AlGORITHM IMPLEMENTATION OF ON-BOARD SAR IMAGING ON FPGA+DSP PLATFORM

Wenyue Yu
Beijing Key Laboratory of Embedded Real-time Information Processing Technology
Beijing Institute of Technology
Beijing 100081, China
yuwenyue_bit@bit.edu.cn

Yizhuang Xie
Beijing Key Laboratory of Embedded Real-time Information Processing Technology
Beijing Institute of Technology
Beijing 100081, China
xyz551_bit@bit.edu.cn

Dan Lu
Shanghai Institute of Satellite Engineering
Shanghai 201109, China
curl_xina@163.com

Bingyi Li
Beijing Key Laboratory of Embedded Real-time Information Processing Technology
Beijing Institute of Technology
Beijing 100081, China
libingyi_bit@bit.edu.cn

He Chen
Beijing Key Laboratory of Embedded Real-time Information Processing Technology
Beijing Institute of Technology
Beijing 100081, China
chenhe@bit.edu.cn

Liang Chen
Beijing Key Laboratory of Embedded Real-time Information Processing Technology
Beijing Institute of Technology
Beijing 100081, China
Beijing Institute of Technology
Chongqing Innovation Center,
Chongqing 401120 China
chenl@bit.edu.cn

***Abstract*—This paper introduces an effective parallel processing method to design the on-board SAR (Synthetic Aperture Radar) real time imaging processor using FPGA+DSP based on the high-resolution imaging algorithm. The architecture of this processor is designed based on the analysis of the algorithm operation characteristics and the inherent time relationship. In order to reduce the time consumption, pipeline and parallel joint processing method is applied. In addition, the system uses a combination of floating-point operations and fixed-point operations, which not only meets the imaging accuracy requirements but also saves the hardware scale of the system. The system requires 24s to focus the GF-3 stripmap SAR raw data with a granularity of 16384*16384 when works in 100MHz. The results demonstrate that our method was effective and the imaging quality can meet the requirements.**

Keywords—real-time, sar imaging, parallel processing architecture, algorithm mapping

I. INTRODUCTION

Spaceborne Synthetic Aperture Radar (SAR) has become an important means of Earth observation because of its ability to work all day, all weather and not affected by bad weather and intensity of care. The spaceborne SAR real-time imaging processor can directly convert the raw data into image data on-board, and the image data can reduce the amount of data by means of in-orbit compression or in-orbit object detection, so it can effectively solve the transmission bottleneck of high-resolution large-width SAR satellite. Therefore, the on-board real-time imaging processor is essential for spaceborne synthetic aperture radar applications.

At present, the Programmable Gate Array (FPGA), Digital Signal Processor (DSP) are, to some extent, superior with respect to SAR real-time imaging processing hardware platform. The DSP is used to implement various algorithms through software programming because of its flexibility and powerful floating-point data processing capability. But the system based on DSP has large volume, weight and power consumption due to its limited processing performance; FPGA is a component based on hardware program and refactoring architecture, it is very suitable to complete digital signal processing tasks such as multiplication and accumulation as its highly parallel processing capability. In this kind of operation, FPGA processing speed is an order of magnitude faster than DSP, but FPGA has no applicable to complex transcendental processing, and it is difficult to develop. Therefore, it is a key and effective method to realize real-time imaging processing of spaceborne SAR by using heterogeneous architecture based on FPGA and DSP which can combines the advantages of each processor. In this way, the system can meet the limitations of the on-orbit environment that a single processor system cannot meet.

In this paper we propose a hybrid heterogeneous parallel accelerating technique combining the advantages of both the FPGA and DSP to realize an on-board real-time SAR processing system. The system implementation results indicate that our design can potentially be used for spaceborne on-board real-time processing.

II. ALGORITHM REVIEW

The CS algorithm is suitable for high-resolution, large-swath imaging processing of spaceborne SAR. Also, the algorithm does not require interpolation processing, it can perform accurate imaging processing only by complex multiplication and FFT. The first step, the algorithm transforms the original echo signal into the range Doppler domain by azimuth FFT, and then multiply with the CS compensation factor φ1 to compensate the distance migration curves on all distance gates to the same shape; The second step, the algorithm transform the signal into the two-dimensional frequency domain, and then multiply

This work was supported in part by the Chang Jiang Scholars Program under Grant T2012122 and in part by the Hundred Leading Talent Project of Beijing Science and Technology under Grant Z141101001514005.

