

Amey Thakur

B - 50

Mumbai University

Terna Engineering College, Nerul, Navi Mumbai

Department of Computer Engineering

Course Code	CSC403	Program	B.E. (CMPN)
Semester	IV	Year	II
Name of the Faculty	Rohini Palve	Class / Div	A & B
Course Title	Computer Organization and Architecture	Academic year	2019-20

Roll No.: 50	Name: Amey Thakur
Class: SE COMPS, Div: B	Batch: B3
Date of Assignment: 3/3/2020	Date of Submission: 08/02/2020

## ASSIGNMENT 1

Q1) Differentiate between computer organization and architecture with suitable examples (CO1)

Q2) Explain PC, MAR, MDR, IR and SP (CO1)

Q3a) Explain the Von Neumann Architecture and explain the Stored Program Concept.

Q3b) Compare Von Neumann Architecture and Harvard Architecture (CO1)

Q4) Explain the instruction cycle state diagram. (CO1)

Q5) Explain the biasing of exponent and normalization of mantissa in floating-point number. (CO2)

Q6) Explain the IEEE 754 floating-point format with two examples. (CO2)

Q7) Draw flowchart for unsigned binary multiplication and solve one numerical. (CO2)

Q8) Draw flowchart for Booth's algorithm for binary multiplication and solve two Numerical. (CO2)

Q9) Draw flowchart for Restoring the division algorithm and solve two Numerical (CO2)

Q10) Draw a flowchart for Non-Restoring division algorithm and solve two Numerical (CO2)

Q11) Draw a flowchart for Floating-point addition, subtraction, multiplication and division (CO2)

## Assignment No. 1

06-01-20 Q1. Computer architecture and computer organization

Ans:

Computer  
Architecture

Computer  
Organization

① It refers to those attributes of a system visible to the programmer.

② Instruction set, number of bits used for data representation, addressing techniques etc. form the part of computer architecture.

③ For example, is there a multiply instruction?

④ All INTEL 80x86 microprocessors share the same basic architecture

① It refers to the implementation of these features and is mostly not known to the user.

② Control signals, interfaces, memory technology etc form the part of computer organization.

③ For example, is there a dedicated hardware multiply unit or it is done by repeated addition?

④ All INTEL 80x86 microprocessors differ in their organization.

Q2. Von neumann architecture and stored program

Ans:

- ① Stored program concept, storage of instructions in computer memory to enable it to perform a variety of tasks in sequence or "intermittently".
- ② Stored program concept is designed by Hungarian mathematician John Von Neumann.
- ③ The von Neumann architecture is a design model for a stored program digital computer that uses a processing unit and a single separate storage structure to hold both instructions and data.
- ④ A stored program digital computer is one that keeps its programmed instructions as well as its data in read-write, random access memory (RAM).
- ⑤ Before any data are processed, instructions are read into memory. The processing starts with the first instruction in the program, which is copied into a control unit circuit. The control unit executes the instructions sequentially until it finds one that causes it to break the sequence and go elsewhere in the program.
- ⑥ Input/output and processing are performed simultaneously.

Q.3. Explain MAR, MBR, IR, SR, PC.

Ans:

① PC - Program Counter

Contains the address of the next instruction pair to be fetched from memory.

② MAR - Memory Address Register

Specifies the address in the memory of the word to be written from or read into MBR.

③ MBR - Memory Buffer Register

Contains a word to be stored in memory or sent to the I/O unit or is used to receive a word from memory or from the I/O unit.

④ IR - Instruction Register

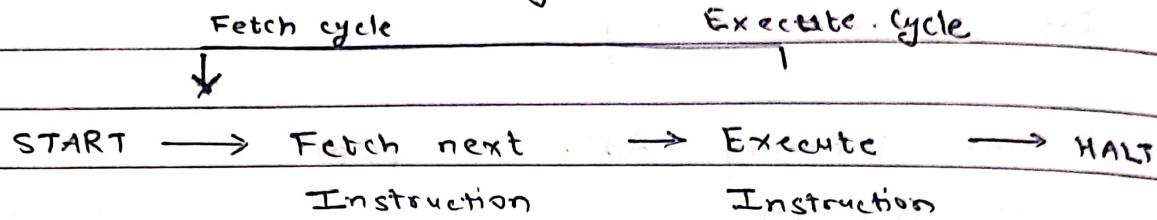
Contains the 8-bit opcode instruction being executed.

⑤ Stack pointer / stack register. -

If there is a user visible stack addressing, then typically there is a dedicated register that points to the top of the stack. This allows implicit addressing; that is push, pop and stack operations which need not contain an explicit stack operand.

Q.4 Instruction CycleAns:

The process required for a single instruction is called an instruction cycle.

Basic Instruction Cycle.

