Booth's Algorithm – Signed Multiplication

The Booth's principle states that

"The value of a series of 1's of binary can be given as the weight of the bit preceding the series minus the weight of the last bit in the series"

For example

1. $(011111111110)_2 = 2^{11} - 2^{1}$

$= 2048 - 2$

$= 2046$

If this is the multiplier of multiplication, then in the shift and add method, we would have to do 10 addition operations,

i.e $2^{10} + 2^9 + 2^8 + \dots 2^1 = 2046$

But in case of Booth's algorithm, we will have to do only 2 operations,

i.e $2^{11} - 2^1$.

Hence, this is called as the Best Case condition of Booth's Algorithm when the series of 1's is very large.

2. $(01010101)_2$

$\Rightarrow 2^7 - 2^6 + 2^5 - 2^4 + 2^3 - 2^2 + 2^1 - 2^0$

$\Rightarrow 128 - 64 + 32 - 16 + 8 - 4 + 2 - 1$

$\Rightarrow (83)_{10}$

If this is the multiplier of multiplication, then in the shift and add method, we would have to do 4 addition operations

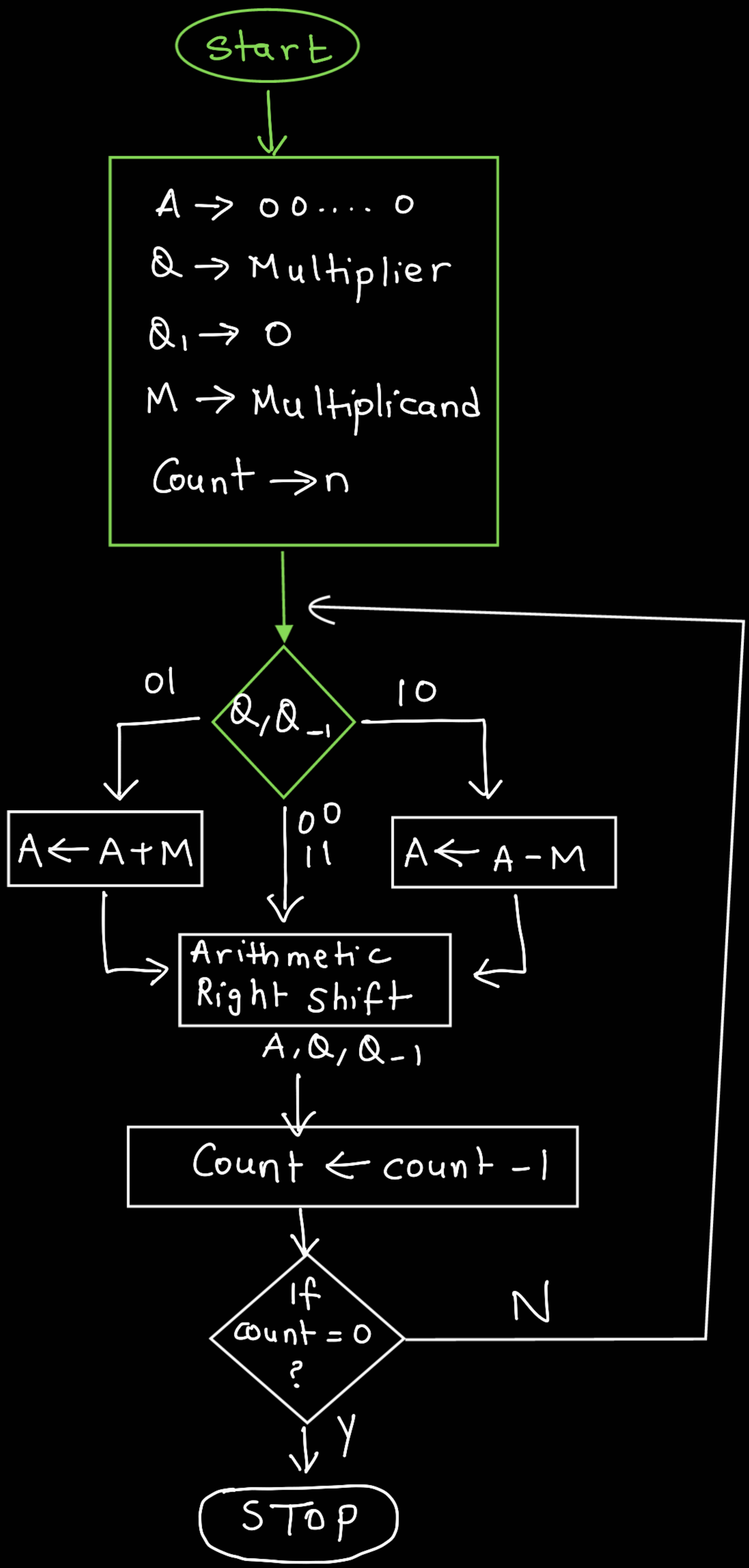
i.e $2^6 + 2^4 + 2^2 + 2^0 = 102$

But in case of Booth's algorithm, we will have to do 8 operations.

i.e $2^7 - 2^6 + 2^5 - 2^4 + 2^3 - 2^2 + 2^1 - 2^0$

Hence, this is called as the "Worst Case" Scenario.

Flowchart



Start
A → 00.... 0
Q → Multiplier
Q₁ → 0
M → Multiplicand
Count → n
01
Q,Q₋₁
10
A ← A+M
00
11
A ← A−M
Arithmetic Right shift
A, Q, Q₋₁
Count ← count −1
If count = 0 ?
N
Y
STOP

Q.1. Multiply 7×-3 using Booth's Algorithm.

