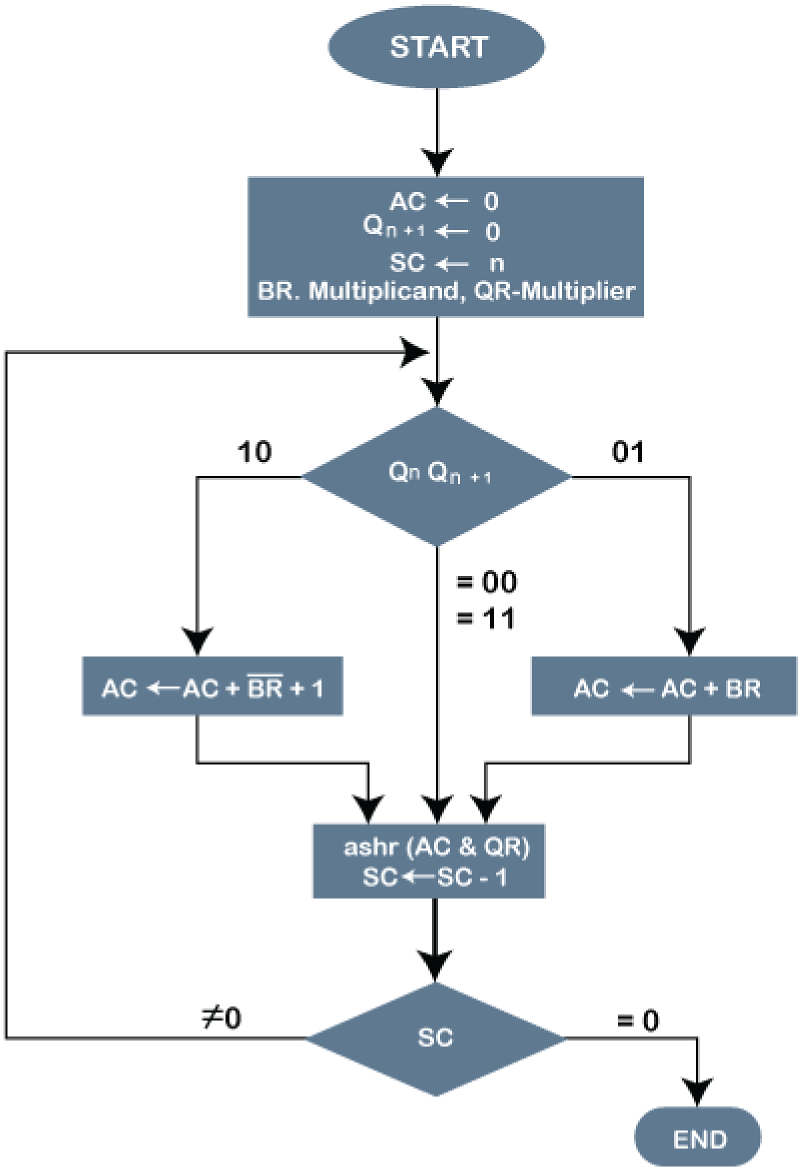
Booth's Multiplication Algorithm

The booth algorithm is a multiplication algorithm that allows us to multiply the two signed binary integers in 2's complement, respectively. It is also used to speed up the performance of the multiplication process. It is very efficient too. It works on the string bits 0's in the multiplier that requires no additional bit only shift the right-most string bits and a string of 1's in a multiplier bit weight 2^k to weight 2^m that can be considered as $\mathbf{2^{k+1} - 2^m}$.

Following is the pictorial representation of the Booth's Algorithm:



In the above flowchart, initially, **AC** and $\mathbf{Q_{n+1}}$ bits are set to 0, and the **SC** is a sequence counter that represents the total bits set **n,** which is equal to the number of bits in the multiplier. There are **BR** that represent the **multiplicand bits,** and QR represents the **multiplier bits**. After that, we encountered two bits of the multiplier as Q_n and Q_{n+1}, where Qn represents the last bit of QR, and Q_{n+1} represents the incremented bit of Qn by 1. Suppose two bits of the multiplier is equal to 10; it means that we have to subtract the multiplier from the partial product in the accumulator AC and then perform the arithmetic shift operation (ashr). If the two of the multipliers equal to 01, it means we need to perform the addition of the multiplicand to the partial product in accumulator AC and then perform the arithmetic shift operation (**ashr**), including $\mathbf{Q_{n+1}}$. The arithmetic shift operation is used in Booth's algorithm to shift AC and QR bits to the right by one and remains the sign bit in AC unchanged. And the sequence counter is continuously decremented till the computational loop is repeated, equal to the number of bits (n).

Booth's algorithm is a multiplication algorithm that multiplies two signed binary numbers in 2's compliment notation.

PROCEDURE:

1. Let M is the multiplicand.
2. Let Q is the multiplier.
3. Consider a 1-bit register Q_{-1} and initialize it to 0.
4. Consider a register A and initialize it to 0.

CONDITIONS:

1. If $Q_0\,Q_{-1}$ are same i.e. 00 or 11 then, perform arithmetic right shift by 1 bit.
2. If $Q_0\,Q_{-1}$ = 10 then perform

   A←A-M

   And then perform arithmetic right shift.
3. If $Q_0\,Q_{-1}$ = 01 then perform

   A←A+M

   And then perform arithmetic right shift.

For example:

Consider two numbers 6 and 2 and we have to perform their multiplication by using Booth's algorithm.

Here 6 is multiplicand (M) and 2 is multiplier (Q).

Now write 6 and 2 in binary form.

M = 6 = 0 1 1 0

Q = 2 = 0 0 1 0 (Q_3, Q_2, Q_1, Q_0)

Booth's algorithm calculates the product in n steps where n is the number of bits used to represent the numbers.

INITIALISE	A	B	Q_{-1}	OPERATIONS
	0 0 0 0 ↓↘↘↘↘	0 0 1 0 ↘↘↘↘	0	
Step 1.	0 0 0 0	0 0 0 1 ↓	0 ↓	Arithmetic right shift
Step 2.	1 0 1 0 ↓↘↘↘↘ 1 1 0 1	0 0 0 1 ↘↘↘↘ 0 0 0 0	0 1	A←A-M Then shift
Step 3.	0 0 1 1 ↓↘↘↘↘ 0 0 0 1 ↓↘↘↘↘	0 0 0 0 ↘↘↘↘ 1 0 0 0 ↘↘↘↘	1 0	A←A+M Then shift
Step 4.	0 0 0 0	1 1 0 0 In binary, 12 = 1100 Hence 3*2 = 12	0	Arithmetic right shift