1) Multiplication of Positive no's.
2) Binary Multiplier
   * Shift - AND - ADD multiplier.
3) Signed multiplication. - Booth Algm

Fast multiplication -
     ↳ Bit pair recording of multipliers
Integer Division.
Restoring Division & Non restoring Division
floating point nos & operation
IEEE stands for floaty point no'.
                    * Carry-save multiplication

Multiplication Of Positive nos :-

Multiplicand (B) = $(13)_{10}$ = 1101 → B_0

Multiplier (Q) = $(11)_{10}$ = 1011 → Q_0

Algm :-

1) C → Single bit (initially 0)
   A → n bit (initially 0)

   B -   n bit - multiplicand
   Q -   n bit - multiplier.

Count is a variable is equal
to n

2) Check the Q_0 value

☆ $Q_0 = 1$ means perform

i) Add $A \leftarrow A + B$
    if carry is generated store
                                  it in $\underline{C}$

ii) Shift left to right
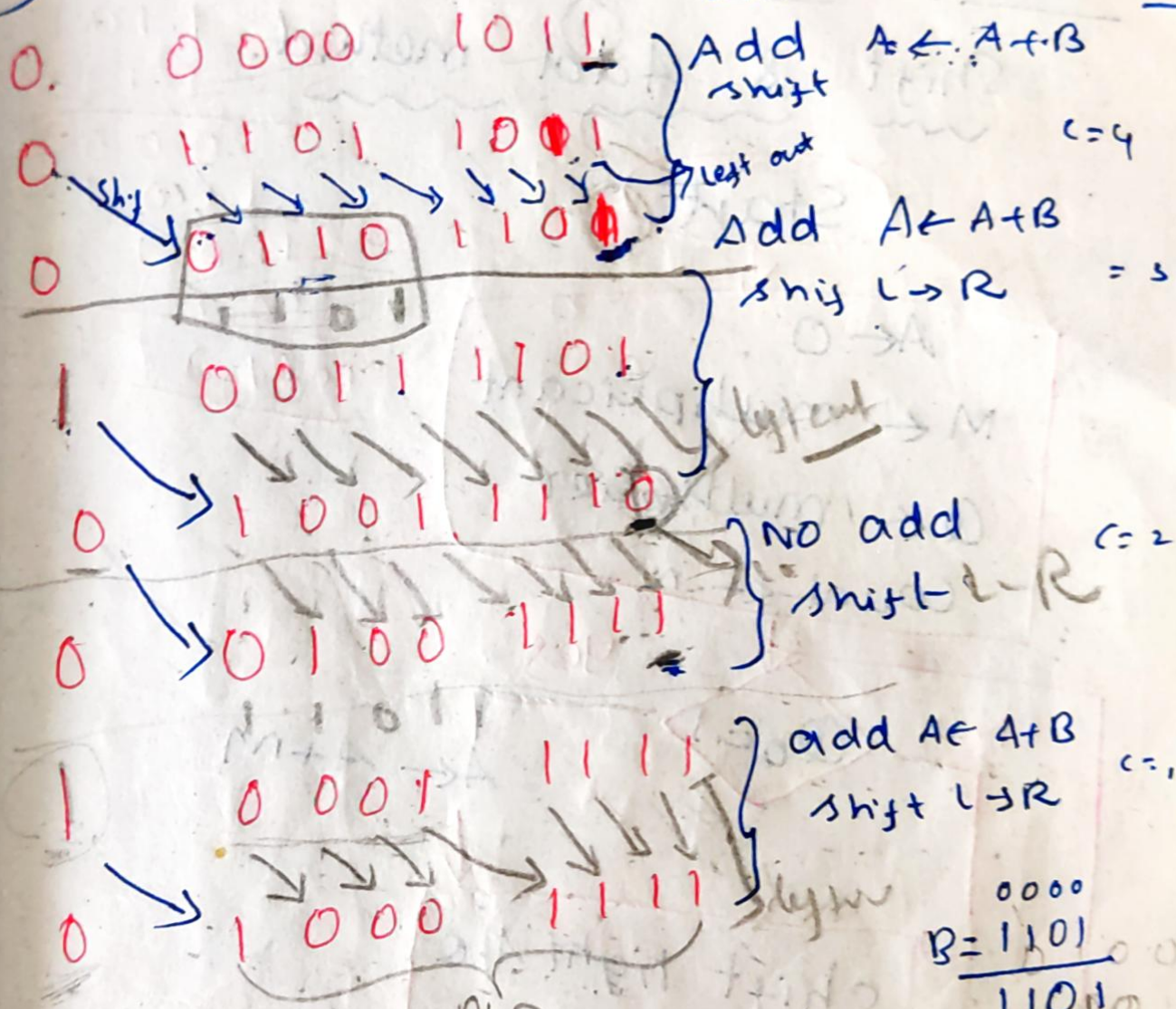
iii) reduce count value (C, A, Q)

$Q_0 = 0$

i) No add.

ii) just shift $L \rightarrow R$

iii) Reduce count value

3) Process value continue until the count
value becomes a zero.

4) Result is store in A & Q reg.

Ex 33

n = 4

Initial condition 4

C	A	Q

0.   0 000   1011   Add   A ← A+B
                             shift

0.   1 101   1001   left out   C = 4
shy
0   0 110   1100   Add   A ← A+B
                             shy L → R   = 3

1   0 011   1101   left out → M

0   1 001   1110   No add   C = 2
                             shift L → R

0   0 100   1111   add A ← A+B   C = 1
                             shift L → R

1   0 001   1111   dyw

0   1 000   1111

                             0000
                          B = 1101
                             1101
                             1101
                             0110
                            1000.

13 × 11 = 143 = 1000 1111

```
  1101  ×  1011
     1101
    1101
   0000
  1101
 1000 1 0 1 1 1
```

0110
1101
0011

Binary Multiplication

Shift & Add method.

```
   1 1 0 1
   1 1 1 0
  ───────
   0 0 0 0
   1 1 0 1
  1 1 0 1
 1 0 0 1
───────────
1 0 0 1 0 1 1 0
```


Flowchart:

- **Start**
- A ← 0
  M ← multiplicant
  Q ← multiplier
  Count ← N
- Q0 = 1 ?
  - **yes** → A ← A + M
  - **no**
- Shift right A, Q
  Count = Count − 1
- Count = 0 ?
  - **yes** → **End**

Eg: $1101 \times 1011 = 10001111$

$13 \times 11 = 143$

Q — mejm M

Booth

C	A	Q	M	
			1011	4
0	0000	1101		$Q_0 = 1$
	+ 1011			Add = A+m
0	1011	1101		shift
0	0101	1110	M = M	$Q_0 = 0$
0	0010	1111		shift
0	1101	1111		$Q_0 = 1$
0	0110	1111		Add A+n Shit
	1011			$Q_0 = 1$
0	0000	1111		Add A+n shit
	1000	1111		

o/p // Q

P

1   0000
    10001

10000
  0000
  00
  00

10001 0000 0000

12 = B
15 = Q

21
00 0016 = 13
00 0117 = (