

# Interpolation Filters for the GNURadio+USRP2 Platform

Project Report for the Course  
442.087 Seminar/Projekt Signal Processing

0173820 Hermann Kureck

# 1 Executive Summary

The USRP2 platform is a typical Software Defined Radio system which can be used for transmitting and receiving various kinds of signals. Usable frequency ranges depend on the used daughterboards. For transmitting 802.11a/g and 802.11p signals for example the GNURadio implementation in [And] can be used. In this case interpolation factors of 5 and 10 are required. The interpolation of the signal is done on the FPGA to convert the sampling rate to the one used by the DA-Converter. Because of poor frequency responses when using odd interpolation factors, where only simple CIC-Filters are used for interpolation, a new interpolation filter which interpolates by a factor of 5 was implemented on the FPGA using the hardware description language Verilog. This report includes analysis of the current TX-chain, design and implementation of the new filter.

## 2 Analysis of current USRP2's DSP Implementation

The current structure of the transmit path is shown in Figure 1. It consists of interpolation stages for converting the sampling rate to the one used in the DAC. This is often done to reduce overall complexity. Then the CORDIC algorithm is used to move the complex baseband signal to an intermediate frequency. This allows for frequency fine tuning and for example also more than one DUC/DDC-chain is possible for simultaneously transmitting/receiving in different channels). After scaling the signal is forwarded to the DAC.

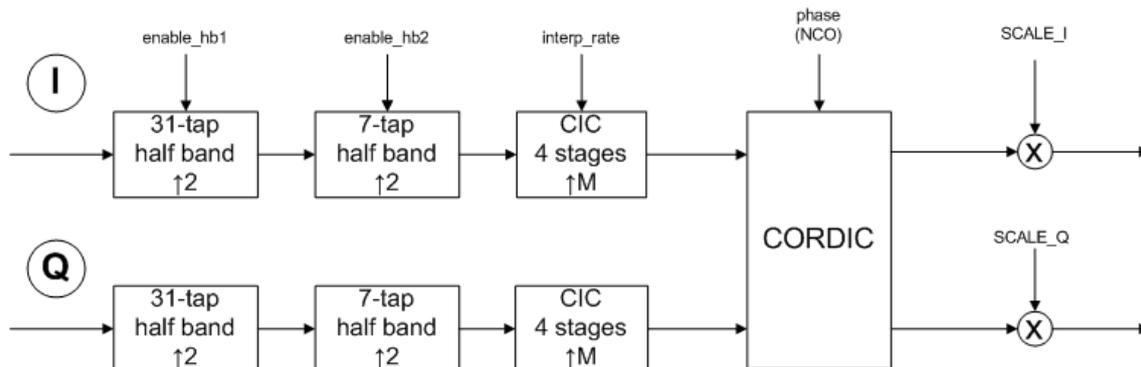


Figure 1: Structure of Transmit Path

### 2.1 The CORDIC Algorithm

This algorithm is widely used in hardware implementations for various purposes including calculating trigonometric function, angles and rotating complex numbers. The last application could also be achieved with complex multiplication with  $e^{j\varphi}$ , but the CORDIC algorithm doesn't use multipliers at all. For algorithm details see [And98].

## 2.2 FIR Half-Band Filters - Properties and Frequency Responses

Let a  $N$ -th order, symmetric half-band filter ( $N$  even!) have the coefficients  $h[n]$ ,  $h[n] = 0$  for  $n \notin [0, N]$ ,  $h[n] = h[N - n]$ , which satisfy

$$h[n] = \begin{cases} 0, & \text{for } n - \frac{N-1}{2} = \text{even and nonzero} \\ \frac{1}{2}, & \text{for } n = \frac{N-1}{2} \end{cases}$$

Then the band edges of the frequency response (stop- and passband frequency) are located symmetrically around the normalized half-band frequency 0.5 ( $\omega_p + \omega_s = 1$ ). Additionally, the pass- and stopband-ripples are equal ( $\delta_p = \delta_s = \delta$ ).

Because every second filter coefficient is 0 (except the one in the middle), half-band filters are efficient for hardware designs. They need only half of the multipliers compared to other FIR filters with symmetric impulse responses. In the example of the USRP2 implementation only 2 multipliers are needed for the 31-tap half band filter for the following reasons:

- There are only 8 different non-zero coefficients for symmetry reasons, so when using pre-additions only 8 multiplications per output have to be performed.
- Since the minimum interpolation factor allowed is 4, and the half-band filter runs at the lowest rate, it has to provide one output in 2 clock cycles in the worst case.
- One of the two polyphase components has nearly all zeroes as impulse response, and one coefficient is  $\frac{1}{2}$ , so nothing needs to be calculated since this is only a bitshift.
- The other polyphase component can use at least 4 clock cycles (every second output) to calculate 8 multiplications.

For similar reasons the 7-tap half-band filter needs only one multiplier (4 non-zero taps, 2 multiplications in 2 clock cycles).

The normalized frequency responses of the two built-in half-band filters are shown in Figure 2.

## 2.3 Cascaded Integrating Comb Filters - Properties and Frequency Responses

The structure of the built-in 4-stage CIC-Filter is shown in Figure 3. A CIC-filter is an efficient implementation of an moving average filter, because of this the frequency response is of *sinc*-shape. The interpolation rate can be arbitrarily chosen. The normalized frequency response of the CIC configured with an interpolation factor of 5 is shown in Figure 2. For more information about how to implement CIC-Filters see [Don00].

