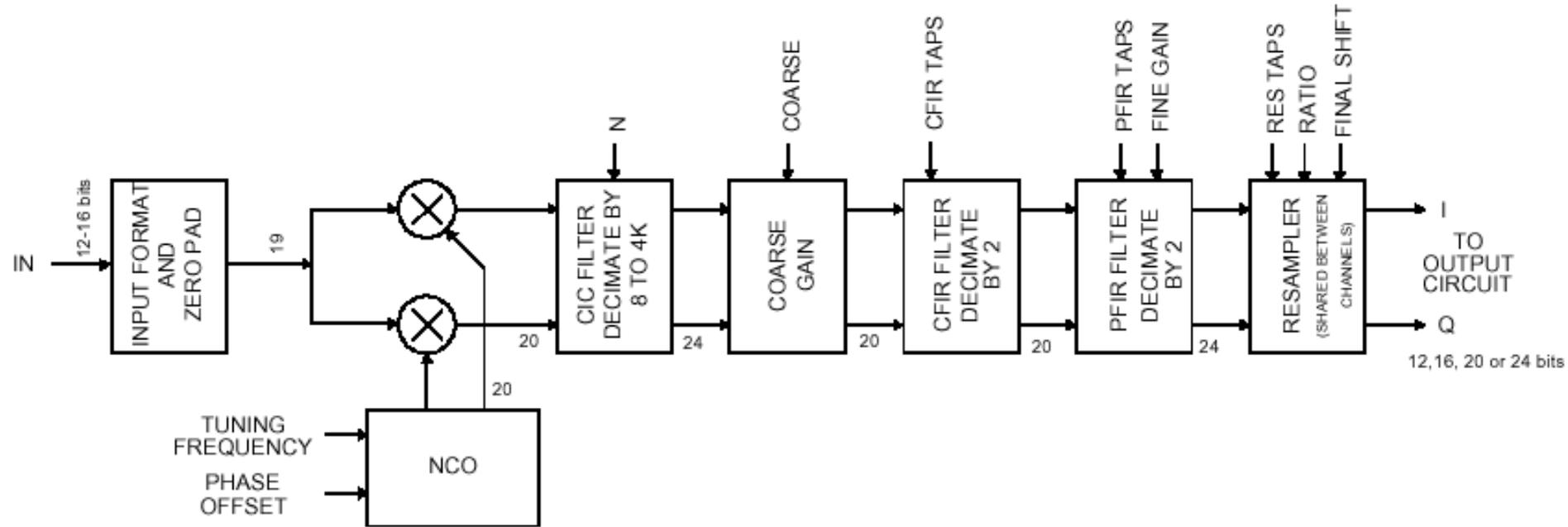


# Greychip, filters, and simulations

BPM project

# Greychip Block Diagram



☀ The block diagram shows 1 of 4 channels in the Greychip

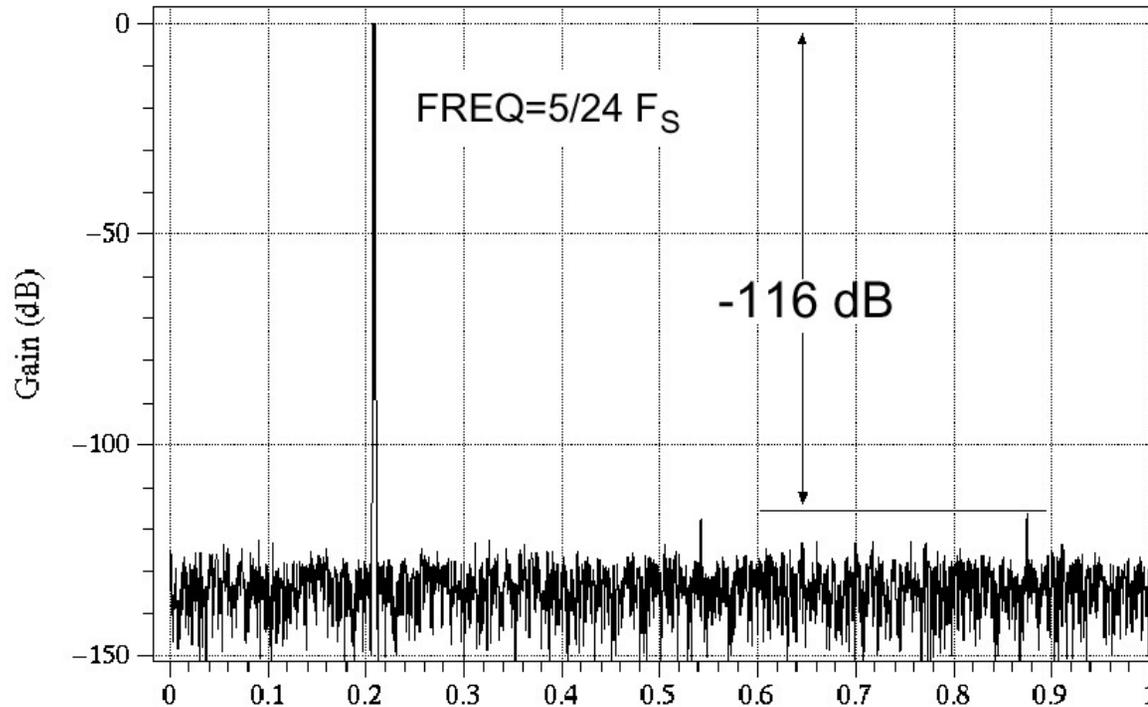
- Data input format
- Down-converter
- CIC filter
- CFIR filter
- PFIR filter
- Resampler

