UNSIGNED MULTIPLICATION: CONVENTIONAL METHOD / PENCIL–PAPER METHOD

1) The Conventional (Pencil-Paper) method is used to **multiply two unsigned numbers**.
2) When we multiply two **"N-bit"** numbers, the answer is "**2 x N**" bits.
3) Three registers A, Q and M, are used for this process.
4) **Q** contains the **Multiplier** and **M** contains the **Multiplicand**.
5) **A** (**Accumulator**) is initialized with 0.
6) At the end of the operation, the **Result** will be stored in (**A & Q**) combined.
7) The process involves **addition** and **shifting**.

Algorithm:

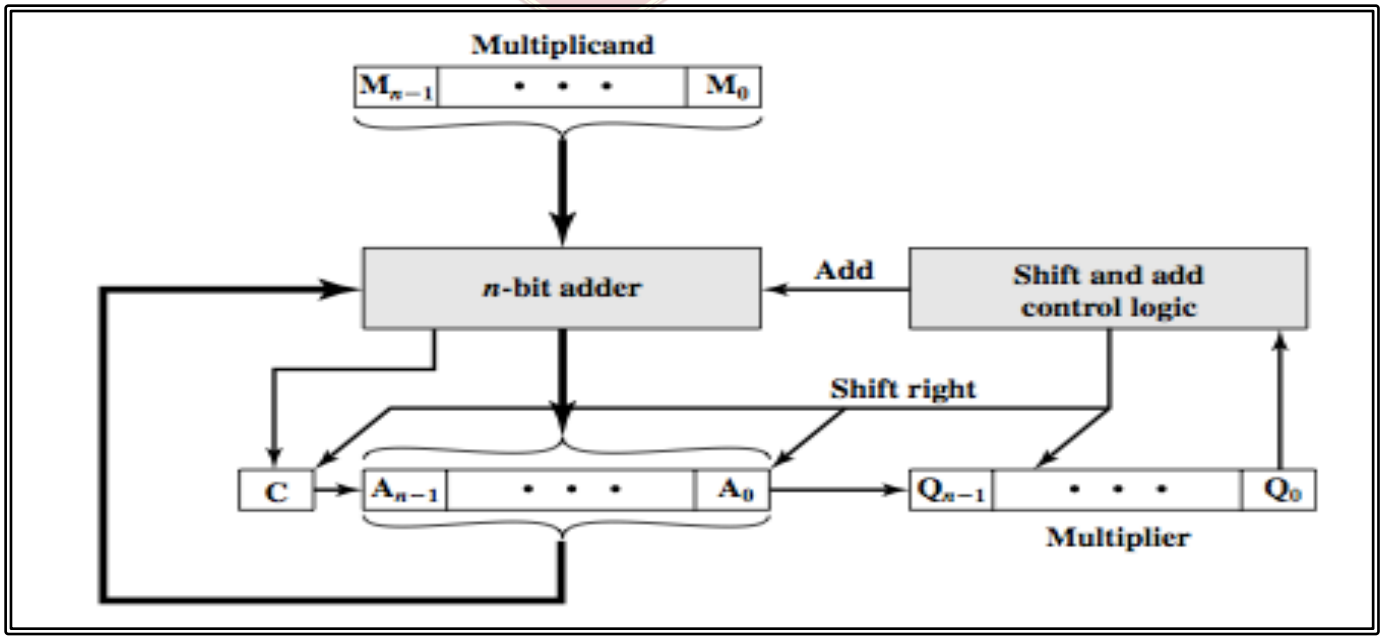
The **number of steps** required is equal to the **number of bits in the multiplier**.

1) At each step, **examine** the current **multiplier bit** starting from the **LSB**.
2) If the current **multiplier bit is "1"**, then the **Partial-Product** is the **Multiplicand** itself.
3) If the current **multiplier bit is "0"**, then the **Partial-Product** is the **Zero**.
4) At each step, **ADD the Partial-Product to the Accumulator**.
5) Now **Right-Shift the Result** produced so far (**A & Q combined**).

Repeat steps 1 to 5 for **all bits** of the multiplier.

The **final answer** will be in **A & Q** combined.

