Implementation of FIR filter based on Xilinx IP core

Jiangchun Yu, Hanhua Zhong, Pengyang Yan
School of Electrical and Automation Engineering, East China Jiaotong University, Nanchang 330013, China

Abstract. **In order to accurately calculate the effective value of voltage and current, it is necessary to filter out the harmonics in voltage and current signals in the process of developing power RTU with Zynq-7000 series FPGA. This paper discusses the use of the Xilinx FIR IP core to achieve this filtering function. Firstly, we use FDAtool to design a 50 Hz band-pass filter, analyze the frequency characteristics of the filter, and then quantize the FIR filter coefficients generated by Matlab tool by calculating the quantization word length, and then import the quantized filter coefficients into the FIR IP core. Finally, the realization and optimization of the IP core are completed by setting the relevant parameters. This method simplifies the design process and makes the realization of the filter easier. The simulation results of MATLAB and FPGA show that the filtering effect is consistent, and the FIR filter consumes fewer resources than the Fourier Algorithm.**

Keywords: FIR filter; Xilinx IP core; Implementation

I. INTRODUCTION

In order to monitor and control the operation of 10kV power supply system of high-speed railway, RTU (remote terminal unit) is set along the railway to collect and calculate real-time data such as current, voltage and power. With the rapid development of microelectronics technology, the Zynq series FPGA products produced by Xilinx company are gradually becoming the mainstream of high-performance system development and design. It is a high-performance chip with an FPGA+arm structure. In this paper, we consider to develop a new generation of high-performance and high integration RTU with the Zynq-7000 chip.

The power supply line transmits 50 Hz AC power frequency. Due to the existence of nonlinear load and electromagnetic interference, the measured voltage and current waveforms will be distorted, making the measured signal may contain a large number of harmonics multiple than 50 Hz, such as 150Hz, 250Hz, 350Hz, etc. [1][1]. According to the national electricity detection standard, the effective value of voltage and current is for the fundamental wave of 50 Hz, and the existence of harmonics will produce great errors. There are many methods to filter out harmonics, and the commonly used methods are fast Fourier transform [2] (FFT) and digital finite filter (FIR). Compared with the traditional algorithm based on CPU, FFT and FIR implemented on FPGA can achieve fast time response performance. Considering that FFT takes up more hardware resources, this paper mainly discusses the method of FIR filtering.

Xilinx company provides a Vivado platform for the development of Zynq series products. The system has a large number of Xilinx IP cores with different uses. The FIRI and IP cores are used to filter out the fundamental wave signal. The development process is simple and convenient. Next, the design process and operation results of the filter are discussed and verified by MATLAB simulation.

II. FILTER DESIGN AND FILTER COEFFICIENT

A. FIR filter design

First, we discuss the principle of FIR filter. The transfer function and difference equation of FIR system with length (tap number) N are shown in equations (1.1) - (1.2) [3].

$$H(z)=\sum_{n=0}^{N-1}h(n)z^{-n} \qquad (1.1)$$

$$y(n)=\sum_{k=0}^{N-1}h(k)x(n-k) \qquad (1.2)$$

From the formula, we can see that the essence of FIR filter is to multiply the input signal sequence and filter coefficients, and then accumulate the results. The performance of the filter is mainly reflected in the establishment of filter coefficients. The number of filter coefficients is also known as the order of the filter. The more the order is, the more calculation times will be, and the more FPGA resources will be consumed, which will be more accurate relative to the filtering results. Therefore, the value of the order of the filter has a great impact on the whole design.

The design methods of FIR filters are mainly based on approximate approximation of the frequency characteristics of ideal filters. These methods include the window function method, frequency sampling method, etc[4]. These methods have been integrated into the filter analysis and design tool FDAtool of Matlab. Users can directly use the tool to design filters according to the indicators. This paper uses the window function method to complete the filter design. The design of the window function method includes the selection of the window and the determination of the stopband cut-off frequency, the starting frequency of the passband, the cut-off frequency of the passband and the sampling frequency.