# Input data format

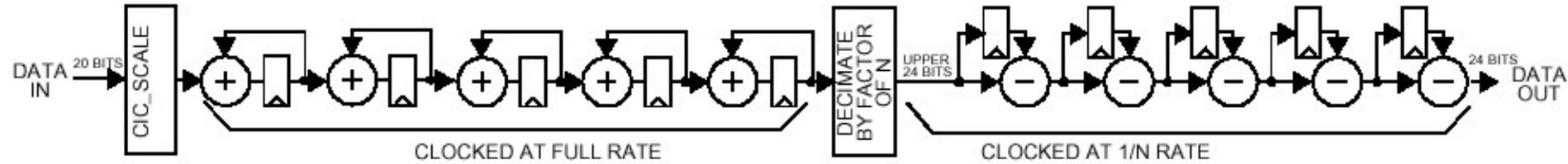
- ☀ Available Greychip data formats
  - Four input ports of 14 bit data
  - Three input ports of 16 bit data
  - Two input ports of 14 bit LVDS data
  - Three input ports of 12 bits plus 3 bit exponent (floating point)
  - Three input ports of multiplexed 12 bits plus 3 bit exponent (floating point)
- ☀ We don't really care much about this list because the Ecotek only works with four input ports of 14 bit data
- ☀ The input can be zero-padded if we want to operate the clock faster than the sample rate.

# Down-converter

- ☀ Each Down-converter contains a Numerically Controlled Oscillator (NCO)
  - The frequency can be tuned to  $2^{-32}$  of  $F_{clk}$  (clock freq.)
  - The phase can be adjusted by  $2^{-16}$  of  $2\pi$ .
  - Dither generator to provide a 121bd spur free clock.



# CIC filter

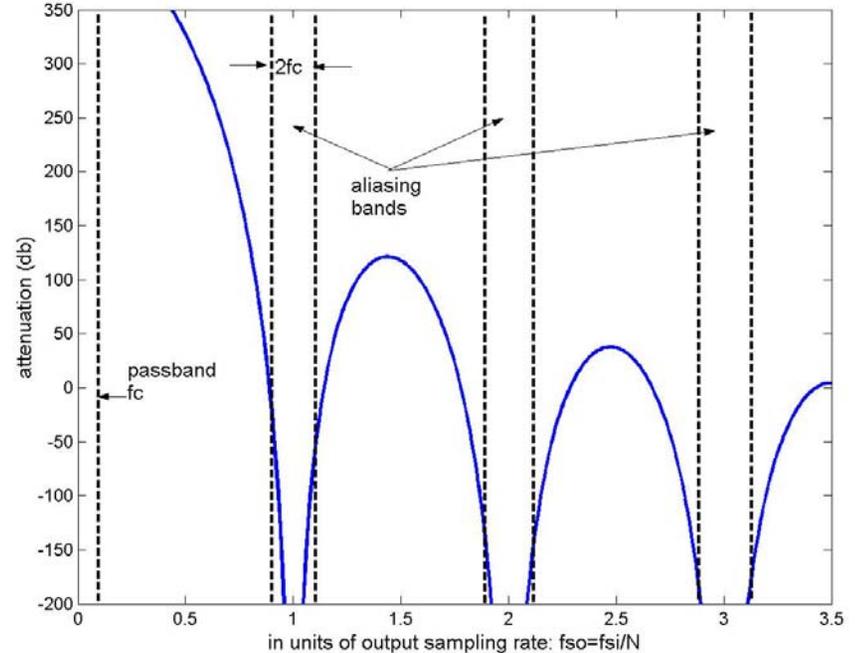


- ☀ 5 stage integrator and 5 stage differentiator.
- ☀ Decimator by 8 to 4096.
- ☀ Input signal gain correction module to correct the  $N^5$  gain factor introduced by the CIC. Where  $N$  is the decimation rate.
- ☀ Coarse gain output signal correction.

## CIC power spectrum

$$P(f) = \left[ \frac{\sin \pi f}{\sin \pi f / N} \right]^{2M}$$

Where  $M$  is 5, the number of stages



# CFIR filter

- ☀ The CFIR filter usually compensates for “droops” in the CIC filter.
- ☀ 21 taps.
  - Symmetric and non-symmetric modes available (only 11 taps in non-symmetric mode).
- ☀ Decimates by 2.
  - Although decimation by 2 can be avoided using it in multichannel mode.
- ☀ Typical CIC implementation has:
  - Frequency response with wide transition bands.
  - Passband ripple as small as possible.
- ☀ The output can be delayed by one sample to allow for multichannel modes. The delay is independent for I and Q, and for each channel.