In its simplest form instruction processing consists of two steps. Two steps are referred to as the fetch cycle and the execute cycle.

$$\underline{Q.5.} \quad M = 11 \quad \text{multiplicand} = 1011$$

$$Q = 13 \quad \text{multiplier} = 1101$$

Ans:

C	A	Q	M	Initial Values
0	0000	1101	1011	

0	1011	1101	1011	Add } First cycle
0	0101	1110	1011	shift }

0	0010	1111	1011	shift } Second cycle
---	------	------	------	----------------------

0	1101	1111	1011	Add } Third cycle
0	0110	1111	1011	shift }

1	1111	1111	1011	Add } Forth cycle
0	1111	1111	1011	shift }

$$(1000 \ 1111)_2 = (143)_{10}$$

Q5. Multiply  $-7 \times 3$  using Booth's algorithm. Register Bit 5

Ans:

$$m = Q\text{'s complement of } 7 = (11001)_2$$

$$-m = (00111)_2$$

$$Q = (00011)_2$$

AC	Q	Q-1	n (no. of bits)
00000	00011	0	5
		1	
			→ AC = AC - m
00111	00011	0	4
00011	10001	1	Q Q-1
		1	
			→ Arithmetic Right Shift
00001	11000	1	3
		1	
			→ AC = AC + m
11010	11000	1	2
11101	01100	0	
		1	
			→ Arithmetic Right Shift
11110	10110	0	1
		1	
			→ Arithmetic Right Shift
11111	01011	0	0

Now we get 11111 01011 → This answer we have to convert into 2's complement form, to get the correct answer.

$$\begin{array}{r} 11111 \\ 00000 \\ \hline +1 \\ \hline 00000 \end{array}$$

$$00000 \quad 10101 \Rightarrow (00000 \quad 10101)_2$$

Q.7. Using booth's algorithm, multiply 7 and 5

Ans:

$$m = 2\text{'s complement of } 7 = (1001)_2$$

$$-m = (0111)_2$$

$$Q = (0101)_2$$

AC      Q      Q-1      n (no. of bits)

$$\begin{array}{cccc} 0000 & 0101 & 0 & 4 \\ & \overbrace{\quad\quad\quad}^1 & \overbrace{\quad\quad\quad}^0 & \\ & \rightarrow 10 \Rightarrow AC = AC - m & & \end{array}$$

$$\begin{array}{cccc} 1001 & 0101 & 0 & 3 \\ 1100 & 1010 & 1 & \\ & \overbrace{\quad\quad\quad}^1 & \overbrace{\quad\quad\quad}^1 & \\ & \rightarrow 01 \Rightarrow AC = AC + m & & \end{array}$$

$$\begin{array}{cccc} 0011 & 1010 & 1 & 2 \\ 0001 & 1101 & 0 & \\ & \overbrace{\quad\quad\quad}^1 & \overbrace{\quad\quad\quad}^1 & \\ & \rightarrow 10 \Rightarrow AC = AC - m & & \end{array}$$

$$\begin{array}{cccc} 1010 & 1101 & 0 & 1 \\ 1101 & 0110 & 1 & \\ & \overbrace{\quad\quad\quad}^1 & \overbrace{\quad\quad\quad}^1 & \\ & \rightarrow 01 \Rightarrow AC = AC + m & & \end{array}$$

$$\begin{array}{cccc} 0100 & 0110 & 1 & 0 \\ 0010 & 0011 & 0 & \end{array}$$

$$\therefore (0010 \ 0011)_2 = (35)_{10}$$

Amey 13 50

PAGE No.	
DATE	/ /

Q8. Using Booth's algorithm, multiply -3 and -7

Ans:

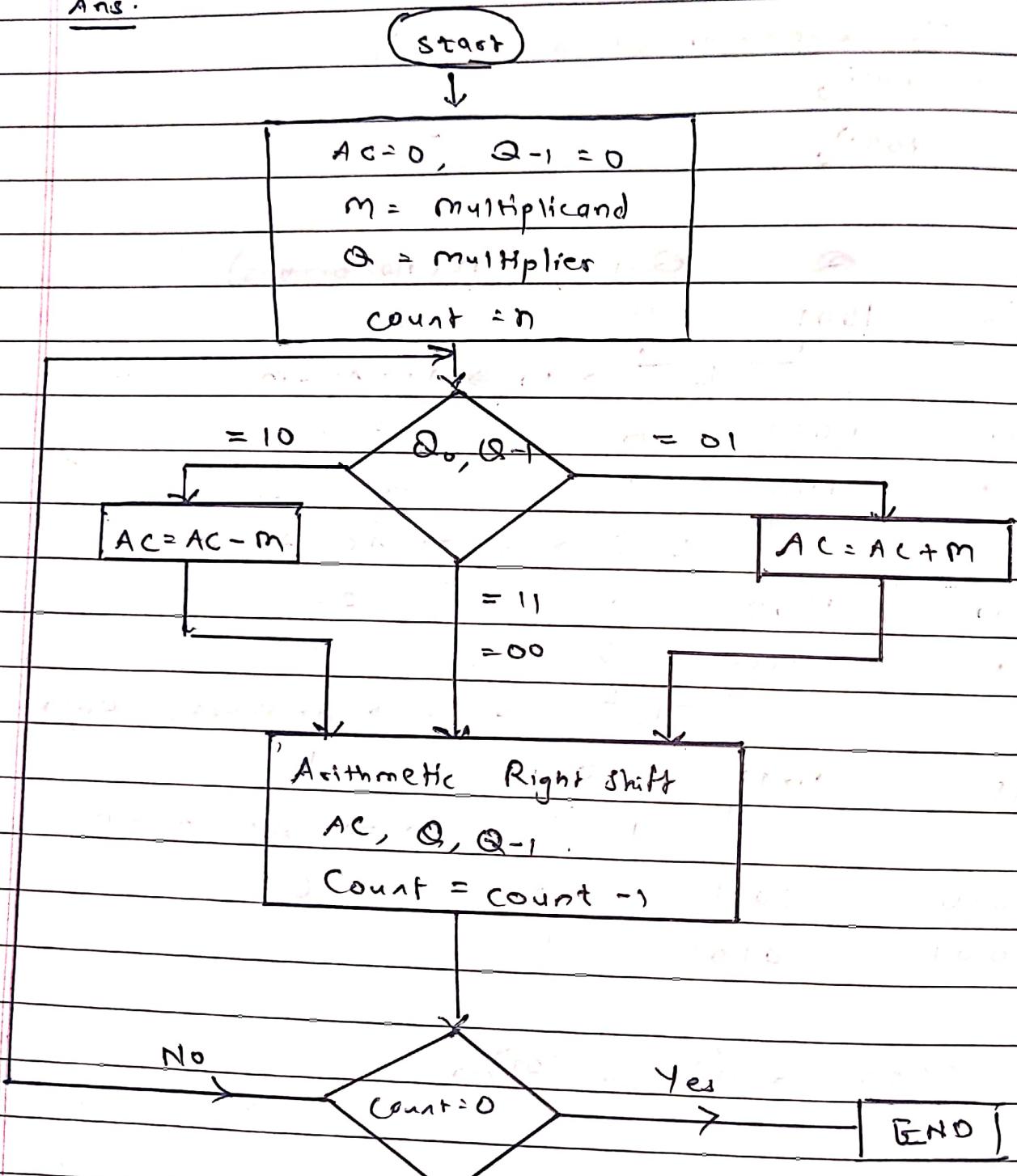
$$\begin{aligned}m &= 2\text{'s complement of } 3 = (1101)_2 \\-m &= (0011)_2 \\Q &= (1001)_2\end{aligned}$$