## 2.4 Overall Frequency Responses

The overall frequency responses depend on the actual interpolation factor. For odd interpolation only the CIC is used, which has a poor frequency response. For even interpolation the low-rate 31-tap FIR filter is additionally used, and if the interpolation factor is a multiple of 4 the higher-rate 7-tap FIR filter is added as a third interpolation stage.

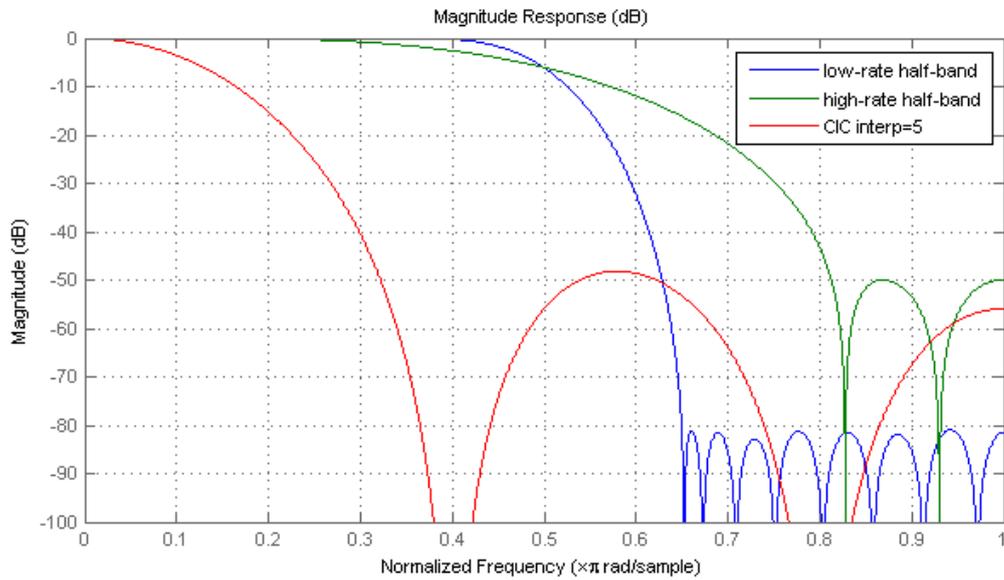


Figure 2: Frequency Responses of the built-in half-band filters and the CIC-Filter for an interpolation factor of 5

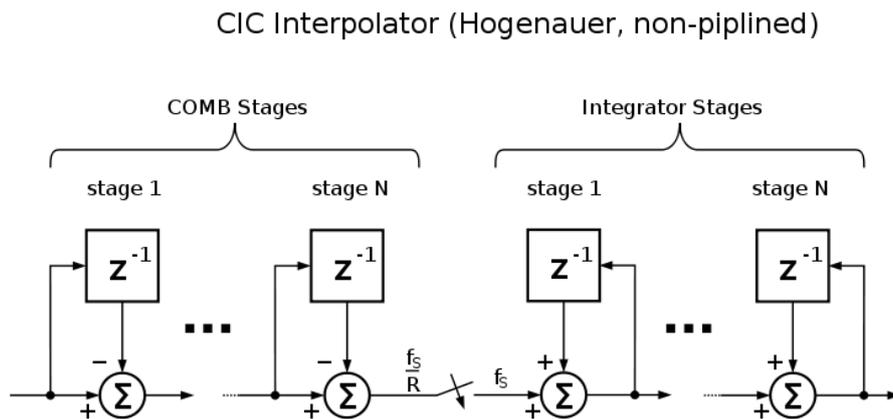


Figure 3: Structure of an CIC Interpolator

In the case of the 802.11g standard we need an overall interpolation factor of 5. Because of the poor frequency response of the currently used highly configurable and flexible CIC-filter a new filter is designed for this special interpolation factor. In the case of the 802.11p standard we need an overall interpolation factor of 10, which will use the low-rate half-band filter and our new interpolation 5 filter, which will result in a flatter passband because no CIC filter stage is involved.

The actual frequency responses for these 2 cases are shown in Figure 4.

