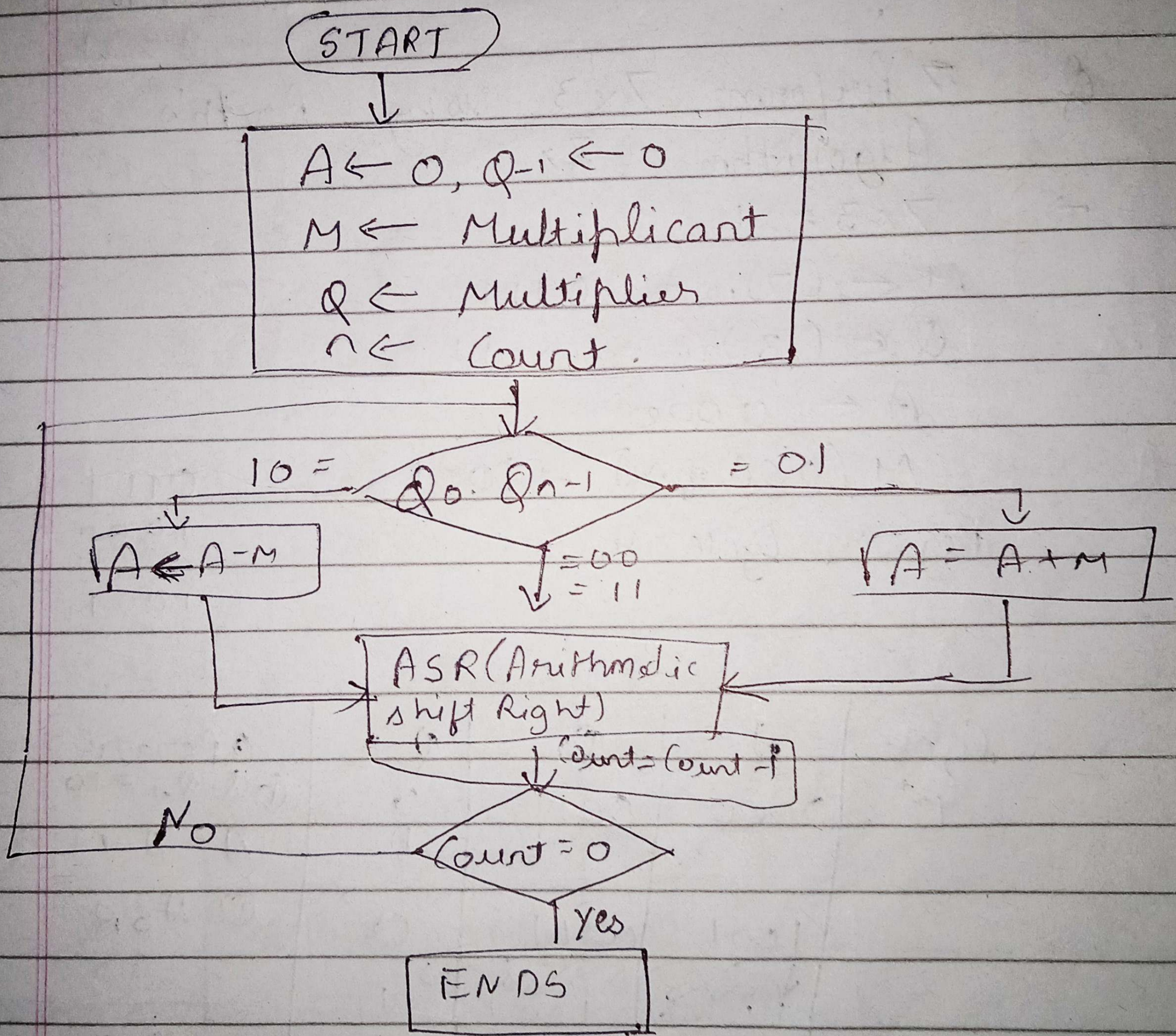
Q2. CO1 — 9 8 1/2

Q3. A CO2 — 3

Q3. B CO3 — 3 2

Module — 4

BOOTH's Algorithm



$Q \quad Q_{-1} \rightarrow$ Previous

0001 0

Shift Right 100101

□ 10010l

Arith Shift Right

100101

= 1 10010l

'n' → (n+1)

7×3, M → $(7)_{10}$ → 0111, Q → $(3)_{10}$ → 0011

Q Perform 7×3 using Booth's Algorithm,

→ 7×3 = 21

M ← $(7)_{10}$ → 0111

Q ← $(3)_{10}$ → 0011

A ← 0000

-M (2's C of M) → 1001

Reg = 4, Cycle = 4

0111
1000
+1
1001

Cycle	A	Q	Q_{-1}	Operation
1st	0000	0011	0	① $Q \cdot Q_{-1}$ = 10 A = A − M
	1001	0011	0	② ASR
	1100	1001	1	
2nd	1110	0100	1	→ $Q \cdot Q_{-1}$ = 11 ASR