AC	Q	Q-1	n (no. of bits)
0000	1001	0	4
		1 → 10	⇒ AC = AC - m
0011	1001	0	3
0001	1100	1	
		1 → 01	⇒ AC = AC + m
1110	1100	1	2
1111	0110	0	
		1 → 00	⇒ Arithmetic Right Shift.
1111	1011	0	1
		1 → 10	⇒ AC = AC - m
0010	1011	0	0
0001	0101	1	

$$\therefore (0001 \ 0101) = (21)_{10}$$

Q.9. Flowchart of Booth's Algorithm.

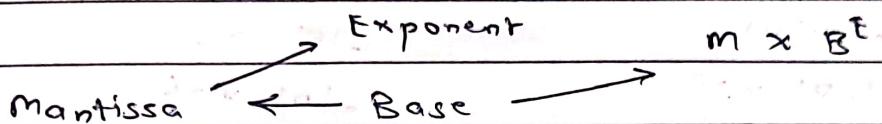
Ans:



Q10. Explain normalization of mantissa in floating point representation and biasing of exponent.

Ans:

- The mantissa  $M$  is referred as the significant or fraction. These three components - ① Mantissa ( $M$ ), ② An exponent ( $E$ ), ③ Base ( $B$ ) represents the real number.



For example, in  $2.5 \times 10^{10}$

$$\text{Mantissa} = 2.5$$

$$\text{Exponent} = 10$$

$$\text{Base} = 10$$

0	1	8	9	31
sign   Biased exponent = 8 bit   Significant = 23 bits				

In floating point numbers representation

- ① Exponent is biased
- ② Mantissa or significant is normalized

### 1] Biased Exponent

Exponent is represented using an excess -128 code.

An 8 bit binary number represents value from 0 to 255.

However, we are adding 128 in the biased exponent.

The actual exponent value represented will be -128 to 127.

Exponent Value	Exponent value after biasing (+128)
----------------	-------------------------------------

-128	0 (000 0000) <sub>2</sub>
------	---------------------------

-127	1 (0000 0001) <sub>2</sub>
------	----------------------------

0	128 (1000 0000) <sub>2</sub>
---	------------------------------

127	255 (1111 1111) <sub>2</sub>
-----	------------------------------

## ii) Normalized mantissa.

A binary floating point number is represented in a normalized form, i.e. The number is of the form  $\pm 0.$  (Significant starting with non-zero bit)  $\times 2^{\pm}$  (exponent value).

Binary Number	Normal Form
11.101	$-11101 \times 2^2$
.00101	$-101 \times 2^{-2}$
$-1.01 \times 2^5$	$-101 \times 2^6$

Q11. Convert  $(-307.1875)$  in single and double precision format

Ans:

## i) Single precision

$$(307)_{10} = (10011001)_2$$

$$.1875 \times 2 = 0.3750 = 0$$

$$.3750 \times 2 = 0.7500 = 0$$

$$.7500 \times 2 = 1.500 = 1$$

$$.500 \times 2 = 1.000 = 1$$

$$\therefore (307.1875)_{10} = (100110011.0011)_2$$

$$100110011.0011 = \underbrace{100110011.0011}_{\text{Significand}} \times 2^8$$

$$E \Rightarrow -127 = 8$$

$$E = 135 = 1.001100110011 \times 2^{135-127}$$

$$(135)_{10} = (10000111)_2$$

0	1	8	9	31
1	0	000111	001100110011...~0000	

## 11 Double precision Format

$$\begin{aligned}100110011.0011 &= 1.00110011 \cdot 0011 \times 2^8 \\&= 1.001100110011 \times 2^{1023} = 1023\end{aligned}$$

$$E - 1023 = 8$$

$$E = 1031$$

$$(1031)_{10} \rightarrow (10000000111)_2$$

0	1	11	12	
63				
				00110011001100...0000

= . | | 1 00000000111 ] 00110011001100...0000

Q12. Explain IEEE 754 floating point representation form

Ans:

- ① IEEE floating point standards addresses a number of such problems
- ② zero has definite representation in IEEE format.
- ③  $\pm \infty$  has been represented in IEEE format.
- ④ A + 00 indicated that the result of an arithmetic expression is too large to be stored.
- ⑤ If an underflow occurs, implying that a result is too small to be represented as a normalized number. It is encoded in denormalized scale.