Considering that the fundamental frequency that we want to filter out is 50 Hz, the filtered harmonic frequency is an integral multiple of the fundamental frequency, which is in different frequency bands. Therefore, the stopband cut off frequency is 10Hz, the passband starting frequency is 40Hz, the passband cutoff frequency is 60Hz, the stopband starting frequency is 100Hz, the stopband attenuation is - 20db, the window function is selected as Kaiser window, and the frequency deviation caused by quantization word length is set to less than 0.001. The RTU sets the sampling frequency to 1600Hz, so the sampling frequency of the filter Fs=1600Hz. The order of the designed filter is 63.

978-0-7381-1141-4/20/$31.00 ©2020 IEEE
DOI 10.1109/ITCA52113.2020.00024

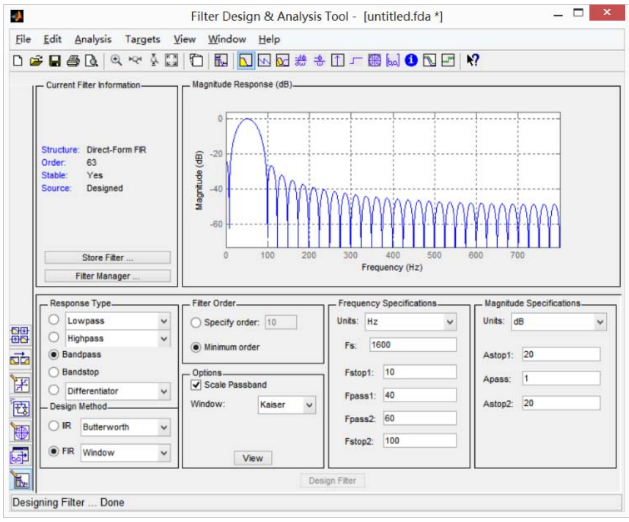

Fig. 1 Design of FIR filter index

Figure 2 is the frequency response diagram. The correctness of the index is analyzed below. It can be seen that the attenuation is -24.66283db when the frequency is 0Hz, -0.000660766db when the frequency is 50 Hz, and-29.67292db when the frequency is 100Hz.

According to the formula: attenuation = - $20\log_{10}K$, K is the gain coefficient (the ratio of input divided by output). When the frequency is 50 Hz, the attenuation is close to 0 db. K is calculated as 1, which means that the input signal is not attenuated at the frequency of 50 Hz. After calculation, the attenuation is - 56.73db and K is 0.00158 at 100Hz. It can be seen that the amplitude attenuation is 0.00158 times of the original at 100Hz frequency. When 0 Hz is the DC frequency point, the attenuation is - 23db, K is 0.07079, and the amplitude of DC signal is 0.0707 times of the original.

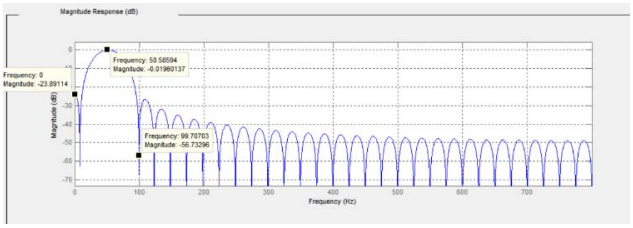

Fig. 2 Frequency response of fir filter

B. *Acquisition of FIR filter coefficients*

When FPGA uses filter coefficients, it needs to convert decimals into fixed-point numbers, which requires the use of coefficient quantization function in FDAtool to achieve this purpose [5]. This part needs to consider the influence of coefficient quantization on the frequency characteristics of the filter. The quantization word length is a part of the coefficient quantization, and its value directly affects the frequency characteristics of the filter, thus causing interference to the output of the filter. Therefore, the more accurate the quantization word length is, the smaller the final output error of the filter will be. According to the design index, the frequency characteristic deviation caused by quantization error should not exceed 0.001[6], so according to the quantization word length formula:

$$(N+1)2^{-b}/2 \le 0.001$$

When N = 63 is brought into the calculation, b > = 15 is obtained. It can be seen that the quantization word length of the filter is at least 15.

Note: N is the order of the filter, b is the quantization word length

The following is a comparison of different frequency characteristic graphs generated by different quantization word length through FDAtool. The quantization effect of filter coefficients at the filter zero position is shown in Fig.3-6. In the case of unquantization, the zeros on the unit circle are dense, but the zeros are still spread in the z plane [7]. The zero-point on the unit circle mainly forms the stopband attenuation, but the zero-point on the unit circle which is not conjugate reciprocal position mainly forms the passband characteristic.

