

FPGA Realization of Farrow Structure for Sampling Rate Change

Bogdan Marković^{1,2}, Jelena Čertić²

Abstract: In numerous implementations of modern telecommunications and digital audio systems there is a need for sampling rate change of the system input signal. When the relation between signal input and output sampling frequencies is a fraction of two large integer numbers, Lagrange interpolation based on Farrow structure can be used for the efficient realization of the resample block. This paper highlights efficient realization and estimation of necessary resources for polynomial cubic Lagrange interpolation in the case of the demand for the signal sampling rate change with the factor 160/147 on Field-Programmable Gate Array architecture (FPGA).

Keywords: Farrow structure, Lagrange interpolation, Finite Impulse Response (FIR), Field-Programmable Gate Array (FPGA), Sampling Rate Change (SCR), implementation, Estimation of necessary resources.

1 Introduction

In modern telecommunications and audio systems there is sometimes a need to design a system as a multirate one. It means that a sampling frequency changes in the signal processing chain. In the case when sampling frequency increases L times, with L as an integer factor; the resampling is realized as an interpolator that consists of the cascaded connection of the upsampler and interpolation filter. For the M times decreasing of the sampling frequency a decimator that consist of antialiasing filter and downsampler.

The sampling frequency change factor can be a fractional number $p = L/M$, where L and M are relatively prime numbers. In the case of fractional value of sampling frequency change factor the resampling structure can be realized as a cascaded connection of an interpolator and a decimator [1]. Interpolation and decimation factors are L and M , respectively. This structure can be additionally optimized by replacing the cascaded connection of the

¹Bitgear Wireless Design Services LLC, Stevana Markovića 8, 11080 Zemun, Belgrade, Serbia;

E-mail: bogdan.markovic@bitgear.rs

²University of Belgrade, School of Electrical Engineering, Bulevar Kralja Aleksandra 73, 11020 Belgrade, Serbia; E-mail: ceretic@etf.rs

interpolation and the decimation filters by a single filter with the cut-off frequency ω_s ,

$$\omega_s = \min\left(\frac{\pi}{M}, \frac{\pi}{L}\right). \quad (1)$$

However, in the case when L and M are large numbers, this type of resampling structure is not suitable for practical realizations. Interpolation and decimation with large factors (L and M) require very sharp filters and a lot of processing power for implementation. Therefore, in the case of large L and M , it is usual to realize resampling part of the system in a different manner. Especially interesting class of filters used for the resampling are the filters based on Lagrange polynomial interpolations. This type of filters can be implemented efficiently by the usage of polynomial cubic Farrow structure [2]. In this paper, we focus on the realization based on the basic Farrow structure. Initial results of the proposed realization structures are presented in [3]. We chose this structure because it was introduced as an efficient design for the case of resampling of the audio signals from *Compact Disc* CD to *Digital Audio Tape* DAT standard [4]. However, the application of the Farrow structure is not limited to audio systems. In modern telecommunications systems, multirate concept is often needed. It is usual to realize resampling as a multistage structure. For example, a CIC filter can be used for the first step of the decimation and then "fine tuning" of the resulting sampling frequency can be achieved by the Farrow structure. Apart from this one, there are many other usages in the implementation of timing recovery solution for QAM [5] and DVB-T systems [6], in echo cancellation in digital modems and equalization in WiMAX and GSM communication systems [7].

The rest of the paper is organized as follows. The basics of Farrow structure based on cubic Lagrange interpolation are given in the Section 2 of this paper. In the Section 3 the efficient realization of sampling rate change for FPGA platform is presented. We present detailed realization for a special case - the conversion of frequency sampling rate from 44.1 kHz to 48 kHz, i.e. with the factor 160/147. The results of implementation and estimation of necessary resources are given in Section 4.

2 Lagrange Interpolation and Farrow Structure

It is already well known in mathematics that a function which passes through $N+1$ of given points can be described by the usage of Lagrange interpolation, i.e. polynomial of order N . Lagrange interpolator is very simply realized in the direct form of FIR filters which are connected in a cascade, which makes it very suitable for the implementation on FPGA platforms. In practice, cubic Lagrange interpolator is mostly used. For that reason, in this

paper we present the realization based on the cubic Lagrange interpolator. Lagrange interpolators of fourth and higher orders can be realized as well.