⑤ Representation

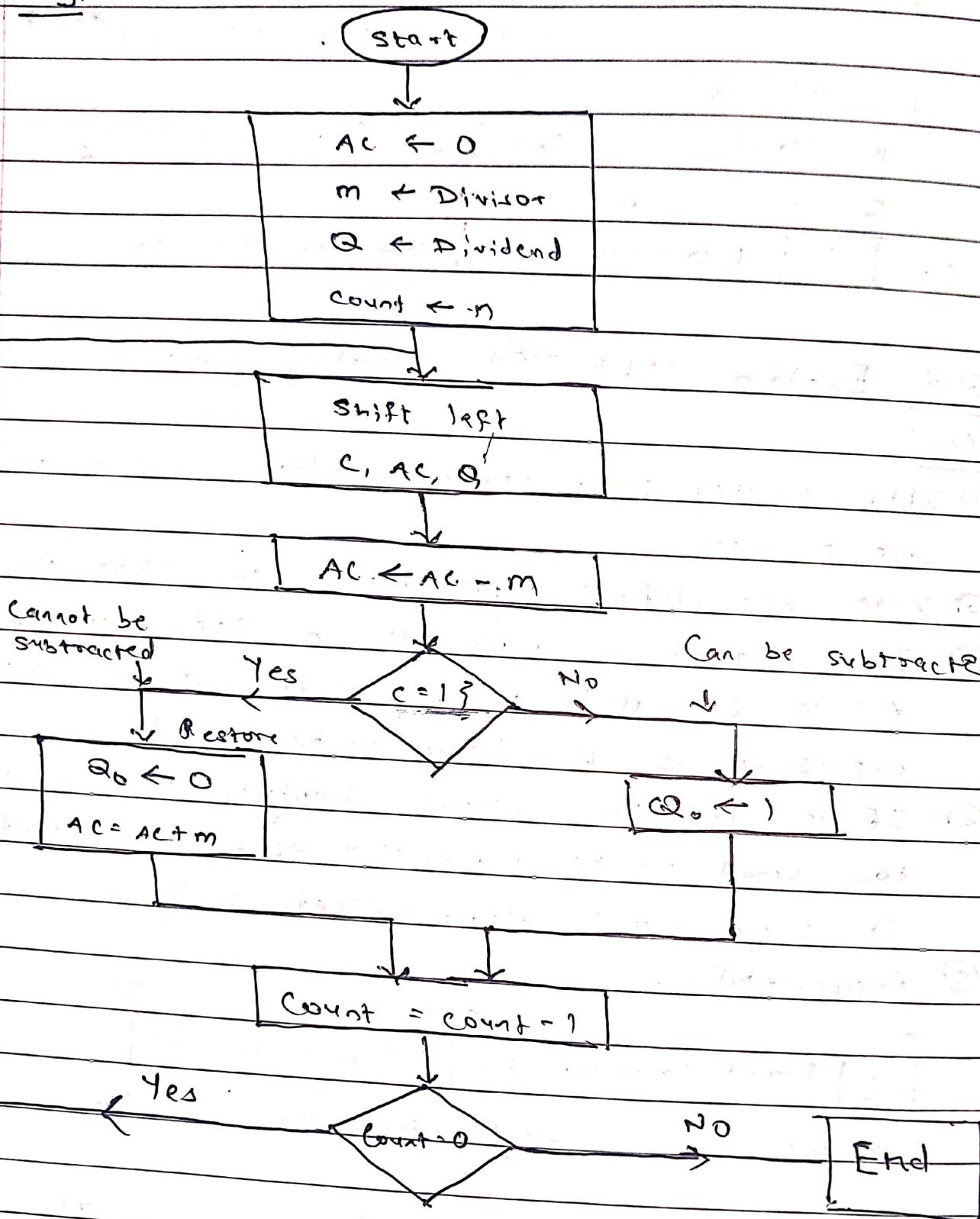
0	1	8 9	31
S   Biased Exponent (E)		Significant (N)	
sign bit		single precision = 32 bits.	

0	1	11 12	63
S   Biased Exponent (E)		Significant (N)	
sign bit		Double precision = 64 bits	

- base = 2
- Significant is in normalized form
- S is sign bit.

Q18. Flowchart for unsigned binary division. (Restoring Division)

Ans:



Q14.  $Q = 1100$

$M = 00011 \rightarrow (n+1) \text{ bits}$

$-M = \underbrace{11101}_{(n+1) \text{ bit}}$  One additional bit to handle carry/borrow sign

Sol<sup>n</sup>

a

C

AC

Q.

