Computer Arithmetic

decimal to binary

$(18)_{10} \rightarrow (\quad)_2$

$2 \mid 18$
$2 \mid 9 \quad 0$
$2 \mid 4 \quad 1$
$2 \mid 2 \quad 0$
$2 \mid 1 \quad 0$
$0 \quad 1$

$\Rightarrow (10010)_2$

Converting to 8-bit

$(00010010)_2$

defines weither -ve or +ve

However this method as drawback as '0' can be written as

$00000000 = 0$
$10000000 = -0$

General Case $\Rightarrow$

$$A = \begin{cases} \sum_{i=0}^{n-2} 2^i a_i & \text{if } a_{n-1} = 0 \\ -\sum_{i=0}^{n-2} 2^i a_i & \text{if } a_{n-1} = 1 \end{cases}$$

$-\sum_{i=0}^{n-2} 2^i a_i$ if $a_{n-1} = 1$

Thus max limit of 8-bit ⇒ $-127 \sim 127$

Representing -ve no in 2's Complement

Example,

18 = 10010

Thus for -18

18 = 000 10010

1's ⇒ 111 01101

\+ 1

2's ⇒ 111 01110 = -18

↓

1 1101110

↓ ↓

-128 +64 + 32 + 0 + 0 + 8 + 4 + 2 + 0 = -18

Generalization of 2's Complement

$n - \sum_{i=0}^{n-2} 2^i a_i$ for $n > 0$

$C = 0$

∴ Range of +ve Number $\Rightarrow -1 \text{ to } +2^{n-1}$

total range $-2^{n-1} \text{ to } 2^{n-1}$

Note : Addition and Substraction are easy in binary do it on your own

In Substraction we just add 2's Compliment

Multiplication of binary

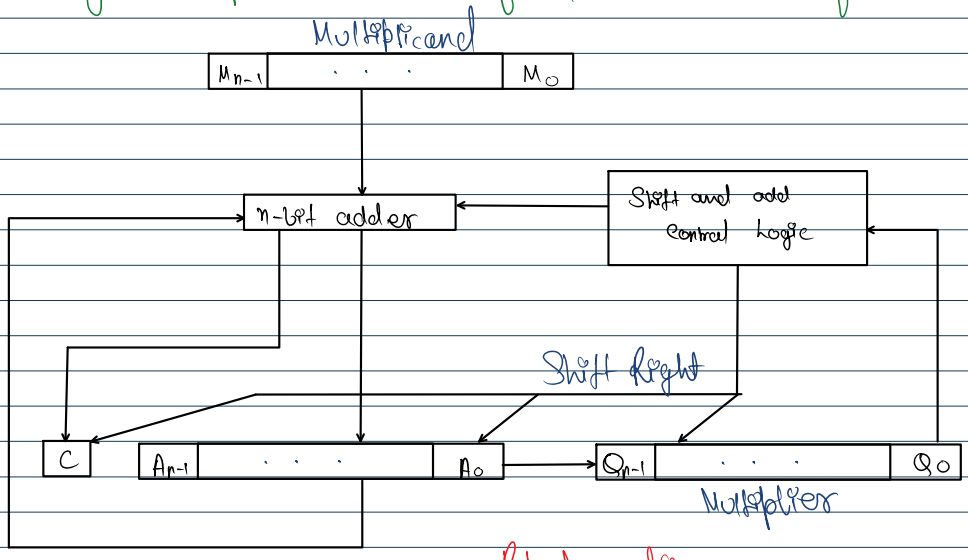
Multiplication of two n-bit number gives result upto 2n-bit number

Example,

$$\begin{array}{r} 47 \\ \times\ 33 \\ \hline 1551 \end{array} \rightarrow \begin{array}{r} 101111 \\ \times\ 100001 \\ \hline 101111 \\ \hline 101111xxxxxx \\ \hline (1551)_2 \end{array} \leftarrow (11000001111)_2$$

