Implementing a Tracking Filter on the QF4A512 Programmable Signal Converter

1) Introduction

There are applications where it is necessary to have a filter that can track an incoming signal which moves in frequency. There are several methods to track a signal using a QF4A512.

This application note includes:

- Overview
- Changing Decimation Factor
- Multiple Adjacent Filters
- Multiple Preprogrammed Filters
- Changing PLL Reference Frequency

2) Overview

Tracking a signal with a filter can be accomplished by a variety of techniques. Some of those which can be implemented with the QF4A512 are:

1. Using multiple channels with adjacent filters
2. Loading multiple preprogrammed filters
3. Changing the PLL reference frequency
4. Changing decimation factor

3) Multiple Overlapping Filters

By running multiple channels of the QF4A512 in parallel, a tracking filter can be implemented by comparing the magnitude of the signals in each of the channels. By having adjacent overlapping filters, the frequency of the signal can be tracked by detecting the amplitude in each of the filters.

Figure 1 shows an example of overlapping filters. Four filters with center frequencies of 200, 250, 300, and 350 Hz are programmed into the four channels of the QF4A512. Signals that are from 150 to 200 Hz would only have large amplitudes in channel 1. Signals between 200 and 250 Hz would show up in channels 1 and 2. And so on for the rest of the frequency band. This technique could be used in cases where the signal frequency needs to be adjusted to keep it centered. In the example, the desired pass band would be 250 to 300 Hz, which would correspond to seeing the signal in channels 2 and 3 at the same time. If the signal showed up in channel 1 and 2, the control loop would make a small change to move it back to the desired frequency. If it showed up only in channel 1, a larger correction could be made. The same type of adjustment could be done for conditions with the frequency being too high.

