<Ex15>: To understand implementation of Fast Fourier Transform (FFT) on DSP.

Objective: Write a program To understand implementation of Fast Fourier Transform (FFT) on DSP using EPB_C5515 target board

Workflows you learned in the **previous lab**

- Connecting your DSP kit EPB_C5515 to CCS5.3
- Creating a new project or copying an existing project into the workspace
- Configuring the linker options and file-search paths
- Building/Compiling and running/Executing a project on the kit EPB_C5515
- Making use of breakpoints for debugging the code and using watch window to track variable values.
- Storing the data read from PC to the buffer in the CCS5.3 and view buffer data runtime.
- Generating new sine wave signal from lookup table
- Re-generating sine wave using loop-back method
- Controlling Gain of encoder for AIC3204 audio codec
- Profiling for FIR filter using linear buffering by C language and assembly language
- Profiling for FIR filter using Circular buffering by C language and assembly language
- Understand Fixed-Point Implementation using EPB_C5515 kit.

After reading **this section** you will be able to,
- Understand FFT using DSP TMS320C5515

Part List:

- PC
- Code Composer Studio
- +5v DC Power supply
- EPB_C5515
- Emulator + Emulator cable (USB A to Mini-A Cable, 14 pin FRC Flat cable)

List of Files Required:

fixed_float.c	(Program application main.c file)
lnkx.cmd	(Command file)
usbstk5515bsl.lib	(Library file)

Lab Goals:

- Compute FFT in software for a fixed length sequence.
- Compute FFT using the efficient routines provided by FFT hardware accelerator co-processor HWAFFT.
- Compare the performances of software and hardware implementations with the help of profiling.

1. Introduction:

Fast Fourier Transform (FFT)

The concept of Discrete Fourier Transform (DFT) was introduced to allow us to work with the frequency domain representation of discrete-time signals while staying in discrete (frequency) domain. Equation (1) gives the expression for the DFT calculation of a sequence x[n]..

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}, \qquad k = 0, 1, \ldots, N-1$$

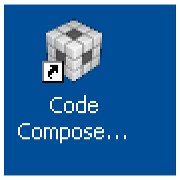
where, x[n] is the input sequence of length N, X[k] is the N-point DFT of x[n] and
$W_N = e^{-j\,2\,\pi/N}$

In general, to calculate the DFT by this algorithm, the computations (in terms of complex multiplications and additions) required vary with N as O (N^2). So, as N grows higher, the number of computations grows at a higher rate and FFT calculation operation begins to act as a bottleneck for the system. This extremely high number of computations required in the naive implementation of DFT was a major obstacle in its acceptance. To overcome this, an efficient algorithm to calculate the DFT using equation (1) was described by J. W. Cooley and J. W. Tuckey in a paper in 1965. At first, they themselves did not realize the importance of the algorithm they proposed. But, the algorithm was widely accepted and it was termed as the Fast Fourier transform[1].

This algorithm uses the ***"divide and conquer"*** technique to calculate DFTs of sequences of very large length. It splits a higher order DFT calculation into several lower order ones and then these lower order DFTs into further lower order ones until this cannot be continued. Now, it can be proved that the number of computations required to calculate a higher order DFT directly is very high as compared to splitting the DFT into lower order DFTs, calculating them and combining the results in appropriate manner to get the initial higher order DFT. This brings the order of computations required to calculate an N point DFT from O (N^2) to O (N $\log_2$(N)) which is a substantial improvement. You are going to study the implementation of FFT on the DSP in this lab session.