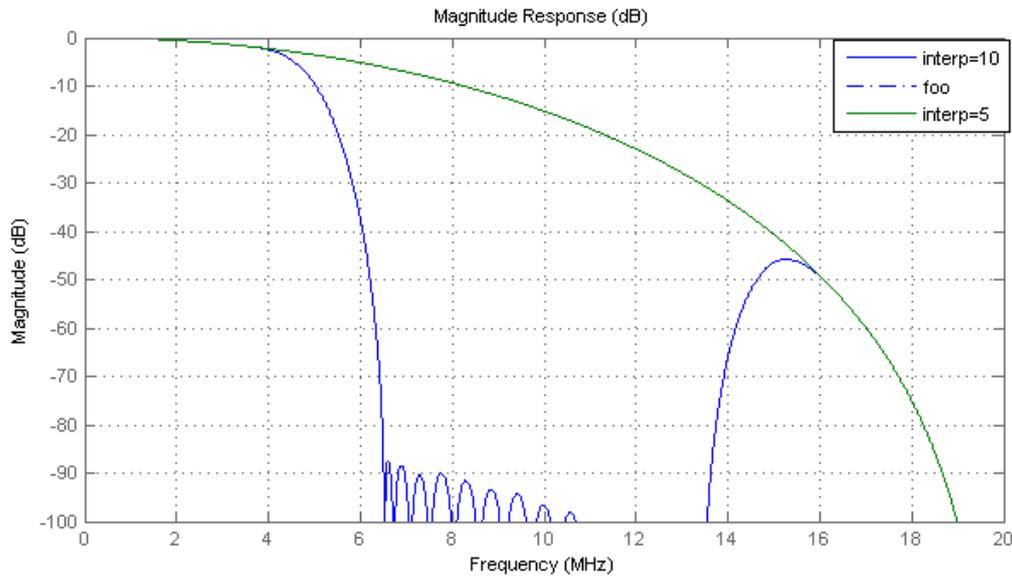


Figure 4: Overall Frequency Responses for Interpolation Factor 5 and 10

### 3 Design of the new Interpolation Filter

#### 3.1 Filter Requirements

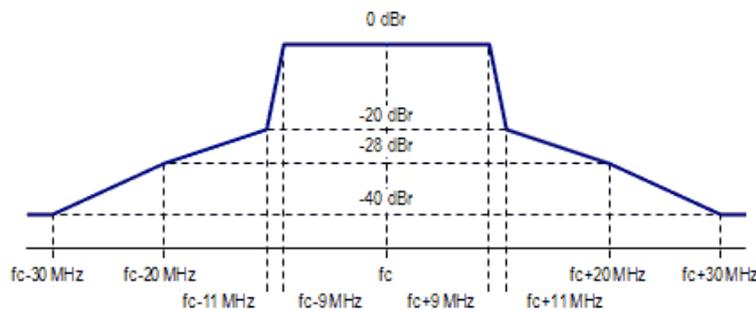


Figure 5: Spectral mask when using 802.11g/a

The spectral mask of the 802.11g/a standard is shown in Figure 5. 52 subcarriers spaced by 312.5 kHz are used, that means the occupied bandwidth is 16.6 MHz in a 20 MHz channel. Speaking in normalized frequencies (related to the nyquist frequency) the signal

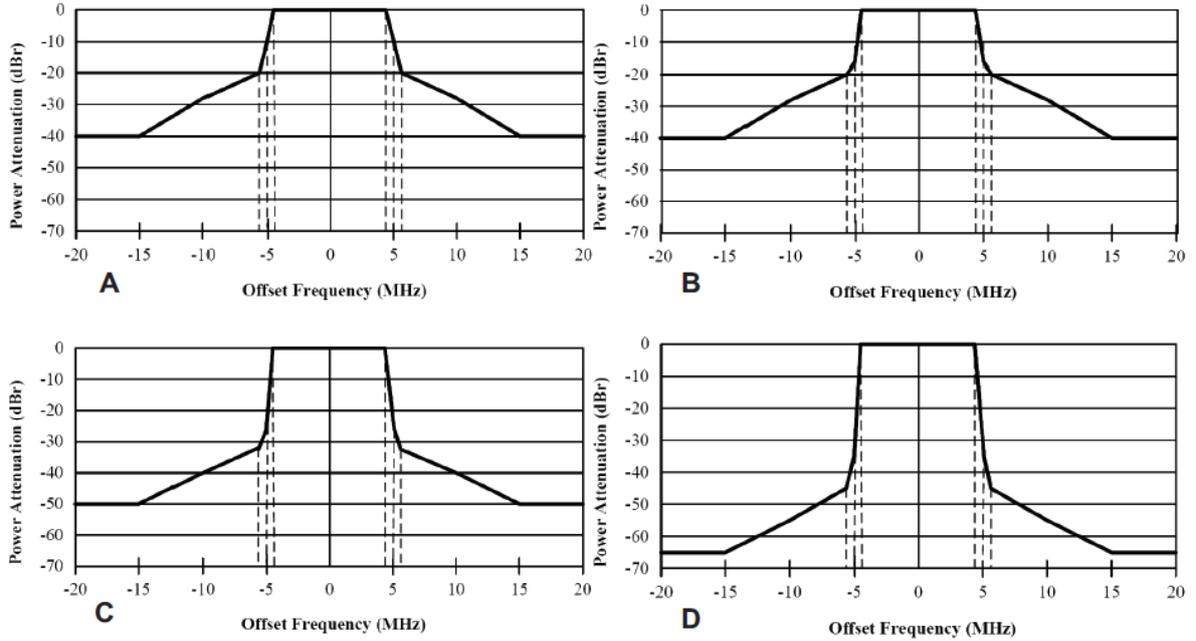


Figure 6: Spectral mask definitions when using 802.11p. Class C allows a transmit power of 100mW.

occupies the band from  $\omega = 0$  to  $\omega = 0.833$  before interpolation. When upsampling the signal by a factor of 5 the wanted signal is in the band from  $\omega = 0$  to  $\omega = 0.167$ , the don't care band reaches from  $\omega = 0.167$  to  $\omega = 0.233$ .

The filter specifications used were:

- $\omega_p = 0.167$
- $\omega_s = 0.233$
- $\delta = 0.2$  dB
- Minimum stopband attenuation: 50 dB to also meet the requirements for 802.11p Class C.

### 3.2 Choosing Filter Type

The decision if an IIR should be implemented was an easy one because:

- The only real advantage of an IIR is that the required filter order is small.
- Coefficient quantization and the implementation is more complicated because of the feedback.
- An IIR can become unstable due to feedback and the quantization as a nonlinear effect can lead to limit cycles.
- They can not take computational advantages in multirate applications as FIR filters do (e.g. polyphase decomposition).

There are many design procedures for efficient FIR filters, like:

- Optimize for low coefficient complexity, that means less ones in the binary (or CSD) representation. In this case multiplications can be avoided and substituted by bitshifts and additions/subtractions. See for example [SB08].
- Other arithmetic stuff like MCM (multiple constant multiplication), see [GD04].
- Filter sharpening techniques. However, in multirate applications this technique can only be useful if multistage decimation/interpolation is possible, which is not in our case because 5 is a prime number.
- Frequency Response Masking (FRM) techniques, which will be discussed now.

