DSPDemo

By
Moe Wheatley
MoeTronix

www.moetronix.com

Sept. 10, 2004

Table of Contents

1 Introduction

1.1 The Idea

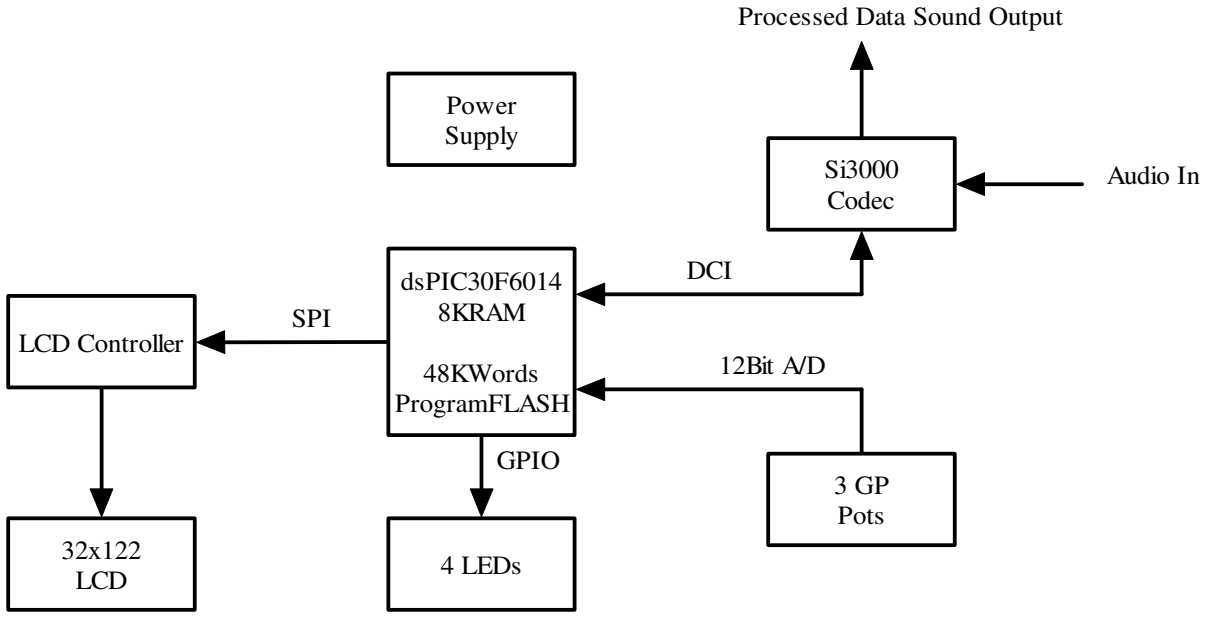
This project was an attempt to evaluate the performance of some of the basic DSP functions of the dsPIC. Several signal sources were created and then fed into an assortment of DSP filters and functions. Output can be analyzed using graphical displays of time or frequency domain data or sent to a D/A for external analysis.

It was desired to evaluate the MPLAB C30 compiler and the DSP library and discover how well it performed so all but a couple of small DSP routines were written in 'C'.

1.2 Hardware

The hardware chosen for this project was the dsPICDEM 1.1 evaluation board from Microchip. It provides a standalone platform for the software with no hardware modifications or additions.

