Real - Time Digital Signal Processing

Prof. Rathna G N
Department of Electrical Engineering
Indian Institute of Science - Bengaluru

Lecture - 23
Lab – Filtering Using FFT

Welcome back to real time digital signal processing lab. So, in the previous lab, we saw, how we can do the DFT and FFT using MATLAB. Today, we will see how we can use code composer studio to run DFT and FFT. So, you would have seen such a simple calling FFT function in MATLAB here we had to write our own routine here in C language and we will be running it. So, welcome.

(Video Starts: 00:57)

So, we will see the code actually. So, the first one is the simulator code what it has been written. So, number of points here it is selected as 512 FFT points and FFT length is also 512 chosen. So, if it has to be modified we can modify it. So, that is what, what it says both are same just defining 2 for convenience say basically, and define FFT length is 351 that is filter length what it is chosen and then the define length s that is signal length is chosen as 3000 and signal block length.

So, you will be seeing that here it is going to be overlap method is going to be used to compute FFT. So, signal length is signal block length what it is chosen is 162 and we have to define pi here unlike MATLAB where π can be directly taken into the thing, so we have to define some structure. So, here you will be defining float real and imaginary and we are calling it as complex. So, you will be calling the function FFT, I will be showing you FFT dot c what it is going to do?

So, you are passing your Y and then, n is your number of points what you will be passing it that is FFT prototype, and then we call the function complex we are going to do the complex multiplication. So, X and then Y separately and then you have to define some of the terms as you can see the thing and then your flag is going to be set that is interrupt service routine when io buffer is full that is what it is going to be set and then we have W points is twiddle constant stored in W.

And then complex what we have samples points primary this thing working buffer and then we will be calling it FFT h number of points and then o temporary are the complex what we have defined? So, include is the data file, this is the filter coefficients what we have it so, we will try to open this. So, you can see that these are the filter coefficient values 351 of them have been pre computed using MATLAB and then stored in this file. So, those are the twiddle factors.

So, you will be going into the main and then you will be doing to the length of your signal. So, you will be generating your sin basically that is 200 into i and then you have a 267 and then 400 this is the input what you are generating that. So, you will be taking FFT that is using the coefficients. So, you will be calculating W^i of real and W^i of imaginary. So, these are the cos and then this is the sine function. So, real component of twiddle constants and these are the imaginary constants what we have it?

So, then you will be computing FFT that is h of i that is real part of it. So, you will be assigning it and then you will be with the new data what you will be calculating this. So, how we will be calculating your imaginary competence initially you will be setting it to 0 and then call the FFT function with FFT h as you have computed here with number of points. So, the bottom 1 will be showing you convolution with overlap.

So, here also we can set the breakpoint and then see whether we are going to run only FFT of it. So, the thing is what if we have seen the data file, so we will see how the FFT dot c function is written here. So, this is again number of points what it has been defined and you will be seeing that is whatever you call it as upper leg, lower leg index of upper and lower butterfly leg what you are going to implement it.

And then you should have a loop counters and difference between upper and lower leg what you will be having it, and then how many steps you are going to have it twiddle constant? And then number of stages what you have to do it. So, using butterfly structure will be computing FFT. So, as we know that number of stages is going to be $\log_2(2N)$. So, for that what will be computing other things, so the code will be shown in this way step between values in twiddle what you will be taking it how we are going to calculate?

So, you will be calculating your lower leg and then temporary files what you need it to store them and then you will be calculating Y of lower leg and then upper leg separately, that is real and imaginary separately is computed and then you will be setting your index t is equal to how many steps what we have taken into and then you will be calculating the length difference by 2 and then you will be setting your step to 2. So, we will be starting this one more loop for $i = 1 < N - 1$.

So, a bit reversal for what is it? Arranging data in the bit reversal manner. So, you will be seeing that however, we will be putting them in your input bit reversal in see what it is done. So, whereas in the case of if we use the assembly program then we can directly access bit reversal module what we discussed in the architecture in DSP processor. So, you will be calculating the complex multiplication what; you will calling the function.

So, you will be doing the real, you are seeing it real multiplied and imaginary multiplication and then imaginary part separately, so you are seeing 4 multiplications and 2 additions are happening in 1 complex multiplication, then you will be returning the Z and then it will be continuing. So, what we will do is we will just chosen this is the active debugger what we have to choose it, first what we will do is we will compile the thing for any error in the thing.

So, it is simulator as it shows dot out is up to date because I have pre compiled and then done the thing if there is any modification is done so it will be doing the recompilation. So, now what we will do is go to the debug mode. So, we will be seeing that output will be in we have so the as you can see, we have put the breakpoint here. This will show using overlap save method has been implemented here in the MATLAB you saw overlap add method so we can run the code.