### 3.3 The principle of FRM techniques

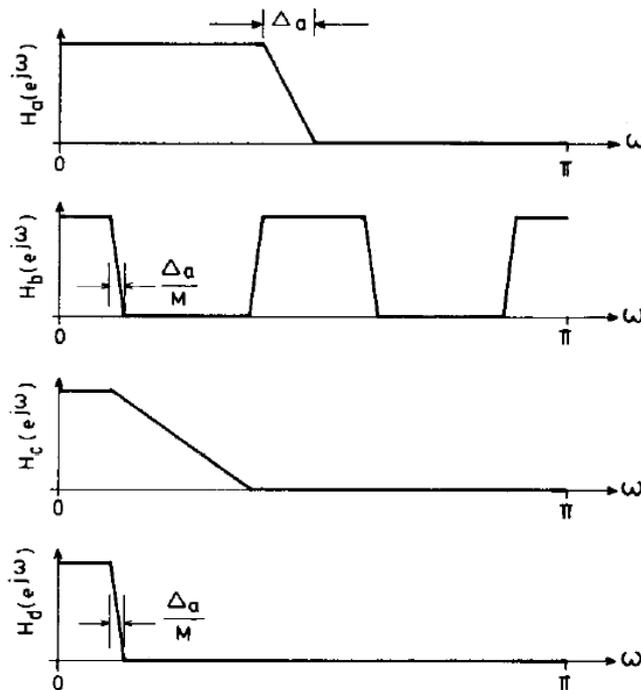


Figure 7: How FRM works. The model filter, the upsampled model filter, the masking filter for image suppression, and the overall frequency responses are shown. In this case  $M$  zeroes are inserted between the coefficients of the model filter. The transition width is lowered from  $\Delta a$  to  $\frac{\Delta a}{M}$ .

The idea of FRM is shown in Figure 7. First the model filter is designed, then it is upsampled, resulting in narrower transition bands. Such a filter also is called periodic filter because periodic images of the passband are introduced with upsampling. The images then have to be removed with an appropriate lowpass filter. Special care has to be taken of the ripples, which could sum up. Of course this simple principle can also be applied to bandpass filters by filtering out an image of the upsampled model filter.

The application of this simple FRM technique (also known as IFIR - Interpolated FIR), however, is limited to narrowband filters, because a higher upsampling factor lowers the bandwidth of the filter. There are advanced FRM techniques, using a complementary model filter and two masking filters (see [Lim86]). However, special design procedures and relations between the filters have to be defined for taking the full computational advantage in polyphase implementation (see for example [Joh05]).

Because the increased effort in design and implementation, and the fact that our filter to be designed has not to be that sharp that it will benefit from advanced FRM techniques, the design choice was a simple IFIR filter.

### 3.4 Designing the IFIR Filter

The filter was designed in MATLAB using the *ifir*-function. Because the masking filter runs at the full sampling rate we want a simple one. Therefore for the model filter only an upsampling factor of 2 was chosen. The *ifir*-routine was called with the 'adv'-flag which additionally allows for a simpler masking filter because it could have a wider transition band which is illustrated in Figure 8. This Figure shows the frequency responses of the designed filters, for comparison also with quantized coefficients. In Figure 9 the frequency response of the combined filter is shown. The overall specifications are easily fulfilled, last but not least also because the filter order could be increased without ending up with more multipliers necessary. Last but not least in Figure 10 the overall frequency response for interpolation factor 10 is shown.