978-1-7281-2345-5/19/$31.00 ©2019 IEEE

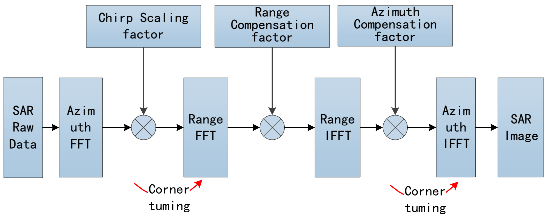

Fig. 1. Chirp scaling algorithm flowchart

by the distance compensation factor φ2 in order to complete the distance migration curve correction， the second distance compression and the distance compression; The third step, the algorithm transform the signal back into the distance Doppler domain through IFFT , and then the azimuth compensation factor φ3 is multiplied to the result of the IFFT to complete the azimuth processing; The last step, Transform the signal back to the time domain to obtain a SAR image. We know from the algorithm flow that the operations involved in the algorithm mainly include: FFT/IFFT operation, three CS factor generation, multiplication. The CS imaging processing algorithm flow is shown in Figure 1

It can be seen from the above algorithm flow that the CS algorithm is divided into two parts: the main flow and the auxiliary flow.

The main flow is a linear operation of the data vector, which is executed sequentially. It mainly includes data reading and writing operation, FFT/IFFT process and multiplication. The auxiliary flow focus on the CS compensation factor generation, including the calculation of the CS compensation factor in three stages of the algorithm. The main process and the auxiliary process can be processed in parallel to improve system timeliness.

III. SYSTEM ARCHITECTURE

This chapter designs a parallel heterogeneous architecture based on FPGA+DSP. FPGA is used to implement the main flow processing of CS algorithm. DSP is used to implement the auxiliary process of CS algorithm. The architecture is shown in Figure 2.

The processing flow of the architecture is as follows: The I/O unit is the interface of the system, it is responsible for receive the SAR radar echo data, and then transmit the data to the DDR control unit, the DDR control unit keeps the data in DDR in a certain rule through the data bus on the board. After receiving the SAR raw data, the control unit sends a data distribution command and a CS factor generation command to DDR control unit and DSP. After receiving the command, the DDR control unit reads two pieces of data from the DDR in the azimuth mode and distributes the data to the processing engine through the data bus. The processing engine unit includes two FFT, it can process the data parallel. After receiving the CS factor generation command, the DSP starts the corresponding CS factor operation and internally buffers the generated factor. The DSP distributes the generated factor to the processing engine unit for complex multiplication processing according to the interrupt requirement. When the processing engine unit completed the processing of the two pieces of data, the flay will be sent to the control unit, and the control unit sends a data storage instruction to DDR control unit, and then the DDR control unit reads the processing result of the processing engine and stores the data to DDR according to the original position. In this way, the processing engine unit completes the processing of the two lines data at a time. The entire processing is completed by the loop multiple times, and the processing result is output to the external system to display through the IO interface. The FPGA of this architecture is XX6VFX130T from Xilinx company, and the DSP is TS201 from ADSP company.

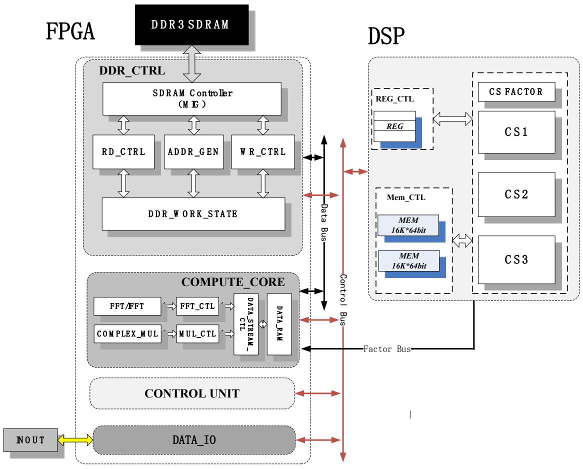

Fig 1 System architecture block diagram

IV. ALGORITHM MAPPING

A. *Design of fixed-point pipeline FFT*

In order to implement the FFT processing engine, on the one hand, the appropriate circuit structure is determined by the application requirements: the optimal balance between processing delay, throughput and circuit scale; on the other hand, the appropriate data format is also important to ensure the accuracy of the FFT operation.

