The Application of FPGA in the Field of Relay Protection

Jianguo Ruan [1], Jiajun lin[1]

[1] Department of electronic engineering, East China University of science and technology, 130 Meilong Road, Shanghai 200237, China

{jgruan, jjlin}@ecust.edu.cn

Abstract

This paper introduces an implementation method of relay protection algorithms based on FPGA. Now 16-bit MCU is always used as the main processor in most of digital relay protection device. But the performance of this kind of device is frequently affected by the MCU operation speed and some ways to balance the contradiction between the speed and precision must be adopted such as predigest the algorithm or reduce the sampling frequency. So it cannot usually meet the practical application. The main characteristic of the new method related in this paper is that FPFA chip can be used as the main computation unit in the relay protection device so that high-speed sampling and complicated algorithm filtering can be accomplished by FPGA instead of by MCU. Simulation results show that the response time of the system filtering is about 5.78μS and much less than that of the system based MCU. In this way, the contradiction between the speed and precision can be conciliated effectively in the case of applying 8-bit or 16-bit MCU.

Keyword: FPGA, protection algorithms, ASIC, sequence component

I. Introduction

Relay protection is a very important part of electric equipments and systems for stability and security in fields of power system, mine, metallurgy and chemical industry. Traditional relay protection devices can only meet some of simple requirements of relay protection because the limitation of the device structure principle. With the development of the MCU technology, some digital relay protection device based on MCU and corresponding protection algorithms have recently appeared and replaced the traditional relay protection devices in some situation. Now 16-bit MCU is always used as the main processor in most of digital protection device. Its advantage is having the capability of the judgment of complicated faults and enhancement of monitor precision and action speed presently. But their processing speed and ability are often unsatisfied in analyzing complicated faults signals and processing some filtering algorithms such as least square filtering. In most cases of fault analysis, it is usually necessary to extract the amplitude of fundamental harmonic and higher order harmonics (also include positive and negative sequence components) from fault signals. These will be faced with a lot of filtering computation work and spend too much time. For example, full-cycle Fourier filtering algorithm will spend many a milliseconds to extract the amplitude of fundamental harmonic and least-square filtering algorithm will spend tens of milliseconds to do so. Therefore this kind of relay protection device based on MCU usually does not meet the requirements of the practical application. Some measures are usually adopted to balance the contradiction between the speed and precision [1]. These measures generally include predigesting the filtering algorithm (such as reducing the precision of the algorithm coefficients) or reducing the sampling frequency (for example choose *fs*=600Hz) or avoiding use complex algorithm (such as least square filtering). But the above measures will inevitably affect the filtering algorithm precision and cannot be adopted in some cases of fault analysis. The solution approaches to the problem are presenting new and more effective algorithm or applying more advanced IC technology. This paper introduces a method applying FPGA to relay protection system. The idea of the system design is that the computation work of filtering algorithm is accomplished by FPGA instead of by MCU. In this way, high-speed sampling and complicated algorithm filtering can be adopted in the system by means of high-speed performance of the FPGA. Simulation results show that the response time of the system filtering is about several microseconds and much less than that of the system based MCU. Thus, the FPGA chip can be used as the core unit to perform the work of high-speed sampling and filtering algorithm and the contradiction between speed and precision can be conciliated. At the same time, the operation speed requirement for MCU is greatly reduced and 8-bit MPU can be applied into the system.

II. Algorithms Presentation

A. *Full-cycle Fourier Filtering*

At present, full-cycle Fourier filtering algorithm is mostly adopted in relay protection to extract the component of the fundamental harmonic. Its discrete formulas are:

$$a_1 = \frac{2}{N}\sum_{k=1}^{N-1} x(k)\sin k\frac{2\pi}{N} \qquad \textbf{(1)}$$

$$b_1 = \frac{2}{N}\left[\frac{x(0)}{2} + \sum_{k=1}^{N-1} x(k)\cos k\frac{2\pi}{N} + \frac{x(N)}{2}\right] \qquad \textbf{(2)}$$

