-: Computer Organization & Architecture Assignment: - 05

Short Answer Questions: —

1) Discuss in short about Signed 1's Complement and 2's complement representation

Ans→

1's Complement	2's Complement
• To get 1's Complement of a binary number, Simply invert the given number.	• To get 2's complement of a binary number, Simply invert the given number and add 1 to the least significant bit (LSB) of given result.
• 1's complement of binary number 110010 is 001101.	• 2's complement of binary number 110010 is 001110.
• Simple implementation which gives uses only NOT gates for each input bit.	• Uses NOT gate along with full adder for each input bit.
• Can be used for signed binary number representation but not suitable as ambiguous representation for number 0.	• Can be used for signed binary number representation and most suitable as unambiguous representation for all numbers.
Examples: - Let numbers be stored using 4 bits. 1's Complement of 7 (0111) is 8 (1000)	Examples: - 2's complement of 7 (0111) is 9 (1001).

2) Write short on floating point representation of Decimal number

-: PTO :-

-: Computer Organization & Architecture Assignment :- 03

Short Answer Questions :-

1) Discuss in short about Signed 1's Complement and 2's complement representation.

Ans->

1's Complement	2's Complement
• To get 1's Complement of a binary number, Simply invert the given number.	• To get 2's complement of a binary number, Simply invert the given number and add 1 to the least significant bit (LSB) of given result.
• 1's complement of binary number 110010 is 001101.	• 2's complement of binary number 110010 is 001110.
• Simple implementation which gives uses only NOT gates for each input bit.	• Uses NOT gate along with full adder for each input bits.
• Can be used for signed binary number representation but not suitable as ambiguous representation for number 0.	• Can be used for signed binary number representation and most suitable as unambiguous representation for all numbers.
Examples:- Let numbers be stored using 4 bits. 1's Complement of 7 (0111) is 8 (1000)	Examples:- 2's complement of 7 (0111) is 9 (1001).

2) Write Short on floating point representation of Decimal number

-: PTO :-

Ans-> The floating-point Representation does not reserve a specific number of bits for the integer part or the fractional part. It reserves a certain number of bits for the number (called the mantissa or significand) and a certain number of bits to say where within that number the decimal place sits called the exponent.

Floating-point is always interpreted to represent a number in the following form : $M \times r^e$.

A floating-point number is said to be normalized if the most significant digit of the mantissa is 1.

n		0
Sign bit	Exponent	Mantissa

Biased form

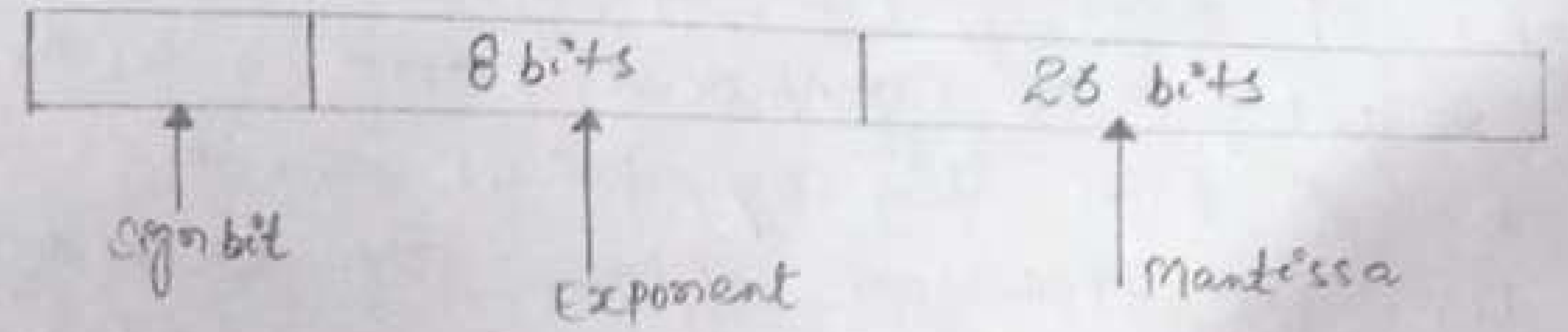
-: Long Answer Questions :-

3) Explain IEEE Floating Point representation in single precision and double precision formats with an example.