So, what is happened is it is unable to find the breakpoint, what I can redo is one way of doing it is I can go and then check on the graph that is single time. What was variable sum? It was 512 was the samples and then I can because we have called floating point we can represent in full floating point. So, sampling frequency at present will keep it at this and then I can give it us let me see sample says declared here. So, you will be seeing I can make it full so you are seeing some garbage in the thing, so what we will do is, I will minimizes this.

So, I can close this and you will be seeing some memory map, why it got this thing that is it gone till the end of it. So, that is why disassembly will be having a problem because it is unable to come back, we will start computing on the board. So, you will be seeing that there was an error. So, what we will put the thing main dot c, as you can see that the breakpoint was remote otherwise in the end what you have to put the breakpoint.

So, we will compile the thing, it is loading onto the board so you have seen in the main it has enter the main function, you will be seeing the program pointer is here. So, now we will see whether I had a breakpoint hopefully it will come up to this breakpoint, we will look at it when I run the thing. There is some abortion is happening it is unable to detect this breakpoint. So, we will come back to this program little later, we will stop the thing and then we will go back to the other demo what I have it.

So, this is one of the students who has developed the FFT basically here it is for the 64 point. So, you will be seeing that even though it is in the same way what the simulator has it has been developed. So, you will be seeing that because I have put the breakpoint here to see that because end main sometimes it will be going the handle is given somewhere. So, whether it would not be able to stop the thing. So, we will see that whether the FFT is using this method that is 64 point will be going through, so I will give the run command.

So, we can see that that is graph single time. So, w is the thing so with we will see what has been declared because this will be floating so we will be declaring it as 32 bit floating point and start address will be w here also it is samples what it has been selected, we will select the thing and will display it. So, what you are seeing in the thing is there is a dip in the thing what you are locating it here in the display part of it so just give me one second what we will do is we will reset our processor once you will be seeing that there will be an error.

Because the processor has got reset so we will be coming back to this $x_1(i)$ will be the magnitude what it is going to be computed. So, as you can see that there was because I had switched on the board a little earlier there was a issue with the thing. So, now we are seeing that it has reached the breakpoint. So, this is how what you have to debug your code whenever there is a problem you can software reset or sometimes what you have to do is unplug the power supply to the board and then reconnect it.

In this case only have a switch actually reset button which I pressed it now will do the plotting of it. So, because it was giving me all garbage values last time so we will be seeing that $x(i)$ I think 64 points what we have it I will go to the this thing 32 bit floating point start address is x what I am interested in and then we will it is x_1 not x that is why it was giving me an error. So, you will be seeing that the peak has got generated. So, what is the frequency?

You can see that what is the peak value of it, what you are getting it sample somewhere around you can see that they have generated it as 10 hertz or something. So, 9 hertz, what will be getting it in this so you will be seeing that this is the frequency we can go back and then check the thing. This is what is it 64 minus what I am supposed to see 60 - 7 or something got, will cross verifying then come back what is the frequency? We will see it I forgot to check the thing. So, what is it 10 hertz, basically.

So, you are seeing that reason little error, what it has got generated, that is what, what we want to see the frequency value is 10 it is coming somewhere around 9 or something like that, the error with respect to whatever the computation, so this is how it runs on board? So, we will go back, we will stop this and then we will see directly whether I can compute my DFT. So, here I have used the FFT to show the thing so whether we can do a directly DFT.

So, here what I have is FFT calculation again? So, but I can go back to and then see that I can put individual this thing what is it dot c files to check in the same thing, whether I am going to get my DFT and FFT correctly or not. So, we have the this thing first what I will do is we will run this is FFT dot c is that so you will be seeing $N = 128$ point and complex what you have this thing declared them and then you will be calculating just like your twiddle factors cos and then sine functions what you are keeping it?

And you will be using it and we will call the FFT of samples what you are going to pass it. So, this will be your FFT dot h samples of FFT function taken from the book Rulph Chassaing basically and then dot h file, we will be defining all of them. Let us see whether this is going to run because I have not tested it, I will be testing in front of you. So, we know that it has completed it is running.

So that means to say that there are no errors in the code what it says is real input data stored in array samples. So, you will be calling FFT function. So, you will be seeing that it is going to give you a printf done and then we have to see these samples whether we have got it or not in memory. One is I can plot and then see how we have got the values? So, you will be seeing that twiddle factors how it has been generated here. So, you have at 0 to 99 and then 100 to 127 what it is showing.

So, in this thing, decimal or octal, what you want to display you can display so, one of the, it has entered into the main function, what we will do is we will put a breakpoint otherwise I will be losing the control on the thing, I want to be in the same code basically. So, we will run the thing. So, you will be seeing that input data stored in array samples, what it says, so we will see this I can view, I can use the memory browser and then I have to enter the locations here. So, we will give it as samples.

