KIET Group of Institutions
Department of Information Technology
COURSE B.Tech., 3rd SEM,
Computer Organization and Architecture (KCS-302)
Session 2020-21

Booths Multiplication Algorithm

Multiplication Algorithms: —

Multiplication of two fixed point binary number in signed magnitude representation is done as follows

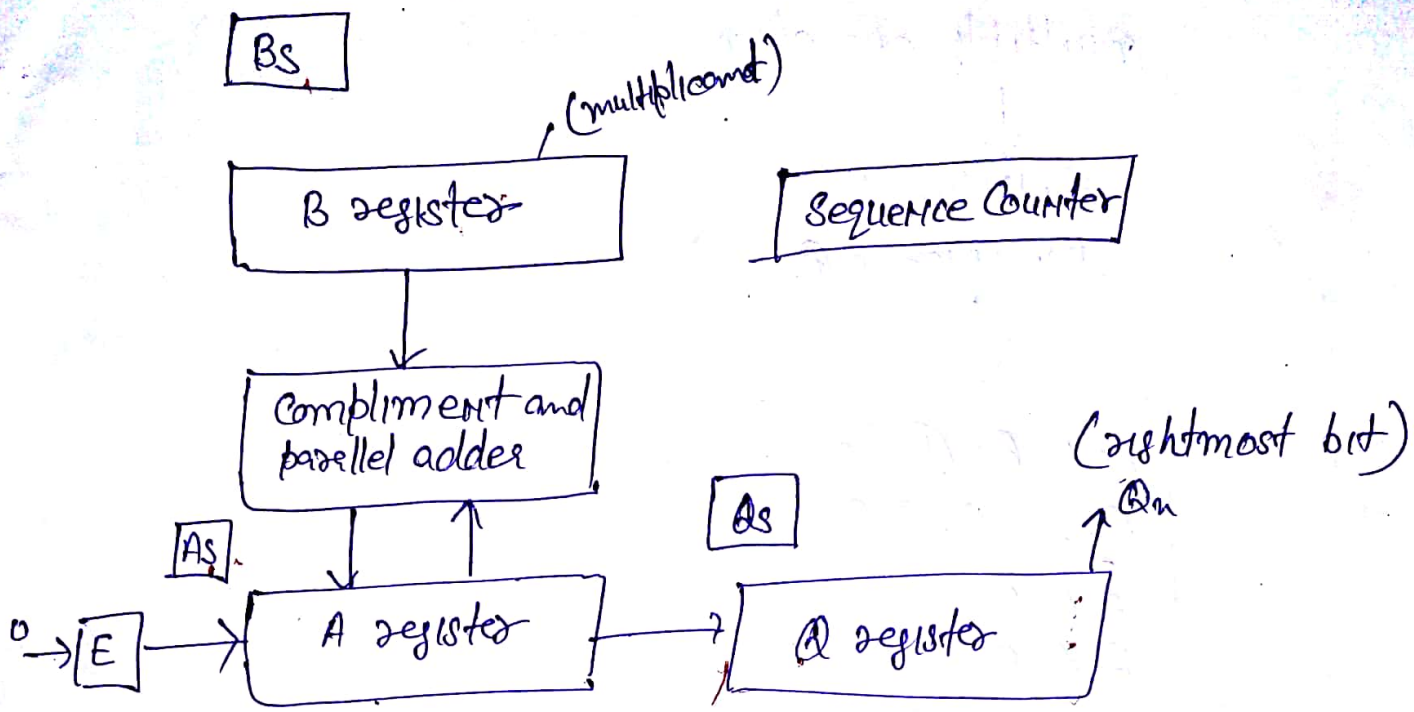
```
  23              10111   multiplicand
 ×19              10011   multiplier
                 ------
                  10111
                 10111
                00000
               00000
              10111
             ----------
 437          110110101   Product
```

The process consist of looking at successive bits of the multiplier, LSB first. If multiplier bit is 1, multiplicand is copied down otherwise 0's are copied down. The numbers copied down in successive lines are shifted one position left.