Sol: $7 \times (-3)$

M ↗ 7, Q ↗ −3

2's complement of 3

$\Rightarrow$ 2's complement of (0011)

$\Rightarrow$ 1101

$\therefore$ M = 0111 Also −M = −(7)

Q = 1101 $\Rightarrow$ 1001

Tracing Table

A	Q	Q-1	m	Count	Action
0000	1101	0	0111	4	Initialization
+ 1001					A = A + (−M)
= 1001	1101	0			ASR
1100	1110	1		3	n−1

+ 0111		1			A = A + M
= 0011	1110	1			ASR
0001	1111	0		2	n − 1
+ 1001					A = A + (−M)
= 1010	1111	0			ASR
1101	0111	1			−1
1110	1011	1		0	ASR, n − 1

To obtain final result in negative form, subtract 1 from the result & complement the bits

$$\therefore \quad \begin{array}{r} 11101011 \\ -\quad 1 \\ \hline 11101010 \end{array}$$

Now complement it $\Rightarrow (00010101)_2$

$\Rightarrow -(21)_{10}$

Final Answer

Q.2 Multiply -7×-3 using Booth's Algorithm.

Sol: $M = -7$ = 2's complement of (0111)

$= (1001)_2$

$-M = -(-7) = (0111)_2$

$Q = -3$ = 2's complement of (0011)

$= 1101$

Tracing Table

A	Q	Q_{-1}	Count	Action
0000	1101	0	4	Initialize
+ 0111				
0111	1101	0		$A \leftarrow A + (-M)$
0011	1110	1	3	ASR, Count--
+ 1001				
1100	1110	1		$A \leftarrow A + M$
1110	0111	0	2	ASR, Count --

A	Q	Q -1	Count	Action
+ 0111				
0101	0111	0		A ← A+(-M)
0010	1011	1	1	ASR, Count--
0001	0101	1	0	ASR

Final Answer $= (00010101)_2$

$= (21)_{10}$

Practice

1. Multiply 13x−11

2. Multiply 16x15

3. M= 110011, Q= 101100

Significance of Booth's Principle

$M = 0110011$

$Q = 0101100$

If we apply Booth's principle,

Q can be expressed as

$$= (2^6 - 2^5) + (2^4 - 2^2)$$

Method 1

$$= (64 - 32) + (16 - 4)$$

$$= 32 + 12$$

$$= 44$$

Or we can opt for the regular way, ie add only the positions, where 1's are present.

$$\Rightarrow 2^2 + 2^3 + 2^5$$

Method 2

$$\Rightarrow 4 + 8 + 32 = 44$$

Method 2 has a slight edge over the 1st method as it consists of only addition operations & will not use any subtractions.

The CPU subsequently makes intelligent decisions based on the efficiency needs.

Q.3 $\underline{(-12) \times (-18)} \rightarrow$ 5 bits

4 bits $\nearrow$

M $\nearrow$ Q $\uparrow$

bits $\Rightarrow 5+1$
$\Rightarrow 6$

As both the numbers are negative, we would have to convert them into their 2's complement form.

$\therefore$ New M = 2's complement of (old M)

$\therefore$ M = 2's complement of (001100)

$= 110011 + 1$

$= 110100$

$-M = -(-12) = 12$

$= 001100$

Similarly, for Q,

Q = −18

∴ New Q = 2's complement of (010010)

= 101101 + 1

= 101110

A	Q	Q_{-1}	Count (C)	Action
000000	101110	0	6	Initialize
000000	010111	0	5	ASR, C--
001100	010111	0	5	A ← A − M
000110	001011	1	4	ASR, C --
000011	000101	1	3	ASR, C --
000001	100010	1	2	ASR, C --
110101	100010	1	2	A ← A + M
111010	110001	0	1	ASR, C --

A	Q	Q_{-1}	C	Action
000110	110001	0	1	$A \leftarrow A - M$
000011	011000	1	0	ASR, C--

Product = 000011011000

= $(216)_{10}$

Q.4: Multiply (−7) with 4 by using Booth's Algorithm for multiplication

Q.5: Multiply (17) with 5.

Solution Problem #3

$A = 110011$ Multiplicand

$B = 101100$ Multiplier

As the MSB bit indicates sign, the numbers reduce to :-

$A = -19 = M$

$B = -12 = Q$

Now the calculation can be continued in the same way

We will need 6 bits to represent these numbers

$\therefore A = M = $ 2's complement of (010011)

$= 101100 + 1$

$= 101101$

$-M = -(-19) = 010011$

$Q = 001100$

$\therefore \; -Q = 110011 + 1$

$= 110100$

A	Q	Q_{-1}	Count	Action
000000	001100	0	6	Initialize
000000	000110	0	5	ASR, Count--
000000	000011	0	4	"
+ 010011				
010011				A ← A − M
001001	100001	1	3	ASR, Count--
000100	110000	1	2	ASR, Count--
+ 101101				
110001				A ← A + M
111000	111000	0	1	ASR, Count--
111100	011100	0	0	ASR, Count--

$\Rightarrow$ 111100 011100

As the MSB is Negative, hence will subtract 1 from the value & complement the bits to get the answer in the true form.

$$
\begin{array}{r}
111100\ 011100 \\
-\quad 1 \\
\hline
111100011011
\end{array}
$$

Complement

$$
\underline{000011100100}
$$

$$
= (228)_{10}
$$