EE442/EE592 Real-Time Digital Signal Processing
Laboratory 4: Introduction to TI DSP: Code Composer Studio & Simple Programs

1 Introduction

This lab steps us through the process to develop, build, debug, load, and run a simple non-real-time program using Code Composer Studio (CCS) which is the Integrated Development Environment (IDE) for the TI DSP. In addition, we will build and run the pass code (dsk_app). The instructions which follow help in understanding the method to create new projects or open existing ones. Further help on how to use the CCS can be on the course website (Related Links tab)

Please report any errors in this lab to pdeleon@nmsu.edu.

2 Using Code Composer Studio

2.1 Creating a New Project:

1) Open Code Composer Studio by double clicking on the desktop icon CCStudio 3.

2) Go to the Project menu and choose New..

3) Write the following items as required (in this example C drive is used; you may use another drive if desired):
   Project Name: lab4
   Location: C:\Documents and Settings\<USER>\Desktop\lab4
   Project type: Executable (.out)
   Target: TMS320C64XX

4) A lab4.pjt project file will be created and its contents will be displayed in the "Project window" on the left side.

2.2 Configuring a Project:

1) Go to the File menu and select New: DSP/BIOS Configuration…

2) Click on the dsk6416.cdb icon and then click OK. After few seconds, you will see a new window named Config1. Settings for the DSP can be changed in the configuration window, however, we do not need to make any changes. Go to the File menu and choose "Save As.." in order to save the configuration settings. Write the following items as required

   File name: lab4 (rename Config1 as lab4)
   Save as type: Configuration Files (*.cdb)
   Location: C:\Documents and Settings\<USER>\Desktop\lab4.

3) Add the lab4.cdb configuration file to the lab4 project by first right-clicking on lab4.pjt in project window and choosing "Add files to Project". Next, a new window to "Add files to project" will appear. By default, the folder in "Look in" will be "lab4" and "Files of type" will be C source Files (*.c, *.ccc). Choose "Configuration File (*.cdb)" from "Files of type" menu using the drop down menu. Click on "lab4" and then click "Open".

4) Once lab4.cdb is added to the project, the configuration manager will create and load the following files:

   lab4cfg.s62 Assembly file
   lab4cfg_c.c C file

5) Click on the "+" sign to the left of "Generated Files" to view the above files. You can also click on the "+" signs to view the different files that have been added to the project. Four other files are also created but they are not added to your project by configuration manager. These files are named:

lab4cfg.h	Header file for C
lab4cfg.h62	Header file for assembly
lab4cfg.cmd	Linker command file
lab4.ccf	

and can be found in C:\Documents and Settings\<USER>\Desktop\lab4.

6) Add the lab4cfg.cmd file to your project by repeating step 3.

2.3 Adding Source Files to a Project:

1) Download lab4.c file from the course website (TI 6416 Code tab)

   and save it in C:\Documents and Settings\<USER>\Desktop\lab4.

2) Add the lab4.c source file to your project following the previous instructions for adding files to a project.

3) Click on the "+" sign to the left of "Source" to view lab4.c. Double click the lab4.c icon to open the file.

4) Simply edit "lab4.c" for programs 1—3 of this lab (see below).

2.4 Building and Running the Project:

1) Go to the Project menu and select Build to build (compile) the project. Any errors or warnings will be displayed in the bottom part of the window. The .out files (object codes) are placed in the Debug folder. A .out file will only be generated if there are no build or compile errors.

2) Before you can download the object file to the DSP, you must connect to the DSP board. Select the Debug menu and click Connect.

3) If no errors are present and the DSP board is connected, go to "File" menu and select "Load Program". Load the file "c:\Documents and Settings\<USER>\Desktop\lab4\lab4.out".

4) A disassembly window will be opened containing the object code. You can minimize or close this window.

5) Press "F5" to run through your code or you may wish to step through code and monitor variables using the instructions below.

2.5 Using Breakpoints and the Watch Window:

1) To watch variables and monitor their values, go to the View menu and select Watch. The Watch window will appear at the bottom right part of your main window.

2) Click on the "Watch1" tab. In the table, type variable names below the Name heading and hit Enter. Values corresponding to that variable will be updated and displayed as code is executed. Variables can alternately be added to the Watch window by selecting the variable name in the lab4.c window, right-clicking, and choosing "Add to Watch Window". Initially variable values may be garbage. However, once you start stepping through code, updated values will be displayed in the watch window.

