

|  |  |
| --- | --- |
| Name | Aryan Dilipbhai Lanhanoja |
| Roll No | 92200133030 |

## What do You mean by radix or base of a number system? Briefly describe why hex representation is used for the addresses and the contents of the memory locations in computer's main memory.

* The radix or base of a number system is the number of unique symbols or digits that are used to represent numbers in that system. For example, the decimal system (base 10) uses ten digits from 0 to 9 to represent numbers, while the binary system (base 2) uses two digits 0 and 1. Hexadecimal is a base-16 number system that uses 16 digits, including the digits 0-9 and the letters A-F to represent numbers.
* Hexadecimal is convenient for representing memory addresses because it allows for compact representation of large addresses, which can be difficult to represent using decimal numbers. For example, a 32-bit memory address can be represented using eight hexadecimal digits, whereas it would require 10 decimal digits to represent the same address.
* the hex representation is also used for the contents of memory locations because it allows for easy conversion between binary and hexadecimal, which is important in computer programming and digital electronics.

## What do you understand by I's and 2's complements of a binary number? What will be the range of decimal numbers that can be represented using 16-bit 2's complement format?

* 1. Ans:-
  2. 1's complement is simply a Bitwise NOT gate, i.e. 1011 becomes 0100.To get 2's complement of binary number is 1's complement of given number plus 1 to the least significant bit (LSB).
  3. A 16-bit integer can store 216 (or 65,536) distinct values. In an unsigned representation, these values are the integers between 0

and 65,535; using two's complement, possible values range from

−32,768 to 32,767.

## Briefly describe the salient features of IEEE- 754 standard of representing floating-point numbers.

**Ans:-**

* 1. Some of the salient features of the IEEE-754 standard for representing floating-point numbers are:

1. Sign bit: The leftmost bit represents the sign of the number, where 0 indicates a positive number and 1 indicates a negative number.
2. Exponent: The next few bits represent the exponent of the number, which determines the scale of the number. The exponent is biased by a fixed value to allow both positive and negative exponents to be represented.
3. Mantissa: The remaining bits represent the mantissa of the number, which is the significant digits of the number.
4. Normalization: The mantissa is normalized so that it has a leading 1 bit, which allows for efficient arithmetic operations
5. Special values: The standard also defines special values such as positive and negative infinity, NaN (Not a Number), and denormalized numbers.
6. Precision: The precision of the floating-point number is determined by the number of bits used to represent the mantissa. The standard defines several formats with different precision levels, such as single-precision (32 bits) and double-precision (64 bits).

## Why was it considered necessary to carry out a revision of IEEE-754 standard? What are the main features of IEEE-754r notation for IEEE-754 under revision?

**Ans:-** Some of the key reasons for the revision were:

* 1. Increased demand for higher precision and accuracy in floating-point arithmetic.
  2. The need to address some ambiguities and inconsistencies in the original standard.
  3. The desire to improve compatibility with other standards, such as the decimal arithmetic standard.
  4. The need to support new features and operations, such as decimal floating- point arithmetic and support for exceptional values.

The main features of IEEE-754r notation for IEEE-754 under revision are:

* + 1. Decimal floating-point arithmetic: The revision introduces support for decimal floating-point arithmetic, which allows for more accurate representation and manipulation of decimal values.
    2. Improved exception handling: The revision defines new handling for exceptional conditions, such as overflow and underflow, to provide more reliable and predictable results.
    3. Additional precision levels: The revision adds new precision levels, such as quadruple precision (128 bits) and half-precision (16 bits), to provide greater flexibility in representing floating-point values.
    4. Improved rounding modes: The revision introduces new rounding modes, such as directed rounding and round-to-nearest-ties-to- even, to provide more flexibility in rounding operations.
    5. Improved compatibility with other standards: The revision improves compatibility with other standards, such as the decimal arithmetic standard, to provide more consistent and interoperable floating-point arithmetic.

## What represents in a floating-point representation :

Ans:-

## a the range of representable numbers;

**b** **precision with which a given number can be represented?**

* The range of representable numbers is determined by the number of bits used for the exponent and the mantissa, as well as the bias applied to the exponent. For example, in the IEEE-754 single- precision format, the exponent is 8 bits with a bias of 127, and the mantissa is 23 bits. This allows for a range of representable numbers from approximately 1.4 x 10^-45 to 3.4 x 10^38.
* The precision with which a given number can be represented is determined by the number of bits used for the mantissa. The more bits used for the mantissa, the greater the precision of the representation. For example, in the IEEE-754 single-precision format, the mantissa is 23 bits, which allows for a precision of approximately 7 decimal digits.

## Why is there need to have floating-point standards that can take care of decimal data and decimal arithmetic also in addition to binary data and arithmetic?