Lagrange interpolation is precise on low frequencies and it never overestimates signal amplitude when the delay (D) is in interval $(N-1)/2 \leq D \leq (N+1)/2$. If we take into account that most of the real-world signals are lowpass and that the amplitude response of interpolator is one, for the delay inside of the above mentioned interval, it makes it very suitable for applications of sampling rate change [8].

Beginning from hybrid analogue - digital model for signal interpolation and decimation, given in Fig. 1, output signal $y[l]$ can be presented in the following form:

$$y[l] = y_c(t_l) = \sum_{k=-N/2}^{N/2-1} x[n_l - k] h_c((k + \mu_l)T_x), \quad (2)$$

where $x[n]$ is sample of input signal, and T_x is the input signal sampling period. In the previous equation it is assumed that sample μ_l is central sample of interval $-NT_x/2 \leq t \leq NT_x/2 - T_x$. In the case of the Lagrange polynomial interpolation, impulse response, $h_c(t)$ can be presented in the following form:

$$h_c((k + \mu_l)T_x) = \sum_{m=0}^M c_m(k) \mu_l^m, \quad (3)$$

where $c_0(k), c_1(k), \dots, c_M(k)$ are coefficients, with $M \leq N-1$ order of polynomial function depending on μ , while k can take values from interval $-N/2, -N/2+1, \dots, N/2-1$, for N even. The form of the output signal of hybrid analogue digital model is given in the following form:

$$y[l] = \sum_{m=0}^M \mu_l^m \left(\sum_{k=-N/2}^{N/2-1} c_m(k) x[n_l - k] \right), \quad (4)$$

with values of μ_l given with:

$$\mu_l = \frac{lT_y}{T_x} - \left\lfloor \frac{lT_y}{T_x} \right\rfloor \in [0,1), \quad (5)$$

where $\lfloor \cdot \rfloor$ defines a floor function, and where T_y stands for the sampling period of the output signal. Coefficients $c_m(k)$ are independent of parameter μ . Based on the equation (5), general block diagram of Farrow structure for interpolation is constructed and given in Fig. 2 [9].

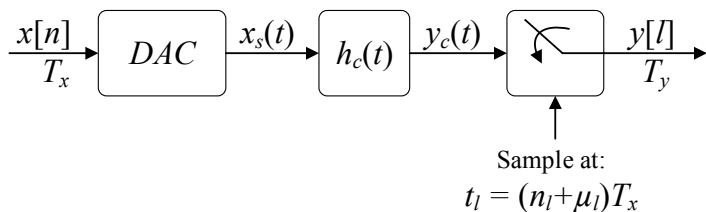


Fig. 1 – Hybrid analogue- digital model of signal interpolation and decimation.

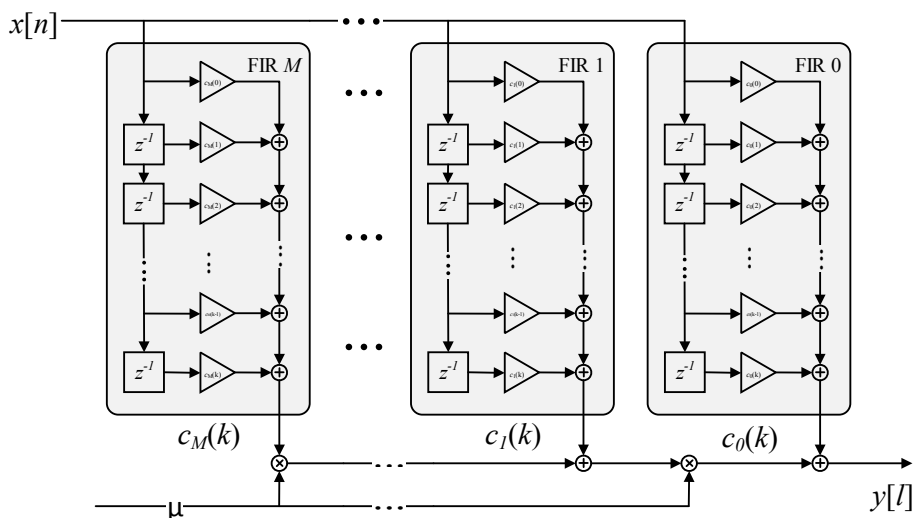


Fig. 2 – General block diagram of Farrow structure in the case when Lagrange polynomial interpolation is the order of M .

