Booth Multiplication Algorithm

Booth Algorithm

•An efficient way to multiply two signed binary numbers expressed in 2's complement notation :

•Reduces the number of operations by relying on blocks of consecutive 1's

•Example:

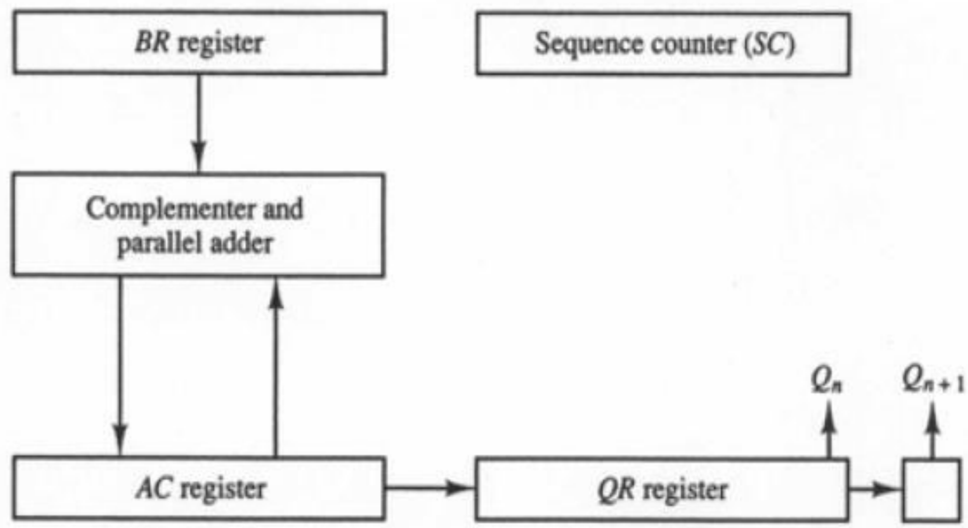
•$Y \times 00111110 = Y \times (2^5+2^4+2^3+2^2+2^1)$.

•$Y \times 00111110 = Y \times (01000000-00000010) = Y \times (2^6-2^1)$.

One addition and one subtraction

Description and Hardware for Booth Multiplication

- QR multiplier
- Q_n least significant bit of QR
- Q_{n+1} previous least significant bit of QR
- BR multiplicand
- $AC = 0$
- SC number of bits in multiplier