Circuit Diagram for Unsigned Multiplication



Example: 7 x 6 = 42

```
                    0  1  1  1   … Multiplicand (7)
          X         0  1  1  0   … Multiplier (6)
-------------------------------------------------
                    0  0  0  0   … Partial-Product
                 0  1  1  1  X              ”
              0  1  1  1  X  X              ”
+ 0  0  0  0  X  X  X              ”
-------------------------------------------------
  0  1  0  1  0  1  0   … Result (42)
-------------------------------------------------
```

Step	C Carry	A Accumulator	Q Multiplier	M Multiplicand	Explanation
	0	0000	0110	0111	*Initial Value*
1	0 0	0000 0000	0110 0011		*Current Multiplier bit is “0” so ADD “0” to Accumulator and Right-Shift*
2	0 0	0111 0011	0011 1001		*Current Multiplier bit is “1” so ADD Multiplicand to Accumulator and Right-Shift*
3	0 0	1010 0101	1001 0100		*Current Multiplier bit is “1” so ADD Multiplicand to Accumulator and Right-Shift*
4	0 0	0101 0010 FINAL ANSWER :-)	0100 1010		*Current Multiplier bit is “0” so ADD “0” to Accumulator and Right-Shift*
For more detailed explanation refer Bharat Sir’s Lecture Notes or Call Bharat Sir on 9820408217.					

Example: 15 x 15 = 225

```
                    1   1   1   1   … Multiplicand (15)
          X         1   1   1   1   … Multiplier (15)
-------------------------------------------------------
                    1   1   1   1   … Partial-Product
                1   1   1   1   X               ”
            1   1   1   1   X   X               ”
        1   1   1   1   X   X   X               ”
+   1   1   1   1   X   X   X   X               ”
-------------------------------------------------------
1   1   1   0   0   0   0   1   … Result (225)
-------------------------------------------------------
```

Step	C Carry	A Accumulator	Q Multiplier	M Multiplicand	Explanation
	0	0000	1111	1111	*Initial Value*
1	0 0	1111 0111	1111 1111		*Current Multiplier bit is “1” so ADD Multiplicand to Accumulator and Right-Shift*
2	1 0	0110 1011	1111 0111		*Current Multiplier bit is “1” so ADD Multiplicand to Accumulator and Right-Shift*
3	1 0	1010 1101	0111 0011		*Current Multiplier bit is “1” so ADD Multiplicand to Accumulator and Right-Shift*
4	1 0	1100 1110 FINAL ANSWER :-)	0011 0001		*Current Multiplier bit is “1” so ADD Multiplicand to Accumulator and Right-Shift*
For more detailed explanation refer Bharat Sir’s Lecture Notes or Call Bharat Sir on 9820408217.					

SIGNED MULTIPLICATION: BOOTH'S ALGORITHM

1) Booth's Algorithm is used to **multiply two SIGNED numbers**.
2) When we multiply two **"N-bit"** numbers, the answer is "**2 x N**" bits.
3) Three registers A, Q and M, are used for this process.
4) **Q** contains the **Multiplier** and **M** contains the **Multiplicand**.
5) **A** (**Accumulator**) is initialized with 0.
6) At the end of the operation, the **Result** will be stored in (**A & Q**) combined.
7) The process involves **addition, subtraction** and **shifting**.