N in the formula is the number of the sampling points. *x*(0) and *x*(N) are respectively the value of sampling of k=0 and N. The amplitude of the fundamental harmonic can be obtained from:

$$X_1 = \sqrt{a_1^2 + b_1^2}$$

B. *Least Square Filtering*

If there is noise in the signal (such as frequency deviation) besides the higher order harmonics, the full-cycle Fourier filtering will produce much larger error. So we can use the least square filtering algorithm. Assuming that sampling signal is given by:

$$y(t) = X_0 + \sum_{k=1}^{L} X_k \sin(k\omega t + \alpha_k) = X_0 + \sum_{k=1}^{L}(X_{sk}\sin k\omega t + X_{ck}\cos k\omega t) \qquad \textbf{(3)}$$

When the sampling period is Ts=1/fs, from equation (**3**), N equations can be obtained by N sampling points. Express into a matrix form:

$$\boldsymbol{AX=Y}$$

According to the theory of the least square filtering:

$$\boldsymbol{A^T AX=A^T Y}$$

Because $\boldsymbol{A^TA}$ is nonsingular square matrix [2][3], $\boldsymbol{X}$ can be solved by:

$$\boldsymbol{X=(A^TA)^{-1}A^TY} \tag{4}$$

Where:

$\boldsymbol{X} = (X_0, X_{S1}, X_{C1}, X_{S2}, X_{C2}, \dots ,X_{SL}, X_{CL})^T$

$\boldsymbol{Y} = (y_1, y_2,\dots, y_N)^T$

$$\boldsymbol{A} = \begin{bmatrix} 1 & \sin\omega T_S & \cos\omega T_S & \cdots & \sin L\omega T_S & \cos L\omega T_S \\ 1 & \sin 2\omega T_S & \cos 2\omega T_S & \cdots & \sin 2L\omega T_S & \cos 2L\omega T_S \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & \sin N\omega T_S & \cos N\omega T_S & \cdots & \sin NL\omega T_S & \cos NL\omega T_S \end{bmatrix}$$

X_{SL} is the sinusoidal component of the fundamental harmonic and X_{CL} is the cosinusoidal component of the fundamental harmonic so the magnitude of the fundamental harmonic component can be obtained from:

$$X_1 = \sqrt{X_{s1}^2 + X_{c1}^2}$$

C. *Sequence Component Filtering*

The sequence filter is a mathematical model of symmetrical component method, which is built based on the rotary magnetic field theory. In the occasion of no neutral line or small current ground, the two phases' sequence component filter is adopted:

$$\dot{I}_1 = 1/\sqrt{3}\left[I_{as1}/2-\sqrt{3}I_{ac1}/2+I_{cs1})+j(I_{ac1}/2+\sqrt{3}I_{as1}/2+I_{cc1})\right]=I_{1r}+jI_{1i} \tag{5}$$

$$\dot{I}_2 = 1/\sqrt{3}\left[(I_{as1}/2+\sqrt{3}I_{ac1}/2+I_{cs1})+j(I_{ac1}/2-\sqrt{3}I_{as1}/2+I_{cc1})\right]=I_{2r}+jI_{2i} \tag{6}$$

The Ias1, Iac1, Ics1 and Icc1 are respectively the real and image parts of the fundamental harmonic of the phase A and phase C currents. They can be solved by the least square filtering.

III. Feasibility of FPGA Implementation

There are a lot of series of devices since the FPGA technology appears, but only two series of FPGA device have the characteristic of the most attractive DSP algorithms at present. They are XC4000 series of Xilinx and FLEX10K series of Altera. The latter is a very representative series of device in the FPGA. Its unique logic structure (EAB and LAB) has revolutionarily changed compared with traditional programmable logic structure. Therefore, it possesses outstanding performances and great flexibility and adaptability. High performance and low cost have already made FLEX10K devices become an ASIC designer's first-selected device.

Though the development of VLSI technology makes the number of equivalent gates of large capacity programmable logic device (FPGA) increase rapidly, realization of a systematic function still need one or many slices of ASIC and CPU, memory, A/D converter and some other analog circuits to finish work in many situations [4]. According to the respects of complexity, stability and cost of the relay protection device, the design should realize the functions required by combining one chip of ASIC with MCU, A/D chip and some other analog circuits. So this system adopts the

EPF10K10 series device of FLEX10K with higher performance-cost ratio. Adopting the bottom-to-top design method and applying the methods of the algorithms pretreatment and of gate-level function module and the pipelining and making use of the finite hardware resources to the maximum extent, the required algorithms function can be embedded into FPGA chip.

