The Algorithm of Direct Current Smooth Filtering Based on the Minimum Resource Consumption in FPGA

Pengfei Hu, Li Shen, Liping Liu
Guizhou Institute of metrology
shineliuliuling@seu.edu.cn

Abstract. The algorithm of direct current smooth filtering was proposed in this paper to satisfy the requirement of minimum resource consumption in FPGA and to meet the goal of calculation period in minimum length. In this work, the detail circuit was designed by FPGA and the key circuit was simulated by the modalism. The result of board-level test is shown that the dither amplitude of original direct current was decreased significantly at 90%. The amplitude was in the stable state to reach expected smoothing effects.

1. Introduction
Lock-in amplifier technique is one of the important methods for weak signal detection[1]. The lock-in amplifier uses a cross-correlation principle design of synchronous detection apparatus[2]. Usually in the measurement, noise is an interference signal, it is restricted and impacts the sensitivity of a measuring instrument and is difficult to shield. In order to reduce the effect of noise on the useful signal and improve the SNR[3], the normal procedure is using narrow band filter to remove the noise out of the band. Because of the problem of center frequency instability and Q value of general filter, it cannot meet the high requirements of filtering noise. The measured signal and referenced signal are synchronously measured from the lock-in amplifier. It does not have the frequency stability problem, and can be regarded as a tracking filter. Its equivalent Q value is decided by the time constant of low pass filter, so the stability of the elements and the environmental requirement is not high.

The comparison reference signal was set as the measured signal and the reference signal with the same phase relationship. It only responds to the measured signal itself and those noise components which have the same frequency and phase with the reference signal. In essence, it is an analog Fourier transformer that it has a output in a DC voltage which proportional to the signal amplitude of a specific frequency in the input signal. Generally, in order to filter out the Gauss noise in DC output better, smoothing filter is needed before DC output to ensure the stability and repeatability of output.

In this paper, a digital DC smoothing filter algorithm was proposed to meet the goal of minimizing the resource consumption and computing cycle of the FPGA. The specific circuit of the algorithm is designed, the key circuit is simulated, and the design goal is verified by the actual board level test.

2. Basic Algorithms of DC Smoothing Filtering
The DC smoothing filter algorithm can equalize the DC which is heavily disturbed by Gauss noise, averages the DC in a certain period of time, and outputs the results in the form of pipeline. The final result is the most stable amplitude in the shortest time. Two indexes need to be guaranteed in the design, the time-consuming and the resource-consuming. It has a relatively simple structure which only has the sum circuit, multiplication circuit and division circuit.
2.1. The sum circuit

The FPGA is good at doing the summation operation. Generally, two data with bit width of D should be added together, and at least the data with the bit width of D+1 should be used to store. The basic summation method is to beat the data continuously and the sum them after alignment. Assuming the length of the data is 5, the summation circuit can be designed according to figure 1. The block diagram of sum circuit is shown in figure 1. Five beats of the data was made up of the current data after two and two data additions of three clocks, the sum of five consecutive data can be obtained.

As shown in the figure 1, the resource consumption of this circuit has 4 adders, 6 registers and 3 clocks. However, incremental updating method is used to realize the lateral summation of windows, which is very useful in large-scale window computation. Assuming that the current computed address is n+1, the computed result is Y(n+1), the previous address is n, the computed result is Y(n), and the input data stream is X(i), then there are:

\[
Y(n) = \sum_{i=-2}^{2} X(n+1) \tag{1-1}
\]

\[
Y(n+1) = \sum_{i=-2}^{2} X(n+1+i) \tag{1-2}
\]

\[
= \sum_{i=-2}^{2} X(n+i) + X(N+3) - X(n-2)
\]

This shows that for each window, there is no need to recalculate all the data in the window, and the new data can be obtained by adding the previous data and then calculating the difference between the new point and the discarded point.

![Figure 1 The block diagram of sum circuit with 5 data.](image)

As shown in Figure 2, this circuit only needs one adder and one subtractor. It can be predicted that no matter how big the data size is, the required adder and subtractor are all one. Therefore, when the data length is relatively large, the ratio can be obtained.

2.2. The design of multiplication and division circuits

For multiplication and division operations with fixed denominator, they can be transformed into shift, addition and multiplication operations which are good at by FPGA through Taylor expansion and shift conversion.
If the effective value of voltage is \( R \) which was converted by a digital/analog system, the reference voltage is 5V and the digit quantization bit width is 14 bits, the max value \( \text{Max} \) needs to be solved. According to the relationship between RMS and \( \text{Max} \) the equation is shown in following.

\[
\text{Max} = \frac{R}{16384} \times 5 \times \sqrt{2}
\]

\[
\approx \frac{R}{16384} \times 7.071068
\]

\[
\approx \frac{R}{16384} \times (2^2 + 2^1 + 2^0 + 2^{-4} + 2^{-7} + 2^{-10})
\]

The calculation error of the formulation (1-3) is \( E \).

\[
E = \left| \frac{\frac{R}{16384} \times (2^2 + 2^1 + 2^0 + 2^{-4} + 2^{-7} + 2^{-10}) - \text{Max}}{\text{Max}} \right| \times 100\%
\]

\[
\approx 0.001069\%
\]

