

# Research on Digital Correlator Algorithm Based on FPGA

Guiling Sun, Jun Jia, Tianyu Geng, and Xudong Ye

**Abstract**—This paper designs and implements a parallel correlation algorithm for ultra-long sequence in large-scale field programmable gate array (FPGA). The operation speed is 64 times faster than traditional algorithms. Combining multiple data values in one RAM address improves the efficiency and avoids the time consumption caused by reading data repeatedly. The multi-group multiply accumulator is used to calculate the data read according to certain rules, which fully utilizes the advantages of parallel operation in FPGA and reduces the operation time. This algorithm overcomes the shortcomings of the limit of the number of multipliers in the serial pipeline operation method, which can only perform correlation operations on short sequences, and makes up that the general parallel algorithm is more convenient only for the correlation operation of the two signals without relative sliding, but the sequential control is too complex when the two signals have relative sliding.

**Index Terms**—Correlation operation, FPGA, parallel algorithm.

## I. INTRODUCTION

The correlation operation mainly includes autocorrelation and cross-correlation, which respectively indicate the degree of correlation between the same sequence and two different sequences at any two different times. Its powerful ability to suppress noise makes it extremely widely used in the field of signal processing [1]. The main application fields include the signal processing of radar signal receiver [2], [3], accurate positioning by measuring the delay difference of the same signal, the synchronization of the communication system by correlation between the receiving sequence and the local sequence, the decision of the signal in the direct sequence spread spectrum communication and the restoration of the original signal, and the application of the GPS receiver [4].

The traditional correlation operation is implemented by software, but it's difficult to meet the requirements of the real-time performance of the system because of the characteristic of the serial operation in computer, so the operation period will be greatly extended. Using FPGA can make full use of the advantages of its parallel operation, and use the hardware circuit to realize the correlation operation, so that the operation cycle is greatly shortened and the operation time of the system is shortened to the maximum [5], [6].

The shift registers can be used to realize the correlation

operations of pipeline in FPGA. M multipliers are needed for the sequence of length M, the resources of multipliers for the super long sequence are insufficient and the operation time of the serial operation is greatly extended, which can't meet the needs actually. The parallel operation of N multipliers can increase the speed of N times, but the input timing control of data is too complicated. Data must first be stored in the RAM within FPGA for super long sequences, and reading data needs to operate on the address of the RAM, only the data in one address can be read for each clock cycle. One address can be used to access N data and read out in one clock cycle for parallel computing. However, since the second signal sequence needs to be slide right and left, it will occur that the N data required by a certain operation is not in the same RAM address, resulting in the complexity of the sequential. This paper also uses every RAM address to access multiple data, which improves the internal algorithm and overcomes the above shortcomings. Making full use of the ability of parallel computing in FPGA, greatly shortens the computing cycle. The time of the correlation operation of two sequence with length 8000 is shortened from the general 640ms to 10ms in the system clock 100MHZ, thus the speed of operation is greatly improved.

## II. THE PRINCIPLE OF CORRELATION OPERATION

For the sequence  $X=\{x[0],x[1],\dots,x[N-1]\}$  and  $Y=\{y[0],y[1],\dots,y[N-1]\}$  with length N, the correlation function is:

$$R(\tau) = \sum_{n=0}^{N-1} x(n) \times y(n - \tau) \quad (1)$$

The length of  $R(\tau)$  is  $2N-1$ , and the operation needs  $N^2$  times multiplication and  $N^2-2N+1$  times addition. There has a set of new values input to the multiplier circuit each clock and the adder circuit adds the result of the previous multiplier. In general, the time spent per step is only one clock cycle, so the work speed of the system is basically equal to the clock frequency, that is, the operation time is  $N^2$ .