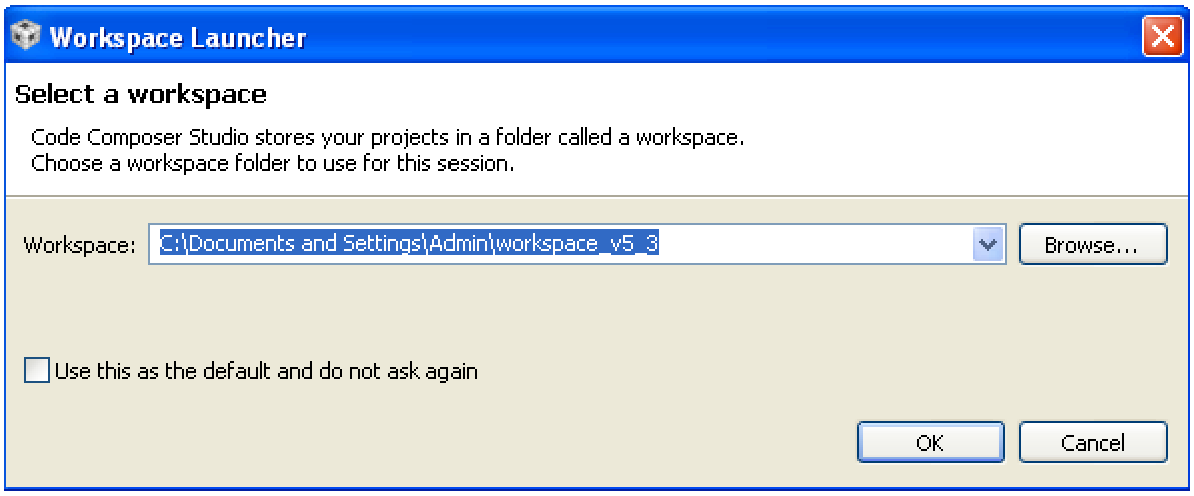
Steps for importing new project:

Open *CCS V5.3* from desktop shortcut



It will open default *CCS V5* screen.

Then it will ask for workspace path
Select path "*C:\Documents and Settings\<User Name>\workspace_v5_3*" for windows XP OS
Select path "*C:\Users\<User Name>\workspace_v5_3*" for windows7 OS