Cycle	A	Q	Q-1	operation
	1110	0100	1	ASR
3rd	0101	0100	1	① Q.Q-1 = 01 A = A+M ② ASR
	0010	1010	0	
4th cycle	0001	0101	0	Q.Q-1 = 00 ASR

$$\begin{array}{r} 1110 \\ 0111 \\ \hline 10101 \end{array}$$

→ 0011

0001 0101 = 21

Q Multiply 4×6 using Booth's Algorithm

→ 4×6 = 24

M ← (4) = 0100

Q ← (6) = 0110

A ← 0000

−M (2's(+M)) → 1100

Reg 4, Cycle = 4

$$\begin{array}{r} 0100 \\ 1011 \\ +\quad 1 \\ \hline 1100 \end{array}$$

Cycle	A	Q	Q-1	Operation
1st	0000	0110	0	Q.Q-1 = 00 A.S.R
	0000	0011	0	
2nd				

=1]

cycle	A	Q	Q-1	operation
	0000	0011	0	Q.Q-1 = 10 A = A - M
step 2				
	1100	0011	0	ASK
	1110	0001	1	
				Q.Q-1 = 11 ASK
step 3	1111	0000	1	
				Q.Q-1 = 01 (1) A = A + M
step 4	0011	0000	1	
	$\boxed{0001 \quad 1000}$		0	

0000
1100
1100

1111
0100
(1)0011

$(00011000)_2 \Rightarrow (24)_{10}$

$2^3 + 2^2 + 2^1 + 0$

QHW Perform multiplication 5x3 using booth's Algorithm

Ans $5 \times 3 = 15$

$M \leftarrow (5)_{10} \rightarrow 0101$

$Q \leftarrow (3)_{10} \rightarrow 0011$

$A \leftarrow 0000$

$-M \leftarrow (2\text{'s c } +M) \Rightarrow 1011$

Reg = 4, cycle = 4

0101
1010
+1
1011

-ve x +ve = -ve
-ve x -ve = 2's complement
+ve x -ve = -ve

ation

10

M 0000
1100
1100

1

+M

1

0

1

sing

+ 0101
10011

Cycle	A	Q	Q_{-1}	Operation
			1	$Q \cdot Q_{-1} = 10$
1st	0000	0011	0	A = A - M
	1001	0011	0	A.S.R
	1100	1001	1	
2nd				A.S.R, $Q \cdot Q_{-1} = 11$
	1110	0100	1	
				$Q \cdot Q_{-1} = 01$ A = A + M
3rd	0011	1100 ~~0100~~	1	A.S.R
	0001	1110 ~~1010~~	0	A.S.R
	0001	1110	0	A.S.R
4th	0000	1111	0	

Q Multiply -7×3 by booth's Algorithm

$-7 \times 3 = -21$

→ ~~A = -7~~ A = 0 (n+2) bits

Q M = -7 ⇓ 5 cycles

Q = 3

A = 00000

~~M = (-7)₁₀ = 0~~ $-M = (7)_{10} = (00111)_2$

$Q = (3)_{10} = (00011)_2$

$M\ (-7)_{10} = 11001$

00111

11000

+1

11001

$$\begin{array}{r} 00001 \\ +\ 11001 \\ \hline 11010 \end{array} \qquad \begin{array}{r} 00000 \\ +\ 00011 \end{array}$$

Cycle	A	Q	Q-1	Operation
1st	00000	00011	0	Q.Q-1 = 10 A = A - M
	00111	00011	0	A.S.R
	00011	10001	1	
2nd	00001	11000	1	A.S.R
3rd	11010	11000	1	A = A + M A.S.R.
	11101	01100	0	A.S.R
4th	11110	10110	0	A.S.R
5th	11111	01011	0	
	00000	10100		

+1

$\boxed{00000 \quad 10101}$

00000,01010

$$\begin{array}{r} 00000 \\ +00101 \\ \hline \end{array}$$

$$\begin{array}{r} 00010 \\ +11011 \\ \hline 11101 \end{array}$$

$$\begin{array}{r} 00101 \\ 11010 \\ +1 \\ \hline 11011 \end{array}$$

Q. Find the product of -5×2 using booth's algorithm

⇒ $-5 \times 2 = -10$

$A = 0 = 00000$

$M = (-5)_{10} = 11011$

$-M\ (5)_{10} = 00101$

$Q = (2)_{10} = 00010$

Reg = Cycle = 5

Cycle	A	Q	Q_{-1}	Operation
1st	00000	00010	0	ASR
	00000	00001	0	
				A = A - M
2nd	00101	00001	0	A.S.R
	00010	10000	1	
3rd				A = A + M
	11101	10000	1	A.S.R
	~~11110~~	11000	0	
4th	11111	01100	0	ASR
5th	11111	10110	0	ASR
	00000	01001		
		+1		
	00000	01010		= 10