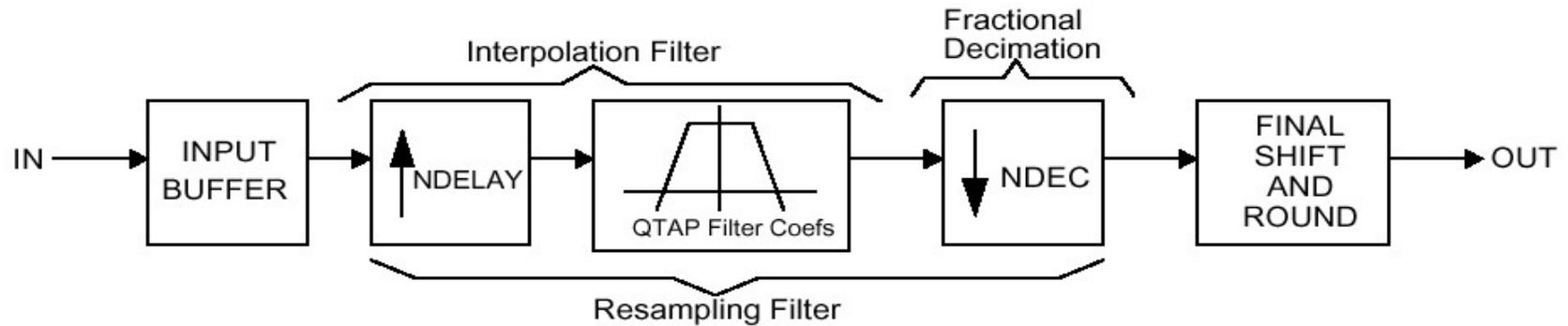
# PFIR filter

- ✱ The PFIR filter has enough taps to make a narrow transition band filter.
- ✱ 63 taps.
  - Symmetric and non-symmetric modes available (only 32 taps in non-symmetric mode).
- ✱ Decimates by 2.
- ✱ Fine gain control stage.
- ✱ Output sample groups can be multiplied by -1 to invert the signal. Groups are selected as follow:
  - Even real outputs.
  - Even imaginary outputs.
  - Odd real outputs.
  - Odd imaginary outputs.

# Multichannel modes

- ✿ The maximum output sample rate per channel (before the Resampler) is  $F_{clk}/32$  (i.e. less than 2MHz at maximum clock freq.).
- ✿ The output bandwidth can be extended by combining channels.
- ✿ The Resampler can produce  $nX$  oversampled data or decimate the data more.
- ✿ The SPLIT I/Q mode uses two channels. One channel to process the real part and the other for the imaginary part. There must be a  $90^\circ$  phase shift between channel NCOs.
- ✿ For wider band, four channels can be combined into one using the PFIR.

# Resampler



- ✿ NDELAY and NDEC can be combined to achieve the desired output sampling frequency (i.e. integer or rational of input freq.).
- ✿ The input buffer must be configured according to the channel (or multichannel) configuration.
- ✿ The total number of taps for filtering in the Resampler is 512, in symmetric mode. Up to four filters can be implemented (i.e. one per channel). In that case each filter has 128 taps.
- ✿ The final shift and round is another gain stage.
- ✿ The Resampler can be bypassed.

# Greychip gain

☀ The big monster is:

$$\text{GAIN} = \left\{ \left( \frac{1}{\text{NZEROS} + 1} \right) N^5 2^{(\text{SHIFT} + \text{SCALE} + 6 \times \text{BIG\_SCALE} - 62)} \right\} (2^{\text{COARSE}}) \left( \frac{\text{CFIR\_SUM}}{65536} \right) \left( \frac{\text{PFIR\_SUM}}{65536} \right) \left( \frac{\text{FINE\_GAIN}}{1024} \right) \left( \frac{\text{RES\_SUM}}{32768 \times \text{NDELAY}} \right) (2^{\text{FINAL\_SHIFT}})$$

We must carefully analyze the signal gain in each stage to prevent overflow and to avoid large rounding errors.

In the CIC filter, overflows go unflagged.

In the CFIR and PFIR the overflows are detected and hard limited.

# Greychip simulation in Matlab

## ☀ Blocks implemented:

- Down-converter.
- CIC filter.
- CFIR filter.
- PFIR filter.
- Resampler.