In order to shorten computing time, parallel computation by multi-group multiplication is needed. For example, M parallel multipliers and an accumulator are used, assuming that N is an integer multiple of M and calculating  $R(\tau_0)$ :

$$R(\tau_0) = \sum_{n=\tau_0-k}^{N-1} x(n) \times y(n - \tau_0) \quad (2)$$

K is the remainder of the M division of  $\tau_0$ , here the sequence Y is extended, make  $y(-k),y(-k+1),\dots,y(-1)$  equal to zero, so the number of terms for the sum is also an integer

Manuscript received March 1, 2019; revised May 1, 2019.

Guiling Sun, Jun Jia, Tianyu Geng, and Xudong Ye are with College of Electronic Information and Optical Engineering, Nankai University, Tianjin, China (e-mail: sungl@nankai.edu.cn, {jjajun, gengty, yexudong}@mail.nankai.edu.cn).

multiple of M.

This paper uses the method that every RAM address access M data. The number of the multipliers needed is  $M^2$ , and the number of accumulators is M. They are divided into M group, each M multipliers and an accumulator. The first M values of sequences X and Y are respectively read in the first clock cycle, these are  $x[0], x[1], \dots, x[M-1]$  and  $y[0], y[1], \dots, y[M-1]$ . The M group multiplier and the accumulator perform the following operations, respectively ( $R(\tau)_M$  represents the sum of the previous M data of  $R(\tau)$ ).

$$\begin{aligned} R(0)_M &= \sum_{n=0}^{M-1} x(n) \times y(n) \\ R(1)_{M-1} &= \sum_{n=0}^{M-1} x(n) \times y(n-1) \\ &\vdots \\ R(M-1)_1 &= \sum_{n=0}^{M-1} x(n) \times y(n-(M-1)) \end{aligned} \quad (3)$$

The M-th to (2M-1)-th values of sequences X and Y are respectively read in the second clock cycle, these are  $x[M], x[M+1], \dots, x[2M-1]$  and  $y[M], y[M+1], \dots, y[2M-1]$ . The M group multiplier and the accumulator perform the following operations, respectively.

$$\begin{aligned} R(0)_{2M} &= R(0)_M + \sum_{n=M}^{2M-1} x(n) \times y(n) \\ R(1)_{2M-1} &= R(1)_{M-1} + \sum_{n=M}^{2M-1} x(n) \times y(n-1) \\ &\vdots \\ R(M-1)_{M+1} &= R(M-1)_1 \\ &+ \sum_{n=M}^{2M-1} x(n) \times y(n-(M-1)) \end{aligned} \quad (4)$$

Proceed in accordance with the same rule. When the final M data operation of sequence X and Y are completed, the values of  $R(0), R(1), \dots, R(M-1)$  are gotten. Then the remaining values are calculated with the same rule, which will be introduced in detail in the next section.

### III. THE IMPLEMENTATION OF CORRELATION OPERATIONS IN FPGA

The input analog signals are sampled by dual AD acquisition module to get the sequence X and Y [7-9]. The eight thousands of 12bit data are obtained after one sample is finished. Data is written to RAM at the same time of data acquisition. Each eight data is combined in the sampling order and written to the same RAM address. The next eight data is written to the next RAM address, and so on. Each sequence occupies one thousand of RAM address, with a total storage space of  $9.6 \times 10^4$  bit, when data acquisition is completed.

After data collection is completed, the data storage status in RAM is shown in Table I, and the data in each address is 96bit.

TABLE I: DATA STORAGE STATUS OF RAM AFTER DATA ACQUISITION IS COMPLETED

addr	0	1	...	998	999
RAM1	x[0-7]	x[8-15]	...	x[984-991]	x[992-999]
RAM2	y[0-7]	y[8-15]	...	y[984-991]	y[992-999]

Table II shows the state of the storage of eight data in address 0, in which  $x[0]$  to  $x[7]$  and  $y[0]$  to  $y[7]$  are both signed numbers of 12bit, and the storage state in other addresses is similar to this.