In order to meet the above requirements, this paper designs a base-2^k SDF structure FFT. The base-2^k FFT processing unit requires fewer multiplication units and fewer twiddle factor storage units than base-2 and base-4 FFT processing units, so that it can effectively reduce the occupation of hardware resources. Figure 3 shows the -2^k SDF FFT basic operation unit. A 16k Point FFT consist of four -2^3 basic operation unit and two -2^2 basic operation unit. Each -2^3 basic operation unit is composed of three processing units (PE1, PE2 and PE3). PE1, PE2 and PE3 Each -2^2 basic operation unit is composed of three processing units (PE1 and PE3). PE1, PE2 and PE3 PE1,

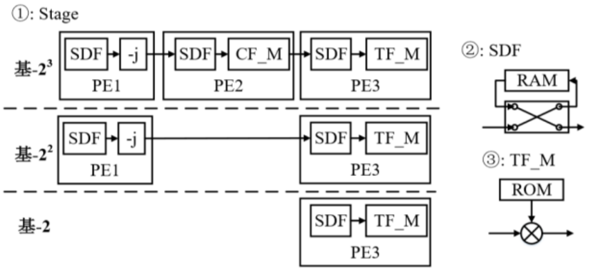

Fig3 FFT basic operation unit

PE2 and PE3 all contains a base-2 butterfly unit, a single-channel delay feedback (SDF) module which contains a feedback storage RAM. For the i-th level (i=1, 2,..., log8N), the RAM depths in the three processing units are $M/2^{3i-2}$, $M/2^{3i-1}$and $M/2^{3i}$; PE1 realize real imaginary part data exchange processing through a operation of -j multiplication; PE2 contains a constant factor multiply module CF_M for completing constant factor multiplication operation, and PE3 contains a twiddle factor multiplication operation module TF_M. Each level of twiddle factor is implemented by a lookup table. Because of the symmetry of the sine and cosine functions, each lookup table only needs to store the sine and cosine values of the 1/4 quadrant angle. For the ith stage, the ROM depth for storing the twiddle factor is $M/2^{3(i-1)}/4$.

Considering the accuracy requirements of the system, the FFT designed in this paper adopts fixed-point implementation, and the real and imaginary parts are 24 bits, which not only meets the accuracy requirements but also reduces the hardware resource overhead.

B. *Design optimization of CS factor generation*

Among the CS algorithm, there are three kinds of compensation factors, and the calculation results of each compensation factor need to be multiplied with the corresponding FFT/IFFT results in the main flow to complete the factor compensation operation. The compensation factor is complicated to calculate, and the calculation speed of DSP cannot keep up with the speed of FFT/IFFT processing, which causes the interruption of system pipeline processing. The interruption of the pipeline will greatly increases the processing time of the algorithm and seriously affects the real-time performance of SAR imaging processing system. Therefore, this paper adopts the method based on fixed step factor update. Within a fixed step size, the system only calculates the compensation factor once and stores the factor in DSP memory, thus effectively reducing the calculation time of the compensation factor. In this paper, we choose the fixed step size eight which can reduce 8x compute complexity and get a 8x performance improvement.

Considering that the three phase factor generation processes are roughly the same, they can be summarized into similar functional forms [2],so we design a large of general hierarchical software processing and shared buffer modules. Different phase factor can generate by calling different processing modules. In this way, we can effectively reduce software design complexity and storage resource consumption.

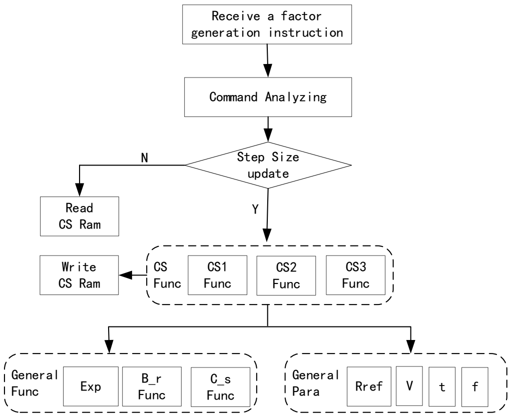

Fig.4 CS factor generation unit software flow

Figure 4 shows the software processing flow of CS factor generation unit. After the DSP receives the factor generation instruction, it first performs command parsing to confirm which stage CS factor will be generated, and then determines whether it is within the fixed step size. If yes, it directly reads the pre-stored factor output. If not, it needs to be performed corresponding factor calculation. The three CS factor has its independent functions, and the functions are mainly realized by calling the general hierarchical software processing and shared buffer modules.

Since the calculation accuracy of the factor is a mainly element that affect the accuracy of the algorithm, the factor generation is implemented in a single precision floating point form.

C. *Corner Turning Memory Access*

For the CS algorithm, the corner turning is needed before each step to fit the step's data operation dimension Therefore, three matrix transpose operations are required during the total processing procedure. Matrix transpose operations is in the main flow of the algorithm, so its transposition efficiency directly affects the processing efficiency of the system. Therefore, the proposed design introduced the submatrix cross-mapping method to improve the efficiency of corner turning.