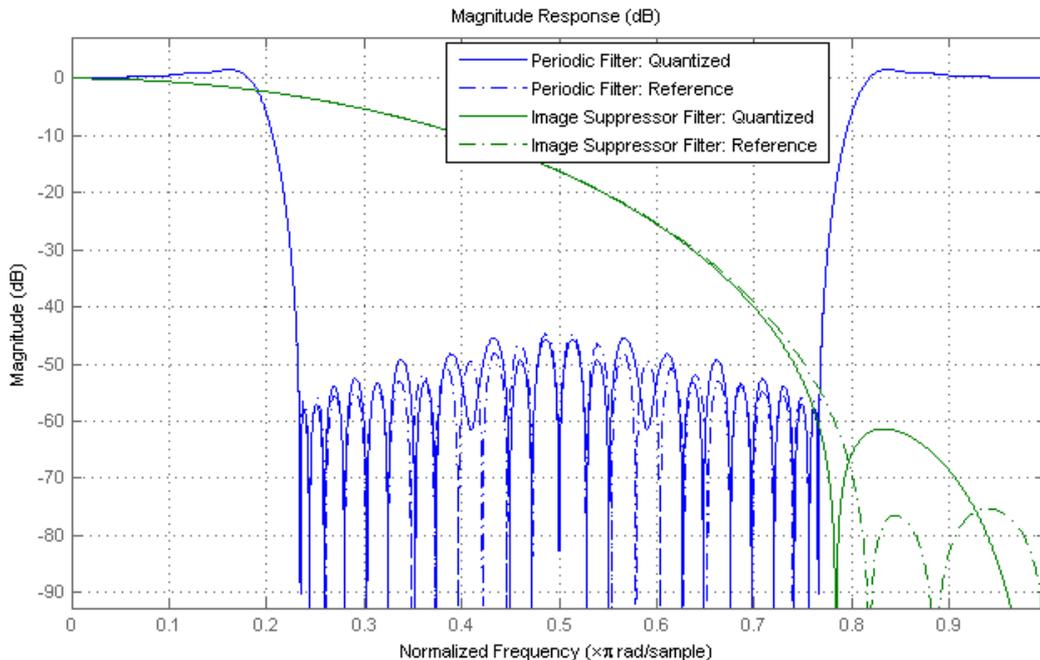


Figure 8: Frequency responses of the designed model and masking filters

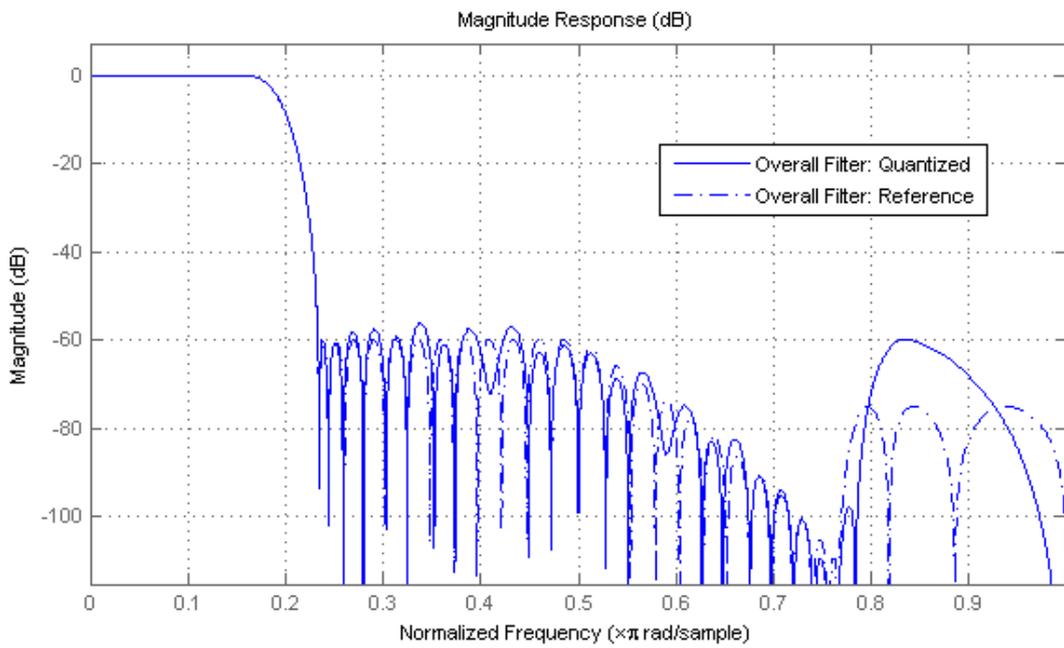


Figure 9: Frequency response of the designed IFIR filter

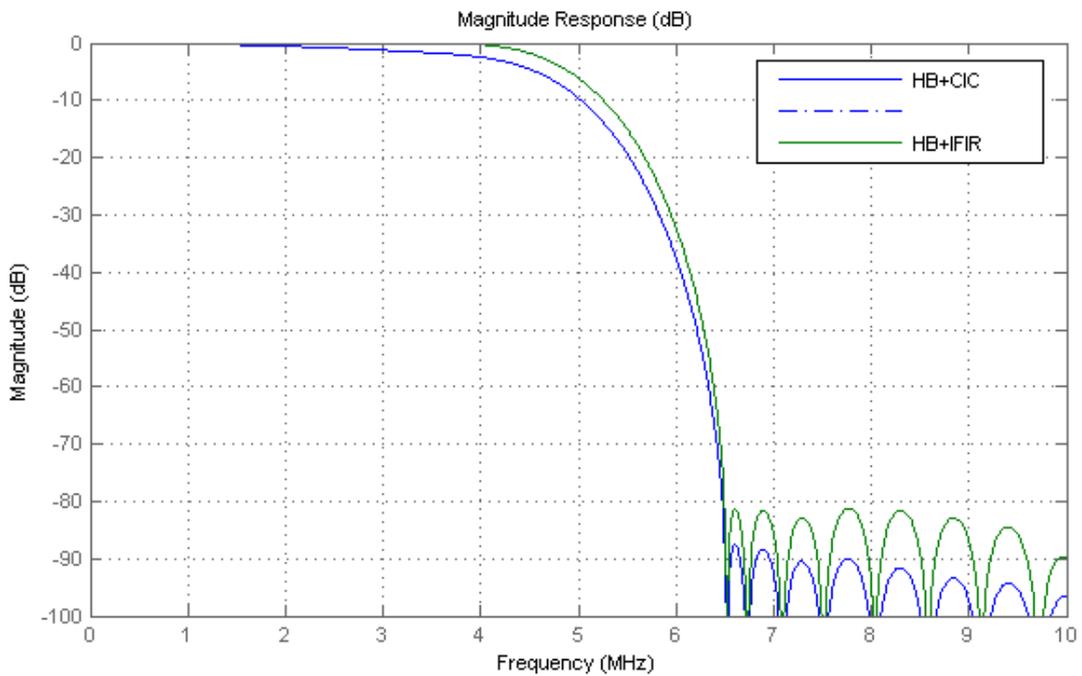


Figure 10: Frequency response of half-band filter+designed IFIR compared to half-band+CIC

## 4 Implementation of the new Interpolation Filter

### 4.1 The Periodic Model Filter

The periodic model filter is of order 78, but has only 40 nonzero coefficients. If the coefficients are  $h[0] \cdots h[39]$  the polyphase decomposition is:

$$\begin{aligned}h_0[n] &= \{h[0], 0, h[5], 0, \dots, h[35], 0\} \\h_1[n] &= \{0, h[3], 0, h[8], \dots, 0, h[38]\} \\h_2[n] &= \{h[1], 0, h[6], 0, \dots, h[36], 0\} \\h_3[n] &= \{0, h[4], 0, h[9], \dots, 0, h[39]\} \\h_4[n] &= \{h[2], 0, h[7], 0, \dots, h[37], 0\}\end{aligned}$$

Because the model filter's impulse response is symmetric ( $h[n] = h[39 - n]$ ),  $h_4[n]$  is symmetric. Also,  $h_0[n]$  is a mirrored version of  $h_3[n]$  and  $h_1[n]$  is a mirrored version of  $h_2[n]$ . In 5 cycles therefore only 20 products have to be calculated (one input sample with all coefficients). This is possible with only 4 multipliers.