TABLE II: DATA STORAGE STATUS IN ADDRESS 0

addr	0							
RAM1	x[0]	x[1]	x[2]	x[3]	x[4]	x[5]	x[6]	x[7]
RAM2	y[0]	y[1]	y[2]	y[3]	y[4]	y[5]	y[6]	y[7]

After sampling, correlation operations are started, use Quartus II as software platform and Verilog HDL language. The adopted FPGA belongs to the series of CYCLONE IV, and the model is EP4CE15F23C8. The frequency of the system clock is 50MHZ, and it's doubled to 100MHZ by phaselocked loop in order to improve the speed of operation.

64 multipliers and 8 accumulators are configured in FPGA since 8 data are accessed in each RAM address. They are divided into 8 groups, that is, 8 multipliers and 1 accumulator in each group, to meet the operation rules. In addition, we need to define a register array C with a data length of 8, which is used for the extension of sequence Y and for storing data read from the previous clock cycle on RAM2. The initial values are all set to 0. As shown in Table III.

TABLE III: THE REGISTER C

Register C	C[0]	C[1]	C[2]	...	C[7]
------------	------	------	------	-----	------

In this paper, take the sequence Y to the right slide as an example, that is, the part of  $\tau \geq 0$  in the correlation operation result  $R(\tau)$ . The data of address 0 in RAM1 and RAM2 is read, a total of 16, at the first clock cycle. The 16 data, together with the 8 data of array C(the initial value is 0), are sent to 64 multipliers for parallel operation according to the following rules.

In the first group, the data in RAM1 and RAM2 multiply correspondingly, that is,  $x[0] \times y[0], x[1] \times y[1] \dots x[6] \times y[6], x[7] \times y[7]$ . As shown in Fig. 1.

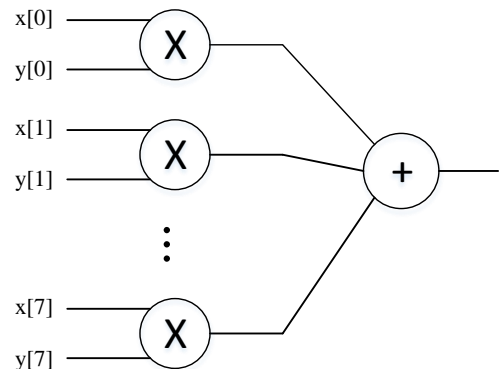


Fig. 1. The operation rule of first group.

In the second group, the sequence Y slide one to the right,

at the same time, expands a data  $y[-1]=0$ , which is located in  $C[7]$ . Multiply correspondingly again, that is,  $x[0] \times y[-1]$ ,  $x[1] \times y[0] \dots x[6] \times y[5]$ ,  $x[7] \times y[6]$ . As shown in Fig. 2.

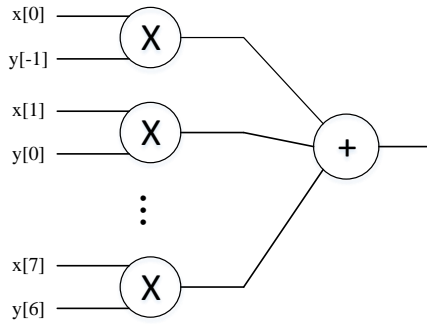


Fig. 2. The operation rule of second group.

In the third group, the sequence Y slide two to the right, at the same time, expands a data  $y[-1]=0$ , which is located in  $C[7]$ , and expands a data  $y[-2]=0$ , which is located in  $C[6]$ . Multiply correspondingly again, that is,  $x[0] \times y[-2]$ ,  $x[1] \times y[-1] \dots x[6] \times y[4]$ ,  $x[7] \times y[5]$ . As shown in Fig. 3.

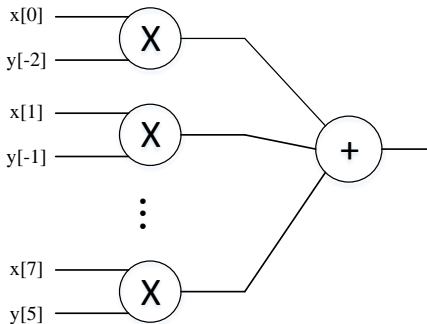


Fig. 3. The operation rule of third group.