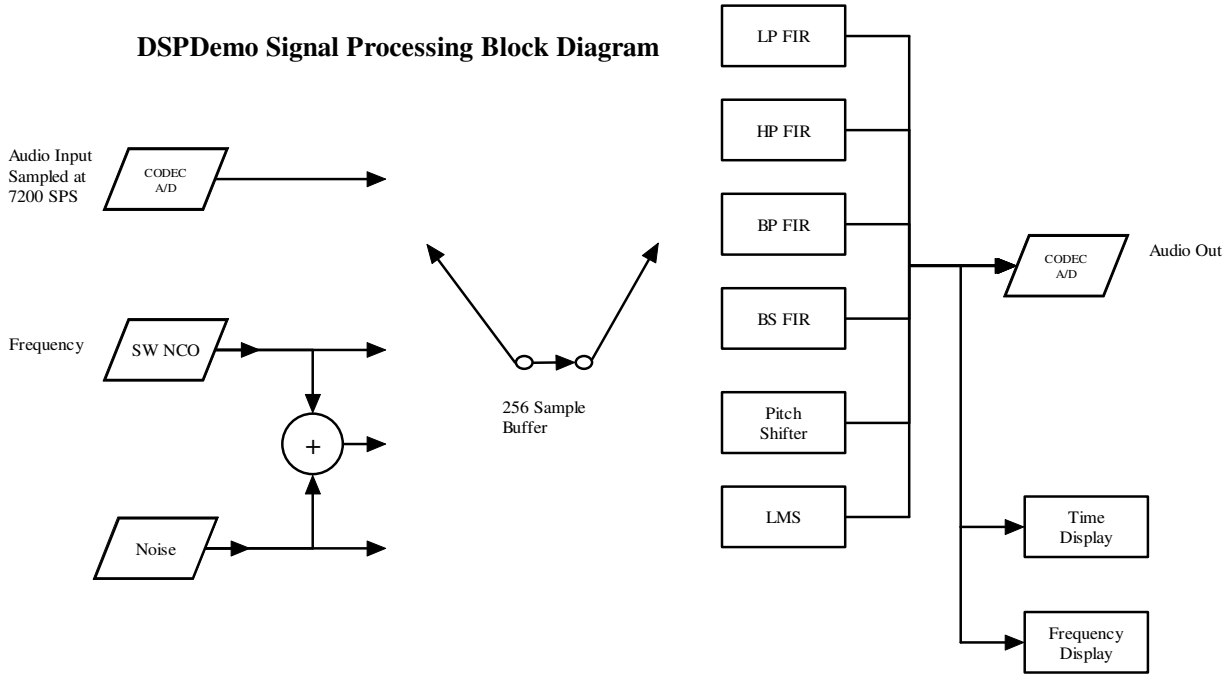
1.2.1 Block Diagram



Input signals can be input to the CODEC audio MIC input jack J16. The processed output is always on the CODEC audio SPKR OUT jack J17.
The three available pots, RP1 and 3 are used to set the input/output level as well as set the generator source frequency. They are hooked to the dsPIC 12bit A/D converter and read periodically and the values used to adjust the parameters.
The LCD and 4 key switches SW1-4 are used to implement a simple menu and display system.
Four LEDs, LED1-4 provide a way to measure the time required to generate input signals, process data, and write to the output device.
DsPIC resources used are a couple timers, the 12 bit A/D converter, the DCI module, the SPI module, internal EEROM, and some I/O ports.

1.3 Software

The software is written primarily in C using the Microchip MPLAB C30 Compiler. Debug and program development was accomplished using the MPLAB IDE and the ICD-2 programming/debug module along with an oscilloscope.



1.3.1 Basic Modules

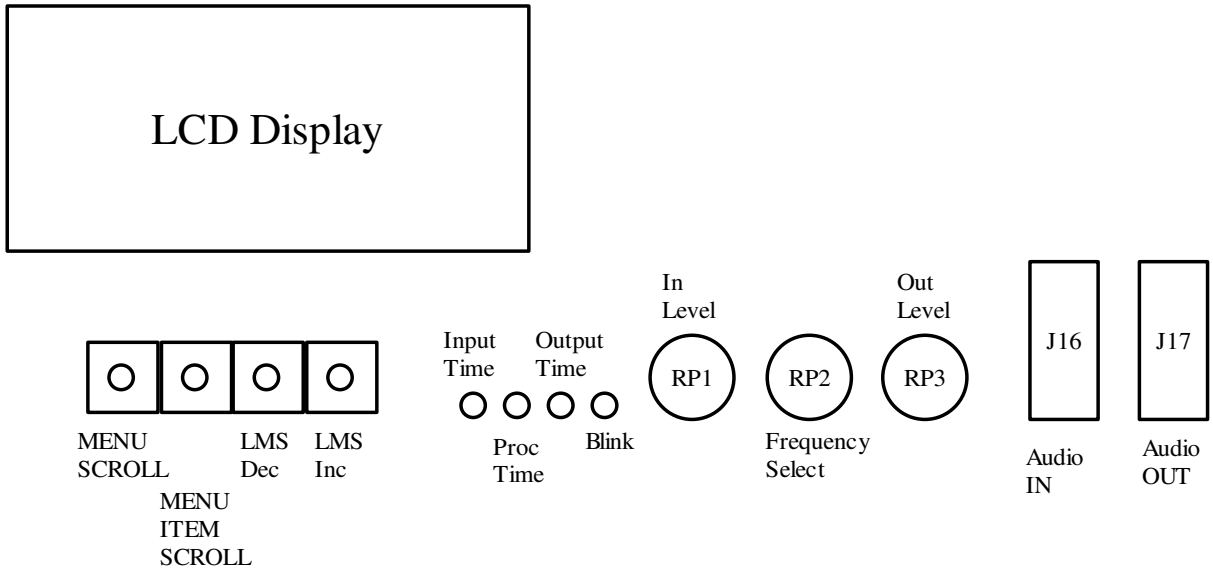
There are nine software modules that are linked together. The following is a list of the modules and their basic functions:

Main.c	Program Entry and main execution loop
Timers.c	Timer setup, IRQ service, and EEROM utilities
AtoD.c	12 bit A/D setup and utilities
Codec.c	Si3000 Codec setup and IRQ service routine
LCD.c	LCD controller setup and interface routines
Menu.c	Menu system implementation
Generators.c	Implements various sine/cosine and noise generators
ProcDsp.c	DSP section which processes the input data
FftFunct.c	Performs FFT and log-magnitude function of output
AsmUtils.s	A couple of assembler code helper functions for the DSP tasks

The following include files are also used:

Timers.h	Module Includes
AtoD.h	Module Includes
Codec.h	Module Includes
LCD.h	Module Includes
Menu.h	Module Includes
Generators.h	Module Includes
ProcDsp.h	Module Includes
FftFunct.h	Module Includes
Common.h	common program definitions
Tables.h	DSP filter tables
DSPDemo.*	Various project files for MPLAB

2 User Operation Guide



2.1 Program Loading

First the MPLAB IDE, the MPLAB C30 Compiler, and ICD-2 software must be installed and the ICD-2 attached to the PC and to the dsPICDEM 1.1 evaluation board as described in the documentation for the tool sets.

Create a folder and place all the DSPDemo project files into it.

From within the MPLAB IDE, load the project file DSPDemo.mcp into the IDE workspace. If the C30 Compiler is setup then the project can be compiled and the demo board programmed with the compiled code in either debug mode or just use the programming mode.

2.2 Initial Setup

Once the program is loaded and running, a splash screen should appear followed by the main menu screen displaying the input signal source.

If headphones or a speaker is connected to the SPKR jack of the demo board. Adjust RP3 for the output volume.

A microphone or other signal source can be plugged into J16. Adjust RP1 to adjust the input level.

2.3 Menus

There are seven menus that can be reached by pressing the Menu Scroll button(SW1)

2.3.1 Input Source

The Input Source menu allows the user to select various input sources using the menu Item scroll key(SW2).

```
        INPUT SOURCE
xxxxx = CODEC, SINE, NOISE, SINE+NOISE
FREQUENCY = 0 to 3500Hz (changed with RP2)
IN SCALE = 0 to 32768 (input scale value adjusted with RP1)
```