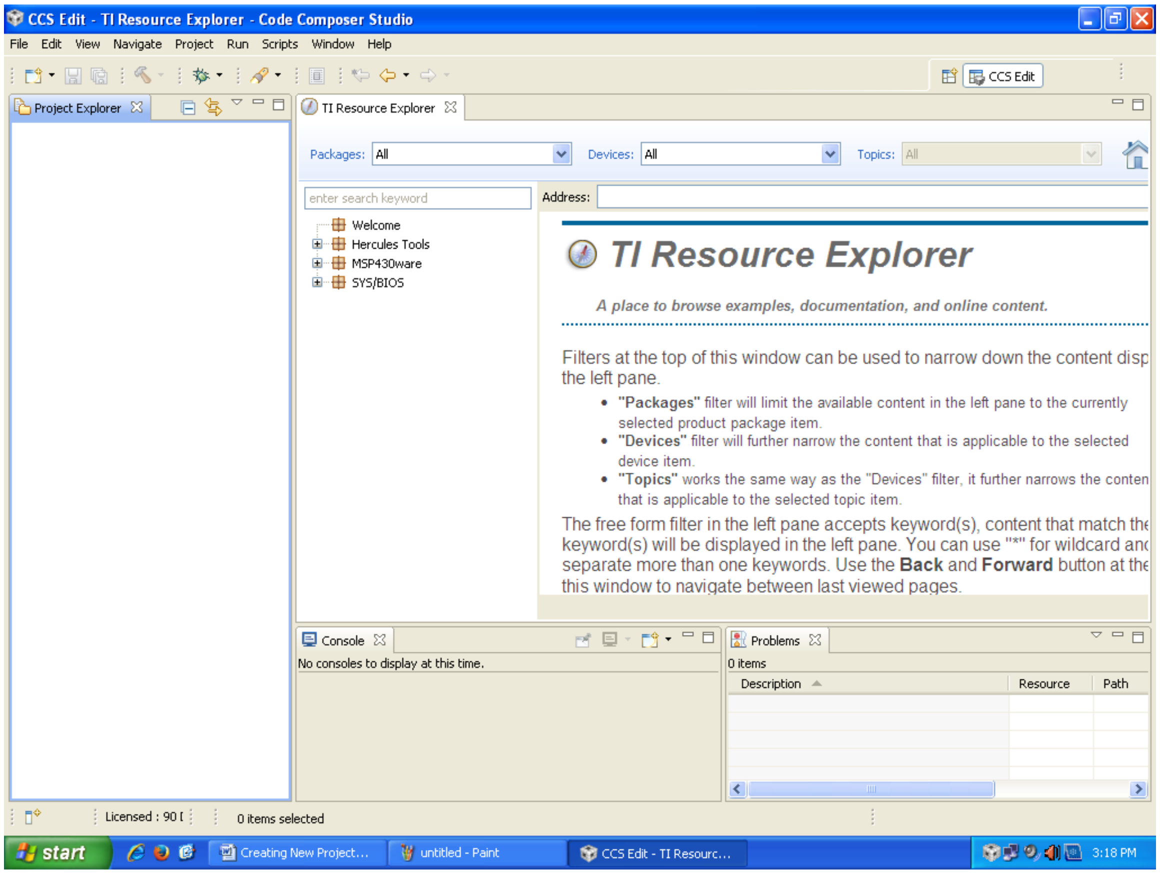
Select a workspace

Code Composer Studio stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: C:\Documents and Settings\Admin\workspace_v5_3

☐ Use this as the default and do not ask again

Then it will open Default CCS5 screen as shown below



TI Resource Explorer

A place to browse examples, documentation, and online content.

Filters at the top of this window can be used to narrow down the content disp the left pane.

- **"Packages"** filter will limit the available content in the left pane to the currently selected product package item.
- **"Devices"** filter will further narrow the content that is applicable to the selected device item.
- **"Topics"** works the same way as the "Devices" filter, it further narrows the conten that is applicable to the selected topic item.

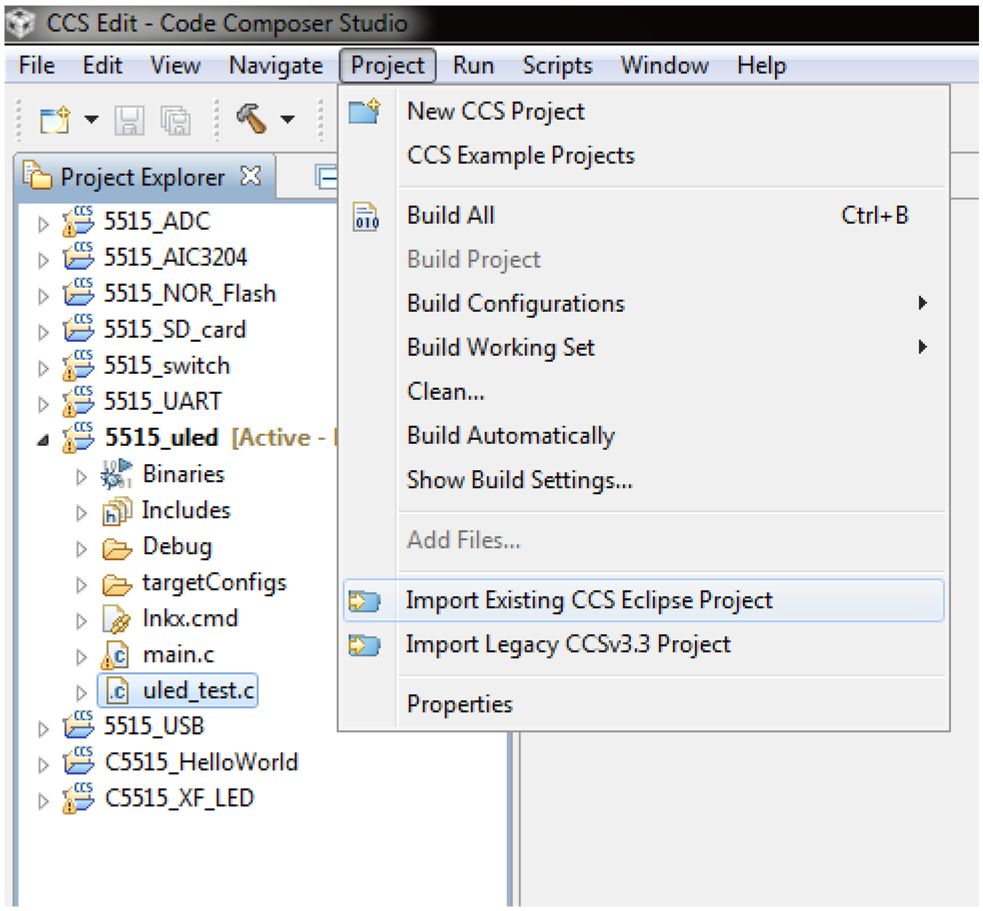
The free form filter in the left pane accepts keyword(s), content that match the keyword(s) will be displayed in the left pane. You can use "*" for wildcard and separate more than one keywords. Use the **Back** and **Forward** button at the this window to navigate between last viewed pages.