According to the same rule, the sequence Y slide seven to the right in the eighth group. Multiply correspondingly again, that is,  $x[0] \times y[-7]$ ,  $x[1] \times y[-6] \dots x[6] \times y[-1]$ ,  $x[7] \times y[0]$ .

The eight results of each group are sent to the parallel accumulator of the group. In this way, the result of the first accumulator is the multiplication and accumulation of the first 8 sets of data when the sequence Y has no relative sliding to the sequence X, that is,  $R(0)_8$ . The result of the second accumulator is the multiplication and accumulation of the first 7 sets of data when the sequence Y slides one to the right, that is,  $R(1)_7$ . The result of the third accumulator is the multiplication and accumulation of the first 6 sets of data when the sequence Y slides two to the right, that is,  $R(2)_6$ . And so on, the result of the eighth accumulator is the multiplication and accumulation of the first 1 set of data when the sequence Y slides seven to the right, that is,  $R(7)_1$ .

The data of address 1 in RAM1 and RAM2 is read, a total of 16, at the second clock cycle. At the same time, the data of address 0 in RAM2 is written to the array C to cover the original zero value, that is,  $C[0]=y[0]$ ,  $C[1]=y[1] \dots C[7]=y[7]$ . Repeat the 8 groups of operations again. In this way, the result of the first accumulator is the multiplication and accumulation of the first 16 sets of data when the sequence Y has no relative sliding to the sequence X, that is,  $R(0)_{16}$ . The result of the second accumulator is the multiplication and accumulation of the first 15 sets of

data when the sequence Y slides one to the right, that is,  $R(1)_{15}$ . The result of the third accumulator is the multiplication and accumulation of the first 14 sets of data when the sequence Y slides two to the right, that is,  $R(2)_{14}$ . And so on, the result of the eighth accumulator is the multiplication and accumulation of the first 9 set of data when the sequence Y slides seven to the right, that is,  $R(7)_9$ .

The data of address 2 in RAM1 and RAM2 is read, a total of 16, at the third clock cycle. At the same time, the data of address 1 in RAM2 is written to the array C to cover the original value, that is,  $C[0]=y[8]$ ,  $C[1]=y[9] \dots C[7]=y[15]$ . Repeat the 8 groups of operations again. Until the final operation of data in address 999 is completed, the values of accumulators is the results of the correlation operation when the sequence Y slide 1, 2  $\dots$  7 to the right, that is,  $R(0)$ ,  $R(1)$ ,  $R(2) \dots R(7)$ . The above is a cycle.

In the second cycle, the register array C is cleared, the data is read from the address 0 in RAM1, and address 1 in RAM2. Until the final operation of data in address 999 is completed, the values of accumulators is the results of the correlation operation when the sequence Y slide 8, 9  $\dots$  15 to the right, that is,  $R(8)$ ,  $R(9)$ ,  $R(10) \dots R(15)$ . In turn, the correlation operations of the sequence Y sliding to the right are completed in the end, that is to say, the part of  $\tau \geq 0$  in the correlation operation result  $R(\tau)$ . Similarly, it has the same rule for the sequence Y sliding to the left and sliding to the right.

#### IV. SYSTEM DEBUGGING AND SIMULATION

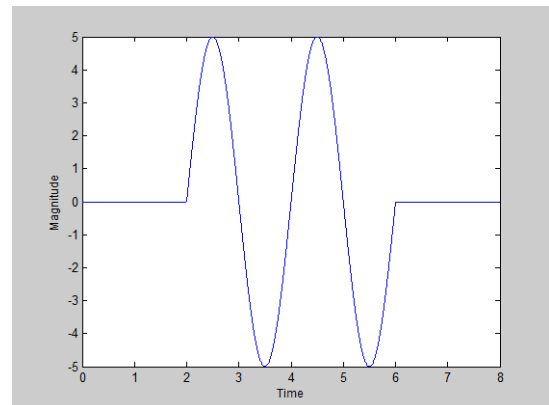


Fig. 4. Sinusoidal signal of the first road.