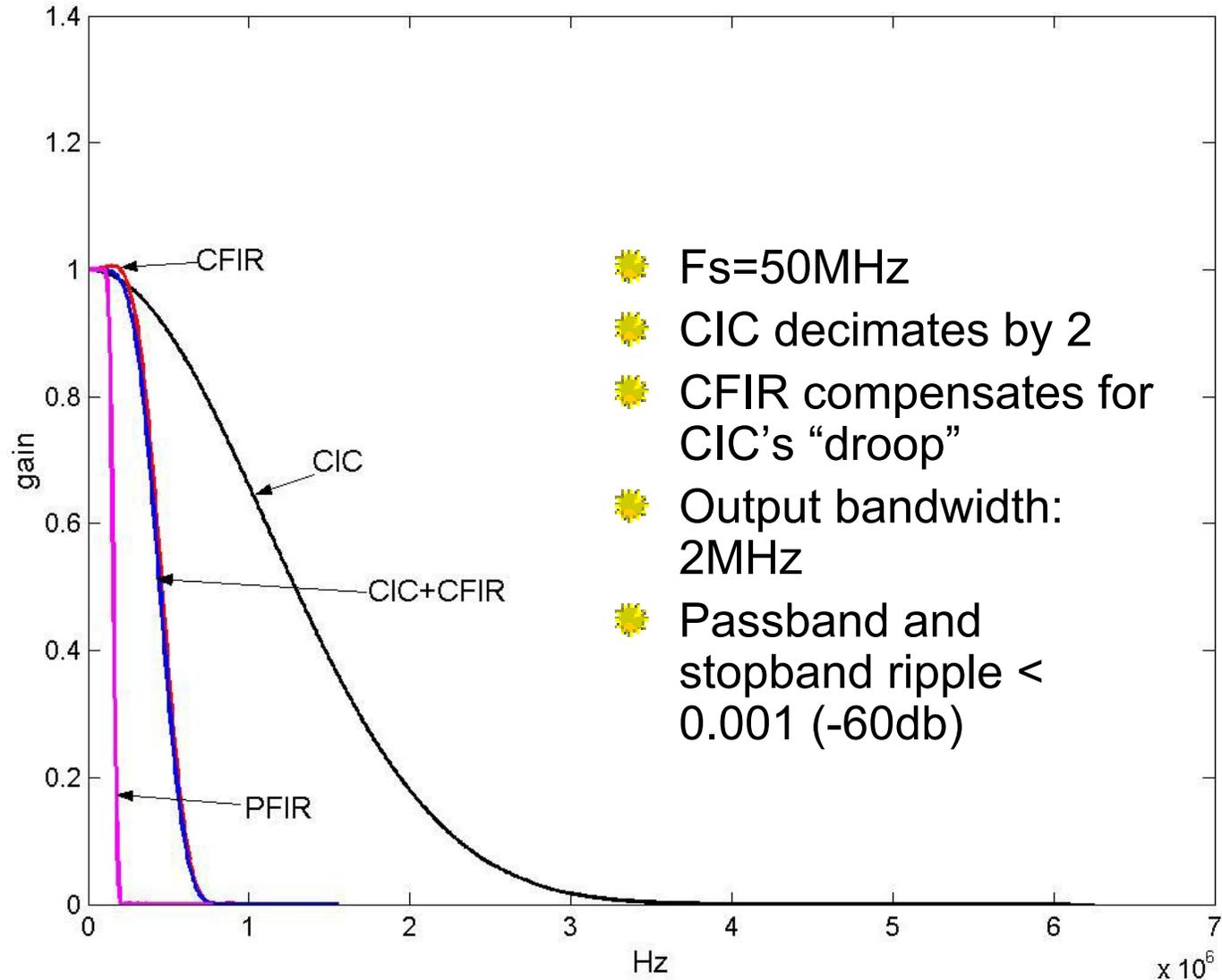
## ☀ Application examples:

- Separation of very simple signal and noise .
- Tevatron A14 A and B signals (provided by Jim Steimel).