The main procedure of this technology is shown in Figure 5, according to the following two steps: 1.Divide the NA ×NR raw data matrix into M×N submatrixes of Na ×Nr order; 2. Map each submatrix into the three-dimensional storage array of the DDR. Each submatrix corresponds to a row of the DDR. 3 Access data as range or azimuth mode.

Range access mode: This mode corresponds to the overall data operation before the second co-processors stage. Take the first range reading operation line as an example. As shown in Figure 5. In this time, two range data are access at the same time. The data are distributed in the coordinates (0,0) to (0,N-1) of the sub-matrix $A_{0,0}$ to $A_{0,\,N-1}$, it is mapped to the DDR SDRAM matrix space of the thN rows and the Bn banks according to the cross mapping method. According to the DDR burst read mode, each half of the burst length m data is divided into two adjacent distance data. Therefore, it takes $N_{R/m}$ cycles to read the two distances of the NR elements.

Azimuth access mode: This mode corresponds to the

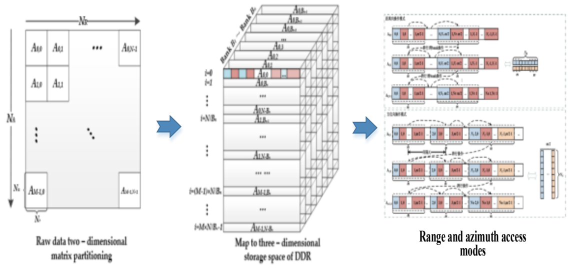
NR
NA
Na
Nr
A0,0
A0,1
A0,N-1
A1,0
A1,1
AM-1,0
AM-1,N-1
Bank B0 ~ Bank Bn
i=0
i=1
i=N/Bn
i=(M-1)×N/Bn
i=M×N/Bn-1
Raw data two – dimensional matrix partitioning
Map to three – dimensional storage space of DDR
Range and azimuth access modes

Fig.5 Corner Turning Memory Access flow

overall data operation before the first and third co-processors stages. Take the m/2 azimuth lines reading operations as an example. These azimuth data are distributed in the 0 to m-1 columns of sub-matrices $A_{0,0}$ to $A_{M-1,\ 0}$. These data are mapped to DDR. $A_{0,0}$, $A_{1,0}$,...,$A_{M-1,\ 0}$ lines, according to the mapping method, these data are mapped in thN rows and in the same bank. Under the DDR burst access mechanism, each burst of m data belongs to m/2 azimuth line data, and the data is not continuously read in each row of DDR, and there are Nr data between each two groups of m data. At intervals, m data is read every clock cycle, and after NA clock cycles, m/2 azimuth data with depth NA is simultaneously read.

V. IMPLEMENTATION

A. *Parallel processing analysis of the system*

In this paper, we apply multi-level parallel processing ways to improve the performance of the system. The first level is the FPGA parallel acceleration of fine-grained vector-data computation. Two parallel FFT/IFFT units and multi-multiplication units are set inside the FPGA, which can process two data at the same time; the second level is heterogeneous parallel combine FPGA and DSP, FPGA implements the main flow of the algorithm, DSP implements the auxiliary process of the algorithm, the main and auxiliary computation can be processed simultaneously.

After the system is powered on, FPGA receives the original echo data and buffer the data to DDR to store. After receive one data-block, the system starts CS imaging processing, and output the result when the processing procedure is completed. The six steps of this procedure are as follows:

- Step 1: Load two range or azimuth lines of data from the DDR to the compute_core unit;
- Step 2: Start the parallel processing modules FFT/IFFT and DSP module at the same time, and then the result of the two modules are multiplied.
- Step 3: Collect the data from the compute_core module and send them to the DDR;

The parallel processing timelines of one round-robin assignment for both the range direction operation and the azimuth direction operation are described in figure 6.

B. *System verification*

In this paper, we design a platform using a xc5vfx130t FPGA, a TS201 DSP and 2GB DDR2 SDRAM. FPGA works in 100MHz, DDR SDRAM works in 400MHz, and the main frequency of DSP is 600MHz. Trough recording the numbers of clock cycles, it takes 24s for the system hardware to process SAR raw data with 16383*16384 data granularity. The hardware platform and the imaging results of the hardware are shown in Figure 7, and the processing accuracy can meet the application requirements.