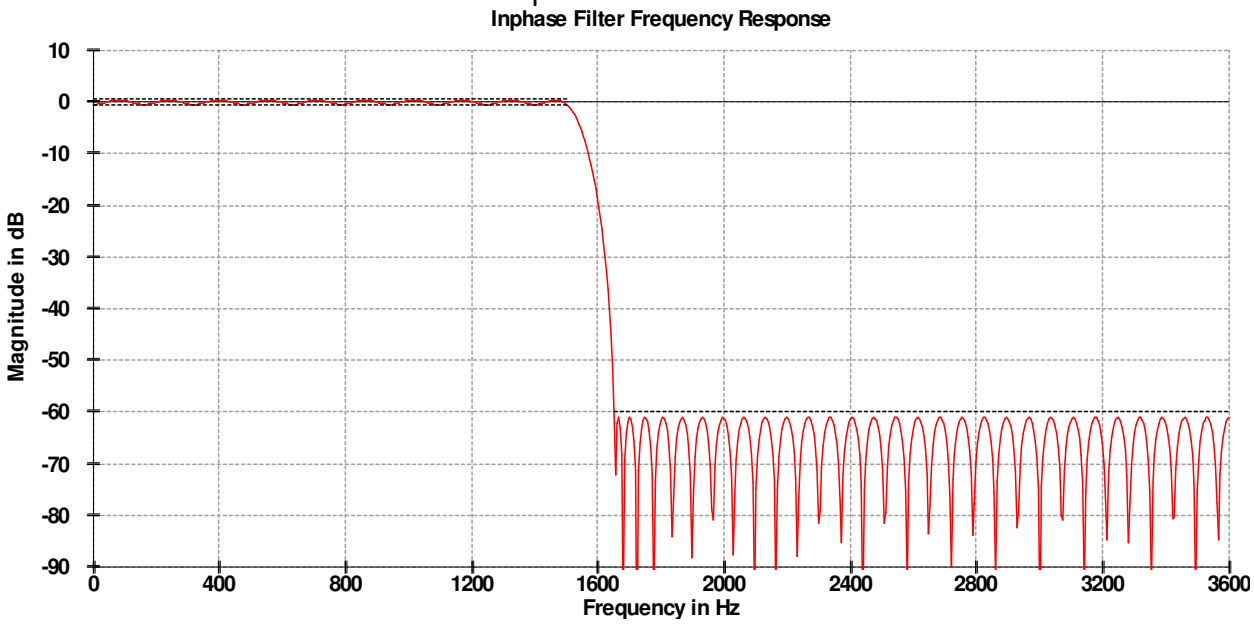
2.3.2 DSP Processing

This menu selects the type of digital signal processing to apply to the Input using the Menu Item Scroll key(SW2).

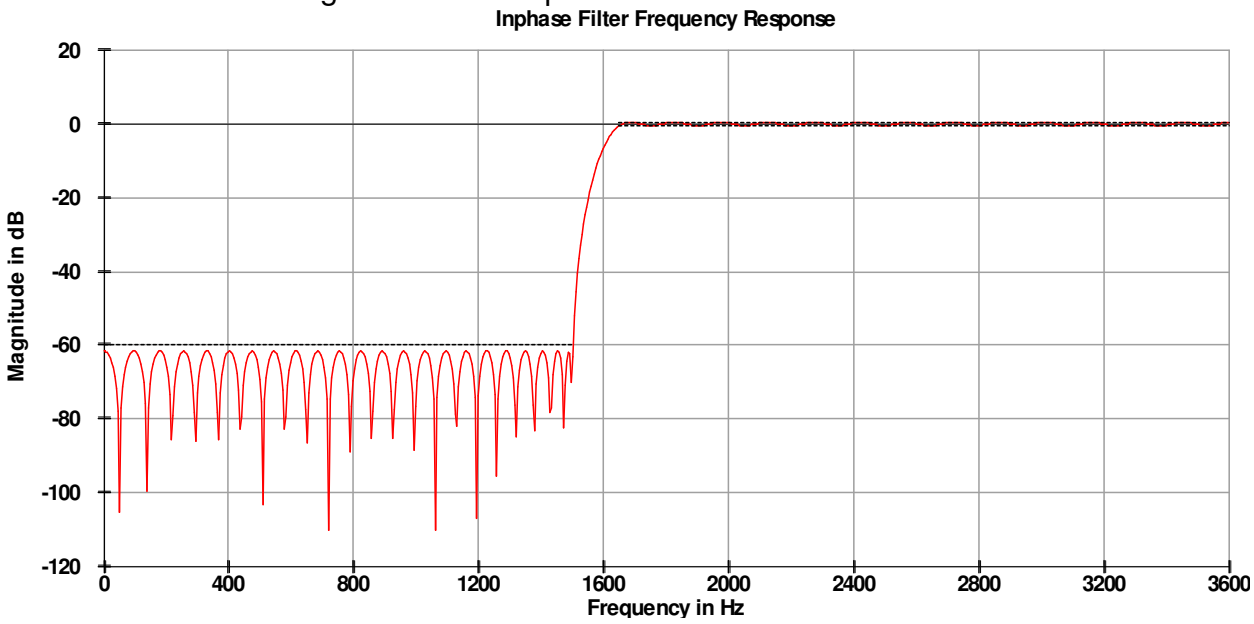
```
        DSP PROCESSING
NONE, LP FIR, HP FIR, BP FIR, BS FIR, FSHIFT, LMS
```