D

0000

1100

$$\begin{array}{l} \text{I}^{\text{st}} \\ \text{cycle} \end{array} \left\{ \begin{array}{lll} 0 & 0001 & 1001 \\ 1 & 1110 & 10010 \end{array} \right.$$

$$\begin{array}{l} \text{II}^{\text{nd}} \\ \text{cycle} \end{array} \left\{ \begin{array}{lll} 1 & 1101 & 0000 \\ 0 & 0000 & 00001 \end{array} \right.$$

$$\begin{array}{l} \text{III}^{\text{rd}} \\ \text{cycle} \end{array} \left\{ \begin{array}{lll} 0 & 0000 & 0011 \\ 1 & 1101 & 00110 \end{array} \right.$$

$$\begin{array}{l} \text{IV}^{\text{th}} \\ \text{cycle} \end{array} \left\{ \begin{array}{lll} 1 & 1010 & 0101 \\ 1 & 1101 & 0100 \end{array} \right. \underbrace{\quad}_{\text{Quotient}}$$

$\therefore$  At the end of  $4(n)$  cycles, AC is negative.

$\therefore$  We perform operation

$$AC \leftarrow AC + M$$

$$\therefore AC = 11101$$

$$M = 00011$$

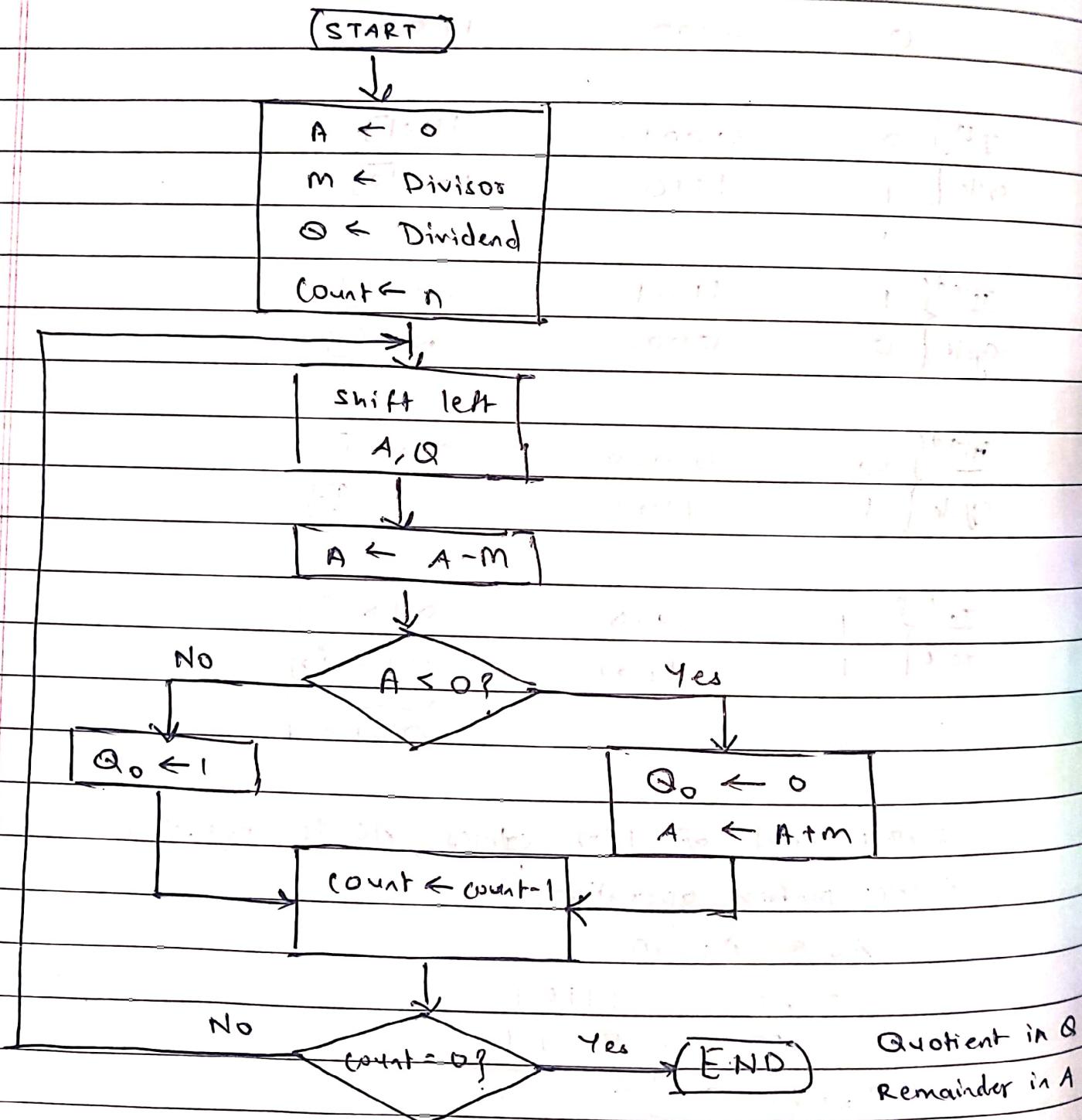
$$\text{Discard} \rightarrow 1] 00000$$

$$\therefore \text{Remainder} = 0000$$

$$\text{Quotient} = 0100 = \underline{\underline{4}}$$

~~Q15.  $\begin{array}{r} 100001 \\ \times 101 \\ \hline 100001 \\ -10000 \\ \hline 101001 \\ -10000 \\ \hline 101 \end{array}$~~

