32 bit Booth Multiplier

K Samhitha[1] G Anusha[2]

[1]EE16BTECH11019

[2]EE16BTECH11011

March 8, 2019

Introduction

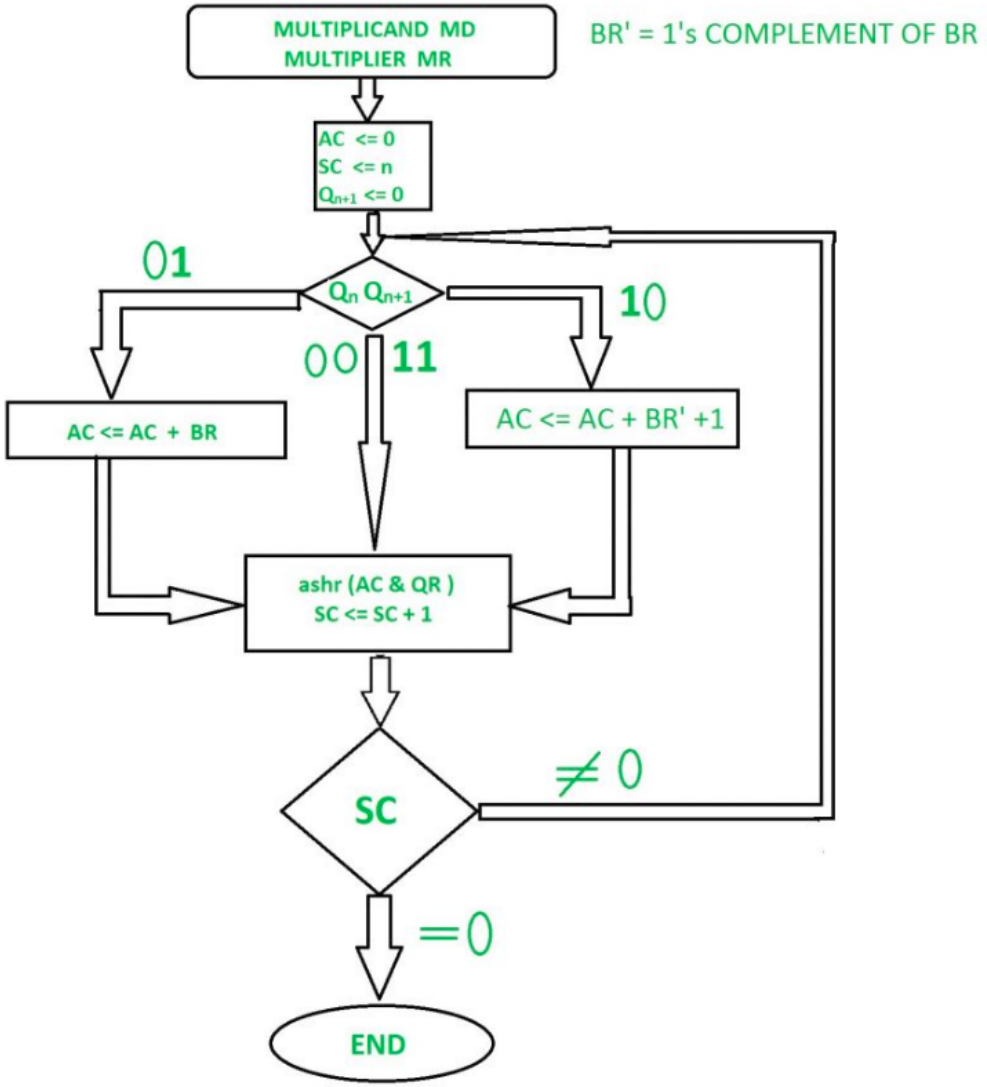
- Booth algorithm gives a procedure for multiplying binary integers in signed 2s complement representation in efficient way, i.e., less number of additions/subtractions required.
- It operates on the fact that strings of 0s in the multiplier require no addition but just shifting and a string of 1s in the multiplier from bit weight 2^k to weight 2^m can be treated as $2^{(k+1)}$ to 2^m.

Hardware Implementation

1. We name the register as A, B and Q, AC, BR and QR respectively.
2. Q_n *designates the least significant bit of multiplier in the register QR.*
3. An extra flip-flop Q_{n+1} *is appended to QR*

to facilitate a double inspection of the multiplier.
The flowchart for booth algorithm is shown below.

Flowchart



from 2019-03-08 16-42-22.png

Implementation

1. AC and the appended bit Q_{n+1} *are initially cleared to* 0
and the sequence SC is set to a number n equal to the number of bits in the multiplier.
2. The two bits of the multiplier in Q_n *and* Q_{n+1} *are inspected*.
3. If the two bits are equal to 10, it means that the first 1 in a string has been encountered.
4. This requires a subtraction of the multiplicand from the partial product in AC.
5. If the 2 bits are equal to 01, it means that the first 0 in a string of 0s has n = been encountered.
This requires the addition of the multiplicand to the partial product in AC.

Implementation

The next step is to shift right the partial product and the multiplier (including Q_{n+1}).
This is an arithmetic shift right (ashr) operation which AC and QR to the right and leaves the sign bit in AC unchanged.
The sequence counter is decremented and the computational loop is repeated n times.

And last

Thank You