IV. The Implementation Method

A. *The Planning of the Function Module and Its Operation Principle in FPGA*

According to the function of the system and algorithm principle, several function modules should be planed and be made operate in line inside ASIC to achieve the design objects (Fig.1). It can be seen that the plan of these modules is based on MAC characteristic of most DSP-algorithms, namely this is consistent with the characteristic of the devices selected. In fact, the algorithm adopted in this paper just has this characteristic, such as equation (**1**) and (**2**). For equations (**4**), (**5**) and (**6**), we can also make them have MAC characteristic through proper process. Fig.1 also shows that the calculation of amplitude of the sequence component is also designed in it.

Function of ASIC: 1) Sample the A/D value of the A and C phase current simultaneously. 2) Calculate the sine component (Ias_1 and Ics_1) and cosine component (Iac_1 and Icc_1) of the fundamental harmonics of two phases current respectively using the filtering algorithm. 3) Calculate the real parts I_{1r}, I_{2r} and image parts I_{1i}, I_{2i} of the positive sequence and negative sequence current with the sequence component filter. 4) Calculate the I_1^2 and I_2^2 values of the positive sequence and negative sequence current. 5) Output the result of the calculation according to the signal of data width. (8/16 bits)

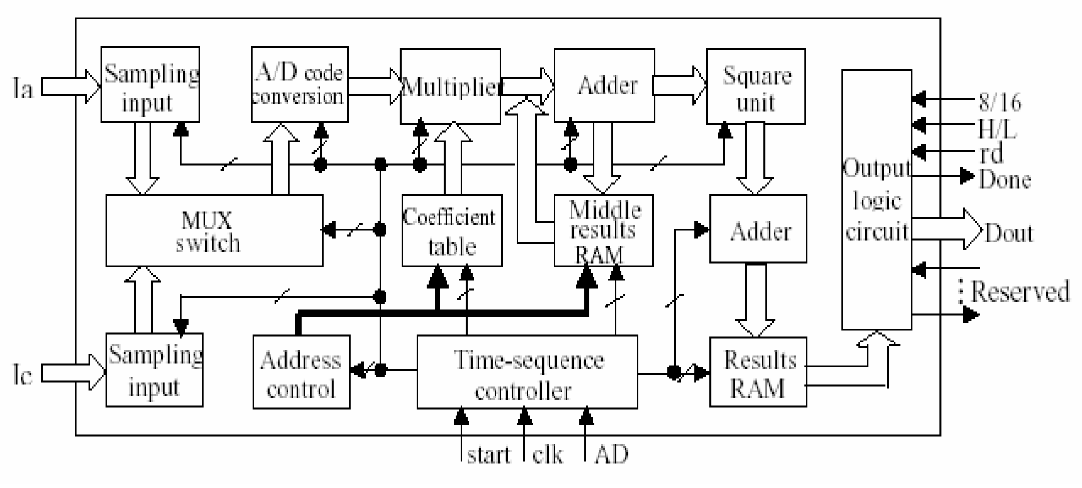


Fig. 1. Framework of ASIC

B. *Algorithm Pretreatment*

The Fourier filtering and least square filtering have the typical MAC characteristic of most DSP algorithms from equation (**1**), (**2**) and (**4**).We can obtain the MAC speed only confined by array multiplier from FPGA through pipelining structure [5]. In order to imbed the function of the algorithm into FPGA effectively in the case of balance the calculation precision and resource consumption, we must carry on the pretreatment to the algorithm to make it consistent with characteristics of the device.

1. *Full-cycle Fourier Filtering*

Equations (**1**) and (**2**) can be expressed as follows:

$$a_1 = \sum_{k=1}^{N-1}\left(\frac{2}{N}\sin k\frac{2\pi}{N}\right)x(k) = c_0x(0) + \sum_{k=1}^{N-1}c_kx(k) + c_Nx(N) = \sum_{k=0}^{N}c_kx(k) \qquad \textbf{(7)}$$

$$b_1 = \frac{x(0)}{N} + \sum_{k=1}^{N-1}\left(\frac{2}{N}\cos k\frac{2\pi}{N}\right)x(k) + \frac{x(N)}{N} = d_0x(0) + \sum_{k=1}^{N-1}d_kx(k) + d_Nx(N) = \sum_{k=0}^{N}d_kx(k) \qquad \textbf{(8)}$$