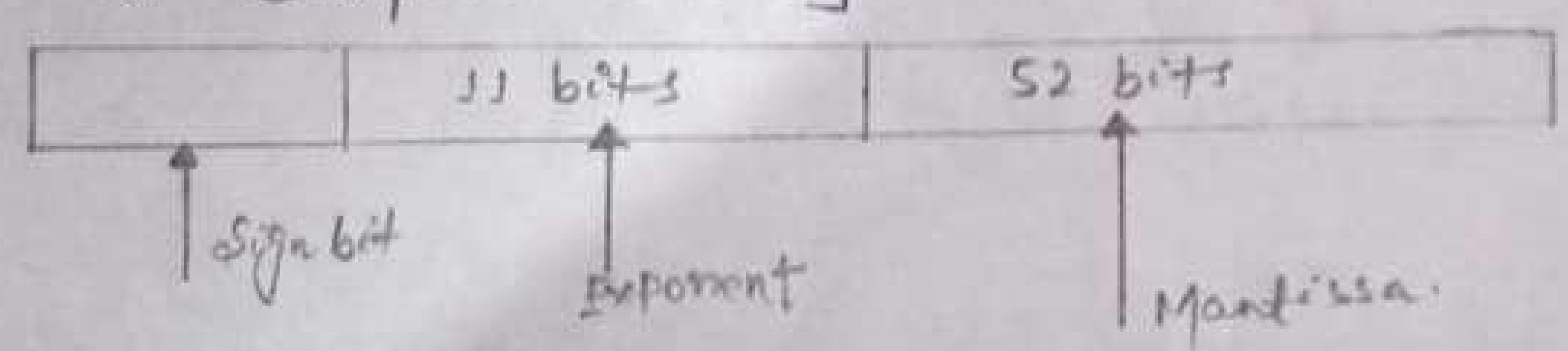
Ans-> According to IEEE standard, floating point number is represented into two ways:-

Precision	Base	Sign	Exponent	Significand
Single Precision	2	1	8	23 + 1
Double Precision	2	1	11	52 + 1

1) Single Precision:- Single Precision is format proposed by IEEE for representation of floating-point numbers. It occupies 32 bits in computer memory.



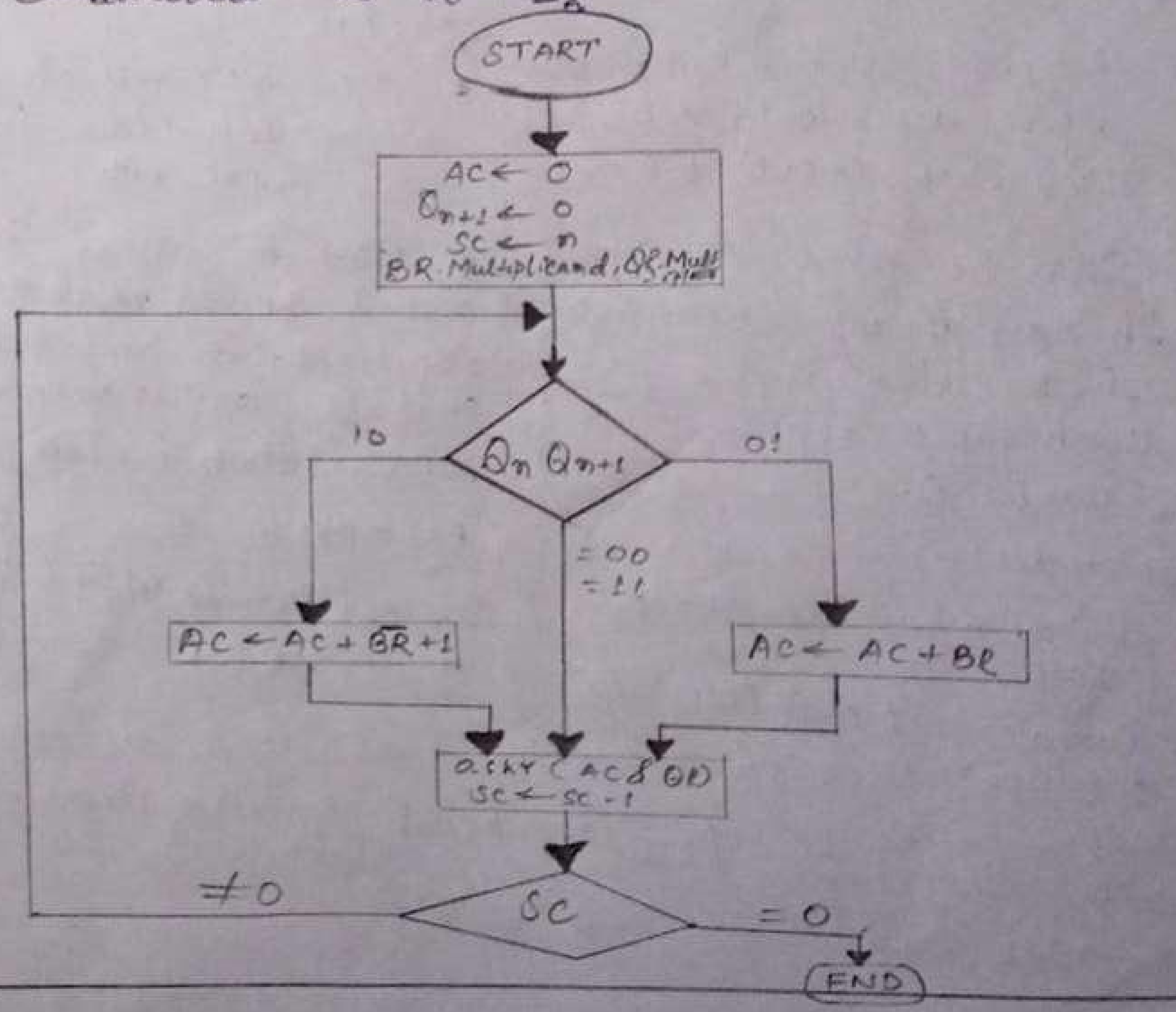
2) Double Precision:- Double precision is also a format given by IEEE for representation of floating-point number. It occupies 64 bits in computer memory.