#### 4.1.1 Structure

The transposed structure is used, that means that the input is first multiplied by the coefficients and the results are stored in the filter's delay line. Of the 5 polyphase filters one has symmetric coefficients, the other 4 filters pairwise have mirror symmetric coefficients. That means the 2 delay-lines of the regarding polyphase filters can share the products and are just in reversed order to each other. So the number of multiplications in one cycle can be further reduced to a half. The structure is shown in Figure 11.

#### 4.1.2 Bit Widths

The bit widths have to be carefully chosen. If too small either overflow can occur or the SNR degrades (6 dB per bit-loss). The coefficients are quantized to 10 bits, the input data is 18 bit wide. So the result of the multiplication is only 28 bit, saving 8 bit per storage element. Because the absolute summation of the impulse responses of each polyphase filter is less than 1 no overflow can happen and per addition no additional bits are necessary. At the end the most significant 18 bits are selected.

### 4.2 The Masking Filter

The masking filter is implemented without multipliers because of the low complexity. Because it is running on the highest rate it would need 3 multipliers! The quantized coefficients in binary representation are:

$$\begin{aligned}g[0] = g[5] &= 00001001 = 2^{-5} + 2^{-8} \\g[1] = g[4] &= 00101001 = 2^{-3} + 2^{-5} + 2^{-8} \\g[2] = g[3] &= 01001110 = 2^{-2} + 2^{-4} - 2^{-7}\end{aligned}$$

The structure of the masking filter is more or less in direct form and shown in Figure 12.

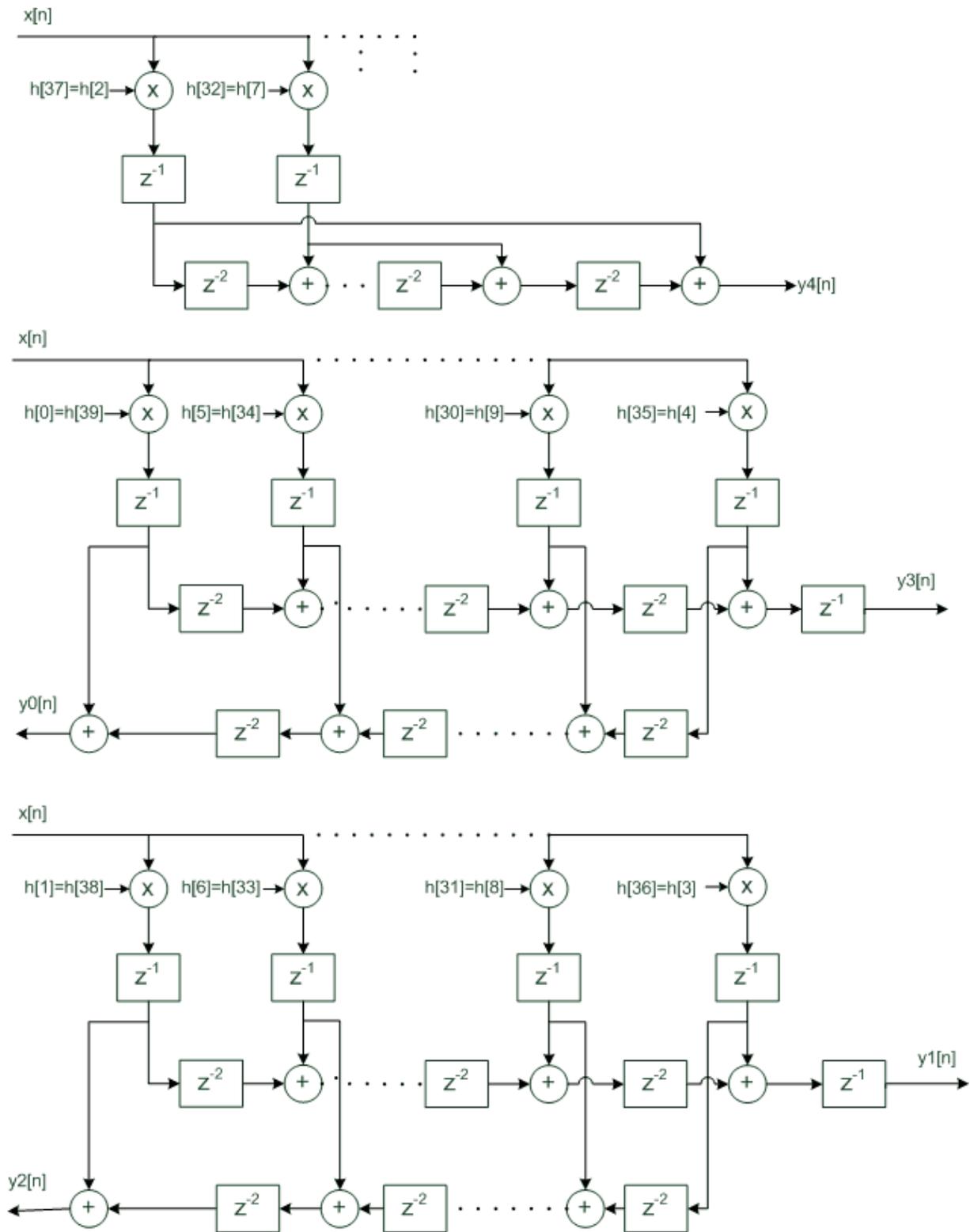


Figure 11: Structure of the periodic model filter. Implemented in Verilog 1:1 except for shared multipliers. Control Logic is missing here, for example multiplexers for multiplier input, enable signals for the registers, multiplexer for selecting one of the 5 outputs.

