International Journal of Multimedia and Ubiquitous Engineering
Vol.11, No.5 (2016), pp. 83-92
http://dx.doi.org/10.14257/ijmue.2016.11.5.09

Model-Based Design Methodology for Sampling Rate Converter

Nie Yang[1, 2], Ge Hua[1] and Jing Li-li[1]

[1] Department of Physics, Jining Normal University, Inner Mongolia, China
[2] Digital Engineering Center, Communication University of China, Beijing, China
E-mail: nieyangwork@163.com

Abstract

The design of high-speed sampling rate converter in multi-rate system is a very complicated problem. The traditional development flow is not simply low efficiency, and but also is difficult to implementation. This paper presents a development flow by model-based design to simplify the design process, to improve work efficiency. Sampling rate converter is designed by a hybrid scheme using polyphase filter and Farrow filter structure. Through the development flow of model-based design, circuit modeling is very easy to do, and the RTL code and test code are automatically generated. The simulation results show that this method can greatly reduce the system consumption and promote the performance of the system.

***Keywords**: Sampling Rate Converter; Polyphase Filter; Farrow Filter; Model-Based Design*

1. Introduction

In the modern communication system, the sampling rate converter (SRC) is used to improve the effectiveness of the whole system. Such systems utilizing multiple sampling rates are known as multirate systems. The basic building blocks of a multirate system have been designed by inserting some factors to construct an appropriate filter. SRC usually adopted two methods of design. For SRC designing，the polyphase filter is a efficient method. However, when inserting the factor is the very large, polyphase filter will Spend a lot of storage space to save huge amounts of the filter coefficients. The other a highly efficient design is Farrow filter structure. Farrow filter can overcome the shortcoming of polyphase filter, which a type is of can realize arbitrary SRC hardware structure. The above two kinds of design method, by using a traditional circuit design process to design two kinds of design method, the design processing is very complex, and is inefficient.

Model-Based Design (MBD) is a mathematical and visual method of addressing problems associated with designing complex control [1-2], signal processing [3] and communication systems. Model-based design provides us with a complete product from idea to generate the code development process. In MBD, a system model is right in the center around the development process, from requirement's development, through design, implementation, and testing. MBD provides a common design environment, which facilitates general communication, data analysis, and system verification between development groups and power electronic application. Engineers can locate and correct errors early in system design, when the time and financial impact of system modification are minimized. MBD can automatically generate code for embedded deployment and create test benches for system verification, saving time and avoiding the introduction of manually coded errors. It is employed in many motion control, industrial equipment, aerospace, and automotive applications. MBD [4-5] is a methodology applied in designing embedded software.

ISSN: 1975-0080 IJMUE

This paper is focused on an efficient implementation of SRC between arbitrary factors. Following this introduction, Section 2 reviews polyphase filter structure and Section 3 discusses Farrow filter structure, which are used to design SRC. Section 4 provides a hybrid scheme. Section 5 discusses circuit implementation by MBD, whereas Section 6 concludes the conclusion.

2. The Design of SRC by Polyphase Filter Structure

A higher-order FIR filter can be realized in a parallel structure based on the polyphase decomposition of the transfer function. The FIR transfer function is decomposed into M lower-order transfer functions, called the polyphase components [6], which are afterwards added together to compose the original overall transfer function. The FIR transfer function $H(z)$ in polyphase form according to

$$H(z) = h[0] + h[1]z^{-1} + h[2]z^{-2} + h[3]z^{-3} + \cdots + h[N-2]z^{-(N-2)} + h[N-1]z^{-(N-1)} \quad (1)$$