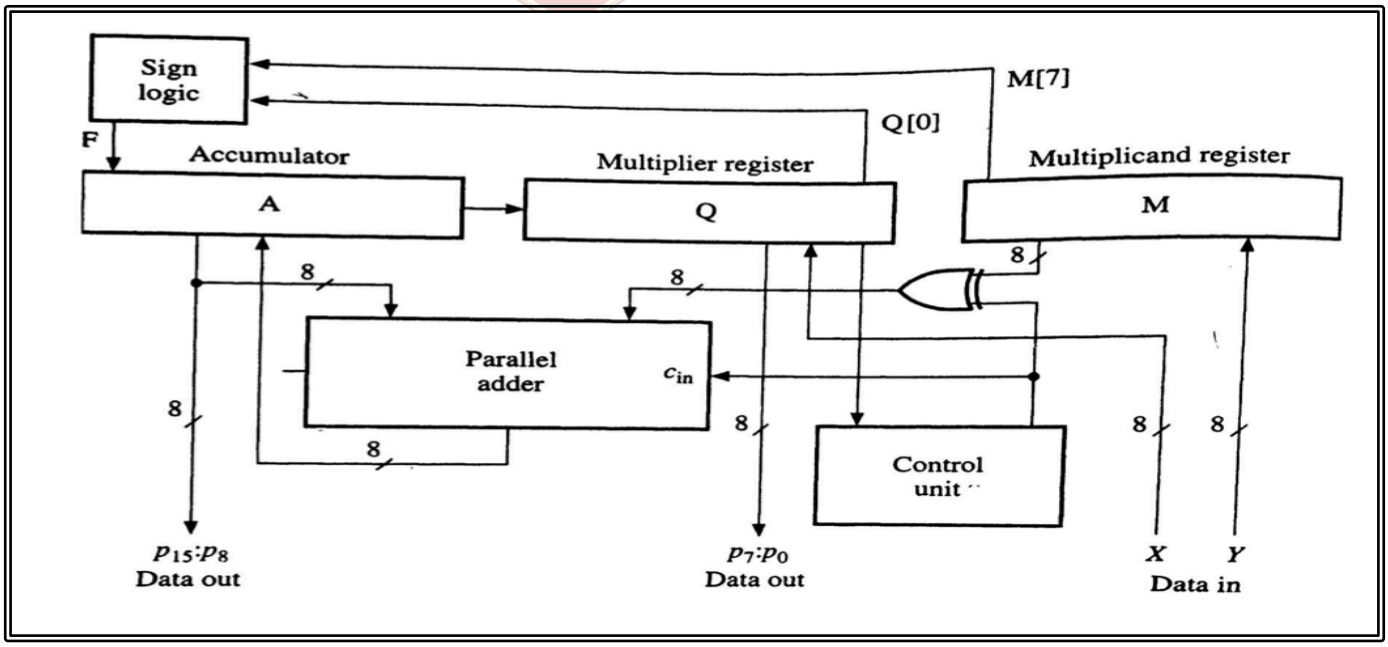
Algorithm:

The **number of steps** required is equal to the **number of bits in the multiplier**.
At the beginning, consider an **imaginary "0" beyond LSB of Multiplier**

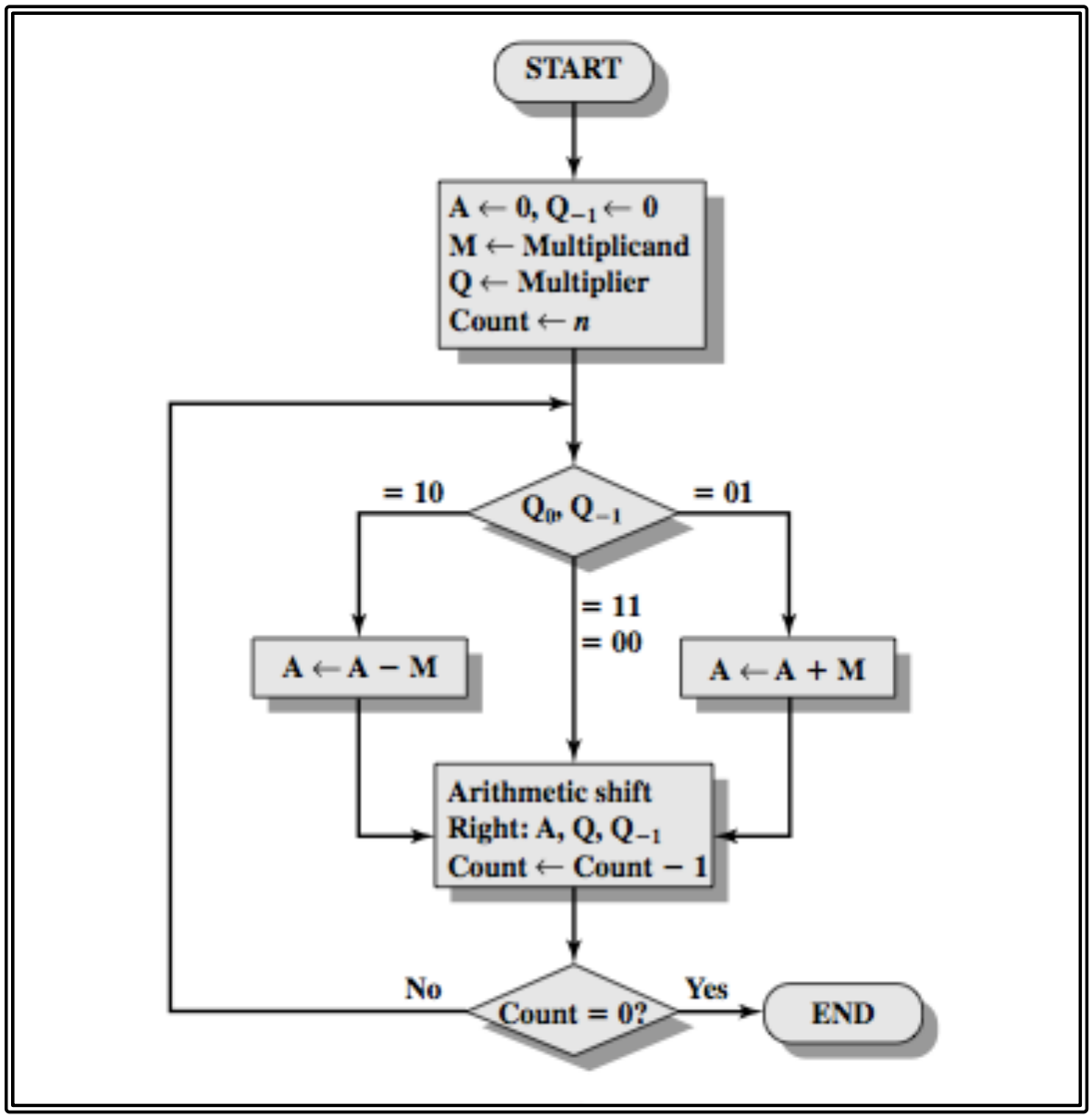
1) At each step, **examine two adjacent Multiplier bits** from **Right to Left**.
2) If the transition is from **"0 to 1"** then **Subtract M** from **A** and **Right-Shift** (A & Q) combined.
3) If the transition is from **"1 to 0"** then **ADD M** to **A** and **Right-Shift**.
4) If the transition is from **"0 to 0"** then **simply Right-Shift**.
5) If the transition is from **"1 to 1"** then **simply Right-Shift**.