NONE == No processing is performed so is straight through path.

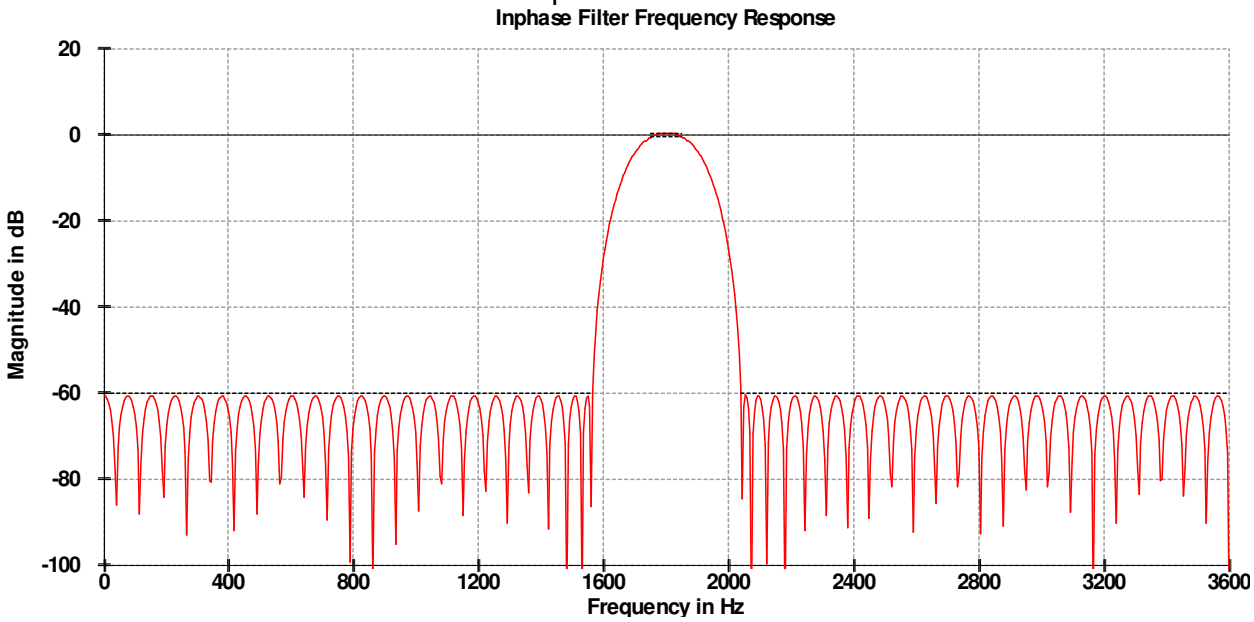
LP FIR == 1500Hz Low Pass 101 Tap FIR filter