Where: $c_0 = c_N = 0$, $c_k = \frac{2}{N}\sin k\frac{2\pi}{N}$, $d_0 = d_N = \frac{1}{N}$, $d_k = \frac{2}{N}\cos k\frac{2\pi}{N}$

Now equation (**7**) and (**8**) have the same MAC form. Coefficients c_k and d_k can be calculated off-line and set them into ASIC beforehand. The results of c_k calculated off-line and 7bit approach values of corresponding binary fraction are listed in table 1 (N=20). The coefficient in the table 1 shows the highest 3 bits of the coefficients are zero and the lowest 4 bits are used in the operation. If the 12-bit A/D converter is used, we only need to establish an 11×4 multiplier instead of an 11×7 multiplier. Thus chip resources can be saved and speed of the multiplier can be increased.

Table 1. Off-line calculation value and corresponding binary value of C_k (N=20)

k	Decimal C_k	Binary C_k	k	Decimal C_k	Binary C_k
0	0	0.0000000	11	– 0.03090170	– 0.0000100
1	0.03090170	0.0000100	12	– 0.05877853	– 0.0001000
2	0.05877853	0.0001000	13	– 0.08090170	– 0.0001010
3	0.08090170	0.0001010	14	– 0.09510565	– 0.0001100
4	0.09510565	0.0001100	15	–1	– 0.0001101
5	1	0.0001101	16	– 0.09510565	– 0.0001100
6	0.09510565	0.0001100	17	– 0.08090170	– 0.0001010
7	0.08090170	0.0001010	18	– 0.05877853	– 0.0001000
8	0.05877853	0.0001000	19	– 0.03090170	– 0.0000100
9	0.03090170	0.0000100	20	0	0.0000000
10	0	0.0000000			

2. *Least Square Filtering*

Equation (**4**) shows that $\boldsymbol{(A^TA)^{-1}A^T}$ is a (2L+1)×N matrix, let:

$$\boldsymbol{C} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T = \begin{bmatrix} c_{00} & c_{01} & c_{02} & \cdots & c_{0N} \\ c_{s10} & c_{s11} & c_{s12} & \cdots & c_{s1N} \\ c_{c10} & c_{c11} & c_{c12} & \cdots & c_{c1N} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ c_{sL0} & c_{sL1} & c_{sL2} & \cdots & c_{sLN} \\ c_{cL0} & c_{cL1} & c_{cL2} & \cdots & c_{cLN} \end{bmatrix}$$

From equation (**4**) we can obtain as follows:

$$X_{s1} = \sum_{k=1}^{N} c_{s1k} y_k \tag{9}$$

$$X_{c1} = \sum_{k=1}^{N} c_{c1k} y_k \tag{10}$$

Now equation (**9**) and (**10**) have the same MAC structure. Table 2 lists the results of the C_{s1k} and C_{c1k} out line calculation. The table uses N=11 to compare with full-cycle Fourier filtering. If the sampling frequency *fs*=1000Hz, the least square filtering only need 10ms(N=11) to converge. If the sampling frequency *fs*=2000Hz, the least square filtering only need 7ms(N=14) to converge. Its speed is much higher than full-cycle Fourier filtering.

Table 2. Off-line calculation value of C_{s1k} and C_{c1k}(N=11)

k	C_{Slk}	k	C_{Slk}	k	C_{Clk}	k	C_{Clk}
1	– 20.2650	7	– 885.7215	1	3.2097	7	– 18.0661
2	109.6669	8	609.5600	2	– 10.9502	8	25.6736
3	– 313.1336	9	– 313.1336	3	21.2757	9	– 21.2757
4	609.5600	10	109.6669	4	– 25.6736	10	10.9502
5	– 885.7215	11	– 20.2650	5	18.0661	11	– 3.2097
6	999.7863			6	0		

We can find that the coefficient of least square filtering is more discrete than that of full-cycle Fourier filtering as in table 2. So more binary bits should be used to approach the binary value of the discrete. (20 bit in this design) This means that the 11×20 multiplier should be built. Chip resource will be expended a lot. So the discrete should be processed separately as equation (**9**) and (**10**):

$$X_{s1} = \sum_{k=1}^{N} (c_{s1k1} + c_{s1k2} + c_{s1k3} + c_{s1k4}) y_k = \sum_{j=1}^{N} \sum_{k=1}^{4} c_{s1jk} y_j \tag{11}$$

$$X_{c1} = \sum_{k=1}^{N} (c_{c1k1} + c_{c1k2} + c_{c1k3} + c_{c1k4}) y_k = \sum_{j=1}^{N} \sum_{k=1}^{4} c_{c1jk} y_j \tag{12}$$

C_{s1jk} and C_{c1jk} are 5 bits binary values with right, so we only need to establish an 11×5 multiplier.