Repeat steps 1 to 5 for **all bits** of the multiplier.
The **final answer** will be in **A & Q** combined.

Circuit Diagram for Signed Multiplication using Booth's Algorithm



FLOWCHART FOR BOOTH'S ALGORITHM



Example: 7 x 6 = 42

Multiplicand (M): **7 = 0111**. **-7 = 1001** (Two's Complement Form)
Multiplier (Q): **6 = 0110**. **-6 = 1010** (Two's Complement Form)

Step	A Accumulator	Q Multiplier	Q_{-1} Imaginary	M Multiplicand
Initial	0000	0110	0	**0111**
1) (0 ← 0) No Add or Sub Right-Shift	0000 0000	0110 0011	0 0	
2) (1 ← 0) Perform **(A - M)** Right-Shift	1001 1100	0011 1001	0 1	
3) (1 ← 1) No Add or Sub Right-Shift	1100 1110	1001 0100	1 1	
4) (0 ← 1) Perform **(A + M)** Right-Shift	0101 **0010** **FINAL ANSWER :-)**	0100 **1010**	1 **0**	

For more detailed explanation refer Bharat Sir's Lecture Notes or Call Bharat Sir on 9820408217.

Example: 5 x 7 = 35

Multiplicand (M): **5 = 0101**. **-5 = 1011** (Two's Complement Form)
Multiplier (Q): **7 = 0111**. **-7 = 1001** (Two's Complement Form)

Step	A Accumulator	Q Multiplier	Q_{-1} Imaginary	M Multiplicand
Initial	0000	0111	0	**0101**
1) (1 ← 0) Perform **(A - M)** Right-Shift	1011 1101	0111 1011	0 1	
2) (1 ← 1) No Add or Sub Right-Shift	1101 1110	1011 1101	1 1	
3) (1 ← 1) No Add or Sub Right-Shift	1110 1111	1101 0110	1 1	
4) (0 ← 1) Perform **(A + M)** Right-Shift	0100 0010	0110 0011	1 **0**	
	FINAL ANSWER :-)			
For more detailed explanation refer Bharat Sir's Lecture Notes or Call Bharat Sir on 9820408217.				