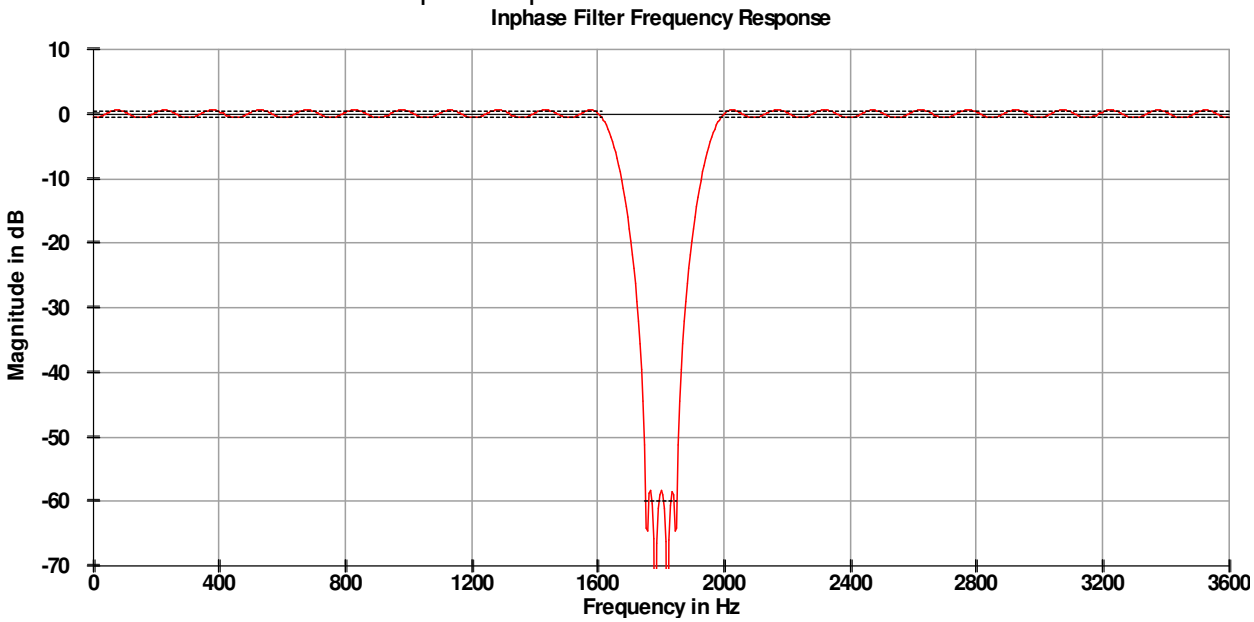
HP FIR == 1500Hz High Pass 101 Tap FIR filter