No consoles to display at this time.

0 items

0 items selected

Click "*Project -> Import Existing CCS Eclipse Project*" menu.



- Now Copy the project path and paste it to "Select search-directory" location and press enter. Or you can browse the path also by clicking browse.
- Select project "IIT_LAB6.1" and also select "Copy projects into workspace" and **Finish**
- Here project is already imported and it can be seen from "**project explorer**"
- Compile the program by "*right click-> build project*" or "*right click-> rebuild project*" as shown. It will generate "IIT_ LAB6.1.*out"* file in "*debug*" folder.
- It will generate IIT_ LAB6.1.*out* file in "*debug*" folder.

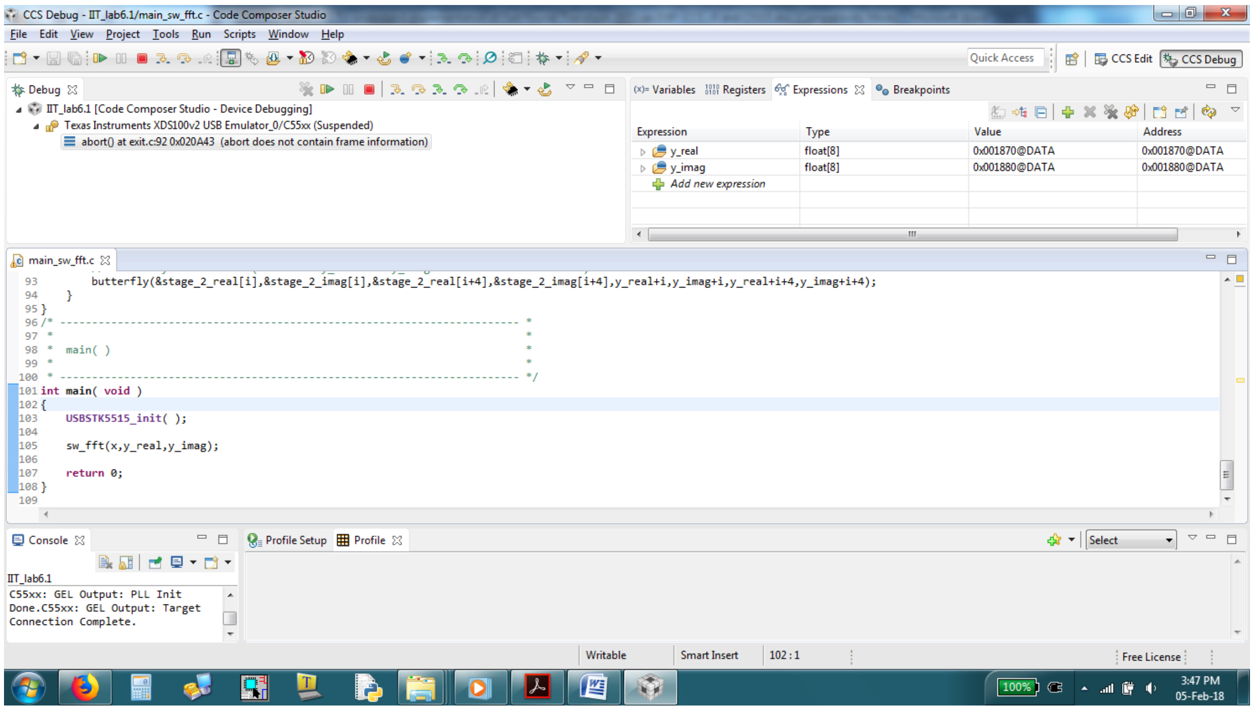
How to Run Program:

Hardware Connection:

- Power on EPB_C5515 hardware using +5V Power supply **or** USB A-to-B cable
- Connect XDS100V2 with EPB_C5515 using USB A-to-miniA cable with CPU
- Reset DSP kit EPB_C5515 by pressing RESET switch.

Steps to Run Program:

- Now to debug the program click “**debug**” icon **OR** from “**run->debug**” menu.
- It will configure/connect EPB_C5515 kit with the CCSV5 using XDS100V2 and download the program in C5515 CPU. It will be done automatically.



IIT_lab6.1 [Code Composer Studio - Device Debugging]
Texas Instruments XDS100v2 USB Emulator_0/C55xx (Suspended)
abort() at exit.c:92 0x020A43 (abort does not contain frame information)

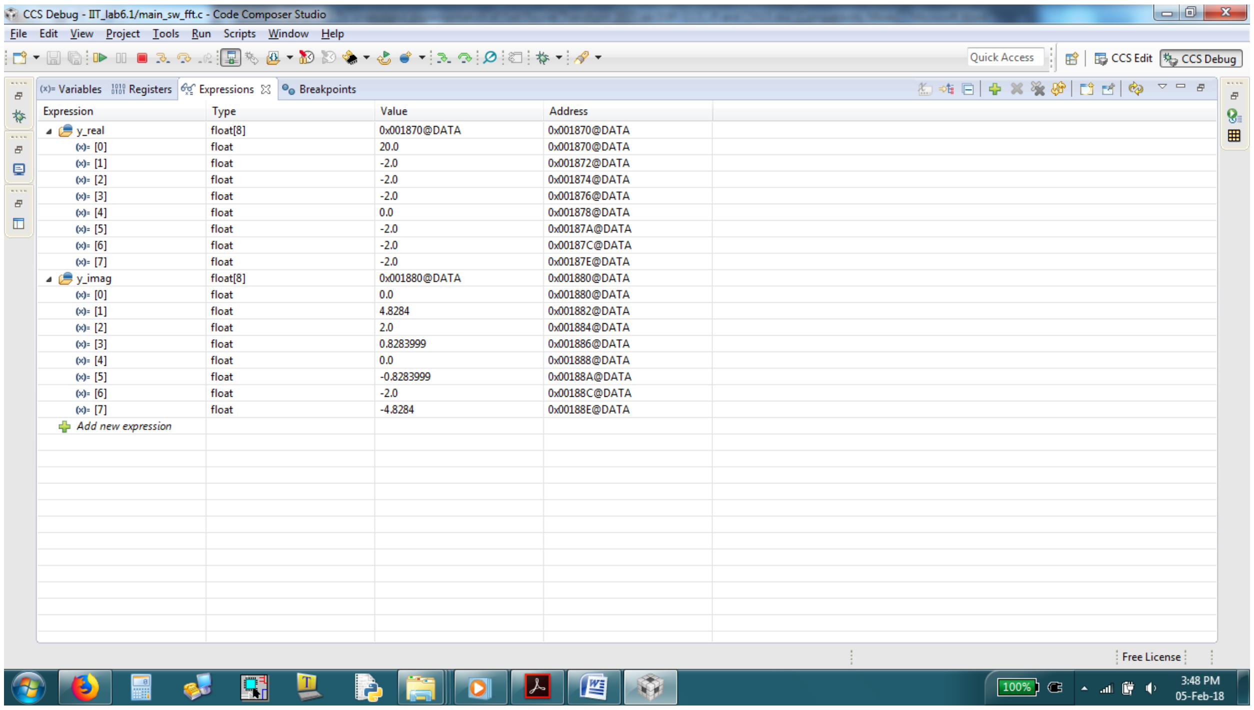
Expression	Type	Value	Address
y_real	float[8]	0x001870@DATA	0x001870@DATA
y_imag	float[8]	0x001880@DATA	0x001880@DATA
Add new expression			

```
        butterfly(&stage_2_real[i],&stage_2_imag[i],&stage_2_real[i+4],&stage_2_imag[i+4],y_real+i,y_imag+i,y_real+i+4,y_imag+i+4);
    }
}
/* ------------------------------------------------------------------------ *
 *                                                                          *
 *  main( )                                                                 *
 *                                                                          *
 * ------------------------------------------------------------------------ */
int main( void )
{
    USBSTK5515_init( );

    sw_fft(x,y_real,y_imag);

    return 0;
}
```