Hardware Implimentation :

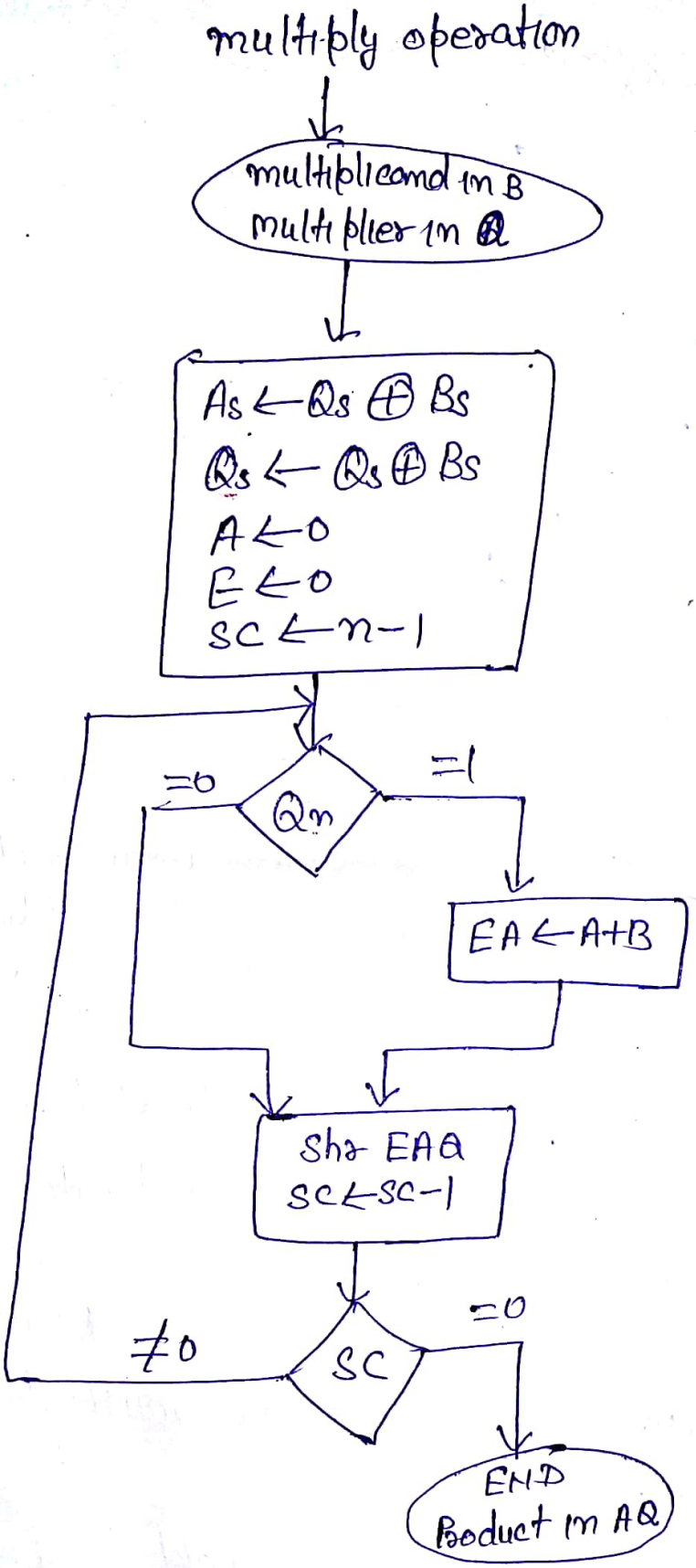
When multiplication is done in a digital computer certain changes are made to the process.

1. Instead of providing registers to store and add simultaneously as many binary numbers as there are bits in multiplier, it is convenient to provide an adder to add two no's and successively accumulate the partial product in a register.
2. Instead of shift multiplicand to the left, partial product is shifted right
3. If corresponding bit of multiplier is 0, no need to copy and add 0's to partial product.



Hardware for multiply operation

→ Sequencer is mutally set to number of bits in multiplier
→ Initally multiplicand is in Register B and multiplier is in Q. The sum of A and B forms a partial product which is transfered to EA. Both partial product and multiplier are shifted right. This shift is denoted by the statement shr EAQ. The LSB of A is shifted into msb of Q, the bit from E is shifted into A and 0 is shifted into E. After the shift, one bit of partial product is shifted INTO Q, pushing the multiplier bit one position to right. In this manner the rightmost flip flop in register Q will hold the bit of multiplier, which must be inspected next.