Table 1
FIR filters coefficients inside Farrow structure interpolators.

k	$m = 0$	$m = 1$	$m = 2$	$m = 3$
-2	0	1/6	0	1/6
-1	0	1	1/2	-1/2
0	1	-1/2	-1	1/2
1	0	-1/3	1/2	-1/6

The advantage of such an approach lays in the fact that filter coefficients do not change. The only parameter that changes its value is μ . This structure is

very suitable for the FPGA implementation. Coefficients of Farrow structure filters are given in **Table 1**.

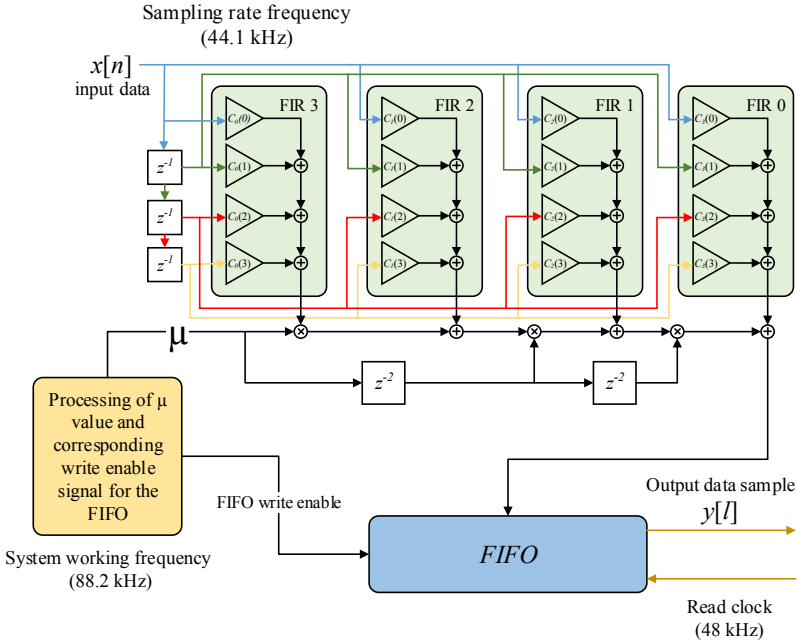


Fig. 3 – Block diagram of realized Farrow interpolation structure by the usage of cubic Lagrange interpolator.

3 Realization of Farrow Structure on FPGA Based Platform

The realization is given on the example of Xilinx Spartan 3A-DSP chip. Block diagram is presented in Fig. 3. For the realization of Farrow structure, we have considered the example in which input signal frequency sampling rate is 44.1kHz. Due to system efficiency, the working frequency of the design is 88.2 kHz, i.e. twice higher than sampling rate. In that way, resource time sharing inside FPGA chip is enabled, as there are two available clocks for the processing of one sample of input signal. Filters realization for sampling change rate can be divided into three parts described in the following subsections.

3.1 Processing of μ value and corresponding write enable signal for the FIFO

Block diagram of the realized block “processing of μ value and the corresponding FIFO write enable signal” is shown in Fig. 4. In this specific case, when sampling rate change of input signal is 160/147, the (5) can be written in recursive form:

$$\begin{aligned}\mu_1 &= 147/160, \\ \mu_n &= \mu_{n-1} + 147/160 - 1.\end{aligned}\tag{6}$$

Based on the (6), the values for μ are generated by simple accumulator whose initial value is set to $T_x/T_y = 147/160$, i.e. $\mu_1 = 0.91875$. The parameter μ value is calculated for every second clock, with exception of the cases when accumulator output is less than one. In the case when the accumulator output is less than one, we need both clocks for the same sample of input signal $x[n]$, as it refers to the interpolation of the input signal.

Accumulator enable signal is generated by one bit counter which works on 88.2kHz (the value of the counter is set to 1 at every second clock), which is connected to comparator output of accumulator output and 1 by logical “or” circuit.

The scheme also includes the block entitled “*remove integer part of number*” in which integer part of accumulator output is cut-off (floor function). This block is realized by simple cut-off of the highest bit of accumulator output.