The above transfer function can be expressed as a sum of two terms. The first term is the even-indexed coefficients and the second term is the odd-indexed coefficients. Without loss of generality, It can assume that *N* is an odd number, and express equation (1) as follows,

$$\begin{aligned} H(z) = h[0] + &+ h[2]z^{-2} + h[4]z^{-4} + \cdots + h[N-3]z^{-(N-3)} + h[N-1]z^{-(N-1)} + \\ &h[1]z^{-1} + + h[3]z^{-3} + \cdots + h[N-4]z^{-(N-4)} + h[N-2]z^{-(N-2)} \end{aligned} \quad (2)$$

Or equivalently

$$\begin{aligned} H(z) = h[0] + &+ h[2]z^{-2} + h[4]z^{-4} + \cdots + h[N-3]z^{-(N-3)} + h[N-1]z^{-(N-1)} + \\ &z^{-1}(h[1] + + h[3]z^{-2} + \cdots + h[N-4]z^{-(N-5)} + h[N-2]z^{-(N-3)}). \end{aligned} \quad (3)$$

By using the notation

$$E_0(z) = h[0] + h[2]z^{-1} + h[4]z^{-2} + \cdots + h[N-3]z^{-(N-3)/2} + h[N-1]z^{-(N-1)/2} \quad (4)$$

$$E_1(z) = h[1] + h[3]z^{-1} + h[4]z^{-2} + \cdots + h[N-4]z^{-(N-5)/2} + h[N-2]z^{-(N-3)/2}, \quad (5)$$

Transfer function (1) can be expressed as the sum of polyphase components $E_0(z)$ and $E_0(z)$

$$H(z) = E_0(z^2) + z^{-1}E_1(z^2). \quad (6)$$

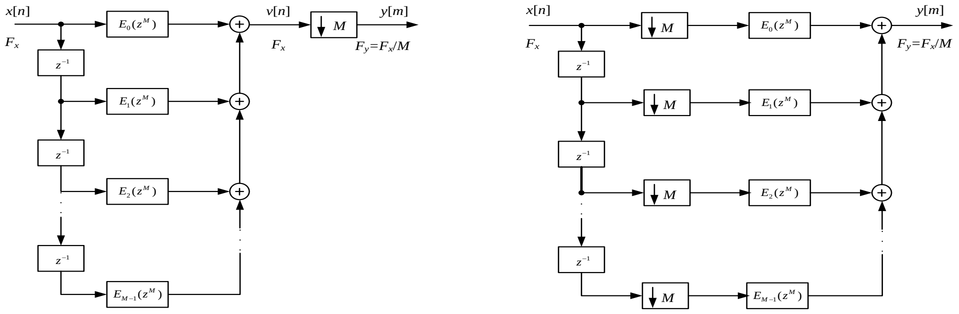
In a general case, an *N*-length transfer function $H(z)$ can be decomposed into *M* polyphase branches $E_0(z), E_1(z), \ldots, E_{M-1}(z)$ in a manner that $H(z)$ is expressible in the form

$$H(z) = \sum_{k=0}^{M-1} z^{-k} E_k(z^M), \quad (7)$$

Where

$$E_k(z) = \sum_{n=0}^{\lfloor N/M \rfloor} h[Mn+k]z^{-n}, \qquad 0 \le k \le M-1. \quad (8)$$

An FIR filter can be implemented as a parallel connection of *M*(*L*) polyphase components, which are added together at the output. The polyphase components are sometimes called polyphase sub-filters or polyphase branches [7-8]. A polyphase component is usually implemented in the direct transversal form. Figure1 shows a decimator composed of the cascade of an FIR filter implemented as a parallel connection of *M* polyphase branches, and factor-of-*M* down-sampler. Here the arithmetic operations in the polyphase branches are to be performed at the input sampling rate, *i.e.* at the higher sampling rate of the system. Instead of down-sampling at the filter output, one can shift the down-sampling operation into the polyphase branches before the output adders.

