LabLektion 4 - Implementation of digital filters I FIR

Simon Berthilsson

March 7, 2013

1 preparations

Before we get started, download the file "Lab-L4_FIR_Filters.c" from Studentportalen and set the environment up, as in the previous labs by including the following libraries:

- ABDAC – Audio Bitstream DAC (driver)
- ADC – Analog to Digital Converter (driver)
- DSPLib – Digital signal processing library (sverice)
- PDCA – Peripheral DMA Controller (driver)
- PM – Power Manager (driver)
- TC – Time/Counter (driver)
- USART – Universal Synchronous/Asynchronous Receiver/Transmitter (driver)
- System Clock Control (service)

2 Scaling

It is time to get our hands dirty and do some filter implementations on the lab equipment. A low pass digital filter is attached in Table 1 and the magnitude and phase response of the filter is shown in Figure 1

2.1 A quick check

Is the low pass filter presented in Table 1 and Figure 1 a linear or a non linear phase filter?

2.2 Scaling of the filter taps

We are interested in implementing this filter in our processor. In order to do this efficiently, we have decided to use the implemented fractional representation, denoted dsp16_t. First, if we assume that the signal we are to filter is normalized, i.e. it varies between at most $\pm$ 1, then we can make sure that the filter computation never saturates. Using the l_1 norm, we can find a scale factor that guarantees that the filter will not saturate. The l_1 norm is defined below.

$$l_1 = \sum |h|, \tag{1}$$

Table 1: An 11 tap low pass FIR filter

Delay [samples]	Tap Value
0	0.004453696074602
1	0.017195050724915
2	0.040423962018854
3	0.069820963074605
4	0.095051057170103
5	0.105049604476010
6	0.095051057170103
7	0.069820963074605
8	0.040423962018854
9	0.017195050724915
10	0.004453696074602

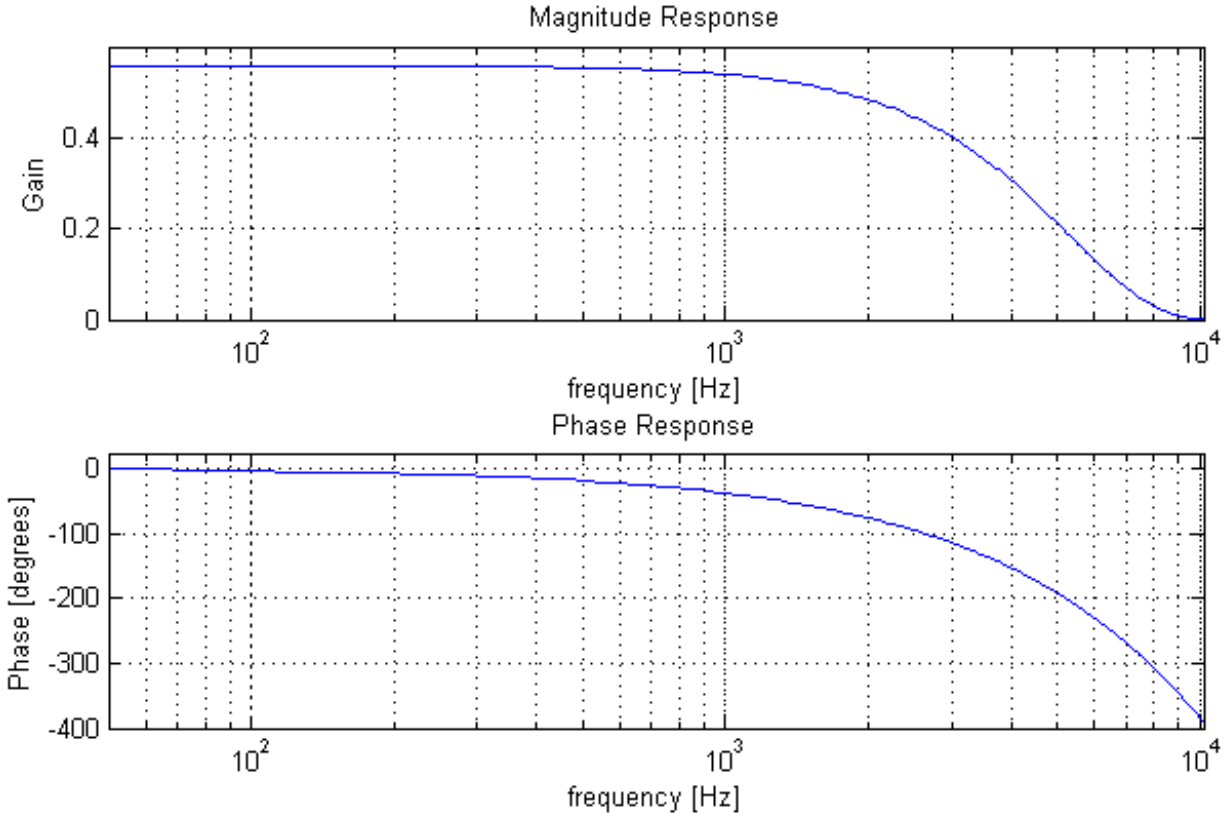

Figure 1: Magnitude and phase response of FIR filter

where h is the impulse response of the filter, or in the FIR filter case, simply the filter coefficients. **Check that the given filter will not saturate**.

Now, you may notice that the filter sum is less than one. This is fine and apparently the behaviour the designer of the filter intended. We however have different concerns than whoever designed the filter. We strive to implement the filter as efficiently and correctly as possible and having this headroom is not beneficial to the filter accuracy. We want to use as many bits as possible, unused bits are wasted bits. So, in order to use the full dynamic range of the processors' word length we want to find the scale factor that brings the filter sum up to the saturation limit. The idea of this is that if we multiply all taps of the filter with this number, we can safely implement it in the processor and get the desired behaviour if we just divide the output with the same factor.

Find and remember the scale factor.

We are now on the verge of implementing this in our processor. There is just one step left, to adapt the filter coefficients to the Q0.15 format. The compiler knows only integers and not fractionals, which means that we need to convert our fractionals to integers before implementing them. The easiest way of doing this is to do the operation $x_Q = round(x * 2^{15})$ where x is the number to be converted.

Find, for each filter tap, the corresponding integer value that enables us to represent the filter as a Q0.15 number.

Multiplication of binary numbers representing fractions differ from multiplications of integers. A method that takes care of this is implemented in the DSP library under the name dsp16_op_mul.

3 Implementation

It is now time to get down to business and do some filtering!

Implement, on Direct form, the given filter. Remember to divide by (multiply by 1/) the scale factor you calculated earlier before outputting the result of the filtering.

Verify that the filter behaves as intended by comparing the filter output to the magnitude and phase characteristics of Figure 1.

Remember to make sure that the filtering frequency matches the intended frequency of the filter (46875 Hz) (what will happen if this is not true?)