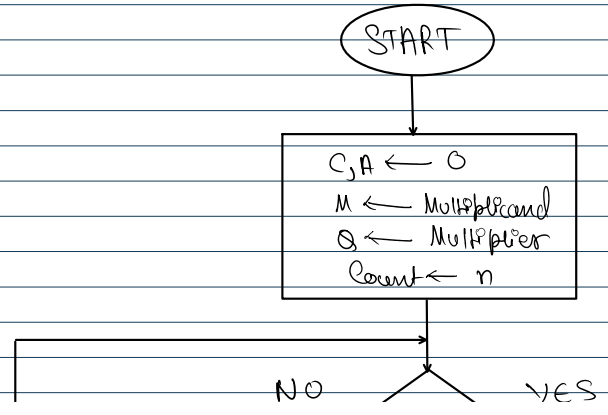
101111 — Multiplicand
100001 — multiplier
101111 — partial product
101111xxxxxx — partial product
(1551)$_2$ ← (11000001111)$_2$ — product

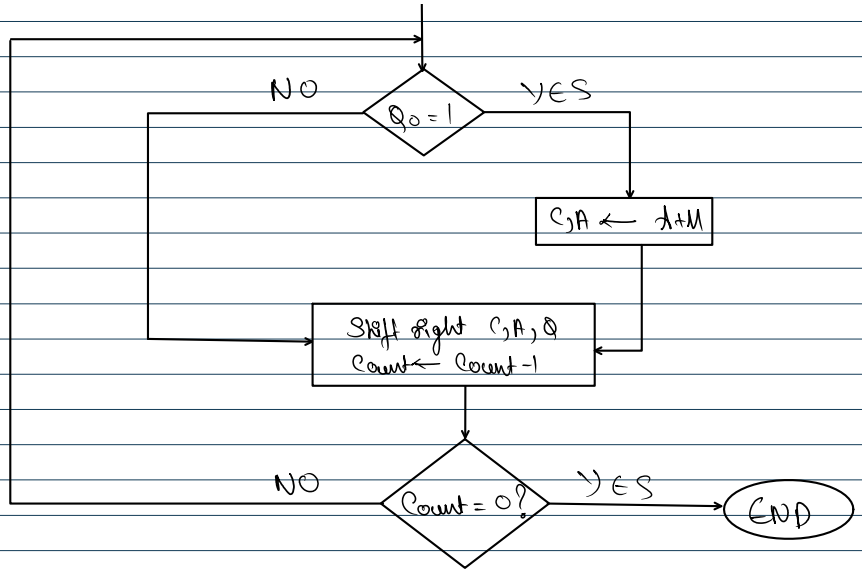
Logical Representation of Multiplication of +ve Binary



A, C, M ; Register

↳ Block diagram





∽ Flowchart

Example, 11×13 ∽ 1101×1011 (Multiplier (m13), Multiplicand (m11))

C	A	Q	M			
0	0000	1101	1011	$Q_0 = 1$, C,A ← A+M		n = 4
0	1011	1101	1011	Added	1st cycle	n = 3
0	0101	1110	1011	Shift right		
0	0010	1111	1011	Shift	2nd cycle	n = 2
0	1101	1111	1011	Add	3rd cycle	n = 1
0	0110	1111	1011	Shift		
1	0001	1111	1011	Add	4th cycle	n = 0 ↓ end
0	1000	1111	1011	Shift		