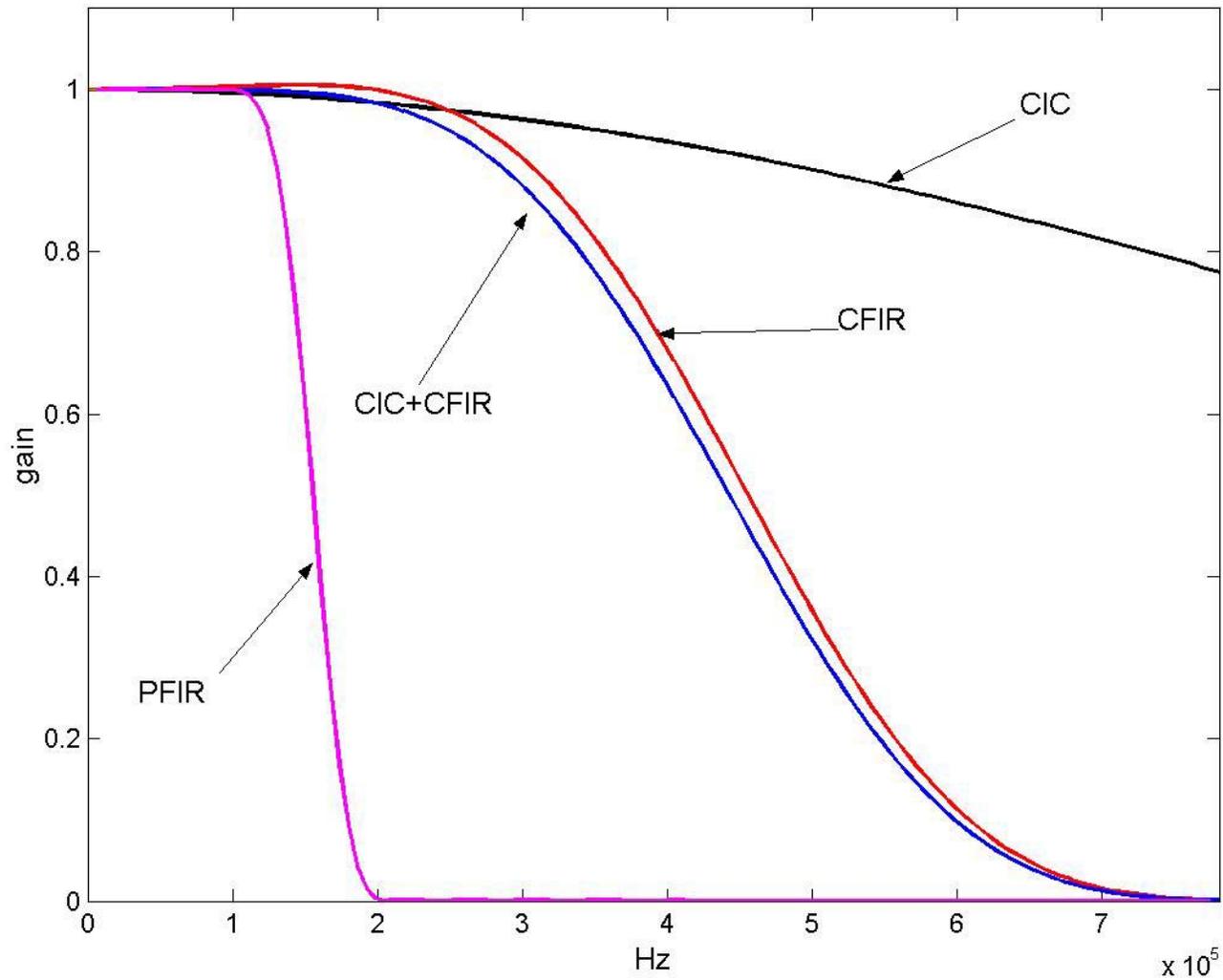
# Matlab filtering routines

- ☀ The filter algorithms available depend on the Matlab toolbox and version available.
- ☀ The new Filter toolbox has everything we may ever need. We should buy it.
- ☀ For now I'm using `remez` and `gremez` algorithms. They use  $L_\infty$ -norm and  $L_2$ -norm optimization.
  - `Remez` generates linear phase FIRs. The input parameters are number of taps, passband/stopband ripple and transition band width.
  - `Gremez` is more general. I use this one for the PFIR. It allows for non linear phase implementation, and can be supplied with more input parameters to constrain the optimization.

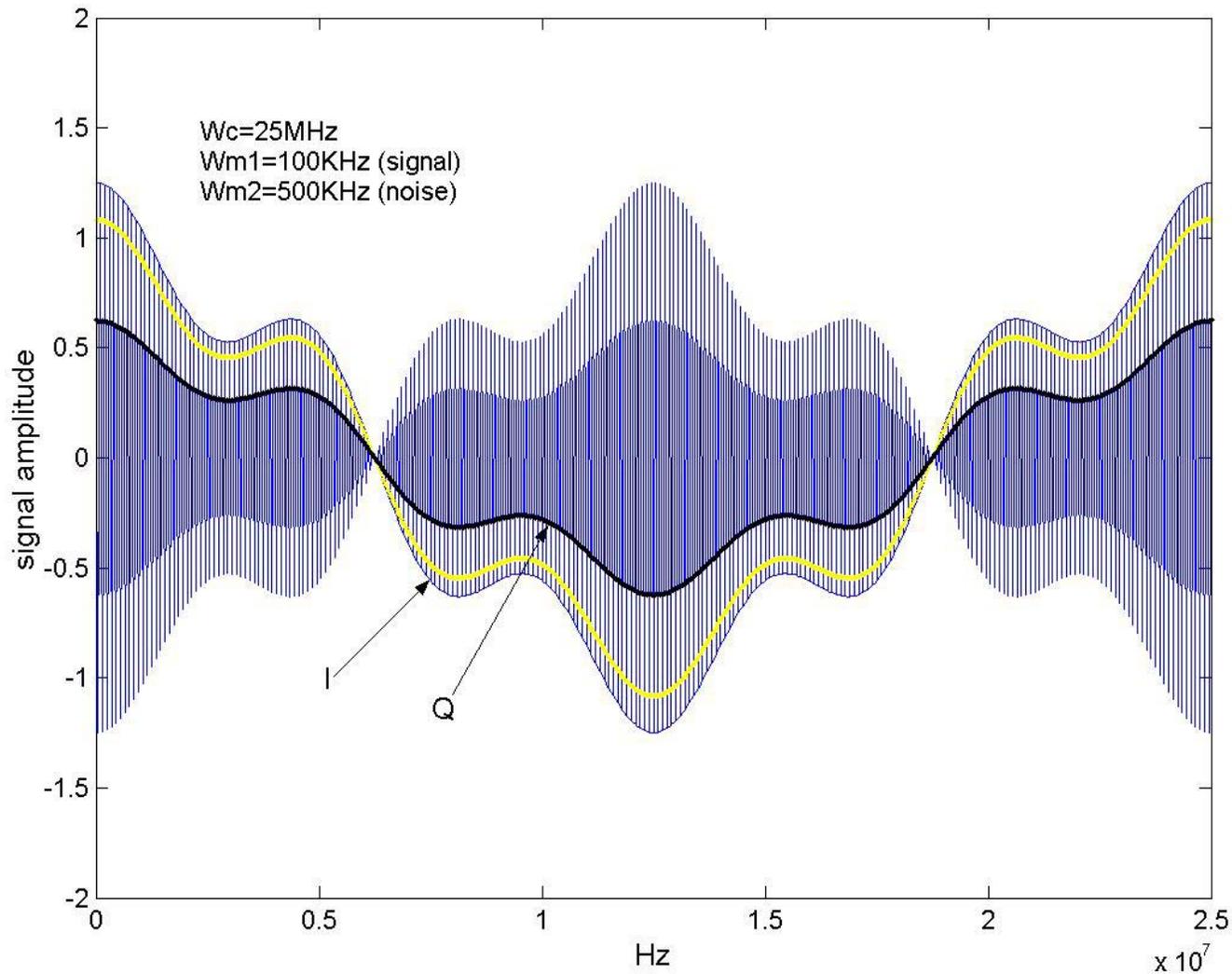
# Filter frequency response (1)