3. *Sequence Component Filtering*

From equations (**5**) and (**6**) we can obtain as follows:

$$I_{1r} = \sum_{k=1}^{3} a_k I_{rk} \quad , \quad I_{1i} = \sum_{k=1}^{3} b_k I_{ik} \quad , \quad I_{2r} = \sum_{k=1}^{3} b_k I_{rk} \quad , \quad I_{2i} = \sum_{k=1}^{3} a_k I_{ik}$$

Where: $a_k = \frac{1}{2}, 1, \frac{\sqrt{3}}{2}$ $\qquad I_{rk} = I_{as1}, I_{cs1}, I_{ac1}$

$b_k = \frac{1}{2}, 1, \frac{\sqrt{3}}{2}$ $\qquad I_{ik} = I_{ac1}, I_{cc1}, I_{as1}$

The above four equations and equations (**7**), (**8**), (**11**) and (**12**) have the typical MAC form. So they can share the same type of calculators.

C. Implementation Method with VHDL

1. Design Method with VHDL

When adopt the language VHDL to design, the occupancy of the abstractness and resource of design is in direct ratio [6]. So this design is based on structured-description method. The principle is: In multi-level design, high-level module calls low-level module, or some complicated logic units with gate-grade (such as array multiplier) are directly designed. Namely the design is made from bottom to top and structured-description method is used mainly. This method can save chip resources most and is most effective.

2. The Establishment of the Function Module

The establishment of the function modules is as follows:

1) The coefficient table (ROM) and the mid operation result register (RAM) are established through LPM. Its characteristic is that it can map PLDs, gate array and standard crystalline grain effectively. The designer can transplant it conveniently on higher logic level. We can design the module of resources with very high efficiency with LPM .In this design, we just use one EAB to establish ROM module and none of the LCs is used.

2) The establishment of multipliers, power unit and address producers is based on gate-grade design. Explanation: 1) Multiplier and power unit are made by array structure instead of S/P or S/S structure although the two structures can save many LC units. Thus high speed can be achieved by the sacrifice of resource. This is also the method that generally adopted in DSP processor at present. 2) VHDL supports multiplication and square operation, but consumption of LC units is enormous and it is difficult to implement in practice. As for the adder unit, the efficiency of operation supported by VHDL has even exceeded that of gate-grade design because of LAB structure characteristic of FLEX10K series. So the addition operation symbol is used directly in this design.

3) Because RTL-description method is very efficient for signals’ transfer among the function modules. So time-sequence controller, I/O logic and the communication among the function modules are established in the way of structured and RTL description mode.

D. The Use Status of Chip Resource

Table 3 lists the use status of chip resources for realizing full-cycle Fourier filtering. It shows that the rate of utilization of EAB module has reached 100%. Due to fully utilized high-efficient characteristic of EAB module that enables the design to reach very high density. Table 3 also shows that there are remaining logic resources in the chip, it can support the logic control of system interface and action export.

Table 3. The use status of chip resource (EPF10K10LC84)

Resource	Used	Resource	Used
Total I/O pins	50/53	Total EABs	3/3 (100%)
Total logic cells	516/576	Average fan-in	3.32/4 (83%)
Total embedded cells	21/24 (87%)	Total fan-in	1717/2304

V. Performance

A. *Filtering Response Speed*

Restricted by speed of operation, lower sampling frequency (*fs*=600Hz) is usually chosen in the device based on 16-bit MCU at present and we should input all sampling values to start the filtering computation. So it is hard to enhance the response speed. In this design, filtering computation is accomplished point-wise. Fig.2 is about a simulating wave during a period of time (*fs*=1000Hz). For observation, the sampling interval is reduced from 1ms to 2μS. Because of the high speed of FPGA, we only need 1.78μS to do filtering operation to the two phase-currents at some time (20MHz). This speed can even make sampling frequency reach *fs* =500KHz, obviously the 16-bit MCU can't be realized.

