Lab 5   Winter 2017   EECS 363 Digital Filtering   due February 27, 2016

We now want to add an FIR filter to the ISR version of the audio loopback program, i.e. Lab 3. The filter will be mono; it will filter one channel and allow the other channel to pass without change as the present loopback program does. Thus we can, e.g., simultaneously observe both filtered and unfiltered versions of a mono input.

We have discussed in class the issues of coefficient representation, word length, circular addressing or buffering, and multiply/accumulate.

The method will be to combine the loopback program with the DSPLIB package, calling the library assembly program `fir.asm`, as we did in the preceding exercise Lab 4. Use the `lnkxA.cmd` file you created in Lab 4, and the various headers and libraries used in Labs 3 and 4. As in Lab 4 you must `#include <Dsplib.h>` in the `main` program and the appropriate search path to its `include` folder, and you must also `#include` your `.h` file defining the filter coefficients. These two headers should be last in your includes in `main`.

Write the ISR to filter only one channel by calling `fir.asm`. Here is an example of how this can be done with a "block length" (corresponding to `NX` of Lab 4) = 1, i.e. `fir` is given one input *x* value on each call, and returns one output *y* value on that call:

Add the declarations

```
Int16 x;
Int16 y;
static DATA *dbptr = db;
```

The variable `*dbptr` must be declared static so that it is unchanged on exiting and reentering the ISR. Between the input and output segments of the ISR insert

```
     x = left_input;
     fir(&x,h,&y,dbptr,1,NH);
     left_output = y;
     right_output = right_input;
```

In this lab we will use the $\{h\}$ coefficients for the notch filter that were determined with the Parks-McClellan method (`pmc`), but first you should set up a test in which the filter simply averages the most recent 16 inputs. Since f_s = 48 kHz, you should have zeroes at 3 kHz, 6 kHz, etc.. Input sinusoids and observe outputs on a scope. <u>That should be part of the demo you give me</u>. This means that you will have two `.h` files representing coefficients.

Now use the filter that you determined in Matlab. Write a new header file to specify those coefficients (which header the program uses is determined by its `include` statements), but in Q15 representation, as should have been provided by the recent Matlab exercise.

The only block length we shall use here is `NX` = 1; I should discuss that issue in class.

As usual, give me a demo and a printout of your source (`main` and the coefficients header file) on or before Feb. 27.

2/9/2017