If the purpose is to only track a signal, a way of continuing to track the signal would be to load additional filters to continue to track the signal up or down in frequency. For the example, if the frequency of the signal was detected to be increasing, with it showing up in channels 3 and 4, channel 1 could be reprogrammed to a center frequency of 400 Hz. As the signal frequency continues to increase, it then would show up in channels 4 and 1, so that channel 2 would be reprogrammed, to 450 Hz in this example. And so on. Since the sample frequency can be different for each channel, as the frequency goes up or down, the sample frequency can be adjusted accordingly.
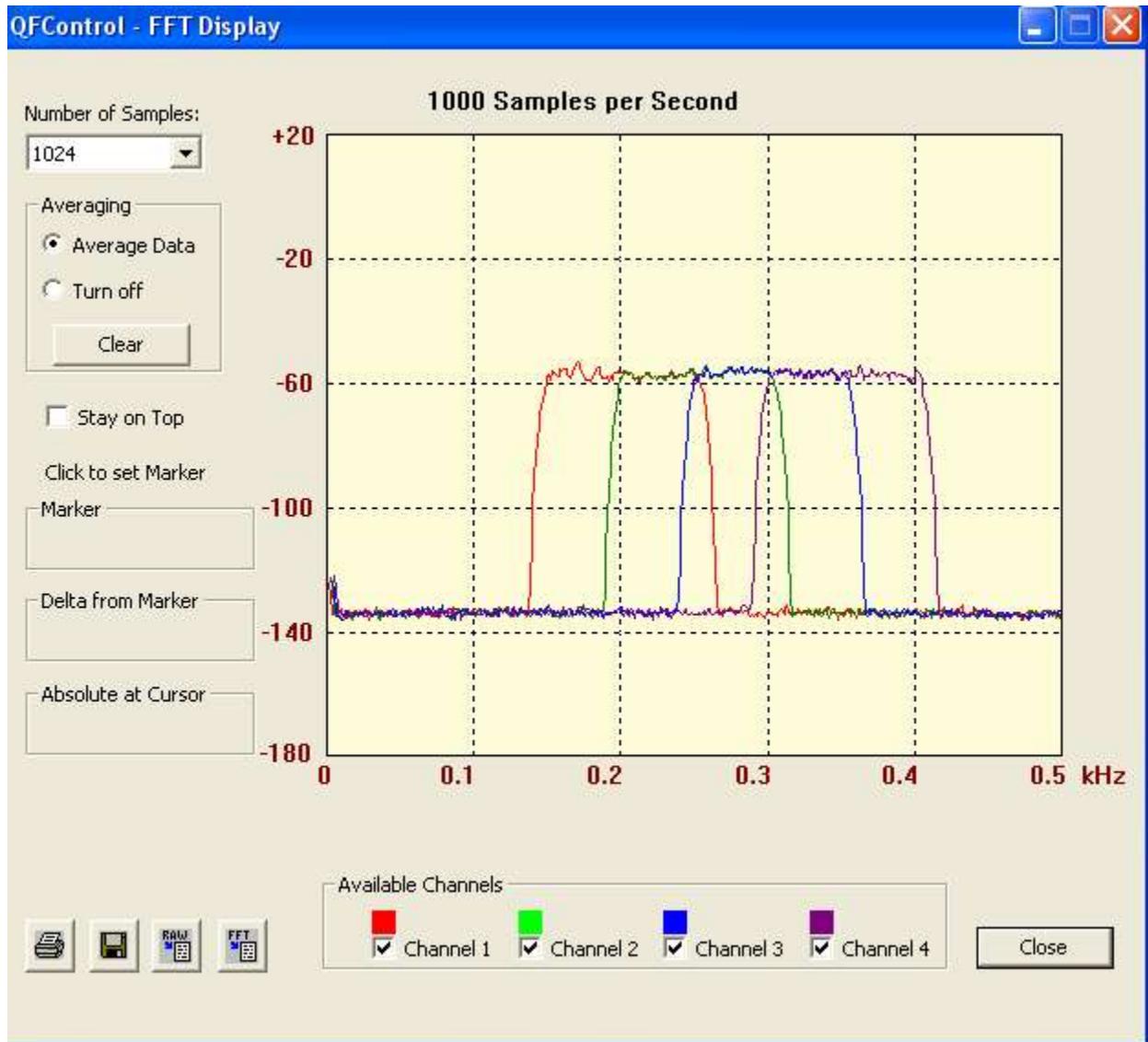
Figure 1: Multiple Adjacent Filters

4) Multiple Preprogrammed Filters

In the case where only a single channel is available for the tracking filter, a similar technique to the previous method is used for multiple filters, but to load them alternately into the same channel. This allows for the same type of tracking filters as above but only utilizing one channel on the device. The limitation is that there will be a period of time in which no valid data is available, so this would be best used for applications in which the frequency of the signal is not expected to move outside the filter bandwidth during the time it takes to reload the filter and to collect the next set of data.

The time in which data would not be valid would be time needed to update the filter configuration (filter coefficients and possible filter configuration) and the time for the filter to settle out with valid data (this is equal to the collection time of the number samples equal to the number of taps of the filter). For the above filters, the minimum time would be about 210 usec to load the coefficients from the EEPROM and another 250 msec to collect the required samples to get valid data out of the FIR filter.

Using the filter characteristics shown in the above example, the filter would initially be alternated between the four filters until the signal is detected. If the signal is detected in the filters centered at 250 and 300 Hz (the filters programmed into channels 2 and 3 above), then those two filters would be alternately sampled. For this example, if

the frequency was detected only in the 250 Hz filter, then the control loop would be adjusted to move the frequency up. And so on for the other conditions.

If tracking of the signal with no adjustment is desired, and the signal was seen only in the 250 Hz filter, then the filter should be alternated between the 200 and 250 Hz filters.

5) Changing PLL Reference Frequency

Another method of changing the filter characteristics is to change the sampling frequency of the digital filter. The filter characteristics are a function of the sample frequency, so by changing the sample frequency the filter characteristics change proportionally. The table below shows an example of the effect of changing the sample frequency.