**Ans:-**

* + - * There is a need to have floating-point standards that can take care of decimal data and arithmetic because many real-world applications deal with decimal data, such as financial calculations, measurements, and scientific computations. These applications require a high level of precision and accuracy in representing and manipulating decimal data, which can be challenging to achieve using binary floating- point arithmetic.
      * In binary floating-point arithmetic, numbers are represented as a binary fraction and an exponent of 2, which can lead to inaccuracies when dealing with decimal data. For example,

the decimal value 0.1 cannot be represented exactly in binary floating-point arithmetic, which can result in rounding errors and other inaccuracies in calculations.

* + - * To address these issues, decimal floating-point standards have been developed, such as the IEEE-754r standard for decimal floating-point arithmetic. These standards use a different format for representing numbers, with a binary- coded decimal (BCD) fraction and an exponent of 10, which allows for more accurate representation and manipulation of decimal data.
      * In addition, decimal floating-point standards also provide support for decimal rounding modes, which are important for financial calculations and other applications that require precise rounding behaviour.

# PROBLEMS

## A.Do the conversions as indicated. a. Eight-bit 2's complement representation of (-23)10

**B. Decimal equivalent of (00010111), represented in 2's complement form.**

**Ans:-**

a. 8 bits 2s complement of (-23)10 = (- 00010111)2 1s complement: (11101000)

2s complement(Adding 1): (11101000) +

(00000001) = (11101001) 1.

b. 8 bits 2s complement of (-23)10 = (- 00010111)2 1s complement: (11101000)

2s complement: (11101000) + (00000001) =

(11101001) 1.

The decimal equivalant of (00010111)2 represened in 2s complement form.

## Two possible binary representations of (-1)10 are (10000001), and (11111111). One of them belongs to sign-bit magnitude format and the other to the 2's complement format. Identify.

**Ans:-**

In sign-bit magnitude format, the most significant bit represents the sign of the number, with 0 indicating positive and 1 indicating negative. The remaining bits represent the magnitude of the number. In this case, the most significant bit is 1, which means the number is negative, and the remaining bits (00000001) represent the magnitude, which is 1.

Therefore, the decimal equivalent of (10000001) in sign-bit magnitude format is -1.

In 2's complement format, the most significant bit is also used to represent the sign of the number, but the remaining bits represent the 2's complement of the magnitude of the number. In this case, the most significant bit is 1, which means the number is negative, and the remaining bits (11111111) represent the 2's complement of the magnitude of the number, which is 1. Therefore, the decimal equivalent of (11111111) in 2's complement format is also -1.

Objective

1. The hexadecimal number (FFF.FF)16 in floating-point notation is written as re, respec-
   1. FFFFF x 162
   2. FFFFF x 23

C. FFFFF x 163

d. FFFFF

**Ans:- C**

1. Two's complement of a certain binary number is 11100101. The binary number is

a. 00011011

b. 00011010

c. 11100110

d. indeterminate from given data

## Ans:-A

1. 15's complement of (0E15.EA)16 is a. (F1EA15)16

b. (F2EA25)16

c. (EOD9FB)16

d. None of these

## Ans:-C

1. One's complement of (1101), in 8-bit arithmetic will be a. 00000010

b. 11110010

c. 00100000

d. None of these

## Ans:-B

15. A certain floating-point format has a 24-bit mantissa and an 8-bit exponent. The range of decimal numbers that can be represented using this format would be approximately

a. 10-38 to 10+38

b. --128 to +127

c. 10 to 10+8

d. 10-77 to 10+77

## Ans:-A

1. is the code capable of error detection and correction.
2. Hamming code
3. Repetition code
4. Unicode
5. Both (a) and (b)

## Ans:-A

1. The generalized form of Hamming code is (P' and 'D' respectively represent parity and data bits)

a. PP2D,P,D2D3D4P4...

b. P,D,P2D2P,D3P4D4...

c. PP2D D2P3P4D,D4...

d. DD2P D3P2P3P4D4...

**Ans:-B**

# Blanks

## Decimal equivalent of 8421 BCD number 10000000.1001 is a. 128.9

**b. 80.9**

## c. 128.5625

**d. None of these Ans:-**

The given BCD number 10000000.1001 represents the decimal number:

1 x 10^7 + 1 x 10^-1 + 0 x 10^-2 + 0 x 10^-3 + 1 x 10^-4

10000000.1001 = 10,000,000.1 + 0.0001

= 10,000,000 + 0.5 + 0.0625

= 10,000,000.5625

the decimal equivalent of the given 8421 BCD number is 10,000,000.5625.

So, the answer is **c) 128.5625.**

## Two's complement of a certain binary number is 11100101. The binary number is

**a. 00011011**

## b. 00011010

**c. 11100110**

## d. indeterminate from given data Ans:-

1’s Complement is:-00011010 Then we add 1:-

000110011

So The original Binary Number is 000110011.