BP FIR == 1800Hz Band Pass 100 Tap FIR filter

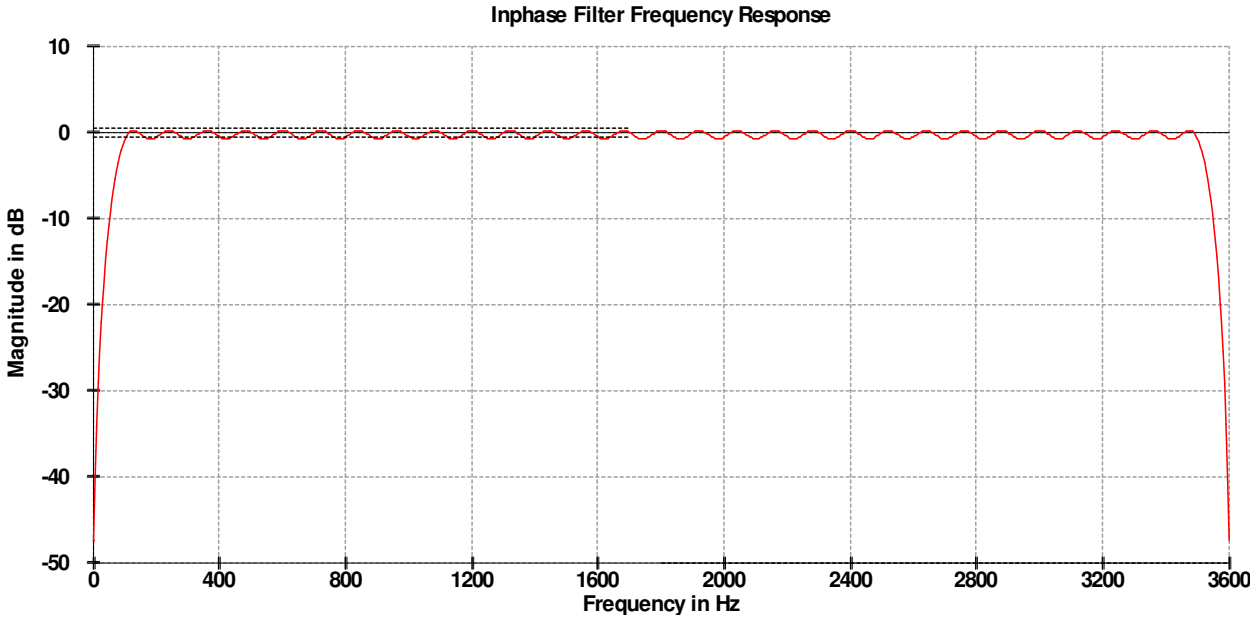


BS FIR == 1800Hz Band Stop 99 Tap FIR filter

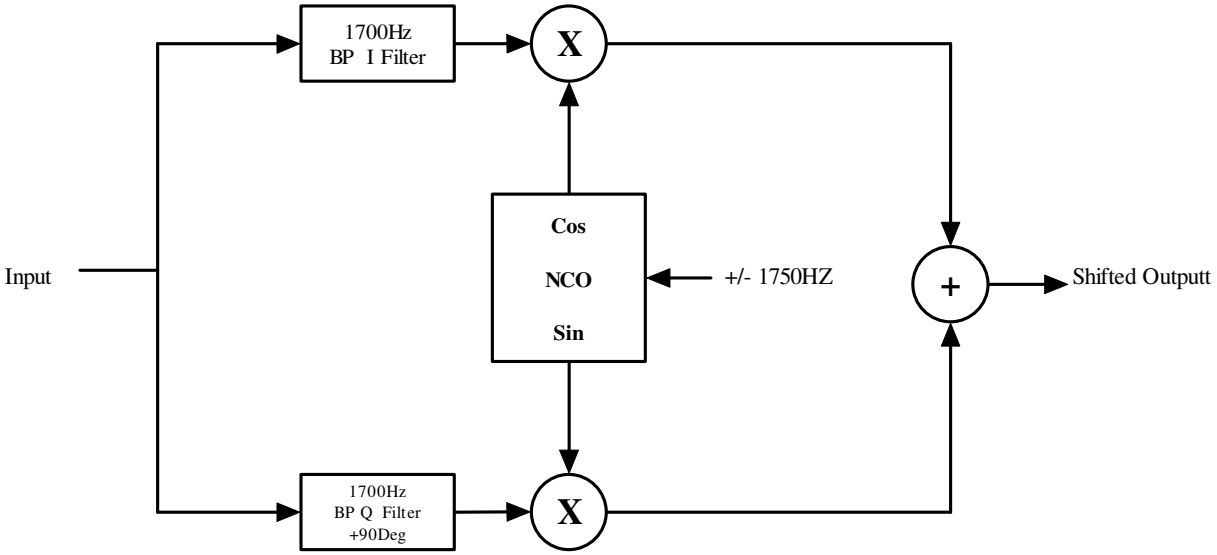


FSHIFT == Complex mixer, +/- 1750Hz NCO, with two 126Tap Hilbert Band Pass filters
Implement simple pitch shifter.