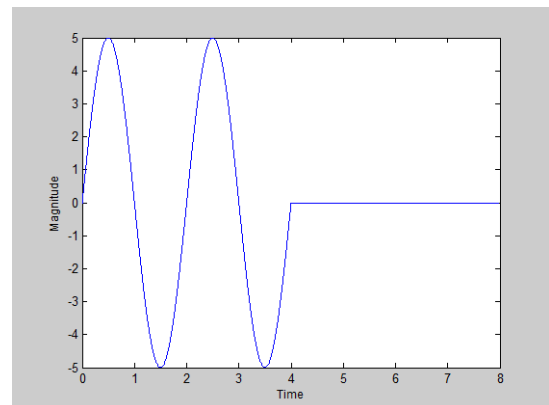


Fig. 5. Sinusoidal signal of the second road.

In order to verify the performance of the system, two

sinusoidal signals of two periods are generated by MATLAB as the input of system simulation. The signal amplitude is  $\pm 5V$ , the sampling frequency is 1MHZ, and the sampling time is 8ms. The two signals have a delay of 2ms, as shown in Fig. 4 and Fig. 5.

The result of the correlation operation is shown in Fig. 6. It is a sequence of length 15999. It can be seen that the peak of the correlation operation appears at the place of  $\tau=2000$ , that is, the discrete sequence Y of the second signal is slide 2000 to the right, and reaches the highest correlation with the discrete sequence X of the first signal. Because the sampling period is  $1\mu s$ , the delay difference is 2ms, and the results are in good agreement with the theory.

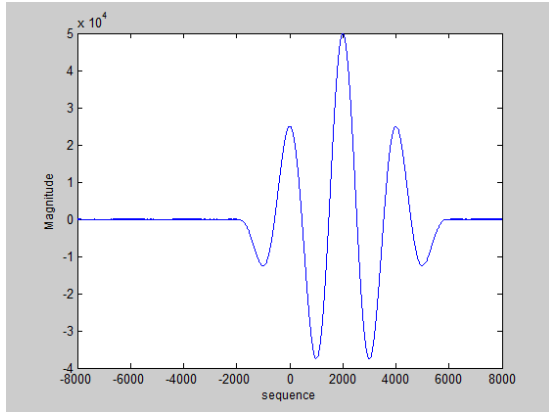


Fig. 6. Results of the system operation.

The period of correlation operations:

$$T_1 = 2 \times \frac{1000 \times (1000 + 1)}{2} \times 10ns = 10.01ms \quad (5)$$

The time period required to access single data using a single RAM address:

$$T_2 = 2 \times \frac{8000 \times (8000 + 1)}{2} \times 10ns = 640ms \quad (6)$$

That is,  $T_2$  is 64 times that of  $T_1$ , so an improved operation method can greatly reduce the time of operation, improve the speed of operation, and better meet the requirements of the system's real-time performance.

## V. SUMMARY

In this paper, the improved correlation algorithm has been successfully applied to FPGA, which has increased the operation speed by 64 times. Moreover, the time sequence is simple and easy to control, which can well meet the needs of practical engineering. In this paper, each RAM address accesses 8 data and 64 multipliers are needed for parallel computation. In fact, it uses the method of sacrificing chip size to exchange speed. If a faster speed is needed, a larger chip area is to sacrifice. If each RAM address accesses N data, according to the algorithm above,  $N^2$  multipliers and N accumulators should be configured, which will get a  $N^2$  times faster speed. In practice, a trade-off between area and speed can be achieved according to the needs of the system. This study provides a way to balance it exactly.

