Fully Parametrable Downsampler Generator

Lirida A. de B. Naviner¹, Jean-François Naviner¹ and Marcelo Alves de Barros²

¹GET/Télécom Paris - CNRS/LTCI
Département Communications et Electronique
46 rue Barrault, 75013 Paris — France
²UFCG - Departamento de Sistemas de Computação - UFCG/DSC
Av. Aprigio Veloso, 882, 58100 Campina Grande — Brazil
Email: lirida.naviner@enst.fr, naviner@enst.fr, marcelobarros@dsc.ufcg.edu.br

Abstract — This paper deals with customized implementation of downsampling processors. We present an all-programmable generator of downsamplers to synthesize efficient decimator filters for many applications. This tool, written in VHDL language, is able to create cascade of generic FIR/CIC filters and helps fast exploration/evaluation of different hardware solutions for a given problem.

Keywords — Hardware IP, filtering, downsampling processors.

1. INTRODUCTION

Signal resampling is the process of changing the sampling rate of a signal and includes lowpass filtering and downsampling (that is, discarding samples) or upsampling (adding samples). When sampling change factors are integers, we refer to decimation for sampling rate reduction and to interpolation otherwise.

This work deals with decimation process, also named downsampling, which is a necessary task in many applications such as radio transmission [1] sigma-delta demodulation [2], subband coding [3], instrumentation [4], and others. The focus is on customized implementation of downsamplers. Of course, different fields imply different constraints and so optimized solutions come from ad hoc implementations.

Downsampling process is illustrated in Figure 1. Input samples \( x_n \) arrive at input sampling rate or frequency \( f_s = \frac{1}{T_s} \). Outputs of interest are \( y(Mn) \) evaluated at frequency given by \( \frac{1}{M \cdot T_s} \). Downsampling is noted by \( \downarrow M \), where \( M \) is the integer decimation factor. Lowpass filter \( H \) is necessary to avoid aliasing after downsampling. That is, filtering ensures that no signal component exists for all frequency \( f > \frac{f_s}{2} \), according to Nyquist criteria.

Customized implementations of downsampling processors come from ad hoc filter design, structure, and architecture. Concerning design of filters, many papers are found in the literature [5–7]. Also, several filter implementations are reported in [8–10]. This paper presents a generator of downsamplers able to synthesize efficient structures for many applications. This tool has been written in VHDL language and helps fast exploration/evaluation of different hardware solutions for a given problem.

The paper is organized as follows. Section 2 presents the structures considered by the downsampler generator and their computation noise issues. Global downsamplers that can be synthesized with the developed generator are discussed in section 3. Finally, some conclusions are outlined in section 4.

2. SYNTHETIZED STRUCTURES

According to their response, the filters are classified into IIR (Infinite Impulse Response) and FIR (Finite Impulse Response). Many applications require linear phase which are easily obtained with FIR filters only. This work is devoted to this family of filters.

The developed generator considers two kinds of structures for FIR filters. These are described in the next paragraphs.
A. Direct non-recursive filter implementation

Let $H_{FIR}$ be a generic FIR filter whose transfer function is given in (1), where $N$ is the number of taps of the filter.

$$H_{FIR} = \sum_{i=0}^{N-1} h_i z^{-i}$$  \hspace{1cm} (1)

Non-recursive filter implementation of filters such as (1) can be seen as the calculation of an inner product $N$-length vectors $\tilde{h}$ (the impulse response of the filter) and $\tilde{x}$ (the signal data samples):

$$y = \tilde{h} \cdot \tilde{x}$$  \hspace{1cm} (2)

$$\tilde{h} = [h_0 \ h_1 \ \cdots \ h_i \ \cdots \ h_{N-2} \ h_{N-1}]^T$$  \hspace{1cm} (3)

$$\tilde{x} = [x_0 \ x_1 \ \cdots \ x_i \ \cdots \ x_{N-2} \ x_{N-1}]^T$$  \hspace{1cm} (4)

The filter output at $t = n \times T_s$ (named actual output and denoted $y_n$) is given by:

$$y_n = \sum_{i=0}^{N-1} h_i \cdot x_i$$  \hspace{1cm} (5)

where $x_i = x[(n-i) \times T_s]$ and $T_s$ is the sampling period.