So, you will be seeing the samples in this what is it? It is representing it as 32 bit hex Texas instruments style. So, if you want to have it as this thing floating point, 32 bit floating point. So, you will be seeing that the samples are stored in the both; you will be seeing that dot real and then imaginary are 0s basically. So, the real values have been stored in this case. So, this is how you will be implementing and then you can do the IFFT so, to see that you have got back here what is the input, what you have passed into the system?

So, what it says is test frequency is 800 hertz and then sampling frequency chosen as 800 hertz and then I should be getting here it is magnitude square function in the not found out. So, you can include your code and then check whether we are going to get the results correctly. Otherwise, what you are seeing is this one so one thing what we have, what I can do is I can put the graph single time will put it so this is I will put it as I think number of samples I did not check the thing it may be 64 samples what it is generated.

So, I will put it as a 32 bit floating point and then start address can be this thing what we have is samples. And if I want I can specify the sampling frequency I know I can put it as 8000 frequency so you will be seeing that the plot is coming. So, you are seeing that 1 peak at here. What is the value? Just it has to be little here, somewhere around 25.07 what you are seeing the peak? So, this is what you will be seeing that, If I calculate the magnitude absolute of it alone, I will be getting the thing. So, what we can do is I can generate the FFT of it. Let us see

FFT magnitude alone what I want to have it, so I will give it as hopefully it is 64. Anyway, again, I did not check how many points we have chosen the thing so you can look at it. So, I will be giving my sampling frequency as 8000 and then start address samples, you want to find the FFT magnitude of it, it has taken frame size, I will give it as 6, in this case also, we will put the thing.

So, you will be seeing some garbage output is going to come. So, if it is for the input, then you will be seeing it properly. So, input is we will see that if you want to see that twiddle, we can look at it where is, my twiddle is also complex. So, the dot real what I am had to plot it basically, hopefully samples at real is not defined in the thing only I can select the samples. So, what you can do is magnitude square function, what you can calculate for this output of the samples?

And then see that your magnitude perfectly represented with 800 hertz, what we are supposed to get that thing because this is what this test frequency what it has been generated. So, at what k value of you will be getting 800 hertz is 1 one has to take it up one more, what we can do is I can remove this FFT dot c and then include a DFT if I am required, so I will put it as add files. So, I am showing you how I can use the same project file to run my different programs basically.

So, I had to select where my codes are lying. So, I will be taking the LCDK book files and then I can versions CC5, what a have it according to the chapters what you are going to have some of the codes you will be seeing that there is a DFT code also. So, if you want directly you can take it or there are different codes what it is there in this directory? So, you can use, that is 1 level DFT, what you can use it and the normal DFT is there or FFT you can use, or you can use the library which is present in the thing.

So, what we will do is we will include this file in the program, I am copying it. So, this is my dftc dot that is L138 or 6748 board what we are using it and then in this N is defined as 100 here also sampling frequencies these chosen as 8,000. So, you will be seeing result, and then real and imaginary. So, $x(k)$ is going to get a result function. And then this is your main function, what you are passing it, you are calling your DFT with the samples and then you will be completing it.

It is one more thing you have to remember in when you are using a code composer studio, that most of the variables if you have defined as a global variable just like C then you will be able to what is it view them in the graph? Otherwise if it is local variables, you will not be able to view them. So, this is one thing what you had to keep it in mind. So, you are seeing the complex samples is a global variable. So, we can still view it the thing so these are what is it, how you will be calculating here?

These are my samples real whatever cos functions, sampling frequency real my imaginary part is 0. And you will be calling DFT of the function. So, we will do the running of it I can put the breakpoint at the return so that I am within the my leg of it, so your DFT is declared here as you can see complex, x and then you will be computing directly here a real part and then imaginary parts separately. So, you will be adding them up and then to the length of it.

So, your output is going to be in the result which you are giving back to your sample. So, I can run this code so you are seeing that it is holding on here. So, what we can, one of the way of doing it is I can look at the values here as you have seen the thing when you are showing the red those are the modified values what you will be looking at it compared to your FFT and here. So, these are the little bit modified values what you will be seeing it so, that is one way of doing it in 32 bit floating point.

Or if what is it sample size is declared as complex and then we have a float basically. So, I can give it a 32 bit floating point so, if you want to view the graph, so, we can go to single time again and the number of samples I can give it as 64 and then it is going to be 32 bit floating point and start address will be samples, you will be seeing that again your peak is somewhere around because number of points what I have chosen it is approximately 20 hertz what it has taken it.

So, can you make out what was the thing 800 hertz was my sampling frequency. So, what was the number of samples what we have taken? Here is 8000 so, where you are going to get your peak, what you have to look at it? With that we will stop this lab here.

(Video Ends: 32:26)

If there are any doubts, I will be open to solve your doubts. Thank you