

Лабораторна робота №2

Тема. Аналого-цифрове та цифро-аналогове перетворення. Формати представлення даних.

Завдання

1. Написати програму конвертації двійкового коду в вісімковий код (вбудовані функції не використовувати).
2. Візуалізація роботи 6 бітного АЦП для трикутного сигналу амплітудою 130 мВ , з періодом по фронту 0,01 мс.

Хід роботи

1. Для перетворення двійкового коду в вісімковий використовується метод розділення вхідного двійкового числа на числа по 3 біта, які потім легко перетворити у вісімкові цифри наступним чином:

Binary	Octal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Функція :

```
function y = bin_to_oct(b)
i = size(b, 2);
n = ceil(i / 3);

% b - двійкове число
% o - вісімкове число

for j = n : -1 : 1
    if i > 3
        o(j) = bin_to_oct_num(b(i - 2 : i));
        i = i - 3;
    else
        o(j) = bin_to_oct_num(b(1 : i));
    end
end

y = o;
```

Функція для переведу в вісімкову систему із двійкової:

```
function y = bin_to_oct_num(x)
switch x
    case {'000', '00', '0'}
        y = 0;
    case {'001', '01', '1'}
        y = 1;
    case {'010', '10'}
        y = 2;
    case {'011', '11'}
        y = 3;
    case {'100'}
        y = 4;
    case {'101'}
        y = 5;
    case {'110'}
        y = 6;
    case {'111'}
        y = 7;
end
```

Результат роботи даної програми:

```
>> bin_to_oct('1110111011')

ans =

    1     6     7     3
```

2. Функція для генерації потрібного трикутного сигналу:

```
function y = sawt(t)
T = 0.01;
fs = 18 / 0.01;
A = 0.13;

y = A * sawtooth(2 * pi * (1 / T) * t, 1/2);
end
```

Скрипт:

```
fun_handle = @sawt;           % Показчик на функцію
X0 = 0;                       % Початок інтервалу, в якому проводиться дискретизація і квантування
X1 = 0.03;                    % Кінець інтервалу, в якому проводиться дискретизація і квантування
bit_capacity = 6;             % Розрядність в бітах
sampling_time = 0.01/18;     % Період дискретизації

[Y, X] = SamplingAndQuantization(fun_handle, X0, X1, sampling_time, bit_capacity);

% Виведення графіка функції
subplot(1, 3, 1);
fplot(fun_handle, [X0, X1]);
grid;
title('Function');
xlabel('Time [S]');
ylabel('Amplitude');

% Виведення графіка функції дискретизованого та квантованого сигналу
subplot(1, 3, 3);
stairs(X, Y);
grid;
title('Sampled and quantized signal');
xlabel('Time [S]');
ylabel('Amplitude');
```

Результат роботи даної програми:

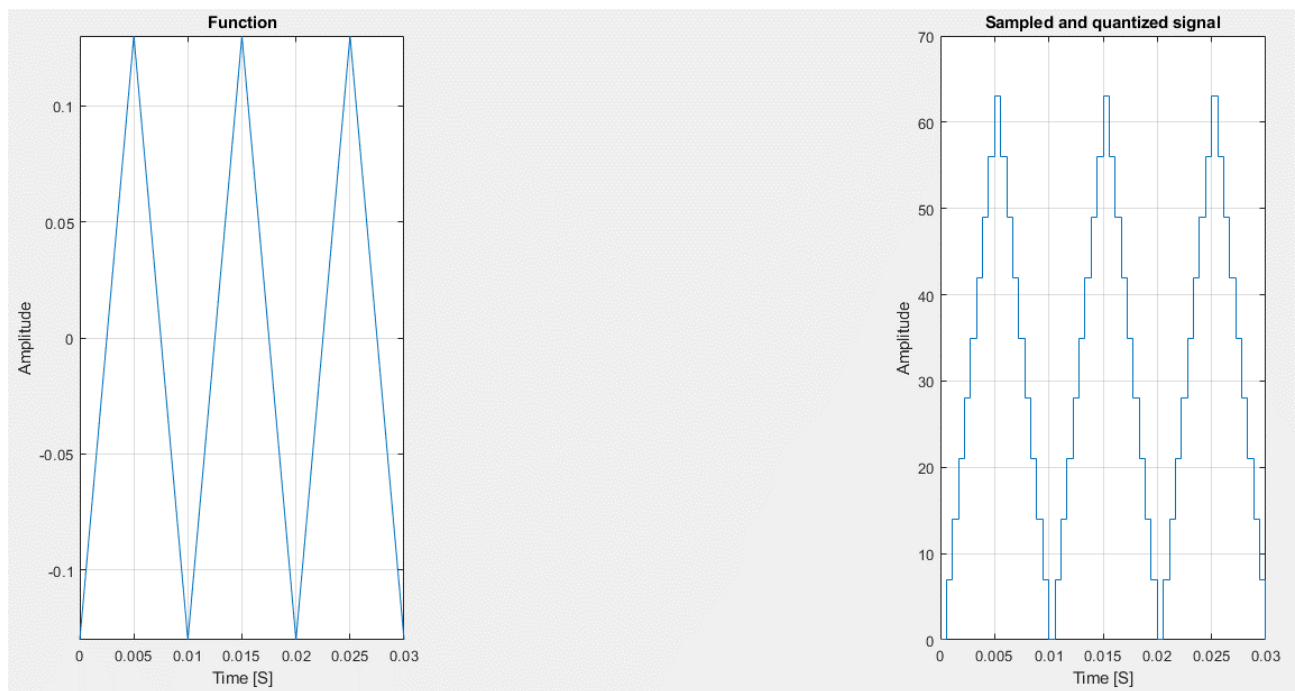


Рисунок 1

Як бачимо з графіків, сигнал генерується та перетворюється коректно.