```
% % Mfile name
%   mtl_aae_sim_dec2bin.m

% Version:
%   Matlab R2007a

% Revised:
%   August 28, 2008

% Authors:
%   Luke Snyder, Dr. Autar Kaw
%   University of South Florida
%   kaw@eng.usf.edu
%   Website: http://numericalmethods.eng.usf.edu

% Purpose
%   To illustrate the concept of the conversion of a decimal number to
%   fixed point binary format.

% Keywords
%   Decimal to binary conversion
%   Fixed point register

% Clearing all data, variable names, and files from any other source and
% clearing the command window after each successive run of the program.
clc
clear all

% Inputs:
%    This is the only place in the program where the user makes the changes
%    based on their wishes.
    % Enter positive number to be converted to binary number:
    decnum = 54.3;

    % Enter number of bits to be used in "integer part"
    m = 8;

    % Enter number of bits to be used in "fractional part"
    n =6;

% ************************************************************************

disp(sprintf('\n\nConcepts of Conversion of Base 10 Number to Base 2'))
disp(sprintf('Fixed Point Register Binary Number'))
disp(sprintf('\nUniversity of South Florida'))
disp(sprintf('United States of America'))
disp(sprintf('kaw@eng.usf.edu'))
disp(sprintf('Website: http://numericalmethods.eng.usf.edu'))
disp(sprintf('\nNOTE: This worksheet illustrates the use of Matlab to convert'))
disp('a decimal number to a fixed point binary number.')

disp(sprintf('\n************************Introduction*************************'))

disp(sprintf('\nThe following worksheet illustrates how to convert a positive'))
```

```
disp(sprintf('decimal number to a fixed point register binary number using '))
disp(sprintf('loops and various conditional statements. The user inputs a '))
disp(sprintf('decimal number in the Input section of the program, and then an '))
disp(sprintf('equivalent binary number is given as an output.'))

disp(sprintf('\n\n**************************Input Data*****************************'))

disp(sprintf('     Decimal number to convert to binary number, decnum = %g', decnum))
disp(sprintf('     Number of bits to be used for "integer part", m = %g', m))
disp(sprintf('     Number of bits to be used for "fractional part", n = %g',n))

disp(sprintf('\n\n**************************Procedure*****************************'))

% Calculating the maximum possible value given the number of bits as
% specified by the user.

sum1 = 0;
for i=0:1:m-1
    sum1 = sum1 + 1*2^i;
end

sum2 = 0;
for i=1:1:n
    sum2 = sum2 + 1*2^(-i);
end

maxval = sum1 + sum2;

% Calculating the minimum value that can be represented.
minval = 1 * 2^(-n);


% Here we determine whether or not the number of bits is sufficient to
% represent the given decimal number.
if maxval < decnum
    disp(sprintf('\nGiven the number of bits used for integer and fractional part,'));
    disp(sprintf('the maximum number represented as a base-2 number is, maxval = %g.',↙
maxval));
    disp(sprintf('\nSince this value, maxval = %g is less than the decimal value',↙
maxval));
    disp(sprintf('to be represented, decnum = %g, the number of bits to be used is  ',↙
decnum));
    disp(sprintf('insufficient to represent this number in binary format. Please'));
    disp(sprintf('enter a larger number of bits to be used.'));
elseif minval > decnum
    disp(sprintf('\nGiven the number of bits used for the fractional portion of'));
    disp(sprintf('the binary conversion, the minimum value that can be represented'));
    disp(sprintf('as a base-2 number is minval = %g.',minval))
    disp(sprintf('\nSince this value is greater than the decimal number to be'))
    disp(sprintf('represented, decnum = %g, the number of bits is insufficient',decnum));
    disp(sprintf('to represent this number in fixed point binary format. Please'));
    disp(sprintf('enter a larger number of bits to be used.'));
else
    disp(sprintf('\nThe maximum value that can be represented given the number'));
```

```
    disp(sprintf('of bits to be used is, maxval = %g and the minimum value is',maxval));
    disp(sprintf('minval = %g.',minval))
    disp(sprintf('\nSince "decnum" is less than "maxval" and greater than "minval",'));
    disp(sprintf('the number of bits to be used is sufficient to represent this'));
    disp(sprintf('number in fixed point binary format.'));


% Using the floor command to isolate each section of the decimal number to
% begin binary number conversion.