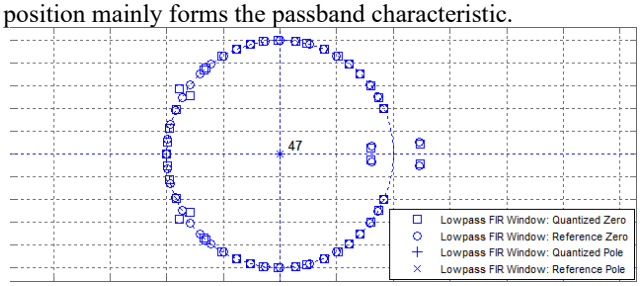

Fig. 3 Eight-bit coefficient of FIR low pass filter

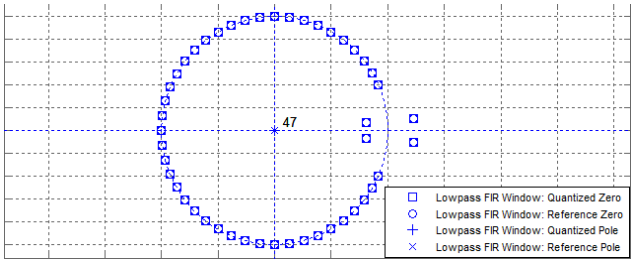

Fig. 4 Twelve-bit coefficient of FIR low pass filter

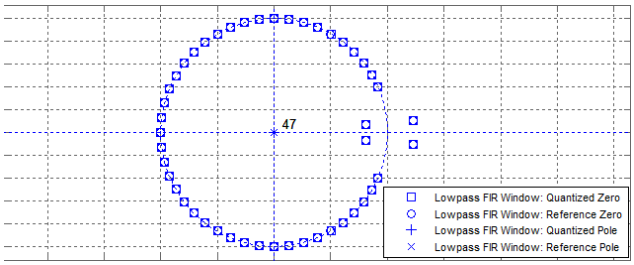

Fig. 5 Sixteen-bit coefficient of FIR low pass filter

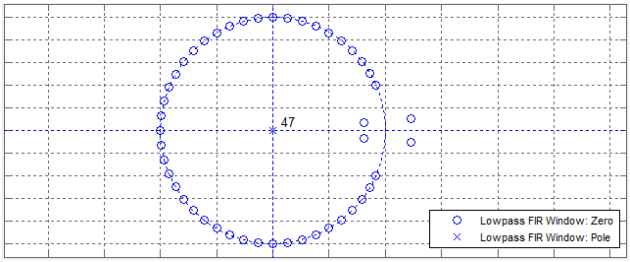

Fig. 6 No quantified coefficient of FIR low pass filter

As shown in Fig.5, for the 16-bit wide quantization effect diagram, there is hardly any movement of the zero-point position; in Fig.4, for the 12-bit wide quantization effect diagram, there is a small amount of movement of the zero-point position; as shown in Fig.3, for the 8-bit wide quantization effect diagram, several zeros move out of the unit circle in pairs to the position of the conjugate reciprocal. Finally, the conclusion is drawn that when the quantization word length is too short, the frequency characteristic error is too large, and the zero-pole drift is abnormal. It can be seen that taking the quantization word length more than 15 bits has little influence on the various characteristics of the filter. In order to facilitate the final FIR IP core to call the coefficient file, 16-bit quantization word length is selected.

To view the filter coefficients generated by FDAtool:

Table 1 Filter coefficient

Coefficient	Quantized Numerator	Quantized Coefficient (16 bits)	FDAtool Generation Coefficient
h(0)	-0.011222695015972076	-735.49021184	fd21
h(1)	-0.019827123717502882	-1299.3903329	faed
h(2)	-0.025953692258885323	-1700.90102784	f95b
h(3)	-0.027054752620958519	-1773.06026776	f913
h(4)	-0.021005328115214612	-1376.65018334	fa9f
h(5)	-0.006643405769890260	-435.3822359552	fe4d
h(6)	0.015828538671275632	1037.3391056896	040d
h(7)	0.04468753755564861	2928.6424612438	0b71
h(8)	0.076837949461913427	5035.6518559359	13ac
...		...	...

