Digital Filter Design using VHDL

Anatoliy Sergiyenko[1], Anastasia Serhienko[1]

[1] *Computer Engineering Dept. of Sikorsky Kyiv Polytechnical Institute, 37 Peremogy ave., Kyiv, Ukraine*

aser@comsys.kpi.ua, ananserr@yahoo.com

Abstract. *In this paper a method is proposed, which consists in integer searching for the filter coefficients, forming the filter structure and modeling it. The use of the VHDL language in all the steps of the filter design helps to speed-up and improve the filter optimization. Examples of the multiplierless IIR filter design show the method effectiveness.*

Keywords

VHDL, FPGA, IIR filter, allpass filter.

1 Introduction

A traditional method of the digital filter design for the field programmable gate array (FPGA) consists in the implementation of the following steps. A set of filter coefficients is searched using the proper CAD tool like Matlab. Then, the coefficients are quantized with the frequency response proving. The rounded coefficients are built in the filter IP core, provided by the FPGA supplier or in the filter model, described by VHDL or Verilog. Finally, the filter model is tested using the proper testbench before and after synthesis [1]. The separate and costly program like Matlab is of demand to implement this method. Another disadvantage is the need of checking the frequency response after each coefficient quantization. Besides, the filter structure, and the rounding scheme infer this frequency response.

In this paper the VHDL language is proposed both for the filter structure description and modeling, and for the coefficient searching, frequency response calculating. This approach is proven by the multiplierless infinite impulse response (IIR) filter synthesis.

2 Properties of VHDL used for the digital filter design

The VHDL language is usually used to describe the application specific digital structure for FPGA. But its ability of the mathematical data processing is undervalued. The library IEEE contains the MATH_REAL and MATH_COMPLEX packages. They consist of floating point types, constants, and functions, which are suitable for the complex number processing. These packages have many features to explore algorithms for digital signal processing. Also, effective routines for solving linear equation systems and discrete Fourier transform are based on these packages. Thus, the possibilities of VHDL language of mathematical processing and simulation are approaching to ones of Matlab [2,3].

The complex variable $Z = e^{j\varphi}$ is usually used for the digital filter analyzing. To get it in VHDL, the following function can be used:

```
function Z(fi: real) return COMPLEX_POLAR is begin
  return exp(COMPLEX_TO_POLAR(MATH_CBASE_J)*fi );
end Z;
```

Then, the transfer function, for example, of the low pass filter

$$H(z) = z^{-1} + \frac{a + a(1+b)z^{-1} + z^{-2}}{1 + a(1+b)z^{-1} + az^{-2}}; \quad (1)$$

is described by the function:

```
function LPF(a, b, fi: real) return COMPLEX_POLAR is
begin
       return Z(-fi) +
   (COMPLEX_TO_POLAR(COMPLEX'(a,0.0)) + a(1+b)*Z(-1.0*fi) + Z(-2.0*fi))/
   (COMPLEX_TO_POLAR(COMPLEX'(1.0,0.0)) + a(1+b)*Z(-1.0*fi) + a*Z(-2.0*fi));
 end LPF;
```

The coefficients a and b can be calculated using the equations [4]:

$$a = \frac{1 - \operatorname{tg} d_f}{1 + \operatorname{tg} d_f};\ b = -\cos f_C \cdot (1 + a),$$

where d_f is the transition band, rad, f_C is passband, rad. The quantized coefficients are adjusted using scanning their values in some circle of a and b convergence. The filter coefficients are selected, which optimize the difference of the function (1) and the filter specification. It is possible to calculate the transfer function diagram in the frequency space using the following process operator, which generates the waveforms of the magnitude, logarithm magnitude and phase frequency responses in the screen of the VHDL-simulator for a thousand clock cycles.

```
process(CLK)
       variable p, phas: real :=0.0;
       variable Hz: COMPLEX_POLAR;
begin
        a <= 1.5;  b <= 0.64;                -- filter coefficients
        if CLK='1' and CLK'event then
               phas := phas + 0.001; -- phase (frequency)counter
               p := phas * MATH_PI * 2.0;  -- normalized phase
               ph <= phas;                   -- frequency signal
        end if;
        Hz := LPF(a, b, p);                      --  H(z)
        Mag <= abs(Hz);                       -- magnitude of H(z)
        Phase <= Hz.ARG;                      -- phase of  H(z)
        Logm <= 20.0*log10(abs(Hz));     -- H(z) in decibels
end process;
```