disp(sprintf('-----------------------------------------------------------------------'));

before_decimal = floor(decnum);
disp(sprintf('\nThe "integer part" of the base-10 number is, before_decimal = %g',↙
before_decimal));

after_decimal = (decnum - floor(decnum));
disp(sprintf('\nThe "fractional part" of the base-10 number is, after_decimal = %g',↙
after_decimal));

% Calculating the binary conversion of the value before the decimal.
% Here we use the variable "Bold" to denote the value of the "integer part"
% of the decimal number.
Bold = before_decimal;

% Using a loop to perform the binary conversion repetitively until the
% necessary number of bits are used.
for i=1:1:m

    % Using the floor command to calculate the Quotient of the decimal
    % number.
    Quotient = floor(Bold/2);

    % Using a combination of the ceil and floor commands to determine the
    % remainder, and inputting this value into an array.
    Binary(i) = ceil(Bold/2) - floor(Bold/2);

    % This "if" statement is used to break the loop when the Quotient is
    % zero.  When this occurs, the remainder will always be 1 and no more
    % bits are necessary. If the "integer part" is initially 0 however, the Binary
    % value is always equal to 0.
    if Quotient == 0 && before_decimal~=0
        Binary(i) = 1;
        break
    elseif before_decimal==0
        Binary(i) = 0;
        break
    end

    % Here we reinitialize the value of Bold so that a new Quotient can be
    % found based on the previous one.
    Bold = Quotient;
end
```

```
% Using this method, the placement of the binary numbers in the array
% "Binary" are backwards and so must be flipped around to form the actual
% binary number.
t = length(Binary);

for j = 1:1:t
     Binary_Before_Dec(j) = Binary((t+1)-j);
end

% Converting the "fractional part" of the decimal into a binary number.
% Here we initialize the sum using a new value, BoldA which represents the
% value after the decimal point.
BoldA = after_decimal;

for i=1:1:n

    % When converting the fractional portion, we first multiply by 2.
    Num = BoldA*2;

    % If the number before the decimal after multiplication is 0, or i.e.,
    % the variable "Num" is less than 1, we enter 0 as a binary number.
    if Num < 1
        Binary_After_Dec(i) = 0;

    % In the case where the number before the decimal after multiplication
    % is greater than 0, the appropriate binary number is 1.
    elseif Num >= 1
        Num = Num - floor(Num);
        Binary_After_Dec(i) = 1;
    end

    % Reinitializing the value of BoldA to be Num so that the process can
    % continue repetitively.
    BoldA = Num;

end

% Displaying the Binary Numbers:

% BBDLen represents the length of the array of the binary number before the
% decimal.
BBDLen = length(Binary_Before_Dec);

% BADLen represents the length of the array of the binary number after the
% decimal.
BADLen = length(Binary_After_Dec);

% Here we use loops to represent the binary number before the decimal
% as a string.
for i = 1:1:BBDLen
    str1(i) = num2str(Binary_Before_Dec(i));
end

% Here we use loops to represent the binary number after the decimal as a
```

```
% string.
for i = 1:1:BADLen
    str2(i) = num2str(Binary_After_Dec(i));
end

% The final binary number is the concatenation of the above strings before
% and after the decimal.
catstr= [str1,'.',str2];
strA = ['The number ', num2str(decnum),' represented in fixed point binary format is '];

Final_Binary = [strA,catstr];

fprintf('\n');
disp(Final_Binary);

% Displaying the total number of bits required to represent the number.
total_bin_length = length(Binary_Before_Dec) + length(Binary_After_Dec);
user_def_bits = m + n;
fprintf('\n');
str1 = ['The equivalent base-10 number of the binary representation ',catstr];
disp(str1)
str2 = ['is ',num2str(decnum)];
disp(str2)


% Finding the relative difference between the fixed point binary
% representation and number to be represented.