### Q15. Restoring Method



Quotient in  $q$   
Remainder in  $A$

Amey

B

150

H

Page No.	
Date	/ /

Q16. 13/5 ← Divide

Soln:

A	Q	M
0 0 0 0 0	0 1 1 0 1	0 0 1 0 1

Initial values

$$\begin{array}{r}
 0 0 0 0 0 \\
 + 1 1 0 1 1 \\
 \hline
 1 1 0 1 1 \\
 + 0 0 1 0 1 \\
 \hline
 0 0 0 0 0
 \end{array}$$

First cycle

$$\begin{array}{r}
 0 0 0 0 1 \\
 + 1 1 0 1 1 \\
 \hline
 1 1 1 0 0 \\
 + 0 0 1 0 1 \\
 \hline
 0 0 0 0 1
 \end{array}$$

Second cycle.

$$\begin{array}{r}
 0 0 0 1 1 \\
 + 1 1 0 1 1 \\
 \hline
 1 1 1 1 0 \\
 + 0 0 1 0 1 \\
 \hline
 0 0 0 1 1
 \end{array}$$

Third cycle

$$\begin{array}{r}
 0 0 1 1 0 \\
 + 1 1 0 1 1 \\
 \hline
 0 0 0 1 1
 \end{array}$$

Fourth cycle

$$\begin{array}{r}
 0 0 0 1 1 \\
 + 1 1 0 1 1 \\
 \hline
 1 1 1 1 0 \\
 + 0 0 1 0 1 \\
 \hline
 0 0 0 1 1
 \end{array}$$

Fifth cycle

Quotient =  $(00010)_2 = (2)_{10}$   
 Remainder =  $(00011)_2 = (3)_{10}$

Q.17  $13/3 \leftarrow$  Divide

Sol:

A	Q	M	
00000	01101	00011	Initial values

$$\begin{array}{r}
 00000 \quad 1101 \quad 00011 \\
 + 11101 \\
 \hline
 11101 \quad 11010 \quad 00011
 \end{array}
 \quad \left. \begin{array}{l} \\ \\ \end{array} \right\} \text{First cycle.}$$

$$\begin{array}{r}
 00001 \quad 1010 \quad 00011 \\
 + 11101 \\
 \hline
 11110 \quad 10100 \quad 00011
 \end{array}
 \quad \left. \begin{array}{l} \\ \\ \end{array} \right\} \text{Second cycle}$$

$$\begin{array}{r}
 00011 \quad 0100 \quad 00011 \\
 + 11101 \\
 \hline
 00000 \quad 01001 \quad 00011
 \end{array}
 \quad \left. \begin{array}{l} \\ \\ \end{array} \right\} \text{Third cycle}$$

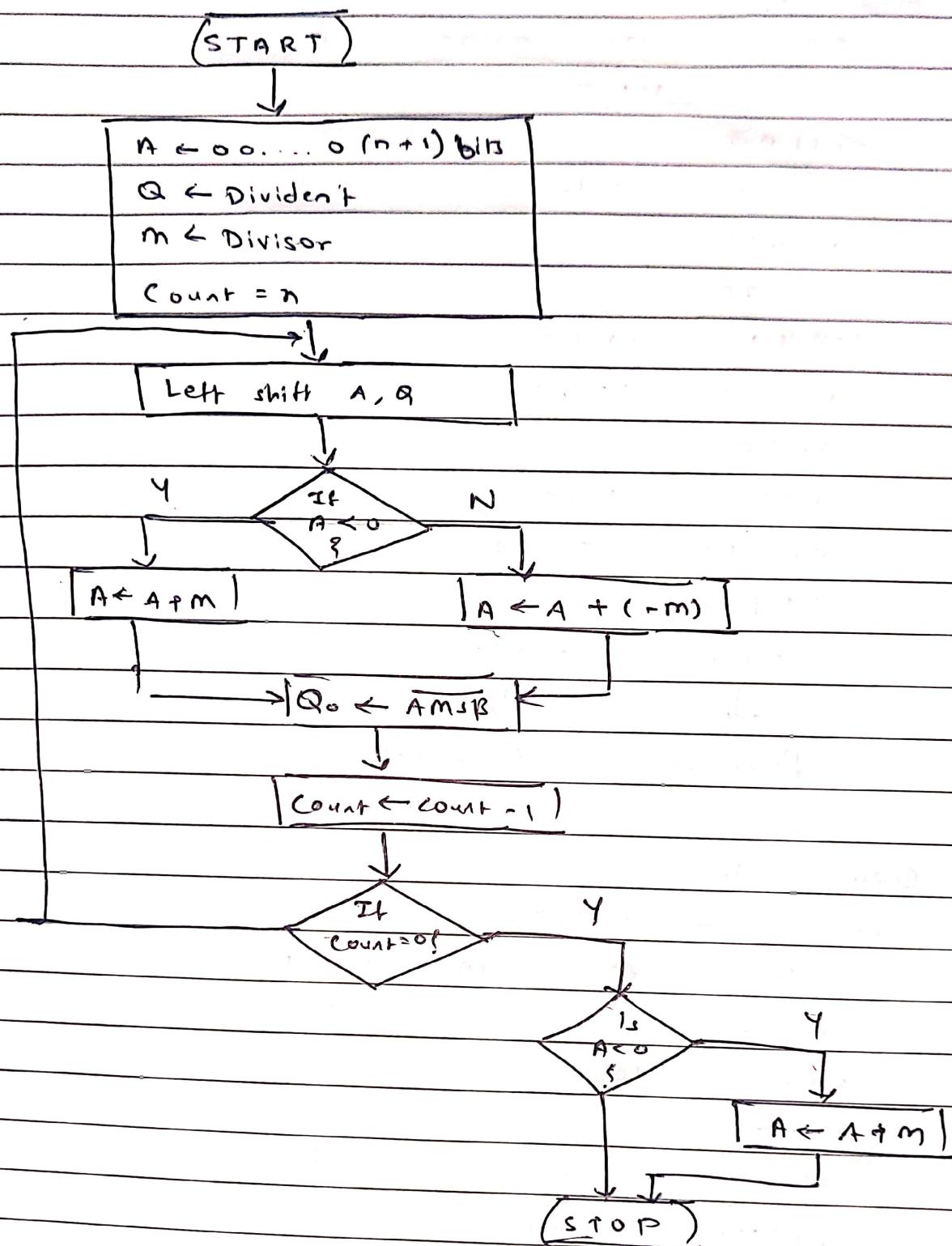
$$\begin{array}{r}
 00000 \quad 1001 \quad 00011 \\
 + 11101 \\
 \hline
 11101 \quad 10010 \quad 00011
 \end{array}
 \quad \left. \begin{array}{l} \\ \\ \end{array} \right\} \text{Fourth cycle}$$