Flow chart for multiply operation

10111 multiplicand
X 10011 multiplier

-9
3
-9

multiplicand B = 10111

	E	A	Q	SC
multiplier in Q				
	0	00000	10011	101
		10111		
Qn=1, add B				
① First partial product	0	10111		
shift right EAQ	0	01011	11001	100
		10111		
Qn=1, add B				
Second partial product	1	00010		
shift right EAQ	0	10001	01100	011
Qn=0 shr EAQ	0	01000	10110	010
Qn=0 shr EAQ	0	00100	01011	001
Qn=1 add B		10111		
	0	11011		
5th partial product, shr EAQ	0	01101	10101	000

Final Product in A Q

Booth Multiplication Algorithm →

Booth gave a procedure for multiplying binary integers in signed 2's compliment representation. It operates on the fact that string of 0's in the multiplier require no addition but just shifting, and a string of 1's in the multiplier from bit weight 2^k to 2^m can be treated as $2^{k+1} - 2^m$.

e.g binary number 001110 (+14) has string of 1's from 2^3 to 2^1 i.e $K=3, m=1$

So $2^{k+1} - 2^m$

$= 2^4 - 2^1 = 16 - 2 = 14.$

Therefore the multiplication $M \times 14$ (M is multiplicand and 14 is multiplier) can be done as $M \times 2^4 - M \times 2^1$. Thus the product can be obtained by shifting the binary multiplicand 4 times left and subtracting M shifted once.

Booth algo requires examination of multiplier bits and shifting of partial product. Prior to the shifting, the multiplicand may be added to, the partial product, subtracted from the partial product or left unchanged as per following rules.

① The multiplicand is subtracted from partial product upon encountering the first least significant 1 in a string of 1's in the multiplier

② The multiplicand is added to partial product upon encountering the first 0 (provided that there was a previous 1) in a string of 0's in the multiplier

③ The partial product does not change when the multiplier bit is identical to the previous multiplier bit.

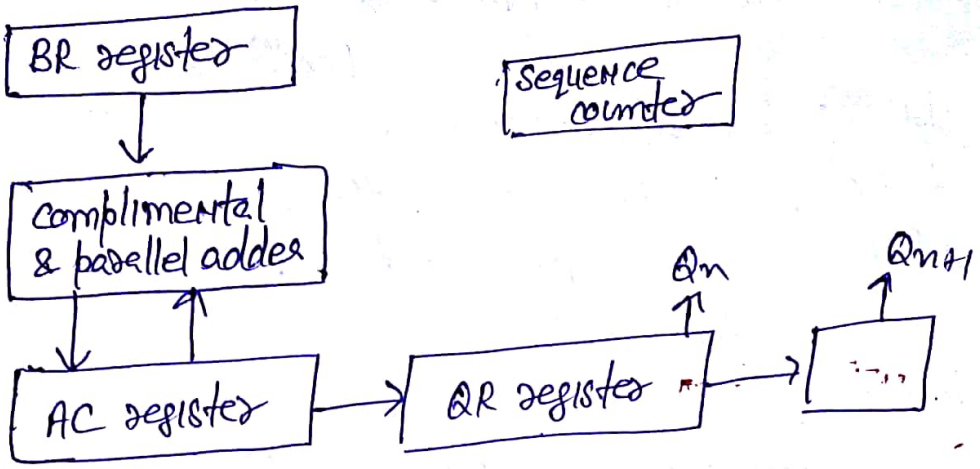


Fig Hardware for booth Algo.

→ Qn designates the LSB of multiplier in register QR
→ An extra flip flop Qn+1 is to facilitate a double bit inspection.