# Filter frequency response (2)

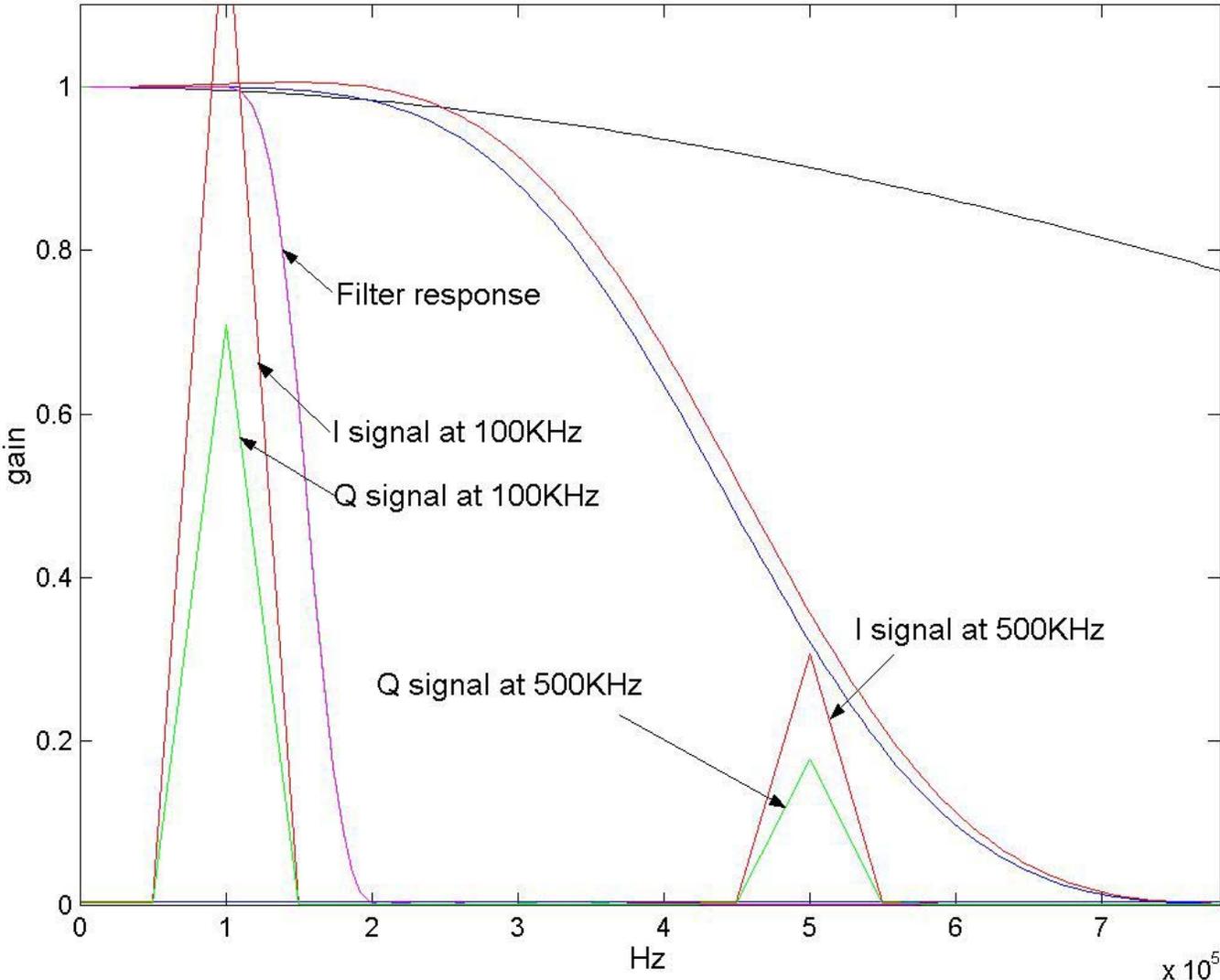


# Example1: Signal plus noise

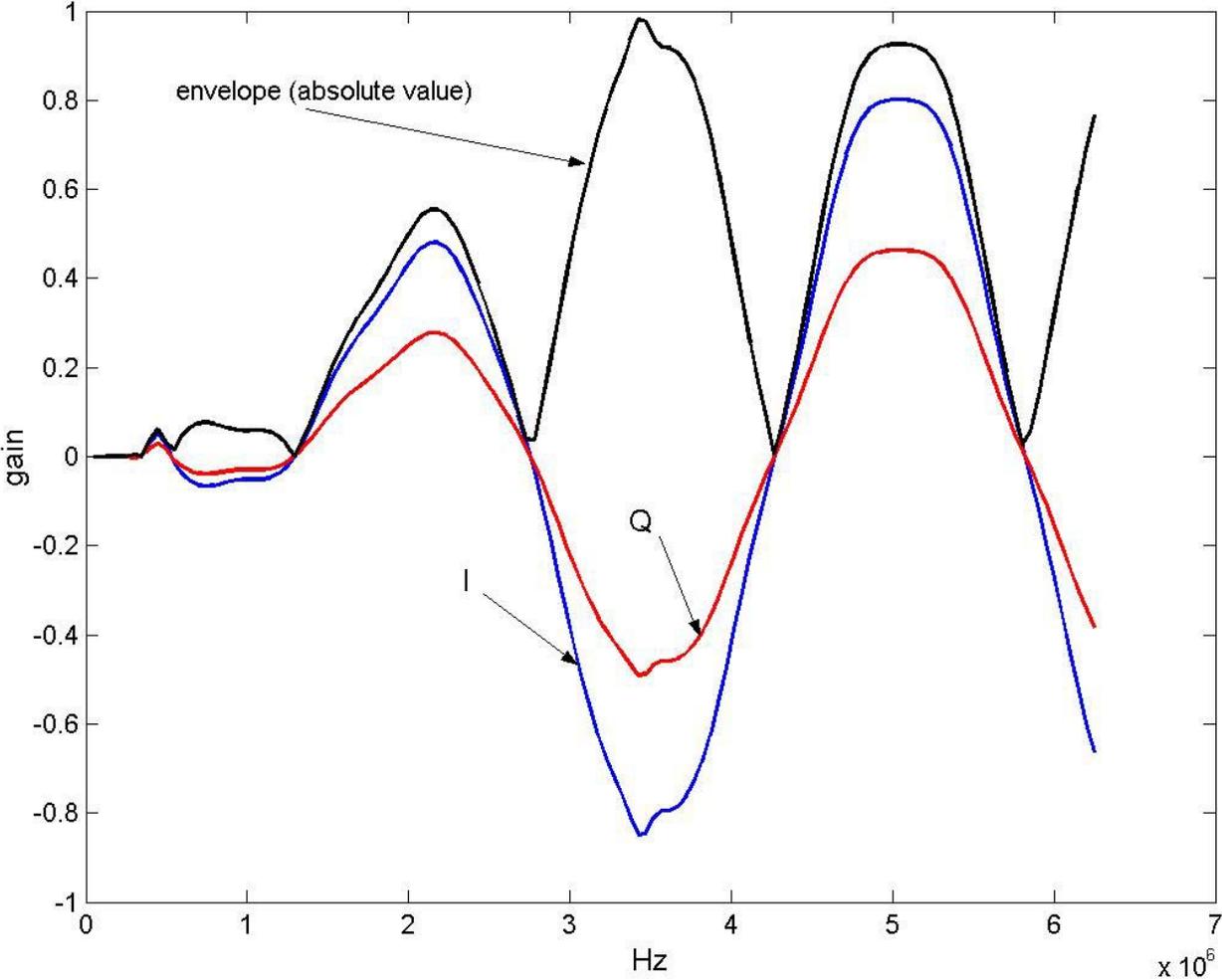