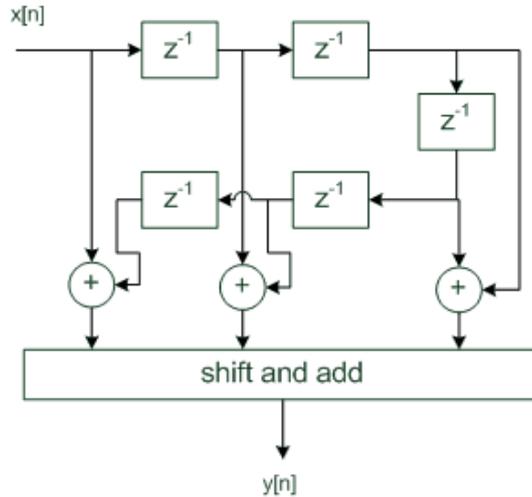


Figure 12: Structure of the masking filter

## 5 Testing

Apart from some trivial testcases (impulse response, random input), one testcase seemed to be important to assure no overflow can happen. Assume  $h[n]$  be the impulse response of an FIR filter. The input signal to maximize the filters output at the time  $N$  is then  $x_{MAX} \cdot \text{sign}(h[N - n])$ , where  $x_{MAX}$  is the maximum number representable and the sign of the input is mirrored to the signs of the impulse response to maximize the convolution sum.

## 6 Results

The results of the measurements with the spectrum analyzer are shown in Figures 13 and 14 for interpolation factor 5 and 10, respectively. The spectral masks are not violated and for interpolation factor 10 a flatter passband is accomplished, as can be seen in Figure 15. The measurements were made at a resolution bandwidth of 100 kHz and a video bandwidth of 30 kHz, as defined in the standard.

The hardware requirements on the FPGA are kept low for arithmetic stuff (multiplications, additions), but are quite high in terms of memory. The implemented filter is of order 83, and that much delay elements are necessary for an FIR filter, regardless of which structure is used. A disadvantage of using the transposed form is that the delay elements have to store more bits of information since the product of a sample with a coefficient has more bits than the sample itself. Using the direct form the number of multipliers would have been doubled because exploiting the mirrored impulse responses of the polyphase components isn't possible in this case.

## 7 Conclusions

The new interpolation filter leads to much better overall frequency responses of the interpolation chain. Therefore the spectral emission is flatter in the band of interest and