The filter coefficients generated by the FDAtool are a series of decimal numbers in the second column of the above table. After quantization, h(n)*216 is obtained. Finally, the quantized coefficient (16-bits) is converted into hexadecimal fixed-point data and used by FIR IP_CORE.

FIR filter IP configuration

III. FIR FILTER IP CONFIGURATION

Vivado development software provides Xilinx FIR IP for free. This IP core is verified and tested by a large number of digital signal processing professionals to ensure its correctness. With the development of generations of developers, its general effect and various functions are very perfect. Compared with other design methods, using the IP core to design FIR filter is more convenient[8]. FIR IP core configuration is shown in the figure.

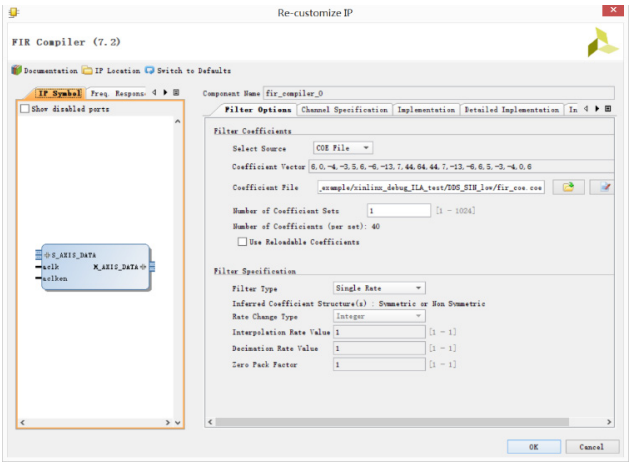

Fig. 7 Fir IP core configuration

The filter coefficients can be obtained by calculation or by calling the .coe file. The IP core provides the coefficient overload function. The filter types can be single rate filter, interpolation filter, decimation filter, etc.

When the system interpolates the input signal by N times, N images will appear in the spectrum, so a filter is needed to filter the image frequency. Interpolation filter is usually

implemented by FIR and CIC filter. In the signal processing system, an anti-aliasing filter is usually used after AD sampling to filter out the frequency of aliasing after decimation, so a decimation filter is needed.

In this paper, we choose to use a single rate filter structure. The single rate FIR filter is the simplest finite impulse response filter. The so-called single rate FIR filter means that the input sampling frequency is equal to the output sampling frequency. FIR compiler IP core to achieve this type of filter is mainly to calculate the convolution sum, as shown in the following formula [9]:

$$y(k) = \sum_{n=0}^{N-1} a(n)x(k-n)$$

Where N is the number of filter coefficients.

The clock frequency of FIR filter system is selected as 100M, and the input sampling frequency can be selected as the frequency mode, then the input data is input into the IP core according to this sampling frequency; if the selected frequency mode is the output sampling frequency, the output data is output according to this frequency.

The filter coefficient type selects a signed type with a coefficient width of 16bit, and the FIR compiler IP core offers four quantization options: integer coefficient, quantification only, maximizing dynamic range, and optimal precision scale. Using the coe coefficient file for import into the IP core, the coefficient quantification option selects the integer factor by default.

FIR compiler IP core provides four quantization options: integer coefficient, quantization only, maximum dynamic range and optimal precision decimal length. The coe coefficient file is imported into the IP core for use. The integer coefficient is selected as the coefficient quantization option by default.

The output of the FIR compiler IP core can be set to full precision, and the output data bit width is the sum of the input data bit width and the added value of the bit width when filtering according to the filter coefficient. The increase of bit width is caused by the multiplication and accumulation operation to realize the basic filtering function. The upper limit of the increase in bit width is the result of rounding up the base-2 logarithm of the nonzero multiplicative operands plus the system bit width. However, in practice, the coefficient is small, and the increase of bit width will not reach the maximum value. In general, the following formula can be used to limit the bit width of multiplication accumulator, i.e. the increased value of bit width [10]:

$$B = [\log_2(\sum_{i=0}^{N-1} |a_i|)]$$

Where N is the tap number of filter coefficients, ai is the coefficient value, and B is the bit width increase value. For the FIR compiler implemented with MAC structure, the bit width increment is calculated automatically according to the actual coefficients.

