Tutorial 1 of MA2213

Qian Lilong

Department of Mathematics

National University of Singapore

September 3, 2018

qian.lilong@u.nus.edu

1 Decimal to Binary system

1.1 convert the integer to binary

$$64_{10} \to \left\{ \begin{array}{l} 2)\underline{64} \quad \uparrow \; 0 \\ 2)\underline{32} \quad \mid \; 0 \\ 2)\underline{16} \quad \mid \; 0 \\ \quad 2)\underline{8} \quad \mid \; 0 \\ \quad 2)\underline{4} \quad \mid \; 0 \\ \quad 2)\underline{2} \quad \mid \; 0 \\ \quad 2)\underline{1} \quad \mid \; 1 \end{array} \right\} 1000000_2 \tag{1}$$

$64_{10} \Rightarrow \boxed{1}\,\boxed{0}\,\boxed{0}\,\boxed{0}\,\boxed{0}\,\boxed{0}\,\boxed{0}$

$$17_{10} \to \left\{ \begin{array}{l} 2)\underline{17} \quad \uparrow \; 1 \\ \quad 2)\underline{8} \quad \mid \; 0 \\ \quad 2)\underline{4} \quad \mid \; 0 \\ \quad 2)\underline{2} \quad \mid \; 0 \\ \quad 2)\underline{1} \quad \mid \; 1 \end{array} \right\} 10001_2 \tag{2}$$

$17_{10} \Rightarrow \boxed{1}\,\boxed{0}\,\boxed{0}\,\boxed{0}\,\boxed{1}$

$$227_{10} \to \left\{ \begin{array}{l} 2)\underline{227} \quad \uparrow \; 1 \\ 2)\underline{113} \quad \mid \; 1 \\ \quad 2)\underline{56} \quad \mid \; 0 \\ \quad 2)\underline{28} \quad \mid \; 0 \\ \quad 2)\underline{14} \quad \mid \; 0 \\ \quad 2)\underline{7} \quad \mid \; 1 \\ \quad 2)\underline{3} \quad \mid \; 1 \\ \quad 2)\underline{1} \quad \mid \; 1 \end{array} \right\} 11100011_2 \tag{3}$$

$227_{10} \Rightarrow \boxed{1}\,\boxed{1}\,\boxed{1}\,\boxed{0}\,\boxed{0}\,\boxed{0}\,\boxed{1}\,\boxed{1}$

The matlab function to convert a integer to binary is `dec2bin(N)`, where N is the integer.

1.2 Convert the fractional part to binary

- For 10.5

  For the integer part

$$10_{10} \to \left\{ \begin{array}{l} 2)\underline{10} \quad \uparrow \; 0 \\ \quad 2)\underline{5} \quad \mid \; 1 \\ \quad 2)\underline{2} \quad \mid \; 0 \\ \quad 2)\underline{1} \quad \mid \; 1 \end{array} \right\} 1010_2 \tag{4}$$

  For fractional part

$$\begin{array}{r|r} & .\quad 5 \\ & \times 2 \\ \hline {\color{red}1} & .\quad 0 \end{array} \qquad \downarrow \qquad \boxed{1} \tag{5}$$

  So

$$10.5_{[10]} = 1010.1_{[2]} \tag{6}$$

- For 12.8

$$12_{10} \to \left\{ \begin{array}{l} 2)\underline{12} \quad \uparrow \; 0 \\ \quad 2)\underline{6} \quad \mid \; 0 \\ \quad 2)\underline{3} \quad \mid \; 1 \\ \quad 2)\underline{1} \quad \mid \; 1 \end{array} \right\} 1100_2 \tag{7}$$

1

For the fractional part

$$
\begin{array}{c|r|c}
 & .\quad 8 & \\
 & \times 2 & \\
\hline
1 & .\quad 6 & \boxed{1} \\
 & \times 2 & \\
\hline
1 & .\quad 2 & \boxed{1} \\
 & \times 2 & \\
\hline
0 & .\quad 4 & \boxed{0} \\
 & \times 2 & \\
\hline
0 & .\quad 8 & \boxed{0} \\
 & \times 2 & \\
\hline
\vdots & \vdots & \vdots
\end{array}
\tag{8}
$$

So

$$12.8_{[10]} = 1100.\overline{1}\,\overline{1}\,\overline{0}\,\overline{0}_{[2]} \tag{9}$$

- For 55.4

  The integer and fractional parts convert:

$$
55_{10} \rightarrow
\left\{
\begin{array}{l|l}
2)\,55 & 1 \\
2)\,27 & 1 \\
2)\,13 & 1 \\
2)\,6 & 0 \\
2)\,3 & 1 \\
2)\,1 & 1
\end{array}
\right\} 110111_2,
\qquad
\begin{array}{c|r|c}
 & .\quad 4 & \\
 & \times 2 & \\
\hline
0 & .\quad 8 & \boxed{0} \\
 & \times 2 & \\
\hline
1 & .\quad 6 & \boxed{1} \\
 & \times 2 & \\
\hline
1 & .\quad 2 & \boxed{1} \\
 & \times 2 & \\
\hline
0 & .\quad 4 & \boxed{0} \\
 & \times 2 & \\
\hline
\vdots & \vdots & \vdots
\end{array}
\tag{10}
$$

  Hence $55.4_{10} = 110111.\overline{0}\,\overline{1}\,\overline{1}\,\overline{0}_2$

- For 1/3

  We only need to convert the fractional part:

$$
\begin{array}{c|r|c}
 & .\quad 1/3 & \\
 & \times 2 & \\
\hline
0 & +\quad 2/3 & \boxed{0} \\
 & \times 2 & \\
\hline
1 & +\quad 1/3 & \boxed{1} \\
 & \times 2 & \\
\hline
\vdots & \vdots & \vdots
\end{array}
\tag{11}
$$

  Hence $\dfrac{1}{3} = 0.\overline{0}\,\overline{1}_2$

Since there is not built-in function in matlab to transform float number to binary, we write a function to calculate it. Usage:

```
>>  f = float2bin(55.4)
f =
    '110111.<0110>...'
```

The part included by "<>" is the repeated digits.

```matlab
function u = float2bin(f)
%This function converts a floating point number to a binary string.

%Floating Point Binary Formats

%Input checking
b = mod(f,1);                    % the fractional part
a = f - b;                       % the integer part

%% for the integer part:

u1 = dec2bin(a);

%%  for the fractional part
    %% initial u2 ,
    u2 = [];                     % empty vector
    B = [];
                                 % fractional part of 2*b
    while  b~=0 && notin(b,B)    % t
         B   = [B,b];
         b   = 2*b;              % calculate 2b
         b_f = mod(b,1);         % fractional part of 2b
         t   = num2str(int32(b-b_f));     % interger part of 2b
         b   = b_f;
         u2  = [u2,t];           % append a 0 or 1 to u if t is 0 or 1
    end
    if ~notin(b,B)               % if repeated
        [~,i] = notin(b,B);
        u2    = [u2(1:i-1),'<',u2(i:end),'>...'];
    end


u = [u1,'.',u2];                      % put integer and fractional parts together
end

function [y,j]=notin(c,d)
%%  fucntion to check whether c is in vector b
%     if not, return 1
%     otherwise, we also return the index the first one
    y=1;
    for j=1:length(d)
        if abs(d(j) - c)< 1e-8 % may have rounding error
            y=0;
            break
        end
    end


end
```