Table 1: Example Filter Performance vs Sample Frequency

Parameter	Original Filter Performance	Modified Filter Performance
Sample Frequency	10 KHz	20 KHz
Center Frequency	500 Hz	1000 Hz
Lower Band Edge	300 Hz	600 Hz
Upper Band Edge	700 Hz	1400 Hz

One way of adjusting the sample frequency is to change the frequency generated out of the PLL. The PLL has two frequency ranges of operation (selected by bit 6 of the PLL_CTRL_0 register) at 20 to 100 MHz and 100 to 300 MHz. The PLL frequency can be adjusted across these frequency ranges either by changing the reference signal to the PLL or changing the pre-divider and loop divider settings in the PLL.

Changing the pre/loop divider settings allows for changing the frequency without having to have a programmable clock input, but will result in the PLL having to re-lock on the reference signal. This will add approximately 200 – 500 usec delay before being able lock to the reference clock. Using this approach, it is possible to vary the clock across the entire PLL frequency range (20 to 300 MHz). The frequency selection, however, is limited by the number of combinations of the pre-divider and loop-divider. This might not provide enough granularity of frequency selection.

Changing the clock input into the QF4A512 allows for changing the frequency without reconfiguring the device. The loop bandwidth of the PLL is approximately 360 KHz for the 20 to 100 MHz frequency range and approximately 475 KHz for the 100 to 300 MHz frequency range. If the frequency of the reference clock is changed within the bandwidth of the PLL loop, the PLL will not lose lock.

One consideration for using this approach is that the PLL clock drives all of the clocks in the device, so changing this for one channel will also change it for the other channels. So if multiple independent filters are implemented in the QF4A512, this approach would not be appropriate.

6) Changing Decimation Factor

A similar approach to changing the PLL frequency is to change the decimation factor. This has the same effect of changing the sampling frequency and filtering parameters, but since the decimation factor is independent for each of the channels, this approach would be appropriate in applications that have more than one filter in the device.

The decimation of the ADC sample data occurs in the CIC block of the QF4A512. It can range from 1 to 131072. The software calculates this value based on the desired sample and reference clock frequency. The easiest approach to use this for a tracking filter is to use the software calculated decimation value from the software, then adjust it with the ratio of the desired to existing frequency.

There are three sets of registers that need to be adjusted to modify the decimation value. First is the actual decimation registers (CIC_X_R registers, where X is the channel number). The CIC_X_SHIFT register, which scales the data based on the decimation data, also needs to be adjusted. Finally, the AREC_X_GAIN registers, which do a gain correction based on the decimation, also needs to be updated.

The CIC decimation value is determined by multiplying the existing decimation value by the ratio of the desired frequency to the original frequency:

$$DEC_{NEW} = ((DEC_{OLD} + 1) * FREQ_{NEW} / FREQ_{OLD}) - 1$$

DEC_{OLD} and DEC_{NEW} are the values in the CIC_X_R registers, the register is the decimation value – 1.

The shift value (CIC_X_SHIFT) is calculated from the new decimation value using this formula:

$$SHIFT = INT(SHIFT_{REAL} + 0.5)$$

where:

$$SHIFT_{REAL} = \log((131072 / DEC_{NEW})^4 / \log(2))$$

And finally, the AREC_X_GAIN value is determined with this equation:

$$AREC = INT(16384 * 2^{(SHIFTREAL - SHIFT)} + 0.5)$$

Included with this application note is a spreadsheet which does the calculation for these values. Figure 2 shows an example from this spreadsheet.
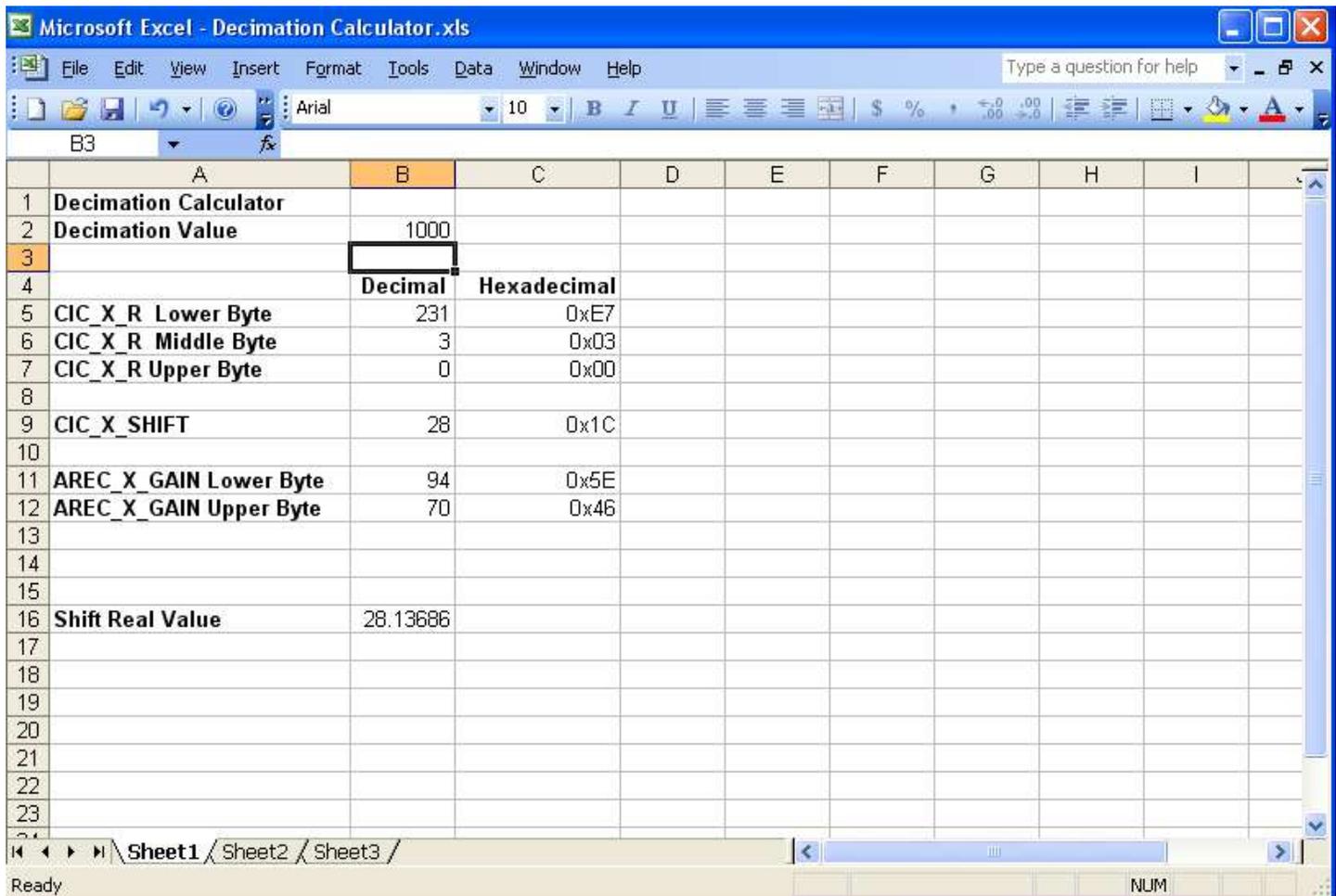
Figure 2: Decimation Spreadsheet Example

As with the previous method, the sample frequency and all of the filter characteristics are scaled by the ratio of the decimation values.

7) Summary

Tracking filters can be implemented in the QF4A512 by several different methods. Each of these have different advantages depending on the system requirements.

Contact Information:

Quickfilter Technologies, Inc.
1024 S. Greenville Avenue, Suite 100
Allen, TX 75002-3324

General: info@quickfilter.net
Applications: apps@quickfilter.net
Sales: sales@quickfilter.net
Phone: 214-547-0460
Fax: 214-547-0481
Web: www.quickfiltertech.com