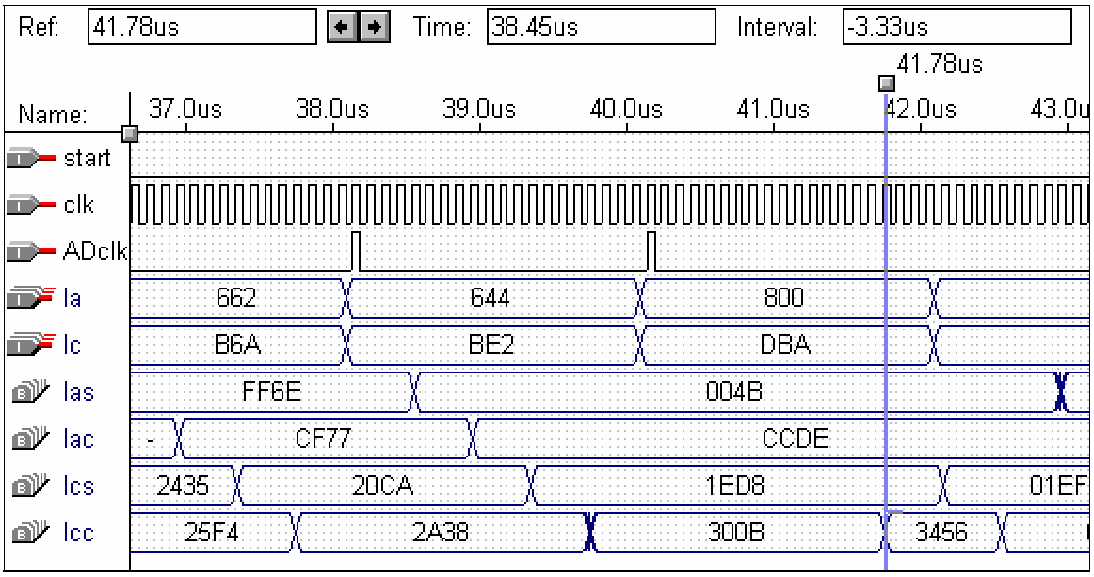


Fig. 2. Simulation wave of filtering operation

Fig.3 is about the simulating wave from the last sampling point (40μS) to the output of sequence component result (upright line). Its operation time is about 5.78μS and it is also the response time of the system filtering. The filtering operation of sequence component doesn't relate to the number of sampling points (N). The operation time only relates to the system clock. Therefore, it is no use to the response speed of system filtering if we increase the sampling points (increase the sampling frequency). Obviously, the 16-bit MCU can't realize it.

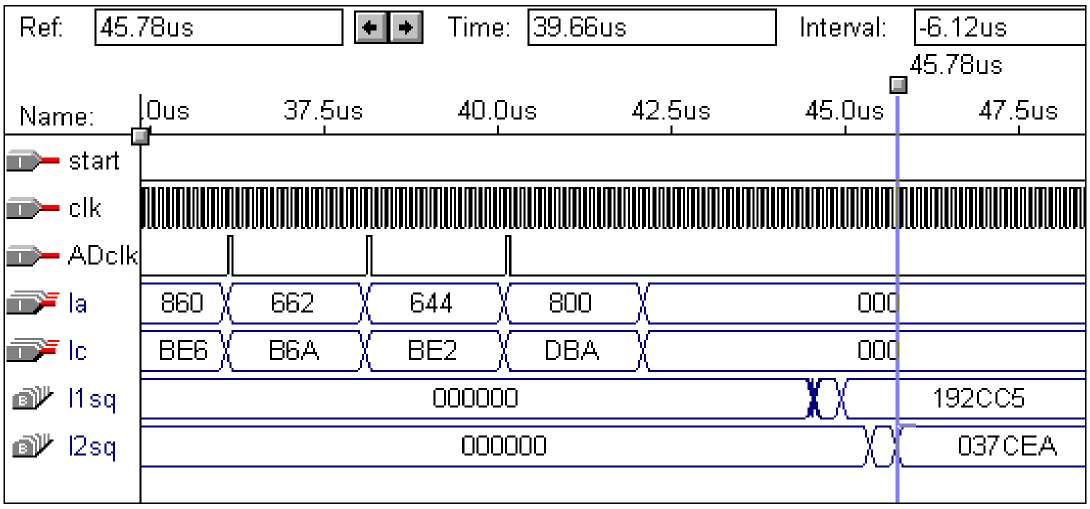


Fig. 3. Simulation wave of sequence component calculation

B. *Filtering Precision*

Typical short-currents (presume A phase and C phase) are as follows:

$$i_A = 10 - 10\sin(\omega t + \tfrac{1}{2}\pi) + 2\sin 2\omega t + 5\sin 3\omega t + 0.5\sin 4\omega t + 0.2\sin 5\omega t$$
$$i_C = 10 - 12\sin(\omega t - \tfrac{2}{3}\pi) + 2.5\sin(\omega t - \tfrac{1}{2}\pi) + 3\sin 3\omega t + \sin 4\omega t + 0.5\sin 5\omega t$$

The real parts and imaginary parts of the phase A and phase C of the current of the fundamental harmonic are:

$$I_{as} = 0, \qquad I_{ac} = -10, \qquad I_{cs} = 6, \qquad I_{cc} = 10.3932$$

We can calculate the positive and negative sequence components from (**5**) and (**6**):

$$I_1 = 9.0185 \ (I_1^2 = 81.3333)$$

$$I_2 = 3.4715 \ (I_2^2 = 12.0513)$$

We can know the value of Ias, Iac, Ics and Icc from Fig.2.The data width is 16 bits and the low 4 bits is decimal fraction. We can know the square value of the positive sequence component and negative sequence component (I_{1sq} and I_{2sq}) from Fig.3. The data width is 22 bits.000H~7FFH are 0~25V in Fig.3. Table 4 lists the components, their magnitude and the comparison between predicted value and actual value. Its operation precision is the same as 16-bit processor, but its precision is higher because of the higher sampling frequency.

Table 4. Filtering results and error

Component	Filtering result (Hex Value)	Corresponding amplitude	Theory value	Error
Ias	004B	0.0488	0	——
Iac	CCDE	– 9.9976	–10	0.24%
Ics	1ED8	6.0180	6	0.30%
Icc	3456	10.2173	10.3923	1.68%
I1sq (I_1^2)	192CC5	81.9492	81.3333	0.76%

I2sq (I_2^2)	37CEA	11.8540	12.0513	1.64%

VI. Conclusion

The application value of FPGA in the field of DSP is revealed increasingly for the characteristics such as high speed, pipeline and MAC of FPGA. So it is inevitable that FPGA is applied in the field of relay protection. The design method in this paper is only a preliminary try of the implementation of digital protection algorithm based on FPGA, and we can assuredly see its advantage.

References

[1] W. J. Wang, *The Fundamental Theory of Relay Protection in Power System*, China Power Publishing House, Beijing, 1996, pp. 2-5.

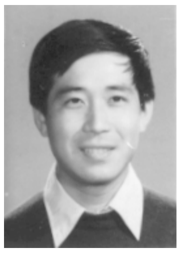
[2] Sachdev. M.S and Baribeau. M.A, “A New Algorithm for Digital Impedance Relays”, *IEEE Trans*. on PAS, vol. PAS-98, 1979, pp. 2232-2240.

[3] Sachdev. M.S and Nagpal. M, “A Recursive Least Error Squares Algorithm for Power System Relay and Measurement Applications”, *IEEE Trans on Power Delivery*, vol.6, no.3, 1991, pp.1008-1015.

[4] C. Y. Xin, *VHDL Hardware Description Language*, National Defence Industry Publishing House, Beijing, 2002, pp. 88-92.

[5] R. Peterson and B. Hutchings, “An Assessment of the Suitability of FPGA-Based Systems for Use in Digital Signal Processing”, *Lecture Notes in Computer science* 975, Springer, Heidelberg, 1995, pp. 293-302.

[6] L. Gan, *Application and Development of VHDL*, Science Publishing House, Beijing, 2003, pp. 108-112.



Jianguo Ruan is currently an associate professor in Department of Electronic and Communication Engineering, the East China University of Science and Technology. His research interests include signal processing, measurement and control.

Jiajun Lin received the Ph.D. degree in signal processing from the Tsinghua University in 1998. He is currently a professor in the East China University of Science and Technology. His research interests include signal processing and data fusion.