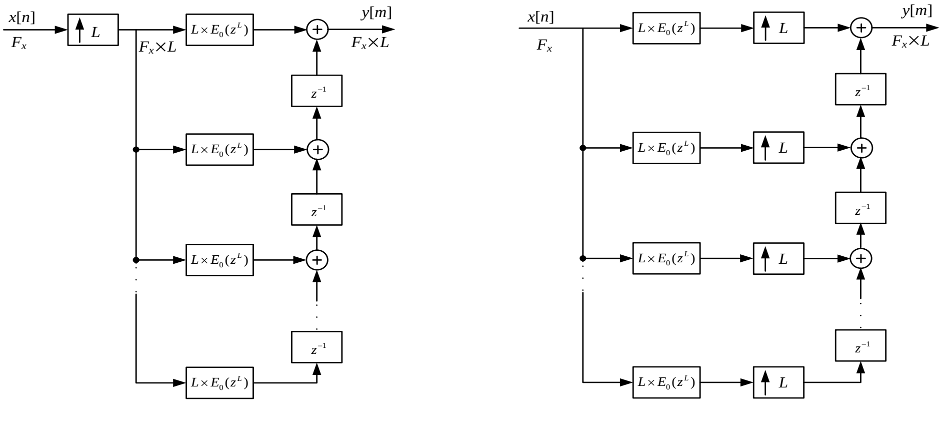


(a) Filter and Down-Sampler (b) Efficient Polyphase Decimator

Figure 1. Polyphase Implementation of FIR Decimator

In the structure of Figure1 (b), the down-sampling-by-*M* occurs at the inputs of the polyphase components $E_0(z), E_1(z),\ldots, E_{M-1}(z)$ and filtering is performed at the sampling rate F_x/M . The overall computational complexity of the decimator is reduced by *M*.

In the interpolator structure of Figure2 (a), the up-sampling precedes filtering, and according to this, filtering in the polyphase components is performed at the higher sampling rate. The structure of Figure2 (a) can be modified to the more efficient structure shown in Figure2 (b). The positions of up-samplers and polyphase components are interchanged, and filtering in the polyphase branches is to be performed at the lower sampling rate.



(a) Filter and Up-Sampler (b) Efficient Polyphase Interpolator

Figure 2. Polyphase Implementation of FIR Interpolator

Polyphase structures are generally considered efficient implementations of multirate filters. However, in the case of fractional sample rate conversion. To resample a signal from 8 kHz to 44.1 kHz, it can be done in a relatively efficient manner in two stages. Overall interpolation factor is 441, and overall decimation factor is 80. The interpolation factor is 147, and the decimation factor is 80 in the first stage of filter. The other interpolation factor is 3 in the second stage of filter. Using MATLAB programming, the resource consumption results of SRC based on polyphase filter structure are as follows:

the number of filter coefficients is 1774; the number of operations per input sample is roughly 95 multiplications and 89 additions. In this kind of design scheme, 1774 coefficients would have to be stored in memory in this case.

3. The Design of SRC Based on Farrow Filter Structure

Polynomial-based filters are another way to overcome the problem of needing a large number of coefficients to be stored. Farrow structures are efficient implementations for such filters [9-11]. It is very useful in providing fractional delay of digital signals. The output of filters is given by

$$y[(m+D+\mu)T]=\sum_{i=0}^{N-1}x[(m-i)T]\square h_{\mu}(i) \tag{9}$$

Where $x[(m)T]$ is the input signal sampled at a period *T*; $h_{\mu}(i)$ is the impulse response of the filters with a system delay of $D+\mu$, where *D* is an integer constant, and μ is the fractional. Delay parameter between zero and one. *N* is the length of $h_{\mu}(i)$.To avoid the implementation of a large number of filters with different delays, Farrow proposed to approximate each impulse response $h_{\mu}(i)$ with the following *P*th-order polynomial in delay value μ such that the delay control is independent of the filter coefficients

$$h_{\mu}(i)=\sum_{n=0}^{P}b_{n}(i)\mu^{n}. \tag{10}$$

Substituting (10) into (9) gives

$$y[(m+D+\mu)T]=\sum_{n=0}^{P}[\sum_{i=0}^{N-1}x[(m-i)T]\square b_{n}(i)]\mu^{n}. \tag{11}$$

Figure3 shows the Farrow structure for implementing (11), where the input signal is passed through a number of subfilters $b_{n}(i)$, $n=0,\ldots,P$, and is multiplied by the appropriate powers of μ to produce the output.