Single Precision	Double Precision
• In Single precision, 32 bits are used to represent floating-point number.	• In double precision, 64 bits are used to represent floating-point number.
• It uses 8 bits for exponent	• It uses 11 bits for exponent
• In single precision, 23 bits are used for mantissa	• In double precision, 52 bits are used for mantissa.
• Bias number is 127.	• Bias number is 1023.
• It is used in simple program like games	• It is used in complex program like scientific calculator.
• This is called binary 32.	• This is called binary 64.

4) Explain Booth's Multiplication Algorithm with an Example using necessary diagrams.

Ans => The Booth algorithm is a multiplication algorithm that allows us to multiply the two signed binary integer in 2's complement, respectively. It is also used to speed up the performance of the multiplication process. It works on the string bits 0's in the multiplier that requires no additional bit only shift the right-most string bits and a string of 1's in a multiplier bit weight 2^k to weight 2^m that can be considered as $2^{k+1} - 2^m$



In the above chart, initially, AC and Q_{n+1} bits are set to 0. and the SC is a sequence counter that represents the total bits set n, which is equal to the number of bits in the multiplier.
There are BR that represent the multiplicand bits, and QR represents the multiplier bits. After that, we encountered two bits of the multiplier as Q_n and Q_{n+1}, where Q_n represents the last bit of QR, and Q_{n+1} represent the incremented bit of Q_n by 1. The arithematic shift operation is used in Booth's algorithm to shift AC and QR bits to the right by one and remains the sign bit in AC unchanged. And the sequence counter is continuously decremented till the computational loop is repeated, equal to the number of bits(n).

Example:- Multiply the two numbers 23 and -9 by using the Booth's multiplication algorithm.

$Q_n.Q_{n+1}$	M = 010111 M+1 = 101001	AC	Q	Q_{n+1}	SC
	Initially	000000	110111	0	6
1 0	Subtract M	101001			
		101001			
	Perform Arithematic right Shift operation	110100	111011	1	5
1 1	Perform Arithematic right shift operation	111101	001110	1	3

0 1	Addition (A+M)	010111		
		010100		
	perform Arithmetic right shift operation	001010	000111	0 2
1 0	Subtract M	101001		
		110011		
	Perform Arithmetic right Shift operation	111001	100011	1 1
1 1	perform Arithmetic right Shift operation	111100	110001	1 0

$Q_{n+1} = 1$, it means the output is negative

Hence, 23 * -9 = 2's complement of 111100110001 ⇒ (000011001111).

5) Explain about Decimal Subtraction Operation using flowchart and hardware configuration with an example.

Ans⇒