Algorithm

Do SC times

$Q_n Q_{n+1} = 10$

$$AC \leftarrow AC + \overline{BR} + 1$$

$Q_n Q_{n+1} = 01$

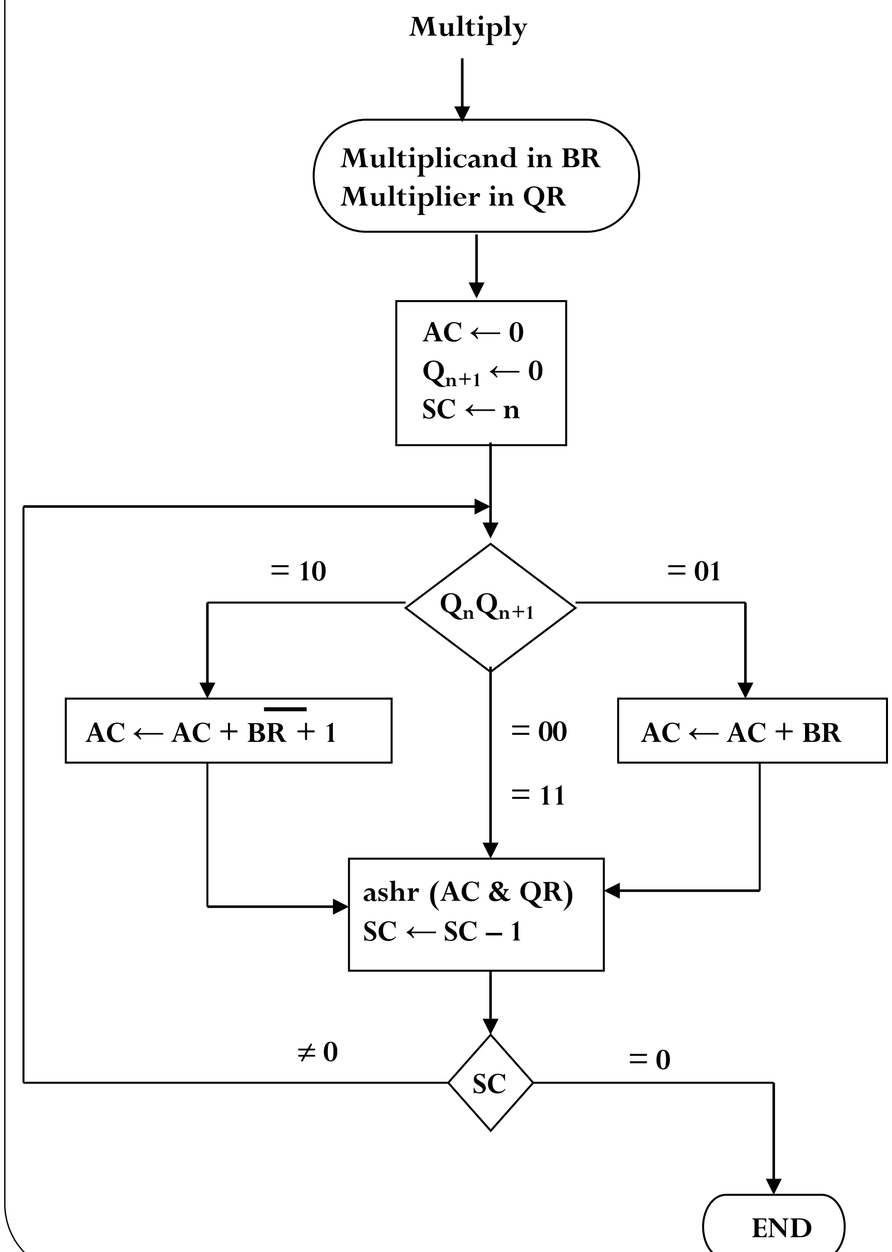
$$AC \leftarrow AC + BR$$

Arithmetic shift right AC & QR

$SC \leftarrow SC - 1$

- For our example, and multiply **(-9) x (-13)**
  - The numerically larger **operand** (**13) would require 4 bits** to represent in binary (1101). So we must use **AT LEAST 5 bits** to represent the operands, to allow for the sign bit.

Flowchart for Booth Multiplication



Example: -9 × -13 = 117

BR = 10111, $\overline{BR}$ + 1 = 01001

Comment	AC	QR	Q_{n+1}	SC
	00000	10011	0	5
Subtract BR	01001			
	01001			
Ashr	00100	11001	1	4
Ashr	00010	01100	1	3
Add BR	10111			
	11001			
Ashr	11100	10110	0	2
Ashr	11110	01011	0	1
Subtract BR	01001			
	00111			
Ashr	00011	10101	1	0

Multiply 7 x 3 using above signed 2's compliment binary multiplication.