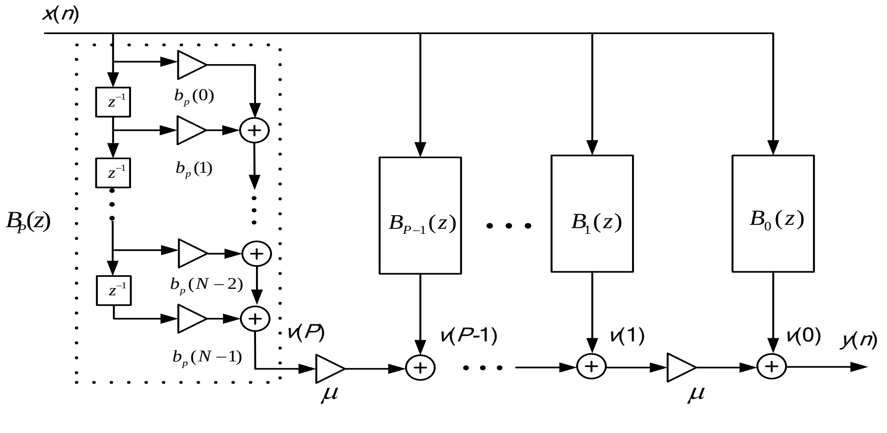


Figure 3. Farrow Filter Structure

Though the Farrow structure is very useful in providing a continuous value of signal delay for digital signals by changing the value of μ, it still requires large number of multiplications for the sub-filter implementation, especially when *P* and *N* are large to provide very precise control of the frequency characteristics of the filter. Fortunately, thanks to the transposed form of the sub-filters, the Farrow structure can be rewritten as in Figure4. In this new structure, the input is multiplied to a number of constant coefficients, which further reduces the complexity of the Farrow structure.

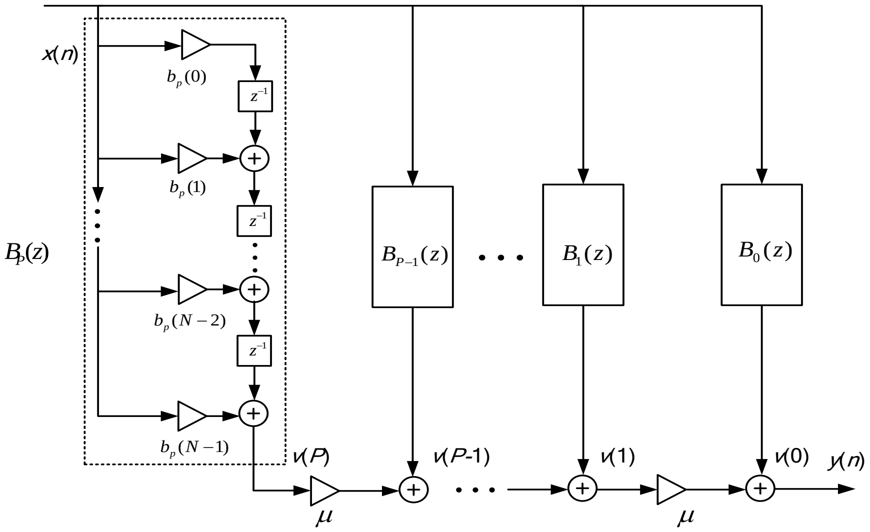


Figure 4. The Transposed form Structure of Farrow Filter

The Farrow structure is very useful in providing a continuous value of signal delay for digital signals by changing the value of, it still requires large number of multiplications for the sub-filter implementation, especially when P and N are large. Using MATLAB programming, the resource consumption results of SRC based on Farrow filter structure are as follows: With 3rd-order polynomials, 16 coefficients are needed and about 66 multiplications per input sample; Fourth-order polynomials provide slightly better lowpass response at a higher cost: 25 coefficients and 121 multiplications per input sample. The frequency response of SRC by three different kinds of design scheme is shown in the Figure 5.

