Booth's Algorithm

Wednesday, November 17, 2021 3:48 PM

Karatsuba → Divide & Conquer

↳ valid for large numbers

A × $\boxed{B}$

↑ Multiplicant ↑ Multiplier.

Booth's

−ve :- Stored as 2's compliment

M = 7 (Multiplicand) A = 0 (Accumulator) Q_{-1} = Previous LSB

Q = 3 (Multiplier) n = 4 (no. of Binary)

= 21

M = 7 = 0111

Q = 3 = 0011

= 10101

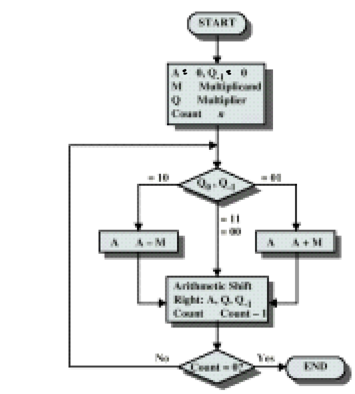

Figure 8.12 Booth's Algorithm for Twos Complement Multiplication

LSB(Q)

	A	Q	Q_{-1}	Operation
#1	0000	0011	0	A = A-M
	1001	0011	0	Right shift
	1100	1001	1	
#2	1100	1001	1	Right Shift
	1110	0100	1	
#3	1110	0100	1	A = A + M
	0101	0100	1	Right shift
	0010	1010	0	
#4	0010	1010	0	Right shift
	0001	0101	0	

0000
1001
———
1001

$\overline{M}$ = 0111 → 1000 → 1001

1 1
1110 → A
0111 → M
———
10101

0001 0101

16 + 4 + 1 = 21

$Q \rightarrow$ $Q_3 \, Q_1 \, Q_1 \, Q_0$ Q_{-1} operation

Q_0	Q_{-1}	operation
0	0	$\rightarrow$ Right shift $A\,Q\,Q_{-1}$
1	1	$\rightarrow$ Right $A\,Q\,Q_{-1}$
1	0	$\rightarrow A - M \rightarrow$ Right shift $A\,Q\,Q_{-1}$
0	1	$\rightarrow A + M \rightarrow$ Right $A\,Q\,Q_{-1}$

$7 - 3 = 7 + (-3)$

0011
$\downarrow$ 1's Comp
1100
$\downarrow +1$
1101

$M = 8$ $Q = 6$

$A = 0000$
1000
1000

A	Q	Q_{-1}	Op
0000	0110	0	Right shift
0000	0011	0	$A = 1000$

#1

0000 0011 1 → 0

A = 1000

Right shift

#2 1000 0011 → 1

#3 1100 0001 → 1

#4 1110 0000

0110 0000

0011 0000

1110

1000

0110

110000

32 16 8 4 2 1

32+16 = 48