001	1	2
Step 3:					
Q[0]=1					
Q[-1]=1					
So, ARS(AQ)	1010	0000	1100	1	1
Step 4:					
Q[0]=0					
Q[-1]=1					
So, A=A=M	1010	1010	1100	1	-
&, ARS(AQ)	1010	1101	0110	0	0

Since, Size Register becomes 0, the algorithm is terminated and the product is AQ= 1101 0110, which shows a –ve number. To get the number in familiar form, 2's complement of the magnitude is taken; hence the result is -42.