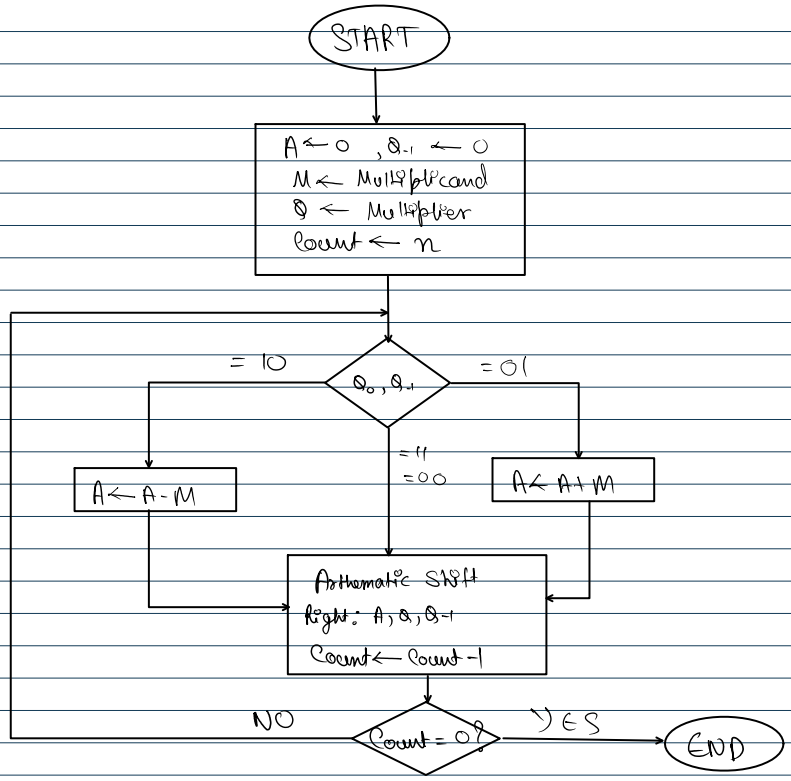
(A+M) = A; A = A+M; carried over

Result : $(10001111)_2 = (143)_{10}$

n°; number of bits in digits

Manual process of multiplication remains Same ;
just convert to 2's Compliment

However the flow chart / Logic changes



Note It is important to convert all -ve values to 2's Compliment

Example,

7 x 3 ⇝ 0111 × 0011 (- M =) 1001
n=4 n=3
M Q

A	Q	Q-1	M	-M (2's Comp.)			
0000	0011 ····› 10 ‹···· 0 (A ← A - M)		0111	1001	n = 4		
0001	0011	0	0111	1001	A = A - M	n = 3	1st Cycle
1100	1001 ···› 11 ‹··· 1 (Shift)		0111	1001	Arithmetic Shift		
1110	0100 ····› 01 ‹···· 1 (Arithm)		0111	1001	Arithmetic Shift	n = 2	2nd Cycle
1 1 0 1 0 1 (Sign bit ignored)	0100	1	0111	1001	A = A + M	n = 1	3rd Cycle
0010	1010 ····› 00 ‹···· 0 (Shift)		0111	1001	Arithmetic Shift		
0001	0101	0	0111	1001	Arithmetic Shift	n = 0 (END)	4th Cycle

