**EECS 363: Digital Filtering**

**Lab 5 - 2/25/2017**

**Karan Shah**

**Code:**

Main File:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* \*/

/\* FILENAME \*/

/\* main.c \*/

/\* \*/

/\* DESCRIPTION \*/

/\* TMS320C5505 USB Stick. Application 1. Getting started. \*/

/\* Take microphone input and send to headphones. \*/

/\* \*/

/\* REVISION \*/

/\* Revision: 1.00 \*/

/\* Author : Richard Sikora \*/

/\*---------------------------------------------------------------------------\*/

/\* \*/

/\* HISTORY \*/

/\* Revision: 1.00 \*/

/\* 5th March 2010. Created by Richard Sikora from TMS320C5510 DSK code. \*/

/\* \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

\* Copyright (C) 2010 Texas Instruments Incorporated - http://www.ti.com/

\*

\*

\* Redistribution and use in source and binary forms, with or without

\* modification, are permitted provided that the following conditions

\* are met:

\*

\* Redistributions of source code must retain the above copyright

\* notice, this list of conditions and the following disclaimer.

\*

\* Redistributions in binary form must reproduce the above copyright

\* notice, this list of conditions and the following disclaimer in the

\* documentation and/or other materials provided with the

\* distribution.

\*

\* Neither the name of Texas Instruments Incorporated nor the names of

\* its contributors may be used to endorse or promote products derived

\* from this software without specific prior written permission.

\*

\* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS

\* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT

\* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR

\* A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT

\* OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,

\* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT

\* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,

\* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY

\* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT

\* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE

\* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

\*

\*/

**#include** "stdio.h"

**#include** "csl\_i2s.h"

**#include** "csl\_intc.h"

**#include** "usbstk5505.h"

**#include** "usbstk5505\_led.h" // added for led control

**#include** "aic3204.h"

//#include "usbstk5505\_gpio.h"//added, compiles without

//#include "usbstk5505\_i2c.h"//added, compiles without

**#include** "PLL.h"

**#include** "stereo.h"

**#include** "stereo.c"

**#include** "Dsplib.h"

**#include** "coefficients.h" //header for coefficients with length 41

**#include** "coefficients1.h" //header for coefficients with length 16

Int16 left\_input; //do these interfere with declarations in aic3204.c?

Int16 right\_input;

Int16 left\_output;

Int16 right\_output;

Int16 mono\_input;

Int16 x;

Int16 y;

**static** DATA \*dbptr = db;

//static DATA \*dbptr1 = db1;

**#define** SAMPLES\_PER\_SECOND 1500000

**unsigned** **long** i = 0;

**unsigned** **long** xstart = 0;

**unsigned** **long** j = 0; //added for led control

**short** toggle = 0; //added for led control

/\* ------------------------------------------------------------------------ \*

\* \*

\* main( ) \*

\* \*

\* ------------------------------------------------------------------------ \*/

**interrupt** **void** **codec\_read\_isr**(**void**);

**void** **main**( **void** )

{

// static DATA \*dbptr = db;

/\* Initialize BSL \*/

USBSTK5505\_init( );

**USBSTK5505\_LED\_init**(); //added for LED control

/\* Initialize PLL \*/

**pll\_frequency\_setup**(100);

/\* Initialise hardware interface and I2C for code \*/

aic3204\_hardware\_init();

/\* Initialise the AIC3204 codec \*/

aic3204\_init();

**printf**("\n\nRunning Getting Started Project\n");

**printf**( "<-> Audio Loopback from Stereo IN --> to HP/Lineout\n" );

/\* Setup sampling frequency and 30dB gain for microphone \*/

set\_sampling\_frequency\_and\_gain(SAMPLES\_PER\_SECOND, 0);// was 30 dB; I changed.

**asm**(" bclr XF");

**IRQ\_plug**(RCV2\_EVENT, &codec\_read\_isr);

**IRQ\_enable**(RCV2\_EVENT);

**IRQ\_globalEnable**();

**while**(1)

{

// begin segment for led control

**if** (j++ == SAMPLES\_PER\_SECOND)

{

toggle = 1-toggle;

**asm**(" SSBX INTM");

**if** (toggle)

**USBSTK5505\_LED\_on**(0);

**else**

**USBSTK5505\_LED\_off**(0);

j = 0;

}

// end segment for led control

**asm**(" RSBX INTM");

}

/\* Disable I2S and put codec into reset \*/

aic3204\_disable();

**printf**( "\n\*\*\*Program has Terminated\*\*\*\n" );

SW\_BREAKPOINT;

}

/\* ------------------------------------------------------------------------ \*

\* \*

\* End of main.c \*

\* \*

\* ------------------------------------------------------------------------ \*/

**interrupt** **void** **codec\_read\_isr**(**void**)

{

**if** (I2S2\_IR & RcvR)

{

left\_input = I2S2\_W0\_MSW\_R; // Read Most Significant Word of first channel

right\_input = I2S2\_W1\_MSW\_R; // Read Most Significant Word of second channel

x = left\_input;

**fir**(&x,h,&y,dbptr,NX,NH);

//fir(&x,h1,&y,dbptr1,NX1,NH1); // averaging the 16 most recent inputs

left\_output = y;

right\_output = right\_input; // Directly connect inputs to outputs.

I2S2\_W0\_MSW\_W = left\_output; // Left output

I2S2\_W1\_MSW\_W = right\_output; // Right output

}

**return**;

}

Header File for length 41:

/\*

\* coefficients.h

\*

\* Created on: Feb 22, 2017

\* Author: Karan

\*/

**#ifndef** COEFFICIENTS\_H\_

**#define** COEFFICIENTS\_H\_

**#define** NX 1

**#define** NH 41

/\* \*/

/\* The type DATA is equivalent to \*/

/\* short and Int16. \*/

/\* \*/

**#pragma** DATA\_SECTION(db,".dbuffer")

DATA db[NH+2];

**#pragma** DATA\_SECTION(h,".coeffs")

DATA h[NH] =

{ /\* filter impulse response \*/

-293,

-215,

2,

328,

494,

340,

20,

-115,

90,

320,

27,

-944,

-1930,

-1812,

-29,

2621,

4272,

3358,

21,

-3721,

27423,

-3721,

21,

3358,

4272,

2621,

-29,

-1812,

-1930,

-944,

27,

320,

90,

-115,

20,

340,

494,

328,

2,

-215,

-293,

};

**#endif** /\* COEFFICIENTS\_H\_ \*/

Header File for length 16:

/\*

\* coefficeints1.h

\*

\* Created on: Feb 24, 2017

\* Author: Karan

\*/

**#ifndef** COEFFICIENTS1\_H\_

**#define** COEFFICIENTS1\_H\_

**#define** NX1 1

**#define** NH1 16

**#define** INPUT\_LENGTH 16

/\* \*/

/\* The type DATA is equivalent to \*/

/\* short and Int16. \*/

/\* \*/

**#pragma** DATA\_SECTION(db1,".dbuffer")

DATA db1[NH1+2];

**#pragma** DATA\_SECTION(h1,".coeffs")

DATA h1[NH1] =

{ /\* filter impulse response \*/

-293,

-215,

2,

328,

494,

340,

20,

-115,

90,

320,

27,

-944,

-1930,

-1812,

-29,

2621,

};

**#endif** /\* COEFFICIENTS1\_H\_ \*/