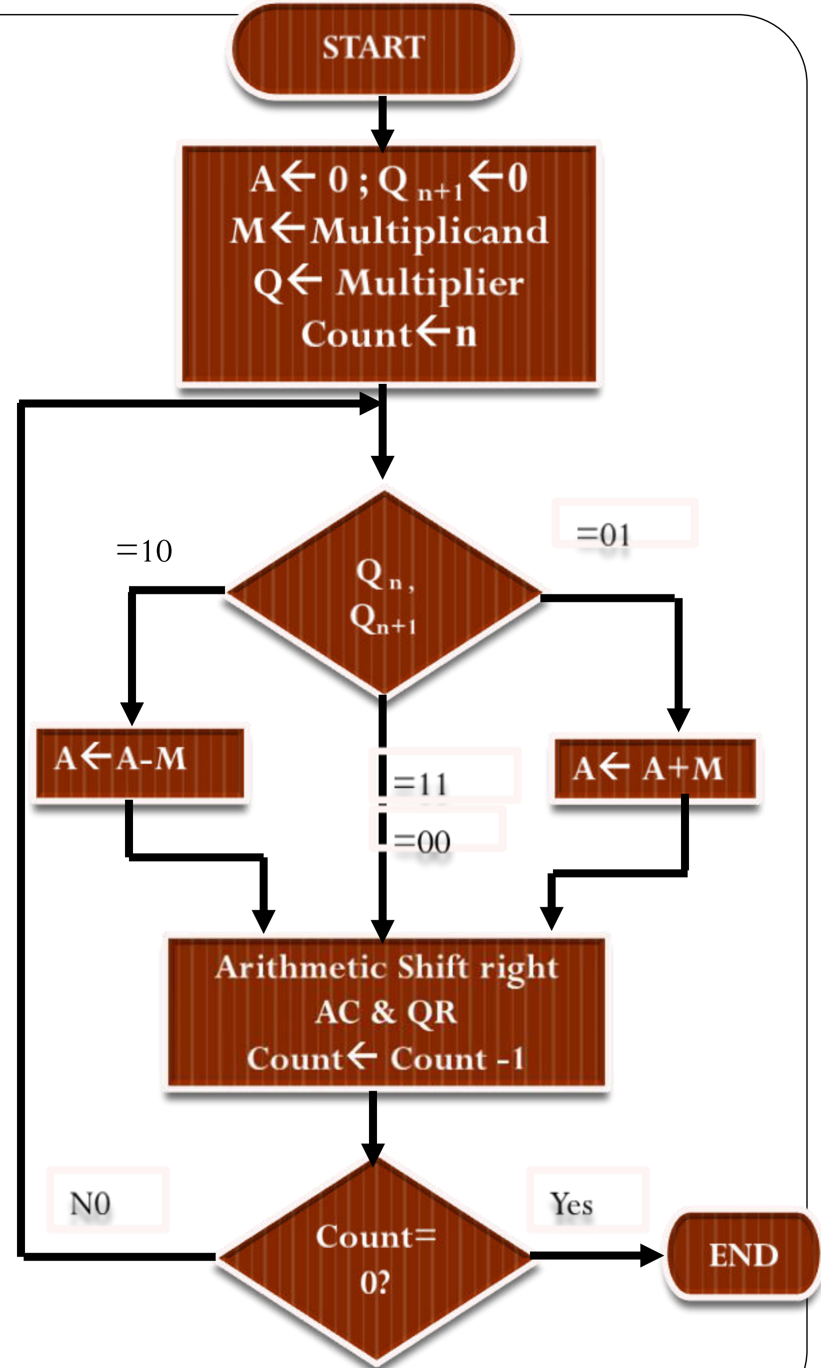
Multiplicand =7 → Binary equivalent is 0111→M

Multiplier = 3 → Binary equivalent is 0011→Q

-7 → Binary equivalent is 1001→ -M

A 0 0 0 0						**A** 1 1 1 0			
+ **-M** 1 0 0 1						+ **M** 0 1 1 1			
A 1 0 0 1						**A** 0 1 0 1			

Step	A	Q	Q_{n+1}	Action	Count
1	0 0 0 0	0 0 1 1	0	Initial	4
2	1 0 0 1	0 0 1 1	0	A←A-M	
2	1 1 0 0	1 0 0 1	1	Shift	3
3	1 1 1 0	0 1 0 0	1	Shift	2
4	0 1 0 1	0 1 0 0	1	A←A+M	
4	0 0 1 0	1 0 1 0	0	Shift	1
5	0 0 0 1	0 1 0 1	0	Shift	0



Examples

- Multiply the following using Booth's algorithm

  7 x -3

  -7 x 3

  -7 x -3

  11 x 13

  -11 x 13

  11 x -13

  -11 x -13

Reference

- Morris Mano, "Computer System Architecture", Pearson Education, 3rd edition (Chapter 10)