Result : $(00010101)_2 = (21)_{10}$

This sort of Multiplication works for both +ve & -ve numbers

Division Of Binary Numbers

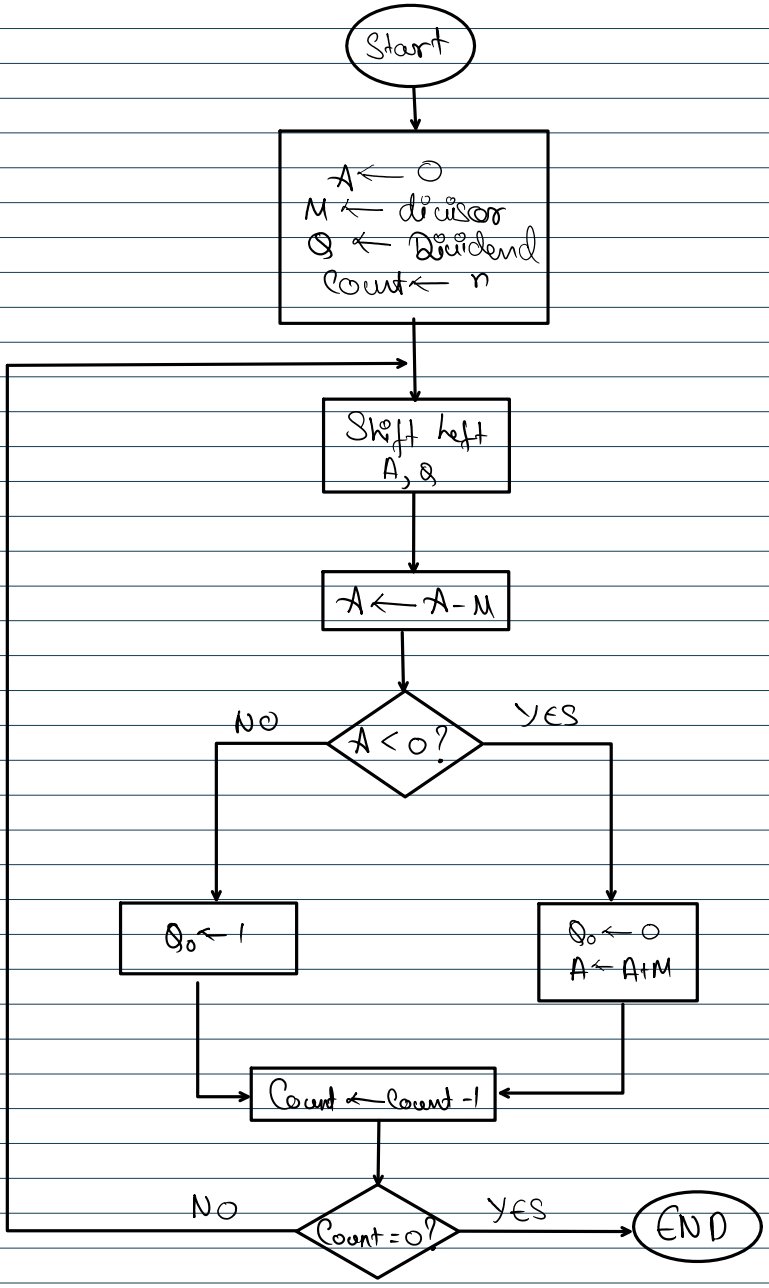
05 May 2022 05:50 PM

Manual Method

Example,

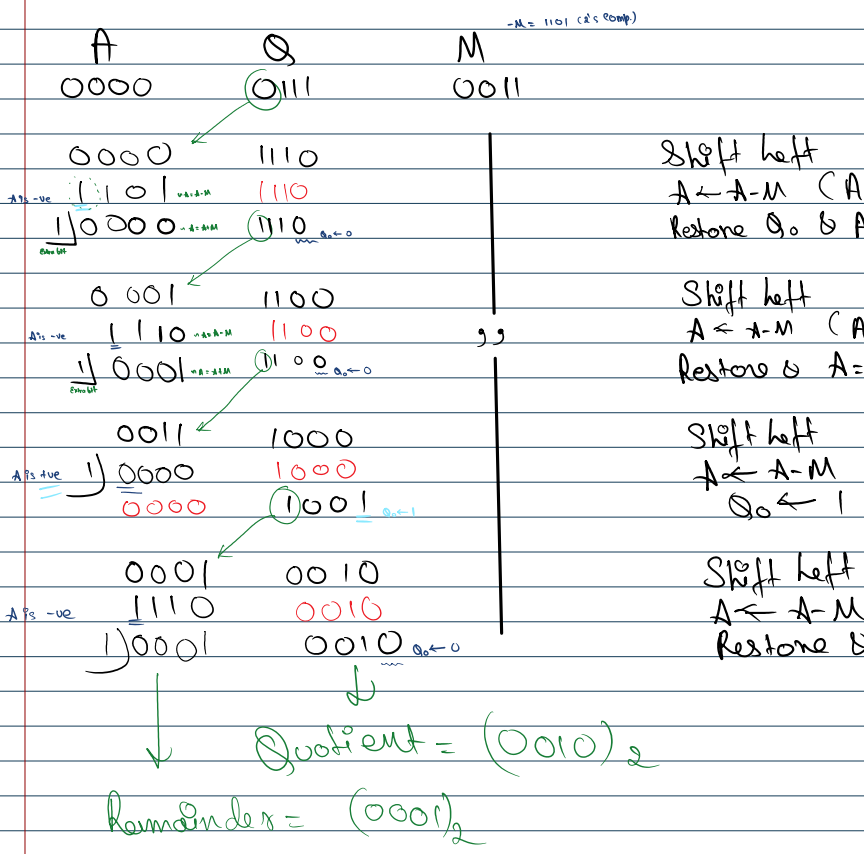
```
                00001101   Quotient
          ------------
divisor 1011 ) 10010011   dividend
               1011
               ----
                00111
                 1011
                 ----
                 0011
                  1011
                  ----
                   100   Remainder
```