The developed filter generator considers direct structure for the inner product calculation, as seen in Figure 2. Due to symmetry peculiar to FIR filters, $h_i$ and $h_{N-1-i}$ are equals and corresponding samples are added before multiplications to save hardware.

In addition, computation power is saved by performing multiplications at the output sampling rate, that is, $f_s/\text{MF}$, where $\text{MF}$ is the decimation factor of the FIR filter.

![Figure 2: FIR filter structure. Example for $N = 5$.](image)

B. Recursive filter implementation

This implementation is considered for special FIR filters such as comb and CIC (comb-integrator-cascade) filters, whose transfer functions are given in (6) and (7), respectively.

$$H_{\text{COMB}} = \frac{1}{M_C} \sum_{i=0}^{M_C-1} z^{-i} = \frac{1}{M_C} \left( \frac{1 - z^{-M_C}}{1 - z^{-1}} \right)$$  \hspace{1cm} (6)

$$H_{\text{CIC}} = \frac{1}{(M_C R)^K} \left( \frac{1 - z^{-M_C R}}{1 - z^{-1}} \right)$$  \hspace{1cm} (7)

For any decimation factor $M_C$ integer, the comb transfer function can be rewritten in such a manner that numerator $(1 - z^{-M_C})$ and denominator $(1 - z^{-1})$ are separated \cite{13}. Denominator is followed by downsampling $(\lfloor M_C \rfloor)$ and numerator becomes $(1 - z^{-1})$. Notice that the denominator part corresponds to an integrator, so carrying to a recursive implementation. Changes in downsampling position and numerator expressions refer to the comutative rule theorem described in \cite{14}.

Frequency response of CIC filter is a generalization of comb frequency response and presents $M_C R$ zeros located at non-null integer multiples of $f = \frac{f_{sa}}{M_C R}$, where $f_{sa}$ is the input sampling rate. Most generally, $R = 1$ but it can be changed in order to adjust frequency response zero locations. Attenuations at zero locations are given by the order of the zeros and is improved by increasing the value of $K$. Efficient CIC filter structure is based on the recursive comb implementation.

CIC structure synthetized with the developed generator consists of $K$ integrators working at input ($f_s$) sampling rate, a decimator by $M_C$, and $K$ differentiators working at output ($f_s/M_C$) sampling rate (see Figure 3).

![Figure 3: CIC filter structure.](image)

C. Computation Noise Issues

The developed filter generator supposes two’s-complement representation for filter coefficients and data samples on $p_h$ and $p_x$ bits, respectively. Therefore, they can be written as:

$$h_i = -h_{i,p_{h-1}} 2^{p_{h-1}} + \sum_{k=0}^{p_{h}-2} h_{i,k} 2^k$$  \hspace{1cm} (8)

$$x_i = -x_{i,p_{x-1}} 2^{p_{x-1}} + \sum_{k=0}^{p_{x}-2} x_{i,k} 2^k$$  \hspace{1cm} (9)

Computation noise relates to insufficient number of bits to represent calculations results. If truncation or rounding on $p_1$ bits is carried out after a calculation, the noise variance associated to the quantization error is given by (10).

$$\sigma^2 = \frac{2^{-2p_1}}{3}$$  \hspace{1cm} (10)
Notice that variance at an adder output is the sum of the variances at adder inputs. So, the effect of quantization on each \( h_i x_i \) is multiplied by \( N \) in a \( N \)-taps filter.

For this reason, it is preferable to quantize only at the output of the filter, that is, after the multidata addition in (1).

Concerning calculations accuracy in comb and CIC filter, the number of bits need to be enough to avoid overflow at accumulators registers. Internal calculations in CIC filter are done with \( b_r = b_{in} + K \log_2(M R) \) bits, where \( b_{in} \) is the input number of bits.