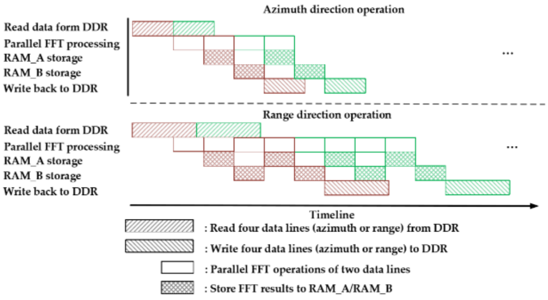
Azimuth direction operation
Read data form DDR
Parallel FFT processing
RAM_A storage
RAM_B storage
Write back to DDR
...
Range direction operation
Read data form DDR
Parallel FFT processing
RAM_A storage
RAM_B storage
Write back to DDR
...
Timeline
: Read four data lines (azimuth or range) from DDR
: Write four data lines (azimuth or range) to DDR
: Parallel FFT operations of two data lines
: Store FFT results to RAM_A/RAM_B

Fig 6 Parallel processing timeline of one round-robin assignment

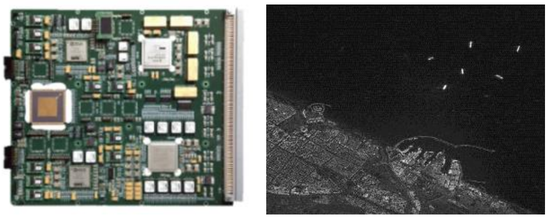

Fig 7 The system hardware and the Imaging result.

VI. CONCLUSION

In this paper, we design an FPGA+DSP heterogeneous parallel processing architecture that can efficiently implement the algorithm based on the analysis of CS algorithm. The system combines pipeline and parallel processing ways according to the time relationship of each operation of the algorithm, which greatly improves the processing efficiency. At the same time, the system combines fixed-point operation and floating-point operation according to different requirements of the algorithm to reduce hardware consumption. Finally, the real-time imaging processing of the heterogeneous parallel processing module is verified with GF3 spaceborne radar data, and the imaging quality can meet the application requirements.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grant 91738302 and in part by the Hundred Leading Talent Project of Beijing Science and Technology under Grant Z141101001514005.

REFERENCES

[1] Franceschetti, G., Lanari, R.: Synthetic Aperture Radar Processing. In: Geoscience and Remote Sensing Symposium, 1994. Igarss '94. Surface and Atmospheric Remote Sensing: Technologies, Data Analysis and Interpretation., International1999, pp. 1961 - 1963.

[2] Yang,C.; Xie,Y.H.; Chen,H.; Ma,C.M.Ahigh-precisionhardware-efficientradix-2(k)FFTprocessorforSAR imaging system. IEICE Electron. Express 2016, 13, 10. [CrossRef].

[3] Bierens, L.; Vollmuller, B.J. On-board Payload Data Processor (OPDP) and its application in advanced multi-mode,multi-spectralandinterferometricsatelliteSARinstruments.InProceedingsof the9thEuropean Conference on Synthetic Aperture Radar, Nuremberg, Germany, 23–26 April 2012; pp. 340–343.

[4] LIU H, LEE H. A high performance four-parallel 128/64-point radix-2 4 FFT/IFFT processor for MIMO-OFDM systems[C]. Circuits and Systems, 2008 APCCAS 2008 IEEE Asia Pacific Conference on. 834-7.

[5] Zhou Yinqing, He Min, Chen Jie, et al. Implementation of Chirp Scaling algorithm based on space-borne real-time SAR processor[J]. Journal of Beijing University of Aeronautics and Astronautics, 2005, 31(2):142-147.(in Chinese)

[6] Wang Liang, Wang Jian, Huang Xiaotao. Research on phase compensating factors in chirp scaling algorithm[J]. Journal of Electronics and Information Technology, 2007,29(1):6-11.

[7] I J, LIU F, LONG T, et al. Research on pipeline R22SDF FFT[C]. Radar Conference, 2009 IET International. 1-5.

[8] JABER M A, MASSICOTTE D. A new FFT concept for efficient VLSI implementation: Part II - Parallel Pipelined Processing[C]. International Conference on Digital Signal Processing. 1-6.

[9] Liu Xiaoning, Chen He, Xie Yizhuang. Research and Implementation of CTM for Real-Time SAR Imaging Processing[C]. IET International Radar Conference 2013. Xian, IET, 2013: 0228.

[10] Gu C F, Chang W G, Li X Y, et al. Multi-core DSP based parallel architecture for FMCW SAR real-time imaging[J], 2015, 24(4): 1084-1090.