Flowchart / Logic for binary division



Example,

7 ÷ 3 0111 / 0011



Shift left
A ← A-M (A<0)
Restore Q0 & A = A+M

Shift left
A ← A-M (A<0)
Restore & A = A+M

Shift left
A ← A-M (A>0)
Q0 ← 1

Shift left
A ← A-M (A<0)
Restore & A = A+M

n = 4

n = 3 } 1st cycle

n = 2 } 2nd cycle

n = 1 } 3rd cycle

n = 0 } 4th cycle

To deal with negative number

$$D = Q \times V + R$$

$$Sign(R) = Sign(D)$$

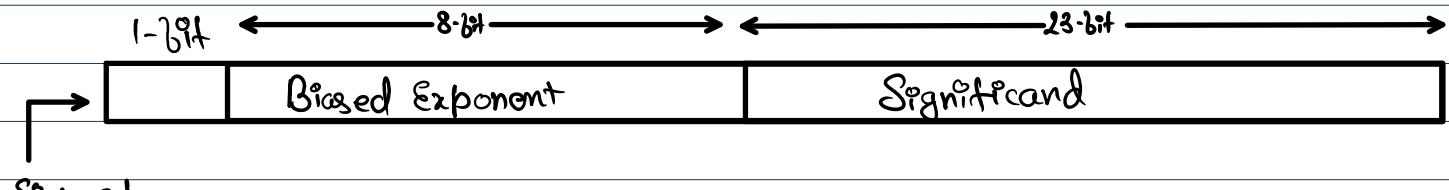
$$Sign(Q) = Sign(D) \times Sign(V)$$

D	=	Q	×	V	+	R
+		+		+		+
+		−		−		+
−		+		−		−
−		−		+		−

Floating point representation

$$\pm S \times B^{\pm E}$$

- S → Significand
- B → Base
- E → Exponent
- $\pm$ → Sign



∴ Range of biased exponent -127 to 127

Example,

+ve

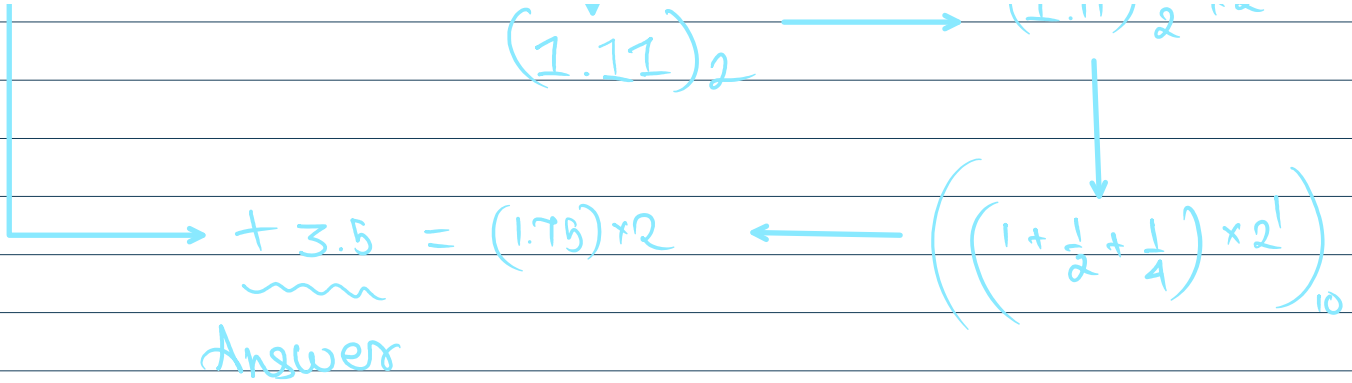
128

$128 - 127 = (1)$ → 2^1

(normalised)