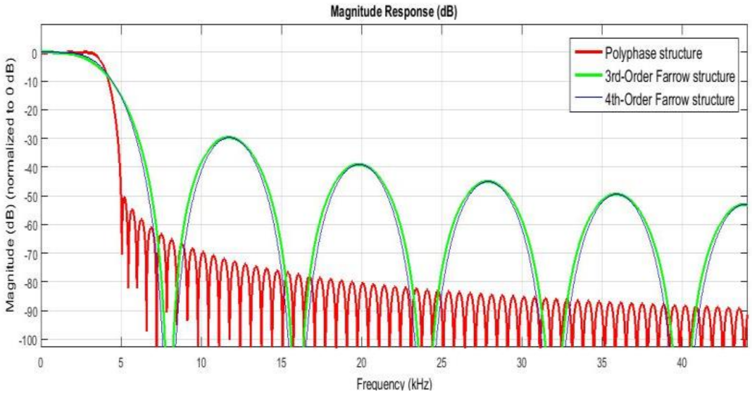


Figure 5. The Frequency Responses of SRC by Farrow Filter and Polyphase Filter

4. The Design of SRC Based on a Hybrid Structure

Polyphase filters are particularly well adapted for interpolation or decimation by an integer factor and for fractional rate conversions when the interpolation and decimation factors are low [12]. Farrow filters can efficiently implement arbitrary (including irrational) rate change factors [13-14]. Now a hybrid solution would make use of the two types of filters that we have previously seen. First, polyphase filters will be interpolated for the original 8 kHz signal. Then, Farrow filter will be interpolated for the intermediate 32 kHz signal to get the desired 44.1 kHz final sampling frequency. The system block diagram is shown in Figure6.

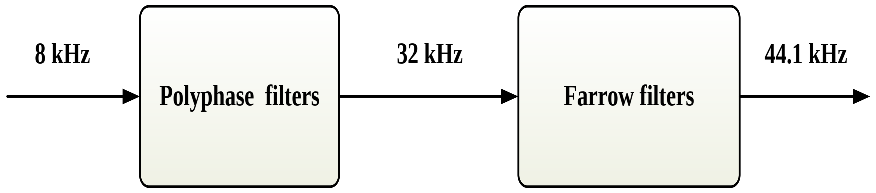


Figure 6. The design of SRC Based on a Hybrid Structure

Using MATLAB programming, the resource consumption results of SRC based on a hybrid structure are as follows: The number of coefficients of this hybrid design is relatively low: 36 and the number of multiplications per input sample is also relatively low: 92. Results resource consumption of the three scheme shown in Table 1. The filter stages the inputs to each stage will be is quantized by 12 bit, which will forbid the data path to grow to very large word lengths. The magnitude response of SRC by the cascaded and quantized filter is shown in Figure7.

Table 1. Resource Consumption of SRC by the Different Design Scheme

Design scheme	Number of coefficients	Multiplications per input sample	Additions per input sample
Polyphase structure	174	95	89
Farrow structure	25	121	99
Hybrid structure	36	92	40