Because of the limited internal resources of the FPGA, if

we want to perform longer sequence correlation operations, we can use the off-chip large capacity memory DDR of the FPGA. DDR technology realizes two read/write operations in a clock cycle, so it does not cause large delay. This not only realizes the correlation operation of super-long sequence, but also guarantees the real-time requirement of the system.

## ACKNOWLEDGMENT

This work was supported by National Key Research and Development Plan (No. 2017YFB0403604) and Tianjin Key Laboratory of Optoelectronic Sensor and Sensor Network Technology.

## REFERENCES

- [1] H. Li, G. Song, Y. Li, *et al.* "The digital correlator design of FPASMR," in *Proc. 2014 IEEE Geoscience and Remote Sensing Symposium*, 2014.
- [2] K. A. Lukin, O. V. Zemlyaniy, D. N. Tatyanko, *et al.* "Noise radar design based on FPGA technology: On-board digital waveform generation and real-time correlation processing," in *Proc. IEEE 2017 18th International Radar Symposium (IRS)*, 2017.
- [3] V. R. Balu and S. M. R. Hasan, "Computationally minimized X-part for FX correlator in big-data interferometers," *IEEE Access*, vol. 5, p. 1, 2017.
- [4] Y. Zeng, Y. Yu, and L. Liu, "Realization of baseband signal processing for Beidou/GPS multi-mode receiver by FPGA," in *Proc. IEEE International Conference on Communication Software and Networks*, 2017, pp. 860-864.
- [5] S. H. Jin, J. U. Cho, D. R. Lee, *et al.* "An FPGA-based voice signal preprocessor for the real-time cross-correlation," in *Proc. 2007 International Conference on Control, Automation and Systems*, 2007, pp. 793-797.
- [6] S. O. Cisneros, J. Rivera, P. M. Villalobos, *et al.* "An image processor for convolution and correlation of binary images implemented in FPGA," in *Proc. 2015 12th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, 2015, pp. 1-5.
- [7] H. Lu, Z. Z. Wang, J. Y. Liu, *et al.* "Design of high-speed digital correlator in fully polarimetric microwave radiometer," *Acta Electronica Sinica*, vol. 1, no. 12, pp. 1-4, 2011.
- [8] T. Fersch, M. F. Alam, R. Weigel, *et al.* "A FPGA correlation receiver for CDMA encoded LiDAR signals," in *Proc. 2017 13th Conference on Ph.D. Research in Microelectronics and Electronics (PRIME)*, 2017, pp. 289-292.
- [9] T. Bhatia, A. Maji, J. Laha, *et al.* "Practical approach for implementation of phase correlation based image registration in FPGA," in *Proc. 2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*, 2018, pp. 1-4.



**Guiling Sun** received the Ph.D from Nankai University in 2004 and has become the professor of Nankai University. She is also the vice director of the Tianjin Institute of Communications, the vice director of EDA Technology Research Society in North China and the leader of Electronic Technology Experiment and Practice Teaching Team of Tianjin City. She has published more than 90 papers and 2 research monographs in a number of areas about the wireless sensor networks, internet of things, compressed sensing, information detection and intelligent control systems and signal processing.



**Jun Jia** was born in Hebei province, China, in 1994. He received his B.S. degree in microelectronics science and engineering from Nankai University, Tianjin, China, in 2017. He is pursuing his M.S. degree at Nankai University. His current research is digital signal processing based on FPGA.



**Tianyu Geng** was born in Tianjin City, China, in 1992. He received the B. S. degree in electronic science and technology from Nankai University, Tianjin, China, in 2014. And he is pursuing his Ph.D. degree in electronic science and technology in Nankai University, Tianjin, China. His research interests are in the areas of compressed sensing and image processing.



**Xudong Ye** received the B. S. degree in electronic information science and technology from Nankai University, Tianjin, China, in 2012. He is pursuing his M.S. degree at Nankai University. His current research is ethernet communication based on FPGA.