FIR compiler IP core provides optimization options: speed option, custom option, area option. In some configurations, the maximum possible speed needs to be at the expense of overall resource utilization. The speed option selects all the possible optimization objectives supported by the kernel, and consumes the most resources to achieve the maximum speed; the custom option selects the optimization objectives by the user and displays them in the form of a list, which can provide more accurate control on the critical path to achieve the specific goal; the area option turns off the optimal function.

User-defined functions can be selected from the list. The functions are as follows:

Data_ Path_ Fanout: adding registers to the data storage side can minimize fan out.

Pre-Adder_ Pipeline: when large width coefficients are used, pipeline pre adders are implemented.

Coefficient_ Fanout: adding pipelined registers to the coefficient storage side minimizes fan outs. It is effective when using parallel channel or large data width filter.

Control_ Path_ Fanout: add pipelined registers to the control logic unit when using parallel channels.

Control_ Column_ Fanout: when the dsp processor is used to implement the filter, the pipeline register is added to the control logic unit.

Control_ Broadcast_ Fanout: add pipeline register in control logic unit to implement parallel symmetric filter.

Control_ LUT_ Pipeline: realize the look-up table method of advanced control logic channel sequence.

No_ BRAM_ Read_ First_ Mode: does not use the specified block RAM read mode.

Optimal_ Column_ Length: when the implementation of asymmetric filter requires multiple dsp processors, the processing speed of partitioned dsp reaches the maximum.

Disable_ Half_ Band_ Center_ Tap: when this option is selected, a dsp chip is used to implement the central tap.

The following table shows the resource usage of the most preferred items after IP core synthesis of FIR compiler:

Table2 Resource usage results

	LUTs	Registers	BRAMs	DSPs
Area	251	761	0	11
Speed	278	765	0	11
Custom	262	763	0	11

Notes: LUT, Register, BRAM, DSP

The table above shows the resource usage in the report of Vivado software and obtains the number of logical resources

occupied by the three optimization modes. We can know that the area mode takes the least resources, the speed mode takes the most, and the custom mode can take into account the requirements of resources and speed training. Custom mode list selection: Data_Path_Fanout, Pre-Adder_Pipeline, Coefficient_Fanout. Adding registers on the data storage side can effectively reduce the fan-out to minimize the delay of data transmission, and FIR timing can be well controlled.

Using the custom mode, you can define the list options that need to be optimized, and select the appropriate hardware resources according to the parameter settings. Therefore, when speed is the most important factor, speed mode is the best choice; when the requirement of data processing speed is not high, area mode can be selected. In general, according to the custom mode, it is more flexible, which can take into account the requirements of resource occupation and speed.

IV. VERIFICATION AND ANALYSIS

In this paper, block design is used in Vivado software for module design, and the simulation results are compared with MATLAB software. The verification diagram is shown in Fig.8:

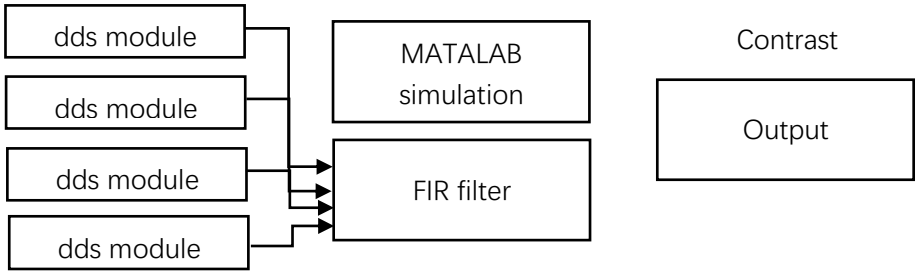

Fig. 8 Functional verification block diagram

The functional verification block diagram of FIR filter is shown in the figure. The sine waves of 50 Hz, 100 Hz, 150 Hz and 200 Hz are generated through the dds signal generator module to simulate the fundamental wave and the second, third and fourth harmonic signals in the line. The fundamental signal is filtered out by a digital filter. The block design of Vivado software is used for modular design, and the design diagram is as follows:

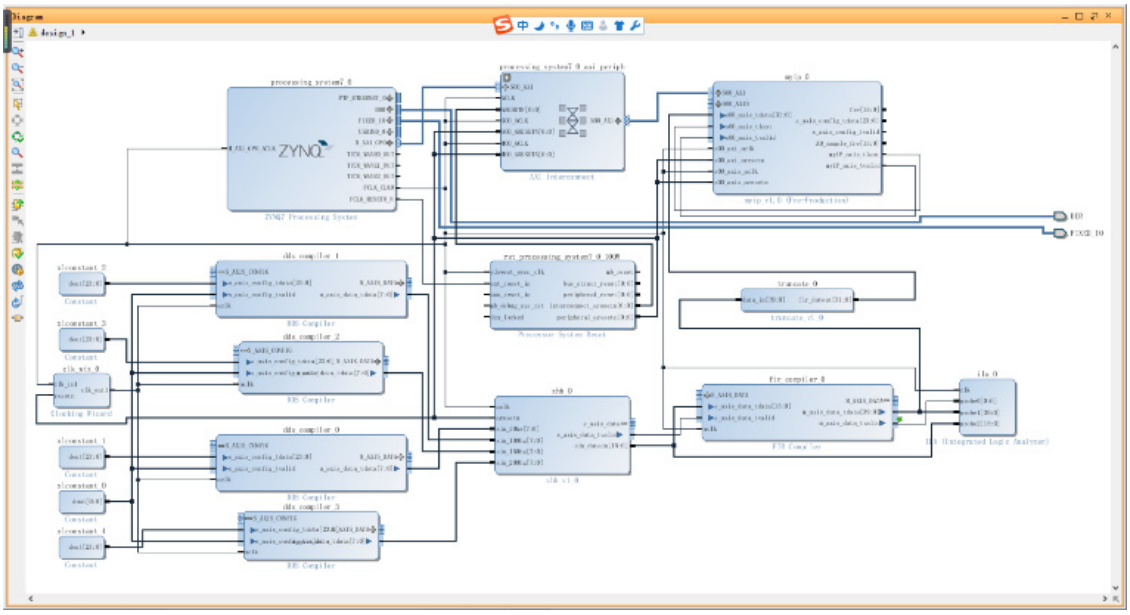

Fig. 9 Modular design diagram

Vivado provides the ability to tag any network at different stages of design. The hardware logic analyzer allows the designer to access the internal signal to realize hardware debugging without the need to lead out the internal logic through FPGA pins.

As shown in FIG.10, ILA (logic analyzer) can grab internal signal after writing in-situ process sequence on FPGA, set trigger signal as FIR IP core, output effective signal, connect FIR output and input signal on ILA core, and then realize hardware debugging after burning program, signal capture situation of ILA can be seen, as shown in the following figure:

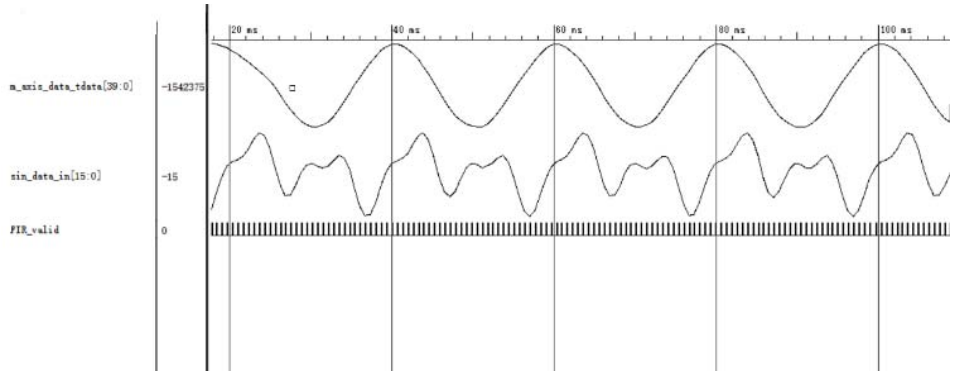

Fig.10 Logic analyzer signal capture diagram

System generate (SysGen) is a FPGA modeling software embedded in MATLAB Simulink. The FIR filter is modeled by SysGen to verify the reliability of the design. Using the sine signal generator as excitation, the FIR filter model is constructed, as shown in FIG. 11