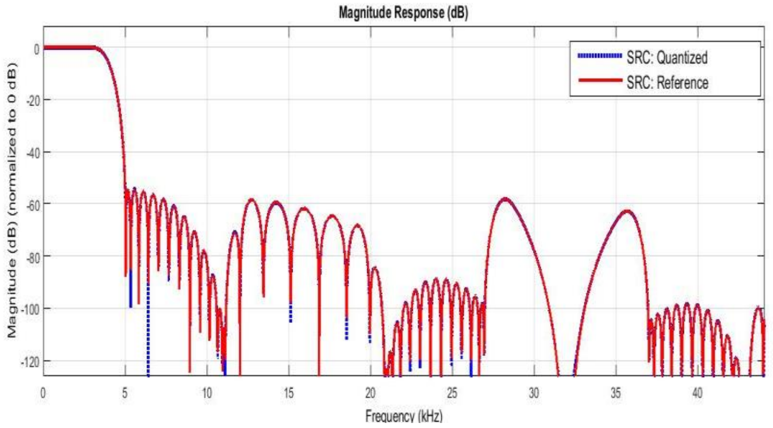


Figure 7. The Frequency Response of SRC by a Hybrid Structure

5. The Implementation of SRC by Model-Based Design

With the continuous development of embedded systems, people are increasingly demanding for real-time of electronic products, which greatly increased the complexity and difficulty of embedded systems product development. If adopted the traditional development flow, it is not conducive to the development of efficient, high-quality products in the fierce competition of market. In addition, different departments shall be responsible for different processing and communication between departments easily lead ambiguity through paper document. Errors will be transferred and influence the progress of the development flow. if there is a problem, the problem need to be analyzed and validated for a long time due to testing in the completion of the entire design. If the problem occurs at the beginning of the design, the whole design cycle and cost bring great negative effects.

In the Model-Based Design (MBD) [15-16], a system model is at the center of the development process, from requirements development, through design, implementation, and testing. The development flow by MBD is shown in Figure 8. Through the establishment of floating point model, the fixed-point model and the system-level model for presenting a complete system design requirements, different professional engineers work efficiently, and can communicate at different stages of the development flow. All aspects of the design can be tested and verified according to the model of the system level and the demand. HDL tool provides a workflow advisor that automates the programming of Xilinx® and Altera® FPGAs. It can control HDL architecture and implementation, highlight critical paths, and generate hardware resource utilization estimates. HDL tool generates VHDL and Verilog test benches for rapid verification of generated HDL code, automatically generates two types of cosimulation models. HDL cosimulation model is applied for performing HDL cosimulation with Simulink and an HDL simulator, such as Cadence Incisive or Mentor Graphics ModelSim. FPGA-in-the-loop (FIL) cosimulation model is applied for verifying design with Simulink and an FPGA board.