In fact, in the process of calculation, the decimal digits can be preserved by controlling the displacement digits of the final additive results, thus improving the accuracy. For example, to maintain the three decimal digits, the formula (1-3) becomes to the following.

\[
\text{Max} \approx 2^3 \times \frac{R}{16384} \times (2^2 + 2^1 + 2^0 + 2^{-4} + 2^{-7} + 2^{-10})
\]

\[
\approx \frac{R}{2048} \times (2^2 + 2^1 + 2^0 + 2^{-4} + 2^{-7} + 2^{-10})
\]

In this method, the maximum \( \text{Max} \) solution can be converted into 11 shift operations and 6 addition operations, while the shift operation is zero overhead for the FPGA and does not occupy the clock cycle.

Figure 3 The block diagram of multiplication and division circuits.

Similarly, similar conversion methods can be used for data of other lengths. The structure of the division circuit is shown in figure 3.

3. Results and Discussion

The Xilinx Artix-7 FPGA development board was used as the experimental verification system, and AD9767 digital/analog interface board was used to verify summation circuit, multiplication and division circuit and actual board level test was carried out respectively.

In summation circuit validation, the validation data stream is set to \( \text{din} \), and the summation circuit begins to calculate when the \( \text{din_valid} \) level of the valid input signal is raised. Figure 4 shows that at the fifth clock, five data to be summed are obtained. At the next clock, the sum of the five data will be calculated. At the same time, the difference \( \text{diff} \) between the beginning and the end of the data stream is also calculated. In the next clock, the current calculated value plus \( \text{diff} \) is the new summation result \( \text{dout} \).

However, the three clock data 0, 30 and 56 that \( \text{dout} \) just started to output are invalid, because the three clock data are at the boundary. In subsequent design, the three clock data can be set as irrelevant items.
In multiplication and division circuit, the experimental data is set to 14-bit quantized data. After adding three clocks and buffering one clock, the result $dout$ is finally output, while $dout_{frac}$ retains the mean of three decimal digits. The results are listed in Table 1. At the same time, a comparison with the theoretical mean is made, which proves that the result of division is correct. Among them, we rounded the integer output, added the last bit plus 1 processing with 3-bit accuracy, and the operation delay was 4 clocks.

In the board level test, the pulsation of DC waveform to be smoothed is set to 40 mV. After testing system, the experimental results are observed by high resolution oscilloscope. The results are shown in figure 5. The yellow line is the original input waveform, and the blue line is the result of data flowing through the DC smoothing filter of the FPGA. The jitter amplitude is reduced by about 90%, and the amplitude is basically stable which reaching the expected level of smoothing effect.

4. Conclusion

In our experiments, we have studied the circuit structure of DC smoothing filter. Compared with the conventional DC filter circuit structure, in this work, the proposed circuit structure can minimize the on-chip resources of the FPGA and improve the operation cycle. After the actual board-level test, the
The proposed smoothing filter circuit can effectively remove the Gauss noise in the circuit, reduce the jitter amplitude, and achieve the desired smoothing effect.

**Acknowledgment**
The authors are grateful for the valuable help of our co-workers. The authors would like to acknowledge the financial support provided by the National Natural Science and Guizhou Science and Technology Departments. (No. QKH2016108, No. QKH20192881).

**References**
[1] JI Yu, XU Aiwei. A new method for automatically measurement of vickers hardness using thick line through transform and least square method[C]// 2009 2nd International Congress on Image&Signal Processing.2009:1-4.
[2] BRETT C R,RAAD J A. Validation of a pulsed eddy current system for measuring wall thinning through insulation[C]// Proceeding of SPIE,1996,2947-2952.
[3] Roesler E, Nelson B. Debug methods for hybrid CPU/FPGA systems[C]// IEEE International Conference on Field-Programmable Technology.2003:243-250.
[4] Ngo H, Shafer J, Ives R, et al. Real Time Iris Segmentation on FPGA[C]// IEEE International Conference on Application-Specific Systems, Architectures and Processors. IEEE Computer Society,2012:1-7.
[5] Seo Y H, Yoo J S, Kim D W. A new parallel hardware architecture for a high-performance stereo matching calculation[J].Integration the Vlsi Jounal,2015,51(C):81-91.
[6] Bates G L, Nooshabadi S. FPGA implementation of a median filter[C]// Tencon 97.IEEE Region 10 Conference.1998:437-440.
[7] Mou Xingang, Zhang Guilin, Hu Ruolan. A design of real-time scene-based nonuniformity correction system. MIPPR 2009:Multispectral Acquisition and Processing,749422-5.
[8] Si T, Gao Y, Qiao J, et al. A Kind of Real-time Infrared Image Enhancement Algorithm[C]// Infrared materials, Devices and Applications.2007.
[9] R H Hosking. Using FPGAs for software radio systems. http://www.technonline.com/community.
[10] Anil Kandangath, Jamming mitigation techniques for spread spectrum communication system, IEEE 598A:Signal processing for wireless communications. December 9,2003.