For each valid value of μ parameter, a suitable write enable signal is generated for FIFO (First In First Out) memory, which is at the same time the enable signal for the work of FIR filters. The value of the control signal is generated as delayed version of the accumulator enable signal.

3.2 Filtering of the input signal through Farrow structure

Based on the block scheme presented in Fig. 2, Farrow structure for cubic Lagrange interpolator is realized, where filter coefficients are given in **Table 1**. In order to get efficient realization of the above mentioned structure, each coefficient from the table is multiplied by 6, so that multiplication is used instead of division. This approach is suitable for the FPGA architecture in which there are already hardware realized multipliers. In that manner, coefficients $c_m(k)$ take values from a set of numbers $\{0, -1, 1, -2, 3, -3, 6\}$. As those are small integer numbers, the multiplying function can be replaced by summation function and logical bit shifting. It is also possible to use a certain level of coefficients symmetry for $m = 2$ and $m = 3$, so multiplying function with number $3(1/2)$ can be implemented only once, while the result with the inverter is changed into multiplying result with $-3(-1/2)$. The example of implementation is shown in Fig. 5 for FIR filter 0, with coefficients $\{1/6, -1/2, 1/2, -1/6\}$, that is $\{1, -3, 3, -1\}$ in the presented realization.

Furthermore, in comparison to the scheme shown in Fig. 2, it is possible to additionally save some resources by removing delay elements in front of the

structure for input signal filtering, as presented in Fig. 3. Each line of the input signal with different delay is marked by different color.

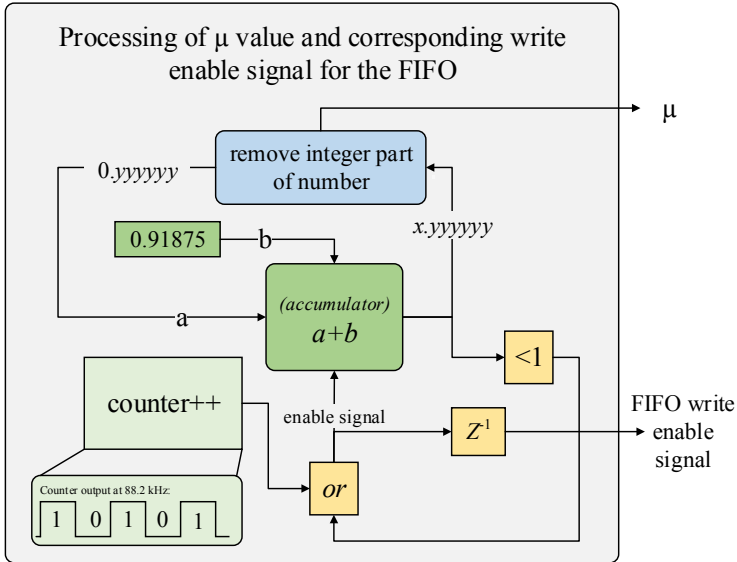


Fig. 4 – Realization of block ‘Creating μ value and FIFO write enable control’.

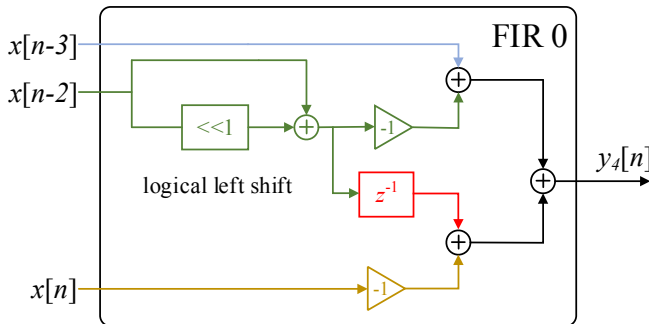


Fig. 5 – Block diagram implementation of FIR 0 by using the feature that filter coefficients are small integer numbers, and the feature of coefficient symmetry.

In the end, it is necessary to multiply filter outputs with appropriate μ values and to sum them. This part of structure is implemented by DSP48 blocks, which are hardware implemented multipliers inside Xilinx Spartan 3A- DSP chip. The blocks are used so that they work in the regime of multipliers and

adder with additional port, i.e. $P = C + A \times B$. In this way, only three multipliers were necessary for the realization of the whole structure. The delay of the multiplying operations results via DSP48 block, which includes two clocks, is compensated by the delay of the appropriate μ value for the same amount of clocks, for each FIR filter output.