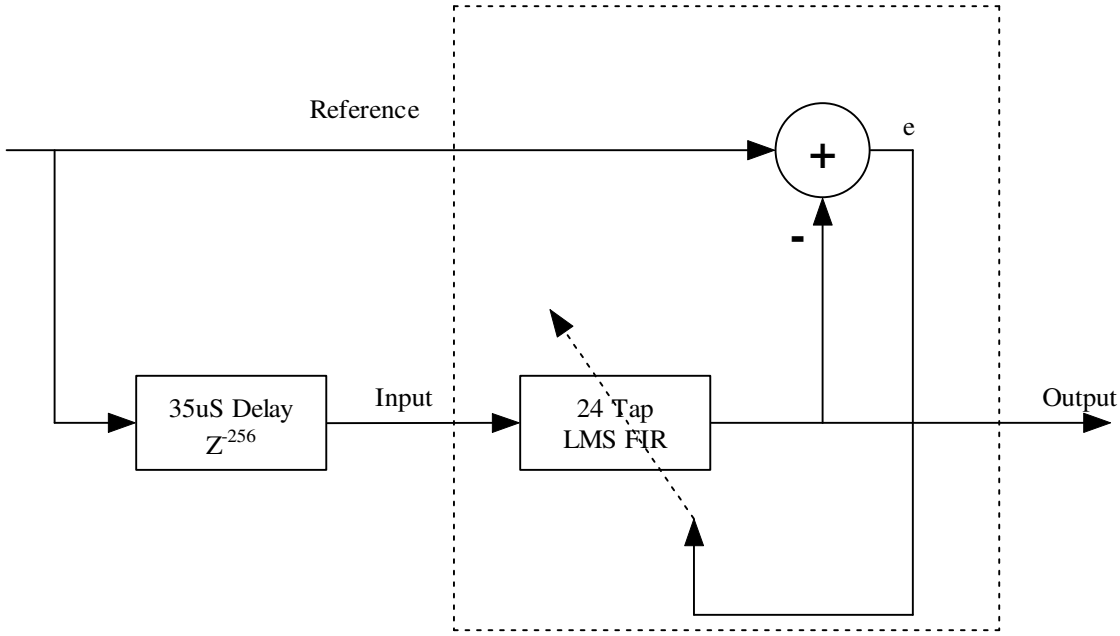
Below is the response of the Hilbert I/Q band-pass filters. Two are used that have identical magnitude responses but whose outputs are 90 degrees out of phase with each other.



Below is a block diagram of the basic pitch shifting algorithm. It is essentially a single sideband modulator operating at audio frequencies.



LMS == LMS noise reduction filter



The LMS filter is implemented with the DSP library LMS function and a single delay line. The basic idea is to enhance any correlated signals and minimize uncorrelated (noise) signals by minimizing the error difference signal which in this case is the difference between the input signal and a delayed version of the input.

2.3.3 Codec Adjust

The Codec Adjust menu displays the Codec output value that is adjusted with pot RP3.

```
     CODEC ADJUST

OUT VOLUME = 0 to 7F
```

The volume value is the Si3000 Codec DAC volume control register and has a range of +12 to -34.5dB. 0dB corresponds to the value of 0x5A.

2.3.4 Time Display

This menu shows the output signal amplitude versus time. It is just the first 123 samples of the data buffer plotted where zero is in the middle and the top and bottom of the screen are +32767 and -32768 respectively.

2.3.5 Frequency Display

This menu displays power versus frequency and is the output of a 256 point FFT of the output signal. The vertical scale is 1 to 32 points corresponding to the log2() of the power output spectrum. This allows most of the 16 bit 96dB dynamic range to be shown on a very limited resolution display.

3 Summary

This project served its purpose in gaining a better understanding of the dsPIC and its DSP library development tool set.

3.1 Processor Resources

This demo project is functional and did not consume all the dsPIC resources in terms of CPU power or program memory. It does use all the available RAM.

The program used about 15Kbytes of program memory and a just about all the 8Kbytes of data RAM.

The worst case CPU load in the main non-interrupt processing loop was about 51% at the slowest dsPICDEM 1.1 board clock rate of 7.3728MHz and an x4 PLL.
The interrupt background tasks add about another 5% to the CPU load.

3.2 Toolset Issues

The MPLAB IDE and C30 compiler operated with few problems. The ICD-2 debugger worked as long as one didn't forget to plug it into the demo board before programming. Usually a program restart and power sequence of the target board and ICD-2 would be required of that occurred.

Most of the debugging was done with a scope and not the IDE. Single stepping was problematic especially when interrupts were flailing away.

3.3 DSP Performance Results

3.3.1 Issues

The FIR filter functions worked as expected using the DSP Library functions.
The Sine generation function creates some artifacts due to the limited table lookup size resolution. Perhaps a longer table or linear interpolation would reduce the noise.
The LMS filter is very sub-optimum and could be greatly improved by adding a leaky coefficient algorithm and also by optimizing all the parameters.
The FFT consumes most of the RAM and a 512 point FFT is probably at the limit of being practical on the 8K RAM processor. Any additional buffering makes the 256 point a more likely limit.

3.3.2 Conclusions

The dsPIC has a very respectable DSP core and was never pushed to the limit by this program. One will probably run out of RAM long before running out of CPU cycles in most applications.