3) Once you have stepped through the program, it must be reloaded in order to be re-run.

4) If you right-click in the lab4.c window and choose "Mixed mode" then the C64xx machine language corresponding to your C code will be displayed. You can revert to the original format by right-clicking in the lab4.c window and selecting "Source Mode".

5) If you want to place a software breakpoint on a line of code, double-click just to the left of the line of code in your source file. Double-clicking again will remove the breakpoint.

6) Finally, you may run your code by hitting F5 or step through your code one line at a time by hitting F8.

3 Simple Programs

The following simple programs will add, multiply and divide two short values on our fixed-point processor.

Program 1: Addition of two shorts

```
/* Addition of two short numbers */
#include <stdio.h>
#include <math.h>
#include <std.h>

short a, b, c;
int    c2;

int main (void)
{
       a = 30000;
       b = 10000;

       c = a + b;    /* Desired Value = 40000 */
                     /* Actual Value = -25536 = 40000 - 32767 - 32768 - 1 */
       c2 = a + b;   /* c2 = 40000 */

       return 0;
}
```

Program 2a: Multiplication of two shorts

```
/* Multiplication of two short numbers */
#include <stdio.h>
#include <math.h>
#include <std.h>

short a, b, c;
int    c2;

int main (void)
{
       a = 3276;     /* a = 0.1 */
       b = 3276;     /* b = 0.1 */

       c = a * b;    /* c = -15728 */

       c2 = a * b;   /* c2 = 10732176 */
       c2 = c2>>15;  /* c2 = 327 */
       c = (short)c2;       /* c = 327 */

       return 0;
}
```

Program 2b: Multiplication of two shorts with rounding

```
/* Multiplication of two short numbers */
#include <stdio.h>
#include <math.h>
#include <std.h>

short a, b, c;
int    c2;

int main (void)
{
       a = 3276;     /* a = 0.1 */
       b = 3276;     /* b = 0.1 */

       c = a * b;    /* c = -15728 */

       c2 = a * b;          /* c2 = 10732176 */
       c2 = cround(c2);     /* convergently round */
       c2 = c2>>15;         /* c2 = 328 */
       c = (short)c2;       /* c = 328 */

       return 0;
}

int cround(int x)
{
       if (x & 0x3fff)                  // If lower 14 bits are not all 0
              x += 0x4000;              //   normal rounding
       else                             // else lower 15 bits are 0x4000
              x += (x>>1) & 0x4000;     //   convergent rounding: add half of lsb of
                                        //   upper half

       return x & 0xffff8000;           // truncate lower 14 bits
}
```

Program 3: Division of two shorts

```
/* Division of two short numbers */

#include <stdio.h>
#include <math.h>
#include <std.h>

short a, c;
int    c2;

int main (void)
{
       a = 2;
       c = 80000 / a;      /* c = -25536 */

       c2 = 80000 / a;     /* c2 = 40000 */

       return 0;
}
```

4 Running the TI Passcode (DSK_APP)

1) Open Code Composer Studio.

2) Download dsk_app.zip from the course website (TI 6416 Code tab). Unpack and save it in C:\Documents and Settings\<USER>\Desktop\dsk_app.

3) Go to the Project menu and choose Open..

4) From within C:\Documents and Settings\<USER>\Desktop\dsk_app, open dsk_app.pjt.

5) Click on + next to Source Files to see the .c source files and click on + next to Include to see the .h files.

6) Build the project, connect to the DSP board, and load the .out file.

7) Run the code by hitting F5.

8) The "line out" and "line in" cables can be found hanging from the back of the PC. Connect the line in cable to output of the sound card or any audio source. The line out cable can be connected to the speakers.

9) Play any audio file and check whether the audio is passed to the speakers via the processor.

5 Altering the Sampling Rate

1) Sample rate definitions for the C6416DSK can be found in the util.h file.

2) On line 313 of dsk_app.c, the following can be found:

```
DSKAPP_AIC23_48KHZ,  /* 8 DSK6416_AIC23_SAMPLERATE Sample rate control  */
```

3) Simply modify the above line with the desired sample rate definition.