Digital Signal Processing [Lab-8]

Table of Contents

- Authors: *Kshitij Srivastava(1412001XXX)* and *Nilambar Saha(1412001XXX)*
- Lab Instructor: *Dr. Ravi Kant Saini*

Objective:

```
%DIT-FFT Algorithm (In this Algorithm, we use DIT-FFT and DIF-FFT Algorithm
%to compute DFT with very less time compelexity. This was then used to see
%the power spectral density)
```

Program:

```
clc;
clear all;
close all;

% * |*Matlab Commands for finding the DFT by DIT-FFT Algorithm*|

x=[2,0,2,0,2,0,2,0];
length=8;%Length of the input
levels=log2(8);%no of multiplication and addition required

index=0:7; %index value of variable x
bit_index=zeros(1,8);%Initializing the variable bit_index
bit_index=bitrevorder(index); % To store the bit reversed index of x

for i=1:8
    x_new(i)=x(bit_index(i)+1); %Finding the array from bit reversed indexes
end

input=x_new;%input to the level is the array from bit reversed indexes
for i=1:3 %Number of levels in DIT

    %W_upper for array with value 1 to be multiplied to upper blocks
    W_upper=ones( 1, 2^(i-1) ); % 2^(level-1) where level=i

    for j=0:( (2^i) /2)-1  %For finding the index for W_lower (j)
        W_found=W( 2^i,j);% W(2^i,2^level-number)
        W_lower(j+1)=W_found;%W_lower stored
    end
    Wxxx=horzcat(W_upper,W_lower);%Concatenating the W_upper and W_lower
    %to create a variable Wxxx to be multiplied to a block
```

```
    %To multiply W with the blocks
    index_plus=(2^i)-1;% Variable to find till what index the block
    %has to be taken
    for k=1:(2^i):8 % Varible k for finding the starting of the block at
        %a level i
        sub_block=input(k:k+index_plus ); %Finding the block from the input
        x_temp(k:k+index_plus)=sub_block.*Wxxx; % Multiplying W with the block
    end

    %Butterfly Addition and Subtraction
    new_block=[];%Creating a empty array for output after every level
    for j=1:(2^i):8 %For finding the starting vertices of every block
        block_index=(2^i)-1;  %(2^i)-1, For finding the end of each block
        block=x_temp(j:j+block_index);%Selecting a block from above step

        add_index=(2^(i-1) )-1; %(2^(level-1) )-1  level=i
        add_block=block(1:1+add_index);%Finding the addition block from the
        %block where addition will take place
        diff_block=block(1+add_index+1:2+2*add_index);%Finding the
        %difference block from block where subtraction will take place

        new_add_block=add_block+diff_block; %Additions
        new_diff_block=add_block-diff_block; %Subtractions
        new_block=horzcat(new_block,new_add_block,new_diff_block);
        %Horizontally concatenating the new_block, new_add_block and
        %new_diff_block to create output of each level
    end
     input=new_block;%Output of each level becomes input to next level
end

xfft_dit=round(new_block);%Rounding the output to get final output
```

Results for Q1:

- **Plot for the Question No 1**

```
xfft_dit
```

```
    xfft_dit =

         8     0     0     0     8     0     0     0
```

Published with MATLAB® R2014a