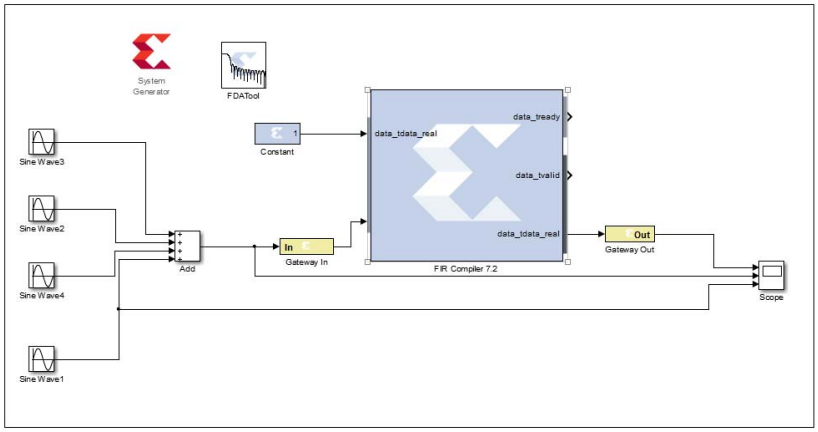

Fig. 11 SysGen FIR filter design diagram

Note: the clock frequency of the system and the sampling frequency of gateway in components must be consistent.

The system built by SysGen is simulated. The simulation results are shown in FIG. 12:

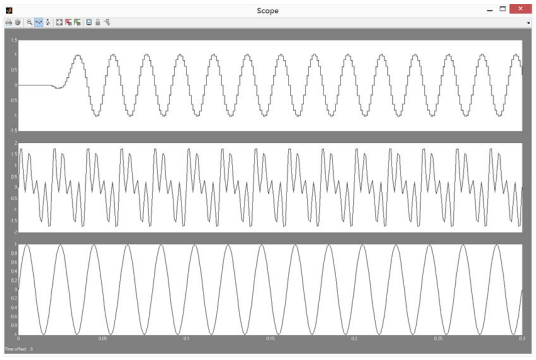

Fig. 12 SysGen simulation results

In Vivado software, the implementation of FIR filter IP core can filter out the high-order harmonic components and leave the waveform of fundamental frequency. Compared with MATLAB simulation software, the reliability of the results is verified.

V. CONCLUSION

In this paper, based on the Xilinx IP core method, the FIR filter is designed and implemented on the FPGA of the Zynq-7000 platform. The function of filtering fundamental wave and eliminating harmonic in RTU electric quantity measurement is completed. The correctness of the filtering function is verified by MATLAB simulation. The implementation method can greatly shorten the research and development cycle, reduce the cost, and select the appropriate hardware resources and data processing rate according to the parameter settings.

REFERENCES

[1] Jing Long, Tang Fen, Wang Zhihe, et al. Power quality control method of traction power supply system based on modular multilevel converter. Automation of Electric Power Systems, 2015(11):173-179.

[2] Xue pin, Li Zhihua, Shen Maoding, et al. Application of finite impulse response filter (FIR) and fast Fourier transform (FFT) in induced polarization method [J]. Science, technology and engineering, 2015 (29): 39-45.

[3] Cheng Peiqing. Digital signal processing course [M]. 3rd Edition. Beijing: Tsinghua University Press, 2007.

[4] LI Chun, HAN Jun-hai. Technology of FIR filtering based on FPGA and application in laser gyroscope. Journal of Chinese Inertial Technology, 2013, 21(6):818-821.

[5] Wang Yaqin, Liu Xuebin, Hu Bingliang. Implementation of a LMS filter on FPGA employing extreme DSP and smart IP-core design[C]//10th International Conference on Electronic Measurement&Instruments. 2011: 341-345.

[6] CAO Zhenji, HL, Min. FIR digital filter based on FPGA and Matlab. Modern Electronics Technique, 2015, 38(19):98-101.

[7] J. Robert Oppenheimer. Discrete - time signal processing. Xi'an Jiaotong University Press, 2001.

[8] Song Wenmiao, Liu Bo. Design of the CIC decimation filter based on SOPC Builder[C]//3rd International Conference on Computer Science and Information Technology.2010:602-605.

[9] Liu Donghua. The description of Xilinx series FPGA chip IP core. Electronic Industry Press. 2013

[10] Gao Yajun. Digital Signal Processing Based on Field-Programmable Gate Array. Electronic Industry Press. 2015:127-132