The derived filter coefficients are put in the filter model described by VHDL. This model can be tested before and after the synthesis using the testbench, available at [5]. By this testing, the inphase REO and quadrature IMO components of the analytical signal are fed to the inputs of two instances of the filter (see Fig.1). The results of the modeling are the magnitude and phase responses, which are outputted in the waveform window of the VHDL simulator.

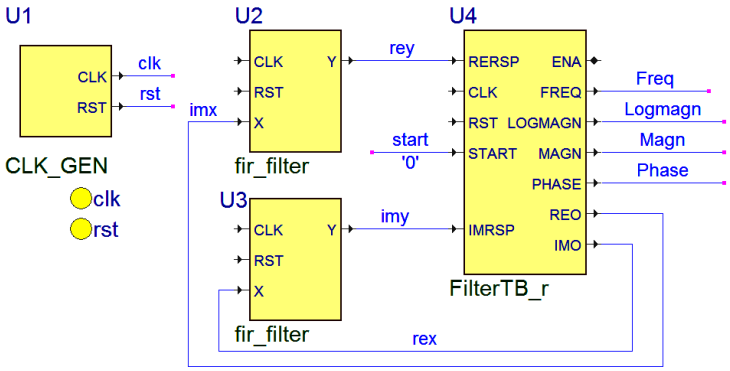

Fig.1. Testbench for the filter response computation

3 Multiplierless IIR filters

IIR filters provide less complexity and higher filtering effectiveness comparing to the finite impulse response (FIR) filters. But they have limited use in the FPGA systems because of the abridged throughput. The speed of IIR filters is limited by the critical path length, which is estimated by the delay in the filter feedback. This delay could not be minimized by the pipelining technique, which is usually used for the FIR filters [1].

One of the effective methods to speed-up the IIR filter in FPGA is substituting the hardware multipliers to a set of adders, which add the shifted multiplicands [1,6]. The modern FPGAs contain the 6-input LUTs, which provide a one-stage network of the three-input adder [7]. In this situation, it is preferable to represent the filter coefficients as the rational numbers in the canonical binary number system:

$$c = k2^p + l2^q + m2^r, \tag{2}$$

where p, q, r are integers, $k,l,m \in \{0, 1, -1\}$.

Due to the usual method, the real coefficient values are derived. Then the truncated values (2) of them are searched in the vicinity of the solution point, which provide the optimum transfer function. This search is performed by the scanning method [8] or the evaluation optimization [6,9,10]. The multiplierless IIR filters, based on the allpass filter like (1), have the minimized number of the coefficients. This filter has the transfer function

$$H(z) = (A_1(z) + A_2(z))/2, \tag{3}$$

where $A_1(z)$, $A_2(z)$ are transfer functions of the allpass filters. These filters are composed of the sections:

$$H_i(z) = \frac{b_i + c_i z^{-1} + z^{-2}}{1 + c_i z^{-1} + b_i z^{-2}}. \quad (4)$$

Such filters are stable and immune to small variations of the coefficients. The last feature provides the successful searching for the coefficients in the form (2). Moreover, in [10] a Stoyanov-Kawamata filter structure is utilized, which provides the minimum number of elements in the form (2). A section calculating the function (3) with the maximum speed is described by the signal flow graph, which is illustrated in the Fig. 2. Here, the bars represent the register delays, circles represent the adders and coefficient multipliers.