3.3 FIFO memory

In order to enable signal transition from one clock domain into another one, it is necessary to implement FIFO memory with independent read and write clock. On the write clock domain side, the data are written on the appropriate control enable signal with the frequency of 88.2 kHz, while they are continuously read with the frequency of 48 kHz. Fig. 3 shows FIFO memory in blue color.

4 Results

The shown implementation is tested for input test signal which is the music audio data captured with 44.1kHz sampling rate. Spectra and time frames of input and output signals are shown in Figs. 6 and 7. As described in Section 3, additional structure efficiency is achieved by integer filter coefficients, resulting in gain of 6 in the output signal. This gain can, if necessary, be compensated by additional multiplier. Figs. 6 and 7 show output signal as 6 times weaker in order to be compared to the input test signal more easily. Based on both pictures, it is clear that the presented implementation has given the expected result. In other words, information in time and frequency domains are preserved.

Table 2 presents the estimations of the necessary resources for the implementation of Farrow structure for sampling rate change on Xilinx Spartan 3A-DSP 3SD3400ACS484-4. As a tool for design development, Xilinx System Generator 13.4 was used within Matlab Simulink.

Table 2
Estimation of necessary resources for the implementation of Farrow structure on Xilinx Spartan 3A-DSP FPA.

	necessary	available	Used [%]
Slices	550	47744	1%
FFs	948	47744	1%
LUTs	621	23872	2%
Block RAMs	0	126	0%
DSP48s	3	126	2%

Only three multipliers were used for structure implementation, which stands for 2% of the overall number of DSP48 blocks. Block RAM memory was not used. The overall number of LUT (Look-Up Table) necessary for filter realization is 2% of the overall number of available resources.

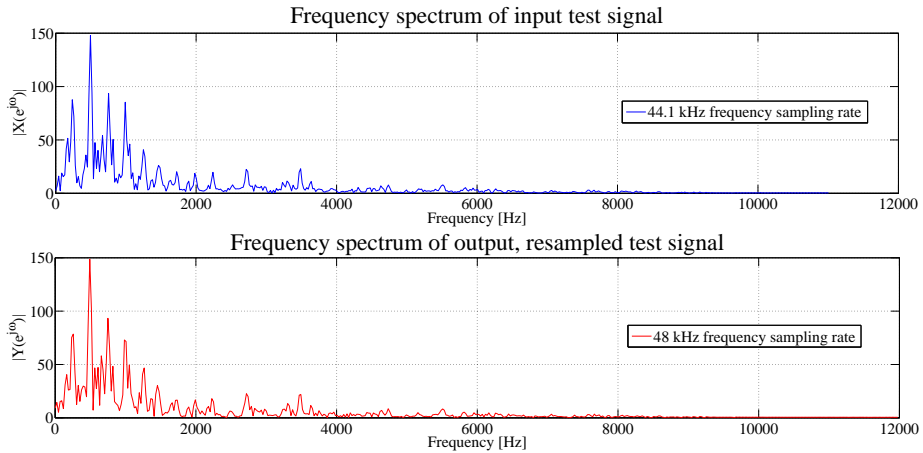


Fig. 6 – Frequency spectrum of input test signal before sampling rate change (picture above, blue color) and frequency spectrum output signal (picture below, red color).

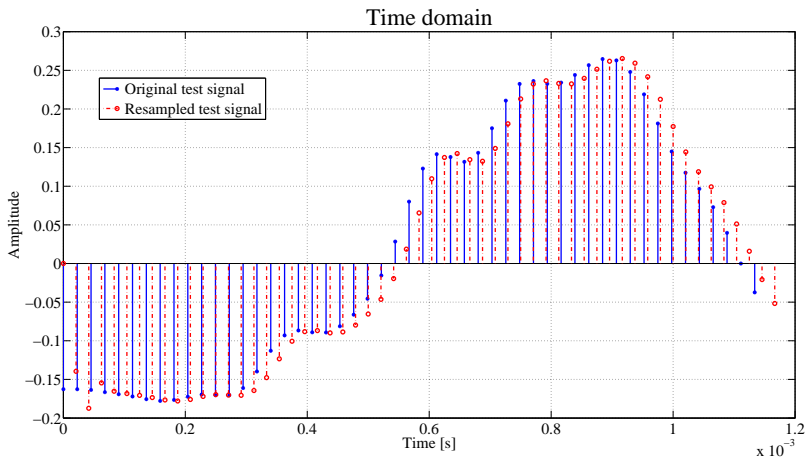


Fig. 7 – Timeframe of input test signal before sampling rate change (blue color) and timeframe of output signal after sampling rate change (red color).