Q. Find product of $4 \times (-3)$ using booths Algorithm.

$4 \times 3 = 12$

→ $A = 00000$

$M = (4)_{10} = 00100$

$-Q = (3)_{10} = 00011$

$Q = (-3)_{10} = 11101$

$-M = (4)_{10} = 11100$

Cycle = register = 5

Cycle	A	Q	Q-1	O/P
1st	00000	11101	0	A = A-M
	11100	11101	0	A.SR
	11110	01110	1	
2nd	00010	01110	1	A = A+M
	000001	00111	0	A.SR
3rd	11101	00111	0	A = A-M
	11110	10011	1	A.S.R
4th	11111	01001	1	A.S.R
5th	11111	10100	1	A.S.R
5th comp	00000	01011		
		+1		

$\boxed{0000 \quad 01100}$

10 ma

* IEEE Std 754 :: ~~Floating~~ point.

→ ~~Sign~~ →
→ ~~Exponent~~ →
→ ~~Mantissa~~ →

Floating point		Single Precision	Double precision
→ Sign	→	1 bit	1 bit
→ exponent	→	8 bit	11 bit
→ Mantissa	→	23 bit	52 bit
		32 bit	64 bit

S.P | sign | Exp | Mantissa |
1 bit, 8 bit, 23 bit

D.P | sign | exp | Mantissa |
1 bit, 11 bit, 52 bit

If Number is +ve → '0' :- sign
If No is -ve → '1' :- sign

Ex Conver $(85.125)_{10}$ using IEEE format into single and double precision format.

→ ① Convert into Binary

~~85~~ 85 = 1010101

2	85
2	42 → 1
2	21 → 0
2	10 → 1
2	5 → 0
2	2 → 1
	1 → 0

(~~0~~ ~~0.125 × 2 =~~

0.125 × 2 = 0.250 → 0
0.250 × 2 = 0.5 → 0
0.5 × 2 = 1 → 1

$(85.125)_{10} = (1010101.001)_2$

$= 1.010101001 \times 2^6$

$\boxed{\text{Single Precision (32 bits)} \quad (1.N) \times 2^{E-127}}$

$E - 127 = 6$

$E = 133$

$(133)_{10} = (10000101)_2$

Mantissa :- $\underbrace{010101001}_{9 \text{ bits}}$ 00000000000000

Sign = 0

Exp = 10000101

Mantissa = 01010100100000000000000

$\boxed{\text{Double Precision} \quad (1.N) \times 2^{E-1023}}$

$\therefore E - 1023 = 6$

$E = (1029)_{10}$

$E = (1029)_{10} = 10000000101$

Mantissa: 010101001 00000000 00000000 0000000 00000 00000000 00000 00000

ex2:

Ex2: For 132.65 obtain IEEE 754 std of single and double precision format

→ $132 = 10000100$

$0.65 \times 2 = 1.3 \rightarrow 1$

$0.3 \times 2 = 0.6 \rightarrow 0$

$0.6 \times 2 = 1.2 \rightarrow 1$

$0.2 \times 2 = 0.4 \rightarrow 0$

$0.4 \times 2 = 0.8 \rightarrow 0$

$(132.65)_{10} = 10000100.101001001001$

$= 1.00001001010\ 11001001 \times 2^{7}$

$S.P = 1.N \times 2^{E-127}$

$E - 127 = 7$

$\therefore E = (134)_{10}$

$= (10000110)_2$

$\therefore M = \underbrace{0000100101001100100110}_{23 \text{ bits}}$

$D.P = 1.N \times 2^{E-1023}$

$E - 1023 = 7$

$\therefore E = 1030$

$\therefore E = 10000000110$

Q x_0 $(263.3)_{10}$

$(263)_{10} = 100000111$

$0.3 \times 2 = 0.6 = 0$
$0.6 \times 2 = 1.2 \rightarrow 1$
$0.2 \times 2 = 0.4 \rightarrow 0$
$0.4 \times 2 = 0.8 \rightarrow 0$
$0.8 \times 2 = 1.6 \rightarrow 1$
$0.6 \times 2 = 1.2 \rightarrow 1$
$0.2 \times 2 = 0.4 \rightarrow 0$
$0.4 \times 2 = 0.8 \rightarrow 0$
$0.8 \times 2 = 1.6 \rightarrow 1$
$0.6 \times 2 = 1.2 \rightarrow 1$
$0.2 \times 2 = 0.4 \rightarrow 0$

$(263.3) = 100000111.01001100110011001100$
$= 1.00000111010011001100 \times 2^8$

$\therefore E = 127 + 8$
$= (135)_{10}$
$= 10000111$

$$M = \underbrace{00000111010011001100110}_{23 \text{ bit}}$$

D.P = E = 1023 + 8

= 1031

= 10000000111

1120