IIT_lab6.1

```
C55xx: GEL Output: PLL Init
Done.C55xx: GEL Output: Target
Connection Complete.
```

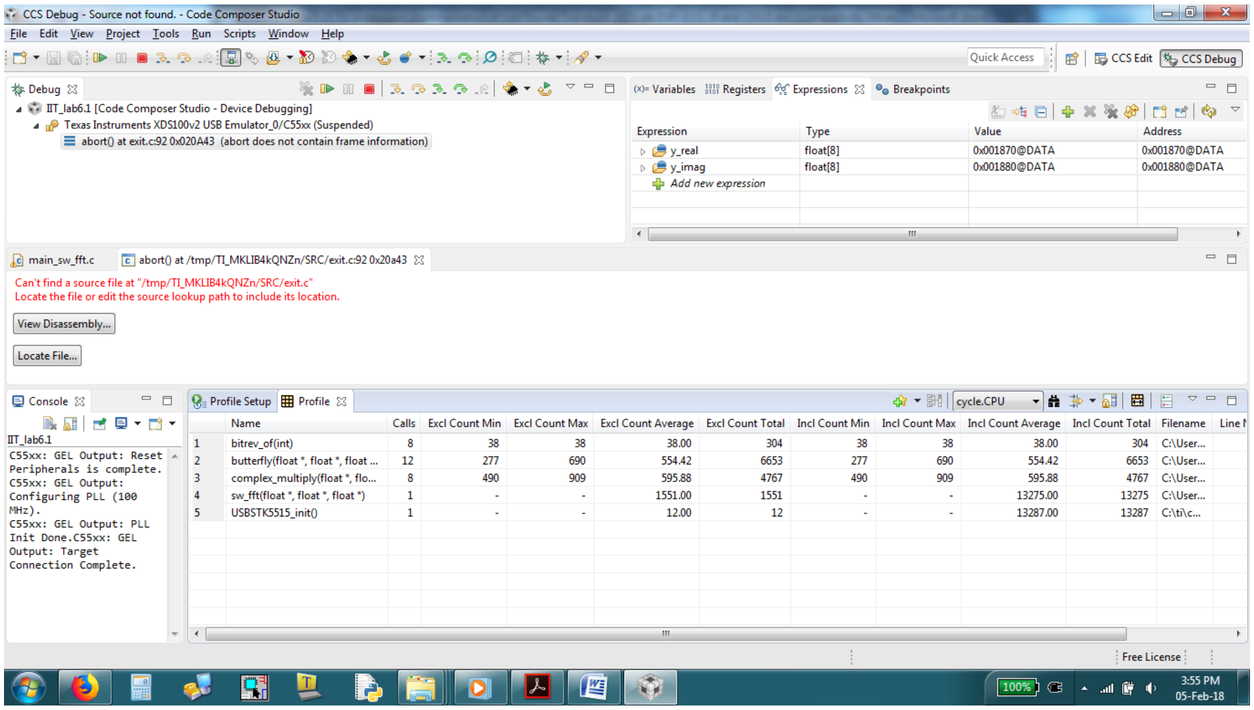
Check result in watch window for real and imaginary numbers.