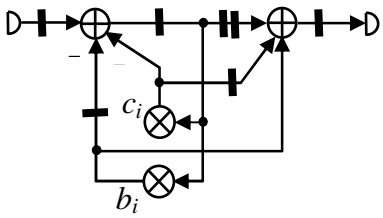

Fig.2. Signal flow graph for the function (4)

The Fig.2 analysis shows that the respective network has only two multiply units, is fully pipelined, the critical path goes through an adder and a multiplier to the coefficient b_i. Therefore, the most performance is achieved when the coefficient b_i has the minimum terms in (2) or even is equal to zero.

The coefficients in the form (2) can be searched using the VHDL program as well. The iteration of such a program consists in the implementation of the following steps. Firstly, a set of the coefficients b_i, c_i is selected from a table of coefficients, which are equal to (2). Then these coefficients are put in the formula (3), which is evaluated using the process operator shown in section 2. Finally, the results are compared with the filter specification, and the effective coefficient set is stored. After several iterations of this algorithm, the optimum set of the filter coefficients is selected.

4 Experimental results

The method described above was used to build a set of IIR filters of the order from 5 to 9. These filters were put in the database of the Web tool IIR Filter Generator [11]. This application generates the synthesizable VHDL model of a filter with the given bit width, stop band frequency. This filter can be the low pass (LP), high pass or half band (HB) filter. Up to 27 models of the HB filters with different frequency responses can be generated by this Web tool. Combining them, the filter with excellent characteristics can be built. In Fig. 3 the frequency responses H_1, H_2 of two HB filters and their cascaded connection $H_P = H_1 H_2$ are shown. These diagrams are derived by the testbench in Fig.1. The parameters of the synthesized HB filter are shown in Table 1. Here, the hardware volume is given in the configurable logic block slices (CLBS). The high clock frequency of this filter is derived due to the fact, that for this filter the coefficients $c_i = 0$. As a result, the structure becomes highly pipelined and has small hardware volume.

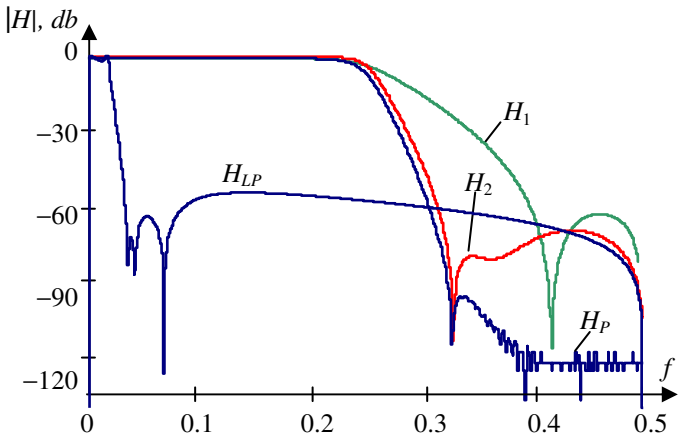

Fig.3. Frequency responses of synthesized HP and LP filters

In Table 1, the results of the synthesis of another HB filter are given. This filter, as well as another filter for comparison, are designed and synthesized on the base of their signal flow graphs and coefficients, given in the

references. This filter is one-staged, but it has up to 6 summands in the coefficient representation (2). As a result, the critical path becomes too long, and the maximum frequency is in 6.4 times lower than for proposed filter.