### 3. SYNTHETIZED DECIMATOR FILTER STRUCTURES

#### A. Single stage and multi stages structures

Specification of a filter naturally carries to a single stage structure. Supposing that the length of this unique non-recursive filter is \( N \), the computation power associated to filtering for a decimation by a factor \( M \) can be approximated to

\[
P_c = N \times \frac{f_s}{M} \quad \text{with} \quad f_s = \frac{1}{T_s}
\]

(11)

An alternative structure is based on a cascade of several filters with lengths \( N_1, N_2, \ldots, N_K \) associated to decimator factors \( M_1, M_2, \ldots, M_K \), as illustrated in Figure 4 for \( K = 2 \) [14]. Equivalency of the structures can be obtained if filters lengths respect \( N = N_1 + N_2 M_1 \) which carries to a computation power given by

\[
P_c = N_1 \times \frac{f_s}{M_1} + N_2 \times \frac{f_s}{M_1 M_2}, \quad \text{with} \quad f_s = \frac{1}{T_s}
\]

(12)

Due to the relationship between \( N \) and the length of the filters in the cascade, computation power in (12) is inferior to that in (11). In fact, the more stages there are in the cascade, the shorter the filters in the cascade need to be. The computation power for a cascade of \( K \) filters to decimate an input signal of a factor \( M = M_1 M_2 \cdots M_K \) is:

\[
P_c = \left( \sum_{i=1}^{K} \frac{N_i}{\prod_{j=0}^{i-1} M_j} \right) \times f_s, \quad \text{with} \quad M_0 = 1
\]

(13)

The developed decimator filter generator is able to generate cascade of generic FIR filters or CIC filters as described in (1) and (7). Typical decimation filter chains consist of a cascade with up to four filters [2, 15, 16].

#### B. Downsampler’s parametrable values

The following values can be chosen in order to obtain a customized structure with the developed downsampler generator:

- Number of stages in the CIC structure, \( K \).
- Decimation factor of the CIC structure, \( M_C \).
- Controller of zeroes placements of the CIC response, \( R \).
- Number of taps in the generic FIR filter, \( N \).
- Taps values of the FIR filter, \( h_{i \in [0, N-1]} \).
- Decimation factor of the FIR structure, \( M_F \).
- Accuracy at CIC and FIR outputs, \( b_{CIC} \) and \( b_{FIR} \).

#### C. Example of downsampler

We consider a data acquisition system where digital samples come from an oversampled analog-to-digital converter that supplies 4bit-words at sampling rate \( f_s = 16 \times f_N \).

The downsampler consists of one CIC filter that performs decimation by factor \( M_C = 8 \) and one generic FIR that achieves the decimation process with \( M_F = 2 \). No multiplication is required for CIC filter so making it attractive in the upstream of a decimation cascade chain. A generic FIR is necessary because attenuation given by CIC filter is insufficient and it generates distortion in passband.

Table 1 presents results for the downsampler obtained with the proposed generator. Figure 5 shows frequency response for this decimator filter. Notice that the FIR filter also compensates the droop generated by the CIC filter in passband.

Filter cost for operative parts is less than 2K LC (logic cells) on a commercially available programmable device\(^1\). Internal calculations in CIC filter are done with 16 bits. Taps in FIR filter are coded with 10 bits. Output accuracy of both CIC and FIR filters is 8 bits.

\(^1\)ALTERA Flex10K FPGA Family
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K$</td>
<td>3</td>
</tr>
<tr>
<td>$M_C$</td>
<td>8</td>
</tr>
<tr>
<td>$R$</td>
<td>2</td>
</tr>
<tr>
<td>$N$</td>
<td>43</td>
</tr>
<tr>
<td>$M_F$</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 1: Downsampler parameters.

Figure 5: Frequency decimator filters responses: CIC (in blue), FIR (in green) and global (in red).

4. CONCLUSIONS

This work dealt with hardware downsampling structures generation. The developed VHDL tool is able to create cascade of generic FIR and CIC filters. Efficient structures exploiting decimation and properties of these filters are considered by the downsampler generator. The developed filter generator helps fast exploration/evaluation of different hardware solutions for a given problem.

REFERENCES