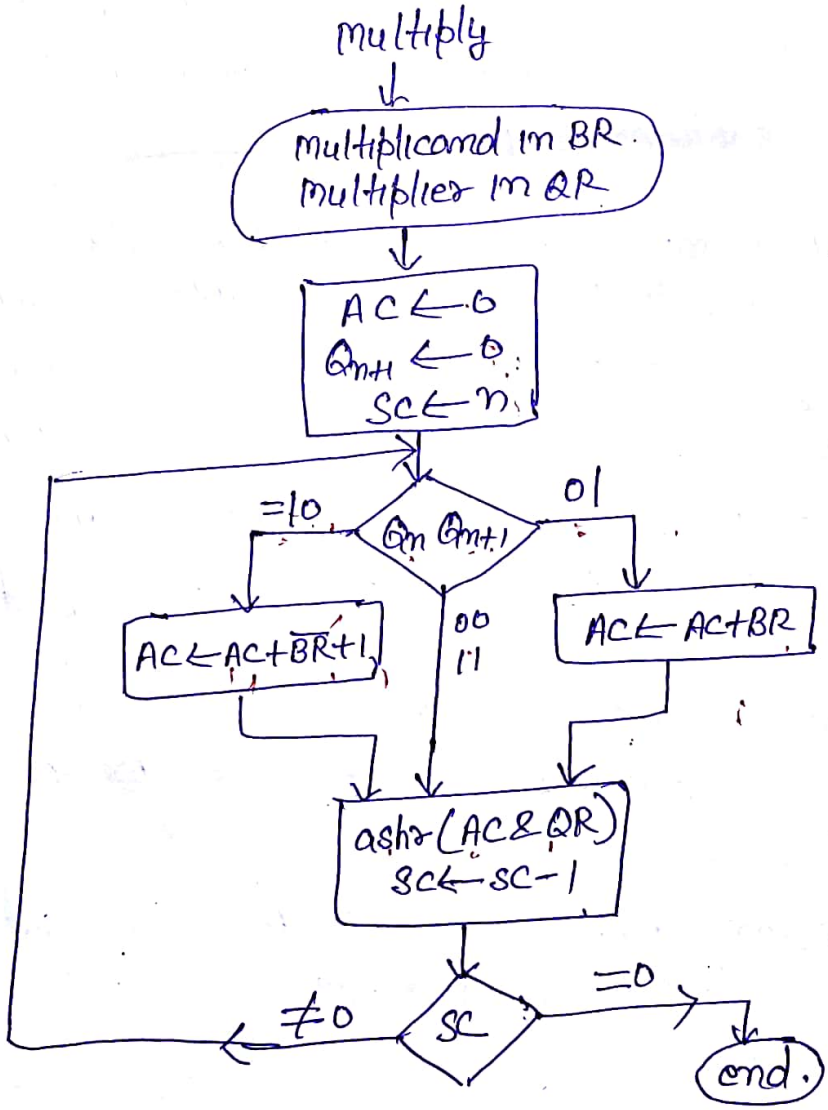


Fig Booth Multiplication of 2's signed 2's compliment numbers

eg Booth multiplication. $(-9) \times (-13)$

Q_n Q_{n+1}	$BR = 10111$ $\overline{BR}+1 = 01001$	AC	QR	Q_{n+1}	SS
	mutial	00000	10011	0	101
1 0	subtract BR	01001			
		01001			
	ashr(AC QR Q_{n+1})	00100	11001	1	100
1 1	ashr(AC, QR, Q_{n+1})	00010	01100	1	011
0 1	Add BR	10111			
		11001			
	ashr(AC, QR, Q_{n+1})	11100	10110	0	010
0 0	ashr(AC, QR, Q_{n+1})	11110	01011	0	001
1 0	subtract BR	01001			
		00111			
	ashr(AC, QR, Q_{n+1})	00011	10101	1	000

(000111010) Ans

2's compliment of $-9 = 10111$

2's compliment of $-13 = {}^{x}10011$

$-9 \times -13 = +117 = 1110101$ Ans

Q 1 Solve Using Booth Algo

$(+15) \times (+13) = (+195) = (0011000011)_2$

BR = 01111 $\quad \overline{BR}+1 = 10001 \quad QR = 01101$

	Q_n Q_{n+1}	AC	QR	Q_{n+1}	Sc
Initial		00000	01101	0	101
10 subtract BR		10001			
		10001			
ashr		11000	10110	1	100
01 Add BR		01111			
		00111			
ashr		00011	11011	0	011
10 subtract BR		10001			
		10100			
ashr		11010	01101	1	010
11 ashr		11101	00110	1	001
01 add BR		01111			
		01100			
ashr		00110	00011	0	000

$(195)_{10}$

Solve Using Booth $(+15) \times (-13) = (-195) = (11001111101)_{2's\ comp\ of\ -195}$

$BR = 01111$ $\overline{BR}+1 = 10001$ $QR = 10011$

Q_n Q_{n+1}		AC	QR	Q_{n+1}	SC
	Initial	00000	10011	0	101
10	subtract BR	10001			
		10001			
	ashr	11000	11001	1	100
11	ashr	11100	01100	1	011
01	add BR	01111			
		01011			
	ashr	00101	10110	0	010
00	ashr	00010	11011	0	001
10	subtract BR	10001			
		10011			
	ashr	11001	11101	1	000

-195