$$\begin{array}{r}
 00001 \quad 0010 \quad 00011 \\
 + 11101 \\
 \hline
 11110 \quad 00100 \quad 00011
 \end{array}
 \quad \left. \begin{array}{l} \\ \\ \end{array} \right\} \text{Fifth cycle}$$

$$\text{Quotient} = (00100)_2 = (4)_{10}$$

$$\text{Remainder} = (00001)_2 = (1)_{10}$$

### G.18. Non-Restoring Method



Q19. 15/4

Sol's	A	Q	m	count
	00000	111	00100	4
	00001	111	0	
	11100			
	11101	1110		3
	11011	1100		
	00100			
	11111	1100		2
	11111	1101		
	00100			
	00011	1001		1
	00111	0011		
	11100			
	00011	0011		
		↓		
			↓	
		Remainder	Quotient	

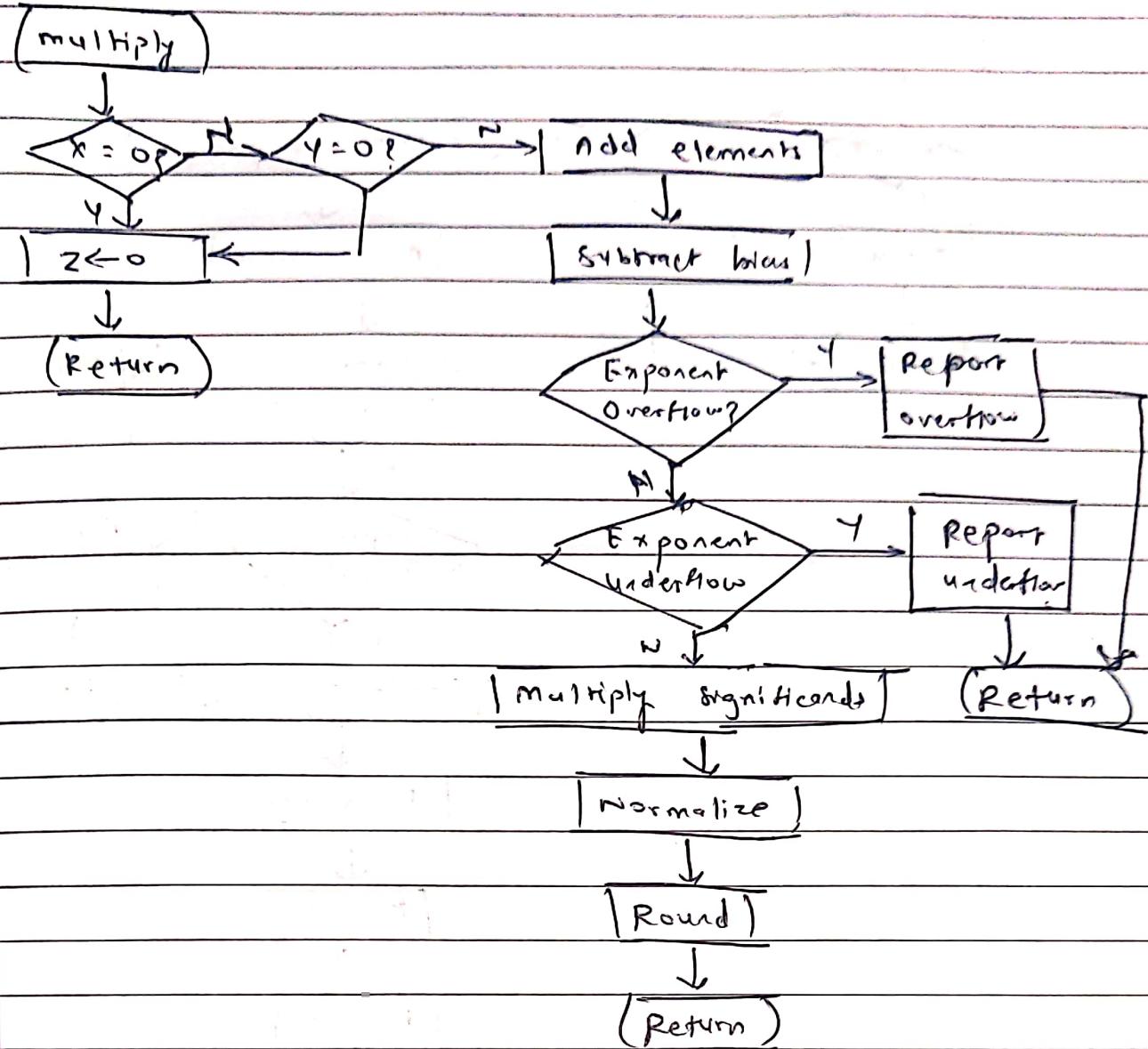
Q20. 12/3

$$\text{Sol}^{\wedge}: -m = 11101$$

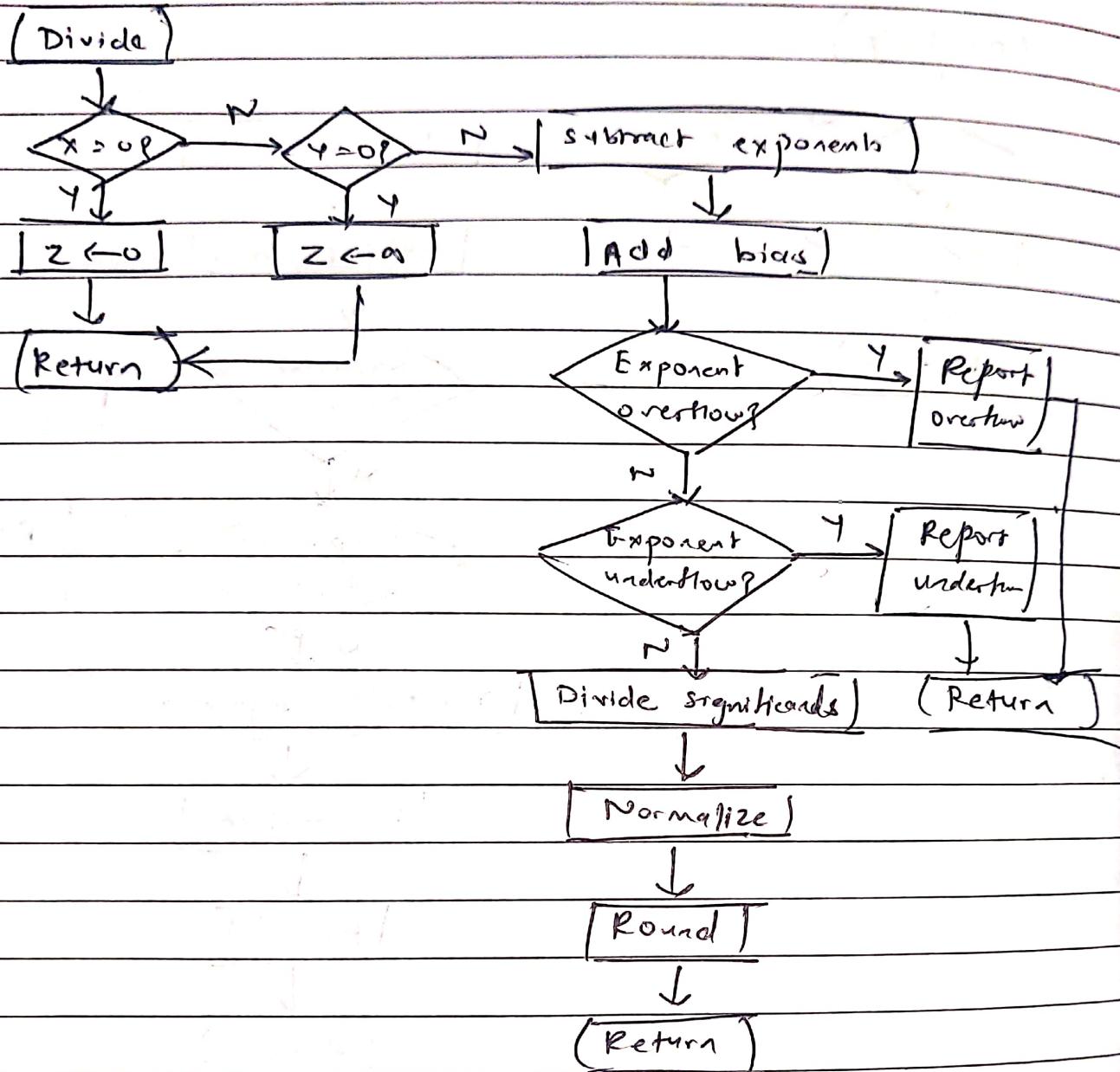
Sol's	A	Q	m	count
	00000	1100	00011	4
	00001	1000		
	11101			
	11110	1000		3
	11101	0001		
	00011			
	00000	0001		2
	00000	0010		
	11101			
	11101	0010		1
	11010	0100		
	00011			
	11101	0100		
	00011			0

Remainder = 00000  
 Quotient = 0100

Q.1 Floating point - multiplication



## Floating point division



~~Addition & Subtraction~~ Addition & subtraction