5 Conclusion

This paper shows one efficient solution to the implementation of basic Farrow structure on FPGA platform. The structure is implemented for frequency sampling rate change of the audio signal, i.e. for the conversion of audio format between CD and DAT standards. Frequency sampling rate of the input signal is 44.1kHz, and frequency sampling rate of the output signal is 48kHz. The problem of existence of two different sampling frequencies, i.e. two different sampling periods, is solved by the usage of FIFO memory. FIR filters are realized without multipliers, so the overall number of necessary multipliers is reduced. Additional saving is achieved by using mutual delay line for all filters.

In existing telecommunications and digital audio systems, sampling rate change represents only a small part of a complex system. For that reason, it is desirable that it uses the least possible amount of available resources. The suggested implementation of Farrow structure for sampling rate change, shown in this paper, can easily be adjusted for the usage in other telecommunications or digital audio systems, for example 8VSB ATSC signal demodulator for solving timing recovery issues. If we take into account that filter coefficients, and their structure, remain unchanged for various relations between input and output sampling rate, the given structure is simply adapted by modification of the block 'processing of μ value and the corresponding FIFO write enable signal' in the specific case, by simple change of the initial value of accumulator.

6 Acknowledgement

This work was partially supported by the Ministry of Education and Science of Serbia under Grants TR-32023 and TR-32028.

7 References

- [1] Lj. Milic: *Multirate Filtering for Digital Signal Processing: MATLAB Applications*, Information Science Reference, Hershey, PA, USA, 2009
- [2] C.W. Farrow: *A Continuously Variable Digital Delay Element*, International Symposium on Circuits and Systems, Espoo, Finland, 07- 09 June 1988, Vol. 3, pp. 2641 – 2645.
- [3] B. Marković, J. Čertić: *Efficient Realization of Farrow Structure for Sampling Rate Change on FPGA Platform*, 2nd International Conference on Electrical, Electronic and Computing Engineering IcETRAN, Silver Lake, Serbia, 08-11 June 2015, pp. EK11.3. 1 – 5.
- [4] K. Rajamani, Y.S. Lai, C.W. Farrow: *An Efficient Algorithm for Sample Rate Conversion from CD to DAT*, IEEE Signal Processing Letters, Vol. 7, No. 10, Oct. 2000, pp. 288 – 290.
- [5] H. Meyr, M. Moeneclaey, S. Fechtel: *Digital Communication Receivers – Synchronization, Channel Estimation and Signal Processing*, John Wiley and Sons, New York, NY, USA, 1998.
- [6] C.L. Wey, S.Y. Lin, T.H. Tsai, M.T. Shiue: *Efficient Implementation of Interpolation Technique for Symbol Timing Recovery*, WSEAS International Conference on Computer Engineering and Applications, Gold Coast, Australia, 17-19 Jan. 2007, pp. 13 – 17.

FPGA Realization of Farrow Structure for Sampling Rate Change

- [7] R. Ratan, S. Sharma, A.K. Kohli: Effect of Compensation and Arbitrary Sampling in interpolators for Different Wireless Standards on FPGA Platform, Research Journal of Applied Sciences, Engineering and Technology, Vol. 6, No. 4, June 2013, pp. 609 – 621.
- [8] K. Rajalakshmi, S. Gondi, A. Kandaswamy: A Fractional Delay FIR Filter based on Lagrange Interpolation of Farrow Structure, International Journal of Electrical and Electronics Engineering, Vol. 1 No. 4, 2012, pp. 103 – 107.
- [9] A. Franck: Efficient Algorithms for Arbitrary Sample Rate Conversion with Application to Wave Field Synthesis, PhD Thesis, Technical University Ilmenau, Ilmenau, Germany, 2012.