Example: 9 x 10 = 90

Multiplicand (M): **9 = 01001**. **-9 = 10111** (Two's Complement Form)
Multiplier (Q): **10 = 01010**. **-10 = 10110** (Two's Complement Form)

Step	A Accumulator	Q Multiplier	Q_{-1} Imaginary	M Multiplicand
Initial	00000	01010	**0**	**01001**
1) (0 ← 0) No Add or Sub Right-Shift	00000 00000	01010 00101	**0** **0**	
2) (1 ← 0) Perform **(A - M)** Right-Shift	10111 11011	00101 10010	**0** **1**	
3) (0 ← 1) Perform **(A + M)** Right-Shift	00100 00010	10010 01001	**1** **0**	
4) (1 ← 0) Perform **(A - M)** Right-Shift	11001 11100	01001 10100	**0** **1**	
5) (0 ← 1) Perform **(A + M)** Right-Shift	00101 00010 **FINAL ANSWER :-)**	10100 11010	**1** **0**	
Practice more sums by yourself. In case of doubts call #bharatsir at 9820408217.				

Expected Ans: 90 = 00010 11010

Example: -9 x 10 = -90

Multiplicand (M): **-9 = 10111** **9 = 01001**. (Two's Complement Form)
Multiplier (Q): **10 = 01010**. **-10 = 10110** (Two's Complement Form)

Step	A Accumulator	Q Multiplier	Q_{-1} Imaginary	M Multiplicand
Initial	00000	01010	**0**	**10111**
1) (0 ← 0) No Add or Sub Right-Shift	00000 00000	01010 00101	**0** **0**	
2) (1 ← 0) Perform **(A - M)** Right-Shift	01001 00100	00101 10010	**0** **1**	
3) (0 ← 1) Perform **(A + M)** Right-Shift	11011 11101	10010 11001	**1** **0**	
4) (1 ← 0) Perform **(A - M)** Right-Shift	00110 00011	11001 01100	**0** **1**	
5) (0 ← 1) Perform **(A + M)** Right-Shift	11010 **11101**	01100 **00110**	**1** **0**	
	FINAL ANSWER :-)			
Practice more sums by yourself. In case of doubts call #bharatsir at 9820408217.				

Expected Ans: -90 = 11101 00110

Example: 9 x -10 = -90

Multiplicand (M): **9 = 01001** **-9 = 10111**. (Two's Complement Form)
Multiplier (Q): **-10 = 10110** **10 = 01010**. (Two's Complement Form)

Step	A Accumulator	Q Multiplier	Q_{-1} Imaginary	M Multiplicand
Initial	00000	10110	0	01001
1) (0 ← 0) No Add or Sub Right-Shift	00000 00000	10110 01011	0 0	
2) (1 ← 0) Perform **(A - M)** Right-Shift	10111 11011	01011 10101	0 1	
3) (1 ← 1) No Add or Sub Right-Shift	11011 11101	10101 11010	1 1	
4) (0 ← 1) Perform **(A + M)** Right-Shift	00110 00011	11010 01101	1 0	
5) (1 ← 0) Perform **(A - M)** Right-Shift	11010 **11101** **FINAL ANSWER :-)**	01101 **00110**	0 **1**	
Practice more sums by yourself. In case of doubts call #bharatsir at 9820408217.				

Expected Ans: -90 = 11101 00110

Example: -9 x -10 = 90

Multiplicand (M): **-9 = 10111** **9 = 01001**. (Two's Complement Form)
Multiplier (Q): **-10 = 10110** **10 = 01010**. (Two's Complement Form)

Step	A Accumulator	Q Multiplier	Q_{-1} Imaginary	M Multiplicand
Initial	00000	10110	0	10111
1) (0 ← 0) No Add or Sub Right-Shift	00000 00000	10110 01011	0 0	
2) (1 ← 0) Perform **(A - M)** Right-Shift	01001 00100	01011 10101	0 1	
3) (1 ← 1) No Add or Sub Right-Shift	00100 00010	10101 01010	1 1	
4) (0 ← 1) Perform **(A + M)** Right-Shift	11001 11100	01010 10101	1 0	
5) (1 ← 0) Perform **(A - M)** Right-Shift	00101 **00010**	10101 **11010**	0 **0**	
	FINAL ANSWER :-)			
Practice more sums by yourself. In case of doubts call #bharatsir at 9820408217.				

Expected Ans: 90 = 00010 11010