$1.\_\_\_\_\_$

$(1.11)_2$ → $(1.11)_2 \times 2^1$

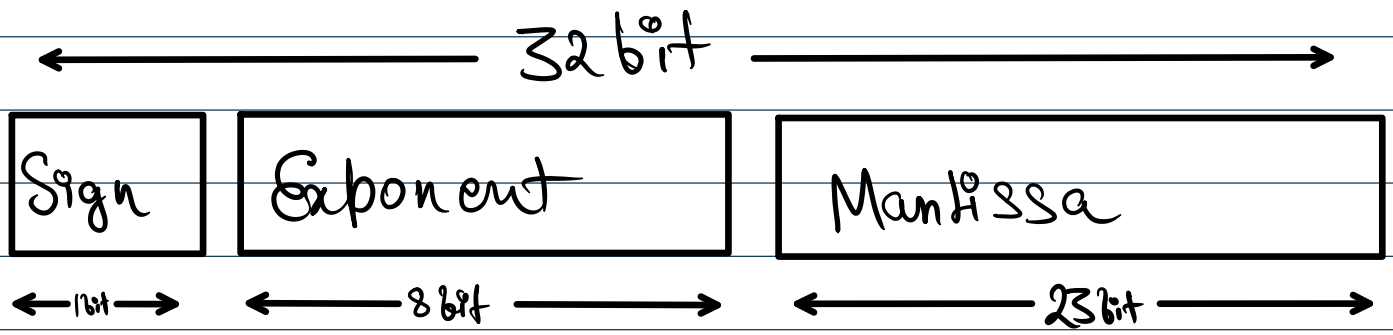


Still Some things are remaining on floating point Representation !

IEEE Standard for Floating-Point Representation

05 May 2022 10:49 PM

IEEE Standard



Single precision

(In double precision
the length is doubled ie 64 bit)
here exponent is 11bit & Mantissa 53bit

Sign : 0 represents +ve & 1 ; -ve

Biased Exponent ; represents both +ve & -ve exponent

Normalized Mantissa ; floating-point number consisting the significant digits

Example

89.125

85.125

85 + 0.125

1010101 .001

$\Rightarrow 1010101.001$

1.010101001×2^6

Sign = +ve → 0

Biased Exponent = 127 + 6 = 133

$(10000101)_2$

Normalized Mantissa = 01010011

Sign Exponent Mantissa

0 10000101 01010100

0's

Special Values

- Zero ; if has special value denoted with an exponent & mantissa 0, -0, +0 are distinct value but are equal

- Denormalized ; If exponents is all zero but mantissa isn't, this means the number isnt in Standard form of 1.—

- Infinity ; $+\infty$ & $-\infty$ are denoted with

Floating point Arithmetic

For addition & substration ensure both operand have same exponent value

↓

Thus, may require shifting radix point on one of the operand to achive alignment

Floating point operation may produce

1. **Exponent Overflow**
   Positive exponent exceeds max possible exponent value
2. **Exponent Underflow**
   A negative exponent is less than min possible exponent value (-127) (Ex. -200 < -127)
3. **Significand Underflow**
   In process of aligning, significand digits may flow off right end of significand
   { Rounding is Required }
4. **Significand Overflow**
   Addition of two significand (of same sign) results in carry out of most significant bit
   { Fixed by Re-alignment }

Floating-Point Numbers	Arithmetic Operations
$X = X_S \times B^{X_E}$ $Y = Y_S \times B^{Y_E}$	$X + Y = (X_S \times B^{X_E - Y_E} + Y_S) \times B^{Y_E}$ $X - Y = (X_S \times B^{X_E - Y_E} - Y_S) \times B^{Y_E}$ } $X_E \le Y_E$ $X \times Y = (X_S \times Y_S) \times B^{X_E + Y_E}$ $\frac{X}{Y} = \left(\frac{X_S}{Y_S}\right) \times B^{X_E - Y_E}$

Damn!

This is too complicated!