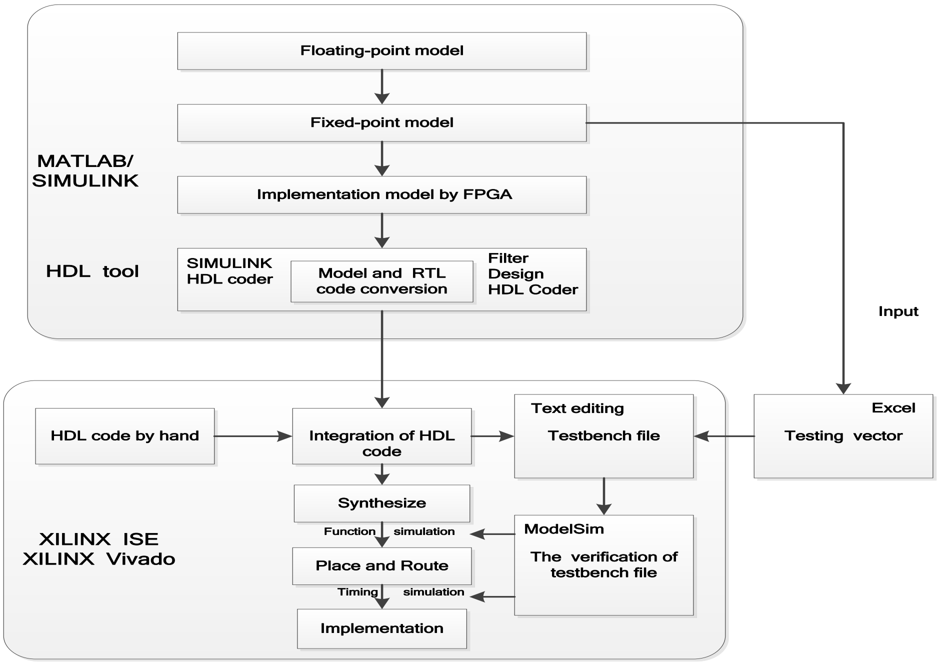


Figure 8. The Development Flow by MBD

In the development flow based on MBD, the model system of SRC is set up to convert the rate of a signal from 8 kHz to 44.1 kHz. HDL Workflow Advisor is a MATLAB tools for supporting the FPGA design, which verifies the model, automatically generates HDL code and configures FPGA. Using XILINX FPGA development ISE as synthesis tool, the HDL code automatically generated of SRC is synthesized and implement by XILINX FPGA chip. The simulation results are shown in Figure 9. It can be seen from the simulation results that it is not only accurate, but also fast and effective by the development flow based on MBD.

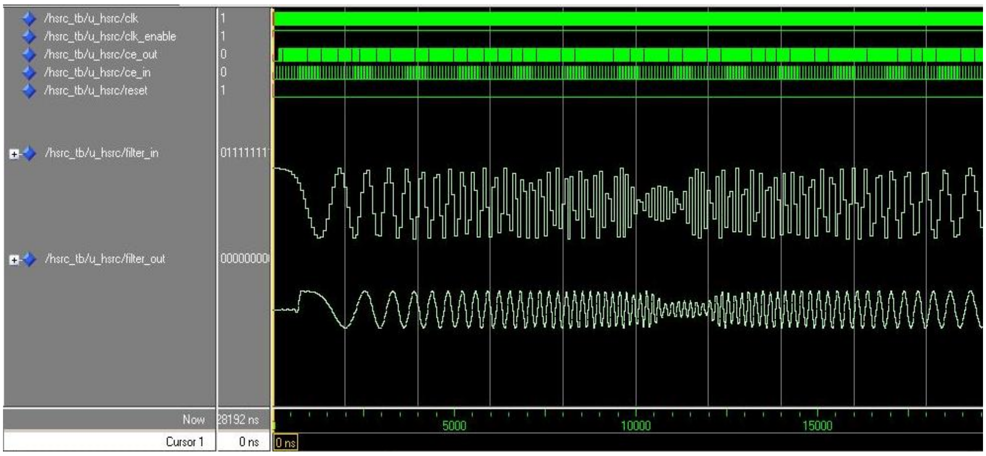


Figure 9. The Simulation Results of SRC by the Development Flow Based on MBD

6. Conclusion

In this paper, we use Farrow and polyphase filters respectively completed the design of SRC when it is desirable to convert the sampling rate of a signal from 8 kHz to 44.1 kHz. Then, a new method for the design and efficient implementation of SRC with a hybrid scheme is presented, and use Model-Based Design method to complete the circuit implementation of SRC. The simulation result is shown that the development flow based on MBD is not only simple and effective, and is useful to reduce the cycle of product. Therefore, this paper has some reference value for engineering.

Acknowledgement

These should be brief and placed at the end of the text before the references. We would like to thank my mentor Professor GE for stimulating discussions with respect to the topic of this paper and laboratory equipment. Moreover, we greatly appreciate the reviewers' comments that lead to an improved presentation of the results. This work was supported by Research Program of science and technology at Universities of Inner Mongolia Autonomous Region (NJZZ14288).

References

[1] S. Wang and K. G. Shin, "Task Construction for Model-Based Design of Embedded Control Software", IEEE Trans. Software Eng, vol. 32, no. 4, **(2006)**, pp. 254-265.