CCS Debug - IIT_lab6.1/main_sw_fft.c - Code Composer Studio

Expression	Type	Value	Address
y_real	float[8]	0x001870@DATA	0x001870@DATA
[0]	float	20.0	0x001870@DATA
[1]	float	-2.0	0x001872@DATA
[2]	float	-2.0	0x001874@DATA
[3]	float	-2.0	0x001876@DATA
[4]	float	0.0	0x001878@DATA
[5]	float	-2.0	0x00187A@DATA
[6]	float	-2.0	0x00187C@DATA
[7]	float	-2.0	0x00187E@DATA
y_imag	float[8]	0x001880@DATA	0x001880@DATA
[0]	float	0.0	0x001880@DATA
[1]	float	4.8284	0x001882@DATA
[2]	float	2.0	0x001884@DATA
[3]	float	0.8283999	0x001886@DATA
[4]	float	0.0	0x001888@DATA
[5]	float	-0.8283999	0x00188A@DATA
[6]	float	-2.0	0x00188C@DATA
[7]	float	-4.8284	0x00188E@DATA
Add new expression			

Also check profiling and check output



CCS Debug - Source not found. - Code Composer Studio

IIT_lab6.1 [Code Composer Studio - Device Debugging]
Texas Instruments XDS100v2 USB Emulator_0/C55xx (Suspended)
abort() at exit.c:92 0x020A43 (abort does not contain frame information)

Expression	Type	Value	Address
y_real	float[8]	0x001870@DATA	0x001870@DATA
y_imag	float[8]	0x001880@DATA	0x001880@DATA
Add new expression			

abort() at /tmp/TI_MKLIB4kQNZn/SRC/exit.c:92 0x20a43

Can't find a source file at "/tmp/TI_MKLIB4kQNZn/SRC/exit.c"
Locate the file or edit the source lookup path to include its location.

Console – IIT_lab6.1

```
C55xx: GEL Output: Reset
Peripherals is complete.
C55xx: GEL Output:
Configuring PLL (100
MHz).
C55xx: GEL Output: PLL
Init Done.C55xx: GEL
Output: Target
Connection Complete.
```

Profile (cycle.CPU)

	Name	Calls	Excl Count Min	Excl Count Max	Excl Count Average	Excl Count Total	Incl Count Min	Incl Count Max	Incl Count Average	Incl Count Total	Filename
1	bitrev_of(int)	8	38	38	38.00	304	38	38	38.00	304	C:\User...
2	butterfly(float *, float *, float ...	12	277	690	554.42	6653	277	690	554.42	6653	C:\User...
3	complex_multiply(float *, flo...	8	490	909	595.88	4767	490	909	595.88	4767	C:\User...
4	sw_fft(float *, float *, float *)	1	-	-	1551.00	1551	-	-	13275.00	13275	C:\User...
5	USBSTK5515_init()	1	-	-	12.00	12	-	-	13287.00	13287	C:\ti\c...