Tab. 1. Parameters of filters configured in Xilinx Kintex FPGA

Filter	Hardware,CLBS	Max. clock frequency, MHz	Suppression, db	Reference
HB	203	690	120	
HB	441	107	106	[12]
LP	179	310	54	
LP	203	189	57	[10]

Another example is the synthesis of a LP filter with the cut frequency of $0.025f_S$, where f_S is the sampling frequency. The filter structure corresponds to (3), where $A_1(z)$ and $A_2(z)$ are transfer functions of 3-d and 4-th order, respectively. The coefficients found by the VHDL program are equal to

$c_0 = -1.00\bar{1}01$; $c_1 = -10.0000\bar{1}$; $c_2 = -10.00\bar{1}$; $c_3 = -10.0000\bar{1}00\bar{1}$.
$b_1 = 1.00\bar{1}01$; $b_2 = 1.00\bar{1}0\bar{1}$; $b_3 = 1.0000\bar{1}01$;

The resulting filter transfer function H_{LP} is shown in Fig. 3, and its characteristics are given in Table1. This filter has less hardware volume and much higher frequency by the approximately equal suppression level, comparing to the analogous filter shown in [10]. This is explained by the fact that the signal flow graph of the analogous filter has much longer critical path due to the complex Stoyanov-Kawamata scheme implemented in.

5 Conclusion

In this paper it is shown that the VHDL language allows for the digital filter design, without going beyond the VHDL editor and simulator. Thus the possibility of rapid modeling of complex optimization processes provides the effective search for optimal structural filter solutions. Unlike the use of common design tools like Matlab, the VHDL simulator provides the filter operation control, taking into account both coefficient and data bit width and method of arithmetic operations, as well as features of its implementation in FPGA. It was proven, that if the coefficients of the multiplierless filter have no more than three summands in their representation, then its pipelined implementation in FPGA has the highest clock frequency. The examples of the IIR miltiplierless filter design show the effectiveness of VHDL use.

References

[1] S. A. Khan, *Digital Design of Signal Processing Systems. A Practical Approach.* UK: Wiley, 2011.
[2] P. J. Ashenden and J. Lewis, *The Designer's Guide to VHDL*. Morgan Kaufmann, 2008.
[3] A. M. Sergiyenko, "VHDL contra Matlab" *Argc&Argv. Programming / Information Technologies / Standards*. vol. 58, no. 1. pp. 40–46, 2005, in Russian.
[4] P. P. Vaidyanathan, P. Regalia and S.K. Mitra, "The Digital All-Pass Filter: A Versatile Signal Processing Building Block" *Proc. IEEE.* vol. 76. no. 1. pp. 19–37. Jan. 1988.
[5] A. Sergiyenko. *Testbench for the filter testing*. Kiev: NTUU "KPI", 2012. [Online]. Available: http://kanyevsky.kpi.ua/VHDLlabenglish/usefulcores_VHDLlab.html
[6] L. D. Milic and M. D. Lutovac, "Design of multiplierless elliptic IIR filters with a small quantization error" *IEEE Trans. on signal processing.* vol. 47, no. 2, pp. 469–479. Feb. 1999.
[7] S. Churiwala, Ed. *Designing with Xilinx FPGAs Using Vivado*, Springer. 2017.
[8] A. T. Mingasin, "Digital filter synthesis for the high speed system on the chip" *Digital Signal Processing*. no. 2. pp. 14 – 23. 2004, in Russian.
[9] S.-T. Pan, "CSD-Coded Genetic Algorithm on Robustly Stable Multiplierless IIR Filter Design" *Mathematical Problems in Engineering,* Hindawi Publ, vol. 2012, Article ID560650, 15 P.
[10] V. I. Anzova, J. Yli-Kaakinen, T. Saramaeki, "An Algorithm for the Design of Multiplierless IIR Filters as a Parallel Connection of Two All-Pass Filters". *IEEE Asia Pacific Conf. on Circuits and Systems, APCCAS.* 2006. pp. 744-747.
[11] A. Sergiyenko. *VHDL design of multiplier-free IIR filters*. Kiev: NTUU "KPI", 2016. [Online]. Available: http://kanyevsky.kpi.ua/GEN_MODUL/index_eng.php
[12] K. S. Yeung, S.C. Chan, "The Design and Multiplier-Less Realization of Software Radio Receivers With Reduced System Delay". *IEEE Trans. On Circuits and Systems – Regular Papers,* vol. 51, pp. 2444–2449. Dec. 2004.