% Using the floor command to isolate the "integer" part of the base-2 number.
int_bin = floor(str2num(catstr));

% Determining the length of the entire binary number and all the characters
% in the string.
n = length(catstr);

% Converting each individual number in the "integer" part back to a string
% which also enters it into a character array.
str_int = num2str(int_bin);

% Determining the length of the character array for use in loop.
m = length(str_int);

% Using a loop to sum values of the "integer" portion of the base-2 number.
% The loop variable 'sumint' is used for summation and is initialized at 0.
sumint = 0;

for i = 1:1:m

    % Converting each number in the character array to a number in another
    % array.
    bin_int(i) = str2num(str_int(i));

    % Summing values based on the values in the new array.  This value is
    % the summation of each individual digit in the "integer" part of the
```

```
    % base-2 number.
    sumint = sumint + bin_int(i)*2^(m-i);

end

% Using a loop to sum values of the "fractional" portion of the base-2
% number. The loop variable 'sumfrac' is used for summation and is
% initialized at 0.
sumfrac = 0;

% This loop variable, 'j' is used to create a new array of numbers that
% represent the "fractional" portion of the binary number.
j = 1;

% Note that the starting point in this loop is the length of the integer
% portion (m), plus 2 which effectively "skips" the decimal point in the
% character array.  The ending point (n) is equal to the length of the entire
% character array (bin_num).
for i = m+2:1:n

    % Using the loop variable 'j' to create a new number array.
    bin_frac(j) = str2num(catstr(i));

    % Creating a new character array of only the "fractional" portion for
    % later use.
    bin_frac_str(j) = catstr(i);

    % Summing values of the "fractional" portion using loop variable
    % 'sumfrac'.
    sumfrac = sumfrac + bin_frac(j)*2^(-j);

    % Adding '1' to the loop variable each time such that it creates a new
    % position in the array for the next iteration.
    j = j+1;

end

% Adding the "fractional" portion of the base-2 number with the "integer"
% portion which yields the base-10 number.
total_dec = sumint + sumfrac;

% Finding the relative percentage difference between the representation and
% the number to be represented.
tot_perc_diff = ((decnum - total_dec)/decnum)*100;

disp(sprintf('\nGiven the number of bits to be used for the integer and fractional '))
disp(sprintf('part, the relative difference between the number to be represented'));
disp(sprintf('and the fixed point binary representation is, tot_diff = %g.',↙
tot_perc_diff));

end

disp(sprintf('\n\n*************************Conclusion*****************************'))
disp(sprintf('This worksheet illustrates the use of Matlab to convert a base-10 '))
```

```
disp(sprintf('number to a base-2 binary number. It is important to understand '))
disp(sprintf('the binary system as it has numerous applications. Critical to this '))
disp(sprintf('understanding is being able to convert binary numbers to base-10 '))
disp(sprintf('numbers, and vice-versa.'))

disp(sprintf('\n\n**************************Refrences*****************************'))
disp('See: <a href = "http://numericalmethods.eng.usf.↙
edu/mtl/gen/01aae/mtl_gen_aae_txt_binaryrepresentation.pdf">Binary Representation of↙
Numbers</a>')

disp(sprintf('\n\nLegal Notice: The copyright for this application is owned'))
disp(sprintf('by the author(s). Neither MathWorks nor the author(s)'))
disp(sprintf('are responsible for any errors contained within and are '))
disp(sprintf('not liable for any damages resulting from the use of this'))
disp(sprintf('material. This application is intended for non-commercial,'))
disp(sprintf('non-profit use only. Contact the author for permission if'))
disp(sprintf('you wish to use this application in for-profit activities.'))
```