**19ELC302 – EMBEDDED DIGITAL SIGNAL PROCESSING**

**DATE:** 21/09/2024 **NAME: SUBA KEERTHANA K**

**ROLL NUMBER : CB.EN.U4ELC2217 Expt No.:** 04

**CONVERSION OF NUMBERS**

**AIM:**

To write a MATLAB code to convert decimal point number to single precision floating point IEEE754 format and vice-versa.

**SOURCE CODE:**

**TASK 1:** To convert the 32 bit (Single Precision) floating point (IEEE 754 format) to decimal number.

% SUBA KEERTHANA K CB.EN.U4ELC22017

% To convert the floating-point number to decimal point number.

% 32-bit (single precision) floating point to decimal

% EXP-4A

clear

clc

Float = [0 1 0 0 0 0 0 1 1 1 0 1 0 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1];

n = 0;

e = 0;

mantisa = 0;

r = -1;

% Extract exponent from bits 2 to 9

for i = 9:-1:2

e = e + (Float(i) \* (2^n));

n = n + 1;

end

exp = e - 127; % Adjust exponent using bias

% Extract mantissa from bits 10 to 32

for m = 10:32

mantisa = mantisa + (Float(m) \* (2^r));

r = r - 1;

end

% Calculate the decimal value of the floating-point number

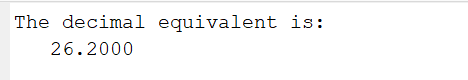
decimal = ((-1)^Float(1)) \* (1 + mantisa) \* (2^exp);

% Display the result

disp('The decimal equivalent is: ');

disp(decimal);

**OUTPUT:**



**TASK 2:** To convert decimal number to 32 bit (Single Precision) floating point (IEEE 754 format).

% SUBA KEERTHANA K CB.EN.U4ELC22017

% Expt 4 - Task 2

% AIM: To convert a decimal number to a 32-bit (Single Precision) IEEE 754 floating point format.

% Input : 26.2

clear;

clc;

% Input decimal number to be converted

decimal = -13.75;

% Step 1: Determine the sign bit

if decimal < 0

signBit = 1; % Negative number

decimal = abs(decimal); % Work with the absolute value

else

signBit = 0; % Positive number

end

% Step 2: Calculate the unbiased exponent

e = 0;

exponentValue = floor(log2(decimal)); % Get exponent value without bias

e = exponentValue + 127; % Add bias (127) to the exponent

% Step 3: Initialize mantissa and calculate fractional part

mantissa = 0;

mantissaValue = decimal / (2^exponentValue) - 1; % Normalized mantissa value

r = -1;

% Step 4: Calculate mantissa bits

mantissaBits = zeros(1, 23); % Array to hold 23 bits of mantissa

for i = 1:23

if mantissaValue >= 2^r

mantissaBits(i) = 1; % Set the corresponding bit to 1

mantissaValue = mantissaValue - 2^r; % Subtract the value

else

mantissaBits(i) = 0; % Set the corresponding bit to 0

end

r = r - 1; % Move to the next fractional place

end

% Step 5: Convert exponent to binary

n = 0; % Initialize counter for exponent bits

exponentBits = zeros(1, 8); % Array for 8-bit exponent

for i = 1:8

exponentBits(9-i) = mod(e, 2); % Get the least significant bit

e = floor(e / 2); % Shift right

end

% Step 6: Combine sign, exponent, and mantissa to form the final float

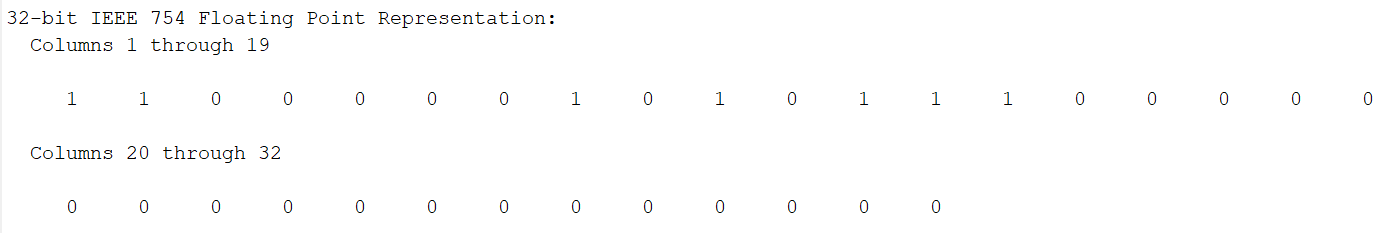
Float = [signBit, exponentBits, mantissaBits];

% Display the result

disp('32-bit IEEE 754 Floating Point Representation:');

disp(Float);

**OUTPUT:**



**RESULT:**

Hence, MATLAB code to convert decimal number to 32-bit floating-point IEEE 754 format number and vice-versa is written and executed successfully.