☀ Filter the 500MHz noise. S/N ratio is only 4

# Example1: Signal plus noise

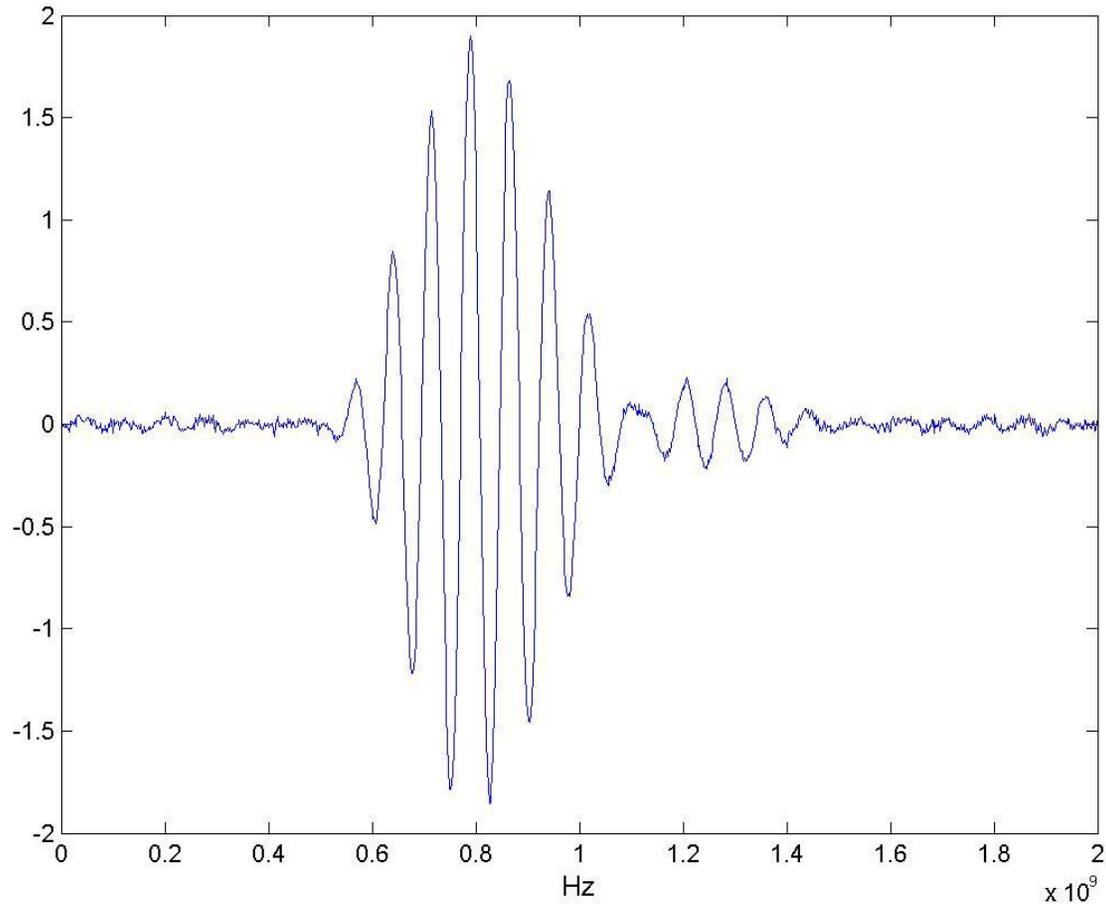


# Example 1: Signal plus noise



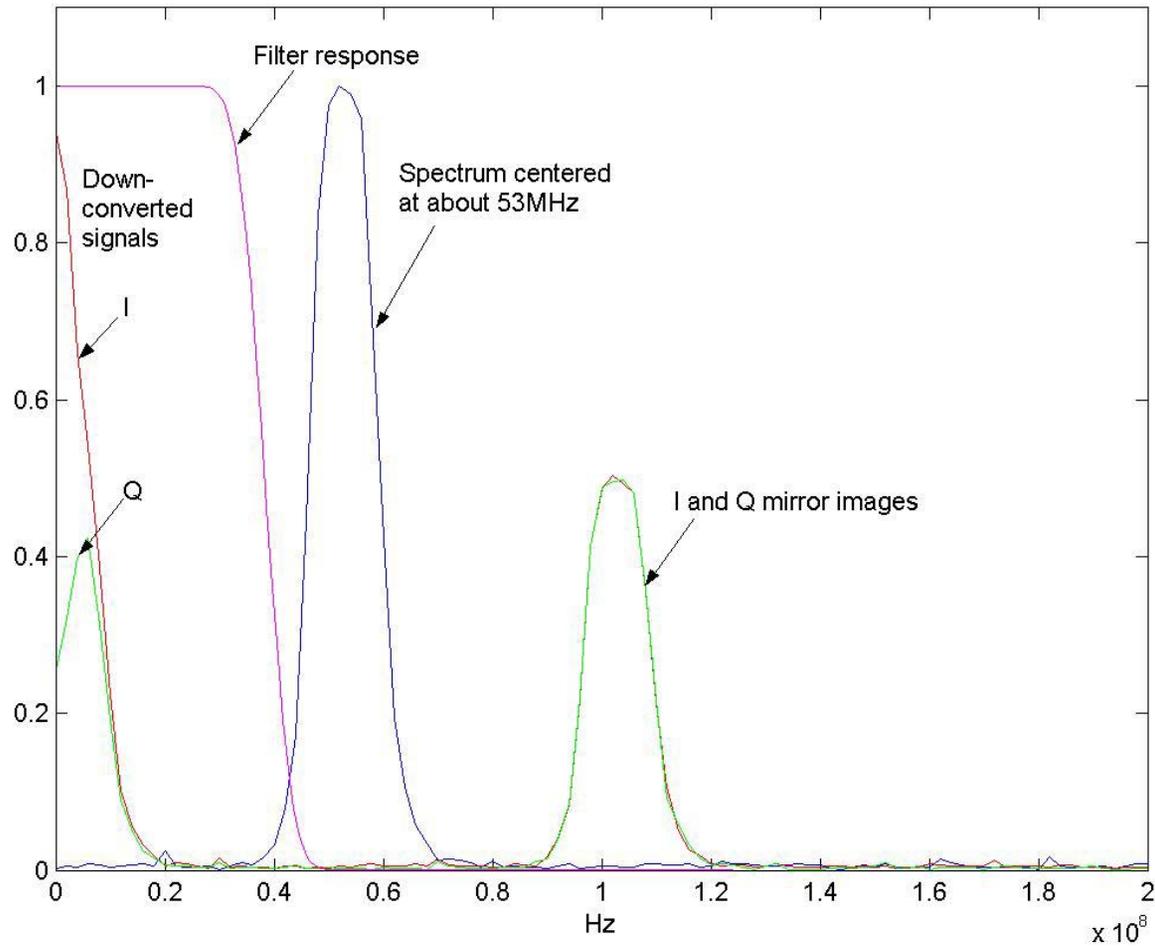
# Example2: A14 signal

- ☀ Assumed:  $f_s=2\text{GHz}$
- ☀ Goal: Recover the envelope



# Example2: A14 signal

## ☀ Downconversion and filtering



# Example2: A14 signal