[2] M. F. Costabile, D. Fogli, P. Mussion and A. Piccinno, "Visual interactive systems for end-user development: A model-based design methodology", IEEE Trans. Syst., Man, Cybern. A, Syst, Humans, vol. 37, no. 6, **(2007)**, pp. 1029-1046.

[3] S. S. Solano, M. B. Jimenez, E. D. Toro, P. B. Jimenez and I. Baturone, "Model-based design methodology for rapid development of fuzzy controllers on FPGAs", IEEE Trans. Ind. Information, vol. 9, no. 3, **(2013)**, pp. 1361-1370.

[4] S. Chatterjee and W. B. Kleijn, "Auditory model-based design and optimization of feature vectors for automatic speech recognition", IEEE Trans. Audio Speech Lang. Process, vol. 19, no. 6, **(2011)**, pp. 1813-1825.

[5] T. Miyajima, H. Fujimoto and M. Fujitsuna, "A precise model-based design of voltage phase controller for IPMSM", IEEE Trans. Power Electron, vol. 28, no. 12, **(2013)**, pp. 5655-5664.

[6] J.-H. Lee and W.-J. Kang, "Designing filters for polyphase filter banks", Proceeding Inst. Elect. Eng. G, vol. 139, **(1992)**, pp. 363-369.

[7] F. Harris, C. Dick and M. Rice, "Digital receivers and transmitters using polyphase filter banks for wireless communications", IEEE Trans. Microw. Theory Tech, vol. 51, no. 4, **(2003)**, pp. 1395-1412.

[8] J. Kaukovuori, K. Stadius, J. Ryynanen and K. Halonen, "Analysis and design of passive polyphase filters", IEEE Trans. Circuits System I, Reg. Papers, vol. 55, no. 10, **(2008)**, pp. 3023-3037.

[9] C. W. Farrow, “A continuously variable digital delay element”, Proceedings of 1988 IEEE International Symposium on Circuits and Systems. Espoo: IEEE, vol. 3, **(1988)**, pp. 2641-2645.
[10] C. K. S. Pun, Y. C. Wu, S. C. Chan and K. L. Ho, “On the design and efficient implementation of the Farrow structure”, IEEE Signal Process. Lett, vol. 10, no. 7, **(2003)**, pp. 189-192.
[11] M. Abbas, O. Gustafsson and H. Johansson, “On the fixed-point implementation of fractional-delay filters based on the Farrow structure”, IEEE Trans. Circuits, Syst. I: Reg. Papers, vol. 60, no. 4, **(2013)**, pp. 926-937.
[12] D. W. Barker, “Efficient resampling implementations”, IEEE Signal Process. Mag, vol. 25, no. 4, **(2008)**, pp. 114-117.
[13] R. Bregovic, Y. J. Yu and Y. C. Lim, “Implementation of linear-phase FIR filters for a rational sampling-rate conversion utilizing the coefficient symmetry”, IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 58, no. 3, **(2011)**, pp. 548-561.
[14] H. Johansson and H. Gockler, “Two-Stage-Based Polyphase Structures for Arbitrary-Integer Sampling Rate Conversion”, Circuits and Systems II: Express Briefs, IEEE Transactions on, vol. 62, no. 5, **(2015)**, pp. 486-490.
[15] C. Madritsch, “Rapid Prototyping using Model-Based Design methodology: A Digital Signal Processing lecture case study”, Proceedings of the 33rd International Convention, IEEE, **(2010)**, pp. 849-851.
[16] S. Perry, “Model based design needs high level synthesis: a collection of high level synthesis techniques to improve productivity and quality of results for model based electronic design”, Proceedings of on Design, Automation and Test in Europe, **(2009)**, pp. 1202-1207.

Authors

Nie Yang, is a Ph.D. in the Digital Engineering Center of Communication University of China and is a lecturer of Jining Normal University, China. His main research interests include Compressive Sensing, High-performance DSP Algorithms and VLSI Architectures.