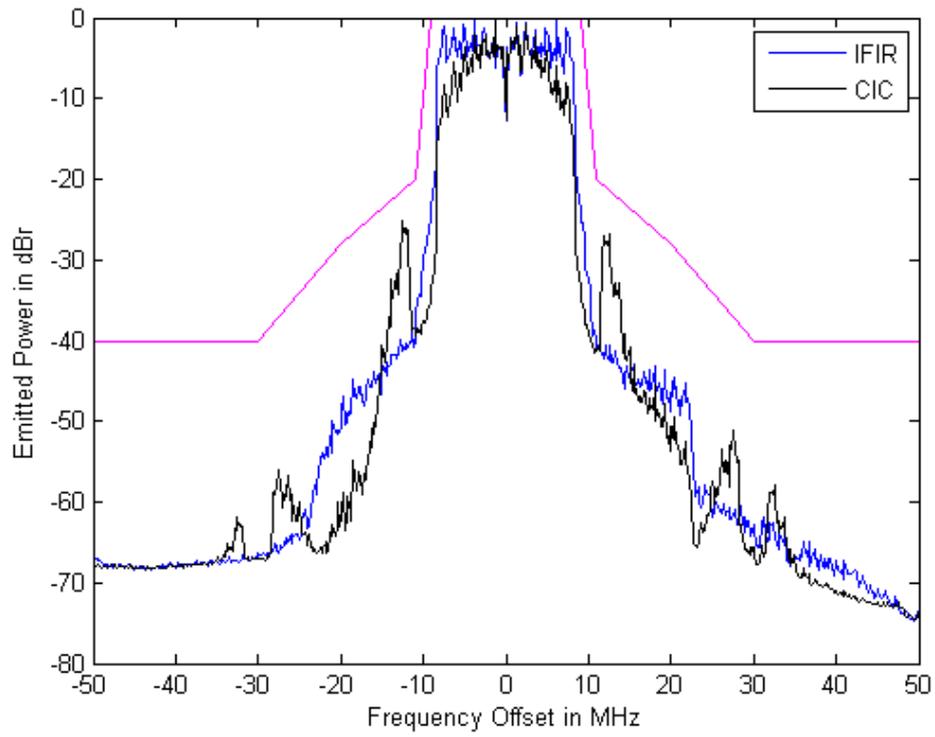


Figure 13: Spectral Power Emission for interpolation factor 5

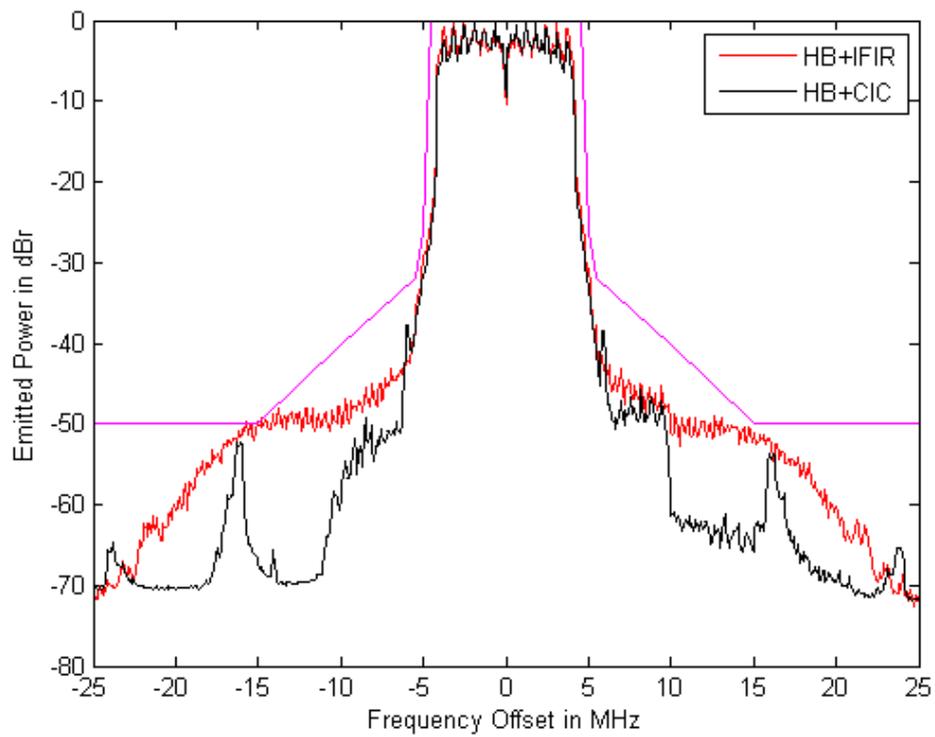


Figure 14: Spectral Power Emission for interpolation factor 10

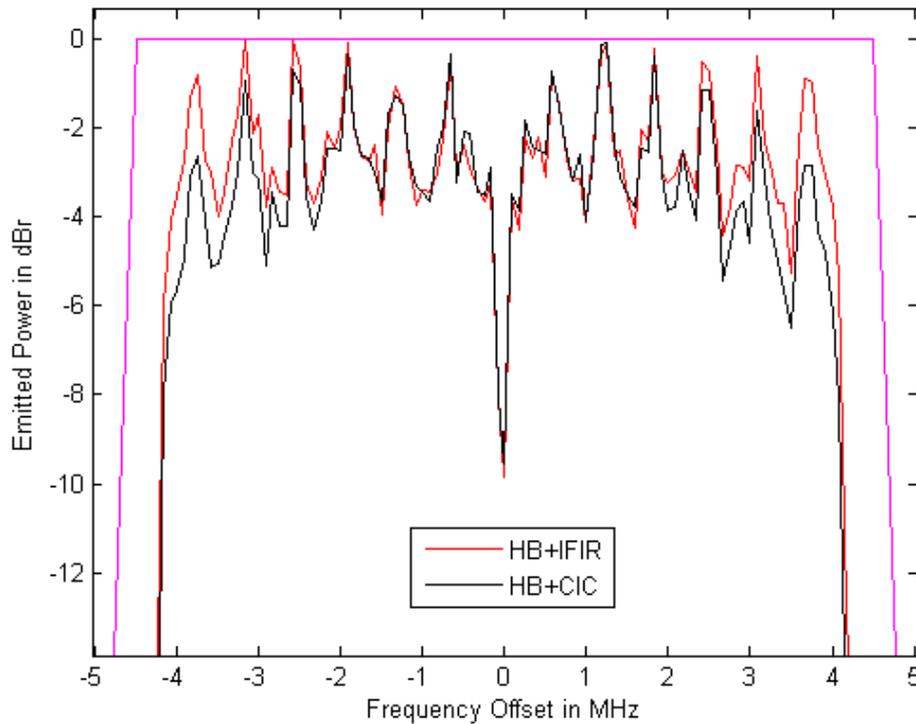


Figure 15: Spectral Power Emission for interpolation factor 10

the images caused by interpolation are better attenuated. The hardware requirements are kept low in terms of multipliers (limited number of entities in the FPGA), but are high in terms of storage elements which are required in high order FIR filters.

## References

- [And] Andrea Costantini, Paul Fuxjaeger, Danilo Valerio, Paolo Castiglione, Giannmarco Zacheo. FTW IEEE802.11a/g/p OFDM Frame Encoder. <https://www.cgran.org/wiki/ftw80211ofdm tx>.
- [And98] Ray Andraka. A survey of CORDIC algorithms for FPGA based computers, 1998.
- [Don00] Matthew P. Donadio. CIC Filter Introduction, 2000.
- [GD04] Oscar Gustafsson and Andrew G. Dempster. On the Use of Multiple Constant Multiplication in Polyphase FIR Filters and Filter Banks, 2004.
- [Joh05] Håkan Johansson. Two Classes of Frequency-Response Masking Linear-Phase FIR Filters for Interpolation and Decimation, 2005.
- [Lim86] Yong Ching Lim. Frequency-Response Masking Approach for the Synthesis of Sharp Linear Phase Digital Filters, 1986.

[SB08] Joëlle Skaf and Stephen P. Boyd. Filter Design With Low Complexity Coefficients, 2008.